11/1999
Rev 0.213

MPC8260 PowerQUICC II™ ADS User's Manual

PowerPC™

MOTOROLA

PowerQUICC II is a trademark of Motorola, Inc.

Mfax is a trademark of Motorola, Inc.

The PowerPC name, the PowerPC logotype, PowerPC 601, PowerPC 603, PowerPC 603e, PowerPC 604, PowerPC 604e, and RS/6000 are trademarks of International Business Machines Corporation used by Motorola under license from International Business Machines Corporation.

I^2C is a registered trademark of Philips Semiconductors

Information in this document is provided solely to enable system and software implementers to use PowerPC microprocessors. There are no express or implied copyright licenses granted hereunder to design or fabricate PowerPC integrated circuits or integrated circuits based on the information in this document.

Motorola Literature Distribution Centers:
USA/EUROPE: Motorola Literature Distribution; P.O. Box 5405; Denver, Colorado 80217; Tel.: 1-800-441-2447 or 1-303-675-2140
JAPAN: Nippon Motorola Ltd SPD, Strategic Planning Office 4-32-1, Nishi-Gotanda Shinagawa-ku, Tokyo 141, Japan Tel.: 81-3-5487-8488
ASIA/PACIFC: Motorola Semiconductors H.K. Ltd.; 8B Tai Ping Industrial Park, 51 Ting Kok Road, Tai Po, N.T., Hong Kong; Tel.: 852-26629298

Mfax™: RMFAX0@email.sps.mot.com; TOUCHTONE 1-602-244-6609; US & Canada ONLY (800) 774-1848;
World Wide Web Address: http://sps.motorola.com/mfax
INTERNET: http://motorola.com/sps

Technical Information: Motorola Inc. SPS Customer Support Center; 1-800-521-6274; electronic mail address: crc@wmkmail.sps.mot.com.
Document Comments: FAX (512) 895-2638, Attn: RISC Applications Engineering.
World Wide Web Addresses: http://www.mot.com/PowerPC/
http://www.mot.com/netcomm/

CONTENTS

CONTENTS

CONTENTS

CONTENTS

ILLUSTRATIONS

ILLUSTRATIONS

TABLES

TABLES

Chapter 1 General Information

1.1 Introduction

This document is an operation guide for the MPC8260ADS board. It contains operational, functional and general information about the MPC8260ADS. This board is meant to serve as a platform for s/w and h/w development around the MPC8260 processor. Using its on-board resources and a debugger, a developer is able to download code, run it, set breakpoints, display memory and registers and connect proprietary h/w via the expansion connectors, to be incorporated into a desired system with the MPC8260 processor.

This board could also be used as a demonstration tool (i.e., application s/w may be programmed[1] into its Flash memory and ran in exhibitions etc.).

1.2 Abbreviation List

- ADS - the MPC8260ADS, the subject of this document.
- UPM - User Programmable Machine
- GPCM - General Purpose Chip-select Machine
- GPL - General Purpose Line (associated with a UPM)
- BSCR - Board Control & Status Register.
- ZIF - Zero Input Force
- BGA - Ball Grid Array
- SIMM - Single In-line Memory Module
- DIMM - Dual In-Line Memory Module
- PBI - Page Based Interleaving
- T/ECOM - T1 or E1 Communication Tool, attachable to this board, via expansion connectors.

1.3 Related Documentation

- MPC8260 PowerQUICC II™ User's Manual
- MPC2605 Data Sheet
- PMC-SIERRA[2] 5350 Long Form Data Sheet
- PMC-SIERRA 5350 Errata Notice
- PMC-SIERA 5350 Reference Design
- PMC-SIERA 5350 Reference Design Errata
- LXT970A (by Level One) Data Sheet (http://www.level1.com/product/quickref.html#network)
- LXT970 Demo Board User's Guide (http://www.level1.com/product/quickref.html#network)

[1] Either on or off-board.
[2] Access to documents in this site requires registration.

1.4 Specifications

The MPC8260ADS specifications are given in Table 1-1 ". MPC8260ADS Specifications" below

Table 1-1. MPC8260ADS Specifications

CHARACTERISTICS	SPECIFICATIONS
Power requirements (no other boards attached)	+5Vdc @ TBD A (Typ.), 3.5 A (Max.) +12Vdc - @1A Max.
Microprocessor	MPC8260 running @ 66 MHz Bus Clock Frequency.
Addressing Total address range on PPC Bus: Total address range on Local Bus: Flash Memory SIMM (PPC Bus) Synchronous Dynamic RAM DIMM (PPC Bus) Synchronous DRAM On Local Bus	 4 Giga Bytes (32 address lines) 256 KBytes External (18 address lines) 4 Giga Bytes Internal (32 address lines internal decoding) 8 MByte, 32 bits wide expandable to 32 MBytes 16 MByte, 64 bits wide. Support for up to 64 MByte 4 MBytes, organized as 1 Meg X 32 bit.
Operating temperature	10^{O}C - 30^{O}C (room temperature)
Storage temperature	-25^{O}C to 85^{O}C
Relative humidity	5% to 90% (non-condensing)
Dimensions: Length Width Thickness	 11.023" (280 mm) 6.417" (163 mm) 0.063" (1.6 mm)

1.5 ADS Features

- 64-bit MPC8260, running @ 66MHz external bus frequency.
- 16 MByte, Unbuffered, 168 pin Synchronous Dram DIMM, residing on 60X bus, controlled by SDRAM machine 1. Support for 64 MBytes DIMM (single-bank[1] only). Optional address Latch - Multiplexer is required if an L2-cache module is assembled.
- Support for PBI (Page Based Interleaving) in both Single and 60X-Bus[2] modes.
- Support for Automatic DIMM Identification via MPC8260's I^2C port and DIMM's serial EEPROM.
- Optional 1/2 MByte L2-Cache on-board using 2 MPC2605 Lookaside cache modules.
- 8 MByte, 80 pin Flash SIMM, buffered from 60X bus. Support for upto 32 MByte, controlled by GPCM, 5V or 12V Programmable, with Automatic Flash SIMM identification, via BCSR. Support for both On and OFF-SIMM Flash reset.
- 5V/12V VPP for Flash SIMM, Jumper Selectable.
- 4 MByte unbuffered SDRAM on Local bus, controlled by SDRAM machine 2, soldered directly on board.
- Board Control & Status Register - BCSR, Controlling Board's Operation.
- Fast Download support via JTAG.
- Power-On Reset Option via JTAG.
- Programmable Power-On-Reset and Hard Reset Configurations via Flash memory, optionally provided by BCSR (dip-switch selectable), providing a rescue mode in case of inadvertent Flash erasure.
- Module Enable Indications for all on-board communication modules.
- High density (MICTOR) Logic Analyzer connectors, carrying all MPC8260 signals, for fast logic analyzer connection.
- 155 Mbps ATM UNI on FCC1 with Optical I/f, connected to the MPC8260 via UTOPIA I/F, using the PMC-SIERA 5350.
- 10/100-Base-T Port on FCC2 with T.P. I/F, MII controlled, using Level-One LXT970.
- Dual RS232 port residing on SCC1 & SCC2.
- Module disable (i.e., low-power mode) option for all communication transceivers -BCSR controlled, enabling use of communication ports, off-board via expansion connectors.
- Dedicated MPC8260's communication ports expansion connectors for convenient tools' connection, carrying also necessary bus signals, for transceivers' M/P I/F connection. Use is done with 2 X 128 pin DIN 41612 receptacle connectors.

---

[1]The DIMM is unbuffered from the 60X bus and therefore should consume as small capacitive drive power as possible.
[2]BCSR controlled for 60X-Bus mode.

- External Tools' Identification & status read Capability, via BCSR.
- Power-On Reset Push - Button
- Soft / Hard[1] Reset Push - Button
- ABORT Push - Button
- Single[2] 5V Supply.
- Reverse / Over Voltage Protection for Power Inputs.
- Multi-Range MPC8260 Internal Logic supply. Ranges include - 1.7V to 1.9V, 1.8V to 2.0V and 2.3V - 2.7V, currently changeable within a range.
- Software Option Switch provides 8 S/W options via BCSR.

1.6 Revision Engineering (ENG) to Revision PILOT Changes

This section describes only the functional changes between the above revisions. It does not contain numerous parts reference designation changes nor parts' value changes etc'. These changes reflect only on the schematics and bill-of-material.

1.6.1 BCSR

- BCSR address space was doubled to eight 32-bit registers, part of which is still reserved to allow for future expansion.
- BCSR0 and BCSR1 were moved to D(0:7) instead of D(24:31).
- Added optional Power-On and Hard Reset configurations, dip-switch enabled, with full MODCK control via dip-switches.
- On BCSR0 added 2 control bits - PBI and DIMM_SIZE, to provide PBI support when working with L2-Cache. These bits have no use in Single-MPC8260 mode.
- Added support for Fast Download via JTAG (BCSR6 & BCSR7), with internal and external bypass options for ENG compatibility. The ADS wakes up in ENG compatibility mode.
- Added support for Power-On Reset via JTAG.
- Codes for L2Cache size were changed: 'X0' are now reserved, '11' - No L2Cache, '01' - 512K L2Cache.

1.6.2 SDRAM DIMM (PPC-Bus)

- Added PBI support. To allow this, the addressing scheme was changed so that BK-SEL(0:2) are connected to (NC,BA1,BA0) correspondingly, rather than -(BA1,BA0,A11) correspondingly. In addition, for 60X mode PBI support, the Latch-Mux was enlarged and qualified by PBI and DIMM_SIZE bits in BCSR0.

[1]Hard reset is applied by depressing BOTH Soft Reset & ABORT buttons.
[2]Unless a 12V programmable Flash SIMM is being used.

1.6.3 Flash SIMM

- Low order address lines are connected to BADDR(27:29) lines. This, to allow operation in 60X mode. However, due to rev 0.X errata[1], **Power On Reset configuration with rev 0.X 8260s, must be taken from BCSR**.
- Added Power-On Reset connection to SIMMs Reset input to allow use of SIMMs which require external reset.

1.6.4 Power

- VDDL may now be regulated within 3 voltage ranges, jumper selectable:
  - 2.3V to 2.7 (original MPC8260 spec)
  - 1.7V to 1.9V (HIP4 spec)
  - 1.8V to 2.0V (2V capable HIP4 devices)
- 3.3V bus, which drives also VDDH bus, may be optionally (production option) regulated between 3.0 to 3.3V.
- Better Heat-Sinking for both Power Regulators.

1.6.5 Communication Ports

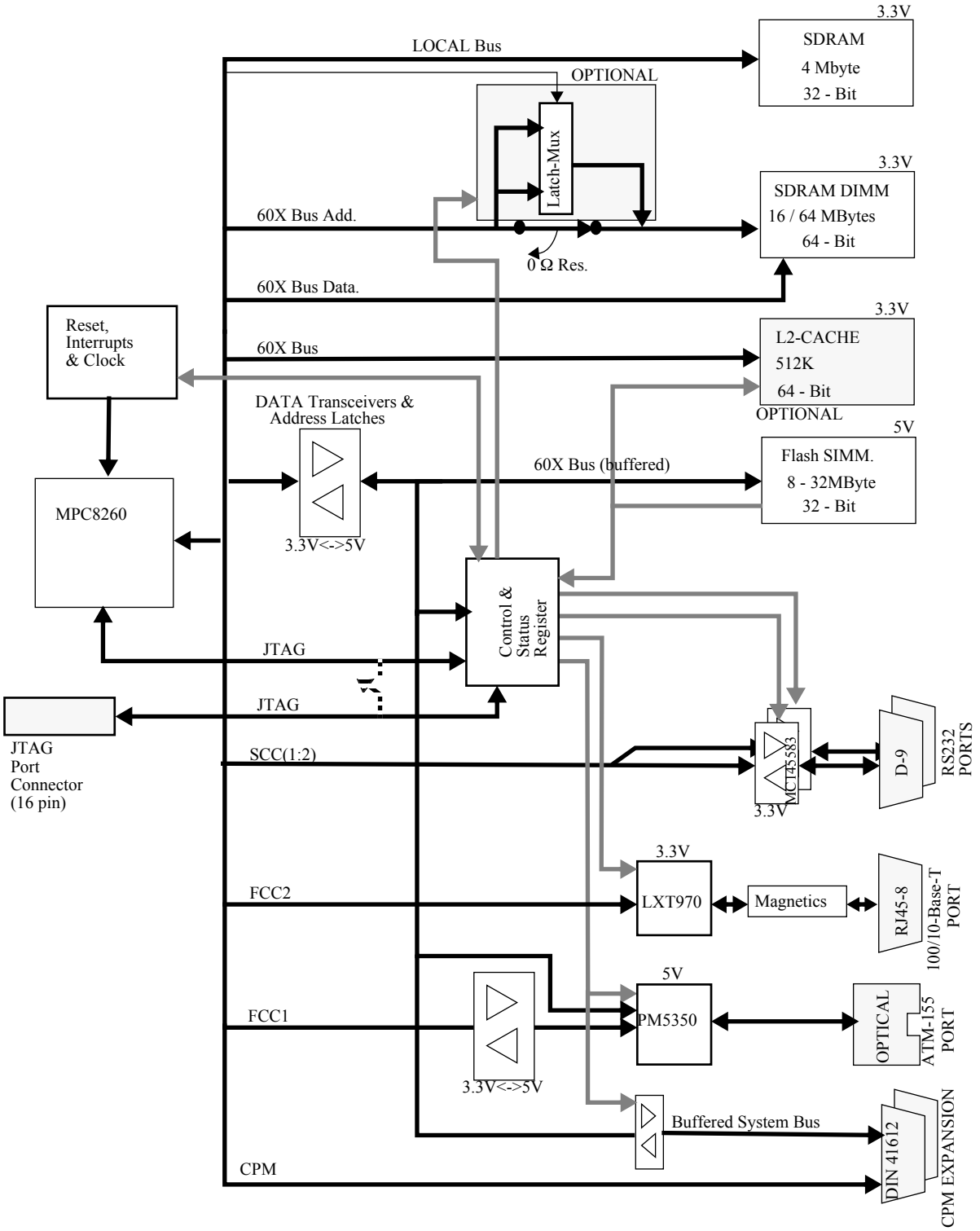
- Unified ATM transceiver's receive and transmit fifo clocks (ATMRFCLK & ATMTF-CLK) into ATMFCLK.
- ATM Rx and Tx indication leds are now operational only when ATM is enabled via BCSR.
- ATMRCLK signal (previously unused) is connected to P4 for SRTS support. When an ENG revision T/ECOM boards is connected to the ADS, this signal is tri-stated.
- RS232 ports' (1,2) CTS~ lines are now controllable by S/W, via PI/O lines.

1.6.6 Miscellaneous Changes

- ALE signal design is changed to allow proper operation in both Single / 60X bus modes (production configured).
- Added more ground lines to P5 - JTAG connector, to provide better noise immunity.
- THERM(0:1) signals are detached from GND plane and available at a dedicated header - J3.
- Reduced most MPC8260's damping resistors to 22 Ω.
- HP Logic Analyzer POD's Shrouds may now be soldered in place.

[1] I.e., BADDR lines are not active during Hard Reset Configuration Acquirement.

Figure 1-1. MPC8260ADS Block Diagram



Chapter 2 Hardware Preparation and Installation

2.1 Introduction

This chapter provides unpacking instructions, hardware preparation, and installation instructions for the MPC8260ADS.

2.2 Unpacking Instructions

NOTE

If the shipping carton is damaged upon receipt, request carrier's agent to be present during unpacking and inspection of equipment.

Unpack equipment from shipping carton. Refer to packing list and verify that all items are present. Save packing material for storing and reshipping of equipment.

CAUTION

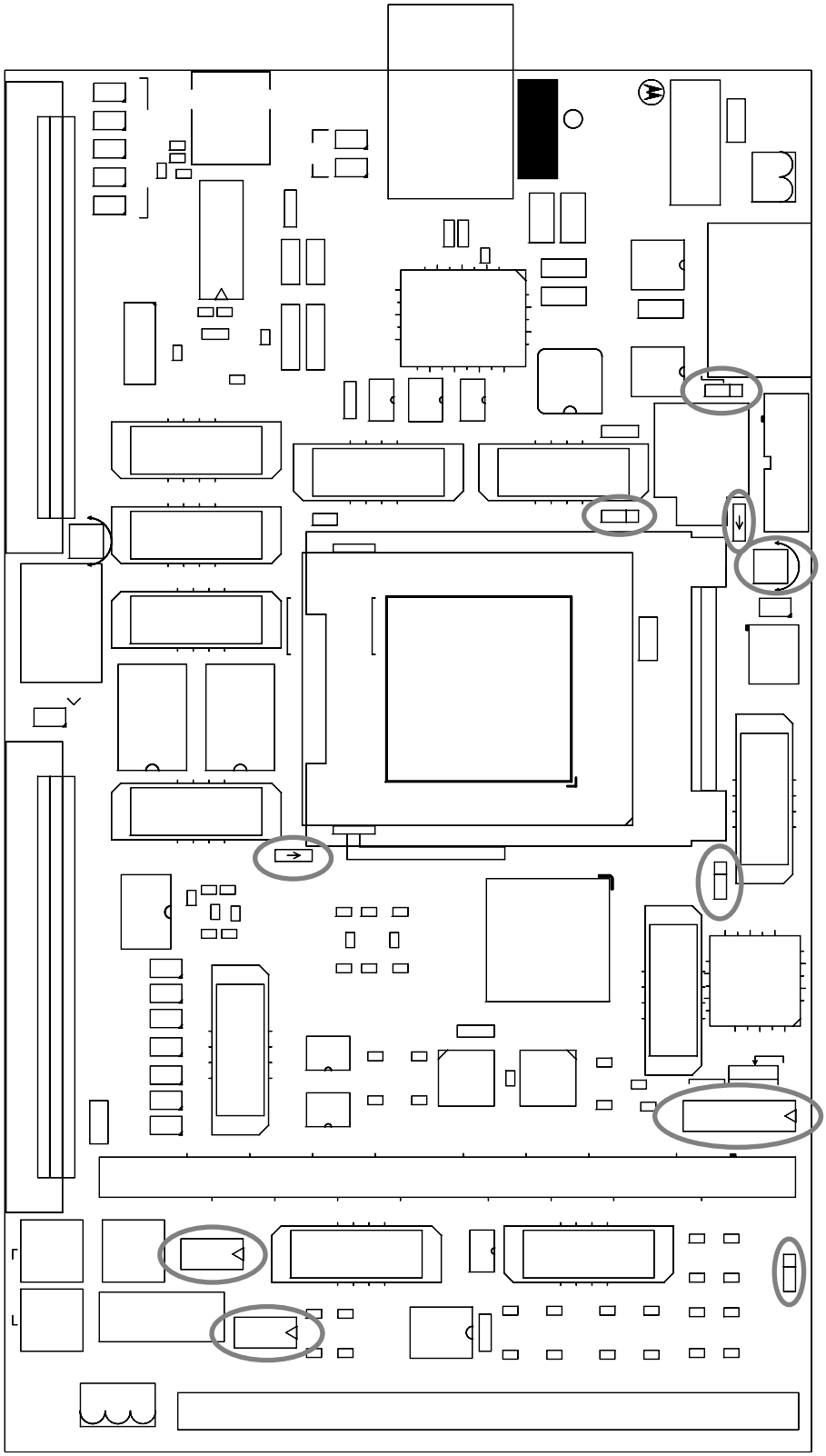
AVOID TOUCHING AREAS OF INTEGRATED CIRCUITRY; STATIC DISCHARGE CAN DAMAGE CIRCUITS.

2.3 Hardware Preparation

To select the desired configuration and ensure proper operation of the MPC8260ADS board, changes of the Dip-Switch settings may be required before installation. The location of the switches, indicators, Dip-Switches, and connectors is illustrated in Figure 2-1 ". MPC8260ADS Top Side Part Location Diagram" on page 18. The board has been factory tested and is shipped with Dip-Switch settings as described in the following paragraphs. Parameters can be changed for the following conditions:

- MPC8260's Internal Logic Supply Level Range Via J1.
- MPC8260's Internal Logic Supply Level within range (VDDL) Via TR1.
- MPC8260's MODCK(1:3). Determining Core's and CPM's PLLs multiplication factor via DS1.
- MPC8260's HARD Reset Configuration Source via DS1.
- MPC8260's MODCKH(0:3) via DS1 (Power-On Reset Source Dependent).
- MPC8260's JTAG's TDI Source Selection via J5.
- SDRAM DIMM's I^2C Slave Address via DS2.

Figure 2-1. MPC8260ADS Top Side Part Location Diagram

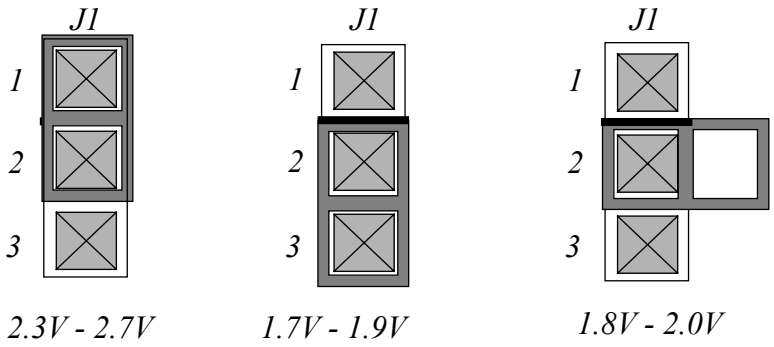


2.3.1 Setting VDDL Level Range - J1

To support future revisions of the MPC8260, provisions are taken to provide necessary voltage levels on VDDL, to match the process by which the MPC8260 is manufactured. Via J1, three voltage level ranges are provided:

1) When a jumper is placed between positions **1 - 2** of J1, a level range of **2.3V to 2.7V** on VDDL is selected. This level matches the current specification for the MPC8260.
2) When a jumper is placed between positions **2 - 3** of J1, a level range of **1.7V to 1.9V** on VDDL is selected. This level range is a preparation for the next revision of the MPC8260.
3) When a jumper is **misplaced** for J1, a level range of **1.8V to 2.0V** is selected for VDDL. This is in preparation for 2V capable future devices.

Figure 2-2. VDDL Range Selection - J1



WARNING

J1 is Factory Set according to the revision of MPC8260 with which it is assembled. Prior to changing a MPC8260 device, Extra Care should be taken with J1 setup. If a selected Voltage Range is above the specification for the newly inserted MPC8260, PERMANENT DAMAGE might be inflicted to the MPC8260.

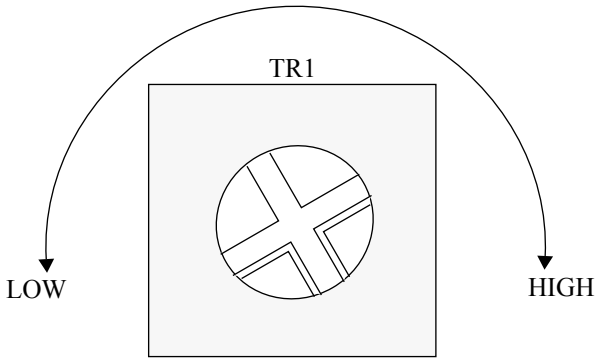
J1 selects only a range of Voltage levels on VDDL. The actual level is selected by TR1. See Setting VDDL Supply Voltage Level on page 19.

2.3.2 Setting VDDL Supply Voltage Level

After VDDL's Voltage Level Range is selected via J1, the actual level of VDDL is tuned via TR1. VDDL may be measured upon J2, using a DVM or any other high input impedance voltage measuring device.

VDDL level is factory set at the mid-range for the appropriate level range, but may be changed via TR1. Rotating TR1 CCW will reduce VDDL voltage down to range-low, while rotating it CW, will increase VDDL upto range-high. LD8, provides visual indication for VDDL level, it illuminates brighter with rise of VDDL. VDDL change Vs. TR1's rotation direction is shown in Figure 2-3 ". VDDL Trimmer - TR1" below:

Figure 2-3. VDDL Trimmer - TR1



Note

On ENG boards VDDL level changed in opposite direction regarding TR1 spin, i.e., it increases CCW and decreases CW.

WARNING

While in higher ranges of VDDL and higher ranges of internal operation frequencies, the MPC8260 might require some sort of COOLING measures to be taken. Failure in doing so, might result in PERMANENT DAMAGE inflicted to the MPC8260.

2.3.3 Setting MODCK(1:3) for PLLs Multiplication Factor - DS1 (#6 - #8)

After (1K cycles) the negation of the Power On Reset signal, the MPC8260 samples the 7 MODCK lines - the lower 3 on MODCK(1-3) and the upper four - MODCKH(0:3) field, is read from the Hard-Reset configuration source[1], to establish the multiplication factors of the CPM's and Core's PLLs. The levels on **MODCK(1:3)** lines are set using **DS1**, switches **#6 - #8**. When an individual switch is at the **OFF** position its associated MODCK line is pulled-**high** ('1'), while when at the ***ON*** position, the associated MODCK is pulled-***down*** ('0'). DS1 is shown in Figure 2-4 ". DS1 Description" on page 21, while the various combinations for DS1 #6 - #8 and their associated MODCK(1:3) values are shown in Table 2-1 ". MODCK(1:3) Encoding" on page 21.

[1]May be either boot memory or BCSR on the ADS.

Figure 2-4. DS1 Description

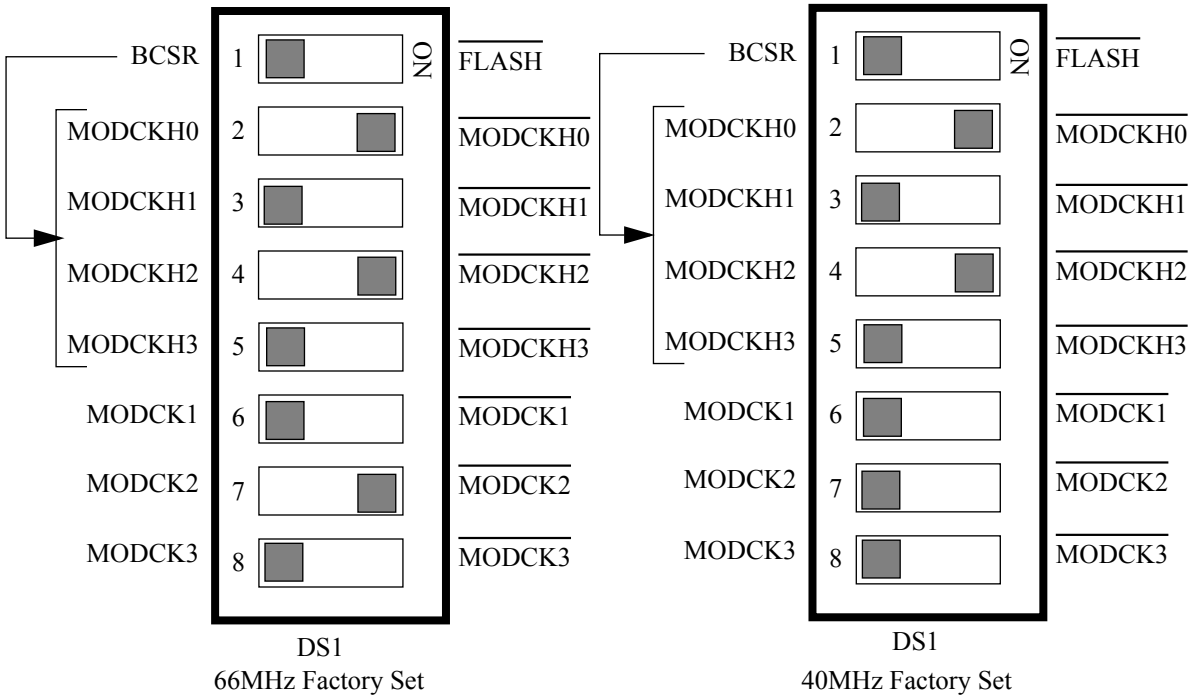


Table 2-1. MODCK(1:3) Encoding

MODCK(1:3)	Switch 6	Switch 7	Switch 8
0	ON	ON	ON
1	ON	ON	OFF
2	ON	OFF	ON
3	ON	OFF	OFF
4	OFF	ON	ON
5	OFF	ON	OFF
6	OFF	OFF	ON
7	OFF	OFF	OFF

2.3.4 Setting Hard - Reset Configuration Source

The Hard - Reset configuration word[1], read by the MPC8260 while HRESET~ is asserted, may be taken from two sources:

1) Flash Memory SIMM

[1]In fact 8 Hard-Reset configuration words are read by a configuration master, however only the first is relevant for a single MPC8260.

2) BCSR

For additional information as for the contents of the Hard-Reset configuration word see 4.1.2.4 "Hard Rest Configuration" on page 43.

When DS1 #1 is **OFF**, the Hard Reset configuration word is taken from **BCSR**, when it is ***ON***, the Hard Reset configuration word is taken from the ***Flash SIMM***.

Warning

With revision 0 of the MPC8260 on the PILOT revision of the ADS, DS1/1 MUST be set to OFF position for proper[1] operation of the ADS.

2.3.5 Setting MODCKH(0:3)[2] - for PLLs Multiplication Factors

When the Hard Reset configuration word is taken from BCSR, i.e., DS1 #1 is OFF, DS1 #2 - #5 become influential and set the upper 4 bits of the MODCK field, during Hard Reset Configuration word acquisition. When an individual switch of DS1 **#2 - #5** is at the OFF position, its corresponding MODCKH line is pulled-**high** ('1') during Hard Reset, while when at the ***ON*** position, pulled-***down*** ('0').

DS1 / #2 - #5 have no effect when DS1/ #1 is ON because the upper 4 bits of the MODCK field, are read from the Flash memory. See 4.1.1 "Power - On Reset" on page 42.

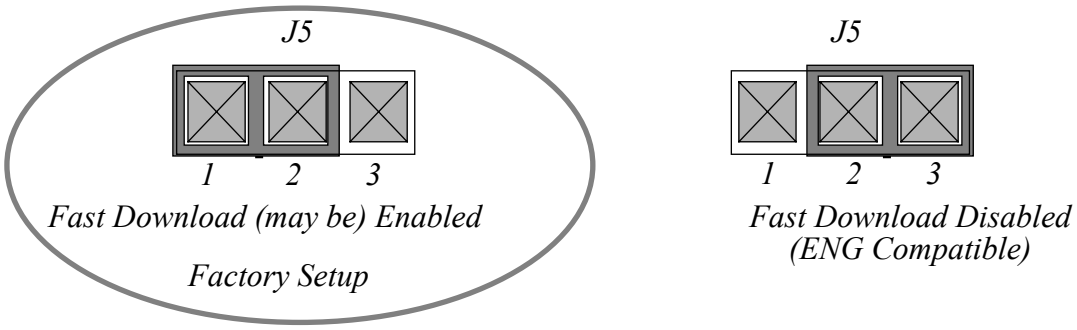
2.3.6 MPC8260 JTAG's TDI Source Selection - J5

On revision PILOT of this board, a new JTAG machine was inserted in front of the MPC8260's COP/JTAG port, this, to provide fast-download capability for the ADS. Since there are available debug tools designed for the previous (ENG) revision of the ADS, compatibility issues might arise.

Via J5, it is possible to bypass[3] the new JTAG machine and be perfectly compatible with the ENG revision of the ADS. When a jumper is placed between positions **1 - 2** of J5, then, the Fast-download JTAG machine may be **enabled** to precede the MPC8260 in the JTAG chain. When a jumper is placed between positions 2–3 of J5, the Fast-download JTAG machine is ***bypassed*** and the TDI input goes directly[4] from the COP/JTAG connector P5 to the MPC8260. See also 4.13.1 "Fast Download Support" on page 69.

J5 should be set between 2 - 3 if problems are encountered with the use of existing COP/JTAG debug equipment since this indicates that its software is not capable of using the fast download machine.

Figure 2-5. J5 - TDI Source Selection



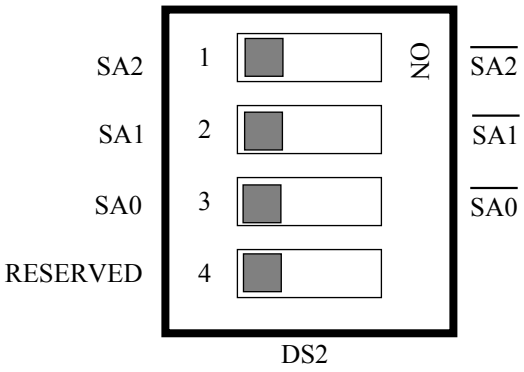
[1] See 4.8.4 "Hard - Reset Configuration - MPC8260 Revision Dependency" on page 55.
[2] Provided that DS1 #1 is in OFF position.
[3] HARD bypass. Even when the Fast download logic enabled via J5, it wakes-up asynchronously bypassed.
[4] Through a noise filtering network.

2.3.7 SDRAM DIMM I^2C Slave Address Selection - DS2

The SDRAM DIMM has a serial configuration EEPROM, which is accessed using I^2C protocol. Each slave device on that bus has an 7 bit address field, 3 of which (the LSBs) within this device may be set externally.

Since the SDRAM DIMM configuration is read using the MPC8260's I^2C port, which may be required for user's application, an option is given to change the SDRAM DIMM configuration EEPROM slave address for the convenience of the user. DS2 is shown in Figure 2-6 ". DS2 Description" on page 23:

Figure 2-6. DS2 Description



The various position combinations of DS2 and their associated SDRAM DIMM configuration EEPROM's I^2C slave addresses are shown in Table 2-2 ". DS2 - SDRAM DIMM Configuration EEPROM Slave Address" below:

Table 2-2. DS2 - SDRAM DIMM Configuration EEPROM Slave Address

Slave Address [bin]	Switch 1	Switch 2	Switch 3
1010**000**	ON	ON	ON
1010**001**	ON	ON	OFF
1010**010**	ON	OFF	ON
1010**011**	ON	OFF	OFF
1010**100**	OFF	ON	ON
1010**101**	OFF	ON	OFF
1010**110**	OFF	OFF	ON
1010**111**	OFF	OFF	OFF

DS2 is factory set to: 1 - ON, 2 - ON, 3 - ON.

2.4 Installation Instructions

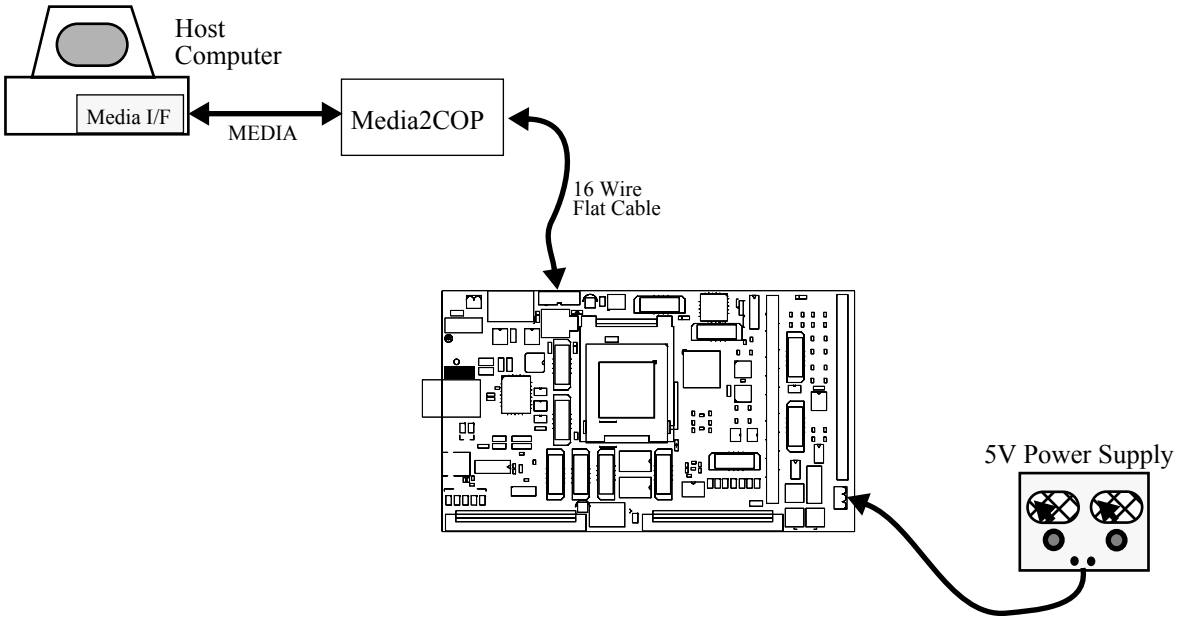
When the MPC8260ADS has been configured as desired by the user, it can be installed according to the required working environment as follows:

- Host Controlled Operation
- Stand-Alone

2.4.1 Host Controlled Operation

In this configuration the MPC8260ADS is controlled by a host computer via the COP port, which is a subset of the JTAG port. This configuration allows for extensive debugging using on-host debugger. The host is connected to the ADS by a COP controller provided by a third party.

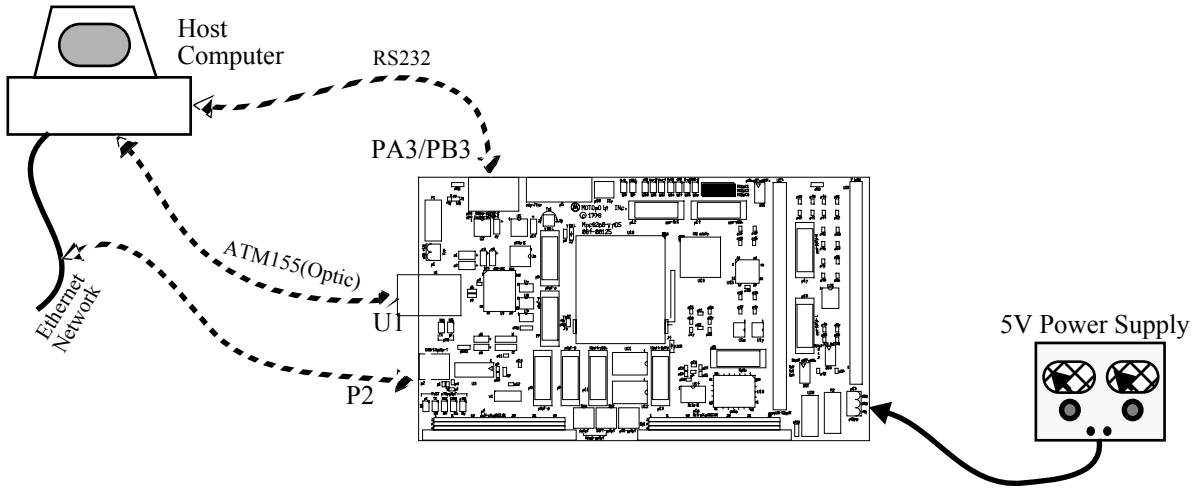
Figure 2-7. Host Controlled Operation Scheme



2.4.2 Stand Alone Operation

In this mode, the ADS is not controlled by the host via the COP port. It may connect to host via one of its other ports, e.g., RS232 port, Fast Ethernet port, ATM155 port etc. Operating in this mode requires an application program to be programmed into the board's Flash memory.

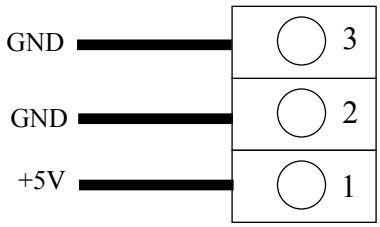
Figure 2-8. Stand Alone Configuration



2.4.3 +5V Power Supply Connection

The MPC8260ADS requires +5 Vdc @ 5 A max, power supply for operation. Connect the +5V power supply to connector P19 as shown below:

Figure 2-9. P19: +5V Power Connector



P19 is a 3 terminal block power connector with power plug. The plug is designed to accept 14 to 22 AWG wires. It is recommended to use 14 to 18 AWG wires. To provide solid ground, two Gnd terminals are supplied. It is recommended to connect both Gnd wires to the common of the power supply, while VCC is connected with a single wire.

NOTE

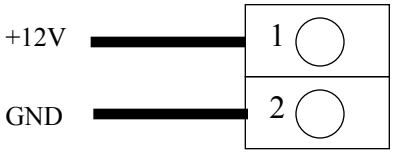
Since hardware applications may be connected to the MPC8260ADS via the expansion connectors P4 and P16, the additional power consumption should be taken into consideration when a power supply is connected to the MPC8260ADS.

2.4.4 P2: +12V Power Supply Connection

The MPC8260ADS may require +12 Vdc @ 1 A max, power supply for 12V programmable Flash SIMM. The MPC8260ADS can work properly without the +12V power supply, if there is no need to program a 12V programmable Flash SIMM.

Connect the +12V power supply to connector P2 as shown below:

Figure 2-10. P2: +12V Power Connector

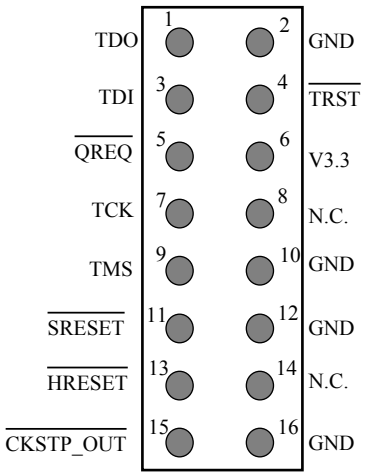


P2 is a 2 terminal block power connector with power plug. The plug is designed to accept 14 to 22 AWG wires. It is recommended to use 14 to 18 AWG wires.

2.4.5 COP/JTAG Connector - P5

The MPC8260ADS COP interface connector, P5, is a 16 pin, male, Header connector. The connection between the MPC8260ADS and the COP controller is by a 16 line flat cable, supplied with the COP controller board obtained from a third party developer. Figure 2-11 ". P5 - COP/JTAG Port Connector" below shows the pin configuration of the connector.

Figure 2-11. P5 - COP/JTAG Port Connector

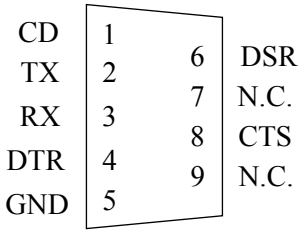


2.4.6 Terminal to MPC8260ADS RS-232 Connection

A serial (RS232) terminal or any other RS232 equipment, may be connected to the RS-232 connectors PA3 and PB3. The RS-232 connectors are a 9 pin, female, D-type connectors, arranged in a stacked configuration. **PA3** connected to SCC2 of the MPC8260 is the **lower** and ***PB3***, connected to SCC1 of the MPC8260, is the ***upper*** in the stack.

The connectors are arranged in a manner that allows for 1:1 connection with the serial port of an IBM-AT[1] or compatibles, i.e. via a flat cable. The pinout which is identical for both PA3 and PB3 is shown in Figure 2-12 ". PA3, PB3 - RS-232 Serial Port Connectors" on page 26.

Figure 2-12. PA3, PB3 - RS-232 Serial Port Connectors



2.4.7 10/100-Base-T Ethernet Port Connection

The 10/100-Base-T port connector - P1, is an 8-pin, 90^{o}, receptacle RJ45 connector. The connection between the 10/100-Base-T port to the network is done by a standard cable, having two RJ45/8 jacks on its ends. The pinout of P1 is described in Table 5-1 ". P1 - Ethernet Port Interconnect Signals" on page 78.

2.4.8 Memory Installation

The MPC8260ADS is supplied with two types of memory modules:

- Synchronous Dynamic Memory DIMM
- Flash Memory SIMM.

[1]IBM-AT is a trademark of International Business Machines Inc.

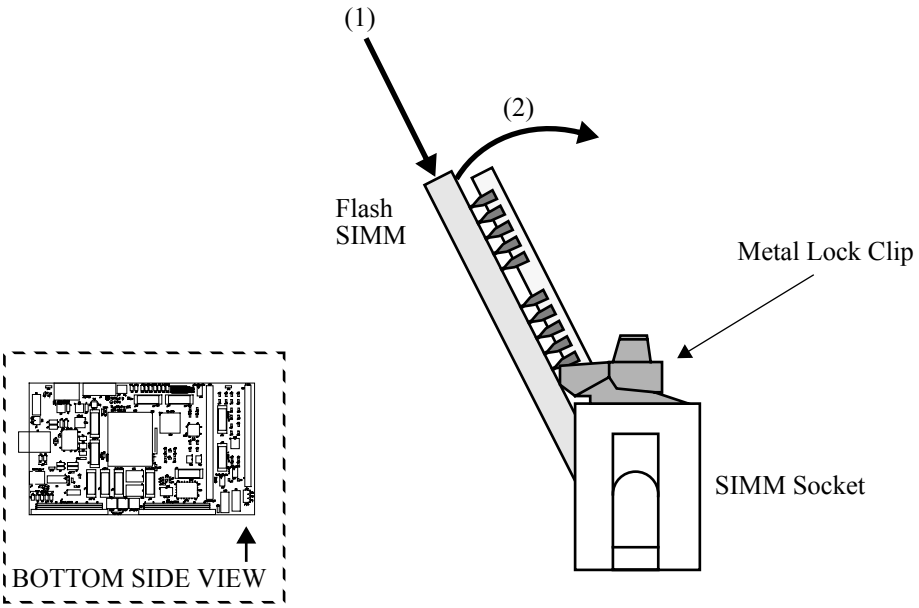
2.4.8.1 Flash Memory SIMM Installation

To install a memory SIMM, it should be taken out of its package, put diagonally in its socket - U25 (no error can be made here, since the Flash socket has 80 contacts, while the SDRAM socket has 168) and then raised to a vertical position until the metal lock clips are locked. See Figure 2-13 ". Flash Memory SIMM Insertion" on page 27.

CAUTION

The memory SIMMs have alignment nibble near their # 1 pin. It is important to align the memory correctly before it is twisted, otherwise damage might be inflicted to both the memory SIMM and its socket.

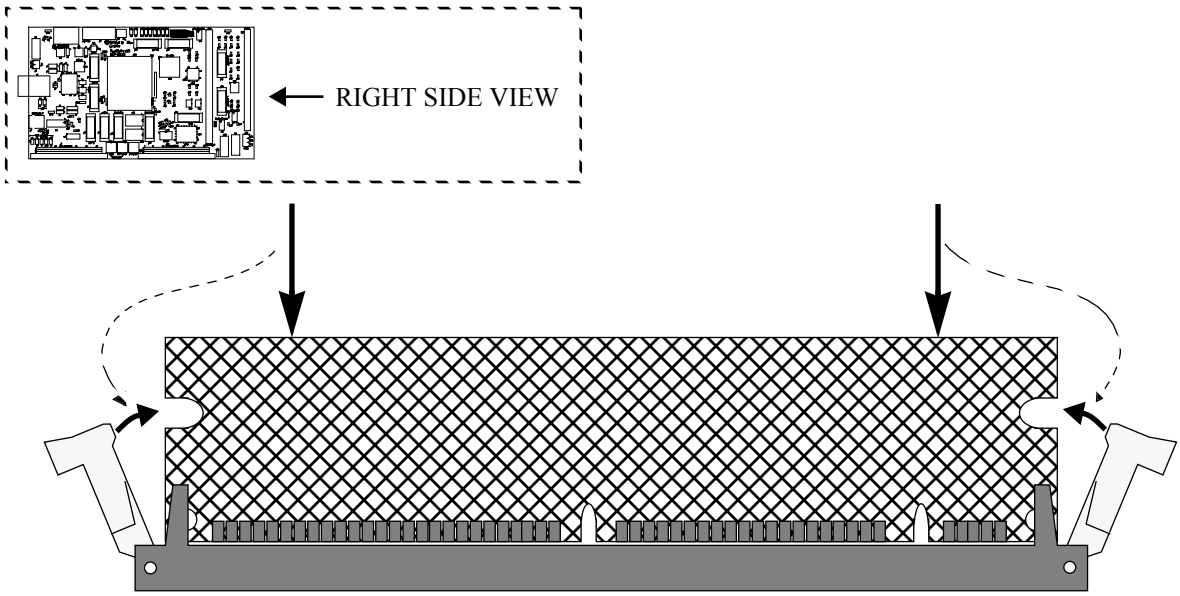
Figure 2-13. Flash Memory SIMM Insertion



2.4.8.2 SDRAM DIMM Installation

The SDRAM DIMM (U22) is inserted in a different manner: The 2 side-latches are pulled aside to unlocked position, the DIMM is placed in a vertical position (similar to its final position) between them, so that the keys, nibbled in the DIMM matches those on the socket and then, the DIMM should be pressed evenly and firmly into its place, locking the side locks on itself. The SDRAM insertion is shown in Figure 2-14 ". SDRAM DIMM Insertion" below:

Figure 2-14. SDRAM DIMM Insertion



Chapter 3 Operating Instructions

3.1 Introduction

This chapter provides necessary information to use the MPC8260ADS in host-controlled and stand-alone configurations. This includes controls and indicators, memory map details, and software initialization of the board.

3.2 Controls and Indicators

The MPC8260ADS has the following switches and indicators.

3.2.1 Power-On RESET Switch - SW1

The Power-On RESET switch SW1 performs Power-On reset to the MPC8260, as if the power was re-applied to the ADS. When the MPC8260 is reset that way, all configuration and all data residing in volatile memories are lost. After PORST~ signal is negated, the MPC8260 re-acquires the power-on reset and hard-reset configuration data from the hard-reset configuration source. (Flash | BCSR).

3.2.2 ABORT Switch - SW2

The ABORT switch is normally used to abort program execution, this by issuing a level 0 interrupt to the MPC8260. If the ADS is in stand alone mode, it is the responsibility of the user to provide means of handling the interrupt, since there is no resident debugger with the MPC8260ADS. The ABORT switch signal is de-bounced, and may be disabled by software.

3.2.3 SOFT RESET Switch - SW3

The SOFT RESET switch SW2 performs Soft reset to the MPC8260 internal modules, maintaining MPC8260's configuration (clocks & chip-selects) and SDRAMs' contents. The switch signal is debounced, and it is not possible to disable it by software.

3.2.4 HARD RESET - Switches - SW2 & SW3

When BOTH switches - SW2 and SW3 are depressed simultaneously, HARD reset is generated to the MPC8260. When the MPC8260 is HARD reset, all its configuration is lost[1], including data stored in the SDRAMs and the MPC8260 has to be re-initialized.

3.2.5 DS1 - Reset Configuration Switch

DS1 is a 8-switch Dip-Switch. For its function see 2.3.3 "Setting MODCK(1:3) for PLLs Multiplication Factor - DS1 (#6 - #8)" on page 20, 2.3.4 "Setting Hard - Reset Configuration Source" on page 21 and 2.3.5 "Setting MODCKH(0:3) - for PLLs Multiplication Factors" on page 22.

3.2.6 DS2 - SDRAM DIMM Configuration Memory I^2C Slave Address Switch

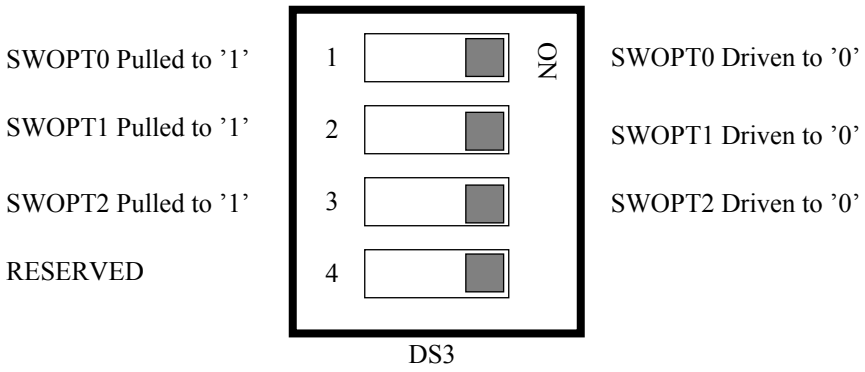
DS2 sets the I^2C slave address for the SDRAM DIMM's serial configuration memory. For its function see 2.3.7 "SDRAM DIMM I2C Slave Address Selection - DS2" on page 23.

[1] Except for Hard-Reset configuration word, which is acquired only once, after PON-Reset.

3.2.7 DS3 - Software Options Switch

DS3 is a 4-switch Dip-Switch. This switch is connected over SWOPT(0:2) lines which are available at BCSR2, S/W options may be manually selected, according to DS3 state. DS3 is factory set to all ON.

Figure 3-1. DS3 - Description



3.2.8 J1 - VDDL Voltage Level Range Selection

J1 selects between 3 different voltage level ranges available for VDDL. For further information over its function see 2.3.1 "Setting VDDL Level Range - J1" on page 19.

3.2.9 J2 - IDDL Measurement

J2 resides in IDDL's main current flow. To measure IDDL, J2 should be removed using a solder tool and a current meter should be connected instead with wires as short and thick as possible.

Warning

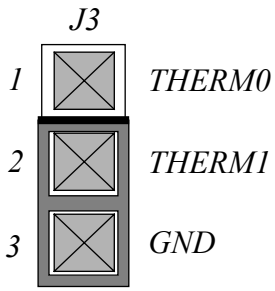
The job of removing J2 and soldering the current meter connections instead is very delicate and should be done by a skilled technician.

If this process is done by unskilled hands or repeated more than 3 times, permanent damage may occur to the MPC8260ADS.

3.2.10 J3 - Thermal Sense Connector

There are 2 dedicated pins THERM(0:1) which provide a way to take internal temperature measurements of the MPC8260. These pins should be connected to GND for normal operation. J3 is factory set with a jumper on its 2 - 3 positions, so that THERM1 is connected to GND.

Figure 3-2. J3 - Therm Connector



3.2.11 J4 - Optional Ventilator Supply

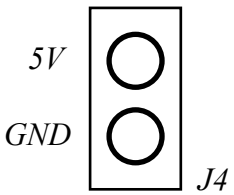
An optional cooling ventilator for the MPC8260 may be powered via J4, a 2 pin header connector (not assembled). In order to connect a ventilator to J4, either a 0.1" pitch header should be soldered to it, to be connected to a matching female connector or the ventilator supply wires may be soldered directly to J4, shown in Figure 3-3 ". J4 - Ventilator Supply" below:

Warning

The job of soldering wires or header to J4 is very delicate and should be done by a skilled technician.

If this process is done by unskilled hands or repeated more than 3 times, permanent damage may occur to the MPC8260ADS.

Figure 3-3. J4 - Ventilator Supply



3.2.12 J5 - COP/JTAG TDI Source Selection

J5 sets the structure of the COP/JTAG chain on the ADS. For further information over its function see 2.3.6 "MPC8260 JTAG's TDI Source Selection - J5" on page 22.

3.2.13 J6- IDDH Measurement

J6 resides in IDDH's main current flow. To measure IDDH, J6 should be removed using a solder tool, and a current meter should be connected, with as wires as short and thick as possible.

Warning

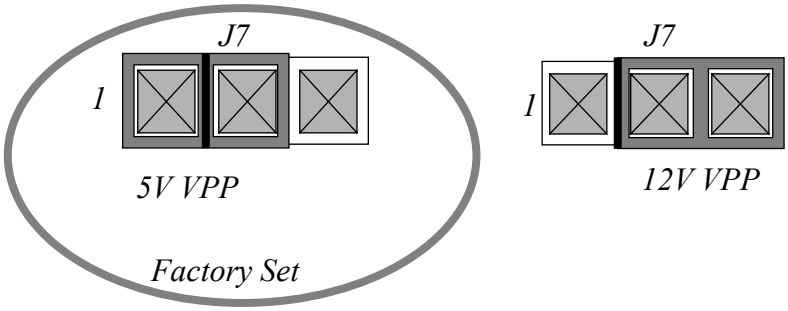
The job of removing J6 and soldering current meter connections instead is very delicate and should be done by a skilled technician.

If this process is done by unskilled hand or repeated more than 3 times, permanent damage might be inflicted to the MPC8260ADS.

3.2.14 J7 - VPP Source Selector

J7 selects the source for VPP - programming voltage for the Flash SIMM. When a jumper is located between pins **1 - 2** of J7 (Factory Set), the VPP is connected to the VCC plane of the ADS, providing **5V VPP**. When a jumper is located between positions ***2 - 3*** of J7, VPP is drawn from P2, that provides ***12V VPP*** if P2 is indeed connected to a 12V supply. J7 options are shown in Figure 3-4 ". J7 - VPP Source Selection" below:

Figure 3-4. J7 - VPP Source Selection



3.2.15 GND Bridges

There are 12 GND bridges on the MPC8260ADS, 11, designated as GND reside on digital ground and 1, designated as AGND resides on analog ground plane. Only 4 of the 11 digital GND bridges are actually assembled, the rest are SMD bridges and are optional. These bridges are meant to assist general measurements and logic-analyzer connection.

Warning

When connecting to a GND bridge, use only INSULATED GND clips. Otherwise, un-insulated clips may cause short- circuits, touching "HOT" points around them. Failure in doing so, might result in permanent damage to the MPC8260ADS.

3.2.16 ATM TX Indicator - LD1

The green ATM Receive LED indicator blinks whenever the PM5350 ATM-UNI is transmitting cells via the ATM port. Illuminates only when the ATM transceiver is enabled via BCSR1.

3.2.17 ATM RX Indicator - LD2

The green ATM Receive LED indicator blinks whenever the PM5350 ATM-UNI is receiving cells via the ATM port. Illuminates only when the ATM transceiver is enabled via BCSR1.

3.2.18 Ethernet RX Indicator - LD3

The green Ethernet Receive LED indicator blinks whenever the LXT970 is receiving data from the 10/100-Base-T port.

3.2.19 Ethernet TX Indicator - LD4

The green Ethernet Receive LED indicator blinks whenever the LXT970 is transmitting data via the 10/100-Base-T port.

3.2.20 Ethernet LINK Indicator - LD5

The yellow Ethernet Twisted Pair Link Integrity LED indicator - LINK, lights to indicate good link integrity on the 10/100-Base-T port. LD4 is off when the link integrity fails.

3.2.21 Fast Ethernet Indicator - LD6

When the LXT970 is enabled and is in 100 Mbps operation mode, the yellow led - LD6 lights.

3.2.22 Fast Ethernet CLSN Indicator - LD7

The red Ethernet Collision LED indicator CLSN, lights whenever a collision condition is detected on the 10/100-Base-T port, i.e., simultaneous receive and transmit. This led functions in this duty provided that bits 7:6) of LXT970's register 19, are cleared.

3.2.23 VDDL Indication - LD8

The green VDDL indicator led - LD8 is lit to indicate a VDDL power activity. Since VDDL level may vary, LD8's illumination level also varies accordingly.

3.2.24 3.3V Indicator - LD9

The green 3.3V led - LD9, indicates the presence of the +3.3V supply on the ADS.

3.2.25 RUN Indicator - LD10

When the green RUN led - LD10 is lit, it indicates that the MPC8260 is performing cycles on the PPC Bus. When dark, the MPC8260 is either running internally or stuck.

3.2.26 General Purpose Indicator 0 - LD11

This green indication led has no dedicated function over the ADS. It is meant to provide some visibility for program behavior. It is controlled by BCSR0.

3.2.27 General Purpose Indicator 1 - LD12

This red indication led has no dedicated function over the ADS. It is meant to provide additional visibility for program behavior. Its different color from LD11 provides additional information. It is controlled by BCSR0.

3.2.28 Fast Ethernet Port Initially Enabled - LD13

When the yellow ETH ON led is lit, it indicates that the fast ethernet port transceiver - the LXT970, is **initially** active. When it is dark, it indicates that the LXT970 is **initially** in power down mode, enabling the use of its associated FCC2 pins off-board via the expansion connectors. The state of LD13 is controlled by BCSR1.

This is a **soft**-indication, i.e., since the LXT970 may be controlled via the MII port, it is possible that the state of LD13 does not reflect correctly the status of the LXT970.

Note

Application S/W should always seek to match the state of LD13 to the status of the LXT970, so that, this indication is made reliable as to the correct status of the LXT970.

3.2.29 ATM ON - LD14

When the yellow ATM ON led is lit, it indicates that the ATM-UNI transceiver - the PM5350, is enabled for communication. When it is dark, the ATM-UNI transceiver is disconnected from the MPC8260, enabling the use of its associated FCC1 pins off-board via the expansion connectors.

ATM ON led is controlled by BCSR1.

3.2.30 RS232 Port 1 ON - LD15

When the yellow RS232 Port 1 ON led is lit, it designates, that the RS232 transceiver connected to PB3

(upper DB9 connector), is active and communication via that medium is allowed. When darkened, it designates that the transceiver is in shutdown mode and its associated SCC1 pins may be used off-board via the expansion connectors.

3.2.31 RS232 Port 2 ON - LD16

When the yellow RS232 Port 2 ON led is lit, it designates that the RS232 transceiver connected to PA3 (lower DB9 connector) is active and communication via that medium is allowed. When darkened, it designates, that the transceiver is in shutdown mode and its associated SCC2 pins may be used off-board via the expansion connectors.

3.3 Memory Map

All accesses to ADS's memory slaves are controlled by the MPC8260's memory controller. Therefore, the memory map is reprogrammable to the desire of the user. After Hard Reset is performed by the debug station, the debugger checks for the existence, size, delay and type of the SDRAM DIMM and Flash memory SIMM mounted on board and initializes the memory controller accordingly. The SDRAM and the Flash memory, respond to all types of memory access i.e., problem / supervisory, program / data and DMA. This memory map is a recommended memory map and since it is a "soft" map, devices' address may moved about the map, to the convenience of any user.

Table 3-1. ADS Memory Map

ADDESS RANGE	Memory Type	Device Name			Port Size
00000000 - 00FFFFFF	SDRAM DIMM	SDCUV6482 (16 MByte)	SDC8UV6484 (64 MByte)		64
01000000 - 03FFFFFF					
04000000 - 043FFFFF	SDRAM (Local Bus[1])	MB811171622A			32
04400000 - 044FFFFF	Empty Space				-

Table 3-1. ADS Memory Map

ADDESS RANGE	Memory Type	Device Name			Port Size
04500000 - 04507FFF	BCSR(0:7)[2]				32
04500000 - 04507FE3	BCSR0				
04500004 - 04507FE7	BCSR1				
04500008 - 04507FEB	BCSR2				
0450000C - 04507FEF	BCSR3				
04500010 - 04507FF3	BCSR4				
04500014 - 04507FF7	BCSR5				
04500018 - 04507FFB	BCSR6				
0450001C - 04507FFF	BCSR7				
04508000 - 045FFFFF	Empty Space				-
04600000 - 04607FFF[3]	ATM UNI Proc. Control	PMC5350 M/P I/F			8
04608000 - 046FFFFF	Empty Space				-
04700000[4] - 0470FFFF	MPC8260 Internal MAP[5]				32
04710000 - FCFFFFFF	Empty Space				-
FE000000[6] - FEFFFFFF	Flash SIMM			32M SIMM - SM73288	32
FF000000 - FF7FFFFF			16M SIMM - SM73248		
FF800000 - FFFFFFFF		8M SIMM - SM73228			

[1]The Local bus is fully transparent to the 60X bus, i.e., no mapping register. If a CS is assigned to the Local bus, its address space is completely visible to the 60X bus.

[2]The device appears repeatedly in multiples of its port-size (in bytes) X depth. E.g., BCSR0 appears at memory locations 4700000, 4700010, 4700020..., while BCSR1 appears at 4700004, 4700014, 4700024... and so on.

[3]The internal space of the ATM UNI control port is 256 bytes, however, the minimal block size that may be controlled by a CS region is 32KBytes.

[4]Initially at hF0000000 - hF000FFFF, set by hard reset configuration.

[5]Refer to the MPC8260 User's Manual for complete description of the MPC8260's internal memory map.
[6]Set by Hard-Reset configuration.

3.4 MPC8260 Register Programming

The MPC8260 provides the following functions on the MPC8260ADS:

1) System functions which include:
    - PPC Bus SDRAM Controller
    - Local Bus SDRAM Controller
    - Chip Select generator.
2) Communication functions which include:
    - ATM SAR
    - Fast Ethernet controller.
    - UART for terminal or host computer connection.

The internal registers of the MPC8260 must be programmed after Hard reset as described in the following paragraphs. The addresses and programming values are in **Hexadecimal** base.

For more information on the following initializations, see the MPC8260 User's Manual.

3.4.1 System Initialization

Table 3-2. Power-On Reset Configuration[1]

Flash Address [hex]	Init Value[hex]	Description
0	0C (1C[2])	Internal arbitration, Internal memory controller, Core enabled, Single MPC8260 (60X Bus mode[2]), 32 Bit boot port size, Exceptions vectored to 0xFFF00000, Internal space 64 bit slave for external master.
8	B2	L2 cache signals configured as BADDRx lines, DP(1:7) configured as L2 cache I/F and IRQ(6:7)~, Initial Internal space @ 0x0F000000
10	02	Boot memory space @ 0xFE000000 - 0xFFFFFFFF, $\overline{\text{ABB/IRQ2}}$ pin is $\overline{\text{ABB}}$, $\overline{\text{DBB/IRQ3}}$ pin is DBB, No masking on bus request lines, AP(1:3) configured as BNKSEL(0:2), APE~ configured as IRQ7~ and CS11~ as CS11~.
18	05	CS10~ configured as CS10~, PCI is configured to agent mode (not relevant for this application), MODCKH(0:3) == 5

[1]Programmed into the Flash memory in addresses 0x0, 0x8, 0x10 & 0x18 and to BCSR.
[2]With L2 cache

Table 3-3. SIU Register Programming

Register	Init Value[hex]	Description
RMR	0001	Check-Stop Reset enabled.
IMMR	04700000	Internal space @ 0x0470_0000
SYPCR	FFFFFFC3	Software watchdog timer count - FFFF, Bus-monitor timing FF, PPC Bus-monitor - Enabled, Local Bus-monitor - Enabled, S/W watch-dog - disabled, S/W watch-dog (if enabled) causes reset, S/W watch-dog (if enabled) - prescaled.
BCR	004C_0000 (8844_4000[1])	Single MPC8260 (60X Bus mode[1]), 0 wait-states on address tenure, No L2 cache (L2 cache assumed[1]), 0 clock hit delay[1] , 1-level Pipeline depth, Extended transfer mode Enabled (disabled[1]) for PPC bus, Extended Transfer mode Enabled for Local Bus, Odd parity for PPC & Local Buses (not relevant for this application), (Non MPC8260 master on EXT_BR2~[1]) (External Master delay enabled[1]), Internal space responds as 64 bit slave for external master (not relevant for this application).

[1]With L2 cache on board

3.4.2 Memory Controller Register Programming

The memory controller on the MPC8260ADS is initialized to 66 MHz operation. I.e., register programming

is based on 66 MHZ timing calculation.

Table 3-4. Memory Controller Initializations For 66Mhz

Reg.	Device Type	Bus	Init Value [hex]	Description
BR0	SM73228XG1JHBG0 by Smart Modular Tech.	PPC	FF801801	Base at 0FF80000, 32 bit port size, no parity, GPCM
	SM73248XG2JHBG0 by Smart Modular Tech.		FF001801	Base at 0FF00000, 32 bit port size, no parity, GPCM
	SM73288XG4JHBG0 by Smart Modular Tech.		FE001801	Base at 0FE00000, 32 bit port size, no parity, GPCM
OR0	SM73228XG1JHBG0 by Smart Modular Tech.		FF800836	8MByte block size, CS early negate, 6 w.s., Timing relax
	SM73248XG2JHBG0 by Smart Modular Tech.		FF000836	16MByte block size, CS early negate, 6 w.s., Timing relax
	SM73288XG4JHBG0		FE000836	32MByte block size, CS early negate, 6 w.s., Timing relax
BR1	BCSR	PPC	04501801	Base at 04500000, 32 bit port size, no parity, GPCM
OR1			FFFF8010	32 KByte block size, all types access, 1 w.s.
BR2	All SDRAM DIMM Supported		00000041	Base at 0, 64 bit port size, no parity, Sdram machine 1
OR2	SDC2UV6482C-84 by Fujitsu	PPC	FF000C80	16MByte block size, 2 banks per device, row starts at A9, 11 row lines, internal bank interleaving allowed, normal AACK operation
	SDC8UV6484C-84 by Fujitsu		FC002CC0	64MByte block size, 4 banks per device, row starts at A9, 12 row lines, internal bank interleaving allowed, normal AACK operation
BR3	Reserved	-	-	-
OR3				
BR4	MB811171622A-84	Local	04001861	Base at 04000000, 32 bit port size, no parity, Sdram machine 2
OR4			FFC00880	4 MByte block size,2 banks per device, row starts at A8, 10 row lines, internal bank interleaving allowed, normal AACK operation
BR5	PM5350 - ATM UNI	PPC	04600801	Base at 04600000, 8 bit port size, no parity, GPCM on PPC bus.
OR5			FFFF8E36	32K Byte block size, delayed CS assertion, early CS and WE negation for write cycle, relaxed timing, 7 w.s. for read, 8 for write, extended hold time after read.

Table 3-4. Memory Controller Initializations For 66Mhz

Reg.	Device Type	Bus	Init Value [hex]	Description
PSDMR	SDC2UV6482C-84 (16 MByte)	PPC Single MPC8260 Bus Mode	416EB452	**Bank** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 1, A(15 - 17) on BNKSEL(0:2), A9 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, no extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.
			C2AAB452	**Page** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 2, A19 on BNKSEL2[1], A8 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, no extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.
	SDC8UV6484C-84 (64 MByte)		412EB452	**Bank** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 1, A(13 - 15) on BNKSEL(0:2), A9 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, no extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.
			C372B452	**Page** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 2, A(16:17) on BNKSEL(1:2)[2], A6 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, no extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.

Table 3-4. Memory Controller Initializations For 66Mhz

Reg.	Device Type	Bus	Init Value [hex]	Description
PSDMR Contd.	SDC2UV6482C-84 (16 MByte) **DIMM_SIZE** in BCSR0 should be **Cleared**.	PPC 60X Bus Mode	416EB45A **PBI** in BCSR0 should be **Cleared**	**Bank** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 1, A(15 - 17) on BNKSEL(0:2), A9 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.
			C2AAB45A **PBI** in BCSR0 should be **Set**	**Page** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 2, A19 on BNKSEL2[1], A8 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.
	SDC8UV6484C-84 (64 MByte) **DIMM_SIZE** in BCSR0 should be **Set**.		412EB45A **PBI** in BCSR0 should be **Cleared**	**Bank** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 1, A(15 - 17) on BNKSEL(0:2), A9 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.
			C372B45A **PBI** in BCSR0 should be **Set**	**Page** Based Interleaving, Refresh enabled, normal operation code, address muxing mode 2, A(16:17) on BNKSEL(1:2)[2], A6 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 4 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, extra cycle on address phase, normal timing for control lines, 2 clocks CAS latency.
LSDMR	MB811171622A-84	Local	418AB552	Refresh enabled, normal operation, address muxing mode 1, A(16-18) on BNKSEL(0:2)[3], A10 on PSDA10, 7 clocks refresh recovery, 3 clocks precharge to activate delay, 2 clocks activate to read/write delay, 8 beat burst length, 1 clock last data out to precharge, 1 clock write recovery time, Internal address muxing, normal timing, 2 clocks CAS latency.
PSRT	All PPC Bus Sdram Supported	PPC	21	Divide MPTPR output by 34 (PSRT +1) Generates refresh every 13.4 µsec, while 16 µsec required. Therefore is refresh redundancy of 5.4 msec throughout full SDRAM refresh cycle which completes in 27.4 msec. I.e., Application s/w may withhold the bus upto app. 5.4 msec in a 32.8 msec period, without jeopardizing the contents of the ppc bus SDRAM DIMM.

Table 3-4. Memory Controller Initializations For 66Mhz

Reg.	Device Type	Bus	Init Value [hex]	Description
LSRT	MB811171622A-84	Local	21	Divide MPTPR output by 34 (LSRT +1) Generates refresh every 13.4 μsec, while 16 μsec required. Therefore is refresh redundancy of 5.4 msec throughout full SDRAM refresh cycle which completes in 27.4 msec. I.e., Application s/w may withhold the bus upto app. 5.4 msec in a 32.8 msec period, without jeopardizing the contents of the local bus SDRAM.
MPTPR	All SDRAMs on board		1900	Divide Bus clock by 26 (MPTPR+1) (decimal)

[1]Although this BSMA value corresponds to A(17:19) on BKSEL(0:2), when PBI is set, only the relevant BKSEL lines, according to number of internal SDARM banks, are VALID, in this case BKSEL2.

[2]Although this BSMA value corresponds to A(15:17) on BKSEL(0:2), when PBI is set, only the relevant BKSEL lines, according to number of internal SDARM banks, are VALID, in this case BKSEL(1:2)

[3]BNKSEL(0:2) are not connected for the Local Bus SDRAM. Use is done with Local bus address lines.

Chapter 4 Functional Description

In this chapter the various ADS's modules are described to their design details.

4.1 Reset & Reset - Configuration

There are several reset levels for the MPC8260, all of which are provided by ADS logic:

1) Power On Reset
2) Hard-Reset
3) Soft-Reset

4.1.1 Power - On Reset

The power on reset to the MPC8260 initializes the processor state after power up. There are 3 sources for power-on reset on the ADS:

1) A dedicated logic, using Seiko S-80728AN-DR-T1, which is a voltage detector of 2.8V +/- 2.4%, asserts PORESET* input to the MPC8260 for a period of ~2.5 sec. This time period is long enough to allow for VDDL stabilization time, powered by a different voltage regulator. It is assumed that the stabilization time for both linear regulators (see Figure 4-10 ". ADS Power Scheme" on page 75) is about the same.
2) Power On Reset may be generated manually as well, by a dedicated push-button SW1, when depressed, simulates a power-up state[1] for the above voltage detector
3) With this revision, Power On reset may also be generated by the JTAG logic, which is integrated with BCSR. See 4.13.2 "JTAG Generated Power-On Reset" on page 74.

4.1.1.1 Power - On Reset Configuration

At the end of the Power - On reset sequence, MODCK(1:3) and MODCKH(0:3) are sampled by the MPC8260 to configure the various clock modes of the MPC8260 (core, CPM, bus, etc). Selection among the MODCK(1:3) combination options is done by means of 3 dip-switches - DS1(6-8), see 2.3.3 "Setting MODCK(1:3) for PLLs Multiplication Factor - DS1 (#6 - #8)" on page 20, while MODCKH(0:3) are obtained from either the Flash memory or from DS1 (#2 - #5) via BCSR, depended on the state of DS1 (#1). See 2.3.5 "Setting MODCKH(0:3) - for PLLs Multiplication Factors" on page 22. This newly introduced feature, helps in cases where a user inadvertently erases the Flash memory, containing the Hard Reset configuration word, essential for system start-up.

The configuration master is determined upon the rising edge of PORST~, according to the state of RSTCONF~ signal, driven low on this board, to set the MPC8260 as a configuration master.

After power-on reset negates, the hard-reset sequence starts, during which, many other different options are configured (see 4.1.2.4 "Hard Rest Configuration" on page 43), among these options, are additional clock configuration bits - MODCKH(0:3) - the most significant bits of the MODCK field, which determine additional options for the clock generator. Although these bits are sampled whenever the hard-reset sequence is entered, they are **influential only once - after power-on reset**. If a hard reset sequence is entered later, MODCKH(0:3), although sampled, are don't care.

[1] I.e., the capacitor on its input is discharged.

4.1.2 Hard Reset

Hard-Reset may be generated on the ADS by the following sources:

1) COP/JTAG Port
2) Manual Hard reset.
3) MPC8260's internal sources.

Hard-Reset, when generated, causes the MPC8260 to reset all its internal hardware except for PLL logic, re-acquires the Hard-reset configuration from its current source, and jumps to the Reset vector in the exception table. Since hard-reset resets also the refresh logic for dynamic RAMs, their content is lost as well.

HRESET~ when asserted, is extended internally by the MPC8260 for additional 512 bus clock cycles at the end of which, the MPC8260 waits for 16 bus clock cycles and then, re-checks the state of the HRESET~ line.

HRESET~ is an open-drain signal and must be driven with an open-drain gate by which ever external source is driving it. Otherwise, contention will occur over that line, which might cause permanent damage to either ADS logic and/or to the MPC8260 itself.

4.1.2.1 COP/JTAG Port Hard - Reset

To provide convenient hard-reset capability for a COP/JTAG controller, HRESET~ line appears at the COP/JTAG port connector - P5. The COP/JTAG controller may directly generate hard-reset by asserting (low) this line.

4.1.2.2 Manual Hard - Reset

Manual hard reset allows run-time Hard-reset, when the COP controller is disconnected from the ADS and to support resident debuggers. Depressing both Soft-Reset and ABORT buttons(SW3 & SW2), asserts the HRESET* pin of the MPC8260, generating a HARD RESET sequence.

Since the HRESET* line may be driven internally by the MPC8260, it is driven to the MPC8260 with an open-drain gate. If off-board H/W connected to the ADS is to drive HRESET~ line, then, it should do so with an open-drain gate, this, to avoid contention over this line.

To save on board area, a dedicated button is not provided, but is shared with the Soft-Reset button and the ABORT button - when both depressed, Hard Reset is generated.

4.1.2.3 Internal Sources Hard - Reset

The MPC8260 has internal sources which generate Hard Reset. Among these sources are:

1) Loss of Lock Reset. When one of the PLLs (Core, CPM), is out of lock, hard-reset is generated.
2) Check-Stop Reset. When the core enters a Check-Stop state from some reason, hard-reset may be generated, depended on CSRE bit in the RMR.
3) Bus Monitor Reset. When the bus monitor is enabled and a bus cycle is not terminated, hard-reset is generated.
4) S/W Watch Dog Reset. When the S/W watch-dog is enabled, and application s/w fails to perform its reset routine, it will generate hard - reset.
5) COP/JTAG Reset (Internal). Hard reset may be forced by driving the HRESET~ line via the external pin's scan chain. Not useful for run time.

In general, the MPC8260 asserts a reset line HARD or SOFT for a period 512 clock cycles after a reset source has been identified. A hard reset sequence is followed by a soft reset sequence.

4.1.2.4 Hard Rest Configuration

When Hard-Reset is applied to the MPC8260 (externally as well as internally), it samples the Hard-Reset configuration. This configuration is taken from the current Hard-Reset configuration source (only over the

MS 8 bits of the data bus, D0-D7) whenever HRESET~ is asserted. The only exception to this are the MODCKH(0:3) bits, which are actually sampled only once - after power-on reset.

With this revision (PILOT) of the ADS, Hard Reset configuration word may be sampled from two sources, selected by DS1-1:

1) When DS1-1 is in the ON position, the configuration word is taken from the Flash memory, located at its base address.

2) When DS1-1 is in the OFF position, the configuration word is taken from BCSR. Unlike with the Flash memory, this word may not be reprogrammed by the user, as it is programmed into programmable logic. This option allows convenient recovery when the Flash memory is inadvertently erased, with Hard Reset Configuration word lost.

For additional information see 2.3.3 "Setting MODCK(1:3) for PLLs Multiplication Factor - DS1 (#6 - #8)" on page 20, 2.3.4 "Setting Hard - Reset Configuration Source" on page 21 and 2.3.5 "Setting MODCKH(0:3) - for PLLs Multiplication Factors" on page 22.

During hard reset sequence, the MPC8260 reads the Hard Reset configuration source at addresses 0, 0x8, 0x18, 0x20,... a byte each time, to assemble the 32 bit configuration word. A total of 32 bytes of data is read from D(0:7) to acquire 8 full configuration words for system that may have up to 8 MPC8260 chips.

The configuration word for a single[1] MPC8260 is stored in the Flash memory SIMM and in BCSR, while the other 7 words are not initialized, as there are no additional MPC8260 chips on this ADS.

Table 4-1. Hard Reset Configuration Word

Field	Data Bus Bits	Prog Value [Bin]	Implication	Offset In Flash [Hex]	Value [Hex]
ERB	0	’0’	Internal Arbitration Selected.	0	0C / 1C
EXMC	1	’0’	Internal Memory Controller. CS0~ active at system boot.		
CDIS	2	’0’	Core Enabled.		
EBM	3	’0’ / ’1’	‘0’ - sets Single MPC8260 Mode for regular ADS boards. ‘1’ - sets 60X Bus Mode[1] for boards with L2 Cache assembled. (L2 or L2C suffix)		
BPS	4:5	’11’	Sets 32 Bit Boot Port Size.		
CIP	6	‘0’	Sets Core Initial Prefix MSR[IP] = 1, so that the system exception table is placed at address 0xFFF00100, regardless of Flash memory size.		
ISPS	7	’0’	64 bit internal space for external master access. In fact don’t care on this board, as neither an external master is present with regular ADS boards nor the internal space of the MPC8260 is non-cached, with L2 cache boards.		

[1]Although the MPC8260 as configuration master reads 8 configuration words, only the first configuration word is influential.

Table 4-1. Hard Reset Configuration Word

Field	Data Bus Bits	Prog Value [Bin]	Implication	Offset In Flash [Hex]	Value [Hex]
L2CPC	8:9	'10[2]'	CI~/BADDR29/IRQ2~ selected as BADDR29 WT~/BADDR30/IRQ3~ selected as BADDR30 L2_HIT~/IRQ4~ unassigned CPU_BG~/BADDR31/IRQ5 selected as BADDR31	8	B2
DPPC	10:11	'11'	Data Parity Pin Configuration: DP0 set as EXT_BR2~, DP1 as EXT_BG2~, DP2 as EXT_DBG2~, DP3 as EXT_BR3~, DP4 as EXT_BG3~, DP5 as EXT_DBG3~, DP6 is IRQ6~, DP7 as IRQ7~.		
-	12	'0'	Reserved, should be cleared.		
ISB	13:15	'010'	IMMR initial value 0x0F000000, i.e., the internal space resides initially at this address.		
BMS	16	'0'	Boot memory (Flash) at 0xFE000000.	10	02
BBD	17	'0'	ABB~/IRQ2~ pin is set to ABB~ DBB~/IRQ3~ pin is set to DBB~		
MMR	18:19	'00'	Mask Masters Request - No masking of Bus Request lines[3].		
LBPC	20:21	'00'	Local Bus pins configured as Local bus pins.		
APPC	22:23	'10'	MODCK1/AP(1)/TC(0) set as BKSEL0 MODCK2/AP(2)/TC(1) set as BKSEL1 MODCK3/AP(3)/TC(2) set as BKSEL2 IRQ7~/APE~ set as IRQ7~[4] CS11~/AP(0) set as CS11~		
CS10PC	24:25	'00'	CS10~/BCTL1/DBG_DIS~ set as CS10~	18	05
-	26:27	'00'	Reserved. Should be cleared.		
MODCK_HI[5]	28:31	'0101[6]'	This fields sets the MODCKH(0:3) field, which is the 4 MSB for the 7 bit MODCK field. It is programmed into Flash and may be taken from there when DS1/1 is at ON position, or it is set by DS1/(2 - 5) when DS1/1 is at OFF position, allows total flexibility in selecting desired MODCK. When MODCKH(0:3) are set to this value (5) and MODCK(1:3) == '101', both CPM and Core MFs are set to 2. When MODCK(1:3) == '111', CPM's MF is set to 2 while Core's MF is set to 3.		

[1]For L2 Cache Boards.
[2]Was '00' with ENG version of this board.
[3]In fact not significant with this application, since the only possible master - the L2 Cache wakes-up disabled.
[4]DP7 is also set as IRQ7~. They are logic OR'ed into the interrupt controller.
[5]Applies only ONCE after Power-On reset.
[6]Programmed into Flash and Factory Set for DS1/(2-5).

4.1.3 Soft Reset

Soft - Reset may be generated on the ADS from the below sources:

1) COP/JTAG Port
2) Manual Soft Reset
3) Internal MPC8260 source.

Soft-Reset, when generated, causes the MPC8260 to reset its internal logic, while keeping its hard-reset configuration and memory controller setup and then jumping to the Reset vector in the exception table. Since soft-reset, does not reset the refresh logic for dynamic RAMs, their contents is preserved.

SRESET~ when asserted, is extended internally by the MPC8260 for an additional 512 bus clock cycles at the end of which, the MPC8260 waits for 16 bus clock cycles and then, re-checks the state of the SRESET~ line.

SRESET~ is an open-drain signal and must be driven with an open-drain gate by every external source driving it. Otherwise, contention will occur over that line, which might cause permanent damage to either the ADS logic and / or to the MPC8260 itself.

4.1.3.1 COP/JTAG Port Soft - Reset

To provide convenient soft-reset capability for a COP/JTAG controller, SRESET~ line appears at the COP/JTAG port connector - P5. The COP/JTAG controller may directly generate Soft-reset by asserting (low) this line.

4.1.3.2 Manual Soft - Reset

Manual soft reset allows run-time soft-reset when the COP controller is disconnected from the ADS and for resident debuggers' support. Depressing the Soft-Reset button (SW3), asserts the SRESET* pin of the MPC8260, generating a Soft Reset sequence.

Since the SRESET* line may be driven internally by the MPC8260, it is driven to the MPC8260 with an open-drain gate. If off-board hardware is connected to the ADS is to drive SRESET~ line, then, it should do so with an open-drain gate to avoid contention over this line, which might inflict permanent damage to either the ADS logic and / or to the MPC8260 itself.

4.1.3.3 Internal Sources Soft - Reset

The only internal Soft-reset source is the COP/JTAG soft-reset, which may be generated using Public JTAG instructions to shift active-value ('0') to the SRESET~ pin via the boundary scan chain. This is not useful for run time.

4.2 Local Interrupter

There are 2 external interrupt which are applied by ADS logic to the MPC8260 via its interrupt controller:

1) ABORT (NMI)
2) ATM UNI interrupt
3) Fast Ethernet PHY Interrupt

4.2.1 ABORT Interrupt

The ABORT (NMI), is generated by a push-button. When this button is depressed, the IRQ0~ input to the MPC8260 is asserted. The purpose of this type of interrupt, is to support the use of resident debuggers if any is made available to the ADS. This interrupt is enabled by setting the MSR[EE] bit.

To support external (off-board) generation of an NMI, the IRQ0* line is driven by an open-drain gate. This allows for external h/w to also drive this line. If external h/w indeed does drive IRQ0*, it is compulsory that

IRQ0* is driven by an open-drain gate.

4.2.2 ATM UNI Interrupt

To support ATM UNI (User Network I/F) event report by means of interrupt, the interrupt output of the UNI (INTB) is connected to IRQ6~ line of the MPC8260.

Since INTB of the UNI is an open-drain output, it is possible to connect additional (off-board) interrupt requesters on the same IRQ6~, provided that they drive IRQ6~ with an open-drain gate as well.

4.2.3 Fast Ethernet Transceiver Interrupt

To support Fast Ethernet Transceiver event report by means of interrupt, the (FDS/MDINT) interrupt output of the LXT970A is connected to IRQ7~ line of the MPC8260.

Since FDS/MDINT of the LXT970A is an open-drain output, it is possible to connect additional (off-board) interrupt requesters on the same IRQ7~, provided that they drive IRQ7~ with an open-drain gate as well.

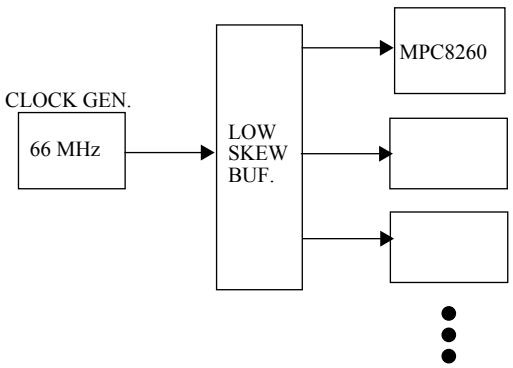
The FDS/MDINT is a dual functionality signal - on its FDS (Full-Duplex Status) function, it indicates whether the LXT970 is configured to Full/Half Duplex mode, while in its alternate function it serves as the transceivers MDINT active-low output. In order to achieve this functionality, bit 1 in register 17 (17.1) must be SET after the ADS comes out of Hard Reset. Setting this bit is allowed through the MDIO port of the LXT970, Data of which is driven / sampled by PC9 of the MPC8260 while its Clock is driven by PC10. Failure in doing so, will result in IRQ7~ pin of the MPC8260, constantly asserted (low).

4.3 Clock Generator

The MPC8260 requires a single clock source for the main clock oscillator. All MPC8260 bus timings are referenced to the main clock input - CLKIN (unlike the 8XX family, timings of which, are referenced to CLKOUT signal). The main clock input is in 1:1 ratio to the bus clock with an internal skew elimination (PLL). This uses a 66MHz 3.3V clock generator, which is connected to a low-interskew buffer to split the load between all various clock consumers on board.

Special care is taken to isolate and terminate the clock routes between the clock-distributor and on-board consumers, this to provide "clean" clock inputs for proper operation.

Figure 4-1. Clock Generator Scheme



4.4 Bus Configuration

The MPC8260 on the ADS, may be configured in 2 possible bus modes, depended upon the presence of L2 Cache on board:

1) Single MPC8260 Mode

2) 60X Bus Mode.

4.4.1 Single MPC8260 Mode

When a L2 Cache is not present on the ADS, the MPC8260 is configured in Single MPC8260 Mode. I.e., assuming only one master on the bus, with no support for external master access. This allows for internal address multiplexing to occur, with the external address multiplexers made redundant and therefore not assembled, improving SDRAM performance.

4.4.2 60x Bus Mode

When L2 Cache is installed on the ADS, the MPC8260 may no longer operate in single MPC8260 mode, this since the address must be seen as is by the cache calling for the introduction of the external address latch - multiplexers, required for the PowerPC bus SDRAM. In this mode, SDRAM performance is harmed due to added wait-state on the first access in a page, caused by the delay associated with the external multiplexers.

4.5 Buffering

In order to achieve best performance, it is necessary to reduce the capacitive load over the 60X bus as much as possible. Therefore, the slower devices on the bus, i.e., the Flash Simm, the BCSR and the ATM UNI M/P i/f are buffered, removing their capacitive load from the PPC bus, while the SDRAM DIMM and the cache are not buffered from the 60X bus.

Latches are provided over address and strobe[1] lines while transceivers are provided for data. Use is done with 74ALVT buffers (by Philips) which are 3.3V operated, 5V tolerant[2] and provide Bus-Hold to reduce pull up / down resistor count. This type of buffer reduces noise on board due to reduced transitions' amplitude.

To further reduce noise and reflections, damping resistors are placed over SDRAM DIMM's address and strobe lines, over all MPC8260's strobe lines and over Local Bus SDRAM's address lines.

The data transceivers open only if there is an access to a valid[3] buffered board address or during Hard - Reset configuration[4]. That way data conflicts are avoided between unbuffered memory reads and the data-buffers.

The MPC8260's local bus is not buffered at all, this since there is only one slave on that bus - i.e., the SDRAM.

4.6 Chip - Select Generator

The memory controller of the MPC8260 is used as a chip-select generator to access on-board[5] memories, saving board's area reducing cost, power consumption and increasing flexibility.

The MPC8260's chip-selects assignment to the various memories / registers on the ADS are as shown in Table 4-2 ". ADS Chip Select Assignments" below:

Table 4-2. ADS Chip Select Assignments

Chip Select:	Assignment	Bus	Timing Machine
CS0~	Flash Memory SIMM	60X (Buffered)	GPCM

[1]If necessary.
[2]Required for Flash and BCSR
[3]An address which covered in a Chip-Select region, that controls a buffered device.
[4]To allow a configuration word stored in Flash memory become active.
[5]And off-board. See further.

Table 4-2. ADS Chip Select Assignments

Chip Select:	Assignment	Bus	Timing Machine
CS1~	BCSR	60X (Buffered)	GPCM
CS2~	SDRAM DIMM (Single Bank only)	60X (Main)	SDRAM Machine 1
CS3~	Reserved for future use.	-	-
CS4~	SDRAM (Soldered on board)	Local	SDRAM Machine 2
CS5~	ATM UNI Microprocessor I/F	60X (Main)	UPMB
CS6~	Comm. Tool M/P I/F Cs 1	60X (Buffered)	GPCM / UPMx
CS7~	Comm. Tool M/P I/F Cs 2	60X (Buffered)	GPCM / UPMx
CS(8 -11)~	Unused	-	-

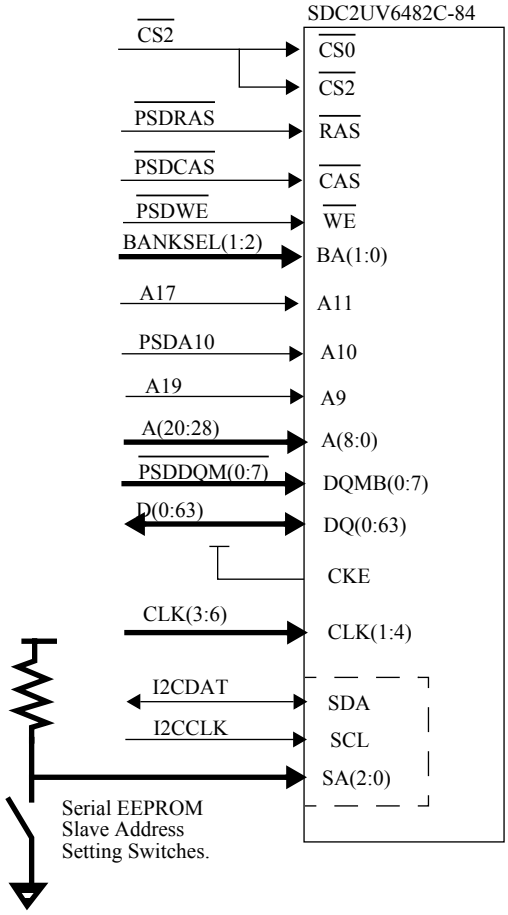
4.7 Synchronous DRAM DIMM (60X Bus)

To enhance performance, especially in higher operation frequencies - 16 MBytes of SDRAM DIMM are provided on board. The SDRAM DIMM's data bus is unbuffered from the MPC8260 60X bus and is configured as 2 X 1M X 64. The SDC2UV6482C-84T-S is an unbuffered 168 pin DIMM by Fujitsu or compatible, which is composed of eight 2 X 1M X 8 SDRAM chips (MB81117822A- 84). The DIMM's data sheet may be obtained on the Internet at URL: ***http://www.fujitsumicro.com/products/memory/sdram_mod.html***, while the SDRAM chips' (from which the DIMM is composed) data sheet may be obtained at URL: ***http://www.fujitsumicro.com/products/memory/sdrams.html***.

The SDRAM's timing is controlled by SDRAM Machine #1, (associated with 60X bus,) via its assigned Chip Select line (See Table 4-2 ". ADS Chip Select Assignments" on page 48).

The SDRAM connection scheme is shown in Figure 4-2 ". SDRAM DIMM Connection Scheme" below.

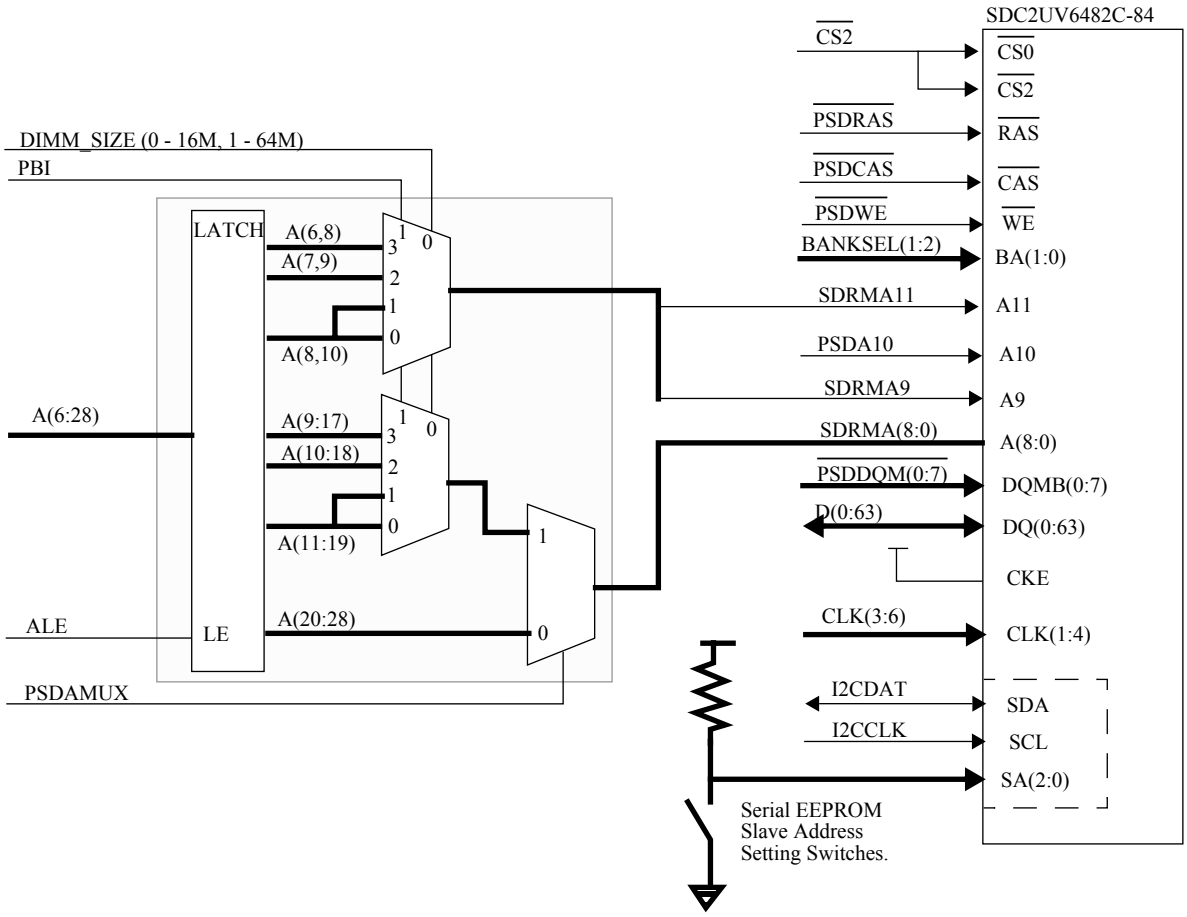
Figure 4-2. SDRAM DIMM Connection Scheme



As can be seen from the above figure, with this revision of the ADS, there are few changes with respect to previous revision of the ADS:

1) MPC8260‘s BKSEL(1:2) are connected to DIMM’s BA(1:0) respectively instead of BK-SEL(0:1)
2) MPC8260’s A17 is connected to DIMM’s A11 instead of BKSEL2
3) BKSEL0 is unused
4) MPC8260’s A19 is connected to DIMM’s A9 instead of MPC8260’s A10.

Figure 4-3. SDRAM DIMM - 60X Bus Connection Scheme



As can be seen from the above, with respect to the previous revision of the ADS, the address latch-mux was expanded and is conditioned with PBI and DIMM_SIZE indications, from BCSR0.

The SDRAM performance, is shown in Table 4-3 ". SDRAM DIMM (84MHz) Performance Figures" on page 51

Table 4-3. SDRAM DIMM (84MHz) Performance Figures

Cycle Type	System Clock Cycles @ 66MHz Bus Clock Freq.	System Clock Cycles @ 66MHz Bus Clock Freq. With L2 cache
Burst Read - Page Miss	6[1],1,1,1	7[1],1,1,1
Burst Read - Page Hit	4[1],1,1,1	5[1],1,1,1
Burst Write - Page Miss	4[1],1,1,1	5[1],1,1,1
Burst Write - Page Hit	2[1],1,1,1	4[1],1,1,1
Refresh	8[2]	8[2]

[1]From TS~ Asserted. First access may be longer due to internal pipeline delay

[2]Not including arbitration overhead.

4.7.1 SDRAM Programming

After power-up, the SDRAM needs to be initialized by means of programming, to establish its mode of operation. The SDRAM is programmed according to the following procedure:

1) Issue Precharge-All command
2) Issue 8 CBR refresh commands
3) Issue MODE-SET command.

When a Mode Register Set command is issued, data is passed to the Mode Register through the SDRAM's address lines. This command is fully supported by the SDRAM machine of the MPC8260.

Mode Register programming values are shown in Table 4-4 ". 66 MHz SDRAM DIMM Mode Register Programming" below:

Table 4-4. 66 MHz SDRAM DIMM Mode Register Programming

SDRAM Address Line[1]	SDRAM Mode Reg Field	Value	Meaning:
A11 (MSB)	Reserved	'0'	
A10	Reserved	'0'	
A9	Opcode	'0' / '1'	0 - Burst Read & Burst Write (Copy-Back data cache) 1 - Burst Read & Single Write (Write-Through Data cache)
A8	Reserved	'0'	
A7	Reserved	'0'	
A6 - A4	CAS Latency	'010'	Data Valid 2 Clocks cycles after CAS Asserted
A3	Burst Type	'0'	Sequential Burst
A2 - A0	Burst Length	'011'	8 Operand Burst Length

[1]Actually SDRAM's A0 is connected to MPC8260's A28 and so on...

The SDRAM machine one of the MPC8260 needs to be initialized as well, this after BCSR2 is read to find out whether a L2 cache is present on board.

The programming of the SDRAM machine 1, is shown in Table 3-4 ". Memory Controller Initializations For 66Mhz" on page 38.

4.7.2 SDRAM Refresh

The SDRAM is refreshed using its auto-refresh mode. I.e., using SDRAM machine one's periodic timer, an auto-refresh command is issued to the SDRAM every 13.4 µsec, so that all 2048[1] SDRAM DIMM rows are refreshed within specified 32.8 msec, while leaving an interval of ~5.4 msec of refresh redundancy within that window, as a safety measure, to cover for possible delays in bus availability for the refresh controller.

[1]In fact each SDRAM component is composed of 2 internal banks each having 2048 rows, but they are refreshed in parallel.

4.7.3 L2-Cache Support Influence On SDRAM Design

To support an optional L2-Cache on the ADS, the following measures are taken:

1) Optional Latch-Multiplexers are added over selected address lines. See Figure 4-2 ". SDRAM DIMM Connection Scheme" on page 50. These latch-mux are normally bypassed by zero ohm resistors, that are not assembled for L2 Cache boards.

2) The MPC8260 supports additional wait-state on PSDMUX line, so that the row-address may be allowed to propagate via the Latch - Multiplexers in time for the Activate command.

3) With this revision of the ADS, support for Sdram PBI (Page Based Interleaving) was added. Therefore, the relative location of the Row-Address field, is shifted up the address lines, depended on the number of internal banks within a SDRAM DIMM. This since the Bank Select line(s) are inserted between the Column (LSB) and Row (MSB) address lines. As can be seen from Figure 4-3 ". SDRAM DIMM - 60X Bus Connection Scheme" on page 51, PBI and DIMM_SIZE signals, driven via BCSR0, select the correct address line group for a specific DIMM size, with PBI set.

Note:

Since there is no indication in H/W for PSDMR[PBI] state and as for DIMM's size[1], these parameters are set to the ADS logic via BCSR. It is the system's programmer responsibility to set these parameters correctly in BCSR, reflecting the current state of PSDMR[PBI] and the DIMM's size. Failure in doing so, will result with improper operation of the SDRAM DIMM, on ADS boards with L2 Cache assembled.

The performance of the SDRAM is harmed by the addition of the external multiplexers of the SDRAM's address lines. The effects may be seen in Table 4-3 ". SDRAM DIMM (84MHz) Performance Figures" on page 51.

4.7.4 SDRAM DIMM Configuration Information

Unlike memory SIMMs which have few presence detect lines for configuration report, the DIMM's configuration information is stored in a 256 Byte Serial EEPROM residing on the DIMM, compatible with I^2C protocol. In fact, all necessary information is in the first half of the EEPROM, while the second half is system available. On the ADS, the DIMM configuration EEPROM is connected to the MPC8260 I^2C controller, to inquire for SDRAM DIMM's configuration, after hard-reset sequence.

An example of such a serial eeprom access protocol may be seen in AT24C02 (by Atmel) data sheet. This document may be obtained on the Internet at URL: ***http://www.atmel.com/atmel/products/prod162.htm***.

As can be seen from the Atmel document, 3 bits of the device's 7 bit slave address b'1010$A_2A_1A_0$R/W' are compared against the SA(2:0) signals on the DIMM. On the ADS these lines are controlled by a DIP-Switch - DS2, so that the DIMM's I^2C slave address may be changed in favor of other devices that may reside on the I^2C bus.

4.8 Flash Memory SIMM

The ADS is provided with 8 Mbytes of 95-nsec Flash memory SIMM, the SM73228XG1JHBG0, by Smart Modular Technology, composed of four LH28F016SCT-L95 chips by Sharp, arranged as 2M X 32 in single bank. Support is given to 16 MBytes and 32 MBytes SIMMs as well. The Flash SIMM resides on an 80 pin

[1]The DIMM size, is recorded within the configuration Serial EEPROM, residing on the DIMM, but obviously may not be used to qualify logic.

SIMM socket.

To minimize use of MPC8260's chip-select lines, only one chip-select line (CS0~) is used to select the Flash as a whole, while distributing chip-select lines among the module's internal banks is done by on-board programmable logic, according to the Presence-Detect lines of the Flash SIMM plugged into the ADS.

The access time of the Flash memory provided with the ADS is 95 nsec, however, devices with other delays may be supported as well. Reading the delay section of the Flash SIMM Presence-Detect lines (see Table 4-11 ". BCSR2 Description" on page 64), the debugger may establish (via OR0) the correct number of wait-states (considering 66MHz Bus clock frequency) required for accessing the Flash SIMM.

The control over the Flash is done using the GPCM and a dedicated CS0~ region, controlling the whole bank. During hard - reset initializations[1], the debugger or any application S/W for that matter, reads the Flash Presence-Detect lines via BCSR and decides how to program BR0 & OR0 registers, within which the size and the delay of the region are determined.

The performance of the Flash memory is shown in Table 4-5 ". Flash Memory Performance Figures" below:

Table 4-5. Flash Memory Performance Figures

	Number of Bus Clock Cycles @ 66 MHz Bus Clock Freq.
Cycle Type \ Flash Delay [nsec]	95
Read Access	8[1]
Write[2] Access	9[1]

[1]From TS~ asserted. However, due to internal activity, these figures may be larger.

[2]The figures in the table refer to the actual write access. The write operation continues internally and the device has to be polled for completion.

4.8.1 Flash Protection Logic

The Flash SIMM provided with the ADS, is divided into 32 blocks of 256KBytes. Each block has an individual Lock-Bit associated with it, allowing individual block erase / write protection.

The Block Lock bits are protected by a Master Lock Bit, creating a second level of protection.

A block lock bit may be set individually, but may be cleared only as a group, i.e., clearing ALL block lock bits. Once a block lock bit is set, erasure / write of that block is possible only when VPP to the Flash SIMM is 12V[2].

When the Master Lock bit is set, setting / clearing of Block Lock bits is possible only when VPP to the Flash SIMM is 12V[2], otherwise VPP of 5V is sufficient.

A Master Lock bit set operation, requires VPP of 12V[2]. Once the Master Lock Bit is Set, it may not be cleared!

4.8.2 Flash Programming Voltage

Support is given to 5V programmable modules as well as 12V programmable modules, programming voltage to which, is applied either internally - 5V or externally via the 12V power input of the ADS - P2, which may be left unconnected in case a 5V programmable module is being used or in case there is no need

[1]I.e., initializations that follow the hard reset sequence at system boot.

[2]I.e., J7 is set between 2 - 3, 12V supply is connected to P2.

for Flash programming. The selection between VPP's voltage levels is done via J7. (See 3.2.14 "J7 - VPP Source Selector" on page 31.)

To avoid inadvertent programming or erasure of the Flash it is recommended to set J7 between 2 - 3 position (while P2 is disconnected), so that no VPP is applied to the Flash SIMM.

4.8.3 Flash and L2Cache

If the L2 cache is installed, the MPC8260 needs to be programmed to 60x bus mode. This requires the latches for the buffered address bus to the Flash[1] to be enabled. On this revision of the ADS, the 3 lowest order address lines for the Flash, are provided by the BADDR(27-29) lines of the MPC8260. However, BADRR29 function of the MPC8260 is multiplexed with CI~ (Cache Inhibit) function over the same pin. Therefore, prior to enabling the L2Cache, any code residing in the Flash, should be moved into the PowerPC bus SDRAM[2], prior to changing BADDR29 function to CI~ via SIUMCR.

4.8.4 Hard - Reset Configuration - MPC8260 Revision Dependency

As described in 4.1 "Reset & Reset - Configuration" on page 42, the Flash may provide Hard - Reset configuration, programmed into it when DS1/1 is set to the corresponding position (See 2.3.4 "Setting Hard - Reset Configuration Source" on page 21). As mentioned above, BADDR(27-29) lines are used with this revision of the ADS, to address the Flash. However, there is an erratum associated with revision 0 of the MPC8260, by which, the BADDR lines do not function during Hard - Reset configuration sequence. As a result, with rev 0 MPC8260 chips the only possible way for the ADS to acquire Hard-Reset configuration, is by taking it from BCSR.

NOTE:

Due to the above, if revision 0 MPC8260 silicon is used, with revision PILOT of this board - DS1/1 MUST be set to OFF position for the ADS to come up correctly.

4.9 Local Bus Synchronous DRAM

To enhance ATM performance, 4 MBytes of SDRAM are provided on the Local Bus, as connection table storage. The SDRAM is unbuffered from the MPC8260's local bus and is configured as 2 X 512K X 32. It is implemented with two MB811171622A- 84 chips by Fujitsu or compatible.

The local SDRAM's timing is controlled by the second SDRAM machine of the MPC8260, which is dedicated for the Local Bus and is assigned to a CS line according to Table 4-2 ". ADS Chip Select Assignments" on page 48.

The local bus SDRAM connection scheme is shown in Figure 4-4 ". Local SDRAM Connection Scheme" on page 56.

The local bus SDRAM performance, is shown in Table 4-6 ". Local Bus SDRAM Performance Figures - 66MHz" on page 55:

Table 4-6. Local Bus SDRAM Performance Figures - 66MHz

Cycle Type	Bus Clock Cycles @ 66 MHz Bus Clock Freq.
Burst Read - Page Miss	6[1],1,1,1

[1]As well as all other slow static devices.
[2]It is required to do so anyway, since the L2Cache must operate within a full 64-bit data bus environment.

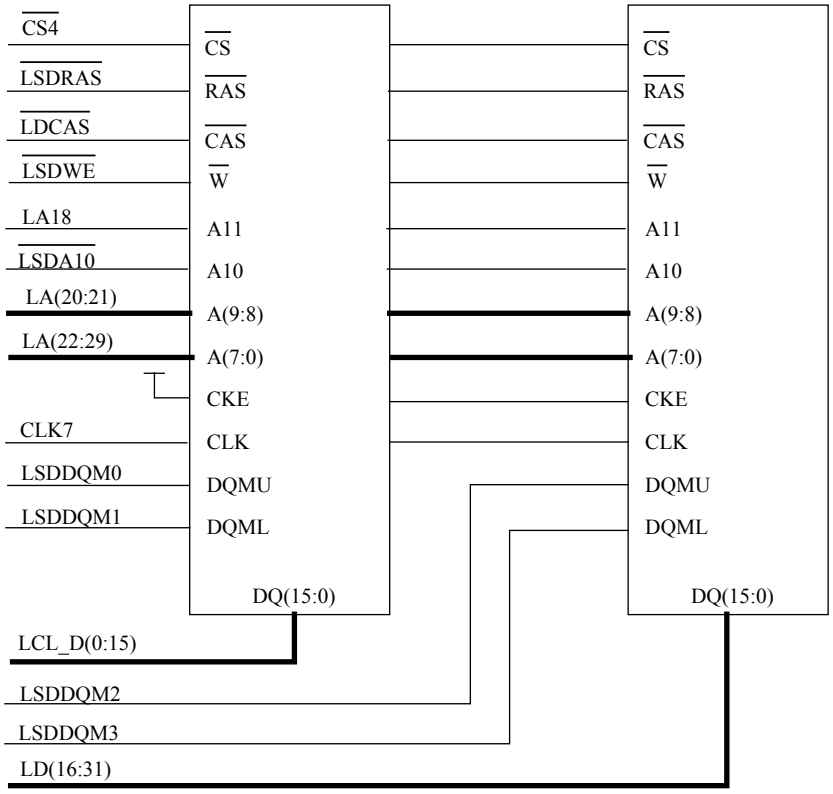
Table 4-6. Local Bus SDRAM Performance Figures - 66MHz

Cycle Type	Bus Clock Cycles @ 66 MHz Bus Clock Freq.
Burst Read - Page Hit	4[1],1,1,1
Burst Write - Page Miss	4[1],1,1,1
Burst Write - Page Hit	2[1],1,1,1
Refresh	8[2]

[1]From TS~ Asserted. First access may be longer due to internal pipeline delay

[2]Not including arbitration overhead.

Figure 4-4. Local SDRAM Connection Scheme



4.9.1 Local Bus SDRAM Programming

After power-up, the local bus SDRAM needs to be initialized by means of programming, to establish its mode of operation. The Local bus Sdram is programmed according to the following procedure:

1) Issue Precharge-All command
2) Issue 8 CBR refresh commands

3) Issue MODE-SET command.

When a Mode Register Set command is issued, data is passed to the Mode Register through the SDRAM's address lines. This command is fully supported by the SDRAM machines of the MPC8260.

Mode Register programming values are shown in Table 4-4 ". 66 MHz SDRAM DIMM Mode Register Programming" below:

Table 4-7. 66 MHz Local Bus SDRAM Mode Register Programming

SDRAM Address Line[1]	SDRAM Mode Reg Field	Value	Meaning:
A11 (MSB)	Reserved	'0'	
A10	Reserved	'0'	
A9	Opcode	'0'	Burst Read & Burst Write (copy back)
A8	Reserved	'0'	
A7	Reserved	'0'	
A6 - A4	CAS Latency	'010'	Data Valid 2 Clocks cycles after CAS Asserted
A3	Burst Type	'0'	Sequential Burst
A2 - A0 (lsb)	Burst Length	'011'	8 Operand Burst Length

[1]Actually SDRAM's A0 is connected to MPC8260's LCL_A29 and so on...

4.9.2 Local Bus SDRAM Refresh

The Local Bus SDRAM is refreshed using its auto-refresh mode. I.e., using SDRAM machine two's periodic timer, an auto-refresh command is issued to the SDRAM every 13.4 µsec, so that all 2048[1] SDRAM DIMM rows are refreshed within specified 32.8 msec, while leaving a 5.4 msec interval of refresh redundancy within that window, as a safety measure, covering for possible delays in bus availability for the refresh controller.

4.10 L2-Cache (Optional)

To enhance benchmarking, optional support is provided for L2-Cache. This is implemented with two MPC2605 devices, each containing 256 KBytes of look-aside[2] cache along with its control, providing a total of 512 KBytes of L2 cache.

The cache is connected directly over the 60X bus and is supported gluelessly by the MPC8260. The cache data sheet may be obtained via the internet at URL: ***http://mot-sps.com/books/dl156/pdf/mpc2605rev5.pdf***.

The presence of the L2-Cache, calls for the introduction of latch - multiplexers over SDRAM's address lines, this, since the MPC2605 snooping logic needs to monitor the address as is (linear rather than multiplexed) and the bus works by the 60X bus protocol, allowing address pipelining[3]. These latch - multiplexers are soldered in place only in case a cache is installed on-board. Otherwise they are omitted and bypassed by zero ohm resistors. See also 4.7.3 "L2-Cache Support Influence On SDRAM Design" on page 53.

---

[1]In fact each SDRAM component is composed of 2 internal banks each having 2048 rows, but they are refreshed in parallel.
[2]I.e., residing on the same bus as the processor.
[3]Only single level is allowed with the MPC8260.

4.10.1 L2 Cache Configuration & Control

The cache is configured via 5 configuration lines - CFG(0:4) for the following functions:

1) Cache size is set by CFG(0:2). The various settings of these lines per each cache module are encoded in the table below:

Table 4-8. CFG(0:2) Settings

L2 Cache Size [Byte]	CFG(0:2)
256K	’000’ (Reserved)
512K	’010’ -First Module (A26 == 0)
	’011’ - Second Module (A26 == 1)

2) Snoop is Enabled - CFG3 driven low for both modules.

3) AACK~ assertion enabled - CFG4 driven high for both modules.

The caches’ HRESET~ lines are connected directly to the SRESET~ line of the MPC8260, so that whenever Soft-reset is asserted to or by the MPC8260, the cache is reset along with it, loosing all data previously stored in it. The cache has 5 control lines that control its operation and state:

- PWRDWN~ which is constantly set to high (no power down support on the ADS)
- L2FLUSH~, assertion of which[1] flushes out the cache array. This signal is controlled by BCSR0.
- L2MISS_INH~ in fact Cache-Lock. When Asserted the cache does not change it contents. Controlled by BCSR0.
- L2TAG_CLR~. Clears all tag memory. Controlled by BCSR0.
- L2UPDATE_INH~. In fact cache freeze (without information loss). Controlled by BCSR0. See Table 4-9 ". BCSR0 Description" on page 62.

All the above signals are connected directly to both cache modules.

4.11 Communication Ports

The ADS includes several communication ports, to allow convenient CPM evaluation. Obviously, it is not possible to provide all types of communication interfaces supported by the CPM, but it provides a convenient connection to communication interface devices to the MPC8260 via the CPM expansion connectors, residing on the edge of the board.

The communication ports’ interfaces provided on the ADS are listed below:

1) 155 Mbps ATM UNI on FCC1 with Optical I/f, connected via UTOPIA I/F.

2) 10/100-Base-T Port on FCC2, MII controlled.

3) Dual RS232 port residing on SCC1 & SCC2.

4.11.1 ATM Port

To support the MPC8260’s ATM controller, a 155.52Mbps User Network Interface (UNI) is provided on board, connected to FCC1 of the MPC8260 via the UTOPIA I/F. This is implemented with a PM5350 S/UNI-155-ULTRA by PMC-SIERA. Although these transceivers are capable of supporting 51.84Mbps rate,

[1]For minimum 8 Bus clock cycles.

support is given only to the higher rate.

The control over the transceiver is done using the microprocessor i/f of the transceiver, controlled by the MPC8260 memory controller's GPCM. Since the UNI is 5V powered and the MPC8260 3.3V powered (5V intolerant), the UNI is buffered (LCX buffers) from the MPC8260 on both the receive part of UTOPIA I/F and the microprocessor control ports.

The ATM transceiver may be enabled / disabled at any time by writing '0' / '1' to the ATMEN~ bit in BCSR1. When ATMEN~ is negated, ('1') the microprocessor control port remains accessible on the MPC8260 memory map while its associated FCC is detached and may be used off-board via the expansion connectors.

The UNI interrupt output is connected to the MPC8260's DP6/CSE0/IRQ6~ pin. This allows for interrupt-based handshake between the MPC8260 and the ATM UNI, in addition to a polling based handshake. This is an open-drain output and is pulled-up on the ADS. It is also appears at the CPM expansion connector, to be shared with an external tool interrupt request. See also 4.2 "Local Interrupter" on page 46.

The ATM transceiver reset input is driven by HRESET~ signal of the MPC8260, so that the UNI is reset whenever a hard-reset sequence occurs. The UNI may also be reset by either asserting ATM_RST bit in BCSR1 (see Table 4-10 ". BCSR1 Description" on page 64) or by asserting ('1') the RESET bit in the Master Reset and Identify / Load Meters register via the UNI m/p i/f.

The UNI transmit and receive clocks is fed with a 19.44 MHz +/- 20 ppm, clock generator, 5 V powered, while the receive and transmit fifos' clocks are provided by the MPC8260 using the same clock (CLK11).

The ATM transceiver has Transmit and Receive visual indications. These, however, are enabled by setting the following bits in the UNI POPC register (offset 0x68 from UNI base):

- TRAFIC to 1
- ALARM to 0
- TOGGLE[1:] to b11.

This will generate a 100 msec pulse over OUT1 and OUT0 pins of the UNI (attached to LD2 and LD1 respectively) indicating a successful ATM cell receive or transmit event, respectively.

The ATM SAR is connected to the physical medium by an optical I/F. This is implemented with HP's HFBR 5205 optical I/F, which operates at 1300 nm with upto 2 Km transmission range.

4.11.2 10/100 Base-T Port

A fast Ethernet port with T.P. (100-Base-TX) I/F is provided on the ADS. This port also supports 10 Mbps ethernet (10-Base-T) via the same transceiver - the LXT970 by Level One.

The LXT970 is connected to FCC2 of the MPC8260 via MII interface[1], which is used for both the device's control and data path. The initial configuration of the LXT970 is done by setting desired values at 8 configuration signals: FDE, CFG(0:1) and MF(0:4). The MF(0:4) pins however, are controlled by four voltage levels, this to allow each pin to configure two functions. On the ADS these pins are driven by factory set zero ohm resistors, connected to a voltage divider, allowing for a future option change during production.

The LXT970 is initially disabled, according to the state of FETHIEN in BCSR1. See Table 4-10 ". BCSR1 Description" on page 64.

The LXT970 reset input is driven by HRESET~ signal of the MPC8260, resetting the transceiver whenever hard-reset sequence is taken. The LXT970 may also be reset by either asserting the FETH_RST bit in BCSR1 (see Table 4-10 ". BCSR1 Description" on page 64) or by asserting bit 0.15 (MSB of LXT970 control register) via the MII I/F.

To allow external use of FCC2, its pins appear at the CPM expansion connectors and the ethernet transceiver

[1]Media Independent Interface.

may be Disabled / Enabled at any time via the MII's MDIO port.

The LXT970 is able to interrupt the MPC8260 via the IRQ7~ line. This line is shared also with the CPM expansion connectors. Therefore, any tool that is connected to IRQ7~ or IRQ6~ for that matter, should drive these lines with an Open Drain buffer. Both IRQ6~ and IRQ7~ are pulled-up on the ADS.

4.11.2.1 LXT970 Control

The LXT970 is controlled via the MII management[1] port which is a 2 wire interface: a clock (MDC) and a bidirectional data line (MDIO). This is in fact a bus, i.e., up to 32 devices may reside over it, while the protocol defines a 5-bit slave address field, which is compared against the slave address set to each device by hardware during device reset, according to the levels on MF(4:0) pins. On the ADS the slave address is hard-set to b00000. The MPC8260 on the ADS interfaces this port using two PI/O pins: PC9 for MDIO and PC10 for MDC. There is no special support within the MPC8260 for the MDIO port and the protocol is implemented in S/W.

The MDIO port may interrupt a host in 2 ways: (a[2]) driving low the MDIO line during IDLE time or (b) using a dedicated interrupt line FDS/MDINT~ which may also serve as Full-Duplex indication. On the ADS, this line is connected to the MPC8260's DP7/CSE1/IRQ7~ line, appearing also at the CPM expansion connectors. After the LX970 is reset, the FDS/MDINT~ pin, wakes-up as FDS rather than MDINT~ and therefore, **MUST be initially programmed to MDINT~ function, by setting 17.1 bit**, otherwise, IRQ7~ may be constantly driven low, possibly generating interrupts to the MPC8260, if not masked properly.

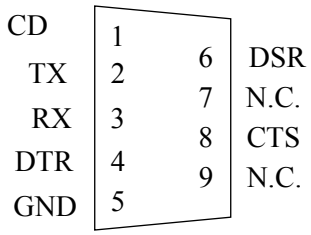
Since IRQ7~ may also be driven by any tool, connected to the expansion connectors, it should be driven with an Open Drain buffer. IRQ7~ is pulled-up on the ADS.

4.11.3 RS232 Ports

To assist user's applications and to provide convenient communication channels with both a terminal and a host computer, two identical RS232 ports are provided on the ADS, connected to SCC1 and SCC2 ports of the MPC8260. This is implemented by an MC145583 transceiver which generates RS232 levels internally using a single 3.3V supply and has a standby mode. When the RS232EN1 or RS232EN2 bits in BCSR1 are asserted (low), the corresponding transceiver is enabled. When negated, the corresponding transceiver enters standby mode, within which, the receiver outputs are tri-stated, enabling use of the corresponding port's pins, off-board via the expansion connectors.

The RS232 ports are implemented with two, 9-pin, female D-Type stacked connectors PA3 (the lower) which is connected to SCC2 and PB3 (the upper) which connected to SCC1. Both connectors, are configured to be directly (via a flat cable) connected to a standard IBM-PC like RS232 connector.

Figure 4-5. RS-232 Serial Port Connectors - PA3 and PB3



4.11.3.1 RS-232 Ports Signal Description

In the list below, the directions 'I', 'O', and 'I/O' are relative to the ADS board. (I.e. 'I' means input to the

[1]Also known as MII MDIO port.
[2]Not supported on the ADS.

ADS)

- CD (O) - Data Carrier Detect. This line is always asserted by the ADS.
- TX (O) - Transmit Data. On Port 1 (Upper connector), connected to SCC1's TxD on PD30. On Port 2 (Lower connector) connected to SCC2's TxD on PD27.
- RX (I) - Receive Data. On Port 1 (Upper connector), connected to SCC1's RxD on PD31. On Port 2 (Lower connector) connected to SCC2's RxD on PD28.
- DTR (I) - Data Terminal Ready. This signal is used by the software on the ADS to detect if a terminal is connected to the ADS board. On Port 1 (Upper connector), connected to SCC1's CD~ on PC14. On Port 2 (Lower connector) connected to SCC2's CD~ on PC12.
- DSR[1] (O) - Data Set Ready. This line is always asserted by the ADS.
- RTS (I) - Request To Send. This line is not connected on the ADS.
- CTS (O) - Clear To Send. On Port 1 (Upper connector), connected to SCC1's RTS~ on PD29. On Port 2 (Lower connector) connected to SCC2's RTS~ on P26.

4.12 Board Control & Status Register - BCSR

Most of the hardware options on the ADS are controlled or monitored by the BCSR, which is a 32 bit wide read / write register file. BCSR resides over the PPC Bus, accessed via the MPC8260's memory controller (see Table 4-2 ". ADS Chip Select Assignments" on page 48) and in fact includes 8 registers: BCSR0 to BCSR7. Since the minimum block size for a CS region is 32KBytes and only A(27:29) lines are decoded by the BCSR for register selection, BCSR0 - BCSR7 are duplicated many times inside that region. See also Table 1-1 ". MPC8260ADS Specifications" on page 12.

The following functions are controlled / monitored by the BCSR:

1) PBI (60X Bus mode only)
2) L2 Cache Inhibit
3) L2 Cache Flush
4) L2 Cache Lock
5) L2 Cache tag Clear.
6) ATM Port Control which includes:
    - Transceiver Enable / Disable
    - Transceiver Reset.
7) Fast Ethernet Port Control which includes:
    - Transceiver Initial Enable
    - Transceiver Reset
8) RS232 port 1 Enable / Disable.
9) RS232 port 2 Enable / Disable.
10) Flash Size / Delay Identification.
11) External (off-board) tools Support which include:
    - Tool Identification
    - Tool Revision
    - Tool Status Information
12) S/W Option Identification.

[1]Since there are only 3 RS232 transmitters in the device, DSR is connected to CD.

13) ADS Revision code.

14) Fast Download via JTAG (Optional)

15) Power On Reset via JTAG (Optional)

Since part of the ADS's modules are controlled by the BCSR and since they may be disabled in favor of external hardware, the enable signals for these modules are presented at the CPM expansion connectors, so that off- board hardware may be mutually exclusively enabled with on-board modules.

4.12.1 BCSR0 - Board Control - Status Register 0

The BCSR0 serves as a control register on the ADS. Although it resides only over D(0:7) lines of the PPC data bus, it is accessed as a **word** at **offset 0** from BCSR base address. It may be read or written at any time. BCSR0 gets its defaults upon Power-On reset. BCSR0 fields are described in Table 4-9 ". BCSR0 Description" below:

Table 4-9. BCSR0 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
0	PBI	***Page Base Interleaving***. In 60X mode (i.e., with L2-Cache), this bit should reflect (**system programmer responsibility**) the state of PBI bit in PSDMR. When active (**High**) it changes the address Muxing scheme for the SDRAM, so that to match a scheme where Bank Select signals are connected below lower row address lines. When Inactive, the addressing scheme is such that Bank Select lines are taken from the higher order address lines (above row address). This signal operates in conjunction to DIMM_SIZE signal below. **In Single MPC8260 Mode (i.e., without L2-Cache), this bit has no effect**.	0	R/W
1	DIMM_SIZE	***Sdram DIMM Size***. In 60X mode (i.e., with L2-Cache), this bit, in conjunction with PBI above, controls the address muxing scheme for the Sdram DIMM. When **Low**, the addressing scheme matches a **16 MByte** DIMM while when ***High***, the addressing scheme matches a ***64 MBytes*** DIMM. **In Single MPC8260 Mode (i.e., without L2-Cache), this bit has no effect**.	0	R/W
2	L2C_INH	***L2 Cache Inhibit***. When this bit is active (**low**), the L2 cache is inhibited and unable to respond to cacheable cycles. However, bus activity is still monitored by the cache so that it may respond immediately after this signal is negated. This signal is connected to the MPC2605's L2 UPDATE INH. This signal has no function in a ADS that does not have an L2 Cache installed.	0	R,W
3	L2C_FLUSH	***L2 Cache Flush***. When this bit is active (**low**) for min. 8 bus cycles, the MPC2605 initiates a process within which, valid lines are marked invalid, while dirty lines are written back to memory and marked invalid. This signal is connected to the L2 FLUSH signal of the MPC2605. This signal has no function in a ADS that does not have an L2 Cache installed.	1	R,W
4	L2C_LOCK	***L2 Cache Lock***. When this bit is active (**low**), the MPC2605 will stop entering new data into the cache, while yet maintaining existing data and responding to cacheable cycles. This signal has no function in a ADS that does not have an L2 Cache installed.	1	R,W
5	L2C_CLEAR	***L2 Cache Clear***. When this bit is active (**Low**) for min. 8 bus clock cycles, the L2 cache invalidates all its entries, without flushing, the same process as with HRESET~ asserted. However, it still monitors the bus, so it can immediately respond when this process ends. This signal is connected to the L2 TAG CLR of the MPC2605, but has no function when a cache is not installed on the ADS.	1	R,W

Table 4-9. BCSR0 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
6	SIGNAL_LAMP_0	***Signal Lamp 0***. When this signal is active (**low**), a dedicated Green LED (LD11) illuminates. When in-active, this led is darkened. This led may be used for S/W signalling to user.	1	R,W
7	SIGNAL_LAMP_1	***Signal Lamp 1***. When this signal is active (**low**), a dedicated Red LED (LD12) illuminates. When in-active, this led is darkened. This led may be used for S/W signalling to user.	1	R,W
8 - 31	Reserved	Un-Implemented.	-	-

4.12.2 BCSR1 - Board Control / Status Register 1

The BCSR1 serves as a control register on the ADS. Although it resides only over D(2:7) lines of the PPC data bus, it is accessed as a **word** at **offset 4** from BCSR base address. It may be read or written at any time. BCSR1 gets its defaults upon Power-On reset. BCSR1 fields are described in Table 4-10 ". BCSR1 Description" below

Table 4-10. BCSR1 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
0 - 1	Reserved	Un-implemented	-	-
2	ATM_EN	***ATM Port Enable***. When asserted (**low**) the ATM UNI chip (PM5350) connected to FCC1 is enabled for transmission and reception. When negated, the ATM transceiver is in fact[1] in standby mode and its associated buffers[2] are in tri-state mode, freeing all its i/f signals for off-board use via the expansion connectors.	1	R,W
3	ATM_RST	ATM Port Reset. When asserted (**low**), the ATM port transceiver is in reset state. This line is driven also by HRESET~ signal of the MPC8260.	1	R,W
4	FETHIEN	***Fast Ethernet Port Initial Enable***. When asserted (**low**) the LXT970's MII port, residing on FCC2, is enabled after Power-Up or after FETH_RST is negated. When negated (**high**), the LXT970's MII port is isolated after Power-Up or after FETH_RST is negated and all i/f signals are tri-stated. After initial value has been set this signal has no influence over the LXT970 and MII isolation may be controlled via MDIO 0.10 bit.	1	R,W
5	FETH_RST	Fast Ethernet port Reset. When active (**low**) the LXT970is reset. This line is also driven by HRESET~ signal of the MPC8260. Since MDDIS pin of the LXT970 is driven low with this application, the negation of this signal causes all the H/W configuration bits to be sampled for initial values and device control is moved to the MDIO channel, which is the control path of the MII port.	1	R,W
6	RS232EN_1	***RS232 port 1 Enable***. When asserted (**low**) the RS232 transceiver for port 1 (upper), is enabled. When negated, the RS232 transceiver for port 1, is in standby mode and SCC1 pins are available for off-board use via the expansion connectors.	1	R,W
7	RS232EN_2	***RS232 port 2 Enable***. When asserted (**low**) the RS232 transceiver for port 2 (lower), is enabled. When negated, the RS232 transceiver for port 2, is in standby mode and SCC2 pins are available for off-board use via the expansion connectors.	1	R,W
8 - 31	Reserved	Un-implemented	-	-

[1]The ATM transceiver itself does not enter standby mode, the fact that it is disconnected from the MPC8260 emulates this state.
[2]Required for voltage levels adaptation.

4.12.3 BCSR2 - Board Control / Status Register - 2

BCSR2 is a status register which is accessed as **word** at **offset 8** from the BCSR base address. Its a **Read-Only** register which may be read at any time. BCSR2's various fields are described in Table 4-11 ". BCSR2 Description" on page 64.

Table 4-11. BCSR2 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
0 - 7	TSTAT(0:7)	***Tool Status (0:7)***. This field is reserved for external tool status report. The exact meaning of each bit within this field is tool unique and therefore will be documented separately per each tool. These signals are available at the System expansion connector - P16.	-	R

Table 4-11. BCSR2 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
8 - 11	TOOLREV(0:3)	***TOOL Revision (0:3)***. This field may contains the revision code of an external tool connected to the ADS. The various combinations of this field will be described per each tool user's manual. These signals are available at the System expansion connector - P16.		R
12 - 15	EXTTOLI(0:3)	***External Tools Identification***. These lines, which are available at the CPM expansion connectors, are intended to serve as tools' identifier. On-board S/W may check these lines to detect The presence of various tools (h/w expansions) at the CPM expansion connectors. For the external tools' codes and their associated combinations see Table 4-14 ". EXTOOLI(0:3) Assignment" on page 66.	-	R
16 - 17	SWOPT(0:1)[1]	Software Option (0:1). This field shows the state of a dedicated dip-switches (DS3/ 1-2) providing an option to manually change a program flow.	0	R
18 - 19	L2CSIZE(0:1)	***L2 Cache Size (0:1)***. This field encodes the size of the L2 Cache, present on the ADS. For the encoding of the various cache sizes see Table 4-17 ". L2 Cache Size Encoding" on page 67.	-	R
20 - 23	BREVN(0:3)	***Board Revision Number (0:3)***. This field represents the revision code, hard-assigned to the ADS. See Table 4-16 ". ADS Revision Encoding" on page 67, for revisions' encoding.	-	R
24	SWOPT2	Software Option 2. This is the LSB of the field. Shows the state of a dedicated dip-switch (DS3/3) providing an option to manually change a program flow. For the setting of DS3 see 3.2.7 "DS3 - Software Options Switch" on page 30.	0	R
25 - 27	FLASH_PD(7:5)	***Flash Presence Detect(7:5)***. These lines are connected to the Flash SIMM presence detect lines, which encode the Delay of Flash SIMM mounted on the Flash SIMM socket. For the encoding of FLASH_PD(4:1) see Table 4-12 ". Flash Presence Detect (7:5) Encoding" on page 66.	-	R
28 - 31	FLASH_PD(4:1)	***Flash Presence Detect(4:1)***. These lines are connected to the Flash SIMM presence detect lines which encode the type of Flash SIMM mounted on the Flash SIMM socket. For the encoding of FLASH_PD(4:1) see Table 4-13 ". Flash Presence Detect (4:1) Encoding" on page 66.	-	R

[1]There is additional bit to this field. See bit 24 in the same table.

Table 4-12. Flash Presence Detect (7:5) Encoding

FLASH_PD(7:5)	FLASH DELAY [nsec]
000	Not Supported
001	150
010	100/120
011	80/90
100	70 nsec
101 - 111	Not Supported

Table 4-13. Flash Presence Detect (4:1) Encoding

FLASH_PD(4:1)	Flash TYPE / SIZE
0000	SM73288XG4JHBG0 - 32 MByte (4 banks of 4 X 2M X 8) by Smart Modular Technology.
0001	SM73248XG2JHBG0 - 16 MByte (2 banks of 4 X 2M X 8) by Smart Modular Technology.
0010	SM73228XG1JHBG0 - 8 MByte (1 bank of 4 X 2M X 8) by Smart Modular Technology.
0011 - 1111	Not Supported

Table 4-14. EXTOOLI(0:3) Assignment

EXTTOOLI(0:3) [hex]	External Tool
0	T/ECOM - MPC8260 Communication tool
1	Reserved
2	T1 Circuit Emulation Tool
3 - E	Reserved
F	Tool Non Existent

Table 4-15. External Tool Revision Encoding

TOOLREV(0:3) [hex]	External Tool Revision
0	ENGINEERING
1	PILOT
2	A[1]
3 - F	Reserved

[1]Future revision

Table 4-16. ADS Revision Encoding

Revision Number (0:3) [Hex]	ADS Revision
0	ENG (Engineering)
1	PILOT
2	A[1]
3 - F	Reserved

[1]Future revision

Table 4-17. L2 Cache Size Encoding

L2CSIZE(0:1)	L2 Cache Size
’00’	Reserved
’01’	512 KBytes
’10’	Reserved
’11’	No L2 Cache

4.12.4 BCSR3 to BCSR5 - Board Control / Status Register 3 - 5

BCSR3 to BCSR5 are additional control / status registers which may be accessed as a **word** at **offset 0xC to 0x14** from BCSR base address. These registers are not implemented. They may be read or written but with no valid data nor any effect on the ADS. The description of BCSR3 to BCSR5 is shown in Table 4-18 ". BCSR3 to BCSR5 Description" below.

Table 4-18. BCSR3 to BCSR5 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
0 - 31	Reserved	Un Implemented	-	-

4.12.5 BCSR6 - Board Control / Status Register 6

BCSR 6 is used for the JTAG Fast Download I/F Status & Control. Although it resides only over D(0:7) lines of the PPC data bus, it is accessed as a **word** at **offset 0x18** from BCSR base. For additional information on that I/F see 4.13.1 "Fast Download Support" on page 69. The description of BCSR6 is shown in Table 4-19 ". BCSR6 Description" below:

Table 4-19. BCSR6 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
0	JTAG_EN	***JTAG Enable***. When this bit is active (**High**) the JTAG machine for fast download is enabled for use. When inactive, TDI is asynchronously driven to MTDO. In this mode (Power-On default) COP controller S/W may operate on the ADS in ENG revision compatible mode.	0	R/W
1 - 6	Reserved	Implemented. Read as b'000000'. Writes have no effect.	0	R
7	JTAG_RX_FULL	***JTAG Receive Full Flag***. When this signal is active (**High**), it indicates, that the JTAG Download register, was fully written by the Host and should be read by the download agent running on the ADS. After the agent has read data from the Jtag Download Data Register, this bit is cleared. This bit is also cleared by either Power-On Reset, JTAG TAP Reset asserted (TRST~) and by JTAG TAP reset state. This bit is Read-Only, writing it has no effect.	0	R
8 - 31	Reserved	Un-Implemented.	-	-

4.12.6 BCSR7 - Board Control / Status Register 7

BCSR7 is used as the JTAG Fast Download I/F data register. Although it resides only over D(0:7) lines of the PPC data bus, it is accessed as **word** at **offset 0x1C** from BCSR base. During download, the host loads this register with serial data through the JTAG I/F. The download agent, running on the ADS, should, after polling the JTAG_RX_FULL flag to be asserted, read the data on this register and write it to the ADS's memory.

BCSR7 is described in Table 4-20 ". BCSR7 Description" below:

Table 4-20. BCSR7 Description

BIT	MNEMONIC	Function	PON DEF	ATT.
0 - 7	JTAG_DOWNLOAD_DATA	JTAG Data. Data shifted into the JTAG Download Data register, may be read here. This is a read-only field. Writes have no effect.	0	R
8 - 31	Reserved	Un-Implemented.	-	-

4.13 COP/JTAG Port

The COP - Control Observation Port, is part of the MPC8260's JTAG machine, implemented as a set of additional instructions and logic within the JTAG permissions. This port may be connected to a dedicated debug station[1], for extensive system debug.

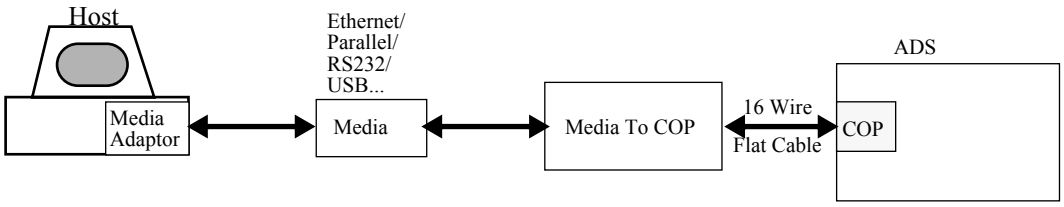
There are several third party debug solutions on the market. These debug-stations may be connected to the

[1]Not provided with the ADS.

host computer via either Ethernet, Parallel-Port, RS232 or any other media.

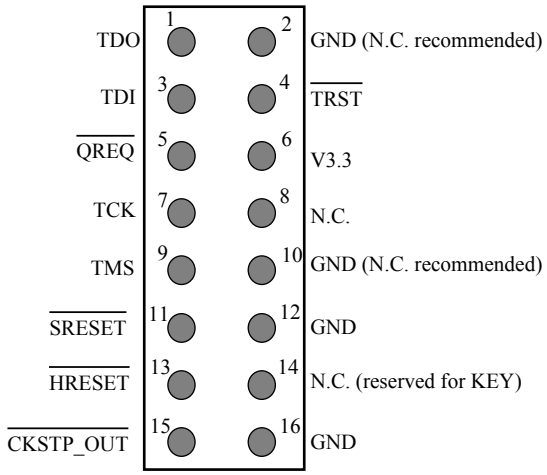
The debug station connection scheme is shown in Figure 4-6 ". Debug Station Connection Schemes" below:

Figure 4-6. Debug Station Connection Schemes



To support debug station connection to the COP/JTAG port, a 16 pin generic header connector (P5) is provided on the ADS, carrying the COP/JTAG signals as well as additional signals aiding in system debug. The pinout of this connector, is a general Motorola recommendation for including a COP/JTAG port in a design. The pinout of the COP/JTAG connector is shown in Figure 4-7 ". COP/JTAG Port Connector" on page 69:

Figure 4-7. COP/JTAG Port Connector[1]



For the detailed description of the COP / JTAG connector signals, see Table 5-5 ". P5 - COP / JTAG Connector - Interconnect Signals" on page 86.

4.13.1 Fast Download Support

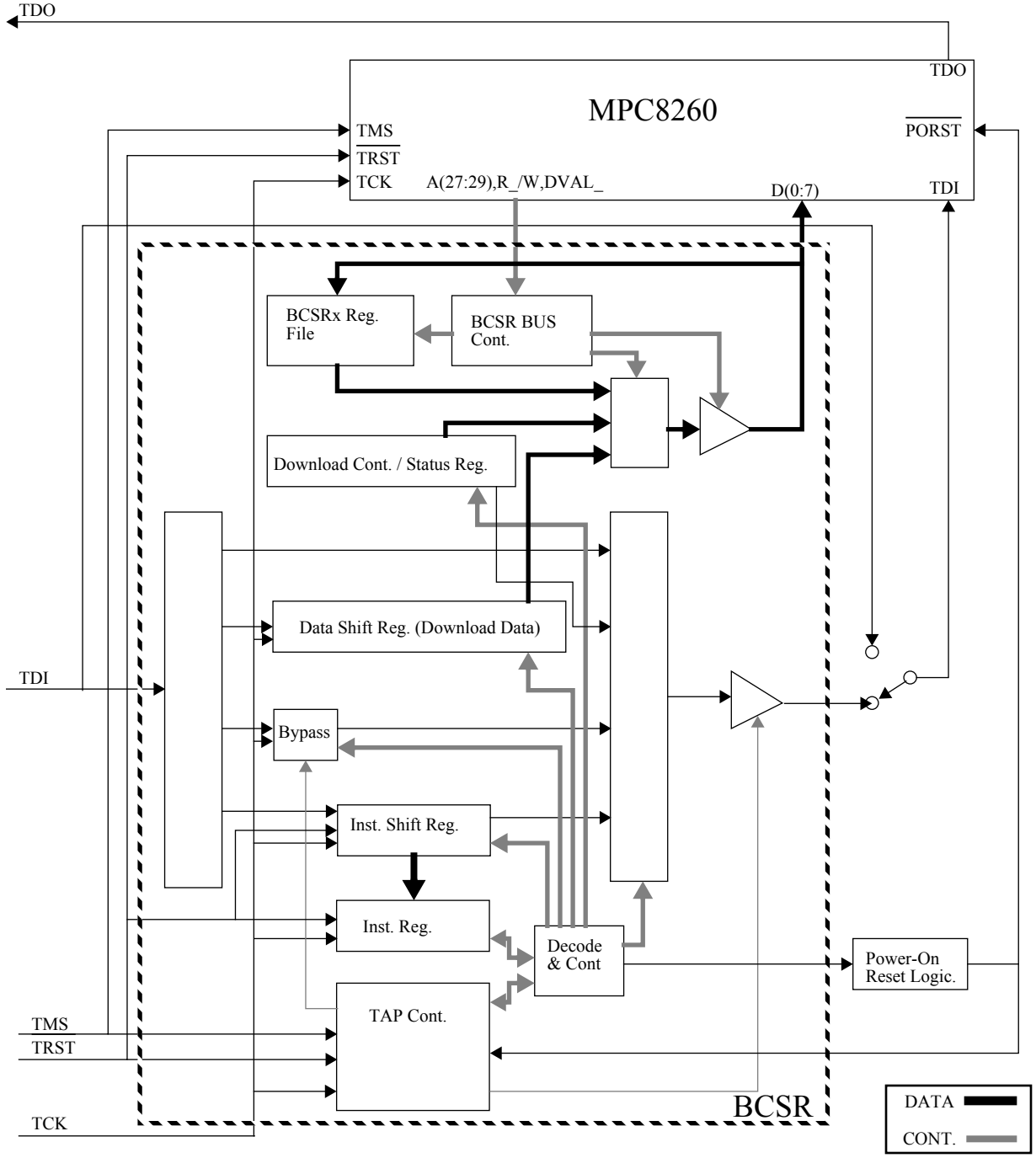
Download rates through the COP port are inherently slow due to very long COP scan chains and the extraneous amount of data transferred. On revision PILOT of the ADS, support was added for fast download through the JTAG port.

In essence, an additional (to MPC8260's) JTAG machine was added in front of the MPC8260's JTAG port. This machine supports the minimal (public) set of JTAG instructions, required by JTAG rules to be a part of a JTAG chain, while having in fact, zero pins[2]. This machine includes a serial to parallel interface - the serial part is driven by JTAG, while the parallel is mapped into the PPC bus (embedded in BCSR's memory

[1]To improve crosstalk immunity for COP/JTAG signals, pins 2 and 10 were connected to GND. According to the general recommendation (and with ENG revision of this board) they should be (were) N.C.
[2]No Boundary Scan pins, Excluding the JTAG I/F pins, which are not count for that matter.

space). This configuration allows zero data overhead during downloads. The block diagram of the JTAG system on the ADS is shown in Figure 4-8 ". ADS JTAG SYSTEM" on page 70.

Figure 4-8. ADS JTAG SYSTEM



As can be seen in the figure above, the JTAG machine includes the following components:

1) TAP Controller
2) Instruction Shift register
3) Instruction register
4) Data Shift register (Download Data register)

5) Download Command and Status register

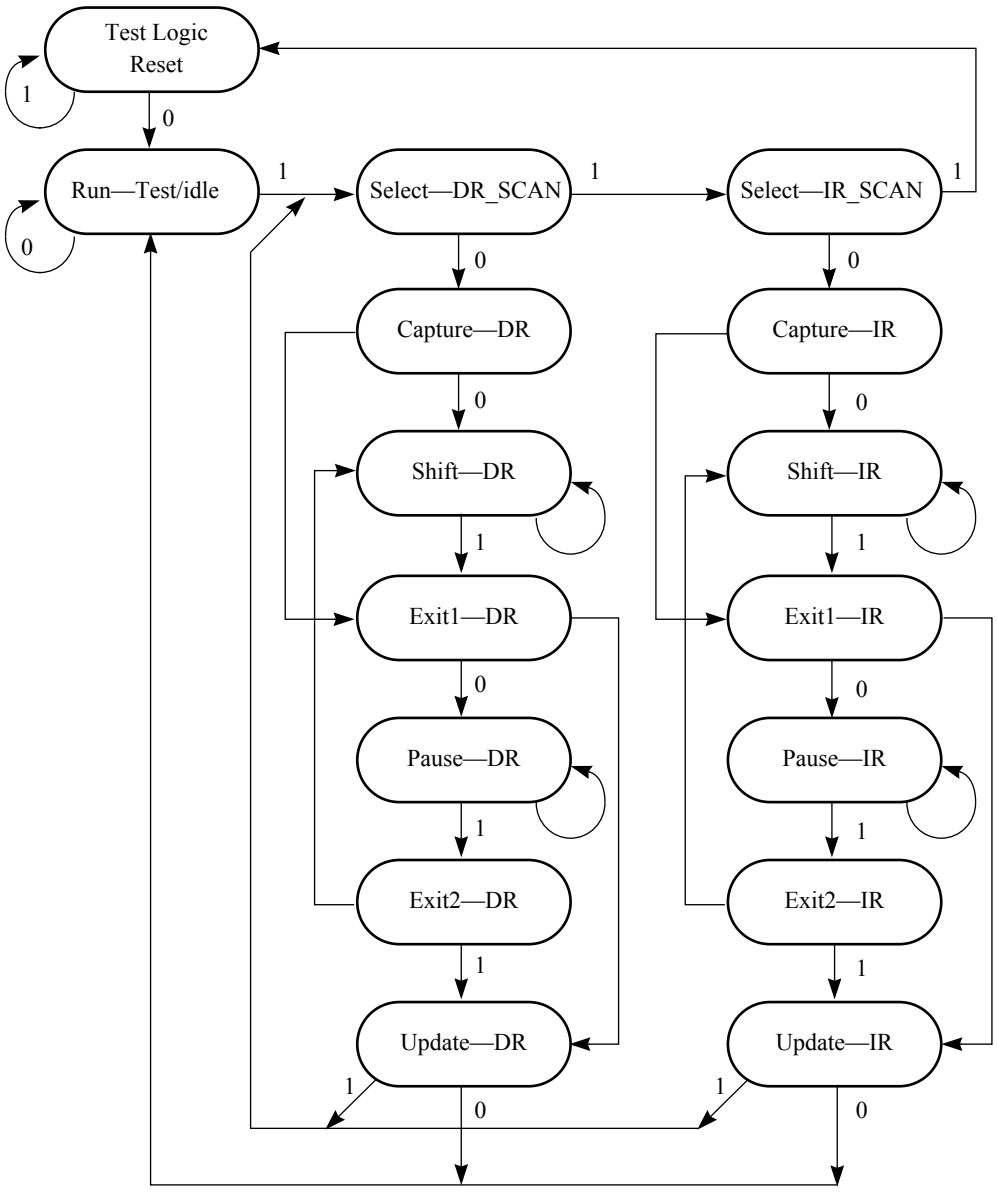
6) Bypass register

4.13.1.1 JTAG TAP Controller

The TAP (Test Access Port) controller is a standard 16-state JTAG TAP controller. Its transitions are determined by the state of TMS upon rising edge of TCK. The state "Test Logic Reset" may be reached synchronously (according to rising TCK) from any state by simply driving TMS high for 5 consecutive rising edges of TCK. The test logic is also reset asynchronously by either driving TRST low by the debug station or by Power On Reset generated by the board's Power-Up, or SW1 depressed, or by this logic[1].

The TAP controller state diagram is shown in Figure 4-9 ". JTAG TAP Controller State Diagram" on page 71.

Figure 4-9. JTAG TAP Controller State Diagram



[1] In a "Suicidal" manner.

4.13.1.2 JTAG Instruction Shift Register - JISR

The instruction shift register is a 3 bit register, selected during Shift-IR state of the TAP controller. The instruction is shifted in during that state, while the output of the shift register (LSB) is driven into MTDO, to be concatenated to the next device on the JTAG chain (the MPC8260). That way, all devices on the JTAG chain may be shifted-in with the desired instruction for them. When the TAP controller moves into Update-IR state, the JTAG Instruction register is loaded with the value shifted into JISR.

During JTAG logic reset, the JISR, is reset into a default state of Bypass.

4.13.1.3 JTAG Instruction Register - JIR

The JTAG Instruction register holds the current JTAG instruction, which determines the JTAG logic operation at any given time. The instructions are loaded into it from JISR when the TAP enters the Update-IR state. The valid instructions and their associated functions are shown in Table 4-21 ". JTAG Instruction Codes" below:

Table 4-21. JTAG Instruction Codes

Mnemonic	Code [Bin]	Function
EXTEST	'000'	**Execute Test**. JTAG public instruction. No function with this application, since there is no built-in test within this JTAG entity. **Defaults to Bypass**.
DOWNLOAD	'001'	**Download**. When JIR holds this instruction and the TAP controller is in Shift-DR mode, the input of the Download Shift register is connected to TDI - of the ADS, while the TDO of this machine reflects the status of JTAG_RX_FULL flag in the Download Control and Status register. Data may be shifted-in to be read by the download agent running on board. When the TAP controller moves into EXIT1-DR state, the Receive-Full flag is set in the Download Control & Status register, so that the download agent may learn about the presence of valid data in the JDSR.
UPLOAD	'010'	**Up-Load**. Not supported with current revision, **defaults to Bypass**.
SAMPLE/ PRELOAD	'011'	**Sample/Preload**. JTAG public instruction. Not implemented, since there are no external pins to this JTAG entity. **Defaults to Bypass**.
Reserved.	'100', '101'	**Reserved**, unimplemented, **defaults to Bypass**.
PON_RESET	'110'	**Power-On Reset**. When this code is loaded into the JIR, Power-On Reset is generated to the VADS, eventually resetting the TAP controller into Test-Logic Reset state. As a result the JIR is reset into Bypass code.
BYPASS	'111'	**Bypass**. JTAG public instruction. When the JIR holds this value and the TAP controller is in Shift-DR state, a single bit shift register is placed between the TDI-TDO of this JTAG machine.

4.13.1.4 Data Shift Register

The data shift register is an 8 bit shift register, shifted in (LSB first) on the rising edge of TCK. When JIR contains the DOWNLOAD code and the TAP controller is in Shift-DR state, the input of this shift register is connected to TDI input of the VADS. The TDO of this logic on the other hand, reflects the state of the JTAG_RX_FULL status bit in the Download Control & Status register. That way the host may check, prior to shifting in data, whether the agent has read the previous byte of data.

When JTAG_RX_FULL flag is active, data remains frozen in the register, until it is read by the agent. That way the host may shift in arbitrary data just to read back the status of JTAG_RX_FULL. Since that flag is set only by the passage through EXIT1_DR, this register will contain the last 8 bits shifted in.

This register is available on the PPC-Bus memory map D(0-7), so it may be read by a download agent running on-board. It is designated as BCSR7. For additional information on BCSR7 see Table 3-1 ". ADS

Memory Map" on page 34 and Table 4-20 ". BCSR7 Description" on page 68.

4.13.1.5 Download Control and Status Register

This register has no direct JTAG access, however, it enables the operation of this machine and contains status information, set by this machine. It is available on the PPC-Bus memory map, designated as BCSR6. For further information on BCSR6, see Table 3-1 ". ADS Memory Map" on page 34 and Table 4-19 ". BCSR6 Description" on page 68.

4.13.1.6 Bypass Register

The bypass register is a single stage register which must exist in any JTAG implementation. Its purpose, is to shorten the scan chain as much as possible for a device residing on the scan chain. When the JIR contains the BYPASS code and the TAP controller is in Shift-DR state, this register is placed between TDI-TDO pair of this machine.

Any unimplemented instruction with this machine, defaults to the BYPASS instruction.

4.13.1.7 JTAG Machine Bypass

There are 3 levels of bypass for this JTAG implementation, this to provide compatibility with earlier versions of this board and debug tools. The bypass options are:

1) **Hard-wired Bypass**: When J5 is set between positions 2-3, then the TDI input to the VADS, is connected directly to the TDI input of the MPC8260. This allows compatibility with existing debug tools which do not support the use of this machine and might suffer from added delay on TDI.
2) **Asynchronous Bypass**: After Power-On Reset, when J5 is set between positions 1-2 (factory set), this machine defaults to asynchronous connection between TDI input of this machine and the MPC8260's TDI (with a 7.5 nsec delay). This allows compatibility with existing debug tools which have not been modified yet to support this machine and can tolerate this delay on the TDI line.
3) **JTAG Bypass**: After Power-On reset or JTAG reset, when this JTAG machine is enabled via BCSR6 and J5 is set between positions 1-2 (factory set), it will be found in Bypass mode, i.e., the default instruction of the machine is BYPASS, so that when the TAP controller is moved into Shift-DR state, a single stage shift register is placed between the TDI input of the ADS and the TDI input of the MPC8260.

4.13.1.8 Fast Download Operation

This section describes the procedure needed to be taken by a debug-station programmer, so that this machine may be utilized. It is assumed here that J5 remains factory-set (1-2) and the ADS is after Power-On reset and initialized. The procedure is as follows:

1) Enable this JTAG machine by writing '1' to the JTAG_EN bit in BCSR6. The machine is now enabled and in JTAG Bypass mode. See Table 4-19 ". BCSR6 Description" on page 68.

   Remember that prior to this operation, the length of the instruction chain is 8 bit (MPC8260 only) while the data scan chain's length depended only on the scan selected within the MPC8260.

   After this operation, the length of the instruction chain is 11 bits (added 3 bits for this machine, preceding the MPC8260's in the chain), while the length of the data scan chain is added with either 1 bit (bypass) or 8 bit (download), preceding the MPC8260 in the chain (MSB side).
2) When this machine is in Bypass, download a s/w agent to free memory space, using the same method done with the previous revision of this board. Remember that the data scan chain is 1-bit longer than it used to be (MSB side).

This agent basically polls the JTAG_RX_FULL flag in the Download Command & Status register (see Table 4-19 ". BCSR6 Description" on page 68), when active - reads the Download Data register (See Table 4-20 ". BCSR7 Description" on page 68) and puts the data byte read, where required. Obviously the minimum it should know, are the base address and size of data buffer being loaded, however its level of sophistication is upto the system programmer.

Run this agent, again, using the same method done with the previous version of this board.

3) Shift in DOWNLOAD instruction for this machine and BYPASS instruction for the MPC8260.
4) Move the TAP controller into Shift-DR state
5) Shift in a Byte of data. The data shifted out, is the state of JTAG_RX_FULL flag. If the byte shifted out does not contain a '0', it indicates that new data was shifted in before the agent was able to read the previous. That way the host may become aware of an errornous situation with the operation of the s/w agent.

   To play safe, it is recommended to shift in a arbitrary data until JTAG_RX_FULL is found cleared. Then a byte of valid data may be shifted-in.
6) Move the TAP to EXIT1-DR -> PAUSE-DR -> EXIT2-DR. The passage through EXIT1-DR sets the JTAG_RX_FULL flag in the Download Command & Status register. The agent then, can read the valid data from the Download Data register. After that data has been read by the agent, the JTAG_RX_FULL is cleared.
7) Repeat steps (4) to (6) above until the end of the buffer.

4.13.2 JTAG Generated Power-On Reset

Since some of the COP debug-stations are Ethernet driven, a need may arise to generate Power-On Reset from a remote location, through the JTAG and to be able to do so, when the board is stuck. To support such action, the PON_RESET instruction was introduced.

When J5 is set to positions 1-2 and JTAG is enabled in the Download Control & Status register, it is possible[1] to Power-On Reset the board through JTAG. The way to do it is:

1) Move the TAP controller into Shift-IR
2) Shift in PON_RESET code.

As a result of the above Power-On reset is generated, resetting the ADS including this JTAG machine, which goes back into "Test Logic Reset" state. The JTAG_EN bit in the Download Control & Status register is reset as well.

Before re-initializing the board, there is a need to wait about 3 seconds, for the board to recover. The state of HESET~ and SRESET~ lines of the MPC8260, is visible via the COP/JTAG connector.

4.14 Power

There are 4 power buses with the MPC8260:

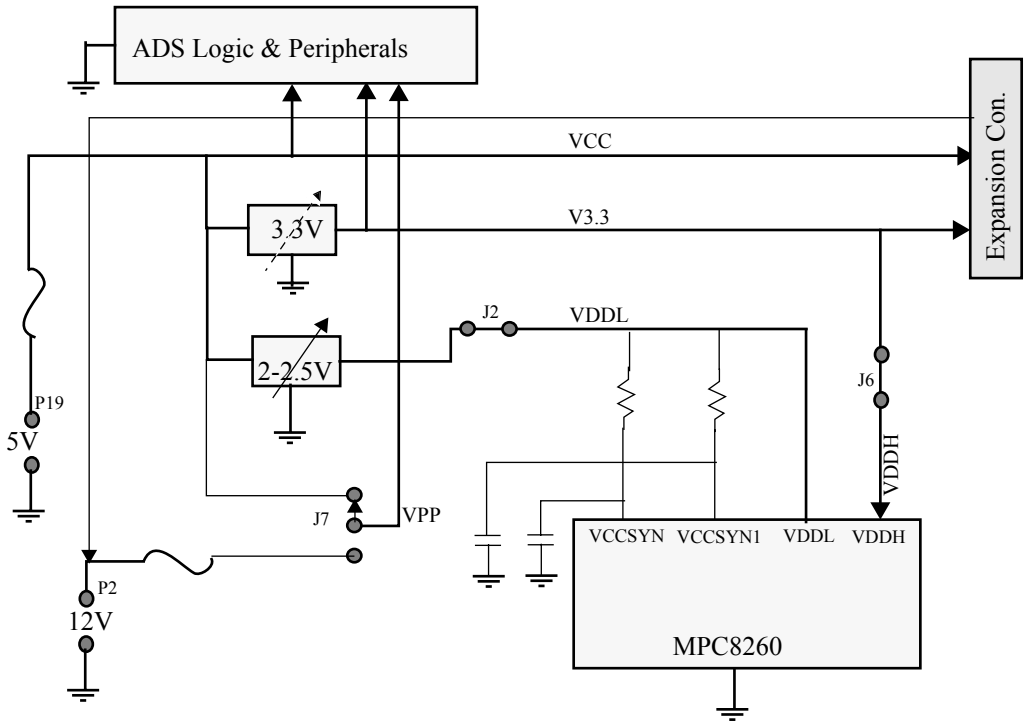
1) VDDH (I/O)
2) VDDL (Internal Logic)
3) VCCSYN (CPM PLL)
4) VCCSYN1 (Core PLL)

and there are 4 power buses on the ADS:

[1]Since HRESET_ and SRESET_ lines appear at the COP/JTAG connector (P5), it is possible to generate Hard-Reset and Soft-Reset, directly. Power-On reset however, may be remote generated only through JTAG.

1) VCC (5V) bus
2) V3.3 (3.3V) bus
3) VDDL (2V-2.5V) bus
4) VPP (5V / 12V) bus

Figure 4-10. ADS Power Scheme



To support off-board application development, two of the power buses are connected to the expansion connectors, so that external logic may be powered directly from the board. The maximum current allowed to be drawn from the board on each bus is shown in Table 4-22 ". Off-board Application Maximum Current Consumption" below:

Table 4-22. Off-board Application Maximum Current Consumption

Power Bus	Max. Current
VCC	2A
V3.3	1.5A

As can be seen from Figure 4-10 ". ADS Power Scheme" above, VPP may be provided also from the expansion connectors. This option is provided for production testing.

To protect on-board devices against supply spikes, decoupling capacitors (typically 0.1µF) are provided between the devices' power leads and GND, located as close as possible to the power leads, while 47 µF bulk capacitors are spread around.

4.14.1 5V Bus

Some of the ADS peripherals reside on the 5V bus. Since the MPC8260 is not 5V tolerant, buffering is provided between 5V peripherals and the MPC8260, protecting the MPC8260 from the higher voltage level. The 5V bus is connected to an external power connector via a fuse (5A).

To protect against reverse-voltage or over-voltage being applied to the 5V inputs a set of high-current diodes and zener diode is connected between the 5V bus GND. When either over or reverse voltage is applied to the ADS, the protection logic blows the fuse, while limiting the momentary effects on board.

4.14.2 3.3V Bus

The MPC8260, the SDRAMs, the address and data buffers are powered by the 3.3 bus, which is produced from the 5V bus using a low-voltage drop, linear voltage regulator made by Micrel, the MIC29501-3.3BU which is capable of driving upto 5A, facilitating operation of external logic as well.

With this revision of the ADS, a production option is made so that the level on this bus may be varied by means of trimming potentiometer - TR2. However this also requires the following changes:

1) U11 to be changed to MIC29752BWT, which is a 7.5A, 5 pin, variable regulator.
2) Soldering in R74 (5KΩ, 1206), R89 (3KW, 1206) & TR2 (1KΩ, BOURNS 3362P-1-102) Table 5-8 ". MPC8260ADS Bill Of Material" on page 94.

WARNING

The values for the above components are calculated to allow 3.0V - 3.6V range over the 3.3V bus. A change in the above values, might extend the voltage over this bus to undesired levels, presence of which, might inflict PERMANENT DAMAGE to the ADS.

4.14.3 VDDL Bus

The MPC8260's internal logic and the PLL are powered with a lower-voltage power source, voltage of which may be in 3 ranges of levels:

- 2.3V - 2.7V
- 1.7V - 1.9V
- 1.8V - 2.0V

Selection between the above range levels is done via J1, which selects between different resistor values within the VDDL's variable regulator (U10) feedback network, while the fine tuning with a range is done by means of a trimming potentiometer - TR1. For the different settings of J1 and their corresponding voltage level ranges see 2.3.1 "Setting VDDL Level Range - J1" on page 19 and 2.3.2 "Setting VDDL Supply Voltage Level" on page 19.

Changing the voltage to the Core logic of the MPC8260, obviously has an influence over the maximal speed of the core. There is the power-speed trade-off, i.e., lower operation speeds may be obtained with lower voltage supply.

4.14.4 VPP Bus

The sole purpose of the VPP bus is to supply VPP (programming voltage) for the Flash SIMM[1]. VPP may be connected to either 5V bus or 12V bus - selected via J7. Normally VPP is connected to the 5V bus (see Figure 3-4 ". J7 - VPP Source Selection" on page 32)

The 12V bus is connected to a dedicated input connector (P2) via a fuse (1A) and is protected from over / reverse voltage appliance, in the same manner done with the 5V bus.

If either the Flash SIMM is 5V programmable or it is 12V programmable but need not be programmed - the 12V supply input connector of the ADS - P2, may be left un-connected.

[1] If necessary.

Chapter 5 Support Information

In this chapter all information needed for support, maintenance and connectivity to the MPC8260ADS is provided.

5.1 Interconnect Signals

The MPC8260ADS interconnects with external devices via the following set of connectors:

1) P1 - 100 / 10 - Base-T Ethernet port
2) P2 - 12V Power Supply
3) PA3 - RS232 port 2 (lower)
4) PB3 - RS232 port 1 (upper)
5) P4 - CPM Expansion
6) P5 - COP / JTAG
7) P6, P7, P8, P9, P10, P12, P13, P14 & P15 - Logic Analyzer MICTOR Connectors
8) P11 - Mach's In System Programming (ISP)
9) P16 - System Expansion

5.1.1 P1 - Ethernet Port Connector

The Ethernet connector on the MPC8260ADS - P1, is a Twisted-Pair (10-Base-T) compatible connector. It is implemented with a 90^{o}, 8-pin, RJ45 connector, signals of which are described in Table 5-1 ". P1 - Ethernet Port Interconnect Signals" below

Table 5-1. P1 - Ethernet Port Interconnect Signals

Pin No.	Signal Name	Description
1	TPTX	Twisted-Pair Transmit Data positive output from the MPC8260ADS.
2	TPTX~	Twisted-Pair Transmit Data negative output from the MPC8260ADS.
3	TPRX	Twisted-Pair Receive Data positive input to the MPC8260ADS.
4	N.C.	Not connected, Bob Smith terminated on the MPC8260ADS.
5		
6	TPRX~	Twisted-Pair Receive Data negative input to the MPC8260ADS.
7	N.C.	Not connected, Bob Smith terminated on the MPC8260ADS.
8		

5.1.2 P2 - 12V Power Connector

The 12V power connector - P2, is a two-lead, 2 part, terminal block connector. P2 supplies, when necessary,

programming voltage to the Flash SIMM.

Table 5-2. P2 - Interconnect Signals

Pin Number	Signal Name	Description
1	12V	12V input from external power supply.
2	GND	GND line from external power supply.

5.1.3 PA3, PB3 - RS232 Ports' Connectors

The RS232 ports' connectors - PA3 and PB3 are 9 pin, 90°, female D-Type Stacked connectors, signals of which are presented in Table 5-3 ". PA3, PB3 Interconnect Signals" below

Table 5-3. PA3, PB3 Interconnect Signals[1]

Pin No.	Signal Name	Description
1	CD	Carrier Detect **output** from the MPC8260ADS.
2	TX	Transmit Data **output** from the MPC8260ADS.
3	RX	Receive Data **input** to the MPC8260ADS.
4	DTR	Data Terminal Ready **input** to the MPC8260ADS.
5	GND	Ground signal of the MPC8260ADS.
6	DSR	Data Set Ready **output** from the MPC8260ADS.
7	N.C.	No connect
8	CTS	Clear To Send **output** from the MPC8260ADS.
9	N.C.	No connect

[1]Refer to 4.11.3 "RS232 Ports" on page 60.

5.1.4 MPC8260ADS's P4 - CPM Expansion Connector

P4 is a 128 pin, 90^0, DIN 41612 connector, which allows for convenient expansion of the MPC8260's serial ports. This connector contains all CPM pins plus power supply pins, to provide for easy tool connection. The pinout of P4 is shown in Table 5-4 ". P4 - CPM Expansion - Interconnect Signals" below:

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
A1	RS_RXD1 (PD31[1])	I/O, T.S.	When RS232 port #1 is enabled, this signal is the receive data line for that port. When this port is disabled, this signal is tristated and may be used to any available alternate function for PD31.
A2	RS_TXD1 (PD30)	I/O, T.S.	When RS232 port #1 is enabled, this signal is the transmit data line for that port. When this port is disabled, this signal may be used to any available alternate function for PD30.
A3	PD29	I/O, T.S	MPC8260's Port D 29 line. Parallel I/O or CPM dedicated line.May be used for any of it's available functions.

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
A4	RS_RXD2 (PD28)	I/O, T.S.	When RS232 port #2 is enabled, this signal is the receive data line for that port. When this port is disabled, this signal is tristated and may be used to any available alternate function for PD28.
A5	RS_TXD2 (PD27)	I/O, T.S.	When RS232 port #2 is enabled, this signal is the transmit data line for that port. When this port is disabled, this signal may be used to any available alternate function for PD27.
A6	PD26	I/O, T.S.	MPC8260's PD(26:18) Port D lines. Parallel I/O or CPM dedicated lines. May be used for any of their available functions.
A7	PD25		
A8	PD24		
A9	PD23		
A10	PD22		
A11	PD21		
A12	PD20		
A13	PD19		
A14	PD18		
A15	ATMRXPTY (PD17)	I/O, T.S.	ATM Receive Parity Line. When the ATM port is enabled, this line is connected to the receive parity of the PM5350 ATM UNI. When this port is disabled, this signal is tristated and may be used for any available function of PD17.
A16	ATMTXPTY (PD16)	I/O, T.S.	ATM Transmit Parity Line. When the ATM port is enabled, this line is connected to the transmit parity of the PM5350 ATM UNI. When this port is disabled, this signal may be used for any available function of PD16.
A17	SDCFGDT (PD15)	I/O, T.S.	SDRAM Configuration Data. This signal is connected to the serial I^2C data line of the SDRAM DIMM configuration Serial EEPROM, holding all DIMMs configuration data. This line is used, in conjunction with SDCFGCK, to read or write this memory, using the I^2C protocol. This line may be used off-board as an I^2C data line for external I^2C device, as long as the I^2C addresses of the SDRAM DIMM and the external device's - do not conflict. See 2.4.8.2 "SDRAM DIMM Installation" on page 27 and 4.7 "Synchronous DRAM DIMM (60X Bus)" on page 49.
A18	SDCFGCK (PD14)	I/O, T.S.	SDRAM Configuration Clock. This signal is connected to the serial I^2C clock line of the SDRAM DIMM configuration Serial EEPROM, holding all DIMMs configuration data. This line is used, in conjunction with SDCFGDT, to read or write this memory, using the I^2C protocol. This line may be used off-board as an I^2C clock line for external I^2C device, as long as the I^2C addresses of the SDRAM DIMM and the external device's - do not conflict. See 2.4.8.2 "SDRAM DIMM Installation" on page 27 and 4.7 "Synchronous DRAM DIMM (60X Bus)" on page 49

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
A19	PD13	I/O, T.S.	MPC8260's PD(13:4) Port D lines. Parallel I/O or CPM dedicated lines. May be used for any of their available functions.
A20	PD12		
A21	PD11		
A22	PD10		
A23	PD9		
A24	PD8		
A25	PD7		
A26	PD6		
A27	PD5		
A28	PD4		
A29	ATRCKDIS	I	ATM Receive Clock Out Disable. When active (H), the ATMRCLK output, on pin C29 of this connector, is Tri-stated. When either not connected or driven low, ATMRCLK on pin C29, is enabled. This provides compatibility with ENG revision of T/ECOM communication tools.
A30	VCC	O	5V Supply. Connected to ADS's 5V VCC plane. Provided as power supply for external tool. For allowed current draw, see Table 4-22 ". Off-board Application Maximum Current Consumption" on page 75.
A31			
A32			
B1	ATMTXEN~ (PA31)	I/O, T.S.	ATM Transmit Enabled (L). When this signal is asserted (Low), while the ATM port is enabled and ATMTFCLK is rising, an octet of data, ATMTXD(7:0), is written into the transmit FIFO of the PM5350. When the ATM port is disabled, this line may be used for any available function of PA31.
B2	ATMTCA (PA30)	I/O, T.S.	ATM Transmit Cell Available (H). When this signal is asserted (High), while the ATM port is enabled, it indicates that the transmit FIFO of the PM5350 is empty and ready to except a new cell. When negated, it may show either that the transmit FIFO is Full or close to Full, depending on PM5350 internal programming. When the ATM port is disabled, this line may be used for any available function of PA30.
B3	ATMTSOC (PA29)	I/O, T.S.	ATM Transmit Start Of Cell (H). When this signal is asserted (High) by the MPC8260, while the ATM port is enabled, it indicates to the PM5350 the start of a new ATM cell over ATMTXD(7:0), i.e., the 1'st octet is present there. When the ATM port is disabled, this line may be used for any available function of PA29.
B4	ATMRXEN~ (PA28)	I/O, T.S.	ATM Receive Enable (L). When this signal is asserted (Low), while the ATM port is enabled and ATMRFCLK[2] goes high, on octet of data is available at the PM5350's ATMRXD(7:0) lines. When negated while ATMRFCLK goes high data on ATMRXD(7:0) is invalid, however driven. When the ATM port is disabled, this line may be used for any available function for PA28.

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
B5	ATMRSOC (PA27)	I/O, T.S.	ATM Receive Start Of Cell (H). When this signal is asserted (High), while the ATM port is enabled, it indicates, that the 1'st octet of data for the received cell is available at the PM5350's ATMRXD(7:0) lines. This line is updated over the rising edge of ATMRFCLK. When the ATM port is disabled, this line is tristated and may be used for any available function for PA27.
B6	ATMRCA (PA26)	I/O, T.S.	ATM Receive Cell Available (H). When this signal is asserted (High), while the ATM port is enabled and ATMRFCLK goes high, it indicates that the PM5350's receive FIFO is either full or that there are 4 empty bytes left in it - PM5350 internal programming dependent. When the ATM port is disabled, this line is tristated and may be used for any available function of PA26.
B7	ATMTXD0 (PA25)	I/O, T.S.	ATM Transmit Data (7^3:0). When the ATM port is enabled, this bus carries the ATM cell octets, written to the PM5350's transmit FIFO. This bus is considered valid only when ATMTXEN~ is asserted and are sampled on the rising edge of ATMTFCLK. When the ATM port is disabled, these lines may be used for any available respective function.
B8	ATMTXD1 (PA24)		
B9	ATMTXD2 (PA23)		
B10	ATMTXD3 (PA22)		
B11	ATMTXD4 (PA21)		
B12	ATMTXD5 (PA20)		
B13	ATMTXD6 (PA19)		
B14	ATMTXD7 (PA18)		
B15	ATMRXD7 (PA17)	I/O, T.S.	ATM Receive Data (7^3:0). When the ATM port is enabled, this bus carries the cell octets, read from the PM5350 receive FIFO. This lines are updated on the rising edge of ATMRFCLK^2. When the ATM port is disabled, these lines are tristated and may be used for any available respective function.
B16	ATMRXD6 (PA16)		
B17	ATMRXD5 (PA15)		
B18	ATMRXD4 (PA14)		
B19	ATMRXD3 (PA13)		
B20	ATMRXD2 (PA12)		
B21	ATMRXD1 (PA11)		
B22	ATMRXD0 (PA10)		

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
B23	PA9	I/O, T.S.	MPC8260's Port A (9:0). Parallel I/O or dedicated CPM lines. May be used for any of their available functions.
B24	PA8		
B25	PA7		
B26	PA6		
B27	PA5		
B28	PA4		
B29	PA3		
B30	PA2		
B31	PA1		
B32	PA0		
C1	FETHTXER (PB31)	I/O, T.S.	Fast-Ethernet[4] Transmit Error (H). When the Ethernet port is enabled, this signal will be asserted (High) by the MPC8260 when an error is discovered in the transmit data stream. When the port is operation at 100 Mbps, the LXT970 responds by sending invalid code symbols on the line. When the Ethernet port is disabled, this line may be used for any available function of PB31.
C2	FETHRXDV (PB30)	I/O, T.S.	Fast-Ethernet Receive Data Valid (H). When this signal is asserted (High) while the Fast Ethernet port is enabled and FETHRXCK goes high, it indicates that data is valid on the MII Receive Data lines - FETHRXD(3:0). When the Fast Ethernet port is disabled, this line is tristated and may be used for any available function go PB30.
C3	FETHTXEN (PB29)	I/O, T.S.	Fast-Ethernet Transmit Enable (H). The MPC8260 will assert (High) this line, to indicate data valid on the FETHTXD(3:0) lines. When the Fast-Ethernet port is disabled, this line may be used for any available function of PB29.
C4	FETHRXER (PB28)	I/O, T.S.	Fast-Ethernet Receive Error (H). When this signal is asserted (High) by the LXT970, while the Ethernet port is enabled and FETHRXCK goes high, it indicates that the port is receiving invalid data symbols from the network. When the Ethernet port is disabled, this line is tristated and may be used for any available function of PB28.
C5	FETHCOL (PB27)	I/O, T.S.	Fast-Ethernet Port Collision Detected (H). When this signal is asserted (High) by the LXT970, while the ethernet port is enabled, it indicates a Collision state over the line. When the LXT970 is in Full-Duplex mode, this line is inactive. When the Ethernet port is disabled, this line is tristated and may be used for any available function of the PB27.

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
C6	FETHCRS (PB26)	I/O, T.S.	Fast-Ethernet Carrier Sense (H). When this signal is asserted (High), while the Ethernet port is enabled and the LXT970 is in half-duplex mode, it indicates that either the transmit or receive media are non-idle. When the LXT970 is in either full-duplex or repeater operation, it indicates that the receive medium is non-idle. When the Ethernet port is disabled, this line may be used for any available function of PB26.
C7	FETHTXD3 (PB25)	I/O, T.S.	Fast Ethernet Transmit Data (3:0). This is the MII transmit data bus. The MPC8260 drives these lines according to rising edge of FETHTXCK. When the ethernet port is disabled, these lines may be used for any available respective function.
C8	FETHTXD2 (PB24)		
C9	FETHTXD1 (PB23)		
C10	FETHTXD0 (PB22)		
C11	FETHRXD0 (PB21)	I/O, T.S.	Fast Ethernet Receive Data (3:0). This is the MII receive data bus. The LXT970 drives these lines according to rising edge of FETHRXCK. When the ethernet port is disabled, these lines are tristated and may be used for any available respective parenthesized function.
C12	FETHRXD1 (PB20)		
C13	FETHRXD2 (PB19)		
C14	FETHRXD3 (PB18)		
C15	PB17	I/O, T.S.	MPC8260's Port B (17:4) Parallel I/O lines. May be used to any of their available functions.
C16	PB16		
C17	PB15		
C18	PB14		
C19	PB13		
C20	PB12		
C21	PB11		
C22	PB10		
C23	PB9		
C24	PB8		
C25	PB7		
C26	PB6		
C27	PB5		
C28	PB4		
C29	ATMRCLK	O, T.S.	ATM Receive Clock. A divide by 8 of the ATM line clock recovered by the ATM receive logic. Added with PILOT revision of the ADS, to assist Circuit Emulation Tool. Enabled only when pin A29 of this connector is either not connected or driven low. Otherwise, Tri-stated.+
C30	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C31			
C32			

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
D1	PC31	I/O, T.S.	MPC8260's Port C (31:22) Parallel I/O lines. May be used to any of their available functions.
D2	PC30		
D3	PC29		
D4	PC28		
D5	PC27		
D6	PC26		
D7	PC25		
D8	PC24		
D9	PC23		
D10	PC22		
D11	ATMTFCLK (PC21)	I/O, T.S.	ATM Transmit FIFO Clock. Upon the rising edge of this clock (driven by the MPC8260), while the ATM port is enabled, the cell octets are written to the PM5350's transmit FIFO. This clock samples ATMTXD(7:0), ATMTXPTY, ATMTXEN~ and ATMTSOC. When the ATM port is disabled, this line may be used for any available function of PC21.
D12	PC20	I/O, T.S.	MPC8260's Parallel I/O Port-C 20. Parallel I/O line. May be used for any of its available functions
D13	FETHRXCK (PC19)	I/O, T.S.	Fast-Ethernet Receive Clock. When the Ethernet port is enabled, this clock (25 MHz for 100 Mbps, 2.5 MHz for 10 Mbps) is extracted from the received data and driven to the MPC8260 to qualify incoming receive data. When the Ethernet port is disabled, this line is tristated and may be used for any available function of PC19
D14	FETHTXCK (PC18)	I/O, T.S.	Fast-Ethernet Transmit Clock. When the Ethernet port is enabled, this clock (25 MHz for 100 Mbps, 2.5 MHz for 10 Mbps) is normally extracted from the received data and driven to the MPC8260 to qualify out coming transmit data. In Slave mode (not used with this application) this clock should be input to the LXT970. When the Ethernet port is disabled, this line is tristated and may be used for any available function of PC18
D15	PC17	I/O, T.S.	MPC8260's Port C (17:15) Parallel I/O lines. May be used to any of their available functions.
D16	PC16		
D17	PC15		
D18	RS_CD1~ (PC14)	I/O, T.S.	RS232 Port 1 Carrier Detect (L). Connected via RS232 transceiver to RS232 DTR1~ input, allowing detection of a connected terminal to this port. This line is simply a PI/O input line to the MPC8260. When RS232 Port 1 is disabled, this line is tristated and may be used for any available function of PC14.
D19	PC13	I/O, T.S.	MPC8260's Port C 13 Parallel I/O line. May be used to any of its available functions.

Table 5-4. P4 - CPM Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
D20	RS_CD2~ (PC12)	I/O, T.S.	RS232 Port 2 Carrier Detect (L). Connected via RS232 transceiver to RS232 DTR2~ input, allowing detection of a connected terminal to this port. This line is simply a PI/O input line to the MPC8260. When RS232 Port 2 is disabled, this line is tristated and may be used for any available function of PC12.
D21	PC11	I/O, T.S.	MPC8260's Port C 11 Parallel I/O line. May be used to any of its available functions.
D22	FETHMDC (PC10)	I/O, T.S.	Fast-Ethernet Port Management Data Clock. This slow clock (S/W generated) qualifies the management data I/O to read / write the LXT970's internal registers. When the Ethernet port is disabled, this line may be used for any available function of PC10.
D23	FETHMDIO (PC9)	I/O, T.S.	Fast-Ethernet Port Management Data I/O. This signal serves as bidirectional serial data line, qualified by FETHMDC, to allow read / write the LXT970's internal registers. When the Ethernet port is disabled, this line may be used for any available function of PC9.
D24	PC8	I/O, T.S.	MPC8260's Port C (8:0) Parallel I/O lines. May be used to any of their available functions.
D25	PC7		
D26	PC6		
D27	PC5		
D28	PC4		
D29	PC3		
D30	PC2		
D31	PC1		
D32	PC0		

[1]The functions in parenthesis, are MPC8260's parallel I/Os.

[2]Normally connected to ATMTFCLK on the ADS.

[3]MS bit.

[4]For that matter, both 100-Base-T and 10-Base-T.

5.1.5 P5 - COP / JTAG Port Connector

P5 is a Motorola standard COP / JTAG connector for the 60X processors family. It is a 16 pin 90^{o} protected header connector with locks. The pinout of P5 is shown in Table 5-5 ". P5 - COP / JTAG Connector - Interconnect Signals" below:

Table 5-5. P5 - COP / JTAG Connector - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
1	TDO	O	Transmit Data Output. This the MPC8260's JTAG serial data output driven by Falling edge of TCK.

Table 5-5. P5 - COP / JTAG Connector - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
2	GND	O	Digital GND. Main GND plane. (Was N.C. with ENG revision)
3	TDI	I	Transmit Data In. This is the JTAG serial data input of the ADS, sampled on the rising edge of TCK. May be connected to either BCSR or directly to the MPC8260, J5 depended. See 3.2.12 "J5 - COP/ JTAG TDI Source Selection" on page 31 and 4.13 "COP/JTAG Port" on page 68.
4	TRST~	I	Test port Reset~ (L). When this signal is active (Low), it resets the JTAG logic of both the MPC8260 and the Fast Download machine. This line is pull-down on the ADS with a 1KΩ resistor, to provide constant reset of the JTAG logic.
5	QREQ~	O	Quiescent Request (L). When asserted (low), this line indicates that the MPC8260 desires to enter low-power mode. This signal may be required by a debug station.
6	V3.3	O	3.3V power supply bus.
7	TCK	I	Test port Clock. This clock shifts in / out data to / from the MPC8260ADS JTAG logic. Data is driven on the falling edge of TCK and is sampled both internally and externally on it's rising edge. TCK is pulled up internally by the MPC8260.
8	N.C.	-	Not Connected.
9	TMS	I	Test Mode Select. This signal qualified with TCK in a same manner as TDI, changes the state of the JTAG machines. This line is pulled up internally by the MPC8260.
10	GND	O	Digital GND. Main GND plane. (Was N.C. with ENG revision)
11	SRESET~	I/O, O.D.	Soft Reset (L). This is the MPC8260's soft reset which is in fact a non-maskable interrupt, making the PPC take the reset exception from the reset vector. This line may be driven by the MPC8260 as well during soft-reset sequence, for 512 system clocks. This line is pulled up on the ADS with a 1KΩ resistor. When driven externally, it **MUST be driven with an Open Drain gate. Failure to do so may result in permanent damage to the MPC8260 and / or to ADS logic.**
12	GND	O	Digital GND. Main GND plane.
13	HRESET~	I/O, O.D.	MPC8260's Hard Reset (L). When asserted by an external H/W, generates Hard-Reset sequence for the MPC8260. During that sequence, asserted by the MPC8260 for 512 system clocks. Pulled Up on the ADS using a 1KΩ resistor. When driven by an external tool, **MUST be driven with an Open Drain gate. Failure to do so may result in permanent damage to the MPC8260 and / or to ADS logic.**
14	N.C.	-	Not Connected.
15	XBR3~ (CKSTOP_OUT~)	I/O	Normally configured as XBR3~ which has no function with this connector. May be configured as CKSTP_OUT~ - Check Stop Out (L). When asserted (Low) indicates that the MPC8260 core has entered a Check-Stop state.
16	GND	O	Digital GND. Main GND plane.

5.1.6 P6, P7, P8, P9, P10, P12, P13, P14 & P15 - Logic Analyzer Connectors

These are 38 pin, SMT, high density, matched impedance connector made by AMP. They contain all MPC8260 signals unbuffered. The pinout of these connectors is shown in Chapter 6, “Schematics.”

For signal description of these connectors, see the MPC8260 User’s Manual.

5.1.7 P11 - Mach’s In System Programming (ISP)

This is a 10 pin generic 0.100" pitch header connector, providing In System Programming capability for Vantis made programmable logic on board. The pinout of P11 is shown in Table 5-6 ". P11 - ISP Connector - Interconnect Signals" below:

Table 5-6. P11 - ISP Connector - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
1	ISPTCK	I	ISP Test port Clock. This clock shifts in / out data to / from the programmable logic JTAG chain.
2	N.C.	-	Not Connected.
3	ISPTMS	I	ISP Test Mode Select. This signal qualified with ISPTCK, changes the state of the prog. logic JTAG machine.
4	GND	O	Digital GND. Main GND plane.
5	ISPTDI	I	ISP Transmit Data In. This is the prog. logic’s JTAG serial data input, sampled by the MPC8260 on the rising edge of TCK.
6	VCC	O	5V power supply bus.
7	ISPTDO	O	ISP Transmit Data Output. This the prog. logic’s JTAG serial data output driven by Falling edge of TCK.
8	GND	O	Digital GND. Main GND plane.
9	N.C.	-	Not Connected.
10	N.C.	-	Not Connected.

5.1.8 MPC8260ADS’s P16 - System Expansion Connector

P16 is a 128 pin, 90^0, DIN 41612 connector, which provide a minimal system I/F required to interface various types of communication transceivers, data path of which passes through MPC8260’s serial ports via P4. This connector contains 16 bit (lower PPC bus) address lines, 16 bit (higher PPC bus) Data lines plus useful GPCM and UPM control lines. The pinout of P16 is shown in Table 5-7 ". P16 - System Expansion -

Interconnect Signals" below:

Table 5-7. P16 - System Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
A1	EXPA16	O	Expansion Address (16^1:31). This is a Latched-Buffered version of the MPC8260's PPC Address lines (16:31), provided for external tool connection. To avoid reflection these lines are series terminated with 43 Ω resistors.
A2	EXPA17		
A3	EXPA18		
A4	EXPA19		
A5	EXPA20		
A6	EXPA21		
A7	EXPA22		
A8	EXPA23		
A9	EXPA24		
A10	EXPA25		
A11	EXPA26		
A12	EXPA27		
A13	EXPA28		
A14	EXPA29		
A15	EXPA30		
A16	EXPA31		
A17	VPPIN	I	This signal is provided to allow Flash programming while the ADS resides in a Card Cage during manufacturing process. These lines may be connected to a 12V / 1A power supply, provided that P2 is disconnected.
A18			
A19	N.C.	-	Not Connected.
A20	V3.3	O	3.3V Power Out. These lines are connected to the main 3.3V plane of the MPC8260ADS, this, to provide 3.3V power where necessary for external tool connected. The amount of current allowed to be drawn from this power bus is found in Table 4-22 ". Off-board Application Maximum Current Consumption" on page 75.
A21			
A22			
A23			
A24			
A25	N.C.	-	Not Connected.

Table 5-7. P16 - System Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
A26	VCC	O	5V Supply. Connected to ADS's 5V VCC plane. Provided as power supply for external tool. For allowed current draw, see Table 4-22 ". Off-board Application Maximum Current Consumption" on page 75.
A27			
A28			
A29			
A30			
A31			
A32			
B1	GND	O	Digital Ground. Connected to main GND plane of the ADS.
B2			
B3			
B4	TSTAT0	I	Tool Status (0^1:7). This lines may be driven by an external tool to be read via BCSR2 of the ADS. These lines are pulled-up on the ADS, by 10 KΩ resistors. See also Table 4-11 ". BCSR2 Description" on page 64.
B5	TSTAT1		
B6	TSTAT2		
B7	TSTAT3		
B8	TSTAT4		
B9	TSTAT5		
B10	TSTAT6		
B11	TSTAT7		
B12	TOOLREV0	I	Tool Revision (0^1:3). This lines should be driven by an external tool with the Tool Revision Code, to be read via BCSR2 of the ADS. These lines are pulled-up on the ADS, by 10 KΩ resistors. See also Table 4-11 ". BCSR2 Description" on page 64.
B13	TOOLREV1		
B14	TOOLREV2		
B15	TOOLREV3		
B16	EXTOLI0	I	External Tool Identification (0^1:3). This lines should be driven by an external tool with the Tool Identification Code, to be read via BCSR2 of the ADS. These lines are pulled-up on the ADS, by 10 KΩ resistors. See also Table 4-11 ". BCSR2 Description" on page 64
B17	EXTOLI1		
B18	EXTOLI2		
B19	EXTOLI3		
B20	N.C.	-	Not Connected
B21	V3.3	O	3.3V Power Out. These lines are connected to the main 3.3V plane of the MPC8260ADS, this, to provide 3.3V power where necessary for external tool connected. The amount of current allowed to be drawn from this power bus is found in Table 4-22 ". Off-board Application Maximum Current Consumption" on page 75.
B22			
B23			
B24			
B25	N.C.	-	Not Connected

Table 5-7. P16 - System Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
B26	VCC	O	5V Supply. Connected to ADS's 5V VCC plane. Provided as power supply for external tool. For allowed current draw, see Table 4-22 ". Off-board Application Maximum Current Consumption" on page 75.
B27			
B28			
B29			
B30			
B31			
B32			
C1	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C2	CLK8	O	Buffered System Clock. This is a low skew buffered version of the MPC8260's CLKIN signal, to be used by an external tool.
C3	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C4	BTOLCS1~	O	Buffered Tool Chip Select 1 (L). This is a buffered MPC8260's CS6~ line, reserved for an external tool.
C5	BTOLCS2~	O	Buffered Tool Chip Select 2 (L). This is a buffered MPC8260's CS7~ line, reserved for an external tool.
C6	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C7	ATMEN~	O	ATM Port Enable (L). This line enables the ATM port UNI's output lines towards the MPC8260. An external tool, using the same pins as does the ATM port should consult this signal before driving the same lines. **Failure to do so might result in permanent damage to the PM5350 ATM UNI.**
C8	ATMRST~	O	ATM Port Reset (L). This signal resets the ATM UNI (PM5350). An external tool may use this signal to its benefit.
C9	FETHRST~	O	Ethernet Port Reset (L). This signal resets the LXT970 Ethernet transceiver. An external tool may use this signal to its benefit.
C10	HRESET~	I/O, O.D.	MPC8260's Hard Reset (L). When asserted by an external H/W, generates Hard-Reset sequence for the MPC8260. During that sequence, asserted by the MPC8260 for 512 system clocks. Pulled Up on the ADS using a 1KΩ resistor. When driven by an external tool, **MUST be driven with an Open Drain gate. Failure to do so might result in permanent damage to the MPC8260 and / or to ADS logic.**
C11	IRQ6~	I	Interrupt Request 6 (L). Connected to MPC8260's DP6/CSE0/IRQ6~ signal. Pulled up on the ADS with a 10 KΩ resistor. This line is shared with the ATM UNI's interrupt line and therefore, when driven by an external tool, **MUST be driven with an Open Drain gate. Failure to do so may result in permanent damage to the MPC8260 or to ADS logic.**

Table 5-7. P16 - System Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
C12	IRQ7~	I	Interrupt Request 7 (L). Connected to MPC8260's DP7/CSE1/IRQ7~ signal. Pulled up on the ADS with a 10 KΩ resistor. This line is shared with the Fast Ethernet transceiver's interrupt line and therefore, when driven by an external tool, **MUST be driven with an Open Drain gate. Failure to do so might result in permanent damage to the MPC8260 and / or to ADS logic.**
C13	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C14	EXPD0	I/O, T.S.	Expansion Data (0^1:15). This is a double buffered version of the PPC bus D(0:15) lines, controlled by on-board logic. These lines will be driven only if BTOLCS1~ or BTOLCS2~ are asserted. Otherwise they are tristated. The direction of these lines is determined by buffered BCTL0, in function of R~/W.
C15	EXPD1		
C16	EXPD2		
C17	EXPD3		
C18	EXPD4		
C19	EXPD5		
C20	EXPD6		
C21	EXPD7		
C22	EXPD8		
C23	EXPD9		
C24	EXPD10		
C25	EXPD11		
C26	EXPD12		
C27	EXPD13		
C28	EXPD14		
C29	EXPD15		
C30	N.C.	-	Not Connected
C31			
C32			
D1	GND	O	Digital Ground. Connected to main GND plane of the ADS.
D2			
D3			
D4	EXPWE0~	O	Expansion Write Enable (0:1) (L). This are buffered GPCM Write Enable lines (0:1). They are meant to qualify writes to GPCM controlled 8/16 data bus width memory devices. This to provide eased access to various communication transceivers. EXPWE0~ controls EXPD(0:7) while EXPWE1~ controls EXPD(8:15). These lines may also function as UPM controlled Byte Select Lines, which allow control over almost any type of memory device.
D5	EXPWE1~		
D6	GND	O	Digital Ground. Connected to main GND plane of the ADS.

Table 5-7. P16 - System Expansion - Interconnect Signals

Pin No.	Signal Name	Attribute	Description
D7	EXPGL0~	O	Expansion General Purpose Lines (0:5) (L). These are buffered GPL(0:5)~ lines which assist UPM control over memory device if necessary. These are output only signals and therefore, do not support H/W controlled UPM waits.
D8	EXPGL1~		
D9	EXPGL2~		
D10	EXPGL3~		
D11	EXPGL4~		
D12	EXPGL5~		
D13	GND	O	Digital Ground. Connected to main GND plane of the ADS.
D14	EXPALE	O	Expansion Address Latch Enable (H). This a buffered MPC8260‘s ALE, provided for expansion board’s use.
D15	EXPCTL0	O	Expansion Control Line 0. This line is a buffered version of MPC8260’s BCTL0 (Bus Control Line 0) which serves as R~/W, provided for expansion board’s use.
D16	GND	O	Digital Ground. Connected to main GND plane of the ADS.
D17			
D18			
D19			
D20			
D21			
D22			
D23			
D24			
D25			
D26			
D27			
D28			
D29			
D30			
D31			
D32			

[1]MS Bit.

5.2 MPC8260ADS Part List

In this section the MPC8260ADS's bill of material is listed according to their reference designation.

The BOM for the MPC8260ADS is shown in Table 5-8 ". MPC8260ADS Bill Of Material" below, while

Table 5-8. MPC8260ADS Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
C1 C2 C102 C180 C193 C198 C199	Capacitor 68µF, 16V, 10%, SMD, Size D, Tantalum	AVX	TAJD686K016R
C3 C5 C6 C7 C8 C103 C141 C142 C147 C153 C154 C155 C158 C164 C165 C167 C171 C176 C178 C195 C196 C206	Capacitor 10µF, 20V, 10%, SMD Size C, Tantalum	SIEMENS	B45196-H4475-K20
C4 C9 C10 C11 C12 C13 C181 C183 C185 C186 C187 C189 C190 C191 C197	Capacitor 10nF, 50V, 10%, NPO, SMD 1210, Ceramic	VITRAMNON	VJ1210A103KXAT
C14 C15 C16 C20 C23 C24 C28 C29 C30 C31 C33 C34 C35 C36 C37 C38 C39 C40 C41 C42 C43 C44 C45 C46 C47 C48 C49 C50 C51 C52 C53 C54 C55 C56 C57 C58 C59 C60 C61 C62 C64 C65 C67 C69 C70 C71 C72 C73 C74 C75 C76 C77 C78 C79 C80 C84 C85 C86 C87 C88 C92 C99 C104 C105 C106 C107 C108 C109 C110 C111 C112 C113 C114 C115 C116 C117 C118 C119 C120 C122 C123 C125 C126 C127 C128 C129 C130 C131 C137 C138 C139 C140 C144 C145 C146 C151 C157 C159 C161 C162 C163 C166 C168 C169 C172 C173 C174 C177 C179 C182 C184 C188 C192 C194 C200 C201 C202 C203 C204 C205	Capacitor 0.1µF,16V, 10%, SMD 0603, Ceramic	AVX	0603YC104KAT20
C63 C68 C93 C100 C101 C121 C132 C134 C149 C160	Capacitor 47µF, 20V, 10%, SMD SIze D, Tantalum	AVX	TAJD476K016
C66	Capacitor 100µF, 10V, 10%, SMD Size D, Tantalum	AVX	TAJD107K016R
C135	Capacitor 1500pF, 50V, 5%, COG, SMD 1206 Ceramic	AVX	12065A152JAT00J
C148 C150 C156	Capacitor 22pF, 50V, 5%, COG, SMD 1206 Ceramic	AVX	1206 5A 220J AT
C152	Capacitor 1µF, 20V, 10%, SMD SIze B, Tantalum	SIEMENS	B45196H5105K109
C170 C175	Capacitor 10pF, 50V, 10%, SMD 1206, Ceramic	SIEMENS	

Table 5-8. MPC8260ADS Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
C207	Capacitor 0.001µF, 2 KV, 10%, SMD SIze 1210, Ceramic	AVX	1210B102K202NT
C208 C209	Capacitor 0.1µF, X7R, 500V, 20%, SMD SIze 1812, Ceramic	JOHANSON DIELECTRIC	501S43W104MV4E
D1	Zener Diode, 5V SMD	Motorola	1SMC5.0AT3
D2 D7	Diode Pair, common cathode	Motorola	MBRD620CT
D3 D4 D5	Diode SMD	Motorola	LL4004G
D6	Zener Diode, 12V SMD	Motorola	1SMC12AT3
DS1	Dip-Switch, 8 X SPST, SMD	GRAYHILL	90HBW08S
DS2 DS3	Dip-Switch, 4 X SPST, SMD	GRAYHILL	90HBW04S
F1	Fuse, 1A/250V Miniature 5 X 20mm, Fast-blow	Little Fuse	217001
F2	Fuse, 5A/250V Miniature 5 X 20mm, Fast-blow	Little Fuse	217005
H1 H2 H3 H4 H5	Gnd Bridge, Gold Plated	PRECIDIP	999-11-112-10
J1 J3 J5 J7	Jumper Header, 3 Pole with Fabricated Jumper		
J2 J6	Jumper, 2 Pole, Soldered, Gold Plated	PRECIDIP	999-11-112-10
L1 L2 L3 L4 L5 L6 L7	Ferrite Bid	Fair rite	2743021447
LD1 LD2 LD3 LD4 LD8 LD9 LD10 LD11	Led Green SMD	SIEMENS	LG T670-HK
LD5 LD6 LD13 LD14 LD15 LD16	Led Yellow SMD	SIEMENS	LY T670-HK
LD7 LD12	Led Red SMD	SIEMENS	LS T670-HK
P1	Connector 8 pin, RJ45 Receptacle, Shielded, 90^{o}	MOLEX	43202_8110
P2	Connector 2 pin, Power, Straight, with false insertion protection.	WB	8113S-253303253
	Connector 2 pin, Power Plug	WB	8113B-253200253
PA3, PB3	Connector 2 X 9 pin Stacked, Female, DType, 90^{o}	EDA Inc.	8LE 009 009 D 3 06H
P4 P16	Connector 128 pin, Female, DIN 41612, 90^{o}	ERNI	ERNI 043326
P5	Connector header, 16 pin, Male, T.H, 90^{O}, With Latches.	ALPHA	FCC-301-16
P6 P7 P8 P9 P10 P12 P13 P14 P15 P17 P18	Connector MICTOR[1] 38 pin, SMD	AMP	2-767004-2

Table 5-8. MPC8260ADS Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
P11	Connector header, 10 pin, dual in-line, SMD	SAMTEC	TSM-105-03-S-DV
P19	Connector 3 pin, Power, Straight, with false insertion protection.	WB	8113S-253303353
	Connector 3 pin, Power Plug	WB	8113B-253200353
R1 R2 R4 R124 R139 R140 R148 R149 R151 R152 R153	Resistor 51.1 Ω, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 51R1FCS
R3 R154	Resistor 75 Ω, 5%, SMD 0603, 0.1W	DRALORIK	D11 075RFCS
R5 R6 R133	Resistor 82.5 Ω, 5%, SMD 1206, 1/4W	DRALORIK	D25 82R5FCS
R7	Resistor 22.1 KΩ, 5%, SMD 1206, 1/4W	DRALORIK	D25-22K1FCS
R8 R39 R44 R49 R56 R57 R77 R88 R94 R95 R96 R100 R112 R121	Resistor 43.2 Ω, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 43R2FCS
R9 R10 R11 R25 R26 R60 R65 R90 R103 R107 R110 R113 R115 R116 R117 R119 R142	Resistor 0 Ω, SMD 0603, 0.1W	RODERSTEIN	D11 000RFCS
R13 R14 R15 R16 R17 R29 R31 R33 R46 R48 R53 R58 R59 R66 R67	Resistor 10 KΩ, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 010KFCS
R18 R19 R122 R125 R128 R129 R134	Resistor 4.7 KΩ, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 04K7FCS
R20 R22 R27 R28 R36 R37 R38 R71	Resistor 150 Ω, 5% SMD 1206, 1/4W	RODERSTEIN	D25-150RFCS
R21 R23 R24	Resistor 1.5 KΩ, 5%, SMD 1206, 1/4W	RODERSTEIN	D25 01K5FCS
R32 R40 R41 R42 R43 R73 R76 R78 R84	Resistor 22 Ω, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 22R0FCS
R63 R79	Resistor 10 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 10R FCS
		AVX	CR32-10ROF-T
R64	Resistor 2.2 KΩ, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 02K2F CS
R68 R70 R80 R81 R86 R87 R92 R93 R97 R99 R101 R102 R109 R136	Resistor 1 KΩ, 5%, SMD 0603, 0.1W	DRALORIK	D11 001KFCS
R69	Resistor 47 KΩ, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 047KFCS
R75	Resistor 0 Ω, SMD 1206, 1/4W	RODERSTEIN	D25 000RFCS
R82	Resistor 0.5 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 0R50F CS

Table 5-8. MPC8260ADS Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
R83	Resistor 51 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25-051J-S
R91	Resistor 243 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 243RF CS
R98	Resistor 110 Ω, 1%, SMD 1206, 1/4W	AVX	CR32-111J-T
R104 R106 R108 R111 R155 R156 R157 R159 R162	Resistor 330 Ω, 5%, SMD 1206, 1/4W	RODERSTEIN	D25 330RJCS
R114 R118 R120	Resistor 1 kΩ, 1%, SMD 1206, 1/4W	AVX	CR32-102F-T
		DRALORIK	D25001KFCS
R126	Resistor 2.2 MΩ, 1%, SMD 1206, 1/4W	RODERSTEIN	D2502M2FCS
R127	Resistor 1.5 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 01R5FCS
R130 R143 R144	Resistor 133 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25 133RFCS
R137 R145 R146 R150 R160	Resistor 2.7 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 2R74FCS
R138	Resistor 100 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25-100RFCS
R147	Resistor 63.4 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25 63R4FCS
R158 R161	Resistor 270 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25 270RFCS
RN1 RN2 RN6 RN8 RN9 RN10 RN11 RN12 RN13 RN21 RN22 RN28 RN30 RN31 RN32 RN33 RN34 RN35 RN40 RN41 RN42 RN43 RN44 RN51 RN52 RN53 RN56 RN57 RN67 RN68 RN74 RN88	Resistor Network 10 KΩ, 5%, 8 resistors, 10 pin, Common Bus	ROHM	RS8A 1002 J
RN3 RN4 RN5 RN7 RN20 RN27 RN80 RN81 RN82 RN83 RN84 RN85 RN86 RN87	Resistor Network 43 Ω, 5%, 4 resistors, 8 pin.	DALE	CRA06S 08 03 430JRT
RN14 RN16 RN17 RN19 RN24 RN29 RN36 RN37 RN38 RN39 RN45 RN46 RN47 RN48 RN49 RN50 RN54 RN55 RN58 RN59 RN60 RN61 RN62 RN63 RN64 RN65 RN66 RN69 RN70 RN71 RN72 RN73 RN75 RN76 RN77 RN78 RN79	Resistor Network 22 Ω, 5%, 4 resistors, 8 pin.	DALE	CRA06S0803220JR
RN15 RN18 RN23	Resistor Network 0 Ω, 4 resistors, 8 pin.	DALE	CRA06S0803 000 RT

Table 5-8. MPC8260ADS Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
SW1	SPDT, push button, RED, Sealed	C & K	KS12R22-CQE
SW2	SPDT, push button, WHITE, Sealed	C & K	KS12R21-CQE
SW3	SPDT, push button, BLACK, Sealed	C & K	KS12R23-CQE
T1	Transistor, Dual, TMOS, VT 2V	Motorola	MMDF3N02HD
TR1	Trimmer Pot. 1 KΩ, Single Turn	BOURNS	3362P-1-102
U1	Fiber Optic I/F Module, 1300 nm wavelength, 2 Km Range	HP	HFBR 5205
U2 U5	3.3V Powered, Single Supply, RS232 Transceiver (3 Tx, 5 Rx).	Motorola	MC145583V
U3	10/100 Base-T Filter network	HALO	TG22-3506ND
U4	Saturn User Network I/F (S/UNI) for 155.52 & 51.84 Mbps, 128 pin PQFP	PMC-Sierra Inc.	PM5350-RC
U6	Clock Generator, 19.44 MHz, 20 ppm, 5V Supply, 4 pins, (8 pin DIP form factor)	COMCLOK	CM42AH-19.440
U7 U9 U23 U37	Quad CMOS buffer with individual Output Enable.	Motorola	74LCX125P
U8 U20 U21	Octal CMOS Buffer.	Motorola	74LCX541J
U10	Voltage Regulator, Variable, 1.5 A output, D^2PAK package	Motorola	LM317D2T
U11	3.3V Linear Voltage regulator. 5A output.	Micrel	MIC29500-3.3BT
U12	MPC8260, 2’nd generation Power QUICC.	Motorola	MPC8260
	Socket for the above, 480 pin, 1.27mm, PGA to PGA, 29 X 29 array, gold plated contacts.	ANDON	0-29-05-480-286-G04-N10
	Adaptor, PGA to BGA, 480 pin, 1.27mm, 29 X 29 array, gold plated contacts.		0-29-05-480-27KP30-N10
	Socket for the above, 480 pin, 1.27mm, PGA to BGA, 29 X 29 array, gold plated contacts, screw-lock.	E-Tec	BPW480-1270-29AD01
U13 U14	SDRAM, 2 Banks X 512KBytes X 16 Bit, 100MHz	Fujitsu	MB811171622A-100
U16	Clock Generator, 66MHz or 40 MHz, 3.3V Supply, 4 pins, (8 pin DIP form factor)	M-TRON	M3H16FCD

Table 5-8. MPC8260ADS Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
U17	M4-128/64 - 64 I/O, 128 Macrocell, 7.5 nsec propagation delay, In System Programmable Logic Device, 100 pin TQPF	AMD	M4-128/64-7VC
U22	16 MByte, 168 pin, 100MHz, SDRAM DIMM organized as 2 X 1 M X 64.	Fujitsu	SDC2UV6482D-100T-S or PDC2UV6484-(102/103/10)T-S
	168 pin, 90°, T.H., DIMM Socket, with Plastic Latches, 3.3V Unbuffered	AMP	390040-6
U24 U29 U30	Low Voltage, CMOS, 5V Tolerant 16 bit buffer, with OEs. 48 pin Plastic TSSOP, Case 1201-01	Motorola	MC74LCX16244DT
U25	8 MByte Flash SIMM, 95 nsec delay, Single bank, composed of four LH28F016SCT-L95 chips by SHARP.	Smart Modular Technology / SHARP	SM73228XG1JHBG0
	80 pin SIMM Socket	AMP	822032-5
U26	Low Voltage, CMOS, 5V Tolerant 16 bit Transceiver. 48 pin Plastic TSSOP, Case 1201-01	Motorola	MC74LCX16245DT
U27 U28	Low Voltage, CMOS, 5V Tolerant 16 bit Transceiver, with Bus-Hold. 48 pin Plastic TSSOP.	Philips	74ALVT16245DL
U27 U28	Low Voltage, CMOS, 5V Tolerant 16 bit Transparent Latch, with Bus-Hold. 48 pin Plastic TSSOP.	Philips	74ALVT16373DL
U33	Octal Tri-State Buffer.	Motorola	74ACT541D
U35	9 Output, Low inter-skew, Clock Buffer.	Motorola	MPC947
U36	Voltage level detector. Range 2.8V ±2%. O.D. output. SOT-23-5 package.	Seiko	S-80828ANMP-EDR-T2
U38	Fast Ethernet Transceiver.	Level 1	LXT970A
Y1	Crystal resonator, 25 MHz, Fundamental Oscillation mode, Frequency tolerance ±30 ppm, Drive-level - 2mW max 10µW - 100 µW recommended, Shunt capacitance - 5pF Max., Load capacitance - 10pF min, Equivalent Series Resistance - 40Ω Max. Insulation Resistance - 500 MΩ min.	EPSON	MA-505

[1]Matched Impedance Connector.

the BOM for the MPC8260ADSL2C - with Level II Cache is presented in Table 5-9 ". MPC8260ADSL2C

Bill Of Material" on page 101.

Table 5-9. MPC8260ADSL2C Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
C1 C2 C102 C180 C193 C198 C199	Capacitor 68µF, 16V, 10%, SMD, Size D, Tantalum	AVX	TAJD686K016R
C3 C5 C6 C7 C8 C103 C141 C142 C147 C153 C154 C155 C158 C164 C165 C167 C171 C176 C178 C195 C196 C206	Capacitor 10µF, 20V, 10%, SMD Size C, Tantalum	SIEMENS	B45196-H4475-K20
C4 C9 C10 C11 C12 C13 C181 C183 C185 C186 C187 C189 C190 C191 C197	Capacitor 10nF, 50V, 10%, NPO, SMD 1210, Ceramic	VITRAMNON	VJ1210A103KXAT
C14 C15 C16 C17 C18 C19 C20 C21 C22 C23 C24 C25 C26 C27 C28 C29 C30 C31 C32 C33 C34 C35 C36 C37 C38 C39 C40 C41 C42 C43 C44 C45 C46 C47 C48 C49 C50 C51 C52 C53 C54 C55 C56 C57 C58 C59 C60 C61 C62 C64 C65 C67 C69 C70 C71 C72 C73 C74 C75 C76 C77 C78 C79 C80 C81 C82 C83 C84 C85 C86 C87 C88 C89 C90 C91 C92 C94 C95 C96 C97 C98 C99 C104 C105 C106 C107 C108 C109 C110 C111 C112 C113 C114 C115 C116 C117 C118 C119 C120 C122 C123 C125 C126 C127 C128 C129 C130 C131 C137 C138 C139 C140 C144 C145 C146 C151 C157 C159 C161 C162 C163 C166 C168 C169 C172 C173 C174 C177 C179 C182 C184 C188 C192 C194 C200 C201 C202 C203 C204 C205	Capacitor 0.1µF,16V, 10%, SMD 0603, Ceramic	AVX	0603YC104KAT20
C63 C68 C93 C100 C101 C121 C132 C134 C149 C160	Capacitor 47µF, 20V, 10%, SMD SIze D, Tantalum	AVX	TAJD476K016
C66	Capacitor 100µF, 10V, 10%, SMD Size D, Tantalum	AVX	TAJD107K016R
C135	Capacitor 1500pF, 50V, 5%, COG, SMD 1206 Ceramic	AVX	12065A152JAT00J
C148 C150 C156	Capacitor 22pF, 50V, 5%, COG, SMD 1206 Ceramic	AVX	1206 5A 220J AT
C152	Capacitor 1µF, 20V, 10%, SMD SIze B, Tantalum	SIEMENS	B45196H5105K109
C170 C175	Capacitor 10pF, 50V, 10%, SMD 1206, Ceramic	AVX	1206 5A 100J AT

Table 5-9. MPC8260ADSL2C Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
C207	Capacitor 0.001µF, 2 KV, 10%, SMD SIze 1210, Ceramic	AVX	1210B102K202NT
C208 C209	Capacitor 0.1µF, X7R, 500V, 20%, SMD SIze 1812, Ceramic	JOHANSON DIELECTRIC	501S43W104MV4E
D1	Zener Diode, 5V SMD	Motorola	1SMC5.0AT3
D2 D7	Diode Pair, common cathode	Motorola	MBRD620CT
D3 D4 D5	Diode SMD	Motorola	LL4004G
D6	Zener Diode, 12V SMD	Motorola	1SMC12AT3
DS1	Dip-Switch, 8 X SPST, SMD	GRAYHILL	90HBW08S
DS2 DS3	Dip-Switch, 4 X SPST, SMD	GRAYHILL	90HBW04S
F1	Fuse, 1A/250V Miniature 5 X 20mm, Fast-blow	Little Fuse	217001
F2	Fuse, 5A/250V Miniature 5 X 20mm, Fast-blow	Little Fuse	217005
H1 H2 H3 H4 H5	Gnd Bridge, Gold Plated	PRECIDIP	999-11-112-10
J1 J3 J5 J7	Jumper Header, 3 Pole with Fabricated Jumper	-	-
J2 J6	Jumper, 2 Pole, Soldered, Gold Plated	PRECIDIP	999-11-112-10
L1 L2 L3 L4 L5 L6 L7	Ferrite Bid	Fair rite	2743021447
LD1 LD2 LD3 LD4 LD8 LD9 LD10 LD11	Led Green SMD	SIEMENS	LG T670-HK
LD5 LD6 LD13 LD14 LD15 LD16	Led Yellow SMD	SIEMENS	LY T670-HK
LD7 LD12	Led Red SMD	SIEMENS	LS T670-HK
P1	Connector 8 pin, RJ45 Receptacle, Shielded, 90^{o}	MOLEX	43202_8110
P2	Connector 2 pin, Power, Straight, with false insertion protection.	WB	8113S-253303253
	Connector 2 pin, Power Plug	WB	8113B-253200253
PA3, PB3	Connector 2 X 9 pin Stacked, Female, DType, 90^{o}	EDA Inc.	8LE 009 009 D 3 06H
P4 P16	Connector 128 pin, Female, DIN 41612, 90^{o}	ERNI	ERNI 043326
P5	Connector header, 16 pin, Male, T.H, 90^{O}, With Latches.	ALPHA	FCC-301-16
P6 P7 P8 P9 P10 P12 P13 P14 P15 P17 P18	Connector MICTOR[1] 38 pin, SMD	AMP	2-767004-2

Table 5-9. MPC8260ADSL2C Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
P11	Connector header, 10 pin, dual in-line, SMD	SAMTEC	TSM-105-03-S-DV
P19	Connector 3 pin, Power, Straight, with false insertion protection.	WB	8113S-253303353
	Connector 3 pin, Power Plug	WB	8113B-253200353
R1 R2 R4 R124 R139 R140 R148 R149 R151 R152 R153	Resistor 51.1 Ω, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 51R1FCS
R3 R154	Resistor 75 Ω, 5%, SMD 0603, 0.1W	DRALORIK	D11 075RFCS
R5 R6 R133	Resistor 82.5 Ω, 5%, SMD 1206, 1/4W	DRALORIK	D25 82R5FCS
R7	Resistor 22.1 KΩ, 5%, SMD 1206, 1/4W	DRALORIK	D25-22K1FCS
R8 R39 R44 R49 R56 R57 R77 R88 R94 R95 R96 R100 R112 R121	Resistor 43.2 Ω, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 43R2FCS
R9 R10 R11 R34 R50 R52 R54 R55 R60 R62 R65 R90 R103 R107 R110 R113 R115 R116 R117 R119 R142	Resistor 0 Ω, SMD 0603, 0.1W	RODERSTEIN	D11 000RFCS
R13 R14 R15 R16 R17 R29 R31 R33 R35 R46 R47 R48 R53 R58 R59 R61 R66 R67	Resistor 10 KΩ, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 010KFCS
R18 R19 R122 R125 R128 R129 R134	Resistor 4.7 KΩ, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 04K7FCS
R20 R22 R27 R28 R36 R37 R38 R71	Resistor 150 Ω, 5% SMD 1206, 1/4W	RODERSTEIN	D25-150RFCS
R21 R23 R24	Resistor 1.5 KΩ, 5%, SMD 1206, 1/4W	RODERSTEIN	D25 01K5FCS
R32 R40 R41 R42 R43 R73 R76 R78 R84	Resistor 22 Ω, 1%, SMD 0603, 0.1W	RODERSTEIN	D11 22R0FCS
R63 R79	Resistor 10 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 10R FCS
		AVX	CR32-10ROF-T
R64	Resistor 2.2 KΩ, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 02K2F CS
R68 R70 R80 R81 R86 R87 R92 R93 R97 R99 R101 R102 R109 R136	Resistor 1 KΩ, 5%, SMD 0603, 0.1W	DRALORIK	D11 001KFCS
R69	Resistor 47 KΩ, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 047KFCS
R75	Resistor 0 Ω, SMD 1206, 1/4W	RODERSTEIN	D25 000RFCS

Table 5-9. MPC8260ADSL2C Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
R82	Resistor 0.5 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 0R50F CS
R83	Resistor 51 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25-051J-S
R91	Resistor 243 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 243RF CS
R98	Resistor 110 Ω, 1%, SMD 1206, 1/4W	AVX	CR32-111J-T
R104 R106 R108 R111 R155 R156 R157 R159 R162	Resistor 330 Ω, 5%, SMD 1206, 1/4W	RODERSTEIN	D25 330RJCS
R114 R118 R120	Resistor 1 kΩ, 1%, SMD 1206, 1/4W	AVX	CR32-102F-T
		DRALORIK	D25001KFCS
R126	Resistor 2.2 MΩ, 1%, SMD 1206, 1/4W	RODERSTEIN	D2502M2FCS
R127	Resistor 1.5 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 01R5FCS
R130 R143 R144	Resistor 133 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25 133RFCS
R137 R145 R146 R150 R160	Resistor 2.7 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25 2R74FCS
R138	Resistor 100 Ω, 1%, SMD 1206, 1/4W	RODERSTEIN	D25-100RFCS
R147	Resistor 63.4 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25 63R4FCS
R158 R161	Resistor 270 Ω, 1%, SMD 1206, 1/4W	DRALORIK	D25 270RFCS
RN1 RN2 RN6 RN8 RN9 RN10 RN11 RN12 RN13 RN21 RN22 RN25 RN26 RN28 RN30 RN31 RN32 RN33 RN34 RN35 RN40 RN41 RN42 RN43 RN44 RN51 RN52 RN53 RN56 RN57 RN67 RN68 RN74 RN88	Resistor Network 10 KΩ, 5%, 8 resistors, 10 pin, Common Bus	ROHM	RS8A 1002 J
RN3 RN4 RN5 RN7 RN20 RN27 RN80 RN81 RN82 RN83 RN84 RN85 RN86 RN87	Resistor Network 43 Ω, 5%, 4 resistors, 8 pin.	DALE	CRA06S 08 03 430JRT
RN14 RN16 RN17 RN19 RN24 RN29 RN36 RN37 RN38 RN39 RN45 RN46 RN47 RN48 RN49 RN50 RN54 RN55 RN58 RN59 RN60 RN61 RN62 RN63 RN64 RN65 RN66 RN69 RN70 RN71 RN72 RN73 RN75 RN76 RN77 RN78 RN79	Resistor Network 22 Ω, 5%, 4 resistors, 8 pin.	DALE	CRA06S0803220JR

Table 5-9. MPC8260ADSL2C Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
SW1	SPDT, push button, RED, Sealed	C & K	KS12R22-CQE
SW2	SPDT, push button, WHITE, Sealed	C & K	KS12R21-CQE
SW3	SPDT, push button, BLACK, Sealed	C & K	KS12R23-CQE
T1	Transistor, Dual, TMOS, VT 2V	Motorola	MMDF3N02HD
TR1	Trimmer Pot. 1 KΩ, Single Turn	BOURNS	3362P-1-102
U1	Fiber Optic I/F Module, 1300 nm wavelength, 2 Km Range	HP	HFBR 5205
U2 U5	3.3V Powered, Single Supply, RS232 Transceiver (3 Tx, 5 Rx).	Motorola	MC145583V
U3	10/100 Base-T Filter network	HALO	TG22-3506ND
U4	Saturn User Network I/F (S/UNI) for 155.52 & 51.84 Mbps, 128 pin PQFP	PMC-Sierra Inc.	PM5350-RC
U6	Clock Generator, 19.44 MHz, 20 ppm, 5V Supply, 4 pins, (8 pin DIP form factor)	COMCLOK	CM42AH-19.440
U7 U9 U23 U37	Quad CMOS buffer with individual Output Enable.	Motorola	74LCX125P
U8 U20 U21	Octal CMOS Buffer.	Motorola	74LCX541J
U10	Voltage Regulator, Variable, 1.5 A output, D^2PAK package	Motorola	LM317D2T
U11	3.3V Linear Voltage regulator. 5A output.	Micrel	MIC29500-3.3BT
U12	MPC8260, 2’nd generation Power QUICC.	Motorola	MPC8260
	Socket for the above, 480 pin, 1.27mm, PGA to PGA, 29 X 29 array, gold plated contacts.	ANDON	0-29-05-480-286-G04-N10
	Adaptor, PGA to BGA, 480 pin, 1.27mm, 29 X 29 array, gold plated contacts.		0-29-05-480-27KP30-N10
	Socket for the above, 480 pin, 1.27mm, PGA to BGA, 29 X 29 array, gold plated contacts, screw-lock.	E-Tec	BPW480-1270-29AD01
U13 U14	SDRAM, 2 Banks X 512KBytes X 16 Bit, 100MHz	Fujitsu	MB811171622A-100
U15 U34	Secondary Cache, 256KByte, Look Aside, for PowerPC Microprocessors.	Motorola	MPC2605ZP66

Table 5-9. MPC8260ADSL2C Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
U16	Clock Generator, 66MHz or 40 MHz, 3.3V Supply, 4 pins, (8 pin DIP form factor)	M-TRON	M3H16FCD
U17	M4-128/64 - 64 I/O, 128 Macrocell, 7.5 nsec propagation delay, In System Programmable Logic Device, 100 pin TQPF	Vantis	M4-128/64-7VC
U18 U19	M4-64/32 - 32 I/O, 64 Macrocell, 7.5 nsec propagation delay, In System Programmable Logic device, 48 pin TQFP.	Vantis	M4-64/32-7VC48
U22	16 MByte, 168 pin, 100MHz, SDRAM DIMM organized as 2X 1 M X 64.	Fujitsu	SDC2UV6482D-100T-S or PDC2UV6484-(102/103/10)T-S
	168 pin, 90^{o}, T.H., DIMM Socket, with Plastic Latches, 3.3V Unbuffered	AMP	390040-6
U24 U29 U30	Low Voltage, CMOS, 5V Tolerant 16 bit buffer, with OEs. 48 pin Plastic TSSOP, Case 1201-01	Motorola	MC74LCX16244DT
U25	8 MByte Flash SIMM, 95 nsec delay, Single bank, composed of four LH28F016SCT-L95 chips by SHARP.	Smart Modular Technology / SHARP	SM73228XG1JHBG0
	80 pin SIMM Socket	AMP	822032-5
U26	Low Voltage, CMOS, 5V Tolerant 16 bit Transceiver. 48 pin Plastic TSSOP, Case 1201-01	Motorola	MC74LCX16245DT
U27 U28	Low Voltage, CMOS, 5V Tolerant 16 bit Transceiver, with Bus-Hold. 48 pin Plastic TSSOP.	Philips	74ALVT16245DL
U27 U28	Low Voltage, CMOS, 5V Tolerant 16 bit Transparent Latch, with Bus-Hold. 48 pin Plastic TSSOP.	Philips	74ALVT16373DL
U33	Octal Tri-State Buffer.	Motorola	74ACT541D
U35	9 Output, Low inter-skew, Clock Buffer.	Motorola	MPC947
U36	Voltage level detector. Range 2.8V ±2%. O.D. output. SOT-23-5 package.	Seiko	S-80828ANMP-EDR-T2
U38	Fast Ethernet Transceiver.	Level 1	LXT970A

Table 5-9. MPC8260ADSL2C Bill Of Material

Reference Designation	Part Description	Manufacturer	Part #
Y1	Crystal resonator, 25 MHz, Fundamental Oscillation mode, Frequency tolerance ±30 ppm, Drive-level - 2mW max 10μW - 100 μW recommended, Shunt capacitance - 5pF Max., Load capacitance - 10pF min, Equivalent Series Resistance - 40Ω Max. Insulation Resistance - 500 MΩ min.	EPSON	MA-505

[1]Matched Impedance Connector.

5.3 Programmable Logic Equations

The MPC8260ADS has one programmable logic device on it - U17, serving the role of Board Control and Status Register and providing miscellaneous system control functions on the ADS. Implemented using an M4-128/64-7VC by Vantis.

The MPC8260ADSL2C (the same ADS but with L2 Cache on it) has an additional 2 programmable logic devices U18 and U19, serving as Latch-Mux for the SDRAM DIMM. Implemented using an M4-64/32-7VC48 by Vantis. The design files for the above devices, are written in ABEL format and are listed below.

5.3.1 U17 - BCSR & System Control

```
MODULE vbcsr12
"*****************************************************************************
"* Pins declaration.
"*****************************************************************************
"* System i/f pins
"*****************************************************************************
SYSCLK              PIN  11 ;
BrdContRegCs_B      PIN  36 ;
DVal_B              PIN  43 ;
R_B_W               PIN  86 ;
BCTL1               PIN  64 ;

A7                  PIN  98 ; " for flash support
A8                  PIN   4 ; " for flash support
A27                 PIN   5 ;
A28                 PIN   7 ;
A29                 PIN   8 ;

D0                  PIN  84   istype 'com' ;
D1                  PIN  78   istype 'com' ;
D2                  PIN  80   istype 'com' ;
D3                  PIN  59   istype 'com' ;
D4                  PIN  70   istype 'com' ;
D5                  PIN  57   istype 'com' ;
D6                  PIN  60   istype 'com' ;
D7                  PIN  72   istype 'com' ;

"*****************************************************************************
"* Board Control Pins. Read/Write.
"*****************************************************************************
PBI                 PIN  44  istype 'reg,buffer' ; " Page Base Interleaving
DimmSize            PIN  53  istype 'reg,buffer' ; " Sdram Dimm Size
L2Inh_B             PIN  15  istype 'reg,buffer' ; " flash enable.
L2Flush_B           PIN  54  istype 'reg,buffer' ; " 60x bus sdram enable
L2Lock_B            PIN  35  istype 'reg,buffer' ; " bursting sram enable
L2Clear_B           PIN  34  istype 'reg,buffer' ; " local bus sdram enable
SignaLamp0_B        PIN  32  istype 'reg,buffer' ; " status lamp 0 for misc s/w visual
SignaLamp1_B        PIN  33  istype 'reg,buffer' ; " status lamp 1 for misc s/w visual
                             " signaling
AtmEn_B             PIN  94  istype 'reg,buffer' ; " atm uni enable
```

```
AtmRst_B NODE    istype 'reg,buffer' ; " atm uni reset bit
AtmRstOut_B          PIN  96  istype 'com' ; " atm uni reset driven by register
                                " or by HRESET_B
FEthEn_B             PIN  95  istype 'reg,buffer' ; " fast ethernet trans. enable
FEthRst_B       NODE     istype 'reg,buffer' ; " fast ethernet trans. reset bit
FEthRstOut_B         PIN   3  istype 'com' ; " fast eth trans reset driven by
                                " register or by HRESET_B
RS232En1_B           PIN  92  istype 'reg,buffer' ; " RS232 port 1 enable
RS232En2_B           PIN  93  istype 'reg,buffer' ; " RS232 port 2 enable
"*****************************************************************************
"* Board Status Registers Chip-Selects
"*****************************************************************************
Bcsr2Cs_B            PIN  42  istype 'com' ;
"*****************************************************************************
"* Flash Associated Pins.
"*****************************************************************************
F_PD1                PIN  71 ;
F_PD2                PIN  69 ;
F_PD3                PIN  67 ;
F_PD4                PIN  65 ;

FlashCs_B            PIN  31 ; " flash bank chip-select

FlashCs1_B           PIN  58  istype 'com' ; " Flash bank1 chip-select
FlashCs2_B           PIN  48  istype 'com' ; " Flash bank2 chip-select
FlashCs3_B           PIN  45  istype 'com' ; " Flash bank3 chip-select
FlashCs4_B           PIN  55  istype 'com' ; " Flash bank4 chip-select
"*****************************************************************************
"* PMI5346 ATM UNI Associated Pins.
"*****************************************************************************
AtmUniCsIn_B         PIN  22 ;
AtmUniCsOut_B        PIN  21  istype 'com' ; " remove if short of pins
"*****************************************************************************
"* Reset & Interrupt Logic Pins.
"*****************************************************************************
PORIn_B              PIN  97 ;
RstConf_B            PIN  28  istype 'com'; " Hard Reset master select.

Rst0                 PIN  20 ; " connected to N.C. of Reset P.B.
Rst1                 PIN  19 ; " connected to N.O. of Reset P.B.

HardReset_B          PIN  16  istype 'com' ; " Actual hard reset output (O.D.)
```

```
SoftReset_B          PIN  29  istype 'com' ; " Actual soft reset output (O.D.)

Abr0                 PIN  18 ; " connected to N.C. of Abort P.B.
Abr1                 PIN  17 ; " connected to N.O. of Abort P.B.

NMIEn           NODE     istype 'com' ; " enables T.S. NMI pin
NMI_B                PIN  41  istype 'com' ; " Actual NMI pin (O,O.D.)

"*****************************************************************************
"* Power On Reset Configuration Support
"*****************************************************************************
MODCKH0              PIN  81 ;
MODCKH1              PIN  85 ;
MODCKH2              PIN  79 ;
MODCKH3              PIN  66 ;
"*****************************************************************************
"* Data Buffers Enables and Reset configuration support
"*****************************************************************************
TEA_B                PIN  61 ; "  Transfer Error Acknowledge.

DataBufEn_B          PIN   9  istype 'com,invert' ; " data buffer enable

ToolCs1_B            PIN  30 ; " comm tool cs line 1.
ToolCs2_B            PIN  91 ; " comm tool cs line 2.

ToolDataBufEn_B      PIN   6  istype 'com,invert' ; " tool data buffer enable

"*****************************************************************************
"* Hard Reset Configuration Logic
"*****************************************************************************
FlashConfEn_B        PIN  10 ; " enable signal from external switch

"*****************************************************************************
"* JTAG Logic
"*****************************************************************************
JtagEn               NODE istype 'reg,buffer';
Tdi                 PIN  56 ;
Tdo                  NODE istype 'reg,buffer' ; " clocked by falling Tck
TdoOut               PIN  82   istype 'com' ;
Tck                 PIN  14 ;
Tms                  PIN  47 ;
Trst_B               PIN  46 ;
```

```
PonResetOut             PIN  68  istype 'com';   " active high
"*******************************************************************************
"* Auxiliary Pins.
"*******************************************************************************

"*******************************************************************************
"*   Internal Signals
"*******************************************************************************
"* System Hard Reset Configuration.
"*******************************************************************************
DataOeNODE istype 'com' ;" data bus output enable on read.
"*******************************************************************************
"* Control Register Write (space saving) Mach 7 required for 66Mhz
"*******************************************************************************
" Bcsr0Write_B NODE istype 'com' ;
" Bcsr1Write_B NODE istype 'com' ;
"*******************************************************************************
"* Reset & Interrupt Logic Pins.
"*******************************************************************************
RstDeb1  NODE istype 'keep,com' ;    " reset push button debouncer
AbrDeb1  NODE istype 'keep,com' ;    " abort push button debouncer

HardResetEnNODE istype 'com' ;    " enables T.S. hard reset pin
SoftResetEnNODE istype 'com' ;    " enables T.S. soft reset pin
"*******************************************************************************
"* data buffers enable.
"*******************************************************************************
SyncHardReset_B  NODE istype 'reg,buffer' ;" synchronized hard reset
DSyncHardReset_B  NODE istype 'reg,buffer' ;" double synchronized hard reset

HoldOffCnt1,
HoldOffCnt0    NODE istype 'reg,buffer' ;       " data buf en hold-off counter

HoldOffTc      NODE istype 'com' ;              " terminal count for that counter
"*******************************************************************************
"* Power On Reset
"*******************************************************************************
S_PORIn_B      NODE  istype 'reg,buffer' ;    " synced pon reset.
"*******************************************************************************
"* JTAG Logic
"*******************************************************************************
```

```
JtagShiftDR0      NODE istype 'reg,buffer' ;    " Jtag shift data register
JtagShiftDR1      NODE istype 'reg,buffer' ;    "
JtagShiftDR2      NODE istype 'reg,buffer' ;    "
JtagShiftDR3      NODE istype 'reg,buffer' ;    "
JtagShiftDR4      NODE istype 'reg,buffer' ;    "
JtagShiftDR5      NODE istype 'reg,buffer' ;    "
JtagShiftDR6      NODE istype 'reg,buffer' ;    "
JtagShiftDR7      NODE istype 'reg,buffer' ;    "

JtagState0    NODE istype 'reg,buffer' ;    " Jtag state machine
JtagState1    NODE istype 'reg,buffer' ;    "
JtagState2    NODE istype 'reg,buffer' ;    "
JtagState3    NODE istype 'reg,buffer' ;    "

JtagResetState     NODE istype 'reg,buffer' ;    " sampling state on falling TCK
JtagShiftIrState   NODE istype 'reg,buffer' ;    " sampling state on falling TCK
JtagShiftDrState   NODE istype 'reg,buffer' ;    " sampling state on falling TCK
JtagTdoEnable      NODE istype 'reg,buffer' ;    " sampling state on falling TCK

JtagShiftIR0  NODE istype 'reg,buffer' ;    " Jtag Inst. Shift register
JtagShiftIR1  NODE istype 'reg,buffer' ;    "
JtagShiftIR2  NODE istype 'reg,buffer' ;    "

JtagIR0      NODE istype 'reg,buffer' ;    " Jtag Inst. register
JtagIR1      NODE istype 'reg,buffer' ;    "
JtagIR2      NODE istype 'reg,buffer' ;    "

JtagReceiveFull      NODE istype 'reg,buffer' ;   " indicates receive shift reg.
                                  " ready for read by memory cont
JtagReceiveFullReset NODE istype 'com' ;          " resets the receive full flag
SReadJtagDownloadData NODE istype 'reg,buffer';

JtagStateReset      NODE istype 'com' ;  " jtag state machine only
JtagReset           NODE istype 'com' ;  " global reset

TdoEnable           NODE istype 'com';   " enables the muxed TdoOut.
"*****************************************************************************
"* Misceleneous.
"*****************************************************************************
KeepPinsConnected NODE istype 'com' ;
"*****************************************************************************
"* Constant Declaration
```

```
"*****************************************************************************
H, L, X, Z = 1, 0, .X., .Z. ;
C, D, U    = .C., .D., .U. ;
"*****************************************************************************
"* Signal groups
"*****************************************************************************
JTAG = 1;
Add = [A27..A29] ;
Data = [D0..D7] ;
ContReg = [PBI,
        DimmSize,
        L2Inh_B,
        L2Flush_B,
        L2Lock_B,
        L2Clear_B,
        SignaLamp0_B,
        SignaLamp1_B,
        AtmEn_B,
        AtmRst_B,
        FEthEn_B,
        FEthRst_B,
        RS232En1_B,
        RS232En2_B,
        JtagEn.fb] ;
ReadBcsr0 = [PBI,
         DimmSize,
         L2Inh_B,
         L2Flush_B,
         L2Lock_B,
         L2Clear_B,
         SignaLamp0_B,
         SignaLamp1_B] ;
ReadBcsr1 = [0,
         0,
         AtmEn_B,
         AtmRst_B.fb,
         FEthEn_B,
         FEthRst_B.fb,
         RS232En1_B,
         RS232En2_B] ;
ReadBcsr3 = [0,
         0,
```

```
            0,
            0,
            0,
            0,
            0,
            JtagEn.fb];
DrivenContReg = [PBI,
                DimmSize,
                L2Inh_B,
                L2Flush_B,
                L2Lock_B,
                L2Clear_B,
                SignaLamp0_B,
                SignaLamp1_B,
                AtmEn_B,
                FEthEn_B,
                RS232En1_B,
                RS232En2_B] ;
ClockedContReg = [PBI,
                 DimmSize,
                 L2Inh_B,
                 L2Flush_B,
                 L2Lock_B,
                 L2Clear_B,
                 SignaLamp0_B,
                 SignaLamp1_B,
                 AtmEn_B,
                 AtmRst_B,
                 FEthEn_B,
                 FEthRst_B,
                 RS232En1_B,
                 RS232En2_B,
                 JtagEn] ;
Bcsr0 = [PBI,          " for simulation
        DimmSize,
        L2Inh_B,
        L2Flush_B,
        L2Lock_B,
        L2Clear_B,
        SignaLamp0_B,
        SignaLamp1_B] ;
Bcsr1 = [AtmEn_B," for simulation
```

```
        AtmRst_B,
        FEthEn_B,
        FEthRst_B,
        RS232En1_B,
        RS232En2_B] ;
Bcsr6 = [JtagEn];   " for simulation
ToolCs =        [ToolCs1_B,ToolCs2_B] ;
FlashCsOut = [FlashCs4_B,FlashCs3_B,FlashCs2_B,FlashCs1_B] ;
Reset = [HardReset_B,SoftReset_B] ;
ResetEn = [HardResetEn,SoftResetEn] ;
TransRst = [AtmRstOut_B,FEthRstOut_B] ;
Rst =       [Rst1,Rst0] ;
Abr =       [Abr1,Abr0] ;
Debounce = [RstDeb1,AbrDeb1] ;
SyncReset = [SyncHardReset_B,DSyncHardReset_B] ;
RstCause = [PORIn_B,Rst1,Rst0,Abr1,Abr0] ;
HoldOffCnt =[HoldOffCnt1, HoldOffCnt0] ;
F_PD =    [F_PD4, F_PD3, F_PD2, F_PD1] ;
Cs =        [FlashCs_B,BrdContRegCs_B,AtmUniCsIn_B,ToolCs1_B,ToolCs2_B] ;
BufEn =         [DataBufEn_B,ToolDataBufEn_B] ;
ConfAdd =       [A27,A28];

@ifndef L2CACHE {
CfgByte0 = [0,0,0,0,1,1,0,0];
CfgByte1 = [1,0,1,1,0,0,1,0];
CfgByte2 = [0,0,0,0,0,0,1,0];
CfgByte3 = [0,0,0,0,MODCKH0,MODCKH1,MODCKH2,MODCKH3];
}
@ifdef L2CACHE {
CfgByte0 = [0,0,0,1,1,1,0,0];
CfgByte1 = [1,0,1,1,0,0,1,0];
CfgByte2 = [0,0,0,0,0,0,1,0];
CfgByte3 = [0,0,0,0,MODCKH0,MODCKH1,MODCKH2,MODCKH3];
}
Modck = [MODCKH0..MODCKH3];

"*****************************************************************************
"* Power On Reset definitions
"*****************************************************************************
PON_RESET_ACTIVE = 0 ;
PON_RESET = (S_PORIn_B.fb == PON_RESET_ACTIVE) ;
"*****************************************************************************
```

```
"* Register Access definitions
"*****************************************************************************
BCSR0_ADD = 0 ;
BCSR1_ADD = 1 ;
BCSR2_ADD = 2 ;
BCSR3_ADD = 3 ;

VGR_WRITE_BCSR_0 = (!BrdContRegCs_B & !DVal_B & R_B_W & !A27 & !A28 & !A29) ;
VGR_WRITE_BCSR_1 = (!BrdContRegCs_B & !DVal_B & R_B_W & !A27 & !A28 & A29) ;
VGR_WRITE_BCSR_2 = (!BrdContRegCs_B & !DVal_B & R_B_W & !A27 & A28 & !A29) ;
VGR_WRITE_BCSR_3 = (!BrdContRegCs_B & !DVal_B & R_B_W & !A27 & A28 & A29) ;
VGR_WRITE_BCSR_4 = (!BrdContRegCs_B & !DVal_B & R_B_W & A27 & !A28 & !A29) ;
VGR_WRITE_BCSR_5 = (!BrdContRegCs_B & !DVal_B & R_B_W & A27 & !A28 & A29) ;
VGR_WRITE_BCSR_6 = (!BrdContRegCs_B & !DVal_B & R_B_W & A27 & A28 & !A29) ;
VGR_WRITE_BCSR_7 = (!BrdContRegCs_B & !DVal_B & R_B_W & A27 & A28 & A29) ;

VGR_READ_BCSR_0 = (!BrdContRegCs_B & !R_B_W & !A27 & !A28 & !A29) ;
VGR_READ_BCSR_1 = (!BrdContRegCs_B & !R_B_W & !A27 & !A28 & A29) ;
VGR_READ_BCSR_2 = (!BrdContRegCs_B & !R_B_W & !A27 & A28 & !A29) ;
VGR_READ_BCSR_3 = (!BrdContRegCs_B & !R_B_W & !A27 & A28 & A29) ;
VGR_READ_BCSR_4 = (!BrdContRegCs_B & !R_B_W & A27 & !A28 & !A29) ;
VGR_READ_BCSR_5 = (!BrdContRegCs_B & !R_B_W & A27 & !A28 & A29) ;
VGR_READ_BCSR_6 = (!BrdContRegCs_B & !R_B_W & A27 & A28 & !A29) ;
VGR_READ_BCSR_7 = (!BrdContRegCs_B & !R_B_W & A27 & A28 & A29) ;
"*****************************************************************************
"*****************************************************************************
"* BCSR 0 definitions.
"*****************************************************************************
"*****************************************************************************
PBI_IN_ACTIVE = 0;
DIMM_SIZE_16M = 0;
L2CACHE_INHIBITED = 0 ;
L2CACHE_FLUSHED = 0 ;
L2CACHE_LOCKED = 0 ;
L2CACHE_CLEARED = 0 ;
SIGNAL_LAMP_ON = 0 ;
"*********************************************
"******* Power On Defaults Assignments *********
"*********************************************
PBI_PON_DEFAULT = PBI_IN_ACTIVE;
DIMM_SIZE_PON_DEFAULT = DIMM_SIZE_16M;
L2CACHE_INH_PON_DEFAULT = L2CACHE_INHIBITED ;
```

```
L2CACHE_FLUSH_PON_DEFAULT = !L2CACHE_FLUSHED ;
L2CACHE_LOCK_PON_DEFAULT = !L2CACHE_LOCKED ;
L2CACHE_CLEAR_PON_DEFAULT = !L2CACHE_CLEARED ;
SIGNAL_LAMP0_PON_DEFAULT = !SIGNAL_LAMP_ON ;
SIGNAL_LAMP1_PON_DEFAULT = !SIGNAL_LAMP_ON ;
"****************************************
"******* Data Bits Assignments *************
"****************************************
PBI_DATA_BIT = [D0] ;
DIMM_SIZE_DATA_BIT = [D1] ;
L2CACHE_INH_DATA_BIT = [D2] ;
L2CACHE_FLUSH_DATA_BIT = [D3] ;
L2CACHE_LOCK_DATA_BIT = [D4] ;
L2CACHE_CLEAR_DATA_BIT = [D5] ;
SIGNAL_LAMP0_DATA_BIT = [D6] ;
SIGNAL_LAMP1_DATA_BIT = [D7] ;
"*******************************************************************************
"*******************************************************************************
"* BCSR 1 definitions.
"*******************************************************************************
"*******************************************************************************
ATM_ENABLED = 0 ;
ATM_RESET_ACTIVE = 0 ;
FETH_ENABLED = 0 ;
FETH_RESET_ACTIVE = 0 ;
RS232_1_ENABLE = 0 ;
RS232_2_ENABLE = 0 ;
"***********************************************
"******* Power On Defaults Assignments *********
"***********************************************
ATM_ENABLE_PON_DEFAULT = !ATM_ENABLED ;
ATM_RESET_PON_DEFAULT = !ATM_RESET_ACTIVE ;
FETH_ENABLE_PON_DEFAULT = !FETH_ENABLED ;
FETH_RESET_PON_DEFAULT = !FETH_RESET_ACTIVE ;
RS232_1_ENABLE_PON_DEFAULT = !RS232_1_ENABLE ;
RS232_2_ENABLE_PON_DEFAULT = !RS232_2_ENABLE ;
"****************************************
"******* Data Bits Assignments *************
"****************************************
ATM_ENABLE_DATA_BIT = [D2] ;
ATM_RESET_DATA_BIT = [D3] ;
FETH_ENABLE_DATA_BIT = [D4] ;
```

```
FETH_RESET_DATA_BIT = [D5] ;
RS232_1_ENABLE_DATA_BIT = [D6] ;
RS232_2_ENABLE_DATA_BIT = [D7] ;
“*****************************************************************************
“*****************************************************************************
“* Jtag Command / Status reg definitions.
“*****************************************************************************
“*****************************************************************************
JTAG_ENABLED = 1;
STATE_JTAG_ENABLED = (JtagEn.fb == JTAG_ENABLED);
“***********************************************
“******* Power On Defaults Assignments *********
“***********************************************
JTAG_ENABLE_PON_DEFAULT = !JTAG_ENABLED ; “ eng compatible
“*****************************************
“******* Data Bits Assignments *************
“*****************************************
JTAG_ENABLE_DATA_BIT = [D0];
“*****************************************************************************
“* Flash Declarations.
“*****************************************************************************
FLASH_ENABLE_ACTIVE = 0 ;
SM73228XU1 = (F_PD == 2) ;   “ 1 X 8 MByte bank
SM73248XU2 = (F_PD == 1) ;   “ 2 X 8 MByte banks
SM73288XU4 =  (F_PD == 0) ;   “ 4 X 8 MByte banks

FLASH_BANK1 = (  SM73228XU1 #
            (SM73248XU2 & !A8) #
            (SM73288XU4 & !A7 & !A8) ) ;
FLASH_BANK2 = ( (SM73248XU2 & A8) #
            (SM73288XU4 & !A7 & A8) ) ;
FLASH_BANK3 = (A7 & !A8 & SM73288XU4) ;
FLASH_BANK4 = (A7 & A8 & SM73288XU4) ;
“*****************************************************************************
“* Reset Declarations.
“*****************************************************************************
HARD_RESET_ACTIVE = 0 ;
SOFT_RESET_ACTIVE = 0 ;

DRIVE_MODCK_TO_VGR = (HardReset_B == HARD_RESET_ACTIVE) ; “ although power-on
                                                “ driven by HRESET.
```

```
HARD_RESET_ASSERTED = (SyncHardReset_B.fb == HARD_RESET_ACTIVE) ;
"*******************************************************************************
"* data buffers enable.
"*******************************************************************************
BUFFER_DISABLED = 1 ;
BUFFER_ENABLED = !BUFFER_DISABLED ;

BUFFER_HOLD_OFF = (HoldOffCnt.fb != 0) ; " the delay is required for read as well since a
                        " fast device (eg bcsr) may content with the flash
END_OF_FLASH_READ = !DVal_B & !FlashCs_B & !R_B_W & DSyncHardReset_B.fb ; " end of flash read cycle.
                                " not during hard reset config

END_OF_ATM_READ = !DVal_B & !AtmUniCsIn_B & !R_B_W ;    " end of atm uni m/p i/f read cycle

END_OF_OTHER_CYCLE = (!DVal_B & FlashCs_B & AtmUniCsIn_B # " another access or
            !DVal_B & !AtmUniCsIn_B & R_B_W #   " atm uni write
            !DVal_B & !ToolCs1_B & R_B_W #" tool 1 write
            !DVal_B & !ToolCs2_B & R_B_W #" tool 2 write
            !DVal_B & !FlashCs_B & R_B_W) ; " flash write

"*******************************************************************************
"* Hard Reset Configuration Logic
"*******************************************************************************
HRESET_CFG_IN_FLASH = (FlashConfEn_B == 0);
HARD_RESET_ASSERTION = ( (HardReset_B == 0) & (SyncHardReset_B.fb == 0) &
              (DSyncHardReset_B.fb == 1) );
CS0_ASSERTED = (FlashCs_B == 0);
FIRST_CFG_BYTE_READ = (!FlashCs_B & !DSyncHardReset_B.fb & (ConfAdd == 0) &
!HRESET_CFG_IN_FLASH & !R_B_W);
SCND_CFG_BYTE_READ = (!FlashCs_B & !DSyncHardReset_B.fb & (ConfAdd == 1) &
!HRESET_CFG_IN_FLASH & !R_B_W);
THIRD_CFG_BYTE_READ = (!FlashCs_B & !DSyncHardReset_B.fb & (ConfAdd == 2) &
!HRESET_CFG_IN_FLASH & !R_B_W);
FORTH_CFG_BYTE_READ = (!FlashCs_B & !DSyncHardReset_B.fb & (ConfAdd == 3) &
```

```
!HRESET_CFG_IN_FLASH & !R_B_W);
"*****************************************************************************
"*****************************************************************************
"* JTAG LOGIC definitions.
"*****************************************************************************
"*****************************************************************************
"* signals' groups
"*-----------------
JtagShiftDR =      [JtagShiftDR0..JtagShiftDR7] ;
JtagC_S_Reg =      [JtagEn.fb,0,0,0,0,0,0,JtagReceiveFull.fb];
JtagState =        [JtagState0..JtagState3] ;
JtagShiftIR =      [JtagShiftIR0..JtagShiftIR2];
JtagIR    =        [JtagIR0..JtagIR2] ;
FallingTckSignals = [JtagResetState,
                     JtagShiftIrState,
                     JtagShiftDrState,
                     JtagTdoEnable];
Read = [BrdContRegCs_B,DVal_B,R_B_W,A27,A28,A29] ;  " for simulation only

"* Constant definition
"*---------------------
JTAG_RESET = 0;
JTAG_IDLE = 1;

JTAG_SELECT_DR = 2;
JTAG_CAPTURE_DR = 3;
JTAG_SHIFT_DR = 4;
JTAG_EXIT1_DR = 5;
JTAG_PAUSE_DR = 6;
JTAG_EXIT2_DR = 7;
JTAG_UPDATE_DR = 8;

JTAG_SELECT_IR = 9;
JTAG_CAPTURE_IR = 10;
JTAG_SHIFT_IR = 11;
JTAG_EXIT1_IR = 12;
JTAG_PAUSE_IR = 13;
JTAG_EXIT2_IR = 14;
JTAG_UPDATE_IR = 15;

STATE_JTAG_RESET = (JtagState.fb == JTAG_RESET) ;
STATE_JTAG_IDLE = (JtagState.fb == JTAG_IDLE) ;
```

```
STATE_JTAG_SELECT_DR = (JtagState.fb == JTAG_SELECT_DR) ;
STATE_JTAG_CAPTURE_DR = (JtagState.fb == JTAG_CAPTURE_DR) ;
STATE_JTAG_SHIFT_DR = (JtagState.fb == JTAG_SHIFT_DR) ;
STATE_JTAG_EXIT1_DR = (JtagState.fb == JTAG_EXIT1_DR) ;
STATE_JTAG_PAUSE_DR = (JtagState.fb == JTAG_PAUSE_DR) ;
STATE_JTAG_EXIT2_DR = (JtagState.fb == JTAG_EXIT2_DR) ;
STATE_JTAG_UPDATE_DR = (JtagState.fb == JTAG_UPDATE_DR) ;

STATE_JTAG_SELECT_IR = (JtagState.fb == JTAG_SELECT_IR) ;
STATE_JTAG_CAPTURE_IR = (JtagState.fb == JTAG_CAPTURE_IR) ;
STATE_JTAG_SHIFT_IR = (JtagState.fb == JTAG_SHIFT_IR) ;
STATE_JTAG_EXIT1_IR = (JtagState.fb == JTAG_EXIT1_IR) ;
STATE_JTAG_PAUSE_IR = (JtagState.fb == JTAG_PAUSE_IR) ;
STATE_JTAG_EXIT2_IR = (JtagState.fb == JTAG_EXIT2_IR) ;
STATE_JTAG_UPDATE_IR = (JtagState.fb == JTAG_UPDATE_IR) ;
"* Instruction codes

INST_CODE_BYPASS = 7;
INST_CODE_EXTEST = 0;
INST_CODE_DOWNLOAD = 1;
INST_CODE_UPLOAD = 2;     " not supported for 1st implementaion
INST_CODE_PON_RESET = 6;
INST_CODE_UN_IMPLEMENTED = (^b011 # ^b100 # ^b101);

NEXT_INST_BYPASS = (JtagShiftIR.fb == INST_CODE_BYPASS);
NEXT_INST_EXTEST = (JtagShiftIR.fb == INST_CODE_EXTEST);
NEXT_INST_DOWNLOAD = (JtagShiftIR.fb == INST_CODE_DOWNLOAD);
NEXT_INST_UPLOAD = (JtagShiftIR.fb == INST_CODE_UPLOAD);
NEXT_INST_PON_RESET = (JtagShiftIR.fb == INST_CODE_PON_RESET);
NEXT_INST_UN_IMPLEMENTED = (JtagShiftIR.fb == INST_CODE_UN_IMPLEMENTED);

INST_IS_BYPASS = (JtagIR.fb == INST_CODE_BYPASS);
INST_IS_EXTEST = (JtagIR.fb == INST_CODE_EXTEST);
INST_IS_DOWNLOAD = (JtagIR.fb == INST_CODE_DOWNLOAD);
INST_IS_UPLOAD = (JtagIR.fb == INST_CODE_UPLOAD);
INST_IS_PON_RESET = (JtagIR.fb == INST_CODE_PON_RESET);

READ_JTAG_DOWNLOAD_CSR = VGR_READ_BCSR_6;
WRITE_JTAG_DOWNLOAD_CSR = VGR_WRITE_BCSR_6;
READ_JTAG_DOWNLOAD_DATA = VGR_READ_BCSR_7;
```

```
RECEIVE_FULL = 1;

JTAG_DOWNLOAD_SHIFT_REG_FULL = (JtagReceiveFull.fb == RECEIVE_FULL);

"*******************************************************************************
"* Equations, state diagrams.
"*******************************************************************************
equations

" !Bcsr0Write_B = (!BrdContRegCs_B & !DVal_B & R_B_W & !A28 & !A29) ;
" !Bcsr1Write_B = (!BrdContRegCs_B & !DVal_B & R_B_W & !A28 & A29) ;
"*******************************************************************************
"*******************************************************************************
"* BCSR 0
"*******************************************************************************
"*******************************************************************************
equations

ClockedContReg.clk = SYSCLK ;
ClockedContReg.ar = 0;
ClockedContReg.ap = 0;
DrivenContReg.oe = ^hfff ;

state_diagram PBI
  state PBI_IN_ACTIVE:
        if (VGR_WRITE_BCSR_0 &
              (PBI_DATA_BIT.pin == !PBI_IN_ACTIVE) &
               (!PON_RESET # (PBI_PON_DEFAULT != PBI_IN_ACTIVE)) #
               (PON_RESET & (PBI_PON_DEFAULT == !PBI_IN_ACTIVE)) ) then
              !PBI_IN_ACTIVE
        else
              PBI_IN_ACTIVE ;
  state !PBI_IN_ACTIVE:
        if (VGR_WRITE_BCSR_0 &
              (PBI_DATA_BIT.pin == PBI_IN_ACTIVE) &
               (!PON_RESET # (PBI_PON_DEFAULT != !PBI_IN_ACTIVE)) #
               (PON_RESET & (PBI_PON_DEFAULT == PBI_IN_ACTIVE)) ) then
              PBI_IN_ACTIVE
        else
              !PBI_IN_ACTIVE ;
"*******************************************************************************
state_diagram DimmSize
```

```
  state DIMM_SIZE_16M:
        if (VGR_WRITE_BCSR_0 &
              (DIMM_SIZE_DATA_BIT.pin == !DIMM_SIZE_16M) &
              (!PON_RESET # (DIMM_SIZE_PON_DEFAULT != DIMM_SIZE_16M)) #
              (PON_RESET & (DIMM_SIZE_PON_DEFAULT == !DIMM_SIZE_16M)) ) then
              !DIMM_SIZE_16M
        else
              DIMM_SIZE_16M ;
  state !DIMM_SIZE_16M:
        if (VGR_WRITE_BCSR_0 &
              (DIMM_SIZE_DATA_BIT.pin == DIMM_SIZE_16M) &
              (!PON_RESET # (DIMM_SIZE_PON_DEFAULT != !DIMM_SIZE_16M)) #
              (PON_RESET & (DIMM_SIZE_PON_DEFAULT == DIMM_SIZE_16M)) ) then
              DIMM_SIZE_16M
        else
              !DIMM_SIZE_16M ;
"*****************************************************************************
state_diagram L2Inh_B
  state L2CACHE_INHIBITED:
        if (VGR_WRITE_BCSR_0 &
              (L2CACHE_INH_DATA_BIT.pin == !L2CACHE_INHIBITED) &
              (!PON_RESET # (L2CACHE_INH_PON_DEFAULT != L2CACHE_INHIBITED)) #
              (PON_RESET & (L2CACHE_INH_PON_DEFAULT == !L2CACHE_INHIBITED)) ) then
              !L2CACHE_INHIBITED
        else
              L2CACHE_INHIBITED ;
  state !L2CACHE_INHIBITED:
        if (VGR_WRITE_BCSR_0 &
              (L2CACHE_INH_DATA_BIT.pin == L2CACHE_INHIBITED) &
              (!PON_RESET # (L2CACHE_INH_PON_DEFAULT != !L2CACHE_INHIBITED)) #
              (PON_RESET & (L2CACHE_INH_PON_DEFAULT == L2CACHE_INHIBITED)) ) then
              L2CACHE_INHIBITED
        else
              !L2CACHE_INHIBITED ;
"*****************************************************************************
state_diagram L2Flush_B
  state L2CACHE_FLUSHED:
        if (VGR_WRITE_BCSR_0 &
              (L2CACHE_FLUSH_DATA_BIT.pin == !L2CACHE_FLUSHED) &
              (!PON_RESET # (L2CACHE_FLUSH_PON_DEFAULT != L2CACHE_FLUSHED)) #
              (PON_RESET & (L2CACHE_FLUSH_PON_DEFAULT == !L2CACHE_FLUSHED)) ) then
              !L2CACHE_FLUSHED
```

```
        else
                L2CACHE_FLUSHED ;
    state !L2CACHE_FLUSHED:
        if (VGR_WRITE_BCSR_0 &
                (L2CACHE_FLUSH_DATA_BIT.pin == L2CACHE_FLUSHED) &
                 (!PON_RESET # (L2CACHE_FLUSH_PON_DEFAULT != !L2CACHE_FLUSHED)) #
                 (PON_RESET & (L2CACHE_FLUSH_PON_DEFAULT == L2CACHE_FLUSHED)) ) then
                L2CACHE_FLUSHED
        else
                !L2CACHE_FLUSHED ;
"*****************************************************************************
state_diagram L2Lock_B
    state L2CACHE_LOCKED:
        if (VGR_WRITE_BCSR_0 &
                (L2CACHE_LOCK_DATA_BIT.pin == !L2CACHE_LOCKED) &
                 (!PON_RESET # (L2CACHE_LOCK_PON_DEFAULT != L2CACHE_LOCKED)) #
                 (PON_RESET & (L2CACHE_LOCK_PON_DEFAULT == !L2CACHE_LOCKED)) ) then
                !L2CACHE_LOCKED
        else
                L2CACHE_LOCKED ;
    state !L2CACHE_LOCKED:
        if (VGR_WRITE_BCSR_0 &
                (L2CACHE_LOCK_DATA_BIT.pin == L2CACHE_LOCKED) &
                 (!PON_RESET # (L2CACHE_LOCK_PON_DEFAULT != !L2CACHE_LOCKED)) #
                 (PON_RESET & (L2CACHE_LOCK_PON_DEFAULT == L2CACHE_LOCKED)) ) then
                L2CACHE_LOCKED
        else
                !L2CACHE_LOCKED ;

"*****************************************************************************
state_diagram L2Clear_B
    state L2CACHE_CLEARED:
        if (VGR_WRITE_BCSR_0 &
                (L2CACHE_CLEAR_DATA_BIT.pin == !L2CACHE_CLEARED) &
                 (!PON_RESET # (L2CACHE_CLEAR_PON_DEFAULT != L2CACHE_CLEARED)) #
                 (PON_RESET & (L2CACHE_CLEAR_PON_DEFAULT == !L2CACHE_CLEARED)) ) then
                !L2CACHE_CLEARED
        else
                L2CACHE_CLEARED ;
    state !L2CACHE_CLEARED:
        if (VGR_WRITE_BCSR_0 &
                (L2CACHE_CLEAR_DATA_BIT.pin == L2CACHE_CLEARED) &
```

```
                (!PON_RESET # (L2CACHE_CLEAR_PON_DEFAULT != !L2CACHE_CLEARED)) #
                (PON_RESET & (L2CACHE_CLEAR_PON_DEFAULT == L2CACHE_CLEARED)) ) then
                L2CACHE_CLEARED
        else
                !L2CACHE_CLEARED ;
"*****************************************************************************
state_diagram SignaLamp0_B
  state SIGNAL_LAMP_ON:
        if (VGR_WRITE_BCSR_0 &
                (SIGNAL_LAMP0_DATA_BIT.pin == !SIGNAL_LAMP_ON) &
                (!PON_RESET # (SIGNAL_LAMP0_PON_DEFAULT != SIGNAL_LAMP_ON)) #
                (PON_RESET & (SIGNAL_LAMP0_PON_DEFAULT == !SIGNAL_LAMP_ON)) ) then
                !SIGNAL_LAMP_ON
        else
                SIGNAL_LAMP_ON ;
  state !SIGNAL_LAMP_ON:
        if (VGR_WRITE_BCSR_0 &
                (SIGNAL_LAMP0_DATA_BIT.pin == SIGNAL_LAMP_ON) &
                (!PON_RESET # (SIGNAL_LAMP0_PON_DEFAULT != !SIGNAL_LAMP_ON)) #
                (PON_RESET & (SIGNAL_LAMP0_PON_DEFAULT == SIGNAL_LAMP_ON)) ) then
                SIGNAL_LAMP_ON
        else
                !SIGNAL_LAMP_ON ;
"*****************************************************************************
state_diagram SignaLamp1_B
  state SIGNAL_LAMP_ON:
        if (VGR_WRITE_BCSR_0 &
                (SIGNAL_LAMP1_DATA_BIT.pin == !SIGNAL_LAMP_ON) &
                (!PON_RESET # (SIGNAL_LAMP1_PON_DEFAULT != SIGNAL_LAMP_ON)) #
                (PON_RESET & (SIGNAL_LAMP1_PON_DEFAULT == !SIGNAL_LAMP_ON)) ) then
                !SIGNAL_LAMP_ON
        else
                SIGNAL_LAMP_ON ;
  state !SIGNAL_LAMP_ON:
        if (VGR_WRITE_BCSR_0 &
                (SIGNAL_LAMP1_DATA_BIT.pin == SIGNAL_LAMP_ON) &
                (!PON_RESET # (SIGNAL_LAMP1_PON_DEFAULT != !SIGNAL_LAMP_ON)) #
                (PON_RESET & (SIGNAL_LAMP1_PON_DEFAULT == SIGNAL_LAMP_ON)) ) then
                SIGNAL_LAMP_ON
        else
                !SIGNAL_LAMP_ON ;
```

```
"*****************************************************************************
"*****************************************************************************
"* BCSR1 State Machines
"*****************************************************************************
"*****************************************************************************
state_diagram AtmEn_B
  state ATM_ENABLED:
        if (VGR_WRITE_BCSR_1 &
              (ATM_ENABLE_DATA_BIT.pin == !ATM_ENABLED) &
               (!PON_RESET # (ATM_ENABLE_PON_DEFAULT != ATM_ENABLED)) #
               (PON_RESET & (ATM_ENABLE_PON_DEFAULT == !ATM_ENABLED)) ) then
              !ATM_ENABLED
        else
              ATM_ENABLED ;
  state !ATM_ENABLED:
        if (VGR_WRITE_BCSR_1 &
              (ATM_ENABLE_DATA_BIT.pin == ATM_ENABLED) &
               (!PON_RESET # (ATM_ENABLE_PON_DEFAULT != !ATM_ENABLED)) #
               (PON_RESET & (ATM_ENABLE_PON_DEFAULT == ATM_ENABLED)) ) then
              ATM_ENABLED
        else
              !ATM_ENABLED ;
"*****************************************************************************
state_diagram AtmRst_B
  state ATM_RESET_ACTIVE:
        if (VGR_WRITE_BCSR_1 &
              (ATM_RESET_DATA_BIT.pin == !ATM_RESET_ACTIVE) &
               (!PON_RESET # (ATM_RESET_PON_DEFAULT != ATM_RESET_ACTIVE)) #
               (PON_RESET & (ATM_RESET_PON_DEFAULT == !ATM_RESET_ACTIVE)) ) then
              !ATM_RESET_ACTIVE
        else
              ATM_RESET_ACTIVE ;
  state !ATM_RESET_ACTIVE:
        if (VGR_WRITE_BCSR_1 &
              (ATM_RESET_DATA_BIT.pin == ATM_RESET_ACTIVE) &
               (!PON_RESET # (ATM_RESET_PON_DEFAULT != !ATM_RESET_ACTIVE)) #
               (PON_RESET & (ATM_RESET_PON_DEFAULT == ATM_RESET_ACTIVE)) ) then
              ATM_RESET_ACTIVE
        else
              !ATM_RESET_ACTIVE ;
"*****************************************************************************
state_diagram FEthEn_B
```

```
  state FETH_ENABLED:
        if (VGR_WRITE_BCSR_1 &
              (FETH_ENABLE_DATA_BIT.pin == !FETH_ENABLED) &
              (!PON_RESET # (FETH_ENABLE_PON_DEFAULT != FETH_ENABLED)) #
              (PON_RESET & (FETH_ENABLE_PON_DEFAULT == !FETH_ENABLED)) ) then
              !FETH_ENABLED
        else
              FETH_ENABLED ;
  state !FETH_ENABLED:
        if (VGR_WRITE_BCSR_1 &
              (FETH_ENABLE_DATA_BIT.pin == FETH_ENABLED) &
              (!PON_RESET # (FETH_ENABLE_PON_DEFAULT != !FETH_ENABLED)) #
              (PON_RESET & (FETH_ENABLE_PON_DEFAULT == FETH_ENABLED)) ) then
              FETH_ENABLED
        else
              !FETH_ENABLED ;
"*****************************************************************************
state_diagram FEthRst_B
  state FETH_RESET_ACTIVE:
        if (VGR_WRITE_BCSR_1 &
              (FETH_RESET_DATA_BIT.pin == !FETH_RESET_ACTIVE) &
              (!PON_RESET # (FETH_RESET_PON_DEFAULT != FETH_RESET_ACTIVE)) #
              (PON_RESET & (FETH_RESET_PON_DEFAULT == !FETH_RESET_ACTIVE)) ) then
              !FETH_RESET_ACTIVE
        else
              FETH_RESET_ACTIVE ;
  state !FETH_RESET_ACTIVE:
        if (VGR_WRITE_BCSR_1 &
              (FETH_RESET_DATA_BIT.pin == FETH_RESET_ACTIVE) &
              (!PON_RESET # (FETH_RESET_PON_DEFAULT != !FETH_RESET_ACTIVE)) #
              (PON_RESET & (FETH_RESET_PON_DEFAULT == FETH_RESET_ACTIVE)) ) then
              FETH_RESET_ACTIVE
        else
              !FETH_RESET_ACTIVE ;
"*****************************************************************************
state_diagram RS232En1_B
  state RS232_1_ENABLE:
        if (VGR_WRITE_BCSR_1 &
              (RS232_1_ENABLE_DATA_BIT.pin == !RS232_1_ENABLE) &
              (!PON_RESET # (RS232_1_ENABLE_PON_DEFAULT != RS232_1_ENABLE)) #
              (PON_RESET & (RS232_1_ENABLE_PON_DEFAULT == !RS232_1_ENABLE)) ) then
              !RS232_1_ENABLE
```

```
            else
                    RS232_1_ENABLE ;
    state !RS232_1_ENABLE:
            if (VGR_WRITE_BCSR_1 &
                    (RS232_1_ENABLE_DATA_BIT.pin == RS232_1_ENABLE) &
                     (!PON_RESET # (RS232_1_ENABLE_PON_DEFAULT != !RS232_1_ENABLE)) #
                     (PON_RESET & (RS232_1_ENABLE_PON_DEFAULT == RS232_1_ENABLE)) ) then
                    RS232_1_ENABLE
            else
                    !RS232_1_ENABLE ;
"*************************************************************************
state_diagram RS232En2_B
    state RS232_2_ENABLE:
            if (VGR_WRITE_BCSR_1 &
                    (RS232_2_ENABLE_DATA_BIT.pin == !RS232_2_ENABLE) &
                     (!PON_RESET # (RS232_2_ENABLE_PON_DEFAULT != RS232_2_ENABLE)) #
                     (PON_RESET & (RS232_2_ENABLE_PON_DEFAULT == !RS232_2_ENABLE)) ) then
                    !RS232_2_ENABLE
            else
                    RS232_2_ENABLE ;
    state !RS232_2_ENABLE:
            if (VGR_WRITE_BCSR_1 &
                    (RS232_2_ENABLE_DATA_BIT.pin == RS232_2_ENABLE) &
                     (!PON_RESET # (RS232_2_ENABLE_PON_DEFAULT != !RS232_2_ENABLE)) #
                     (PON_RESET & (RS232_2_ENABLE_PON_DEFAULT == RS232_2_ENABLE)) ) then
                    RS232_2_ENABLE
            else
                    !RS232_2_ENABLE ;
"*************************************************************************
"*************************************************************************
"* BCSR 3
"*************************************************************************
"*************************************************************************
@ifdef JTAG {

state_diagram JtagEn
    state JTAG_ENABLED:
            if (WRITE_JTAG_DOWNLOAD_CSR &
                    (JTAG_ENABLE_DATA_BIT.pin == !JTAG_ENABLED) &
                     (!PON_RESET # (JTAG_ENABLE_PON_DEFAULT != JTAG_ENABLED)) #
                     (PON_RESET & (JTAG_ENABLE_PON_DEFAULT == !JTAG_ENABLED)) ) then
                    !JTAG_ENABLED
```

```
            else
                    JTAG_ENABLED ;
   state !JTAG_ENABLED:
            if (WRITE_JTAG_DOWNLOAD_CSR &
                    (JTAG_ENABLE_DATA_BIT.pin == JTAG_ENABLED) &
                     (!PON_RESET # (JTAG_ENABLE_PON_DEFAULT != !JTAG_ENABLED)) #
                     (PON_RESET & (JTAG_ENABLE_PON_DEFAULT == JTAG_ENABLED)) ) then
                    JTAG_ENABLED
            else
                    !JTAG_ENABLED ;   }
"*****************************************************************************
"*****************************************************************************
" External Read Registers' Chip-Selects
"*****************************************************************************
"*****************************************************************************
equations
Bcsr2Cs_B.oe = H ;
!Bcsr2Cs_B = VGR_READ_BCSR_2 ;
"*****************************************************************************
"*****************************************************************************
"* Read Registers.
"* All registers have read capabilty. (BCSR2 is read externally)
"*****************************************************************************
"*****************************************************************************
equations
   DataOe = VGR_READ_BCSR_0 #
          VGR_READ_BCSR_1 #
@ifdef JTAG {    READ_JTAG_DOWNLOAD_DATA #
              READ_JTAG_DOWNLOAD_CSR #       }
          !HRESET_CFG_IN_FLASH & !FlashCs_B & !DSyncHardReset_B.fb ;
   Data.oe = DataOe.fb ;

   when (VGR_READ_BCSR_0) then
            Data = ReadBcsr0 ;
   else when (VGR_READ_BCSR_1) then
            Data = ReadBcsr1 ;
@ifdef JTAG {
       else when (READ_JTAG_DOWNLOAD_DATA) then
            Data = JtagShiftDR.fb;
       else when (READ_JTAG_DOWNLOAD_CSR) then
            Data = JtagC_S_Reg;              }
       else when (FIRST_CFG_BYTE_READ) then
```

```
        Data = CfgByte0;
    else when (SCND_CFG_BYTE_READ) then
        Data = CfgByte1;
    else when (THIRD_CFG_BYTE_READ) then
        Data = CfgByte2;
    else when (FORTH_CFG_BYTE_READ) then
        Data = CfgByte3;

"*******************************************************************************
"* Reset Logic
"*******************************************************************************
equations

Reset.oe = ResetEn ;

Reset = 0 ;" open drain

RstDeb1 = !( Rst1 & (!( RstDeb1.fb & Rst0) ) ) ;    " Reset push-button debouncer

AbrDeb1 = !( Abr1 & (!( AbrDeb1.fb & Abr0) ) ) ;    " Abort push-button debouncer


HardResetEn = RstDeb1.fb & AbrDeb1.fb  ;" both buttons are depressed;

SoftResetEn = RstDeb1.fb & !AbrDeb1.fb ;" only reset button depressed

TransRst.oe = 3 ;" transceivers' reset, always enabled.

!AtmRstOut_B = !AtmRst_B.fb # !HardReset_B ;

!FEthRstOut_B = !FEthRst_B.fb # !HardReset_B ;

"*******************************************************************************
"* Hard reset configuration
"*******************************************************************************
equations

RstConf_B.oe = H;

RstConf_B = L;

"*******************************************************************************
```

```
"* NMI generation
"*****************************************************************************
equations

NMI_B.oe = NMIEn ;

NMI_B = 0 ;" O.D.

NMIEn = !RstDeb1.fb & AbrDeb1.fb ;" only abort button depressed

"*****************************************************************************
"* local data buffers enable
"*****************************************************************************
equations

SyncHardReset_B.clk = SYSCLK ;
SyncHardReset_B.ar = 0;
SyncHardReset_B.ap = 0;

DSyncHardReset_B.clk = SYSCLK ;
DSyncHardReset_B.ar = 0;
DSyncHardReset_B.ap = 0;

SyncHardReset_B := HardReset_B ;
DSyncHardReset_B := SyncHardReset_B.fb ;

DataBufEn_B.oe = H ;

!DataBufEn_B = ( !FlashCs_B  #     " covers also hard reset config
            !BrdContRegCs_B #
            !AtmUniCsOut_B #   " provides data-hold for write
            !ToolCs1_B #
            !ToolCs2_B)  &
                  (!BUFFER_HOLD_OFF)  ;

ToolDataBufEn_B.oe = H ;

!ToolDataBufEn_B = (!ToolCs1_B #
              !ToolCs2_B)  &  (!BUFFER_HOLD_OFF)  ;

"*****************************************************************************
"* local data buffers disable (data contention protection)
```

```
"*******************************************************************************
"* Since with MPC8260, hard-reset conf is read from flash during HRESET
"* asserted and since these are all consequitive read cycles and since
"* the cycles following hard reset are also reads (boot) the hold-off
"* state machine may be left in NO_HOLD_OFF for HRESET_B asserted duration
"* without warrying about contention between flash and data buffers.

equations

HoldOffCnt.clk = SYSCLK ;
HoldOffCnt.ar = 0;
HoldOffCnt.ap = 0;

HoldOffTc = (HoldOffCnt.fb == 3) ;

when ( (((END_OF_FLASH_READ # END_OF_ATM_READ ) & (HoldOffCnt.fb == 0)) #
       (HoldOffCnt.fb != 0)) & !HoldOffTc.fb & DSyncHardReset_B.fb  ) then
    HoldOffCnt := HoldOffCnt.fb + 1 ;
else
    HoldOffCnt := 0 ;
"*******************************************************************************
"* Flash Chip Select
"*******************************************************************************
equations

FlashCsOut.oe = ^hf ;

!FlashCs1_B = !FlashCs_B & FLASH_BANK1 & HRESET_CFG_IN_FLASH #
          !FlashCs_B & FLASH_BANK1 & !HRESET_CFG_IN_FLASH & DSyncHardReset_B.fb;
!FlashCs2_B = !FlashCs_B & FLASH_BANK2 & HRESET_CFG_IN_FLASH #
          !FlashCs_B & FLASH_BANK2 & !HRESET_CFG_IN_FLASH & DSyncHardReset_B.fb;
!FlashCs3_B = !FlashCs_B & FLASH_BANK3 & HRESET_CFG_IN_FLASH #
          !FlashCs_B & FLASH_BANK3 & !HRESET_CFG_IN_FLASH & DSyncHardReset_B.fb;
!FlashCs4_B = !FlashCs_B & FLASH_BANK4 & HRESET_CFG_IN_FLASH #
          !FlashCs_B & FLASH_BANK4 & !HRESET_CFG_IN_FLASH & DSyncHardReset_B.fb;
"*******************************************************************************
"* ATM UNI Chip Select
"*******************************************************************************
equations

AtmUniCsOut_B.oe = H ;
```

```
!AtmUniCsOut_B = !AtmUniCsIn_B;
"*******************************************************************************
"* Power On Reset
"*******************************************************************************
equations

S_PORIn_B.clk = SYSCLK ;
S_PORIn_B.ar = 0;
S_PORIn_B.ap = 0;

S_PORIn_B := PORIn_B ;
"*******************************************************************************
"*******************************************************************************
"* JTAG LOGIC equations
"*******************************************************************************
"*******************************************************************************
equations
JtagStateReset = (!Trst_B # !PORIn_B) ;  " only for jtag state machine
JtagReset = (!Trst_B # !PORIn_B # !JtagResetState.fb);  " global reset

JtagState.clk = Tck;
JtagState.ar = JtagStateReset.fb ;
JtagState.ap = 0;
"*******************************************************************************
"* Standard JTAG state machine

state_diagram JtagState
    state JTAG_RESET:
        if (!Tms) then
            JTAG_IDLE
        else
            JTAG_RESET;
    state JTAG_IDLE:
        if (Tms) then
            JTAG_SELECT_DR
        else
            JTAG_IDLE;
    state JTAG_SELECT_DR:          " DR
        if (!Tms) then
            JTAG_CAPTURE_DR
        else
            JTAG_SELECT_IR;
```

```
state JTAG_CAPTURE_DR:
    if (!Tms) then
        JTAG_SHIFT_DR
    else
        JTAG_EXIT1_DR;
state JTAG_SHIFT_DR:
    if (Tms) then
        JTAG_EXIT1_DR
    else
        JTAG_SHIFT_DR;
state JTAG_EXIT1_DR:
    if (!Tms) then
        JTAG_PAUSE_DR
    else
        JTAG_UPDATE_DR;
state JTAG_PAUSE_DR:
    if (Tms) then
        JTAG_EXIT2_DR
    else
        JTAG_PAUSE_DR;
state JTAG_EXIT2_DR:
    if (Tms) then
        JTAG_UPDATE_DR
    else
        JTAG_SHIFT_DR;
state JTAG_UPDATE_DR:
    if (!Tms) then
        JTAG_IDLE
    else
        JTAG_SELECT_DR;
state JTAG_SELECT_IR:          " IR
    if (!Tms) then
        JTAG_CAPTURE_IR
    else
        JTAG_RESET;
state JTAG_CAPTURE_IR:
    if (!Tms) then
        JTAG_SHIFT_IR
    else
        JTAG_EXIT1_IR;
state JTAG_SHIFT_IR:
    if (Tms) then
```

```
            JTAG_EXIT1_IR
        else
            JTAG_SHIFT_IR;
    state JTAG_EXIT1_IR:
        if (!Tms) then
            JTAG_PAUSE_IR
        else
            JTAG_UPDATE_IR;
    state JTAG_PAUSE_IR:
        if (Tms) then
            JTAG_EXIT2_IR
        else
            JTAG_PAUSE_IR;
    state JTAG_EXIT2_IR:
        if (Tms) then
            JTAG_UPDATE_IR
        else
            JTAG_SHIFT_IR;
    state JTAG_UPDATE_IR:
        if (!Tms) then
            JTAG_IDLE
        else
            JTAG_SELECT_DR;
"*******************************************************************************
"* Jtag Dedicated State Signals (Falling Edge TCK)
equations

FallingTckSignals.clk = !Tck;
FallingTckSignals.ar = JtagStateReset.fb ;
FallingTckSignals.ap = 0;

!JtagResetState := STATE_JTAG_RESET;   " Active-Low

JtagShiftIrState := STATE_JTAG_SHIFT_IR;

JtagShiftDrState := STATE_JTAG_SHIFT_DR;
JtagTdoEnable := (STATE_JTAG_SHIFT_IR # STATE_JTAG_SHIFT_DR) ;
"*******************************************************************************
"* Jtag Instruction Shift Register
equations

JtagShiftIR.clk = Tck;
```

```
JtagShiftIR.ap = JtagReset.fb ;   "  reset -> bypass
JtagShiftIR.ar = 0;

when (JtagShiftIrState.fb & STATE_JTAG_ENABLED) then
    JtagShiftIR := [Tdi,JtagShiftIR0.fb,JtagShiftIR1.fb];
else when (STATE_JTAG_CAPTURE_IR & STATE_JTAG_ENABLED) then
    JtagShiftIR := INST_CODE_BYPASS;  " interim default
else
    JtagShiftIR := JtagShiftIR.fb;
"*******************************************************************************
"* Jtag Instruction Register
equations

JtagIR.clk = !Tck;
JtagIR.ap = JtagReset.fb ;  " reset -> bypass
JtagIR.ar = 0;

when (STATE_JTAG_UPDATE_IR & (NEXT_INST_DOWNLOAD # NEXT_INST_PON_RESET) ) then
      JtagIR := JtagShiftIR.fb;
else when (STATE_JTAG_UPDATE_IR & (!(NEXT_INST_DOWNLOAD #
                                NEXT_INST_PON_RESET) ) ) then
      JtagIR := INST_CODE_BYPASS ;
else
      JtagIR := JtagIR.fb;    " hold value
"*******************************************************************************
"* Download Shift / Bypass Register
equations

JtagShiftDR.clk = Tck;
JtagShiftDR.ar = JtagStateReset.fb ;
JtagShiftDR.ap = 0;

when (JtagShiftDrState.fb & INST_IS_BYPASS #
     JtagShiftDrState.fb & INST_IS_DOWNLOAD & !JTAG_DOWNLOAD_SHIFT_REG_FULL) then
    [JtagShiftDR0..JtagShiftDR7] := [Tdi,JtagShiftDR0.fb,JtagShiftDR1.fb,
                                JtagShiftDR2.fb,JtagShiftDR3.fb,
                                JtagShiftDR4.fb,JtagShiftDR5.fb,
                                JtagShiftDR6.fb];
else
    JtagShiftDR := JtagShiftDR.fb ;  " hold
"*******************************************************************************
"* Receive Full Flag
```

```
equations

JtagReceiveFull.clk = !Tck;

JtagReceiveFull.ar = JtagReceiveFullReset.fb ;
JtagReceiveFull.ap = 0;

when (STATE_JTAG_EXIT1_DR & INST_IS_DOWNLOAD) then  " end of download byte
    JtagReceiveFull := RECEIVE_FULL;
else
    JtagReceiveFull := JtagReceiveFull.fb;          " maintain value
"*****************************************************************************
"* Receive Full Flag Reset

SReadJtagDownloadData.clk = SYSCLK;
SReadJtagDownloadData.ar = 0;
SReadJtagDownloadData.ap = 0;

SReadJtagDownloadData := READ_JTAG_DOWNLOAD_DATA & !DVal_B;


JtagReceiveFullReset = ( SReadJtagDownloadData.fb #
                !Trst_B #
                !PORIn_B #
                !JtagResetState.fb) ;

"*****************************************************************************
"* TDO Selection
equations

Tdo.clk = !Tck;
Tdo.ar = 0;
Tdo.ap = JtagStateReset.fb ;

when (STATE_JTAG_SHIFT_IR) then
  Tdo := JtagShiftIR2.fb;
else when (STATE_JTAG_SHIFT_DR & INST_IS_BYPASS) then
  Tdo := JtagShiftDR0.fb;
else when (STATE_JTAG_SHIFT_DR & INST_IS_DOWNLOAD) then
  Tdo := JtagReceiveFull.fb;
```

```
when (STATE_JTAG_ENABLED) then
  TdoOut = Tdo.fb;
else when (!STATE_JTAG_ENABLED) then
   TdoOut = Tdi;

TdoEnable = (STATE_JTAG_ENABLED & JtagTdoEnable.fb #
         !STATE_JTAG_ENABLED) ;

TdoOut.oe = TdoEnable.fb;
"*******************************************************************************
"* Power On Reset Generation
equations
PonResetOut.oe = 1;
PonResetOut = INST_IS_PON_RESET ;
"*******************************************************************************
"* Auxiliary functions
"*******************************************************************************
equations
KeepPinsConnected = TEA_B & BCTL1 # KeepPinsConnected.fb;

END
```

5.3.2 U18 - SDRAM's Latch-Mux - Low (L2-Cache Only)

```
MODULE vmuxlow5

TITLE  'MPC8260 Sdram 1st Latch - Mux'
"*******************************************************************************
"* Pins declaration.
"*******************************************************************************
"* Control pins
"*******************************************************************************
Ale_B              PIN      5;
AleIn              PIN     29;
AleOut_B           PIN     27 istype 'com' ; " inverted AleIn, connected externaly to
                                " Ale_B
R_C_B              PIN     33;
DimmSize           PIN     37;
PBI                PIN     25;
"*******************************************************************************
"* Address Input lines
"*******************************************************************************
" row
A10                PIN     34;
A11                PIN      1;
A12                PIN      2;
A13                PIN      3;
A14                PIN      9;
A15                PIN     10;
A16                PIN     11;
A17                PIN     12;
A18                PIN     35;
A19                PIN     36;
" col
A21                PIN     38;
A22                PIN     39;
A23                PIN     40;
A24                PIN     44;
A25                PIN     45;
A26                PIN     46;
A27                PIN     47;
A28                PIN     48;

"*******************************************************************************
```

```
"* Address Output lines
"*****************************************************************************
SdramA0         PIN     13 istype 'com' ;
SdramA1         PIN     14 istype 'com' ;
SdramA2         PIN     15 istype 'com' ;
SdramA3         PIN     16 istype 'com' ;
SdramA4         PIN     20 istype 'com' ;
SdramA5         PIN     21 istype 'com' ;
SdramA6         PIN     22 istype 'com' ;
SdramA7         PIN     23 istype 'com' ;
"*****************************************************************************
"*****************************************************************************
"* Latched Address lines
"*****************************************************************************
LA10        NODE istype 'reg_D,buffer' ;
LA11        NODE istype 'reg_D,buffer' ;
LA12        NODE istype 'reg_D,buffer' ;
LA13        NODE istype 'reg_D,buffer' ;
LA14        NODE istype 'reg_D,buffer' ;
LA15        NODE istype 'reg_D,buffer' ;
LA16        NODE istype 'reg_D,buffer' ;
LA17        NODE istype 'reg_D,buffer' ;
LA18        NODE istype 'reg_D,buffer' ;
LA19        NODE istype 'reg_D,buffer' ;
" col
LA21        NODE istype 'reg_D,buffer' ;
LA22        NODE istype 'reg_D,buffer' ;
LA23        NODE istype 'reg_D,buffer' ;
LA24        NODE istype 'reg_D,buffer' ;
LA25        NODE istype 'reg_D,buffer' ;
LA26        NODE istype 'reg_D,buffer' ;
LA27        NODE istype 'reg_D,buffer' ;
LA28        NODE istype 'reg_D,buffer' ;
"*****************************************************************************
"*  Constant Declaration
"*****************************************************************************
H, L, X, Z = 1, 0, .X., .Z. ;
C, D, U    = .C., .D., .U. ;
"*****************************************************************************
"* Signal groups
"*****************************************************************************
Add = [A10..A19,A21..A28] ;
```

```
LAdd = [LA10..LA19,LA21..LA28] ;

RowAddNormal = [LA12..LA19] ;
RowAddPBI_16M = [LA11..LA18] ;
RowAddPBI_64M = [LA10..LA17] ;

ColAdd = [LA21..LA28] ;

SdramAdd = [SdramA7..SdramA0] ;

ROW = (R_C_B == 1) ;
COL = !ROW ;

SIZE_16M = 0;
SIZE_64M = 1;

SDRAM_16M = (DimmSize == SIZE_16M);
SDRAM_64M = !SDRAM_16M;

SDRAM_NORMAL_MODE = (PBI == 0);
SDRAM_PBI_MODE = !SDRAM_NORMAL_MODE;

MuxCont = [PBI,DimmSize];
"*******************************************************************************
"* Equations, state diagrams.
"*******************************************************************************
"* Input Latch
"*******************************************************************************
equations

!AleOut_B = AleIn ;   " inverted Ale

LAdd.le = Ale_B ;

LAdd.d = Add ;    " latching the address
"*******************************************************************************
"* Output Mux
"*******************************************************************************
equations

SdramAdd.oe = ^hff ;   "always enabled
```

```
when (ROW & SDRAM_NORMAL_MODE) then
    SdramAdd = RowAddNormal.q
else when (ROW & SDRAM_PBI_MODE & SDRAM_16M) then
    SdramAdd = RowAddPBI_16M.q
else when (ROW & SDRAM_PBI_MODE & SDRAM_64M) then
    SdramAdd = RowAddPBI_64M.q
else
    SdramAdd = ColAdd.q ;

END
```

5.3.3 U19 - SDRAM's Latch-Mux - High (L2-Cache Only)

```
MODULE vmuxhig5
TITLE  'MPC8260 Sdram 2nd Latch - Mux'
"*******************************************************************************
"* This file contains the 2'nd part of Address latch & mux for the MPC8260 ADS
"* sdram.
"* The mux is required only with L2 Cache installed on board. Otherwise
"* it is not assembled and SdramA(13:8) are connected to MPC8260's proper
"* address lines.
"*******************************************************************************
"* In this file (4) Rev PILOT 02/18/99:
"* - Added support for PBI - Page Based Interleaving feature added with
"*   MPC8260 rev A. To provide that support the following were made:
"*   - Added DimmSize input, which provides SDRAM dimm size information.
"*     Only 16Meg (reset default) and 64Meg DIMMs are supported.
"*     This signal originates in BCSR.
"*   - Added PBI indication. Since PBI's actual setting is done within the
"*     MPC8260, it is the system programmer responsibility to set the correct
"*     value for this signal in BCSR, to ensure proper operation.
"*   - To make room for the above, SdramA(9:8) are moved to another
"*     device, along with A20.
"*   - The output of this mux is now qualified with the DimmSize and PBI
"*     information to provide the correct address lines to the sdram dimm.
"*******************************************************************************
"* In this file (5) Rev PILOT 03/21/99:
"* - Pinout changed to M4-64/32-7VC48 (TQFP48 package).
"*******************************************************************************
"*******************************************************************************
"* Pins declaration.
"*******************************************************************************
"* Control pins
"*******************************************************************************
Ale_B              PIN     5;
R_C_B              PIN     1;
DimmSize           PIN     2;
PBI                PIN     3;
"*******************************************************************************
"* Address Input lines
"*******************************************************************************
" row
A6                 PIN     37;
```

```
A7              PIN     38;
A8              PIN     39;
A9              PIN     40;
A10             PIN     44;
A11             PIN     45;
" col
A20             PIN     46;
"*****************************************************************************
"* Address Output lines
"*****************************************************************************
SdramA8         PIN     9 istype 'com' ;
SdramA9         PIN     24 istype 'com' ;
SdramA11        PIN     23 istype 'com' ;
"*****************************************************************************
"* Latched Address lines
"*****************************************************************************
LA6         NODE istype 'reg_D,buffer' ;
LA7         NODE istype 'reg_D,buffer' ;
LA8         NODE istype 'reg_D,buffer' ;
LA9         NODE istype 'reg_D,buffer' ;
LA10        NODE istype 'reg_D,buffer' ;
LA11        NODE istype 'reg_D,buffer' ;

" col
LA20        NODE istype 'reg_D,buffer' ;
"*****************************************************************************
"*  Constant Declaration
"*****************************************************************************
H, L, X, Z = 1, 0, .X., .Z. ;
C, D, U    = .C., .D., .U. ;
"*****************************************************************************
"* Signal groups
"*****************************************************************************
Add = [A6..A11,A20] ;

LAdd = [LA6..LA11,LA20] ;

RowAddNormal = [LA8,LA10,LA11] ;
RowAddPBI_16M = [LA7,LA9,LA10] ; " LA7 not really required
RowAddPBI_64M = [LA6,LA8,LA9] ;

ColAddNormal = [LA8,LA10,LA20] ;
```

```
ColAddPBI_16M = [LA7,LA9,LA20] ; " LA7 not really required
ColAddPBI_64M = [LA6,LA8,LA20] ;

SdramAdd = [SdramA11,SdramA9,SdramA8] ;

ROW = (R_C_B == 1) ;
COL = !ROW ;

SIZE_16M = 0;
SIZE_64M = 1;

SDRAM_16M = (DimmSize == SIZE_16M);
SDRAM_64M = !SDRAM_16M;

SDRAM_NORMAL_MODE = (PBI == 0);
SDRAM_PBI_MODE = !SDRAM_NORMAL_MODE;

MuxCont = [PBI,DimmSize];

"*******************************************************************************
"* Equations, state diagrams.
"*******************************************************************************
"* Input Latch
"*******************************************************************************
equations
LAdd.le = Ale_B ;

LAdd.d = Add ;    " latching the address
"*******************************************************************************
"* Output Mux
"*******************************************************************************
equations

SdramAdd.oe = ^h7 ;   "always enabled

when (ROW & SDRAM_NORMAL_MODE) then
    SdramAdd = RowAddNormal.q
else when (ROW & SDRAM_PBI_MODE & SDRAM_16M) then
    SdramAdd = RowAddPBI_16M.q
else when (ROW & SDRAM_PBI_MODE & SDRAM_64M) then
    SdramAdd = RowAddPBI_64M.q
else when (COL & SDRAM_NORMAL_MODE) then
```

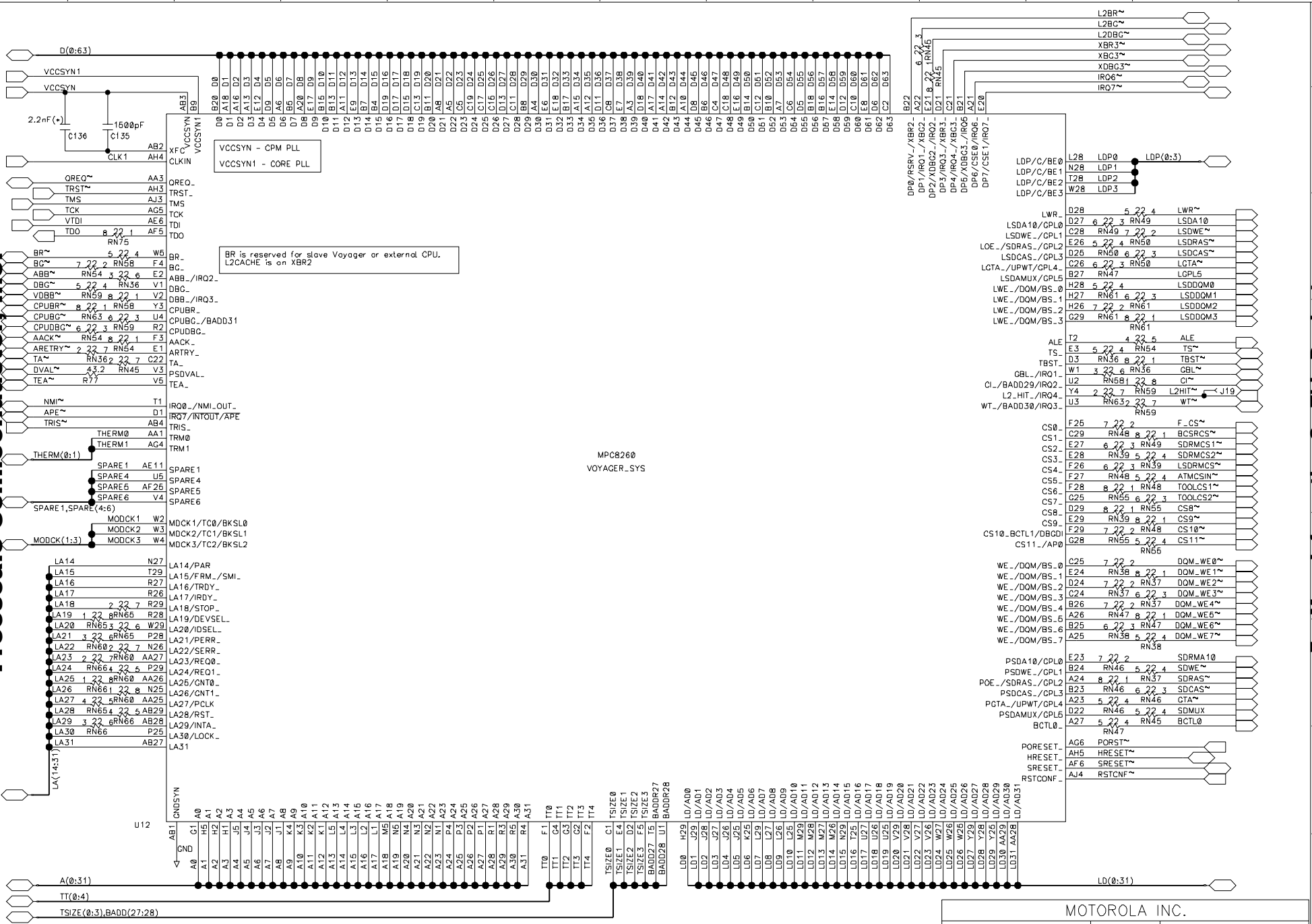
```
    SdramAdd = ColAddNormal.q
else when (COL & SDRAM_PBI_MODE & SDRAM_16M) then
    SdramAdd = ColAddPBI_16M.q
else when (COL & SDRAM_PBI_MODE & SDRAM_64M) then
    SdramAdd = ColAddPBI_64M.q;

END
```

Chapter 6 Schematics

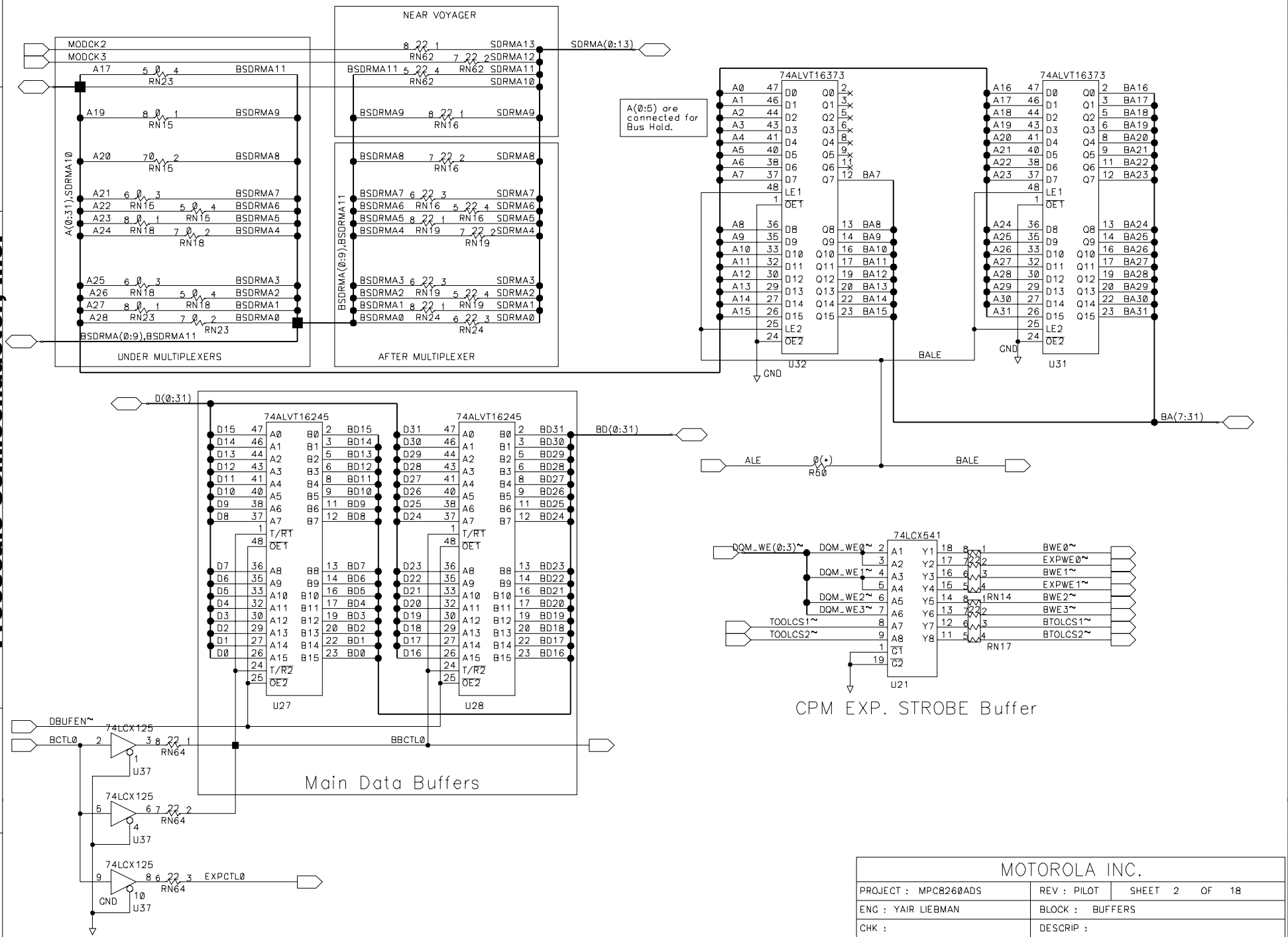
6.1 Introduction

This chapter shows the schematics for the 8260 ADS board.



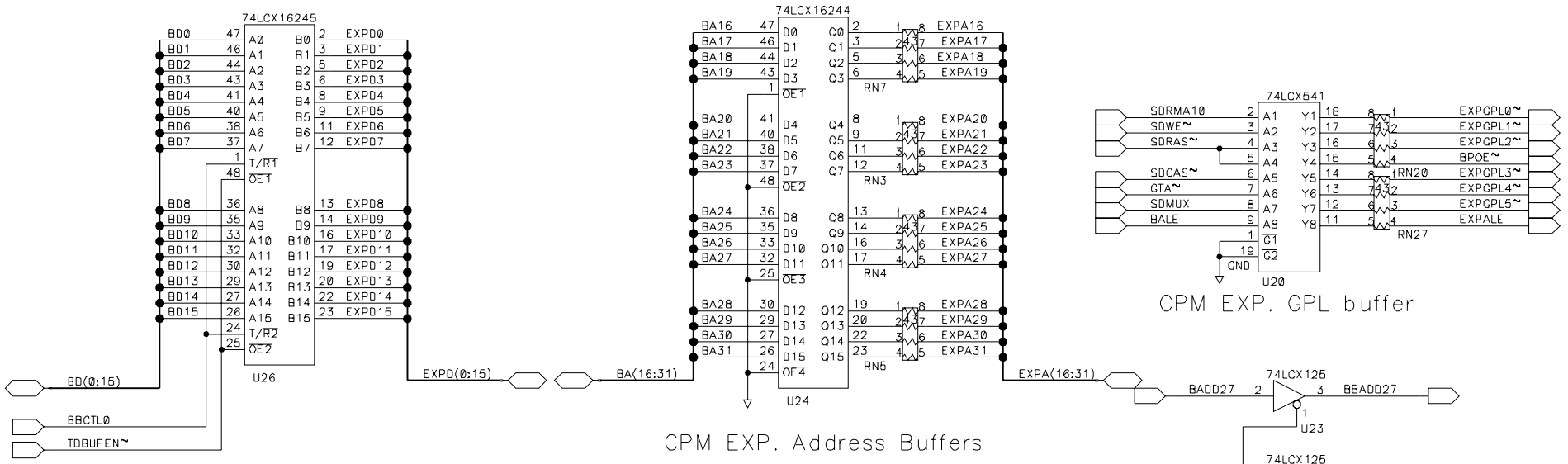
MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 1 OF 18
ENG : YAIR LIEBMAN	BLOCK : VOYAGER SYSTEM	
CHK :	DESCRIP :	

Freescale Semiconductor, Inc.

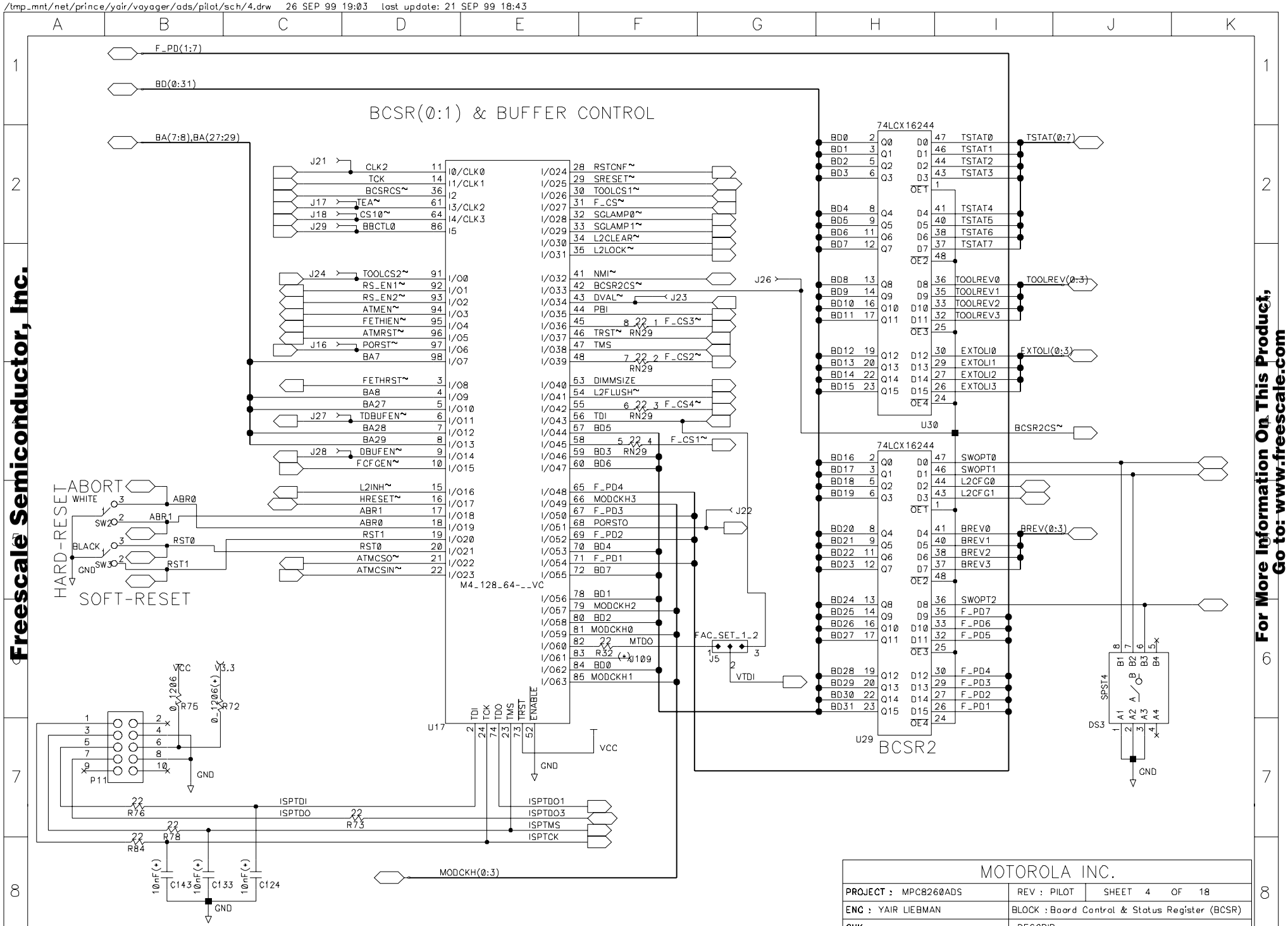


MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 2 OF 18
ENG : YAIR LIEBMAN	BLOCK : BUFFERS	
CHK :	DESCRIP :	

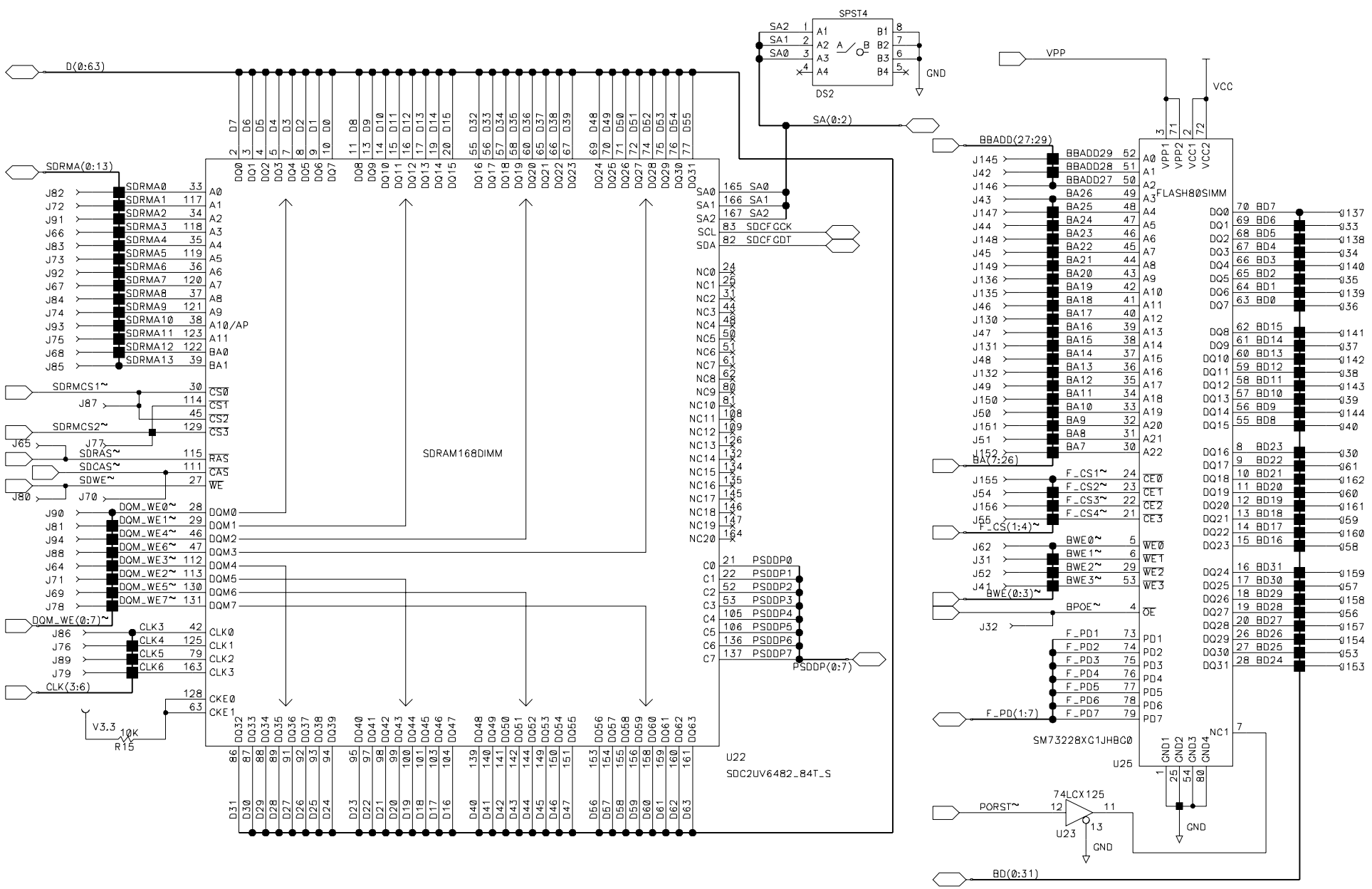
Freescale Semiconductor, Inc.

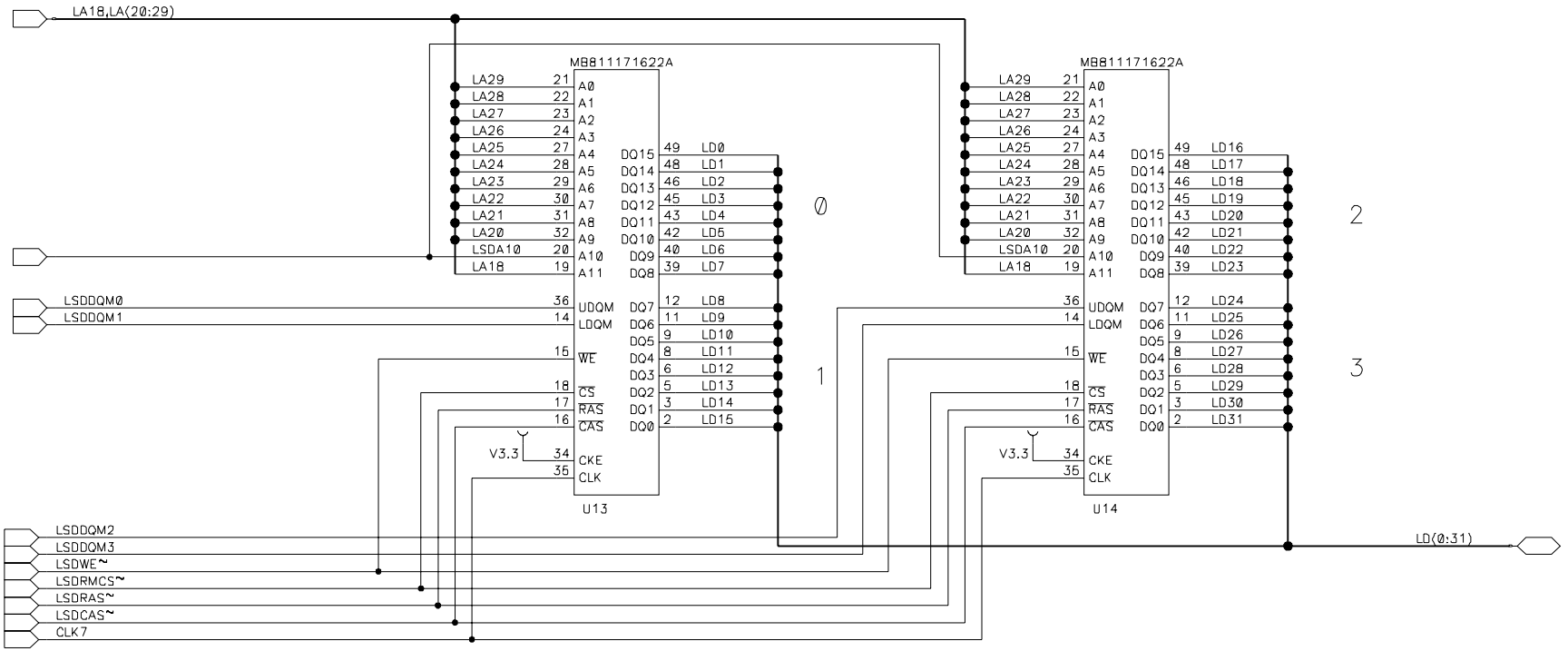


MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 3 OF 18
ENG : YAIR LIEBMAN	BLOCK EXP. CONN., CLOCKs & PON RESET CONF.	
(•): NOT ASSEMBLED	CHK :	DESCRIP :

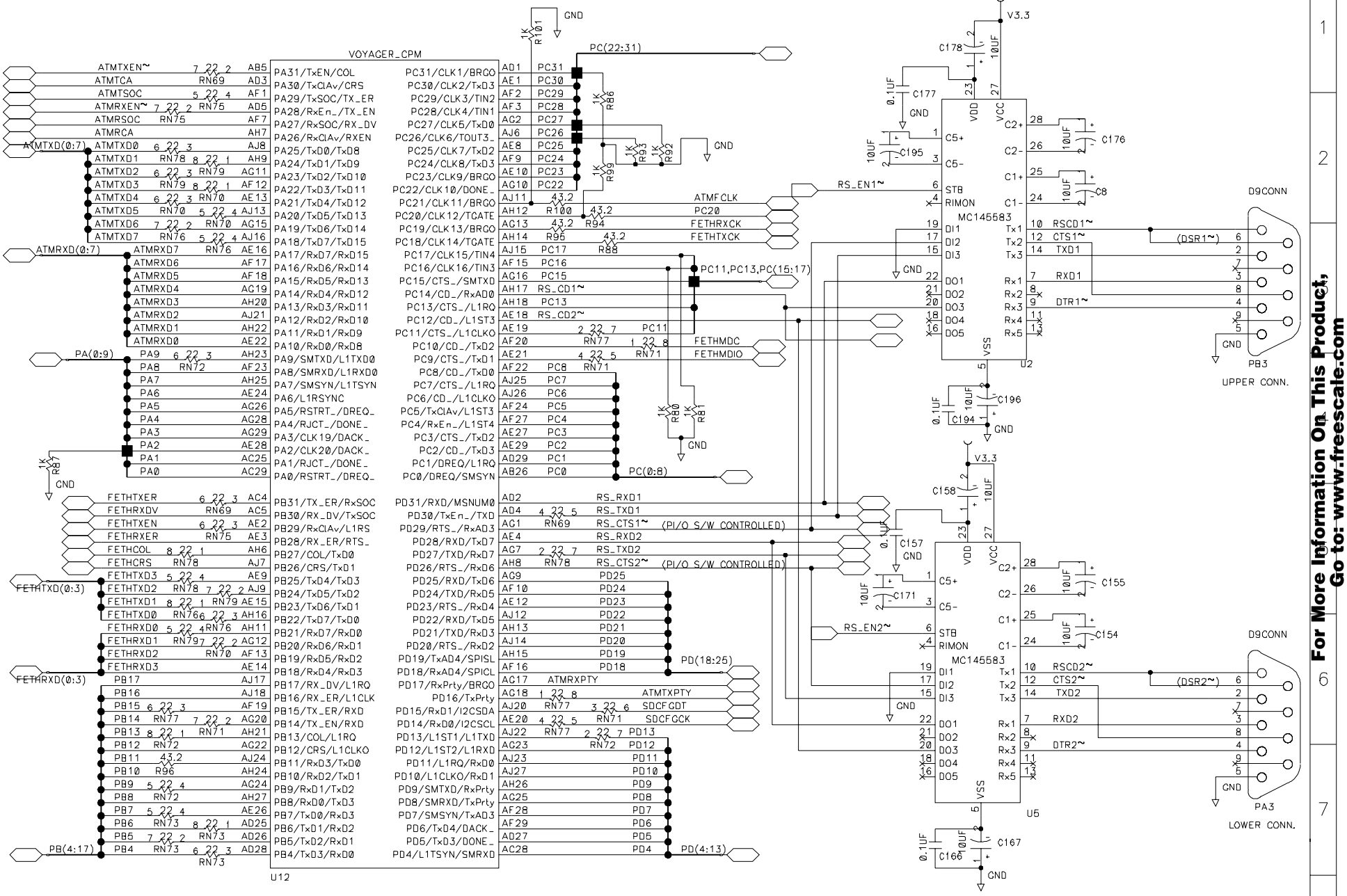


MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 4 OF 18
ENG : YAIR LIEBMAN	BLOCK : Board Control & Status Register (BCSR)	
CHK :	DESCRIP :	

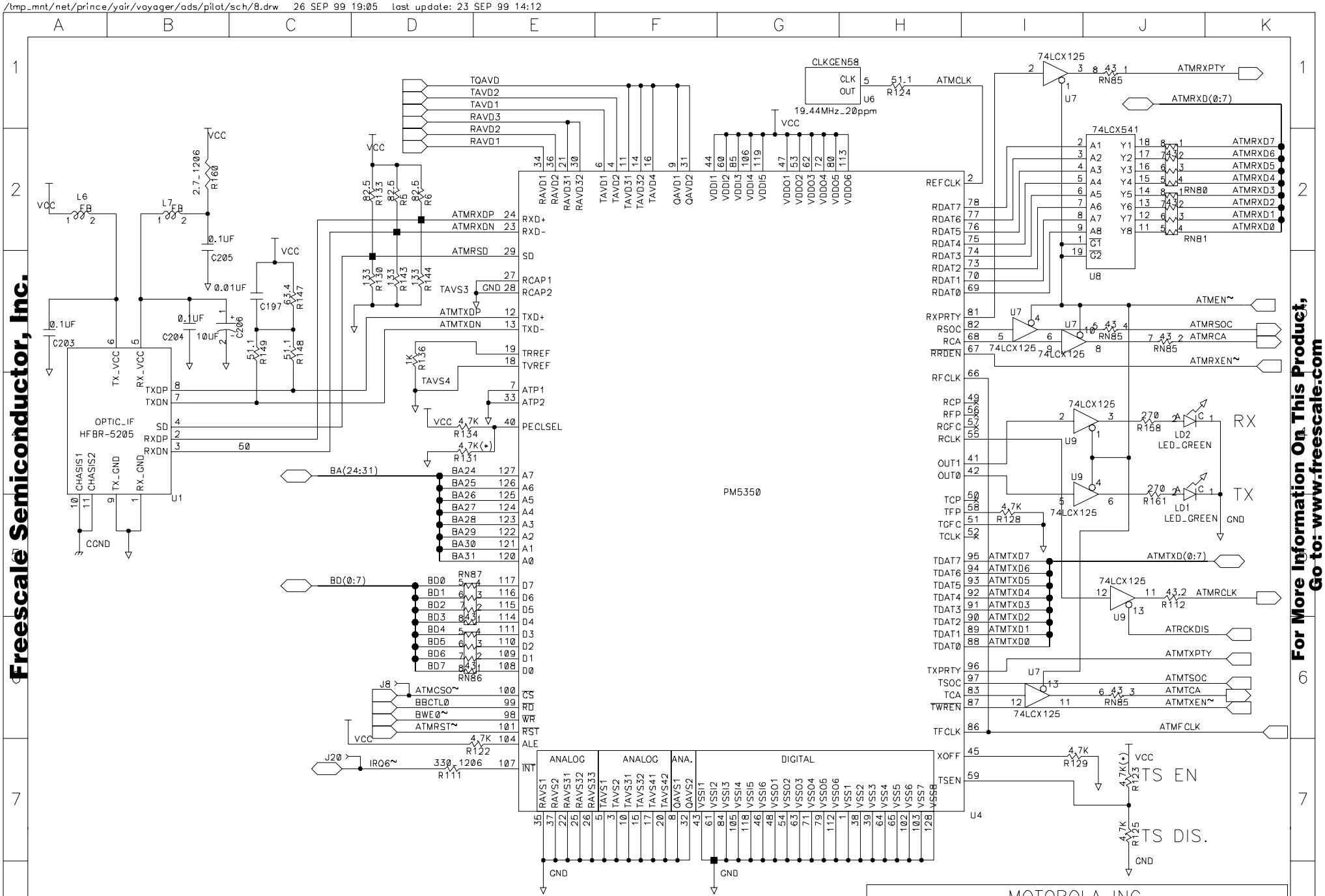


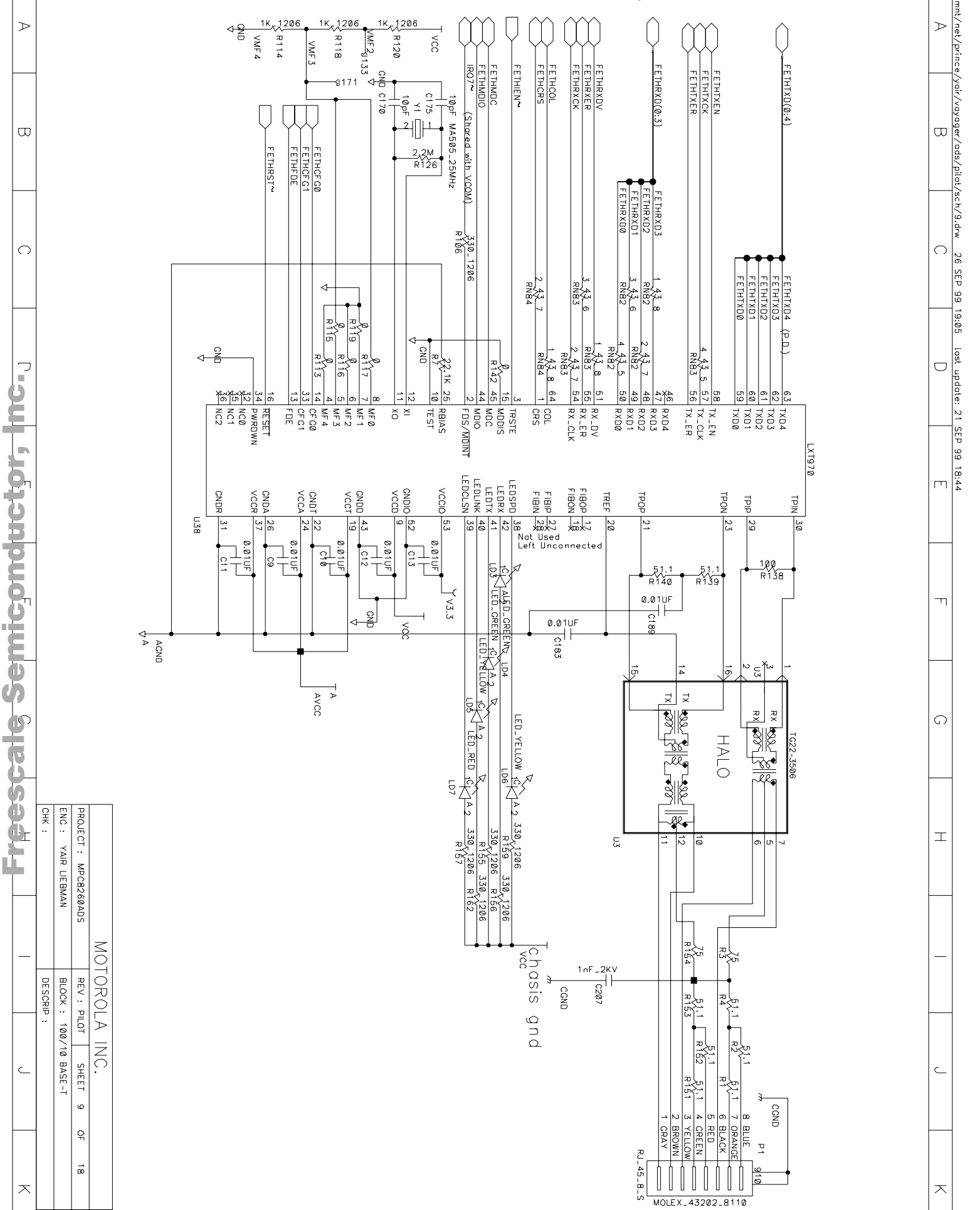


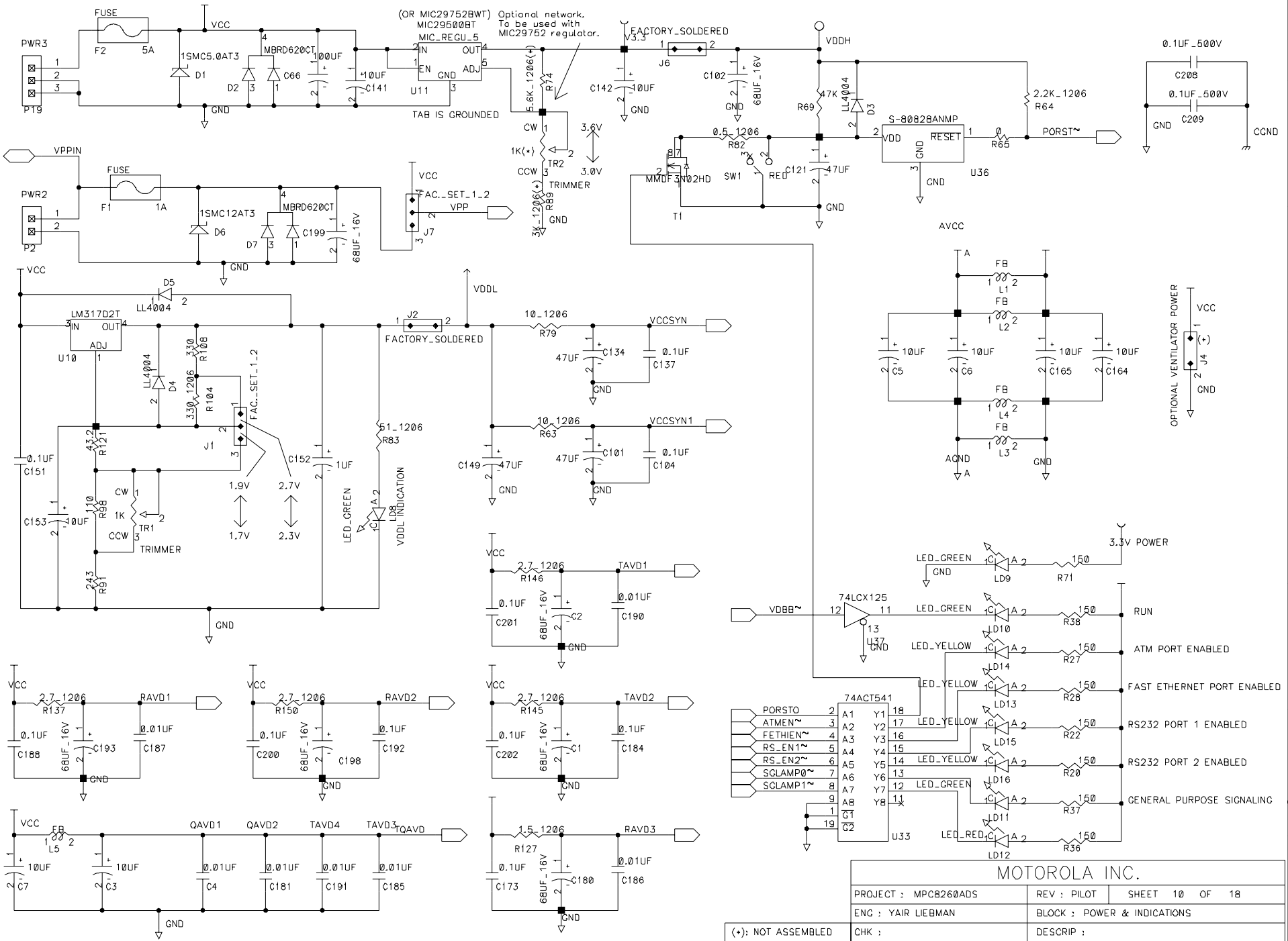
MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 6 OF 18
ENG : YAIR LIEBMAN	BLOCK : LOCAL BUS SDRAM	
CHK :	DESCRIP :	



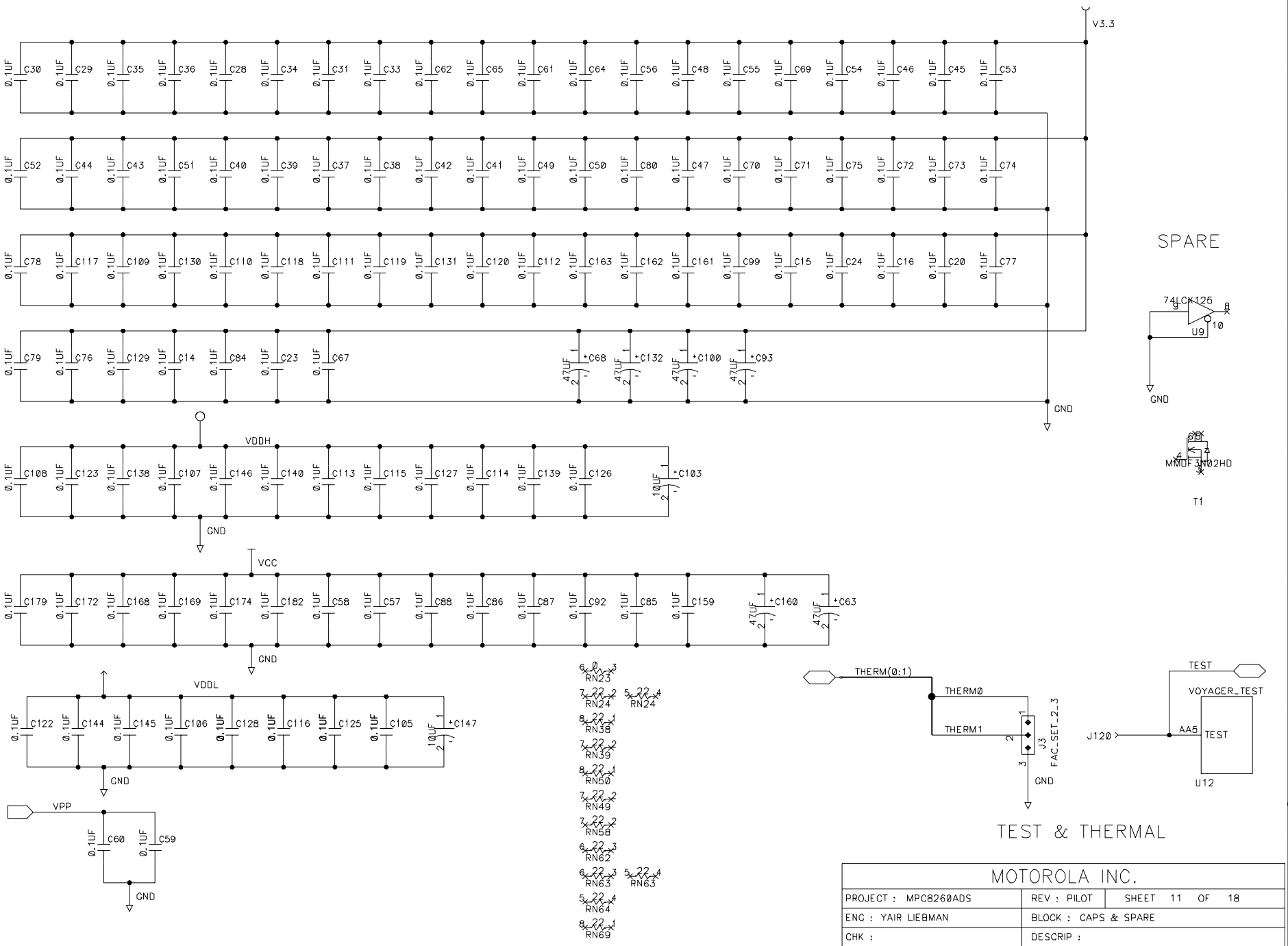
MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 7 OF 18
ENG : YAIR LIEBMAN	BLOCK : CPM & RS232	
CHK :	DESCRIP :	



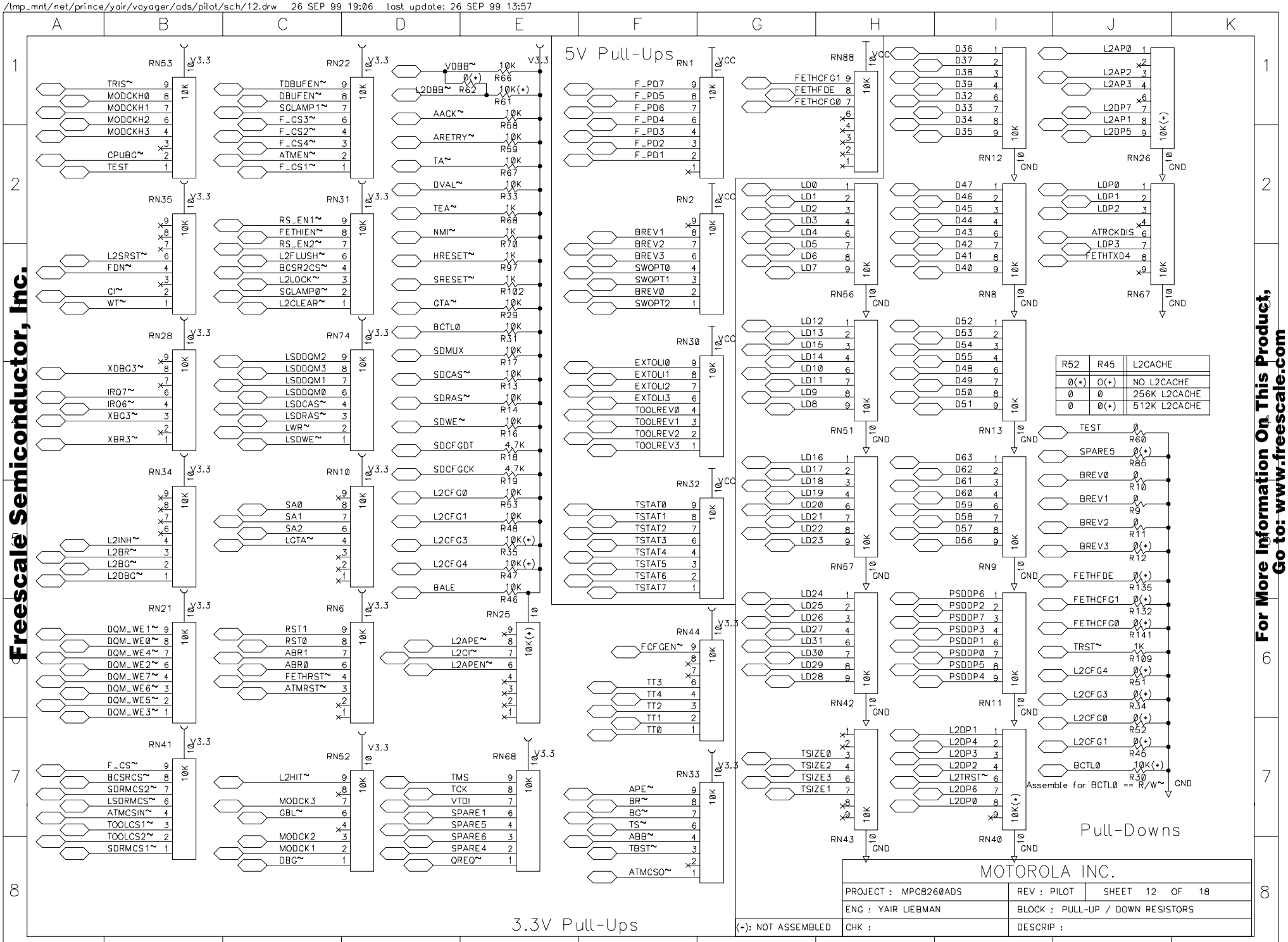


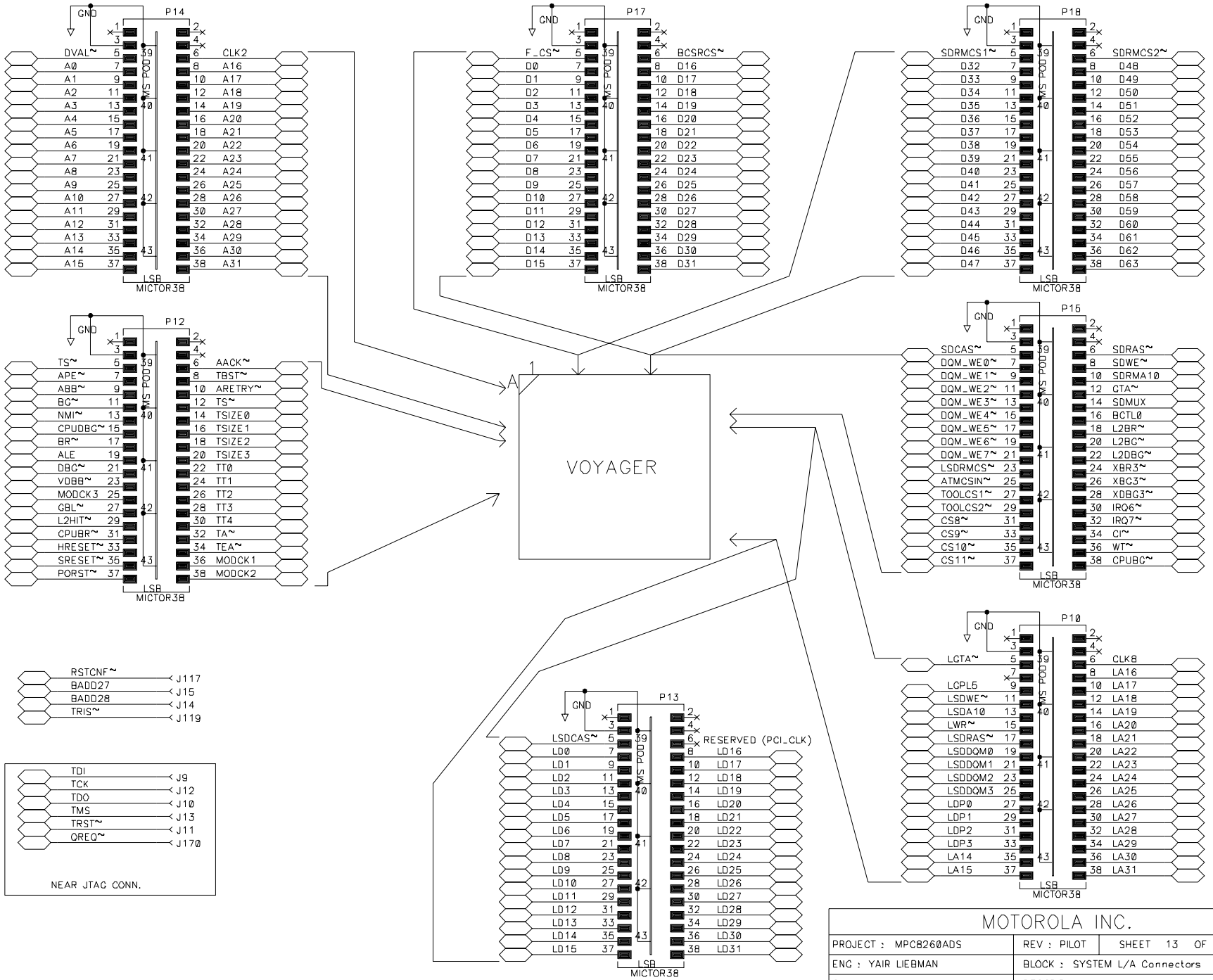


MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 10 OF 18
ENG : YAIR LIEBMAN	BLOCK : POWER & INDICATIONS	
(*): NOT ASSEMBLED	CHK :	DESCRIP :

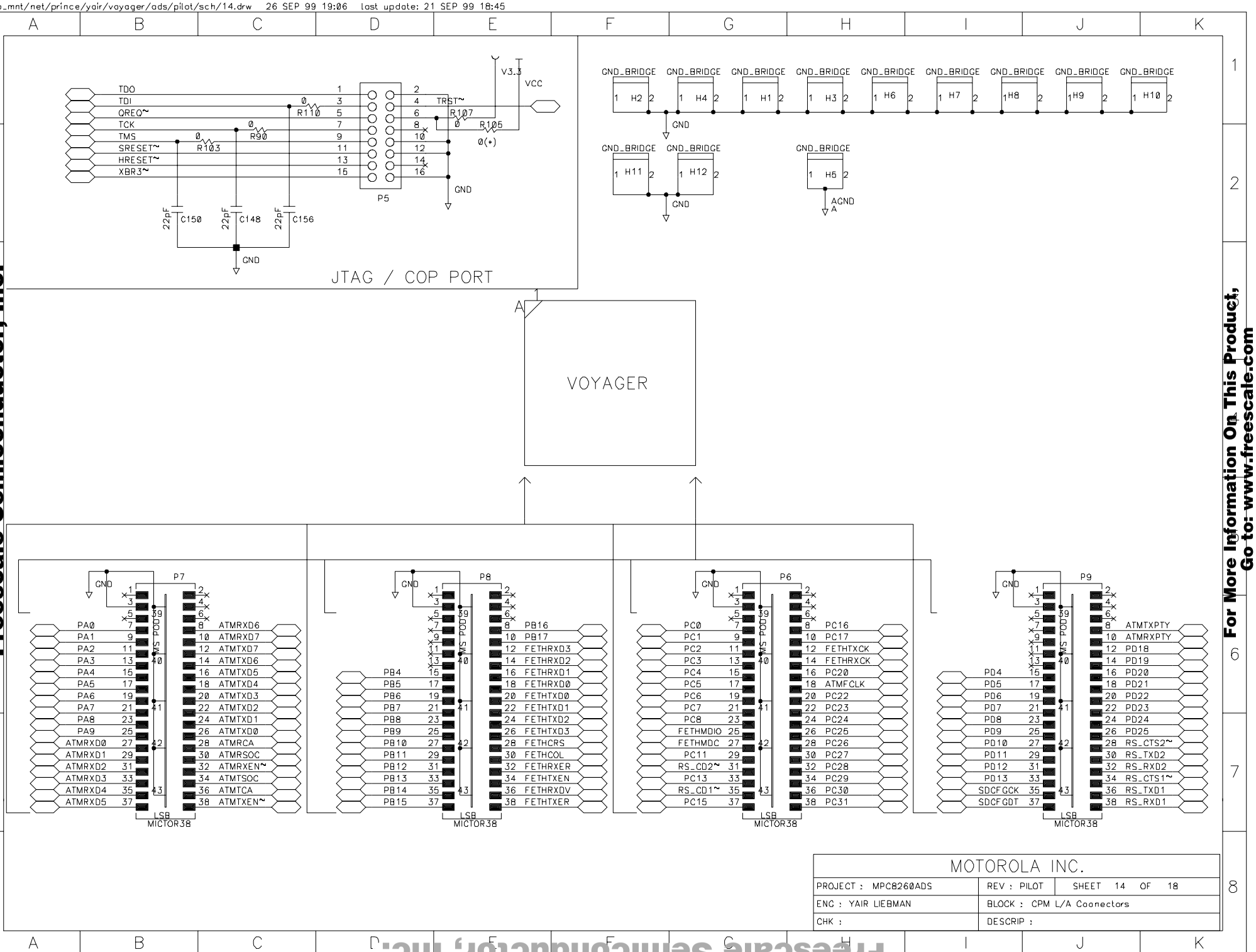


MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 11 OF 18
ENG : YAIR LIEBMAN	BLOCK : CAPS & SPARE	
CHK :	DESCRIP :	

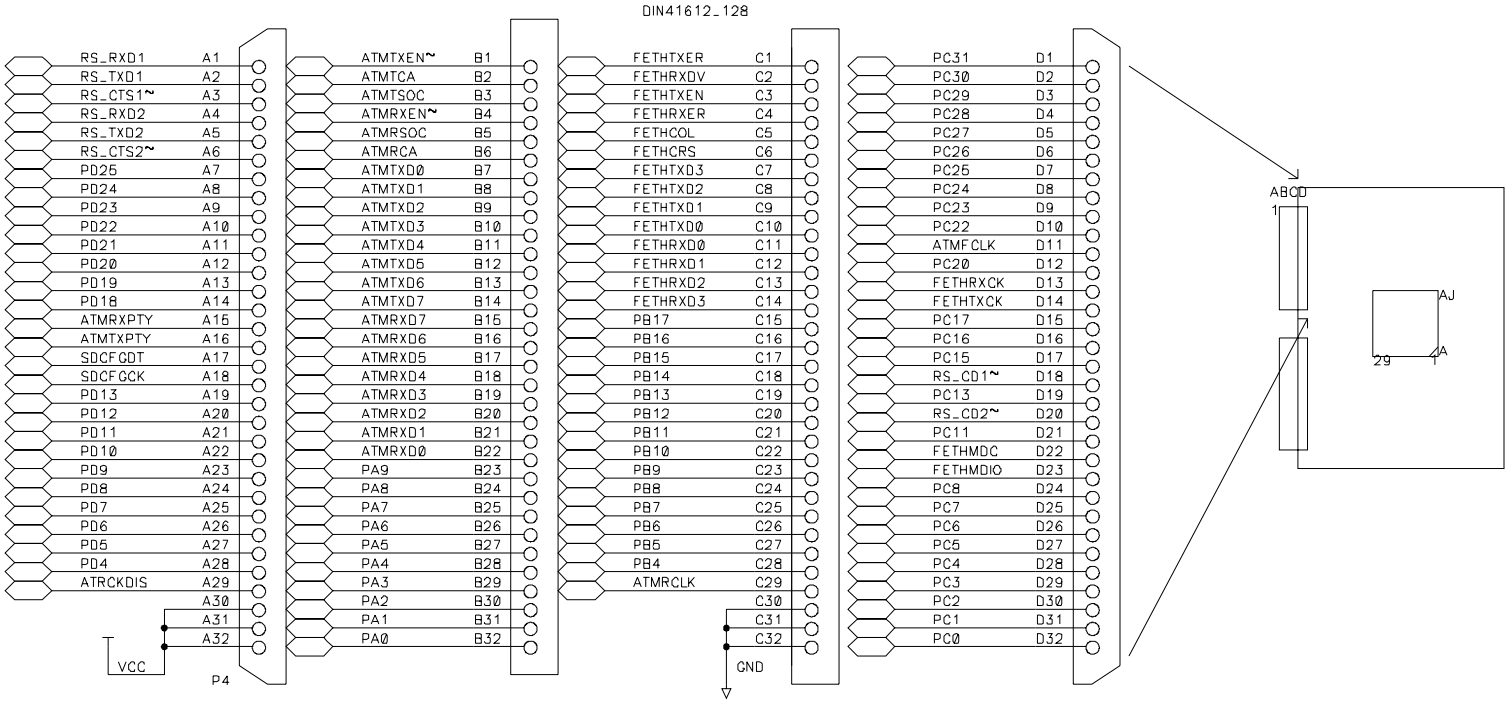




MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 13 OF 18
ENG : YAIR LIEBMAN	BLOCK : SYSTEM L/A Connectors	
CHK :	DESCRIP :	

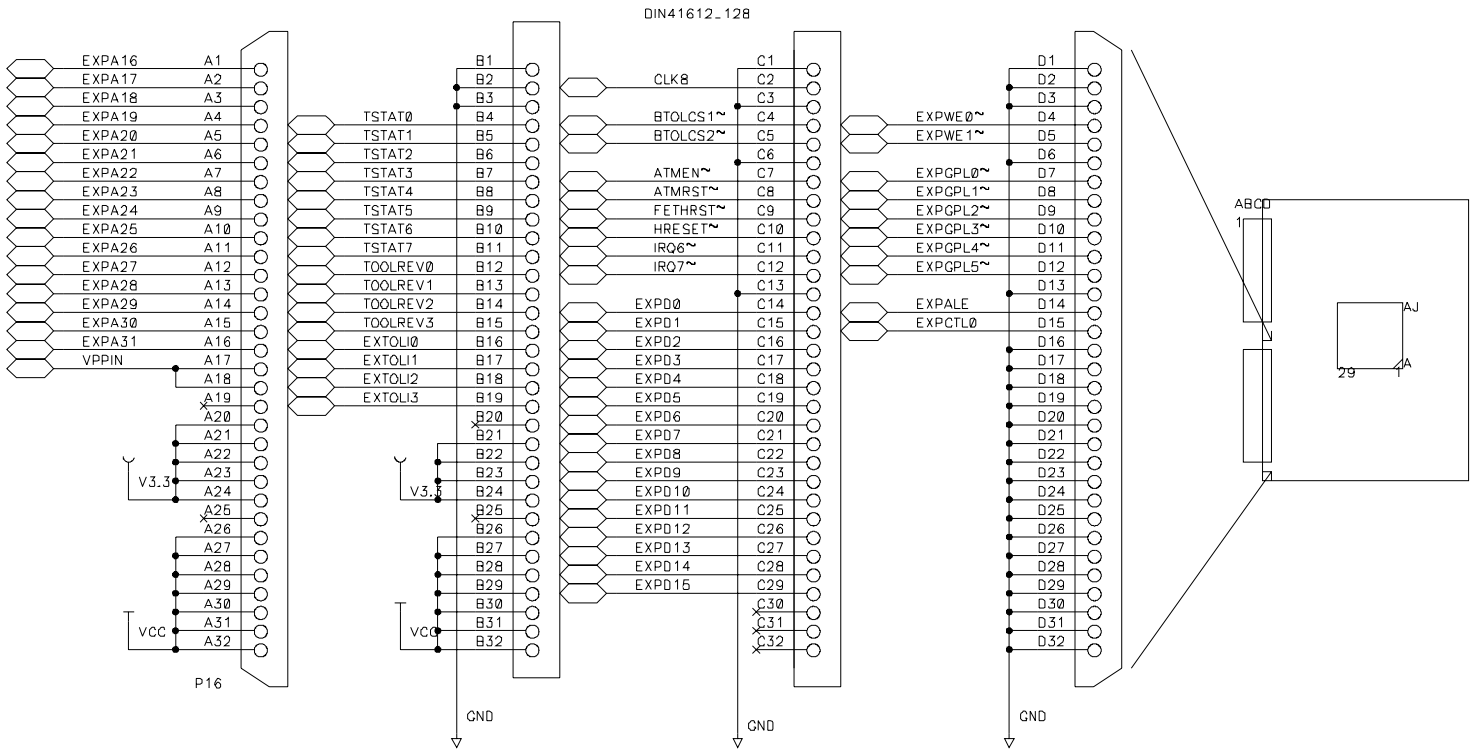


MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 14 OF 18
ENG : YAIR LIEBMAN	BLOCK : CPM L/A Coonectors	
CHK :	DESCRIP :	

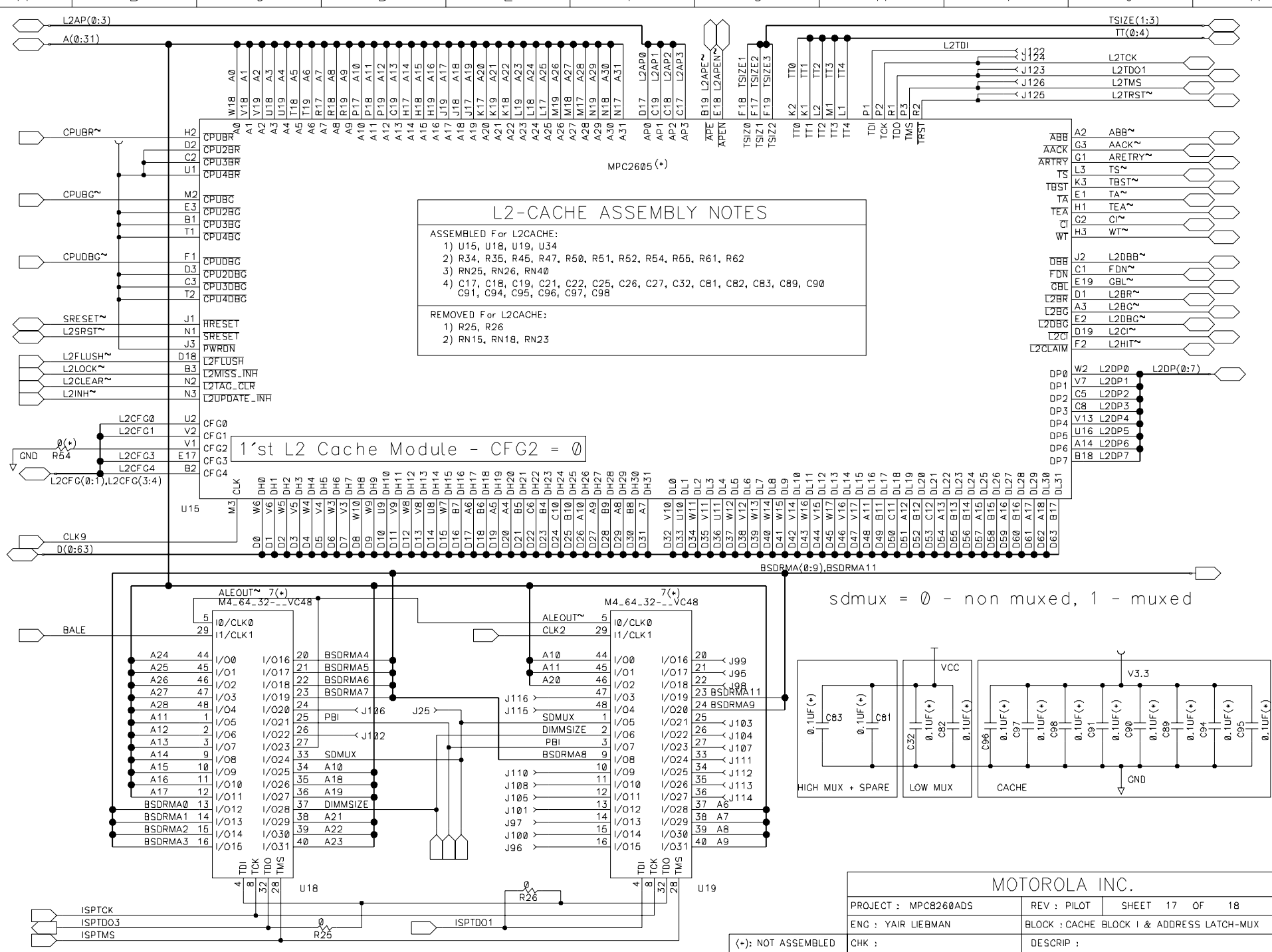


MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 15 OF 18
ENG : YAIR LIEBMAN	BLOCK : CPM EXPANSION CONNECTOR I	
CHK :	DESCRIP :	

Freescale Semiconductor, Inc.



MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 16 OF 18
ENG : YAIR LIEBMAN	BLOCK : CPM EXPANSION CONN. II	
CHK :	DESCRIP :	



L2-CACHE ASSEMBLY NOTES

ASSEMBLED For L2CACHE:

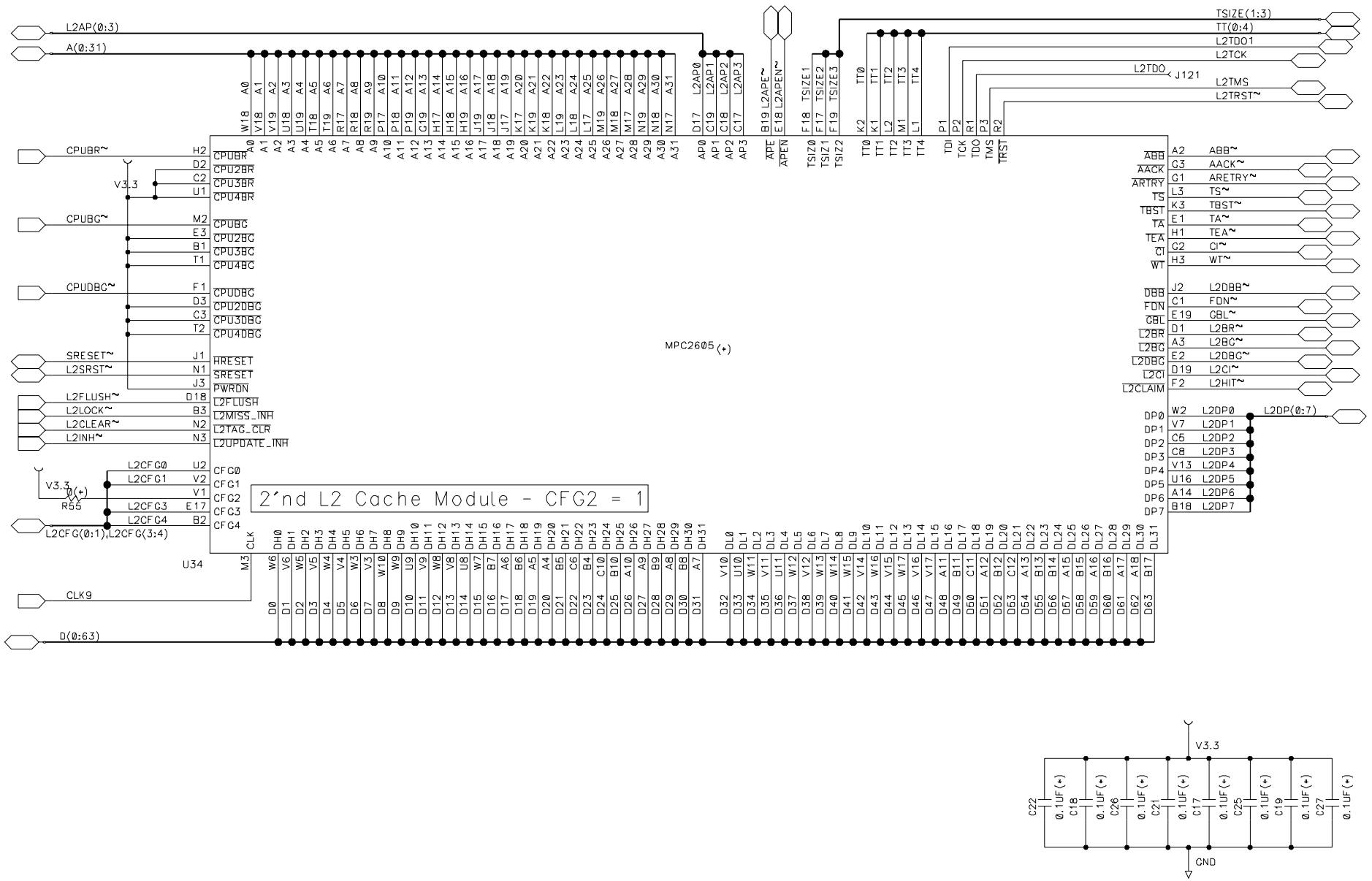
1) U15, U18, U19, U34
2) R34, R35, R45, R47, R50, R51, R52, R54, R55, R61, R62
3) RN25, RN26, RN40
4) C17, C18, C19, C21, C22, C25, C26, C27, C32, C81, C82, C83, C89, C90 C91, C94, C95, C96, C97, C98

REMOVED For L2CACHE:

1) R25, R26
2) RN15, RN18, RN23

(*): NOT ASSEMBLED

MOTOROLA INC.		
PROJECT : MPC8260ADS	REV : PILOT	SHEET 17 OF 18
ENG : YAIR LIEBMAN	BLOCK : CACHE BLOCK I & ADDRESS LATCH-MUX	
CHK :	DESCRIP :	



MOTOROLA INC.		
PROJECT : MPC8260ADS L2C	REV : PILOT	SHEET 18 OF 18
ENG : YAIR LIEBMAN	BLOCK : CACHE BLOCK II	
(*): NOT ASSEMBLED	CHK :	DESCRIP :