OPERATING MANUAL

PENTEK MODEL 6526

32−Channel Digital Receiver
VMEbus Board with RACEway

Pentek, Inc.
One Park Way
Upper Saddle River, NJ 07458
(201) 818−5900
http://www.pentek.com/

Manual Part No: 800.65260 Rev: A − September 20, 2000

Manual Revision History

Date	Manual Rev	Applicable Serial #'s	Comments
06/18/99	Preliminary	9908001 – Forward	Initial release. – Adapted from Preliminary Engineering Spec & Manual.
9/20/00	A		Updated for relase

Warranty

Pentek warrants that all products manufactured by Pentek conform to published Pentek specifications and are free from defects in materials and workmanship for a period of one year from the date of delivery when used under normal operating conditions and within the service conditions for which they were furnished.

The obligation of Pentek arising from a warranty claim shall be limited to repairing or at its option, replacing without charge, any product which in Pentek's sole opinion proves to be defective within the scope of the warranty.

Pentek must be notified in writing of the defect or nonconformity within the warranty period and the affected product returned to Pentek within thirty days after discovery of such defect or nonconformity.

Buyer shall prepay shipping charges, taxes, duties, and insurance for products returned to Pentek for warranty service. Pentek shall pay for the return of products to buyer except for products returned from another country.

Pentek shall have no responsibility for any defect or damage caused by improper installation, unauthorized modification, misuse, neglect, inadequate maintenance, or accident, or for any product which has been repaired or altered by anyone other than Pentek or its authorized representatives.

The warranty described above is buyer's sole and exclusive remedy and no other warranty, whether written or oral, is expressed or implied. Pentek specifically disclaims fitness for a particular purpose. Under no circumstances shall Pentek be liable for any direct, indirect, special, incidental, or consequential damages, expenses, losses or delays (including loss of profits) based on contract, tort, or any other legal theory.

Copyrights

Trademarks

Printed in the United States of America.

Table of Contents

Table of Contents (continued)

Page

Chapter 2: Installation and Connections (continued)

Chapter 3: Memory Maps and Register Descriptions

Table of Contents (continued)

Page

***Chapter 3: Memory Maps and Register Descriptions** (continued)*

Table of Contents

Page

Chapter 3: Memory Maps and Register Descriptions *(continued)*

Table of Contents

Page

Chapter 3: Memory Maps and Register Descriptions *(continued)*

Table of Contents

Page

Chapter 3: Memory Maps and Register Descriptions *(continued)*

Table of Contents

Page

Chapter 3: Memory Maps and Register Descriptions *(continued)*

Table of Contents

Chapter 1: Overview

1.1 Introduction

The Model 6526 is a 16−Channel, two input, Narrowband Digital Receiver for the VMEbus. It features a RACEway™ interface for output data, and several on−board control sections for supporting input switching, multi−board synchronization, and time code stamping of data. The Model 6526 is a single−slot, 6U VME board with front panel connections for input clock, input data, and synchronization signals. It includes a 32−bit VMEbus slave interface for control and status.

The RACEway interface allows packets of data from each receiver channel to be directed to different RACEway−equipped VME boards, including memory and DSP functions. RACEway is a high−speed, synchronous backplane bus capable of deliver−ing 32−bit word transfers between VME boards at a peak rate of 160 MB/sec. It offers significant advantages for VMEbus systems by providing a high−speed data channel completely independent of the VMEbus. The interface utilizes the 64 user−defined pins of the VME P2 connector, which are usually unconnected pins in most backplanes.

1.2 Digital Receiver

The Model 6526 uses four Graychip GC4014 Quad Narrowband Digital Drop Receivers (DDRs) to provide a total of 16 receiver channels. Two front panel parallel digital data inputs operate at either TTL or differential ECL logic levels, and support up to 16 bits of data and one clock at sampling rates up to 62.5 MHz for ECL and 50 MHz for TTL. The clock for the two inputs must be the same to meet the timing requirements of the GC4014's. The front panel inputs are directly compatible with the digital output of Pentek's Series 64xx A/D Converters.

The two parallel input signals are connected to two 16−bit inputs of each GC4014. Within the GC4014, input crossbar switches allow any DDR channel to independently select either of the two input signals. The GC4014's are controlled by commands from the VMEbus and by an on−board TMS320C31 DSP (Digital Signal Processor) for setting all operational parameters of the receiver channels.

1.3 Synchronization

The front panel of the Model 6526 has a Sync bus that can be bridged across all Model 6526's in a system. It is used to distribute four synchronization signals to all connected boards for synchronizing each of the of the same−channel DDR sections on each board. An on−board Sync Generator controlled by VMEbus commands can generate any number of sync signals (1 to 4) for output to the Sync bus.

1.4 Time Stamp Counter

The Time Stamp Counter is a 32−bit digital counter which acts as a master time code reference for the board. It is used for time stamping data packets from the receivers, and for determining when input switching commands are performed. It advances its code using a nominal 10 µsec clock derived by dividing (pre−scaling) the input clock from either parallel digital input.

1.5 Command Controller / DSP

The TMS320C31 Digital Signal Processor controls the receivers by using a list of input switching times for each of the 16 channels. This list is written through the VMEbus interface into a table, and then examined once every 10 µsec to determine if any DDR channel should change its input switch setting for the current 32−bit time stamp.

1.6 Channel Formatters

Sixteen identical Channel Formatter sections accept serial output data from the GC4014 DDRs, convert the data to 32−bit parallel words, and then form data packets (blocks) containing channel identification, the block number, the time stamp value, and a pro−grammable number of complex DRR data samples. The channel identification and block counter values are programmable over the VMEbus interface.

Optionally, the Channel Formatter also inserts a special Sync Code pattern into the data sequence, replacing two consecutive DDR samples following the receipt of the Sync bus signal for that channel.

Additionally, the Channel Formatter stores the RACEway routing code and the RACE−way address for each channel into VME−programmable registers. This allows each channel's data packet to be directed to any RACEway board and then steered to any resource on that board.

Data packets from the Channel Formatter are delivered to 4k x 32 synchronous Output FIFOs, one for each of the 16 channels. Once a data packet is delivered to a FIFO, a sig−nal is sent to the RACEway Controller. The RACEway Controller then retrieves the RACEway routing code and address from the Channel Formatter, and starts a RACE−way bus transfer using the packet stored in the FIFO.

1.7 RACEway Interface

The Model 6526's RACEway interface is implemented using the Cypress PitCREWjr chip set. The RACEway interface circuitry creates separate RACEway packets for each DDR on the board and sends them to the desired destination processors. Each DDR's output can be sent to one or more independent processors on another VME board in the same VME chassis via RACEway Interlink Modules. For more information about RACEway Interlinks, see Appendix B, and refer to the Pentek RACEway Handbook, Pentek part number 800.00003.

The Model 6526 is directly compatible with all of Pentek's RACEway–compatible DSP processors, including the Models 4290 and 4291 (equipped with a Model 6219 or 6220 RACEway Interface VIM module) and the Model 4285 (equipped with the Option 034 RACEway interface). The Model 6526 supports local RACEway packet steering to any of the processors on these boards, as determined by the RACEway address.

1.8 Digital Interfaces

Model 6526 Option 002 accepts two 16–bit differential ECL–level input signals, at a maximum data rate of 62.5 MSPS. This input option is compatible with Pentek A/D Converter Models 6402 Option 002, 6410 Option 002, 6425 Option 001, 6441 Option 002, 6465, 6470, and 6472. Model 6526 Option 019 accepts single–ended TTL–level signals, at a maximum rate of 50 MSPS. This input option is compatible with the standard versions of Pentek A/D Converter Models 6402, 6410, 6420, 6425, and 6441.

Another input data source for the Model 6526 is the Watkins–Johnson WJ–9107 Wideband Telecom Tuner. The WJ–9107 output is a 36–pin connector delivering differential ECL output levels. Two WJ–9107's may be located in an external VXI chassis, each supplying one digitized data signal to the two data inputs on one Model 6526. A special cable, available separately from Pentek as Model 2126, provides direct connection for the two input data signals from two WJ–9107's to the Model 6526.

1.9 VMEbus Interface

The Model 6526 is a single–slot, VME stand–alone board that meets VME C.1 specifications. It provides slave A32/D32 access, A24/D16 access, and A16/D16 access. It does not allow D8 access. It is a VMEbus interrupter but it is not a VMEbus interrupt handler. It passes the Bus Grant and IACK daisy chain signals.

The VMEbus Slave Interface allows read/write control of registers and access to a dual port SRAM used for passing parameters to the DDRs. These resources are memory mapped into a unique A24/A32 VME address space. A second address space, identical to the first, provides write–only access to all the same functions. The base address of each of these two spaces is programmable in A16 address space. This allows multiple boards to share the same write–only space for broadcast commands sent to several boards using a single VMEbus cycle. Commands for individual boards can always be sent through the unique read/write space.

1.10 Model 6526 Block Diagram

The block diagram for the Model 6526 is shown in Figure 1–1, below.

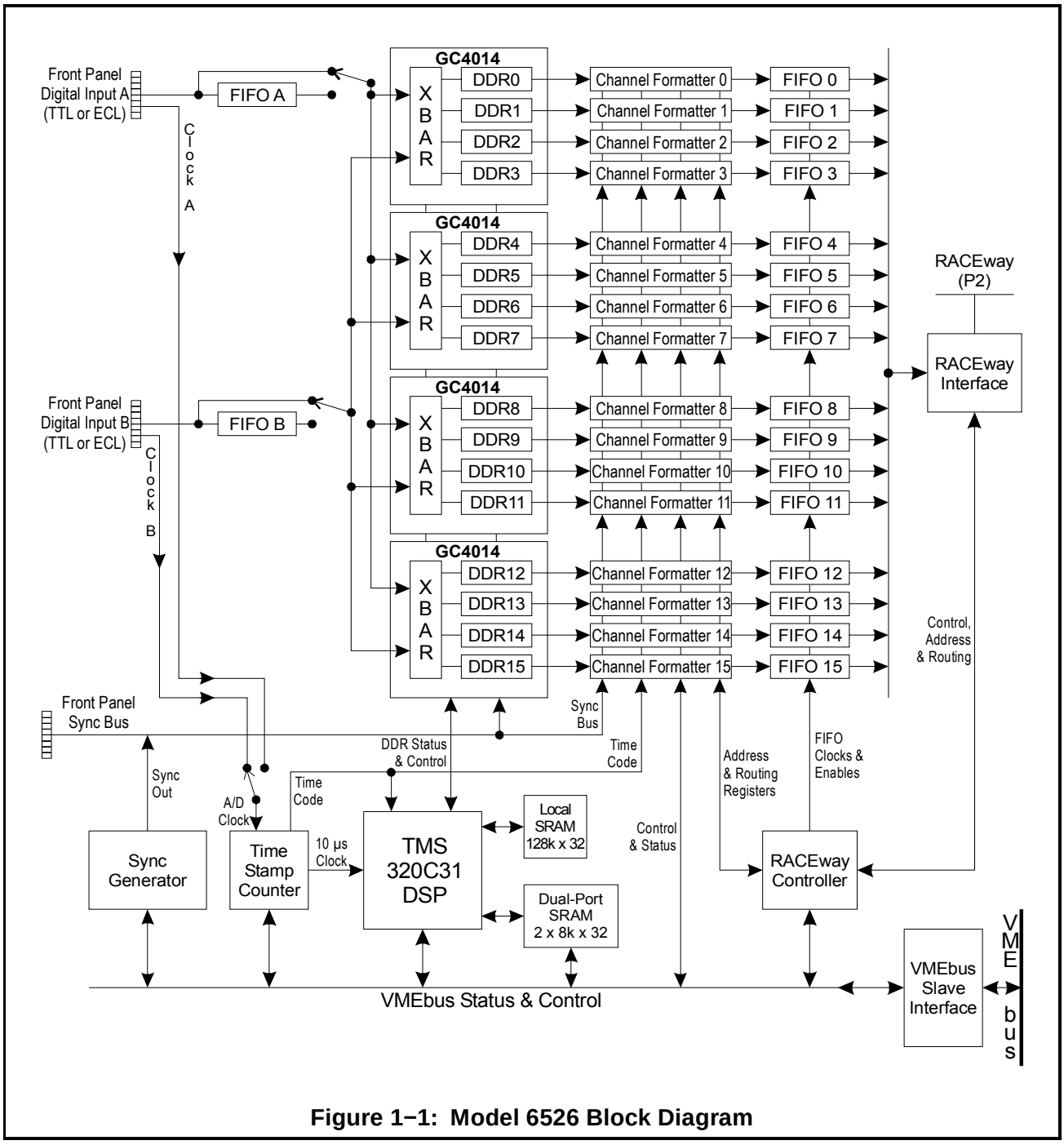


Figure 1–1: Model 6526 Block Diagram

1.11 Specifications

Receiver Chips: 4 (four) Graychip GC4014s, 16–bit input data
(See Appendix C for chip specifications)

Digital Inputs

Input Connectors: Front panel 80–pin 3M connector for 0.025" pitch cable; accepts two input signals of up to 16 data bits each, using 80–conductor cable (3M # 3756/80, Pentek # 378.68000) & 80–pin connector (3M # 82080–6006, Pentek # 353.08005)

Option 002: Differential ECL inputs compatible with Pentek Models 6402–002, 6410–002, 6441–002, and 6465 A/D Converters (using Pentek's Model 2117 cable assembly), and Pentek Models 6425–001, 6470, and 6472 A/D Converters (using Pentek's Model 2119 cable assembly); these A/D Converters can provide data rates up to 70 MHz

Option 019: Single–ended TTL inputs compatible with Pentek Models 4261–019, 6402, 6410, 6420, 6421, and 6441 (using Pentek's Model 2117 cable assembly), and Pentek Model 6425 (using Pentek's Model 2119 cable assembly); data rates up to 41 MHz can be provided by these units

Input Signals (2):

Data:	16 (sixteen) bit lines/input signal (16 differential pairs (32 lines) for Option 002)
Clock:	1 (one) bit line/input signal
Clock Rate:	62.5 MHz, maximum (Option 002) 50 MHz, maximum (Option 019)
External Sync:	2 (two) bit lines/input (filter & accumulator syncs)
Data Setup Time:	2 nsec before rising edge of clock
Data Hold Time:	2 nsec after rising edge of clock

Input Multiplexers: Each receiver can independently select Input A or Input B

Input FIFO Memory: 4k x 32 Synchronous FIFO for each receiver

RACEway Interface: One PitCREWjr chip set; transfers data at 160 MByte/sec peak and 140 MByte/sec sustained throughput
(See Appendix D for chip specifications)

1.11 Specifications (continued)

VMEbus Interface: Slave; A32/D32; I1–7;
Power up with no A24/A32 address mapping;
Bus Grant daisy chain bussed through;
Interrupt daisy chain supported

- **A16 Memory:** A16 base address set with rotary switch;
  256–byte address space maximum;
  A24/A32 base address configuration registers
- **A24/32 Memory:** Mapping disabled on power up, configured with A16 memory registers;
  1 MB address space in A24;
  4 MB address space in A32
- **VMEbus Data Rate:** 200 nsec DS to DTACK delay max (with no contention for global bus by 'C31);
  280 nsec bus cycle time max (with 80 nsec master response overhead)

Processor: One Texas Instruments TMS320C31, 60 MHz;
On–board XDS–500 emulator connector (12 pins)

- **'C31 Memory:** 128k x 32 Global SRAM, one wait state, shared with VMEbus;
  256k x 8 Flash User EEPROM

Dual–Ported SRAM

- **Size:** 64 kbytes, 1 bank, 16k x 32
- **'C31 Access:** Yes (controlled by on–board register)
- **VMEbus access:** Yes, A24/A32 (controlled by on–board register)

Power: 6.5 A @ +5 VDC maximum

Dimensions: Standard 6U VMEbus board

- **Depth:** 160.0 mm (6.3")
- **Height:** 233.5 mm (9.2")
- **Panel Width:** 20.3 mm (0.8")

Chapter 2: Installation and Connections

2.1 Inspection

After unpacking, inspect the unit carefully for possible damage to connectors or components. If any damage is discovered, contact Pentek at (201) 818−5900. Please save the original shipping container and packing material in case re−shipment is required.

The Model 6526 contains two printed circuit (PC) assemblies. The board with the VME−bus P1 and P2 connectors (the VME board, Pentek part # 004.65260) contains the VME interface, the 'C31 DSP, its associated memory resources, and the FIFOs. The other board (the Mezzanine board, Pentek part # 004.65261), houses the input connectors and the Digital Receiver circuitry. Drawings of the component and solder sides of both PC assemblies are shown in Figures 2−2 through 2−5, on the next four pages.

2.2 VMEbus A16 Base Address Switches

Three separate address spaces are defined for the VMEbus environment. These three regions are known as A16, A24, and A32. Each device in your VMEbus card cage must have a unique address in any address space in which it must be accessible. On the Model 6526, the A16_base address (the lowest address in the A16 region occupied by this device) is set by two rotary switches. These switches, SW1 and SW2, are located on the VME board and are accessible through a notch at the rear of the Mezzanine board. Set these switches for the desired A16_base address before installing the Model 6526 in the VMEbus card cage. Figure 2−1, below, shows what these switches look like, and Figure 2−2, on the next page, shows the location of these switches on the VME board.

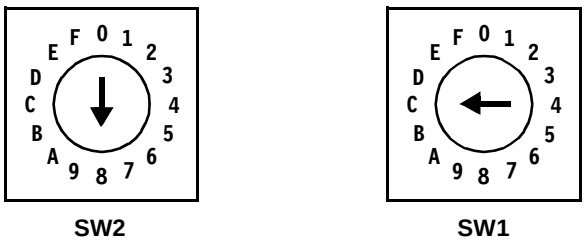


Figure 2−1: Model 6526 − Rotary Switches

The settings of these 16−position rotary switches each defines four bits of the Model 6526's A16_base address. Each of the 16 positions on the switches denotes a hex digit (0 to F), whose four−bit binary value is used as part of the device's base address. SW2 defines the four most significant bits (MSBs) of the address (A15 – A12), and SW1 defines the next four bits (A11 – A8). Thus, with 16 settings available on each of two switches, 256 different base addresses may be selected. Figure 2−1, above, illustrates the switches set for an A16_base address of 0x8C00.

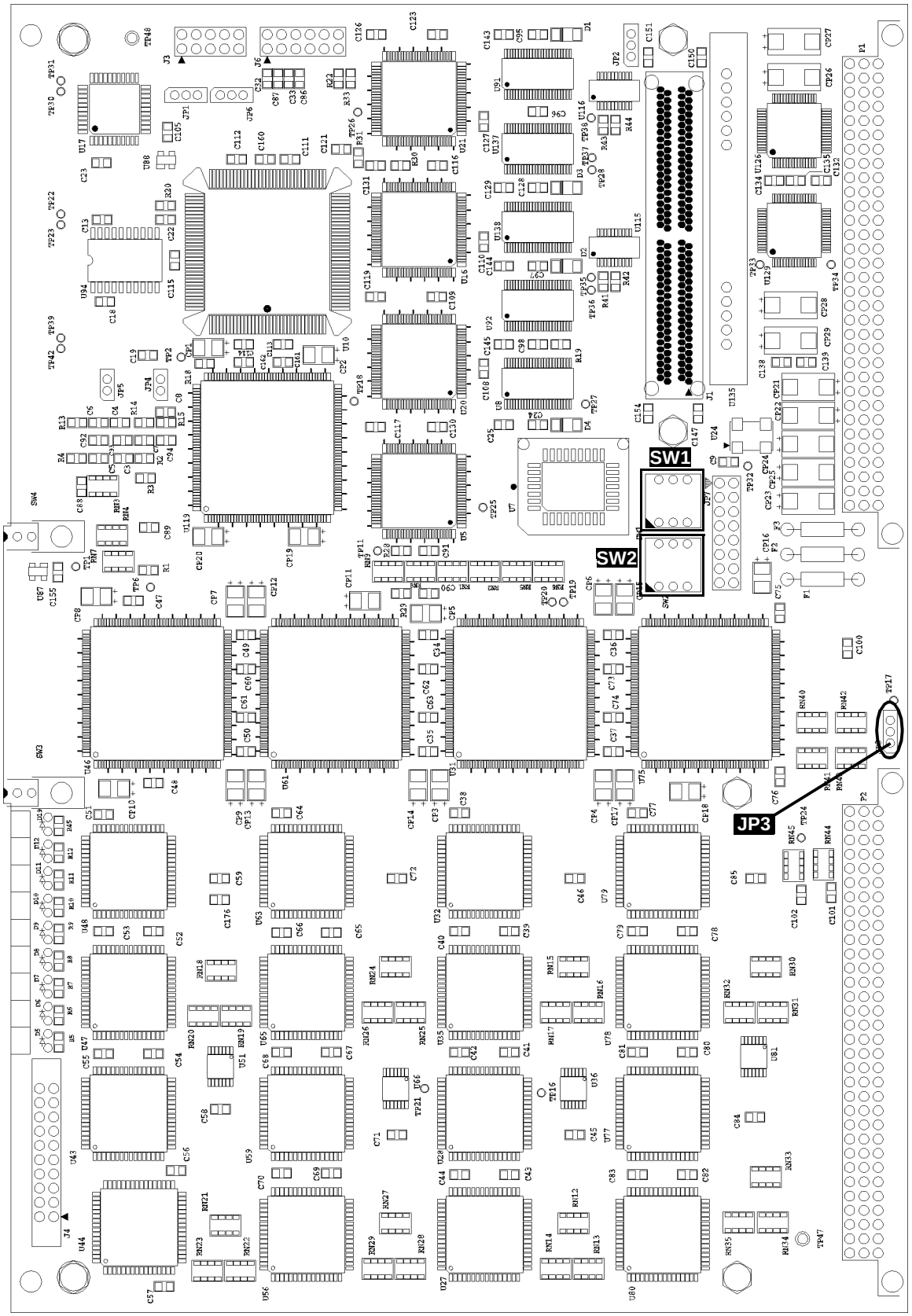


Figure 2−2: Model 6526 VME Board − Component Side Assembly Drawing

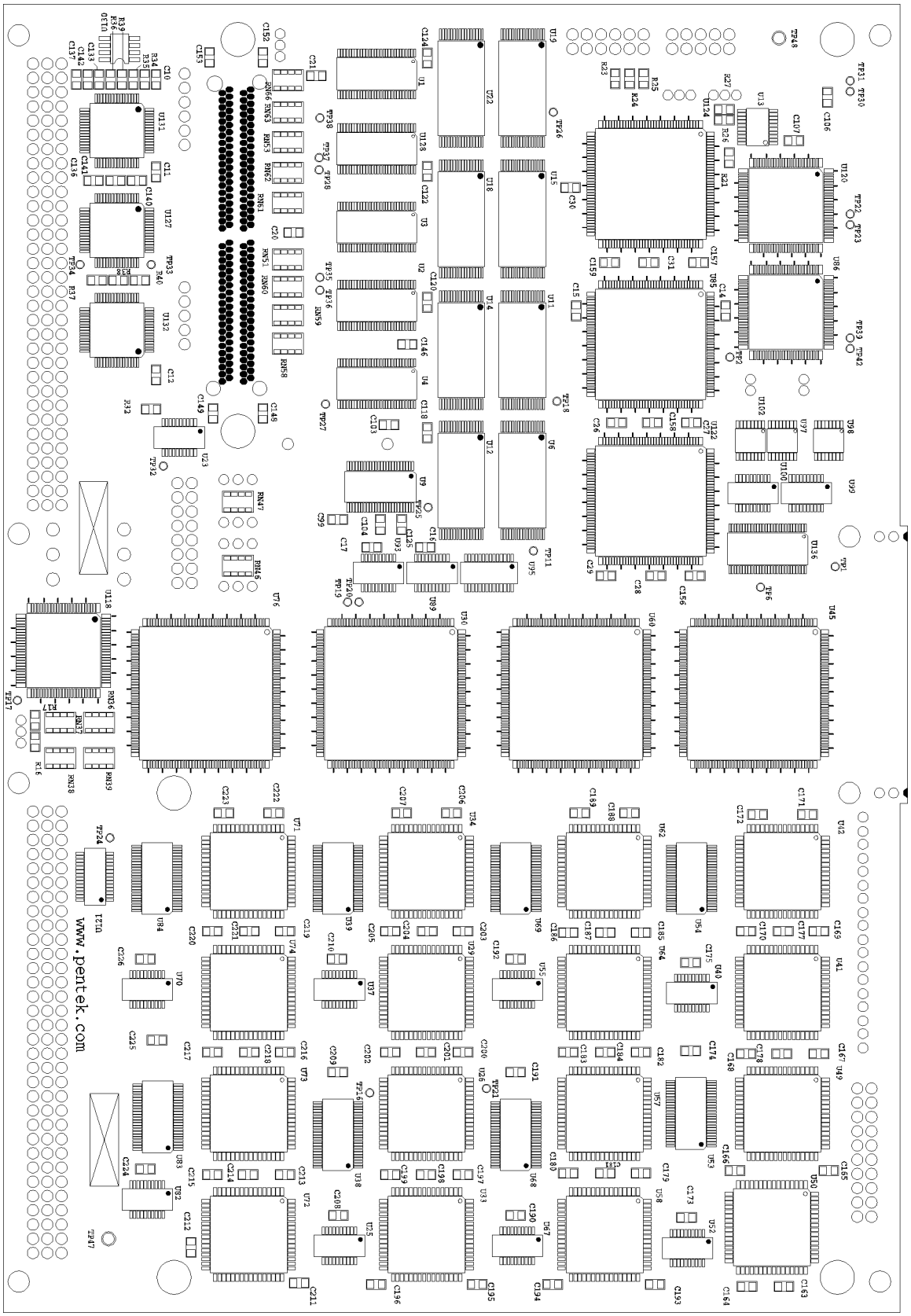


Figure 2−3: Model 6526 VME Board − Solder Side Assembly Drawing

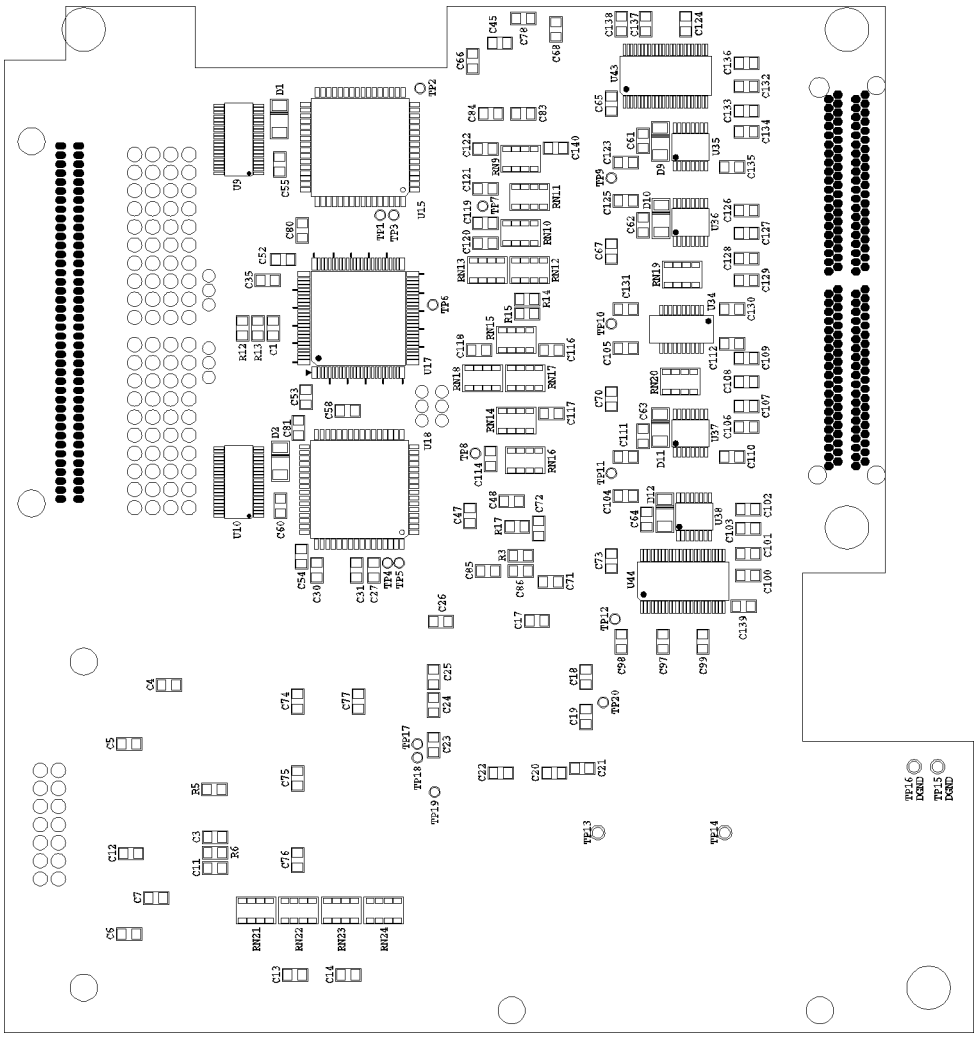


Figure 2−4: Model 6526 Mezzanine Board − Solder Side Assembly Drawing

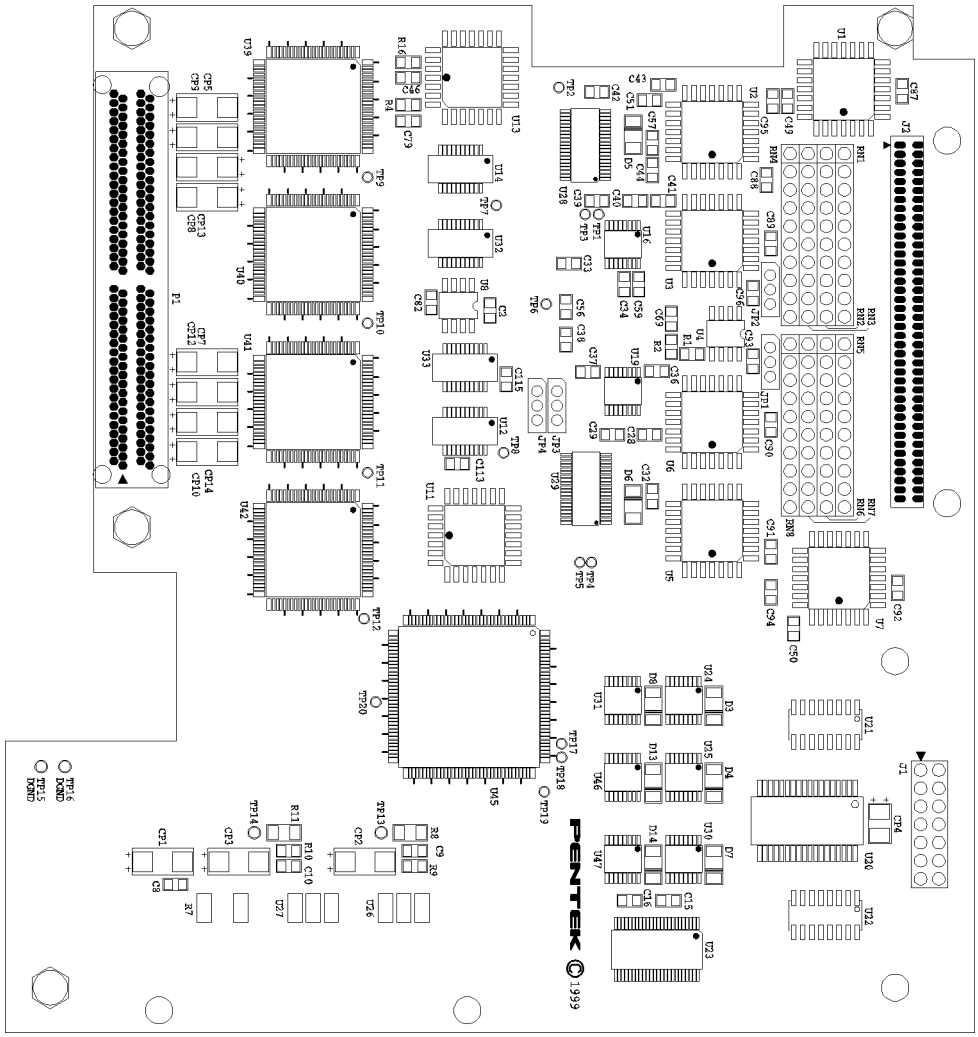


Figure 2−5: Model 6526 Mezzanine Board − Component Side Assembly Drawing

2.3 Jumper Block Settings on the Model 6526

Most of the jumper blocks on the Model 6526 are for factory testing purposes. These are all configured for proper field operation at the conclusion of testing, and should not be changed by the user.

There is, however, one jumper the user might want to set before installing the unit in the card cage. Jumper block JP3 determines whether or not the Model 6526 is capable of generating the RACEway RESET signal. By default, the jumper on this block is placed between pins 1 and 2, which disables the Model 6526 as a RACEway RESET generator (removing this jumper from the block also serves this purpose). To allow the Model 6526 to generate the RACEway RESET signal, place the jumper between pins 2 and 3 of JP3. This is summarized in Table 2–1, below.

Table 2–1: RACEway RESET Enable Jumper JP3

Jumper Position	RACEway RESET Generation
Pin 1 – Pin 2*	Disabled*
Pin 2 – Pin 3	Enabled
* Factory Default Setting – also Disabled if no jumper is installed.	

The JP3 jumper block is located just above the P2 VME connector on the Model 6526 main VME board (see Figure 2–2, on page 18).

2.4 Model 6526 Front Panel

The front panel of Pentek's Model 6526 is shown in Figure 2–6, at the right. Available on the front panel are an 80–pin Digital Data In connector, a Reset button, a 14–pin Sync bus connector, several indicator LEDs, and a 20–pin Serial connector for access to the TMS320C31's Serial Port. Each of these is described in a subsection below.

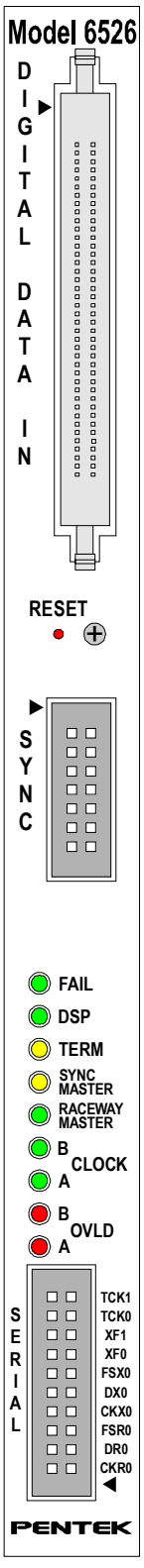


Figure 2–6: Model 6526 Front Panel

2.4.1 Digital Data Input Connector

The input connector, labeled DIGITAL DATA IN, is an 80–pin flat ribbon cable header manufactured by 3M. The part number for the header is 3M #81080–620203 (Pentek part # 354.08001).

The mating flat cable socket connector is 3M #82080–6006, and the strain relief is 3M #3858–080. Pentek's part number for the connector, with the strain relief, is 353.08005. One of each of these items is supplied with the Model 6526. The recommended 0.025" pitch flat ribbon cable is 3M #3756/80.

A 68–conductor cable of the same type may also be used for applications requiring only 12 data bits. This cable is type 3M #3756/68. The 68–conductor cable should be crimped into the center of the 80–pin cable connector such that pins 1 through 6 and pins 75 through 80 are left unconnected.

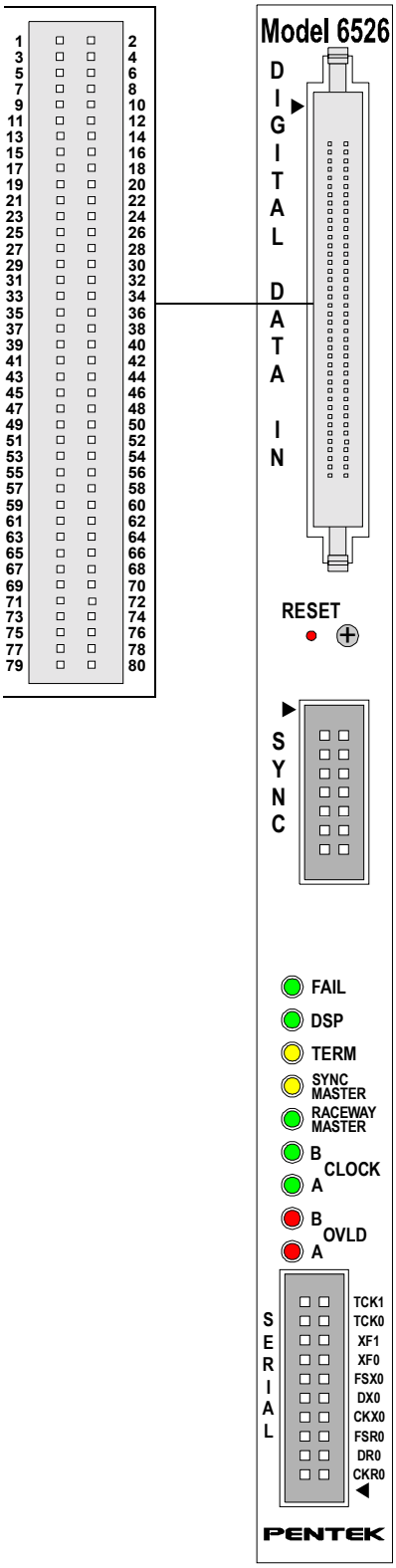
The 80–pin connector is composed of two similar 40–pin groups, one for each input signal. The ribbon cable can be split in half, forming two 40–pin cables. The pin assignment for the connector is shown on the next page, in Figure 2–7, for the two input signal options.

- Model 6526 Option 002 accommodates two 16–bit differential ECL–level input signals, available from such devices as Pentek A/D Converter Models 6402 Option 002, 6410 Option 002, 6425 Option 001, 6441 Option 002, 6465, 6470, and 6472.

- Model 6526 Option 019 accepts single–ended TTL–level signals, and is compatible with the standard versions of Pentek A/D Converter Models 6402, 6410, 6420, 6425, and 6441.

Input Connector Pin Assignments Model 6526-002 (Differential ECL)			
Pin	**Signal**	**Pin**	**Signal**
1	Ch B - D1 in	**2**	Ch B - $\overline{D1}$ in
3	Ch B - D2 in	**4**	Ch B - $\overline{D2}$ in
5	Ch B - D3 in	**6**	Ch B - $\overline{D3}$ in
7	Ch B - D4 in	**8**	Ch B - $\overline{D4}$ in
9	Ch B - D5 in	**10**	Ch B - $\overline{D5}$ in
11	Ch B - D6 in	**12**	Ch B - $\overline{D6}$ in
13	Ch B - D7 in	**14**	Ch B - $\overline{D7}$ in
15	Ch B - D8 in	**16**	Ch B - $\overline{D8}$ in
17	Ch B - D9 in	**18**	Ch B - $\overline{D9}$ in
19	Ch B - D10 in	**20**	Ch B - $\overline{D10}$ in
21	Ch B - D11 in	**22**	Ch B - $\overline{D11}$ in
23	Ch B - D12 in	**24**	Ch B - $\overline{D12}$ in
25	Ch B - D13 in	**26**	Ch B - $\overline{D13}$ in
27	Ch B - D14 in	**28**	Ch B - $\overline{D14}$ in
29	Ch B - D15 in	**30**	Ch B - $\overline{D15}$ in
31	Ch B - SS in	**32**	Ch B - $\overline{SS}$
33	Reserved	**34**	Reserved
35	Ch B - CK	**36**	Ch B - $\overline{CK}$
37	Ch B - D0 in	**38**	Ch B - $\overline{D0}$ in
39	GND	**40**	GND
41	Ch A - D4 in	**42**	Ch A - $\overline{D4}$ in
43	Ch A - D5 in	**44**	Ch A - $\overline{D5}$ in
45	Ch A - D6 in	**46**	Ch A - $\overline{D6}$ in
47	Ch A - D7 in	**48**	Ch A - $\overline{D7}$ in
49	Ch A - D8 in	**50**	Ch A - $\overline{D8}$ in
51	Ch A - D9 in	**52**	Ch A - $\overline{D9}$ in
53	Ch A - D10 in	**54**	Ch A - $\overline{D10}$ in
55	Ch A - D11 in	**56**	Ch A - $\overline{D11}$ in
57	Ch A - D12 in	**58**	Ch A - $\overline{D12}$ in
59	Ch A - D13 in	**60**	Ch A - $\overline{D13}$ in
61	Ch A - D14 in	**62**	Ch A - $\overline{D14}$ in
63	Ch A - D15 in	**64**	Ch A - $\overline{D15}$ in
65	Ch A - SS in	**66**	Ch A - $\overline{SS}$ in
67	Reserved	**68**	Reserved
69	Ch A - CK	**70**	Ch A - $\overline{CK}$
71	Ch A - D0	**72**	Ch A - $\overline{D0}$
73	GND	**74**	GND
75	Ch A - D1	**76**	Ch A - $\overline{D1}$
77	Ch A - D2	**78**	Ch A - $\overline{D2}$
79	Ch A - D3	**80**	Ch A - $\overline{D3}$

Input Connector Pin Assignments Model 6526-019 (Single-ended TTL)			
Pin	**Signal**	**Pin**	**Signal**
1	Ch B - D1 in	**2**	GND
3	Ch B - D2 in	**4**	GND
5	Ch B - D3 in	**6**	GND
7	Ch B - D4 in	**8**	GND
9	Ch B - D5 in	**10**	GND
11	Ch B - D6 in	**12**	GND
13	Ch B - D7 in	**14**	GND
15	Ch B - D8 in	**16**	GND
17	Ch B - D9 in	**18**	GND
19	Ch B - D10 in	**20**	GND
21	Ch B - D11 in	**22**	GND
23	Ch B - D12 in	**24**	GND
25	Ch B - D13 in	**26**	GND
27	Ch B - D14 in	**28**	GND
29	Ch B - D15 in	**30**	GND
31	Ch B - SS in	**32**	GND
33	Reserved	**34**	GND
35	Ch B - CK	**36**	GND
37	Ch B - D0 in	**38**	GND
39	GND	**40**	GND
41	Ch A - D4 in	**42**	GND
43	Ch A - D5 in	**44**	GND
45	Ch A - D6 in	**46**	GND
47	Ch A - D7 in	**48**	GND
49	Ch A - D8 in	**50**	GND
51	Ch A - D9 in	**52**	GND
53	Ch A - D10 in	**54**	GND
55	Ch A - D11 in	**56**	GND
57	Ch A - D12 in	**58**	GND
59	Ch A - D13 in	**60**	GND
61	Ch A - D14 in	**62**	GND
63	Ch A - D15 in	**64**	GND
65	Ch A - SS in	**66**	GND
67	Reserved	**68**	GND
69	Ch A - CK	**70**	GND
71	Ch A - D0	**72**	GND
73	GND	**74**	GND
75	Ch A - D1	**76**	GND
77	Ch A - D2	**78**	GND
79	Ch A - D3	**80**	GND



Note: For each input signal, Bit D0 is the LSB.

The mating cable socket connector is 3M #82080−6006,
and the strain relief is 3M #3858−080.
The recommended 0.025" pitch flat ribbon cable is 3M # 3756/080.

Figure 2−7: Front Panel Digital Signal Input Connectors − Pin Assignments

2.4 Model 6526 Front Panel (continued)

2.4.1 Digital Data Input Connector (continued)

2.4.1.1 Connecting a Watkins–Johnson Model WJ–9107 to Model 6526

Another input data source for the Model 6526 is the Watkins–Johnson WJ–9107 Wideband Telecom Tuner. Two WJ–9107's may be located in an external VXI chassis, each supplying one digitized data signal to the two data inputs on one Model 6526. The WJ–9107 output connector is a 36–pin dual header delivering differential ECL output levels.

The Model 2126 cable, available separately from Pentek, provides a direct connection for two input data signals from two WJ–9107's to the Model 6526. This cable is a 68–conductor ribbon cable with 0.025" conductor spacing (3M part # 3756/068). One end of the cable connects to the 80–pin connector on the front panel of the Model 6526. The 68–conductor cable is installed in the center 68 pins of the 80–pin front panel connector of the Model 6526. The cable is divided into two 34–conductor ribbon cables, each of which is connected to a 36–pin dual–inline header connector for the WJ–9107 digital output. The Model 2126 cable is 48 inches in length and can connect a Model 6526 to two WJ–9107 tuners.

Table 2–2: Input Connections – Watkins–Johnson Model 9107 to Pentek Model 6526 with Pentek Model 2126 Cable

Pentek 6526 Signal	6526 Pin #	9107 pin #	W-J 9107 Signal	Pentek 6526 Signal	6526 Pin	9107 Pin #	W-J 9107 Signal	Pentek 6526 Signal	6526 Pin #	9107 pin #	W-J 9107 Signal	Pentek 6526 Signal	6526 Pin	9107 Pin #	W-J 9107 Signal
Ch B - D1 in	1	—	N/C	Ch B - $\overline{D1}$ in	2	—	N/C	Ch A - D4 in	41	1	D0_H	Ch A - $\overline{D4}$ in	42	2	D0_L
Ch B - D2 in	3	—	N/C	Ch B - $\overline{D2}$ in	4	—	N/C	Ch A - D5 in	43	3	D1_H	Ch A - $\overline{D5}$ in	44	4	D1_L
Ch B - D3 in	5	—	N/C	Ch B - $\overline{D3}$ in	6	—	N/C	Ch A - D6 in	45	5	D2_H	Ch A - $\overline{D6}$ in	46	6	D2_L
Ch B - D4 in	7	1	D0_H	Ch B - $\overline{D4}$ in	8	2	D0_L	Ch A - D7 in	47	7	D3_H	Ch A - $\overline{D7}$ in	48	8	D3_L
Ch B - D5 in	9	3	D1_H	Ch B - $\overline{D5}$ in	10	4	D1_L	Ch A - D8 in	49	9	D4_H	Ch A - $\overline{D8}$ in	50	10	D4_L
Ch B - D6 in	11	5	D2_H	Ch B - $\overline{D6}$ in	12	6	D2_L	Ch A - D9 in	51	11	D5_H	Ch A - $\overline{D9}$ in	52	12	D5_L
Ch B - D7 in	13	7	D3_H	Ch B - $\overline{D7}$ in	14	8	D3_L	Ch A - D10 in	53	13	D6_H	Ch A - $\overline{D10}$ in	54	14	D6_L
Ch B - D8 in	15	9	D4_H	Ch B - $\overline{D8}$ in	16	10	D4_L	Ch A - D11 in	55	15	D7_H	Ch A - $\overline{D11}$ in	56	16	D7_L
Ch B - D9 in	17	11	D5_H	Ch B - $\overline{D9}$ in	18	12	D5_L	Ch A - D12 in	57	17	D8_H	Ch A - $\overline{D12}$ in	58	18	D8_L
Ch B - D10 in	19	13	D6_H	Ch B - $\overline{D10}$ in	20	14	D6_L	Ch A - D13 in	59	19	D9_H	Ch A - $\overline{D13}$ in	60	20	D9_L
Ch B - D11 in	21	15	D7_H	Ch B - $\overline{D11}$ in	22	16	D7_L	Ch A - D14 in	61	21	D10_H	Ch A - $\overline{D14}$ in	62	22	D10_L
Ch B - D12 in	23	17	D8_H	Ch B - $\overline{D12}$ in	24	18	D8_L	Ch A - D15 in	63	23	D11_H	Ch A - $\overline{D15}$ in	64	24	D11_L
Ch B - D13 in	25	19	D9_H	Ch B - $\overline{D13}$ in	26	20	D9_L	Ch A - SS in	65	25	N/C	Ch A - $\overline{SS}$ in	66	26	N/C
Ch B - D14 in	27	21	D10_H	Ch B - $\overline{D14}$ in	28	22	D10_L	Reserved	67	27	N/C	Reserved	68	28	N/C
Ch B - D15 in	29	23	D11_H	Ch B - $\overline{D15}$ in	30	24	D11_L	Ch A - CK	69	29	CK_H	Ch A - $\overline{CK}$	70	30	CK_L
Ch B - SS in	31	25	N/C	Ch B - $\overline{SS}$ in	32	26	N/C	Ch A - D0	71	—	N/C*	Ch A - $\overline{D0}$	72	—	N/C*
Reserved	33	27	N/C	Reserved	34	28	N/C	GND	73	33	GND	GND	74	34	GND
Ch B - CK	35	29	CK_H	Ch B - $\overline{CK}$	36	30	CK_L	Ch A - D1	75	—	N/C	Ch A - $\overline{D1}$	76	—	N/C
Ch B - D0 in	37	—	N/C*	Ch B - $\overline{D0}$ in	38	—	N/C*	Ch A - D2	77	—	N/C	Ch A - $\overline{D2}$	78	—	N/C
GND	39	33	GND	GND	40	34	GND	Ch A - D3	79	—	N/C	Ch A - $\overline{D3}$	80	—	N/C

* Pins 31 and 32 of the WJ-9107 connector are grounded. The Model 2126 cable does not connect these lines to the Model 6526.
Note that pins 35 and 36 of each of the two WJ-9107 36-pin connectors are not connected to the two 34-conductor cables; and pins 1 – 6, 37, 38, 71, 72, and 75 – 80 of the Model 6526 80-pin connector are not connected (N/C) to the 68-conductor cable. Also, because the WJ-9107 provides only 12 bits of data per output, the WJ-9107 outputs only drive the 12 most significant bits of the 16-bit GC4014 data input capability.

2.4 Model 6526 Front Panel (continued)

2.4.1 Digital Data Input Connector (continued)

2.4.1.2 Input Equivalent Circuits – Option 002

On the Model 6526 Option 002, the 68 differential ECL data input lines (D0 in – D15 in, $\overline{D0}$ in – $\overline{D15}$, SS in, and $\overline{SS}$ in on inputs A and B) are terminated with 150 Ω resistors to –2 V_{DC}. These signals are then converted to single–ended TTL by an MC10H605 registered ECL/TTL Translator. The four differential clock input lines (CK and $\overline{CK}$, on inputs A and B) are terminated in 2 kΩ resistors to –5.2 V_{DC}, and drive the inputs of MC10EL16 differential receivers. The differential ECL outputs of these two receivers are used to clock the 10H605 translators.

Note: The terminating resistors are socketed and removable if more than one Model 6526 is daisy–chained from one input source. Terminating resistors are required only at the end of the daisy chain.

Figure 2–8, below, illustrates the input equivalent circuits schematically.

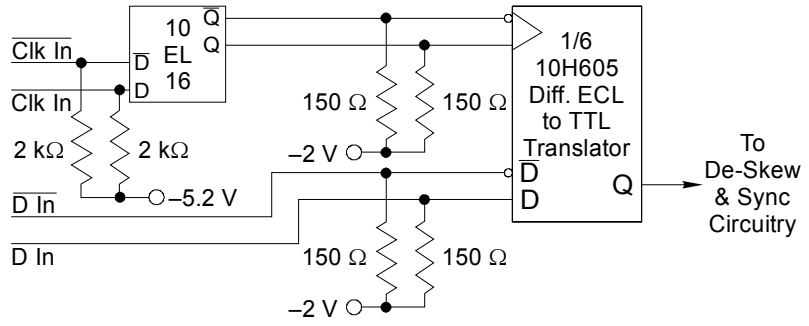


Figure 2–8: Clock & Data Input Equivalent Circuits

2.4 Model 6526 Front Panel (continued)

2.4.2 Sync Bus

The Model 6526 uses a Sync bus to support several methods for synchronizing multiple boards in a VMEbus card cage. This allows changes to occur simultaneously in each of the following functions:

- Issuing DDR commands to specific channels, including tuning and input switching.
- Updating the Time Stamp Counter and reprogramming Time Stamp Pre–Scaler.
- Substituting a "Sync word" pattern for DDR output data in specific channels.
- Resetting the RACEway packet Block Counter for specific channels.
- Triggering a consecutive group of RACEway packets to be sent from the channel formatter.
- Synchronizing and buffering the Input Data FIFOs.

The synchronization signals are generated by a single designated Model 6526 in the group of boards, and the synchronization signals are broadcast across all boards with a front panel ribbon cable bus.

2.4.2.1 Sync Bus Connector

The Sync bus interface is a 14–pin male, dual–in–line shrouded connector on the Model 6526's front panel. The connector is 3M part # 2514–5002–UB (Pentek part # 354.01403). It consists of two rows of seven pins on a grid of 0.100 inch. The connector mates with any standard DIL female socket mass–termination connector using 0.050" flat ribbon cable, such as 3M part # 3385–6000, no strain relief (Pentek part # 353.01404), or 3M part # 3385–6014, with plastic strain relief (Pentek does not stock this part). The recommended flat ribbon cable is 3M part # 3365/14 (Pentek part # 378.61400) or an equivalent. Table 2–3, at the top of the next page, gives the pinout of the front panel Sync bus connector.

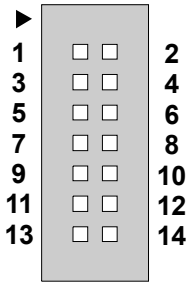
2.4 Model 6526 Front Panel (continued)

2.4.2 Sync Bus (continued)

2.4.2.1 Sync Bus Connector (continued)

Table 2−3: Front Panel Sync Bus Connector Pinout

Pin #	Signal	Pin #	Signal
1	Sync0	2	$\overline{\text{Sync0}}$
3	GND	4	Sync1
5	$\overline{\text{Sync1}}$	6	GND
7	Sync2	8	$\overline{\text{Sync2}}$
9	GND	10	Sync3
11	$\overline{\text{Sync3}}$	12	GND
13	N/C	14	N/C



Mating connector: 3M # 3385/60XX or eq.
Recommended flat ribbon cable: 3M # 3365/14 or eq.

The four sync signals received by the Model 6526 are distributed with one sync signal going to the SI (sync input) pin on each of the GC4014 DDRs. Programmable registers within the GC4014 allow the SI signal to be used selectively for several internal GC4014 functions for each of the four inputs. The Sync bus signals are assigned as shown in Table 2−4, at the top of the next page.

2.4 Model 6526 Front Panel (continued)

2.4.2 Sync Bus (continued)

2.4.2.2 Sync Bus / Receiver Assignments

Table 2−4: Sync Bus Signals and their Associated Receivers

Sync	Receivers
Sync0	0 – 3
Sync1	4 – 7
Sync2	8 – 11
Sync3	12 – 15

In addition, each of the sixteen channel formatters uses the same Sync bus signal distribution scheme as the GC4014s. Specifically, each one of the four Sync bus signals connects to four channel for−matters per the table above. The Time Stamp Counter and Pre−Scaler reset functions are global resources and are synchronized only by SYNC0.

2.4.2.3 Sync Bus Signal Characteristics

The sync signals are distributed as Low Voltage Differential Sig−nals (LVDS) to other Model 6526 boards. Each board contains a tri−state LVDS Sync bus driver which is enabled when that board is programmed to be the Sync bus master. Only one master is allowed for each Sync bus. The Sync bus master must be located at one end of the Sync bus front panel cable.

Each Model 6526 board also contains a Sync bus LVDS active ter−mination device (active SCSI−type) which can be enabled under program control. The board at the opposite end of the Sync bus cable from the Sync bus master must have this termination enabled to properly match the cable impedance and minimize ringing on the bus. System software must ensure that only one board is enabled for termination.

At power up, the Sync bus master and terminate functions default to inactive.

2.4 Model 6526 Front Panel (continued)

2.4.2 Sync Bus (continued)

2.4.2.4 Sync Bus Equivalent Circuit

Each of the sync lines is connected to each of the following items:

1) A Low Voltage Differential SCSI–type active terminator (Unitrode UCC5630MWP). This device is normally disabled at power–up, but must be enabled ONLY in the unit that is at the opposite end of the Sync bus from the sync master. The termination is enabled by setting the Sync Term bit in the Sync Bus Control Register (see Section 3.4.2) to the logic '1' state.

2) The output of a 3V Differential Line Driver (National Semiconductor DS90LV031A). This device is also normally disabled at power–up, and must be enabled ONLY in the Model 6526 that is the sync master. The driver is enabled by setting the Sync Master bit in the Sync Bus Control Register (see Section 3.4.2) to the logic '1' state. Devices should be arranged in your card cage such that the sync master drives one end of the sync cable, and the unit with the terminator enabled is at the opposite end of the sync cable.

3) The input of a 3V Differential Line Receiver (National Semiconductor DS90LV032A). This device is always enabled in all Model 6526's.

The Sync bus equivalent circuit is shown schematically in Figure 2–9, below.

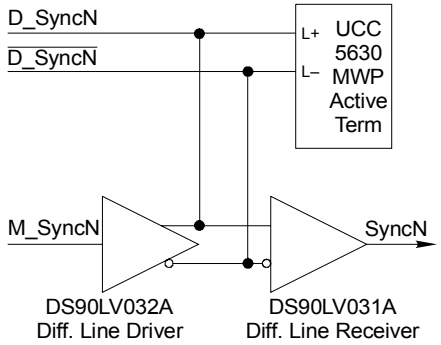


Figure 2–9: Sync Bus Equivalent Circuit

2.4 Model 6526 Front Panel (continued)

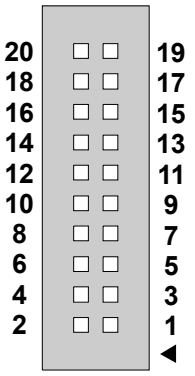
2.4.3 TMS320C31 Serial Port

The 'C31 DSP processor features a serial port, whose pins are brought directly to a front panel 20−pin connector. The signal description is found in the Texas Instruments TMS320C3x User's Guide, TI publication number 2558539−9761. Note that these pins are **UNBUFFERED** connections to the 'C31. Please exercise extreme caution when making connections to this header.

The Serial Port is available on the front panel via a 20−pin male, dual−in−line (DIL) shrouded connector. The connector is type 3M 2520−5002−UB. It consists of two rows of 10 pins on a grid of 0.100 inch. The connector mates with any standard DIL female socket mass−termination connector using 0.050" flat ribbon cable, such as 3M part # 3421−6020, with strain relief (Pentek part # 353−02006). The recommended flat ribbon cable is 3M part # 3365/20 (Pentek part # 378.62000) or an equivalent. Table 2−5, below, gives the pinout of this connector.

Table 2−5: Front Panel Serial Port Connector

Pin #	Signal	Pin #	Signal
20	GND	19	TCLK1
18	GND	17	TCLK0
16	GND	15	XF1
14	GND	13	XF0
12	GND	11	FSX0
10	GND	9	DX0
8	GND	7	CLKX0
6	GND	5	FSR0
4	GND	3	DR0
2	GND	1	CLKR0



Mating connector: 3M part # 3421−6020 or eq.
Recommended flat ribbon cable: 3M part # 3365/20 or eq.

2.4.4 Reset Button

The Model 6526's Reset button is located between the Digital Data Input connector and the Sync bus connector on the front panel. Pressing this switch returns the Model 6526 to its power−up reset state.

2.4 Model 6526 Front Panel (continued)

2.4.5 LED Indicators

There are nine LED indicators on the front panel of the Model 6526, arranged in a single, vertical row above the ‘C31 Serial Port connector. The subsections below describes the conditions that these LEDs indicate.

2.4.5.1 Overload Indicator LEDs (OVLD A & B)

These red LEDs for each input channel indicate an overload on the respective channel when illuminated. The overload detection capabilities of each channel are controlled by the Overload Detection Control Registers (see Section 3.8.7 for further details).

2.4.5.2 Clock Signal Present LEDs (CLOCK A & B)

These green LEDs for each input channel are illuminated when a clock signal for that input is detected. If this LED is not illuminated, then no clock signal has been detected on the indicated input, and no data from that stream can be processed.

2.4.5.3 RACEway Master LED (RACEWAY MASTER)

This green LED is illuminated on the Model 6526 that is designated master of the RACEway interface (when the D6 bit in the RACEway Status Register, Section 3.5.6, is set to the logic '1' state). When this LED is off, the Model 6526 is not acting as a RACEway master.

2.4.5.4 Sync Bus Master LED (SYNC MASTER)

This yellow LED is illuminated on the Model 6526 that is designated as sync bus master (when the D0 bit in the Sync Bus Control Register, Section 3.4.2, is set to the logic '1' state). The sync master should be located at one end of the Sync bus, and the device at the opposite end must be programmed as the sync terminator (see Section 2.4.5.5, below). When this LED is off, the 6526 is not the sync master.

2.4.5.5 Sync Bus Terminator LED (TERM)

This yellow LED is illuminated on the Model 6526 that is designated as sync bus terminator (when the D1 bit in the Sync Bus Control Register, see Section 3.4.2, is set to the logic '1' state). The sync terminator should be located at one end of the Sync bus, and the device at the opposite end must be programmed as the sync master (see Section 2.4.5.4, above). When this LED is off, the 6526 is not the sync terminator.

2.4 Model 6526 Front Panel (continued)

2.4.5 LED Indicators (continued)

2.4.5.6 DSP LED

This yellow LED on the Model 6526 is controlled by the D9 bit in the 'C31 LED Register (see Section 3.8.1). The LED is illuminated when that bit is set to the logic '1' state, and the LED is off when the bit is cleared to the logic '0' state.

2.4.5.7 FAIL LED

This LED indicates a failure in the built–in self test on the Model 6526. This LED is illuminated when the D0 bit in the 'C31 LED Register is set to the logic '1' state (see Section 3.8.1).

2.5 Inserting and Removing the Model 6526 in the VMEbus Card Cage

CAUTION!!

Never INSERT or REMOVE the Model 6526 while card cage power is turned on!

2.5.1 Inserting the Unit

Align the Model 6526 card edge with the cage card guides and push evenly on both ejector handles until the front panel seats against the upper and lower rails. Secure the unit in the cage by tightening the upper and lower retaining screws in the front panel.

2.5.2 Removing the Unit

Loosen the upper and lower retaining screws in the front panel of the Model 6526 — these are captive screws and will remain part of the panel assembly. Gently push both ejector handles apart (away from the center of the panel) until the unit unseats itself from the cage connectors. Then, pull outward to remove the unit.

This page is intentionally blank

Chapter 3: Memory Maps and Register Descriptions

3.1 Programming Overview

The Model 6526 is controlled by commands sent over the VMEbus to memory−mapped control registers. Each of these registers is described in detail in this chapter. A VME−bus memory map to all the Model 6526's registers is included in this chapter. The 6526 acts as a VMEbus slave device.

Six control registers are mapped into the Model 6526's VMEbus A16 memory space. Access to this address space is determined by the A16 Base Address Switches, which are described in Section 2.2. All control register addresses are given as offsets from the base address defined by the settings of these switches. See Section 3.4 for description of the VMEbus A16 memory space.

Each Model 6526 board can occupy up to four distinct regions in A24 and/or A32 VMEbus space. Read/Write access to each individual 6526 board is provided in a unique address range determined by the contents of an A16 space register (see Section 3.4.4). An additional access region, accessible only for write operations, is provided in which all Model 6526s can be accessed simultaneously, in Broadcast mode, to simplify the programming of those operating parameters that are common among all boards. The Broadcast address range is determined by the contents of another A16 register (see Section 3.4.5). Access in each of these four address regions (A24 unique, A32 unique, A24 broadcast, and A32 broadcast) can be enabled or disabled at any time under pro−gram control. See Section 3.5 for description of the VMEbus A24/A32 memory space.

The Model 6526 includes a RACEway Master Interface. Logic is provided to add 32−bit words containing framing information and a small header to each RACEway data packet that is created and sent out for each data channel. The number of complex sam−ples in each packet is programmable via a VMEbus accessible register, ranging from 12 to 2044 32−bit words. (This number is always four words less than the total RACEway packet size because four additional words are required for the header and framing information.) The specification and ordering of all 32−bit words (framing, header, and data) in each RACEway packet is described in this chapter. See Section 3.6 for descrip−tion of the VMEbus A24/A32 channel formatter memory space.

The Model 6526 also includes Dual Port Memory (DPSRAM), which is used for passing parameters, commands, and status between the VMEbus and the 'C31 processor, and for supporting Pentek's SwiftNet communications software. It is arranged as 16k x 32 and is mapped into the address space of the 'C31 and into A24/A32 VMEbus address space. See Section 3.7 for description of the Dual Port Memory and Command Tables.

A complete memory map of the VME memory accessible to the TMS320C31 processor is provided in Section 3.8. Several of the registers that are unique to the 'C31 are also described in that section.

3.2 VMEbus Address Modifier Codes

There are three addressing modes defined for VMEbus transfers: 32−bit, 24−bit, and 16−bit modes. These modes are provided to support a variety of bus devices, ranging from simple I/O functions with only a few memory locations to large RAM arrays with hundreds of Megabytes of storage. The addressing mode used for a given transaction is determined by a special 6−bit code that is placed on six dedicated VMEbus lines by the current bus master. This code is known as the Address Modifier, or AM, code. The VMEbus specification defines the usage of certain AM codes as shown in Table 3−1, below. The Model 6526 responds only to those codes shown in the table.

Table 3−1: VMEbus Address Modifier Codes

AM Code	Access	AM Code	Access
0x09	A32 − Extended Non−privileged Data Access	0x0B	A32 − Extended Non−privileged Block Transfer
0x0D	A32 − Extended Supervisory Data Access	0x0F	A32 − Extended Supervisory Block Transfer
0x29	A16 − Short Non−Privileged Access	0x2D	A16 − Short Supervisory Access
0x39	A24 − Standard Non−privileged Data Access	0x3B	A24 − Standard Non−privileged Block Transfer
0x3D	A24 − Standard Supervisory Data Access	0x3F	A24 − Standard Supervisory Block Transfer
NOTE: The Model 6526 **ONLY** responds to the address modifier codes shown in this table.			

3.3 VMEbus Access

The Model 6526 contains two 16−position rotary switches (see Section 2.2, on page 17, for details) used to assign the base address of the 256−byte region in VMEbus A16 address space for various control and status registers. This address is referred to in subsequent sections of this manual as the "A16_base address", and all other A16 addresses are referenced as offsets from this address. The address form used is A16_base+0xYZ, where YZ represents the offset from the base address, in bytes.

In A16 address space on the Model 6526 are two A24/A32 base address registers. By writing to these base address registers during system initialization, you may set the VME base address for all of the A24 and A32 slave accessible functions on the board. The two base addresses provide redundant, identical images of all A24 and A32 resources on the board to two different VMEbus locations. One of these base address registers is designated as the Broadcast Base Address Register and provides write−only access. The second base address register is designated the Unique Base Address Regis−ter and provides full read/write access.

In using these features, you may assign the same Broadcast Base Address to several Model 6526 boards and send the same data to all of the boards simultaneously by writ−ing to addresses in the common region.

3.4 VMEbus A16 Slave Memory

VMEbus A16 address space on the Model 6526 is reserved for basic global board functions such as reset, Sync bus mode control, configuration of the A24 and A32 base address registers, and self test. The memory address map for the six resources available in VMEbus A16 space is given in Table 3–2, below. Note that most of these resources are accessible for both read and write operations (R/W). The exception is the VME–to–'C31 interrupt address, which is accessible only for write operations (W.O.).

To access these registers, the bus master that conducts the read or write cycle must select A16 address space by using AM code 2D or 29 (see Table 3–1, previous page). All resources on the Model 6526 are mapped on 4–byte boundaries (the least significant hex address digit is either 0, 4, 8, or C) to simplify master access to the board by DSP–based devices that use 32–bit addressing, such as Pentek's Models 4283, 4284, 4285, and 4290. Since A16 address space only supports 16–bit data transfers, only the 16 least significant data bits of the VMEbus are used.

Table 3–2: VMEbus A16 Slave Memory Map

Address	Access	Resource Name
A16_base+0x00	R/W	VMEbus Board Control Register
A16_base+0x04	R/W	Sync Bus Control Register
A16_base+0x08	W.O.	VMEbus Interrupt to 'C31 DSP
A16_base+0x0C	R/W	Unique A24/A32 Base Address Register
A16_base+0x10	R/W	Broadcast A24/A32 Base Address Register
A16_base+0x14	R/W	Built–In Self Test Control Register

All VMEbus A16 space addresses are expressed in hexadecimal notation, as offsets from the A16_base address. This address is determined by the settings of the two 16–position rotary switches, SW1 and SW2 (see Section 2.2). The A16 memory space occupied by this device is a 256–byte block starting at the A16_base address. All accesses within this region cause the Model 6526 to handshake with the VMEbus, **and no other VMEbus device may be mapped within this same 256–byte region of A16 address space**. If more than one 6526 is used in the same card cage, the A16 Base Address switches must be set to give each board a unique address.

The subsections beginning on the next page provide tables that show the contents of all of these registers, and give detailed descriptions of how the bits and bit–fields contained within these registers are used.

3.4 VMEbus A16 Slave Memory (continued)

3.4.1 VMEbus Board Control Register – R/W @ A16_base+0x00

The VMEbus Board Control Register utilizes 10 of the 16 bits at the Model 6526's A16_base address. Of these, four are reset functions (one of which is a Read–Only completion status bit), and five are access selection. The remaining bit is a word swap (endian) selection bit. Table 3–3, below, shows the arrangement of bits in the VMEbus Board Control Register. The subsections following the table describe these bits.

Table 3–3: VMEbus Board Control Register
R/W @ A16_base+0x00

Bit #	D15 – D10						D9*	D8
Bit Name	Reserved – Not Used						General_ Reset_Done	Word_ Swap
Function	Write with zeros, Mask when reading						1 = Cfg. Done 0 = Configuring	1 = Big Endian 0 = Little Endian
Bit #	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Bit Name	VME_ Access	A32_ Broadcast	A24_ Broadcast	A32_ Unique	A24_ Unique	Special_ HW_Reset	DSP_ Reset	General_ HW_Reset
Function	1 = VME 0 = 'C31	1 = Enabled 0 = Disabled				1 = Reset 0 = Run		

* This bit is Read Only
All bits default to the logic '0' state at power up, except for General_Reset_Done

3.4.1.1 General Hardware Reset – Bit D0

When this bit is set to the logic '1' state, all of the general hardware functions on the board, except the 'C31 Digital Signal Processor, are held in their reset conditions, and all A24/A32 registers are cleared. When this bit is cleared to the logic '0' state (its default condition), all general hardware functions on the board, except the 'C31, are released for normal operation.

3.4.1.2 DSP Reset – Bit D1

When this bit is set to the logic '1' state, the 'C31 Digital Signal Processor is held in its RESET state. When this bit transitions from the logic '1' state to the logic '0' state, the DSP begins execution from the RESET condition. The default logic state for this bit is '0'.

3.4 VMEbus A16 Slave Memory (continued)

3.4.1 VMEbus Board Control Register (continued)

3.4.1.3 Special Hardware Reset – Bit D2

When this bit is set to the logic '1' state, all of the general hardware functions on the board, except the 'C31 Digital Signal Processor, are held in their reset conditions, but the A24/A32 registers are NOT cleared. When this bit is cleared to the logic '0' state (its default condition), all general hardware functions on the board are released for normal operation.

3.4.1.4 A24 Unique Address Space Enable – Bit D3

When this bit is set to the logic '1' state, slave access to the Model 6526 in the A24 Unique Address Space is enabled using the address specified in the A24/A32 Unique Base Address Register (see Section 3.4.4). When this bit is cleared to the logic '0' state (its default condition), slave access to the Model 6526 in the A24 Unique Address Space is disabled.

3.4.1.5 A32 Unique Address Space Enable – Bit D4

When this bit is set to the logic '1' state, slave access to the Model 6526 in the A32 Unique Address Space is enabled using the address specified in the A24/A32 Unique Base Address Register (see Section 3.4.4). When this bit is cleared to the logic '0' state (its default condition), slave access to the Model 6526 in the A32 Unique Address Space is disabled.

3.4.1.6 A24 Broadcast Address Space Enable – Bit D5

When this bit is set to the logic '1' state, slave access to the Model 6526 in the A24 Broadcast Address Space is enabled using the address specified in the A24/A32 Broadcast Base Address Register (see Section 3.4.5). When this bit is cleared to the logic '0' state (its default condition), slave access to the Model 6526 in the A24 Broadcast Address Space is disabled.

3.4.1.7 A32 Broadcast Address Space Enable – Bit D6

When this bit is set to the logic '1' state, slave access to the Model 6526 in the A32 Broadcast Address Space is enabled using the address specified in the A24/A32 Broadcast Base Address Register (see Section 3.4.5). When this bit is cleared to the logic '0' state (its default condition), slave access to the Model 6526 in the A32 Broadcast Address Space is disabled.

3.4 VMEbus A16 Slave Memory (continued)

3.4.1 VMEbus Board Control Register (continued)

3.4.1.8 VMEbus Access – Bit D7

This bit must be set to the logic '1' state, to enable any A24/A32 VMEbus slave access to the registers and other resources on the local bus of the Model 6526. When this bit is cleared to the logic '0' state (its default condition), these resources are accessible ONLY to the on−board 'C31 DSP. Note that when VMEbus access is enabled (i. e., when this bit contains a logic '1'), 'C31 access to these resources is disabled.

3.4.1.9 Word Swap – Bit D8

This bit is used to select the endian mode, which is the order in which the most significant and least significant halves of 32−bit memory locations in A24/32 space are addressed when accessed in D16 mode. When this bit is set to the logic '1' state, Big Endian mode is selected, meaning that the least significant 16−bit half of a 32−bit word is accessed at an odd address. Host computers based on Motorola processors use Big Endian addressing mode. When this bit is cleared to the logic '0' state (its default condition), Little Endian mode is selected, meaning that the least significant 16−bit half of a 32−bit word is accessed at an even address. Host com−puters based on Intel processors use Little Endian addressing mode.

3.4.1.10 General Reset Done – Bit D9, Read Only

This read−only bit represents the status of the FPGA (Field Pro−grammable Gate Array) logic configuration after power up or after release of a General Hardware Reset (after the transition of the D0 bit in this register from the logic '1' state to the logic '0' state). Most A24/A32 registers on the Model 6526 will not be accessible until this configuration is complete. This bit will read back in the logic '0' state during the FPGA configuration cycle, and in the logic '1' state at all other times.

This bit can also be used to detect an FPGA configuration failure. If this bit does not read back in the logic '1' state within approxi−mately two seconds following the transition of the General Hard−ware Reset bit (D0) from logic '1' to logic '0', a configuration failure is indicated. Note that neither the DSP Reset (D1 in this register) nor the Special Hardware Reset (D2 in this register) initiate an FPGA configuration cycle.

3.4 VMEbus A16 Slave Memory (continued)

3.4.2 Sync Bus Control Register – R/W @ A16_base+0x04

Only two bits of this register are active. One bit is used to enable the Model 6526 as a sync bus master, and the other is used to enable the 6526 as a sync bus terminator. Table 3–4, below, shows the placement of these bits. The subsections following the table describe these bits.

Table 3–4: Sync Bus Control Register
R/W @ A16_base+0x04

Bit #	D15 – D2	D1	D0
Bit Name	Reserved – Not Used	Sync_Term	Sync_Master
Function	Write with zeros, Mask when reading	1 = Enable 0 = Disable	1 = Master 0 = Slave
All bits default to the logic '0' state at power up			

3.4.2.1 Sync Bus Master – Bit D0

When this bit is set to the logic '1' state, the Low Voltage TTL bus drivers for the front panel Sync bus connector are activated. This function should be enabled only for one Model 6526 in a group of 6526s sharing a common Sync bus, and the enabled board should be located at one end of the front panel ribbon cable. All other devices connected to that Sync bus should be slaves, which means that this bit should be cleared to the logic '0' state (its default condition) in those devices.

3.4.2.2 Sync Bus Termination – Bit D1

When this bit is set to the logic '1' state, the active termination circuitry for the front panel Sync bus is activated. This function should be enabled only for one Model 6526 in a group of 6526s sharing a common Sync bus, and the enabled board should be at the opposite end of the front panel ribbon cable from the Sync bus master (see Section 3.4.2.1, above). All other devices connected to that Sync bus should have the termination disabled, which means that this bit should be cleared to the logic '0' state (its default condition) in those devices.

3.4 VMEbus A16 Slave Memory (continued)

3.4.3 VMEbus Interrupt to 'C31 DSP – W. O. @ A16_base+0x08

Any write access to this address will generate an interrupt to the Model 6526's TMS320C31 DSP. This interrupt is asserted at the 'C31's $\overline{\text{INT1}}$ input. The data used for the write access is neither used nor stored, and is therefore arbitrary. Read operations to this address will return meaningless data.

3.4.4 VMEbus Unique A24/A32 Base Address Register – R/W @ A16_base+0x0C

This register holds the base addresses used to access a given Model 6526 in VMEbus A24 and/or A32 space. All bits in this register default to the logic '0' state. At power−up, only A16 access is enabled. After writing the desired unique base address for the unit into this register, access in the desired region(s) must be enabled by setting the appropriate bits in the VMEbus Board Control Register (D4 for unique A32 space and/or D3 for unique A24 space, see Section 3.4.1) to the logic '1' state. Table 3−5, below, shows the two fields in this register. The subsections following the table describe their use.

Table 3−5: VMEbus Unique A24/A32 Base Address Register
R/W @ A16_base+0x0C

Bit #	D15	D14	D13	D12	D11	D10	D9	D8
Bit Name	UA32_7	UA32_6	UA32_5	UA32_4	UA32_3	UA32_2	UA32_1	UA32_0
Function	VMEbus Unique A32 base address (Most significant eight bits)							
Bit #	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	UA24_6	UA24_5	UA24_4	UA24_3	UA24_2	UA24_1	UA24_0	Not Used
Function	VMEbus Unique A24 base address (Most significant seven bits)							Write '0', Mask Read
All bits default to the logic '0' state at power up								

3.4.4.1 VMEbus Unique A32 Base Address – Bits D15 to D8

These eight bits specify the lowest address in the region of VME−bus A32 address space that this 6526's unique A32−addressable resources will occupy. The board may occupy one of 256 regions, each representing 16 MBytes of VMEbus A32 address space. The value set in bits UA32_7 to UA32_0 should be unique for each Model 6526 in the system, so that when commands are issued in this address space, only the desired board will respond.

The eight most significant bits (the two most significant hex digits) of the VME address bus are compared with the contents of this field to determine if this is the board being addressed. Note that this value corresponds to the "VME_base" designation used in the A24/A32 address map descriptions in Section 3.5.

3.4 VMEbus A16 Slave Memory (continued)

3.4.4 VMEbus Unique A24/A32 Base Address Register (continued)

3.4.4.2 VMEbus Unique A24 Base Address – Bits D7 to D1

These seven bits specify the lowest address in the region of VME–bus A24 address space that this Model 6526's unique A24–addres–sable resources will occupy. The board may occupy one of 128 regions, each representing 128 kBytes of VMEbus A24 address space. The value set in bits UA24_6 to UA24_0 in this register should be unique for each Model 6526 in the system, so that when commands are issued in this address space, only the desired board will respond.

VME address bits A23 – A17 (the third and fourth hex address digits) are compared with the contents of this register field to determine if this is the board being addressed. Note that this value corresponds to the "VME_base" designation used in the A24/A32 address map descriptions in Section 3.5.

NOTES: (1) Because the Model 6526 occupies 128 kBytes of VMEbus memory in A24 address space, it can only be mapped on EVEN A24 VME_base addresses. Therefore, the least significant bit of this register must always be written with a logic '0'.

(2) **The contents of the eight least significant bits in this register have no effect upon the Unique A32 base address.** In other words, if you write 0xDCBA to this register, the Unique A32 base address is 0xDC00 0000, and the Unique A24 base address is 0xBA 0000.

3.4 VMEbus A16 Slave Memory (continued)

3.4.5 VMEbus Broadcast A24/A32 Base Address Register – R/W @ A16_base+0x10

This register holds the base addresses used for broadcast access to a group of Model 6526s in VMEbus A24 and/or A32 space. All bits in this register default to the logic '0' state. At power−up, only A16 access is enabled. After writing the desired broadcast base address for the unit into this register, access in the desired region(s) must be enabled by setting the appropriate bits in the VMEbus Board Control Register (D6 for A32 broadcast space and/or D5 for A24 broadcast space, see Section 3.4.1) to the logic '1' state. Table 3−6, below, shows the two fields in this register. The subsections that follow describe their use.

Table 3−6: VMEbus Broadcast A24/A32 Base Address Register
R/W @ A16_base+0x10

Bit #	D15	D14	D13	D12	D11	D10	D9	D8
Bit Name	BA32_7	BA32_6	BA32_5	BA32_4	BA32_3	BA32_2	BA32_1	BA32_0
Function	VMEbus Broadcast A32 base address (Most significant eight bits)							
Bit #	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	BA24_6	BA24_5	BA24_4	BA24_3	BA24_2	BA24_1	BA24_0	Not Used
Function	VMEbus Broadcast A24 base address (Most significant seven bits)							Write '0', Mask Read
All bits default to the logic '0' state at power up								

3.4.5.1 VMEbus Broadcast A32 Base Address − Bits D15 to D8

These eight bits specify the lowest address in the region in VME−bus A32 address space that a given group of Model 6526s' broad−cast A32−addressable resources will occupy. The boards may occupy one of 256 regions, each representing 16 MBytes of VME−bus A32 address space. The value set in bits BA32_7 to BA32_0 should be identical for all Model 6526s in a specific broadcast group, so that when commands are issued in this address space, all the desired boards will respond.

NOTE: This address region supports write accesses **ONLY!** Read functions are disabled and are supported only in the Unique Address Space. Read−only resources are inaccessible in A32 Broadcast Address Space.

The eight most significant bits (the two most significant hex digits) of the VME address bus are compared with the contents of this field to determine if this unit is among the boards being addressed. Note that this value corresponds to the "VME_base" designation used in the A24/A32 address map descriptions in Section 3.5.

3.4 VMEbus A16 Slave Memory (continued)

3.4.5 VMEbus Broadcast A24/A32 Base Address Register (continued)

3.4.5.2 VMEbus Broadcast A24 Base Address – Bits D7 to D1

These seven bits specify the lowest address in the region in VME–bus A24 address space that a given group of Model 6526s' broad–cast A24 address resources will occupy. The boards may occupy one of 128 regions, each representing 128 kBytes of VMEbus A24 address space. The value set in bits BA24_6 to BA24_0 in this reg–ister should be identical for all Model 6526s in a specific broadcast group, so that when commands are issued in this address space, all desired boards will respond.

NOTE: This address region supports write accesses **ONLY!** Read functions are disabled and are supported only in the Unique Address Space. Read–only resources are inaccessible in A24 Broadcast Address Space.

VME Address Bits A23 to A17 (i. e., the third and fourth hex address digits) are compared with the contents of this register field to determine if this unit is the among the boards being addressed. Note that this value corresponds to the "VME_base" designation used in the A24/A32 address map descriptions in Section 3.5.

NOTES: (1) Because the Model 6526 occupies 128 kBytes of VMEbus memory in A24 address space, it can only be mapped on EVEN A24 VME_base addresses. Therefore, the least significant bit of this register must always be written with a logic '0'.

(2) **The contents of the eight least significant bits in this register have no effect upon the Broadcast A32 base address.** In other words, if you write 0x1234 to this register, the Broadcast A32 base address is 0x1200 0000, and the Broadcast A24 base address is 0x34 0000.

3.4 VMEbus A16 Slave Memory (continued)

3.4.6 Built–In Self–Test (BIST) Control Register – R/W @ A16_base+0x14

Only one bit in this register is write–accessible, to enable the self–test cycle. One of the remaining bits indicates that the self–test is in progress, the other eight bits indicate the self–test results. Table 3–7, below, shows the register's bit layout. The subsections following the table describe these bits.

Table 3–7: Built–In Self–Test Control Register
R/W @ A16_base+0x14

Bit #	D15 – D9								
Bit Name	Reserved – Not Used								
Function	Write with zeros – Mask when reading								
Bit #	D8*	D7*	D6*	D5*	D4*	D3*	D2*	D1*	D0
Bit Name	GC4014_3 _Result	GC4014_2 _Result	GC4014_1 _Result	GC4014_0 _Result	DPSRAM_ _1_Result	DPSRAM _0_Result	LSRAM_ Result	BIST _Active	BIST _Enable
Function	0 = Passed 1 = Failed							1 = Active 0 = Inactive	1 = Run BIST 0 = Complete

* These bits are Read Only
All bits default to the logic '0' state at power up

3.4.6.1 BIST Enable – Bit D0

The self–test cycle is initiated by the on–board 'C31 processor when a VMEbus master sets this bit to the logic '1' state. This bit remains in the logic '1' state while the test executes, and the 'C31 DSP clears the bit to the logic '0' state (its default state) after the test has completed.

3.4.6.2 BIST Active – Bit D1, Read Only

This bit will read as logic '1' while the self–test is in progress. At all other times, this bit will read as logic '0' (its default state).

3.4.6.3 BIST Results – Bits D8 to D2, Read Only

This field is reserved for the DSP to write a code indicating the results of the self–test. For all tests, a logic '0' in the associated bit indicates that the test passed, and a logic '1' in the associated bit indicates that the test failed. Thus, if this register reads back with all bits cleared to the logic '0' state, the entire self–test passed.

Three bits in this field are for the results of on–board memory tests: bit D2 is for the 'C31 local SRAM test, bit D3 is for the low bank of dual–port SRAM, and bit D4 is for the high bank's test. The remaining four bits indicate the results of the tests on the GC4014 DDR chips: D5 is for DDR0 (receivers 0 – 3), D6 is for DDR 1 (receivers 4 – 7), D7 is for DDR2 (receivers 8 – 11), and D8 is for DDR3 (receivers 12 – 15).

3.5 VMEbus A24/A32 Global Slave Register Memory

This section describes control and status functions common to all receiver channels, such as the Time Stamp Counter and Sync bus generator commands. This section also describes special global access registers, which allow single command access to certain functions within all 16 channel formatters. These registers are mapped into both the broadcast and unique address spaces defined in the A16 registers for those functions. In the unique address space, all functions shown as read/write will support both reads and write operations. In the broadcast address space, all read functions are disabled, even though the accessibility shown at the top of each register content table might seem to indicate otherwise.

VMEbus A24 addresses are specified using 6 hexadecimal digits (or 24 bits). When we use the term 'VME_base' in discussing A24 resources, we refer to the hexadecimal equivalent of bits UA24_6 through UA24_0 in the VMEbus Unique Base Address Register (at address A16_base+0x0C), or BA24_6 through BA24_0 in the VMEbus Broadcast Base Address Register (at address A16_base+0x10), as defined in Sections 3.4.4.2 and 3.4.5.2, respectively, plus a zero least significant bit (representing A16). Since the least significant bit of the A24 VME_base must always be zero, VME_base is always restricted to even values in A24 space. Access to the Model 6526 in VMEbus A24 address space must be enabled by setting the E24U and/or E24B bits in the VMEbus Board Control Register (at address A16_base+0x00), described in Sections 3.4.1.4 and 3.4.1.6, to the logic '1' state.

VMEbus A32 addresses are specified using 8 hexadecimal digits (32 bits). When we use the term 'VME_base' in discussing A32 resources, we refer to the hexadecimal equivalent of bits UA32_7 through UA32_0 in the VMEbus Unique Base Address Register (at address A16_base+0x0C), or BA32_7 through BA32_0 in the VMEbus Broadcast Base Address Register (at address A16_base+0x10), as defined in Sections 3.4.4.1 and 3.4.5.1, respectively. Access to the Model 6526 in VMEbus A32 address space must be enabled by setting the E32U and/or E32B bits in the VMEbus Board Control Register (at address A16_base+0x00), described in Sections 3.4.1.5 and 3.4.1.7 to the logic '1' state.

Most of the registers that reside in this VMEbus address region are also accessible by the on−board TMS320C31 Digital Signal Processor. The base address for 'C31 access to these resources is 0x30 0000, and the 'C31 address offset of each resource is equal to the VME address offset divided by four (see Section 3.8 for further information on the 'C31 memory maps and registers). The resources that are accessible to VMEbus masters in this region that are not accessible to the 'C31 are those registers associated with FIFO access control. Table 3−8, at the top of the next page, shows the Memory Map for Global VMEbus Slave Register space on the Model 6526.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

Table 3–8: VMEbus A24/A32 Global Slave Register Memory Map

VMEbus Address	'C31 Address	Unique Access	Brdcast Access	Resource Name	Symbolic Address
VME_base+0x0000 0000	0x30 0000	R/W	W.O.	VMEbus Board Control Register	Board_Control
VME_base+0x0000 0004	0x30 0001	R/W	W.O.	Time Stamp Counter Control Register	Timestamp_Counter_Cntl
VME_base+0x0000 0008	0x30 0002	R.O.	N/A	Time Stamp Counter Output Register	Timestamp_Counter
VME_base+0x0000 000C	0x30 0003	R/W	W.O.	Sync Generator Mask Register	Sync_Gen_Mask
VME_base+0x0000 0010	0x30 0004	R/W	W.O.	Channel Formatter Reset Control Register	Ch_Format_Reset_Cntl
VME_base+0x0000 0014	0x30 0005	R.O.	N/A	RACEway Status Register	RACEway_Status
VME_base+0x0000 0018	0x30 0006	R/W	W.O.	Sync Generator Execute Register	Sync_Gen_Execute
VME_base+0x0000 001C	0x30 0007	R.O.	N/A	Output FIFO Status Register	FIFO_Status
VME_base+0x0000 0020	0x30 0008	R.O.	N/A	Clock Status Register	Clock_Status
VME_base+0x0000 0024 – 2C	0x30 0009 – B	—	—	Reserved	—
VME_base+0x0000 0030	—	R/W	N/A	Input FIFO A Delay Control Register	Input_A_Delay_Cntl
VME_base+0x0000 0034	—	R/W	N/A	Input FIFO B Delay Control Register	Input_B_Delay_Cntl
VME_base+0x0000 0038	—	R/W	N/A	Input FIFO A Control Register	Input_A_Control
VME_base+0x0000 003C	—	R/W	N/A	Input FIFO B Control Register	Input_B_Control

NOTES: These registers write globally to all formatters. Separate values for each channel formatter may also be read or written individually (see Section 3.6). They are accessible to the VMEbus when the VMEbus Access Enable bit in the VMEbus Board Control Register is set to the logic '1' state, or to the 'C31 when the VMEbus Access Enable bit is cleared to the logic '0' state.

3.5.1 VMEbus Board Control Register – R/W @ Board_Control

This register is the same register as the Board Control Register described in the A16 space registers in Section 3.4.1. It is mapped again here for convenience. Table 3–9, below, shows the arrangement of bits in the VMEbus Board Control Register. See Section 3.4.1 for the bit descriptions.

Table 3–9: VMEbus Board Control Register
R/W @ Board_Control

Bit #	D15 – D10						D9*	D8
Bit Name	Reserved – Not Used						General_ Reset_Done	Word_ Swap
Function	Write with zeros, Mask when reading						1 = Cfg. Done 0 = Configuring	1 = Big Endian 0 = Little Endian
Bit #	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Bit Name	VME_ Access	A32_ Broadcast	A24_ Broadcast	A32_ Unique	A24_ Unique	Special_ HW_Reset	DSP_ Reset	General_ HW_Reset
Function	1 = VME 0 = 'C31	1 = Enabled 0 = Disabled				1 = Reset 0 = Run		

* This bit is Read Only
All bits default to the logic '0' state at power up, except for General_Reset_Done

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.2 Time Stamp Counter Control Register – R/W @ Timestamp_Counter_Cntl

The Time Stamp Counter clock is driven by the output of a 10–bit pre–scaling frequency divider, followed by a divide by two stage (see Figure 3–1, below). The pre–scaler is driven by the either one of the two input data sample clocks. With the divide by two stage included, the input clock division factor ranges from 100 to 2048 in steps of 2. With a nominal 65 MHz input clock, the pre–scaler may be set to divide the input clock by 650 to produce a 100 kHz clock.

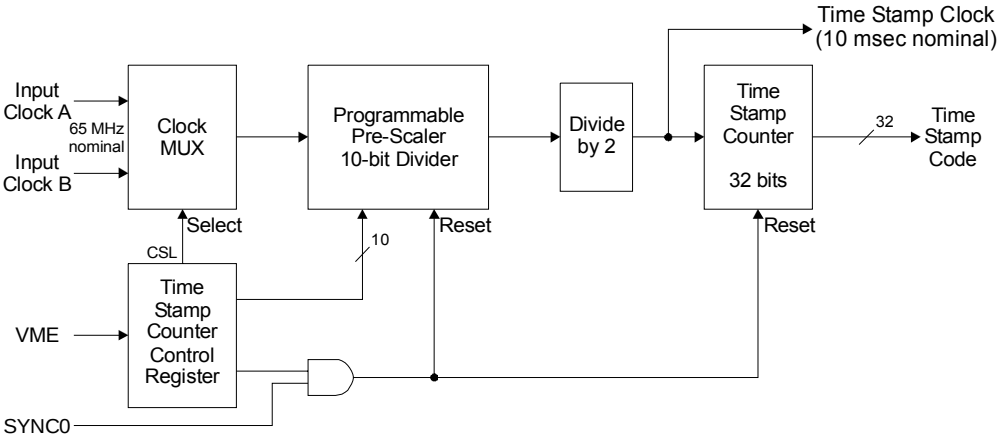


Figure 3–1: Time Stamp Counter Circuit

The Time Stamp Counter Control Register contains a 10–bit field that defines a pre–scaling factor (divisor), a bit that selects the clock source for the pre–scaler, and a bit to allow the SYNC0 signal from the front panel Sync bus to reset both the Time Stamp Counter and Time Stamp Pre–Scaler. Table 3–10, below, gives the bit layout of this register. The subsections following the table describes these bits.

Table 3–10: Time Stamp Counter Control Register
R/W @ Timestamp_Counter_Cntl

Bit #	D31 – D12											
Bit Name	Reserved – Not Used											
Function	Write with zeros, Mask when reading											
Bit #	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	PSD_9	PSD_8	PSD_7	PSD_6	PSD_5	PSD_4	PSD_3	PSD_2	PSD_1	PSD_0	PS_Clk_ Src_Select	Sync_Reset _Enable
Function	Time Stamp Pre–Scaler Divisor										1 = B input 0 = A Input	1 = Enable 0 = Disable
All bits default to the logic '0' state at power up												

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.2 Time Stamp Counter Control Register (continued)

3.5.2.1 Synchronous Reset Enable – Bit D0

When this bit is set to the logic '1' state, both the Time Stamp Counter and the Time Stamp Pre–Scaler are reset to a count of zero by the active state of the SYNC0 signal from the front panel Sync bus. When this bit is cleared to the logic '0' state (its default condition), the Time Stamp Counter and Time Stamp Pre–Scaler are not affected by the state of the SYNC0 signal.

3.5.2.2 Pre–Scaler Clock Source Select – Bit D1

This bit selects the front panel digital A/D input sample clock that is used as the clock signal for the Time Stamp Pre–Scaler. When this bit is cleared to the logic '0' state (its default condition), the clock signal from A/D input A is selected. When this bit is set to the logic '1' state, the clock signal from A/D input B is selected.

3.5.2.3 Pre–Scaler Divisor – Bits D11 to D2

This 10–bit value determines the divisor used by the Time Stamp Pre–Scaler. All bits in this field default to the logic '0' state. The divisor is determined by adding one to the binary value of bits PSD_9 to PSD_0 and multiplying by 2, according to the formula:

Divisor = ([PSD_9 to PSD_0] + 1) x 2

The valid range of effective divisor settings is all even integers from 100 to 2048. Thus, valid data for the Time Stamp Pre–Scaler Divisor field range from 0x31 (decimal 49) to 0x3FF (decimal 1023).

3.5.3 Time Stamp Counter Output Register – R. O. @ Timestamp_Counter

This read–only register, illustrated in Table 3–11 below, contains a 32–bit binary word showing the current value of the Time Stamp Counter.

Table 3–11: Time Stamp Counter Output Register
R. O. @ Timestamp_Counter

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	TS31	TS30	TS29	TS28	TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16
Function	16 Most Significant Bits of Time Stamp Counter Output															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Function	16 Least Significant Bits of Time Stamp Counter Output															

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.4 Sync Generator Mask Register – R/W @ Sync_Gen_Mask

This register contains four active bits that are used to determine which of the four front panel Sync bus signals is driven when the Sync Generate Command is issued as described in Section 3.5.7. Note that any or all of the Sync bus signals may be enabled. Setting one of these bits to the logic '1' state enables the SYNC Generate command to drive the associated Sync bus signal. Clearing one of these bits to the logic '0' state (their default condition) will leave the associated signal unaffected by the SYNC Generate command. Each Sync_Mask_n bit in the table below is associated with a corresponding SYNCn signal on the Sync bus (where 'n' = 0, 1, 2, or 3). Table 3–12, below, shows the bit layout of the Sync Generator Mask Register.

Table 3–12: Sync Generator Mask Register
R/W @ Sync_Gen_Mask

Bit #	D31 – D4	D3	D2	D1	D0
Bit Name	Reserved – Not Used	Sync_Mask_3	Sync_Mask_2	Sync_Mask_1	Sync_Mask_0
Function	Write with zeros, Mask when reading	1 = Enabled 0 = Disabled			
All bits default to the logic '0' state at power up					

3.5.5 Channel Formatter Reset Control Register – R/W @ Ch_Format_Reset_Cntl

This register contains 16 active bits, one for each digital receiver channel on the Model 6526. The number which follows FR in Table 3–13, below, designates which channel is affected, from 0 to 15 (e.g., FR12 is for channel 12). Setting the bit in this register associated with a given channel holds that channel formatter's state machine in its reset state and the clears the contents of its respective Output FIFO, making it empty. To receive data in a given channel, the bit in this register associated with that channel must be cleared to the logic '0' state (its default condition).

Table 3–13: Channel Formatter Reset Control Register
R/W @ Ch_Format_Reset_Cntl

Bit #	D31 – D16															
Bit Name	Reserved – Not Used															
Function	Write with zeros, Mask when reading															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	FR15	FR14	FR13	FR12	FR11	FR10	FR9	FR8	FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0
Function	1 = Reset 0 = Run															
All bits default to the logic '0' state at power up																

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.6 RACEway Status Register – R. O. @ RACEway_Status

This read–only register contains a 16–bit field indicating which receiver channels have data ready for transmission over the RACEway, and a 4–bit Active Channel field, indicating the receiver channel that is currently selected for RACEway service. It also contains six other RACEway status bits (Sending, Suspended, Master, Slave, Master Go, and Master Error), and five diagnostic bits for factory use. Table 3–14, below, shows the bit layout of the RACEway Status Register. The subsections following the table give descriptions of the bits in this register.

Table 3–14: RACEway Status Register
R. O. @ RACEway_Status

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	RQ15	RQ14	RQ13	RQ12	RQ11	RQ10	RQ9	RQ8	RQ7	RQ6	RQ5	RQ4	RQ3	RQ2	RQ1	RQ0
Function	1 = RACEway Channel Request Pending 0 = No Request Pending															
Bit #	D15		D14		D13		D12		D11		D10		D9		D8	
Bit Name	State_3		State_2		State_1		State_0		Request_ Flag_Found		Not Used		Master_Error		Master_Go	
Function	State Machine Present State (for factory diagnostic purposes only)								(diagnostic use only)		Write '0', Mask Read		1 =Error on Master Read		1 = Master State Machine Started	
Bit #	D7		D6		D5		D4		D3		D2		D1		D0	
Bit Name	Slave		Master		Suspended		Sending		Active_Ch_3		Active_Ch_2		Active_Ch_1		Active_Ch_0	
Function	1 = Slave 0 = Not Slave		1 = Master 0 = Not Master		1 = Suspended 0 = Not Suspended		1 = Sending 0 = No Packet		Receiver Channel selected for RACEway service (binary)							

3.5.6.1 Active Channel – Bits D3 to D0

This 4–bit field specifies the channel number of the channel presently selected to be serviced by the RACEway controller. The range of the Active Channel number is from 0 to 15 decimal, (encoded in 4–bit binary), covering all 16 channels on the board. The RACEway controller will sequence through all channels which have been enabled for RACEway sending by appropriately setting the Channel Request (RQ) bit(s), as defined in Section 3.5.6.9, on the page after next.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.6 RACEway Status Register (continued)

3.5.6.2 Sending – Bit D4

When this bit reads back in the logic '1' state, it signifies that the DDR channel indicated by the content of the Active Channel field (see Section 3.5.6.1, on the previous page) is currently sending a data packet over the RACEway interface. When this bit reads back in its default logic '0' state, no data packets are currently being sent over the RACEway interface.

3.5.6.3 Suspended – Bit D5

When this bit reads back in the logic '1' state, it signifies that the transmission of the data packet that was being sent by the DDR channel indicated by the content of the Active Channel field (see Section 3.5.6.1, on the previous page) is suspended. When a packet is being sent over the RACEway interface, or when no packet is in the suspended state, this bit will read back in its default logic '0' state.

3.5.6.4 Master – Bit D6

The PitCREW RACEway controller will set this bit to the logic '1' state whenever it is mastering a RACEway transaction. If this bit reads back in the '0' state, then the PitCREW on the Model 6526 is not mastering a RACEway transaction. This bit follows the state of the PitCREW chip's MASTER signal (see Appendix D).

3.5.6.5 Slave – Bit D7

The PitCREW RACEway controller will set this bit to the logic '1' state whenever it is participating in a RACEway transaction as a slave. If this bit reads back in the '0' state, then the PitCREW on the Model 6526 is not currently a RACEway slave. This bit follows the state of the PitCREW's SLAVE signal (see Appendix D).

3.5.6.6 Master Go – Bit D8

This bit is a readback of the PitCREW RACEway controller's MASTER GO input signal, which is generated by the Model 6526's RACEway state machine. This is an asynchronous signal that indicates that the state machine has been started when set to the logic '1' state. This bit will generally read back in the logic '0' state.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.6 RACEway Status Register (continued)

3.5.6.7 Master Error – Bit D9

The PitCREW RACEway controller will set this bit to the logic '1' state when it has encountered an error during a master read transaction. This bit will read back in its default logic '0' state under all other circumstances.

3.5.6.8 Diagnostics – Bits D15 to D11

These five bits are included for factory diagnostic purposes. The Request Flag Found bit (D11) is set to the logic '1' state whenever a receiver channel sets its request flag bit in this register (see Section 3.5.6.9, below). Under all other conditions, this bit will read back in its default logic '0' state.

The other four diagnostic bits (D15 to D12) comprise a field that indicates the state of the Model 6526's state machine. These four bits may be masked or simply ignored when reading this register.

3.5.6.9 Channel Request – Bits D31 to D16

This field contains one bit associated with each digital receiver channel on the Model 6526. The RQn bit is set to the logic '1' state when receiver channel n has data ready to be transmitted over the RACEway interface, where 'n' is the channel number from 0 to 15 (e. g., receiver channel 9 will set the RQ9 bit). Note that any or all of these bits may be set simultaneously. When one of these bits is in its default logic '0' state, the associated receiver channel does not have data ready for RACEway transmission.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.7 Sync Generator Execute Register – R/W @ Sync_Gen_Execute

This register contains three active bits, two of which are read–only status bits for the sync generator flip–flop after synchronization. The lone read/write bit is the sync generate command bit. Table 3–15, below, shows the bit arrangement of the Sync Generator Execute Register. The subsections following the table give descriptions of these bits.

Table 3–15: Sync Generator Execute Register
R/W @ Sync_Gen_Execute

Bit #	D31 – D3	D2*	D1*	D0
Bit Name	Reserved – Not Used	Clock_B_ Readback	Clock_A_ Readback	Sync_Gen _Command
Function	Write with zeros, Mask when reading	See Sec. 3.5.7.3	See Sec. 3.5.7.2	1 = Generate 0 = Off
* These bits are Read Only				

3.5.7.1 Sync Generate Command – Bit D0

This bit is used to generate Sync bus signals in the Model 6526 that is configured as the Sync Master. When this bit's state is switched from logic '0' to logic '1', the front panel Sync bus lines which have had their Sync Generate Mask bits enabled (see Section 3.5.4) will become active. The active edge of the Sync bus signals is the low–to–high transition.

3.5.7.2 Clock A Readback – Bit D1, Read Only

This bit shows the status of the Sync Generator flip–flop after the Sync Generate Command bit has been synchronized by input Clock A. This is useful in allowing the 'C31 to verify that the Sync Generate Command flip–flop is set, so that the 'C31 will not return the bit value to zero before synchronization and execution on the Sync bus.

3.5.7.3 Clock B Readback – Bit D2, Read Only

This bit shows the status of the Sync Generator flip–flop after the Sync Generate Command bit has been synchronized by input Clock B. This is useful in allowing the 'C31 to verify that the Sync Generate Command flip–flop is set, so that the 'C31 will not return the bit value to zero before synchronization and execution on the Sync bus.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.8 Output FIFO Status Register – R. O. @ FIFO_Status

The 32 bits in this register reflect the current, unlatched status of the Full and Empty flags from each of the Model 6526's Output FIFOs. If these FIFO flags are used to interrupt the DSP, then the latched Interrupt Status Registers (see Section 3.8.10) should be read to determine which flag(s) caused the interrupt. The number which follows FF or FE in Table 3–16, below, designates which channel is affected (e.g., FF5 is for channel 5). The subsections following the table give descriptions of these bits.

Table 3–16: Output FIFO Status Register
R. O. @ FIFO_Status

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	FF15	FF14	FF13	FF12	FF11	FF10	FF9	FF8	FF7	FF6	FF5	FF4	FF3	FF2	FF1	FF0
Function	1 = FIFO Full 0 = FIFO Not Full															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	FE15	FE14	FE13	FE12	FE11	FE10	FE9	FE8	FE7	FE6	FE5	FE4	FE3	FE2	FE1	FE0
Function	1 = FIFO Empty 0 = FIFO Not Empty															

3.5.8.1 FIFO Empty Flags – Bits D15 to D0

When the Output FIFO in a given channel is empty, the bit in this register for that channel's number (e.g., FE5 for channel 5) is set to the logic '1' state. These bits are not latched, so they will transition to the logic '0' state whenever any data is present in the associated Output FIFO.

3.5.8.2 FIFO Full Flags – Bits D31 to D16

When the Output FIFO in a given channel is full, the bit in this register for that channel's number (e.g., FF5 for channel 5) is set to the logic '1' state. These bits are not latched, so they will transition to the logic '0' state whenever any space is available in the associ–ated Output FIFO.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.9 Clock Status Register – R. O. @ Clock_Status

This register contains three read–only bits, which are used to indicate that the associated clock signals are active. One bit is associated with the clock signal on input A, another with the clock signal on input B, and the third is associated with the clock signal on the RACEway backplane fabric. For any of these three bits, a logic '1' indicates that a clock signal is active on the associated interface, and a logic '0' indicates that no clock signal is detected there. This information is summarized in Table 3–17, below.

Table 3–17: Clock Status Register
R. O. @ Clock_Status

Bit #	D31 – D3	D2	D1	D0
Bit Name	Reserved – Not Used	RACEway_ Clock_Status	Chan_A_ Clock_Status	Chan_B_ Clock_Status
Function	Write with zeros, Mask when reading	1 = Active 0 = Not Active		

3.5.10 Input Data De–Skewing FIFOs

The Graychip GC4014 receivers require that all parallel data inputs to a given receiver chip utilize the same clock signal. Since the Model 6526 uses two of the four inputs on the GC4014, the clocks for each input must be at the same frequency. Also, the clock edge to data transition relationship (data–to–clock setup time) must be the same for both inputs to the receiver chip.

With data sources like the WJ–9107 (see Section 2.4.1.1), a clock is supplied with each data channel. While the data–to–clock setup time may be maintained within each channel, there is no guarantee that the phase of the clocks is maintained between two channels. In fact, the relative phase of the two clocks will be initially different between two channels from the same WJ–9107 and will definitely be different between two WJ–9107's, even though the two units share a common frequency reference. In addition, the two channels will be subject to phase differences because of cable length and will normally drift in phase over time and temperature.

This would cause a significant problem for the GC4014 since it can only accept a single clock for both data channels. In order to solve the problem, the Model 6526 incorporates dual–input FIFO memories. At the input, each of these Input FIFOs accepts both clock and data from one of the two input sources. Data is clocked out of both Input FIFOs and into the GC4014s by a common clock (which can be either one of the two input clocks).

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.10 Input Data De−Skewing FIFOs (continued)

This scheme allows the Input FIFOs to act as elastic buffers to accommodate phase skewing between the two input channels to eliminate the problems noted above. Circuitry surrounding the FIFOs helps establish a buffer head−room of several data samples so that even quite significant phase skewing can be handled. This circuitry is controlled by the four registers that are discussed in the next four sections, beginning on the next page.

The block diagram of this section is shown in Figure 3−2, below. At the input side of each FIFO, data is clocked directly from its respective front panel input source. At the output side of each FIFO, data is clocked into the GC4014s by the selected input clock signal.

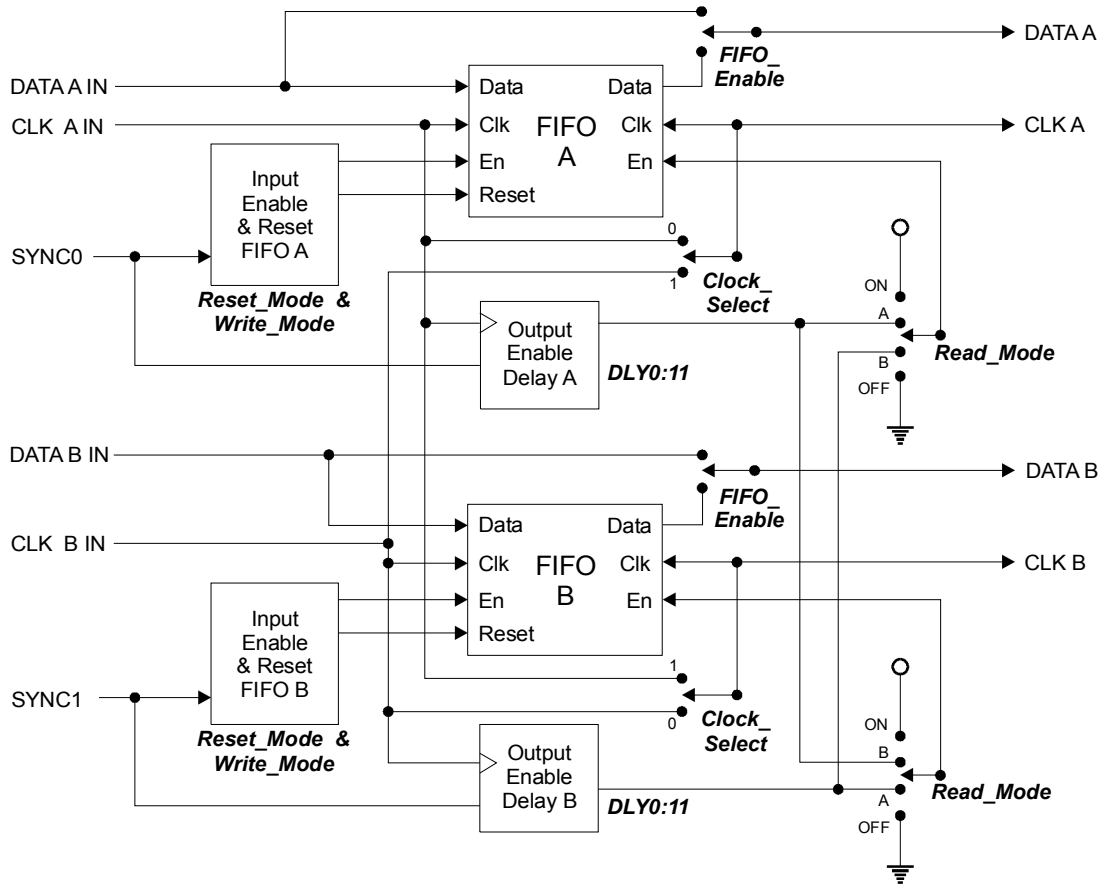


Figure 3−2: Model 6526 − Input Data De−Skewing FIFOs & Control Circuitry

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.10 Input Data De–Skewing FIFOs (continued)

The input data and clock for each Input FIFO are provided from the two front panel connectors (through the input line receivers). In order to assure synchronization between the two channels and between Model 6526 boards, the Input Enable and Reset circuitry allows the FIFOs to be reset and then allows the FIFO input enables to be turned on following Sync bus signals, common to all boards in a Sync bus group. The Reset Mode and Write Mode control bits in the Input FIFO Control Registers (described in Sections 3.5.13 and 3.5.14) control these functions.

The Output Enable Delay Generators allow a programmable number of input data samples to be written into the FIFO before the output clock is enabled. These samples support the elasticity required to support clock phase slippage between channels. Once the output enable becomes active, since the average rates of the input and output clocks are equal, the number of samples in the FIFO should remain more or less constant. In most systems, there should only need to be three or four samples at most.

The process of setting up an initial buffer of samples in the Input FIFO is required only once at system initialization, or if the input clocking or cables are changed.

A typical scenario for setting up the Model 6526 for the WJ–9107 receiver is as follows. Refer to Figure 3–2, on the previous page, for bit references.

- Assuming that the Input A clock is to be used as the common clock for the GC4014, the Clock_Select bits for both FIFO A and B must select Clock A.
- Set the FIFO_Enable bits to enable both FIFO paths.
- Set the Read_Mode bits to disable reads of both FIFOs.
- Set the Reset_Mode and Write_Mode bits to reset both FIFOs and disable writes to both FIFOs.
- Set the Reset_Mode and Write_Mode bits to enable writes to both FIFOs after SYNC.
- Program the Output Enable Delay Generators for a depth of 4 samples (i. e., 4 clock cycles of delay).
- Set the Read_Mode bits to enable reads of both FIFOs after SYNC0.
- Generate simultaneous SYNC0 and SYNC1 pulses to start the input data, the delay generators, and the output data.
- Set the Read_Mode, Reset_Mode, and Write_Mode bits to enable reads and writes for both FIFOs.

The bit definitions for all functions discussed above are given in detail in Sections 3.5.11 through 3.5.14, which follow immediately.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.11 Input FIFO A Delay Control Register – R/W @ Input_A_Delay_Control

This register controls the delay created by Output Enable Delay Generator A. (Refer to Section 3.5.10 and Figure 3–2 for additional information.) The purpose of this delay is to let the Input FIFOs fill with enough samples to sufficiently buffer phase variations between the two input clocks.

The 12–bit value programmed for DLY11 to DLY0 determines the number of input clocks from CLK A IN to CLK A. DLY0 is the least significant bit, and DLY11 is the most significant. When the SYNC0 signal occurs, an internal counter starts counting the cycles of CLK A IN, and generates a FIFO output enable after the number of clock cycles written to this register have elapsed. This can be used to enable either FIFO A or FIFO B, depending on how the Read Mode bits in the Input FIFO Control Registers are set (see Sections 3.5.13.3 and 3.5.14.3). Table 3–18, below, shows the contents of this register.

Table 3–18: Input FIFO A Delay Control Register
R/W @ Input_A_Delay_Control

Bit #	D31 – D12											
Bit Name	Reserved – Not Used											
Function	Write with zeros, Mask when reading											
Bit #	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	DLY11	DLY10	DLY9	DLY8	DLY7	DLY6	DLY5	DLY4	DLY3	DLY2	DLY1	DLY0
Function	Number of clocks from SYNC0 to FIFO Output Enable A											

3.5.12 Input FIFO B Delay Control Register – R/W @ Input_B_Delay_Control

This register controls the delay created by Output Enable Delay Generator B. The 12–bit value in this register is coded the same as in the Input FIFO A Delay Control Register, Section 3.5.11, above.

When the SYNC1 signal occurs, an internal counter starts counting the cycles of CLK B IN, and generates a FIFO output enable after the number of clock cycles written to this register have elapsed. This can be used to enable either FIFO A or FIFO B, depending on how the Read Mode bits in the Input FIFO Control Registers are set (see Sections 3.5.13.3 and 3.5.14.3). Table 3–19, below, shows the contents of this register.

Table 3–19: Input FIFO B Delay Control Register
R/W @ Input_B_Delay_Control

Bit #	D31 – D12											
Bit Name	Reserved – Not Used											
Function	Write with zeros, Mask when reading											
Bit #	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	DLY11	DLY10	DLY9	DLY8	DLY7	DLY6	DLY5	DLY4	DLY3	DLY2	DLY1	DLY0
Function	Number of clocks from SYNC1 to FIFO Output Enable B											

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.13 Input FIFO A Control Register – R/W @ Input_A_Control

This register controls the data input and output enables, external sync, input word length, and reset functions for Input FIFO A. (Refer to Section 3.5.10 and Figure 3–2 for additional information.) This FIFO allows synchronization of both input channels with one clock when both inputs are of the same rate but not in phase. Table 3–20, below, shows this register's bit layout, and the subsections following the table describe these bits and fields.

Table 3–20: Input FIFO A Control Register
R/W @ Input_A_Control

Bit #	D31 – D10									
Bit Name	Reserved – Not Used									
Function	Write with zeros, Mask when reading									
Bit #	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Front_Panel_ Sync_Enable	Input_Word _Length	FIFO_ Enable	Clock_ Select	Read_ Mode_1	Read_ Mode_0	Write_ Mode_1	Write_ Mode_0	Reset_ Mode_1	Reset_ Mode_0
Function	1 = Enable 0 = Disable	1 = 16 bits 0 = 12 bits	1 = Enable 0 = Bypass	1 = Chan B 0 = Chan A	0,0 = Enabled 0,1 = Sync0 Enable 1,0 = Sync1 Enable 1,1 = Disabled		0,0 = Enabled 0,1 = Sync0 Enable 1,x = Disabled		0,0 = Reset Off 0,1 = Sync0 Reset 1,x = Hold Reset	
All bits default to the logic '0' state at power up and reset										

3.5.13.1 Reset Mode – Bits D1 to D0

These two bits determine when and how Input FIFO A is reset. These reset modes are summarized in Table 3–21 below. When both of these bits are cleared to the logic '0' state, Input FIFO A is ready to accept data (or is accepting data), and will remain so until the state of bit D1 changes. When D1 is set to the logic '1' state (regardless of the state of D0), the FIFO is placed in its reset state — all data in FIFO A is cleared and the Input FIFO A Full Flag is cleared in the Interrupt Status Register (see Section 3.8.10). When bit D0 is set to the logic '1' state and D1 is a logic '0', Input FIFO A is reset when the SYNC0 signal goes to a logic '0', and remains in its reset state until SYNC0 goes to a logic '1'.

Table 3–21: Input FIFO A Reset Mode

D1	D0	Reset Mode
0	0	Reset Off (default after reset)
0	1	Reset on Sync0
1	Don't Care	Hold in Reset

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.13 Input FIFO A Control Register (continued)

3.5.13.2 Write Mode – Bits D3 to D2

These two bits determine when and how data can be written to Input FIFO A. When both of these bits are cleared to the logic '0' state, Input FIFO A is ready to accept data (or is accepting data), and will remain write–enabled until the state of bit D3 changes. When D3 is set to the logic '1' state (regardless of the state of D2), the FIFO write is disabled — all data in FIFO A at the time of this transition is retained, but no further data is accepted. When bit D2 is set to the logic '1' state and D3 is a logic '0', Input FIFO A is write–enabled when SYNC0 goes to a logic '1'.

Table 3–22: Input FIFO A Write Mode

D3	D2	Write Mode
0	0	Write Enabled (default after reset)
0	1	Write Enabled on Sync0
1	Don't Care	Write Disabled

3.5.13.3 Read Mode – Bits D5 to D4

These two bits determine when and how data can be read from Input FIFO A. When both of these bits are cleared to the logic '0' state, Input FIFO A is ready to output data (or is outputting data). When both bits are set to the logic '1' state, the FIFO read is dis–abled — all data in FIFO A is retained and data is accepted until the FIFO is filled, but no data may be read from the FIFO.

When bit D4 is set to logic '1' and D5 is a logic '0', Input FIFO A is read–enabled after the delay set in the Input FIFO A Delay Control Register has elapsed following the assertion of the SYNC0 signal (see Section 3.5.11). When bit D5 is set to logic '1' and D4 is a logic '0', Input FIFO A is read–enabled after the delay set in the Input FIFO B Delay Control Register has elapsed following the assertion of the SYNC1 signal (see Section 3.5.12).

Table 3–23: Input FIFO A Read Mode

D5	D4	Read Mode
0	0	Read Enabled (default after reset)
0	1	Read Enabled on FIFO A Delay after Sync0
1	0	Read Enabled on FIFO B Delay after Sync1
1	1	Read Disabled

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.13 Input FIFO A Control Register (continued)

3.5.13.4 Clock Select – Bit D6

This bit selects which input's clock signal is used as the Read clock for Input FIFO A. Clear this bit to the logic '0' state to select the clock signal from Input A. Set this bit to the logic '1' state to select the clock signal from Input B.

3.5.13.5 FIFO Enable – Bit D7

This bit determines whether or not the Input FIFO is used. If you are only using one input for all channels, or if you are certain that the clock signals in the two inputs are **ABSOLUTELY** identical in frequency and phase, the input FIFO may be bypassed by clearing this bit to the logic '0' state (its default state). To allow for phase differences between the two inputs by using Input FIFO A, set this bit to the logic '1' state.

3.5.13.6 Input Word Length – Bit D8

This bit determines the length of the input data words that Input FIFO A will accept. To program Input FIFO A to accept 12–bit input data, clear this bit to the logic '0' state (its default state). This data will occupy the upper 12 bits (D15 – D4) of the output data word when read from the FIFO. When this bit is set to the logic '1' state, Input FIFO A will accept 16–bit input data.

When this bit is cleared to the logic '0' state, to select 12–bit input data, the four least significant input data bits to FIFO A are actually forced to the logic '0' state. Thus, for 14–bit input sources, this bit should be set to logic '1' state, to select the 16–bit input mode.

3.5.13.7 Front Panel Sync Enable – Bit D9

This bit enables the System Sync (SS) input on the front panel dig–ital input connector. One external sync input is available for each of the two digital inputs. These signals are designated Ch A SS and Ch B SS, and are brought into the Model 6526 along with the digital data inputs, on the same cable. The SS signals are at the same logic levels as the digital input data, and are synchronized to the CK clock input signals. The SS and CK input pin designations are shown in the table in Figure 2–7.

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.13 Input FIFO A Control Register (continued)

3.5.13.7 Front Panel Sync Enable (continued)

External sync input is enabled for each channel input by setting Bit D9 in this register to the logic '1' state. When this bit is set, it is OR'ed with the Sync bus signals as follows:

- Ch A SS is logically OR'ed with SYNC0 and SYNC1
- Ch B SS is logically OR'ed with SYNC2 and SYNC3

This allows externally generated events to control the same on–board hardware controlled by the SYNC0, SYNC1, SYNC2, and SYNC3 signals. Note that the SS inputs are NOT sent out over the LVDS Sync bus connector — they are only used locally on those boards programmed to use them.

To use the external SS inputs correctly, the following rules apply:

1) The active edge of the SS inputs is the low–to–high transition.

2) Ch A SS must be synchronous with the Ch A CK (clock) signal.

3) Ch B SS must be synchronous with the Ch B CK (clock) signal.

4) Ch A SS is enabled by setting the D9 bit in the Input FIFO A Control Register to the logic '1' state (see Section 3.5.13). If the Ch A SS signal line is not driven by an external source, this bit should be cleared to the logic '0' state to avoid oscillations.

5) Ch B SS is enabled by setting the D9 bit in the Input FIFO B Control Register to the logic '1' state (see Section 3.5.14). If the Ch B SS signal line is not driven by an external source, this bit should be cleared to the logic '0' state to avoid oscillations.

6) Ch A SS can be used to generate sync functions within the two GC4014s for channels 0 through 7, and channel formatters 0 through 7, by clearing the CSEL0, CSEL1, SSEL0, and SSEL1 bits (D0, D1, D4, & D5, respectively) to the logic '0' state, in the ‘C31 Control Register (see Section 3.8.5).

7) Ch B SS can be used to generate sync functions within the two GC4014s for channels 8 through 15, and channel formatters 8 through 15, by setting the CSEL2, CSEL3, SSEL2, and SSEL3 bits (D2, D3, D6, & D7, respectively) to the logic '1' state in the ‘C31 Control Register (see Section 3.8.5).

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.14 Input FIFO B Control Register – R/W @ Input_B_Control

This register controls the data input and output enables, external sync, input word length, and reset functions for Input FIFO B. (Refer to Section 3.5.10 and Figure 3–2 for additional information.) This FIFO allows synchronization of both input channels with one clock when both inputs are of the same rate but not in phase. Table 3–24, below, gives this register's bit layout, and the sub–sections following the table describe these bits and fields.

Table 3–24: Input FIFO B Control Register
R/W @ Input_B_Control

Bit #	D31 – D10									
Bit Name	Reserved – Not Used									
Function	Write with zeros, Mask when reading									
Bit #	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Bit Name	Front_Panel_ Sync_Enable	Input_Word _Length	FIFO_ Enable	Clock_ Select	Read_ Mode_1	Read_ Mode_0	Write_ Mode_1	Write_ Mode_0	Reset_ Mode_1	Reset_ Mode_0
Function	1 = Enable 0 = Disable	1 = 16 bits 0 = 12 bits	1 = Enable 0 = Bypass	1 = Chan A 0 = Chan B	0,0 = Enabled 0,1 = Sync1En 1,0 = Sync0En 1,1 = Disabled		0,0 = Enabled 0,1 = Sync1 Enable 1,x = Disabled		0,0 = Reset Off 0,1 = Sync1 Reset 1,x = Hold Reset	
All bits default to the logic '0' state at power up and reset										

3.5.14.1 Reset Mode – Bits D1 to D0

These two bits determine when and how Input FIFO B is reset. These reset modes are summarized in Table 3–25 below. When both of these bits are cleared to the logic '0' state, Input FIFO B is ready to accept data (or is accepting data), and will remain so until the state of bit D1 changes. When D1 is set to the logic '1' state (regardless of the state of D0), the FIFO is placed in its reset state — all data in FIFO B is cleared and the Input FIFO B Full Flag is cleared in the Interrupt Status Register (see Section 3.8.10). When bit D0 is set to the logic '1' state and D1 is a logic '0', Input FIFO B is reset when the SYNC1 signal goes to a logic '0', and remains in its reset state until SYNC1 goes to a logic '1'.

Table 3–25: Input FIFO B Reset Mode

D1	D0	Reset Mode
0	0	Reset Off (default after reset)
0	1	Reset on Sync1
1	Don't Care	Hold in Reset

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.14 Input FIFO B Control Register (continued)

3.5.14.2 Write Mode – Bits D3 to D2

These two bits determine when and how data can be written to Input FIFO B. When both of these bits are cleared to the logic '0' state, Input FIFO B is ready to accept data (or is accepting data), and will remain write–enabled until the state of bit D3 changes. When D3 is set to the logic '1' state (regardless of the state of D2), the FIFO write is disabled — all data in FIFO B at the time of the transition is retained, but no further data is accepted. When bit D2 is set to the logic '1' state and D3 is a logic '0', Input FIFO B is write–enabled when the SYNC1 signal goes to the logic '1' state.

Table 3–26: Input FIFO B Write Mode

D3	D2	Write Mode
0	0	Write Enabled (default after reset)
0	1	Write Enabled on Sync1
1	Don't Care	Write Disabled

3.5.14.3 Read Mode – Bits D5 to D4

These two bits determine when and how data can be read from Input FIFO B. When both of these bits are cleared to the logic '0' state, Input FIFO B is ready to output data (or is outputting data). When both bits are set to the logic '1' state, the FIFO read is dis–abled — all data in FIFO B is retained and data is accepted until the FIFO is filled, but no data may be read from the FIFO.

When bit D4 is set to logic '1' and D5 is a logic '0', Input FIFO B is read–enabled after the delay set in the Input FIFO B Delay Control Register has elapsed following the assertion of the SYNC1 signal (see Section 3.5.11). When bit D5 is set to logic '1' and D4 is a logic '0', Input FIFO B is read–enabled after the delay set in the Input FIFO B Delay Control Register has elapsed following the assertion of the SYNC0 signal (see Section 3.5.12).

Table 3–27: Input FIFO B Read Mode

D5	D4	Read Mode
0	0	Read Enabled (default after reset)
0	1	Read Enabled on FIFO A Delay after Sync1
1	0	Read Enabled on FIFO B Delay after Sync0
1	1	Read Disabled

3.5 VMEbus A24/A32 Global Slave Register Memory (continued)

3.5.14 Input FIFO B Control Register (continued)

3.5.14.4 Clock Select – Bit D6

This bit selects which input's clock signal is used as the Read clock for Input FIFO B. Clear this bit to the logic '0' state to select the clock signal from Input B. Set this bit to the logic '1' state to select the clock signal from Input A.

3.5.14.5 FIFO Enable – Bit D7

This bit determines whether or not the Input FIFO is used. If you are only using one input for all channels, or if you are certain that the clock signals in the two inputs are **ABSOLUTELY** identical in frequency and phase, the input FIFO may be bypassed by clearing this bit to the logic '0' state (its default state). To allow for phase differences between the two inputs by using Input FIFO B, set this bit to the logic '1' state.

3.5.14.6 Input Word Length – Bit D8

This bit determines the length of the input data words that Input FIFO B will accept. To program Input FIFO B to accept 12–bit input data, clear this bit to the logic '0' state (its default state). When this bit is set to the logic '1' state, Input FIFO B will accept 16–bit input data.

3.5.14.7 Front Panel Sync Enable – Bit D9

This bit is used to enable the System Sync (SS) input on the front panel digital input connector. This sync signal may be used in lieu of the front panel Sync bus signals. For details on using this bit and the SS signals, refer to Section 3.5.13.7.

3.6 VMEbus A24/A32 Channel Formatter Register Memory

This section describes control and status functions for the sixteen Channel Formatters. One formatter is associated with each of the sixteen receiver channels. These functions are mapped into sixteen identical memory spaces, one for each channel. The maps are applicable to both the broadcast and unique address spaces defined in the A16 registers for those functions (see Sections 3.4.4 and 3.4.5). In the Unique Address space, all functions shown in this section as read/write will support both read and write operations. In the Broadcast Address space, all read functions are disabled, regardless of the accessibility indicated on top of each bit map table. The A24 and A32 base address designations are the same as described for the A24/A32 Global Registers in Section 3.5.

Some of the read–only functions referred to in this section are provided simply because they are available in the state machine design with no additional complexity or circuitry. For example, the Time Stamp Counter appears in each of the sixteen Channel Formatter maps and is redundant with the global register described in Section 3.5.3. Its appearance in each map may provide a more convenient access path for the control software.

Each of the channel formatter maps is identical for all sixteen channels. In Table 3–28, below, 'n' represents the decimal channel number. The hexadecimal digit shown as 'W' in the VMEbus Address column (and in all VME addresses given in this section) also represents the channel number, but in hex rather than decimal. The value of W ranges from 0x0 to 0xF, corresponding to channel numbers 0 through 15, respectively. For 'C31 access to these registers, the register set for each formatter occupies a unique address range indicated in Table 3–28, below, as RY0 – RZ2. Table 3–29, at the top of the next page, gives the 'C31 address range associated with each channel's formatter.

Table 3–28: VMEbus A24/A32 Channel Formatter Register Memory Map

VMEbus Address	'C31 Address	Unique Access	Brdcast Access	Resource Name	Symbolic Address
VME_base+0x0000 1W00	0x30 0RY0	R. O.	N/A	Packet Start Code Register	Packet_Start_Code
VME_base+0x0000 1W04	0x30 0RY1	R. O.	N/A	Ch. n Block Counter Output Register	Ch_n_Block_Counter
VME_base+0x0000 1W08	0x30 0RY2	R. O.	N/A	Time Stamp Counter Output Register	Time_Stamp
VME_base+0x0000 1W0C	0x30 0RY3	R. O.	N/A	Ch. n DDR Complex Output Data Register	Ch_n_DDR_Output
VME_base+0x0000 1W10	0x30 0RY4	R. O.	N/A	Packet Stop Code Register	Packet_Stop_Code
VME_base+0x0000 1W14	0x30 0RY5	—	—	Reserved – Not Used	N/A
VME_base+0x0000 1W18	0x30 0RY6	R. O.	N/A	Channel n Formatter Status Register	Ch_n_Status
VME_base+0x0000 1W1C	0x30 0RY7	—	—	Reserved – Not Used	N/A
VME_base+0x0000 1W20	0x30 0RY8	R/W	W. O.	Channel n Formatter Control Register	Ch_n_Control
VME_base+0x0000 1W24	0x30 0RY9	R/W	W. O.	Channel n ID Tag Register	Ch_n_ID_Tag
VME_base+0x0000 1W28	0x30 0RYA	R/W	W. O.	Ch. n RACEway Packet Size Register	Ch_n_RW_Packet_Size
VME_base+0x0000 1W2C	0x30 0RYB	R/W	W. O.	Channel n Sync Code Word Register	Ch_n_Sync_Code
VME_base+0x0000 1W30 – 1W3C	0x30 0RYC – F	—	—	Reserved – Not Used	N/A
VME_base+0x0000 1W40	0x30 0RZ0	R/W	W. O.	Channel n RACEway Routing Register	Ch_n_RW_Routing
VME_base+0x0000 1W44	0x30 0RZ1	R/W	W. O.	Ch. n RACEway Start Address Register	Ch_n_RW_Start_Address
VME_base+0x0000 1W48	0x30 0RZ2	R/W	W. O.	Ch. n RACEway End Address Register	Ch_n_RW_End_Address

These registers are accessible to the VMEbus when the VMEbus Access Enable bit in the VMEbus Board Control Register is set to logic '1', or to the 'C31 when the VMEbus Access Enable bit is cleared to logic '0'.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

Table 3−29: 'C31 Address Ranges for Channel Formatter Registers			
Channel	**Address Range**	**RY =**	**RZ =**
0	0x30 0400 – 0x30 0412	40	41
1	0x30 0440 – 0x30 0452	44	45
2	0x30 0480 – 0x30 0492	48	49
3	0x30 04C0 – 0x30 04D2	4C	4D
4	0x30 0500 – 0x30 0512	50	51
5	0x30 0540 – 0x30 0552	54	55
6	0x30 0580 – 0x30 0592	58	59
7	0x30 05C0 – 0x30 05D2	5C	5D
8	0x30 0600 – 0x30 0612	60	61
9	0x30 0640 – 0x30 0652	64	65
10	0x30 0680 – 0x30 0692	68	69
11	0x30 06C0 – 0x30 06D2	6C	6D
12	0x30 0700 – 0x30 0712	70	71
13	0x30 0740 – 0x30 0752	74	75
14	0x30 0780 – 0x30 0792	78	79
15	0x30 07C0 – 0x30 07D2	7C	7D

3.6.1 RACEway Data Packet Structure

The function of the Model 6526's Channel Formatters is to assemble RACE−way data packets by adding 32−bit words containing framing information to each RACEway data packet that it creates from the DDR output data, to be sent out over the RACEway backplane fabric. A RACEway packet that con−tains N Complex data samples will, after formatting, consist of N+4 32−bit words. The structure of all 32−bit words (framing, header, and data) in each RACEway packet of size N words is shown in Table 3−30, below.

Table 3−30: RACEway Data Packet Structure		
Word #	**Contents**	**Description**
Word 1	Start Code Word	0xAAAA AAAA
Word 2	Channel ID/ Block Counter Word	Source DDR Board ID (D31 – D28) Source DDR Channel ID (D27 – D24) Destination DSP Board ID (D23 – D20) Destination Processor ID (D19 – D16) Block Count (D15 – D0)
Word 3	Time Stamp Word	0x0000 0000 – 0xFFFF FFFF
Word 4 through Word N−1	Complex DDR Data Samples OR SYNC Words	Real Part of Sample (D31 – D16) Imaginary Part of Sample (D15 – D0) OR SYNC Code (0xFEED C0DE)
Word N	Stop Code Word	0x5555 5555

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.1 RACEway Data Packet Structure (continued)

The data used to assemble the formatted packet is taken from the six Channel Formatter registers that are discussed in Sections 3.6.2 through 3.6.7. Note that, with the exception of the Channel ID Tag Register (which stores the upper 16 bits of Word 2), these registers are located at the five lowest consec–utive addresses in this section.

The START and STOP code words are used to frame the complex data series so that the receiving DSP can distinguish discrete packets.

The CHANNEL ID is comprised of sixteen bits as follows:

- D31 – 28 are written to a register available to the VMEbus. This allows an external processor to assign a unique ID to each of up to sixteen Model 6526 boards.

- D27 – 24 are fixed and unique for each of the sixteen DDR channels on any given Model 6526.

- D23 – 20 designate the destination RACEway processor board to which the packet is sent. These bits are redundant with the bits in the RACEway Routing Register and serve only as an error check to the receiving processor.

- D19 – 16 designate the processor on the destination board to which the data packet should be routed. Receiving RACEway DSP boards should examine this field and route the packet to the appropriate processor.

The BLOCK COUNTER is a counter contained in each channel formatter that is incremented every time a data packet is created using data from that chan–nel's DDR.

The TIME STAMP word is the output of the Time Stamp Counter. This out–put is latched by the frame sync signal of the GC4014 that corresponds to the first of the complex samples that are stored in the packet.

The number of complex samples in each packet is programmable using each channel's RACEway Packet Size Register, with eight different sizes available ranging from 12 to 2044 32–bit words. This number is always four words less than the total RACEway packet size because of the four additional words for the header and trailer described above. See Section 3.6.10 for details about programming the number of samples contained in a RACEway packet.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.2 Packet Start Code Register – R. O. @ Packet_Start_Code

This read–only register stores the fixed packet start code, which is the hexa–decimal value 0xAAAA AAAA. The Model 6526's RACEway state machine places this code word as the first 32–bit word in every RACEway packet that it transmits. The value is supplied here for reference only. Table 3–31, below, shows the register's layout and contents.

Table 3–31: Packet Start Code Register
R. O. @ Packet_Start_Code

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Binary Value	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Hex Value	A				A				A				A			
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Binary Value	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
Hex Value	A				A				A				A			

3.6.3 Channel n ID Tag Registers – R/W @ Ch_n_ID_Tag

These registers store data used to identify the source DDR board and channel, and the destination DSP board and processor. The data transferred from the upper 16 bits of one of these registers will become the upper 16 bits of the sec–ond 32–bit word in each RACEway packet transmitted by the Model 6526. The layout of the register is shown in Table 3–32, below, and each field's function is described in the subsections following the table.

Table 3–32: Channel n ID Tag Registers
R/W @ Ch_n_ID_Tag

Bit #	D31	D30	D29	D28	D27	D26	D25	D24
Field Name	Source DDR Board ID				Source DDR Channel ID			
Function	Packet Source Board ID # (0x0 – 0xF)				Packet Source Channel ID # (0x0 – 0xF)			
Bit #	D23	D22	D21	D20	D19	D18	D17	D16
Field Name	Destination Board ID				Destination Processor ID			
Function	Packet Destination Board ID # (0x0 – 0xF)				Packet Destination Processor ID # (0x0 – 0xF)			

Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Field Name	Reserved – Not Used															
Function	Write with zeros, Mask when reading															

All bits default to the logic '0' state at power up and reset

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.3 Channel n ID Tag Register (continued)

3.6.3.1 Destination Processor ID – Bits D19 to D16

This 4–bit value can be programmed to identify one of sixteen DSP processors that may exist on the DSP processor board that has been identified in the Destination Board ID field (see Section 3.6.3.2, above) as the destination of the RACEway packet. These bits are redundant with the RACEway address and serve as an error check on an incoming packet. These four bits appear in bits D19 to D16 of RACEway Data Packet Word 2 as discussed in Section 3.6.1 and Table 3–30.

3.6.3.2 Destination Board ID – Bits D23 to D20

This 4–bit value can be programmed to identify one of sixteen DSP processor boards that may exist in your system as the destination of the RACEway packet. These bits are redundant with the RACEway routing code and serve as an error check on an incoming packet. These four bits are placed in bits D23 to D20 of RACEway Data Packet Word 2 as discussed in Section 3.6.1 and Table 3–30.

3.6.3.3 Source DDR Channel ID – Bits D27 to D24

This 4–bit value is fixed for each of the 16 DDR channels on the source Model 6526 board defined by the Source DDR Board ID field (see Section 3.6.3.4, above), and should be set to the number of the DDR channel that is the source of the data to be transmitted via the RACEway. These four bits are placed in bits D27 to D24 of RACEway Data Packet Word 2 as discussed in Section 3.6.1 and Table 3–30.

3.6.3.4 Source DDR Board ID – Bits D31 to D28

This 4–bit field should be programmed to identify one of sixteen Model 6526 boards that may be installed in your system as the source of RACEway data. These four bits are placed in bits D31 to D28 of RACEway Data Packet Word 2 as discussed in Section 3.6.1 and Table 3–30.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.4 Channel n Block Counter Output Register – R. O. @ Ch_n_Block_Counter

The 16–bit numbers read from these registers represent the current value of the channel block counters. The block counter in a given channel is incre–mented each time data from that channel is used to create a RACEway packet. The format is binary with BC15 as the most significant bit. The contents of a block counter register is placed in the lower 16 bits of the second 32–bit word of every RACEway packet transmitted by the 6526. Table 3–33, below, shows the layout of this register.

Table 3–33: Channel n Block Counter Output Register
R. O. @ Ch_n_Block_Counter

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	Reserved – Not Used															
Function	Mask when reading															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	BC15	BC14	BC13	BC12	BC11	BC10	BC9	BC8	BC7	BC6	BC5	BC4	BC3	BC2	BC1	BC0
Function	Current output value from Channel n Block Counter															

3.6.5 Time Stamp Counter Output Register – R. O. @ Time_Stamp

This read–only register contains a 32–bit binary word showing the current value of the Model 6526's Time Stamp Counter. The format is binary and TS31 is the most significant bit. This value is placed in the third 32–bit word of every RACEway packet transmitted by the 6526. This is the same value shown in the Time Stamp Counter global register (see Section 3.5.3). The lay–out of this register is shown in Table 3–34, below.

Table 3–34: Time Stamp Counter Output Register
R. O. @ Time_Stamp

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	TS31	TS30	TS29	TS28	TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16
Function	16 Most Significant Bits of Time Stamp Counter Output															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Function	16 Least Significant Bits of Time Stamp Counter Output															

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.6 Channel n DDR Complex Output Data Register – R. O. @ Ch_n_DDR_Output

These registers store the complex output samples from the GC4014 DDRs. Data read from these registers occupy the fourth through the next–to–last 32–bit word positions in every RACEway packet transmitted by the Model 6526. The In–Phase (or real) data is stored in the upper 16 bits, and the Quadrature (or imaginary) data is stored in the lower 16 bits. The layout of these registers is shown in Table 3–35, below, and the subsections following the table describe the two values contained in these bits.

Table 3–35: Channel n DDR Complex Data Output Register
R. O. @ Ch_n_DDR_Output

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	DI15	DI14	DI13	DI12	DI11	DI10	DI9	DI8	DI7	DI6	DI5	DI4	DI3	DI2	DI1	DI0
Function	16–bit In–Phase (Real) Component of DDR Output Sample															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	DQ15	DQ14	DQ13	DQ12	DQ11	DQ10	DQ9	DQ8	DQ7	DQ6	DQ5	DQ4	DQ3	DQ2	DQ1	DQ0
Function	16–bit Quadrature (Imaginary) Component of DDR Output Sample															

3.6.6.1 DDR Output Data: In–Phase (Real) Component – Bits D31 to D16 (DI15 to DI0)

The upper 16 bits of these registers store a 16–bit binary word representing the in–phase component of the data output of the GC4014 DDR. The format is 2's complement binary and DI15 is the most significant bit.

3.6.6.2 DDR Output Data: Quadrature (Imaginary) Component – Bits D15 to D0 (DQ15 to DQ0)

The lower 16 bits of these registers store a 16–bit binary word representing the quadrature component of the data output of the GC4014 DDR. The format is 2's complement binary and DQ15 is the most significant bit.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.7 Packet Stop Code Register – R. O. @ Packet_Stop_Code

This read–only register stores the fixed packet stop code, which is the hexa–decimal value 0x5555 5555. The Model 6526's RACEway state machine places this code word as the last 32–bit word in every RACEway packet that it trans–mits. The value is supplied here for reference only. Table 3–36, below, shows the register's layout and contents.

Table 3–36: Packet Stop Code Register
R. O. @ Packet_Stop_Code

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Binary Value	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Hex Value	5				5				5				5			
Bit #	**D15**	**D14**	**D13**	**D12**	**D11**	**D10**	**D9**	**D8**	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Binary Value	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
Hex Value	5				5				5				5			

3.6.8 Channel n Formatter Status Register – R. O. @ Ch_n_Status

These Read–Only registers have four active bits, used to indicate the states of various operating conditions within the formatter. Table 3–37, below, identi–fies the bits contained in these registers, and the subsections following the table describe these bits.

Table 3–37: Channel n Formatter Status Register
R. O. @ Ch_n_Status

Bit #	D31 – D8							
Bit Name	Reserved – Not Used							
Function	Mask when Reading							
Bit #	**D7**	**D6**	**D5**	**D4**	**D3**	**D2**	**D1**	**D0**
Bit Name	Reserved – Not Used			Blk_Cnt_Sync _Reset_Armed	Sync_Wrd_ Sub_Armed	Pckt_Send _Req	FIFO_Full	Not Used
Function	Mask when Reading			1 = True 0 = False				Mask when Reading
All bits default to the logic '0' state at power up and reset								

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.8 Channel n Formatter Status Register (continued)

3.6.8.1 RACEway FIFO Full – Bit D1

When this bit reads back in the logic '1' state, the RACEway Output FIFO is full and the channel formatter cannot write any more data to it. This condition can be used to generate an interrupt to the on–board 'C31 DSP by setting the FIFO Full bit (D6) in the 'C31's Interrupt Mask Register (see Section 3.8.9) to the logic '1' state. If this bit reads back in the logic '0' state, then the RACEway FIFO is able to accept more data.

3.6.8.2 Packet Send Request – Bit D2

The formatter sets this bit to the logic '1' state to indicate that it has finished writing a data packet to its Output FIFO and is ready for the RACEway controller to send the packet. If this bit reads back in the logic '0' state, then any data that may exist in the RACEway FIFO is NOT a complete packet and is NOT ready to be sent over the RACEway.

3.6.8.3 Sync Word Substitution Armed – Bit D3

When this bit reads back in the logic '1' state, it indicates that the formatter has received the command to substitute a Sync word for the next two DDR output data words when the next Sync pulse occurs (i. e., that the Sync Word Substitution Enable bit (D0) in the channel's Formatter Control Register (see Section 3.6.9.1) has been set to the logic '1' state). This bit can be useful in synchronizing software commands with Sync events. The Sync word defaults to 0xFEED C0DE, but can be any 32–bit value written to the Sync Code Word Register (see Section 3.6.11). If this bit reads back in its default logic '0' state, then the next Sync pulse will not cause the channel to output Sync Words.

3.6.8.4 Block Counter Synchronous Reset Armed – Bit D4

When this bit reads back in the logic '1' state, it indicates that the channel formatter has received a command to reset the block counter when the next Sync pulse occurs (i. e., that the Block Counter Synchronous Reset Enable bit (D1) in the Channel's Formatter Control Register (see Section 3.6.9.2) has been set to the logic '1' state). This is useful in synchronizing software commands with Sync events. If this bit reads back in its default logic '0' state, then the block counter will be unaffected by the next Sync pulse.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.9 Channel n Formatter Control Register – R/W @ Ch_n_Control

Each DDR channel's Formatter Control Register contains 17 active bits. A 4–bit field in this register determines the number of packets per trigger in triggered mode, and an 8–bit field determines the delay from the trigger to the first output sample. Individual bits are provided to enable the formatter, packet output, synchronous reset of the block counter, and substitution of a Sync Word for an Output Sample, and to select triggered or free running operation. Table 3–38, below, shows the register's bit layout, and the following subsections describe these bits and fields.

Table 3–38: Channel n Formatter Control Register
R/W @ Ch_n_Control

Bit #	D31 – D17	D16	D15	D14	D13
Bit Name	Reserved – Not Used	Trgd_ Pkts_3	Trgd_ Pkts_2	Trgd_ Pkts_1	Trgd_ Pkts_0
Function	Write with Zeros, Mask when Reading	Number of Packets per Trigger (Triggered Mode only)			

Bit #	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Pkt_Strt_ Delay_7	Pkt_Strt_ Delay_6	Pkt_Strt_ Delay_5	Pkt_Strt_ Delay_4	Pkt_Strt_ Delay_3	Pkt_Strt_ Delay_2	Pkt_Strt_ Delay_1	Pkt_Strt_ Delay_0	Mode_ Select	Chn_ Format_ Enable	Pckt_ Output_ Enable	Blk_Cnt_ Reset_ Enable	Sync_ Wrd_Sub _Enable
Function	Number of Output Sample Periods of delay from Trigger to first Output Sample								1=Free_r 0=Trigger	1=Enable 0=Disable			

All bits default to the logic '0' state at power up and reset

3.6.9.1 Sync Word Substitution Enable – Bit D0

When this bit is set to the logic '1' state, the Sync Word substitution feature is enabled. This feature causes the next occurrence of the Sync signal associated with the channel in question to replace two consecutive DDR output samples with the data contained in the channel's SyncWord Code Register (see Section 3.6.11). Setting this bit will also set the Sync Word Substitution Armed bit (D3) in the associated channel's Formatter Status Register to the logic '1' state (see Section 3.6.8.3).

When this bit is cleared to its default logic '0' state, Sync Word substitution is disabled and a Sync signal will not replace DDR output data samples. Changes in value of this bit are only recognized at the beginning of the next packet.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.9 Channel n Formatter Control Register (continued)

3.6.9.2 Block Counter Synchronous Reset Enable – Bit D1

When this bit is set to the logic '1' state, the synchronous reset feature is enabled. This feature causes the next occurrence of the Sync signal associated with the channel in question to reset that channel's block counter. Setting this bit will also set the Block Counter Synchronous Reset Armed bit (D4) in the associated channel's Formatter Status Register to the logic '1' state (see Section 3.6.8.4).

When this bit is cleared to its default logic '0' state, synchronous reset is disabled and a Sync signal will not reset the channel's block counter. Changes in value of this bit are only recognized at the beginning of the next packet.

3.6.9.3 Packet Output Enable – Bit D2

When this bit is set to the logic '1' state, packet output is enabled and the formatter outputs packet data normally. When this bit is cleared to its default logic '0' state, packet output is disabled and the formatter does not output packets to RACEway.

When packet output is disabled, however, (assuming that the channel formatter is enabled, see the NOTE at the end of Section 3.6.9.4, on the next page) the block counter continues to count packets and packets continue to be formed as if they were being sent to RACEway. In this way, when the state of this bit is changed (i. e., when packet output is enabled after being disabled), the first transmitted packet has the same data content as if packet output had not been disabled. A change in value of this bit is only recognized at the beginning of the next packet.

NOTE: For this bit to operate as specified above, the Channel Formatter Enable bit (D3 in this register, see Section 3.6.9.4, above) **MUST** be set to the logic '1' state to enable packet creation.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.9 Channel n Formatter Control Register (continued)

3.6.9.4 Channel Formatter Enable – Bit D3

This bit **MUST** be set to the logic '1' state to enable channel formatter state machine to begin forming RACEway packets. When this bit is cleared to its default logic '0' state, no RACEway packets are formed, and any output samples from the GC4014 are ignored and lost.

NOTE: This bit **MUST** be set to the logic '1' state, to enable packet creation, in order for the Packet Output Enable bit (see Section 3.6.9.3, below) to operate properly.

3.6.9.5 Mode Selection – Bit D4

This bit determines whether the DDR channel operates in triggered or free–run mode. When this bit is cleared to the logic '0' state (its default), the channel operates in free–run mode and delivers packets to the RACEway continuously, with a single start–up delay as determined by the contents of the Packet Start Delay field in this register (D12 to D5, see Section 3.6.9.6, above).

When this bit is set to the logic '1' state, the DDR channel operates in triggered mode. In this mode, no output samples are delivered until the channel receives a Sync signal. Upon receipt of the Sync, and after a programmed delay period (see Section 3.6.9.6, above), the channel will assemble and deliver the number of RACEway packets set into this register's Packets per Trigger field (see Section 3.6.9.7), then wait for the next Sync before starting the process again. Changes in the state of this bit do not take effect until the Model 6526 has finished assembling the packet in progress.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.9 Channel n Formatter Control Register (continued)

3.6.9.6 Packet Start Delay – D12 to D5

This 8–bit field determines the number of output sample periods of delay the channel formatter waits through following a Sync signal, before it starts packing output samples into the first RACEway packet. This delay applies only to the first packet within each triggered packet group if the channel is set for triggered mode (if D4 in this register is cleared to the logic '0' state) or to the first packet that would occur after the channel has been programmed for free–run mode (when bit D4 in this register is set to the logic '1' state).

The range of values that this field can accept is from 0 to 255 (decimal), with D5 as the least significant bit. After the Sync signal is received, the delay counter begins counting the number of GC4014 output samples. When the number of sample periods specified in this bit field has been reached, the next sample is taken as the first data sample in the first packet being formed. If the delay is set to zero (if all bits in this field are cleared to logic '0', which is their default state), then there is no delay between the Sync signal and the first data sample in the first packet.

Note that the delay range provided by this field is long enough to allow the GC4014 to purge its internal filter memory after a transient event such as switching inputs. In this way, the same Sync signal can be used for GC4014 control and for triggering a delayed RACEway packet, while still allowing the GC4014 to settle.

3.6.9.7 Number of Packets per Trigger – Bits D16 to D13

The contents of this 4–bit field are only used when the DDR channel operates in triggered mode (when bit D4 in this register is cleared to the logic '0' state). In triggered mode, these four bits specify the number consecutive RACEway packets that are delivered in response to each trigger. The range of values for this field is from 0 to 15 (decimal), and D13 is the least significant bit. A value of 0 is allowed in this field (its default state), but will result in zero packets being sent over the RACEway for each trigger.

When a channel is programmed for free–run mode (when bit D4 in this register is set to the logic '1' state), the content of this field is not used and is therefore not relevant.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.10 Channel n RACEway Packet Size Register – R/W @ Ch_n_RW_Packet_Size

The size of the RACEway packets that are delivered by a given channel are determined by the contents of the three least significant bits of this register. Eight different packet sizes are available. (Remember that each packet contains four 32−bit words in addition to the block of complex data samples, three for the header and one for the trailer — see Section 3.6.1 for a complete description of the RACEway packet structure.) The number of complex data samples in each RACEway packet ranges from 12 to 2044 32−bit words.

Table 3−39, below, illustrates the contents of this register, and Table 3−40, at the bottom of this page, shows the number of words, samples, and bytes in each RACEway packet for a given setting. Note that writing to this register affects each channel individually, supporting different packet sizes for each channel, and that 2048 bytes is the largest packet size recommended by the RACEway specification.

Table 3−39: Channel n RACEway Packet Size Register
R/W @ Ch_n_RW_Packet Size

Bit #	D31 – D3	D2	D1	D0
Bit Name	Reserved − Not Used	Pkt_Size_2	Pkt_Size_1	Pkt_Size_0
Function	Write with 0's, Mask when Reading	See Table 3−40, below		
All bits default to the logic '0' state at power up and reset				

Table 3−40: Channel n RACEway Packet Size Register − Packet Size Settings

D2	D1	D0	Decimal Value	Total Packet Size (32-bit words)	Total Packet Size (8-bit bytes)	Complex DDR Samples (32-bit words)
0	0	0	**0**	16	64	12
0	0	1	**1**	32	128	28
0	1	0	**2**	64	256	60
0	1	1	**3**	128	512	124
1	0	0	**4**	256	1024	252
1	0	1	**5**	512	2048	508
1	1	0	**6**	1024	4096	1020
1	1	1	**7**	2048	8192	2044

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.11 Channel n Sync Code Word Register – R/W @ Ch_n_Sync_Code

This register stores a Sync code word for each channel. This value will replace two consecutive data samples after the occurrence of a Sync in the channel if the Sync Word Substitution Enable bit (D0) in the Channel Formatter's Control Register (see Section 3.6.9.1) is set to the logic '1' state. These registers can be programmed to specify a unique Sync code word for each individual channel. By default, these registers' bits are initialized with the hexadecimal value 0xFEED C0DE.

Table 3−41: Channel n Sync Code Word Register
R/W @ Ch_n_Sync_Code

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Binary Value	1	1	1	1	1	1	1	0	1	1	1	0	1	1	0	1
Hex Value	F				E				E				D			
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Binary Value	1	1	0	0	0	0	0	0	1	1	0	1	1	1	1	0
Hex Value	C				0				D				E			

Bit values shown above are the power−up / reset defaults.
These registers may be re−written such that each DDR channel has its own unique Sync code.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.12 Channel n RACEway Routing Register – R/W @ Ch_n_RW_Routing

This register stores the route word, which is the first longword transmitted during the RACEway data cycle, prior to transmission of the actual data packet. When the Model 6526 acts as a RACEway master, the appropriate route word must be written to this register (in addition to writing the Address register and assembling the data packet(s) to be transmitted) before beginning the data cycle. The route word contains up to nine 3–bit route fields, two 2–bit fields (one holds a broadcast accept code and the other a routing priority code), and a broadcast/single mode select bit. Table 3–42, below, shows the bit layout in the register. The following subsections describe these register's fields and bits.

Table 3–42: Channel n RACEway Routing Register
R/W @ Ch_n_Routing

Bit #s	D31–D29	D28–D26	D25–D23	D22–D20	D19–D17	D16–D14	D13–D11	D10–D8	D7–D5	D4 – D3	D2 – D1	D0
Bit Name	Route_0	Route_1	Route_2	Route_3	Route_4	Route_5	Route_6	Route_7	Route_8	Broadcast _Accept _Code	Routing _Priority _Code	Mode_ Select
Function	May contain up to nine 3–bit route codes, and/or up to six address bits – See Table 3–44 and Section 3.6.12.4, next page									See Sec. 3.6.12.3	See Sec. 3.6.12.2	1=Broadcast 0 = Single
All bits default to the logic '0' state at power up and reset												

3.6.12.1 Mode Selection – Bit D0

When the Model 6526 acts as a RACEway Master, set this bit to the logic '1' state for RACEway broadcast operations. For single port routing, clear this bit to the logic '0' state.

3.6.12.2 Routing Priority Code – Bits D2 to D1

Arbitration for RACEway resources is distributed rather than centralized. This means that each crossbar switch involved in a given transaction must be arbitrated for at the time that the message arrives at the crossbar. The primary method of resolving RACEway resource contention is by means of transaction priority. The Route word, being the first word transmitted by the master in any given transaction, contains a two–bit priority field. If a transaction with a higher priority arrives at a crossbar that is in use for a lower–priority transaction, the crossbar sends a Kill request to the master of the lower–priority transaction. The master of the lower–priority transaction must then complete its current data cycle and suspend the transaction. When the higher–priority transaction has completed, the master of the lower–priority transaction may resume at the address at which the transaction was suspended.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.12 Channel n RACEway Routing Register (continued)

3.6.12.2 Routing Priority Code (continued)

Table 3–43, below, lists the Routing Priority codes and the priority levels associated with those codes.

Table 3–43: RACEway Routing Register – Priority Codes

Routing Priority Field		Priority Level
D2	D1	
0	0	0 (Lowest)
0	1	1
1	0	2 (Highest)
1	1	Reserved

This function is only active when the Mode Select bit (D0) is set to the logic '1' state, to enable broadcast mode. In single mode, these two bits must be cleared to the logic '0' state.

When the Model 6526 acts as a RACEway master, write the route priority code appropriate to the upcoming transaction into this bit field.

For transactions with equal route priorities, the next contention settling method is called port priority. The contended resource will be granted to the transaction entering the switch at the highest priority port, where port F has the highest priority and port A has the lowest. Finally, for transactions arriving during the same cycle, at the same port, with the same route priority, a round–robin arbitration scheme is used.

3.6.12.3 Broadcast Accept Code – Bits D4 to D3

Some RACEway Slave devices are equipped with circuitry that allows them to accept or reject individual broadcast messages, based on a broadcast key code. Such Slave devices compare the broadcast accept code in the Route word with a broadcast key code in an internal register, and accept only those broadcasts with accept codes that match their key codes. When the 6526 Masters a RACEway broadcast involving this type of Slave device, write the accept code into this bit field.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.12 Channel n RACEway Routing Register (continued)

3.6.12.4 RACEway Route Fields – Nine 3–bit fields, bits D31 to D5

The route codes are used to specify which crossbar port will be used as the exit for the ensuing data transaction. The RACEway master writes one 3–bit exit code for each crossbar that will handle this data stream. The exit code for the first crossbar is written into the Route_0 field, the second code goes in the Route_1 field, and so on. Table 3–44, below, defines the exit codes.

Table 3–44: RACEway Routing Register – Route Field Exit Codes

Code		Single Mode Exit Port	Broadcast Mode Exit Ports
Hex	Binary		
7	1,1,1	A	B, C & D if entered at A, else A
6	1,1,0	B	A, C & D if entered at B, else B
5	1,0,1	C	A, B & D if entered at C, else C
4	1,0,0	D	A, B & C if entered at D, else D
3	0,1,1	E	E
2	0,1,0	F	F
1	0,0,1	E 1st, adaptive route	A, B, C, D & E
0	0,0,0	F 1st, adaptive route	A, B, C, D & F

The format of a RACEway transmission is such that the route word is the first word transmitted. When the transmitted data arrives at the first crossbar in the data path, the crossbar reads the three most significant bits of the route word, to determine which exit port this data packet should be directed to. Before passing the data onward, the crossbar shifts the data in this 27–bit field three bits to the left, thus removing the field it read and moving the next route code into the Route0 field (i. e., into the three most significant bits). All subsequent crossbars involved in the data path perform the same operation, such that when the data stream exits the last crossbar, all routing information has been removed. Note that this field may also contain up to six high–order address bits, which can be used in the Slave device. These bits are written immediately to the right of the last route code. In this manner, when all route codes have been shifted out, those address bits will occupy the most signifi–cant bit positions in this register.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.13 Channel n RACEway Start Address Register – R/W @Ch_n_RW_Start_Address

This register stores the address word, which will become the second long–word transmitted during the RACEway data cycle (the route word is transmitted first). When the Model 6526 acts as a RACEway master, the appropriate address word must be written to this register (in addition to writing the Route register and assembling the data packet(s) to be transmitted) before beginning the data cycle. The address word contains a 4–bit width/alignment field, a 25–bit address field, a read flag, and a lock flag. Table 3–45, below, shows the bit layout in the registers. The following subsections describe the registers' fields.

Table 3–45: Channel n RACEway Start Address Register
R/W @ Ch_n_RW_Start_Address

Bit #s	D31 – D28	D27 – D3	D2	D1	D0
Bit Name	Width/Alignment_ Field	Start_Address_Field (SA24 – SA0)	Not Used	Read Flag	Lock Flag
Function	Binary '1011' – See Section 3.6.13.4	RACEway Start Address – See Section 3.6.13.3	Write '0', Mask Read	1 = Read 0 = Write	1 = Unlock 0 = Lock
All bits default to the logic '0' state at power up and reset					

3.6.13.1 Lock Flag – Bit D0

The lock flag must be cleared to the logic '0' state by the RACEway master for Read–Modify–Write and similar operations that require exclusive, uninterrupted access to a RACEway data route. If a crossbar attempts to suspend a locked operation, the master may ignore the Kill request until the transaction is complete. This bit should be set to the logic '1' state, to unlock the route, when the locked transaction has been completed, or at any time when exclusive access to a route is not required.

3.6.13.2 Read Flag – Bit D1

The read flag bit must be set to the logic '1' state by the RACEway master for Read, Read–Modify–Write, and Split Read operations. For Write and Broadcast transactions over the RACEway, the master must clear the read flag bit to the logic '0' state.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.13 Channel n RACEway Start Address Register (continued)

3.6.13.3 Start Address Field – Bits D27 to D3

When a RACEway packet arrives at a destination board, the packet can be directed to a specific starting address in local memory on that board using this starting address field. RACEway supports up to 31 address bits. The most significant six bits are derived from the contents of the Channel's Routing Register, bits D31 through D26, after the Routing words have been shifted left and out of this region by the crossbar switches (see Section 3.6.12.4). The 25 least significant bits of the 31–bit packet destination address is specified by this 25–bit Start Address field.

The utilization of the address by the destination RACEway board depends on the board. For example, in the Pentek Model 4290 with the Model 6220 RACEway interface, only four bits (SA24 to SA21, corresponding to D27 to D24 in this register) of the RACE–way Start Address are used to steer the incoming packets to the four 'C6x DSPs on the board. Each of these four bits can be inde–pendently set to enable the packet to flow into the mezzanine Bi–FIFO of each processor. For write operations (typical for the Model 6526), multiple bits may be set to support broadcasting packets to all DSPs on the DSP board.

The Model 6526 supports two modes of addressing: incrementing mode, for writing to memory devices, and non–incrementing mode, for writing to FIFOs or register–based devices appearing at fixed addresses. This is described further in Section 3.6.14.2.

3.6.13.4 Width/Alignment Field – Bits D31 to D28

The four most significant bits of the RACEway Address Registers are used to inform the crossbar switches involved in the transac–tions about the width (in bytes) and alignment of the data that will subsequently be transmitted.

Although this 4–bit field can be used to specify different width/byte alignment combinations, the only method to transfer data from the Model 6526 is using 8–byte transfers to send two I/Q pairs in each cycle (physically, this is accomplished by sending two 32–bit I/Q pairs in two consecutive cycles, which is transparent to the user). The code for this is binary '1011', in bits D31 to D28. See the RACEway Interlink Standard for more details.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.14 Channel n RACEway End Address Register – R/W @ Ch_n_RW_End_Address

This register contains a single bit used to select the address modes of RACE–way transfers, and a 15–bit field that specifies the ending address for transfer blocks consisting of one or more packets. Table 3–46, below, shows this reg–ister's bit layout, and the subsections following the table describe how these bits are used.

Table 3–46: Channel n RACEway End Address Register
R/W @ Ch_n_RW_End Address

Bit #	D31 – D16	
Bit Name	Reserved – Not Used	
Function	Write with Zeros, Mask when Reading	
Bit #	**D15**	**D14 – D0**
Bit Name	RACEway_ Address_Mode	End_Address_Field (EA14 – EA0)
Function	1 = Incrementing 0 = Non–Incrementing	RACEway End Address – See Section 3.6.14.1
All bits default to the logic '0' state at power up and reset		

3.6.14.1 End Address Field – Bits D14 to D0

If a long, contiguous block of data requires multiple packets to complete, each successive packet will automatically be formed with the correct RACEway address for the first data word in each packet. In order to let the channel formatter correctly handle these multiple packet transfers, whenever the RACEway Address Mode bit (see Section 3.6.14.2) is set to the logic '1' state (to select the incrementing address mode), a RACEway End Address must also be supplied. This allows the channel formatter to automatically reset the hardware address counter to its initial value (the RACE–way Start Address, see Section 3.6.13.3) at the end of the multiple packet transfer block, so it is ready for a new block.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.14 Channel n RACEway End Address Register (continued)

3.6.14.1 End Address Field (continued)

The RACEway End Address (EA) is a 15−bit binary integer that is computed as follows:

EA14 to EA0 = [PacketSize x # Packets] / 2 + [SA14 to SA0] – 1

where:

PacketSize = number of 32−bit words in each RACEway packet (see Section 3.6.10)

Packets = number of packets in the multiple packet block

SA14 to SA0 = Least significant 15 bits of RACEway Start Address (see Section 3.6.13.3)

This supports multiple packet block lengths of up to 256 kBytes. Note that blocks cannot cross page boundaries of 256 kBytes. The full 31−bit RACEway address is given by the six most significant bits left in the Routing Word (see Section 3.6.12) followed by bits SA24 to SA0. Page boundaries of 256 kBytes occur when address bit SA15 changes.

NOTE: If a block transfer is stopped midstream, and is not allowed to resume to completion, the hardware address counter must be re−initialized before starting a new block. This is accomplished by re−writing the correct value to the RACEway Start Address Register (see Section 3.6.13). In general, it is always safer to re−write this value whenever starting a new block, even though the channel formatter normally takes care of re−initializing the counter at the end of each block.

3.6 VMEbus A24/A32 Channel Formatter Register Memory (continued)

3.6.14 Channel n RACEway End Address Register (continued)

3.6.14.2 RACEway Address Mode – Bit D15

This bit selects between two types of RACEway addressing:

- Incrementing mode, which is used for sending packets to a memory array where sequential addresses are required for each successive transfer.

- Non–incrementing mode, which is used for sending packets to a fixed, constant address (such as a FIFO).

The non–incrementing address mode is selected by clearing this bit to the logic '0' state (which is its default). In this mode, the channel formatter always sends packets with a constant, fixed RACEway address, as determined by the contents of the RACE–way Start Address Register (see Section 3.6.13). The RACEway End Address bits are meaningless in this mode.

The incrementing address mode is selected by setting this bit to the logic '1' state. This mode supports both single packet transfers and transfers requiring multiple packets. The RACEway address is maintained in a hardware counter whose initial value is created by using the RACEway Start Address (see Section 3.6.13.3). For each pair of 32–bit words transmitted over RACEway, this hardware counter advances by one count, consistent with the minimum RACEway transfer cycle of 64 bits (two 32–bit words). As each RACEway packet is formed, the value in the hardware address counter is used to create the RACEway address word in the packet header. Packet transfers continue in this manner until the value in the counter is equal to the value in the RACEway End Address field (see Section 3.6.14.1), or the transaction is “killed”.

In the incrementing mode, if a packet transfer is interrupted (“killed”) by a higher–priority packet, after the higher priority packet is sent, the remainder of the killed packet will resume with a new packet header reflecting the current RACEway address of the first word in the packet. This will tell the receiving device exactly where to write the remaining data.

3.7 Dual Port Memory and Command Tables

The Dual Port Memory (DPSRAM) is used for passing parameters, commands, and status between the VMEbus and the 'C31 processor on the Model 6526, and for supporting Pentek's SwiftNet communications software.

DPSRAM is arranged as 16k x 32 and is mapped into the address space of the 'C31 and into A24/A32 VMEbus address space as shown in Table 3–47 below. The conventions for VMEbus base addressing are detailed in Section 3.5.

Table 3–47: Dual Port SRAM Usage Memory Map

VMEbus Address Range	'C31 Address Range	Unique Access	Broadcast Access	Resource / Use	Symbolic Address
VME_base+0x0001 0000 – VME_base+0x0001 7FFF	0x20 0000 – 0x20 1FFF	R/W	W.O.	SwiftNet Space	SwiftNet_Space
VME_base+0x0001 8000 – VME_base+0x0001 FFF4	0x20 2000 – 0x20 3FFD	R/W	W.O.	Command Table Space	Cmd_Table_Space
VME_base+0x0001 FFF8 – VME_base+0x0001 FFFB	0x20 3FFE	VME: R/W 'C31: R.O.	VME: W.O. 'C31: N/A	'C31 Inbound Mailbox (causes 'C31 interrupt)	'C31_MBox_In
VME_base+0x0001 FFFC – VME_base+0x0001 FFFF	0x20 3FFF	VME: R.O. 'C31: R/W	VME: N/A 'C31: W.O.	'C31 Outbound Mailbox (no interrupt)	'C31_MBox_Out

The Dual Port SRAM is divided into two regions (each 8k x 32, or 32 kBytes), one for Command Tables and the other for SwiftNet. The VMEbus addresses memory as bytes, but the 'C31 always addresses memory as 32–bit words. This explains the factor–of–four difference in the address range between the two access regions.

These resources also appear in a second 'C31 address space, but only when the VMEbus Access Enable bit in the Board Control Register (Section 3.4.1.8) is cleared to the logic '0' state. The alternate address range is from 0x30 4000 to 0x30 7FFF. This may be useful for diagnostic/testing purposes.

The lower half of the DPSRAM is reserved for use by Pentek's SwiftNet DSP Networking and Communications Software. This memory region is used to pass commands and data tables between the 'C31 and the workstation used for code development and debugging. SwiftNet supports 'C31 code development software tools which execute on Sun, HP, Digital, and PC platforms.

The last two words of the Command Table space in DPSRAM are reserved for two 32–bit Mailbox Registers, useful in passing control parameters and generating 'C31 interrupts. The 'C31 Inbound Mailbox has read/write access from the VMEbus and read–only access from the 'C31. A write to this location generates a maskable interrupt on 'C31 interrupt INT0 as described in Sections 3.8.9 and 3.8.10. The 'C31 Outbound Mailbox has read/write access from the 'C31 and read–only access from the VMEbus. The 'C31 Outbound Mailbox does not generate any interrupts. Instead, the 'C31 uses the VINT bit in the 'C31 Control Register to generate VMEbus interrupts (see Section 3.8.5).

3.7 Dual Port Memory and Command Tables (continued)

The Command Table Space of the Dual Port SRAM is partitioned into blocks called Command Tables, which are used for staging a series of commands or parameters for a specified group of channels. The structure of a Command Table is relatively simple.

The first four words of each Command Table are the Command Table Header Words.

- The first header word is a control word specifying the number of commands following the four header words, plus control bits to control loading and access.
- The second header word specifies the channels affected by this Command Table.
- The third header word is the Timestamp, used for scheduling the execution of the commands within this Command Table.
- The fourth header word is a command status word, which reports any errors in command processing.

Beginning with the fifth word and continuing to the end of the Command Table are commands and parameters to be executed. Refer to the ReadyFlow board support software documentation for the Model 6526 (Pentek part #801.65260) for a description of the operations and commands for command table processing.

The default size of each Command Table is sixty−four 32−bit words, which means that the default number of Command Tables in the Command Table Space is 128. Other table dimensions and numbers of tables are possible by changing the firmware used by the 'C31 in processing the contents of the DPSRAM Command Table Space.

In processing the Command Tables, the 'C31 scans Command Table Header Word 1, to see if that table needs processing. If so, it then uses other bits in Header Word 1, to see how many commands are stored in the table. Header Word 2 then tells the 'C31 which channels the parameters apply to. This allows the 'C31 to quickly scan and skip any tables which do not need processing.

When a Command Table is not being processed by the 'C31, it can be updated for sub−sequent operation by the VMEbus controller. In normal operation, when a new Com−mand Table has been formed, it can be "enabled" by a control bit in the header so that the 'C31 processes it during the next scan. Command Tables for common functions or modes can be kept intact in Dual Port SRAM and enabled or disabled at the appropriate times by writing the header control bit.

3.7 Dual Port Memory and Command Tables (continued)

The remainder of this section consists of subsections that describe the structure and contents of the four Command Table Header Words. These words are accessible from both the VMEbus and the 'C31 processor, as shown in Table 3–47, on page 91. As indicated in that table, the base address for VMEbus access to the Command Tables is VMEbus_base+0x0001 8000. The base address for 'C31 access to these tables is 0x20 2000, and the 'C31 address offset of each header word is equal to the VME address offset divided by four (see Section 3.8 for further information on the 'C31 memory maps and registers).

Table 3–48, below, shows the Memory Map for the Command Table Header Words of the first Command Table. Note that the starting address for each Command Table is provided in the following table. These starting addresses assume that the default table size is sixty–four 32–bit words (256 bytes).

Table 3–48: Command Table Header Words Memory Map

Resource Name	VMEbus Address*	'C31 Address*	Symbolic Address
Command Table Header Word #1	VME_base+0x0001 8000	0x20 2000	Command_Control
Command Table Header Word #2	VME_base+0x0001 8004	0x20 2001	Channel_Mask
Command Table Header Word #3	VME_base+0x0001 8008	0x20 2002	Timestamp
Command Table Header Word #4	VME_base+0x0001 800C	0x20 2003	Command_Status
Command Table commands & parameters	VME_base+0x0001 8010 – VME_base+0x0001 80FF	0x20 2004 – 0x20 203F	–

* Addresses shown are for the first Command Table in the Command Table Space; see Table 3–49, below, for the starting address of each of the 128 command tables.

These resources are accessible to the VMEbus when the VMEbus Access Enable bit in the VMEbus Board Control Register is set to logic '1', or to the 'C31 when the VMEbus Access Enable bit is cleared to logic '0'.

Table 3–49: Command Table Start Addresses

Command Table*	VMEbus Start Address	'C31 Start Address
First (1st) Command Table	VME_base+0x0001 8000	0x20 2000
2nd Command Table	VME_base+0x0001 8100	0x20 2040
3rd Command Table	VME_base+0x0001 8200	0x20 2080
4th Command Table	VME_base+0x0001 8300	0x20 20C0
5th Command Table	VME_base+0x0001 8400	0x20 2100
. . .	. . .	. . .
127th Command Table	VME_base+0x0001 FE00	0x20 3F80
Last (128th) Command Table	VME_base+0x0001 FF00	0x20 3FC0

* Assumes that the size of each Command Table is at the default of sixty–four 32–bit words (256 bytes), and there are, thus, 128 tables

3.7 Dual Port Memory and Command Tables (continued)

3.7.1 Command Table Header Word #1: Command Control – R/W @ Command_Control

The first header word in the command table is a control word specifying the number of commands following the four header words, and four control bits to control loading and access. Table 3–50, below, shows this register's bit lay–out, and the subsections following the table describe these bits.

Table 3–50: Command Table Header Word #1: Command Control
R/W @ Command_Control

Bit #	D31 – D18		D17	D16
Bit Name	Reserved – Not Used		Format_1	Format_0
Function	Write with Zeros – Mask when Reading		See Table 3–51	
Bit #	**D15 – D6**	**D5 – D2**	**D1**	**D0***
Bit Name	Table Length (TL9 – TL0)	Reserved – Not Used	Schedule _Command	Single _Load
Function	Number of commands in command table (excluding header)	Write with Zeros, Mask when Reading	1 = Schedule 0 = Immediate	1 = Process 0 = Done

* This bit can ONLY be set by the VMEbus, and can ONLY be cleared by the 'C31
All bits default to the logic '0' state at power up and reset

3.7.1.1 Single Load – Bit D0

This bit is set to logic '1' by the VMEbus to signal to the 'C31 DSP that the table should be processed once. When the 'C31 has pro–cessed the table it clears this bit to logic '0'. In this way the VME–bus can monitor this bit to determine if the table has been processed. The 'C31 will only process the table once since it resets the bit to '0' after processing. This bit can ONLY be set by the VMEbus, and can ONLY be cleared by the 'C31.

3.7.1.2 Schedule Command – Bit D1

This bit determines whether the 'C31 should process a command table immediately or schedule the command for execution at a later time, as indicated by the timestamp in Command Table Header Word 3. When this bit is set to logic '0', the 'C31 should execute the command now. When bit is set to logic '1', the 'C31 should schedule the command for execution later, at the time set in the Timestamp Word 3, see Section 3.7.3.

3.7 Dual Port Memory and Command Tables (continued)

3.7.1 Command Table Header Word #1: Command Control (continued)

3.7.1.3 Command Table Length – Bits D15 to D6

This 10–bit binary number represents the number of commands in the command table following the header. The minimum number is 0 and the maximum number is three less than the table length, or 61 for the default table length of 64. The most significant bit is TL9 (D15).

3.7.1.4 Data Format – Bits D17 to D16

This 2–bit field defines the data format of all floating–point parameter values associated with the command. Table 3–51, below, lists the different codes and the formats that are associated with them.

Table 3–51: Command Table Header Word 1 – Data Format

Data Format		Format
D17	D16	
0	0	*32–Bit Integer Pre–Calculated Register Setting:* Some parameters require a calculation to be performed to determine the appropriate register setting. When this data format is specified, no calculation is performed. The parameter is written directly into the register.
0	1	*32–Bit Integer Format:* Some parameters can be specified as floating point or as integers. This data format identifies the parameter as a 32–bit integer.
1	0	*Two's–Complement Floating–Point Format ('C4x and 'C3x Processors):* Some parameters can be specified as floating point. Texas Instruments' 'C30 and 'C40 generate a two's complement floating point value. This data format identifies the parameter as a 32–bit two's complement float.
1	1	*IEEE Single–Precision Standard 754 Floating–Point Format:* Some parameters can be specified as floating point. This data format identifies the parameter as a 32–bit IEEE float.

3.7 Dual Port Memory and Command Tables (continued)

3.7.2 Command Table Header Word #2: Channel Mask – R/W @ Channel_Mask

The second header word specifies the channels affected by this Command Table. Each of these sixteen bits represents one of the 16 DDR channels of the Model 6526. The number that follows each CM bit in the table below designates which channel is affected (e.g., CM9 is for channel 9). To execute the command in a given channel, the bit in this register associated with that channel must be set to the logic '1' state. Any combination of bits can be enabled, each bit signifying that the commands in the current table apply to that channel. Table 3–52, below, shows this register's bit layout.

Table 3–52: Command Table Header Word #2: Channel Mask
R/W @ Channel_Mask

Bit #	D31 – D16															
Bit Name	Reserved – Not Used															
Function	Write with Zeros, Mask when Reading															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	CM15	CM14	CM13	CM12	CM11	CM10	CM9	CM8	CM7	CM6	CM5	CM4	CM3	CM2	CM1	CM0
Function	0 = Channel INACTIVE 1 = Channel ACTIVE															
All bits default to the logic '0' state at power up and reset																

3.7.3 Command Table Header Word #3: Timestamp – R/W @ Timestamp

The third header word is the Timestamp, which is used for scheduling the execution of the commands within this Command Table. This timestamp is used to schedule the execution of a command when the Schedule Command bit, D1, in Command Table Header Word 1 is set to 1, see Section 3.7.1.2. Table 3–53, below, shows this register's bit layout — bit TS31 is the MSB.

Table 3–53: Command Table Header Word #3: Timestamp
R/W @ Timestamp

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	TS31	TS30	TS29	TS28	TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16
Function	16 Most Significant Bits of Timestamp															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Function	16 Least Significant Bits of Timestamp															
All bits default to the logic '0' state at power up and reset																

3.7 Dual Port Memory and Command Tables (continued)

3.7.4 Command Table Header Word #4: Command Status – R/W @ Command_Status

The fourth header word is a command status word, which reports any errors in command processing. The bits in this register provide indication of whether the command was executed correctly, or an error occurred while processing the command. Table 3–54, below, shows this register's bit layout, and the subsections following the table describe these bits.

Table 3–54: Command Table Header Word #4: Command Status
R/W @ Command_Status

Bit #	D31 – D16											
Bit Name	Reserved – Not Used											
Function	Write with Zeros, Mask when Reading											
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7 – D3	D2	D1	D0
Bit Name	CST7	CST6	CST5	CST4	CST3	CST2	CST1	CST0	Reserved – Not Used	Command_ Par_Error	Command_ Error	Command_ Complete
Function	Command Error Location								Write with Zeros, Mask when Reading	0 = OK 1 = Error	0 = OK 1 = Error	0 = Not 1 = Done
All bits default to the logic '0' state at power up and reset												

3.7.4.1 Command Complete – Bit D0

This bit indicates whether the command was completed or not. When read as a logic '0', the command is NOT completed. When read as a logic '1', the command is completed.

3.7.4.2 Command Error – Bit D1

This bit indicates that an error occurred while processing one or more of the commands in the command table. When this bit is read as a logic '0', there is no command error. When read as a logic '1', there is a command error.

3.7.4.3 Command Parameter Error – Bit D2

This bit indicates that the command contains an invalid command parameter. When read as logic '0', the command parameters are correct. When read as a logic '1', there is a parameter error.

3.7.4.4 Command Error Location – Bits D15 to D8

This 8–bit binary field identifies the position, relative to Command Table Header Word 1, of the first command which caused an error.

3.7 Dual Port Memory and Command Tables (continued)

3.7.5 Command Set

A list of commands is provided in the ReadyFlow board support software documentation for the Model 6526 (Pentek part #801.65260). The command set provides access to resources not available from the VMEbus. This includes the setup of the GC4014 Quad Receiver Chips, mapping local 'C31 Interrupt Status to VME Interrupts, Threshold Detection, and Input Toggling.

3.7.6 'C' Callable Library Functions

'C'−callable Library Functions are provided in the ReadyFlow board support software package for the Model 6526 to perform the low−level setting of the 6526 Registers. All functions are written to be platform independent — therefore, they can be called by the Local 'C31 or by any VMEbus Master. Routines are provided for initializing the GC4014 Quad Receiver Chip, Channel Formatter, Sync Generator Interrupt Mask, Timestamp Control, and Input Data De−Skewing FIFO's.

3.8 'C31 DSP Memory Maps and Registers

This section shows all memory resources available to the 'C31 processor. Some of these registers are also accessible in A24/A32 VMEbus address space, but only when the VMEbus Access Enable bit in the Board Control Register (Section 3.4.1.8) is set to the logic '1' state. These registers are described in the referenced sections of this manual (Sections 3.5, 3.6, and 3.7). The registers that are accessible only to the 'C31 processor are described in the subsections following these tables (3.8.x).

The 'C31 addressing convention in the following tables uses six hexadecimal digits, with 'X' indicating a 'don't care' digit.

Table 3−55: 'C31 DSP Memory Map − Local SRAM, Dual−Port SRAM, VMEbus Registers

'C31 Address	Register Description	Access	Additional Information
0x00 0000 – 0x00 0FFF	Reserved for Boot Loader Operations	–	–
0x00 1000 – 0x0F FFFF	U n u s e d	–	–
0x10 0000 – 0x1F FFFF	Local SRAM	Read/Write	–
0x12 0000 – 0x1F FFFF	U n u s e d	–	–
0x20 0000 – 0x20 1FFF	Dual Port SRAM: SwiftNet Space − 'C31 Port	Read/Write	Section 3.7
0x20 2000 – 0x20 3FFD	Dual Port SRAM: Command Tables − 'C31 Port	Read/Write	Section 3.7
0x20 3FFE	'C31 Inbound Mailbox Register	Read Only	Section 3.7
0x20 3FFF	'C31 Outbound Mailbox Register	Write Only	Section 3.7
0x30 0000	VME Board Control Register	Read/Write*	Section 3.5.1
0x30 0001	VME Time Stamp Counter Control Register	Read/Write*	Section 3.5.2
0x30 0002	VME Time Stamp Counter Register	Read Only*	Section 3.5.3
0x30 0003	VME Sync Generator Mask Register	Read/Write*	Section 3.5.4
0x30 0004	VME Channel Formatter Reset Control Register	Read/Write*	Section 3.5.5
0x30 0005	VME RACEway Status Register	Read Only*	Section 3.5.6
0x30 0006	VME Sync Generator Execute Register	Read/Write*	Section 3.5.7
0x30 0007	VME Output FIFO Status Register	Read Only*	Section 3.5.8
0x30 0008	VME Clock Status Register	Read Only*	Section 3.5.9
0x30 0009 – 0x30 3FFF	Unused	–	–

* These VME A24/A32 registers are accessible at the addresses listed in the table above ONLY when the VMEbus Access Enable bit in the A16 Board Control Register is cleared to the logic '0' state, which disconnects the VMEbus from the A24/A32 registers and connects them to the 'C31.

3.8 'C31 DSP Memory Maps and Registers (continued)

Table 3−56: 'C31 DSP Memory Map − Channel Formatter Registers			
'C31 Address (see Table 3−57)	**Register Description**	**Access***	**Additional Information**
0x30 0RY0	Channel #n Formatter Packet Start Code	Read Only	Section 3.6.2
0x30 0RY1	Channel #n Formatter Block Counter	Read Only	Section 3.6.4
0x30 0RY2	Channel #n Formatter Time Stamp Counter Register	Read Only	Section 3.6.5
0x30 0RY3	Channel #n Formatter DDR Output Register	Read Only	Section 3.6.6
0x30 0RY4	Channel #n Formatter Packet Stop Code	Read Only	Section 3.6.7
0x30 0RY5	Unused	–	–
0x30 0RY6	Channel #n Formatter Status Register	Read Only	Section 3.6.8
0x30 0RY7	Unused	–	–
0x30 0RY8	Channel #n Formatter Control Register	Read/Write	Section 3.6.9
0x30 0RY9	Channel #n Formatter ID Tag Register	Read/Write	Section 3.6.3
0x30 0RYA	Channel #n Formatter RACEway Packet Size	Read/Write	Section 3.6.10
0x30 RYB	Channel #n Formatter Sync Code Word Register	Read/Write	Section 3.6.11
0x30 0RYC – 0x30 0RYF	Unused	–	–
0x30 0RZ0	Channel #n Formatter RACEway Routing	Read/Write	Section 3.6.12
0x30 0RZ1	Channel #n Formatter RACEway Start Address	Read/Write	Section 3.6.13
0x30 0RZ2	Channel #n Formatter RACEway End Address	Read/Write	Section 3.6.14
0x30 0RZ3 – 0x30 0RZF	Unused	–	–
* These VME A24/A32 registers are accessible at the addresses listed in the table above ONLY when the VMEbus Access Enable bit in the A16 Board Control Register is cleared to the logic '0' state, which disconnects the VMEbus from the A24/A32 registers and connects them to the 'C31.			

Table 3−57: 'C31 Address Ranges for Channel Formatter Registers			
Channel	**Address Range**	**RY =**	**RZ =**
0	0x30 0400 – 0x30 0412	40	41
1	0x30 0440 – 0x30 0452	44	45
2	0x30 0480 – 0x30 0492	48	49
3	0x30 04C0 – 0x30 04D2	4C	4D
4	0x30 0500 – 0x30 0512	50	51
5	0x30 0540 – 0x30 0552	54	55
6	0x30 0580 – 0x30 0592	58	59
7	0x30 05C0 – 0x30 05D2	5C	5D
8	0x30 0600 – 0x30 0612	60	61
9	0x30 0640 – 0x30 0652	64	65
10	0x30 0680 – 0x30 0692	68	69
11	0x30 06C0 – 0x30 06D2	6C	6D
12	0x30 0700 – 0x30 0712	70	71
13	0x30 0740 – 0x30 0752	74	75
14	0x30 0780 – 0x30 0792	78	79
15	0x30 07C0 – 0x30 07D2	7C	7D

3.8 'C31 DSP Memory Maps and Registers (continued)

Table 3−58: 'C31 DSP Memory Map − VME port DPSRAM, Boot Flash, 'C31, Graychip

'C31 Address	Register Description	Access	Additional Information
0x30 4000 – 0x30 5FFF	Dual Port SRAM: SwiftNet Space − VME Port	Read/Write*	Section 3.7
0x30 6000 – 0x30 7FFF	Dual Port SRAM: Command Tables − VME Port	Read/Write*	Section 3.7
0x40 0000 – 0x40 7FFF	Boot Load EEPROM	Read Only	Appendix A
0x40 8000 – 0x4F FFFF	U n u s e d	–	–
0x5X XX0X†	'C31 LED Register	Read/Write	Section 3.8.1
0x5X XX1X†	'C31 Sync Arm Register	Read/Write	Section 3.8.2
0x5X XX2X†	'C31 Sync Word Substitution Register	Read Only	Section 3.8.3
0x5X XX3X†	'C31 Sync Block Counter Reset Register	Read Only	Section 3.8.4
0x5X XX4X†	'C31 Control Register	Read/Write	Section 3.8.5
0x60 0000 – 0x61 FFFF	Flash Memory	Read/Write	–
0x62 0000 – 0x6F FFFF	U n u s e d	–	–
0x70 0000 – 0x70 001F	Graychip GC4014 #0 Registers	Read/Write	Appendix C
0x70 0020 – 0x70 003F	Graychip GC4014 #1 Registers	Read/Write	Appendix C
0x70 0040 – 0x70 005F	Graychip GC4014 #2 Registers	Read/Write	Appendix C
0x70 0060 – 0x70007F	Graychip GC4014 #3 Registers	Read/Write	Appendix C
0x70 0080 – 0x7F FFFF	U n u s e d	–	–

* These VME A24/A32 registers are accessible at the addresses listed in the table above ONLY when the VMEbus Access Enable bit in the A16 Board Control Register is cleared to the logic '0' state, which disconnects the VMEbus from the A24/A32 registers and connects them to the 'C31.

† X = "Don't Care" — the states of these address bits are not decoded.

3.8 'C31 DSP Memory Maps and Registers (continued)

Table 3−59: 'C31 DSP Memory Map − Internal, Time Stamp, Overload, Interrupts

'C31 Address	Register Description	Access	Additional Information
0x80 0000 – 0x80 7FFF	Reserved	–	–
0x80 8000 – 0x80 97FF	Internal Peripheral Bus Memory Mapped Registers	Read/Write	See Texas Instruments TMS320C3x User's Guide, TI publication number 2558539−9761
0x80 9800 – 0x80 9BFF	Internal RAM − Block 0	Read/Write	
0x80 9C00 – 0x80 9FC0	Internal RAM − Block 1	Read/Write	
0x80 9FC1 – 0x80 9FFF	User Program, Interrupt, & Trap Branches	Read/Write	
0x80 A000 – 0x8F FFFF	Unused	–	–
0x9X XXXX†	Time Stamp Counter Output Register	Read Only	Section 3.8.6
0xAX XX0X†	Overload A Control Register	Read/Write	Section 3.8.7
0xAX XX1X†	Overload B Control Register	Read/Write	Section 3.8.7
0xAX XX2X†	Interrupt Vector Register	Read/Write	Section 3.8.8
0xAX XX30†	Interrupt Mask Register	Read/Write	Section 3.8.9
0xAX XX40†	Interrupt Status Register #0	Read/Clear	Section 3.8.10
0xAX XX41†	(Reserved for Interrupt Status Register #1)	–	Section 3.8.11
0xAX XX42†	Interrupt Status Register #2	Read/Clear	Section 3.8.12
0xAX XX43†	(Reserved for Interrupt Status Register #3)	–	Section 3.8.13
0xAX XX5X†	Built−In Self−Test Register	Read/Write	Section 3.8.14
0xAX XX6X – 0xFX XXXX†	Unused	–	–

† X = "Don't Care" — the states of these address bits are not decoded.

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.1 'C31 LED Control Register – R/W @ 'C31 Address 0x5X XX0X

This register controls two of the LEDs on the Model 6526 front panel. Table 3–60, below, shows this register's bit layout, and the subsections following the table describe these bits.

Table 3–60: 'C31 LED Control Register
R/W @ 'C31 Address: 0x5X XX0X (X = "Don't Care" bits)

Bit #	D15 – D10	D9	D8 – D1	D0
Bit Name	Reserved – Not Used	DSP_LED	Reserved – Not Used	BIST_Fail
Function	Write with Zeros, Mask when Reading	0 = Off 1 = On	Write with Zeros, Mask when Reading	0 = Off 1 = On
All bits default to the logic '0' state at power up and reset				

3.8.1.1 BIST Fail – Bit D0

This bit drives the front panel FAIL LED (see Section 2.4.5.7). It is set by the DSP to indicate a failure in the Built–In Self–Test (see also Section 3.8.14, the Built–In Self–Test Register). This LED is on when this bit is set to the logic '1' state, and off when the bit is cleared to the logic '0' state.

3.8.1.2 DSP LED – Bit D9

This bit drives the front panel DSP LED (see Section 2.4.5.6). It is set by the DSP to indicate that the DSP is operating. This LED is on when this bit is set to the logic '1' state, and off when the bit is cleared to the logic '0' state.

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.2 'C31 Sync Arm Register – R/W @ 'C31 Address 0x5X XX1X

This register enables the generation of local sync pulses and the Time Stamp Counter reset. Table 3–61, below, shows this register's bit layout, and the subsections following the table describe these bits.

Table 3–61: 'C31 Sync Arm Register
R/W @ 'C31 Address: 0x5X XX1X (X = "Don't Care" bits)

Bit #	D15 – D5	D4	D3	D2	D1	D0
Bit Name	Reserved – Not Used	Time_Stamp_ Cntr_Arm	Local_Sync_ Arm3	Local_Sync_ Arm2	Local_Sync_ Arm1	Local_Sync_ Arm0
Function	Write with Zeros, Mask when Reading	0 = Disarm 1 = Arm	0 = Disarm Local Sync Generation 1 = Arm Local Sync Pulse Generation			
All bits default to the logic '0' state at power up and reset						

3.8.2.1 Local Sync Arm – Bits D3 to D0

This function allows the 'C31 to generate local SYNC pulses for use within the board. By setting the Local Sync Arm bits to logic '1', the next Time Stamp Clock (from the Time Stamp Pre–Scaler) will generate a SYNC pulse on the selected local Sync bus signal. When set to logic '0', local sync generation is not armed. Bit D0 is for SYNC 0, D1 is for SYNC 1, and so on.

The Local Sync Arm bit is automatically cleared by the generated SYNC pulse, and thus the SYNC pulse will last exactly one Time Stamp Clock interval.

NOTE: SYNC pulses created by the 'C31 using this function are local to the board and are not sent out on the front panel Sync bus to other boards, even if the board is configured as a Sync bus master.

3.8.2.2 Time Stamp Counter Reset Arm – Bit D4

This function allows the 'C31 to enable the next Time Stamp Clock (from the Time Stamp Pre–Scaler) to reset the Time Stamp Counter to zero. This reset is armed when this bit is set to the logic '1' state, and disarmed when the bit is cleared to the logic '0' state.

This bit is automatically cleared by the Time Stamp Clock, which resets the Time Stamp Counter.

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.3 'C31 Sync Word Substitution Register – R/W @ 'C31 Address 0x5X XX2X

This register enables Sync Word Substitution in the data packet output formed by Channel Formatter 'n', where 'n' of the bit name 'SSn' is the channel formatter number from 0 to 15. Substitution is enabled when the bit is set to the logic '1' state, and disabled when the bit is cleared to the logic '0' state. These enable bits are OR'ed with the VME channel formatter control register bits of the same function, so for proper operation from this register, all VME Sync Substitution Enables must be turned off (see Section 3.6.9 for more details). This register is not automatically cleared by a SYNC pulse.

Table 3–62, below, shows this register's bit layout.

Table 3–62: 'C31 Sync Word Substitution Register
R/W @ 'C31 Address: 0x5X XX2X (X = "Don't Care" bits)

Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	SS15	SS14	SS13	SS12	SS11	SS10	SS9	SS8	SS7	SS6	SS5	SS4	SS3	SS2	SS1	SS0
Function	0 = Disable Sync Word Substitution 1 = Enable Sync Word Substitution															
All bits default to the logic '0' state at power up and reset																

3.8.4 'C31 Sync Block Counter Reset Register – R/W @ 'C31 Address 0x5X XX3X

This register enables Block Counter Reset in Channel Formatter 'n', where 'n' of the bit name 'BRn' is the channel formatter number from 0 to 15. This function is enabled when the bit is set to the logic '1' state, and disabled when the bit is cleared to the logic '0' state. These enable bits are OR'ed with the VME channel formatter control register bits of the same function, so for proper operation from this register, all VME Block Counter Reset Enables must be turned off (see Section 3.6.9 for more details). This register is not automatically cleared by a SYNC pulse.

Table 3–63, below, shows this register's bit layout.

Table 3–63: 'C31 Sync Block Counter Reset Register
R/W @ 'C31 Address: 0x5X XX3X (X = "Don't Care" bits)

Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
Function	0 = Disabled 1 = Enabled															
All bits default to the logic '0' state at power up and reset																

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.5 'C31 Control Register – R/W @ 'C31 Address 0x5X XX4X

This register controls several clock sources, plus provides a local reset of the board. Table 3–64, below, shows this register's bit layout, and the subsections following the table describe these bits.

Table 3–64: 'C31 Control Register
R/W @ 'C31 Address: 0x5X XX4X (X = "Don't Care" bits)

Bit #	D15 – D12	D11	D10	D9	D8	D7 – D4	D3 – D0
Bit Name	Reserved - Not Used	Local_General_ HW_Reset	VME_Interrupt	Test_Clock	Not Used	Sync_Source	Clock_Source
Function	Write with Zeros, Mask when Reading	0 = Off 1 = Reset	0 = No Int 1 = Interrupt	0 = Normal 1 = Test Clock	Write 0's - Mask Read	0 = Clock A 1 = Clock B	0 = Clock A 1 = Clock B
All bits default to the logic '0' state at power up and reset							

3.8.5.1 Clock Source – Bits D3 to D0

These four bits select the clock sources for the GC4014 DDR's and their associated data channels. Input Clock B is selected when the bit is set to the logic '1' state, and Input Clock A is selected when the bit is cleared to the logic '0' state. The channel and GC4014 associations for each bit are shown in Table 3–65 below.

Table 3–65: Clock Source / Channel Association

Clock_Source bit	GC4014	Data Channels
D0	#0	0, 1, 2, 3
D1	#1	4, 5, 6, 7
D2	#2	8, 9, 10, 11
D3	#3	12, 13, 14, 15

Refer to the block diagram and discussion in Section 3.5.10 for more details.

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.5 'C31 Control Register (continued)

3.8.5.2 Sync Source – Bits D7 to D4

This function selects the clock source for the 'C31 Sync Pulse Generation circuit. Input Clock B is selected when the bit is set to the logic '1' state, and Input Clock A is selected when the bit is cleared to the logic '0' state. This should be set to the same clock source as is used for all GC4014s being synchronized. Note that this SYNC pulse generation is OR'ed with the SYNC pulse generated by VMEbus commands using the Sync Generate Mask and Sync Generate Command Registers, Sections 3.5.4 and 3.5.7. The SYNC signals associated with each bit are shown in Table 3–66 below.

Table 3–66: Sync Source / Signal Association

Sync_Source bit	SYNC
D4	#0
D5	#1
D6	#2
D7	#3

3.8.5.3 Test Clock – Bit D9

This bit selects the on–board 30 MHz clock as the input data clock for the GC4014s and Packet Formatters. This may be used for built–in test without the need for an input clock. The on–board clock is selected when the bit is set to the logic '1' state, and the input clock is selected when the bit is cleared to the logic '0' state.

3.8.5.4 VME Interrupt – Bit D10

This bit generates a VME interrupt. A VME Interrupt Acknowledge (IACK cycle) automatically clears it. An Interrupt Request is generated when the bit is set to the logic '1' state, and no interrupt is requested when the bit is cleared to the logic '0' state.

3.8.5.5 Local General Hardware Reset – Bit D11

When this bit is set to logic '1', all general hardware functions on the Model 6526 board are held in reset except for the 'C31 processor. In addition, all A24/A32 registers are cleared. This bit is logically OR'ed with the General Hardware Reset bit in the VME Board Control Register (see Section 3.4.1).

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.6 Time Stamp Counter Output Register – R/W @ 'C31 Address 0x9X XXXX

This register is the 32−bit Time Stamp Counter's output value. TS31 is the most significant bit and TS0 the least significant bit. It is a read−only register. Table 3−67, below, shows this register's bit layout.

Table 3−67: Time Stamp Counter Output Register
R.O. @ 'C31 Address: 0x9X XXXX (X = "Don't Care" bits)

Bit #	D31	D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17	D16
Bit Name	TS31	TS30	TS29	TS28	TS27	TS26	TS25	TS24	TS23	TS22	TS21	TS20	TS19	TS18	TS17	TS16
Function	16 Most Significant Bits of Timestamp															
Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	TS15	TS14	TS13	TS12	TS11	TS10	TS9	TS8	TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0
Function	16 Least Significant Bits of Timestamp															
All bits default to the logic '0' state at power up and reset																

3.8.7 Overload Detection Control Registers – R/W @ 'C31 Address 0xAX XXSX

These registers control several Overload Detection features for the two input channels of the Model 6526. There are two registers — the register for input channel A is at 'C31 address 0xAX XX0X, and the register for input channel B is at address 0xAX XX1X. Table 3−68, below, shows this register's bit layout, and the subsections following the table describe these bits.

Table 3−68: 'C31 Overload Detection Control Registers
R/W @ 'C31 Address: 0xAX XXSX (X = "Don't Care" bits; S = 0 for channel A, 1 for channel B)

Bit #	D15 – D8	D7	D6 – D5	D4 – D3	D2 – D0
Bit Name	Reserved − Not Used	Detector _Enable	Threshold_Level	Not Used	Crossing_Count
Function	Write with Zeros, Mask when Reading	0 = Disabled 1 = Enabled	See Table 3-70	Write Zeros, Mask Read	See Table 3-69
All bits default to the logic '0' state at power up and reset					

3.8.7.1 Crossing Count − Bits D2 to D0

These bits select the Threshold Crossing Counter value that can generate an interrupt to the 'C31 DSP. Table 3−69, on the next page, shows the bit codes for this field and the counter value that each code selects.

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.7 Overload Detection Control Registers (continued)

3.8.7.1 Crossing Count (continued)

Table 3−69: Crossing Count Codes			
D2	**D1**	**D0**	**Count**
0	0	0	2
0	0	1	8
0	1	0	32
0	1	1	128
1	0	0	512
1	0	1	2048
1	1	0	8192
1	1	1	32768

3.8.7.2 Threshold Level − Bits D6 to D5

These bits select the data input level that is recognized as the threshold, in accordance with Table 3−70 below. The Threshold Crossing Counter is incremented for each data sample on the channel of interest that is either greater than or equal to the positive data value, or less than or equal to the negative data value.

Table 3−70: Threshold Level Codes				
D6	**D5**	**Threshold Setting**	**Positive Data Value (Hex)**	**Negative Data Value (Hex)**
0	0	±1/2 of Full Scale	≥0x0400	≤0x0BFF
0	1	±5/8 of Full Scale	≥0x0500	≤0x0AFF
1	0	±3/4 of Full Scale	≥0x0600	≤0x09FF
1	1	±7/8 of Full Scale	≥0x0700	≤0x08FF

3.8.7.3 Overload Detector Enable − Bit D7

When this bit is cleared to the logic '0' state, the Overload Detection function is disabled. The low−going transition of this bit resets the Threshold Crossing Counter to the zero count. To enable the Overload Detection function and release the Threshold Crossing counter, set this bit to the logic '1' state. The counter reset function should be implemented as a toggle (i. e., clear this bit, then set it).

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.8 Interrupt Vector Register – R/W @ 'C31 Address 0xAX XX2X

This register is the interrupt vector for an interrupt from the 'C31 DSP to a VMEbus master. Table 3–71, below, shows this register's bit layout. Bit V7 is the most significant bit of the address.

Table 3–71: Interrupt Vector Register
R/W @ 'C31 Address: 0xAX XX2X (X = "Don't Care" bits)

Bit #	D15 – D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Reserved – Not Used	V7	V6	V5	V4	V3	V2	V1	V0
Function	Write with Zeros, Mask when Reading	Interrupt Vector							
All bits default to the logic '0' state at power up and reset									

3.8.9 Interrupt Mask Register – R/W @ 'C31 Address 0xAX XX30

The interrupt mask register allows events or conditions to interrupt the 'C31. When the interrupt mask bit is set to logic '1', the event is enabled to cause an interrupt. Once the interrupt event occurs, it is latched in the Interrupt Status Register (see Section 3.8.10). Table 3–72, below, shows this register's bit lay–out. Table 3–73, below, identifies the interrupt condition for each bit.

Table 3–72: Interrupt Mask Register
R/W @ 'C31 Address: 0xAX XX30 (X = "Don't Care" bits)

Bit #	D15 – D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Reserved – Not Used	DPRU	DPRL	IFB	IFA	FIF	CLKR	CLKB	CLKA	TS	OVLB	OVLA
Function	Write with Zeros, Mask when Reading	0 = Interrupt Disabled 1 = Interrupt Enabled										
All bits default to the logic '0' state at power up and reset												

Table 3–73: Interrupt Bit Conditions

Bit #	Bit Name	Interrupt Condition
D0	OVLA	Input Channel A Overload Detector Interrupt
D1	OVLB	Input Channel B Overload Detector Interrupt
D2	TS	Time Stamp Interrupt
D3	CLKA	Input Channel A Clock Loss Interrupt
D4	CLKB	Input Channel B Clock Loss Interrupt
D5	CLKR	Raceway Clock Loss Interrupt
D6	FIF	Output FIFO Full / Formatter Sync Loss Interrupt (see NOTE, next page)
D7	IFA	Input FIFO A Full Interrupt
D8	IFB	Input FIFO B Full Interrupt
D9	DPRL	Dual Port SRAM Mailbox (Lower Byte) Interrupt
D10	DPRU	Dual Port SRAM Mailbox (Upper Byte) Interrupt

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.9 Interrupt Mask Register (continued)

NOTE: The FIF interrupt bit (D6) enables interrupts from all sixteen Channel Formatter Sync Error Interrupts, which are caused by the channel formatter being unable to write to a full Output FIFO.

3.8.10 Interrupt Status Register #0 – R/Clr @ 'C31 Address 0xAX XX40

The interrupt status registers allow an interrupt service routine (ISR) to deter–mine the source and cause of the interrupt. Since many of the interrupting events and conditions may be transient, they are latched in these registers so that the processor can determine the cause. When a bit is read as logic '1', an interrupt request is active, when read as logic '0', there is no interrupt active.

The ISR can clear any bit in this register that is in the active state (i. e., in the logic '1' state) by re–writing a '1' to that bit, followed by a '0'. (That is, writing a '1' followed by a '0' to any bit in this register that is in the logic '1' state, clears it to logic '0'.) Table 3–74, below, shows this register's bit layout. Table 3–73 in Section 3.8.9, above, identifies the interrupt condition for each bit.

Table 3–74: Interrupt Status Register #0

R/Clr @ 'C31 Address: 0xAX XX40 (X = "Don't Care" bits)

Bit #	D15 – D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Reserved – Not Used	DPRU	DPRL	IFB	IFA	FIF	CLKR	CLKB	CLKA	TS	OVLB	OVLA
Function	Write with Zeros, Mask when Reading	Read: 0 = No Interrupt, 1 = Interrupt Active Clear: 1 = Clear, 0 = Normal Operation										
All bits default to the logic '0' state at power up and reset												

3.8.11 Interrupt Status Register #1: VME Interrupt – R/W @ 'C31 Address 0xAX XX41

This register is not implemented, but reserved for future use. To minimize interrupt processing overhead, an incoming VME interrupt to Model 6526 drives the 'C31 Interrupt 1 (INT1). There is no status register associated with this interrupt since the VME interrupt is the only cause of this interrupt. Data read from or written to this register is meaningless.

Table 3–75: Interrupt Status Register #1

R/W @ 'C31 Address: 0xAX XX41 (X = "Don't Care" bits)

Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Reserved – Not Used															
Function	N/A															

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.12 Interrupt Status Register #2: Channel Formatter Error / FIFO Full Interrupt – R/Clr @ 'C31 Address 0xAX XX30

This register indicates that an error has occurred in Channel Formatter 'n', where 'n' of the bit name 'FFn' is the channel formatter number from 0 to 15. This error can occur when sending RACEway packet data to the Output FIFO if that FIFO is full. When this occurs, packet integrity is lost since the packet contents may be corrupted. This is a latched condition. When a bit is read as logic '1', an interrupt request is active, when read as logic '0', there is no interrupt active.

The interrupt service routine can clear any bit in this register that is in the active state (in the logic '1' state) by writing a '1' to that bit, followed by a '0'. (That is, writing a '1' followed by a '0' to any bit in this register that is in the '1' state, clears it to the logic '0' state.) Table 3–76, below, shows this register's bit layout.

Table 3–76: Interrupt Status Register #2

R/Clr @ 'C31 Address: 0xAX XX42 (X = "Don't Care" bits)

Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	FF15	FF14	FF13	FF12	FF11	FF10	FF9	FF8	FF7	FF6	FF5	FF4	FF3	FF2	FF1	FF0
Function	Read: 0 = No Interrupt, 1 = Interrupt Active Clear: 1 = Clear, 0 = Normal Operation															
All bits default to the logic '0' state at power up and reset																

3.8.13 Interrupt Status Register #3: Time Stamp Interrupt – R/W @ 'C31 Address 0xAX XX43

This register is not implemented, but reserved for future use. To minimize interrupt processing overhead, the Time Stamp Clock interrupt drives the 'C31 Interrupt 3 (INT3). There is no status register associated with this interrupt since the Time Stamp Clock interrupt is the only cause of this interrupt. Data read from or written to this register is meaningless.

Table 3–77: Interrupt Status Register #3

R/W @ 'C31 Address: 0xAX XX43 (X = "Don't Care" bits)

Bit #	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Bit Name	Reserved – Not Used															
Function	N/A															

3.8 'C31 DSP Memory Maps and Registers (continued)

3.8.14 Built–In Self–Test (BIST) Register – R/W @ 'C31 Address 0xAX XX5X

This register controls the Built–In Self–Test function of the Model 6526. It allows you to start self–test, to monitor for completion of testing, and to determine the test results. Table 3–78, below, shows this register's bit layout. This register is the same as Built–In Self–Test Register described in the A16 Register Maps, Section 3.4.6.

Table 3–78: Built–In Self–Test Register
R/W @ 'C31 Address: 0xAX XX5X (X = "Don't Care" bits)

Bit #	D15 – D9								
Bit Name	Reserved – Not Used								
Function	Write with zeros, Mask when reading								
Bit #	D8*	D7*	D6*	D5*	D4*	D3*	D2*	D1*	D0
Bit Name	GC4014_3 _Result	GC4014_2 _Result	GC4014_1 _Result	GC4014_0 _Result	DPSRAM_ _1_Result	DPSRAM _0_Result	LSRAM_ Result	BIST _Active	BIST _Enable
Function	0 = Passed 1 = Failed							1 = Active 0 = Inactive	1 = Run BIST 0 = Complete

* These bits are Read Only
All bits default to the logic '0' state at power up and reset

This page is intentionally blank

Appendix A: Boot EEPROM Source Code Listing

A.1 Boot EPROM Source Code Listing

```
       .list
*
*       TITLE 'PROCESSOR INITIALIZATION'
*
       .global RESET,INIT,BEGIN
       .global NMI,INT0,INT1,INT2,INT3
       .global NON_MASK,ISR0,ISR1,ISR2,ISR3
       .global LOOK,SET
*
*********************************************************************
*                                                                   *
*               LOOK/SET memory usage                               *
*       HOST   C31                                                  *
*       -----  --------                                             *
*       0x800  0x200200 = Jump address from Host                    *
*       0x804  0x200201 = Address for look/set function             *
*                       = Source address for block move             *
*       0x808  0x200202 = Data from Host for set function           *
*                       = Destination address for block move        *
*       0x80c  0x200203 = Data to Host from look function           *
*                       = Word count-1  for block move              *
*       0x810  0x200204 = Ready to Host  (0=not ready / 1=ready)    *
*       0x814  0x200205 = Pointer to LOOK function                  *
*       0x818  0x200206 = Pointer to SET function                   *
*       0x81c  0x200207 = Pointer to  BLOCK MOVE function           *
*       0x820  0x200208 = Aux ready flag for factory use            *
*       0x824  0x200209 = Test pattern 0xa5a5a5a5                   *
*       0x828  0x20020a = Test pattern 0x5a5a5a5a                   *
*       0x82c  0x20020b = Test pattern 0x12345678                   *
*       0x830  0x20020c = Test pattern 0x87654321                   *
*       0x834  0x20020d = Test pattern 0xffffffff                   *
*       0x838  0x20020e = Test pattern 0x00000000                   *
*       0x83c  0x20020f = Boot Code Version     -- 1.0c --          *
*       0x840  0x200210 = Test pattern 0x80??0200                   *
*                                                                   *
*       FLASH support                                               *
*                                                                   *
*       0x844  0x200211 = Pointer to FLASH BOOT function            *
*       0x848  0x200212 = Pointer to FLASH LOOK function            *
*       0x84c  0x200213 = Pointer to FLASH SET function             *
*       0x850  0x200214 = Pointer to FLASH ERASE function           *
*       0x854  0x200215 = Pointer to FLASH ERASE SECTOR function    *
*       0x858  0x200216 = Pointer to FLASH BYTE LOAD function       *
*       0x85c  0x200217 = Pointer to FLASH BYTE UNLOAD function     *
*       0x860  0x200218 = Pointer to FLASH WORD LOAD function       *
*       0x864  0x200219 = Pointer to FLASH WORD UNLOAD function     *
*                                                                   *
*********************************************************************
```

A.1 Boot EPROM Source Code Listing (continued)

```
*
*       PROCESSOR INITIALIZATION FOR THE TMS320C30.
*
*
*       IN THIS SECTION, CONSTANTS THAT CANNOT BE REPRESENTED
*       IN THE SHORT FORMAT ARE INITIALIZED.
       .text
*
       BR      INIT           ;
INTTBL  .word   INT0           ;
INT0    BR      ISR0           ; INT0-
INT1    BR      ISR1           ; INT1-
INT2    BR      ISR2           ; INT2-
INT3    BR      ISR3           ; INT3-
XINT0   BR      NOTHIN         ;
RINT0   BR      NOTHIN         ;
       BR      NOTHIN         ;
       BR      NOTHIN         ;
TINT0   BR      TIME0          ; Timer 0 interrupt processing
TINT1   BR      TIME1          ; Timer 1 interrupt processing
DINT0   BR      NOTHIN         ;
*
LOOKADR      .word   LOOK      ; LOOK function
SETADR       .word   SET       ; SET function
MOVEADR      .word   BLKMOV    ; Block move function
ZIPPO        .word   NOPRTN    ; At boot up stik donothin
PATRNS       .word   PATRN     ; Test patterns
FBOOTA       .word   FBOOT     ; FLASH BOOT function
FLOOKA       .word   FLOOK     ; FLASH LOOK function
FSETA        .word   FSET      ; FLASH SET function
FLDBA        .word   FLDB      ; FLASH LOAD BYTE function
FUNLDBA      .word   FUNLDB    ; FLASH UNLOAD BYTE function
FLDWA        .word   FLDW      ; FLASH LOAD WORD function
FUNLDWA      .word   FUNLDW    ; FLASH UNLOAD WORD function
ERASEA       .word   ERASE     ; FLASH ERASE function
SERASEA      .word   SERASE    ; FLASH SECTOR ERASE function
;
;
;
MASK         .word   0ffffffffH      ;
BOOTF        .word   0003fffffH      ; Boot failed blink value
BOOTP        .word   00003ffffH      ; Boot passed blink value
IIEMASK      .word   000000102H      ; Interrupt enable mask
IVTADR       .word   000809fc1H      ;*Beginning address of interrupt vectors
CTRL         .word   000808000H      ; Pointer for peripheral-bus memory map
C31REG       .word   000500000H      ; Processor IO registers
STCK         .word   000809f00H      ; Beginning of stack
*
PRIMINT      .word   000000000H      ; Init of local memory interface control (4)
LOCALINT     .word   000000000H      ; Init of local memory interface control (4)
FLASHINT     .word   000000000H      ; Init of local during flash access (4)
```

A.1 Boot EPROM Source Code Listing (continued)

```
*
PATRN         .word   0A5A5A5A5H      ;
              .word   05A5A5A5AH      ;
              .word   012345678H      ;
              .word   087654321H      ;
              .word   0ffffffffH      ;
              .word   000000000H      ;
              .word   06526010bH      ; Version of Boot code
PROC_ADDR     .word   000000000H      ;
PROC_TEST     .word   000000000H      ; test address
*
JUMPW         .word   sub_w
JUMPH         .word   sub_h
JUMPB         .word   sub_b
*
*
*
*
*
*
PRO1_ADDR     .word 00200200H    ; Memory "A"
*
*
*
GBASE         .word 00200000H    ;
GBOOTL        .word 00200000H    ; Points to boot global flag
GBOOTJ        .word 00200001H    ; Contains global address to boot from
FBASE         .word 00600000H    ;
FBOOTL        .word 00600000H    ; Points to boot flash flag
FBOOTJ        .word 00600001H    ; Contains flash address to boot from
FSEQ55        .word 00605555H    ;
FSEQAA        .word 00602AAAH    ;
FSEQRST       .word 0f0f0f0f0H   ; Read/Reset
FSEQWRT       .word 0a0a0a0a0H   ; Write
FSEQER1       .word 080808080H   ; Erase seq start
FSEQER2       .word 010101010H   ; Chip erase
FSEQER3       .word 030303030H   ; Sector erase
FLASHFG       .word 0A55A5AA5H   ; Boot flash flag
FLASHNB       .word 0c33c3cc3H   ; Block boot flash flag
FBOOTT        .word 0            ; Holds boot address
FAA           .word 0AAAAAAAAH   ;
F55           .word 055555555H   ;
F80           .word 080808080H   ;
BISTCNTL      .word 00A00050H    ; Bist Control Register
LMST          .word 00100000H    ;
LMLEN         .word 00020000H    ;
BIST_RESULTS .word 000000000H   ; Bist Results
errcode       .word 0    ;
erradr        .word 0    ;
err7          .word 0    ;
err1          .word 0    ;
err6          .word 0    ;
```

A.1 Boot EPROM Source Code Listing (continued)

```
*
INIT                                    ;
       LDI     080h,DP                  ; Point the DP register to page 80h
       LDI     @STCK,SP                 ; Init stack pointer to 2ffff00h
       LDI     @CTRL,AR0                ; Point to control register
       LDI     @PRIMINT,R0              ; Init PRIMARY interface control
       STI     R0,*+AR0(064H)           ; External ready only
* Move interrupt vectors
       LDI     @INTTBL,AR1              ; Setup interrupt vectors
       LDI     @IVTADR,AR2              ;
       LDI     10,RC                    ; Set count
       RPTB    IVTMVL                   ;
       LDI     *AR1++,R0                ;
IVTMVL
       STI     R0,*AR2++                ;
* Turn off front panel LED
       LDI     @C31REG,AR3              ;
       LDI     0300H,R0                 ; LED off / Bus timeout disable
       STI     R0,*AR3                  ; Set register
*
PROC1S
       LDI     @PRO1_ADDR,R0            ;
       STI     R0,@PROC_ADDR            ;
       ADDI    9,R0                     ;
       STI     R0,@PROC_TEST            ;
*
* Setup pointers for HOST to use
       LDI     @PROC_ADDR,AR2           ; Processor base address
       LDI     @ZIPP0,R0                ; Init FUNC address
       STI     R0,*+AR2(0)              ;
       LDI     @LOOKADR,R0              ; Init LOOK address
       STI     R0,*+AR2(5)              ;
       LDI     @SETADR,R0               ; Init SET address
       STI     R0,*+AR2(6)              ;
       LDI     @MOVEADR,R0              ; Init BLOCK MOVE address
       STI     R0,*+AR2(7)              ;
       LDI     @FBOOTA,R0               ; Init BOOT address
       STI     R0,*+AR2(17)             ;
       LDI     @FLOOKA,R0               ; Init LOOK address
       STI     R0,*+AR2(18)             ;
       LDI     @FSETA,R0                ; Init SET address
       STI     R0,*+AR2(19)             ;
       LDI     @ERASEA,R0               ; Init ERASE address
       STI     R0,*+AR2(20)             ;
       LDI     @SERASEA,R0              ; Init SECTOR ERASE address
       STI     R0,*+AR2(21)             ;
       LDI     @FLDBA,R0                ; Init LOAD BLOCK BYTE address
       STI     R0,*+AR2(22)             ;
       LDI     @FUNLDBA,R0              ; Init UNLOAD BLOCK BYTE address
       STI     R0,*+AR2(23)             ;
       LDI     @FLDWA,R0                ; Init LOAD BLOCK WORD address
       STI     R0,*+AR2(24)             ;
```

A.1 Boot EPROM Source Code Listing (continued)

```
       LDI     @FUNLDWA,R0             ; Init UNLOAD BLOCK WORD address
       STI     R0,*+AR2(25)            ;
*
* Store data patterns in memory
       LDI     0,R0
       STI     R0,@BIST_RESULTS         ;
       LDI     @PATRNS,AR1             ; Processor test patterns
       LDI     @PROC_TEST,AR4          ; Processor test address
       LDI     *AR1++,R0               ;
       RPTS    6                       ; Move seven patterns
       STI     R0,*AR4++               ;
||      LDI     *AR1++,R0               ;
       STI     R0,*AR4++               ;
       NOP                             ;
       NOP                             ;
       NOP                             ;
*
* Check test patterns - Always do this whether BIST is selected or not
       LDI     @PATRNS,AR1             ; Processor test patterns
       LDI     @PROC_TEST,AR4          ; Processor test address
       LDI     6,AR5                   ;
CKDATA
       LDI     *AR1++,R0               ;
||      LDI     *AR4++,R1               ;
       CMPI    R0,R1                   ;
       BNZ     DPRBD                   ;
       DB      AR5,CKDATA              ;
CKEND
       NOP                             ;
       NOP                             ;
       NOP                             ;
        BLSRTST
DPRBD
       LDI     8,R0
       STI     R0,@BIST_RESULTS         ;
*
* Perform a Memory Test on Local SRAM if BIST Enabled
*
LSRTST
        LDI     @BISTCNTL,AR1           ; Load BIST Control word into r0
        LDI     *AR1,R0
        TSTB    1,R0                    ; If D0 = 1, Run Memtest
        BZ      CHKBIST                 ; If D0 = 0, Bypass Memtest
        OR      2,R0                    ; Set BIST Active
        ANDN    03fch,R0                ; Clear Bist Results
        LDI     @BIST_RESULTS,R1        ; Load BIST Results word into r1
        OR      R1,R0
        STI     R0,*AR1
;       LDI     @C31REG,AR0             ; Turn OFF Fail LED
;       LDI     0001H,R0                ; LED bit
;       LDI     *+AR0(0),R1             ; Get register setting
;       ANDN    R0,R1                   ; Clear Failed LED bit
```

A.1 Boot EPROM Source Code Listing (continued)

```
;       STI     R1,*+AR0(0)             ; Set register
        LDI     @LMST,AR0               ; Load internal memory test parms
        LDI     @LMLEN,AR1              ;
        CALL    mt                      ;
        CMPI    0,R5                    ; If R5 is non-zero, test failed
        BZ      CHKBIST                 ;
        LDI     @BIST_RESULTS,R0        ; Load BIST Control word into r0
        OR      4,R0
        STI     R0,@BIST_RESULTS
CHKBIST LDI     @BIST_RESULTS,R0        ; Load BIST Control word into r0
        CMPI    0,R0
        BZ      READGD
READBD  LDI     @BISTCNTL,AR1           ; Load BIST Control word into r0
        LDI     *AR1,R0
        OR      @BIST_RESULTS,R0        ; Set DPSRAM 0 Failed
        STI     R0,*AR1
       B        SETTMR                  ;
READGD
       LDI      @C31REG,AR0             ;
       LDI      0001H,R0                ; LED bit
       LDI      *+AR0(0),R1             ; Get register setting
       OR       R0,R1                   ; Turn Off Failed LED bit
       STI      R1,*+AR0(0)             ; Set register
SETTMR
       LDI      @BOOTP,R0               ; Timer period Pass
       LDI      @CTRL,AR0               ; Point to control register
       STI      R0,*+AR0(028H)          ;
       LDI      03C0H,R0                ; Start timer
       STI      R0,*+AR0(020H)          ;
*
*       STI     R0,*+AR2(8)             ;
*       LDI     1,R0                    ;
*       STI     R0,*+AR2(4)             ; Set ready flag
*
*       IACK    *+AR2(0)                ; Reset pld
*
       IACK     *+AR2(0)                ; Reset pld
       CALL     CKBOOT                  ; See if boot from flash/global set
       LDI      1,R0                    ;
       STI      R0,*+AR2(4)             ; Set ready flag
       LDI      0,IF                    ; Clear any pending interrupts
       LDI      1800H,ST                ; Clear and enable cache and
                                  ; disable OVM
        LDI     @BISTCNTL,AR1           ; Load BIST Control word into r0
        LDI     *AR1,R0
        ANDN    3,R0                    ; Clear BIST ACTIVE and BIST Enable
        STI     R0,*AR1
* Enable INT1 and TIMER0
       LDI      @IIEMASK,IE             ; Enable interrupts
       OR       2000H,ST                ; Global interrupt enable
*
```

A.1 Boot EPROM Source Code Listing (continued)

```
WAIT
       IDLE                            ;
       B       WAIT                    ;
*
*
NOTHIN
       RETI                    ;
TIME0           ; Timer 0 interrupt processing
       PUSH    ST                      ;
       PUSH    R0                      ;
       PUSHF   R0                      ;
       PUSH    R1                      ;
       PUSHF   R1                      ;
       PUSH    AR0                     ;
       PUSH    DP                      ;
       LDI     080h,DP                 ; Point the DP register to page 80h
* Toggle front panel LED
       LDI     @C31REG,AR0             ;
       LDI     0200H,R0                ; LED bit
       LDI     *+AR0(0),R1             ; Get register setting
       XOR     R1,R0                   ; Flip LED bit
       STI     R0,*+AR0(0)             ; Set register
       POP     DP                      ;
       POP     AR0                     ;
       POPF    R1                      ;
       POP     R1                      ;
       POPF    R0                      ;
       POP     R0                      ;
       POP     ST                      ;
       RETI                            ;
*
TIME1           ; Timer 1 interrupt processing
       RETI                            ;
*
ISR0            ; INT0-
       RETI                    ;
ISR1            ; INT1- Mix interrupt processing
       PUSH    ST                      ;
       PUSH    R0                      ;
       PUSHF   R0                      ;
       PUSH    AR0                     ;
       PUSH    AR1                     ;
       PUSH    AR2                     ;
       PUSH    DP                      ;
       LDI     080h,DP                 ; Point the DP register to page 80h
       LDI     @PROC_ADDR,AR2          ; Processor base address
       LDI     *+AR2(0),R0             ; Get address from HOST
       CALLU   R0                      ;
RETURN
       LDI     @C31REG,AR0             ; Point to IO registers
       LDI     1,R0                    ;
       STI     R0,*+AR0(4)             ; Clear VME interrupt
```

A.1 Boot EPROM Source Code Listing (continued)

```
        LDI     1,R0                    ;
        STI     R0,*+AR2(4)             ; Set ready flag to Host
        LDI     0,IF                    ; Clear any extra interrupts
        POP     DP                      ;
        POP     AR2                     ;
        POP     AR1                     ;
        POP     AR0                     ;
        POPF    R0                      ;
        POP     R0                      ;
        POP     ST                      ;
        RETI                            ;
LOOK
        LDI     *+AR2(1),AR0            ; Get address from 0x201
        LDI     *AR0,R0                 ; Read word
        STI     R0,*+AR2(3)             ; Store into 0x203
        RETSU                           ;
SET
        LDI     *+AR2(1),AR0            ; Get address from 0x201
        LDI     *+AR2(2),R0             ; Get word to write
        STI     R0,*AR0                 ; Store at pointer
        RETSU                           ;
*
BLKMOV
        LDI     *+AR2(1),AR0            ; Get src address from 0x201
        LDI     *+AR2(2),AR1            ; Get destination address
        LDI     *+AR2(3),RC             ; Get count
        RPTB    BLKMVL                  ;
        LDI     *AR0++,R0               ;
BLKMVL
        STI     R0,*AR1++               ;
*
*
NOPRTN
        RETSU                           ;
CKBOOT
        LDI     @GBOOTL,AR1             ; Check for boot from global first
        LDI     *AR1,R0                 ;
        CMPI    @FLASHFG,R0             ;
        BEQ     CKBOOTG                 ; Boot from global
        CMPI    @FLASHNB,R0             ; See if bypass flash boot set
        BEQ     CKBOOTX                 ; Yes
        B       CKBOOTF                 ; Check for boot from flash
CKBOOTG
        LDI     @GBOOTJ,AR1             ;
        LDI     *AR1,R0                 ;
        STI     R0,@FBOOTT              ;
        LDI     @FBOOTT,AR1             ;
        B       FBOOTE                  ; Do boot
```

A.1 Boot EPROM Source Code Listing (continued)

```
CKBOOTF
       LDI     @FBOOTL,AR1             ;
       LDI     *AR1,R0
       CMPI    @FLASHFG,R0             ;
       BNE     CKBOOTX                 ;
       LDI     @FBOOTJ,AR1             ;
       LDI     *AR1,R0
       STI     R0,@FBOOTT              ;
       CALL    FLASHI                  ;
       LDI     @FBOOTT,AR1             ;
       B       FBOOTE                  ; Do boot
CKBOOTX
       RETSU
FBOOT           ; BOOT function
       LDI     *+AR2(1),AR1            ; Get BOOT address from 0x201
FBOOTE
       LDI    0,R0               ; set start address flag off
       LDI    *AR1++(1),R1       ; load eprom mem. width
       LDI    @JUMPW,AR3         ; full-word size subroutine address -> AR3
       LSH    26,R1              ; test bit 5 of mem. width word
       BN     load0              ; if '1' start PGM loading (32 bits width)
       NOP    *AR1++(1)          ; jump last half word from mem. word
       LDI    @JUMPH,AR3         ; half-word size subroutine address -> AR3
       LSH    1,R1               ; test bit 4 of mem. width word
       BN     load0              ; if '1' start PGM loading (16 bits width)
       LDI    @JUMPB,AR3         ; byte size subroutine address -> AR3
       ADDI   2,AR1              ; jump last 2 bytes from mem. word
load0
       CALLU  AR3                ; load new word according to mem. width
       STI    R1,*+AR0(64h)      ; set primary bus control
load2
       CALLU  AR3                ; load new word according to mem. width
       LDI    R1,RC              ; set block size for repeat loop
       CMPI   0,RC               ; if 0 block size start PGM
       BZ     AR2
       SUBI   1,RC               ; block size -1
       CALLU  AR3                ; load new word according to mem. width
       LDI    R1,AR4             ; set destination address
       LDI    R0,R0              ; test start address loaded flag
       LDIZ   R1,AR2             ; load start address if flag off
       LDI    -1,R0              ; set start & dest. address flag on
       SUBI   1,AR3              ; sub address with loop
       CALLU  AR3                ; load new word according to mem. width
       LDI    1,R0               ; set dest. address flag off
       ADDI   1,AR3              ; sub address without loop
       BR     load2              ; jump to load a new block when loop
loop_h
       RPTB   load_h             ; PGM load loop
```

A.1 Boot EPROM Source Code Listing (continued)

```
sub_h
       LDI    *AR1++(1),R1       ; load LSB half-word
       AND    0FFFFh,R1
       LDI    *AR1++(1),R2       ; load MSB half-word
       LSH    16,R2
       OR     R2,R1              ; R1 = a new 32 bits word
       LDI    R0,R0              ; test load address flag
       BNN    end_h
load_h
       STI    R1,*AR4++(1)       ; store new word to dest. address
end_h
       RETSU                     ; return from subroutine
loop_w
       RPTB   load_w             ; PGM load loop
sub_w
       LDI    *AR1++(1),R1       ; read a new 32 bits word
       LDI    R0,R0              ; test load address flag
       BNN    end_w
load_w
       STI    R1,*AR4++(1)       ; store new word to dest. address
end_w
       RETSU                     ; return from subroutine
loop_b
       RPTB   load_b             ; PGM load loop
sub_b
       LDI    *AR1++(1),R1
       AND    0FFh,R1            ; load 1st byte ( LSB )
       LDI    *AR1++(1),R2
       AND    0FFh,R2
       LSH    8,R2
       OR     R2,R1              ; load 2nd byte
       LDI    *AR1++(1),R2
       AND    0FFh,R2
       LSH    16,R2
       OR     R2,R1              ; load 3rd byte
       LDI    *AR1++(1),R2       ; load 4th byte ( MSB )
       LSH    24,R2
       OR     R2,R1              ; R1 = a new 32 bits word
       LDI    R0,R0              ; test load address flag
       BNN    end_b
load_b
       STI    R1,*AR4++(1)       ; store new word to dest. address
end_b
       RETSU                     ; return from subroutine
FLOOK          ; FLASH LOOK function
       CALL    FLASHI                  ;
       LDI     *+AR2(1),AR0            ; Get address from 0x201
       LDI     *AR0,R0                 ; Read word
       STI     R0,*+AR2(3)             ; Store into 0x203
       CALL    FLASHU                  ;
       RETSU                           ;
FSET           ; FLASH SET function
```

A.1 Boot EPROM Source Code Listing (continued)

```
        CALL    FLASHI                  ;
        LDI     @FSEQ55,AR3             ; Write command
        LDI     @FAA,R0                 ;
        STI     R0,*AR3                 ;
        LDI     @FSEQAA,AR3             ;
        LDI     @F55,R0                 ;
        STI     R0,*AR3                 ;
        LDI     @FSEQ55,AR3             ;
        LDI     @FSEQWRT,R0             ;
        STI     R0,*AR3                 ;
        LDI     *+AR2(1),AR3            ; Get address from 0x201
        LDI     *+AR2(2),R0             ; Get word to write
        STI     R0,*AR3                 ; Store at pointer
        AND     @F80,R0                 ;
FSETL
        LDI     *AR3,R1                 ; Wait till write complete
        AND     @F80,R1                 ;
        CMPI    R0,R1                   ;
        BNE     FSETL                   ;
        CALL    FLASHU                  ;
        RETSU                           ;
FLDB            ; FLASH LOAD BYTE function
        CALL    FLASHI                  ;
        LDI     *+AR2(1),AR0            ; Get src address from 0x201
        LDI     *+AR2(2),AR1            ; Get destination address
        LDI     *+AR2(3),RC             ; Get count
        RPTB    FLDBL                   ;
        LDI     *AR0++,R0               ;
        CALL    WRITEFLASH              ;
FLDBL
        NOP     *AR1++                  ;
        CALL    FLASHU                  ;
        RETSU                           ;
FUNLDB          ; FLASH UNLOAD BYTE function
        CALL    FLASHI                  ;
        LDI     *+AR2(1),AR0            ; Get src address from 0x201
        LDI     *+AR2(2),AR1            ; Get destination address
        LDI     *+AR2(3),RC             ; Get count
        RPTB    FUNLDBL                 ;
        LDI     *AR0++,R0               ;
FUNLDBL
        STI     R0,*AR1++               ;
        CALL    FLASHU                  ;
        RETSU                           ;
FLDW            ; FLASH LOAD WORD function
        CALL    FLASHI                  ;
        LDI     *+AR2(1),AR0            ; Get src address from 0x201
        LDI     *+AR2(2),AR1            ; Get destination address
        LDI     *+AR2(3),RC             ; Get count
        RPTB    FLDWL                   ;
        LDI     *AR0++,R0               ;
        CALL    WRITEFLASH              ;
```

A.1 Boot EPROM Source Code Listing (continued)

```
FLDWL
       NOP     *AR1++                  ;
       CALL    FLASHU                  ;
       RETSU                           ;
FUNLDW          ; FLASH UNLOAD WORD function
       CALL    FLASHI                  ;
       LDI     *+AR2(1),AR0            ; Get src address from 0x201
       LDI     *+AR2(2),AR1            ; Get destination address
       LDI     *+AR2(3),RC             ; Get count
       RPTB    FUNLDWL                 ;
       LDI     0,R0                    ;
       LDI     *AR0++,R1               ;
       AND     0FFh,R1                 ;
       OR      R1,R0                   ;
       LDI     *AR0++,R1               ;
       AND     0FFh,R1                 ;
       LSH     8,R1                    ;
       OR      R1,R0                   ;
       LDI     *AR0++,R1               ;
       AND     0FFh,R1                 ;
       LSH     16,R1                   ;
       OR      R1,R0                   ;
       LDI     *AR0++,R1               ;
       AND     0FFh,R1                 ;
       LSH     24,R1                   ;
       OR      R1,R0                   ;
FUNLDWL
       STI     R0,*AR1++               ;
       CALL    FLASHU                  ;
       RETSU                           ;
ERASE           ; FLASH ERASE function
       CALL    FLASHI                  ;
       LDI     @FSEQ55,AR3             ; Erase Chip command
       LDI     @FAA,R0           ;
       STI     R0,*AR3                 ;
       LDI     @FSEQAA,AR3             ;
       LDI     @F55,R0           ;
       STI     R0,*AR3                 ;
       LDI     @FSEQ55,AR3             ;
       LDI     @FSEQER1,R0             ;
       STI     R0,*AR3                 ;
       LDI     @FSEQ55,AR3             ;
       LDI     @FAA,R0           ;
       STI     R0,*AR3                 ;
       LDI     @FSEQAA,AR3             ;
       LDI     @F55,R0           ;
       STI     R0,*AR3                 ;
       LDI     @FSEQ55,AR3             ;
       LDI     @FSEQER2,R0             ;
       STI     R0,*AR3                 ;
ERASEL
       LDI     *AR3,R1                 ; Wait till ERASE complete
```

A.1 Boot EPROM Source Code Listing (continued)

```
        AND     @F80,R1          ;
        CMPI    @F80,R1          ;
        BNZ     ERASEL                  ;
        CALL    FLASHU                  ;
        RETSU                           ;
SERASE          ; FLASH SECTOR ERASE function
        CALL    FLASHI                  ;
        LDI     @FSEQ55,AR3             ; Erase Chip command
        LDI     @FAA,R0          ;
        STI     R0,*AR3                 ;
        LDI     @FSEQAA,AR3             ;
        LDI     @F55,R0          ;
        STI     R0,*AR3                 ;
        LDI     @FSEQ55,AR3             ;
        LDI     @FSEQER1,R0             ;
        STI     R0,*AR3                 ;
        LDI     @FSEQ55,AR3             ;
        LDI     @FAA,R0          ;
        STI     R0,*AR3                 ;
        LDI     @FSEQAA,AR3             ;
        LDI     @F55,R0          ;
        STI     R0,*AR3                 ;
        LDI     *+AR2(1),AR3            ; Get address from 0x201
        LDI     @FSEQER3,R0             ;
        STI     R0,*AR3                 ;
SERASEL
        LDI     *AR3,R1                 ; Wait till ERASE complete
        AND     @F80,R1          ;
        CMPI    @F80,R1
        BNZ     SERASEL                 ;
        CALL    FLASHU                  ;
        RETSU                           ;
FLASHI
        LDI     @CTRL,AR0               ; Point to control register
        LDI     @FLASHINT,R0            ; Init local memory interface control
        STI     R0,*+AR0(4)             ;
        CALL    RFLASH                  ; Reset FLASH
        RETSU                           ;
FLASHU
        LDI     @CTRL,AR0               ; Point to control register
        LDI     @LOCALINT,R0            ; Init local memory interface control
        STI     R0,*+AR0(4)             ;
        CALL    RFLASH                  ; Reset FLASH
        RETSU                           ;
RFLASH
        LDI     @FSEQ55,AR3             ; Read/Reset command
        LDI     @FAA,R0                 ;
        STI     R0,*AR3                 ;
        LDI     @FSEQAA,AR3             ;
        LDI     @F55,R0                 ;
        STI     R0,*AR3                 ;
        LDI     @FSEQ55,AR3             ;
```

A.1 Boot EPROM Source Code Listing (continued)

```
       LDI     @FSEQRST,R0             ;
       STI     R0,*AR3                 ;
       LDI     @FBASE,AR3              ;
       LDI     *AR3,R0                 ;
       RETSU                           ;
WRITEFLASH
; Enter with address in AR1 and data in R0
       PUSH    R0                      ;
       LDI     @FSEQ55,AR3             ; Write command
       LDI     @FAA,R1                 ;
       STI     R1,*AR3                 ;
       LDI     @FSEQAA,AR3             ;
       LDI     @F55,R1                 ;
       STI     R1,*AR3                 ;
       LDI     @FSEQ55,AR3             ;
       LDI     @FSEQWRT,R1             ;
       STI     R1,*AR3                 ;
       STI     R0,*AR1                 ; Store data at destination
       AND     @F80,R0                 ;
FWRTL
       LDI     *AR1,R1                 ; Wait till write complete
       AND     @F80,R1                 ;
       CMPI    R0,R1                   ;
       BNE     FWRTL                   ;
       POP     R0                      ;
       RETSU                           ;
;
;       TITLE 'Memory Test Diagnostic'
;
;*********************************************
;* ENTER with ar0 = start address             *
;*            ar1 = test length               *
;*                                            *
;* On ERROR condition                         *
;*          ar6 = contains address of failure *
;*          r0  = contains good word          *
;*          r2  = contains bad word           *
;*          r5  = contains error code         *
;*********************************************
pptrs     .word  tp_st
pptre     .word  tp_end
staddr    .word  00100000h
tlen      .word  020000h
mt
        ldpk    02fh            ;
        sti     ar0,@staddr     ; Store start address
        sti     ar1,@tlen       ; Store test length
        ldi     @pptrs,ar4      ; Load current pattern address
        ldi     @pptre,ar5      ; Load last pattern address
        ldi     @staddr,ar6     ; Load start address
        ldi     @tlen,ar2       ; Load test length
        nop     *ar2--          ; -1 to length
```

A.1 Boot EPROM Source Code Listing (continued)

```
; Forward moving inversions memory test
        ldi     *ar4++,r7       ; Load memory test area with current pattern
        ldi     *ar4,r6         ; Load next test pattern
        rpts    ar2             ;
        sti     r7,*ar6++       ; Repeat fill memory
mt1
        ldi     @staddr,ar6     ; Load start address
        ldi     ar2,rc          ; Load length
        rptb    mt1l            ;
        ldi     *ar6++,r1       ; Get test pattern
        cmpi    r7,r1           ; Compare memory to pattern
        bnz     mt1er1          ;
mt1l
        nop                     ;
mt2
        ldi     @staddr,ar6     ; Load start address
        ldi     ar2,rc          ; Load length
        rptb    mt2l            ;
        ldi     *ar6,r1         ; Get previous test pattern
        cmpi    r7,r1           ; Check previous pattern
        bnz     mt2er1          ;
        sti     r6,*ar6        ; Store next pattern
        ldi     *ar6++,r1       ;
        cmpi    r6,r1          ; Verify that it wrote correctly
        bnz     mt2er2          ;
mt2l
        nop
        cmpi    ar4,ar5         ; See if all patterns tested
        bz      mt3             ;
        ldi     *ar4++,r7       ; Load memory test area with current pattern
        ldi     *ar4,r6        ; Load next test pattern
        b       mt1             ; Continue testing
mt3
        ldi     @pptrs,ar4      ; Load current pattern address
        ldi     @pptre,ar5      ; Load last pattern address
        ldi     @staddr,r0      ; Load start address
        ldi     @tlen,r1        ; Load test length
        ldi     @tlen,ar2       ; Load test length
        nop     *ar2--          ; -1 to length
        addi    r0,r1           ;
        subi    1,r1            ;
        sti     r1,@staddr      ;
        ldi     @staddr,ar6     ; Load start address
; Backward moving inversions memory test
        ldi     *ar4++,r7       ; Load memory test area with current pattern
        ldi     *ar4,r6        ; Load next test pattern
        rpts    ar2             ;
        sti     r7,*ar6--       ; Repeat fill memory
mt4
        ldi     @staddr,ar6     ; Load start address
        ldi     ar2,rc          ; Load length
        rptb    mt4l            ;
```

A.1 Boot EPROM Source Code Listing (continued)

```
        ldi     *ar6--,r1       ; Get test pattern
        cmpi    r7,r1           ; Compare memory to pattern
        bnz     mt4er1          ;
mt4l
        nop                     ;
mt5
        ldi     @staddr,ar6     ; Load start address
        ldi     ar2,rc          ; Load length
        rptb    mt5l            ;
        ldi     *ar6,r1         ; Get previous test pattern
        cmpi    r7,r1           ; Check previous pattern
        bnz     mt5er1          ;
        sti     r6,*ar6        ; Store next pattern
        ldi     *ar6--,r1       ;
        cmpi    r6,r1          ; Verify that it wrote correctly
        bnz     mt5er2          ;
mt5l
        nop
        cmpi    ar4,ar5         ; See if all patterns tested
        bz      mt_end          ;
        ldi     *ar4++,r7       ; Load memory test area with current pattern
        ldi     *ar4,r6        ; Load next test pattern
        b       mt5             ; Continue testing
mt_end
        ldi     0,r5            ; Return test passed code
        b       mt_rtn          ;
; ar6 contains address of failure
; r7 contains good word
; r1 contains bad word
mt1er1
        ldi     0e1h,r5         ;
        b       mt_rtn          ;
mt4er1
        ldi     0e4h,r5         ;
        b       mt_rtn          ;
; ar6 contains address of failure
; r6 contains good word
; r1 contains bad word
mt2er1
        ldi     0e2h,r5         ;
        b       mt_rtn          ;
mt5er1
        ldi     0e5h,r5         ;
        b       mt_rtn          ;
; ar6 contains address of failure
; r6 contains good word
; r1 contains bad word
mt2er2
        ldi     0e3h,r5         ;
        b       mt_rtn          ;
```

A.1 Boot EPROM Source Code Listing (continued)

```
mt5er2
        ldi     0e6h,r5         ;
        b       mt_rtn          ;
mt_rtn                          ; b       mt_rtn
        bz      mt_rtnx         ; See if error is set
        sti     r5,@errcode     ;
        sti     ar6,@erradr     ;
        sti     r7,@err7        ;
        sti     r1,@err1        ;
        sti     r6,@err6        ;
        b       mt_rtnx         ; Bypass halt
mt_rtnx
        rets                    ;
ISR2                                    ; INT2-
       RETI                             ;
ISR3                                    ; INT3-
       RETI                             ;
tp_st     .word  000000000h
          .word  0ffffffffh
          .word  000000000h
          .word  055555555h
          .word  0aaaaaaaah
          .word  00000000fh
          .word  0000000f0h
          .word  000000f00h
          .word  00000f000h
          .word  0000f0000h
          .word  000f00000h
          .word  00f000000h
          .word  0f0000000h
          .word  000000000h
          .word  0ff000000h
          .word  000ff0000h
          .word  00000ff00h
          .word  0000000ffh
tp_end    .word  000000000h
       .end
```

This page is intentionally blank

Appendix B: RACEway Interlink Modules

B.1 Overview

To join the RACEway busses of two or more boards, a backplane circuit board containing RACEway data switches and sockets that mate with the 64 pins of the P2 connector must be installed. These assemblies, called RACEway Interlink Modules (ILK), come in sizes that bridge 4, 8, 12, and 16 VMEbus slots, with combinations that can bridge up to 20 slots. These modules are available from Pentek as Model 8250, options –004 through –016.

The RACEway switches on the ILKs are called RACEway Crossbar switches. Each crossbar switch connects six separate RACEway buses, automatically routing the data packets from one bus to any other, based on the routing header in the packet. The ILKs support multiple 160 MB/sec RACEway transfers simultaneously, depending on the number of slots and crossbar switches utilized.

B.2 RACEway ILK1

A RACEway ILK1 is used to add a single VMEbus slot connection to the RACEway fabric. This type if Interlink is useful when, for example, you have five VMEbus slots that need to be connected to the RACEway fabric. For this example, a single ILK4 and a single ILK1 can be used to join the five VMEbus slots. Two single ILK1 modules can be used to connect two VMEbus slots to the RACEway fabric. However ILK1 modules cannot be used to bridge two larger ILK modules.

B.3 RACEway ILK 4 Example

The RACEway ILK4 connects four VMEbus slots to the RACEway fabric. Figure B–1 shows the ILK4 module and the crossbar ports on it.

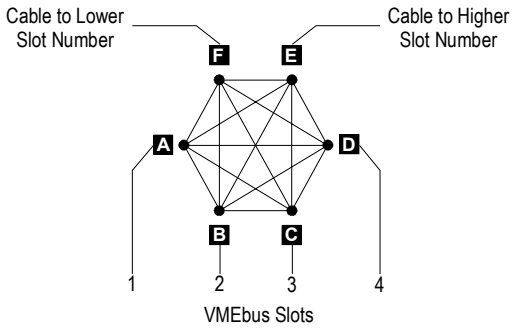


Figure B–1: RACEway ILK4 Example

B.4 RACEway ILK 8 Example

The RACEway ILK 8 modules allows up to eight VMEbus slots to be connected to the RACEway fabric. Figure B–2 shows one possible topology for an ILK8 module. Interlink manufacturers can use a variety of topologies, so check with the ILK manufacturer to determine the topology used in your Interlink.

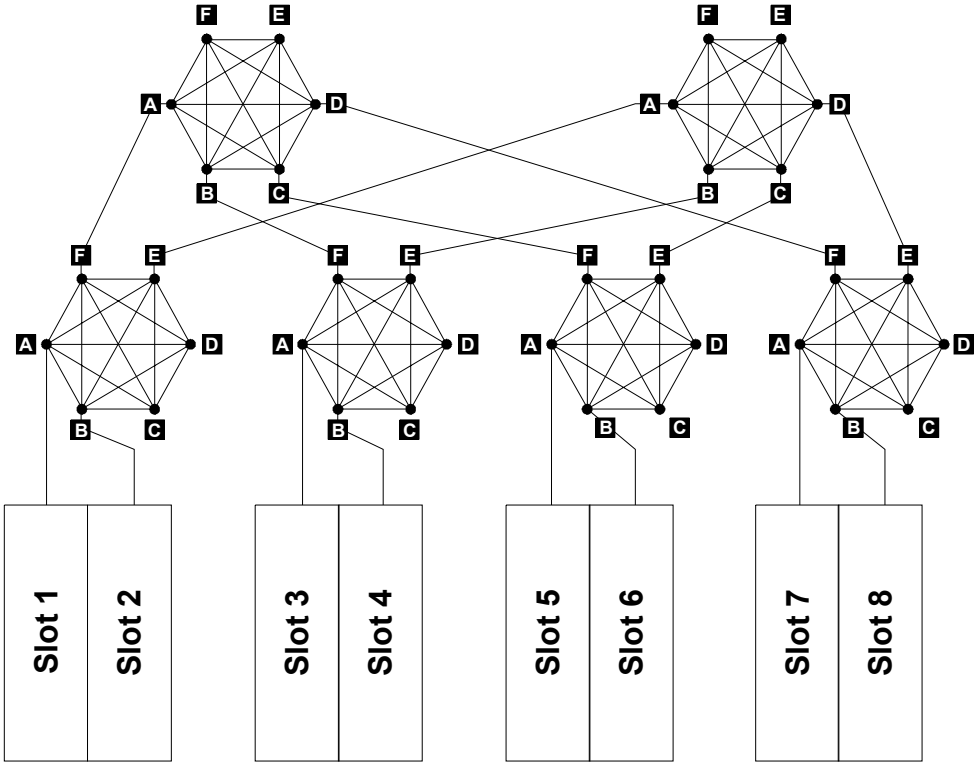


Figure B–2: RACEway ILK8 Topology Example

B.4 RACEway ILK 8 Example (continued)

B.4.1 ILK8 Example RACEway Transaction Routing Codes

Table B–1, below, shows an example of two different RACEway transaction routing codes. One transaction is from VMEbus slot 1 to VMEbus slot 8, the other is from VMEbus slot 3 to slot 4. Both transactions are single point–to–point transactions.

Table B–1: ILK8 Routing Code Examples

Transaction 1: VMEbus Slot 1 to VMEbus Slot 8							
Bit#	**D31–D29**	**D28–D26**	**D25–D23**	**D22–D20**	**D19–D17**	**D16–D14**	**D13–D11**
Bit Name	Route 0	Route 1	Route 2	Route 3	Route 4	Route 5	Route 6
Value	**010**	**100**	**110**	**000**	**000**	**000**	**000**
Route	A to F	A to D	F to B	Not Used	Not Used	Not Used	Not Used
Bit#	**D10–D09**	**D07–D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Route 7	Route 8	Broadcast Accept Code		Routing Priority		Mode Select
Value	**000**	**000**	**xx**		**xx**		**0**
Route	Not Used	Not Used					
Transaction 2: VMEbus Slot 3 to VMEbus Slot 4							
Bit#	**D31–D29**	**D28–D26**	**D25–D23**	**D22–D20**	**D19–D17**	**D16–D14**	**D13–D11**
Bit Name	Route 0	Route 1	Route 2	Route 3	Route 4	Route 5	Route 6
Value	**110**	**000**	**000**	**000**	**000**	**000**	**000**
Route	A to B	Not Used	Not Used	Not Used	Not Used	Not Used	Not Used
Bit#	**D10–D09**	**D07–D05**	**D04**	**D03**	**D01**	**D01**	**D00**
Bit Name	Route 7	Route 8	Broadcast Accept Code		Routing Priority		Mode Select
Value	**000**	**000**	**xx**		**xx**		**0**
Route	Not Used	Not Used					
All '**x**' values are determined by the programmer							

B.5 RACEway ILK12 Example

The RACEway ILK12 modules allow up to 12 VMEbus slots to be connected to the RACEway fabric. Figure B–3, below illustrates one possible RACEway ILK12 topology. Check with your ILK manufacturer to determine the RACEway topology used in your Interlink.

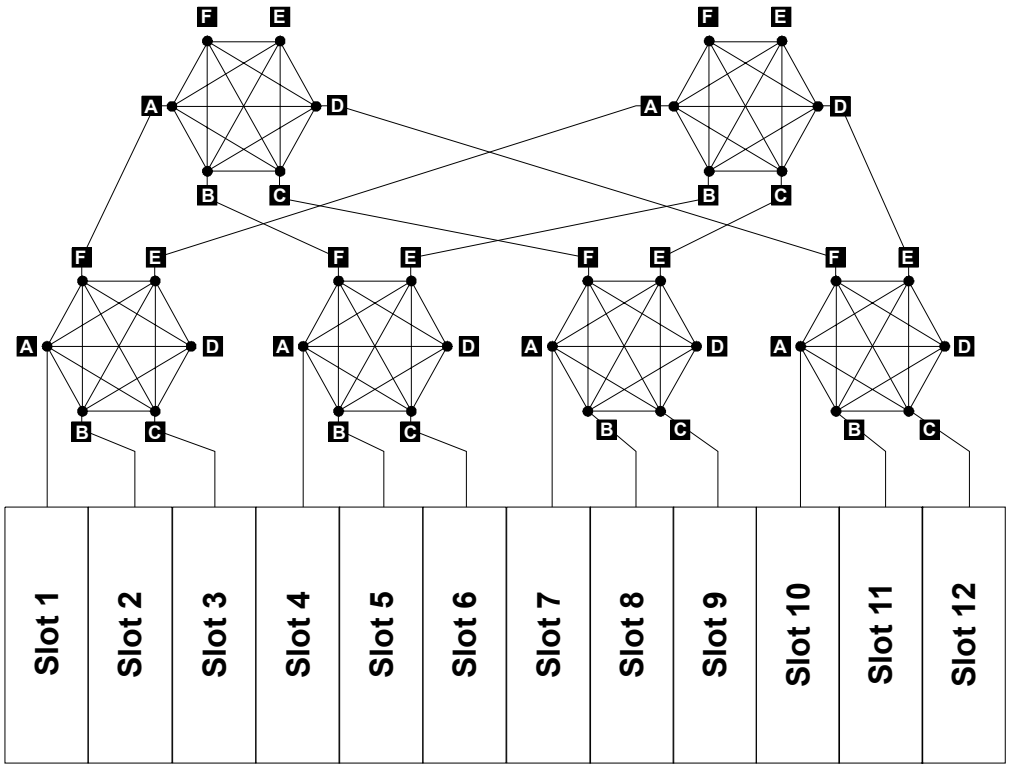


Figure B–3: RACEway ILK12 Topology

B.6 RACEway ILK16 Example

The RACEway ILK16 modules allow up to 16 VMEbus slots to be connected to the RACEway fabric. Figure B–4, below illustrates one possible RACEway ILK16 topology. Check with your ILK manufacturer to determine the RACEway topology used in your Interlink.

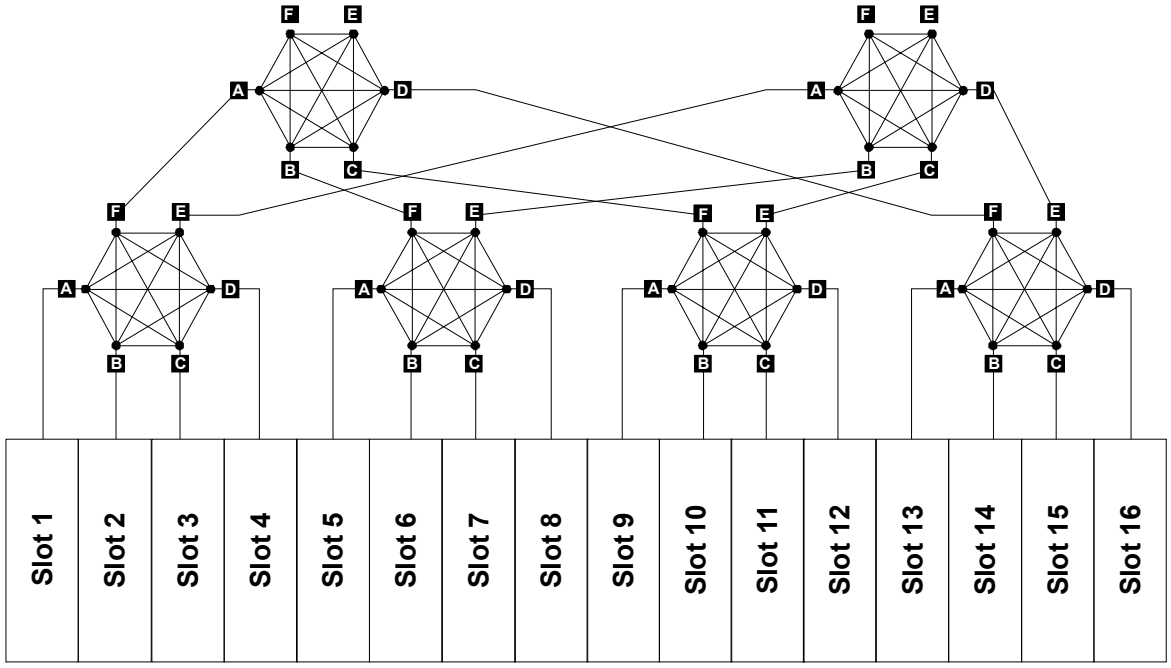


Figure B–4: RACEway ILK16 Topology

This page is intentionally blank

Appendix C: Graychip 4014

C.1 Introduction

The following pages are a reprint of the Graychip 4014 Quad Narrowband Digital Receiver Data Sheet.

The chip is configured by writing to eight–bit control registers, as described in the attached data sheet.

This page is intentionally blank

GRAYCHIP
DSP CHIPS AND SYSTEMS
(650) 323-2955
FAX (650) 323-0206
2185 Park Blvd., Palo Alto, CA 94306

GC4014

QUAD RECEIVER CHIP

DATASHEET

April 27, 1999

Rev 0.5

REVISION HISTORY

Revision	Date	Description
0.0	1 Dec. 1997	Original
0.1	14 Jan 1998	Page iii, added mask revision table to list of tables Pages 8, last line, added non-symmetry mode errata Page 10, footnote, changed “1 milliseconds” to “10 microseconds”. Page 11, modified Figure 10 (c) Page 12, first line Section 3.7, changed “output” to “input”. Page 18, address 1, bit 6, added non-symmetry errata Page 19, address 4, bits 4-7, corrected bit assignments Pages 20-21, corrected address numbering for blanking and flush registers Page 22, address 12, bits 0,1, changed “12 LSBs” to “14 LSBs”. Page 24, address 16, added mask revision table. Throughout, corrected table and figure reference
0.2	21 Jan 1998	Page 30, Changed data setup time from 4 to 2 ns. Page 8, 18 Non-symmetry mode works for parts marked with all mask codes except 55532B Page 24, Changed Table 5 to add mask code 55532C Page 28, Table 7 min and max recommended Vcc changed to 3.1 to 3.5 volts. Page 28, Table 7 max junction temperature changed to 125C.
0.3	5 Feb 1998	Page 38, Changed 800 to 8000 in Table 17. Page 38, Changed E5 to EA in note 1, Table 16. Page 37, Checksum for test4 changed to D2.
0.4	23 Apr 1999	Page 38, Changed CS to CE in the control interface timing description, Table 10 Pages 9,19, Changed GAIN equation from “NARROW*1.97” to “NARROW*0.97 + 1" Page 28, Table 7, Changed max Vup voltage to 5.5v. Page 39, New gain application note. Pages 5, 25, Positive frequency to downconvert. Pages 36, 37, changed 26 -> 2A for address 00 Page 29, Table 9, changed V_{IH} for CK, CK2X to 2.4V from 2.0V. Page 29, Table 9, changed $I_{OH/L}$ to +/- 4mA from 2.0mA Page 30, F_{CK} changed to 64MHz, clock to output changed to 20ns.
0.5	27 Apr 1999	Pages 9,19,33,35,39, changed gain equation G/64 -> G/32

CONTACTING GRAYCHIP

CORPORATE OFFICES:

GRAYCHIP, Inc.

2185 Park Blvd.

Palo Alto, CA 94306

PHONE:

(650) 323-2955

FAX:

(650) 323-0206

WEB PAGE:

www.graychip.com

E-MAIL:

sales@graychip.com

tech-support@graychip.com

LIST OF FIGURES

LIST OF TABLES

GC4014 DATASHEET

1.0 KEY FEATURES

- Input rates up to 64 MSPS
- Four real input down-convert channels or
  Two complex input downconvert channels
- Independent tuning frequencies
- Independent phase/gain controls
- 4 by 4 14 bit Input Crossbar switch or
  3 by 4 16 bit Input Crossbar switch
- Decimation factors of
  16 to 32,768 in the real output mode
  32 to 65,536 in the complex output mode
- Zero padding for lower decimation factors
- Outputs can be either:
  bit serial,
  nibble serial (link port)
  or memory mapped
- Output summing for beamforming
- 8 to 16 bit output samples
- 0.02 Hz tuning resolution
- 0.14 dB gain resolution
- Less than 0.05 dB peak to peak passband ripple
- Greater than 100 dB far image rejection
- Greater than 95 dB spur free dynamic range
- User programmable 63 tap output filter
- Nyquist filtering for QPSK or QAM symbol data
- Meets GSM, AMPS and DAMPS Cellular specifications
- Microprocessor interface for control, output, and diagnostics
- Built in diagnostics
- Microprocessor interface will accept either 3.3 or 5 volt input levels
- 250 mW per channel at 50 MHz, 3.3 volts
- 100 pin thin QFP package

2.0 BLOCK DIAGRAM

A block diagram illustrating the major functions of the chip is shown in Figure 1.

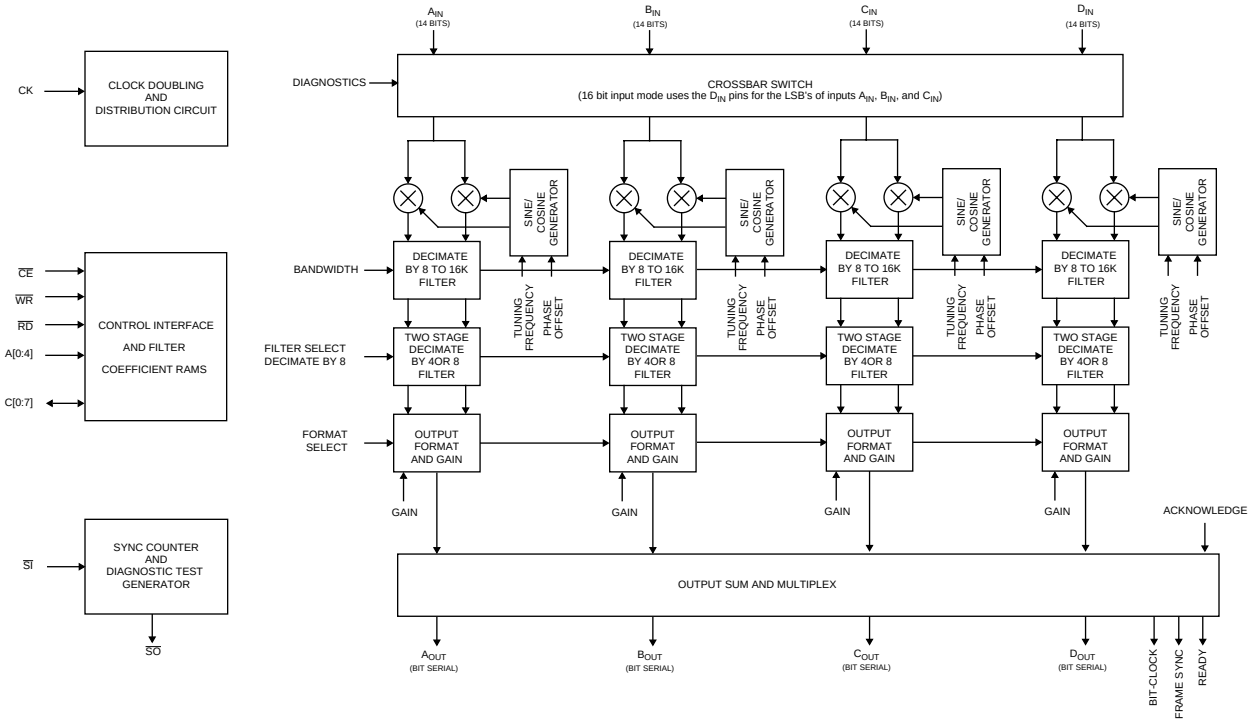


Figure 1. GC4014 Block Diagram

3.0 FUNCTIONAL DESCRIPTION

The GC4014 quad receive chip contains four identical down-conversion circuits. Each down-convert circuit accepts a real sample rate up to 62.5 MHz, down converts a selected carrier frequency to zero, decimates the signal rate by a programmable factor ranging from 16 to 32768 (32 to 65,536 for complex outputs), and optionally sums it with other down converted samples. The chip outputs the four down-converted signals, or their sum. The chip contains a user programmable output filter which can be used to arbitrarily shape the received data's spectrum. This filter can be used as a Nyquist receive filter for digital data transmission.

Two down-converter paths can be merged to be used as a single complex input down-conversion circuit.

The down-converters are designed to maintain over 95 dB of spur free dynamic range and over 100 dB of out of band rejection. Each down-convert circuit accepts 16 bit inputs and produces 16 bit outputs (bit serial). The frequencies and phase offsets of the four sine/cosine sequence generators can be independently specified, as can the gain of each circuit. The down converters share the same bandwidth, filter coefficients and input formats. A special mode allows the downconverters to support GSM and DAMPS blocker requirements (see Sections 7.5 and 7.6).

On chip diagnostic circuits are provided to simplify system debug and maintenance.

The chip receives configuration and control information over a microprocessor compatible bus consisting of an 8 bit data I/O port, a 5 bit address port, a chip enable strobe, a read strobe and a write strobe. The chip's control registers (8 bits each) are memory mapped into the 5 bit address space of the control port.

Section 7.5 Describes a typical application, including control register values and the proper sequence of operations required to use the chip.

3.1 CONTROL INTERFACE

The chip is configured by writing control information into sixty four control registers within the chip. The contents of these control registers and how to use them are described in Section 5. The registers are written to or read from using the **C[0:7]** , **A[0:4]** , $\overline{\textbf{CE}}$, $\overline{\textbf{RD}}$ and $\overline{\textbf{WR}}$ pins. Each control register has been assigned a unique address within the chip. This interface is designed to allow the GC4014 to appear to an external processor as a memory mapped peripheral (the pin $\overline{\textbf{RD}}$ is equivalent to a memory chip's $\overline{\textbf{OE}}$ pin).

An external processor (a microprocessor, computer, or DSP chip) can write into a register by setting **A[0:4]** to the desired register address, selecting the chip using the $\overline{\textbf{CE}}$ pin, setting **C[0:7]** to the desired value and then pulsing $\overline{\textbf{WR}}$ low. The data will be written into the selected register when both $\overline{\textbf{WR}}$ and $\overline{\textbf{CE}}$ are low and will be held when either signal goes high.

To read from a control register the processor must set **A[0:4]** to the desired address, select the chip with the $\overline{\textbf{CE}}$ pin, and then set $\overline{\textbf{RD}}$ low. The chip will then drive **C[0:7]** with the contents of the selected register. After

the processor has read the value from **C[0:7]** it should set $\overline{\textbf{RD}}$ and $\overline{\textbf{CE}}$ high. The **C[0:7]** pins are turned off (high impedance) whenever $\overline{\textbf{CE}}$ or $\overline{\textbf{RD}}$ are high or when $\overline{\textbf{WR}}$ is low. The chip will only drive these pins when both $\overline{\textbf{CE}}$ and $\overline{\textbf{RD}}$ are low and $\overline{\textbf{WR}}$ is high.

One can also ground the $\overline{\textbf{RD}}$ pin and use the $\overline{\textbf{WR}}$ pin as a read/write direction control and use the $\overline{\textbf{CE}}$ pin as a control I/O strobe. Figure 2 shows timing diagrams illustrating both I/O modes.

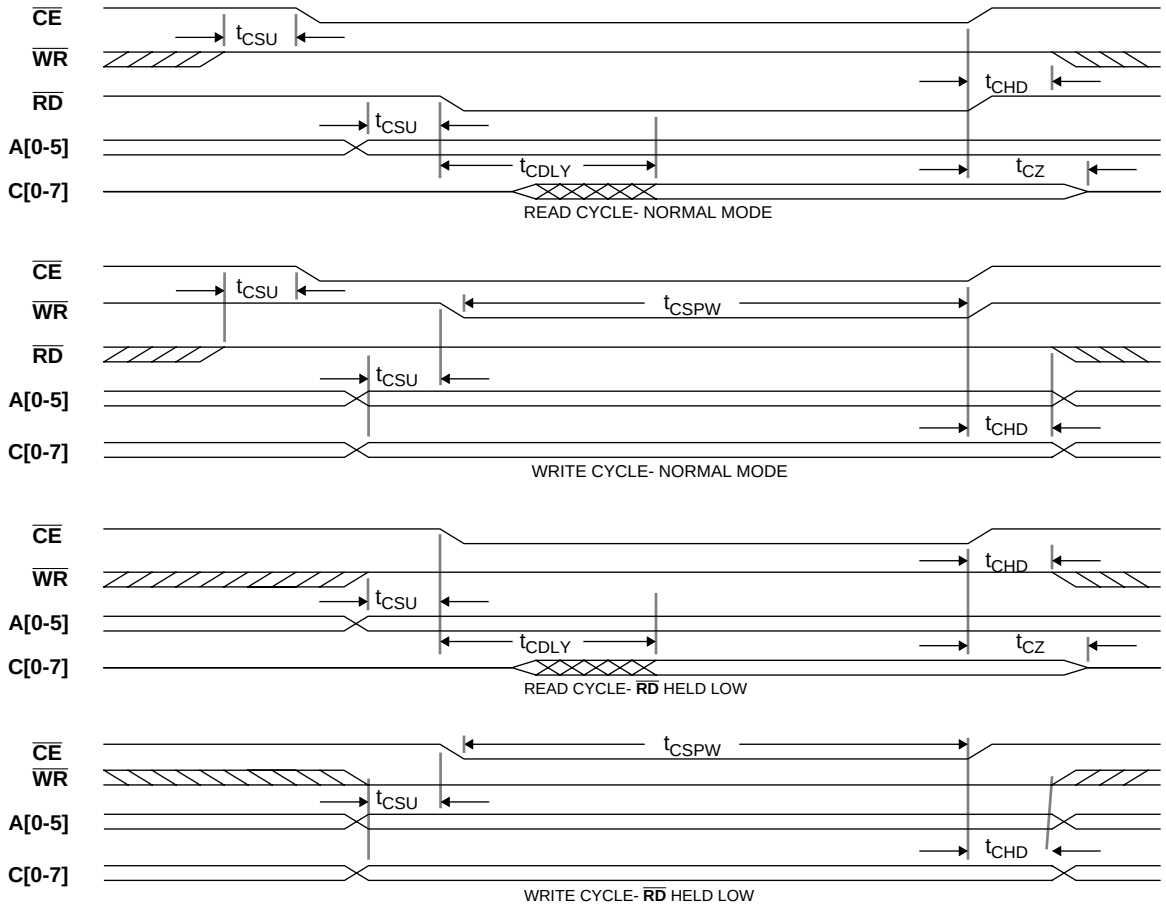


Figure 2. Control I/O Timing

The setup, hold and pulse width requirements for control read or write operations are given in Section 6.0.

The **C**, **A**, $\overline{\textbf{WR}}$, $\overline{\textbf{RD}}$ and $\overline{\textbf{CE}}$ pins will accept either 5 volt or 3.3 volt input levels. A separate power supply voltage pin ($\mathbf{V_{UP}}$) is provided on the chip to enable this feature.

3.2 INPUT FORMAT

Both 14 bit and 16 bit input formats are accepted. In the 14 bit mode the inputs are 14 bit samples from four different sources. In the 16 bit mode, the inputs are 16 bit samples from three different sources. In either case, a crossbar switch allows the user to route any input to any down-converter channel. The input samples are normally clocked into the chip at the clock rate, i.e., the input sample rate is equal to the clock rate. Input rates lower than the clock rate can be accepted by using the zero pad mode. The zero pad mode will insert up to 15 zeroes between

each input sample, allowing input data rates down to 1/16th the clock rate. Zero padding also lowers the effective decimation ratio. For example, the minimum decimation is normally factor of 32. If the input data rate is 5 MSPS and the chip can be clocked at 40 MHz, then the zero pad function can be used to pad the 5 MSPS input data up by a factor of 8 to 40 MSPS. The minimum decimation of 32, once the zero padding is done, becomes a minimum decimation of 4 relative to the original 5 MSPS data.

3.3 THE DOWN CONVERTERS

Each down converter uses an NCO and mixer to quadrature down convert a signal to baseband and then uses a 4 stage CIC[1] filter and a two-stage decimate by 4 or 8 filter to lowpass filter and to isolate the desired signal. A block diagram of each filter is shown below:

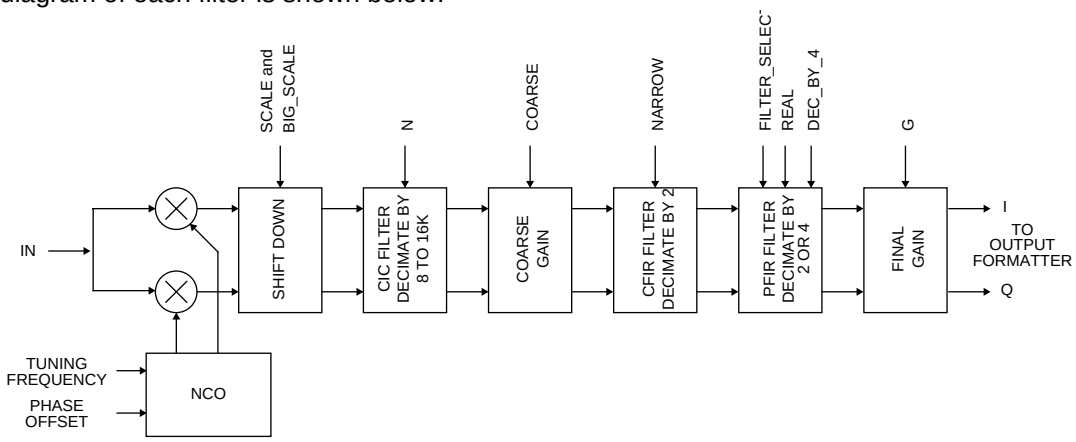


Figure 3. The Down Converter Channel

The CIC filter reduces the sample rate by a programmable factor ranging from 8 to 16,384. The CIC outputs are followed by a coarse gain stage and then followed by a two stage decimate by 4 or 8 filter. The coarse gain circuit allows the user to boost the gain of weak signals up to 42 dB in 6 dB steps. The first stage of the two stage filter is a compensating 21 tap decimate by 2 filter (CFIR) with a choice of two sets of fixed tap weights. The first set is designed to be flat from $-0.5F_S$ to $+0.5F_S$, where F_S is the output sample rate, and to reject out of band energy by at least 80dB. The second set has a narrower output passband ($-0.25F_S$ to $+0.25F_S$), but more out of band rejection. The second set is ideal for systems such as GSM, which require more far band rejection (>97 dB), but with relaxed adjacent band rejection. The second stage is a 63 tap decimate by 2 or 4 programmable filter (PFIR) with either internal or user supplied tap weights. The internal filter is designed to be flat from $-0.4F_S$ to $+0.4F_S$ of the output sample rate and to reject out of band energy by at least 85 dB. The user can also design and download their own final filter to customize the channel's spectral response. Typical uses of the programmable filter include matched (root-raised cosine) filtering, or filtering to generate oversampled outputs with greater out of band rejection. The 63 tap symmetrical filter is downloaded into the chip as 32 words, 16 bits each. The programmable PFIR coefficients must be used to bandlimit the output in the decimate by 4 mode.

1. Hogenauer, Eugene B., An Economical Class of Digital Filters for Decimation and Interpolation, IEEE transactions on Acoustics, Speech and Signal Processing, April 1981.

The PFIR will also, if desired, convert the complex output data to real. The complex to real conversion also doubles the output sample rate so that the PFIR decimation is 1 or 2 in the real mode.

The PFIR filter is followed by a gain and output format circuit. The gain circuit allows the user to add an additional 18 dB of gain in 0.14 dB steps. The output format circuit can also delete every other sample without filtering. Anti-aliasing filtering must have already been performed (in the second stage filter). This is useful to achieve deeper far-band rejection since the stopband performance of the CIC filter is a function of the decimation that follows it.

3.3.1 The Numerically Controlled Oscillator (NCO)

The tuning frequency of each down converter is specified as a 32 bit word and the phase offset is specified as a 16 bit word. The NCOs can be synchronized with NCOs on other chips. This allows multiple down converter outputs to be coherently combined, each with a unique phase and amplitude. A block diagram of the NCO circuit is shown in Figure 4.

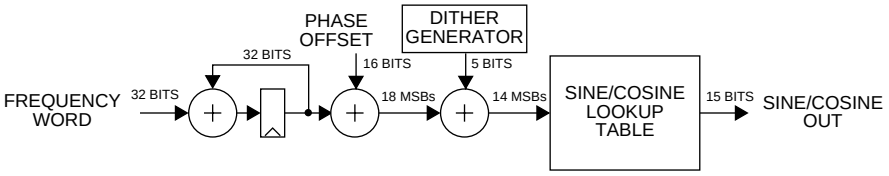
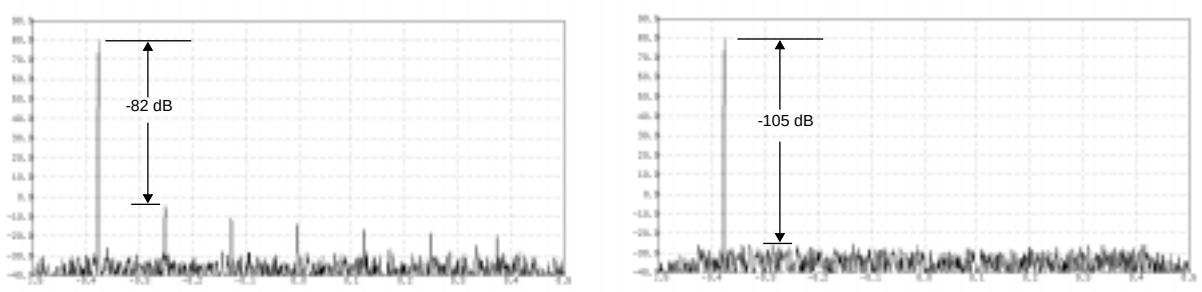


Figure 4. NCO Circuit

The tuning frequency is set to FREQ according to the formula FREQ = $2^{32}F/F_{CK}$, where F is the desired tuning frequency and F_{CK} is the chip's clock rate. The 16 bit phase offset setting is PHASE = $2^{16}P/2\pi$, where P is the desired phase in radians ranging between 0 and 2π. Note that a positive tuning frequency is used to downconvert the signal. A negative tuning frequency can be used to upconvert the negative image of a real signal (inverting the spectrum).

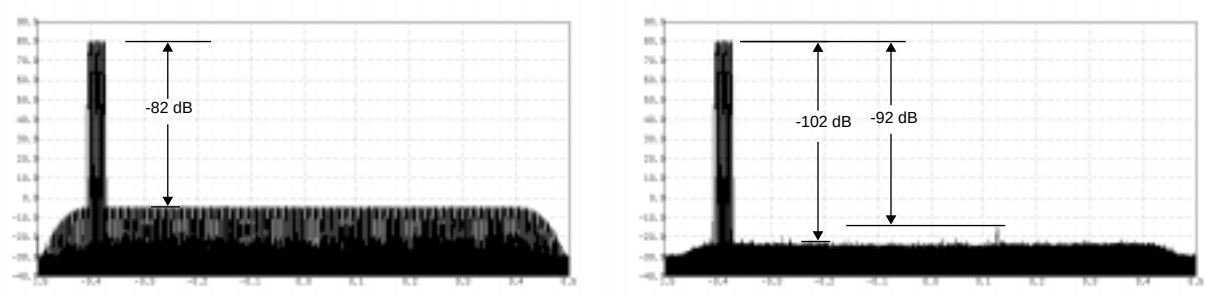
The NCO's spur level is reduced to below -92 dBc through the use of phase dithering. The spectrums in Figure 5 show the NCO spurs for an example tuning frequency before and after dithering has been turned on. Notice that the spur level decreases from -82 dB to -105 dB.



(a) Before Dithering (b) After Dithering

Figure 5. Example NCO Spurs

Figure 6 shows the maximum spur levels as the tuning frequency is scanned over a portion of the frequency range with the peak hold function of the spectrum analyzer turned on. Notice that the peak spur level is -82 dB before dithering and is between -92 and -102 after dithering has been turned on.



(a) Before Dithering (b) After Dithering

Figure 6. NCO Peak Spur Scan Plot

The worst case NCO spurs at -92 dB, such as the one shown in figure 6(b), are due to a few frequencies that are related to the sampling frequency by small rational numbers (for example FREQ = 3/16 * Fck). In these cases the rounding errors in the sine/cosine lookup table repeat in a regular fashion, thereby concentrating the error power into a single frequency, rather than spreading it across the spectrum. These worst case spurs can be eliminated by selecting an initial phase that minimizes the errors or by changing the tuning frequency by a small amount (50 Hz). All spurs can be made to fall below -96 dB with the selection of a proper initial phase or tuning frequency.

3.3.2 Four Stage CIC Filter

The mixer outputs are decimated by a factor of N in a four stage CIC filter, where N is any integer between 8 and 16,384. The programmable decimation allows the chip's usable output bandwidth to range from less than a kilo-Hertz to 1.5 MHz when the input rate (which is equal to the chip's clock rate) is 62.5 MHz. A block diagram of the CIC filter is shown in Figure 7.

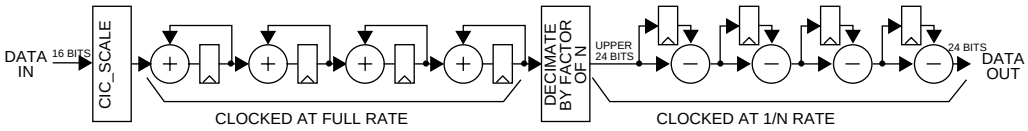


Figure 7. Four Stage CIC Decimate by N Filter

The CIC filter has a gain equal to N^4 which must be compensated for in the "CIC_SCALE" circuit shown in Figure 7. This circuit has a gain equal to $2^{(SCALE+6*BIG\_SCALE-55)}$, where SCALE ranges from 0 to 5 and BIG_SCALE ranges from 0 to 7. The gain of the CIC circuit is equal to: $GAIN = N^4 2^{(SCALE + 6 \times BIG\_SCALE - 56)}$. The user must select values for SCALE and BIG_SCALE such that GAIN is less than one, i.e., SCALE and BIG_SCALE must be selected such that: $(SCALE + 6 \times BIG\_SCALE) \le (56 - 4\log_2 N)$. Overflows due to improper gain settings will go undetected if this relationship is violated. For example, if N is equal to 8, then this restriction means that BIG_SCALE and SCALE

should be less than or equal to 7 and 2 respectively. The BIG_SCALE and SCALE settings are common to all channels.

3.3.3 Coarse Channel Gain

The gain of each channel can be boosted up to 42 dB by shifting the output of the CIC filter up by 0 to 7 bits prior to rounding it to 16 bits. The coarse gain is: $\mathrm{GAIN} = 2^{\mathrm{COARSE}}$, where COARSE ranges from 0 to 7. Overflows in the coarse gain circuit are saturated to plus or minus full scale. The coarse gain is used to increase the gain of an individual signal after the input bandwidth of the downconverter has been reduced by a factor of N in the CIC filter. If the signal power across the input bandwidth is relatively flat, as is the case in most frequency division multiplexed (FDM) systems, then one would want to boost the signal power out of the CIC filter by a factor of $\mathrm{GAIN} = \sqrt{\mathrm{N}}$. Each channel can be given its own coarse gain setting. Note that the final gain stage described in Section 3.4 can boost the overall gain by up to 24 more dB.

3.3.4 The Compensating Decimate By Two Filter (CFIR)

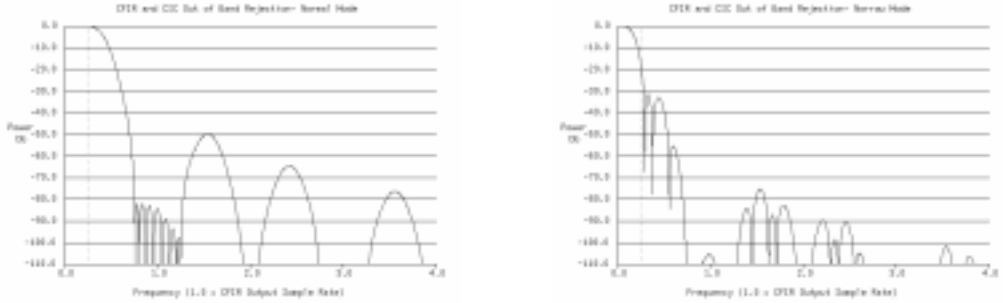
The CIC/Coarse gain outputs are filtered by two stages of filtering. The first stage is a 21 tap decimate by 2 filter with two sets of fixed coefficients. The first set of coefficients is used in the normal mode to give a passband which is flat (0.01 dB ripple) over 100% of the final output bandwidth and which has 85 dB of out of band rejection. The filter also compensates for the droop associated with the CIC programmable decimation filter. The filter is symmetric with the following taps:

```
29, -85, -308, -56, 1068, 1405, -2056, -6009, 1303, 21121, 32703
```

The narrow set of coefficients are intended for applications that need deeper stop bands or need oversampled outputs. These requirement are common in cellular systems where out of band rejection requirements can exceed 100 dB. The filter coefficients for the narrow mode are:

```
-98, -679, -2016, -3234, -2537, 850, 6053, 12060, 18230, 23239, 25212
```

The combined frequency response of the CIC and CFIR filter for both modes is shown below:



(a) Normal Mode (b) Narrow Mode

Figure 8. Combined CFIR and CIC Frequency Response

The dashed vertical line in the plots shows the output Nyquist rate for the chip when the PFIR is in the decimate by 2 mode. The narrow mode filter introduces a gain of 1.97 (5.9 dB).

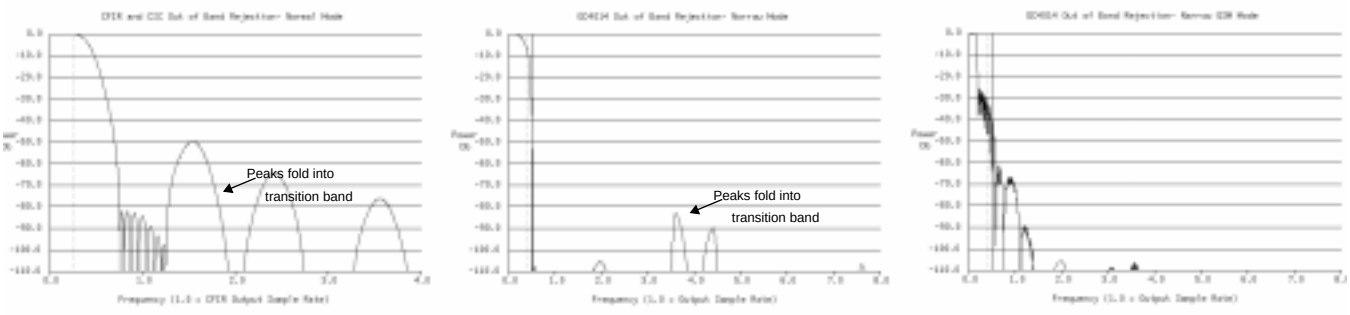
3.3.5 The Programmable Final Filter (PFIR)

The second stage decimate by two or four filter uses either internal ROM based coefficients, or externally downloaded filter coefficients. The internal 80% bandwidth filter has 80 dB of out of band image rejection and 0.03 dB peak to peak passband ripple. The internal filter is a 63 tap symmetric filter. The 32 unique coefficients are:

```
-14, -20, 19, 73, 43, -70, -82, 84, 171, -49, -269, -34, 374, 192, -449,
-430, 460, 751, -357, -1144, 81, 1581, 443, -2026, -1337, 2437, 2886,
-2770, -6127, 2987, 20544, 29647
```

Figure 9(a) shows the overall response for the internal PFIR when the CFIR is in the normal mode. Figure 9(b) shows the overall response when the CFIR is in the narrow mode. Note that the peaks in the stop band at 3.5 times the output sample rate will, after decimation, fold into the transition band from 0.4 to 0.5 of the output sample rate. This out of band power, if necessary, can be filtered out by either using a custom PFIR filter with a narrower passband, or by post-filtering.

An overall response using custom coefficients suitable to meet the stringent GSM Cellular requirements is shown in Figure 9(c). See Sections 7.5 and 7.6 for more details on GSM and DAMPS configurations.



(a) Normal CFIR Mode (b) Narrow CFIR Mode (c) GSM Coefficients

Figure 9. Overall CIC-CFIR-PFIR Response

The externally downloaded coefficients can be used to tailor the spectral response to the user's needs. For example, it can be programmed as a Nyquist (typically a root-raised-cosine) filter for matched filtering digital data. The user downloaded filter coefficients are 16 bit 2's complement numbers. Unity gain will be achieved through the filter if the sum of the 63 coefficients is equal to 65536. If the sum is not 65536, then the PFIR will introduce a gain equal to (sum of coefficients)/65536.

The 63 coefficients are identified as coefficients h_0 through h_{62}, where h_{31} is the center tap. The coefficients are assumed to be symmetric, so only the first 32 coefficients (h_0 through h_{31}) are loaded into the chip. A non-symmetric mode allows the user to download a 32 tap non-symmetric filter as taps h_0 through h_{31}. ERRATA: The non-symmetry mode does not work properly for parts marked with mask code 55532B, Contact GRAYCHIP for details.

3.3.6 Real Mode

The PFIR will output either complex or real data. Complex data is output at a rate equal to $F_{CK}/(4N)$ or $F_{CK}/(8N)$ in the decimate by 4 mode. If the output samples are real, then the filter translates the output spectrum up by multiplying the filtered data by the complex sequence +1, -j, -1, +j, …, and then outputting the real part at a rate equal to $F_{CK}/2N$. The real output mode can be used to create double sided signals out of single sideband data. The real outputs are packed into complex words for output. The first sample of a real pair is put into the I-half and the second is put into the Q-half. Note that the decimate by 4 mode is invalid in the real mode.

3.4 FINAL GAIN ADJUSTMENT

The final gain of each channel is adjusted by multiplying each output sample by G/32, where G is a 10bit 2's complement gain word. Since G can range between -512 and +511, the gain adjustment will range from -16.0 to +15.98. Setting G to zero clears the channel. This provides a final gain adjustment range from -∞ to +24 dB in approximately 0.28 dB steps. A different gain can be specified for each channel. Note that the overall gain of the chip is also a function of the amount of decimation programmed into the chip (N), the scale circuit setting in the CIC filter, the coarse gain setting, the narrow mode in the CFIR, and the sum of the PFIR coefficients. The overall gain is shown below where the first term in braces is fixed for all four channels and must be less than or equal to unity. The terms in square brackets can be different for each channel. NARROW is "1" in the narrow CFIR mode, "0" otherwise. See Section 7.9 for a discussion on properly setting the chip's gain.

$$\text{GAIN} = \{N^4 2^{(\text{SCALE} + 6 \times \text{BIG\_SCALE} - 56)}\}[\,2^{\text{COARSE}}]\{1.0 + \text{NARROW} \times 0.97\}\{\frac{\text{PFIR\_SUM}}{65536}\}[\frac{G}{32}]$$

3.5 SUMMATION MODES

The chip can be programmed to output the four individual channels, the sum of pairs of channels, or the sum of all four channels.These modes are used to process four real input signals, two complex input signals, or one beamformed signal. When processing two complex input signals, the I inputs are put in channels A and C, and the Q inputs are put in channels B and D. The summation mode then adds channels A and B together and channels C and D together.

Summation is disabled in Sum Mode 0. In Sum Mode 1 the channel A output is replaced by the sum of channels A and B, the channel B output is replaced by the sum of channels C and D, and the channel C and D outputs are left alone. In Sum Mode 2 the channel A output becomes the sum of all four channels and channels B, C and D are left alone. These modes are summarized in the following table:

Table 1: Output Summation Modes

SUM MODE	CHANNEL OUTPUT			
	OUT_A	OUT_B	OUT_C	OUT_D
0	CH_A	CH_B	CH_C	CH_D
1	$CH_A + CH_B$	$CH_C + CH_D$	CH_C	CH_D
2	$CH_A + CH_B + CH_C + CH_D$	CH_B	CH_C	CH_D

3.6 OUTPUT MODES

The channel or summation outputs are accessible either through internal control registers, through bit serial outputs, or through nibble serial (link mode) outputs. Note that the bit serial and link mode outputs start, after power up, in a tri-state condition and must be turned on when the chip is configured.

3.6.1 Internal Control Registers

The internal control registers are loaded by the chip once every output sample period (OSP)[1] and held for the rest of the period. The user is notified that new samples are ready and a new OSP has begun, either through an interrupt signal provided by the chip's "READY" pin (RDY/ACK pin), or through a control register bit.

3.6.2 Serial Outputs

The chip provides a bit serial clock (SCK), a frame strobe (SFS) and four data bit lines (SOUT A,B,C and D) to output the data. A MUX_MODE control specifies whether the four data outputs are transmitted on four separate bit-serial pins, or multiplexed onto two, or just one pin in a TDM format. Separate output pins are not provided for the I and Q halves of complex data. The I and Q outputs are always multiplexed onto the same bit-serial pin. The 16 bit I-component is output first, followed by the Q-component. The "packed mode" allows a complex pair to be treated as a single 32 bit word. The "READY" signal is used to identify the first word of a complex pair or of the TDM formatted output. The TDM modes are summarized in the following table (See Table 1 for a definition of OUT in the summation modes):

Table 2: TDM Modes

MUX MODE	SERIAL OUTPUT			
	A_{OUT}	B_{OUT}	C_{OUT}	D_{OUT}
0	OUT_A	OUT_B	OUT_C	OUT_D
1	OUT_A, OUT_B		OUT_C, OUT_D	
2	OUT_A, OUT_B, OUT_C, OUT_D			

The bit serial outputs use the format shown in Figure 10. Figure 10(a) shows the standard output mode (the PACKED mode bit is low). The chip clocks the frame and data out of the chip on the rising edge of SCK (or falling edge if the SCK_POL bit in the input control register is set). The chip sends the 16 bits (I data first) by setting SFS high (or low if SFS_POL in the input control register is set) for one clock cycle, and then transmitting the data, MSB first, on the next 16 clocks. The I/Q data is transmitted "back to back" as shown in Figure 10(a). If the PACKED control bit is high, then the I and Q components are sent as a single 32 bit word with only one SFS strobe as shown in Figure 10(b). If two or more channels are multiplexed out the same serial pins, then the subsequent I/Q channel

1. Output sample period (OSP) refers to the interval between output samples at the decimated output rate. For example, if the input rate (and clock rate) is 10 MHz and the overall decimation factor is 100 (N=25) the OSP will be10 microseconds. An OSP starts when a new sample is ready and stops when the next one is ready.

words will be transmitted immediately following the first I/Q pair as shown in Figure 10(c). Figure 10(c) also shows how the RDY signal can be used to identify the I and Q channels in the TDM serial transmission.The bit-serial output rate is programmable as a power-of-2 division of the input clock.

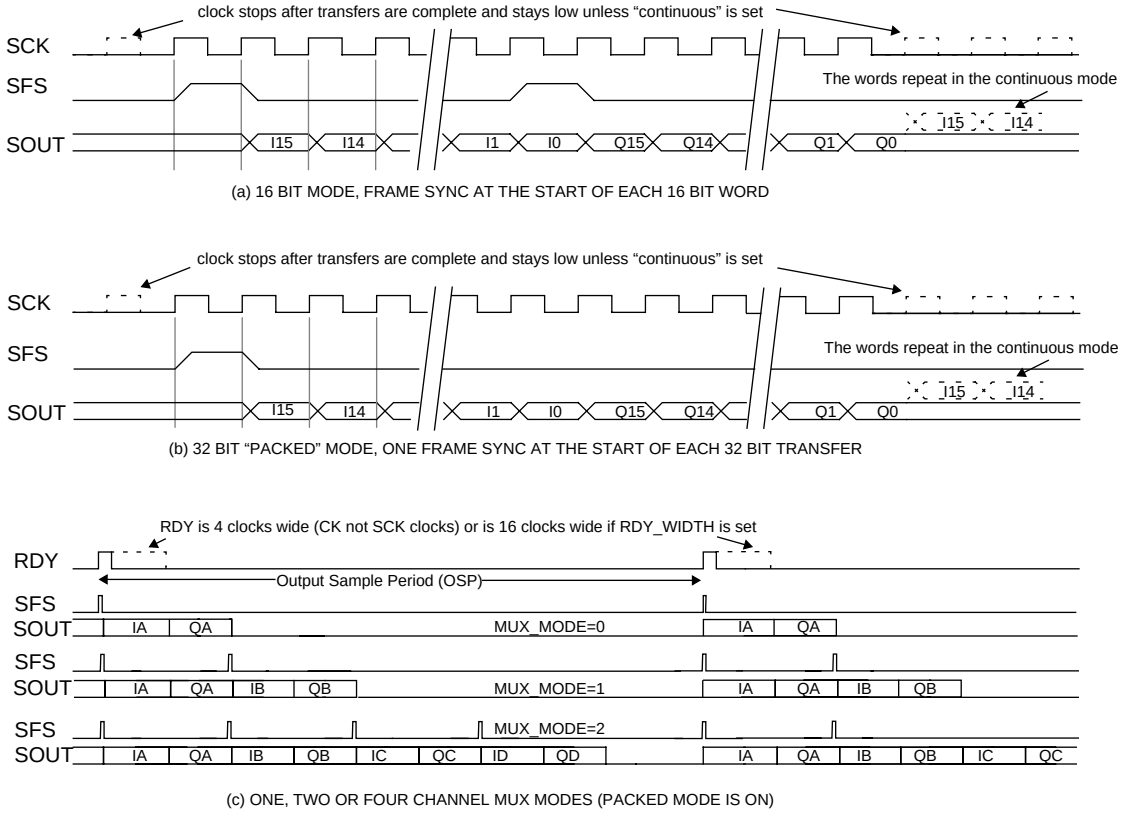


Figure 10. Serial Output Formats

The serial clock (SCK) will normally stop after the last bit transfer of each OSP. The user can force a continuous clock by setting the CONTINUOUS control bit in the output control register. In the continuous mode the data is repeated until the next OSP. This may be useful if the user wants to multiplex the outputs from multiple chips onto the same serial bus. Note: The frame syncs are not intended to be used in the continuous mode. After the proper number of frame syncs have been output as shown in Figure 10, the next frame sync will be missing in the continuous mode. The frame syncs will then repeat every 16 (or 32 in the packed mode) bit clocks. Note also that the number of bit clocks per output frame may not be a rational number, resulting in a truncated bit clock at the end of the OSP.

3.6.3 Link Mode Output

The four serial output pins and the bit clock and frame sync pins can be configured as an ADSP-2106x SHARC DSP chip link port. These pins are in a tri-state condition when the chip powers up. A control bit is set to enable these pins and another control bit is set to enable the link port mode. In the link mode the READY output pin becomes the ACK (acknowledge) input which is used to receive the link port “LACK” signal.

The link port's timing is as follows: The GC4014 checks the state of the ACK pin at the start of an OSP. If ACK is low, the chip does nothing. If ACK is high, then the chip will transmit one, two or four complex words out of the link port. The words are either the channel outputs or the sums of channels depending upon the summation mode settings. The number of outputs transmitted is determined by the MUX_MODE settings. If MUX_MODE is 0, only OUT_A will be transmitted. If MUX_MODE is 1, then OUT_A and OUT_B will be transmitted. If MUX_MODE is 2, then all four will be transmitted. See Table 1 for a definition of OUT in the summation modes.

The data is transmitted in four bit nibbles on the rising edge of the bit clock. The transmission is completed in 8, 16 or 32 clocks to transmit one, two or four complex pairs. If the ACK signal is low at the end of a word (after clocks 8, 16, 24 and 32), then the clock will remain high and the transmission of the next word will be delayed until ACK goes high again. If the ACK signal is low at the start of an OSP, then the transmission will be held off until the next OSP. The clock remains low at the end of the transmission until the next OSP starts. The bit clock rate is programmable as a division of the chip's clock.

3.7 CLOCKING

The chip can be clocked in one of two modes. In the standard mode, the clock rate is equal to the input data rate which can be up to 62.5 MHz. An internal clock doubler doubles the clock rate so that the internal circuitry is clocked at twice the data rate. To use the standard mode the CKMODE pin must be grounded and the internal control register bit EN_DOUBLER must be set high (See Section 5.10).

The alternate clock mode (pin CKMODE is high) accepts a double rate clock on the CK2X pin and bypasses the clock doubler circuit. The EN_DOUBLER control bit should be low. In the alternate mode the user must provide both the standard clock and the double rate clock.

3.8 POWER DOWN MODES

The chip has a power down and keep alive circuit. This circuit contains a slow, nominally 1 KHz, oscillator and a clock-loss detect cell. This circuit is used to detect the loss of clock and provide a slow keep-alive clock to the chip. The circuit is also used to power down the chip by switching from the high speed input clock to the low speed keep-alive clock. The low speed clock rate is slow enough to power down the chip while fast enough to refresh the dynamic nodes within the chip. The user can select whether this circuit is in the automatic clock-loss detect mode, is always on (power down mode), or is disabled (the slow clock never kicks in). The whole chip, or individual down converter channels can be powered down. Using the power down mode for individual channels can save significant power.

3.9 SYNCHRONIZATION

Each GC4014 chip can be synchronized through the use of a sync input signal, an internal one shot sync generator, or a sync counter. Each circuit within the chip, such as the sine/cosine generators or the decimation

control counter can be synchronized to one of these sources. These syncs can also be output from the chip so that multiple chips can be synchronized to the syncs coming from a designated “master” GC4014 chip.

3.10 DATA LATENCY

The latency through the chip, including all pipeline delays and filtering group delays, is shown in the following table (N is the CIC filter’s decimation ratio, see Section 3.5, **$\overline{\text{SI}}$** is the sync input to the chip, **$\overline{\text{SO}}$** is the sync output from the chip, and the **RDY** signal marks the beginning of each output frame, See Figure 10):

Table 3: Latency

FROM INPUT	TO OUTPUT	LATENCY	UNITS	COMMENT
$\overline{\text{SI}}$	**$\overline{\text{SO}}$**	3	Clocks	sync in to sync out, Register settings: OUTPUT_SYNC = 1,SO_INT_MODE =0
$\overline{\text{SI}}$	**RDY**	3.5N+9	Clocks	Sync in to first valid **RDY** out
IN[0:13] at RDY	**OUT** (First)	5	Outputs	Data sample input coincident with **RDY**, to the first output affected by it
IN[0:13] at RDY	**OUT** (Midpoint)	22	Outputs	-to the closest midpoint output affected by **IN**
IN[0:13] at RDY	**OUT** (Last)	37	Outputs	-to the last output affected by **IN**
IN[0:13] at RDY	**OUT** (Step Response)	86N+15	Clocks	Step function delay, step edge is input coincident with **RDY**, to the step edge output

The last entry can be used to identify the group delay through the chip for time tagging events which pass through the chip, where the time tag needs to be accurate to fractions of the output sample. Note that the overall decimation in the complex output mode is one sample every 4N inputs. This means that the step edge will come out 21 samples plus (2N+15) clocks later. A good time tag algorithm would be to count the number of clock cycles between the tagged input sample and the next **RDY** signal (the number D), and then tag the output sample that comes 21 **RDY** signals later with a time tag which is adjusted by (D - 2N-15) clocks. To insure that the adjustment is always positive, one would wait 22 **RDY** signals (22 outputs) and tag the sample with an adjustment of (D+2N-15) clocks. Note that the output sample to be tagged is the sample that is output between the 22nd **RDY** signal and the next **RDY** signal (see Figure 10).

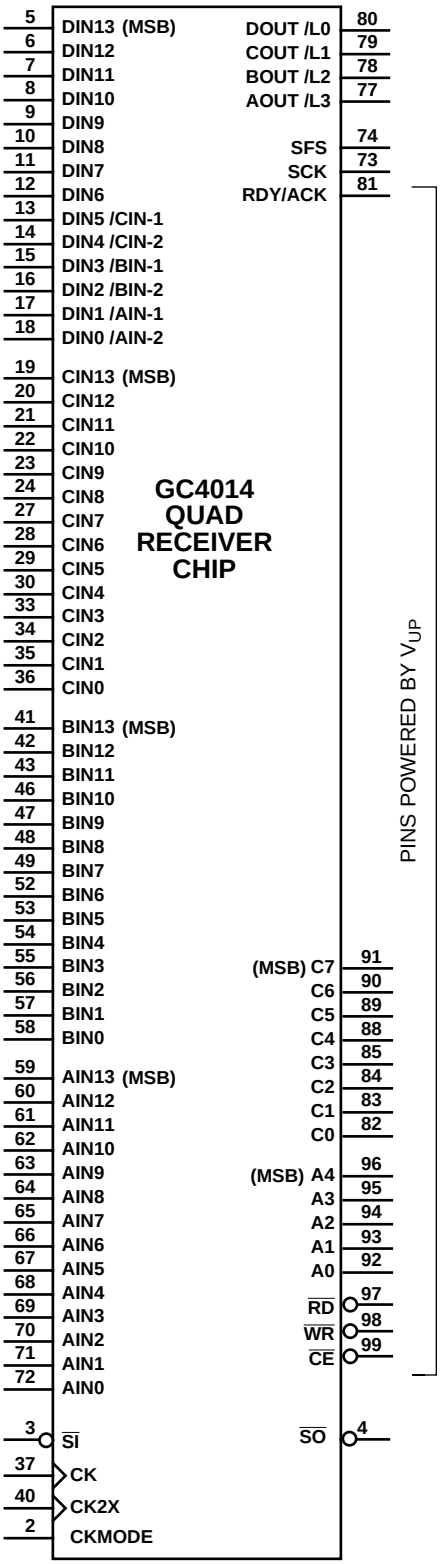
3.11 DIAGNOSTICS

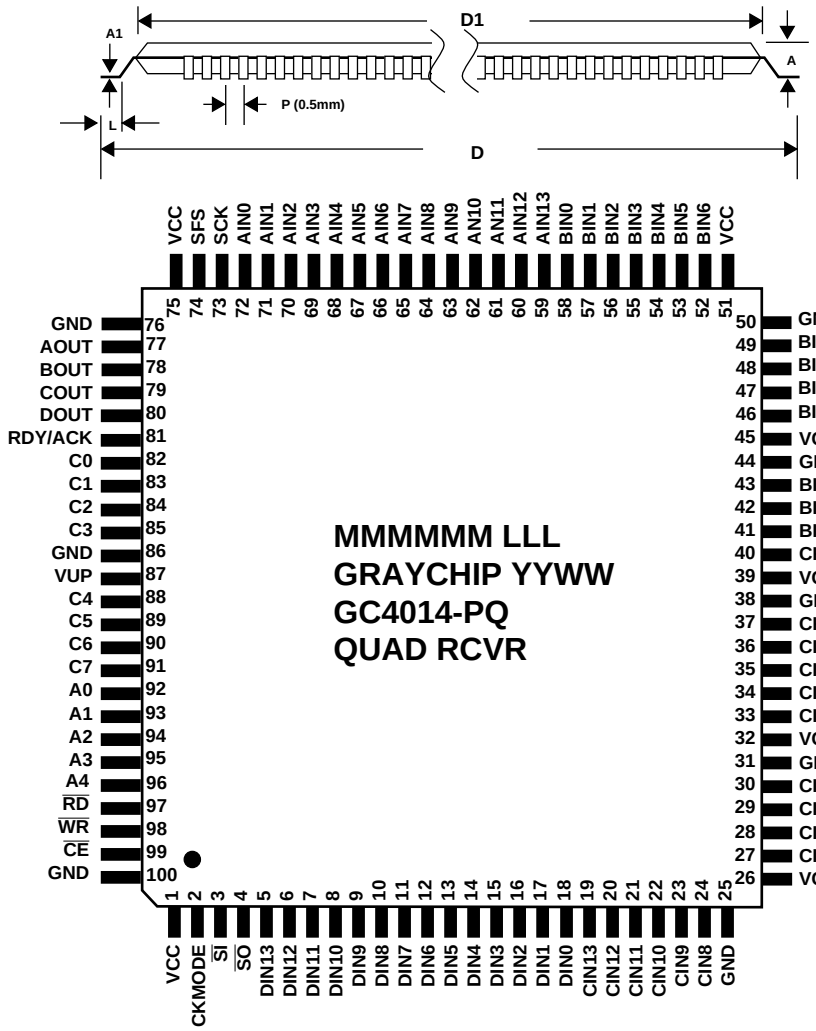
The chip has an internal ramp generator which can be used in place of the data inputs for diagnostics. An internal checksum circuit generates a checksum of the output data to verify the chip’s operation. Section 7.7 gives suggested checksum configurations and their expected checksums.

Besides the internal diagnostics, the chip can support board level testing, an output test configuration which can help initial debug as well as production test is described in Section 7.8.

4.0 PACKAGING

The GC4014 chip comes in a 100 pin thin plastic quad flatpack package





100 PIN THIN QUAD FLAT PACK
GC4014-PQ: PLASTIC PACKAGE
GC4014-CQ: CERAMIC PACKAGE

DIMENSION		PLASTIC	CERAMIC
D	(width pin to pin)	16.0 mm (0.630")	17.2 mm (0.677")
D1	(width body)	14.0 mm (0.551")	14.0 mm (0.551")
P	(pin pitch)	0.5 mm (0.020")	0.5 mm (0.020")
B	(pin width)	0.22 mm (0.009")	0.20 mm (0.008")
L	(leg length)	0.60 mm (0.024")	0.70 mm (0.028")
A	(height)	1.5 mm (0.059")	3.1 mm (0.122")
A1	(pin thickness)	0.15 mm (0.006")	0.2 mm (0.008")

MMMMM = Mask Code
LLL = Lot Code
YYWW = Date Code

VCC PINS: 1, 26, 32, 39, 45, 51, 75

GND PINS: 25, 31, 38, 44, 50, 76, 86,100

VUP PIN: 87

NOTE: 0.01 to 0.1 µf DECOUPLING CAPACITORS SHOULD BE PLACED AS CLOSE AS POSSIBLE TO EACH SIDE OF THE CHIP

SIGNAL	DESCRIPTION
AIN, BIN, CIN, DIN	**INPUT DATA,** *Active high* The 14 bit 2's complement input data for the four channels. The data is clocked into the chip on the rising edge of the clock (**CK**). The LSBs of **DIN** are used as the LSBs of **AIN**, **BIN** and **CIN** in the16 bit input mode (the pins **DIN0/AIN-2**, **DIN2/BIN-2** and **DIN4/CIN-2** are the LSBs of the 16 bit **AIN**, **BIN** and **CIN** inputs).
AOUT, BOUT, COUT, DOUT	**BIT SERIAL OUTPUT DATA,** *Active high* The bit serial output data are transmitted on these pins. In the serial mode these are individual outputs, in the link mode these form a four bit nibble (**DOUT/L0** is the LSB of the Nibble, **AOUT/L3** is the MSB). The output bits are clocked out on the rising edge of **SCK** (falling edge if SCK_POL=1). These pins are tri-stated at power up and are enabled by the OUTPUT_ENABLE control register bit.
SCK	**BIT SERIAL DATA CLOCK***, Active high or low* The serial data bits are clocked out of the chip by this clock. The active edge of the clock is user programmable. This pin is tri-stated at power up and is enabled by the OUTPUT_ENABLE control register bit.
SFS	**BIT SERIAL FRAME STROBE***, Active high or low* The bit serial word strobe. This strobe delineates the 16 or 32 bit words within the bit serial output streams. This strobe is a pulse at the beginning of each bit serial word. The polarity of this signal is user programmable. This pin is tri-stated at power up and is enabled by the OUTPUT_ENABLE control register bit.
RDY/ACK	**READY OR ACKNOWLEDGE FLAG***, programmable active high or low* The chip asserts this signal in the serial output mode to identify the beginning of an output sample period (OSP). The width in input clock cycles and polarity of this signal are user programmable. This signal is typically used as an interrupt to a DSP chip, but can also be used as a start pulse to dedicated circuitry. In the link mode this pin is an input pin and is tied to the LACK handshake output from an ADSP-2106x SHARC DSP link port. This pin is tri-stated at power up and is enabled in the serial mode by the OUTPUT_ENABLE control register bit.
CK	**INPUT CLOCK.** *Active high* The clock input to the chip. The **AIN, BIN, CIN, DIN** and **$\overline{\text{SI}}$** input signals are clocked into the chip on the rising edge of this clock.
CK2X	**DOUBLE RATE INPUT CLOCK.** *Active high* The double rate clock input to the chip. Used in the alternate clock mode to clock the chip. This clock must be exactly twice the frequency of the **CK** clock. Should be grounded in the normal clock mode.
CKMODE	**CLOCK MODE,** *Active high* The clock mode control. The chip uses **CK2X** when this pin is tied high (alternate mode) to clock the internal circuitry. When this signal is grounded (normal mode) the chip doubles the **CK** clock to use as the internal clock.
$\overline{\text{SI}}$	**SYNC IN**. *Active low* The sync input to the chip. All timers, accumulators, and control counters are, or can be, synchronized to **$\overline{\text{SI}}$**. This sync is clocked into the chip on the rising edge of the input clock (**CK**).
$\overline{\text{SO}}$	**SYNC OUT**. *Active low* This signal is either a delayed version of the input sync **$\overline{\text{SI}}$**, the sync counter's terminal count (TC), or a one-shot strobe. The **$\overline{\text{SO}}$** signal is clocked out of the chip on the rising edge of the input clock (**CK**).
C[0:7]	**CONTROL DATA I/O BUS**. *Active high* This is the 8 bit control data I/O bus. Control register data is loaded into the chip or read from the chip through these pins. The chip will only drive these pins when **$\overline{\text{CE}}$** is low and **$\overline{\text{RD}}$** is low and **$\overline{\text{WR}}$** is high.
A[0:4]	**CONTROL ADDRESS BUS**. *Active high* These pins are used to address the control registers within the chip. Each of the control registers within the chip are assigned a unique address. A control register can be written to or read from by setting **A[0:4]** to the register's address.
$\overline{\text{RD}}$	**READ ENABLE**. *Active low* This pin enables the chip to output the contents of the selected register on the **C[0:7]** pins when **$\overline{\text{CE}}$** is also low.
$\overline{\text{WR}}$	**WRITE ENABLE**. *Active low* This pin enables the chip to write the value on the **C[0:7]** pins into the selected register when **$\overline{\text{CE}}$** is also low.
$\overline{\text{CE}}$	**CHIP ENABLE**. *Active low* This control strobe enables the read or write operation. The contents of the register selected by **A[0:4]** will be output on **C[0:7]** when **$\overline{\text{RD}}$** is low and **$\overline{\text{CE}}$** is low. If **$\overline{\text{WR}}$** is low and **$\overline{\text{CE}}$** is low, then the selected register will be loaded with the contents of **C[0:7].**
Vup	**MICROPROCESSOR INTERFACE POWER SUPPLY**. *Power Supply* This pin provides power for the microprocessor interface to allow it to interface to 5 volt logic. Input pins (A[0:4], $\overline{\text{RD}}$,$\overline{\text{WR}}$,$\overline{\text{CE}}$,C[0:7], and RDY/ACK) must not be driven above Vup+0.3V. The output pins (C[0:7], RDY/ACK) will drive a logic one to Vup under no load.

5.0 CONTROL REGISTERS

The chip is configured and controlled through the use of eight bit control registers. These registers are accessed for reading or writing using the control bus pins (**$\overline{CE}$**, **$\overline{RD}$**, **$\overline{WR}$**, **A[0:4]** , and **C[0:7]**) described in the previous section. The register names and their addresses are:

The Mode and Control Registers are addresses 0 to 15

ADDRESS	NAME	ADDRESS	NAME
0	Sync Mode	8	Blank Control
1	Decimation Mode	9	Channel Flush Control
2	Decimation Byte 0	10	Counter Byte 0
3	Decimation Byte 1	11	Counter Byte 1
4	Scale Control	12	Test Mode
5	Channel Gain	13	Page Map
6	Output Format	14	Status
7	Output Mode	15	Checksum

Addresses 16 to 31 are used in sixteen pages as determined by the page select control bits in the page map register. The page assignments are:

PAGE	NAME	PAGE	NAME
0	Channel Outputs	8	Coefficients 0 to 7
1	Keepalive Status	9	Coefficients 8 to 15
2	unused	10	Coefficients 16 to 23
3	unused	11	Coefficients 24 to 31
4	Channel Control A	12	unused
5	Channel Control B	13	unused
6	Channel Control C	14	unused
7	Channel Control D	15	unused

The following sections describe each of these registers. The type of each register bit is either R, W, or R/W indicating whether the bit is read only, write only, or read/write. All bits are active high.

5.1 SYNC MODE REGISTER

The Sync mode control register determines how the circuits within the chip are synchronized. Each circuit which requires synchronization can be configured to be synchronized to the sync input **(SI**), or to the terminal count of the sync counter (**TC**). The sync to each circuit can also be set to be always on or always off. Each circuit is given a two bit sync mode control which is defined as:

Table 4: Sync Modes

MODE	SYNC DESCRIPTION
0	“0” (never asserted)
1	SI
2	TC (or OS, if USE_ONESHOT is set)
3	“1” (always active)

NOTE: the internal syncs are active high. The **$\overline{\text{SI}}$** input has been inverted to be the active high sync **SI** .

ADDRESS 0: Sync Mode *, suggested default = 0x65*

BIT	TYPE	NAME	DESCRIPTION
0,1 (LSBs)	R/W	DEC_SYNC	Synchronizes the decimation control counter. The decimation counter controls the filter timing of each channel and the serial timing of the output signals.
2,3	R/W	COUNTER_SYNC	Synchronizes the sync counter. This counter is used to generate the periodic “TC” sync pulses. Mode 2 in Table 4 is always OS.
4,5	R/W	OUTPUT_SYNC	The selected sync is inverted and output on the **$\overline{\text{SO}}$** pin.
6	R/W	USE_ONESHOT	The terminal count mode in table 4 is replaced by the one shot pulse (OS) when this bit is set.
7	R/W	ONE_SHOT	The one shot sync pulse (OS) is generated when this bit is set. This bit must be cleared before another one shot pulse can be generated.

If the user wishes to allow the chip to free run, asynchronous to other chips, then the sync settings can be set to zero. If one wishes to synchronize several chips to a single sync source, then the sync mode selections should be set to one. The suggested default is to output the one-shot (USE_ONESHOT = 1, OUTPUT_SYNC=2) and set all other syncs to SI. The user should tie the **$\overline{\text{SO}}$** output pin of one GC4014 chip to the **$\overline{\text{SI}}$** input pin of all other GC4014 chips in a system in order to cleanly synchronize and initialize one or more GC4014 chips. If there is only a single GC4014 chip, then all sync mode selections can be set to “2” to receive the one-shot directly. A one-shot should be sent after initialization and each time the decimation ratio is changed.

5.2 DECIMATION MODE REGISTER

Registers 1 and 2 control the decimation modes for the chip. These settings are common to all channels

ADDRESS 1: Decimation Mode *, suggested default = 0x80, power up resets to 0.*

BIT	TYPE	NAME	DESCRIPTION
0 LSB	R/W	REAL	Enables the PFIR's real output mode. (See Section 3.3.6). The real outputs are formatted into complex pairs in the real mode. The I-output words contain are the even-time real outputs and the Q-outputs contain the odd-time real outputs.
1	R/W	FILTER_SELECT	The user downloaded filter coefficients are used instead of the built in filter coefficients for the second stage FIR filter when this bit is set.
2	R/W	RDY_POL	This control bit inverts the polarity of the **RDY** output. Normally **RDY** pulses high when a new sample is ready and an output sample period (OSP) is starting. **RDY** will pulse low when RDY_POL is high.
3	R/W	RDY_WIDTH	Normally the **RDY** pin will pulse active for four clock cycles. This control bit forces **RDY** to be active for 16 clocks.
4	R/W	LINK_MODE	Output the data in the nibble-serial link mode. The RDY/ACK pin becomes an input pin in this mode. NOTE: To use the link mode this bit must be set before the OUTPUT_ENABLE control bit in register 8 is set, otherwise the RDY/ACK pin will be driven as an output, possibly damaging the pin.
5	R/W	SO_INT_MODE	The $\overline{\textbf{SO}}$ output pin is used as an overflow interrupt pin when this bit is set. If an overflow due to gain settings occurs in any of the channels the $\overline{\textbf{SO}}$ pin will go low if this bit is set.
6	R/W	NO_SYMMETRY	The second stage decimate by two filter is normally a 63 tap symmetric filter. It becomes a 32 tap non-symmetric filter when this bit is set. ERRATA: The non-symmetry mode does not work properly for parts marked with mask code 55532B, parts with other mask codes work. Contact GRAYCHIP for details.
7 MSB	R/W	EN_DOUBLER	This bit must be set to enable the clock doubler circuit when the **CKMODE** pin is low. This bit is ignored when **CKMODE** pin is tied high.

5.3 CIC DECIMATION REGISTERS

Registers 2, and 3 contain the 14 bit CIC decimation ratio control.

ADDRESS 2: Decimation Byte 0 *, suggested default = 0x07*

BIT	TYPE	NAME	DESCRIPTION
0-7	R/W	**DEC**[0:7]	The LSBs of the decimation control

ADDRESS 3: Decimation Byte 1 *, suggested default = 0x00*

BIT	TYPE	NAME	DESCRIPTION
0-5	R/W	**DEC**[8:13]	The 6 MSBs of the decimation control
6,7	R	zero	These bits are read only zeros.

Where **DEC** is equal to **N**-1. The chip decimates the input data by a factor of 2N for real output data and 4N for complex output data (or 8N if DEC_BY_4 is set in Register 13), where **N** ranges from 8 to 16384. This provides an decimation range from 32 to 65,536 for complex output signals and 16 to 32,768 for real output signals.

5.4 SCALE CONTROL REGISTER

Register 4 controls the CIC filter gain for the chip. These settings are common to all channels

ADDRESS 4: CIC Scale *, suggested default = 0x71*

BIT	TYPE	NAME	DESCRIPTION
0-2	R/W	SCALE	SCALE ranges from 0 to 5.
3	R/W	unused	
4-6	R/W	BIG_SCALE	BIG_SCALE ranges from 0 to 7.
7	R/W	unused	

The CIC filter has a gain which is equal to N^4. To remove this gain the CIC inputs are pre-scaled down by (55-SCALE-6*BIG_SCALE) bits before filtering. The overall gain of each channel is equal to:

$$\text{GAIN} = \{N^4 2^{(\text{SCALE} + 6 \times \text{BIG\_SCALE} - 56)}\}[\ 2^{\text{COARSE}}]\{1.0 + \text{NARROW} \times 0.97\}\{\frac{\text{PFIR\_SUM}}{65536}\}[\frac{G}{32}]$$

where COARSE and G are unique for each channel (See Section 5.17). PFIR_SUM is the sum of the 63 PFIR coefficients if FILTER_SELECT in Register 1 is set, and NARROW is the CFIR narrow mode bit in Register 13. The values of SCALE and BIG_SCALE must be such that the term in braces is less than unity, i.e., $(\text{SCALE} + 6 \times \text{BIG\_SCALE}) \le (56 - 4\log_2 N)$. Overflows due to improper gain settings will go undetected if this relationship is violated. For example, this restriction means that BIG_SCALE and SCALE should be less than or equal to 7 and 2 respectively for N equal to 8. The BIG_SCALE and SCALE settings are common to all channels. See Section 7.9 for a discussion on how to optimally set the gain of the chip.

5.5 CHANNEL GAIN REGISTER

Register 5 contains the most significant 2 bits of each channel's gain setting G. The least significant bits are stored in each channel's control page (See Section 5.17).

ADDRESS 5: Channel Gain *, suggested default = 0x00*

BIT	TYPE	NAME	DESCRIPTION
0,1	R/W	GA[8:9]	2 MSBs of Channel A's gain.
3,4	R/W	GB[8:9]	2 MSBs of Channel B's gain.
4,5	R/W	GC[8:9]	2 MSBs of Channel C's gain.
6,7	R/W	GD[8:9]	2 MSBs of Channel D's gain.

Since the gain is G/32, and these bits are only used if G is greater than 256 (except for negative values), then setting this register to zero still allows the user to add up to 18 dB of gain by just using the 8 LSB's set in the channel control pages. If more than 18 dB is desired, then these control register bits can be used. See Section 7.9 for a discussion on how to optimally set the gain of the chip.

5.6 OUTPUT FORMAT REGISTER

This register controls the output bit serial format.

ADDRESS 6: Output Format Register *, suggested default = 0x01*

BIT	TYPE	NAME	DESCRIPTION
0-3 LSB	R/W	RATE[0:3]	The bit serial rate is set at $F_{CK}2^{-RATE}$, where RATE can range from 0 to 10. If RATE=0, the SCK pin will not toggle and the serial rate is equal to the clock rate.
4	R/W	PACKED	Puts the serial outputs into 32 bit transfer mode where each complex pair is packed into 32 bit words. The complex pair is formatted as I word in the upper byte and the Q word in the lower byte. Each word is formatted as MSB first.
5	R/W	CONTINUOUS	The serial clock normally stops when the last bit of each transmission is complete and stays low until the next OSP.
6	R/W	SCK_POL	This bit inverts the polarity of the serial clock. Normally SOUT and SFS change on the rising edge of **SCK**. They change on the trailing edge when this bit is set.
7 MSB	R/W	SFS_POL	The **SFS** signal is treated as active low when this bit is set. Otherwise the signal is treated as active high.

5.7 OUTPUT MODE REGISTER

This register controls the output summation, multiplexing and rounding.

ADDRESS 7: Output Mode Register *, suggested default = 0x00*

BIT	TYPE	NAME	DESCRIPTION
0,1 LSB	R/W	SUM_MODE	The channel outputs are replaced by the sum of outputs as shown in Table 1 of Section 3.5
2,3	R/W	MUX_MODE	The outputs are multiplexed as described in Section 3.6.2.
4	R/W	RND8	Round into the 8 MSBs of the 16 bit output words.
5	R/W	RND10	Round into the 10 MSBs of the 16 bit output words.
6	R/W	RND12	Round into the 12 MSBs of the 16 bit output words.
7 MSB	R/W	RND14	Round into the 14 MSBs of the 16 bit output words.

Only one round control bit can be set. If none are set the output is 16 bits. Bits below the rounding point are set to zero.

5.8 BLANKING CONTROL REGISTER

This register controls the blanking mode.

ADDRESS 8: Blank Control Register *, suggested default = 0x50, power up reset to 0.*

BIT	TYPE	NAME	DESCRIPTION
0-3 LSB	R/W	BLANK_RATE	The number of zeroes to insert between each sample in the blanking mode. Ranges from 0 to 15.
4,5	R/W	BLANK_SYNC	The sync selection from Table 4 for synchronizing the zero stuffing.
6	R/W	OUTPUT_ENABLE	Turns on the serial output pins including **SFS** and **SCK**. **RDY** is also turned on if LINK_MODE is off (See Section 5.2).
7 MSB	R/W	RAM_TEST	Used for factory tests. Should be kept low.

Blanking is turned on for each channel using the channel mode register in each channel's control page.

5.9 CHANNEL FLUSH CONTROL REGISTER

This register controls flushing the four channels. Each channel is flushed when the selected sync occurs The sync is selected according to Table 4 in Section 5.1.

ADDRESS 9: Channel Flush Register *, suggested default = 0x55*

BIT	TYPE	NAME	DESCRIPTION
0,1 LSB	R/W	FLUSH_A[0:1]	The flush sync for channel A.
2,3	R/W	FLUSH_B[0:1]	The flush sync for channel B.
4,5	R/W	FLUSH_C[0:1]	The flush sync for channel C.
6,7 MSB	R/W	FLUSH_D[0:1]	The flush sync for channel D.

Each channel should be flushed when the chip is being initialized or when the decimation control is changed. The flush lasts for 8N clocks after the sync occurs. The channel flush syncs will normally be left in a “never” mode. If a channel is unused, then the user should leave the channel in the “always” flush mode which will clear the datapath, clear the channel’s output, and lower its power consumption. During diagnostics the channels will need to be flushed at the beginning of each sync cycle.

The user may wish to flush a channel when a new frequency is selected in order to purge the datapath of the last signal.

5.10 COUNTER MODE REGISTERS

Registers 10, and 11 set the counter’s cycle period.

ADDRESS 10: Counter Byte 0 *, suggested default = 0xff*

BIT	TYPE	NAME	DESCRIPTION
0-7	R/W	**CNT**[0:7]	The LSBs of the counter cycle period

ADDRESS 11: Counter Byte 1 *, suggested default = 0xff*

BIT	TYPE	NAME	DESCRIPTION
0-7	R/W	**CNT**[8:15]	The 8 MSBs of the counter cycle period

The chip’s internal sync counter counts in cycles of 128(**CNT** +1) clocks. A terminal count signal (TC) is output at the end of each cycle. The counter can be synchronized to an external sync as specified in the Sync mode Register (See Section 5.1). If **CNT** is set so that 128(**CNT** +1) is a multiple of twice the decimation ratio (i.e., a multiple of 16N), then the terminal count of this counter can be output on the $\overline{\textbf{SO}}$ pin and used to periodically synchronize multiple GC4014 chips.

5.11 TEST MODE REGISTER

Register 12 controls the test and diagnostic features of the chip.

ADDRESS 12: Test Mode Register *, suggested default = 0x08, power up reset to 0.*

BIT	TYPE	NAME	DESCRIPTION
0,1 LSB	R/W	DIAG_SOURCE	This two bit field selects the diagnostic input source used when the DIAG bit in each channel's control register is set (See Section 5.17). DIAG_SOURCE = 0 selects the 14 LSB's of the counter (see Section 5.10) as a diagnostic ramp. DIAG_SOURCE=1 is a zero input, DIAG_SOURCE=2 is unused, DIAG_SOURCE= 3 gives a 0x4000 constant input.
2,3	R/W	DIAG_SYNC	The Checksum generator is strobed by this sync. See Table 4 for the possible sync selections.
4	R/W	COUNT_TEST	Used during factory tests. Should be cleared during normal operation.
5	R/W	PD_CLOCK_OFF	Turns off the clock in the power down mode.
6,7	R/W	POWER_DOWN	This two bit field controls the power down and keep alive circuit as follows: POWER_DOWN — MODE 0 — Clock loss detect mode 1 — Power down mode 2 — Disabled 3 — Test The power_down mode defaults to 0 (clock loss detect mode) upon power up.

5.12 PAGE MAP REGISTER

This register selects which page is been accessed by addresses 16 through 31. This register also contains several miscellaneous control bits.

ADDRESS 13: Page Map Register

BIT	TYPE	NAME	DESCRIPTION
0-3 LSBs	R/W	PAGE	The page selection. PAGE=0 selects the channel output data page, PAGE=1 selects the status page, page=4,5,6 and 7 select the channel control pages, pages 8,9,10 and 11 select the PFIR coefficient pages.
4	R/W	NARROW	Selects the narrow mode CFIR coefficients.
5	R/W	DEC_BY_4	Enables the decimate by 4 mode of the PFIR.
6	R/W	16_BIT_INPUT	Enables the 16 bit input mode. DIN[6-13] are unused. DIN[0,1], DIN[2,3], and DIN[4,5] become two additional LSB's for AIN, BIN, and CIN respectively.
7 MSB	R/W	MSB_POL	Invert the input MSB polarity. This will convert an offset binary formatted input to 2's complement format.

5.13 STATUS CONTROL REGISTER

This register contains miscellaneous control and status information.

ADDRESS 14: Status Control Register *, suggested default = 0x00*

BIT	TYPE	NAME	DESCRIPTION
0 LSB	R/W	READY	The user sets this bit after reading the output registers. The chip clears this bit when new values have been loaded and it is time to read them.
1	R/W	MISSED	The chip sets this bit If the user has not set the READY bit before the chip loads the output registers. This bit high indicates that an error has occurred.
2	R/W	unused	
3	R/W	GAIN_OVERFLOW	Indicates an overflow in the final gain circuit.
4	R/W	OVERFLOW_A	Indicates overflow in channel A's coarse gain.
5	R/W	OVERFLOW_B	Indicates overflow in channel B's coarse gain.
6	R/W	OVERFLOW_C	Indicates overflow in channel C's coarse gain.
7 MSB	R/W	OVERFLOW_D	Indicates overflow in channel D's coarse gain.

The READY bit is used to tell an external processor when new output samples are ready to be read. If desired, the **RDY** pin can be used as an interrupt to the external processor (See Section 5.2) to tell the processor when to read new samples. The user does not need to set the READY bit if **RDY** is used. If READY is not set, however, the MISSED flag will not be valid.

The overflow bits are set when an overflow occurs and stays set until the user clears them. If the SO_INT_MODE bit in control register 1 is set, then the $\overline{\textbf{SO}}$ pin will go low if OVERFLOW_A, OVERFLOW_B, OVERFLOW_C, or OVERFLOW_D go active. GAIN_OVERFLOW will not cause $\overline{\textbf{SO}}$ to go low. GAIN_OVERFLOW is set if the final gain circuit detects an overflow in any channel (or sum of channels if SUM_MODE is active).

5.14 CHECKSUM REGISTER

The checksum register is a read only register which contains the checksum of the output data. The checksum is stored in the checksum register and then starts over again each time the DIAG_SYNC (See Section 5.11) occurs. This is a read only register.

ADDRESS 15: Checksum Register

BIT	TYPE	NAME	DESCRIPTION
0-7	R	**CHECKSUM**[0:7]	The checksum.

5.15 CHANNEL OUTPUT PAGE (PAGE = 0)

Addresses 16 through 31 are used to read output values. The outputs are 16 bit two's complement numbers which are read as two 8 bit bytes, the lower address contains the lower byte. See Table 1 for the output value definitions when SUM_MODE is used. The address assignments are:

ADDRESSES	NAME	ADDRESSES	NAME
16,17	A_{OUT}, I-half	24,25	C_{OUT}, I-half
18,19	A_{OUT}, Q-half	26,27	C_{OUT},Q-half
20,21	B_{OUT}, I-half	28,29	D_{OUT}, I-half
22,23	B_{OUT},Q-half	30,31	D_{OUT},Q-half

These are all read only registers.

5.16 KEEPALIVE STATUS PAGE (PAGE = 1)

ADDRESS 16: Clock Status

BIT	TYPE	NAME	DESCRIPTION
0	R	KACK	This bit monitors the keepalive clock.
1	R	KA_MODE	This bit monitors the keepalive mode.
2-7	-	unused	

These bits are used for factory test purposes only.

ADDRESS 17: Mask Revision

BIT	TYPE	NAME	DESCRIPTION
0-7	R	REVISION	Mask revision number.

This address can be used to determine the mask revision number for the GC4014. The mask revision numbers are shown in Table 5 below (the mask codes are printed on the GC4014 package).

Table 5: Mask Revisions

Mask Revision Number (Address 17)	Release Date	Mask Code on Package	Description
0	October 1997	55532B	Original
0	January 1998	55532C	Corrected problems with non-symmetry mode, did not change mask revision number in address 17.

5.17 CHANNEL CONTROL PAGES (PAGES 4, 5, 6, 7)

Pages 4, 5, 6 and 7 contain the frequency, phase, gain and control settings for the four channels. To configure channels A, B, C and D use pages 4, 5, 6 and 7 respectfully. All registers are read/write.

ADDRESSES 16, 17, 18, 19: Frequency

ADDRESS	NAME	DESCRIPTION
16	FREQ[0:7]	Byte 0 (LSBs) of FREQ
17	FREQ[8:15]	Byte 1 of FREQ
18	FREQ[16:23]	Byte 2 of FREQ
19	FREQ[24:31]	Byte 3 (MSBs) of FREQ

The 32 bit frequency control word is defined as:

$$FREQ = 2^{32}F/F_{CK}$$

where F is the desired tuning frequency and F_{CK} is the chip's clock rate (not the CK2X rate). Use positive frequency values to downconvert signals. Use negative frequency values to invert the signal's spectrum. The 32 bit 2's complement frequency words are entered as four bytes, the least significant byte in the lowest address, the most significant in the highest address.

ADDRESSES 20, 21: Phase

ADDRESS	NAME	DESCRIPTION
20	PHASE[0:7]	Byte 0 (LSBs) of PHASE
21	PHASE[8:15]	Byte 1 (MSBs) of PHASE

The 16 bit phase offset is defined as:

$$PHASE = 2^{16}P/2\pi$$

where P is the desired phase in radian from 0 to 2π.

ADDRESS 22: Gain *, suggested default = 0x80*

ADDRESS	NAME	DESCRIPTION
22	G[0:7]	Byte 0 (LSBs) of G

The upper two bits of G are stored in control register 5. Note that G is only part of the chip's gain and should be used in conjunction with SCALE, BIG_SCALE and COARSE. See Sections 5.4 and 5.5 for details. See Section 7.9 for a discussion on how to optimally set the gain of the chip.

ADDRESS 23: Channel Control *, suggested default = 0x0, power up reset to 0.*

BIT	TYPE	NAME	DESCRIPTION
0,1 LSB	R/W	INPUT	This two bit field selects which chip input should be used by the channel. INPUT = 0,1,2 and 3 correspond to inputs A, B, C and D.
2	R/W	DIAG	Use the diagnostic source as the channel input. See Address 12 (Section 5.11).
3	R/W	BLANK	Turn on blanking for this channel. See address 8, Section 5.8
4-6	R/W	COARSE	The COARSE gain setting used for this channel. See Sections 5.4 and 5.5 for details.
7 MSB	R/W	POWER_DOWN	Used to force the channel into the power down state.

ADDRESS 24: Channel Sync Modes *, suggested default = 0x5f*

BIT	TYPE	NAME	DESCRIPTION
0,1 LSB	R/W	FREQ_SYNC	The new frequency setting takes affect on this sync.
2,3	R/W	PHASE_SYNC	The new phase offset takes affect on this sync
4,5	R/W	NCO_SYNC	The NCO is initialized to the phase setting by this sync
6,7 MSB	R/W	DITHER_SYNC	The dither circuit is initialized by this sync to zero.

These syncs use the selections shown in Table 4.

The NCO_SYNC is usually set to be always off, unless the user wants to coherently control the phases of multiple channels.

The FREQ_SYNC and PHASE_SYNC are typically set to be always on so that frequency and phase settings will take effect immediately as they are written into their control registers.

The DITHER_SYNC is used to turn on or off the dithering of the NCO phase. To turn off dithering set the DITHER_SYNC to be always on so that it remains initialized to zero. To turn dithering on set the sync to be always off.

During diagnostics the NCO_SYNC and DITHER_SYNC should be set to “TC”.

5.18 COEFFICIENT STORAGE PAGES (PAGES 8, 9, 10 and 11)

Addresses 16 to 31 are used to download the 32 user programmable filter coefficients when PAGE is set to 8, 9, 10 and 11. Page 8 is for coefficients 0 through 7, page 9 is for coefficients 8 through 15, page 10 is for coefficients 16 through 23, and page 11 is for coefficients 24 through 31, where coefficient 0 is the first coefficient and coefficient 31 is the middle coefficient of the filter's impulse response. The 16 bit 2's complement coefficients are stored in two bytes, least significant byte first, for example, the LSBs of coefficient 0 are stored in address 16 and the MSBs in address 17.

TO LOAD A COEFFICIENT THE USER MUST WRITE THE LSBYTE FIRST FOLLOWED BY THE MSBYTE. Unknown values will be written into the LSBs if the MSB is written first. The coefficient registers are write only.

6.0 SPECIFICATIONS

6.1 ABSOLUTE MAXIMUM RATINGS

Table 6: Absolute Maximum Ratings

PARAMETER	SYMBOL	MIN	MAX	UNITS	NOTES
DC Supply Voltage	V_{CC}	-0.3	4.1	V	
Control I/O, CKMODE and RDY/ACK Supply Voltage	V_{UP}	-0.3	6.0	V	
Input voltage (undershoot and overshoot)	V_{IN}	-0.5	V_{CC}+0.5	V	1
Storage Temperature	T_{STG}	-65	150	^{O}C	
Lead Soldering Temperature (10 seconds)			300	^{O}C	

Notes:
1. MAX is V_{UP}+0.5 for the Control I/O and ACK pins.

6.2 RECOMMENDED OPERATING CONDITIONS

Table 7: Recommended Operating Conditions

PARAMETER	SYMBOL	MIN	MAX	UNITS	NOTES
DC Supply Voltage	V_{CC}	3.1	3.5	V	1
Control I/O, CKMODE and RDY/ACK Supply Voltage	V_{UP}	V_{CC}	5.5	V	
Temperature Ambient, no air flow	T_A	-40	+85	^{O}C	2
Junction Temperature	T_J		125	^{O}C	2

1. DC and AC specifications are tested for this range. The GC4014 will operate at derated specifications for lower supply voltages.
2. Thermal management may be required for full rate operation, See Table 8 below

6.3 THERMAL CHARACTERISTICS

Table 8: Thermal Data

THERMAL CONDUCTIVITY	SYMBOL	GC4014-CQ		GC4014-PQ		UNITS
		0.5 Watt	1 Watt	0.5 Watt	1 Watt	
Theta Junction to Ambient	θja	54	37	40	32	^{O}C/W
Theta Junction to Case	θjc	17	15	13	11	^{O}C/W

Note: Air flow will reduce θ**ja** and is highly recommended.

6.4 DC CHARACTERISTICS

All parameters are industrial temperature range of -40 to 85 ^{O}C ambient unless noted.:

Table 9: DC Operating Conditions

PARAMETER	SYMBOL	Vcc = 3.1 to 3.5V		UNITS	NOTES
		MIN	MAX		
Voltage input low	V_{IL}		0.8	V	2
Voltage input high, AIN,BIN,CIN,DIN,$\overline{SI}$,C	V_{IH}	2.1		V	2,3
Voltage input high CK, CK2X	V_{IH}	2.4		V	2
Input current (V_{IN} = 0V)	I_{IN}	Typical +/- 20		uA	2
Voltage output low (I_{OL} = 4mA)	V_{OL}		0.5	V	2
Voltage output high (I_{OH} = -4mA)	V_{OH}	2.4	3.3	V	2,4
Data input capacitance (All inputs except **CK**)	C_{IN}	Typical 4		pF	1
Clock input capacitance (**CK** input)	C_{CK}	Typical 10		pF	1

Notes:

1. Controlled by design and process and not directly tested. Verified on initial parts evaluation.
2. Each part is tested at 85^{o}C for the given specification.
3. For V_{UP}=5V, V_{IH}=2.5V for the Control I/O, CKMODE and ACK pins.
4. For V_{UP}=5V, V_{OH}=2.8V (MIN) and V_{OH}=5V (MAX) for the Control I/O and RDY pins.

6.5 AC CHARACTERISTICS

Table 10: AC Characteristics (-40 TO +85 ^{O}C Ambient, unless noted)

PARAMETER	SYMBOL	3.1V to 3.5V		UNITS	NOTES
		MIN	MAX		
Clock Frequency	F_{CK}	Note 5	64	MHz	2, 3
Clock low period (Below V_{IL})	t_{CKL}	6		ns	2
Clock high period (Above V_{IH})	t_{CKH}	6		ns	2
Clock rise and fall times (V_{IL} to V_{IH})	t_{RF}		2	ns	1
Input setup before **CK** goes high (**AIN** , **BIN, CIN, DIN** , or $\overline{\textbf{SI}}$)	t_{SU}	2		ns	2
Input hold time after **CK** goes high	t_{HD}	2		ns	2
Data output delay from rising edge of **CK** . (**AOUT** , **BOUT, COUT, DOUT, SFS, SCK, RDY** , or $\overline{\textbf{SO}}$)	t_{DLY}	2 Note 1	15 Note 2	ns	4
Control Setup before both $\overline{\textbf{CE}}$, $\overline{\textbf{WR}}$ or $\overline{\textbf{RD}}$ go low (See Figure 2.0)	t_{CSU}	3		ns	2, 8
Control hold after $\overline{\textbf{CE}}$, $\overline{\textbf{WR}}$ or $\overline{\textbf{RD}}$ go high (See Figure 2.0)	t_{CHD}	3		ns	2, 8
Control strobe ($\overline{\textbf{CE}}$ or $\overline{\textbf{WR}}$) pulse width (Write operation, See Figure 2.0)	t_{CSPW}	20		ns	2, 8
Control output delay $\overline{\textbf{CE}}$ and $\overline{\textbf{RD}}$ low to **C** (Read Operation, See Figure 2.0	t_{CDLY}		30	ns	2, 6, 8
Control tristate delay after $\overline{\textbf{CE}}$ and $\overline{\textbf{RD}}$ go high (See Figure 2.0)	t_{CZ}		10	ns	1
Quiescent supply current (V_{IN}=0 or V_{CC}, F_{CK} = 1KHz or POWER_DOWN=1)	I_{CCQ}		7	mA	1
Supply current (F_{CK} =64MHz, N=8)	I_{CC}		400	mA	2, 7

Notes:

1. Typical and not directly tested. Verified on initial part evaluation.
2. Each part is tested at 85 deg C for the given specification.
3. The chip may not operate properly at clock frequencies below MIN and above MAX.
4. Capacitive output load is 20pf. Delays are measured from the rising edge of the clock to the output level rising above V_{IH} or Falling below V_{IL}.
5. The minimum clock rate must satisfy $F_{CK}/(4N) > 1KHz$.
6. Capacitive output load is 80pf.
7. Current changes linearly with voltage and clock speed: Icc (MAX) $= \left(\frac{VCC}{3.3}\right)\left(\frac{F_{CK}}{50M}\right)\left[31 + A\left(29 + \frac{320}{N}\right)\right]mA$

   where A is the number of active channels (0 to 4) and N is the CIC decimation ratio.
8. See timing diagram in Figure 2 and description in Section 3.1.

7.0 APPLICATION NOTES

7.1 POWER AND GROUND CONNECTIONS

The GC4014 chip is a very high performance chip which requires solid power and ground connections to avoid noise on the V_{CC} and GND pins. If possible the GC4014 chip should be mounted on a circuit board with dedicated power and ground planes and with at least two decoupling capacitors (0.01 and 0.1 μf) adjacent to each GC4014 chip. If dedicated power and ground planes are not possible, then the user should place decoupling capacitors adjacent to each V_{CC} and GND pair.

IMPORTANT

The GC4014 chip may not operate properly if these power and ground guidelines are violated.

7.2 STATIC SENSITIVE DEVICE

The GC4014 chip is fabricated in a high performance CMOS process which is sensitive to the high voltage transients caused by static electricity. These parts can be permanently damaged by static electricity and should only be handled in static free environments.

7.3 SYNCHRONIZING MULTIPLE GC4014 CHIPS

A system containing two or more GC4014 chips will need to be synchronized if coherent operation is desired. To synchronize multiple GC4014 chips connect all of the sync input pins together so they can be driven by a common sync strobe. The common sync strobe can be from an external source, or can be the sync output from one of the chips. If the sync output from one of the chips is used, then the user can choose to output a one shot sync pulse from that chip, or the terminal count from the chip's sync counter. If the terminal count is used, then the sync cycle must be a multiple of 8N and the FLUSH (Address 9), NCO_SYNC and DITHER_SYNC (Address 24 of each channel page) syncs must be set to "never" (see Table 4) after initial synchronization.

7.4 THERMAL MANAGEMENT

The junction temperature must be kept below 125 $^{\circ}$C for reliable operation. The chip's power dissipation should be calculated using the equation for supply current in Section 6.5 and then the chip's junction temperature can be calculated using the package's thermal conductivity shown in Section 6.3. At full rate operation (F_{CK}=62.5) the power is 1.3 Watts and the junction to ambient rise is 32 degrees per Watt for the plastic package. This represents a rise of 42 degrees over ambient. This means that under these conditions the ambient temperature has to be less than 83 $^{\circ}$C. Air flow will decrease the thermal resistance by 10% to 40%, allowing ambient temperatures between 87°C and 100 $^{\circ}$C. Increasing the decimation ratio (N) or decreasing the number of active channels (A) will also allow a higher ambient temperature.

7.5 GSM APPLICATION

The chip meets the stringent GSM out of band rejection requirements by using the narrow CFIR mode. The overall response using the narrow CFIR mode and a programmable PFIR filter coefficient set targeted towards GSM is shown in Figure 11, along with GSM's out of band rejection mask. In band ripple is 0.2 dB (peak to peak).

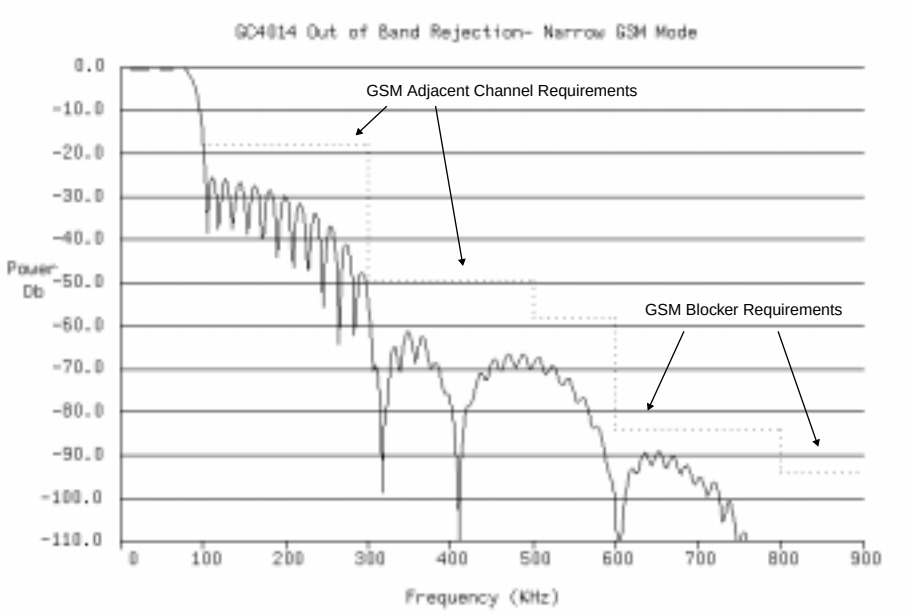


Figure 11. Overall Frequency Response for GSM

This response assumes the input sample rate is equal to 8*N*B, where N is the decimation in the CIC filter (See Section 5.3) and B is the GSM bit rate (270.833 KHz). The data is output as two complex samples per bit (541.667 KHz) when the PFIR is in the decimate by 2 mode (DEC_BY_4 = 0 in register 13), or is output as one sample per bit (270.833 KHz) when the PFIR is in the decimate by 4 mode (DEC_BY_4=1).

The programmable PFIR coefficients for the GSM mode are:

```
1007, -1853, 79, 1807, 1633, 423, 265, 175, -527, -1331, -1454, -1087,
-721, -149, 1008, 1985, 2164, 2005, 1483, 61, -1756, -3134, -3953,
-4016, -2714, 134, 4003, 8361, 12934, 17009, 19565, 20353
```

The suggested GSM control register settings for the chip with an input sample rate of 54.166 MHz (N=25) is shown in Table 12 (other input rates can be used up to 62.5 MHz, 54.166 is used as an example):

Table 11: GSM Configuration

Control Registers		Channel pages 4,5,6,7		Coefficient Pages			
Address	Data	Address	Data	Page 8	Page 9	Page 10	Page 11
00 (HEX)	65 (HEX)[1]	10 (HEX)	FREQ[0:7]	EF	F1	74	66
01	80->82[2]	11	FREQ[8:15]	03	FD	08	F5
02	19	12	FREQ[16:23]	C3	CD	D5	86
03	00	13	FREQ[24:31]	F8	FA	07	00
04	60	14	00 (HEX)	4F	52	CB	A3
05	00	15	00	00	FA	05	0F
06	11	16	2E	0F	C1	3D	A9
07	00	17	00	07	FB	00	20
08	50	18	5F	61	2F	24	86
09	55	19		06	FD	F9	32
0A	FF	1A		A7	6B	C2	71
0B	FF	1B		01	FF	F3	42
0C	08	1C		09	F0	8F	6D
0D	1P[3]	1D		01	03	F0	4C
0E	00	1E		AF	C1	50	81
0F	read only	1F		00	07	F0	4F

1. Initialize to 65 while configuring the chip(s), then set to E5, then back to 65 to fire off the one-shot sync.This assumes that $\overline{SO}$ is tied to $\overline{SI}$
2. Initialize to 80, then set to 82 after external coefficients are loaded.
3. “P” is the page number. The upper nibble should stay at “1¨.

The variables SCALE, BIG_SCALE (address 4) and G (address 16_{HEX} of pages 4,5,6,and 7) are set as follows. The values of SCALE and BIG_SCALE must be set to satisfy: $(\text{SCALE} + 6 \times \text{BIG\_SCALE}) \le (56 - 4\log_2 N)$. N is 25, so ($4\log_2 N$) is 37.14. This means $(\text{SCALE} + 6 \times \text{BIG\_SCALE}) \le 37$, which is satisfied by setting SCALE=1 and BIG_SCALE=6. SCALE, however, needs to be decreased to 0 to prevent overflow in the CFIR, which has a gain of 1.97. The overall gain is set using “G” according to:

$$\text{GAIN} = \{25^4 2^{(36-56)}\}[\,2^{\text{COARSE}}]\{1.97\}\{\tfrac{\text{PFIR\_SUM}}{65536}\}\left[\tfrac{G}{32}\right] = 0.0438G$$

Where COARSE is 0 and PFIR_SUM is 125151. The optimal setting of GAIN is 2.0 in order to compensate for the loss in the tuning process (See Section 7.9). A value of G=46 ($2E_{HEX}$) will give a gain of 2.015.

The output serial format is set in address 6 to have a bit rate of one half the chip’s clock rate and to be used in the packed mode. The user will need to configure the output format as is necessary for the application.

7.6 DAMPS APPLICATION

The chip meets the DAMPS out of band rejection requirements by using the narrow CFIR mode. The overall response using the narrow CFIR mode and a programmable PFIR filter coefficient set targeted towards DAMPS is shown in Figure 12, along with the out of band rejection mask for DAMPS. The in-band response matches the desired root-raised-cosine receive filter shape to within 0.1 dB.

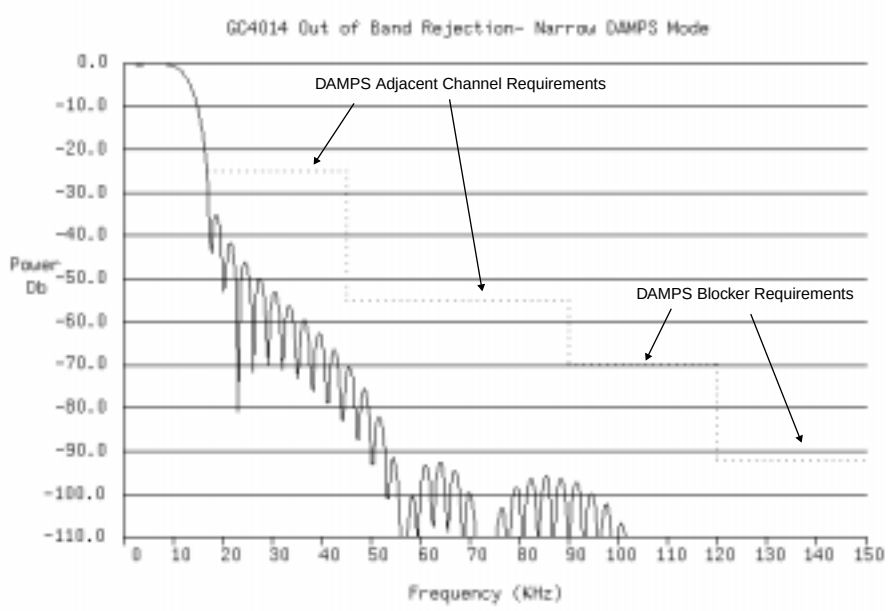


Figure 12. Overall Frequency Response for DAMPS

This response assumes the input sample rate is equal to 16*N*B, where N is the decimation in the CIC filter (See Section 5.3) and B is the DAMPS symbol (baud) rate (24.3 KHz). The data is output as four complex samples per symbol (97.2 KHz) when the PFIR is in the decimate by 2 mode (DEC_BY_4 = 0 in register 13), or is output as two samples per symbol (24.3 KHz) when the PFIR is in the decimate by 4 mode (DEC_BY_4=1).

The programmable PFIR coefficients for the DAMPS mode are:

```
264, 398, 413, 286, 34, -286, -584, -761, -729, -442, 85, 766, 1451,
1954, 2086, 1708, 768, -660, -2371, -4041, -5269, -5640, -4801, -2532,
1194, 6185, 12037, 18172, 23923, 28618, 31691, 32759
```

The suggested DAMPS control register settings for the chip with an input sample rate of 49.7664 MHz (N=128) is shown in the following table (other input rates can be used up to 62.5 MHz, 49.7664 is used as an example):

Table 12: DAMPS Configuration

Control Registers		Channel pages 4,5,6,7		Coefficient Pages			
Address	Data	Address	Data	Page 8	Page 9	Page 10	Page 11
00 (HEX)	65 (HEX)[1]	10 (HEX)	FREQ[0:7]	08	27	00	AA
01	80->82[2]	11	FREQ[8:15]	01	FD	03	04
02	7F	12	FREQ[16:23]	8E	46	6C	29
03	00	13	FREQ[24:31]	01	FE	FD	18
04	43	14	00 (HEX)	9D	55	BD	05
05	00	15	00	01	00	F6	2F
06	11	16	12	1E	FE	37	FC
07	00	17	00	01	02	F0	46
08	50	18	5F	22	AB	6B	73
09	55	19		00	05	EB	5D
0A	FF	1A		E2	A2	F8	CA
0B	FF	1B		FE	07	E9	6F
0C	08	1C		B8	26	3F	CB
0D	1P[3]	1D		FD	08	ED	7B
0E	00	1E		07	AC	1C	F7
0F	read only	1F		FD	06	F6	7F

1. Initialize to 65 while configuring the chip(s), then set to E5, then back to 65 to fire off the one-shot sync.This assumes that $\overline{SO}$ is tied to $\overline{SI}$
2. Initialize to 80, then set to 82 after external coefficients are loaded.
3. “P” is the page number. The upper nibble should stay at “1¨.

The variables SCALE, BIG_SCALE (address 4) and G (address 16_{HEX} of pages 4,5,6,and 7) are set as follows. The values of SCALE and BIG_SCALE must be set to satisfy: $(SCALE + 6 \times BIG\_SCALE) \le (56 - 4\log_2 N)$. N is 128, so ($4\log_2 N$) is 28. This means $(SCALE + 6 \times BIG\_SCALE) \le 28$, which is satisfied by setting SCALE=4 and BIG_SCALE=4. SCALE, however, needs to be decreased to 3 to prevent overflow in the CFIR, which has a gain of 1.97. The overall gain is set using “G” according to:

$$GAIN = \{128^4 2^{(27-56)}\}[\,2^{COARSE}](1.97)(\frac{PFIR\_SUM}{65536})\left(\frac{G}{32}\right) = 0.113G$$

Where COARSE is 0 (see Section 7.9) and PFIR_SUM is 240593. The optimal setting of GAIN is 2.0 in order to compensate for the loss in the tuning process (See Section 7.9). A value of G=18 (12_{HEX}) will give a gain of 2.034.

The output serial format is set in address 6 to have a bit rate of one half the chip’s clock rate and to be used in the packed mode. The user will need to configure the output format as is necessary for the application.

7.7 DIAGNOSTICS

Four diagnostic tests are described here. These tests use the diagnostic ramp as the input data source and the counter for synchronization. The tests are run by loading the configurations, waiting for the checksum to stabilize, and then reading the checksum from address 15 and comparing it to the expected checksum shown in each configuration table for address 15.

Table 13: Diagnostic Test 1 Configuration

Control Registers		Channel pages 4,5,6,7					Coefficient Pages			
Address	Data	Address	Page 4	Page 5	Page 6	Page 7	Page 8	Page 9	Page 10	Page 11
00 (HEX)	2A (HEX)	10 (HEX)	12 (HEX)	45 (HEX)	AA (HEX)	55 (HEX)	Unused			
01	80	11	23	23	55	AA				
02	07	12	34	34	AA	55				
03	00	13	01	02	00	03				
04	72	14	55	55	55	55				
05	00	15	55	55	55	55				
06	00	16	FF	AA	55	80				
07	00	17	04	04	04	04				
08	51	18	AA	AA	AA	AA				
09	AA	19								
0A	FF	1A								
0B	0F	1B								
0C	88	1C								
0D	0P[2]	1D								
0E	00	1E								
0F	**85**[1]	1F								

1. This is the read-only expected checksum
2. P = desired page number

Table 14: Diagnostic Test 2 Configuration

Control Registers		Channel pages 4,5,6,7					Coefficient Pages			
Address	Data	Address	Page 4	Page 5	Page 6	Page 7	Page 8	Page 9	Page 10	Page 11
00 (HEX)	2A (HEX)	10 (HEX)	55 (HEX)	12 (HEX)	45 (HEX)	AA (HEX)	Unused			
01	80	11	AA	23	23	55				
02	0F	12	55	34	34	AA				
03	00	13	03	00	01	02				
04	64	14	AA	AA	AA	AA				
05	00	15	AA	AA	AA	AA				
06	01	16	AA	55	7F	FF				
07	00	17	04	04	04	04				
08	51	18	AA	AA	AA	AA				
09	AA	19								
0A	FF	1A								
0B	0F	1B								
0C	88	1C								
0D	0P	1D								
0E	00	1E								
0F	**80**[1]	1F								

1. This is the read-only expected checksum

Table 15: Diagnostic Test 3 Configuration

Control Registers		Channel pages 4,5,6,7					Coefficient Pages			
Address	Data	Address	Page 4	Page 5	Page 6	Page 7	Page 8	Page 9	Page 10	Page 11
00 (HEX)	2A (HEX)	10 (HEX)	12 (HEX)	45 (HEX)	AA (HEX)	55 (HEX)	AA	AA	AA	AA
01	80->82[2]	11	23	23	55	AA	55	55	55	55
02	07	12	34	34	AA	55	AA	AA	AA	AA
03	00	13	00	01	02	03	55	55	55	55
04	72	14	55	55	55	55	AA	AA	AA	AA
05	00	15	55	55	55	55	55	55	55	55
06	00	16	FF	AA	55	80	AA	AA	AA	AA
07	00	17	04	04	04	04	55	55	55	55
08	51	18	AA	AA	AA	AA	AA	AA	AA	AA
09	AA	19					55	55	55	55
0A	FF	1A					AA	AA	AA	AA
0B	0F	1B					55	55	55	55
0C	88	1C					AA	AA	AA	AA
0D	0P	1D					55	55	55	55
0E	00	1E					AA	AA	AA	AA
0F	**BA**[1]	1F					55	55	55	55

1. This is the read-only expected checksum
2. Initialize to 80, then set to 82 after coefficients have been loaded

Table 16: Diagnostic Test 4 Configuration

Control Registers		Channel pages 4,5,6,7					Coefficient Pages			
Address	Data	Address	Page 4	Page 5	Page 6	Page 7	Page 8	Page 9	Page 10	Page 11
00 (HEX)	2A (HEX)	10 (HEX)	55 (HEX)	12 (HEX)	45 (HEX)	AA (HEX)	55	55	55	55
01	80->82[2]	11	AA	23	23	55	AA	AA	AA	AA
02	FF	12	55	34	34	AA	55	55	55	55
03	00	13	03	00	01	02	AA	AA	AA	AA
04	40	14	AA	AA	AA	AA	55	55	55	55
05	00	15	AA	AA	AA	AA	AA	AA	AA	AA
06	00	16	AA	55	7F	FF	55	55	55	55
07	00	17	04	04	04	04	AA	AA	AA	AA
08	51	18	AA	AA	AA	AA	55	55	55	55
09	AA	19					AA	AA	AA	AA
0A	FF	1A					55	55	55	55
0B	0F	1B					AA	AA	AA	AA
0C	88	1C					55	55	55	55
0D	0P	1D					AA	AA	AA	AA
0E	00	1E					55	55	55	55
0F	**D2**[1]	1F					AA	AA	AA	AA

1. This is the read-only expected checksum
2. Initialize to 80, then set to 82 after coefficients have been loaded

7.8 OUTPUT TEST CONFIGURATION

The following configuration allows the user to debug the output interface to insure that the GC4014 data is being received properly by the following circuitry. The configuration in the following table will generate a fixed output sequence of four values (two complex pairs) which will repeat indefinitely:

Table 17: Output Test Configuration

Control Registers		Channel pages 4,5,6,7	
Address	Data	Address	Data
00 (HEX)	6A (HEX)[1]	10 (HEX)	00 (HEX)
01	81	11	00
02	0F[2]	12	00
03	00[2]	13	00
04	63[2]	14	00
05	00	15	00
06	11[3]	16	80
07	10	17	74
08	50	18	FF
09	AA	19	
0A	FF	1A	
0B	FF	1B	
0C	0F	1C	
0D	0P[4]	1D	
0E	00	1E	
0F	read only	1F	

1. Initialize to 6A while configuring the chip(s), then set to EA, then back to 6A to fire off the one-shot sync.
2. Gives an overall decimate by 64, See Table 18 for other values.
3. Value is application dependent.
4. “P” is the page number. The upper nibble should stay at “0¨.

The programmable PFIR coefficients are not used and do not need to be loaded. The user should change address 6 (Output Format Register) to reflect the desired serial or link output mode. The expected results for various decimation ratios are shown below:

Table 18: Test Output Sequence

Decimation Controls			Output Sequence			
Overall Decimation	Addresses 2, 3	Address 4	I_0	Q_0	I_1	Q_1
32	07, 00	71	8000	FD00	7F00	0300
64	0F, 00	63				
128	1F, 00	55				
256	3F, 00	51				
512	7F, 00	43				
1024	FF, 00	35				
2048	FF, 01	31				
100	18, 00	60	8000	FE00	7F00	0200

The output sequence is the same for all power-of-two decimations. Other decimation ratios, with the SCALE and BIG_SCALE values being the maximum which satisfy: $(\mathrm{SCALE} + 6 \times \mathrm{BIG\_SCALE}) \le (55 - 4\log_2 \mathrm{N})$, will result in sequences with the same I values, but with slightly different Q values.

7.9 OPTIMAL GAIN SETTINGS

The overall gain of the chip is the product of the CIC gain, the coarse gain, the CFIR gain, the PFIR gain and the final gain. Each of these components are:

$$\text{CIC gain} = N^4 2^{(\text{SCALE} + 6 \times \text{BIG\_SCALE} - 56)}$$

$$\text{Coarse gain} = 2^{\text{COARSE}}$$

$$\text{CFIR gain} = (1.0 + \text{NARROW} \times 0.97)$$

$$\text{PFIR gain} = \frac{\text{PFIR\_SUM}}{65536}$$

$$\text{Final gain} = \frac{G}{32}$$

The signal flows through these sections in the order given. The gain settings which optimize the dynamic range of the chip, are the ones that maximize the signal amplitude without clipping at the output of each of these processing stages. A conservative approach to gain would be to set each gain component so that the product of the gains at each processing point in the flow is less than or equal to unity.

The conservative approach, described above, is usually less than optimal. The optimal gain takes the following considerations into account.

7.9.1 Tuning Loss

The input to the chip can be described as a signal S(t) modulated to a center frequency of “w”. The input is, therefore, $S(t)\cos(wt) = S(t)(e^{jwt}+e^{-jwt})/2$. If the downconverter tunes to the frequency “w”, then the tuner output will be $S(t)(1.0+e^{-2wt})/2$. The filters will then reject the component at “-2w”, leaving just the signal S(t)/2. This loss of one-half amplitude can be compensated for by setting the overall gain equal to 2, not unity. The tuning gain loss occurs after the CIC filters, so the optimum gain approach is to use gain settings that keep the gain product after the coarse, CFIR, PFIR and final gain stages equal to 2.

7.9.2 Uniform Power Inputs

The gain can be further optimized if the user has control over the power levels of the signals in the input bands. If all of the signals in the input are close to equal power, then the gain of the downconverted signal can be boosted to maximize its dynamic range. For example, if there are “M” signals of equal power in the input band, then the amplitude of each signal is $\frac{1}{\sqrt{M}}$. This means that the gain can be boosted by a factor of $\sqrt{M}$ within the downconverter. The coarse gain can be used to add the additional gain.

Examples of applications which can use this feature are FM-FDM systems, cellular systems which use power control, and wireless local loop systems that fix the power level of each remote transmitter.

Appendix D: PitCREWjr

D.1 Introduction

The following pages are a reprint of the data sheet and specifications for the PitCREWjr RACEway interface chip set.

This page is intentionally blank

Interfacing to RACEway: PitCREWjr

- Used to interface between FIFOs and the RACEway protocol.
- Drives/receives a RACEway port directly.
- Simple master control, automatic slave response.
- Moves data at 160 MByte/sec peak and 140 MByte/sec sustained throughput.

Reference Documents

When using this application note refer to the following documents for more information:

- *RACEway Interlink – Data Link and Physical Layers*, VITA 5-1994, available from the VITA Standards organization (VSO)
- Cypress CY7C4245 4K x 18 Synchronous FIFO data sheet

The VHDL is free for the Pitcrew interface and is available from your local Cypress Field Applications Engineer or on the Cypress BBS (408) 943-2954.

General

PitCREWjr is a simple full-duplex on-ramp to the RACEway fabric. The device has a standard RACEway port and FIFO port. The controller functions either as a RACEway slave, moving data between RACEway and local FIFOs or as a RACEway master, again moving data between RACEway and local FIFOs. It connects to and drives a RACEway interlink port, directly providing all required handshaking and control signaling. PitCREWjr's local FIFO port consists of a 32-bit bidirectional data bus and control signals for moving data between PitCREWjr and industry-standard FIFO components. The data flow between the RACEway and FIFOs is shown in *Figure 1*. The PitCREWjr has no programmable internal registers. Internal PitCREWjr state machines assemble and disassemble the route, address, and data long words embedded in the RACEway protocol. RACEway mastering is accomplished by controlling a single input signal. *Figure 2* shows the block diagram for PitCREWjr and *Table 1* shows the driver and signal name description for each pin on the PitCREWjr controller.

FIFOs

The timing generated by PitCREWjr is designed to match with CY7C4245 4K x 18 synchronous FIFOs. PitCREWjr signals can be connected directly to data and control signals of these FIFO components as shown in *Figure 3*. The input FIFO PAE flag should be set to 2. The output FIFO PAF flag should be set at least 16 entries from full.

Slave Function

The slave function of PitCREWjr is accessed whenever an incoming RACEway transaction is received on the RACEway port (REQI is asserted to PitCREWjr) During a slave transaction, the PitCREWjr asserts a status output pin called "SLAVE," which indicates that the PitCREWjr slave state machine is active. When a route word is received from the RACEway, it is driven onto the FIFO data bus. A PitCREWjr output called "ROUTE" is asserted for one XCLKI clock to indicate that a valid route word is present. When an address word is received from the RACEway, PitCREWjr drives this address word onto the FIFO data bus. An output called "ADDR" is asserted by PitCREWjr for one XCLKI clock to indicate that a valid address word is present on the FIFO data bus. PitCREWjr then acknowledges the RACEway with "REPLYIO."

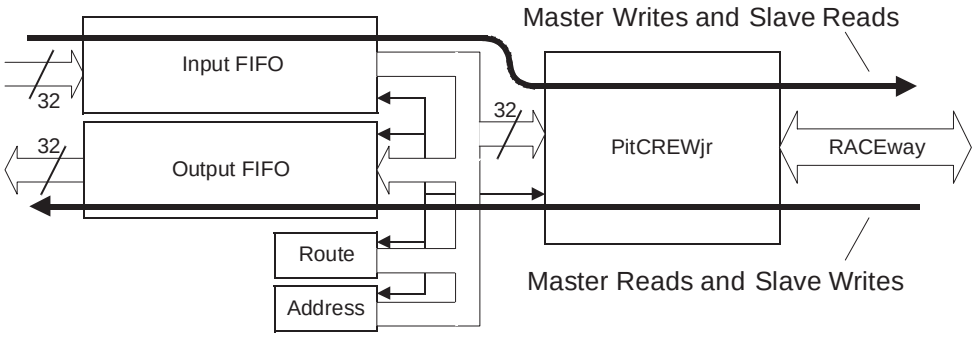


Figure 1. PitCREWjr Data Flow

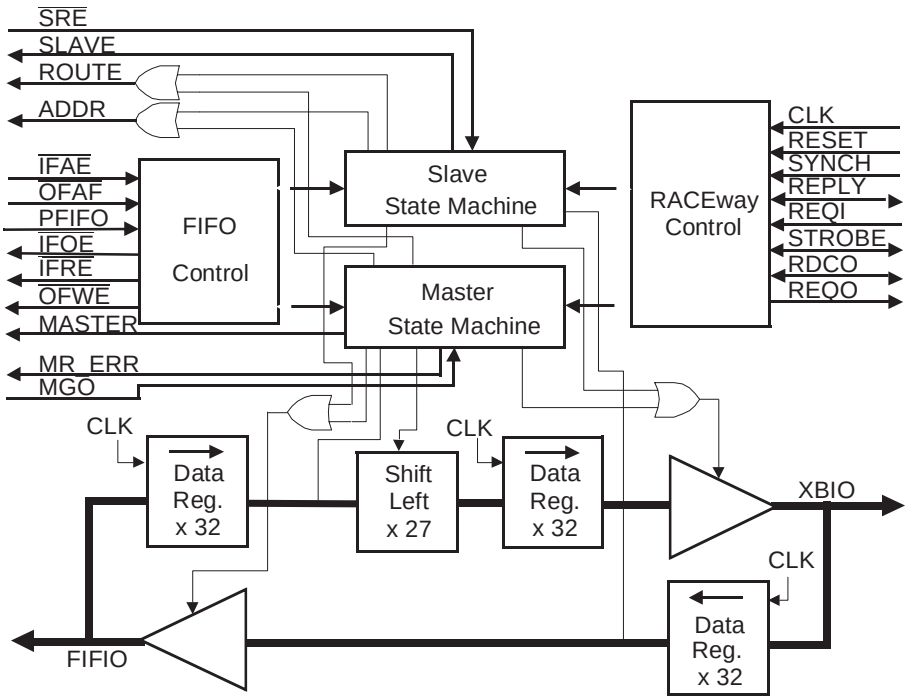


Figure 2. PitCREWjr Block Diagram

Table 1. PitCREWjr Interface Signals

Signal	Source	Function
FIFIO[31:0]	PitCREWjr/Input FIFO	FIFO Data Bus
XBIO[31:0]	PitCREWjr/RACEway	RACEway Data Bus
CLK	RACEway	Crossbar clock
RESET	RACEway	Reset from RACEway
SYNC	RACEway	Crossbar Sync - Provides control and phase information
REPLY	PitCREWjr/RACEway	Gives permission to send the address or data over the data bus
REQI	RACEway	Request In indicates the RACEway crossbar is requesting control of the data bus
STROBE	PitCREWjr/RACEway	Strobe indicates address or data is being sent on the data bus.
$\overline{\text{RDCO}}$	PitCREWjr/RACEway	Indicates to the crossbar to three-state the data bus so read data can be driven. It also indicates when a read error has occurred.
REQO	PitCREWjr	Request Out indicates the PitCREWjr is requesting control of the data bus
$\overline{\text{OFAF}}$	Output FIFO	Output FIFO almost full
$\overline{\text{OFWE}}$	PitCREWjr	Output FIFO write enable
PFIFO	User Hardware	Program output FIFO almost full flag
$\overline{\text{IFAE}}$	Input FIFO	Input FIFO almost empty
$\overline{\text{IFOE}}$	PitCREWjr	Input FIFO output enable
$\overline{\text{IFRE}}$	PitCREWjr	Input FIFO read enable
COUNT	PitCREWjr	Byte counter for master transfers
MR_ERR	PitCREWjr	Error occurred on a master read
MGO	User Hardware	Master GO - starts master state machine
SLAVE	PitCREWjr	Slave transaction in progress
$\overline{\text{SRE}}$	User Hardware	Slave read enable
ROUTE	PitCREWjr	PitCREWjr expecting route to be placed in FIFO data bus
ADDR	PitCREWjr	PitCREWjr expecting address to be placed on FIFO data bus
MASTER	PitCREWjr	Master transaction in progress

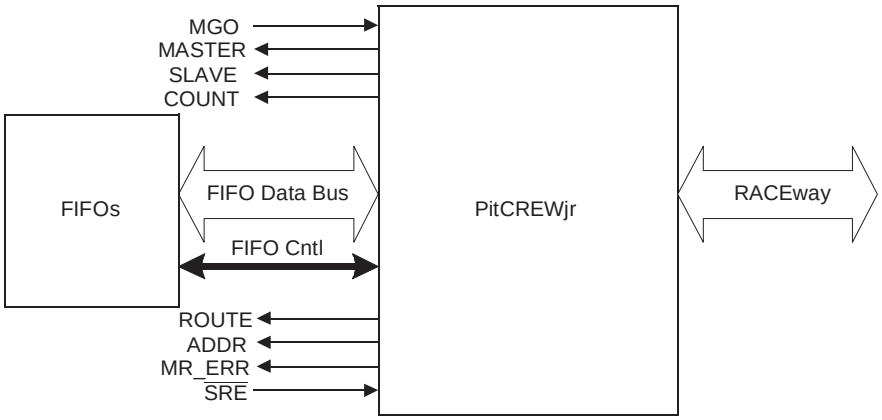


Figure 3. PitCREWjr Signals

The RACEway protocol communicates data direction in bit 1 of the address word. PitCREWjr's slave state machine branches on this bit value. If the direction of the data is from the RACEway to the local FIFO, the transaction is a slave write (bit 1 of address word is false). As data arrives from the RACEway, it is registered and driven onto the FIFO data bus. (See *Figure 2*.) The PitCREWjr writes the data received from the RACEway to the output FIFO by asserting "$\overline{\text{OFWE}}$" each time a valid word is ready on the FIFO data bus. A PitCREWjr input called "$\overline{\text{OFAF}}$" is used to indicate to PitCREWjr that the output FIFO is full. Assertion of "$\overline{\text{OFAF}}$" causes PitCREWjr to send a kill request to the RACEway master, effectively ending the RACEway transaction. "$\overline{\text{OFAF}}$" would typically be connected to the output FIFO programmable almost full flag. On completion of the RACEway data transfer, PitCREWjr three-states the FIFO data bus and deasserts the "SLAVE" status output.

If the direction of the data is from the input FIFO to the RACEway (a slave read, bit 1 of address word is true), then the FIFO data bus is three-stated by PitCREWjr and PitCREWjr asserts the signal "$\overline{\text{IFRE}}$" and then "$\overline{\text{IFOE}}$" to enable data from the input FIFO onto the FIFO data bus. PitCREWjr asserts this signal pair each time a new word is required from the FIFO. If the input FIFO becomes empty, as signaled by the "$\overline{\text{IFAE}}$" PitCREWjr input, PitCREWjr stops reading the input FIFO for the balance of that transaction and issues an error signal to the RACEway master on completion of the transaction. The kill request is also sent in this case, so that the master ends the transaction soon after the underflow. On completion of the RACEway data transfer, PitCREWjr deasserts the "SLAVE" status output.

The intent of the "SLAVE" pin is to indicate a slave transaction in progress. It can be used to tag incoming data, select a data destination, or as a board logic control input.

Note that PitCREWjr will NOT cause route and address header words received from the RACEway to be written to the output FIFO. External logic would be required to place address and/or route words in the output FIFO.

Master Function

The master function of PitCREWjr is accessed whenever the "MGO" PitCREWjr input is asserted. The assertion of "MGO" launches the PitCREWjr master state machine. This state machine is clocked by the RACEway data clock "XCLKI". Two clocks after "MGO" is sampled asserted, PitCREWjr asserts its "ROUTE" output. Local board hardware should use "ROUTE" to enable a route word onto the FIFO data bus. PitCREWjr asserts its "MASTER" output when it drives this route word onto the RACEway and then drives the "shifted route" prescribed by the RACEway protocol. "MGO" should be deasserted once PitCREWjr's "MASTER" output is true. This is because "MGO" will cause a slave in progress to issue a kill over the RACEway. When "change to address" reply is received from the RACEway, "ROUTE" is deasserted, and one clock later "ADDR" is asserted. Local board hardware should use "ADDR" to enable an address word onto the FIFO data bus. PitCREWjr relays the address word to the RACEway and waits for a "$\overline{\text{DSE}}$" reply from the RACEway. When the reply is received, PitCREWjr deasserts the "ADDR" signal.

The RACEway protocol communicates data direction in bit 1 of the address word. PitCREWjr's master state machine branches on this bit value. If the direction of the data is from the local FIFO to the RACEway (a master write, bit 1 of address word is false), then data is read from the local input FIFO, registered inside the PitCREWjr, and driven onto the RACEway XBIO bus. The PitCREWjr FIFO data bus pins remain three-stated and PitCREWjr asserts the signals "$\overline{\text{IFRE}}$" and "$\overline{\text{IFOE}}$" to enable the input FIFO data onto the FIFO data bus. PitCREWjr asserts this signal pair each time a new word is required from the FIFO. If the input FIFO becomes empty, as signaled by the "$\overline{\text{IFAE}}$" PitCREWjr input, PitCREWjr stops reading the input FIFO and ends the RACEway transaction.

If the direction of data is from the RACEway to the local FIFO (a master read, bit 1 of address word is true), then as data arrives from the RACEway, it is registered inside the PitCREWjr and driven onto the FIFO data bus. The PitCREWjr writes the data received from the RACEway to the output FIFO by asserting "OFWEN" each time a valid word is ready on the FIFO data bus. A PitCREWjr input called "$\overline{\text{OFAF}}$" is used to indicate to PitCREWjr that the output FIFO is full. Assertion of "$\overline{\text{OFAF}}$" causes PitCREWjr to suspend transfer requests to the RACEway slave, effectively stalling the RACEway transaction until the signal is deasserted. "$\overline{\text{OFAF}}$" would typically be connected to the output FIFO programmable almost full flag. On completion of the RACEway data transfer

as indicated by the deassertion of “MASTER,” PitCREWjr three-states the FIFO data bus.

Additional Features

A slave read enable input “$\overline{\text{SRE}}$” is provided to lock out slave access from the RACEway side of the interface. This signal may be used to “protect” data in the input FIFO when that FIFO is being used for both master and slave data. Slave read can be disallowed when data is being queued up in the input FIFO for a master write.

The “MR_ERR” output of the PitCREWjr is an indicator that a master read operation received an error response from its target slave. The signal is a “one-shot”, pulsing HIGH for one XCLKI clock period at the end of a master read access for which the RACEway slave signaled a read error.

The “COUNT” output signal strobes each time an 8-byte data beat occurs on the raceway when PitCREWjr is master. For writes, “COUNT” is asserted for each 8 bytes sent. For reads, “COUNT” is asserted for each 8 bytes requested.

The “PFIFO” input is used to assist in loading the output FIFO almost full flag. When “PFIFO” is asserted, PitCREWjr three-states its FIFIO data bus drivers, and asserts “$\overline{\text{OFWE}}$.” The signal that connects to “PFIFO” can also be used to enable the “almost empty” value onto the FIFIO data bus.

PitCREWjr Operation

Figure 4 illustrates master write behavior. The “MGO” PitCREWjr input is asserted to start RACEway master (read or write) function. It should be deasserted when PitCREWjr asserts “MASTER”. Master write is stopped by asserting “$\overline{\text{IFAE}}$” to the PitCREWjr. Notice that two data words are read after “$\overline{\text{IFAE}}$” is asserted. “ROUTE” and “ADDR” are shown enabling route and address information respectively onto the FIFIO data bus from external hardware. The “COUNT” PitCREWjr output pulses once for each 8 bytes sent over the RACEway.

Figure 5 illustrates master read behavior. Data arriving from the RACEway is to be taken from the FIFIO data bus on the rising edge of the RACEway data clock “CLK”. Again “COUNT” pulses once for each 8 bytes requested from the RACEway. Master read is stopped by asserting the PitCREWjr input “$\overline{\text{OFAF}}$.” Note that eight data values are delivered after “$\overline{\text{OFAF}}$” is signalled. This figure shows the timing when data traverses one RACEway crossbar. Latency will increase by two for each additional crossbar in the data path.

Figures 6 and *7* illustrate slave timing. “ROUTE” and “ADDR” PitCREWjr outputs mark the timing of valid route and address information on the FIFIO data bus. Bit 1 of the RACEway address field is captured by PitCREWjr, causing the appropriate FIFO control signalling for the data direction. For writes, “$\overline{\text{OFWE}}$” is asserted as data is driven by PitCREWjr onto the FIFIO data bus. For reads, “IFOEN” and “$\overline{\text{IFRE}}$” are asserted as shown and data is sampled from the FIFIO data bus on the rising edge of the RACEway data clock, “CLK”.

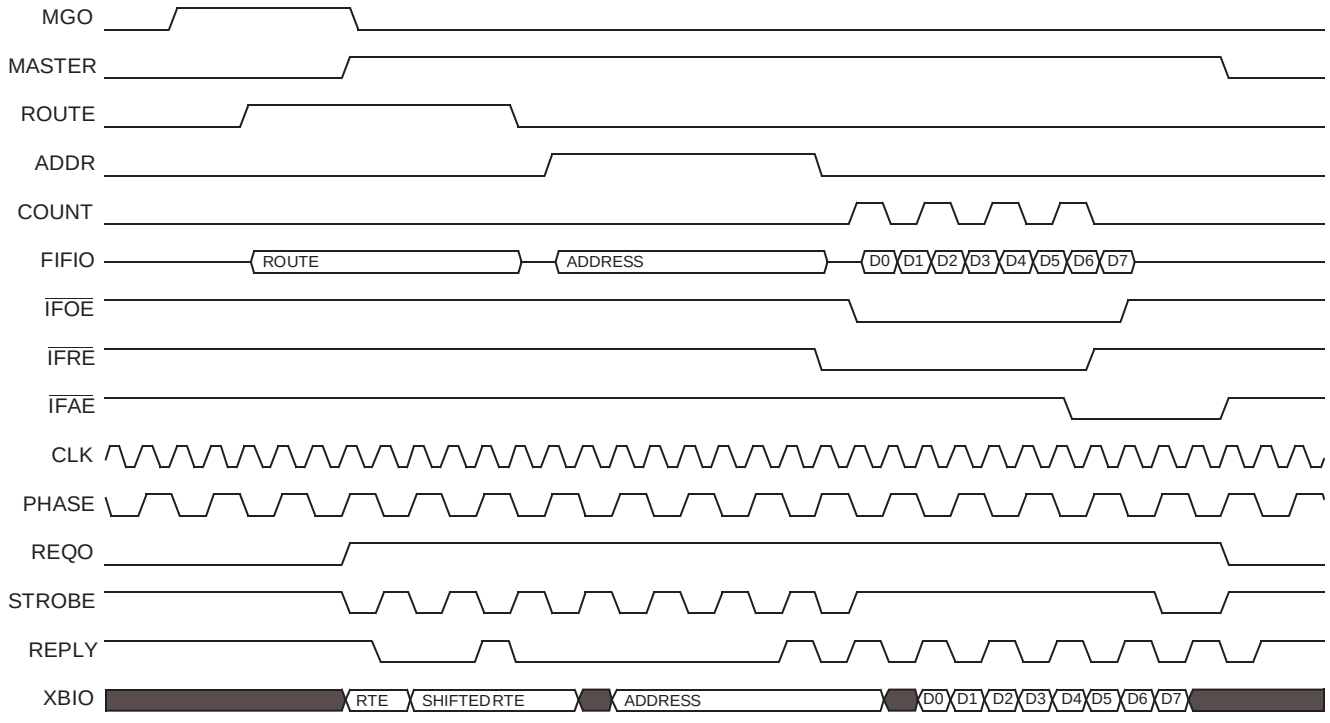


Figure 4. Master Write

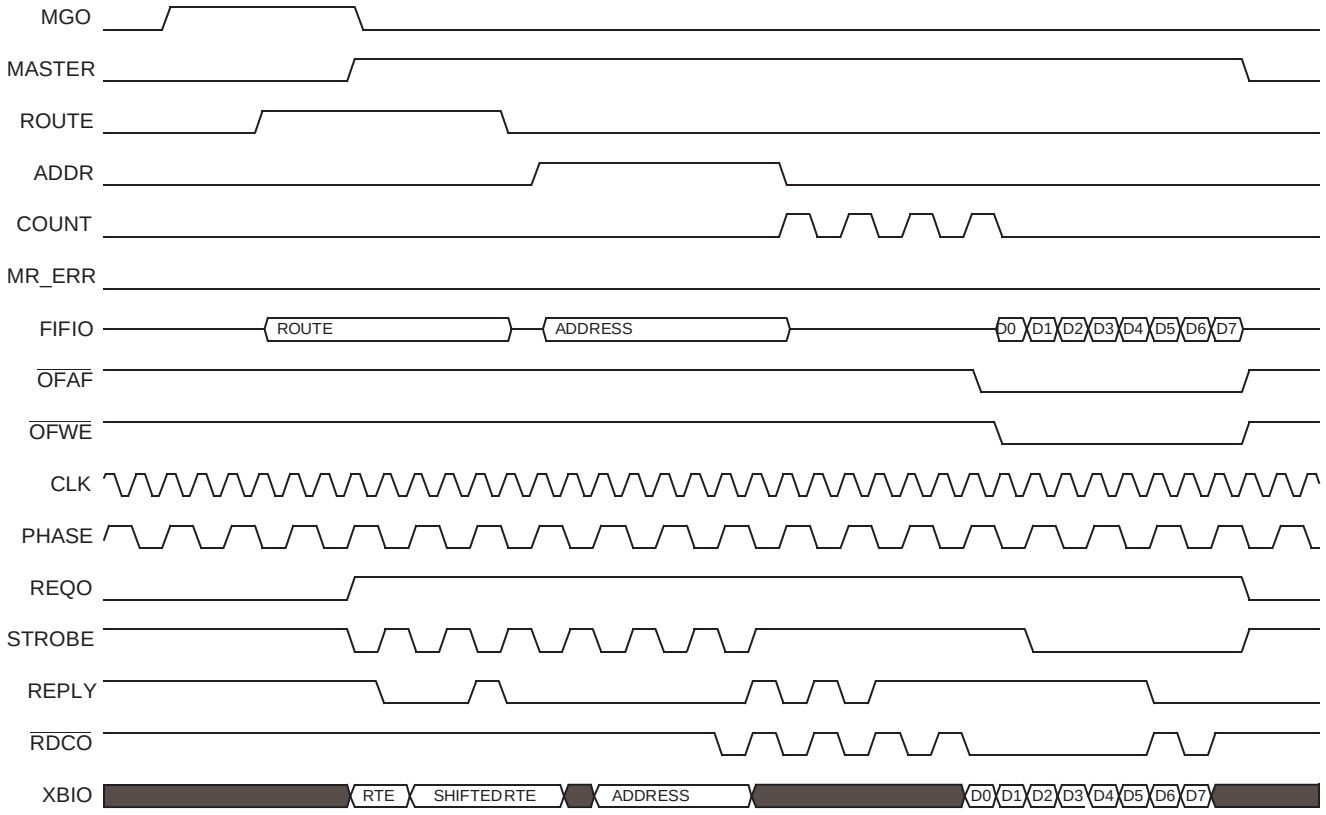


Figure 5. Master Read

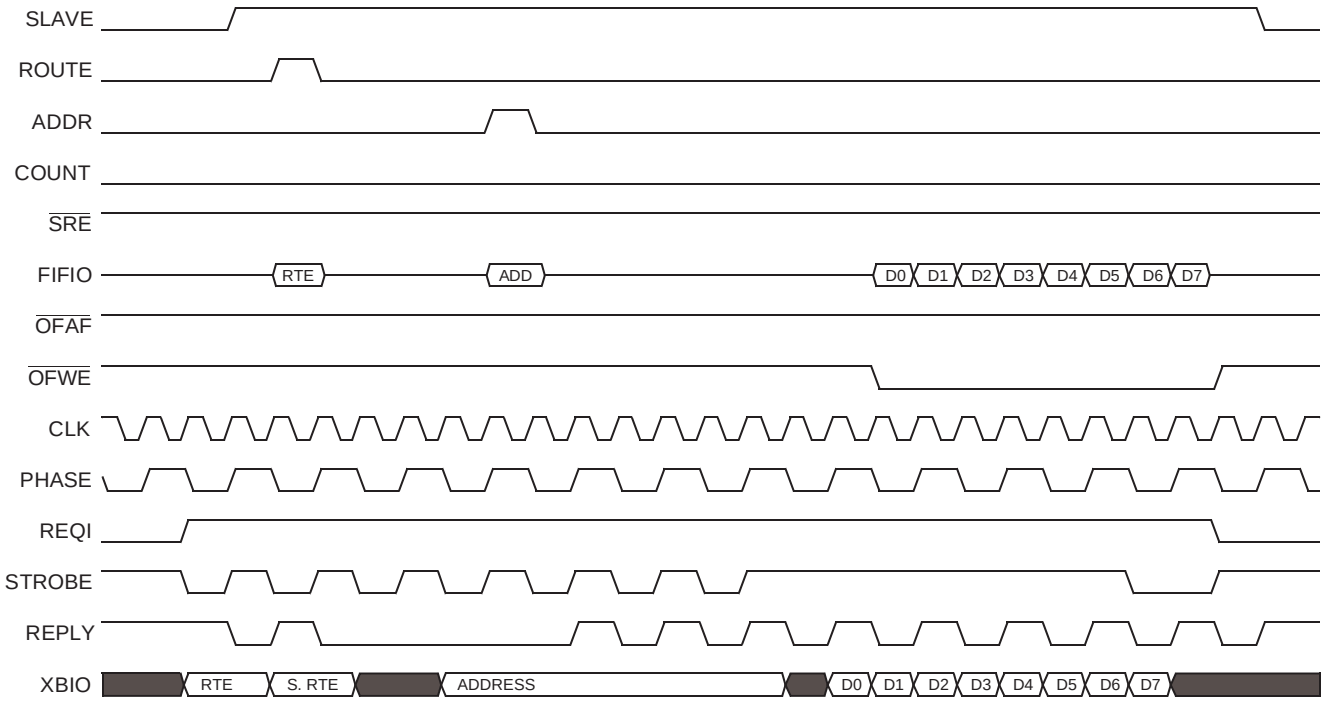


Figure 6. Slave Write

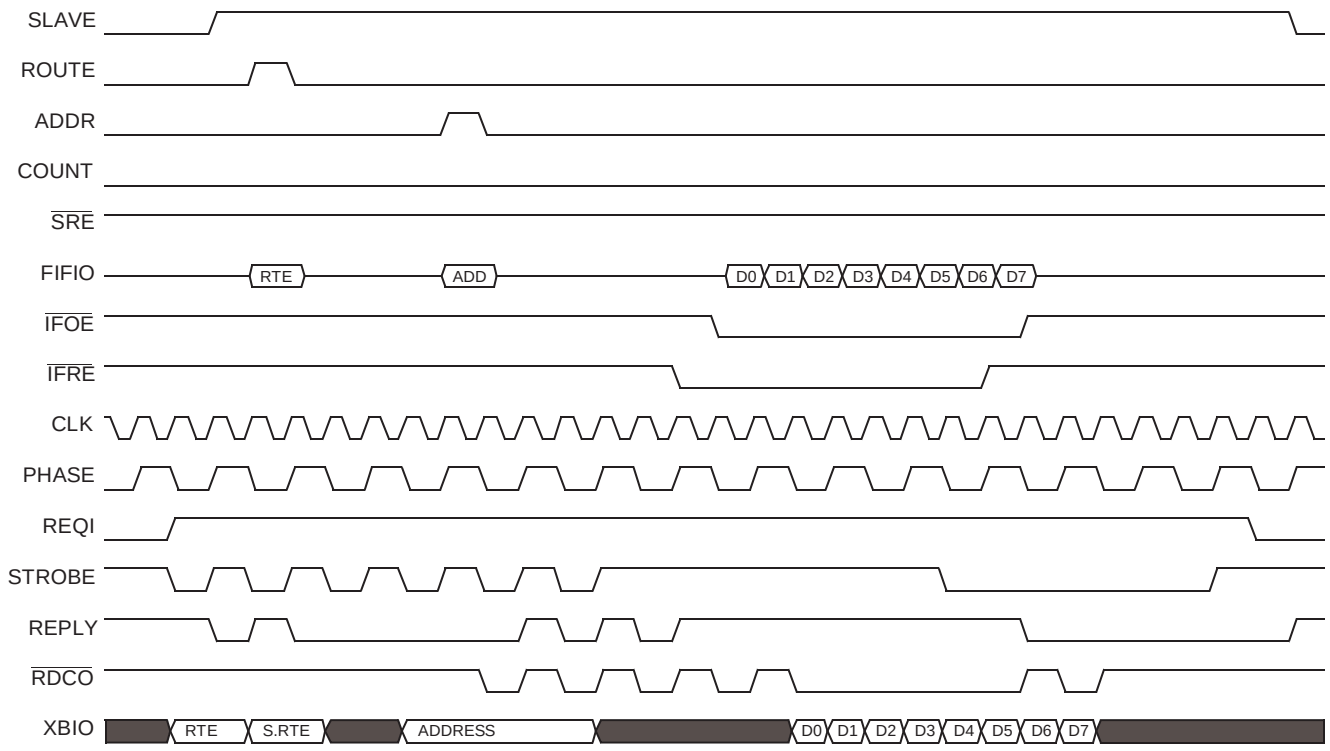


Figure 7. Slave Read

Figure 8 shows a PitCREWjr master writing to a PitCREWjr slave across one RACEway crossbar. The slave signals have an (S) suffix. In this example, the slave PitCREWjr input "$\overline{\text{OFAF}}$" signals that the slave is "almost full". The slave PitCREWjr signals "REQO" (a RACEway protocol kill). This kill propagates through the intervening RACEway crossbar to the PitCREWjr master, terminating the master transaction. The amount of data the slave must absorb after "$\overline{\text{OFAF}}$" is signalled is shown for a single intervening crossbar. Two additional FIFO write cycles will be required for each additional "crossbar hop".

Figure 9 shows the utility of the "$\overline{\text{SRE}}$" PitCREWjr input. It can be used to block PitCREWjr's response to a slave read from the RACEway. This feature allows the input FIFO facility to be multiplexed between master write and slave read without coordinating with the remote master across the RACEway. Master data being queued up in the input FIFO can be "protected" from a slave read operation as shown. The timing of "MGO" assertion with respect to slave arrival from the RACEway is arbitrary. "$\overline{\text{SRE}}$" may be deasserted any time after the assertion of "MASTER" by the PitCREWjr.

Figure 10 shows a PitCREWjr master reading from a PitCREWjr slave across one RACEway crossbar. The slave signals have an (S) suffix. In this example, the slave PitCREWjr input "$\overline{\text{IFAE}}$" signals that the slave is "almost empty." The slave PitCREWjr signals "REQO" (a RACEway protocol kill). This kill propagates through the intervening RACEway crossbar to the PitCREWjr master, terminating the master transaction. The slave PitCREWjr stops reading from its input FIFO two clocks after "$\overline{\text{IFAE}}$" is asserted; however, the RACEway protocol compels the slave to send until the RACEway master stops. By the time the PitCREWjr master responds to the kill, several long words of bad data have been written to the PitCREWjr master's output FIFO. The PitCREWjr output "MR_ERR" pulses HIGH for one data clock to signal that this error has occurred.

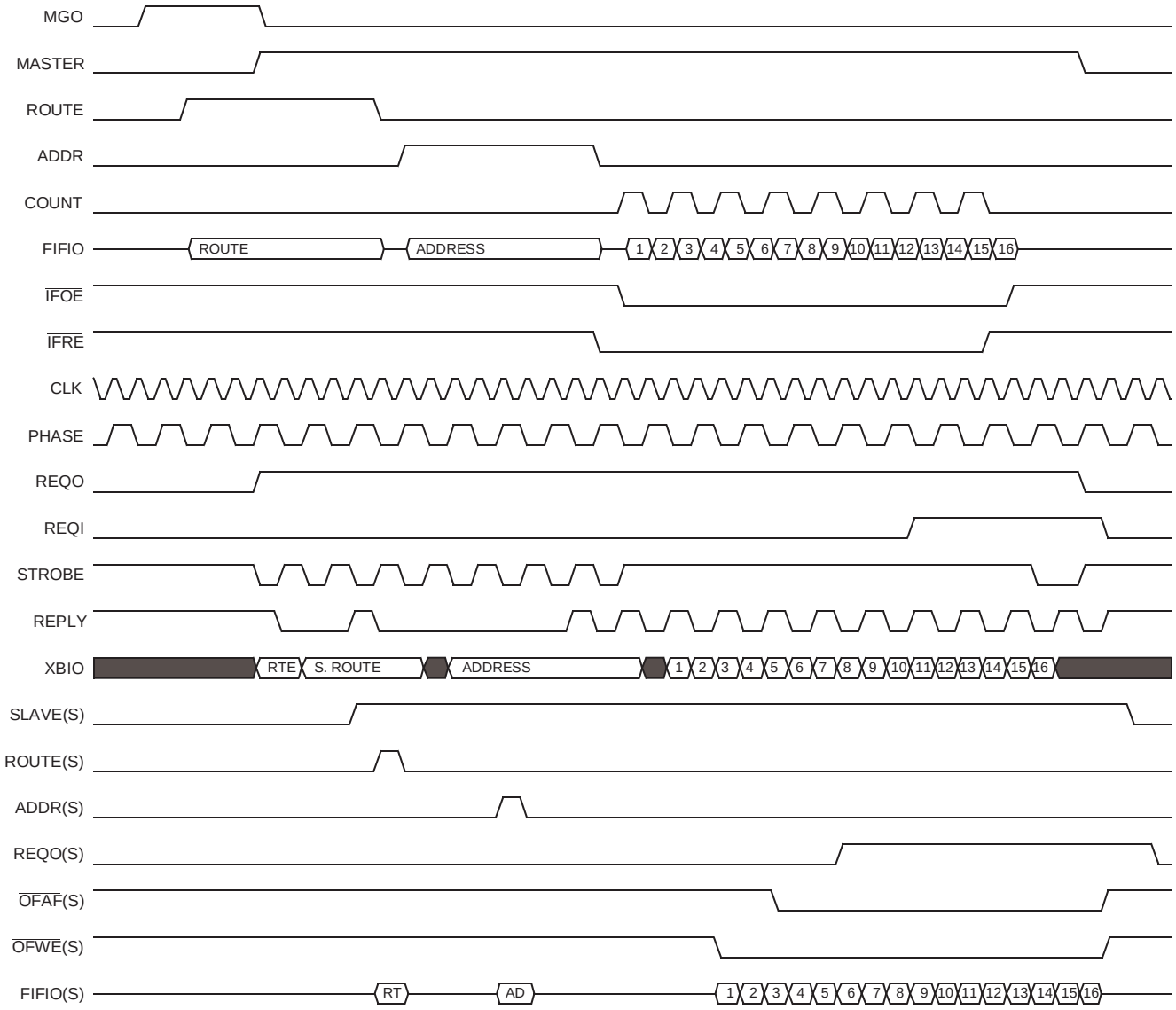


Figure 8. Master Write Overflow

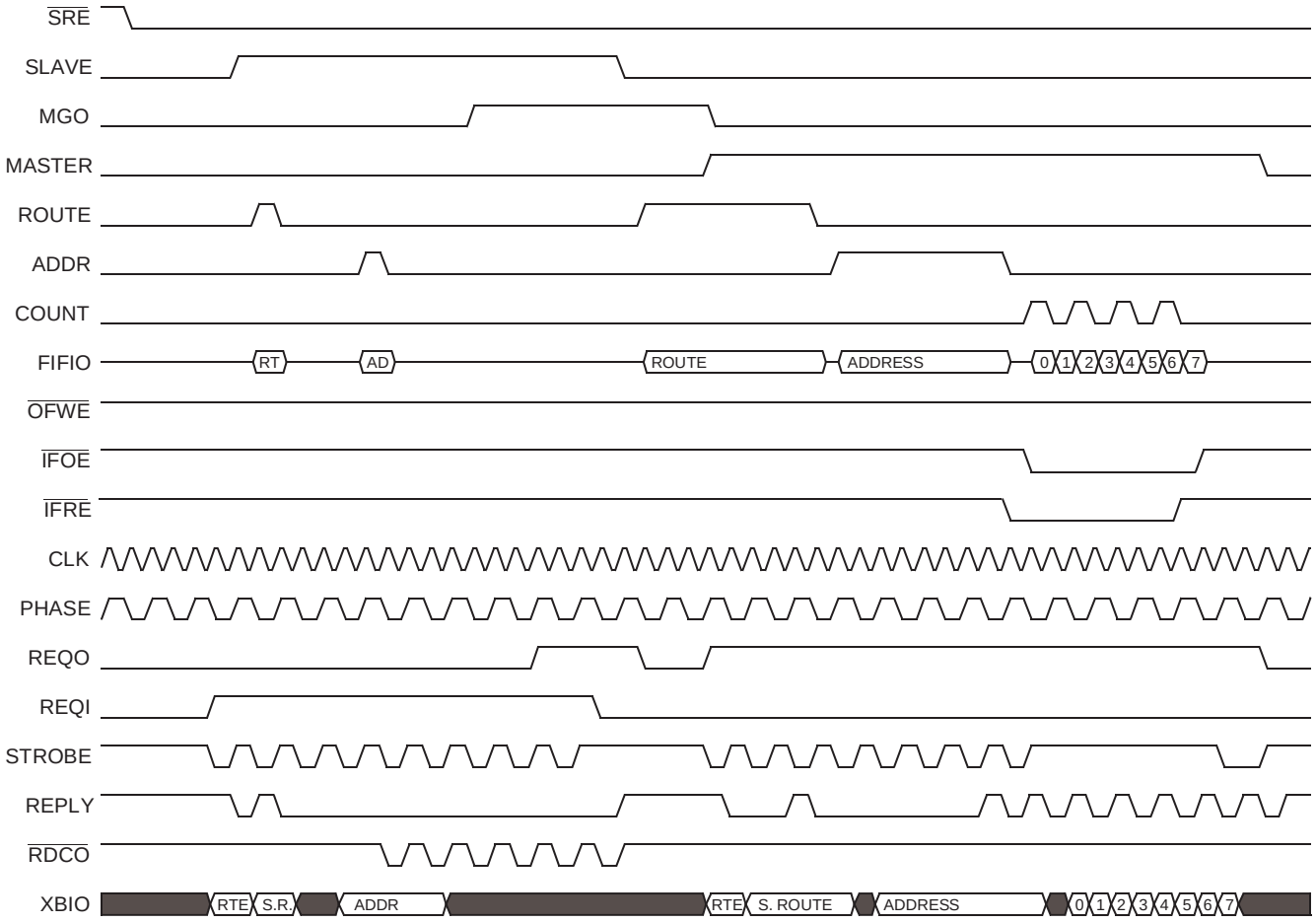


Figure 9. $\overline{\text{SRE}}$ Function

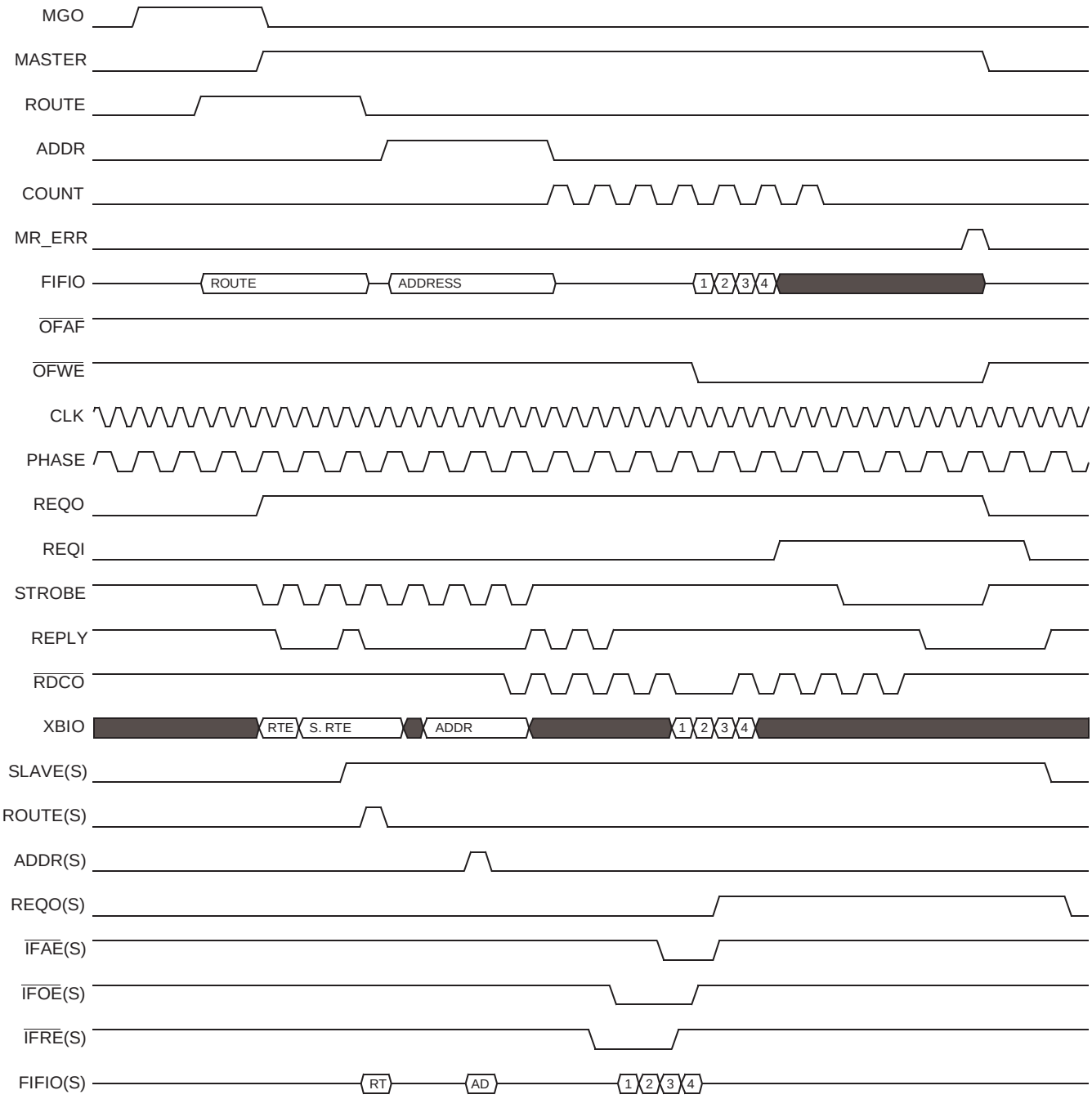


Figure 10. Master Read Error

This page is intentionally blank