SPEEDTRONIC
Mark V
Turbine Control

Application Manual

SPEEDTRONIC
Mark V
Turbine Control

Application Manual

GEH-6195F

Issue Date: June 25, 1993
Revision A: July 21, 1993
Revision B: June 1994
Revision C: September 1997
Revision D: February 1998
Revision E: March 2003
Revision F: May 2003

Printed in the United States of America.

Print Date: May 29, 2003

Safety Symbol Legend

WARNING

Commands attention to an operating procedure, practice, condition or statement, which, if not strictly observed, could result in personal injury or death

CAUTION

Commands attention to an operating procedure, practice, condition or statement, which, if not strictly observed, could result in damage to or destruction of equipment.

NOTE

Commands attention to an essential operating or maintenance procedure, condition, or statement that must be highlighted.

WARNING

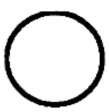
This equipment contains a potential hazard of electric shock or burn. Only personnel who are adequately trained and thoroughly familiar with the equipment and the instructions should install, operate, or maintain this equipment.

Isolation of test equipment from the equipment under test presents potential electrical hazards. If the test equipment cannot be grounded to the equipment under test, the test equipment's case must be shielded to prevent contact by personnel.

To minimize hazard of electrical shock or burn, approved grounding practices and procedures must be strictly followed.

To prevent personal injury or equipment damage caused by equipment malfunction, only adequately trained persons should modify any programmable machine.

Safety Symbol Legend (cont.)

Symbol	Publication	Description
	IEC 417, No. 5031	Direct current
	IEC 417, No. 5032	Alternating current
	IEC 417, No. 5033	Both direct and alternating current
	IEC 617-2, No. 02-02-06	Three-phase alternating current
	IEC 417, No. 5017	Earth (ground) TERMINAL
	IEC 417, No. 5019	PROTECTIVE CONDUCTOR TERMINAL
	IEC 417, No. 5020	Frame or chassis TERMINAL
	IEC 417, No. 5021	Equipotentiality
	IEC 417, No. 5007	On (Supply)
	IEC 417, No. 5008	Off (Supply)
	IEC 417, No. 5172	Equipment protected throughout by DOUBLE INSULATION or REINFORCED INSULATION (equivalent to Class II of IEC 536)
	ISO 3864, No. B.3.6	Caution, risk of electric shock
	ISO 3864, No. B.3.1	Caution

Notes:

TABLE OF CONTENTS

Section/Subject **Page**

Section/Subject **Page**

Section/Subject **Page**

Section/Subject **Page**

Section/Subject **Page**

Section/Subject **Page**

CHAPTER 1

INTRODUCTION

1-1. ORGANIZATION OF DOCUMENTATION

Documentation for the Mark V turbine control system consists of two types: unit-specific drawings and instruction books. A unique set of requisition-specific documentation is supplied with each control system (see Section 1-1.1) and instruction books are available for the specific needs of each user (see Section 1-1.2).

1-1.1. Requisition Specific Drawings

Requisition or unit specific drawings are provided by various sources with each Mark V Turbine Control System. General Electric Drive Systems (GEDS) Turbine Products Division provides drawings to describe the hardware and software configuration for each requisition, including:

- **I/O Report** contains the unit-specific assignment of I/O terminations in the Mark V control panel. This report also has I/O related information such as the signal names, scale type, cabling information, termination points, and device nomenclature.
- **Control Sequence Program Printout** is a unit-specific printout that shows a functional representation of the Big Blocks and sequencing of a particular requisition. Software on the operator interface allows editing and printing of this document from any location.
- **Outline Drawings** provide an external view of the control panel and primary operator interface. The drawings furnish information needed for handling and installing the equipment.
- **Case Layout Drawing** supplies an internal view of the control panel. The primary purpose of this drawing is to furnish information needed to route interconnect cables.
- **Case Wiring Drawing** defines the factory cabling internal to the control panel case. The drawing's primary purpose is to document the internal wiring for maintenance use.
- **Core Drawings** provide an isometric drawing of the core depicting the cards and their respective locations within the core. For each card, the physical location and identification of removable parts, such as connectors and hardware jumpers, is highlighted. The core drawing is placed in a pocket on the inside of the core door.

Additional documentation is provided by the turbine manufacturer directly to the customer.

1-1.2. Instruction Books

The manuals provided by GEDS for the Mark V Turbine Control System are designed to meet the special needs of operators, maintenance personnel, and application engineers.

- For the operator using Mark V, *Mark V User's Manual, GEH-5979*
- For the operator using HMI, *HMI Turbine Control Operator's Guide, GEH-6126 Vol.I*
- For the maintenance technician, *Mark V Maintenance Manual, GEH-5980*
- For the maintenance technician using HMI, *HMI Turbine Control Application Guide, GEH-6126 Vol.II*
- For the application engineer, *Mark V Application Manual*, *GEH-6195*

1-1.2.1. USER'S MANUAL. The user's manual provides information needed by a turbine operator to understand both the primary and back-up Mark V operator interfaces. Topics in the manual include:

- Main Menu and Display
- PASSWORD Administration
- Synonyms
- Alarm Management
- User-Defined Displays
- Trip Log Display
- EPA Display
- Back-up Operator Interface Operation
- Printer Functions
- Multi-Unit Operator Interfaces

1-1.2.2. MAINTENANCE MANUAL. The maintenance manual provides information needed by control system maintenance personnel for installation, calibration, and troubleshooting the Mark V control system. Topics in the manual include:

- Control System Installation
- Control Constant Adjustment
- Dynamic Rung Display
- Logic Forcing
- Pre-voted Data Display
- LCC Operation
- Terminal Interface Monitor Operation
- DIAGC Display Operation
- VIEW Tools

1-1.2.3. APPLICATION MANUAL. The application manual is an engineer's reference for the Mark V control system. Topics in the manual include:

- Introduction To Mark V Controls
- Specifications & I/O Capacities
- The Screen Builder
- The Control Sequence Editor
- I/O Application Examples
- Regulator Descriptions & Diagrams
- Stage Link Application Rules
- MODBUS Configuration Instructions
- The I/O Configurator
- Signal Flow Diagrams
- Hardware Jumper Application Notes
- Big Block Reference

1-2. MARK V TURBINE CONTROL PRODUCT OVERVIEW

Turbine Control Systems have been produced for several decades and have enjoyed widespread acceptance in both new unit and retrofit applications. The Mark V represents the latest in a line of microprocessor-based turbine control systems designed specificaly for controlling turbines. The Mark V can be used on medium or large steam turbines, heavy duty gas turbines (single or two shaft), and aircraft derivative gas turbines.

Unit control and protection is accomplished by using the Mark V in combination with sensors and devices mounted on the unit and its auxiliaries. Unit reliability is improved by using redundant sensors and devices for feedback, control, and protection of critical functions. Should one of the redundant devices fail, operation is not adversely affected. The connection of redundant devices to the control panel and their regulation by the control software were considered to be crucial factors in designing the Mark V. This fail-safe approach results in a highly reliable control and protection system for the turbine.

In its most common configuration, the Mark V further improves unit reliability by using three redundant control processors. This triple modular redundant (TMR) design is capable of safely operating, controlling, and protecting a unit in the event of the failure of one of its control processors or control processor components. The TMR design permits a single control processor to be shutdown and repaired without shutting the turbine down.

Another attribute of the Mark V TMR control system is its use of software-implemented fault tolerance (SIFT) technology. Each control processor in a TMR control panel makes its own determination of control and protection functions based on separate inputs. The control processors individually vote the inputs used to make these determinations. Should one control processor fail to read an input correctly, its erroneous value would be "out-voted."

The following example illustrates the manner in which SIFT voting is enacted by the control system: a logic signal (either a logic " 0 " or a logic " 1 "), representing a digital input from a single pressure switch that senses lube oil pressure, is communicated to each of the three redundant processors in a TMR control panel over individual I/O communication networks (IONETs). Each processor accepts what it believes the value of the logic signal to be (the pre-voted value) then communicates that value to the other two processors over a single data exchange communication network (DENET). Each processor then performs a two-out-of-three "vote" of the digital input's logic signal value and uses the voted value in its control and protection algorithms/sequencing. Therefore, a failure does not result in a turbine trip signal being generated by that processor. (The condition described above is reported as a voting mismatch Diagnostic Alarm.)

The SIFT voting technique will tolerate multiple failures without initiating a turbine trip. For example, one control processor might determine that a turbine trip should be initiated as the result of a low lube oil pressure switch input and a second control processor might determine that the turbine should be tripped on a high exhaust temperature based on a faulty thermocouple input. Without SIFT, the two control processors initiate a turbine trip generated by two different input devices. However, using SIFT, the control processors use the voted values of the inputs and do not initiate a turbine trip.

Another feature, Control Lockout, places the primary operator interface into a **view only** mode (unless control capability is turned on with the correct password.

1-2.1. Primary Operator Interface, <I>

The Mark V Turbine Control System's primary operator interface <I> consists of an IBM-compatible personal computer (PC), color CRT, keyboard, cursor positioning device (CPD), either touchscreen CRT and/or trackball or mouse, and a printer. The <I> is used to issue commands to start/stop the unit, load/unload the unit, manage and log alarms, and monitor unit operation. With the exception of the Plant Load Control option, no control or protection of the unit is accomplished by the <I>. It is simply an operator's/technician's interface to the Mark V control panel(s) with which it communicates.

<I>s are connected to a Mark V turbine control panel(s) with coaxial cable using ARCNET LAN (Local Area Network) communication-style interface. This connection between <I>s and Mark V control panels is called the Stage Link. In some cases, the Stage Link may include fiber optic cables and repeaters in order to accommodate long distances between the <I> computer(s) and the turbine control panel. Figure 1-1 shows an installation in which three <I>s are used to control two turbines and their driven devices.

An <I> can also be used to configure or modify the control, protection, monitoring, and logging functions of the Mark V Turbine Control System using programs supplied on the <I> computer. The ability to modify or configure these Mark V functions is password protected. Options available for the <I> include color printers and laser printers.

The Mark V control system has powerful features for customizing control strategy for each site. For example, one <I> can interface with up to eight gas or steam turbines (or any combination thereof). In addition, more than one <I> can be used (each interfacing with up to eight turbines or a subset of the eight, if desired). A hierarchy of control can be programmed on-site when multiple <I>s are used.

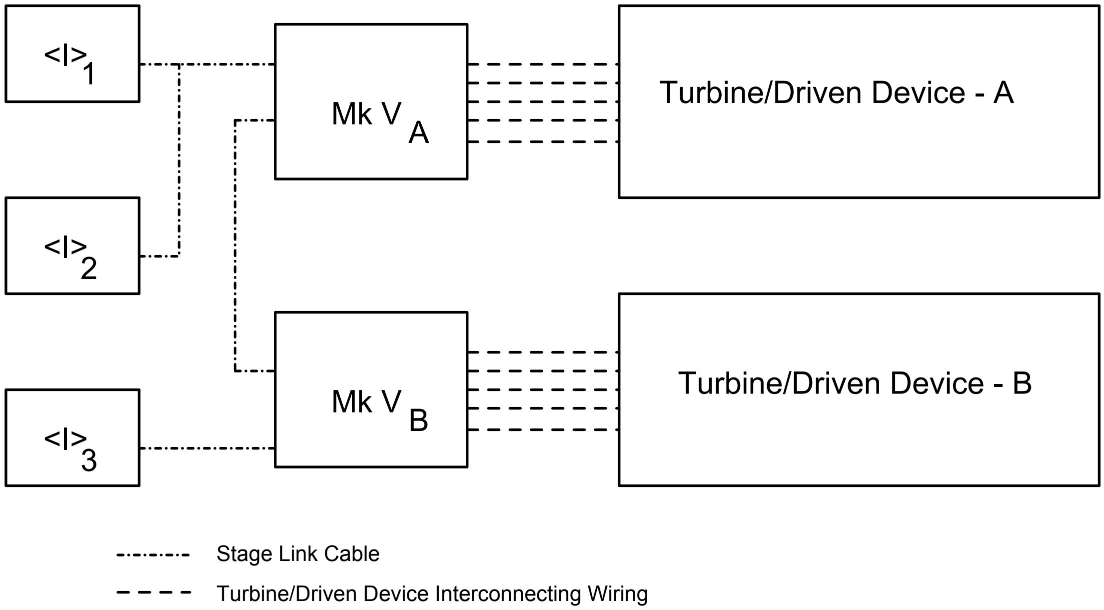


Figure 1-1. Multi-Unit Installation Employing Three <I>s

1-2.2. Backup Operator Interface Panel - <BOI>

The Mark V System also provides a secondary means of monitoring/controlling the turbine functions. This ancillary device is known as the Backup Operator Interface or <BOI>. The <BOI> has its own communications link which is directly connected to the three control processors <R>, <S>, and<T>.

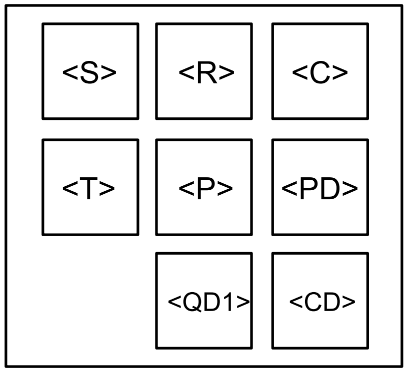
An LCD panel with a keypad, this device is usually mounted on the control panel. It also can be used to start and stop the unit, load or unload it, silence acknowledge alarms, reset process alarms, and monitor unit operation.

1-2.3. Control Panel Configurations

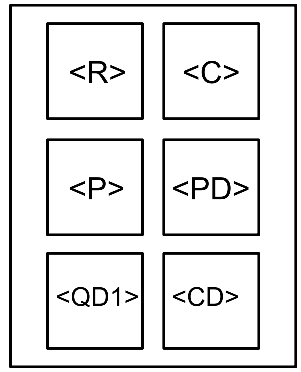
The Mark V control panel is supplied in one of two configurations: triple modular redundant (TMR) or single modular (Simplex). Refer to Figure 1-2.

New gas turbine units frequently use a TMR control panel, but may also be equipped with a Simplex control panel. Existing gas turbine control system retrofit applications can be equipped with either a TMR or a Simplex control panel. New steam turbine units can be equipped with either a TMR or Simplex control panel. Existing steam turbines can also be retrofitted with either TMR or Simplex control panels.

Printed circuit cards and terminal boards in a Mark V control panel are contained in or are mounted on **cores**. Cores are sheetmetal housings that can have stationary and movable printed circuit card holders called card carriers. The cores have a maximum of five printed circuit cards mounted on the card carriers. In addition, up to four I/O terminal boards (printed circuit cards with high-density terminal boards) can be mounted on a single core.



Mark V TMR Control Panel



Mark V SIMPLEX Control Panel

<C> - Communicator Core

<S> - Redundant Control Processor Core

<P> - Protective Core

<QD1>- Digital I/O Core for Control Processor(s)

<R> - (Redundant) Control Processor Core

<T> - Redundant Control Processor Core

<PD> - Power Distribution Core

<CD> - Communicator Digital I/O Core

Notes:

CHAPTER 2

SPECIFICATIONS

2-1. INTRODUCTION

This Chapter provides product specifications for the Mark V control panel. The panel construction meets NEMA type 1 and utilizes natural convection cooling, circulating air via lower and upper vents. these vents are located in the forward facing doors. Panel material consists of hot-rolled, low carbon, 12-gauge steel (.105") with one exception, the access plates use thicker gauge (.187") steel.

The standard paint process takes into consideration current Environmental Protection Agency (EPA) requirements. The Mark V paint process employs a powder coating process which includes a multi stage pretreatment section, an environmentally controlled structure for the coating booths, and a curing oven. The pretreatment section cleans, removes oxide scale, provides corrosion protection, and promotes uniform adhesion of the powder coat. The pretreated metal parts are then electrically grounded and passed through a cloud of negatively charged powder, completely covering the part with powder. The parts then enter an oven to “flow” the powder and cure the finish. Three major benefits of Powder Coating are:

- A very high transfer efficiency rate of 98% verses the more typical 40-60% usually obtained when spraying liquid.
- Powder coating reduces emissions of Volatile Organic Compounds.
- Powder coating increases the quality and improves the appearance of the Mark V.

The Mark V panel has three basic physical configurations, Simplex, TMR and Large Steam TMR. The following table summarizes the weights and dimensional information. Outline drawings for the Simplex and TMR are on the next two pages. Because the large steam TMR panel is constructed from two standard TMR panels a separate outline drawing is not provided.

PANEL TYPE:	WEIGHT		HEAT DISSIPATION	DIMENSIONS (Width x Depth x Height)	
	(LBS)	(KG)	(WATTS)	(INCHES)	(MM)
SIMPLEX	800	362.9	500	36 X 20 X 90	914 X 508 X 2286
TMR	1000	453.6	700	54 X 20 X 90	1372 X 508 X 2286
LARGE STEAM TMR	2000	907.2	1000	108 X 20 X 90	2744 X 508 X 2286

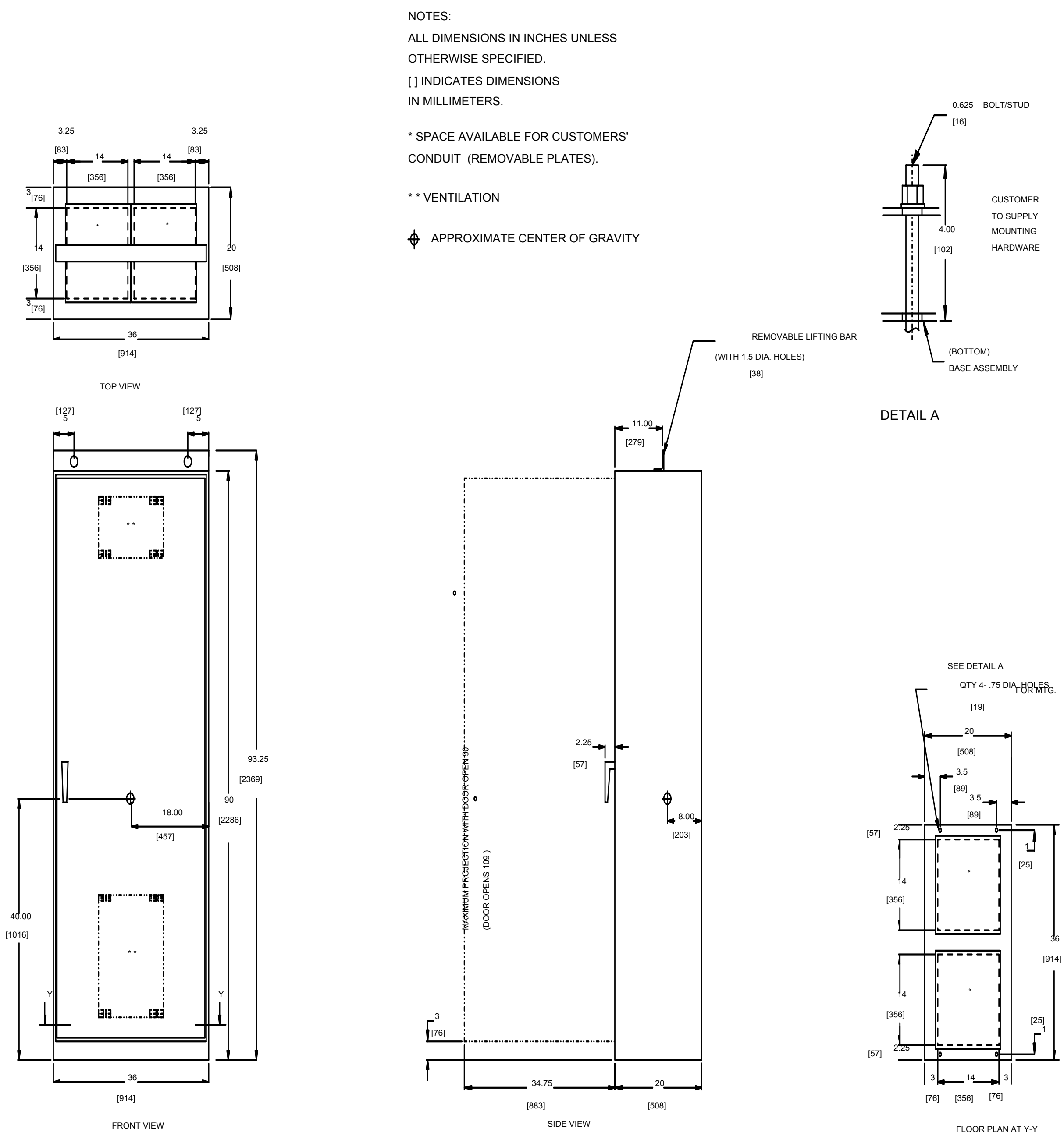


Figure 2-1. Typical Mark V Simplex Panel Outline Drawing

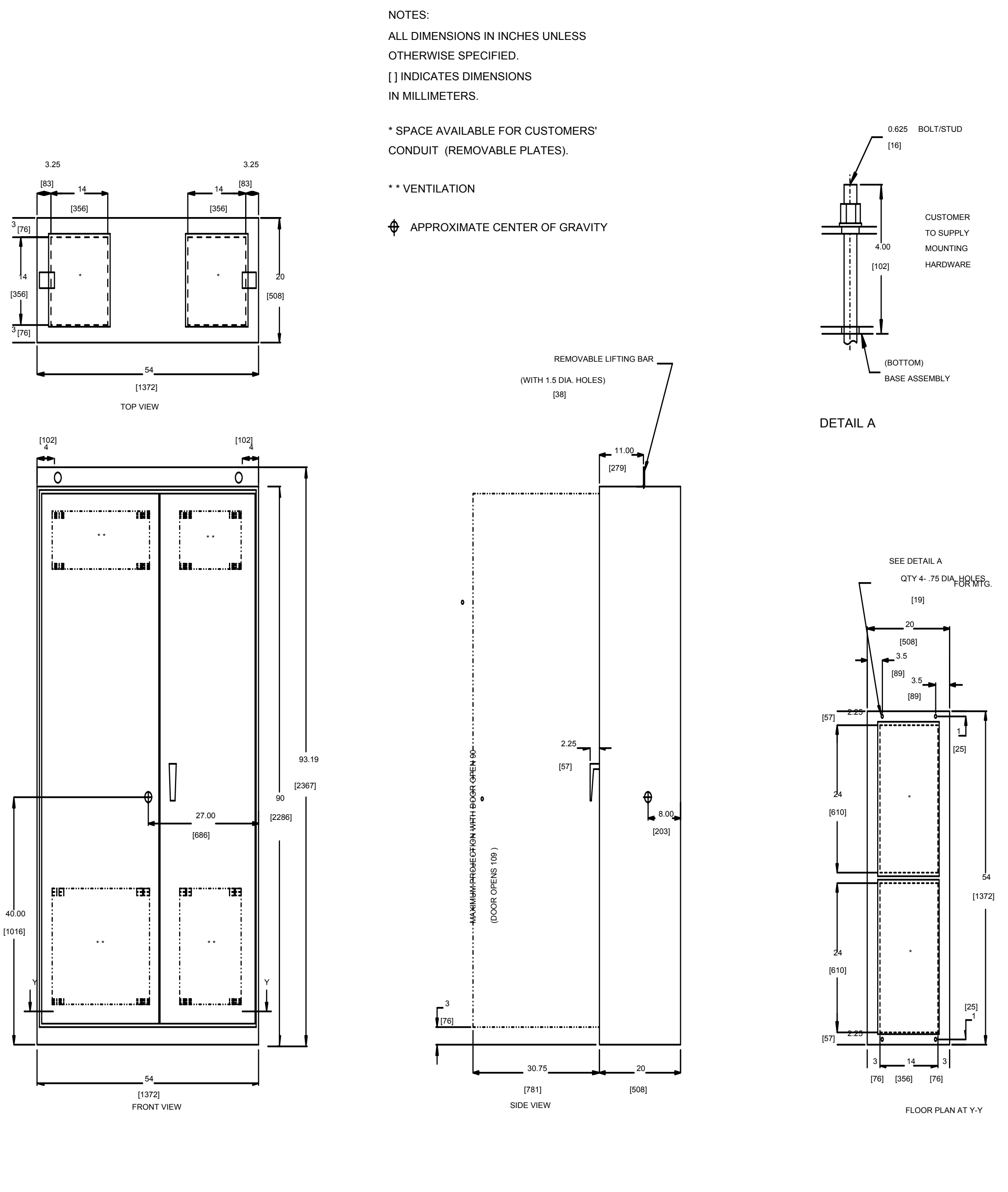


Figure 2-2. Typical Mark V TMR Panel Outline Drawing

2-2. LOCATION AND INSTALLATION

The panel is of NEMA Type 1 construction and must be located in an area where conditions meet the requirements as shown in the table below. The preferred location would be in an equipment room adjacent to the control room or in the control room itself. The panel must be mounted where the floor surface allows for attachment in one plane (ie. a flat, level and continuous surface). The outline drawings on the previous pages depicts the panel secured at its base by four 0.625 x 4.00 inch studs which is the recommended approach for floor attachment. The mounting hardware is provided by the customer. Lifting lugs are provided and if used, the lifting cables must not exceed 45° from the vertical plane. Finally, the panel is equipped with a door handle which can be padlocked for security.

Interconnecting cables can be brought into the panel from the top or the bottom via removable access plates. Convection cooling of the panel requires that conduits be sealed to the access plates. Also, air passing through the conduit must be within the acceptable temperature range, as shown in section 2-3. This applies to both top and bottom access plates.

2-3. ENVIRONMENT

ENVIRONMENTAL REQUIREMENTS DURING OPERATION		
Ambient Temperature	0 — 45 °C	
Relative Humidity	5 — 95 %	
Seismic Capability	Designed to Universal Building Code (UBC) - Seismic Code section 2312 Zone 4	
Vibration	1.0 G horizontal, 0.5 G vertical at 15 to 120 Hz.	
Surge	Designed To ANSI C37.90.1	
Radio Interference	Operation of a 5 W radio transmitter at 27 mHz, 150 mHz and 480 mHz will not disrupt operation. Note: Panel doors must be closed	
Altitude	2000 meters maximum	

The preceding table depicts the primary environmental requirements during operation. The 45 degrees Centigrade ambient operating limit for the panel is based upon the commercial component specification design limits of 70 degrees Centigrade. The NEMA 1 panel has been designed for a +15 degrees Centigrade maximum rise from inlet to exhaust air temperature, plus another +10 degrees Centigrade maximum rise for a ‘hot spot’ at the component / circuit board device level.

Temperature and vibration values have different allowable values during shipping and storage. The temperature requirements during shipping and storage are -40 to +70 degrees Centigrade. The tested values for vibration during shipping are 72 hours at 0.3 G rms between the frequencies of 4 to 16 Hz, and 3 shocks of 15 G, each as a 2 millisecond impulse and repeated for all three axes.

2-4. CONTAMINANTS

The Mark V control equipment is designed to withstand 10 parts per billion (PPB) of the following contaminants:

- Reactive Sulfur
- Reactive Chlorine
- Hydrogen Sulfide
- Sulfur Dioxide
- Chlorine Dioxide
- Sulfuric Acid
- Hydrochloric Acid
- Hydrogen Chloride
- Ammonia

2-5. CODES and STANDARDS

The control is designed to operate within the constraints and conditions specified by the following:

UL 508A Safety Standard Industrial Control Equipment

CSA 22.2 No. 14 Industrial Control Equipment

CE - Electromagnetic Compatibility (EMC) Directive 89/336/EEC

EN 50082-2:1994	Generic Immunity Industrial Environment
EN 61000-4-2	Electrostatic Discharge Susceptibility
ENV 50140:1993	Radiated RF Immunity
EN 50141	Conducted RF Immunity
EN 61000-4-4	Electrical Fast Transient Susceptibility
EN 61000-4-5	Surge Immunity
EN 50081-2	Gemeric Emissions Standard
EN 55011:1991	ISM equipment emissions (CISPR 11)

CE - Low Voltage Directive 72/23/EEC (In Process)

EN 61010	Safety of Electrical Equipment, Industrial Machines
IEC 529	Intrusion Protection Codes/NEMA 1/IP 20

CE - Machinery Directive 89/392/EEC (In Process)

EN 60204-1	Electrical Equipment for Machines
EN 292-1	Basic Terminology, Methodology
EN954-1	General Design Principals
prEN 50100-1	ESPD General Requirements

Printed Wire Board Assemblies (Mark V Boards)

UL 796 Printed Circuit Boards
UL recognized Card manufacturer, UL file number E110691
ANSI IPC guidelines
ANSI IPC/EIA guidelines

2-6. POWER SOURCES

Mark V panel can accept power from multiple power sources. Each power input source (example: the dc and two ac sources) should be fed through its own external 30 Ampere 2 pole thermal magnetic circuit breaker before entering the Mark V panel. It is a recommendation that the circuit breaker be near the equipment. The breaker ratings are 250 volts and 30 amperes with a minimum withstand rating of 10,000 amperes. The breaker should be supplied in accordance with EN61010-1 section 6.12.3.1 and marked as CE. Power sources can be any combination of a 125 V dc source and/or up to two 120/240 V ac sources. Each core within the panel has its own power supply card all of which operate from a common 125 V dc panel distribution bus. For further detail on the panel's power distribution system refer to Appendix E.

```
             ┌──────┐        125 VDC PANEL POWER DISTRIBUTION BUS
125 VDC      │  DC  ├─>|───┬──────────────────────────────────────>
SOURCE       └──────┘      │
                           │
             ┌──────┐      │
120/240 VAC  │ AC 1 ├─>|───┤ < HIGHEST DC VOLTAGE SELECTED
SOURCE #1    └──────┘      │
                           │
             ┌──────┐      │
120/240 VAC  │ AC 2 ├─>|───┘
SOURCE #2    └──────┘
```

Figure 2-3. Simplified Power Source Diagram

POWER SOURCE REQUIREMENTS TABLE				
VOLTAGE		FREQUENCY		CURRENT DRAW (measured at nominal voltage)
NOMINAL	TOLERANCE RANGE	NOMINAL	TOLERANCE RANGE	TYPICAL
125 V dc	100 to 144 V dc (Note 5)	N/A	N/A	7.0 A dc (Note 1)
120 V ac	108 to 132 V ac (Note 6)	60 Hz	47 to 63 HZ	7.0 A RMS (Notes 2 and 4)
240 V ac	216 to 264 V ac (Note 6)	50 Hz	47 to 63 HZ	3.5 A RMS (Notes 3 and 4)

Notes:

1.) Add 0.5 A dc continuous for each dc solenoid powered.
2.) Add 6.0 A RMS for a continuously powered ignition transformer (2 maximum).
3.) Add 3.5 A RMS for a continuously powered ignition transformer (2 maximum).
4.) Add 0.5 A RMS continuous for each ac solenoid powered (inrush 3.0 A).
5.) Ripple not to exceed 10 volts peak-to-peak.
6.) Total Harmonic Distortion not to exceed 7.0%.

2-7. EQUIPMENT GROUNDING

Within the Mark V panel, each controller is equipped with a single control common wire which connects to the panel control common ground bar, abbreviated as C-comm. The control common ground (C-comm) bar along with the shield bars are connected to the panel ground bar located at the bottom of the panel through the panel's steel structure. The panel ground bar must be attached to earth or grid ground with a minimum #4 AWG copper cable. This connection is a panel ground for safety and helps eliminate electrical noise.

WARNING

Although the panel itself is metal and grounded, it is an unacceptable practice to use it as a grounding point in place of an earth or grid ground.

Notes:

CHAPTER 3

CONFIGURATION FILES

3-1. INTRODUCTION

A Mark V Turbine Control System is customized for a particular application using "site specific" and "unit specific" configuration files. An automated program produces ASCII files containing the configuration information that are then placed on the hard disk of a primary operator interface computer, converted to hexadecimal format (if necessary), and downloaded to the control and communication processors of a Mark V control panel. Site specific configuration files include information regarding the printers available, the DCS communication links (see Chapters 10 and 11), the Stage Link communications (see Chapter 9), and the displays for the <I>'s. Unit specific configuration files include the Control Sequence Program segment files, Table files, and the I/O Configuration Constant files. To maintain the flexibility of the Mark V System, the files produced by the factory may be customized at any time, compiled, and downloaded to meet changing requirements. This chapter first introduces the terminology, then describes these configuration files.

3-1.1. Control Signals

Each Mark V Control Processor, <R>, <S>, and <T>, and Communication Processor, <C> and <D>, has its own Control Signal Database, or CDB. The Mark V uses the control signals in these databases for the control, protection and monitoring of the turbine via the Control Sequence Program, or CSP. The interfaces to the Mark V, including the <I>, <BOI>, and remote devices via MODBUS (see Chapter 10) or GSM (see Chapter 11), can display the name, value and engineering units of these same CDB signals.

Each CDB exists in RAM on the SDCC (Drive Control Card) in each Mark V Control and Communication processor. The Control Signal Databases on each of the control processors are identical, though separate, and contain identical values for the signals. There are other signal databases on the SDCC, as well as elsewhere in the Mark V on such cards as the TCQA and TCEA. Functions separate, though related to, the Control Sequence Program use these other signal databases. The control processors copy the required signal values from the other databases to the Control Signal Database. The signals not copied to the CDB are not accessible to the Control Sequence Program or the Mark V Interfaces. The SDCC microprocessors execute the Control Sequence Program using only the control signals and their values in the Control Signal Database on that SDCC card.

The signals in the Control Signal Database are of interest as the CSP performs most of the control and protection of the turbine unit. When reading the CSP, it is necessary to understand the nature of the control signals used in the sequencing. When modifying the CSP, it is often necessary to create new signals in the Control Signal Database. The "creation" of new signals is actually an "assigning" of names to spare signals at pre-existing memory locations in the SDCC RAM. The use of names greatly simplifies writing the CSP by eliminating the tedious and cumbersome task of using the signal's memory location.

There are two groups of control signals: Logics and Variables. These two groups can be divided into several signal types.

3-1.1.1. LOGIC - Any signal that can only have a value of zero (or "False") or one ("True"), is a logic signal. These signals can be thought of as one bit numbers, though they may actually be at least two bits. The <I> Logic Forcing Display can force most logic signals, and thus the signal needs a second, or "Force", bit. The Interfaces are otherwise not able to write to logics, unless they are Logic States or Push Buttons (see below). Aside from logic forcing, a logic signal gets it's value from the CSP, a Mark V input, or from an embedded algorithm (see below). The five types of Logics are:

- **Local Logic** - These logics are local to the respective processor, <C>, <D>, <R>, <S>, or <T>. <B> (collective term for <C> and/or <D>) Control Sequencing writes to the <B> local signals, and each <B> then communicates it's values to the <Q> (collective term for <R>, <S>, and/or <T>) processors. The <B>

Control Sequencing does not write to <Q> local signals, nor does the <Q> Control Sequencing write to the <B> local signals. <Q> Control Sequencing writes to <Q> local signals, and <Q> communicates their values to the <B> processors. If a <Q> processor is re-initialized, then the local signals necessary to begin processing are received from the other <Q> processors.

- **Private Logic** - These logics are unique to the respective <Q> processors: their values are not voted by the control processors. They can be used by the Control Sequence Program in the respective processor, and each processor may have different values for the signals. Their use is normally limited to functions where a non-voted value is required.
- **Fast Voted Logic** - These logics are unique to <Q> processors. The signal values are voted on each frame and are reserved for critical control and protection functions.
- **Logic States** - These logics are "command signals" that can be written to by any one of the operator interfaces. The operator interface sends a state value of one or zero to the Mark V. Once written to they are maintained in the commanded state until another command is received by the processors to change the state, or until the Control Sequence Program overwrites the commanded state. An example of this is where the permissives are not met to allow the execution of a command, thus the "command signal" is overwritten to reset it to an allowable state.
- **Push Buttons** - These logics are also command signals but they are momentary, not maintained like the Logic States. The operator interface sends a "frame sweeps" value, the number of frames (typically four) to hold it at the commanded value where a frame is one execution cycle of the Mark V processor. For example, an operator interface could have two different command targets on a screen for a `Raise Load` command; one for four sweeps, one for twelve sweeps. If each sweep increased load 0.5%, the four sweep command would increase load 2% while the twelve sweep command would raise load 6%.

3-1.1.2. VARIABLES - Signals that have values other than true or false are variables. These signals are typically sixteen (16) bit registers in the Control Signal Database. The most significant bit is the sign bit, making the number positive or negative. The sign bit and the fifteen remaining bits allow variable signals to have a range of positive or negative 32767. (For more information on how math functions are performed in the Mark V, see Appendix C.) The scale type of the variable signal determines it's range in engineering units. For example, a signal with a scale type of PCT has a range of +/-128.00 %. Analog Setpoints or Control Constants are the only variables written to from the operator interfaces. The CSP, the Mark V inputs, and the embedded functions write to the various variable signals. The five types of variables are:

- **Local Variable** - Similar in function to Local Logics, these variables are local to the respective processor. <B> Control Sequencing writes to the <B> local signals, and each <B> then communicates it's values to the <Q> processors. The <B> Control Sequencing does not write to <Q> local signals, nor does the <Q> Control Sequencing write to the <B> local signals. <Q> Control Sequencing writes to <Q> local signals, and <Q> communicates their values to the <B> processors. If a <Q> processor is re-initialized, the local signals necessary to begin processing are received from voted value of the other <Q> processors.
- **Private Variable** - Similar in function to Private Logics, these variables are unique to the respective <Q> processors: their values are not voted by the control processors. They can be used by the Control Sequence Program in the respective processor, and each processor may have different values for the signals. Their use is normally limited to functions where a non-voted value is required.
- **Fast Voted Variable** - These variables are unique to <Q> processors. The signal values are voted on each frame and are reserved for critical control and protection functions.
- **Analog Setpoints** - These variables are maintained command signals written to by the interfaces. Similar in function to Logic States, the CSP may overwrite the analog setpoint signal's value.
- **Control Constants** - Operating characteristics such as control curves, design criteria, alarm and trip levels are determined by the values of the Control Constants used in the Control Sequence Program. Control constants are loaded from EEPROM to RAM when the Mark V processor is initialized. There are two methods to change the values in RAM. The first is through the Control Constant Adjust Display, which changes the RAM values while the unit is on-line. A function on the Control Constant Adjust Display updates the EEPROM values with all the RAM values when executed. The second method is through editing the table file `CONST_Q.SRC` on the <I>, table compiling, downloading to the Mark V processors, and then re-initializing the processors to get the new values from EEPROM into RAM. (For more details, see the following sections of this Chapter and Chapter 4 of the Maintenance Manual, GEH-5980.)

CAUTION

Control Constant values are specified by the turbine manufacturing facility and should not be changed without the express written consent of the turbine manufacturer.

Combining the two types of Mark V processors, <B> and <Q>, with the different types of signals results in the following table of abbreviated names for the various control signals:

	<B>	<Q>
Local Logic	LB	LQ
Private Logic		LQP
Fast-Voted Logic		LQV
Logic State		LQLS
Push Button		LQPB
Local Variable	VB	VQ
Private Variable		VQP
Fast-Voted Variable		VQV
Analog Setpoint	VBAS	VQAS
Control Constant		VQCC

Table 3-1. Control Signal Abbreviations

<B> processors do not have any Private Logics, Fast-Voted Logics, Logic States, Push Buttons, Private Variables, Fast-Voted Variables, or Control Constants because they typically have limited sequencing (or none at all) and do not require these types of control signals. If a <B> processor requires any of these, such as a Control Constant, it reads <Q>'s control signals.

The SDCC card's RAM and/or CDB stores the control signal values obtained from one of five sources, depending on the type of signal:

- **CSP** - the Control Sequence Program calculates values and stores the results in control signals. Control signals written by the CSP include local logics, private logics, fast-voted logics, logic states, local variables, private variables, fast-voted variables, and analog setpoints.
- **Mark V Input** - The signal is **structured** or assigned to a Mark V input. The input terminations are read by I/O cards, which in turn write values to the related control signal. As examples, the TCQA reads LVDT inputs, the TCDA reads digital inputs, and the TCEA reads emergency overspeed inputs. Mark V Inputs write to local logics and local variables.
- **Mark V Interfaces** - The <I> and the <BOI> issue commands to the Mark V by writing to command signals in the Control Signal Database. The CSP then reads these signals and uses them in its calculations accordingly. These command signals include the logic states, push buttons, and analog setpoints. The <I> can change the value of most of the non-command logic signals, as well as the command logic state and push button signals, with the Logic Forcing Display. The Control Constant Adjust Display on the <I> changes the RAM values of control constants.

- **Embedded Mark V I/O Card Algorithms** - The Mark V controls and protects the turbine through the CSP. There are other algorithms embedded into the Mark V controller that use control signals for their functions. Examples of these functions include detecting NOx injection out of limits, and an emergency overspeed condition. These embedded algorithms are executed by many of the I/O cards, including the SDCC, and they write to local logics and local variables in the Control Signal Database.
- **EEPROM on the SDCC's** - Data downloaded from an <I> to the Mark V remains intact in the EEPROM when the Mark V is powered down. The Mark V copies this data into RAM when the Mark V processors are initialized or powered up. A portion of this EEPROM data is the Control Constant values, and thus the RAM Control Constants get their values when the processors are initialized. This way they use the same data each time the processors are re-initialized.

CBD types from Table 3-1	Can the CDB types below be written to by:				
	CSP	Mark V Inputs	Operator Interfaces	Embedded I/O Algorithms	EEPROM
LB	Yes	Yes	Yes	Yes	
VB	Yes	Yes		Yes	
VBAS	Yes		Yes		
LQ	Yes	Yes	Yes	Yes	
LQP	Yes		Yes	Yes	
LQV	Yes	Yes		Yes	
LQLS	Yes		Yes		
LQPB			Yes		
VQ	Yes	Yes		Yes	
VQP	Yes			Yes	
VQV	Yes	Yes		Yes	
VQAS	Yes		Yes		
VQCC			Yes		Yes

Table 3-2. Summary of sources for signals for various CDB Types

Some logics and variables are "structured", meaning that their memory location within the Control Signal Database is tied directly to some specific function. Naming and using a specific structured signal relates to a specific function. This was seen above with the local logics and local variables that received their values from Mark V inputs. Structured signals include:

- **I/O** - These structured signals read their values from Mark V inputs, such digital inputs, speed inputs, temperature inputs, etc. They also write their values to Mark V outputs, such as digital outputs, milliAmp outputs, servo valve position references, etc.
- **Voted Double Words** - These structured signals are voted as 32-bit double words. They may be used in Control Sequence Program functions where high accuracy and agreement between the processors is required.
- **Alarms** - These structured signals are tied to specific alarm drops. The state of a signal here gets reported to the interfaces as a specific alarm drop.

Table 3-3 summarizes the differences between structured and unstructured software control signal assignments (sometimes called "software points").

Structured Software Control Signals	Unstructured Software Control Signals
Relates to a specific function and must be in factory assigned location of the Control Signal Database.	Can be placed in any free location of the Control Signal Database.
Assignments made in: • `IO.ASG` for Mark V I/O related signals. • `ALLOCSSP.ASG` for fast-voted double word and alarm drop structured software signals.	Assignments made in: • `FACTORY.ASG` for those made by the factory. • `SITE.ASG` for those made at the customer site.
Must choose the specific control signal when assigning a name, as well as specifying the scale type of the signal.	The assignments in these two files require the signal type, (from Table 3-1), the signal name, and the scale type. The <I> Data Dictionary tools decide the "memory location - to - control signal name assignments" in the CDB, relieving the user of this task.

Table 3-3. Structured and Unstructured Signals

3-2. <I> UNIT-SPECIFIC DIRECTORY

Each Mark V control panel assigned to communicate with an <I> has a unit-specific directory and subdirectory on the hard disk of the <I>. These directories have names which refer to the unit. They are located on the F: drive. The unit-specific directory for the first unit the <I> communicates with is named `F:\UNIT1`; its subdirectory is `F:\UNIT1\PROM`. Subsequent unit-specific directories and their subdirectories would be `F:\UNIT2` and `F:\UNIT2\PROM`, `F:\UNIT3` and `F:\UNIT3\PROM`, etc...

Configuration files contained in a unit-specific directory can be broken up into the following groups:

- Assignment files
- Data Dictionary files
- I/O Configuration Constant files
- Table Files
- CSP segment files.

These five groups of files are detailed below.

The `F:\CONFIG.DAT` file, partially shown in Figure 3-1, is a text file that contains information about the units with which the <I> can communicate. The lines in this file that begin with a semi-colon (;) are comments which do not affect operation and are ignored by <I> programs that use the file. The section of the file shown in the example defines the unit information for the <I>.

Information in the `UNIT_DATA` section denotes the unit(s) with which an <I> can communicate. Each line in the section represents a particular unit; that is, the unit number, the unit name, and the path to the unit's configuration information (its unit-specific directory). This information is necessary to determine where the unit-specific files for a particular unit reside. The cautions specified in the comment should be observed when making modifications to this section, and the <I> must be restarted in order for the changes to take effect.

```
;  This section defines the unit numbers, unit names, and the path to the
;  directory that contains the unit information. Each line contains the
;  unit number (decimal), the unit name (2 char max), and the path to the
;  unit configuration data (64 char max). The unit numbers should be in
;  order, and if a unit number is repeated, the last entry wins.
;
;      UNIT    UNIT     PATH TO
;     NUMBER   NAME     CONFIG DATA
;     ------   ----     -------------------------
UNIT_DATA
        1       T1      F:\UNIT1
        2       T2      F:\UNIT2
        3       T3      F:\UNIT3
        4       T4      F:\UNIT4
```

Figure 3-1. Portion of F:\CONFIG.DAT

3-3. UNIT-SPECIFIC ASSIGNMENT FILES

Assignment files, while not downloaded to a Mark V control panel's processors, contain unit-specific control signal database pointnames and scale types for many of the control signals. The information in assignment files is used when creating the primary unit Data Dictionary file, `UNITDATA.DAT`. This file contains all of the unit-specific control signal database pointname information.

For each unit, GEDS provides the following four assignment files in the <I>'s unit-specific directory: `IO.ASG`, `FACTORY.ASG`, `ALLOCSSP.ASG`, and `SITE.ASG`. These are all American Standard Code for Information Interchange (ASCII) text files (sometimes called DOS text files). They can be viewed or modified using any ASCII text editor.

When I/O devices are connected to a Mark V control panel, they must be assigned a control signal database pointname and a scale type. I/O devices connected to a Mark V control panel are specified in the I/O assignment file, `IO.ASG`. In this file, a control signal database pointname and a scale type are assigned to the location which is being used for a particular device.

A Mark V control panel's processors have multiple spare control signal database memory locations (points) which are available for use (or assignment). To make use of these spare points for new or additional control and protection functions it is necessary to define the type of point required, the control signal database pointname for the point, and the scale code for the point. These definitions are made in one of the following assignment files: `FACTORY.ASG`, `ALLOCSSP.ASG`, or `SITE.ASG`. The file in which the assignment is made depends on the type of signal required as well as on who is making the assignment (that is, factory personnel or site personnel, customer or GEPG/Business Associate field personnel).

GEDS or GEPG's business associates assign control signal database pointnames and scale types to spare memory locations in FACTORY.ASG. This file may be altered to accommodate customization of the Control Sequence Program for a particular application. The only types of assignments not made in FACTORY.ASG are for additional I/O, spare double-word variables, and spare alarm logic points.

Spare double-word variables and alarm logic points which are required for a particular application are assigned pointnames and scale types in ALLOCSSP.ASG (which stands for ALLOCation of Structured Software Points). Both factory (GEDS and GEPG Business Associates) and field/site personnel can make assignments for these two types of points in ALLOCSSP.ASG.

Customer and/or GEPG/Business Associate field personnel are to make assignments of signal pointnames and scale types to spare control signal database memory locations in SITE.ASG for points other than I/O, double-word variables, and alarm logic points.

3-4. UNIT-SPECIFIC DATA DICTIONARY FILES

Data Dictionary files contain information about unit-specific control signal database pointnames, alarm text messages (for both process and diagnostic alarms), and display information for signal pointnames (type/units, messages, etc.).

The primary unit Data Dictionary file, `UNITDATA.DAT`, can be created on an <I> in the unit-specific directory. Assignment files and template files (see below and section 3-1.1.) are used in the creation of `UNITDATA.DAT`. Many configuration programs on an <I> require information from `UNITDATA.DAT` when modifying or compiling unit configuration files for downloading.

Some control signal database pointnames are common to applications (steam turbines or gas turbines) and must reside in memory at specific locations and must not be changed. These common, fixed pointnames are contained in template files. The fixed control signal database pointnames, the I/O assignments, and spare memory locations being specified in the assignment files must be included in the `UNITDATA.DAT` file. If any new assignments are made, they must be included in a new `UNITDATA.DAT` file.

`UNITDATA.DAT` is created by the program `DDLOCATE`. This program uses the assignment files which are specified at the time `DDLOCATE` is executed in addition to three template files in the unit-specific PROM sub-directory: `UNITDATA.TPL`, `UNITFREE.TPL`, and `UNITMAP.TPL`. Information from both the assignment files and the `.TPL` files (TPL stands for "template") in the PROM sub-directory are used to create the unit-specific `UNITDATA.DAT` file. The command-line format for executing `DDLOCATE` is:

```
DDLOCATE IO.ASG FACTORY.ASG ALLOCSSP.ASG SITE.ASG
```

Although their order is unimportant, all assignment files for a particular unit **must** be specified on the command line each time `DDLOCATE` is executed. If a modification is made to `ALLOCSSP.ASG` only (such as to use a spare alarm logic point), all the assignment files must be specified on the command line when `DDLOCATE` is executed. Each time `DDLOCATE` is executed, a new `UNITDATA.DAT` file is created; all the assignments must be included in this new file. For more details see the `MK5MAKE.BAT` file description in Chapter 4 of the Maintenance Manual, GEH-5980.

Other Data Dictionary files that must be present in the unit-specific directory for proper operation include:

- `ALARM.DAT` Process and Diagnostic Alarm messages (Max-case)
- `ENUMDATA.DAT` Display messages for Enumerated Data types
- `SCLEDATA.DAT` Scale code information (English, Metric, etc.)

The following unit-specific Data Dictionary files are optional and not required for proper operation of an <I>:

- `LONGNAME.DAT` CDB Pointname Descriptions
- `SYNONYM.DAT` Site-specific CDB Pointname Synonyms

Unit-specific Data Dictionary files are not downloaded to a Mark V control panel's processors, but are loaded into the <I>'s RAM each time the <I> is turned on or reset. This information is used to scale and display control signal database pointname information on the <I> as well as for alarm and event logging. As discussed above, some programs on the <I> require information from UNITDATA.DAT.

3-5. UNIT-SPECIFIC I/O CONFIGURATION CONSTANTS

I/O Configuration Constants are used to scale or condition signals to and from I/O devices connected to the Mark V control panel. I/O devices include pressure transducers, temperature switches, electro-hydraulic servo-valves, position transducers or reactors, thermocouples, RTDs, etc.. Many of these I/O devices, being of the same type, can have differing outputs or require dissimilar inputs. For example, thermocouples produce a millivoltage proportional to temperature, however, a Type K thermocouple produces a different millivoltage than a Type T for the same temperature. An I/O Configuration Constant can be used to appropriately scale the input signals from various types of thermocouples. Milliamp transducers come in several output ranges: 4-20 mA, 0-1 mA, 0-10 mA, etc.. More than one type of milliamp transducer may be used on a unit or its auxiliaries. I/O Configuration Constants are used to scale the input for use in controlling, protecting, or monitoring the unit.

I/O Configuration Constants are initially contained in the I/O configuration files in the unit-specific directory. The files are: `IOCFG_Q.DAT, IOCFG_C.DAT`, and `IOCFG_D.DAT`. All three files will be present in the unit-specific directory for each Mark V control panel that is to communicate with the <I>, even if the control panel does not include a <D> backup communication processor. The information in these files is in hexadecimal format, and can be viewed using the I/O Configurator program, `IO_CFG`, usually available from the <I> Main Menu. The screens presented in the I/O Configurator depend on the configuration data files found in the PROM directory for the unit (see Figure 3-2). `PROM\IO_CFG.DAT` contains the list of files required for the I/O Configurator,such as TCCA_CFG.DAT. Figure 3-2 represents these files with `?????CFG.DAT`. The I/O configuration files may be downloaded to a Mark V control panel's processor(s) without any intermediate steps (such as compiling).

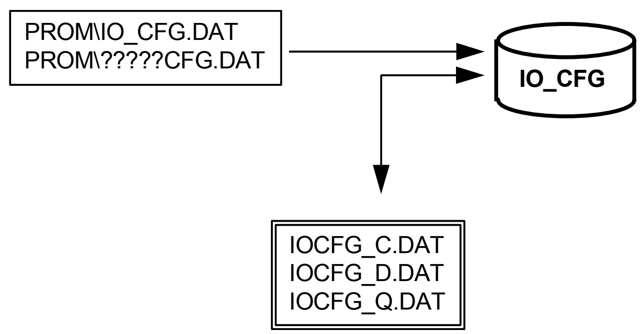


Figure 3-2. Creation and Modification of I/O Configuration Files

3-6. UNIT-SPECIFIC TABLE FILES

The majority of unit-specific configuration files are Table Files. These files contain tabular listings of control signal database pointnames and information about their type, use, value, etc. Table Files contain information in an ASCII text format which, when compiled and downloaded, is used by functions such as the Control Sequence Program and the loggers of the Mark V control panel's processors. Figure 3-3 shows a list of Table Files and a brief description of their contents. Several of the source Table Files are dummy files and contain no information. They have been created for symmetry and possible future use.

Modifications can be made to any of the ASCII text Table Files (known as source files) using any ASCII text editor. Prior to downloading the information in the source Table Files, it must be converted into hexadecimal format using the Table Compiler program, `TABLE_C`. The command line format for executing the Table Compiler to compile all the Table Files is:
`TABLE_C ALL`

Using the Table Compiler, information in the source Table Files will be converted into hex format in files with the same filename but with a `.DAT` filename extension. (For example, `CONST_Q.SRC` would be compiled into `CONST_Q.DAT`.)

The Table Compiler uses information contained in UNITDATA.DAT and one of the scale code files (ENGLISH.SCA, by default) when converting the source files into hex files. Since no control signal database pointnames are downloaded to the Mark V control panel processors, the Table Compiler finds the software signal pointname in UNITDATA.DAT, and uses its memory location/address and scale code and point type when creating the downloadable Table Files from the information in the source Table Files.

```
CONST_B.SRC Dummy file; blank
CONST_Q.SRC All Control Constants and their (initial) values
EPA_B.SRC   A list of pointnames for emissions logging purposes
EPA_Q.SRC   Dummy file; blank
MAOUT_B.SRC Dummy file; blank
MAOUT_Q.SRC A list of pointnames and ranges for <C> mA outputs
CHNG_B.SRC  A list of <C> analog pointnames and ranges monitored for
            excursions and logged as events to the Historian
CHNG_Q.SRC  A list of <Q> analog pointnames and ranges monitored for
            excursions and logged as events to the Historian
EVENT_B.SRC A list of <C> logic signal pointnames logged as events
EVENT_Q.SRC A list of <Q> logic signal pointnames logged as events
TOTT_B.SRC  A list of pointnames to configure the Hold List
TOTT_Q.SRC  A list of pointnames for which data is totalized
HIST_B.SRC  A list of pointnames included in the Trip History log
HIST_Q.SRC  Dummy file; blank
CBLR_B.SRC  A list of digital inputs to <C> which are associated with
            command pushbuttons in the CSP
CBLR_Q.SRC  A list of digital inputs to <Q> which are associated with
            command pushbuttons in the CSP
```

Figure 3-3. List of Table Files

3-7. UNIT-SPECIFIC CSP SEGMENT FILES

A CSP segment is an ASCII text file which contains information such as Control Blocks, parameters, comments, and/or relay ladder diagram sequencing. The Control Sequence Program for a unit is made up of at least two segments — one for <Q> and one for <B>. CSP segments can be executed at different frequencies (such as 4, 8, 16, or 32 Hz, depending on the application) and at different skews, or offsets. Segments are subsets of the CSP containing sequencing functions which are related and/or must be executed at a certain frequency. There can be as many as eight CSP segments for <B> and eight CSP segments for <Q>.

CSP segments can be viewed and modified using the Control Sequence Editor program, MSE (usually available from the Main Menu of an <I>). Refer to Chapter 5 for the Control Sequence Editor. In some cases, all of the unit's control and protection (other than emergency overspeed trip and servo regulator loops) can be accomplished in one CSP segment in <Q>. However, memory characteristics of an <I> may require that the CSP be broken up into multiple segments in order to be displayed by the Control Sequence Editor. CSP segment files can have any valid DOS filename (eight characters max) but must have an .SRC filename extension.

Prior to downloading to a Mark V control panel's <B> and <Q> processors, the CSP must be converted to hexadecimal format using the Control Sequence Compiler, or CSP Compiler. The CSP Compiler uses information from UNITDATA.DAT, BBL definition files in the unit-specific PROM subdirectory (PRIMITIV.DEF and BIGBLOCK.DEF), and the names of CSP segment files which have been specified in a unit-specific control sequencing configuration file, MSTR_SEQ.CFG. The CSP Compiler creates two hex format downloadable CSP files — SEQ_B.DAT and SEQ_Q.DAT. The CSP Compiler can be executed from the Main Menu of an <I> or at the command line of the unit-specific directory with the COMP_SEQ command.

MSTR_SEQ.CFG (a text file, refer to Figure 3-4) contains two sections which define the names of CSP segment files which are compiled for <Q>'s CSP and <B>'s CSP. In addition, it defines the rates and the offsets/skews as well as the order in

which CSP segments are compiled and executed. (The first segment file specified will be executed first, the second segment file specified will be executed next, the third segment file specified will be executed next, and so on.)
CSP segments are initially created using BBLs (see Appendix C), relay ladder diagram rungs, and comment rungs. They are customized by GEDS or its Business Associates to match a particular application or Customer's requirements and can be modified in the field using the Control Sequence Editor. New CSP segments can be created using the Control Sequence Editor. If a new segment is created, the name of a new segment must be added to `MSTR_SEQ.CFG` to be included in the downloaded CSP files. The maximum of segments per <Q> and <B> that can be compiled is eight.

```
---------------------------------------------------------------------
--  F:\UNITn\MSTR_SEQ.CFG Configuration file for sequencing compiler
---------------------------------------------------------------------
#LIST
                  Major     Minor
                  rev       rev
#BBL_REVISION     6         2

---------------------------------------------------------------------
--  <R>, <S>, <T> Segments
---------------------------------------------------------------------
#<Q>_SEGMENTS
                            Frequency       Skew within
                            power of 2 (>0) frequency
    #SEGMENT    SEQU_Q1     1               0
    #SEGMENT    SEQU_Q2     2               0
    #SEGMENT    SEQU_Q3     2               1

---------------------------------------------------------------------
--  <C> and <D> Segments
---------------------------------------------------------------------
#<B>_SEGMENTS
                            Frequency       Skew within
                            power of 2 (>0) frequency
    #SEGMENT    SEQU_B      1               0

#END
```

Figure 3-4. Unit-specific Master CSP Configuration File, MSTR_SEQ.CFG

3-8. COMPILING UNIT-SPECIFIC CONFIGURATION FILES

The unit-specific Table and CSP files must be converted to hexadecimal format prior to downloading to the processors in a Mark V control panel. Figure 3-5 represents the creation of the primary unit-specific Data Dictionary file (`UNITDATA.DAT`) and its use by the various <I> programs in creating the unit-specific downloadable CSP and Table Files.

3-9. DOWNLOADING UNIT-SPECIFIC CONFIGURATION FILES

When the unit-specific ASCII text Table and CSP files are compiled to hexadecimal format, they along with the I/O Configuration Constants can be downloaded to the processors in the Mark V control panel using the EEPROM Downloader program. The EEPROM Downloader program, `EEPROM`, is usually available from the Main Menu of an <I>. It can also be executed from the DOS command line.

The EEPROM Downloader program will transfer unit configuration file information (sometimes known as EEPROM partitions or sections) from the <I> computer's hard disk to a EEPROM chip on the SDCC card of a processor. It will transfer only the information requested to one processor at a time. The information can be one, several, or all, of the downloadable unit-specific configuration files.

NOTE

There is a hardware jumper that must be in the correct position to allow the modification of EEPROM values. See Appendix A.

If the EEPROM Downloader program was executed from the Main Menu of an <I>, all the unit-specific information for the <C> processor of unit T1 can be downloaded to the communication processor's EEPROM with the following command:

```
DOWN T1 C USER
```

For more information about the EEPROM Downloader, its uses, and its command-line syntax refer to GEH-5980, Maintenance Manual, Chapter 4. A depiction of the downloading of unit-specific hex format configuration files is shown in Figure 3-6. The downloading to <Q> occurs through <C> (or <D>, if so equipped and <C> is powered-down) via the DENET. Do not reset <C> (or <D>) while downloading to <R>, <S>, or <T> until the download is complete.

3-10. CONTROL PANEL CONFIGURATION FILES

Once downloaded to the processor EEPROM, the configuration files must be transferred, or loaded, into the processor's RAM for use in control, protection, monitoring, and logging functions. To transfer the configuration information from processor EEPROM to RAM, the processor must be re-booted to start the initialization process. Resetting can be accomplished in one of two ways, either by pressing the RESET button on the SDCC card of the processor or by shutting down the processor and reapplying power. The first method is called a soft reset, the second, a hard reset.

Resetting the SDCC processor causes it to lose the current contents of its RAM on the SDCC, where the CSP, Control Constants, and other related configuration information are stored for execution by the processor's 80186 and 80196 micro- and co-processors. Resetting a processor transfers some I/O Configuration Constants from EEPROM to RAM on the individual I/O cards associated with the processor. For example, the I/O Configuration Constants for the TCDA cards associated with the <R> control processor are transferred from <R>'s SDCC's EEPROM to RAM on the TCDA cards each time <R> is reset or turned on.

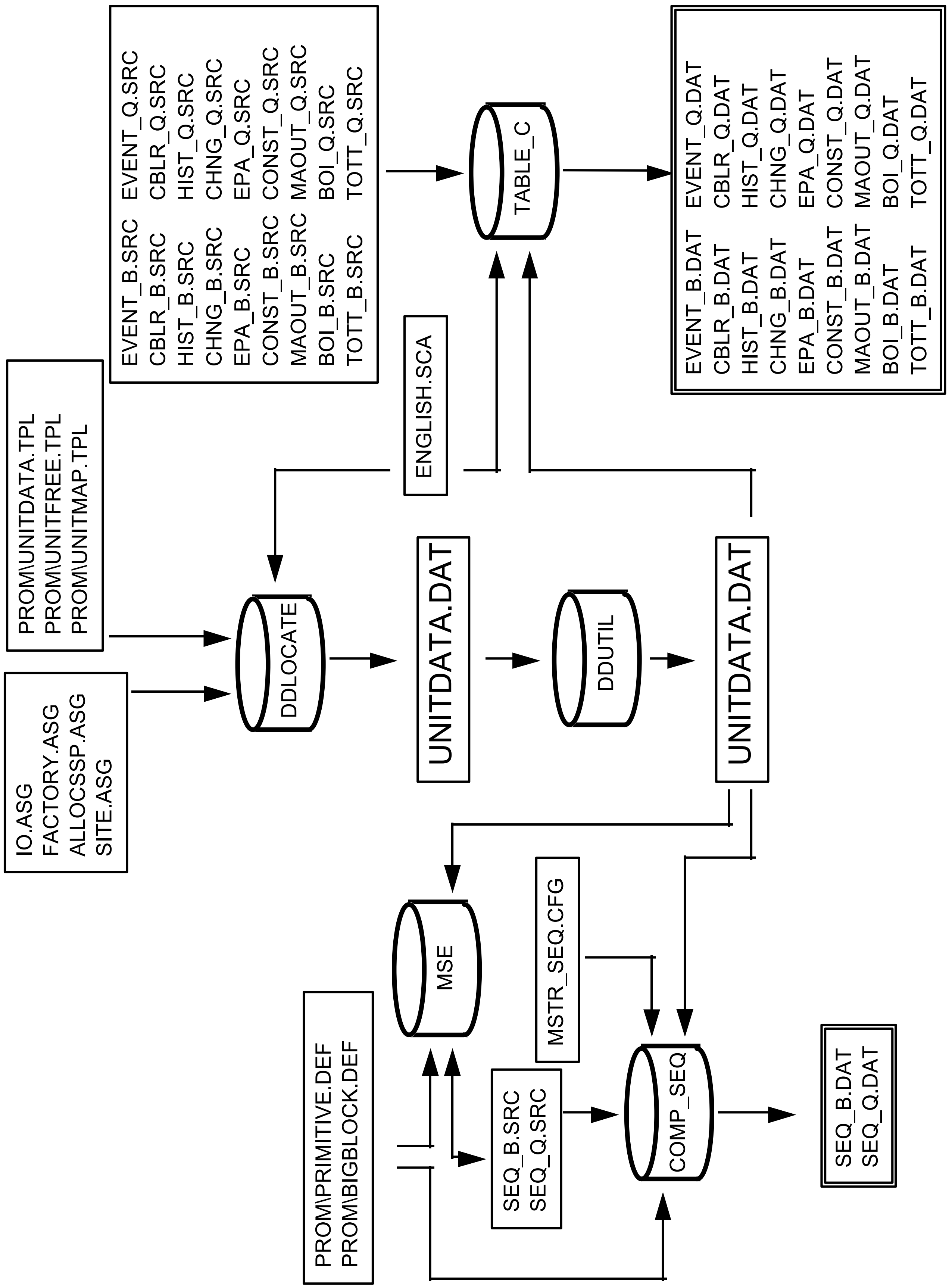


Figure 3-5. Creation/Relationship of UNITDATA.DAT to CSP and Table Files

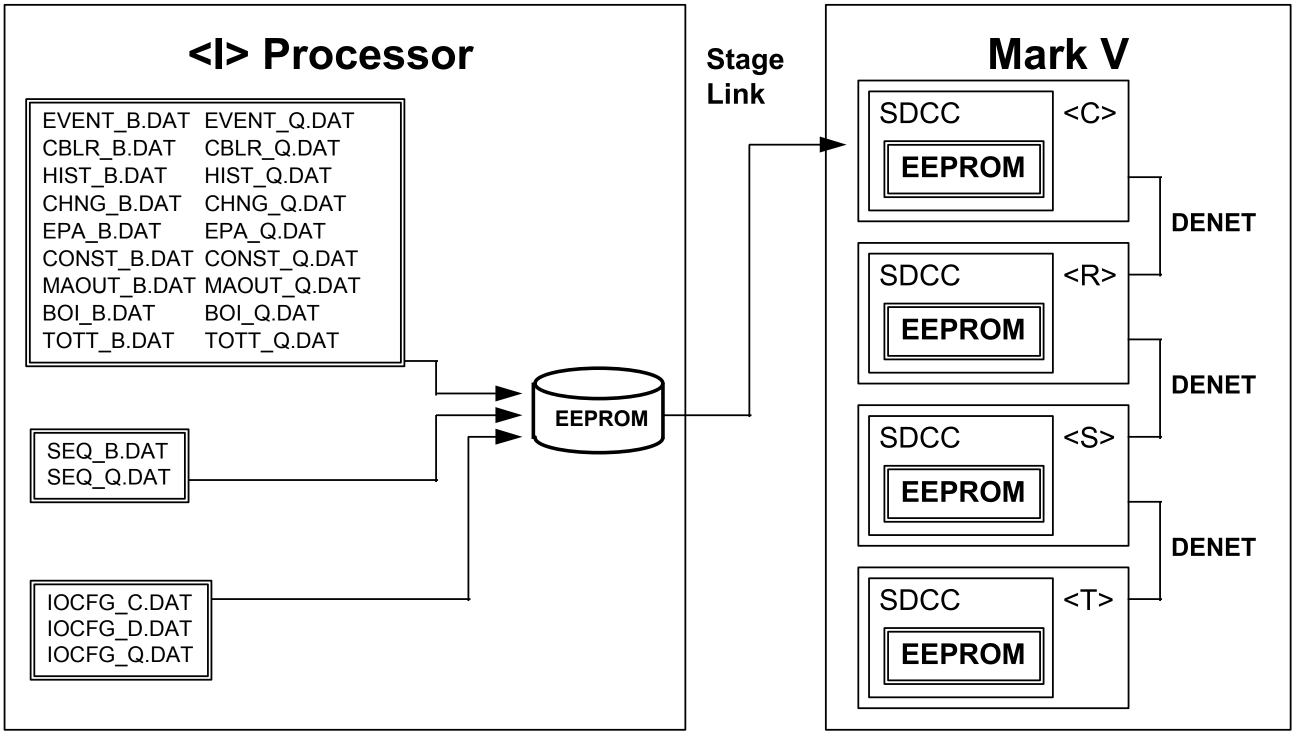


Figure 3-6. Downloading Unit-specific Configuration Files

NOTE

I/O Configuration Constants for the protective processor cards <X>, <Y>, and <Z> are transferred from the control processor EEPROM on the SDCC to RAM on the TCEA cards only when the protective processors are reset by removing their power and then reapplying it as described above.

The above explanation demonstrates that the processors in the Mark V control panel use configuration information stored in their RAM for control and protection of the unit. In order to change the RAM information, the configuration files on an <I> must be modified and then compiled (if necessary). These files must then be downloaded to the appropriate processor(s) and the processor(s) reset in order for the change to become effective. The one exception to this is the modification of Control Constants.

3-11. CONTROL CONSTANTS

Control Constants which are used by a Mark V control panel processor (<Q> or <C>) during the execution of the CSP are stored in RAM and were transferred there from EEPROM when the processor was last reset (or turned on). The values of Control Constants in the processor's RAM can be changed using the Control Constant Adjust Display program, `CCONST`, usually available from the Main Menu of an <I>. Control Constants can be adjusted while the unit is running, although the rate of change of the Control Constant's value is quite slow when the unit is running to prevent a rapid change from tripping the turbine. For more information on the Control Constant Adjust Display, refer to GEH-5980, Maintenance Manual.

A feature of the Control Constants Adjust Display is the ability to modify the processor's EEPROM value for a Control Constant whose value in RAM was changed. By clicking on the target EEPROM UPDATE and then clicking on the

EXECUTE COMMAND display target, the current RAM value of **every** Control Constant will be copied to the processor's EEPROM.

However, there is no automatic method of updating the values of Control Constants in the configuration file `CONST_Q.SRC`. If a Control Constant in a Mark V control panel is modified using the Control Constant Adjust Display and the value of the Control Constant in `CONST_Q.SRC` is not subsequently changed to match the unit's value, a re-compiling and downloading of Control Constants will cause the EEPROM value of the Control Constant to revert to the old value in `CONST_Q.SRC`.

NOTE

Whenever a Control Constant is modified using the Control Constant Adjust Display, the Control Constant source Table File, `CONST_Q.SRC`, should be edited to reflect the new value and compiled. This will assure the Control Constant Table File hexadecimal file, `CONST_Q.DAT`, will contain the new value and any subsequent downloads will be done with the correct value.

The other option is to use the `EEPROM Downloader UP` option to upload or bring the values from the SDCC's RAM to a data file (`.DAT`) and reverse table compile using the Reverse Table Compiler program, `REVX_TAB`. The command line format for executing the Reverse Table Compiler is `REVX_TAB CONST`.

NOTE

Reverse Table Compiling `CONST_Q.DAT` will overwrite the current `CONST_Q.SRC` file. All comments except those in the header will be lost, and the Control Constants will be sorted by their memory location. Aside from these differences, the new `CONST_Q.SRC` will be a valid file.

Notes:

CHAPTER 4

DISPLAY APPLICATIONS

4-1. ANIMATED DISPLAYS INTRODUCTION

An animated display on an <I> is a display containing graphic representations. It is created by running the animated screen builder (ANIM.EXE) with an item list file. Animated displays get their name from the fact that items of the display can be made to change color based on logic signals. Certain graphic items, such as bargraphs, are dynamic and can change with each update of the display's information. The item list file, which is read by the animated screen builder, is an ASCII text file containing commands and information about items to be shown on an <I> display. Graphic items, including vertical bargraphs, command targets, text information, and so on, can be put in an animated display.

4-2. ANIMATED DISPLAY BASICS

Displays are called by name. When searching for the display definition file, the ANIM program looks for the file:

```
F:\RUNTIME\display.A0
```

then

```
F:\RUNTIME\display.Au
```

and then

```
F:\UNITn\display.A
```

where u in `.Au` represents the unit number and `UNITn` represents the unit configuration directory defined for that unit. This allows for the cloning of units by copying all files in the unit configuration directory. Customers should make any changes in the `F:RUNTIME` directory and leave the version in `F:\UNITn` as the "as shipped" version.

4-2.1. The Animated Display Area

The animated screen builder is used to create displays in the area of an <I> display (above the Alarm Window and Display Targets at the bottom of the screen). The display area can be thought of as an x-y grid with the 0,0 coordinates at the lower left-hand corner of the display. This area includes the section at the top of the screen where the date/time and unit number fields are displayed. The grid is 1000 x 1000 units, 0-999 units in the x direction and 0-999 units in the y direction. The grid is independent of video resolution. Positive x and y directions are from left to right and from bottom to top starting in the lower left-hand side of the screen (above the alarm window), respectively.

4-2.1.1. Display Switching. The following targets are used to move between the different displays.

Page Down/Page Up are used to move to the next or previous page of a multi-page display. These targets are visible only if the current display definition includes a PGDN or PGUP item.

Prev Page returns to the previous display. Repeated strikes of this target toggles you between two displays (it is not a queue with a depth of previous screens).

Hot Spots allow the user to switch to different animated displays and return to the current display without exiting to the Main Menu (see section 4-3 of this manual). There can be up to 128 hot spots defined on a display.

4-2.1.2. Cursor Locators. The following functions show the position of the cursor on the screen, making it easier to place new items at the proper location.

`ALT D` – **Display Cursor Location** toggles the option to show the current cursor position on the line above the time. The position is dynamic and updates continuously. Selecting `ALT D` again turns off the display.

`ALT F` – **Find Cursor** shows the location of the cursor on the line above the time. This position is static and changes only when `ALT F` is selected.

It is beneficial to use `ALT F` to show the location of one corner of a region and use `ALT D` to see the other corner of the region. This gives the two corner points for the region, which is useful for defining item commands (see section 4-8 of this manual). The `ALT D` location, which is updated continuously, is redrawn after the screen is erased and redrawn. The `ALT F` location updates only when the `ALT F` key is selected (redrawing the screen erases the value).

4-2.2. Defining Items In Item List Files

Items that appear in an animated display are defined by commands in an item list file (`anim list_file`). The commands follow a basic pattern:

```
Item?          Starting where?       What color(s)?
item_command   start_x  start_y      fore_color    back_color
```

The entries of an item command must be separated from each other by at least one space (the use of spaces or tabs between entries is encouraged to make the item list files easier to read). The underscore characters in the example item command above are **place holders** which are used to indicate that spaces in entries are not allowed except in text strings, in which case the animated screen builder will display spaces as entered. Item commands can, and usually do, contain more entries than those shown in the example above.

Each individual item, its required format, and its usage is explained in section 4-2.3, Item Commands. Example items are used to explain some of the basics of defining items. When defining items and their colors in an item list file, the case is unimportant. However, case sensitivity is important when specifying text in quoted strings.

4-2.3. Item Command Starting Coordinates

Each item in an animated display must be defined in terms of its starting coordinates. Coordinates for the starting point of an item can be expressed in either **absolute** or **relative** units. For example, when used in an item list file, the two item commands below will cause the animated screen builder to horizontally display two pointnames `(TNH,TNR)` that is, their respective values and engineering units in a double high font in the default foreground color.

```
hpt2    150   200  df   tnh
hpt2   +000  -040  df   tnr
```

The coordinates of the first item are expressed in absolute units that is, with respect to the lower left-hand corner of the screen area above the Alarm Window. The coordinates of the second item are expressed in relative units with respect to the **starting-point** coordinates of the **previous** item. The first item command will display the value and engineering units for TNH (beginning 150 units from the left of the display and 200 units up from the Alarm Window). The second item command will display the same information for TNR, also beginning 150 units from the left of the display but only 160 units up from the Alarm Window (40 units below the starting y-coordinate of the **previous** item).

Adding a third item line similar to the one below will display a double-high text string in the default foreground color beginning 350 units to the right of the previous line and at the same y-coordinate as the previous line:

```
tx2 +350   +0    df   "Turbine Speed Reference"
```

The three item lines when used in an item list file would appear on an animated display like the portion shown in Figure 4-1.

```
TNH    99.98  %
TNR   102.37  %   Turbine Speed Reference
```

Figure. 4-1. Portion of Animated Display Showing Absolute and Relative Coordinates

4-2.4. Defining Color

Sixteen colors can be used in an animated display. They are as follows:

```
bk - Black                rd - Red
gr - Green                mg - Magenta
cy - Cyan                 bl - Blue
yl - Yellow               wh - White
dg - Dark Grey            lg - Light Gray
lb - Light Blue           lc - Light Cyan
lr - Light Red            br - Brown
db - Default Background
df - Default Foreground
```

Most items in an animated display have both static and dynamic portions, or fields. For example, an item defined in an item list file using the `hptn` command has two static portions (the CDB signal pointname and the point's engineering units) and a dynamic portion (the point's value); see Figure 4-2. The static field of an item is the portion which will not change based on the value of the item's CDB point. The dynamic field of an item is the portion which will be updated based on the value of its CDB point with each update of the animated display screen.

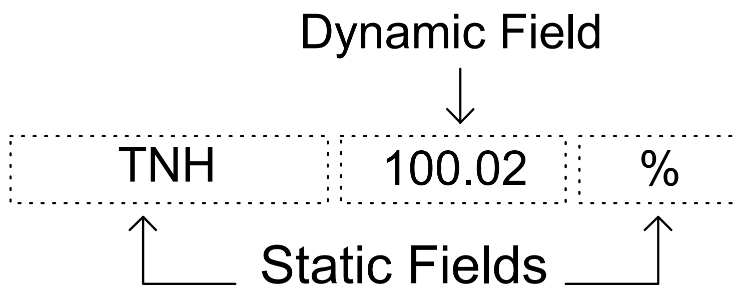


Figure 4-2. Static and Dynamic Fields Created with hpt2 Item Command

The colors of the dynamic portions of most items in an animated display must be specified in the item's command in terms of foreground and background colors. The foreground color of an item is the color in which its dynamic portion will be displayed; the background color is the color of the background of the dynamic portion of the item. When specifying colors for items, the foreground color of an item is expressed first and its background color is expressed second using the following notation:

`fore_color.back_color`

where the foreground and background colors are expressed using the two-character abbreviation from the list in section 4-2.4. and separated by a period (no spaces may be used). For example, if the color definition for an item were expressed as `bk.gr`, the dynamic portion of the item is displayed in black on a green background. If no background color definition is specified, the default background color will be used by the animated screen builder when displaying the item.

When defining a CDB signal pointname item in an item list file, the default foreground color for the pointname and its engineering units (the static portions of the item) is light grey while its default background color is the default background. The foreground and background colors of the dynamic portions of items in a display can be toggled based on the value of a CDB logic signal pointname (see Logic Color Options, section 4-2.5). The default foreground and background colors of the static portions of items in a display can also be defined using an item list file command (see `cset` description).

4-2.5. Logic Color Options

The foreground and background colors of the **dynamic portions** of display items can be made to toggle between two colors based on the values of CDB logic signal points. Known as the logic color option, three variations of this selection are available.

In the first option, a single logic point will toggle **only** the foreground color of the dynamic portion of an item by using the following notation when specifying the item's color in its command line:

`lpoint?tfcolor|ffcolor`

where:
- `lpoint` is the CDB logic signal pointname to be used to toggle colors,
- `tfcolor` is the foreground color when the CDB logic signal point's value is a logic " 1 ", or true, and
- `ffcolor` is the foreground color when the CDB logic signal point's value is a logic " 0 ", or false.

(The line between `tfcolor` and `ffcolor` is called the vertical bar. The vertical bar is usually the shifted function of the backslash key on a computer keyboard.)

Using the above option, the background color of the dynamic portion of the item will be default background.

In the second option, shown below, a single logic point will toggle **both** the foreground and background colors of the dynamic portion by using the following notation when specifying the item's color in its command line:

```
lpoint?tfcolor.tbcolor|ffcolor.fbcolor
```

where:

`lpoint` is the CDB logic signal pointname to be used to toggle colors,
`tfcolor` is the foreground color when the CDB logic signal point's value is a logic " 1 ", or true,
`tbcolor` is the background color when the CDB logic signal point's value is a logic " 1 ", or true,
`ffcolor` is the foreground color when the CDB logic signal point's value is a logic " 0 ", or false, and
`fbcolor` is the background color when the CDB logic signal point's value is a logic " 0 ", or false.

A third option allows four colors based on two logical signals. This four color option can be used anywhere the two color options can be used. The notation is as follows:

```
lpoint1&lpoint2?color3|color2|color1|color0
```

where:

`lpoint1` and `lpoint2` are the CDB logic signal pointnames to be used to toggle colors,
`color3` is the color when the CDB logic signal points' value are a logic " 1 " and " 1 ", or true and true,
`color2` is the color when the CDB logic signal points' value are a logic " 1 " and " 0 ", or true and false,
`color1` is the color when the CDB logic signal points' value are a logic " 0 " and " 1 ", or false and true, and
`color0` is the color when the CDB logic signal points' value are a logic " 0 " and " 0 ", or false and false,

4-2.6. Font Sizes

Text string and CDB pointname values and engineering units can be displayed in single- or double-high fonts. The last digit in such item commands will indicate the size of the font used in the display. In the examples above, a double-high font was used (`hpt2`) in the command to display a CDB signal, its pointname, value, and engineering units in a double-high font. Single-high fonts (`hpt1`) are used on most <I> displays to show the sitename, date/time, unit number of the display, and so on. Double-high fonts are used to display points, values, and engineering units in User-Defined Displays, the Alarm Display, the Control Constants Adjust Display, and so on.

4-2.7. Specifying Multi-Unit CDB Pointnames

Animated displays on multi-unit <I>s can have CDB points and command targets for all the units with which the <I> is capable of communicating. It must be understood that when points from other units are specified in a unit-specific or plant control display that the animated screen builder **does not** identify them on the display as being associated with any particular unit. Therefore, it is **extremely important** that CDB points and command targets for multiple units be identified. Text strings and boxes can be useful tools for labeling and grouping CDB points. Try to use the same method of identifying points or targets for units other than the currently selected one on all animated displays.

CAUTION

Identify any CDB points or command targets on an animated display for units other than the currently selected unit. This is to ensure that the user is made aware of which unit the point or command target is associated with.

To place a CDB point or command target from another unit on an animated display, use the following notation in the item command when specifying the pointname:

```
UN:pointname
```

where:

`UN` is the two-character unit name of the unit control signal database the point is associated with,
`:` is the delimiter between the unit name and pointname (the colon must be used to separate them), and
`pointname` is the CDB pointname.

For example, to display the percent of unit #2's (unit name G2) high-pressure shaft speed in the default foreground color on an animated display for unit #1 (the item list file will have an extension of `.A1`), the following item command would be entered in the item list file:

```
hpt1  start_x  start_y  df  G2:tnh
```

The pointname `TNH`, its value, and engineering units will be shown on the animated display in the specified location and color with no indication that the point is for unit #2. A text string should be added above, below, or to the side of the point to identify it as being unit G2's point/value; or a box could be drawn around the point and a text string used to label the box's contents as being from unit G2.

4-2.8. Animated Displays and the Synonym Feature

When specifying control signal database pointnames in item commands in an item list file, the CDB pointname **must** be used; the animated screen builder will **not** recognize synonyms in an item command.

If the Synonym Feature is enabled (and it is enabled by default each time the <I> is re-booted if a unit-specific `SYNONYM.DAT` file exists with entries) the animated screen builder displays the CDB pointname's synonym. If the Synonym Feature is disabled (from the Synonym Administration Display), the animated screen builder displays the CDB pointname.

4-3. SAMPLE DISPLAY & SCRIPT FILE

An animated display like the one shown in Figure 4-3 is the result of running the animated screen builder with the item list file shown in Figure 4-4. An analysis of the animated display in Figure 4-3 and its relationship to the item list file in Figure 4-4 will help to understand the flexibility and power of the animated screen builder.

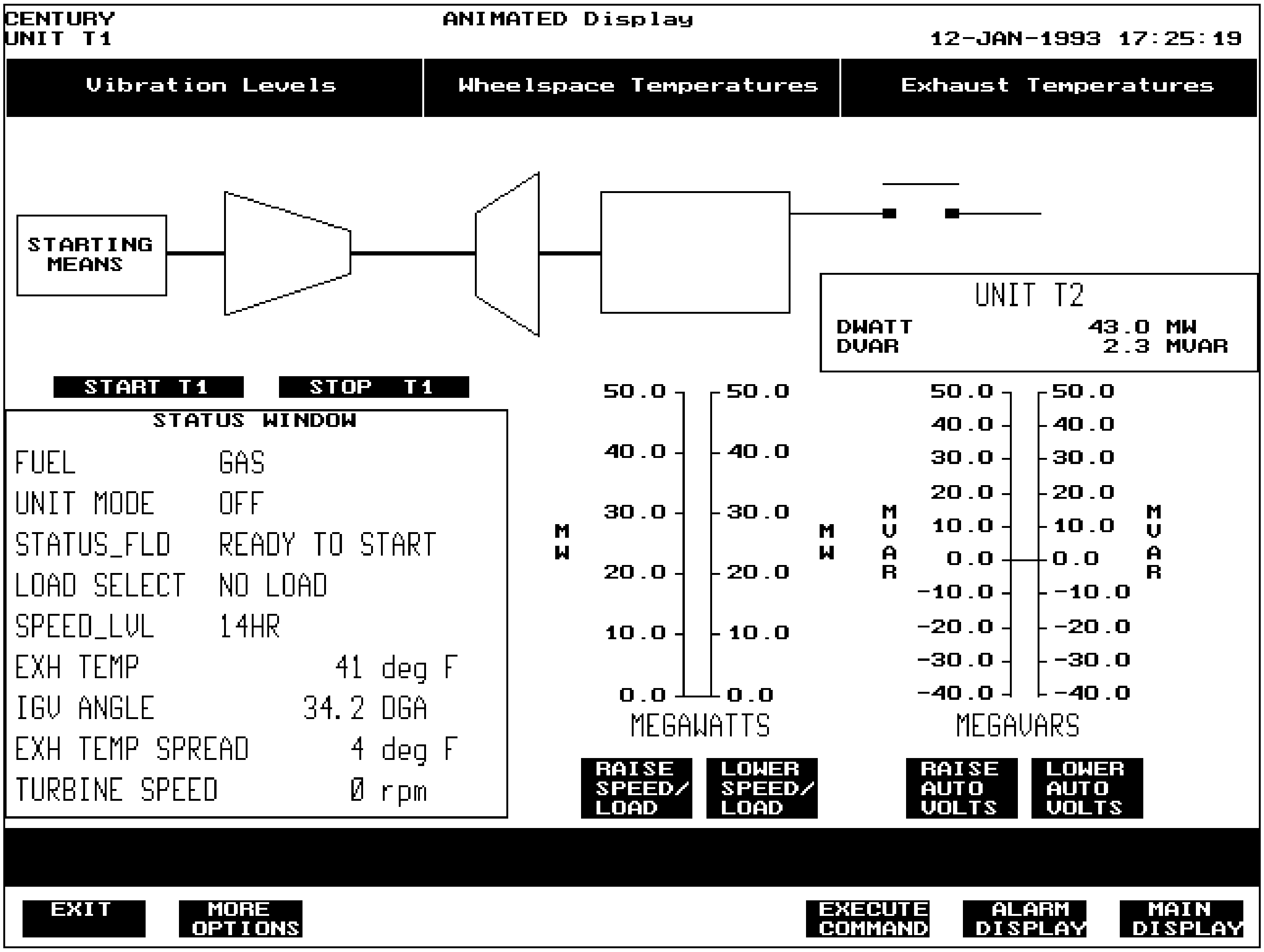


Figure 4-3. Sample Animated Display

The display in Figure 4-3 depicts line drawings of a single-shaft, heavy duty gas turbine and generator, its starting means, and the generator breaker. It also contains a Status Window, two bargraphs, some command targets and a **Hot Spot Bar** which contains three hot spots at the top of the display. By clicking in one of the three Hot Spots, the user can quickly switch to different animated displays and return to the display currently shown without first exiting to the Main Menu. There are also two points and their values from Unit T2 (a second unit which can be controlled from the <I>) that are contained in the box labelled **UNIT T2**.

In the item list file in Figure 4-4, lines **not** beginning with semicolons are commands to the animated screen builder program. Lines beginning with a semicolon are comments and are ignored by the animated screen builder program; however, they can be useful and informative for readers of the item list file. The first line without a semicolon instructs the animated screen builder to place a line of single-high text (a text string) in the default foreground color beginning at x-coordinate 350 and y-coordinate 975. This is the display's title, `ANIMATED` Display. The animated screen builder will display text in the case in which it was entered in the item list file including non-English characters/language. Comments can be added to lines in item list files by leaving at least one space after the last entry of an item command, typing a semicolon, and then typing the comment. The next item creates a hot spot for changing to the animated display file defined by `F:\RUNTIME\SAMPLE.A2` in the area of the sitename\unit number field at the top of the display.

```
;This is an item list file for an example of an Animated Display for Unit T1
;Title the display
tx1       350    975    df    "ANIMATED Display"
;Add a hot spot for changing to the Animated Display for Unit T2
hot       000    950    +200  +50     t2:sample
;Add "Hot Spot Bar" to top of display
sbox      000    940    mg    999     -075
;Draw light grey box around "Hot Spot Bar"
box      +000   +000    lg    999     -075
;Split the "Hot Spot Bar" into three sections
line     +333   +000   lg    +000     -075
line     +333   +000   lg    +000     -075
;Label the first "Hot Spot Bar" section "Vibration Levels"
tx1      -600   -045   df.mg "Vibration Levels"
hot      -066   +045   +333  -075     vib
;Label the second "Hot Spot Bar" section "Wheelspace Temperatures"
tx1      +363   -045   df.mg "Wheelspace Temperatures"
hot      -030   +045   +333  -075     wheelspc
;Label the third "Hot Spot Bar" section "Exhaust Temperatures"
tx1      +383   -045   df.mg "Exhaust Temperatures"
hot      -050   +045   +333  -075     exhaust
;Draw a combustion turbine, generator, and generator breaker
;First draw the starting means and label it
box       010    650    lg    +120    +100
tx1      +010   +050    l4de?rd|df    "STARTING"
tx1      +015   -025    l4de?rd|df    "MEANS"
;Next draw the axial compressor
line      175    775    lg    +100    -050
line     +100   -050    lg    +000    -050
line     +000   -050    lg    -100    -052
line     -100   -050    lg    +000    +150
;Draw a shaft to connect the starting means and axial compressor
sbox      175    698    lg    -045    +004
;Draw the turbine
line      375    650    lg    +000    +100
line     +000   +100    lg    +050    +052
line     +050   +050    lg    +000    -200
line     +000   -200    lg    -048    +050
;Draw a shaft to connect the compressor and turbine
sbox      375    698    lg    -098    +004
;Draw the generator
box       475    775    lg    +150    -150
;Draw a shaft to connect the turbine and generator
line      475    698    lg    -049    +004
;Draw a single line out of the generator to represent phase leads
line      625    750    lg    +075    +000;Draw a vertical bargraph for megawatts,
label it, and add speed/load command targets
.vba      440    200    bl    +225    +400   lru   dwatt   0     3210   10
&bar     df     dwatt
&end
tx2      +060   -045    bl    "MEGAWATTS"
cmd1     -040   -105   7 3 ! L70R4R_CPB  = 4  L70R 1 " RAISE" " SPEED/" " LOAD"
cmd1     +100   +000   7 3 ! L70R4L_CPB  = 4  L70L 1 " LOWER" " SPEED/" " LOAD"
;Draw a vertical bargraph for megavars, label it, and add auto voltage command
targets
.vba      700    200   bl    +225    +400   lru   dvar  -2570  3210   10
&bar     df     dvar
&end
tx2      +060   -045   bl    "MEGAVARS"
cmd1     -040   -105   7 3 ! L90VR_CPB   = 4  L70R 1 " RAISE" " AUTO" " VOLTS"
cmd1     +100   +000   7 3 ! L90VL_CPB   = 4  L70L 1 " LOWER" " AUTO" " VOLTS"
;Create a box for important status information and call it a Status Window
box       000    010   lg    +410    +500
tx1      +120   +475   gr    "STATUS WINDOW"   ;title
```

```
tx2      -100   -065   lg    "FUEL"
v2       +162   -000   df    ss43f
tx2      -162   -050   lg    "UNIT MODE"
v2       +162   -000   df    ss43
hpt2     -162   -050   df    status_fld
tx2      -000   -050   lg    "LOAD SELECT"
v2       +162   -000   df    ss43load
hpt2     -162   -050   df    speed_lvl
tx2      -000   -050   lg    "EXH TEMP"
hve2     +192   -000   df    ttxm
tx2      -192   -050   lg    "IGV ANGLE"
hve2     +192   -000   df    csgv
tx2      -192   -050   lg    "EXH TEMP SPREAD"
hve2     +192   -000   df    ttxsp1
tx2      -192   -050   lg    "TURBINE SPEED"
hve2     +192   -000   df    tnh_rpm
;Add Unit Start and Stop Command Targets
cmd1      040    525   12 1 ? L1START_CPB  = 4  L1S  1 "  START T1"
cmd1     +180   -000   12 1 ? L1STOP_CPB   = 4  L94X 1 "  STOP  T1"
;Add Megawatt and Megavar information for Unit T2
box       650    675   t2:l52gx?rd|gr +349  -120
tx2      +125   -050   t2:l52gx?rd|gr "UNIT T2"
hpt1     -110   -025   t2:l52gx?rd|gr t2:DWATT
hpt1     +000   -025   t2:l52gx?rd|gr t2:DVAR
```

Figure 4-4. - Item List File for Animated Display in Figure 4-3.

The next several lines create the Hot Spot Bar. A magenta-colored solid box is drawn (the solid box will highlight the hot spots, which otherwise would not be visible to the user). The solid box is then surrounded with a border of light grey by drawing a light grey box of the same dimensions beginning at the same coordinates. Next, the box is divided into three sections by drawing two light grey lines an equal distance apart and the same length as the height of the solid box. The three sections are then labeled and defined as **hot spots** areas that, when clicked on, will change the screen to the specified display. The first hot spot (on the left), entitled Vibration Levels, instructs the animated screen builder program to change the screen to the animated display defined by the item list file `VIB.An`. The second and third hot spots cause animated wheelspace and exhaust temperatures displays to be depicted respectively.

The combustion turbine, its starting means and generator, and the unit's generator breaker were drawn using the item commands, `line`, `box`, and `sbox`. Note how the starting coordinates of each component are expressed in absolute terms, and the starting coordinates of any other items used to draw the rest of the component are defined in relative terms.

Style Tip. Using absolute coordinates for the origins of each component (or group of components) and relative coordinates for any related instructions, the component (or group) can be moved anywhere in the animated display screen by changing only the origin point coordinates. Such a change will not affect the positions of other unrelated items or components. The ending coordinates of the line item commands used to draw the axial compressor and turbine diagonal lines are adjusted by one or two units in order to prevent lines from extending past each other. The same is done for the `sboxes` used to draw the unit's shafts.

Style Tip. To prevent items from extending past others or overlapping, adjust the origin or endpoints by adding or subtracting one or two units as required to neaten the display. (The over-extension occurs because the size of the x and y units of the grid are not equal; that is, the length of the horizontal axis is not equal to the height of the vertical axis while both have the same number of units, 1000.)

The `tx1` item commands labelling the Starting Means use a logic color option for specifying the color of the text. The text strings will be displayed in one of two colors depending on the value of CDB logic signal `L4DE`. When `L4DE` is a logic " 0 " the text will be displayed in the default foreground color. However, when `L4DE` is a logic " 1 ", the text will be displayed in red. (`L4DE` is the logic signal which will be a logic " 1 " when the Starting Means is running.) In both cases, the default background color will be used when displaying the text since a background color parameter was not specified.

Style Tip. By displaying items in foreground and/or background colors controlled by the value of a CDB logic signal, important information about the status of the item or unit operation can be quickly conveyed to the viewer. This logic color option was used for the generator breaker. In this case, two line item commands were used to indicate generator breaker status. The first line will be displayed in green when the CDB logic signal `L52GX` is a logic " 0 " and then change to the default background color when `L52GX` is a logic " 1 ". The second line item command, at a slightly lower position on the display (its starting y-coordinate is 30 units below the first line), will be displayed in the default background color when `L52GX` is a logic " 0 ". The color of this item will change to red when `L52GX` is a logic " 1 ". By using this scheme, the generator breaker status line will appear to move and change colors on the display as the generator breaker is closed or opened. This illusion is created by using two lines which seem to disappear when their colors change to default background.

Two vertical bargraphs, one for megawatts, the other for megavars, were added to the display and labeled. Below this item are two pairs of immediate action-type command targets. One pair (below the megawatt bargraph) is used to raise/lower the turbine speed reference/load while the other is used to raise/lower the generator exciter regulator automatic voltage setpoint. A Status Window was created by drawing a light grey box and labeling it with green text in a single-high font. Several CDB points and their values are placed inside the box (using relative starting coordinates). A combination of different item commands are used to display most of the information without displaying the CDB signal pointnames. Two arm/execute-type targets are added just above the Status Window to start and stop the unit. The targets were drawn just one row high and 12 columns wide. Finally, a box was drawn to enclose two pointnames, their values, and engineering units from unit T2 (megawatts and megavars). A text string was placed in the box to indicate that the points were displaying information for unit T2. The logic color option is used to toggle the colors of the box, the text string, and the points when unit T2s generator breaker was opened/closed for quick recognition of the power generating status of unit T2. Multi-item components in the display defined by the item list file in Figure 4-4 were created using absolute origin coordinates for the first item command. This item list file writing technique can be added to any component easily without changing all the relative coordinates of any successive components. By using creative techniques such as those employed for the Starting Means text strings and the generator breaker status line, unique and informative animated displays can be created. In the next section, the individual item commands, their format, and their usage will be explained.

4-4. RUNNING THE SCREEN BUILDER PROGRAM

The animated screen builder is an executable file (`ANIM.EXE`) in the `G:\EXEC` directory of an <I>. When invoked with the name of an animated display item list file, this program will produce an <I> display on the screen containing graphic items. The item list file contains item commands that are read and executed by the animated screen builder to produce an animated display. Users of an <I> who do not have the proper password authority are limited to viewing animated displays that are accessible from the Main Menu or via hot spots on other animated displays.

Those users with access to Maintenance Password level (or above) can exit from the <I> Main Menu and run the animated screen builder from the DOS command line. Since `G:\EXEC` is in the path statement for an <I>, the animated screen builder can be invoked from any directory in the <I>. The command line format for running the animated screen builder is:

```
anim list_file
```

where `list_file` is the name of the file **only**, not including the filename extension or path to the file of an animated display item list file. The animated screen builder will search for the filename and produce the animated display defined by the file's contents. For example, to invoke the animated display defined by the item list file `F:\RUNTIME\UNIT.A1`, the following command would be typed in at the DOS prompt of any directory followed by pressing `Enter` (if unit 1 is the currently selected directory when the <I> Main Menu was exited):

```
anim unit
```

Item list files must have a filename extension of either `.A` or `.An`, where the lower-case `n` stands for the unit number for which the display/file was created. For example, `UNIT.A1` might be the name of a item list file containing unit-related information for unit number 1. The item list files for animated displays should be stored in the `F:\RUNTIME` directory. (This is the directory where the Main Menu and User-Defined Display definition files are stored along with other display-related files.) Item list files in the `RUNTIME` directory should have an `.An` filename extension. Item list files can also be stored in the unit-specific directory, `F:\UNITn`, where `n` stands for the unit number which can be controlled by the <I>. However, item list files in the unit-specific directory must only have a filename extension of `.A`. The animated screen builder will ignore any item list file in the unit-specific directory with a unit number in the filename extension.

4-5. CREATING AN ANIMATED DISPLAY

To create a unit-specific animated display, access to the DOS command line must first be gained by enabling the current status of Maintenance Password level (or above). Once at the DOS command line, type the following command to invoke the animated screen builder:

```
anim list_file
```

and then press `Enter`. The `list_file` filename specified should be unique, descriptive, no more than eight characters in length, and should not contain a filename extension as the animated screen builder will create an extension with the unit number of the currently selected unit (that is, the unit which was selected when the <I> Main Menu was exited). The animated screen builder searches `F:\RUNTIME` for an item list file with the filename specified, and then search the unit-specific directories. If the file is not found, a blank, empty animated display will appear on the screen with the message `"Unable to locate list_file."` Clicking on the MORE OPTIONS and EDIT FORM display targets in succession invoke one of two possible ASCII text editors (QEdit or the MS-DOS Editor). The editor that is called will display an empty file with the `list_file` filename specified. (The default directory for creating item list files using this method is `F:\RUNTIME`; the default filename extension is `.An` , where the lower-case n is the number of the currently selected unit.) Using the ASCII text editor, the item list file for an animated display can be written using animated display item commands.

Starting the screen builder and accessing the item list file in the above manner allows the user to quickly view the results of writing or editing the item list file. This can be done by saving any changes and exiting the text editor. The animated screen builder reads the item list file and immediately displays the effects of changes or additions. Further changes or additions can be quickly made by clicking on the MORE OPTIONS and EDIT FORM display targets (in succession) to return to the text editor and item list file. An animated display can be quickly created and refined using this method.

Multi-unit <I>s can have a single animated display showing information for all the units controlled by the <I>. Such a display is sometimes called a plant control display. The item list file for a plant control display must be created from the DOS command line using an ASCII text editor. It must reside in the `F:\RUNTIME` directory, have a unique filename (such as PLANT), and have a filename extension of `.A0`. Once the file has been created, it can be edited from the display itself by clicking on the MORE OPTIONS and EDIT FORM display targets in succession. The Current Status of Maintenance Password level, or above, must be enabled.

4-6. ADDING A DISPLAY TO THE MAIN MENU OR ANOTHER ANIMATED DISPLAY

After an animated display has been created, it can be added to the Main Menu of an <I> by using an ASCII text editor and adding the following line to an appropriate section of the Main Menu definition file `F:\RUNTIME\MENU.DAT:`

```
" Display Name"  "g:\exec\anim.exe"  "list_file"
```

where:

`Display Name` is the title of the animated display,
`g:\exec\anim.exe` is the path to and the name of the animated screen builder, and
`list_file` is the name of the file ONLY (do not include the path or filename extension) of the display.

4-7. EDITING AN EXISTING ANIMATED DISPLAY

Editing an existing animated display requires Maintenance Level Password privileges (or above). Those users with access to the Maintenance password level or above (after enabling the Current Status of their password level), can select the desired animated display, click on the MORE OPTIONS display target, and then click on the EDIT DISPLAY target. This will start an ASCII text editor and load the item list file of the current display into the editor for viewing or modification. Saving the changes and exiting the editor returns to the screen display where the results of the modifications can be seen. Care should be taken to disable the password status when editing is complete.

NOTE

If there is insufficient RAM on the <I> to load the MS-DOS Editor or the `list_file`, a message will be briefly displayed. If this occurs, the user must exit the Main Menu to the DOS command line and edit the `list_file` directly using the MS-DOS Editor, or run the animated screen builder.

4-8. ITEM COMMANDS

Individual item list file commands along with their format and usage are explained below. Examples are provided where possible. Items that can be placed in animated displays include:

- Horizontally displayed CDB points, their values, engineering units, and any combination of those three (pointname only, value and engineering units only, and so on).
- Horizontally displayed text strings see (`txn`, and `txln`)
- CDB double-word values see `fxn`
- Box outlines and solid boxes (a box filled with a color) (see `box`, and `sbox`)
- Circle outlines and solid circles (a circle filled with a color) (see `cir`, and `scir`)
- Command targets (see `cmd1`)
- Hot spots (see `hot`)
- Vertical bargraphs (see `.vba`, `&bar`, and `&end`)
- Dynamic x-y plotting of plant conditions (see `.xya`)

4-8.1. hpt1, hpt2

These commands display a CDB signal pointname, its value, and its engineering units horizontally. The form of the display is:

```
POINTNAME  VALUE  UNITS
```

The `hpt1` and `hpt2` commands cause information to be displayed in single-high and double-high fonts respectively. The specified foreground and background colors are used to display the dynamic portion of the item (the point's value field). The format for this item's command is:

```
hptn  start_x  start_y  fore_color.back_color  pointname
```

where:
`hptn` means either `hpt1` or `hpt2`,
`start_x` is the starting x-coordinate value,
`start_y` is the starting y-coordinate value,
`fore_color` is the color of the CDB pointname's value,
`back_color` (if specified) is the background color of the CDB pointname's value, and
`pointname` is the CDB signal point whose name, value, and engineering units are displayed.

Both logic and variable CDB points can be displayed with the `hptn` instruction. Enumerated state pointnames can also be displayed using `hptn`, however there are no units associated with enumerated state points; only the text strings associated with their current states (GAS, AUTO, etc. . .). One of the logic color options may be used when specifying the color(s).

4-8.2. `n1, n2`

These item commands will horizontally display a CDB signal's pointname ONLY (that is, no value or engineering units for the point will be shown). The form of the display will be:

```
POINTNAME
```

The `n1` and `n2` commands will cause the information to be displayed in a single-high font or double-high font respectively. The format for this item command is:

```
nn  start_x  start_y  fore_color.back_color  pointname
```

where:

`nn` means either `n1` or `n2`,
`start_x` is the starting x-coordinate value,
`start_y` is the starting y-coordinate value,
`fore_color` is the color of the CDB pointname,
`back_color` (if specified) is the background color of the CDB pointname, and
`pointname` is the CDB signal point whose name is to be displayed.

This item is dynamic if the color is defined using a logic variable, otherwise it is static. If synonyms are being used (see Section 4-2.8), the synonym is what will be shown.

4-8.3. `ln1, ln2`

These item commands will horizontally display the LONG version of a CDB signal's pointname ONLY (that is, no value or engineering units for the point will be shown). The form of the display will be:

```
LONGNAME
```

The `ln1` and `ln2` commands will cause the information to be displayed in a single-high font or double-high font respectively. The format for this item command is:

```
lnn  start_x  start_y  fore_color.back_color  pointname
```

where:

`lnn` means either `ln1` or `ln2`,
`start_x` is the starting x-coordinate value,
`start_y` is the starting y-coordinate value,
`fore_color` is the color of the long version of the CDB pointname,
`back_color` (if specified) will be the background color of the long version of the CDB pointname, and
`pointname` is the CDB signal point whose longname is to be displayed.

This item will be dynamic if defined with a logic color option; otherwise it is static.

4-8.4. v1, v2

These item commands display a CDB signal pointname's value ONLY (that is, no pointname or engineering units for the point will be shown). The form of the display will be:

```
VALUE
```

The `v1` and `v2` commands will cause the information to be displayed in a single-high font or double-high font respectively. The specified foreground and background colors will be used to display the dynamic portion of the item (the point's value field). The format for this item command is:

```
vn  start_x  start_y  fore_color.back_color  pointname
```

where:

`vn` means either `v1` or `v2`,
`start_x` is the starting x-coordinate value,
`start_y` is the starting y-coordinate value,
`fore_color` is the color of the CDB signal pointname's value,
`back_color` (if specified) is the background color of the CDB pointname's value, and
`pointname` is the CDB signal point whose value will be displayed.

CDB logic, variable, and enumerated state points can be displayed using `vn`. One of the logic color options may be used when specifying the item's color(s). For more information refer to the Color Option Section.

4-8.5. hve1, hve2

These item commands horizontally display a CDB signal's value and engineering units ONLY (that is, no pointname for the point is shown). The form of the display is:

```
VALUE UNITS
```

The `hve1` and `hve2` commands cause the information to be displayed in a single-high font or double-high font respectively. The specified foreground and background colors are used to display the dynamic portion of the item (the point's value field). The format for this item command is:

```
hven  start_x  start_y  fore_color.back_color  pointname
```

where:

`hven` means either `hve1` or `hve2`,
`start_x` is the starting x-coordinate value,
`start_y` is the starting y-coordinate value,
`fore_color` is the color of the CDB signal pointnames value,
`back_color` (if specified) is the background color of the CDB pointname's value, and
`pointname` is the CDB signal point whose value and engineering units are to be displayed.

CDB logic, variable, and enumerated state points can be displayed using `hven`.

4-8.6. e1, e2

These item commands horizontally display a CDB signal's engineering units ONLY (that is, no pointname or value for the point is shown). The form of the display is:

```
UNITS
```

The `e1` and `e2` commands cause the information to be displayed in a single-high font or double-high font respectively. The format for this item command is:

```
en  start_x  start_y  fore_color.back_color  pointname
```

where:
- `en` means either `e1` or `e2`,
- `start_x` is the starting x-coordinate value,
- `start_y` is the starting y-coordinate value,
- `fore_color` is the color of the CDB signal pointnames engineering units,
- `back_color` (if specified) is the background color of the CDB pointname's engineering units, and
- `pointname` is the CDB signal point whose engineering units are to be displayed.

This item can be dynamic if the color is defined with a logic variable, otherwise it will be static.

4-8.7. tx1, tx2

These item commands will horizontally display a specified text string. "`tx1`" will display the text string in a single-high font; `tx2` will display the text string in a double-high font. The format for this item command is:

```
txn  start_x  start_y  fore_color.back_color "text_string"
```

where:
- `txn` means either `tx1` or `tx2`,
- `start_x` is the starting x-coordinate,
- `start_y` is the starting y-coordinate,
- `fore_color` is the color in which the text string will be displayed,
- `back_color` (if specified) is the text string's background color, and
- `"text_string"` is the text that will be displayed.

The text string must be enclosed in double quotation marks. Spaces in the text string are permitted. The text string displayed will use the case(s) specified in the item list file. One of the logic color options may be used when specifying the item's color(s).

4-8.8. `txl1, txl2`

These item commands will horizontally display one of two specified text strings based upon the value of a CDB logic signal point using the foreground and background colors specified. The text string will be displayed in a single-high font using "`txl1`" and a double-high font when using `tx12`.

The format for this item command is:

```
txln  start_x  start_y  fore_color.back_color   pointname   "true_string"
   "false_string"
```

where:

`txln` means either `txl1` or `txl2`,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore_color` is the color of the text string,
`back_color` is the text string's background color,
`pointname` is the CDB logic signal point whose value will decide which text string is displayed,
"`true_string`" is the text string displayed when the logic signal pointname's value is a logic " 1 ", and
"`false_string`" is the text string displayed when the logic signal pointname's value is a logic " 0 ".

The text string must be enclosed in double quotation marks; spaces in the string are permitted. The text string displayed will use the case(s) specified in the double quotation marks in the item's command in the item list file. One of the logic color options may be used when specifying the item's color(s).

4-8.9. `fx1, fx2`

These item commands will horizontally display a 32-bit value consisting of two specified 16-bit CDB variable pointnames. The value will be displayed in a single-high font when using `fx1` and in a double-high font when using `fx2`. The format for this item command is:

```
fxn start_x start_y fore_color.back_color MSW_pointname LSW_pointname gain decimals
```

where:

`fxn` means either `fx1` or fx2,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore_color` is color of the text strings,
`back_color` is the text string's background color,
`MSW_pointname` is the CDB variable signal pointname for the most significant word of the 32-bit value,
`LSW_pointname` is the CDB variable signal pointname for the least significant word of the 32-bit value,
`gain` is the decimal value that the 32-bit value can be multiplied by to get into the proper units for display, and
`decimals` is the number of decimal places to use when displaying the 32-bit value.

Examples of 32-bit values are counters and timers (such as manually initiated starts, fired hours, emergency trips, mass steam flow accumulators, and so on). One of the logic color options may be used when specifying the item's color(s).

4-8.10. `line`

This item command will create a line according to origin and endpoint location definitions. The color of this line may also be defined. Lines can be drawn at any angle and can be of any length. Lines are drawn only in the foreground color specified (a background color, if specified, will be ignored). The format for this item's command is:

```
line    start_x    start_y    fore_color    end_x    end_y
```

where:

`line` is the item command,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore_color` is the color in which the line will be drawn,
`end_x` is the ending x-coordinate, and
`end_y` is the ending y-coordinate.

The ending x- and y-coordinates can be specified in either absolute or relative terms (relative in this case meaning to the **`line`** origin) and define the line's length.

NOTE

If the item immediately following a `line` item command has relative starting coordinates, they are relative to the starting coordinates of the `line` command, not its ending coordinates. The foreground logic color option may be used when specifying the foreground color for this item.

4-8.11. `box`

This item command will draw the outline of a box in the specified color between origin and endpoint location definitions. The box can be rectangular or square but must be drawn in the specified foreground color. A background color, if specified, will be ignored. It will be empty except for the background of any items it may surround. The box will be drawn along the x- and y- axes. The format for this item's command is:

```
box    start_x    start_y    fore_color    end_x    end_y
```

where:

`box` is the item command,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore_color` will be the color of the box outline,
`end_x` is the ending x-coordinate, and
`end_y` is the ending y-coordinate.

The ending x- and y-coordinates can be specified in either absolute or relative terms (relative in this case meaning to the box's origin). The starting coordinates and the ending coordinates are at diagonally opposite corners of the item. Specifying the ending coordinates in relative terms makes moving the box a simple matter of changing its origin coordinates only. The foreground logic color option may be used when specifying the item's foreground color.

4-8.12. `sbox`

This item command will draw a solid or filled box of the specified color between origin and endpoint location definitions. The solid box can be rectangular or square and is drawn in the specified foreground color only; a background color, if specified, will be ignored. The solid box will be drawn along the x- and y- axes. The format for this item's command is:

```
sbox    start_x    start_y    fore_color    end_x    end_y
```

where:

`sbox` is the item command,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore_color` will be the color of the box,
`end_x` is the ending x-coordinate, and
`end_y` is the ending y-coordinate.

The ending x- and y-coordinates can be specified in either absolute or relative terms (relative in this case meaning to the box's origin). The starting coordinates and the ending coordinates are at diagonally opposite corners of the item. Specifying the ending coordinates in relative terms makes moving the solid box a simple matter of changing its origin coordinates only. The foreground logic color option may be used when specifying the item's foreground color.

Style Tip. Do not specify a solid box which extends higher than a y-coordinate of 950 units; it will overwrite the sitename and unit number in the upper left corner of the display. Do not specify a solid box with a blue background; blue is the background color of the Alarm Window.

4-8.13. `cir`

This item command will draw a circle centered at the origin ending at the endpoint. The circle may be any size, but all the points on the circle must be on the screen. The circle will be drawn in the specified foreground color. It will be empty except for the background of any items it may surround. A background color (if specified) will be ignored. The format for this item's command is:

```
cir    start_x    start_y    fore_color    end_x
```

where:

`cir` is the item command,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore_color` will be the color of the circle outline, and
`end_x` is the ending x-coordinate.

The endpoint x-coordinate can be given in absolute coordinates; however, it is easier and more common to specify it relative to the circle's origin. The foreground logic color option may be used when specifying the item's foreground color.

4-8.14. `scir`

This item command will draw a solid or filled circle of the specified color centered at the origin and passing through the endpoint location. The solid circle will be drawn in the specified foreground color only; a background color, if specified, will be ignored. The circle may be any size, but any points on the circle must also be on the screen. The format for this item's command is:

```
scir    start_x    start_y    fore_color    end_x
```

where:
`scir` is the item command,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore_color` is the color of the box, and
`end_x` is the ending x-coordinate.

The ending x-coordinate can be specified in absolute terms. It is easier and more common, however, to specify the endpoint relative to the circle's origin. Specifying the ending coordinates in relative terms makes moving the solid circle a simple matter of changing its origin coordinates only. The foreground logic color option may be used when specifying the item's foreground color.

Style Tip. Do not specify a solid circle which extends higher than a y-coordinate of 950 units; it will overwrite the sitename and unit number in the upper left corner of the display. Do not specify a solid circle with a blue background; blue is the background color of the Alarm Window.

4-8.15. `cmd1`

This item command will create a command target on a display. The command target can be defined as one of three types: arm/execute, immediate action, or an analog setpoint. The command target will be displayed beginning at the specified x- and y-coordinates and will contain the designated number of rows (height) and columns (width). Single-high font text strings can be imbedded in the command target. As with command targets on User-Defined Displays, the text string can be made to turn yellow based on the value of a specified CDB logic signal. The format of the item's command is:

```
cmd1 start_x start_y cols rows type pointname = value fdbk fdbk_sense "text_string"
```

where:
`cmd1` is the item command,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`cols` is the width of the target (in columns),
`rows` is the height of the target (in single-high font rows),
`type` is the type of command target (see below),
`pointname` is the CDB signal pointname which is to be affected by the target,
= means to set the pointname equal to a value (there **must** be at least one space before and after the equal sign),
`value` is the number (in hexadecimal) to which the pointname is to be set equal to (a value must be specified for all command targets),
`fdbk` is the CDB logic signal pointname that will be used to toggle the target's text string (a feedback must be specified, even if it's not used),
`fdbk_sense` is the value of the feedback logic signal that will change the target's text string, and
`"text_string"` is the text to be placed in the target. The text string must be contained in double quotations.

1000 grid units in the horizontal direction is 80 columns wide. A command target can be as much as 80 columns wide. The height of command targets is limited to three single-high font rows. Pointnames affected by command targets can be command pushbutton types, logic state types, or enumerated point types.

Valid single-character entries for the command target type are:

" **?** " for an arm/execute command target
" **!** " for an immediate action command target
" **#** " for an analog setpoint target

The equal sign " = " must be included in the item command and must have at least one space to either side. The type entry depends on the kind of command target being created. Command pushbutton CDB points (the pointname usually ends in `_CPB`) can have a hexadecimal value between 00 and FF. The hexadecimal value for all units running at 16 Hz is the number of sixteenths (1/16) of a second that the command pushbutton logic signal will be held at a logic " 1 ". For unit control systems running at 32 Hz, the units of the value entrered is 1/32 of a second. Logic state CDB points (the points usually end in `_CMD`) can have a value of zero or one. Enumerated command CDB points can have the appropriate decimal value specified (usually a number between zero and seven). Analog setpoint command targets have their values specified from the animated display after they are armed, just as on User-Defined Displays. To define an analog setpoint command target in the item command, use the equal sign followed by the letter " X " for the value entry (the " X " is just a placeholder in the item command). Command targets on an animated display do not require the use of a feedback signal to toggle the target's text string. However, if one is not used, the word " `none` " must be specified in the item's command. The feedback sense can be either a zero or a one. When a feedback sense is specified, the target's text string(s) will toggle to yellow when the logic point's value is equal to the feedback sense specified. The command target's text string(s) will toggle to black when the feedback signal's logic state is not equal to the feedback sense specified. A feedback sense must be specified even if it's " `none` " (although its value is irrelevant) in order for the item command to function correctly when being read by the animated screen builder. The text string(s) to be displayed in the target (one text string per row of height) must be enclosed in double quotation marks. Blank lines in a command target must be specified by an empty pair of double quotation marks. If the bottom row of a command target contains no text string it is not necessary to specify it with a pair of empty double quotation marks. The animated screen builder will display command target text strings in the case(s) specified in the item's command.

Style Tip. Begin each command target text string with a space; without a leading space, the first character of the text string will overlap the left border of the command target.

Style Tip. All targets should be labeled in upper case text.

4-8.16. hot

This item command will define a hot spot in a display. Clicking on the area defined by the hot spot will instruct the animated screen builder to switch immediately to the specified animated display (switching animated displays usually requires exiting to the Main Menu and clicking on an animated display selection). The format of the item's command is:

```
hot  start_x  start_y  end_x  end_y  list_file
```

where:
- `hot` is the item command,
- `start_x` is the starting x-coordinate,
- `start_y` is the starting y-coordinate,
- `end_x` is the ending x-coordinate,
- `end_y` is the ending y-coordinate, and
- `list_file` is the name of the file of the animated display to be switched to.

The ending x- and y-coordinates can be specified in either absolute or relative terms (relative, in this case, meaning to the hot spot's origin). Specifying the coordinates in relative terms makes moving the hot spot easier by changing its origin coordinates only. The `list_file` must be the name of the file ONLY (not including the path or filename extension) of an existing animated display item list file. After clicking on a hot spot, to switch to another animated display, the viewer can switch back to the previous one by clicking on the PREV PAGE display target. Animated displays can be nested, that is an animated display accessed by clicking on a hot spot can have hot spots, and those animated displays can also have hot spots, and so on. Clicking on PREV PAGE will "back up" one animated display. Use care not to overlap multiple hot spots on a display. When clicking in an area of overlapping hot spots, the first hot spot to be defined will be selected. Also, do not overlap hot spots and command targets; clicking in such an overlapping area will enable/execute the command target and the hot spot will be ignored.

4-8.17. vba

This item command is a part of a complex item (multiple lines are used to define the item) that is used to define a vertical bargraph. Vertical bargraphs can be used to graphically display information about unit operation using either CDB logic or variable signal points. The scaling of the bargraph's axes can be specified to within a range that will make it easier for the user to comprehend (that is, improve the resolution). The bargraphs can be constructed so that the bars emanate from the top or bottom of the graph. Also, using the logic color option, the bars in a bar graph can be made to change color to highlight them to the user. To define a bargraph, a minimum of three item lines must be used to make up the item's command. The first line, which starts with `.vba`, defines the position, color, size, options, scaling information, and divisions on the axes. The second (and any that follow) line which begins with `&bar`, defines the CDB signal pointname whose value is to be graphed. There can 64 CDB pointnames/bars on a single bargraph and the number of bargraphs that can be incorporated into a display is restricted only by their size and available display area. The last line which begins with `&end` signifies the end of the item command to the animated screen builder and must be included in the item's command. The format for this item command is:

```
.vba start_x start_y s_color end_x end_y options pointname raw_bottom raw_top ticks
&bar bar_color bar_pointname
   .
   .
   .
&end
```

where:

`.vba` is the beginning of the item command for a vertical bargraph (the period must be included),
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`s_color` is the static color of the bargraph's axes, labels, and ticks,
`end_x` is the ending x-coordinate for the bargraph,
`end_y` is the ending y-coordinate for the bargraph,
`options` is the list of options for the axes labels and/or bar emanation(s),
`pointname` is the CDB pointname whose scale type will be used for the axes scaling,
`raw_bottom` is the bargraph's lower limit (expressed in raw decimal counts of the pointname's scale type),
`raw_top` is the bargraph's upper limit (expressed in raw decimal counts of the pointname's scale type),
`ticks` is the number of "tick marks" (or divisions) the axes should have,
`&bar` is the command to the animated screen builder that tells it to place a bar on the previously defined bargraph (the ampersand must be included),
`bar_color` is the color the bargraph will displayed in,
`bar_pointname` is the CDB signal pointname whose value is to be graphed,
`&end` is the command to the animated screen builder that the bargraph definition is complete (the ampersand must be included).

The specified `s_color` will be the color of the vertical bargraph's left and right axes, its zero point (usually the bottom axis), its engineering units, and tick marks (if specified). A background color may be used if desired. If used, the region will be filled with the specified color. The `end_x` coordinate will define the horizontal length of the bargraph and the `end_y` coordinate will define its height.

The following options can be specified which will define the bargraph's labeling and characteristics:

L	Left axis tick marks labeled
R	Right axis tick marks labeled
N	Neither axis tick marks labeled
U	Engineering Units name (from CDB pointname scale type) printed on labeled axis
+	Bars drawn up from bottom of graph (default)
-	Bars drawn down from top of graph

Table 4-1. Bargraph Options

The bargraphs in the item list file in Figure 4-4 each have the options `LRU` specified in the item's command. These options tell the animated screen builder to display the bargraphs with both the left and right axes labeled and to use engineering units on the labeled axis. The default direction in which bars are drawn is from the bottom of the graph. The pointname specified instructs the animated screen builder to use the scale type from that CDB signal pointname (logic or variable) when displaying and scaling the bargraph. All bars on the bargraph should have the same scale type as the one specified in the first line of the item command. For example, if two bargraphs were defined using pointnames with varying scale types (for example `PRESS` -- max CDB scale of 2048.0 psi, and `HUMID` -- max CDB scale of 0.2500 # water/# air), the individual bars may not appear to be in proportion to each other. The `raw_bottom` value is the bargraph's lower limit expressed in raw decimal counts. The value must be expressed in proportion to the max CDB scale counts of the bargraph's specified pointname. The `raw_top` value is the bargraph's upper limit expressed in raw decimal counts. This value, too, must be expressed in proportion to the max CDB scale counts of the bargraph's specified pointname. For example, the megawatt bargraph in Figure 4-3 was defined in the item list file using the pointname, `raw_bottom`, `raw_top` and values as follows:

```
pointname   raw_bottom     raw_top
  dwatt        0           3210
```

The `raw_bottom` and `raw_top` values were calculated by determining the maximum CDB scale value in both engineering units and raw counts for the scale type of the specified CDB signal pointname. This information can be determined from the gain and offset of the scale type, found in SCLEDATA.DAT, for most of the CDB signal points. `DWATT` has engineering units of `MW`; the scale type for `MW` units is `MWATT`, and the gain and offset for this scale type is 512 and 0 respectively. The minimum and positive maximum values of the signal can be determined from these two variables. The positive maximum value is equal to the gain plus the offset and the minimum value is zero plus the offset. The maximim and minimum values can then be related to the maximum and minimum raw count values, 32768 and zero respectively.

The megawatt bargraph in Figure 4-3 had its `raw_bottom` value set to zero, for zero megawatts. Its maximum value was 50 megawatts, set by the `raw_top` value 3210. The `raw_top` value was calculated using the ratio below:

$$\frac{50.0}{512.0} = \frac{\mathrm{raw\_top}}{32768}$$

Solving for `raw_top` yields:

$$\mathrm{raw\_top} = \frac{50x32768}{512.0} = 3200.0$$

Entering a `raw_top` value of 3200 would not show the top `tick` value of `50 MW`. By adding a small additional number of counts to the result (in this case ten) the overall scaling of the bargraph will be negligibly affected and the top `tick` value of 50 will be displayed on the bargraph. (3210 counts corresponds to 50.156 MW.)

The megavar bargraph in Figure 4-3 was scaled using the pointname `DVAR`. The CDB signal `DVAR` has engineering units of `MVARS`, and the minimum and maximum count values are zero and 32767, respectively (as can be seen from Figure 4-5). The positive maximum scale CDB value for `MVARS` is `512.0 MVARS`. The megavar bargraph had its minimum value set to `-40 MVARS` and its maximum value set to `50 MVARS`. The `raw_top` value for the megavar bargraph was specified to be the same as that for the megawatt bargraph (3210) using the same ratios as above. The `raw_bottom` value was calculated with the knowledge that a leading power factor (leading MVARS) is considered negative in the Mark V turbine control system using the following formula:

$$\frac{-40}{512.0} = \frac{\mathrm{raw\_bottom}}{32768}$$

Solving for `raw_bottom` yields:

$$\text{raw\_bottom} = \frac{-40x32768}{512.0} = -2560.0$$

A value of -10 counts was added to the result to enable the display of the minimum `tick` value produced the `raw_bottom` value of -2570 for the megavar bargraph. The `ticks` value in the item command's first line tells the animated screen builder how many `ticks` (divisions) to scale the axes with. The animated screen builder will automatically calculate and scale the bargraph using the specified number of `ticks` and the `raw_bottom` and `raw_top` values. Attempts to use a large number of `ticks` will cause the animated screen builder to place them one on top of another, making them difficult or impossible to read.

4-8.17.1. `&bar` The second line (and others that follow) in a vertical bargraph item command tells the animated screen builder which CDB signal pointname values to plot on the bargraph. Each line that defines a bar must begin with `&bar` (the ampersand must be included). The bar will be displayed in the `bar_color` color specified. The logic color option may be used to toggle the color of a bar based on the value of a specified CDB logic signal pointname. The last entry on an `&bar` line, `bar_pointname`, tells the animated screen builder which CDB signal pointname value to plot.

4-8.17.2. `&end` To end the definition of a vertical bargraph, the `&end` command must be used. The ampersand must be included.

4-8.18. .xya

The `.xya` option is a multi-item command that permits dynamic x - y plotting of plant conditions. Changes in real-time data received from the unit correspond to movement of a cross-hair relative to an x, y bounding box axis. A type of window, the bounding box defines the area in which the cross-hair will be visible. The scaling of the axis borders of this box is defined within the item list file. Besides being scaled, the sides of the bounding box can be labeled and set off by tick marks in such a way as to accommodate any plant variable. If the tick marks defined for the box include the plot's origin point (0, 0), the x-y axes will automatically be drawn within the box when the display is called (this presumes that the x-y axes has been selected to be included in the display (see `axis_options`). In addition to the cross-hair, static lines or curves can be defined for the display to provide a visual reference for items such as valid operating parameters.

In circumstances where allowable operating parameters are contingent on a plant variable, a family of curves or lines can be defined which, though not actually depicted, will be used to perform an interpolation calculation. The result of this operation will then be displayed as a dynamic line or curve.

NOTE

Only one variable can be plotted per bounding box. Multiple plots within a single display will require an equal number of bounding boxes.

The section of an item list file that defines an x-y plot must begin with the `.xya` command. This command informs the screen builder program that subsequent instructions are for the purpose of creating an x-y plot display. Also occupying the same command line are instructions that define the position, size, color, and options of the bounding box axes. The format of this sequence is as follows:

```
.xya  start_x  start_y  s_color  end_x  end_y  axis_options
```

The x and y coordinates determine the size of the bounding box. The static color option determines the color of the bounding box, the x-y axes (if present), and (if specified) the engineering unit designations and tick marks. The `s_color` item cannot be defined by a logic point. Selections for `Axis_options` parameters include:

L	Left axis tick marks labeled
R	Right axis tick marks labeled
T	Top axis tick marks labeled
B	Bottom axis tick marks labeled
N	Neither axis will be labeled
A	Axis within plot box will be drawn
U	Engineering Units will be printed on labeled axis

Table 4-2. Axis Options

This options list is similar to the one defined for the bounding box of the `.vba` item (see Section 4-8.13 of this manual). That is, the `L`, `R`, `N`, and `U` functions are available and may be implemented in the same manner. Supplemental options include `T`, `B`, and `A`. The first two additional selections permit the x-y axes of the bounding box to be set off by tick marks. The third selection of this group, `A`, instructs the program to overlay the x-y axes within the borders of the bounding box. The axes will be drawn relative to the scaling and tick marks assigned to the axes of the bounding box (the x-y axes may therefore be depicted in whole, or in part, or not at all). Figure 4-6 is a sample of a reactive capability curve display. It incorporates and illustrates the effects of implementing all of the options listed above (with the exception of the `N` option).

Both the x and y axis (top and bottom) have been scaled and labeled. The x axis measures Megawatts while the y measures Megavars. Since, in this instance, the scaling definition encompassed the axis origin (0, 0), the x-y axes are also shown.

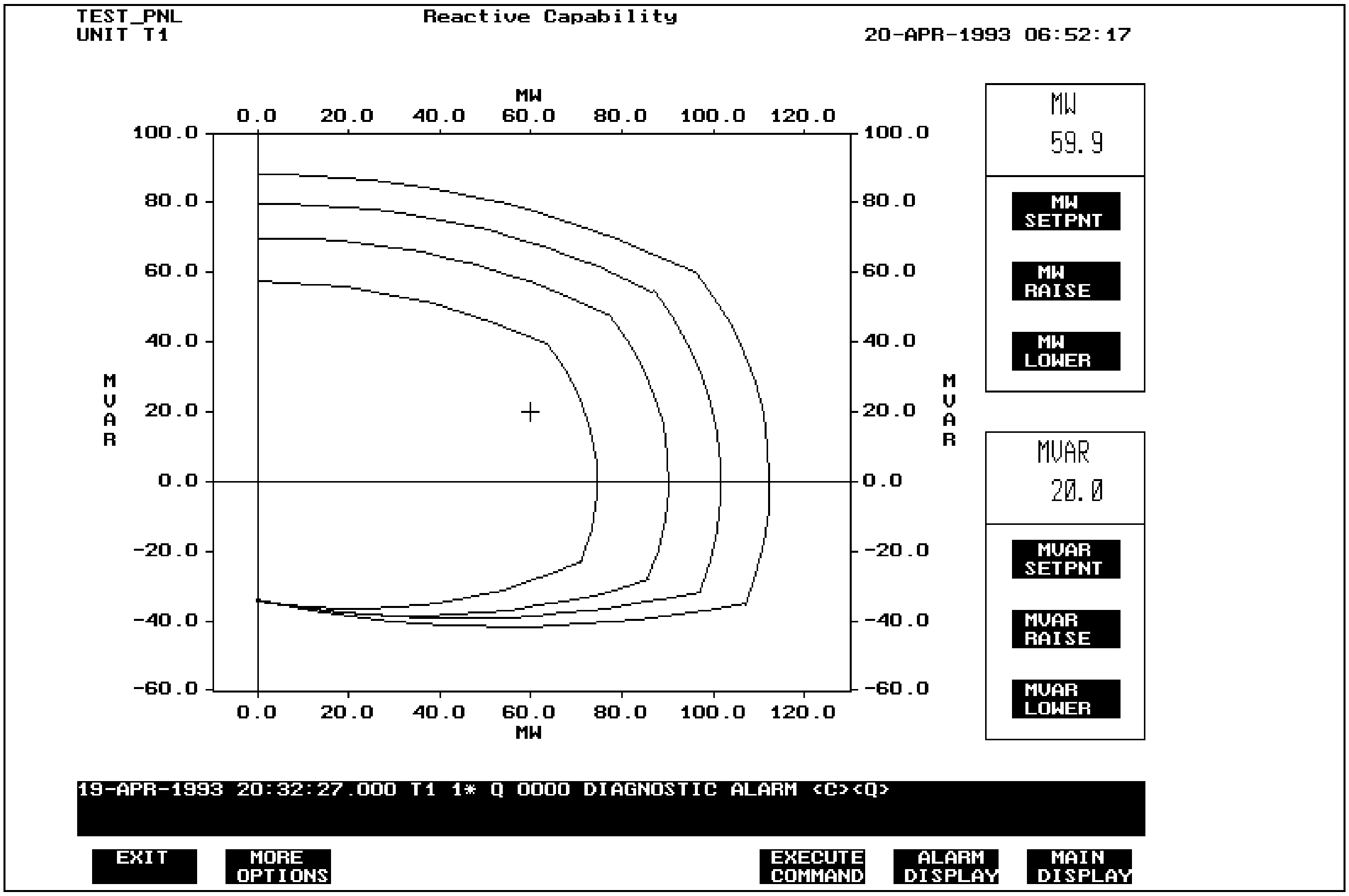


Figure 4-6. Animated Reactive Capability Display

CAUTION

The ratings curves are different for each application. They are shown as an example only and should not be used unless they match the generator ratings. Showing the wrong curves may cause the user to operate the unit in excess of ratings.

```
tx1     325     975     df      "Reactive Capability"
.xya    25      50      lg              +800    +850    BRLUTA
&sx     0       32768   0.0     512             ; Full range is 512 Megawatts
&sy     0       32768   0.0     512             ; Full range is 512 Megavars
&x      dwatt   -10     130     10
&y      dvar    -60     100     10
; The following lines define the static temperature curves. The numbers are in MW
; and MVAR.
;-----------------------------------------------------------------------------
;WARNING: These curves are different for each application.
;-----------------------------------------------------------------------------
;50 F Cold Gas                                  ; Label for first curve
&curve  mg      0       88      54      80      96      60      ; AB line segment coordinates
&curve  mg      96      60      112     14      107     -35     ; BC line segment coordinates
&curve  mg      107     -35     60      -41     0       -34     ; CD line segment coordinates
; 104 F Cold Gas                                        ; Label for second curve
&curve  mg      0       69.5    44      64      77      48      ; AB line segment coordinates
&curve  mg      77      48      88      22      86      -28     ; BC line segment coordinates
&curve  mg      86      -28     64      -35     0       -34     ; CD line segment coordinates
; 140 F Cold Gas                                        ; Label for third curve
&curve  mg      0       57.5    30      54      63.5    39.5    ; AB line segment coordinates
&curve  mg      63.5    39.5    74      10      71      -23     ; BC line segment coordinates
&curve  mg      71      -23     30      -36     0       -34     ; CD line segment coordinates
;The following defines the interpolate temperature curves
&icurve gr      cght    0       32768   0.0     2048.0
&...    50      0       88      54      80      96      60; AB line segment coordinates #1 (50øF)
&...    104     0       69.5    44      64      77      48; AB line segment coordinates #2 (104øF)
&...    140     0       57.5    30      54  63.5  39.5; AB line segment coordinates #3 (140øF)
&icurve gr      cght    0       32768   0.0     2048.0
&...    50      96      60      112     14      107  -35; BC line segment coordinates #1 (50øF)
&...    104     77      48      88      22      86   -28; BC line segment coordinates #2 (104øF)
&...    140     63.5    39.5    74      10      71   -23; BC line segment coordinates #3 (140øF)
&icurve gr      cght    0       32768   0.0     2048.0
&...    50      107     -35     60      -41     0    -34; CD line segment coordinates #1 (50øF)
&...    104     86      -28     64      -35     0    -34; CD line segment coordinates #2 (104øF)
&...    140     71      -23     30      -36     0    -34; CD line segment coordinates #3 (140øF)
&end
; The following defines the command targets to the left of the bounding box
box     850     900     df      999     -400
line    850     -120    df      999     +0
tx2     875     850     df      "  MW"
v2      +0      -50     df      dwatt
cmd1    +0      -90     8  2 #  dwatt   = 0 ...         1 "MW""SETPNT"
cmd1    +0      -90     8  2 !  L70R4R_CPB      = 4 ....        1 "MW""RAISE"
cmd1    +0      -90     8  2 !  L70R4L_CPB      = 4 ....        1 "MW""LOWER"
box     850     450     df      999     -400
line    850     450     df      999     +0
tx2     875     400     df      "MVAR"
v2      +0      -50     df      dvar
cmd1    +0      -90     8  2  # dvar    = 0  .... 1     "MVAR"  "SETPNT"
cmd1    +0      -90     8  2  ! L70R4R_CPB      =  4  .... 1   "MVAR""RAISE"
cmd1    +0      -90     8  2  ! L70R4L_CBP      =  4  .... 1   "MVAR""LOWER"
```

Figure 4-7. File List for Reactive Capability Display

4-8.18.1. `&sx` The `&sx` and `&sy` commands work in conjunction with the `axis_options` in that they permit customization of the scaling for the x-y axes. Line or curve definitions (`&_items`) that follow these instructions in the item list file will be scaled according to the values supplied on the `&sx and &sy` command lines. The format of these item commands is:

```
&sx   count_min   count_max   real_min real_max
&sy   count_min   count_max   real_min real_max
```

The `count_min` and `count_max`, and `real_min` and `real_max` values, respectively, define delimiters for raw (machine) count and engineering unit (for example Megawatts) scalings (see `SCELDATA.DAT` for CDB pointname scaling information). These two sets of values also define the scaling relationship between the two units of measure. If no `&sx` or `&sy` definitions are inserted in the `.xya` item, all units will be scaled in raw counts. As a general rule, only one set of `&sx` and `&sy` scaling commands should be defined per `.xya` item.

4-8.18.2. `&x, &y` These item commands define the CBD pointnames to be used in the x-y plot, the range of the scaled units ascribed to each axis, and the number of tick marks with which each axis will be labeled. The format of these item commands is as follows:

```
&x pointname   x_left   x_right  ticks
&y pointname   y_bottom y_top    ticks
```

If a range is not specified for a particular axis, the screen builder will automatically establish an axis range of zero to one; this fact will become obvious when the axis is set off by tick marks. In the sample display above, the CDB points DW (driven watts) and DVAR (driven VARS) are defined for the x and y axes respectively; this fact is denoted by the axis labelling (a consequence of implementing the `U` axis option). For the x axis (DW), the range is defined as (-10 → 130). For the y axis (DVAR), the range is (-60 → 100). The two ranges determine whether the origin point (0, 0) or x and/or y axes will be visible (again, assuming the `A` axis option is implemented). Also, they are responsible for specifying the orientation of the origin point relative to the borders of the bounding box. For example, the `&x` and `&y` axis parameters for Figure 4-6 stipulate that the x-y axis should be oriented (more or less) toward the bottom left corner of the bounding box:

```
&x dwatt -10   130   10
&y dvar  -60   100   10
```

This orientation can be changed so that the origin point is shifted toward the upper right-hand corner:

```
&x dwatt 130   -10   10
&y dvar  100   -60   10
```

A bounding box whose `axis_options` include any or all of the `L`, `R`, `T`, or `B` parameters will have one or more of its borders incremented (set off by tick marks). The number of tick marks displayed will reflect the value inserted as the `ticks` parameter at the end of the `&x` and/or `&y` command lines. The distance between tick marks will automatically be calculated by the screen builder program as a result of the scaling (`&sx`, `&sy`) and range (`&x`, `&y`) of the particular axis.
Be careful in applying tick marks to a particular axis. If too many ticks are inserted, the tick mark labels may overlap and become illegible.

4-8.18.3. `&line` This command permits static lines to be drawn on the display. Typically, these lines serve as an operating reference that can easily be recognized and compared to the current operating status. The current operating status of the unit can be displayed in terms of a cross-hair or a dynamic line or curve. The format of this item is outlined below:

```
&line s_color  x1 y1 x2  y2
```

The color definition of the line (`s_color`) must be static; it cannot be based on a logic variable. In addition, the endpoints (`x1, y1` and `x2, y2`) must lie within the range of the graph. If either of these points fall outside this range, the line will not be drawn. The scaling of the endpoints of the line will be in raw counts unless a previous `&sx, &sy` item command line has been specified. If the program encounters such a command, the endpoints of the line will be scaled accordingly.

4-8.18.4. `&curve` This item command permits static curves to be drawn on the display. Typically, these curves serve as an operating reference that can easily be recognized and compared to the current operating status. The format of this command is:

```
&curve   s_color  x1  y1   x2  y2   x3  y3
```

The color definition of the curve (`s_color`) must be static; it cannot be based on a logic plant variable. Since the curve is considered to be a portion of a circle, it is defined according to three plot coordinate points (`x1`, `y1`, `x2`, `y2`, `x3`, `y3`). Like the endpoints of the `&line` item, the three points that define the curve must lie within the range of the graph; if any of these points fall outside this range, the curve will not be drawn. The scaling of the three points that define the curve will be in raw counts unless a previous `&sx`, `&sy` item command line has been specified; if such a command is specified, the points of the curve will be scaled accordingly.

4-8.18.5. `&iline` The interpolated command item (`&iline`) defines a family of lines which, though not actually depicted on the display, are used by the screen builder program to construct a dynamic line that reflects the plant's current operating condition. The format of this command is:

```
&iline   s_color  pointname   count_min   count_max   real_min real_max
&...  value    x1  y1      x2  y2
&...  value    x1  y1      x2  y2
&...  value    x1  y1      x2  y2
```

The color of the dynamic line is defined by the `s_color` parameter. The `pointname` specifier defines for the program, the CDB point that is interrogated to form the interpolated line (only one point may be defined for a dynamic line). The `count_min`, `count_max`, `real_min`, and `real_max` parameters are optional; when used, they provide the ability to define the `&iline` item's interpolated CDB point in terms of engineering units instead of raw counts.

Following the `&iline` command line are a set of "`&...`" items. These define an interpolation value and the associated endpoints for that value in the plot box. The values in the interpolation data lines must be in ascending order. This is required for the interpolation calculation. If the current plant state is not inside the list of interpolation values, the line shown in the plot box will be extrapolated from the closest two values.

Style Tip. As standard practice, static and dynamic lines should be given distinct color definitions; this helps to avoid confusion in differentiating between the two line types.

4-8.18.6. `&icurve` The interpolated curve item command defines a family of curves which, though not actually depicted on the display are used by the screen builder program to construct a dynamic curve that reflects the plant's current operating condition. The format of this item command is as follows:

```
&icurve  s_color  pointname   count_min   count_max   real_min real_max
&... t1     x1 y1    x2 y2    x3 y3
&... t2     x1 y1    x2 y2    x3 y3
&... t3     x1 y1    x2 y2    x3 y3
```

Implementation of the `&icurve` item is similar to that of the `&iline` command. A static color definition (`s_color`) is specified for the dynamic curve as are minimum and maximum scaling parameters (`count_min`, `count_max`, `real_min`, `real_max`) (optional). The `pointname` specifier defines for the program, the CDB point that is interrogated to form the interpolated line (only one point may be defined for a dynamic line).

Following the `&icurve` command line are a set of `&...` items. These define an interpolation value and the associated x,y coordinates for that values curve. The values in the interpolation data lines must be in ascending order. This is required for the interpolation calculation. If the current plant state is not inside the list of interpolation values, the curve shown in the plot box will be extrapolated from the two closest values. The x,y coordinates used here for the RCC curves can be taken directly from the x,y coordinates given in the `&curve` portion of the file.

When defining the points used to interpolate a family of curves, the midpoints should be defined as closely as possible to the actual midpoint of the curve. This practice can prove to be beneficial if the current plant condition requires that the curve be extrapolated; the lines will extrapolate best if the midpoints of each curve in the family of curves fall on a common line, originating from the (0,0) point on the graph.

In some applications it is desirable to plot Megawatts on the vertical axis. To accomplish this, the data for the `&x` and `&y` lines needs to be interchanged, as well as each pair of x,y coordinates for the `&curve` and the `&icurve` command items. Finally, the text for the label of each axes must also be interchanged.

4-8.18.7. `&dot` The x-y plots show a cross hair at the current operating point. As plant conditions change, the cross hair moves to show the new operating point. This command leaves a dot on the screen at each position that the cross hair was drawn, effectively producing a dot trail. The cross hair is updated once per second, which means that the spacing between the dots will give an indication of how fast the plant conditions are changing. The format of this command (which should be added to the `.xya` item) is:

```
&dot s_color
```

The color is optional and must be static if provided. If no color is provided, the dots will appear dark gray. The dot trail starts when the display is started, and any action that erases and redraws the screen restarts the trail as well. A CONTROL/W, which repaints the screen, also restarts the trail.

4-8.18.8. `&end` This command indicates the end of a multi-line item. It is not optional. The item list file in Figure 4-7 details the commands used to construct the reactive capability display in Figure 4-6.. It may prove useful in understanding how such displays are created. This file should be considered an example only. It should not therefore be copied or used for an actual application.

4-8.19. cset

This item list file command is used to change the foreground and background colors of the static portions of certain display items. The format for this command is:

```
cset  fore_color.back_color
```

where:

`cset` is the command,
`fore_color` is the new foreground color to be used, and
`back_color` is the new default background color to be used.

The background color does not have to be specified if the default background color is not going to be changed. Some display items have static and dynamic fields. For example, items defined using the `hptn` command have two static fields (the CDB pointname and its engineering units) and a dynamic field (the pointname's value). The `cset` command will change only the foreground and background colors of the static portions of succeeding items in the item list file.

The following command in an item list file will cause the default foreground color of the static portions of all successive items to be displayed in red:

```
cset  rd
```

The `cset` command can be used more than once in the item list file of an animated display.

Style Tip. Do not change the default background color if screen images will be printed on a dot-matrix or laser printer; the chosen color may not print correctly on a black-and-white printer.

4-8.20. `incl`

This item list file command will instruct the animated screen builder to include another animated display into the current one. The animated screen builder will read the specified item list file and display its contents on the current display in the default foreground color. The format for this command is:

```
incl list_file
```

where:

`incl` is the command,
`list_file` is the name of the file ONLY of the animated display item list file to be included in the current display.

Style Tip. Objects drawn using groups of item commands can be used on more than one display by placing them in their own item list file(s) and using the `incl` command. Using this method, objects will be the same on each display and appear in the same location lending consistency to the displays. Typical uses would include target fields, unit layout drawings, and so on.

4-8.21. `pf`

This command instructs the animated screen builder to replay a picture file. This is similar to `incl`, so do not get confused between the two. The `incl` command includes another animated display, while `pf` uses only pictures. The format for this command is:

```
pf    start_x   start_y    fore_color    filename
```

where:

`pf` is the command,
`start_x` is the starting x-coordinate,
`start_y` is the starting y-coordinate,
`fore-color` is the foreground to be displayed in, and
`filename` is the picture file to be displayed.

Most picture files include initial color sets and absolute screen coordinates, so the x and y coordinates and the color are of little use. They are, however, necessary for this command, so do not leave them out.

4-8.22. `pgdn`

This command will cause the animated screen builder to include a `PAGE DOWN` display target at the bottom of the animated display below the Alarm Window. When the user clicks on the display target, the animated screen builder will switch to the specified animated display. The format for this command is:

```
pgdn  list_file
```

where:
`pgdn` is the command and
`list_file` is the name of the file of the animated display item list file to be switched to.
The filename specified in the command must not include the filename extension nor the path to the file.

4-8.23. `pgup`

Style Tip. The `pgdn` command is another method of nesting animated displays.

This command will cause the animated screen builder to include a `PAGE UP` display target at the bottom of the animated display below the Alarm Window. When the user clicks on the display target, the animated screen builder will switch to the specified animated display. The format for this command is:

```
pgup  list_file
```

where:
`pgup` is the command,
`list_file` is the name of the file of the animated display item list file to be switched to.
The filename specified in the command must not include the filename extension nor the path to the file.

Style Tip. The `pgup` command is another method of nesting animated displays.

4-8.24. Multi-Unit Animated Displays

On sites with multiple units, an animated display is capable of displaying CDB pointnames, their values and engineering units for units other than the currently selected unit. Command targets for units other than the currently selected unit can also be shown on animated displays. It is very important to understand that the animated screen builder does not automatically identify points or targets on a display for units other than the currently selected unit. It is the responsibility of the creator or editor of the item list file to provide some indication of the association between multiple unit points or targets and their respective units. The viewer or user of an animated display that has multi-unit points or targets must be made aware of the fact that issued commands via the display may affect more than one unit.

CAUTION

Identify CDB points or command targets on an animated display for units other than the currently selected unit. This ensures that the user is aware of which unit the point or command target is associated with.

On a multi-unit <I>, two methods exist for changing the animated display to the next unit. The first involves exiting to the Main Menu and clicking on the unit number field until the desired unit number is the currently selected unit. (From the Main Menu, the currently selected unit can also be changed by pressing F9 until the desired unit number is the currently selected unit.) The second entails adding a hot spot to the animated display in the area of the unit name/currently selected unit fields. The hot spot should be located at coordinates (000, 950); its should be 200 units wide (x) and 50 units tall (y). Clicking on this target will execute a call to the item list file that has been defined for the hot spot. This call can execute the retrieval of any valid unit list file if the appropriate unit identifier is specified. This identifier is a two-character unit name followed by a colon ":" (for example `T2:sample` would be used to change to the `sample.a2` item list file for unit T2).

Style Tip. Use the multi-unit animated display feature to change to animated displays which are similar in appearance and function to the one being changed from. For example, a Unit Control animated display for each unit on a multi-unit <I> could be created by copying the original file (for example, UNIT.A1) to files with the same name and the proper unit-specific extension (such as UNIT.A2 and UNIT.A3 on an <I> controlling three units). By adding a hot spot to each display to change to the next unit in sequence (from unit T1 to unit T2, from unit T2 to unit T3, and from unit T3 to unit T1 in the above example), the animated display will change units sequentially just like when changing units from the Main Menu (or from User-Defined Displays).

4-8.25. Animated Display Style Guide

This section provides creators of animated displays with some ideas and tips to add consistency and continuity of style to animated displays. It is grouped into the following sections:

- General Tips and Hints
- Color
- Targets
- Hot Spots
- Display Titles
- Coordinates
- Multi-Unit Displays
- Bargraphs

4-8.25.1. GENERAL TIPS AND HINTS. Be consistent with the use of colors, the placement of targets, hot spots, objects, and so on. This will aid greatly in the recognition and familiarization of displays by users and will help convey information quickly to the viewer. A general rule of thumb is that data points are displayed in the lower left quadrant of the display, command targets in the lower right quadrant of the display, and graphics and/or bargraphs in the upper half of the display.

- Advanced users can add relay ladder diagram sequencing to the CSP(s) of unit(s) (preferably in <C>'s CSP) to create logic points that toggle based on time or some sequence of events. These logic points can be used to toggle the foreground and/or background colors of display items, bars in bargraphs, and so on, to bring attention to conditions, simulate motion, and so on. (This is **not** recommended for those with little or no CSP programming experience or background.)

- Do not place items too high in displays, particularly in the upper left- and right corners where they may overlap and/or obscure the sitename & unit number.

- Use comments generously in item list files; they will aid those creating and reading the file. Comment lines in an item list file must begin with a semicolon and are ignored by the animated screen builder. Comments may be added to the ends of lines by placing a space and a semicolon at the end of the item command or command. The animated screen builder will ignore any text on an item command line after a semi-colon.

- Do not let any objects, especially targets, touch the alarm window; they may accidentally SILENCE the alarm.

- Test print each animated display if the <I> is equipped with either a dot matrix or laser printer. Not all colors will print as displayed or as specified in item list files.

- Creators of animated display item list files who have a favorite ASCII text editor installed on the <I>'s hard disk can access that editor by creating a batch file in the `C:\CUSTOM` directory with the filename `GEEDIT.BAT` to run their editor. A batch file of the same name exists in the `G:\EXEC` directory which will execute `C:\CUSTOM\GEEDIT.BAT` if found when the `EDIT DISPLAY` target is clicked on from an animated display. Examine the batch file `G:\EXEC\GEEDIT.BAT` for more information.

4-8.25.2. COLOR. Do not draw a solid box that fills the entire display in the color blue using the sbox command; it is the same color used as the background of the Alarm Window. A blue background in an animated display will be indistinguishable from the Alarm Window. Use the logic color options to toggle the colors of bargraphs, text strings, and other items. For example, use the alarm or trip logic signals for temperatures or pressures to toggle bargraph colors to red to alert the user to the condition. The logic color option can also be used to hide items by specifying the default background color to hide the item; when the logic value changes, the item will appear on the animated display.

Do not use light red to display items; the animated screen builder uses light red to show that an undefined CDB signal pointname has been specified. If using a dot matrix or laser printer, print all animated displays before adding them to the Main Menu or including them in hot spots of other animated displays to ensure all important items of the display will print correctly. Color on color text will not print on a black and white printer because color on color shows up as black on black. Therefore, text should be a default background on any color, or any color on a default background. Any item specified to display as white will not print (it will be printed in white); use default foreground when specifying colors to be displayed as white so that they will print properly. Default background (black on the screen) prints as white on a dot matrix or laser printer. Black prints as black. Also, do not use Light Cyan (LC); it prints as yellow on a color printer (HP Paintjet). Test all displays by printing them to be sure the colors specified will print as desired.

Use the following colors cautiously in solid boxes and as background colors as they are used as the background colors of command and display targets: green, red, light grey, and white. Additionally, cyan (light blue) and magenta (purple) are used to show that targets have been armed and commands have been sent to the Mark V Control Panel, respectively. Also remember that blue is used as the background color of the Alarm Window.

4-8.25.3. TARGETS. Use uppercase text for all command targets (that is the case used for command target text in User-Defined Displays). Use the same text for every usage of a particular command target in every animated display. For example, a command target to increase the speed/load of a unit should always have the text `RAISE SPEED/LOAD` each time the target is used. Do not use `RAISE` on one display and `INCREASE` or `INC` on another. Try to use the same text as on similar targets on User-defined Displays to avoid confusion.

Use a space as the first character in each command target text string; without a leading space the first character of the text string will overlap the left border of the target. Try to put command targets in a consistent location on each animated display, such as the lower right quadrant or along the bottom of the available screen area (above the Alarm Window) or along the right side of the display (similar to User-Defined Display command targets). Test ALL targets and their feedback signals to be certain they work as desired before adding an animated display to the Main Menu or specifying one in a hot spot.

4-8.25.4. HOT SPOTS. Do not hide hot spots. The preferred method is a blue diamond or a magenta or blue solid box at the location of a hot spot to show its presence. Whatever method is chosen, use it consistently on all displays which include hot spots. (Remember, items can occupy the same area on animated displays.) Do not use green, red, light grey, or white as backgrounds for hot spots as these colors are used as backgrounds for command and display targets.

Place hot spots in consistent locations on displays; for example, in a bar at the top of the displays, along the left side of the displays, or along the bottom of the displays (above the Alarm Window). Do not overlap hot spots. The first hot spot specified in the item list file will take precedence over any other that overlaps the same area.

4-8.25.5. DISPLAY TITLES. Display titles are defined as the text that is located at the top center portion of the animated display. For the sake of consistency, certain rules should apply. First, use all capital letters in the title. Use the single high font (the `tx1` command item). The title's color should be the default foreground color and no background color should be used.

The line of the title should be located on the Y coordinate 975. To center the title, calculate the X coordinate using the following equation:

$$X = \{500 - (6.25 * No\_of\_characters\_in\_title)\}$$

The second line of the title should be used for spillovers for long titles and for the display page number as well. It should be located at the Y coordinate 950 and its X coordinate should be calculated using the equation above.

4-8.25.6. COORDINATES. Group objects created with individual items (such as a turbine or compressor or combustion can) by using absolute coordinates for the origin of the first item and relative coordinates for the origins of each successive related item. In this manner, objects can easily be moved by changing only the first item's origin (the others will follow). Text strings or points/values can easily be added to objects and will follow the object when moved if relative coordinates have been used without affecting the positions of unrelated items or objects.

Use Piping Schematic Diagrams when creating animated displays for unit systems/functions such as Steam Injection, Fuel Mixture, Dry Low NOx, Hydraulic Oil System, and so on.

Make sure all lines that are supposed to meet actually meet and do not extend past their intended stopping places. Occasionally, it may be necessary to adjust the length of a line by one or two units to make it actually meet with another or not extend past another. To improve readability of the item list file, specify all coordinates using 4 characters, one for the relative delimiter (" + " or " - ") three for the digits of the value, even if the first characters are blanks or zeros (for example, " 000", or "+975").

Some <I>s may have grid files in either the `F:\RUNTIME` or unit-specific directory which can be included into a display using the incl command. These files will draw grids in a display which can be used to help in determining coordinates for items. Place the incl command followed by the grid item list file name at or near the beginning of the item list file being worked on. When they have served their purpose, they can be commented out of the item list file by placing a semicolon at the beginning of the incl line (or the line can be deleted). By including one of the grid files described above as the only line in a new item list file and printing the display, the user can have a blank grid to sketch items on and get some ideas about placement, sizing, coordinates, and so on. If no grid files like the ones described above exist on an <I>, they can be created. The grid files described above are just item list files with a series of line item commands to draw lines on the display horizontally and vertically at equally spaced intervals.

A four line sample of item commands to draw lines at 100 unit intervals is presented as follows:

```
line   000   000  df  +000  +999
line  +100  +000  df  +000  +999
line   000   000  df  +999  +000
line  +000  +100  df  +999  +000
```

`Lines` of different colors can be specified at periodic intervals to help with recognizing grid unit multiples (for example, red for increments of 10 units, cyan for increments of 50 units, and so on.)

4-8.25.7. MULTI-UNIT ANIMATED DISPLAYS. <I>s which are capable of controlling multiple units can have a single animated display showing information from all units. Such a display is sometimes called a plant control display. A plant control display's item list file must reside in `F:\RUNTIME` and have a filename extension of `".A0"`. This file **must** be created from the DOS command line using an ASCII text editor. CDB signal pointnames must be made unit-specific by preceding them with the unit's two-character name and a colon. For example, percent of high-pressure turbine shaft speed from unit T2 would be defined in an item command as `T2:TNH`. TNH would be displayed only as TNH, with no indication of which unit's speed is being displayed.

Command targets will also not be displayed with any unit-specific identifier. For this reason, it is extremely important to make it very clear which unit the points/targets are associated with. Do this by blocking areas of the display using boxes for individual units and identifying them with text strings. Make absolutely certain it is very **clear** which unit is to be receiving commands if command targets for multiple units appear on a single animated display.

Place a hot spot at x-coordinate 000 and y-coordinate 950 that is 200 units long in the x direction and 50 units high in the y direction to change to the similar animated display for the next unit in sequence (for example from unit T1 to unit T2, from unit T2 to unit T3, and from unit T3 to unit T1 on a three-unit multi-unit <I>). By going to the next unit in sequence, the animated displays will function like the User-Defined Displays and the Main Menu when changing to the next unit by clicking on the current selected unit field including the F9 hot key.

4-8.25.8. BARGRAPHS. Bargraphs can be used to display information in an easy-to-understand format. Draw lines on bargraphs (using the line item command) to show levels of interest, such as various speed levels, or high wheelspace or bearing drain temperatures, and so on. Do not mix bars (CDB signal points) with different scale types on a single bargraph. Use points with identical scale types on the same bargraph, such as `PCT`, `TC`, `PSI`, and so on.

If a vertical bargraph is not specified to be wide enough, it will not automatically add tick marks (divisions) or engineering units, if specified. In such cases where there is not enough room to widen the bargraph, it may be necessary to add text strings and/or tick marks using item commands. If a vertical bargraph with a single bar is specified with the options "`lru`", it must be at least 225 units wide (in the x direction) in order for it to display the bar, labels and tick marks on both axes. Do not specify too many tick marks on a bargraph. They will overlap each other and may at times seem to disappear.

If the point status of a bargraph is known, green should be used for the normal state, yellow for the alarm state, and red for the trip state (using the four color logic option). When the point status of a bargraph is not known, then either light grey, light blue, or cyan should be used to signify that, preferably using cyan. A sensor fault should be indicated by using white text on a red background. Use a `Page Down` (`pgdn` command) or a hot spot to show data value, alarm level, and trip level if data does not fit on the current display.

4-9. CONFIGURING THE BACKUP OPERATOR INTERFACE

The BOI_Q.SRC source file consists of a series of tables that define the <BOI> displays that can be configured (Menu, All Points, Operator Demand, Normal). These tables or tabled "SECTIONS" are defined by header (DEMAND, DISPLAY, POINT-TAG, ENUMDATA, SCALE) and footer (END DEMAND, END DISPLAY, END POINT, END ENUM, END SCALE) command statements. These section commands are not case sensitive, however, each must begin in the first column and occupy its own command line.

Although the ordering of sections in the source file is not consequential, the manner in which they are assembled is significant. Changes to individual sections must conform to the original file format. Columns within sections may not be reordered. Entries in each table cannot begin in the first column (this is how beginning and end SECTION commands are recognized). Comment lines must begin with a semicolon. Normal display POINT_TAGs that are not used must be defined by the character string BLANK_DSPLY. Finally, all columns must be separated by at least a single blank space.

NOTE

Each of the sections described below includes a generic example to facilitate understanding. The text incorporated within these examples represents documentation that is inserted within each factory prepared BOI_Q.SRC file.

4-9.1. Demand Section

The DEMAND SECTION of the BOI_Q.SRC file can be used to configure the types of displays (Normal, Enumerated Switch) that are presented on the <BOI> when the DEMAND display (Shift + Help) key is depressed. Also, this segment can be configured to define the functionality of each display type. Building DEMAND displays from the <BOI> causes this section of the (downloaded) BOI_Q.SRC file to be overwritten. Conversely, re-downloading the <BOI> source file causes its DEMAND Section instructions to be re-initiated.

```
;--------------------------------------------------------------
;
;  DEMAND SECTION
;
;  This section defines the demand displays (two) that the customer is allowed
;  to change from the panel.
;  "N" (Normal) type displays are normal displays of up to 4 points,
;  "E" (Enumerated) type displays are enumerated point displays.
;
;  LIMIT;  There must be exactly 2 displays shown here.
;
DEMAND
;   E   DISPLAY NAME      SIGNAL_TAG     MASK_TAG       ENUM-TAG          COMMAND_TAG
;or
;   N   DISPLAY NAME      POINT_TAG      POINT_TAG      POINT_TAG       POINT_TAG
;   -   ----------------- ----------     ----------     ---------       --------------
    N   "DEMAND, PAGE 1 " BLANK DISPLAY  BLANK DISPLAY  BLANK DISPLAY   BLANK DISPLAY
    N   "DEMAND, PAGE 2 " BLANK DISPLAY  BLANK DISPLAY  BLANK DISPLAY   BLANK DISPLAY

END DEMAND
```

Figure 4-8. Demand Section of BOI_Q.SRC File

NOTE

Command lines that incorporate quotes (`"DEMAND, PAGE 1 "`) will be read verbatim when the file is executed. Case sensitivity thereby becomes consequential when quotes are used. Other than this exception, case sensitivity does not affect the `BOI_Q.SRC` file.

Notice that other than the `DEMAND` and `END DEMAND` statements, there are only two command lines in this section. These commands define the type of displays to be implemented (E, N), their names as they will appear on the <BOI> screen (""), and their corresponding functions (by means of `POINT_TAG` and/or `SIGNAL_TAG`, `MASK_TAG`, `ENUM_TAG`, `COMMAND_TAG` selection). For Normal displays, specific signals are retrieved according to pointers or `POINT_TAGs` and their related signal names. These relationships (as well as the scaling of each signal) are established in the `POINT_TAG` Section of the source file. As an example, a `DEMAND` Section that produces a turbine speed setpoint display would be identical to the file above except for the following text which would replace the first command line:

```
N     "FSR MAN CTRL" FSRMAN_CMD  L43FSRS  FSR  FSR_CONTROL
```

The resulting <BOI> screen will display the `POINT_TAG` text strings and the integer or logic signal variables connected with the specified `POINT_TAG`s (this assumes that all of the above `POINT_TAG`s are pointing to valid signal names, which in turn, have valid scale codes assigned to them (see `POINT_TAG` Section). The quoted character string will appear only in the menu of the help display. It will not actually be shown on the data display itself.

To depict and execute modes of operation, Enumerated Switch displays must include four definitions: `SIGNAL_TAG`, `MASK_TAG`, `ENUM_TAG`, `COMMAND_TAG`. These definitions or pointers reference <BOI> signal names (`SIGNAL_TAG`, `COMMAND_TAG`), permissive bit masks (`MASK_TAG`), and enumerated text strings (`ENUM_TAG`). The `SIGNAL_TAG` and `COMMAND_TAG` signal definitions allow the <BOI> to compare the current state of a feedback signal (`SIGNAL_TAG`) with the desired command signal (`COMMAND_TAG`). If the two do not agree, the command signal is sent to correct the mismatch. To insure that only legitimate options are offered, a bit mask is applied to each Enumerated Switch Display. These options are correlated to relevant display text strings by the `ENUM_TAG` value (see `ENUM_TAG` Section).

4-9.2. `Display` Section

```
;------------------------------------------------------------
;
;  DISPLAY SECTION
;
;  This section defines each display in terms of POINT_TAGS.
;  "N" (Normal) type displays ar normal displays of up to 4 points.
;  "E" (Enumerated) type displays are enumerated point displays.
;
;  LIMIT: 32 displays.
;
DISPLAY
;   E  DISPLAY NAME      SIGNAL_TAG     MASK TAG       ENUM_TAG        COMMAND_TAG
;or
;   N  DISPLAY NAME      POINT_TAG      POINT_TAG      POINT_TAG       POINT_TAG
;   -  -------------     ----------     ----------     ----------      -----------
    E  "F1 CTRL LCT SEL" C_LOCAT        C_LOCAT_MSK    Etag_01         C_LOCAT_CMD
    N  "F2 RESET SELECT" L43REWS_CPD    L86HR1_CPD     L4T             BLANK_DSPLY
    N  "F3 SPD SETPNT  " TNHR_CMD       TNHR_RCMD      TNHR            TNH
```

Figure 4-9. Display Section of BOI_Q.SRC File

Similar in structure to the DEMAND Section, the DISPLAY Section of the source file also designates Display Types and their function. It is however, the location for defining all of the user-defined (non-DEMAND) displays for the <BOI>. The sequential order of this segment indexes the displays as they appear under the Menu Display and the function keys (F1 - F20) that are used to implement them. If a display is added or removed from the DISPLAY Section list, the Menu Display is similarly affected including the function key assignments for each display.

As every display is uniquely labelled with a quoted ("") function key designation, care should be taken to update these appointments when a change in the order of the displays is made -- otherwise, Menu displays will appear with incorrect function key labels. The maximum allowable number of displays for this section is thirty two (32).

4-9.3. Normal Display

```
NORMAL   10 ; This defines which of the above displays is the NORMAL display.
           This is an index into the above list, 1 means the first display.
```

Typically defined immediately after the DISPLAY Section, the NORMAL Display represents a single selection from the DISPLAY Section's list. The number selected (1 - 32) will define the display that is exhibited when the NORMAL display key is depressed.

4-9.4. `Point_Tag` Section

```
; POINT_TAG SECTION
;
;This section defines for each point tag a point definition. Each definition consists of:
;
; POINT TAG:   The point tag used in the display section.
; SIGNAL NAME: The Mark V signal name, must exist in UNITDATA.DAT.
; DISPLAY NAME:       The name (6 chars) that is shown on the <BOI>.
; E/N FLAG:           "N" for normal points, "E" for enumerated points.
; SCALE/ETAG:  "N" points have the scale name or scale number here. If a
;                     scale name is used, it must exist in ENGLISH.SCA. The scale
;                     name or number must resolve to a scale code from 0 to 247.
;                     Scale codes 248-255 are user programmable via the SCALE_DATA
;                     section below.
;                     "E" points have the enumerated tag here.
;
; Since this section is used for the "ALL POINTS" display, it is customary to
; sort this section upon the DISPLAY field so that the ALL POINTS display
; presents the points in sorted order. The exception - BLANK_DSPLY must be
; the first point in the list.
;
; LIMIT: 128 points in this list. (BLANK_DSPLY plus 127 others.)
;
POINT
;      POINT_TAG      POINTNAME             DISPLAY N/E    N:SCALE or E:NUM_TAG
;      -----------    ------------   --------       ---    ---------------------
       BLANK_DSPLY    BLANK_DSPLY           "....." N      NULL
       AFLR_CMD       AFLR_CMD              "AFLR_C"        N      PCT
       DVAR           DVAR                  "DVAR  "        N      MVARS
       DWATT          DWATT                 "DVAR  "        N      MWATT
       DPF            DPF                   "PF    "        N      PF


END POINT
```

Figure 4-10. Point Tag Section of BOI_Q.SRC File

Consisting of five columns, the `POINT_TAG` Section of the source file designates point definitions for each `POINT_TAG`. These definitions link `POINT_TAG` variables to Mark V database signals (`POINTNAME` must be found in `UNITDATA.DAT)`, assign `SCALE` codes (Normal displays) or `ENUM_TAGs` (Enumerated Switch) to each signal point, and appoint display text strings (`DISPLAY`) that denote accessed signals when they appear on the <BOI>. Comparable with the `DISPLAY` Section of the source file, items within the `POINT_TAG` Section table must be identified as either Normal (N) or Enumerated Switch (E) types. A Normal Display (as defined by the `DISPLAY` Section) that incorporates an Enumerated Switch signal (`POINTNAME`) will only depict the signal's text string (`DISPLAY`); Normal display points may not be introduced as part of an Enumerated Switch display.

Although the `POINT_TAG` table is limited to a maximum of 128 individual signal point entries, a degree of flexibility is maintained in that multiple `POINT_TAG`'s can be assigned to a single `POINTNAME`. As a duplicate of the `POINT-TAG` Section, the `ALL POINTS DISPLAY` can be configured in the same manner. As noted in the source file instructions above, `BLANK_DSPLY` must be the first point in the table. This constraint permits the use of this text string to define empty `POINT_TAG`s in the `DEMAND` and `DISPLAY` Sections of the file.

4-9.5. `Enumdata` Section

```
;------------------------------------------------------------
;
;  ENUMDATA SECTION
;
;  This section defines the strings associated with each ENUM_TAG.
;  The strings are limited to 8 characters each. The list must end with the
;  special END line.
;
;  LIMIT: The total number of bytes used for all ENUM_TAGs must be <= 512 bytes.
;         There can be no more than 32 enumerated sets defined.
;
;
;
ENUM
;
      ETAG_00
            "OFF"
            "MANUAL"
            "AUTO"
;
;
      ETAG_01
            "LOCAL"
            "REMOTE"
            "CABLE"
;
;
      ETAG_02
            "SHUTDOWN"
            "MANUAL"
            "TEMP"
            "STARTUP"
            "ACCELR"
            "DROOP"
            "ISOCH"

END ENUM
```

Figure 4-11. Enumdata Section of BOI_Q.SRC File

Responsible for defining the strings associated with each `ENUM_TAG`, the ENUM data Section must not exceed 512 bytes in size. The number of enumerated sets contained in the table is contingent on this limit factor (remember that each text string has an added terminating byte). Each enumerated set must begin with an ETAG number definition (`ETAG_n`) that occupies its own command line. For reasons of display, quoted strings remain case sensitive; as well, each string must not exceed eight characters (nine bytes total, eight characters and one terminating byte) in length.

4-9.6. `Scale Data` Section

```
;  SCALE DATA SECTION
;
;  This section defines the scale codes for the 8 user programmable scale codes
;  that the unit supports. These scale codes are used when the above POINT_TAG
;  section defined a point as having a scale code of 248 to 255. (These must
;  have been defined by number, not by name, in the POINT_TAG section.)
;
;  If not all scale sets are defined for a given scale code, the un_specified
;  scale sets will be displayed in the lowest number scale set that was
;  specified.
;
;  FORMAT IS:
;
;  code  set  gain  shift  offset resol  decimal units
;
;  where;
;      CODE:         Scale code being defined (248..255)
;      SET:          Scale set being defined:
;                           0 = Hardware
;                           1 = English
;                           2 = Metric
;                           3 = Custom
;      GAIN:         Gain in decimal counts.
;      OFFSET:       Offset in decimal counts.
;      SHIFT: Shift in decimal. (Can be negative.)
;      RESOLUTION:   The number of counts to change for each RAISE/LOWER command.
;      DECIMALS:     The number of places after the decimal point.
;      UNITS: The engineering units string (5 chars).
;
;      LIMIT: 8 scale codes (248..255) with up to 4 scale sets (0..3) per code.
;
SCALE
;
;             CODE   SET    GAIN   SHIFT  OFFSET RESOL        DEC    UNITS
;             ----   ----   ----- -----   ------ ------ ----   ------
               248    0     16384    1      0           1            1    "cnt15"
               249    0     16383    1      0           1            1    "cnt15"
               249    1     20480    0      0           1            1    "def F"
               249    2     11380    0     -18          2            1    "deg C"

END SCALE
```

Figure 4-12. Scale Data Section of BOI_Q.SRC File

The <BOI> uses the first 248 (0-247) scale codes of any of the *.sca files. The Mark V determines which scaling file to use for the <BOI> via the control constant BOI_SCAL_SET in CONST_Q.SRC. Use one of the the following control constant values to determine the scale code set.

0 Hardware scaling as defined in HARDWARE.SCA
1 English scaling as defined in ENGLISH.SCA
2 Metric scaling as defined in METRIC.SCA
3 Custom scaling as defined in CUSTOM.SCA

As a generic example, the samplein figure 4-12 illustrates the way in which the `SCALE` Section may be used to implement user-programmable scale codes. Although only two codes are used here (248, 249), up to eight can be supported (248-255). Each of these codes may have as many as four (0-3) different scale sets that define units of measure. Three of these sets are pre-defined (Hardware/raw count, English, Metric) while one is user-definable (Custom). (This latter set allows practically any unit of measure to be used). The `GAIN` values of this segment define the allowable range of each set. From the example, ranges for Hardware (raw count), English, and Metric units of measure are defined as 0-16384, 0-20480, and 0-11380 respectively. In addition to a `GAIN` constant, each code `SET` must have `SHIFT` (negative values allowed), `OFFSET` (negative values allowed), `RESOL` (resolution), `DEC` (decimal), and `UNITS` specifiers. Failure to include these definitions will result in errors when the source file is compiled.

4-9.8. Compiling BOI_Q.SRC

Changes to the BOI_Q.SRC file may be implemented only after the file has been saved, compiled, and downloaded. Saving the file can be accomplished with the text editor used to make changes to the file. Compiling the file requires the use of the <I>'s table compiler. This can be accomplished from the UNITn directory by typing the DOS command TABLE_C BOI or TABLE_C ALL. Downloading the compiled file to EEPROM can then be executed by the DOS command EEPROM DOWN [UNIT], [CORE] (= R, S, T) BOI. The <BOI> can be configured to receive data from any one of the three cores <R>, <S>, <T>. For this reason, it is important to download updated <BOI> source files to all three cores in TMR panels.

CHAPTER 5

EDITING THE CONTROL SEQUENCE PROGRAM

5-1. INTRODUCTION

The Control Sequence Editor (the editor) is an <I> utility that allows fixed format graphics editing of Mark V control sequencing software. The source files created by the editor are then compiled by the Control Sequence Compiler, and downloaded to the control panel using the EEPROM download program.

The editor operates on a basic unit of software known as a segment. A Mark V Controller (<C> or <Q>) can have a maximum of eight segments using up to 64K of controller memory. A single <Q> image is utilized in all three control processors of a Triple Modular Redundant (TMR) panel. Each segment can be assigned a unique scan rate and execution offset during the compilation process.

A segment is made up of a series of sequentially executed rungs. A rung is the basic unit that the editor operates upon. Before discussing the details of editing a rung, the user should be familiar with the four different types of rungs that the editor allows, as the basic editing operations on the four types of rungs are slightly different.

The four types of rungs that can be present in a segment are:

- Pure relay ladder diagram rungs (RLD rungs).
- Rungs that combine RLD operations with a primitive functional block call (PRIMITIVE rungs).
- Rungs that have no RLD operations, but call a Big Block with extensive functionality (BBL rungs).
- Rungs that contain a textual comment describing the operation of the segment (COMMENT rungs).

The default rung type for a new rung is RLD. Several targets in the element type menu can be used to modify the type of rung being edited. Each of these targets will be described in succession in the corresponding sections below. Once a rung has been modified from an RLD rung to one of the other types, there is no way to return to an RLD rung except to delete the rung and add a new, empty one.

Although there is a limited amount of relay ladder flexibility in the primitive type rung, it is not possible to mix these four rung types within a rung.

5-2. FILE STRUCTURE

The editor uses several unit specific system level files that contain signal name database definitions, and definitions of the software building blocks available to the user. Several types of files for maintaining the current contents of controller segments are also used. These files are summarized in Figure 5-1. `PRIMITIV.DEF` and `BIGBLOCK.DEF` files are ASCII system files that detail the interface to all of the high level programming blocks available in EPROM on a particular Mark V Turbine controller.

CAUTION

`BIGBLOCK.DEF` and `PRIMITIVE.DEF` files are unique to each unit, and should not be copied or interchanged from unit to unit.

Modification of these files is not necessary, and not recommended, as their content correlates directly to the contents of the controller EPROM. If a new block is added to a system, or an existing block is modified, the new files will be provided from the factory along with the new controller EPROMs.

The remaining files in Figure 5-1 are those that are created by the editor to support each individual segment as it is being manipulated. The <segment>.SRC files are the source files that the editor loads and writes when it is saved. When a source file is saved, the previous copy is renamed to <segment>.BAK to allow for recovery from serious editing errors.

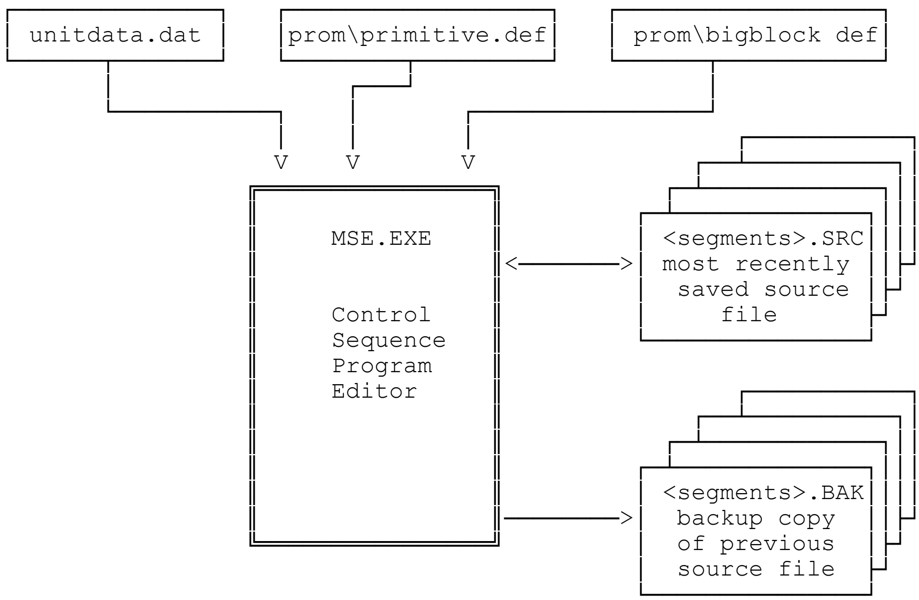


Figure 5-1. The Control Sequence Editor File Structure

5-3. STARTING THE EDITOR

The editor can be invoked from the <I> system Main Menu or from the DOS command line. It should be run while in the unit-specific directory for a particular unit. Start the editor by making the proper menu pick or by issuing the command `MSE` at the DOS prompt. It is not necessary to give a filename at this point, however, the user may pass the name of the file that requires editing. If a filename is not supplied at the command line, the editor prompts for a source filename.

If available, the editor loads the signal name database into memory, initializes the graphics system, and then prompts for a source file name. It is not necessary to include the .SRC extension on the file name. If no name is entered (carriage return) a default name is used `(M5MSESEG.SRC),` otherwise, the specified source file is loaded (if it exists). When retrieved, the first rung of the segment is displayed.

5-4. EDITOR WINDOWS

The editor display consists of five major windows (see Figure 5-2). Several minor windows (not shown) display auxiliary display information. The windows break down the actions necessary to edit a segment into logical groupings designed to guide the user through the process. Each window is explained in detail below.

5-4.1. Rung Display Window

The largest window (Window 1 in Figure 5-2) is the "rung display window." This window displays the contents of the rung currently being manipulated. This window also permits changes to be made to the current rung. Only one rung can be displayed at a time. Provisions for moving among the rungs that make up a segment are provided in the "action window" (see Section 5-4.2). The process of actually editing rungs is described in Section 5-6.

5-4.2. Action Window

The action window (Window 2) is multi-functional. It controls loading, saving, and quitting operations as well as movement among the segment rungs.

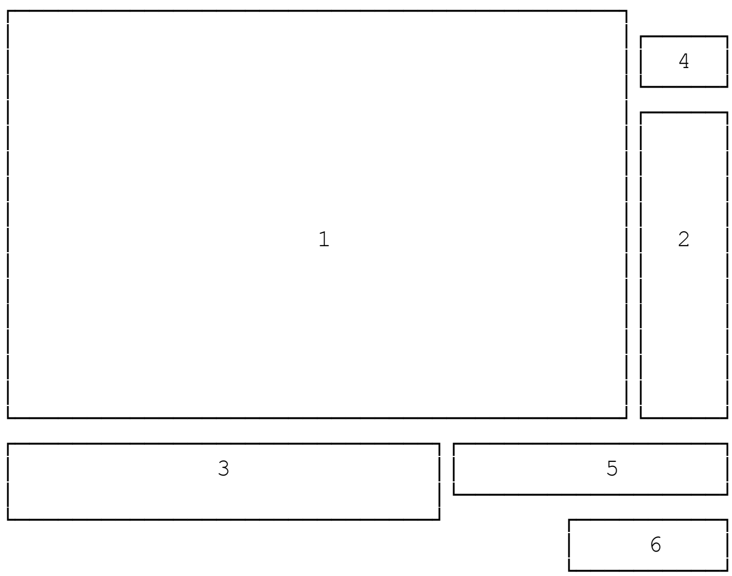


Figure 5-2. Control Sequence Editor Window Structure

5-4.3. Element Type Window

The "element type window" (Window 3) controls the type of component that will be placed when the mouse button is clicked for RLD or primitive rung editing purposes. If a primitive, big block, or comment is needed, this window provides the capability to modify the rung type through the insertion of these elements.

5-4.4. Signal Name Window

The "signal name window" (Window 4) displays the signal name currently entered into the editor. This signal name will be used when RLD components are placed or when an argument is identified on a primitive or big block. The current signal name remains in the window and can be used in editing operations until a new signal name is entered. The window can be cleared by striking the ESC key.

5-4.5. Message Window

When the editor needs to prompt the user for information or display a warning or error message, it uses the "message window" (Window 5). The data entered by the user (after being prompted) will be echoed in the message window. When the data in the message window is no longer needed, the window will be cleared. For this reason, the message window is generally empty during normal editor operation.

5-4.6. Memory Available Window

The "memory window" (Window 6) displays the amount of memory available to hold additional rungs. If this number approaches zero, the segment being edited is becoming too large. If such a problem occurs, break the segment up into several smaller sections. These smaller sections can then be edited in turn without risk of total memory depletion. Less memory is available when the editor is run from the Main Menu.

5-5. CURSOR POSITIONING DEVICE (CPD) SUPPORT

The following sections provide information concerning the use of the cursor positioning device (CPD). They detail the following functions:

- Editing a segment(s)
- Moving within a segment(s)
- Adding/Deleting a rung(s)
- Copying/Moving a rung(s)
- Loading/Saving a segment(s)
- Exiting from the editor

5-5.1. Segment Editing Operations

When editing a segment, there are several operations performed at the segment level, such as loading a segment from the disk, finding an existing rung that needs modification, creating new rungs to perform additional functionality, and saving a modified segment to the disk. These operations are performed using cursor targets located in the action window.

5-5.2. Moving Around Within a Segment

Several targets in the action window are dedicated to moving around within a segment in order to find a particular rung. The two most obvious of these are the `NEXT RUNG` and `PREV RUNG` targets. A CPD click on either of these targets will advance the editor to the next rung, or move it back to the previous one. A message will be displayed in the message window if a move beyond the end of the segment is attempted. A third method is the `GOTO RUNG` feature, which allows for direct movement to a particular rung number.

In addition to simply moving forward and backward within the segment, the user may wish to find the rung or rungs in which a particular signal is used. The `FIND NEXT` and `FIND PREV` targets accomplish this task. In order to use these targets, a signal name must be present in the signal name window in the upper right corner of the screen. To enter a signal name into this window, simply start typing when the editor is not expecting any other information (this will be noted by an empty message window). Any characters typed in this manner will automatically be routed to the signal name window. Once the desired signal name is present in the window, click the cursor on either one of the targets, depending upon whether a forward search or a backward search is desired. The next rung in the segment in which that signal name occurs (as a passed parameter) will then become the current rung and will be displayed in the rung display window, ready for editing.

5-5.3. Adding and Deleting Rungs

When editing a segment, users may wish to add new rungs and delete existing ones to modify the function of the segment. There are three targets in the action window dedicated to this type of operation; these include the `ADD NEXT`, `ADD PREV`, and `DEL RUNG` targets. As it's name implies, the `ADD NEXT` target inserts an empty rung immediately after the rung currently being displayed. The new rung is made the current rung, and it's empty RLD grid is displayed in the rung display window. The user is then free to edit the new rung as desired. The `ADD PREV` target operates in much the same way as `ADD NEXT`, except that the new rung is inserted immediately prior to the current rung.

An existing rung in the edited segment that is no longer needed can be deleted by using the `DEL RUNG` target. The `DEL RUNG` target first prompts the user to insure that the deletion request is not a mistake, then deletes the rung (if so instructed). The user should be certain that the rung being displayed is the one that needs to be deleted before allowing the editor to proceed. Once the rung is deleted, the only way to recover it is to quit the editor and lose all changes, or to re-enter the rung by hand.

5-5.4. Copying and Moving Rungs

Rung(s) can be copied or moved from one place to another (within the segment). Four targets in the action menu are used: `SLCT RUNG` (select rung), `COPY RUNG`, `MOVE RUNG`, and `DSLCT ALL` (deselect all).

5-5.4.1. SLCT RUNG TARGET. The `SLCT RUNG` target is used to select one or more rungs for copy or move operations. When the user clicks on the `SLCT RUNG` target, the current rung is selected. This will be indicated by a change in color (white to green). More than one rung can be selected at a time and selected rungs do not need to be contiguous. Individual selected rungs can be deselected by clicking the `SLCT RUNG` target a second time. Once selected, rungs remain selected until they are moved using the `MOVE RUNG` target, or explicitly deselected using the `DSLCT ALL` or `SLCT RUNG` target. This feature allows duplication of a series of rungs within a segment.

5-5.4.2. `COPY RUNG` TARGET. Clicking on the `COPY RUNG` target will copy all selected rungs in the segment to the point immediately after the rung currently being displayed. The original rungs will remain in their current positions within the segment, and will remain selected to allow a series of rungs to be copied to several places within a segment. The new rungs will be added in the order in which they occur in the segment and will not be selected. A pictorial example of a copy operation is shown in the Figure 5-3:

Original Segment With Selected Rungs
R1
R2 - selected
R3
R4 - selected
R5 - current rung
R6
R7 - selected
R8

Segment After Copy Operation
R1
R2 - selected
R3
R4 - selected
R5 - current rung
R2 - new
R4 - new
R7 - new
R6
R7 - selected
R8

Figure 5-3. Copying Rungs

Original Segment With Selected Rungs	Segment After Move Operation
R1	R1
R2 - selected	R2 (Formerly R3)
R3	R3 (Formerly R5)
R4 - selected	R4 (Formerly R2)
R5 - current rung	R5 (Formerly R4)
R6	R6 (Formerly R7)
R7 - selected	R7 (Formerly R6)
R8	R8

Figure 5-4. Moving Rungs

5-5.4.3. `MOVE RUNG` Target. The `MOVE RUNG` target moves all selected rungs in the segment to the point immediately after the rung currently being displayed. The original rungs will no longer exist in their original positions within the segment. The moved rungs will be inserted in the order in which they occur in the segment, and will subsequently not be selected. A pictorial example of a move operation is shown in Figure 5-4.

5-5.4.4. `DSLCT ALL` Target. The `DSLCT ALL` target removes the selected attribute from all rungs in the segment. It should be used after a `COPY` operation has been completed, or if one or more rungs are mistakenly selected. A single rung can be deselected by clicking the `SLCT RUNG` while positioned on the already selected rung.

5-5.5. Loading and Saving a Segment

Three action menu targets are dedicated to performing the disk I/O related functions of loading and saving segment files to and from the editor.

5-5.5.1. `SAVE SEG` TARGET. The `SAVE SEG` target saves an updated copy of the segment source file to disk. All changes made up to the time at which the target is selected will be saved. The editor makes no attempts to verify signal name validity or correct rung construction. The previous source file will be renamed with a `.BAK` extension to allow recovery of the previous file if desired. Since the editor keeps no journal files, it is a recommended practice to periodically click on this target during the course of a segment edit to protect against power failures or system malfunction.

5-5.5.2. `LOAD SEG` TARGET. When the editor is started from the DOS prompt, the program must load data from the system signal database into memory so that it can check names and variable types during the editing process. The `LOAD SEG` target allows the user to load a new segment without exiting to DOS and restarting.

When editing has been completed on the segment currently loaded, any changes made should be saved as described in the previous paragraph. The `LOAD SEG` target can then be selected. If any unsaved modifications exist in the segment, a message in the message window will prompt the user to this fact; the user will then be permitted to abort the operation. If there are no unsaved modifications, or if the user desires to lose the changes made, the filename prompt will be displayed and the user can load a new segment. Once the new segment is loaded, editing can proceed in the usual fashion.

5-5.6. Exiting from the Editor

When all editing operations are completed, the `QUIT EDIT` target can be used to exit the editor and return to the DOS prompt or the main menu. If any unsaved modifications exist in the segment being edited, a message in the message window will prompt the user to this fact; the user will then be permitted to abort the operation. If all modifications have been saved to disk, the quit operation will proceed without interruption.

5-6. RUNG EDITING OPERATIONS

Once a new rung is entered into the segment being edited, or the desired existing rung is displayed on the screen, the rung's contents must be edited in order to enter or modify the information desired. This is done using the basic rung editing operations present in the editor.

5-6.1. Automatic Name Checking Against `UNITDATA.DAT`

An additional feature of the editor is the ability to automatically check the signal names entered into a segment against those that are present in the Control Signal Database (CDB) for the controller contained in the `UNITDATA.DAT` file. If the `UNITDATA.DAT` file is present in the unit specific directory when the editor is first executed, signal name and data type information will be loaded into memory for the remainder of the editing session.

Each time a signal name is used on a rung element, or as a parameter to a PRIMITIVE or BBL, the editor will verify it's existence and check it's data type against the type expected for the operation. Any inconsistencies will be annunciated to the user in the message window, as well as with an audible chirp (which may be disabled by holding the `ALT` key and pressing the `B` key). However, the editor does not prohibit any operations, including the entering of signal names not present in the database. This allows the user to enter new names in the segment as it is edited, with the intent that these points are added to the CDB after editing is complete.

The Control Sequence Compiler will flag all occurrences of undefined signal names or improper signal data types in all segments as errors, and will not generate a downloadable file until all errors have been resolved.

5-6.2. Editing RLD Rungs

Relay Ladder Diagram editing occurs on the 64 element (8 x 8) matrix that appears when a new rung is created. Each matrix element can hold a single RLD component, along with a vertical bus connection to other elements if desired. Editing each rung element can be thought of as a three step process:

1. Type the signal name for the desired element.

2. Select the component type from the element type menu.

3. Position the cursor over the horizontal portion of the rung grid and click the left button on the CPD.

However, it is not necessary to follow this three step process for every element entered. Once a component type is selected, that selection remains in effect for all subsequent component placements on that rung. In addition, once a name is entered into the signal name window, that name will be used until a new name is entered.

All RLD rungs must have an element in the upper left corner of the rung display window.

With certain restrictions it is possible to have more than one output coil on a single RLD rung. The rung must perform normal combinatorial logic up to a point where a single value is brought out to a coil in the right-most column. Once this initial output value has been calculated, it can be used to derive additional coil outputs utilizing intervening RLD logic if desired. The main restriction is that once the initial value has been determined, no additional connections can be brought in from the left hand bus. This feature is illustrated in Figures 5-5 & 5-6.

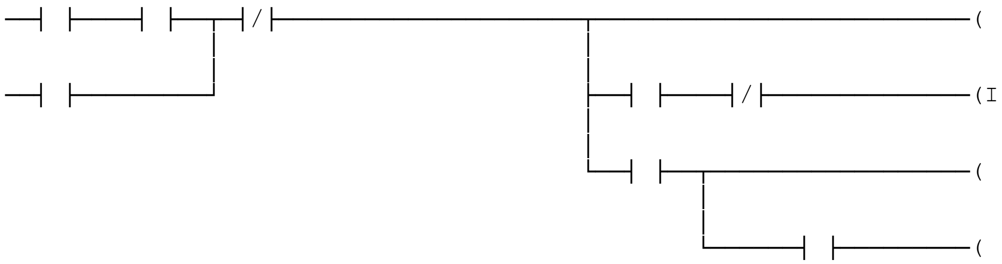


Figure 5-5. A Valid Rung

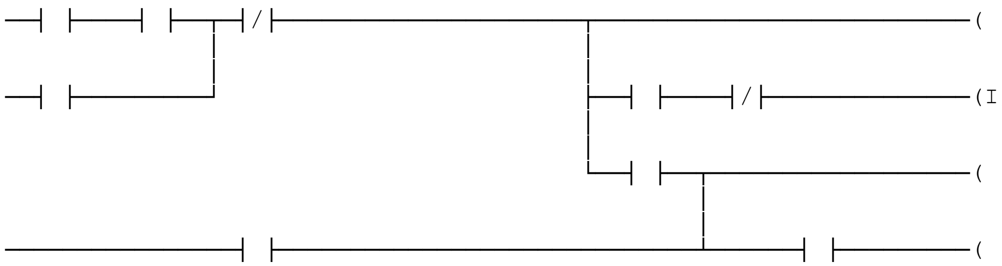


Figure 5-6. An Invalid Rung

All standard RLD elements are available in the element type menu, including normally open contacts, normally closed contacts, horizontal bus, and standard output coils. In addition to these, an additional type of coil, the inverted coil (see Figures 5-5 & 5-6), is available. The signal name currently in the signal name window will be "dropped" on the RLD element with the use of the left-most CPD button.

Vertical bus components are placed by positioning the cursor over the vertical portion of the rung grid and clicking the left-most CPD button. The signal name field and selected component have no effect on the placement of vertical bus.

Several special features make the placement of coils in RLD rungs particularly user friendly. First, if either coil type (normal or inverted) is the selected component, it will automatically be placed into the eighth (right most) column if a click occurs in any other column. Additionally, if a click occurs in the eighth column when the selected component is not a coil, the normal (non-inverted) coil will automatically be used instead of the selected component. Any time a coil is placed in the eighth column, including by the automatic means described above, any unfilled elements to the left of the coil will automatically be filled with horizontal bus. These three features allow the placement of coils without an undue amount of cursor movement and component selection.

Component deletion is another frequently performed operation that the editor makes particularly easy. However, the method for performing this operation differs depending on the number of buttons available on the CPD. For a three button device, simply position the cursor over the component to be deleted and press the center button. The component and it's name will be deleted. Any vertical bus connections must be deleted with an additional CPD button press, and vertical connections can be deleted without deleting the main component in an element.

If the editor is being operated with a two button CPD, then deletions are made by clicking on the `DEL` target in the element type window, and then using the left mouse button in the same manner as the center button was used with the three button device. The `DEL` target will remain highlighted until another component target is picked to allow multiple component deletions. The operator should ensure that an alternate target is picked as soon as deletions are completed to prevent inadvertent component deletions.

One final RLD editing operation that is available in the editor is "picking up" a signal name with the right most CPD button. Signal names can be read off the Rung Display window into the signal name window by placing the cursor over the signal name.

5-6.3. Editing Primitive Rungs

Primitive rungs consist of a limited amount of RLD logic (8 high x 4 wide matrix) identical to that described above, plus a block that controls a subroutine call. The RLD logic portion of the rung can supply up to eight logical inputs into the block, and the block itself can have up to sixteen parameters of any type (logic, analog, count, or constant value).

Before the user can insert a primitive block into a rung, the rung type must be changed from RLD to PRIMITIVE. This is done using the `PRIM` target in the element type window. Clicking the CPD on this target changes the 8 x 8 RLD matrix into the 4 x 8 primitive matrix, clearing an area of the screen to display the primitive block. The primitive name will be displayed in the message window, where the user must type in the name of the primitive insert. If a valid name is entered, the primitive is displayed, and the attached RLD logic can then be entered.

RLD editing operations in primitive rungs are identical to the operations described in the previous section. The user must ensure that a logic string has been connected to each logical input location on the left edge of the primitive block. Beyond this, there are no other requirements. The function of the output coil on the right of the primitive block varies with the block that has been selected, and it is not mandatory to assign a signal name to it.

Parameter name editing within the primitive block starts with the entry of the name in the signal name window in the manner that has been previously described. The CPD cursor is then positioned immediately below the parameter title of interest, and the left CPD button clicked. Again, the name entered will remain in the signal name window and can be entered multiple times if desired. To change a single name, simply reenter the desired point, or the click on the right CPD button. This second option can be used in manner similar to the one used for name changing on coils or contacts. The center CPD button or the `DEL` target in the element type window can be used to delete signal names if necessary.

5-6.4. Editing Big Block Rungs

Big block rungs contain no relay ladder operations whatsoever. They represent subroutine calls that can have up to 63 parameters of any type. Before the user can insert a big block into a rung, the rung type must be changed from RLD to BBL. This is done using the `BBL` target in the element type window. Clicking the CPD on this target will totally blank out the RLD matrix so that the big block can be displayed. A particular big block can be selected by positioning the cursor anywhere within the rung display window and clicking the left-most button. A prompt for the big block name will then be displayed in the message window, and the user must type in the name of the big block that is to be inserted. If a valid name is entered, the block will be displayed, and the parameters can then be edited.

Parameter name editing within big blocks is identical to the operation for primitives. A name is entered in the signal name window and the left CPD button is then clicked with the cursor immediately below the parameter title of interest. The right CPD button can be used to change names, and the center CPD button or the `DEL` target in the element type window can be used to delete names.

5-6.5. Editing Comment Rungs

The final type of rung available in the editor is the comment rung. Comment rungs contain no executable operations. They are a way of textually annotating the surrounding code in order to make it easier to understand. A comment rung can contain up to 39 lines of 66 characters each.

To change a rung into a comment rung, the cursor should be clicked on the `CMNT` target in the element type window. Again, the RLD matrix will be blanked out so that the comment can be displayed.

The comment can be edited by positioning the cursor anywhere within the rung display window and clicking the left-most button. At this point, a separate, block style text cursor will be displayed within the rung display window. This is the cursor that is used to edit comments. It is manipulated using the arrow keys on the keyboard instead of the CPD. The CPD cursor will still be active and can be moved out of the area of interest for the time being.

The comment rung can now be edited using standard ASCII text editing techniques. Characters typed in will appear at the text cursor position, or at the end of the current line if the cursor has been moved beyond the end. New lines can be added by typing a carriage return. As new text is added in the middle of a line, the remaining text will be push over to the right. If the word at the end of the current line reaches the end of the line, it will automatically be moved to the beginning of the next line. Characters can be deleted using the `BACKSPACE` and `DELETE` keys in the normal manner. When text editing is completed, holding the `ALT` key and pressing the `C` key will exit text editing mode, remove the text cursor, and reactivate the CPD cursor.

Notes:

CHAPTER 6

GENERAL INPUT AND OUTPUT CAPACITIES AND SPECIFICATIONS

6-1. CONTACT INPUTS

The Mark V uses the 125 volts DC bus from the internal power distribution core to provide a fuse isolated and current limited interrogation voltage for the contact inputs. Other input voltages between 24 volts DC and 125 volts DC can be used, however, they may require a power source external to the Mark V control. On the terminal boards DTBA and DTBB are jumpers that can be used to isolate field grounds by disconnecting the interrogation voltage in groups of eight contact inputs. Obviously this should only be done while the turbine is not running.

Each contact input is optically isolated on the TCDA card(s), which is located on the contact input and output (I/O) core. Simplex and TMR applications have one (<CD>) core and can have one (<QD1>) or two (<QD1> and <QD2>) digital I/O cores. <QDn> cores interface with the <Q> set of cores. <Q> is the <R> core in Simplex applications and voted combination of <R>, <S>, and <T> cores in TMR systems. The <CD> core transfers digital I/O data to the <C> core.

Each contact input receives a time stamp within 1 millisecond of a status change by the processor located on the TCDA card. The Mark V printer can log each contact input status change with the time stamp if that input is selected for logging. Diagnostic circuitry tests the internal electronics of each contact input every 1 millisecond and initiate an alarm if the circuitry fails. For contact inputs which feed <R>, <S>, and <T> in a TMR system, additional diagnostics compare the value of each contact input and **vote** the data. The result of this "two out of three" voting is the contact status used in the execution of the software, and diagnostic alarms annunciate any discrepancies which exist between the three controllers.
Every contact input destined for the <R>, <S>, and <T> controllers terminates on one terminal point, and is internally parallel wired to all three controllers. See Appendix D for Signal Flow Diagrams.

GAS TURBINE APPLICATIONS

Core	Number of Inputs		Ratings	Comments
	Simplex	TMR		
QD1	96	96 to 3	4 mA at 125 VDC per input: 12 mA for paralleled inputs	Consult factory for voltages other than 125 VDC.
CD	96	96		
P	7	7	Series contacts for "4's" circuit	Manual and external trip inputs to Master Trip Circuit. Open to trip.
P	1 to 3	1 to 3	12 mA at 125 VDC	52G aux contact
P	1	1	Breaker close current	Close Permissive
P	1	1	Breaker close current	Manual Sync Input. Close breaker through sync check only.
QD2*	Not Avail.	96 to 3	12 mA at 125 VDC	Optional, consult factory.

Explanations of the table abbreviations and other notes are following the Large Steam Turbine table.

MEDIUM STEAM TURBINE APPLICATIONS

Core	Number of Inputs		Ratings	Comments
	Simplex	TMR		
QD1	96	96 to 3	4 mA at 125 VDC per input: 12 mA for paralleled inputs	Consult factory for voltages other than 125 VDC.
CD	96*	96		
P	7	7	Series contacts for "4's" circuit	Manual and external trip inputs to Master Trip Circuit. Open to trip.
P	1 to 3	1 to 3	12 mA at 125 VDC	52G aux contact
P	1	1	Breaker close current	Close Permissive
P	1	1	Breaker close current	Manual Sync Input. Close breaker through sync check only
QD2*	Not Avail.	96 to 3*	12 mA at 125 VDC	Optional, consult factory.

LARGE STEAM TURBINE APPLICATIONS

Core	Number of Inputs		Ratings	Comments
	Simplex	TMR		
QD1	96	96 to 3	4 mA at 125 VDC per input: 12 mA for paralleled inputs	Consult factory for voltages other than 125 VDC.
CD	96*	96		
P	7	7	Series contacts for "4's" circuit	Manual and external trip inputs to Master Trip Circuit. Open to trip.
P	1 to 3	1 to 3	12 mA at 125 VDC	52G aux contact
P	1	1	Breaker close current	Close Permissive
P	1	1	Breaker close current	Manual Sync Input. Close breaker through sync check only
QD2*	Not Avail.	96 to 3*	12 mA at 125 VDC	Optional, consult factory.

Table abbreviations and notes:

1. Electro-mechanical relay contacts only.
2. "96 to 3" means 96 contacts parallel wired to 3 cores or cards; either <R>, <S>, <T> in <Q> or <X>, <Y>, <Z> in <P>.
3. Q refers to <R>, <S>, and <T> in TMR applications or <R> in Simplex applications.
4. * means optional, might require additional hardware, consult factory.

6-2. CONTACT and SOLENOID OUTPUTS

The Mark V exclusively uses plug-in type magnetic relays for contact outputs (i.e. no solid state outputs). With the exception of the <P> and <PLU> cores, all of the relays are mounted on the two TCRA cards in each of the <CD>, <QD1>, and optional <QD2> contact I/O cores. The <QDn> core(s) interface with the <R> controller in Simplex applications or with <R>, <S>, and <T> controllers in a TMR system. <CD> core transfers digital I/O (i.e. contact inputs and outputs) to the <C> core.

In a TMR system, the three controllers <R>, <S>, and <T> separately determine the contact output status and the "two out of three" voting takes place at the relay driver. If the three controllers do not agree, a diagnostic alarm is annunciated.

The capacities listed below under "Simplex" and "TMR" are the **total contact outputs available**. By selecting jumpers on the terminal boards, certain outputs can be internally powered by 125 volts DC or 115 volts AC. These internally powered outputs are also called solenoid outputs. Each terminal board is supplied by a 15 amp fuse in the Power Distribution (<PD>) core. The "Solenoid Outputs" column of these tables provide the number of contact outputs that can be internally powered by 125 volts DC or 115 volts AC, depending the application.

Each card has 30 contact outputs; 18 of these can be self powered; 16 of the 18 can be internally powered by 125 volts DC; 2 of the 18 may be powered by 120 or 240 volts AC (15 amps). In gas turbine applications, these two AC outputs are normally used for the ignition transformers.

GAS TURBINE APPLICATIONS

Core	Total Contacts		Number of Contacts that can be Solenoid Outputs	Comments
	Simplex	TMR		
QD1	60	60 from 3	34, 18 per card	2 of 18 are self powered by AC, normally used for ignition transformers
CD	60	60	34, 18 per card	
P	4 from 3	4 from 3	All 4 are self powered and designed as solenoid outputs.	
P	1 from 3	1 from 3		Series logic for automatic sync signal and complete sequence.
P	1 from 3	1 from 3		Breaker Close coil connection, series connection of 25, 25P, 25A relays.
QD2*	Not Avail.	60 from 3	34, 18 per card	AC circuits not available.

Explanations of the table abbreviations and other notes are following the Large Steam Turbine table.

MEDIUM STEAM TURBINE APPLICATIONS

Core	Total Contacts		Number of Contacts that can be Solenoid Outputs	Comments
	Simplex	TMR		
QD1	60	60 from 3	34, 18 per card	2 of 18 are self powered by 120/240 AC.
CD	60*	60	34, 18 per card	
P	2 from 3	2 from 3	Both are self powered and designed as solenoid outputs.	
P	1 from 3	1 from 3		Series logic for automatic sync signal and complete sequence.
P	1 from 3	1 from 3		Breaker Close coil connection, series connection of 25, 25P, 25A relays.
QD2*	Not Avail.	60 from 3	34,18 per card	AC circuits not available

LARGE STEAM TURBINE APPLICATIONS

Core	Simplex	TMR	Number of Contacts that can be Solenoid Outputs	Comments
QD1	60	60 from 3	16	2 of 18 are self powered by 120/240 VAC.
CD	60*	60	16	
P	6 from 3	6 from 3	All are internally powered and designed as solenoid outputs.	
P	1 from 3	1 from 3		Series logic for automatic sync signal and complete sequence.
P	1 from 3	1 from 3		Breaker Close coil connection, series connection of 25, 25P, 25A relays.
QD2*	Not Avail.	60 from 3	16	AC circuits not available
PLU	Not Avail.	16	16	

Table abbreviations and notes:

1. Electro-mechanical relay contacts are factory rated at:

Volts	Inductive	Resistive	Other	Comments
120 AC	2 amps	3 amps	10 amps in-rush	Breaker Control circuits and similar circuits requiring additional interrupting capacity should use other devices to interrupt current flow, i.e. auxiliary contacts.
220-240 AC	2 amps	3 amps	10 amps in-rush	
125 DC	0.2 amps (no suppression)	0.5 amps	0.5 amps (with suppression)	
28 DC	2 amps	3 amps		

2. All contacts are "form C"; three wire; one normally open and one normally closed contact with a common center conductor.
3. "60 from 3" means 60 contact outputs that are voted at the relay driver from <R>, <S>, <T>.
4. "Q" refers to <R>, <S>, and <T> in TMR applications or <R> in Simplex applications.
5. * means optional, might require additional hardware, consult factory.
6. "4 from 3" means 4 outputs that are voted at the relay contacts from X, Y, and Z.

6-3. THERMOCOUPLE INPUTS

Thermocouple inputs to the Mark V are organized in two groups: thermocouples which feed the TCQA cards in the <R>, <S>, and <T> Controllers (or <R> in Simplex applications) and thermocouples which feed the TCCA card in the <C> Controller. A typical Gas turbine application may have eighteen exhaust thermocouples and feed six to EACH controller for redundancy, while steam turbine applications take thermocouples used for rotor stress calculations and split them between <R>, <S>, and <T>. Any temperature which is control or trip related is monitored in this manner to provide two out of three redundancy for both the temperature monitoring device and the internal electronics. Simplex applications have a total or fifteen thermocouple inputs since they only have the <R> Controller.

The <C> Controller can monitor forty-two thermocouples which are divided into three groups of fourteen inputs with separate solid state cold junction compensation. The <C> inputs are typically used for monitoring only - no control or trip functions - with the exception of gas turbine Simplex applications. With a Simplex unit, the exhaust temperature thermocouples are parallel wired at the terminal boards to the <R> and <C> Controllers for redundancy. This configuration provides redundant over-temperature protection because each controller has separate power supplies, processors, trip outputs, and relaying.

Core	Simplex	TMR	Comments
Q	15	45	15 to each core for monitoring and protection. TCQA
Q*	15	45	15 to each core for monitoring and protection. TCQE
C	42	42	In TMR these are used for monitoring only, not for protection. TCCA

Table abbreviations and notes:

1. The following types of thermocouples may be used in any configuration. Each input is set separately in software, thus it is not necessary to group types together.

Type	Range deg F	Range deg C
"E"	-60 to +1150	-51 to +621
"J"	-60 to +1500	-51 to +816
"K"	-60 to +2000	-51 to +1093
"T"	-60 to +750	-51 to +399

2. Thermocouples may be grounded or ungrounded.
3. Thermocouples may have up to 1000 feet (305 meters) of 16 or 18 AWG wire.
4. To provide redundancy, thermocouples are usually split between cores.
5. A separate solid state cold junction compensation device is mounted on the card where the field wires terminate for each group of inputs.
6. Thermocouple diagnostics vary with the application, but any input will alarm is the circuit opens.
7. Linearization for different thermocouple types is provided in software.
8. * Optional feature for (GHD) Gas Heavy Duty prom set only, using the TCQF cards.

6-4. RESISTANCE TEMPERATURE DEVICE (RTD) INPUTS

Core	Simplex	TMR	Comments
C	30	30	TCCA card.
C*	14	14	TCCB card.
C#	14	N/A	LMCB (TCCB) card.
R#	4		LMCB card.

Table abbreviations and notes:

1. The following types of RTD's may be used in any configuration. Each input is set separately in software, thus it is not necessary to group types together.

Ohms	Composition	Manufacturer / Standard	Range deg F	Range deg C
10	Copper	SAMA or GE	-60 to +500	-51 to +260
100	Platinum	SAMA	-60 to +1100	-51 to +593
100	Platinum	DIN 43760	-60 to +1292	-51 to +700
100	Platinum	MINCO (PA)	-60 to +1292	-51 to +700
100	Platinum	MINCO (PB)	-60 to +1292	-51 to +700
100	Platinum	ROSEMONT 104	-60 to +1292	-51 to +700
120	Nickel	MINCO(NA)	-60 to +480	-51 to +249
200 #	Platinum		-60 to 400	-51.1 to 204.4

2. RTD's may have up to 1000 feet (305 meters) of 12 to 18 AWG wire.
3. Linearization for different types of RTD's is done in software.
4. RTD's may be grounded or ungrounded.
5. * means optional, might require additional hardware, consult factory.
6. # LM6000 applications have TCAA, TCCB, and LMCB cards for a total of 58 RTD inputs.

6-5. SEISMIC (VELOCITY) VIBRATION MEASUREMENT DEVICE INPUTS

For application information see Chapter 7.

Core	Simplex	TMR
Q	12	12 TO 3
Q #	3	

Table abbreviations and notes:

1. # LM6000 applications only

6-6. PROXIMITY TRANSDUCER INPUTS

For application information see Chapter 7.

Core	Simplex	TMR	Comments
Q*	18 (8#)	18 to 3	3 of these 18 may be configured as accelerometer inputs.
Q	10 (2#)	10 to 3	Position inputs.
Q	2 (1#)	2 to 3	Key phasor inputs.

Table abbreviations and notes:

1. * In some applications, might require additional hardware, consult factory.
2. # LM6000 applications

6-7. FLAME DETECTOR INPUTS

For application information see Chapter 7.

Core	Simplex	TMR	Comments
P	8 to 3	8 to 3	335 VDC (+1-15V) excitation provided by internal power supplies.

Table abbreviations and notes:

1. "8 to 3" means 8 inputs parallel wired to <X>, <Y>, and <Z>.

6-8. PULSE RATE INPUTS

For application information see Chapter 7.

Core	Simplex	TMR	Comments
Q	7 to <R>	1 to 3, and 6 to each of <R>, <S>, <T> for a total of 19	Magnetic speed sensors, minimum sensitivity 0.03 volts
Q	4 to <R>	2 to 3, and 2 to each of <R>, <S>, <T> for a total of 8	TTL inputs
Q#	4 to <R>	4 to <R>, <S>, <T>	Magnetic speed sensors, minimum sensitivity 0.03 volts
P*	2 to each of <X>,<Y>, <Z> for a total of 6	2 to each of <X>, <Y>, <Z> for a total of 6	Magnetic speed sensors, minimum sensitivity 0.03 volts

Table abbreviations and notes:
1. "1 to 3" means 1 input parallel wired to three; either <X>, <Y> and <Z> in <P> or <R>, <S>, and <T> in <Q>.
2. * In LM6000 applications, 2 to each <X> and <Y> for a total of 4.
3. # Optional feature for (GHD) Gas Heavy Duty prom set only.

6-9. LVDT/LVDR POSITION FEEDBACK INPUTS

For application information see Chapter 7.

Core	Simplex	TMR	Comments
Q	16	16 to 3	TBQC terminal board in location 9 of the <R> core
Q	15	7 to each, 8 to all 3, for a total of 29	TBQF terminal board in location 9 of the <R> core
Q*	6	6	TBQB <S> location 7, optional
Q	20		LM6000 applications only
Q#	8	8 to 3	TBQC terminal board in location 9 of the <S> core

Table abbreviations and notes:

1. "1 to 3" means 1 input parallel wired to three; either <X>, <Y> and <Z> in <P> or <R>, <S>, and <T> in <Q>.
2. * means optional, requires additional hardware, consult factory.
3. All inputs may not be available in all applications. See Chapter 7.
4. Excitation power source provided in Mark V.
5. # Optional feature for (GHD) Gas Heavy Duty prom set only.

6-10. SERVO VALVE OUTPUTS

For application information see Chapter 7.

Core	Simplex	TMR	Comments
Q	8 two coil	8 three coil	-10 to +10 mA, 0 to 20 mA, 0 to 40 mA, or 0 - 200 mA selected by jumpers
Q#	2, two coil		+/- 10 mA to +/- 40 mA
Q#	2, two coil		+/- 10 mA to +/- 240 mA
Q*	4 two coil	4 three coil	-10 to +10 mA only

1. "1 to 3" means 1 input parallel wired to three; either <X>, <Y> and <Z> in <P> or <R>, <S>, and <T> in <Q>.
2. 24 outputs available, one output per coil, three coils per valve, 8 valves total.
3. # LM6000 applications only
4. * Optional feature for (GHD) Gas Heavy Duty prom set only.

6-11. ANALOG VOLTAGE AND CURRENT INPUTS AND OUTPUTS

Most of the analog inputs to the Mark V come directly from transducers on the turbine such as vibration or speed sensors; however, 4 - 20 mA and 0 - 1 mA inputs are provided for other types of transducers.

Each analog input can receive isolated 21 volts DC excitation power for the transducer from the Mark V. The inputs are connected to the terminal board on the TBQC card. Jumpers are provided for such selections as (1) current or voltage input, (2) burden resistor size, and (3) grounding options.

Some inputs can be used as either voltage or current inputs by the selection of a jumper. In current input mode a burden resistor produces a voltage from the transducer current. This voltage is then internally parallel wired with ribbon cables to the TCQA cards in <R>, <S>, and <T> or just <R> in Simplex applications. Any discrepancy between the three controller inputs is annunciated as an internal diagnostic fault while the control and protection system continue to perform their normal calculations based on the median signal. Additional diagnostics monitor the inputs to insure that they are within their proper range.

An additional input to <Q> is provided for the megawatt transducer input on generator drive units. This input is similar to a general purpose 4 - 20 mA input except that it terminates on the QTBA card where a jumper selects whether the input is 0 - 1ma (enabling a 5k ohm burden resistor) or the input is 4 - 20 mA (enabling a 250 ohm burden resistor). The megawatt input is then parallel wired to the TCQC cards in <R>, <S>, and <T>.

If a transducer is powered by the Mark V or by an isolated power supply, a grounding jumper should be installed on the terminal board. If the transducer is separately powered by a grounded supply, the jumper should be removed. Refer to Appendix D for jumper numbers and connection diagrams.

Both 4 - 20 mA (500 ohm max burden) and 0 - 200 mA (40 ohm max burden) analog current outputs are available. An optional card can provide two additional outputs. The TCCA card in the <C> Controller has 4 - 20 mA (500 ohm max burden) outputs available via the CTBA terminal board. These outputs can be independently configured from the Mark V Operator Interface, and are typically used for driving remote instrumentation for monitoring. The <C> controller's 4 - 20 mA outputs cannot be used to drive self powered meters due to voltage regulating conflicts with the meters.

Core	Simplex	TMR	Comments
Q	15	15 to 3	4-20 mA current inputs. 21 volt power. <R>9 TBQC or TBQF 35-80.
Q	4	2 to each, 2 parallel to all 3 for a total of 8	4-20 mA current OR +/-10 VDC voltage inputs. <R>9 TBQB 23 - 50.
Q	2	2 from 3	4-20 or 0-200 mA current outputs. <R>9 TBQC or TBQF 81-84.
Q	1	1 to 3	4-20 or 0-1 mA current OR +/- 10 VDC voltage input. Usually used for MW transducer input. 21 volt power. <R>-6 QTBA 63-64.
Q	1	1 to each OR 1 parallel to all 3	4-20 mA current OR 0-5 VDC voltage. Usually used for pressure transducer inputs. 21 volt power. <R> TBQB-7 1-12.
Q*	2	2 from 3	Current outputs. <S>7 TBQD 105-108.
Q#	6	6 from 3	4 - 20 ma current outputs, 2 on <S> TBQC 81-84, 4 on <S> TBQG 25-32.
Q#	15	15 to 3	4 - 20 ma current inputs, <S> TBQC 35-80.
Q#	4	4 to 3	4 -20 ma current OR +/- 10 VDC voltage inputs, <S> TBQG 33-44.
P	2, 3 phase delta connected		Nominal 120 VAC inputs from Generator Potential Transformers. 1 for Bus volts, 1 for Gen volts.
P	3, isolated, 2 wire		Nominal 5 amp Current Transformer connections.
C	2	2	Stage Link connections. BNC connectors.<C>6.
C	1	1	Time Tic. <C>6 CTBA 35-36.
C	1	1	Shaft voltage input. <C>6 CTBA 79-80.
C	1	1	Shaft current input. <C>6 CTBA 81-82.
C	16	16	4-20 mA current outputs. <C>6 CTBA 1-32.
C	14	14	4-20 mA current inputs. 21 volt power. <C>6 CTBA 37-78.
C*	22	22	Current inputs. 21 volt power. <C>7 TBCB 1-66. 14 of 22 are 0-20 mA, 8 can be 4-20 OR 0-1 mA.

Table abbreviations and notes:

1. * In some applications, might require additional hardware, consult factory.
2. For application notes see Chapter 7.
3. "21 volt power" means 21 volt DC excitation power for the transducer is available. See Appendix D for connection information.
4. "<R>9 TBQC 35-80" means <R> core, location number 9, TBQC terminal board, termination points 35 to 80. See Appendix D for connection information.

5. "15 to 3" means 15 inputs that are internally wired to all 3 controllers.
6. "2 from 3" means 2 outputs from the hardwired median of the three controllers.
7. # Optional feature for (GHD) Gas Heavy Duty prom set only.

6-11.1 LM6000 Simplex Base Engine

Core	Simplex	Comments
Q	15	4-20 mA current inputs. 21 volt power. <R>9 TBQC or TBQF 35-80.
Q	4	4-20 or 40-200 mA current outputs. <R>9 TBQC 81-84 and TBQE 15-18.
Q	1	4-20 or 0-1 mA current OR +/- 10 VDC voltage input. Usually used for MW transducer input. 21 volt power. <R>-6 QTBA 63-64.
Q*	22	Current inputs. 21 volt power. <R>7 TBQD 105-108. 14 of 22 are 0-20 mA, 8 can be 4-20 OR 0-1 mA.
P	2, 3 phase delta connected	Nominal 120 VAC inputs from Generator Potential Transformers. 1 for Bus volts, 1 for Gen volts.
P	3, isolated, 2 wire	Nominal 5 amp Current Transformer connections.
C	2	Stage Link connections. BNC connectors.<C>6.
C	1	Time Tic. <C>6 CTBA 35-36.
C	1	Shaft voltage input. <C>6 CTBA 79-80.
C	1	Shaft current input. <C>6 CTBA 81-82.
C	16	4-20 mA current outputs. <C>6 CTBA 1-32.
C	14	4-20 mA current inputs. 21 volt power. <C>6 CTBA 37-78.
C*	22	Current inputs. 21 volt power. <C>7 TBCB 1-66. 14 of 22 are 0-20 mA, 8 can be 4-20 OR 0-1 mA.

CHAPTER 7

APPLICATION SPECIFIC FUNCTIONS

7-1. PULSE RATE INPUTS

Pulse rate inputs monitored by the Mark V Control Panel fall into three categories: magnetic speed sensors which feed the TCQC cards in the <R>,<S>, and <T> cores (or just <R> in Simplex applications); TTL type inputs which feed the TCQC card(s); and magnetic speed sensors which feed the TCEA cards in the Protection (<P>) core. To provide TMR operation for critical protective functions, there are three independent cards in the <P> core (including Simplex controls). The most common application for the pulse rate inputs is monitoring the turbine shaft speed with magnetic speed sensors. Speed is sensed by magnetic field speed sensors mounted near a multi-toothed wheel attached to the turbine shaft. These are usually 60 teeth wheels but they can be programmed for any number. Voltage pulses are generated by the speed sensors as the teeth of the wheel pass through the magnetic fields surrounding the speed sensors.

The Mark V can interface with the standard passive or the optional active speed sensors with an effective frequency range of 2 to 10,000 Hz. Sensitivity at 2 Hz (that is 2 rpm for a 60 tooth wheel) is 0.033 V peak-to-peak which allows detecting zero speed without the need for any external devices. The maximum input voltage is 200 V peak-to-peak. Six magnetic speed sensors can be monitored by each of the <R>, <S>, and <T> controllers. One additional magnetic pickup is paralleled to all three controllers. A typical application has three speed sensors wired independently to each <R>, <S>, and <T> controller. The voting software then selects the median value for the governing speed parameter and the primary overspeed trip signal in TMR systems. Simplex may have three speed sensors also. Consult the vendor turbine documentation or control sequence program (CSP) for details.

AC permanent magnet generators and pulse tachometers can be used provided they are bolted directly to the shaft. The Mark V Control Panel requires a signal at its I/O terminal board which does not have a dc offset. If a speed sensor signal has a dc offset, the signal must be conditioned to remove the dc offset.

CAUTION

Flexible couplings must not be used for connecting Mark V speed feedback tachometers to the turbine shaft.

Turbines which do not have a mechanical bolt for the "backup" or emergency overspeed protection require a totally separate and independent set of electronics for emergency trip protection. This is provided by the three TCEA cards in the <P> core which has independent power supplies and processors. Each TCEA card can interface with two speed sensors, and the overspeed trip settings are adjustable. (See section 7-3.)

7-2. THE MASTER TRIP CIRCUIT AND THE PROTECTION CORE

Turbine protection in the Mark V Control system is performed by multiple cores within the control panel. The following text and illustrations show the manner in which various protective functions are implemented.

There are two parts to the Master Trip Circuit: the inputs to the Mark V and the outputs from the Mark V.

The "hardwired" or remote trip inputs to the Mark V (contact open to trip) connect to the "4's" relay coils (known as the "4's" for the ANSI standard device number). These relays are energized during normal operation by 125 V dc from the power distribution core. This circuit has redundant relay coils connected to the positive dc bus in series with the external trip contact inputs and another set of redundant relay coils connected to the negative dc bus. (See Figure D-46.) A failure of any one of the four "4's" relay coils will not accidently trip the turbine because the resultant hardware contact vote will "out vote" the failed relay. Only de-energizing both relays connected to the same (positive or negative) dc bus will trip the turbine.

If an external trip signal is received, the "4's" trip circuit will de-energize and the voting contacts circuit will de-energize the internal 24 V dc protective bus. The supply to this 24 V dc protective bus comes from a circuit which selects the maximum voltage from three power supplies; one on each TCEA card in the <P> core.

The 24 V dc protective bus then supplies all the primary trip relays (PTRs), emergency trip relays (ETRs), and other relays which interface with the turbine trip solenoids. These PTR's and ETR's comprise the output portion of the Master Trip Circuit. The microprocessors on the TCEA cards monitor the status of the "hardwired" or remote trip inputs, however, the actual tripping is independent of any microprocessor.

The PTR's and ETR's are located on a card which is specifically designed for a particular turbine application; either the TCTG for gas turbines, TCTS for medium sized steam turbines, or TCTL for large sized steam turbines. The different TCTx (TCTx is an abbreviation for TCTG, TCTS, or TCTL) cards are described in detail in the following sections along with a list of common master protective trip tests offered by each application.

<Q>, which consists of the <R>, <S>, and <T> processors in a TMR system or just <R> in a Simplex system, provides protection for conditions such as loss of flame, starting means failure, excessive exhaust temperature spreads in gas turbines, excessive vibration, over temperatures, primary overspeed trip, and loss of hydraulic oil via the PTRs of the Master Trip Circuit. The PTR trips are JOB SPECIFIC and are defined in the CSP.

The <P> core consists of three independent, identical TCEA printed circuit cards known as <X>, <Y>, and <Z> (Simplex applications also have TMR emergency tripping, therefore, all Mark V systems will have three cards). These cards have their own power supplies and provide emergency overspeed protection, gas turbine flame detection, the automatic synchronization signal, and testing of overspeed trips for certain applications. If two out of three cards "vote" for a trip, a shutdown is initiated by sending signals to the ETR's on the trip card. The automatic synchronization signal energizes an output relay on the trip card and, if two out of three TCEA cards agree, will close the generator's power circuit breaker. Each card has an 80196 microprocessor that runs a continuous loop program which is stored on firmware (PROMs). The <P> core is standard with firmware that does not change with each application and cannot be modified without changing PROMs.

The <P> core master protective functions (ETRs) will be covered in this section. Job specific master protective functions (PTR's) will be covered in the CSP and other turbine vendor documentation.

Figure 7-1 is a simplified block diagram of the protective functions provided by each card (<X>,<Y>, <Z>) in the <P> core. The blocks used are representations of the firmware which can only be changed by replacing the PROMs. The types of blocks used are as follows:

AND	Logical AND function. All inputs must be "true" for the output to be "true".
COMPARE	Compares two signals, with the output being defined by the text. For example, the software parameters are compared with the hardware "Berg" jumpers. If they are not identical, a diagnostic alarm is generated. If the software and hardware are identical, the configuration information is sent on to the next block.
ISO	Isolates and conditions signals from field devices to the levels used in the processors.
LATCH	Latches the trip signal until reset manually.
LIMIT	Limits the input to the range of the MAX and MIN values.
RANGE	Limits the range of the output to within the limits defined by the MIN and MAX
CHECK	values. If the input is within range, the output equals the input . If the input is not within range, a diagnostic alarm will be generated.

OR	Logical OR function. Any "true" input will result in a "true" output.
RD	Relay Driver, converts logic level signals to drive the 24 V dc ETR relays.
SCALE	Scales the speed signal pulses using the configuration data to be a percent of rated speed.

The small boxes near the signal lines contain the typical Mark V software signal names for these signals. The values of these control signal data base point names can be monitored via the operator interface (<I>).

The text in quotations is the diagnostic alarm message that will be annunciated if that alarm condition is active.

The table on the first page gives dc voltage values for TP4 (or ANA OUT or ANALOG OUT as it may be called on the card) at various speeds of the H.P. shaft. This is an output from the microprocessor and should only be used for diagnostic purposes. Use only high input impedance (10 Megohms or greater) and non-grounded (differential input) measurement devices like a digital multi-meter (DMM), oscilligraph, or oscilloscope in differential mode, and reference to DCOM (TP1).

A logic high or "1" from any trip signal will de-energize the ETR relays and cause the turbine to shut down. The status of the <P> core trip signals while in normal running condition is a logic low or "0".

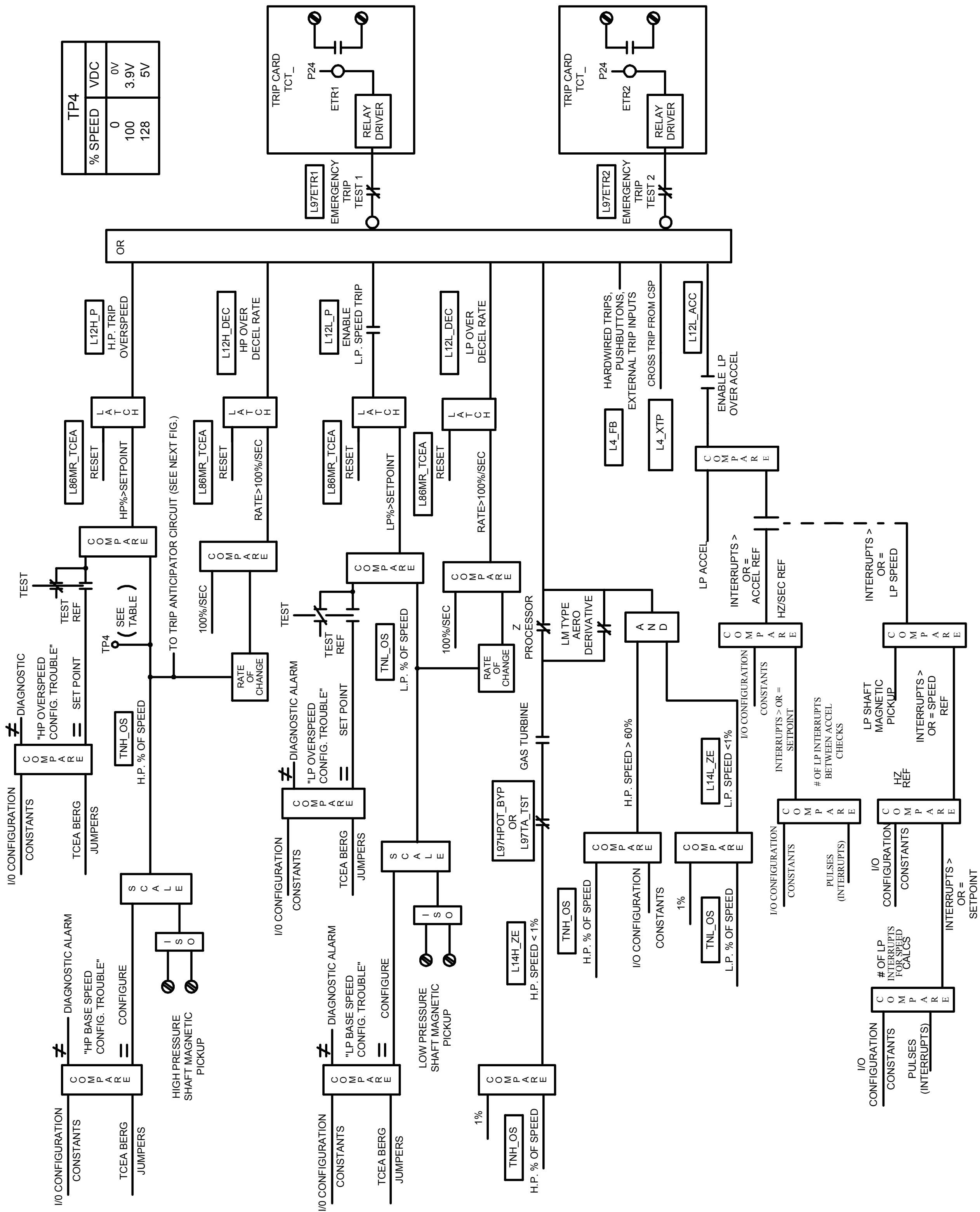


Figure 7-1 Simplified Block Diagram - TCEA Cards

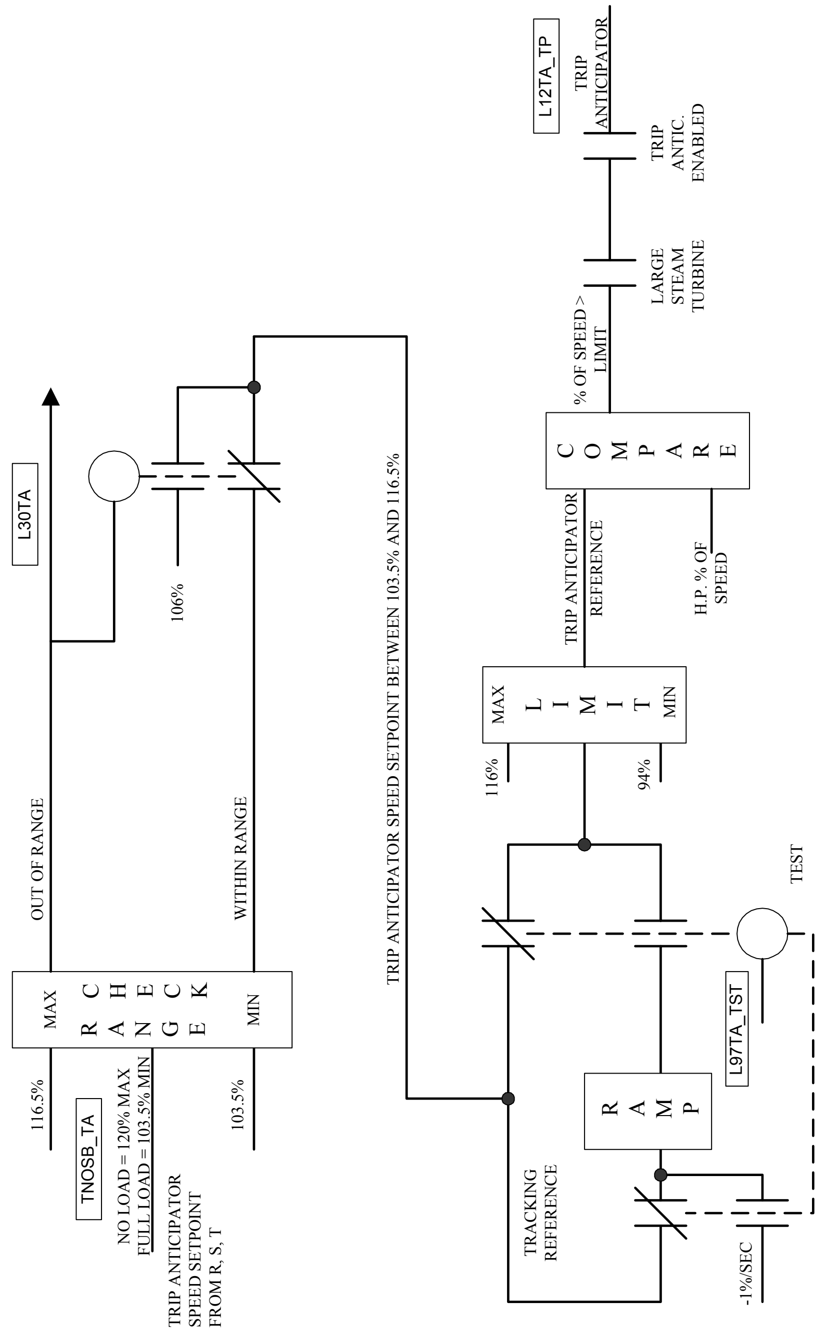


Figure 7-1 Simplified Block Diagram - TCEA Cards, continued

The emergency overspeed trips that originate in the <P> core are as follows:

1. High Pressure Shaft (HP) Overspeed Trip:
   If the HP shaft scaled speed feedback exceeds the setpoint, an HP shaft overspeed trip will be initiated. This trip signal will be latched until a master reset command is given. For testing a different setpoint can be substituted for the design value.

2. High Pressure Shaft Over Deceleration Rate:
   If the HP shaft scaled speed feedback decreases faster than 100% per second, an HP shaft over-rate trip will be initiated. This fast rate of change usually indicates a defective speed sensor or wiring. This trip signal will be latched until a master reset command is given.

3. Low Pressure Shaft (LP) Overspeed Trip and LP Shaft Over Deceleration Rate:
   If enabled in a dual shaft system, these trips function in the same fashion as respective HP shaft trips.

4. LM Type (Aero Derivative) Speed Trips:
   If the system is an LM type aero derivative, two additional protective functions are provided in the X and Y cards only (the cards are identical; jumpers select the type of system). Type LM machines normally have only two speed sensors, therefore, the Z processor output is inhibited. If the HP shaft speed is greater than 60% and the LP shaft speed is less than 1%, a trip will be initiated. Also, any time the HP shaft speed is less than 1% of rated speed a trip will be initiated.

5. Large Steam Turbine Trip Anticipator:
   In large steam applications, a trip anticipator (TA) function is available. The overspeed trip setpoint is reduced as the load is increased which will reduce the turbine speed overshoot in case of full load rejection. <R>, <S>, and <T> determine the TA speed setpoint and the median value is sent to the <P> core. This setpoint is confirmed to be within a valid range, limited to high and low preset values and becomes the TA reference. If the TA setpoint from <R>, <S>, and <T> is out of range, a diagnostic signal is sent back to them and a fixed value of 106% is used for the TA reference. If the HP shaft scaled speed feedback signal exceeds the TA reference, a trip is initiated. A test circuit tracks the TA speed setpoint and when test is selected, the processor inhibits a trip then reduces the setpoint at a rate of 1% per minute until a TA trip is activated.

6. Hardwired Trips:
   Any pushbuttons or other hardwired trips initiated outside the Mark V control system will be connected to the PTBA terminal board in the <P> core (see Appendix D, D-46). These hardwired trips will de-energize the 24 V dc supply from all the ETR and PTR relays. The <P> core will also initiate a trip.

7. Cross Trip from <Q>:
   If the <Q> core initiates a trip to the PTR relays, a backup signal will be sent to the <P> core to also trip the ETR relays.

Remember that each of these trip functions is derived independently by three separate cards (X, Y, Z) from separate speed feedback devices (in a complete TMR system). These cards then send their respective trip signals to three separate relays on the trip (TCTx) card. The TMR voting takes place with the relay contacts.

7-2.1. The Trip Cards: TCTL, TCTS, TCTE, and TCTG

The trip cards have at least two separate sets of three primary trip relays (PTR1 & 2), and two separate sets of three emergency trip relays (ETR1 & 2). Some applications require additional relays for other functions. Refer to the following sections for details. The PTRs are controlled from each of the TCQA cards in <R>, <S>, and <T>, and the two ETRs are controlled from the TCEA cards in <P>. If two out of three PTRs or two out of three ETRs de-energize, the turbine will trip. Throughout the documentation and drawings, the abbreviation "2/3" has been adopted to mean "two out of three".

7-2.1.1. LARGE STEAM TURBINE TRIP CIRCUIT - TCTL. If 2/3 ETR1s or 2/3 PTR1s are de-energized or the trip pushbutton is pressed, the mechanical trip solenoid valve (MTSV) will energize and trip the turbine. If 2/3 ETR2s or 2/3 PTR2s are de-energized, the electrical trip solenoid valve (ETSV) will de-energize and trip the turbine. The MTSV is powered from the 125 V dc bus in the internal power distribution core. The ETSV is powered from the same 24 V dc protective bus (described in section 5.2.1 of this manual) that powers all the ETR and PTR relays. The other solenoids are powered from the internal 115 V ac feeder (see Appendix B for the card diagram).

In addition to the trip control relays, the TCTL card has the following four auxiliary control outputs available:

1. Electrical Lock Out Solenoid (ELO)
2. Mechanical Lock Out Solenoid (MLO)
3. Oil Reset Solenoid Valve (ORST)
4. Oil Test Solenoid (OTS)

Several tests can be performed with a full complement of hardware. Some of the tests that the Mark V can perform include:

1. Mechanical Trip Piston Tests (on and off line)
2. Electrical Trip Test (on and off line)
3. TMR Cross Trip Test (on and off line)
4. Emergency Over Speed Cross Trip Test (on and off line)
5. Electrical Overspeed Trip Test (on line)
6. Electrical Overspeed Trip Test (off line)
7. Actual Mechanical Overspeed Trip Test (off line)
8. Actual Electrical Overspeed Trip Test (off line)

As each application differs, the performance of applicable tests will be described in the CSP and other requisition specific documentation supplied by the turbine vendor.

7-2.1.2. MEDIUM STEAM TURBINE TRIP CIRCUIT - TCTS. If 2/3 PTR1s or 2/3 ETR1s de-energize, the electrical trip device solenoid #1 (ETD1) will de-energize and trip the turbine. Similarly, if 2/3 PTR2s or two 2/3 ETR2s de-energize, the electrical trip device solenoid #2 (ETD2) will de-energize and trip the turbine. Separate primary (KP1 & 2) and emergency (KE1 & 2) bypass relays are provided to facilitate on-line testing of the PTRs and ETRs, respectively. (See Appendix B for the card diagram.)

Several tests can be performed with a full complement of hardware. Some of the tests that the Mark V can perform include:

1. Primary Overspeed Test (on and off line)
2. Emergency Overspeed Test (on and off line)
3. Electrical Trip Device Tests with hydraulic lockout (on line)
4. Electrical Trip Device Tests (off line)
5. Stop Valve Test (on line)
6. Reheat Stop Valve Test (on line)

As each application differs, the performance of applicable tests will be described in the CSP and other requisition specific documentation.

7-2.1.3. LARGE/MEDIUM STEAM TURBINE TRIP CIRCUIT - TCTE. If 2/3 PTR1s or 2/3 ETR1s de-energize, the electrical trip device solenoid #1 (ETD1) will de-energize and trip the turbine. Similarly, if 2/3 PTR2s or two 2/3 ETR2s de-energize, the electrical trip device solenoid #2 (ETD2) will de-energize and trip the turbine. If 2/3 ETR3s or 2/3 PTR3s are de-energized, the electrical trip solenoid valve (ETSV) will de-energize and trip the turbine. ETD1 and ETD2 are powered from the 125 V dc bus in the internal power distribution core. The ETSV is powered from the same 24 V dc protective bus (described in section 5.2.1 of this manual) that powers all the ETR and PTR relays. (See Appendix B for the card diagram.)

Several tests can be performed with a full complement of hardware. Some of the tests that the Mark V can perform include:

1. Primary Overspeed Test (on and off line)
2. Emergency Overspeed Test (off line)
3. Electrical Trip Soleniod Valve Test (on and off line)

As each application differs, the performance of applicable tests will be described in the CSP and other requisition specific documentation.

7-2.1.4. GAS TURBINE TRIP CIRCUIT - TCTG. The TCTG card has four sets of three PTRs, two sets of three ETRs, and three relays for synchronizing in addition to other auxiliary relays. The Table 7-1 details which solenoid output will de-energize for a given relay output. The "X" indicates this solenoid will de-energize if the relay is de-energized. See Appendix D for the card diagram.

Table 7-1. Solenoid Output

Relays	Solenoid 1 typically 20FG	Solenoid 2 typically 20HD	Solenoid 3 typically 20FL	Solenoid 4
2/3 ETR 1	X		X	
2/3 ETR 2		X		X
2/3 PTR 1	X			
2/3 PTR 2		X		
2/3 PTR 3			X	
2/3 PTR 4				X

Several tests can be performed with a full complement of hardware. Some of the tests that the Mark V can perform include:

1. Primary Overspeed Test (on and off line)
2. Emergency Overspeed Test (off line)
3. Mechanical Overspeed Bolt Test (off line)

As each application differs, the performance of applicable tests will be described in the CSP and other requisition specific documentation.

7-3. SETTING THE PRIMARY AND EMERGENCY OVERSPEED TRIP SETPOINTS

The Mark V control system provides two independent levels of electronic overspeed protection—primary and emergency—for up to two turbine shafts. These two electronic overspeed functions are in addition to any mechanical overspeed protection systems provided by the turbine vendor, such as a mechanical overspeed bolt mechanism.

Primary overspeed detection and trip initiation is accomplished by the overspeed protection Big Block Language (BBL) in the CSP in the <R>, <S>, and <T> control processors (or just <R> in Simplex) using the primary turbine shaft speed sensors. The Control Constants define the trip setpoints for the blocks. The magnetic speed sensor pulses are filtered and scaled (converted to percent of rated shaft speed) by the I/O cards. When the scaled turbine shaft speed signal exceeds a Control Constant setpoint, the CSP overspeed block initiates a turbine trip.

Designed to be used in place of the mechanical overspeed mechanism, emergency overspeed protection is accomplished by the protective processors (<X>, <Y>, and <Z>) in the protective core, <P>. Emergency overspeed protection is completely independent of primary overspeed protection. A second set of turbine shaft speed sensors is usually connected to the PTBA I/O terminal board on the <P>. If the additional shaft speed sensors are not available, the primary shaft speed sensors can be paralleled to the emergency shaft over speed sensor input points.

NOTE

The Mark V will not permit turbine operation without HP shaft speed sensor inputs connected to the protective processors. LP shaft emergency speed sensors must be connected, as well, if the unit has an LP shaft.

Emergency overspeed protection setpoints are configured both by I/O Configuration software settings and by hardware jumper settings on the TCEA cards in <P>. If these two are not set the same, a diagnostic alarm will be annunciated, and the panel initialization will not complete successfully.

The frequency of the pulses from the shaft speed sensors depends on the number of teeth on the feedback wheel as well as shaft RPM. For example, most heavy duty gas turbines use magnetic speed pickups with 60-toothed speed sensing wheels for primary and emergency shaft speed feedback signals. In this case there are 60 pulses per revolution, so the frequency of the speed signal is equal to the RPM of the shaft (1 Hz equals 1 RPM). Many steam turbines use magnetic speed pickups with 160-toothed speed sensing wheels, thus 2.667 Hz equals 1 RPM. The Mark V I/O card(s) scale the primary speed feedback signals to be a percent of rated shaft speed. Rated shaft speed is the speed at which the turbine is designed to be operated, not the highest possible maximum shaft speed.

Emergency speed feedback signals are normally scaled to percent of rated design speed, not necessarily normal operating speed. For example, a turbine whose rated design speed is different from its normal operating speed is a Frame 6 heavy duty single shaft gas turbine being used as the prime mover for a generator through a gear box. To operate at synchronous speed, the input to the gear box must be at 5196 RPM. The rated design speed for a Frame 6 heavy duty single-shaft gas turbine is 5100 RPM. In this case, the normal operating speed would be 5196 RPM. Therefore it is possible that the scaled primary speed feedback signal and the scaled emergency speed feedback signal may not have the same value for percent of HP shaft speed.

Sections 7-3.1 and 7-3.2 describe the operation and adjustment of the primary and emergency overspeed protection functions. Section 7-3.3 covers additional concerns of Simplex applications.

7-3.1. Primary Overspeed Protection

The primary turbine speed sensors are individually connected to a pulse rate input on the QTBA I/O card in location 6 of each control processor core. Thus, each control processor has its own independent shaft speed sensor. The frequency from each primary shaft speed sensor is converted to a percent of normal operating speed by the I/O Configuration software on the TCQA card of each control processor. The definition of "normal operating speed" is made on the TCQA card "`Pulse Rate Definition`" screen of the I/O Configurator (see Appendix B, TCQA Card, screen 17/21).

The frequency defined for the input's `Max Pulse Rate` (100% rating) is the input frequency which will be equal too normal operating turbine speed. This value must be expressed in frequency, not in RPM; the input frequency will depend on the type of input device and the number of teeth on the speed sensing wheel. Each processor's digitally scaled speed feedback signal is communicated to the other control processors and to the communication processor <C>. Then each control processor "votes" the data by choosing the median of the three and uses the voted value in the execution of the CSP. The Control Data Base (CDB) point name for HP turbine shaft speed is almost always "TNH" and its value is always expressed in percent of normal operating speed.

The voted value of "TNH" is used in the overspeed detection BBL block (see Appendix C) in each control processor's CSP. If the voted value of "TNH" exceeds the primary overspeed trip value from the Control Constant table file (usually TNKHOS and also expressed as a percent of normal operating speed), that processor will vote for a trip via the primary trip relays. (See Section 7-2.) As with all Control Constants, the value of the turbine primary overspeed Control Constant is determined by the turbine vendor, and should not be modified without the concurrence of the factory.

7-3.2. Emergency Overspeed Protection

Emergency overspeed protection is provided by the Mark V control panel to be used in place of a mechanical overspeed mechanism (for example, an overspeed bolt). Even when a bolt is used, the HP (High Pressure) shaft emergency overspeed protection function must have speed feedback because it will always be active. If LP (Low Pressure) shaft emergency speed protection is not necessary, it can be disabled by setting the I/O Configuration Constant overspeed setting values and hardware jumpers to zero. In either TMR or Simplex applications there must be three HP shaft speed feedback signals for emergency overspeed protection. If three seperate emergency overspeed speed sensors are not available, the primary speed feedback signals may be paralleled to the emergency speed feedback inputs.

The I/O Configuration overspeed setting and hardware jumper settings must agree for the control processor (<R>, <S>, or <T>) associated with each TCEA card to reach "ready to run" status (the first two digits of the display on the LCC card momentarily display "A7"). The HP shaft emergency overspeed setting (and hence the TCEA HP shaft emergency overspeed hardware jumper settings) are defined by the turbine vendor and should not be modified without concurrence; it can be calculated using the formula shown on the I/O Configurator screen and in Appendix A.

The speed feedback frequency from the sensing device is scaled to a digital value proportional to rated design speed. The CDB point name for scaled HP shaft emergency speed feedback is typically TNH_OS and it is expressed as a percent of rated design speed (not necessarily the same as normal operating speed: see Section 7-3). An example of an HP shaft emergency overspeed I/O Configurator screen can be seen in Appendix B.

The microprocessor on each TCEA card compares TNH_OS to the overspeed setpoint and when TNH_OS exceeds the setpoint a turbine trip is initiated by de-energizing the ETRs on the TCTx card (see Section 7-2).

LP shaft emergency overspeed detection and trip is accomplished in an identical manner, with CDB pointname for the scaled LP shaft emergency speed feedback being TNL_OS. If a unit does not have an LP shaft, then the LP shaft emergency overspeed setpoints (both software and hardware) must all be set to zero.

7-3.3. Simplex Applications

Mark V Simplex control panels have only one control processor, <R>, but will still have three protective processors (<X>, <Y>, and <Z>) in <P>. Simplex control panels will require at least one speed sensor input per shaft to <R> and three individual speed sensor inputs per shaft to <P> (one for each of the protective processors). Simplex control panels have the ability to accept a second, redundant speed sensor input per shaft to <R>. If provided, redundant speed sensors require appropriate modifications to the CSP in order to choose the higher of the two speed feedback signals for use in the CSP as well as the primary overspeed BBL block.

7-4. SERVO VALVE DRIVE SYSTEM

The servo valve drive system acts as the interface between the Mark V control system and the hydraulic actuators that position the mechanical devices. The basic system compares the actual position of the hydraulic actuators to a setpoint and outputs a position control signal that maintains the system balance.

The system consists of the following components:

- Mark V Digital Closed Loop Regulator and Interface circuit.
- Servo Valve for control of oil to the hydraulic actuator or cylinder.
- Feedback devices such as LVDTs, LVDRs, pressure transducers, or rate of flow detectors.
- Hydraulic actuator and a source of hydraulic power to position mechanical devices.

7-4.1. The Servo Valve

The servo valve is used to control the direction and rate of movement of a position actuator of a control device. In effect, it acts as the interface between the electrical and mechanical systems by converting an electrical signal to a hydraulic output. In response to this electrical input signal (typically less than one watt), the servo valve modulates high pressure hydraulic fluid to the actuator.

The Mark V can control servo-valves with up to three coils. In a typical TMR system, the <R> control processor is connected to one coil, <S> is connected to the second coil, and <T> is connected to the third coil of a three-coil servo valve. The three currents through three coils wound around one steel laminated core produce one magnetic field. This magnetic field moves a valve that controls the flow of high pressure oil to the actuator. This is a form of "hardware voting", where the individual outputs of the redundant control processors are "voted" (summed) by the action of the servo-valve's coils. This avoids a potential single point failure within the electronics from tripping the turbine. A failure of a controller, its output port, or the physical connection to the output coil will result in the other two servo drives compensating for the failed channel and keeping the valve properly positioned.

The servo-valve outputs of a Mark V Simplex control panel will generally be connected to a two-coil servo valve. Each output has a second output terminal to which one wire of the second coil of a two-coil servo-valve can be connected. If one servo-valve coil's circuit were to open (loose wire, broken coil connection , and so on), the other coil is capable of positioning the device until the servo-valve can be replaced.

7-4.2. Regulator Feedback Devices

This section describes the operation of the following feedback devices:

- LVDT or LVDR Position Feedback
- Rate of Flow of Liquid Fuel (sometimes called flow-rate feedback)
- Pressure Feedback from pressure transducers

For additional details on the operation of regulator, see the Mark V Digital Closed Loop Regulator and Interface section, below.

7-4.2.1. LVDT or LVDR POSITION FEEDBACK The physical position of the actuator is detected by a LVDT (Linear Variable Differential Transformer) or a LVDR (Linear Variable Differential Reactor) and converted to a voltage signal that is fed back to the control processors. The variable ac voltage signal from an LVDT or LVDR is converted from an analog value to a digital value, then scaled by the regulator to a value proportional to position for use by the regulator. When the actual position is equal to the setpoint, the signal to the servo valve holds the valve in the lapped, closed, or null position. If the system is not in balance (the hydraulic actuator is not at the setpoint position), the controller will position the valve in the proper direction, restoring the system balance by repositioning the hydraulic actuator.

7-4.2.2. RATE OF FLOW OF LIQUID FUEL FEEDBACK Flow-rate feedback signals to the digital control regulators are generally provided by magnetic speed pickups. The pulse rate from a magnetic speed pickup is scaled by the I/O Configuration Constants to a value proportional to flow-rate for use by the regulator.

7-4.2.3. PRESSURE FEEDBACK Pressure feedback signals to the digital control regulators are generally provided by either voltage or current transducers. After being converted from an analog value to a digital value, the voltage or current signal from the transducer is scaled by the I/O Configuration Constants to a value proportional to pressure.

7-5. MARK V INTERFACE

The current of each servo-valve output is the result of the comparison of a reference value and a feedback value by a digital control regulator (or, regulator) defined for that output. A functional block diagram of the basic operation of a digital control regulator, its inputs, and its output are shown in Figure 7-2.

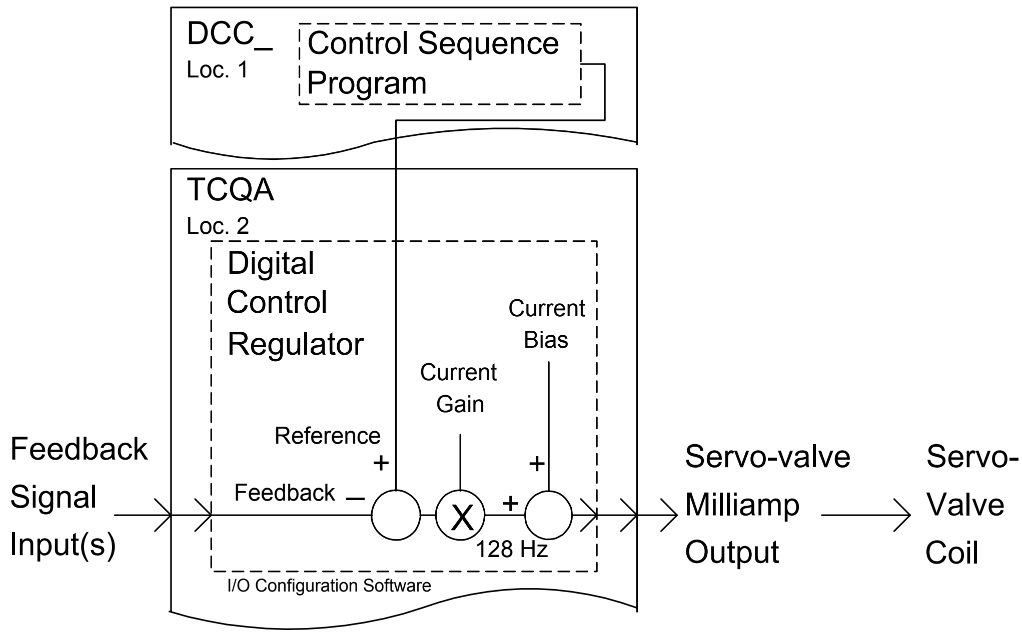


Figure 7-2. Functional Block Diagram of a Digital Control Regulator

The reference value for a servo-valve output's digital control regulator is calculated in the control processor's CSP, where it is written into the processor's control signal database and then read from the control signal database by its defined regulator. The association of a CSP-calculated reference value to a particular servo-valve output is made when the reference value's CDB pointname is assigned to an output point in the unit's I/O assignment file (IO.ASG). The type of regulator associated with a particular servo-valve output is made when a function type and sub-type is defined on the servo-valve output's regulator definition screen of the I/O Configurator. A graphical representation of how a CDB pointname and its value are assigned to a servo-valve output and how a regulator function is defined for that output is shown in Figure 7-3. The value of CDB pointname V1_OUT was assigned to servo-valve output #1 (SVO1) in IO.ASG, and regulator function type 43 was defined for SVO1 in the I/O Configurator.

The digital control regulators are part of the I/O Configuration software on the TCQA I/O card in each control processor in the Mark V. While the type of regulator function (position, flow-rate, or pressure control) defined for a servo-valve output can be changed using the I/O Configurator, the operation of the regulator cannot. Algorithms used by the digital control regulator functions are stored on EPROM chips on the TCQA card and, as such, cannot be altered or modified unless new EPROM chips are obtained from by GEDS.

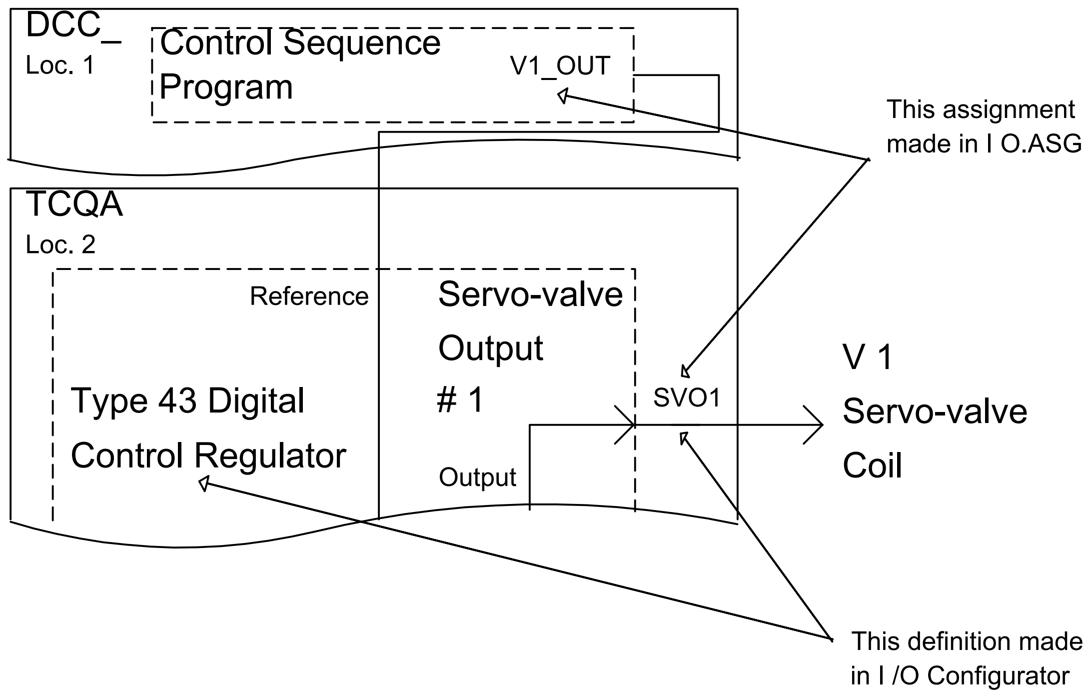


Figure 7-3. Reference and Regulator Function Definition for Servo-Valve Output

The type of regulator defined for a particular servo-valve output is dependent upon the type of control function required of the output. The following five digital control regulator functions are available in the Mark V control system:

1. Position control with cylinder and pilot valve position feedback
2. Position control with position feedback
3. Flow control with flow-rate feedback
4. Flow control with flow-rate and position feedback
5. Pressure control with pressure and position feedback

For more information about the operation of each regulator, see Section 7-5.3 of this manual.

Each servo-valve output has a regulator definition screen in the TCQA Card Definition portion of the I/O Configurator. A two character hexadecimal number, such as 53 or 7A, is used to specify the regulator function type and sub-type. The first character, or most significant digit, is called the "basic" regulator type number and defines the digital control regulator function (one of the five listed above). The second character (least significant digit) defines the number of feedback signals and how the regulator will handle redundant feedback signals.

Because the digital control regulators are used on various applications with differing characteristics, the parameters used by the regulators (gains, biases, offsets, limits, and so on) vary with the unit and may be adjusted during start-up. The majority of I/O Configuration Constant values for each regulator are entered on the servo-valve ouput's regulator definition screen of the I/O Configurator. Pressure control and flow control regulators have some of their feedback scaling constants defined on separate screens in the I/O Configurator.

CAUTION

After stable operation has been achieved, the regulator parameters should not need to be modified unless critical system components have been changed. The digital regulators will not drift or change with time, and excessive adjustment could make the system unstable or oscillatory.

7-5.1. Regulator Function Types and Sub-Types

The five digital control regulator functions listed below are the "basic" types of regulators in a Mark V control system along with common examples of their usage.

Regulator Function Type Number	Digital Control Regulator Function	Examples of usage/application
2	Position control with cylinder and pilot valve position feedback	GE-Fitchburg low-pressure hydraulic actuator steam turbine admission valves
4	Position control with position feedback	Steam turbine admission and extraction valves; gas fuel gas control valve; variable inlet guide vanes
5	Flow control with flow-rate feedback	Liquid fuel bypass valve
6	Flow control with flow-rate and position feedback	Liquid fuel bypass valve
7	Pressure control with pressure and position feedback	Gas fuel stop/ratio valve (heavy duty gas turbines)

Table 7-2. Digital Control Regulator Function Type Numbers

The regulator function sub-type number defines the number of feedback signals to be used by the regulator and how the regulator is to handle redundant feedback signals if they are being provided. The following five tables list the regulator function sub-type numbers for each regulator type/function.

Type	Sub-type	Position Control Regulators with Pilot and Cylinder Position Feedback
2	D	Two assigned LVDT/LVDRs
2	E	Maximums of two pairs of assigned LVDT/LVDRs

Table 7-3. Position Control Regulators

Type	Sub-type	Position Control with Position Feedback
4	0	No position feedback
4	1	First assigned LVDT/LVDR
4	3	Maximum of two assigned LVDT/LVDRs
4	9	Median of three assigned LVDT/LVDRs

Table 7-4. Position Control with Position Feedback

Type	Sub-type	Flow Control with Flow-rate Feedback
5	1	First flow input
5	2	Second flow input
5	3	Maximum of two flow inputs

Table 7-5. Flow Control

Type	Sub-type	Flow Control with Flow-rate and Position Feedback
6	4	First flow input and maximum of two assigned LVDT/LVDRs
6	5	Second flow input and maximum of two assigned LVDT/LVDRs
6	6	Maximum of two flow inputs and maximum of two assigned LVDT/LVDRs

Table 7-6. Flow Control with Position Feedback

Type	Sub-type	Pressure Control with Pressure and Position Feedback
7	7	First pressure input and maximum of two assigned LVDT/LVDRs
7	8	Second pressure input and maximum of two assigned LVDT.LVDRs
7	A	First pressure input and first assigned LVDT/LVDR
7	B	Second pressure input and first assigned LVDT/LVDR
7	C	Maximum of two pressure inputs and maximum of two assigned LVDT/LVDRs

Table 7-7. Pressure Control

7-5.2. Regulator Feedback Signals

The digital control regulators, with the exception of type 40, use feedback signals to accurately position the devices they control via the servo-valves. The connection and distribution of feedback signals vary with the application and understanding this is helpful when configuring or troubleshooting a Mark V.

Several of the feedback signal input points to a Mark V control panel are "reserved" (that is, primarily intended) for use by the digital control regulators and their servo-valve outputs. Each control processor (<R>, <S>, and <T>) in a Mark V control panel has provisions for its own feedback device(s), either magnetic speed pickups, pressure transducers, or LVDT/LVDRs. LVDT/LVDR position feedback signals are handled in one of two methods in a Mark V control panel depending on the control philosophy of the application.

7-5.2.1. MAGNETIC SPEED PICKUP/PULSE RATE FEEDBACK SIGNALS. Figures 7-4, 7-5, and 7-6 show how the two basic types of flow control regulators handle their flow-rate feedback inputs. Note that each control processor has its own individual pulse-rate inputs. Magnetic speed pickup/pulse-rate inputs MPU3 and MPU4 to each control processor (control processor I/O terminal board QTBA, location 6, terminals 55 & 56 and 57 & 58, respectively) are intended primarily for use as liquid fuel flow-rate inputs by the two basic types of flow control regulator functions. For flow control regulators, MPU3 is defined as the first flow-rate input and MPU4 the second flow-rate input for each control processor. Depending on the regulator function sub-type, the regulator will use either the first flow input (MPU3), the second flow input (MPU4), or the maximum/greater feedback signal from both flow inputs. Each control processor in a Mark V control panel has a QTBA I/O terminal board which handles I/O dedicated to that control processor; refer to Appendix D. For additional information, see Sections 7-1 and 6-8.

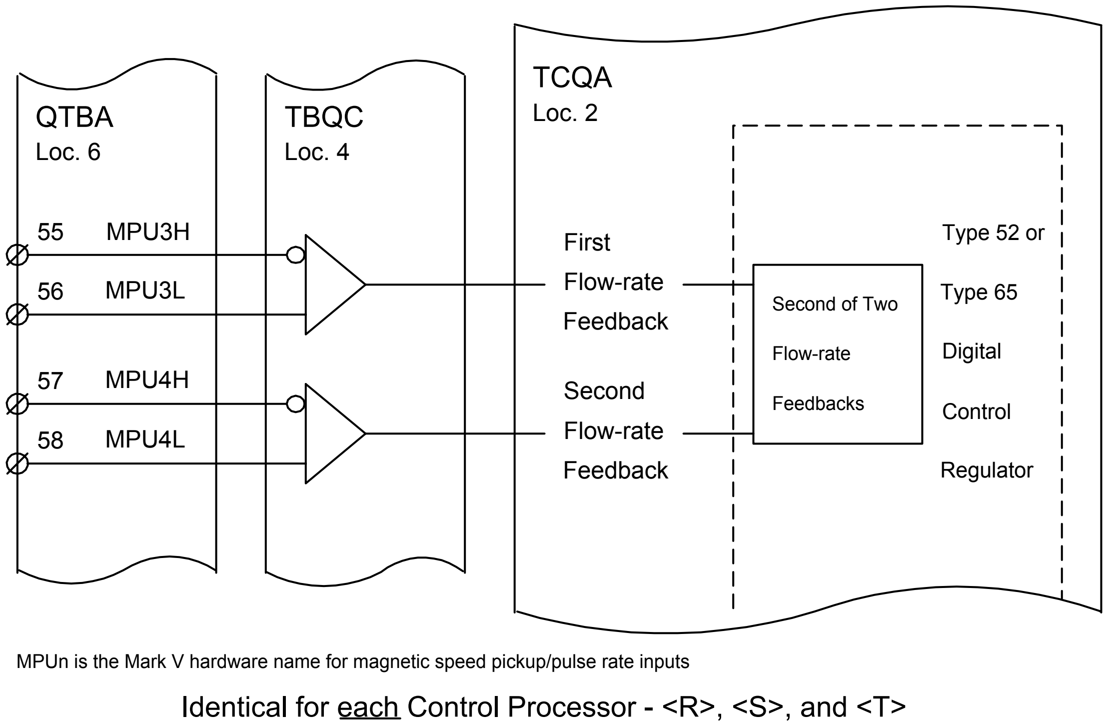


Figure 7-4. Type 52 or 65 Regulator

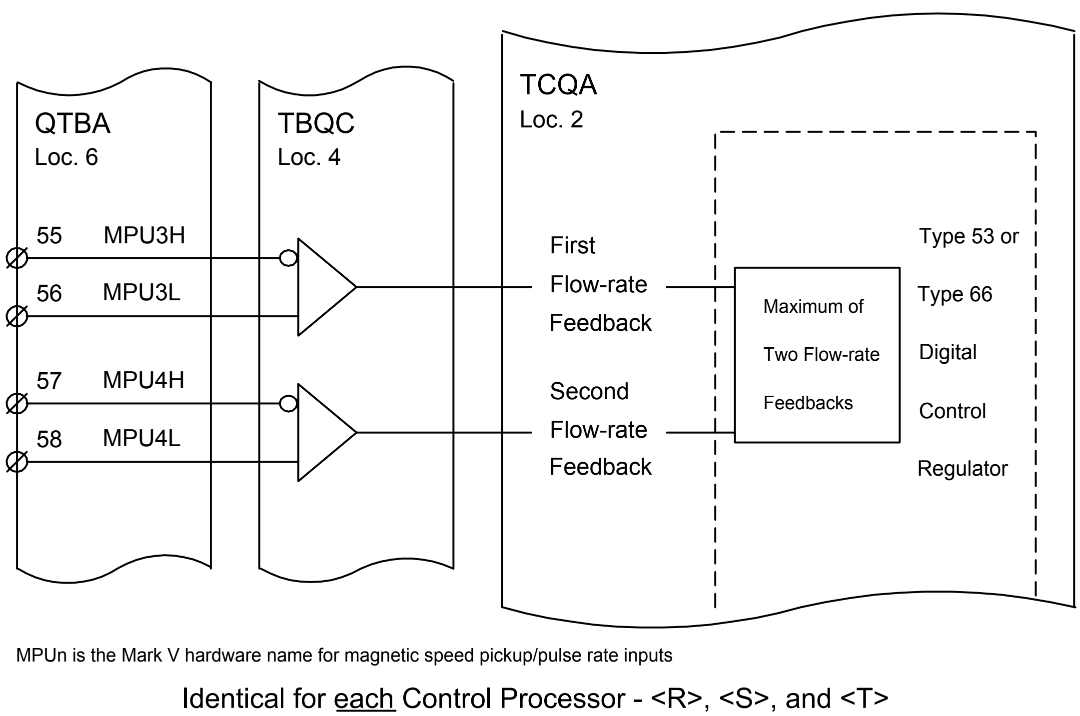
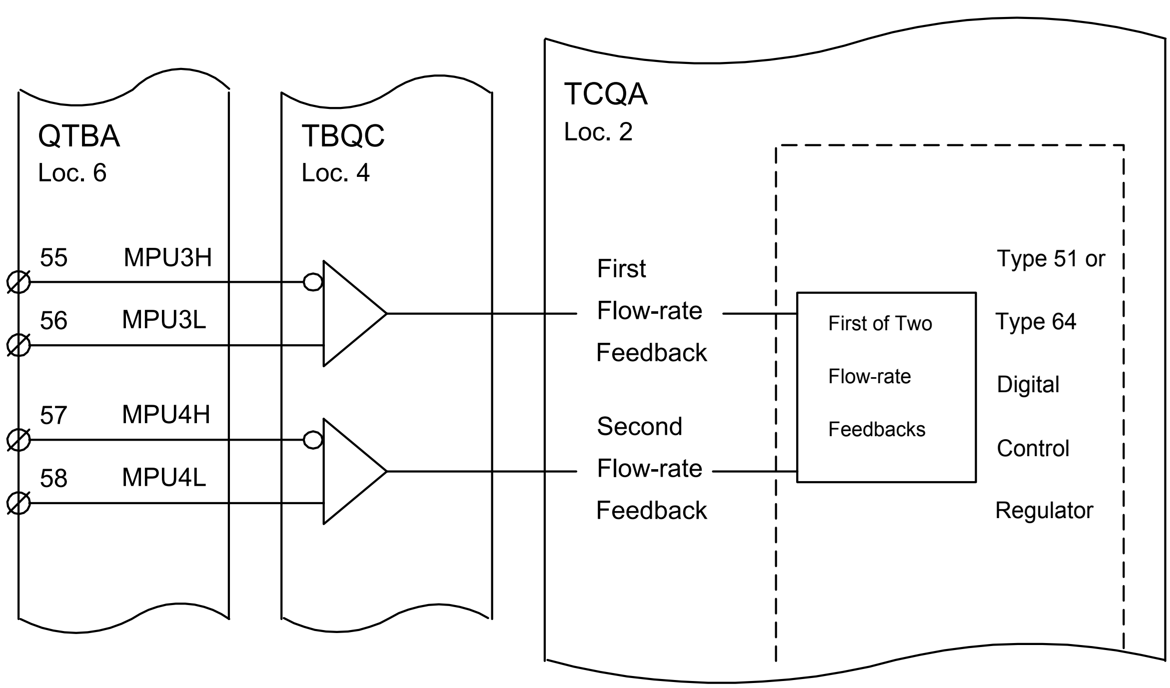


Figure 7-5. Type 53 or 66 Regulator



MPUn is the Mark V hardware name for magnetic speed pickup/pulse rate inputs

Identical for each Control Processor - <R>, <S>, and <T>

Figure 7-6. Type 51 or 64 Regulator

Flow Control Regulator Application Notes

1) A flow control type regulator can usually be defined for any servo-valve output (see Note 3 below). With two flow-rate feedback inputs available, only two servo-valve outputs can be defined with flow control regulators. (In a Simplex control panel using redundant flow-rate feedbacks from the same process (that is, liquid fuel flow divider), only one servo-valve output can be defined with a flow control regulator.)

2) In Simplex applications, flow control regulators are usually defined only on servo-valve outputs 1, 2, 3, or 4 in order to take advantage of the emergency overspeed trip servo current bias function (refer to the signal flow diagrams for the TCEA I/O card).

3) Type 6 flow control regulators cannot usually be defined for servo-valve outputs 1 and 7; refer to the I/O Configurator servo-valve Regulator Definition Screens for a list valid regulator types for each output.

4) Type 5 and 6 flow control regulators require flow feedback pulse rate signal scaling I/O Configuration constants to be entered on a separate I/O Configurator screen.

5) Type 6 flow control regulators require additional I/O Configuration Constants (integrator convergence gain, position reference gain, position reference time constant, and position limits) to be entered at the bottom of the Regulator Definition Screen.

6) Type 6 flow control regulators must have their integrator convergence assigned in IO.ASG.

7) Unused flow control regulator pulse rate inputs may be used for other purposes (for example, display/monitoring/logging, or in a CSP function).

8) If a single pulse rate input is connected to an unused regulator input in a TMR control panel, the input signal must be jumpered to all three input points for proper operation.

7-5.2.2. PRESSURE TRANSDUCER FEEDBACK SIGNALS. Inputs VDC1 and VDC2 are the generic hardware names for 0-10 VDC inputs. They are primarily intended as pressure transducer inputs for use as feedback by pressure control regulators. Because of the "critical" nature of Stop/Ratio Valve control there are three pressure transducer inputs each for VDC1 and VDC2—for individual inputs from redundant transducers to each control processor for those two feedback signals. They are all terminated on the TBQB I/O terminal board in location 9 on the <R> processor (see Figure 7-7). The first of the redundant transducer inputs for VDC1 and VDC2 is communicated to the TCQA I/O card in the <R> control processor only by a dedicated ribbon cable. In a TMR control panel, the second of the redundant inputs for both feedback signals is communicated to the TCQA I/O card in the <S> control processor only by a dedicated ribbon cable, and the third of the redundant inputs is communicated to the TCQA card in the <T> control processor only by a dedicated ribbon cable. Appendix B contains the signal flow diagrams that show this point.

For pressure control regulator feedbacks, VDC1 is defined as the first pressure input and VDC2 the second pressure input. Depending on the regulator function sub-type, the regulator will use either the first pressure input (VDC1), the second pressure input (VDC2), or the maximum feedback signal from both pressure inputs. (Hardware jumpers are provided on the TBQB I/O terminal board to connect a 250 ohm resistor across the signal input terminals if the feedback signal is in milliamps instead of volts.)

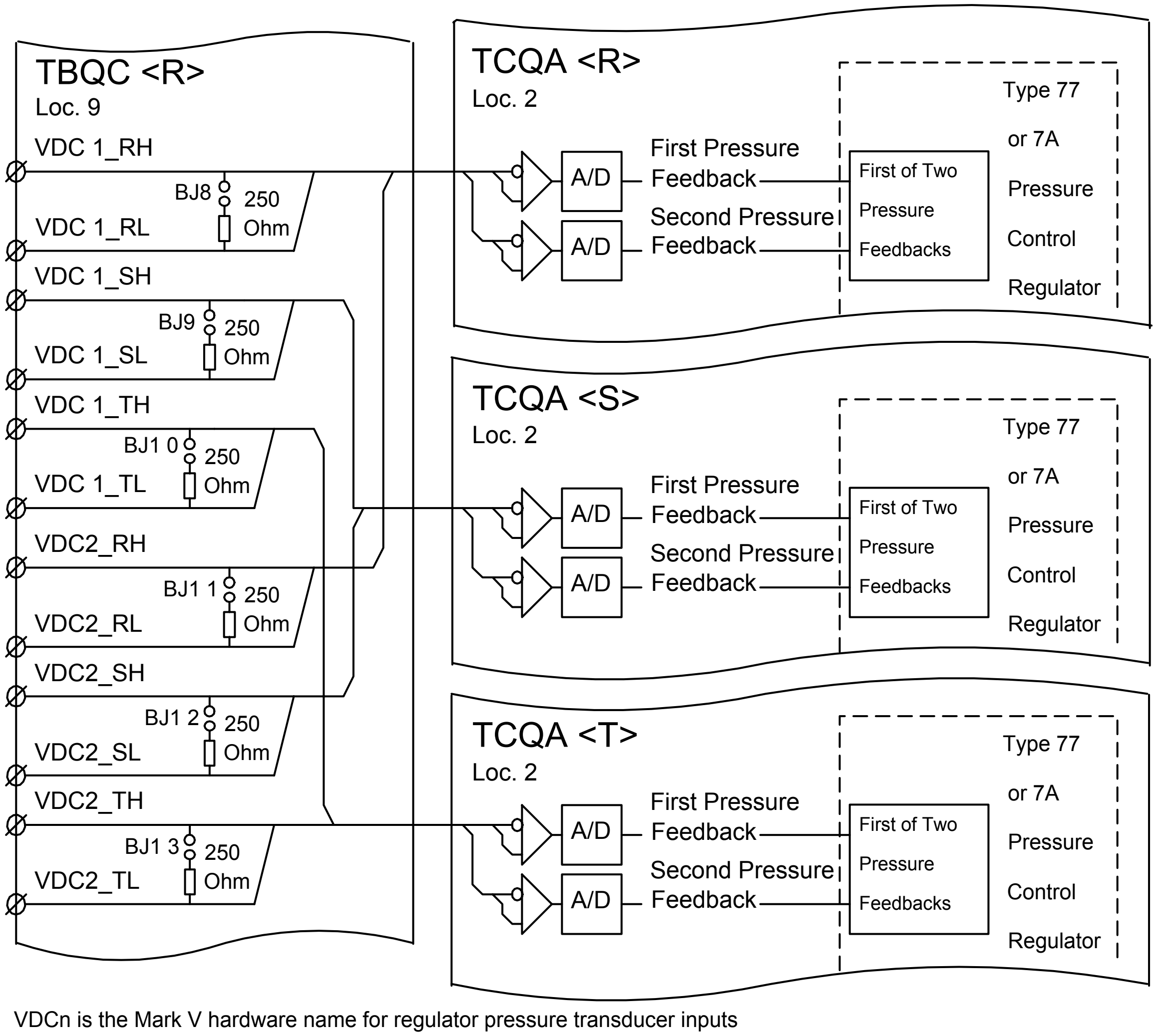


Figure 7-7. Type 77 or 7A Regulator Functions

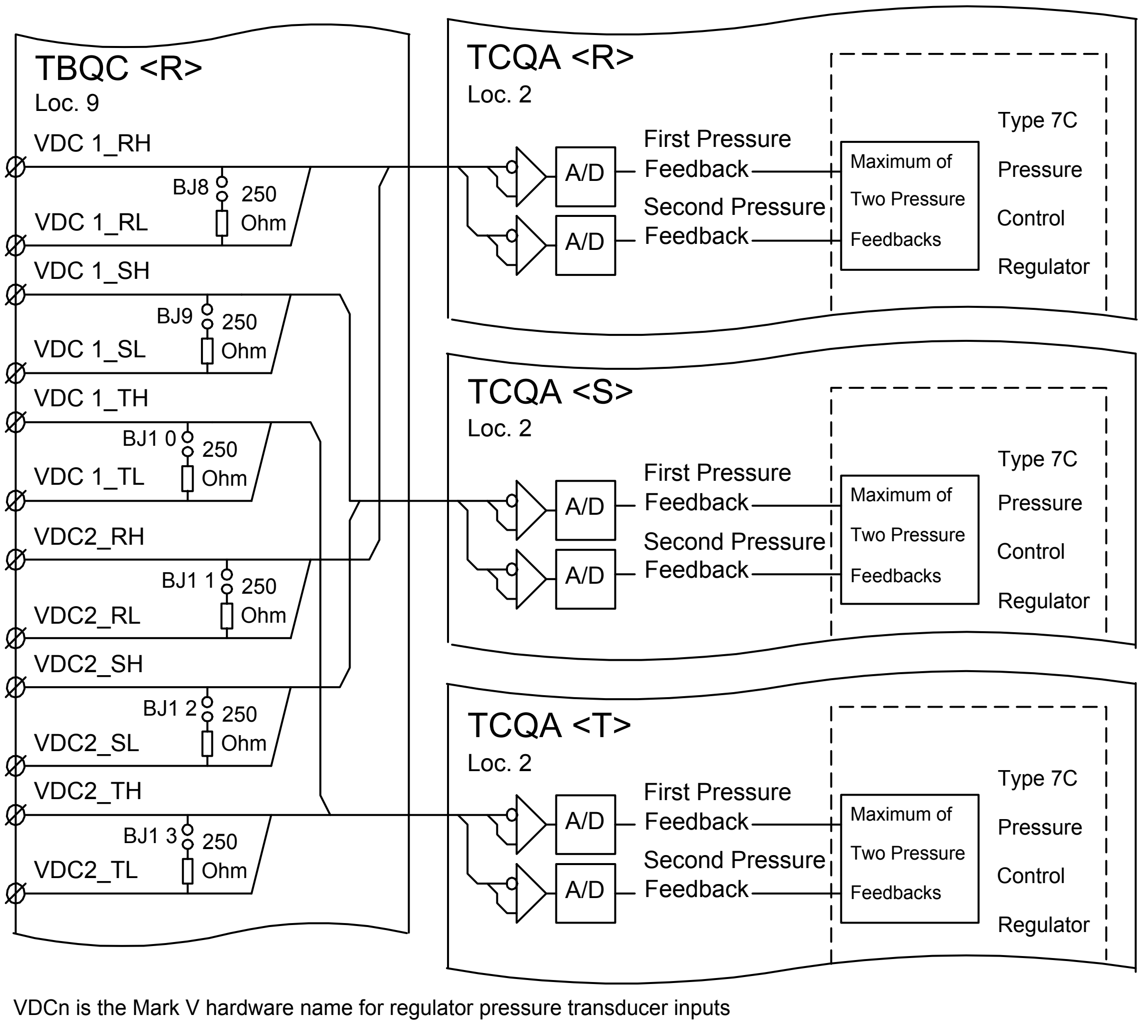


Figure 7-8. Type 7C Regulator Function

Pressure Control Regulator Application Notes

1) Type 7 pressure control regulators can usually only be defined for servo-valve outputs 1 and 7; refer to the I/O Configurator servo-valve output Regulator Definition Screens for a list valid regulator types for each output.

2) Type 7 regulators require additional I/O Configuration Constants (position limits, integrator convergence gain, and position reference gain) to be entered at the bottom of the Regulator Definition Screen. Pressure control regulator feedback signal scaling I/O Configuration Constants must be supplied on a separate screen, also. If the pressure control regulator includes the variable integrator time constant function (also known as the SRV Warmup function), I/O Configuration Constants must also be entered on a separate screen for each Type 7 regulator defined.

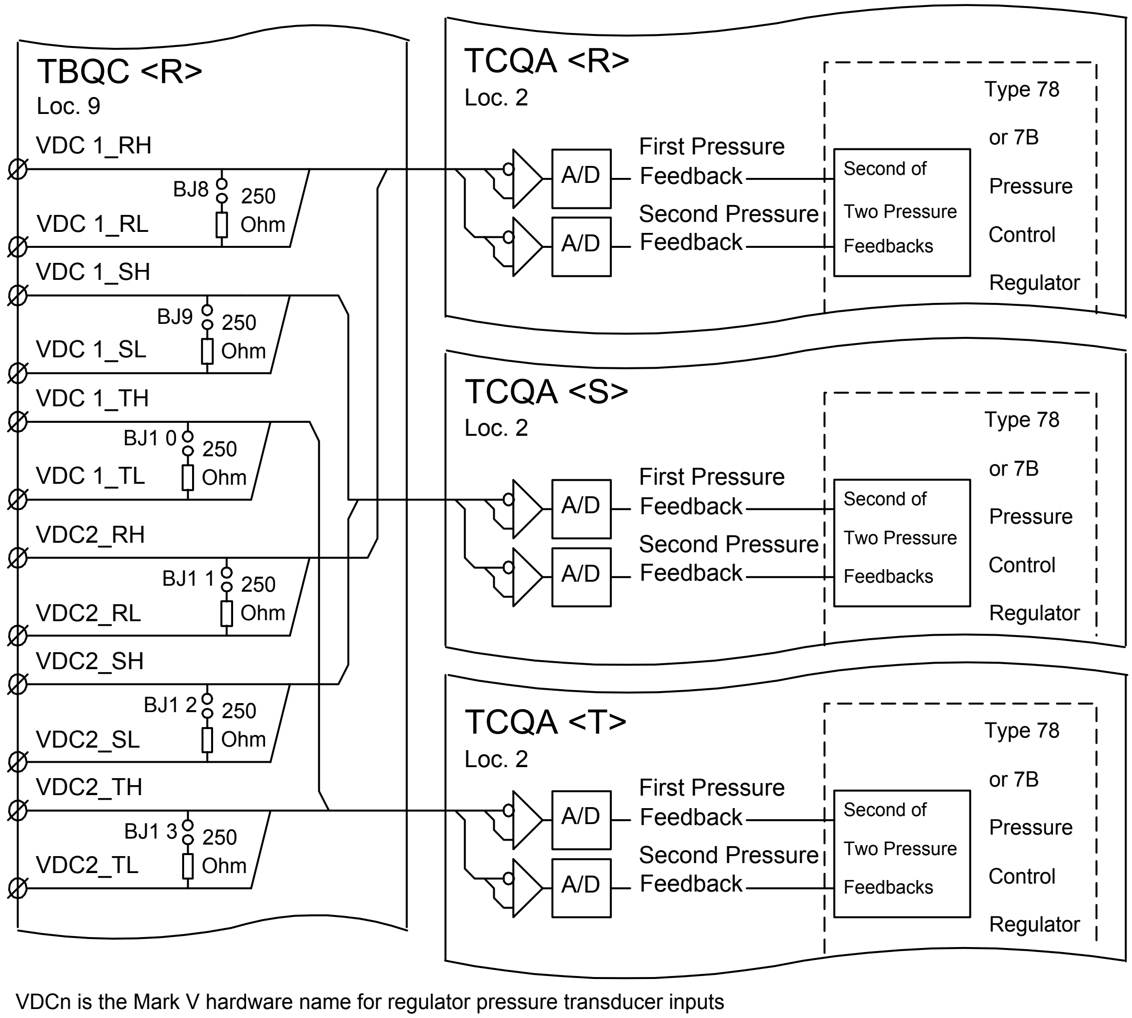


Figure 7-9. Type 78 or 7B Pressure Regulator

3) Unused digital control regulator pressure transducer feedback inputs may be used for other purposes such as display, monitoring, logging, or in a CSP function

4) If a single transducer is connected to an unused digital control regulator input in a TMR control panel the input signal must be jumpered to all three input points for proper operation.

5) Type 7 regulators must have their integrator convergence assigned in IO.ASG.

7-5.2.3. LVDT/LVDR POSITION FEEDBACK SIGNALS. The Mark V handles LVDT/LVDR position feedback inputs in two different methods. One way to determine which method is employed by a particular Mark V is to check which I/O terminal board is in location 9 in the <R> core. TBQC I/O terminal boards in gas turbine and some medium steam turbine applications share (connect in parallel) each of the 16 position feedback signals to all three control processors. TBQF I/O terminal boards in large steam turbine and some medium steam turbine applications wire 27 of the 29 position feedback signals directly to individual control processors. The remaining 2 position feedback signals are "shared" by their assigned servo-valve output regulators in each control processor. Both methods, shared (connected in parallel) and individual position feedback, are described below. (Some sites which have both types of turbines, gas and steam, may have both position feedback methods. When troubleshooting, it is important to know which I/O terminal board is in use.)

Shared Position Feedback Assignments for All Regulators using Position Feedback except Type 49 and Type 2E	
Position Feedback Input Signal	Servo-valve Output
POS01 POS02	1
POS03 POS04	2
POS05 POS06	3
POS07 POS08	4
POS09 POS10	5
POS11 POS12	6
POS13 POS14	7
POS15 POS16	8

Shared Position Feedback Assignments for Type 49 Regulators	
Position Feedback Input Signal	Servo-valve Output
POS01 POS02 POS16	1
POS03 POS04 POS15	2
POS05 POS06 POS14	3
POS07 POS08 POS13	4
POS09 POS10 POS12	5
POS11	6
N/A	7
N/A	8

Shared Position Feedback Assignments for Type 2E Regulators	
Position Feedback Input Signal	Servo-valve Output
POS01 POS02 POS09 POS10	1
POS03 POS04 POS11 POS12	2
POS05 POS06 POS13 POS14	3
POS07 POS08 POS15 POS16	4
N/A	5
N/A	6
N/A	7
N/A	8

POSnn is the Mark V hardware name (silk screened on the card) for LVDT/LVDR feedback signals

Table 7-8. Shared LVDT/LVDR Position Feedback Input Assignments

Shared Position Feedback Signals

In a Mark V control panel with shared position feedback signals, all of the LVDT/LVDR feedback signals are connected in parallel to the TCQA I/O card in each control processor. Each of the sixteen LVDT/LVDR position feedback signals (<R> I/O terminal board TBQC, terminals 1 through 32, called POS01 through POS16, respectively) is assigned for use by the regulator that is defined for a specific servo-valve output. The regulator sub-type number/letter for a particular servo-valve output defines the type of shared position feedback signal assignment, and cannot be modified. The relationship between the feedback signal and the servo-valve are outlined in the tables above. Depending on the regulator function type and sub-type, a position regulator will use either the greater of two, the minimum of two, median of three, or only the first assigned LVDT/LVDR feedback input.

Refer to Figure 7-10 for an example of shared LVDT/LVDR position feedback inputs to a Type 43 regulator function defined for servo-valve output #4.

Shared Position Feedback Application Notes

1) Caution must be exercised when Type 49 regulator functions have been defined. When a Type 49 regulator function is defined for servo-valve outputs 1, 2, 3, 4, or 5, its third LVDT/LVDR feedback input will be of the feedback inputs normally assigned to one of the other servo-valve outputs (see the tables above). Thus only five Type 49 regulators can be defined in one Mark V with shared position feedback inputs. Additionally, if a Type 49 regulator function is defined for servo-valve outputs 2 or 4, regulator functions defined for servo-valve outputs 7 or 8 must not require position feedback.

2) Only if a regulator does not require position feedback or only the first assigned feedback is used, the second position feedback input is available for display or CSP functions.

Individual Position Feedback Signals

In a Mark V with individual position feedback signals, 27 of the possible 29 LVDT/LVDR feedback signals are connected to only one of the three control processors, although all LVDT/LVDR feedback signals terminate in the <R> core on the TBQF terminal board. The remaining two LVDT/LVDR feedback signals are connected in parallel to all three control processors. The following chart gives the TBQF terminal screws for the position feedback signals.

Note the "numbering" of the individual position feedback signal inputs; they were numbered to maintain symmetry with the shared position feedback signal inputs (POS1 & -2 for servo-valve output #1, and so on). Note also that the "second" assigned LVDT/LVDR feedback signal input for servo-valve outputs 1, 4, 5, 6, 7, & 8 is shared by all three control processors in a TMR control panel. Additionally, both LVDT/LVDR feedback signals for servo-valve output #3 are shared by all three control processors in a TMR control panel. Servo-valve output #2 has only one LVDT/LVDR feedback signal.

An example of a Type 41 position control defined for servo-valve output #1 is shown in Figure 7-11.

Individual Position Feedback Application Notes

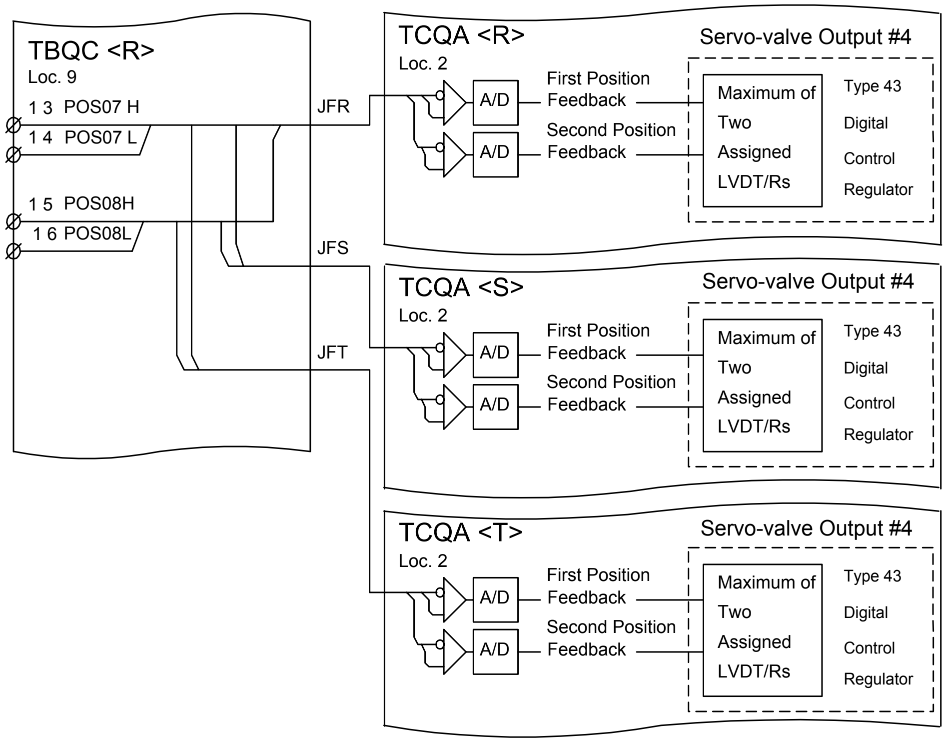
1) Only seven Type 41 regulators with individual position feedback signals can be defined; servo-valve output #3 does not have three individual position feedback signal inputs.

2) Only six Type 43 or 44 regulators with individual position feedback signals can be defined; servo-valve output #2 does not have a second assigned position feedback and servo-valve output #3 has no individual position feedback signals.

3) Unused position feedback signal inputs can be used for other purposes (display/monitoring, and so on).

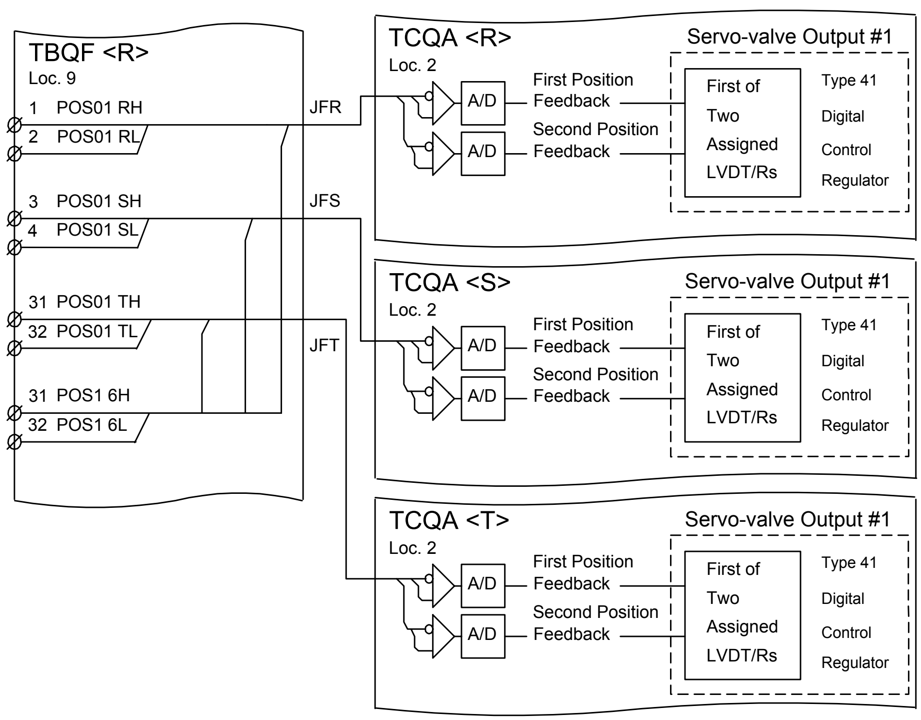
Individual Position Feedback Assignments		
Mark V name for the Position Feedback Signal	Servo-valve Output	TBQF Term. Screws
LVDR1R/POS1R	<R> 1	1, 2
LVDR1S/POS1S	<S> 1	3, 4
LVDR1T/POS1T	<T> 1	5, 6
LVDR2	<R>, <S>, <T> 1	7, 8
LVDR3R/POS3R	<R> 2	9, 10
LVDR3S/POS3S	<S> 2	11, 12
LVDR3T/POS3T	<T> 2	13, 14
LVDR5/POS5	<R>, <S>, <T> 3	15, 16
LVDR6/POS6	<R>, <S>, <T> 3	17, 18
LVDR7R/POS7R	<R> 4	19, 20
LVDR7S/POS7S	<S> 4	21, 22
LVDR7T/POS7T	<T> 4	23, 24
LVDR8	<R>, <S>, <T> 4	25, 26
LVDR9R/POS9R	<R> 5	27, 28
LVDR9S/POS9S	<S> 5	29, 30
LVDR9T/POS9T	<T> 5	31, 32
LVDR10	<R>, <S>, <T> 5	33, 34
LVDR11R/POS11R	<R> 6	85, 86
LVDR11S/POS11S	<S> 6	87, 88
LVDR11T/POS11T	<T> 6	89, 90
LVDR12	<R>, <S>, <T> 6	91, 92
LVDR13R/POS13R	<R> 7	93, 94
LVDR13S/POS13S	<S> 7	95, 96
LVDR13T/POS13T	<T> 7	97, 98
LVDR14	<R>, <S>, <T> 7	99, 100
LVDR15R/POS15R	<R> 8	101, 102
LVDR15S/POS15S	<S> 8	103, 104
LVDR15T/POS15T	<T> 8	105, 106
LVDR16	<R>, <S>, <T> 8	107, 108

Table 7-9. Individual LVDT/LVDR Position Feedback Assignments



POSnn is the Mark V hardware name for shared LVDT/LVDR position feedback inputs

Figure 7-10 Shared LVDT/LVDR Position Feedback Signal - Type 43



POSnn is the Mark V hardware name for individual LVDT/LVDR position feedback inputs

Figure 7-11 Individual LVDT/LVDR Position Feedback Signals - Type 41

7-5.3. Digital Control Regulator Functions

The reference to these regulators is calculated in the CSP, written into the control signal database (CDB) by the control processor, and read from the database by the digital regulator. The CSP often has longer time constant outer loops (megawatt, temperature, mass flow or torque for example), so any inner loop disturbances will be cancelled out by the outer loop(s).

The feedback signals, LVDT/LVDR, pulses, or analog voltages/currents are converted from an analog signal to a proportional digital value by the TCQA card. The regulator then compares this digital value of feedback to a digital reference to calculate the output (or error signal). As discussed above, the feedback device is selected via the regulator function sub-type.

The digital output of these regulators is converted to a current to drive the servo-valve by digital to analog converters and amplifiers. The range or magnitude of the actual servo-valve milliamp output is set using hardware jumpers on the TCQC card in the control processor.

Type 6 and 7 regulators have integrating outer loops to increase the steady state gain and reduce error. Adjustments are provided for rate of integration or lead time constant, depending upon the application.

7-5.3.1. REGULATOR FUNCTION TYPE 2 (Position control with cylinder and pilot valve position feedback) This regulator function type is generally only used on GE-Fitchburg steam turbines which have low-pressure hydraulic servo-valves and actuators with either one or two LVDT/LVDR(s) on the servo-valve pilot piston and either one or two LVDT/LVDR(s) on the cylinder/actuator. Type 2 regulators (see Figure 7-12.) are two proportional (or straight gain) position regulator loops in series, with a convergence function for TMR applications (to keep the output signals of all three processors equal) and a lag (time delay) in the output signal. This lag helps to prevent the system from over compensating in the event of a rapid feedback signal change. SIMPLEX applications will have the integrator convergence gain set to zero. The outer loop is the actuator position regulator with the reference calculated in the CSP as described above. The output of the actuator position regulator is the reference to the inner loop pilot valve position regulator. The output of the convergence lag circuit is added or subtracted from the pilot valve loop output, which becomes the digital control regulator’s output signal.

7-5.3.2. REGULATOR FUNCTION TYPE 4 (Position control with position feedback). The type 4 regulator functions (see Figure 7-13) use position feedback signals from one or more LVDT/LVDRs (with the exception of type 40). These are proportional or straight gain regulators that calculate the difference between the CSP calculated reference and the feedback to determine a position error. The position error signal is then multiplied by a regulator gain. A current bias, which can be used to overcome the mechanical bias of the servo-valve's fail-safe spring, is then added to the position error signal, which becomes the digital control regulator's output signal.

7-5.3.3. REGULATOR FUNCTION TYPE 5 (Flow control with flow-rate feedback). The type 5 regulator functions (see Figure 7-14) use flow-rate feedback signals from one or more speed pickups (usually magnetic). These are proportional or straight gain regulators that calculate the difference between the CSP calculated reference and the feedback to determine a flow-rate error. This error signal is then multiplied by a regulator gain. A current bias, which can be used to overcome the mechanical bias of the servo-valve's fail-safe spring, is then added to the error signal, which becomes the digital control regulator's output signal.

7-5.3.4. REGULATOR FUNCTION TYPE 6 (Flow control with flow-rate and position feedback). The Type 6 regulator functions (see Figure 7-15) use flow-rate feedback signals from one or more speed pickups (usually magnetic) and position feedback signals from one or two assigned LVDT/LVDRs for improved stability. The outer loop is a proportional plus integral flow-rate regulator, comparing the flow-rate reference from the CSP to a flow-rate feedback defined by the regulator sub-type. A convergence correction is added to the error signal for use in TMR applications to help balance the three regulators. The output of the flow-rate loop is the reference to the inner straight gain actuator position loop. Simplex applications will have the integrator convergence gain set to zero. The output of the actuator loop is then multiplied by a regulator gain. A current bias, which can be used to overcome the mechanical bias of the servo-valve's fail-safe spring, is then added to the digital output signal.

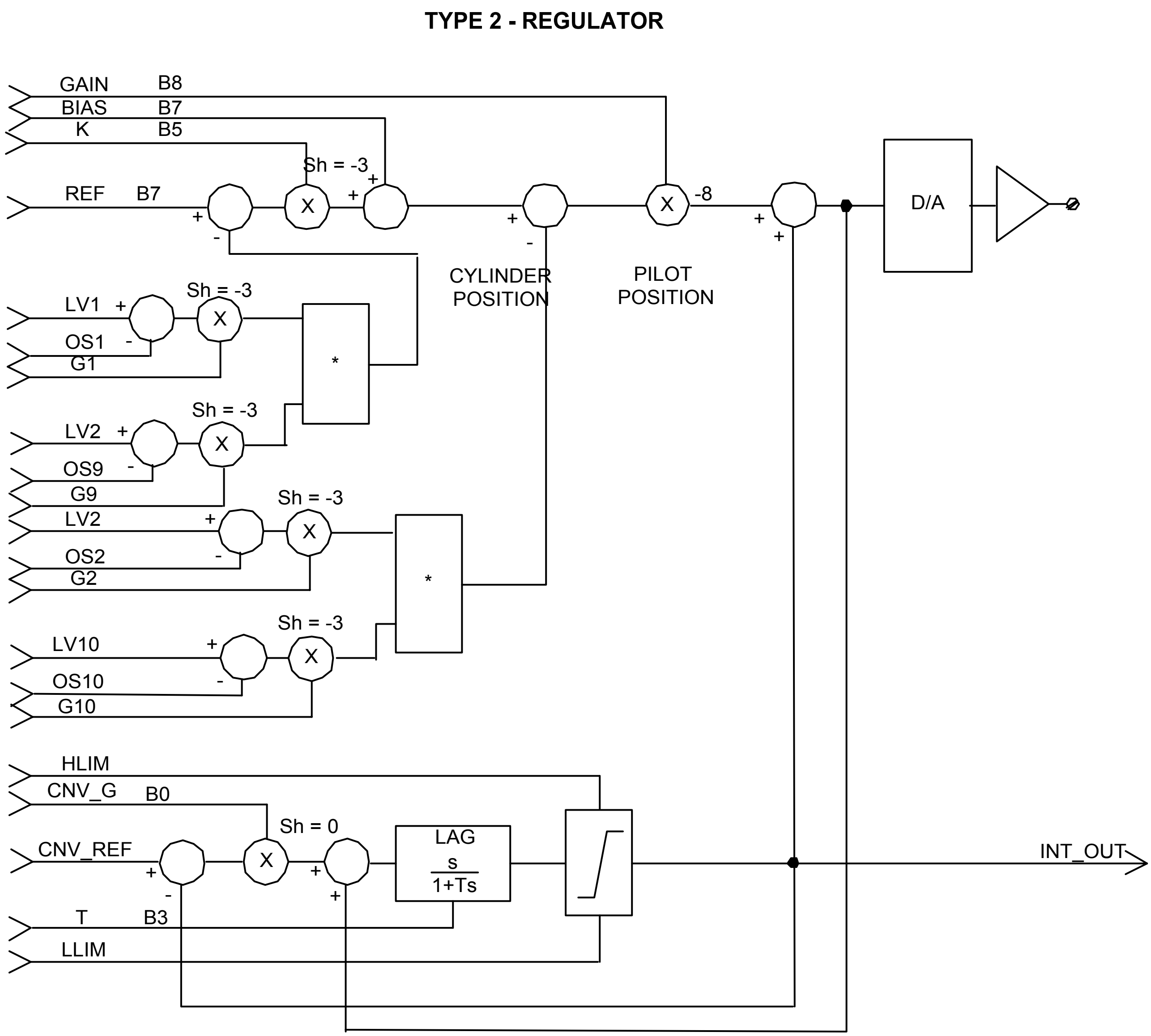
7-5.3.5. REGULATOR FUNCTION TYPE 7 (Pressure control with pressure and position feedback). The Type 7 regulator function (see Figure 7-16) uses a pressure feedback signal from one of the two possible pressure transducer inputs and position feedback signals from one or two assigned LVDT/LVDRs for improved stability. The outer loop is an integrating pressure regulator plus lead circuit which provides a high gain for high frequencies and a lower gain for lower frequencies. A convergence correction is added to the error signal for use in TMR applications to help balance the three regulators. Simplex applications will have the integrator convergence gain set to zero. The output of the pressure regulator is the reference to the straight gain valve position regulator. The position error signal is then multiplied by a regulator gain. A current bias, which can be used to overcome the mechanical bias of the servo-valve's fail-safe spring, is then added to the digital error signal, which becomes the digital control regulator's output signal.

WARNING

The regulators defined above are carefully adjusted during the start-up phase. The digital regulators will not drift or change with time. No further adjustment should be necessary. Incorrect adjustment could have detrimental effects to system performance, possibly resulting in unstable or oscillatory conditions. Do not adjust the control constants or I/O configurator settings without concurrence from GEDS or the turbine vendor.

7-6. EXAMPLES OF REGULATOR APPLICATIONS

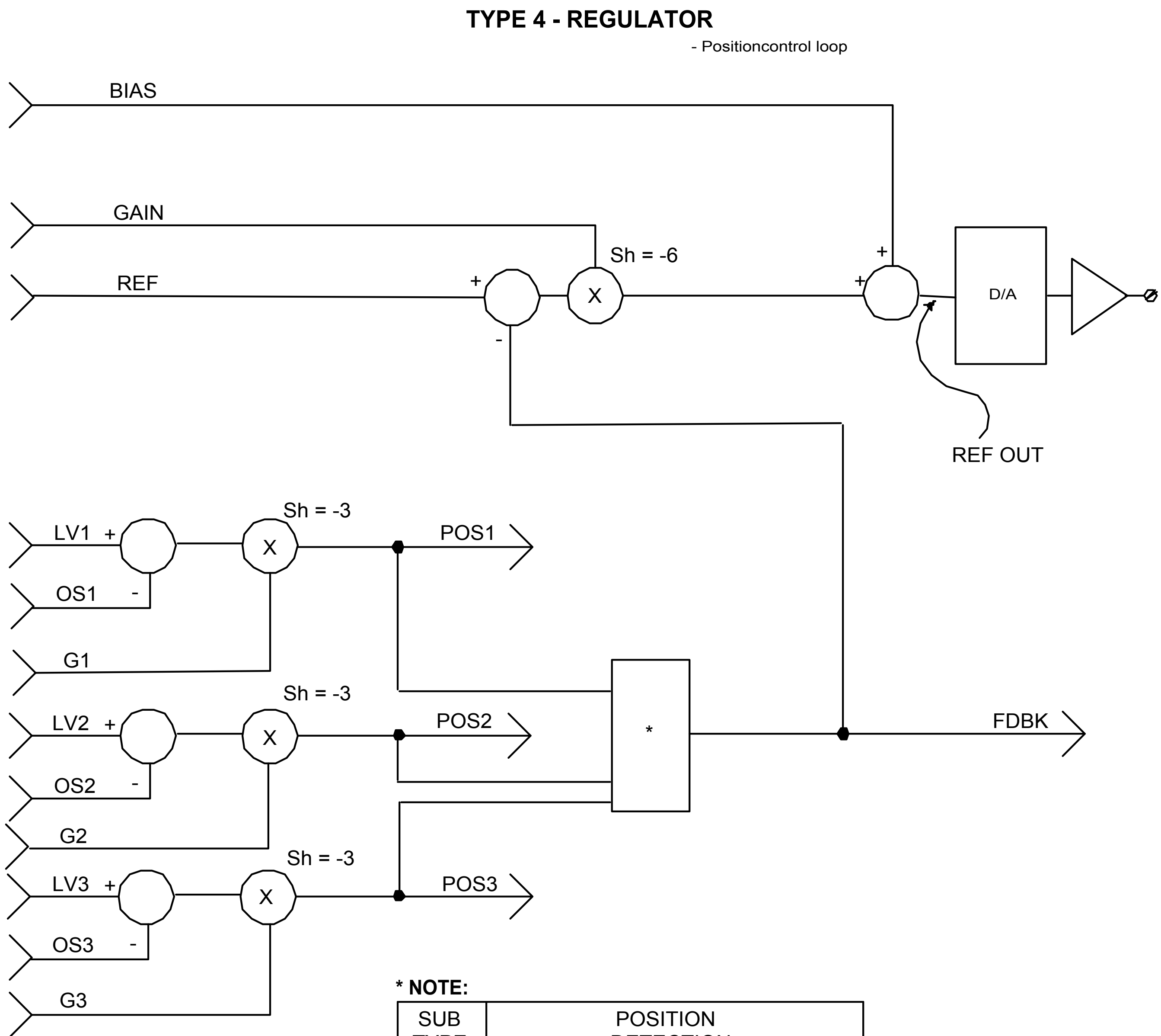
Figures 7-17 to 7-24 depict typical applications of the most common digital control regulators in a Mark V turbine control system, showing feedback signals, servo-valve outputs, and associated hardware (Mark V I/O terminal boards, Mark V I/O cards, servo-valves, actuators, and feedback devices). The diagrams represent typical applications only, and are not site- or unit-specific.



*** NOTE:**

SUB TYPE	POSITION DETECTION
D	Cylinder - first assigned LVDT/R Pilot - second assigned LVDT/R
E	Cylinder - max of the first pair of assigned LVDT/Rs Pilot - max of second pair of assigned LVDT/R

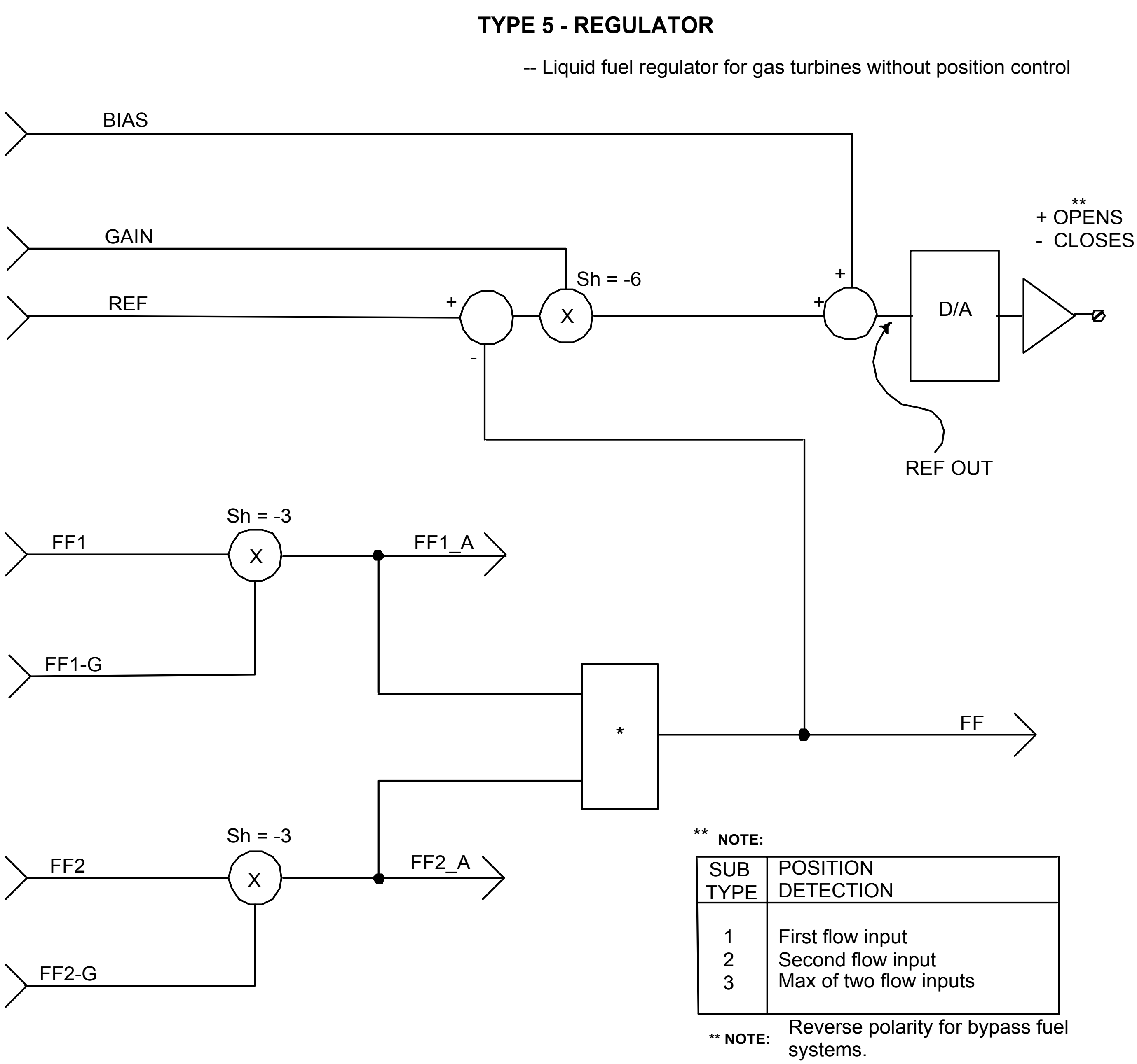
Figure 7-12. Type 2 Regulator Signal Flow Diagram



*** NOTE:**

SUB TYPE	POSITION DETECTION
0	No position feedback
1	First assigned LVDT/R
2	Second assigned LVDT/R
3	Max of two assigned LVDT/Rs
9	Median of three assigned LVDT/Rs

Figure 7-13. Type 4 Regulator Signal Flow Diagram

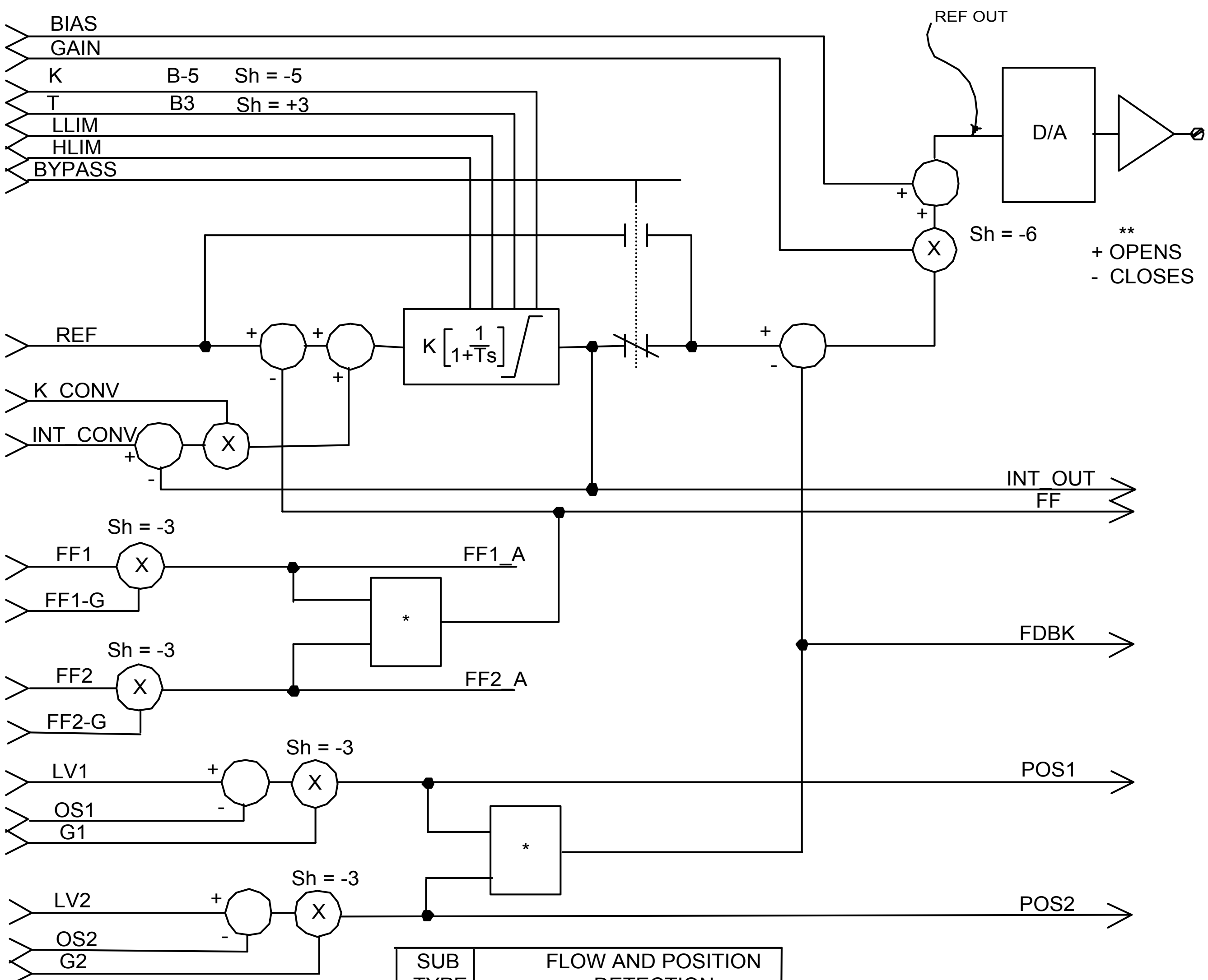


SUB TYPE	POSITION DETECTION
1	First flow input
2	Second flow input
3	Max of two flow inputs

Figure 7-14. Type 5 Regulator Signal Flow Diagram

TYPE 6 REGULATOR

- Liquid fuel regulator with position loop



* **NOTE:**

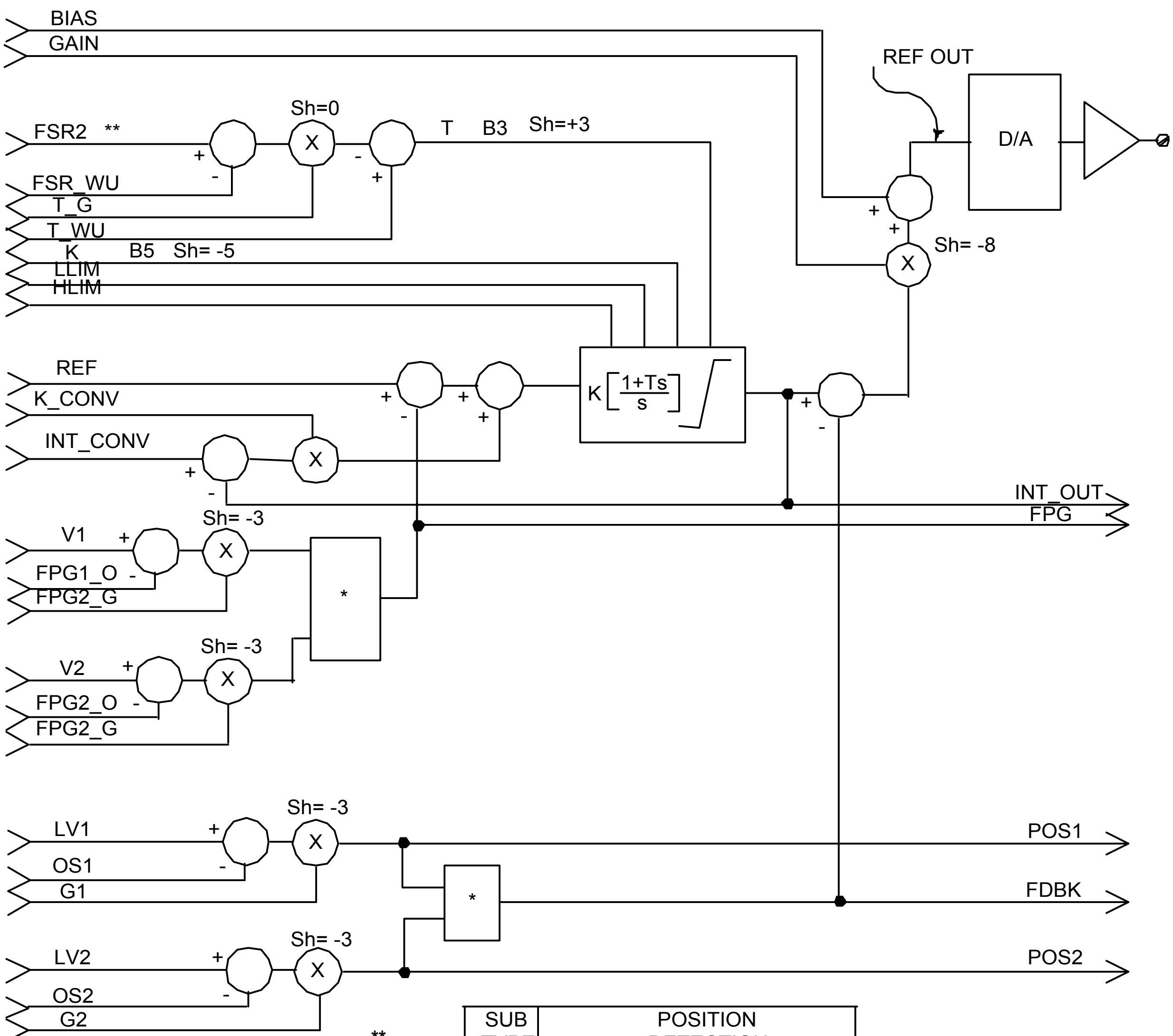
SUB TYPE	FLOW AND POSITION DETECTION
4	First flow input and max of 2 assigned LVDT/Rs
5	Second flow input and max of 2 assigned LVDT/Rs
6	Max of both flow inputs and max of 2 assigned LVDT/Rs

** **NOTE:** Reverse polarity for bypass fuel systems.

Figure 7-15. Type 6 Regulator Signal Flow Diagram

TYPE 7 - REGULATOR

- Speed ratio stop valve



** **NOTE:**

SUB TYPE	POSITION DETECTION
7	First pressure input and max of 2 assigned LVDT/Rs
8	Second pressure input and max of 2 assigned LVDT/Rs
A	First pressure input and first assigned LVDT/R
B	Second pressure input and first assigned LVDT/R
C	Max of 2 pressure inputs and max of 2 assigned LVDT/Rs

** **NOTE:** This variable is the reference to the corresponding Gas Control Regulator, refer to configuration tables.

Figure 7-16. Type 7 Regulator Signal Flow Diagram

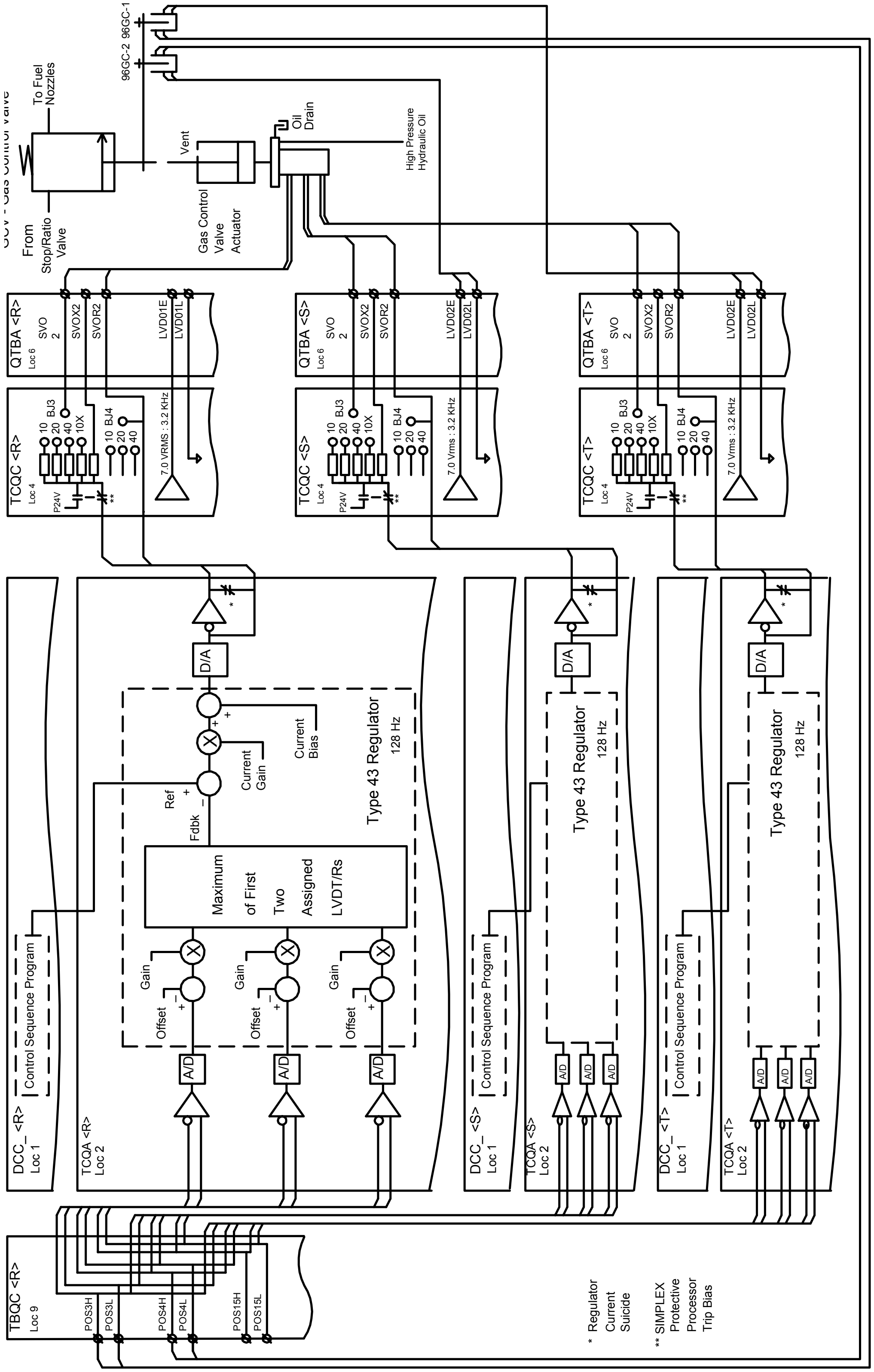


Figure 7-17 Servo-Valve Output #3 with a Type 43 Regulator

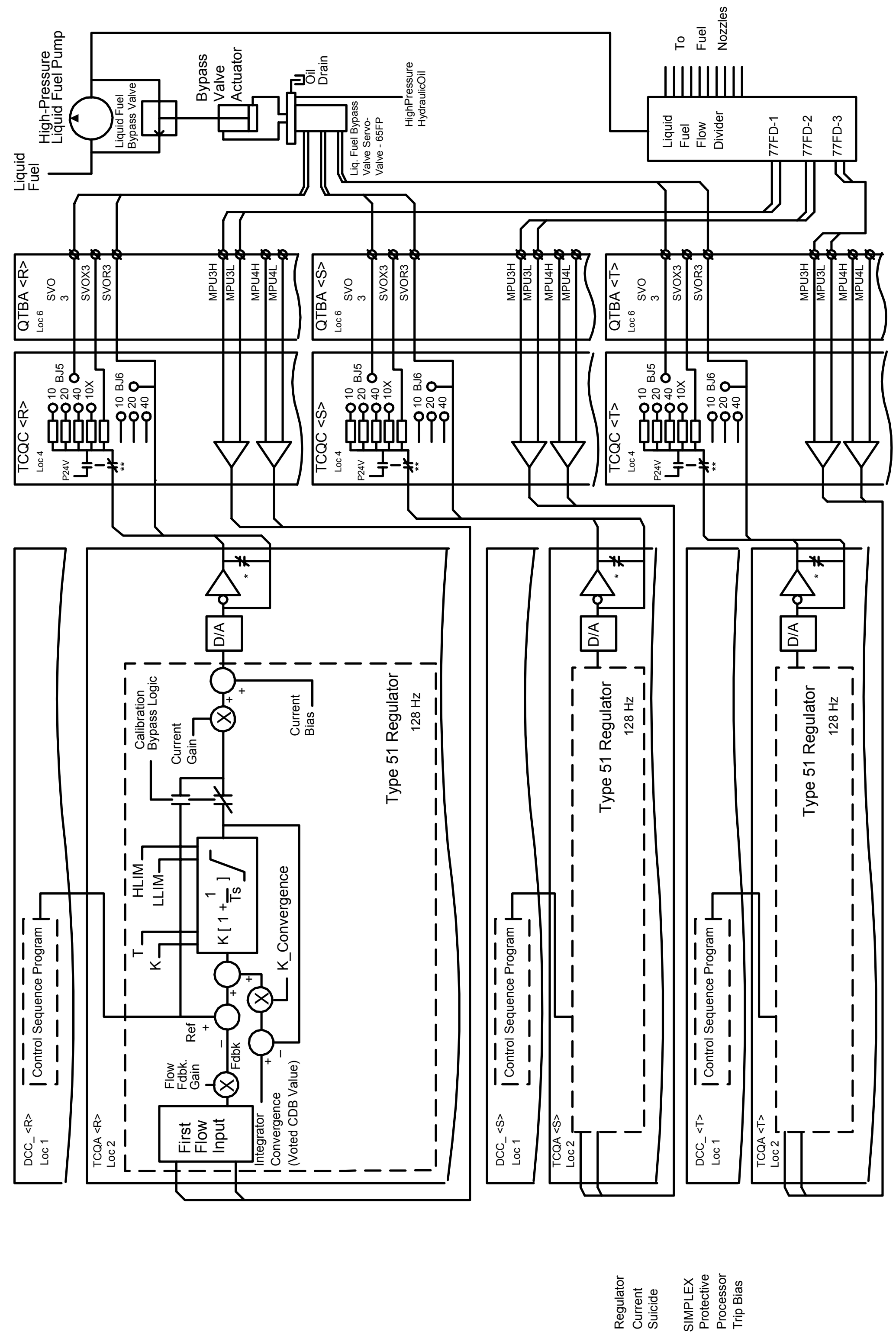


Figure 7-18 Servo-Valve Output #3 with a Type 51 Regulator

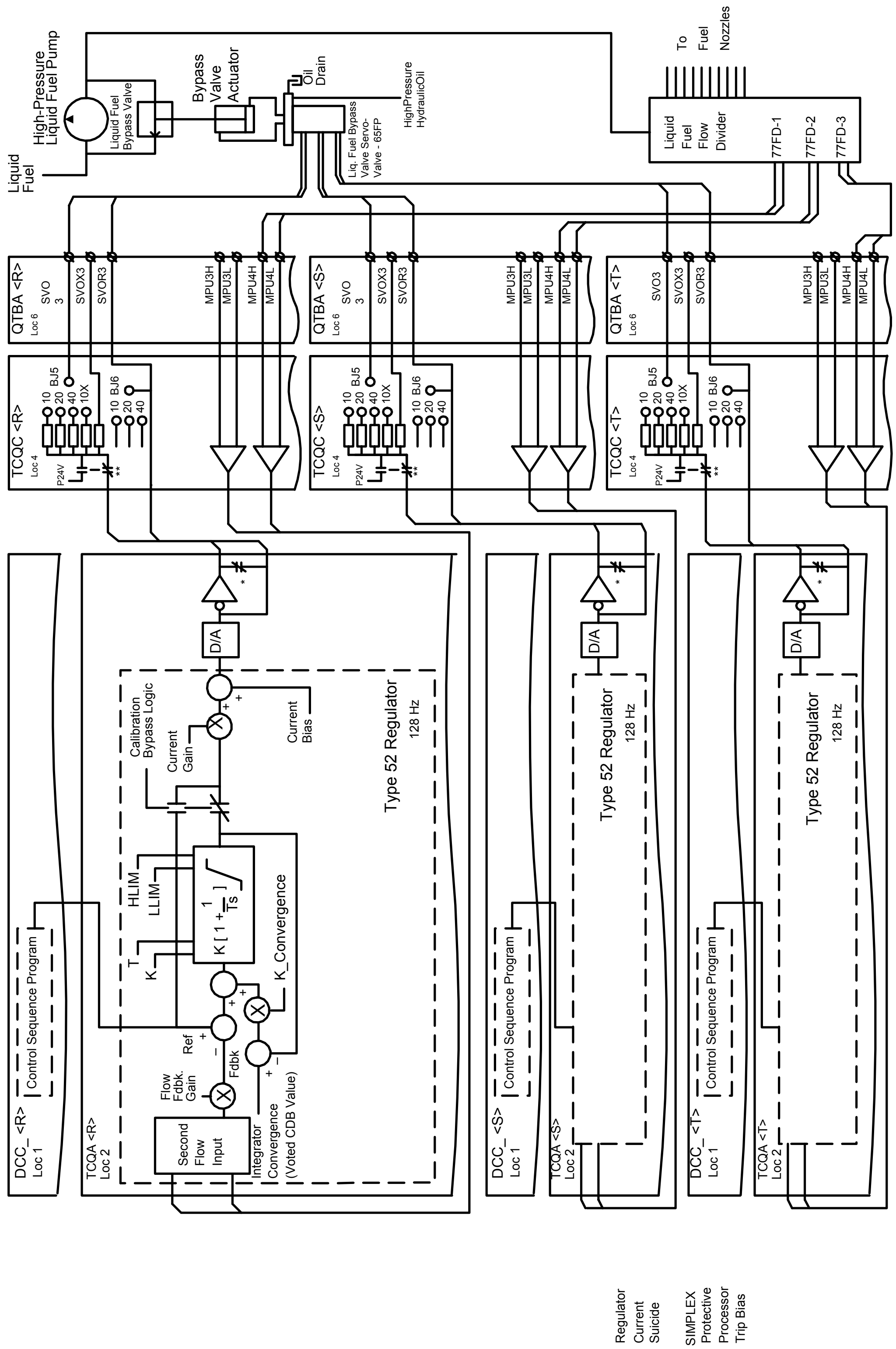


Figure 7-19 Servo-Valve Output #3 with a Type 52 Regulator

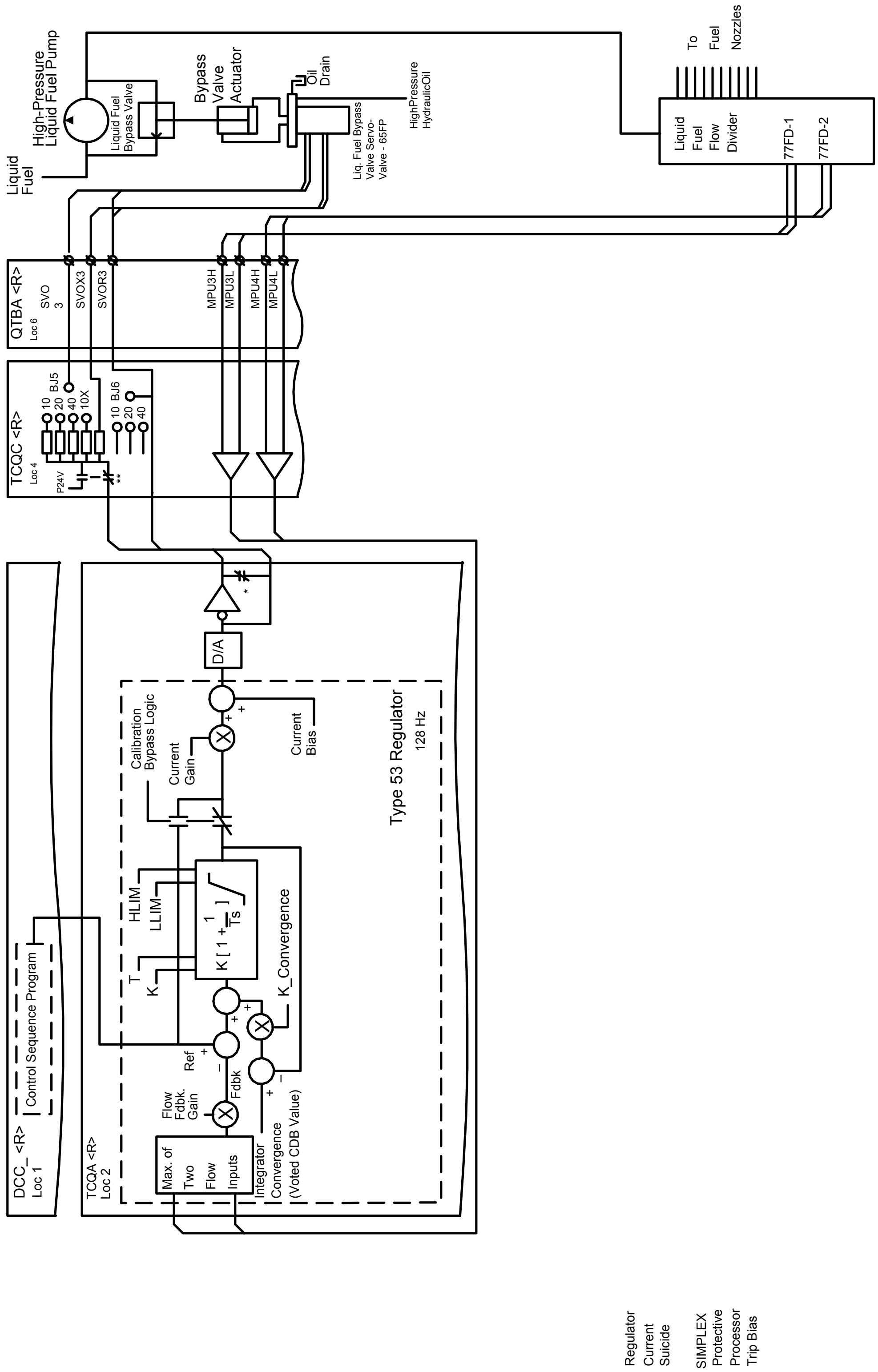


Figure 7-20 Servo-Valve Output #3 with a Type 53 Regulator

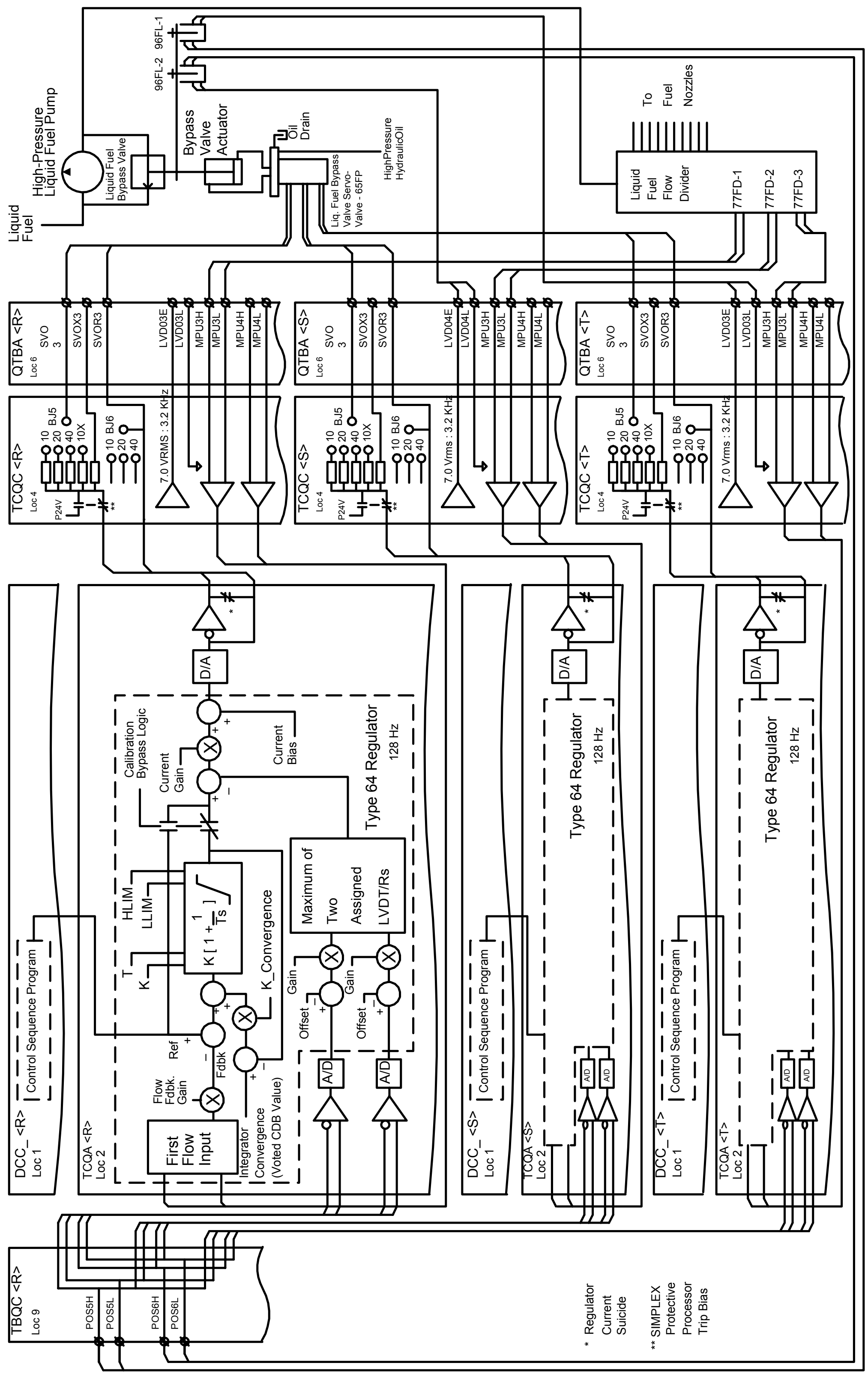


Figure 7-21 Servo-Valve Output #3 with a Type 64 Regulator

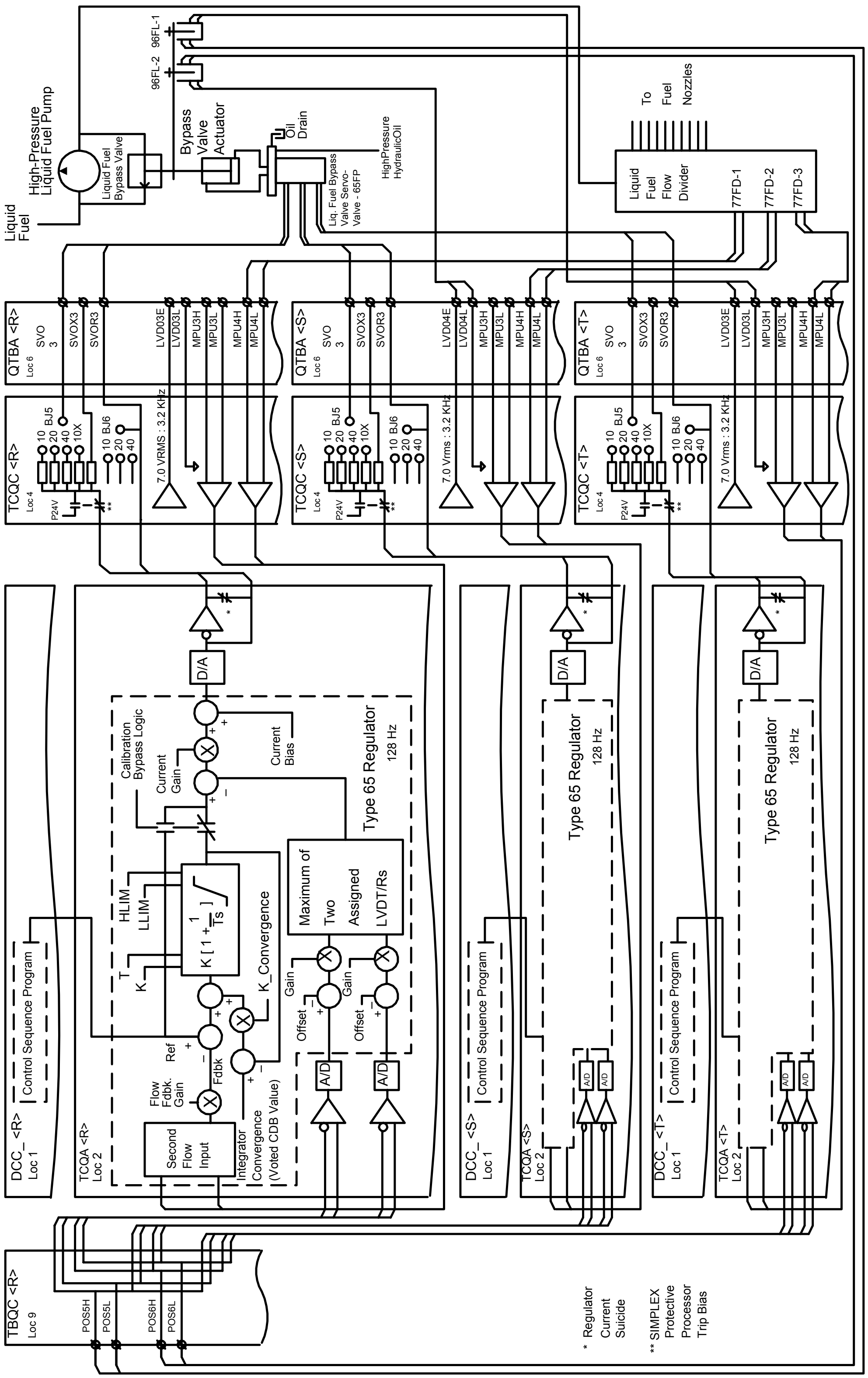


Figure 7-22 Servo-Valve Output #3 with a Type 65 Regulator

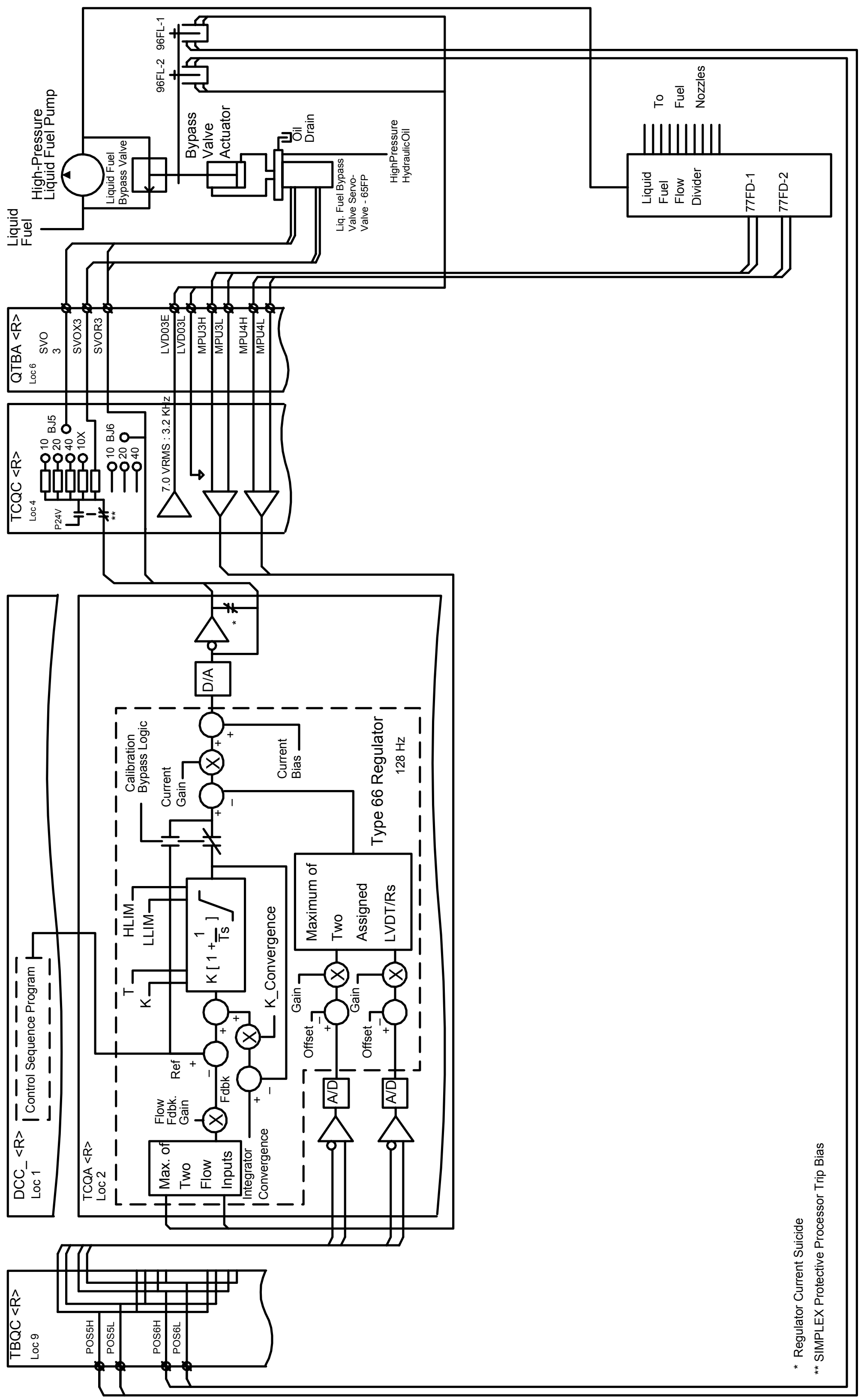


Figure 7-23 Servo-Valve Output #3 with a Type 66 Regulator

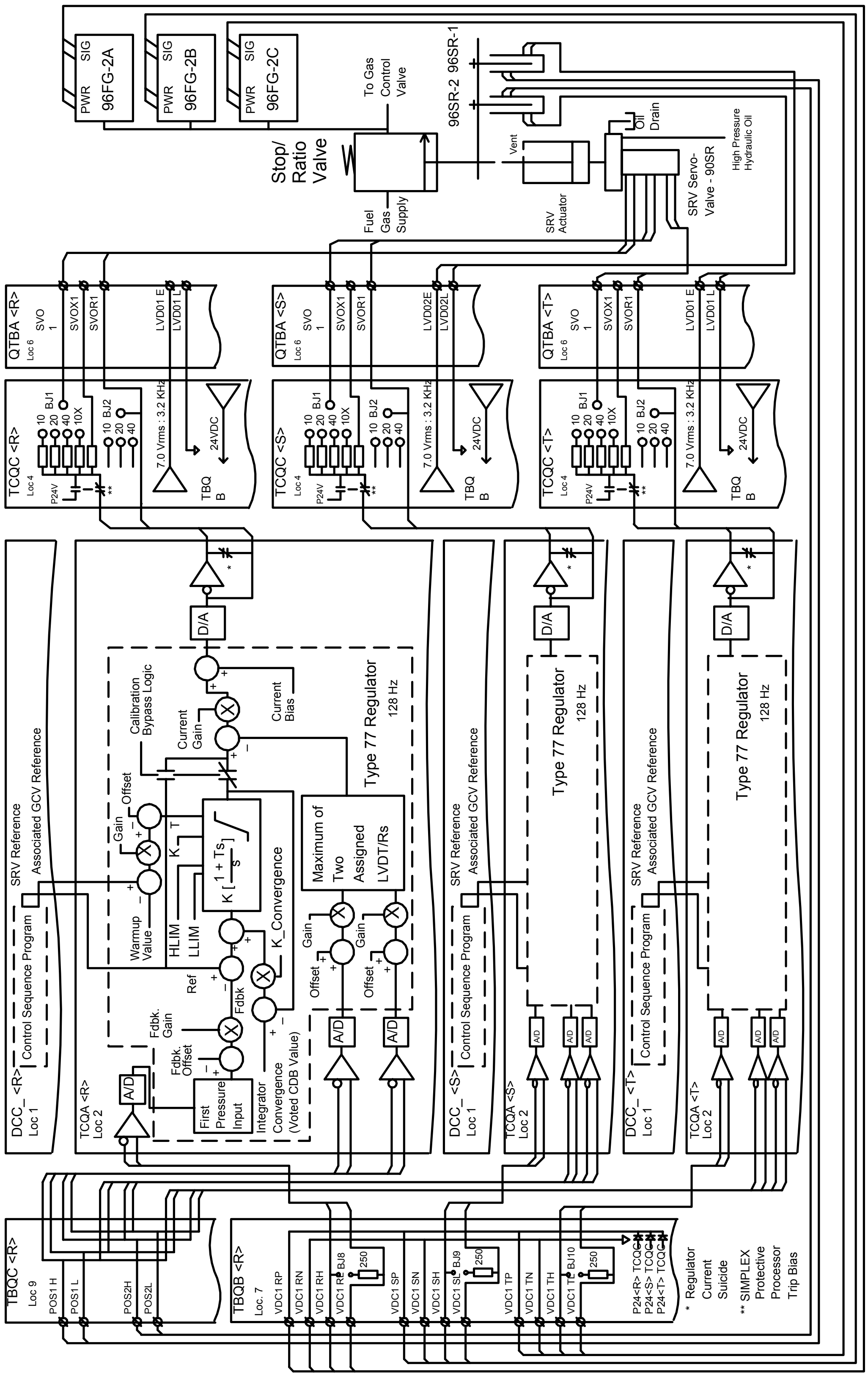


Figure 7-24 Servo-Valve Output #1 with a Type 77 Regulator

7-7. DEFINING REGULATORS IN THE I/O CONFIGURATOR

Regulator functions are defined for servo-valve outputs using the I/O Configurator. To define a regulator for a servo-valve output or to modify the values (I/O Configuration Constants) used by a regulator, the TCQA Card Definition Screens of the <Q> Configuration Menu of the I/O Configurator must be used.

Figure 7-25 shows a typical I/O Configurator Regulator Definition Screen (see Appendix B) for servo-valve output #6. Regulator Definition Screens can be divided into four regions. The first region includes the function type and sub-type field and a value must be entered in this field for all servo-valve outputs. The second region includes the suicide enable and current bias and gain fields. If a servo-valve output is being used, (that is, a value other than 00 has been entered for the function type & sub-type) values must be entered in these fields. The third region includes the zero and 100% stroke voltage values used to scale the first two of the possible assigned LVDT/R position feedback signals. If the regulator being defined for the output does not use position feedback signals, no entry is required is these fields. (If the LVDT/LVDR position feedback signal input is being used for some other purpose, then entries are required.) The fourth region includes any I/O Configuration Constants required by a type 2, 6, or 7 regulator function's integrator. If the regulator function being defined for the servo-valve output is one of the type(s) enclosed in the angle brackets, then entries are required in the fields. Each of the various fields will be described below.

```
                 TCQA Card Definition - Socket 1 - Screen 9/21

                    Regulator 6 (Q_Q_SVO6) Definition

    Function type & sub-type:           41
      Valid types <00, 40, 41, 43, 44, 51, 52, 53, 64, 65, 66>

    Suicide enable :-   Current Fault:    YES      LVDT/R fault:   YES

    Current Bias: (0 to 100% rated [10,20,40])        3.0
    Current Gain: (0 to 200% rated_cur./%pos.)        7.72

    Zero Stroke (Q_R_POS11) (0 to 6.667 Vrms) :   LVDT 1: 0.6999 LVDT 2: 0.6999
    100% Stroke (Q_R_POS12) (0 to 6.667 Vrms) :   LVDT 1: 3.478  LVDT 2: 3.478

    <6>   Pos limits          (-128% to 128%) :   Low: -100.0   High:   100.0
    <6>   Integrator convergence gain  (0 to  1 %/%): 0.3
    <6>   Position reference Gain      (0 to 32 %/%): 0.309
    <6>   Position ref time constant   (0 to 8 Sec) : 1.0
          Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

Figure 7-25. Regulator Definition Screen for Servo-Valve Output #6

7-7.1. Function Type & Sub-type

The two-digit hexadecimal number of the desired regulator function type and sub-type for the servo-valve output are entered here. If a servo-valve output is not used, it must have a regulator function type and sub-type 00 defined for it (to prevent nuisance diagnostic alarms). Valid types/sub-types for each servo-valve output are listed on each output's regulator definition screen. Appropriate cautions must be used when defining type 49 or type 2E regulators (see above).

7-7.2. Suicide Enable

There are two types of regulator suicides: a hardware suicide initiated when a current fault is detected and a software suicide initiated when an LVDT/R feedback fault is detected. A servo-valve output current fault is defined as the condition when either: 1) the called-for current output differs from the actual current output by more than 25% when the output is less than 80% of rated current, or 2) the called-for current output differs from the actual by more than 55% when the output is above 80% of rated current. (Whenever a servo-valve output current fault is detected, a Diagnostic Alarm to warn of the condition is annunciated, even if the suicide is not enabled.) If the current fault suicide is enabled by entering YES in the appropriate I/O Configurator field, a relay on the TCQA card will be de-energized and a pair of normally-closed contacts will short the output's terminals together. It is important to note that there are only four servo-valve output shorting relays per TCQA card for eight servo-valve outputs. Therefore, each shorting relay is used for a pair of servo-valve outputs—the pair associated with each shorting relay depends on the revision of the TCQA card (refer to Tables 10 & 11, below). A current fault detected on either output of a shorting relay will cause both outputs associated with the relay to be shorted.

Current Fault Suicide Shorting Relay/Servo-valve Output Assignments for Rev. G1A TCQA I/O Cards	
Shorting Relay	Assigned Servo-valve Outputs
1	SVO1 & SVO2
2	SVO3 & SVO4
3	SVO5 & SVO6
4	SVO7 & SVO8

Table 7-10. TCQA G1A Current Suicide Shorting Relay

Current Fault Suicide Shorting Relay/Servo-valve Output Assignment for Rev. G1B TCQA Cards	
Shorting Relay	Assigned Servo-valve Outputs
1	SVO1 & SVO5
2	SVO2 & SVO6
3	SVO3 & SVO7
4	SVO4 & SVO8

Table 7-11. TCQA G1B Current Suicide Shorting Relay

An LVDT/LVDR fault is defined as the condition when the position feedback signal from one or more of the assigned LVDT/LVDRs for the servo-valve output either: 1) drops below 0.38 Vac RMS, or 2) exceeds 6.44 Vac RMS. The detection limits are not field-adjustable. (Whenever a servo-valve output LVDT/LVDR is detected, a Diagnostic Alarm to warn of the condition is annunciated—even if the suicide is not enabled.) If the LVDT/LVDR current fault is enabled by entering YES in the appropriate I/O Configurator field, the output will be driven to 0.0 milliamps by software on the TCQA.

7-7.3. Current Bias and Current Gain

The current bias value of the servo-output's regulator is used to overcome the mechanical bias of the servo-valve's fail-safe spring. (If all current to the servo-valve were suddenly shorted or removed, it is the function of the servo's fail-safe spring to drive the device to the zero-flow or closed position.) In a TMR application, the value of the regulator's current bias should be one-third that required to overcome the fail-safe spring's mechanical bias. The units for the current bias value are percent of rated current.

The current gain of the servo-valve output's regulator is the gain of the regulator. The units for the current gain value are percent of rated current (milliamps) per percent output.

7-7.4. Zero Stroke and 100% Stroke

Required only if the regulator uses LVDT/LVDR position feedback, the values entered in these fields are used to scale the first two of the possible assigned LVDT/LVDR position feedback signals for the servo-valve output. In effect, these determine the offset and gain values for scaling the first two of the possible assigned position feedback signals. These values are site- and calibration-dependent and will change each time the LVDT/LVDR feedback signal(s) are re-calibrated. The Auto-Calibrate function of the Mark V will automatically change these values during the calibration sequence. When manually calibrating LVDT/LVDR position feedback signals, the measured values for the first two of the possible assigned position feedback signals must be entered.) Note that when a type 49 or type 2E regulator is defined for a particular servo-valve output, the other assigned LVDT/LVDR position feedback signal(s) zero and 100% stroke values will appear on a different Regulator Definition Screen. Units for the entries in these fields are V ac RMS.

If the regulator defined for a particular servo-valve output does not use LVDT/LVDR position feedback, it is not necessary to enter any values in these fields. If the regulator defined for a particular servo-valve output does not use LVDT/LVDR position feedback but the second of the two possible assigned LVDT/LVDR position feedback inputs is being used for another purpose (display/monitoring/logging), the measured voltages must be entered to scale the feedback signal for use.

7-7.5. Position Limits

Typically used to prevent regulator wind-up (sitting in a limit) and/or to limit the integrator function of type 2, 6, or 7 regulators, the values in these two fields (Low and High) are only required if the regulator being defined for the servo-valve output is one of the type(s) enclosed in the angle brackets at the left of the line. (The regulator signal flow diagram mnemonics for the low and high position limits are LLIM and HLIM, respectively.) The units for the position limits are percent (±128.0%), although servo-valve output currents are typically never expressed as being more than 100% or less than -100%.

7-7.6. Integrator Convergence Gain

Used by the integrator function to prevent drift of the three output signals on a TMR control panel, the integrator convergence gain is only required if the regulator being defined for the servo-valve output is one of the type(s) enclosed in the angle brackets at the left of the line. (The regulator signal flow diagram mnemonic for the integrator convergence gain is " K_Convergence ".)

7-7.7. Position Reference Gain

Used by the integrator to convert the input signal value's units to percent stroke and apply a gain to the function, the position reference gain is only required if the regulator being defined for the servo-valve output is one of the type(s) enclosed in the angle brackets at the left of the line. For a type 6 regulator function, the units of the position reference gain are percent stroke per percent of rated flow. For a type 7 regulator, the units of the position reference gain are percent stroke per psi (in English units; in metric units it would be percent stroke per bar). (The regulator signal flow diagram mnemonic for the position reference gain is " K ".)

7-7.8. Position Reference Time Constant

Used by the integrator to determine the response time of the transfer function, the position reference time constant is only required if the regulator being defined for the servo-valve output is one of the type(s) enclosed in the angle brackets at the left of the line. (The regulator signal flow diagram mnemonic for the position reference time constant is " T ".)

7-7.9. Flow Control Regulator Flow Feedback Pulse Rate Definitions

Flow control regulators (type 5 or type 6) must have their pulse rate definition values specified in the I/O Configurator but on a separate screen from the regulator definition screen. A typical Pulse Rate Definition screen is shown in Figure 7-26. MPU3 and MPU4 are the two flow control regulator pulse-rate feedback signal inputs. Each pulse rate input has three fields. Either the type of input must be specified in the Application Type field or five dashes must be entered to indicate the input is not being used (this is to prevent nuisance Diagnostic Alarms associated with the input). If the input is being used, then appropriate values must be entered into the Gain Scaling Base and Max Pulse Rate fields. Each of the three fields will be described below.

```
                TCQA Card Definition - Socket 1 - Screen 17/21

                           Pulse Rate Definition
                           Gain Scaling      Max         Application
                               Base       Pulse Rate        Type
                           (power of 2)  (100% rating)  (see notes)
                          -------------- -------------  -----------
  Pulse rate 1 (Q_Q_MPU1):     8192          5104        speed

  Pulse rate 2 (Q_Q_MPU2):     8192          5104        -----

  Pulse rate 3 (Q_Q_MPU3):     8192          3980         fuel

  Pulse rate 4 (Q_Q_MPU4):     8192          3980        -----


          Note: Valid Application Types (speed, fuel, pmg, -----)
```

Figure 7-26. Flow Control Regulator - I/O Configurator Screen

7-7.9.1. FLOW FEEDBACK PULSE RATE DEFINITION "APPLICATION TYPE" FIELD An entry in this field is required even if the input is not being used and indicates which type of input signal is being connected. The four valid entries for this field are `"speed", "fuel", "pmg",` and `"-----"` (five dashes). If the input is a magnetic speed pickup sensing turbine shaft speed (such as for HP or LP shaft speed pickups which would normally be connected to MPU1 or MPU2) then `"speed"` should be entered in the Application Type field. If the input is a magnetic speed pickup sensing liquid fuel flow divider speed (or possibly a conditioned TTL input from a fuel flow meter) then `"fuel"` should be entered in the Application Type field. If the input is from a permanent magnet generator being used to sense speed or flow-rate then `"pmg"` should be entered in the Application Type field. If the input is not being used for any purpose then `"-----"` (exactly five dashes) must be entered to prevent nuisance Diagnostic Alarms associated with the input from being annunciated.

7-7.9.2. FLOW FEEDBACK PULSE RATE DEFINITION "MAX PULSE RATE" FIELD. For each input being used (no entry in this field is required if the input is not being used) the pulse rate which will be produced by the device at the rated operating speed or flow-rate of the device must be entered into this field. This is not necessarily the pulse rate produced at the maximum expected speed or flow-rate. For example, the rated design speed of a single-shaft, heavy-duty Frame 6 gas turbine is 5100 rpm. Typically, a 60-toothed wheel is used along with a magnetic speed pickup to provide the shaft speed input signal to the Mark V, thus 1 Hz equals 1 RPM. However, when connected to a driven device through a reduction gear the normal operating speed of the turbine might be 5134 RPM per the gear ratio. For such a unit, the appropriate `Max Pulse Rate` value for the shaft speed pickup would be 5134. For the liquid fuel flow-divider pulse rate feedback input in Figure 7-26 (MPU3), the pulse rate produced by the flow-divider at the unit's rated isobaric power output would be 3980 Hz, even though on cold days the liquid fuel flow could produce a feedback signal in excess of 3980 Hz.

7-7.9.3. FLOW FEEDBACK PULSE RATE DEFINITION "GAIN SCALING BASE" FIELD. For each input being used (no entry is required if the input is not being used) the power of 2 closest to but greater than the maximum expected pulse rate which will be produced by the device must be entered into this field. For example, the liquid fuel flow-divider pulse rate input in Figure 7-26 (MPU3) could be greater than 3980 Hz on a cold day. The nearest power of 2 (not multiple of 2) which is closest to 3980 but greater than 3980 is 4096, but the maximum expected pulse rate from the flow divider could exceed 4096 Hz on a cold day. the next closest power of 2 greater than 4096 is 8192. Therefore, 8192 is the appropriate entry for the `Gain Scaling Base` field in this example.

7-7.10. Pressure Control Regulator Pressure Transducer Feedback Signal Scaling

Pressure control regulators (type 7) must have their pressure transducer feedback signal scaling values specified in the I/O Configurator, but on a separate screen from the regulator definition screen. A typical Voltage Input screen is shown in Figure 7-27. Values entered in the fields of the Voltage Input screen of the I/O Configurator are used to scale the pressure feedback signal appropriately for use by the regulator.

It should be noted that Signals 1 & 2 refer to the three possible pressure transducer feedbacks for the two pressure feedback inputs to a regulator. (Each control processor in a TMR control panel has its own individual pressure transducer inputs 1 & 2.) Signals 3 & 4 are voltage/current inputs which are not available for use as regulator feedback signal inputs. They are individual inputs which are "fanned out" to all three control processors in a TMR control panel (<R> only in a SIMPLEX panel).

7-7.10.1. VOLTAGE INPUT "SIGNAL IN USE" FIELD. For each regulator pressure feedback input (Signal 1 and Signal 2), there are four fields. If either input is being used, `YES` must be entered in the `Signal In Use` field. If either input is not being used, `NO` must be entered in order to prevent nuisance Diagnostic Alarms associated with the input from being annunciated. If a signal is not being used, values in the other field need not be specified.

```
                TCQA Card Definition - Socket 1 - Screen 14/21

                              Voltage Input

                      Signal      Full Scale    Min (0v)     Max (10v)
                      in use      CDB value     CDB value    CDB value
                      -------     ----------    ---------    ---------
Signal  1             YES         2048              0.0        600.0
Signal  2             NO          0                 0.0        0.0

Signal  3             NO          0                 0.0        0.0
Signal  4             YES         1                 0.0        0.26
```

Figure 7-27. Pressure Control Regualtor - I/O Configurator Screen

7-7.10.2. VOLTAGE INPUT. "FULL SCALE CDB VALUE" FIELD. The maximum Mark V control signal database (CDB) value for the scale type of each pressure control regulator pressure feedback signal must entered in this field. (The usual Mark V scale type for pressure control regulator pressure feedback signals is `PRESS.` The maximum English units CDB value for `PRESS` is 2048.0 psi, and the maximum Metric units CDB value for `PRESS` is 141.21 bar. The scale type for an input point can be determined from the I/O assignment file; the maximum can usually be determined from information in the beginning of the SCLEDATA.DAT file.) All values entered into any I/O Configurator field must be in English units. Therefore, the usual entry in the Full Scale CDB Value field for pressure control regulator pressure feedback signals is 2048.

7-7.10.3. VOLTAGE INPUT "MIN (0V) CDB VALUE" FIELD. The Min (0v) CDB Value field entry must be the minimum, or "zero", pressure transducer feedback signal value (in engineering units) which produces 0.0 Vdc at the I/O terminal board terminals. This is the feedback scaling "offset" value. To calculate the Min (0v) CDB Value, the "gain" (psi/Vdc) of the pressure transducer feedback signal at terminals on the TBQB I/O terminal board must be calculated using the following formula:

where: Calibrated Pressure Range=(Calibrated Full Scale Pressure)-(Calibrated Zero Scale Pressure)

$$Gain = \frac{Calibrated\ Pressure\ Range}{Calibrated\ Voltage\ Range}$$

Calibrated Voltage Range=(Calibrated Full Scale Voltage)-(Calibrated Zero Scale Voltage)

Substituting the calculated gain of the pressure transducer feedback in the following formula will produce the `Min (0v) CDB Value`:

Note the negative sign!

$$\text{Min } (0v)\ CDB\ Value = -\ [(I / O\ terminal\ board\ zero\ pressure\ voltage)\ x\ Gain]$$

For example, if a voltage pressure transducer calibrated for 0.0-5.0 Vdc @ 0.0-300.0 psi is being used for the pressure control regulator pressure feedback input #1, the calibrated pressure range would be 300.0 psi (300.0 psi - 0.0 psi) and the calibrated voltage range (at the TBQB I/O terminal board terminals 25 & 26) would be 5.0 Vdc (5.0 Vdc - 0.0 Vdc). The gain would be 60.0 psi/Vdc (300.0 psi / 5.0 Vdc). Knowing the calculated gain, the entry in the Min (0v) CDB Value field for Signal 1 would be 0.0 psi (**-**(0.0 Vdc * 60.0 psi/Vdc)).The units are implied and must not be entered in the field with the value.

4-to-20 mA pressure transducers can be used instead of voltage pressure transducers as pressure control regulator pressure feedback inputs. When such a transducer is used, a hardware jumper must be installed which will place a 250 ohm dropping resistor across the transducer feedback signal input terminals. A 4.0 mA current through a 250 ohm resistor will produce 1.0 Vdc at the TBQB I/O terminal board (the calibrated zero scale voltage), and a 20.0 mA current through a 250 ohm resistor will produce 5.0 Vdc at the TBQB I/O terminal board terminals. This results in a calibrated voltage range of 4.0 Vdc. If a 4-to-20 milliamp pressure transducer calibrated for 4-to-20 mA @ 0.0-300.0 psi were being used for the pressure control regulator pressure feedback input #1, the calibrated pressure range will be 300.0 psi. Using the formula above, the entry in the Min (0v) CDB Value field for Signal 1 would be -75.0 psi (-(1.0 Vdc * 75.0 Vdc)). Again, the units are implied and must not be entered in the field with the numeric value.

7-7.10.4. VOLTAGE INPUT. `"MAX (10V) CDB VALUE"` FIELD. The `Max (10v) CDB Value` field must be the maximum pressure transducer feedback signal value, in engineering units, which produces 10.0 Vdc at the I/O terminal board. The Mark V I/O Configuration software is expecting a 10.0 Vdc input at full-scale transducer output. To calculate the `Max (10v) CDB Value` for a pressure control regulator pressure transducer feedback use the gain calculated when determining the `Min (0v) CDB value` in the following formula:

Max (10v) CDB Value = (Gain x 10.0 Vdc) + (Min (0v) CDB Value)

Typically, the voltage transducers used on GEPG-provided heavy duty gas turbines have been 0.0-5.0 Vdc transducers. For example, if a voltage transducer calibrated for 0.0-5.0 Vdc output @ 0.0-300.0 psi is being used for the pressure control regulator pressure feedback input #1, the gain for Signal 1 would be 60.0 psi/Vdc and the `Min (0v) CDB Value` would be 0.0 psi. Substituting these values into the formula above, the `Max (10v) CDB Value` would be 600.0 psi ((60.0 psi/Vdc * 10 Vdc) + (-0.0 Vdc)). The units are implied and must not be entered in the field with the numeric value.

If a 4-to-20 mA pressure transducer calibrated for an output of 4-to-20 mA @ 0.0-300.0 psi were being used for the pressure control regulator pressure feedback input #1, the gain for Signal 1 would be 75 psi/Vdc and the `Min (0v) CDB Value` would be -75.0 psi. Substituting these values into the formula above, the `Max (10v) CDB Value` would be 675.0 psi ((75.0 psi/Vdc * 10.0 Vdc) + (-75.0 psi)). The units are implied and must not be entered in the field with the numeric value.

7-7.11. Pressure Control Regulator Time Constant Data

All type 7 pressure control regulators have a variable integrator time constant. Values must be supplied in order for the I/O Configuration software to calculate the regulator's variable integrator time constant (refer to Figure 7-28). Future Type 7 pressure control regulators may not have this variable integrator time constant feature and these I/O Configuration Constants will not have to be supplied. These I/O Configuration Constants are determined by GEPG and/or its Business Associates and should not be modified. If a type 7 pressure control regulator is required for an application and there is no information available about time constants or the feature is not necessary, it can effectively be disabled by entering a value of 0.0 in the SRV Warmup Time Constant field.

```
             TCQA Card Definition - Socket 1 - Screen 13/21

               Time Constant Data for Type 7 Regulators

(Q_Q_SVO1)
Type 77 or 7A:: SRV Warmup time constant (0 to 8 sec):          0.2119
Type 77 or 7A:: SRV Warmup time constant gain (0 to .625)       0.0
Type 77 or 7A:: System Warmup  FSR (0 to 128%):                 20.0

   Note: Calibrate pressure feedback by adjusting Voltage Input no. 1.

(Q_Q_SVO7)
Type 78 or 7B:: SRV Warmup time constant (0 to 8 sec):          0.2119
Type 78 or 7B:: SRV Warmup time constant gain (0 to .625)       0.0
Type 78 or 7B:: System Warmup  FSR (0 to 128%):                 20.0

   Note: Calibrate pressure feedback by adjusting Voltage Input no. 2.
```

Figure 7-28. Pressure Control Regulator - I/O Configurator Screen

7-7.12. Servo-Valve Diagnostics

Diagnostics alarms are provided throughout the servo drive system. These alarms will be annunciated on the operator control display and include:

1. A/D out of range
2. D/A out of range
3. MA input out of range
4. Position input out of range
5. LVDT/LVDR excitation out of range
6. Servo coil open or defective wiring

7-8. FLAME DETECTION (GAS TURBINES ONLY)

The flame detection system detects a flame in the gas turbine combustion cans and trips the turbine when improper combustion is detected in the turbine firing chambers. Two principle requirements of the flame detection system are:

1. To protect the turbine during startup.
   During startup, the turbine fuel is ignited by spark plugs. These plugs are usually fired for one minute. By the end of the firing time, the flame detector must sense the presence of flame in the combustion cans or the startup attempt will be aborted, the fuel stop valve will close, the ignition circuits will de-energize, and the turbine will require a shutdown before another start attempt.

2. To protect the turbine against a flameout while running.
   If flame is lost while the turbine is running, fuel will be shut off and the turbine will be tripped.

Ultraviolet flame detectors monitor flame in the combustion chamber by detecting the ultraviolet radiation emitted by the flame. Each detector consists of a window body filled with hydrogen gas and a cathode element made of pure copper. When flame is present in the combustion chamber, the energy present in the ultraviolet radiation of the flame ionizes the gas and releases electrons from the pure copper cathode, causing a "cascade effect," and a pulse of current flows.

When ultraviolet radiation is present in the tube, the resulting gas ionization causes current to pass through it, discharging (firing) the tube's capacitance. When the voltage across the tube is reduced sufficiently, the tube stops conducting. This process is repeated as the voltage across the tube again starts to rise toward its "firing voltage." The voltage level at which the tube fires is dependent upon the intensity of the ultraviolet light present. This process continues by giving out current pulses from the tube as long as the ultraviolet (UV) light is present.

The Mark V <P> core contains three independent, identical TCEA cards (<X>, <Y>, and <Z>) that have their own processors and power supplies. The power supplies provide the 335 V dc excitation current for the detectors as well as the control power for the card. Eight UV flame detectors can be connected to the <P> core terminal boards. Each detector is then internally parallel wired to each of the three cards. This redundancy exists in both the Mark V TMR and Simplex systems (Simplex applications also have TMR emergency tripping, therefore, all Mark V systems will have three cards). This design avoids the need for interposing transducers and separate 335 V excitation supplies.

The flame detection takes place in the <P> core and sends the information to the control processors (<R>, <S>, <T>). The CSP determines if and when to trip the PTRs. See the CSP and turbine vendor documentation for details.

Appendix D Figure D-63, shows a simplified diagram of the flame detector circuit and a table with terminal numbers, cable and pin numbers, and Mark V names for flame intensity frequency, threshold values, and logic names for the output of the detectors. The threshold is the number of pulses generated by the flame detectors in 1/16th of a second. The threshold values are stored in the I/O configuration file, and should not be modified without direction from the manufacturer. The flame intensity frequency (for example FD_INTENS_1) and logic names for the detector output can be displayed on the operator interface .

7-9. VIBRATION MEASUREMENT AND PROTECTION

Three different types of vibration sensors can be directly connected to the Mark V control panel. These sensors or transducers are mounted on several critical components of turbine and driven load to protect against long term excessive vibration damaging the turbine or load.

Each sensor terminates on one set of terminal board points, and then is parallel wired via ribbon cables to each of the three control processors. In TMR systems, a discrepancy between the three control processors inputs is alarmed as an internal diagnostic fault message while the vibration protection system continues to perform its normal calculations based on the median vibration signal.

7-9.1. Seismic (Velocity) Sensors

Normally, there are three to five transducers on the turbine which generate a small ac current by passing a magnet through a fixed coil. The Mark V control processors interface directly with up to twelve vibration sensors or "pickups".

These sensors output a millivolt signal which is converted to a digital value proportional to vibration on the TCQA card in each control processor. The control processor's CSP uses this information in controlling and protecting and the turbine. The seismic vibration pickups are terminated in the Mark V on the TBQB I/O Terminal board in location 7 on the <R> core. The signal from each pickup is paralleled ("fanned out") to all the control processors in the Mark V control panel via ribbon cables from the TBQB card on <R> to the TCQA card in each control processor.

The types of sensor diagnostic alarms provided include:

1. Open sensor detection and alarm. A small dc current is continuously passed through the vibration sensor at all times to detect a broken wire or other open input.
2. Disabled sensor alarm. Failed sensors can be manually disabled via the operator interface and will result in a permanent alarm condition.

The vibration protection system will either trip or initiate a controlled shutdown of the turbine or generator under any one of the following conditions:

1. A shutdown will occur if several sensors fail or are manually disabled. (The number of sensors varies with each application, thus the number of sensors out of service before a shutdown occurs will vary.)
2. A trip will occur if a high level vibration is detected on any one sensor, and;
   - 2a. an adjacent pair of sensors is disabled or alarmed,
   - 2b. **or** any other sensor is above the alarm limit.
   - 2c. **or** two (2) or more sensor inputs are disabled.

Vibration inputs are monitored by the TCQA card which has a range of 0 to 0.75 v-peak and an input impedance of 100 to 2,000 ohms. Typical sensitivity ranges are 0.1 to 0.4 V peak-to-peak per inch per second.

7-9.1.1. CONFIGURING SEISMIC VIBRATION PICKUPS IN THE MARK V. Figure 7-29 is the I/O Configurator screen which is used to enable the inputs disable diagnostic alarms associated with unused inputs, and to provide the I/O Configuration Constants for scaling. Additional I/O Configuration information for seismic vibration pickup Diagnostic Alarms must be entered also be entered on a separate screen (see Figure 7-30).

`Transducer Used` Field If a seismic vibration pickup is connected to an input point, the value `YES` must be entered into the `Transducer Used` field on the screen and the sensitivity of the pickup must be entered in the corresponding `Vibration Sensitivity` field. If no pickup is connected to an input point, the value `NO` must be entered in the `Transducer used` field to prevent diagnostic alarms associated with the unused input point from being annunciated. If `NO` is entered under `Transducer used`, the value in the corresponding `Vibration Sensitivity` field is of no consequence.

`Vibration Sensitivity` Field The value entered in the `Vibration Sensitivity` field is the sensitivity rating of the velocity-type seismic vibration pickup, and must be in English engineering units (V/inch-per-second).

`Seismic Vibr. Xdcr. Open Ckt.` Field The entry in the `Seismic Vibr. Xdcr. Open Ckt.` field is the voltage above which a Diagnostic Alarm indicating a seismic vibration pickup open circuit condition exists (for all seismic vibration pickups). In order to accurately calculate this value, the pickup's characteristics must be known, particularly the maximum expected resistance of the pickup circuit at maximum running ambient temperatures. The formula for calculating the value will be displayed on the appropriate TCQA I/O Configurator screen.

```
                    TCQA Card Definition - Socket 1 - Screen 18/21
                               Vibration Definitions
                            Transducer used       Vibration Sensitivity
                                              (0.05 to 0.40 V peak / ips peak)
                            ---------------   --------------------------------
Transducer  1 (Q_R_VIB01):        YES                   0.15
Transducer  2 (Q_R_VIB02):        YES                   0.15
Transducer  3 (Q_R_VIB03):         NO                   0.15
Transducer  4 (Q_R_VIB04):        YES                   0.15
Transducer  5 (Q_R_VIB05):        YES                   0.15
Transducer  6 (Q_R_VIB06):         NO                   0.15
Transducer  7 (Q_R_VIB07):        YES                   0.15
Transducer  8 (Q_R_VIB08):        YES                   0.15
Transducer  9 (Q_R_VIB09):        YES                   0.15
Transducer 10 (Q_R_VIB10):        YES                   0.15
Transducer 11 (Q_R_VIB11):        YES                   0.15
Transducer 12 (Q_R_VIB12):        YES                   0.15
```

Figure 7-29. Seismic Vibration I/O Configurator Screen

```
               TCQA Card Definition - Socket 1 - Screen 21/21
                       Miscellaneous Values

      Cold Junction Filtering  (power of 2):         16
      Seismic Vibr. Xdcr. Open  Ckt. (0 to 3Vdc):    1.000
          <       V = 30R / (40000+R)    {eg,  1V=1379Ω }     >
          < Xdcr Resistance = 40000*V/(30-V) {eg,  1379Ω=1V } >

      125 Vdc Under Voltage Limit (0 to 125 Vdc):    105.0
      125 Vdc Bus Ground Fault    (0 to  65 Vdc):    31.24
          {voltage at which ground fault is indicated}
```

Figure 7-30. Seismic Vibration I/O Configurator Screen

```
            TCQB Card Definition - Socket 2 - Screen 7/10
                  Configuration Information
  Prox Chan 1 to 3 Device type:     ACCEL (valid types are:
                                          PROXimitor and ACCELerometer.
                                          Verify configuration jumpers
                                          on TCQB board when change
                                          is made)
  HP rpm max scaling value  :     16384
  IP rpm max scaling value  :      8192
  LP rpm max scaling value  :      8192
```

Figure 7-31. Accelerometer Vibration I/O Configurator Screen

```
                 TCQB Card Definition - Socket 2 - Screen 10/10
                       LM Vibration Inputs

                                Normalized      Normalized     Scaling
                               Scaling Gain    Scaling Offset   Shift
                               ------------    --------------  -------
  Transducer 1 (LMHP_S_PRX01):    0.0              0.0           0
  Transducer 2 (LMIP_S_PRX01):    0.0              0.0           0
  Transducer 3 (LMLP_S_PRX01):    0.0              0.0           0

  Transducer 4 (LMHP_S_PRX02):    0.0              0.0           0
  Transducer 5 (LMIP_S_PRX02):    0.0              0.0           0
  Transducer 6 (LMLP_S_PRX02):    0.0              0.0           0

  Transducer 7 (LMHP_S_PRX03):    0.0              0.0           0
  Transducer 8 (LMIP_S_PRX03):    0.0              0.0           0
  Transducer 9 (LMLP_S_PRX03):    0.0              0.0           0
```

Figure 7-32. Aero-Derivitive Turbine - Vibration I/O Configurator Screen

7-9.2. Accelerometer Inputs

Accelerometers are used to monitor vibration on aircraft derivative gas turbines. The charge amplifiers, which are located on the turbine base, feed a velocity signal to the TBQD termination card. Three accelerometers can be monitored. Internal tracking filters are used to select the appropriate frequencies which result in alarm and trip protection of the turbine.

The Mark V is capable of both exciting accelerometer vibration sensors and scaling accelerometer feedback signals. The first three proximity transducer/accelerometer input points to the TCQB I/O card (PRX01, PRX02, and PRX03) can be used as accelerometer inputs (this involves setting I/O Configuration Constants and positioning hardware jumpers on the TCQB card to accept accelerometer inputs rather than proximity transducer inputs). Configuring the Mark V to scale accelerometer input signals involves entering information on two I/O Configurator screens; refer to figures 7-31 and 7-32.

`Prox Chan 1 to 3 Device type` Field The first three vibration input points to the TCQB card can be configured to accept/scale the input signals from either proximity transducers or accelerometers. This involves defining the appropriate I/O Configuration Constants and setting hardware jumpers on the card. The entry in the `Prox Chan 1 to 3 Device type` field defines the input point as either being from a proximity transducer or from an accelerometer. When defining one or more of the first three vibration inputs to the TCQB card as an accelerometer input, the value must be entered into the appropriate field(s) on the TCQB vibration proximity transducer shaft location assignment screen.

`HP, IP,` and `LP rpm max scaling value` Fields The entry in each of the three rpm maximum scaling value fields in Figure 7-31 must be a power of 2 greater than the highest expected shaft speed for the high (`HP`), intermediate(`IP`) and low (`LP`) pressure shafts of the turbine when accelerometers are being connected to the Mark V. These entries are not related to the highest expected frequency from the shaft speed pickups, but the highest expected scaled shaft speed feedback signals. Entries in these fields are of no consequence when proximity transducers are defined for the inputs.

7-9.3. Proximity Transducer Inputs

Proximity transducers use radio frequency waves to measure distance between an object and the probe. The transducer's output is a voltage signal inversely proportional to this distance. The ac component of the transducer's output is interpreted as vibration, while the dc component is interpreted as a change in position.

The optional Mark V proximity transducer inputs provide a direct three wire interface to the proximity transducer for monitoring, alarm and trip. The number of probes each Mark V has the capability to interface with changes with the application, but typically values are:

- 18 vibration inputs
- 10 position inputs
- 2 key phasor inputs

Each proximity transducer receives -24 V dc power from the Mark V. The vibration inputs produce a dc voltage with an ac component which must be within 3 ac V peak-to-peak and -2 to -18 V dc. Each input terminates on a single termination point and is then internally parallel wired with ribbon cables to the TCQB cards in control processors <R>, <S>, and <T> for TMR or <R> for Simplex. A discrepancy between the three control processor inputs is alarmed as an internal diagnostic fault message while the protection system continues to perform its normal calculations based on the correct median signal. The Mark V gas turbine Simplex is the only application where additional panel size may be required for additional terminal boards for these inputs.

Typical applications for proximity transducers are:

- Vibration - For steam turbine generators rotating at speeds above 500 rpm, these proximity inputs measure the peak-to-peak radial displacement of the journal (that is. the shaft motion in the bearing) in one or two radial direction(s). This system uses a non-contacting probe(s) and proximity transducers(s) and results in alarm, trip, and fault detection.

- Axial Position - A probe is mounted in a bracket assembly off the thrust bearing casing to observe the motion of a collar on the turbine rotor. This system uses non-contacting probes and proximity transducers and results in thrust bearing wear alarm, trip, and fault detection. Two and three channel systems are available.

- Differential Expansion - The rotor temperature changes faster than the turbine shell due to smaller mass, higher heat transfer coefficients and almost total immersion in the steam path. This results in expansion of the rotor and shell at different rates. The turbine shell is free to expand axially away from where it is keyed to the foundation, and the turbine rotor is free to expand axially away from its thrust bearing. The detection of this differential expansion utilizes non-contacting probes and proximity transducers and results in alarm, trip, and fault detection for excessive expansion in either direction.

- Eccentricity - A probe is mounted at least 18 inches away from the bearing centerline and continuously up-dates the steam turbine control via a proximity transducer while the turbine is on the turning gear. Alarm and fault indications are provided.

The input signals from proximity transducer (and associated key phasor) inputs will be wired to the TBQD I/O terminal board on location 7 on the <S> processor and will be paralleled (fanned out) to all the control processors via ribbon cables to the TCQB card in each control processor. The first eighteen proximity transducer input points are designated for vibration sensor inputs (PRX01 - PRX18); the next ten proximity transducer input points are designated for axial position/differential expansion sensor inputs; and the last two proximity transducer input points are designated for key phasor inputs. Several TCQB I/O Configurator screens are used to configure proximity transducer inputs, depending on the type of input signal (that is, vibration, axial position/differential expansion, or key phasor).

`Proximitor Shaft Location Assignments` Fields Figure 7-33 is the TCQB I/O Configurator screen used to set the shaft assignments for the proximity transducers. Valid entries for the field for each input are `HP, IP, LP,` and `--` (`HP` stands for high-pressure shaft, `IP` for intermediate-pressure shaft, `LP` for low-pressure shaft, and `--` for not used). For example, in Figure 7-33 vibration proximity transducer input #1 is connected to a sensor mounted on a high-pressure turbine shaft bearing, therefore HP is entered in the field for the #1 input, meaning that the input signal will be filtered using the HP shaft speed.

Note that proximity transducer inputs 1, 2, and 3 may be used for accelerometer-type vibration inputs from an aero-derivative gas turbine. The HP shaft input will usually be connected to input #1, the IP shaft (if present) to input #2, and the LP shaft to input #3. If accelerometer inputs are used, then -- (exactly two dashes) must be entered in the shaft location assignment fields. Also note there are hardware jumpers on the TCQB card which must be put in the proper position for inputs 1, 2, & 3, depending on whether proximity transducers or accelerometers are connected to the inputs. When defining one or more of the first three vibration inputs to the TCQB card as a proximity transducer input, the value PROX must be entered into the Prox Chan 1 to 3 Device type field on the TCQB configuration screen.

The process of configuring vibration proximity transducer inputs must be completed by filling in the appropriate information in the screens shown in Figure 7-34.

Proximitors :- Peak to Peak Vibration Transducer Sensitivity Fields The vibration proximity transducers' nameplate sensitivity must be entered in the transducer sensitivity fields shown in Figure 7-34 in order for the I/O Configuration software to scale the input voltage appropriately. The value must be entered using English engineering units, V/0.001 inch, or V/mil. Enter only the numeric value, units are not necessary.

Proximitor low (or high) limit diag: Fields Two fields are shown in Figure7-34 in which voltage values corresponding to low and high limits for Diagnostic Alarms (indicating out-of-rage conditions) are entered. Common proximity transducers normally have an output range of -20.0 Vdc to -2.0 Vdc, with a typical voltage output of -10.0 Vdc when properly installed and the shaft is at rest. Values in these fields will determine when the out of limits Diagnostic Alarm for the first eighteen proximity transducer inputs will be annunciated (these two values are for all of the first eighteen proximity transducer inputs).

Proximitors :- Position Inputs Transducer Sensitivity Fields Axial Position/Differential Expansion proximity transducer input points 19 through 28 are designated for use as either axial position or differential expansion inputs. The transducer's sensitivity, from the manufacturer's nameplate, must be entered in this field as shown in Figure 7-34 for each input point to which a transducer is connected in order for I/O Configuration software to scale the input voltage appropriately. The value must be entered using English engineering units, V/0.001 inch, or V/mil.

Proximitors :- Position Inputs Position Offset Fields The entries in the position offset fields of Figure 7-34 will be dictated by the application. The offset setting is the zero position, at rest position, or starting position. The value must be entered using English engineering units, 0.001 inch, or mils.

Proximitor low (or high) limit diag: Fields Two fields are shown at the bottom of Figure 7-34 in which voltage values corresponding to low and high limits for Diagnostic Alarms (indicating out-of-rage conditions) are entered. Common proximity transducers normally have an output range of -20.0 Vdc to -2.0 Vdc, with a typical voltage output of -10.0 Vdc when properly installed and the shaft is at rest. Values in these fields will determine when the "out of limits" Diagnostic Alarm for proximity transducer input points 19 through 28 will be annunciated (these two values are for all proximity transducer input points 19 through 28).

Proximitor Keyphasor Inputs Transducer Sensitivity Field Proximity transducer input points 29 and 30 are designated for use as keyphasor (shaft speed sensor) inputs. The transducer's sensitivity, from the manufacturer's nameplate, must be entered in this field for each input point to which a transducer is connected in order for I/O Configuration software to scale the input voltage appropriately and recognize the shaft's rotation indicator. The value must be entered using English engineering units, V/0.001 inch, or V/mil.

Proximitor Keyphasor Inputs Position Offset Field The entry in the position offset field of Fig. 7-34 for keyphasor inputs will be determined by field/Customer personnel as it must be the unit's desired offset in its zero speed condition. As of this writing, the value must be entered using English engineering units, 0.001 inch, or mils.

Proximitor Keyphasor Inputs Low and High Diag: Field Two fields are shown at the bottom of Figure 7-34 in which voltage values corresponding to low and high limits for Diagnostic Alarms (indicating out-of-range conditions) are entered. Common proximity transducers normally have an output range of -20.0 Vdc to -2.0 Vdc, with a typical voltage output of -10.0 Vdc when properly installed. When entering a value in these fields, the negative voltage sign is implied and must not be entered with the value. Values in these fields will determine when the out of limits Diagnostic Alarm for proximity transducer input points 29 and 30 will be annunciated (these two values are for both proximity transducer input points 29 and 30). Note that the value in the high limit field is in excess of the usual -20.0 Vdc; this is to disable the Diagnostic Alarm when key phasors are connected because of their tendency to saturate when the shaft rotation indicator passes.

```
                  TCQB Card Definition - Socket 2 - Screen 2/10
                    Proximitor shaft location assignments

Q_S_PRX01: HP      Q_S_PRX06: HP       Q_S_PRX11:  --      Q_S_PRX15:  --
Q_S_PRX02: HP      Q_S_PRX07: HP       Q_S_PRX12:  --      Q_S_PRX16:  --
Q_S_PRX03: HP      Q_S_PRX08: HP       Q_S_PRX13:  --      Q_S_PRX17:  --
Q_S_PRX04: HP      Q_S_PRX09: --       Q_S_PRX14:  --      Q_S_PRX18:  --
Q_S_PRX05: HP      Q_S_PRX10: --

     Notes::    Valid locations:    IP,  HP,  LP
                      not used :    --
                Assign channels 1 thru 3 as 'not used' when connecting
                  accelerometers as inputs.
```

Figure 7-33. Proximity Transducer - I/O Configurator Screen

```
       TCQB Card Definition - Socket 2 - Screen 3/10
       Proximitors  (1 thru 9)   :-  Peak to Peak Vibration
                        Transducer Sensitivity
                          (0 to 20.53 V/mil)
                         --------------------
 Proximitor 1 (Q_S_PRX01):      0.2
 Proximitor 2 (Q_S_PRX02):      0.2
 Proximitor 3 (Q_S_PRX03):      0.2
 Proximitor 4 (Q_S_PRX04):      0.2
 Proximitor 5 (Q_S_PRX05):      0.2
 Proximitor 6 (Q_S_PRX06):      0.2
 Proximitor 7 (Q_S_PRX07):      0.2
 Proximitor 8 (Q_S_PRX08):      0.2
 Proximitor 9 (Q_S_PRX09):      0.2
      Proximitor (1 thru 18) low  limit diag: (0 to 18v)     4.0
      Proximitor (1 thru 18) high limit diag: (0 to 18v)     17.0

       TCQB Card Definition - Socket 2 - Screen 4/10
            Proximitors  (10 thru 18)   :-  Peak to Peak Vibration
                              Transducer Sensitivity
                          (0 to 20.53 V/mil)
                         --------------------
 Proximitor 10 (Q_S_PRX01):      0.2
 Proximitor 11 (Q_S_PRX02):      0.2
 Proximitor 12 (Q_S_PRX03):      0.2
 Proximitor 13 (Q_S_PRX04):      0.2
 Proximitor 14 (Q_S_PRX05):      0.2
 Proximitor 15 (Q_S_PRX06):      0.2
 Proximitor 16 (Q_S_PRX07):      0.2
 Proximitor 17 (Q_S_PRX08):      0.2
 Proximitor 18 (Q_S_PRX09):      0.2

       TCQB Card Definition - Socket 2 - Screen 5/10
       Proximitors  (19 thru 28)   :-  Position Inputs
                          Transducer Sensitivity       Position Offset
                           (0 to 19.1345 V/mil)       (0 to 2048 mils)
                         --------------------        ----------------
 Proximitor  19 (Q_S_PRX19):     0.2                         45.0
 Proximitor  20 (Q_S_PRX20):     0.2                         45.0
 Proximitor  21 (Q_S_PRX21):     0.2                         45.0
 Proximitor  22 (Q_S_PRX22):     0.2                         45.0
 Proximitor  23 (Q_S_PRX23):     0.2                         45.0
 Proximitor  24 (Q_S_PRX24):     0.2                         45.0
 Proximitor  25 (Q_S_PRX25):     0.                          0.0
 Proximitor  26 (Q_S_PRX26):     0.                          0.0
 Proximitor  27 (Q_S_PRX27):     0.                          0.0
 Proximitor  28 (Q_S_PRX28):     0.                          0.0
     Proximitor (19 thru 28) low  limit diag: (0 to 18v)     4.0
     Proximitor (19 thru 28) high limit diag: (0 to 18v)     17.0
```

Figure 7-34. Proximity Transducer - I/O Configurator Screen

7-10. SYNCHRONIZING (Generator Drive Turbines Only)

The Mark V synchronizing system consists of synchronizing (sync) check protection and automatic sync. Two single phase potential transformer outputs (115 V ac nominal) are connected to the PTBA termination board on the <P> core, then go to the TCEA card in <P> where isolation transformers reduce this nominal 115 V ac input to 5 V ac. The TCEA cards in <P> digitize these signals and perform the automatic sync function, while the IOMA card(s) in the <R>, <S>, and <T> Controllers (or <R> in Simplex) perform the sync check function. Manual sync is performed remotely and can include the sync check feature as described herein.

During automatic synchronization the control sequence program performs speed matching and voltage matching at either a fast or slow rate depending on the difference between line frequency and generated frequency. The slow rate is used when the difference between frequencies is small to help eliminate overshoot. Voltages and frequencies are measured by zero voltage crossing techniques and the control sequence program confirms the following permissive are met: under and over voltage and frequency of the generator, under and over voltage and frequency of the line, and differential between the line and generated voltages.

A normally open contact on the power circuit breaker (ANSI designation 52G) is monitored to measure the time that it actually takes to close the breaker. This time is used for self adaptive adjustment of the breaker closure time anticipation constant and diagnostics. Two sets of breaker data can be maintained for power systems that require two synchronization points. Limits are set for the circuit breaker close time. If these limits are exceeded, a diagnostic alarm will be annunciated on the <I>.

Three Mark V internal functions must be satisfied before a breaker close command is issued:

1. Synchronize permissive or complete sequence from <R>, <S>, and <T>. (25P)
2. Sync check from <R>, <S>, and <T>. (25X)
3. Automatic synchronization signal from <X>, <Y>, <Z> (TCEA cards) in <P>. (25)

Synchronize permissive items may change with the application, but typically include such things as:

1. automatic synchronization selected,
2. dead bus closure selected,
3. field breaker closed,
4. unit at approximately synchronous speed,
5. turbine startup sequence is complete,
6. no remote sync inhibits are present.

7-10.1. Sync Check (25X)

An output from <R>, <S>, and <T> that is voted on at the output card (TCTx), sync check has an adjustable window inside which a breaker close is permitted. The following parameters must be within the limits programmed for a particular application to get a sync check output:

1. generator and line voltages above a minimum limit

and 2. difference between generator and line voltage less than a limit

and 3. difference in frequency between generator and line (also called slip), typically set to 0.27 Hz (max value of 0.33 Hz)

and 4. difference in phase between generator and line, expressed in degrees, typically set to 10 degrees (max value of 30 degrees)

A graphical representation of numbers 3 and 4 above is shown below.

```
                                   Slip
                                     |+0.27 Hz
The breaker will close       ┌───────┼───────┐
if the phase and slip        |       |       |
are "inside the              |       |       |
window".                 ────┼───────┼───────┼── Phase
                         -10 |       |       |+10 deg
                             |       |       |
                             └───────┼───────┘
                                     |-0.27 Hz
```

Figure 7-35 is a functional diagram of the synch check circuit. The diagram helps describe what hardware and software signals are invloved in determining what picks up the 25X relay.

7-10.2. Automatic Sync

The automatic sync signal is calculated in each of the three TCEA cards from independently derived data similar to the sync check. In addition the four defined above, the TCEA cards also take into account the following when calculating the breaker close signal:

1. automatic sync enabled and armed
2. bypass logic to provide for dead bus closure
3. the generator speed is accelerating
4. the anticipated circuit breaker closing delay (self measured, as described above)
5. generator speed is greater than sync speed, and has been for 12 consecutive power system cycles (0.2 seconds at 60 Hz, 0.24 seconds at 50 Hz)
6. closure will take place at a positive slip (that is, before 12 o'clock on the sync scope)

Remember each TCEA card calculates its own breaker close signal based on data it derives separately. The result of these calculations is voted on the output card TCTx. Two out of three TCEA cards must agree for the 25 relay (ANSI designation) to energize and close a contact to close the power circuit breaker.

Figure 7-36 is a functional diagram of the auto synch circuit. The diagram helps describe what hardware and software signals are invloved in determining what picks up the 25 relay.

CAUTION

The contact output to the generator breaker is rated to make and carry 10 amperes at 125 V dc; however, it is only rated to interrupt 0.5 amperes at 125 V dc. Consequently, it is imperative to interrupt the breaker close coil with an external device, normally a normally closed (NC) auxiliary contact of the breaker itself.

The automatic sync system has a Monitor Mode of operation. In this mode (selected with the operator interface) the automatic sync system functions normally except final output is disabled. This causes the system to automatically speed match and voltage match but inhibit actual synchronization until the operator observes all of the key sync parameters on the <I>.

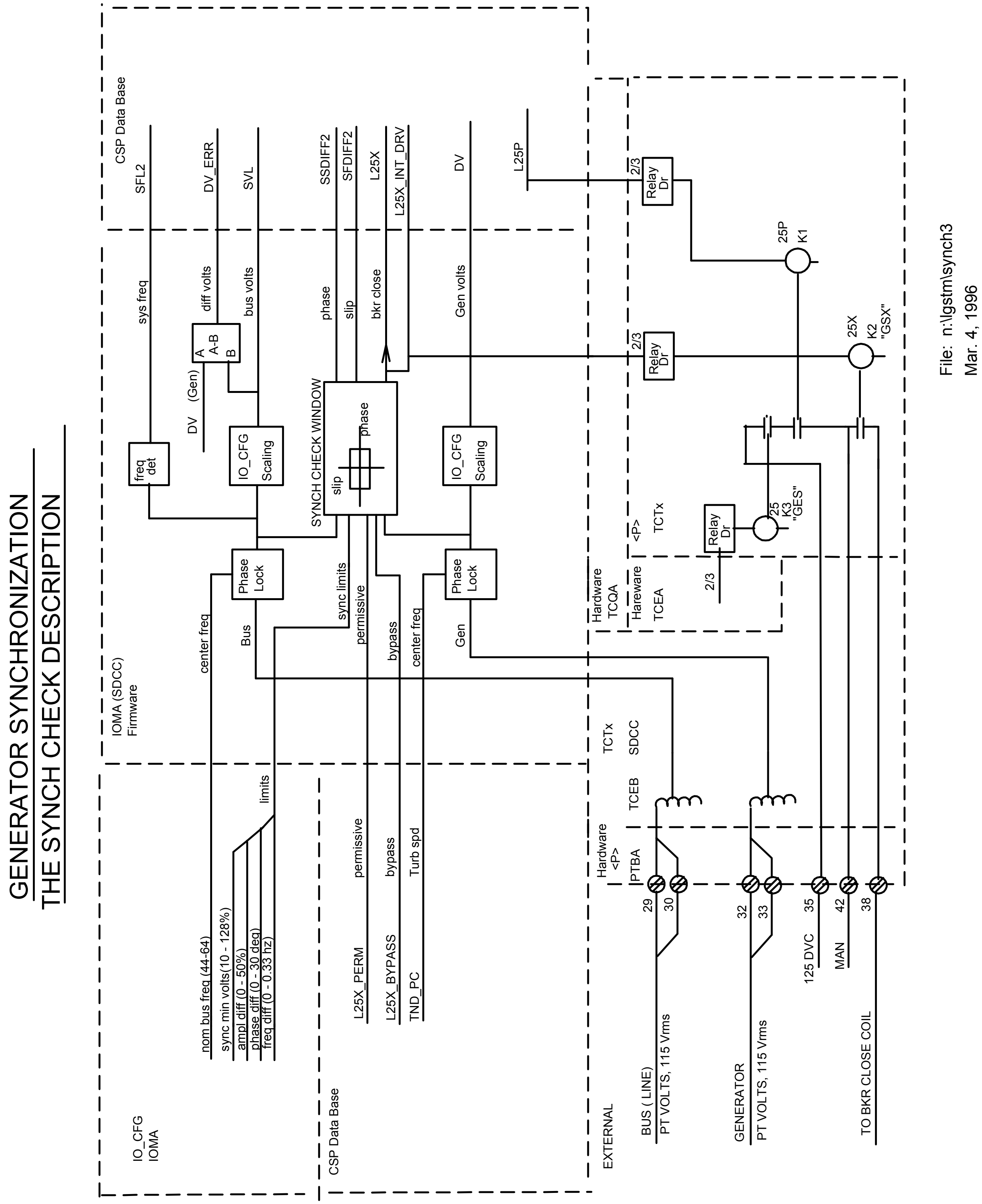


Figure 7-35. Generator Synch Check Description

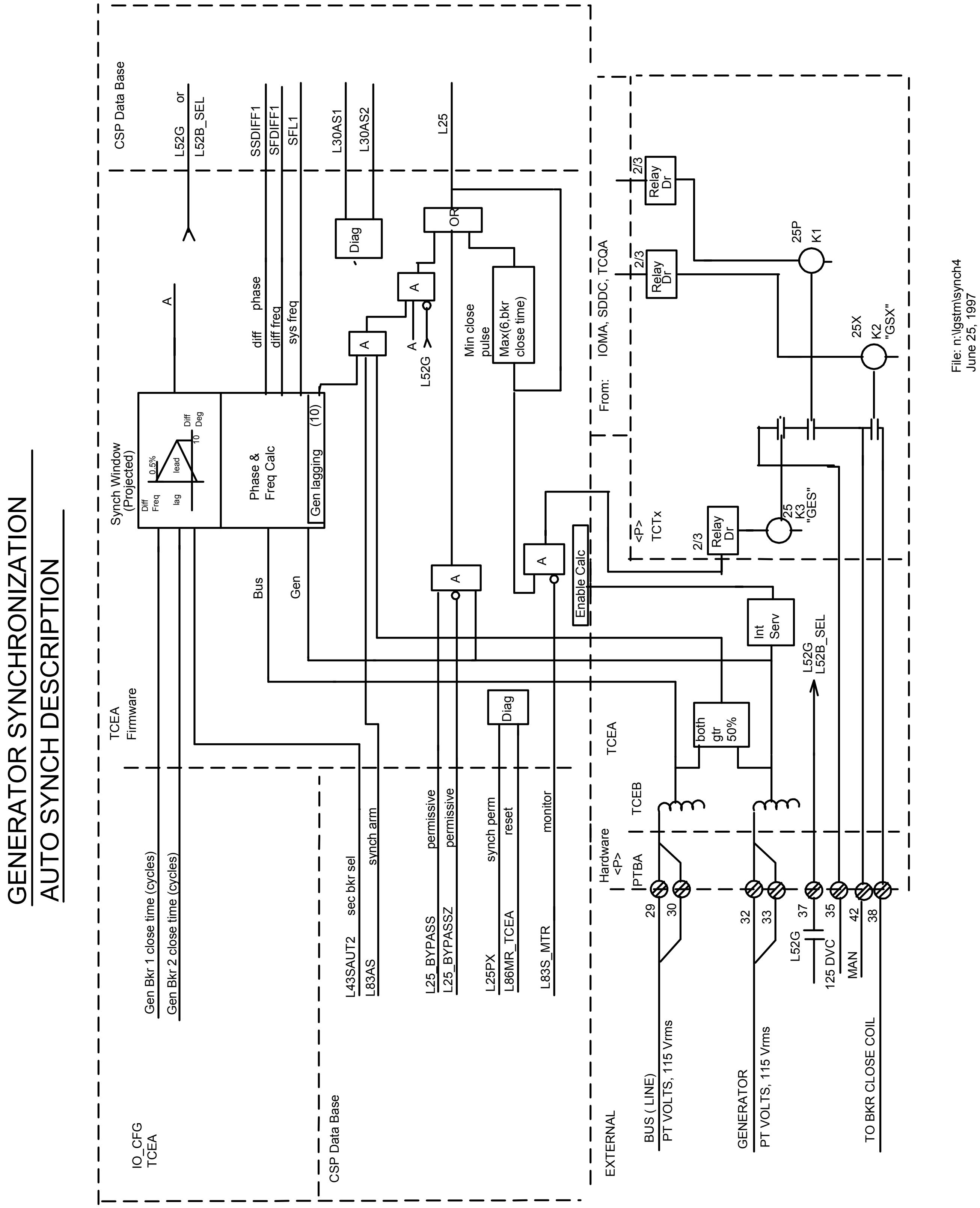


Figure 7-36. Generator Auto Synch Description

7-11. SHAFT VOLTAGE AND CURRENT MONITOR INPUTS

The Mark V is capable of accepting voltage inputs from shaft voltage and current monitors that monitor ac and dc voltages and ac and dc currents which might be present or develop on the turbine and/or generator shaft(s). Appendix B, section B-8 is the I/O Configurator screen used to define the I/O Configuration Constants which are used by the I/O Configuration software to scale the voltage input signal(s) proportional to shaft voltage and/or current. The values shown in B-8 are provided for units having the optional shaft voltage and current monitor functions. The following values scale the input signals (typically they are voltages) prior to being written into the control signal database where they are used by the shaft voltage/current monitor software. Consult GEDS or the turbine vendor before changing these values.

`Full Scale CDB Value` Field The value in this field is the maximum CDB value for the scale type chosen for the input (the scale type assigned in the scale column of the unit-specific IO.ASG file). Since shaft voltage and current monitor inputs are usually between 0.0 and 16.0 V dc, the usual CDB scale type selected for these inputs is V32 which has a maximum CDB value of 32.00 V dc.

`Min (0v) CDB Value` Field This is the minimum, or zero voltage, value for the input in V dc (typically 0.0 V dc) used to scale the input signal prior its value being written into the control signal database.

`Max (16v) CDB Value` Field This is the maximum, or full-range, value for the input in V dc (typically 16.0 V dc) used to scale the input signal prior its value being written into the control signal database.

7-12. POWER LOAD UNBALANCE

The Power Load Unbalance (PLU) option is used on Large Steam Turbines to protect the machine from overspeed under load rejection. It looks for an unbalance between mechanical and electrical power. This function initiates Control Valve (CV) and Intercept Valve (IV) fast closing actions under load rejection conditions where rapid acceleration could lead to an overspeed event. Valve actuation does not occur under stable fault conditions that are self-clearing (such as grid faults).

The PLU function is enabled through the I/O Configurator Editor as follows:

1. Exit to the Main Menu and select the I/O Configurator Editor option.
2. Select the <Q> Configuration Menu (from the bottom of the display).
3. Click on the "`TCPA`" I/O card target (the card's revision information is shown).
4. Select the "`Next Screen`" (from the bottom of the display). The screen in Figure 7-38 is displayed.
5. Select the "`Enable PLU Function`" option by clicking on the `YES/NO` command target. Selection is indicated by a change in the target's color: light blue = deselected, white = selected). Click on the target until it reads "YES."

Valve action occurs when the difference between turbine power and generator load is approximately 40% of rated load or greater, and the load is lost at a rate equivalent to going from rated to zero load in approximately 35 milliseconds (or less).

The 40% PLU level setting is standard. If it becomes necessary to deviate from this setting for a specific unit, the fact will be noted by the unit-specific documentation. This constant may be adjusted from the same I/O Configurator screen that is used to enable the PLU function. To change the PLU level setting, follow steps 1-4 and then select "`PLU unbalance threshold`." Enter the desired value using the keyboard.

Turbine mechanical power is derived from a milliamp reheat steam pressure signal. The generator load is assumed to be proportional to the sum of the three phase currents, thereby discriminating between load rejection and power line faults. This discrimination would not be possible if a true Megawatt signal was used.

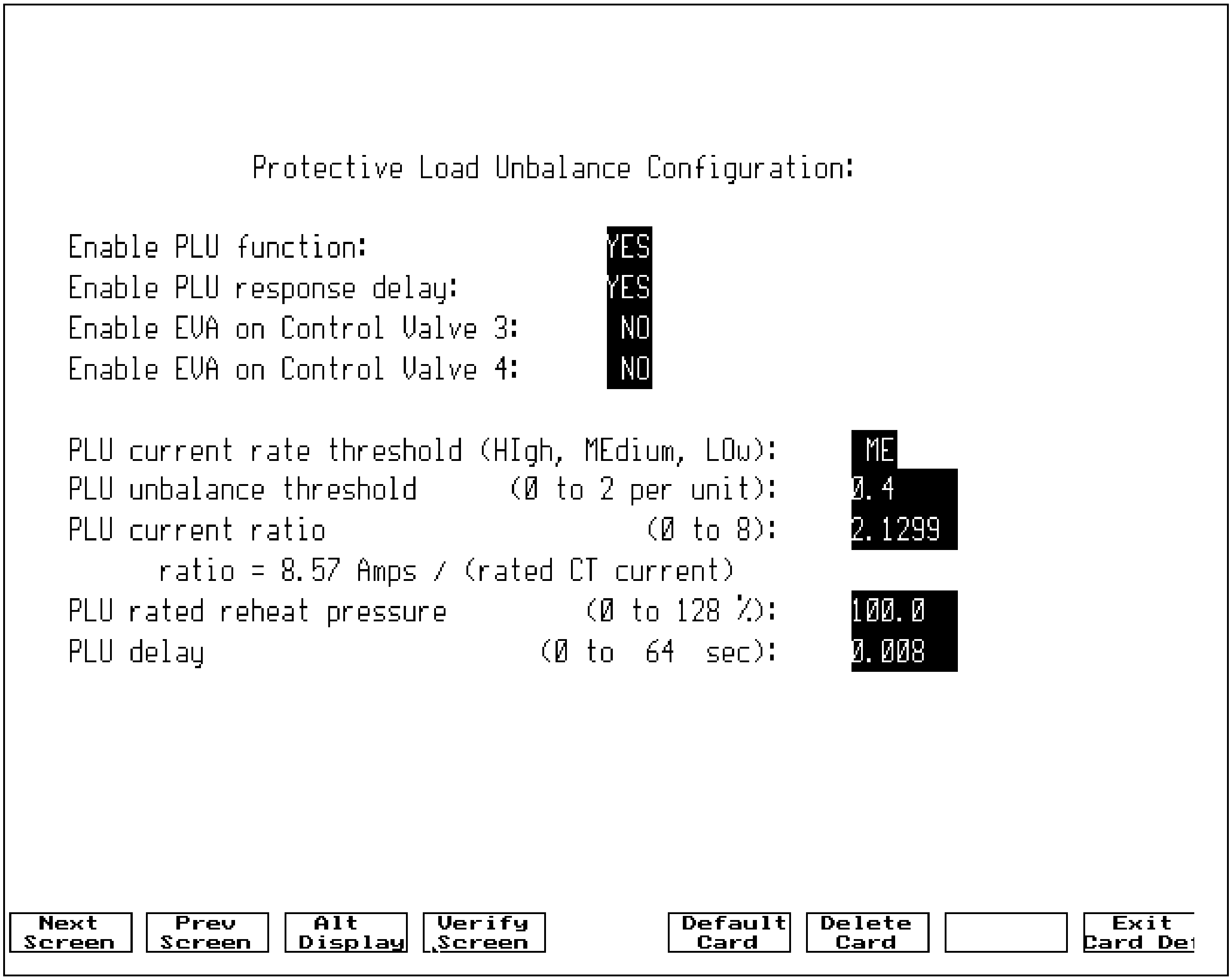


Figure 7-38. I/O Configurator Screen Showing PLU Configuration

The PLU signal actuates the CV and IV fast closing solenoids and resets the Load Reference signal to the no-load value (and performs some auxiliary functions).

CAUTION

The PLU function is an important part of the overspeed protective system. Do not disable during turbine operation.

The three current signals from the station current transformers are reduced by three auxiliary transformers, conditioned and digitized by the PTBA, TCEB, and TCPA printed circuit cards. These signals are summed and compared to the power pressure signal from the reheat pressure sensor. The signals are qualified according to the "`PLU current ratio`" and "`PLU rated reheat pressure`" inputs respectively (see Figure 7-38). This comparison yields a qualified unbalance measure of the PLU (see Figure 7-39, signal B). The output of the total generator current is also fed into the current rate amplifier. This comparison provides a measure of the rate of change of the generator current (see Figure 7-39, signal A).

The current rate level may be adjusted through the "`PLU rate threshold`" function of the I/O Configurator screen shown in Figure 7-38. Selections for this function are High, Medium, and Low. These settings correspond to 50, 35, and 20 millisecond rates respectively.

If these comparators operate simultaneously, PLU action is initiated and latched, making continuation of the PLU action dependent only on the unbalance for all functions except IV fast closing. The IVs do not lock in, but remain closed for approximately one second and then begin to re-open regardless of PLU duration.

A time-delay may be implemented for the PLU function. To initiate the delay, select `Enable PLU response delay` option (see Figure 7-38) and change the command target to read `YES`. The duration of the time-delay can be adjusted by altering the value supplied to the `PLU delay` option.

The IVs and CVs may be operated through test signals from the Control Sequence Program (CSP). These signals are executed individually and are logic ORed with the above signals. The IVs may also be driven by the Early Valve Actuation (EVA) and IV Trigger (IVT) functions. Each solonoid has a unique drop-out point delay. These values are tabulated as follows:

IV1	0.35	sec.
IV2	0.50	sec.
IV3	0.75	sec.
IV4	0.35	sec.
IV5	0.75	sec.
IV6	0.50	sec.
CV1	1.10	sec.
CV2	2.00	sec.
CV3	3.00	sec.
CV4	4.00	sec.

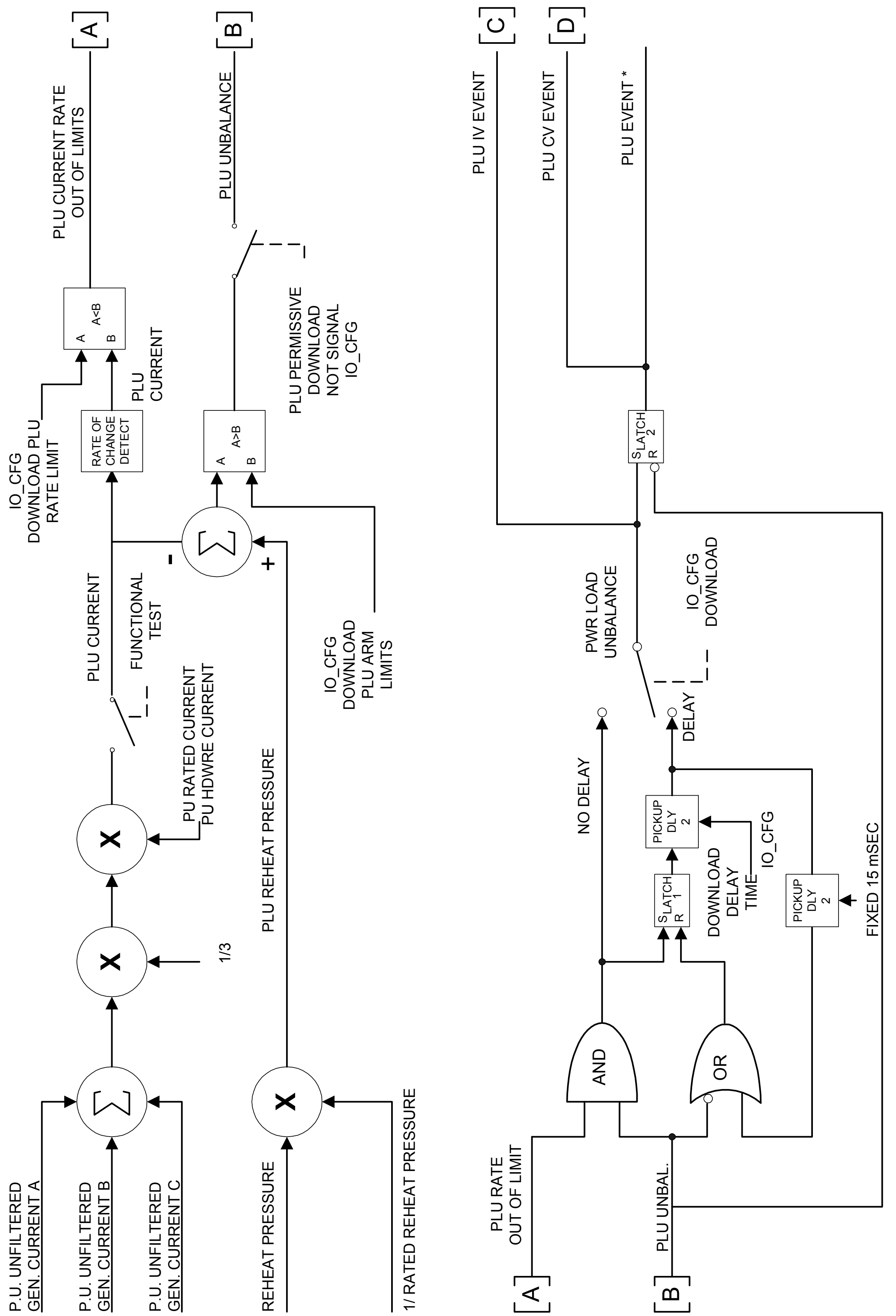


Figure 7-39. Valve Actuation Logic

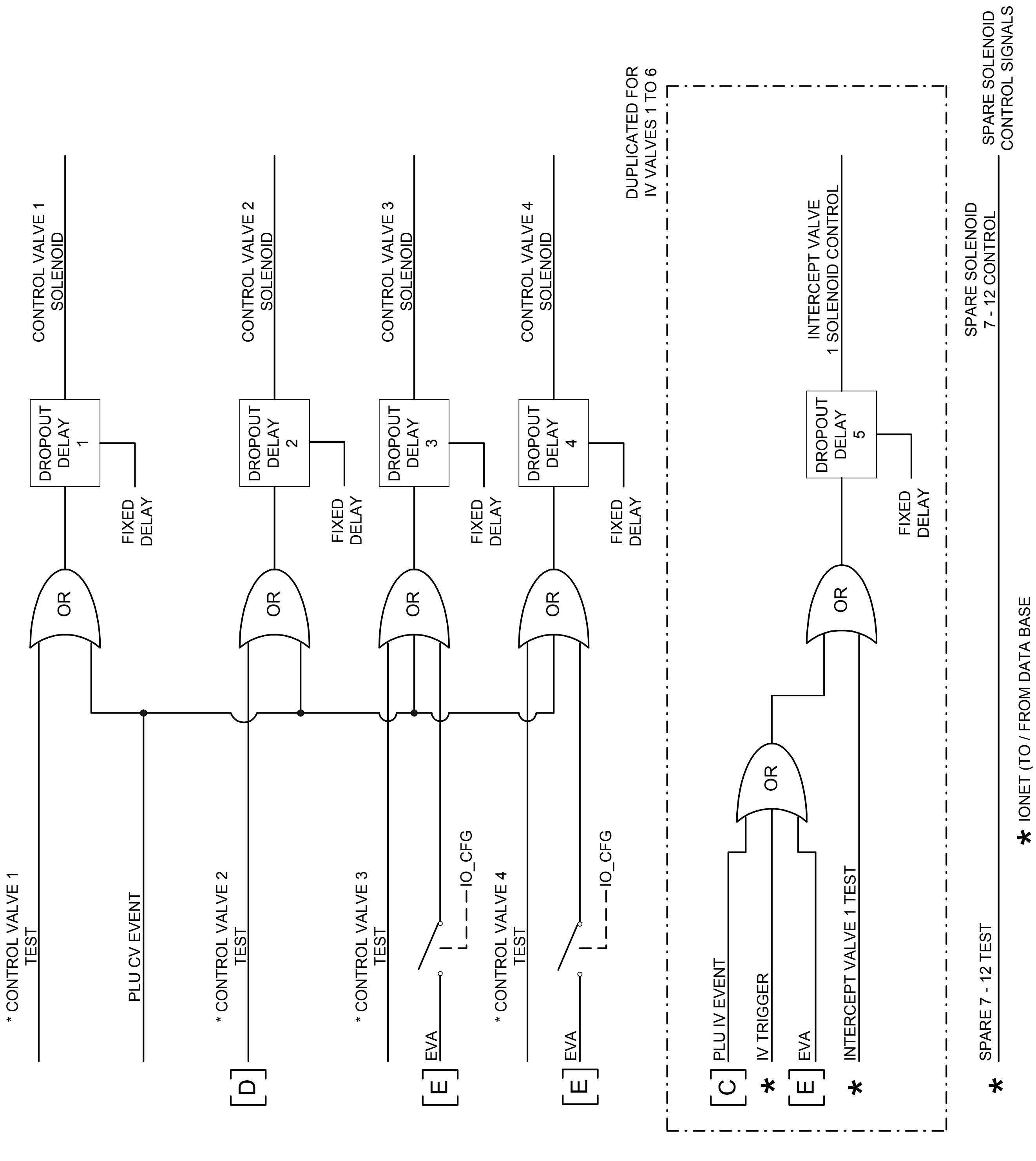


Figure 7-40. EVA Functional Diagram

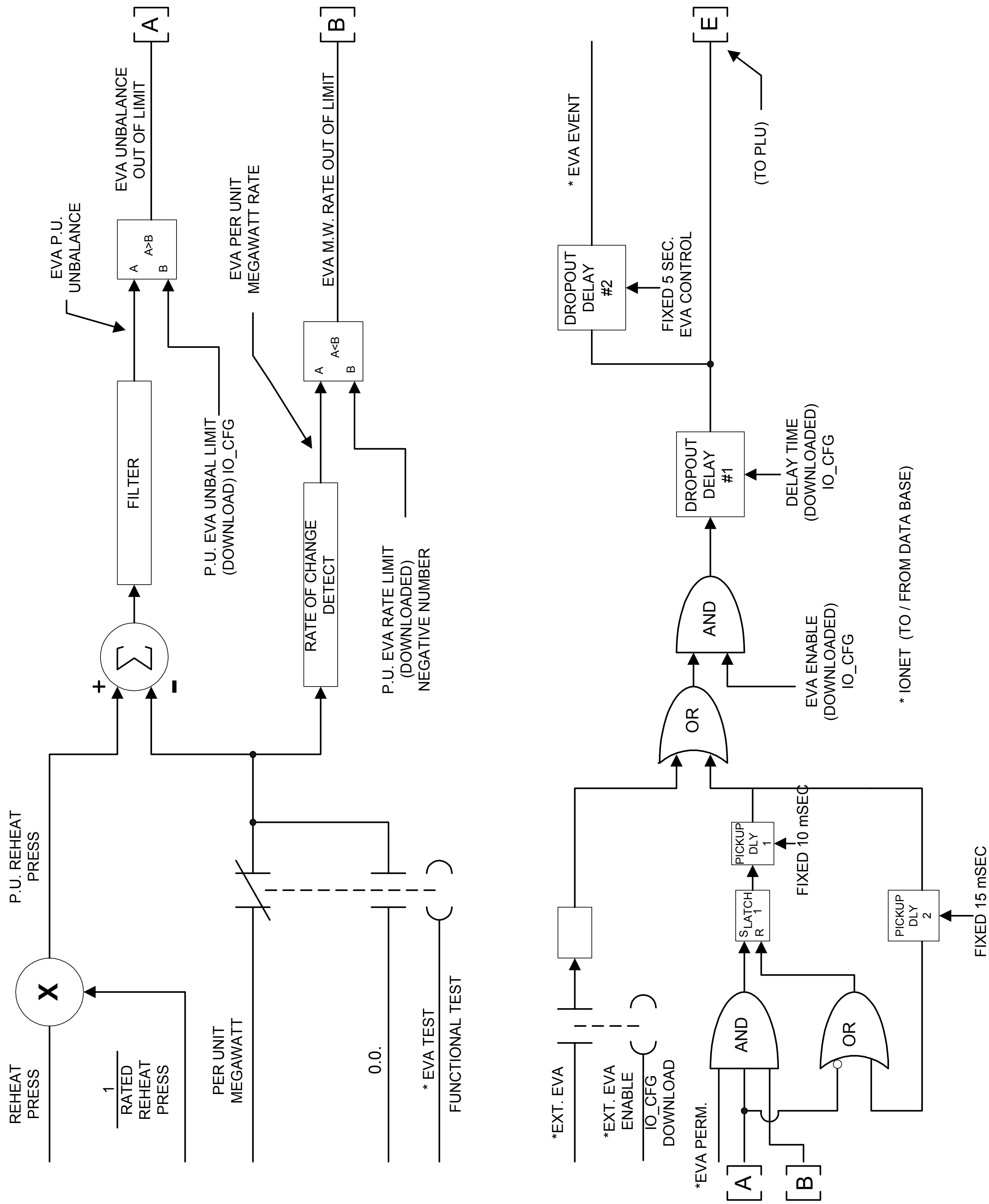


Figure 7-41. EVA Functional Diagram

7-12.1. Intercept Valve Trigger

The peak speed following rejection of 10% or greater rated load cannot be maintained within limits on some units by the normal speed and servo control action. Approximately 70% of turbine power is generated in the Reheat and Low Pressure turbine sections (the boiler re-heater volume represents a significant acceleration energy source). Fast closing of the IVs can therefore quickly reduce turbine power and peak overspeed. The action fulfills the first basic function of normal overspeed control—limiting peak speed. The IVT signal is produced in the CSP as a result of the IVT algorithm and associated sequencing. Information on this function can therefore be found by examining the CSP documentation. The `IVT` signal is shown in Figure 7-40.

7-12.2. Early Valve Actuation (EVA)

The EVA function may be implemented on sites where instability, such as loss of synchronization, presents a problem. An active controlling device, EVA closes the IVs for approximately one second upon sensing a fault that is not a load rejection. This action reduces the available mechanical power, thereby inhibiting the loss of synchronization that can occur as a result of increased machine angle (unbalance between mechanical and electrical power). If the fault persists, the generator loses synchronization and the turbine is tripped by the overspeed control or out-of-step relaying.

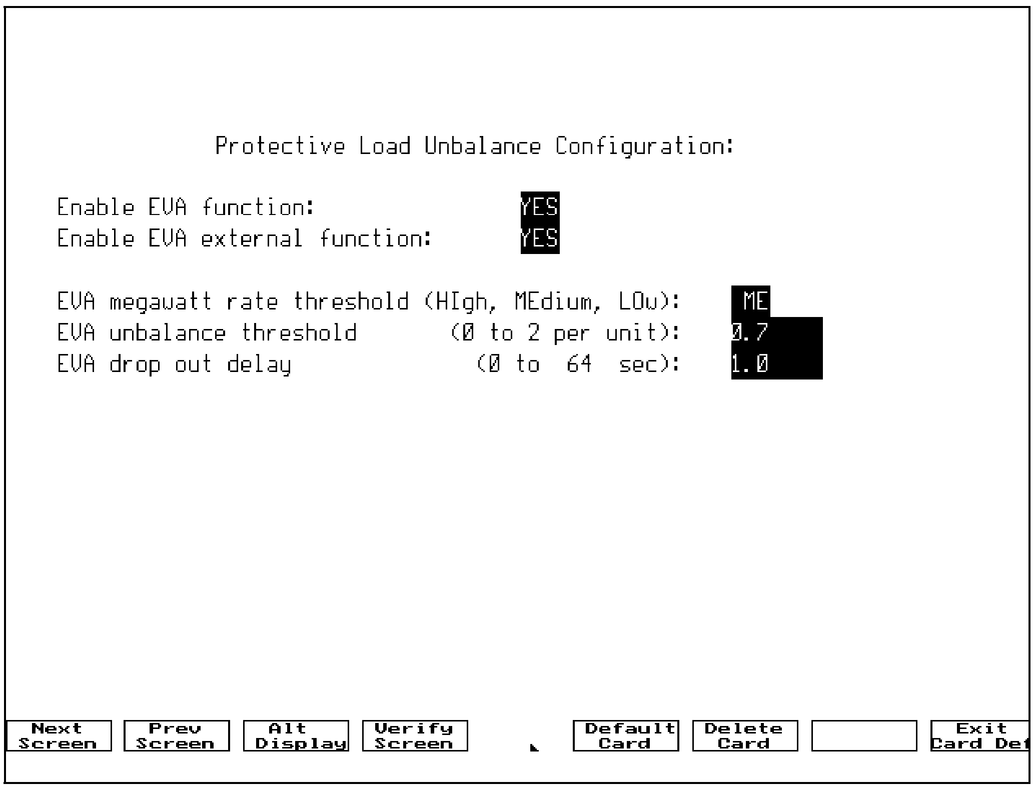


Figure 7-42. I/O Configuration Screen Depicting EVA Configuration Information

The EVA is enabled by selecting `Next Screen` from the Configurator screen shown in Figure 7-38. The screen in Figure 7-42 is revealed. Type "YES", on the `Enable EVA function` command targets. The conditions for EVA action are as follows:

1. The difference between mechanical power (reheat pressure) and electrical power (megawatts) exceeds the I/O Configurator `EVA unbalance threshold` input value.

2. Electrical power (megawatts) decreases at a rate equivalent to (or greater than) one of three rates defined via the I/O Configurator function (`EVA megawatt rate threshold`). This value is adjustable according to three settings: HIgh, MEdium, and LOw. These settings correspond to 50, 35, and 20 millisecond rates respectively.

NOTE

The megawatt signal is derived with voltage and current signals provided by customer-supplied transformers located on the generator side of the circuit breaker.

The `EVA unbalance threshold` value represents the largest fault a particular generator can sustain without losing synchronization. Although the standard setting for this constant is 70%, it may be adjusted up or down 0 to 2 per unit as indicated by the I/O Configurator screen in Figure 7-42. Refer to Figure 7-41 for a block diagram description. All EVA events are annunciated.

7-13. AUTOMATIC TURBINE STARTUP

Automatic Turbine Startup (ATS) monitors the turbine and, if in automatic mode, provides thermal stress based control of Fossil Steam Turbine startups. It also generates recommendations for use by an operator. ATS is a cost option requiring additional hardware on the turbine in addition to software described herein.

This document covers configuration and use of the ATS product. For information on the cause and effect of thermal stress and the rationale for Turbine Automatic Control, refer to the GE document GEK-92609, Thermal Stress Control.

The details of ATS have been broken into several sections. Functions cover the details of what each component of ATS provides and refers to the Control Constants and I/O Requirements sections when discussing specific signals and their use. The Control Constants section, Appendix F, discusses the general approach used with constants and each individual signal. I/O Requirements, also in Appendix F, lists each I/O point and covers how each is used.

7-13.1. Modes of Operation

ATS has two modes of operation, Monitor and Control. Monitor is the default mode of operation. The turbine is controlled by the operator, while recommendations, alarms and events are generated by ATS. Control mode of operation is entered if the operator selects Automatic. When in Automatic, ATS controls certain aspects of the turbine, explained in the Functions section. All features of Monitor mode continue in Control mode. Provisions have been made for ATS to transfer to Semi-Automatic and leave Control mode. This will happen when stable conditions for control by ATS are not met. Refer to Scheduling under the Functions section for more information.

7-13.2. Functions

Functions in ATS may run, based on scheduling and turbine conditions. Functions can also be disabled by control constant settings. This is covered in detail for each ATS function.

Each function runs at one of four predetermined rates. These rates are base, fast, slow, and fundamental. Base rate functions run once per second, the base cycle time of ATS. Fast rate functions run once every four seconds, while slow rate functions run once every fifteen seconds. Fundamental rate functions run once per minute.

The functions of ATS produce alarms and events based on turbine conditions. The alarms and events can produce holds when the turbine is running in automatic. Holds in the system are triggered by the Hold List function; refer to the Maintenance Manual, GEH-5980, Chapter 4 for additional details.

7-13.3. Scheduling

The scheduling function monitors turbine conditions, control constants and elapsed time to determine if and when the other functions are to run. Key to scheduling are the critical control constants. If any of them are unacceptable values, no other functions will be scheduled to run. Refer to the Control constants section in Appendix F for details of the critical control constants.

Scheduling is responsible for determining whether the stress calculations are stable. This is based on the amount of time the calculations have been able to run and the temperature differentials from the last known calculations. Also provided is the minutes remaining until stress calculations are stable. If thermocouple data is bad, stress calculations are not considered stable.

This is a base rate function and is always enabled.

7-13.3.1. CONTROL CONSTANTS: KATS_CONFIG, KCRITERIA1, KCRITERIA2, KCRITERIA3, KCRITERIA4, KSTABL_TM, KSTART_TM, KTURBTYPE.

FUNCTION I/O:

Inputs: L14HA, L3XT_ATS, L52GX_ATS, RT_RBS, TT_FSS, TT_XOU.

Outputs: LCALCS_STAB, L83STAB, RUN_TIMER, TIME_TO_STAB.

7-13.4. Rotor Stress

This function provides three fundamental pieces of rotor data on which most recommendations and decisions are based: bore temperatures, surface stresses and bore stresses. Distinct geometric and material factors for the different locations in the turbine are represented by various control constants. Also, a special sign convention is used for thermal stresses. Positive values represent heating and negative values cooling.

An indicator of the severity of a thermal cycle from heating through cooling is derived. Also, the accumulation of results from each thermal cycle is kept. These measurements of the cyclic life expenditure are calculated in percentage of available life of the rotor.

Bore stress excursions are when bore stress limits are exceeded. Counters of bore stress excursions are recorded by this function. The severity of the excursion is registered by three zones.

To accurately calculate stresses and rotor bore temperatures, this function needs at least one hour of uninterrupted rotor surface temperature history. This duration is adjusted by the stabilization time control constant KSTABL_TM.

Bore temperatures are used to determine the need for and progress of rotor prewarming. They are also used to determine the allowable bore stress. Comparison of the rotor stresses with allowable stress limits determines acceleration rates, speed holds for heat soaking and loading rate calculations.

This is a fundamental rate function and is enabled as long as the thermocouple data has been good for a period longer that the time defined by the startup time control constant KSTART_TM.

7-13.4.1. CONTROL CONSTANTS: KALF0, KALF1, KALF2, KALF3, KBAL_1, KBAL_2, KBAL_3, KBAL_4, KBAL_5, KBOREDIA1, KBOREDIA2, KBOREDIA3, KCT_1, KCT_2, KCT_3, KC0_1, KC0_2, KC0_3, KC1_1, KC1_2, KC1_3, KC2_1, KC2_2, KC2_3, KC3_1, KC3_2, KC3_3, KDF0_1, KDF0_2, KDF0_3, KDF1_1, KDF1_2, KDF1_3, KDF2_1, KDF2_2, KDF2_3, KDF3_1, KDF3_2, KDF3_3, KEMODCST_1, KEMODCST_2, KEMODCST_3, KEM0, KEM1, KEM2, KEM3, KK2_1, KK2_2, KK2_3, KPOISSON, KRATSPD, KSAL_1, KSAL_2, KSAL_3, KSAL_4, KSHFTDIA, KSURFDIA1, KSURFDIA2, KSURFDIA3, KS2N_1_1, KS2N_1_2, KS2N_1_3, KS2N_2_1, KS2N_2_2, KS2N_2_3, KS2N_3_1, KS2N_3_2, KS2N_3_3, KWRIMDIA.

FUNCTION I/O:

Inputs: L83CLE_HI, L83CLE_LO, L83CLE_ME, L83STAB, RT_RBS, TN_RPM, TT_FSS, TT_XOU.

Outputs: BORE_LA_1, BORE_LA_2, BORE_LA_3, BORE_STR_1, BORE_STR_2, BORE_STR_3, CLE_HP, CLE_RH, CLE_XO, DBZONE_HP_1, DBZONE_HP_2, DBZONE_HP_3, DBZONE_RH_1, DBZONE_RH_2, DBZONE_RH_3, DBZONE_XO_1, DBZONE_XO_2, DBZONE_XO_3, DS7_ATS_1, DS7_ATS_2, DS7_ATS_3, DS9_ATS_1, DS9_ATS_2, DS9_ATS_3, LBZCOUNT, LSURFHEAT1, LSURFHEAT2, LSURFHEAT3, RLE_HP, RLE_RH, RLE_XO, ROTOR_1_TEMP, ROTOR_2_TEMP, ROTOR_3_TEMP, SURF_LA_1, SURF_LA_2, SURF_LA_3, SURF_STR_1, SURF_STR_2, SURF_STR_3, S2_ATS_1, S2_ATS_2, S2_ATS_3.

7-13.5. Preparation for Rolloff

This function is mainly concerned with rotor prewarming and chest warming. If gives guidance to an operator, but takes no action other than to request holds if conditions dictate.

Turbine rotor prewarming is considered to be a manual operation which can be done with some computer guidance as to prewarming progress. The purpose is to assure that the rotor bore material has enough ductility for the centrifugal stresses during acceleration to high hold speed. After all three minimum temperatures have been reached and the rotor warming mode has been turned off, chest warming and proper warming rate should be selected on the control panel. Rotor prewarming holds are enabled with the control constant KLRTR_PWRM.

Chest warming consists of two phases, control valve chest pressurization and heat soaking. First it is determined if slow chest pressurization is required or if the stop valve internal bypass valve can be opened rather fast by selecting the proper warming rate. It is determined if chest warming continues satisfactorily or if the difference between inner and outer wall temperatures is large enough to warrant a hold in chest warming. If the chest pressure reaches 85% of main steam pressure, the pressurization phase of chest warming is finished and the heat soaking phase will begin in this second phase. The control valve chest is ready for turbine rolloff if the differences between main steam temperature and chest outer wall temperature and between inner and outer wall temperatures have decayed sufficiently enough to prevent any excessive temperature differences across the chest wall during turbine acceleration. Chest warming holds are enabled with the control constant KLCST_PWRM.

High stress holds will occur if the surface of bore stresses exceed 50%. They can be suppressed with the control constant KLPRE_STRS.

This function also calculates several values to help the operator in determining the progress of warming of the rotor and chest. They are heating rates, main steam to outer control valve temperature differential and target as well as chest to main steam pressure ratio.

This is a fast rate function. It is enabled when the breaker is open and the turbine is below rolloff speed. However, there are several control constants to disable portions of the function.

7-13.5.1. CONTROL CONSTANTS: KDEL_CV_1, KDEL_CV_2, KDEL_CV_3, KDEL_CV_4, KDEL_CV_5, KLCST_PWRM, KLPRE_STRS, KLRTR_PWRM, KMINRTMP1, KMINRTMP2, KMINRTMP3, KPREP1, KPREP2, KPREP3, KPREP4, KPREP5, KPREP6, KRATMSP.

FUNCTION I/O:

Inputs: CP_CHST_ATS, FP_MSP1, FT_MSP, L83CW, VT_CVI, VT_CVO, VT_CV_ALLOW.

Outputs: CHST_MSP_PCT, HEAT_RATE_1, HEAT_RATE_2, HEAT_RATE_3, LCHST_RDYRL, LCHST_WRM_CM, LCHT_PRE_REQ, LCV_CHST_HLD, LEXCSTR_PR, LHEAT_SK_REQ, LPRESS_OVER, LROT_PRE_REQ, LSLO_PRS_REQ, MSOTR_DIFTAR, MST_OUTR_DIF, TSAT.

7-13.6. Acceleration

This function controls the speed and acceleration rate selections of the turbine while in Control Mode, in accordance with prewarming requirements and thermal stress level limitations. Recommendations are determined in both modes of operation. Acceleration also calculates lookahead stresses, but the values are only displayed while close to speed targets.

Acceleration may request speed holds to conduct heat soaks. This is to reduce thermal stresses in the rotor, and if necessary, to permit the proper warming of the generator field. Lookahead stresses may also cause speed holds due to steam to metal temperature mismatches. This is to limit thermal stresses when accelerating to speed targets.

This is a fast rate function and is enabled when the breaker is open and the turbine is above rolloff speed..

7-13.6.1. CONTROL CONSTANTS: KACC1_1, KACC1_2, KACC1_3, KACC2_1, KACC2_2, KACC2_3, KAK_1, KAK_2, KAK_3, KATS_A_1_1, KATS_A_1_2, KATS_A_1_3, KATS_A_2_1, KATS_A_2_2, KATS_A_2_3, KATS_A_3_1, KATS_A_3_2, KATS_A_3_3, KATS_B_1_1, KATS_B_1_2, KATS_B_1_3, KATS_B_2_1, KATS_B_2_2, KATS_B_2_3, KATS_B_3_1, KATS_B_3_2, KATS_B_3_3, KATS_BA_1, KATS_BA_2, ATS_BA_3, KATS_BB_1, KATS_BB_2, ATS_BB_3, KEMERG_SET, KK2_1, KK2_2, KK2_3, KOS_STMARGIN, KOS_MARGIN, TNKR_HI, TNKR_LO, TNKR_ME.

FUNCTION I/O:

Inputs: L14HA, L14RTX, L68DW, L83A_HI, L83A_LO, L83A_ME, L83FF, L83M_AU, L83M_BYP, L83N_CL, L83N_HI, L83N_ME, L83SF, TN_RPM, TNH1.

Outputs: E_L18ACC_REC, LEXC_LA_STR, LRTR_WRM_REQ, L3_OS_ATS, L43A_HIR_ATS, L43A_LOR_ATS, L43A_MER_ATS, L43N_HIR_ATS, L43N_MER_ATS, L43N_RTR_ATS.

7-13.7. Loading Rate

This function calculates the loading rate used during automatic loading. The rate of change of rotor stresses and steam temperatures and the level of rotor stresses are used to calculate a loading rate such that turbine rotor stresses are not exceeding limits. The maximum allowable loading rate is limited by the operator selected loading rate.

During load increase and below 10% of rated load, loading rate is limited to the initial loading rate. The initial loading rate constitutes a loading rate the turbine will be able to sustain over the entire loading range with conservative margin.

The calculated loading rate constitutes an optimal loading rate. Exceeding the calculated loading rate during manual or automatic operation will lead to over stressing of the turbine rotors. Customers should use the calculated loading rate for manual operation and variable boiler pressure operation.

This is a slow rate function and is enabled when the breaker is closed and the turbine is above rolloff speed.

7-13.7.1. CONTROL CONSTANTS: KFB_1, KFB_2, KFB_3, KFR1, KFR2, KFR3, KFR4, KFR5, KFS_1, KFS_2, KFS_3, KRATLOAD, KVMULT.

FUNCTION I/O:

Inputs: DW1, FP_MSP1, L3DWR_UNL, L83M_AU, RT_RBS, TT_FSS.

Outputs: DWR_TARR_MAX, LOAD_RAT_REC, LOADRATE_ATS.

7-13.8. Loading Monitor

This function monitors turbine conditions and requests load holds if deemed necessary. It also will request a transfer to forward flow if the correct conditions are met.

Load holds can be suppressed when below a minimum percent load by use of control constants. The control constant KLMIN_LOAD enables the suppression and the control constant KMIN_LOAD determines the percentage below which no hold will be requested.

A load hold will be requested if control or stop valve temperature differentials are beyond allowable limits. This request can be suppressed with the control constant KLTMP_LOAD. High stresses will also cause a load hold to be requested.

Requests to transfer to forward flow will only be issued when intercept value references are greater than 50% and rotor stresses are within acceptable ranges to not cause excessive stress when the transfer occurs.

This is a fast rate function and is enabled when the breaker is closed and the turbine is above rolloff speed..

7-13.8.1. CONTROL CONSTANTS: KK2_1, KK2_2, KK2_3, KKLAB, KKLAS, KLD1, KLD2, KLMIN_LOAD, KLTMP_LOAD, KMIN_LOAD, KRATLOAD.

FUNCTION I/O:

Inputs: DW1, IVR, L3DWR_UNL, L68DW, L83FF, L83M_AU, L83M_BYP, VT_CVI, VT_CVO, VT_MSVI_ATS, VT_MSVO_ATS, VT_MSV_ALLOW, VT_CV_ALLOW.
Outputs: LEXC_VLV_DIF, LROT_HS_REQ, L43FFR_ATS.

7-13.9. Generator Monitor

This function calculates armature and field current limits, and requests holds if either of these limits are exceeded. A hold will also be requested if the cold gas temperature exceeds a limit.

It is necessary to calculate the field temperature if one is not provided. If LCALC_TEMP is TRUE, the field temperature will be calculated. If the calculated field temperature exceeds the field temperature limit, a hold will be requested.

This is a base rate function and is enabled when the breaker is closed. There are control constants to disable the entire Generator Monitor function or just the field temperature calculation.

7-13.9.1. CONTROL CONSTANTS: KFTEMP_LIM, KH2PRES_1, KH2PRES_2, KH2PRES_3, KH2PRES_4, KIAL_1, KIAL_2, KIAL_3, KIAL_4, KIFL_1, KIFL_2, KIFL_3, KIFL_4, KLGEN_CALC, KRATCUR, KRESIST.

FUNCTION I/O:

Inputs: AP_H2_ATS, AT_H2CG, DA_PU_AV, DAF_ATS, DVF_ATS, LCALC_TEMP.

Outputs: ARM_CUR_LIM, FLD_CUR_LIM, FLD_TEMP, LARM_CUR_EXC, LAT_H2CG_EXC, LFLD_CUR_EXC, LFLD_TMP_EXC.

7-13.10. EEPROM Access

This function accesses the EEPROM storage area to record historical data. This is done on demand for data that rarely changes, or every half hour for data that changes frequently. It also retrieves the historical data that must be preserved when power to the panel is lost.

This is a base rate function and is always enabled.

7-13.10.1. CONTROL CONSTANTS: None.

FUNCTION I/O:

Inputs: DBZONE_HP_1, DBZONE_HP_2, DBZONE_HP_3, DBZONE_RH_1, DBZONE_RH_2, DBZONE_RH_3, DBZONE_XO_1, DBZONE_XO_2, DBZONE_XO_3, RLE_HP, RLE_RH, RLE_XO.

Outputs: Same as Inputs.

7-13.11. Admission Mode Selection

The AMS function is used to control rotor stresses at HP (High Pressure) 1st stage outlet. AMS will transfer between FA (full arc) and PA (partial arc) whenever rotor stresses have to be limited, and such action is thought to be of value as described below. After startups and load changes, except for complete shutdowns or permanent operation below a load of 10% of rated MW load, AMS will transfer to PA at a selected rate of change of admission mode reference. Different levels of HP rotor stresses have been considered.

The admission mode will be changed using the following logic, considering different turbine operation.

1) Startup and Loading
   a) Below and equal to 10% load always in FA.
   b) Above 10% rated load, admission mode will be changed to limit HP 1st stage rotor stresses.
   c) Above 80% of target load, any further transfer to FA is prevented because it is unnecessary.
   d) After Target Load is reached, transfer to PA per 2d

2) Major Load Increase (>= 40% load)
   a) Operator could transfer to FA at slow transfer rate at least 30 minutes before the start of a load increase to keep stress generated during transfer to a minimum.
   b) If not at FA at start of load increase, transfer to FA will be initiated under AMS stress control with maximum transfer rate of 10% per minute.
   c) Above 80% of target load, see 1c.
   d) After target load is reached, transfer to PA is initiated.

3) Minor Load Increase (< 40% load)
   This load change is so small that the normal stress control with AMS is not required. AMS will be programmed to stay at the current admission mode or transfer to PA, if required by AMS stress control. No transfer to FA for HP rotor heating occurs.

4) Shutdown
   a) For shutdown the operator could select 10% or less for target load.
   b) For shutdown or continuous operation below 10% load, both will be in FA.
   c) Initialization is described in 5a.

5) Major Load Decrease (>= 40% load)
   a) Initially a transfer to FA will be made. If the initial admission mode is FA, it should stay in FA. For this initial transfer, a control stress lower stress limit of a smaller value will be used and the transfer rate will be set to only 10% per minute to prevent excessive stress levels.
   b) Unloading will occur in FA.
   c) After target load is reached and this target load is above 10%, a transfer to PA will be implemented. PA is desired because of the better heat rate.

6) Minor Load Decrease (< 40% load)
   This load change is so small that normal stress control with AMS is not required. AMS will be programmed to stay at the current admission mode. Excessive HP rotor stresses during small load reductions are less likely because during unloading, boilers will generate approximately constant throttle temperatures.

This function should not be entered if the turbine has one or two admissions only because the benefits of using AMS for rotor stress control are negligible.

This is a slow rate function and is enabled when the breaker is closed and the turbine is above rolloff speed. There is a control constant to disable this function, KLAMS_CALC.

7-13.11.1. CONTROL CONSTANTS: KAD, KAMS1, KAMS2, KAMS3, KAMS4, KAMS5, KAMS6, KAMS7, KLAMS_CALC, KRLIM.

FUNCTION I/O:

Inputs: CVAMS_AU, CVR, DW_TTA_ATS, DWR_R1, DWR_TAR2, DW1, L3DWR_LDG, L3DWR_UNL, L39DE_LW, L83FF, L83M_AU, L83M_CO, L83M_UMC, L83M_VPO.

Outputs: AMS_RATE_REC, DAR_ATS, E_L83AMS_REC, L43AMS_F_ATS, L43AMS_P_ATS.

7-13.12. Temperature Mismatch

This function calculates steam to metal mismatch temperatures. Due to the inability to measure temperatures directly, the first-stage discharge and reheat exhaust (also called crossover) steam temperatures are calculated here.

Steam to metal mismatch temperatures are used in determining lookahead stresses in the Acceleration function, as well as determining conditions to transfer to forward flow in the loading Monitor function.

This is a slow rate function and is enabled when the turbine is above rolloff speed.

7-13.12.1. CONTROL CONSTANTS: KRATMSP.

FUNCTION I/O:

Inputs: FP_MSP1, FP_HRSP_DPY, FT_HRS, FT_MSP, L52GX_ATS, L83FF, L83SF, RT_RBS, TT_FSS, TT_XOU.

Outputs: LENTHALPY_OK, TM_ATS_1, TM_ATS_2, TM_ATS_3.

7-13.13. Average Temperature Change

This function takes the latest main and reheat steam temperatures reading and averages it with the history of previous readings to calculate an average temperature change for areas to be evaluated. History is weighted three times as heavily as the most recent reading. It also calculates the allowable stop valve and control valve temperature differences, along with the control and stop valve inner/outer temperature differential.

This is a base rate function and is always enabled.

7-13.13.1. CONTROL CONSTANTS: KDEL_CV_1, KDEL_CV_2, KDEL_CV_3, KDEL_CV_4, KDEL_CV_5, KDEL_CVU_2, KDEL_CVU_3, KDEL_CVU_4, KDEL_CVU_5, KDEL_SV_2, KDEL_SV_3, KDEL_SV_4, KDEL_SV_5.

FUNCTION I/O:

Inputs: DTMS_ATS, DTRH_ATS, FT_HRS, FT_MSP, L3DWR_UNL, VT_CVI, VT_CVO, VT_MSVI_ATS, VT_MSVO_ATS.

Outputs: DTMS_ATS, DTRH_ATS, VT_MCV_DIFF, VT_MSV_DIFF, VT_CV_ALLOW, VT_MSV_ALLOW.

7-13.14. Temperature Recommendations

This function calculates the minimum and maximum recommended steam temperatures for the high pressure and reheat areas. If the actual temperatures are outside of the recommended ranges, an event will be logged.

This is a slow rate function and is enabled when the turbine is below rolloff speed.

7-13.14.1. CONTROL CONSTANTS: KATS_BA_3, KATS_BB_3, KFNMAX_1, KFNMAX_2, KFNMIN_1, KFNMIN_2, KFPOSB_1, KFPOSB_2, KFPOSS_1, KFPOSS_2, KK2_1, KK2_2, KK2_3, KMINMAXT_1, KMINMAXT_2, KRATRHT, KRATMSP, KRATMST, KS2N_1_1, KS2N_1_2, KS2N_1_3, KS2N_2_1, KS2N_2_2, KS2N_2_3, KS2N_3_1, KS2N_3_2, KS2N_3_3.

FUNCTION I/O:

Inputs: FP_MSP1, FT_HRS, FT_MSP, L83CLE_HI, L83CLE_ME, L83FF, RT_RBS, TT_FSS.

Outputs: E_FT_MST_REC, E_FT_RHT_REC, FT_HRS_MAX, FT_HRS_MIN, LMST_OUT_RAN, LRHT_OUT_RAN, STEAM_MAX, STEAM_MIN.

7-13.15. CONTROL CONSTANTS

While the control constants are actually part of the I/O, they are important enough to warrant their own section, see Appendix F. All control constants are inputs to ATS and required for proper operation.

Control constants will change on a requisition basis. This means that once the correct values have been entered, they should not have to be changed. For this reason, no values have been listed with the summary in Appendix F.

7-13.16. Critical Constants

Critical control constants are constants that must be correct for ATS to function. These constants must be within the defined range of allowable values or the scheduling function will not let any other function run.

The following constants are critical: **KATS_CONFIG, KSTABL_TM, KSTART_TM, KTURBTYPE, KLOCATIONS, KCRITERIA1, KCRITERIA2, KCRITERIA3, KCRITERIA4.** Refer to the Individual Constant Summary section to determine the allowable values for the critical constants.

7-14. PLANT LOAD CONTROL

The Plant Load Control option implements a background program running in the <I> processor that can send individual `MW RAISE` and `MW LOWER` commands to regulate each turbine in a multi-turbine site to maintain a total plant load, or the balance of load between turbines.

7-14.1. Adding Plant Load Control to an <I> Processor

PLC is a function that runs in the <I> processor. PLC allows the user to define a plant MW setpoint, and the <I> will send the needed raise and lower commands to the individual units to match the plant setpoint. Balance control is also available which will balance the output of units so that they are all producing the same percentage of rated capacity. Individual units can be enabled or disabled for PLC, so some units can be running at a defined load (such as base or peak) and the rest of the units can be used to match the plant megawatt setpoint.

To implement PLC on an <I>, the following steps are necessary:

1) Define the <I> processor as a unit.
2) Define a data dictionary for the <I> processor.
   a) Define the UNITDATA.DAT file.
   b) Define the SCLEDATA.DAT file.
   c) Define the ENUMDATA.DAT file.
3) Create a PLC configuration file to indicate how the PLC is to operate.
4) Create the desired displays to view and control the PLC program.
5) Indicate to the <I> that PLC should be started upon restart.

This document explains how to go about these steps, and includes example files for a three unit site.

7-14.1.1. DEFINE THE <I> PROCESSOR AS A UNIT. The F:\CONFIG.DAT file contains a list of the units that exist at this site. PLC requires a unit to be added for the <I> processor. The <I> processor unit holds all the PLC data points. The <I> unit is added like any other unit EXCEPT the <I> unit is given a unit type field of "1". This tells the <I> that this unit is the <I> itself. Note that the UNIT TYPE field is optional, and defaults to a MARK V unit type (a value of zero) if not specified.

For example, to add the <I> to a three unit site, change F:\CONFIG.DAT from:

```
+-----------------------------------------------------+
|;                                                    |
|;      UNIT    UNIT     PATH TO           UNIT |
|;     NUMBER   NAME     CONFIG DATA       TYPE |
|;     ------   ----     ---------------   ----    |
|UNIT_DATA                                            |
|        1       T1      F:\UNIT1           0   |
|        2       T2      F:\UNIT2           0   |
|        3       T3      F:\UNIT3           0   |
+-----------------------------------------------------+
```

To:

```
+-----------------------------------------------------+
|;                                                    |
|;      UNIT    UNIT     PATH TO           UNIT |
|;     NUMBER   NAME     CONFIG DATA       TYPE |
|;     ------   ----     ---------------   ----    |
|UNIT_DATA                                            |
|        1       T1      F:\UNIT1           0   |
|        2       T2      F:\UNIT2           0   |
|        3       T3      F:\UNIT3           0   |
|        4       I1      F:\UNIT4           1   |
+-----------------------------------------------------+
```

Please note that the <I> unit should not be added into the NETWORK_DATA section of the CONFIG.DAT file, only the UNIT_DATA section.

7-14.1.2. DEFINE A DATA DICTIONARY FOR THE <I>. The data dictionary consists of three files for the <I>. The UNITDATA.DAT file defines all the <I> points, the SCLEDATA.DAT defines the scaling for these points, and the ENUMDATA.DAT file defines the strings associated with the enumerated state points.

There is no alarm scanning done in the <I>, so it is not necessary to create an ALARM.DAT file. The data dictionary loader will comment about the missing ALARM.DAT file while loading the <I> data dictionary, but this can be ignored. (An ALARM.DAT file can be created with an unused alarm in it to silence the dictionary loader's warning.)

The following pages define the contents of these three required files along with an example of a three unit site.

Define the UNITDATA.DAT file. The UNITDATA.DAT file defines the points in the <I> data dictionary. The points included here are used to support the PLC function. The following example can probably be used as-is, with the number of PLCSTATE_n points updated to match the number of units that really exist in the plant. (PLCSTATE_1 is the PLC STATE of unit 1, PLCSTATE_2 is the PLC STATE of unit 2, and so on). Figure 7-43 shows an example UNITDATA.DAT file.

```
;  <I> data dictionary for example Plant Load Control (PLC) site.
;
;
#point_data
;
;   Memory Offset (Hex) ----------------------------------+
;   Memory Segment (Hex) ----------------------------+     |
;   Flags (Hex) ------------------------------+     |     |
;   High/low limits type -----------------+     |     |     |
;   Plotting limits type -----------+     |     |     |     |
;   Scale code type ----------+     |     |     |     |     |
;   Point type ---------+     |     |     |     |     |     |
;   Point Number -+     |     |     |     |     |     |     |
;   Name          |     |     |     |     |     |     |     |
<---------->    <---> <-> <--> <--> <--> <--> <--> <-->
PMW              0001  003  0001  0000  0000  0000  0000  0000
PMWSP            0002  003  0001  0000  0000  0006  0000  0004
PMWERR           0003  003  0001  0000  0000  0000  0000  0008
PSPDB            0004  003  0001  0000  0000  0006  0000  000C
BALDB            0005  003  0001  0000  0000  0006  0000  0010
PMVAR            0006  003  0002  0000  0000  0000  0000  0014
SCANINT          0007  005  0003  0000  0000  0006  0000  0018
PLCSTATE         0008  009  0001  0000  0000  000A  0000  001A
BALSTATE         0009  009  0001  0000  0000  000A  0000  001C
PSPMATCH         0010  001  0000  0000  0000  0000  0000  001E
BALMATCH         0011  001  0000  0000  0000  0000  0000  001F
PLCSTATE_1       0012  009  0001  0000  0000  000A  0000  0020
PLCSTATE_2       0013  009  0001  0000  0000  000A  0000  0022
PLCSTATE_3       0014  009  0001  0000  0000  000A  0000  0024
```

Figure 7-43. UNITDATA.DAT File Example

Define the SCLEDATA.DAT file. The SCLEDATA.DAT file defines the various scale codes used in displaying points in the <I> data dictionary. There are very few scale codes actually used in the <I> data dictionary. Because the <I> uses floating point numbers for calculations (not fixed point like the Mark V) there is no need to have a lot of different scale codes so that the numbers do not go out of range. A floating point number has such a large range that the scale code used for total plant megawatt can be the same scale code that is used for megawatt deadbands. This limits the number of scale codes required.

Figure 7-44 shows an example SCLEDATA.DAT file.

```
;   SCLEDATA.DAT
;
;This file defines engineering units conversions for each scale code.
;Each line contains a scale code number, gain, offset, decimal places,
;and engineering units string.
;
#scale_data    0         1.000000         0.000000   0   LOGIC   LOGIC
#scale_data    1         1.000000         0.000000   1   MW      MWATT
#scale_data    2         1.000000         0.000000   1   MVAR    MVARS
#scale_data    3         1.000000         0.000000   0   sec     SEC
```

Figure 7-44. SCLEDATA.DAT File Example

Define the ENUMDATA.DAT file. The ENUMDATA.DAT file defines the text strings associated with the enumerated points defined for a unit. The PLC algorithm uses an enumerated state for each unit, the PLC state for the plant, and the BALANCE state for the plant. In this example, they all use the same enumerated state table, which defines the states as "OFF" and "ON".

Figure 7-45 shows an example ENUMDATA.DAT file.

```
;
; Sample enumerated data definition file for <I> unit with PLC.
;
;         <data> <string>   <num>
;         <type> <length> <strings> <attrib> <strings>
#enum_data      1        3      2       0000    #???
                                        0000    #OFF
                                        0000    #ON
;
```

Figure 7-45. ENUMATA.DAT File Example

7-14.1.3. CREATE THE PLC CONFIGURATION FILE. The PLC configuration file defines, for the <I> processor, the constants and points to be used for the PLC functions. This file must match the configuration of the plant in order for the PLC functions to perform as expected.

The PLC data file must be located on the <I> processor as the file F:\PLC.DAT. This follows the standard that all files that configure the <I> processor itself are located at the top level of the F: drive. The file consists of multiple SECTIONS. Each section is started by a section line that starts in the first column. The lines within each section must be indented at least one space. Spaces or tabs can be used to indent each line.

Within each section there are two types of entries. One defines a constant that is required for the PLC calculations, the other defines the name of a point used in the calculations. The constants and the pointnames must each be one "word". Comments can be added after the values as long as there is some white space (one or more spaces or tabs) between the constant value or pointname and the comment. Comments should begin with a semicolon; any line that starts with a semicolon in column one is taken to be a comment line and is ignored.

7-14.1.4. CREATE DISPLAYS FOR PLC. Displays can be created for PLC using either the DEMAND display system or the ANIMATED display system. The DEMAND displays are handy when a few units are desired, but can only support 12 targets. ANIMATED displays are limited to 128 targets, so they are often used for PLC. Animated displays often contain bar graphs showing the MW and MVAR output from each unit along with the targets used to control the PLC programs.

The basic PLC display should allow the user to:
- Enable or disable each unit for PLC commands,
- Enable or disable PLC,
- Set the plant MW setpoint using an analog setpoint target, or
- Enable or disable the balance control.

In addition, some operators can be allowed to:
- Set the plant MW setpoint deadband using an analog setpoint target,
- Set the balance control deadband using an analog setpoint target, and/or
- Set the scan interval for the PLC calculations.

This can be done by enabling or disabling each unit for PLC commands.When a unit is enabled for PLC commands, the PLC program can send RAISE and LOWER commands to this unit for PLC or balance control. No PLC commands will be sent to units not enabled for PLC.

Enabling and disabling a unit for PLC is usually done by two control pushbuttons. One enables PLC, one disables PLC. The feedback signal for these buttons are the PLCSTATE_n signals, so the use has very visible feedback on whether PLC is enabled or disabled for each unit.

NOTE

The DEMAND display annuniciates the enumerated state for feedback, because it can only show two states. Since there are only two PLC states (ON and OFF) this warning can be ignored.

To enable a unit for PLC commands, write a one to the PLCSTATE_n variable for that unit. To disable a unit for PLC commands, write a zero to the PLCSTATE_n variable.

Enabling or disabling PLC. The <I> will only do the PLC calculations if the plant-wide PLC is enabled. To enable PLC, at least one unit must be enabled for PLC. Plant-wide PLC will turn itself off if no units are enabled for PLC, since it can't do anything if no units are allowed to be controlled.

To enable plant-wide PLC, write a one to the PLCSTATE_n variable in the <I> data dictionary. To disable plant-wide PLC, write a zero to the PLCSTATE_n variable. This is usually done with two separate targets, one to enable and one to disable. The feedback signal should be the PLCSTATE_n variable.

Setting the plant MW setpoint using an analog setpoint target. When plant-wide PLC is off, the plant MW setpoint will track the current plant output. This gives a bumpless transfer when PLC is turned on. Once PLC has been turned on, the plant MW setpoint can be changed by writing to the PMWSP point using an analog setpoint target.

To change the plant MW setpoint, create an analog setpoint target that writes to the PMWSP point in the <I> data dictionary.

Enabling or disabling balance control. If PLC is enabled, the balance control can be enabled. Balance control will cause all units enabled for PLC to produce the same percentage of rated output. At least two units must be selected for PLC in order to run balance control. Balance control will turn itself off if less than two units are enabled for PLC.

To enable balance control, write a one to the BALSTATE variable in the <I> data dictionary. To disable balance control, write a zero to the BALSTATE variable in the <I> data dictionary. This is usually done with two separate targets, one to enable and one to disable. The feedback signal should be the BALSTATE variable.

Setting the plant MW setpoint deadband using an analog setpoint target. If the ability to change the plant MW setpoint deadband is desired, you can include an analog setpoint target that writes to the PSPDB variable in the <I> data dictionary. If the user writes a value that is above the PSPDBMAX constant defined in the PLC data file is written, the value will be set to the PSPDBMAX value. If a value below the PSPDBMIN constant defined in the PLC data file is written, the value will be set to the PSPDBMIN value.

To write to the plant MW setpoint deadband, create an analog setpoint target that writes to the PSPDB point in the <I> data dictionary.

Setting the balance control deadband using an analog setpoint target. If the ability to change the balance setpoint deadband is desired, include an analog setpoint target that writes to the BALDB variable in the <I> data dictionary.

If a value above the BALDBMAX constant defined in the PLC data fileis written, the value will be set to the BALDBMAX value. If a value below the BALDBMIN constant defined in the PLC data fileis written, the value will be set to the BALDBMIN value.

To write to the balance setpoint deadband, create an analog setpoint target that writes to the BALDB point in the <I> data dictionary.

Setting the scan interval for the PLC calculations. If the ability to change the scan interval (in seconds) of the PLC calculations is desired, include an analog setpoint target that writes to the SCANINT variable in the <I> data dictionary.

If a value above the SCANINTMAX constant defined in the PLC data file is written, the value will be set to the SCANINTMAX value. If a value below the SCANINTMIN constant defined in the PLC data file is written, the value will be set to the SCANINTMIN value.

To write to the scan interval, create an analog setpoint target that writes to the SCANINT point in the <I> data dictionary.

7-14.1.5. INFORM THE <I> THAT PLC SHOULD BE STARTED UPON RESTART. To tell the <I> to start the PLC programs when the <I> is restarted,t the F:\CONFIG.DAT file must be edited and included in the OPTIONS section the line: "PLC = Yes".

This causes the PLC programs to start when the <I> is restarted.

7-14.2. Keywords

The following pages show the keywords used in the data file and a brief description of what each keyword defines. In the keyword definitions, the attributes of the keyword are shown inside square brackets. These attributes are:

CONSTANT	-	This keyword defines a constant value.
POINT	-	This keyword defines a point name.
REQUIRED	-	This keyword is always required.
CONTROL	-	This keyword is needed for control.

NOTE

A full example of a three unit site (Section 7-14.3) is included after the definition of the keywords. Please note that the values of the constants shown here may not be the same as those needed at a different site.

7-14.2.1. KEYWORD DEFINITIONS. The following is a list of keywords used in the data file and a brief description of what each keyword represents. The attributes of the keyword definitions are shown in brackets.

NUNITS SECTION. This section provides information concerning the number of turbines that are used.

- **NUNITS <integer value> - [CONSTANT, REQUIRED]** This tells the PLC program the total number of units that can be enabled for PLC. This number does not include the unit defined for the <I> processor.

UNIT SECTION. These definitions provide information concerning the individual units and their characteristics.

- **UNIT <unitname> - [CONSTANT, REQUIRED]** This tells the PLC program that another unit definition is being started. The data file is defined this way to allow the "cloning" of units. There will be one unit section per unit, and the number of unit sections should match the NUNITS value.

- **CONTROL <0 or 1> - [CONSTANT, REQUIRED]** This indicates whether this unit can receive RAISE and LOWER commands to change its output. If a unit is disabled for control, its MW and MVAR output will be summed in the plant total, but no commands will be sent to the unit to change its output. (A value of zero indicates no control actions, a one indicates the unit can be made available for PLC actions.)

- **CAP <mw capacity of unit> - [CONSTANT, CONTROL]** The unit CAPacity specifies the rated capacity of the unit. Units are given commands based upon their percentage of rated capacity. For example, when BALANCE control is enabled, the units will balance so they are producing the same percentage of rated capacity, not the same number of MW. When normal PLC commands are sent, a larger percentage of the RAISE or LOWER commands are sent to the units with the highest capacity. The exact value is not critical, but should be reasonable for the turbine generator set.

- **MIN <minimum MW under PLC> - [CONSTANT, CONTROL]** The MINimum MW value defines the lowest MW value that PLC will drive a unit towards. A unit will not be sent A LOWER command if the unit's output is less than this value. This is usually set above zero to prevent a unit from falling into negative power.

- **MWPBDUR <integer value> - [CONSTANT, CONTROL]** The MW PushButton DURation is used to send the RAISE and LOWER pushbutton commands to the Mark V. With each command to push a button, the <I> sends a value of how long to push the button. The duration is specified as the number of sequencing frame scans which have a typical duration of 1/16 or 1/32 of a second. This defines the constant that is sent to the Mark V. The RAISE and LOWER commands are sent from the <I> once per second. If MWPBDUR is defined to push the button for one second, the maximum unit ramp possible is given. If the push button duration is defined for less than a full second, the ramp rate of the unit can be controlled for PLC functions. (This value affects only the PLC function, not any other operator displays that may have been created.)

- **MWPBRATE <mw/pb> - [CONSTANT, CONTROL]** The MW pushbutton RATE defines the unit's response to a single RAISE or LOWER command. The PLC algorithm decides how many MW it wants the unit to move, then divides this value by the MWPBRATE. This results in the number of RAISE or LOWER commands to send to the unit. This value will be effected by a change in the MWPBDUR (defined above) and the ramp rate set in the unit. If the value is too low, the unit may overshoot its intended MW setting. If the value is too high, it will take longer for the plant to zero in on the desired setpoint. The value should be smaller than the plant setpoint deadband, since changes in setpoints are assumed to be in even multiples of this MWPBRATE.

- **MW <pointname> - [POINT, REQUIRED]** MW provides the name of the signal that is the MW output from the unit. It is required for all units.

- **MVAR <pointname> - [POINT, REQUIRED]** MVAR provides the name of the signal that is the MVAR output from the unit. It is required for all units.

- **52G <pointname> - [POINT, REQUIRED]** 52G provides the name of the logic signal that is the state of the breaker. When the plant MW and MVAR are created, only units where the 52G signal is TRUE are included. This prevents sensor drift from units that are not on-line from being included in plant totals.

- **STATE <I_pointname> - [POINT, REQUIRED]** STATE provides the name of the <I> ENUMERATED STATE signal that indicates whether the unit is enabled for PLC or not. This must be a signal in the <I> data dictionary. The signal is usually given the name PLCSTATE_n, where "n" is the unit number. Writing a one to this point will cause a unit to be enabled for PLC actions. Writing a zero to this point will disable the unit from PLC actions.

- **TEMP <pointname> - [POINT, CONTROL]** TEMP provides the name of the signal that indicates the unit is on temperature control. No RAISE commands will be sent to units with this signal TRUE. If definition of this signal is not desired for a unit, any signal from the unit that is always FALSE (like LFALSE) can be used instead.

- **COMMHTHY <pointname> - [POINT, CONTROL]** COMMHTHY provides the name of any signal that comes from the unit. The point chosen is not important, but is used for a communication health check. If the unit specified in this point is not healthy, the unit will be dropped from PLC. If automatically removing a unit from PLC is not desired, specify any point from the <I> data dictionary here. Communication with the <I> is ALWAYS healthy, so the unit is never disabled from PLC due to a communication loss.

- **BASE <pointname> - [POINT, CONTROL]** BASE supplies the name of the signal that indicates the unit was selected for BASE LOAD. When a unit has been selected for BASE, no RAISE or LOWER commands will be sent to the unit. The unit is disabled from PLC when this signal is TRUE. If implementing this feature is not desired, supply a signal name that is always FALSE (such as LFALSE) for this signal.

- **PEAK <pointname> - [POINT, CONTROL]** PEAK supplies the name of the signal that indicates the unit was selected for PEAK LOAD. When a unit has been selected for PEAK, no RAISE or LOWER commands will be sent to the unit. The unit is disabled from PLC when this signal is TRUE. If implementing this feature is not desired (and for units that do not include PEAK support), supply a signal name that is always FALSE (such as LFALSE) for this signal.

- **START <pointname> - [POINT, CONTROL**] START supplies the name of the signal that indicates a START is in progress. The unit is disabled from PLC when this signal is found TRUE.

- **STOP <pointname> - [POINT, CONTROL]** STOP supplies the name of the signal that indicates a STOP is in progress. The unit is disabled from PLC when this signal is found TRUE.

- **READY <pointname> - [POINT, CONTROL]** READY supplies the name of an additional signal that indicates the unit is ready for PLC. The unit will be disabled from PLC when this signal is found FALSE. If no additional permissives are received to go onto PLC, supply the name of a signal that is always TRUE (such as LTRUE) for this signal.

- **MWPBR <pb_pointname> - [POINT, CONTROL]** MW PushButton Raise supplies the name of the MW RAISE pushbutton. When an MW RAISE command is desired, it is sent to this pushbutton pointname using the duration supplied by the constant MWPBDUR.

- **MWPBL <pb_pointname> - [POINT, CONTROL]** MW PushButton Lower supplies the name of the MW LOWER pushbutton. When an MW LOWER command is desired, it is sent to this pushbutton pointname using the duration supplied by the constant MWPBDUR.

SETPOINT SECTION. The following section provides definitions of constants and points used in the PLC calculations.

- **SETPOINT -** This indicates the start of the SETPOINT section. The SETPOINT section defines constants and points used in the PLC calculations that effect the entire plant. There is only one SETPOINT section. PLC attempts to match the plant MW output to a user-defined plant MW setpoint. It does this by sending RAISE and LOWER commands to units that have been enabled for PLC. Not all units need to be enabled, some units may be sitting at BASE or PEAK output, letting other units take the load swings defined by the changes in the plant MW setpoint.

- **PSPDBI <mw_value> - [CONSTANT, CONTROL]** The Plant SetPoint DeadBand Initial value indicates the initial value of the plant setpoint deadband. The plant MW is considered as matching the plant MW setpoint if the difference between the plant MW setpoint and the total plant MW is less than or equal to this value. Typical values are on the order of a couple of MW. The actual deadband can be changed while running PLC. This defines the initial value when the <I> is rebooted.

- **PSPDBMIN <mw_value> -** (See the definition for PSPDBMAX.)

- **PSPDBMAX <mw_value> - [CONSTANT, CONTROL]** The Plant SetPoint DeadBand MIN and MAX values define the acceptable range that the plant setpoint deadband can be set to while running PLC. If an attempt is made to set the plant setpoint deadband above the MAX value, it is set to the MAX value. If an attempt is made to set it below the MIN value, it is set to the MIN value.

- **SCANINTI <integer_seconds> - [CONSTANT, CONTROL]** The SCAN INTerval Initial value specifies the initial number of seconds between PLC calculations. During each scan interval, the PLC calculations are run and the RAISE and LOWER commands found necessary are queued to be sent to the unit at a one per second rate. Typical values for the scan interval are every 10 to 30 seconds. The actual scan interval can be changed while running PLC. This defines the initial value when the <I> is rebooted.

- **SCANINTMIN <integer_seconds> -** (See the definition for SCANINTMAX.)
- **SCANINTMAX <integer_seconds> - [CONSTANT, CONTROL]** The SCAN INTerval MIN and MAX values define the acceptable range that the scan interval can be set to while running PLC. If an attempt is made to set the scan interval above the MAX value, it is set to the MAX value. If an attempt is made to set it below the MIN value, it is set to the MIN value.

- **PMW <I_pointname> - [POINT, REQUIRED]** Plant MW supplies the name of the <I> point that contains the total plant MW. The MW from each unit that has its breaker closed (52G signal is TRUE) is summed to make the plant MW total. This calculation is done once per second.

- **PMVAR <I_pointname> - [POINT, REQUIRED]** Plant MVAR supplies the name of the <I> point that contains the total plant MVAR. The MVAR from each unit that has its breaker closed (52G signal is TRUE) is summed to make the plant MVAR total. This calculation is done once per second.

- **PMWSP <I_pointname> - [POINT, CONTROL]** Plant MW SetPoint supplies the name of the <I> point that holds the plant MW setpoint. This point must be an analog setpoint command in the <I> data dictionary. While PLC is off, the plant MW setpoint will track the current plant MW. This gives a bumpless transfer when PLC is enabled. While PLC is on, the value of the plant MW setpoint can be set from any display that has an analog setpoint target that writes to the plant setpoint. If PLC is off, this is updated every second. If PLC is on, it is updated when the user changes the setpoint.

- **PMWERR <I_pointname> - [POINT, CONTROL]** Plant MW ERRor supplies the name of the <I> point that holds the plant MW error. The error is defined as the difference between the plant MW setpoint and the current plant MW. This value is updated once per second.

- **PSPDB <I_pointname> - [POINT, CONTROL]** Plant SetPoint DeadBand supplies the name of the <I> point that holds the plant setpoint deadband in MW. This point must be an analog setpoint command in the <I> data dictionary. The plant is considered to match the plant setpoint if the absolute value of the plant MW error is less than or equal to this value. The plant setpoint deadband can be changed while running PLC using any display that can write to this analog setpoint.

- **SCANINT <I_pointname> - [POINT, CONTROL]** SCAN INTerval supplies the name of the <I> point that holds the scan interval (in seconds) for the PLC algorithm. This point must be an analog setpoint command in the <I> data dictionary. The scan interval defines how often the PLC algorithm will run and queue up RAISE and LOWER commands to the units. The scan interval can be changed while running PLC using any display that can write this analog setpoint.

- **PLCSTATE <I_pointname> - [POINT, CONTROL]** PLC STATE supplies the name of the <I> point that indicates whether PLC is OFF or ON. This point must be an enumerated state command in the <I> data dictionary. If PLC is OFF, no PLC actions will take place. If PLC is ON, the units enabled for PLC can be given RAISE and LOWER commands to meet the desired plant MW setpoint or load balance the units. The PLC can be toggled ON or OFF from any display that can send this enumerated state command. (State 0 is OFF, state 1 is ON.)

- **PSPMATCH <I_pointname> - [POINT, CONTROL]** Plant SetPoint MATCHed supplies the name of an <I> point that indicates whether the plant has achieved the goal of matching the plant MW setpoint or not. This must be a logic point in the <I> data dictionary. If the plant MW error is less than the plant MW deadband, this point will be given the value of TRUE. If the plant does not match the plant setpoint, it will be FALSE.

BALANCE SECTION. The following section provides definitions of constants and points used in the calculations for PLC Load Balance.

- **BALANCE -** This indicates the start of the BALANCE section. The BALANCE section defines constants and points used in the PLC LOAD BALANCE calculations on a plant wide basis. There is only one BALANCE section. Balance control can only be enabled if PLC is also enabled. Balance control will cause each unit enabled for PLC to produce the same percentage of rated power as each other unit, within a balance deadband.

- **BALDBI <mw_value> - [CONSTANT, CONTROL]** The BALance DeadBand Initial value indicates the initial value of the balance control deadband. Each unit calculates what its ideal balanced output should be, and if the unit is within this deadband of the ideal value it is considered balanced. If it is outside this deadband, RAISE or LOWER commands are sent to the unit to balance it. The balance deadband can be changed while running PLC. This defines the initial value assigned after a reboot of the <I>.

- **BALDBMIN <mw_value> -** (See the definition for BALDBMAX.)
- **BALDBMAX <mw_value> - [CONSTANT, CONTROL]** The BALance DeadBand MIN and MAX values define the acceptable range that the balance deadband can be set to while running PLC. If an attempt is made to set the value above the MAX value, it will be set to the MAX value. If an attempt is made to set the value below the MIN value, it will be set to the MIN value.

- **BALSTATE <I_pointname> - [POINT, CONTROL]** BALance STATE supplies the name of an <I> point that indicates whether balance control is ON or OFF. This must be an enumerated state command in the <I> data dictionary. If balance control is OFF, no RAISE or LOWER commands will be sent to balance the output from each unit. If balance control is ON, RAISE and LOWER commands will be sent to units selected for PLC to even out the percentage of rated power each unit is producing while still meeting the total plant MW setpoint. PLC must be ON to turn on balance control. The balance control can be toggled ON or OFF from any display that can send this enumerated state command. (State 0 is OFF, state 1 is ON.)

- **BALMATCH <I_pointname> - [POINT, CONTROL]** BALance control MATCHed supplies the name of an <I> point that indicates whether the plant has achieved the goal of balancing the units' MW output or not. This must be a logic point in the <I> data dictionary. If all units selected for PLC are within the balance deadband of their ideal balanced output, this point will be set to TRUE. If balance control is OFF, or a unit is not within the balance deadband of ideal balanced output, this value will be FALSE.

- **BALDB <I_pointname> - [POINT, CONTROL]** BALance DeadBand supplies the name of the <I> point that holds the balance deadband in MW. This point must be an analog setpoint command in the <I> data dictionary. A unit is considered to be balanced if its MW output is within this deadband of the ideal balance MW. The ideal balanced deadband is defined as producing the same percentage of rated capacity from each unit selected for PLC. The balance deadband can be changed while running PLC using any display that can write this analog setpoint.

7-14.3. Example PLC Configuration File

The following is a printout of a sample PLC Configuration file for a three turbine facility.

```
; PLC.DAT PLC Configuration File
; Mark V GE Drive Systems - Sample File
;
; This file configures the PLC program
```

```
;    - Lines beginning with a semi-colon are comment lines
;    - Blank lines are ignored
;    - Indentation is required for all non-section lines
;    - One or more spaces or tabs can be used to indent items
;    - Items on a line can be separated by one or more spaces or tabs
;    - Any mixture of upper and lower case can be used


NUNITS 3                        ; Number of turbines in this plant

UNIT T1                         ; Name of 1st Gas Turbine
    CONTROL     1               ; Controllable unit, 0 = No, 1 = Yes
    CAP         100             ; Peak Capacity, MW
    MIN         10              ; Minimum Output, MW
    MWPBDUR     8               ; MW pushbutton, duration in scans
    MWPBRATE    0.12789         ; MW pushbutton, MW per push

    MW          T1:DWATT_BOI    ; Generator MW Transducer, MW
    MVAR        T1:DVAR_BOI     ; Generator MVAR Transducer, MVAR
    52G         T1:L52GX        ; Generator Breaker
    STATE       I1:PLCSTATE_1   ; PLC state
    TEMP        T1:L26TCX       ; Is on temperature control
    COMMHTHY    T1:LTRUE        ; For communication health check
    BASE        T1:L83B         ; Base capacity is selected
    PEAK        T1:L83P         ; Peak capacity is selected
    STOP        T1:L94X         ; Shutdown in progress
    START       T1:L1X          ; Startup in progress
    READY       T1:LTRUE        ; Is ready for PLC
    MWPBR       T1:L70R4R_CPB   ; MW pushbutton - RAISE command
    MWPBL       T1:L70R4L_CPB   ; MW pushbutton - LOWER command


UNIT T2                         ; Name of 2nd Gas Turbine
    CONTROL     1               ; Controllable unit, 0 = No, 1 = Yes
    CAP         100             ; Peak Capacity, MW
    MIN         10              ; Minimum Output, MW
    MWPBDUR     8               ; MW pushbutton, duration in scans
    MWPBRATE    0.12789         ; MW pushbutton, MW per push

    MW          T2:DWATT_BOI    ; Generator MW Transducer, MW
    MVAR        T2:DVAR_BOI     ; Generator MVAR Transducer, MVAR
    52G         T2:L52GX        ; Generator Breaker
    STATE       I1:PLCSTATE_2   ; PLC state
    TEMP        T2:L26TCX       ; Is on temperature control
    COMMHTHY    T2:LTRUE        ; For communication health check
    BASE        T2:L83B         ; Base capacity is selected
    PEAK        T2:L83P         ; Peak capacity is selected
    STOP        T2:L94X         ; Shutdown in progress
    START       T2:L1X          ; Startup in progress
    READY       T2:LTRUE        ; Is ready for PLC
    MWPBR       T2:L70R4R_CPB   ; MW pushbutton - RAISE command
    MWPBL       T2:L70R4L_CPB   ; MW pushbutton - LOWER command

UNIT T3                         ; Name of Steam Turbine
    CONTROL     1               ; Controllable unit, 0 = No, 1 = Yes
    CAP         100             ; Peak Capacity, MW
    MIN         10              ; Minimum Output, MW
    MWPBDUR     8               ; MW pushbutton, duration in scans
    MWPBRATE    0.12789         ; MW pushbutton, MW per push

    MW          T3:DWATT_BOI    ; Generator MW Transducer, MW
    MVAR        T3:DVAR_BOI     ; Generator MVAR Transducer, MVAR
    52G         T3:L52GX        ; Generator Breaker
    STATE       I1:PLCSTATE_3   ; PLC state
```

```
    TEMP         T3:L26TCX        ; Is on temperature control
    COMMHTHY     T3:LTRUE         ; For communication health check
    BASE         T3:L83B          ; Base capacity is selected
    PEAK         T3:L83P          ; Peak capacity is selected
    STOP         T3:L94X          ; Shutdown in progress
    START        T3:L1X           ; Startup in progress
    READY        T3:LTRUE         ; Is ready for PLC
    MWPBR        T3:L70R4R_CPB    ; MW pushbutton - RAISE command
    MWPBL        T3:L70R4L_CPB    ; MW pushbutton - LOWER command


SETPOINT                          ; PLANT SETPOINT items.

    PSPDBI       1.0              ; Initial Plant Setpoint deadband, MW
    PSPDBMIN     1.0              ; Minimum Plant Setpoint deadband, MW
    PSPDBMAX     10.0             ; Maximum Plant Setpoint deadband, MW
    SCANINTI     10               ; Initial PLC scan interval, sec
    SCANMIN      10               ; Minimum PLC scan interval, sec
    SCANMAX      60               ; Maximum PLC scan interval, sec

    PMW          I1:PMW           ; Plant total MW point name
    PMVAR        I1:PMVAR         ; Plant total MVAR point name
    PMWSP        I1:PMWSP         ; Plant MW Setpoint
    PMWERR       I1:PMWERR        ; Plant MW error
    PSPDB        I1:PSPDB         ; Plant Setpoint Deadband
    SCANINT      I1:SCANINT       ; Scan interval for PLC calculation, sec.
    PLCSTATE     I1:PLCSTATE      ; PLC State:
                                  ;     0 = PLC OFF
                                  ;     1 = PLC ON
    PSPMATCH     I1:PSPMATCH      ; Plant MW error is within deadband


BALANCE

    BALDBI       1.0              ; Initial value of Balance deadband, MW
    BALDBMIN     1.0              ; Minimum value of Balance deadband, MW
    BALDBMAX     10.0             ; Maximum value of Balance deadband, MW

    BALSTATE     I1:BALSTATE      ; Balance ON/OFF state:
                                  ;     0 = Balance Control OFF
                                  ;     1 = Balance Control ON
    BALMATCH     I1:BALMATCH      ; Balance error is within deadband
    BALDB        I1:BALDB         ; Balance deadband point name
```

CHAPTER 8

I/O APPLICATIONS

8-1. INTRODUCTION

The process of determining where to place or "land" a new contact input, naming the associated software signal, and verifying or modifying the I/O configuration is covered in this section. For information on Control Sequence Program modifications required as a result of an addition or subtraction of a contact input, refer to Chapter 5.

CAUTION

Exercise care when using an ASCII text editor with any Mark V system file. Do not perform any modifications other than those specified in this document. Incorrect modifications to these files could be detrimental to system performance. It is highly suggested that system backups are performed before any modifications are done to these files. These changes should only be done by qualified personnel.

8-2. ADDING A CONTACT INPUT

At times it is desirable or necessary to monitor an additional contact input (or digital input) for control, protection, logging, monitoring or alarm purposes. This can be done by the Mark V Control Panel as it typically contains multiple spare contact input locations for both <C> and <Q> (through <CD> and <QDn>, respectively). The addition of a contact input to the control panel requires modifications to the Control Sequence Program of <Q> or <C> and may require changes to the I/O Configuration of the control panel.

The steps involved in renaming a spare software signal in the primary operator interface are outlined in the flow-chart in Figure 8-1. Many of the steps are performed from the DOS command line, not from the IDOS menu/display screens and require the use of an ASCII text editor. The steps are explained in detail in sections 8-2.1. through 8-2.8. as follows.

Step 1	Determine if the new contact input is to be added to <CD> or <QDn>.
Step 2	Choose a spare contact input point from the proper digital I/O core from the I/O assignment file (IO.ASG) and rename it.
Step 3	Create a new `UNITDATA.DAT` file.
Step 4	Check and modify, if necessary, the point's I/O Configuration.
Step 5	Download the new I/O configuration to Control Panel using EEPROM Downloader.
Step 6	Reset the control panel processor(s) one at a time to load new I/O configuration information into processor RAM.
Step 7	Reset the <I> computer to load the new `UNITDATA.DAT` information into <I>'s RAM.
Step 8	Transfer the new `IO.ASG, UNITDATA.DAT`, and `IOCFG_?.DAT` to any other <I>'s controlling the unit to which the contact input was added and reset them.

In this section, a new contact input will be added for the limit switch of a critical regulating valve. That is, the regulating valve (VR-1) will have its limit switch contacts calibrated and adjusted such that the normally closed contacts will be open whenever the regulating valve is closed. The device number for a limit switch is **33**. In the example, the contacts being wired into the Control Panel will originate from the device 33VR-1. For control purposes, the signal pointname which is **driven** by the new contact input should be a logic " 1 " when the valve is closed, and a logic " 0 " whenever the valve is open.

8-2.1. Step 1 - Determine Where to Add New Contact Input

When a contact input is to be added to the control panel, it is necessary to determine where the new input will be landed on the digital I/O core. This decision is made the basis of whether the new input will be part of the **critical** (for example: unit trip and/or protection) or non-critical sequencing (for example: alarm annunciation and/or monitoring purposes) of the unit. Generally, field modifications which might include a contact input addition are made to enhance unit control and protection and are therefore critical in nature. This in turn requires that the contact input be wired to <QDn>. If the unit/process can survive the **failure** or loss of the signal from the new contact input, it may be added to <CD>. If the unit\process cannot survive the failure or loss of the new contact input, the input should be wired into <QDn>. In the example, the regulating valve is critical to the operation of the unit and the process; accordingly, the contacts will be wired into <QD1>.

8-2.2. Step 2 - Choose a Spare Contact Input Point

After it has been determined that a new contact input is to be added to a specific digital I/O core, the user should choose a name for the new point. Point names should be as descriptive as possible (up to a maximum of 12 characters) and follow the signal/device numbering scheme used elsewhere in the control panel. In the example, a point is required for device 33VR-1, which in turn should be a logic " 1 " when the valve is closed; an appropriate pointname is `L33VR1C.`

Point names can only be assigned to one point number in the Mark V turbine control, therefore, the `UNITDATA.DAT` file must be searched to determine if the pointname to be selected is already in use. This can be done with an ASCII text editor. From the Main Menu, exit to DOS by pressing ALT + X. Change to the appropriate unit directory (if the modification is to be made to Unit 1, change to `F:\UNIT1).` Open the `UNITDATA.DAT` file and perform a search of the file for the pointname to be used. If the desired pointname is used, a new pointname must be chosen. This will entail a new search that will hopefully produce a pointname that is not being used.

To find a spare location for a supplemental contact input (both physically and in software), open the I/O assignment file `IO.ASG,` and using an ASCII text editor, scroll through the file until an appropriate point can be found (such as a point that has the correct hardware and signal name in the digital I/O core).

CAUTION

Do not edit or change `UNITDATA.DAT` file. `UNITDATA.DAT` is automatically created by programs on the <I> and modifications to it could result in unpredictable operation.

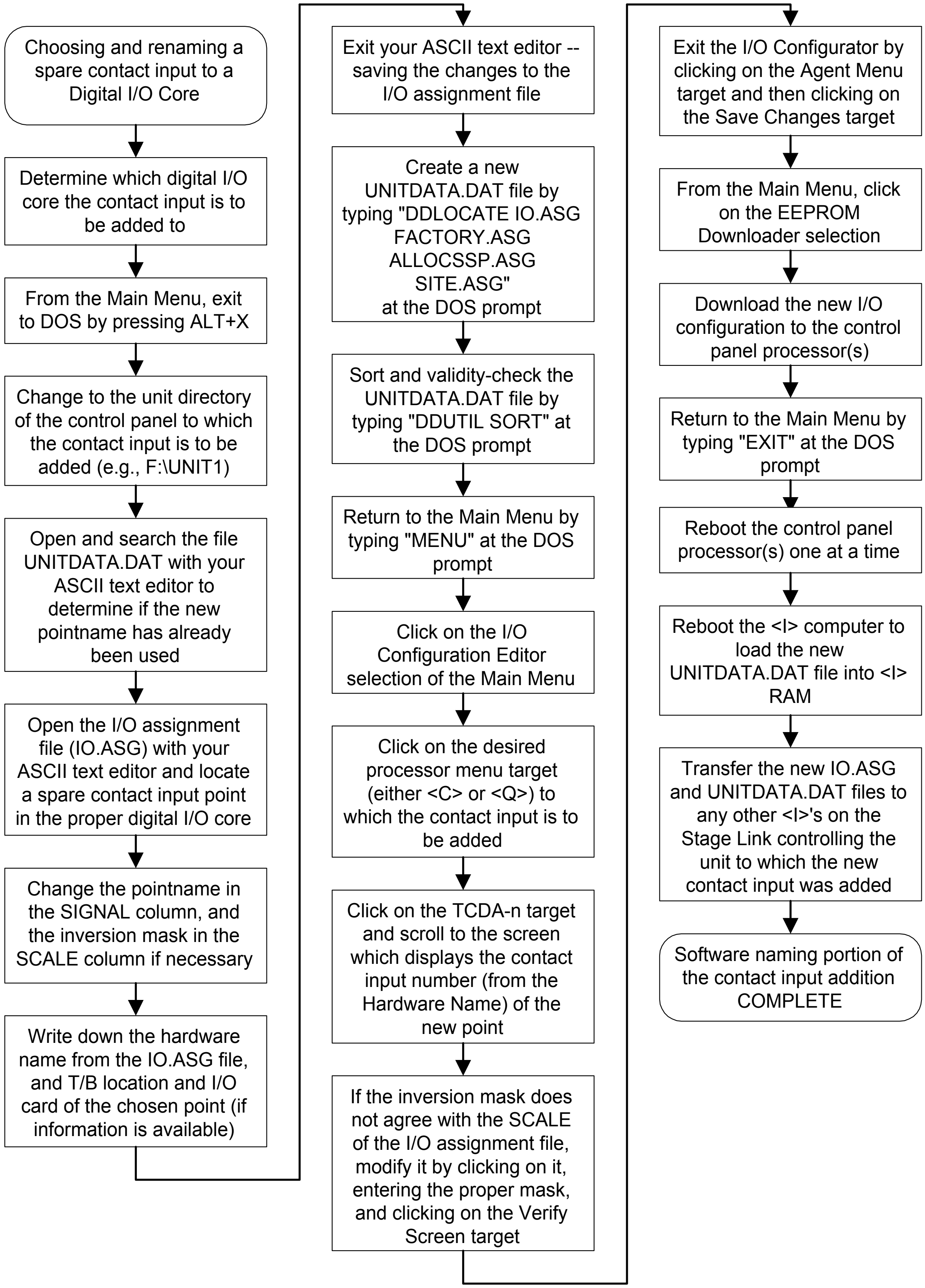


Figure 8-1. Adding a Contact Input to a Digital I/O Core

A section of the I/O assignment file showing several contact inputs is shown as follows.

```
;HDW NAME    SIGNAL       SCALE        ;COMMENTS

Q_QD1_CI04   L63HG1L      CIM_I ; Q1 TCDA/DTBA-  7
Q_QD1_CI05   Q_QD1_CI05   CIM   ; Q1 TCDA/DTBA-  9
Q_QD1_CI06   Q_QD1_CI06   CIM   ; Q1 TCDA/DTBA- 11
```

`HDW NAME` defines:

- processor to which the point is coupled (Q)
- digital I/O core to which the point will be terminated in the control panel (QD1)
- type and number of the input signal (CI for contact input, 05 for fifth contact input)

`SIGNAL` defines:

- software signal pointname of the contact input

`SCALE` is a note about the inversion mask of the contact input:

- CIM_I denotes an inverted input
- CIM denotes a non-inverted input

`COMMENTS` defines the terminal board location and the I/O card to which the point is linked:

- TCDA is the I/O card to which the point is linked to and can be located under in the I/O Configurator
- DTBA - 9 is I/O termination board DTBA, terminal #9 for the signal input leg

Once an appropriate spare contact input point has been identified, change the name in the `SIGNAL` column to the desired software signal pointname. In the example, a contact input from device 33VR-1 is added; the corresponding software signal pointname is L33VR1C.

If necessary, the intended inversion mask of the pointname may be changed using an ASCII text editor. Specifically, this involves editing the `SCALE` column of the selected point. **The actual inversion masking of a contact input is determined by the I/O Configurator**; the information in the `SCALE` column of IO.ASG for contact input points is descriptive and does not actually affect/reflect the actual inversion mask of the point.

The following example shows how the `HDW NAME` location `Q_QD1_CIO5` can be modified so that the valve will close when a logic " 1 " is received. As the inverse of the original configuration, the conversion mask must be changed to `CIM_I.`

```
;HDW NAME    SIGNAL       SCALE        ;  COMMENTS
Q_QD1_CI04   L63HG1L      CIM_I ; Q1 TCDA/DTBA-  7
Q_QD1_CI05   L33VR1C      CIM_I ; Q1 TCDA/DTBA-  9
Q_QD1_CI06   Q_QD1_CI06   CIM   ; Q1 TCDA/DTBA- 11
```

Exit the ASCII text editor making sure to save the modified I/O assignment file with the same name (i.e., `IO.ASG).` Return to the IDOS Main Menu by typing MENU at the DOS prompt.

8-2.3. Step 3 - Create a New `UNITDATA.DAT` File

The unit Data Dictionary must now be updated to include the new software signal pointname of the contact input point being added. The file `UNITDATA.DAT` must contain all of the signal pointnames and therefore must be updated (a **new** `UNITDATA.DAT` file will actually be created with the new software signal pointname from `IO.ASG` included in the file). `UNITDATA.DAT` is created by executing `DDLOCATE` using the four assignment files (one of which is `IO.ASG)` as input to the program. All of the assignment files must be used.

To run the `DDLOCATE` program, enter the following command at the DOS prompt of the unit directory.

```
DDLOCATE  IO.ASG  FACTORY.ASG  ALLOCSSP.ASG  SITE.ASG
```

When the Data Dictionary loader program loads the information from `UNITDATA.DAT` into <I> RAM, the signal pointnames should be in alphanumeric ordered by pointname for fast loading. However, the `UNITDATA.DAT` file created by `DDLOCATE` is not sorted in this fashion. To accomplish an alphanumeric sorting, run the `DDUTIL` program with the `SORT` option specified by typing in the following command at the DOS prompt of the unit directory:

```
DDUTIL SORT
```

In addition to sorting the new `UNITDATA.DAT` file in alphanumeric order by signal pointname, `DDUTIL` also performs a validity check of the signal pointnames in the file. (Validity checking is the process of checking that no signal pointname is assigned to more than one point in `UNITDATA.DAT.)` The results of sorting and validity-checking will displayed on the screen. The following message should be displayed:

```
SORTING COMPLETE: UNITDATA.DAT is new file, UNITDATA.BAK is old file
```

This is an indication that no problems were encountered during sorting and validity-checking.

Run with the `SORT` option specified, `DDUTIL` will store the unsorted and un-validity-checked `UNITDATA.DAT` file produced by `DDLOCATE` as a back-up file with the name `UNITDATA.BAK.` It will also store the results of sorting the file as the new `UNITDATA.DAT` file.

Return to the IDOS Main Menu by typing MENU at the DOS prompt and pressing Enter.

NOTE

Several of these steps can be done automatically by using the `MK5MAKE.BAT` file. See Maintenance Manual GEH-5980, Chapter 4.

8-2.4. Step 4 - I/O Configuration Check/Modification

The status of the inversion mask of the new contact input must be compared with that of the I/O Configuration file using the I/O Configurator. This is done regardless of whether or not it was modified in the last step. From the Main Menu, point and click on the I/O Configuration Editor selection. After a few seconds, the I/O Configurator display will appear on the CRT. Check the top of the I/O Configurator display for an indication of the selected processor (<C> or <Q>). If necessary, click on the proper target at the bottom of the screen to switch to the appropriate processor (in the example, a contact input was added to <QD1>, therefore, the <Q> Processor Menu must be selected). Click on the appropriate I/O card target (TCDA-1 in the example) to display the configuration information for contact input number 5.

The first screen which appears will be card revision information. Click on the NEXT SCREEN target to advance to the contact input information. (See TCDA I/O configurator screens in Appendix B.) Look for the contact input number (in the example, contact input number 5). Read across to the column under `Inversion Masks` and find the value which corresponds to the contact input number. If the value is a " 0 ", the input is not inverted; if it is a " 1 ", it is inverted.

In the example, a point was inverted in the I/O assignment file. This means that the I/O Configurator display must be changed to agree with the I/O assignment file. To perform the required changes, click on the desired point (the background color of the point will change to white indicating it is selected), type in the new value (in the example, " 1 ") and then point and click on the VERIFY SCREEN target at the bottom of the display. The contact inversion will now reflect the change.

When the point information in the I/O assignment file and the I/O Configurator agree, exit the I/O configurator by clicking on the AGENT MENU target and then clicking on the SAVE CHANGES target. The I/O Configurator program will be terminated and the I/O configuration files (`IOCFG_Q.DAT` and/or `IOCFG_C.DAT)` will be generated during the termination. The screen will return to the IDOS Main Menu upon termination of the I/O Configurator.

8-2.5. Step 5 - Download New I/O Configuration

From the Main Menu, initiate the EEPROM downloading function by pointing and clicking on EEPROM Downloader. This will return the screen to a command line with the following prompt: `"EEPROM Downloader>"`.

In the example, the unit name/designation for the control panel being modified is T1. Download the `IOCFG_Q.DAT` file to each processor of <Q> (<R>, and <S> and <T> in a TMR control panel) by entering `"DOWN T1 R IOCFG"`.
Repeat the command for <S> and <T> in a TMR control panel. When the I/O configuration has been downloaded to each of the appropriate processor(s), type EXIT at the prompt to return to the Main Menu.

8-2.6. Step 6 - Reset Processor(s).

For the contact input addition's inversion mask change to become active, reset (or cycle) each processor **one at a time.** Wait until the processor returns to normal operating condition (state A7) before resetting the next processor. Resetting the processor transfers the modified I/O configuration file from processor EEPROM to be used in the processor RAM.

8-2.7. Step 7 - Reset <I> Computer

The new `UNITDATA.DAT` file must now be loaded into the <I> computer's RAM so that the new software signal pointname can be displayed with units and a value when inserted into a user Defined Display. The <I> computer must be reset (press CTRL + ALT + DEL) in order for the new `UNITDATA.DAT` file to be loaded into <I> RAM. When the re-boot process is complete, the screen will return to the IDOS Main Display.

8-2.8. Step 8 - Transfer New Data Dictionary Information

If there are any other <I>'s on the Stage Link for the unit to which a new contact input was added, the new data dictionary information must be copied to floppy disk(s) and transferred to them from the <I> where the modification was made. After the new Data Dictionary is loaded, the <I> must be reset in order for it to display the new point name when entered.

8-3. ADDING A CONTACT OUTPUT (RELAY OR SOLENOID)

The Mark V panel has numerous contact outputs. Many of the contact outputs can be either a relay output or a solenoid output, depending on the placement of two jumpers. As referred to in this section, a relay output is a **C-form** contact output (electrically common normally open and normally closed contacts of a relay available at a terminal board). See Figure 8-2. These contacts are sometimes referred to as **dry contacts.** This means that they are to be used to switch an externally supplied power source for some load; for example an indicating lamp, motor starter, status indication, or similar.

A solenoid output is a also a C-form contact output. However, when application-specific Isolation jumpers **PX** and **MX** are in place, these contacts are powered by an internal power source (usually, either 125 V DC or 120 V AC). Solenoid outputs are used to drive solenoid-operated valves used in turbine and/or auxiliary control.

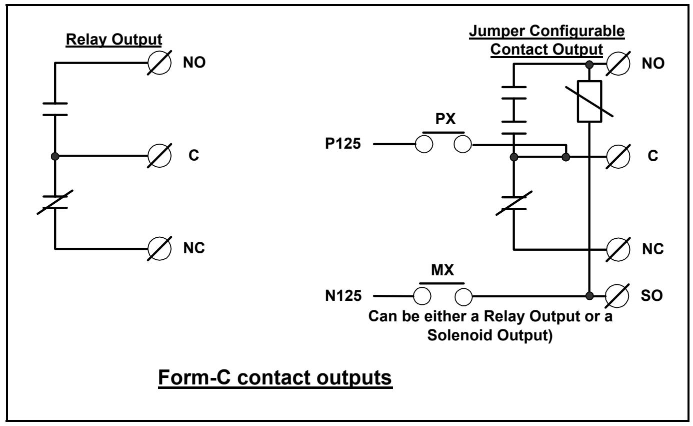


Figure 8-2. Mark V Relay/Solenoid Outputs

In order to operate a relay output or solenoid output from the Mark V control panel, modifications to the Control Sequence Program are necessary to drive the relay's pointname from some logic in the Program (see Chapter 5). The pointname which drives the relay must be selected from the available spare contact output logic points in the I/O assignment file (IO.ASG) and renamed. The process of selecting the spare contact output logic point, renaming the point, determining the associated terminal board locations, and checking the Isolation jumpers are covered in this section.

The steps involved in finding and renaming a spare contact output logic point are outlined in the flow chart in Figure 8-3. The steps are explained in detail in sections 8-3.1 through 8-3.6. as follows.

Step 1 Determine to which digital I/O core the new relay output or solenoid output will be added.
Step 2 Choose a spare contact output logic point from the I/O assignment file and rename it.
Step 3 Create a new `UNITDATA.DAT` file.
Step 4 Reset the <I> computer to load the new `UNITDATA.DAT` information into <I>'s RAM.
Step 5 Check to see that the I/O output terminal board Jumpers are set properly.
Step 6 Transfer the new Data Dictionary information to any other <I>'s on the stage link which will have their Control Sequence Programs modified to include the new local logic pointname.

As an example, a new relay output will be added to Unit 1. The relay output will be used to drive an indicating lamp in a remote location which is non-critical to the operation of the unit/process.

8-3.1. Step 1 - Determine Where To a Add New Contact Output

After it has been decided that a relay output or solenoid output is to be added to the control panel, it is necessary to determine the digital I/O core to which it will be added. The factor on which this decision is made is whether the new output will be part of the **critical** (unit trip and/or protection) or non-critical control functions of the unit. If the unit/process can survive the **failure** or loss of the signal from the new output (such as when <C> might be powered-down or inoperable), then it should be added to <CD>. If the unit\process cannot survive the failure or loss of the new contact output the new contact output should be wired into <QDn>. In the example, the relay output is not critical to the operation of the unit or process, therefore, it will be added to <CD>.

After the digital I/O core to which the new output is to be added has been determined, a name for the new point should be chosen. Pointnames should be as descriptive as possible (up to a maximum of 12 characters) and follow the signal/device numbering scheme used elsewhere in the control panel. In our example, the new relay output will drive a remote indicating lamp. An appropriate pointname would be `L30REM_IND`.

8-3.2. Step 2 - Choose a Spare Contact Output Point

Point names can only be assigned to one point number in the Mark V turbine control, therefore, the `UNITDATA.DAT` file must be searched to determine if the pointname to be selected is already in use. This can be done with an ASCII text editor. From the Main Menu, exit to DOS by pressing ALT + X. Change to the appropriate unit directory (if the modification is to be made to Unit 1, change to `F:\UNIT1`). Open the `UNITDATA.DAT` file and perform a search of the file for the pointname to be used. If the desired pointname is used, a new pointname must be chosen. This will entail a new search that will hopefully produce a pointname that is not being used.

To find a spare contact output point, open the I/O assignment file—IO.ASG—file using the ASCII text editor and scroll through the file looking for the contact output section of the core to which the point is to be added. Spare contact output points will have the same hardware name and signal name. A section of the I/O assignment file showing several contact outputs is as follows:

```
;HDW NAME    SIGNAL       SCALE#       ;COMMENTS
C_CD_CO01    C_CD_CO01    LOG   ; C1 DTBC-  3  SOLENOID OUT
C_CD_CO02    C_CD_CO02    LOG   ; C1 DTBC-  7  SOLENOID OUT
C_CD_CO03    C_CD_CO03    LOG   ; C1 DTBC- 11  SOLENOID OUT
C_CD_CO04    C_CD_CO04    LOG   ; C1 DTBC- 15  SOLENOID OUT
```

`HDW NAME` defines:

- processor to which the point is coupled (C)
- digital I/O core to which the point will be terminated in the control panel (CD)
- type and number of the output signal (CO for contact output, 01 for the first contact output)

`SIGNAL` defines:

- software signal pointname of the contact output

`SCALE` defines the type of contact output:

- LOG is a logic output

`COMMENTS` defines the terminal board location of the contact output point:

- `DTBC- 3` is <CD> I/O termination board DTBC, terminal #3 for the Common output leg of an output which is normally reserved for a `SOLENOID` but can be either a relay output or solenoid output depending on the placement of the Isolation jumpers

Once a spare contact output point has been identified, change the name in the SIGNAL column to the desired software signal pointname. In the example, HDW NAME location C_CD_CO01 was chosen. This contact output can be either a solenoid output or a relay output, depending on the placement of Isolation jumpers on the <CD> DTBC card. Since only a relay output point is required, the final check should ensure that the jumpers are not set to configure this output as a solenoid output. Refer to the following example for the necessary changes to implement the addition in the previous I/O assignment file example.

```
;HDW NAME    SIGNAL       SCALE#        ;COMMENTS
C_CD_CO01    L30REM_IND   LOG   ; C1 DTBC-  3  CONTACT OUT
C_CD_CO02    C_CD_CO02    LOG   ; C1 DTBC-  7  SOLENOID OUT
C_CD_CO03    C_CD_CO03    LOG   ; C1 DTBC- 11  SOLENOID OUT
C_CD_CO04    C_CD_CO04    LOG   ; C1 DTBC- 15  SOLENOID OUT
```

Exit the ASCII text editor making sure to save the modified I/O assignment file with the same name. Remain in DOS for the next step.

8-3.3. Step 3 - Create New UNITDATA.DAT File

The unit Data Dictionary must now be updated to include the new software signal pointname of the contact output point being added. The file UNITDATA.DAT (one of the unit Data Dictionary files) must contain all of the signal pointnames and therefore must be updated (a **new** UNITDATA.DAT file will actually be created with the new software signal pointname from IO.ASG included in the file). UNITDATA.DAT is created by executing the DDLOCATE program and using the four assignment files (one of which is IO.ASG) as input to the program. All four of the assignment files must be used even though only IO.ASG was modified in this example. A new UNITDATA.DAT is actually being created by the DDLOCATE program and all the assignments must be included in the new file.

To run the DDLOCATE program, enter the following command at the DOS prompt of the unit directory:
```
DDLOCATE IO.ASG FACTORY.ASG ALLOCSSP.ASG SITE.ASG
```

When the Data Dictionary loader program loads the information from UNITDATA.DAT into <I> RAM, the signal pointnames should be in alphanumeric order by pointname for fast loading. However, the UNITDATA.DAT file created by DDLOCATE is not sorted in this fashion.To accomplish an alphanumeric sorting, run the DDUTIL program with the SORT option specified by typing in the following command at the DOS prompt of the unit directory:
```
DDUTIL SORT
```

In addition to sorting the new UNITDATA.DAT file in alphanumeric order by signal pointname, DDUTIL also performs a validity check of the signal pointnames in the file. Validity checking is the process of checking that no signal pointname is assigned to more than one point in UNITDATA.DAT. The results of sorting and validity-checking will displayed on the screen. The following message should be presented indicating that no problems were encountered during sorting and validity-checking:
```
SORTING COMPLETE: UNITDATA.DAT is new file, UNITDATA.BAK is old file
```

DDUTIL run with the SORT option specified will store the unsorted and un-validity-checked UNITDATA.DAT file produced by DDLOCATE as back-up file with the name UNITDATA.BAK and stores the results of sorting the file as the new UNITDATA.DAT file. Return to the IDOS Main Menu by typing MENU at the DOS prompt and pressing Enter.

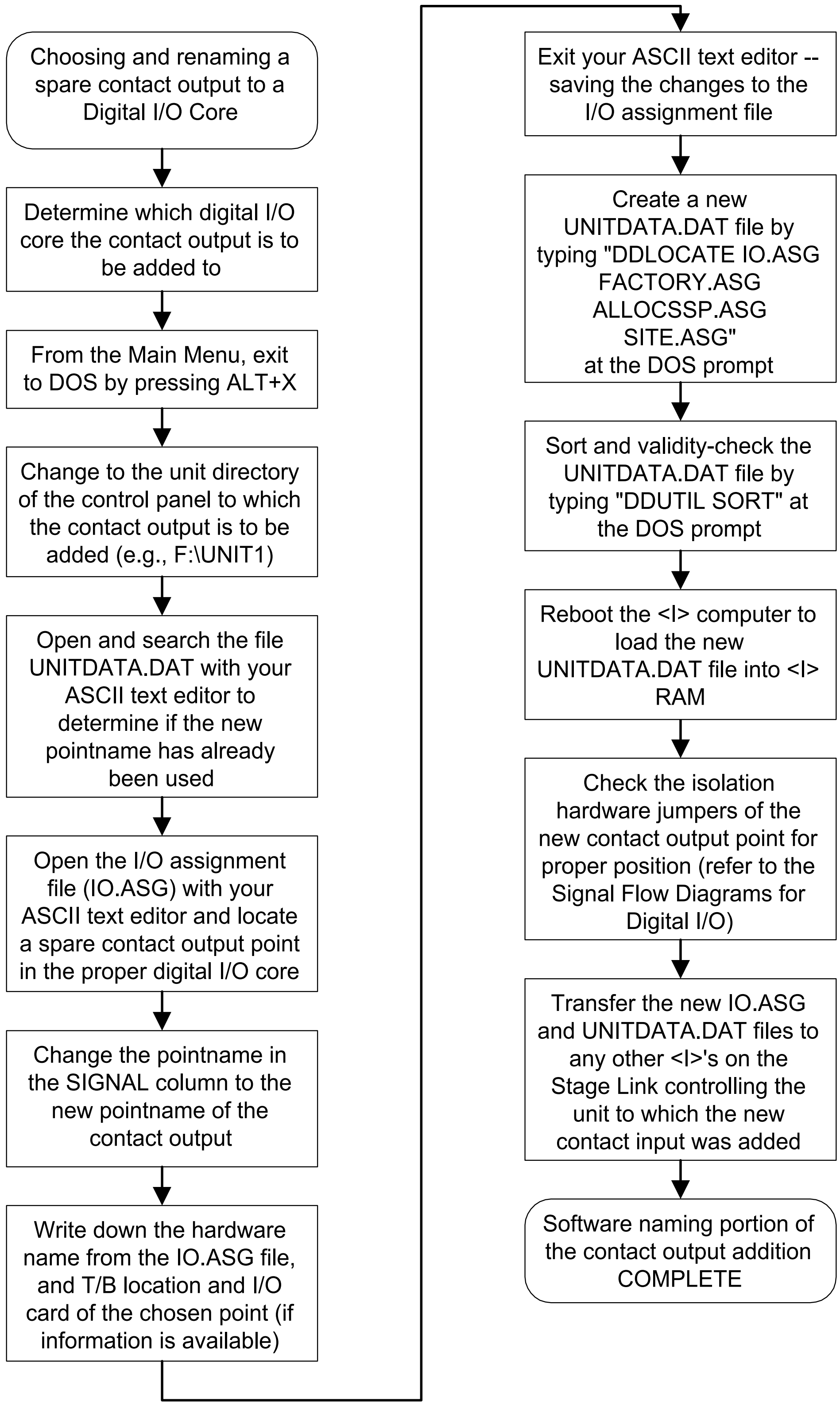


Figure 8-3. Choosing/Renaming Spare Contact Output (Relay or Solenoid)

8-3.4. Step 4 - Reset <I> Computer

The new `UNITDATA.DAT` file must now be loaded into the <I> computer's RAM so that the new software signal pointname can be displayed with units and a value when inserted into a User Defined Display. To load the new file into <I> RAM, the primary operator interface computer must be reset (press CTRL + ALT + DEL) for the new `UNITDATA.DAT` file to be loaded into <I> RAM. When the reset process is complete, the IDOS Main Display will return.

NOTE

Several of these steps can be done automatically by using the `MK5MAKE.BAT` file. See Maintenance Manual GEH-5980, Chapter 4.

Contact Output	DTBC TERM SCREW				Isolation Jumpers	
#	SOL	NO	C	NC	PX	MX
1	1	2	3	4	P1	M1
2	5	6	7	8	P2	M2
3	9	10	11	12	P3	M3
4	13	14	15	16	P4	M4
5	17	18	19	20	P5	M5
6	21	22	23	24	P6	M6
7	25	26	27	28	P7	M7
8	29	30	31	32	P8	M8
9	33	34	35	36	P9	M9
10	37	38	39	40	P10	M10
11	41	40	43	44	P11	M11
12	45	46	47	48	P12	M12
13	49	50	51	52	P13	M13
14	53	54	55	56	P14	M14
15	57	58	59	60	P15	M15
16*	61	62	63	64	P16	M16
17*	65	66	67	68	P17	M17
18*	69	70	71	72	P18	M18
19	--	73	74	75	--	--
20	--	76	77	78	--	--
21	--	79	80	81	--	--
22	--	82	83	84	--	--
23	--	85	86	87	--	--
24	--	88	89	90	--	--
25	--	91	92	93	--	--
26	--	94	95	96	--	--
27	--	97	98	99	--	--
28	--	100	101	102	--	--
29	--	103	104	105	--	--
30	--	106	107	108	--	--

Figure 8-4. Portion of a Digital Output Sheet

* Reserved Solenoid Circuits (see Application Specific Documentation and Appendix E).

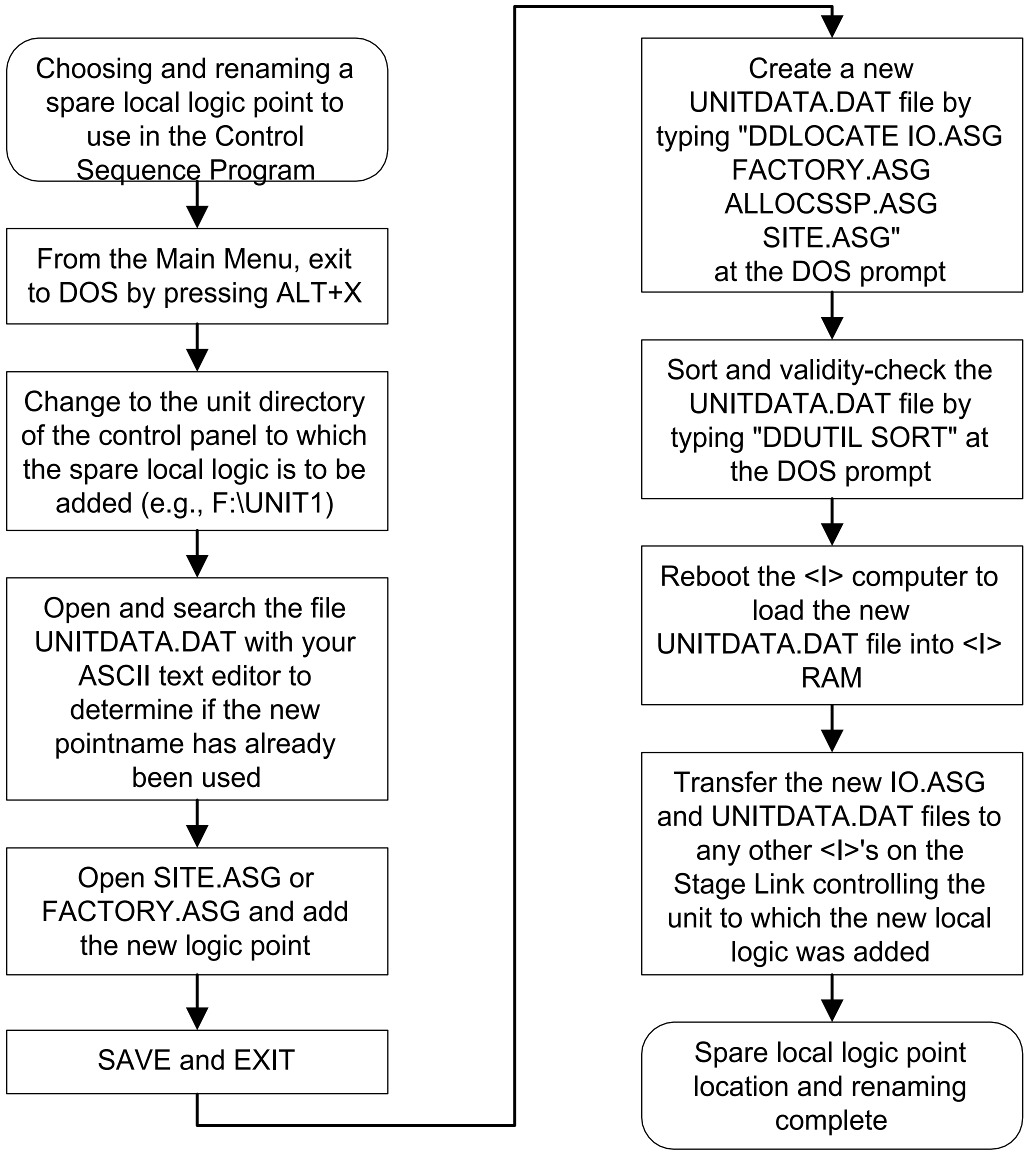


Figure 8-5. Choosing/Renaming Spare Local Logic Point

8-3.5. Step 5 - Check I/O Terminal Board Jumpers

Mark V Control Panel contact outputs can be either a relay output or a solenoid output, depending on the placement of jumpers **PX** and **MX**. Using the Digital I/O Signal Flow Diagrams section of this manual, it can be determined whether the jumpers used to configure contact outputs as solenoid outputs. Figure 8-4 shows part of a table of a Digital Output sheet including information on contact outputs, terminal board locations, and the associated isolation jumpers.

In the example, Contact Output # 1 is used. This output could be configured as either a relay output or a solenoid output depending on the placement of Isolation jumpers **P1** and **M1**. While terminating the wires for the new contact output on terminal board DTBC, check to ensure that the Jumpers P1 and M1 are **not** installed. Remove them if they are.

8-3.6. Step 6 - Transfer New Files to Other <I>s

If there are other <I>s on the Stage Link for the unit to which a new contact output was added, the new `IO.ASG` and `UNITDATA.DAT` must be copied to floppy disk and transferred to them from the <I> where the modification was made. After the new files are loaded, each <I> must be reset in order for it to display the new point name when entered.

8-4. CHOOSING/RENAMING SPARE LOCAL LOGIC POINT

When modifications to the Control Sequence Program are necessary, the use of a spare local logic point may be required. A spare local logic point is a (logic) point in memory which is not directly associated with a contact input, a relay output, a solenoid output, or a command or push-button. Local logic points can reside in <C> or <Q> and may be fast-voted, slow-voted, or non-voted. As shipped, the Mark V Control Panel contains numerous spare local logic points. Spare local logic points are found in the `VARDEF.DAT` file in the proper unit directory.

The steps involved in renaming a spare local logic point for use in the Control Sequence Program are outlined in the flow chart in Figure 8-5. The steps are explained in detail in sections 8-4.1 through 8-4.5. The majority of steps are performed from the DOS command line, except where noted, and require the use of an ASCII text editor. In this section, a spare local logic point is renamed for use in a Control Sequence Program modification.

8-4.1. Step 1 - Choose Spare Local Logic Point

Modifications to the Control Sequence Program of either <C> or <Q> can be completed through the implementation of spare local logic points. Unless otherwise directed by GEDS factory personnel, all CSP modifications should be made to <Q>; therefore, spare local logic points should be chosen from <Q>'s Control Data Base. After it has been determined that a spare local logic point is required for a Control Sequence Program modification, a name for the new point should be chosen. Pointnames should be as descriptive as possible (up to a maximum of 12 characters) and follow the signal/device numbering scheme used elsewhere in the control panel. In the example, a spare local logic point will be required to be used in conjunction with a BBL primitive `COPY` block. An appropriate pointname is `LCOPY01`.

8-4.2. Step 2 - Modify Appropriate Assignment File (`FACTORY.ASG` or `SITE.ASG`)

Point names can only be assigned to one point number in the Mark V turbine control, therefore, the `UNITDATA.DAT` file must be searched to determine if the pointname to be selected is already in use. This can be done with an ASCII text editor. From the Main Menu, exit to DOS by pressing ALT + X. Change to the appropriate unit directory (if the modification is to be made to Unit 1, change to `F:\UNIT1).` Open the `UNITDATA.DAT` file and perform a search of the file for the pointname to be used. If the desired pointname is used, a new pointname must be chosen. This will entail a new search that will hopefully produce a pointname that is not being used.

To specify a spare local logic point for use in a Control Sequence Program Modification, open either the `FACTORY.ASG` or `SITE.ASG` file using an ASCII text editor; the file to be opened will depend on where the modification is being made and/or who is making the modification. Modifications to `FACTORY.ASG` should only be made by GEDS and/or BA factory personnel. Modifications made by GE and/or BA field personnel or Customer/Owner representatives should be made in `SITE.ASG.` For the purposes of this example, it is assumed that the modification is being made after the unit has been installed; therefore, the specification of a spare local logic point will be made in `SITE.ASG.`

`SITE.ASG` and `FACTORY.ASG` are the assignment files where spare signal points can be specified (type, name, and scale) to be added to the unit Data Dictionary. The header of these two files contain the codes used to specify what type of point is to be added. For example, a spare variable (or word) in <C> which is a Control Constant (?VQCC), or a spare local logic in <Q> (?LQ). In the example, a spare logic point is added in <Q>; as such, the ?LQ code must be used to specify the type of spare point to be added to the unit Data Dictionary.

```
;  SITE.ASG  - This file contains the non-prom related assignments of
;              database variables as implemented by SITE PERSONNEL during
;              customization of the control sequence program.
;
;***************************************************************************
;Free point (spare) floating assignments for non-hard-coded signal references.
;
;VARIABLE_TYPES DESCRIPTION
;-------------- --------------------------
;?LB            Logic in C, local
;?LBLS          Logic in C, logic state
;?LBPB          Logic in C, push button
;?LBPUB         Byte  in C, private unsigned byte
;?LQ            Logic in Q, local
;?LQLS          Logic in Q, logic state
;?LQPB          Logic in Q, push button
;?LQPUB         Byte  in Q, private unsigned byte
;?VB            Variable(word) in C, local
;?VBAS          Variable(word) in C, analog setpoint
;?VQ            Variable(word) in Q, local
;?VQAS          Variable(word) in Q, analog setpoint
;?VQCC          Variable(word) in Q, control constant
;
;***************************************************************************
;TYPE  SIGNAL_NAME  SCALE   COMMENTS
;----  ----------- -----   -------------------------------------------
```

As shown at the bottom of the example, the `SITE.ASG` (and the `FACTORY.ASG` file, as well) contains a minimum of three columns. The first column, `TYPE,` defines the type of spare signal point to be added to the unit Data Dictionary. The second column, `SIGNAL_NAME,` defines the software signal pointname of the point being added. The third column, `SCALE,` specifies the scale name of the point. The last column (if present), `COMMENTS,` contains comments about the point being added. Comments are not necessary for the unit Data Dictionary, however, they may be helpful in the future to determine why the point was added. All comments must be preceeded by a semicolon (;).

Using an ASCII text editor, add a line to the `SITE.ASG` file (or `FACTORY.ASG,` as appropriate) to specify a spare local logic in <Q>'s Control Data Base which has the software signal pointname `LCOPY01` and the scale type `LOG.` The `SITE.ASG` should contain a line as follows:

```
;**************************************************************************
;TYPE  SIGNAL_NAME  SCALE   COMMENTS
;----  ------------ -----   -------------------------------------------
?LQ    LCOPY01      LOG   ; Local logic required for COPY block mod 4/12/92
```

Exit the ASCII text editor making sure to save the modified UNITDATA.DAT file with the same name. Return to the Main Menu by typing MENU at the DOS prompt.

8-4.3. Step 3 - Create New UNITDATA.DAT File

The unit Data Dictionary must now be updated to include the new software signal pointname of the spare local logic point being added. The file UNITDATA.DAT (one of the unit Data Dictionary files) must contain all of the signal pointnames and therefore must be updated (a **new** UNITDATA.DAT file will actually be created with the new software signal pointname from SITE.ASG included in the file). UNITDATA.DAT is created by executing the DDLOCATE program and using the four assignment files (one of which is SITE.ASG) as input to the program. All four of the assignment files must be used even though only SITE.ASG was modified in this example. This is due to the fact that a new UNITDATA.DAT is actually being created by the DDLOCATE program and all assignments must be included in the new file.

To run the DDLOCATE program, type in the following command at the DOS prompt of the unit directory:

```
DDLOCATE IO.ASG FACTORY.ASG ALLOCSSP.ASG SITE.ASG
```

When the Data Dictionary loader program loads the information from UNITDATA.DAT into <I> RAM, the signal pointnames should be in alphanumeric order by pointname for fast loading. However, the UNITDATA.DAT file created by DDLOCATE is not sorted in this fashion.To accomplish an alphanumeric sorting, run the DDUTIL program with the SORT option specified by typing in the following command at the DOS prompt of the unit directory:

```
DDUTIL SORT
```

In addition to sorting the new UNITDATA.DAT file in alphanumeric order by signal pointname, DDUTIL also performs a validity check of the signal pointnames in the file. Validity checking is the process of checking that no signal pointname is assigned to more than one point in UNITDATA.DAT. The results of sorting and validity-checking will be displayed on the screen. The following message should be displayed indicating that no problems were encountered during sorting and validity-checking:

```
SORTING COMPLETE: UNITDATA.DAT is new file, UNITDATA.BAK is old file
```

DDUTIL run with the SORT option specified will store the unsorted and un-validity-checked UNITDATA.DAT file produced by DDLOCATE as back-up file with the name UNITDATA.BAK and stores the results of sorting the file as the new UNITDATA.DAT file.

NOTE

Several of these steps can be done automatically by using the `MK5MAKE.BAT` file. See Maintenance Manual GEH-5980, Chapter 4.

8-4.4. Step 4 - Reset <I> Computer

The new `UNITDATA.DAT` file must now be loaded into the <I> computer's RAM so that the new software signal pointname can be displayed with units and a value when inserted into a User Defined Display. To load the new `UNITDATA.DAT` file into <I> RAM, the primary operator interface computer must be reset by pressing CTRL + ALT + DEL. When the reset process is complete, the screen will return to the Main Display.

8-4.5. Step 5 - Transfer New Files To Other <I>s

If there are other <I>s on the Stage Link that communicate with the unit and therefore also require the addition of a spare logic point, the new `SITE.ASG` (or `FACTORY.ASG`) and `UNITDATA.DAT` must be copied to floppy disk and transferred to them from the <I> where the modification was made. After the new files are loaded, each <I> must be reset in order for it to display the new point name when entered.

8-5. ADDING AN ANALOG INPUT (T/C, RTD, MA INPUT)

As shipped from the factory, the Mark V Turbine Control contains some spare analog input points. As referred to in this section, an analog input is an input to the control panel which has some analog value associated with it (millivolts from a thermocouple, milliamps from a transducer, or variable voltage feedback from an RTD). Analog inputs differ from digital (or contact) inputs in that they have a variable feedback signal not just a logic " 1 " or a logic " 0 ".

Adding an analog input to the Mark V control panel for use in monitoring, protection, controlling, or logging purposes requires modification of the I/O configuration of the panel. To implement a new input in control or protection, modifications to the Control Sequence Program are also necessary. This section describes the process of selecting a spare analog input point, renaming it, making the necessary modifications to the I/O Configuration, and determining the I/O terminal board locations to terminate the new input. For information on Control Sequence Program modifications see Chapter 5.

The steps involved in finding and renaming a spare analog input point are explained in detail in sections 8-5.1 through 8-5.8.

Step 1	Choose a spare analog input point from the I/O assignment file and rename it
Step 2	Modify the I/O Configuration for proper scaling of the new analog input point
Step 3	Create a new `UNITDATA.DAT` file
Step 4	Download the new I/O Configuration to Processor EEPROMs
Step 5	Reset the control panel processor(s) one at a time
Step 6	Reset the <I> computer to load the new `UNITDATA.DAT` information into <I>'s RAM
Step 7	Transfer the new `IO.ASG` and `UNITDATA.DAT` to any other <I>s on the Stage Link

As an example, three new analog inputs will be added to Unit 1:

- 4-20mA input from a pressure transducer which is externally powered (i.e., a 4-wire device); the transducer is calibrated for 0-1500 psig (4.0 mA = 0.0psig, 20.0 mA = 1500 psig) and is monitoring a 1200 psig steam line
- type K thermocouple which is monitoring the steam temperature in the 1200 psig line (normal steam temperature in the line is 1000 F)
- 10-ohm copper RTD which is in a turbine bearing lube oil (L.O.) drain line (normal L.O. drain temperature between 190 F and 200 F).

8-5.1. Step 1 - Choose Spare Analog Input Point

When a new analog input is to be added to the Mark V control panel, an appropriate scale type must be chosen. This selection can be made from the scale type information in the SCLEDATA.DAT file. To accomplish this task, exit to DOS from the Main Menu by pressing ALT + X and change to the appropriate unit directory (if the modification is to be made to Unit 1, change to `F:\UNIT1`). Sections from the scale type information of the `SCLEDATA.DAT` file are as follows:

```
;/* sc_name   max_scale  |        comments                 /*

;/* PRESH | 16384. psi   | PRESSURE: 16384. PSI            /*
;/* PRESL |  32.00 psi   | PRESSURE:  32.00 PSI            /*
;/* PRESM | 256.00 psi   | PRESSURE: 256.00 PSI            /*
;/* PRESS | 2048.0 psi   | PRESSURE: 2048.0 PSI            /*
;/* TC    |  2048. deg F | TEMPERATURE                     /*
```

The scale name `PRESS` is used for the pressure transducer input (the input will be scaled more appropriately later) and `TC` for the thermocouple and RTD inputs.

After the scale type/name has been determined, a name must be chosen for the new point. Pointnames should be as descriptive as possible (up to a maximum of 12 characters) and follow the signal/device numbering scheme used elsewhere in the control panel. In the example, the new pointnames for the inputs will be:

- `SPH1` for the 4-20 mA transducer input
- `STH1` for the thermocouple
- `LTBRGDRN1` for the RTD

None of the above inputs are critical: the unit/process can survive the failure or loss of the inputs.

Point names can only be assigned to one point number in the Mark V turbine control, therefore, the `UNITDATA.DAT` file must be searched to determine if the pointname to be selected is already in use. This can be done with an ASCII text editor. From the Main Menu, exit to DOS by pressing ALT + X. Change to the appropriate unit directory (for example, if the modification is to be made to Unit 1, change to `F:\UNIT1`). Open the `UNITDATA.DAT` file and perform a search of the file for the pointname to be used. If the desired pointname is used, a new pointname must be chosen. This will entail a new search that will hopefully produce a pointname that is not being used.

To find spare analog input points scroll through the I/O assignment file, `IO.ASG` file, with the ASCII text editor looking for suitable points with the same hardware name and signal name. In our example, we are looking for three spare analog inputs; appropriate sections of the I/O assignment file are as follows:

```
;HDW NAME    SIGNAL       SCALE#       ;COMMENTS

R_R_TC9      R_R_TC9      TC    ;  2048. deg F ; R TBQA- 17  Spare TC
R_R_TC10     R_R_TC10     TC    ;  2048. deg F ; R TBQA- 19  Spare TC
R_R_TC11     R_R_TC11     TC    ;  2048. deg F ; R TBQA- 21  Spare TC

C_C_MAI01    C_C_MAI01    PCT   ; 128.00 %     ; C CTBA- 37  20 MA
C_C_MAI02    C_C_MAI02    PCT   ; 128.00 %     ; C CTBA- 41  20 MA
C_C_MAI03    C_C_MAI03    PCT   ; 128.00 %     ; C CTBA- 43  20 MA
C_C_MAI04    C_C_MAI04    PCT   ; 128.00 %     ; C CTBA- 47  20 MA

C_C_RTD01    C_C_RTD01    TC    ;  2048. deg F ; C TBCA-  1  RTD
C_C_RTD02    C_C_RTD02    TC    ;  2048. deg F ; C TBCA-  4  RTD
C_C_RTD03    C_C_RTD03    TC    ;  2048. deg F ; C TBCA-  7  RTD
```

`HDW NAME` defines:

- processor to which the point is coupled (`R_` or `C_`)
- core to which the input will be terminated in the control panel (`R_` or `C_`)
- type and number of the input signal (`TC` for thermocouple, `MAI` for milliamp input, `RTD` for RTD; the two-digit suffix is the number of the input).

`SIGNAL` defines:

- pointname of the input

`SCALE` # defines the type of input:

- `TC` for temperature compensated scale
- `RTD` for RTD
- `PCT` for percentage

`COMMENTS` defines the terminal board location of the analog input point, for example:

- `TBQA- 17` is terminal number 17 on the analog I/O termination board located on the <R> core (refer to Appendix D for termination information); `CTBA- 37` is terminal 37 on the CTBA analog I/O termination board on the <C> core; and, `TBCA- 1` is terminal number 1 on the TBCA analog termination board on the <C> core.

When choosing spare thermocouple (T/C) inputs, special care must be taken when deciding to add them to <Q>. <Q> T/C inputs are brought into the control panel in three corresponding groups of 15. One group of 15 to each of the TCQA analog input cards of <R>, <S>, and <T>. T/C input #1 is the first T/C routed to the TCQA of <R>; T/C input #16 is the first T/C routed to the TCQA of <S>; T/C input #31 is the first T/C routed to the TCQA of <T>; T/C input #2 is the second T/C routed to the TCQA of <R>; and so on. (See Table 1.) Two cases of thermocouple inputs to <Q> exist, voted and non-voted. Voted, redundant inputs are used in control and protection; non-voted, non-redundant inputs are used primarily in monitoring and logging functions. In the example, a single non-critical T/C input is added. As it does not require voting, it could have been added to either <C> or <Q>.

<Q> Thermocouples		
<R>	<S>	<T>
TC01	TC16	TC31
TC02	TC17	TC32
TC03	TC18	TC33
TC04	TC19	TC34
TC05	TC20	TC35
TC06	TC21	TC36
TC07	TC22	TC37
TC08	TC23	TC38
TC09	TC24	TC39
TC10	TC25	TC40
TC11	TC26	TC41
TC12	TC27	TC42
TC13	TC28	TC43
TC14	TC29	TC44
TC15	TC30	TC45

Table 1. Thermocouple Inputs

For the purposes of this example, it is being added to <Q>. When adding non-voted T/C inputs to <Q>, the other two corresponding processor T/C inputs must be checked to determine if they have been used or are still spares. If either or both of the other corresponding T/C inputs have been used, it must be determined what type of T/C(s) is connected; only a T/C of the same type can be connected to the remaining (corresponding) T/C input(s). In the example, a T/C is connected to spare T/C input #9; therefore, the other two corresponding T/C inputs to <S> and <T>, numbers 24 and 39 should be checked to determine if they have been used. If they have, it should be determined what type of T/C they are connected to. For the purposes of the example, no T/C's have been connected to 24 or 39. In the future, however, if T/C's were added to either T/C inputs 24 or 39, they would have to be non-voted type K's since a type K has been added to #9.

If voted, the redundant T/C's must all be the same type and must all be connected to unused corresponding T/C inputs. For example, to connect three redundant T/C's to the control panel beginning at T/C input #9, T/C inputs 24 and 39 must be unused. To make three corresponding T/C inputs voted, the `IO.ASG` file must be modified to let the control panel know that the three redundant signals are to be voted as shown in the following section of the `IO.ASG` file. T/C inputs 1, 16, and 31 have been identified as voted inputs by renaming the signal name to TTI. T/C inputs 9, 24, and 39 are not identified as yet as voted inputs (the hardware name and signal name are the same).

```
;HDW NAME    SIGNAL       SCALE#       ;COMMENTS

TC01_16_31_V TTI          TC    ;  2048. deg F ; TCQA_TC01_Q Inlet steam line
TC02_17_32_V TC02_17_32_V TC    ;  2048. deg F ; TCQA_TC02_Q
TC03_18_33_V TC03_18_33_V TC    ;  2048. deg F ; TCQA_TC03_Q
TC04_19_34_V TTBYP        TC    ;  2048. deg F ; TCQA_TC04_Q Byp stm disch temp
TC09_24_39_V TC09_24_39_V TC    ;  2048. deg F ; TCQA_TCO9_Q
```

Once available and appropriate spare analog input points have been identified, change the name in the `SIGNAL` column to the desired signal pointname and modify the scale type, if necessary. Also, if the maximum scale information for the point exists in the comments section, modify it to match the scale type information from the SCLEDATA.DAT file. For the necessary changes to implement the new analog inputs in the example:

```
;HDW NAME   SIGNAL       SCALE#       ;COMMENTS

R_R_TC9     STH1         TC    ;  2048. deg F ; R TBQA- 17  Stm Hdr Temp T/C#1
R_R_TC10    R_R_TC10     TC    ;  2048. deg F ; R TBQA- 10  Spare TC
R_R_TC11    R_R_TC11     TC    ;  2048. deg F ; R TBQA- 11  Spare TC

C_C_MAI01   SPH1         PRESS ; 2048.0 psi   ; C CTBA- 37  Stm Hdr Pres Xdcr#1
C_C_MAI02   C_C_MAI02    PCT   ; 128.00 %     ; C CTBA- 41  20 MA
C_C_MAI03   C_C_MAI03    PCT   ; 128.00 %     ; C CTBA- 43  20 MA
C_C_MAI04   C_C_MAI04    PCT   ; 128.00 %     ; C CTBA- 47  20 MA

C_C_RTD01   LTBRGDRN1    TC    ;  2048. deg F ; C TBCA-  1  #1 Brg Drn RTD
C_C_RTD02   C_C_RTD02    TC    ;  2048. deg F ; C TBCA-  4  RTD
C_C_RTD03   C_C_RTD03    TC    ;  2048. deg F ; C TBCA-  7  RTD
```

Exit the ASCII text editor making sure to save the modified I/O assignment file with the same name. Return to the Main Menu for the next step by typing `MENU` at the DOS prompt and pressing Enter.

8-5.2. Step 2 - Modify I/O Configuration

The I/O configuration must now be modified to reflect the new analog input points. Click on the `I/O Configurator` selection of the Main Menu. After a few seconds, the I/O configurator display will appear on the CRT. Check the top of the I/O Configurator display for an indication that the processor which is currently selected (either <C> or <Q>). Click on the proper target at the bottom of the screen to switch to the appropriate processor, if necessary.

The example requires that the <Q> Processor Menu be selected. To do this, click on the appropriate I/O card target (TCQA-1) to display the configuration information for thermocouple input number 9.

The first screen which appears will be card revision information; click on the `Next Screen` target until the thermocouple input information screen appears (See Figure 8-6.).

```
                 TCQA Card Definition - Socket 1 - Screen 13/19

                      Thermocouple Type Selection

      TC  1:  K          TC  5:  K          TC  9:   K         TC 13:   K
      TC  2:  K          TC  6:  K          TC 10:   K         TC 14:   K
      TC  3:  K          TC  7:  K          TC 11:   K         TC 15:   K
      TC  4:  K          TC  8:  K          TC 12:   K

            Valid types:    K, J, E, T     not used :    -
```

Figure 8-6. I/O Configurator Screen for <Q> Thermocouple Type Selection

Since the I/O Configuration for the TCQA T/C input #9 is already defined to be a type K and we are adding a type K input to T/C input #9, no changes are necessary on this display.

Click on the `Exit Card Def` target to return to the initial screen of the I/O configurator. In the example, two new analog inputs are added to <C>, so the user should click on the Load <C> Menu target and then click on the appropriate card target. (Both the CTBA and TBCA analog I/O termination boards are connected to the TCCA card; therefore, the user must click on the TCCA card target.) The first screen which appears displays card revision information; click on the `Next Screen` target until the proper input configuration information is displayed.

To configure the new transducer input, access the Milliamp Input Definition screen of the <C> processor's TCCA card, shown in Figure 8-7.

```
                 TCCA Card Definition - Socket 1 - Screen 4/7
                     Milliamp Input Definitions   1 - 8
                     Signal      Full Scale    CDB value    CDB value
                     in use      CDB value      at 4 mA      at 20 mA
                     -------    -----------    ---------    ----------
       Signal  1:      NO         128           0.0          100.0
       Signal  2:      NO         128           0.0          100.0

       Signal  3:      NO         128           0.0          100.0
       Signal  4:      NO         128           0.0          100.0

       Signal  5:      NO         128           0.0          100.0
       Signal  6:      NO         128           0.0          100.0

       Signal  7:      NO         128           0.0          100.0
       Signal  8:      NO         128           0.0          100.0
```

Figure 8-7. <C> Processor TCCA Card Milliamp Input Definitions Screen

Milliamp Input Signal #1 is used. The cursor is on the area of the screen which is white. Click on `NO` in the `Signal in use` column for `Signal 1` and type in `YES,` followed by Enter. The cursor will advance to the next column of `Signal 1`. In this column, type in the `Full Scale value CDB value` (Control Data Base) of the signal type which was chosen and then press Enter (in this case, that would be 2048 for the signal type PRESS). The cursor will advance to the next column of `Signal 1, CDB Value at 4mA,` where the minimum value of the calibrated range of the transducer should be typed in before pressing Enter (in the example the value 0.0 was typed in since the transducer will be calibrated for a range of 0-1500 psig). The cursor will advance to the last column of `Signal 1, CDB Value at 20mA ,` where the maximum calibrated range of the transducer should be typed in before pressing Enter. In the example, the maximum calibrated range of the transducer is 1500.0 psi, therefore the value 1500.0 should be typed in. The Milliamp Input Definition Screen in the example would appear as follows:

```
                     Signal      Full Scale    CDB value    CDB value
                     in use      CDB value      at 4 mA      at 20 mA
                     -------    -----------    ---------    ----------
       Signal  1:      YES       2048           0.0          1500.0
       Signal  2:       NO        128           0.0          100.0
```

To complete the Milliamp Input Definition, click on the `Verify Screen` target at the bottom of the display to verify any modifications. Click on the `Exit Card Def` target to return to the <C> Processor Configuration Menu.
In the example, an RTD input is added to <C>. For this reason, the user should click on the `TCCA` target selection again, and scroll to the proper RTD Type Selection screen (as shown in Fig.8-8).

```
                    TCCA Card Definition - Socket 1 - Screen 3/7
                        RTD Type Selection

    RTD  1:   DIN     RTD  9:   DIN      RTD 17:    DIN   RTD 25:    DIN
    RTD  2:   DIN     RTD 10:   DIN      RTD 18:    DIN   RTD 26:    DIN

    RTD  3:   DIN     RTD 11:   DIN      RTD 19:    DIN   RTD 27:    DIN
    RTD  4:   DIN     RTD 12:   DIN      RTD 20:    DIN   RTD 28:    DIN

    RTD  5:   DIN     RTD 13:   DIN      RTD 21:    DIN   RTD 29:    DIN
    RTD  6:   DIN     RTD 14:   DIN      RTD 22:    DIN   RTD 30:    DIN

    RTD  7:   DIN     RTD 15:   DIN      RTD 23:    DIN
    RTD  8:   DIN     RTD 16:   DIN      RTD 24:    DIN


         Valid types:   N120, CU10, USIND, SAMA, PURE, DIN
             not used :    -----
```

Figure 8-8. I/O Configurator Screen - TCCA Card RTD Type Selection

RTD Input No.1 is used. Click on `DIN` next to `RTD` 1: to move the cursor to that location. As a 10-ohm copper RTD is being implemented, the string, CU10, should be typed in before striking the Enter key. The RTD Selection Screen in the example would appear as follows:

```
                        RTD Type Selection

    RTD  1:  CU10     RTD  9:   DIN      RTD 17:    DIN   RTD 25:    DIN
    RTD  2:   DIN     RTD 10:   DIN      RTD 18:    DIN   RTD 26:    DIN
```

To complete the RTD Type Selection, click on the `Verify Screen` target at the bottom of the display to confirm the modification and then click on the `Exit Card Def` target to return to the <C> Processor Configuration Menu.

Once the analog input I/O modifications is complete, click on the Save and Exit target at the bottom of the display. This saves the new I/O configuration files to be downloaded to the processors in Step 4, exits the I/O Configurator, and returns the screen to the Main Menu.

8-5.3. Step 3 - Create New `UNITDATA.DAT` File

The unit Data Dictionary must now be updated to include the new software signal pointname of the contact input point being added. The file `UNITDATA.DAT` (one of the unit Data Dictionary files) must contain all of the signal pointnames and therefore must be updated (a new `UNITDATA.DAT` file will actually be created with the new software signal pointname from `IO.ASG` included in the file). `UNITDATA.DAT` is created by executing the `DDLOCATE` and using the four assignment files (one of which is `IO.ASG)` as input to the program. All four of the assignment files must be used even though only `IO.ASG` was modified in this example. This is because a new `UNITDATA.DAT` is actually being created by the `DDLOCATE` program and all the assignments must be included in the new file.

To run the `DDLOCATE` program, type in the following command at the DOS prompt of the unit directory:

```
DDLOCATE IO.ASG FACTORY.ASG ALLOCSSP.ASG SITE.ASG
```

When the Data Dictionary loader program loads the information from UNITDATA.DAT into <I> RAM, the signal pointnames should be in alphanumeric order by pointname for fast loading. However, the UNITDATA.DAT file created by DDLOCATE is not sorted in this fashion.To accomplish an alphanumeric sorting, run the DDUTIL program with the SORT option specified by typing in the following command at the DOS prompt of the unit directory:

```
DDUTIL SORT
```

In addition to sorting the new UNITDATA.DAT file in alphanumeric order by signal pointname, DDUTIL also performs a validity check of the signal pointnames in the file. Validity checking is the process of checking that no signal pointname is assigned to more than one point in UNITDATA.DAT. The results of sorting and validity-checking will displayed on the screen. The following message displays, indicating there were no problems during sorting and validity-checking:

```
SORTING COMPLETE: UNITDATA.DAT is new file, UNITDATA.BAK is old file
```

DDUTIL run with the SORT option specified will store the unsorted and un-validity-checked UNITDATA.DAT file produced by DDLOCATE as back-up file with the name UNITDATA.BAK and stores the results of sorting the file as the new UNITDATA.DAT file. Return to the IDOS Main Menu by typing MENU at DOS prompt and pressing Enter.

NOTE

Several of these steps can be done automatically by using the MK5MAKE.BAT file. See Maintenance Manual GEH-5980, Chapter 4.

8-5.4. Step 4 - Download New I/O Configuration

Access the Main Menu and initiate the EEPROM downloading function by pointing and clicking on EEPROM Downloader. This returns the screen to a command line with the following prompt: EEPROM Downloader>.

To download IOCFG_Q.DAT to each processor of <Q> (<R>, and <S> and <T> in a TMR control panel), type in the following command with the appropriate Unit Name (if other than T1):

```
DOWN T1 R IOCFG
```

Repeat the command for <S> and <T> in a TMR control panel. When the I/O configuration has been downloaded to the appropriate processor(s), type EXIT at the prompt to return to the Main Menu when downloading is completed.

8-5.5. Step 5 - Reset Processor(s)

For the new analog input addition(s) to become active in the processor(s) to which it was downloaded, each processor must be reset or cycled **one-at-a-time.** Wait until the processor returns to state A7, normal operating condition, before resetting the next processor. Resetting the processor transfers the modified I/O configuration file from processor EEPROM to processor RAM where it is now active.

8-5.6. Step 6 - Reset <I> Computer

The new UNITDATA.DAT file must now be loaded into the <I> computer's RAM so that the new software signal pointname can be displayed with units and a value when inserted into a User Defined Display. To load the new file into <I> RAM, the primary operator interface computer must be reset (press CTRL + ALT + DEL) for the new UNITDATA.DAT file to be loaded into <I> RAM. When the reset process is complete, the Main Display will return.

8-5.7. Step 7 - Transfer New Files to Other <I>s

If there are any other <I>s on the Stage Link for the unit to which new analog input were added, the new `IO.ASG, UNITDATA.DAT, and IOCFG_?.DAT` (where ? is either C or Q, depending on which processor's digital I/O configuration was modified) must be copied to floppy disk and transferred to them from the <I> where the modification was made. After the new files are loaded, each <I> must be reset in order for it to display the new point name when entered.

Notes:

CHAPTER 9

STAGE LINK CONFIGURATIONS

9-1. OVERVIEW

Communication between the operator interface(s) (<I>) and the Mark V panel(s) is carried out by means of the control system's Stage Link. In its simplest configuration, the Stage Link connects one Mark V turbine control panel to a single <I> (or node) across a single segment (see segment definition below). This communication topology may however be expanded to accommodate multiple <I>s and/or multiple panels. For example, a single operator interface can be configured to issue commands to and receive turbine data from up to eight Mark V gas/steam turbine controls. In addition, multiple <I>s may be attached to the Stage Link — each <I> communicating with multiple control panels. In this way, the Stage Link provides enhanced flexibility for establishing effective communications which can be tailored to individual site needs.

The Stage Link was designed specifically to address turbine control needs such as downloading or uploading software between the Mark V and the <I>, issuing commands, alarm management, and monitoring. Distributed control systems (DCS) interface to the Mark V via separate communication link(s) routed to the <I>, typically using a ModBus protocol.

9-2. INTRODUCTION

This Chapter provides guidance and rules for successfully mapping Stage Links. Examples are used to help explain how certain topologies maximize communication link availability and/or enhance network distances.

9-2.1. Terms of Reference

- **NODE**: Any device connected to the Stage Link system that has a valid address; a <C> core (communications processor), <D> core (backup communications processor), or an <I>.

- **REPEATER**: Electronic device that receives, amplifies and re-transmits Stage Link signals. The <C> and <D> cores and a hub are considered repeaters, <I>'s are not.

- **HUB**: For the Stage Link, the hub is a 4-port **repeater** (two coax ports and two fiber optic ports) for converting electrical signals to or from light pulses for fiber optic transmission or reception. It can also be used as a repeater to amplify coax signals. It is not considered a node because it does not have an address.

- **SEGMENT**: Any Stage Link section that joins two repeaters **or** connects one repeater to one or more high impedance devices and ends with a terminating resistor. A segment may have multiple taps for high impedance connections to <I>'s.

9-3. STAGE LINK CHARACTERISTICS

The Stage Link consists of a 2.5 MHz/2.5 Megabit-per-second ARCNET system that uses either fiber optic or standard RG-62 A/U copper cabling. Either type can be purchased with a variety of insulation systems such as flame retardant teflon or high density polyethylene.

In applications that must meet IEC codes, GEDS recommends using armored co-axial cable. These cable types have a metal sheath outer layer that functions as both a mechanical shield and as an electrical conductor that can alleviate lightning induced disturbances on short outdoor runs. This outer layer must be grounded at each building's entrances and exits. Fiber optic cabling prevents electromagnetic interference and is often a better alternative for long outdoor segments (See Section 9-10).

General Specifications	
Local Area Network (LAN) Type	ARCNET
Communication Type	Baseband
Frequency/Speed	2.5 MHz/ 2.5Mbps
Propagation Delay (Maximum)	31 micro seconds
Maximum Network Length, based on Propagation Delay	6,000 meters or 19,680 feet
Repeater Nodes	<C>, <D>
Other Repeaters	Fiber optic Hubs
High Impedance Nodes	<I>

9-3.1. The Mark V Panel

Within the Mark V panel, the ARCNET cable should be connected to the CTBA card, which is located within the <C> (or <D>) communication processor(s). The CTBA card communicates directly with the <C> or <D> communication processor(s). This data exchange is carried out through the one internal port of a three port repeater; the remaining two ports are for external customer use. Signals entering any one of these three legs are amplified and sent out through the other two. Therefore, a signal entering the first external port will be sent to <C> and re-transmitted on the second external port. Signals entering the internal port will be sent out on both external ports. Should the CTBA card loose control power, a relay de-energizes and connects the two external ports. In this manner, all the other nodes on the Stage Link can continue to function as long as the topology is designed in accordance with the distance rules provided later in this Chapter.

9-3.2. The Primary Operator Interface, <I>

The operator interface (or <I>) utilizes a single high impedance port that distributes signals in both directions on the Stage Link via a "T" type connector. The ARCNET card within the <I> receives data by tapping off a portion of signal transmitted on the Stage Link.

9-4. CABLE RECOMMENDATIONS

Copper Cable Recommendations	
Indoor Cable	RG-62 A/U Co-axial Cable
Outdoor Cable	Armored Co-axial or Tri-axial Cable
Connector Type	BNC Male (both ends)
Fiber Optic Cabling Recommendations	
Cable	Multi-mode with 62.5 micron core/120 micron cladding
Connector Type	ST bayonet
Hub Power	120 VAC/60 HZ or 240 VAC/50 HZ (Customer supplied)
Hub Configuration	2 Co-axial ports and 2 Fiber-optic ports (4 ST type connectors)

If the turbine control application requires a segment too long for a co-axial cable, a fiber optic cable should be used. For more on fiber optic installation, see section 9-10.

9-5. STAGE LINK RULES

Summary of Topology Rules	
No Loops	
Maximum Number of Nodes Allowed	100
Maximum Number of <I>s in one network	16
Maximum Time Delay between any two Nodes	31 microseconds
Both Ends Of The Stage Link Must Have A 93 ohm Terminating Impedance	
Every Node must have a unique network address	
Maximum Segment Lengths	
Co-ax Repeater To Repeater	609.6 meters
Co-ax Repeater To Single <I>	609.6 meters
Co-ax Repeater to More Than One <I>	304.8 meters
<I> Between Two Repeaters	304.8 meters
Maximum <I>s Per Segment	8
Fiber Optic Cable Hub To Hub (62.5/120 micron fiber)	1825 meters
Minimum cable length between <I>s	1.5 meters

9-6. SEGMENT RULES

1. The cable and nodes between two repeaters is called a segment. The segment distance cannot exceed 609.6 meters for a coax connection. Fiber optic segments can go farther as described below.

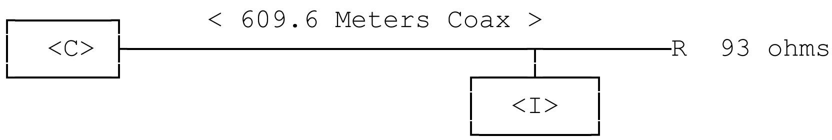
```
┌──────┐    < 609.6 meters, coax >     ┌──────┐
│ <C>  ├───────────────────────────────┤ <C>  │
└──────┘                               └──────┘
```

If a <C> or <D> looses power total length becomes the sum of the two adjacent segments (see rule 8 below).

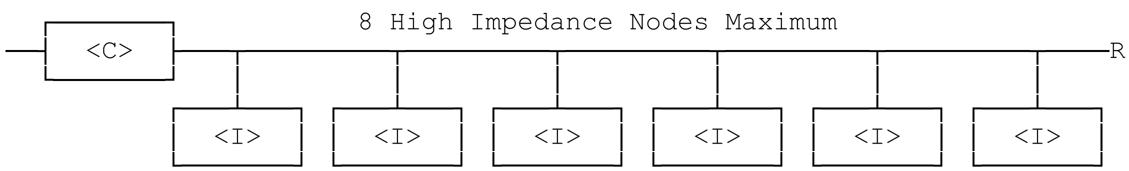
2. The segment between fiber optic repeaters can be as much as 1825 meters.

The typical fiber optic hub is a 4 port repeater
that contains two coax and two fiber optic ports.

3. A two node segment with one repeater and one high impedance node forming the end of the link may be 609.6 meters long. **The <I> must have a cable terminating resistor (shown as R).**

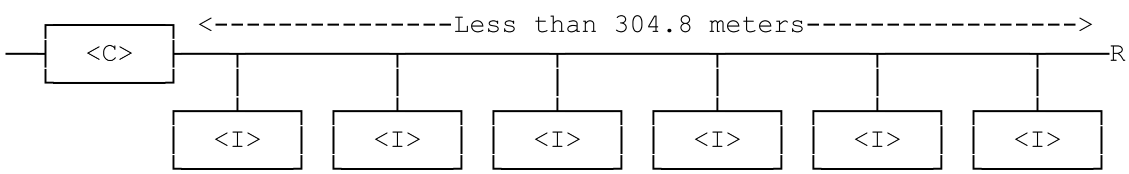


4. No more than 8 <I>'s can be used in one segment. Each node must use a proper "T" connector in the cable to minimize reflection. The "T" is located on the card, and does not have a length of cable between the "T" and the card.

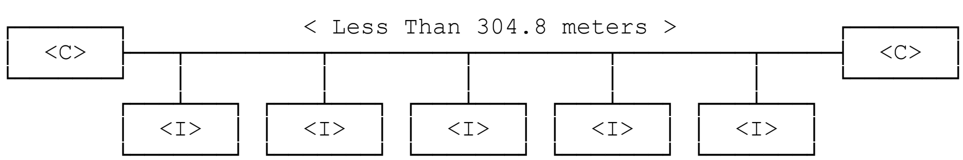


5. High impedance nodes must be separated by a minimum of 1.5 meters of cable between the T's.

6. On a segment that has one <C> and two or more high impedance nodes, the maximum segment cable distance must be limited to 304.8 meters or less.



7. The length of a segment with two <C>'s, and one to eight <I>'s, must be limited to 304.8 meters or less.



8. Each <C> and <D> repeater has a relay that drops out when the power is off, connecting the two ports in order to maintain communication among the remaining nodes. This complicates the distances allowed between nodes because the segment formed by any single failure still must not exceed 609.6 meters. For segments containing <I>'s this distance drops to 304.8 meters.

```
                  <---------------Less than 304.8 meters----------------->
                             and no more than eight <I>'s
        ┌───────┐                                   ┌──────┐
       ─┤  <C>  ├───────┬─────────┬─────────┬───────┤ <C>  ├─────┬─────────┬─────────┬────R
        └───────┘       │         │         │       └──────┘     │         │         │
                ┌───────┴┐ ┌──────┴─┐ ┌─────┴──┐          ┌──────┴─┐ ┌─────┴──┐ ┌────┴───┐
                │  <I>   │ │  <I>   │ │  <I>   │          │  <I>   │ │  <I>   │ │  <I>   │
                └────────┘ └────────┘ └────────┘          └────────┘ └────────┘ └────────┘
```

9-7. TOTAL EFFECTIVE DISTANCE RULES

Always calculate the total "effective cable distance" between the two network nodes that have the longest **effective** distance. This is not always the nodes that are farthest apart physically. Each repeater has a delay equal to the delay in 25 meters of cable. Effective distance is calculated as follows:

Copper (coax) cable length + (Fiber optic cable length * 1.25) + (number of repeaters * 25 meters)
with all lengths in meters

Total "effective cable distance" may not exceed 6000 meters.

The maximum 31 micro second propagation delay is approximately equal to the delay in 6000 meters of cable. As it is easier to calculate "effective cable distance" than it is to measure propagation delay, this approximation is used. The deciding factor is **propagation delay, not total length.** If questions arise about a particular application, it may be necessary to measure the propagation delay.

9-8. REDUNDANT SYSTEM RULES

The two Stage Link systems, <C> and <D>, are not interconnected. Interconnections are not allowed as they will lower the communication interface reliability; that is, both networks could be brought down by a common failure such as a shorted coax or one node continuously transmitting.

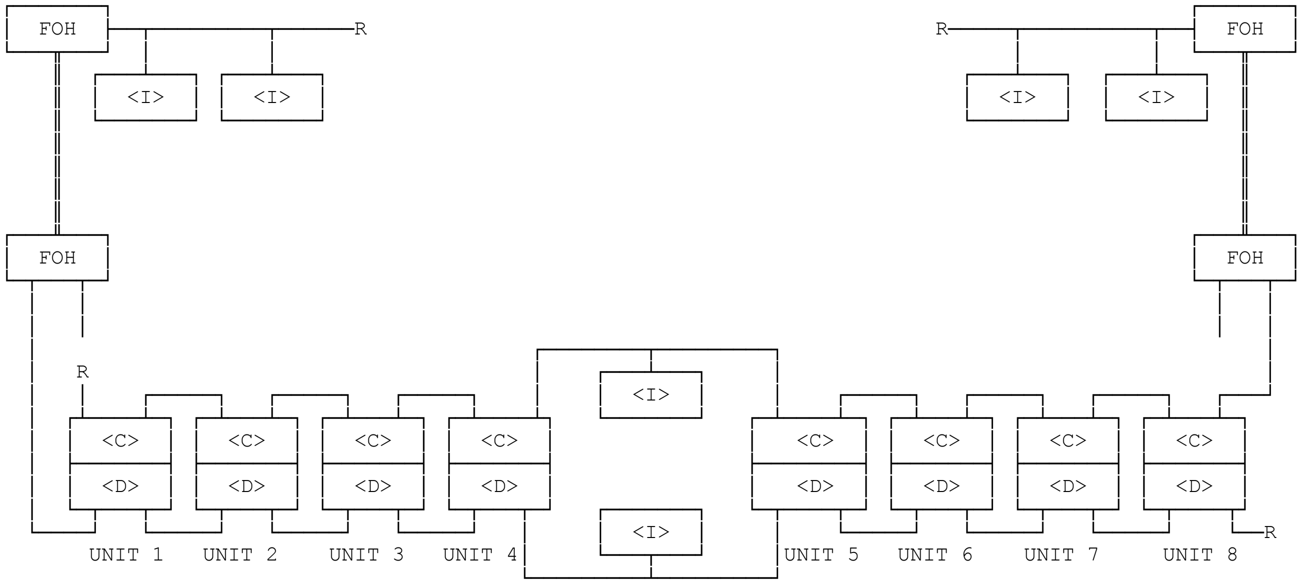
The <C> and <D> cables can be routed independently to minimize opportunities for a common failure.

Typically unit control nodes are connected in a "daisy-chain" configuration. A cable from the <C> Stage Link is routed from one physical end of the daisy-chain to the central control room. The <D> cable to the central control room is attached to the opposite side of the unit control string. In this way, a break in the cable or loss of power to a <C> and/or <D> leaves all other nodes accessible from some <I> in the central control room.

On a system with a fiber optic link, the fiber optic repeater pair does not have bridging relays. Some customers may therefore prefer to use two links to the central control room, one from each end of the chain of unit controls as shown on the Figure of Example 1. In this example, one set of <I>s in the central control room is operational if any one of the four fiber optic active hubs fails.

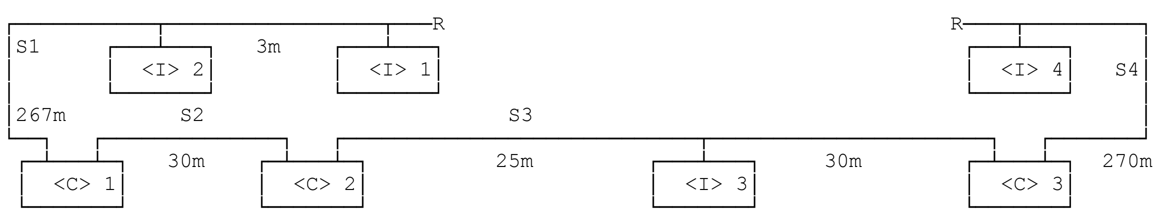
9-9. EXAMPLES

9-9.1 Example 1: Redundant Link Stage Application



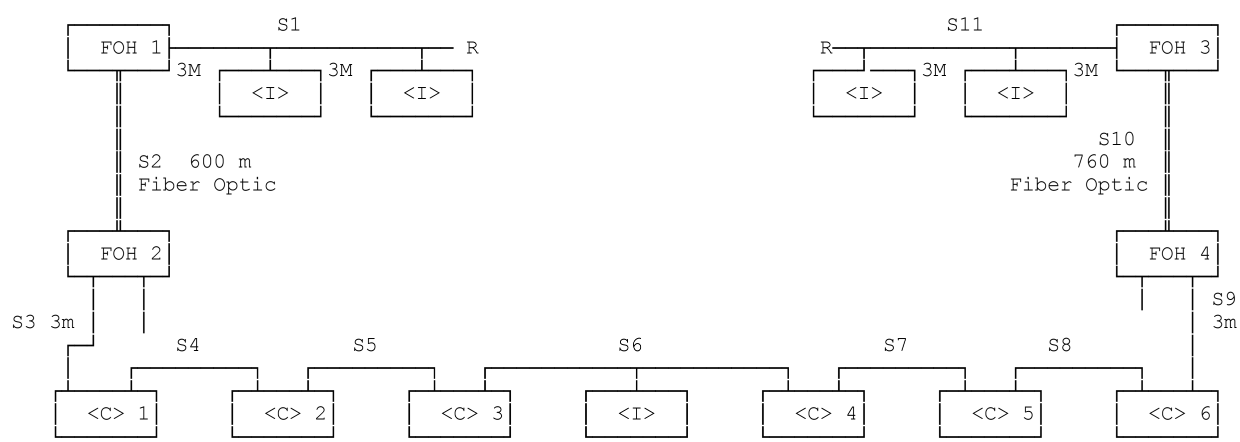
<C> - Communications Processor
<D> - Back up Communications Processor
FOH - Fiber Optic Hub
<I> - Operator Interface
R - Terminating Resistor (93 ohms)

9-9.2. Example 2: A Simple Plant Application



	SEGMENTS				TOTALS and COMMENTS
	S1	S2	S3	S4	
Cable length, meters	270	30	55	270	625
Effective cable length	3 <C>s at 25 = 75, plus 625				700, well below 6000 meter limit
Maximum Segment with 1 Node Failure Calculations	Failure of Node:			* All combined segments have <I>'s, thus the maximum is 304.8 meters. <C> 2 to <I> 4 the <C> 1 to <C> 2 segment. <I> 1 to <C> 2 is close to the limit but should work if <C>1 fails, depending upon conditions.	
	<C> 1	<C> 2	<C> 3		
	Results in combined segment of:				
	<I>1 to <C> 2	<C> 1 to <C> 3	<C> 2 to <I> 4		
	300	85	325*		
Total number of Nodes					7, well below 100 maximum

9-9.3. Example 3: Complex Plant Application



<C> - Communications Processor
FOH - Fiber Optic Hub
<I> - Operator Interface

<table>
<tr><th rowspan="2"></th><th colspan="11">SEGMENTS</th><th rowspan="2">TOTAL</th></tr>
<tr><td>S1</td><td>S2</td><td>S3</td><td>S4</td><td>S5</td><td>S6</td><td>S7</td><td>S8</td><td>S9</td><td>S10</td><td>S11</td></tr>
<tr><td>Fiber cable length</td><td></td><td>600</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>760</td><td></td><td>1360</td></tr>
<tr><td>Co-ax cable length</td><td>6</td><td></td><td>3</td><td>30</td><td>30</td><td>180</td><td>30</td><td>150</td><td>3</td><td></td><td>6</td><td>438</td></tr>
<tr><td>Max. Effective cable length</td><td colspan="11">(6 <C>s at 25) + (4 FOH at 25) + {1.25 * (600 + 760)} + 438</td><td>2388, well within 6000 meter maximum</td></tr>
<tr><td rowspan="5">Maximum Segment with 1 Node Failure Calculations</td><td colspan="11">Failure of Node:</td><td rowspan="5">All are less than 609 or 304 maximums.</td></tr>
<tr><td colspan="2"><C> 1</td><td colspan="2"><C> 2</td><td colspan="2"><C> 3</td><td colspan="2"><C> 4</td><td colspan="2"><C> 5</td><td><C> 6</td></tr>
<tr><td colspan="11">Results in combined segment of:</td></tr>
<tr><td colspan="2">FOH 2 to <C> 2</td><td colspan="2"><C>1 to <C> 3</td><td colspan="2"><C> 2 to <C> 4</td><td colspan="2"><C> 3 to <C> 5</td><td colspan="2"><C> 4 to <C> 6</td><td><C> 5 to FOH 4</td></tr>
<tr><td colspan="2">33</td><td colspan="2">60</td><td colspan="2">210</td><td colspan="2">210</td><td colspan="2">180</td><td>153</td></tr>
<tr><td colspan="7">Total number of Nodes</td><td colspan="6">15, much less than 100 maximum</td></tr>
</table>

9-10. FIBER OPTICS

Fiber optics can be an effective substitute for copper coax cabling, especially in cases where longer distances are required, or electrical disturbances are a serious problem.

9-10.1. Advantages

- Larger diameter fiber extends this to 9000 feet (2,740 m) because the light transmitter gets more light into the fiber.
- If the plant is in a high lightning area, fiber optic segments can reduce the control outages caused by lightning.
- Grounding problems are avoided with optical cable. The ground potential can rise when there is a ground fault on transmission lines, caused by currents coming back to the generator neutral grounding point.
- Optical cable can be routed through switch yards and other electrically noisy areas with no interference. This can shorten the required runs and simplify the installation.
- With proper cable jacket materials it can be run direct buried, in trays, or in conduit, being careful not to drop below the bend radius.
- High quality optical fiber cable is light, tough, and easily pulled.
- The total cost of installation and maintenance of a fiber optic segment may be less than a coax segment.

9-10.2. Disadvantages

- Fiber optic links require powered hubs, with a reliable source of AC power. Failure of power to either hub causes a link failure.
- The effective distance of a fiber segment is 1.25 times the actual cable routing distance. The rule for Stage Link is that the total effective distance between the farthest apart devices must not exceed 20,000 feet.
- The extra equipment required for fiber links can increase maintenance.
- Cost, particularly for short runs, may be more for a fiber optic Stage Link segment.
- Inexpensive fiber optic cable is easily broken during installation and more prone to mechanical and performance degradation over time. The highest quality cable is recommended.

9-10.3. Review of Components

This section will review all of the basic components of a fiber optic system, including cable, hubs, and connectors.

9-10.3.1. BASICS. The recommended fiber optic hub accepts two copper coax connections and two fiber optic links. A message coming in on any one of the four ports is repeated out the other three ports. Each fiber port consists of an outgoing fiber and an incoming fiber. The incoming signal is picked up with a photo transistor and converted to an electrical signal. The outgoing signal is converted from a train of electrical pulses to infrared light using a light emitting diode. On the fiber segment the optical output of one hub is connected through the fiber cable to the optical input of the other hub. Two fibers are needed for each segment.

Multimode fiber, with a graded index of refraction core and an outer cladding, is recommended for the Stage Link (See Section 9-4.). The amount of light that gets into the fiber depends on the brightness of the light source and the area of the light carrying portion of the fiber. The amount of light that comes out the other end depends on the clarity of the glass, the distribution of the index of refraction, the condition of the fiber, and the attenuation of connectors. The amount of electrical signal generated depends on the light coming out of the fiber and the area and sensitivity of the photo transistor. Tracking all this is done by using a power budget (Section 9-10.4.).

The fiber is protected with "buffering" which is the equivalent of insulation on metallic wires and protects the cable from excessive bends. Mechanical stress can damage fibers. One way to protect the fiber is to spiral it on the inside of a tube filled with gel. A more reliable system uses tight buffering with precision tensioned Kevlar fibers which carry the stress of pulling and vertical runs.

WARNING

Never look directly into a fiber. Although most fiber links use light emitting diodes which cannot damage the eyes, some longer fiber links use lasers which can cause permanent damage to the eyes.

9-10.3.2. CABLE.

- High quality fiber is recommended, especially for long distance links. It should be 62.5/125 optical cable as well.
- Cable attenuation should be between 3.0 and 3.3 dB/km at 850 nm, and around 1 to 1.2 dB/km at 1300 nm.
- The acrylate protective layer of the fiber should be specified with a 100 kpsi proof test and a 500 micrometer coating, rather than the 50 kpsi and 250 micrometer coating.
- Gel filled (or "loose tube") cables should not be used because of the special care required during installation, the difficulty of making terminations and problems of maintaining the gel seal, particularly in vertical runs where hydrostatic pressure can cause gel leakage.
- Use a high quality "break out" cable, which will make each fiber in itself sturdy cable that helps prevent too sharp bends.
- Combine the sub-cables with more strength and filler members to build up the cable for resisting mechanical stress and outside environment attack.
- In Section 9-10.6, there are two sample specifications for fiber cable; one without armor and one with armor. Rodent damage is one of the **major** causes of failure of optical cable. If there is a possibility of wire insulation damage from rodents, the armored cable should be chosen. Otherwise, the armor is not recommended because it is heavier, has a larger bend radius, is more expensive, attracts lightning currents, and has lower impact and crush resistance. Particularly for underground runs, a direct lightning strike through the earth to the cable shield can cause explosive formation of steam in damp earth that can mechanically damage the cable.
- Test the optical characteristics of the cable with either an optical time domain reflectometer (OTDR) which can be provided by the manufacturer or with a simpler device that compares light levels at both ends of the cable.
- Four-fiber cables can be used to bring redundant communications to a central control room, or the extra fibers can be retained as spares. A less expensive option is to get the same cable with only two fibers.

9-10.3.3. HUBS. The type of hub described throughout this chapter is built particularly for ARCNET communications and has the proper impedance to match the ARCNET line (93 ohms). For this reason a fiber hub intended for ETHERNET, for example, will not function properly on the Stage Link. It consists of a power supply that runs from 120 or 220 volts AC 50 or 60 Hz. The two models can be converted by moving an internal jumper to accept the other voltage in case an error was made in ordering.

The rack contains a power supply with sufficient power for 4 "expansion" cards. Each card has two copper coax ports and two fiber optic ports. A signal coming in on any port will be amplified and transmitted on the other three ports. Ordinarily only one card is used. If the system uses two fiber segments, a second hub is recommended to improve the communications availability. On the card, normally only one copper port and one fiber port are used for the same reason. The fiber optic ports in the hub's card have ST connectors which are the bayonet type. The light gray is the transmit port, the dark gray the receive port. In service a light gray connector always attaches to a dark gray one.

9-10.3.4. CONNECTORS. Connectors come in two forms, SMA and ST. The ST connectors give less problems in the field because they are bayonet type and not subject to over tightening. Over tightening the SMA connector can chip the glass fiber surface causing problems with reflections and loss of transmission. The bayonet type effectively uses a spring to push the two connecting fibers together with the proper force.

Ceramic, glass filled plastic, or stainless steel are used to make the connectors. They come in three standard sizes to fit the different diameter fibers. Connectors for the 62.5/125 micron fiber are relatively easy to procure. The ceramic connectors can be precisely made and match the coefficient of expansion of glass.

9-10.4. System Considerations

Having two <I>'s in the central control room allows one to be down for maintenance while continuing control of the turbines from the control room from the remaining <I>. Similarly, having two fiber segments also allows for failure of one of the hubs, or the power to it. A failure of any one of the copper segments also retains control of all machines.

Often only <I>'s and possibly an <H> (Historian) will be near the central control room. It will have reliable AC, and if that AC is gone, control from that location stops. Therefore, reliable AC in the control room is satisfactory for the hubs as well.

Another system consideration is the optical power budget for the Stage Link. The total budget refers to the brightness of the light source divided by the sensitivity of the light receiver. These ratios of power are usually measured in dB to make calculations easier. The difference between the dB power of the source and the dB power of the receiver represents the total power budget. This must be compared to the link loss budget which is made up of the loss in the connectors and optical cable. Installation of the fiber can decrease its performance over the new cable condition. The LED light source can get dimmer over time, the connections can get dirty, the cable loss increase with aging, and the receiver can become less sensitive. For all these reasons there must be a margin between the available power budget and the link loss budget of a minimum of 3 dB. A good installation, including using correct parts and cabling, preparing connectors properly, and laying the cable so as to avoid sharp bends and hot locations will help keep availability.

One fiber segment is specified to operate as far as 6000 feet with 62.5/125 microns fiber, and 9000 feet for the 100/140 microns fiber, by the hub manufacturer. These distance limitation have been incorporated into the Stage Link layout rules. It is recommended that fiber optic sections used in the Stage Link must not be longer than the specified 6,000 and 9,000 feet for the two fiber diameters. If the application significantly exceeds these distances another hub must be added to amplify the optical signals.

9-10.5. Installation

- Install the fiber optic cable in accordance with all local safety codes. Polyurethane and PVC are two possible options for cable materials that might meet local safety codes. See Section 9-10.6. for some examples of specific cables.
- Proper planning is extremely important. Layout for the level of redundancy needed, cable routing distances, proper application of the distance rules, and procurement of excellent quality hubs, UPS systems, fiber cable, and connectors should all be included in planning the Stage Link.
- Install the system so that it will be strong enough for indoor and outdoor applications, including direct burial.
- Strictly adhere to the manufacturer's recommendations on the minimum bend radius and maximum pulling force.
- Test the installed fiber to measure the losses caused by the cable and the connectors. A substantial measured power margin is the best proof of a high quality installation.
- The process of attaching the fiber connectors involves stripping the buffering from the fiber, inserting the end through the connector, and casting it into an epoxy or other plastic. This typically involves using a kit designed for the particular connector system. After the epoxy has hardened, the end of the fiber must be cut off, ground, and polished.
- The fiber hubs need reliable power, and should be placed in a location that will minimize the amount of movement they must endure, yet keep them accessible for maintenance.

9-10.6. Specifications

The following sections provide specification information for Four-Fiber Cable with/without Armor, the Fiber Optic Hub, and Fiber Optic connectors.

9-10.6.1. FOUR FIBER CABLE WITHOUT ARMOR.

Optical Cable Corporation Part (or its equivalent): `RK920929-A`

Fiber and buffering:		
Fiber Type:	Multimode	
Core diameter:	62.5 microns	
Cladding Diameter:	125 microns	
Fiber proof test:	100 kpsi	
Coating Diameter:	500 microns	
Tight buffer diameter:	900 microns	
Tight buffer material:	Hard elastomeric; plastic not acceptable.	
Numerical aperture:	0.275	
Attenuation & Bandwidth	**Attenuation**	**Bandwidth**
850 nm	3.5 dB/km	160 MhzKm
1300 nm	1.3 dB/km	500 MhzKm
Stripping ability:	All layers can be easily removed with commercially available tools.	
Sub Cables:	Four sub cables each with one fiber	
Fiber strength member:	Aramid yarn	
Sub-cable diameter:	2.5 + - 0.125 mm	
Sub-cable jacket:	Elastomeric	
Color coded:	Standard -- blue, orange, green, and brown	
Cable construction:	Sub-cables with filler/strength member	
Jacket:	Tight bound pressure extruded Flame retardant polyurethane	
Color:	Black	
Cable weight:	65 kg/km	
Cable diameter:	8.0 mm	
Strength members:	Aramid yarn with individual precise tensioning	
Conductivity:	No electrical conductors may be used.	
Installation:		
Min bend radius:	16 cm (when pulling)	
Max tensile load:	2200 N	
Location:	Aerial, direct burial, or duct	
Pulling:	Ordinary cable grips	

Operating:

Min bend radius:	8 cm
Max tensile load:	550 N
Temperature:	-40❍C to +85❍C
Immersion:	No damage
Storage:	-55❍C to +85❍C

Test specification:	EIA-STD-RS-455 (or equivalent):
Impact resistance:	1500 impacts
Crush resistance:	2200 N/cm
Cyclic flexing:	2000 cycles

9-10.6.2. FOUR FIBER CABLE WITH ARMOR.

Optical Cable Corporation Part (or equivalent): `RK920929-A-CST`

Comprised of the same cable as described above, but surrounded with steel tape and polyethylene over jacket. Recommended vendor: Optical Cable Corporation, 5290 Concourse Drive, Roanoke VA 24019 (shipping); PO Box 11967, Roanoke VA 24022-1967 (mailing); 1 (800) 622-7711 or 1 (540) 265-0690.

Fiber and buffering:

Fiber Type:	Multimode
Core diameter:	62.5 microns
Cladding Diameter:	125 microns
Fiber proof test:	100 kpsi
Coating Diameter:	500 microns
Tight buffer diameter:	900 microns
Tight buffer material:	Hard elastomeric; plastic not acceptable.

Numerical aperture: 0.275

Attenuation & Bandwidth	**Attenuation**	**Bandwidth**
850 nm	3.5 dB/km	160 MhzKm
1300 nm	1.3 dB/km	500 MhzKm

Stripping ability: All layers easily removed with commercially available tools.

Sub Cables:	Four sub cables each with one fiber
Fiber strength member:	Aramid yarn
Sub-cable diameter:	2.5 +- 0.125 mm
Sub-cable jacket:	Elastomeric
Color coded:	Standard -- blue, orange, green, and brown

Cable construction:	Sub-cables with filler/strength member
Jacket:	Tight bound pressure extruded and flame retardant polyurethane
Color:	Black
Armor:	Steel tape nominal 0.155 mm
Armor overlap:	2 mm, Bonded, corrugations in register.
Over jacket:	Polyethylene 1 to 1.5 mm thick
Cable weight:	174 kg/km
Cable diameter:	13.0 mm
Strength members:	Aramid yarn with individual precise tensioning

Installation:	
Min bend radius:	26 cm (when pulling)
Max tensile load:	2660 N
Location:	Aerial, direct burial, or duct
Pulling:	Ordinary cable grips
Operating:	
Min bend radius:	13 cm
Max tensile load:	532 N
Temperature:	-40❍C to +65❍C
Immersion:	No damage
Storage:	-55❍C to +70❍C
Test specification:	EIA-STD-RS-455:(or equivalent)
Impact resistance:	50 impacts
Crush resistance:	440 N/cm

9-10.6.3. FIBER OPTIC HUB. The Hub Assembly is typically a metal box with a power supply and card slots. The Stage Link application typically calls for two hub assemblies each with one card. They can be ordered for table or flange mounting, and for 120 or 240 volt application. Moving an internal jumper can reconfigure for the other voltage if a mistake has been made.

	TABLE MOUNTED	FLANGE MOUNTED
120 vac 50/60 hz	MODHUB-16 EXP-CXS/FOG-ST	MODHUB-16F EXP-CXS/FOG-ST
240 vac 50/60 hz	MODHUB-16E EXP-CXS/FOG-ST	MODHUB-16EF EXP-CXS/FOG-ST

MODHUB-16 can hold four cards, each with 4 ports for a total of 16 ports. This expansion module has two optical ports and two copper coax ports. Recommended vendor: Contemporary Control Systems Inc., 2512 Wisconsin Avenue, Downers Grove IL 60515, 1 (708) 963-7070.

9-10.6.4. FIBER OPTIC CONNECTORS. *3M Connector model 6100* (or equivalent). This ST bayonet type zirconia connector is already filled with a thermoplastic material that is melted for the insertion of the fiber. The installation kit is model 6150A (or its equivalent). The filler is melted, fiber inserted, end cleaved and polished with only one paper. This connector makes fast and reliable connections and is gaining popularity.

Thomas & Betts Connector model 91810-125-2P (or equivalent). ST connector of composite polymer, glass capillary, crimp and polish termination. Filler is a fast drying epoxy. Assembly Polishing Kit model 91000AKP (or equivalent) includes all parts needed and instructions.

Amphenol Connector model 953-101-5010 (or equivalent). ST connector made of glass reinforced polymeric, ceramic ferrule, copper crimp ferrule, and PVC bend relief boot. Termination kit, model 927-100-5000 (or equivalent), includes stripper, curing oven, microscope, crimp tool, snips, polish board, training video. Add on termination kit has a cleave tool, polishing tool, cable preparation template and instructions. Filler is heat cured epoxy.

9-11. TYPICAL STAGE LINK ADDRESSES

Table 9-1 shows typical Stage Link addresses as assigned by the factory. Any valid two digit hexadecimal number may be used for any Stage Link address except for <G> processor addresses. For for information about using a <G> processor for Ethernet Communications with a DCS system, see Chapter 11.

How to assign a new communication processor, Mark V Trainer, or <I> processor node to an existing Stage Link is described in Chapter 5 of the Maintenance Manual, GEH-5980.

Address	Description	
01	Addresses between 01 and the start of the <G> processor addresses are not reserved.	
02		
.		
.		
.		
0F		
10	8th <G> processor	Reserved for communication with customer's DCS via Ethernet
11	7th <G> processor	
12	6th <G> processor	
13	5th <G> processor	
14	4th <G> processor	
15	3rd <G> processor	
16	2nd <G> processor	
17	1st <G> processor	
18	8th <I> processor	Reserved for <I> processor addresses.
19	7th <I> processor	
1A	6th <I> processor	
1B	5th <I> processor	
1C	4th <I> processor	
1D	3rd <I> processor	
1E	2nd <I> processor	
1F	1st <I> processor	
Addresses between 20 and the start of the Communication Core addresses are not reserved.		
.	.	Reserved for the Mark V Communication Core(s), <C> or <D>.
.	.	
EE	.	
EF	.	
FA	5th Mark V	
FB	4th Mark V	
FC	3rd Mark V	
FD	2nd Mark V	
FE	1st Mark V	

*Address 00 is reserved
*Address FF is not reserved but typically not used

Notes:

CHAPTER 10

MODBUS™ CONFIGURATION

10-1. INTRODUCTION

The Mark V control interface (<I>) acts as a slave to the Distributed Control System (DCS) master. The DCS can thereby request information from each Mark V controller (such as flow rate values) by sending a MODBUS data request message via the MODBUS link to a specific slave address. The DCS master can also issue MODBUS command messages to initiate operator commands to each Mark V. These command messages support both push button commands (such as START, STOP) and analog setpoint commands (such as preselected load setpoint).

The rate at which the data can be collected is limited by the transmission rate on the RS232 link, and by the turn-around times of the computers on each end of the link. The DCS may issue no more than ten operator command messages per second.

The <I> processor slave configured with the appropriate address will reply with the requested data. At sites where the optional multi-unit <I> functionality is implemented (maximum of eight Mark V panels to an <I>), a single <I> can act as multiple slaves responding to multiple slave addresses.

The MODBUS interface uses Gould Modicon's MODBUS Protocol. The following text provides application information only. It does not specify MODBUS Protocol.

Detailed information on the MODBUS protocol is available from Gould Inc.'s Reference Guide PI-MBUS-300 Rev B, January 1985. This is **not** the MODBUS Plus specification. It is recommended that both this document and the Gould reference guide are used together.

The <I> system allows easy addition, deletion, or modification of the data lists by the DCS vendor or customer after equipment installation. The data point capability of the Mark V turbine control system generally far exceeds the actual needs of the DCS.

10-2. EXTERNAL COMMUNICATION LINKS

External communication links allow an operator at a remote location to initiate any operator command by sending a logical command or an analog setpoint to an <I>. Logical commands are used to initiate automated sequences which reside in the Mark V control system, and analog setpoints are used to set a variable, such as the turbine load, to a predefined level.

It is occasionally more efficient to set targets with an analog number, such as megawatts, rather than manually entering raise/lower commands. The operator can send a target setpoint which the Mark V enters into its data base and ramps to the target at a predetermined ramp rate.

Any operator command which can be initiated from the <I> can be initiated from the remote control system through a communication link. This does not preclude the use of hard wired remote control interface. However, a traditional hard wired interface can be I/O intensive and needs to be evaluated on a case-by-case basis. Maintenance commands (editing application software, changing tuning constants, etc.) are generally restricted to being issued from an <I>.

All application data in the Mark V can be monitored by the remote control system via the remote serial communication link. The Mark V data base contains several thousand points, but usually less than 500 points are of interest to the remote control system. These consist of logical data for alarm and event messages and "analog" or numeric data for such variables as speed, load, vibration, temperature, and so forth.

One <I> can interface with up to eight Mark Vs. Therefore, the external communication link to the remote control system has the capability to communicate through the <I> to up to eight Mark Vs. An <I> can be located local, remote, or anywhere within the distance limitations of the Stage Link system (for details on the Stage Link, see Chapter 9). The DCS or remote control system can be located anywhere within the restrictions of the RS232 serial communication link.

10-3. RS232 & MODBUS

GEDS offers MODBUS protocol with an RS232 link as a level #1 communication link because it is compatible with most control systems. The DCS is the master, and it issues a command to the Mark V which is the slave. Logical and analog setpoint commands can be issued from the DCS to preassigned points in <I> to initiate control action. Requests for data can be sent to preassigned points in the <I> and these points can be modified to include any point in the data base.

MODBUS, with an RS232 link, is adequate for most applications, but its limitations must be recognized. The limitations are a 19,200 baud (bit/sec) transmission rate and the lack of individual time tags for alarms and events.

The RS232 link transmission rate makes it satisfactory for one or two units on a single communication link depending on the amount of data which is requested for transmission. A major factor in determining the suitability of the MODBUS and RS232 is whether the historical data base for the Mark V is local such as a Mark V Historian <H> or remote in the DCS. If a Mark V Historian <H> is used, the communication link to the DCS is used to retrieve data from the Mark V. This data is used to support the displays which currently appear on the monitors in the control room. This retrieval requires minimal data transmission from the Mark V.

The communication link load is a function of the amount of data which the DCS is requesting and how often it is to be transmitted. Data in the Mark V requires one word per analog point and one word for each 16 logic points (such as the state of alarms and events). An evaluation of the link load for a specific application requires reviewing the amount and frequency of the data transmission(s) and the specified baud rate and verifying that all is within the desired sampling rate.

In order to minimize the sampling rate for alarms, a contact output is provided from the Mark V to the plant alarm horn system that informs the DCS that a new alarm has occurred. The DCS can then request a transmission of the alarm states data. This eliminates the need to continuously sample the alarm queue at a fast rate. Some sites provide a remote printer to get high resolution local alarm (62ms), event (62ms), and sequence of event logs in the control room without investing in more sophisticated equipment.

10-4. PHYSICAL LINK LAYER/FORMAT

These sections detail the physical link layer/format of the MODBUS serial link.

10-4.1. Link Layer

Each <I> processor can provide one point-to-point MODBUS RS232 serial link to a customer's DCS. Multiple MODBUS serial links from any one <I> are **not** possible. Since the master may communicate with multiple slaves, the <I> will send data upon request only. No periodic transmission of data from the <I> is possible, except as a reply to a periodically transmitted request sent from the DCS.

10-4.2. Physical Layer

The Mark V <I> processor is used as the communications port for MODBUS, typically using the <I>'s COM2 serial communications port. If the COM2 port is not available, as in cases where a touch screen is connected to COM1 and the cursor positioning device (trackball or mouse) is connected to COM2, a serial expander card is required to supply eight more serial ports. The port used is configured with a DB9P (Male) connector as Data Terminal Equipment (DTE). Diagrams showing this connection are shown in Figure 10-2.

The system is asynchronous RS232 compatible, 300 to 19,200 baud, programmable parity. It is compatible with full duplex data sets (i.e. modems). Modem generated signals such as CTS (Clear to Send), CD (Carrier Detect) and RI (Ring Indicator) are **not** required to enable the transmitter. The RTS (Ready to Send) and the DTR (Data Terminal Ready) on the <I> are always equal to "1" when the <I> is powered up. This prohibits any multi-drop DCS configurations, as these signals cannot enable or disable communications. Hardware handshaking (flow control via RTS/CTS signals) is **not** supported. It is up to the master to only request data that it can reasonably expect to handle in a single burst. (See the section below on <I> MODBUS Configuration for timeout configuration.)

GEDS does not recommend remote control of turbine products over a telephone modem due to the inherent unreliability in this type of communication, and due to MODBUS timing errors introduced by error correcting modems. If it is necessary to collect data or operate over telephone modems, the maximum number of bits that can be communicated is ten. Therefore the standard eight bits of MODBUS data and the two bits for start/stop is all that can be accommodated. This means that PARITY is **not** supported over telephone modems.

RS232 Modems are used for distances exceeding 50 feet. RS232 systems require two metallic shielded twisted pair wires to connect the short-haul modems. While modems are supplied by GEDS, the customer must supply the cabling and terminations. The standard modems supplied by GEDS require power on pin four. The modem connected to the <I> obtains its power from the <I> serial port. The other modem must receive power on pin four from the DCS (see Figure 10-3). If the DCS cannot supply power, the **optional** modem set is required and must be specified when ordering from GEDS. This set is powered by an external source.

RS232 transmission distances are defined in Figure 10-1.

RS232 Transmission Distances in Miles with Modems				
Baud	26 Gauge	24 Gauge	22 Gauge	19 Gauge
300	10.0	12.0	15.0	25.0
1200	6.0	7.5	9.0	15.0
2400	4.5	5.5	7.5	11.0
4800	3.5	4.5	5.5	7.0
9600	2.2	3.2	4.0	5.0
19,200	1.0	1.2	1.5	2.0

Figure 10-1. RS232 Transmission Distances

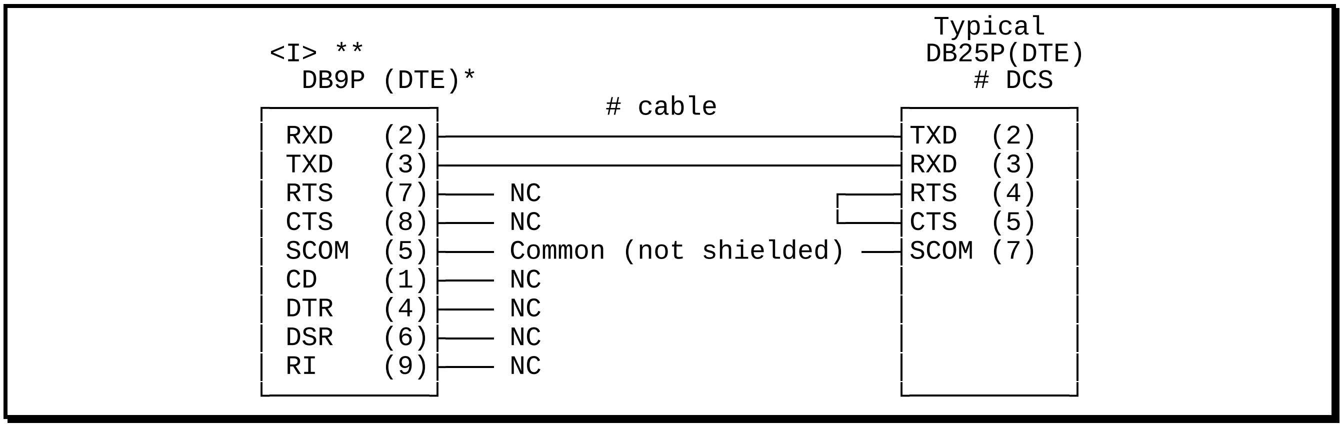


Figure 10-2. RS232 connection to DB9 serial ports with no modems

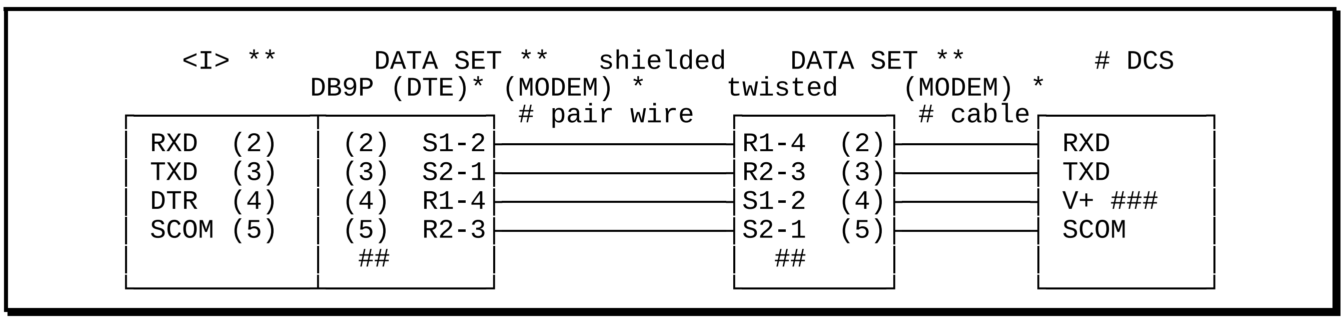


Figure 10-3. RS232 connection to DB9 Serial port with modems

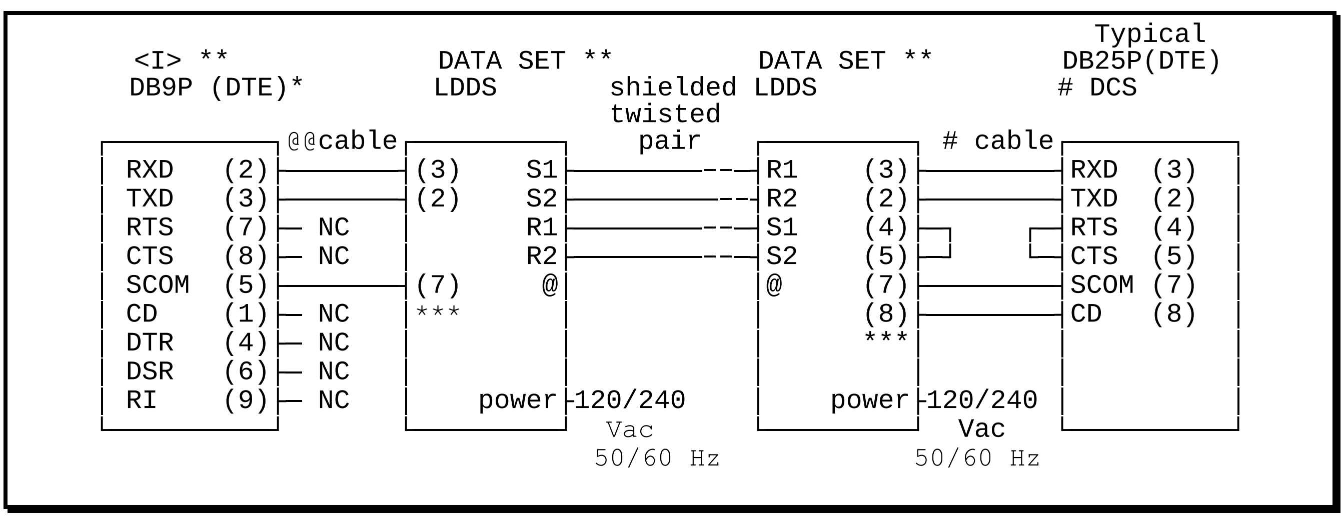


Figure 10-4. Optional RS232 connection with LDDS

*	9 pin D-type connector DB9P (male/<I>, female/cable)
**	GEDS supplied
***	25 pin D-type connector (i.e., AMP 205208-1) (male/cable, female/modem)
NC	No Connection
#	Customer supplied
##	Short-haul modems
###	3-5 mA @ 10 V DC. Nominal 12 V DC, minimum 6 V DC, maximum 25 V DC (not recommended due to lack of protection).
LDDS	Limited distance data set (short haul modem)
@	Terminal board connection
@@	GE supplied cable - <I> (DB9P female) to Data Set (DB25 male)

10-5. <I> MODBUS CONFIGURATION

This section defines the required additions and/or modifications to configuration files on an <I> to accomplish the following:

- enable and configure the MODBUS task on the <I>
- enable and configure a serial port for MODBUS communication
- set up the MODBUS mapping tables
- generate the related document for the DCS vendor

10-5.1. F:\CONFIG.DAT: MODBUS Enable and Configuration

Add the following line under the "`OPTIONS`" heading in the <I> system configuration file `F:\CONFIG.DAT` to enable the MODBUS programs on the <I> processor (add the heading if not supplied):

```
OPTIONS
    MODBUS = YES
```

Any option listed under the header must begin with at least one space or tab in the left-most column.

The MODBUS link out of the <I> supports a TIMEOUT value. This is the amount of time that the <I> processor is willing to wait for the Mark V to return the data before it gives up and sends a NAK back to the master. The default timeout time is 4.75 seconds. **The timeout value in the <I> processor MUST be shorter than the timeout value in the master.** If it is not, the master will timeout first and send a second request. In this case, the <I> processor, which had not yet timed out on the first message, may then send the first reply back. The master may interpret the reply to the first message as the reply to the second message and thus process the data incorrectly. If a timeout of 4.75 seconds is not desired, the timeout time may be changed by using another option under the `OPTIONS` header (see above). The format of this option is:

```
    MODBUS_TIMEOUT = time
```

where `time` is in seconds and fractional seconds n.nn. There must be a digit before the decimal point. Timeout values of less than 1.25 seconds are not recommended.

When the timeout period has expired, the <I> will send a NAK code back to the master. The default NAK code is 06, which is listed as DEVICE BUSY. The <I> can issue a NAK code of 04, which is listed as FAILURE IN DRIVE (see the section on Message Errors, below). This can be done by using another option under the `OPTIONS` header (see above). The format of this option is:

```
    MODBUS_TIMEOUT_NAK = NAK4
```

The only valid option values are `NAK4` and `NAK6.` Any other value will cause it to revert to the default of `NAK6.` A message is given in the file `G:\LOG\STARTUP.LOG` if value is unknown.

10-5.2. F:\IO_PORTS.DAT: MODBUS Link Definition

The <I> uses information in the <I> port definition/configuration file, `F:\IO_PORTS.DAT`, to assign functions to its available I/O ports. These ports include the parallel port (LPT1) and serial port (COM2), plus, the serial ports on an expander card (if supplied). A serial port is configured here as a MODBUS link. The <I> responds to commands sent from the DCS based on the slave address(es) set up in this file, and decides what Mark V unit is to supply data for that address. The format of the transmitted data (the mode) is also set here.

To customize `F:\IO_PORTS.DAT`, use the standard text file editor. `F:\IO_PORTS.DAT` contains textual information to help define a MODBUS link.

First define a serial port by adding the keyword "MODBUS" to the port definition. For example, to use COM2 as a MODBUS port running at ninety-six hundred (9600) baud with no parity, add the line:

```
S2  BAUD 9600  PARITY NONE  DATA 8  STOP 1  IRQ 3  BASE_PORT 2F8  MODBUS
```

The port may be S2, or D1 through D8 if an expander card is supplied. `F:\IO_PORTS.DAT` contains a list of the key words and defaults available for defining the physical ports (see that file for additional details). Do not use the port S1, as it is **always** used for the cursor positioning device (configured in `C:\AUTOEXEC.BAT`). Do not use the S2 port if a touch screen and/or track ball or mouse are supplied as the track ball or mouse will be configured on COM2. An expander card is required in this situation. Only one port on an <I> may be defined for MODBUS at one time.

When the port has been defined, add lines to define the slave addresses, the Mark V panel that will supply the data to each address, and the format of the communicated data. In the example below, the <I> acts as both slave 1 and 2. When the DCS issues a command message to slave 1, the <I> communicates with Mark V unit "T1". When the DCS issues a command message to slave 2, the <I> communicates with Mark V unit "T2". Both use Native Mark V data formats.

```
MODBUS  PORT S2  SLAVE 1  UNIT T1  MODE NATIVE
MODBUS  PORT S2  SLAVE 2  UNIT T2  MODE NATIVE
```

port n specifies the physical port enabled above.

slave n specifies the <I>'s MODBUS address in decimal in the range (1-255).

unit <name> indicates the Mark V panel that is associated to this slave address when replying to a MODBUS request. Refer to the <I> file `F:\CONFIG.DAT` for available unit names.

mode <function> controls the format of the transmitted MODBUS data. It is one of NATIVE, SIGN16, UNS16, or UNS12.

The <I> must be restarted (rebooted) for changes in `IO_PORTS.DAT` to take effect.

10-5.3. Holding Coils, Input Coils, Holding Registers, Input Registers

Each Mark V <I> that supplies MODBUS data must have a MODBUS Mapping file for each Mark V panel. This file contains mapping tables to associate the registers and coils with Mark V control signal database (CDB) pointnames.

There are **two** categories of point mapping tables. They are, *holding* and *input*. The MODBUS protocol supports reading and writing of holding table points, and reading but **not** writing of input table points.

The <I> does not make any restrictions on what Mark V CDB pointnames can be in input and holding tables. The <I> knows on a point by point basis which points are Mark V commands and which points are not. A Mark V command is one of the following:

```
LOGICS:  pushbuttons    logic state variables.
ANALOGS: analog setpoints  enumerated state variables
```

As such, the holding tables may be used exclusively, and the input tables may be omitted completely. Sending a MODBUS write command to a point on a holding table that is not mapped to a Mark V command will have no effect and will be ignored.

There are two classes of signals found on the two tables: coils and registers. A coil is one bit, and maps to a logic Mark V CDB pointname, with a value of zero or one. A register is 16 bits, and maps to an analog Mark V CDB pointname, or to a set of (up to) 16 packed logicals.

Multiple holding coils or registers may be mapped to any one Mark V CDB pointname. If the pointname is a Mark V command, a write command from the DCS to any of the mapped holding coils or registers will cause the <I> to execute the Mark V command.

Combining the two categories of tables and the two classes of points yields four separate tables. Each table is limited to 2000 entries, numbered 1 to 2000. The table types are:

`HC` - *holding coils* one bit logic signals that are both readable and writable. Logic Mark V commands **must** be in this table for the <I> to issue the command to the Mark V; however, not all points in this table must be logic Mark V commands. Logic Mark V commands are **not** writable from the holding register table.

`IC` - *input coils* one bit logic signals that are readable but not writable. Any Mark V logic control database signal can be in this table, but logic Mark V commands will **not** be writable from this table.

`HR` - *holding registers* 16 bit analog signals that are both readable and writable. Analog Mark V commands **must** be in this table for the <I> to send the command to the Mark V, but not all points in this table must be analog Mark V commands. They can be any analog control database signal, or a set of (up to) 16 bit packed logic control database signals. Mark V logic commands can be packed into a holding register, however, they will **not** be writable in that format. For the <I> to issue a signal to a **logic** Mark V command, it must be in the holding coil table. Simply put: **packed holding registers are not writable.**

`IR` - *input registers* 16 bit analog values that are readable but not writable. Any analog control database signal, or a set of (up to) sixteen (16) bit packed logic control database signals, may be in this table. Analog or logic Mark V commands will **not** be writable from this table.

10-5.4. @SPARE: Unused Coils and Registers

If a request for any read or write command from the DCS using a coil or register not mapped to a Mark V CDB pointname, the <I> will respond with a `NAK` code of 02, which is listed as INVALID ADDRESS (see the section on Message Errors, below). This prevents the DCS from interpreting the reply from the <I> as data from the Mark V.

To define coils or registers without mapping them to Mark V CDB pointnames, map them to @SPARE in the mapping table. This tells the <I> that the Coil or register is not used, but is legally defined. The <I> will respond to any DCS request with a data value of zero (0) for all coils and registers defined as "@SPARE". It is not necessary to map **all** the points in the mapping table. The "@SPARE" is used only to fill in the gaps or to format the table as required by the DCS.

Bit packed registers (logic control database signals packed into a 16 bit register) should **not** have unused bits defined as "@SPARE". When any one bit of a register is defined as a packed bit, the entire register is defined as a packed register. Therefore, omit defining the unused bits. Bits in a packed register not mapped to a logic Mark V CDB pointnames will always send a value of 0 to the DCS. "@SPARE" can only be used on an **unpacked** register when the entire register is unused.

10-5.5. F:\UNITN\MODBUS.DAT: MODBUS Mapping File Format

The following section contains a simple template as an example for filling out the MODBUS mapping file F:\UNITn\MODBUS.DAT. Each line of the mapping file has three required entries:

`Table_Type   Table_Point   CDB_Pointname`

where `Table_Type` is holding or input, `Table_Point` is 1 through 2000, and `CDB_Pointname` is a valid Mark V control signal database pointname. The `Table_Point` 1 is requested from the DCS as point 0. Bit packed registers are defined with a digit after a decimal point in the register mapping. The 16 bits are numbered [0-15] (see Figure 10-5).

When adding or modifying entries in this file, the following rules and guidelines apply:

- It is **not** necessary to define all the table points in this file, only the ones that are desired.
- Never define a bit of a packed holding register (such as HR0001.1) as @SPARE. Defining just one of the bits as packed defines the entire register as a packed register. A 0 is returned for bits not mapped.
- When a semi-colon is found in a line, the rest of the line is considered a comment.
- Do not add scaling information or definitions in this file, only the required three fields.
- Never add page breaks (form feeds) in this file.
- The Table and Location may be joined or separated, such as HC 0001, or HC0001, and the Location may be expressed as 0001 or 01 or 1, etc.
- The Table entries may be in any order.
- Do not map any one register to both packed Mark V CDB logic pointnames and to an analog Mark V CDB pointname.
- Do not have any one coil or register mapped more that once.
- Group similar Mark V CDB points together, such as temperatures, speed signals, or Mark V command feedback signals.

```
; HOLDING_COIL TABLE
HC  0001  logic_signal_name
HC  0002  logic_signal_name
HC  0003  @SPARE
; INPUT_COIL TABLE [Optional]
IC  0101  logic_signal_name
IC  0102  logic_signal_name
IC  0103  @SPARE
; HOLDING_REGISTER TABLE
HR  0001    analog_signal_name
HR  0002    @SPARE
HR  0004.0  logic_signal_name
HR  0004.1  logic_signal_name
HR  0003    analog_signal_name
; INPUT_REGISTER TABLE [Optional]
IR  0101    analog_signal_name
IR  0102    @SPARE
IR  0004.4  logic_signal_name
IR  0004.5  logic_signal_name
```

Figure 10-5. F:\UNITn\MODBUS.DAT

From the example in Figure 10-5, it can be seen that not all table points are defined or in order, comments are added using a semi-colon (;), there are no duplicate table point definitions, etc. However, the rules outlined above have been followed and the file can therefore be considered correct.

The <I> must be restarted for changes to take effect. After the <I> has restarted, inspect the file `G:\LOG\STARTUP.LOG` for any errors or warnings (search for "MODBUS").

10-5.6. MODBUS_L.EXE: MODBUS Listing Program

After F:\UNITn\MODBUS.DAT is completed, the document F:\UNITn\MODBUS.LST is created for use by the DCS vendor for setting up the DCS. This file documents the MODBUS tables.

The MODBUS Listing Program, G:\EXEC\MODBUS_L.EXE, reads the necessary configuration files on the <I> and creates F:\UNITn\MODBUS.LST. This file contains a list of the coil and register maps, complete with scaling information. A new listing file must be created each time F:\UNITn\MODBUS.DAT is changed, or when the mode defined for the MODBUS link is changed in F:\IO_PORTS.DAT. This MODBUS listing file is the key to programming the DCS end of the MODBUS link.

The MODBUS listing file defines the format and scaling of each mapped coil and register. Included in this list is an indication of which signals are Mark V commands. To create a new listing file F:\UNITn\MODBUS.LST, type MODBUS_L from the DOS line in the unit specific directory F:\UNITn.

The format is shown by typing MODBUS_L HELP which returns printout shown in Figure 10-6.

```
MODBUS_L:  MODBUS LISTING PROGRAM

Command format:  MODBUS_L  [options]
    options:
       ENGLISH, METRIC, CUSTOM, HARDWARE
           These options control the scale code set used to supply the gain,
           offset, and engineering units for each point.

       NATIVE, SIGN16, UNS16, UNS12
           These options control the formatting of the raw data, with the
           appropriate changes to the gain and offset for each point.

       QUIET
           This option disables the printout to the terminal of messages
           indicating that a point name was not found in the dictionary.
           The messages, however, will be indicated in the MODBUS.LST file.

       NOLONG
           This option produces a MODBUS.LST file without "longnames".

       LONG
           This option makes a MODBUS.LST with a nonlogic longname field of
           66 characters. The nonlogic longname field default is 40 char.
```

Figure 10-6. MODBUS_L Command Format

The options in Figure 10-6 are used to modify the format of F:\UNITn\MODBUS.LST as follows:

scaling (ENGLISH, METRIC, CUSTOM, HARDWARE) determines which F:\UNITn*.SCA file is used (ENGLISH.SCA is the default file).

mode (NATIVE, SIGN16, UNS16, UNS12) used must match the mode defined in F:\IO_PORTS.DAT. MODBUS_L does not read this file, it must be added manually.

quiet if used, F:\UNITn\MODBUS.LST should be checked for error messages, as the errors will not be displayed on the screen.

nolong omits longnames (Mark V control signal database pointname definitions) defined in the file `F:\UNITn\LONGNAME.DAT`.

long is used to widen `F:\UNITn\MODBUS.LST` to avoid trimming longnames of nonlogic (analog) signals.

`MODBUS_L` will flag duplicate coil or register assignments, undefined Mark V CDB pointnames, invalid formats, invalid coil or register numbers, conversion errors, etc. **Resolve all errors before communicating with the DCS.**

10-5.7. F:\UNITn\MODBUS.LST: MODBUS Listing File

An example `F:\UNITn\MODBUS.LST` file is shown in Figure 10-7.

```
+HR 0001    | SWREF_CMD   | SIGN16 |    512 | 0 | MW    |
+HR 0002    | DRVAR_CMD   | SIGN16 |    512 | 0 | MVAR  |
+HR 0003    | SC43LOAD    |  UNS16 |  65536 | 0 | STATE |
 HR 0004    | @SPARE      |  spare |
 HR 0005    | @SPARE      |  spare |
+HR 0006    | L90PSEL_CMD | SIGN16 |    512 | 0 | MW    | PRESELECTED LOAD ANALOG SETPOINT
+HR 0009    | SC43        |  UNS16 |  65536 | 0 | STATE | TURBINE CONTROL SELECTION
 HR 00010   | SS43        |  UNS16 |  65536 | 0 | STATE | TURBINE COMMAND STATE SELECTION
 HR 0200.00| L52GX        | PACKED | GENERATOR BREAKER CLOSURE
 HR 0200.01| L94X         | PACKED | NORMAL SHUTDOWN
 HR 0200.02| L30D_SD      | PACKED | NORMAL DISPLAY MESSAGE "SHUTDOWN STATUS"
 HR 0200.03| L30D_SU      | PACKED | NORMAL DISPLAY MESSAGE "STARTUP STATUS"
+HC 0001    | L1FAST_CPB  |  LOGIC | FAST LOAD START SIGNAL
+HC 0002    | L1START_CPB |  LOGIC | START SIGNAL
+HC 0003    | L70R4R_CPB  |  LOGIC | SPEED SETPOINT RAISE COMMAND PUSHBUTTON
+HC 0004    | L70R4L_CPB  |  LOGIC | SPEED SETPOINT LOWER COMMAND PUSHBUTTON
```

Figure 10-7. F:\UNITn\MODBUS.LST

Each line in Figure 10-7, `F:\UNITn\MODBUS.LST` contains the following information in this order:

`(Write Status) Table Point | CDB Pointname | Mode | Gain | Offset | Units | Longname`

Table Point indicates which one of the four Mapping Tables and what point on the table is used for this point.

(Write Status) A preceding plus sign "+" indicates that this is mapped to a Mark V command. In this case it is legal for the DCS to request a write to the point, and thus the <I> issues the command to the Mark V. The listing in Figure 10-7 shows holding register 1 as an analog command, holding coil 1 as a logic command, and holding register 9 as a enumerated state variable that can be written to the Mark V. Logic Mark V commands can not be written if packed into holding registers (see above section on holding coils and holding registers). In this example, a 0 may select "OFF", 1 may select "CRANK", 2 may select "AUTO", etc. Enumerated state variables are discussed in Chapter 5.

CDB Pointname Mark V control signal database pointname that the coil or register maps.

Mode transmission mode that the data is sent in over the MODBUS link.

Gain required to convert the MODBUS value of registers to engineering units.

Offset required to convert the MODBUS value of registers to engineering units.

Engineering Units displayed units associated with the above MODBUS Gain and Offset for registers. A entry of "HEX" indicates that this point is not normally converted into engineering units, but displayed in hexadecimal. Special points, such as bit masks, are defined this way.

Longname description of the signal found in F:\UNITn\LONGNAME.DAT.

Once `F:\UNITn\MODBUS.LST` is generated, do not edit it as running MODBUS_L overwrites any changes. It may be printed using the DOS `PRINT` command or the `IDOS_SPL` command. Do not use the DOS `COPY` command to copy any file to the desired printer port. This takes the port away from the IDOS system, and the <I> must be reset to re-enable it.

10-6. MODBUS DATA FORMAT AND SCALING

10-6.1. MODBUS Data Conversions: Logics

Logic control database signal values can be transmitted to the DCS using either coils or registers. Coil data conversion corresponds to a single logic control database signal with the LSB indicating its state.

When registers are used to transmit logic control database signal values, each bit corresponds to the state of a logic control database signal. A register having 16 bits corresponds to a group of (up to) 16 logicals with the least significant bit being the first logic CDB signal in the group, and the MSbit as the last logic CDB signal in the group. The two bytes are packed into the MODBUS message using the normal rules for MSB and LSB order. The values of undefined bits in a packed register are transmitted as 0. See section 10-6.2 for details on the modes used in transmission of data, as some modes may not be suitable for bit packed registers.

10-6.2. MODBUS Data Conversions: Analogs

Analog control database signal values can be transmitted using registers only. Since DCS vendors handle analog signals in different ways, such as 12 bit positive numbers, the <I> may be configured to transmit analog data in one of several modes. The following modes are currently supported to meet customer needs. This section defines how to convert each data type to engineering units. The MODBUS listing file (see above) defines the data type for each signal.

The conversion algorithms presented here define the conversion process at the most understandable level. The implementation of these algorithms do not always follow these algorithms directly as there are many different schemes used to prevent mathematical overflows in computations. The choice of the final implementation is left to the DCS vendor.

10-6.2.1. NATIVE. If a point is of data type NATIVE, all analog signals are transmitted in their internal Mark V representation. This is the recommended mode because the signals can be transmitted directly without any manipulation which could result in decreased resolution. The most common signal types used in the Mark V are as follows:

L1 logic signal, least significant bit indicating state zero or one.

F2 16 bit signed two's complement representation, range: [-32767,+32767].

C2 16 bit unsigned representation, range: [0,+65535].

H2 16 bit unsigned representation used for bit masks.

X2 16 bit unsigned representation used to form the lower 16 bits of long, 32 bit values.

10-6.2.2. UNS12. If a point is of data type UNS12, it is an unsigned 12 bit number. The function of this mode is to restrict signals to a range of [0,+4095]. The MODBUS listing file's MODBUS Gain and Offset define the conversion into engineering units, where the conversion is:

Engineering units = (Raw_value) / 4096 * GAIN + OFFSET

For example, if the register value of a thermocouple is requested, the listing file shows that the MODBUS Gain is 4096 and the Offset is -2048 to get to DEG F. The conversion of the register value of 2118 to engineering units is:

Engineering units = (2118) / 4096 * 4096 - 2048
Engineering units = 70 DEG F

The conversion from SIGN16 to UNS12 causes a loss in resolution of 16:1 and should be avoided.

Certain internal signal types (e.g. H2, X2) become meaningless if restricted to the UNS12 range and would therefore be transmitted directly without conversion (if requested by the DCS). Logics packed into registers also become meaningless if the register's value is converted. The MODBUS listing file will list these points with a mode of UNS16. Fortunately most analog signals that are of interest to the DCS are in an F2 representation. A gain of 1/16th is applied to F2 signals and 2048 is added to convert the range from [-32767,+32767] to [0,+4095]. The unsigned 16 bit signal type C2 is transmitted Modulo 4096 to also restrict the range to [0,+4095].

10-6.2.3. HW12. A special version of the UNS12 mode is available for use with some models of DCS systems. These systems collect data using the normal UNS12 mode described above, but are unable to send analog setpoints to the Mark V using the same GAIN and OFFSET conversion. In these DCS systems, no OFFSET is allowed in the conversion of analog setpoint commands to MODBUS register values, only a GAIN.

To accommodate these systems, the HW12 mode reports data to the DCS using UNS12 mode, but uses a different formula for analog setpoint commands sent from the DCS to the Mark V. This scheme allows the DCS the full positive range of the Mark V command, but does prevent sending negative values. Analog setpoint commands in these systems are usually scaled in percentages.

Make sure that the programming in the DCS reflects the fact that the scaling of the analog setpoint command to the Mark V is different than the scaling of the feedback signal from the Mark V. (The HW12 mode analog setpoint command is the only place where this is true.)

For data from the Mark V to the DCS, the conversion is the same as the UNS12 mode:

Engineering units = (Raw_value) / 4096 * GAIN + OFFSET

For setpoint commands from the DCS to the Mark V, the conversion is:

Raw_command_value = (Engineering units / GAIN) * 8192

Example: If a value position is requested, the listing file shows that the MODBUS Gain is 256 and the Offset is -128 to get percent. The conversion of the register value of 2656 to engineering units is :

Engineering units = (2656) / 4096 * 256 - 128
Engineering units = 38%

To send the analog setpoint command to set the valve to this same position of 38%, the analog setpoint command value that must be sent is:

Raw_command_value = (38) / 256 * 8192
Raw_command_value = 1216

10-6.2.4. UNS16. If a point is of data type UNS16, it is an unsigned 16 bit number. This mode transmits F2 analog signals as unsigned 16 bit numbers by adding 32768. The most significant bit therefore turns from a sign bit into a bit with a weight of 32768. This transforms the range of F2 signals to [0,+65535]. The MODBUS listing file defines the MODBUS Gain and Offset required to calculate the value into engineering units, with the conversion being:

Engineering units = (Raw_value) / 65536 * GAIN + OFFSET

Example 1: If an UNS16 is used for a normal counter, the raw value is the value of the counter. If a counter has the value of 418, it is converted as:

Engineering units = (418) / 65536 * 65536 + 0
Engineering units = 418

Example 2: A number that is normally a signed number in the Mark V is requested as an unsigned number. If the register value of a thermocouple was requested, it shows that in UNS16 mode the MODBUS Gain is 4096, and the Offset is -2048 for DEG F. The conversion of the raw data value of 33888 to engineering units is:

Engineering units = (33888) / 65536 * 4096 - 2048
Engineering units= 70 DEG F

No loss of resolution is caused by converting a SIGN16 to an UNS16.

10-6.2.5. SIGN16. If a point is of data type SIGN16, it is a signed 16 bit number. In this mode, the most significant bit of the 16 bit analog signal is treated as a sign bit, where a one is used to for negative values. C2 type signals are transmitted Modulo 32768 to prevent the sign bit from being set. H2 and X2 types are transmitted directly with an assumed value of (-32768 to +32767). The convention of assuming the raw value is a fraction from -1.0 to +1.0 was used in defining the MODBUS Gain and Offset. This allows the Gain to indicate the maximum (and minimum) value when the Offset is zero. The MODBUS listing file defines the MODBUS Gain and Offset required to calculate the value into engineering units, with the conversion being:

Engineering units = (Raw_value) / 32768 * GAIN + OFFSET

For example, Thermocouples are scaled with a gain of 2048 and an offset of 0 for conversion to DEG F. If the register value is 1120, the conversion to engineering units is:

Engineering units = (1120) / 32768 * 2048 + 0
Engineering units = 70 DEG F

10-7. MODBUS COMMAND AND RESPONSE DEFINITION

10-7.1. Introduction

This section describes the commands and responses supported by the <I> to implement the MODBUS communications and control functions needed. The <I> will be a slave station on an RS232 data link and will respond to commands from the master. Messages are transmitted and received using the RTU transmission mode, which is one of two MODBUS transmission modes, where RTU transmits data in eight bit bytes. The other mode, where characters are transmitted in ASCII, is **not** supported. The RTU transmission mode uses the format in Figure 10-9 (where slave address, function code, `CRC-16 MSB, CRC-16 LSB` are all bytes.

SLAVE ADDRESS	FUNCTION CODE		...DATA...	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-9. RTU Transmission Mode

slave address number from 0 to 255 and specifies the unit with which to communicate (a slave address of zero is considered a broadcast message to all slaves).

function code specifies the purpose and format of the remaining message portion.

CRC-16 bytes are two bytes that complete each and every MODBUS message. The abbreviation *CRC* stands for Cycle Redundancy Check, *MSB* stands for Most Significant Byte, and *LSB* stands for Least Significant Byte. These bytes are used for error checking and are calculated for each transmitted and received message to insure that no transmission error has occurred while the message was in transit. The method for calculating the CRC-16 is a public protocol and is described in numerous text books and the Gould Modicon MODBUS Protocol Reference Guide. Please refer to these other documents for information on calculation of a correct CRC.

10-7.2. Message Errors

When a message is received that cannot be acted upon, the message is either ignored and the <I> waits for the next message or it is responded to with an exception message.

Any messages that are misunderstood, incomplete, or altered in some manner as indicated by framing error, parity error, or CRC-16 error are always ignored due to the fact that it is not possible to reliably determine for whom the message was intended. It is the master's responsibility to detect this timeout condition and resend the message as necessary (see the section <I> MODBUS Configuration above).

Any time a message receipt is in progress and an interval of time corresponding to 3.5 character times (based on the baud rate) occurs without receipt of a character, the message receipt in progress is aborted and ignored. Message failures due to other causes are responded to with an exception response if no reception error has occurred and if the message was not a broadcast (slave address was zero).

The exception code responses that are supported when a normal response is impossible is shown in Figure 10-10 (see the section above on <I> MODBUS Configuration concerning exception codes 04 and 06).

Exception Code	Name	Meaning
01	Illegal Function	The message function received is not supported.
02	Illegal Data Address	The address referenced in the data field is not in a permissible range.
03	Illegal Data Value	The value transmitted in the data field is illegal.
04	Failure in Associated Device	The information requested cannot be provided due to a communication failure with the associated unit as specified by the slave address.
06	Device Busy	The information requested cannot be provided due to a communication time out with the associated unit as specified by the slave address.

Figure 10-10. Exception Code Responses

Format of exception message reply from the slave is shown in Figure 10-11.

SLAVE ADDRESS	FUNCTION CODE	EXCEPTION CODE	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-11. Exception Code Response Format

slave address must be in the range [1-255]. Zero is not allowed as a reply, as it is only used in broadcast messages from the master.

function code is always equal to the master's function code **with the most significant bit set**. Therefore, an exception response sent back to a master that has sent a message with function code 02 hex would have a function code of 82 hex (or 130 decimal) in the exception reply.

exception code as in Figure 10-10, where only codes 01 through 04 and 06 are supported.

10-7.3. Function Code Details

The function codes supported by the <I> included in the messages sent from the DCS are listed in Figure 10-12. Six of the eight function codes implemented in the <I> are used to read from and/or write to the four table types. See the above sections regarding mapping tables. Each function code and the <I>'s replies are described in sections 10-7.3.1. through 10-7.3.8.

Function Code-hex	Description
01	Read Holding Coils
02	Read Input Coils
03	Read Holding Registers
04	Read Input Registers
05	Force (Write) Single Holding Coil
06	Preset (Write) Single Holding Register
07	Read Exception Status
08	Loopback Maintenance

Figure 10-12. DCS Function Codes

10-7.3.1. FUNCTION CODE 01: READ HOLDING COILS. Function code 01 is used to read the holding coil table. The format of a message from the master is shown in Figure 10-13.

SLAVE ADDRESS	FUNCTION CODE 01	START COIL # (MSB)	START COIL # (LSB)	NUMB COILS (MSB)	NUMB COILS (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-13. Function Code 01 Message Format

slave address must be in the range [1-255], zero is not allowed.

starting holding coil number two bytes in length and may be any value less than the highest holding coil number available in the holding coil table. The starting holding coil number is equal to one less than the number of the first holding coil returned in the normal response to this request, i.e. to get the first holding coil, holding coil number 1, enter 0 for the starting holding coil number. The high order byte of the starting holding coil number field is sent as the first byte. The low order byte is sent next.

number of holding coils value two bytes in length and must be in the range from 1 to 2000 inclusive. It specifies the number of holding coils returned in the normal response. The sum of the starting holding coil value and the number of holding coils value must be less than or equal to the highest holding coil number available in the holding coil table. The high order byte of the number of holding coils field is sent as the first byte. The low order byte is sent next.

Format of normal message reply from the slave is shown in Figure 10-14.

SLAVE ADDRESS	FUNCTION CODE 01	BYTE COUNT	DATA BYTE 1	DATA BYTE 2	... DATA BYTE n	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-14. Function Code 01 Reply Format

byte count is a binary number from 1 to 250; the specified number of data bytes follow.

data field is packed holding coil status data. Each byte contains eight holding coil values. The LSbit of the first byte contains the value of the holding coil whose number is equal to the starting holding coil number plus one. The value of the holding coils are ordered by number starting with the LSbit of the first byte of the data field and ending with the MSbit of the last byte of the data field. If the number of the holding coils is not a multiple of eight, then the last data byte contains zeros in one to seven of its highest order bits.

10-7.3.2. FUNCTION CODE 02: READ INPUT COILS. Function code 02 is used to read the input coil table. The format of a message from the master is shown in Figure 10-15.

SLAVE ADDRESS	FUNCTION CODE 02	START COIL # (MSB)	START COIL # (LSB)	NUMB COILS (MSB)	NUMB COILS (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-15. Function Code 02 Message Format

slave address must be in the range [1-255], zero is not allowed.

starting input coil number two bytes in length and may be any value less than the highest input coil available in the input coil table. The starting input coil number is equal to one less than the number of the first input coil returned in the normal response to this request, i.e. to get the first input coil, input coil number one, enter zero for the starting input coil number. The high order byte of the starting input coil field is sent as the first byte. The low order byte is sent next.

number of input coils value two bytes in length and must be in the range from 1 to 2000 inclusive. It specifies the number of input coils returned in the normal response. The sum of the starting input coil value and the number of input coils value must be less than or equal to the highest input coil available in the input coil table. The high order byte of the number of input coils field is sent as the first byte. The low order byte is sent next.

Format of normal message reply from the slave is shown in Figure 10-16.

SLAVE ADDRESS	FUNCTION CODE 02	BYTE COUNT	DATA BYTE 1	DATA BYTE 2	... DATA BYTE n	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-16. Function Code 02 Reply Format

byte count binary number from 1 to 250; the specified number of data bytes follow.

data field is packed input coil status data. Each byte contains eight input coil values. The LSbit of the first byte contains the value of the input coil whose number is equal to the starting input coil plus one. The value of the inputs are ordered by number starting with the LSbit of the first byte of the data field and ending with the MSbit of the last byte of the data field. If the number of the input coils is not a multiple of eight, then the last data byte contains zeros in one to seven of its highest order bits.

10-7.3.3. FUNCTION CODE 03: READ HOLDING REGISTERS. Function code 03 is used to read holding registers. The format of a message from the master is shown in Figure 10-17.

SLAVE ADDRESS	FUNCTION CODE 03	START REG # (MSB)	START REG # (LSB)	NUMB REGS (MSB)	NUMB REGS (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-17. Function Code 03 Message Format

slave address must be in the range [1-255], zero is not allowed.

starting holding register number is two bytes in length and may be any value less than the highest holding register number available in the holding register table. The starting holding register number is equal to one less than the number of the first holding register returned in the normal response to this request, i.e. to get the first holding register number, holding register number one, enter zero for the starting holding register number. The high order byte of the starting holding register number field is sent as the first byte. The low order byte is sent next.

number of holding registers value is two bytes in length and must be in the range from 1 to 128 inclusive. It specifies the number of holding registers returned in the normal response. The sum of the starting holding register value and the number of holding registers value must be less than or equal to the highest holding register number available in the holding register table. The high order byte of the number of holding registers field is sent as the first byte. The low order byte is sent next.

Format of normal message reply from the slave is shown in Figure 10-18.

SLAVE ADDRESS	FUNCTION CODE 03	BYTE COUNT	FIRST REGSTR (MSB)	FIRST REGSTR (LSB)	... LAST REGSTR (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-18. Function Code 03 Reply Format

byte count even binary number from 2 to 254, or zero. If the byte count is zero (0), then the master is to assume 256 data bytes follow. Otherwise, the specified number of data bytes follow. The byte count specifies the total number of bytes in the message following the byte count, not including the two CRC-16 bytes.

holding registers are returned in the data field in order of number with the lowest number holding register in the first two bytes, and the highest number holding register in the last two bytes of the data field. The number of the first holding register in the data field is equal to the starting holding register number plus one. The high order byte is sent before the low order byte of each holding register.

10-7.3.4. FUNCTION CODE 04: READ INPUT REGISTERS. Function code 04 is used to read input registers. The format of a message from the master is shown in Figure 10-19.

SLAVE ADDRESS	FUNCTION CODE 04	START REG # (MSB)	START REG # (LSB)	NUMB REGS (MSB)	NUMB REGS (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-19. Function Code 04 Message Format

slave address must be in the range [1-255], zero is not allowed.

starting input register number is two bytes in length and may be any value less than the highest input register number available in the input register table. The starting input register number is equal to one less than the number of the first input register returned in the normal response to this request, i.e. to get the first input register, input register number one, enter zero for the starting input register number. The high order byte of the starting input register number field is sent as the first byte. The low order byte is sent next.

number of input registers value is two bytes in length and must be in the range from 1 to 125 inclusive. It specifies the number of input registers returned in the normal response. The sum of the starting input register value and the number of input registers value must be less than or equal to the highest input register number available in the input register table. The high order byte of the number of input registers field is sent as the first byte. The low order byte is sent next.

Format of normal message reply from the slave is shown in Figure 10-20.

SLAVE ADDRESS	FUNCTION CODE 04	BYTE COUNT	FIRST REGSTR (MSB)	FIRST REGSTR (LSB)	... LAST REGSTR (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-20. Function Code 04 Reply Format

byte count is an even binary number from 2 to 250. The specified number of data bytes follow. The byte count specifies the total number of bytes in the message following the byte count, not including the two CRC-16 bytes.

input registers are returned in the data field in order of number with the lowest number input register in the first two bytes, and the highest number input register in the last two bytes of the data field. The number of the first input register in the data field is equal to the starting input register number plus one. The high order byte is sent before the low order byte of each input register.

10-7.3.5. FUNCTION CODE 05: FORCE SINGLE HOLDING COIL. Function code 05 is used to force (or write) a single holding coil in the holding coil table. The format of a message from the master is shown in Figure 10-21.

SLAVE ADDRESS	FUNCTION CODE 05	HOLDING COIL # (MSB)	HOLDING COIL # (LSB)	STATE 00H OR 0FFH	00H	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-21. Function Code 05 Message Format

slave address zero indicates that the message received is a broadcast. Non-broadcast messages use a slave address in the range [1-255]. If the the message is not a broadcast, the holding coil written will be to the unit as defined by the slave address and the <I> system configuration file. In the event that the message is a broadcast, the holding coil will be written to all units with which the <I> acts as a slave as defined in the <I> system configuration file.

holding coil number is two bytes in length and may be any value less than the highest holding coil number available in the holding coil table. The holding coil number is equal to one less than the number of the holding coil forced, i.e. to change the first holding coil, holding coil number one, enter zero for the holding coil number. The high order byte of the starting holding coil number field is sent as the first byte. The low order byte is sent next.

state byte sent by the master with only two possible values. A zero is sent (00h) to turn the specified holding coil off (set false). A value of 255 is sent (FFh) to turn the specified holding coil on (set true). The state byte is always followed by a single byte with value zero.

Format of normal message reply from the slave is identical to the received message. If the message was a broadcast there is no reply.

10-7.3.6. FUNCTION CODE 06: PRESET SINGLE HOLDING REGISTER. Function code 06 is used to preset (or write) to a single holding register. The format of a message from the master is shown in Figure 10-22.

SLAVE ADDRESS	FUNCTION CODE 06	HOLDING REG # (MSB)	HOLDING REG # (LSB)	REG DATA (MSB)	REG DATA (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-22. Function Code 06 Message Format

slave address zero indicates that the message received is a broadcast. Non-broadcast messages use a slave address in the range [1-255]. If the message is not a broadcast, the holding register written will be to the unit as defined by the slave address and the <I> system configuration file. In the event that the message is a broadcast, the holding register will be written to all units with which the <I> acts as a slave as defined in the <I> system configuration file.

holding register number is two bytes in length and may be any value less than the highest holding register number available in the holding register table. The holding register number is equal to one less than the number of the holding register changed by this request, i.e. to change the first holding register, holding register number one (1), enter zero (0) for the holding register number. The high order byte of the starting holding register number field is sent as the first byte. The low order byte is sent next.

holding register data field is two bytes in length and contains the value to which the holding register specified by the holding register number field is preset. The first byte in the data field contains the high order byte of the preset value. The second byte in the data field contains the low order byte.

Format of normal message reply from the slave is identical to the received message. If the message was a broadcast there is no reply.

10-7.3.7. FUNCTION CODE 07: READ EXCEPTION STATUS. Function code 07 is used to read the exception status data, defined as the first eight holding coils. These can be used to indicate slave controller status or condition of any other state. Function code 07 thus provides a short form of request for the purpose of reading these first eight holding coils, holding coils one through eight. The format of a message from the master is shown in Figure 10-23.

SLAVE ADDRESS	FUNCTION CODE 07	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-23. Function Code 07 Message Format

slave address must be in the range [1-255], zero is not allowed.

Format of normal message reply from the slave is shown in Figure 10-24.

SLAVE ADDRESS	FUNCTION CODE 07	DATA BYTE	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-24. Function Code 07 Reply Format

data byte field of the normal response is packed holding coil status data. The data byte contains eight holding coil values. The LSbit of the first byte contains the value of the holding coil number one (1). The MSbit contains the value of holding coil number eight. The value of the holding coils are ordered by number starting with the LSbit of the first byte of the data field and ending with MSbit.

10-7.3.8. FUNCTION CODE 08: LOOPBACK TEST. Function code 08 is used to run the loopback diagnostic tests. The purpose of the Loopback test is to test the communications system (the <I>); it does not affect the controller (the Mark V). The format of a message from the master is shown in Figure 10-25.

SLAVE ADDRESS	FUNCTION CODE 08	DIAG CODE (MSB)	DIAG CODE (LSB)	DATA (MSB)	DATA (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-25. Function Code 08 Message Format

slave address zero indicates that the message received is a broadcast. Non-broadcast messages use a slave address in the range [1-255]. The <I> will respond to requests to its designated slave addresses, or to a broadcast. If the message is not a broadcast, the data returned will be from the unit as defined by the slave address and the <I> system configuration file. In the event that the message is a broadcast, no data will be returned. Only commands to reset counters and change Listen Only mode will be accepted as broadcast messages.

diagnostic codes shown in Figure 10-27.

data fields zero unless specified otherwise.

Format of normal response message from the slave is shown in Figure 10-26.

SLAVE ADDRESS	FUNCTION CODE 08	DIAG CODE (MSB)	DIAG CODE (LSB)	DATA (MSB)	DATA (LSB)	CRC-16 (MSB)	CRC-16 (LSB)

Figure 10-26. Function Code 08 Reply Format

When a response is sent at all, the normal response to a loopback message by the master is shown above, with each of the fields listed above identical to the corresponding field sent in the message received from the master except for *diagnostic codes* 0Ah through 0Eh, where the required data is returned. If the message received was a broadcast there is no reply.

Diag Code (hex)	Name	Meaning
00	Return Query Data	The data passed in the data field will be returned to the DCS without modification.
01	Restart Comm Option	This code causes the <I> to reset the RS232 port if necessary. It will also cause a reset of all error counters associated with the MODBUS link if the high order byte of the data field is 255 (0FFh). If the <I> is in Listen Only caused by a previously issued command, the condition is reset to normal. A normal response is returned to the master before the restart is executed if and only if the <I> was not in Listen Only Mode (LOM).
04	Force slave to Listen Only Mode (LOM)	No response is issued. When this command is received, the <I> will not respond to any commands other than the Restart Comm Option command listed above.
0A	Clear Counters	All error counters and the diagnostic register will be reset. A normal response will be sent. All counters are cleared upon each power up, reboot, Clear Counters command or Restart Comm Option command.
0B	Return Bus Message Count	The total number of messages addressed to this <I> (broadcast and non-broadcast messages) modulo 65536 will be returned to the master. This counter is cleared upon each power up, reboot, Clear Counters command or Restart Comm Option command.
0C	Return Bus CRC Error Count	The total number of messages detected by the <I> that had a bad CRC modulo 65536 will be returned to the master. This counter is cleared upon each power up, reboot, Clear Counters command or Restart Comm Option command.
0D	Return Bus Exception Error Count	The total number of exception response messages sent by the <I> modulo 65536 will be returned to the master. This counter is cleared upon each power up, reboot, Clear Counters or Restart Comm Option.
0E	Return Bus Slave Message Count	The total number of messages addressed to this <I> (broadcast and non-broadcast messages) modulo 65536 that did not generate an exception condition will be returned to the master. This counter is cleared upon each power up, reboot, or clear counter command or Restart Comm Option command.

Figure 10-27. Function Code 08 Diagnostic Codes

10-8. GLOSSARY

Analog	Sixteen (16) bit signals.
Baud	The communication speed (bits per second) of MODBUS.
CDB	Control Signal DataBase. This database exists in the Mark V panel, and is used in the Control Sequence Program. Pointnames used in the Control Sequence Program are available for use in the MODBUS mapping tables.
`CONFIG.DAT`	The <I> system configuration file which defines what Mark V control panels are available, and what <I> options are enabled (MODBUS). Found in `F:\`.
CRC-16	Cyclic Redundancy Check that generates a 16 bit check word. This word verifies that the message has been transmitted and received properly. Refer to the Gould MODBUS Protocol specification or standard text books for more detailed information.
Engineering Units	Displayed units for signals, such as %, MWATT, etc.
Exception Code	The exception code in slave response lists an error that occurred.
Function Code	A message includes a function code to specify the purpose of the message.
GEDS	General Electric Drive Systems
HC	Holding coil which permits both reading and writing access. A preceding "+" found in the MODBUS listing file flags the write access.
HR	Holding register which permits both reading and writing access. A preceding "+" found in the MODBUS listing file flags the write access.
<I>	The GE Mark V operator interface computer.
IC	Input coil which permits reading.
`IO_PORTS.DAT`	The <I> port definition/configuration file which sets up the parallel port and the serial ports on the <I>. Found in F:\.
IR	Input register which permits reading.
LSbit	Least Significant Bit
LSB	Least Significant Byte
Logic	One (1) bit signals.
MSbit	Most Significant Bit
MSB	Most Significant Byte
Mapping Table	Tables which map registers and coils to Mark V CDB pointnames. See `MODBUS.DAT`.
master	The requestor (the DCS) on a MODBUS communication link.
Message	A MODBUS transmission that includes a formatted request or reply.

MODBUS.DAT	The MODBUS mapping file which determines which MODBUS points are mapped to Mark V CDB pointnames. Found in F:\UNITn.
MODBUS.LST	The MODBUS listing file which details the mapping tables and the scaling associated with the Mark V CDB points. This is used by the DCS vendor for programming the DCS. Found in F:\UNITn.
MODBUS_L	The MODBUS listing file generator. Running this creates MODBUS.LST. Found in G:\EXEC.
mode	The format of the transmitted data over the MODBUS link.
Modulo n	A method of transmitting data that limits the data to the value n. For example, Modulo 4096 would transmit 0 to 4095 unaltered. Data greater than 4095 would be divided by 4095.
packed register	A register which uses its 16 bits for (up to) 16 logics.
RS232	Two-wire serial communication link.
RTU	One of two main MODBUS transmission modes. Mode RTU (Remote Terminal Unit) transmits data as 8-bit bytes. Mode ASCII (not supported by the <I>) transmits data as 7-bit bytes, approximately doubling the transmission time.
slave	The receiver (the <I>) of the master (the DCS) requests on a MODBUS link.

CHAPTER 11

GSM GATEWAY (<G>)

11-1. INTRODUCTION

The GEDS Standard Message (GSM) Gateway enables up to eight Distributed Control Systems (DCS) to communicate with Mark V turbine controls. Specifically, this device permits the issuance of commands (Administrative and Command requests) and retrieval of (time-tagged) "periodic" and "event-driven" data from any valid Mark V database point.

Essentially an <I> configured for data translation, the Gateway comes in two configurations: a stand-alone communications gateway, <G>, and a combination gateway/operator interface, <I>/<G>. A <G> configuration is a single-purpose communication central processing unit (CPU) with no peripherals (keyboard, mouse, monitor). The device does not operate as an <I>.

Some applications require transmitting the high resolution local time tags in the Mark V for alarms (62ms), system events (62ms), and sequence of events (SOE's) for contact inputs (1ms) to the DCS. Traditional SOE's have required multiple contacts for each trip contact, with one contact wired to the turbine control to initiate a trip and the other contact wired to a separate SOE instrumentation rack for monitoring. This SOE must then be integrated into the overall control system. Mark V control systems use dedicated processors in each contact input module. This time stamps all contact inputs with a 1ms time stamp, eliminating this initial cost and long term maintenance.

An ETHERNET link is now available with TCP/IP protocol to facilitate transmitting this data with the local time tags to the plant level control. ETHERNET provides high speed 10 megabit transmission rates combined with TCP/IP, which is widely used throughout the world. GEDS supplies an application layer protocol called GSM (i.e. GEDS Standard Messages) which supports four classes of application level messages.

- **Adminstration Messages** are sent from the Mark V <I> to the DCS with a "Supported" message which describes general communication link availability and which Mark V's are available for communication on that specific link.

- **Event Driven Messages** are sent from the Mark V to the DCS spontaneously when a system alarm occurs or clears, a system event occurs or clears and a contact input (SOE) closes or opens. Each logic point is transmitted with an individual time tag.

- **Periodic Data Messages** are groups of data points which are defined by the DCS and transmitted with a group time tag. All of the 5,000 data points in the Mark V are available for transmission to the DCS at periodic rates down to 1 second. One or more data lists can be defined by the DCS using Mark V controller names and point names.

- **Command Request Messages** are sent from the DCS to the Mark V. These include turbine control commands and alarm queue commands. Turbine control commands include momentary logical commands such as raise / lower / start / stop and analog "setpoint target" commands. Alarm queue commands consist of acknowledge and reset commands as well as alarm dump requests which cause the entire alarm queue to be transmitted from the Mark V to the DCS.

The ETHERNET link supports:
1. Up to 768 alarm points per Mark V. Note that a typical Mark V contains less than 200.
2. Up to 288 contact input SOE points per Mark V.
3. Up to 128 logical system events per Mark V.
4. Up to 2,400 simultaneously defined periodic data points per Mark V.

The <I>/<G> is equipped with a color monitor, keyboard, cursor positioning device (CPD) and a printer. In a <I>/<G> configuration, the processor has two separate and complete computer systems; one for <G> and one for <I>. The <I>/<G> enclosure contains two CPU cards, two ARCNET cards, two video controller cards, two hard disk drives, and two floppy disk drives (3.5 inch diskette) and a single power supply. Access to each system is provided by a split backplane Texas Microsystems Inc (TMI) enclosure located at the rear of the housing. A two channel Serve View keyboard controlled switch from Rose Electronics allows the user to access the <G> or <I> processor (see vendor supplied documentation). The <G> half contains an ETHERNET controller board input.

For proper operation, the Gateway must be supplied with the necessary <G> communication software as well as the appropriate <I> configuration files. For more information on Gateway software, see section 11-3.

11-2. COMMUNICATION

Communication between the Gateway and the DCS is carried over a network comprised of five separate levels or layers. Layers are defined by the International Standards Organization's Open Systems Interconnection (OSI) principles. The five layers are Physical, Data Link, Network, Transport, and Application.

11-2.1. Physical Layer

The first layer or "physical layer" is an ETHERNET link. An industry standard, ETHERNET supports 10 megabit transmission rates over thin or thick coax cabling. The characteristics of these cables are as follows:

ThinWire™
- (10Base2): per IEEE 802.3
- < 607' (185m) without repeaters
- Cable type: 50 ohm, RG 58 C/U

ThickWire™
- : (10Base5): per IEEE 802.3
- < 1,640' (500 m) without repeaters
- Cable type: 50 ohm, ETHERNET cable

Termination of the ETHERNET link at the Gateway requires a BNC "T" type connector equipped with a 50 ohm resistor.

NOTE

ETHERNET and ARCNET connections are made with BNC connectors. Make sure that the two different network cables are attached to the correct ports at the back of the Gateway. The ARCNET cable is connected to the port containing Dual Inline Package (DIP) switches and the ETHERNET cable connects to the other port.

Although other physical networks may be implemented in the future (token ring, ARCNET, etc.), ETHERNET is the only physical link layer supported by GEDS.

11-2.2. Data Link

The Data Link layer maintains a reliable communications link between adjacent nodes on the network. It inserts addresses in the data frames and provides error control for the data. Within the Gateway, the Data Link is implemented by a packet driver such as an ETHERNET driver program.

11-2.3. Network Layer

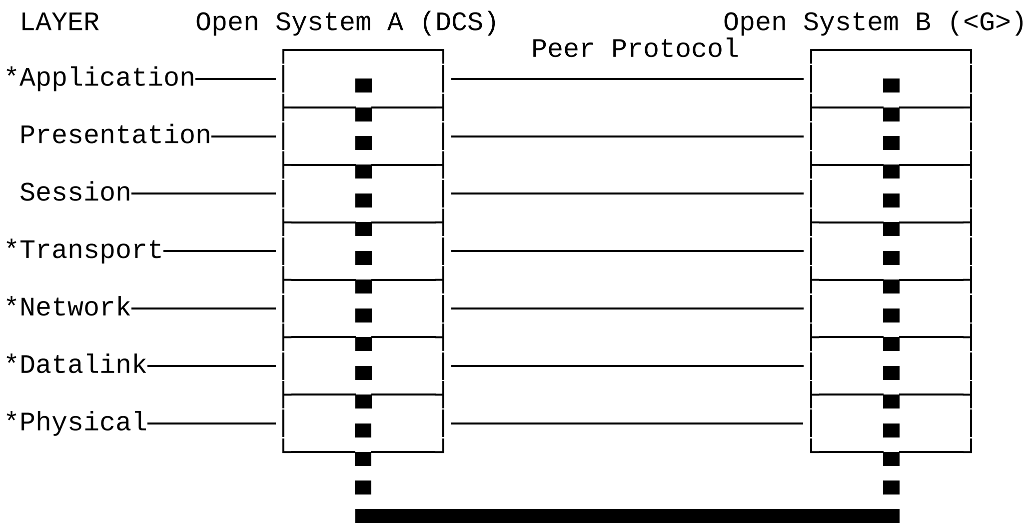
The Network layer is responsible for routing transmissions over the network. The protocol selected for this layer establishes a path between the source and destination nodes. GEDS supports the use of the Internet Protocol (IP).

11-2.4. Transport Layer

The Transport layer facilitates transmission over the communication network by segmenting source-to-destination messages into manageable "packets" before sending. It also re-assembles the packets at the message destination. For this application the Transmission Control Protocol (TCP) is used.

11-2.5. Application Layer

The Application layer permits file transfers, virtual terminal emulation, and remote database access. It acts as a direct interface for the "end user." GEDS supplies a standard message (GSM) protocol for this purpose (see GEI-100165.) Figure 11-1 depicts the layers that make up the OSI principles. Layers denoted by an * apply to the DCS/Gateway network (other layers have zero length).



* Applicable Layers

Figure 11-1. OSI Layer Reference

11-3. INSTALLATION/STARTUP

Installation and startup of the Gateway differs slightly for <G> and <G>/<I> configurations. These differences are limited to hardware connections.

11-3.1. Connections

A <G> is a PC-based processor that has ETHERNET and ARCNET connections, but no other peripheral attachments. To connect the device, attach the power cord and the two coax cables to the appropriate ports.
A keyboard and monitor is needed to install or update a <G>'s IDP Product Code and data files.

The <G>/<I> split backplane enclosure has two connection points each for the keyboard, monitor and CPD.
The keyboard controlled switch is used to share the monitor, keyboard and CPD between the <I> and <G> halves of the enclosure. There are four cable assemblies attached to the rear of the switch as follows (see Figure 11-2 for a connection diagram:

- PC-based computer power cord. Connect to the proper voltage using the international cord set and power multistrip provided.

- Common cable assembly tying the switch's input/ouput directly to the keyboard, monitor and CPD.

- Cable assembly connecting the switch's channel one input/output to the keyboard/monitor/CPD connectors for the <I> half of the PC enclosure.

- Cable assembly connecting the switch's channel two input/output to the keyboard/monitor/CPD connectors for the <G> half of the PC enclosure.

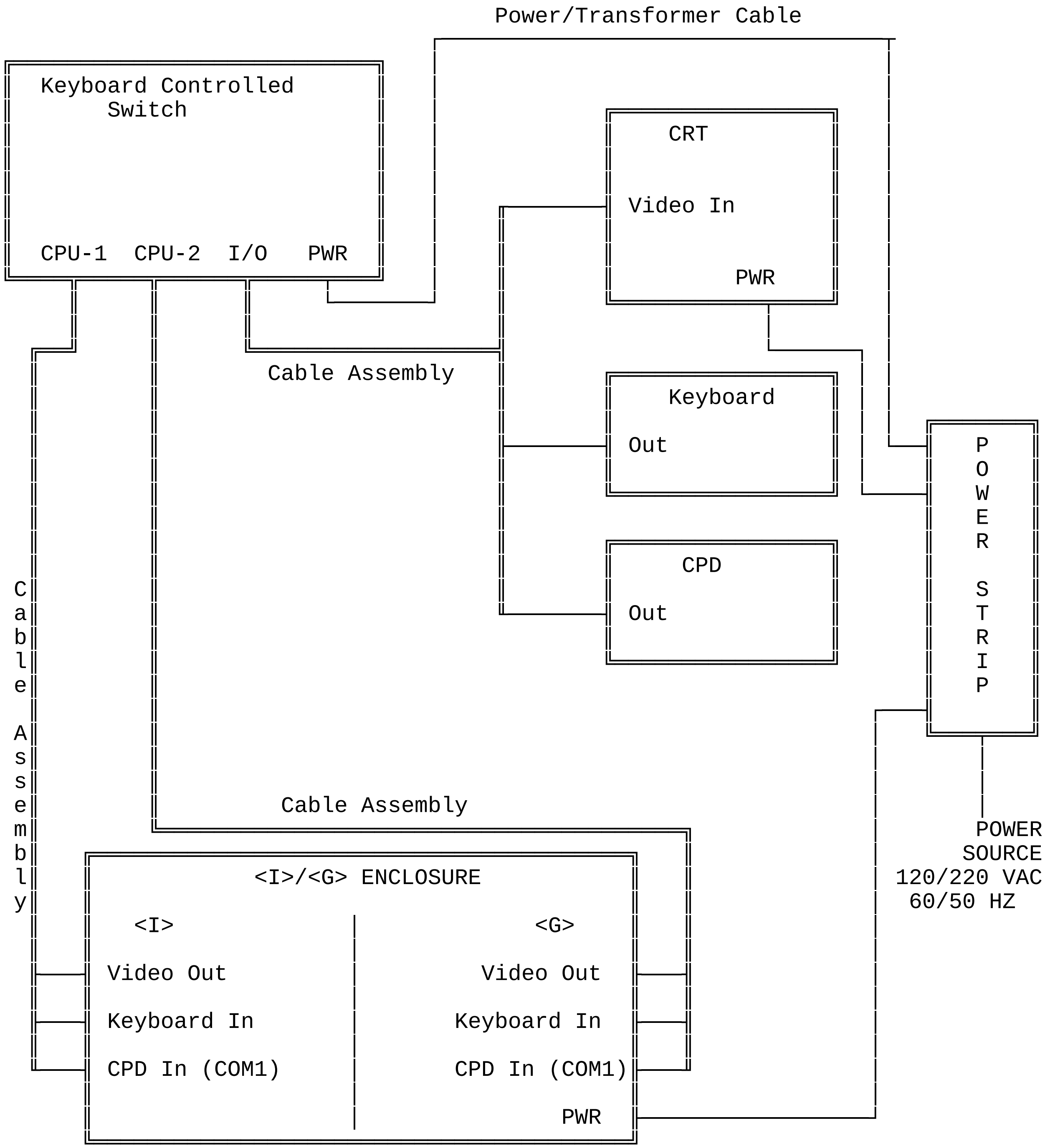


Figure 11-2. <I>/<G> Peripheral Connections

11-3.2. Applying Power

When all connections are made, power may be supplied to the CPU enclosure, monitor and keyboard controlled switch (see figures 11-2 and 11-3). The keyboard controlled switch determines the keyboard type by monitoring I/O between the CPU's and the keyboard during the power-on self test. Therefore, use the following order to apply power:

1. Apply power to electronic switch.
2. Turn on Monitor
3. Turn on CPU.

11-3.2.1. SCAN MODE. If the Gateway is an <I>/<G> device, the electronic switch allows "scan" mode. The scan button stays depressed when pushed, then releases with a second push. The depressed position is where the selection takes place. If the button is depressed it activates (or deactivates), but it is recommended to push the button a second time to return it to the released position to be ready for the next cycle. Scan causes the screen to oscillate (at five second intervals) between <I> and the <G> displays.

11-3.2.2. PROCESSOR SELECTION. The switch face has lights that show CPU status. A red light indicates which CPU is connected to the keyboard, monitor and cursor positioning device.

Since <I>/<G> devices have only one keyboard, trackball, and monitor it is necessary to switch between the <I> CPU and the <G> CPU. The electronic switch "NEXT" button toggles between the <I> and <G> CPU's.

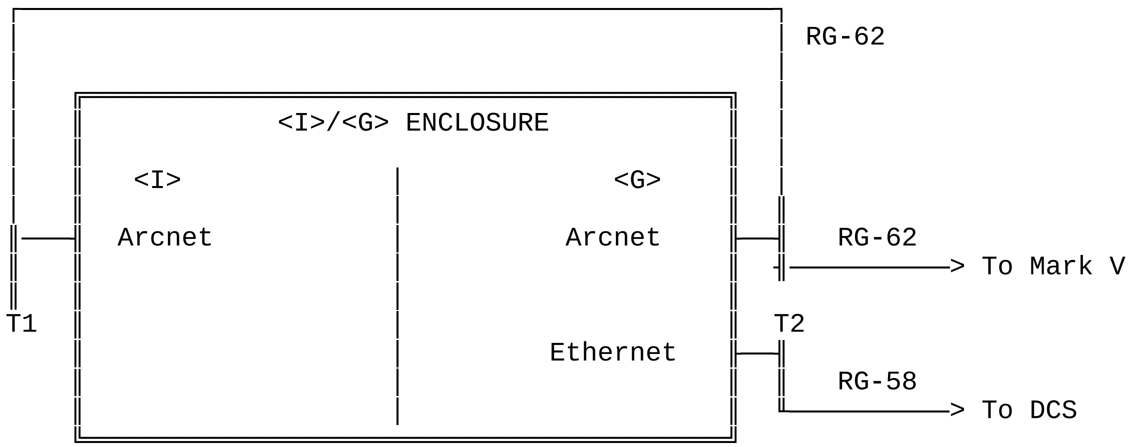
To obtain a screen prompt to recognize which processor is selected, use the following steps:

In the <I>:
1. From AUTOEXEC.BAT
2. Update the prompt command to read PROMPT=lIg pg $
3. This results in the screen prompt <I> C:\>

In the <G>:
1. From AUTOEXEC.BAT
2. Update the prompt command to read PROMPT=lGg pg $.
3. This results in the screen prompt <G> C:\>

11-3.3. Installing Software

This section contains information on installing the <I> configuration files, FTP software, and the CONFIG.DAT file, as well as information about ARCNET addresses, the PC/TCP kernel, and time tagging. It will also explain how to check the system once everything has been installed.



T1 = 93 ohm terminating resistor (ARCNET)
T2 = 50 ohm terminating resistor (ETHERNET)

Figure 11-3. Network connections for an <I>/<G> combination enclosure.

11-3.3.1. <I> CONFIGURATION FILES USED IN <G>. With the necessary hardware properly connected and powered-up, the Gateway may be loaded with the software that will permit communication with the DCS client. For the <G>, this involves installing the appropriate Mark V configuration files and communication software. This can be accomplished by following the same procedure used to load the software on the <I> processor. Four disks are supplied for configuring the <I> PC. Three of the disks are the IDP software and the fourth is the specific unit software (F:). Install this same software on <G> by inserting the disks in the <G> A: drive and using the A: INSTALL command. Install the IDP disks in order then install the unit-specific software to the F: disk. Not all of the files loaded from the `F:` disk are required to run the <G>. The following data files must be loaded in the `F:\UNITn` directory (multiple panels require multiple directories such as T1, T2, T3).

- `UNITDATA.DAT`
- `ALARM.DAT`
- `SCLEDATA.DAT`
- `LONGNAME.DAT`

All other files in `F:\UNIT1` (including the `\PROM` subdirectory) may be deleted, but it is not required. The product code IDP V3.5 or later is necessary for the gateway to operate properly.

11-3.3.2. FTP SOFTWARE. The process by which GSM sends data to the ETHERNET network requires FTP Software Inc.'s PC/TCP software. If this software has not been installed, the initial <G> screen will depict a DOS prompt such as `C:\>`. Install the FTP software as follows:

1. Create a subdirectory for the FTP software. Type the command `MKDIR C:\PCTCP`

2. Change the default directory to `C:\PCTCP`. Type `C:` and press `ENTER`

3. Change to the directory, PCTCP. Type `CD PCTCP` and press `ENTER`. The prompt `C:\PCTCP>` should be displayed.

4. Load the FTP software. Insert disk #1 into the A: drive and type `XCOPY A: *.*`

5. Repeat this step four for disks #2 and #3.

6. Copy the packet driver to the hard drive. Insert the disk labelled "SUPPORTED PACKET DRIVER DISK" into the A: drive and type `COPY A:\3C503.COM`

11-3.3.3. CONFIG.DAT. Inspect the file `CONFIG.DAT` file on the `F:\` drive to see that the GSM Server program is enabled. The keyword "OPTIONS" must begin on the first column.Under OPTIONS, the keywords EVENT, SOE, and GSM must be preceded by at least one blank space as follows: (there may be other options enabled in this section)

```
OPTIONS
        EVENT = YES
        SOE   = YES
        GSM   = YES
```

Edit the `IO_PORTS.DAT` file on the `F:\` drive to comment out the following line by putting a semicolon in front of it. This prevents an error message during startup of <G>.

```
;MODBUS PORT S2 SLAVE1 UNIT T1 MODE
```

The port number (S2), unit designation (T1), and mode (native) are typical examples only. The PC can now be configured to automatically run the <G> software by copying the template file from `G:\DATA` to the `AUTOEXEC.BAT` file. To install this file type the following command:

```
COPY G:\DATA\AUTOEXEC.GDP C:\AUTOEXEC.BAT
```

After the computer is restarted, the `AUTOEXEC.GDP` file provides the prompt display `<G> C:\>`.

The <G> software is designed to operate without using a keyboard, monitor, or CPD. As such, the device's BIOS SETUP program must be altered to allow boot-up without a keyboard connection. To enter the program and make the necessary corrections, press `<CTRL><ALT>S.` Use the up/down arrow keys to the "KEYBOARD" parameter on the setup screen. Use the left/right arrow keys to select NOT INSTALLED. Press `<ESCAPE>` to exit setup. This also automatically reboots the machine and makes the changes permanent.

NOTE

The BIOS SETUP program must be altered for <G> gateways, and is recommended for <G>/<I> combination installations.

11-3.3.4. ARCNET ADDRESS. Each Gateway must be assigned an ARCNET address. Gateway address assignments read as follows:

G1 ==> 17H	G2 ==> 16H	G3 ==> 15H	G4 ==> 14H
G5 ==> 13H	G6 ==> 12H	G7 ==> 11H	G8 ==> 10H

Setting the Gateway's ARCNET address involves manipulating the ARCNET's DIP switches, located at the back of the CPU enclosure for the <G>. The split backplane of a combined <I>/<G> unit has three cards for the <I> on the left (facing the back) and four cards on the right for the <G>. The ARCNET card is identified with an BNC connector at the top, two LEDs in the middle, and DIP switches near the bottom of the card. Reading from top to bottom, these switches are numerically labelled 8-1. (On one type of ARCNET card: this may vary. See the IDP_CARD diskette sent with each <I> and any replacement card ordered for the <I>. This diskette contains several text files that may have different instructions for different types of ARCNET cards.) The least significant nibble is set by switches 1-4 and the most significant nibble is set by switches 5-8. Therefore, setting the ARCNET address to 17H would involve setting the DIP switches as shown in Figure 11-4.

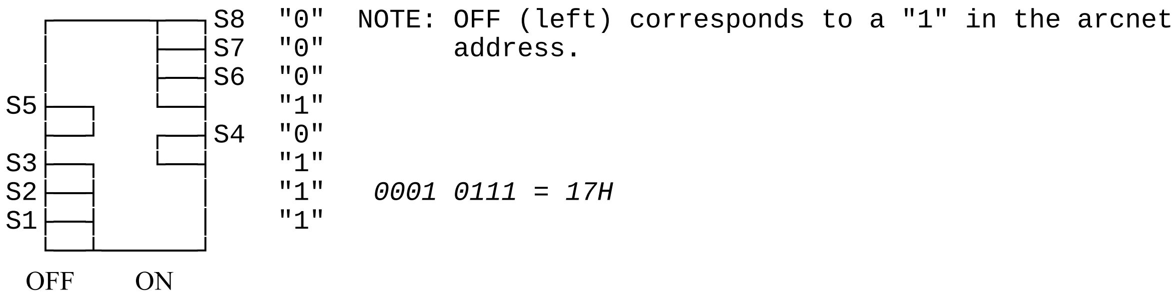


Figure 11-4. Example of ARCNET DIP Switch Configuration

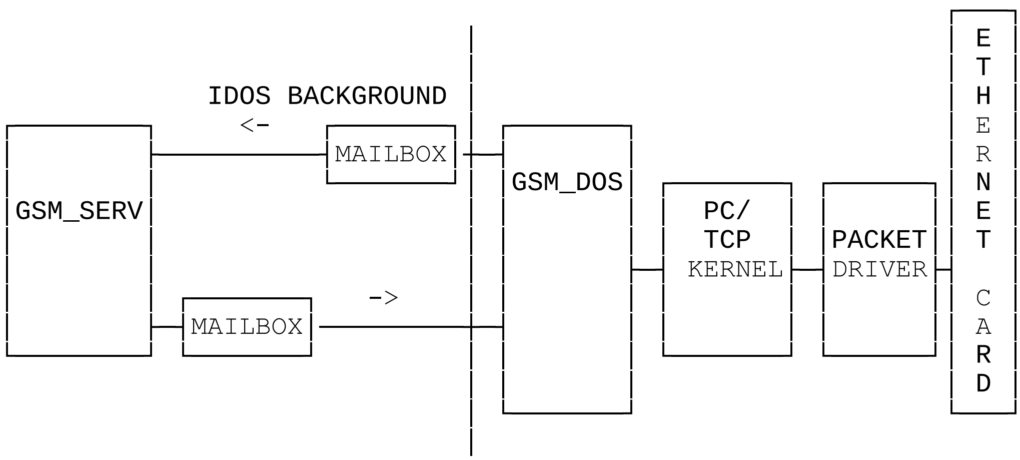


Figure 11-5. Gateway Signal Routing

11-3.3.5. PC/TCP KERNEL. Figure 11-5 shows how information is transferred from the IDOS background task GSM_SERV to the ETHERNET packet drivers, and subsequently to the ETHERNET card itself. The PC/TCP kernel uses a configuration file for various functions. The file `C:\PCTCP/PCTCP.INI` contains various sections whose names are surrounded by square brackets ([ ]). For example, the beginning of parameters used by the kernel section is delimited by a line containing "[pctcp kernel]." Use the following steps to properly set up the PC/TCP configuration files on <G>.

1. Copy the files `PCTCP.INI and HOSTNAME.TXT.` from G:\DATA to the C:\PCTCP subdirectory:
   To install these files type

```
COPY G:\DATA\PCTCP.INI C:\PCTCP\
COPY G:\DATA\HOSTNAME.TXT C:\PCTCP\
```

2. Edit `C:\PCTCP\HOSTNAME.TXT.` The following is a sample file that shows how the file is structured. In each line, substitute your sitename for "sitename."

```
192.1.1.1 gl.sitename.com g1
192.1.1.2 g2.sitename.com g2
192.1.1.3 g3.sitename.com g3
192.1.1.4 g4.sitename.com g4
192.1.1.5 g5.sitename.com g5
192.1.1.6 g6.sitename.com g6
192.1.1.7 g7.sitename.com g7
192.1.1.8 g8.sitename.com g8
#
192.1.1.33 dcs1.sitename.com dcs1
192.1.1.34 dcs2.sitename.com dcs2
192.1.1.35 dcs3.sitename.com dcs3
192.1.1.36 dcs4.sitename.com dcs4
192.1.1.37 dcs5.sitename.com dcs5
192.1.1.38 dcs6.sitename.com dcs6
192.1.1.39 dcs7.sitename.com dcs7
192.1.1.40 dcs8.sitename.com dcs8
```

3. Edit the `C:\PCTCP\PCTCP.INI` file.

   a. Under the "[pctcp general]" section:
   Set the "host-name" parameter to "Gx" —where "x" is 1 to 8 (<G> number).
   Set the "domain" parameter to sitename.com. Substitute your sitename for "sitename".

   b. Under the "[pctcp general]," section:
   Set "time-zone" to the correct value (EST, CST, MST, PST, etc.).
   Set "time-zone-offset" to the correct offset from GMT (300 if EST, 360 if CST, 420 if MST, etc.) (GMT - Greenwich Mean Time)

   c. Under the "[pctcp ifcust 0]" section the IP address for each Gateway must be set. Use the following IP addresses for the "ip-address" parameter:

G1 ===> 192.1.1.1	G2 ===> 192.1.1.2	G3 ===> 192.1.1.3	G4 ===> 192.1.1.4
G5 ===> 192.1.1.5	G6 ===> 192.1.1.6	G7 ===> 192.1.1.7	G8 ===> 192.1.1.8

   d. Under the "[pctcp ifcust 0]" section:.
   Set the "subnet-mask" parameter to 255.255.255.0.
   These are Class C Internet addresses. Note that the ETHERNET link used by the Gateways and associated DCSs must be a self-contained network with no other connections to the Internet.

e. Under the "[pctcp kernel]" section:
If the Gateways's PC/TCP software is loaded as described in section 11-3.3.7., the software serial number and authentication key must be set. This information can be found on the diskettes used for the installation.
Type in the proper software "serial-number = xxxx-xxxx-xxxx"
Type in the proper "authentication key = xxxx-xxxx-xxxx"

f. Under the "[pctcp kernel]" section:
Set the "window" parameter to 4096
Set the "host-table" to `C:\PCTCP\HOSTNAME.TXT.`
Comment out the "large-packets" and "small-packets" parameters by placing a semicolon at the beginning of these lines.
Set the value of "tcp-connections =10".
Set the value of "udp-connections =4".

11-3.3.6. TIME TAGGING. The application layer used by the GSM is documented to time-tag data using GMT or UTC (Universal Time Coordinated) as the time base. If the Mark V is using local time instead of UTC time, copy the time base conversion datafile from `G:\DATA` into the top level of `F:\`. The command for this operation is as follows:

```
x:>COPY G:\DATA\GSM_SERV.DAT F:\
```

This template datafile has sample entries for EST. Add additional lines if required. Change the time correction factor as required for the appropriate time zone the unit is operating.

Time corrections are: (S = Standard D = Daylight)
EST = 300 minutes
EDT = 240 minutes
CST = 360 minutes
CDT = 300 minutes
See section 11-5 for more details.

11-3.3.7. SYSTEM CHECK. When the steps described in sections 11-3.1.1—11-3.1.7 are completed, test the Gateway setup. For <I>/<G>, press `CTRL-ALT-DEL` to reboot. Un-plug the keyboard from the <G> immediately but leave the monitor connected. (The reset pushbutton on an <I>/<G> unit resets only the <I> and has no effect on the <G>.)

For <G>, temporarily unplug the keyboard from the <G> and press `RESET.` Leave the monitor connected.

The startup display looks similar to an <I> startup. If the system is functioning properly, the display ends up in fifty-line mode (see figure 11-4) with the IDOS idle time showing in the upper left-hand corner. The IDOS time occupies the upper right of the display. Below these time designation is a list of up to eight "clients" (these will likely show zero connections and empty distribution lists). The IP address is displayed near the bottom of the screen. The physical address appears on the same line.

To exit from the <G> NORMAL GSM_DOS DISPLAY press `ESC`.

To re-enter the display type `RUN_GDP` or `GSM_DOS`.

11-4. GATEWAY SCREENS

Figures 11-6, 11-7, and 11-8 below show the network information provided by the GSM_DOS screens. In each display, the time of day is updated once per second. This is local <G> time, not necessarily the time in the Mark V(s). All other information is updated at 10 second intervals.

All displays use a text mode containing 50 lines by 80 characters.

On the upper left of the display is IDOS idle time expressed in percent. Below that is information regarding TCP/IP connections for each possible DCS client. Unused connections are left blank. For each TCP/IP connection, the IP address (using standard dotted-decimal notation) and foreign TCP/IP socket number (in decimal) is shown. To the right of these, is a list of "distribution lists" for any turbine unit associated with the client. Up to eight turbine units are supported by the <G>.

If the DCS client is on a distribution list for a given unit, the unit number is displayed (otherwise, it is blank). Possible distribution lists include alarms, software detected events, and digital inputs (SOE). In the sample displays shown, the client is on the distribution list for turbine units 1, 2 and 3. The information on the far right is a summary of the number of "periodic" data lists defined by the DCS, and the grand total number of periodic data points collected. This is the grand total of all units combined.

Below the DCS client information lines are two lines showing network information. This information includes the following:

- <G>'s local IP address (using standard dotted-decimal notation)
- <G>'s physical ETHERNET address (in hyphenated hex format)
- total number of ETHERNET packets received and transmitted
- total number of ETHERNET receive and transmission errors. Excessive errors, particularly transmission errors, generally indicates an unterminated network, or a bad network configuration.

This information is common to all three display pages. Display pages two and three show internal diagnostic counters. To cycle through each diagnostic display page, press `<CTRL-D>` (for diagnostics). The display is immediately cleared and reformatted. However, display information may take up to 10 seconds to appear on the screen. Press `<CTRL-D>` once to cycle to page two. Press again to cycle to page three. A third press cycles back to page one.

The page two diagnostic shows information about the ETHERNET interface. This includes a summary of receive errors due to various reasons and a summary of transmission errors due to other reasons. The main item on this page is the "minimum" number of large and small buffers that are free at any one time. If either of these counters is zero, then the <G> communication system was overloaded at some point in time.

The page three diagnostic shows information regarding the PC/TCP kernel. There are four groups of information regarding the protocols used by the kernel. These include Transmission Control Protocol (TCP) counters, Internet Protocol (IP) counters, User Datagram Protocol (UDP) counters and Internet Control Message Protocol (ICMP) counters. The <G> computer does not use UDP, so all of the UDP counters should be zero.

```
IDOS IDLE TIME:  90.4%                  IDOS TIME: 03-NOV-1993 14:59:55

                    IP          TCP    ALARM     EVENT     DIGIN   PERIODIC
                  ADDRESS      SOCKET  LISTS     LISTS     LISTS   LISTS/#POINTS
                                      --------UNIT NUMBERS--------
CLIENT 1       192.1.1.2       3644     123       123       123       6 /   33

CLIENT 2

CLIENT 3

CLIENT 4

CLIENT 5

CLIENT 6

CLIENT 7

CLIENT 8

Host IP Address: 192.1.1.1   Ethernet Address: 02-60-8C-AA-D6-41
Pkts in:     95  Pkts out:    103  Recv Errors:      0  Xmit Errors:      0
```

Figure 11-6. Normal GSM_DOS Display

```
IDOS IDLE TIME:  97.1%                  IDOS TIME: 03-NOV-1993 15:00:09

                  IP          TCP    ALARM     EVENT     DIGIN   PERIODIC
                ADDRESS      SOCKET   LISTS     LISTS     LISTS   LISTS/#POINTS
                                     --------UNIT NUMBERS--------
CLIENT 1       192.1.1.2      3644     123       123       123      6 /   33

CLIENT 2

CLIENT 3

CLIENT 4

CLIENT 5

CLIENT 6

CLIENT 7

CLIENT 8

Host IP Address: 192.1.1.1   Ethernet Address: 02-60-8C-AA-D6-41
Pkts in:    104  Pkts out:    111  Recv Errors:      0  Xmit Errors:      0

Debugging Information For Ethernet Interface
Interrupts: 0 (104 receive, 0 transmit)
Receive errors: 0, Unknown type: 0
        Runts: 0, Aligns: 0, CRC: 0, Parity: 0, Overflow: 0
        Pkt Too Big: 0, Out Of Buffers: 0, RCV Timeouts: 0, RCV Reset: 0

Transmit errors: 0
        Collisions: 0, Underflows: 0, XMT Timeouts: 0, XMT Resets: 0
        Lost Carrier Sense: 0, Heartbeat Failed: 0

25 Large Buffers, 24 Free Now, Minumum of 21 Free
40 Small Buffers, 40 Free Now, Minumum of 36 Free
```

Figure 11-7. GSM_DOS Diagnostic (Page 1)

```
IDOS IDLE TIME:  96.9%                  IDOS TIME: 03-NOV-1993 15:00:29

                 IP            TCP    ALARM     EVENT     DIGIN   PERIODIC
               ADDRESS        SOCKET   LISTS     LISTS     LISTS   LISTS/#POINTS
                                      --------UNIT NUMBERS--------
CLIENT 1       192.1.1.2       3644    123       123       123      6 /   33

CLIENT 2

CLIENT 3

CLIENT 4

CLIENT 5

CLIENT 6

CLIENT 7

CLIENT 8

Host IP Address: 192.1.1.1   Ethernet Address: 02-60-8C-AA-D6-41
Pkts in:    115  Pkts out:    121  Recv Errors:      0  Xmit Errors:      0

Kernel TCP Stats: 118 pkts sent, 113 pkts rcvd, 0 bad checksums
  43869 bytes sent, 31814 bytes rcvd, 4 rexmits, 3 duplicate pkts rcvd
0 protocol errs, 2 resets, 1 segments timed-out

Kernel IP stats: 119 pkts sent, 113 pkts rcvd, 0 frags, 0 errors
  0 protocol errs, 0 timeouts, 0 bad checksums, 0 security errs
  0 bad addrs, 0 bad fragments

Kernel UDP stats:  0 pkts sent, 0 pkts rcvd, 0 no port listening
  0 bad checksums, 0 truncated rcvs, 0 dropped datagrams

Kernel ICMP stats: 0 pckts sent (0 errs), 0 pkts rcvd (0 bad)
  DestUn: 0 sent, 0 rcvd, ParamProb: 0 sent, 0 rcvd
  TimeEx: 0 sent, 0 rcvd, Redir: 0 rcvd, SourceQ: 0 rcvd
```

Figure 11-8. GSM_DOS Diagnostic (Page 2)

11-5. PERIODIC UPDATING OF <G> CONFIGURATION

The <G> computer may require periodic updating. This may include updates to IDP Product code, but generally will only involve the updating of a single datafile within the <G>.

To stop GSM_DOS for <G> updates, press `<ESCAPE>.` Any current DCS client TCP/IP connections are immediately terminated. Next, type `IDOSEXIT,` to terminate the protected mode communication programs. The system is now ready for updating.

IDP Product Code updates are done the same way as they are applied to an <I> computer.

One file that may need periodic updating is `F:\GSM_SERV.DAT`. This is a datafile that defines corrections to Mark V "Local" time so the <G> can properly convert "Local" time to UTC Universal Time Coordinated. The reason for "periodic" updating of this file is that shifting of "Local" time to/from daylight savings time is not consistent from year to year.

If the Mark V panel(s) are already running using UTC, this file does not need to exist. Otherwise, this file needs one line for each shift of standard time to/from daylight savings time. If "Local" time never switches back and forth to daylight savings time, this file only requires a single line in it to define the static correction to UTC.

Figure 11-9 shows a sample `F:\GSM_SERV.DAT` showing corrections for the U.S. Eastern Time Zone. If the local time zone was the U.S. Central Time Zone, each "correction" shown would have 60 minutes added to it. If the local time zone is the Atlantic Time Zone, each "correction" would have 60 minutes subtracted from it.

To restart the <G> after updates are made, at the DOS prompt type `RUN_GDP.`

NOTE

If the Mark V configuration files are modified in the <I> used to download the panel (`UNITDATA.DAT`, `ALARM.DAT`, `SCLEDATA.DAT` and/or `LONGNAME.DAT`), then these files must be copied from <I> to <G>. Failure to do so can result in invalid data/command translations in the <G>.

11-6. SYSTEM CAPACITIES

The following represent the maximum limits for Gateways using the DOS-based program `GSM_DOS.EXE`:

- Maximum of eight Mark Vs per Gateway
- Maximum of eight GSM client connections per Gateway
- Maximum of 32 "event driven" distribution lists per Gateway
- Maximum GSM message size = 4078 bytes
- Maximum of 200 "periodic data lists" per Gateway, each with a maximum of 96 data points defined per list (19,200 data point maximum per Gateway).

```
;
;  F:\GSM_SERV.DAT                                           15-SEP-1993
;
;  This is an optional data file, only required if the internal Mark V
;  time is Local Time instead of GMT.  This file contains one time
;  correction entry for each "switchover" point.
;
;  The format a time correction entry is:
;
;          dd-mmm-yyyy hh:mm:ss.cc nnnn
;
;  where "dd-mmm-yyyy hh:mm:ss.cc" is the date/time specification,
;  and "nnnn" is the difference, in minutes, of GMT minus Local Time.
;
;----------------------------------------------------------------------
;
; This sample is for a Mark V that is using Eastern Time Zone, USA as
; its internal time.
;
25-OCT-1992 02:00:00.00 300
04-APR-1993 02:00:00.00 240
31-OCT-1993 02:00:00.00 300
03-APR-1994 02:00:00.00 240
30-OCT-1994 02:00:00.00 300
;
;  Add other lines as required...
;
```

Figure 11-9. SAMPLE F:\GSM_SERV.DAT DATAFILE

Notes:

APPENDIX A

HARDWARE JUMPERS

TABLE OF CONTENTS

Notes:

APPENDIX A

HARDWARE JUMPERS

A-1. INTRODUCTION

Hardware jumpers are used to configure Mark V hardware (printed circuit cards and terminal boards) to meet specific application requirements. The purpose of this appendix is to introduce the specific manner that the jumper customizing process is applied.

A hardware jumper consists of a pin, or series of pins mounted perpendicularly to a printed circuit board, and a plastic and metal jumper that can be manually inserted over pairs of pins. When the jumper is placed over two pins, a connection is made and the function associated with those pins is enabled. When not in use, a jumper may be stored in proximity to a specific pin arrangement by placing the jumper over only one pin. This does not create a circuit connection, but will be available if a change is necessary.

WARNING

Hardware jumpers perform a wide variety of functions. These devices regulate operations such as card frequency, memory storage, overspeed trip settings, and system grounding. Improper application of hardware jumpers can cause significant damage to the system or personnel.

A-1.1. Table Formatting

Tables have been used to show options available. Each table in Appendix A represents a breakdown of hardware jumpers that correspond to a specific circuit card or terminal board.

For printed circuit cards, tables are divided into five separate columns (see Figure A-1). Reading from left to right, the first column indexes the card that the table is referencing such as DCC or LCC; the core(s) where the card is found such as <R>, <S>, or <T>; the location(s) of the card within these core(s) (identical cards may occupy multiple locations within multiple cores); and a card revision number (Rev.). The second column gives the number of the jumper being identified. The third column defines the position(s) to place a jumper for a specific pin configuration. This category is subdivided into separate listings when an individual pin configuration has multiple functions. The fourth column of the table (Application Notes) describes the functions of the various pin configurations.

ABCD <C> Location 1	Jumper	Position		Application Notes

Figure A-1. Table Formatting

Shaded Sections of the following tables indicate jumper functions that are not designed to be changed or do not vary with the requisition. These jumpers should not be changed.

Default Setting is a point of reference with regard to a jumper's pin configuration (see Section A-1.2 of this Appendix) as it is shipped from the factory. This is done to relate as closely as possible to what is depicted by a board's silkscreen. A default position is **not** be regarded as an absolute reference. Differing job applications may require jumper settings other than the default position setting.

A-1.2. Hardware Jumper Configurations

The following hardware jumper configuration examples show printed circuit card silkscreens and jumper positions.

```
      SYMBOL DEFINITION                           TYPICAL EXAMPLES
   1. Indicates Pin Position                  1.  Three pin, two circuit:
                        │
  ┌───────────┬────┐    │                  ┌────┬────┬────┐
┌─│ l    l    │ l  │────┘                  │ l  │ l  │ l  │
│ └───────────┴────┘                       └────┴────┴────┘
│                                                    └───┘
└« Silkscreen as it appears on card.
                                               Bridge Indicates
                                               Default Position
                                                 if present.

    2.  Three pin, two circuit:                  3.  Two pin, one circuit:
            ┌────┬────┬── ──┐                        ┌───────────┐
            │ l  │ l  │  l  │                        │ l     l   │
            └────┴────┴── ──┘                        └───────────┘

    Solid Lines can also be used              (One Circuit)  Insert Or Remove
    to indicate Default Position              Jumper can be stored on one pin.

    4.  Five pin, four circuit:                  5.  Four pin, three circuit:
                ┌────┐                                 ┌── ──┐
                │ l  │                                 │  l  │
          ┌── ──┤    ├── ──┐                     ┌─────┴─────┼── ──┐
          │ l   │ l  │  l  │                     │ l     l   │  l  │
          └── ──┼────┼── ──┘                     └───────────┴── ──┘
                │ l  │
                └── ─┘
```

A-1.3. Old and New Jumpers

A new version of the hardware jumpers has arrived. Both the old version and the new version will remain in use. The help for the old version will continue to be available in this appendix. Help for the new version resides here as well, but the primary help for the new version should be accessed on-line. The main menu of the <I> should have a help button for hardware jumpers. If not, theanimated display files CARD_HLP and HJ_HELP can be accessed to create the help file.

DCC <R>,<S>,<T>, <C>,<D> Location 1 Rev. DCCA	Jumper	Position		Application Notes
		Default		
	JP1	1-2 = Disable	2-3 = Enable	Enable write to EEPROM. Enable = normal operation
	JP7 JP8	1-2 = Normal		Not Used in Mark V Applications.
	JP15	in = enable	out = card test	Oscillator Enable. In = normal operation.
	JP19	1-2 = other applications	2-3 = MarkV applications	Water Flow Input (WF1) Must be in 2-3 position for MarkV applications.
	JP20	1-2 = other applications	2-3 = MarkV applications	Megawatt Input Must be in 2-3 position for MarkV applications.
	JP21	1-2 = other applications	2-3 = MarkV applications	Water Flow and Megawatt Input Enable Must be in 2-3 position for MarkV applications.
	JP22	In = Enable	Out = Disable	I/O Master Oscillator Enable
	JP23	1-2 = other applications	2-3 = MarkV applications	Direct Memory Access Must be in 2-3 position for MarkV applications.
	JP31	1-2 = enable	2-3 = disable	I/O Network Communication Must be in 1-2 position for Mark V applications.
	JP32	1-2 = MarkV applications	2-3	Drive Systems function Must be in 1-2 position for Mark V applications.
	JP33	In = Enable	Out = Disable	320 Co-processor oscillator enable
	JP34	1-2	1-3 4-1= operate	Microprocessor timer, 4-1 is operating position.
	JP35	1-2 = other applications	2-3 = MarkV applications	27C1010 or 170 EPROM chip Must be in 2-3 position for MarkV applications.
	JP36	1-2 = other applications	2-3 = MarkV applications	128 or 256 K RAM chip select Must be in 2-3 position for MarkV applications.

SDCC <R>,<S>,<T>, <C>,<D> Location 1 Rev. SDCCG#A	Jumper	Position		Application Notes
		Default		
	JP1	1-2 = Disable	2-3 = Enable	Enable write to EEPROM. Enable = normal operation
	JP7	1-2 = Normal		Not Used in Mark V Applications.
	JP8			
	JP12	1-2 = Disable	2-3 = Enable	Flash memory enable (Drive Applications Only) Must be in 1-2 position for MarkV applications.
	JP14	1-2 = Disable	2-3 = Enable	Flash memory enable (Drive Applications Only) Must be in 1-2 position for MarkV applications.
	JP16	1-2 = Enable	2-3 = Disable	Power supply for memory chips Enable = normal operation.
	JP22	In = Enable	Out = Disable	I/O Master Oscillator Enable
	JP23	1-2 = other applications	2-3 = MarkV applications	Direct Memory Access Must be in 2-3 position for MarkV applications.

LCC <R>,<S>,<T>, <C>,<D> Location 1 Rev. LCCA	Jumper	Position		Application Notes
		Default		
	J14	1-2 = Isolated DLAN circuit	2-3 = RS422 Drivers and Receivers	RS422 DLAN/Isolated DLAN circuits Default = normal operation
	J15			
	J16			
	J17			
	J18			
	J19	1-2 = Enable	2-3 = Disable Card test.	Oscillator Enable. Factory use only. Enable = Normal operation

SLCC <R>,<S>,<T>, <C>,<D> Location 1 Rev. SLCCG#A	Jumper	Position		Application Notes
		Default		
	JP14	1-2 = Isolated DLAN circuit	2-3 = RS422 Drivers and Receivers	RS422 DLAN/Isolated DLAN circuits Default = normal operation
	JP15			
	JP16			
	JP17			
	JP18			
	JP19	in = Enable	out = Disable Card test.	Oscillator Enable. Factory use only. Enable = Normal operation
	JP20	1-2 = Enable	2-3 = Disable	Power Supply for Memory chips. Enable = Normal operation

TCCA <C> Location 2 Rev. TCCAG#A	Jumper	Position		Application Notes
		Default		
	J1 SCPT	enable	disable	Disable Serial RS232 Port Enable = normal operation
	J2 FTST	in = enable	out = disable	Oscillator Enable Enable = normal operation

TCCA <C> Location 2 Rev. TCCAG#B	Jumper	Position		Application Notes
		Default		
	J1 SCPT	1-2 = enable	2-3 = disable	Disable Serial RS232 Port Enable = normal operation
	J2 FTST	in = enable	out = disable	Oscillator Enable Enable = normal operation
	JP3	in = enable	out = disable	Stall Enable Enable = normal operation

TCCB <C> Location 3 Optional Rev. TCCBG#A	Jumper	Position		Application Notes
	J2	**Default** = enable	disable	Disable Serial RS232 Port Enable = normal operation
	BJ1	Insert all three in for normal operation. Out = card test.		Oscillator Enable
	BJ2			Oscillator Enable
	BJ3			Oscillator Enable

TCCB <C> Location 3 Optional Rev. TCCBG#B	Jumper	Position		Application Notes
		Default		
	J1	1-2 = TC2000 Application	2-3 = EX2000 Application	Generator and Bus PT Voltage Monitoring
	J2	1-2 = TC2000 Application	2-3 = EX2000 Application	Generator and Bus PT Voltage Monitoring
	J3	1-2 = TC2000 Application	2-3 = EX2000 Application	Generator and Bus PT Voltage Monitoring
	J4	1-2 = TC2000 Application	2-3 = EX2000 Application	Generator and Bus PT Voltage Monitoring
	J5	1-2 = TC2000 Application	2-3 = EX2000 Application	Generator and Bus CT Current Monitoring
	J14	1-2 = Enable	2-3 = Disable	Disable Serial RS232 Port Enable = normal operation
	J15	1-2 = Enable	2-3 = Disable	Oscillator Enable Enable = normal operation
	J16	1-2 = Enable	2-3 = Disable	Oscillator Enable Enable = normal operation

TCDA <QDn> Location 1,2,3 <CD> Location 1 Rev. TCDAG#A and TCDA G#B	Jumper	Position		Application Notes
		Default		
	J1	Off	On = Factory Test	Card Test Enable OFF for proper operation.
	J2 J3	In = Terminated	Out = Not Terminated	IONET Termination Resistor
		0	1 Binary weighted value	
	J6 J5 J4	0 0 0	1 2 4	IONET Address This configuration varies for: <QDn>,<CD> See table next page.
	J7	1 = enable	0 = disable	Enable Stall Timer Enable = normal operation
	J8	In = operate	Out = test mode	Test Enable In = normal operation

CORE	Location	Address	J6	J5	J4
<QD1>	1	0	0	0	0
<QD2>	2	4	1	0	0
<CD>	1	0	0	0	0

TCDA --Binary Summation Card Addressing - J4, J5, J6

TCQA <R>,<S>,<T> Location 2 Rev. TCQAG#A and TCQAG#B	Jumper	Position		Application Notes
		Default		
	J1	Simplex	TMR or Redundant Control	20/200 mA Output Circuit #1 Mode Select*
	J2			20/200 mA Output Circuit #2 Mode Select*
	J5	20mA Max	200mA Max	Circuit #1 Current Range*
	J6			Circuit #2 Current Range*
	J7	Enable	Card test	RS232 Port Enable Enable = normal operation
	J8	in = enable	Out for test	Oscillator Enable Enable = normal operation

* See Appendix D, Fig D-48 and D-49.

TCQB <R>,<S>,<T> Location 3 Rev. TCQBG#A	Jumper	Position		Application Notes
		Default		
	J1	S = Simplex	R = Redundant	20\200mA Output Circuit #1 Mode Select*
	J2	20mA Max	200mA Max	20\200mA Circuit #1 Current Range*
	J3	S = Simplex	R = Redundant	20\200mA Output Circuit #2 Mode Select*
	J4	20mA Max	200mA Max	20\200mA Circuit #2 Current Range*
	J5	Accelerometer is commonly used as a vibration input on LM machines	Default Proximity Transducer	LM (Aircraft Engine Derivative) Accelerometer or Proximity Transducer Select Hardware Jumpers 5-13 should all be set the same.
	J6			
	J7			
	J8			
	J9			
	J10			
	J11			
	J12			
	J13			
	J14	disable	enable	Card Test Enable Disable = normal operation
	J15	in = enable	out = disable	Oscillator Enable Enable = normal operation
	J16			

*See Appendix D, Figure D-50.

TCQB <R>,<S>,<T> Location 3 Rev. TCQBG#B	Jumper	Position		Application Notes
		Default		
	J1	1-2 = Simplex	2-3 = Redundant	20\200mA Output Circuit #1 Mode Select*
	J2	1-2 = 20mA Max	2-3 = 200mA Max	20\200mA Circuit #1 Current Range*
	J3	1-2 = Simplex	2-3 = Redundant	20\200mA Output Circuit #2 Mode Select*
	J4	1-2 = 20mA Max	2-3 = 200mA Max	20\200mA Circuit #2 Current Range*
	J5	1-2 = Proximity Transducer	2-3 = Accelerometer is commonly used as a vibration input on LM machines	LM (Aircraft Engine Derivative) Accelerometer or Proximity Transducer Select Hardware Jumpers 5-10 should all be set the same.
	J6			
	J7			
	J8			
	J9			
	J10			
	J14	1-2 = enable	2-3 = disable	RS232 Port Enable Enable = normal operation
	J15	in = enable	out = disable	Oscillator Enable Enable = normal operation
	J16	1-2 = enable	2-3 = disable	Oscillator Enable Enable = normal operation
	J17	in = enable	out = disable	Stall Enable Enable = normal operation

*See Appendix D, Figure D-50

TCQC	Jumper	Position		Application Notes
<R>,<S>,<T> Location 4	BJ1 BJ2	These hardware jumpers are selected in pairs for the output current range, 10mA to 40 mA.	Regulator #1	For details, see Chapter 7 Servo Valve Regulator
Rev. TCQCG#A	BJ3 BJ4	10X=10mA Simplex Gas Turbin10 = 10mA TMR Gas	Regulator #2	Outputs and Appendix D, Figure D-44.
	BJ5 BJ6	20 = 20mA TMR or Simplex Medium Steam Turbine	Regulator #3	The even numbered jumpers select the feedback
	BJ7 BJ8	40 = 40mA TMR or Simplex Large Steam Turbine	Regulator #4	scaling, while the odd numbered jumpers select
	BJ9 BJ10		Regulator #5	the source output resistance and, therefore, the current
	BJ11 BJ12		Regulator #6	range
	BJ13 BJ14		Regulator #7	
	BJ15 BJ16		Regulator #8	
		Default		
	BJ17	Connect	DCOM not connected	DCOM connection for RS232 Monitor Port Connect = normal operation
	BJ18 BJ20	In = No additional Current limit.	Out = Current limit supply voltage.	Out for intrinsically safe installations; further limits P15 and N15 supply to proximity transducers. (LM installations)
	BJ21	Enabled	Disabled	Enable Stall Timer. Enable = normal operation
	BJ22	Enabled	Disabled	Oscillator Enable Enable = normal operation
	BJ23 BJ24	Out = termination resistor out.	In = termination resistor in.	<BOI> RS232 termination resistor. On TMR applications, <T> core has BJ23 & BJ24 in, <R> & <S> they are out. On Simplex applications, <R> core has BJ23 & BJ24 in.

TCQC <R>,<S>,<T> Location 4 Rev. TCQCG#B	Jumper	Position	Application Notes	
	BJ1 BJ2 BJ25 BJ26 BJ27	These hardware jumpers are used to select the output current range, 10mA to 240 mA. Reference TCQCG#B Table 1 for details.	Regulator #1	For details, see Chapter 7 Servo Valve Regulator Outputs and Appendix D, Figure D-45, D-46.
	BJ3 BJ4 BJ28 BJ29 BJ30		Regulator #2	
	BJ5 BJ6 BJ31 BJ32 BJ33		Regulator #3	
	BJ7 BJ8 BJ34 BJ35 BJ36		Regulator #4	
	BJ9 BJ10	These hardware jumpers are used to select the output current range, 10mA to 40 mA.Reference TCQCG#B Table 2 for details.	Regulator #5	
	BJ11 BJ12		Regulator #6	
	BJ13 BJ14		Regulator #7	
	BJ15 BJ16		Regulator #8	

Jumper	Default		Application Notes
BJ17	Connect	DCOM not connected	DCOM connection for RS232 Monitor Port Connect = normal operation
BJ18 BJ20	In = No additional Current limit.	Out = Current limit supply voltage.	Out for intrinsically safe installations; further limits P15 and N15 supply to proximity transducers. (LM installations)
BJ21	1-2 = Enabled	2-3 = Disabled	Enable Stall Timer. Enable = normal operation
BJ22	Enabled	Disabled	Oscillator Enable Enable = normal operation
BJ23 BJ24	Out = termination resistor out.	In = termination resistor in.	<BOI> RS232 termination resistor. On TMR applications, <T> core has BJ23 & BJ24 in, <R> & <S> they are out. On Simplex applications, <R> core has BJ23 & BJ24 in.
JP38	1-2 = 0.195Vp	2-3 = 0.3868 Vp	FF2 Mag Pickup Hysteresis Level
JP39	1-2 = 0.195Vp	2-3 = 0.3868	FF1 Mag Pickup Hysteresis Level

TCQCG#B Table 1

Nominal Output (mA)	Servo Coil (Ohms)	BJ1,3,5,7					BJ2,4,6,8			BJ25,28,31,34		BJ26,29,32,35		BJ27,30,33,36	
		10	10X	20	40	80	DEF	Y	Z	DEF	A	DEF	B	DEF	C
10 (Gas Simplex)	1000		X				X			X		X		X	
10 (Other)	1000	X					X			X		X		X	
20	125			X				X		X		X		X	
40	62.5				X				X	X		X		X	
80	22					X			X		X	X		X	
120	40	X							X		X		X	X	
240	37.5	X							X		X		X		X

TCQCG#B Table 2

Nominal Output (mA)	Servo Coil (Ohms)	BJ9,11,13,15				BJ10,12,14,16		
		10	10X	20	40	10	20	40
10 (Gas Simplex)	1000		X			X		
10 (Other)	1000	X				X		
20	125			X			X	
40	62.5				X			X

TCQF* <R>,<S>,<T> Location 3 Rev. TCQFG#A	Jumper	Position		Application Notes
		Default		
	J1	in = enable	out = card test	
	J2	1-2	2-3	Port Enable Disable port = normal operation
	J3	1-2 (Simplex)	2-3 (TMR)	MAO #1 (TBQC) **
	J4	1-2 (Simplex)	2-3 (TMR)	MAO #2 (TBQC) **
	J5	1-2 (Simplex)	2-3 (TMR)	MAO #3 (TBQG) **
	J6	1-2 (Simplex)	2-3 (TMR)	MAO #4 (TBQG) **
	J7	1-2 (Simplex)	2-3 (TMR)	MAO #5 (TBQG) **
	J8	1-2 (Simplex)	2-3 (TMR)	MAO #6 (TBQG) **

* The TCQF card is an optional card for the Gas Heavy Duty (GHD) prom set and will replace the TCQB card.
** See Appendix D, Fig D-51 and D-52.

<table>
<tr><td rowspan="11">TCEA
<P>
X Loc. 1
Y Loc. 3
Z Loc. 5

Rev. TCEAG#A</td><td>Jumper</td><td colspan="2">Position

Default</td><td>Binary Value</td><td>Application Notes</td></tr>
<tr><td>J1</td><td>0</td><td>1 = test</td><td></td><td>Test Function: 1 = Enable</td></tr>
<tr><td>J2
J3</td><td>0</td><td>1 = term</td><td></td><td>IONET Termination Resistors
0 = resistors not connected</td></tr>
<tr><td>J4
J5
J6</td><td colspan="2">See Note 1 table below</td><td>1
2
4</td><td>IONET Address
This configuration varies for X,Y,Z</td></tr>
<tr><td>J8
J9
J10
J11</td><td colspan="2">Application Specific</td><td>1
2
4
8</td><td>Base Frequency, Low Pressure Shaft
See Note 2, next page</td></tr>
<tr><td>J12
J13
J14
J15</td><td colspan="2">Application Specific</td><td>1
2
4
8</td><td>Base Frequency, High Pressure Shaft
See Note 3, next page</td></tr>
<tr><td>J16
J17
J18
J19
J20
J21</td><td colspan="2">Application Specific</td><td>1
2
4
8
16
32</td><td>High Pressure Shaft Trip Frequency
See Note 5, next page</td></tr>
<tr><td>J22
J23
J24
J25
J26
J27</td><td colspan="2">Application Specific</td><td>1
2
4
8
16
32</td><td>Low Pressure Shaft Trip Frequency
See Note 5, next page</td></tr>
<tr><td>J28
J29</td><td colspan="2">Application Specific</td><td>1
2</td><td>Type of control application
See Note 4, next page</td></tr>
<tr><td>J30</td><td>1=enable</td><td>0=stall disable</td><td rowspan="2"></td><td>Stall Function: must be in 1 position.</td></tr>
<tr><td>J31</td><td>In</td><td>Out</td><td>Factory test. Install for proper field operation.</td></tr>
</table>

TCEA, continued

Note 1: TCEA - X,Y,Z Card Definition Determined According to Binary Summation

Card	Address	J4	J5	J6
X	4	0	0	1
Y	5	1	0	1
Z	6	0	1	1

Note 2: TCEA -- Base Frequency of the speed sensors for the Low Pressure Shaft of a dual shaft machine. Select Table Frequency which is the closest to: {rated RPM * #teeth /60 sec/min}.

Frequency	J11	J10	J9	J8
No Trip*	0	0	0	0
2880**	0	1	1	0
3300	0	0	0	1
4150**	0	1	0	1
4670	0	0	1	0
4980	0	0	1	1
6500	0	1	0	0
Not Used	1	1	1	1

*No Trip is the default setting for single shaft machines

**Available with TCEAFIACJ on later proms

Note 4: TCEA -- Type of Control Application

Type of Application	J29	J28
LM2500 or LM 5000	0	1
Simplex	1	0
All other	1	1

Note 3: TCEA -- Base Frequency of the speed sensors for a single shaft machine or the High Pressure Shaft of a dual shaft machine. Select Table Frequency which is the closest to: {rated RPM * #teeth/60 sec/min}

Frequency	J15	J14	J13	J12
3000	0	0	0	1
3600	0	0	1	0
3980*	1	0	0	1
4800	0	0	1	1
4860	0	1	0	0
5100	0	1	0	1
7100	0	1	1	0
7491	0	1	1	1
7833	1	0	0	0
Not Used	1	1	1	1

*Available with TCEAFIACK on later PROMS.

Note 5: TCEA -- High Pressure and Low Pressure Shaft Trip Frequency is the pulses per second from the speed sensors ("magnetic pickups") that will result in an emergency overspeed trip. See Chapter 7 of this manual for additional information about overspeed trips.

Trip Frequency = Trip RPM * #teeth/60 sec/min = Base Frequency * {100 + Binary Sum/4}/100

or, solving for Binary Sum,

Binary Sum = {(Trip Frequency/Base Frequency) -1} * 400

Where Binary Sum is the decimal equivalent of the binary weighted values.

For example:

Gas Turbine	Rated RPM = 5100, Emergency Overspeed set to 112%, 60 teeth on pickup wheel. Base Frequency = 5100 * 60/60 = 5100 Set J15 = 0, J14 = 1, J13 = 0, J12 = 1. Binary Sum = {(5712/5100) - 1} * 400 = {1.12 - 1} * 400 = 48; in binary, 48 = 110000. Set J21 = 1, J20 = 1, J19 = 0, J18 = 0, J17 = 0, J16 = 0.
Steam Turbine	Rated RPM = 3600, Emergency Overspeed set to 110%, 160 teeth on pickup wheel. Base Frequency = 3600 * 160/60 = 9600. Set J15 = 1, J14 = 0, J13 = 0, J12 = 0. Binary Sum = {(3690/3600) - 1} * 400 = 40; in binary, 40 = 101000. Set J21 = 1, J20 = 0, J19 = 1, J18 = 0, J17 = 0, J16 = 0.

WARNING

Later versions of software set the emergency overspeed jumpers differently (TCEA F#BDA or later firmware). Run the I/O Configurator for jumper settings. Enter the appropriate setting in the “HP input frequency for overspeed trip setting” field, and then clck on the verify screen target. This will update the required hardware jmper settings.

TCEA <P> X Loc. 1 Y Loc. 3 Z Loc. 5 Rev. TCEAG#B	Jumper	Position		Binary Value	Application Notes
		Default			
	J1	0	1 = test		Test Function: 1 = Enable
	J2 J3	0	1 = termination resistors		IONET Termination Resistors 0 = resistors not connected
	J4 J5 J6	See Note 1 table for TCEAG#A		1 2 4	IONET Address This configuration varies for X,Y,Z
	J8 J9 J10 J11 J22 J23 J24 J25 J26 J27	Application Specific		1 2 4 8 16 32 64 128 256 512	Low Pressure Shaft Trip Frequency See Note 6, next page
	J12 J13 J14 J15 J16 J17 J18 J19 J20 J21	Application Specific		1 2 4 8 16 32 64 128 256 512	High Pressure Shaft Trip Frequency See Note 6, next page
	J28 J29	Application Specific		1 2	Type of control application See Note 4 for TCEAG#A
	J30	1=enable	0=stall disable		Stall Function: must be in 1 position.
	J31	In	Out		Factory test. Install for proper field operation.

TCEA, continued

Note 6: TCEA -- High Pressure and Low Pressure Shaft Trip Frequency is the pulses per second from the speed sensors ("magnetic pickups") that will result in an emergency overspeed trip. For the G#B vintage of the TCEA the exact frequency is set via the I/O Configurator. While in the I/O Configurator, the required settings for the jumpers will be displayed on the screen. The binary value of the jumpers MUST be equal to the selected trip frequency divided by 16.

Binary Value = (Trip Frequency/16)

Where Binary Value is the hexadecimal equivalent of the binary weighted values.

The following tables document the structure for the HP and LP settings:

HP Trip (Hz)	Decimal value	Hex Value	J21 (512)	J20 (256)	J19 (128)	J18 (64)	J17 (32)	J16 (16)	J15 (8)	J14 (4)	J13 (2)	J12 (1)	
16383	1023	3FF	1	1	1	1	1	1	1	1	1	1	Max
16	1	001	0	0	0	0	0	0	0	0	0	1	Min
0	0	000	0	0	0	0	0	0	0	0	0	0	Shaft Not Used

LP Trip (Hz)	Decimal value	Hex Value	J27 (512)	J26 (256)	J25 (128)	J24 (64)	J23 (32)	J22 (16)	J11 (8)	J10 (4)	J9 (2)	J8 (1)	
16383	1023	3FF	1	1	1	1	1	1	1	1	1	1	Max
16	1	001	0	0	0	0	0	0	0	0	0	1	Min
0	0	000	0	0	0	0	0	0	0	0	0	0	Shaft Not Used

Example:

HP Shaft Overspeed Trip Frequency set to 3960 (110% of 3600)

3960/16 = 247.5 ===> round DOWN (always!) to 247 decimal = 0F7 Hex

0	F	7
J21 J20	J19 J18 J17 J16	J15 J14 J13 J12
0 0	1 1 1 1	0 1 1 1

TCPD <PD> Location 1 Rev. TCPDG#B	Jumper	Position		Application Notes
	BJS	In = provide ground reference for 125 VDC.	Out = 125 VDC system already has ground reference point.	Ground Reference Jumper Remove for systems with external ground reference on 125 VDC system. For circuit drawing, see Appendix E, Power Distribution Core Drawings, Figure E-1.

CTBA <C> Location 6 Rev. CTBAG#A	Jumper	Position		Application Notes
	BJ1 to BJ14	In = connects NEG of mA current inputs to DCOM	Out = mA input is not referenced to DCOM.	If the hardware jumper is installed, the NEG terminal of the respective mA input is connected to DCOM. See Connection Examples in Appendix D, Figure D-1, and Figure D-17.
	BJ15	**Default** connected	DCOM not connected	DCOM connection for RS232 Monitor Port (TIMN)

QTBA <R>,<S>,<T> Location 6 Rev. QTBAG#A	Jumper	Position		Application Notes
	J1	**Default** 20mA	1mA	0-1 mA on 4-20 mA Input Signal Select Special MW Transducer Input*

*See Appendix D, Figure D-37

Board	Jumper	Isolates Inputs	Position		Application Notes
DTBA Location 6 and DTBB Location 7 <QDn>,<CD> Revisions: DTBAG#A and DTBBG#A					
	BJ1	1-9	In = Connect +125 VDC Interrogation voltage to Contact Inputs	Out = Disconnect +125 VDC Interrogation voltage from Contact Inputs	Remove for testing only. Isolates the 125VDC positive bus from the output wiring to help troubleshoot ground faults on the 125 VDC system. See Appendix D, Figure D-3, and D-4 for more information.
	BJ2	10-18			
	BJ3	19-27			
	BJ4	28-36			
	BJ5	37-48			

DTBC <QDn>,<CD> Location 8 Rev. DTBCG#A	Jumpers Pn & Mn	Position		Application Notes
	1	**BOTH** In = MarkV Supplies power to output. Solenoid Driver Output Voltage depends on application.	**BOTH** Out = "Dry" Contact Output	**ALWAYS INSTALL JUMPERS IN PAIRS** Install corresponding (Pn & Mn) Jumpers for Solenoid Driver Output. eg: Insert P8 and M8 for solenoid circuit 8. Remove corresponding (Pn & Mn) Jumpers for dry contact outputs. eg: Remove P10 and M10 for dry contact circuit 10. For more details, see Appendix D, Figure D-5.
	2			
	3			
	4			
	5			
	6			
	7			
	8			
	9			
	10			
	11			
	12			
	13			
	14			
	15			
	16			
	17			
	18			

DTBD <QDn>,<CD> Location 9 Rev. DTBDG#A	Jumpers Pn & Mn	Position		Application Notes
	1	**BOTH** In = MarkV Supplies power to output. Solenoid Driver Output Voltage depends on application.	**BOTH** Out = "Dry" Contact Output	**ALWAYS INSTALL JUMPERS IN PAIRS** Install corresponding (Pn & Mn) Jumpers for Solenoid Driver Output. eg: Insert P8 and M8 for solenoid circuit 8. Remove corresponding (Pn & Mn) Jumpers for dry contact outputs. eg: Remove P10 and M10 for dry contact circuit 10. For more details, see Appendix D, Figure D-6.
	2			
	3			
	4			
	5			
	6			
	7			
	8			
	9			
	10			
	11			
	12			
	13			
	14			
	15			
	16			

TBQB optional <R> Location 7 Rev. TBQBG#A	Jumper	Position		Application Notes
	BJ1	All In = Fan one input to all 3 processors	All Out = one input to each processor	Pressure Input #1 Transducer Configuration*
	BJ2			
	BJ3			
	BJ4			
	BJ5	In = mA input, 250 ohm burden resistor installed; one for each input to <R>, <S>, and <T>.	Out = voltage input, no burden resistor. If BJ1 - BJ5 are in, BJ6 and BJ7 MUST be out.	Pressure Transducer Input #1 Current or Voltage Input Burden Resistor (250 ohm) Configuration CPD inputs*
	BJ6			
	BJ7			
	BJ8	<R> input, In = current	Out = Voltage input, no burden resistor	± 4-20 mA or ±10 Vdc Burden Resistor (250 ohm) Configuration
	BJ9	<S> input, In = current		
	BJ10	<T> input, In = current		
	BJ11	<R> input, In = current	Out = Voltage input, no burden resistor	± 4-20 mA or ±10 Vdc Burden Resistor (250 ohm) Configuration
	BJ12	<S> input, In = current		
	BJ13	<T> input, In = current		
	BJ14	In = current input to all 3 processors	Out = Voltage input, no burden resistor	± 4-20 mA or ±10 Vdc Burden Resistor (250 ohm) Configuration
	BJ15	In = current input to all 3 processors	Out = Voltage input, no burden resistor	± 4-20 mA or ±10 Vdc Burden Resistor (250 ohm) Configuration

*See Appendix D, Figure D-41 and D-43

TBQC <R> Location 9 Rev. TBQCG#A optional <S> Location 9	Jumper	Position		Application Notes
	BJ1	In = connects NEG of mA current inputs to DCOM.	Out = mA input is not referenced to DCOM.	If the hardware jumper is installed, the NEG terminal of the respective mA input is connected to DCOM. See Connection Examples in Appendix D, Figure D-1, Figure D-21, and D-22.
	BJ2			
	BJ3			
	BJ4			
	BJ5			
	BJ6			
	BJ7			
	BJ8			
	BJ9			
	BJ10			
	BJ12			
	BJ13			
	BJ14			
	BJ15			
	BJ16	**Default** 20 mA Max	200 mA Max	20/200mA output current range select for TCQA mA outputs circuit 1, and circuit 2
	BJ17			

TBQD optional <R> for Simplex <S> for TMR Location 7 Rev. TBQDG#A	Jumper	Position		Application Notes
	BJ1 BJ2	**Default** 20 mA Max	200 mA Max	20/200mA Actuator Output Current Range See Appendix D, Figure D-50

TBQE Optional <S> Location 7 Rev. TBQE G#B	Jumper	Position		Application Notes
	BJ1 BJ2	**Default** 20 mA Max	200 mA Max	20/200mA Actuator Output Current Range See Appendix D, Figure D-77

TBQG Optional <S> Location 7 Rev. TBQG G#A	Jumper	Position		Application Notes
	JP1	**Default** 1-2 = voltage input	2-3 = current input	4-20 mA or ±10 Vdc Burden Resistor (250 ohm) Configuration See Appendix D, Figure D-42 Input #1
	JP2	1-2	2-3	Input #2
	JP3	1-2	2-3	Input #3
	JP4	1-2	2-3	Input #4

<table>
<tr><td rowspan="13">TBCB
<C>
Location 7

OPTIONAL

Rev. TBCBG#A</td><td colspan="2">Jumper</td><td colspan="2">Position</td><td>Application Notes</td></tr>
<tr><td colspan="2">BJ1 to BJ22 are for Input Circuits 1 to 22, respectively</td><td>In = connects NEG of mA current inputs to DCOM</td><td>Out = mA input is not referenced to DCOM.</td><td>If the hardware jumper is installed, the NEG terminal of the respective mA input is connected to DCOM.

See Connection Examples in Appendix D, Figure D-1, D-8, and D-9.</td></tr>
<tr><td>Jumper</td><td>Input Circuit</td><td colspan="2">Position</td><td>Application Notes</td></tr>
<tr><td>BJ23</td><td>15</td><td rowspan="8">In = 4-20 mA input</td><td rowspan="8">Out = 0-1 mA input</td><td rowspan="8">Hardware Jumpers 23-30 modify the current range characteristics of circuits 15-22 (BJ 15-22).</td></tr>
<tr><td>BJ24</td><td>16</td></tr>
<tr><td>BJ25</td><td>17</td></tr>
<tr><td>BJ26</td><td>18</td></tr>
<tr><td>BJ27</td><td>19</td></tr>
<tr><td>BJ28</td><td>20</td></tr>
<tr><td>BJ29</td><td>21</td></tr>
<tr><td>BJ30</td><td>22</td></tr>
</table>

<table>
<tr><td rowspan="2">PTBA
<P>
Location 6
Rev. PTBAG#A</td><td>Jumper</td><td colspan="2">Position</td><td>Application Notes</td></tr>
<tr><td>BJ1
ALM</td><td>In = Enabled</td><td>Out = Disabled</td><td>Local Audible Alarm Enable
Remove to silence alarm.
See Appendix D, Figure D-65</td></tr>
</table>

APPENDIX B

I/O CONFIGURATION SCREENS & TERMINAL BOARD DESCRIPTIONS

Notes:

INTRODUCTION

This appendix uses two font types: `Courier indicates I/O configuration screens` and Times Roman is used to describe their function. The values shown on the screen are examples only and may not be the same used in an operating system.

CAUTION

Do not use these values in an actual site I/O configuration.

I/O configuration screens for a unit may be printed as follows:

1. Switch to Main Menu of the desired unit.
2. Select the I/O Configurator Editor menu pick.
3. Select "List Screens" display target below the alarm window. This creates an `I/O CONFIG.LST` file in the F: drive of the unit specific directory (`UNIT`n). This file contains a copy of all the unit's I/O configuration screens in ASCII text format to be used as desired. This file should be renamed to avoid overwriting.

B-1. TCQA, <Q> CORE

```
TCQA Card Definition - Socket 1 - Screen 1/21

    ePROM Revision Information:

         Major Rev:   4

         Minor Rev:   0
```

The TCQA is a multifunction card located in the <R>, <S>, and <T> cores in location 2 of each. For Simplex applications, only one card is required for the <R> core. The TCQA card configuration defines the following functions:

- Servo Regulator Control Loops
- Position Milliamp Outputs
- Regulator Time Constants
- Voltage Inputs
- Trace and Tune Definitions
- Thermocouple Inputs
- Pulse Rate Inputs
- Vibration Inputs
- mA Inputs
- Vibration Circuit Diagnostics
- Bus Under Voltage and Ground Monitoring.

For regulator application information, see Chapter 7.

```
                 TCQA Card Definition - Socket 1 - Screen 2/21

               Regulator Function Type Summary


Type:         00   Regulator not used in this application

Type:         2x   Position control with pilot control

Sub_type "x": D    Feedback: 1st LVDT/LVDR for main vlv, 2nd LVDT/LVDR for pilot
              E    Feedback: Max 1st LVD_ and 1st LVD_ for reg#+4 for main vlv,
                             max 2nd LVD_ and 2nd LVD_ for reg#+4 for pilot vlv.

Type:         4x   Position control
Sub_type "x": 0    Feedback: None
              1    Feedback: First assigned LVDT/R
              3    Feedback: Maximum of two assigned LVDT/R's
              9    Feedback: Median of three assigned LVDT/R's
```

```
                 TCQA Card Definition - Socket 1 - Screen 3/21

               Regulator Function Type Summary (continued)

Type:         5x   Flow control
Sub_type "x": 1    Feedback: First assigned Flow_input
              2    Feedback: Second assigned Flow_input
              3    Feedback: Maximum of two assigned Flow_inputs

Type:         6x   Flow control with position feedback
Sub_type "x": 4    Feedback: First Flow_input and max of 2 assigned LVDT/R's
              5    Feedback: Second Flow_input and max of 2 assigned LVDT/R's
              6    Feedback: Both Flow_inputs and max of 2 assigned LVDT/R's

Type:         7x   Pressure control with position feedback
Sub_type "x": 7    Feedback: Pressure_in (V1) and max of 2 assigned LVDT/R's
              8    Feedback: Pressure_in (V2) and max of 2 assigned LVDT/R's
              A    Feedback: Pressure_in (V1) and first assigned LVDT/R
              B    Feedback: Pressure_in (V2) and first assigned LVDT/R
              C    Feedback: Max Pressure_in (V1, V2) and max of 2 LVDT/R's
```

```
                 TCQA Card Definition - Socket 1 - Screen 4/21

            Regulator Definition for Servo Output 1

   Function type & sub-type:           77
   Valid types <00, 2D, 2E, 40, 41, 43, 49, 51, 52, 53, 77, 78, 7A, 7B, 7C>

   Suicide enable :-   Current Fault:    NO      LVDT/LDVR fault:   NO

   Current Bias: (0 to 100% rated [10,20,40])        8.0
   Current Gain: (0 to 200% rated_cur./%pos.)        5.4

   Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
   100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

   <7>  Pos limits   (-128% to 128%)  :-        Low: -100.0   High:   100.0
   <7>  Integrator convergence gain (0 to 16 psi/%): 0.0
   <7>  Position reference Gain     (0 to  2 %/psi): 0.2
        Note: for type 5, enter fuel flow data on Pulse Rate screen.
              for type 7, enter time constant data on screen #12.
```

Later software versions contain the following suicide position limits setting, for each of the eight outputs:

```
Suicide position limits (%)::        Low:  -10.0       High: 105.0
```

```
                 TCQA Card Definition - Socket 1 - Screen 5/21

            Regulator Definition for Servo Output 2

   Function type & sub-type:           43
     Valid types <00, 2D, 2E, 40, 41, 43, 49, 51, 52, 53, 64, 65, 66>

   Suicide enable :-   Current Fault:    NO      LVDT/LVDR fault:   NO

   Current Bias: (0 to 100% rated [10,20,40])        8.0
   Current Gain: (0 to 200% rated_cur./%pos.)        5.4

   Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
   100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

   <6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
   <6>   Integrator convergence gain  (0 to  1 %/%): 0.3
   <6,2>   Position reference Gain      (0 to 32 %/%): 0.309
   <6,2>   Position ref time constant   (0 to 8 Sec) : 1.0
         Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen
```

```
               TCQA Card Definition - Socket 1 - Screen 6/21

          Regulator Definition for Servo Output 3

Function type & sub-type:            51
  Valid types <00, 2D, 2E, 40, 41, 43, 49, 51, 52, 53, 64, 65, 66>

Suicide enable :-   Current Fault:    NO      LVDT/LVDR fault:   NO

Current Bias: (0 to 100% rated [10,20,40])        3.0
Current Gain: (0 to 200% rated_cur./%pos.)        7.72

Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

<6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
<6>   Integrator convergence gain  (0 to  1 %/%): 0.3
<6,2>   Position reference Gain      (0 to 32 %/%): 0.309
<6,2>   Position ref time constant   (0 to 8 Sec) : 1.0
      Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
               TCQA Card Definition - Socket 1 - Screen 7/21

          Regulator Definition for Servo Output 4

Function type & sub-type:            43
  Valid types <00, 2D, 2E, 40, 41, 43, 49, 51, 52, 53, 64, 65, 66>

Suicide enable :-   Current Fault:    NO      LVDT/LVDR fault:   NO

Current Bias: (0 to 100% rated [10,20,40])        3.0
Current Gain: (0 to 200% rated_cur./%pos.)        7.72

Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

<6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
<6>   Integrator convergence gain  (0 to  1 %/%): 0.3
<6,2>   Position reference Gain      (0 to 32 %/%): 0.309
<6,2>   Position ref time constant   (0 to 8 Sec) : 1.0
      Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
              TCQA Card Definition - Socket 1 - Screen 8/21

         Regulator Definition for Servo Output 5

 Function type & sub-type:           43
   Valid types <00, 40, 41, 43, 49, 51, 52, 53, 64, 65, 66>

 Suicide enable :-   Current Fault:    NO      LVDT/LVDR fault:   NO

 Current Bias: (0 to 100% rated [10,20,40])        3.0
 Current Gain: (0 to 200% rated_cur./%pos.)        30.13

Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

 <6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
 <6>   Integrator convergence gain  (0 to  1 %/%): 0.3
 <6>   Position reference Gain      (0 to 32 %/%): 0.309
 <6>   Position ref time constant   (0 to 8 Sec) : 1.0
       Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
              TCQA Card Definition - Socket 1 - Screen 9/21

         Regulator Definition for Servo Output 6

 Function type & sub-type:           52
   Valid types <00, 40, 41, 43, 51, 52, 53, 64, 65, 66>

 Suicide enable :-   Current Fault:    NO      LVDT/LVDR fault:   NO

 Current Bias: (0 to 100% rated [10,20,40])        3.0
 Current Gain: (0 to 200% rated_cur./%pos.)        7.72

 Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
 100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

 <6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
 <6>   Integrator convergence gain  (0 to  1 %/%): 0.3
 <6>   Position reference Gain      (0 to 32 %/%): 0.309
 <6>   Position ref time constant   (0 to 8 Sec) : 1.0
       Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
               TCQA Card Definition - Socket 1 - Screen 10/21

          Regulator Definition for Servo Output 7

   Function type & sub-type:            51
     Valid types <00, 40, 41, 43, 51, 52, 53, 77, 78, 7A, 7B, 7C>

   Suicide enable :-   Current Fault:    NO      LVDT/LVDR fault:   NO

   Current Bias: (0 to 100% rated [10,20,40])        3.0
   Current Gain: (0 to 200% rated_cur./%pos.)        7.72

   Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
   100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

   <7>  Pos limits   (-128% to 128%)  :-        Low: -100.0   High:   100.0
   <7>  Integrator convergence gain (0 to 16 psi/%): 0.3
   <7>  Position reference Gain     (0 to  2 %/psi): 0.1
        Note: for type 5, enter fuel flow data on Pulse Rate screen.
              for type 7, enter time constant data on screen #13.
```

```
               TCQA Card Definition - Socket 1 - Screen 11/21

          Regulator Definition for Servo Output 8

   Function type & sub-type:            43
     Valid types <00, 40, 41, 43, 51, 52, 53, 64, 65, 66>

   Suicide enable :-   Current Fault:    NO      LVDT/LVDR fault:   NO

   Current Bias: (0 to 100% rated [10,20,40])        3.0
   Current Gain: (0 to 200% rated_cur./%pos.)        7.72

   Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
   100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478

   <6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
   <6>   Integrator convergence gain  (0 to  1 %/%): 0.3
   <6>   Position reference Gain      (0 to 32 %/%): 0.309
   <6>   Position ref time constant   (0 to 8 Sec) : 1.0
         Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
                 TCQA Card Definition - Socket 1 - Screen 12/21

                   Time Constant Data for Type 7 Regulators


    Type 77, 7A or 7C:: SRV Warmup time constant (0 to 8 sec):      0.3601
    Type 77, 7A or 7C:: SRV Warmup time constant gain (0 to .625)   0.0
    Type 77, 7A or 7C:: System Warmup  FSR (0 to 128%):             20.0

       Note: Calibrate pressure feedback by adjusting Voltage Input no. 1.


    Type 78, 7B or 7C:: SRV Warmup time constant (0 to 8 sec):      0.2119
    Type 78, 7B or 7C:: SRV Warmup time constant gain (0 to .625)   0.02728
    Type 78, 7B or 7C:: System Warmup  FSR (0 to 128%):             20.0

       Note: Calibrate pressure feedback by adjusting Voltage Input no. 2.
```

Suicide Enable - Current Fault. Servo valve regulator outputs can be configured to be driven to zero output current on current fault detection. A current fault is defined as the condition where actual current differs from requested current by more than 25% of the maximum output current. A current fault condition causes a diagnostic alarm to be generated. The function Suicide Enable allows the individual processors to short the output and force their individual output current to zero (0). The objective is to allow the other two processors to control the output. The desired zero (0) output current is achieved by shorting the output with a pair of relay contacts. The method used has the affect of shorting two regulator outputs (1 and 2, 3 and 4, and such for A systems; 1 and 5, 2 and 6, and such for B systems). If this feature is enabled, control loss of one regulator causes two regulator output currents to be forced to zero (0). The overall system design needs to take into account limitations which affect regulator assignments.

Valid Answers: YES/NO

Suicide Enable - LVDT/R Fault. The function LVDT/R fault reduces the servo-valve regulator output current to zero (0Ma) when the position feedback is out of the limits either set by the **Suicide Position Limits** variable for newer version systems or preset limits of -2% and 102% for older systems. The regulator position feedback is monitored to determine if the device is operating properly. If the feedback is determined to be in error, a diagnositc alarm is generated. In addition, if LVDT/R fault suicide is enabled, the output current reference is set to zero (0) to eliminate only the specific regulator.

Current Bias is the amount of rated current required to hold the hydraulic servo at its present location. Hardware jumpers on the TCQC select the rated regulator current capability. The hardware jumper settings are written on the board, and are as follows:

10 Ma Gas Turbine
20 Ma Fitchburg Medium Steam Turbines
40 Ma Large Steam Turbines

The current bias is set to 3% for heavy duty Gas and Steam Turbines and 25% for aircraft derivative Gas Turbines.

Current Gain is application dependent. The value is expressed in percent (%) of output current expected for a 1% change in position reference. Application specific values are furnished in turbine department control specifications.

Zero and 100% Stroke constants correspond to the number of volts representing 0% and 100% stroke. Application specific values are furnished in turbine control specifications, but adjustments will need to be made based on the characteristics of the LVDT.

Position Limits. Type 6 and 7 regulators are position integrator (PI) control loops. Thus, the integrator needs to have negative and positive saturations defined. Typically, these values are -10% and 100%.

Integrator Convergence Gain. For type 6 and 7 regulators, the integrator output reference values are voted and returned to the individual regulators in a TMR system to prevent divergence of position references. The convergence value must be given significance when it is factored back into the individual control loops. The Integrator Convergence Gain provides this significance, for example 1/3 = 0.3333. Experience has shown that .3 is a good approximation.

Type 6

$$H(x) = Kp * \left[1 + \frac{1}{Ts} \right]$$

Type 7

$$H(x) = Kp * \left[\frac{1+Ts}{s} \right]$$

Position Reference Gain applies for type 6 and type 7 regulators only. The value is defined as Kp in the following PI Transfer Function. This number is the Gain of the PI controller

$$H(x) = Kp * \left[1 + \frac{1}{Ts} \right]$$

Position Reference Time Constant applies only to the type 6 regulator. The constant is the Ts portion of the PI controller.

Time Constant Data for Type 7 Regulators. A typical application for this regulator type is Speed Ratio Valve control. The Speed Ratio Valve is paired with the Gas Control Valve shown in the following configuration.

```
          Speed Ratio Valve     Gas Control Valve
                _____                _____
                \   /  Pressure P2   \   /
Fuel --->  ======\ /==================\ /====== ---> FSR
Flow       ======/ \==================/ \======
                /   \                /   \
                _____                _____
```

At startup, with zero to low fuel flow, the GCV is relatively closed and is limiting the total system flow. A small change in the SRV's position produces a high rate of change in the pressure P2. Therefore, the intervalve time constant is high under this condition.

When the turbine is running normally, the GCV is open, and limits the flow less severely. A small change in the SRV's position produces a small rate of change in the pressure P2 and the intervalve time constant is low.

To compensate for these changing conditions, an algorithm changes the intervalve time constant in a linear fashion with respect to FSR. The line to specify such a relationship is shown below.

```
                 |       +
                 |      /
                 |     /
        Tc       |    /
 (Time Constant) |   / \
                 |  /   \ Slope
                 | +----------
                 | (x,y)
                 +-------------
                     FSR
```

To define this line we need an (x,y) position and the slope of the line. The values of these constants are application dependent and must be specified by the Turbine Department. This information corresponds to the following I/O configurator constants.

SRV Warmup Time Constant is the y coordinate, from 0.0000 to 8.0000 seconds.

SRV Warmup Time Constant Gain is the line's slope, from 0.0000 to .62500.

System Warmup FSR is the x coordinate, from 0% to 128% FSR.

```
                 TCQA Card Definition - Socket 1 - Screen 13/21

                     Position Milli-Amp Output Configuration

Milli-Amp Output 1:

   Output used (enables output and basic diagn.):            YES
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       0
   Minimum ( 4 mA/ 40 mA) CDB value:    0.
   Maximum (20 mA/200 mA) CDB value:    0.

Milli-Amp Output 2:

   Output used (enables output and basic diagn.):            YES
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       0
   Minimum ( 4 mA/ 40 mA) CDB value:    0.
   Maximum (20 mA/200 mA) CDB value:    0.
```

Output Used. Yes or No. Enables the output and the diagnostics associated with the circiut.

Suicide Enable - Current Fault. Current fault on Outputs 1 and 2 are individual. Therefore, the choice of suicide enable is application dependent. These are typically not enabled for Simplex applications and are enabled for TMR applications. A current fault is the condition where the actual output differs from the reference by 25% of the requested output's full scale. This condition alarms loss of valve control. The suicide enable function allows the control to short the output from an individual processor across a relay contact to drive the current being supplied by that processor to zero and allowing the other two processors to control the output.

Control Data Base (CDB) Variable Full Scale is set to the maximum value of the scale type of the point assigned.

Maximum and Minimum CDB Values are used to calibrate the milliamp output to the desired range.
Minimum Value 4 mA/ 40 mA

Maximum Value 20 mA /200 mA

```
            TCQA Card Definition - Socket 1 - Screen 14/21

                            Voltage Input

                        Signal      Full Scale    Min (0v)     Max (10v)
                        in use      CDB value     CDB value    CDB value
                        -------     ----------    ---------    ---------
     Signal  1:           YES         0             0.           0.
     Signal  2:            NO         0             0.           0.

     Signal  3:            NO         0             0.           0.
     Signal  4:            NO         0             0.           0.

Signal 1 associated with regulator type 77, 7A, or 7C, when used.
Signal 2 associated with regulator type 78, 7B, or 7C, when used.
```

General. The first two inputs (signals 1 and 2) are intended for gas fuel pressures. Therefore, the outputs from the devices must be connected to three separate pairs of screws, one pair for each of <R>, <S>, and <T>. The TCQA also can supply 24 volts regulated excitation for the devices. The last two inputs (signals 3 and 4) have only one pair of screws each as they are fanned out to the separate processors.

Signal In Use defines if the voltage input is used.

Full Scale Control Data Base Value constant is set to the maximum value of the scale type of the point assigned with it.

Minimum and Maximum Control Data Base Value are used to calibrate the voltage input to the desired range.

```
                TCQA Card Definition - Socket 1 - Screen 15/21

                          Trace Buffer Definition

    Regulator Mode (tune, trace):                Trace

    First  Regulator selection (tune & trace):  1
    Second Regulator selection (trace only):    2

    Number/2 of recordings to be taken after Trigger
     (trace only) 0=0 recordings...255=510 recordings: 40 (0 to 255)

    Trace sample rate in hertz (trace only):    128Hertz(1 to 128)

    Tune disturbance magnitude (tune only):     3.906%(-10 to 10)

       Note:: For type 7 regulators, 1% equals 16 psi.
```

Tune Mode causes the valve to step by a user defined percentage while data of actual position vs. requested position is gathered at 128 Hz. The data can then be plotted to allow the user to tune the system. This is a manual tuning process and not an auto tune.

Regulator Selection defines the regulator to monitor. Valid selections are regulators 1 through 8. When in Tune mode only the first regulator selected is monitored.

 # Traces after Trigger applies to Trace Mode only.

Sample Rate in Hertz applies to Trace Mode only, as Tune Mode is always sampled at 128 Hz for two seconds:
- 1 second for Disturbance Period
- 1 second for Normal Reaction.

Disturbance Magnitude is the percentage to step the reference in Tune Mode The default value is 5%. For example, if the reference is steady at 10% then the disturbance puts the reference at 15%. After one second, the reference goes back to its value of 10%.

Trace Mode allows the user to record the actual valve position vs. requested valve position. A trigger, such as the turbine tripping, is set up in sequencing. At the occurence of this trigger, the trace buffer records the above information. The data is frozen until it is reset by a one-shot pulse applied to the database point Trace.

Regulator Selection defines the regulators to monitor. Valid selections are regulators 1 through 8. The Trace function works for two separate regulator feedback pairs.

 # Traces after Trigger determines the number of samples to save after the trigger occurs.

Sample Rate in Hertz determines the Trace mode sampling rate, as given by the following equation:
Period of sample = (128/Sample Rate) seconds

Valid sampling rates are 32, 16, 8, 4, 2, and 1 for sampling periods of 4, 8, 16, 32, 64, and 128 seconds respectively.

```
               TCQA Card Definition - Socket 1 - Screen 16/21

                    Thermocouple Type Selection

    TC  1:  K         TC  5:  K         TC  9:   K         TC 13:   K
    TC  2:  K         TC  6:  K         TC 10:   K         TC 14:   K
    TC  3:  K         TC  7:  K         TC 11:   K         TC 15:   K
    TC  4:  K         TC  8:  K         TC 12:   K

           Valid types:    K, J, E, T
             not used :    -
```

Thermocouples. This page is used to set the thermocouple types. These are typically set at type K for Gas and type E for Steam. Older gas units (retrofits) may require type J. The last valid type is T and Not used is noted by a dash "-".

```
               TCQA Card Definition - Socket 1 - Screen 17/21

                            Pulse Rate Definition

                 Gain Scaling Base   Max Pulse Rate  Application Type
                   (power of 2)       (100% rating)    (see notes)
                 -----------------   --------------  ----------------

  Pulse rate 1:      8192                 5104            speed

  Pulse rate 2:      8192                 5104            speed

  Pulse rate 3:      8192                 3980             fuel

  Pulse rate 4:      8192                 3980             fuel

    Notes:
       Valid Application Types (speed, fuel, pmg, -----)
       Pulse rate input 3 is used with regulators types 51, 53, 64, & 66.
       Pulse rate input 4 is used with regulators types 52, 53, 65, & 66.
```

Gain Scaling Base is the maximum frequency the card reads for the particular input. It must be expressed in a power of 2, larger than the Max Pulse Rate value.

Max Pulse Rate is the signal value equivalent to 100%.

Application Type determines which algorithm is used to process the input signal. Valid types are:

speed = Speed Sensor Applications
fuel = Fuel Flow Divider Applications
pmg = Permanent Magnetic Generator (A Type of Speed Input)
---- = Not Used

```
                    TCQA Card Definition - Socket 1 - Screen 18/21

                                   Vibration Definitions

                             Transducer used       Vibration Sensitivity
                                                 (0.05 to 0.50 V peak / ips peak)
                             ---------------   --------------------------------
   Transducer  1:                  YES               0.15
   Transducer  2:                  YES               0.15
   Transducer  3:                  YES               0.15
   Transducer  4:                   NO               0.15
   Transducer  5:                   NO               0.15
   Transducer  6:                   NO               0.15
   Transducer  7:                   NO               0.15
   Transducer  8:                   NO               0.15
   Transducer  9:                   NO               0.15
   Transducer 10:                   NO               0.15
   Transducer 11:                   NO               0.15
   Transducer 12:                   NO               0.15
```

Transducer Used defines if the transducer is used or not. Valid answers are YES and NO.

Vibration Sensitivity is the # Volts/in/sec produced by the Transducer. This is a transducer constant and must be supplied by the Turbine Department. Typical values are 0.15 and 0.20.

```
                     TCQA Card Definition - Socket 1 - Screen 19/21

                            Milliamp Input Definitions   1 - 8

                          Low   High
             Signal  Input  Input   Full Scale  Minimum (4mA)  Maximum (20mA)
             in use  Diagn  Diagn   CDB value    CDB value      CDB value
             ------  -----  -----   ----------  -------------  --------------
 Signal  1:    NO      NO     NO      0             0.              0.
 Signal  2:    NO      NO     NO      0             0.              0.

 Signal  3:    NO      NO     NO      0             0.              0.
 Signal  4:    NO      NO     NO      0             0.              0.

 Signal  5:    NO      NO     NO      0             0.              0.
 Signal  6:    NO      NO     NO      0             0.              0.

 Signal  7:    NO      NO     NO      0             0.              0.
 Signal  8:    NO      NO     NO      0             0.              0.
```

TCQA Card Definition - Socket 1 - Screen 20/21

Milliamp Input Definitions 9 - 15

	Signal in use	Low Input Diagn	High Input Diagn	Full Scale CDB value	Minimum (4mA) CDB value	Maximum (20mA) CDB value
Signal 9:	NO	NO	NO	0	0.	0.
Signal 10:	NO	NO	NO	0	0.	0.
Signal 11:	NO	NO	NO	0	0.	0.
Signal 12:	NO	NO	NO	0	0.	0.
Signal 13:	NO	NO	NO	0	0.	0.
Signal 14:	NO	NO	NO	0	0.	0.
Signal 15:	NO	NO	NO	0	0.	0.

Milliamp Input Screens

Signal Used is set to either YES or NO.

Low Input Diagnostic is set to either YES or NO. Enabling the function will send a diagnostic alarm indicating that the input current has gone below the predefined value of 3.68 mA.

High Input Diagnostic is set to either YES or NO. Enabling the function will send a diagnostic alarm indicating that the input current has gone above the predefined value of 20.32 mA.

Full Scale Control Data Base Value is set to the maximum value of the scale type of the point assigned with it.

Minimum and Maximum Control Data Base Value are used to calibrate the milliamp input to the desired range. An example would be to set the minimum at 90% stroke for an input of 4 mA, and to set the maximum to 102% stroke for an input of 20mA.

```
              TCQA Card Definition - Socket 1 - Screen 21/21

                      Miscellaneous Values

      Cold Junction Filtering  (power of 2):         16

      Seismic Vibr. Xdcr. Open  Ckt. (0 to 3VDC):    1.0
          <       V = 30R / (40000+R)    {eg,  1V=1379Ω }     >
          < Xdcr Resistance = 40000*V/(30-V) {eg,  1379Ω=1V } >

      125 VDC Under Voltage Limit (0 to 125 VDC):    105.0

      125 VDC Bus Ground Fault    (0 to  65 VDC):    31.23
          {voltage at which ground fault is indicated}

      P15AD & N15AD Diagnostics enable:     YES

      P15BD & N15BD Diagnostics enable:      NO
```

The following screen is a sample from the later software version:

```
              TCQA Card Definition - Socket 1 - Screen 21/21

                      Miscellaneous Values

      Seismic Vibr. Xdcr. Open  Ckt. (0 to 3VDC):    1.43
          <       V = 30R / (40000+R)    {eg,  1V=1379Ω }     >
          < Xdcr Resistance = 40000*V/(30-V) {eg,  1379Ω=1V } >

      125 VDC Over  Voltage Limit (0 to 145 VDC):    140.0

      125 VDC Under Voltage Limit (0 to 125 VDC):    105.0

      125 VDC Bus Ground Fault    (0 to  65 VDC):    31.24
          {voltage at which ground fault is indicated}

      P15AD & N15AD Diagnostics enable:      NO

      P15BD & N15BD Diagnostics enable:      NO
```

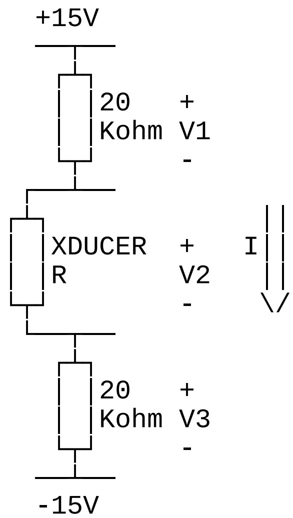
Cold Junction Filtering determines the number of readings used in a floating average noise filter. Cold junction filtering is defined by Mark V system designers and is changed based on experience. It must be set in a power of 2.

Seismic Vibration Transducer Open Circuit is expressed in terms of the Open Circuit voltage expected to indicate that the transducer has failed (open circuit). The circuit below illustrates the transducer circuit. The voltage V2 is the value being measured. The resistance R is dependent on the type of transducer used and is provided by the Turbine department. In general, any value of R greater than 2 Kohms is considered a faulty transducer. This corresponds to a voltage V2 of approx. 1.43 V.

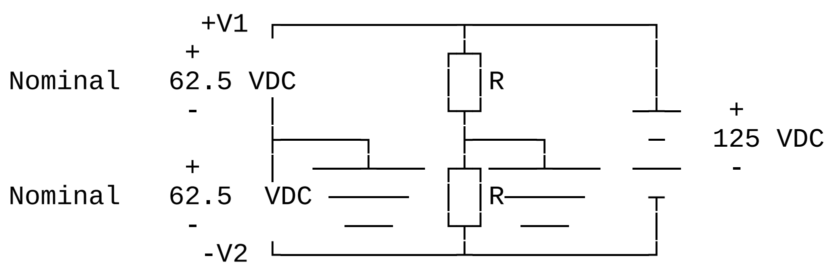
A small DC current is fed into the circuit to determine a transducer fault condition. The DC current I produces a voltage V2 at the terminal boards. The voltage is measured and compared to the Seismic Vibration Transducer Open Circuit constant. If the measured voltage is greater than the constant, a diagnostic alarm is generated. The actual vibration is the AC component that lies on top of the DC voltage signal from the transducer.

Equations

```
      Vtot = V1+V2+V3
         I = V2/R
        V2 = I*R
      Vtot = (15-(-15))
           = 30
        V1 = 20,000*I
        V3 = 20,000*I
Work
        30 = 20,000*I + 20,000*I + V2
        30 = 40,000*I + V2
        30 = 40,000*V2/R  + V2
      30*R = 40,000*V2 + V2*R
 (30-V2)*R = 40,000*V2

         R = (40,000*V2)/(30-V2)

  or
      30*R = (40,000+R)*V2
        V2 = (30*R)/(40,000+R)
```



Ground Faults. The following schematic is used for diagnostic alarms.



125 VDC Over Voltage Limit is the highest allowable DC Bus value, as measured between the positive and negative buses, before a diagnostic alarm is generated.

125 VDC Under Voltage Limit is the lowest allowable DC Bus value, as measured between the positive and negative buses, before a diagnostic alarm is generated.

125 VDC Bus Ground Fault is the absolute value of the minimum voltage allowed on either the positive or negative bus, measured with respect to ground, before a diagnostic alarm is generated.

B-2. TCQB, <Q> CORE

```
                TCQB Card Definition - Socket 2 - Screen 1/10

           TCQB/320 ePROM Revision Information:

              TCQB firmware:
                   Major Rev:    1
                   Minor Rev:    1

               320 firmware:
                   Major Rev:    1
                   Minor Rev:    1
                   Prom Page Select:    0
```

The TCQB multifunction card is located in the <R>, <S>, and <T> cores in location 3.. For Simplex applications, only one card is required for the <R> core. It is an optional card used for extended I/O functions. The TCQB configuration defines the following functions:

- proximitor inputs
- LVDT/LVDR inputs
- 2 regulator milliamp outputs
- LM vibration inputs.

The card is used on Steam Simplex and some TMR application when proximitors are required.

```
                TCQB Card Definition - Socket 2 - Screen 2/10

                  Proximitor shaft location assignments


     Prox  1: HP     Prox  6: HP          Prox 11:  --      Prox 15:  --
     Prox  2: HP     Prox  7: HP          Prox 12:  --      Prox 16:  --
     Prox  3: HP     Prox  8: HP          Prox 13:  --      Prox 17:  --
     Prox  4: HP     Prox  9: --          Prox 14:  --      Prox 18:  --
     Prox  5: HP     Prox 10: --

Notes::    Valid locations:    IP,  HP,  LP
                      not used :    --

Assign channels 1-3 as 'not used' when connecting accelerometers as inputs.
```

Proximitor Types General are RF devices that measure the distance between an object and the device itself, such as the distance between turbine shaft and the proximitor. The device's output is a voltage signal inversely proportional to this distance. The voltage's AC component is interpreted as vibration while changes in the DC component are interpreted as position changes.

Proximitor Shaft Location Assignments specify the input shaft speed to an algorithm. The vibration frequency spectrum is different for the three separate shaft rotation speeds. Sampling is performed at 8 times the fundamental frequency to get first and second order vibration signal components. Valid Selections are:

IP = Intermediate Shaft (LM applications)
HP = High Pressure Shaft
LP = Low Pressure Shaft
-- = Not Used or Used for Accelerometers (channels 1 - 3)

```
               TCQB Card Definition - Socket 2 - Screen 3/10

       Proximitors  (1 thru 9)   :-  Peak to Peak Vibration
                 Transducer Sensitivity
                  (0 to 35.747 V/mil)
                 ---------------------
    Proximitor 1:        0.
    Proximitor 2:        0.
    Proximitor 3:        0.
    Proximitor 4:        0.
    Proximitor 5:        0.
    Proximitor 6:        0.
    Proximitor 7:        0.
    Proximitor 8:        0.
    Proximitor 9:        0.

    Proximitor 1 thru 18 diag. low  limit: (0 to 18v)     4.0
    Proximitor 1 thru 18 diag. high limit: (0 to 18v)     17.0
 Note: Set Sensitivity to 0 to disable diagnostics when input is not used.


             TCQB Card Definition - Socket 2 - Screen 4/10

       Proximitors  (10 thru 18)   :-  Peak to Peak Vibration

                 Transducer Sensitivity
                   (0 to 35.747 V/mil)
                  --------------------
    Proximitor 10:       0.
    Proximitor 11:       0.
    Proximitor 12:       0.

    Proximitor 13:       0.
    Proximitor 14:       0.
    Proximitor 15:       0.

    Proximitor 16:       0.
    Proximitor 17:       0.
    Proximitor 18:       0.
```

Peak to Peak Vibration. The Proximitors indicate vibration by the AC signal component. The vibration inputs measure only this AC component.

Transducer Sensitivity, expressed in Volts/mil of vibration, is a function of the transducer. The transducer types are either generic Bently Nevada Position/Proximitor Probes or Differential Expansion Probes. The Gas Turbine department specifies these values. Typical values for these transducers are:

- Generic Vib/Position Probes .2 Volts/Mil
- Differential Expansion .1 Volts/Mil

Proximitor Low Limit Diag & Proximitor High Limit Diag. The Bently Nevada proximitors have an operating range of 0 to 20 V. However, the probes are only linear from 2 to 16 V. Any value out of this range is non-linear and the device is considered failed when used to sense vibration.

```
TCQB Card Definition - Socket 2 - Screen 5/10

            Proximitors  (19 thru 28)   :-  Position Inputs
                       Transducer Sensitivity        Position Offset
                        (0 to 19.1345 V/mil)          (0 to 2048 mils)
                        --------------------         ---------------
     Proximitor  19:        0.2                         45.0
     Proximitor  20:        0.2                         45.0
     Proximitor  21:        0.2                         45.0
     Proximitor  22:        0.2                         45.0
     Proximitor  23:        0.2                         45.0

     Proximitor  24:        0.2                         45.0
     Proximitor  25:        0.                          0.0
     Proximitor  26:        0.                          0.0
     Proximitor  27:        0.                          0.0
     Proximitor  28:        0.                          0.0

     Proximitor 19 thru 28 diag. low  limit: (0 to 18v)     4.0
     Proximitor 19 thru 28 diag. high limit: (0 to 18v)     17.0
Note: Set Sensitivity to 0 to disable diagnostics when input is not used.
```

Position Inputs measure the output signal's dc component. A change in the dc component indicates change in position.

Transducer Sensitivity is described on the previous page. The Gas Turbine department specifies these values.

Position Offset specifies the starting distance between the proximitor and the device being measured. The proximitor is installed at a 45 Mil distance from the monitored object under ideal conditions. However, this is not always possible or exact due to the method used to install these devices. The constant zeroes this measurement.

Proximitor Low and High Limit Diagnostics are described above.

```
                TCQB Card Definition - Socket 2 - Screen 6/10

                     Proximitors  29 and 30
                        Keyphasor inputs

                      Transducer Sensitivity       Position Offset
                        (0 to 18.333 V/mil)        (0 to 2048 mils)
                       --------------------        ----------------
     Proximitor  29:          0.02292                    45.0
     Proximitor  30:          0.                         0.0
     Proximitor 29 & 30 diag. low  limit:  (0 to 18.333)  0.15
     Proximitor 29 & 30 diag. high limit:  (0 to 18.333)  0.69
```

Key Phasor inputs are a special proximitor input application. A key phasor is a notch cut in the turbine shaft. The leading edge of the notch is defined as zero degrees. The phase information is used by algorithms in the Mark V to perform Fast Fourier Transforms.

Proximitor Diagnostic Low and High Limits are set to 0 V and 18 V respectively to alarm device failure. These limits can be set out of the device's linear range since the only point of interest is the leading edge of the notch. The key phasor notch may actually be deeper than the total distance the proximitor can measure. Therefore, readings of 0 or 18 V dc, the maximum and minimum distances respectively, are permitted since 0 V indicates the leading edge.

```
                TCQB Card Definition - Socket 2 - Screen 7/10

                      Configuration Information

    Prox Chan 1 to 3 Device type:     PROX  (valid types are:
                                            PROXimitor and ACCELerometer.
                                            Verify configuration jumpers
                                            on TCQB board when change
                                            is made)

    HP rpm max scaling value  :      8192
    IP rpm max scaling value  :      8192
    LP rpm max scaling value  :      8192
```

Prox Channel 1 - 3 Device Type specifies the input types for proximitors 1 - 3. The inputs can either be accelerometers or proximitors. Accelerometers are used on LM applications as vibration inputs. When these channels are used for accelerometers, set them as Not Used on the previous Proximitor Type Screen.

HP, IP, LP RPM Max Scaling Value. The Shaft RPM Max Scaling Values are set to the maximum value of the scale type of the points assigned with it.

```
                TCQB Card Definition - Socket 2 - Screen 8/10

                           Lvdt/ Lvdr Inputs

                           Minimum Voltage      Maximum Voltage
                          (0 to 11.651 Vrms)   (0 to 11.651 Vrms)
                          ------------------   ------------------
     Lvdt/Lvdr    1:          0.7                  3.5
     Lvdt/Lvdr    2:          0.7                  3.5
     Lvdt/Lvdr    3:          0.7                  3.5

     Lvdt/Lvdr    4:          0.7                  3.5
     Lvdt/Lvdr    5:          0.7                  3.5
     Lvdt/Lvdr    6:          0.7                  3.5
```

LVDT/LVDR Inputs (General). Using different technology, an LVDT measures distances in the same manner that a proximitor device does. However, the LVDT/LVDR can measure distances up to 14 inches.

Minimum and Maximum Voltage specifies the input voltages equivalent to 0% and 100% stroke.

```
TCQB Card Definition - Socket 2 - Screen 9/10

                Position Milli-Amp Output Configuration

Milli-Amp Output 1:

   Output used (enables output and basic diagnostics):       YES
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA/ 40 mA) CDB value:    0.0
   Maximum (20 mA/200 mA) CDB value:    100.0


Milli-Amp Output 2:

   Output used (enables output and basic diagnnostics):      YES
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA/ 40 mA) CDB value:    0.0
   Maximum (20 mA/200 mA) CDB value:    100.0
```

Output Used. Yes or No. Enables the output and the diagnostics associated with the circiut.

Suicide Enable - Current Fault. Current fault on Outputs 1 and 2 are individual. Therefore, the choice of suicide enable is application dependent. These are typically not enabled for Simplex applications and are enabled for TMR applications. A current fault is the condition where the actual output differs from the reference by 25% of the requested output's full scale. This condition alarms loss of valve control. The suicide enable function allows the control to short the output from an individual processor across a relay contact to drive the current being supplied by that processor to zero and allowing the other two processors to control the output.

Control Data Base (CDB) Variable Full Scale is set to the maximum value of the scale type of the point assigned.

Maximum and Minimum CDB Values are used to calibrate the milliamp output to the desired range.
Minimum Value 4 mA/ 40 mA
Maximum Value 20 mA /200 mA

```
                TCQB Card Definition - Socket 2 - Screen 10/10

                      LM Vibration Inputs

Accelerometer Inputs:      Transducer Sensitivity     Position Offset
                             (0 to 0.5882 Vp/ips)       (0 to 8.000 ips)
                           ---------------------     ---------------
    Transducer 1 HP:           0.2                        0.0
    Transducer 1 IP:           0.2                        0.0
    Transducer 1 LP:           0.2                        0.0

    Transducer 2 HP:           0.2                        0.0
    Transducer 2 IP:           0.2                        0.0
    Transducer 2 LP:           0.2                        0.0

    Transducer 3 HP:           0.2                        0.0
    Transducer 3 IP:           0.2                        0.0
    Transducer 3 LP:           0.2                        0.0
```

Normalized Scaling Gain, Offset, and Shift require definition by Mark V development for future application.

B-3. TCDA

The TCDA digital I/O cards associated with <Q> are located in the <QD1>, and <QD2> cores.

-TCDA_1 is associated with <QD1>
-TCDA_2 is assiciated with <QD2>

A Simplex <QD1> core has one TCDA_1 card in location 1, as there is only the one controller <R>. A TMR <QD1> core has three TCDA_1 cards, one for <R> in location 1, one for <S> in location 2, and one for <T> in location 3. The same rules apply to <QD2>.

The TCDA cards are the interface for contact inputs and relay outputs. The I/O configurator is used to set two aspects of the TCDA contact inputs:

- inversion masks
- change detect

B-3.1. <QD1> Core

TCDA_1 Card Definition - Socket 4 - Screen 1/3

ePROM Revision Information:

Major Rev: 3

Minor Rev: 1

TCDA_1 Card Definition - Socket 4 - Screen 2/3

Contact Input Assignments - <QD1> DTBA points 1 thru 92
& <QD1> DTBB points 1 thru 4

Contact Inputs:	1-10			11-20			21-30			31-40			41-48		
	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det
CI x1:	1	1	1	21	1	1	41	1	1	61	1	1	81	1	1
CI x2:	3	1	1	23	1	1	43	1	1	63	1	1	83	1	1
CI x3:	5	1	1	25	0	1	45	1	1	65	1	1	85	1	1
CI x4:	7	1	1	27	1	1	47	1	1	67	1	1	87	1	1
CI x5:	9	1	1	29	1	1	49	1	1	69	1	1	89	1	1
CI x6:	11	1	1	31	0	1	51	1	1	71	1	1	91	1	1
CI x7:	13	0	1	33	0	1	53	1	1	73	0	1	DTBB1	1	1
CI x8:	15	1	1	35	1	1	55	1	1	75	0	1	DTBB3	1	1
CI x9:	17	1	1	37	1	1	57	1	1	77	0	1			
CI x0:	19	1	1	39	1	1	59	1	1	79	1	1			

TCDA_1 Card Definition - Socket 4 - Screen 3/3

Contact Input Assignments - <QD1> DTBB points 5 thru 100

Contact Inputs:	49-60			61-70			71-80			81-90			91-96		
	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det
CI49:	5	1	1												
CI50:	7	1	1												
CIx1:	9	1	1	29	1	1	49	1	1	69	1	1	89	1	1
CIx2:	11	1	1	31	1	1	51	1	1	71	1	1	91	1	1
CIx3:	13	1	1	33	1	1	53	1	1	73	1	1	93	1	1
CIx4:	15	0	1	35	0	1	55	1	1	75	1	1	95	1	1
CIx5:	17	0	1	37	0	1	57	1	1	77	1	1	97	1	1
CIx6:	19	0	1	39	1	1	59	1	1	79	1	1	99	1	1
CIx7:	21	0	1	41	1	1	61	1	1	81	1	1			
CIx8:	23	1	1	43	1	1	63	1	1	83	1	1			
CIx9:	25	0	1	45	1	1	65	1	1	85	1	1			
CIx0:	27	0	1	47	1	1	67	1	1	87	1	1			

Inversion Masks (Inv Msk). The logic points associated with the hardware points are inverted when the inversion masks are set to a "1". When this is the case, an open input (a "0") results in setting the logic point to "1". This is the default for fail-safe tripping and alarming. When the point is used, the scale type for the associated software point determines its inversion. CIM receives a "0" inversion, and CIM_I receives a "1" inversion.

Change Cell Detect masks (Chg Det) are defaulted to a "1". This enables the point to be detected by the system when it toggles. This information can then be sent to a printer, a file, or elsewhere. The field or customer can disable this feature for a point that toggles frequently by setting its change cell mask to a "0".

B-3.2. <QD2> Core

TCDA_2 Card Definition - Socket 5 - Screen 1/3

ePROM Revision Information:

Major Rev: 1

Minor Rev: 1

TCDA_2 Card Definition - Socket 5 - Screen 2/3

Contact Input Assignments - <QD2> DTBA points 1 thru 92
& <QD2> DTBB points 1 thru 4

Contact Inputs:	1-10			11-20			21-30			31-40			41-48		
	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det
CI x1:	1	1	1	21	1	1	41	1	1	61	1	1	81	1	1
CI x2:	3	1	1	23	1	1	43	1	1	63	1	1	83	1	1
CI x3:	5	1	1	25	1	1	45	1	1	65	1	1	85	1	1
CI x4:	7	1	1	27	1	1	47	1	1	67	1	1	87	1	1
CI x5:	9	1	1	29	1	1	49	1	1	69	1	1	89	1	1
CI x6:	11	1	1	31	1	1	51	1	1	71	1	1	91	1	1
CI x7:	13	1	1	33	1	1	53	1	1	73	1	1	DTBB1	1	1
CI x8:	15	1	1	35	1	1	55	1	1	75	1	1	DTBB3	1	1
CI x9:	17	1	1	37	1	1	57	1	1	77	1	1			
CI x0:	19	1	1	39	1	1	59	1	1	79	1	1			

TCDA_2 Card Definition - Socket 5 - Screen 3/3

Contact Input Assignments - <QD2> DTBB points 5 thru 100

Contact Inputs:	49-60			61-70			71-80			81-90			91-96		
	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det
CI49:	5	1	1												
CI50:	7	1	1												
CIx1:	9	1	1	29	1	1	49	1	1	69	1	1	89	1	1
CIx2:	11	1	1	31	1	1	51	1	1	71	1	1	91	1	1
CIx3:	13	1	1	33	1	1	53	1	1	73	1	1	93	1	1
CIx4:	15	1	1	35	1	1	55	1	1	75	1	1	95	1	1
CIx5:	17	1	1	37	1	1	57	1	1	77	1	1	97	1	1
CIx6:	19	1	1	39	1	1	59	1	1	79	1	1	99	1	1
CIx7:	21	1	1	41	1	1	61	1	1	81	1	1			
CIx8:	23	1	1	43	1	1	63	1	1	83	1	1			
CIx9:	25	1	1	45	1	1	65	1	1	85	1	1			
CIx0:	27	1	1	47	1	1	67	1	1	87	1	1			

Inversion Masks (Inv Msk). The logic points associated with the hardware points are inverted when the inversion masks are set to a "1". When this is the case, an open input (a "0") results in setting the logic point to "1". This is the default for fail-safe tripping and alarming. When the point is used, the scale type for the associated software point determines its inversion. CIM receives a "0" inversion, and CIM_I receives a "1" inversion.

Change Cell Detect masks (Chg Det) are defaulted to a "1". This enables the point to be detected by the system when it toggles. This information can then be sent to a printer, a file, or elsewhere. The field or customer can disable this feature for a point that toggles frequently by setting its change cell mask to a "0".

B-4. LCCQ, <Q> CORE

```
LCCQ Card Definition - Socket 8 - Screen 1/1

   ePROM Revision Information:

   Major Rev:   1

   Minor Rev:   0
```

B-5. DCCQ, <Q> CORE

The earlier version of software for the DCC card is contained in section B-5.1 and the latest version of the software is contained in section B-5.2

B-5.1 DCCQ, <Q> CORE - Earlier Version

```
DCCQ Card Definition - Socket 12 - Screen 1/6

      ePROM Revision Information:

            Major Rev:   1

            Minor Rev:   1


DCCQ Card Definition - Socket 12 - Screen 2/6

      MegaWatt Input

      MEGAWATT INPUT (Q_Q_MAI16) :

            Offset  (0 to   3 volts)  :     0.0

            Gain    (0 to 256 MW/volt):     50.0
```

```
          DCCQ Card Definition - Socket 12 - Screen 3/6

          Frequency Inputs  (output scaled as %, max = 128%)
                            (for #/sec scaling use next screen)

                                                          Rate Table:
TTL INPUT 1    (Q_Q_TTL1)    :                         -------------------
    Gain       (%/(cnt/scan)):            0.0           0  input not used
    Rate_index    (refer to rate table)   0             1  16 Hz scan rate
                                                        2   8 Hz scan rate
                                                        4   4 Hz scan rate
                                                        8   2 Hz scan rate
                                                       16   1 Hz scan rate

   <  Gain == 100% * scan_rate / (100%_rated_Hz_input)  >
   < i.e., @16hz scan, 1000 hz rated input,  gain = 1.6 >
   < i.e., @ 8hz scan, 1000 hz rated input,  gain = 0.8 >
```

```
          DCCQ Card Definition - Socket 12 - Screen 4/6

          Frequency Inputs  (output scaled as #/sec, max = 64#/sec)
                            (for % scaling, use previous screen)

                                                          Rate Table:
TTL INPUT 1    (Q_Q_TTL1)    :                         -------------------
    Gain          ((#/sec)/(cnt/scan)):   0.0           0  input not used
    Rate_index    (refer to rate table)   0             1  16 Hz scan rate
                                                        2   8 Hz scan rate
                                                        4   4 Hz scan rate
                                                        8   2 Hz scan rate
                                                       16   1 Hz scan rate

   <  Gain == rated_#/sec * scan_rate / (Hz_input_rated_flow) >
   < i.e., @16hz scan, 16 #/sec @ 500 hz input,  gain = 0.512 >
   < i.e., @ 8hz scan, 16 #/sec @ 500 hz input,  gain = 0.256 >
```

```
          DCCQ Card Definition - Socket 12 - Screen 5/6

          Frequency Inputs (output scaled as %, max = 128%)
                           (for #/sec scaling use next screen)
                                                          Rate Table:
TTL INPUT 2    (Q_Q_TTL2)    :                         -------------------
    Gain          (%/(cnt/scan)):         0.0           0  input not used
    Rate_index    (refer to rate table)   0             1  16 Hz scan rate
                                                        2   8 Hz scan rate
                                                        4   4 Hz scan rate
                                                        8   2 Hz scan rate
                                                       16   1 Hz scan rate

   <  Gain == 100% * scan_rate / (100%_rated_Hz_input)  >
   < i.e., @16hz scan, 1000 hz rated input,  gain = 1.6 >
   < i.e., @ 8hz scan, 1000 hz rated input,  gain = 0.8 >
```

```
          DCCQ Card Definition - Socket 12 - Screen 6/6

          Frequency Inputs  (output scaled as #/sec, max = 64#/sec)
                            (for % scaling, use previous screen)

                                                               Rate Table:
 TTL INPUT 2    (Q_Q_TTL2)    :                             -------------------
     Gain          ((#/sec)/(cnt/scan)):   0.0               0  input not used
     Rate_index    (refer to rate table)   0                 1  16 Hz scan rate
                                                             2   8 Hz scan rate
                                                             4   4 Hz scan rate
                                                             8   2 Hz scan rate
                                                            16   1 Hz scan rate

    <  Gain == rated_#/sec * scan_rate / (Hz_input_rated_flow) >
    < i.e., @16hz scan, 16 #/sec @ 500 hz input,  gain = 0.512 >
    < i.e., @ 8hz scan, 16 #/sec @ 500 hz input,  gain = 0.256 >
```

B-5.2 DCCQ, <Q> CORE - Latest Version

```
                  DCCq Card Definition - Socket 12 - Screen 1/2


              GASQ ePROM Revision Information:

                        Major Rev:   7

                        Minor Rev:   2


                  DCCq Card Definition - Socket 12 - Screen 2/2

               MegaWatt and Frequency Inputs

 MEGAWATT INPUT (Q_Q_MAI16) :
        Offset  (0 to   3 volts)  :     3.0
        Gain    (0 to 256 MW/volt):     100.0


 TTL INPUT 1    (Q_Q_TTL1)
        Pulses/second at 100 % rated input (% scale):   4000.1
       [Pulses/second at 50 #/sec flow (#/sec scale)]

 TTL INPUT 2    (Q_Q_TTL2)
        Pulses/second at 100 % rated input (% scale):   4000.1
       [Pulses/second at 50 #/sec flow (#/sec scale)]
```

B-6. IOMA, <Q> CORE

```
IOMA Card Definition - Socket 13 - Screen 1/4

      IOMA/320 ePROM Revision Information:

      IOMA firmware:

            Major Rev:        3
            Minor Rev:        1

      320 firmware:

            Major Rev:        1
            Minor Rev:        1
            Prom Page Select: 0


IOMA Card Definition - Socket 13 - Screen 2/4

      Pulse Rate Definitions

               100% Rated
            Pulse Frequency

PN:                   0

LP2:                  0
HP2:                  0

IP1                   25600.0
IP2                   25600.0
```

Later versions of the IOMA software have the following alternate second screen. Note that the pulse rate input for PN has been moved positionally.

```
IOMA Card Definition - Socket 13 - Screen 2/4

                        Pulse Rate Definitions

                               100% Rated
                            Pulse Frequency


              Pulse Rate 5:          0

              Pulse Rate 6:          0

              Pulse Rate 7 (PN):     0

            IP1                        4096.0
            IP2                        4096.0
```

```
          IOMA Card Definition - Socket 13 - Screen 3/4

                      CPD  Definitions

  CPD transducer - offset                       (volts):   -0.001
  CPD transducer - gain                      (psi/volt):   40.01
  CPD transducer - Max CDB value           (power of 2):   2048

  CPD pressure low limit - alarm and
     compressor stall detection disable (0 to 512 psi):   12.5

  CPD drop threshold gain                      (0 to 1):   0.87
  CPD drop threshold intercept          (0 to 1024 psi):   3.4

  CPD rate threshold clamp          (0 to 4096 psi/sec):   100.0
  CPD rate threshold gain                  (0 to 8 /sec):   4.0001
  CPD rate threshold intercept           (0 to 512 psi):   25.0
```

```
          IOMA Card Definition - Socket 13 - Screen 4/4

                    Miscellaneous Definitions

     Bus Voltage target at 115v input         (0 to 64KV)  13.8
     Generator Voltage target at 115v input   (0 to 64KV)  13.8

     Nominal Bus Frequency            (0 to 64):        60
     Sync Minimum Voltage             (0 to 64KV):      11.0

     Amplitude Diff Limit             (0 to 64KV):      1.38
     Phase Diff Limit                 (0 to 512 deg):   10.0
     Frequency Diff Limit             (0 to 8 hz):      0.27

     Bus frequency - Max CDB value         (power of 2):     64
     Sync frequency error  - Max CDB value (power of 2):     64
     Sync phase error - Max CDB value      (power of 2):     512
```

Later versions of the IOMA software contain the following alternate forth screen for configuration information.

```
                    IOMA Card Definition - Socket 13 - Screen 4/4

                       Miscellaneous Definitions

   Bus PT secondary volts at rated (100%) primary volts (60 to 140 V) 115.0
   Generator PT sec volts at rated (100%) primary volts (60 to 140 V) 115.0

   Nominal Bus Frequency            (44 to 64):       60

   Sync Minimum Voltage           (10 to 128%):       50.0
   Amplitude Diff Limit             (0 to 50%):       10.0

   Phase Diff Limit               (0 to 30 deg):      10.0
   Frequency Diff Limit          (0 to 0.33 hz):      0.30
```

B-7. TCEA, <P> CORE

The three TCEA cards are in the MarkV <P> Protective Core, and are called <X>, <Y>, and <Z> individually. Both TMR and Simplex panels have all three TCEA cards. TMR applications configure only TCEA_1, as each controller <R>, <S>, and <T> view only one card each. Simplex applications configure TCEA_1, TCEA_2, and TCEA_3, as all three cards are seen and voted in <R>. The TCEA card configuration defines the following functions:

- Breaker and Frequency Shift Definitions
- Flame Detect Thresholds
- Overspeed Trip configuration

The PTBA termination board is used to connect the input signal wires for the <P> Core functions. There are no requisition variable hardware jumpers to set on the PTBA termination card.

PTBA Termination Points for Speed Sensors		
	HP	LP
X	1,2	3,4
Y	5,6	7,8
Z	9,10	11,12

PTBA Termination Points for Flame Detectors

Flame Det	PTBA Term
1	13, 14
2	15, 16
3	17, 18
4	19, 20
5	21, 22
6	23, 24
7	25, 26
8	27, 28

The earlier version of software for the TCEA cards is contained in section B-7.1 and the latest version of the software is contained in section B-7.2.

B-7.1 TCEA - Earlier Version

```
        TCEA_1 Card Definition - Socket 15 - Screen 1/5

              ePROM Revision Information:

                    Major Rev:   1

                    Minor Rev:   1


        TCEA_1 Card Definition - Socket 15 - Screen 2/5


        Breaker and Frequency Shift Definitions


  Breaker 1 close time:         5    cycles (0 - 255)
  Breaker 2 close time:         5    cycles (0 - 255)
  Breaker adjust limit:         3    cycles (0 - 255)

  Breaker self adapt correction on?     YES


  Scaling Frequency shift:   ( n =      0    )

     where max scaling frequency = 64 hz * 2_
```

Breaker 1 close time defines the number of cycles required between the time the breaker closure is initiated from the Mark V controller to the time the breaker actually closes. The constant is specified in cycles so it is independent of the turbine/generator set frequency. The constant is necessary for breaker closure to occur at approximately zero phase difference between the generator and the system bus for a zero current flow through the breaker at closure time. The Turbine vendor specifies the value since the constant is dependent on breaker type.

Breaker 2 close time performs the same function as breaker 1 close time for a second breaker if required.

Breaker adjust limit. The Mark V synchronizing function adjusts the Breaker Close Times defined above by comparing the time the breaker closure signal was sent from the MK5 to the time the breaker actually closed, as the actual time may change with the age of the breaker. The Breaker Adjust Limit constant specifies the limit the control is allowed to adjust the Breaker Close Time constants. The adjustments are made in increments of one. For the values above, the initial close time is set at 5 cycles. If the actual required close time becomes 8 cycles, the Mark V controller adjust the 5 to an 8, taking three closures to make this change. If the time required becomes 9 cycles, the Mark V does not make another adjustment, but leaves it at 8 and will generate a diagnostic alarm. The Turbine vendor specifies the constant.

Breaker self adaptive correction enables the Mark V adaptive control function. The majority of applications will use this function. The Turbine vendor should make the final decision if there is a question.

Scaling frequency shift is used in internal calculations based on the Maximum Frequency Scaling. A shift of 0 corresponds to a max scale of 64Hz. The number can change only if internal changes are done to the Mark V database and is therefore set by the system designers at GEDS.

```
                TCEA_1 Card Definition - Socket 15 - Screen 3/5

                 Flame Detect Threshold Values

                          Detection Level A     Detection Level B
                          Values (0 to 255)     Values (0 to 255)
                          ----------------      -----------------
       Flame detect 1:          2                       4
       Flame detect 2:          2                       4
       Flame detect 3:          2                       4
       Flame detect 4:          2                       4

       Flame detect 5:          2                       4
       Flame detect 6:          2                       4
       Flame detect 7:          2                       4
       Flame detect 8:          2                       4

                         Detection level values are flame detector pulses
                         during the 1/16 second sample period.
```

Flame Detect Threshold is based on the number of pulses generated by the flame detectors in 1/16th of a second. A value of 2 in this field is equivalent to a 32 Hz signal from the flame detectors. Normal operation produces a 300 Hz signal, or about 18 pulses in 1/16th of a second.

The detection levels A and B are both low limits that set the number of pulses per 1/16th of a second and indicate flame is present. Two Mark V logic signals determine which of the two sets of limits to use, one for detectors 1-4 and one for detectors 5-8. Set Detection Level A to 2 and Detection Level B to 4. Individual applications, such as Dry Low Nox, may require other values. Feedback from the field is required to change these numbers.

```
               TCEA_1 Card Definition - Socket 15 - Screen 4/5

          Configuration Constants                System     Board Jumpers:
                                                 Config.    J29     J28
 Trip Board Id  (med, gas, large)   :   gas      LM          0       1
 System Configuration (LM, SMX, ---): ---        SMX         1       0
 Trip Anticipation ? (YES, NO)      :  NO        ___         1       1
    <large steam only>
                                          index  Base_Freq  J15 J14 J13 J12
 HP base_speed index:  5                    1    3000 hz     0   0   0   1
  (100 % design speed                       2    3600 hz     0   0   1   0
    frequency input)                        3    4800 hz     0   0   1   1
                                            4    4860 hz     0   1   0   0
                                            5    5100 hz     0   1   0   1
                                            6    7100 hz     0   1   1   0
                                            7    7500 hz     0   1   1   1
                                            8    7833 hz     1   0   0   0
 HP overspeed_index:   44      (0 to 63 as sum of jumper values where:
                               J21=32, J20=16, J19=8, J18=4, J17=2, J16=1)
  < overspeed frequency setting = 100% + (0.25 * overspeed_index)% >
```

Trip Board ID sets the trip board type used by the system. The trip board connects to the TCEA card and configures it with fixed ID jumpers. The constant must match the trip board type. Allowable values include:

Constant	Application	Trip Board
med	Medium and Fitchburg Steam	TCTS
gas	All Gas Turbines	TCTG
large	Large Steam Turbines	TCTL

System configuration sets the application type. The system configuration corresponds to hardware jumper settings. If the hardware jumpers do not match the system configuration constant, a diagnostic alarm is generated. Valid settings and their corresponding hardware settings are as follows:

	Hardware Jumpers				
Constant	J28	J29	L3LM	L3SMX	Logic Signal Application
LM	1	0	1	0	LM Gas Turbine
SMX	0	1	0	1	Simplex
---	1	1	0	0	All other

Trip Anticipation function monitors the turbine's acceleration. The acceleration can be so high that the turbine reaches and exceeds the overspeed point in an extremely short time. By the time the controller issues the trip and the valves react accordingly, the turbine may already have been damaged. When this high acceleration rate is detected and this function is enabled, the Mark V issues a trip instead of waiting for the overspeed detection. Trip Anticipate is used exclusively on large steam turbines.

HP base speed index specifies the HP shaft design speed. The rotor design speed is fixed and is independent of operating speed. For example, a 5001 is designed to run at 5100 RPM. But on certain gear box combinations the actual operating speed is 5105 RPM. Therefore, to the controller, 100% speed is 5105. However, to the TCEA card 100% speed is 5100 RPM. Additionally, there are a finite set of speeds that are acceptable. The constant must be equivalent to the hardware jumper settings or a diagnostic alarm will be generated. The results are posted in the Mark V variable TNH_PC. Allowable constant settings and hardware jumper positions are:

	Hardware Jumpers				
Constant	J15	J14	J13	J12	Design Speed
1	0	0	0	1	3000 MS9001
2	0	0	1	0	3600 MS7001 Med
3	0	0	1	1	4800 Large
4	0	1	0	0	4660
5	0	1	0	1	5100 MS5001 MS6001
6	0	1	1	0	7100 M53002
7	0	1	1	1	7500
8	1	0	0	0	7833

HP overspeed index is the rated DESIGN speed percentage that the rotor is allowed to reach before an overspeed condition is annunciated. To determine the percent overspeed, simply divide the number of counts by 4 (1 CNT = 1/4 %). Additionally, the control constant must match hardware jumper settings. The hardware jumpers are J16, J17, J18, J19, J20, and J21 where J16 is the LSB. For hardware jumpers settings see Appendix A.

```
              TCEA_1 Card Definition - Socket 15 - Screen 5/5

                 LP Configuration Constants

                                                   Board Jumpers:
   LP base_speed index:  0      index  Base_Freq  J11 J10 J9  J8
    (100 % design speed           1    3300 hz     0   0   0   1
      frequency input)            2    4670 hz     0   0   1   0
                                  3    4980 hz     0   0   1   1
                                  4    6500 hz     0   1   0   0

                                  0    NOT USED    0   0   0   0
                                       (Single shaft, LP trip disabled)

   LP overspeed_index:   44      (0 to 63 as sum of jumper values where:
                                 J27=32, J26=16, J25=8, J24=4, J23=2, J22=1)

     < overspeed frequency setting = 100% + (0.25 * overspeed_index)% >
```

LP base_speed index specifies the LP shaft design speed. The LP shaft design speed is fixed and is independent of the operating speed. Therefore, 100% operating speed may or may not correspond to 100% design speed. Additionally, there are a finite number of speeds acceptable. The constant must be equivalent to the hardware jumper settings or a diagnostic alarm is generated. The results are posted in the TNL_PC Mark V variable. Allowable constant settings and Hardware Jumper positions are as follows:

	Hardware Jumpers				
Constant	J11	J10	J9	J8	Design Speed
0	0	0	0	0	NO TRIP
1	0	0	0	1	3600
2	0	0	1	0	4670 MS5002
3	0	0	1	1	4980
4	0	1	0	0	6500 MS3002

LP overspeed_index is the design speed percentage the rotor reachs before an overspeed condition is annunciated. To determine the percent overspeed, simply divide the number of counts by 4 (1 CNT = 1/4 %). Additionally, the control constant must match hardware jumper settings. The hardware jumpers are J22, J23, J24, J25, J26, and J27 where J22 is the LSB. For example, to set an LP Overspeed index equal to 10%, the constant and hardware jumpers would be as follows:

	Hardware Jumpers						
Constant	J27	J26	J25	J24	J23	J22	Percent
40	1	0	1	0	0	0	10%

To determine the Hardware jumper settings convert the decimal constant into its binary equivalent, with J22 as the LSB.

```
              TCEA_2 Card Definition - Socket 16 - Screen 1/5

                   ePROM Revision Information:

                        Major Rev:   1

                        Minor Rev:   1
```

See notes for TCEA_1

```
              TCEA_2 Card Definition - Socket 16 - Screen 2/5

              Breaker and Frequency Shift Definitions

    Breaker 1 close time:         5    cycles (0 - 255)
    Breaker 2 close time:         5    cycles (0 - 255)
    Breaker adjust limit:         3    cycles (0 - 255)

    Breaker self adapt correction on?     YES

    Scaling Frequency shift:   ( n =      0    )
       where max scaling frequency = 64 hz * 2_
```

See Notes for TCEA_1

```
              TCEA_2 Card Definition - Socket 16 - Screen 3/5

               Flame Detect Threshold Values

                        Detection Level A     Detection Level B
                        Values (0 to 255)     Values (0 to 255)
                        ----------------      -----------------
     Flame detect 1:          2                      4
     Flame detect 2:          2                      4
     Flame detect 3:          2                      4
     Flame detect 4:          2                      4

     Flame detect 5:          2                      4
     Flame detect 6:          2                      4
     Flame detect 7:          2                      4
     Flame detect 8:          2                      4

                       Detection level values are flame detector pulses
                       during the 1/16 second sample period.
```

See Notes for TCEA_1

```
              TCEA_2 Card Definition - Socket 16 - Screen 4/5

         Configuration Constants                System     Board Jumpers:
                                                Config.    J29     J28
Trip Board Id  (med, gas, large)   :   gas      LM          0       1
System Configuration (LM, SMX, ---): SMX        SMX         1       0
Trip Anticipation ? (YES, NO)      :  NO        ___         1       1
   <large steam only>
                                         index  Base_Freq  J15 J14 J13 J12
HP base_speed index:  5                    1    3000 hz     0   0   0   1
 (100 % design speed                       2    3600 hz     0   0   1   0
   frequency input)                        3    4800 hz     0   0   1   1
                                           4    4860 hz     0   1   0   0
                                           5    5100 hz     0   1   0   1
                                           6    7100 hz     0   1   1   0
                                           7    7500 hz     0   1   1   1
                                           8    7833 hz     1   0   0   0
HP overspeed_index:   44      (0 to 63 as sum of jumper values where:
                              J21=32, J20=16, J19=8, J18=4, J17=2, J16=1)
 < overspeed frequency setting = 100% + (0.25 * overspeed_index)% >
```

See Notes for TCEA_1

```
              TCEA_2 Card Definition - Socket 16 - Screen 5/5

                 LP Configuration Constants

                                                 Board Jumpers:
   LP base_speed index:  0      index  Base_Freq  J11 J10 J9  J8
    (100 % design speed           1    3300 hz     0   0   0   1
      frequency input)            2    4670 hz     0   0   1   0
                                  3    4980 hz     0   0   1   1
                                  4    6500 hz     0   1   0   0

                                  0    NOT USED    0   0   0   0
                                       (Single shaft, LP trip disabled)

   LP overspeed_index:   44      (0 to 63 as sum of jumper values where:
                                 J27=32, J26=16, J25=8, J24=4, J23=2, J22=1)

     < overspeed frequency setting = 100% + (0.25 * overspeed_index)% >
```

See Notes for TCEA_1

```
              TCEA_3 Card Definition - Socket 17 - Screen 1/5

                   ePROM Revision Information:

                        Major Rev:   1

                        Minor Rev:   1
```

See Notes for TCEA_1

```
              TCEA_3 Card Definition - Socket 17 - Screen 2/5

              Breaker and Frequency Shift Definitions


     Breaker 1 close time:         5    cycles (0 - 255)
     Breaker 2 close time:         5    cycles (0 - 255)
     Breaker adjust limit:         3    cycles (0 - 255)

     Breaker self adapt correction on?     YES


     Scaling Frequency shift:   ( n =      0    )

        where max scaling frequency = 64 hz * 2_
```

See Notes for TCEA_1

```
              TCEA_3 Card Definition - Socket 17 - Screen 3/5

               Flame Detect Threshold Values

                        Detection Level A     Detection Level B
                        Values (0 to 255)     Values (0 to 255)
                        ----------------      -----------------
      Flame detect 1:         2                      4
      Flame detect 2:         2                      4
      Flame detect 3:         2                      4
      Flame detect 4:         2                      4

      Flame detect 5:         2                      4
      Flame detect 6:         2                      4
      Flame detect 7:         2                      4
      Flame detect 8:         2                      4

                       Detection level values are flame detector pulses
                       during the 1/16 second sample period.
```

See Notes for TCEA_1

```
              TCEA_3 Card Definition - Socket 17 - Screen 4/5

         Configuration Constants                System     Board Jumpers:
                                                Config.    J29     J28
 Trip Board Id  (med, gas, large)   :   gas     LM          0       1
 System Configuration (LM, SMX, ---): SMX       SMX         1       0
 Trip Anticipation ? (YES, NO)      :  NO       ___         1       1
    <large steam only>
                                         index  Base_Freq  J15 J14 J13 J12
 HP base_speed index:  5                   1    3000 hz     0   0   0   1
  (100 % design speed                      2    3600 hz     0   0   1   0
    frequency input)                       3    4800 hz     0   0   1   1
                                           4    4860 hz     0   1   0   0
                                           5    5100 hz     0   1   0   1
                                           6    7100 hz     0   1   1   0
                                           7    7500 hz     0   1   1   1
                                           8    7833 hz     1   0   0   0
 HP overspeed_index:   44      (0 to 63 as sum of jumper values where:
                               J21=32, J20=16, J19=8, J18=4, J17=2, J16=1)
  < overspeed frequency setting = 100% + (0.25 * overspeed_index)% >
```

See Notes for TCEA_1

```
              TCEA_3 Card Definition - Socket 17 - Screen 5/5

                 LP Configuration Constants

                                                  Board Jumpers:
    LP base_speed index:  0      index  Base_Freq  J11 J10 J9  J8
     (100 % design speed           1    3300 hz     0   0   0   1
       frequency input)            2    4670 hz     0   0   1   0
                                   3    4980 hz     0   0   1   1
                                   4    6500 hz     0   1   0   0

                                   0    NOT USED    0   0   0   0
                                        (Single shaft, LP trip disabled)

    LP overspeed_index:   44      (0 to 63 as sum of jumper values where:
                                  J27=32, J26=16, J25=8, J24=4, J23=2, J22=1)

      < overspeed frequency setting = 100% + (0.25 * overspeed_index)% >
```

See Notes for TCEA_1

B-7.2. TCEA - Later Version

```
              TCEA_1 Card Definition - Socket 15 - Screen 1/6

                   ePROM Revision Information:
                        Major Rev:   4
                        Minor Rev:   1
```

Notes: See Section B-7.1.

```
              TCEA_1 Card Definition - Socket 15 - Screen 2/6
                      Generator Breaker Definitions
Gen Breaker 1 close time:                    5    cycles (0 - 255)
Gen Breaker 1 close time diagnostic enable?   NO
Gen Breaker 2 close time:                    5    cycles (0 - 255)
Gen Breaker 2 close time diagnostic enable?   NO
Breaker adaptive adjustment limit:           3    cycles (0 - 255)
Breaker self adapt correction enable?        YES
Gen Panel 125 vdc voltage diagnostic enable?  NO
```

Gen Panel 125 vdc voltage diagnostic enable controls the activity of a diagnostic alarm generated upon the loss of availability of generator breaker close voltage. This diagnostic is typically not enabled for mechanical or compresor drive applications.

For notes on the other settings, see Section B-7.1.

```
              TCEA_1 Card Definition - Socket 15 - Screen 3/6

               Flame Detect Threshold Values

                        Detection Level A     Detection Level B
                        Values (0 to 255)     Values (0 to 255)
                        ----------------      -----------------
      Flame detect 1:         2                      4
      Flame detect 2:         2                      4
      Flame detect 3:         2                      4
      Flame detect 4:         2                      4

      Flame detect 5:         2                      4
      Flame detect 6:         2                      4
      Flame detect 7:         2                      4
      Flame detect 8:         2                      4

                       Detection level values are flame detector pulses
                       during the 1/16 second sample period.
```

For notes on these settings, see Section B-7.1.

```
              TCEA_1 Card Definition - Socket 15 - Screen 4/6

         Configuration Constants

   Trip Board Id                       (med, gas, large):      large
   System Configuration                   (LM, SMX, ---):      ---
   Trip Anticipation Enabled?         <large steam only>:       NO

   HP input frequency for 100 percent rated speed:             4800

   HP input frequency for overspeed trip setting:              5352

   HP Overspeed Trip Board Jumper Positions (derived from trip frequency):
        J21   J20      J19   J18   J17   J16      J15   J14    J13   J12
         0     1        0     1     0     0        1     1      1     0

   System Configuration Board Jumper Positions (derived from Sys Config):
                       J29   J28
                        1     1
```

Trip Board ID - See section B-7.1.

System Configuration - See section B-7.1.

Trip Anticipation - See section B-7.1.

HP input frequency for 100 percent rated speed - Enter the pulse frequency from the magnetic pickups for 100% rated speed. For example, if 100% rated speed is 1800 RPM and the pickup wheel has 120 teeth, the correct value is:

```
1800 RPM x 120 pulses/revolution / 60 seconds/minute = 3600 pulses/second
```

HP input frequency for overspeed trip setting - Enter the pulse frequency for the emergency overspeed trip setpoint. For the example above, if the emergency overspeed setting was to be 110% of rated speed, then the value should be set to 3600 times 110%, or 3960 pulses/second.

HP Overspeed Trip Board Jumper Positions (derived from trip frequency) and **System Configuration Board Jumper Positions (derived from Sys Config)** - Click on VERIFY SCREEN, the <I> will then calculate the jumper settings for the specified overspeed trip setting and system configuration.

```
              TCEA_1 Card Definition - Socket 15 - Screen 5/6

                 LP Configuration Constants


LP input frequency for 100 percent rated speed:              0

LP input frequency for Overspeed Trip Setting:               0
   (0 = NOT USED; Single shaft, LP trip disabled)

HP % speed at which to check for LP shaft turning:           0.0


LP Overspeed Trip Board Jumper Positions (derived from trip frequency):
     J27   J26      J25   J24   J23   J22      J11   J10   J9    J8
      0     0        0     0     0     0        0     0     0     0
```

HP % speed at which to check for LP shaft turning - This is the % of the HP shaft speed at which the LP shaft should be turning. If it is not turning at this speed, a trip and alarm will occur.

For notes on the other settings, see the HP settings definitions in `Screen 4/5` shown earlier.

```
              TCEA_1 Card Definition - Socket 15 - Screen 6/6

                 LP Acceleration Protection
                  (applicable to GAS only)

Enable LP Acceleration protection:  NO
(used in GAS Turbine System ONLY)

Number of LP speed interrupts between acceleration checks: 6
              (each interrupt represents 8 teeth passing)

Number of LP speed interrupts for speed calculation: 9

Minimum LP speed (in Hz) to enable protection: 4150

LP Acceleration Limit (in hz/sec): 5533
```

TCEA_2 and TCEA_3 card definitions are identical to TCEA_1. These cards are configured for Simplex applications only.

B-8. TCCA, <C< CORE

```
TCCA Card Definition - Socket 1 - Screen 1/7

   ePROM Revision Information:

        Major Rev:   2

        Minor Rev:   5
```

The TCCA card is located in the TC2000 <C> core location 2. It is the "Turbine Control <C> Micro Application Card A". The TCCA configuration defines the following functions:

- <C> thermocouples
- <C> RTDs
- <C> milliamp inputs
- <C> cold junction filtering
- shaft voltage and current monitoring.

Sixteen milliamp outputs are supplied on the TCCA card, but their assignments and scaling are defined through the table file MAOUT_Q.SRC, not through the I/O configurator. This file is table compiled and downloaded to <R>, <S>, and <T> to drive the <C> milliaamp outputs.

The TCCA is supplied for indication, communication, and alarming primarily. As data from this card is not redundant.

```
                  TCCA Card Definition - Socket 1 - Screen 2/7

                          Thermocouple Type Selection

TC  1:  J      TC 10:  J      TC 19:  J       TC 28:  J      TC 37:  J
TC  2:  J      TC 11:  J      TC 20:  J       TC 29:  J      TC 38:  J
TC  3:  J      TC 12:  J      TC 21:  J       TC 30:  J      TC 39:  J

TC  4:  J      TC 13:  J      TC 22:  J       TC 31:  J      TC 40:  J
TC  5:  J      TC 14:  J      TC 23:  J       TC 32:  J      TC 41:  J
TC  6:  J      TC 15:  J      TC 24:  J       TC 33:  J      TC 42:  J

TC  7:  J      TC 16:  J      TC 25:  J       TC 34:  J
TC  8:  J      TC 17:  J      TC 26:  J       TC 35:  J
TC  9:  J      TC 18:  J      TC 27:  J       TC 36:  J

                          Valid types:    K, J, E, T
                            not used :    -
```

Valid types:
K = Chromel-Alumel
J = Iron-Constantan
E = Chromel-Constantan
T = Copper Constantan

Thermocouples are set at type K (Chromel-Alumel) for Gas, and type E (Chromel-Constantan) for Steam. Older gas units (retrofits) may require type J (Iron-Constantan). The last valid type is T (Copper-Constantan). Not used is noted by one dash: "-".

```
                   TCCA Card Definition - Socket 1 - Screen 3/7

                    RTD Type Selection
                                                          Type Description:(at 0°C)
 RTD  1: SAMA   RTD 11: SAMA   RTD 21: SAMA     -------------------------
 RTD  2: SAMA   RTD 12: SAMA   RTD 22: SAMA     CU10  - 9.035 Ω Copper
                                                           (10 Ω at 25° C)
 RTD  3: SAMA   RTD 13: SAMA   RTD 23: SAMA     DIN   - 100 Ω Platinum,
 RTD  4: SAMA   RTD 14: SAMA   RTD 24: SAMA                 α = 0.00385
                                                PURE  - 100 Ω Platinum,
 RTD  5: SAMA   RTD 15: SAMA   RTD 25: SAMA                 α = 0.003926
 RTD  6: SAMA   RTD 16: SAMA   RTD 26: SAMA     SAMA  - 98.129 Ω Platinum,
                                                            α = 0.00385
 RTD  7: SAMA   RTD 17: SAMA   RTD 27: SAMA     USIND - 100 Ω Platinum,
 RTD  8: SAMA   RTD 18: SAMA   RTD 28: SAMA                 α = 0.00391
                                                N120  - 120 Ω Nickel
 RTD  9: SAMA   RTD 19: SAMA   RTD 29:  DIN
 RTD 10: SAMA   RTD 20: SAMA   RTD 30:  DIN     ----  - not used
```

The RTD_types are defaulted to the standard DIN. The actual type required depends on the requisition.

```
                  TCCA Card Definition - Socket 1 - Screen 4/7

                     Milliamp Input Definitions   1 - 8

                       Signal     CDB value      CDB value    CDB value
                       in use     Full Scale      at 4 mA     at 20 mA
                       -------    ----------     ---------    ----------
    Signal  1:            NO        0              0.           0.
    Signal  2:            NO        0              0.           0.

    Signal  3:            NO        0              0.           0.
    Signal  4:            NO        0              0.           0.

    Signal  5:            NO        0              0.           0.
    Signal  6:            NO        0              0.           0.

    Signal  7:            NO        0              0.           0.
    Signal  8:            NO        0              0.           0.
```

The later version of the software contains a screen like the sample shown below:

```
                  TCCA Card Definition - Socket 1 - Screen 4/7

                      Milliamp Input Definitions   1 - 8

               Signal    CDB value    CDB value  CDB value  Enable   Enable
               in use    Full Scale    at 4 mA   at 20 mA   Low diag High diag
               -------   ----------   ---------  ---------  -------- ---------
   Signal  1:    YES        128         0.0        113.8       NO       NO
   Signal  2:    YES        2048        0.0        100.0       NO       NO

   Signal  3:    YES        256         0.0        9.96        NO       NO
   Signal  4:    YES        2048        0.0        2000.0      NO       NO

   Signal  5:    YES        2048        0.0        426.7       NO       NO
   Signal  6:     NO        128         0.0        100.0       NO       NO

   Signal  7:    YES        2048        0.0        100.0       NO       NO
   Signal  8:     NO        128         0.0        100.0       NO       NO
```

Milliamp Input Screen. The inputs are 8-bit D/A with limited resolution (more suitable for alarming and indication).

Signal in Use is set to either Yes or No.

CDB Value Full Scale is set to the max. value of the scale type of the point assigned with it.

CDB Value at 4mA and 20mA are used to calibrate the milliamp input to the desired range. For example, set the minimum at 90% stroke for an input of 4 mA, and to the maximum to 102% stroke for an input of 20mA.

Enable Low and High Diag - Enables the diagnostic alarm for the low and/or high ranges if entered as YES.

```
                    TCCA Card Definition - Socket 1 - Screen 5/7

                        Milliamp Input Definitions   9 - 14


                        Signal     CDB value      CDB value    CDB value
                        in use     Full Scale      at 4 mA     at 20 mA
                        -------    ----------     ---------    ----------
     Signal  9:            NO        0              0.            0.
     Signal 10:            NO        0              0.            0.

     Signal 11:            NO        0              0.            0.
     Signal 12:            NO        0              0.            0.

     Signal 13:            NO        0              0.            0.
     Signal 14:            NO        0              0.            0.
```

See Notes for TCCA — screen 4/7

The later version of the software contains a screen like the sample shown below:

```
                TCCA Card Definition - Socket 1 - Screen 5/7

                        Milliamp Input Definitions   9 - 14

            Signal    CDB value    CDB value  CDB value  Enable   Enable
            in use    Full Scale    at 4 mA   at 20 mA   Low diag High diag
            -------   ----------   ---------  ---------  -------- ---------
Signal  9:    YES        128         0.0        100.0       NO       NO
Signal 10:    YES        128         0.0        100.0       NO       NO

Signal 11:    YES        128         0.0        100.0       NO       NO
Signal 12:    YES        128         0.0        100.0       NO       NO

Signal 13:    YES        128         0.0        100.0       NO       NO
Signal 14:    YES        128         0.0        100.0       NO       NO
```

See Notes for TCCA — screen 4/7

```
            TCCA Card Definition - Socket 1 - Screen 6/7

                    Miscellaneous Values

       Cold Junction Filtering  (power of 2):    16
```

Cold junction filtering determines the number of readings used in a floating average noise filter. Cold junction filtering is defined by the Mark V system designers and is changed based on experience. It must be set in a power of 2.

```
              TCCA Card Definition - Socket 1 - Screen 7/7

             Shaft Voltage and Current Monitor Definitions

                       Full Scale    Min (0v)      Max (16v)
                       CDB value     CDB value     CDB value
                       ----------    ---------     ---------
  AC Current:              32           0.0           16.0

  DC Voltage:              32           0.0           16.0

  DC Current:              32           0.0           16.0
```

This application monitors AC current and AC and DC voltages that develop on the shaft. These develop due to static electricity and to eddy currents that originate at the generator and travel through the shaft. These are monitored to avoid pitting on the bearings caused by continuous arcing. These functions are typically used for Steam applications.

Full Scale Control Database Values are set to the maximum value associated with the input, usually 32 volts or amps.

Minimum and Maximum Control Database Values determine the input to the card, usually 0 to 16 volts or amps, and can have a maximum value of 32768.

The later version of the software contains a screen like the sample below:

```
                        TCCA Card Definition - Socket 1 - Screen 7/7

                  Shaft Voltage Monitor Definitions

     Current Shunt Resistance (0.0005 to 0.125 ohms):    0.005

Shaft current monitor reports the AC rms current through the shunt.

The shaft voltage monitor reports the frequency at which the shaft voltage
exceeds fixed hardware limits.

An AC test will inject signals to the TCCA card circuits and produce 1000Hz
for the voltage monitor frequency and 12 to 20A for the current monitor.

A DC test will inject a DC current into the grounding brushes and the shunt.
The resultant voltages are used to calculate the resistance to ground for each.
```

B-9. TCCB, <C> CORE

The earlier version of the software is contained in section B-9.1 and the later version of the software is contained in section B-9.2.

B-9.1 TCCB, <C> Core - Earlier version

```
        TCCB Card Definition - Socket 2 - Screen 1/6

                     ePROM Revision Information:

                          Major Rev:   0

                          Minor Rev:   0
```

The TCCB card is located in the TC2000 <C> core location 3. It is "Turbine Control <C> Micro Application Card B". The TCCB configuration defines the following functions:

- <C> RTDs
- <C> milliamp inputs
- <C> RTD and milliamp filtering

It is intended for indication, communication, and alarming only. Control use of input data is advised only in applications where redundancy is not required or not practical.

```
                 TCCB Card Definition - Socket 2 - Screen 2/6

              RTD Type Selection
                                        Type Description: (at 0°C)
                                        ------------------------------------
     RTD 1:  DIN   RTD  8:  DIN     CU10  - 9.035 Ω Copper (10 Ω at 25° C)

     RTD 2:  DIN   RTD  9:  DIN     DIN   - 100 Ω Platinum, α = 0.00385

     RTD 3:  DIN   RTD 10:  DIN     PURE  - 100 Ω Platinum, α = 0.003926

     RTD 4:  DIN   RTD 11:  DIN     SAMA  - 98.129 Ω Platinum, α = 0.00385

     RTD 5:  DIN   RTD 12:  DIN     USIND - 100 Ω Platinum, α = 0.00391

     RTD 6:  DIN   RTD 13:  DIN     N120  - 120 Ω Nickel

     RTD 7:  DIN   RTD 14:  DIN      ----  - not used
```

RTD_types are defaulted to the standard DIN. The actual type required depends on the requisition.

```
                  TCCB Card Definition - Socket 2 - Screen 3/6

               Milliamp Input Definitions   1 thru 8  (4 to 20 mA)

                        Signal    CDB maximum    CDB value    CDB value
                        in use    Scale value     at 4 mA     at 20 mA
                        -------   -----------    ---------    ----------
     Signal  1:           YES         128           0.0          40.96
     Signal  2:           YES         128           0.0          40.96

     Signal  3:           YES         128           0.0          40.96
     Signal  4:           YES         128           0.0          40.96

     Signal  5:           YES         128           0.0          40.96
     Signal  6:           YES         128           0.0          40.96

     Signal  7:           YES         128           0.0          40.96
     Signal  8:           YES         128           0.0          40.96
```

Milliamp Input Screens

Signal in Use is set to either Yes or No.

CDB Maximum Scale Value is set to the maximum value of the scale type of the point assigned with it.

CDB Value at 4mA and 20mA are used to calibrate the milliamp input to the desired range. For example, set the minimum at 90% stroke for an input of 4 mA, and set the maximum to 102% stroke for an input of 20 mA. Inputs 15 through 22 can be set for 0 to 1 mA.

TCCB Card Definition - Socket 2 - Screen 4/6

Milliamp Input Definitions 9 thru 14 (4 to 20 mA)

	Signal in use	CDB maximum Scale value	CDB value at 4 mA	CDB value at 20 mA
Signal 9:	YES	128	0.0	40.96
Signal 10:	YES	128	0.0	40.96
Signal 11:	YES	128	0.0	40.96
Signal 12:	YES	128	0.0	40.96
Signal 13:	YES	128	0.0	40.96
Signal 14:	YES	128	0.0	40.96

See Notes for TCCB screen 3/6

TCCB Card Definition - Socket 2 - Screen 5/6

Milliamp Input Definitions 15 thru 22 (4 to 20 mA) or (0 to 1 mA)

	Signal in use	CDB maximum Scale value	CDB value at 4 mA (0.2 mA)	CDB value at 20 mA (1.0 mA)
Signal 15:	YES	128	0.0	40.96
Signal 16:	YES	128	0.0	40.96
Signal 17:	YES	128	0.0	40.96
Signal 18:	YES	128	0.0	40.96
Signal 19:	YES	128	0.0	40.96
Signal 20:	YES	128	0.0	40.96
Signal 21:	YES	128	0.0	40.96
Signal 22:	YES	128	0.0	40.96

See Notes for TCCB screen 3/6

```
TCCB Card Definition - Socket 2 - Screen 6/6
       Miscellaneous Values

RTD and mA Input Filtering  (power of 2):    16
```

RTD and mA Input filtering determines the number of readings used in a floating average noise filter. Cold junction filtering is defined by the Mark V system designers and are changed based on experience. It must be set in a power of 2.

B-9.2 TCCB, <C> Core - Later Version

```
TCCB Card Definition - Socket 2 - Screen 1/7

      ePROM Revision Information:
           Major Rev:   2
           Minor Rev:   2
```

For notes, see section B-9.1.

```
          TCCB Card Definition - Socket 2 - Screen 2/7
      RTD Type Selection
                              Type Description: (at 0°C)
                              ------------------------------------
RTD 1:-----   RTD  8:-----    CU10  - 9.035 Ω Copper (10 Ω at 25° C)
RTD 2:-----   RTD  9:-----    DIN   - 100 Ω Platinum, α = 0.00385
RTD 3:-----   RTD 10:-----    PURE  - 100 Ω Platinum, α = 0.003926
RTD 4:-----   RTD 11:-----    SAMA  - 98.129 Ω Platinum, α = 0.00385
RTD 5:-----   RTD 12:-----    USIND - 100 Ω Platinum, α = 0.00391
RTD 6:-----   RTD 13:-----    N120  - 120 Ω Nickel
RTD 7:-----   RTD 14:-----     ----  - not used
```

See notes for TCCB screen 2/6 in section B-9.1.

TCCB Card Definition - Socket 2 - Screen 3/7

Milliamp Input Definitions 1 thru 8 (4 to 20 mA)

	Signal in use	CDB value Full Scale	CDB value at 4 mA	CDB value at 20 mA	Enable Low diag	Enable High diag
Signal 1:	NO	128	0.0	40.96	NO	NO
Signal 2:	NO	128	0.0	40.96	NO	NO
Signal 3:	NO	128	0.0	40.96	NO	NO
Signal 4:	NO	128	0.0	40.96	NO	NO
Signal 5:	NO	128	0.0	40.96	NO	NO
Signal 6:	NO	128	0.0	40.96	NO	NO
Signal 7:	NO	128	0.0	40.96	NO	NO
Signal 8:	NO	128	0.0	40.96	NO	NO

For notes, see screen 3/6 of section B-9.1 and screen 4/7 in section B-8.

TCCB Card Definition - Socket 2 - Screen 4/7

Milliamp Input Definitions 9 thru 14 (4 to 20 mA)

	Signal in use	CDB value Full Scale	CDB value at 4 mA	CDB value at 20 mA	Enable Low diag	Enable High diag
Signal 9:	NO	128	0.0	40.96	NO	NO
Signal 10:	NO	128	0.0	40.96	NO	NO
Signal 11:	NO	128	0.0	40.96	NO	NO
Signal 12:	NO	128	0.0	40.96	NO	NO
Signal 13:	NO	128	0.0	40.96	NO	NO
Signal 14:	NO	128	0.0	40.96	NO	NO

For notes, see screen 3/6 of section B-9.1

```
               TCCB Card Definition - Socket 2 - Screen 5/7

     Milliamp Input Definitions  15 thru 22  (4 to 20 mA) or (0 to 1 mA)

               Signal    CDB value    CDB value  CDB value  Enable   Enable
               in use    Full Scale    at 4 mA   at 20 mA   Low diag High diag
                                       (0.2 mA)   (1.0 mA)
               -------   ----------   ---------  ---------  -------- ---------
   Signal 15:     NO        128         0.0        40.96       NO       NO
   Signal 16:     NO        128         0.0        40.96       NO       NO

   Signal 17:     NO        128         0.0        40.96       NO       NO
   Signal 18:     NO        128         0.0        40.96       NO       NO

   Signal 19:     NO        128         0.0        40.96       NO       NO
   Signal 20:     NO        128         0.0        40.96       NO       NO

   Signal 21:     NO        128         0.0        40.96       NO       NO
   Signal 22:     NO        128         0.0        40.96       NO       NO
```

For notes, see screen 3/6 of section B-9.1

```
               TCCB Card Definition - Socket 2 - Screen 6/7

                Generator Configuration Information


     Nominal Bus Frequency               (0 to 64 hz):    60.0

     Bus Voltage (at 115 VRMS PT out   :  0 to 256KV):    88.0

     Gen Voltage (at 115 VRMS PT out   :  0 to 256KV):    88.0

     Gen Current (at 5 AMPS CT out : 0 to 32768 amps):    32768

     Power                            ( 0 to 2048 MW):    2000.0
            Power = Gen Voltage * Gen Current * 1.732


               TCCB Card Definition - Socket 2 - Screen 7/7

                     Miscellaneous Values


          RTD and mA Input Filtering  (power of 2):    16
```

For notes, see screen 6/6 of section B-9.1.

B-10. TCDA, <C> CORE

TCDA_3 Card Definition - Socket 4 - Screen 1/3

ePROM Revision Information:

Major Rev: 1

Minor Rev: 1

See Notes for TCDA_1 in Section B-3.

TCDA_3 Card Definition - Socket 4 - Screen 2/3

Contact Input Assignments - <CD> DTBA points 1 thru 92
& <CD> DTBB points 1 thru 4

Contact Inputs:	1-10			11-20			21-30			31-40			41-48		
	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det	DTBA Pnt	Inv Msk	Chg Det
CI x1:	1	1	1	21	1	1	41	1	1	61	0	1	81	1	1
CI x2:	3	0	1	23	1	1	43	1	1	63	1	1	83	1	1
CI x3:	5	1	1	25	1	1	45	1	1	65	1	1	85	0	1
CI x4:	7	1	1	27	0	1	47	1	1	67	1	1	87	0	1
CI x5:	9	1	1	29	1	1	49	0	1	69	1	1	89	1	1
CI x6:	11	1	1	31	1	1	51	0	1	71	1	1	91	1	1
CI x7:	13	1	1	33	0	1	53	1	1	73	1	1	DTBB1	0	1
CI x8:	15	0	1	35	1	1	55	1	1	75	1	1	DTBB3	1	1
CI x9:	17	0	1	37	1	1	57	1	1	77	1	1			
CI x0:	19	0	1	39	1	1	59	0	1	79	0	1			

See Notes for TCDA_1 <Q> Core in section B-3.

TCDA_3 Card Definition - Socket 4 - Screen 3/3

Contact Input Assignments - <CD> DTBB points 5 thru 100

Contact Inputs:	49-60			61-70			71-80			81-90			91-96		
	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det	DTBB Pnt	Inv Msk	Chg Det
CI49:	5	1	1												
CI50:	7	0	1												
CIx1:	9	0	1	29	0	1	49	1	1	69	1	1	89	1	1
CIx2:	11	0	1	31	0	1	51	1	1	71	1	1	91	1	1
CIx3:	13	0	1	33	1	1	53	1	1	73	1	1	93	0	1
CIx4:	15	0	1	35	0	1	55	1	1	75	1	1	95	1	1
CIx5:	17	1	1	37	1	1	57	0	1	77	1	1	97	0	1
CIx6:	19	1	1	39	1	1	59	0	1	79	1	1	99	0	1
CIx7:	21	0	1	41	1	1	61	1	1	81	1	1			
CIx8:	23	1	1	43	0	1	63	1	1	83	1	1			
CIx9:	25	0	1	45	1	1	65	1	1	85	1	1			
CIx0:	27	0	1	47	1	1	67	1	1	87	1	1			

See Notes for TCDA_1 <Q> Core in section B-3.

B-11. LCCC, <C> CORE

LCCC Card Definition - Socket 8 - Screen 1/1

ePROM Revision Information:

Major Rev: 1

Minor Rev: 0

B-12. DCCB, <C> CORE

DCCB Card Definition - Socket 12 - Screen 1/1

ePROM Revision Information:

Major Rev: 1

Minor Rev: 1

Later versions of this software contain the second screen documented below.

```
            DCCb Card Definition - Socket 12 - Screen 2/2

        EX2000 Exciter Event Configuration Data:

          Disable EX2000 Exciter Events:   YES
```

Disable EX2000 Exciter Events should be set to NO if an EX2000 is present on the Stagelink, and YES if an EX2000 is not present.

B-13. IOMA, <C> CORE

```
          IOMA Card Definition - Socket 13 - Screen 1/1

     IOMA/320 ePROM Revision Information:

        IOMA firmware:
             Major Rev:    3
             Minor Rev:    1

         320 firmware:
             Major Rev:    1
             Minor Rev:    1
             Prom Page Select:    0
```

B-14. TCCA <D> CORE

```
           TCCA Card Definition - Socket 1 - Screen 1/1

              ePROM Revision Information:

                   Major Rev:   2

                   Minor Rev:   0
```

B-15. LCCD <D> CORE

```
           LCCD Card Definition - Socket 8 - Screen 1/1

              ePROM Revision Information:

                   Major Rev:   1

                   Minor Rev:   0
```

B-16. DCCB <D> CORE

```
                 DCCB Card Definition - Socket 12 - Screen 1/1

                     ePROM Revision Information:

                          Major Rev:   1

                          Minor Rev:   1
```

Later versions of this software contain the second screen documented below.

```
              DCCb Card Definition - Socket 12 - Screen 2/2


                EX2000 Exciter Event Configuration Data:


                  Disable EX2000 Exciter Events:   YES
```

Disable EX2000 Exciter Events should be set to NO if an EX2000 is present on the Stagelink, and YES if an EX2000 is not present.

B-17. IOMA <D> CORE

```
                 IOMA Card Definition - Socket 13 - Screen 1/1

            IOMA/320 ePROM Revision Information:

               IOMA firmware:
                    Major Rev:    1
                    Minor Rev:    1

                320 firmware:
                    Major Rev:    1
                    Minor Rev:    1
                    Prom Page Select:    0
```

B-18. QTBA TERMINAL BOARD I/O

- Contains six (6) magnetic pulse input circuits used for High Pressure (HP) and Low Pressure (LP) speed and fuel flow divider pickup signals (single or dual shaft turbines).
- Servo Valve outputs
- Water injection flow meter inputs
- Megawatt transducer inputs
- Supplies excitation current for LVDT's
- RS232 port for Terminal Interface Monitor (TIMN)

B-19. TBQA TERMINAL BOARD I/O

- TBQA card is devoted to thermocouple circuits

B-20. TBQB TERMINAL BOARD I/O

- Compressor discharge pressure inputs
- Magnetic pickup circuit
- Undedicated 0-10 V input
- Hardware jumper configurable - (Voltage (+/- 10 V dc) / Current type) inputs.
- Vibration inputs (Seismic)

B-21. TBQC TERMINAL BOARD I/O

- Terminates 4-20ma circuits
  a. 2 wire (external power) b. 3 wire (internal power) c. 4 wire (internal power)
- Receives LVDT input signals
- 4-20ma / 200ma current output (hardware jumper select)

B-22. TBQD TERMINAL BOARD I/O (optional)

- Proximiter vibration input (if used)
- Additional LVDT input capability
- Additional current outputs (4-20 / 200ma Hardware Jumper select)

B-23. TBQE TERMINAL BOARD I/O (optional for LM6000 applications)

- Proximiter vibration input (if used)
- Additional LVDT input capability
- Additional current outputs (4-20 / 200ma Hardware Jumper select)
- Additional Magnetic pickup circuits
- Provides inputs to <Q> processor (RTDs)

B-24. TBQF TERMINAL BOARD I/O (optional for Large/Medium Steam applications)

- Additional 4-20ma inputs
- Additional LVDT input capability
- Additional current outputs (4-20 / 200ma Hardware Jumper select)

B-25. TBQG TERMINAL BOARD I/O (optional for GHD applications)

- Additional 10ma 3-Coil servo outputs
- Additional voltage/current inputs (jumper selectable)
- Additional 4-20ma current outputs
- Additional pulse rate inputs

B-26. CTBA TERMINAL BOARD I/O

- Interfaces I/O signals to <C> processor
- 4-20ma input (can be configured for internal or external power)
- Current outputs driven from <C> processor (4-20ma)
- Voltage monitor input circuitry
- RS232 port for TIMN monitor
- Input for time tic (master clock)
- JX plug for I/O net termination
- Contains two ARCNET connectors for STAGE LINK

B-27. TBCA TERMINAL BOARD I/O

- Provides inputs to <C> processor (RTDs)

B-28. TBCB TERMINAL BOARD I/O

- Optional board, provides 4-20ma inputs to <C> (Hardware Jumper select for intern. or extern. power)
- Additional RTD inputs

B-29. PTBA TERMINAL BOARD I/O

- External interface to <P> core
- Contains six magnetic pickup circuits for emergency overspeed inputs (single & dual shaft machines)
- Flame detection inputs
- Bus and generator potential transformer inputs (allows reading of voltage on generator side of breaker and bus)
- Three current transformer inputs to detect current from generator
- Provides connectors to alarm horn inside <P> core
- Contains generator breaker interface circuitry
- Termination points for emergency trip pushbutton hookup mounted on Mark V door
- Auxiliary manual trip pushbuttons (remove jumpers to enable)
- Points for external trip (process related)
- Solenoid drivers to drive emergency trip relay solenoid

B-30. DTBA TERMINAL BOARD I/O

- Provides interface capability with either <CD> or <QD> cores
  a. Contact input circuits
  b. Plug connection to <PD> core to carry 125 V dc power

B-31. DTBB TERMINAL BOARD I/O

- Interface with either <CD> or <QD>
- Contact input circuitry
- Plug in connector to <PD> core for 125vdc (daisy-chained from DTBA)

B-32. DTBC TERMINAL BOARD I/O

- Contacts for digital output (<QD>, <CD>) in both solenoid and "C" form contact (hardware jumper select for some outputs)
- Power hookup to <PD> core to supply power to solenoid and contact outputs
- Circuits 1-15 jumper configurable (solenoid driver or dry contact)
- Circuits 16, 17, 18 special circuits - 16 = resistor equipped, 17 + 18 supplied with fuses (in QD1)
- Dry form "C" contacts outputs

B-33. DTBD TERMINAL BOARD I/O

- Interface with either <CD> or <QD>
- Contact input circuitry
- Plug in connector to <PD> core for 125 V dc (daisy-chained from DTBA)
- Interface circuitry for ignition transformers

B-34. SIGNAL CONDITIONING BOARDS

Signal conditioning boards accept inputs from one or more terminal boards, conditions the signals for the smart-I/O and microprocessor cards, and routes those signals to the appropriate processor.

B-35. MICRO PROCESSOR CARDS

The DCC and LCC daughter board microprocessor cards provide multi-role proccessing support for a number of GEDS product lines. For the Mark V panel, these devices execute the control sequence program, provide data processing for I/O and communication signals, and perform a synchronization check function via respective Intel 80186 and 80196, and Texas Instruments 32015 microprocessors.

B-36. TCPS CARD

- Supplies power to cores within Mark V panel
- +/- 24 V (5 A fused)
- +/- 15 V
- +/- 5 V
- +/- 15 V (A) ungrounded
- +/- 15 V (B) ungrounded

B-37. TCPD CARD

- Receives one dc and up to two ac inputs, converts to one dc source for core power supply

The following I/O card configuration screens are application specific and may not be present for a specific site.

B-38. LMQA / TCQA CARD

```
                 LMQA Card Definition - Socket 1 - Screen 1/14

                     ePROM Revision Information:

                          Major Rev:   1

                          Minor Rev:   9


                 LMQA Card Definition - Socket 1 - Screen 2/14
                          VIGV Regulator Definition


Suicide enable :-   Current Fault:   YES      LVDT/R fault:  YES

Current Bias: (0 to 100% rated [ 120ma])          16.67
Current Gain: (0 to 200% rated_cur / %pos)        6.37

Zero Stroke (+/-1.0  (V1-V2)/(V1+V2) ratio)  LVDT 1:-0.4812  LVDT 2:-0.4802  100%
Stroke (+/-1.0  (V1-V2)/(V1+V2) ratio)  LVDT 1:0.4698   LVDT 2:0.4767


                 LMQA Card Definition - Socket 1 - Screen 3/14

                         VBV  Regulator Definition


Suicide enable :-   Current Fault:   YES      LVDT/R fault:  YES
Current Bias: (0 to 100% rated [ 120ma])          -16.67
Current Gain: (0 to 200% rated_cur / %pos)        5.68

Zero Stroke (+/-1.0  (V1-V2)/(V1+V2) ratio)  LVDT 1:-0.4751  LVDT 2:-0.4748
100% Stroke (+/-1.0  (V1-V2)/(V1+V2) ratio)  LVDT 1:0.2268   LVDT 2:0.2277


                 LMQA Card Definition - Socket 1 - Screen 4/14
                           VSV Regulator Definition


Suicide enable :-   Current Fault:   YES      LVDT/R fault:  YES

Current Bias: (0 to 100% rated [ 120ma])          16.67
Current Gain: (0 to 200% rated_cur / %pos)        9.88

Zero Stroke (+/-1.0  (V1-V2)/(V1+V2) ratio)  LVDT 1:-0.4826  LVDT 2:-0.4768
100% Stroke (+/-1.0  (V1-V2)/(V1+V2) ratio)  LVDT 1:0.4807   LVDT 2:0.4793
```

The regulators above are hardcoded for variable geometry control servos.

The **SUICIDE ENABLE** fields, **CURRENT FAULT** and **LVDT/R FAULT** are always enabled for LM6000.

CURRENT BIAS and **CURRENT GAIN** are constants calculated from the LM6000 control specification and should not be changed.

The **LVDT** stroke values are set during calibration in accordance with the GE LM6000-MKV Site Setting Instructions.

```
                 LMQA Card Definition - Socket 1 - Screen 5/14
                          TTB  Regulator Definition

   Suicide enable :-   Current Fault:    NO      LVDT/R fault:   NO

   Current Bias: (0 to 100% rated [120 mA])          3.0
   Current Gain: (0 to 200% rated_cur./%pos.)        7.72


   Pos limits          (-128% to 128%) :- Low:  -100.0   High:   100.0
   Integrator convergence gain  (0 to  1 %/%):  0.3
   Position reference Gain      (0 to 32 %/%):  0.309
   Position ref time constant   (0 to 8 Sec) :  1.0

   Note: Enter PTB Pressure data on Milliamp Input screen.
```

The **TTB Regulator** is assigned to control engine thrust balance air and is not currently used on the LM6000 Base Engine control. Regulator settings are similar to the TCQA Type 40 regulator.

```
                 LMQA Card Definition - Socket 1 - Screen 6/14
                     Position Milli-Amp Output Configuration
Milli-Amp Output 1:

   Output used (enables output and basic diagnostics):        NO
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA/ 40 mA) CDB value:    11.11
   Maximum (20 mA/200 mA) CDB value:    100.0

Milli-Amp Output 2:

   Output used (enables output and basic diagnnostics):       NO
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA/ 40 mA) CDB value:    12.0
   Maximum (20 mA/200 mA) CDB value:    133.13
```

The **MINIMUM** and **MAXIMUM CDB** values are set during calibration in accordance with the GE LM6000-MKV Site Setting Instructions.

```
              LMQA Card Definition - Socket 1 - Screen 7/14
                              Voltage Input

                      Signal      Full Scale    Min (0v)     Max (10v)
                      in use      CDB value     CDB value    CDB value
                      -------     ----------    ---------    ---------
   Signal  1:           NO         2048          0.0          500.0
   Signal  2:           NO         128           0.0          100.0

   Signal  3:           NO         128           0.0          100.0
   Signal  4:           NO         128           0.0          100.0

   Scale of PS3 in CDB:  2048
```

Signals 1 through 4 are not used for the LM6000 Base Engine. The scale for PS3 is the MAX CDB for the LM6000 database PRESS scale code.

```
              LMQA Card Definition - Socket 1 - Screen 8/14

                      Trace Buffer Definition

  Regulator Mode (tune, trace):               Tune

  First  Regulator selection (tune & trace):  1
  Second Regulator selection (trace only):    2

  # of trace samples after trigger:           40
  Trace sample rate in hertz:                 32

  Tune disturbance magnitude (0 to 10%):      3.0
```

This is used as described in the TCQA Card Definition.

```
              LMQA Card Definition - Socket 1 - Screen 9/14
                   Thermocouple Type Selection

   TC  1:  K        TC  5:  K         TC  9:   K         TC 13:   -
   TC  2:  K        TC  6:  K         TC 10:   K         TC 14:   -
   TC  3:  K        TC  7:  K         TC 11:   -         TC 15:   -
   TC  4:  K        TC  8:  K         TC 12:   K

          Valid types:    K, J, E, T
            not used :    -
```

Enter the type of connected thermocouple or a dash if input is not used.

```
              LMQA Card Definition - Socket 1 - Screen 10/14
                           Pulse Rate Definition

                         Gain Scaling   Pulse Rate
                          Base (rpm)    at Scaling
                         (power of 2)   Base Value
     XN25               ------------   ----------
     Pulse rate :           16384          12273
```

The **GAIN SCALING BASE** is the MAX CDB of the LM6000 database RPMX scale code. The Pulse Rate at MAX CDB is based on the ratio of .7491 Hz/rpm for HP rotor speed.

```
              LMQA Card Definition - Socket 1 - Screen 11/14
                           Vibration Definitions

                     Transducer used          Vibration Sensitivity
                                        (0.05 to 0.40 V peak / ips peak)

  Transducer  1:                NO                                0.15
  Transducer  2:                NO                                0.15
  Transducer  3:                NO                                0.15
  Transducer  4:                NO                                0.15
  Transducer  5:                NO                                0.15
  Transducer  6:                NO                                0.15
  Transducer  7:                NO                                0.15
  Transducer  8:                NO                                0.15
  Transducer  9:                NO                                0.15
  Transducer 10:                NO                                0.15
  Transducer 11:                NO                                0.15
  Transducer 12:                NO                                0.15
```

This is used as described in the TCQA Card Definition.

LMQA Card Definition - Socket 1 - Screen 12/14
Milliamp Input Definitions 1 - 8

	Signal in use	Low Input Diagn	High Input Diagn	Full Scale CDB value	Minimum (4mA) CDB value	Maximum (20mA) CDB value
Signal 1:	YES	YES	YES	2048	0.0	500.0
Signal 2:	YES	YES	YES	2058	0.0	500.0
Signal 3:	YES	YES	YES	2048	0.0	300.0
Signal 4:	YES	YES	YES	256	0.0	150.0
Signal 5:	NO	YES	YES	128	0.0	100.0
Signal 6:	YES	YES	YES	2048	0.0	1000.0
Signal 7:	YES	YES	YES	32	0.0	20.0
Signal 8:	NO	YES	YES	128	0.0	100.0

LMQA Card Definition - Socket 1 - Screen 13/14
Milliamp Input Definitions 9 - 15

	Signal in use	Low Input Diagn	High Input Diagn	Full Scale CDB value	Minimum (4mA) CDB value	Maximum (20mA) CDB value
Signal 9:	NO	YES	YES	128	0.0	100.0
Signal 10:	YES	YES	YES	512	-40.0	40.0
Signal 11:	YES	YES	YES	32	-10.0	10.0
Signal 12:	NO	YES	YES	128	0.0	100.0
Signal 13:	NO	YES	YES	128	0.0	100.0
Signal 14:	NO	YES	YES	128	0.0	100.0
Signal 15:	NO	YES	YES	128	0.0	100.0

This milliamp inputs are used as described in the TCQA Card Definition.

```
              LMQA Card Definition - Socket 1 - Screen 14/14
                      Miscellaneous Values


   Cold Junction Filtering  (power of 2):         16


   Seismic Vibr. Xdcr. Open  Ckt. (0 to 3VDC):    1.000

       <       V = 30R / (40000+R)    {eg,  1V=1379Ω }     >
       < Xdcr Resistance = 40000*V/(30-V) {eg,  1379Ω=1V } >

   125 VDC Under Voltage Limit (0 to 125 VDC):    105.0

   125 VDC Bus Ground Fault    (0 to  65 VDC):    31.24
       {voltage at which ground fault is indicated}
```

This is used as described in the TCQA Card Definition.

B-39. TCQE CARD

```
                  TCQE Card Definition - Socket 2 - Screen 1/9

                   TCQE 196/320 Card Information:

                      TCQE 196 firmware:
                           Major Rev:        2
                           Minor Rev:        1

                      TCQE 320 firmware:
                           Major Rev:        2
                           Minor Rev:        1
```

```
                  TCQE Card Definition - Socket 2 - Screen 2/9
                   Proximitor shaft location and configuration

         Prox  1: LP       Prox  2: LP       Prox  3: LP       Prox  4: LP
         Prox  5: --       Prox  6: --       Prox  7: --       Prox  8: --


                                    HP rpm      IP rpm      LP rpm
Max CDB scaling value  :            16384       8192        8192


     Notes::    Valid Prox. locations:    IP,  HP,  LP,  -- (not used)
```

Proximitor inputs are measured as a function of shaft speed. The shaft which a proximitor is measuring vibration is entered at the top part of the screen. The **MAX CDB** for the appropriate LM6000 database scale code is then entered for each shaft.

```
                  TCQE Card Definition - Socket 2 - Screen 3/9
            Proximitors 1 thru 8   :-  Peak to Peak Vibration
                       Transducer Sensitivity
                        (0 to 35.747 V/mil)
                        --------------------
      Proximitor 1:          0.2
      Proximitor 2:          0.2
      Proximitor 3:          0.2

      Proximitor 4:          0.2
      Proximitor 5:          0.
      Proximitor 6:          0.

      Proximitor 7:          0.
      Proximitor 8:          0.

      Proximitor 1 thru 8 diag. low  limit: (0 to 18 v)     2.002
      Proximitor 1 thru 8 diag. high limit: (0 to 18 v)     14.002


                  TCQE Card Definition - Socket 2 - Screen 4/9
            Proximitor Position and Keyphasor Configuration

 Position Inputs:         Transducer Sensitivity     Position Offset
                          (0 to 185.856 V/mil)      (0 to 2048 mils)
                          ----------------------     ----------------
         Proximitor   9:          0.                         0.0
         Proximitor  10:          0.                         0.0

 Keyphasor Input:        Transducer Sensitivity      Position Offset
                         (0 to 185.856 V/mil)      (0 to 2048 mils)
                         ----------------------     ----------------
         Proximitor  11:          0.                         0.0

         Proximitor 9 and 10 diag. low  limit: (0 to 18v)     17.319
         Proximitor 9 and 10 diag. high limit: (0 to 18v)     2.473
         Proximitor 11 diagnostic low  limit : (0 to 18v)     17.319
         Proximitor 11 diagnostic high limit : (0 to 18v)     2.473
```

Proximitor, Position, and Keyphasor are configured in accordance with the TCQB Card Definition.

```
                  TCQE Card Definition - Socket 2 - Screen 5/9
                            Lvdt/ Lvdr Inputs

                             Minimum Voltage       Maximum Voltage
                            (0 to 10.00 Vrms)     (0 to 10.00 Vrms)
                           -------------------   -------------------
     Lvdt/Lvdr    1:              2.106                 5.798

     Lvdt/Lvdr    2:              0.758                 0.758

     Lvdt/Lvdr    3:              0.758                 3.558

     Lvdt/Lvdr    4:              0.758                 3.558
```

LVDT/R values are entered in accordance with the GE LM6000-MKV Site Setting Instructions.

```
                  TCQE Card Definition - Socket 2 - Screen 6/9
                Position Milli-Amp Output Configuration

Milli-Amp Output 1:

   Output used (enables output and basic diagnostics):       YES
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA/ 40 mA) CDB value:    11.5
   Maximum (20 mA/200 mA) CDB value:    124.6


Milli-Amp Output 2:

   Output used (enables output and basic diagnnostics):       NO
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA/ 40 mA) CDB value:    13.5
   Maximum (20 mA/200 mA) CDB value:    129.8
```

The **MINIMUM** and **MAXIMUM CDB** values are set during calibration in accordance with the GE LM6000-MKV Site Setting Instructions.

```
                 TCQE Card Definition - Socket 2 - Screen 7/9
                       LM Vibration Inputs
 Accelerometer Inputs:     Transducer      Transducer Sensitivity
                             Used          (0 to 0.5882 V/ips peak)
                           ----------      ------------------------
     Transducer 1  HP:       YES             0.1
                   IP:        NO             0.1
                   LP:       YES             0.1

     Transducer 2  HP:       YES             0.1
                   IP:        NO             0.1
                   LP:       YES             0.1

     Transducer 3  HP:        NO             0.1
                   IP:        NO             0.1
                   LP:        NO             0.1

     Transducer    Diagnostic low  limit : (0 to 12v)     7.001
     Transducer    Diagnostic high limit : (0 to 12v)     12.0
```

Accelerometer inputs are configured in accordance with the TCQB Card Definition.

```
                 TCQE Card Definition - Socket 2 - Screen 8/9
                             Pulse Rate Definitions

                   Gain Scaling   Pulse Rate    Application    Overspeed
                    Base (rpm)    at Scaling       Type        Trip Limit
                   (power of 2)   Base Value    (see notes)      (rpm)
                   ------------   ----------    -----------    ----------
 Pulse rate 1&2:    8192           6554          speed          4050
 Pulse rate 3&4:    8192           6554          speed          4050

          Note: Valid Application Types (speed, fuel, pmg, -----)

      Turbine Speed Differential Diagnostic Limit (rpm):    100
```

The **GAIN SCALING BASE** is the **MAX CDB** of the LM6000 database RPM scale code. The Pulse Rate at **MAX CDB** is based on the ratio of .8 Hz/rpm for LP rotor speed. Overspeed trip limit is set as required by the LM6000 control specification.

```
                 TCQE Card Definition - Socket 2 - Screen 9/9
                     RTD Type Selection
                                                Type Description:(at 0°C)
                                                -------------------------
            RTD  1:LM200                        DIN   - 100 Ω Platinum
                                                              α = 0.00385
            RTD  2:LM200                        PURE  - 100 Ω Platinum
                                                              α = 0.003926
            RTD  3:LM200                        SAMA  - 98.129 Ω Platinum
                                                              α = 0.00385
            RTD  4:LM200                        USIND - 100 Ω Platinum
                                                              α = 0.00391
                                                N120  - 120 Ω Nickel
                                                LM200 - 200 Ω Platinum
                                                              α = 0.00391
                                        ----  - not used
```

RTD's are configured as described in the TCCA Card Definition. The TCQE inputs are reserved for the T2 and T25 RTD's in the LM6000 application since these inputs are the only ones that can be configured as 200 ohm (LM200) RTD's

B-40. TCPA, <PLU> CORE

```
          TCPA Card Definition - Socket 14 - Screen 1/8

            TCPA 196/320 ePROM Revision Information:

               TCPA 196 firmware:
                    Major Rev:    1
                    Minor Rev:    2

               TCPA 320 firmware:
                    Major Rev:    1
                    Minor Rev:    2


                 TCPA Card Definition - Socket 14 - Screen 2/8

                 Protective Load Unbalance Configuration:

     Enable PLU function:                YES
     Enable PLU response delay:          YES
     Enable EVA on Control Valve 3:       NO
     Enable EVA on Control Valve 4:       NO

     PLU current rate threshold (HIgh, MEdium, LOw):     ME
     PLU unbalance threshold      (0 to 2 per unit):    0.4
     PLU current ratio                     (0 to 8):    2.1299
           ratio = 8.57 Amps / (rated CT current)
     PLU rated reheat pressure         (0 to 128 %):    100.0
     PLU delay                       (0 to  64  sec):    0.008
                   TCPA Card Definition - Socket 14 - Screen 3/8

                 Protective Load Unbalance Configuration:

     Enable EVA function:                 NO
     Enable EVA external function:        NO

     EVA megawatt rate threshold (HIgh, MEdium, LOw):    ME
     EVA unbalance threshold       (0 to 2 per unit):   0.7
     EVA drop out delay              (0 to  64  sec):   1.0


                 TCPA Card Definition - Socket 14 - Screen 4/8

                        Milliamp Input Definitions   1 - 4


                   Low     High
          Signal   Input   Input   Full Scale  Minimum (4mA)  Maximum (20mA)
          in use   Diagn   Diagn   CDB value    CDB value      CDB value
          ------   -----   -----   ----------  -------------  --------------
Signal  1:   YES     YES     YES      2048         0.0            500.0
Signal  2:   YES     YES     YES      2048         0.0            500.0
Signal  3:   YES     YES     YES      2048         0.0            300.0
Signal  4:   YES     YES     YES      256          0.0            150.0
```

```
          TCPA Card Definition - Socket 14 - Screen 5/8

        Proximitors  (1 thru 10) :-  Peak to Peak Vibration

                      Input         Transducer Sensitivity
                      Used            (0 to 12.80 V/mil)
                      -----         -----------------------
 Proximitor 1:        YES                   0.2
 Proximitor 2:        YES                   0.2
 Proximitor 3:         NO                   0.2
 Proximitor 4:         NO                   0.2
 Proximitor 5:         NO                   0.2
 Proximitor 6:         NO                   0.2
 Proximitor 7:         NO                   0.2
 Proximitor 8:         NO                   0.2
 Proximitor 9:         NO                   0.2
 Proximitor 10:        NO                   0.2

 Proximitor 1 thru 10 diag. low  limit: (0 to 18v)     1.0
 Proximitor 1 thru 10 diag. high limit: (0 to 18v)     17.0

         TCPA Card Definition - Socket 14 - Screen 6/8

        Proximitors  (11 thru 19)   :-  Position Inputs

                  Input     Transducer Sensitivity  Position Offset
                  Used        (0 to 18.15 V/mil)    (0 to 2048 mils)
                  -----     ----------------------  ---------------
Proximitor  11:    YES             0.2                  45.0
Proximitor  12:    YES             0.2                  45.0
Proximitor  13:     NO             0.2                  45.0
Proximitor  14:     NO             0.2                  45.0
Proximitor  15:     NO             0.2                  45.0
Proximitor  16:     NO             0.2                  45.0
Proximitor  17:     NO             0.2                  45.0
Proximitor  18:     NO             0.2                  45.0
Proximitor  19:     NO             0.2                  45.0

Proximitor 11 thru 19 diag. low  limit: (0 to 18v)     1.0
Proximitor 11 thru 19 diag. high limit: (0 to 18v)     17.0


             TCPA Card Definition - Socket 14 - Screen 7/8

               Proximitor 20 ::   Keyphasor input


                  Input     Transducer Sensitivity  Position Offset
                  Used       (0 to 19.1345 V/mil)   (0 to 2048 mils)
                  -----     ---------------------   ---------------

Proximitor  20:    YES              0.2                  45.0

Proximitor 20 diagnostic low  limit:  (0 to 18)         1.0
Proximitor 20 diagnostic high limit:  (0 to 18)         17.0


             TCPA Card Definition - Socket 14 - Screen 8/8

               Generator Configuration Information

Voltage Gain             (0 to 256KV/pu):    88.0
```

```
Voltage ratio                    (0 to 2):    1.479
  ratio = 162.7 Vrms / (rated PT voltage)

Voltage Phase shift      (-60 to 60 deg):    0.0

Current Gain        (0 to 32768 amps/pu):    16060

Nominal System Frequency       (0 to 64):    60

Power Gain         (0 to 128 percent/pu):    100.0
```

B-41. TCQF, <Q> CORE

```
              TCQF Card Definition - Socket 7 - Screen 1/17

                      ePROM Revision Information:

                           Major Rev:   1

                           Minor Rev:   1
```

The TCQF is a multifunction card located in the <R>, <S>, and <T> cores in location 3 of each. For Simplex applications, only one card is required for the <R> core. It is an optional card used for extended I/O functions within the Gas Heavy Duty (GHD) prom set. The TCQF configuration defines the following functions:

- Servo Regulator Control Loops
- Position Milliamp Outputs
- Regulator Time Constants
- Voltage Inputs
- Trace and Tune Definitions
- Thermocouple Inputs
- Pulse Rate Inputs
- mA Inputs

For regulator application information, see Chapter 7.

```
                TCQF Card Definition - Socket 7 - Screen 2/17

                    Regulator Type Summary (continued on next screen)

Type:        00   Regulator not used in this application

Type:        4x   Position control

Sub_type "x": 0   Feedback: None
              1   Feedback: First assigned LVDT/LVDR
              3   Feedback: Maximum of two assigned LVDT/LVDR's
```

```
                 TCQF Card Definition - Socket 7 - Screen 3/17

                     Regulator Type Summary (continued from previous screen)

Type:        5x   Flow control
Sub_type "x": 1   Feedback: First assigned Flow_input
              2   Feedback: Second assigned Flow_input
              3   Feedback: Maximum of two assigned Flow_inputs

Type:        6x   Flow control with position feedback
Sub_type "x": 4   Feedback: First  Flow_input and max of 2 assigned LVDT/LVDR's
              5   Feedback: Second Flow_input and max of 2 assigned LVDT/LVDR's
              6   Feedback: Both  Flow_inputs and max of 2 assigned LVDT/LVDR's

Type:        7x   Pressure control with position feedback
Sub_type "x": 7   Feedback: Pressure_in (V1) and max of 2 assigned LVDT/LVDR's
              8   Feedback: Pressure_in (V2) and max of 2 assigned LVDT/LVDR's
              A   Feedback: Pressure_in (V1) and first assigned LVDT/LVDR
              B   Feedback: Pressure_in (V2) and first assigned LVDT/LVDR
              C   Feedback: Max Pressure_in (V1, V2) and max of 2 LVDT/LVDR's


                   TCQF Card Definition - Socket 7 - Screen 4/17

            Regulator Definition for Servo Output 1
   Function type & sub-type<00 40 41 43 51 52 53 77 78 7A 7B 7C>=77

   Current Fault Suicide Enable= NO
   Position Limits Suicide Enable= NO  Low=-5.0  %  High=105.0 %

   LVDT Fault Detect Enable= NO   High=100.0 %   Delay=0     Tics
   LVDT Fault Auto Reset Enable= NO   Delay=0     Tics(sec/128)

   Current Bias  (0 to 100% rated [10]) =3.0
   Current Gain  (0 to 200% rated_cur./%pos.) =7.72
   Zero Stroke   (0 to 6.667 Vrms)  LVDT 1=0.6999  LVDT 2=0.6999
   100% Stroke   (0 to 6.667 Vrms)  LVDT 1=3.478   LVDT 2=3.478
   <7> Pos limits   (-128% to 128%) Low=-100.0    High=100.0
   <7> Integrator convergence gain (0 to 16 psi/%)=0.3
   <7> Position reference Gain     (0 to  2 %/psi)=0.1
        Note: for type 5, enter fuel flow data on Pulse Rate screen.
              for type 7, enter time constant data on screen #8.
```

```
            TCQF Card Definition - Socket 7 - Screen 5/17

         Regulator Definition for Servo Output 2
Function type & sub-type[00 40 41 43 51 52 53 64 65 66]:43

Current Fault Suicide Enable: NO
Position Limits Suicide Enable: NO  Low:-5.0  %  High:105.0 %

LVDT Fault Detect Enable: NO   High:100.0 %   Delay:0.0
LVDT Fault Auto Reset Enable: NO   Delay:0.0

Current Bias: (0 to 100% rated [10])        3.0
Current Gain: (0 to 200% rated_cur./%pos.)        7.72
Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478
<6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
<6>   Integrator convergence gain  (0 to  1 %/%): 0.3
<6>   Position reference Gain      (0 to 32 %/%): 0.309
<6>   Position ref time constant   (0 to 8 Sec) : 1.0
      Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
                TCQF Card Definition - Socket 7 - Screen 6/17

         Regulator Definition for Servo Output 3
Function type & sub-type[00 40 41 43 51 52 53 64 65 66]:51

Current Fault Suicide Enable: NO
Position Limits Suicide Enable: NO  Low:-5.0  %  High:105.0 %

LVDT Fault Detect Enable: NO   High:100.0 %   Delay:0.0
LVDT Fault Auto Reset Enable: NO   Delay:0.0

Current Bias: (0 to 100% rated [10])        3.0
Current Gain: (0 to 200% rated_cur./%pos.)        7.72
Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478
<6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
<6>   Integrator convergence gain  (0 to  1 %/%): 0.3
<6>   Position reference Gain      (0 to 32 %/%): 0.309
<6>   Position ref time constant   (0 to 8 Sec) : 1.0
      Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
                TCQF Card Definition - Socket 7 - Screen 7/17

           Regulator Definition for Servo Output 4
   Function type & sub-type[00 40 41 43 51 52 53 64 65 66]:43

   Current Fault Suicide Enable: NO
   Position Limits Suicide Enable: NO  Low:-5.0  %  High:105.0 %

   LVDT Fault Detect Enable: NO   High:100.0 %   Delay:0.0
   LVDT Fault Auto Reset Enable: NO   Delay:0.0

   Current Bias: (0 to 100% rated [10])        3.0
   Current Gain: (0 to 200% rated_cur./%pos.)        7.72
   Zero Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 0.6999   LVDT 2: 0.6999
   100% Stroke      (0 to 6.667 Vrms) :-     LVDT 1: 3.478    LVDT 2: 3.478
   <6>   Pos limits          (-128% to 128%) :- Low: -100.0   High:   100.0
   <6>   Integrator convergence gain  (0 to  1 %/%): 0.3
   <6>   Position reference Gain      (0 to 32 %/%): 0.309
   <6>   Position ref time constant   (0 to 8 Sec) : 1.0
         Note: for type 5 & 6, enter fuel flow data on Pulse Rate screen.
```

```
TCQF Card Definition - Socket 7 - Screen 8/17

                   Time Constant Data for Type 7 Regulators


    Type 77, 7A or 7C:: SRV Warmup time constant (0 to 8 sec):      0.2119
    Type 77, 7A or 7C:: SRV Warmup time constant gain (0 to .625)   0.0
    Type 77, 7A or 7C:: System Warmup  FSR (0 to 128%):             0.0

       Note: Calibrate pressure feedback by adjusting Voltage Input no. 1.


    Type 78, 7B or 7C:: SRV Warmup time constant (0 to 8 sec):      0.2119
    Type 78, 7B or 7C:: SRV Warmup time constant gain (0 to .625)   0.0
    Type 78, 7B or 7C:: System Warmup  FSR (0 to 128%):             0.0

       Note: Calibrate pressure feedback by adjusting Voltage Input no. 2.
```

For definitions of the variables listed above, see Notes for TCQA.

```
                  TCQF Card Definition - Socket 7 - Screen 9/17

                      Position Milli-Amp Output Configuration

Milli-Amp Output 1:

   Output used (enables output and basic diagn.):              NO
   Current suicide enable (YES for TMR, NO for Simplex):       NO
   CDB variable full scale value:       128
   Minimum ( 4 mA)        CDB value:    0.0
   Maximum (20 mA)        CDB value:    100.0


Milli-Amp Output 2:

   Output used (enables output and basic diagn.):              NO
   Current suicide enable (YES for TMR, NO for Simplex):       NO
   CDB variable full scale value:       128
   Minimum ( 4 mA)        CDB value:    0.0
   Maximum (20 mA)        CDB value:    100.0

                    TCQF Card Definition - Socket 7 - Screen 10/17

                      Position Milli-Amp Output Configuration

Milli-Amp Output 3:

   Output used (enables output and basic diagn.):              NO
   Current suicide enable (YES for TMR, NO for Simplex):       NO
   CDB variable full scale value:       128
   Minimum ( 4 mA)        CDB value:    0.0
   Maximum (20 mA)        CDB value:    100.0


Milli-Amp Output 4:

   Output used (enables output and basic diagn.):              NO
   Current suicide enable (YES for TMR, NO for Simplex):       NO
   CDB variable full scale value:       128
   Minimum ( 4 mA)        CDB value:    0.0
   Maximum (20 mA)        CDB value:    100.0
```

```
                 TCQF Card Definition - Socket 7 - Screen 11/17

                     Position Milli-Amp Output Configuration

Milli-Amp Output 5:

   Output used (enables output and basic diagn.):             NO
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA)        CDB value:    0.0
   Maximum (20 mA)        CDB value:    100.0


Milli-Amp Output 6:

   Output used (enables output and basic diagn.):             NO
   Current suicide enable (YES for TMR, NO for Simplex):      NO
   CDB variable full scale value:       128
   Minimum ( 4 mA)        CDB value:    0.0
   Maximum (20 mA)        CDB value:    100.0
```

For definitions of the variables listed above, see Notes for TCQA.

```
                   TCQF Card Definition - Socket 7 - Screen 12/17

                                 Voltage Input

                        Signal      Full Scale    Min (0v)     Max (10v)
                        in use      CDB value     CDB value    CDB value
                        -------     ----------    ---------    ---------
     Signal  1:           YES         2048           0.0          500.0
     Signal  2:            NO         128            0.0          100.0

     Signal  3:            NO         128            0.0          100.0
     Signal  4:            NO         128            0.0          100.0


   Notes: Signal 1 associated with regulator type 77, 7A, or 7C, when used.
          Signal 2 associated with regulator type 78, 7B, or 7C, when used.
```

For definitions of the variables listed above, see Notes for TCQA.

```
        TCQF Card Definition - Socket 7 - Screen 13/17

                   Regulator Tune/Trace Function

    Mode (tune, trace): Trace

    1st regulator number (tune or trace): 1
    2nd regulator number (trace only)   : 2

    Number/2 of recordings to be made after Trigger
     (trace only) 0=0 recordings...255=510 recordings : 40 (0 to 255)

    Recording rate (trace only): 128Hertz(1 to 128)

    Disturbance magnitude (tune only): 3.906%(-10 to +10)

       Note:: For type 7 regulators, 1% equals 16 psi.
```

For definitions of the variables listed above, see Notes for TCQA.

```
                TCQF Card Definition - Socket 7 - Screen 14/17

                   Thermocouple Type Selection

     TC  1:  K         TC  5:  K         TC  9:   K         TC 13:   K
     TC  2:  K         TC  6:  K         TC 10:   K         TC 14:   K
     TC  3:  K         TC  7:  K         TC 11:   K         TC 15:   K
     TC  4:  K         TC  8:  K         TC 12:   K

            Valid types:    K, J, E, T (v=factory test only)
              not used :    -
```

For definitions of the variables listed above, see Notes for TCQA.

TCQF Card Definition - Socket 7 - Screen 15/17

Pulse Rate Definition

	Gain Scaling Base (power of 2)	Max Pulse Rate (100% rating)	Application Type (see notes)
Pulse rate 1:	8192	5104	speed
Pulse rate 2:	8192	5104	speed
Pulse rate 3:	8192	3980	fuel
Pulse rate 4:	8192	3980	fuel

Notes:
Valid Application Types (speed, fuel, pmg, -----)
Pulse rate input 3 is used with regulators types 51, 53, 64, & 66.
Pulse rate input 4 is used with regulators types 52, 53, 65, & 66.

For definitions of the variables listed above, see Notes for TCQA.

TCQF Card Definition - Socket 7 - Screen 16/17

Milliamp Input Definitions 1 - 8

	Signal in use	Low Input Diagn	High Input Diagn	Full Scale CDB value	Minimum (4mA) CDB value	Maximum (20mA) CDB value
Signal 1:	NO	NO	NO	128	0.0	100.0
Signal 2:	NO	NO	NO	128	0.0	100.0
Signal 3:	NO	NO	NO	128	0.0	100.0
Signal 4:	NO	NO	NO	128	0.0	100.0
Signal 5:	NO	NO	NO	128	0.0	100.0
Signal 6:	NO	NO	NO	128	0.0	100.0
Signal 7:	NO	NO	NO	128	0.0	100.0
Signal 8:	NO	NO	NO	128	0.0	100.0

```
             TCQF Card Definition - Socket 7 - Screen 17/17

                        Milliamp Input Definitions   9 - 15

                          Low    High
             Signal  Input  Input    Full Scale  Minimum (4mA)  Maximum (20mA)
             in use  Diagn  Diagn    CDB value    CDB value      CDB value
             ------  -----  -----    ----------  -------------  --------------
Signal  9:     NO      NO     NO       128           0.0            100.0
Signal 10:     NO      NO     NO       128           0.0            100.0

Signal 11:     NO      NO     NO       128           0.0            100.0
Signal 12:     NO      NO     NO       128           0.0            100.0

Signal 13:     NO      NO     NO       128           0.0            100.0
Signal 14:     NO      NO     NO       128           0.0            100.0

Signal 15:     NO      NO     NO       128           0.0            100.0
```

For definitions of the variables listed above, see Notes for TCQA.

Notes:

APPENDIX C

BIG BLOCK REFERENCE

TABLE OF CONTENTS

Notes:

C-1. INTRODUCTION

The Mark V control system's software is configured by using a "block" system. Blocks are simply standardized control functions which are divided into the following categories:

- Primitives
- Generic Big Blocks
- Application Specific Big Blocks

Primitives and generic big blocks are common to all applications and may be utilized for minor control modifications. Application specific big blocks are specific to individual turbine product types, performing dedicated control and protection functions.

This Appendix covers only primitives and generic big blocks, with a separate section for each category. The primitive blocks represent relatively uncomplicated functions such as Add, Subtract, Compare A≥B, Shift, etc.; they are implemented as part of a sequencing rung involving additional control operations. These blocks (once enabled) will energize a coil output when a condition has been satisfied. These operations have universal applicability and may be used in any application. Generic Big Blocks may be used in a similar fashion, but do not have the ability to accept sequencing on its (enable) input or to set a coil output.

In using this Appendix, the words "tile" and "picture" are used in reference to the block illustrations. These terms refer to the Control Sequence Editor displays and Control Sequence Program Printout respectively. (See Chapter five of this manual for more information on the CSP Editor.)

To implement a block correctly, the programmer must be aware of both the functionality of the block and the type of information that must be passed to it. Data received or sent by a block is called a " parameter. " Parameters fall into two categories; those that can be assigned and those that are imbedded within the block and cannot be changed. Parameters that can be assigned are referred to as "passed parameters" and are depicted in a block's Tile. Parameters that may not be changed are referred as "automatics" and both are shown in the control sequence program printout. Blocks read from and write to the control signal database according software signal names assigned to the block. In the case of passed parameters, the Control Sequence Editor (CSE) can be used to make the assignments.

CAUTION

Software signal names are assigned data types. When implementing functions, an understanding of data types and their compatibility is critical. Failure to apply data types correctly may cause incorrect values to be passed to the control signal database. This may in turn, adversely affect turbine operation. (See Summary of data types below.)

C-2. BIG BLOCK ELEMENTS

A BBL is a Big Block, or Big Block Language. This is an algorithm coded by GE Drive Systems and is on the PROMs located on the Mark V controller and communicator processors. The Control Sequence Program, or CSP, for the control system may be written to "call" any available BBL. The BBL is then executed making calculations for the control and protection of the turbine unit. The <I> processor interface to the Mark V has .PIC files which detail the BBLs functions using ASCII text, where the BBL is drawn as a double-lined box. A block is a drawing of a calculation performed by the CSP. It can refer to the entire BBL or to any of the simpler blocks used within the BBL, some of which are called primitives. These .PIC files are used in the CSP printout as well as in the <I>s Rung Display. This document lists the elements used in the .PIC files in order to aid in the reading and understanding of the calculations the BBLs perform.

Each of the Mark V processors has Control Signal Databases, or CDBs. These CDBs contain all the control signals used by the CSP, which are also viewable from the <I>. Each signal has a value used in the control, protection, and monitoring of the turbine unit. A signal has a name, with functional meaning, as well as a scale type that is used to display the signal's value in engineering units.

The signals are either logics or variables. The logics have values of zero or one (false or true, respectively). Variables are typically 16-bit numbers that are scaled by the <I> to represent them in engineering units, such as °F. Signals typically enter the BBL from the left, indicating the BBL reads the signal's value. A signal that exits out the right indicates the signal's value is written by the BBL. The path the value of the signal takes is indicated by the "wire" drawn under the signal name. Sometimes descriptive text within the block replaces the drawing of a wire. This convention holds true most of the time, but be aware that some BBLs are drawn where a signal value is read from the left, but is also over-written subject to internal processing of the inputs.

C-2.1. Signal Names

An upper case signal name indicates that a signal is a control signal, and can be used elsewhere in the CSP as well as viewed from the <I>. If the control signal is inside the block, it is an automatic signal and the block has been coded to always use that specific control signal's value. If it is outside the block, it is a parameter signal, passed to the block. A lower case signal name within the block indicates that a signal is an internal signal, where it's name refers to the signal's function. An internal signal cannot be viewed from the <I>, nor used elsewhere in the CSP. It's value is either calculated as an intermediate signal within the block, or it is passed a value from a parameter signal.

C-2.1.1. AUTOMATIC SIGNAL NAME. When a block is coded to always read and/or write a specific control signal, that signal is an automatic signal and is shown using upper case letters in the signal's name. Automatics are shown inside the BBL, where the symbols ">──╫──" and "──╫──<" show that the internal signal enters or exits the block from or to the Control Signal Database, and is thus viewable by the <I>. The left example shows a signal entering the BBL and the right example shows a signal exiting the BBL.

```
    ║TNH                      CSRIH║
>───╫─────────           ──────────╫───<
```

C-2.1.2. PARAMETER SIGNAL. When a block is passed a value from a control signal name, the parameter signal is passed to an internal signal. The upper case name refers to the control signal and is shown outside the block. The lower case name inside the block is the internal signal and again refers to the function of the signal, possibly even the expected control signal's name. The BBLs have numbers next to input and output parameter signals, indicating the order in which the sequencing editor assigns the parameters. The first example (below, left) shows a control signal "`LK3VIB1`" entering the BBL from the left, where it is referred to internally as "`alm_lim`". The second example (below, right) shows the internal signal "`vib_alm`" exiting out the right side of the BBL as control signal "`L30VIB_ALM1`". Parameter signals enter and exit the BBL from outside the BBL. This indicates this control signal may be changed in order to read or write a different control signal.

```
LK3VIB1          0║alm_lim       vib_alm║6   L30VIB_ALM1
─────────────────╫─────────      ───────╫───────────────────
```

C-2.1.3. INTERMEDIATE SIGNAL. If a wire cannot be drawn to show where an intermediate signal goes within the block (due to lack of space), the signal is given a lower case name. The "——>" symbol is used to indicate "it goes somewhere within this BBL", and the ">——" symbol is used to indicate "it comes from somewhere within this BBL". The first `ctifxbx` (below, far left) "goes somewhere", the second "comes from somewhere". Sometimes the left and right arrows (> and <) are omitted. Sometimes the name is in uppercase, but this is typically done when the signal is not a control signal and it never exits the block. In place of a name, a value can be displayed, such as `2`, `100%`, `zero`, or `max`.

```
ctifxbx   ctifxbx    mode_ss   MRATE    2     100%      zero    max
───────>  >────────  ────────  ───────  ────  ────────  ──────  ─────
```

C-2.1.4. SYSTEM SIGNALS. BBLs use internal signals not accessible with the <I>. The "current control sequencing segment time" is `cur_seg_time`. This signal tells the BBL how often it is executed. It then can adjust its use of timer signals accordingly.

```
cur_seg_time
────────────────
```

C-2.1.5. PREVIOUS SIGNAL VALUE. There are instances where it may be desirable to utilize the previous value of a signal to calculate a new output value. This is represented by the symbol below, where the input is typically from the right side and the previous value exiting from the left.

```
 ┌───┐
 │ -1│
─┤Z  ├─
 └───┘
```

C-2.2. Signal Wires, Contacts, and Coils

The conventions used in the BBLs to signify various connections (and crossovers) are demonstrated and explained in the following sections..

C-2.2.1. WIRE JUNCTIONS. The first example (below, left) shows a horizontal wire and vertical wire not intersecting, using a "`(`" to show that a wire goes over another wire. Horizontal and vertical wires do intersect when two wires touch, as in the next example (below, center), meaning the signal on the one wire splits and goes into two or more directions. The last example (below, right) uses both techniques. The "ribbon-cable" is the vertical wire. A signal on a horizontal wire goes under the ribbon-cable and travels off by itself. The "`/`" indicates that the signal also travels onto the ribbon-cable to join other signals. The ribbon-cable has multiple signals on it, which typically are logic signals headed for an OR block.

```
  │        ───┬───        │
──(──         │          /│
  │           │       ───(────
                          │
```

C-2.2.2. CONTACT. A contact is represented as a break in a signal wire. A logic signal is used to open or close the contact, which enables or disables the signal to pass along it's wire. The name of the logic signal that can open and close the contact is the name directly over the contact, or it is found by following the dots ("····") and wires from the contact to the signal name.

Normally Open Contact. A logic signal with a value of TRUE will close a normally open contact. The first symbol (below, left) shows the logic signal's name placed directly above the coil. The second symbol shows the logic signal entering from the top left and traveling right and down to the contact. The logic signal opens or closes the contact, and the signal traveling along the wire interrupted by the contact will pass when the logic signal is TRUE.

```
                 .
              ──┐
 L84TG            .
──┤ ├─        ──┤ ├─
```

Normally Closed Contact. A logic signal with a value of TRUE will open a normally closed contact. The first symbol (below, left) shows the logic signal's name placed directly above the coil. The next symbol (below, right) shows the signal entering from the top left and traveling down to the contact. The logic signal opens or closes the contact, and the signal traveling along the wire interrupted by the contact will pass when the logic signal is FALSE.

```
               .
             ──┐
 L84TG          .
──┤/├─       ──┤/├─
```

Contact Variation. One logic signal can be used to set multiple contacts.

```
─────┤   ├────┬────
       .      │
─────┤ / ├────┘
       .
_   ..  ─┘
```

C-2.2.3. COIL. If a coil is found at the end of a wire, the signal above the coil is set (written) with respect to the value of the signal on the wire. For the first "store" coil represented (below, left), a TRUE value on the wire will store a TRUE value at the coil's signal, and a FALSE will store a FALSE. For the second coil represented (below, right), the "store-inverse" coil, a TRUE value on the wire will store a FALSE value at the coil's signal, and a FALSE will store a TRUE.

```
  L4             L4
──( )          ──(I)
```

C-2.2.4. SIGNAL ARRAYS. An array is a group of similar signals, each with a possibly different value. Arrays are indicated using a lower case "n" in the signal's name. The "n" can be replaced with a number, incrementing up from zero as needed. Arrays are used to change functions, such as control curves, based on logic signals. Below, the array of L83JTn is used to decide which signal to use out of the TTKn_I array. The L83JTn array includes L83JT0, L83JT1, L83JT2, and so on; and the TTKn_I array includes TTK0_I, TTK1_I, TTK2_I, and so forth. If L83JT1 is TRUE and the rest of the array is FALSE, then TTK1_I is used.

```
TTKn_I
──────────┤ ├──
L83JTn    .
──────────┘
```

C-2.2.5. MASK BITS. Mask signals are used to "mask" other signals. Each mask signal is a 16-bit word made up of two bytes. The Most Significant Byte ("msbyte" or "MSB") is the left-most half, the Least Significant Byte ("lsbyte" or "LSB") is the right-most. For example, if the value of the mask signal was 04ED hexidecimal, then 04 is the value of the msbyte and ED is the value of the lsbyte. Each byte is then looked at in binary representation, such as 04 equals 0000 0100 and ED equals 1110 1101. The right-most bit is bit zero and the left-most bit is bit seven. The binary value is "laid out" over the mask, and each of these bits opens or closes a contact, enabling or disabling the passing of a signal (see Contacts — section C-2.2.2).

```
preset0    .     │
──────────┤ ├────┤
preset1    .     │
──────────┤ ├────┤
preset2    .     │
──────────┤ ├────┤
preset3    .     │
──────────┤ ├────┤
           .     └─
mask       .msbyte
───────────┤     ┌─
    lsbyte.      │
perm0      .     │
──────────┤ ├────┤
perm1      .     │
──────────┤ ├────┤
perm2      .     │
──────────┤ ├────┤
perm3      .     │
──────────┤ ├────┤
```

C-2.3. Boolean Logic

The following sections define and illustrate Boolean Logic symbols used in the BBLs.

C-2.3.1. NOT. The NOT logic symbol illustrated below (left) shows its input being inverted (a TRUE ["1"] input becomes a FALSE ["0"] and a FALSE input becomes TRUE). In the second example, the NOT symbol is represented by just the "o" on the middle logic input of this AND logic symbol. The middle logic signal's value is inverted before entering the AND symbol.

```
  ┌─┐            ┌───┐
──┤N├o──       ──┤   │
  └─┘          ─o┤AND├─
               ──┤   │
                 └───┘
```

C-2.3.2. OR If any logic input to an OR symbol is TRUE, the logic output is TRUE. The first symbol (below, left) shows two inputs. The second symbol (below, right) shows multiple logic signals ribbon-cabled into the OR symbol (see Wire Junctions section C-2.2.1 for more information concerning ribbon-cables).

```
  ┌──┐          ┌──┐
──┤  │        ┌─┤OR├─
  │OR├─       │ └──┘
──┤  │       /│
  └──┘     ──(──────
              │
             /│
           ──(──────
```

C-2.3.3. AND All of the logic inputs into an AND symbol must be TRUE for the logic output to also be TRUE. The symbols illustrated below show two inputs.

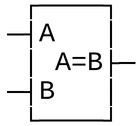
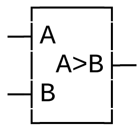
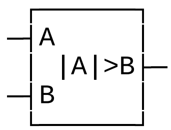
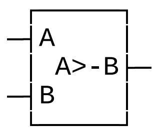
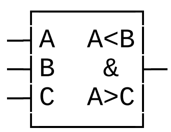
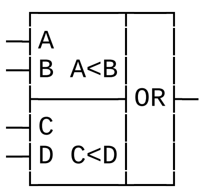
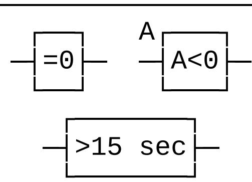
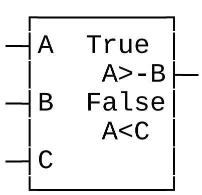
```
  ┌───┐        ┌─┐
──┤   │      ──┤A│
  │AND├─       │N├─
──┤   │      ──┤D│
  └───┘        └─┘
```

C-2.4. Miscellaneous Operations

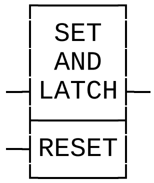
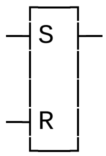
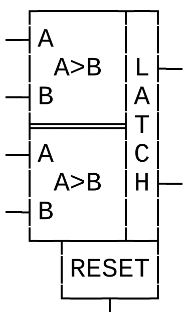
C-2.4.1. SELECTS. Selects may be used to output the minimum, maximum, or median value of the input values.

```
  ┌───┐       ┌───┐       ┌───┐       ┌───┐
──┤MIN│     ──┤MED│     ──┤MAX│     ──┤HI │
──┤SEL├─    ──┤SEL├─    ──┤SEL├─    ──┤SEL├─
  └───┘     ──┤   │       └───┘       └───┘
              └───┘
```

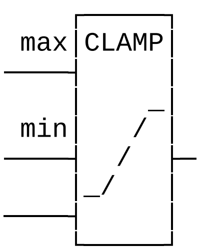
C-2.4.2. COMPARE. In a Compare block, two input signals are compared. The logic output signal is TRUE if the specified condition is met by the two inputs, otherwise the logic output signal is FALSE.

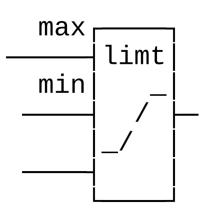
	Two input signals are compared. Being equal (=), . the condition is met so the logic output signal is true.
	One being greater than (>) or less than (<), or one being greater than or equal to (≥) or being less than or equal to (≤) the other. If the condition is met, the output signal is true.
	This example performs an absolute value operation on input A before comparison to input B.
	This example performs a negate operation on input B before comparison to input A.
	This example shows the logic output signal is set true when two conditions are met, A is less than B and A is greater than C, otherwise it is set false.
	Two compare blocks combined with an OR block. The logic output is set TRUE when either one of two conditions are met, A is less than B or C is less than D, otherwise it is set FALSE.
	The input is compared to a number specified in the block. If the condition is met, then the logic output is set TRUE, otherwise it is set FALSE.
	Two compare blocks are combined, where the logic output is set TRUE if A is greater than or equal to B, or set FALSE if A is less than or equal to C. The value of B is greater than the value of C to provide deadband.

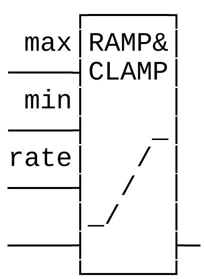
C-2.4.3. SET AND LATCH. If the logic input to a Set and Latch is TRUE, the logic output is TRUE, and is latched. This means that if the logic input becomes FALSE, the logic output will be held TRUE until the logic reset input is TRUE. If the logic input is still TRUE when the RESET is TRUE, the output logic will remain TRUE.

	If the logic input is TRUE, the logic output is TRUE and latched. This means that if the input becomes FALSE the output will remain TRUE until the reset input is TRUE.
	
	
	The logic output of each compare block is latched. The single reset logic input serves to reset (unlatch) all logic outputs.

C-2.4.4. CLAMP. An input signal to a Clamp is copied to the output signal. If the input should be too high or too low, the Clamp will keep the output less than or equal to "max", or greater than or equal to "min". A special case of the Clamp block is the Ramp and Clamp, where the output signal ramps to the input signal's value at the rate set by the "rate" variable, subject to normal clamping operation.







C-2.4.5. SINGLE SHOT. When the logic input to a single shot (SS) transitions from FALSE to TRUE, the output is set TRUE for one or more sequencing scan intervals, and then is automatically reset. The second example shows a compare combined with a Single Shot to determine if the Single Shot should react; if the value of input `A` is greater than `15 seconds`, the Single Shot operates.

```
  ┌──┐          ┌────┐
─┤SS├─       A│ SS │
  └──┘        ─┤A>15├─
               │secs│
               └────┘
```

C-2.4.6. TIME DELAY. A logic output from a Time Delay is TRUE only if the logic input has been TRUE for a time set by the time delay input. The last example (below, right) shows an OR block with a time delay. The OR block has multiple signals on the ribbon-cable, where the "`n`" is the number of signals.

```
                              │n
  ┌──┐        ┌───┐         ┌┴─┐
─┤td├─     ─┤DLY├─        │or│
  └─┬┘        └─┬─┘         │& │
                            ─┤td├─
                             └──┘
```

C-2.5. Math

The mathematics used to determine many of the functions is detailed in the following sections.

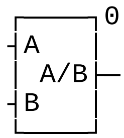
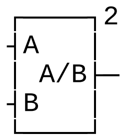
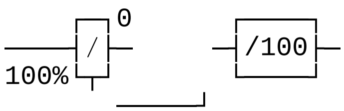
C-2.5.1. SUMMING JUNCTION. Input signals are added at the junction. If the sign is a "`+`", the positive value is added in, if it is a "`-`", the negative value is added in (it is subtracted). The sign associated with each input to a summing junction is located on the clockwise side of the input "wire."

```
  +                           │
───O───          ┌─┐       +│-
  +│            ─┤+│     ────O────
   │            ─┤+├─      +│
                ─┤-│        │
                 └─┘
```

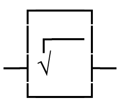
C-2.5.2. MULTIPLY JUNCTION. An input signal is multiplied by another input signal, the result is the output. The shift is shown at the upper right of the block.

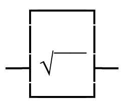
```
  ┌─┐-5
─┤x├──
  └┬┘
```

C-2.5.3. DIVIDE FUNCTION. An input signal is divided by another input signal, the result is the output. The shift is shown at the upper right of the block.

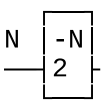
	This example shows the A input being divided by the B input which is then shifted by 0.
	This example shows the A input being divided by the B input and then shifted by 2.
	Two examples showing the input being divided by 100% and 100.

C-2.5.4. SQUARE ROOT. This block calculates the square root value of the input signal.

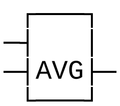




C-2.5.5. RAISE TO A POWER. The "Raise to a Power" block calculates a value of two (2) raised to a power of -N, where N is the input signal.



C-2.5.6. AVERAGE. The "Average" block calculates the average of the input signals. The example illustrated below has two inputs.



C-2.5.7. INTEGRATED TIME CONSTANT (ITC). The output value follows the input signal at an integrated rate (changing very fast if the difference is great, slower as the difference decreases). The rate at which the output changes is set by the time constant signal "T". The output is often double-word (32 bits), containing a Most Significant Word and a Least Significant Word (the MSW contains the desired value, the LSW gives accuracy). Some ITC blocks also include a RESET, where a logic input can be used to set the output equal to the input, bypassing the integrating operation. The relationship of input signals is specified by the equation shown in each block, regardless of variations in labelling.

```
 ┌───────┐                ┌───────┐                 ┌──────────┐    ┌──────────┐
 │  ITC  │      Time_Con  │  ITC  │                 │    ITC   │   ─┤T   V     ├─
─┤V      │      ──────────┤K      │                ─┤A    B    │    │   ────=OUT│
 │   V   │                │   V   │Intg_Out         │   ────=OUT├─ ─┤V  1+Ts   │
 │  ---- ├─               │  ---- ├────────        ─┤B  1+As   │    ├──────────┤
 │  1+Ks │      Intg_In   │  1+Ks │(MSW of)         ├──────────┤   ─┤RESET:OUT=V│
─┤K      │      ──────────┤V      │   [*2]         ─┤RESET:OUT=B│    └──────────┘
 └───────┘                └───────┘                 └──────────┘
```

C-2.5.8. ABSOLUTE VALUE. Represented by the signal name with a vertical line on each side, the output signal is a positive number, equal to the magnitude of the input number (in other words a negative number is made positive, and a positive number is left positive). In the fourth example, the positive value of A is used in a compare block. In the last one, the result of subtracting B from A is made positive.

```
─┤X├─      ┌───┐      ┌───┐      ┌───────┐        ┌───────┐
          ─┤|A|├─    ─┤ABS├─    ─┤A      │       ─┤A      │
           └───┘      └───┘      │ |A|>B ├──      │ |A-B| ├─
                                ─┤B      │       ─┤B      │
                                 └───────┘        └───────┘
```

C-2.5.9. DELTA. The amount of change of a signal since the last time this block was executed. "dt" represents the "delta time", the amount of time that has passed since the block was last executed. This can be used to add time to a timer, etc., or if a delta of a signal is divided by delta time, this represents the rate of change.

```
dt       ┌  ┐
──       │d │
        ─┤──├─
         │dt│
         └  ┘
```

C-3. MARK V MATH

To provide the fastest and most efficient microprocessor operation, the Mark V performs calculations in binary twos-complement math format. It is necessary to understand how these calculations are executed when making modifications to the control system. This section consists of three parts: Binary Number Review, Mark V Scaling of variables, and Application rules and notes. The binary math system the Mark V uses is standard throughout the computer industry and, as such, is widely available at colleges and universities if additional information is necessary.

C-3.1. Binary Number Review

The Mark V represents analog values as signed 16-bit binary numbers in the following form:

Most Significant Byte															Least Significant Byte
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Sign Bit	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
"Weight" of bit, which is also the power of 2															

Table C-1. Analog Value Representation

As shown in figure C-1, the most significant digit (the one furthest to the left) determines the sign of the binary number; zero indicates a positive and one indicates a negative value. The digits represent powers of 2, increasing from 2^0 at the rightmost digit up to 2^{14} at the immediate right of the sign bit. Each digit, called a bit, can take on values of zero or one; if a bit is one, it is referred to as "set" or "on."

The value of a positive binary number is simply the addition of the corresponding values of the set bits. These integer values are dimensionless and are referred to as COUNTS. For example:

0000 0001 0010 0001 = $2^8 + 2^5 + 2^0$ = 256 + 32 + 1 = 289 COUNTS
0100 0010 0001 1000 = 16,384 + 512 + 16 + 8 = 16,920 COUNTS

Therefore, the largest positive value that may be represented by a signed 16-bit integer will be a binary string of all one's, with a zero sign bit.

0111 1111 1111 1111 = 16,384 + 8192 + 4096 + 2048 + 1024 + 512+ 256 + 128 + 64 + 32 + 16+ 8 + 4 + 2 + 1 = 32,767 COUNTS

Negative binary numbers are handled differently than positive. To see the binary representation of a negative number, it is easiest to begin with its positive representation and negate it. To negate a positive binary number, invert all the bits and add one to the result.

```
     0000 0000 0000 0001       (+1)
     1111 1111 1111 1110       (inverted)
+    0000 0000 0000 0001       (add 1)
     1111 1111 1111 1111  =    -1
```

```
      0111 1111 1111 1111      (+32,767)
      1000 0000 0000 0000      (inverted)
  +   0000 0000 0000 0001      (add 1)
      1000 0000 0000 0001  =   -32,767
```

Conversely, to negate a negative number add -1 and invert the result.

```
      1111 1111 1111 1100      (-4)
  +   1111 1111 1111 1111      (add -1)
      1111 1111 1111 1011      (invert)
      0000 0000 0000 0100  =   +4

      1111 1111 1011 1101      (-67)
  +   1111 1111 1111 1111      (add -1)
      1111 1111 1011 1100      (invert)
      0000 0000 0100 0011  =   +67
```

The largest negative number which can be represented by a 16-bit integer is,

1000 0000 0000 0000 = -32,768 COUNTS

Notice that if -32,768 is negated, it does not behave properly.

```
      1000 0000 0000 0000      (-32,768)
  +   1111 1111 1111 1111      (add -1)
      0111 1111 1111 1111      (invert)
      1000 0000 0000 0000  =   -32,768
```

The valid range for numbers in the Mark V is -32,767 to +32,767. Negating -32,768 does not give +32,768 but rather -32,768. For this reason, -32,768 is considered an illegal value in the Mark V. Attempting to set an analog setpoint signal to -32,768 will result in a value of -32,767 for that signal. Similarly, attempting to set an analog setpoint to +32,768, will result in a value of +32,767. However, if an attempt is made to set an analog setpoint to a value larger in magnitude than +/-32,768, the value will be remain unchanged.

C-3.2. Scaling

Even though the Mark V may perform calculations with dimensionless integers, they represent physical values that may be viewed on the <I>. For this purpose, every point in the Control signal DataBase (CDB) has an associated scale type. Information pertaining to scale types is found in `ENGLISH.SCA`, `METRIC.SCA`, `CUSTOM.SCA`, and `HARDWARE.SCA` in the `F:\UNITn` directory. The currently selected scaling information can also be found in `SCLEDATA.DAT.` A sample of the scaling file `SCLEDATA.DAT` is shown below.

```
;   SCLEDATA.DAT
;
; This file defines engineering units conversions for each scale code.
; Each line contains a scale code number, gain, offset, decimal places
; and engineering units string.
#scale_data 11  128.000000    0.000000  1    DGA      ANGLE
#scale_data 61  512.000000    0.000000  1    MW       MWATT
#scale_data 83 2048.000000    0.000000  1    psi      PRESS
               (gain)         (offset)(dec) (units)   (type)
```

The gain listed for each scale type is the maximum value that a point of that scale type will achieve, as seen on the <I>. Both the gain and offset determine how the integer value of the Mark V will be displayed on the <I>. The conversion formula between the Mark V and the <I> are:

Information about the number of decimal places displayed is also found in the scaling data files. This will determine the resolution of the display on the <I>, whereas the gain determines the internal resolution in the Mark V. Consider the scale type `VIBVL`, which has a gain of eight and displays two decimal places with the units inches per second (in./sec). The internal resolution of the Mark V for the scale type `VIBVL` is:

$$\frac{\mathit{1\ Count}}{\mathit{32{,}768\ Counts}} * \frac{\mathit{8\ inches}}{\mathit{second}} = \frac{\mathit{0.00024\ inches}}{\mathit{second}}$$

Therefore, one integer Count in the Mark V is worth 0.00024 in./sec of physical vibration. By increasing the gain, the maximum value of that scale type will be increased, but the Mark V resolution will decrease. The scale type `VIBVL` displays 2 decimal places, corresponding to a minimum value of 0.01 in./sec. This determines the resolution of the <I> display. Continuing the above example:

$$\frac{0.01\ \frac{\mathit{inches}}{\mathrm{sec}}}{8\ \frac{\mathit{inches}}{\mathrm{sec}}} * \mathit{32{,}768\ Counts} = \mathit{41\ Counts}$$

Thus, every time the Mark V value increases 41 Counts a 0.01 in./sec change will be seen for the corresponding signal at the <I>.

Whatever the scaling of a signal happens to be, it has no direct effect on how the Mark V performs calculations with that signal. Consider how the Celsius and Fahrenheit temperature scales represent the point at which water freezes. Celsius represents this temperature as 0 °C and Fahrenheit as 32 °F. Even though the values and units differ, these temperatures both represent the same physical temperature. Therefore, the scaling information for°C will be different from °F, but this is only to **display** a different value on the <I>. However, scaling does indirectly affect how math primitives are implemented in the Mark V.

C-3.3. Mark V Applications and Rules for Math Primitives

Math primitives can be split into two categories: ADDs and MULTIPLYs. The ADDs include the primitives `ADD`, `SUB`, and `CMP`. The MULTIPLYs include the primitives `MPY`, `DVD`, and `SHIFT`. The `SHIFT` primitive is applicable to either group but is lumped with the MULTIPLYs because of the required shift input.

C-3-3.1. Rules for ADDs

When adding sequencing, keep in mind that the Mark V is operating with dimensionless integer values. The value seen on the <I> is actually a converted Mark V Count value. For example, the Mark V views 1024 °F (scale type `TC`) and 64% (scale type `PCT`) exactly the same, as 16,384 Counts. This is one reason why scaling is important in Mark V math.

Rule #1 Always ADD, SUBtract, or CoMPare identical scale types and scale the output the same way.
Rule # 2 Verify there is no overflow. The input and output magnitudes cannot exceed the maximum value of the output scale type.

To illustrate the importance of the first rule, consider a comparison (CMP) of two signals, both with units of "Hz," but scaled differently.

SIGNAL	VALUE	SCALE TYPE
frequency_1	27 Hz	(`FREQ`)
frequency_2	24 Hz	(`FRQLN`)

Examining the scaling file `SCLEDATA.DAT` shows that the gain of `FREQ` is 8192 and `FRQLN` has a gain of 64. Before comparing the signals, convert these values to Mark V values.

frequency_1 = (27 Hz/8192 Hz) * 32,768 Counts = 108 Counts
frequency_2 = (24 Hz/64 Hz) * 32,768 Counts = 12,288 Counts

Clearly, frequency_1 is greater than frequency_2, but the Mark V will not agree since 12,288 > 108.

The following example illustrates the importance of the second rule. In this example, two values are added (`INPUT1 and INPUT2`), both as `PCT` scale types with units "%."

SIGNAL	VALUE	SCALE TYPE
`INPUT 1`	58 %	`PCT`
`INPUT 2`	85 %	`PCT`

Following Rule #1, the output must be scaled as a `PCT` scale type. `PCT` has a gain of 128, and performing this addition will exceed the maximum value of the output scale type. The Mark V performs saturated math; that is, when the output overflows, the Mark V recognizes this and sets the output to the maximum Count value of 32767. Adding these two signals gives the erroneous answer of 128 %. If these signals are to be added, a new scale type will need to be defined with a larger gain and these values shifted to the new scale type. This may not be accomplished by increasing the gain of the `PCT` scale type to accommodate the larger values because existing scales are related to embedded functions within application BBLs, **do not change existing scales**. New scale types may be added to accomodate special calculations, however, the resulting signals may not be compatible with any existing functions. For example:

1. Define the new scale type PCTX with a gain of 256.0, an offset of 0.0 and units of "%" in the scaling (`*.SCA`) files and `SCLEDATA.DAT`.
2. Define two new points, `INPUT 1` and `INPUT 2`, into which the `PCT` scaled signals will be shifted.
3. Define a new signal, `INPUT3X`, as `PCTX` to receive the output.
4. Using the SHIFT primitive, perform a binary shift on the Mark V values.
   4.1. Add a SHIFT primitive block, with `INPUT1` as the input, `INPUT1X` as the output, and a +1 shift.
   4.2. Add a SHIFT primitive block, with `INPUT2` as the input, `INPUT2X` as the output, and a +1 shift.
5. The `PCTX` signals will now contain half the Counts as their `PCT` counterparts. With the gain doubled, however, no difference will be seen in their <I> values.
6. It is now possible to add `INPUT1X` and `INPUT2X` and get the correct answer (143%) for `INPUT3X` **which is not compatible with existing control functions**.

C-3.3.2. Rules for MULTIPLYs

As stated earlier, scaling does not directly affect how the Mark V performs calculations. However it does affect the design of sequencing. Scaling has a more profound effect on the MULTIPLY category of primitives, because scaling has a direct influence on calculating the shift value. The three primary rules are:

Rule #1. Insure anticipated units are correct for the output signal scale type.
Rule #2. Insure the magnitudes of the inputs and corresponding output do not exceed the maximum value of the output scale type.
Rule #3. Input a shift (required).

The first rule determines it is necessary that if a signal with units "°F/sec" is multiplied by a signal with units "sec" the output should be scaled to have units "°F". It may be necessary to define a new scale type (with appropriate units) to handle the output signal. It is important to "carry the units" through a calculation and make sure that the assigned scale types have units to match.

The second rule is the same as Rule #2 for ADDs. MULTIPLY primitives can cause the output signal to overflow.

MULTIPLY primitives also require a shift as an input. The correct shift is essential to get the correct answer for the output signal. Calculating the shift is simple, but *B* numbers need to be defined first.

Each scale type has an associated *B* number. The *B* number is determined by the gain of that scale type. The gain of a scale type is:

$$2^{B\ number}$$

The *B* number of a scale type locates the radix (binary decimal point) of a signal of that type, and allows for calculation of the shift. Notice that *B* numbers are integers only for gains that are powers of 2. There are instances in which a scale type will have a non-power of 2 gain, which would correspond to a non-integer *B* number. This situation is very common for the scaling information in `METRIC.SCA`. When using *B* numbers to calculate the shift, only integers should be handled. Every scale type has an integer *B* number, even though it may not be readily apparent from the scaling information. In the development of the scale types, all of the scaling information was calculated using English units. Once the gains were fixed in English units they were converted to Metric units, resulting in gains that are not integer powers of 2.

Gain	*B* Number
1	0
2	1
4	2
8	3
16	4
32	5
64	6
128	7
256	8
512	9
1024	10
2048	11
4096	12
.	.
.	.
.	.

Table C-2. B values

There are cases in which the gain is not a power of 2 for Metric or English scaling. There are a couple of possible explanations for this occurring. As was the case for conversion from English to Metric, some scales are converted from one

English type to another. For example, a scale type with units of "sec," with a gain of 16. This is clearly a scale depicting time. Another scale depicting time could have units of "min." The scale with "min" as units may have been derived from the scale with units of "sec" by dividing its gain by 60, and should be treated as a *B*4.

Another possibility is that the scale type has magnitude built into the units. A scale with units of Parts-Per-Million "PPM" with a gain of 976.6 can be explained in such terms. Since the units "PPM" have "per million" built in, this corresponds to a factor of 10^{-6}. The factor 2^{-10} is equal to 0.0009766. When this factor is multiplied by 10^6 the result is 976.6. The gain of the scale with units "PPM" is $2^{-10} \times 10^6$, and should have a *B* number equal to -10.

C-3.3.3. Shifts

- Calculating the Shift

Calculation of the shift is simple, once the concept of *B* numbers is known. For a multiplication operation,

SHIFT = *B* [output] - *B* [input_a] - *B* [input_b]

For a division operation,

SHIFT = *B* [quotient] - *B* [dividend] + *B* [divisor]

- Purpose for Shifts

The `MPY` and `DVD` primitives require a shift as an input. Additionally, the shift must have a scale type of `CNT15` for proper block execution.

To understand why shifts are necessary, it is necessary to understand how the Mark V performs multiplication. Consider a simple decimal example of a MULTIPLY. Two three-digit decimal integers are to be multiplied. The maximum value that may be stored in a three-digit decimal integer is 999. Therefore, the maximum value of the output is 999 * 999 = 998,001. Therefore six digits must be reserved for the output signal. If this multiplication is constrained to a three-digit output, there will be a problem with overflow when the inputs get too large.

Now consider a multiplication of two Mark V analog signals. For now, the scale types of the inputs will be ignored. The Mark V represents analog signals with 16-bit binary strings. When the Mark V multiplies the two inputs, both are 16-bit binary numbers (that is, the Counts representing the analog signals) will be multiplied to give the output signal, which is also defined as being 16 bits. As with the decimal example, the Mark V output signal will overflow if precautionary measures are not taken.

During the execution of a multiplication, the 16-bit binary numbers are multiplied and the result is stored in an intermediate 32-bit result. This eliminates the risk of overflow, since multiplying the 16-bit inputs will not exceed the 32-bit intermediate result. However, the answer will need to be reduced to 16 bits from the intermediate 32-bit answer. This is accomplished with the shift input. With the shift set to zero, the output signal will be set to the most significant 16 bits of the 32-bit intermediate result. Conversly, with the shift set to 16, the most significant 16 bits will be lost resulting in the output signal being the least significant 16 bits. Neither of these options will produce the desired result everytime, and the output "viewing window" (the 16 bits to be stored and used) must be shifted to achieve the correct answer. See the following example for the method of calculating the correct shift values.

- Multiply Example

Consider the multiplication of two signals, `ARG1` scaled as a `PCT_S`, and `ARG2` scaled as an `SEC16`. The scaling information for these two scale types is given below. Following rule 1, multiplying the units of the inputs gives "%": therefore the output, `OUT1`, needs to be scaled as a `PCT`. (See the third row in the table below.)

	Scale Code	Gain	Offset	Decimal Points	Units	Scale Type	Value
ARG1	73	16.0	0.0	3	%/sec	PCT_S	7.8 %/sec
ARG2	90	2048.0	0.0	1	sec	SEC16	12.2 sec
OUT1	68	128.0	0.0	2	%	PCT	95.1%

Table C-3. Scaling Information

These are the values as seen on the <I>, but inside the Mark V these are Counts. Following the procedure for <I> to Mark V value conversion:

$$ARG1 = \frac{7.8\,\frac{\%}{\text{sec}} - 0\,\frac{\%}{\text{sec}}}{16.0\,\frac{\%}{\text{sec}}} * 32767 = 15{,}973\ Counts \qquad ARG2 = \frac{12.2\ \text{sec} - 0\ \text{sec}}{2048.0\ \text{sec}} * 32767 = 195\ Counts$$

The Mark V then multiplies these two integers together to get the 32 bit intermediate result: 15,973 * 195 = 3,114,735 Counts. This intermediate result is larger than 32767 Counts, illustrating the need for the intermediate 32-bit answer. To calculate the shift for this example:

SHIFT = *B* [PCT] - *B* [PCT_S] - *B* [SEC16]
SHIFT = 7 - 4 - 11 = - 8

To see the importance of choosing the correct shift, view this intermediate result as a binary number:

3,114,735 = 0000 0000 0010 1111 1000 0110 1110 1111

With the shift set to zero, the output signal will be the most significant 16 bits (shaded bits shown below):

0000 0000 0010 1111 1000 0110 1110 1111

This corresponds to a value of 47 Counts or 0.2 %, not the correct answer. The shift input will actually move the bits in Mark V memory, but for the purpose of illustration this can be thought of as moving a "window" through which the output is viewed. The shift will "move" the 16-bit output "window" and increase, or decrease, the significance of the set bits. The 16-bit output window is "moved" left or right depending on the sign of the shift value. Beginning at the immediate right of the sign bit, count over the same number of digits as the shift, right for negative and left for positive shifts. Here, the output window is moved as follows:

0000 0000 0 010 1111 1000 0110 1 110 1111

which corresponds to a value of 24,333 Counts or 95.1%, which is the correct answer.

C-3.3.4. ERROR DETECTION The Mark V performs saturated math. Meaning, in the event of an overflow or error the signal is set to the maximum Count value (32,767). After the shift is performed, two tests are performed.

- First, the sign bit is tested for correctness. If the two inputs in the multiplication have like sign bits, a zero will be expected for the output sign bit. On the other hand, if the inputs have different sign bits, the sign bit should be a one. If this test fails, the Mark V saturates the signal.
- Secondly, if any of the bits to the left of the "post-shift" sign bit are a one, there was an overflow and the signal is saturated. This could be caused from the multiplication of inputs which are too large for the output scale type or because of an incorrect shift.

Valid shifts range from +15 to -31, but any shift value is allowed. The value +15 is the limit for left shifts because if the shift is +16 or greater, all significance of the answer will be lost. The same thing goes with right shifts, shifting -31 or more will shift all meaningful information out of the output window. Shifts outside the valid range will result in some erronous value being displayed.

C-4. SUMMARY OF DATA TYPES

L1 Logic state that may have a value of 1 or 0. A logic value may be forced to 0 or 1 via the logic forcing screen on the operator interface, <I>.

If the bit is forced, a "greater than" symbol (>) is displayed in front of the value. For example:

`signal_name`	`1`	`LOGIC`	the signal is not forced.
`signal_name`	`>1`	`LOGIC`	the signal is forced to a logic 1.
`signal_name`	`>0`	`LOGIC`	the signal is forced to logic 0.

C1 Raw count value with a range of 0 to 255 (eight bit word).

C2 Raw count value with a range of 0 to 65535 (16 bit word).

C4 Raw count value with a range of 0 to (2^{32} - 1) (32 bit word).

F2 A signed analog number whose applied value is controlled by a gain and/or offset value. The range for this data type in raw numbers is -32768 to + 32767.

S1 Logic State, 0 or 1.

S2 Enumerated state value
Enumerated states are usually presented as text, but stored and executed as a numeric value. For example, an enumerated state that represents the current status of a turbine could be:
0 stopped
1 on turning gear
2 rolling
3 at or above 70 % speed
4 at or above 95% speed
5 at 100% speed
6 decelerating
7 tripped
.
.
etc

The number values are stored and used in the execution of blocks; the text is assigned to the number in a look-up table. The text and/or value can be displayed on the screen. S2 has a range of 0 to 65535.

H1, 2, 4 Hexadecimal value, H1 is 2 digits (range 0 to FF), H2 is 4 digits (0 to FFFF), and H4 is 8 digits.

X2 Extended representation (Double Precision Register). Most significant word will typically be displayed. This data type has no implied conversion.

V1 Various typed single byte variable, will accept any single byte data type assignment (that is, L1, C1).

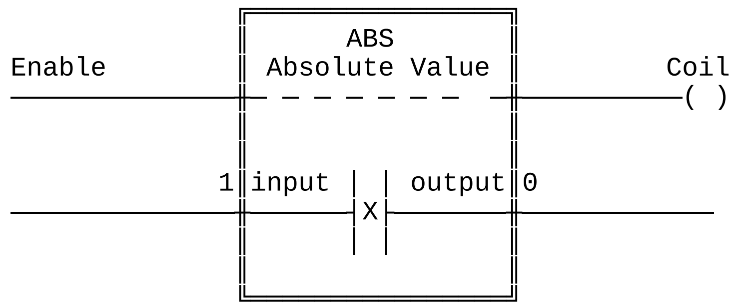
V2 Various typed analog variable, will accept any two byte data type assignment (that is, F2, X2, C2, and so on).

K2 Word constant value, calculated and supplied by the sequencing editor, to define the number of data definitions used by iterative generic blocks.

A2 Numeric constant value, supplied by the user, to define a specific reference number; this is not a database point type definition.

C-5. PRIMITIVE BIG BLOCK PROGRAMMING TOOLS

C-5.1. ABS -- ABSOLUTE VALUE



Description

Execution of the Absolute Value (ABS) block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block performs an absolute value function on the input value and passes the result to the output.

Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines block execution mode. A 1 will permit function execution; a zero (0) will leave the output unchanged.
input	F2	Analog quantity whose absolute value will be passed to the output.

Outputs

Parameter	Data Type	Description:
output	F2	Absolute value of the input figure.
Coil	L1	Logical output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.2. ADD

```
                    ╔══════════════════╗
                    ║       ADD        ║
 Enable             ║                  ║          Coil
────────────────────╫─ ─ ─ ─ ─ ─ ─ ────╫──────────( )
                    ║                  ║
                   1║input_a +  output ║0
────────────────────╫─────────O────────╫──────────
                    ║        +|        ║
                   2║input_b  |        ║
────────────────────╫─────────┘        ║
                    ╚══════════════════╝
```

Description

Execution of the Add block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block executes an addition operation. Analog inputs a and b are added and their sum is passed to the output. For proper operation, both inputs and the output must be scaled identically.

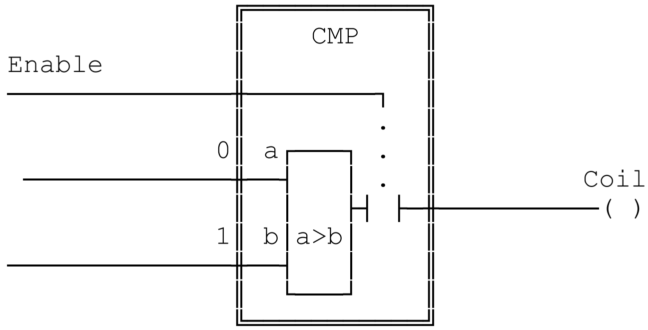
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines block execution mode. A one (1) will permit function execution; a zero (0) will leave output a + b unchanged.
input_a	F2	Analog value that is added to input b.
input_b	F2	Analog value that is added to input a.

Outputs

Parameter	Data Type	Description:
output	F2	Analog sum of input values a and b.
Coil	L1	Logic output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.3. CMP or CMP_A>B -- COMPARE A > B



Description

The CMP (CMP_A>B) block performs a compare operation on analog inputs a and b. Execution of the block is controlled by the Enable logical input. When the Enable input is a logic one (1) and the value of analog input a is greater than input b, the Coil is set to a logic one (1).

Note: this block may be specified as either CMP or CMP_A>B.

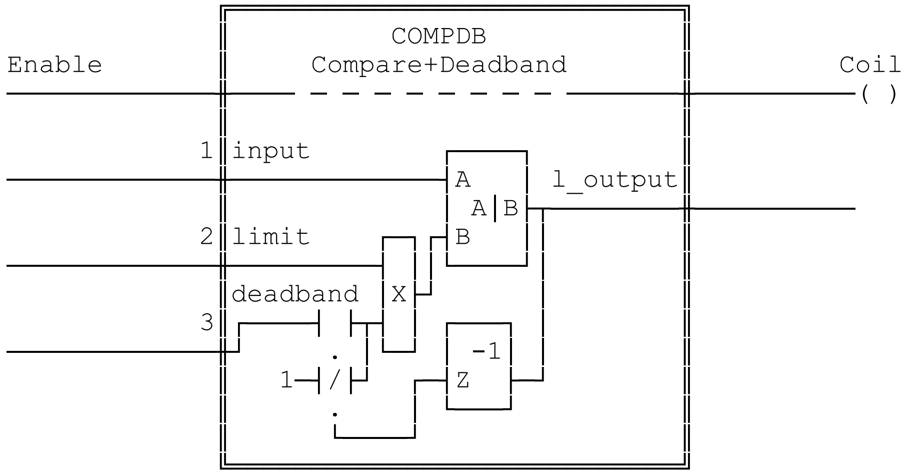
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines whether the function will execute. A logic one (1) will permit block function execution; A zero (0) will set the output to zero (0).
a	F2	Analog value to which input b is compared.
b	F2	Analog value that is compared to input a.

Outputs

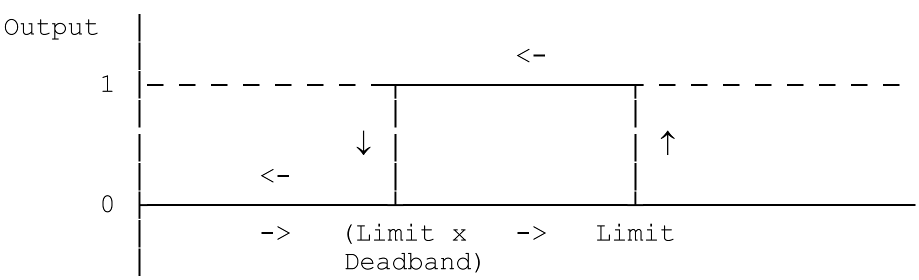
Parameter	Data Type	Description:
Coil	L1	Output that is set to one (1) when input a is greater than input b.

C-5.4. COMPDB -- COMPARE WITH DEADBAND



Description

Execution of the Compare with Deadband block (COMPDB) is controlled by an Enable logical input, which is passed through to the Coil output when the block is executed. The block implements an analog compare operation with a deadband calculation after the compare limit has been exceeded. Starting with the logical output parameter l_output reset, the input value is compared with the limit (input and limit values must be scaled identically). l_output remains reset while the input is less than the limit. As soon as this input exceeds the limit, the l_output is set. l_output remains set until the value of the input drops below the limit value times the deadband value. The deadband value is always between zero (0) and one (1) and should always be expressed as a B0 number.



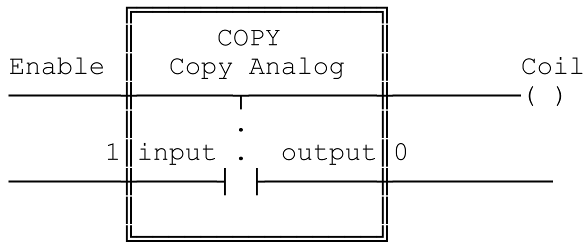
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines whether the function will execute. A one (1) will permit function execution; a zero (0) will prevent execution and leave the output unchanged.
input	F2	Value compared against the limit below.
limit	F2	The input value above which l_output will initially be set.
deadband	F2	The multiplier from which the second (return to zero) limit is calculated.

Outputs

Parameter	Data Type	Description:
l_output	L1	The logical result of the comparison. When enabled, transitions from zero (0) to one (1) when the input becomes greater than the limit, then back to zero (0) when the input drops below limit deadband.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.5. COPY



Description

Execution of the Copy block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block passes the analog input to the output.

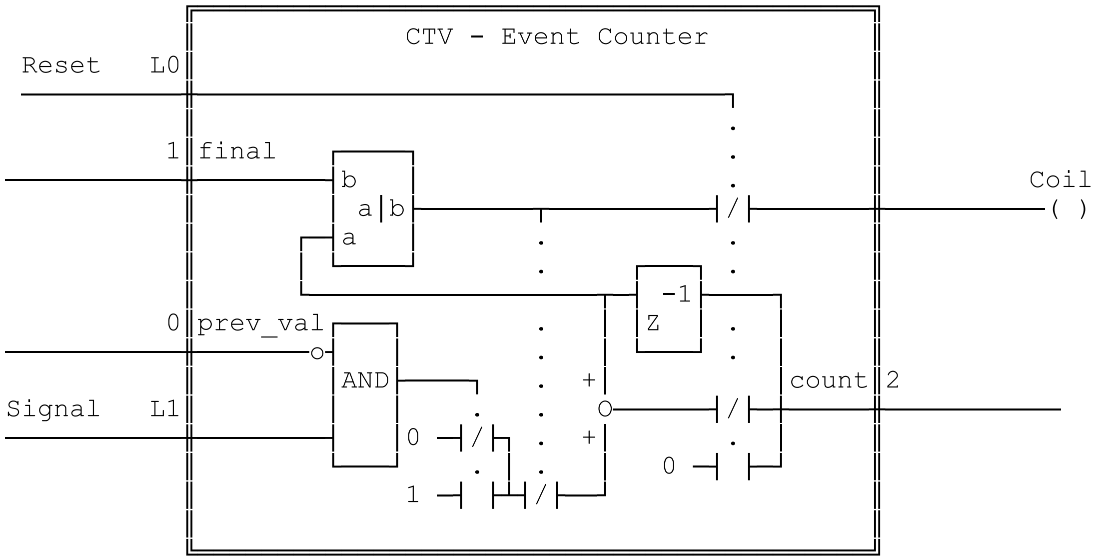
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit block execution; a zero (0) will leave the output unchanged.
input	V2	Analog value that is passed to the output.

Outputs

Parameter	Data Type	Description:
output	V2	Analog value that is passed from the input.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.6. CTV -- EVENT COUNTER



Description

The CTV block executes a count operation when the logic Signal input transitions from zero (0) to one (1). When a transition occurs, the event count is incremented by one. To recognize the next transition in the logic signal, the new logic input is stored in the prev_val parameter. If the count reaches a point equal to the final value, the output Coil is set to one (1). The event count and the Coil output are reset to zero (0) when a logic one (1) is received from the Reset input.

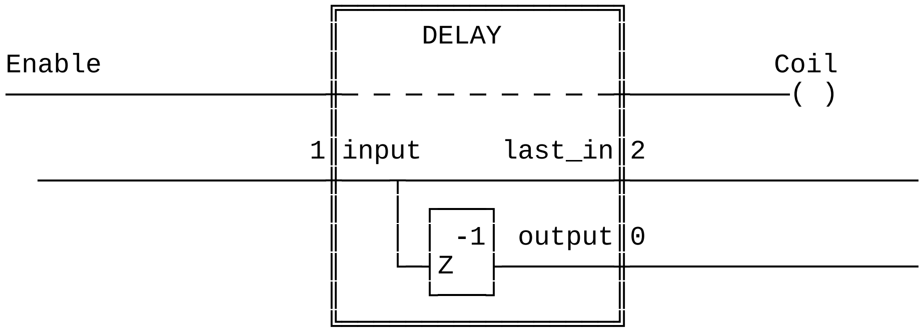
Inputs

Parameter	Data Type	Description:
Reset	L1	Logic input (1) that resets the event counter and the Coil output.
Signal	L1	The logic input whose zero (0) to one (1) transitions are counted.
prev_val	L1	Logic parameter use to store the previous value of the Signal input.
final	C2	The block will set the Coil output to one (1) when the counter reaches a value equal to this figure.

Outputs

Parameter	Data Type	Description:
Coil	L1	Output that is set to one (1) if the counter figure reaches a quantity equal to the final value. It will be reset when a logic one (1) is received from the Reset input, and remains reset until the count again exceeds the final value.
count	C2	Records number of 0 to 1 transitions in logic value of the Signal input.

C-5.7. DELAY



Description

Execution of the Delay block is controlled by the Enable logical input, which is passed through to the output when the block is executed. The Delay block performs a single sweep delay function on analog input data before it is passed to the output. The previous value is held (last_in) until the next execution of the block. When the next execution occurs, the value is passed to the output.

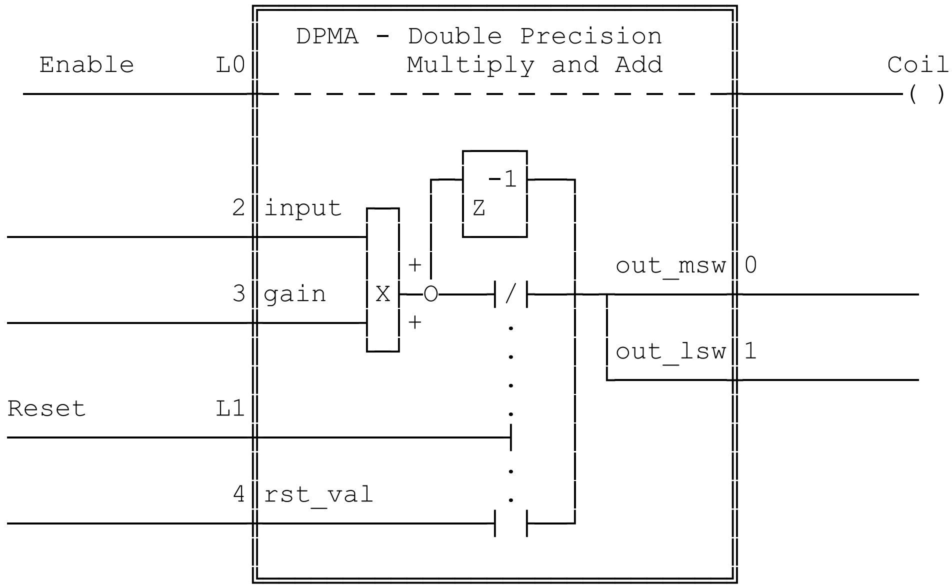
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution (enable set); a zero (0) will leave the output unchanged.
input	F2	Analog value that is stored in the Temp storage location.

Outputs

Parameter	Data Type	Description:
output	F2	Analog value that has been stored for one execution of the block.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).
last_in	F2	Temporary storage location that holds the value of the input for one execution of the block before it is passed to the output.

C-5.8. DPMA -- DOUBLE PRECISION MULTIPLY AND ADD



Description

Execution of the Double Precision Multiply and Add Block (DPMA) is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block executes two mathematical operations. The initial operation multiplies the input times the gain. The second takes the result of this multiplication and adds it to a running sum that is tallied in a double precision register consisting of the two 16 bit words out_msw and out_lsw (Note: The CRT screen will normally only display the most significant word). This sum is reset to rst_val when the Reset logic signal to the block reads one (1). The next execution of the block (with Reset = 0) will see the count in the double precision register begin at this point.

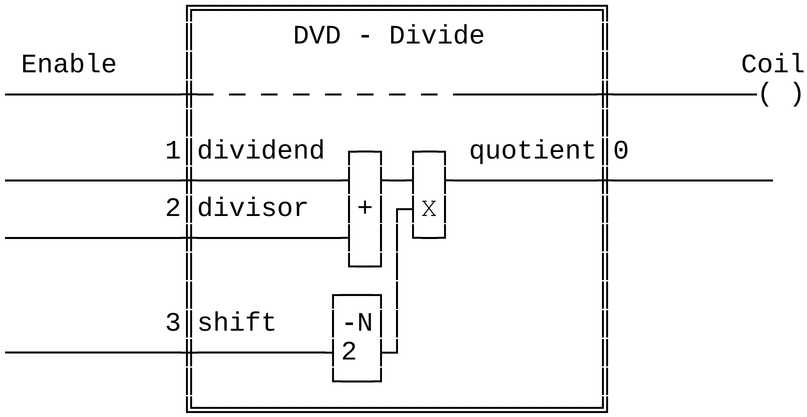
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines block execution mode. A one (1) will permit function execution; a zero (0) will leave the outputs unchanged.
input	F2	Analog value that is multiplied by the gain.
gain	F2	Multiplication factor (variable) by which the input will be multiplied.
rst_val	F2	Analog value to which the Msw of the running sum will be reset if the block receives a logic one (1) Reset input (the Lsw is reset to 0).
Reset	L1	The running sum will be reset if a logic one (1) is received on this input.

Outputs

Parameter	Data Type	Description:
out_msw	F2	The most significant word of the output. The upper 16 bit portion of a the 32 bit register that stores the relevant portion of the output.
out_lsw	X2	The least significant word of the output. The lower 16 bit portion of the 32 bit register that allows the block to maintain added resolution.
Coil	L1	Output that is set to one (1) if the block is executed (enable set); otherwise, it will be set to zero (0).

C-5.9. DVD -- DIVIDE



Description

Execution of the Divide block is controlled by the Enable logical input, which is passed through to the Coil output when the function is executed. The block executes a divide operation using analog inputs for the dividend and divisor. The binary scaling of the result (quotient) is then modified by a shift function before it is passed to the output. Dividing analog values causes a shifting of the binary point in the result. For example, a B6 number divided by a B2 number will result in a B4 number. The shift constant can be used to modify the binary point location in the output as needed.

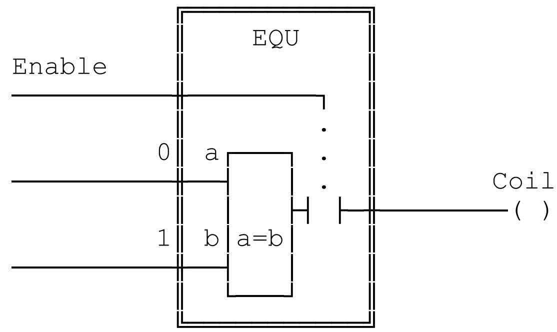
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines block execution mode. A one (1) will permit function execution; a zero (0) will leave the quotient unchanged.
dividend	F2	Analog input that is divided by the input value of the divisor.
divisor	F2	Analog input divisor by which the dividend is divided.
shift	F2	A constant that modifies the binary point location of the quotient.

Outputs

Parameter	Data Type	Description:
quotient	F2	Analog value that is the result of the division of the dividend by the divisor.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.10. EQU OR CMP_A=B -- COMPARE A = B



Description

The EQU (CMP_A=B) block performs a compare operation on analog input a and b. Execution of the block is controlled by the Enable logical input. When the Enable input is a logic one (1) and analog inputs a and b are equal, the Coil is set to a logic one (1).

This block may be specified as either EQU or CMP_A=B.

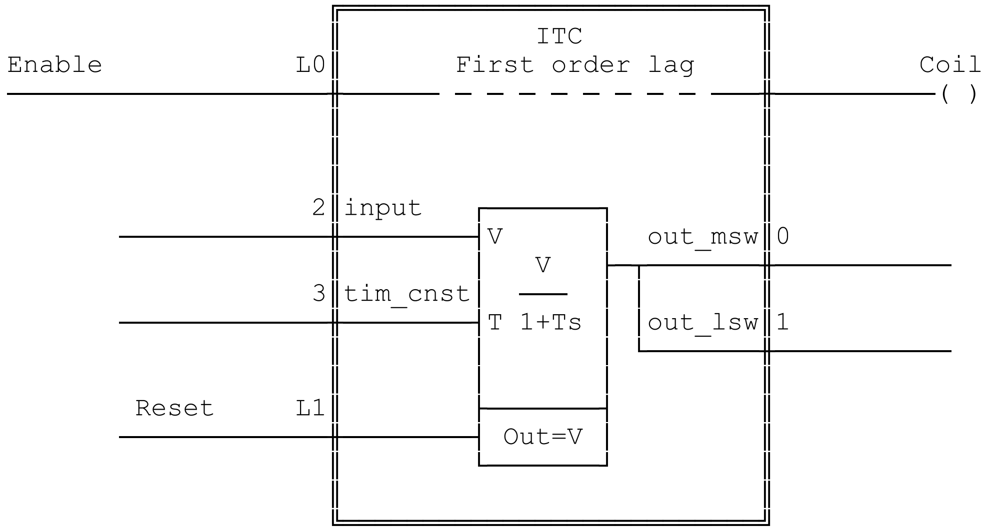
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines whether the function will execute. A one (1) will permit function execution; a zero (0) will set the Coil to zero (0).
a	V2	Analog value that is compared to input b.
b	V2	Analog value that is compared to input a.

Outputs

Parameter	Data Type	Description:
Coil	L1	Output that is set to one (1) when the value of input a equals the value of input b.

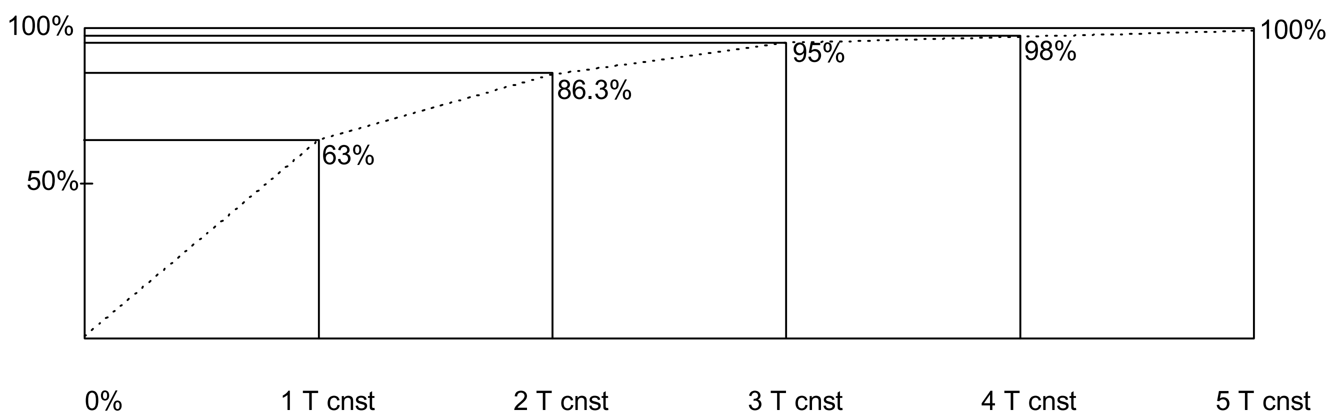
C-5.11. ITC -- INTEGRAL TIME CONSTANT



Description

Execution of the Integral Time Constant block is controlled by the Enable logical input—which is passed through to the Coil output when the block is executed. The ITC block executes a lag function. The value of the output approaches the value of the input at a rate determined by the time constant (input and output data must be scaled identically). The difference between the input and output values represents 100% of the change of the output. The time constant represents the time needed to raise (or lower) the output 63% of the difference between the input and output. Therefore, it takes five (5) time constants to change the output approx. 100% of the difference (1 sec.= 63% of 100%, 2nd sec. = 63% of 37%, 3rd sec. = 63% of 13.7%, 4th sec. 63% of 5%, 5th sec. = 63% of 2%). (See example below.)

Example: Input = 100% Output = 0% Tim_Cnst = 1sec.



Output figures are stored in a dual precision register consisting of the two 16 bit words out_msw and out_lsw (Note: the CRT screen will normally display the most significant word). A logic 1 on the Reset logical sets the output to the current input value immediately. The time constant value should be at least four times the sequencing scan interval.

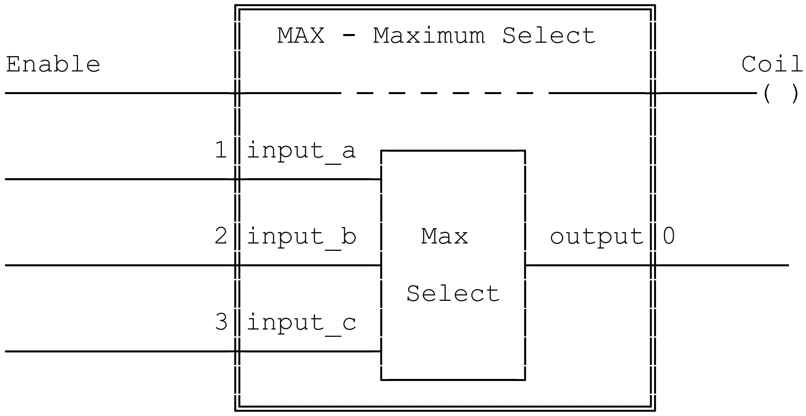
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
in	F2	Analog value that is used to calculate the current value of the output.
tim_cnst	F2	Analog value that represents the time needed to raise/ lower the output 63% of the current difference between the input and output.
Reset	L1	A one (1) on the reset logical sets the output to the current input value immediately.

Outputs

Parameter	Data Type	Description:
out_msw	F2	The most significant word of the output is a 16 bit portion of a 32 bit register that stores the relevant segment of the output.
out_lsw	X2	The least significant word of the output is a 16 bit portion a 32 bit register that allows the block to maintain added resolution.
Coil	L1	Output is set to one (1) when the function is executed (enable set): otherwise, it will be set to zero (0).

C-5.12. MAX -- MAXIMUM SELECT



Description

Execution of the Maximum Select block is controlled by the Enable logical input, which is passed through to the Coil output as when the block is executed. The block implements a comparison of three analog inputs. The largest of these inputs is selected and passed to the output. For proper operation, all three inputs and the output must be scaled identically.

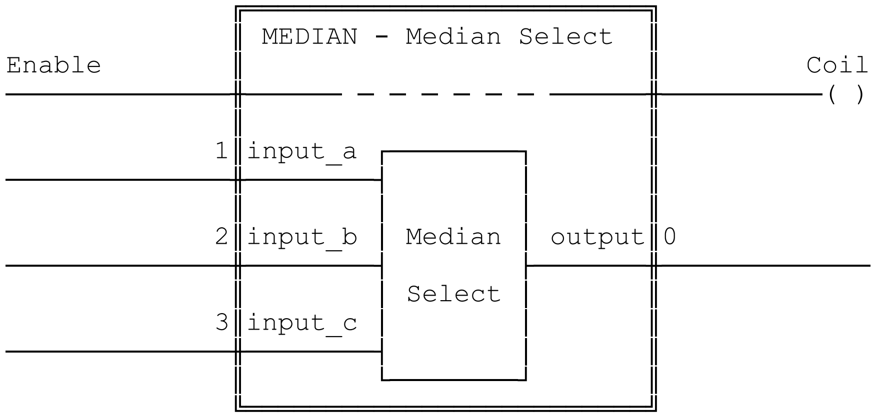
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
input_a	F2	Analog value that is compared to inputs b and c.
input_b	F2	Analog value that is compared to inputs a and c.
input_c	F2	Analog value that is compared to inputs a and b.

Outputs

Parameter	Data Type	Description:
output	F2	Analog quantity set to the greatest value of inputs 1, 2, and 3.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.13. MEDIAN -- MEDIAN SELECT



Description

Execution of the Median Select block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block implements a comparison of three analog inputs. The middle value of the three is selected and passed to the output. For proper operation, all three inputs and the output must be scaled identically.

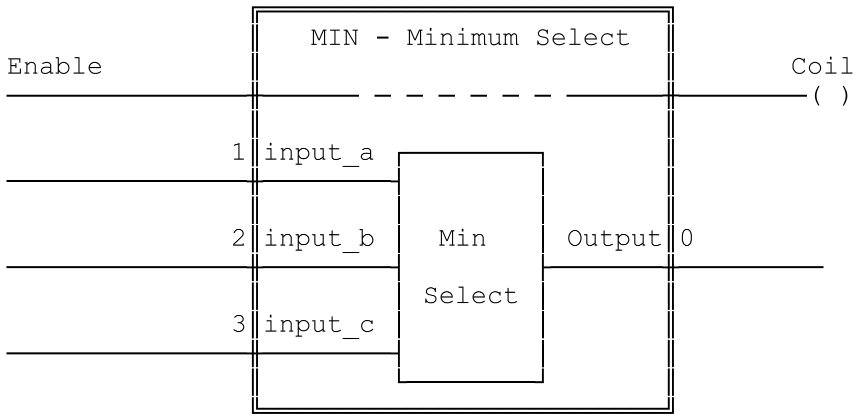
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines block execution mode. A 1 will permit function execution; a 0 will leave the output unchanged.
input_a	F2	Analog value that is compared to inputs b and c
input_b	F2	Analog value that is compared to inputs a and c.
input_c	F2	Analog value that is compared to inputs a and b.

Outputs

Parameter	Data Type	Description:
output	F2	Analog quantity set to the median value of inputs a, b, and c.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.14. MIN -- MINIMUM SELECT



Description

Execution of the Minimum Select block is controlled by the Enable logical input, which is passed through to the Coil when the block is executed. The block implements a comparison of three analog inputs. The smallest of these inputs is selected and passed to the output. For proper operation, all three inputs and the output must be scaled identically.

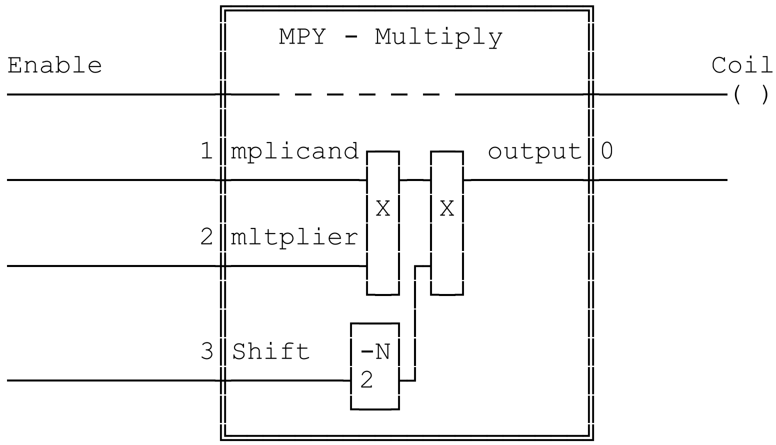
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
input_a	F2	Analog value that is compared to inputs a and c.
input_b	F2	Analog value that is compared to inputs A and C.
input_c	F2	Analog value that is compared to inputs a and b.

Outputs

Parameter	Data Type	Description:
output	F2	Analog quantity set to the smallest value of inputs a, b, and c.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.15. MPY -- MULTIPLY



Description

Execution of the Multiply block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block executes a multiplication operation on analog input values mplicand and mltplier. The binary scaling of the result is then modified by a shift function before it is passed to the output. Multiplication of the input variables causes a shifting of the binary point in the result. For example, a B3 number multiplied by a B5 number results in a B8 number. The shift constant can be used to modify the binary point location in the output.

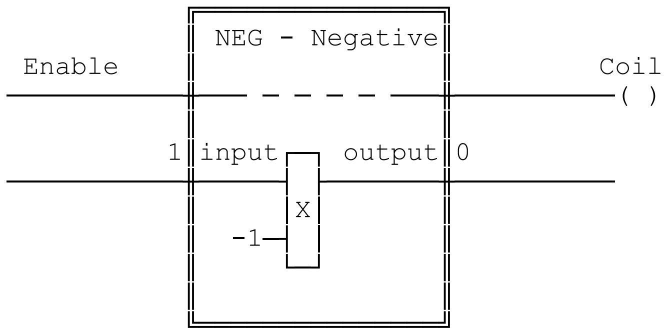
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
mplicand	F2	Analog value that is multiplied by input mltplier (multiplier).
mltplier	F2	Analog value that is multiplied by input mplicand (multiplicand).
shift	F2	A constant that modifies the binary point location of the output.

Outputs

Parameter	Data Type	Description:
output	F2	The result of the multiplication of the inputs with a binary point adjustment determined by the shift input.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.16. NEG -- NEGATIVE



Description

Execution of the Negative (NEG) block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block multiplies analog input data by -1 before it is passed to the output.

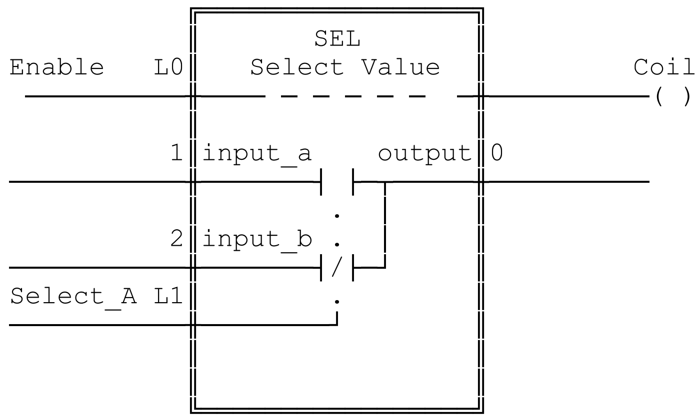
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
input	F2	Analog value that is multiplied by -1 before it is passed to the output.

Outputs

Parameter	Data Type	Description:
Coil	L1	Output that is set to one (1) if the block is executed (enable set); otherwise, it will be set to zero (0).
output	F2	Analog value that has been multiplied by -1.

C-5.17. SEL -- SELECT



Description

Execution of the Select block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. When Select_A is false, the block passes the analog value of input_b to the output. However, when the block receives a logic 1 input on Select_A, input_a is passed to the output.

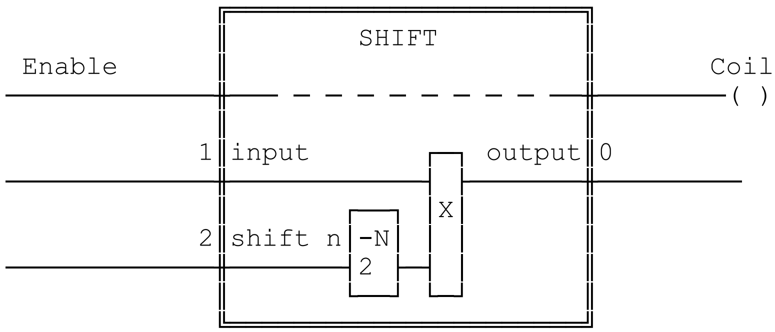
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
Select_A	L1	When false, the Select_A logical input passes the value of input_b to the output; otherwise, the value of input_a is passed to the output.
input_a	V2	Analog value that replaces input_b as the output when the Select_A logic input reads one (1).
input_b	V2	Analog value which the block will pass to the output when a logic 0 signal is received on the Select_A input.

Outputs

Parameter	Data Type	Description:
Output	V2	Analog value selected from either input_a or input_b.
Coil	L1	Output that is set to one (1) if the function is executed; otherwise, it will be set to zero (0).

C-5.18. SHIFT



Description

Execution of the Shift block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block executes a shift function on analog input data. Input values have their binary points modified according to the shift constant before they are passed to the output.

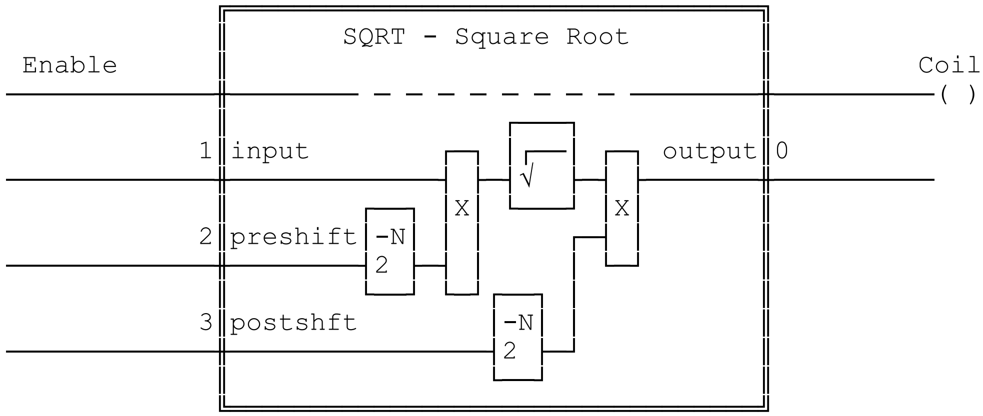
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
input	F2	Analog value that has its binary point modified by a shift function before it is passed to the output.
shift	F2	A constant that modifies the binary point location of the input before it is passed to the output.

Outputs

Parameter	Data Type	Description:
output	F2	Analog value resulting from a binary point modification of the input by the shift constant.
coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.19. SQRT -- SQUARE ROOT



Description

Execution of the Square Root block is controlled by the Enable logical input, which is passed to the Coil output when the block is executed. The block executes a preshift operation that insures that the input value is assigned an odd binary point. Analog input values must be assigned odd binary points before a square root function can be performed The preshift operation should be used to assign a higher binary point to the input value to prevent the loss of significant data. Following the preshift operation, the block executes a square root function. The binary point of the result of this operation is assigned using the following equation before the value is passed to the output:

$$BP\ Output\ =\ \left[\frac{BP\ Input\ +\ 1\ +\ Preshift}{2}\right]\ +\ Postshift$$

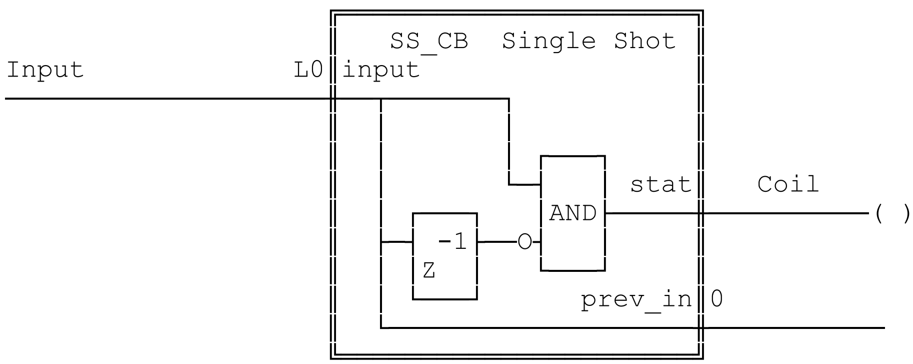
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
input	F2	Analog value that is preshifted before the square root function is executed.
preshift	F2	A constant that assigns an odd binary point to the input before the square root function occurs.
postshft	F2	A constant that assigns a desired binary point to the result of the square root function before it is passed to the output.

Outputs

Parameter	Data Type	Description:
output	F2	The result of the Square Root operation after the postshift function has been executed.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.20. SS_CB -- SINGLE SHOT



Description

The SS_CB block executes a single shot function. A logic TRUE (1) will be returned on the coil output for one execution cycle when the input becomes a logic TRUE (1). Otherwise the output returns FALSE (0).

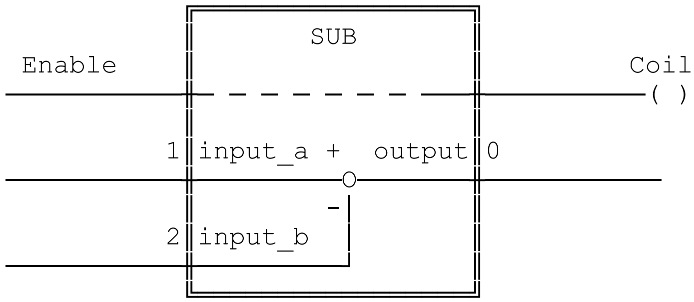
Inputs

Parameter	Data Type	Description:
input	L1	Logical signal that is used to trigger the single shot function.
prev_in	L1	Logical signal used for starage of the previous value of the logical input.

Outputs

Parameter	Data Type	Description:
coil	L1	Logical output that is set true for one execution cycle when the input becomes true. Otherwise false is returned.

C-5.21. SUB -- SUBTRACT



Description

Execution of the Subtract block is controlled by the Enable logical input, which is passed through to the Coil output when the block is executed. The block executes a subtraction operation. The analog value of input_b is subtracted from input_a. The difference between the two inputs is then passed to the output. For proper operation, both inputs and the output must be scaled identically.

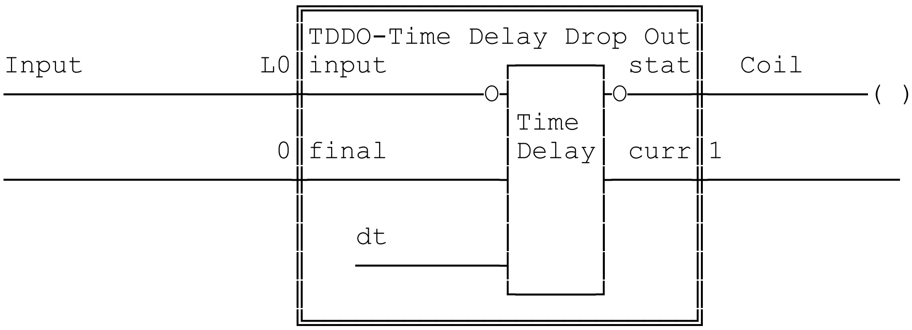
Inputs

Parameter	Data Type	Description:
Enable	L1	The Enable logical input determines the block execution mode. A one (1) will permit function execution; a zero (0) will leave the output unchanged.
input_a	F2	Analog value from which input_b will subtracted.
input_b	F2	Analog value that is subtracted from input_a.

Outputs

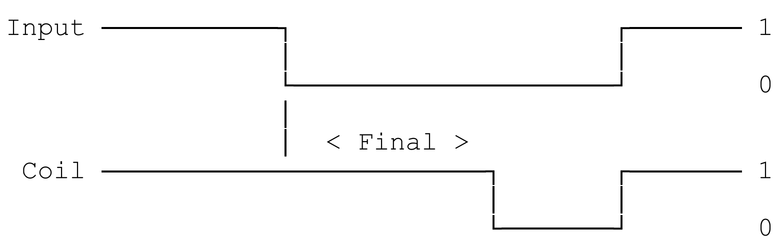
Parameter	Data Type	Description:
output	F2	Analog difference between input_a and input_b.
Coil	L1	Output that is set to one (1) if the function is executed (enable set); otherwise, it will be set to zero (0).

C-5.22. TDDO -- TIME DELAY DROP OUT



Description

The TDDO block performs a time delay function. Execution of the block is controlled by the logical input. When a logic zero (0) is received, the timer function begins executing. When the input signal transitions from one (1) to zero (0), elapsed time (sec.) is tabulated in the current counter. When the total in the counter equals the value of the final constant, the Coil output is set to zero (0) (Note: both final constant and current counter should be scaled identically). A true input signal will reset the current counter and the Coil output to one (1) immediately.



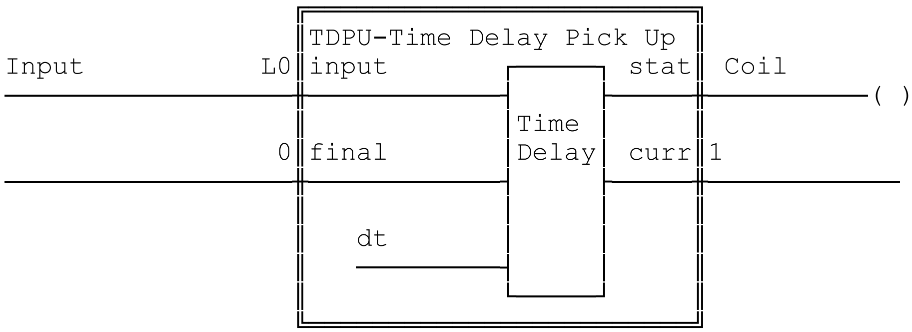
Inputs

Parameter	Data Type	Description:
Input	L1	The input determines the block execution mode. A zero (0) will permit function execution; a one (1) will reset the counter and the Coil output to 1.
final	F2	A time duration constant that when reached by the counter, will permit the Coil output to be set to zero (0).

Outputs

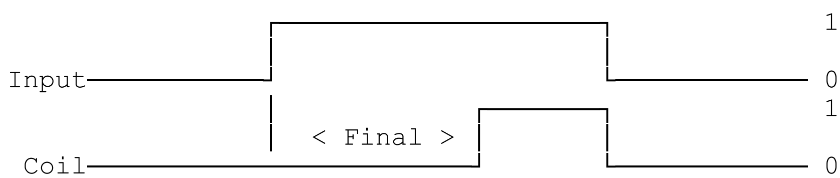
Parameter	Data Type	Description:
Coil	L1	Output that is set to zero (0) when the amount in the current counter equals the Final value.
curr	F2	Records elapsed time (sec.) after a logic 0 is received from the Input. Maximum recordable time is determined by system scaling of time in seconds.

C-5.23. TDPU -- TIME DELAY PICK UP



Description

The TDPU block performs a time delay function. Execution of the block is controlled by the logical input. When a logic one (1) is received, the timer function begins executing. When the input signal transitions from zero (0) to one (1), elapsed time (sec.) is tabulated in the current counter. When the total in the counter equals the value of the final constant, the Coil output is set to one (1) (Note: both final constant and current counter should be scaled identically). A false input signal will reset the current counter and the Coil output to zero (0) immediately.



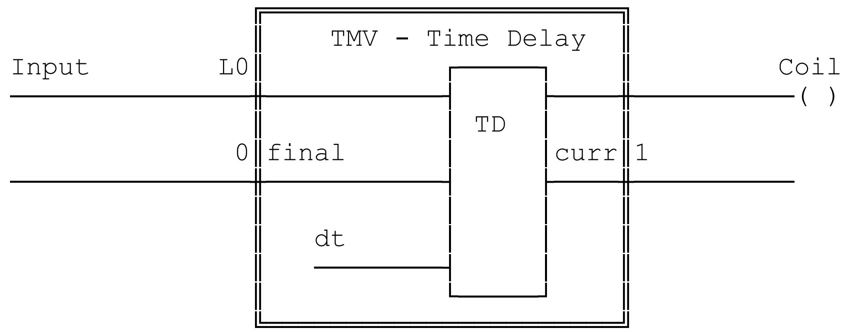
Inputs

Parameter	Data Type	Description:
Input	L1	The input determines the block execution mode. A one (1) will permit function execution; a zero (0) will reset the counter and the Coil output to 0.
final	F2	A time duration constant that when reached by the counter, will permit the Coil output to be set to one (1).

Outputs

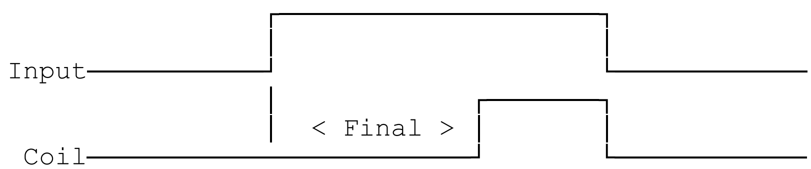
Parameter	Data Type	Description:
Coil	L1	Output that is set to one (1) when the amount in the current counter equals the Final value.
curr	F2	Records elapsed time (sec.) after a logic 1 is received from the Input. Maximum recordable time is determined by system scaling of time in seconds.

C-5.24. TMV -- TIME DELAY



Description

The TMV block performs a time delay function. Execution of the block is controlled by the logical input. When a logic one (1) is received, the timer function begins executing. When the input signal transitions from zero (0) to one (1), elapsed time (sec.) is tabulated in the current counter. When the total in the counter equals the value of the final constant, the Coil output is set to one (1) (Note: both final constant and current counter should be scaled identically). A false input signal will reset the current counter and the Coil output to zero (0) immediately.



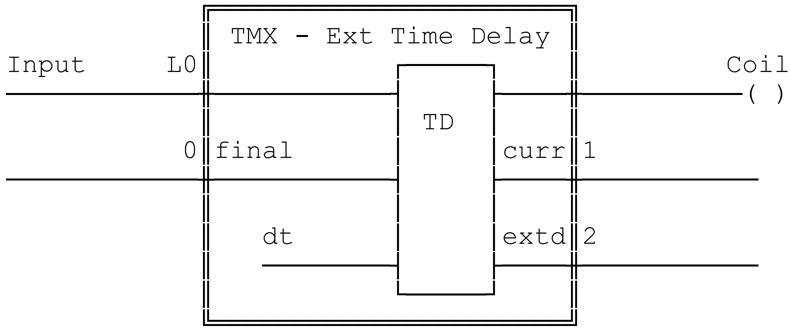
Inputs

Parameter	Data Type	Description:
Input	L1	The input determines the block execution mode. A one (1) will permit function execution; a zero (0) will reset the counter and the Coil output to 0.
final	F2	A time duration constant that when reached by the counter, will permit the Coil output to be set to one (1).

Outputs

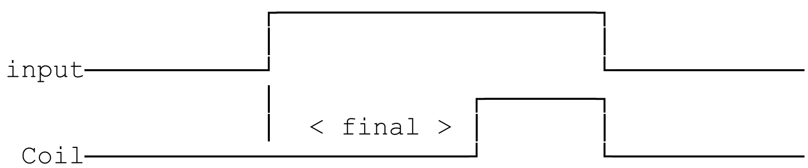
Parameter	Data Type	Description:
Coil	L1	Output that is set to one (1) when the amount in the current counter equals the Final value.
curr	F2	Records elapsed time (sec.) after a logic 1 is received from the Input. Maximum recordable time is determined by system scaling of time in seconds.

C-5.25. TMX -- TIME DELAY (EXTENDED)



Description

The TMX block performs an extended (longer range) time delay function. When a logic 1 is received, the block begins executing. When the input signal transitions from zero (0) to one (1), elapsed time is tabulated in the extended counter; periodically, the extended counter resets to zero (0) and a count of one will be added to the current counter. When the total in the current counter equals the value of the final constant (these two values should be scaled identically), the Coil output is set to one (1). A False input signal will reset the current and extended counters, and the Coil output to zero (0) immediately.



Inputs

Parameter	Data Type	Description:
Input	L1	The Input determines when the time delay function begins executing. A one (1) will permit function execution. A zero (0) will reset the counter and the Coil output to zero (0).
final	F2	A time value that when reached by the counter, will permit the Coil output to be set to one (1).

Outputs

Parameter	Data Type	Description:
Coil	L1	Output that is set to one (1) when the amount in the current counter equals the final value.
extd	V2	Records fractional elapsed time after a logic 1 is received from the input.
curr	F2	Records elapsed time to be compared to final value. Max recordable time is determined by system scaling (XTMRM or XTMRH).

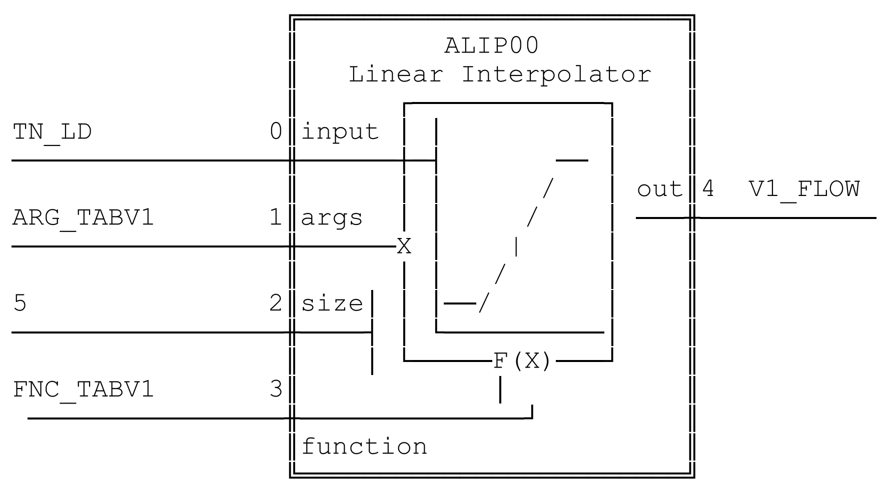
C-6. GENERIC BIG BLOCK PROGRAMMING TOOLS

C-6.1. ALIP00 -- ANALOG LINEAR INTERPOLATION BLOCK

ALIP00	
input	output
	arg_tab
	func_tab
	tab_size

Description

The Analog Linear Interpolation block approximates a continuous relationship between the input and the output as a series of connected linear segments. These segments are defined by the pairs of corresponding values in the argument table and the function table. The argument table values are the values of the input where a new linear segment in the approximation begins. The argument entries must be entered in a numerically increasing order. The function table values are the desired output values at the points defined in the argument table. For input values located between argument table entries, standard linear interpolation is performed between the corresponding function table entries to calculate the output value. If the input value is outside of the range of the argument table, the output is clamped to either the first or last element of the function table. As the number of argument table locations increases, the table size should increase correspondingly (Note: table_size must be positive). As table size increases, resolution of the linear interpolation will increase and execution speed will decrease. An illustration of this function is given in the following diagrams.

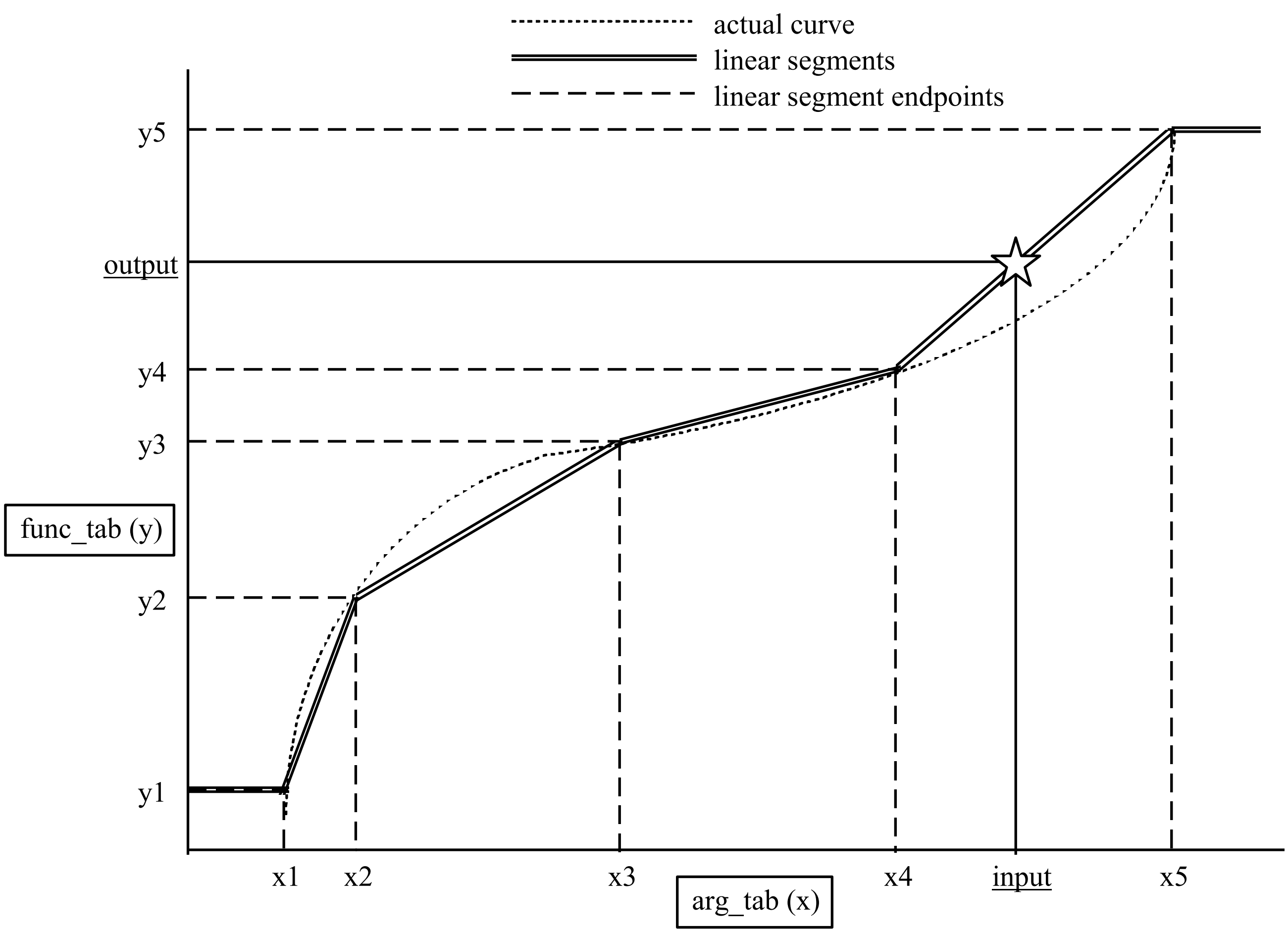


Inputs

Parameter	Data Type	Description:
input	F2	Analog value used to locate the proper linear segment in the argument table, and interpolate between its endpoints.
arg_tab	F2	Index of constant values that define where a new linear segment of the approximation begins. Must be the same size as the function table.
func_tab	F2	Desired output values at points defined in the argument table.
size	A2	Number of entries in the argument table and function table.

Outputs

Parameter	Data Type	Description:
out	F2	The approximation to the output value for the function represented in the Argument and Function Tables.

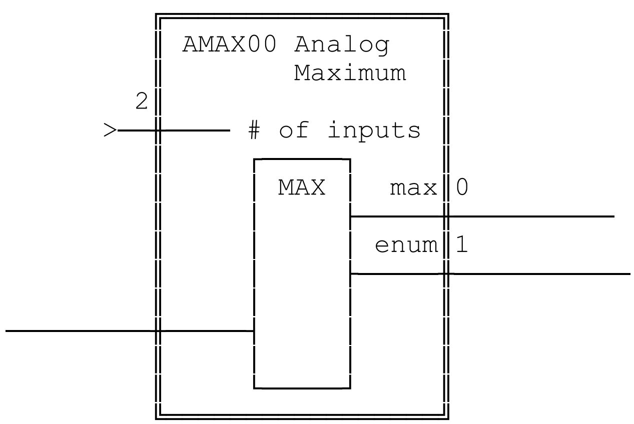


C-6.2. AMAX00 -- ANALOG MAXIMUM SELECT

```
                              AMAX00
input_1 input_10 input_19 input_28 input_37 input_46  max

input_2 input_11 input_20 input_29 input_38 input_47  enum

input_3 input_12 input_21 input_30 input_39 input_48 input_55

input_4 input_13 input_22 Input_31 input_40 input_49 input_56

input_5 input_14 input_23 input_32 input_41 input_50 input_57

input_6 Input_15 input_24 input_33 input_42 input_51 input_58

input_7 input_16 input_25 input_34 input_43 input_51 input_59

input_8 input_17 input_26 input_35 input_44 input_52 input_60

input_9 input_18 input_27 input_36 input_45 input_53
```

Description

The AMAX00 (Analog Maximum Select) block executes a function that interrogates a list of analog inputs (up to 60), selects the maximum value, and passes that value to the output (max). The " enum " output is set equal to the number of the input which was selected as the maximum value.



Inputs

Parameter	Data Type	Description:
input_n(1-60)	F2	Analog inputs. The AMAX00 block can receive up to 60 analog inputs.
# of inputs	K2	Number of entry inputs for AMAX00 algorithm. This block may be used multiple times within a single Control Sequence Program (max. 60 entry inputs per call of the block). This value does not have to be assigned. The value will be assigned by the sequencing compiler based on the number of entry inputs.

Outputs

Parameter	Data Type	Description:
enum	S2	Enumerated State variable that indicates which analog input has been selected as the maximum value.
max	F2	Analog value that represents the maximum analog input received by the block.

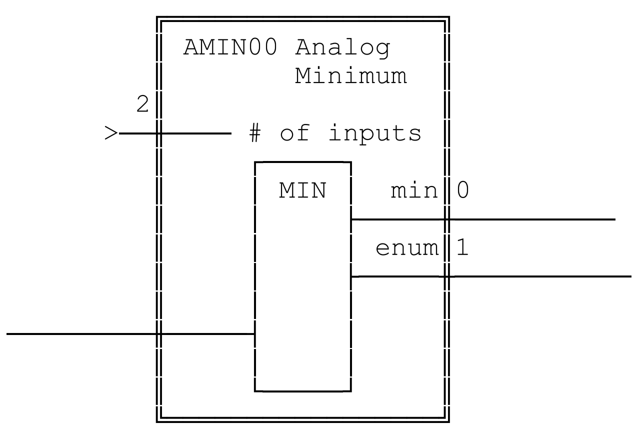
C-6.3. AMINOO-- ANALOG MINIMUM SELECT

AMIN00

input_1	input_10	input_19	input_28	input_37	input_46	min
input_2	input_11	input_20	input_29	input_38	input_47	enum
input_3	input_12	input_21	input_30	input_39	input_48	input_55
input_4	input_13	input_22	Input_31	input_40	input_49	input_56
input_5	input_14	input_23	input_32	input_41	input_50	input_57
input_6	Input_15	input_24	input_33	input_42	input_51	input_58
input_7	input_16	input_25	input_34	input_43	input_51	input_59
input_8	input_17	input_26	input_35	input_44	input_52	input_60
input_9	input_18	input_27	input_36	input_45	input_53	

Description

The AMIN00 (Analog Minimum Select) block executes a function that interrogates a list of analog inputs (up to 60), selects the minimum value, and passes that value to the output (min). The " enum " output is set equal to the number of the input which was selected as the minimum value.



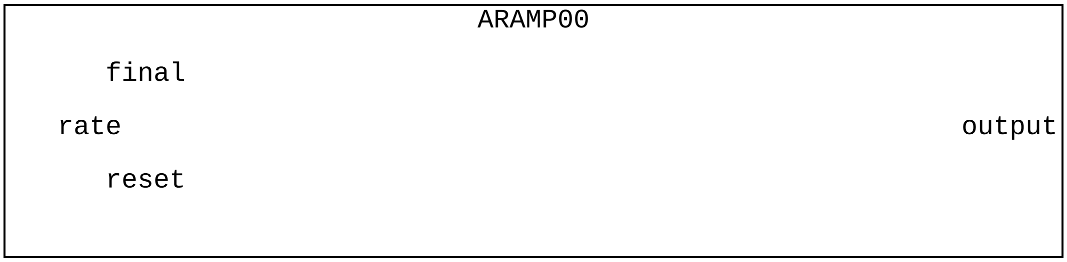
Inputs

Parameter	Data Type	Description:
input_n(1-60)	F2	Analog inputs. The AMIN00 block can receive up to 60 analog inputs.
# of inputs	K2	Number of entry inputs for AMIN00 algorithm. This block may be used multiple times within a single Control Sequence Program (max. 60 inputs per call of the block). This value does not have to be assigned. The value will be assigned by the sequencing compiler based on the number of entry inputs.

Outputs

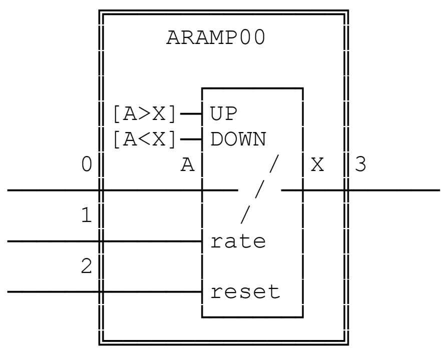
Parameter	Data Type	Description:
enum	S2	Enumerated State variable that indicates which analog input has been selected as the minimum value.
min	F2	Analog value that represents the minimum analog input received by the block.

C-6.4. ARAMP00 -- ANALOG RAMP BLOCK



Description

The Analog Ramp block executes a function that increments a value (X) up (A>X) or down (A<X) until the final (input) value (A) is reached. The output (X) is modified during each scan according to the value supplied to the rate input. The binary scaling of the rate value equals (Binary scale of final - 3 and is expressed in (final) units per sec.: this input must be a positive value. The output is compared to the final input and the rate value is added or subtracted from the output according to the result of the compare. The output will be clamped when it reaches the final value. If the reset logical input reads 1, the output will be set to 0.



Inputs

Parameter	Data Type	Description:
final	F2	Analog value that the output will reflect when the block has executed its function.
rate	F2	Analog constant that represents the rate at which the output is ramped (up or down) to reflect the final input. This value is scaled 3 binary points below the final input scale.
reset	L1	Logical input that sets the Output to 0 when it reads 1.

Outputs

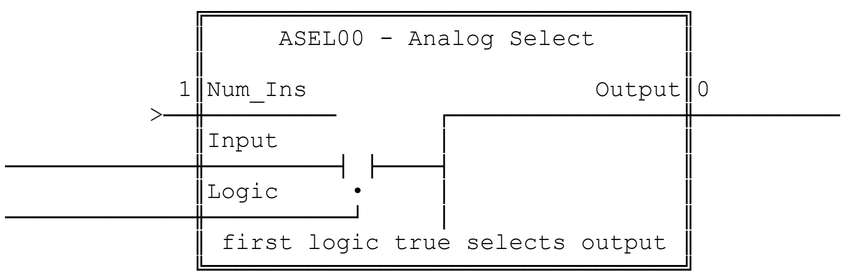
Parameter	Data Type	Description:
output	F2	Analog value that equals the Final input when the block has executed its function.

C-6.5. ASEL00 -- ANALOG SELECT

```
                           ASEL00
                                                  output

input_1  logic_1  input_2  logic_2  input_3  logic_3  input_4
logic_4  input_5  logic_5  input_6  logic_6  input_7  logic_7
input_8  logic_8  input_9  logic_9  input_10 logic_10 input_11
logic_11 input_12 logic_12 input_13 logic_13 input_14 logic_14
input_15 logic_15 input_16 logic_16 input_17 logic_17 input_18
logic_18 input_19 logic_19 input_20 logic_20 input_21 logic_21
input_22 logic_22 input_23 logic_23 input_24 logic_24 input_25
logic_25 input_26 logic_26 input_27 logic_27 input_28 logic_28
```

Description

The ASEL00 block reads a list of logic inputs (max. 28), determines the first true logic, and passes its corresponding analog word variable to the output. The picture below is an abbreviated representation of the block's functionality.



Inputs

Parameter	Data Type	Description:
num_ins	K2	Number of entry sets for ASEL00. This block may be used multiple times within a single Control Sequence Program (max. 28 sets of entries per call of the block). This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry sets.
input_n	V2	Analog word variables that correspond to logic inputs.
logic_n	L1	Logic inputs.

Outputs

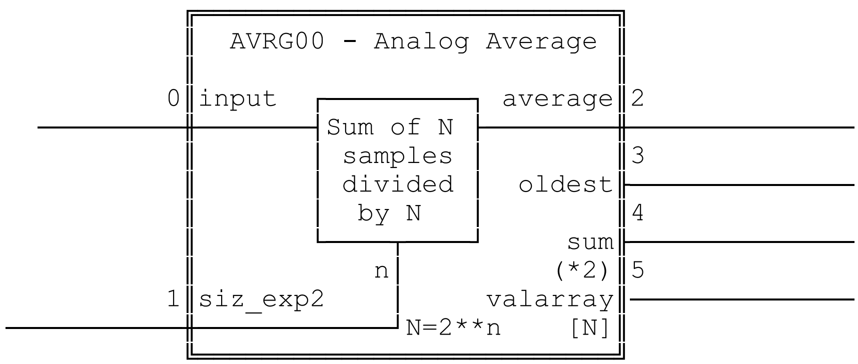
Parameter	Data Type	Description:
output	S2	Analog value that represents the corresponding word variable of the first sequential logic input that reads true.

C-6.6. AVRG00 -- ANALOG AVERAGE BLOCK

```
                              AVRG00
   input         valarray                                  average
   siz_exp2      oldest                                        sum
```

Description

The Average block calculates the rolling average of a sequence of input values. A list of previous values is stored in a value array (valarray []). Once the average is calculated, the value is passed to the output. The first execution of the block initializes the array to the very first input value. Subsequent analog values stored in this location are entered and exited on a first in - first out basis. That is, new values entering the array successively displace the oldest value of the list. To accelerate the computation of the array's rolling average, the oldest value is temporarily stored in oldest. This value is recalled and utilized to calculate the new average. Subsequent calculations will use different oldest values as they are replaced with each update of the array. The value assigned as the length (siz_exp2) must equal the array's actual number of storage cells to generate accurate results from this operation.



Inputs

Parameter	Data Type	Description:
input	F2	Analog value that is used to generate the rolling average.
siz_exp2	C2	The storage capacity of the value array. Actual array size is 2^{siz_exp2}.

Outputs

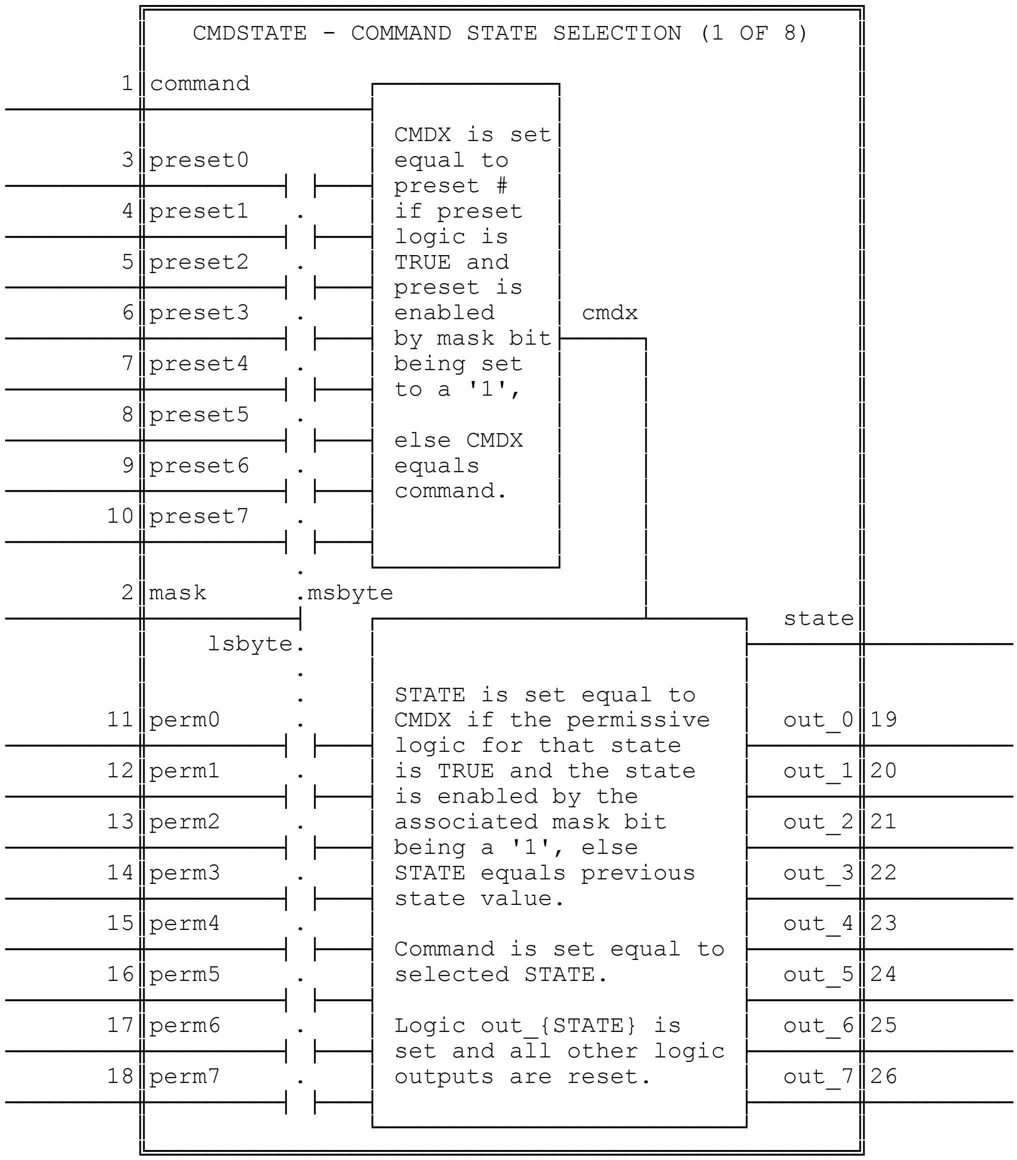
Parameter	Data Type	Description:
average	F2	Analog value equal to the rolling average of the input values over the number of samples given in siz_exp2.
oldest	C2	Temporary storage location that records the oldest value in the value array while the block is running.
sum	F2	Double word (*2) summation of values in value array
valarray	F2	Array of previous values (max. 32 entries, specified by siz_exp2).

C-6.7. CMDSTATE -- COMMAND STATE

```
                              CMDSTATE
           preset0   perm0          state
command    preset1   perm1          out_0
mask       preset2   perm2          out_1
           preset3   perm3          out_2
           preset4   perm4          out_3
           preset5   perm5          out_4
           preset6   perm6          out_5
           preset7   perm7          out_6
                                    out_7
```

Description

The Command State block selects one of eight mutually exclusive operational states and generates data reflecting the selected state. The (0-7) command variable received from the<I>processor must pass the conditions set by a 16 bit mask before it is passed to the State output. The mask value is a control constant that contains the bit values of the mask. The upper or most significant byte of the mask contains enabling data for a consecutive series of 8 preset states (0-7). If a 1 logic input to a preset state is true, and the corresponding bit of the mask value is set, the command variable is changed to that state. The command variable will be changed to the first preset state that meets these criteria. When the preset condition evaluation is completed, the command variable is processed against the second part (least significant byte) of the mask (Note: This comparison will occur whether the command variable value is changed or not). The LSB of the mask contains enabling data for a consecutive series (0-7) of permissive states. If the command variable corresponds to a Permissive state that has a logic true input and an LSB mask bit set, the variable is passed to the state output. This output regulates the 8 logic state outputs of the block and sends status information to the<I>processor. If the State output is set to 0, the first logic output is set to 1 and the other outputs are reset: similarly, higher numbered states will set the associated output and reset the rest of the array.


CMDSTATE - COMMAND STATE SELECTION (1 OF 8)
1 command
3 preset0
4 preset1
5 preset2
6 preset3
7 preset4
8 preset5
9 preset6
10 preset7
CMDX is set equal to preset # if preset logic is TRUE and preset is enabled by mask bit being set to a '1',
else CMDX equals command.
cmdx
2 mask
.msbyte
lsbyte.
state
11 perm0
12 perm1
13 perm2
14 perm3
15 perm4
16 perm5
17 perm6
18 perm7
STATE is set equal to CMDX if the permissive logic for that state is TRUE and the state is enabled by the associated mask bit being a '1', else STATE equals previous state value.
Command is set equal to selected STATE.
Logic out_{STATE} is set and all other logic outputs are reset.
out_0 19
out_1 20
out_2 21
out_3 22
out_4 23
out_5 24
out_6 25
out_7 26

Inputs

Parameter	Data Type	Description:
command	S2	Input variable that is compared to mask requirements before it is passed to the output.
mask	H2	Set of conditions stored in two a two byte configuration (preset, permissive) that the command variable must pass before it is passed to the State output.
preset0 to preset7	L1	A logic 1 input for one of these variables requests that the command variable be changed to a value of 0 to 7, respectively.
perm0 to perm7	L1	A logic 1 input for one of these variables, and a corresponding LSB mask bit that is set will allow the command variable to be passed to the State output if the command variable value is 0 to 7, respectively.

Outputs

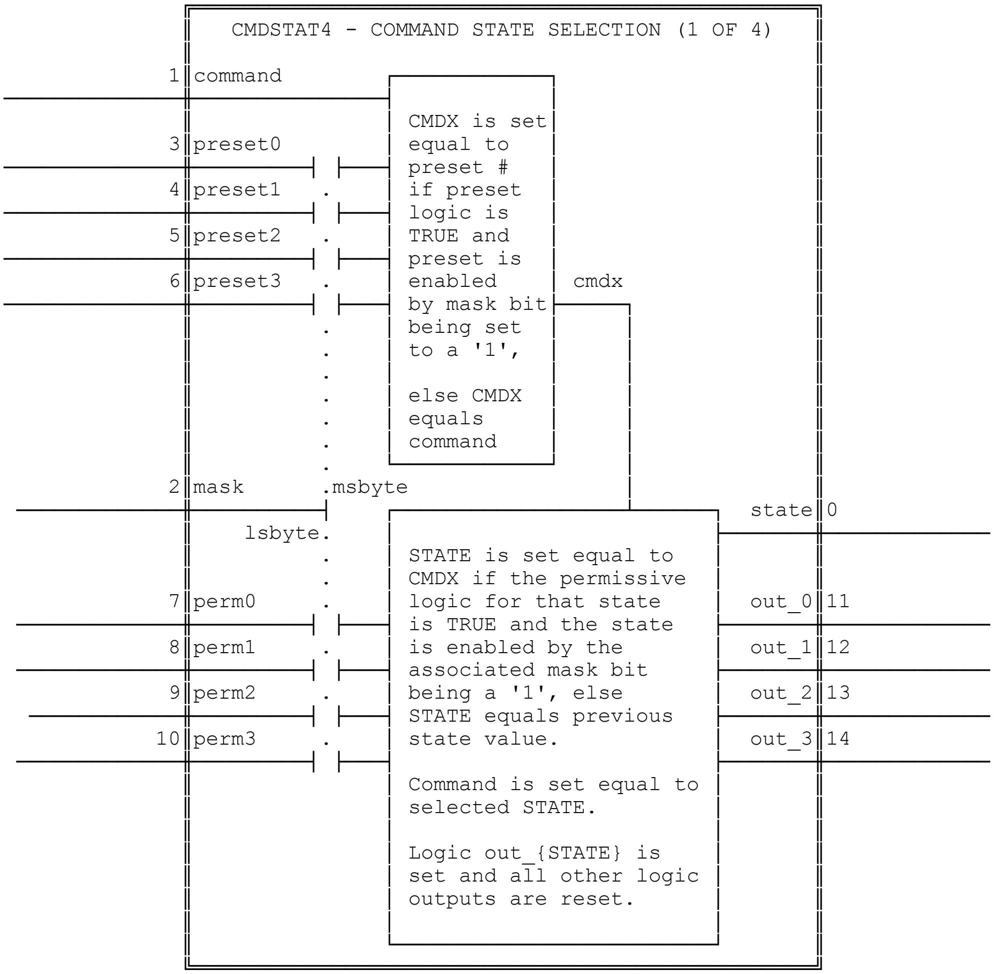
Parameter	Data Type	Description:
state	S2	Output that regulates the 8 logic outputs of the block and sends status information to the <I> processor.
out_0 to out_7	L1	Logic outputs that correspond to the 8 permissive states the block will allow.

C-6.8. CMDSTAT4 -- COMMAND STATE FOUR

```
                        CMDSTAT4
                                                       state
                        preset0     perm0              out_0
command                 preset1     perm1              out_1
mask                    preset2     perm2              Out_2
                        preset3     perm3              out_3
```

Description

The Command State block selects one of four mutually exclusive operational states and generates data reflecting the selected state. The (0-4) command variable received from the <I> processor must pass the conditions set by an eight (8) bit mask before it is passed to the state output. The mask value is a control constant that contains the bit values of the mask. The upper or most significant byte of the mask contains enabling data for a consecutive series of four (4) preset states (0-3). If a logic input to a preset state is true, and the corresponding bit of the mask value is set, the command variable is changed to that state. The command variable will be changed to the first preset state that meets these criterion. When the preset condition evaluation is completed, the command variable is processed against the second part (least significant byte) of the mask (Note: This comparison will occur whether the command variable value is changed or not). The LSB of the mask contains enabling data for a consecutive series (0-3) of permissive states. If the command variable corresponds to a permissive state that has a logic true input and an LSB mask bit set, the variable is passed to the state output. This output regulates the four (4) logic state outputs of the block and sends status information to the<I>processor. If the State output is set to 0, the first logic output is set to 1 and the other outputs are reset: similarly, higher numbered states will set the associated output and reset the rest of the array.



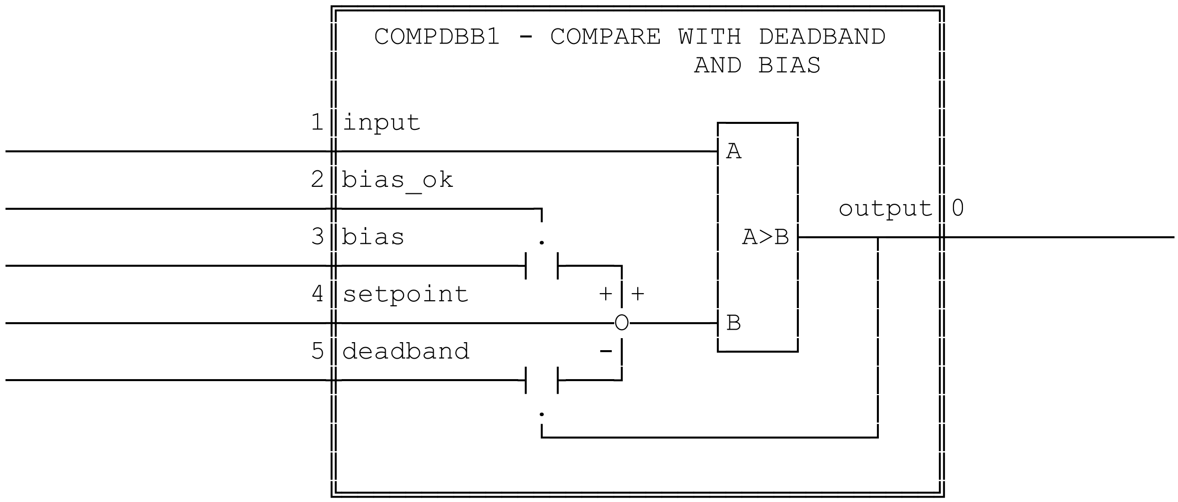
Inputs

Parameter	Data Type	Description:
command	S2	Input variable that is compared to mask requirements before it is passed to the output.
mask	H2	Set of conditions stored in two a two byte configuration (preset, permissive) that the command variable must pass before it is passed to the State output.
preset0 to preset3	L1	A logic 1 input for these variables requests that the command variable be changed to a value of 0 to 3, respectively.
perm0 to perm3	L1	LSB of CMDSTAT4 eight bit mask. A logic " True " permissive in conjunction with a corresponding " True " Preset State will cause the particular Command Variable to be passed to the output

Outputs

Parameter	Data Type	Description:
state	S2	Output that regulates the four (4) logic outputs of the block and sends status information to the <I> processor.
out_0 to out_3	L1	Logic outputs that correspond to the 4 permissive states the block will allow.

C-6.9. COMPDBBI -- COMPARE WITH DEADBAND AND BIAS



Description

The Compare with Deadband and Bias block executes a function that compares an analog variable (input) with a setpoint value; if the input exceeds the setpoint value, the output will be set. A True bias_ok logic input will cause a bias value to be added to the setpoint constant before it is compared against the input variable. When the output is set, a deadband calculation is initiated. The output is reset when the input drops below the difference of the setpoint and deadband values.

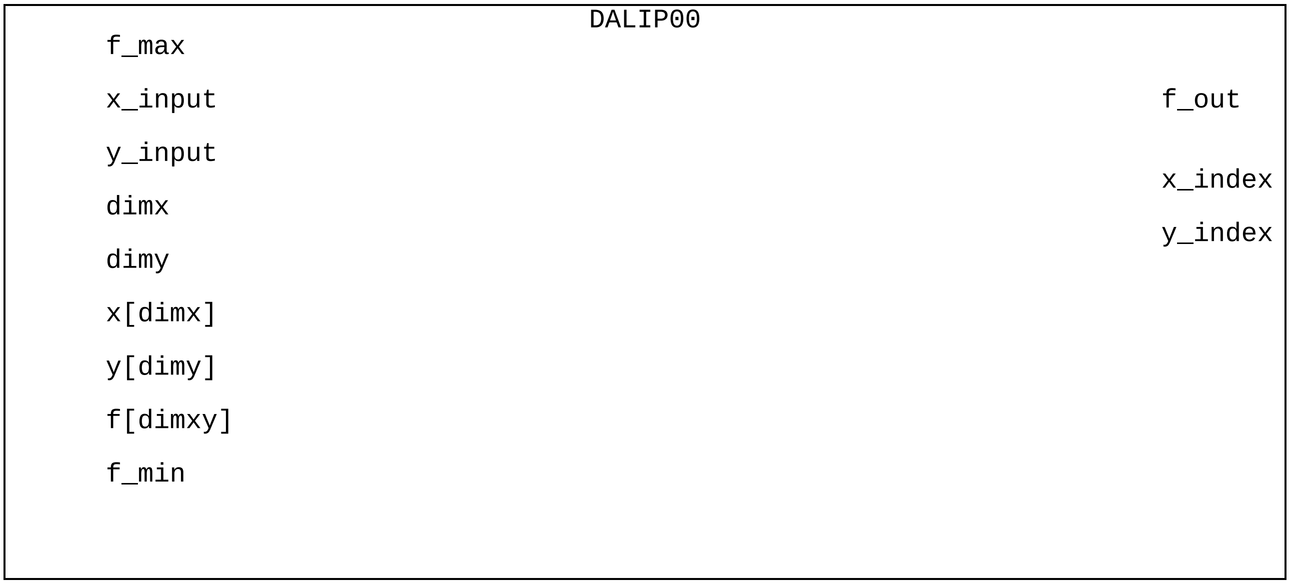
Inputs

Parameter	Data Type	Description:
input	F2	Analog variable that is compared to the setpoint value. If the input exceeds the setpoint value, the output is set to one (1).
bias_ok	L1	Logic input that when True, causes a bias value to be added to the setpoint before it is compared to the input.
bias	F2	Value that is added to the setpoint input when bias_ok is True.
setpoint	F2	Analog constant that is compared to the input.
deadband	F2	Analog constant that is subtracted from the setpoint when the output is set.

Outputs

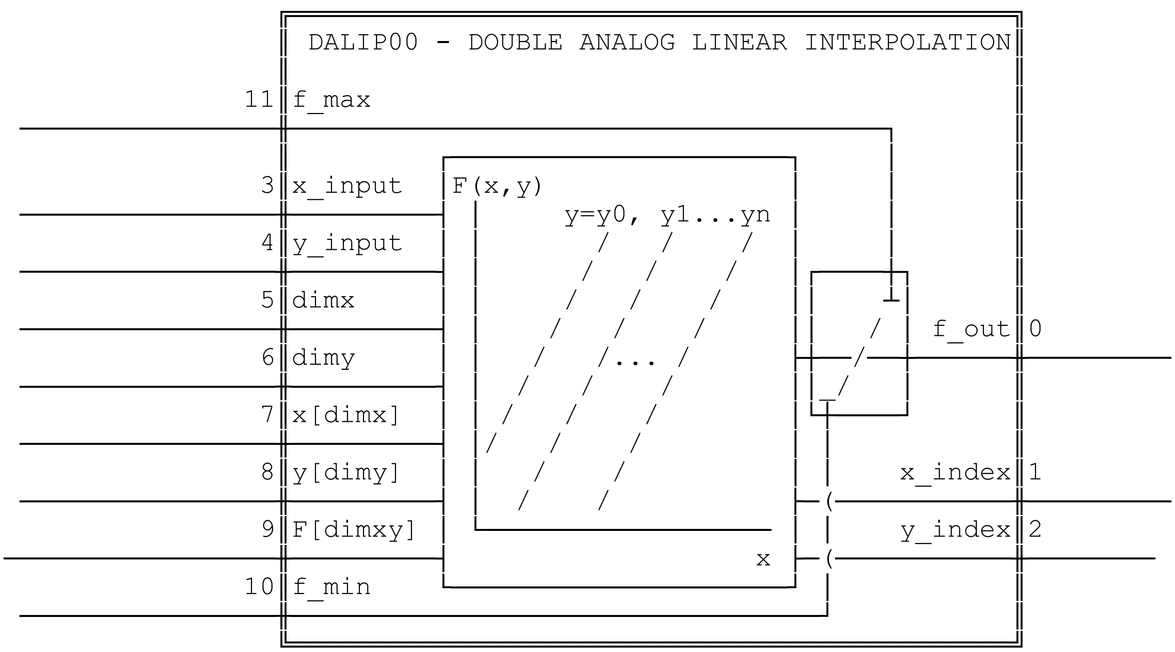
Parameter	Data Type	Description:
output	L1	Logic that reads True when the input value exceeds the setpoint value.

C-6.10. DALIP00 -- DOUBLE ANALOG LINEAR INTERPOLATION



Description

The Double Analog Linear Interpolation block solves for an output as a function of two input variables (i.e., f(x, y)) by performing a dual linear interpolation calculation based on the analog input variables x_input and y_input. The algorithm interpolates linear segments along two axes (x, y) which are in turn defined by x and y table arrays (x[dimx], y[dimy] of size dimx, dimy. [Note: size inputs must be positive; as well, table value inputs must be entered in ascending order). These interpolation calculations locate and define a third value that is passed to the output (f_out). In this manner, the DALIP00 block approximates a continuous (three dimensional) relationship between the x_input and y_input inputs, and the output reference. To perform the interpolation, the block requires at least four known points to be defined; these points are supplied by the f[dimxy] array input. If the x_input and y_input signals exceed the range of the dimx and/or dimy arrays, the output will be clamped to either the first or last value of f[dimxy]. The output reference will be clamped according to f_min or f_max. inputs.



The example below provides a functional example of the double interpolation performed by the DALIP00 block. The module estimates a missing value (F(x,y)) from four (4) known values at neighboring points {F(0,0), F(0,1), F(1,0), F(1,1)}.

```
            |                             |                              |
Y(1)- - -0-----------------------------0------------------------------0
            | F(0,1)                      | F(x,1)                       | F(1,1)
            |                             |                              |
            |                             |                              |
            |                             |                              |
            |                             |                              |
            |                             |                              |
Y    - - -0-----------------------------0 F(x,y)                       0
            |                             |                              |
            |                             |                              |
            |                             |                              |
            |                             |                              |
            |                             |                              |
            |                             |                              |
Y(0)- - -0-----------------------------0------------------------------0
            | F(0,0)                      | F(x,0)                       | F(1,0)
           X(0)                           X                             X(1)
```

First, F(x, 0) is linearly interpolated between F(1, 0) and F(0, 0).

$$F(x,0) = F(0,0) + [\frac{x - X(0)}{X(1) - X(0)}] * [F(1,0) - F(0,0)]$$

Second, F(x, 1) is linearly interpolated between F(1, 1) and F(0, 1).

$$F(x,1) = F(0,1) + [\frac{x - X(0)}{X(1) - X(0)}] * [F(1,1) - F(0,1)]$$

Finally, the linear interpolation between F(x, 0) and F(x, 1) gives F(x, y).

$$F[x,y] = F(x,0) + [\frac{y - Y(0)}{Y(1) - Y(0)}] * [F(x,1) - F(x,0)] =$$

$$F(0,0) + [\frac{x - X(0)}{X(1) - X(0)} * [F(1,0) - F(0,0)]] + [\frac{y - Y(0)}{Y(1) - Y(0)} * [F(0,1) - F(0,0)]]$$

$$[\frac{y - Y(0)}{Y(1) - Y(0)} * \frac{x - X(0)}{X(1) - X(0)} * [F(1,1) - F(0,1) - F(1,0) - F(0,0)]]$$

Inputs

Parameter	Data Type	Description:
f_max	F2	Analog constant that clamps the maximum value of the output.
x_input	F2	Input
y_input	F2	Input
dimx	F2	X array size.
dimy	F2	Y array size
x[dimx]	F2	First signal name (pointer) of the X array
y[dimy]	F2	First signal name (pointer) of the Y array
f[dimxy]	F2	First signal name (pointer) of the F array defining the output corresponding to the fixed points of x and y arrays. The F array is a one dimensional array ordered as: F(0,0), F(0,1), F(0,2),...F(1,0), F(1,1), F(1,2),...
f_ min	F2	Analog constant that clamps the minimum value of the output.

Outputs

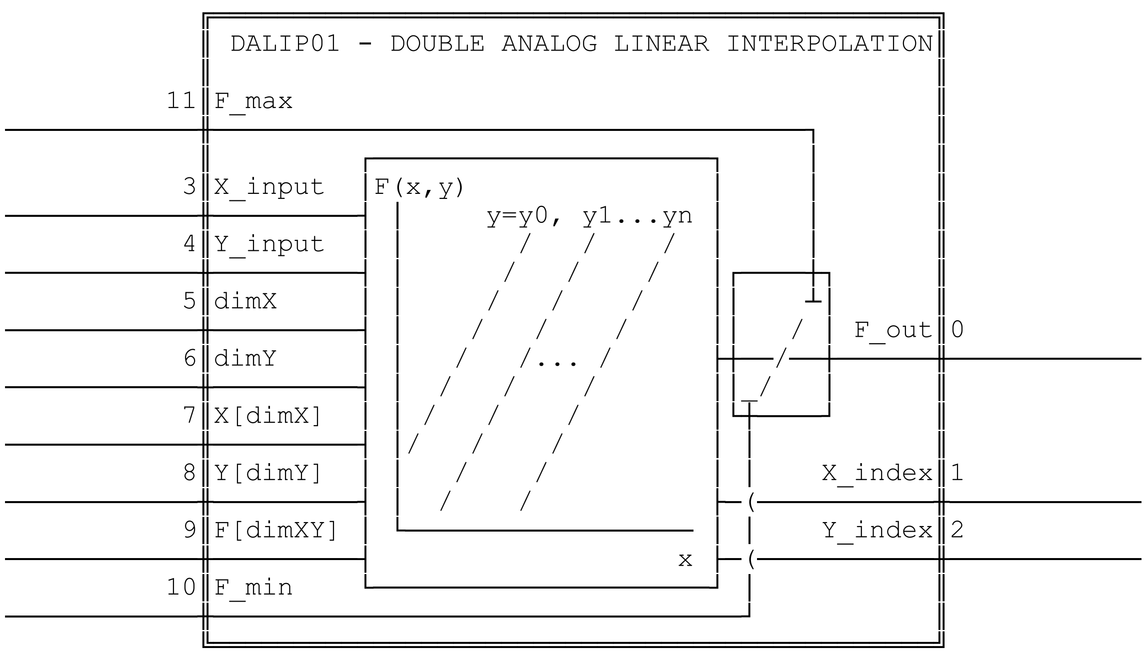
Parameter	Data Type	Description:
f_out	F2	Result of double linear interpolation calculation.
x_index	F2	x array index.
y_index	F2	y array index.

C-6.11. DALIP01 -- DOUBLE ANALOG LINEAR INTERPOLATION

```
                              DALIP01

   x_input                                                        f_out

   y_input
                                                                  x_index
   dimx              x[dimx]
                                                                  y_index
   dimy              y[dimy]

                     f[dimxy]

   f_min             f_max
```

Description

The Double Analog Linear Interpolation block solves for an output as a function of two input variables (i.e., f(x, y)) by performing a dual linear interpolation calculation based on the analog input variables x_input and y_input. The algorithm interpolates linear segments along two axes (x, y) which are in turn defined by x and y table arrays (x[dimx], y[dimy] of size dimx, dimy. [Note: size inputs must be positive; as well, table value inputs must be entered in ascending order). These interpolation calculations locate and define a third value that is passed to the output (f_out). In this manner, the DALIP00 block approximates a continuous (three dimensional) relationship between the x_input and y_input inputs, and the output reference. To perform the interpolation, the block requires at least four known points to be defined; these points are supplied by the f[dimxy] array input. If the x_input and y_input signals exceed the range of the dimx and/or dimy arrays, the output will be clamped to either the first or last value of f[dimxy]. The output reference will be clamped according to f_min or f_max. inputs. Note: The only difference between this BBL and the DALIP00 BBL is the data type for **dimx** and **dimy**. In addition, the example provided for the BBL DALIP00 also applies to this block as well.



Inputs

Parameter	Data Type	Description:
f_max	F2	Analog constant that clamps the maximum value of the output.
x_input	F2	Input
y_input	F2	Input
dimx	A2	X array size.
dimy	A2	Y array size
x[dimx]	F2	First signal name (pointer) of the X array
y[dimy]	F2	First signal name (pointer) of the Y array
f[dimxy]	F2	First signal name (pointer) of the F array defining the output corresponding to the fixed points of x and y arrays. The F array is a one dimensional array ordered as: F(0,0), F(0,1), F(0,2),...F(1,0), F(1,1), F(1,2),...
f_ min	F2	Analog constant that clamps the minimum value of the output.

Outputs

Parameter	Data Type	Description:
f_out	F2	Result of double linear interpolation calculation.
x_index	F2	x array index.
y_index	F2	y array index.

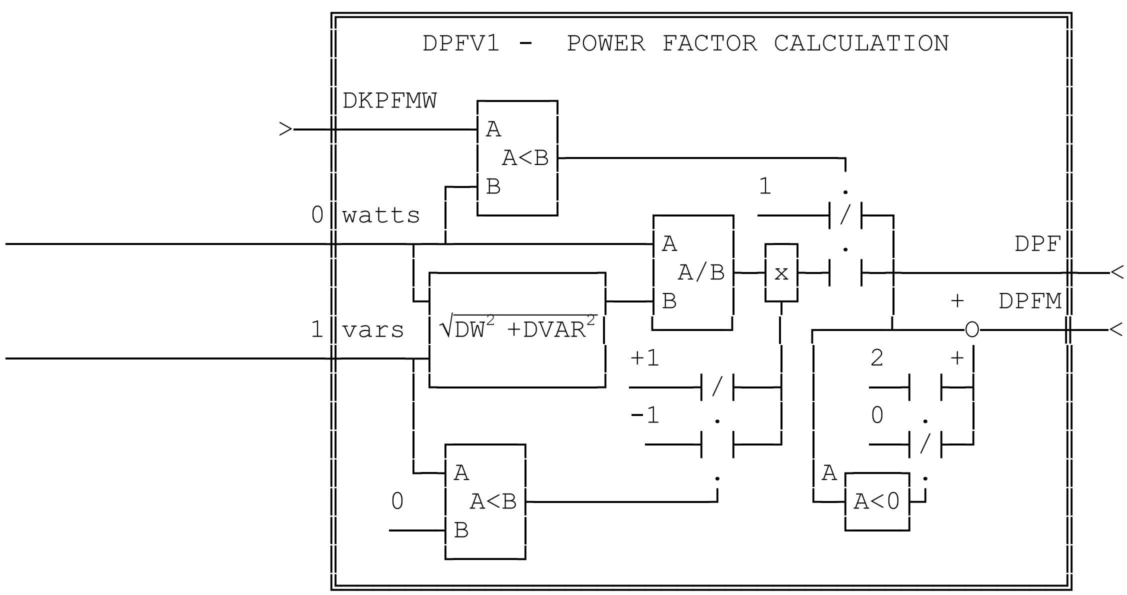
C-6.12. DPFV1 -- POWER FACTOR CALCULATION

```
                              DPFV1

    dwatt
    dvar
```

Description

The Power Factor Calculation block executes a function that reads watts, and reactive volt-amperes (vars) calculated from analog inputs and computes the power factor at the point connection. The DPF analog output of the block will remain at 1 until the watt input exceeds the DKPFMW (constant power factor - megawatt) input value. The sign of the output is determined by the dvar input and designates the direction of reactive power flow. This value is positive for the generator supplying vars to the system. Note: when the power factor calculation is negative (A<0), the DPFM (driven power factor modified) output will pass the value in terms of positive numbers; this is accomplished by adding a positive value of two to the power factor number.

$$Powerfactor = \operatorname{Cos}\theta = \frac{Watts}{\sqrt{Watts^2 + VARS^2}}$$

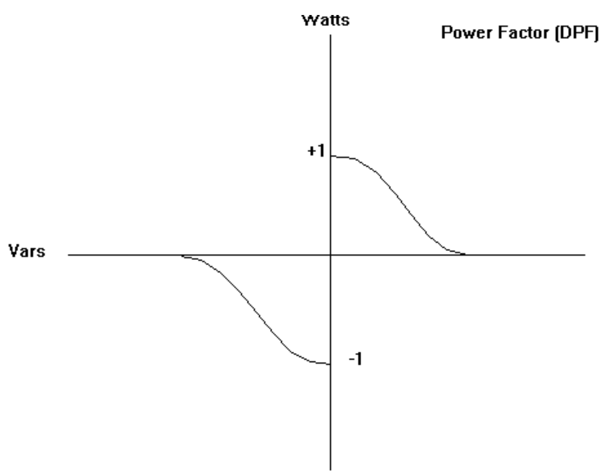


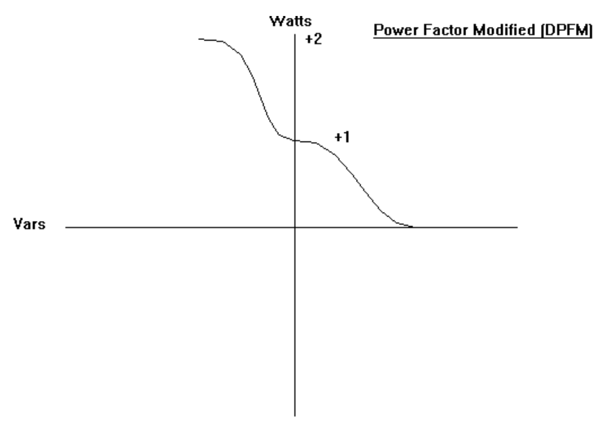
Inputs

Parameter	Data Type	Description:
DKPFMW	F2	Power Factor constant that is compared to the watts input. If the watts input is less than this value, the block will pass a value of one (1) to the output(s).
watts	F2	Analog input that represents the amount of watts being measured.
vars	F2	Analog input that represents the amount of vars (volt-ampere reactive) being measured.

Outputs

Parameter	Data Type	Description:
DPF	F2	Analog value that represents the power factor as calculated from the Watts and Vars analog inputs. The sign of the dpf output is determined by Vars input; an input value greater than 0 will generate a positive output while one that is less than 0 will produce a negative output.
DPFM	F2	Power Factor Modified: Analog value that represents the power factor in terms of positive numbers.





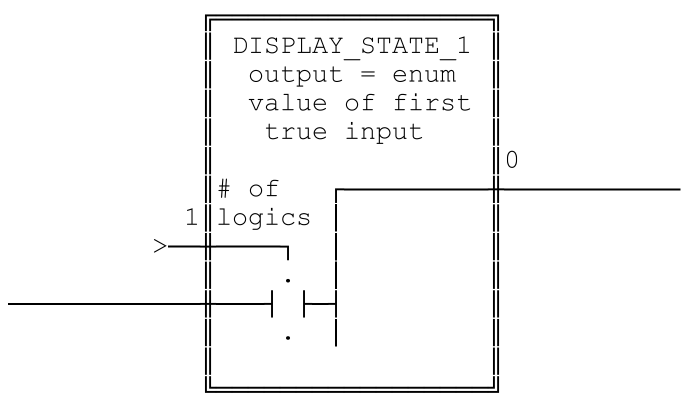
C-6.13. DPYSTAT1 -- DISPLAY STATE 1

DPYSTAT1

						output
input_1	input_2	input_3	input_4	input_5	input_6	input_7
input_8	input_9	input_10	input_11	Input_12	input_13	input_14
input_15	input_16	input_17	input_18	input_19	input_20	input_21
input_22	input_23	input_24	input_25	input_26	input_27	input_28
input_29	input_30	input_31	input_32	input_33	input_34	input_35
input_36	input_37	input_38	input_39	input_40	input_41	input_42
input_43	input_44	input_45	input_46	input_47	input_48	input_49
input_50	input_51	input_52	input_53	input_54	input_55	input_56

Description

The Display State 1 block generates a value for an Enumerated State Variable. This is accomplished by interrogating a list of logic variable inputs (1-56). The first consecutive logic input that reads True will have its corresponding Enumerated State value passed to the output. If none of the logic inputs are True, the enumerated state will read 0. For efficient block operation, only those (consecutive) input logic variables that will be utilized for function definition should be filled in.



Inputs

Parameter	Data Type	Description:
# of logics	K2	Number of entry sets for DPYSTAT1 This block may be used multiple times within a single Control Sequence Program (max. 56 sets of entries per call of the block). This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry inputs.
input_n	L1	The DPYSTAT1 block has 56 possible logic inputs that correspond to 56 Enumerated State Variables. The first consecutive logic True input will determine which ESV. is passed to the output.

Outputs

Parameter	Data Type	Description:
output	S2	Enumerated State Variable whose value is set corresponding to the sequential order number of the first logic True input.

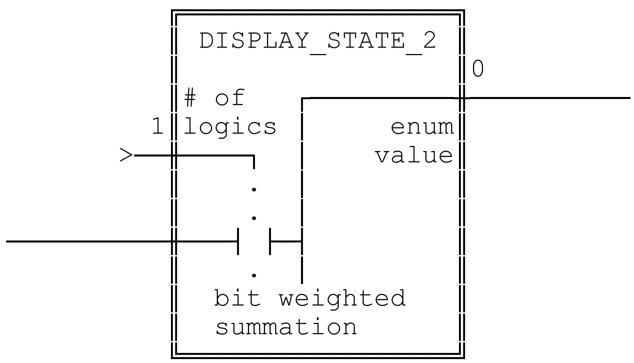
C-6.14. DPYSTAT2 -- DISPLAY STATE 2

DPYSTAT2

input_1	input_2	input_3	input_4	
input_5	input_6	input_7	input_8	output
input_9	input_10	input_11	input_12	
input_13	input_14	input_15	input_16	

Description

The Display State 2 block generates an Enumerated State Variable and passes that value to the output. This is accomplished by interrogating a list of logic variables (1-16) and setting the enumerated state value equal to the bit weighted summation of the true logic entries (that is, the block is able to generate an enumerated state value that can range from 0 to (2^{15})-1). When the logic True inputs are tallied, the equivalent enumerated state value is passed to the output.



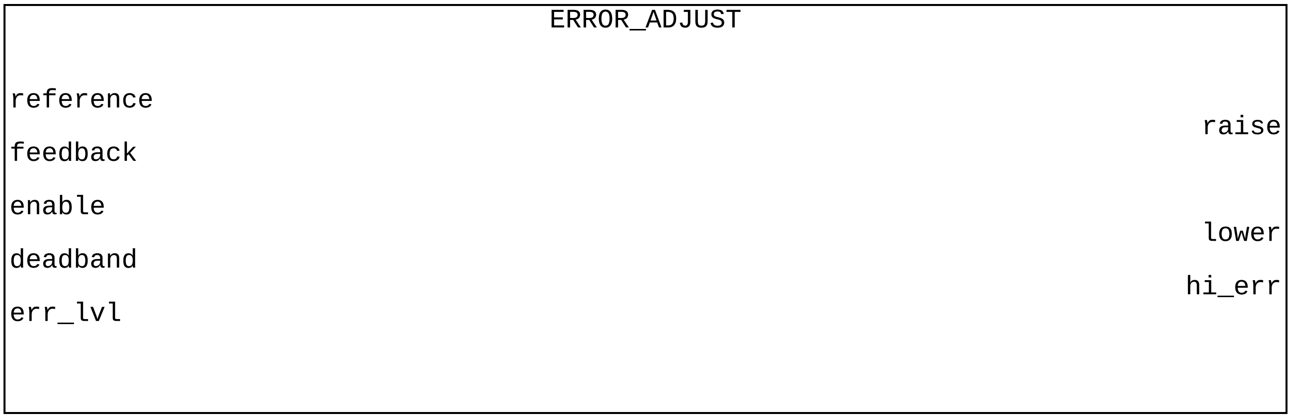
Inputs

Parameter	Data Type	Description:
input_n	L1	Logic inputs. There are 16 possible logic inputs for the Display State block.
# of logics	K2	Number of entry inputs for DPYSTAT2. This block may be used multiple times within a single Control Sequence Program (max. 16 sets of entries per call of the block). This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry inputs.

Outputs

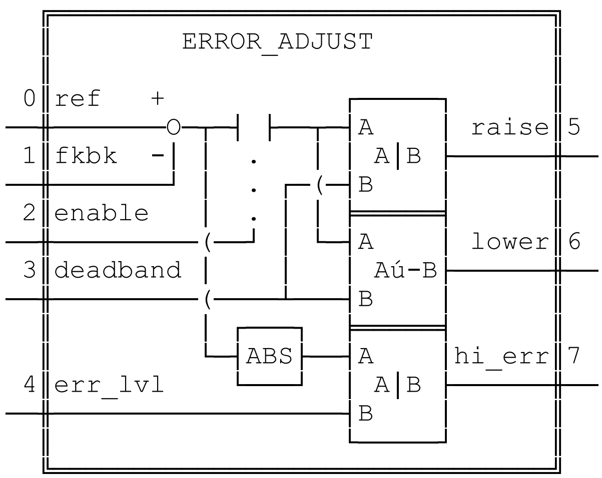
Parameter	Data Type	Description:
output	S2	Enumerated State Variable whose value is equal to the weighted bit summation of the logic True inputs.

C-6.15. ERROR ADJUST



Description

The Error Adjust block is controlled by an enable logical input; a logic 1 signal on this input will permit block execution. The block executes a comparison between a reference value and a feedback value and generates raise and lower logic signals. The value assigned to deadband determines how far the feedback value may deviate from the reference value before a raise or lower signal is generated. When the difference between reference and feedback exceeds the deadband value an output signal is produced. The error level constant defines when a high error signal will be generated. If the absolute difference between reference and feedback exceeds the error level, a logic 1 is passed to the high error output (Note: The value assigned to error level should be greater than the deadband value). For proper block operation, reference, feedback, deadband, and error level values must be scaled identically.



Inputs

Parameter	Data Type	Description:
reference	F2	Analog quantity that represents the desired parameter the block is to maintain with the raise and lower outputs.
feedback	F2	The actual value being controlled with the raise and lower outputs.
enable	L1	The enable logical input determines whether the block will execute. A 1 will permit block execution; a 0 will bypass the block.
deadband	F2	Analog constant that determines how far the feedback value may deviate from the Reference value before a Raise or Lower signal is generated.
err_lvl	F2	Analog constant that determines how far the feedback value may deviate from the reference value before an alarm is generated.

Outputs

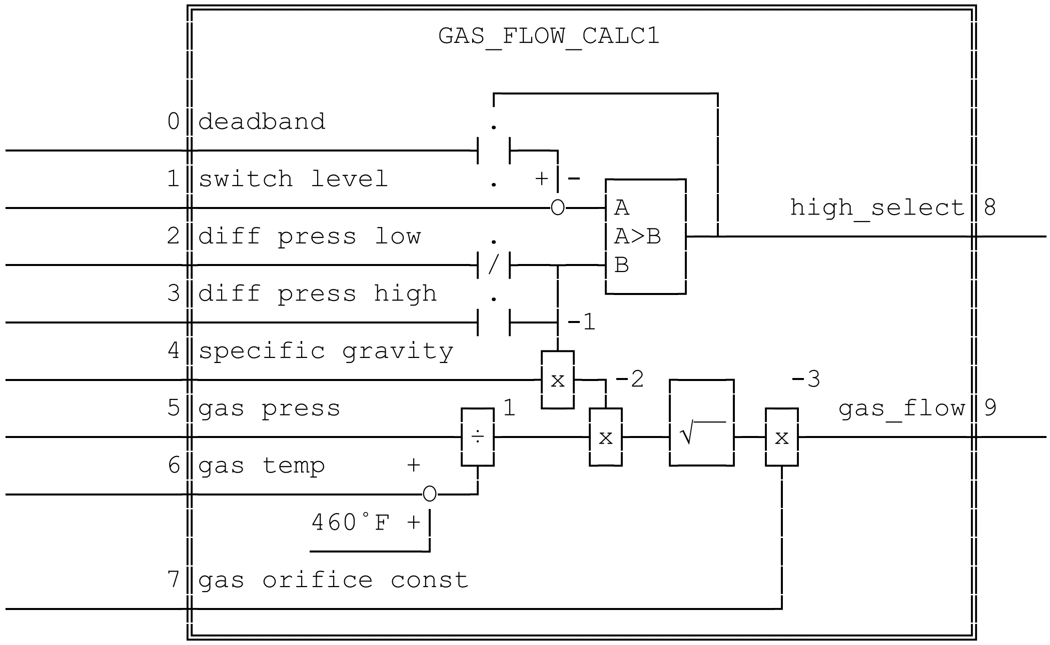
Parameter	Data Type	Description:
lower	L1	Logic signal that is generated when the Feedback value exceeds the reference plus the deadband.
raise	L1	Logic signal that is generated when the feedback value drops below the reference minus the deadband.
hi_err	L1	Logic signal that is generated when the absolute difference between reference and feedback exceeds the error Level.

C-6.16. GAS_FLOW_CALC1

```
                                GAS_FLOW_CALC1

deadband

sw_level
                                                                    1Selhigh
low_dp       high_dp
                                                                    gas_flow
sg_input

g_press

g_temp

o_const
```

Description

The Gas_Flow_Calc1 block executes a function that performs a gas flow calculation based on one of two pressure transducer inputs (diff press low, diff press high). The algorithm performs the gas flow calculation on the value received from the low range differential pressure input (low_dp) until this value exceeds the switch level constant (switch level). When this occurs, the high select logic output (high_select) will become true and the block will perform the gas flow computation on the value received from the high range differential pressure transducer input. The gas flow calculation will continue to be performed based on the high diff. press. input until this value equals or drops below, the switch level minus the deadband value.



Inputs

Parameter	Data Type	Description:
deadband	F2	Analog constant that, when multiplied by the switch level input (sw_level), defines the limit the high range differential pressure input value (high_dp) must drop below before the block begins using the low range differential pressure input.
switch level	F2	Analog constant that determines which transducer input is selected to calculate gas flow.
diff press low	F2	Analog input value received from the low range pressure differential transducer.
diff press high	F2	Analog input value received from the high range pressure differential transducer.
spcific gravity	F2	Analog input that supplies the volume per unit mass value for the gas flow calculation.
gas press	F2	Gas Pressure (absolute) input value used in the gas flow calculation.
gas temp	F2	Gas Temperature input value used in the gas flow calculation.
gas orifice constant	F2	Orifice constant.

Outputs

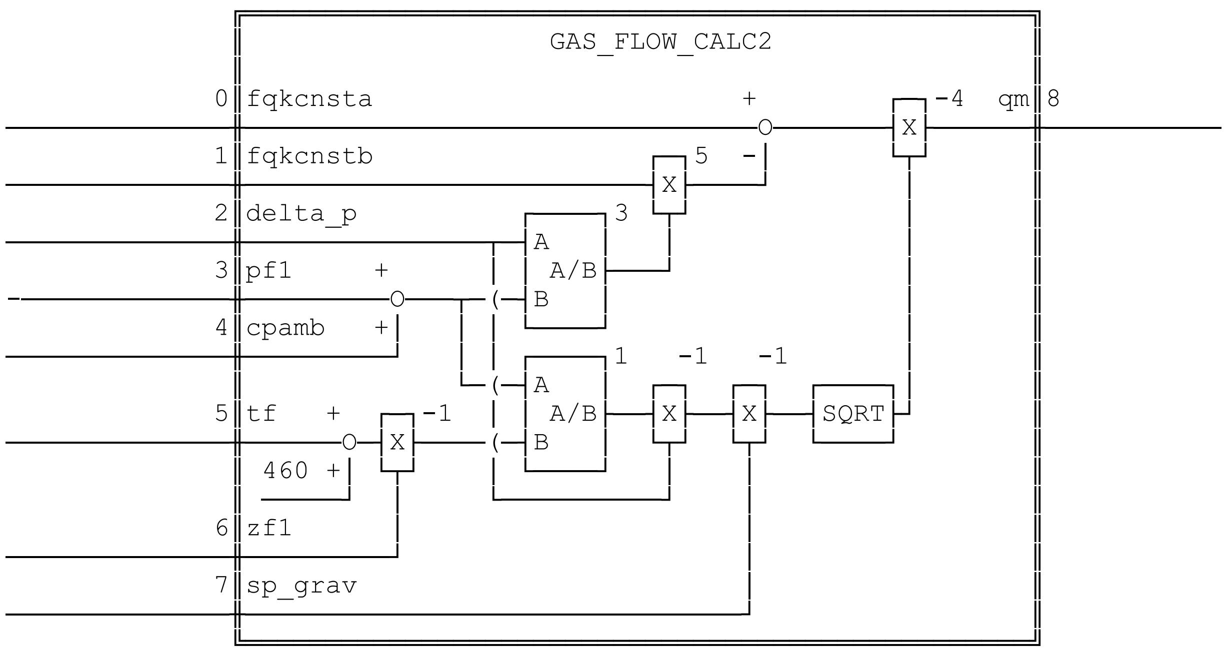
Parameter	Data Type	Description:
high_select	L1	Logic output that reads true (1) when the block selects the high differential pressure transducer input to calculate gas flow. This output will read false (0) when the low pressure differential transducer input is selected.
gas_flow	F2	Analog value that represents the result of the block's gas flow calculation.

C-6.17. GAS_FLOW_CALC2

```
                              GAS_FLOW_CALC2
fqkcnsta                                                            qm
fqkcnstb
delta_p
pf1              cpamb
tf
zf1
sp_grav
```

Description

The Gas_Flow_Calc2 block executes a function that performs a gas flow calculation based on a differential pressure transducer input (delta_p).



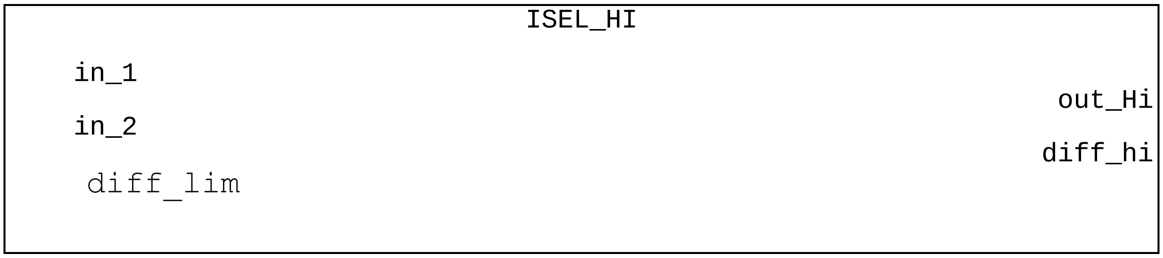
Inputs

Parameter	Data Type	Description:
fqkcnsta	F2	Analog constant.
fqkcnstb	F2	Analog constant.
delta_p	F2	Analog input value received from the differential pressure transducer.
pf1	F2	Analog input value received from the gauge pressure transducer.
cpamb	F2	Analog input that supplies the ambient pressure.
tf	F2	Gas temperature input value used in the gas flow calculation.
zf1	F2	Gas compressibilty factor input value used in the gas flow calculation.
sp_grav	F2	Gas specific gravity.

Outputs

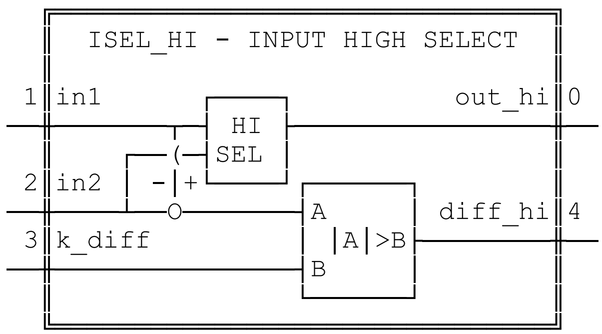
Parameter	Data Type	Description:
qm	F2	Analog value that represents the result of the block's gas flow calculation.

C-6.18. ISEL_HI -- INPUT HIGH SELECT



Description

The ISEL_HI High Select block executes a function that reads two inputs (in1, in2), selects the higher of the two, and passes that value as the output. If the difference between these values exceeds the limit specified by the k_diff input constant, a diff_hi logic output is generated. Note: in1, in2, and k_diff input values should be scaled identically.



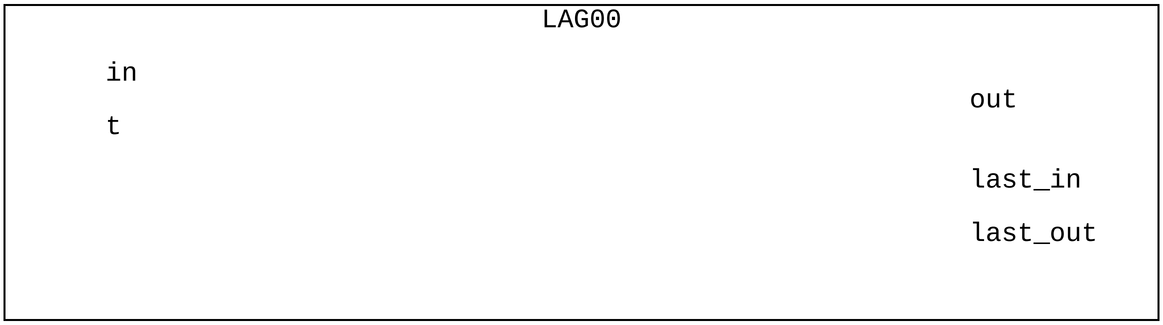
Inputs

Parameter	Data Type	Description:
in1	F2	Analog value compared to in_2.
in2	F2	Analog value compared to in_1.
k_diff	F2	Analog limit constant that defines the range of variance that may occur between in1 and in2 before a diff_hi logic output is generated.

Outputs

Parameter	Data Type	Description:
out_hi	F2	Analog value that represents the higher value between in1 and in2.
diff_hi	L1	Logical output that reads 1 when the difference between in1 and in2 exceeds the k_diff constant.

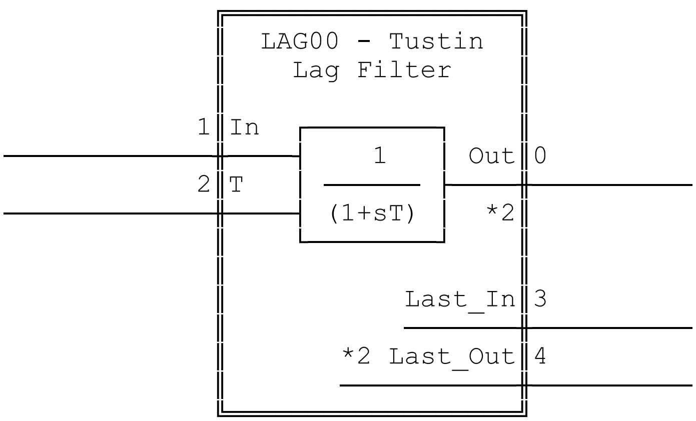
C-6.19. LAG00 -- LAG



Description

The LAG00 block executes a (discrete) filter function that approximates a first order lag. The value of the output approaches the value of the input at a rate determined by the Time Constant (input and output data must be scaled identically). The difference between the input and output values represents 100% of the change of the output. The time constant represents the time needed to raise (or lower) the output 63% of the difference between the input and output. Therefore, it takes 5 time constants to change the output approx. 100% of the difference (1 sec.= 63% of 100%, 2nd sec. = 63% of 37%, 3rd sec. = 63% of 13.7%, 4th sec. 63% of 5%, 5th sec. = 63% of 2%). (See example below.)

Damping characteristics of the block will improve for varying analog inputs (in) as the time constant value is increased. The time constant value should be at least four times the sequencing scan interval.



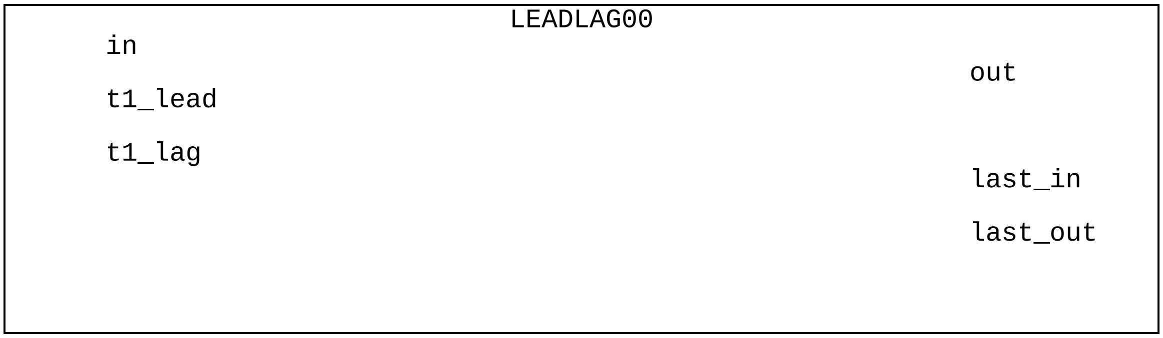
Inputs

Parameter	Data Type	Description:
in	F2	Analog variable is used to calculate the current value of the output.
t	F2	Analog variable that represents the time needed to raise/lower the outut 63% of the current difference between the input and output.

Outputs

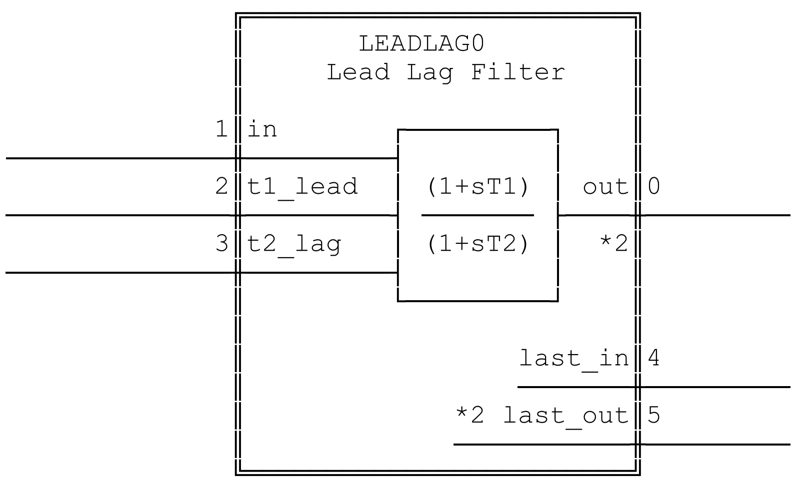
Parameter	Data Type	Description:
out	F2	Analog double-word (*2) variable that is incremented toward the input (in) at a rate determined by the block's time constant (t).
last_in	F2	Single-word variable that stores the most recent input (in).
last_out	F2	Double-word variable (*2) that stores the last outupt value.

C-6.20. LEADLAG00 -- LEAD LAG



Description

The LEADLAG00 block executes a (discrete) filter function that combines both Lead and Lag compensator characteristics. The properties of the Lead compensator are typified by an output that, with an appropriate time constant (T1), is proportional to the sum of the input signal (in) and its derivative (slope). As the derivative action tends to uncover that part of a signal which is not constant (i.e., time varying.), the Lead compensator, with the appropriate time constant, tends to act similar to a high pass filter (magnitude response) except low frequencies are passed with unity gain, while high frequency components of the input signal are amplified. The resultant phase shift for this type of compensator is typically positive—where the output leads the input. The Lag portion of the algorithm acts as integrator, ramping the output toward the input at a rate defined by the time constant, T2. Phase response for this compensator is typically characterized by a lagging phase shift, where the output lags the input. The manner in which the LEADLAG filter compensates the input signal's phase and magnitude response is contingent on the values supplied for the two time constants (T1, T2). Proper block operation requires a T1 value that is two (minimum) to eight (maximum) times the T2 value; as such, the algorithm will function predominantly as a lead compensator, passing lower frequency components of the input with variable gain, and higher frequency components with amplification equal to the ratio T1/T2. In addition, the T2 constant should be assigned a value that is at least four (4) times the sequencing scan interval.



Inputs

Parameter	Data Type	Description:
in	F2	Analog input variable.
t1_lead	F2	Analog variable that defines the block's lead compensator charateristics.
t2_lag	F2	Analog variable that defines the block's lag compensator characteristics.

Outputs

Parameter	Data Type	Description:
out	F2	Analog double-word (*2) variable that is the compensated version of the input.
last_in	F2	Single-word variable that stores the most recent input (in).
last_out	F2	Double-word variable (*2) that stores the last output value.

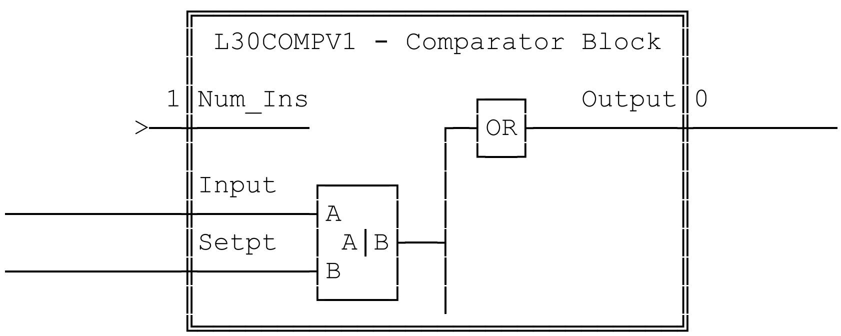
C-6.21. L30COMPV1 -- COMPARATOR BLOCK SIGNAL LEVEL COMPARISON

L30COMPV1

						output
input_1	input_2	input_3	input_4	input_5	input_6	input_7
setpt_1	setpt_2	setpt_3	setpt_4	setpt_5	setpt_6	setpt_7
input_8	input_9	input_10	input_11	input_12	input_13	input_14
setpt_8	setpt_9	setpt_10	setpt_11	setpt_12	setpt_13	setpt_14
input_15	input_16	input_17	input_18	Input_19	input_20	input_21
setpt_15	setpt_16	Setpt_17	setpt_18	setpt_19	setpt_20	setpt_21
input_22	input_23	input_24	input_25	input_26	input_27	input_28
setpt_22	setpt_23	setpt_24	setpt_25	setpt_26	setpt_27	setpt_28

Description

The Comparator (1) block executes a comparison function that compares an analog input signal to a corresponding analog setpoint value (Max. 28 pairs). If any of the analog input signals become equal to or exceed a corresponding setpoint value, the logic output is set to 1. Otherwise, the output is set to 0.



Inputs

Parameter	Data Type	Description:
input_n	F2	Analog input that is compared to a corresponding setpoint value.
setpt_n	F2	Analog constant to which a corresponding input is compared.
Num_Ins	K2	Number of entry sets for L30COMPV1 algorithm. This block may be implemented multiple times within a single Control Sequence Program (max. 28 sets of entries per call of the block.) This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry sets.

Outputs

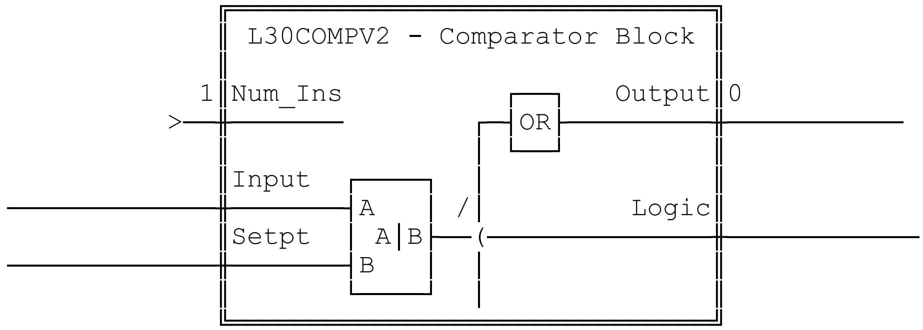
Parameter	Data Type	Description:
output	L1	Logic output that is set to 1 when any of the analog inputs become equal to or exceed a corresponding setpoint value.

C-6.22. L30COMPV2 -- COMPARATOR BLOCK SIGNAL LEVEL COMPARISON

```
                              L30COMPV2
                                                         Output
input_1  input_2  input_3  input_4  input_5  input_6  input_7
setpt_1  setpt_2  setpt_3  setpt_4  setpt_5  setpt_6  setpt_7
logic_1  logic_2  logic_3  logic_4  logic_5  logic_6  logic_7
input_8  input_9  input_10 input_11 input_12 input_13 input_14
setpt_8  setpt_9  setpt_10 setpt_11 setpt_12 setpt_13 setpt_14
logic_8  logic_9  logic_10 logic_11 logic_12 logic_13 logic_14
input_15 logic_15 input_16 input_17 logic_17 input_18
setpt_15 logic_16 setpt_16 setpt_17 logic_18 setpt_18
```

Description

The Comparator (2) block executes a comparison function that compares an analog input to a corresponding setpoint value (Max. 18 pairs). When an analog input equals or exceeds a corresponding setpoint value, a logic output relating uniquely to that comparison is set to 1. If any comparison generates a discriminate logic 1 signal, the block's logic Output is also set to 1. Otherwise, the logic output is set to 0.



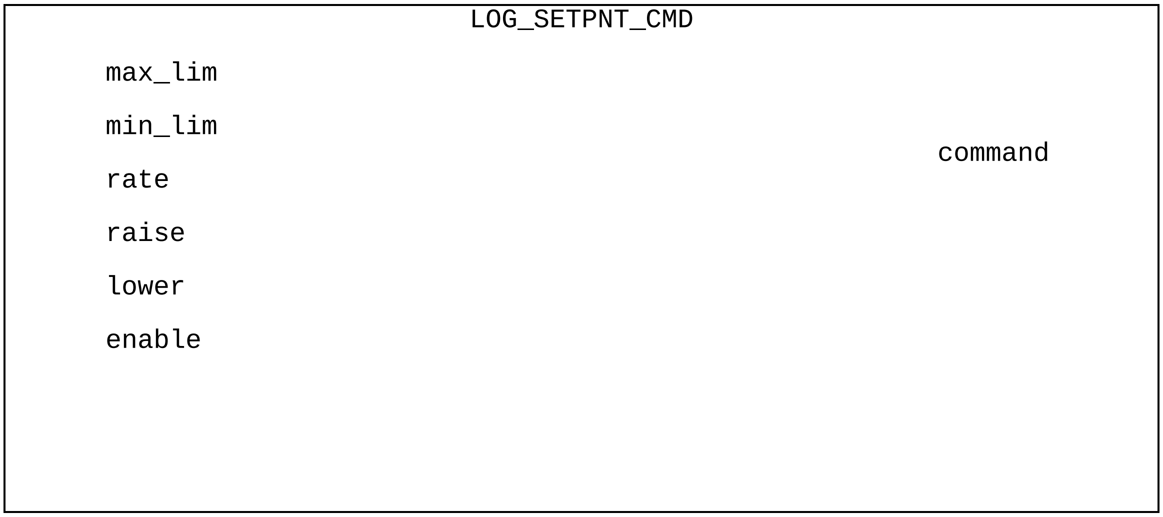
Inputs

Parameter	Data Type	Description:
input_n	F2	Analog input that is compared to a corresponding setpoint value.
setpt_n	F2	Analog value to which a corresponding analog input is compared.
Num_Ins	K2	Number of entry sets for L30COMPV2 algorithm. This block may be used multiple times within a single Control Sequence Program (max. 18 sets of entries per call of the block.) This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry sets.

Outputs

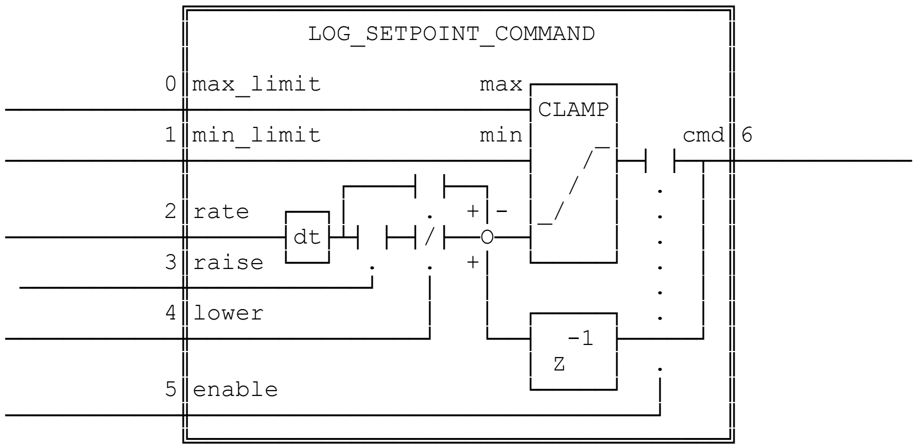
Parameter	Data Type	Description:
logic_n	L1	Logic output that specifically corresponds to an analog comparison: will be set to 1 when an analog input equals or exceeds a corresponding setpoint value.
output	L1	Logic output that is set to 1 when any of the inputs become equal to or exceed a corresponding setpoint value.

C-6.23. LOG_SETPNT_CMD -- LOGIC SETPOINT COMMAND



Description

Execution of the Logic Setpoint Command block is controlled by the enable input; a logic one (1) signal on this input will permit block execution. Using logic signals, the Logic Setpoint Command block ramps setpoint commands up and down. The analog inputs max_lim and min_lim define the range of the command value output (these inputs and the command output value must be scaled similarly). The analog rate input regulates how rapidly the command value is altered within that range. The binary scaling of the rate input must correspond, in terms of maximum expressible values, to 1/8th the binary scaling of the Command output. For example, a command output scaled at 256 (psi) will have a rate input scaled at 32 (psi/sec). Note: Rate-of-change values may be expressed in either units-per-second or units-per-minute. The raise and lower logic inputs determine the positive or negative direction of change. The command output value will continue to increase or decrease until a limit is reached or the raise/lower logic input reading 1 transitions to 0.



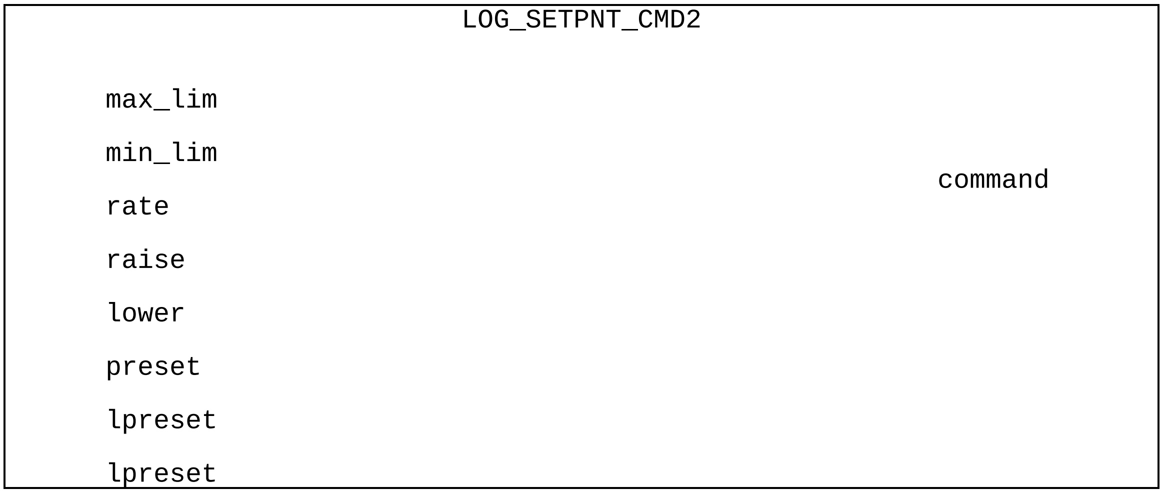
Inputs

Parameter	Data Type	Description:
max_lim	F2	Analog input that determines the highest value that can be passed to command output.
min_lim	F2	Analog input that determines the lowest value that can be passed to the command output.
rate	F2	Analog input that determines how quickly the command output transitions to a new value. Rate scale, in terms of maximum expressible values, should be 1/8 scale of the command output.
raise	L1	The command output value will increase at the rate value as long as this input reads logic 1, or, until the maximum limit is reached.
lower	L1	The command output value will decrease at the rate value as long as this input reads logic 1, or, until the minimum limit is reached.
enable	L1	The enable logical input determines whether the block will execute. A 1 will permit block execution; a 0 will bypass the block.

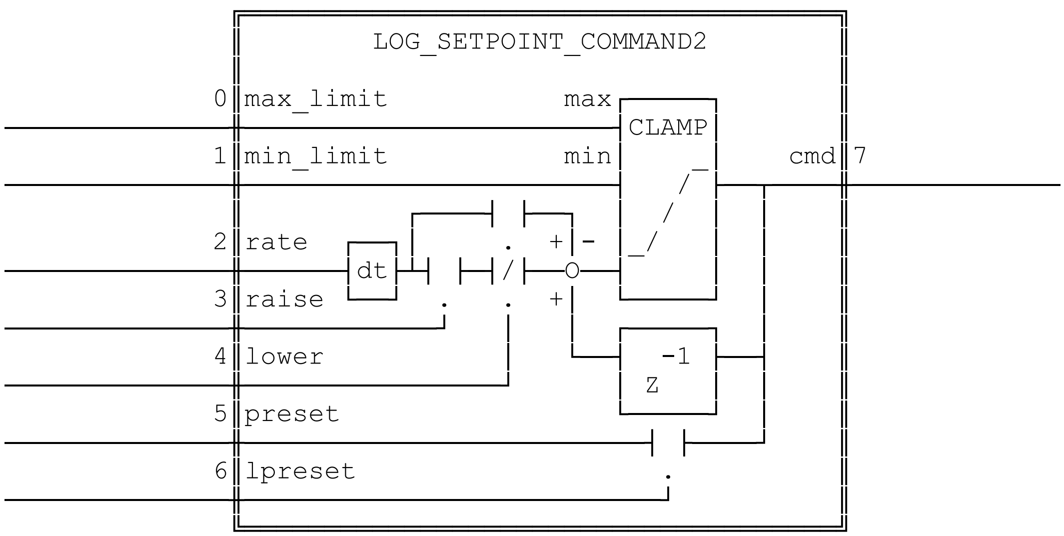
Outputs

Parameter	Data Type	Description:
command	F2	Analog output value that is modified according to the raise, lower, rate, min_lim, and max_lim inputs.

C-6.24. LOG_SETPOINT_COMMAND2



Using logic signals, the Logical Setpoint Command (2) block ramps setpoint commands up and down. The analog inputs max_lim and min_lim define the range of the command value output (these inputs and the command output value must be scaled similarly). The analog rate input regulates how rapidly the command value is altered within that range. The binary scaling of the rate input must correspond, in terms of maximum expressible values, to 1/8th the binary scaling of the command output. For example, a command output scaled at 256 (psi) will have a rate input scaled at 32 (psi/sec). Note: Rate-of-change values may be expressed in either units-per-second or units-per-minute. The raise and lower logic inputs determine the positive or negative direction of change. The command output value will continue to increase or decrease until a limit is reached or the raise/lower logic input reading 1 transitions to 0. If the lpreset input reads one (1), the block will not execute the ramping function, but will pass the Preset value to the output.



Inputs

Parameter	Data Type	Description:
max_lim	F2	Analog input that determines the highest value that can be passed to command output.
min_lim	F2	Analog input that determines the lowest value that can be passed to the command output.
rate	F2	Analog input that determines how quickly the command output transitions to a new value. rate scale, in terms of maximum expressible values, should be 1/8 scale of the command output.
raise	L1	The command output value will increase at the rate value as long as this input reads logic 1, or, until the maximum limit is reached.
lower	L1	The command output value will decrease at the rate value as long as this input reads logic 1, or, until the minimum limit is reached.
preset	F2	Analog value that is passed to the output when the lpreset input reads one (1).
lpreset	L1	A logic one (1) from the lpreset input will cause the the preset value to be passed to the output.

Outputs

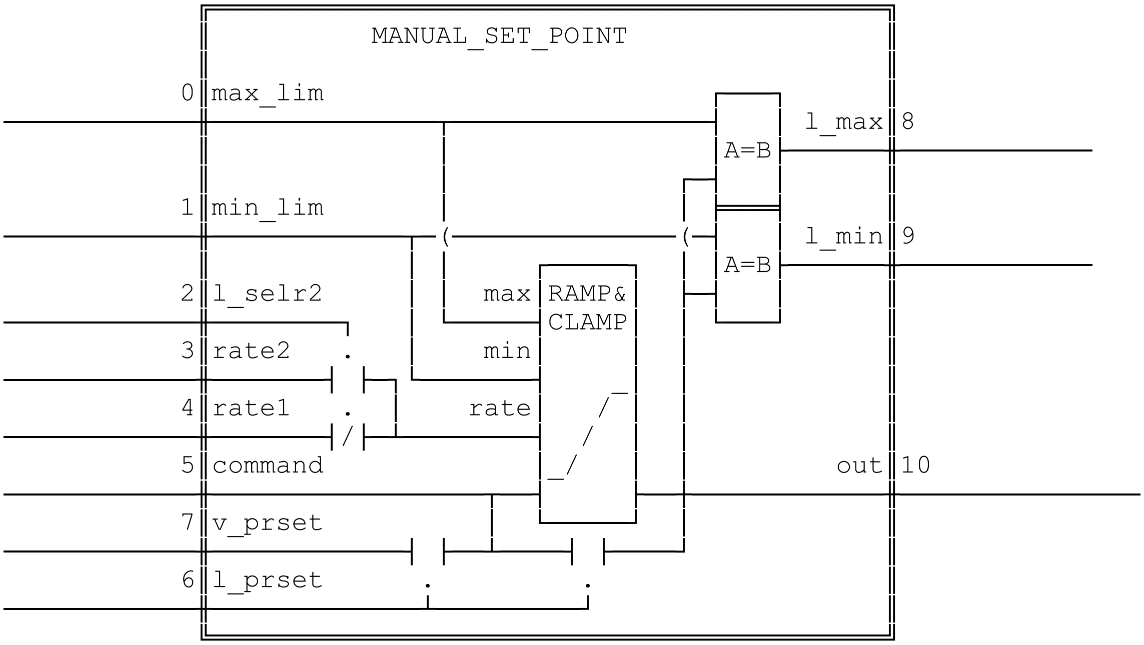
Parameter	Data Type	Description:
command	F2	Analog output value that is modified according to the raise, lower, rate, min_lim, and max_lim inputs.

C-6.25. MANUAL_SET_POINT

```
                         MANUAL_SET_POINT
   max_lim
                                                         l_max
   min_lim
                                                         l_min
   1_selr2

   rate2

   rate1
                                                           out
   command

   l_prset

   v_prset
```

Description

The Manual Setpoint block ramps the analog output up or down until it equals the analog command input value. The analog inputs max_lim and min_lim define the range of adjustment for the output. The rate at which the adjustment is implemented is determined by the analog inputs rate1 and rate2. The block will alter the output value according to the analog rate1 input until the 1_selr2 logic input reads true. At this point, the value is adjusted according to the rate2 schedule (Note: Scaling of the rate1 and rate2 analog inputs, in terms of maximum expressible values, should be 1/8th the binary scaling of the command input and the output -- for example, a command output of 256psi. will have a rate input of 32psi/sec.). When the output reaches a value equal to the maximum or minimum range limits, the logic outputs l_max or l_min are set to 1 respectively. These outputs will remain set to zero (0) until a limit is reached. If the block receives a logic 1 l_preset signal, the command input and the output are immediately adjusted to the v_preset analog value (subject to limits max_lim and min_lim). For proper block operation, the max_lim, min_lim, command, and v_preset inputs, and the output must be scaled identically.



Inputs

Parameter	Data Type	Description:
max_lim	F2	Analog input that determines the highest value the output can be adjusted to.
min_lim	F2	Analog input that determines the lowest value the output can be adjusted to.
rate1	F2	Analog input that determines how quickly the output is raised or lowered to command input value. The rate1 value is 1/8 binary scale of the output.
l_selr2	L1	A logic 1 on this input will instruct the block to adjust the output according to rate2 schedule.
rate2	F2	Analog input that determines how quickly the output is raised or lowered to the command input value when the l_selr2 rate input reads 1. The rate2 input is 1/8 binary scale of the output.
command	F2	Analog input that the output will be adjusted to. If this value exceeds the maximum or minimum range limits of the block, the output will be set to the limit that is applicable.
l_prset	L1	Logic input that enables the v_prset value to be passed immediately to the output subject to the block's min. and max. limits.
v_prset	F2	Analog value that is passed directly to the output when the l_prset logic input is a 1. (This value is subject to the algorithm's min. and max. limits.

Outputs

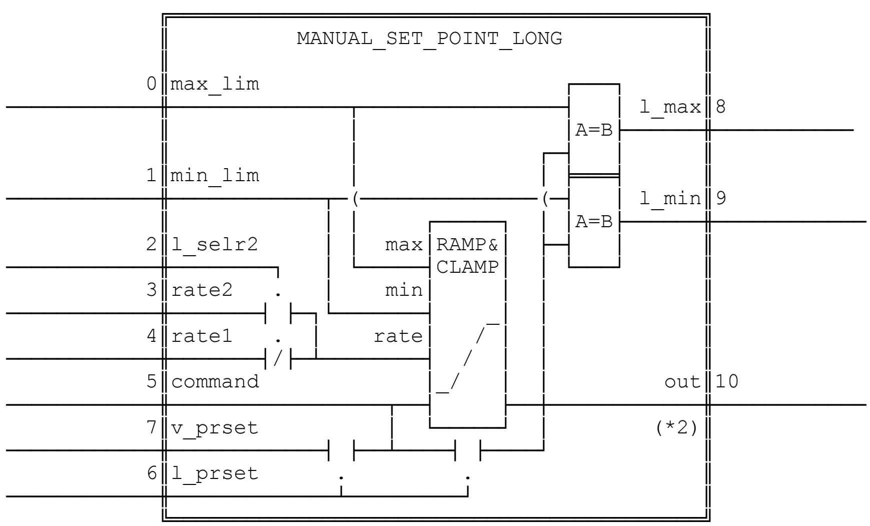
Parameter	Data Type	Description:
l_max	L1	Logical output that is set to 1 when the output value equals the maximum limit value.
l_min	L1	Logical output that is set to 1 when the output value equals the minimum limit value.
output	F2	Analog value that is adjusted to reflect the current command input value.

C-6.26. MANUAL_SET_POINT_LONG

```
                    MANUAL_SET_POINT_LONG
max_lim
min_lim                                              l_max
l_selr2                                              l_min
rate2
rate1
command                                              out
l_prset
v_prset
```

Description

The Manual Setpoint block ramps the analog output up or down until it equals the analog command input value. The analog inputs max_Lim and min_Lim define the range of adjustment for the output. The rate at which the adjustment is implemented is determined by the analog inputs rate1 and rate2. The block will alter the output value according to the analog rate1 input until the r2_sel logic input reads true. At this point, the value is adjusted according to the rate2 schedule (Note: scaling of the rate1 and rate2 analog inputs, in terms of maximum expressible values, should be 1/8th the binary scaling of the cmmand input and the output -- for example, a command output of 256psi. will have a rate input of 32psi/sec.). When the output reaches a value equal to the maximum or minimum range limits, the logic outputs l_max or l_min are set to 1 respectively. These outputs will remain set to zero (0) until a limit is reached. If the block receives a logic 1 l_preset signal, the command input and the output are immediately adjusted to the v_preset analog value (this value is subject to the algorthm's min. and max. limits). For proper block operation, the max_lim, min_lim, command, and v_preset inputs, and the output must be scaled identically. This block differs from the manual setpoint command algorithm as it uses a 32 bit value in its calculations; this allows for a slower ramp rate.



Inputs

Parameter	Data Type	Description:
max_lim	F2	Analog input that determines the highest value the output can be adjusted to.
min_lim	F2	Analog input that determines the lowest value the output can be adjusted to.
rate1	F2	Analog input that determines how quickly the output is raised or lowered to command input value. The rate1 value is 1/8 binary scale of the output.
1_selr2	L1	A logic 1 on this input will instruct the block to adjust the output according to rate2 schedule.
rate2	F2	Analog input that determines how quickly the output is raised or lowered to the command input value when the 1_selr2 input reads 1. The rate2 input is 1/8 binary scale of the o tput.
command	F2	Analog input that the output will be adjusted to. If this value exceeds the maximum or minimum range limits of the block, the output will be set to the limit that is applicable.
l_prset	L1	Logic input that enables the v_prset value to be passed immediately to the output (subject to the block's min. and max. limits).
v_prset	F2	Analog value that is passed directly to the output when the l_prset logic is a one (1). (This value is subject to the block's min. and max. limits.

Outputs

Parameter	Data Type	Description:
at_max	L1	Logical output that is set to 1 when the output value equals the maximum limit value.
at_min	L1	Logical output that is set to 1 when the output value equals the minimum limit value.
output	F2	MSW of a Double Word Variable, Analog value that is adjusted to reflect the current command input value.

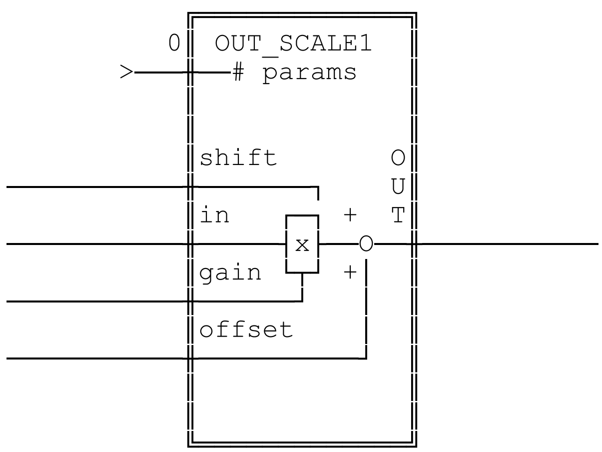
C-6.27. OUT_SCALE1 -- SCALING MODULE FOR OUTPUTS

OUT_SCALE1

outpt_1	outpt_2	outpt_3	outpt_4	outpt_5	outpt_6	outpt_7
shift_1	shift_2	shift_3	shift_4	shift_5	shift_6	shift_7
input_1	input_2	input_3	input_4	input_5	input_6	input_7
gain_1	gain_2	gain_3	gain_4	gain_5	gain_6	gain_7
offst1	offst_2	Offst_3	Offst_4	Offst_5	Offst_6	offst_7
outpt_8	shift_8	outpt_9	shift_9	input_9	outpt-10	shift_10
input_8	gain_8	gain_9	offst_9	input_10	gain_10	offst_10
offst_8	outpt_11	shift_11	outpt_12	shift_12	input_12	
input_11	gain_11	offst_11	gain_11	offst_12		

Description

The Scaling Module for Outputs (Out_Scale1) block executes a function that modifies the scaling of a signal. This process begins by multiplying the analog Input value by a gain factor. The binary value of this multiplication is shifted according to the analog value assigned as the shift constant. When the shift function has been executed, the scaled value is added to a machine unit analog offset constant. This sum is then passed to the output.



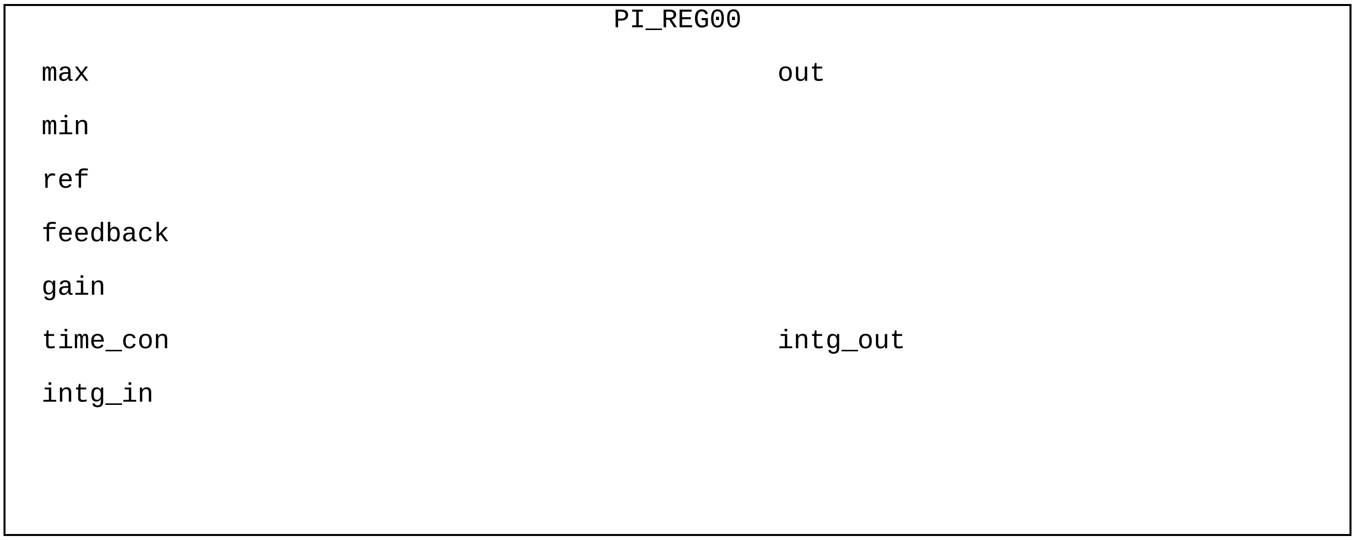
Inputs

Parameter	Data Type	Description:
# params	K2	Number of entry sets for OUT_SCALE1 algorithm. This block may be used multiple times within a single Control Sequence Program (max. 12 sets of entries per call of the block). This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry sets.
in	F2	Analog value scaled in engineering units.
gain	F2	Analog value by which the Input is multiplied before the shift function is executed.
shift	F2	Analog constant value that modifies the binary point location of the input value before the offset function is executed.
offset	F2	Analog constant value that is added to the result of the gain and shift operations.

Outputs

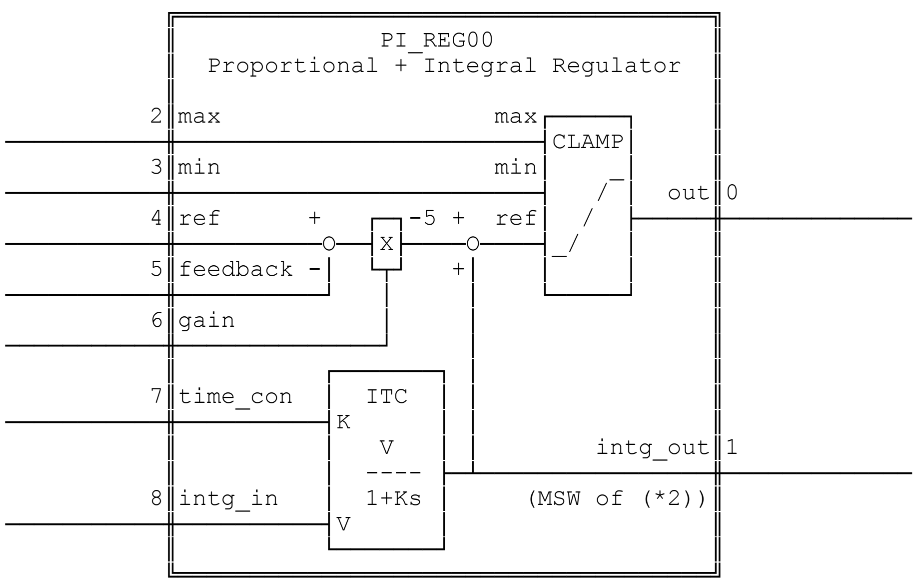
Parameter	Data Type	Description:
output	F2	Analog value expressed in the desired units.

C-6.28. PI_REG00 -- PROPORTIONAL + INTEGRAL REGULATOR



Description

The PI_REG00 block executes a function that increments the output (out) toward a reference (ref) at a rate determined by proportional and integral calculations. Initially, the proportional action is performed by subtracting a feedback value from the reference. The difference (error) is then multiplied by a gain (the -5 value indicates a binary shift). The block's integral action is defined by the integral time constant (time_con); this value is expressed in seconds. (Note: the time constant value should be at least four times the sequencing scan interval). The output of the integrator is added to the result of the proportional calculation. This value is clamped according to the min and max inputs before being passed to the output (out).



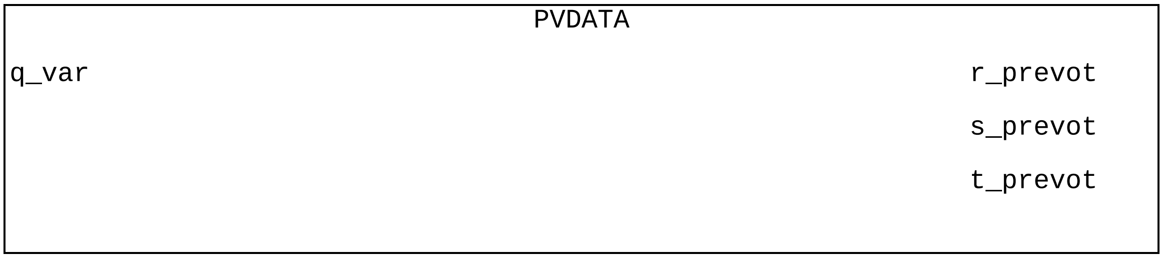
Inputs

Parameter	Data Type	Description:
max	F2	Analog value that defines the maximum value that can be passed to the output (out).
min	F2	Analog value that defines the minimum value that can be passed to the output (out).
ref	F2	Input value from which the feedback signal is subtracted.
feedback	F2	Analog input that provides the device's current position.
gain	F2	Constant input that is multiplied by the difference between the reference (ref) and the feedback values. The result of this calculation has its binary point shifted by five places (-5).
time_con	F2	Constant that defines the integral time constant.
intg_in	F2	Analog input upon which the integral action is performed.

Outputs

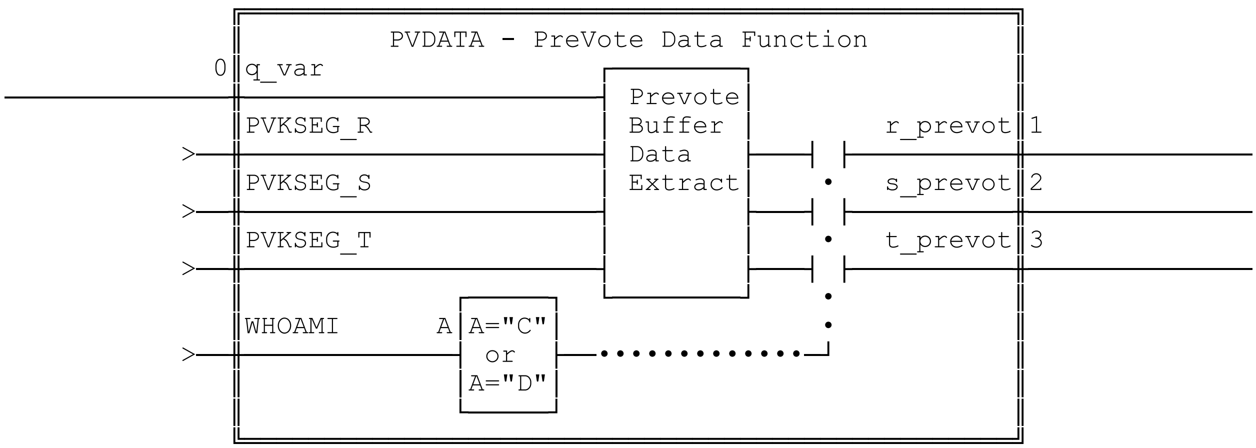
Parameter	Data Type	Description:
out	F2	Resulting value of the block's proportional and integral actions.
intg_out	U2	Result of integral operation. This value represents the MSW of a two-word variable.

C-6.29. PVDATA -- PREVOTE DATA FUNCTION



Description

This module extracts the prevote values for the designated input variable and stores the values in the specified output locations. This function may only be executed in the <C> or <D> processors. It requires that sufficient local variable space be available to store the results.



Inputs

Parameter	Data Type	Description:
q_var	V2	Variable input from <Q>.

Outputs

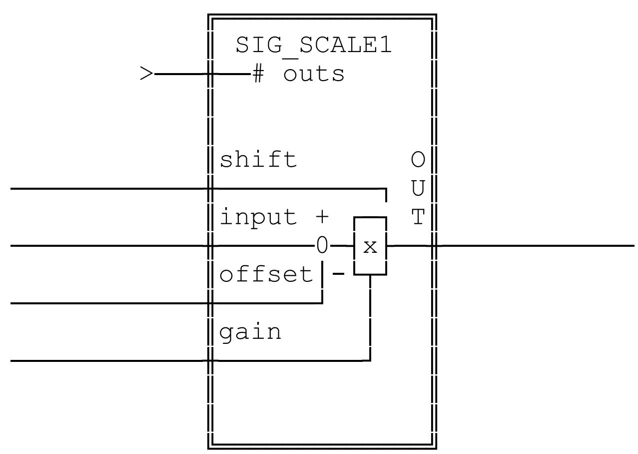
Parameter	Data Type	Description:
r_prevot	V2	Variable output corresponding to <R> cores prevoted value stored in the <C> core.
s_prevot	V2	Variable output corresponding to <S> cores prevoted value stored in the <C> core.
t_prevot	V2	Variable output corresponding to <T> cores prevoted value stored in the <C> core.

C-6.30. SIG_SCALE1 -- SIGNAL SCALING

```
                                   SIG_SCALE1
outpt_1  outpt_2  outpt_3  outpt_4  outpt_5  outpt_6  outpt_7
shift_1  shift_2  shift_3  shift_4  shift_5  shift_6  shift_7
input_1  input_2  input_3  input_4  input_5  input_6  input_7
offst_1  offst_2  offst_3  offst_4  Offst_5  offst_6  offst_7
gain_1   gain_2   gain_3   gain_4   gain_5   Gain_6   gain_7
outpt_8  shift_8  outpt_9  shift_9  input_9  outpt_10 shift_10
input_8  offst_8  offst_9  gain_9   input_10 offst_10 gain_10
gain_8   outpt_11 shift_11 outpt_12 shift_12 input_12
input_11 offst_11 gain_11  offst_12 gain_12
```

Description

The Signal Scaling (Sig_Scale1) block executes a function that converts inputs into scaled engineering units. Initially, an offset constant is subtracted from an analog input value (eg. milliamps). This sum is then multiplied by a gain factor; the binary value is shifted according to the analog value assigned as the shift constant. When the Shift function has been executed, the scaled value is passed to the output.



Inputs

Parameter	Data Type	Description:
input	F2	Analog value from which the offset is subtracted, multiplied times a gain, and shifted before it is passed to the output.
gain	F2	Multiplication factor applied to the result of the subtraction operation.
offset	F2	Analog value that is subtracted from the Input before it is multiplied by the gain factor.
shift	F2	Analog constant value that modifies the binary point location of the output.
# outs	K2	Number of entry sets for SIG_SCALE1 algorithm. This block may be used multiple times within a single Control Sequence Program (max. 12 sets of entries per call of the block). This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry sets.

Outputs

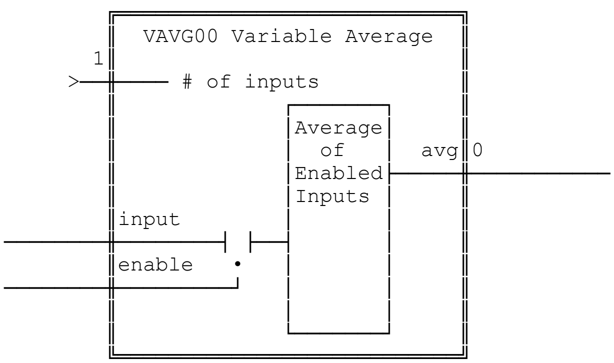
Parameter	Data Type	Description:
output	F2	Analog value that has been converted to different scaling units.

C-6.31. VAVG00 -- VARIABLE AVERAGE

```
                              VAVG00
                                                                        output
input_1  enable_1  input_2  enable_2  input_3  enable_3  input_4  enable_4
input_5  enable_5  input_6  enable_6  input_7  enable_7  input_8  enable_8
input_9  enable_9  input_10 enable_10 input_11 enable_11 input_12 enable_12
input_13 enable_13 input_14 enable_14 input_15 enable_15 input_16 enable_16
input_17 enable_17 input_18 enable_18 input_19 enable_19 input_20 enable_20
input_21 enable_21 input_22 enable_22 input_23 enable_23 input_24 enable_24
input_25 enable_25 input_26 enable_26 input_27 enable_27 input_28 enable_28
input_29 enable_29 input_30 enable_30
```

Description

The Variable Average block calculates an average of all the enabled inputs. Up to thirty analog values can be defined as inputs to the block. With each analog values is a corresponding logical input. If the logical input is set to one (1) then the associated analog value is used in the averaging calculation. If the logical input is set to zero (0) then the associated analog value is not used. The calculation is the total of all enabled analog inputs divided by the total number of logical inputs set to one (1).



Inputs

Parameter	Data Type	Description:
input_n	F2	Analog variable inputs to be averaged. Should all configured with the same scale code.
enable_n	L1	Corresponding logical input. Set to one (1) to enable its analog variable to be used in calculation.

Outputs

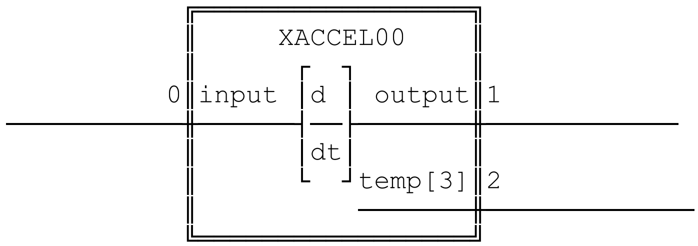
Parameter	Data Type	Description:
avg	F2	Output analog value, average of all enabled analog inputs.

C-6.32. XACCEL00 -- ANALOG DIFFERENTIATOR

```
                        XACCEL00
 input                                              output
                          temp
```

Description

The Analog Differential block executes a function that computes the differential of the analog input with respect to time.



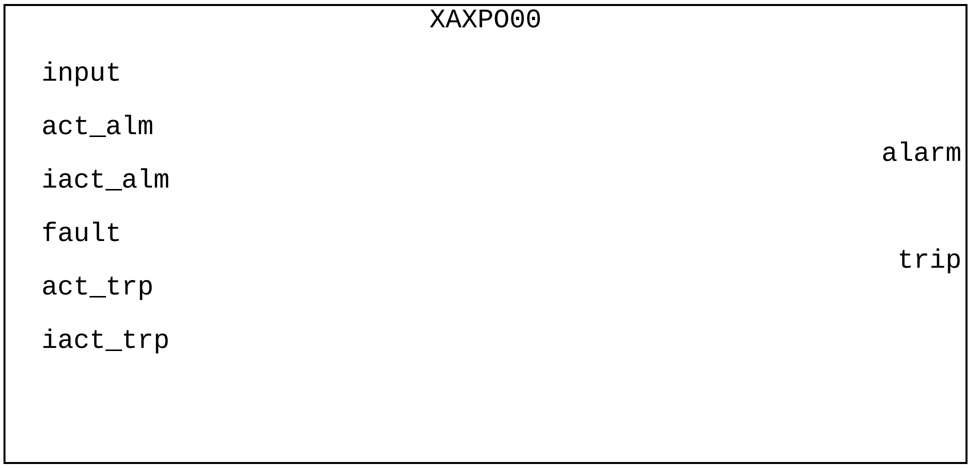
Inputs

Parameter	Data Type	Description:
input	F2	Analog input.
temp	F2	Temporary storage location that stores three previous input values. Values are stored on a first in - first out basis.

Outputs

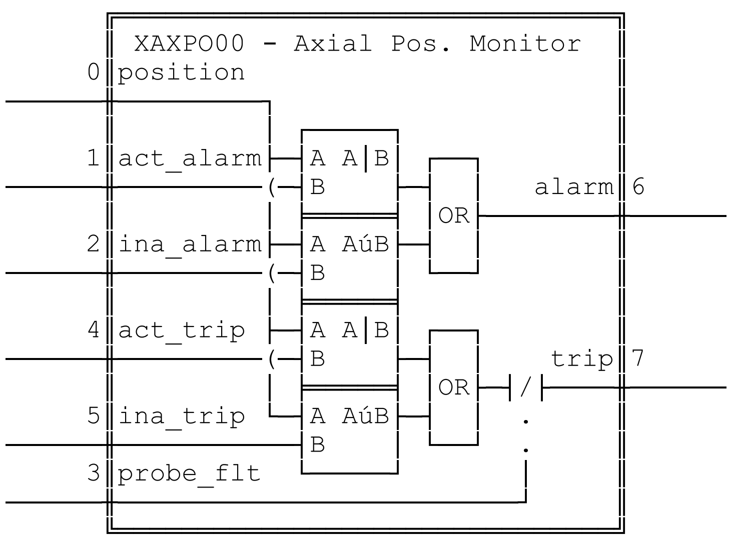
Parameter	Data Type	Description:
output	F2	Analog value that represents the rate of change of the input over time.

C-6.33. XAXPO00 -- AXIAL POSITION MONITOR



Description

The Axial Position Monitor block checks the axial position of the turbine and generates alarm and trip logic outputs according to four limit constants (active alarm, inactive alarm, active trip, inactive trip). The algorithm receives conditioned analog signals from an I/O processor card mounted within the panel, which in turn acquires its data from a turbine mounted proximeter (note that "conditioned" here implies that the I/O processor card converts analog values received from the proximeter to the applicable engineering units, and adds an offset to those figures before they are passed to the block). This last device reads sensor measurements of active and inactive turbine shaft movements. Analog shaft movement input values that equal or exceed the active/inactive limit constants supplied to the block will cause a corresponding alarm or trip logic signal to be generated. A probe fault detection by the I/O processor card will result in a logic 1 being sent to the fault input, thereby disabling of the block's trip output. Should this occur however, the alarm output will remain enabled. All inputs should be scaled identically for proper block operation.



Inputs

Parameter	Data Type	Description:
input	F2	Analog input that represents the turbine shaft's active/inactive position as defined by the engineering units applicable (must be scaled identically to other block inputs).
act_alm	F2	Analog constant value that when equaled or exceeded by the input, will cause a logic 1 to be passed to the alarm output. Must be scaled identically to other block inputs.
iact_alm	F2	Analog constant value that when equaled or exceeded by the input, will cause a logic 1 to be passed to the alarm output. Must be scaled identically to other block inputs.
fault	L1	Logic input that disables the algorithm's trip output when a logic 1 is received.
act_trp	F2	Analog constant value that when equaled or exceeded by the input, will cause a logic 1 to be passed to the trip output. Must be scaled identically to other block inputs.
iact_trp	F2	Analog constant value that when equaled the input, will cause a logic 1 to be passed to the trip output. Must be scaled identically to other block inputs.

Outputs

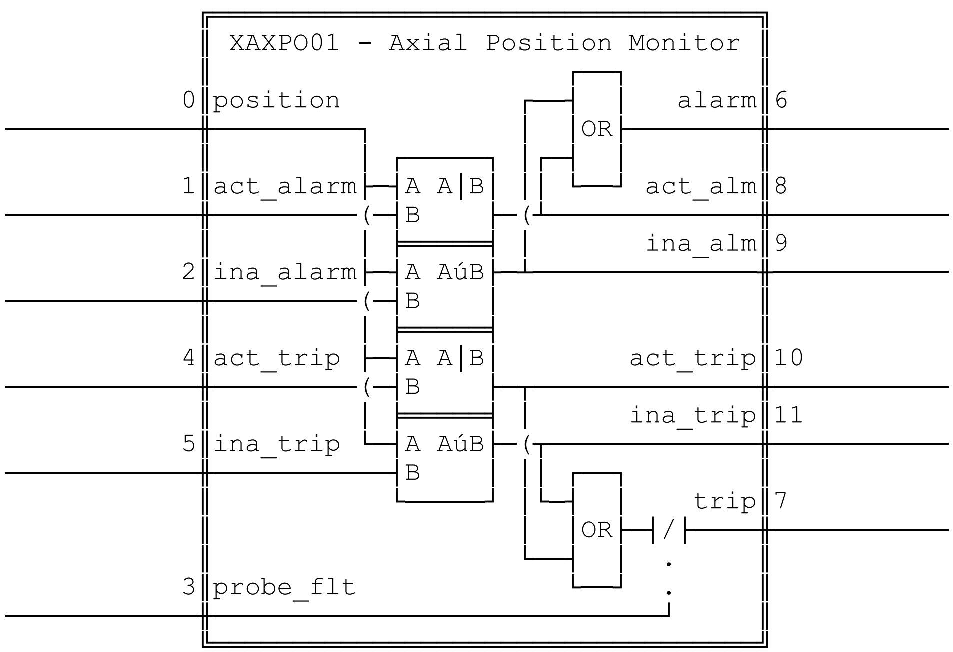
Parameter	Data Type	Description:
alarm	L1	Logic output that reads 1 when the Input equals or exceeds either the act_alm or iact_alm values.
trip	L1	Logic output that reads 1 when the Input equals or exceeds either the act_trp or iact_trp values.

C-6.34. XAXPO01 -- AXIAL POSITION MONITOR

XAXPO01	
position	alarm
act_alm	act_alm
iact_alm	ina_alm
act_trp	act_trip
iact_trp	ina_trip
fault	trip

Description

The Axial Position Monitor block (version01) checks the axial position of the turbine and generates alarm (alarm, act_alm, ina_alm) and trip (act_trip, ina_trip, trip) logic outputs according to four limit constants (active alarm, inactive alarm, active trip, inactive trip). The algorithm receives conditioned analog signals from an I/O processor card mounted within the panel, which in turn acquires its data from a turbine mounted proximeter (note that "conditioned" here implies that the I/O processor card converts analog values received from the proximeter to the applicable engineering units and adds an offset to those figures before they are passed to the block). This last device reads sensor measurements of active and inactive turbine shaft movements. Analog shaft movement " position " values that equal or exceed the active/inactive limit constants supplied to the block will cause a corresponding alarm or trip logic signal to be generated. A probe fault detection by the I/O processor card will result in a logic 1 being sent to probe fault input, thereby disabling of the block's trip output. Should this occur however, the alarm output will remain enabled. All inputs should be scaled identically for proper block operation.



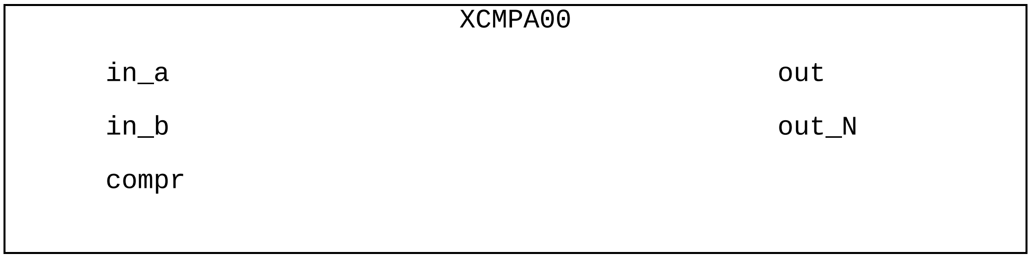
Inputs

Parameter	Data Type	Description:
position	F2	Analog input that represents the turbine shaft's active/inactive position as defined by the engineering units applicable (must be scaled identically to other block inputs).
act_alarm	F2	Analog constant value that when equaled or exceeded by the position input, will cause a logic 1 to be passed to the alarm and active alarm output. Must be scaled identically to other block inputs.
ina_alarm	F2	Analog constant value that when equaled or exceeded by the posiiton input, will cause a logic 1 to be passed to the alarm and inacitve alarm outputs. Must be scaled identically to other block inputs.
act_trp	F2	Analog constant value that when equaled or exceeded by the position input, will cause a logic 1 to be passed to the trip output. Must be scaled identically to other block inputs.
iact_trp	F2	Analog constant value that when equaled or exceeded by the position input, will cause a logic 1 to be passed to the trip output. Must be scaled identically to other block inputs.
fault	L1	Logic input that disables the algorithm's trip output when a logic 1 is received.

Outputs

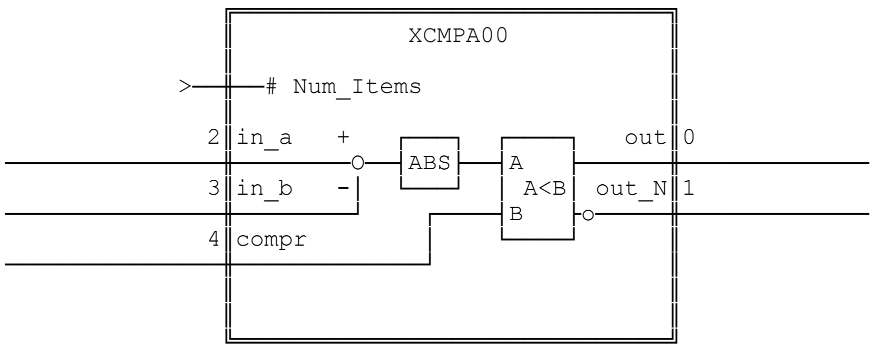
Parameter	Data Type	Description:
alarm	L1	Logic output that reads 1 when the position input equals or exceeds either the act_alm or iact_alm values.
act_alm	L1	Logic output that reads 1 when the position input equals or exceeds act_alrm input constant.
ina_alm	L1	Logic output that reads 1 when the position input equals or exceeds the ina_alm analop constant.
act_trip	L1	Logic output that reads 1 when the position input equals or exceeds the act_trip analog constant.
ina_trip	L1	Logic output that reads 1 when the position input equals or exceeds the ina_trip analog constant.
trip	L1	Logic output that reads 1 when the position input equals or exceeds either the act_trip or iact_trip values.

C-6.35. XCMPA00 -- COMPARE ABSOLUTE VALUE



Description

The XCMP00 block executes a function that writes to two logic outputs (out, out_N) based upon a comparison between an input signal (compr) and the absolute value (ABS) of the difference between two analog values (in_a, in_b). If the (absolute) differential value is less than the input value the out and out_N outputs will be set to true and false respectively; a differential value that is greater than the cmpr input will cause the logic states of the two outputs to be reversed.



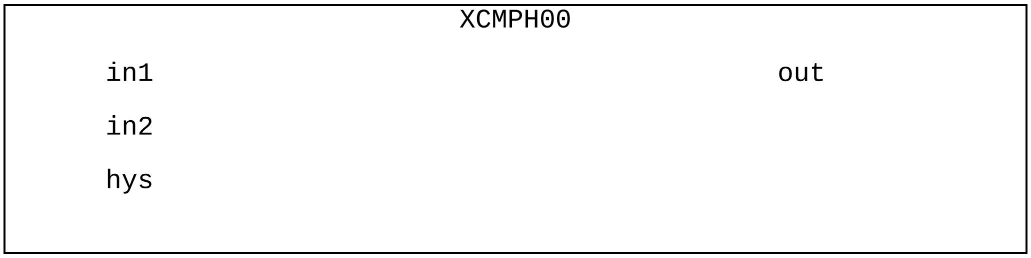
Inputs

Parameter	Data Type	Description:
# Num_items	K2	Number of entry sets for XCMP00 algorithm. This block may be used multiple times within a single Control Sequence Program (max. 12 sets of entries per call of the block). This value does not require a value to be assigned. The value will be assigned by the sequencing compiler based on the number of entry sets.
in_a	F2	Analog variable from which the analog variable in_b is subtracted..
in_b	F2	Anolog variable that is subtracted from the in_a analog variable.
compr	F2	Analog value that is compared to the absolute difference between inputs in_a and in_b.

Outputs

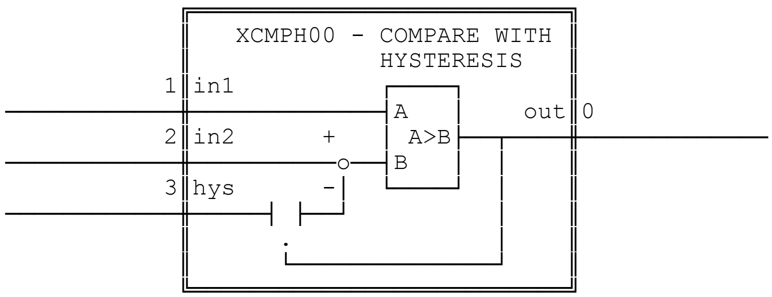
Parameter	Data Type	Description:
out	L1	Logic output that is set true when the absolute difference between in_a and in_b is less than the compr input constant.
out_N	L1	Logic output that is set true when the absolute difference between in_a and in_b inputs is greater than the compr input constant.

C-6.36. XCMPH00 -- COMPARE WITH HYSTERESIS



Description

The XCMPH00 block executes a function that writes a logic signal output when input A (in1) exceeds input B (in2). When this occurs, the hysteresis is enabled and the hys value is subtracted from the in2 input. This operation effetively lowers the setpoint value the input must drop below before the output will reset to zero (0).



Inputs

Parameter	Data Type	Description:
in1	F2	Analog variable that is compared to in2. If in1 exceeds in2, the ouput will be set.
in2	F2	Analog variable that is subtracted from the in_a.
hys	F2	Analog variable that is subtracted from in2 when out is true.

Outputs

Parameter	Data Type	Description:
out	L1	Logic output that is set true when in1 exceeds in2.

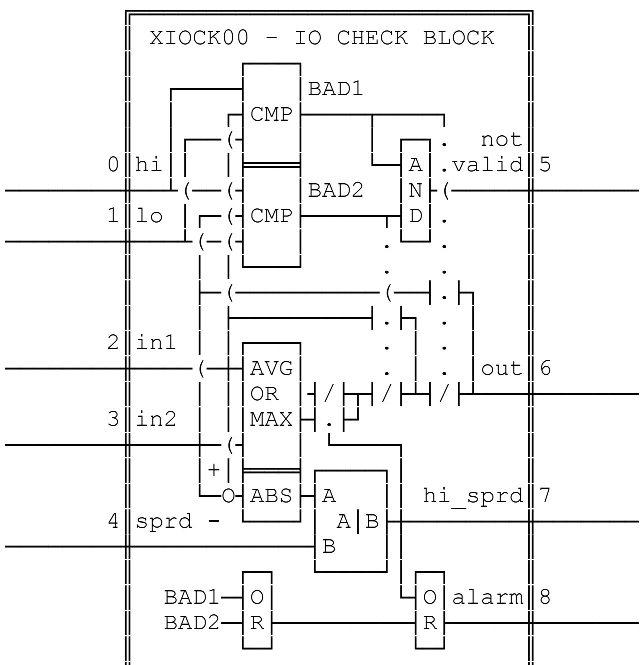
C-6.37. XIOCK00 -- DUAL INPUT CHECK BLOCK

XIOCK00	
high_lim	invalid
low_lim	output
input_1	hi_sprd
input_2	alarm
sprd_lim	

Description

The Dual Input Check block (XIOXCK00) executes a function that compares a pair of analog (typically thermocouple) inputs against each other and high (high_lim) and low (Low_Lim) analog limit constants. The spread limit (spd_lim) analog constant determines the amount of variance that can occur between the input_1 and input_2 values before a logic 1 is passed to the hi_sprd output (typically this value will be less than high_lim and low_lim inputs). If the inputs are within the spread limit and do not exceed either high or low limits, the block will pass the average of the input values as the output. When the input values exceed the spd_lim but remain within the high and low limits, the higher of the two values is passed to the output: should a single input value exceed a limit, the concurring value is passed. A condition where both inputs exceed the limit values will cause an logic 1 invalid output to be passed. If an input value exceeds any of the limits expressed above, an alarm output signal is generated.

All input values must be scaled identically for proper block operation.



Inputs

Parameter	Data Type	Description:
high_lim	F2	Analog constant that determines the maximum value an input can reach before an alarm output is generated.
low_lim	F2	Analog constant that determines the minimum value an input can reach before an alarm output is generated.
input_1	F2	The first of a pair of analog input values that are compared against each other and the high and low limit constants.
input_2	F2	The second of a pair of analog input values that are compared against each other and the high and low limit constants.
sprd_lim	F2	Analog constant value that determines the degree to which the two input values can vary before a hi_sprd logic output is generated.

Outputs

Parameter	Data Type	Description:
invalid	L1	Logical output that reads 1 when both inputs exceed a high or low limit.
output	F2	Analog value that expresses one of three conditions: an average of the two inputs, a single input if a corresponding input exceeds a high or low limit, or, the greater of the two input values if the spread limit has been exceeded but the high or low limits have not.
hi_sprd	L1	Logical output that reads 1 when the variance between the inputs exceeds the sprd_lim value.
alarm	L1	Logical output that reads 1 when any of the block limits are exceeded by an input value.

C-6.38. XMOV01 -- MOTOR OPERATED VALVE BLOCK (Version 01)

```
                                        XMOV01
  acloscmd  mcloscmd        vlvclsd           closevlv
  aopencmd  mopencmd        vlvopen           openvlv
  aintrcmd  mintrcmd        vlvintr           poweron
  auto_sel
  kbacktrk
  ktotal
                      memtim   movstate         alarm
```

Description

The Motor Operated Valve block (XMOV01) executes a function that controls valve position by issuing valve motor commands (openvlv, closevlv). These instructions are passed according to one of two possible modes (manual, auto). If the manual mode is selected, the block reads its manual input commands (manual open, manual close, manual intermediate) and the position of the valve (valve open, valve closed, valve in intermediate) -- if the two readings do not correspond, the motor is instructed to move the valve to the desired location. An auto mode selection will see the block read a new set of inputs (auto open, auto close, auto intermediate) before the current valve position is read; instructions passed to the block are thereby determined by automated sequencing. In either mode, movement of the valve is controlled by the poweron and either closevlv or openvlv logic outputs (only one of the closevlv/openvlv logic signals will read true at a time). To specify the valve's position or motion, the block passes enumerated state values that correspond to the following Motor Operated Valve states (movstate): mov state unknown, mov trouble (alarm will be generated), closed, open, in intermediate, closing, opening, closing toward intermediate, opening toward intermediate, backtracking toward intermediate. The movstate will read "mov trouble" and the block will generate and alarm (alarm) when the valve's travel time (memtim) exceeds the interval specified by the ktotal time constant. To compensate for motor inertia and resultant valve drift that occurs when a valve is moved to an intermediate position, the block executes a backtrack function. This operation causes the valve to move in a direction opposite to the original open/close command for a period of time specified by the kbacktrk constant. This action is denoted by the movstate "backtracking towards intermediate." During initialization, the block will have no input to reference the valve's position; under this circumstance, the movstate will read "mov state unknown." This block is generally used for steam turbine valve control.

```
        XMOV01  - Motor Operated
            Valve Driver Version 1

          INPUTS         OUTPUTS

     0  auto close cmd    close valve  10
     1  auto open cmd     open  valve  11
     2  auto interm cmd          run   12

     3  manual close cmd
     4  manual open cmd
     5  manual interm cmd

     6  valve is closed
     7  valve is open
     8  valve in intermed

     9  auto/man selected
                                alarm  13
    14  backtrack time
    15  max travel time
    16  (timer)           valve state  17
```

Inputs

Parameter	Data Type	Description:
auto_sel	L1	Logic input that instructs the block to set the valve position according to auto sequencing commands when true; otherwise, the block will set the valve position according to manual commands.
aopencmd	L1	Logic input that instructs the block to set the valve position to full open when the block is in auto mode.
acloscmd	L1	Logic input that instructs the block to set the valve position to full close when the block is in auto mode.
aintrcmd	L1	Logic input that instructs the bock to set the valve position to intermediate when the block is in auto mode.
mopencmd	L1	Logic input that instructs the block to set the valve the position to full open when block is in manual mode.
mcloscmd	L1	Logic input that instructs the block to set the valve position to full close when the block is in manual mode.
mintrcmd	L1	Logic input that instructs the block to set the valve position to intermediate when the block is in manual mode.
vlvlclsd	L1	Logic input that reads 1 when the valve is in the full close position.
vlvopen	L1	Logic input that reads 1 when the valve is in the full open position.
vlvIntr	L1	Logic input that reads 1 when the valve is in the intermediate position.
kbacktrk	F2	Analog constant that defines the period of time the valve is allowed to backtrack.
ktotal	F2	Analog constant that defines the maximum allowable time period the block may transmit a valve motor command without generating an alarm signal.
memtim	F2	Timer input that begins counting when a valve motor command is generated. This timer will reset to zero (0) when a logic signal is received confirming the completion valve's movement.

Outputs

Parameter	Data Type	Description:
openvlv	L1	Logic output that instructs the motor to move the valve position toward full open.
closevlv	L1	Logic output that instructs the motor to move the valve position toward full close.
poweron	L1	Logic output that turns the valve motor on.
alarm	L1	Logic output that is generated when memtim exceeds ktotal.
movstate	S2	Enumerated state variable that denotes valve action or position.

C-6.39. XMOV02 -- MOTOR OPERATED VALVE BLOCK (Version 02)

```
                                XMOV02
acloscmd  mcloscmd        auto_sel          closevlv
aopenCmd  mopenCmd        vlvposit          openvlv
aintrCmd  mintrcmd                          poweron
intrref  kbacktrk
khyster  ktotal
                    memtim   movstate        alarm
```

Description

The Motor Operated Valve block (XMOV02) executes a function that controls valve position by issuing valve motor commands (openvlv, closevlv). These instructions are passed according to one of two possible modes (manual, auto). If the manual mode is selected, the block reads its manual input commands (manual open, manual close, manual intermediate) and the position of the valve (valve open, valve closed, valve in intermediate) -- if the two readings do not correspond, the motor is instructed to move the valve to the desired location. An auto mode selection will see the block read a new set of inputs (auto open, auto close, auto intermediate) before the current valve position is read; instructions passed to the block are thereby determined by automated sequencing. In either mode, movement of the valve is controlled by the poweron and either closevlv or openvlv logic outputs (only one of the closevlv/openvlv logic signals will read true (1) at a time). As the location of the valve (vlvposit) is referenced according to an analog feedback signal (0% = closed, 100% = OPEN), a hysteresis function is supplied to prevent valve control dithering. The hysteresis constant (khyster) specifies the distance the valve may deviate above or below required setpoints before corrective action is taken (khyster values are expressed in %). It should be noted that the relationship between % values and valve stroke can be manipulated using the I/O Configurator. In other words, valve position (0% - 100%) can be configured to correspond to any valve travel range (eg. 0% - 10%). The algorithm augments percent (%) valve position data by passing enumerated state values that correspond to the following motor operated valve states (movstate): mov state unknown, mov trouble (alarm will be generated), closed, open, in intermediate, closing, opening, closing toward intermediate, opening toward intermediate, backtracking toward intermediate. The movstate will read "mov trouble" and the block will generate and alarm (alarm) when the valve's travel time (memtim) exceeds the interval specified by the ktotal time constant. To compensate for motor inertia and resultant valve drift that occurs when a valve is moved to an intermediate position, the block executes a backtrack function (the intermediate position is defined by the intermediate position reference (intrref) constant (this value is also expressed in %). This operation causes the valve to move in a direction opposite to the original open/close command for a period of time specified by the kbacktrk constant (again, the hysteresis function is implemented to avoid dithering). This action is denoted by the movstate "backtracking towards intermediate." During initialization, the block will have no input to reference the valve's position. Under this circumstance, the movstate will read "mov state unknown." This block is generally used for steam turbine valve control.

```
                 XMOV02  - Motor Operated
                     Valve Driver Version 2

                  INPUTS          OUTPUTS

              0 auto close cmd   close valve 8
              1 auto open cmd    open  valve 9
              2 auto interm cmd         run  10

              3 manual close cmd
              4 manual open cmd
              5 manual interm cmd

              6 auto/man selected

              7 valve position

             12 interm posit. ref
             13 posit.hysteresis       ALARM 11
             14 backtrack time
             15 max travel time
             16 (timer)          valve state 17
```

Inputs

Parameter	Data Type	Description:
auto_sel	L1	Logic input that instructs the block to set the valve position according to auto sequencing commands when TRUE; otherwise, the block will set the valve position according to manual commands.
aopencmd	L1	Logic input that instructs the block to set the valve position to full open when the block is in auto mode.
acloscmd	L1	Logic input that instructs the block to set the valve position to full close when the block is in auto mode.
aintrcmd	L1	Logic input that instructs the bock to set the valve position to intermediate when the block is in auto mode.
mopencmd	L1	Logic input that instructs the block to set the valve position to full open when block is in manual mode.
mcloscmd	L1	Logic input that instructs the block to set the valve position to full close when the block is in manual mode.
mintrcmd	L1	Logic input that instructs the block to set the valve position to intermediate when the block is in manual mode.
vlvposit	F2	analog feedback reference that indicates the valve's position by means of a % value.
intrref	F2	analog constant that defines the valve's intermediate position. This value is expressed in % value.
khyster	F2	analog constant that defines the amount by which a valve may deviate above or below a specified setpoint without causing an alarm to be generated.
kbacktrk	F2	analog constant that defines the period of time the valve is allowed to backtrack.
ktotal	F2	analog constant that defines the maximum allowable time period the block may transmit a valve motor command without generating an alarm signal.
memtim	F2	Timer input that begins counting when a valve motor command is generated. This timer will reset to zero (0) when a logic signal is received confirming the completion valve's movement.

Outputs

Parameter	Data Type	Description:
openvlv	L1	Logic output that instructs the motor to move the valve position toward full open.
closvlv	L1	Logic output that instructs the motor to move the valve position toward full close.
poweron	L1	Logic output that turns the valve motor on.
alarm	L1	Logic output that is generated when memtim exceeds ktotal.
MOVState	S2	Enumerated state variable that denotes valve action or position.

C-6.40. XPID01 -- PROPORTIONAL INTEGRAL DERIVATIVE BLOCK

```
                                XPID01
setpoint
prp_gain
rev_act
int_tc                                                          output
meas_in                                                       int_comp
deriv_tc                                                       prev_in
bias
lim_array
man_sel  man_stpt
```

Description

The Proportional Integral Derivative (PID) control block (XPID01) executes a function that reads analog feedback information (meas_in), and according to the mode selected (manual, auto), produces an output that controls the process being monitored. When the man_sel logic input reads one, the block transitions to a manual procedure that causes the manual setpoint (man_sp) value to be passed directly to the output (this value is passed regardless of the min_out and max_out limits stipulated in the limit array (lim_array). If the man_sel reads 0, the block will perform a PID operation that sets the output according to a summation of Proportional, Integral, and Derivative compensations, and a bias value. Unlike the manual select mode, the auto mode output is clamped between the min_out and max_out limits stored in the limit array (lim_array).

The Proportional action of the block is defined by the gain (prop_gain). This action is equal to error (measured value meas_in - setpoint) * Prop_Gain. A logic FALSE rev_acting input will cause an increasing output for a positive or negative Error; a logic TRUE Rev_Acting input will cause a decreasing output for a positive or negative Error.

The amount of integral action is defined by the integral time constant (int_tc). This value and the derivative time constant are both expressed in seconds; the value of each time constant should be at least four (4) times the sequencing scan interval at which the block is executed. The integrator is clamped so as to remain within the limits min_out and max_out -- it will not wind up (continue to max. value) when the output is saturated. If the output is saturated high or low, the integrator will not increase or decrease respectively. When a saturation occurs, an inverse value (+ -) increment is added to the integrator. When the output is saturated low, the integrator will not decrease; an increment will be added to the integrator only if it is negative. If the integral time constant is set to 0 the integrator will also be 0 and the integral action will be disabled.

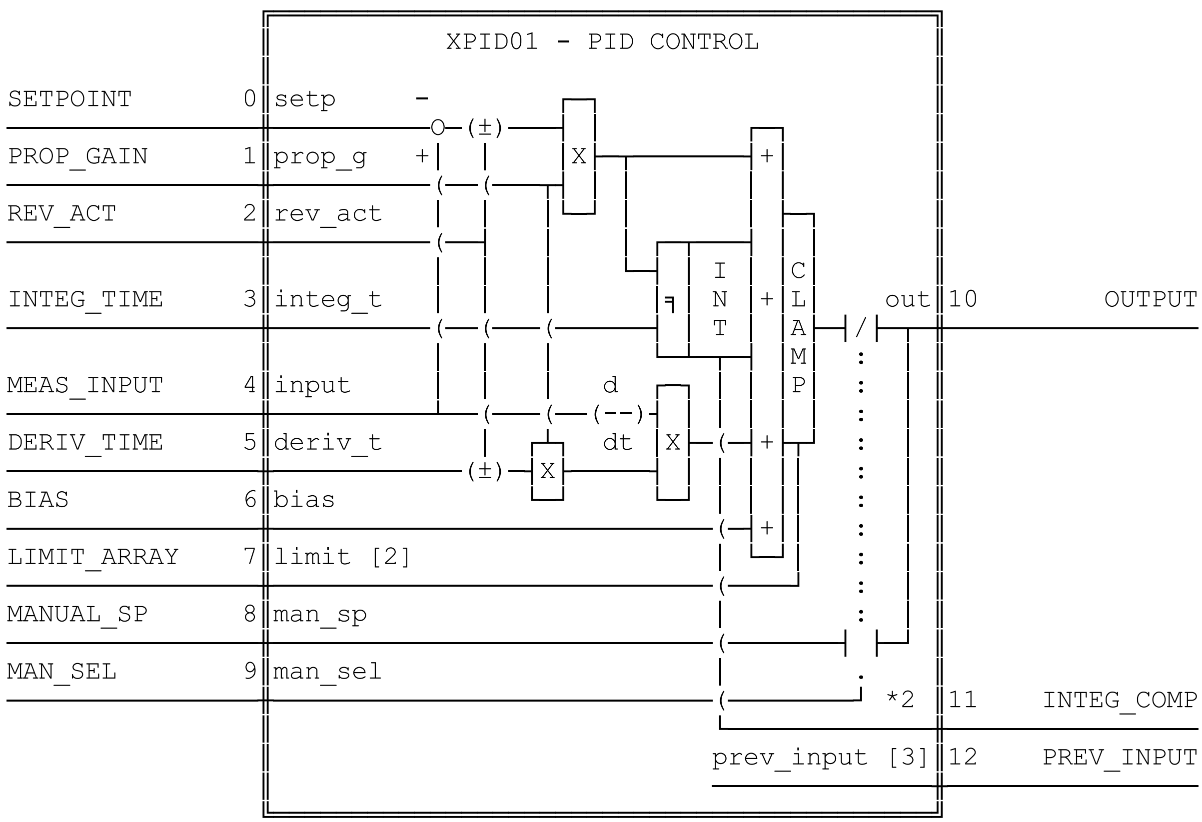
The value of the integrator is saved for use in the next sample. If the block is in manual mode, and the Int_TC is not set to 0, the integrator will be modified to cause the automatic control to track manual_sp (i.e. I = manual_sp - (P+D+B). The automatic control will track manual_sp only to the limits min_out and max_out. Under these conditions, transfer from manual to automatic mode is bumpless if manual_sp remains within the automatic mode limits.

The amount of Derivative action is defined by the derivative time constant (deriv_tc) and is expressed in seconds. It has been designed to operate on the meas_Input (measured value) rather then the error so that there will not be a spike in the output if there is a step function in the setpoint. The previous three meas_input values are stored in the 3 element array prev_input. The block takes the derivative of the proportional section using techniques that minimize noise. If deriv_tc is set to 0 the derivative action is also zero (0). The storage of the previous meas_Input continues so that a valid change in measured value will be obtained immediately if deriv_tc is changed from 0 to non-0. The bias is added directly to the proportional + integral + derivative actions before the output clipping takes place. The units of bias are the same units of the output. The scaling of the inputs setpoint and meas_input should be in percent.

PID loop transfer function uses the following equation:

$$Output(s) = (-Setpoint + Meas\_in)(s) * [Prp\_Gain\ (1 + \frac{1}{Int\_TC(s)} + Deriv\_TC(s))]$$

Where (s) is the LaPlace transform function. Further information can be obtained from the factory, if necessary.



Inputs

Parameter	Data Type	Description:
setpoint	F1	Analog value that represents the nominal value the block maintains when in Auto mode. Should be scaled in percent.
prp_gain	F2	Analog value that defines proportional action. Should be scaled in percent.
rev_act	L1	A logic 1 on this input causes a decrease in the output value while a 0 produces an increase.
int_tc	F2	Integral time constant; defines the integral action of the block. Expressed in seconds.
meas_in	F2	Analog input that represents the feedback value. Should be scaled in percent.
deriv_tc	F2	Analog constant that defines derivative action of the block. Expressed in seconds.
bias	F2	Analog constant that is added to the sum of proportional, integral, and derivative values. Should be scaled in percent.
lim_array	F2	Two element array that contains minimum and maximum output limits. The first element is the minimum limit, and the second element is the maximum limit. Both should be scaled in percent.
man_stpt	F2	Analog value that is passed directly to the output when the man_sel input reads 1. Should be scaled in percent.
man_sel	L1	Logic input that determines if the man_stpt value is passed to the output. If this input reads 0, the output will equal the sum of the proportional, integral, and derivative actions, and the bias value.

Outputs

Parameter	Data Type	Description:
output	F2	Analog value that represents either the man_sp value (man_sel 1), or, the sum of the proportional, integral, and derivative actions, and the bias value (man_sel 0). Should be scaled in percent.
int_comp	F2	Double word variable that stores the incremental change and accumulative total of integral action. Should be scaled in percent.
prev_in	F2	3 element array containing the three previous measured values. Should be scaled in percent.

C-6.41. XPID02 -- PROPORTIONAL INTEGRAL DERIVATIVE BLOCK (VER. 02)

```
                                   XPID02
  setpoint
  prp_gain
  rev_act
  int_tc                                                            output
  meas_in                                                         int_comp
  deriv_tc                                                          rev_in
  deriv_db
  bias
  lim_array
  man_sel  man_stpt
```

Description

The Proportional Integral Derivative (PID) control block (XPID02) executes a function that reads analog feedback information (input), and according to the mode selected (manual, auto), produces an output that controls the process being monitored. When the man_sel logic input reads one, the block transitions to a manual procedure that causes the manual setpoint (man_stpt) value to be passed directly to the output (this value is passed regardless of the min. and max. limits stipulated in the limit array (lim_array). If the man_sel reads 0, the block will perform a PID operation that sets the output according to a summation of proportional, integral, and derivative compensations, and a bias value. Unlike the manual select mode, the auto mode output is clamped between the min_out and max_out limits stored in the limit array (limits).

The proportional action of the block is defined by the gain (prop_gain). This action is equal to error (measured Value meas_in - setpoint) * prop_gain. A logic false rev_act input will cause an increasing output for a positive or negative error; a logic true rev_acting input will cause a decreasing output for a positive or negative error.

The amount of integral action is defined by the integral time constant (integ_tc). This value and the derivative time constant are both expressed in seconds; the value of the time constant should be at least four (4) times the sequencing scan interval at which the block is executed. The integrator is clamped so as to remain within the min. and max. limits stipulated by the limits input-- it will not wind up (continue to max. value) when the output is saturated. If the output is saturated high or low, the integrator will not increase or decrease respectively. When a saturation occurs, an inverse value (+ -) increment is added to the integrator. When the output is saturated low, the integrator will not decrease; an increment will be added to the integrator only if it is negative. If the integral time constant is set to 0 the integrator will also be 0 and the integral action will be disabled.

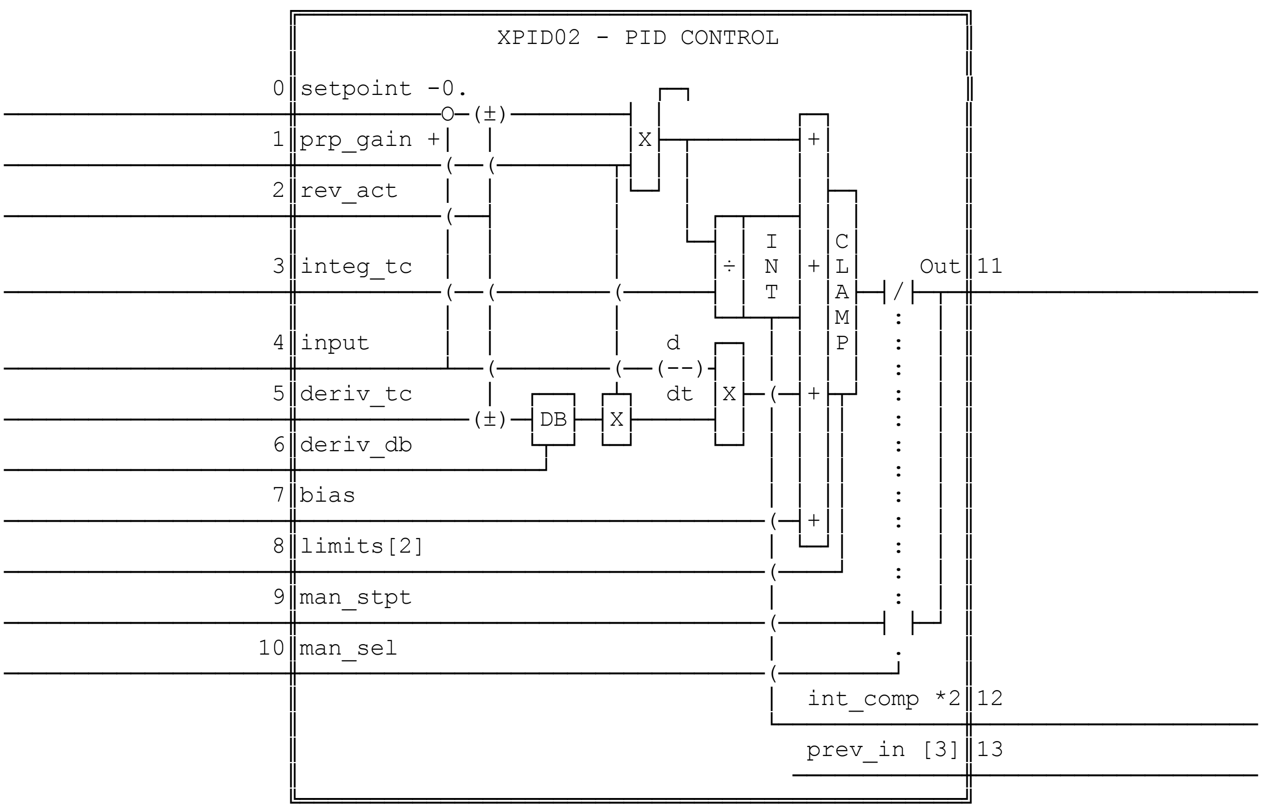
The value of the integrator is saved for use in the next sample. If the block is in manual mode, and the integ_tc is not set to 0, the integrator will be modified to cause the automatic control to track manl_stpt (i.e. I = manual_sp - (P+D+B). The automatic control will track manl_stpt only to the min and max limits. Under these conditions, transfer from manual to automatic mode is bumpless if man_stpt remains within the automatic mode limits.

The amount of Derivative action is defined by the derivative time constant (deriv_tc) and is expressed in seconds. It has been designed to operate on the input (measured value) rather then the error so that there will not be a spike in the output if there is a step function in the setpoint. In addition, the block permits a deadband (deriv_ db) to be introduced to the derivative operation to minimize noise. Three previous three input values are stored in the 3 element array prev_in. The block takes the derivative of the proportional section using techniques that minimize noise. If deriv_tc is set to 0 the derivative action is also 0. The storage of the previous meas_input continues so that a valid change in measured value will be obtained immediately if deriv_tc is changed from 0 to non-0. The bias is added directly to the proportional + integral + derivative actions before the output clipping takes place. The units of bias are the same units of the output. The scaling of the inputs setpoint and meas_input should be in percent.

PID loop transfer function uses the following equation:

$$Output(s) = (-Setpoint + Meas\_in)(s) * [Prp\_Gain (1 + \frac{1}{Int\_TC(s)} + Deriv\_TC(s))]$$

Where (s) is the LaPlace transform function. Further information can be obtained from the factory, if necessary.



Inputs

Parameter	Data Type	Description:
setpoint	F1	Analog value that represents the nominal value the block maintains when in Auto mode. Should be scaled in percent.
prp_gain	F2	Analog value that defines proportional action. Should be scaled in percent.
rev_act	L1	A logic 1 on this input causes a decrease in the output value while a 0 produces an increase.
int_tc	F2	Integral time constant; defines the integral action of the block. Expressed in seconds.
meas_in	F2	Analog input that represents the feedback value. Should be scaled in percent.
deriv_tc	F2	Analog constant that defines derivative action of the block. Expressed in seconds.
deriv_db	F2	Deadband value introduced to derivative, that acts to minimize noise. Should be scaled in percent.
bias	F2	Analog constant that is added to the sum of proportional, integral, and derivative values. Should be scaled in percent.
lim_array	F2	Two element array that contains minimum and maximum output limits. The first element is the minimum limit, and the second element is the maximum limit. Both should be scaled in percent.
man_stpt	F2	Analog value that is passed directly to the output when the man_sel input reads 1. Should be scaled in percent.
man_sel	L1	Logic input that determines if the man_stpt value is passed to the output. If this input reads 0, the output will equal the sum of the proportional, integral, and derivative actions, and the bias value.

Outputs

Parameter	Data Type	Description:
output	F2	Analog value that represents either the man_sp value (man_sel 1), or, the sum of the proportional, integral, and derivative actions, and the bias value (man_sel 0). Should be scaled in percent.
int_comp	F2	Double word variable that stores the incremental change and accumulative total of integral action. Should be scaled in percent.
prev_input	F2	3 element array containing the three previous measured values. Should be scaled in percent.

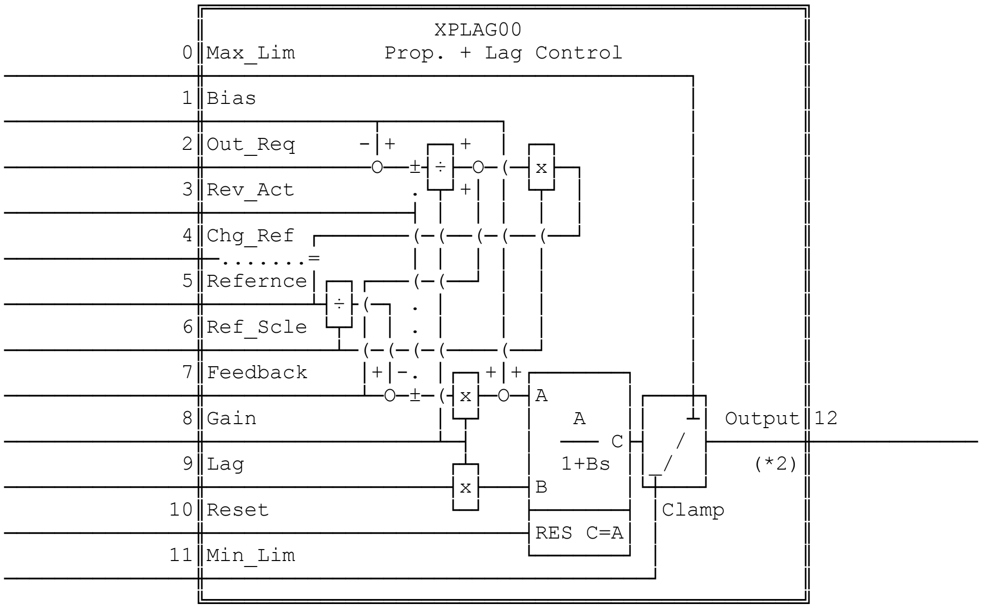
C-6.42. XPLAG00 -- PROPORTIONAL PLUS LAG CONTROL (Ver. 00)

```
                                XPLAG00

max_lim     min_lim
bias
out_req
rev_act    chg_ref                                              output
reference   ref_scale
feedback
gain    lag
reset
```

Description

When the change reference input (chg_ref) reads false, the Proportional Plus Lag (Ver 00) block implements a function that adjusts the analog output to reflect the analog reference value according to the proportional action over a period of time specified by the lag constant. The block begins by reading the analog feedback value and subtracting it from analog reference constant; this operation generates an error value (Note: The scale of the reference input may have to be changed (ref_scle) as feedback and reference values may at times be dissimilar). The error is multiplied by the gain constant to define the block's proportional action. The proportional gain is scaled as a B7 number.

For system tuning purposes, the XPLAG block multiplies the gain by the time (lag) constant. This functionality allows the gain to be adjusted without changing the crossover frequency (gain = 1). Note 1: An extremely high gain value will in effect cause the block to simply execute an integral function. Note 2: A logic 1 reset input will disable the time Lag function of the block. The positive or negative control action of the block is defined by the reverse action (rev_act) input. A logic TRUE on this input causes the sign of the output to reflect the sign of the error value while a FALSE reading will produce a signed output that is the inverse of the error value. When the logic input change reference (chg_ref) is TRUE, the reference value is updated based on the requested output value (out_req), and the output will equal the requested output value (out_req). Note: If the output reaches a value equal to either the max_lim or min_lim constants, it is clamped to that respective value.



Inputs

Parameter	Data Type	Description:
max_lim	F2	Analog constant that determines the highest value the output can be adjusted to.
min_lim	F2	Analog constant that determines the lowest value the output can be adjusted to.
bias	F2	Analog constant added to the error value after it has been multiplied by the gain.
out_req	F2	Analog value that the output will reflect when the chg_rq input reads True.
rev_act	L1	Logical input that determines the positive or negative action of the block. A logic TRUE implements an increasing action while a FALSE signal effects the reverse.
chg_ref	L1	A logic TRUE chg_ref input will cause the reference value to be updated (based on the out_req) and the Output to be altered to equal the out_req value.
refernce	F2	Analog value from which the error and proportional action are derived.
ref_scle	F2	Analog input that alters scale of reference value.
feedback	F2	Analog input that is subtracted from the reference value to generate an error value.
gain	F2	Analog constant that is multiplied by the error value in order to define proportional action of the block.
lag	F2	Analog constant that determines the amount of time it will take the block to implement a new Output.
reset	L1	A logic 1 on this input will cause the lag function to be disabled.

Outputs

Parameter	Data Type	Description:
out	F2	Analog value that is adjusted according to proportional action over a period of time specified by the time lag constant.

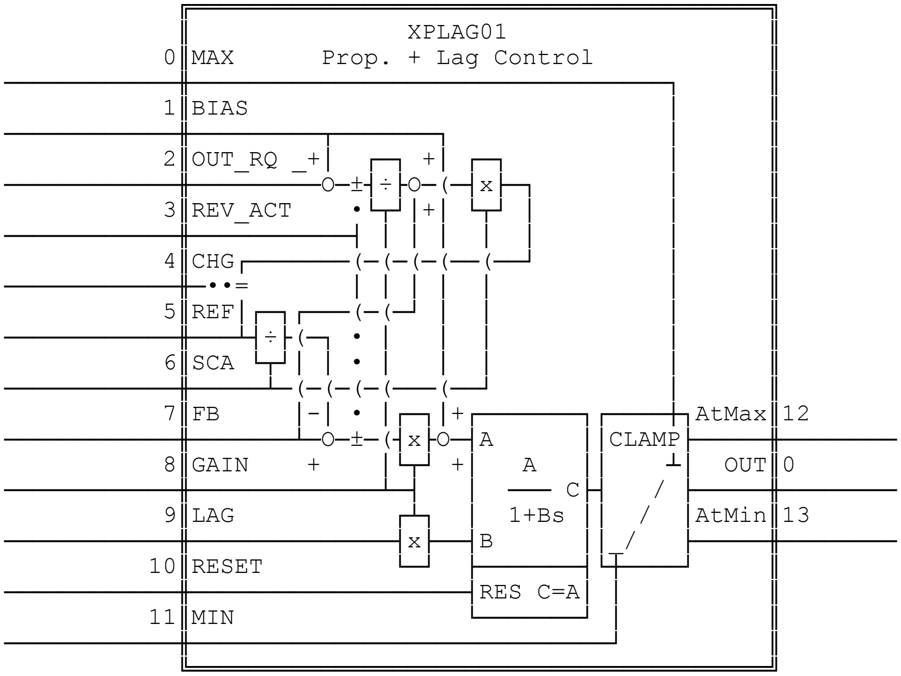
C-6.43. XPLAG01 -- PROPORTIONAL PLUS LAG CONTROL (Ver 01)

```
                              XPLAG01

max_lim     min_lim
bias
out_req                                                 atmax
rev_act    chg_ref                                      output
reference   ref_scale                                   atmin
feedback
gain    lag
reset
```

Description

When the change reference input (chg_ref) reads false, the Proportional Plus Lag (Ver 01) block implements a function that adjusts the analog output to reflect the analog reference value according to the proportional action over a period of time specified by the lag constant. The block begins by reading the analog feedback value and subtracting it from analog reference constant; this operation generates an error value (Note: The scale of the reference input may have to be changed (ref_scle) as feedback and reference values may at times be dissimilar). The error is multiplied by the gain constant to define the block's proportional action. The proportional gain is scaled as a B7 number.

For system tuning purposes, the XPLAG block multiplies the gain by the time (lag) constant. This functionality allows the gain to be adjusted without changing the crossover frequency (gain = 1). Note 1: An extremely high gain value will in effect cause the block to simply execute an integral function. Note 2: A logic 1 reset input will void the time Lag function of the block. The positive or negative control action of the block is defined by the reverse action (rev_act) input. A logic TRUE on this input causes the sign of the output to reflect the sign of the error value while a FALSE reading will produce a signed output that is the inverse of the error value. When the logic input change reference (chg_ref) is TRUE, the reference value is updated based on the requested output value (out_req), and the output will equal the requested output value (out_req). Note: If the output reaches a value equal to either the max_lim or min_lim constants, it is clamped to that respective value; the atmax or atmin logic outputs will signal the clamping status.



Inputs

Parameter	Data Type	Description:
max_lim	F2	Analog constant that determines the highest value the output can be adjusted to.
min_lim	F2	Analog constant that determines the lowest value the output can be adjusted to.
bias	F2	Analog constant added to the error value after it has been multiplied by the gain.
out_req	F2	Analog value that the output will reflect when the chg_rq input reads TRUE.
rev_act	L1	Logical input that determines the positive or negative action of the block. A logic TRUE implements an increasing action while a FALSE signal effects the reverse.
chg_ref	L1	A logic TRUE chg_ref input will cause the reference value to be updated (based on the out_req) and the Output to be altered to equal the out_req value.
feference	F2	Analog value from which the error and proportional action are derived.
ref_scle	F2	Analog input that alters scale of reference value.
feedback		F2 Analog input that is subtracted from the reference value to generate an error value.
gain	F2	Analog constant that is multiplied by the error value in order to define proportional action of the block.
lag	F2	Analog constant that determines the amount of time it will take the block to implement a new Output.
reset	L1	A logic 1 on this input will cause the lag function to be disabled.

Outputs

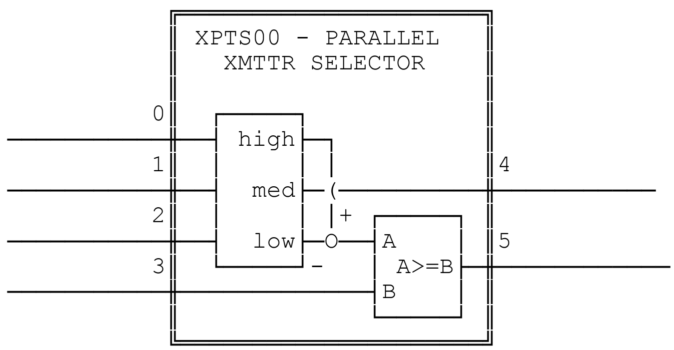
Parameter	Data Type	Description:
at max	L1	Logic ouput that reads TRUE (1) when the ouput (out) is clamped to the maximum value (max).
out	F2	Analog value that is adjusted according to proportional action over a period of time specified by the time lag constant.
at min	L1	Logic ouput that reads TRUE (1) when the ouput (out) is clamped to the minimum value (min).

C-6.44. XPTS00 -- PARALLEL TRANSMITTER SELECTOR

```
                         XPTS00
input_1
input_2                                                 med_out
input_3                                                diff_alm
diff_Lim
```

Description

The Parallel Transmitter block executes a function that selects the median value of three analog signals and passes the result to the output. The block generates a differential alarm if the difference between the maximum and minimum values of the three signals becomes greater than or equal to the analog signal limit (dif_lim).



Inputs

Parameter	Data Type	Description:
input_1	F2	Analog input that is compared to input_2 and input_3 input values in order to calculate a median value output.
input_2	F2	Analog input that is compared to the input_1 and input_3 input values in order to calculate a median value output.
input_3	F2	Analog input that is compared to the input_1 and input_2 inputs in order to calculate a median value output.
diff_lim	F2	Analog constant value that determines when an differential will be generated.

Outputs

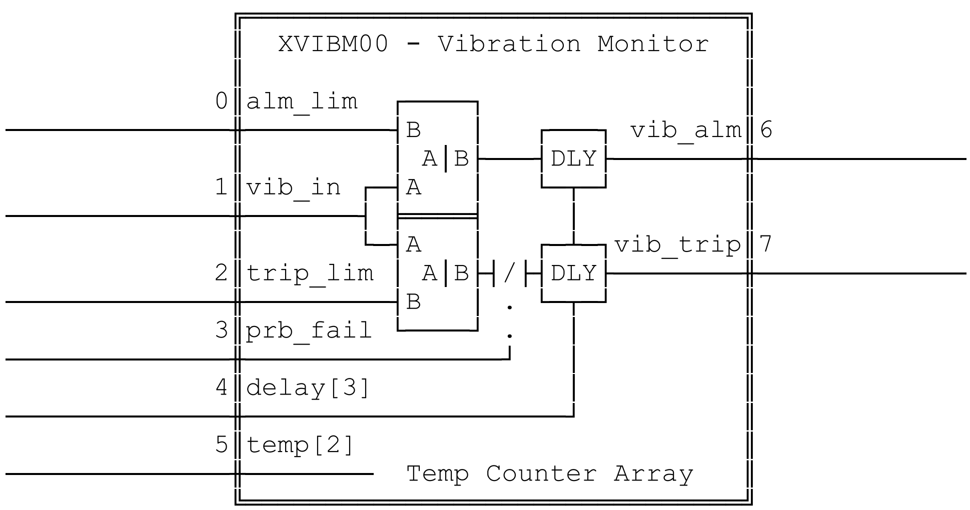
Parameter	Data Type	Description:
diff_alm	L1	Logical output that reads 1 when the difference between the max. and min. of three input values exceeds the differential limit.
med_out	F2	Analog value that represents the median value of hi, med, and low inputs.

C-6.45. XVIBM00 -- VIBRATION MONITOR

```
                              XVIBM00

vib_in                                                      vib_alm
alm_lim  trip_lim
                                                            vib_trip
prb_fail
delay[3]                                                    temp[2]
```

Description

The Vibration Monitor block reads analog vibration (vib_in) and logical probe failure (prb_fail) inputs. The vibration input value is compared to the alarm limit (alm_lim) and trip limit (trip_lim) analog constants to determine whether they equal or exceed them. When the vibration input value equals or exceeds either of these constants for a period of time specified by a respective delay[3] constant value (alarm, trip), a logic 1 is passed to a corresponding (vib_alm, vib_trip) output. The alarm and trip time constants are saved in the temp[2] temporary storage location. These values are replaced by the probe delay constant if the block receives a logic 1 on the probe fail (prb_fail) input. This replacement will continue until the prb_fail input transitions back to 0. If the vib_in value is still above either the alarm or trip limits when the transition occurs, the corresponding counter in memory will begin counting down from the probe delay value. The probe delay constant is normally larger than the alarm or trip delay values and should allow time for the vib_in signal to return to a nominal level. If the vib_in remains equal to or above an alarm or trip limit after the probe delay time period has run down, the corresponding delay constant is reinstated in memory and begins to be counted down. If that time allotment runs out, a logic 1 is passed to the corresponding output. vib_in, alm_lim, and trip_lim inputs must all be scaled similarly.



Inputs

Parameter	Data Type	Description:
vib_in	F2	Analog value that is compared to the alm_lim and trip_lim inputs.
alm_lim	F2	Analog constant that represents the maximum value the vib_in may reach before a logic 1 is passed to the vib_alm output.
trip_lim	F2	Analog constant that represents the maximum value the vib_in may reach before a logic 1 is passed to the trip_lim output.
prb_fail	L1	Logic input that reads 1 when a probe failure occurs.
delay[3]	F2	Analog inputs (alarm, trip, probe fail) that determine the amount of time an alarm state is allowed to exist before corresponding alarm is generated.

Outputs

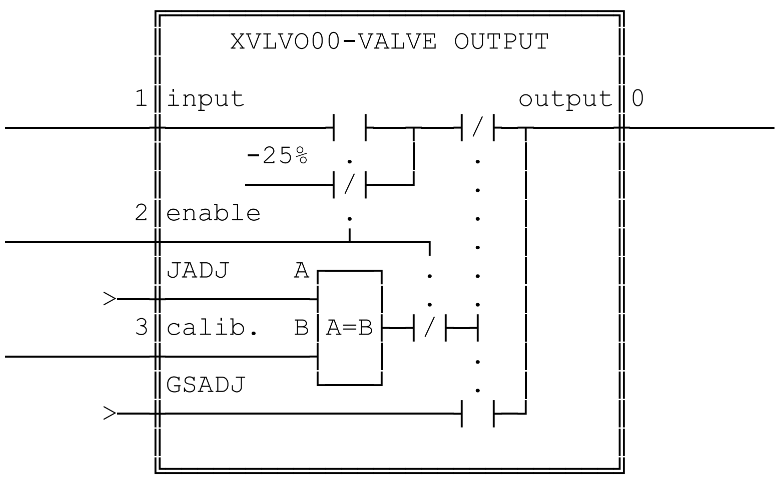
Parameter	Data Type	Description:
temp[2]	F2	Temporary storage location where alarm and trip delay values are saved. If the prb_fail input reads 1, these values are replaced by the prb_fail delay value.
vib_alm	L1	Logic output that reads 1 when the vib_in value exceeds the vibration limit for longer than the period of time specified by the vibration alarm delay constant.
vib_trip	L1	Logic output that reads 1 when the vib_in value exceeds the vibration trip limit for longer than the period of time specified by the Vibration Trip Delay constant.

C-6.46. XVLVO00 -- VALVE OUTPUT BLOCK

XVLVO00	
enable	output
input	
jadj_num	

Description

The XVLV00 Servo Valve Output block executes a function that passes analog valve control information. If the enable input reads 0, the block will direct the valve to maintain a -25% bias. When this input reads one (1) the input reference data is passed directly to the output. A logic 0 enable input will also permit the block to execute a manual output for calibration, if required. Once enabled, the block will output the value GSADJ for calibration, when it receives jadj and calibration inputs that match (A=B). Respectively, these values represent system variables and calibration points that correspond to specific valves. When a match occurs, the gsadj valve adjust input value is passed directly to the output.



Inputs

Parameter	Data Type	Description:
enable	L1	Logic input that allows the block to pass Input information or execute a calibration function.
input	F2	Analog value that represents a valve reference position.
jadj_num	A2	Analog system variable that corresponds to a specific valve.

Outputs

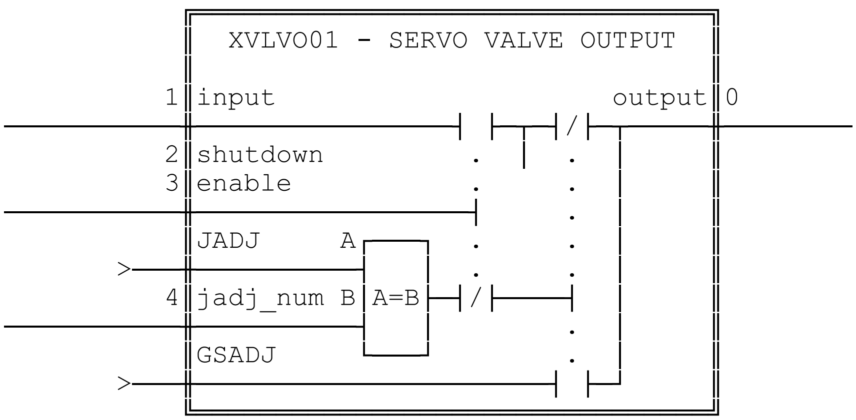
Parameter	Data Type	Description:
output	F2	Analog servo valve reference point command.

C-6.47. XVLVO01 -- VALVE OUTPUT BLOCK

XVLVO01	
enable	output
input	
jadj_num	

Description

The XVLVO01 Servo Valve Output block executes a function that passes analog valve control information. If the enable input reads 0, the block will pass to the valve the shutdown analog value (unlike XVLVO00, this function permits customization of the valve "shutdown" bias). When this input reads 1, input reference data is passed directly to the output. A logic 1 enable input will permit the block to execute a manual output for calibration if required. The block will output the value GSADJ for calibration when it receives jadj and calibration inputs that match (A=B). Respectively, these values represent system variables and calibration points that correspond to specific valves. When a match occurs, the gsadj valve adjust input value is passed directly to the output.



Inputs

Parameter	Data Type	Description:
enable	L1	Logic input that allows the block to pass input information or execute a calibration function.
input	F2	Analog value that represents a valve reference position.
jadj_num	A2	Analog system variable that corresponds to a specific valve.
shutdown	F2	Analog constant value that will be passed to the valve when the enable input reads 0.

Outputs

Parameter	Data Type	Description:
output	F2	Analog servo valve reference point command.

APPENDIX D

SIGNAL FLOW DIAGRAMS

D-1. DEFINITIONS OF ABBREVIATIONS

Boxes on the diagrams in Appendix D contain the following abbreviations.

A/D	Analog to Digital Converter
CL	Current Limit
CTR	Counter
D/A	Digital to Analog Converter
MP	Micro Processor
MUX	Multiplexer, switches multiple inputs to one output
RD	Relay Driver
USART	Universal Synchronous/Asynchronous Receiver/Transmitter
VCD	Voltage Controlled Oscillator

(VCD and CTR combined are the same as an A/D Converter)

D-2. SIGNAL FLOW DIAGRAMS

Milliamp Input Tables

Core, Card, and Location	Generic Signal Name from left column of IO.ASG file	Signal Termination Points			JBB or JHH Pin Numbers		Hardware Jumper
		POS	NEG	EXC			BJC
<C>	C_C_MAI01	37	38	39	1	2	1
Core	C_C_MAI02	41	42	40	3	4	2
CTBA	C_C_MAI03	43	44	45	5	6	3
Card	C_C_MAI04	47	48	46	7	8	4
Location	C_C_MAI05	49	50	51	9	10	5
<C> 6	C_C_MAI06	53	54	52	11	12	6
	C_C_MAI07	55	56	57	13	14	7
	C_C_MAI08	59	60	58	15	16	8
	C_C_MAI09	61	62	63	17	18	9
	C_C_MAI10	65	66	64	19	20	10
	C_C_MAI11	67	68	69	21	22	11
	C_C_MAI12	71	72	70	23	24	12
	C_C_MAI13	73	74	75	25	26	13
	C_C_MAI14	77	78	76	27	28	14
<C>	C_C_MAI15	1	2	3	1	2	1
Core	C_C_MAI16	5	6	4	3	4	2
TBCB	C_C_MAI17	7	8	9	5	6	3
Card	C_C_MAI18	11	12	10	7	8	4
Location	C_C_MAI19	13	14	15	9	10	5
<C>	C_C_MAI20	17	18	16	11	12	6
7optional	C_C_MAI21	19	20	21	13	14	7
card	C_C_MAI22	23	24	22	15	16	8
	C_C_MAI23	25	26	27	17	18	9
	C_C_MAI24	29	30	28	19	20	10
	C_C_MAI25	31	32	33	21	22	11
	C_C_MAI26	35	36	34	23	24	12
	C_C_MAI27	37	38	39	25	26	13
	C_C_MAI28	41	42	40	27	28	14

Core, Card, and Location	Generic Signal Name from left column of IO.ASG file	Signal Termination Points			JHH Pin Numbers		Hardware Jumpers	
		POS	NEG	EXC			BJC	BJL
<C>	C_C_MAI29	43	44	45	29	30	15	23
Core	C_C_MAI30	47	48	46	31	32	16	24
TBCB	C_C_MAI31	49	50	51	33	34	17	25
	C_C_MAI32	53	54	52	35	36	18	26
Location	C_C_MAI33	55	56	57	37	38	19	27
<C> 7	C_C_MAI34	59	60	58	39	40	20	28
optional	C_C_MAI35	61	62	63	41	42	21	29
card	C_C_MAI36	65	66	64	43	44	22	30

Core, Card, and Location	Generic Signal Name from left column of IO.ASG file	Signal Termination Points			JB Pin Numbers		Hardware Jumpers BJC
		POS	NEG	EXC			
<Q>	Q_R_MAI01	35	36	37	1	2	1
Core	Q_R_MAI02	39	40	38	3	4	2
	Q_R_MAI03	41	42	43	5	6	3
TBQC	Q_R_MAI04	45	46	44	7	8	4
or	Q_R_MAI05	47	48	49	9	10	5
TBQF	Q_R_MAI06	51	52	50	11	12	6
Card	Q_R_MAI07	53	54	55	13	14	7
	Q_R_MAI08	57	58	56	15	16	8
Location	Q_R_MAI09	59	60	61	17	18	9
<R> 9	Q_R_MAI10	63	64	62	19	20	10
	Q_R_MAI11	65	66	67	21	22	11
	Q_R_MAI12	69	70	68	23	24	12
	Q_R_MAI13	71	72	73	25	26	13
	Q_R_MAI14	75	76	74	27	28	14
	Q_R_MAI15	77	78	79	29	30	15

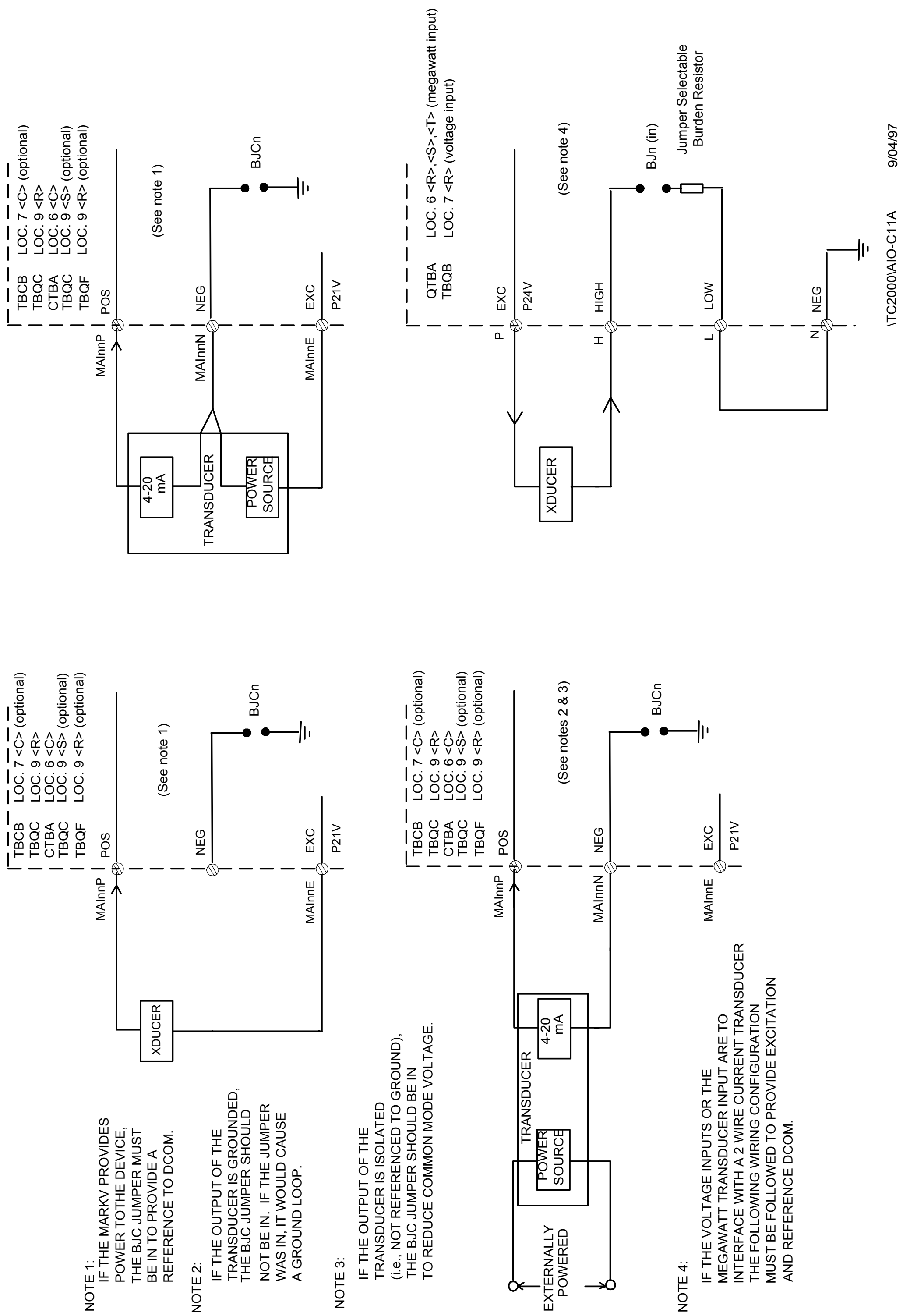


Figure D-1. mA Connection Examples and Notes
(See Tables on previous pages for connection points and hardware jumpers.)

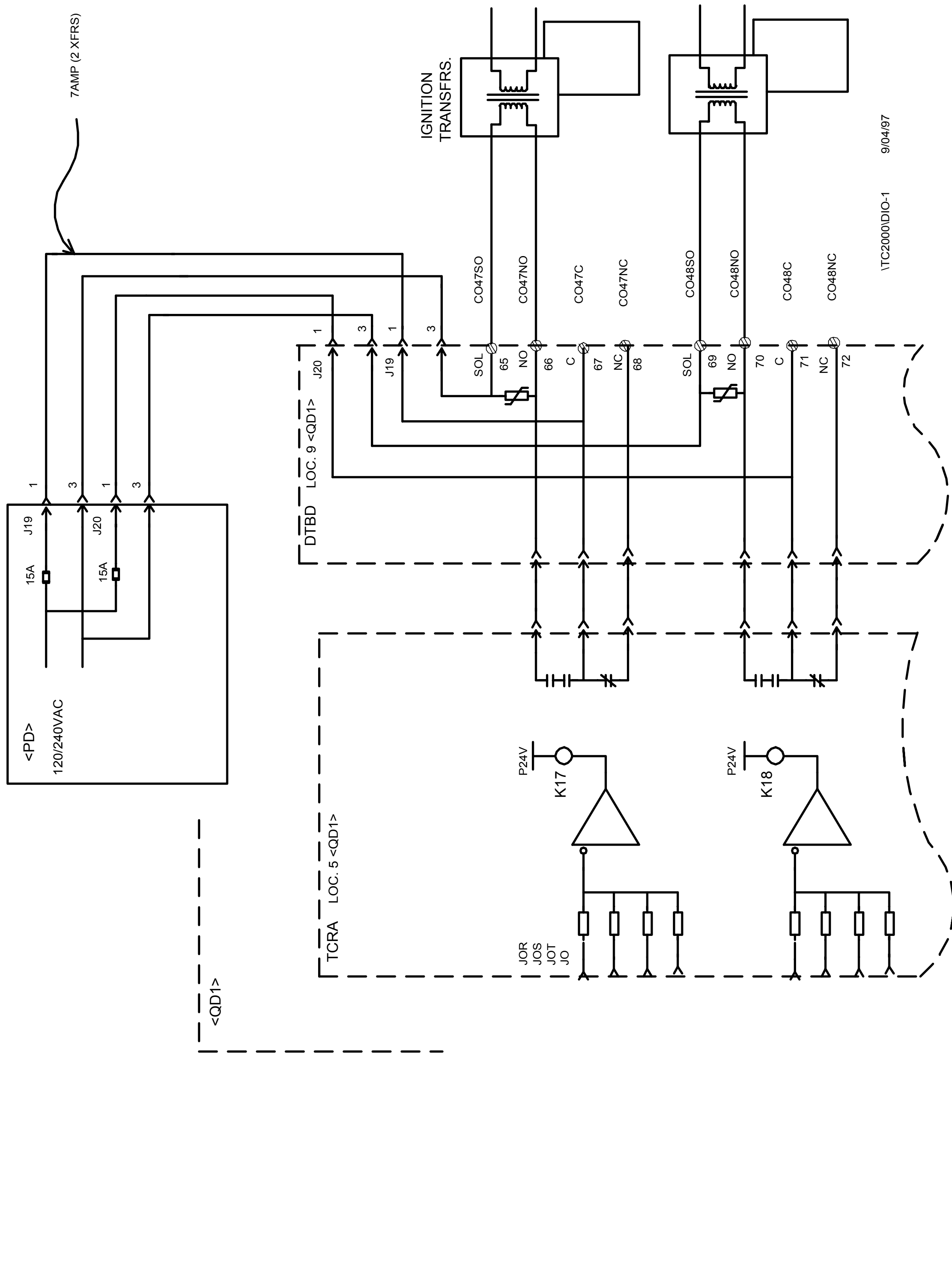
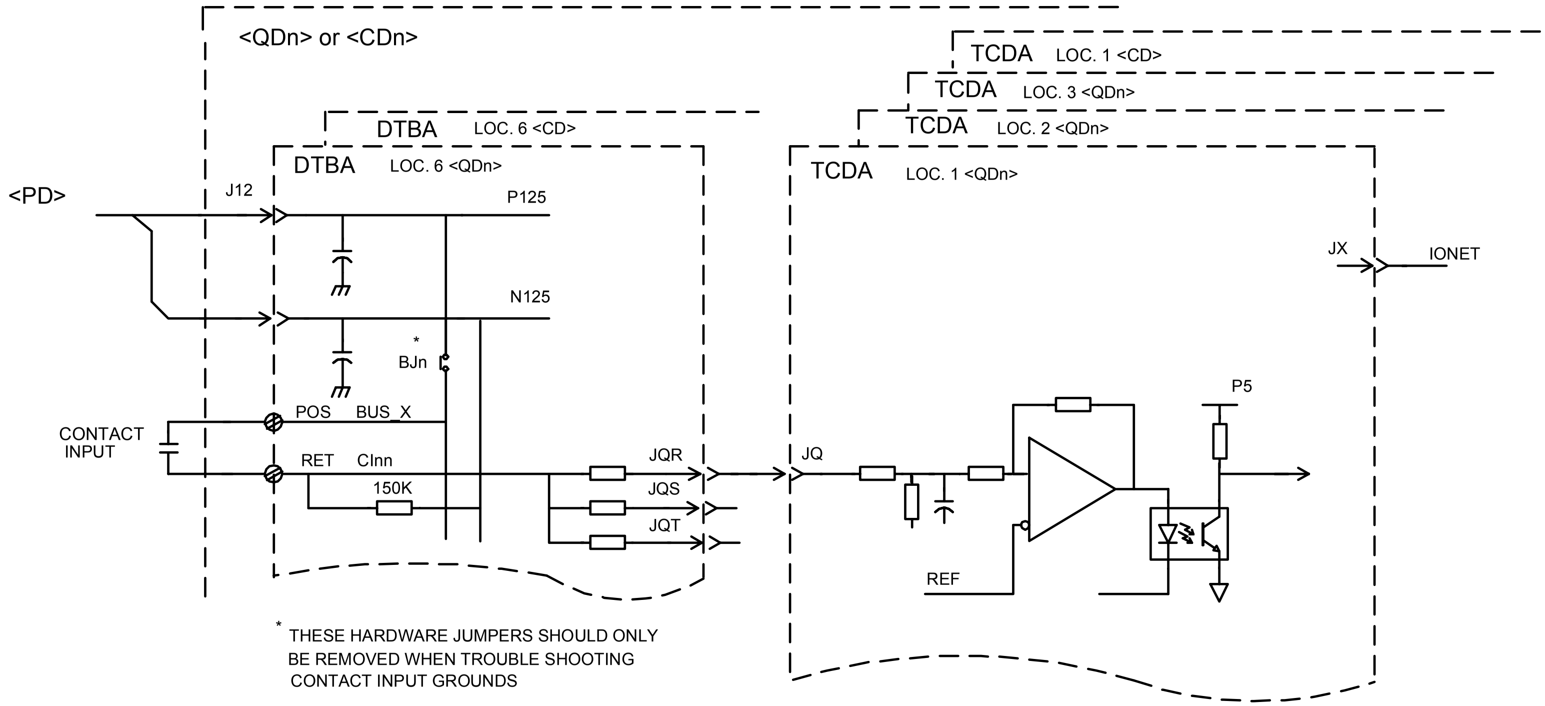


Figure D-2. <QDn> Core - Ignition Transformers on DTBD Terminal Board



* THESE HARDWARE JUMPERS SHOULD ONLY BE REMOVED WHEN TROUBLE SHOOTING CONTACT INPUT GROUNDS

CONTACT #	DTBA SCREW	
	POS	RET
CI0 1	2	1
CI0 2	4	3
CI0 3	6	5
CI0 4	8	7
CI0 5	10	9
CI0 6	12	11
CI0 7	14	13
CI0 8	16	15
CI0 9	18	17
CI0 10	20	19
CI0 11	22	21
CI0 12	24	23
CI0 13	26	25
CI0 14	28	27
CI0 15	30	29
CI0 16	32	31
CI0 17	34	33
CI0 18	36	35
CI0 19	38	37
CI0 20	40	39

CONTACT #	DTBA SCREW	
	POS	RET
CI0 21	42	41
CI0 22	44	43
CI0 23	46	45
CI0 24	48	47
CI0 25	50	49
CI0 26	52	51
CI0 27	54	53
CI0 28	56	55
CI0 29	58	57
CI0 30	60	59
CI0 31	62	61
CI0 32	64	63
CI0 33	66	65
CI0 34	68	67
CI0 35	70	69
CI0 36	72	71
CI0 37	74	73
CI0 38	76	75
CI0 39	78	77
CI0 40	80	79

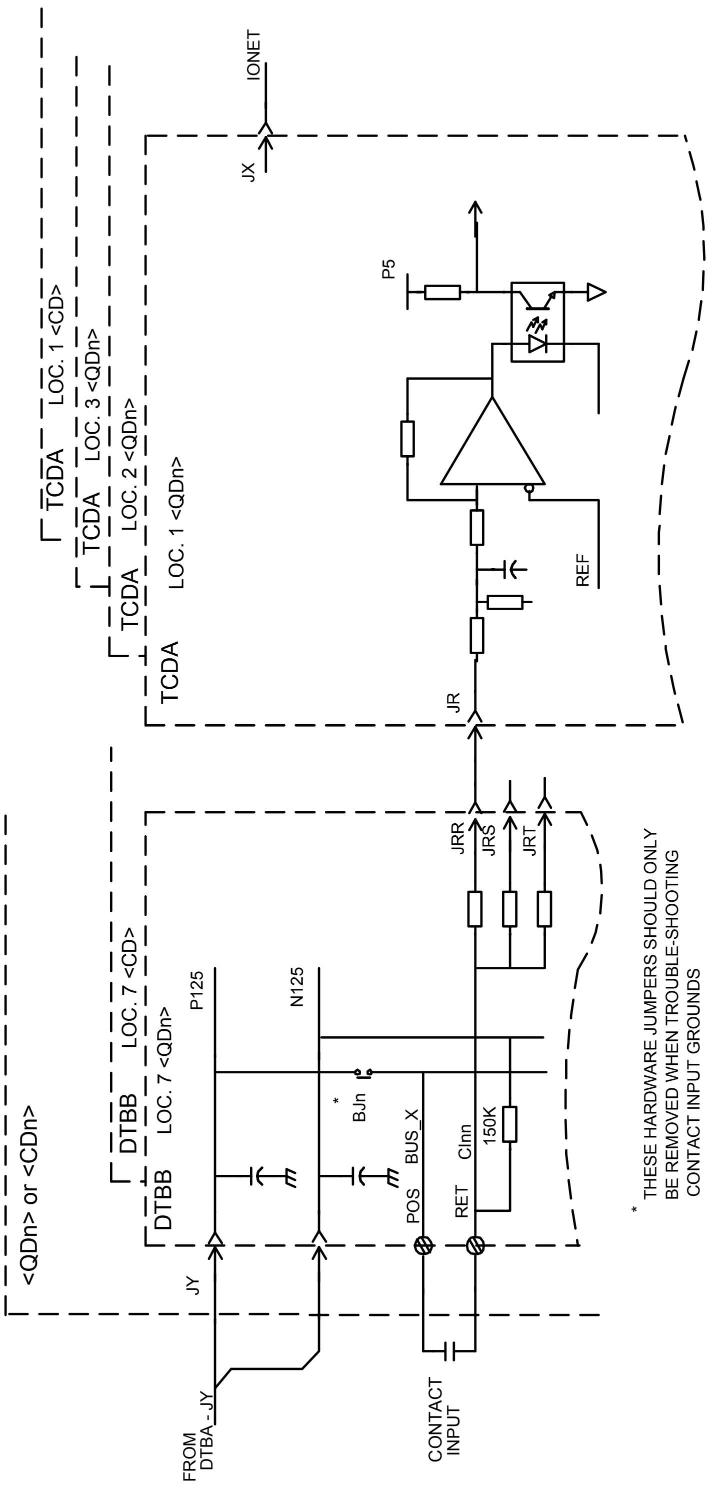
CONTACT #	DTBA SCREW	
	POS	RET
CI0 41	82	81
CI0 42	84	83
CI0 43	86	85
CI0 44	88	87
CI0 45	90	89
CI0 46	92	91

HARDWARE JUMPER ISOLATION TABLE

BJn *	INPUTS
BJ1	#1 to #9
BJ2	#10 to #18
BJ3	#19 to #27
BJ4	#28 to #36
BJ5	#37 to #46

\TC2000\DIO-2 8/31/97

Figure D-3. <QDn> or <CD> Core - Digital Inputs on DTBA Terminal Board



CONTACT #	DTBB SCREW POS	DTBB SCREW RET
CI0 47	2	1
CI0 48	4	3
CI0 49	6	5
CI0 50	8	7
CI0 51	10	9
CI0 52	12	11
CI0 53	14	13
CI0 54	16	15
CI0 55	18	17
CI0 56	20	19
CI0 57	22	21
CI0 58	24	23
CI0 59	26	25
CI0 60	28	27
CI0 61	30	29
CI0 62	32	31
CI0 63	34	33
CI0 64	36	35
CI0 65	38	37
CI0 66	40	39

CONTACT #	DTBB SCREW POS	DTBB SCREW RET
CI0 67	42	41
CI0 68	44	43
CI0 69	46	45
CI0 70	48	47
CI0 71	50	49
CI0 72	52	51
CI0 73	54	53
CI0 74	56	55
CI0 75	58	57
CI0 76	60	59
CI0 77	62	61
CI0 78	64	63
CI0 79	66	65
CI0 80	68	67
CI0 81	70	69
CI0 82	72	71
CI0 83	74	73
CI0 84	76	75
CI0 85	78	77
CI0 86	80	79

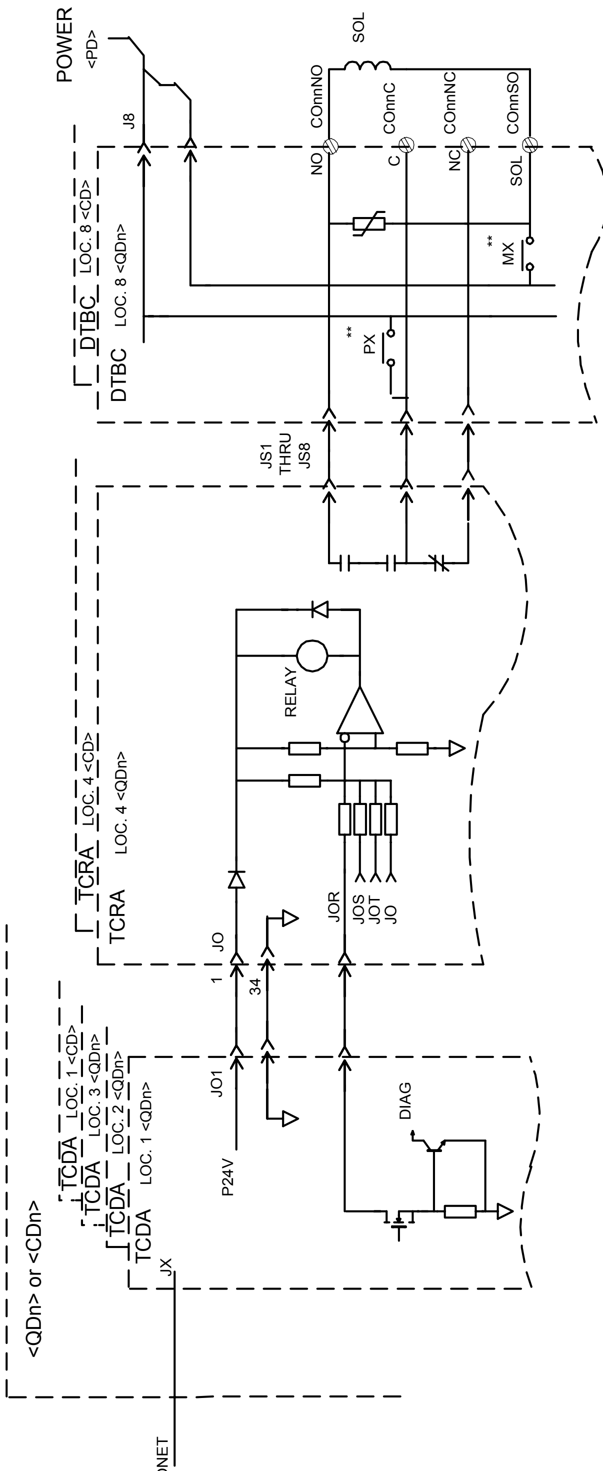
CONTACT #	DTBB SCREW POS	DTBB SCREW RET
CI0 87	82	81
CI0 88	84	83
CI0 89	86	85
CI0 90	88	87
CI0 91	90	89
CI0 92	92	91
CI0 93	94	93
CI0 94	96	95
CI0 95	98	97
CI0 96	100	99

HARDWARE JUMPER ISOLATION TABLE

BJn *	INPUTS
BJ1	#47 to #56
BJ2	#57 to #66
BJ3	#67 to #76
BJ4	#77 to #86
BJ5	#87 to #96

\TC2000 \DIO-3 8/31/97

Figure D-4. <QDn> or <CD> Core - Digital Inputs on DTBB Terminal Board



OUTPUT #	DTBC TERM SCREW SOL	NO	C	NC	ISOLATION JUMPERS PX**	MX**
CO0 1	1	2	3	4	P1	M1
CO0 2	5	6	7	8	P2	M2
CO0 3	9	10	11	12	P3	M3
CO0 4	13	14	15	16	P4	M4
CO0 5	17	18	19	20	P5	M5
CO0 6	21	22	23	24	P6	M6
CO0 7	25	26	27	28	P7	M7
CO0 8	29	30	31	32	P8	M8
CO0 9	33	34	35	36	P9	M9
CO0 10	37	38	39	40	P10	M10
CO0 11	41	42	43	44	P11	M11
CO0 12	45	46	47	48	P12	M12
CO0 13	49	50	51	52	P13	M13
CO0 14	53	54	55	56	P14	M14
CO0 15	57	58	59	60	P15	M15

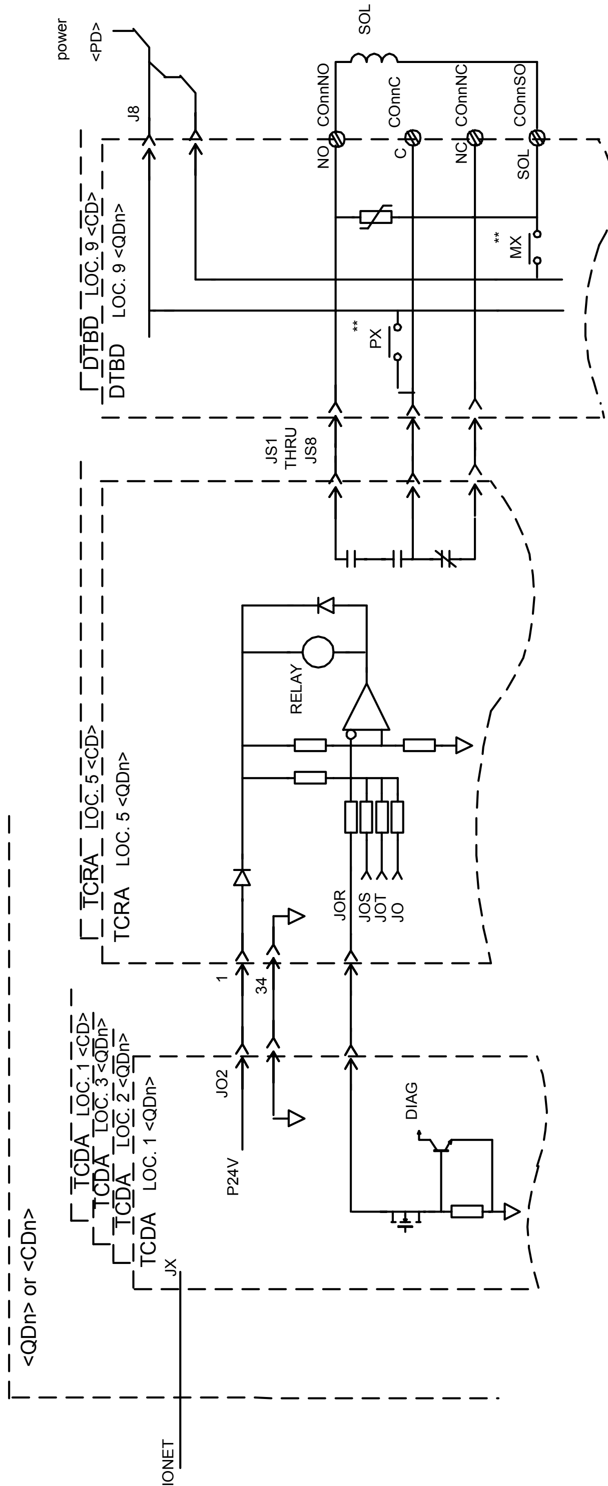
OUTPUT #	DTBC TERM SCREW SOL	NO	C	NC	ISOLATION JUMPERS PX	MX
CO0 16*	61	62	63	64	P16	M16
CO0 17*	65	66	67	68	P17	M17
CO0 18*	69	70	71	72	P18	M18
CO0 19	-	73	74	75	-	-
CO0 20	-	76	77	78	-	-
CO0 21	-	79	80	81	-	-
CO0 22	-	82	83	84	-	-
CO0 23	-	85	86	87	-	-
CO0 24	-	88	89	90	-	-
CO0 25	-	91	92	93	-	-
CO0 26	-	94	95	96	-	-
CO0 27	-	97	98	99	-	-
CO0 28	-	100	101	102	-	-
CO0 29	-	103	104	105	-	-
CO0 30	-	106	107	108	-	-

** These hardware jumper settings are unit specific and should only be in if the Mark V is to supply power to a solenoid on the specific contact output

\TC2000\DIO-4 12\17/96

* THESE ARE SPECIAL SOLENOID CIRCUITS IN <QD1> - THEY INCLUDE J15, J16 FOR EXTERNAL COMPONENTS (SEE APPENDIX E)

Figure D-5. <QDn> or <CD> Core - Relay Outputs on DTBC Terminal Board



OUTPUT #	DTBD TERM SCREW SOL	NO	C	NC	ISOLATION JUMPERS PX**	MX**
CO031	1	2	3	4	P1	M1
CO032	5	6	7	8	P2	M2
CO033	9	10	11	12	P3	M3
CO034	13	14	15	16	P4	M4
CO035	17	18	19	20	P5	M5
CO036	21	22	23	24	P6	M6
CO037	25	26	27	28	P7	M7
CO038	29	30	31	32	P8	M8
CO039	33	34	35	36	P9	M9
CO040	37	38	39	40	P10	M10
CO041	41	42	43	44	P11	M11
CO042	45	46	47	48	P12	M12
CO043	49	50	51	52	P13	M13
CO044	53	54	55	56	P14	M14
CO045	57	58	59	60	P15	M15

OUTPUT #	DTBD TERM SCREW SOL	NO	C	NC	ISOLATION JUMPERS PX	MX
CO046	61	62	63	64	P16	M16
CO047 *	65	66	67	68	-	-
CO048 *	69	70	71	72	-	-
CO049	-	73	74	75	-	-
CO050	-	76	77	78	-	-
CO051	-	79	80	81	-	-
CO052	-	82	83	84	-	-
CO053	-	85	86	87	-	-
CO054	-	88	89	90	-	-
CO055	-	91	92	93	-	-
CO056	-	94	95	96	-	-
CO057	-	97	98	99	-	-
CO058	-	100	101	102	-	-
CO059	-	103	104	105	-	-
CO060	-	106	107	108	-	-

** These hardware jumper settings are unit specific and should only be in if the Mark V is to supply power to a solenoid on the specific contact output

\TC2000\DIO-5 12/17/96

* THESE ARE SPECIAL CIRCUITS -- THEY ARE POWERED WITH VOLTAGE FROM J19, J20 (REFERENCE DRAWING WITH IGNITION TRANSFORMERS)

Figure D-6. <QDn> or <CD> Core - Relay Outputs on DTBD Terminal Board

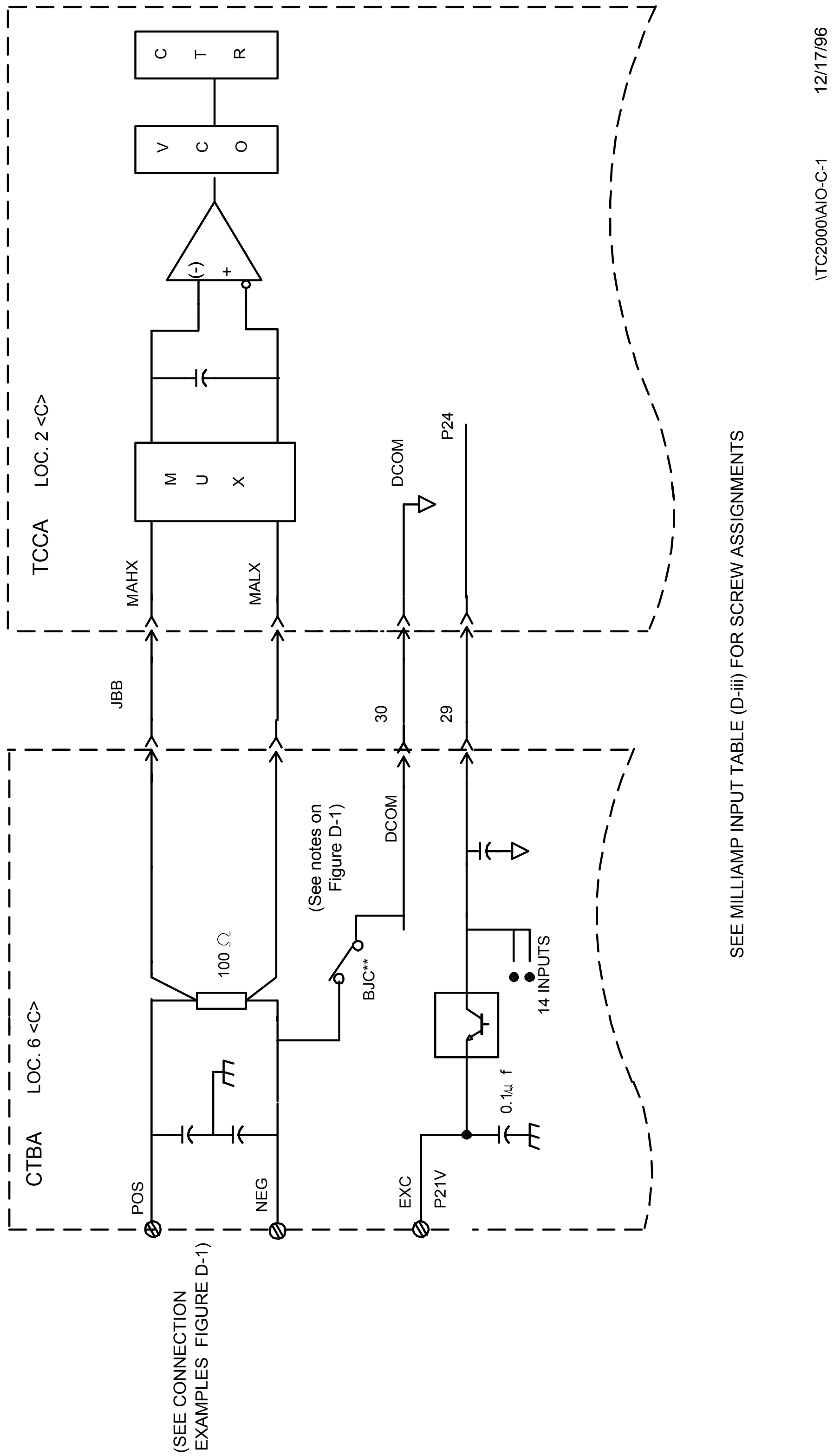


Figure D-7. <C> Core - 4-20 mA Inputs on CTBA Terminal Board

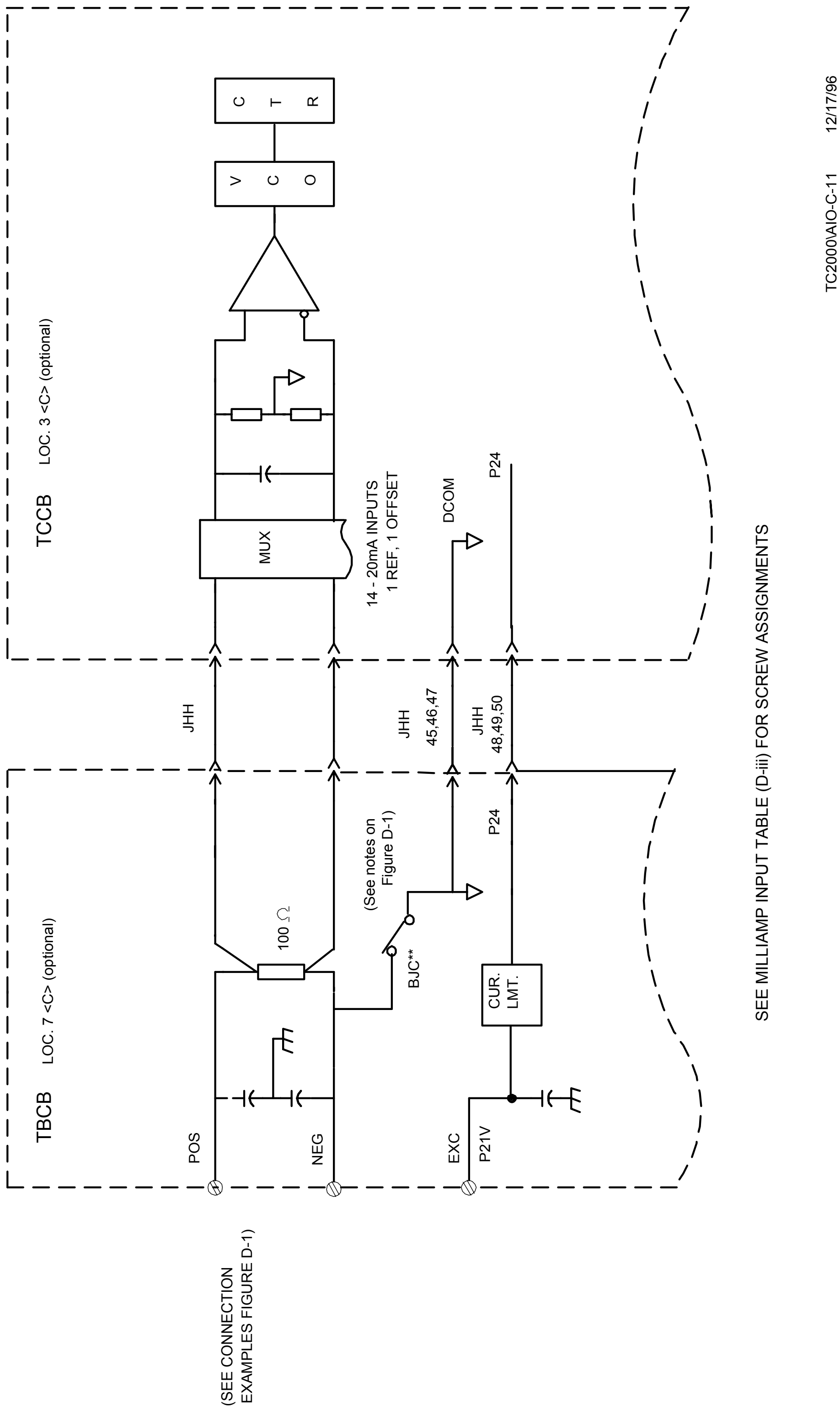


Figure D-8. <C> Core - 4-20 mA Inputs on TBCB Terminal Board

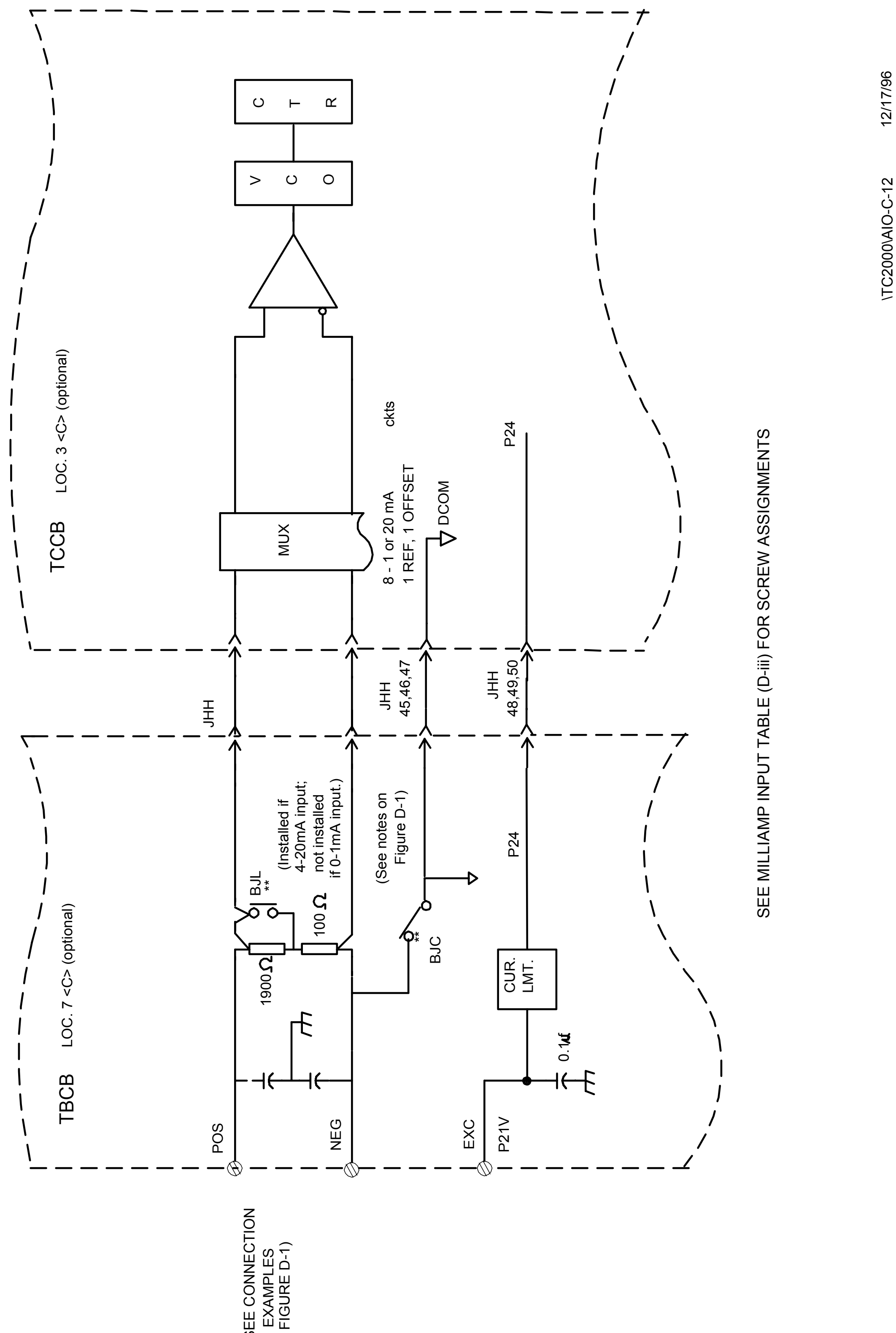
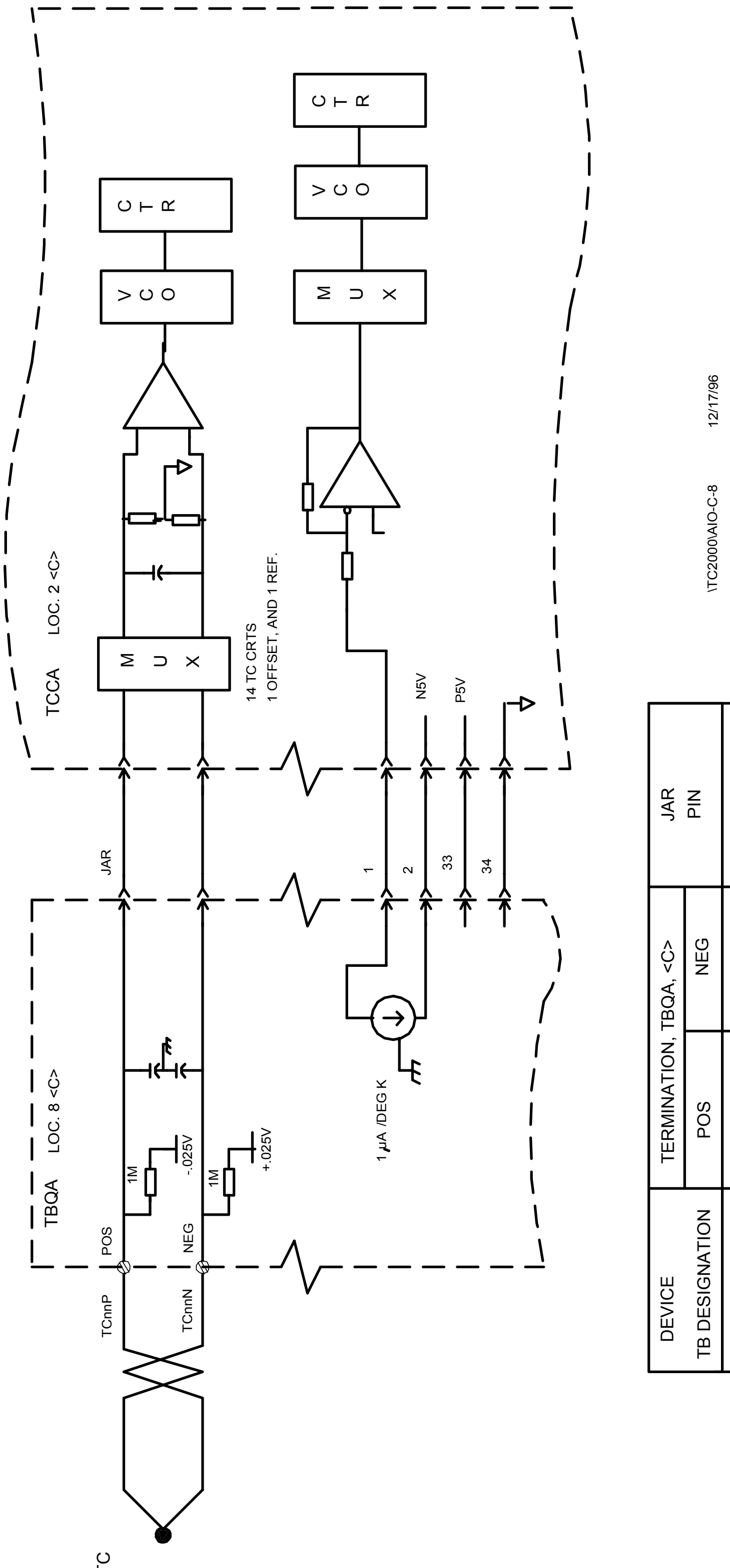
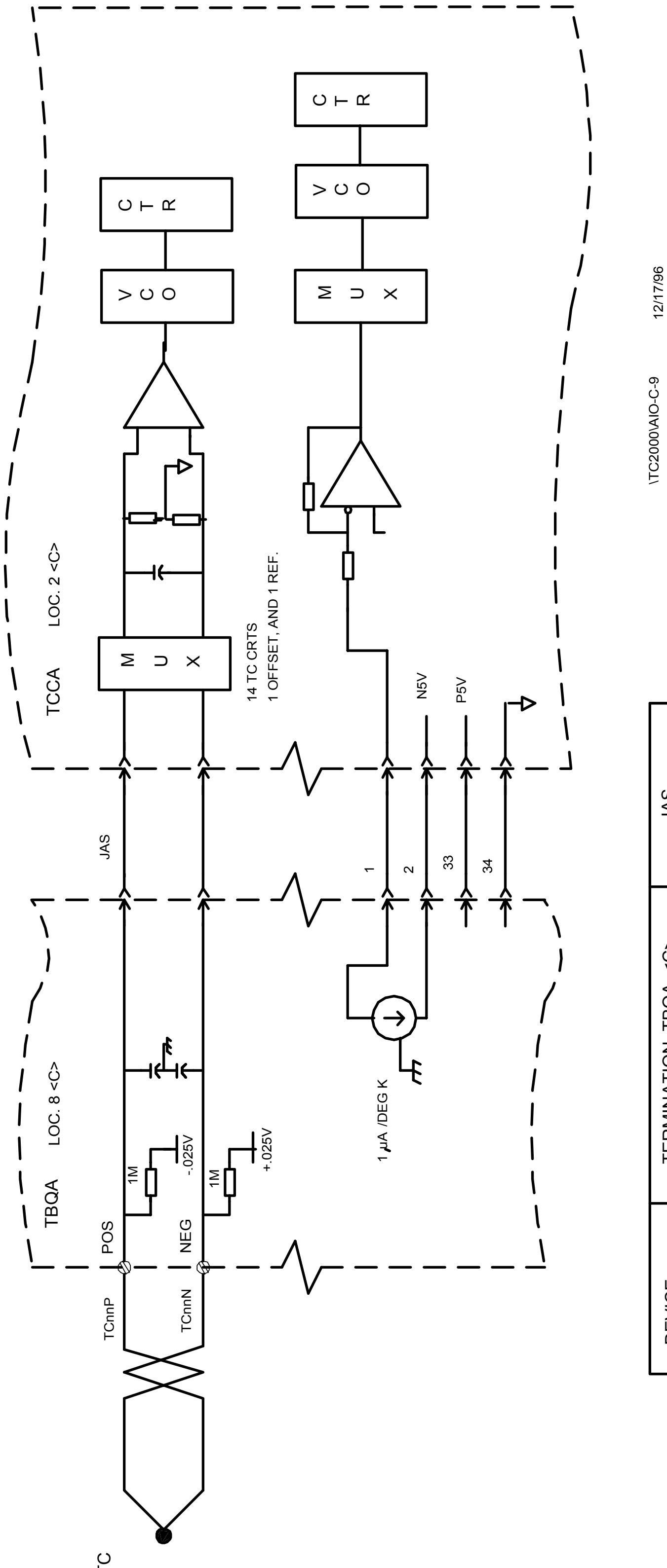


Figure D-9. <C> Core - 0-1 & 4-20 mA Inputs on TBCB Terminal Board



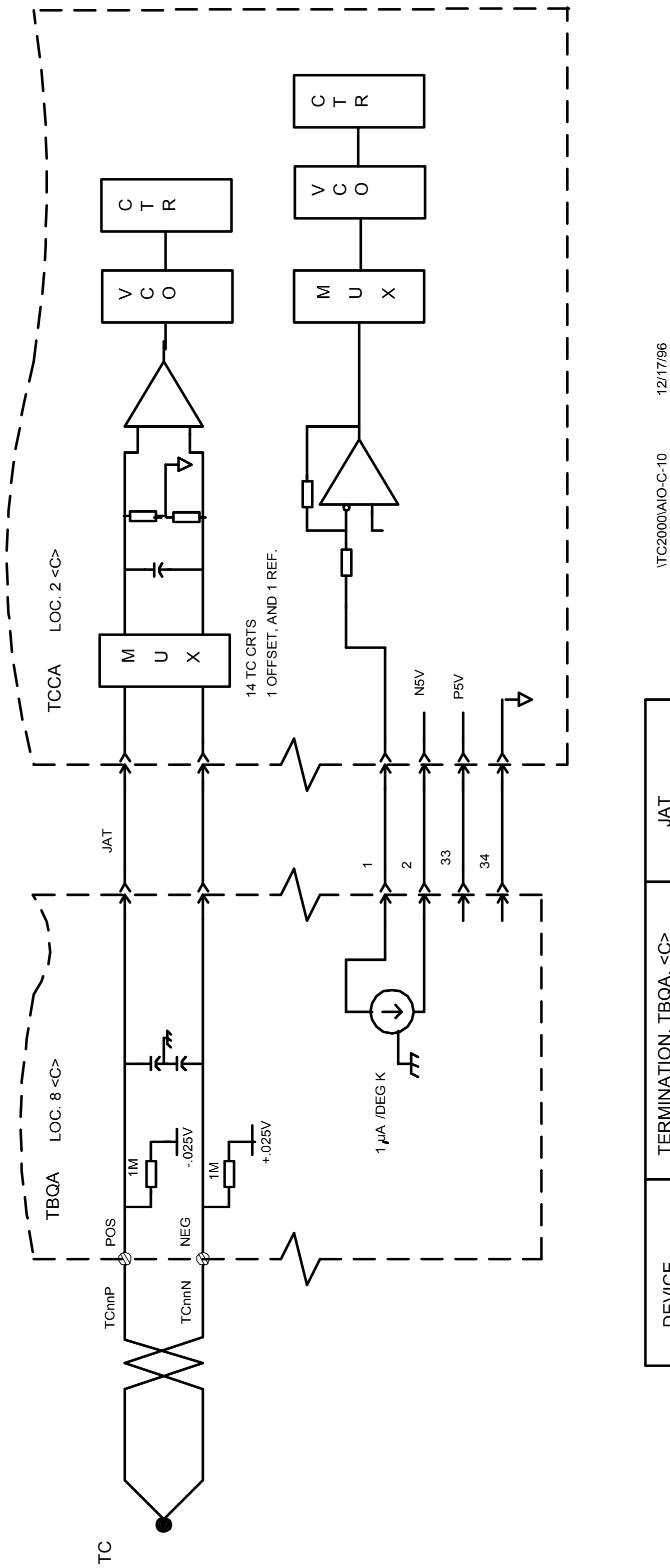
DEVICE TB DESIGNATION	TERMINATION, TBQA, <C> POS	NEG	JAR PIN
TC01	1	2	3,4
TC02	3	4	5,6
TC03	5	6	7,8
TC04	7	8	9,10
TC05	9	10	11,12
TC06	11	12	13,14
TC07	13	14	15,16
TC08	15	16	17,18
TC09	17	18	19,20
TC10	19	20	21,22
TC11	21	22	23,24
TC12	23	24	25,26
TC13	25	26	27,28
TC14	27	28	29,30
TC15	29	30	31,32

Figure D-10. <C> Core - Thermocouple Inputs on TBQA Terminal Board



DEVICE TB DESIGNATION	TERMINATION, TBQA, <C> POS	NEG	JAS PIN
TC16	31	32	3,4
TC17	33	34	5,6
TC18	35	36	7,8
TC19	37	38	9,10
TC20	39	40	11,12
TC21	41	42	13,14
TC22	43	44	15,16
TC23	45	46	17,18
TC24	47	48	19,20
TC25	49	50	21,22
TC26	51	52	23,24
TC27	53	54	25,26
TC28	55	56	27,28
TC29	57	58	29,30
TC30	59	60	31,32

Figure D-11. <C> Core - Thermocouple Inputs on TBQA Terminal Board - Continued



DEVICE TB DESIGNAITON	TERMINATION, TBQA, <C> POS	NEG	JAT PIN
TC31	61	62	3,4
TC32	63	64	5,6
TC33	65	66	7,8
TC34	67	68	9,10
TC35	69	70	11,12
TC36	71	72	13,14
TC37	73	74	15,16
TC38	75	76	17,18
TC39	77	78	19,20
TC40	79	80	21,22
TC41	81	82	23,24
TC42	83	84	25,26

Figure D-12. <C> Core - Thermocouple Inputs on TBQA Terminal Board - Continued

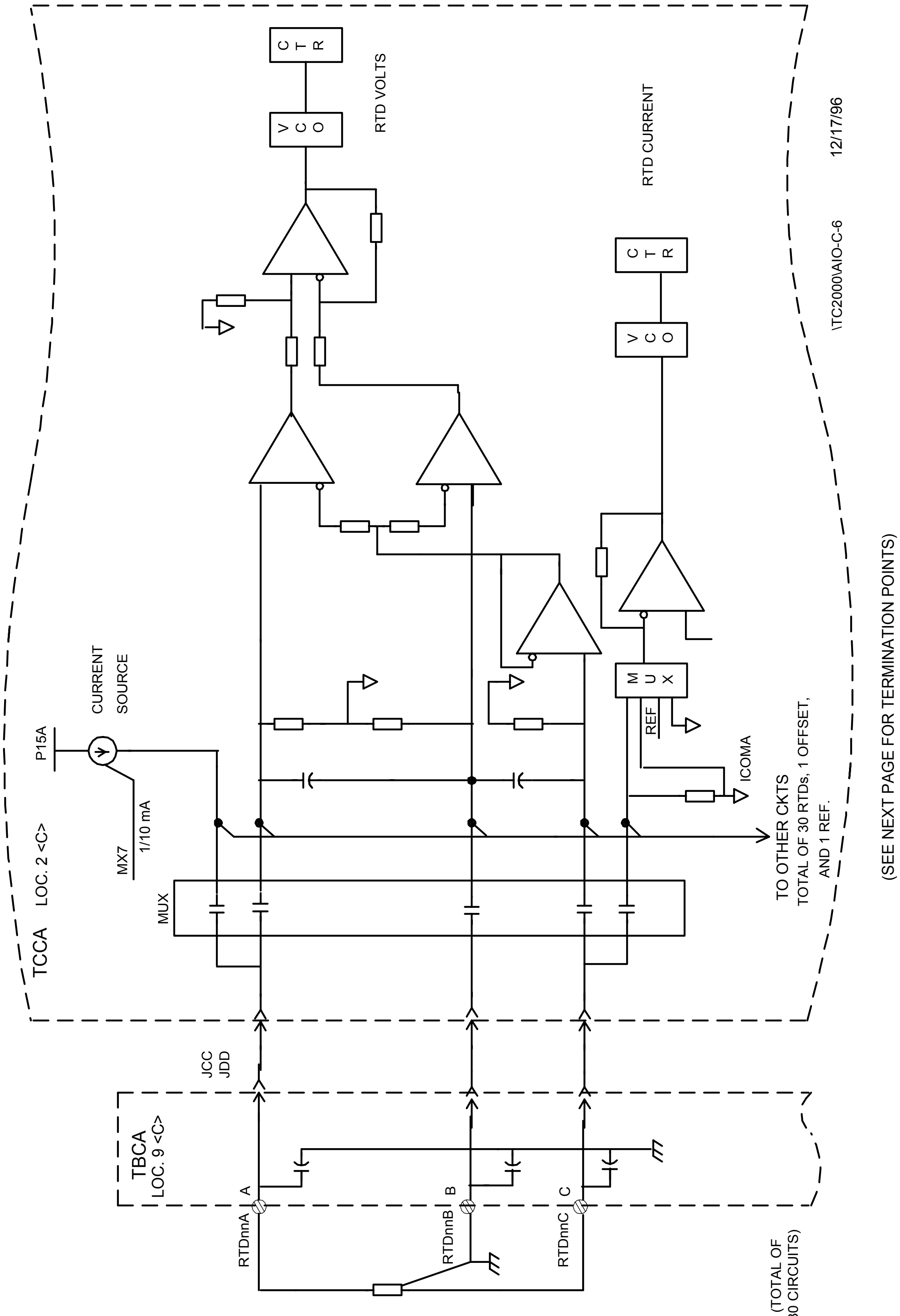
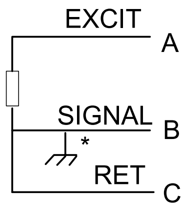


Figure D-13. <C> Core - RTD Inputs on TBCA Terminal Board



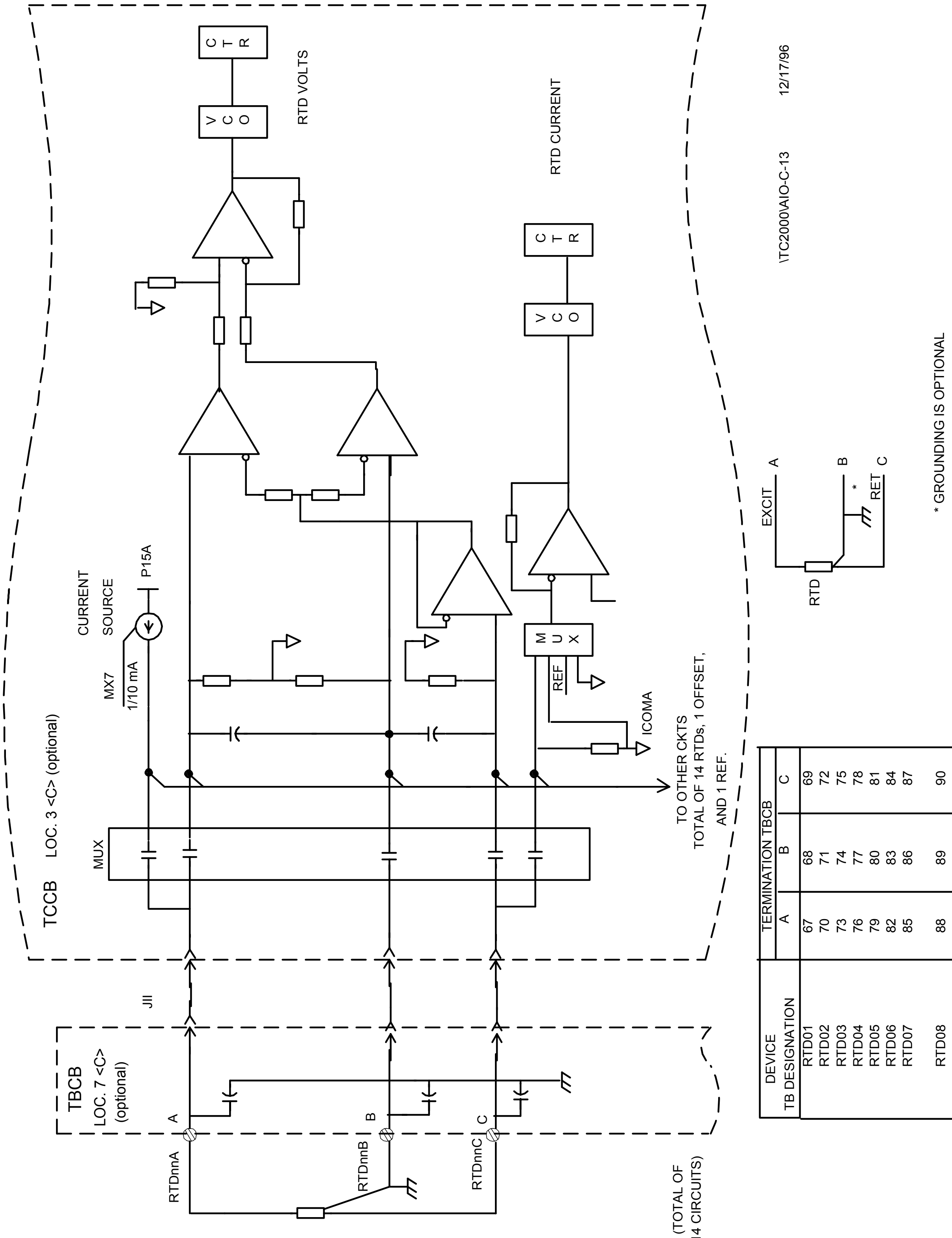
* GROUNDING IS OPTIONAL

DEVICE TB DESIGNATION	TERMINATION TBCA		
	A	B	C
RTD01	1	2	3
RTD02	4	5	6
RTD03	7	8	9
RTD04	10	11	12
RTD05	13	14	15
RTD06	16	17	18
RTD07	19	20	21
RTD08	22	23	24
RTD09	25	26	27
RTD10	28	29	30
RTD11	31	32	33
RTD12	34	35	36
RTD13	37	38	39
RTD14	40	41	42
RTD15	43	44	45
RTD16	46	47	48
RTD17	49	50	51
RTD18	52	53	54
RTD19	55	56	57
RTD20	58	59	60
RTD21	61	62	63
RTD22	64	65	66
RTD23	67	68	69
RTD24	70	71	72
RTD25	73	74	75
RTD26	76	77	78
RTD27	79	80	81
RTD28	82	83	84
RTD29	85	86	87
RTD30	88	89	90

\TC2000\AIO-C-7 01/07/97

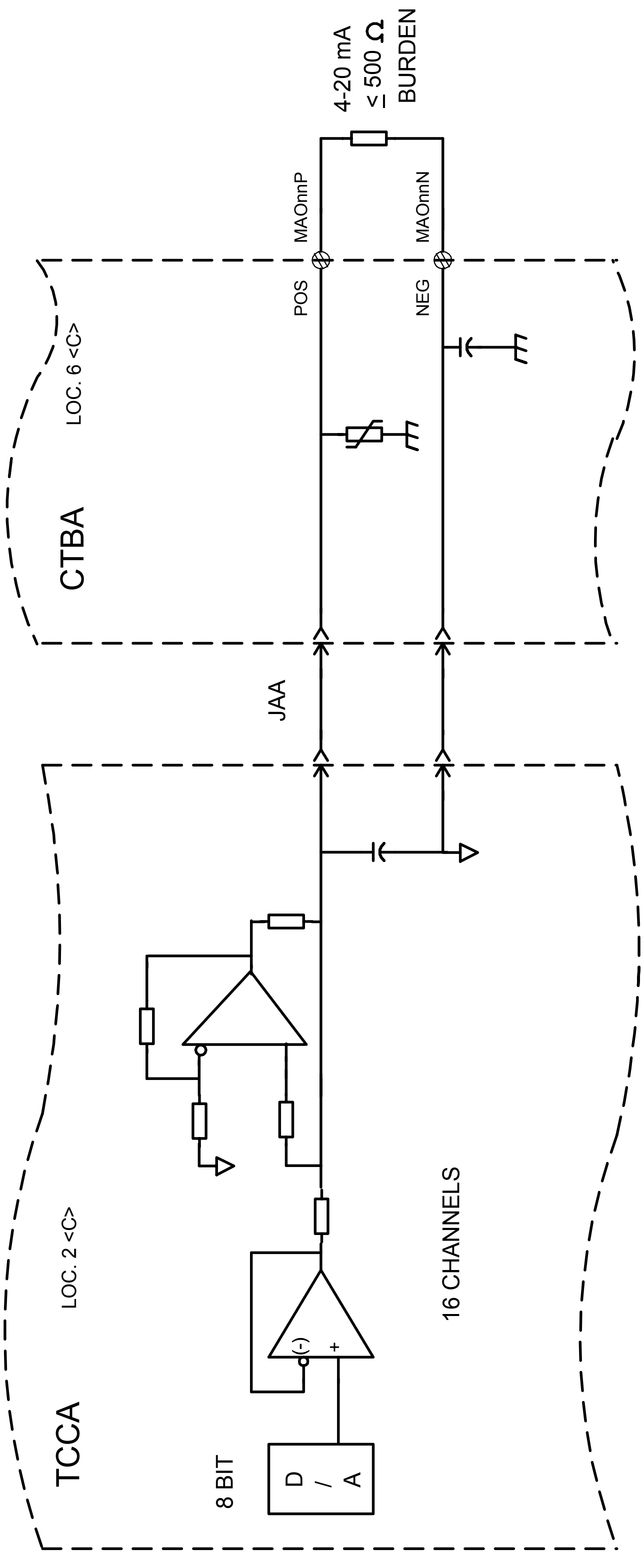
(SEE PREVIOUS PAGE FOR DIAGRAM)

Figure D-14. <C> Core - RTD Inputs on TBCA Terminal Board - Continued



DEVICE TB DESIGNATION	TERMINATION TBCB A	B	C
RTD01	67	68	69
RTD02	70	71	72
RTD03	73	74	75
RTD04	76	77	78
RTD05	79	80	81
RTD06	82	83	84
RTD07	85	86	87
RTD08	88	89	90
RTD09	91	92	93
RTD10	94	95	96
RTD11	97	98	99
RTD12	100	101	102
RTD13	103	104	105
RTD14	106	107	108

Figure D-15. <C> Core - RTD Inputs on TBCB Terminal Board



DEVICE TB DESIGNATION	SIGNAL TERMINATION POS	NEG
MAO01	1	2
MAO02	3	4
MAO03	5	6
MAO04	7	8
MAO05	9	10
MAO06	11	12
MAO07	13	14
MAO08	15	16
MAO09	17	18
MAO10	19	20
MAO11	21	22
AMO12	23	24
MAO13	25	26
MAO14	27	28
MAO15	29	30
MAO16	31	32

Figure D-16. <C> Core - 4-20 mA Outputs on CTBA Terminal Board

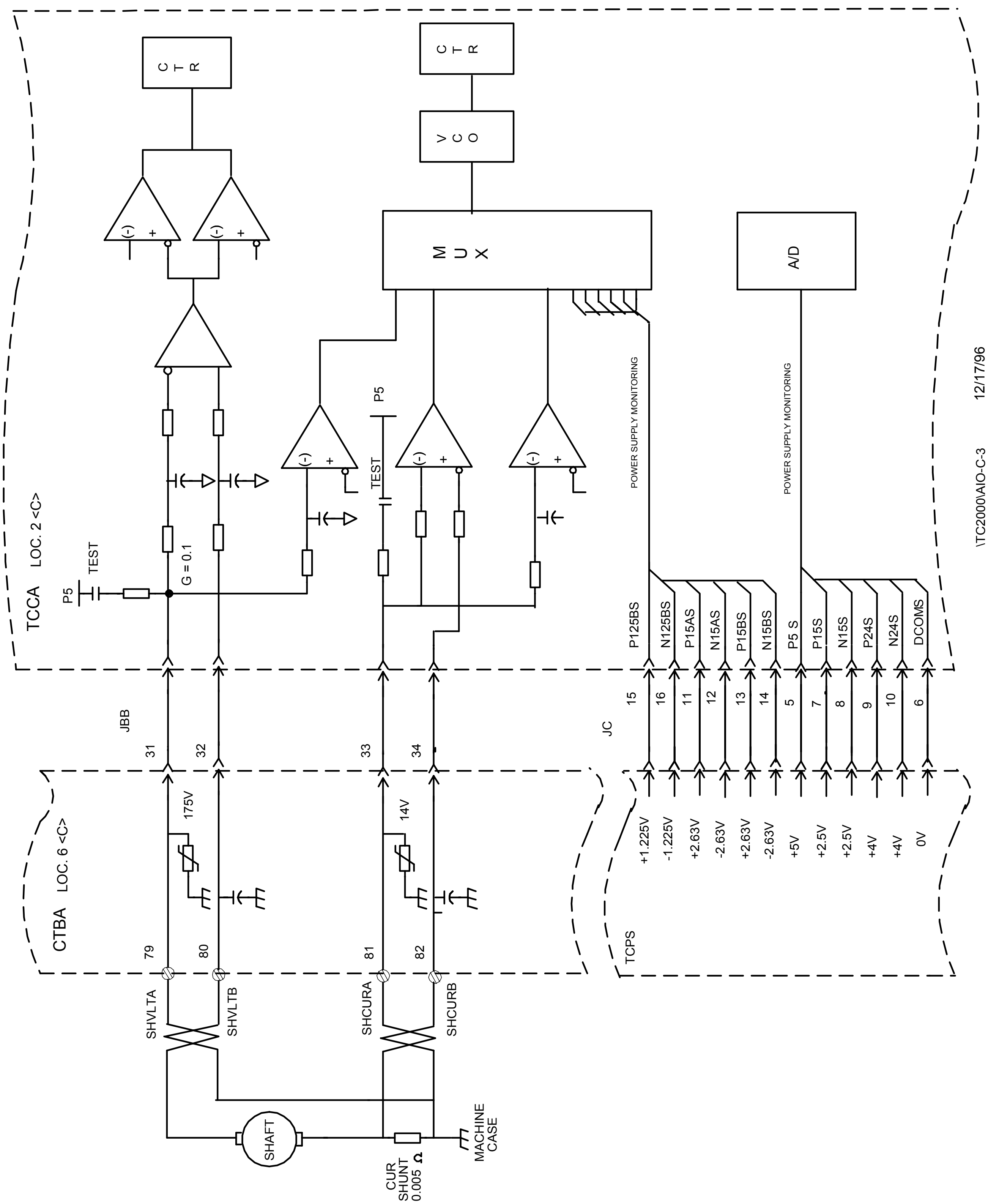


Figure D-17. <C> Core - Shaft Voltage Monitoring on CTBA Terminal Board

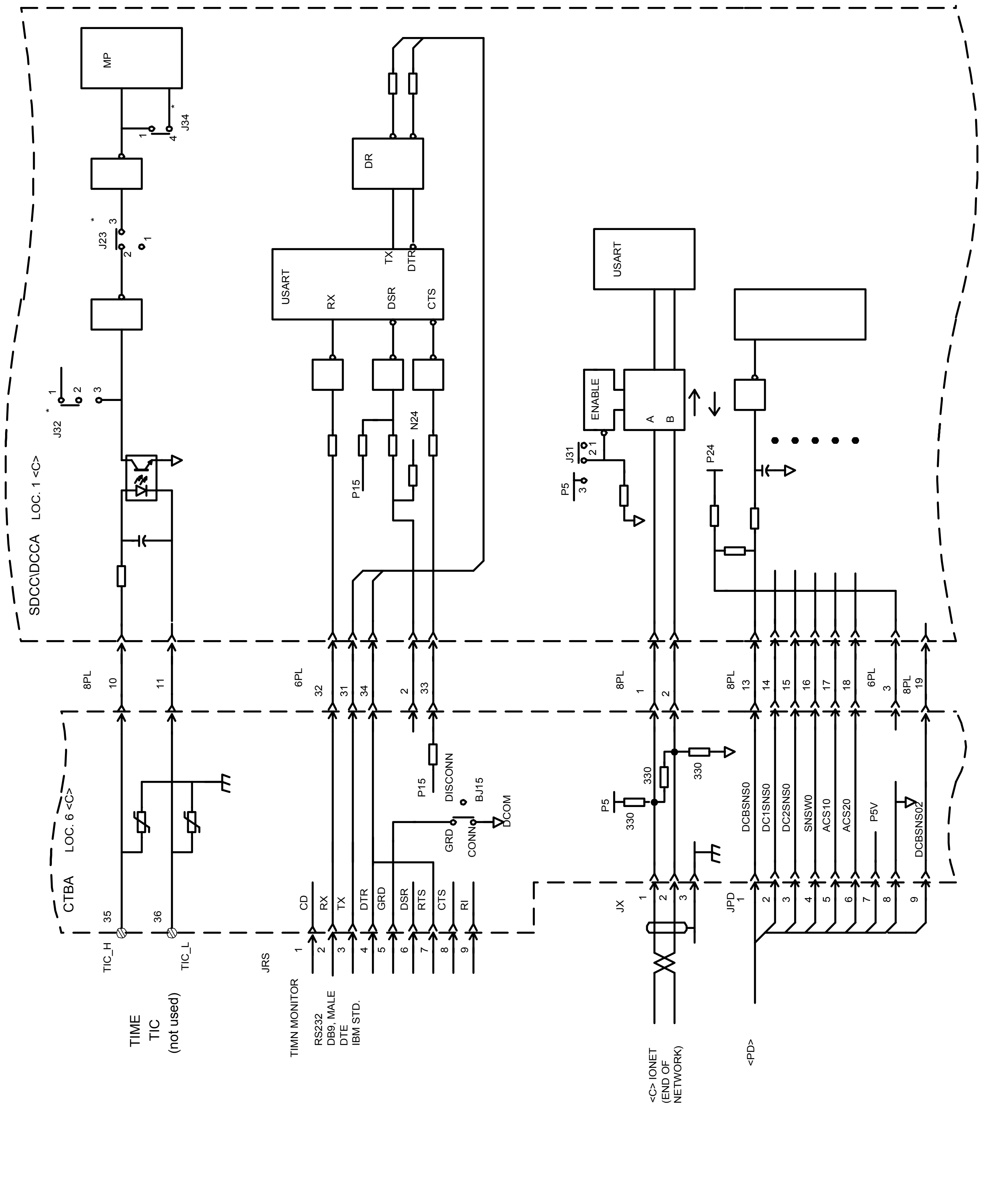


Figure D-18. <C> Core - TIMN Monitor and I/O Net Connections on CTBA

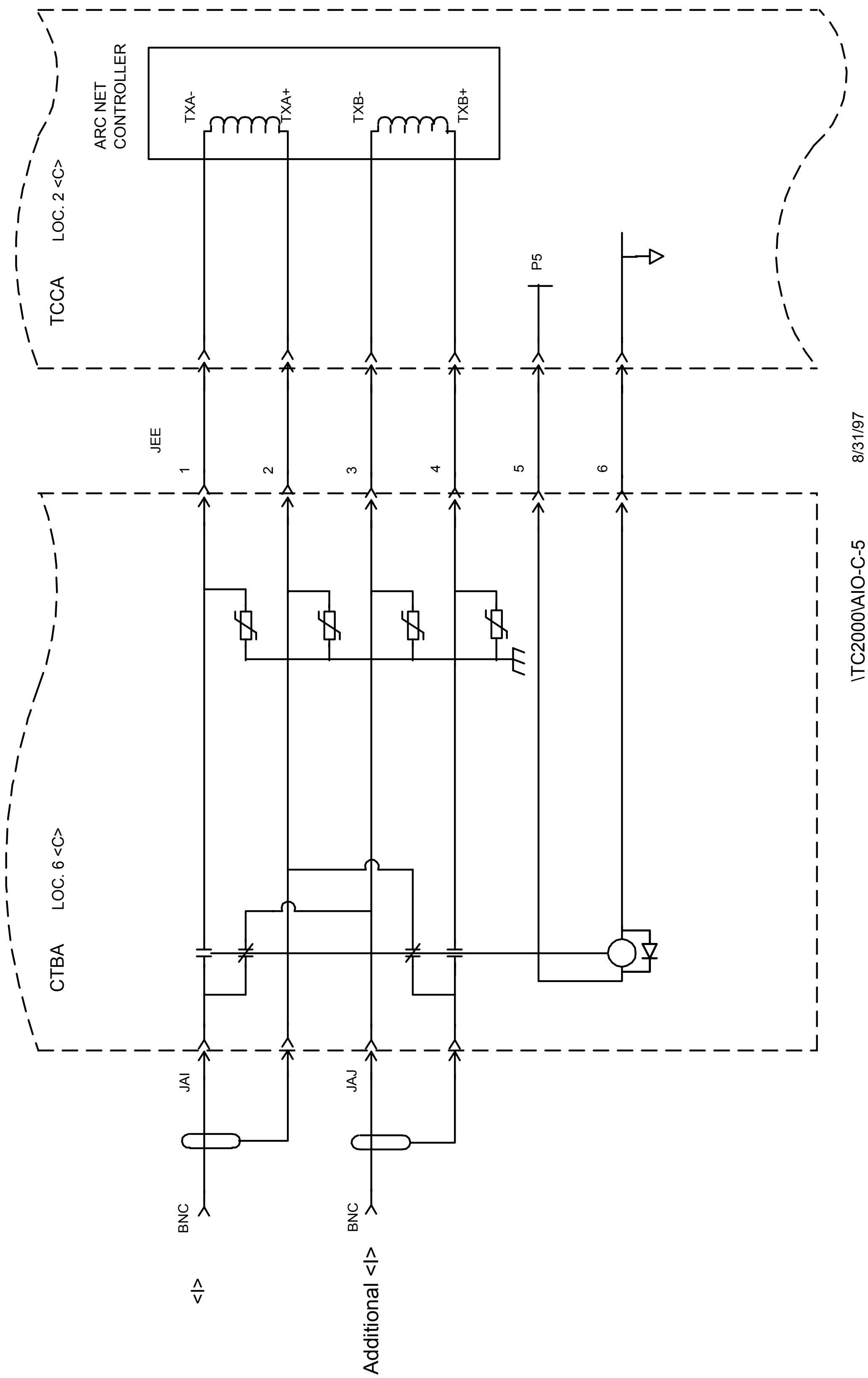


Figure D-19. <C> Core - Stage Link Connections on CTBA Terminal Board

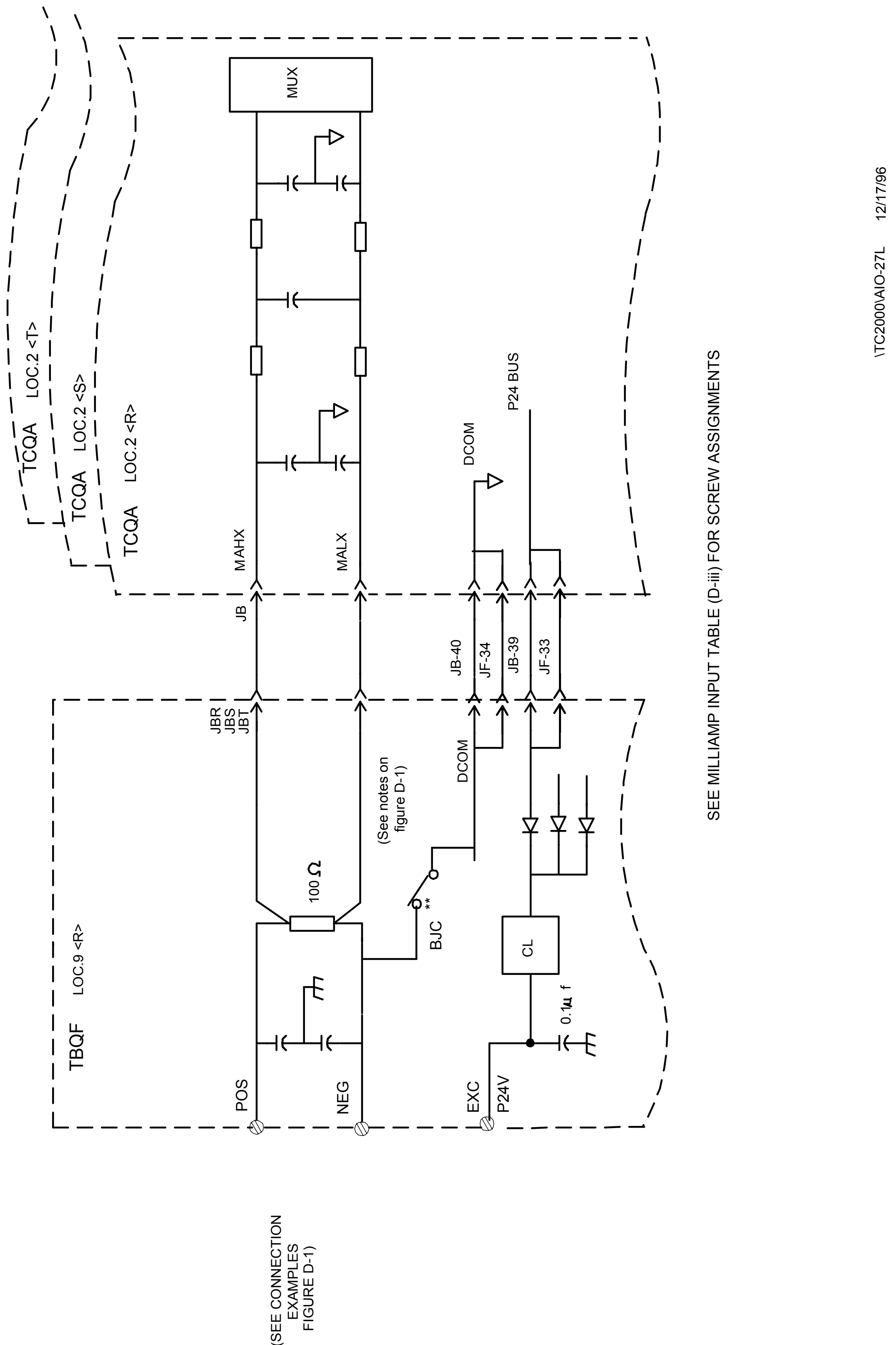


Figure D-20. <Q> Core - 4-20 mA Inputs on TBQF Terminal Board

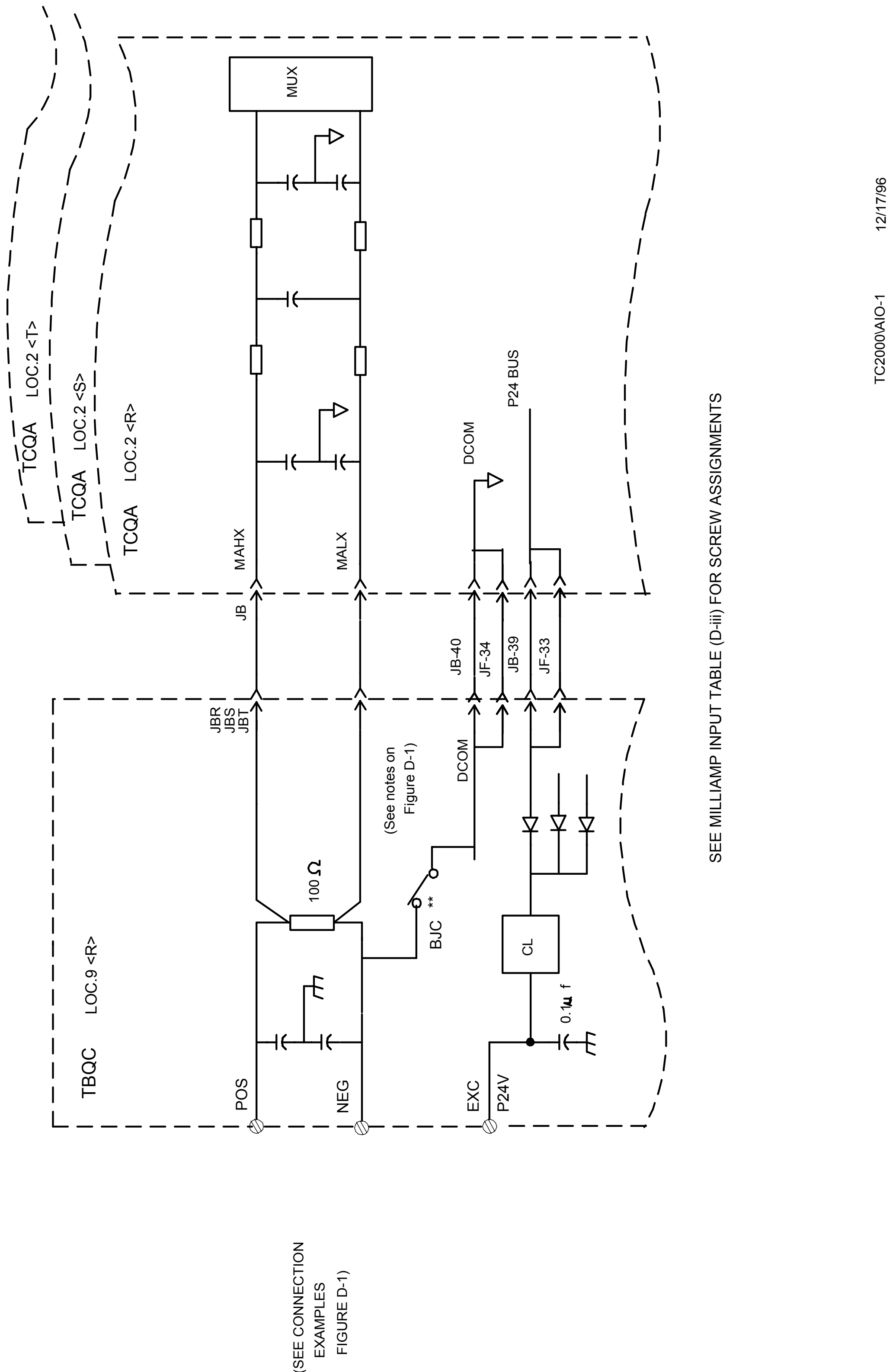
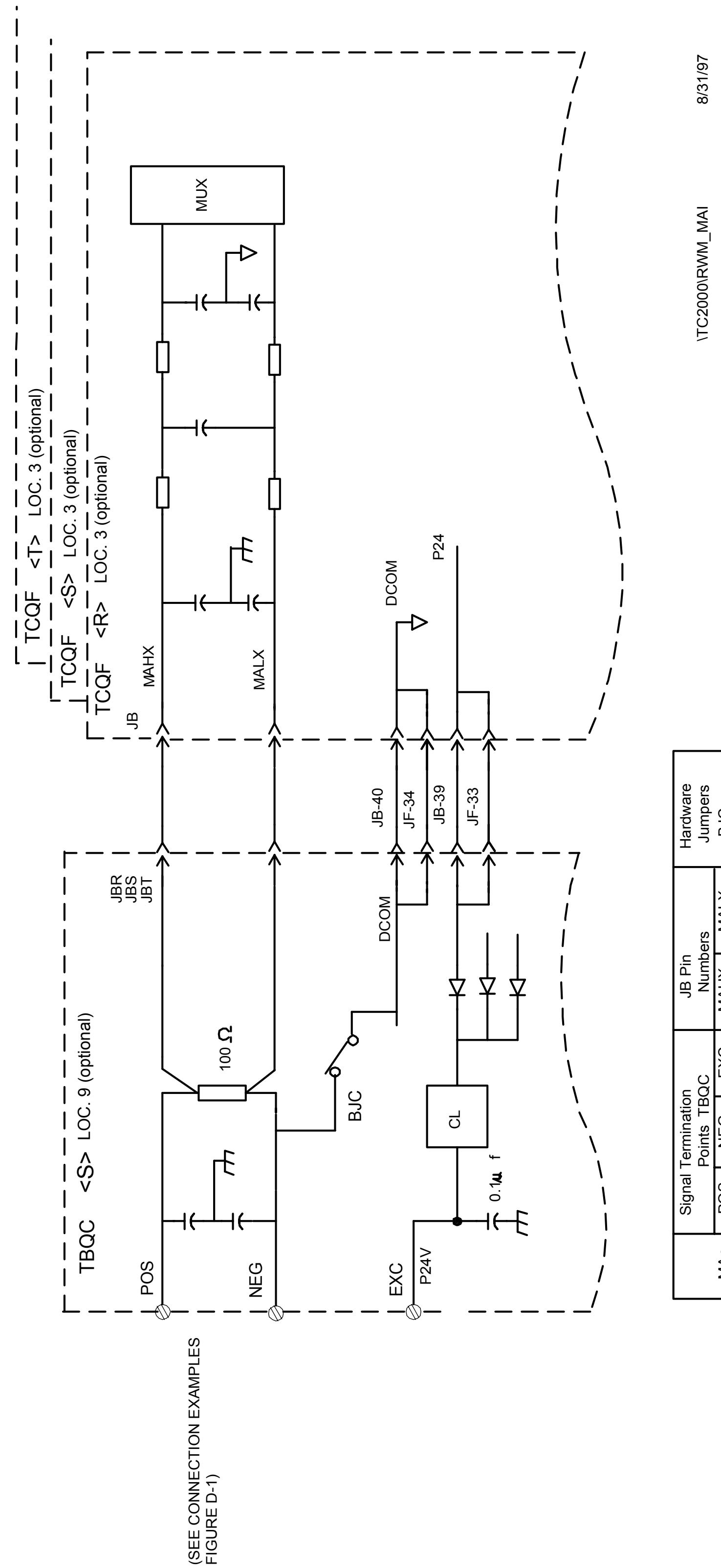


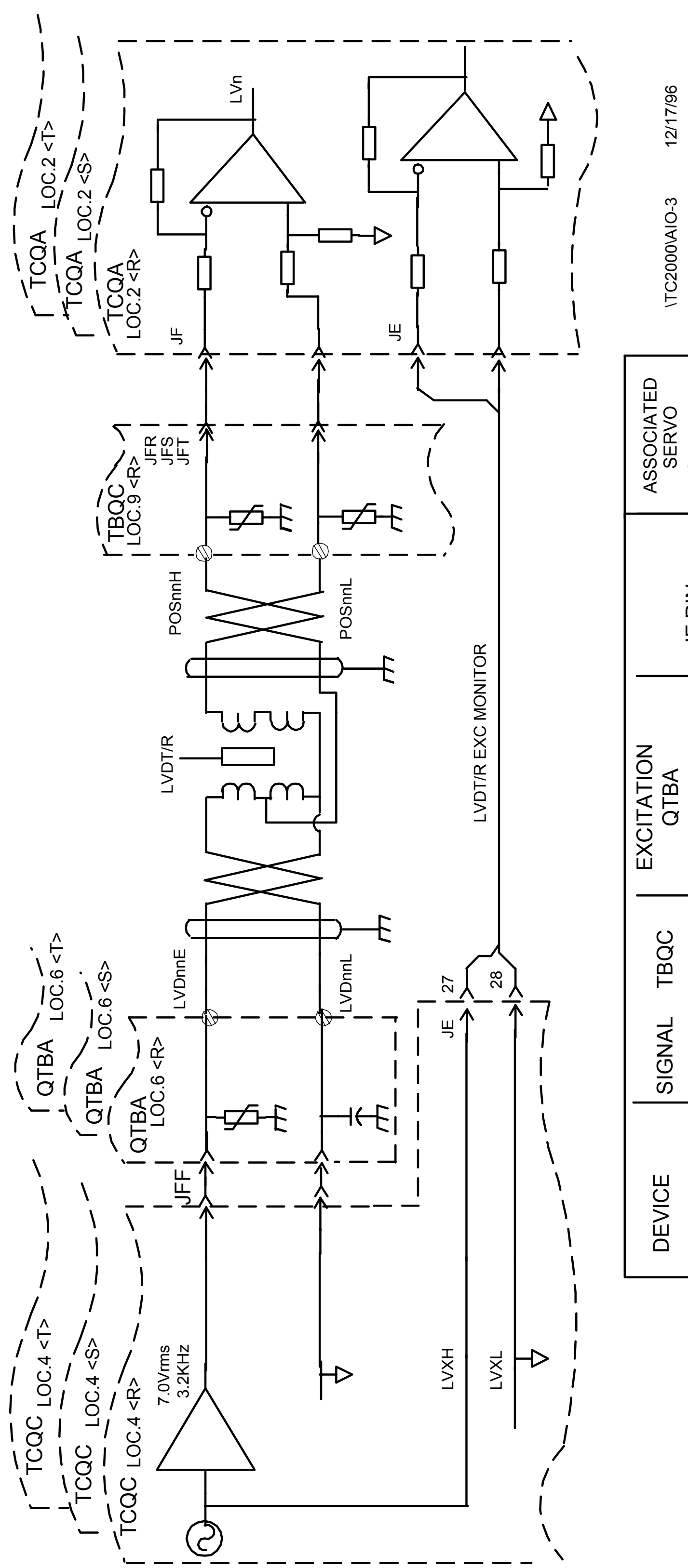
Figure D-21. <Q> Core - 4-20 mA Inputs on TBQC Terminal Board



4-20 MA Inputs for TCQF/TBQC

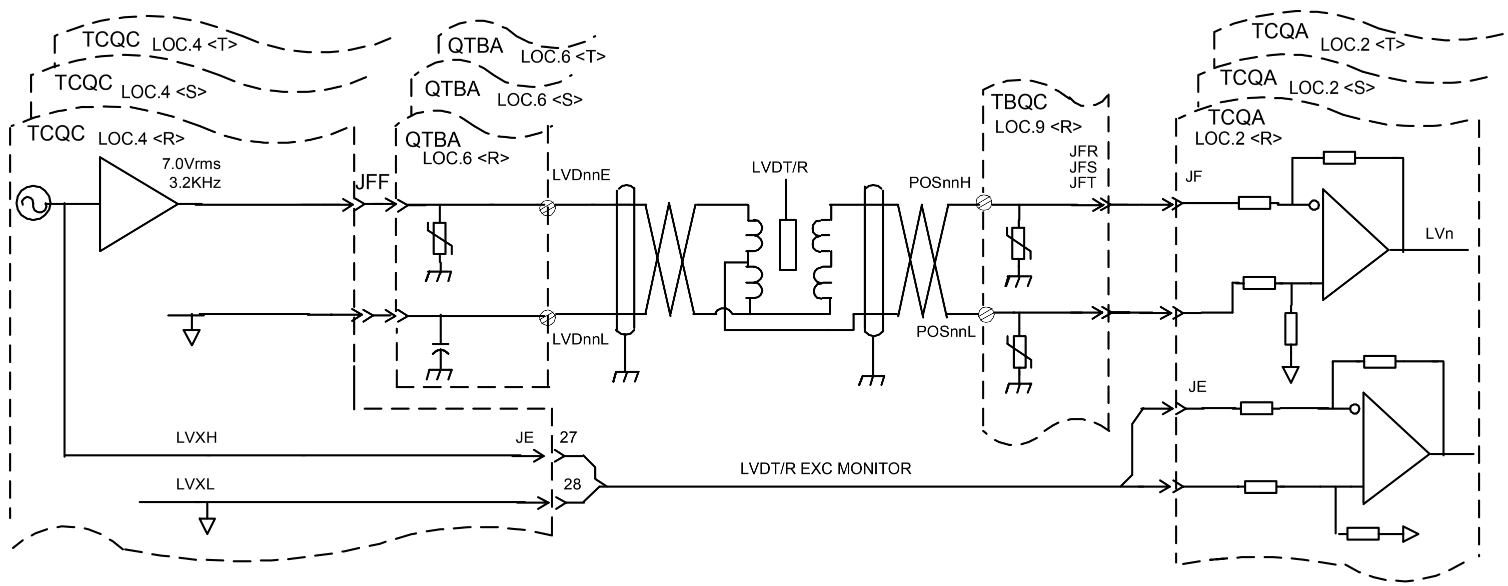
MAn	Signal Termination Points TBQC			JB Pin Numbers		Hardware Jumpers
	POS	NEG	EXC	MAHX	MALX	BJC
1	35	36	37	1	2	1
2	39	40	38	3	4	2
3	41	42	43	5	6	3
4	45	46	44	7	8	4
5	47	48	49	9	10	5
6	51	52	50	11	12	6
7	53	54	55	13	14	7
8	57	58	56	15	16	8
9	59	60	61	17	18	9
10	63	64	62	19	20	10
11	65	66	67	21	22	11
12	69	70	68	23	24	12
13	71	72	73	25	26	13
14	75	76	74	27	28	14
15	77	78	79,80	29	30	15

Figure D-22. <Q> Core - 4-20 mA Inputs on TBQC Terminal Board Using TCQF



DEVICE TB DESIGNATION	SIGNAL TBQC POSnnH POSnnL	EXCITATION QTBA LVDnnE LVDnnL	JF PIN	ASSOCIATED SERVO OUTPUT
LVD01	1,2	<R> 1,2	1,2	SV01
LVD02	3,4	<S> 1,2	3,4	SV01
LVD03	5,6	<T> 1,2	5,6	SV02
LVD04	7,8	<R> 3,4	7,8	SV02
LVD05	9,10	<S> 3,4	9,10	SV03
LVD06	11,12	<T> 3,4	11,12	SV03
LVD07	13,14	<R> 5,6	13,14	SV04
LVD08	15,16	<S> 5,6	15,16	SV04
LVD09	17,18	<T> 5,6	17,18	SV05
LVD10	19,20	<R> 7,8	19,20	SV05
LVD11	21,22	<S> 7,8	21,22	SV06
LVD12	23,24	<T> 7,8	23,24	SV06
LVD13	25,26	<R> 9,10	25,26	SV07
LVD14	27,28	<S> 9,10	27,28	SV07
LVD15	29,30	<T> 9,10	29,30	SV08
LVD16	31,32	<R> 11,12	31,32	SV08

Figure D-23. <Q> Core - LVDT Excitation an Feedback Signal on QTBA/TBQC

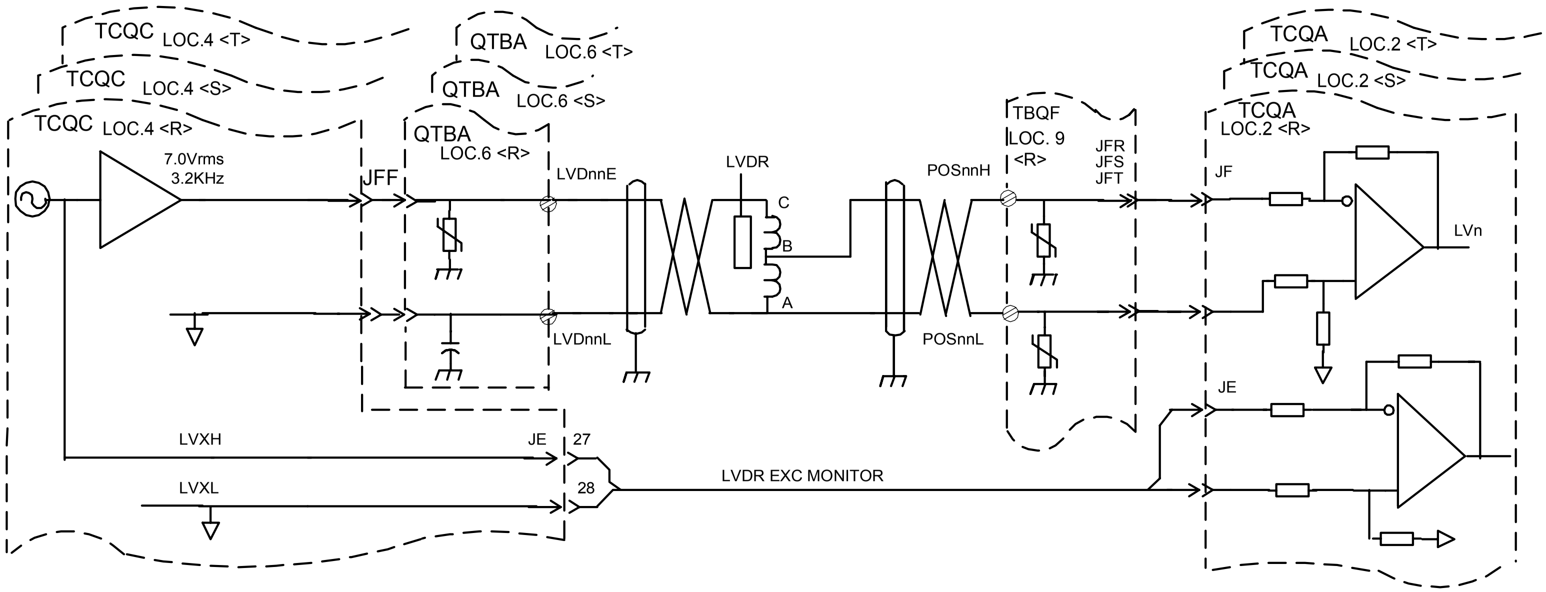


\TC2000\AIO-28Y 12/17/96

DEVICE TB DESIGNATION	SIGNAL TBQC POSnnH POSnnL	EXCITATION QTBA LVDnnE LVDnnL	JF PIN	ASSOCIATED SERVO OUTPUT
LVD01	1,2	<R> 1,2	1,2	SV01
LVD02	3,4	<S> 1,2	3,4	SV01
LVD16	31,32	<T> 1,2	31,32	SV01
LVD03	5,6	<R> 3,4	5,6	SV02
LVD04	7,8	<S> 3,4	7,8	SV02
LVD15	29,30	<T> 3,4	29,30	SV02
LVD05	9,10	<R> 5,6	9,10	SV03
LVD06	11,12	<S> 5,6	11,12	SV03
LVD14	27,28	<T> 5,6	27,28	SV03
LVD07	13,14	<R> 7,8	13,14	SV04
LVD08	15,16	<S> 7,8	15,16	SV04
LVD13	25,26	<T> 7,8	25,26	SV04
LVD09	17,18	<R> 9,10	17,18	SV05
LVD10	19,20	<S> 9,10	19,20	SV05
LVD12	23,24	<T> 9,10	23,24	SV05
LVD11	21,22	<R> 11,12	21,22	SV06

NOTE: LVDT's ARE GROUPED FOR MEDIUM STEAM (SCHDY) APPLICATIONS USING REGULATOR TYPES 89,99

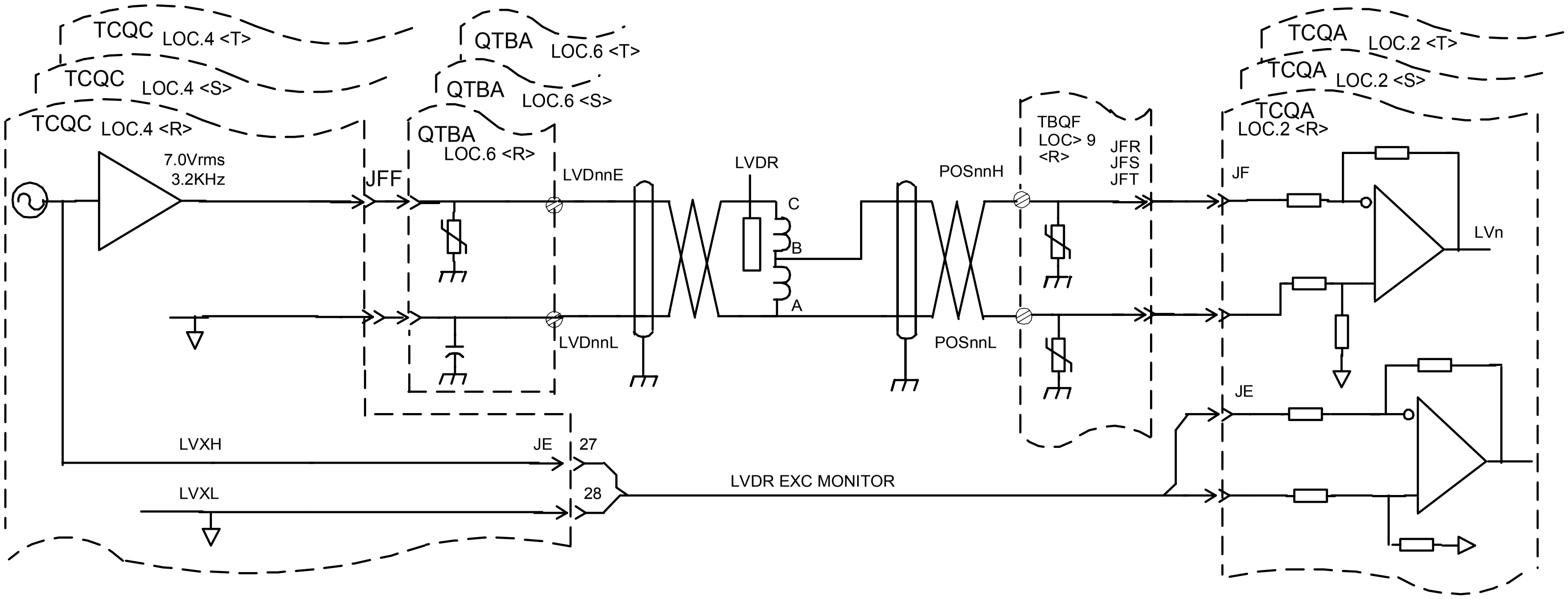
Figure D-24. <Q> Core - LVDT Excitation an Feedback Signal on QTBA/TBQC



CONTROL DEVICE	ASSOC. REGUL	LVDR	SCREW TERM, EXCITATION	SCREW TERM, FDBK SIGNAL	CABLE CONNECTION	TYP APPL
#1	1	LVDR1R	<R> QTBA 1,2	TBQF 1,2	JFR 1,2	CV1
		LVTR1S	<S> QTBA 1,2	TBQF 3,4	JFS 1,2	
		LVDR1T	<T> QTBA 1,2	TBQF 5,6	JFT 1,2	
#2	1	LVDR2	<R> QTBA 13,14	TBQF 7,8	JFR,S,T 3,4	MSV1
#3	2	LVDR3R	<R> QTBA 3,4	TBQF 9,10	JFR 5,6	CV2
		LVDR3S	<S> QTBA 3,4	TBQF 11,12	JFS 5,6	
		LVDR3T	<T> QTBA 3,4	TBQF 13,14	JFT 5,6	
#4	3	LVDR5	<S> QTBA 13,14	TBQF 15,16	JFR,S,T 9,10	MSV2
#5	3	LVDR6	<T> QTBA 13,14	TBQF 17,18	JFR,S,T 11,12	SPAR
#6	4	LVDR7R	<R> QTBA 5,6	TBQF 19,20	JFR 13,14	CV3
		LVDR7S	<S> QTBA 5,6	TBQF 21,22	JFS 13,14	
		LVDR7T	<T> QTBA 5,6	TBQF 23,24	JST 13,14	
#7	4	LVDR8	<R> QTBA 9,10	TBQF 25,26	JFR,S,T 15,16	MSV3
#8	5	LVDR9R	<R> QTBA 7,8	TBQF 27,28	JFR 17,18	CV4
		LVDR9S	<S> QTBA 7,8	TBQF 29,30	JFS 17,18	
		LVTR9T	<T> QTBA 7,8	TBQF 31,32	JFT 17,18	
#9	5	LVDR10	<S> QTBA 9,10	TBQF 33,34	JFR,S,T 19,20	MSV4
#10	6	LVDR11R	<R> QTBA 9,10	TBQF 85,86	JFR 21,22	IV1
		LVDT11S	<S> QTBA 9,10	TBQF 87,88	JFS 21,22	
		LVDR11T	<T> QTBA 9,10	TBQF 89.90	JFT 21,22	
#11	6	LVDR12	<T> QTBA 9,10	TBQF 91,92	JFR,S,T 23,24	IV4
#12	7	LVDR13R	<R> QTBA 11,12	TBQF 93,94	JFR 25,26	IV2
		LVDR13S	<S> QTBA 11,12	TBQF 95,96	JFS 25,26	
		LVDR13T	<T> QTBA 11,12	TBQF 97,98	JFT 25,26	
#13	7	LVDR14	<R> QTBA 11,12	TBQF 99,100	JFR,S,T 27,28	IV5
#14	8	LVDR15R	<R> QTBA 1,2	TBQF 101,102	JFR 29,30	IV3
		LVDR15S	<S> QTBA 1,2	TBQF 103,104	JFS 29,30	
		LVDR15T	<T> QTBA 1,2	TBQF 105,106	JFT 29,30	
#15	8	LVDR16	<S> QTBA 11,12	TBQF 107,108	JFR,S,T 31,32	IV6

TC2000\AIO-28L 8/31/97

Figure D-25. <Q> Core - LVDT Excitation an Feedback Signal on QTBA/TBQF G#A

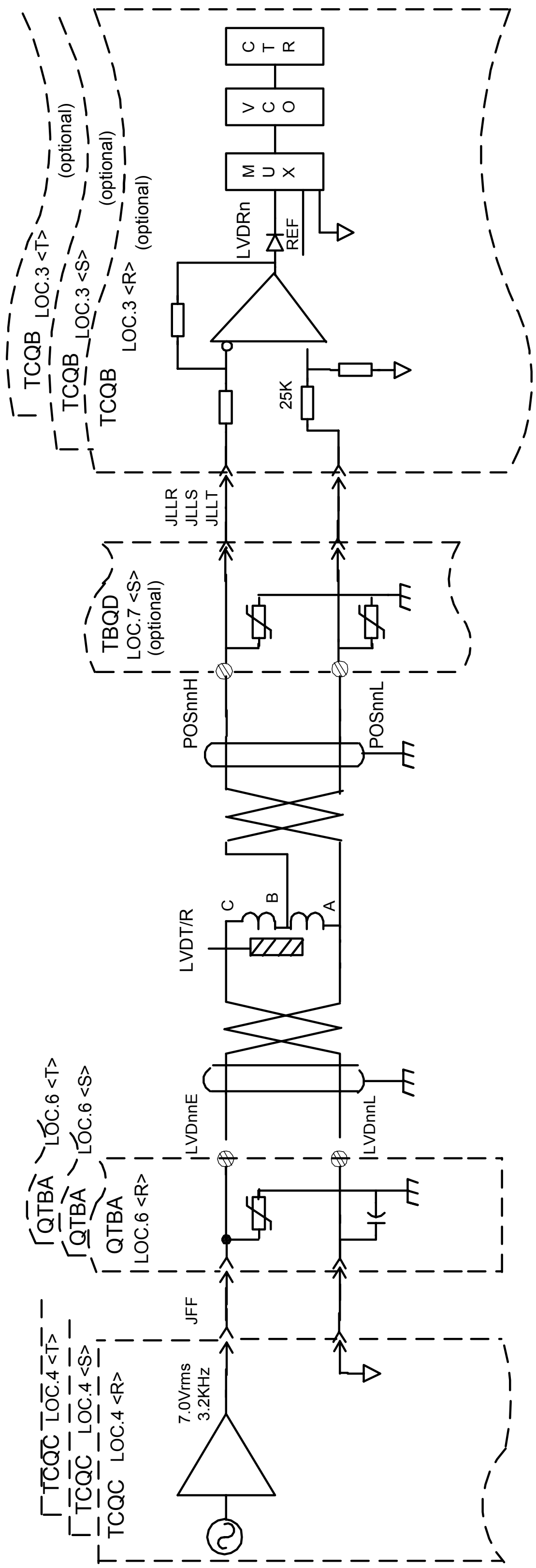


CONTROL DEVICE	ASSOC. REGUL	LVDR	SCREW TERM, EXCITATION	SCREW TERM, FDBK SIGNAL	CABLE CONNECTION	TYP APPL
#1	1	LVDR1R	<R> QTBA 1,2	TBQF 1,2	JFR 1,2	CV1
		LVTR1S	<S> QTBA 1,2	TBQF 3,4	JFS 1,2	
		LVDR1T	<T> QTBA 1,2	TBQF 5,6	JFT 1,2	
#2	1	LVDR2	<R> QTBA 13,14	TBQF 7,8	JFR,S,T 3,4	MSV1
#3	2	LVDR3R	<R> QTBA 3,4	TBQF 9,10	JFR 5,6	CV2
		LVDR3S	<S> QTBA 3,4	TBQF 11,12	JFS 5,6	
		LVDR3T	<T> QTBA 3,4	TBQF 13,14	JFT 5,6	
#4	5	LVDR9	<S> QTBA 13,14	TBQF 15,16	JFR,S,T 17,18	MSV2
#5	5	LVDR10	<T> QTBA 13,14	TBQF 17,18	JFR,S,T 19,20	SPAR
#6	3	LVDR5R	<R> QTBA 5,6	TBQF 19,20	JFR 9,10	CV3
		LVDR5S	<S> QTBA 5,6	TBQF 21,22	JFS 9,10	
		LVDR5T	<T> QTBA 5,6	TBQF 23,24	JFT 9,10	
#7	3	LVDR6	<R> QTBA 9,10	TBQF 25,26	JFR,S,T 11,12	MSV3
#8	4	LVDR7R	<R> QTBA 7,8	TBQF 27,28	JFR 13,14	CV4
		LVDR7S	<S> QTBA 7,8	TBQF 29,30	JFS 13,14	
		LVDR7T	<T> QTBA 7,8	TBQF 31,32	JFT 13,14	
#9	4	LVDR8	<S> QTBA 9,10	TBQF 33,34	JFR,S,T 15,16	MSV4
#10	6	LVDR11R	<R> QTBA 9,10	TBQF 85,86	JFR 21,22	IV1
		LVDT11S	<S> QTBA 9,10	TBQF 87,88	JFS 21,22	
		LVDR11T	<T> QTBA 9,10	TBQF 89.90	JFT 21,22	
#11	6	LVDR12	<T> QTBA 9,10	TBQF 91,92	JFR,S,T 23,24	IV4
#12	7	LVDR13R	<R> QTBA 11,12	TBQF 93,94	JFR 25,26	IV2
		LVDR13S	<S> QTBA 11,12	TBQF 95,96	JFS 25,26	
		LVDR13T	<T> QTBA 11,12	TBQF 97,98	JFT 25,26	
#13	7	LVDR14	<R> QTBA 11,12	TBQF 99,100	JFR,S,T 27,28	IV5
#14	8	LVDR15R	<R> QTBA 1,2	TBQF 101,102	JFR 29,30	IV3
		LVDR15S	<S> QTBA 1,2	TBQF 103,104	JFS 29,30	
		LVDR15T	<T> QTBA 1,2	TBQF 105,106	JFT 29,30	
#15	8	LVDR16	<S> QTBA 11,12	TBQF 107,108	JFR,S,T 31,32	IV6

TC2000\AIO-28LX 8/31/97

NOTE: This definition is applicable where the interface termination board DS200TBQFG1B is used. Note that the assignments are CV1-4 to REGUL 1-4 thus allowing the use of the 80 mA servo dr option.

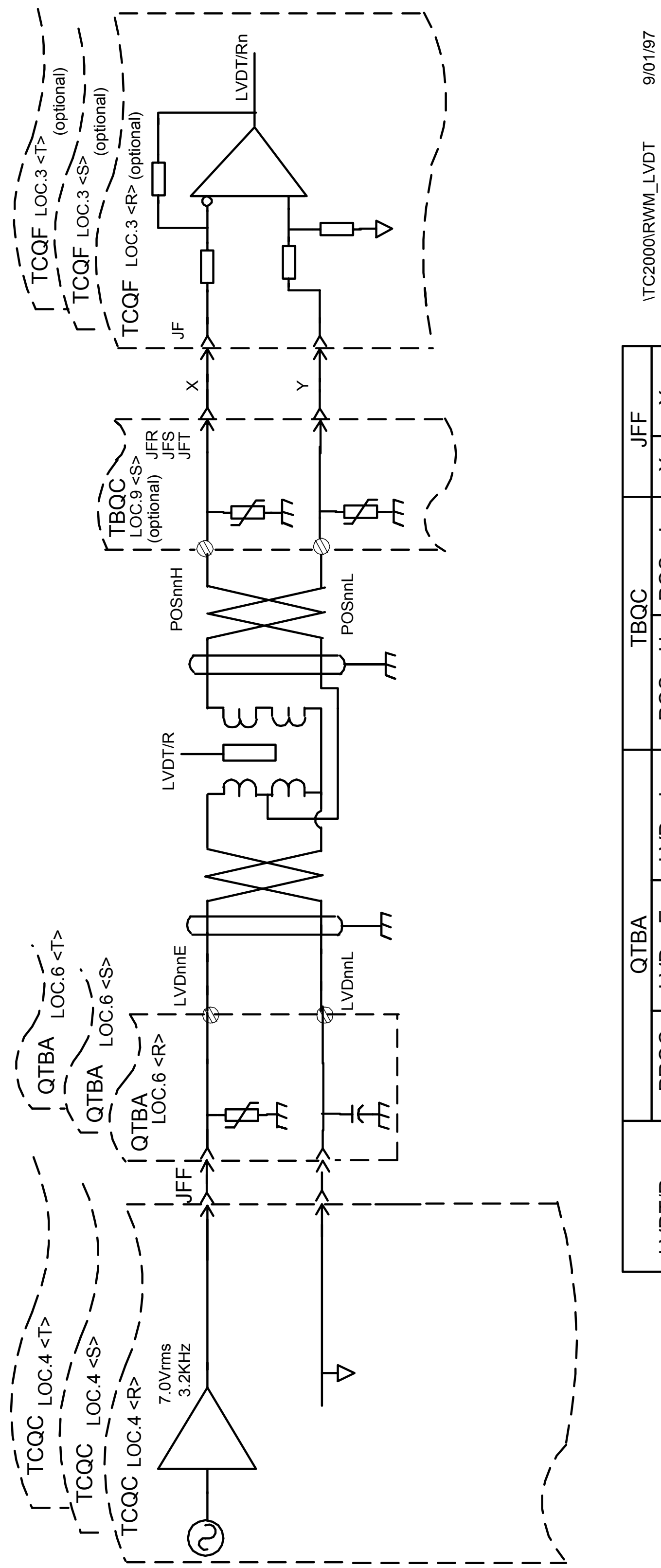
Figure D-26. <Q> Core - LVDT Excitation an Feedback Signal on QTBA/TBQF G#B



\TC2000\AIO-15 12/17/96

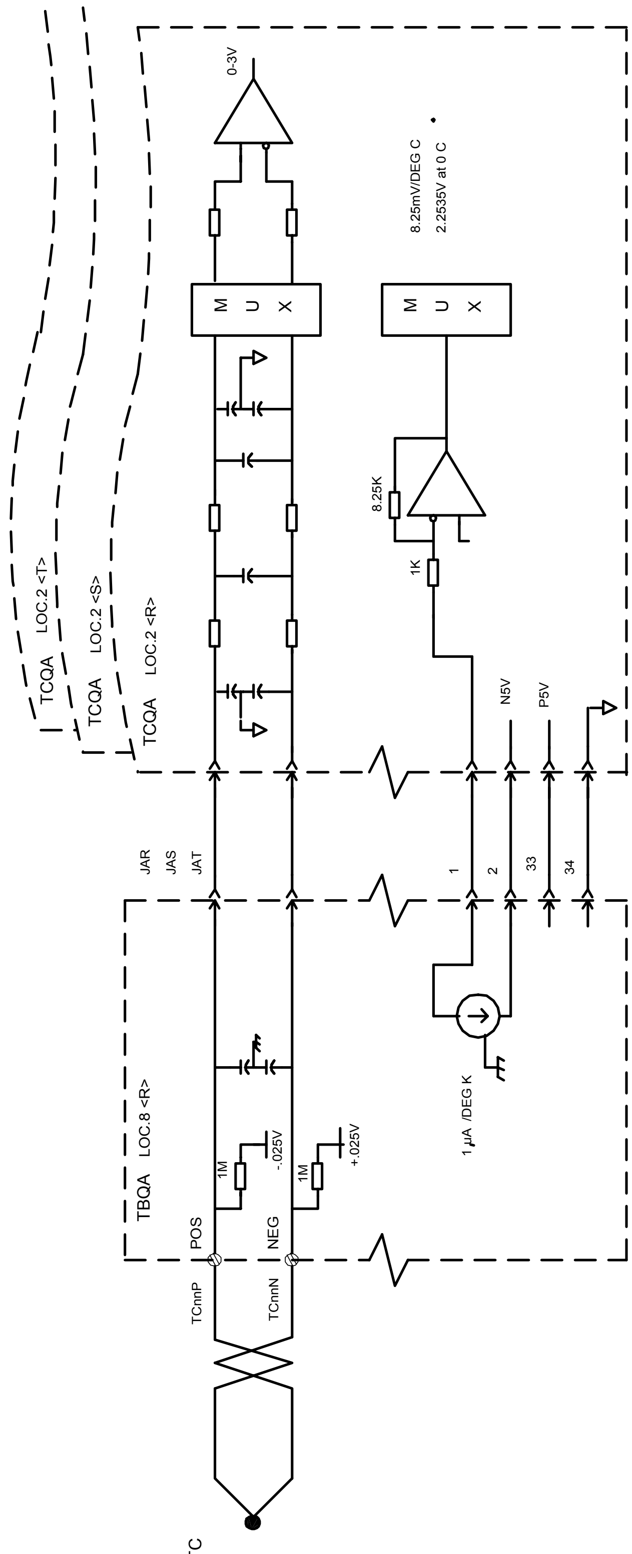
LVDR	SCREW TERM, EXCITATION LVDnnE LVDnnL	SCREW TERM, FDBK SIGNAL POSnnH POSnnL	CABLE CONNECTION	TYP APPL
LVDR1	<R> QTBA 15,16	TBQD 91,92	JLLR,S,T 11,12	ISV1
LVDR2	<S> QTBA 15,16	TBQD 93,94	JLLR,S,T 13,14	ISV2
LVDR3	<T> QTBA 15,16	TBQD 95,96	JLLR,S,T 15,16	ISV3
LVDR4	<R> QTBA 3,4	TBQD 97,98	JLLR,S,T 17,18	ISV4
LVDR5	<S> QTBA 3,4	TBQD 99,100	JLLR,S,T 19,20	ISV5
LVDR6	<T> QTBA 3,4	TBQD 101,102	JLLR,S,T 21,22	ISV6

Figure D-27. <Q> Core - LVDT Excitation an Feedback Signal on QTBA/TBQD



LVDT/Rn	QTBA			TBQC		JFF	
	PROC.	LVDnnE	LVDnnL	POSnnH	POSnnL	X	Y
LVDT/R1	<R>	15	16	1	2	1	2
LVDT/R2	<S>	15	16	3	4	3	4
LVDT/R3	<T>	15	16	5	6	5	6
LVDT/R4	<R>	3	4	7	8	7	8
LVDT/R5	<S>	3	4	9	10	9	10
LVDT/R6	<T>	3	4	11	12	11	12
LVDT/R7	<S>	11	12	13	14	13	14
LVDT/R8	<T>	11	12	15	16	15	16

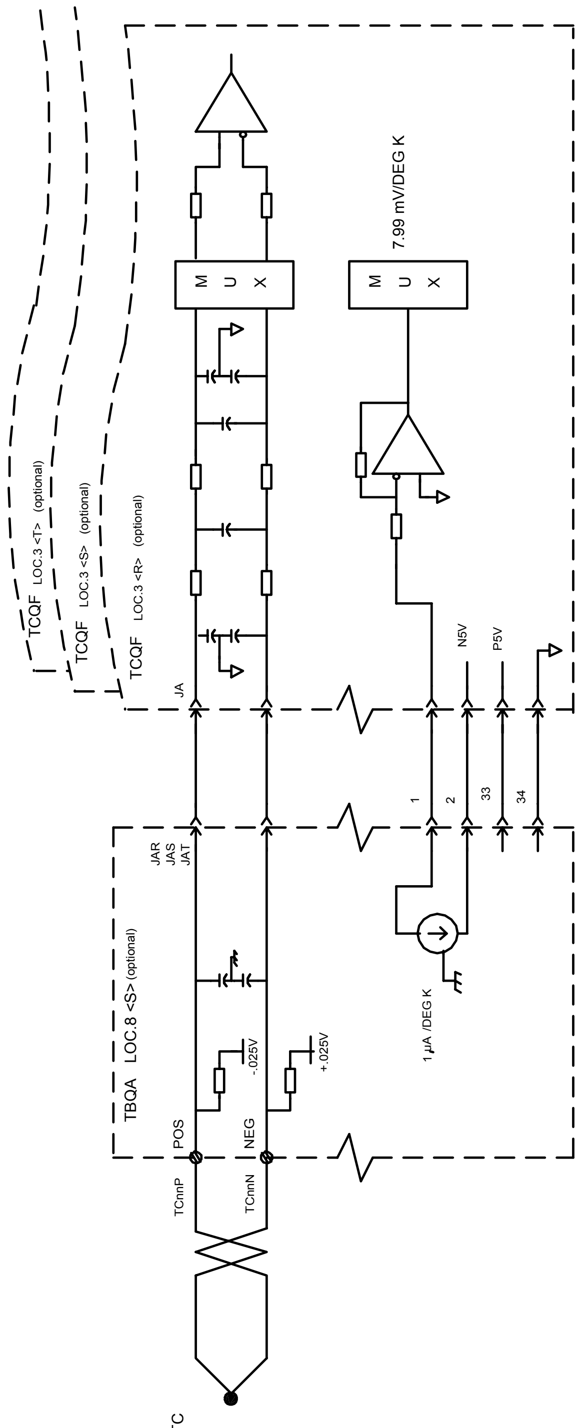
Figure D-28. <Q> Core - LVDT Excitation an Feedback Signal on QTBA/TBQC Using TCQF



\TC2000\AIO-11 12/17/96

<R> DEVICE TB DESIGNATION	TBQA TERMINATION POS	TBQA TERMINATION NEG	<S> DEVICE TB DESIGNATION	TBQA TERMINATION POS	TBQA TERMINATION NEG	<T> DEVICE TB DESIGNATION	TBQA TERMINATION POS	TBQA TERMINATION NEG
TC01	1	2	TC16	31	32	TC31	61	62
TC02	3	4	TC17	33	34	TC32	63	64
TC03	5	6	TC18	35	36	TC33	65	66
TC04	7	8	TC19	37	38	TC34	67	68
TC05	9	10	TC20	39	40	TC35	69	70
TC06	11	12	TC21	41	42	TC36	71	72
TC07	13	14	TC22	43	44	TC37	73	74
TC08	15	16	TC23	45	46	TC38	75	76
TC09	17	18	TC24	47	48	TC39	77	78
TC10	19	20	TC25	49	50	TC40	79	80
TC11	21	22	TC26	51	52	TC41	81	82
TC12	23	24	TC27	53	54	TC42	83	84
TC13	25	26	TC28	55	56	TC43	85	86
TC14	27	28	TC29	57	58	TC44	87	88
TC15	29	30	TC30	59	60	TC45	89	90

Figure D-29. <Q> Core -Thermocouple Inputs on TBQA Terminal Board



\TC2000\RWM_TC 9/01/97

GENERIC VOTED I/O SIGNAL NAME *	<R> DEVICE TB DESIGNATION	TBQA TERMINATION POS	TBQA TERMINATION NEG	<S> DEVICE TB DESIGNATION	TBQA TERMINATION POS	TBQA TERMINATION NEG	<T> DEVICE TB DESIGNATION	TBQA TERMINATION POS	TBQA TERMINATION NEG	JA POS	JA NEG
Q_S_TC17	TC01	1	2	TC16	31	32	TC31	61	62	3	4
Q_S_TC18	TC02	3	4	TC17	33	34	TC32	63	64	5	6
Q_S_TC19	TC03	5	6	TC18	35	36	TC33	65	66	7	8
Q_S_TC20	TC04	7	8	TC19	37	38	TC34	67	68	9	10
Q_S_TC21	TC05	9	10	TC20	39	40	TC35	69	70	11	12
Q_S_TC22	TC06	11	12	TC21	41	42	TC36	71	72	13	14
Q_S_TC23	TC07	13	14	TC22	43	44	TC37	73	74	15	16
Q_S_TC24	TC08	15	16	TC23	45	46	TC38	75	76	17	18
Q_S_TC25	TC09	17	18	TC24	47	48	TC39	77	78	19	20
Q_S_TC26	TC10	19	20	TC25	49	50	TC40	79	80	21	22
Q_S_TC27	TC11	21	22	TC26	51	52	TC41	81	82	23	24
Q_S_TC28	TC12	23	24	TC27	53	54	TC42	83	84	25	26
Q_S_TC29	TC13	25	26	TC28	55	56	TC43	85	86	27	28
Q_S_TC30	TC14	27	28	TC29	57	58	TC44	87	88	29	30
Q_S_TC31	TC15	29	30	TC30	59	60	TC45	89	90	31	32

* Signals are only available as Voted Signals

Figure D-30. <Q> Core -Thermocouple Inputs on TBQA Terminal Board Using TCQF

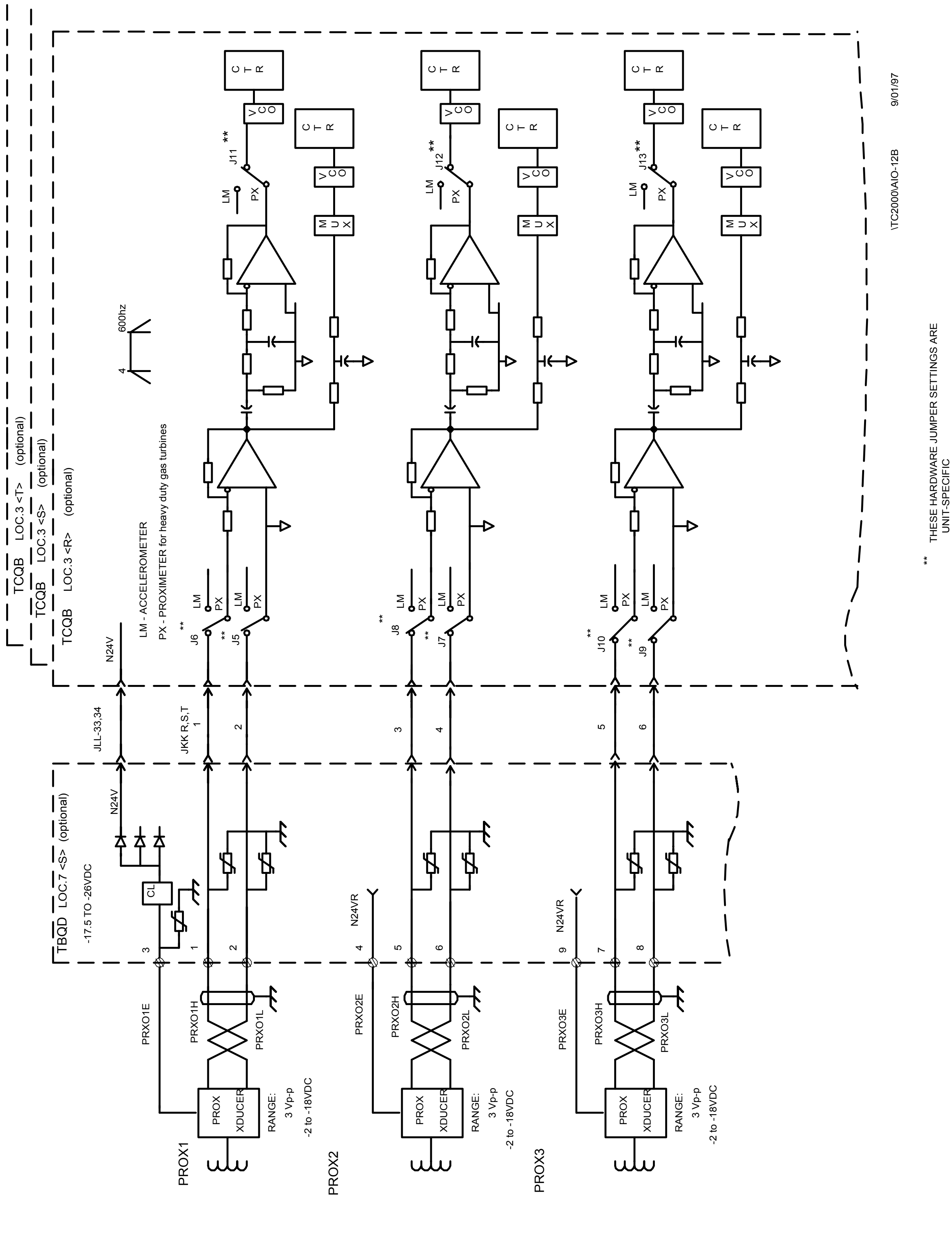
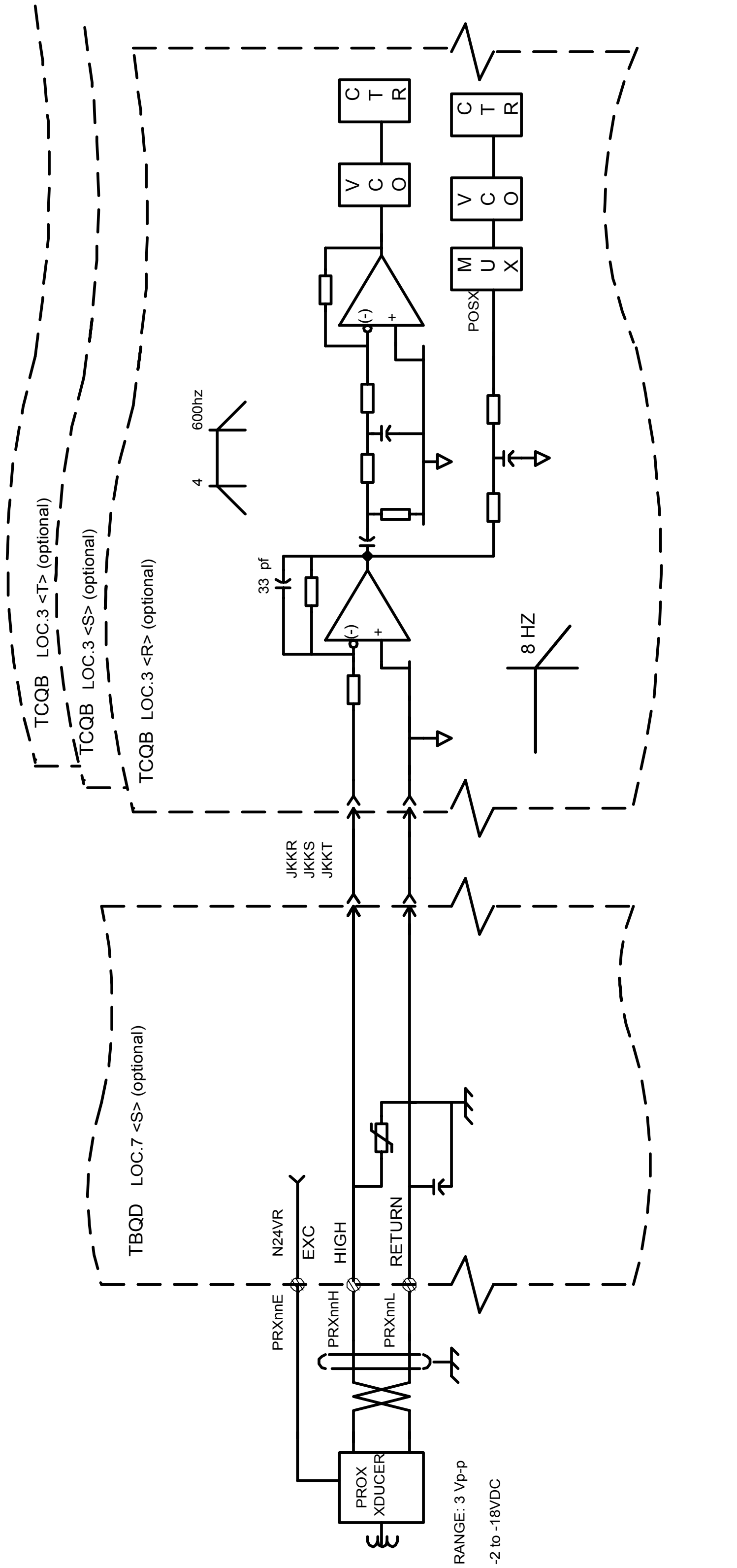
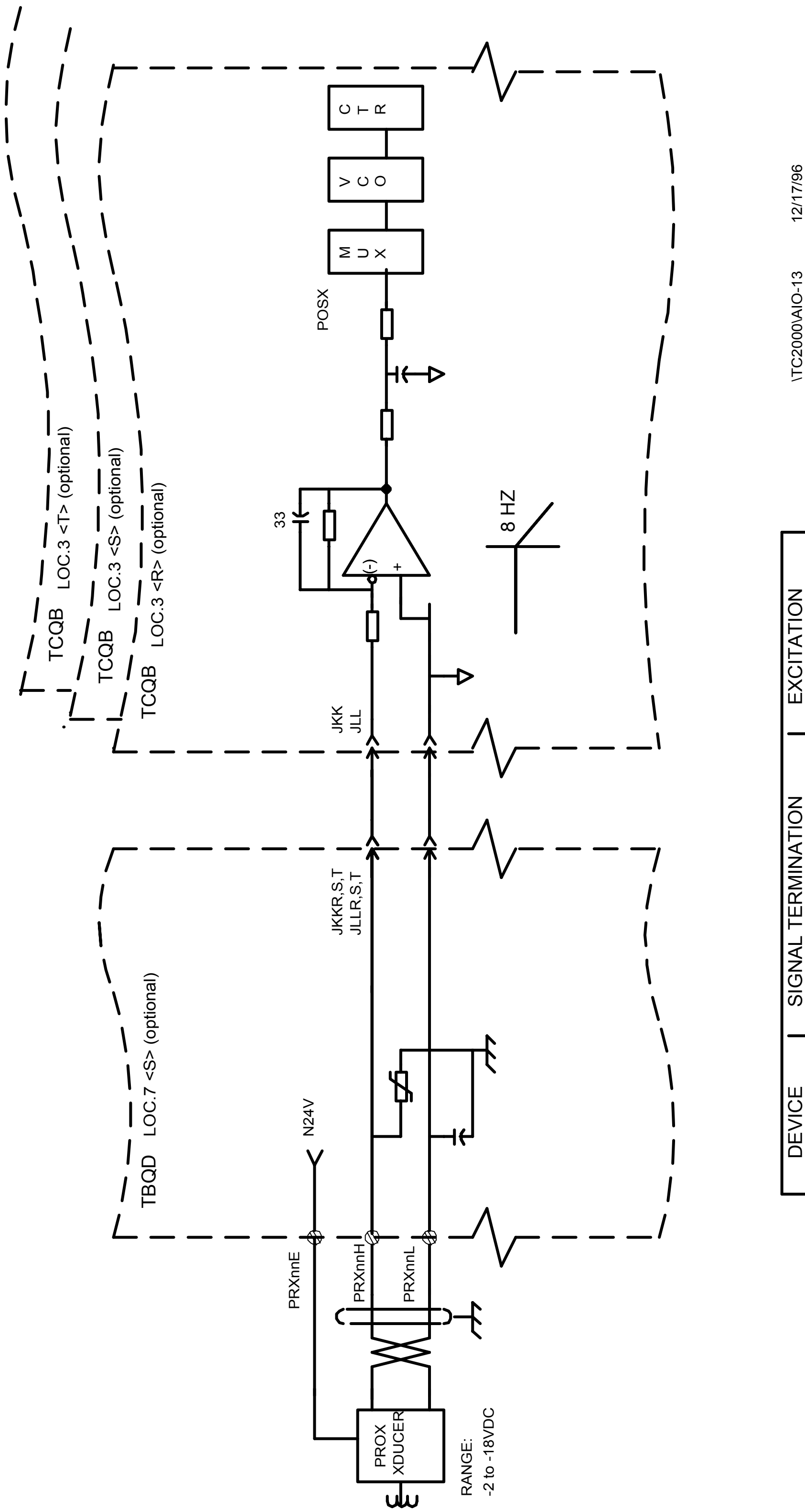


Figure D-31. <Q> Core -Proximity Transducer Inputs on TBQD Terminal Board



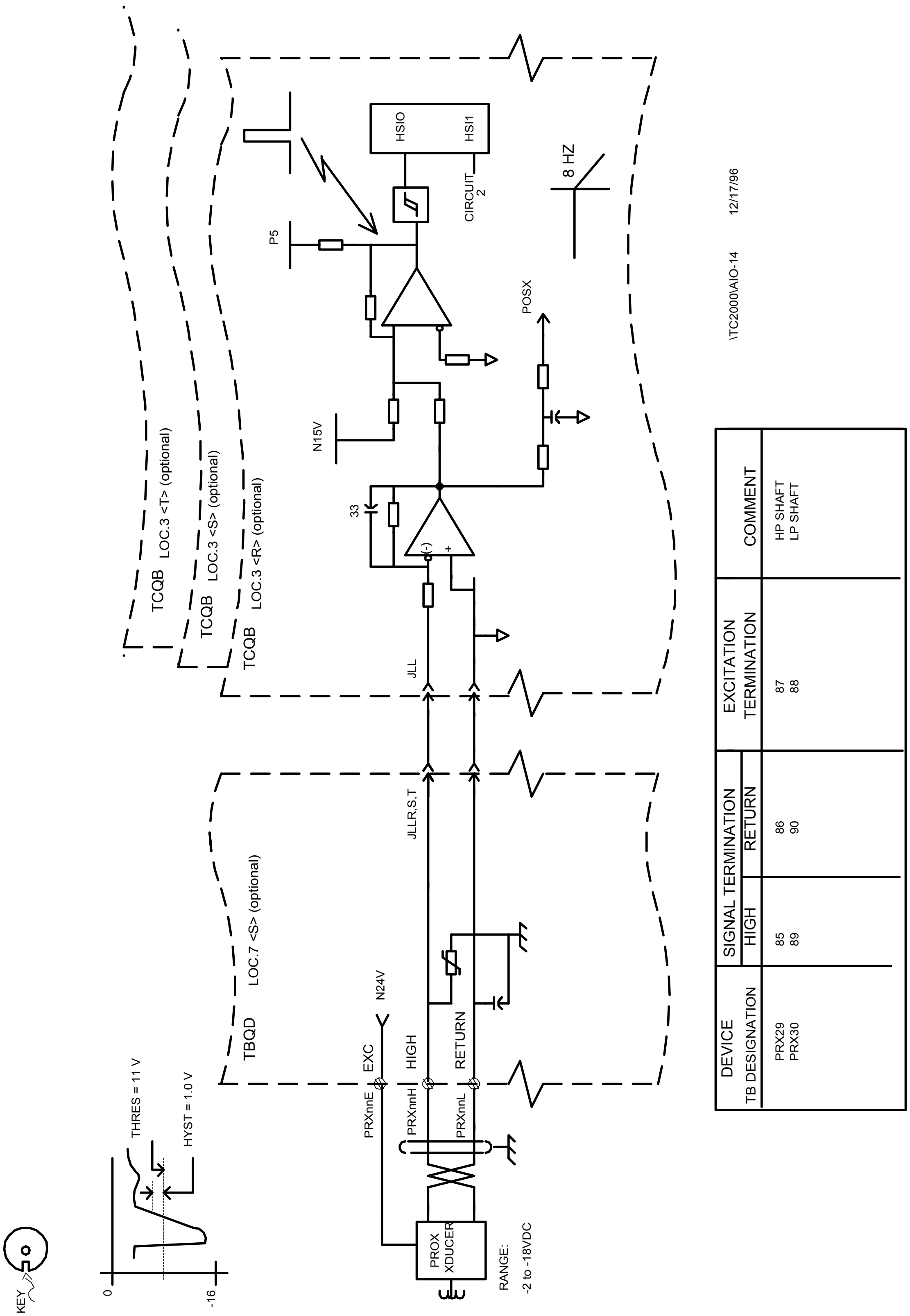
DEVICE TB DESIGNATION	SIGNAL TERMINATION HIGH	SIGNAL TERMINATION RETURN	EXCITATION TERMINATION
PRX04	11	12	10
PRX05	13	14	15
PRX06	17	18	16
PRX07	19	20	21
PRX08	23	24	22
PRX09	25	26	27
PRX10	29	30	28
PRX11	31	32	33
PRX12	35	36	34
PRX13	37	38	39
PRX14	41	42	40
PRX15	43	44	45
PRX16	47	48	46
PRX17	49	50	51
PRX18	53	54	52

Figure D-32. <Q> Core - Proximity Transducer Inputs on TBQD Terminal Board - Cont.



DEVICE	SIGNAL TERMINATION		EXCITATION
TB DESIGNATION	HIGH	RETURN	TERMINATION
PRX19	55	56	57
PRX20	59	60	58
PRX21	61	62	63
PRX22	65	66	64
PRX23	67	68	69
PRX24	71	72	70
PRX25	73	74	75
PRX26	77	78	76
PRX27	79	80	81
PRX28	83	84	82

Figure D-33. <Q> Core - Proximity Transducer Inputs on TBQD Terminal Board - Cont.



DEVICE TB DESIGNATION	SIGNAL TERMINATION HIGH	SIGNAL TERMINATION RETURN	EXCITATION TERMINATION	COMMENT
PRX29	85	86	87	HP SHAFT
PRX30	89	90	88	LP SHAFT

Figure D-34. <Q> Core - Proximity Transducer Inputs on TBQD Terminal Board - Cont.

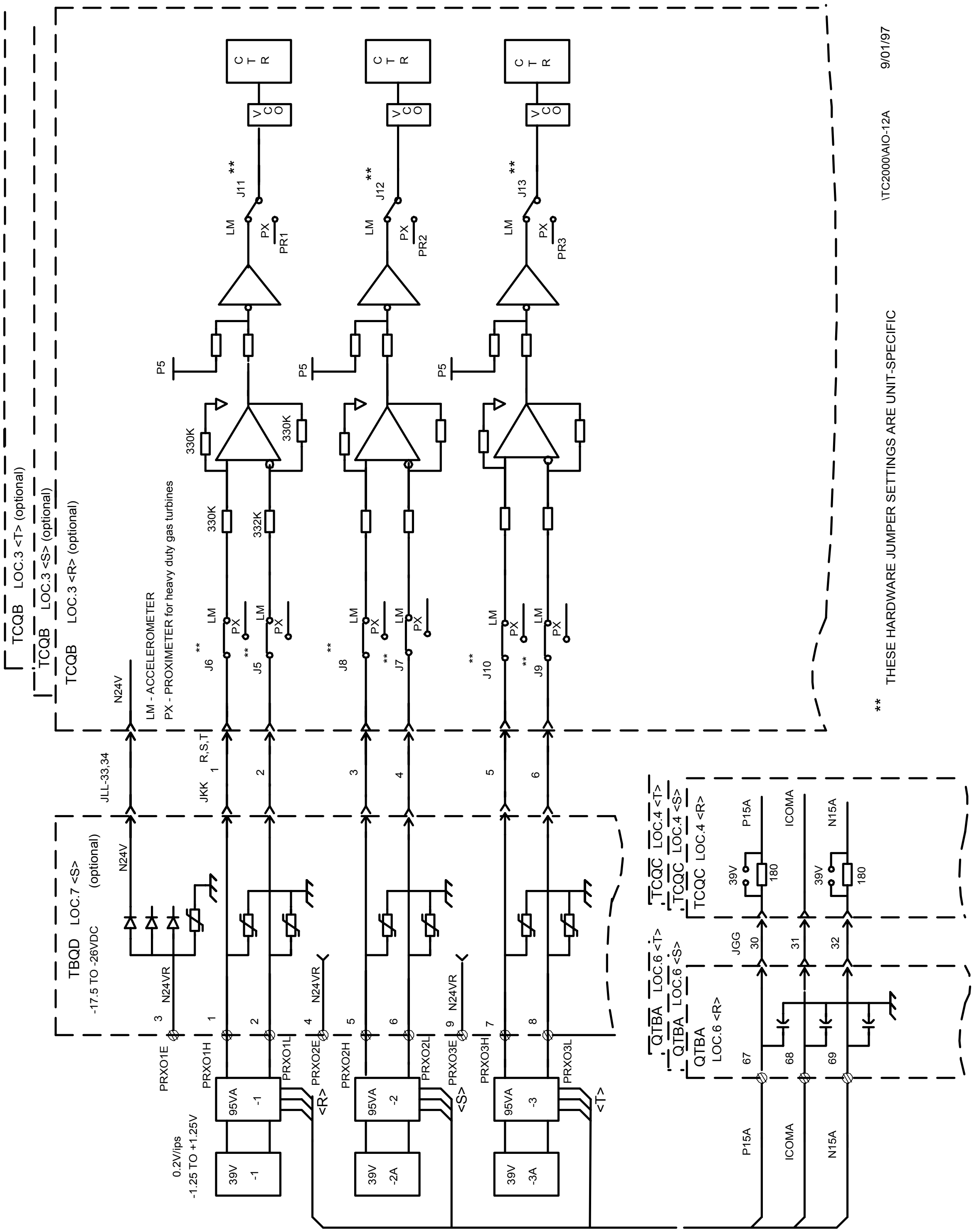
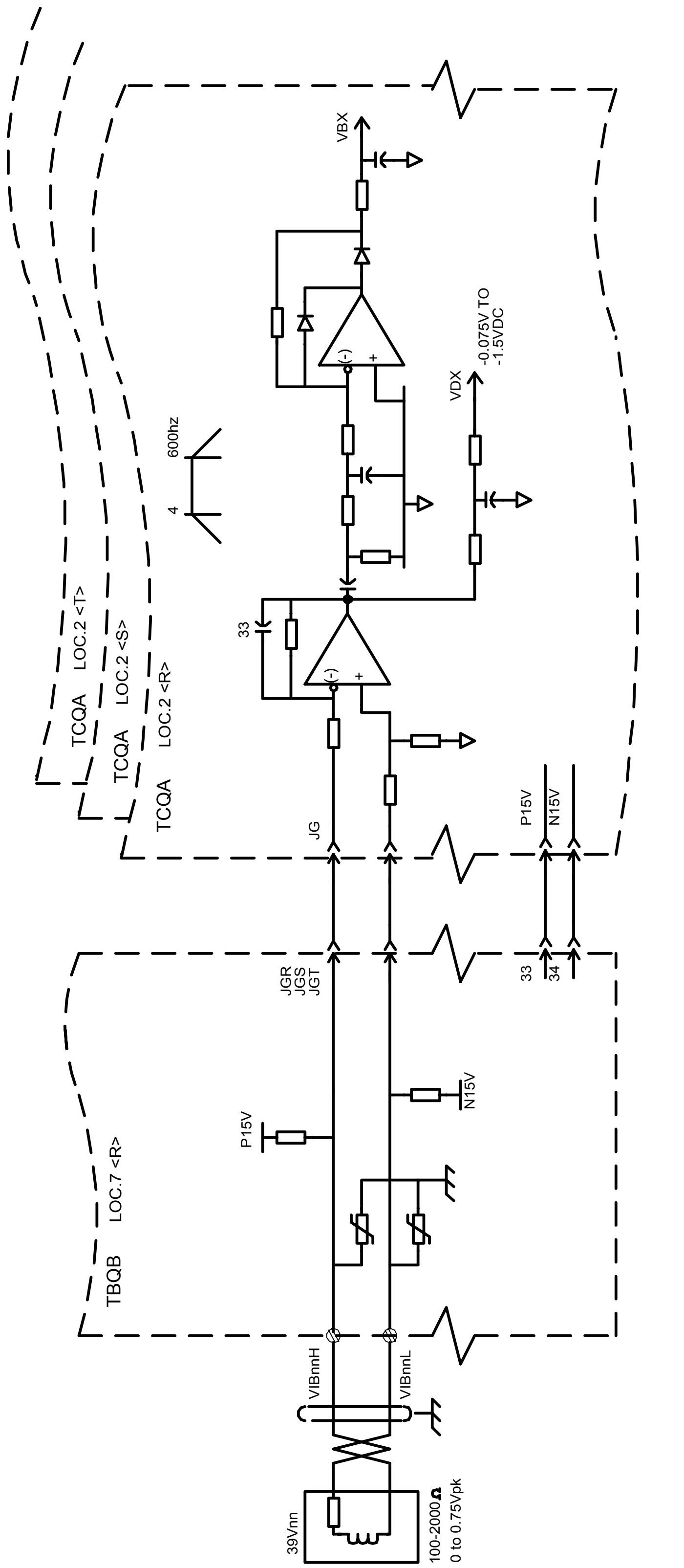


Figure D-35. <Q> Core - Accelerometer Inputs on TBQD Terminal Board



DEVICE TB DESIGNATION	TERMINATION SCREW VIBnnH VIBnnL	JG PIN #
39V01	51,52	9,10
39V02	53,54	11,12
39V03	55,56	13,14
39V04	57,58	15,16
39V05	59,60	17,18
39V06	61,62	19,20
39V07	63,64	21,22
39V08	65,66	23,24
39V09	67,68	25,26
39V10	69,70	27,28
39V11	71,72	29,30
39V12	73,74	31,32

Figure D-36. <Q> Core - Vibration Inputs on TBQB Terminal Board

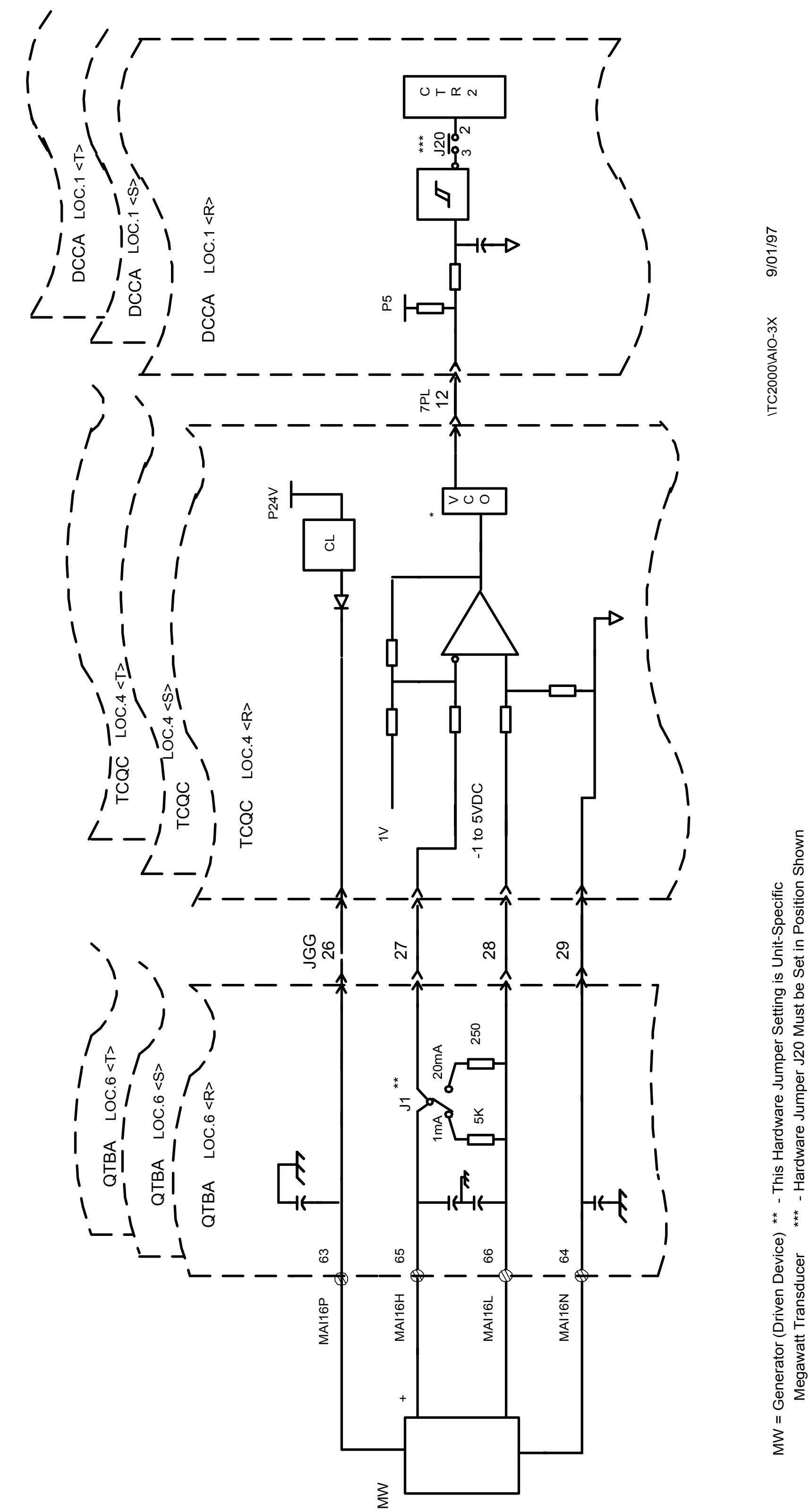


Figure D-37. <Q> Core - MegaWatt Transducer Inputs on QTBA Terminal Board

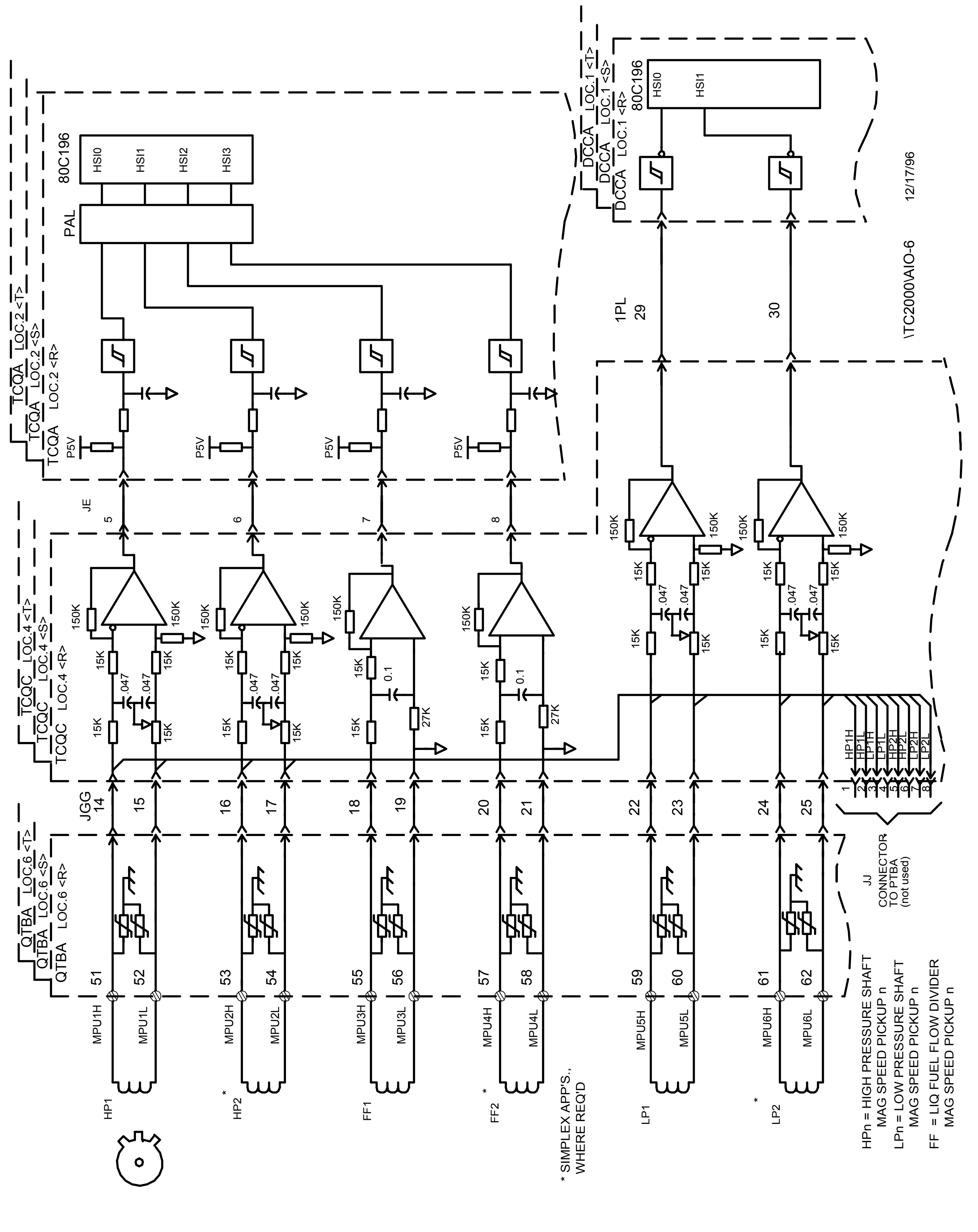


Figure D-38. <Q> Core - Pulse Rate Inputs on QTBA Terminal Board

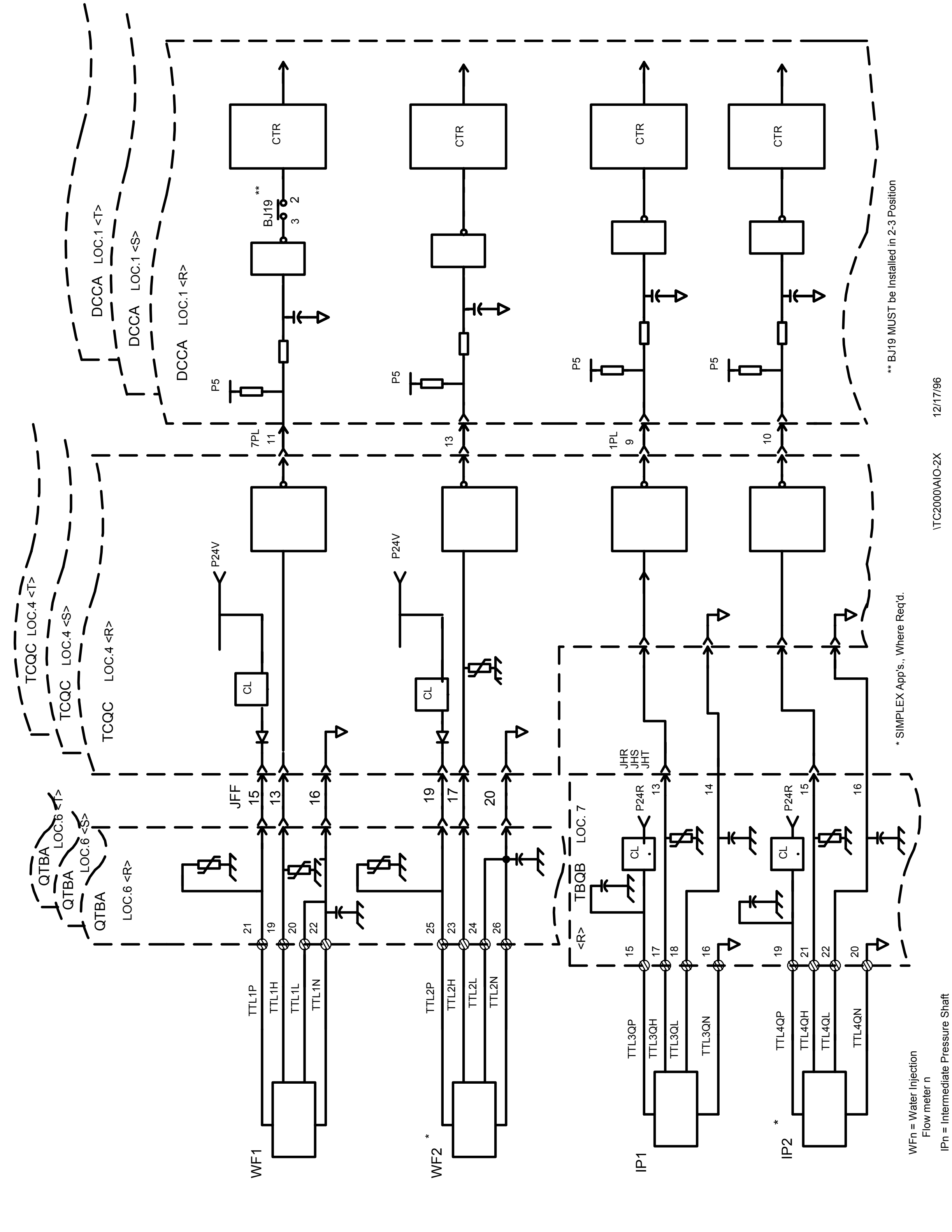
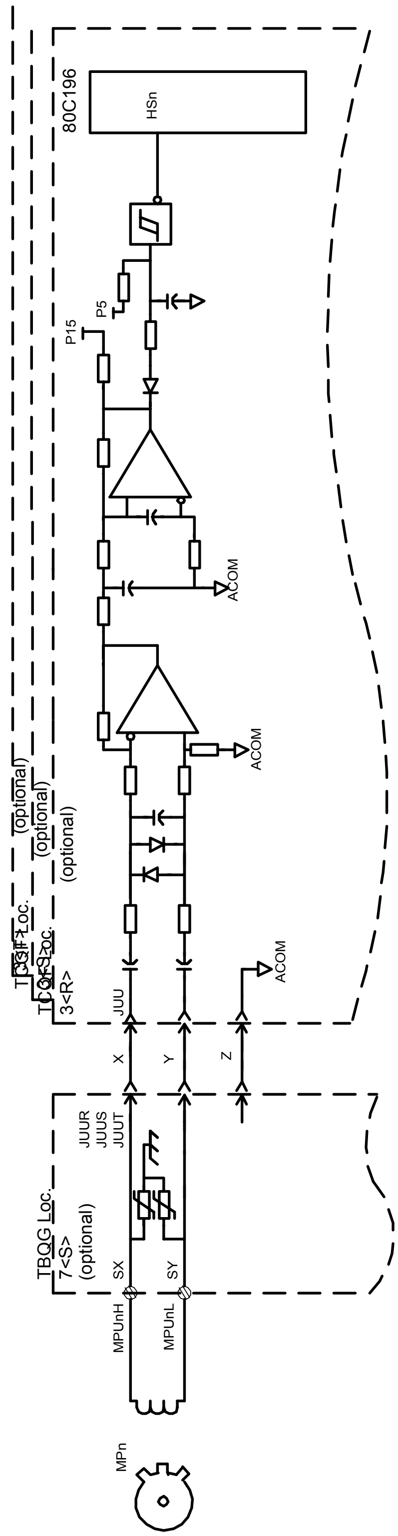


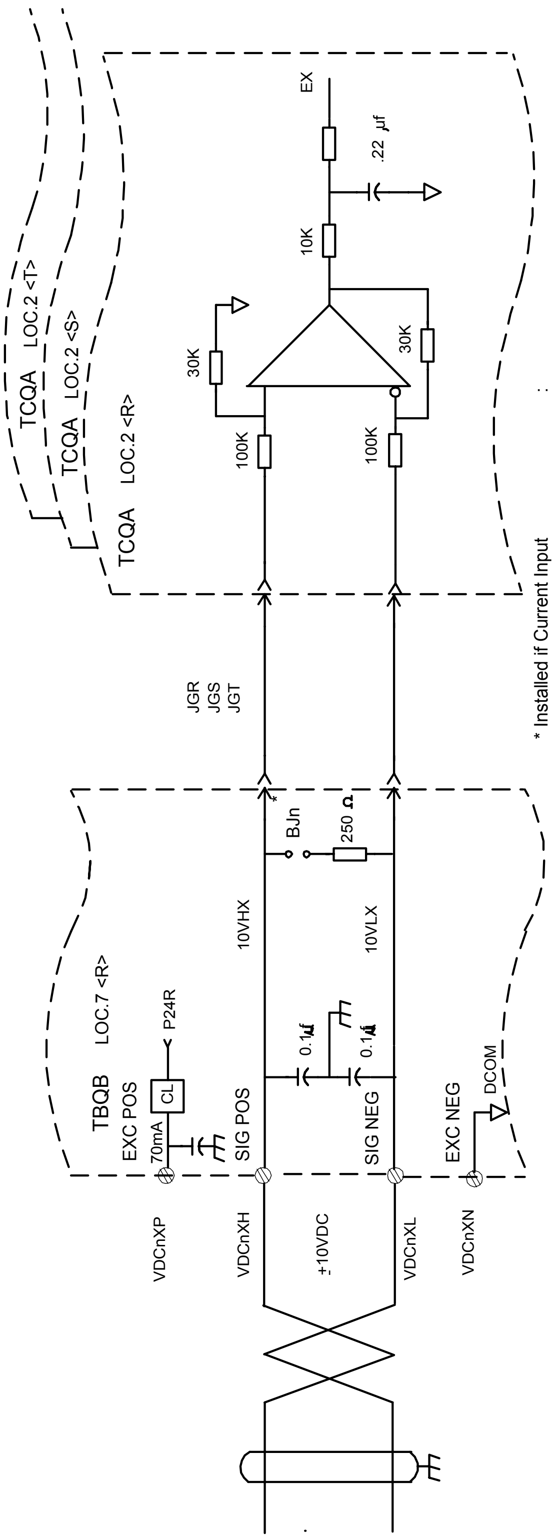
Figure D-39. <Q> Core - Pulse Rate Inputs on QTBA Terminal Board



MPn	MPUnH	MPUnL	SX	SY	JUU X	JUU Y	JUU Z	HSn
MP1	MPU8H	MPU8L	45	46	35	36	37	HS13
MP2	MPU9H	MPU9L	47	48	38	39	40	HS12
MP3	MPU10H	MPU10L	49	50	41	42	43	HS11
MP4	MPU11H	MPU11L	51	52	44	45	46	HS10

\TC2000\RWM_PR 6/27/97

Figure D-40. <Q> Core - Pulse Rate Inputs on TBQC Terminal Board Using TCQF

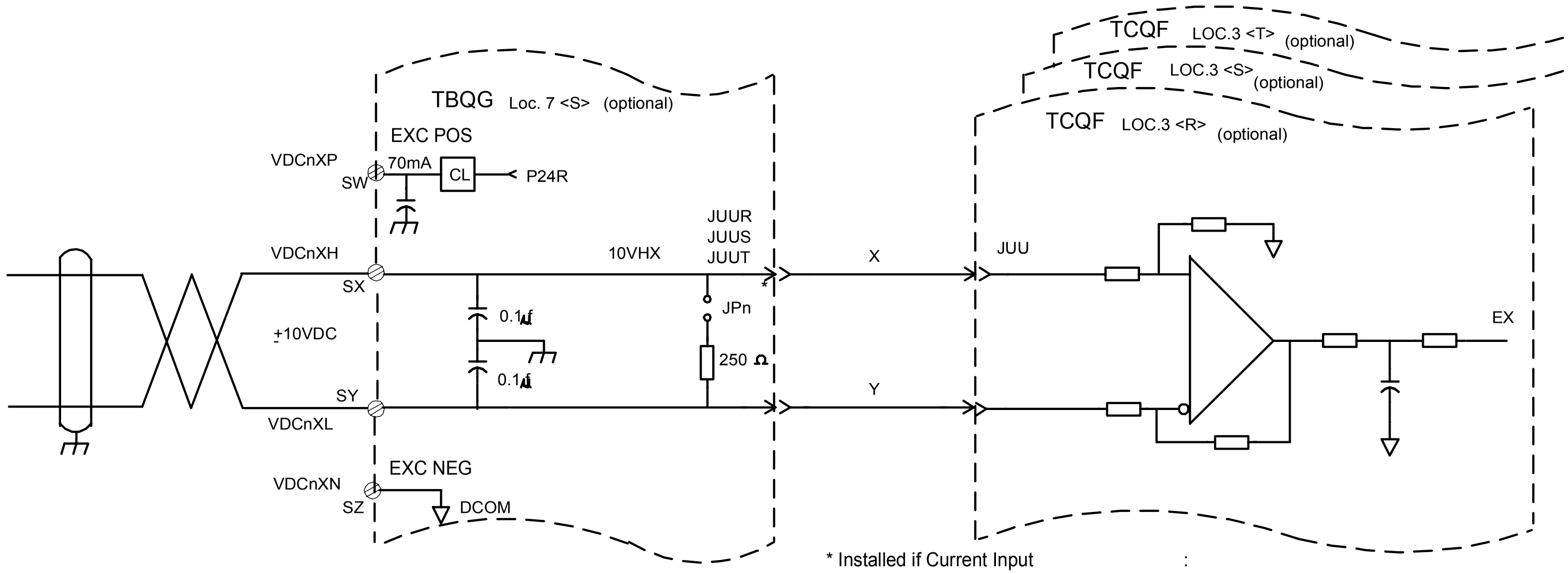


* Installed if Current Input :
Not Installed if Voltage Input;
These Hardware Jumper Settings are Unit-Specific

DEVICE	TERMINATION, TBQB				BJn *	JG CONNECT PIN	COMMENT
	SIGNAL		EXCITATION				
TB DESIGNATION	POS	NEG	POS	NEG	JUMPERS	SIGNAL	
VDC1_R	25	26	23	24	BJ8	JGR- 1, 2	TO R ONLY (TMR) , FUEL PRESS GAS 1
VDC1_S	29	30	27	28	BJ9	JGS- 1, 2	TO S ONLY (TMR) , FUEL PRESS GAS 1
VDC1_T	33	34	31	32	BJ10	JGT- 1, 2	TO T ONLY (TMR) , FUEL PRESS GAS 1
VDC2_R	37	38	35	36	BJ11	JGR- 3, 4	TO R ONLY (TMR) , FUEL PRESS GAS 2
VDC2_S	41	42	39	40	BJ12	JGS- 3, 4	TO S ONLY (TMR) , FUEL PRESS GAS 2
VDC2_T	45	46	43	44	BJ13	JGT- 3, 4	TO T ONLY (TMR) , FUEL PRESS GAS 2
VDC3	47	48	-	-	BJ14	JGR/S/T- 5, 6	TO R,S,T (FANOUT)
VDC4	49	50	-	-	BJ15	JGR/S/T- 7, 8	TO R,S,T (FANOUT)

\TC2000\AIO-6X 12-17-96

Figure D-41. <Q> Core - Analog Current and Voltage Inputs on TBQB Terminal Board



* Installed if Current Input :
Not Installed if Voltage Input;
These Hardware Jumper Settings are Unit-Specific

VDCnXH	VDCnXL	TB POINT		JUU		VDCnXP	VDCnXN	TB POINT	
		SX	SY	X	Y			SW	SZ
VDC5XH VDC6XH	VDC5XL VDC6XL	35 37	36 38	25 27	26 28	VDC5XP	VDC5XN	33	34
VDC7XH VDC8XH	VDC7XL VDC8XL	41 43	42 44	29 31	30 32	VDC7XP	VDC7XN	39	40

Analog Current and Voltage Inputs for TCQF/TBQG

\TC2000\RWM_10 9/01/97

Figure D-42. <Q> Core - Analog Current and Voltage Inputs on TBQG Terminal Board Using TCQF

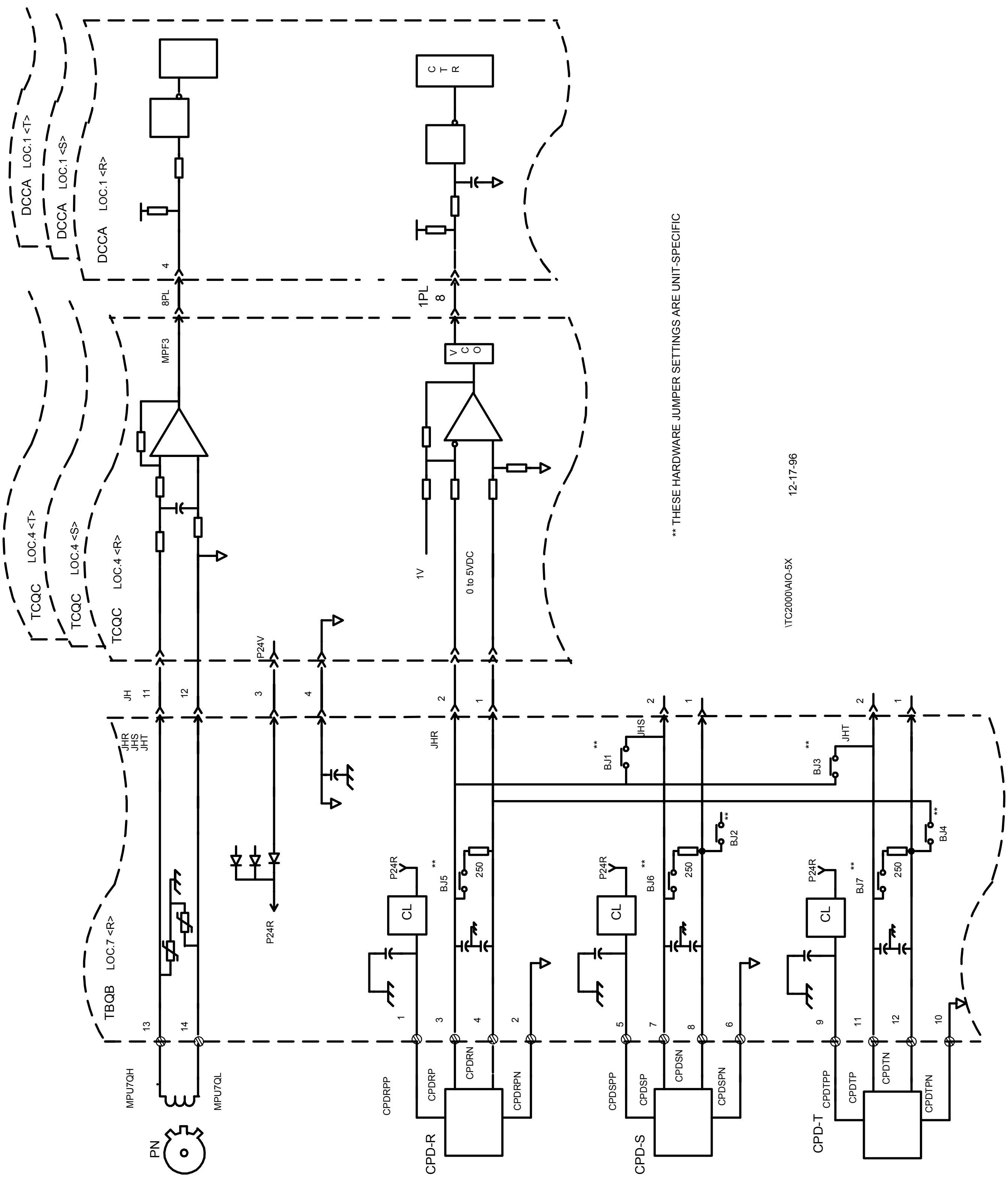
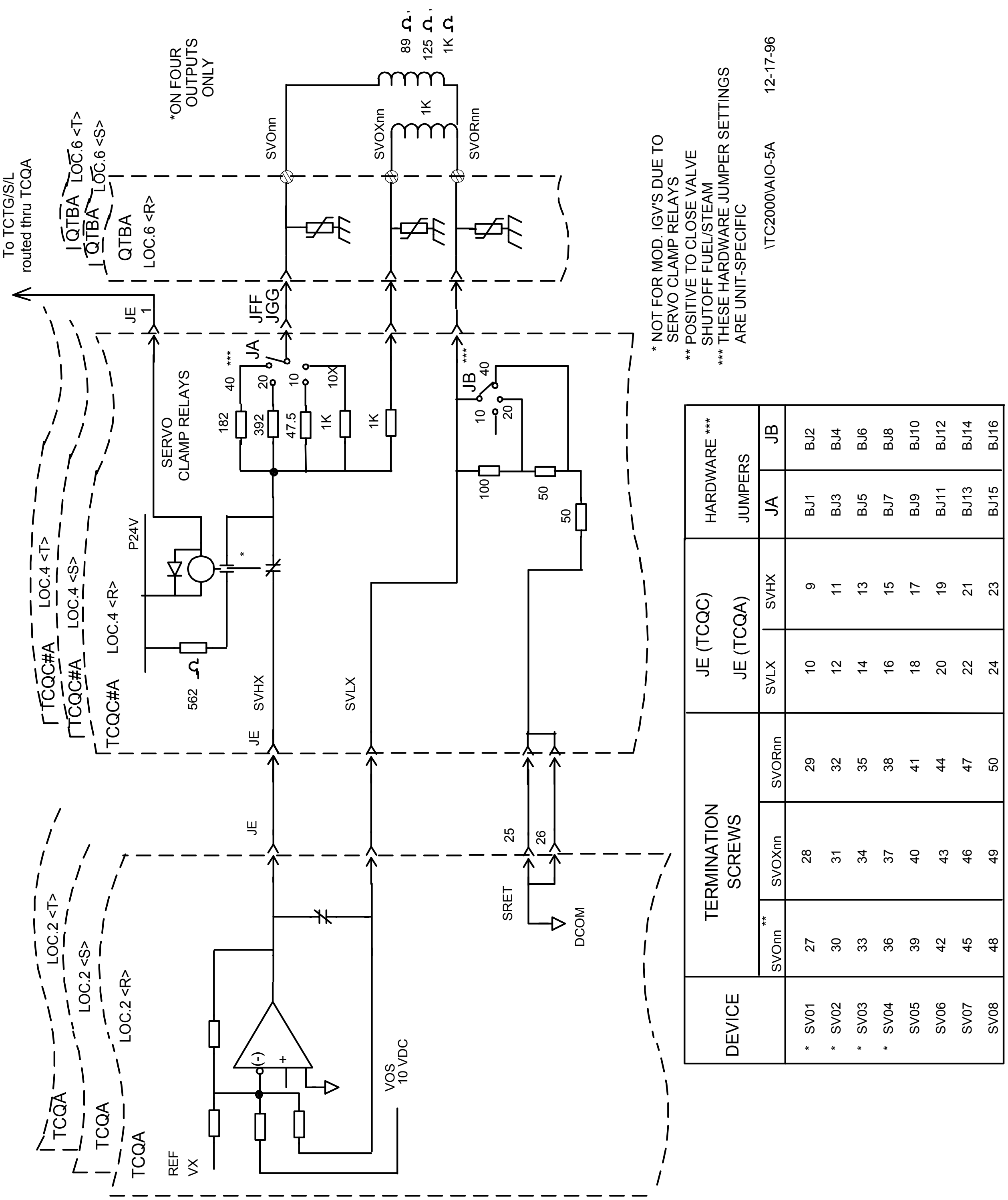
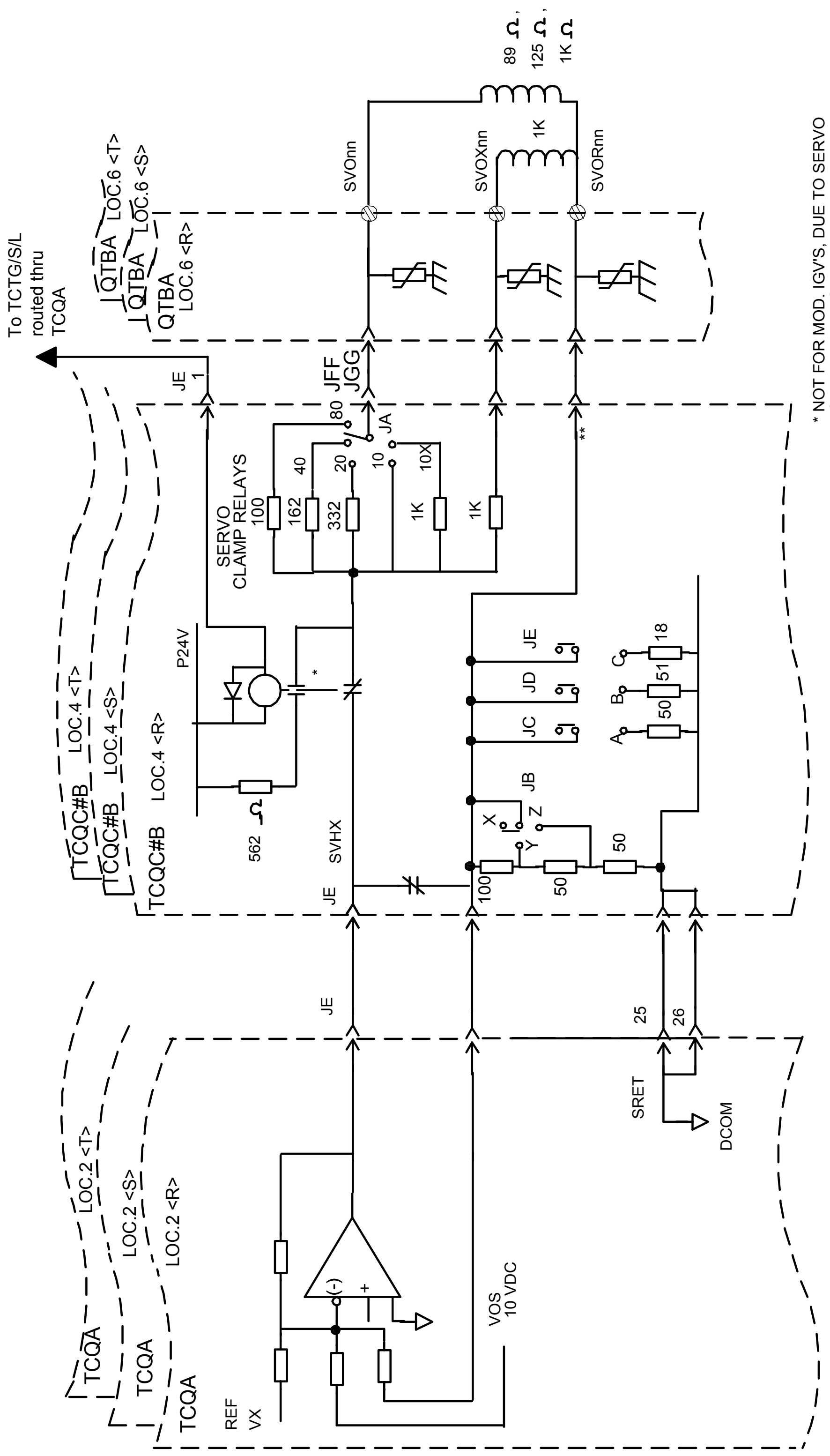


Figure D-43. <Q> Core - Compressor Discharge Pressure Inputs on TBQB



DEVICE	TERMINATION SCREWS			JE (TCQC) JE (TCQA)		HARDWARE *** JUMPERS	
	SVOnn **	SVOXnn	SVORnn	SVLX	SVHX	JA	JB
* SV01	27	28	29	10	9	BJ1	BJ2
* SV02	30	31	32	12	11	BJ3	BJ4
* SV03	33	34	35	14	13	BJ5	BJ6
* SV04	36	37	38	16	15	BJ7	BJ8
SV05	39	40	41	18	17	BJ9	BJ10
SV06	42	43	44	20	19	BJ11	BJ12
SV07	45	46	47	22	21	BJ13	BJ14
SV08	48	49	50	24	23	BJ15	BJ16

Figure D-44. <Q> Core - 3-Coil Servo Outputs on QTBA Terminal Board for TCQC G#A

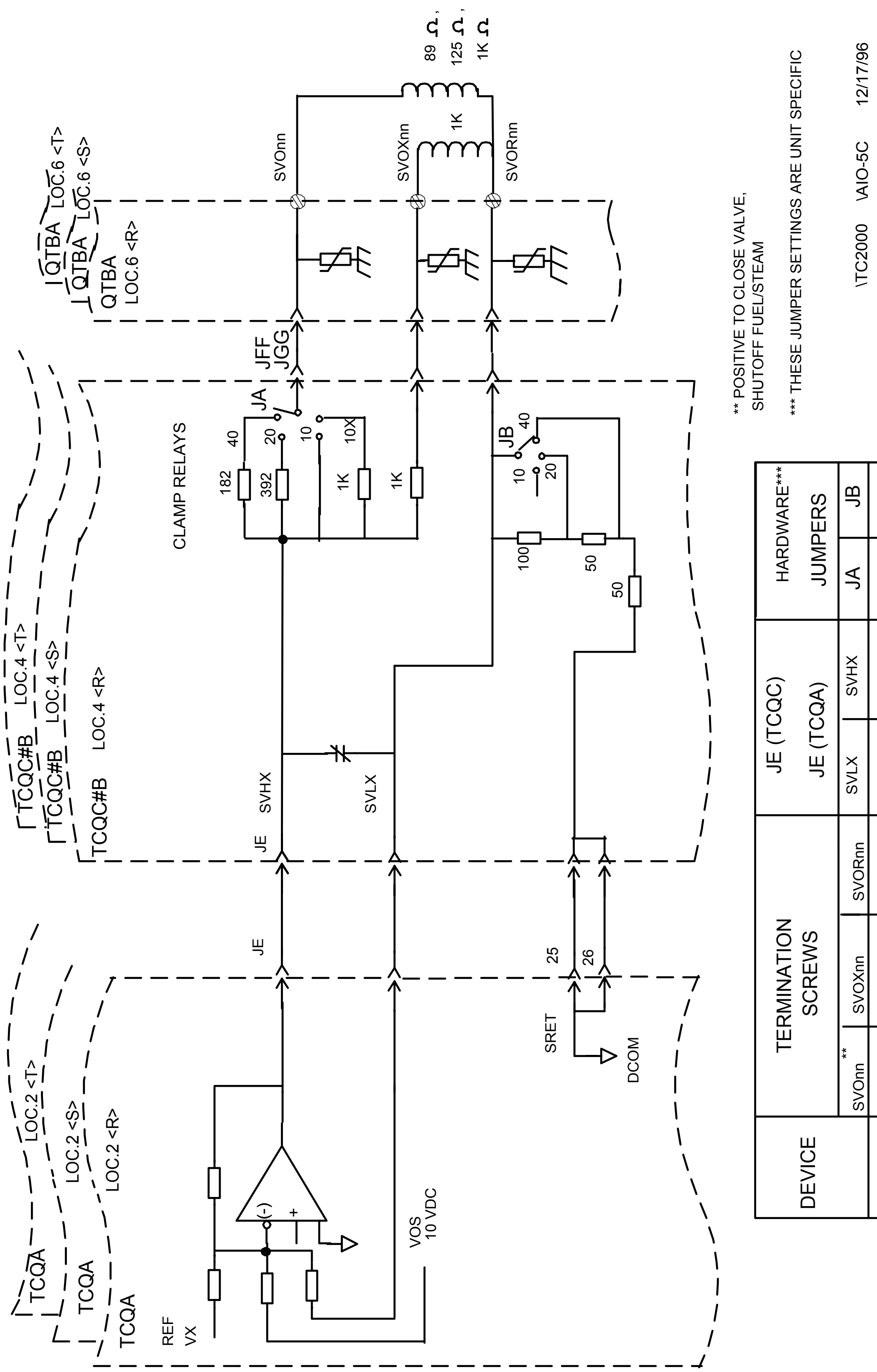


DEVICE	TERMINATION SCREWS			JE (TCQC) JE (TCQA)		HARDWARE*** JUMPERS				
	SVOnn **	SVOXnn	SVORnn	SVLX	SVHX	JA	JB	JC	JD	JE
* SV01	27	28	29	10	9	BJ1	BJ2	BJ25	BJ26	BJ27
* SV02	30	31	32	12	11	BJ3	BJ4	BJ28	BJ29	BJ30
* SV03	33	34	35	14	13	BJ5	BJ6	BJ31	BJ32	BJ33
* SV04	36	37	38	16	15	BJ7	BJ8	BJ34	BJ35	BJ36

* NOT FOR MOD. IGV'S, DUE TO SERVO CLAMP RELAYS
** POSITIVE TO CLOSE VALVE, SHUTOFF FUEL/STEAM
*** THESE JUMPERS ARE UNIT-SPECIFIC

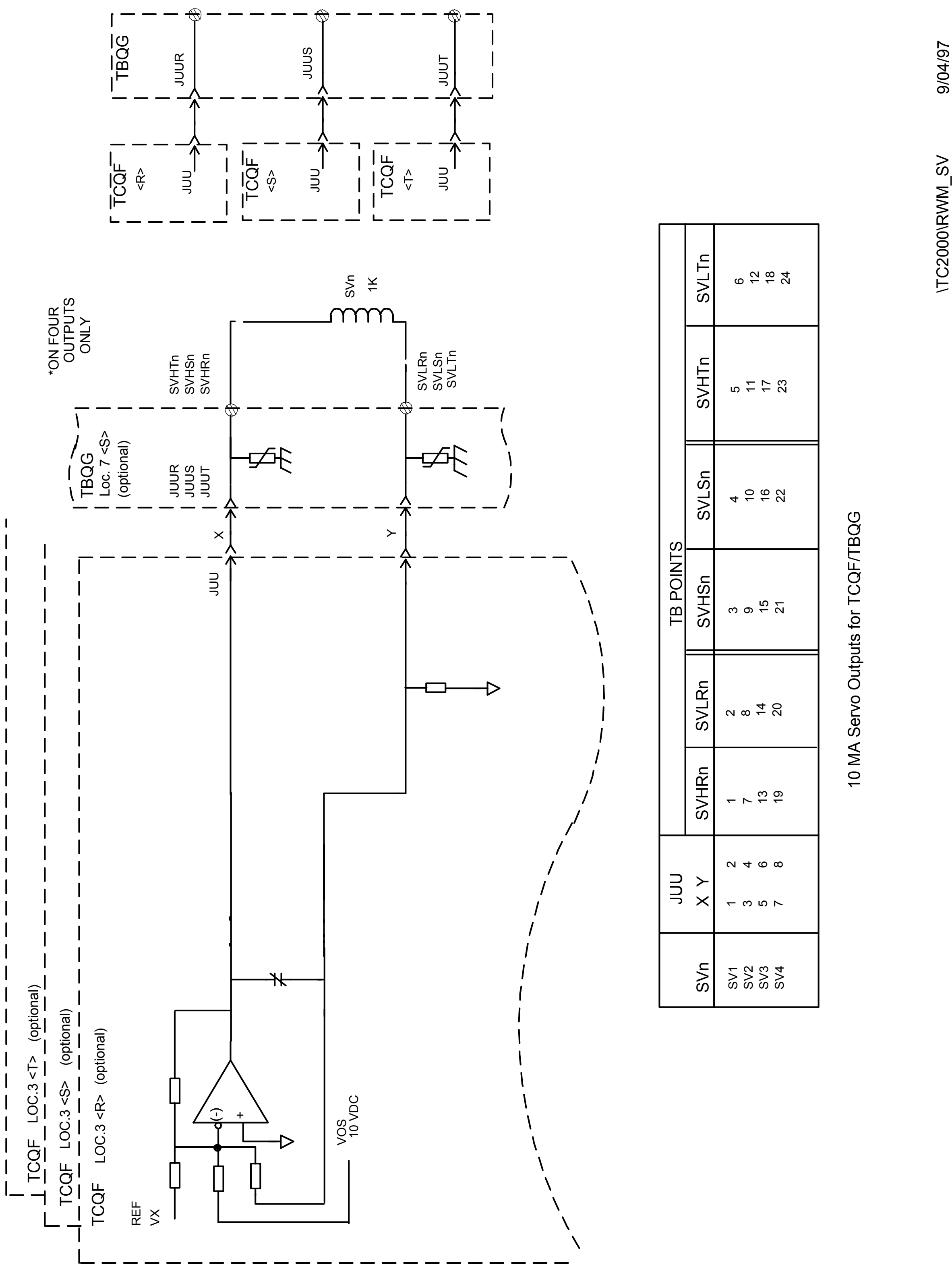
\TC2000\AIO-5B 12/17/96

Figure D-45. <Q> Core - 3-Coil Servo Outputs on QTBA Terminal Board for TCQC G#B



DEVICE	TERMINATION SCREWS			JE (TCQC) JE (TCQA)		HARDWARE*** JUMPERS	
	SVOnn **	SVOXnn	SVORnn	SVLX	SVHX	JA	JB
SV05	39	40	41	18	17	BJ9	BJ10
SV06	42	43	44	20	19	BJ11	BJ12
SV07	45	46	47	22	21	BJ13	BJ14
SV08	48	49	50	24	23	BJ15	BJ16

Figure D-46. <Q> Core - 3-Coil Servo Outputs on QTBA Terminal Board for TCQC G#B



SVn	JUU X	JUU Y	SVHRn	SVLRn	SVHSn	SVLSn	SVHTn	SVLTn
			TB POINTS					
SV1	1	2	1	2	3	4	5	6
SV2	3	4	7	8	9	10	11	12
SV3	5	6	13	14	15	16	17	18
SV4	7	8	19	20	21	22	23	24

Figure D-47. <Q> Core - 3-Coil Servo Outputs on TBQG Terminal Board Using TCQF

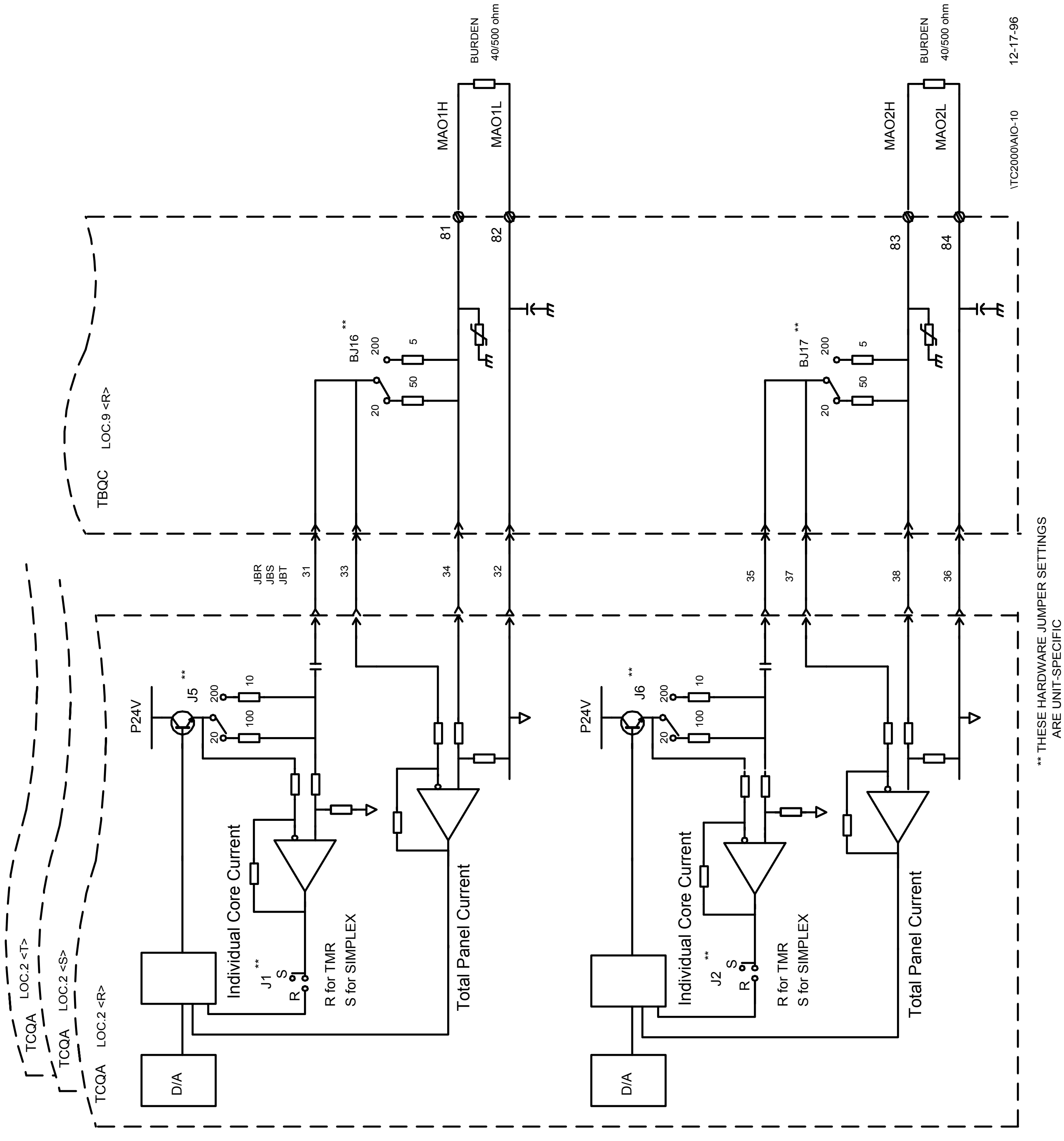


Figure D-48. <Q> Core - 4-20 mA Outputs on TBQC Terminal Board

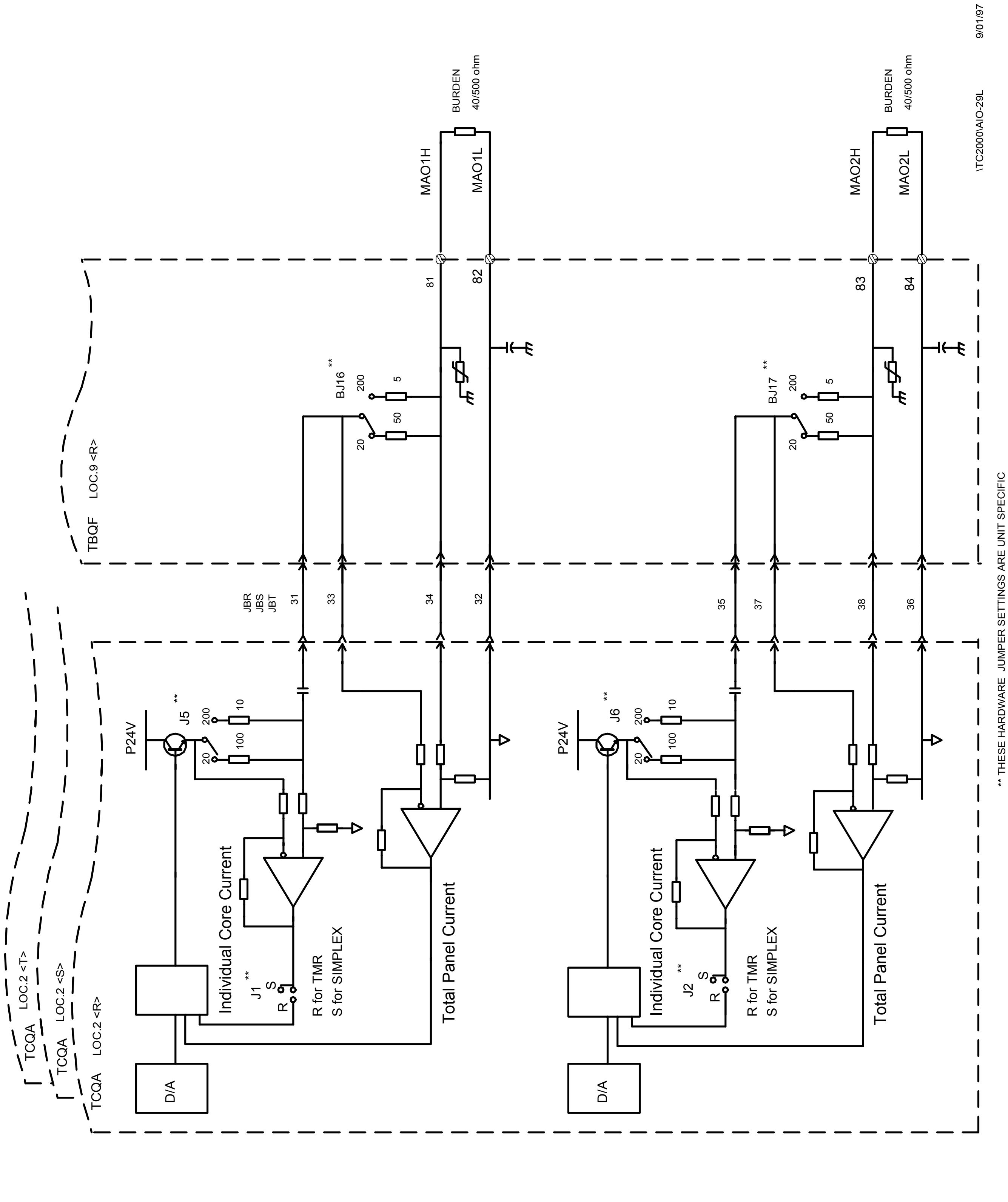


Figure D-49. <Q> Core - 4-20/200 mA Outputs on TBQF Terminal Board

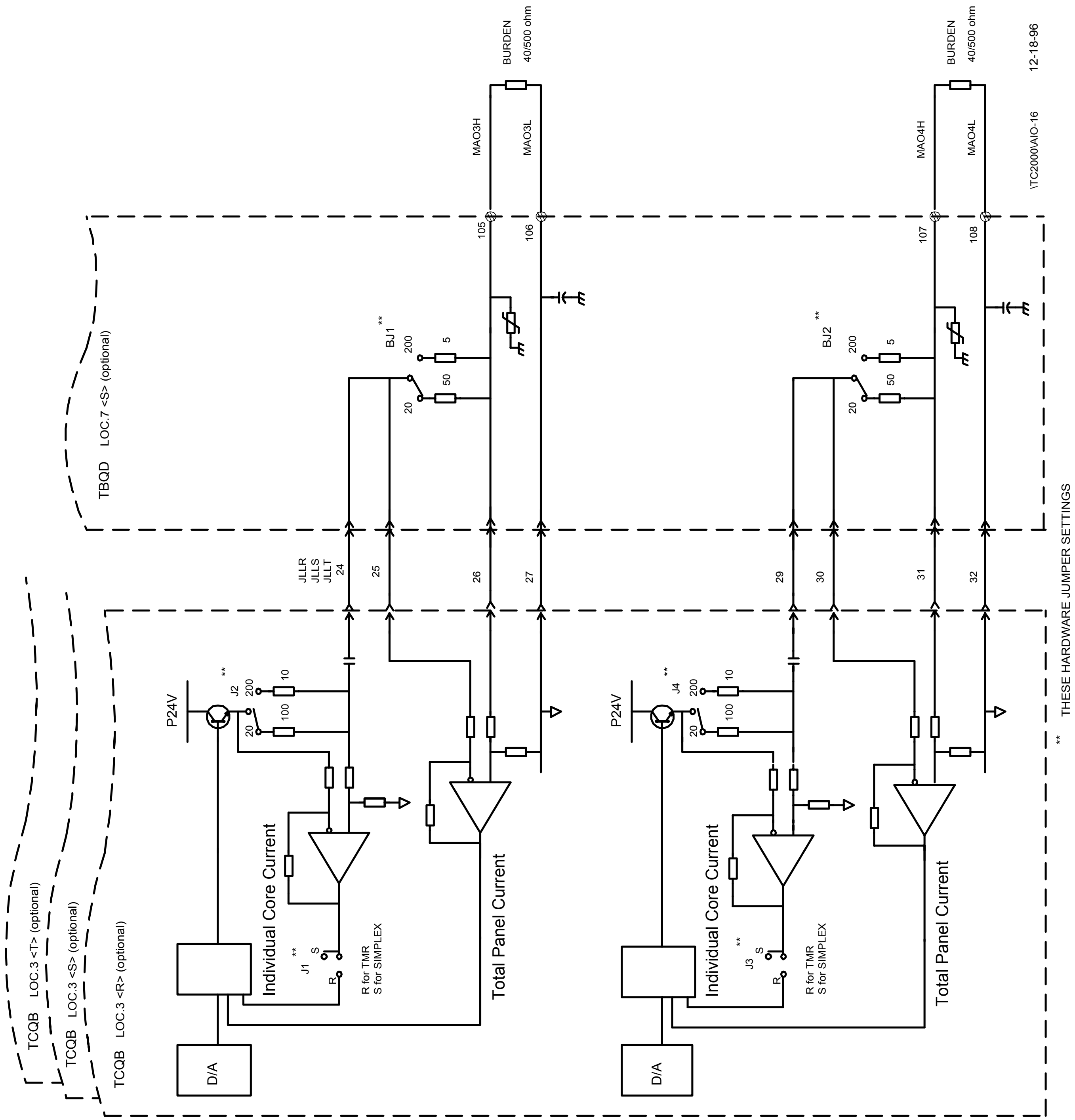


Figure D-50. <Q> Core - 4-20/200 mA Outputs on TBQD Terminal Board

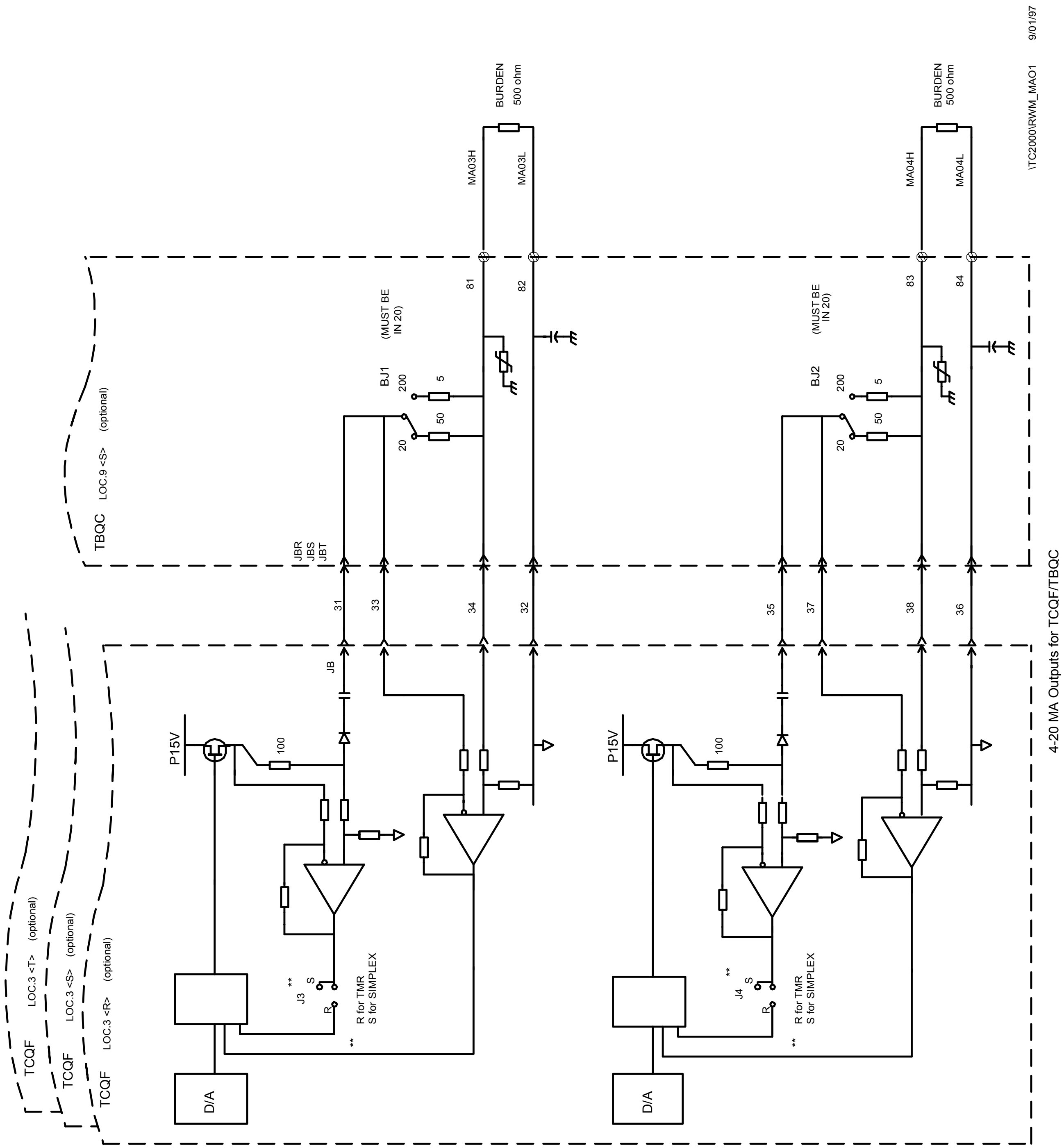
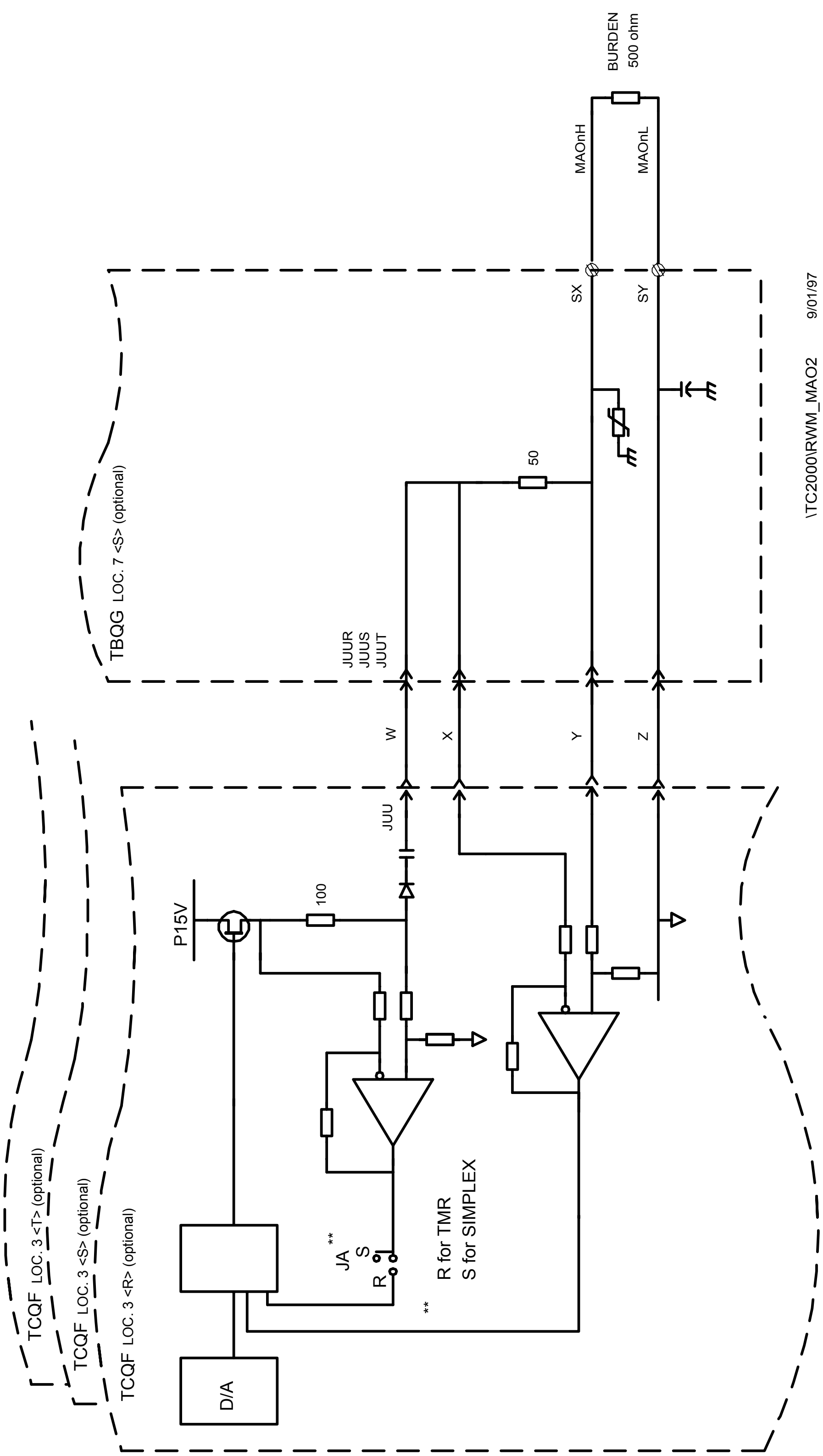


Figure D-51. <Q> Core - 4-20 mA Outputs on TBQC Terminal Board Using TCQF



MAOnH	MAOnL	TB POINT		JUU				HARDWARE JUMPERS
		SX	SY	W	X	Y	Z	JA
MAO5H	MAO5L	25	26	9	10	11	12	J5
MAO6H	MAO6L	27	28	13	14	15	16	J6
MAO7H	MAO7L	29	30	17	18	19	20	J7
MAO8H	MAO8L	31	32	21	22	23	24	J8

4-20 MA Outputs TCQF/TBQG

Figure D-52. <Q> Core - 4-20 mA Outputs on TBQG Terminal Board Using TCQF

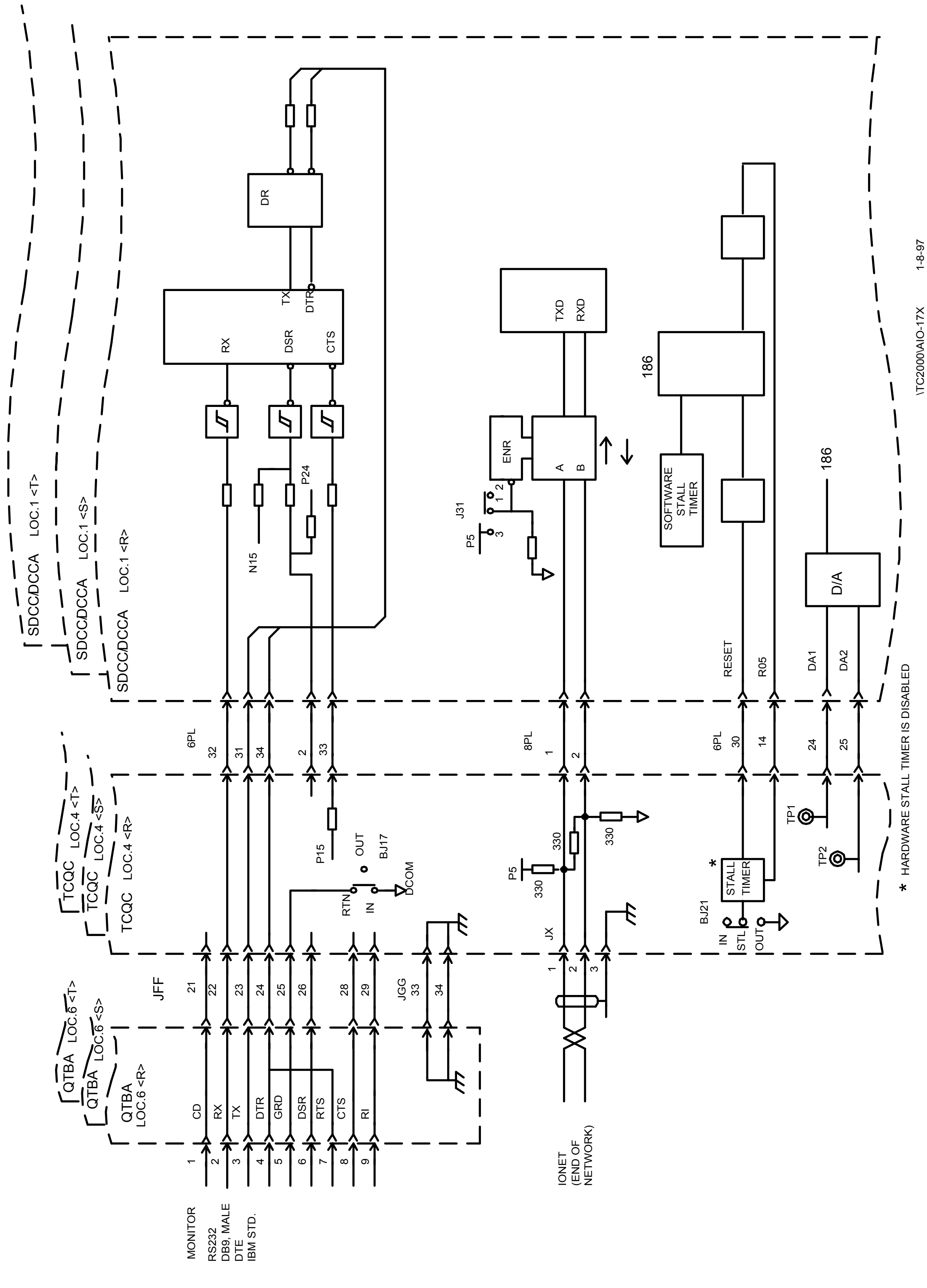


Figure D-53. <Q> Core - TIMN Monitor Connections on QTBA Terminal Board

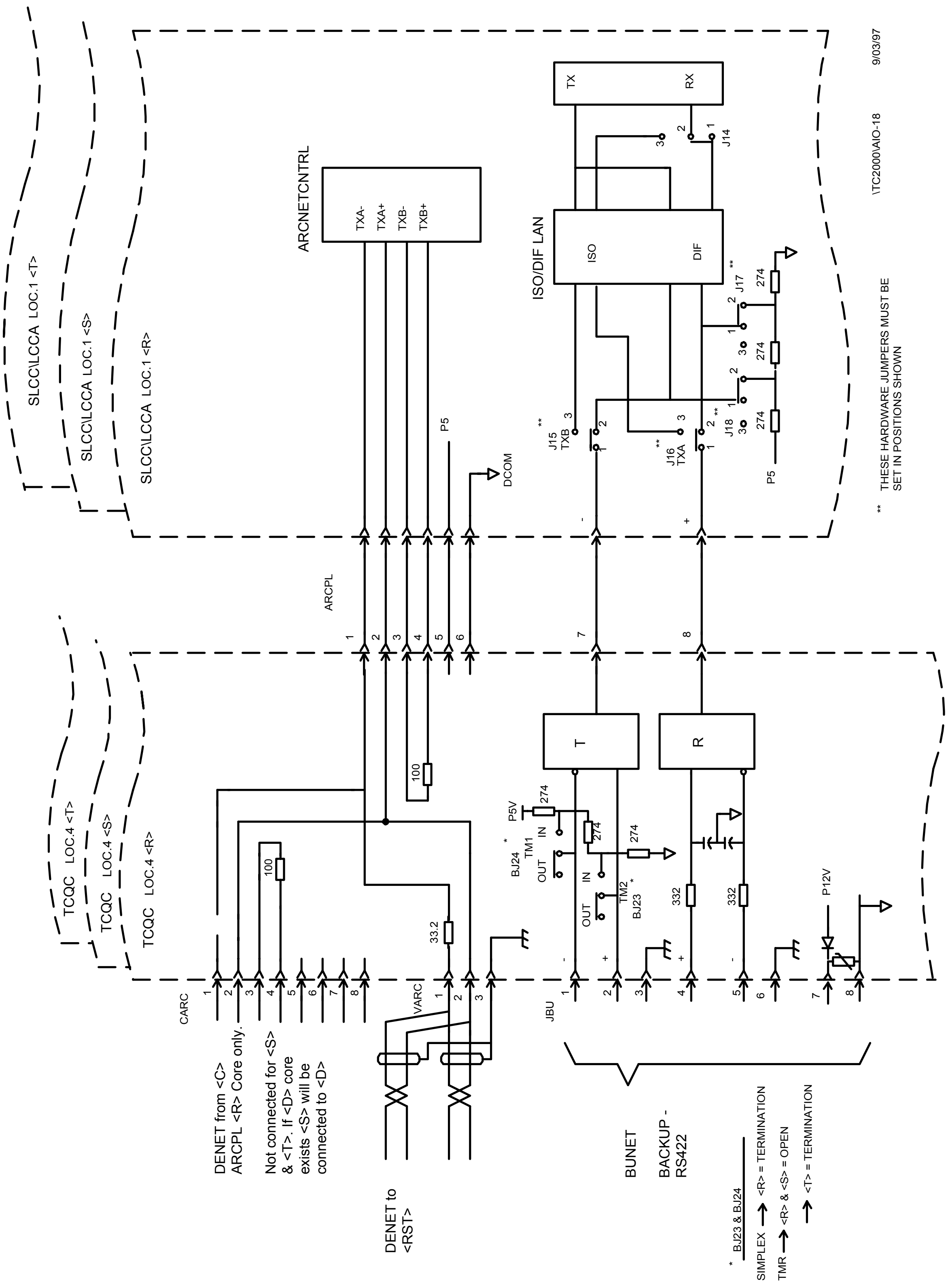


Figure D-54. <Q> Core - DENet and BUNet Connections on TCQC Terminal Board

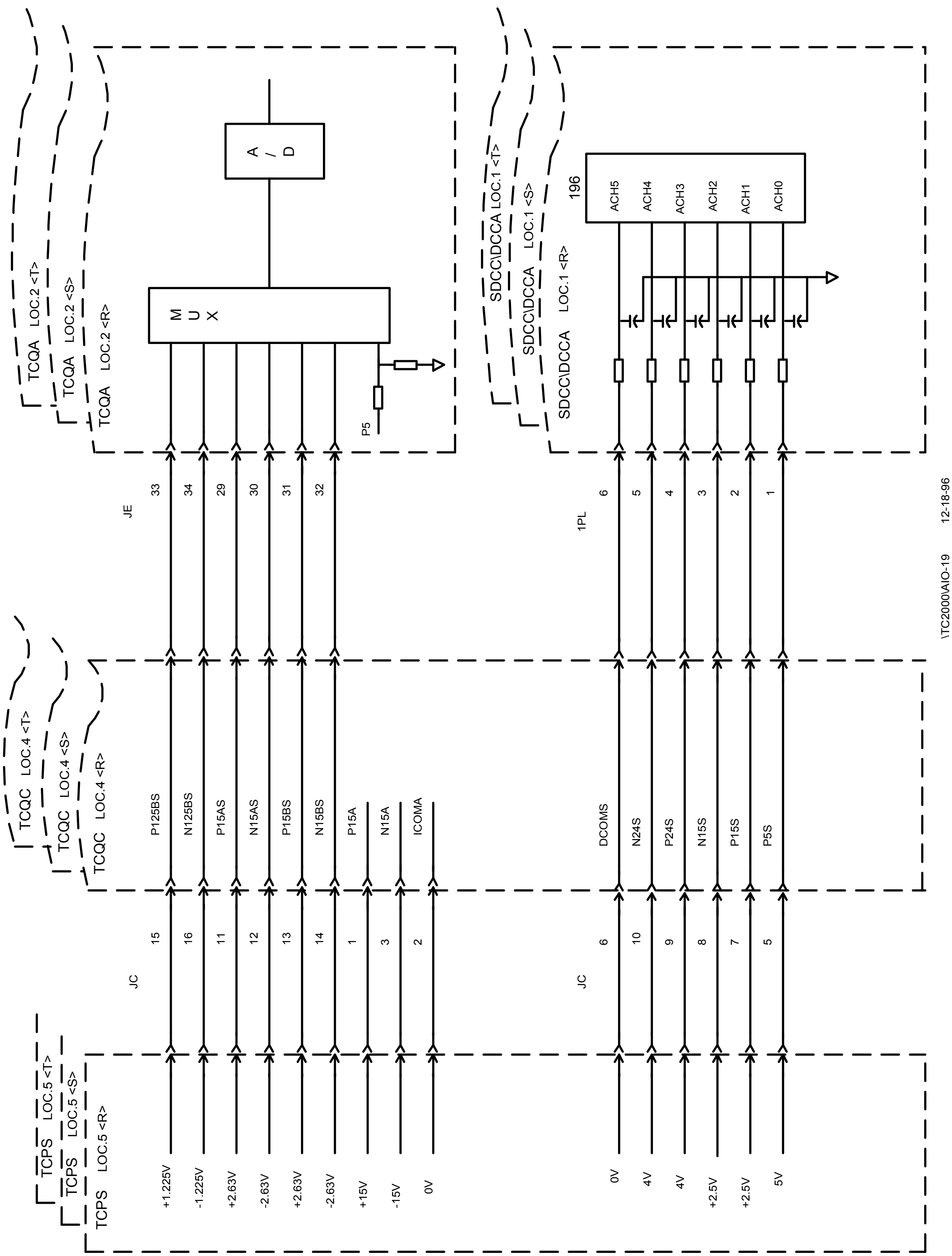


Figure D-55. <Q> Core - Power Supply Monitoring

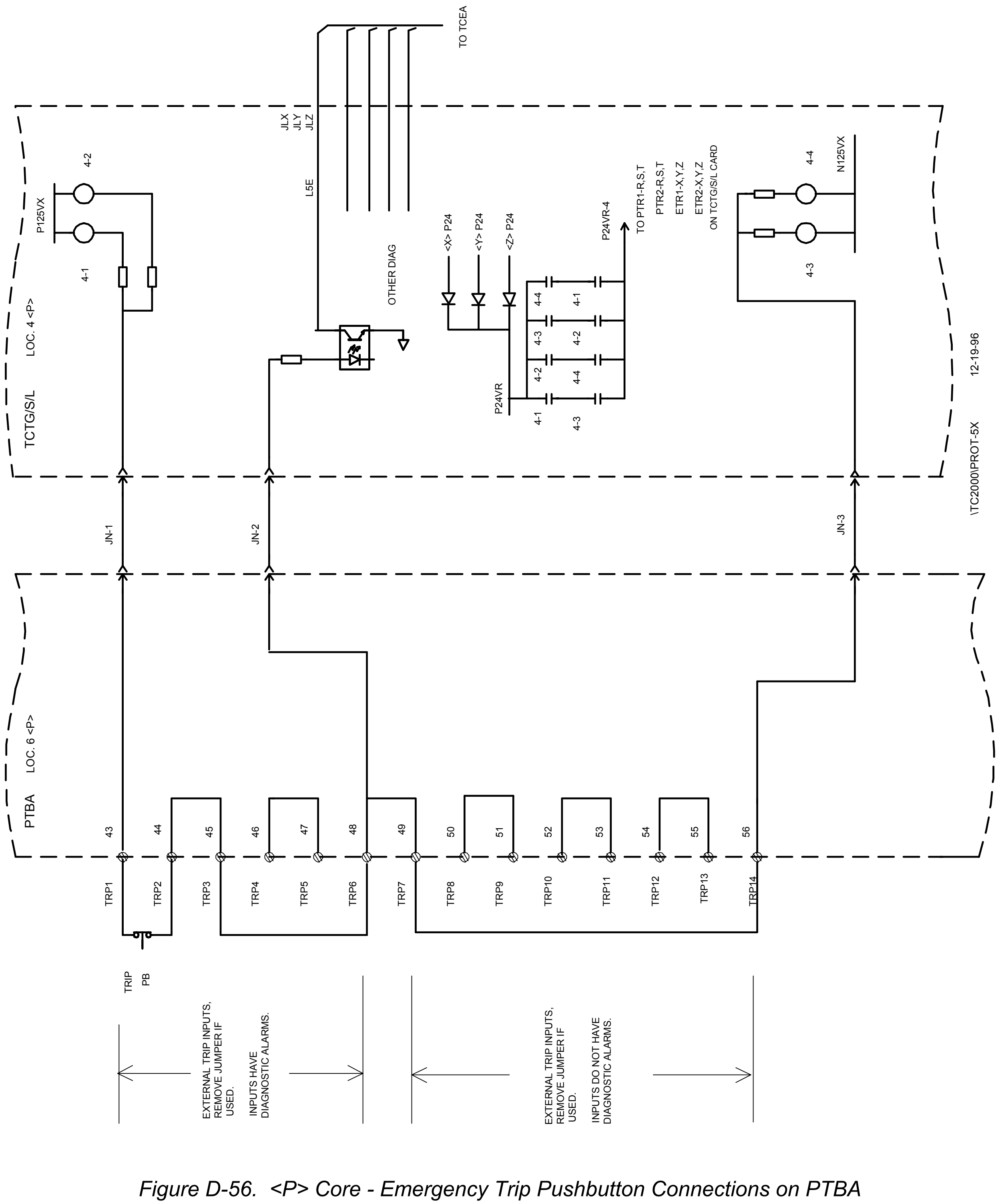


Figure D-56. <P> Core - Emergency Trip Pushbutton Connections on PTBA

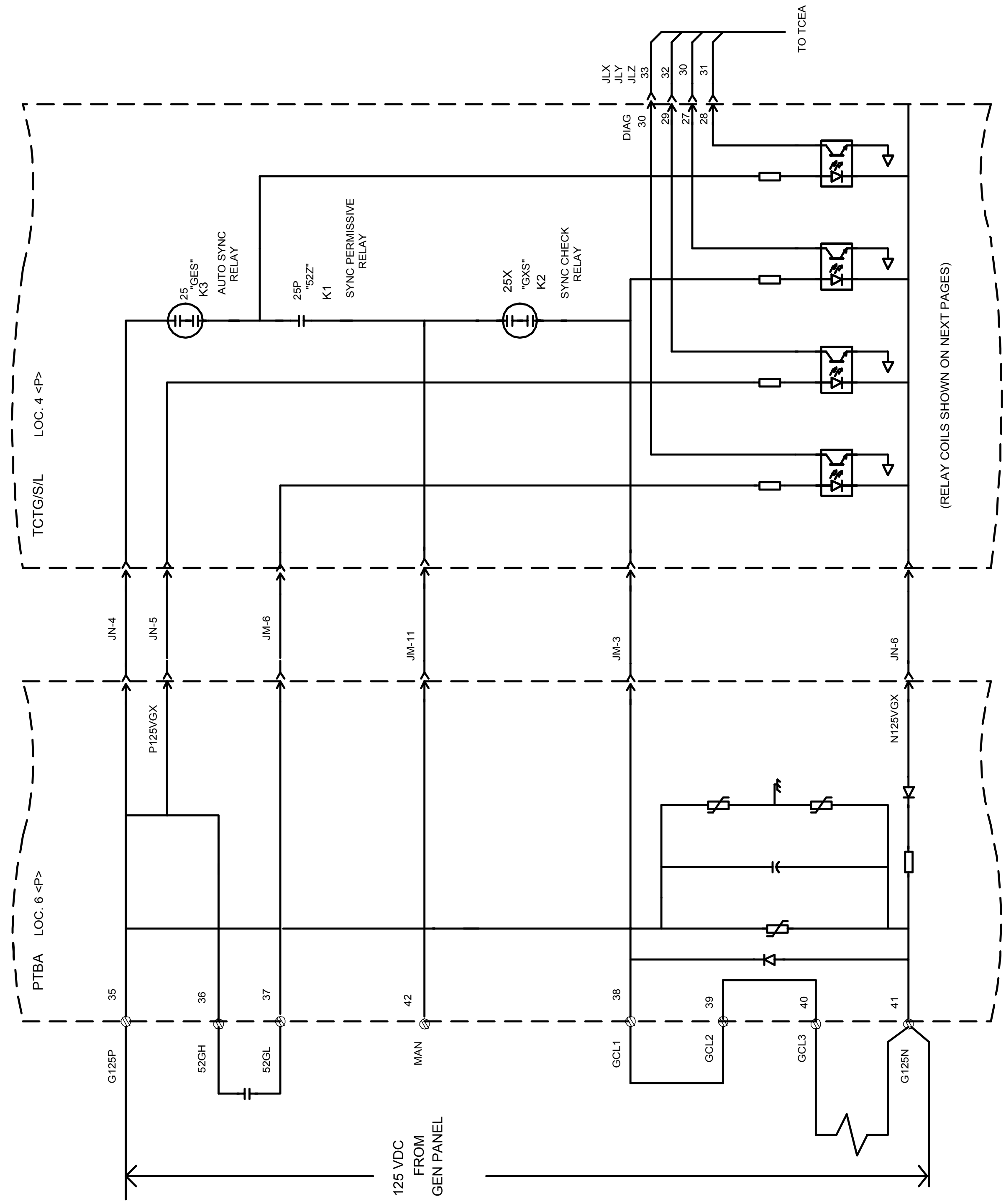


Figure D-57. <P> Core - Circuit Breaker (52G) Close Circuit

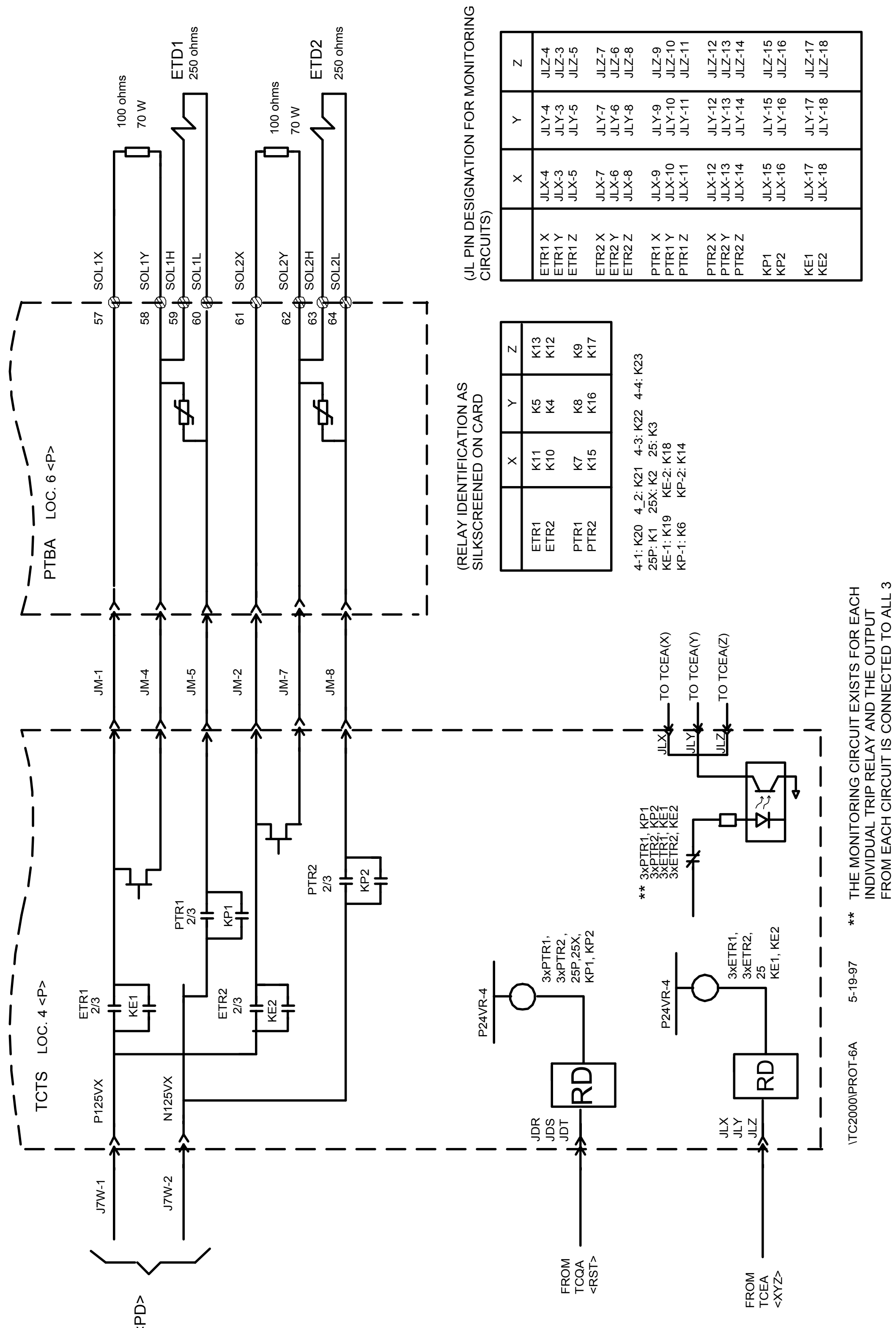
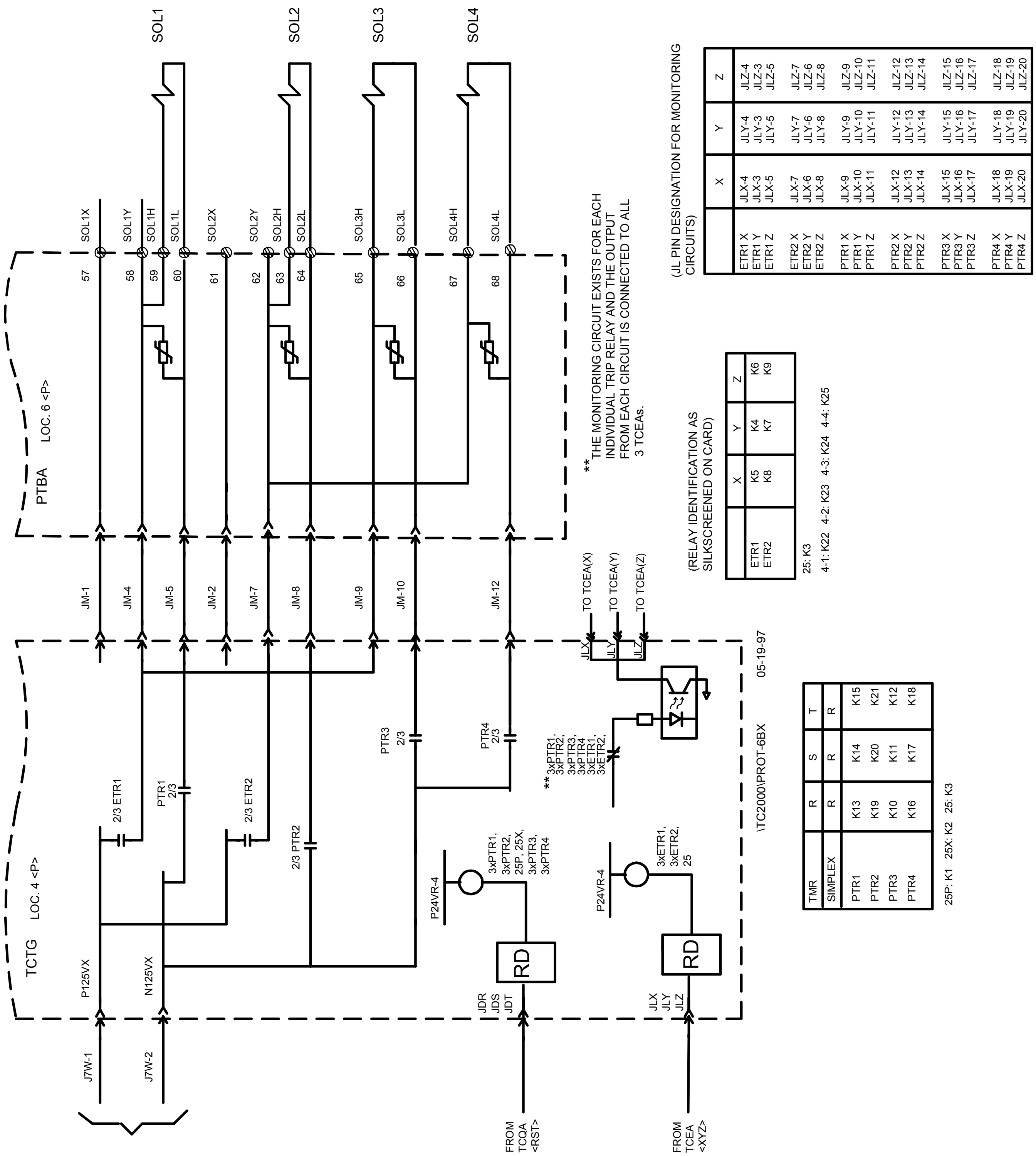


Figure D-58. <P> Core - Emergency Trip Circuit - Medium Steam Turbine



(JL PIN DESIGNATION FOR MONITORING CIRCUITS)

	X	Y	Z
ETR1 X	JLX-4	JLY-4	JLZ-4
ETR1 Y	JLX-3	JLY-3	JLZ-3
ETR1 Z	JLX-5	JLY-5	JLZ-5
ETR2 X	JLX-7	JLY-7	JLZ-7
ETR2 Y	JLX-6	JLY-6	JLZ-6
ETR2 Z	JLX-8	JLY-8	JLZ-8
PTR1 X	JLX-9	JLY-9	JLZ-9
PTR1 Y	JLX-10	JLY-10	JLZ-10
PTR1 Z	JLX-11	JLY-11	JLZ-11
PTR2 X	JLX-12	JLY-12	JLZ-12
PTR2 Y	JLX-13	JLY-13	JLZ-13
PTR2 Z	JLX-14	JLY-14	JLZ-14
PTR3 X	JLX-15	JLY-15	JLZ-15
PTR3 Y	JLX-16	JLY-16	JLZ-16
PTR3 Z	JLX-17	JLY-17	JLZ-17
PTR4 X	JLX-18	JLY-18	JLZ-18
PTR4 Y	JLX-19	JLY-19	JLZ-19
PTR4 Z	JLX-20	JLY-20	JLZ-20

(RELAY IDENTIFICATION AS SILKSCREENED ON CARD)

	X	Y	Z
ETR1	K5	K4	K6
ETR2	K8	K7	K9

25: K3

4-1: K22 4-2: K23 4-3: K24 4-4: K25

TMR	R	S	T
SIMPLEX	R	R	R
PTR1	K13	K14	K15
PTR2	K19	K20	K21
PTR3	K10	K11	K12
PTR4	K16	K17	K18

25P: K1 25X: K2 25: K3

Figure D-59. <P> Core - Emergency Trip Circuit - Gas Turbine

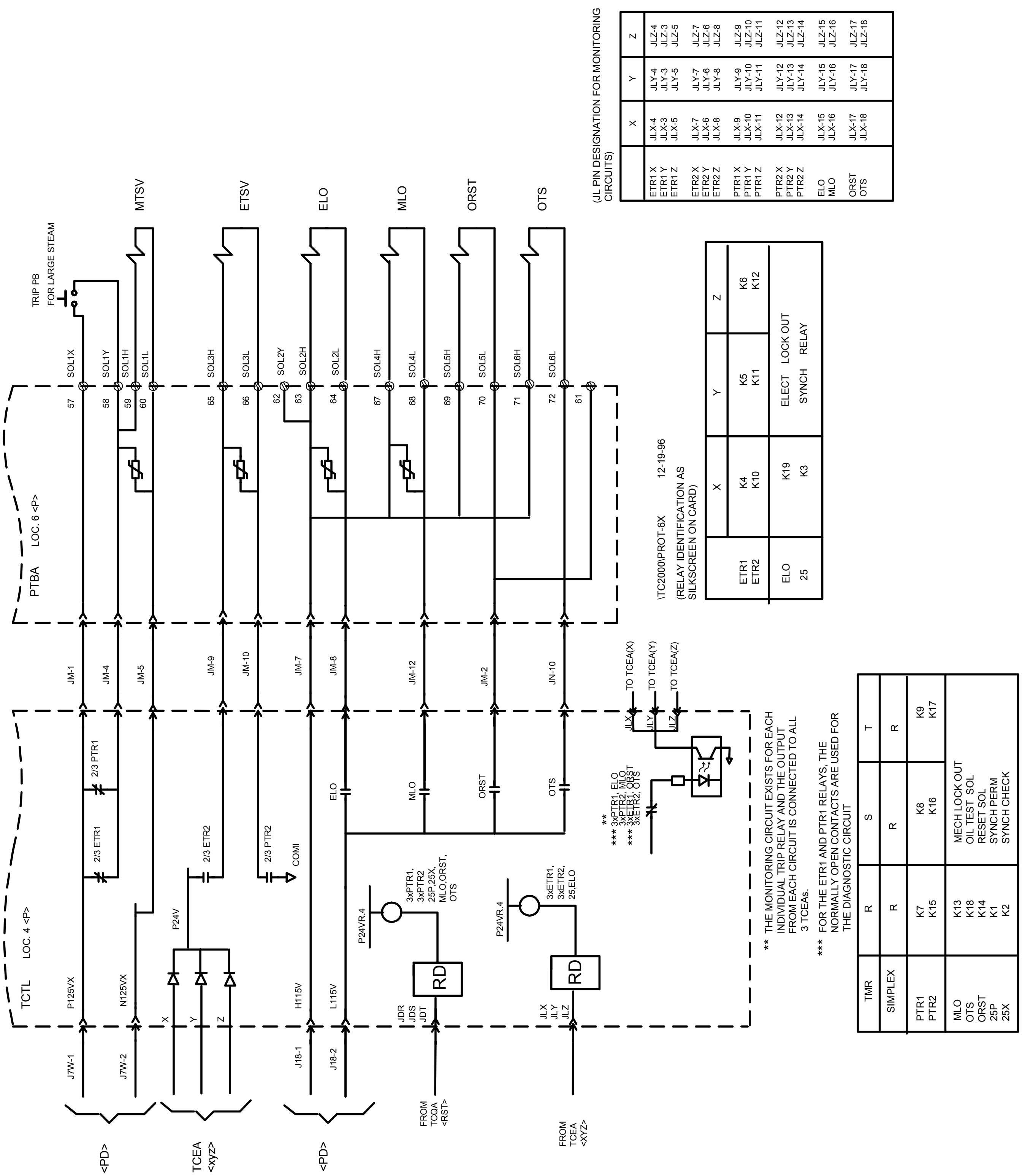


Figure D-60. <P> Core - Emergency Trip Circuit - Large Steam Turbine

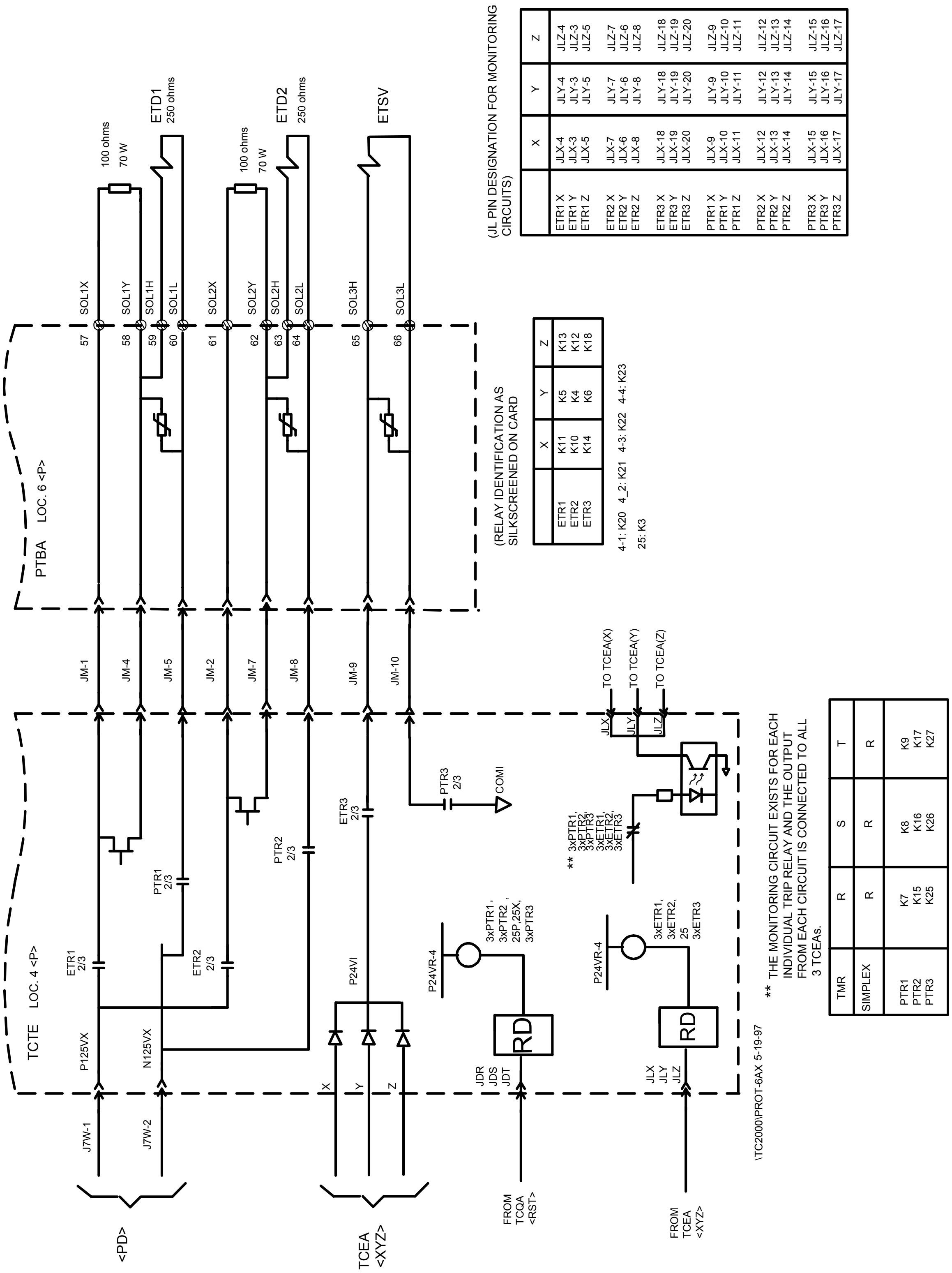


Figure D-61. <P> Core - Emergency Trip Extended Circuit - Large/Medium Steam Turbine

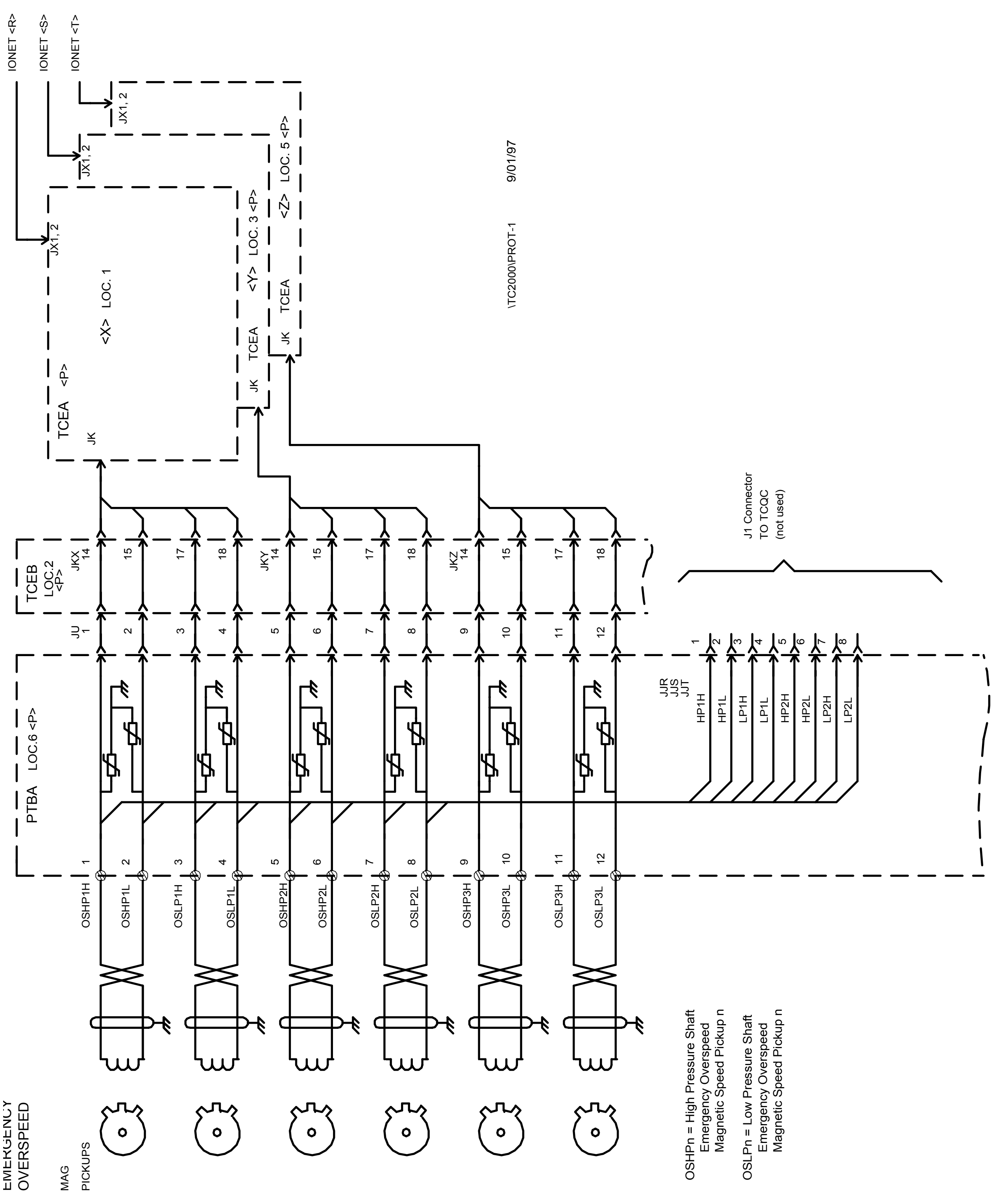
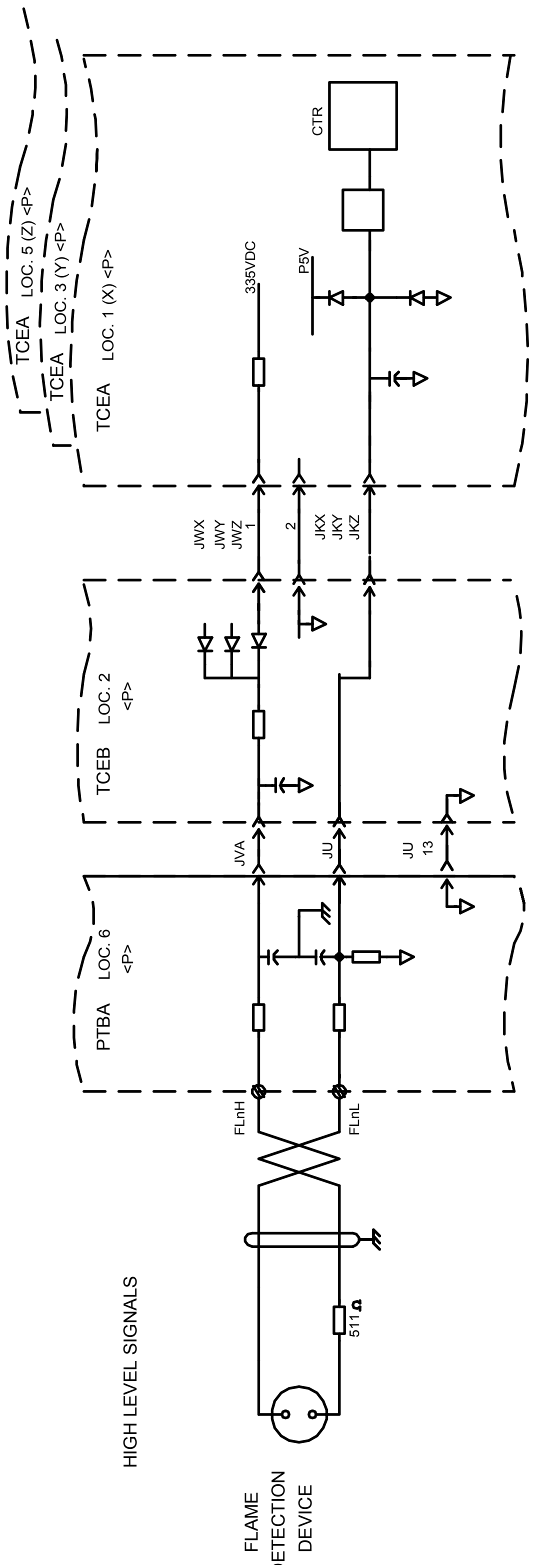


Figure D-62. <P> Core - Emergency Overspeed Magnetic Pickups on PTBA



\TC2000\PROT-2 12-19-96

DEVICE TB DESIGNATION	TERMINATION SCREW	TCEB PIN	TCEA PIN	<I> SIGNAL NAMES: FLAME INTENSITY HZ	THRESHOLD TCE_CONFIG_L28_HI/LO: HIGH	LOW	FLAME LOGIC
FL1H	13	JVA-1	-				
FL1L	14	JU-19	JK-1	FD_INTENS_1	_HI[0]	_LO[0]	L28FDA
FL2H	15	JVA-2	-				
FL2L	16	JU-20	JK-2	FD_INTENS_2	_HI[1]	_LO[1]	L28FDB
FL3H	17	JVA-3	-				
FL3L	18	JU-21	JK-3	FD_INTENS_3	_HI[2]	_LO[2]	L28FDC
FL4H	19	JVA-4	-				
FL4L	20	JU-22	JK-4	FD_INTENS_4	_HI[3]	_LO[3]	L28FDD
FL5H	21	JVA-5	-				
FL5L	22	JU-23	JK-5	FD_INTENS_5	_HI[4]	_LO[4]	L28FDE
FL6H	23	JVA-6	-				
FL6L	24	JU-24	JK-6	FD_INTENS_6	_HI[5]	_LO[5]	L28FDF
FL7H	25	JVA-7	-				
FL7L	26	JU-25	JK-7	FD_INTENS_7	_HI[6]	_LO[6]	L28FDG
FL8H	27	JVA-8	-				
FL8L	28	JU-26	JK-8	FD_INTENS_8	_HI[7]	_LO[7]	L28FDH

Figure D-63. <P> Core - Flame Intensity Inputs on PTBA Terminal Board

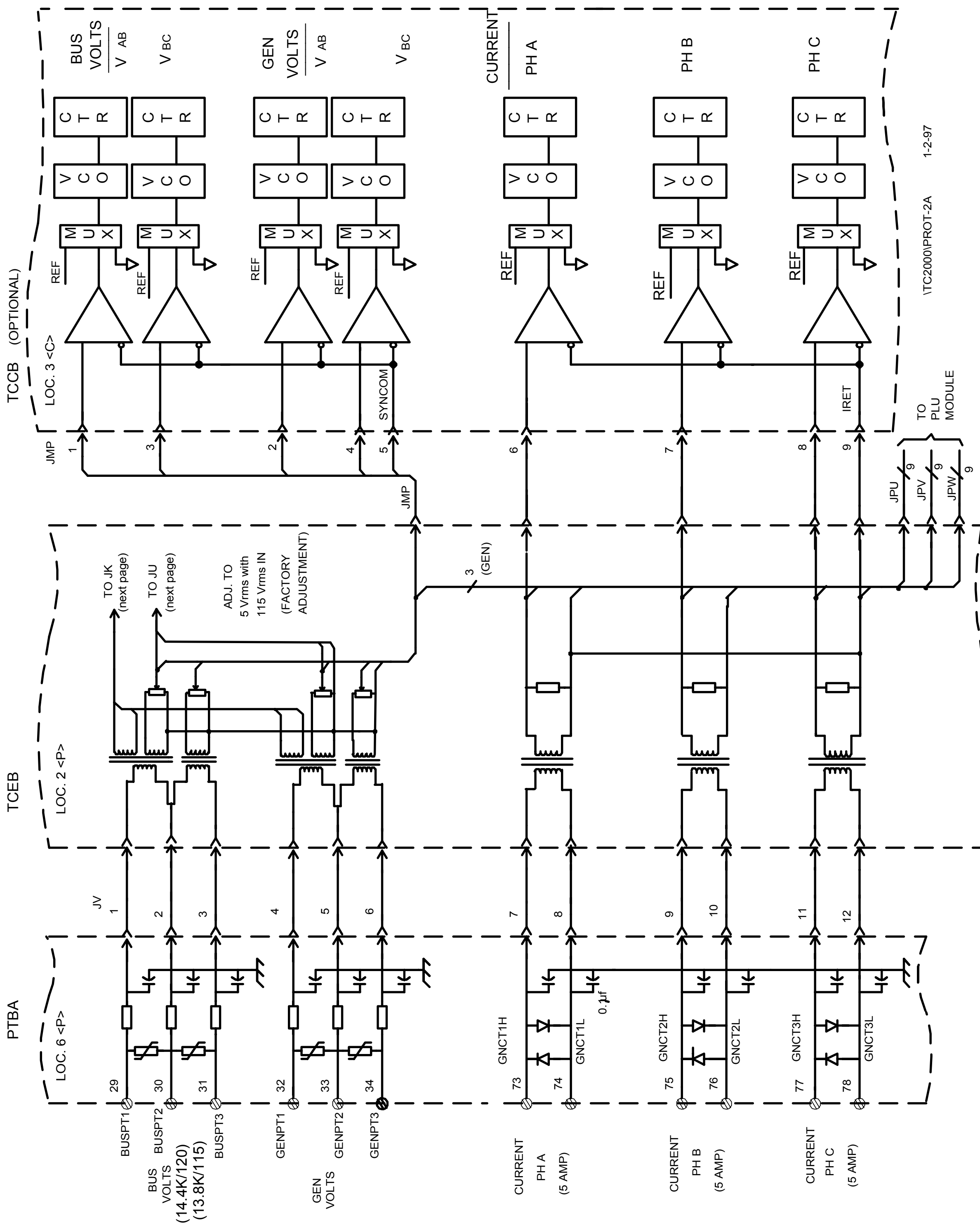


Figure D-64. <P> Core - Bus Generator PT Inputs on PTBA Terminal Board

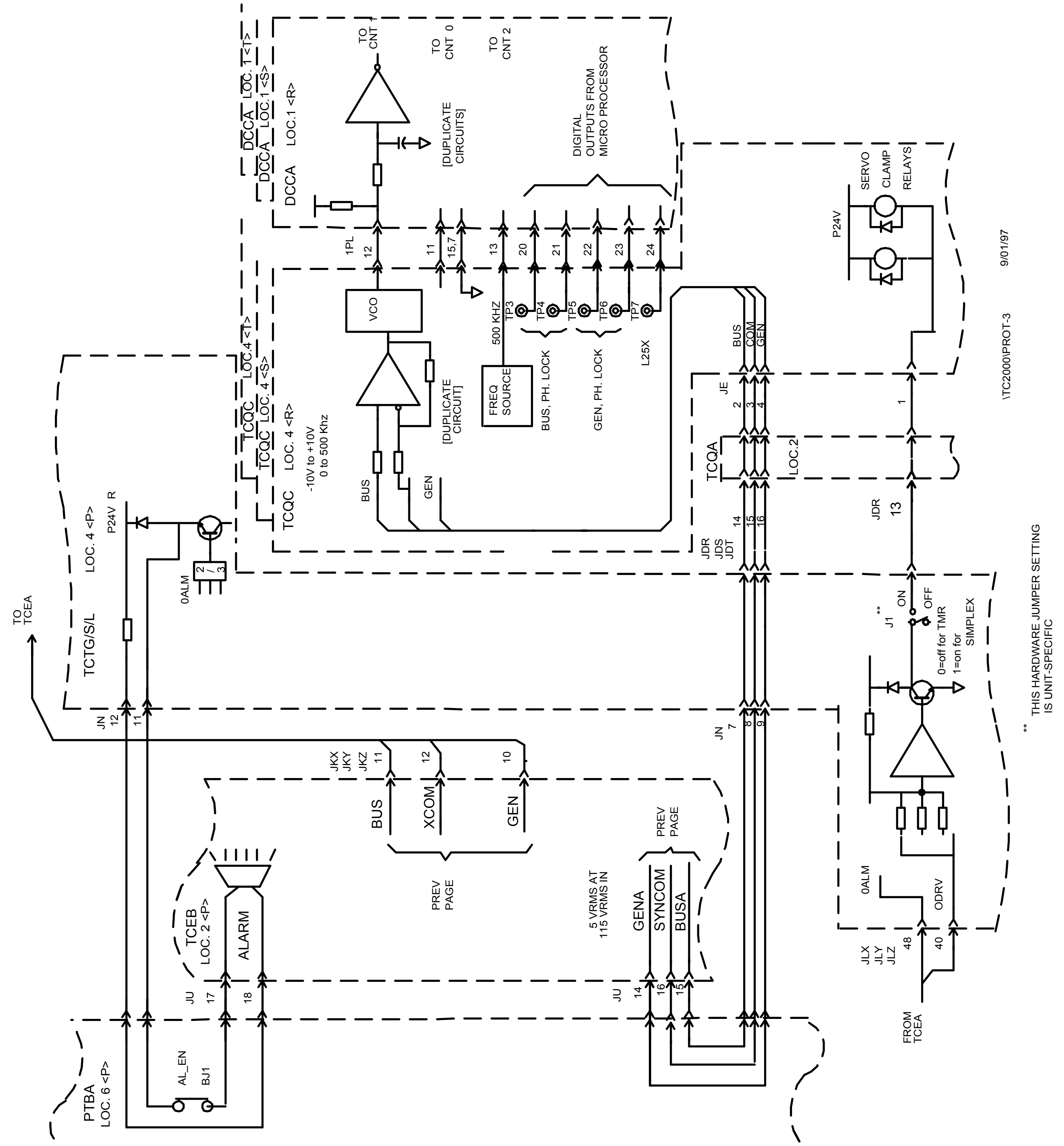


Figure D-65. <P> Core -Alarm Horn Circuit

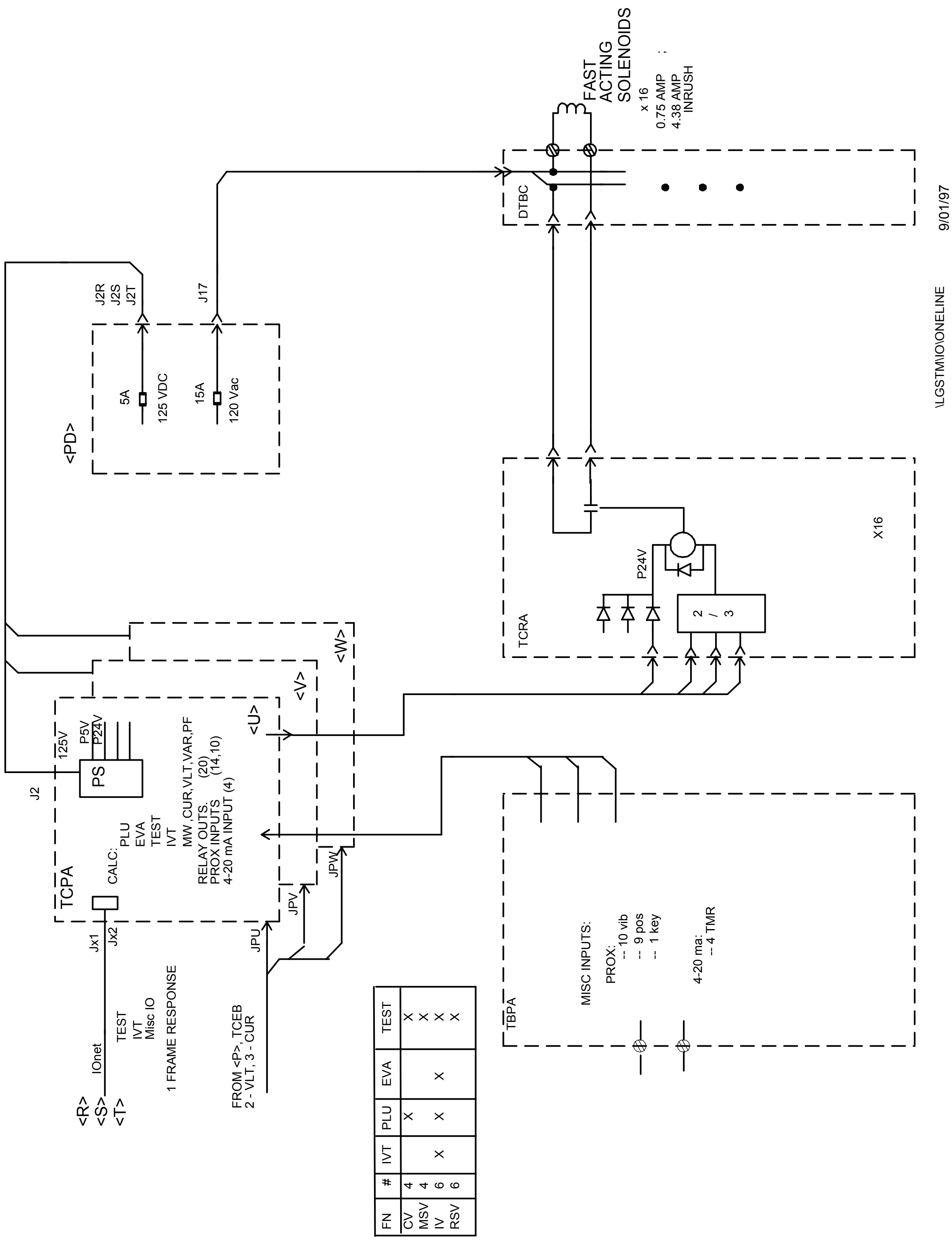
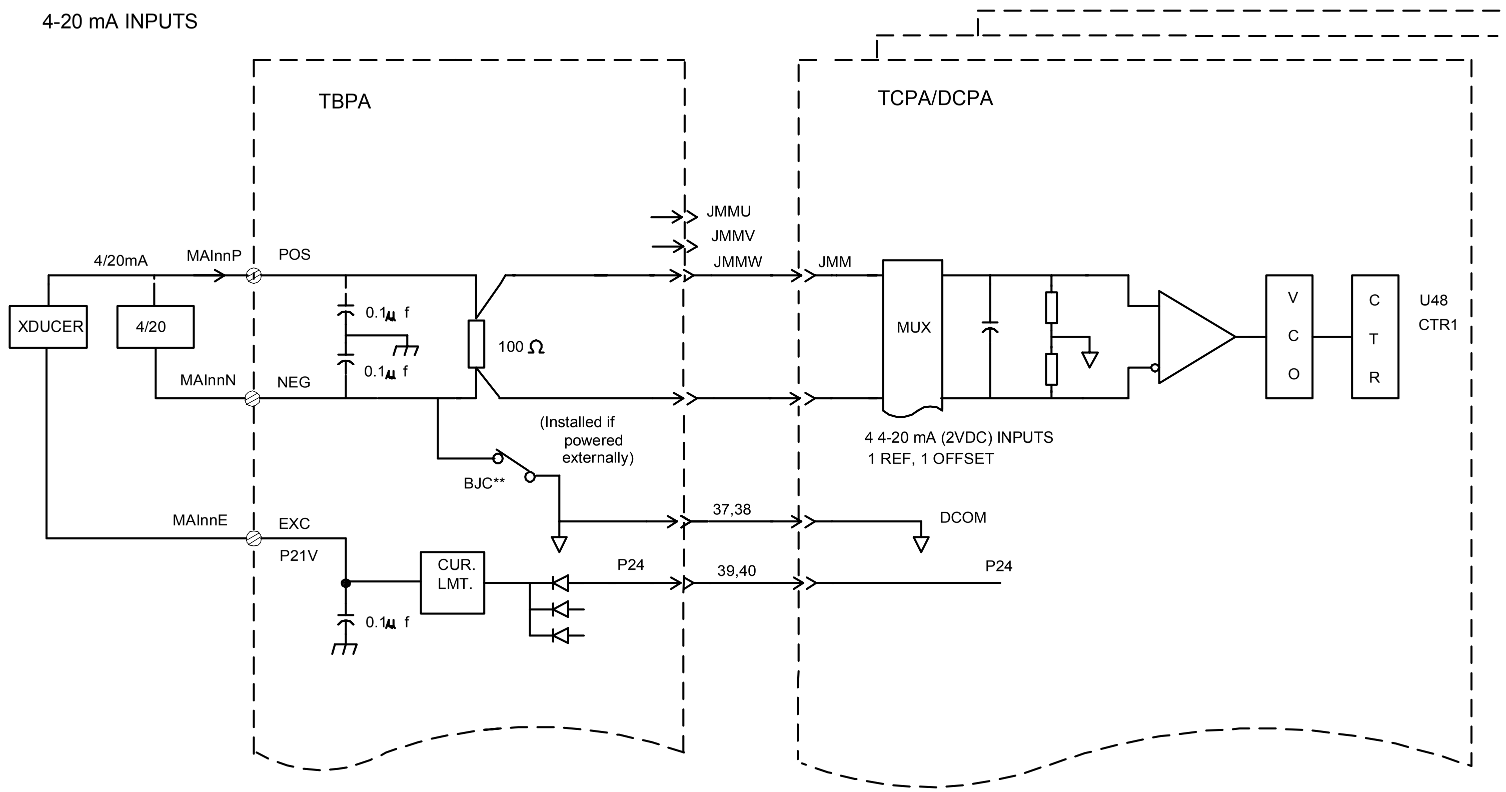


Figure D-66. PLU Core - Block Diagram

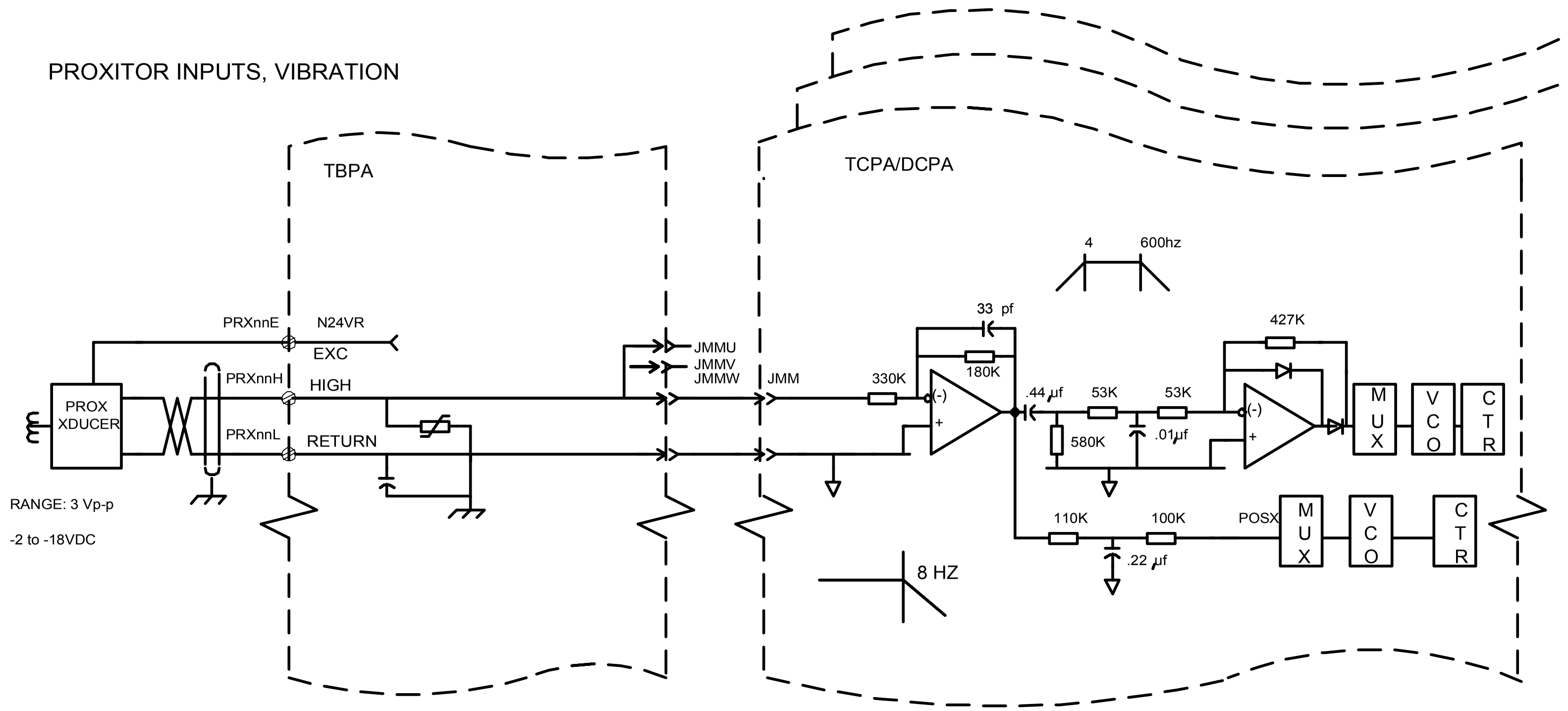


**THESE JUMPER SETTINGS
ARE UNIT-SPECIFIC

\LGSTM\IO\AIO-1 12/17/96

DEVICE TB DESIGNATION	SIGNAL TERMINATION POS	NEG	EXCITIATION TERMINATION (P21V)	JMM PIN		BJC JUMPER **
mA1-A	1	2	3	1,2	JMMU	BJ1
mA1-B	5	6	4	1,2	JMMV	BJ2
mA1-C	7	8	9	1,2	JMMW	BJ3
mA2-A	11	12	10	3,4	JMMU	BJ4
mA2-B	13	14	15	3,4	etc.	BJ5
mA2-C	17	18	16	3,4		BJ6
mA3-A	19	20	21	5,6		BJ7
mA3-B	23	24	22	5,6		BJ8
mA3-C	25	26	27	5,6		BJ9
mA4-A	29	30	28	7,8		BJ10
mA4-B	31	32	33	7,8		BJ11
mA4-C	35	36	34	7,8		BJ12

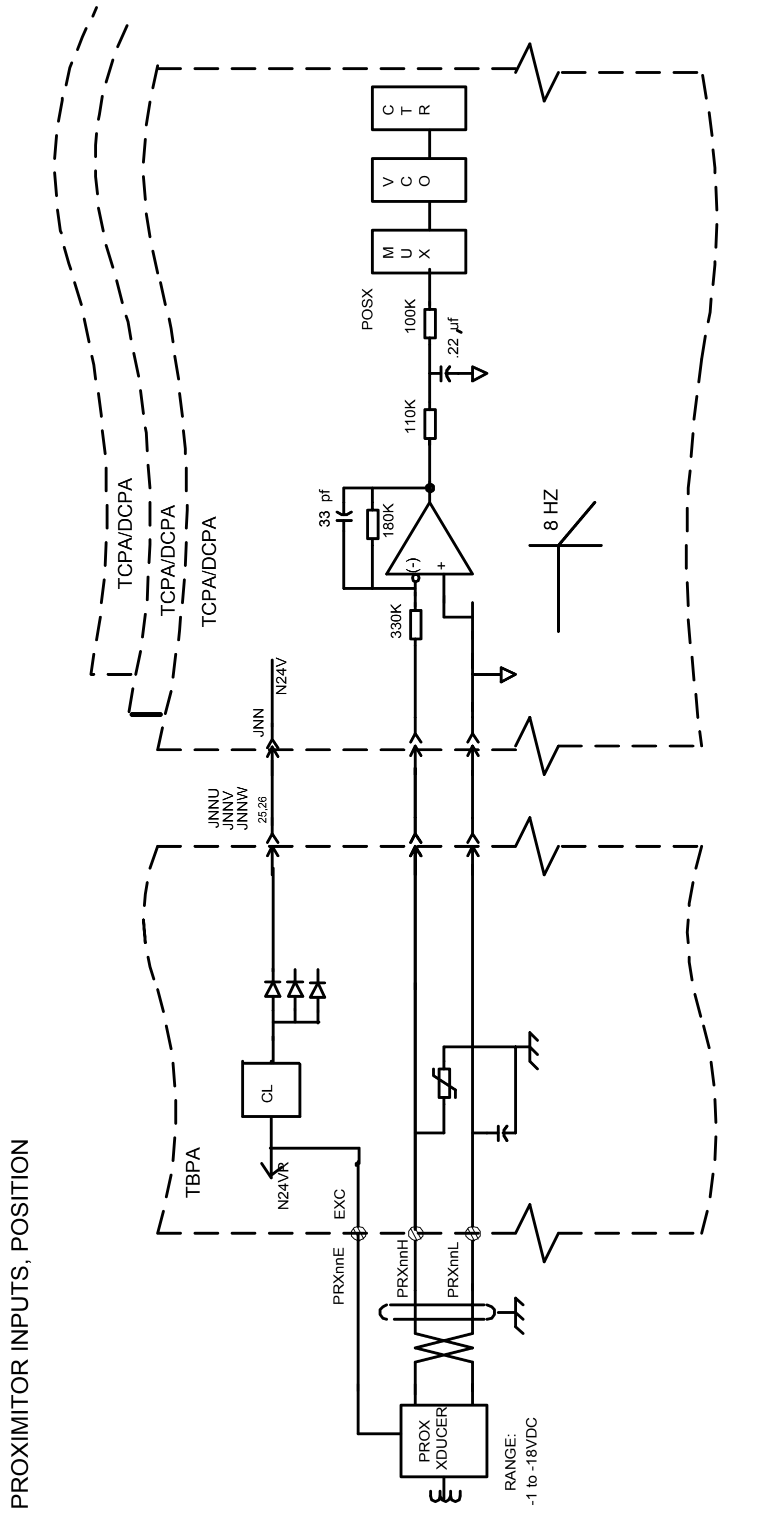
Figure D-67. PLU Core - Milliamp Inputs to TBPA



DEVICE TB DESIGNATION	SIGNAL TERMINATION HIGH	SIGNAL TERMINATION RETURN	EXCITATION TERMINATION	JMM
PROX01	37	38	39	9,10
PROX02	41	42	40	11,12
PROX03	43	44	45	13,14
PROX04	47	48	46	15,16
PROX05	49	50	51	17,18
PROX06	53	54	52	19,20
PROX07	55	56	57	21,22
PROX08	59	60	58	23,24
PROX09	61	62	63	25,26
PROX10	65	66	64	27,28

\LGSTM\IO\AIO-2 12/17/96

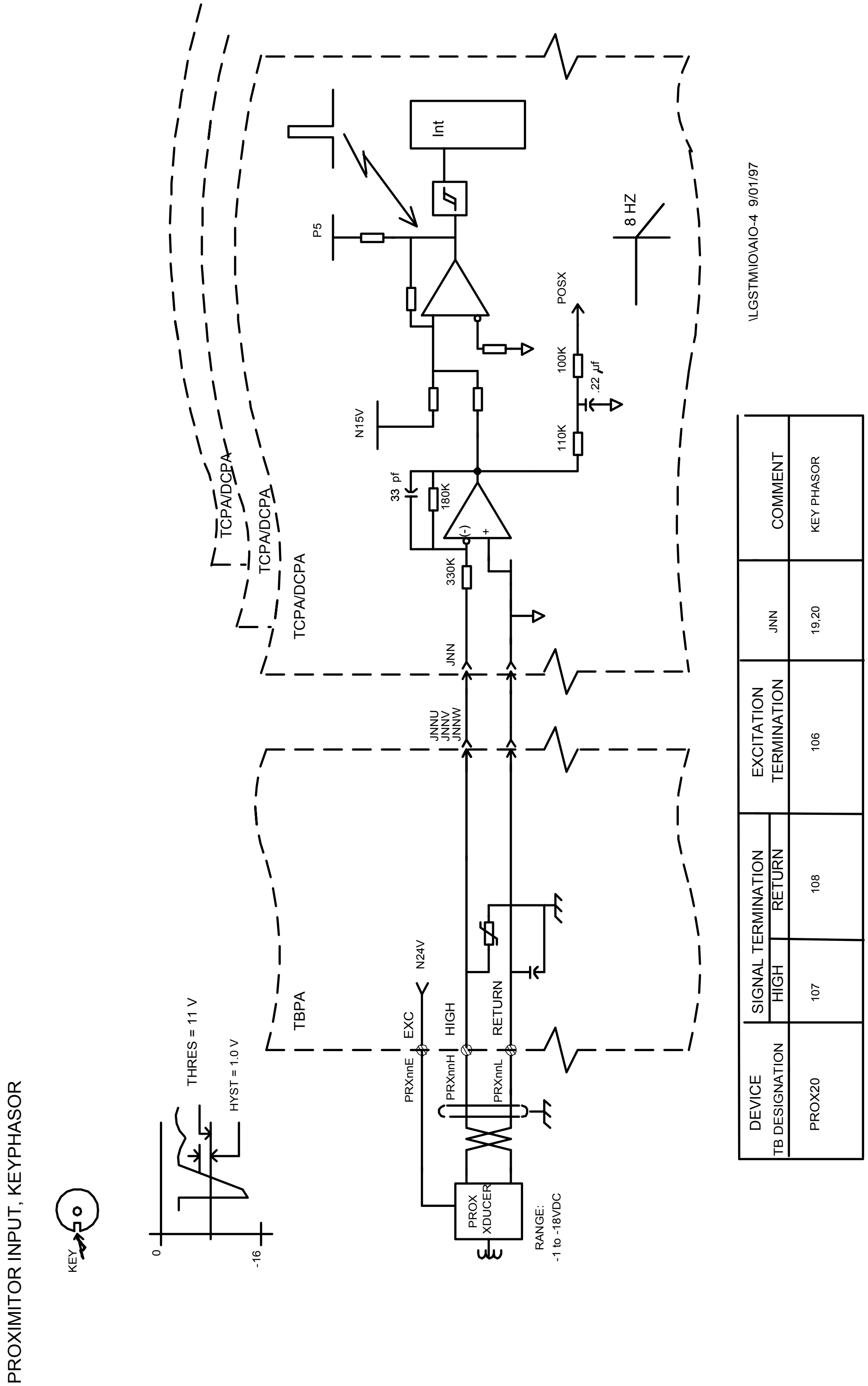
Figure D-68. PLU Core - Proximity Transducer Inputs, Vibration



DEVICE	SIGNAL TERMINATION		EXCITATION TERMINATION	JNN
TB DESIGNATION	HIGH	RETURN		
PROX11	79	80	81	1,2
PROX12	83	84	82	3,4
PROX13	85	86	87	5,6
PROX14	89	90	88	7,8
PROX15	91	92	93	9,10
PROX16	95	96	94	11,12
PROX17	97	98	99	13,14
PROX18	101	102	100	15,16
PROX19	103	104	105	17,18

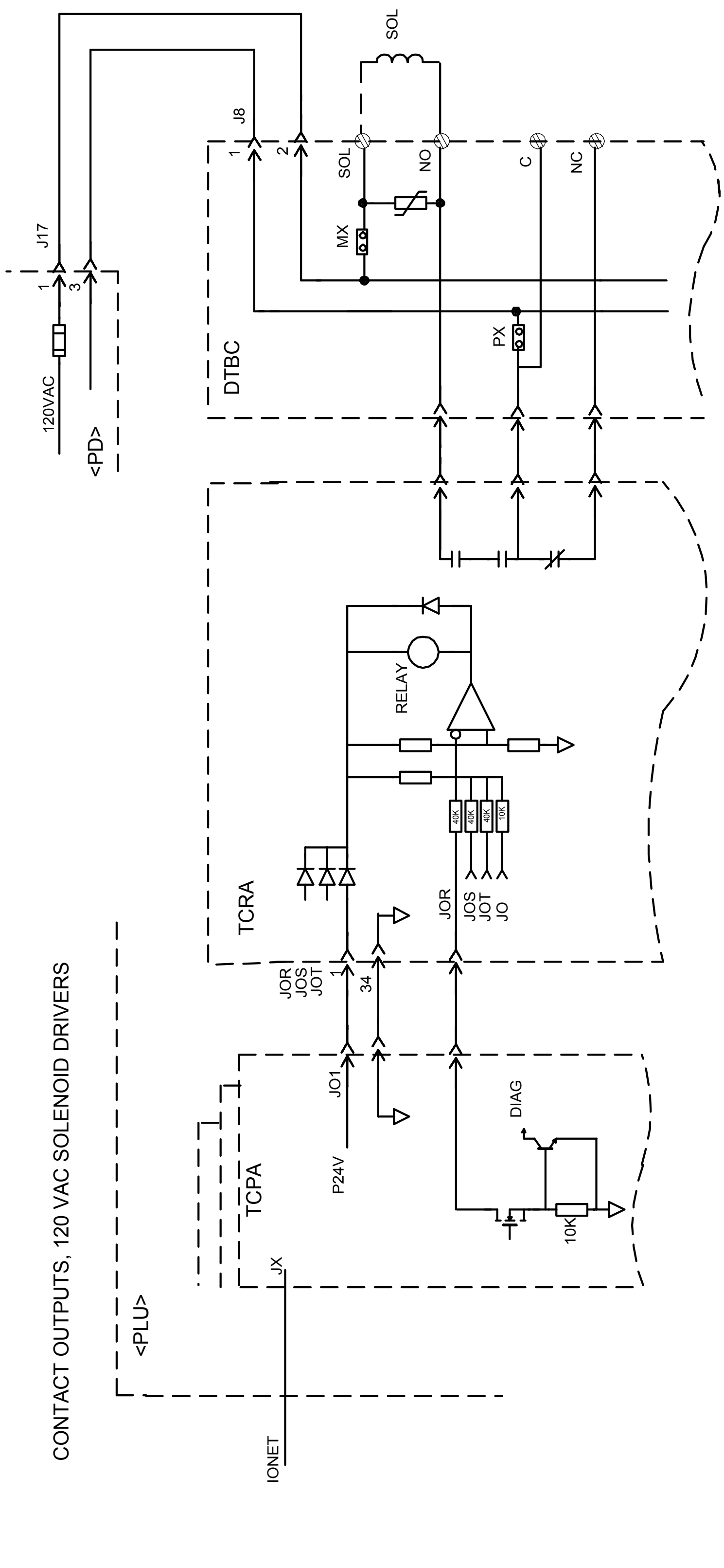
\LGSTMIO\AIO-3 9/01/97

Figure D-69. PLU Core - Proximity Transducer Inputs, Position



DEVICE TB DESIGNATION	SIGNAL TERMINATION HIGH	SIGNAL TERMINATION RETURN	EXCITATION TERMINATION	JNN	COMMENT
PROX20	107	108	106	19,20	KEY PHASOR

Figure D-70. PLU Core - Proximity Transducer Input, Keyphasor



OUTPUT #	TERM SCREW DTBC	ISOL JUMPERS	OUTPUT RELAYS TCRA	DEDICATED FA SOLENOIDS
SOL01	1,2	P1,M1	K1	MSV1
SOL02	5,6	P2,M2	K2	MSV2
SOL03	9,10	P3,M3	K3	CV1
SOL04	13,14	P4,M4	K4	CV2
SOL05	17,18	P5,M5	K5	CV3
SOL06	21,22	P6,M6	K6	CV4
SOL07	25,26	P7,M7	K7	IV1
SOL08	29,30	P8,M8	K8	IV2
SOL09	33,34	P9,M9	K9	IV3
SOL10	37,38	P10,M10	K10	IV4
SOL11	41,42	P11,M11	K11	IV5
SOL12	45,46	P12,M12	K12	IV6
SOL13	49,50	P13,M13	K13	RSV1
SOL14	53,54	P14,M14	K14	RSV2
SOL15	57,58	P15,M15	K15	SPARE
SOL16	61,62	P16,M16	K16	SPARE

Figure D-71. PLU Core - Contact Outputs

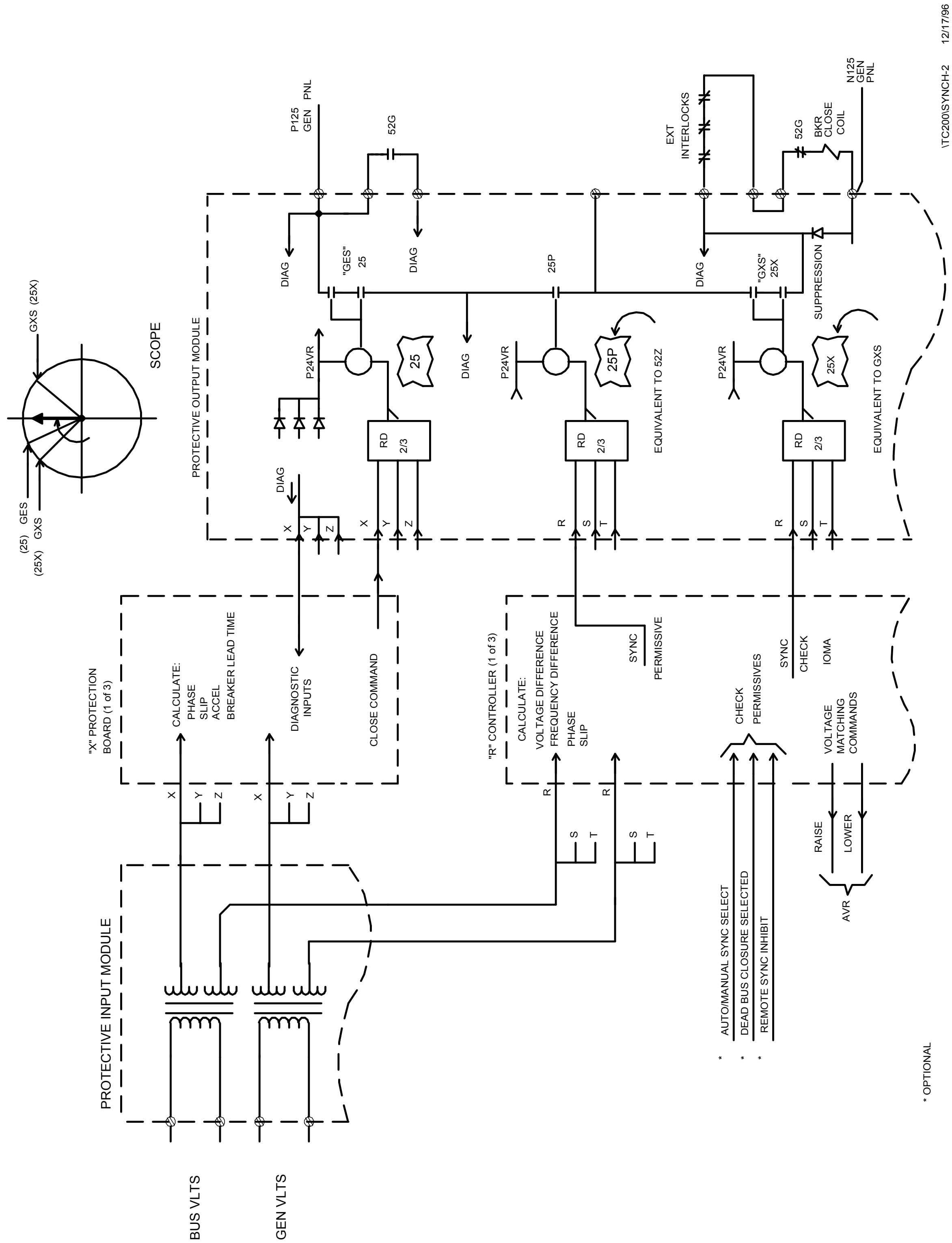


Figure D-72. Mark V Synchronize

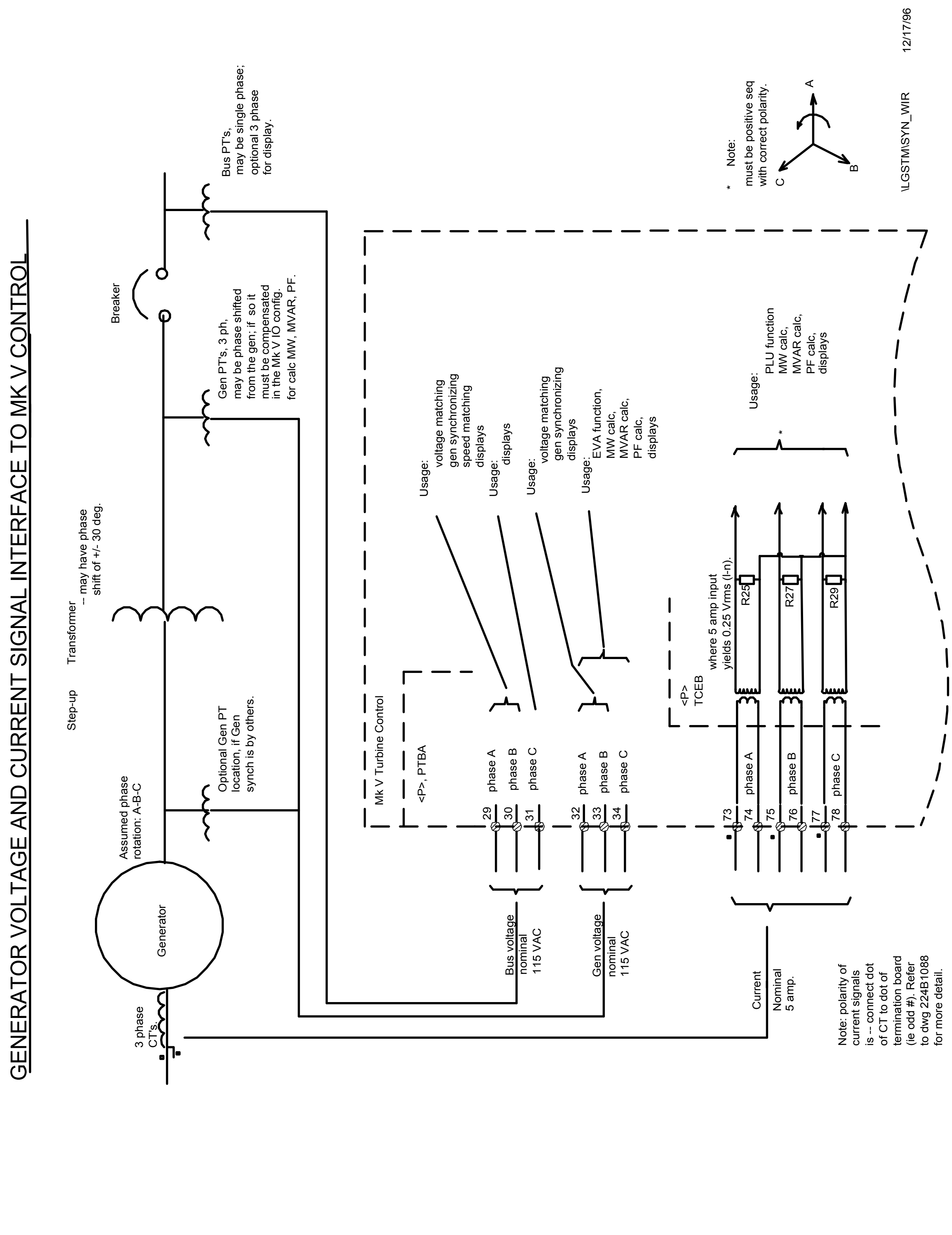
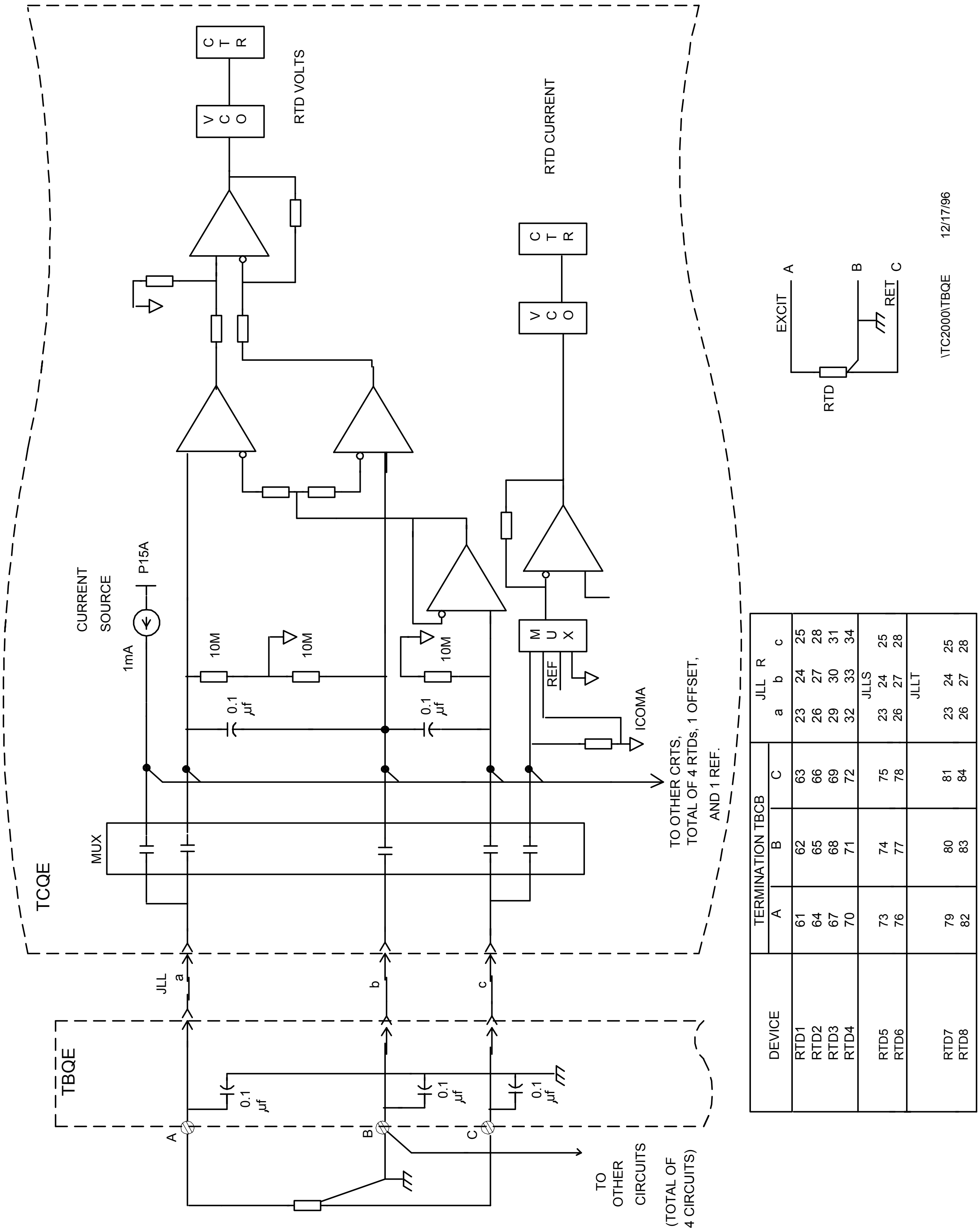


Figure D-73. <P> Core - Generator Voltage and Current Signal Interface



DEVICE	TERMINATION TBCB A	B	C	JLL R a	b	c
RTD1	61	62	63	23	24	25
RTD2	64	65	66	26	27	28
RTD3	67	68	69	29	30	31
RTD4	70	71	72	32	33	34
				JLLS		
RTD5	73	74	75	23	24	25
RTD6	76	77	78	26	27	28
				JLLT		
RTD7	79	80	81	23	24	25
RTD8	82	83	84	26	27	28

Figure D-74. RTD Inputs to <Q>

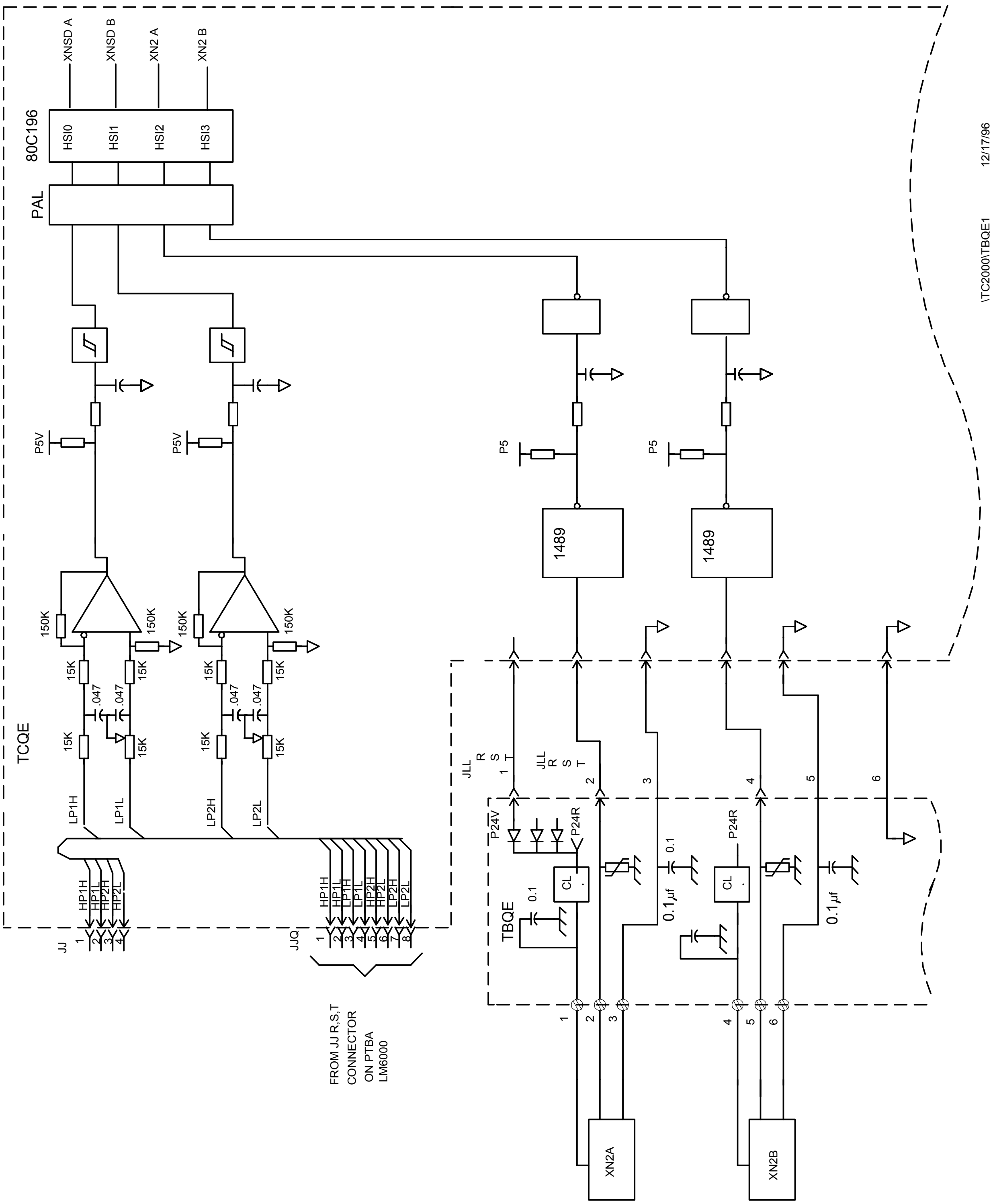
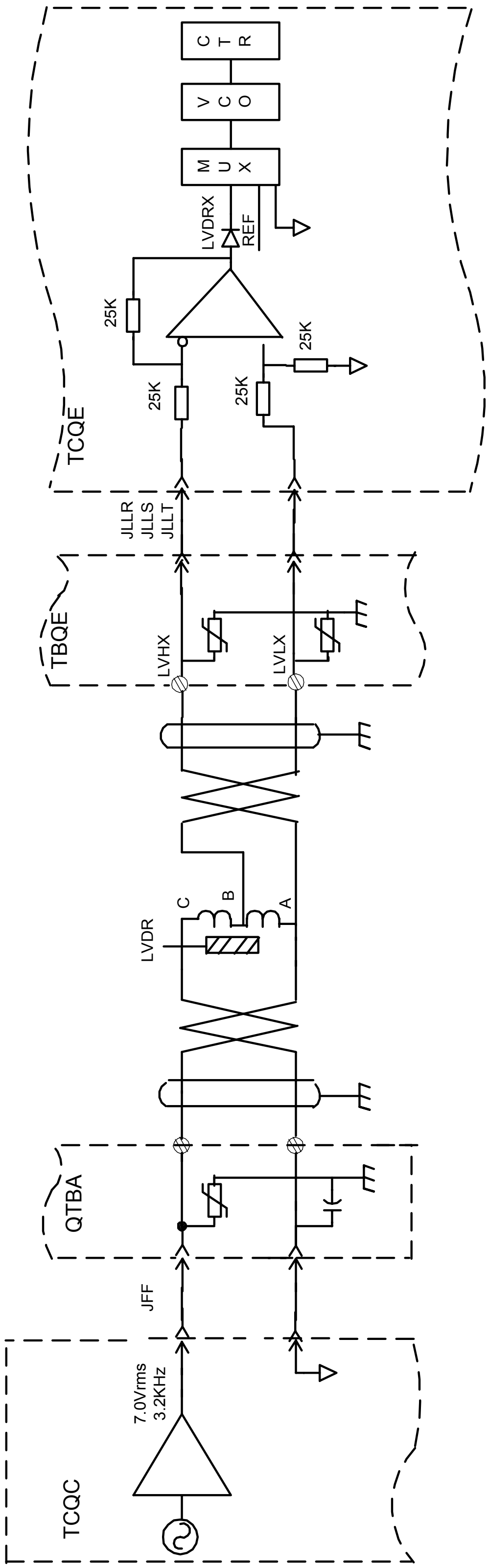


Figure D-75. <Q> Core - Pulse Rate Inputs on TBQE Terminal Board



DEVICE	INPUT (TBQE) TERMINATION	EXCITATION (QTBA) TERMINATION	JLL PIN #
LVDR17	7,8	<S> 13,14	7,8
LVDR18	9,10	<T> 13,14	9,10
LVDR19	11,12	<R> 15,16	11,12
LVDR20	13,14	<S> 15,16	13,14

\TC2000\TBQE2 12/17/96

Figure D-76. <Q> Core - LVDT/R Excitation and Feedback Signal on QTBA/TBQE

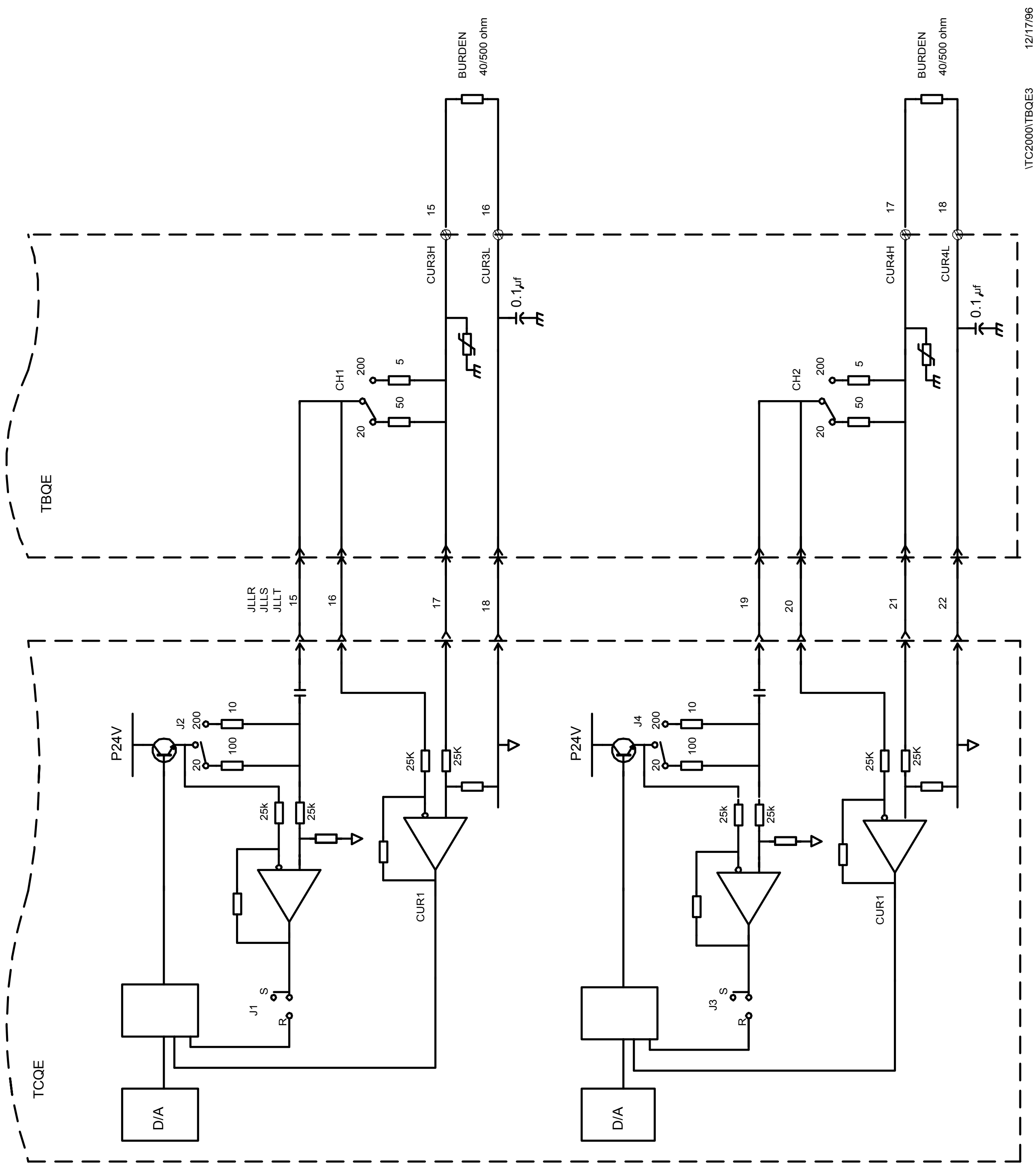


Figure D-77. <Q> Core - 4-20/200 mA Outputs on TBQE Terminal Board

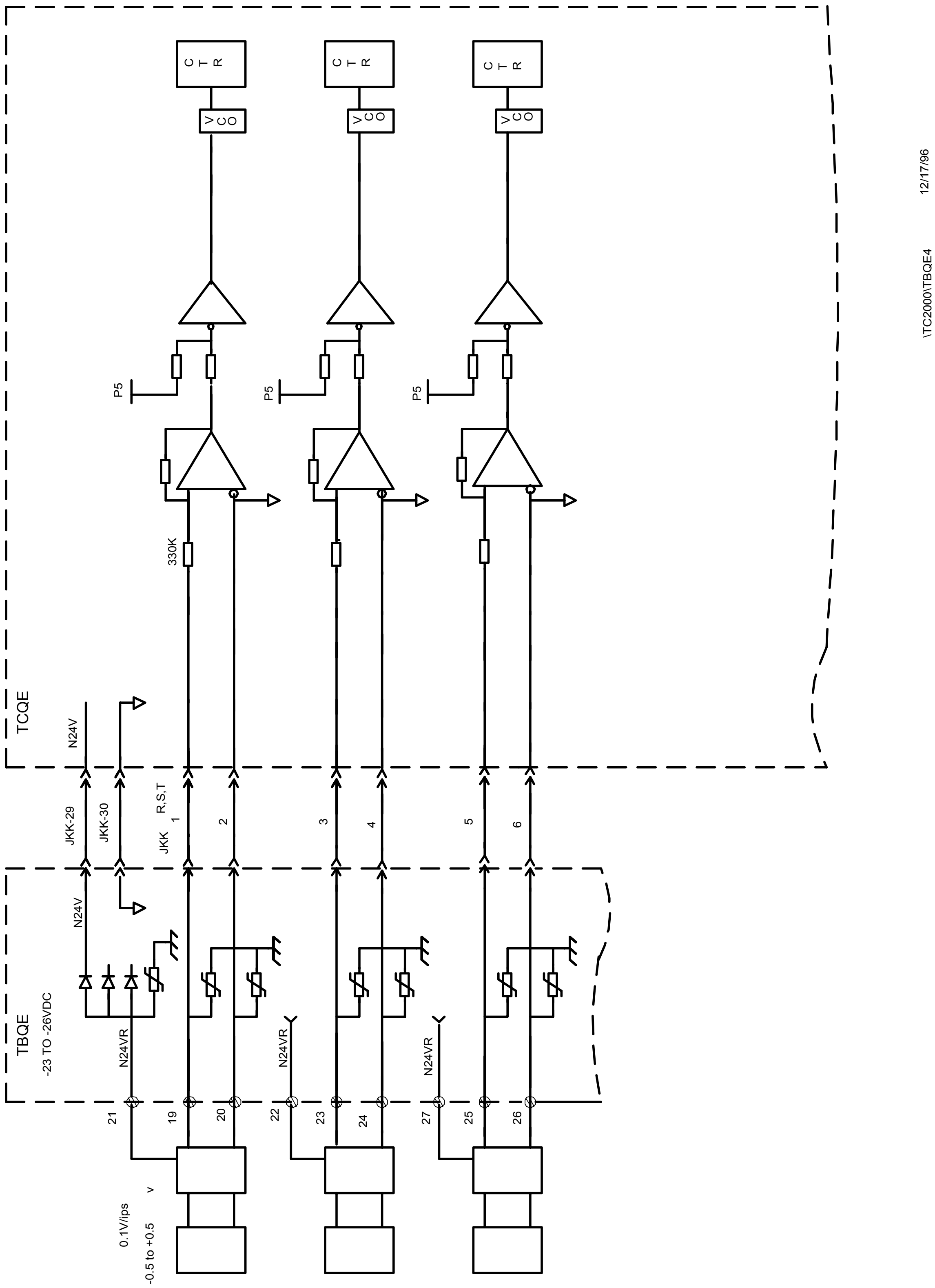
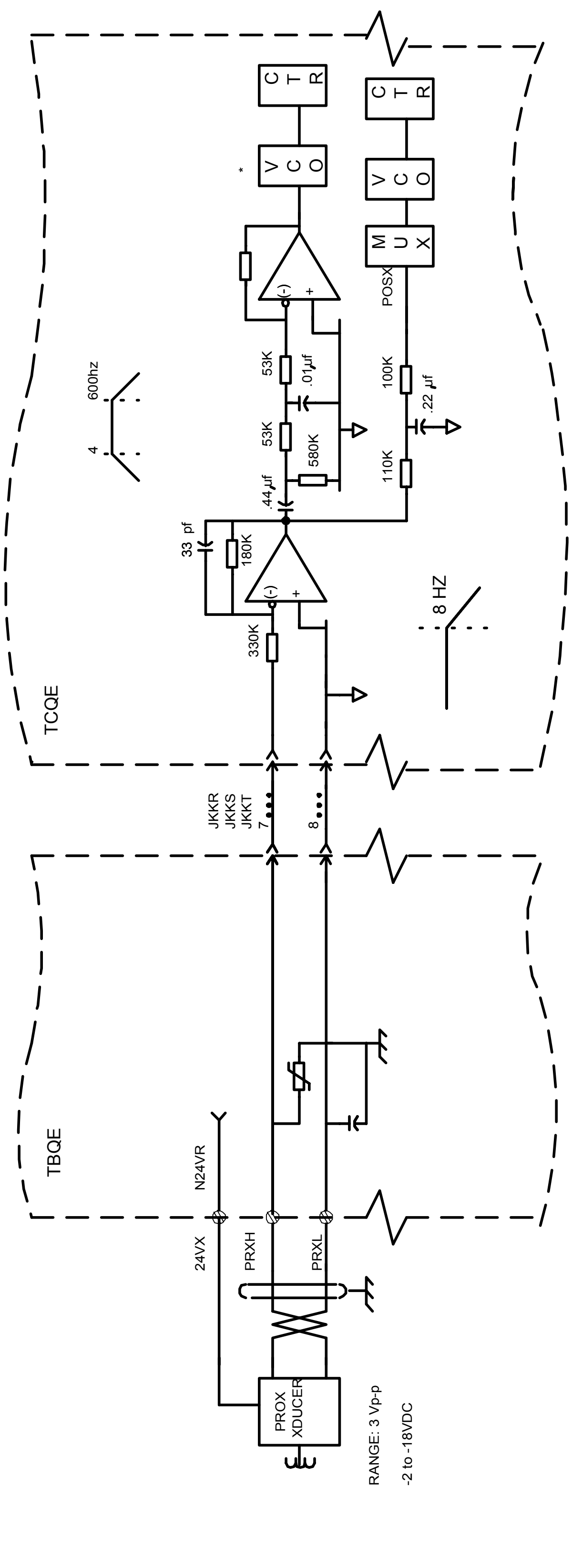
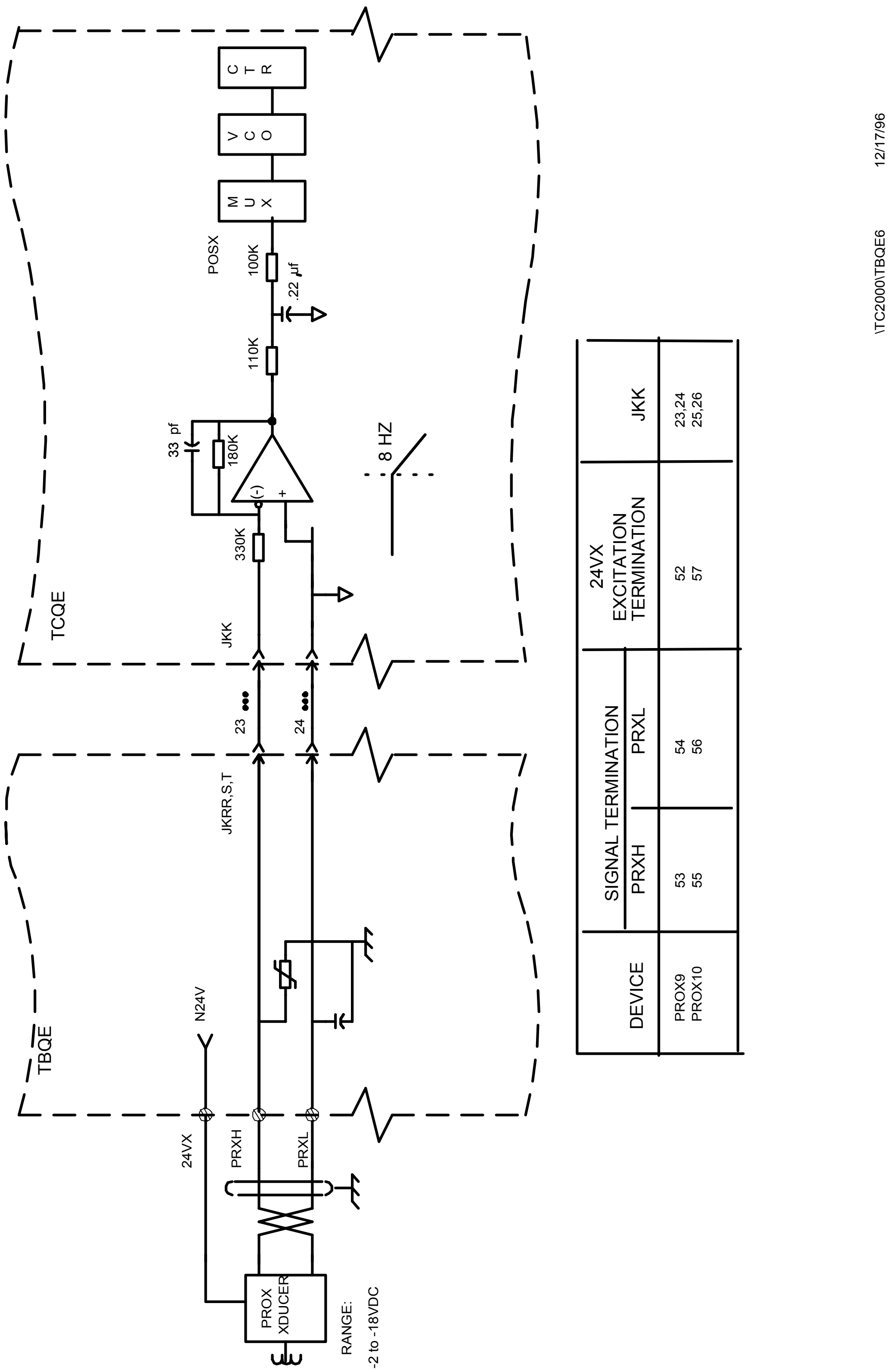


Figure D-78. <Q> Core - Vibration Inputs on TBQE Terminal Board



DEVICE	SIGNAL TERMINATION PRXH	SIGNAL TERMINATION PRXL	24XV EXCITATION TERMINATION	JKK
PROX1	29	30	28	7,8
PROX2	31	32	33	9,10
PROX3	35	36	34	11,12
PROX4	37	38	39	13,14
PROX5	41	42	40	15,16
PROX6	43	44	45	17,18
PROX7	47	48	46	19,20
PROX8	49	50	51	21,22

Figure D-79. <Q> Core - Proximity Transducer Inputs on TBQE Terminal Board



DEVICE	SIGNAL TERMINATION PRXH	SIGNAL TERMINATION PRXL	24VX EXCITATION TERMINATION	JKK
PROX9	53	54	52	23,24
PROX10	55	56	57	25,26

Figure D-80. <Q> Core - Proximity Transducer Inputs on TBQE Terminal Board

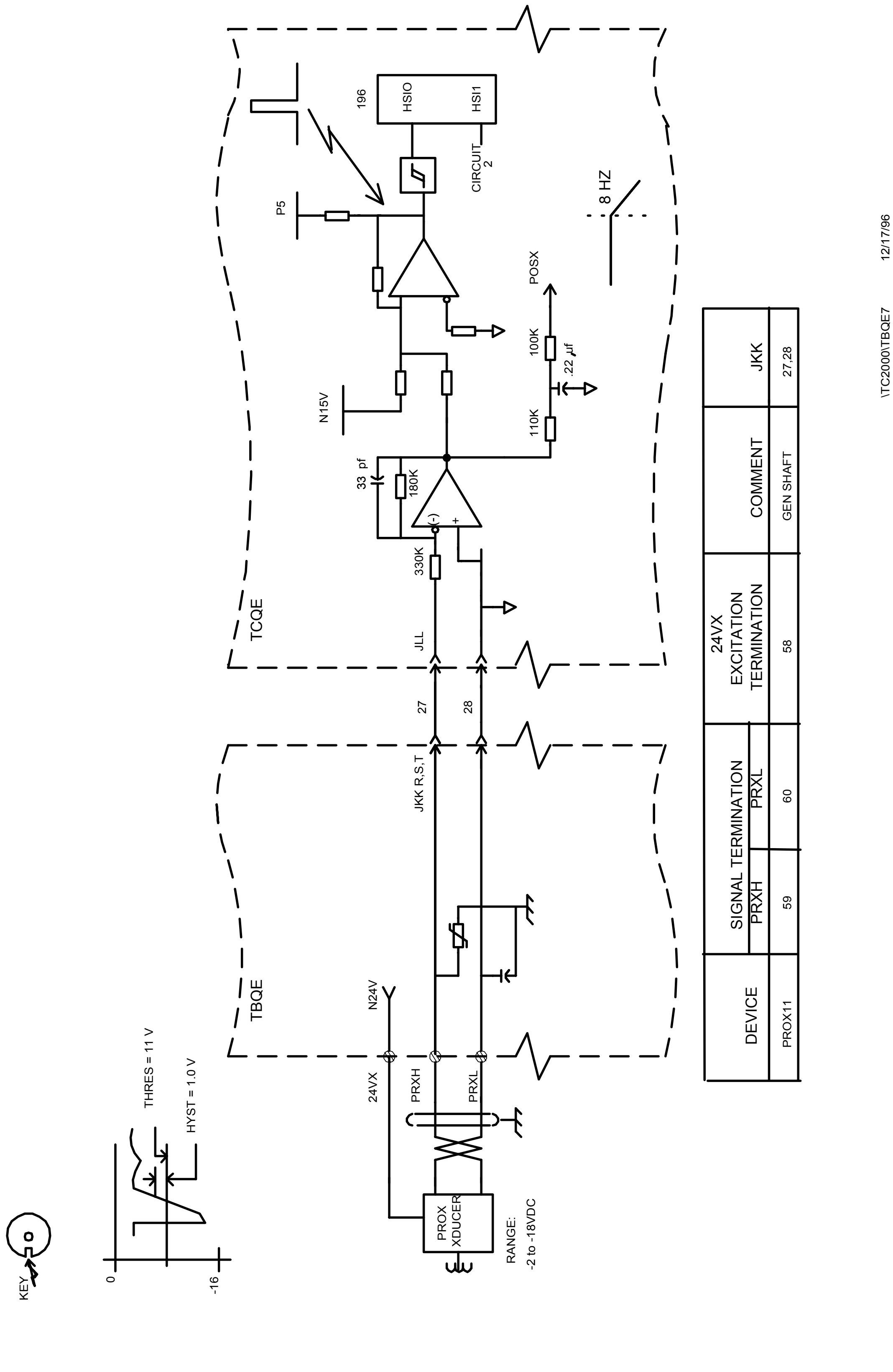


Figure D-81. <Q> Core - Proximity Transducer Inputs on TBQE Terminal Board

Notes:

APPENDIX E

POWER DISTRIBUTION CORE DIAGRAMS

E-1. INTRODUCTION AND APPLICATION NOTES

The following pages are diagrams of the Power Distribution Core <PD>. For simplification, not every component is shown.

Since the AC supply voltage can be either 120 or 240 volts, the transformer and rectifier assembly must change.

The JZ4 and JZ5 plugs are the auxiliary AC source connection point for applications that require an isolated input for an uninterruptable power source connection(s) and/or switched back up power source. The 'HOT' side can be isolated with external circuitry connected to these plugs, but the neutral cannot.

CAUTION

Care must be used when connecting auxiliary AC sources into the JZ4 and JZ5 plugs. The neutral of the auxiliary AC input (JZ4 and JZ5 plugs) is connected to the neutral of the critical AC input (TB1, points 7, 8, 11, and 12).

The 27 boxes (for ANSI standard designation number) are undervoltage devices that will send a logic signal to the <C> core if the input voltage, either AC or DC, is no longer present. These signals can be incorporated into sequencing to be used as alarms. The AC undervoltage boxes incorporate a LED to give a visual indication of fuse status; in the DC circuits, the LED is still provided, but in a seperate circuit.

Example of the undervoltage detectors as shown on the figures of this appendix.

```
     │
     │
     │
┌─────────────┐
│   27        │
└─────────────┘
     │    │
     │    └──DIAGNOSTIC
     │
```

If the green LED is off or the red LED is on, the fuse is open.

Switches are provided on newer revisions of the <PD> core. These should be used when it is necessary to remove power from only one core or card at a time.

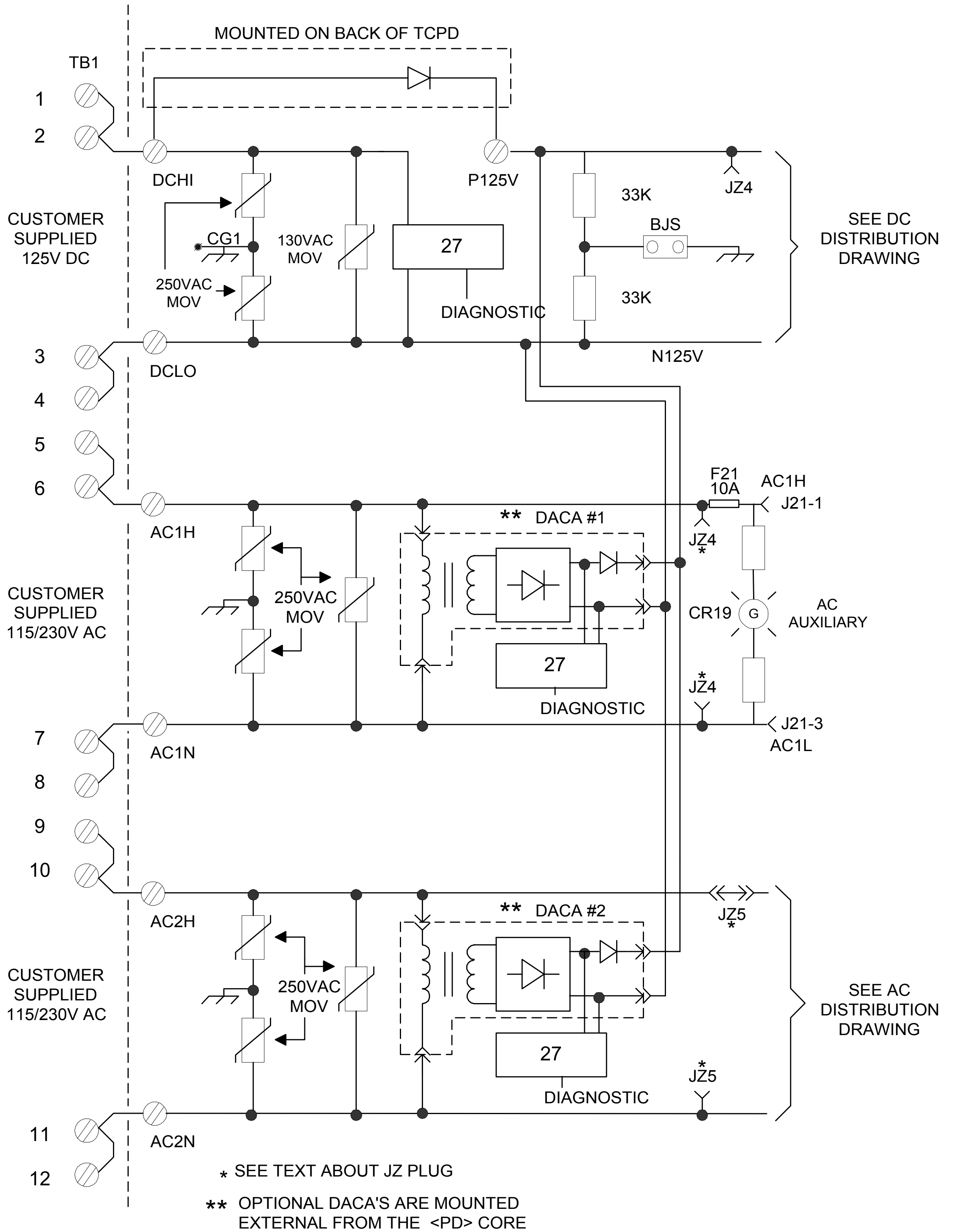
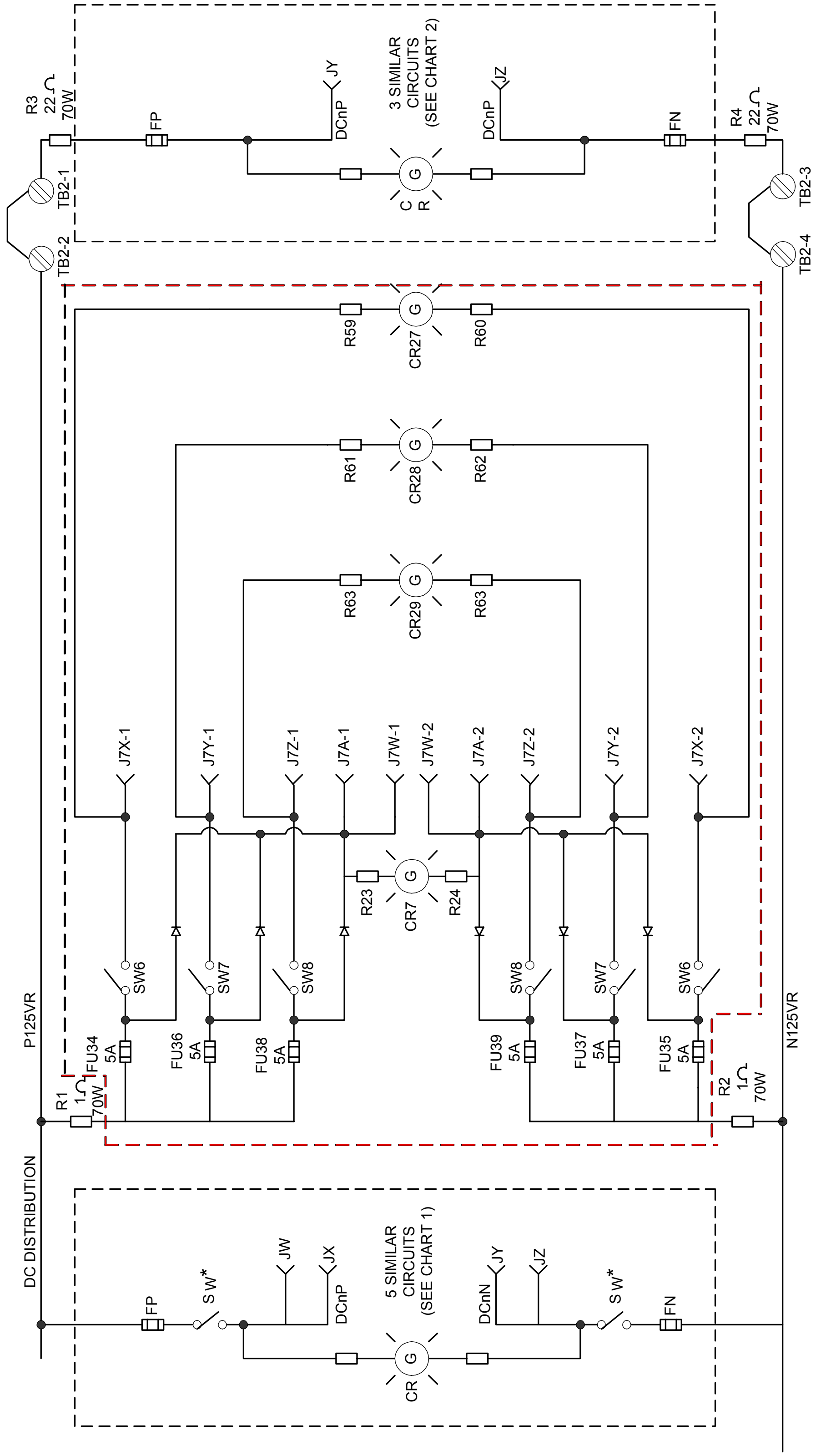


Figure E-1. <PD> Core Incoming AC & DC Circuits



* In Chart 1, Circuits 6-9 are similar to circuits 1-5 except there are no switches

Figure E-2. <PD> Core DC Power Distribution

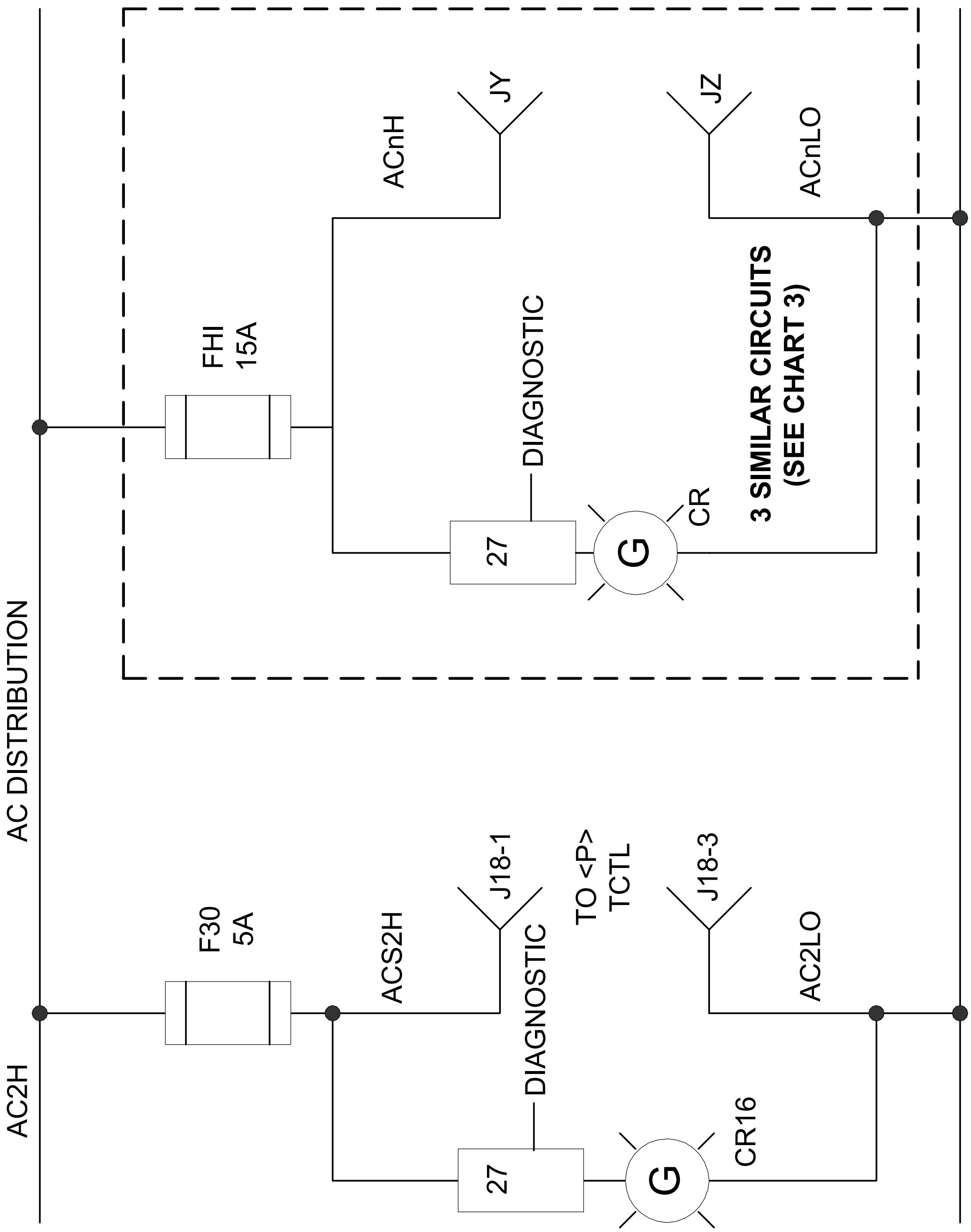


Figure E-3. <PD> Core AC Power Distribution

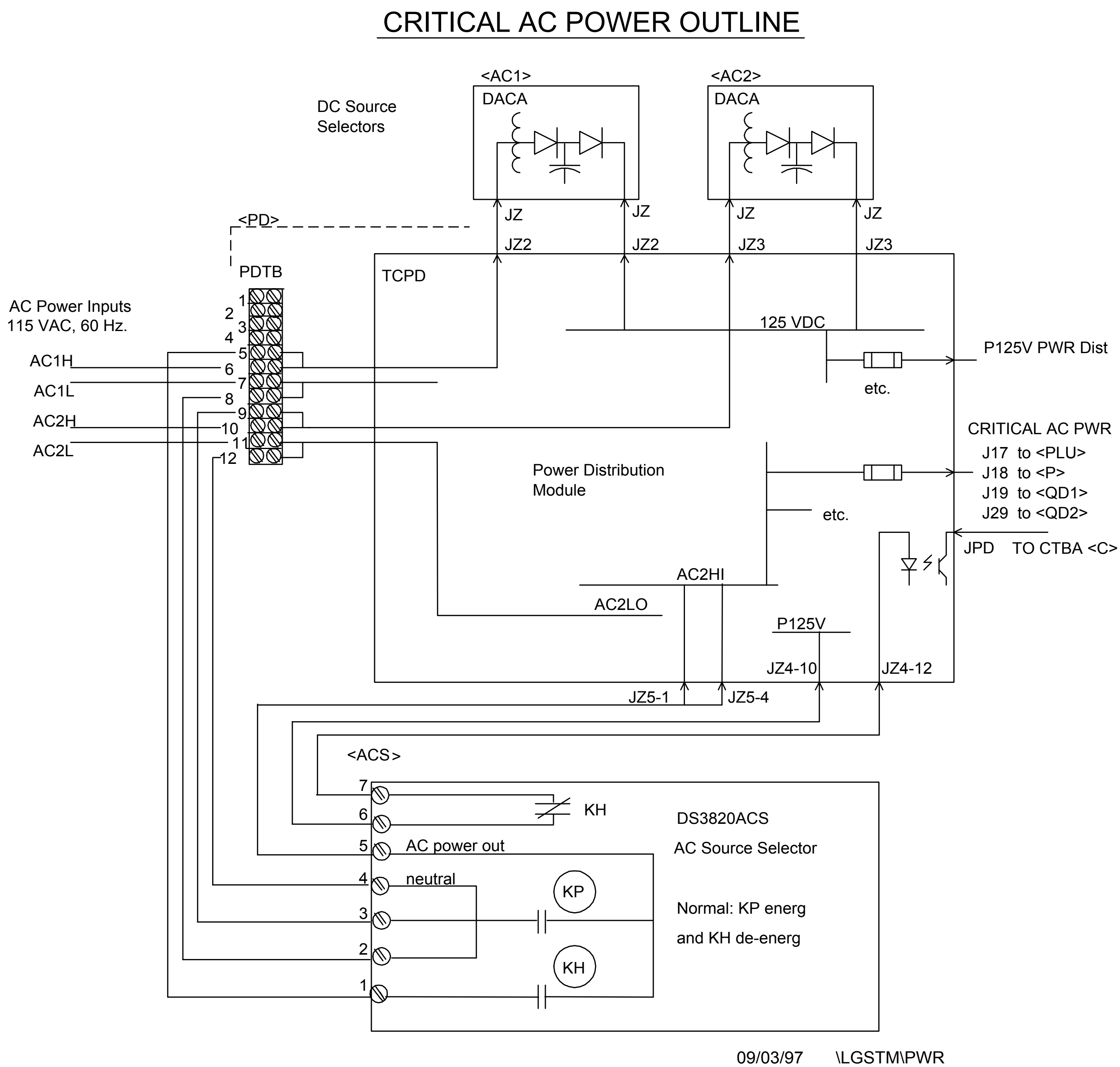


Figure E-3A. <PD> Core Critical AC Power Distribution (Large Steam)

DC DISTRIBUTION, CHART 1

Circuit	CRn	FP	FN	Amps	SW	CR	Positive		Negative		CORE
							JW	JX	JY	JZ	
1	1	FU1	FU2	5	1	1	J1R-1	J2R-1	J1R-2	J2R-2	R
2	2	FU3	FU4	5	2	2	J1S-1	J2S-1	J1S-2	J2S-2	S
3	3	FU5	FU6	5	3	3	J1T-1	J2T-1	J1T-2	J2T-2	T
4	4	FU7	FU8	5	4	4	J1C-1	N/A	J1C-2	N/A	C
5	5	FU9	FU10	5	5	5	J1D-1	N/A	J1D-2	N/A	D
6	8	FU13	FU14	15	N/A	8	J8A-1	N/A	J8A-2	N/A	QD1
7	9	FU15	FU16	15	N/A	9	J8B-1	N/A	J8B-2	N/A	CD
8	10	FU17	FU18	15	N/A	10	J8C-1	N/A	J8C-2	N/A	*
9	11	FU19	FU20	15	N/A	11	J8D-1	N/A	J8D-2	N/A	*

*These connections can go to either the <CD> or <QD2> core depending on the Mark V application. See the Mark V case wiring diagram for actual connections for the specific job.

DC DISTRIBUTION, CHART 2

Circuit	DCnP DCnNn	FP	FN	Amps	CR	Positive	Negative	CORE
						JY	JZ	
1	12	FU21	FU22	1.5	12	J12A-1	J12A-2	QD1
2	13	FU23	FU24	1.5	13	J12B-1	J12B-2	CD1
3	14	FU25	FU26	1.5	14	J12C-1	J12C-2	QD2

AC DISTRIBUTION, CHART 3

Circuit	ACnH ACnLO n	FHI	Amps	CR	"HOT"	Neutral	CORE
					JY	JZ	
1	1	FU29	15	15	J17-1	J17-3	AUX
2	2	FU30	5	16	J18-1	J18-3	*TCTL
3	3	FU31	15	17	J19-1	J19-3	QD1
4	4	FU32	15	18	J20-1	J20-3	QD1

*Second circuit used only on large steam applications

In gas turbine applications, AC3H and AC4H are used for the ignition transformers. In some steam turbine applications, these are often used for the test solenoids, reference Fig. 3A. The AC distribution voltage depends on the application.

E-2. Special Use Contact Outputs and Related Components

Two contact output circuits contain unique components for special applications. Contact output 16 and 17 are designed for gas turbine solenoid circuits; contact output 16 for compressor bleed valve solenoid, and contact output 17 for liquid fuel forwarding stop valve solenoid. Figure E-4 shows contact output 16 (`Q_QD1_CO16`) and Figure E-5 shows contact output 17 (`Q_QD1_CO17`).

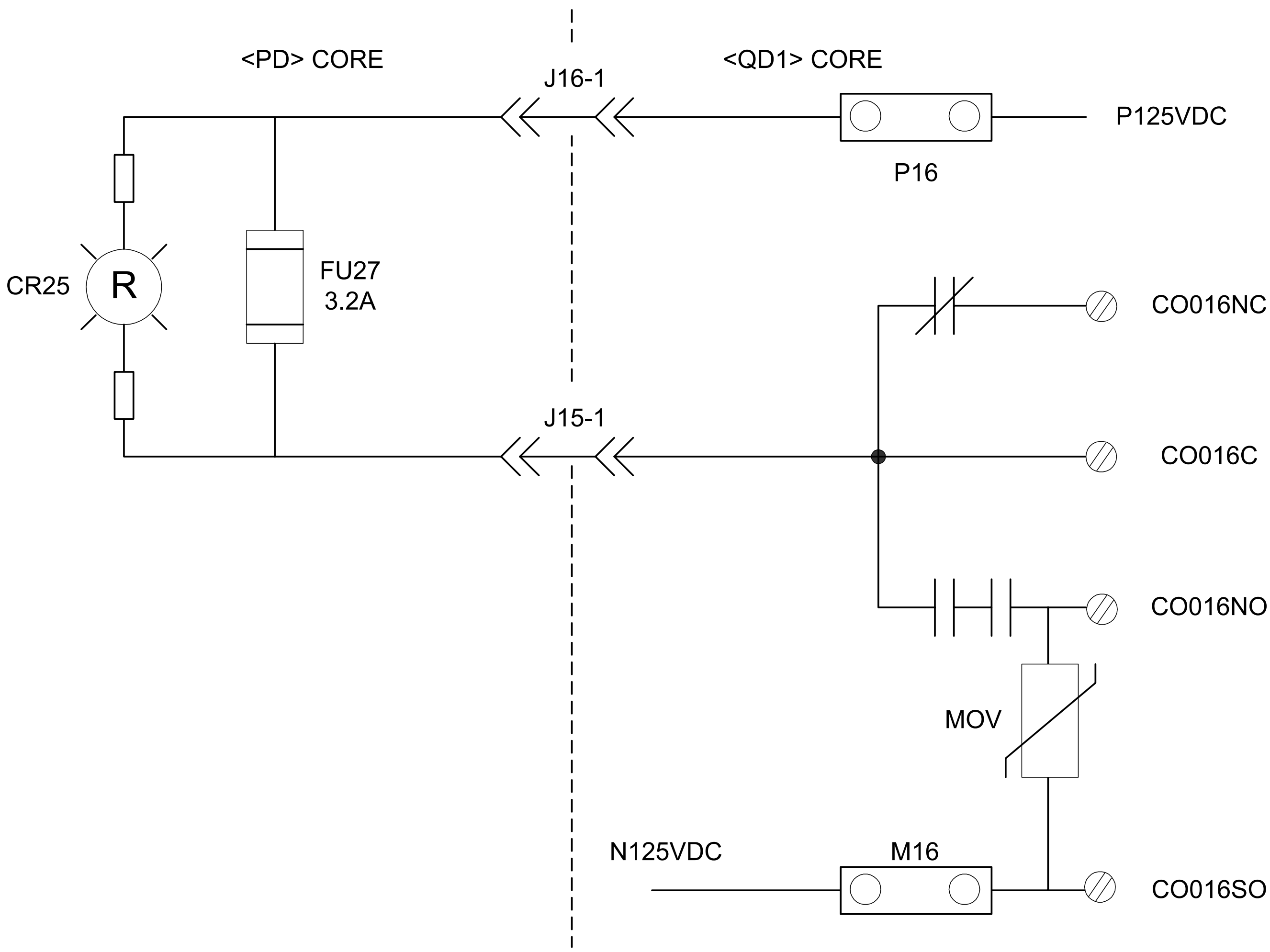


Figure E-4. Contact Output 16 Circuit

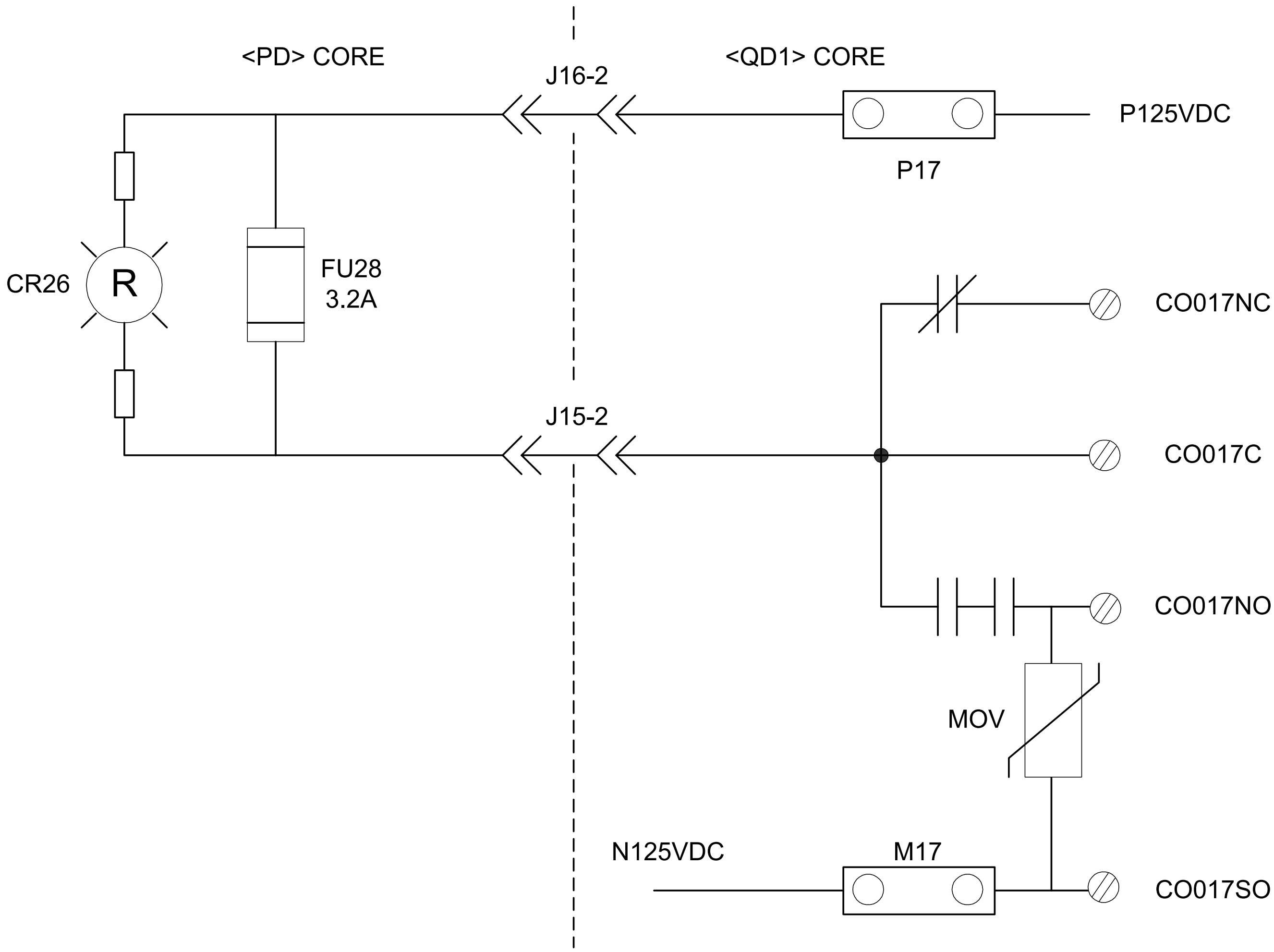


Figure E-5. Contact Output 17 Circuit

Contact output 18 (`Q_QD1_CO18`) is designed for the high pressure liquid fuel pump clutch soleniod in gas turbine applications. If this solenoid requires reduced voltage, remove the jumper from JZ2 to JZ3. Figure E-5 shows the components of this circuit.

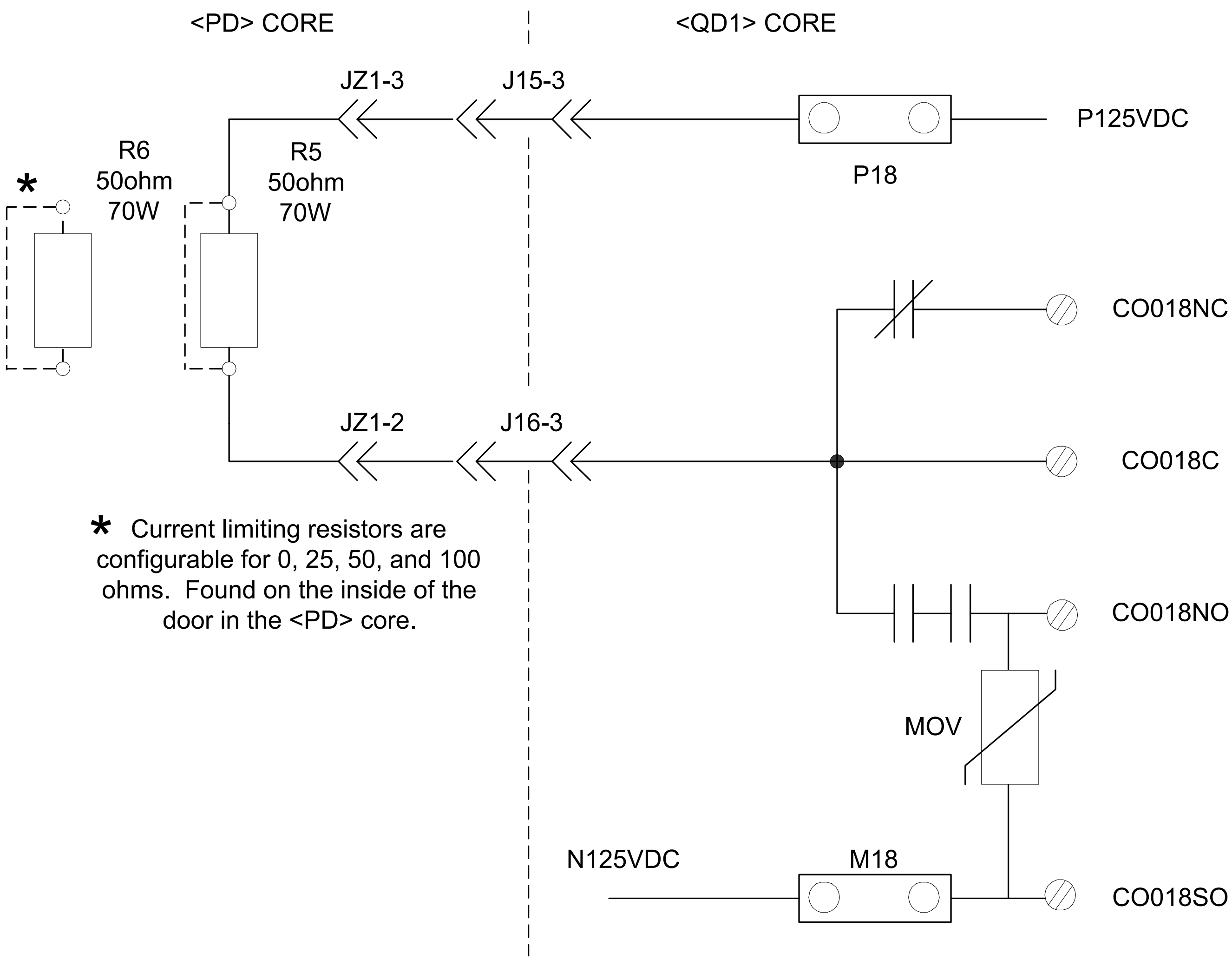


Figure E-6. Contact Output 18 Circuit

Notes:

APPENDIX F

CONTROL CONSTANT AND I/O SUMMARY

F-1. CONTROL CONSTANT FORMAT AND INDIVIDUAL POINT

Some constants are referred to as initialization control constants. These constants are only read at initialization time. This means that the constant can be changed, but the new value will not be used until ATS is reinitialized.
Entries for individual constants and points are in the following format:

NAME Description.
Scaling and Where used (functions).
Value rational, as necessary.

F-2. LIST OF CONTROL CONSTANTS FOR AUTOMATIC TURBINE STARTUP

KACC1_1 Temperature to stress conversion constant for the first area to evaluate.
10**3 Used in Acceleration function.

KACC1_2 Temperature to stress conversion constant for the second area to evaluate.
10**3 Used in Acceleration function.

KACC1_3 Temperature to stress conversion constant for the third area to evaluate.
10**3 Used in Acceleration function.

KACC2_1 Temperature intersect at zero stress for area one.
DEG F Used in Acceleration function.

KACC2_2 Temperature intersect at zero stress for area two.
DEG F Used in Acceleration function.

KACC2_3 Temperature intersect at zero stress for area three.
DEG F Used in Acceleration function.

KAD Multiplier for 3 or 4 admission turbine.
PU Used in Admission Mode Selection function.

KAK_1 Stress decay coefficient at medium RPM.
PU Used in Acceleration function.

KAK_2 Stress decay coefficient at high RPM.
PU Used in Acceleration function.

KAK_3 Stress decay coefficient at rated RPM.
PU Used in Acceleration function.

KALF0 Thermal expansion coefficient.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-05 in its implementation. Thus, the number entered will be divided by 100,000 before being used in the algorithm. For example, if the value 0.5700 was entered for the constant, the value 0.5700E-05 would be used by the algorithm.

KALF1 Thermal expansion coefficient.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-08 in its implementation. Thus, the number entered will be divided by 100,000,000 before being used in the algorithm. For example, if the value 0.5150 was entered for the constant, the value 0.5150E-08 would be used by the algorithm.

KALF2 Thermal expansion coefficient.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-11 in its implementation. Thus, the number entered will be divided by 100,000,000,000 before being used in the algorithm. For example, if the value -0.1790 was entered for the constant, the value -0.1790E-11 would be used by the algorithm.

KALF3 Thermal expansion coefficient.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this consant has an assumed exponent of E-15 in its implementation. Thus, the number entered will be divided by 1,000,000,000,000,000 before being used by the algorithm. For example, if the value 0.9450 was entered for the constant, the value 0.9450E-15 would be used by the algorithm.

KAMS1 Stress level for switch divider AMS_DIV.
PU Used in Admission Mode Selection function.

KAMS2 Used in controlling stress SC.
PU Used in Admission Mode Selection function.

KAMS3 Used in deriving stress limit for unloading.
PU Used in Admission Mode Selection function.

KAMS4 Partial arc bias in VPO.
. . . Used in Admission Mode Selection function.

KAMS5 Maximum transfer rate to full arc while loading.
. . . Used in Admission Mode Selection function.

KAMS6 Maximum negative stress limit for transfer to PA.
PU Used in Admission Mode Selection function.

KAMS7 Maximum positive stress limit for transfer to PA .
PU Used in Admission Mode Selection function.

KATS_A_1_1 Surface lookahead stress coefficient at medium speed for area one.
. . . Used in Acceleration function.

KATS_A_1_2 Surface lookahead stress coefficient at medium speed for area two.
. . . Used in Acceleration function.

KATS_A_1_3 Surface lookahead stress coefficient at medium speed for area three.
. . . Used in Acceleration function.

KATS_A_2_1 Surface lookahead stress coefficient at high speed for area one.
. . . Used in Acceleration function.

KATS_A_2_2 Surface lookahead stress coefficient at high speed for area two.
. . . Used in Acceleration function.

KATS_A_2_3 Surface lookahead stress coefficient at high speed for area three.
. . . Used in Acceleration function.

KATS_A_3_1 Surface lookahead stress coefficient at rated speed for area one.
. . . Used in Acceleration and Temperature Recommendations functions.

KATS_A_3_2 Surface lookahead stress coefficient at rated speed for area two.
. . . Used in Acceleration and Temperature Recommendations functions.

KATS_A_3_3 Surface lookahead stress coefficient at rated speed for area three.
. . . Used in Acceleration function.

KATS_B_1_1 Bore lookahead stress coefficient at medium speed for area one.
. . . Used in Acceleration function.

KATS_B_1_2 Bore lookahead stress coefficient at medium speed for area two.
. . . Used in Acceleration function.

KATS_B_1_3 Bore lookahead stress coefficient at medium speed for area three.
. . . Used in Acceleration function.

KATS_B_2_1 Bore lookahead stress coefficient at high speed for area one.
. . . Used in Acceleration function.

KATS_B_2_2 Bore lookahead stress coefficient at high speed for area two.
. . . Used in Acceleration function.

KATS_B_2_3 Bore lookahead stress coefficient at high speed for area three.
. . . Used in Acceleration function.

KATS_B_3_1 Bore lookahead stress coefficient at rated speed for area one.
. . . Used in Acceleration and Temperature Recommendations functions.

KATS_B_3_2 Bore lookahead stress coefficient at rated speed for area two.
. . . Used in Acceleration and Temperature Recommendations functions.

KATS_B_3_3 Bore lookahead stress coefficient at rated speed for area three.
. . . Used in Acceleration function.

KATS_BA_1 S1 mismatch coefficient in bypass mode for area one.
. . . Used in Acceleration function.

KATS_BA_2 S1 mismatch coefficient in bypass mode for area two.
. . . Used in Acceleration function.

KATS_BA_3	S1 mismatch coefficient in bypass mode for area three. . . . Used in Acceleration and Temperature Recommendations functions.	
KATS_BB_1	S6 mismatch coefficient in bypass mode for area one. . . . Used in Acceleration function.	
KATS_BB_2	S6 mismatch coefficient in bypass mode for area two. . . . Used in Acceleration function.	
KATS_BB_3	S6 mismatch coefficient in bypass mode for area three. . . . Used in Acceleration and Temperature Recommendations functions.	
KATS_CONFIG	ATS configuration. CNT15 Used in Scheduling function. Zero will inhibit all functions.	
KBAL_1	Allowable bore temperature for Rotor Stress function. DEG F Used in Rotor Stress function.	
KBAL_2	Allowable bore stress (raw) if bore temperature is within allowable. 10**3 Used in Rotor Stress function.	
KBAL_3	Temperature to stress conversion coefficient if bore temperature is within allowable. . . . Used in Rotor Stress function.	
KBAL_4	Allowable bore stress (raw) if bore temperature has exceeded allowable. 10**3 Used in Rotor Stress function.	
KBAL_5	Temperature to stress conversion coefficient if bore temperature has exceeded allowable. . . . Used in Rotor Stress function.	
KBOREDIA1	Rotor bore diameter in area one. in. Used in Rotor Stress function. Note that this is an initialization control constant.	
KBOREDIA2	Rotor bore diameter in area two. in. Used in Rotor Stress function. Note that this is an initialization control constant.	
KBOREDIA3	Rotor bore diameter in area three. in. Used in Rotor Stress function. Note that this is an initialization control constant.	
KC0_1	Coefficient for cyclic life expenditure calculation in area one. PU Used in Rotor Stress function. Note that this is an initialization control constant.	
KC0_2	Coefficient for cyclic life expenditure calculation in area two. PU Used in Rotor Stress function. Note that this is an initialization control constant.	

KC0_3 Coefficient for cyclic life expenditure calculation in area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant.

KC1_1 Coefficient for cyclic life expenditure calculation in area one.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-04 in it's implementation. Thus, the number entered will be divided by 10,000 before being used in the algorithm. For example, if the value 0.1029 was entered for the constant, the value 0.1029E-04 would be used by the algorithm.

KC1_2 Coefficient for cyclic life expenditure calculation in area two.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-04 in it's implementation. Thus, the number entered will be divided by 10,000 before being used in the algorithm. For example, if the value 0.1076 was entered for the constant, the value 0.1076E-04 would be used by the algorithm.

KC1_3 Coefficient for cyclic life expenditure calculation in area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-04 in it's implementation. Thus, the number entered will be divided by 10,000 before being used in the algorithm. For example, if the value 0.0768 was entered for the constant, the value 0.0768E-04 would be used by the algorithm.

KC2_1 Coefficient for cyclic life expenditure calculation in area one.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-09 in it's implementation. Thus, the number entered will be divided by 1,000,000,000 before being used in the algorithm. For example, if the value -0.2134 was entered for the constant, the value -0.2134E-09 would be used by the algorithm.

KC2_2 Coefficient for cyclic life expenditure calculation in area two.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-09 in it's implementation. Thus, the number entered will be divided by 1000,000,000 before being used in the algorithm. For example, if the value -0.2336 was entered for the constant, the value -0.2336E-09 would be used by the algorithm.

KC2_3 Coefficient for cyclic life expenditure calculation in area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-09 in it's implementation. Thus, the number entered will be divided by 1000,000,000 before being used in the algorithm. For example, if the value -0.2085 was entered for the constant, the value -0.2085E-09 would be used by the algorithm.

KC3_1 Coefficient for cyclic life expenditure calculation in area one.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-14 in it's implementation. Thus, the number entered will be divided by 10,000,000 before being used in the algorithm. For example, if the value 0.1599 was entered for the constant, the value 0.1599E-14 would be used by the algorithm.

KC3_2 Coefficient for cyclic life expenditure calculation in area two.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-14 in it's implementation. Thus, the number entered will be divided by 10,000,000 before being used in the algorithm. For example, if the value 0.1831 was entered for the constant, the value 0.1831E-14 would be used by the algorithm.

KC3_3 Coefficient for cyclic life expenditure calculation in area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-14 in it's implementation. Thus, the number entered will be divided by 10,000,000 before being used in the algorithm. For example, if the value 0.2142 was entered for the constant, the value 0.2142E-14 would be used by the algorithm.

KCRITERIA1 Criteria to evaluate in area one.
CNT15 Used in Scheduling function.
Must be a value from one to two, inclusive.

KCRITERIA2 Criteria to evaluate in area two.
CNT15 Used in Scheduling function.
Must be a value from one to two, inclusive.

KCRITERIA3 Criteria to evaluate in area three.
CNT15 Used in Scheduling function.
Must be a value from one to six, inclusive.

KCRITERIA4 Criteria to evaluate in area four.
CNT15 Used in Scheduling function.
Must be a value from one to six, inclusive. Reserved for future enhancements.

KCT_1 Coefficient for cyclic life expenditure calculation in area one.
10**3 Used in Rotor Stress function.
Note that this is an initialization control constant.

KCT_2 Coefficient for cyclic life expenditure calculation in area two.
10**3 Used in Rotor Stress function.
Note that this is an initialization control constant.

KCT_3 Coefficient for cyclic life expenditure calculation in area three.
10**3 Used in Rotor Stress function.
Note that this is an initialization control constant.

KDEL_CV_1 Allowable CV temperature differential breakpoint during loading.
DEG F Used in Preparation for Rolloff and Average Temperature Change functions.
Note that this is an initialization control constant.

KDEL_CV_2 Allowable CV temperature differential intercept for values greater than the breakpoint during loading.
DEG F Used in Preparation for Rolloff and Average Temperature Change functions.
Note that this is an initialization control constant.

KDEL_CV_3 Allowable CV temperature differential slope for values greater than the breakpoint during loading.
PU Used in Preparation for Rolloff and Average Temperature Change functions.
Note that this is an initialization control constant.

KDEL_CV_4 Allowable CV temperature differential intercept for values less than the breakpoint during loading.
DEG F Used in Preparation for Rolloff and Average Temperature Changefunctions.
Note that this is an initialization control constant.

KDEL_CV_5 Allowable CV temperature differential slope for values less than the breakpoint during loading.
PU Used in Preparation for Rolloff and Average Temperature functions.
Note that this is an initialization control constant.

KDEL_CVU_2 Allowable CV temperature differential intercept for values greater than the breakpoint during unloading.
DEG F Used in Average Temperature Change function.
Note that this is an initialization control constant.

KDEL_CVU_3 Allowable CV temperature differential slope for values greater than the breakpoint during unloading.
PU Used in Average Temperature Change function.
Note that this is an initialization control constant.

KDEL_CVU_4 Allowable CV temperature differential intercept for values less than the breakpoint during unloading.
DEG F Used in Average Temperature Change function.
Note that this is an initialization control constant.

KDEL_CVU_5 Allowable CV temperature differential slope for values greater than the breakpoint during unloading.
PU Used in Average Temperature Change function.
Note that this is an initialization control constant.

KDEL_SV_2 Allowable SV temperature differential intercept for values greater than the breakpoint during loading.
DEG F Used in Average Temperature Change function.
Note that this is an initialization control constant.

KDEL_SV_3 Allowable SV temperature differential slope for values greater than the breakpoint during loading.
PU Used in Preparation for Rolloff and Average Temperature Change functions.
Note that this is an initialization control constant.

KDEL_SV_4 Allowable SV temperature differential intercept for values less than the breakpoint during loading.
DEG F Used in Average Temperature Change function.
Note that this is an initialization control constant.

KDEL_SV_5 Allowable SV temperature differential slope for values greater than the breakpoint during loading.
PU Used in Average Temperature Change function.
Note that this is an initialization control constant.

KDF0_1 Coefficient for temperature dependent thermal diffusivity for area one.
PU Used in Rotor Stress function.
Note that this is an initialization control constant.

KDF0_2 Coefficient for temperature dependent thermal diffusivity for area two.
PU Used in Rotor Stress function.
Note that this is an initialization control constant.

KDF0_3 Coefficient for temperature dependent thermal diffusivity for area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant.

KDF1_1 Coefficient for temperature dependent thermal diffusivity for area one.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-03 in it's implementation. Thus, the number entered will be divided by 1000 before being used in the algorithm. For example, if the value -0.2201 was entered for the constant, the value -0.2201E-03 would be used by the algorithm.

KDF1_2 Coefficient for temperature dependent thermal diffusivity for area two.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-03 in it's implementation. Thus, the number entered will be divided by 1000 before being used in the algorithm. For example, if the value -0.2201 was entered for the constant, the value -0.2201E-03 would be used by the algorithm.

KDF1_3 Coefficient for temperature dependent thermal diffusivity for area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-03 in it's implementation. Thus, the number entered will be divided by 1000 before being used in the algorithm. For example, if the value -0.2201 was entered for the constant, the value -0.2201E-03 would be used by the algorithm.

KDF2_1 Coefficient for temperature dependent thermal diffusivity for area one.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-07 in it's implementation. Thus, the number entered will be divided by 10,000,000 before being used in the algorithm. For example, if the value 0.7410 was entered for the constant, the value 0.7410E-07 would be used by the algorithm.

KDF2_2 Coefficient for temperature dependent thermal diffusivity for area two.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-07 in it's implementation. Thus, the number entered will be divided by 10,000,000 before being used in the algorithm. For example, if the value 0.7410 was entered for the constant, the value 0.7410E-07 would be used by the algorithm.

KDF2_3 Coefficient for temperature dependent thermal diffusivity for area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-07 in it's implementation. Thus, the number entered will be divided by 10,000,000 before being used in the algorithm. For example, if the value 0.7410 was entered for the constant, the value 0.7410E-07 would be used by the algorithm.

KDF3_1 Coefficient for temperature dependent thermal diffusivity for area one.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-10 in it's implementation. Thus, the number entered will be divided by 10,000,000,000 before being used in the algorithm. For example, if the value -0.9374 was entered for the constant, the value -0.9374E-10 would be used by the algorithm.

KDF3_2 Coefficient for temperature dependent thermal diffusivity for area two.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-10 in it's implementation. Thus, the number entered will be divided by 10,000,000,000 before being used in the algorithm. For example, if the value -0.9374 was entered for the constant, the value -0.9374E-10 would be used by the algorithm.

KDF3_3 Coefficient for temperature dependent thermal diffusivity for area three.
PU Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E-10 in it's implementation. Thus, the number entered will be divided by 10,000,000,000 before being used in the algorithm. For example, if the value -0.9374 was entered for the constant, the value -0.9374E-10 would be used by the algorithm.

KEM1 Coefficient for modulus of elasticity.
. . . Used in Rotor Stress function.
Note that this is an initialization control constant.

KEM2 Coefficient for modulus of elasticity.
. . . Used in Rotor Stress function.
Note that this is an initialization control constant.

KEM3 Coefficient for modulus of elasticity.
. . . Used in Rotor Stress function.
Note that this is an initialization control constant.

KEMERG_SET Emergency governor setting.
rpm Used in Acceleration function.

KEMODCST_1 Modulus of elasticity for area one.
CNT15 Used in Rotor Stress function.
This constant has an assumed exponent of E+04 in it's implementation. Thus, the number entered will be multiplied by 10,000 before being used in the algorithm. For example, if the value 2600.0 was entered for the constant, the value 2600.0E+04, or 26,000,000, would be used by the algorithm.

KEMODCST_2 Modulus of elasticity for area two.
CNT15 Used in Rotor Stress function.
This constant has an assumed exponent of E+04 in it's implementation. Thus, the number entered will be multiplied by 10,000 before being used in the algorithm. For example, if the value 2600.0 was entered for the constant, the value 2600.0E+04, or 26,000,000, would be used by the algorithm.

KEMODCST_3 Modulus of elasticity for area three.
CNT15 Used in Rotor Stress function.
This constant has an assumed exponent of E+04 in it's implementation. Thus, the number entered will be multiplied by 10,000 before being used in the algorithm. For example, if the value 2600.0 was entered for the constant, the value 2600.0E+04, or 26,000,000, would be used by the algorithm.

KEM0 Coefficient for modulus of elasticity.
10**3 Used in Rotor Stress function.
Note that this is an initialization control constant. Also, this constant has an assumed exponent of E+03 in it's implementation. Thus, the number entered will be multiplied by 1,000 before being used in the algorithm. For example, if the value 30847 was entered for the constant, the value 30.847E+06 would be used by the algorithm.

KFB_1	Bore factor for determining initial loading rate for area one.	
	. . .	Used in Loading Rate function.
	Note that this is an initialization control constant.	
KFB_2	Bore factor for determining initial loading rate for area two.	
	. . .	Used in Loading Rate function.
	Note that this is an initialization control constant.	
KFB_3	Bore factor for determining initial loading rate for area three.	
	. . .	Used in Loading Rate function.
	Note that this is an initialization control constant.	
KFNMAX_1	Maximum limit on negative factor for area one.	
	PU	Used in Temperature Recommendations function.
KFNMAX_2	Maximum limit on negative factor for area two.	
	PU	Used in Temperature Recommendations function.
KFNMIN_1	Minimum limit on negative factor for area one.	
	PU	Used in Temperature Recommendations function.
KFNMIN_2	Minimum limit on negative factor for area two.	
	PU	Used in Temperature Recommendations function.
KFPOSB_1	Factor in calculating bore related positive temperature mismatch for area one.	
	PU	Used in Temperature Recommendations function.
KFPOSB_2	Factor in calculating bore related positive temperature mismatch for area two.	
	PU	Used in Temperature Recommendations function.
KFPOSS_1	Factor in calculating surface related positive temperature mismatch for area one.	
	PU	Used in Temperature Recommendations function.
KFPOSS_2	Factor in calculating surface related positive temperature mismatch for area two	
	PU	Used in Temperature Recommendations function.
KFR1	Maximum stress for stress level term.	
	PU	Used in Loading Rate function.
KFR2	Divider for temperature rate of change term.	
	. . .	Used in Loading Rate function.
KFR3	Multiplier in stress rate of change term.	
	. . .	Used in Loading Rate function.
KFR4	Minimum/Maximum increase in loading rate.	
	% / min	Used in Loading Rate function.
KFR5	Maximum stress above which there is no AMS change.	
	PU	Used in Loading Rate function.
KFS_1	Surface factor for determining initial loading rate for area one.	
	PU	Used in Loading Rate function.
	Note that this is an initialization control constant.	

KFS_2 Surface factor for determining initial loading rate for area two.
PU Used in Loading Rate function.
Note that this is an initialization control constant.

KFS_3 Surface factor for determining initial loading rate for area three.
PU Used in Loading Rate function.
Note that this is an initialization control constant.

KFTEMP_LIM Generator field temperature limit.
DEG C Used in Generator Monitor function.

KH2PRES_1 Hydrogen pressure point one.
PSI Used in Generator Monitor function.

KH2PRES_2 Hydrogen pressure point two.
PSI Used in Generator Monitor function.

KH2PRES_3 Hydrogen pressure point three.
PSI Used in Generator Monitor function.

KH2PRES_4 Hydrogen pressure point four.
PSI Used in Generator Monitor function.

KIAL_1 Armature current limit at pressure point one.
AMPS Used in Generator Monitor function.

KIAL_2 Armature current limit at pressure point two.
AMPS Used in Generator Monitor function.

KIAL_3 Armature current limit at pressure point three.
AMPS Used in Generator Monitor function.

KIAL_4 Armature current limit at pressure point four.
AMPS Used in Generator Monitor function.

KIFL_1 Field current limit at pressure point one.
AMPS Used in Generator Monitor function.

KIFL_2 Field current limit at pressure point two.
AMPS Used in Generator Monitor function.

KIFL_3 Field current limit at pressure point three.
AMPS Used in Generator Monitor function.

KIFL_4 Field current limit at pressure point four.
AMPS Used in Generator Monitor function.

KK2_1 Centrifugal Stress at rated speed for area one.
10**3 Used in Acceleration, Loading Monitor, Temperature Recommendations, and Rotor Stress functions.
Note that this is an initialization control constant.

KK2_2 Centrifugal Stress at rated speed for area two.
10**3 Used in Acceleration, Loading Monitor, Temperature Recommendations, and Rotor Stress functions.
Note that this is an initialization control constant.

KK2_3 Centrifugal Stress at rated speed for area three.
10**3 Used in Acceleration, Loading Monitor, Temperature Recommendations, and Rotor Stress functions.
Note that this is an initialization control constant.

KKLAB Coefficient for bore stress permissive to transfer to forward flow.
. . . Used in Loading Monitor function.

KKLAS Coefficient for surface stress permissive to transfer to forward flow.
. . . Used in Loading Monitor function.

KLAMS_CALC Enables Admission Mode Selection function.
CNT15 Used in Admission Mode Selection function.
Zero disables the function, any other value enables.

KLCST_PWRM Enables chest prewarming hold logic.
CNT15 Used in Preparation for Rolloff function.
Non-zero to enable holds based on chest prewarming.

KLD1 Temperature to stress conversion constant.
. . . Used in Loading Monitor function.

KLD2 Temperature intersect at zero stress.
DEG F Used in Loading Monitor function.

KLGEN_CALC Enables Generator Monitor function.
CNT15 Used in Generator Monitor function.
Zero disables the function, any other value enables.

KLMIN_LOAD Enables minimum load no-hold logic.
CNT15 Used in Loading Monitor function.
Non-zero enables check that minimum load is achieved before permitting load holds.

KLOCATIONS Number of location to evaluate.
CNT15 Used in Scheduling function.
Must be a value from one to three, inclusive.

KLPRE_STRS Enables high stress in preroll hold logic.
CNT15 Used in Preparation for Rolloff function.
Non-zero to enable holds based on high stresses during preroll.

KLRTR_PWRM Enables rotor prewarming hold logic.
CNT15 Used in Preparation for Rolloff function.
Non-zero to enable holds based on rotor prewarming.

KLTMP_LOAD Enables valve temperature differential hold logic.
CNT15 Used in Loading Monitor function.
Non-zero enables check of stop and control valve temperatures to cause load hold.

KMIN_LOAD Minimum limit for load hold selection.
% Used in Loading Monitor function.

KMINMAXT_1 Main steam min/max coefficient.
CNT15 Used in Temperature Recommendations function.

KMINMAXT_2 Reheat steam min/max coefficient.
CNT15 Used in Temperature Recommendations function.

KMINRTMP1 Minimum rotor bore temperature before rolloff in area one.
DEG F Used in Preparation for Rolloff function.

KMINRTMP2 Minimum rotor bore temperature before rolloff in area two.
DEG F Used in Preparation for Rolloff function.

KMINRTMP3 Minimum rotor bore temperature before rolloff in area three.
DEG F Used in Preparation for Rolloff function.

KOS_MARGIN Margin for over speed test.
PU Used in Acceleration function.

KOS_STMARGIN Margin on over stressing.
PU Used in Acceleration function.

KPOISSON Poisson coefficient.
PU Used in Rotor Stress function.

KPREP1 CV coefficient for slow pressurization.
PU Used in Preparation for Rolloff function.

KPREP2 CV coefficient for chest warming.
PU Used in Preparation for Rolloff function.

KPREP3 Pressure coefficient for chest pressurization.
PU Used in Preparation for Rolloff function.

KPREP4 CV coefficient for heat soaking.
PU Used in Preparation for Rolloff function.

KPREP5 Pressure coefficient for chest pressurization.
PU Used in Preparation for Rolloff function.

KPREP6 CV chest warming coefficient.
PU Used in Preparation for Rolloff function.

KRATCUR Rated current.
CNT15 Used in Generator Monitor function.

KRATLOAD Rated load.
MW Used in Loading Monitor and Loading Rate functions.

KRATMSP Rated main steam (inlet) pressure.
CNT15 Used in Preparation for Rolloff, Temperature Mismatch and Recommendations functions.

KRATMST Rated main steam (inlet) temperature.
DEG F Used in Temperature Recommendations function.

KRATRHT Rated hot reheat steam temperature.
DEG F Used in Temperature Recommendations function.

KRATSPD Rated speed in RPM.
RPM Used in Rotor Stress function.

KRESIST Resistance coefficient used in calculating field temperature.
PU Used in Generator Monitor function.

KRLIM Breakpoint for 3 or 4 admission turbine.
. . . Used in Admission Mode Selection function.

KS2N_1_1 Allowable surface stress for area one with low CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_1_2 Allowable surface stress for area one with normal CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_1_3 Allowable surface stress for area one with high CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_2_1 Allowable surface stress for area two with low CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_2_2 Allowable surface stress for area two with normal CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_2_3 Allowable surface stress for area two with high CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_3_1 Allowable surface stress for area three with low CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_3_2 Allowable surface stress for area three with normal CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KS2N_3_3 Allowable surface stress for area three with high CLE selection.
10**3 Used in Rotor Stress and Temperature Recommendations functions.
Note that this is an initialization control constant.

KSAL_1	Coefficient for surface stress calculation. PU Used in Rotor Stress function.
KSAL_2	Coefficient for surface stress calculation. PU Used in Rotor Stress function.
KSAL_3	Coefficient for surface stress calculation. PU Used in Rotor Stress function.
KSAL_4	Coefficient for surface stress calculation. PU Used in Rotor Stress function.
KSHFTDIA	Shaft diameter. IN Used in Rotor Stress function. Note that this is an initialization control constant.
KSTABL_TM	Stabilization time in seconds. CNT15 Used in Scheduling function. May not be zero, and should be greater than KSTART_TM.
KSTART_TM	Start up time in seconds. CNT15 Used in Scheduling function. May not be zero.
KSURFDIA1	Rotor surface diameter in area one. in. Used in Rotor Stress function. Note that this is an initialization control constant.
KSURFDIA2	Rotor surface diameter in area two. in. Used in Rotor Stress function. Note that this is an initialization control constant.
KSURFDIA3	Rotor surface diameter in area three. in. Used in Rotor Stress function. Note that this is an initialization control constant.
KTURBTYPE	Turbine type. CNT15 Used in Scheduling function. 1=LST 2=NucMono 3=Nuc 4=MSTM, all other values are invalid.
KVMULT	Multiplier for initial loading rate. PU Used in Loading Rate function.
KWRIMDIA	Wheel rim diameter. in. Used in Rotor Stress function. Note that this is an initialization control constant.
TNKR_HI	High speed setpoint. % Used in Acceleration function.
TNKR_LO	Low speed setpoint. % Used in Acceleration function.
TNKR_ME	Medium speed setpoint % Used in Acceleration function.

F-3. LIST OF INDIVIDUAL POINTS (I/O REQUIREMENTS)

The following is a list of entries for individual points.

AMS_RATE_REC	AMS transfer rate for display.	
	%/MIN	Used in Admission Mode Selection function.
AP_H2_ATS	Hydrogen pressure.	
	PSI	Used in Generator Monitor function.
ARM_CUR_LIM	Armature current limit.	
	PU	Used in Generator Monitor function.
AT_H2CG	Hydrogen cold gas temperature.	
	DEG C	Used in Generator Monitor function.
BORE_LA_1	Bore lookahead stress for area one.	
	%	Used in Rotor Stress function.
BORE_LA_2	Bore lookahead stress for area two.	
	%	Used in Rotor Stress function.
BORE_LA_3	Bore lookahead stress for area three.	
	%	Used in Rotor Stress function.
BORE_STR_1	Bore stress for area one.	
	%	Used in Rotor Stress function.
BORE_STR_2	Bore stress for area two.	
	%	Used in Rotor Stress function.
BORE_STR_3	Bore stress for area three.	
	%	Used in Rotor Stress function.
CHST_MSP_PCT	Chest pressure to main steam pressure ratio.	
	%	Used in preparation for Rolloff function.
CLE_HP	Cyclic life expenditure on last thermal cycle for rotor in area one.	
	%	Used in Rotor Stress function.
CLE_RH	Cyclic life expenditure on last thermal cycle for rotor in area two.	
	%	Used in Rotor Stress function.
CLE_XO	Cyclic life expenditure on last thermal cycle for rotor in area three.	
	%	Used in Rotor Stress function.
CP_CHST_ATS	Steam chest pressure.	
	PSI	Used in preparation for Rolloff function.

CVAMS_AU	Admission mode transfer rate.	
	PU/S	Used in Admission Mode Selection function.
CVR	Control valve reference.	
	%	Used in Admission Mode Selection function.
DA_PU_AV	Average armature current.	
	PU	Used in Generator Monitor function.
DAF_ATS	Generator field current.	
	AMPS	Used in Generator Monitor function.
DAR_ATS	Required change of admission reference, raw (before clamping).	
	. . .	Used in Admission Mode Selection function.
DBZONE_HP_1	Bore stress excursion counter for zone I for area one.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_HP_2	Bore stress excursion counter for zone I for area two.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_HP_3	Bore stress excursion counter for zone I for area three.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_RH_1	Bore stress excursion counter for zone II for area one.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_RH_2	Bore stress excursion counter for zone II for area two.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_RH_3	Bore stress excursion counter for zone II for area three.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_XO_1	Bore stress excursion counter for zone III for area one.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_XO_2	Bore stress excursion counter for zone III for area two.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DBZONE_XO_3	Bore stress excursion counter for zone III for area three.	
	CNT15	Used in Rotor Stress and EEPROM Access functions.
DS7_ATS_1	Rate of change of normalized surface stress for area one.	
	. . .	Used in Rotor Stress function.
DS7_ATS_2	Rate of change of normalized surface stress for area two.	
	. . .	Used in Rotor Stress function.
DS7_ATS_3	Rate of change of normalized surface stress for area three.	
	. . .	Used in Rotor Stress function.

DS9_ATS_1 Rate of change of normalized bore stress for area one.
. . . Used in Rotor Stress function.

DS9_ATS_2 Rate of change of normalized bore stress for area two.
. . . Used in Rotor Stress function.

DS9_ATS_3 Rate of change of normalized bore stress for area three.
. . . Used in Rotor Stress function.

DTMS_ATS Main steam temperature rate of change.
F/MIN Used in Average Temperature Change function.

DTRH_ATS; Reheat steam temperature rate of change.
F/MIN Used in Average Temperature Change function.

DVF_ATS Generator field voltage.
V DC Used in Generator Monitor function.

DW_TTA_ATS UMC load target.
% Used in Admission Mode Selection function.

DWR_R1 MW reference rate.
%/SEC Used in Admission Mode Selection function.

DWR_TARR_MAX Recommended maximum loading rate reference.
%/SEC Used in Loading Rate function.

DWR_TAR2 Target reference load.
% Used in Admission Mode Selection function.

DW1 MW load.
% Used by Admission Mode Selection, Loading Rate, and Loading Monitor functions.

E_FT_MST_REC Main steam temperature condition.
ENM23 Used in Temperature Recommendations function.

E_FT_RHT_REC Reheat steam temperature condition.
ENM23 Used in Temperature Recommendations function.

E_L18ACC_REC Recommended acceleration rate.
ENM21 Used in Acceleration function.

E_L83AMS_REC Recommended admission mode.
ENM22 Used in Admission Mode Selection function.

FLD_CUR_LIM Field current limit.
AMPS Used in Generator Monitor function.

FLD_TEMP Field temperature.
DEG C Used in Generator Monitor function.

FP_MSP1 Main steam pressure.
% Used in Preparation for Rolloff, Temperature Recommendations, Temperature Mismatch and Loading Rate functions.

FP_HRSP_DPY Reheat steam pressure.
PSI Used in Temperature Mismatch function.

FT_HRS Hot reheat steam temperature.
DEG F Used in Average Temperature Change, Temperature Mismatch and Temperature Recommendations functions.

FT_HRS_MAX Maximum recommended reheat steam temperature.
DEG F Used in Temperature Recommendations function.

FT_HRS_MIN Minimum recommended reheat steam temperature.
DEG F Used in Temperature Recommendations function.

FT_MSP Main steam temperature.
DEG F Used in Preparation for Rolloff, Average Temperature Change, Temperature Mismatch and Temperature Recommendations functions.

HEAT_RATE_1 Heating rate in area one.
F/MIN Used in preparation for Rolloff function.

HEAT_RATE_2 Heating rate in area two.
F/MIN Used in preparation for Rolloff function.

HEAT_RATE_3 Heating rate in area three.
F/MIN Used in preparation for Rolloff function.

IVR Intercept valve reference.
% Used by Loading Monitor function.

LARM_CUR_EXC Excessive armature current.
LOG Used in Generator Monitor function.

LAT_H2CG_EXC Excessive H2 cold gas temperature.
LOG Used in Generator Monitor function.

LBZCOUNT Excursion past stress limits occurred.
LOG Used in Rotor Stress function.

LCALC_TEMP Enables field temperature calculation.
LOG Used in Generator Monitor function.

LCALCS_STAB Stress calculations are stable.
LOG Used in Scheduling function.

LCHST_RDYRL Control valve chest ready for rolloff.
LOG Used in Preparation for Rolloff function.

LCHST_WRM_CM Chest warming is finished.
LOG Used in Preparation for Rolloff function.

LCHT_PRE_REQ Chest prewarming required.
LOG Used in Preparation for Rolloff function.

LCV_CHST_HLD Control valve chest prewarming required.
LOG Used in Preparation for Rolloff function.

LENTHALPY_OK Throttle enthalpy is OK.
LOG Used in Temperature Mismatch function.

LEXC_LA_STR Excessive lookahead stress.
LOG Used in Acceleration function.

LEXC_VLV_DIF Excessive valve temperature differential.
LOG Used by Loading Monitor function.

LEXCSTR_PR Excessive stress during preroll.
LOG Used in Preparation for Rolloff function.

LFLD_CUR_EXC Excessive field current.
LOG Used in Generator Monitor function.

LFLD_TMP_EXC Excessive field temperature.
LOG Used in Generator Monitor function.

LHEAT_SK_REQ Heat soaking required.
LOG Used in Preparation for Rolloff function.

LMST_OUT_RAN Main steam temperature is out of range.
LOG Used in Temperature Recommendations function.

LOAD_RAT_REC Recommended loading rate.
MW/MIN Used in Loading Rate function.

LOADRATE_ATS Recommended load rate, raw (stress-based derivation only).
. . . Used in Loading Rate function.

LPRESS_OVER Chest pressurization is finished.
LOG Used in Preparation for Rolloff function.

LRHT_OUT_RAN Reheat steam temperature is out of range.
LOG Used in Temperature Recommendations function.

LROT_HS_REQ High stress hold request.
LOG Used by Loading Monitor function.

LROT_PRE_REQ Rotor Prewarming required.
LOG Used in Preparation for Rolloff function.

LRTR_WRM_REQ Rotor warming required.
LOG Used in Acceleration function.

LSLO_PRS_REQ Slow pressurization is required.
LOG Used in Preparation for Rolloff function.

LSURFHEAT1 Excessive surface heating in area one.
LOG Used in Rotor Stress function.

LSURFHEAT2 Excessive surface heating in area two.
LOG Used in Rotor Stress function.

LSURFHEAT3	Excessive surface heating in area three.	
	LOG	Used in Rotor Stress function.
L14HA	Above rolloff speed.	
	LOG	Used in Scheduling and Acceleration functions.
L14RTX	Turbine at rated speed.	
	LOG	Used in Acceleration function.
L3_OS_ATS	Over speed test permissive.	
	LOG	Used in Acceleration function.
L3DWR_LDG	Load increasing.	
	LOG	Used in Admission Mode Selection function.
L3DWR_UNL	Load decreasing.	
	DEG F	Used in Admission Mode Selection, Average Temperature Change,
	Loading Rate, and Loading Monitor functions.	
L3XT_ATS	TC readings are invalid.	
	LOG	Used in Scheduling function.
L39DE_LW	Differential expansion long warning.	
	LOG	Used in Admission Mode Selection function.
L43A_HIR_ATS	Fast acceleration request.	
	LOG	Used in Acceleration function.
L43A_LOR_ATS	Slow acceleration request.	
	LOG	Used in Acceleration function.
L43A_MER_ATS	Medium acceleration request.	
	LOG	Used in Acceleration function.
L43AMS_F_ATS	Request to go to full arc.	
	LOG	Used in Admission Mode Selection function.
L43AMS_P_ATS	Request to go to partial arc.	
	LOG	Used in Admission Mode Selection function.
L43FFR_ATS	Request to go to forward flow.	
	LOG	Used by Loading Monitor function.
L43N_HIR_ATS	High RPM target request.	
	LOG	Used in Acceleration function.
L43N_MER_ATS	Medium RPM target request.	
	LOG	Used in Acceleration function.
L43N_RTR_ATS	Rated RPM target request.	
	LOG	Used in Acceleration function.
L52GX_ATS	Breaker closed.	
	LOG	Used in Scheduling and Temperature Mismatch functions.

L68DW Load/speed hold.
LOG Used in Acceleration and Loading Monitor functions.

L83A_HI Select high acceleration speed.
LOG Used in Acceleration function.

L83A_LO Select low acceleration speed.
LOG Used in Acceleration function.

L83A_ME Select medium acceleration speed.
LOG Used in Acceleration function.

L83CLE_HI High cyclic life expenditure selection.
LOG Used in Rotor Stress and Temperature Recommendations functions.

L83CLE_LO Low cyclic life expenditure selection.
. . . Used in Rotor Stress function.

L83CLE_ME Medium cyclic life expenditure selection.
LOG Used in Rotor Stress and Temperature Recommendations functions.

L83CW Chest warming mode.
LOG Used in Preparation for Rolloff function.

L83FF Forward flow state.
LOG Used in Acceleration, Admission Mode Selection, TemperatureRecommendations, Temperature Mismatch and Loading Monitor functions.

L83M_AU Automatic mode.
LOG Used in Acceleration, Admission Mode Selection, Loading Rate, and Loading Monitor functions.

L83M_BYP Turbine bypass mode.
LOG Used in Acceleration and Loading Monitor functions.

L83M_CO Remote submode.
LOG Used in Admission Mode Selection function.

L83M_UMC Unit Master Control in control.
LOG Used in Admission Mode Selection function.

L83M_VPO Variable Pressure Operation mode.
LOG Used in Admission Mode Selection function.

L83N_CL Select close valves.
LOG Used in Acceleration function.

L83N_HI Select high target speed.
LOG Used in Acceleration function.

L83N_ME Select medium target speed.
LOG Used in Acceleration function.

L83SF Split flow mode.
LOG Used in Acceleration and Temperature Mismatch functions.

L83STAB	Stabilizing mode; stress calculations are not stable. LOG Used in Scheduling and Rotor Stress functions.	
MSOTR_DIFTAR	Main steam and CV chest outer temperature differential target. DEG F Used in preparation for Rolloff function.	
MST_OUTR_DIF	Main steam and CV chest outer temperature differential. DEG F Used in preparation for Rolloff function.	
RLE_HP	Accumulated cyclic life expenditure for area one. % Used in Rotor Stress and EEPROM Access functions.	
RLE_RH	Accumulated cyclic life expenditure for area two. % Used in Rotor Stress and EEPROM Access functions.	
RLE_XO	Accumulated cyclic life expenditure for area three. % Used in Rotor Stress and EEPROM Access functions.	
ROTOR_1_TEMP	Bore temperature for area one. DEG F Used in Rotor Stress function.	
ROTOR_2_TEMP	Bore temperature for area two. DEG F Used in Rotor Stress function.	
ROTOR_3_TEMP	Bore temperature for area three. DEG F Used in Rotor Stress function.	
RT_RBS	Reheat bowl surface temperature. DEG F Used in Rotor Stress, Scheduling, Temperature Recommendations, Temperature Mismatch and Loading Rate functions.	
RUN_TIMER	ATS pass counter. CNT15 Used in Scheduling function.	
STEAM_MAX	Maximum recommended main steam temperature. DEG F Used in Temperature Recommendations function.	
STEAM_MIN	Minimum recommended main steam temperature. DEG F Used in Temperature Recommendations function.	
SURF_LA_1	Surface lookahead stress for area one. % Used in Rotor Stress function.	
SURF_LA_2	Surface lookahead stress for area two. % Used in Rotor Stress function.	
SURF_LA_3	Surface lookahead stress for area three. % Used in Rotor Stress function.	
SURF_STR_1	Surface stress for area one. % Used in Rotor Stress function.	
SURF_STR_2	Surface stress for area two. % Used in Rotor Stress function.	

SURF_STR_3	Surface stress for area three. % Used in Rotor Stress function.	
S2_ATS_1	Derived allowable surface stress less 25,000 psi for area one. . . . Used in Rotor Stress function.	
S2_ATS_2	Derived allowable surface stress less 25,000 psi for area two. . . . Used in Rotor Stress function.	
S2_ATS_3	Derived allowable surface stress less 25,000 psi for area three. . . . Used in Rotor Stress function.	
TIME_TO_STAB	Minutes until stress calculations are stable. MIN Used in Scheduling function.	
TM_ATS_1	Steam to metal temperature mismatch for area one. DEG F Used in Temperature Mismatch function.	
TM_ATS_2	Steam to metal temperature mismatch for area two. DEG F Used in Temperature Mismatch function.	
TM_ATS_3	Steam to metal temperature mismatch for area three. DEG F Used in Temperature Mismatch function.	
TN_RPM	Turbine speed in rpm. RPM Used in Acceleration and Rotor Stress functions.	
TNH1	Turbine speed as percent of rated. % Used in Acceleration function.	
TT_FSS	First stage shell temperature. DEG F Used in Rotor Stress, Scheduling, Temperature Recommendations, Temperature Mismatch and Loading Rate functions.	
TT_XOU	Crossover upper surface temperature. DEG F Used in Rotor Stress, Temperature Mismatch and Scheduling functions.	
TSAT	Saturation temperature in area one. DEG F Used in preparation for Rolloff function.	
VT_CVI	CV chest inner metal temperature. DEG F Used in Preparation for Rolloff, Average Temperature Change and Loading Monitor functions.	
VT_CVO	CV chest outer metal temperature. DEG F Used in Preparation for Rolloff, Average Temperature Change andLoading Monitor functions.	
VT_MCV_DIFF	Control valve temperature differential. DEG F Used in Average Temperature Change function.	
VT_MSVI_ATS	MSV inner metal temperature. DEG F Used in Average Temperature Change and Loading Monitor functions.	

VT_MSVO_ATS	MSV outer metal temperature.	
	DEG F	Used in Average Temperature Change and Loading Monitor functions.
VT_MSV_DIFF	Stop valve temperature differential.	
	DEG F	Used in Average Temperature Change function.
VT_MSV_ALLOW	Allowable stop valve temperature differential.	
	DEG F	Used in Average Temperature Change and Loading Monitor functions.
VT_CV_ALLOW	Allowable control valve temperature differential.	
	DEG F	Used in Average Temperature Change, Preparation for Rolloff and Loading Monitor functions.

Notes:

Problem Report

General Electric Company

To:
GE Industrial Control Systems
Product Service, Room 191
1501 Roanoke Blvd.
Salem, VA 24153-6492 USA

Fax: 001-540-387-8606

Name	Telephone Number
Company Name	Fax Number
Address	Job Site
	GE Requisition Number

Mark V Turbine Controls: Application Manual

Problem Description (Describe problem thoroughly. Include sequence leading up to problem..)

Detach, fold, and mail – or fax this side only.

Fold here and close with staple or tape

Place
stamp
here

GE Industrial Control Systems
Product Service, Room 191
1501 Roanoke Blvd.
Salem, VA 24153-6492 USA

Fold here first

Reader Comments

General Electric Company

To:
GE Industrial Control Systems
Technical Publications, Room 191
1501 Roanoke Blvd.
Salem, VA 24153–6492 USA

Fax: 001-540-387-8606

We welcome comments and suggestions to make this publication more useful.

Name	Telephone Number
Company Name	Fax Number
Address	Job Site
	GE Requisition Number

Publication No.	Issue/Revision Date	Today's Date

GENERAL COMMENTS

	Excellent	**Good**	**Fair**	**Poor**
Contents	○	○	○	○
Organization	○	○	○	○
Technical accuracy	○	○	○	○
Clarity	○	○	○	○
Completeness	○	○	○	○
Drawings, figures, and tables	○	○	○	○
Referencing and tables of contents	○	○	○	○
Readability	○	○	○	○

SPECIFIC COMMENTS (Corrections, information that could be expanded, and so on)

Page Number	**Comment**

Other suggestions for improving this publication:

As compared to publications from other manufacturers of similar products, how would you rate this document?

○ **Superior** ○ **Comparable** ○ **Inferior** ○ **Do not know**

Indicate the type of user/reader function that best describes you:

○ **System Integrator** ○ **Distributor** ○ **OEM** ○ **Installation**
○ **Programmer** ○ **Maintenance** ○ **Operator** ○ **Other (please specify)**

Detach, fold, and mail – or fax this side only.

.... Fold here and close with staple or tape ....

Place
stamp
here

GE Industrial Control Systems
Technical Publications, Room 191
1501 Roanoke Blvd.
Salem, VA 24153–6492 USA

.... Fold here first ....

GE Industrial Control Systems

General Electric Company
1501 Roanoke Blvd.
Salem, VA 24153-6492 USA