

**Freescale Semiconductor, Inc.**

*User's Manual*

*PQ2FADS-VR-UM*

*Revision 0.0*

*June 12, 2003*

*PQ2FADS-VR User's  
Manual*



**MOTOROLA**  
*intelligence everywhere™*

*digital dna™*

# PQ2FADS-VR

## User's Manual

**Freescale Semiconductor, Inc.**

© Motorola, Inc., 2003

**For More Information On This Product,  
Go to: [www.freescale.com](http://www.freescale.com)**

## Important Notice to Users

While every effort has been made to ensure the accuracy of all information in this document, Motorola assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document, its updates, supplements, or special editions, whether such errors are omissions or statements resulting from negligence, accident, or any other cause. Motorola further assumes no liability arising out of the application or use of any information, product, or system described herein: nor any liability for incidental or consequential damages arising from the use of this document. Motorola disclaims all warranties regarding the information contained herein, whether expressed, implied, or statutory, *including implied warranties of merchantability or fitness for a particular purpose*. Motorola makes no representation that the interconnection of products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting or license to make, use or sell equipment constructed in accordance with this description.

## Trademarks

This document includes these trademarks:

Motorola and the Motorola logo are registered trademarks of Motorola, Inc.

Motorola, Inc., is an Equal Opportunity / Affirmative Action Employer.

For an electronic copy of this book, visit Motorola's web site at <http://e-www.motorola.com/>

© Motorola, Inc., 2003; All Rights Reserved

---

## Table of Contents

---

**Section 1****General Information**

1.1	Introduction .....	1
1.2	Definitions, Acronyms, and Abbreviations .....	2
1.3	Related Documentation .....	2
1.4	Specifications .....	3
1.5	PQ2FADS-VR Features .....	3

**Section 2****Hardware Preparation and Installation**

2.1	Introduction .....	6
2.2	Unpacking Instructions .....	6
2.3	Hardware Preparation .....	6
2.3.1	Setting VDDL Level Range - P26 .....	7
2.3.2	Setting VDDL Supply Voltage Level .....	8
2.3.3	Setting MODCK(1:3) for PLLs Multiplication Factor - SW5 (#6 - #8) .....	8
2.3.4	Setting Hard - Reset Configuration Source - JP7 .....	10
2.3.5	Setting Boot Source .....	10
2.3.6	Setting MODCKH(0:3) - for PLLs Multiplication Factors .....	11
2.3.7	Setting PCI_MODCK - for PCI Bus Clock .....	11
2.3.8	Setting PCI_ARBITER - for PCI Mode Enabled .....	11
2.3.9	Setting PCI_DLL - for PCI Mode Enabled .....	11
2.3.10	60x Bus Parity Support .....	12
2.3.11	Clock-In Source selection .....	12
2.3.12	FCC2 Ethernet Port mode - MII/RMII .....	12
2.3.13	FCC3 Ethernet Port mode - MII/RMII .....	13
2.3.14	USB Speed selection .....	13
2.3.15	USB Mode selection .....	13
2.3.16	COP/JTAG Connection .....	13
2.3.17	Power On/Off Switch .....	14
2.4	Installation Instructions .....	14
2.4.1	Host Controlled Operation .....	14
2.4.2	Stand Alone Operation .....	15
2.4.3	COP/JTAG Connector - P16 .....	15
2.4.4	Terminal to PQ2FADS-VR RS-232 Connection .....	16
2.4.5	10/100-Base-T Ethernet Ports Connection .....	16
2.4.6	Memory Installation .....	17

2.4.6.1 Flash Memory SIMM Installation ..... 17

**Section 3  
Operating Instructions**

3.1 Introduction ..... 18

3.2 Controls and Indicators ..... 18

3.2.1 Power-On RESET Switch - SW1 ..... 18

3.2.2 ABORT Switch - SW3 ..... 18

3.2.3 SOFT RESET Switch - SW2 ..... 18

3.2.4 HARD RESET - Switches - SW2 & SW3 ..... 18

3.2.5 SW6 - Reset Configuration Switch ..... 19

3.2.6 SW7 - Software Options Switch ..... 19

3.2.7 P26 - VDDL Voltage Level Range Selection ..... 19

3.2.8 JP12 - IDDL Measurement ..... 19

3.2.9 JP6 - Thermal Sense Connector ..... 19

3.2.10 JP11 - IDDH Measurement ..... 20

3.2.11 JP13 - VPP Source Selector ..... 20

3.2.12 GND Bridges ..... 20

3.2.13 Power O.K. Indicator - LD1 ..... 21

3.2.14 12V Indicator - LD2 ..... 21

3.2.15 UTOPIA 16 Bit Indicator - LD15 ..... 21

3.2.16 UTOPIA Multi PHY Indicator - LD16 ..... 21

3.2.17 5V Indicator - LD3 ..... 21

3.2.18 3.3V Indicator - LD4 ..... 21

3.2.19 USB Power Indicator - LD17 ..... 21

3.2.20 -12V Indicator - LD5 ..... 21

3.2.21 RUN Indicator - LD6 ..... 21

3.2.22 ATM ON - LD7 ..... 21

3.2.23 Fast Ethernet Port 2 Enabled - LD10 ..... 22

3.2.24 Fast Ethernet Port 1 Enabled - LD8 ..... 22

3.2.25 RS232 Port 1 ON - LD11 ..... 22

3.2.26 Fast Ethernet Port 1 Full Duplex Indicator - LD18 ..... 22

3.2.27 RS232 Port 2 ON - LD12 ..... 22

3.2.28 Fast Ethernet Port 1 100Base-Tx Indicator - LD19 ..... 22

3.2.29 Ethernet Port 1 Tx/Rx Indicator - LD20 ..... 22

3.2.30 General Purpose Led 2 Indicator - LD13 ..... 22

3.2.31 Ethernet Port 1 LINK Indicator - LD21 ..... 22

3.2.32 Fast Ethernet Port 2 Full Duplex Indicator - LD22 ..... 22

3.2.33 General Purpose Led 1 Indicator - LD14 ..... 23

3.2.34 Fast Ethernet Port 2 100Base-Tx Indicator - LD23 ..... 23

3.2.35 USB Enabled Indicator - LD9 ..... 23

3.2.36 Ethernet Port 2 LINK Indicator - LD24 ..... 23

3.2.37 Ethernet Port 2 Tx/Rx Indicator - LD25 ..... 23

3.2.38 VDDL Indication - LD26 ..... 23

3.2.39 Parallel Port connection - LD27 ..... 23  
 3.2.40 External Debugger Connection Indicator - LD28 ..... 23

**Section 4  
 Functional Description**

4.1 Reset & Reset - Configuration ..... 24  
 4.1.1 Power - On ResetPQ2 ..... 24  
     4.1.1.1 Power - On Reset Configuration ..... 24  
 4.1.2 Hard Reset ..... 25  
     4.1.2.1 COP/JTAG Port Hard - Reset ..... 25  
     4.1.2.2 Manual Hard Reset ..... 25  
     4.1.2.3 Internal Sources Hard - Reset ..... 26  
     4.1.2.4 Hard Reset Configuration ..... 26  
 4.1.3 Soft Reset ..... 31  
     4.1.3.1 COP/JTAG Port Soft - Reset ..... 31  
     4.1.3.2 Manual Soft Reset ..... 31  
     4.1.3.3 Internal Sources Soft - Reset ..... 32  
 4.1.4 PCI Bus Reset ..... 32  
 4.2 Local Interrupter ..... 32  
     4.2.1 ABORT Interrupt ..... 32  
     4.2.2 ATM UNI Interrupt ..... 32  
     4.2.3 Fast Ethernet PHY Interrupt ..... 33  
     4.2.4 PCI Interrupt ..... 33  
 4.3 Clock Generator ..... 36  
     4.3.1 PQ2 Clock ..... 36  
     4.3.2 PCI Clock ..... 36  
 4.4 Bus Configuration ..... 37  
     4.4.1 Single PQ2 Mode ..... 37  
     4.4.2 60X Bus Mode ..... 37  
 4.5 Buffering ..... 38  
 4.6 Chip - Select Generator ..... 38  
 4.7 Synchronous Dram (60X Bus) ..... 39  
     4.7.1 SDRAM Programming ..... 41  
     4.7.2 SDRAM Refresh ..... 41  
     4.7.3 L2-Cache Support Influence On SDRAM Design ..... 41  
     4.7.4 SDRAM Error Correction Support ..... 42  
 4.8 Flash Memory SIMM ..... 43  
     4.8.1 Flash Programming Voltage ..... 44  
     4.8.2 Flash and L2Cache ..... 44  
 4.9 E2PROM Memory ..... 45  
 4.10 PCI Bus ..... 46  
 4.11 L2-CACHE Support ..... 47  
     4.11.1 L2 Cache Configuration & Control ..... 48  
 4.12 Communication Ports ..... 49

4.12.1	ATM Port .....	49
4.12.2	100/10 Base - T Ports .....	50
4.12.2.1	DM9161 Control .....	50
4.12.3	RS232 Ports .....	51
4.12.3.1	RS-232 Ports' Signal Description .....	51
4.12.4	USB Port .....	52
4.12.5	PC Parallel Port .....	52
4.13	Board Control & Status Register - BCSR .....	52
4.13.1	BCSR0 - Board Control - Status Register 0 .....	53
4.13.2	BCSR1 - Board Control - Status Register 1 .....	54
4.13.3	BCSR2 - Board Control - Status Register - 2 .....	55
4.13.4	BCSR3 - Board Control - Status Register 3 .....	59
4.13.5	BCSR4 - Board Control - Status Register 4 .....	59
4.13.6	BCSR5 and BCSR7- Board Control - Status Register 3 & 5 .....	60
4.14	COP/JTAG Port .....	61

**Section 5**  
**Memory Map and Initialization**

5.1	Memory Map .....	64
5.2	PQ2 Register Programming .....	68
5.2.1	System Initializations .....	69
5.2.2	Memory Controller Registers Programming .....	70

**Section 6**  
**Physical Properties**

6.1	Power Supply .....	74
6.1.1	5V Rail .....	76
6.1.2	3.3V Rail .....	76
6.1.3	5V Stand By Rail .....	76
6.1.4	VDDH Rail .....	76
6.1.5	VDDL Bus .....	76
6.1.6	12V Rail .....	76
6.1.7	-12V Rail .....	76
6.2	Connectors .....	77
6.2.1	ATX Power Connector .....	77
6.2.2	Fast Ethernet Port Connectors .....	77
6.2.3	ATM 155 Port Connection .....	77
6.2.4	RS232 PortS Connector .....	77
6.2.5	CPM Expansion Connector .....	77
6.2.6	COP/JTAG Port Connector .....	77
6.2.7	Logic Analyzer Connectors .....	78
6.2.8	Mach's In System Programming (ISP) Connector .....	78

6.2.9	PCI Connectors	78
6.2.10	System Expansion Connector	78
6.2.11	USB Connector	78
6.2.12	Parallel Port Connector	78
6.3	PCB Layout	78

**Section 7  
Support Information**

7.1	Interconnect signals	79
7.1.1	P1 - RS232 ports 1 and 2 Connectors	79
7.1.2	P3 and P4 - 100/10 - Base-T Ethernet port Connector	80
7.1.3	P16 - COP / JTAG Connector	80
7.1.4	P7 - CPM Expansion Connector	81
7.1.5	P9, P10, P11, P15, P17, P18, P19, P24, P28,P29, P30- Logic Analyzer MICTOR Connectors 88	
7.1.6	P10, P8, P9 - PCI Connectors	89
7.1.7	P5 - ATX Power Supply Connector	91
7.1.8	P21,P23,P25 - Mach/Lattice ISP Connector	91
7.1.9	P27 - System Expansion Connector	92
7.1.10	P2 - USB Connector	98
7.2	Programmable Logic Equations	98
7.2.1	U35 - BCSR Code	98
7.2.2	U41 - Power switch debounce	161
7.3	Bill Of Materials	166



---

List of Tables

---

1-1	PQ2FADS-VR specifications .....	3
2-1	. MODCK(1:3) Encoding .....	9
4-1	BCSR/FLASH Hard Reset Configuration Word .....	27
4-2	E2PROM Hard Reset Configuration Word .....	29
4-3	PCI Interrupt Register Description .....	34
4-4	PCI Interrupt Mask Register Description .....	35
4-5	PQ2FADS-VR Chip Select Assignments .....	39
4-6	100 MHz SDRAM Mode Register Programming .....	41
4-7	L2 Cache CFG(0:2) Settings .....	48
4-8	BCSR0 Description .....	54
4-9	BCSR1 Description .....	55
4-10	BCSR2 Description .....	56
4-11	FLASH Presence Detect (7:5) Encoding .....	57
4-12	FLASH Presence Detect (4:1) Encoding .....	57
4-13	EXTOOLI(0:3) Assignment .....	57
4-14	PQ2 Board Version Encoding .....	58
4-15	PQ2 Board Revision Encoding .....	58
4-16	External Tool Revision Encoding .....	58
4-17	L2 Cache Size Encoding .....	58
4-18	BCSR3 Description .....	59
4-20	PCI Board Present Signal Definitions .....	60
4-19	BCSR4 Description .....	60
4-21	BCSR5 to BCSR7 Description .....	61
4-22	COP/JTAG Port Signals Description .....	62
5-1	PQ2FADS-VR Memory Map - FLASH (or BCSR) as Boot Device .....	64
5-2	PQ2FADS-VR Memory Map - E2PROM as Boot Device .....	66
5-3	BCSR/FLASH Power On Reset Configuration .....	69
5-4	E2PROM Power On Reset Configuration .....	69
5-5	SIU REGISTERS' PROGRAMMING .....	70
5-6	Memory Controller Initializations For 100Mhz - FLASH as Boot Device .....	71
5-7	Memory Controller Initializations For 100Mhz - E2PROM as Boot Device .....	72
5-8	Memory Controller Initializations For 100Mhz .....	73
6-1	Expansion Connectors Maximum Current Consumption .....	75
6-2	Maximum Power Consumption Per Add-In Card .....	75
7-1	P1 Connector .....	79
7-2	P3,P4 - 100/10 Base-T Ethernet Connector .....	80
7-3	P16 - COP/JTAG Connector .....	80
7-4	P7 - CPM Expansion Connector .....	82
7-5	P12, P13, P14 - PCI Connectors .....	89
7-6	P5 - ATX Power Supply Connector .....	91

7-7	P25 - Lattice ISP Connector .....	92
7-8	P27 - System Expansion Connector .....	93
7-9	P2 - USB Connector .....	98
7-10	PQ2FADS-VR Bill of Materials .....	166

---

List of Figures

---

1-1PQ2FADS-VR Block Diagram 5 IX  
 2-1PQ2FADS-VR Top Side Part Location Diagram 7 IX  
 2-2VDDL Range Selection - P26 7 IX  
 2-3VDDL Trimmer - RP2 8 IX  
 2-4SW5 Description 9 IX  
 2-5Hard Reset Configuration Source Selection - JP7 10 IX  
 2-6SW6 Description 11 IX  
 2-7JP10 - 60x Parity Support Selection 12 IX  
 2-8Clock Source Selection 12 IX  
 2-9FCC2 Ethernet Mode Selection 13 IX  
 2-10FCC3 Ethernet Mode Selection 13 IX  
 2-11Host Controlled Operation Scheme - Command Converter 14 IX  
 2-12Host Controlled Operation Scheme - Parallel Port 15 IX  
 2-13Stand Alone Configuration 15 IX  
 2-14P16 - COP/JTAG Port Connector 16 IX  
 2-15P1A/P1B - RS232 Serial Port Connector 16 IX  
 2-16Flash Memory SIMM Insertion 17 IX  
 3-1SW7 - Description 19 IX  
 3-2JP6 - Therm Connector 20 IX  
 3-3JP13 - VPP Source Selection 20 IX  
 4-1PCI Host Configuration Registers 31 IX  
 4-2PCI Interrupt Routing Scheme 33 IX  
 4-3Main Clock Generator Scheme 36 IX  
 4-4PCI Clock Generator Scheme 37 IX  
 4-560x SDRAM Connection Scheme - No L2 Cache 40 IX  
 4-6SDRAM - 60x Bus Connection Scheme with L2 Cache 40 IX  
 4-760x SDRAM Data Parity Support 42 IX  
 4-8FLASH SIMM Connection Scheme 44 IX  
 4-9E2PROM Connection scheme 46 IX  
 4-10PCI Bus Scheme 47 IX  
 4-11RS232 Serial Ports Connector 51 IX  
 4-12Debug Station Connection Schemes 61 IX  
 4-13COP/JTAG Port Connector 62 IX  
 6-1PQ2FADS-VR Power Scheme 75 IX

1-1	PQ2FADS-VR Block Diagram .....	5
2-1	PQ2FADS-VR Top Side Part Location Diagram .....	7
2-2	VDDL Range Selection - P26 .....	7
2-3	VDDL Trimmer - RP2 .....	8
2-4	SW5 Description .....	9
2-5	Hard Reset Configuration Source Selection - JP7 .....	10

2-6	SW6 Description	11
2-7	JP10 - 60x Parity Support Selection	12
2-8	Clock Source Selection	12
2-9	FCC2 Ethernet Mode Selection	13
2-10	FCC3 Ethernet Mode Selection	13
2-11	Host Controlled Operation Scheme - Command Converter	14
2-12	Host Controlled Operation Scheme - Parallel Port	15
2-13	Stand Alone Configuration	15
2-14	P16 - COP/JTAG Port Connector	16
2-15	P1A/P1B - RS232 Serial Port Connector	16
2-16	Flash Memory SIMM Insertion	17
3-1	SW7 - Description	19
3-2	JP6 - Therm Connector	20
3-3	JP13 - VPP Source Selection	20
4-1	PCI Host Configuration Registers	31
4-2	PCI Interrupt Routing Scheme	33
4-3	Main Clock Generator Scheme	36
4-4	PCI Clock Generator Scheme	37
4-5	60x SDRAM Connection Scheme - No L2 Cache	40
4-6	SDRAM - 60x Bus Connection Scheme with L2 Cache	40
4-7	60x SDRAM Data Parity Support	42
4-8	FLASH SIMM Connection Scheme	44
4-9	E2PROM Connection scheme	46
4-10	PCI Bus Scheme	47
4-11	RS232 Serial Ports Connector	51
4-12	Debug Station Connection Schemes	61
4-13	COP/JTAG Port Connector	62
6-1	PQ2FADS-VR Power Scheme	75

## **General Information**

### **1.1 Introduction**

This document is an operation guide for the PQ2FADS-VR board. It contains operational, functional and general information about the PQ2FADS-VR. This board is meant to serve as a platform for s/w and h/w development for the POWER QUICC II family of processors in a TEPBGA package (516 Pins for Hip4 and Hip7). 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 POWER QUICC II processors.

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

---

1. Either on or off-board.

## 1.2 Definitions, Acronyms, and Abbreviations

PQ2FADS-VR	PowerQUICC II Family ADS Board For TEPBGA Package (516 pins in a plastic package)
PQ2	PowerQuicc 2
MPC8250	PowerQuicc 2 Hip4 in TEPBGA package
MPC8275	PowerQuicc 2 Hip7 in TEPBGA package
VOYAGER	MPC8260 - PowerQUICC 2
PPC	PowerPC
PCI	Peripheral Components Interconnect
USB	Universal Serial Bus
CPM	Communication Processor Module
SDRAM	Synchronous Dynamic Random Access Memory
VADS	Voyager Application Development System
Kbyte	1024 bytes
LSB	Least Significant Byte
lsb	least significant bit
Mbyte	1048576 bytes
DIMM	Dual In-line Memory Module
SIMM	Single In-line Memory Module
TBD	To Be Defined
UPM	User Programmable Machine
EVB	Evaluation Board
GPCM	General Purpose Chip-select Machine
GPL	General Purpose Line
BCSR	Board Control and Status Register
FLASH	Non volatile reprogrammable memory.
ZIF	Zero Input Force
BGA	Ball Grid Array
ADI	Application Development Interface.
COP	Common On-chip Processor
SAR	Segmentation And Reassembly
UTOPIA	Universal Test & Operations Interface for ATM

## 1.3 Related Documentation

- MPC8260, MPC8250, MPC8275 - User's Manual.
- VADS Users' Manual.
- PMC-SIERRA 5384 Long Form Data Sheet
- DM9161 (by DAVICOM) Data Sheet

## 1.4 Specifications

The PQ2FADS-VR specifications are given in Table 1-1.

**Table 1-1. PQ2FADS-VR specifications**

CHARACTERISTICS	SPECIFICATIONS
Power requirements (no other boards attached)	+5Vdc @ TBD A (Typ.), TBD A (Max.) +3.3Vdc @ TBD A (Typ.), TBD A (Max.) +12Vdc - @TBD A Max. -12Vdc - @TBD A Max.
Microprocessor	MPC8250 running @ 66 MHz Bus Clock Frequency. MPC8275 running @ up to 100 MHz Bus Clock Frequency.
Addressing Total address range on PPC Bus: Total address range on Local Bus:	4 Giga Bytes (32 address lines) 256 KBytes External (18 address lines) 4 Giga Bytes Internal (32 address lines internal decoding)
Flash Memory SIMM (PPC Bus) Synchronous Dynamic RAM DIMM (PPC Bus)	8 MByte, 32 bits wide expandable to 32 MBytes 32 MByte, 64 bits wide with optional parity.
Synchronous DRAM On Local Bus	8 MBytes, 32 bit wide with optional parity.
Operating temperature	0°C - 70°C (room temperature)
Storage temperature	-25°C to 85°C
Relative humidity	5% to 90% (non-condensing)
Dimensions: Length Width Thickness	12" (305 mm) 9" (229 mm) 0.063" (1.6 mm)

## 1.5 PQ2FADS-VR Features

- Supports MPC8250 (Hip4) and MPC8275 (Hip7) processors.
- 64 bit PowerQUICC II Communication Processor, running @ up to 100MHz external bus frequency.
- 32 MByte Synchronous Dram (soldered on-board), residing on 60X bus (PBI mode) with optional parity support, controlled by SDRAM machine 1. Optional address Latch - Multiplexer is available if L2 cache module is assembled.
- Optional 1/2 MByte L2-Cache on-board using 2 MPC2605 Look-Aside cache modules.
- 8 MByte, 80 pin Flash SIMM, buffered from 60X bus. Support for upto 32 MByte, controlled by GPCM, 5V/12V Programmable, with Automatic Flash SIMM identification, via BCSR. Support for both On and OFF SIMM Flash reset.

- 5V/12V VPP (in-circuit programming voltage) for Flash SIMM - jumper selectable.
- 8 KBytes E<sup>2</sup>PROM, buffered from the 60x bus, controlled by the GPCM.
- Board Control & Status Register - BCSR, Controlling Boards' Operation.
- On-board COP/JTAG connector.
- On-board logic to support direct connection to standard Parallel Port (EPP/SPP modes) in Desk Top PC for debug purposes using MetroWerks CodeWarrior tools.
- Power-On Reset Option via JTAG.
- Programmable Power-On Reset and Hard-Reset Configuration via E<sup>2</sup>PROM or via Flash memory for the PQ2 core.
- PCI Local Bus is PCI Standard 2.2 compliant.
- 3 PCI slots are available to host up to 3 masters/targets cards @ 3.3V only - arbitration is supported by the on-chip Arbiter.
- PCI bus supports 25 - 66 MHz @ 3.3V devices (determined by the user).
- Simple generic Interrupt Controller to handle the PCI interrupts (4 in each PCI slot).
- Module Enable Indications for all on-board modules.
- High density (MICTOR) Logic Analyzer connectors, carrying all 60x, local bus and CPM signals, for fast logic analyzer connection.
- 155 Mbps ATM UNI on FCC1 with Optical I/F, connected to the PQ2 via UTOPIA Level 2 I/F supporting 8/16 bit in single/multi PHY, using the PMC-SIERA 5384.
- Two 100/10-Base-T Ports on FCC2 and FCC3 with T.P. I/F, MII/RMII controlled, using Davicom DM9161.
- USB Port, USB 1.1 Standard Compliant, using Philips PDIUSBP11 USB transceiver. USB Port is with shutdown option and speed selectable - BCSR controlled.
- 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 the expansion connectors.
- Dedicated PQ2 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.
- External Tools' identification & status read capability, via BCSR.
- Separate Power-On Reset Push - Button, Soft / Hard<sup>1</sup> Reset Push - Button and ABORT Push - Button.
- ATX Power Supply.
- Multi-Range PQ2 internal logic operation voltage - selectable by jumper between three ranges - 1.3V to 1.7V for MPC8275 (Hip7), 1.7V to 1.9V for PQ2 (Hip4) or 2.3V to 2.7V.
- Software Option Switch provides 8 S/W options via BCSR.

---

1. Hard reset is applied by depressing BOTH Soft Reset & ABORT buttons.

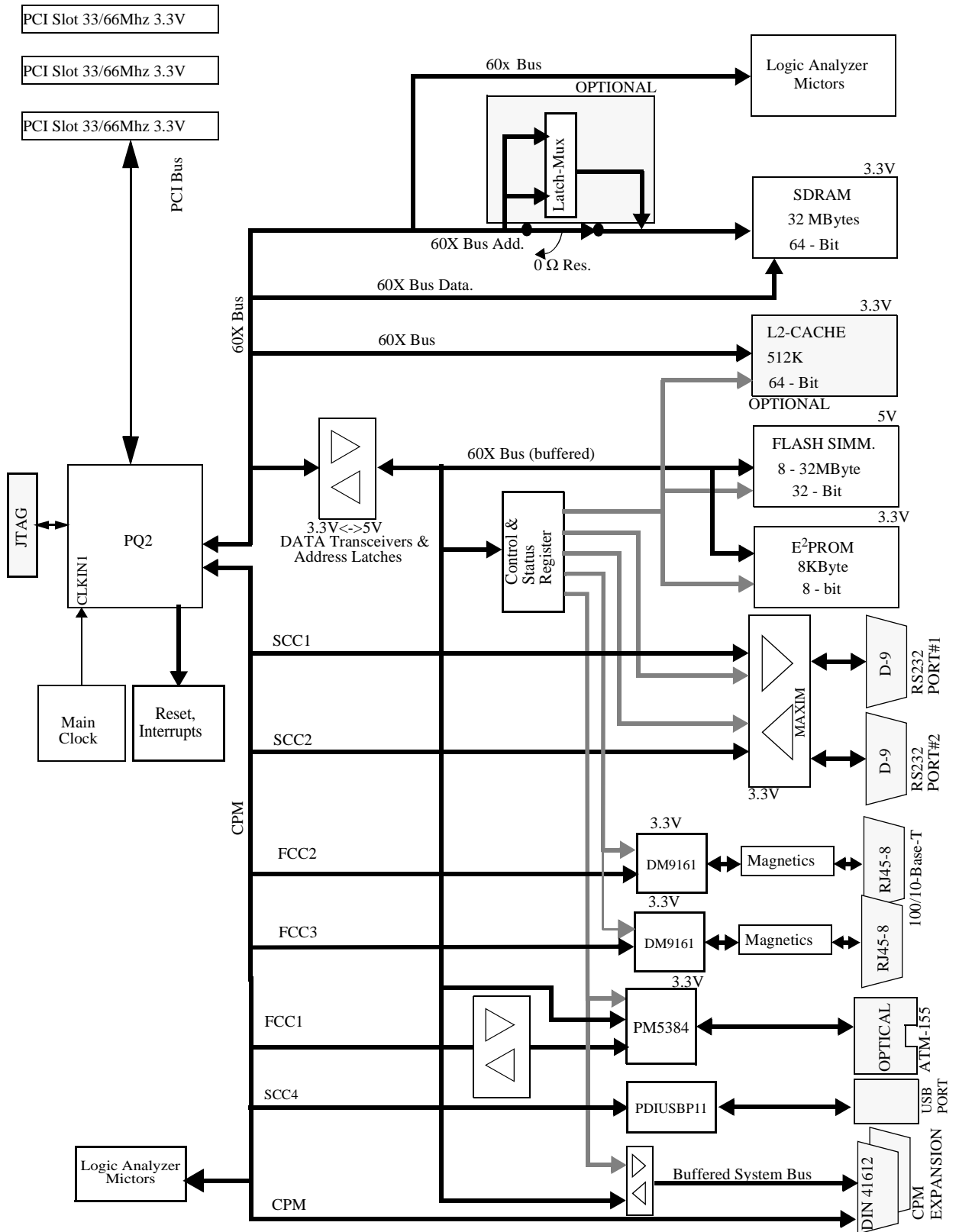


Figure 1-1. PQ2FADS-VR Block Diagram

## Hardware Preparation and Installation

### 2.1 Introduction

This chapter provides unpacking instructions, hardware preparation, and installation instructions for the PQ2FADS-VR.

### 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 PQ2FADS-VR 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](#). 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:

- PQ2's Internal Logic Supply Level Range Via connector P26.
- PQ2's Internal Logic Supply Level within range (VDDL) Via potentiometer RP2.
- PQ2's MODCK(1:3). Determining Core's and CPM's PLLs multiplication factor via dip-switches SW5(#6 - #8).
- PQ2's Hard Reset Configuration word Source - BCSR or Memory (FLASH/EEPROM) - via

jumper JP7.

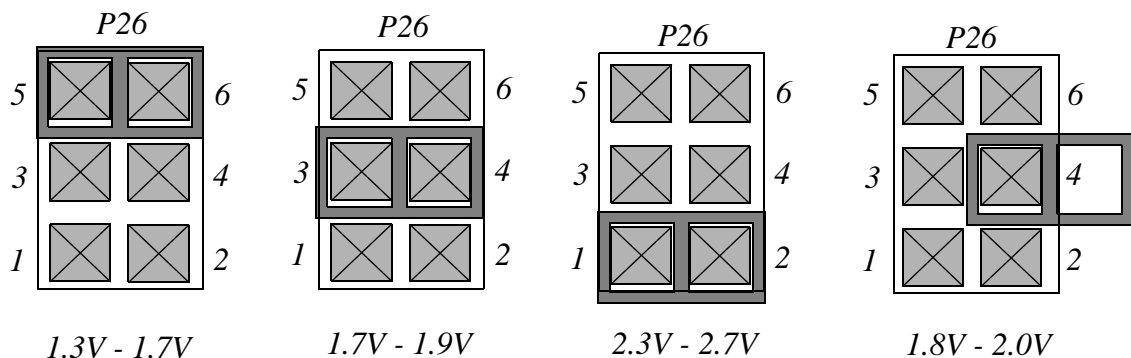
- PQ2's Boot code Source - EEPROM/FLASH - via dip-switch SW6(1).
- PQ2's MODCKH(0:3) via SW5(1-4).
- PQ2's PCI\_MODCK via SW5(5).
- PQ2's  $\overline{\text{PCI\_ARBITER}}$  via SW6(2).
- PQ2's PCI\_DLL via SW6(3).
- PQ2's 60x Bus parity support On/Off - via jumper JP10.
- Clock-In source - External or On-Board clock oscillator - JP9.
- FCC2 and FCC3 MII/RMII modes - via jumpers JP2 and JP3 respectively.
- USB speed (12Mbits/s or 1.5Mbits/s) and mode (Host or Slave) - software controlled in BCSR.
- PQ2's COP/JTAG connection - COP/JTAG connector (P16) or direct connection to PC parallel port (P31) - selected automatically by connecting parallel cable.
- ATX Power Supply On/Off Switch - via SW4.

**Figure 2-1.** PQ2FADS-VR Top Side Part Location Diagram

### 2.3.1 Setting VDDL Level Range - P26

To support all revisions of the PQ2, provisions are taken to provide necessary voltage levels on VDDL, to match the process by which the PQ2 is manufactured. Via P26, four voltage level ranges are provided (P26 setting options are shown in [Figure 2-2](#)):

1. When a jumper is placed between positions **1 - 2** of P26, a level range of **2.3V to 2.7V** on VDDL is selected. This level matches the specification for the MPC8260 (Hip3).
2. When a jumper is placed between positions **3 - 4** of P26, a level range of **1.7V to 1.9V** on VDDL is selected. This level matches the specification for the MPC8250 (Hip4).
3. When a jumper is placed between positions **5 - 6** of P26, a level range of **1.3V to 1.7V** is selected for VDDL. This level matches the specification for the MPC8275 (Hip7).
4. When a jumper is **misplaced** for P26, a level range of **1.8V to 2.0V** is selected for VDDL.



**Figure 2-2.** VDDL Range Selection - P26

### WARNING

P26 is Factory Set according to the revision of PQ2 with which it is assembled. Prior to changing a PQ2 device, Extra Care should be taken with P26 setup. If a selected Voltage Range is above the specification for the newly inserted PQ2, PERMANENT DAMAGE might be inflicted to the device.

P26 selects only a range of Voltage levels on VDDL. The actual level is selected by RP2. See next paragraph.

### 2.3.2 Setting VDDL Supply Voltage Level

After VDDL's Voltage Level Range is selected via P26, the actual level of VDDL is tuned via RP2. VDDL may be measured upon JP12, 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 RP2. Rotating RP2 CCW will increase VDDL voltage up to range-high, while rotating it CW, will decrease VDDL down to range-low. LD26 provides visual indication for VDDL level, it illuminates brighter with rise of VDDL. VDDL change Vs. RP2's rotation direction is shown in Figure 2-3.:

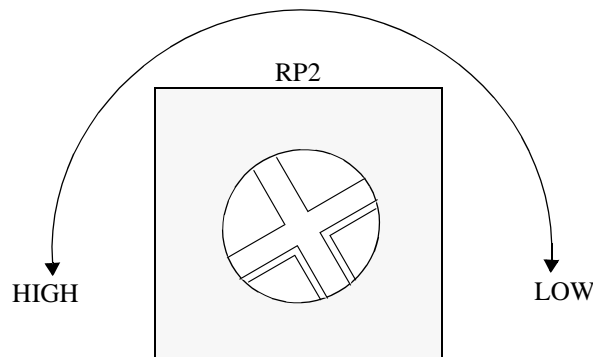


Figure 2-3. VDDL Trimmer - RP2

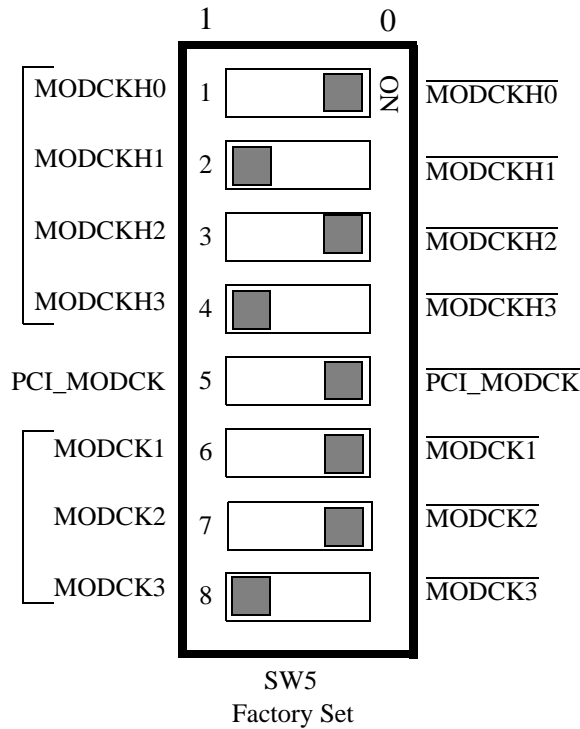
### WARNING

While in higher ranges of VDDL and higher ranges of internal operation frequencies, the PQ2 might require some sort of COOLING measures to be taken. Failure in doing so, might result in PERMANENT DAMAGE inflicted to the PQ2.

### 2.3.3 Setting MODCK(1:3) for PLLs Multiplication Factor - SW5 (#6 - #8)

After (1K cycles) the negation of the Power On Reset signal, the PQ2 samples the 7 MODCK

lines - the lower 3 on MODCK(1-3) and the upper four - MODCKH(0:3) field, to establish the multiplication factors of the CPM's and Core's PLLs. The levels on **MODCK(1:3)** lines are set using **SW5**, 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'). SW5 is shown in Figure 2-4., while the various combinations for SW5 (#6 - #8) and their associated MODCK(1:3) values are shown in Table 2-1..



**Figure 2-4. SW5 Description**

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

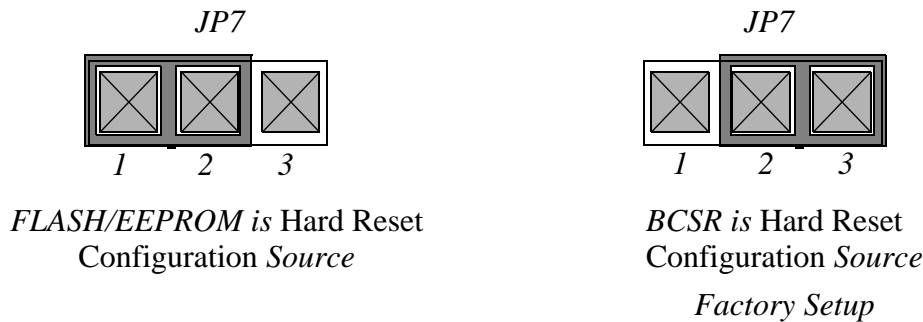
MODCK(1:3)	Switch 6	Switch 7	Switch 8
0	ON	ON	ON
1	ON	ON	OFF
2	ON	OFF	ON
3	ON	OFF	OFF
4	OFF	ON	ON
5	OFF	ON	OFF
6	OFF	OFF	ON
7	OFF	OFF	OFF

### 2.3.4 Setting Hard - Reset Configuration Source - JP7

The Boot sequence which starts when  $\overline{\text{HRESET}}$  is asserted, may be from two sources:

1. BCSR (default Hard-Reset Configuration Word -  $\overline{\text{CS0}}$  is assumed to be assigned to the FLASH)
2. Memories (FLASH/EEPROM - user controlled Hard-Reset Configuration Word)

When a jumper is placed between positions 1 - 2 of JP7, the Hard Reset Configuration source is a memory (FLASH/EEPROM) as configured by switch SW6-1. When a jumper is set between positions 2 - 3 of JP7, the Hard Reset Configuration source is the BCSR. See Figure 2-5..



**Figure 2-5.** Hard Reset Configuration Source Selection - JP7

### 2.3.5 Setting Boot Source

The Hard - Reset configuration word<sup>1</sup>, read by the PQ2 while  $\overline{\text{HRESET}}$  is asserted, may be taken from three sources:

1. Flash Memory SIMM
2. EEPROM
3. BCSR

For additional information as for the contents of the Hard-Reset configuration word see 4.1.2.4 "Hard Reset Configuration" on page 26.

SW6#1 actually assigns  $\overline{\text{CS0}}$  to the FLASH (default when booting from the BCSR) or to the EEPROM. When SW6 #1 is **OFF**, the Hard Reset configuration word is taken from **EEPROM**,

<sup>1</sup>.In fact 8 Hard-Reset configuration words are read by a configuration master, however only the first is relevant for a single PQ2.

when it is **ON**, the Hard Reset configuration word is taken from the *Flash SIMM*. See [Figure 2-6](#).

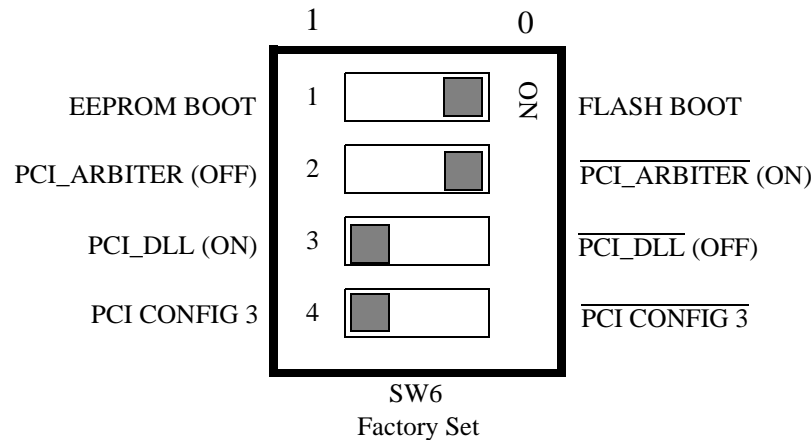


Figure 2-6. SW6 Description

### 2.3.6 Setting MODCKH(0:3) - for PLLs Multiplication Factors

Since the PCI mode in the PQ2 (Hip4 and Hip7) is enabled, the MODCKH(0:3) lines are taken from SW5(#1 - #4) and the MODCKH(0:3) bits in the Hard Reset Configuration Word are ignored. SW5(#1 - #4) set the upper 4 bits of the MODCK field during Hard Reset Configuration acquisition. When an individual switch of SW4 #1 - #4 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') (see [Figure 2-4](#)).

### 2.3.7 Setting PCI\_MODCK - for PCI Bus Clock

The settings of this line, determines the frequency of the PCI bus (when the PQ2 is in PCI mode). When PCI\_MODCK is set low, the PCI bus frequency is set by the MODCK lines. When set high, the PCI bus frequency is half of what is set by the MODCK lines. When switch SW6 #5 is at the **OFF** position, its corresponding PCI\_MODCK line is pulled-**high** ('1' - enabled), while when at the **ON** position, pulled-**down** ('0 - disabled') (see [Figure 2-4](#)).

### 2.3.8 Setting $\overline{\text{PCI\_ARBITER}}$ - for PCI Mode Enabled

The settings of this line, determines the operation of the PCI Arbiter (when the PQ2 is in PCI mode). When  $\overline{\text{PCI\_ARBITER}}$  is set low, the PCI Arbiter in the PQ2 is enabled. When set high, the PCI Arbiter is disabled and an external arbiter can be used. When switch SW6 #2 is at the **OFF** position, its corresponding  $\overline{\text{PCI\_ARBITER}}$  line is pulled-**high** ('1' - disabled), while when at the **ON** position, pulled-**down** ('0' - enabled) (see [Figure 2-6](#)).

### 2.3.9 Setting PCI\_DLL - for PCI Mode Enabled

The settings of this line, determines the operation of the DLL for PCI Mode enabled. When PCI Mode is enabled, the DLL must be enabled. When PCI\_DLL is set low, the DLL is disabled. When set high, the DLL is enabled. When switch SW6 #3 is at the **OFF** position, its corresponding PCI\_DLL line is pulled-**high** ('1' - **enabled**), while when at the **ON** position, pulled-**down** ('0' - **disabled**) (see [Figure 2-6](#)).

### 2.3.10 60x Bus Parity Support

Error correction (parity) on the 60x bus transactions is optional by setting JP10. Since the 8 data parity pins are muxed with other functions, Bus Mux is used to connect the data parity pins to the SDRAM device. When a jumper is placed between positions **1 - 2** of JP10, the 60x parity support is **disabled**. When a jumper is placed between positions **2-3** of JP10, the 60x parity support is **enabled**. See [Figure 2-7](#).



Figure 2-7. JP10 - 60x Parity Support Selection

### 2.3.11 Clock-In Source selection

The main clock source can be selected between an external (off-board) source by connecting to P22 or an on-board clock oscillator. The selection is done by setting JP11. When a jumper is placed between positions **1 - 2** of JP9, the **external** clock source is **enabled**. When a jumper is placed between positions **2-3** of JP9, the **on-board** clock oscillator is **enabled**. See [Figure 2-8](#).

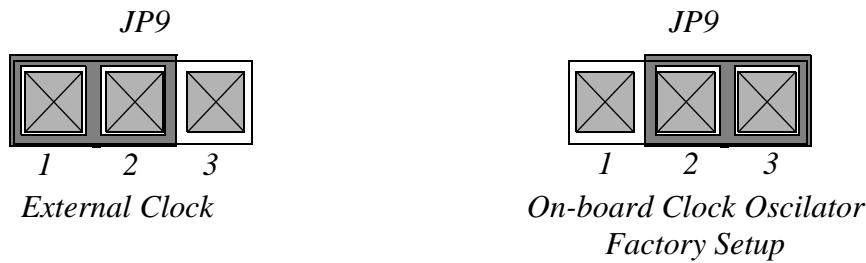


Figure 2-8. Clock Source Selection

### 2.3.12 FCC2 Ethernet Port mode - MII/RMII

The Ethernet PHY on FCC2 is set by default to 100Base-Tx Full Duplex and can be configured to operate in MII or RMII interface. The selection is done by setting JP2. When a jumper is placed between positions **1 - 2** of JP2, the **MII** interface is **enabled**. When a jumper is placed between positions **2-3** of JP2, the **RMII** interface is **enabled**. See [Figure 2-9](#).



**Figure 2-9.** FCC2 Ethernet Mode Selection

**NOTE:** For the mode change to take place, the setting of JP2 should be done while the board is powered-off.

### 2.3.13 FCC3 Ethernet Port mode - MII/RMII

The Ethernet PHY on FCC3 is set by default to 100Base-Tx Full Duplex and can be configured to operate in MII or RMII interface. The selection is done by setting JP3. When a jumper is placed between positions 1 - 2 of JP3, the **MII** interface is **enabled**. When a jumper is placed between positions 2–3 of JP3, the **RMII** interface is **enabled**. See [Figure 2-10](#).



**Figure 2-10.** FCC3 Ethernet Mode Selection

**NOTE:** For the mode change to take place, the setting of JP3 should be done while the board is powered-off.

### 2.3.14 USB Speed selection

The USB port supports two speeds - 12Mbits/s and 1.5Mbits/s. The selection is software controlled in the BCSR. At power-up, the **default** selection is **12Mbits/s**.

### 2.3.15 USB Mode selection

The USB port supports two modes - Host and Slave. The selection is software controlled in the BCSR. At power-up, the **default** selection is **Host**.

### 2.3.16 COP/JTAG Connection

There are two options to connect to the COP port of the PQ2 - COP/JTAG connector or a Parallel port (of a PC). The COP/JTAG connector requires a command converter while the second option

connects directly to the parallel port of a PC and eliminates the need for one. The selection is done automatically - if a cable is connected to the parallel port in a PC then this connection has the priority over the COP/JTAG connector.

### 2.3.17 Power On/Off Switch

The Power-On or Off is done by switching SW4.

## 2.4 Installation Instructions

When the PQ2FADS-VR 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 PQ2FADS-VR 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. There are two options to connect to the COP port:

1. The host is connected to the board by a COP controller (command converter) provided by a third party.

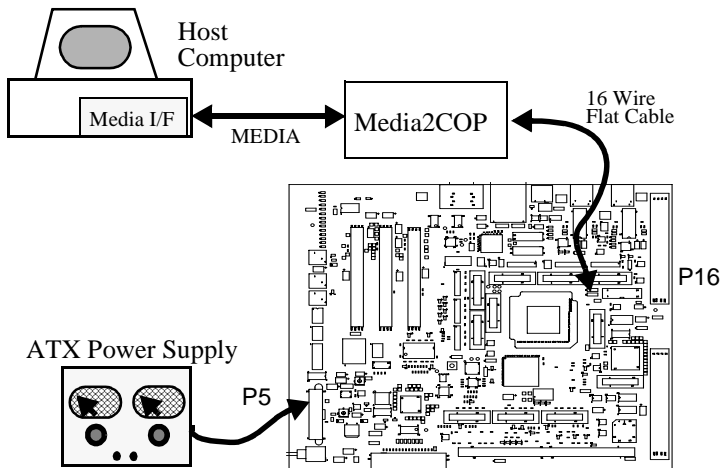
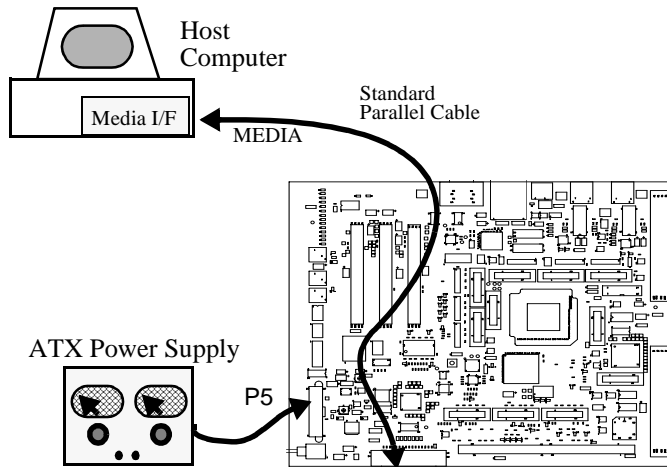


Figure 2-11. Host Controlled Operation Scheme - Command Converter

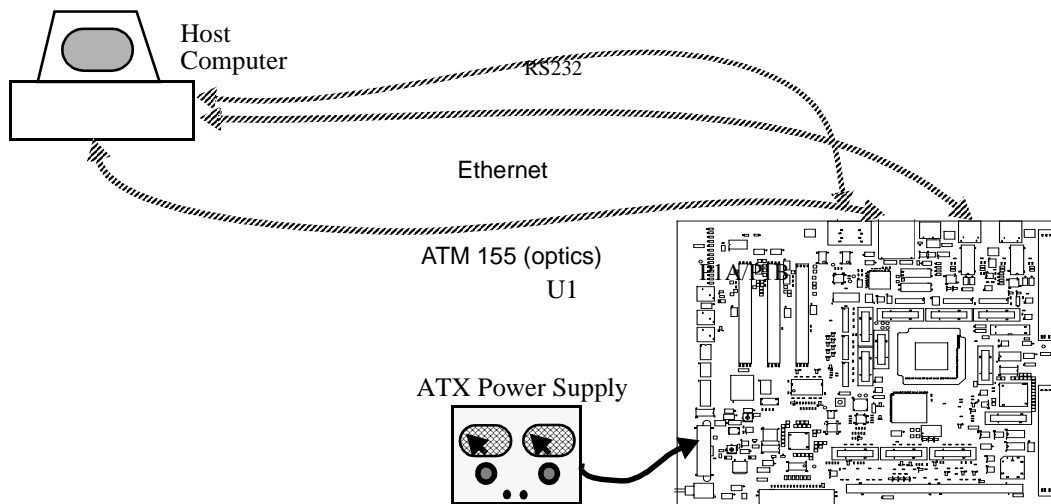
2. The host is connected to the board directly from the host's parallel port.



**Figure 2-12. Host Controlled Operation Scheme - Parallel Port**

### 2.4.2 Stand Alone Operation

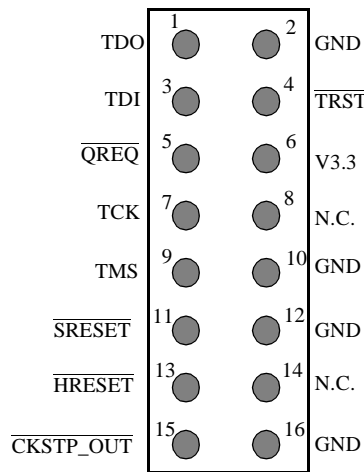
In this mode, the board is not controlled by the host via the COP port. It may connect to host via one of its other ports, e.g., RS232 port, Fast Ethernet port, ATM155 port etc. Operating in this mode requires an application program to be programmed into the board's Flash memory.



**Figure 2-13. Stand Alone Configuration**

### 2.4.3 COP/JTAG Connector - P16

The PQ2FADS-VR COP interface connector, P16, is a 16 pin, male, Header connector. The connection between the PQ2FADS-VR and the COP controller is by a 16 line flat cable, supplied with the COP controller board obtained from a third party developer. [Figure 2-14.](#) shows the pin configuration of the connector.

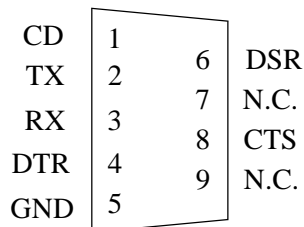


**Figure 2-14.** P16 - COP/JTAG Port Connector

### 2.4.4 Terminal to PQ2FADS-VR RS-232 Connection

A serial (RS232) terminal or any other RS232 equipment, may be connected to the RS-232 connectors P1A and P1B. The RS-232 connectors are a 9 pin, female, D-type connectors, arranged in a stacked configuration. P1B connected to SCC2 of the PQ2 is the **lower** and P1A, connected to SCC1 of the PQ2, 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<sup>1</sup> or compatibles, i.e. via a flat cable. The pinout which is identical for both P1A and P1B is shown in Figure 2-15..



**Figure 2-15.** P1A/P1B - RS232 Serial Port Connector

### 2.4.5 10/100-Base-T Ethernet Ports Connection

The 10/100-Base-T port connectors - P3 and P4, are an 8-pin, 90°, receptacle RJ45 connector. The connection between the 10/100-Base-T ports to the network is done by a standard cable, having two RJ45/8 jacks on its ends. The pinout of P3 and P4 is described in Table 7-2. "P3,P4 - 100/10 Base-T Ethernet Connector" on page 80.

<sup>1</sup>.IBM-AT is a trademark of International Business Machines Inc.

## 2.4.6 Memory Installation

The PQ2FADS-VR is supplied with one type of memory module:

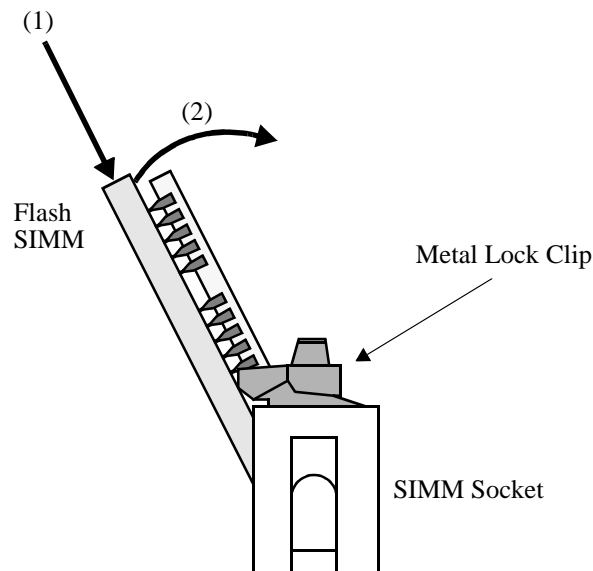
- Flash Memory SIMM.

### 2.4.6.1 Flash Memory SIMM Installation

To install a memory SIMM, it should be taken out of its package, put diagonally in its socket - U50 - and then raised to a vertical position until the metal lock clips are locked. See [Figure 2-16](#).

#### **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 2-16.** Flash Memory SIMM Insertion

## Operating Instructions

### 3.1 Introduction

This chapter provides necessary information to use the PQ2FADS-VR 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 PQ2FADS-VR 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 PQ2, as if the power was re-applied to the ADS. When the PQ2 is reset that way, all configuration and all data residing in volatile memories are lost. After  $\overline{\text{PORST}}$  signal is negated, the PQ2 re-acquires the power-on reset and hard-reset configuration data from the hard-reset configuration source. (Flash | EEPROM | BCSR).

#### 3.2.2 ABORT Switch - SW3

The ABORT switch is normally used to abort program execution, this by issuing a level 0 interrupt to the PQ2. 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 PQ2FADS-VR. The ABORT switch signal is debounced, and may be disabled by software.

#### 3.2.3 SOFT RESET Switch - SW2

The SOFT RESET switch SW3 performs Soft reset to the PQ2 internal modules, maintaining PQ2'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 PQ2. When the PQ2 is HARD reset, all its configuration is lost<sup>1</sup>, including data stored in the SDRAMs and the PQ2 has to be re-initialized.

<sup>1</sup>.Except for Hard-Reset configuration word, which is acquired only once, after PON-Reset.

### 3.2.5 SW6 - Reset Configuration Switch

SW5 is a 4-switch Dip-Switch. For its function see [Section 2.3.5](#).

### 3.2.6 SW7 - Software Options Switch

SW7 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 SW7 state. SW7 is factory set to all ON. See [Figure 3-1](#).

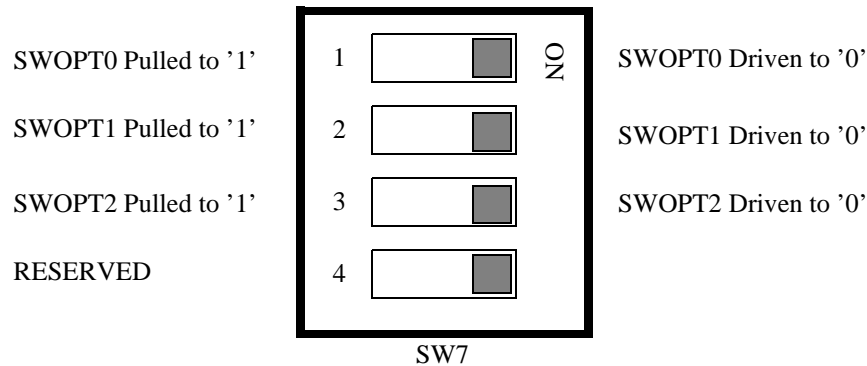


Figure 3-1. SW7 - Description

### 3.2.7 P26 - VDDL Voltage Level Range Selection

P26 selects between 4 different voltage level ranges available for VDDL. For further information over its function see [Section 2.3.1](#).

### 3.2.8 JP12 - IDDL Measurement

JP12 resides in IDDL's main current flow. To measure IDDL, JP12 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 JP12 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 PQ2FADS-VR.

### 3.2.9 JP6 - Thermal Sense Connector

There are 2 dedicated pins THERM(0:1) which provide a way to take internal temperature measurements of the PQ2. These pins should be connected to GND for normal operation. JP6 is factory set with a jumper on its 2 - 3 positions, so that THERM1 is connected to GND.

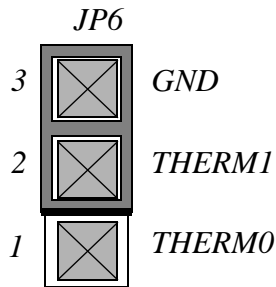


Figure 3-2. JP6 - Therm Connector

### 3.2.10 JP11 - IDDH Measurement

JP11 resides in IDDH’s main current flow. To measure IDDH, JP11 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 JP11 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 PQ2FADS-VR.

### 3.2.11 JP13 - VPP Source Selector

JP13 selects the source for VPP - programming voltage for the Flash SIMM. When a jumper is located between pins 2 - 3 of JP13 , the VPP is connected to the VCC plane of the board, providing 5V VPP. When a jumper is located between positions 1 - 2 of JP13, VPP is drawn from the 12V plane, that provides 12V VPP. JP13 options are shown in Figure 3-3.

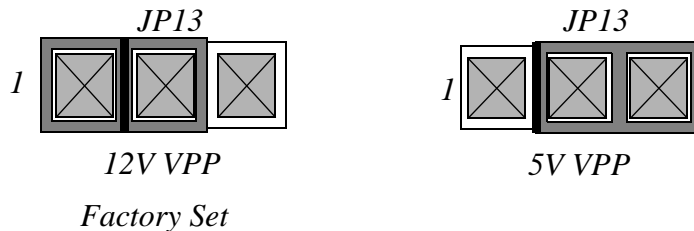


Figure 3-3. JP13 - VPP Source Selection

### 3.2.12 GND Bridges

There are 7 GND bridges on the PQ2FADS-VR. 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 PQ2FADS-VR.

#### **3.2.13 Power O.K. Indicator - LD1**

The green Power O.K. LED indicator lights if the ATX power supply is generating all the voltages.

#### **3.2.14 12V Indicator - LD2**

The green 12V led - LD2, indicates the presence of the +12V supply on the board.

#### **3.2.15 UTOPIA 16 Bit Indicator - LD15**

The green UTOPIA16 led - LD15, indicates that the UTOPIA is in 16 bit mode. When off - the UTOPIA is in 8 bit mode.

#### **3.2.16 UTOPIA Multi PHY Indicator - LD16**

The green Multi PHY led - LD16, indicates that the UTOPIA is in Multi PHY mode. When off - the UTOPIA is in single PHY mode.

#### **3.2.17 5V Indicator - LD3**

The green 5V led - LD3, indicates the presence of the +5V supply on the board.

#### **3.2.18 3.3V Indicator - LD4**

The green 3.3V led - LD4, indicates the presence of the +3.3V supply on the board.

#### **3.2.19 USB Power Indicator - LD17**

The green USB Power led - LD17, indicates the presence of 5V in the USB cable.

#### **3.2.20 -12V Indicator - LD5**

The green -12V led - LD5, indicates the presence of the -12V supply on the board.

#### **3.2.21 RUN Indicator - LD6**

When the green RUN led - LD6 is lit, it indicates that the PQ2 is performing cycles on the PPC Bus. When dark, the PQ2 is either running internally or stuck.

#### **3.2.22 ATM ON - LD7**

When the yellow ATM ON led is lit, it indicates that the ATM-UNI transceiver - the PM5384, is enabled for communication. When it is dark, the ATM-UNI transceiver is disconnected from the PQ2, enabling the use of its associated FCC1 pins off-board via the expansion connectors.

ATM ON led is controlled by BCSR1.

**3.2.23 Fast Ethernet Port 2 Enabled - LD10**

When the yellow ETH2 ON led is lit, it indicates that the fast ethernet port 2 transceiver - the DM9161, is connected to FCC3. When it is dark, it indicates that the DM9161 is in power down mode and disconnected from FCC3, enabling the use of its associated FCC3 pins off-board via the expansion connectors. The state of LD10 is controlled by BCSR1.

**3.2.24 Fast Ethernet Port 1 Enabled - LD8**

When the yellow ETH1 ON led is lit, it indicates that the fast ethernet port 1 transceiver - the DM9161, is connected to FCC2. When it is dark, it indicates that the DM9161 is in power down mode and disconnected from FCC2, enabling the use of its associated FCC2 pins off-board via the expansion connectors. The state of LD8 is controlled by BCSR1.

**3.2.25 RS232 Port 1 ON - LD11**

When the yellow RS232 Port 1 ON led is lit, it designates, that the RS232 transceiver connected to P1A (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.

**3.2.26 Fast Ethernet Port 1 Full Duplex Indicator - LD18**

When the Dm9161 on FCC2 is enabled and is in Full Duplex operation mode, the red led - LD18 lights.

**3.2.27 RS232 Port 2 ON - LD12**

When the yellow RS232 Port 2 ON led is lit, it designates, that the RS232 transceiver connected to P1B (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.

**3.2.28 Fast Ethernet Port 1 100Base-Tx Indicator - LD19**

When the DM9161 on FCC2 is enabled and is in 100 Mbps operation mode, the green led - LD19 lights.

**3.2.29 Ethernet Port 1 Tx/Rx Indicator - LD20**

The green Ethernet Transmit/Receive LED indicator blinks whenever the Dm9161 on FCC2 is transmitting or receiving data via the 10/100-Base-T port.

**3.2.30 General Purpose Led 2 Indicator - LD13**

This is a general purpose red LED which is user controlled by BCSR0.

**3.2.31 Ethernet Port 1 LINK Indicator - LD21**

The yellow Ethernet Twisted Pair Link Integrity LED indicator - LINK, lights to indicate good link integrity on the 10/100-Base-T port. LD21 is off when the link integrity fails.

**3.2.32 Fast Ethernet Port 2 Full Duplex Indicator - LD22**

When the Dm9161 on FCC3 is enabled and is in Full Duplex operation mode, the red led - LD20

lights.

### **3.2.33 General Purpose Led 1 Indicator - LD14**

This is a general purpose green LED which is user controlled by BCSR0.

### **3.2.34 Fast Ethernet Port 2 100Base-Tx Indicator - LD23**

When the DM9161 on FCC3 is enabled and is in 100 Mbps operation mode, the green led - LD22 lights.

### **3.2.35 USB Enabled Indicator - LD9**

The yellow USB enable LED indicates that the USB transceiver is connected to the PQ2.

### **3.2.36 Ethernet Port 2 LINK Indicator - LD24**

The yellow Ethernet Twisted Pair Link Integrity LED indicator - LINK, lights to indicate good link integrity on the 10/100-Base-T port. LD24 is off when the link integrity fails.

### **3.2.37 Ethernet Port 2 Tx/Rx Indicator - LD25**

The green Ethernet Transmit/Receive LED indicator blinks whenever the Dm9161 on FCC3 is transmitting or receiving data via the 10/100-Base-T port.

### **3.2.38 VDDL Indication - LD26**

The green VDDL indicator led - LD26 is lit to indicate a VDDL power activity. Since VDDL level may vary, LD26's illumination level also varies accordingly.

### **3.2.39 Parallel Port connection - LD27**

The green Parallel Port connection LED indicates that the board is connected directly to the Pc's parallel port and the COP/JTAG connector (P16) is irrelevant.

### **3.2.40 External Debugger Connection Indicator - LD28**

The green external debugger connection LED indicates that a command converter can be connected to the COP/JTAG connector (P16).

## Functional Description

In this chapter the various modules combining the PQ2FADS-VR are described to their design details.

### 4.1 Reset & Reset - Configuration

There are several reset sources on the PQ2FADS-VR:

1. Power On Reset
2. Manual Hard-Reset
3. Manual Soft-Reset
4. PCI bus reset
5. PQ2 Internal Sources. (See also the PQ2 U/M)

#### 4.1.1 Power - On ResetPQ2

The power on reset to the PQ2 initializes the processor state after power up. A dedicated logic, using Seiko S-80728AN-DR-T1, which is a voltage detector of 2.8V +/- 2.4%, asserts  $\overline{\text{PORESET}}$  input to the PQ2 for a period of ~2.5sec. This time period is long enough to cover also the VDDL stabilization, powered by a different voltage regulator. It is assumed that the stabilization time for both linear regulators (see also [Section 6.1 Power Supply](#)) are about the same. Power-On-Reset may be generated manually as well by an on-board dedicated push-button (SW1). Power-On Reset can also be generated by the JTAG logic, which is integrated with BCSR.

##### 4.1.1.1 Power - On Reset Configuration

At the end of Power - On reset sequence, MODCK(1:3) are sampled by the PQ2 to configure the various clock modes of the PQ2 (core, cpm, bus, PCI...). Selection between the MODCK(1:3) combination options is done by means of dip-switches ([Section 2.3.3](#)) on the mother board while PCI\_MODCKH(0:3) are obtained from the relevant dedicated pins (by means of dip-switches - [Section 2.3.6](#)).

The configuration master is determined upon the rising edge of  $\overline{\text{PORST}}$ , according to the state of  $\overline{\text{RSTCONF}}$  ([Section 2.3.5](#)) signal, driven low on this board, to set the PQ2 as a configuration master.

After power-on reset negates, the hard-reset sequence starts, during which, many other different options are configured (see [Section 4.1.2.4 "Hard Reset Configuration" on page 26](#)), among these options, are additional clock configuration bits - PCI\_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.

The PCI\_MODCK signal, which is sampled concurrently with the PCI\_MODCK(0:3) pins, determines the PCI bus clock frequency (see Section 2.3.7). When set high, it divides the PCI bus frequency by two. When reset low, the PCI bus frequency is as determined by the MODCK(1:3) and PCI\_MODCKH(0:3) signals.

### 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. PQ2's internal sources.

Hard-Reset, when generated, causes the PQ2 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.

$\overline{\text{HRESET}}$  when asserted, is extended internally by the PQ2 for additional 512 bus clock cycles at the end of which, the PQ2 waits for 16 bus clock cycles and then, re-checks the state of the  $\overline{\text{HRESET}}$  line.

$\overline{\text{HRESET}}$  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 board logic and/or to the PQ2 itself.

#### 4.1.2.1 COP/JTAG Port Hard - Reset

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

#### 4.1.2.2 Manual Hard Reset

To allow run-time Hard-reset, when the COP controller is disconnected from the PQ2FADS-VR and to support resident debuggers, manual Hard is facilitated. Depressing both Soft-Reset (SW3) and ABORT (SW2) buttons asserts the  $\overline{\text{HRESET}}$  pin of the PQ2, generating a HARD RESET sequence.

Since the  $\overline{\text{HRESET}}$  line may be driven internally by the PQ2, it must be driven to the PQ2 with an open-drain gate. If off-board H/W connected to the PQ2FADS-VR is to drive  $\overline{\text{HRESET}}$  line, then it should do so with an open-drain gate, this, to avoid contention over this line.

When Hard Reset is generated, the PQ2 is reset in a destructive manner, i.e., the hard reset configuration is re-sampled and all registers (except for the PLL's) are reset, including memory controller registers - reset of which results in a loss of dynamic memory contents.

To save on board's real-estate, this button is not a dedicated one, but is shared with the Soft-Reset button and the ABORT button - when both are depressed, Hard Reset is generated.

## 4.1.2.3 Internal Sources Hard - Reset

The PQ2 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  $\overline{\text{HRESET}}$  line via the external pin's scan chain. Not useful for run time.

In general, the PQ2 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 Reset Configuration

When Hard-Reset is applied to the PQ2 (externally as well as internally), it samples the Hard-Reset configuration word. This configuration may be taken from an internal default, in case  $\overline{\text{RSTCONF}}$  is negated during  $\overline{\text{HRESET}}$  asserted or taken from the Flash <sup>1</sup>/E<sup>2</sup>PROM/BCSR (MS 8 bits of the data bus) in case  $\overline{\text{RSTCONF}}$  signal is asserted along with  $\overline{\text{HRESET}}$ . The default configuration word can be taken from the E<sup>2</sup>PROM/BCSR in case the Flash has been tampered with. The selection between the BCSR, FLASH and the E<sup>2</sup>PROM as the source of the default configuration word is determined by a dedicated dip-switch (see [Section 2.3.5](#)) and a jumper (see [Section 2.3.4](#)).

During hard reset sequence, the configuration master<sup>2</sup> reads the Flash (or E<sup>2</sup>PROM or BCSR) memory at addresses 0, 8, 0x18, 0x20,... a byte each time, to assemble the 32 bit configuration word. A total of 64 bytes of data is read from D(0:7) to acquire 8 full configuration words for system that may have upto 8 PQ2 chips.

The configuration word for a single<sup>3</sup> PQ2 is stored in the Flash memory SIMM, in the E<sup>2</sup>PROM or as default in the BCSR, while the other seven words are not initialized, as there are no additional PQ2 on the PQ2FADS-VR. The default configuration word is shown in [Table 4-1](#). for the FLASH and in [Table 4-2](#). for the E<sup>2</sup>PROM. PCI module configuration is 256 Bytes long and should start at address 0x100.

There are four possible configuration words:

- PQ2FADS-VR without L2 Cache - FLASH/BCSR is the boot device.  $\overline{\text{CS0}}$  is assigned to the FLASH and  $\overline{\text{CS4}}$  is assigned to the E<sup>2</sup>PROM.

- 
1. In general, from any device residing on  $\overline{\text{CS0}}$ .
  2. In general, The PQ2 for which  $\overline{\text{RSTCONF}}$  is asserted along with  $\overline{\text{PORST}}$  asserted or in particular, the PQ2 residing on the PQ2FADS-VR.
  3. Although the PQ2 as configuration master reads 8 configuration words, only the 1'st configuration word is influential.

- PQ2FADS-VR without L2 Cache - E<sup>2</sup>PROM is the boot device.  $\overline{CS0}$  is assigned to the E<sup>2</sup>PROM and  $\overline{CS4}$  is assigned to the FLASH.
- PQ2FADS-VR with L2 Cache - FLASH is the boot device.  $\overline{CS0}$  is assigned to the FLASH and  $\overline{CS4}$  is assigned to the E<sup>2</sup>PROM.
- PQ2FADS-VR with L2 Cache - E<sup>2</sup>PROM is the boot device.  $\overline{CS0}$  is assigned to the E<sup>2</sup>PROM and  $\overline{CS4}$  is assigned to the FLASH.

**Table 4-1. BCSR/FLASH 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 <sup>a</sup>
EXMC	1	'0'	Internal Memory Controller. $\overline{CS0}$ active at system boot.		
CDIS	2	'0'	Core Enabled.		
EBM	3	'0'/'1'	'0' - Single PQ2 Mode for boards without L2Cache '1' - 60X Bus Mode <sup>a</sup> for boards with L2Cache		
BPS	4:5	11	32 Bit Boot Port Size		
CIP	6	'0'	Sets Core Initial Prefix MSR[IP]=1, so that system exception table is placed at address 0xFFFF00100 regardless of FLASH memory size		
ISPS	7	'0'	64 bit internal space for external master accesses. In fact don't care on this board since external master is not supported.		
L2CPC	8:9	'10'	$\overline{CI}/\overline{BADDR}(29)/\overline{IRQ2}$ selected as $\overline{BADDR}(29)$ $\overline{WT}/\overline{BADDR}(30)/\overline{IRQ3}$ selected as $\overline{BADDR}(30)$ $\overline{L2\_HIT}/\overline{IRQ4}$ selected as unassigned $\overline{CPU\_BG}/\overline{BADDR}(31)/\overline{IRQ5}$ as $\overline{BADDR}(31)$	8	B2
DPPC	10:11	'11'	Data Parity Pin configuration as: DP0 as EXT_BR2 DP1 as EXT_BG2 DP2 as EXT_DBG2 DP3 as EXT_BR3 DP4 as EXT_BG3 DP5 as EXT_DBG3 DP6 as $\overline{IRQ6}$ DP7 as $\overline{IRQ7}$		
Reserved	12	'0'	Reserved.		
ISB	13:15	'010'	IMMR initial value 0x0F000000, i.e., the internal space resides initially at this address.		

### Table 4-1. BCSR/FLASH Hard Reset Configuration Word

Field	Data Bus Bits	Prog Value [Bin]	Implication	Offset In Flash [Hex]	Value [Hex]
BMS	16	'0'	Boot memory (Flash) at 0xFE000000.	10	36 / 02 <sup>b</sup>
BBD	17	'0'	$\overline{ABB}/\overline{IRQ2}$ pin is $\overline{ABB}$ $\overline{DBB}/\overline{IRQ3}$ pin is $\overline{DBB}$		
MMR	18:19	'11'/00'	'11' - Mask Masters Requests. Boot Master is PCI when PCI is enabled in the FLASH. '00' - No masking, Local Bus SDRAM mode in the BCSR.		
LBPC	20:21	'01'/00'	'11' - Local Bus pins function as PCI bus (FLASH). '00' - Local Bus pins function as Local Bus (BCSR).		
APPC	22:23	'10'	MODCK1/AP(1)/TC(0) functions as BKSEL0 MODCK2/AP(2)/TC(1) functions as BKSEL1 MODCK3/AP(3)/TC(2) functions as BKSEL2 IRQ7~/APE~ functions as IRQ7~ CS11~/AP(0) functions as CS11~		
CS10PC	24:25	'01'	CS10~/BCTL1/DBG_DIS~ functions as BCTL1		
ALD_EN	26	'0'	PCI Auto Load Enable. When high, PCI Bridge Configuration is done automatically from the FLASH/E <sup>2</sup> PROM (CPM is configuration master - PPC core should be disabled) right after the Hard Configuration Word. When low, the PPC Core should configure the PCI Bridge.		
Reserved	27	'0'	Reserved.		
MODCK_HI <sup>c</sup>	28:31	'0101'	Determines the Core's frequency out of power-up reset. Actually, not relevant when the PCI is active since the PCI_MODCK(0:3) take presidency.		

- a. For L2 Cache Boards.
- b. BCSR is set for no PCI configuration
- c. Applies only ONCE after power-up reset.

**Table 4-2. E<sup>2</sup>PROM 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	04 / 14 <sup>a</sup>
EXMC	1	'0'	Internal Memory Controller. $\overline{CS0}$ active at system boot.		
CDIS	2	'0'	Core Enabled.		
EBM	3	'0' / '1'	'0' - Single PQ2 Mode for boards without L2Cache '1' - 60X Bus Mode <sup>a</sup> for boards with L2Cache		
BPS	4:5	'01'	8 Bit Boot Port Size		
CIP	6	'0'	Sets Core Initial Prefix MSR[IP]=1, so that system exception table is placed at address 0xFFFF00100 regardless of FLASH memory size		
ISPS	7	'0'	64 bit internal space for external master accesses. In fact don't care on this board since external master is not supported.		
L2CPC	8:9	'10'	$\overline{CI}/\overline{BADDR}(29)/\overline{IRQ2}$ selected as $\overline{BADDR}(29)$ $\overline{WT}/\overline{BADDR}(30)/\overline{IRQ3}$ selected as $\overline{BADDR}(30)$ $\overline{L2\_HIT}/\overline{IRQ4}$ selected as unassigned $\overline{CPU\_BG}/\overline{BADDR}(31)/\overline{IRQ5}$ as $\overline{BADDR}(31)$	8	B2
DPPC	10:11	'11'	Data Parity Pin configuration as: DP0 as EXT_BR2 DP1 as EXT_BG2 DP2 as EXT_DBG2 DP3 as EXT_BR3 DP4 as EXT_BG3 DP5 as EXT_DBG3 DP6 as $\overline{IRQ6}$ DP7 as $\overline{IRQ7}$		
Reserved	12	'0'	Reserved.		
ISB	13:15	'010'	IMMR initial value 0x0F000000, i.e., the internal space resides initially at this address.		

**Table 4-2. E<sup>2</sup>PROM Hard Reset Configuration Word**

Field	Data Bus Bits	Prog Value [Bin]	Implication	Offset In Flash [Hex]	Value [Hex]		
BMS	16	'0'	Boot memory (E <sup>2</sup> PROM) at 0xFE000000.	10	36		
BBD	17	'0'	$\overline{ABB}/\overline{IRQ2}$ pin is $\overline{ABB}$ $\overline{DBB}/\overline{IRQ3}$ pin is $\overline{DBB}$				
MMR	18:19	'11'	Mask Masters Requests. Boot Master is PCI.				
LBPC	20:21	'01'	Local Bus pins function as PCI bus.				
APPC	22:23	'10'	MODCK1/AP(1)/TC(0) functions as BKSEL0 MODCK2/AP(2)/TC(1) functions as BKSEL1 MODCK3/AP(3)/TC(2) functions as BKSEL2 IRQ7~/APE~ functions as IRQ7~ CS11~/AP(0) functions as CS11~				
CS10PC	24:25	'01'	CS10~/BCTL1/DBG_DIS~ functions as BCTL1			18	45
ALD_EN	26	'0'	PCI Auto Load Enable. When high, PCI Bridge Configuration is done automatically from the FLASH/E <sup>2</sup> PROM (CPM is configuration source - PPC core should be disabled) right after the Hard Configuration Word. When low, the PPC Core should configure the PCI Bridge.				
Reserved	27	'0'	Reserved.				
MODCK_HI <sup>b</sup>	28:31	'0101'	Determines the Core's frequency out of power-up reset. Actually, not relevant when the PCI is active since the PCI_MODCK(0:3) take presidency.				

- a. For L2 Cache Boards.
- b. Applies only ONCE after power-up reset.

The PCI configuration registers which are set at Hard-Reset sequence are shown in Figure 4-1.

				Address Offset (Hex)
□ Reserved				
Device ID (0x18C0)		Vendor ID (0x1057)		00
PCI Status		PCI Command		04
Class Code	Subclass Code	Standard Programming	Revision ID	08
BIST Control	Header Type	Latency Timer	Cache Line Size	0C
PIMMR Base Address Register				10
				14
				18
				1C
Subsystem ID		Subsystem Vendor ID		2C
				28
Capability Pointer				34
////////				38
MAX LAT	MIN GNT	Interrupt Pin	Interrupt Line	3C
////////				40
PCI Arbiter Control		PCI Function		44

Figure 4-1. PCI Host Configuration Registers

### 4.1.3 Soft Reset

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

1. COP/JTAG Port
2. Manual Soft Reset
3. Internal PQ2 source.

Soft-Reset, when generated, causes the PQ2 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.

$\overline{\text{SRESET}}$  when asserted, is extended internally by the PQ2 for an additional 512 bus clock cycles at the end of which, the PQ2 waits for 16 bus clock cycles and then, re-checks the state of the  $\overline{\text{SRESET}}$  line.

$\overline{\text{SRESET}}$  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 boards' logic and / or to the PQ2 itself.

#### 4.1.3.1 COP/JTAG Port Soft - Reset

To provide convenient soft-reset capability for a COP/JTAG controller,  $\overline{\text{SRESET}}$  line appears at the COP/JTAG port connector - P3. The COP/JTAG controller may directly generate Soft-reset by asserting (low) this line.

#### 4.1.3.2 Manual Soft Reset

To allow run-time Soft-reset, when the COP controller is disconnected from the PQ2FADS-VR and to support resident debuggers, a Soft Reset push-button is provided. When the Soft Reset

push-button is depressed, the  $\overline{\text{SRESET}}$  line is asserted to the PQ2, generating a Soft Reset sequence.

Since the  $\overline{\text{SRESET}}$  line may be driven internally by the PQ2, it must be driven by an open-drain gate, to avoid contention over that line. If off-board H/W connected to the PQ2FADS-VR is to drive  $\overline{\text{SRESET}}$  line, then, it should do so with an open-drain gate, this, to avoid contention over this line.

#### **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  $\overline{\text{SRESET}}$  pin via the boundary scan chain. This is not useful for run time.

#### **4.1.4 PCI Bus Reset**

The PCI Module in the PQ2 can generate a reset signal dedicated for PCI devices which reside on the PCI bus. This is a reset to the PCI bus which is initiated by the PCI bus Host - the PQ2 on this board. This reset can also be initiated by a Soft PCI Reset by setting a dedicated bit in a PCI control register (consult the PQ2 User Manual for details).

## **4.2 Local Interrupter**

There are external interrupts which are applied to the PQ2 via its interrupt controller:

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

### **4.2.1 ABORT Interrupt**

The ABORT (NMI), is generated by a push-button. When this button is depressed, the  $\overline{\text{IRQ0}}$  input to the PQ2 is asserted. The purpose of this type of interrupt, is to support the use of resident debugger if any is made available to the board. This interrupt is enabled by setting the MSR[EE] bit.

To support external (off-board) generation of an NMI, the  $\overline{\text{IRQ0}}$  line, is driven by an open-drain gate. This allows for an external h/w, to also drive this line. If an external h/w indeed does so, it is compulsory that  $\overline{\text{IRQ0}}$  is driven by an open-drain (or open-collector) 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  $\overline{\text{IRQ7}}$  line of the PQ2. This  $\overline{\text{IRQ7}}$  input is shared with the Fast Ethernet PHY Interrupt. Since INTB of the UNI is an open-drain output, it is possible to connect additional (on and off-board) interrupt requesters on the same  $\overline{\text{IRQ7}}$ , provided that they drive  $\overline{\text{IRQ7}}$  with open-drain gate as well. When an interrupt request appears in  $\overline{\text{IRQ7}}$ , it is necessary to

check the source of the interrupt whether it's the ATM UNI or the Fast Ethernet PHY.

### 4.2.3 Fast Ethernet PHY Interrupt

To support the two fast Ethernet Transceivers event reports by means of interrupt, the interrupt outputs of the DM9161 are connected to  $\overline{IRQ7}$  line of the PQ2. This  $\overline{IRQ7}$  input is shared with the ATM UNI Interrupt.

### 4.2.4 PCI Interrupt

Each PCI slot can generate up to four interrupts to a total of twelve (3 slot x 4 interrupts each). Each PCI expansion board can generate an interrupt at any given time. Since there is only one interrupt input available in the PQ2, an Interrupt Controller is used. The Interrupt Controller receives all the possible interrupts from the PCI slots and generate one interrupt ( $\overline{IRQ6}$ ) to the PQ2.

A simple generic Interrupt Controller is implemented using a CPLD device. The Interrupt Controller is implemented as an Interrupt Register and an Interrupt Mask Register. The Interrupt Controller has its' own dedicated chip-select line ( $\overline{CS8}$ ). A simple priority scheme is devised to prioritize the interrupts from different slots. The PCI IRQ routing are according to Figure 4-2..

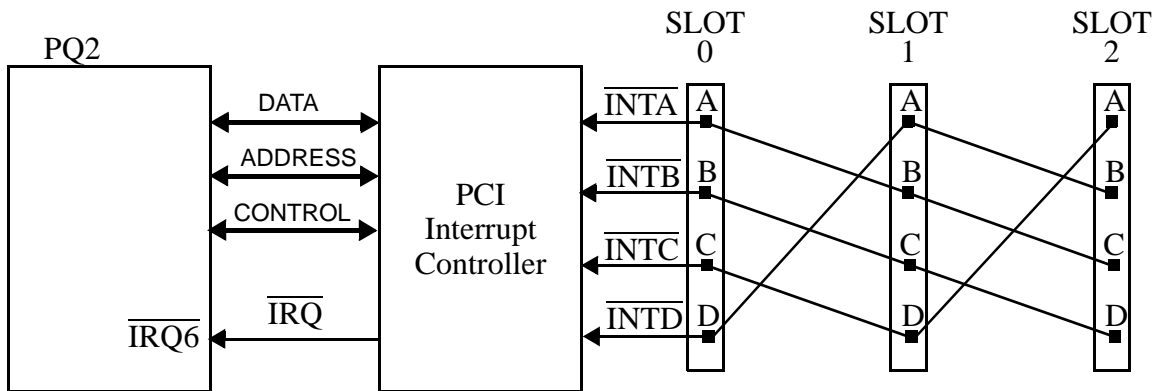


Figure 4-2. PCI Interrupt Routing Scheme

An interrupt request in any of the  $\overline{INTx}$  lines, will set three interrupt bits in the PCI Interrupt Register (if not masked in the Interrupt Mask Register) since there are three possible interrupt sources for every  $\overline{INTx}$  line. It is up to the user to implement a polling process to verify the real interrupt source (by polling the Interrupt Pending bit in the PCI device) and clear the other two. The PCI Interrupt Register can be read at any time and accessed at offset 0x0 from  $\overline{CS8}$  base address. The description of the PCI Interrupt Register is in Table 4-3..

**Table 4-3. PCI Interrupt Register Description**

BIT	MNEMONIC	Function	PON DEF	ATT.
0	PCI0_INTA	<b>PCI Slot 0 INTA.</b> PCI Slot 0 Interrupt A: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
1	PCI0_INTB	<b>PCI Slot 0 INTB.</b> PCI Slot 0 Interrupt B: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
2	PCI0_INTC	<b>PCI Slot 0 INTC.</b> PCI Slot 0 Interrupt C: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
3	PCI0_INTD	<b>PCI Slot 0 INTD.</b> PCI Slot 0 Interrupt D: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
4	PCI1_INTA	<b>PCI Slot 1 INTA.</b> PCI Slot 1 Interrupt A: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
5	PCI1_INTB	<b>PCI Slot 1 INTB.</b> PCI Slot 1 Interrupt B: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
6	PCI1_INTC	<b>PCI Slot 1 INTC.</b> PCI Slot 1 Interrupt C: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
7	PCI1_INTD	<b>PCI Slot 1 INTD.</b> PCI Slot 1 Interrupt D: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
8	PCI2_INTA	<b>PCI Slot 2 INTA.</b> PCI Slot 2 Interrupt A: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
9	PCI2_INTB	<b>PCI Slot 2 INTB.</b> PCI Slot 2 Interrupt B: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
10	PCI2_INTC	<b>PCI Slot 2 INTC.</b> PCI Slot 2 Interrupt C: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
11	PCI2_INTD	<b>PCI Slot 2 INTD.</b> PCI Slot 2 Interrupt D: '0' - no interrupt was requested '1' - an interrupt was requested and waiting to be handled	0	R
12-31	Reserved	Un-implemented		R/W

Also available is an Interrupt Mask Register which provides the user with the option to mask any of the possible PCI interrupt sources. It can be read or written at any time and accessed at offset 0x4 from CS8 base address. The description of the PCI Interrupt Mask Register is in [Table 4-4.](#)

Table 4-4. PCI Interrupt Mask Register Description

BIT	MNEMONIC	Function	PON DEF	ATT.
0	MPCI0_INTA	<b>Mask PCI Slot 0 INTA.</b> Mask PCI Slot 0 Interrupt A: '0' - interrupt is available '1' - interrupt is masked	0	R/W
1	MPCI0_INTB	<b>Mask PCI Slot 0 INTB.</b> Mask PCI Slot 0 Interrupt B: '0' - interrupt is available '1' - interrupt is masked	0	R/W
2	MPCI0_INTC	<b>Mask PCI Slot 0 INTC.</b> Mask PCI Slot 0 Interrupt C: '0' - interrupt is available '1' - interrupt is masked	0	R/W
3	MPCI0_INTD	<b>Mask PCI Slot 0 INTD.</b> Mask PCI Slot 0 Interrupt D: '0' - interrupt is available '1' - interrupt is masked	0	R/W
4	MPCI1_INTA	<b>Mask PCI Slot 1 INTA.</b> Mask PCI Slot 1 Interrupt A: '0' - interrupt is available '1' - interrupt is masked	0	R/W
5	MPCI1_INTB	<b>Mask PCI Slot 1 INTB.</b> Mask PCI Slot 1 Interrupt B: '0' - interrupt is available '1' - interrupt is masked	0	R/W
6	MPCI1_INTC	<b>Mask PCI Slot 1 INTC.</b> Mask PCI Slot 1 Interrupt C: '0' - interrupt is available '1' - interrupt is masked	0	R/W
7	MPCI1_INTD	<b>Mask PCI Slot 1 INTD.</b> Mask PCI Slot 1 Interrupt D: '0' - interrupt is available '1' - interrupt is masked	0	R/W
8	MPCI2_INTA	<b>Mask PCI Slot 2 INTA.</b> Mask PCI Slot 2 Interrupt A: '0' - interrupt is available '1' - interrupt is masked	0	R/W
9	MPCI2_INTB	<b>Mask PCI Slot 2 INTB.</b> Mask PCI Slot 2 Interrupt B: '0' - interrupt is available '1' - interrupt is masked	0	R/W
10	MPCI2_INTC	<b>Mask PCI Slot 2 INTC.</b> Mask PCI Slot 2 Interrupt C: '0' - interrupt is available '1' - interrupt is masked	0	R/W
11	MPCI2_INTD	<b>Mask PCI Slot 2 INTD.</b> Mask PCI Slot 2 Interrupt D: '0' - interrupt is available '1' - interrupt is masked	0	R/W
12-31	Reserved	Un-implemented		R/W

### 4.3 Clock Generator

There are two main clock circuits on board:

1. PQ2 System Clock
2. PCI Clock

#### 4.3.1 PQ2 Clock

The PQ2 requires a single clock source as the main clock source. All PQ2 60x bus timings are referenced to the main clock input - CLKIN1. The main clock input is in 1:1 ratio to the bus clock, with internal skew elimination (PLL). Use is done with 66MHz (Hip4 devices) 3.3V clock oscillator (100MHz for Hip7 device), which is connected to a low inter-skew buffer (U36) to split the load between all various clock consumers on both boards.

Special care is taken to isolate and terminate the clock route between the on-board PLL and the PQ2, this to provide a "clean" clock input for proper operation. The main clock scheme is shown in Figure 4-3.

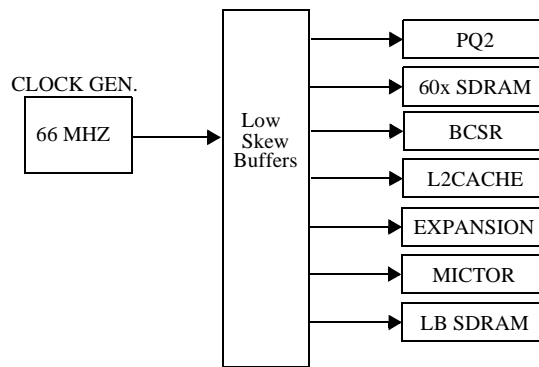


Figure 4-3. Main Clock Generator Scheme

#### 4.3.2 PCI Clock

The PCI bus clock is derived internally from the main clock input CLKIN1. The generated PCI clock is output from a PCI-dedicated PLL (named DLL). That clock output is feeding an on-board low-skew and fast clock distributor which distributes the PCI clock to all on-board PCI devices. One of the outputs is fed back to the PCI clock to the PQ2 through CLKIN2 input. This clock input is driven to the DLL which synchronizes the DLL output clock to the CLKIN2 input clock and thus, maintains low skew between the DLL output and CLKIN2 input. All PCI bus timings are referenced to the CLKIN2 input clock. Special care was taken when the board layout was done to keep all copper traces from the Clock Distributor outputs at the same lengths, including the output that is fed back to CLKIN2. This is in compliance with the PCI standard to achieve bus

synchronization and low skew. The PCI clock scheme is shown in Figure 4-4.

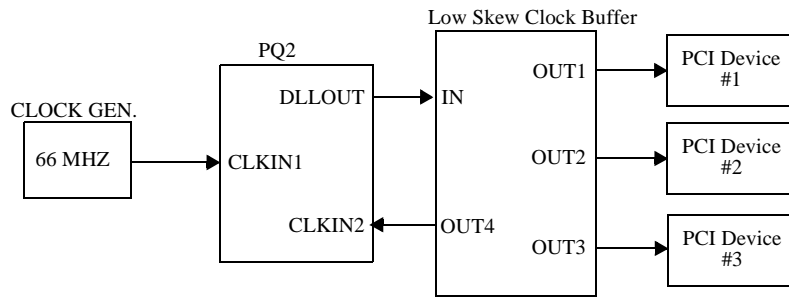


Figure 4-4. PCI Clock Generator Scheme

## 4.4 Bus Configuration

The PQ2 may be configured in 2 possible bus modes depending on the presence of L2 cache on board.

1. Single PQ2 Mode
2. 60X Bus Mode.

### 4.4.1 Single PQ2 Mode

When a L2 Cache is not present on the board, the PQ2 is configured in Single PQ2 Mode. I.e., assuming only one PQ2 on the 60x bus, with no support for external master access. This allows for internal address multiplexing to occur which makes the external address multiplexers redundant and therefore not assembled. This improves SDRAM performance.

### 4.4.2 60X Bus Mode

When L2 Cache is installed on the PQ2FADS-VR, the PQ2 may no longer operate in single PQ2 mode since the address must be seen as is by the cache. That requires the use of the external address multiplexers for the SDRAM. In this mode, SDRAM performance is decreased due to added wait-state, caused by the delay associated with the external multiplexers, on the 1'st access in a page,.

#### **NOTE**

In this mode, only devices which are 60x compatible (or devices which have 64 bit data bus and are buffered from the 60x bus) can operate on the 60x bus. This due to the 60x bus address tenure feature. This means that when the L2

Cache is used, the Flash, EEPROM, BCSR and PCI Interrupt Controller are not accessible. For further details, consult the PQ2 User Manual.

## 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, E<sup>2</sup>PROM, ATM UNI M/P interface, PCI Interrupt Controller and the BCSR are buffered, while the SDRAM and the cache are not buffered from the 60X bus.

Latches are provided over address and strobe (when necessary) lines while transceivers are provided for data. Use is done with 74ALVT buffers (by Philips) which are 3.3V operated and 5V tolerant<sup>1</sup> and provide bus hold to reduce pull-up/pull-down resistors count (as required by the PQ2). This type of buffers reduces noise on board due to reduced transitions' amplitude.

To further reduce noise and reflections, serial damping resistors are placed over SDRAM address and all PQ2 strobe lines.

The data transceivers are open only if there is an access to a valid<sup>2</sup> buffered board address or during Hard - Reset configuration<sup>3</sup>. That way data conflicts are avoided in case an unbuffered memory read or off-board memory is read - provided that it is not mapped to an address valid on board. It is the users' responsibility to avoid such errors.

## 4.6 Chip - Select Generator

The memory controller of the PQ2 is used as a chip-select generator to access on-board (and off-board) memories, saving boards' area, reducing cost, power consumption and increasing flexibility. To enhance off-board application development, memory modules (including the BCSRx) may be disabled via BCSR<sup>4</sup> in favor of an external memory connected via the expansion connectors. That way, a CS line may be used off-board via the expansion connectors, while its associated local memory is disabled.

When a CS region, assigned to a buffered<sup>5</sup> memory, is disabled via BCSR, the local data transceivers are disabled during access to that region, avoiding possible<sup>6</sup> contention over data lines.

- 
1. Required for Flash, E<sup>2</sup>PROM, Interrupt Controller and BCSR
  2. An address which is covered in a Chip-Select region, that controls a buffered device.
  3. To allow a configuration word stored in the Flash/E<sup>2</sup>PROM memory to become active.
  4. After the BCSR is removed from the local memory map, there is no way to access it but to re-apply power to the PQ2FADS-VR.
  5. When an unbuffered CS region is being accessed, buffers do not open anyway.
  6. During read cycles.

The PQ2 chip-select assignments to the various memories / registers on the PQ2FADS-VR are shown in Table 4-5.

**Table 4-5. PQ2FADS-VR Chip Select Assignments**

<b>Chip Select:</b>	<b>Assignment</b>	<b>Bus</b>	<b>Timing Machine</b>
$\overline{CS0}$	Flash SIMM / E <sup>2</sup> PROM <sup>a</sup>	60X (Buffered)	GPCM
$\overline{CS1}$	BCSR	60X (Buffered)	GPCM
$\overline{CS2}$	SDRAM	60X (Main)	SDRAM Machine 1
$\overline{CS3}$	Unused, user available	-	-
$\overline{CS4}$	E <sup>2</sup> PROM / Flash SIMM <sup>a</sup>	60X (Buffered)	GPCM
$\overline{CS5}$	ATM UNI Microprocessor I/F	60X (Main)	GPCM
$\overline{CS6}$	Communication Tool M/P Interface $\overline{CS1}$ .	60X (Buffered)	GPCM/UPMx
$\overline{CS7}$	Communication Tool M/P Interface $\overline{CS2}$ .	60X (Buffered)	GPCM/UPMx
$\overline{CS8}$	PCI Interrupt Controller	60X (Buffered)	GPCM
$\overline{CS(9-11)}$	Unused, user available	-	-

a. Selection is done by a dip-switch.

## 4.7 Synchronous Dram (60X Bus)

To enhance performance, especially in higher operation frequencies - 32MBytes of SDRAM are provided on board. The SDRAM is unbuffered from the PQ2 60X bus. Use is done with two MTLC4M32B2 by Micron or compatibles, which each is 1M X 32bit X 4banks.

The SDRAM's timing is controlled by SDRAM Machine #1 associated with 60X bus, via its assigned Chip Select lines (See Table 4-5.). The SDRAM Machine supports PBI (Page Bank Interleave) which increases the SDRAM throughput. The SDRAM connection scheme when no

L2 cache is used is shown in Figure 4-5.

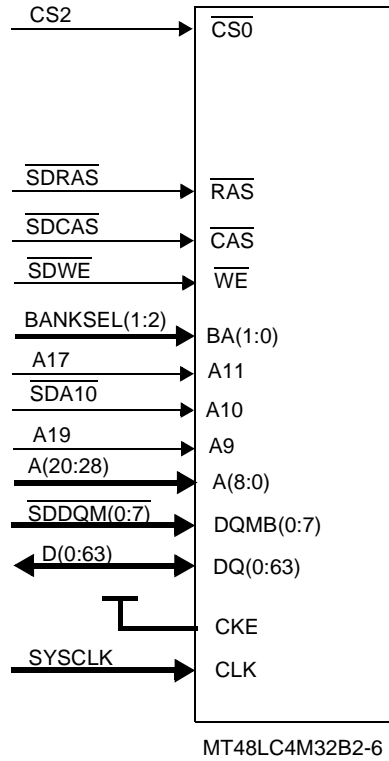


Figure 4-5. 60x SDRAM Connection Scheme - No L2 Cache

The SDRAM connection scheme when L2 cache is installed is shown in Figure 4-6.

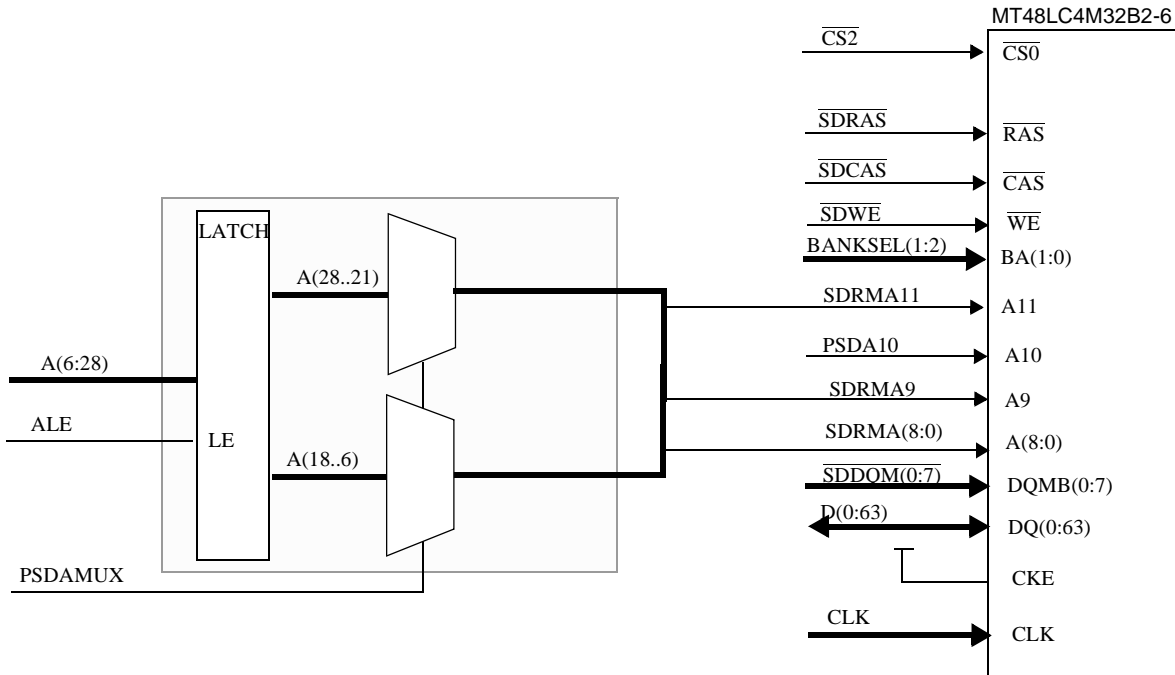


Figure 4-6. SDRAM - 60x Bus Connection Scheme with L2 Cache

### 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.

An SDRAM is programmed by issuing a Mode Register Set command. During that command, data is passed to the Mode Register through the SDRAMs' address lines. This command is fully supported by the SDRAM machine of the PQ2. Before that can take place, the SDRAM machine of the PQ2 has to be initialized.

Mode Register programming values are shown in [Table 4-6](#):

**Table 4-6. 100 MHz SDRAM Mode Register Programming**

SDRAM Address Line <sup>a</sup>	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	'011'	Data Valid 3 Clocks cycles after CAS Asserted
A3	Burst Type	'0'	Sequential Burst
A2 - A0	Burst Length	'010'	4 Operand Burst Length

a. Actually SDRAMs' A0 is connected to PQ2s' A28 and so on...

### 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 8.2  $\mu$ sec, so that all 4096 SDRAM rows are refreshed within specified 34 msec, while leaving an interval of ~30 msec of refresh redundancy within that window, as a safety measure, to cover for possible delays in bus availability for the refresh controller.

### 4.7.3 L2-Cache Support Influence On SDRAM Design

To support an optional L2-Cache on the PQ2FADS-VR, the following measures need to be taken:

1. Optional Latches - Multiplexers are added over selected address lines. See [Figure 4-6](#).  
These Latches - Multiplexers are normally by-passed by 0  $\Omega$  resistors that are not assem-

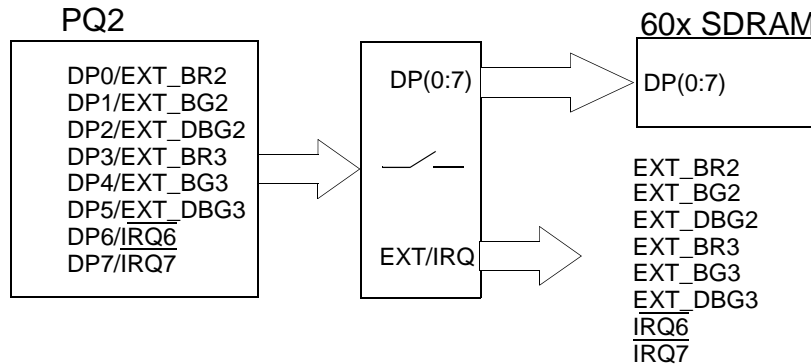
bled in L2cache boards.

2. The PQ2 supports additional wait-state on SDMUX line, so that the row-address may be allowed to propagate via the Latch - Multiplexers in time for the Activate command.
3. To support SDRAM PBI (Page Based Interleaving), the relative location of the Row-Address field, is shifted up the address lines, depended on the number of internal banks within an SDRAM. This since the Bank Select line(s) are inserted between the Column (LSB) and Row (MSB) address lines.
4. The L2 Cache used is the MPC2605. This device can operate at maximum speed of 66MHz. Therefore, the **USE OF L2 CACHE WILL LIMIT THE 60X BUS FREQUENCY TO 66MHZ ONLY** (compared to 100MHz without L2 Cache - for the MPC8280).

The performance of the SDRAM is decreased by the addition of the external multiplexers of the SDRAMs' address lines.

#### 4.7.4 SDRAM Error Correction Support

The PQ2FADS-VR has an optional support for Parity Error Correction for SDRAM accesses. To support that option, the DP(0:7) lines are connected to the SDRAM DP(0:7) lines. Since the PQ2 muxes DP(0:7) signals with other signals, bus switch is used to select between DP(0:7) signals and other functions.



**Figure 4-7.** 60x SDRAM Data Parity Support

**NOTE:** When using the Data Parity option, IRQ6 and IRQ7 pins change functionality to Data Parity pins. Therefore, the two interrupt lines are switched to IRQ2 and IRQ3 so the user should be aware and switch to work with the relevant IRQs. To be able to work with IRQ2 and IRQ3, this function must be enabled in SIUMCR register.

## 4.8 Flash Memory SIMM

The PQ2FADS-VR is provided with 8Mbyte of 95 nsec flash memory SIMM, the SM73228XG1JHBGO by Smart Modular Technology which is composed of four LH28F016SCT-L95 chips by Sharp, arranged as 2M X 32 in a single bank. Support is given also to 16MBytes and 32 MBytes simms. The Flash SIMM resides on an 80 pin SIMM socket and is buffered from the 60X bus to reduce capacitive load over it.

To minimize use of PQ2s' chip-select lines, only one chip-select line ( $\overline{CS0}$  or  $\overline{CS4}$  if the E<sup>2</sup>PROM is using  $\overline{CS0}$ ) 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 inserted to the PQ2FADS-VR.

The access time of the Flash memory provided with the PQ2FADS-VR is 95 nsec, however, devices with different delay are supported as well. By reading the delay section of the Flash SIMM Presence-Detect lines (see [Table 4-11.](#)), the debugger can establish (via register OR0 in case  $\overline{CS0}$  is used or OR4 if  $\overline{CS4}$  is used) the correct number of wait-states needed to access the Flash SIMM (considering 100MHz system clock frequency).

The control over the Flash is done with the GPCM and a dedicated  $\overline{CS0}$  (or  $\overline{CS4}$ ) region which controls the whole bank. During hard - reset initialization<sup>1</sup>, the debugger or any application S/W for that matter, reads the Flash Presence-Detect lines via BCSR and determines how to program registers BR0 & OR0 (or BR4 & OR4), within which the size and the delay of the region are determined. The Flash module may be disabled / enabled at any time by writing '1' / '0'

---

1. i.e., initialization that follow the hard reset sequence at system boot.

respectively to the  $\overline{\text{FlashEn}}$  bit in BCSR1. The Flash connection scheme is shown in Figure 4-8..

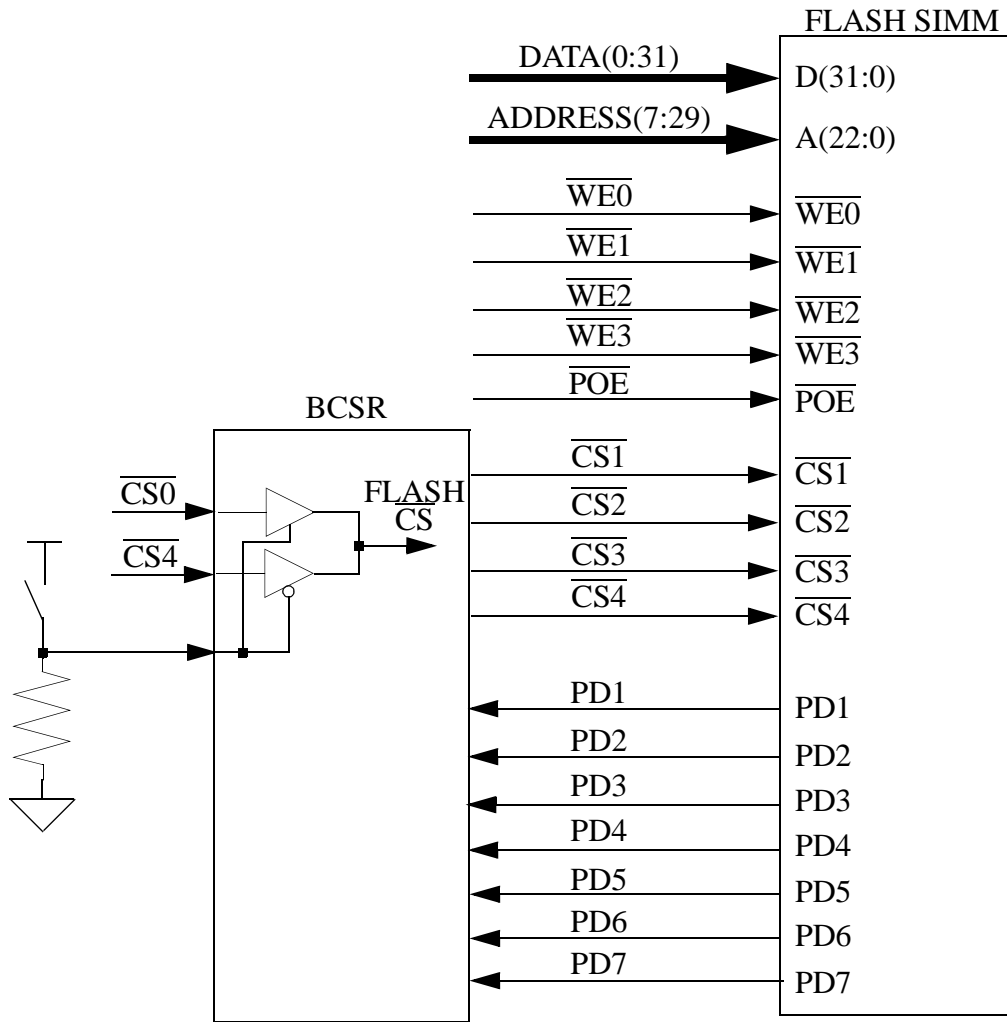


Figure 4-8. FLASH SIMM Connection Scheme

As can be seen in Figure 4-8., the FLASH  $\overline{\text{CS}}$  is distributed to four  $\overline{\text{CS}}$  signals. The distribution depends on the size of the FLASH module installed - it is read by the BCSR using the PD(1-7) pins.

The Hard-Reset configuration word stored in the FLASH differs from the one stored in the E<sup>2</sup>PROM in the BPS field which is the Boot Port Size - the E<sup>2</sup>PROM is 8 bits while the FLASH is 32 bits.

### 4.8.1 Flash Programming Voltage

Support is given to 5V as well as 12V programmable modules. The selection between VPP's voltage levels is done via a dedicated jumper. To avoid inadvertent programming or erasure of the Flash it is recommended to leave the jumper open so that no VPP is applied to the Flash SIMM.

### 4.8.2 Flash and L2Cache

If the L2 cache is installed, the PQ2 needs to be programmed to 60x bus mode. This requires the latches for the buffered address bus to the Flash (As well as all other slow static devices) to be

enabled. The 3 lowest order address lines for the Flash, are provided by the BADDR(27-29) lines of the PQ2. However, BADRR29 function of the PQ2 is multiplexed with  $\overline{CI}$  (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<sup>1</sup>, prior to changing BADDR29 function to  $\overline{CI}$  via SIUMCR.

## 4.9 E<sup>2</sup>PROM Memory

The PQ2FADS-VR is provided with 8 KBytes of E<sup>2</sup>PROM memory in a PLCC package. The E<sup>2</sup>PROM resides on a socket in case it is desired to replace or re-program a different configuration for the board. The E<sup>2</sup>PROM is used only for the purpose of supplying the Reset Configuration Word during power-on reset and for storing the PCI configuration data. It is used as a back-up for the Flash memory in case the Flash is not installed or the data it holds is incorrect. As a back-up, it holds the default Hard-Reset configuration word and the default PCI configuration. The Hard-Reset configuration word stored in the E<sup>2</sup>PROM differs from the one stored in the FLASH in the BPS field which is the Boot Port Size - the E<sup>2</sup>PROM is 8 bits while the FLASH is 32 bits. It uses a single chip-select,  $\overline{CS0}$  or  $\overline{CS4}$ , which depends on the chip-select used by the Flash. The selection of the chip-select is done by a dip-switch. The E<sup>2</sup>PROM connection scheme is shown in [Figure 4-9](#).

The device used is ATMEL AT28HC64B, a 5V Byte alterable E<sup>2</sup>PROM, 150ns access time with byte-wide JEDEC pinout. Although the device is placed in a socket, it can be programmed on-board. In order to program the device on-board, it has to be unlocked - it can be locked to prevent unauthorized alterations of its contents. The lock can be done by hardware or software. The hardware lock is done by write inhibit - the PQ2 does not assert  $\overline{WE}$  during write cycles (set in the BRx register). The software lock is achieved by writing a unique sequence to the device. To

---

1. It is required to do so anyway, since the L2Cache must operate within a full 64-bit data bus environment.

unlock, a different unique sequence has to be written.

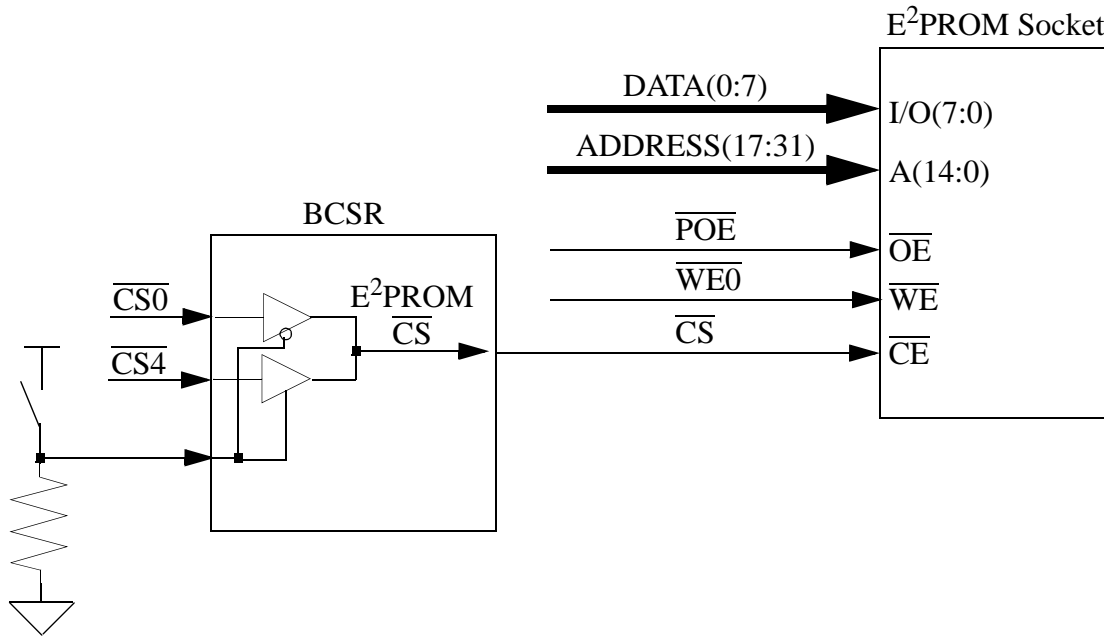


Figure 4-9. E<sup>2</sup>PROM Connection scheme

Additional address lines are connected to the socket according to the JEDEC format as an option to use E<sup>2</sup>PROM up to 32 KByte. To allow proper operation with the L2 Cache, the PQ2 needs to be set to 60X bus mode in which the address bus for the E<sup>2</sup>PROM<sup>1</sup> is latched.

## 4.10 PCI Bus

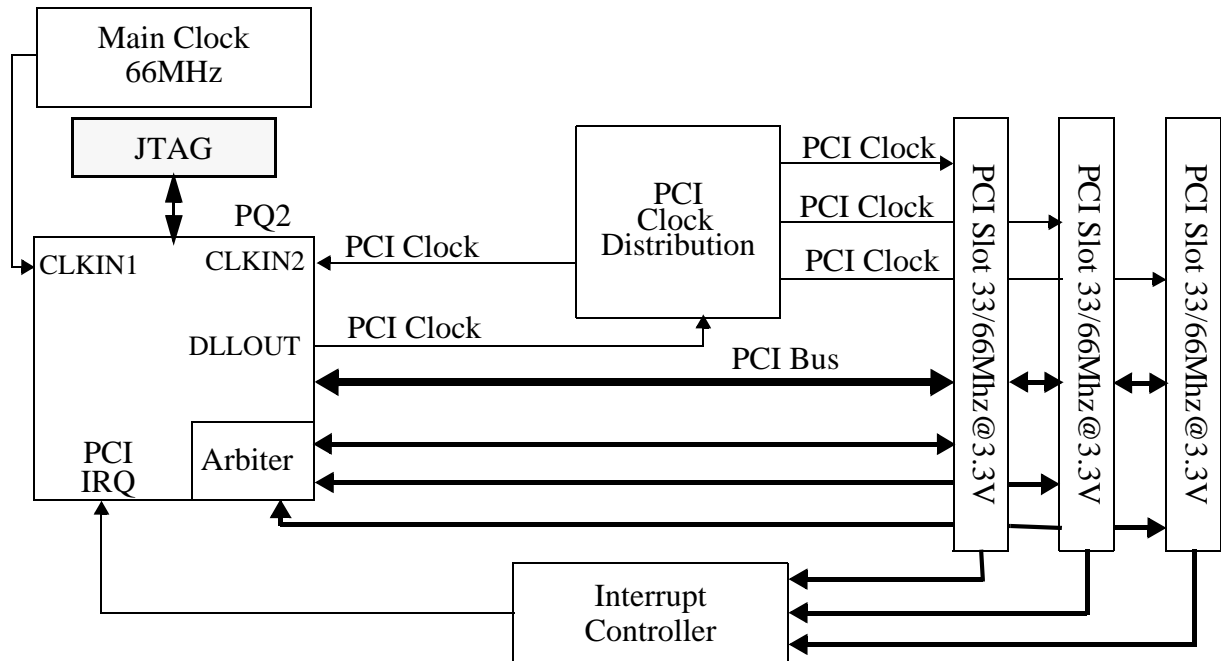
The PQ2 has a PCI module which enables it to act as an Host (Master) or a Target. On this board, the PQ2 serves only as a PCI host - a bridge between the PCI Bus and the PowerPC core.

The PQ2 PCI Bridge is designed to connect the PowerPC processor and memory system to the PCI system bus, to which I/O components are connected. The PCI Bridge enables the PQ2 to gluelessly bridge PCI masters and agents to a PowerPC system host. It uses a 32-bit multiplexed, address/data bus that can run from 25MHz up to 66MHz. The interface provides address and data parity with error checking and reporting. It also provides three physical address spaces: 32-bit address memory; 32-bit address I/O; and the PCI configuration space.

The PQ2 also includes an on-chip Arbiter which enables arbitration of up to three PCI masters. Only three PCI slots are supported on the PQ2FADS-VR because of the Arbiter capacity. Each slot can host either a PCI master or PCI target. The PQ2 as a Bridge can support more PCI devices but that will require extra slots that can host PCI targets only. Therefore, to avoid dedicated slots for PCI targets, only three slots are implemented.

1. As well as all other slow static devices.

The PCI Bridge is implemented on the PQ2 Local Bus. Due to PCI Standard restrictions, no other application can reside on the local bus. The PCI bus can operate at frequencies of 25MHz up to 66MHz @ 3.3V only. The 3.3V restriction is due to the PQ2 which is not 5V compliant. The PCI bus layout is shown in Figure 4-10. Special care was taken when the layout of the PQ2FADS-VR was done so that the PCI standard recommendations are followed strictly.



**Figure 4-10. PCI Bus Scheme**

The clock source for the PQ2 is Main Clock 66MHz (or 100MHz) clock oscillator. The PCI Clock is derived internally from the Main Clock and output at DLLOUT. That clock is then distributed to each PCI device on the bus in a way that they are all synchronized (by keeping all clock traces the same length). The PCI Clock is also fed back to the PQ2 for synchronization and skew elimination purposes.

An interrupt from any PCI slot is handled by a simple generic Interrupt Controller. Each slot can generate up to four interrupts for a total of twelve interrupts that the controller will support. It will be made of two register mapped in a dedicated  $\overline{CS}$  region. One is an Interrupt Register (see Table 4-3.) and the second is Interrupt Mask Register (see Table 4-4.). A simple priority scheme is devised to allow the controller to support more than one interrupt concurrently.

## 4.11 L2-CACHE Support

To enhance benchmarking, optional support is provided for L2-Cache. Use is done with two MPC2605 devices, each containing 256KBytes of look-aside<sup>1</sup> cache along with its control, providing a total of 512KByte L2-cache.

The cache is connected directly over the 60X bus and is supported gluelessly by the PQ2.

The presence of the L2-Cache, calls for the introduction of latch - multiplexers over SDRAMs' address lines because 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<sup>1</sup>. These latch - multiplexers are soldered in place only in case a cache is installed on-board. Otherwise they are omitted and bypassed by 0 Ω resistors. See also [Section 4.7.3 L2-Cache Support Influence On SDRAM Design](#).

### 4.11.1 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 [Table 4-7](#).

**Table 4-7. L2 Cache CFG(0:2) Settings**

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

2.  $\overline{\text{Snoop}}$  is Enabled - CFG3 driven low for both modules.
3.  $\overline{\text{AACK}}$  assertion enabled - CFG4 driven high for both modules.

The caches'  $\overline{\text{HRESET}}$  lines are connected directly to the  $\overline{\text{SRESET}}$  line of the PQ2 so that whenever Soft-reset is asserted to or by the PQ2, 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:

- $\overline{\text{PWRDWN}}$  - constantly set to high (no power down support on the PQ2FADS-VR)
- $\overline{\text{L2FLUSH}}$  - assertion of which<sup>2</sup> flushes out the cache array. This signal is controlled by BCSR0.
- $\overline{\text{L2MISS\_INH}}$  - in fact Cache-Lock. When Asserted the cache does not change its contents. Controlled by BCSR0.
- $\overline{\text{L2TAG\_CLR}}$  - Clears all tag memory. Controlled by BCSR0.
- $\overline{\text{L2UPDATE\_INH}}$  - In fact cache freeze (without information loss). Controlled by BCSR0.

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

---

1. i.e., residing on the same bus as the processor.  
 1. Only single level is allowed with the PQ2.  
 2. For minimum 8 Bus clock cycles.

## 4.12 Communication Ports

The PQ2FADS-VR has several communication ports, to allow convenient evaluation of the CPM features. Obviously, it is not possible to provide all types of communication interfaces supported by the CPM, but it is made convenient to connect any communication interface devices to the PQ2 via the CPM Expansion connectors, residing on the edge of the board.

All CPM pins are visible on MICTOR connectors. In order to avoid long routes and stubs, bus muxing devices are used to direct the CPM signals to a communication element on-board or to the expansion connector. A signal that is used on-board, will not be visible in the expansion connector and vice-versa. The control is done by enabling/disabling the communication elements on-board.

The communication ports' interfaces provided on the PQ2FADS-VR are listed below:

1. 155 Mbps ATM UNI on FCC1 with Optical interface, using the UTOPIA Level 2 interface - support for 8 or 16 bit in multi or single PHY.
2. Two 100/10-Base-T Ports on FCC2 and FCC3 with T.P. interface, MII or RMII (on Hip7 devices only) controlled.
3. Dual RS232 ports residing on SMC1 & SMC2.
4. USB port, 1.1 USB standard compliant, with speed control (12 or 1.5 Mbps) and mode control (Host or slave).

### 4.12.1 ATM Port

To support the PQ2s' ATM controller, a 155.52Mbps User Network Interface (UNI) is provided on board, connected to FCC1 of the PQ2 via UTOPIA I/F. Use is done with PM5384 S/UNI-155-ULTRA by PMC-SIERA. Although these transceivers are capable of supporting 51.84Mbps rate, support is given to 155.52Mbps only. The PHY supports UTOPIA level 2 which means support for 8 or 16 bit UTOPIA bus in single or multi PHY mode. The control over the mode of UTOPIA bus connection is done through BCSR3.

The control over the transceiver is done using the microprocessor interface of the transceiver, controlled by the PQ2 memory controllers' GPCM. Since the UNI is 5V powered and the PQ2 is 3.3V powered (5V intolerant), the UNI is buffered (LCX buffers) from the PQ2 on both the receive part of UTOPIA interface and the microprocessor control ports.

The ATM transceiver may be enabled / disabled at any time by writing '0' / '1' respectively to the  $\overline{\text{ATMEN}}$  bit in BCSRx. When  $\overline{\text{ATMEN}}$  is negated, ('1') the microprocessor control port is also detached from the PQ2 and its associated FCC may be used off-board via the expansion connectors.

The ATM transceiver reset input is driven by  $\overline{\text{HRESET}}$  signal of the PQ2, 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 4-9.) or by asserting ('1') the RESET bit in the Master Reset and Identify / Load Meters register via the UNI microprocessor interface.

The UNI transmit and receive clocks are fed with a 19.44 MHz +/- 20 ppm, clock generator, 5 V powered, while the receive and transmit fifos' clocks of the UTOPIA interface are provided by the PQ2. The PQ2 can provide the same clock for both UTOPIA transmit and receive or separate

clocks for each, hard-configured<sup>1</sup>.

The ATM SAR is connected to the physical medium by an optical interface. Use is done with HP's HFBR 5805 optical interface, which operates at 1300 nm with upto 2 Km transmission range.

The ATM PHY is connected to  $\overline{\text{IRQ7}}$  and generates an interrupt when an appropriate event occurs.

**NOTE:** When the 60x Data Parity option is on,  $\overline{\text{IRQ7}}$  pin switches functionality to parity and the interrupt output is routed to  $\overline{\text{IRQ3}}$ . It is the responsibility of the user to set the appropriate functionality of the  $\overline{\text{IRQ3}}$  pin (SIUMCR register).

**NOTE:** When 16 bit UTOPIA bus is used, the extra pins are in conflict with other functions. In that case, the 16 bit UTOPIA bus will disable the USB, RS232 port 2 and the Fast Ethernet MDC functions.

**NOTE:** When Multi PHY UTOPIA bus is used, the extra pins are in conflict with other functions. In that case, the multi PHY UTOPIA bus will disable the two RS232 port functions.

### 4.12.2 100/10 Base - T Ports

Two fast Ethernet ports with T.P. (100-Base-TX) I/F is provided on the PQ2FADS-VR. These ports also support 10 Mbps ethernet (10-Base-T) via the same transceiver - the DM9161 by Davicom.

The DM9161 are connected to FCC2 and FCC3 of the PQ2 via MII or RMII interface, which is used for both - devices' control and data path. The initial configuration of the DM9161 on the PQ2FADS-VR is set by external resistors - 100Base-Tx Full Duplex in MII mode. The selection between MII/RMII for FCC2 and FCC3 is done by jumpers JP2 and JP3 respectively. The DM9161 must be set to MII or RMII while in power-down.

The DM9161 reset input is driven by either asserting the FETH\_RST bit in BCSR1 (see [Table 4-9](#).) or by asserting a specific bit in an internal register via MII I/F.

To allow external use of FCC2 and FCC3, their pins appear at the CPM expansion connectors and the ethernet transceiver may be Disabled / Enabled at any time via the MIIs' MDIO port.

The DM9161 is able to interrupt the PQ2 via  $\overline{\text{IRQ7}}$  line. This line is shared also with the CPM expansion connectors. Therefore, any tool that is connected to  $\overline{\text{IRQ7}}$ , should drive these lines with an Open Drain buffer.

**NOTE:** When the 60x Data Parity option is on,  $\overline{\text{IRQ7}}$  pin switches functionality to parity and the interrupt output is routed to  $\overline{\text{IRQ3}}$ . It is the responsibility of the user to set the appropriate functionality of the  $\overline{\text{IRQ3}}$  pin (SIUMCR register).

#### 4.12.2.1 DM9161 Control

The DM9161 is controlled via the MII management<sup>2</sup> port which is a 2 wire interface: a clock

---

1. Using resistors.

2. Also known as MII MDIO port.

(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 some pins. On the board, the slave address is hard-set to b00000 for FCC2 and b00011 for FCC3. The PQ2 interfaces this port using two PI/O pins: PC9 for MDIO and PC10 for MDC. There is no special support within the PQ2 for the MDIO port and the protocol is implemented in S/W.

The MDIO port may interrupt a host in 2 ways: (a<sup>1</sup>) driving low the MDIO line during IDLE time or (b) using a dedicated interrupt line  $\overline{\text{MDINT}}$ . This line is connected to the PQ2's DP7/CSE1/ $\overline{\text{IRQ7}}$  line, appearing also at the CPM expansion connectors.

Since  $\overline{\text{IRQ7}}$  may also be driven by any tool, connected to the expansion connectors, it should be driven with an Open Drain buffer.  $\overline{\text{IRQ7}}$  is pulled-up on the board.

**NOTE:** If ATM 16 bit UTOPIA bus or USB port are enabled, either one will conflict with the MDC and MDIO signals. Therefore, the MDC and MDIO functionality will switch to PC3 and PC2 respectively.

### 4.12.3 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 PQ2FADS-VR, connected to SCC1 and SCC2 ports of the PQ2. Use is done with MAX3241 transceiver which generates RS232 levels internally using a single 3.3V supply and has a standby mode. When the  $\overline{\text{RS232EN1}}$  or  $\overline{\text{RS232EN2}}$  bits in BCSR1 are asserted (low), the corresponding transceiver is enabled. When negated, the corresponding transceiver is in standby mode, within which the receiver outputs are tri-stated, enabling the use of the corresponding ports' pins off-board via the expansion connectors.

Nine pins, female D-Type stacked connector is used, configured to be directly (via a flat cable) connected to a standard IBM-PC like RS232 connector.

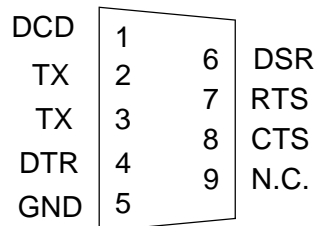


Figure 4-11. RS232 Serial Ports Connector

#### 4.12.3.1 RS-232 Ports' Signal Description

In the list below, the directions 'I', 'O', and 'I/O' are relative to the PQ2FADS-VR board. (i.e.'I' means input to the PQ2FADS-VR)

- CD (O) - Data Carrier Detect. This line is always asserted by the PQ2FADS-VR.
- TX (O) - Transmit Data.
- RX (I) - Receive Data.

1. Not supported on the board.

- DTR (I) - Data Terminal Ready. This signal is used by the software on the PQ2FADS-VR to detect if a terminal is connected to the board.
- DSR (O) - Data Set Ready. This line is always asserted by the PQ2FADS-VR.
- RTS (I) - Request To Send. This line is not connected in the PQ2FADS-VR.
- CTS (O) - Clear To Send. This line is always asserted by the PQ2FADS-VR.

**NOTE:** RS232 port 2 (SCC2) functionality is in conflict with ATM 16 bit UTOPIA bus and Multi PHY UTOPIA bus. RS232 port 1 is in conflict with Multi PHY UTOPIA bus. It is up to the user to determine the desired function on the shared pins.

### 4.12.4 USB Port

The USB port resides on the PQ2FADS-VR and is driven by the USB port of the MPC8275 (Hip7 only) through SCC4. A dedicated USB transceiver - the PDIUSBP11 by PHILIPS is provided, along with a tri-state buffer, separating this port from the MPC8275's USB port, this to allow Port disable option and off-board use of MPC8275 USB pins.

To correctly support the 2 speed modes of the USB, detachable pull-up resistors (3.3V) are provided over D+ and D- lines of the USB, controlled by the USB\_SPD bit of BCSR4. When USB\_SPD is in low-speed level (low) D- is pulled-up while D+ remains floating. When USB\_SPD bit is in high-speed level, D+ is being pulled-up and D- floats.

Also, 5V power will optionally be provided for the USB connector, controlled by USB\_VCC0 in BCSR4. When USB\_VCC0 is driven low, a 5V supply will be connected to pin 1 of the USB connectors.

**NOTE:** The USB function is in conflict with ATM 16 bit UTOPIA bus and Fast Ethernet MDC functions. It is up to the user to select the desired function on the shared pins.

### 4.12.5 PC Parallel Port

A new feature to this board is the direct connection to a PC parallel port for the purpose of debugger connection (CodeWarrior). An on-board logic is used to interface to the parallel port and translate the signals to COP/JTAG format. The parallel port support both EPP and SPP modes of the parallel port in a PC. The direct connection eliminates the need for an external command converter. When connected to a PC's parallel port, the parallel port connection has automatic priority over the COP/JTAG connector interface.

## 4.13 Board Control & Status Register - BCSR

Most of the hardware options on the PQ2FADS-VR are controlled or monitored by the BCSR, which is a 32 bit wide read / write register file. The BCSR is accessed via the PQ2s' memory controller (see [Table 4-5.](#)) 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 inside that region.

The following functions are controlled / monitored by the BCSR:

1. PBI
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.
  - UTOPIA 8/16 bit
  - UTOPIA single/multi PHY
7. Fast Ethernet Ports Control which includes:
  - Transceiver Initial Enable
  - Transceiver Reset
8. RS232 port 1 Enable / Disable.
9. RS232 port 2 Enable / Disable.
10. USB Port Control which includes:
  - Transceiver Initial Enable
  - USB Speed
  - USB Power
11. Flash Size / Delay Identification.
12.  $\overline{CS0}$  assignment after hard-Reset to FLASH SIMM / E<sup>2</sup>PROM.
13. External (off-board) tools Support which include:
  - Tool Identification
  - Tool Revision
  - Tool Status Information
14. S/W Option Identification.
15. Board revision code.
16. Power-on Reset via JTAG (optional).
17. PCI cards Present Detect and card type.
18. Local Bus Mode

Since part of the PQ2FADS-VRs' 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 exclusive enabled with on-board modules.

#### 4.13.1 BCSR0 - Board Control - Status Register 0

The BCSR0 is a control register on the PQ2FADS-VR. It is accessed at **offset 0** from BCSR base

address. It may be read or written at any time<sup>1</sup>. BCSR0 gets its defaults upon Power-On reset. BCSR0 fields are described in Table 4-8.

**Table 4-8. BCSR0 Description**

BIT	MNEMONIC	Function	PON DEF	ATT.
0	PBI	<b>Page Base Interleaving.</b> In 60X mode (i.e., with L2-Cache), this bit should reflect (system programmer responsibility) the state of PBI bit in PSDMR. In Single PQ2 Mode (i.e., without L2-Cache), this bit has no effect.	0	R/W
1	Reserved	<b>S</b>	0	R/W
2	L2C_INH	<b>L2 Cache Inhibit.</b> When this bit is active ( <b>low</b> ), 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 PQ2FADS-VR that does not have an L2 Cache installed.	0	R,W
3	L2C_FLUSH	<b>L2 Cache Flush.</b> When this bit is active ( <b>low</b> ) 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 PQ2FADS-VR that does not have an L2 Cache installed.	1	R,W
4	L2C_LOCK	<b>L2 Cache Lock.</b> When this bit is active ( <b>low</b> ), 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 PQ2FADS-VR that does not have an L2 Cache installed.	1	R,W
5	L2C_CLEAR	<b>L2 Cache Clear.</b> When this bit is active ( <b>Low</b> ) 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 PQ2FADS-VR.	1	R,W
6 - 31	Reserved	Un-implemented	0	R

**4.13.2 BCSR1 - Board Control - Status Register 1**

The BCSR1 is a control register on the PQ2FADS-VR. It is accessed at **offset 4** from BCSR base address. It may be read or written at any time<sup>2</sup>. BCSR1 gets its defaults upon Power-On reset. The

1. Provided that BCSR is not disabled.  
 2. Provided that BCSR is not disabled.

fields are described in Table 4-9.

**Table 4-9. BCSR1 Description**

BIT	MNEMONIC	Function	PON DEF	ATT.
0	Conf_Word	<b>Config_Source.</b> When asserted ( <b>low</b> ) Hard Reset Configuration Word is sourced from the BCSR. When negated, Hard Reset Configuration Word is sourced from the FLASH/EEPROM. The assignments selection is done via a dedicated jumper JP7.	0	R
1	FLASH_CS0	<b>FLASH_CS0.</b> When asserted ( <b>low</b> ) CS0 is assigned to the FLASH SIMM and CS4 is assigned to E <sup>2</sup> PROM. When negated, CS0 is assigned to the E <sup>2</sup> PROM and CS4 is assigned to the FLASH SIMM. The assignments selection is done via a dedicated jumper.	0	R
2	ATM_EN	<b>ATM Port Enable.</b> When asserted ( <b>low</b> ) the ATM UNI chip (PM5350) connected to FCC1 is enabled for transmission and reception. When negated, the ATM transceiver is in standby mode and its associated buffers <sup>a</sup> 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	<b>ATM Port Reset.</b> When asserted ( <b>low</b> ), the ATM port transceiver is in reset state. This line is driven also by HRESET signal of the PQ2.	1	R,W
4	FETHIEN1	<b>Fast Ethernet Port 1 Initial Enable.</b> When asserted ( <b>low</b> ) the DM9161's MII port, residing on FCC2, is enabled after Power-Up or after FETH_RST is negated. When negated ( <b>high</b> ), the DM9161'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 DM9161 and MII isolation may be controlled via MDIO 0.10 bit.	1	R,W
5	FETH1_RST	<b>Fast Ethernet port 1 Reset.</b> When active ( <b>low</b> ) the DM9161 is reset. This line is also driven by HRESET signal of the PQ2. Since MDDIS pin of the DM9161 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	<b>RS232 port 1 Enable.</b> When asserted ( <b>low</b> ) the RS232 transceiver for port 1, 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	<b>RS232 port 2 Enable.</b> When asserted ( <b>low</b> ) the RS232 transceiver for port 2, 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	0	R

a. Required for voltage levels adaptation.

### 4.13.3 BCSR2 - Board Control - Status Register - 2

BCSR2 is a status register which is accessed at offset 8 from the BCSR base address. Its a read-

only register which may be read at any time<sup>1</sup>. BCSR2s' various fields are described in [Table 4-10](#).

**Table 4-10. BCSR2 Description**

BIT	MNEMONIC	Function	PON DEF	ATT.
0 - 7	TSTAT(0:7)	<b>Tool Status (0:7)</b> . 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.	-	R
8 - 11	TOOLREV(0:3)	<b>TOOL Revision (0:3)</b> . This field may contain the revision code of an external tool connected to the PQ2. The various combinations of this field will be described per each tool users' manual. These signals are available at the System expansion connector. The revision option for the external tools are shown in <a href="#">Table 4-16</a> .		R
12 - 15	EXTTOLI(0:3)	<b>External Tools Identification</b> . 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 <a href="#">Table 4-13</a> .	-	R
16 - 17	SWOPT(0:1) <sup>a</sup>	<b>Software Option (0:1)</b> . This field shows the state of a dedicated dip-switches providing an option to manually change a program flow.	0	R
18 - 19	L2CSIZE(0:1)	<b>L2 Cache Size (0:1)</b> . This field encodes the size of the L2 Cache, present on the PQ2FADS-VR. For the encoding of the various cache sizes see <a href="#">Table 4-17</a> .	-	R
20 - 21	BVERN(0:1)	<b>Board Version Number (0:1)</b> . This field represents the version code, hard-assigned to the PQ2FADS-VR. See <a href="#">Table 4-14</a> ., for version encoding.	11	R
22 - 23	BREVN(0:1)	<b>Board Revision Number (0:1)</b> . This field represents the revision code, hard-assigned to the PQ2FADS-VR. See <a href="#">Table 4-15</a> ., for revisions' encoding.	-	R
24	SWOPT2	<b>Software Option 2</b> . This is the LSB of the field. Shows the state of a dedicated dip-switch providing an option to manually change a program flow.	0	R
25 - 27	FLASH_PD(7:5)	<b>Flash Presence Detect(7:5)</b> . 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(7:5) see <a href="#">Table 4-11</a> .	-	R
28 - 31	FLASH_PD(4:1)	<b>Flash Presence Detect(4:1)</b> . 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 <a href="#">Table 4-12</a> .	-	R

a. There is additional bit to this field. See next on the same table.

1. Provided that BCSR is not disabled.

**Table 4-11. 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
101 - 111	Not Supported

**Table 4-12. 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 4-13. EXTTOOLI(0:3) Assignment**

EXTTOOLI(0:3)	External Tool
0	T/ECOM - PQ2 Communication tool
1	Reserved
2	T1 Circuit Emulation Tool
3 - E	Reserved
F	Tool Non Existent

**Table 4-14. PQ2 Board Version Encoding**

<b>Version Number (0:1) [Hex]</b>	<b>PQ2 Board Version</b>
0	PQ2FADS-ZU
1	Reserved
2	PQ2FADS-VR
3	PQ27e ADS

**Table 4-15. PQ2 Board Revision Encoding**

<b>Revision Number (0:1) [Hex]</b>	<b>PQ2 Board Revision</b>
0	ENG (Engineering)
1	PILOT
2	A
3	Reserved

**Table 4-16. External Tool Revision Encoding**

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

**Table 4-17. L2 Cache Size Encoding**

<b>L2CSIZE(0:1)</b>	<b>L2 Cache Size</b>
'00'	Reserved
'01'	512 KBytes
'10'	Reserved
'11'	No L2 Cache

#### 4.13.4 BCSR3 - Board Control - Status Register 3

BCSR3 is a control register which is accessed at **offset 0xC** from the BCSR base address. Its a read- write register which may be read or written at any time<sup>1</sup>. BCSR3s' various fields are described in [Table 4-19](#).

**Table 4-18. BCSR3 Description**

BIT	MNEMONIC	Function	PON DEF	ATT.
0	USB_EN	<b>USB Port Enable.</b> When asserted ( <b>low</b> ) the USB chip connected to SCC4 is enabled for transmission and reception. When negated, the USB transceiver is in standby mode and its associated buffers <sup>a</sup> are in tri-state mode, freeing all its i/f signals for off-board use via the expansion connectors.	1	R/W
1	USB_HI_SPEED	<b>USB Hi Speed.</b> When asserted ( <b>low</b> ) the USB chip connected to SCC4 is set for hi speed (12 Mbps) transmission and reception. When negated, the USB transceiver is set to low speed (1.5 Mbps) transmission and reception	0	R/W
2	USBVCC0	<b>USB Port VCC EN.</b> When asserted ( <b>high</b> ), 5V power is applied to the USB Bus. When negated, power to the USB port is disconnected.	0	R/W
3	FETHIEN2	<b>Fast Ethernet Port 2 Initial Enable.</b> When asserted ( <b>low</b> ) the DM9161's MII port, residing on FCC3, is enabled after Power-Up or after FETH_RST is negated. When negated ( <b>high</b> ), the DM9161'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 DM9161 and MII isolation may be controlled via MDIO 0.10 bit.	1	R/W
4	FETH2_RST	<b>Fast Ethernet port 2 Reset.</b> When active ( <b>low</b> ) the DM9161 is reset. This line is also driven by HRESET signal of the PQ2. Since MDDIS pin of the DM9161 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
5	ATM16	<b>ATM 16 bit UTOPIA.</b> When asserted ( <b>low</b> ) the UTOPIA is set for 16 bit. When negated ( <b>high</b> ), the UTOPIA is set for 8 bit..	1	R/W
6	ATM_SINGLE_PHY	<b>ATM SINGLE PHY.</b> When asserted ( <b>low</b> ) the UTOPIA is set to Multi PHY. When negated ( <b>high</b> ), the UTOPIA is set for Single PHY.	1	R/W
7	PCI_MODE	<b>PCI_MODE.</b> When asserted ( <b>low</b> ) the Local Bus function is set to PCI. When negated ( <b>high</b> ), the Local Bus is set for Local Bus SDRAM.		R
8-31	Reserved	un-implemented		

a. Required for voltage levels adaptation.

#### 4.13.5 BCSR4 - Board Control - Status Register 4

BCSR4 is a status register which is accessed at **offset 0x10** from the BCSR base address. Its a

1. Provided that BCSR is not disabled.

read- only register which may be read at any time<sup>1</sup>. BCSR4s' various fields are described in [Table 4-19](#).

**Table 4-19. BCSR4 Description**

BIT	MNEMONIC	Function	PON DEF	ATT.
0 - 1	PCI0_PRSENT(0:1)	<b>PCI Slot 0 Present (0:1)</b> . This field holds a code that tells whether a PCI expansion board is plugged in PCI slot 0 and the total power requirements of the board according to the PCI spec. The different expansion board types are listed in <a href="#">Table 4-20</a> .	11	R
2 - 3	PCI1_PRSENT(0:1)	<b>PCI Slot 1 Present (0:1)</b> . This field holds a code that tells whether a PCI expansion board is plugged in PCI slot 1 and the total power requirements of the board according to the PCI spec. The different expansion board types are listed in <a href="#">Table 4-20</a> .	11	R
4 - 5	PCI2_PRSENT(0:1)	<b>PCI Slot 2 Present (0:1)</b> . This field holds a code that tells whether a PCI expansion board is plugged in PCI slot 2 and the total power requirements of the board according to the PCI spec. The different expansion board types are listed in <a href="#">Table 4-20</a> .	11	R
6	M66EN	<b>66MHz Enable</b> . This field shows if one of the expansion boards used is not capable of operating in 66MHz mode: '1' - All expansion boards are 66MHz capable '0' - One of the expansion boards is not 66MHz capable	1	R
7	PCI_MODCK	<b>PCI_MODCK</b> . This field shows the PCI bus clock settings.	-	R
8-31	Reserved	un-implemented		

**Table 4-20. PCI Board Present Signal Definitions**

PCIx_PRSENT (0:1) [Hex]	Expansion Configuration
0	Expansion board present, 7.5W maximum
1	Expansion board present, 25W maximum
2	Expansion board present, 15W maximum
3	No expansion board present

### 4.13.6 BCSR5 and BCSR7- Board Control - Status Register 3 & 5

BCSR5 to BCSR7 are additional control / status registers which may be accessed as a word at offset 0x14 to 0x1C 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 board. The description of BCSR3 and

1. Provided that BCSR is not disabled.

BCSR5 is shown in [Table 4-21](#).

**Table 4-21. BCSR5 to BCSR7 Description**

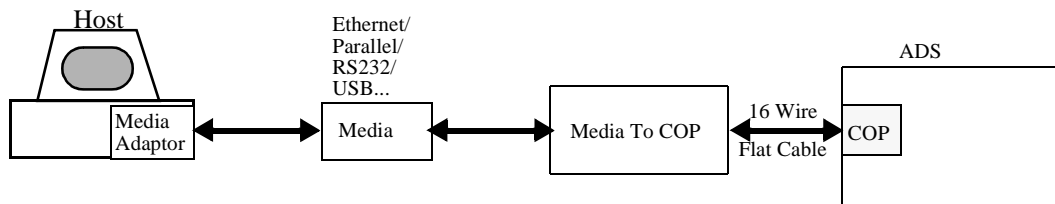
BIT	MNEMONIC	Function	PON DEF	ATT.
0 - 31	Reserved	Un Implemented	-	-

## 4.14 COP/JTAG Port

The COP - Control Observation Port, is part of the PQ2'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<sup>1</sup>, for extensive system debug.

There are several third party debug solutions on the market. These debug-stations may be connected to the host computer via either Ethernet, Parallel-Port, RS232 or any other media.

The debug station connection scheme is shown in [Figure 4-12](#).



**Figure 4-12. Debug Station Connection Schemes**

To support debug station connection to the COP/JTAG port, a 16 pin generic header connector is provided on the PQ2FADS-VR, carrying the COP/JTAG signals as well as additional signals aiding in system debug. The pinout of this connector, which is a general Motorola recommendation for including a COP/JTAG port in a design, is shown in [Figure 4-13](#) and detailed

1. Not provided with the PQ2FADS-VR.

in Table 4-22..

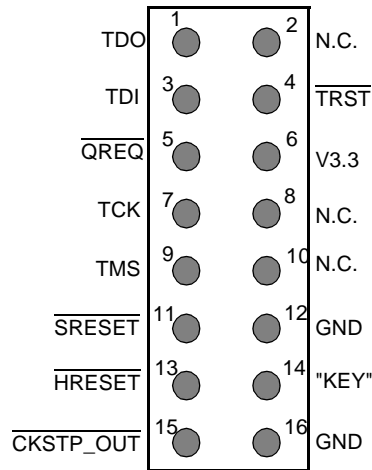


Figure 4-13. COP/JTAG Port Connector

Table 4-22. COP/JTAG Port Signals Description

Pin No.	Signal Name	Attribute	Description
1	TDO	O	Transmit Data Out. This the JTAG's serial data output driven by Falling edge of TCK.
2	N.C.	-	Not Connected.
3	TDI	I	Transmit Data In. This is the JTAG serial data input, sampled by the PQ2 on the rising edge of TCK. This line is pulled up internally by the PQ2.
4	$\overline{\text{TRST}}$	I	Test port Reset (L). When this signal is active (Low), it resets the JTAG logic. This line is pull-down on the PQ2FADS-VR with a 1K $\Omega$ resistor, to provide constant reset of the JTAG logic.
5	$\overline{\text{QREQ}}$	O	Quiescent Request (L). When asserted (low), this line indicates that the PQ2 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 PQ2 JTAG port. 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 PQ2.
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 machine. This line is pulled up internally by the PQ2.
10	N.C.	-	Not Connected.

**Table 4-22. COP/JTAG Port Signals Description**

Pin No.	Signal Name	Attribute	Description
11	$\overline{\text{SRESET}}$	I/O, O.D.	Soft Reset (L). This is the PQ2's soft reset which is in fact a non-maskable interrupt, making the PowerPC take the reset exception from the reset vector. This line may be driven by the PQ2 as well during soft-reset sequence, for 512 system clocks. This line is pulled up on the PQ2FADS-VR with a 1K $\Omega$ resistor. When driven externally, it <b>MUST be driven with an Open Drain gate. Failure in doing so might result in permanent damage to the PQ2 and / or to board logic.</b>
12	GND	O	Digital GND. Main GND plane.
13	$\overline{\text{HRESET}}$	I/O, O.D.	PQ2's Hard Reset (L). When asserted by an external H/W, generates Hard-Reset sequence for the PQ2. During that sequence, asserted by the MPC for 512 system clocks. Pulled Up on the PQ2FADS-VR using a 1K $\Omega$ resistor. When driven by an external tool, <b>MUST be driven with an Open Drain gate. Failure in doing so might result in permanent damage to the PQ2 and / or to board logic.</b>
14	N.C.	-	Not Connected.
15	$\overline{\text{XBR3}}$ ( $\overline{\text{CKSTOP\_OUT}}$ )	I/O	Normally configured as $\overline{\text{XBR3}}$ which has no function with this connector. May be configured as $\overline{\text{CKSTOP\_OUT}}$ - Check Stop Out (L). When asserted (Low) indicates that the PQ2 core has entered a Check-Stop state.
16	GND	O	Digital GND. Main GND plane.

## Memory Map and Initialization

### 5.1 Memory Map

All accesses to PQ2FADS-VR's memory slaves are controlled by the PQ2'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 existence, size, delay and type of the FLASH memory SIMM mounted on board and decides on the assignments of  $\overline{CS0}$  and  $\overline{CS4}$  (E<sup>2</sup>PROM and FLASH) and programs the memory controller accordingly. The SDRAM, E<sup>2</sup>PROM and the FLASH memory, respond to all types of memory access i.e., program / supervisory, program / data and DMA.

This memory map is a recommended memory map and since it is a "soft" map, devices' address may be moved about the map, to the convenience of any user. There are actually two memory maps which depend on the device assigned to  $\overline{CS0}$  (regardless of the Hard Reset Configuration Word source). The memory address for the device assigned to  $\overline{CS0}$  is always the same as determined in the Hard-Reset configuration word. Since the FLASH and E<sup>2</sup>PROM require different memory spaces, different memory maps are devised for each case. For details see [Table 5-1.](#) and [Table 5-2.](#)

**Table 5-1. PQ2FADS-VR Memory Map - FLASH (or BCSR) as Boot Device**

Address Range	Memory Type	Device Name		Port Size	Memory Size
00000000 - 01FFFFFF	60x SDRAM	32MByte	64MByte	64	64 MByte
01000000 - 03FFFFFF					
04000000 - 044FFFFFF	Empty Space	Optional 4MByte local bus SDRAM for legacy support		-	5 MByte

**Table 5-1. PQ2FADS-VR Memory Map - FLASH (or BCSR) as Boot Device**

Address Range	Memory Type	Device Name	Port Size	Memory Size
04500000 - 04507FFF	BCSR(0:7) <sup>a</sup>		32	32 KByte
04500000 - 04507FE3	BCSR0			4 Byte
04500004 - 04507FE7	BCSR1			4 Byte
04500008 - 04507FEB	BCSR2			4 Byte
0450000C - 04507FEF	BCSR3			4 Byte
04500010 - 04507FF3	BCSR4			4 Byte
04500014 - 04507FF7	BCSR5			4 Byte
04500018 - 04507FFB	BCSR6			4 Byte
0450001C - 04507FFF	BCSR7			4 Byte
04508000 - 045FFFFFFF	Empty Space		-	~1 MByte
04600000 - 04607FFF <sup>b</sup>	ATM UNI Proc. Control	PMC5384 M/P I/F	8	32 KByte
04608000 - 046FFFFFFF	Empty Space		-	~1 MByte
04700000 <sup>c</sup> - 0471FFFF	PQ2 Internal MAP <sup>d</sup>		32	128 KByte
04720000 - 0472FFFF	Empty Space		-	64 KByte
04730000 - 04737FFF	PCI Interrupt Controller		32	32 KByte
04738000 - 047FFFFFFF	Empty Space		-	~800 KByte
04800000 - 04FFFFFFF	PCI Memory	Agents PIMMR (via PCI Direct)		~ 8 MByte
05000000 - 7FFFFFFF	Empty Space	Tool Board is located at 60000000 and 70000000		~ 2 GByte
80000000 - BFFFFFFF	PCI Memory	PCI Agents GPL WIndows	32	1 Gbyte

**Table 5-1. PQ2FADS-VR Memory Map - FLASH (or BCSR) as Boot Device**

Address Range	Memory Type	Device Name		Port Size	Memory Size
C0000000 - C1FFFFFF	Empty Space			-	32 MByte
C2000000 <sup>e</sup> - C2007FFF	E <sup>2</sup> PROM	ATMEL AT28HC64B		8	32 KByte
C2008000 - CFFFFFFF	Empty Space				~200 MByte
D0000000 - D07FFFFFFF	Empty Space				8 MByte
D0800000 - FFFFFFFF	Empty Space				~1 GByte
FE000000 <sup>f</sup> - FFFFFFFF	Flash SIMM		32M SIMM - SM73288	32	32 MByte
FF000000 - FF7FFFFFFF			16M SIMM - SM73248		
FF800000 - FFFFFFFF		8M SIMM - SM73228			

- The device appears repeatedly in multiples of its port-size (in bytes) X depth. E.g., BCSR0 appear at memory locations 4700000, 4700020, 4700040..., while BCSR1 appears at 4700004, 4700024, 4700044... and so on.
- The internal space of the ATM UNI control port is 256 bytes, however, the minimal block size that may be controlled by the GPCM is 32 KBytes.
- Initially at h0F000000 - h0F00FFFF, set by hard reset configuration.
- Refer to the PQ2 User's Manual for complete description of the internal memory map.
- An 8 Kbyte device is used (16 Kbyte and 32 Kbyte devices can also be used) so it appears repeatedly in 8Kbyte multiples starting from C2000000.
- Set by hard-reset configuration.

**Table 5-2. PQ2FADS-VR Memory Map - E<sup>2</sup>PROM as Boot Device**

Address Range	Memory Type	Device Name		Port Size	Memory Size
00000000 - 00FFFFFF	SDRAM DIMM	32 MByte	64 MByte	64	64 MByte
01000000 - 03FFFFFF					
04000000 - 044FFFFFFF	Empty Space			-	5 MByte

**Table 5-2. PQ2FADS-VR Memory Map - E<sup>2</sup>PROM as Boot Device**

Address Range	Memory Type	Device Name	Port Size	Memory Size
04500000 - 04507FFF	BCSR(0:7) <sup>a</sup>		32	32 KByte
04500000 - 04507FE3	BCSR0			4 Byte
04500004 - 04507FE7	BCSR1			4 Byte
04500008 - 04507FEB	BCSR2			4 Byte
0450000C - 04507FEF	BCSR3			4 Byte
04500010 - 04507FF3	BCSR4			4 Byte
04500014 - 04507FF7	BCSR5			4 Byte
04500018 - 04507FFB	BCSR6			4 Byte
0450001C - 04507FFF	BCSR7			4 Byte
04508000 - 045FFFFFFF	Empty Space		-	~1 MByte
04600000 - 04607FFF <sup>b</sup>	ATM UNI Proc. Control	PMC5384 M/P I/F	8	32 KByte
04608000 - 046FFFFFFF	Empty Space		-	~1 MByte
04700000 <sup>c</sup> - 0471FFFF	PQ2 Internal MAP <sup>d</sup>		32	128 KByte
04720000 - 0472FFFF	Empty Space		-	64 KByte
04730000 - 04737FFF	PCI Interrupt Controller		32	32 KByte
04738000 - 0477FFFF	Empty Space		-	~800 KByte
04800000 - 04FFFFFFF	PCI Memory	Agents PIMMR (via PCI Direct)		~ 8 MByte
05000000 - 7FFFFFFF	Empty Space	Tool Board is located at 60000000 and 70000000		~ 2 GByte
80000000 - BFFFFFFF	PCI Memory	PCI Agents GPL WIndows	32	1 Gbyte

**Table 5-2. PQ2FADS-VR Memory Map - E<sup>2</sup>PROM as Boot Device**

Address Range	Memory Type	Device Name			Port Size	Memory Size
C0000000 - C1FFFFFF	Empty Space				-	32 MByte
C2000000 - C2FFFFFF	Flash SIMM			32M SIMM - SM73288	32	32 MByte
C3000000 - C37FFFFFFF			16M SIMM - SM73248			
C3800000 - C3FFFFFFF		8M SIMM - SM73228				
C4000000 - CFFFFFFF	Empty Space					~200 MByte
D0000000 - D07FFFFFFF	Empty Space					8 MByte
D0800000 - FFFFDFFF	Empty Space					~1 GByte
FFF00000 <sup>e</sup> - FFFFFFFF	E <sup>2</sup> PROM	ATMEL AT28HC64B			8	32 KByte

- The device appears repeatedly in multiples of its port-size (in bytes) X depth. E.g., BCSR0 appears at memory locations 47000000, 47000020, 47000040..., while BCSR1 appears at 47000004, 47000024, 47000044... and so on.
- The internal space of the ATM UNI control port is 256 bytes, however, the minimal block size that may be controlled by the GPCM is 32 KBytes.
- Initially at h0F000000 - h0F00FFFF, set by hard reset configuration.
- Refer to the PQ2 User's Manual for complete description of the PQ2's internal memory map.
- An 8 Kbyte device is used (16 Kbyte and 32 Kbyte devices can also be used) so it appears repeatedly in 8Kbyte multiples starting from FFF00000.

## 5.2 PQ2 Register Programming

The PQ2 provides the following functions on the PQ2FADS-VR:

- System functions which include:
  - PPC Bus SDRAM Controller
  - Local Bus Host to PCI Bridge
  - Chip Select generator
- Communication functions which include:
  - ATM SAR
  - Dual Fast Ethernet controller
  - UART for terminal or host computer connection
  - USB Controller

The internal registers of the PQ2 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 PQ2 User’s Manual.

### 5.2.1 System Initializations

The Power-On Reset Configuration word is set in the BCSR or FLASH or in the E<sup>2</sup>PROM. There are two configuration words - one for the BCSR and FLASH (when it is assigned to  $\overline{CS0}$ ) and the other to the E<sup>2</sup>PROM (when it is assigned to  $\overline{CS0}$ ). The two configurations are detailed in [Table 5-3](#) and [Table 5-4](#). respectively.

**Table 5-3. BCSR/FLASH Power On Reset Configuration<sup>a</sup>**

Flash Address [hex]	Init Value[hex]	Description
0	0C / (1C <sup>b</sup> )	Internal arbitration, Internal memory controller, Core enabled, Single PQ2 (60X Bus mode <sup>b</sup> ), 32 Bit boot port size, Exceptions vectored to 0xFFFxxxxx, Internal space 64 bit slave for external master.
8	B2	L2cache signals configured as BADDRx lines, DP(1:7) configured as L2 cache I/F and IRQ(6:7),Initial internal space @ 0x0F000000
10	32 <sup>c</sup> / (36 <sup>d</sup> )	Boot memory space @ 0xFE000000 - 0xFFFFFFFF, $\overline{ABB}/\overline{IRQ2}$ pin is $\overline{ABB}$ , $\overline{DBB}/\overline{IRQ3}$ pin is $\overline{DBB}$ , No masking on bus request lines, Local bus pins function as (in BCSR) PCI (in FLASH), PCI is boot master, AP(1;3) configured as BNKSEL(0:2), $\overline{APE}$ configured as $\overline{IRQ7}$ and $\overline{CS11}$ as $\overline{CS11}$ .
18	45	$\overline{CS10}$ configured as $\overline{BCTL1}$

- a. Programmed into the Flash (E<sup>2</sup>PROM) memory in addresses 0x0, 0x8, 0x10 & 0x18
- b. With L2 Cache
- c. Programmed in BCSR - Local Bus pins function is Local Bus
- d. Programmed in FLASH - Local Bus pins function is PCI

**Table 5-4. E<sup>2</sup>PROM Power On Reset Configuration<sup>a</sup>**

EEPROM Address [hex]	Init Value[hex]	Description
0	04 / (14 <sup>b</sup> )	Internal arbitration, Internal memory controller, Core enabled, Single PQ2 (60X Bus mode <sup>b</sup> ), 8 Bit Boot size, Exceptions vectored to 0xFFFxxxxx, Internal space 64 bit slave for external master.
8	B2	L2cache signals configured as BADDRx lines, DP(1:7) configured as L2 cache I/F and IRQ(6:7),Initial internal space @ 0x0F000000

**Table 5-4. E<sup>2</sup>PROM Power On Reset Configuration<sup>a</sup>**

EEPROM Address [hex]	Init Value[hex]	Description
10	36	Boot memory space @ 0xFE00000 - 0xFFFFFFFF, $\overline{ABB}/\overline{IRQ2}$ pin is $\overline{ABB}$ , $\overline{DBB}/\overline{IRQ3}$ pin is $\overline{DBB}$ , No masking on bus request lines, Local bus pins function as PCI, PCI is boot master, AP(1;3) configured as BNKSEL(0:2), APE configured as $\overline{IRQ7}$ and $\overline{CS11}$ as $\overline{CS11}$ .
18	45	$\overline{CS10}$ configured as $\overline{BCTL1}$

- a. Programmed into the E<sup>2</sup>PROM in addresses 0x0, 0x8, 0x10 & 0x18
- b. With L2 Cache

**Table 5-5. SIU REGISTERS' PROGRAMMING**

Register	Init Value[hex]	Description
RMR	0001	Check-Stop Reset enabled.
IMMR	04700000	Internal space @ 0x04700000
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	100C0000 (88444000 <sup>a</sup> )	Single PQ2 (60X Bus mode <sup>a</sup> ), 1 wait-states on address tenure, No L2Cache (L2Cache assumed <sup>a</sup> ), 1 clock hit delay (when L2cache available), 1-level Pipeline depth, Extended transfer mode enabled for PCC, Extended transfer mode disabled for Local Buses, Odd parity for PPC & Local Buses, External Master delay enabled, Internal space responds as 64 bit slave for external master (not relevant for this application).

- a. With L2 Cache

### 5.2.2 Memory Controller Registers Programming

The memory controller on the PQ2FADS-VR is initialized to 100MHz operation, i.e., registers' programming is based on 100MHz timing calculation (it will also work for slower bus speeds but the timing will have to be optimized). There are two possible initializations for the memory controller:

- Flash SIMM is assigned to  $\overline{CS0}$  and E<sup>2</sup>PROM is assigned to  $\overline{CS4}$ .
- Flash SIMM is assigned to  $\overline{CS4}$  and E<sup>2</sup>PROM is assigned to  $\overline{CS0}$ .

Both options are shown in [Table 5-6.](#)and [Table 5-7.](#)

**Table 5-6. Memory Controller Initializations For 100Mhz - FLASH as Boot Device**

Reg.	Device Type	Bus	Init Value [hex]	Description
BR0	SM73228XG1JHBG0 by Smart Modular Tech.	PPC	FF801801	Base at FF800000, 32 bit port size, no parity, GPCM
	SM73248XG2JHBG0 by Smart Modular Tech.		FF001801	Base at FF000000, 32 bit port size, no parity, GPCM
	SM73288XG4JHBG0 by Smart Modular Tech.		FE001801	Base at FE000000, 32 bit port size, no parity, GPCM
OR0	SM73228XG1JHBG0 by Smart Modular Tech.	PPC	FF800876	8MByte block size, CS early negate, 11 w.s., Timing relax
	SM73248XG2JHBG0 by Smart Modular Tech.		FF000876	16MByte block size, CS early negate, 11 w.s., Timing relax
	SM73288XG4JHBG0 by Smart Modular Tech.		FE000876	32MByte block size, CS early negate, 11 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	SDRAM MT48LC4M32B2 by MICRON	PPC	00000041	Base at 0, 64 bit port size, no parity, SDRAM machine 1
OR2			FE002EC0	32MByte block size, 4 banks per device, row starts at A7, 12 row lines, internal bank interleaving allowed, normal AACK operation
BR4	E <sup>2</sup> PROM	PPC	C2000801	Base at C2000000, 8 bit port size, write protect disabled, no parity, GPCM
OR4	AT28HC64B-70JC by Atmel		FFFF8866	32 KByte block size, $\overline{CS}$ output half a clock after address, all types access, 6 w.s., Timing relax
BR5	PM5384 - ATM UNI	PPC	04600801	Base at 04600000, 8 bit port size, no parity, GPCM on PPC bus.
OR5			FFFF8E56	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.
BR8	PCI Interrupt Controller	PPC	04731801	Base at 04730000, 32 bit port size, no parity, GPCM on PPC bus.
OR8			FFFF8010	32 KByte block size, all types access, 1 w.s.

**Table 5-7. Memory Controller Initializations For 100Mhz - E<sup>2</sup>PROM as Boot Device**

Reg.	Device Type	Bus	Init Value [hex]	Description
BR0	E <sup>2</sup> PROM	PPC	FFF00801	Base at FFFFE000, 8 bit port size, write protect disabled, no parity, GPCM
OR0	AT28HC64B-70JC by Atmel		FFFF8866	32 KByte block size, $\overline{CS}$ output half a clock after address, all types access, 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	SDRAM MT48LC4M32B2 by MICRON	PPC	00000041	Base at 0, 64 bit port size, no parity, SDRAM machine 1
OR2			FE002EC0	32MByte block size, 4 banks per device, row starts at A7, 12 row lines, internal bank interleaving allowed, normal AACK operation
BR4	SM73228XG1JHBG0 by Smart Modular Tech.	PPC	C3801801	Base at C3800000, 32 bit port size, no parity, GPCM
	SM73248XG2JHBG0 by Smart Modular Tech.		C3001801	Base at C3000000, 32 bit port size, no parity, GPCM
	ASM73288XG4JHBG0 by Smart Modular Tech.		C2001801	Base at C2000000, 32 bit port size, no parity, GPCM
OR4	SM73228XG1JHBG0 by Smart Modular Tech.	PPC	FF800876	8MByte block size, CS early negate, 11 w.s., Timing relax
	SM73248XG2JHBG0 by Smart Modular Tech.		FF000876	16MByte block size, CS early negate, 11 w.s., Timing relax
	SM73288XG4JHBG0 by Smart Modular Tech.		FE000876	32MByte block size, CS early negate, 11 w.s., Timing relax
BR5	PM5384 - ATM UNI	PPC	04600801	Base at 04600000, 8 bit port size, no parity, GPCM on PPC bus.
OR5			FFFF8E56	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.
BR8	PCI Interrupt Controller	PPC	04731801	Base at 04730000, 32 bit port size, no parity, GPCM on PPC bus.
OR8			FFFF8010	32 KByte block size, all types access, 1 w.s.

**Table 5-8. Memory Controller Initializations For 100Mhz**

Reg.	Device Type	Bus	Init Value [hex]	Description
PSDMR	MT48LC4M32B2 (32 MByte)	PPC  Single PQ2 Bus Mode	C24B36A3	<b>Page</b> Based Interleaving, Refresh enabled, normal operation mode, address muxing mode 2, A14-A16 on BNKSEL, A8 on PSDA10, 8 clocks refresh recovery, 3 clocks precharge to activate delay, 3 clocks activate to read/write delay, 4 beat burst length, 2 clock last data out to precharge, 2 clock write recovery time, no extra cycle on address phase, normal timing for control lines, 3 clocks CAS latency.
PSRT	PPC Bus Sdram Supported	PPC	13	Divide MPTPR output by 20 (PSRT +1) Generates refresh every 8.2 $\mu$ sec, while 15.6 $\mu$ sec required. This will work also for 66MHz bus (12.4 $\mu$ sec).
MPTPR	All SDRAMs on board		2800	Divide Bus clock by 41 (MPTPR+1) (decimal)

## Physical Properties

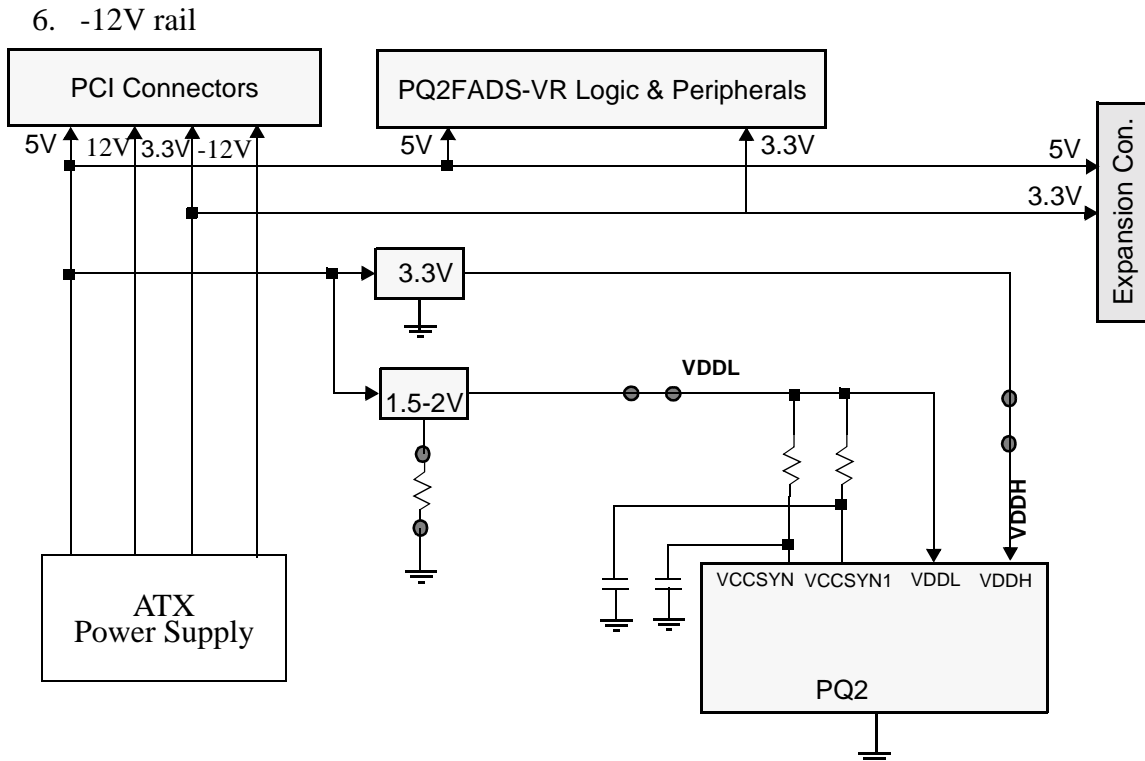
### 6.1 Power Supply

The board gets the power from the ATX Power Supply (it seats in an ATX Chassis). All the power rails on the board are derived from the ATX Power Supply. There are 4 power rails with the PQ2:

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

and there are 5 power rails on the PQ2FADS-VR:

1. VCC (5V) rail
2. Stand By (5V) rail
3. V3.3 (3.3V) rail
4. VDDL (1.7V-2.5V) rail
5. +12V rail



**Figure 6-1. PQ2FADS-VR Power Scheme**

To support off-board application development, 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 also depends on the current drawn by the PCI bus. The figures are shown in [Table 6-1](#).

**Table 6-1. Expansion Connectors Maximum Current Consumption**

Power Bus	Max. Current
VCC	TBD
V3.3	TBD

The PCI Standard specifies that each Add-In card should consume maximum 25Watt from all power sources combined. The maximum current consumption allowed per power source for a total of 25Watt according to the PCI Standard is shown in [Table 6-2](#).

**Table 6-2. Maximum Power Consumption Per Add-In Card**

Power Rail	Add-In Card
5V	5A Max. (system depended)
3.3V	7.6A Max. (system depended)
12V	500mA
-12V	100mA

### **6.1.1 5V Rail**

Some of the PQ2FADS-VR peripherals (not including the PCI Add-In cards which should be 3.3V ONLY on the PCI interface but can use 5V for other components on-board) reside on the 5V bus. Since the PQ2 is not 5V tolerant, buffering is provided between 5V peripherals and the PQ2, protecting the PQ2 from the higher voltage level.

### **6.1.2 3.3V Rail**

The PQ2, SDRAM, PCI Add-In cards, address and data buffers are powered by the 3.3 bus, which is produced from the ATX power supply.

### **6.1.3 5V Stand By Rail**

The 5V stand by power rail comes from the ATX Power Supply. Its' only use is to power the logic required to support the power button in the front panel on the ATX chassis.

### **6.1.4 VDDH Rail**

The PQ2's VDDH power bus (3.3V) is produced from the 5V bus using a low-voltage drop linear voltage regulator made by Micrel, the MIC29501-3.3BU.

A production option is made so that the level on this bus may be varied by means of trimming potentiometer - TR2. However this will requires replacing some components. This option allows the VDDH to be in the range of 3.0V - 3.6V.

### **6.1.5 VDDL Bus**

The PQ2'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 a jumper, which selects between different resistor values within the VDDL's variable regulator feedback network, while the fine tuning within a range is done by means of a trimming potentiometer.

Changing the voltage to the Core logic of the PQ2, 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.

### **6.1.6 12V Rail**

The 12V bus from the ATX Power Supply supports the PCI slots and the VPP 12V option from programming the FLASH.

### **6.1.7 -12V Rail**

The -12V bus from the ATX Power Supply supports the PCI slots.

## 6.2 Connectors

The PQ2FADS-VR has connectors attached, to serve the following functions:

1. ATX Power Supply
2. 100 / 10 - Base-T Ethernet ports
3. ATM 155Mbps port
4. RS232 port 1
5. RS232 port 2
6. CPM Expansion
7. COP / JTAG
8. Logic Analyzer Connectors
9. Programmable logic In System Programming (ISP)
10. PCI Connectors
11. System Expansion
12. USB Connector
13. Parallel Port Connector

### 6.2.1 ATX Power Connector

The ATX power connector is a 20-lead, standard ATX power connector. The female part is soldered to the PCB, while the plug is connected to the power supply. That way fast connection / disconnection of power is facilitated.

### 6.2.2 Fast Ethernet Port Connectors

The Ethernet connector on the PQ2FADS-VR is a Twisted-Pair (100/10-Base-T) connector. Use is done with 90° RJ45-8 connector.

### 6.2.3 ATM 155 Port Connection

The ATM 155 I/F to the media, is optical rather than electrical. Use is done with HP's HFBR 5805 optical I/F which is placed on the edge of the board for convenient connection.

### 6.2.4 RS232 Ports Connector

The RS232 port connector is a stacked 9 pin, 90°, female D-Type connector, which saves on board space (made of two connectors for two ports).

### 6.2.5 CPM Expansion Connector

The CPM expansion connectors carries all CPM pins, i.e., Port A to Port D signals. Use done with DIN 41612, 128 pin T.H. PCB connector, residing on the board, allowing convenient vertical connection to off-board tools. Power supply pins are also provided through this connector.

### 6.2.6 COP/JTAG Port Connector

The debug port connector is a Motorola standard COP/JTAG connector for the 60X processors

family. It is a generic 16 pin (2 X 8), Male, SMD, 90° protected header connector.

### 6.2.7 Logic Analyzer Connectors

To support fast connection to HPs' 16500 Logic Analyzers series for debugging purposes, a set of dedicated connectors is provided. Use is done with 38 pin, SMT, high density, matched impedance MICTOR connectors made by AMP.

These connectors carry the unbuffered 60X signals and should be placed as near to the PQ2 as possible to provide short PCB routes, yielding better reflections and crosstalk immunity. They do not carry the PCI bus signals due to the restrictions enforced by the PCI Standard. There are also connectors for the CPM signals.

### 6.2.8 Mach's In System Programming (ISP) Connector

This is a 10 pin generic 0.100" pitch header connector, providing In System Programming capability for Vantis made programmable logic on board.

### 6.2.9 PCI Connectors

A set of three standard PCI 3.3V keyed, 124 pin, 32-bit connectors is provided for connecting up to three PCI Add-In cards.

### 6.2.10 System Expansion Connector

The System Expansion Connector is a 128 pin, DIN 41612 connector, which provides a minimal system I/F required to interface to other tool-boards which may use the CPM Expansion Connector. This connector contains 16 bit (lower PPC bus) address lines, 16 bit (higher PPC bus) Data lines plus useful GPCM and UPM control lines.

### 6.2.11 USB Connector

The USB connector is standard type A USB connector.

### 6.2.12 Parallel Port Connector

The parallel connector is a standard 25 pin D-Type male connector.

## 6.3 PCB Layout

The PQ2FADS-VR layout was done in a manner suitable for high-frequency operation and it follows closely the PCI Standard layout recommendations. Following is a list of measures which are taken to meet this design goal:

- Traces are as short as possible.
- Clock signals and sensitive strobe signals are shielded and routed as a chain.
- Multilayer PCB, with ground and supply layers.
- PCI signals lengths and impedance according to PCI Standard Rev. 2.2.

## Support Information

In this chapter all information needed for support, maintenance and connectivity to the PQ2-ADS-PCI is provided.

### 7.1 Interconnect signals

The PQ2FADS-VR interconnects with external devices via the following set of connectors:

1. P1 - RS232 ports 1 and 2
2. P2 - USB Connector
3. P3 and P4 - 100 / 10 - Base-T Ethernet ports
4. P15 - COP / JTAG
5. P7 - CPM Expansion
6. P11, P12, P13, P14, P16, P17, P18, P23, P28,P29,P30 - Logic Analyzer MICTOR Connectors
7. P10, P8, P9 - PCI Slots Connectors
8. P27 - ATX Power Supply Connector
9. P26,P20 - Mach/Lattice and ALTERA In System Programming (ISP)
10. P25 - System Expansion
11. P31 - Parallel Port connector

#### 7.1.1 P1 - RS232 ports 1 and 2 Connectors

P1 is a dual 9 Pin D-Type connectors as described in [Table 7-1](#).

**Table 7-1. P1 Connector**

Pin No.	Signal Name	Description
1	CD	Carrier Detect <b>output</b> from the PQ2FADS-VR.
2	TX	Transmit Data <b>output</b> from the PQ2FADS-VR.
3	RX	Receive Data <b>input</b> to the PQ2FADS-VR.
4	DTR	Data Terminal Ready <b>input</b> to the PQ2FADS-VR.
5	GND	Ground signal of the PQ2FADS-VR.
6	DSR	Data Set Ready <b>output</b> from the PQ2FADS-VR.

**Table 7-1. P1 Connector**

Pin No.	Signal Name	Description
7	N.C.	No connect
8	CTS	Clear To Send <b>output</b> from the PQ2FADS-VR.
9	N.C.	No connect

**7.1.2 P3 and P4 - 100/10 - Base-T Ethernet port Connector**

P3 or P4 is a RJ-45 Type Connector for Twisted Pair Ethernet as described in [Table 7-2](#).

**Table 7-2. P3,P4 - 100/10 Base-T Ethernet Connector**

Pin No.	Signal Name	Description
1	TPTX	Twisted-Pair Transmit Data positive output from the PQ2FADS-VR.
2	TPTX~	Twisted-Pair Transmit Data negative output from the PQ2FADS-VR.
3	TPRX	Twisted-Pair Receive Data positive input to the PQ2FADS-VR.
4	N.C.	Not connected, Bob Smith terminated on the PQ2FADS-VR.
5		
6	TPRX~	Twisted-Pair Receive Data negative input to the PQ2FADS-VR.
7	N.C.	Not connected, Bob Smith terminated on the PQ2FADS-VR.
8		

**7.1.3 P16 - COP / JTAG Connector**

P16 is a Motorola standard COP / JTAG connector for the 60X processors family. It is a 16 pin protected header connector as described in [Table 7-3](#).

**Table 7-3. P16 - COP/JTAG Connector**

Pin No.	Signal Name	Attribute	Description
1	TDO	O	Transmit Data Output. This the PQ2's JTAG serial data output driven by Falling edge of TCK.
2	GND	O	Digital GND. Main GND plane.
3	TDI	I	Transmit Data In. This is the JTAG serial data input of the ADS, sampled on the rising edge of TCK.
4	TRST#	I	Test port Reset~ (L). When this signal is active (Low), it resets the JTAG logic of the PQ2. This line is pull-down on the ADS with a 1KΩ resistor, to provide constant reset of the JTAG logic.

Table 7-3. P16 - COP/JTAG Connector

Pin No.	Signal Name	Attribute	Description
5	QREQ#	O	Quiescent Request (L). When asserted (low), this line indicates that the PQ2 desires to enter low-power mode. This signal may be required by a debug station.
6	3v3	O	3.3V power supply bus.
7	TCK	I	Test port Clock. This clock shifts in / out data to / from the 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 PQ2.
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 machine. This line is pulled up internally by the PQ2.
10	GND	O	Digital GND. Main GND plane.
11	SRESET#	I/O, O.D.	Soft Reset (L). This is the PQ2'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 PQ2 as well during soft-reset sequence, for 512 system clocks. This line is pulled up on the ADS with a 1K $\Omega$ resistor. When driven externally, it <b>MUST be driven with an Open Drain gate. Failure to do so may result in permanent damage to the PQ2 and / or to ADS logic.</b>
12	GND	O	Digital GND. Main GND plane.
13	HRESET#	I/O, O.D.	PQ2's Hard Reset (L). When asserted by an external H/W, generates Hard-Reset sequence for the PQ2. During that sequence, asserted by the PQ2 for 512 system clocks. Pulled Up on the ADS using a 1K $\Omega$ resistor. When driven by an external tool, <b>MUST be driven with an Open Drain gate. Failure to do so may result in permanent damage to the PQ2 and / or to ADS logic.</b>
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 PQ2 core has entered a Check-Stop state.
16	GND	O	Digital GND. Main GND plane.

### 7.1.4 P7 - CPM Expansion Connector

P7 is a 128 pin, 90<sup>0</sup>, DIN 41612 connector, which allows for convenient expansion of the PQ2's serial ports. This connector contains all CPM pins plus power supply pins, to provide for easy tool connection as described in [Table 7-4](#).

**Table 7-4. P7 - CPM Expansion Connector**

Pin No.	Signal Name	Attribute	Description
A1	RS_RXD1 (PD31 <sup>a</sup> )	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.	PQ2's Port D 29 line. Parallel I/O or CPM dedicated line. May be used for any of its available functions.
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.	PQ2'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	I2CSDA (PD15)	I/O, T.S.	This signal is connected to the serial I <sup>2</sup> C data line. This line may be used off-board as an I <sup>2</sup> C data line for external I <sup>2</sup> C device.
A18	I2CSCL (PD14)	I/O, T.S.	This signal is connected to the serial I <sup>2</sup> C clock line. This line may be used off-board as an I <sup>2</sup> C clock line for external I <sup>2</sup> C device.

**Table 7-4. P7 - CPM Expansion Connector**

Pin No.	Signal Name	Attribute	Description
A19	PD13	I/O, T.S.	PQ2'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	ATMRCLKDIS	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	EXPVCC	O	5V Supply. Connected to ADS's 5V VCC plane. Provided as power supply for external tool.
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 PQ2, 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.

**Table 7-4. P7 - CPM Expansion Connector**

Pin No.	Signal Name	Attribute	Description
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 <sup>b</sup> 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.
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 <sup>st</sup> 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 <sup>c</sup> :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 <sup>c</sup> :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 <sup>b</sup> . 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 7-4. P7 - CPM Expansion Connector**

Pin No.	Signal Name	Attribute	Description
B23	PA9	I/O, T.S.	PQ2'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 <sup>d</sup> Transmit Error (H). When the Ethernet port is enabled, this signal will be asserted (High) by the PQ2 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 PQ2 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 7-4. P7 - CPM Expansion Connector**

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 PQ2 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.	PQ2'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. Provided to assist Circuit Emulation Tool. Enabled only when pin A29 of this connector is either not connected or driven low. Otherwise, Tri-stated.

**Table 7-4. P7 - CPM Expansion Connector**

Pin No.	Signal Name	Attribute	Description
C30	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C31			
C32			
D1	PC31	I/O, T.S.	PQ2'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 PQ2), 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.	PQ2'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 PQ2 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 PQ2 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.	PQ2's Port C (17:15) Parallel I/O lines. May be used to any of their available functions.
D16	PC16		
D17	PC15		

**Table 7-4. P7 - CPM Expansion Connector**

Pin No.	Signal Name	Attribute	Description
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 PQ2. 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.	PQ2's Port C 13 Parallel I/O line. May be used to any of its available functions.
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 PQ2. 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.	PQ2'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.	PQ2'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		

- a. The functions in parenthesis, are PQ2's parallel I/Os.
- b. Normally connected to ATMTFCLK on the ADS.
- c. MS bit.
- d. For that matter, both 100-Base-T and 10-Base-T.

**7.1.5 P9, P10, P11, P15, P17, P18, P19, P24, P28,P29, P30- Logic Analyzer MICTOR Connectors**

These are 38 pin, SMT, high density, matched impedance connector made by AMP. They contain

the PQ2 60X bus, 60X system and memory controller signals, unbuffered. The pinout of these connectors is shown in the schematics. For signal description of these connectors, see the PQ2 User's Manual.

### 7.1.6 P10, P8, P9 - PCI Connectors

These are 2 X 62 , 3.3V keyed, 32 bit PCI connectors. The pinout of each connector is available in [Table 7-5](#).

For signal descriptions for these connectors, see the PCI v2.2 Standard.

**Table 7-5. P12, P13, P14 - PCI Connectors**

Pin Number	Side B	Comments	Side A	Comments
1	-12V	Not Connected	TRST#	
2	TCK		+12V	
3	Ground		TMS	
4	TDO		TDI	
5	+5V		+5V	
6	+5V		INTA#	
7	INTB#	Not Connected	INTC#	Not Connected
8	INTD#	Not Connected	+5V	
9	PRSNT1#	Connected to GND	Reserved	Not Connected
10	Reserved	Not Connected	+3.3V(I/O)	
11	PRSNT2#	Connected to GND	Reserved	Not Connected
12	CONNECTOR KEY	3.3 volt key	CONNECTOR KEY	3.3 volt key
13	CONNECTOR KEY	3.3 volt key	CONNECTOR KEY	3.3 volt key
14	Reserved	Not Connected	3.3Vaux	Not Connected
15	Ground		RST#	
16	CLK		+3.3V (I/O)	
17	Ground		GNT#	
18	REQ#		Ground	
19	+3.3V (I/O)		PME#	Not Connected
20	AD[31]		AD[30]	
21	AD[29]		+3.3V	

### Table 7-5. P12, P13, P14 - PCI Connectors

Pin Number	Side B	Comments	Side A	Comments
22	Ground		AD[28]	
23	AD[27]		AD[26]	
24	AD[25]		Ground	
25	+3.3V		AD[24]	
26	C/BE[3]#		IDSEL	
27	AD[23]		+3.3V	
28	Ground		AD[22]	
29	AD[21]		AD[20]	
30	AD[19]		Ground	
31	+3.3V		AD[18]	
32	AD[17]		AD[16]	
33	C/BE[2]#		+3.3V	
34	Ground		FRAME#	
35	IRDY#		Ground	
36	+3.3V		TRDY#	
37	DEVSEL#		Ground	
38	Ground		STOP#	
39	LOCK#	Not Connected	+3.3V	
40	PERR#		SDONE	Not Connected
41	+3.3V		SBO#	Not Connected
42	SERR#		Ground	
43	+3.3V		PAR	
44	C/BE[1]#		AD[15]	
45	AD[14]		+3.3V	
46	Ground		AD[13]	
47	AD[12]		AD[11]	
48	AD[10]		Ground	
49	M66EN	Coupled to GND, using a 0.01uF capacitor	AD[09]	
50	Ground		Ground	
51	Ground		Ground	

**Table 7-5. P12, P13, P14 - PCI Connectors**

Pin Number	Side B	Comments	Side A	Comments
52	AD[08]		C/BE[0]#	
53	AD[07]		+3.3V	
54	+3.3V		AD[06]	
55	AD[05]		AD[04]	
56	AD[03]		Ground	
57	Ground		AD[02]	
58	AD[01]		AD[00]	
59	+3.3V (I/O)		+3.3V (I/O)	
60	ACK64#	Not Connected	REQ64#	Not Connected
61	+5V		+5V	
62	+5V		+5V	

**7.1.7 P5 - ATX Power Supply Connector**

This is a standard ATX Form Factor Power Connector as described in [Table 7-6](#).

**Table 7-6. P5 - ATX Power Supply Connector**

Pin	Signal	Pin	Signal
1	+3.3VDC	11	+3.3VDC-Sense
2	+3.3VDC	12	-12VDC
3	Ground	13	Ground
4	+5VDC	14	$\overline{\text{Power\_On}}$
5	Ground	15	Ground
6	+5VDC	16	Ground
7	Ground	17	Ground
8	Power_OK	18	-5VDC
9	+5VStand_By	19	+5VDC
10	+12VDC	20	+5VDC

**7.1.8 P21,P23,P25 - Mach/Lattice ISP Connector**

This is a 10 pin generic 0.100" pitch header connector, providing In System Programming (ISP)

capability for Lattice made programmable logic on board. The pinout of P25 is shown in [Table 7-7](#).

**Table 7-7. P25 - Lattice ISP Connector**

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 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.

**7.1.9 P27 - System Expansion Connector**

P27 is a 128 pin, 90<sup>0</sup>, DIN 41612 connector, which provides a minimal system I/F required to interface various types of communication transceivers. 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 P17 is shown in [Table 7-8](#).

**Table 7-8. P27 - System Expansion Connector**

Pin No.	Signal Name	Attribute	Description
A1	EXPA16	O	Expansion Address (16 <sup>a</sup> :31). This is a Latched-Buffered version of the PQ2'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	EXP12V	O	These can be connected to the positive 12V source from the PCI edge connector thru J3. This line is fused by a 0.5A resettable poly-switch.
A18			
A19	N.C.	-	Not Connected.
A20	EXP3.3V	O	3.3V Power Out. These lines are connected to the main 3.3V plane of the PQ2PCIAI-ADS, this, to provide 3.3V power where necessary for external tool connected.
A21			
A22			
A23			
A24			
A25	N.C.	-	Not Connected.

**Table 7-8. P27 - System Expansion Connector**

Pin No.	Signal Name	Attribute	Description
A26	EXPVCC	O	5V Supply. Connected to ADS's 5V VCC plane. Provided as power supply for external tool.
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 <sup>a</sup> :7). These 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 4-10. "BCSR2 Description" on page 56.
B5	TSTAT1		
B6	TSTAT2		
B7	TSTAT3		
B8	TSTAT4		
B9	TSTAT5		
B10	TSTAT6		
B11	TSTAT7		
B12	TOOLREV0	I	Tool Revision (0 <sup>a</sup> :3). These 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 4-10. "BCSR2 Description" on page 56.
B13	TOOLREV1		
B14	TOOLREV2		
B15	TOOLREV3		
B16	EXTOLI0	I	External Tool Identification (0 <sup>a</sup> :3). These 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 4-10. "BCSR2 Description" on page 56
B17	EXTOLI1		
B18	EXTOLI2		
B19	EXTOLI3		
B20	N.C.	-	Not Connected
B21	EXP3.3V	O	3.3V Power Out. These lines are connected to the main 3.3V plane of the PQ2PCIAI-ADS, this, to provide 3.3V power where necessary for external tool connected.
B22			
B23			
B24			

**Table 7-8. P27 - System Expansion Connector**

Pin No.	Signal Name	Attribute	Description
B25	N.C.	-	Not Connected
B26	EXPVCC	O	5V Supply. Connected to ADS's 5V VCC plane. Provided as power supply for external tool.
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..
C3	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C4	BTOOLCS1#	O	Buffered Tool Chip Select 1 (L). This is a buffered PQ2's CS6# line, reserved for an external tool.
C5	BTOOLCS2#	O	Buffered Tool Chip Select 2 (L). This is a buffered PQ2'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 PQ2. An external tool, using the same pins as does the ATM port should consult this signal before driving the same lines. <b>Failure to do so might result in permanent damage to the PM5350 ATM UNI.</b>
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.	PQ2's Hard Reset (L). When asserted by an external H/W, generates Hard-Reset sequence for the PQ2. During that sequence, asserted by the PQ2 for 512 system clocks. Pulled Up on the ADS using a 1K $\Omega$ resistor. When driven by an external tool, <b>MUST be driven with an Open Drain gate. Failure to do so might result in permanent damage to the PQ2 and / or to ADS logic.</b>
C11	IRQ6#	I	Interrupt Request 6 (L). Connected to PQ2's DP6/CSE0/IRQ6# signal. Pulled up on the ADS with a 10 K $\Omega$ resistor. This line is shared with the ATM UNI's interrupt line and therefore, when driven by an external tool, <b>MUST be driven with an Open Drain gate. Failure to do so may result in permanent damage to the PQ2 or to ADS logic.</b>

Table 7-8. P27 - System Expansion Connector

Pin No.	Signal Name	Attribute	Description
C12	IRQ7#	I	Interrupt Request 7 (L). Connected to PQ2'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, <b>MUST be driven with an Open Drain gate. Failure to do so might result in permanent damage to the PQ2 and / or to ADS logic.</b>
C13	GND	O	Digital Ground. Connected to main GND plane of the ADS.
C14	EXPD0	I/O, T.S.	Expansion Data (0 <sup>a</sup> :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 BTOOLCS1# or BTOOLCS2# are asserted. Otherwise they are tristated. The direction of these lines is determined by buffered BCTL0, in function of W/R#.
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). These 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#		

**Table 7-8. P27 - System Expansion Connector**

Pin No.	Signal Name	Attribute	Description
D6	GND	O	Digital Ground. Connected to main GND plane of the ADS.
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 is the buffered PQ2's ALE, provided for expansion board's use.
D15	EXPCTL0	O	Expansion Control Line 0. This line is a buffered version of PQ2's BCTL0 (Bus Control Line 0) which serves as W/R#, 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			

a.MS Bit.

**7.1.10 P2 - USB Connector**

This is a four pin standard USB connector type-A. The pinout is shown in .

**Table 7-9. P2 - USB Connector**

Pin No.	Signal Name	Description
1	5V Power	Power line of the USB cable
2	D-	Twisted-Pair Transmit Data negative
3	D+	Twisted-Pair Receive Data positive
4	GND	Ground connection

**7.2 Programmable Logic Equations**

There are 4 programmable logic devices on board.

1. U35 - BCSR and PCI Interrupt Controller
2. U41 - Power switch debounce

**7.2.1 U35 - BCSR Code**

MODULE PQ2HipXBCSR

TITLE 'MPC82xx ads control status register'

\*\*\*\*\*

"\* In this file (Prototype) the following changes were made (12/03/02):

"\* - Added support for LBPC in Hard Reset Config Word (determined by external signal nPCI\_Mode)

\*\*\*\*\*

\*\*\*\*\*

"\* In this file (Prototype) the following changes were made (11/01/02):

"\* - Added support for USB, Second Fast Ethernet, PARITY option on 60x, CPM MUX control.

\*\*\*\*\*

\*\*\*\*\*

"\* In this file (Prototype) the following changes were made (07/15/02):

"\* - Added support for a second Fast Ethernet PHY.

"\* - Removed support for fast down-load through JTAG.

\*\*\*\*\*

\*\*\*\*\*

"\* Device declaration. \*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

"\* Pins declaration. \*

\*\*\*\*\*

"\* System i/f pins

\*\*\*\*\*

SYSCLK PIN 124;

IntContCs\_B PIN 48; " PCI INterrupt Controller CS

BrdContRegCs\_B PIN 47; " BCSR CS

DVal\_B PIN 53;

R\_B\_W PIN 46; " BCTL0 signal

BCTL1 PIN ; " Alternate Buffers Enable source

A7 PIN 69; " for flash support

A8 PIN 68; " for flash support

A27 PIN 15;

A28 PIN 12;

A29 PIN 11;

D0 PIN 75;istype 'com' ;

D1 PIN 22;istype 'com' ;

D2 PIN 13;istype 'com' ;

D3 PIN 77istype 'com' ;  
 D4 PIN 16istype 'com' ;  
 D5 PIN 142istype 'com' ;  
 D6 PIN 60istype 'com' ;  
 D7 PIN 87istype 'com' ;  
 D8 PIN 66istype 'com' ;  
 D9 PIN 72istype 'com' ;  
 D10 PIN 70istype 'com' ;  
 D11 PIN 39istype 'com' ;

\*\*\*\*\*

"\* Board Control Pins. Read/Write.

\*\*\*\*\*

L2Inh\_B PIN 130istype 'reg,buffer' ; " flash enable.  
 L2Flush\_B PIN 42istype 'reg,buffer' ; " 60x bus sdram enable  
 L2Lock\_B PIN 112istype 'reg,buffer' ; " bursting sram enable  
 L2Clear\_B PIN 128istype 'reg,buffer' ; " local bus sdram enable

SignalLamp0\_B PIN 44istype 'reg,buffer' ; " status lamp 0 for misc s/w visual  
 SignalLamp1\_B PIN 38istype 'reg,buffer' ; " status lamp 1 for misc s/w visual

AtmEn\_B PIN 134istype 'reg,buffer' ; " atm uni enable  
 AtmDis\_B PIN 114istype 'com' ; " atm uni disable  
 Atm16\_B PIN 28istype 'reg,buffer' ; " UTOPIA 16 bit enable  
 Atm8\_B PIN 116istype 'com' ; " UTOPIA 8 bit enable  
 AtmSinglePHY\_B PIN 4istype 'reg,buffer' ; " UTOPIA Single PHY enable  
 AtmMultiPHY\_B PIN 58istype 'com' ; " UTOPIA Multi PHY enable  
 AtmRst\_B NODE istype 'reg,buffer' ; " atm uni reset bit  
 AtmRstOut\_B PIN 76istype 'com' ; " atm uni reset driven by register  
 " or by HRESET\_B  
 USBEn\_B PIN 5istype 'reg,buffer' ; " USB enable

USBDis\_B PIN 86istype 'com' ; " USB disable  
 USBHiSpd\_B PIN 26istype 'reg,buffer' ; " USB Hi Speed Select  
 USBLowSpd\_B PIN 133istype 'com' ; " USB Low Speed Select  
 USBVccO PIN 7istype 'reg,buffer' ; " USB Line Voltage Select  
  
 FEthEn1\_B PIN 9istype 'reg,buffer' ; " fast ethernet trans. 1 enable  
 FEthDis1\_B PIN 23istype 'com' ; " fast ethernet trans. 1 Disable  
 FEthEn2\_B PIN 40istype 'reg,buffer' ; " fast ethernet trans. 2 enable  
 FEthDis2\_B PIN 79istype 'com' ; " fast ethernet trans. 2 Disable  
 FEthRst1\_B NODE istype 'reg,buffer' ; " fast ethernet trans. 1 reset bit  
 FEthRstOut1\_B PIN 139istype 'com' ; " fast eth trans 1 reset driven by  
 " register or by HRESET\_B  
 FEthRst2\_B NODE istype 'reg,buffer' ; " fast ethernet trans. 2 reset bit  
 FEthRstOut2\_B PIN 110istype 'com' ; " fast eth trans 2 reset driven by  
 " register or by HRESET\_B  
 FEthMDSel1 PIN 10istype 'com' ; " F. Eth. MDIO MDC Mux1  
 FEthMDSel2 PIN 6istype 'com' ; " F. Eth. MDIO MDC Mux2  
  
 RS232En1\_B PIN 32istype 'reg,buffer' ; " RS232 port 1 enable  
 RS232Dis1\_B PIN 56istype 'com' ; " RS232 port 1 Disable  
 RS232En2\_B PIN 3istype 'reg,buffer' ; " RS232 port 2 enable  
 RS232Dis2\_B PIN 8istype 'com' ; " RS232 port 2 Disable  
  
 PCI\_Mode\_B PIN 19 ; " Local Bus PCI Select  
 Local\_Bus\_B PIN 98istype 'com' ; " Local Bus SDRAM Select  
  
 ModckH0 PIN 67 ; " MODCKH0  
 ModckH1 PIN 65 ; " MODCKH1  
 ModckH2 PIN 61 ; " MODCKH2  
 ModckH3 PIN 59 ; " MODCKH3

PCI\_IRQ\_B            PIN   100istype 'com,buffer' ; " PCI Interrupt to PQ2 (o.d.)

PCI\_INTA\_B           PIN   97 ; " PCI Interrupt from PCI card

PCI\_INTB\_B           PIN   126 ; " PCI Interrupt from PCI card

PCI\_INTC\_B           PIN   125 ; " PCI Interrupt from PCI card

PCI\_INTD\_B           PIN   120 ; " PCI Interrupt from PCI card

\*\*\*\*\*

"\* Board Status Registers Chip-Selects

\*\*\*\*\*

Bcsr2Cs\_B            PIN   89istype 'com' ;

Bcsr4Cs\_B            PIN   81istype 'com' ;

\*\*\*\*\*

"\* Flash/EEPROM Associated Pins.

\*\*\*\*\*

F\_PD1                PIN   57 ;

F\_PD2                PIN   55 ;

F\_PD3                PIN   45 ;

F\_PD4                PIN   43 ;

Cs0\_B                PIN   54 ; " flash/eprom chip-select input

Cs4\_B                PIN   94 ; " eprom/flash chip-select input

EEpromCs\_B           PIN   137istype 'com' ; " EEPROM chip-select

FlashCs1\_B           PIN   144istype 'com' ; " Flash bank1 chip-select

FlashCs2\_B           PIN   138istype 'com' ; " Flash bank2 chip-select

FlashCs3\_B           PIN   143istype 'com' ; " Flash bank3 chip-select

FlashCs4\_B           PIN   140istype 'com' ; " Flash bank4 chip-select

\*\*\*\*\*

"\* PM5384 ATM UNI Associated Pins.

\*\*\*\*\*

AtmUniCsIn\_B        PIN 119 ;  
 AtmUniCsOut\_B     PIN 62istype 'com' ; " remove if short of pins

\*\*\*\*\*

"\* Reset & Interrupt Logic Pins.

\*\*\*\*\*

PORIn\_B            PIN 41 ;  
  
 "RstConf\_B        PIN istype 'com'; Hard Reset master select.  
  
 Rst0              PIN 33 ; " connected to N.C. of Reset P.B.  
 Rst1              PIN 31 ; " connected to N.O. of Reset P.B.  
  
 HardReset\_B        PIN 18istype 'com' ; " Actual hard reset output (O.D.)  
 SoftReset\_B        PIN 17istype 'com' ; " Actual soft reset output (O.D.)

Abr0              PIN 30 ; " connected to N.C. of Abort P.B.  
 Abr1              PIN 29 ; " connected to N.O. of Abort P.B.

NMIEn        NODE istype 'com' ; " enables T.S. NMI pin  
 NMI\_B        PIN 20istype 'com' ; " Actual NMI pin (O,O.D.)

\*\*\*\*\*

"\* Data Buffers Enables and Reset configuration support

\*\*\*\*\*

TEA\_B                    PIN 102 ; " Transfer Error Acknowledge.

DataBufEn\_B            PIN 85istype 'com,invert' ; " data buffer enable

ToolCs1\_B              PIN 27 ; " comm tool cs line 1.

ToolCs2\_B              PIN 21 ; " comm tool cs line 2.

ToolDataBufEn\_B        PIN 91istype 'com,invert' ; " tool data buffer enable

\*\*\*\*\*

**"\* Hard Reset Configuration Logic**

\*\*\*\*\*

boot\_device\_B PIN 118 ; " selects EEPROM/FLASH\_B as boot device

bcsrConfEn PIN 93 ; " selects Hard Reset Configuration Source

" as BCSR or EEPROM/FLASH.

\*\*\*\*\*

**"\* Auxiliary Pins.**

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

**"\* System Hard Reset Configuration.**

\*\*\*\*\*

DataOeNODE istype 'com' ; " data bus output enable on read.

DataPCIOeNODE istype 'com' ; " data bus output enable on PCI read.

\*\*\*\*\*

**"\* Control Register Enable Protection.**

\*\*\*\*\*

\*\*\*\*\*

"\* Reset & Interrupt Logic Pins.

\*\*\*\*\*

RstDeb1NODE istype 'keep,com' ; " reset push button debouncer

AbrDeb1NODE 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

HoldOffCnt2,

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.

\*\*\*\*\*

"\* PCI Interrupt Register.

\*\*\*\*\*

Slot0IntANODE istype 'reg,invert' ; " PCI Slot 0 Interrupt A

Slot0IntBNODE istype 'reg,invert' ; " PCI Slot 0 Interrupt B

Slot0IntCNODE istype 'reg,invert' ; " PCI Slot 0 Interrupt C

Slot0IntDNODE istype 'reg,invert' ; " PCI Slot 0 Interrupt D  
 Slot1IntANODE istype 'reg,invert' ; " PCI Slot 1 Interrupt A  
 Slot1IntBNODE istype 'reg,invert' ; " PCI Slot 1 Interrupt B  
 Slot1IntCNODE istype 'reg,invert' ; " PCI Slot 1 Interrupt C  
 Slot1IntDNODE istype 'reg,invert' ; " PCI Slot 1 Interrupt D  
 Slot2IntANODE istype 'reg,invert' ; " PCI Slot 2 Interrupt A  
 Slot2IntBNODE istype 'reg,invert' ; " PCI Slot 2 Interrupt B  
 Slot2IntCNODE istype 'reg,invert' ; " PCI Slot 2 Interrupt C  
 Slot2IntDNODE istype 'reg,invert' ; " PCI Slot 2 Interrupt D

\*\*\*\*\*

"\* PCI Interrupt Mask Register.

\*\*\*\*\*

Slot0IntAMaskNODE istype 'reg,buffer' ; " PCI Slot 0 Interrupt A Mask  
 Slot0IntBMaskNODE istype 'reg,buffer' ; " PCI Slot 0 Interrupt B Mask  
 Slot0IntCMaskNODE istype 'reg,buffer' ; " PCI Slot 0 Interrupt C Mask  
 Slot0IntDMaskNODE istype 'reg,buffer' ; " PCI Slot 0 Interrupt D Mask  
 Slot1IntAMaskNODE istype 'reg,buffer' ; " PCI Slot 1 Interrupt A Mask  
 Slot1IntBMaskNODE istype 'reg,buffer' ; " PCI Slot 1 Interrupt B Mask  
 Slot1IntCMaskNODE istype 'reg,buffer' ; " PCI Slot 1 Interrupt C Mask  
 Slot1IntDMaskNODE istype 'reg,buffer' ; " PCI Slot 1 Interrupt D Mask  
 Slot2IntAMaskNODE istype 'reg,buffer' ; " PCI Slot 2 Interrupt A Mask  
 Slot2IntBMaskNODE istype 'reg,buffer' ; " PCI Slot 2 Interrupt B Mask  
 Slot2IntCMaskNODE istype 'reg,buffer' ; " PCI Slot 2 Interrupt C Mask  
 Slot2IntDMaskNODE istype 'reg,buffer' ; " PCI Slot 2 Interrupt D Mask

\*\*\*\*\*

"\* PCI Interrupt Request to PQ2.

\*\*\*\*\*

PCI\_InterruptNODE istype 'com' ; " generated Interrupt to PQ2

\*\*\*\*\*

"\* Misceleneous.

\*\*\*\*\*

KeepPinsConnected NODE istype 'com' ;

\*\*\*\*\*

\*\*\*\*\*

H, L, X, Z = 1, 0, .X., .Z. ;

C, D, U = .C., .D., .U. ;

\*\*\*\*\*

"\* SIMULATION = 1 ;

\*\*\*\*\*

"\* Signal groups

\*\*\*\*\*

Add = [A27..A29] ;

Data = [D0..D7] ;

DataPCI = [D0..D11] ;

ContReg = [L2Inh\_B,

L2Flush\_B,

L2Lock\_B,

L2Clear\_B,

Signalamp0\_B,

Signalamp1\_B,

AtmEn\_B,

AtmRst\_B,

Atm16\_B,

AtmSinglePHY\_B,

FEthEn1\_B,

FEthRst1\_B,

FEthEn2\_B,

FEthRst2\_B,

RS232En1\_B,

RS232En2\_B,

USBEn\_B,

USBHiSpd\_B,

USBVccO] ;

ReadBcsr0 = [0,

0,

L2Inh\_B,

L2Flush\_B,

L2Lock\_B,

L2Clear\_B,

Signalamp0\_B,

Signalamp1\_B] ;

ReadBcsr1 = [bcsrConfEn,

boot\_device\_B,

AtmEn\_B,

AtmRst\_B.fb,

FEthEn1\_B,

FEthRst1\_B.fb,

RS232En1\_B,

RS232En2\_B] ;

```
ReadBcsr3 = [USBE_n_B,  
             USBHiSpd_B,  
             USBVccO,  
             FEthEn2_B,  
             FEthRst2_B.fb,  
             Atm16_B,  
             AtmSinglePHY_B,  
             PCI_Mode_B];
```

```
DrivenContReg = [L2Inh_B,  
                 L2Flush_B,  
                 L2Lock_B,  
                 L2Clear_B,  
                 Signalamp0_B,  
                 Signalamp1_B,  
                 AtmEn_B,  
                 Atm16_B,  
                 AtmSinglePHY_B,  
                 FEthEn1_B,  
                 FEthEn2_B,  
                 RS232En1_B,  
                 RS232En2_B,  
                 USBEn_B,  
                 USBHiSpd_B,  
                 USBVccO];
```

```
ClockedContReg = [L2Inh_B,  
                  L2Flush_B,  
                  L2Lock_B,  
                  L2Clear_B,
```

Signalamp0\_B,  
Signalamp1\_B,  
AtmEn\_B,  
AtmRst\_B,  
Atm16\_B,  
AtmSinglePHY\_B,  
FEthEn1\_B,  
FEthEn2\_B,  
FEthRst1\_B,  
FEthRst2\_B,  
RS232En1\_B,  
RS232En2\_B,  
USBEn\_B,  
USBHiSpd\_B,  
USBVccO] ;

IntReg = [Slot0IntA,  
Slot0IntB,  
Slot0IntC,  
Slot0IntD,  
Slot1IntA,  
Slot1IntB,  
Slot1IntC,  
Slot1IntD,  
Slot2IntA,  
Slot2IntB,  
Slot2IntC,  
Slot2IntD] ;

IntMaskReg = [Slot0IntAMask,  
Slot0IntBMask,

```

Slot0IntCMask,
Slot0IntDMask,
Slot1IntAMask,
Slot1IntBMask,
Slot1IntCMask,
Slot1IntDMask,
Slot2IntAMask,
Slot2IntBMask,
Slot2IntCMask,
Slot2IntDMask];

```

```

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,FEthRstOut1_B,FEthRstOut2_B];
Rst = [Rst1,Rst0];
Abr = [Abr1,Abr0];
Debounce = [RstDeb1,AbrDeb1];
SyncReset = [SyncHardReset_B,DSyncHardReset_B];
RstCause = [PORIn_B,Rst1,Rst0,Abr1,Abr0];
HoldOffCnt = [HoldOffCnt2,HoldOffCnt1,HoldOffCnt0];
F_PD = [F_PD4, F_PD3, F_PD2, F_PD1];
Cs =
[Cs0_B,Cs4_B,BrdContRegCs_B,IntContCs_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 = [0,0,0,0,0,1,1,0];
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 = [0,0,0,0,0,1,1,0];
CfgByte3 = [0,0,0,0,ModckH0,ModckH1,ModckH2,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 ;

```

```

BCSR4_ADD = 4 ;

```

```

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_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) ;

```

```

*****
*****
"* BCSR 0 definitions.
*****
*****

```

```

L2CACHE_INHIBITED = 0 ;
L2CACHE_FLUSHED = 0 ;
L2CACHE_LOCKED = 0 ;
L2CACHE_CLEARED = 0 ;
SIGNAL_LAMP_ON = 0 ;

```

```

*****
***** Power On Defaults Assignments *****
*****
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 \*\*\*\*\*

\*\*\*\*\*

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.

\*\*\*\*\*

\*\*\*\*\*

BCSR\_BOOT = 0 ;" bcsrConfEn = 0 Hard Reset Conf Word from BCSR
MEMORY\_BOOT = 1 ;" bcsrConfEn = 1 Hard Reset Conf from EEPROM/FLASH
FLASH\_BOOT = 0 ;" boot\_device\_B = 0
EEPROM\_BOOT = 1 ;" boot\_device\_B = 1
ATM\_ENABLED = 0 ;
ATM\_RESET\_ACTIVE = 0 ;
FETH1\_ENABLED = 0 ;
FETH1\_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 ;
FETH1_ENABLE_PON_DEFAULT = !FETH1_ENABLED ;
FETH1_RESET_PON_DEFAULT = !FETH1_RESET_ACTIVE ;
RS232_1_ENABLE_PON_DEFAULT = !RS232_1_ENABLE ;
RS232_2_ENABLE_PON_DEFAULT = !RS232_2_ENABLE ;
    
```

\*\*\*\*\*

\*\*\*\*\* Data Bits Assignments \*\*\*\*\*

\*\*\*\*\*

```

CONF_WORD_DATA_BIT = [D0] ;
BOOT_DEVICE_DATA_BIT = [D1] ;
ATM_ENABLE_DATA_BIT = [D2] ;
ATM_RESET_DATA_BIT = [D3] ;
FETH1_ENABLE_DATA_BIT = [D4] ;
FETH1_RESET_DATA_BIT = [D5] ;
RS232_1_ENABLE_DATA_BIT = [D6] ;
RS232_2_ENABLE_DATA_BIT = [D7] ;
    
```

\*\*\*\*\*

\*\*\*\*\*

/\* BCSR 3 definitions.

\*\*\*\*\*

\*\*\*\*\*

```

USB_ENABLED = 0 ;
USB_SPEED_HIGH = 0 ;
USB_VCCO_ON = 1 ;
FETH2_ENABLED = 0 ;
FETH2_RESET_ACTIVE = 0 ;
    
```

```
ATM16_ENABLED = 0 ;
ATM_SINGLE_PHY_ENABLED = 0 ;
```

```
*****
```

```
***** Power On Defaults Assignments *****
```

```
*****
```

```
USB_ENABLE_PON_DEFAULT = !USB_ENABLED ;
USB_SPEED_PON_DEFAULT = USB_SPEED_HIGH ;
USB_VCCO_PON_DEFAULT = !USB_VCCO_ON ;
FETH2_ENABLE_PON_DEFAULT = !FETH2_ENABLED ;
FETH2_RESET_PON_DEFAULT = !FETH2_RESET_ACTIVE ;
ATM16_ENABLE_PON_DEFAULT = !ATM16_ENABLED ;
ATM_SINGLE_PHY_ENABLE_PON_DEFAULT = ATM_SINGLE_PHY_ENABLED ;
```

```
*****
```

```
***** Data Bits Assignments *****
```

```
*****
```

```
USB_ENABLE_DATA_BIT = [D0] ;
USB_SPEED_DATA_BIT = [D1] ;
USB_VCCO_DATA_BIT = [D2] ;
FETH2_ENABLE_DATA_BIT = [D3] ;
FETH2_RESET_DATA_BIT = [D4] ;
ATM16_ENABLE_DATA_BIT = [D5] ;
ATM_SINGLE_PHY_ENABLE_DATA_BIT = [D6] ;
LOCAL_BUS_DATA_BIT = [D7] ;
```

```
*****
```

```
*** PCI Interrupt Register Access definitions
```

```
*****
```

```
IntReg_ADD = 0 ;
IntMaskReg_ADD = 1 ;
```

```
"VGR_WRITE_IntReg    = (!IntContCs_B & !DVal_B & R_B_W & !A27 & !A28 & !A29) ;
VGR_WRITE_IntMaskReg = (!IntContCs_B & !DVal_B & R_B_W & !A27 & !A28 & A29) ;
```

```
VGR_READ_IntReg     = (!IntContCs_B & !R_B_W & !A27 & !A28 & !A29) ;
VGR_READ_IntMaskReg = (!IntContCs_B & !R_B_W & !A27 & !A28 & A29) ;
```

```
*****
```

```
"* Interrupt Request Definitions.
```

```
*****
```

```
"IrqOe = (Slot0IntA #
" Slot0IntB #
" Slot0IntC #
" Slot0IntD #
" Slot1IntA #
" Slot1IntB #
" Slot1IntC #
" Slot1IntD #
" Slot2IntA #
" Slot2IntB #
" Slot2IntC #
" Slot2IntD) ;
```

```
*****
```

```
*****
```

```
"* PCI Interrupt Register definitions.
```

```
*****
```

```
*****
```

```
Slot0IntA_Active = 1 ; " PCI Slot 0 Interrupt A asserted
```

```
Slot0IntB_Active = 1 ; " PCI Slot 0 Interrupt B asserted
```

---

Slot0IntC\_Active = 1 ; " PCI Slot 0 Interrupt C asserted  
Slot0IntD\_Active = 1 ; " PCI Slot 0 Interrupt D asserted  
Slot1IntA\_Active = 1 ; " PCI Slot 1 Interrupt A asserted  
Slot1IntB\_Active = 1 ; " PCI Slot 1 Interrupt B asserted  
Slot1IntC\_Active = 1 ; " PCI Slot 1 Interrupt C asserted  
Slot1IntD\_Active = 1 ; " PCI Slot 1 Interrupt D asserted  
Slot2IntA\_Active = 1 ; " PCI Slot 2 Interrupt A asserted  
Slot2IntB\_Active = 1 ; " PCI Slot 2 Interrupt B asserted  
Slot2IntC\_Active = 1 ; " PCI Slot 2 Interrupt C asserted  
Slot2IntD\_Active = 1 ; " PCI Slot 2 Interrupt D asserted

\*\*\*\*\*  
\*\*\*\*\* Power On Defaults Assignments \*\*\*\*\*  
\*\*\*\*\*

Slot0IntA\_PON\_DEFAULT = !Slot0IntA\_Active ;  
Slot0IntB\_PON\_DEFAULT = !Slot0IntB\_Active ;  
Slot0IntC\_PON\_DEFAULT = !Slot0IntC\_Active ;  
Slot0IntD\_PON\_DEFAULT = !Slot0IntD\_Active ;  
Slot1IntA\_PON\_DEFAULT = !Slot1IntA\_Active ;  
Slot1IntB\_PON\_DEFAULT = !Slot1IntB\_Active ;  
Slot1IntC\_PON\_DEFAULT = !Slot1IntC\_Active ;  
Slot1IntD\_PON\_DEFAULT = !Slot1IntD\_Active ;  
Slot2IntA\_PON\_DEFAULT = !Slot2IntA\_Active ;  
Slot2IntB\_PON\_DEFAULT = !Slot2IntB\_Active ;  
Slot2IntC\_PON\_DEFAULT = !Slot2IntC\_Active ;  
Slot2IntD\_PON\_DEFAULT = !Slot2IntD\_Active ;

\*\*\*\*\*  
\*\*\*\*\* Data Bits Assignments \*\*\*\*\*  
\*\*\*\*\*

Slot0IntA\_DATA\_BIT = [D0] ;

```
Slot0IntB_DATA_BIT = [D1] ;
Slot0IntC_DATA_BIT = [D2] ;
Slot0IntD_DATA_BIT = [D3] ;
Slot1IntA_DATA_BIT = [D4] ;
Slot1IntB_DATA_BIT = [D5] ;
Slot1IntC_DATA_BIT = [D6] ;
Slot1IntD_DATA_BIT = [D7] ;
Slot2IntA_DATA_BIT = [D8] ;
Slot2IntB_DATA_BIT = [D9] ;
Slot2IntC_DATA_BIT = [D10] ;
Slot2IntD_DATA_BIT = [D11] ;
```

```
*****
```

```
*****
```

```
/* PCI Interrupt Mask Register definitions.
```

```
*****
```

```
*****
```

```
Slot0IntAMask_Active = 1 ; " PCI Slot 0 Interrupt A Masked
Slot0IntBMask_Active = 1 ; " PCI Slot 0 Interrupt B Masked
Slot0IntCMask_Active = 1 ; " PCI Slot 0 Interrupt C Masked
Slot0IntDMask_Active = 1 ; " PCI Slot 0 Interrupt D Masked
Slot1IntAMask_Active = 1 ; " PCI Slot 1 Interrupt A Masked
Slot1IntBMask_Active = 1 ; " PCI Slot 1 Interrupt B Masked
Slot1IntCMask_Active = 1 ; " PCI Slot 1 Interrupt C Masked
Slot1IntDMask_Active = 1 ; " PCI Slot 1 Interrupt D Masked
Slot2IntAMask_Active = 1 ; " PCI Slot 2 Interrupt A Masked
Slot2IntBMask_Active = 1 ; " PCI Slot 2 Interrupt B Masked
Slot2IntCMask_Active = 1 ; " PCI Slot 2 Interrupt C Masked
Slot2IntDMask_Active = 1 ; " PCI Slot 2 Interrupt D Masked
```

```
*****
```

\*\*\*\*\* Power On Defaults Assignments \*\*\*\*\*

\*\*\*\*\*

Slot0IntAMask\_PON\_DEFAULT = Slot0IntAMask\_Active ;  
Slot0IntBMask\_PON\_DEFAULT = Slot0IntBMask\_Active ;  
Slot0IntCMask\_PON\_DEFAULT = Slot0IntCMask\_Active ;  
Slot0IntDMask\_PON\_DEFAULT = Slot0IntDMask\_Active ;  
Slot1IntAMask\_PON\_DEFAULT = Slot1IntAMask\_Active ;  
Slot1IntBMask\_PON\_DEFAULT = Slot1IntBMask\_Active ;  
Slot1IntCMask\_PON\_DEFAULT = Slot1IntCMask\_Active ;  
Slot1IntDMask\_PON\_DEFAULT = Slot1IntDMask\_Active ;  
Slot2IntAMask\_PON\_DEFAULT = Slot2IntAMask\_Active ;  
Slot2IntBMask\_PON\_DEFAULT = Slot2IntBMask\_Active ;  
Slot2IntCMask\_PON\_DEFAULT = Slot2IntCMask\_Active ;  
Slot2IntDMask\_PON\_DEFAULT = Slot2IntDMask\_Active ;

\*\*\*\*\*

\*\*\*\*\* Data Bits Assignments \*\*\*\*\*

\*\*\*\*\*

Slot0IntAMask\_DATA\_BIT = [D0];  
Slot0IntBMask\_DATA\_BIT = [D1];  
Slot0IntCMask\_DATA\_BIT = [D2];  
Slot0IntDMask\_DATA\_BIT = [D3];  
Slot1IntAMask\_DATA\_BIT = [D4];  
Slot1IntBMask\_DATA\_BIT = [D5];  
Slot1IntCMask\_DATA\_BIT = [D6];  
Slot1IntDMask\_DATA\_BIT = [D7];  
Slot2IntAMask\_DATA\_BIT = [D8];  
Slot2IntBMask\_DATA\_BIT = [D9];  
Slot2IntCMask\_DATA\_BIT = [D10];  
Slot2IntDMask\_DATA\_BIT = [D11];

\*\*\*\*\*

"\* Flash Declarations.

\*\*\*\*\*

FLASH\_ENABLE\_ACTIVE = 0 ;

" the presence detect encoding for the below is fictional

" needs to be updated with real data.

CP29020 = (F\_PD == 8) ; " 1 X 2 MByte bank

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 = ( CP29020 #  
                   SM73228XU1 #  
                   (SM73248XU2 & !A8) #  
                   (SM73288XU4 & !A7 & !A8) ) ;

FLASH\_BANK2 = ( (SM73248XU2 & A8) #  
                   (SM73288XU4 & !A7 & A8) ) ;

FLASH\_BANK3 = ( A7 & !A8 & SM73288XU4 ) ;

FLASH\_BANK4 = ( A7 & A8 & SM73288XU4 ) ;

\*\*\*\*\*

"\* ATM UNI Declarations.

\*\*\*\*\*

\*\*\*\*\*

"\* Reset Declarations.

\*\*\*\*\*

HARD\_RESET\_ACTIVE = 0 ;

SOFT\_RESET\_ACTIVE = 0 ;

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/eeprom

END\_OF\_FLASH\_EEPROM\_READ = !DVal\_B & (!Cs0\_B # !Cs4\_B) & !R\_B\_W &  
DSyncHardReset\_B.fb ;

" end of flash/eeprom read cycle.

" not during hard reset config

END\_OF\_PCI\_INT\_CONT\_READ = !DVal\_B & !IntContCs\_B & !R\_B\_W ;

" end of PCI Interrupt Controller read cycle.

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 & Cs0\_B & Cs4\_B & AtmUniCsIn\_B &  
IntContCs\_B #

!DVal\_B & !AtmUniCsIn\_B & R\_B\_W #

!DVal\_B & !ToolCs1\_B & R\_B\_W #

```

!DVal_B & !ToolCs2_B & R_B_W #
!DVal_B & (!Cs0_B # !Cs4_B) & R_B_W #
!DVal_B & !IntContCs_B & R_B_W);
" another access or atm uni write or tool 1 write or tool 2 write or
" flash/EEPROM write PCI int cont write
*****
"* Hard Reset Configuration Logic
*****

HRESET_CFG_IN_BCSR = (bcsrConfEn == 1); " HRESET Conf Word in BCSR
HRESET_BOOT_IN_FLASH = ((bcsrConfEn == 0) & (boot_device_B == 0));
" HRESET Conf Word and Boot Code in FLASH
BOOT_IN_FLASH = ((bcsrConfEn == 1) & (boot_device_B == 0));
" HRESET Conf Word in BCSR and Boot Code in FLASH
HRESET_BOOT_IN_EEPROM = ((bcsrConfEn == 0) & (boot_device_B == 1));
" HRESET Conf Word and Boot Code in EEPROM
BOOT_IN_EEPROM = ((bcsrConfEn == 1) & (boot_device_B == 1));
" HRESET Conf Word in BCSR and Boot Code in EEPROM
HARD_RESET_ASSERTION = ( (HardReset_B == 0) & (SyncHardReset_B.fb == 0) &
    (DSyncHardReset_B.fb == 1) );

CS0_ASSERTED = (Cs0_B == 0);
CS4_ASSERTED = (Cs4_B == 0);

FIRST_CFG_BYTE_READ = (CS0_ASSERTED & !DSyncHardReset_B.fb & (ConfAdd
== 0) &
HRESET_CFG_IN_BCSR & !R_B_W);
SCND_CFG_BYTE_READ = (CS0_ASSERTED & !DSyncHardReset_B.fb & (ConfAdd
== 1) &
HRESET_CFG_IN_BCSR & !R_B_W);
THIRD_CFG_BYTE_READ = (CS0_ASSERTED & !DSyncHardReset_B.fb & (ConfAdd
== 2) &
HRESET_CFG_IN_BCSR & !R_B_W);

```

```
FORTH_CFG_BYTE_READ = (CS0_ASSERTED & !DSyncHardReset_B.fb & (ConfAdd
== 3) &
HRESET_CFG_IN_BCSR & !R_B_W);
```

```
*****
"* Equations, state diagrams.
*****
*****
*****
```

equations

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

```
*****
*****
"* BCSR 0
*****
*****
```

equations

```
*****
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 FEthEn1_B
state FETH1_ENABLED:
if (VGR_WRITE_BCSR_1 &
(FETH1_ENABLE_DATA_BIT.pin == !FETH1_ENABLED) &
(!PON_RESET # (FETH1_ENABLE_PON_DEFAULT != FETH1_ENABLED)) #
(PON_RESET & (FETH1_ENABLE_PON_DEFAULT == !FETH1_ENABLED)) ) then
!FETH1_ENABLED
else

```

```
FETH1_ENABLED ;
state !FETH1_ENABLED:
if (VGR_WRITE_BCSR_1 &
(FETH1_ENABLE_DATA_BIT.pin == FETH1_ENABLED) &
(!PON_RESET # (FETH1_ENABLE_PON_DEFAULT != !FETH1_ENABLED)) #
(PON_RESET & (FETH1_ENABLE_PON_DEFAULT == FETH1_ENABLED)) ) then
FETH1_ENABLED
else
!FETH1_ENABLED ;
*****

state_diagram FEthRst1_B
state FETH1_RESET_ACTIVE:
if (VGR_WRITE_BCSR_1 &
(FETH1_RESET_DATA_BIT.pin == !FETH1_RESET_ACTIVE) &
(!PON_RESET # (FETH1_RESET_PON_DEFAULT != FETH1_RESET_ACTIVE)) #
(PON_RESET & (FETH1_RESET_PON_DEFAULT == !FETH1_RESET_ACTIVE)) )
then
!FETH1_RESET_ACTIVE
else
FETH1_RESET_ACTIVE ;
state !FETH1_RESET_ACTIVE:
if (VGR_WRITE_BCSR_1 &
(FETH1_RESET_DATA_BIT.pin == FETH1_RESET_ACTIVE) &
(!PON_RESET # (FETH1_RESET_PON_DEFAULT != !FETH1_RESET_ACTIVE)) #
(PON_RESET & (FETH1_RESET_PON_DEFAULT == FETH1_RESET_ACTIVE)) )
then
FETH1_RESET_ACTIVE
else
!FETH1_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 & (Atm16_B & AtmMultiPHY_B) &
(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 ;

\*\*\*\*\*

\*\*\*\*\*

"\* BCSR3 State Machines

\*\*\*\*\*

\*\*\*\*\*

state\_diagram USBEn\_B

state USB\_ENABLED:

if (VGR\_WRITE\_BCSR\_3 &

(USB\_ENABLE\_DATA\_BIT.pin == !USB\_ENABLED) &

(!PON\_RESET # (USB\_ENABLE\_PON\_DEFAULT != USB\_ENABLED)) #

(PON\_RESET & (USB\_ENABLE\_PON\_DEFAULT == !USB\_ENABLED)) ) then

!USB\_ENABLED

else

USB\_ENABLED ;

state !USB\_ENABLED:

if (VGR\_WRITE\_BCSR\_3 & Atm16\_B &

(USB\_ENABLE\_DATA\_BIT.pin == USB\_ENABLED) &

(!PON\_RESET # (USB\_ENABLE\_PON\_DEFAULT != !USB\_ENABLED)) #

(PON\_RESET & (USB\_ENABLE\_PON\_DEFAULT == USB\_ENABLED)) ) then

USB\_ENABLED

else

!USB\_ENABLED ;

\*\*\*\*\*

state\_diagram USBHiSpd\_B

state USB\_SPEED\_HIGH:

if (VGR\_WRITE\_BCSR\_3 &

(USB\_SPEED\_DATA\_BIT.pin == !USB\_SPEED\_HIGH) &

(!PON\_RESET # (USB\_SPEED\_PON\_DEFAULT != USB\_SPEED\_HIGH)) #

```

(PON_RESET & (USB_SPEED_PON_DEFAULT == !USB_SPEED_HIGH)) ) then
!USB_SPEED_HIGH
else
USB_SPEED_HIGH ;
state !USB_SPEED_HIGH:
if (VGR_WRITE_BCSR_3 &
(USB_SPEED_DATA_BIT.pin == USB_SPEED_HIGH) &
(!PON_RESET # (USB_SPEED_PON_DEFAULT != !USB_SPEED_HIGH)) #
(PON_RESET & (USB_SPEED_PON_DEFAULT == USB_SPEED_HIGH)) ) then
USB_SPEED_HIGH
else
!USB_SPEED_HIGH ;
*****

state_diagram USBVccO
state USB_VCCO_ON:
if (VGR_WRITE_BCSR_3 &
(USB_VCCO_DATA_BIT.pin == !USB_VCCO_ON) &
(!PON_RESET # (USB_VCCO_PON_DEFAULT != USB_VCCO_ON)) #
(PON_RESET & (USB_VCCO_PON_DEFAULT == !USB_VCCO_ON)) ) then
!USB_VCCO_ON
else
USB_VCCO_ON ;
state !USB_VCCO_ON:
if (VGR_WRITE_BCSR_3 &
(USB_VCCO_DATA_BIT.pin == USB_VCCO_ON) &
(!PON_RESET # (USB_VCCO_PON_DEFAULT != !USB_VCCO_ON)) #
(PON_RESET & (USB_VCCO_PON_DEFAULT == USB_VCCO_ON)) ) then
USB_VCCO_ON
else
!USB_VCCO_ON ;
*****

```

```

state_diagram FEthEn2_B
state FETH2_ENABLED:
if (VGR_WRITE_BCSR_3 &
(FETH2_ENABLE_DATA_BIT.pin == !FETH2_ENABLED) &
(!PON_RESET # (FETH2_ENABLE_PON_DEFAULT != FETH2_ENABLED)) #
(PON_RESET & (FETH2_ENABLE_PON_DEFAULT == !FETH2_ENABLED)) ) then
!FETH2_ENABLED
else
FETH2_ENABLED ;
state !FETH2_ENABLED:
if (VGR_WRITE_BCSR_3 &
(FETH2_ENABLE_DATA_BIT.pin == FETH2_ENABLED) &
(!PON_RESET # (FETH2_ENABLE_PON_DEFAULT != !FETH2_ENABLED)) #
(PON_RESET & (FETH2_ENABLE_PON_DEFAULT == FETH2_ENABLED)) ) then
FETH2_ENABLED
else
!FETH2_ENABLED ;
*****

state_diagram FEthRst2_B
state FETH2_RESET_ACTIVE:
if (VGR_WRITE_BCSR_3 &
(FETH2_RESET_DATA_BIT.pin == !FETH2_RESET_ACTIVE) &
(!PON_RESET # (FETH2_RESET_PON_DEFAULT != FETH2_RESET_ACTIVE)) #
(PON_RESET & (FETH2_RESET_PON_DEFAULT == !FETH2_RESET_ACTIVE)) )
then
!FETH2_RESET_ACTIVE
else
FETH2_RESET_ACTIVE ;
state !FETH2_RESET_ACTIVE:
if (VGR_WRITE_BCSR_3 &
(FETH2_RESET_DATA_BIT.pin == FETH2_RESET_ACTIVE) &

```

```

(!PON_RESET # (FETH2_RESET_PON_DEFAULT != !FETH2_RESET_ACTIVE)) #
(PON_RESET & (FETH2_RESET_PON_DEFAULT == FETH2_RESET_ACTIVE)) )
then
FETH2_RESET_ACTIVE
else
!FETH2_RESET_ACTIVE ;
*****

state_diagram Atm16_B
state ATM16_ENABLED:
if (VGR_WRITE_BCSR_3 &
(ATM16_ENABLE_DATA_BIT.pin == !ATM16_ENABLED) &
(!PON_RESET # (ATM16_ENABLE_PON_DEFAULT != ATM16_ENABLED)) #
(PON_RESET & (ATM16_ENABLE_PON_DEFAULT == !ATM16_ENABLED)) ) then
!ATM16_ENABLED
else
ATM16_ENABLED ;
state !ATM16_ENABLED:
if (VGR_WRITE_BCSR_3 & (USBEn_B & RS232En2_B) &
(ATM16_ENABLE_DATA_BIT.pin == ATM16_ENABLED) &
(!PON_RESET # (ATM16_ENABLE_PON_DEFAULT != !ATM16_ENABLED)) #
(PON_RESET & (ATM16_ENABLE_PON_DEFAULT == ATM16_ENABLED)) ) then
ATM16_ENABLED
else
!ATM16_ENABLED ;
*****

state_diagram AtmSinglePHY_B
state ATM_SINGLE_PHY_ENABLED:
if (VGR_WRITE_BCSR_3 & RS232En2_B &
(ATM_SINGLE_PHY_ENABLE_DATA_BIT.pin == !ATM_SINGLE_PHY_ENABLED) &
(!PON_RESET
# (ATM_SINGLE_PHY_ENABLE_PON_DEFAULT != ATM_SINGLE_PHY_ENABLED))

```

```
#
(PON_RESET
 & (ATM_SINGLE_PHY_ENABLE_PON_DEFAULT ==
!ATM_SINGLE_PHY_ENABLED)) ) then
!ATM_SINGLE_PHY_ENABLED
else
ATM_SINGLE_PHY_ENABLED ;
state !ATM_SINGLE_PHY_ENABLED:
if (VGR_WRITE_BCSR_3 &
(ATM_SINGLE_PHY_ENABLE_DATA_BIT.pin == ATM_SINGLE_PHY_ENABLED) &
(!PON_RESET
 # (ATM_SINGLE_PHY_ENABLE_PON_DEFAULT !=
!ATM_SINGLE_PHY_ENABLED)) #
(PON_RESET
 & (ATM_SINGLE_PHY_ENABLE_PON_DEFAULT ==
ATM_SINGLE_PHY_ENABLED)) ) then
ATM_SINGLE_PHY_ENABLED
else
!ATM_SINGLE_PHY_ENABLED ;
```

\*\*\*\*\*

```
equations
AtmDis_B = !AtmEn_B ;
Atm8_B = !Atm16_B ;
AtmMultiPHY_B = !AtmSinglePHY_B ;
```

```
USBDIs_B = !USBEn_B ;
USBLowSpd_B = !USBHiSpd_B ;
```

```
FEthDis1_B = !FEthEn1_B ;
FEthDis2_B = !FEthEn2_B ;
```

FEthMDSel1 = (!FEthEn1\_B # !FEthEn2\_B) & (!Atm16\_B # !USBEn\_B) ;

FEthMDSel2 = !FEthEn1\_B # !FEthEn2\_B # !Atm16\_B # !USBEn\_B ;

RS232Dis1\_B = !RS232En1\_B ;

RS232Dis2\_B = !RS232En2\_B ;

Local\_Bus\_B = !PCI\_Mode\_B ;

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

"\* PCI Interrupt Register

\*\*\*\*\*

\*\*\*\*\*

equations

IntReg.clk = SYSCLK ;

IntReg.ar = 0 ;

IntReg.ap = 0 ;

\*\*\*\*\*

state\_diagram Slot0IntA

state Slot0IntA\_Active:

if ( ((HardReset\_B == 0) & (Slot0IntA\_PON\_DEFAULT == !Slot0IntA\_Active)) #  
 (!(HardReset\_B == 0) & ((PCI\_INTA\_B & !Slot0IntAMask.fb) #  
 Slot0IntAMask.fb)) )

then

!Slot0IntA\_Active

else

Slot0IntA\_Active ;

```
state !Slot0IntA_Active:
if ( ((HardReset_B == 0) & (Slot0IntA_PON_DEFAULT == Slot0IntA_Active)) #
    (!(HardReset_B == 0) & !PCI_INTA_B & !Slot0IntAMask.fb) )
then
Slot0IntA_Active
else
!Slot0IntA_Active ;
```

\*\*\*\*\*

```
state_diagram Slot0IntB
state Slot0IntB_Active:
if ( ((HardReset_B == 0) & (Slot0IntB_PON_DEFAULT == !Slot0IntB_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTB_B & !Slot0IntBMask.fb) #
    Slot0IntBMask.fb)) )
then
!Slot0IntB_Active
else
Slot0IntB_Active ;
state !Slot0IntB_Active:
if ( ((HardReset_B == 0) & (Slot0IntB_PON_DEFAULT == Slot0IntB_Active)) #
    (!(HardReset_B == 0) & !PCI_INTB_B & !Slot0IntBMask.fb) )
then
Slot0IntB_Active
else
!Slot0IntB_Active ;
```

\*\*\*\*\*

```
state_diagram Slot0IntC
state Slot0IntC_Active:
if ( ((HardReset_B == 0) & (Slot0IntC_PON_DEFAULT == !Slot0IntC_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTC_B & !Slot0IntCMask.fb) #
    Slot0IntCMask.fb)) )
then
```

```

!Slot0IntC_Active
else
Slot0IntC_Active ;
state !Slot0IntC_Active:
if ( ((HardReset_B == 0) & (Slot0IntC_PON_DEFAULT == Slot0IntC_Active)) #
    (!(HardReset_B == 0) & !PCI_INTC_B & !Slot0IntCMask.fb) )
then
Slot0IntC_Active
else
!Slot0IntC_Active ;
*****

state_diagram Slot0IntD
state Slot0IntD_Active:
if ( ((HardReset_B == 0) & (Slot0IntD_PON_DEFAULT == !Slot0IntD_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTD_B & !Slot0IntDMask.fb) #
    Slot0IntDMask.fb)) )
then
!Slot0IntD_Active
else
Slot0IntD_Active ;
state !Slot0IntD_Active:
if ( ((HardReset_B == 0) & (Slot0IntD_PON_DEFAULT == Slot0IntD_Active)) #
    (!(HardReset_B == 0) & !PCI_INTD_B & !Slot0IntDMask.fb) )
then
Slot0IntD_Active
else
!Slot0IntD_Active ;
*****

state_diagram Slot1IntA
state Slot1IntA_Active:
if ( ((HardReset_B == 0) & (Slot1IntA_PON_DEFAULT == !Slot1IntA_Active)) #

```

```
!(HardReset_B == 0) & ((PCI_INTD_B & !Slot1IntAMask.fb) #
Slot1IntAMask.fb)) )
```

then

!Slot1IntA\_Active

else

Slot1IntA\_Active ;

state !Slot1IntA\_Active:

```
if ( ((HardReset_B == 0) & (Slot1IntA_PON_DEFAULT == Slot1IntA_Active)) #
```

```
!(HardReset_B == 0) & !PCI_INTD_B & !Slot1IntAMask.fb) )
```

then

Slot1IntA\_Active

else

!Slot1IntA\_Active ;

\*\*\*\*\*

state\_diagram Slot1IntB

state Slot1IntB\_Active:

```
if ( ((HardReset_B == 0) & (Slot1IntB_PON_DEFAULT == !Slot1IntB_Active)) #
```

```
!(HardReset_B == 0) & ((PCI_INTA_B & !Slot1IntBMask.fb) #
Slot1IntBMask.fb)) )
```

then

!Slot1IntB\_Active

else

Slot1IntB\_Active ;

state !Slot1IntB\_Active:

```
if ( ((HardReset_B == 0) & (Slot1IntB_PON_DEFAULT == Slot1IntB_Active)) #
```

```
!(HardReset_B == 0) & !PCI_INTA_B & !Slot1IntBMask.fb) )
```

then

Slot1IntB\_Active

else

!Slot1IntB\_Active ;

\*\*\*\*\*

```

state_diagram Slot1IntC
state Slot1IntC_Active:
if ( ((HardReset_B == 0) & (Slot1IntC_PON_DEFAULT == !Slot1IntC_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTB_B & !Slot1IntCMask.fb) #
        Slot1IntCMask.fb)) )
then
!Slot1IntC_Active
else
Slot1IntC_Active ;
state !Slot1IntC_Active:
if ( ((HardReset_B == 0) & (Slot1IntC_PON_DEFAULT == Slot1IntC_Active)) #
    (!(HardReset_B == 0) & !PCI_INTB_B & !Slot1IntCMask.fb) )
then
Slot1IntC_Active
else
!Slot1IntC_Active ;
*****

state_diagram Slot1IntD
state Slot1IntD_Active:
if ( ((HardReset_B == 0) & (Slot1IntD_PON_DEFAULT == !Slot1IntD_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTC_B & !Slot1IntDMask.fb) #
        Slot1IntDMask.fb)) )
then
!Slot1IntD_Active
else
Slot1IntD_Active ;
state !Slot1IntD_Active:
if ( ((HardReset_B == 0) & (Slot1IntD_PON_DEFAULT == Slot1IntD_Active)) #
    (!(HardReset_B == 0) & !PCI_INTC_B & !Slot1IntDMask.fb) )
then
Slot1IntD_Active

```

```

else
!Slot1IntD_Active ;
*****
state_diagram Slot2IntA
state Slot2IntA_Active:
if ( ((HardReset_B == 0) & (Slot2IntA_PON_DEFAULT == !Slot2IntA_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTC_B & !Slot2IntAMask.fb) #
        Slot2IntAMask.fb)) )
then
!Slot2IntA_Active
else
Slot2IntA_Active ;
state !Slot2IntA_Active:
if ( ((HardReset_B == 0) & (Slot2IntA_PON_DEFAULT == Slot2IntA_Active)) #
    (!(HardReset_B == 0) & !PCI_INTC_B & !Slot2IntAMask.fb) )
then
Slot2IntA_Active
else
!Slot2IntA_Active ;
*****
state_diagram Slot2IntB
state Slot2IntB_Active:
if ( ((HardReset_B == 0) & (Slot2IntB_PON_DEFAULT == !Slot2IntB_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTD_B & !Slot2IntBMask.fb) #
        Slot2IntBMask.fb)) )
then
!Slot2IntB_Active
else
Slot2IntB_Active ;
state !Slot2IntB_Active:
if ( ((HardReset_B == 0) & (Slot2IntB_PON_DEFAULT == Slot2IntB_Active)) #

```

```

    (!(HardReset_B == 0) & !PCI_INTD_B & !Slot2IntBMask.fb) )
then
Slot2IntB_Active
else
!Slot2IntB_Active ;
*****

state_diagram Slot2IntC
state Slot2IntC_Active:
if ( ((HardReset_B == 0) & (Slot2IntC_PON_DEFAULT == !Slot2IntC_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTA_B & !Slot2IntCMask.fb) #
    Slot2IntCMask.fb)) )
then
!Slot2IntC_Active
else
Slot2IntC_Active ;
state !Slot2IntC_Active:
if ( ((HardReset_B == 0) & (Slot2IntC_PON_DEFAULT == Slot2IntC_Active)) #
    (!(HardReset_B == 0) & !PCI_INTA_B & !Slot2IntCMask.fb) )
then
Slot2IntC_Active
else
!Slot2IntC_Active ;
*****

state_diagram Slot2IntD
state Slot2IntD_Active:
if ( ((HardReset_B == 0) & (Slot2IntD_PON_DEFAULT == !Slot2IntD_Active)) #
    (!(HardReset_B == 0) & ((PCI_INTB_B & !Slot2IntDMask.fb) #
    Slot2IntDMask.fb)) )
then
!Slot2IntD_Active
else

```

```

Slot2IntD_Active ;
state !Slot2IntD_Active:
if ( ((HardReset_B == 0) & (Slot2IntD_PON_DEFAULT == Slot2IntD_Active)) #
    (!(HardReset_B == 0) & !PCI_INTB_B & !Slot2IntDMask.fb) )
then
Slot2IntD_Active
else
!Slot2IntD_Active ;
*****
*****
"* PCI Interrupt Mask Register
*****
*****

equations

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

*****

state_diagram Slot0IntAMask
state Slot0IntAMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntAMask_DATA_BIT.pin == !Slot0IntAMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntAMask_PON_DEFAULT == !Slot0IntAMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntAMask_PON_DEFAULT == !Slot0IntAMask_Active)) )
then
!Slot0IntAMask_Active
else
Slot0IntAMask_Active ;
    
```

```

state !Slot0IntAMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntAMask_DATA_BIT.pin == Slot0IntAMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntAMask_PON_DEFAULT == Slot0IntAMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntAMask_PON_DEFAULT == Slot0IntAMask_Active)) )
then
Slot0IntAMask_Active
else
!Slot0IntAMask_Active ;
*****

state_diagram Slot0IntBMask
state Slot0IntBMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntBMask_DATA_BIT.pin == !Slot0IntBMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntBMask_PON_DEFAULT == !Slot0IntBMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntBMask_PON_DEFAULT == !Slot0IntBMask_Active)) )
then
!Slot0IntBMask_Active
else
Slot0IntBMask_Active ;
state !Slot0IntBMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntBMask_DATA_BIT.pin == Slot0IntBMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntBMask_PON_DEFAULT == Slot0IntBMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntBMask_PON_DEFAULT == Slot0IntBMask_Active)) )
then
Slot0IntBMask_Active
else
!Slot0IntBMask_Active ;

```

```
*****
state_diagram Slot0IntCMask
state Slot0IntCMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntCMask_DATA_BIT.pin == !Slot0IntCMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntCMask_PON_DEFAULT == !Slot0IntCMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntCMask_PON_DEFAULT == !Slot0IntCMask_Active)) )
then
!Slot0IntCMask_Active
else
Slot0IntCMask_Active ;
state !Slot0IntCMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntCMask_DATA_BIT.pin == Slot0IntCMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntCMask_PON_DEFAULT == Slot0IntCMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntCMask_PON_DEFAULT == Slot0IntCMask_Active)) )
then
Slot0IntCMask_Active
else
!Slot0IntCMask_Active ;
*****
state_diagram Slot0IntDMask
state Slot0IntDMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntDMask_DATA_BIT.pin == !Slot0IntDMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntDMask_PON_DEFAULT == !Slot0IntDMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntDMask_PON_DEFAULT == !Slot0IntDMask_Active)) )
then
!Slot0IntDMask_Active
```

```

else
Slot0IntDMask_Active ;
state !Slot0IntDMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot0IntDMask_DATA_BIT.pin == Slot0IntDMask_Active) &
    (!(HardReset_B == 0) # (Slot0IntDMask_PON_DEFAULT == Slot0IntDMask_Active))
#
    ((HardReset_B == 0) & (Slot0IntDMask_PON_DEFAULT == Slot0IntDMask_Active)) )
then
Slot0IntDMask_Active
else
!Slot0IntDMask_Active ;
*****
state_diagram Slot1IntAMask
state Slot1IntAMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot1IntAMask_DATA_BIT.pin == !Slot1IntAMask_Active) &
    (!(HardReset_B == 0) # (Slot1IntAMask_PON_DEFAULT == !Slot1IntAMask_Active))
#
    ((HardReset_B == 0) & (Slot1IntAMask_PON_DEFAULT == !Slot1IntAMask_Active)) )
then
!Slot1IntAMask_Active
else
Slot1IntAMask_Active ;
state !Slot1IntAMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot1IntAMask_DATA_BIT.pin == Slot1IntAMask_Active) &
    (!(HardReset_B == 0) # (Slot1IntAMask_PON_DEFAULT == Slot1IntAMask_Active))
#
    ((HardReset_B == 0) & (Slot1IntAMask_PON_DEFAULT == Slot1IntAMask_Active)) )
then
Slot1IntAMask_Active

```

else

!Slot1IntAMask\_Active ;

\*\*\*\*\*

state\_diagram Slot1IntBMask

state Slot1IntBMask\_Active:

if (VGR\_WRITE\_IntMaskReg &

    (Slot1IntBMask\_DATA\_BIT.pin == !Slot1IntBMask\_Active) &

    (!(HardReset\_B == 0) # (Slot1IntBMask\_PON\_DEFAULT == !Slot1IntBMask\_Active))

#

    ((HardReset\_B == 0) & (Slot1IntBMask\_PON\_DEFAULT == !Slot1IntBMask\_Active)) )

then

!Slot1IntBMask\_Active

else

Slot1IntBMask\_Active ;

state !Slot1IntBMask\_Active:

if (VGR\_WRITE\_IntMaskReg &

    (Slot1IntBMask\_DATA\_BIT.pin == Slot1IntBMask\_Active) &

    (!(HardReset\_B == 0) # (Slot1IntBMask\_PON\_DEFAULT == Slot1IntBMask\_Active))

#

    ((HardReset\_B == 0) & (Slot1IntBMask\_PON\_DEFAULT == Slot1IntBMask\_Active)) )

then

Slot1IntBMask\_Active

else

!Slot1IntBMask\_Active ;

\*\*\*\*\*

state\_diagram Slot1IntCMask

state Slot1IntCMask\_Active:

if (VGR\_WRITE\_IntMaskReg &

    (Slot1IntCMask\_DATA\_BIT.pin == !Slot1IntCMask\_Active) &

    (!(HardReset\_B == 0) # (Slot1IntCMask\_PON\_DEFAULT == !Slot1IntCMask\_Active))

#

    ((HardReset\_B == 0) & (Slot1IntCMask\_PON\_DEFAULT == !Slot1IntCMask\_Active)) )

```

then
!Slot1IntCMask_Active
else
Slot1IntCMask_Active ;
state !Slot1IntCMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot1IntCMask_DATA_BIT.pin == Slot1IntCMask_Active) &
    (!(HardReset_B == 0) # (Slot1IntCMask_PON_DEFAULT == Slot1IntCMask_Active))
#
    ((HardReset_B == 0) & (Slot1IntCMask_PON_DEFAULT == Slot1IntCMask_Active)) )
then
Slot1IntCMask_Active
else
!Slot1IntCMask_Active ;
*****

state_diagram Slot1IntDMask
state Slot1IntDMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot1IntDMask_DATA_BIT.pin == !Slot1IntDMask_Active) &
    (!(HardReset_B == 0) # (Slot1IntDMask_PON_DEFAULT == !Slot1IntDMask_Active))
#
    ((HardReset_B == 0) & (Slot1IntDMask_PON_DEFAULT == !Slot1IntDMask_Active)) )
then
!Slot1IntDMask_Active
else
Slot1IntDMask_Active ;
state !Slot1IntDMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot1IntDMask_DATA_BIT.pin == Slot1IntDMask_Active) &
    (!(HardReset_B == 0) # (Slot1IntDMask_PON_DEFAULT == Slot1IntDMask_Active))
#
    ((HardReset_B == 0) & (Slot1IntDMask_PON_DEFAULT == Slot1IntDMask_Active)) )

```

```

then
Slot1IntDMask_Active
else
!Slot1IntDMask_Active ;
*****

state_diagram Slot2IntAMask
state Slot2IntAMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntAMask_DATA_BIT.pin == !Slot2IntAMask_Active) &
    (!(HardReset_B == 0) # (Slot2IntAMask_PON_DEFAULT == !Slot2IntAMask_Active))
#
    ((HardReset_B == 0) & (Slot2IntAMask_PON_DEFAULT == !Slot2IntAMask_Active)) )
then
!Slot2IntAMask_Active
else
Slot2IntAMask_Active ;
state !Slot2IntAMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntAMask_DATA_BIT.pin == Slot2IntAMask_Active) &
    (!(HardReset_B == 0) # (Slot2IntAMask_PON_DEFAULT == Slot2IntAMask_Active))
#
    ((HardReset_B == 0) & (Slot2IntAMask_PON_DEFAULT == Slot2IntAMask_Active)) )
then
Slot2IntAMask_Active
else
!Slot2IntAMask_Active ;
*****

state_diagram Slot2IntBMask
state Slot2IntBMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntBMask_DATA_BIT.pin == !Slot2IntBMask_Active) &

```

```

        (!(HardReset_B == 0) # (Slot2IntBMask_PON_DEFAULT == !Slot2IntBMask_Active))
#
        ((HardReset_B == 0) & (Slot2IntBMask_PON_DEFAULT == !Slot2IntBMask_Active)) )
then
!Slot2IntBMask_Active
else
Slot2IntBMask_Active ;
state !Slot2IntBMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntBMask_DATA_BIT.pin == Slot2IntBMask_Active) &
    (!(HardReset_B == 0) # (Slot2IntBMask_PON_DEFAULT == Slot2IntBMask_Active))
#
    ((HardReset_B == 0) & (Slot2IntBMask_PON_DEFAULT == Slot2IntBMask_Active)) )
then
Slot2IntBMask_Active
else
!Slot2IntBMask_Active ;
*****
state_diagram Slot2IntCMask
state Slot2IntCMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntCMask_DATA_BIT.pin == !Slot2IntCMask_Active) &
    (!(HardReset_B == 0) # (Slot2IntCMask_PON_DEFAULT == !Slot2IntCMask_Active))
#
    ((HardReset_B == 0) & (Slot2IntCMask_PON_DEFAULT == !Slot2IntCMask_Active)) )
then
!Slot2IntCMask_Active
else
Slot2IntCMask_Active ;
state !Slot2IntCMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntCMask_DATA_BIT.pin == Slot2IntCMask_Active) &

```

```

    (!(HardReset_B == 0) # (Slot2IntCMask_PON_DEFAULT == Slot2IntCMask_Active))
#
    ((HardReset_B == 0) & (Slot2IntCMask_PON_DEFAULT == Slot2IntCMask_Active)) )
then
Slot2IntCMask_Active
else
!Slot2IntCMask_Active ;

```

\*\*\*\*\*

```

state_diagram Slot2IntDMask
state Slot2IntDMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntDMask_DATA_BIT.pin == !Slot2IntDMask_Active) &
    (!(HardReset_B == 0) # (Slot2IntDMask_PON_DEFAULT == !Slot2IntDMask_Active))
#
    ((HardReset_B == 0) & (Slot2IntDMask_PON_DEFAULT == !Slot2IntDMask_Active)) )
then
!Slot2IntDMask_Active
else
Slot2IntDMask_Active ;
state !Slot2IntDMask_Active:
if (VGR_WRITE_IntMaskReg &
    (Slot2IntDMask_DATA_BIT.pin == Slot2IntDMask_Active) &
    (!(HardReset_B == 0) # (Slot2IntDMask_PON_DEFAULT == Slot2IntDMask_Active))
#
    ((HardReset_B == 0) & (Slot2IntDMask_PON_DEFAULT == Slot2IntDMask_Active)) )
then
Slot2IntDMask_Active
else
!Slot2IntDMask_Active ;

```

\*\*\*\*\*

\*\*\*\*\*

" External Read Registers' Chip-Selects

\*\*\*\*\*

\*\*\*\*\*

equations

Bcsr2Cs\_B.oe = H ;

Bcsr4Cs\_B.oe = H ;

!Bcsr2Cs\_B = VGR\_READ\_BCSR\_2 ;

!Bcsr4Cs\_B = VGR\_READ\_BCSR\_4 ;

\*\*\*\*\*

\*\*\*\*\*

"\* Read Registers.

"\* All registers have read capability. (BCSR2 and BCSR4 are read externally)

\*\*\*\*\*

\*\*\*\*\*

equations

DataOe = VGR\_READ\_BCSR\_0 #

    VGR\_READ\_BCSR\_1 #

    VGR\_READ\_BCSR\_3 #

    (HRESET\_CFG\_IN\_BCSR & CS0\_ASSERTED & !DSyncHardReset\_B.fb) ;

Data.oe = DataOe.fb ;

when (VGR\_READ\_BCSR\_0) then

    Data = ReadBcsr0 ;

else when (VGR\_READ\_BCSR\_1) then

    Data = ReadBcsr1 ;

else when (VGR\_READ\_BCSR\_3) then

    Data = ReadBcsr3 ;

```

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;

```

```

DataPCIOe = VGR_READ_IntReg #
    VGR_READ_IntMaskReg ;

```

```

DataPCI.oe = DataPCIOe.fb ;

```

```

when (VGR_READ_IntReg) then
    DataPCI = IntReg.fb ;
else when (VGR_READ_IntMaskReg) then
    DataPCI = IntMaskReg.fb ;

```

```

***** brd_ctl *****
*****
*****
"* Reset Logic
*****
*****
*****

```

equations

```

Reset.oe = ResetEn ;

```

```

Reset = 0 ;" open drain

```

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

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

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

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

TransRst.oe = 7 ;" transceivers' reset, always enabled.

!AtmRstOut\_B = !AtmRst\_B.fb # !HardReset\_B ;

!FEthRstOut1\_B = !FEthRst1\_B.fb # !HardReset\_B ;

!FEthRstOut2\_B = !FEthRst2\_B.fb # !HardReset\_B ;

\*\*\*\*\*

"\* Hard reset configuration

\*\*\*\*\*

"equations

"RstConf\_B.oe = H;

"RstConf\_B = L;

\*\*\*\*\*

"\* NMI generation

\*\*\*\*\*

equations

NMI\_B.oe = NMIEEn ;

NMI\_B = 0 ;" O.D.

NMIEEn = !RstDeb1.com & AbrDeb1.com ;" 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 = ( !Cs0\_B # " covers also hard reset config

!Cs4\_B #

!BrdContRegCs\_B #

!IntContCs\_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 Voyager, hard-reset conf is read from flash/eeprom during HRESET
"* asserted and since these are all consecutive 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 worrying 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_EEPROM_READ # END_OF_ATM_READ )
```

```
  & (HoldOffCnt.fb == 0)) #
```

```
  (HoldOffCnt.fb != 0)) & !(HoldOffCnt.fb == 4) & DSynchHardReset_B.fb ) then
```

```
  HoldOffCnt := HoldOffCnt.fb + 1 ;
```

```
else
```

HoldOffCnt := 0 ;

\*\*\*\*\*

"\* Flash/EEPROM Chip Selects

\*\*\*\*\*

equations

FlashCsOut.oe = ^hf ;

!FlashCs1\_B = CS0\_ASSERTED & FLASH\_BANK1 & HRESET\_BOOT\_IN\_FLASH #  
CS0\_ASSERTED & FLASH\_BANK1 & BOOT\_IN\_FLASH &  
DSyncHardReset\_B.fb #  
CS4\_ASSERTED & FLASH\_BANK1 & (HRESET\_BOOT\_IN\_EEPROM #  
BOOT\_IN\_EEPROM);

!FlashCs2\_B = CS0\_ASSERTED & FLASH\_BANK2 & HRESET\_BOOT\_IN\_FLASH #  
CS0\_ASSERTED & FLASH\_BANK2 & BOOT\_IN\_FLASH &  
DSyncHardReset\_B.fb #  
CS4\_ASSERTED & FLASH\_BANK2 & (HRESET\_BOOT\_IN\_EEPROM #  
BOOT\_IN\_EEPROM);

!FlashCs3\_B = CS0\_ASSERTED & FLASH\_BANK3 & HRESET\_BOOT\_IN\_FLASH #  
CS0\_ASSERTED & FLASH\_BANK3 & BOOT\_IN\_FLASH &  
DSyncHardReset\_B.fb #  
CS4\_ASSERTED & FLASH\_BANK3 & (HRESET\_BOOT\_IN\_EEPROM #  
BOOT\_IN\_EEPROM);

!FlashCs4\_B = CS0\_ASSERTED & FLASH\_BANK4 & HRESET\_BOOT\_IN\_FLASH #  
CS0\_ASSERTED & FLASH\_BANK4 & BOOT\_IN\_FLASH &  
DSyncHardReset\_B.fb #  
CS4\_ASSERTED & FLASH\_BANK4 & (HRESET\_BOOT\_IN\_EEPROM #  
BOOT\_IN\_EEPROM);

EEPromCs\_B.oe = H ;

!EEPromCs\_B = CS0\_ASSERTED & HRESET\_BOOT\_IN\_EEPROM #  
CS0\_ASSERTED & BOOT\_IN\_EEPROM & DSyncHardReset\_B.fb #  
CS4\_ASSERTED & (HRESET\_BOOT\_IN\_FLASH # BOOT\_IN\_FLASH) ;

\*\*\*\*\*

"\* 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 ;

\*\*\*\*\*

\*\*\*\*\*

"\* Generating Interrupt Request to the PQ2.

\*\*\*\*\*

\*\*\*\*\*

equations

PCI\_Interrupt = (Slot0IntA #

Slot0IntB #

Slot0IntC #

Slot0IntD #

Slot1IntA #

Slot1IntB #

Slot1IntC #

Slot1IntD #

Slot2IntA #

Slot2IntB #

Slot2IntC #

Slot2IntD) ;

PCI\_IRQ\_B.oe = PCI\_Interrupt ; " Open-Drain output

!PCI\_IRQ\_B = PCI\_Interrupt ; " Interrupt Request shows after OE

\*\*\*\*\*

"\* Auxiliary functions

\*\*\*\*\*

equations

KeepPinsConnected = TEA\_B & BCTL1 # KeepPinsConnected.com;

\*\*\*\*\*

END

### 7.2.2 U41 - Power switch debounce

MODULE Power\_Debouncer

TITLE 'MPC8280 Power Debouncer'

\*\*\*\*\*

"\* Device declaration. \*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

"\* Pins declaration. \*

\*\*\*\*\*

"\* System i/f pins

\*\*\*\*\*

SYSCLK PIN 5 ;

ChasisPowerIn\_B PIN 15 ; "Chassis Power Switch

PowerOn\_B PIN 16 istype 'reg' ; " Power Supply Power-On

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

"\* Chassis Power Switch Buffer.

\*\*\*\*\*

Power\_Buffer NODE istype 'reg,buffer' ;

```
*****  
** Creating internal clock generator.  
*****
```

```
inv1 NODE istype 'com,keep' ;  
inv2 NODE istype 'com,keep' ;  
inv3 NODE istype 'com,keep' ;  
inv4 NODE istype 'com,keep' ;  
inv5 NODE istype 'com,keep' ;
```

```
counter0,  
counter1,  
counter2,  
counter3,  
counter4,  
counter5,  
counter6,  
counter7,  
countera0,  
countera1,  
countera2,  
countera3,  
countera4,  
countera5,  
countera6,  
countera7,  
counterb0,  
counterb1,  
counterb2,  
counterb3,  
counterb4,
```

```
counterb5,
counterb6,
counterb7     NODE istype 'reg,buffer' ;
```

```
*****
```

```
*****
```

```
H, L, X, Z = 1, 0, .X., .Z. ;
```

```
C, D, U  = .C., .D., .U. ;
```

```
*****
```

```
"* SIMULATION = 1 ;
```

```
*****
```

```
"* Signal groups
```

```
*****
```

```
counter  = [counter7,counter6,counter5,counter4,
            counter3,counter2,counter1,counter0] ;
```

```
countera = [countera7,countera6,countera5,countera4,
            countera3,countera2,countera1,countera0] ;
```

```
counterb = [counterb7,counterb6,counterb5,counterb4,
            counterb3,counterb2,counterb1,counterb0] ;
```

```
*****
```

```
"* ATX Power Declarations.
```

```
*****
```

```
PowerOn = 0 ;
```

```
PowerOff = 1 ;
```

```
*****
```

```
"* Equations, state diagrams. *
```

\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*

equations

\*\*\*\*\*  
\*\*\*\*\*  
"\* Generating PowerOn signal to the ATX Power Supply.  
\*\*\*\*\*  
\*\*\*\*\*

equations

inv1 = !inv5.com ;" generating internal clock oscilator  
inv2 = !inv1.com ;  
inv3 = !inv2.com ;  
inv4 = !inv3.com ;  
inv5 = !inv4.com ;

counter.ar = 0 ;  
counter.ap = 0 ;  
counter.clk = !inv5.com ;

when ( counter.fb == 255 ) then counter := 0 else counter := counter + 1 ;

countera.ar = 0 ;  
countera.ap = 0 ;  
countera.clk = ( counter.fb == 0 ) ;

when ( countera.fb == 255 ) then countera := 0 else countera := countera + 1 ;

```

counterb.ar = 0 ;
counterb.ap = 0 ;
counterb.clk = ( countera.fb == 0 ) ;

when ( counterb.fb == 255 ) then counterb := 0 else counterb := counterb + 1 ;

```

```

Power_Buffer.ar = 0 ;
Power_Buffer.ap = 0 ;
Power_Buffer.clk = ( counterb.fb == 0 ) ;
Power_Buffer := ChasisPowerIn_B ;

```

```

PowerOn_B.oe = H ;
PowerOn_B.ar = 0 ;
PowerOn_B.ap = 0 ;
PowerOn_B.clk = ( counterb.fb == 0 ) ;
PowerOn_B := !Power_Buffer.fb ;

```

```

*****

```

```

*****

```

```

@ifdef SIMULATION {
}
END

```

### 7.3 Bill Of Materials

The following is the Bill Of Materials for the PQ2FADS-VR including the L2Cache option.

**Table 7-10. PQ2FADS-VR Bill of Materials**

Item	Quantity	Reference	Part	Part Number	Manufacturer
1	4	"C1,C2,C3,C5"	0.01UF-2KV	202S49W103KV4E	JOHANSON DIELECTRIC
2	19	"C4,C16,C17,C24,C40,C42," "C43,C44,C47,C49,C54,C62," "C65,C80,C81,C90,C122," "C123,C256"	47uF	TAJD476K016	AVX

Table 7-10. PQ2FADS-VR Bill of Materials

Item	Quantity	Reference	Part	Part Number	Manufacturer
3	281	"C6,C7,C10,C11,C12,C13," "C14,C15,C18,C19,C20,C21," "C22,C23,C31,C32,C34,C36," "C37,C38,C39,C41,C45,C46," "C48,C50,C51,C52,C53,C55," "C58,C59,C61,C64,C66,C67," "C68,C69,C70,C71,C72,C73," "C74,C75,C76,C77,C78,C79," "C84,C85,C86,C87,C88,C89," "C91,C92,C93,C94,C95,C96," "C97,C98,C99,C100,C101," "C102,C103,C104,C105,C106," "C107,C108,C109,C110,C111," "C112,C113,C125,C126,C127," "C129,C130,C131,C132,C133," "C134,C142,C143,C144,C145," "C146,C147,C151,C152,C153," "C154,C155,C156,C157,C158," "C161,C162,C163,C164,C165," "C166,C167,C168,C169,C170," "C171,C176,C177,C178,C179," "C180,C181,C182,C183,C184," "C185,C186,C187,C188,C189," "C190,C191,C196,C197,C198," "C199,C200,C201,C202,C203," "C204,C205,C206,C207,C208," "C209,C210,C211,C212,C213," "C214,C218,C219,C223,C225," "C226,C227,C228,C229,C230," "C231,C232,C233,C234,C235," "C238,C241,C244,C245,C248," "C249,C250,C251,C255,C257," "C261,C262,C263,C266,C267," "C268,C269,C270,C271,C275," "C276,C277,C278,C281,C282," "C283,C284,C285,C286,C289," "C290,C291,C292,C293,C294," "C295,C296,C297,C298,C299," "C300,C302,C303,C304,C305," "C306,C307,C308,C309,C310," "C311,C312,C313,C314,C315," "C316,C319,C320,C321,C322," "C323,C324,C325,C326,C327," "C328,C329,C330,C331,C332," "C333,C334,C335,C336,C337," "C340,C341,C342,C343,C344," "C345,C346,C347,C350,C351," "C352,C353,C354,C355,C356," "C357,C358,C359,C360,C361," "C362,C363,C364,C365,C366," "C367,C368,C369,C370,C371," "C372,C373,C374,C375,C376," "C377,C378,C379,C380,C381,"	100nF	0603YC104KAT2A	AVX

### Table 7-10. PQ2FADS-VR Bill of Materials

Item	Quantity	Reference	Part	Part Number	Manufacturer
		"C382,C383,C384,C404,C406," C407			
4	19	"C8,C9,C25,C26,C27,C28," "C29,C33,C35,C56,C57,C83," "C137,C141,C148,C279,C301," "C405,C408"	10uF	TAJC106K025R	AVX
5	55	"C30,C128,C135,C136,C138," "C139,C140,C149,C150,C159," "C160,C172,C173,C174,C175," "C192,C193,C194,C195,C215," "C216,C217,C220,C221,C222," "C224,C236,C237,C239,C240," "C242,C243,C246,C247,C252," "C253,C254,C258,C259,C260," "C264,C265,C273,C274,C280," "C287,C288,C317,C318,C338," "C339,C348,C349,C385,C402"	10nF	06035C103KAT2A	AVX
6	1	C60	100uF	TAJD107K016R	AVX
7	1	C63	68uF-16V	TAJD686M020R	AVX
8	1	C82	1uF	B45196H5105K109	SIEMENS
9	24	"C114,C115,C116,C117,C118," "C119,C120,C121,C386,C387," "C388,C389,C390,C391,C392," "C393,C394,C395,C396,C397," "C398,C399,C400,C401"	1nF	AVX12065C102KA	AVX
10	2	"C403,C124"	100nF-500V	501S43W104MV4E	JOHANSON
11	1	C272	1500pF	12065A152JAT00J	AVX
12	3	"D1,D2,D3"	LL4004	LL4004G	TSC
13	1	F1	SMD150/33-2	SMD150/33-2	RAYCHEM
14	5	"JP1,JP5,JP8,JP11,JP12"	GND_Bridge254	PD-999-11-11010	PRECIDIP
15	8	"JP2,JP3,JP4,JP6,JP7,JP9," "JP10,JP13"	JUMPER1x3	87156-0303	MOLEX
16	7	"J1,J2,J3,J4,J5,J6,J7"	GND_Bridge	PD-999-11-11210	PRECIDIP
17	18	"LD1,LD2,LD3,LD4,LD5,LD6," "LD9,LD14,LD15,LD16,LD17," "LD19,LD20,LD23,LD25,LD26," "LD27,LD28"	LED_GREEN	KPT-3216SGD	KINGBRIGHT
18	7	"LD7,LD8,LD10,LD11,LD12," "LD21,LD24"	LED_YELLOW	KPT-3216YD	KINGBRIGHT
19	1	LD13	LED_RED	KPT-3216YD	KINGBRIGHT
20	2	"LD22,LD18"	LED_RED	KPT-3216ID	KINGBRIGHT

**Table 7-10. PQ2FADS-VR Bill of Materials**

Item	Quantity	Reference	Part	Part Number	Manufacturer
21	2	"L1,L2"	BEAD_FERRITE	2743021447	FAIR RITE
22	4	"L3,L4,L21,L22"	NFM60R30T222T1	NFM60R30T222T1	MURATA
23	17	"L5,L6,L7,L8,L9,L10,L11," "L12,L13,L14,L15,L16,L17," "L18,L19,L20,L23"	BLM18AG121SN1	BLM18AG121SN1	MURATA
24	1	P1	RS232-PORT2	8LE009009D306H	EDA
25	1	P2	787616-1	787616-1	AMP
26	2	"P4,P3"	RJ45	43202-8110	MOLEX
27	1	P5	ATX_Power_Connect or	39-29-9202	MOLEX
28	2	"P8,P6"	QSE-020-01-L-D-A	QSE-020-01-L-D-A	SAMTEC
29	2	"P7,P27"	23762	23762	ERNI
30	11	"P9,P10,P11,P15,P17,P18," "P19,P24,P28,P29,P30"	MICTOR38	2-767004-2	AMP
31	3	"P12,P13,P14"	PCI_CONNECTOR	145154-4	AMP
32	1	P16	COP/JTAG	LPH-16SA-SG	KCC
33	4	"P20,P21,P23,P25"	CON10AP	TSM-10501-SDV-AP	SAMTEC
34	1	P22	SMB Straight	82SMB-50-0-1/111	SUHNER
35	1	P26	CON6AP	HEADER 3x2 SMT	SAMTEC
36	1	P31	DNR-25PCB-SG	DNR-25PCB-SG	KCC_Keltron
37	2	"Q2,Q1"	MMDF2P02HD	MMDF2P02HD	ON SEMI- CONDUCTOR
38	2	"Q4,Q3"	MMDF4N01HD	MMDF4N01HD	ON SEMI- CONDUCTOR
39	31	"RN1,RN6,RN7,RN8,RN9,RN13," " "RN14,RN15,RN17,RN24,RN25," " "RN26,RN37,RN38,RN48,RN61," " "RN64,RN65,RN66,RN67,RN68," " "RN69,RN70,RN71,RN72,RN78," " "RN82,RN83,RN84,RN85,RN87"	10K	RS8A1002J	ROHM

### Table 7-10. PQ2FADS-VR Bill of Materials

Item	Quantity	Reference	Part	Part Number	Manufacturer
40	33	"RN2,RN3,RN4,RN5,RN10," "RN11,RN12,RN27,RN28,RN29, "RN30,RN31,RN32,RN33,RN34, "RN35,RN36,RN39,RN41,RN44, "RN45,RN46,RN49,RN50,RN55, "RN56,RN57,RN58,RN62,RN63, "RN73,RN74,RN86"	22	CRA3A4E220JT	AVX
41	4	"RN16,RN18,RN19,RN60"	470		
42	6	"RN20,RN21,RN22,RN23,RN75, RN76"	43	CRA06S0803430JRT	DALE
43	12	"RN40,RN42,RN43,RN47,RN51, "RN52,RN53,RN59,RN77,RN79, "RN80,RN81"	0	CRA06S0803000RT	DALE
44	1	RN54	33		AVX
45	2	"RP1,RP2"	1K	3362P-1-102	BOURNS
46	1	R1	2K2	D2502K2FCS	ROEDER-STEIN
47	83	"R2,R37,R48,R49,R50,R96," "R122,R126,R127,R128,R130," "R131,R141,R142,R144,R148," "R150,R153,R155,R156,R157," "R159,R164,R165,R166,R172," "R173,R179,R181,R187,R188," "R189,R192,R195,R196,R201," "R202,R203,R204,R206,R212," "R213,R214,R215,R216,R221," "R225,R226,R227,R230,R231," "R253,R266,R275,R277,R281," "R282,R283,R284,R291,R292," "R293,R294,R295,R296,R305," "R309,R310,R313,R323,R327," "R328,R333,R334,R341,R349," "R350,R361,R362,R363,R370," "R372,R373"	10K	D11010KFCS	ROEDER-STEIN
48	8	"R3,R4,R5,R6,R116,R117," "R118,R119"	75	D11075RFCS	DRALORIK
49	4	"R7,R13,R16,R17"	49R9	D2549R9FCS	ROEDER-STEIN
50	4	"R8,R11,R12,R15"	78R7	D2578R7FCS	D2578R7FCS
51	2	"R9,R10"	158	CRCW0603-1580F	DALE
52	25	"R14,R18,R22,R23,R34,R35," "R38,R39,R51,R52,R53,R54," "R182,R183,R190,R191,R256," "R257,R259,R279,R290,R297," "R298,R299,R300"	22R1	D1122R1FCS	ROEDER-STEIN

**Table 7-10. PQ2FADS-VR Bill of Materials**

Item	Quantity	Reference	Part	Part Number	Manufacturer
53	48	"R19,R28,R33,R36,R41,R71," "R75,R76,R77,R80,R93,R94," "R95,R132,R184,R193,R194," "R218,R219,R240,R241,R242," "R248,R249,R258,R260,R261," "R262,R269,R270,R271,R303," "R308,R311,R314,R320,R325," "R330,R331,R335,R336,R338," "R342,R348,R357,R360,R369," R371	1K	D11001KFCS	DRALORIK
54	44	"R20,R21,R24,R25,R26,R27," "R30,R31,R32,R40,R72,R82," "R139,R140,R145,R151,R152," "R161,R162,R176,R186,R197," "R198,R199,R200,R207,R208," "R209,R210,R211,R228,R229," "R232,R233,R234,R235,R236," "R237,R245,R246,R247,R264," "R286,R287"	4K7	D1104K7FCS	ROEDER-STEIN
55	2	"R29,R42"	10	D25010RFCS	ROEDER-STEIN
56	5	"R43,R44,R45,R46,R47"	33R2	D11 33R2FCS	ROEDER-STEIN
57	16	"R55,R56,R57,R58,R59,R61," "R63,R64,R65,R217,R222," "R243,R252,R254,R255,R356"	43R2	D1143R2FCS	ROEDER-STEIN
58	19	"R60,R88,R89,R90,R91,R92," "R114,R174,R238,R239,R265," "R272,R273,R321,R322,R324," "R326,R332,R337"	0	D11000RFCS	ROEDER-STEIN
59	2	"R62,R78"	0R005	WSL2512 0.005ohm 1%	DALE
60	15	"R66,R67,R68,R83,R86,R87," "R316,R318,R343,R344,R351," "R353,R354,R366,R367"	330		
61	15	"R69,R70,R73,R74,R84,R85," "R317,R319,R345,R346,R347," "R352,R364,R365,R368"	220		
62	1	R79	110	CR32111J-T	AVX
63	21	"R81,R97,R98,R99,R100," "R101,R102,R103,R104,R105," "R106,R107,R110,R112,R120," "R149,R158,R170,R177,R180," R359	330	D25330RJCS	ROEDER-STEIN
64	2	"R111,R108"	1K	D25001KFCS	DRALORIK

### Table 7-10. PQ2FADS-VR Bill of Materials

Item	Quantity	Reference	Part	Part Number	Manufacturer
65	10	"R109,R143,R146,R147,R154," "R167,R168,R355,R374,R375"	150	D25150RFCS	ROEDER-STEIN
66	1	R113	47K	D25047KFCS	ROEDER-STEIN
67	1	R115	0R5	D250R50FCS	ROEDER-STEIN
68	12	"R121,R125,R171,R178,R244," "R250,R251,R263,R267,R268," "R276,R285"	1K5	D2501K5FCS	ROEDER-STEIN
69	2	"R124,R123"	24R3	D2524R3FCS	ROEDER-STEIN
70	1	R129	330	D25 330RJCS	ROEDER-STEIN
71	1	R133	220	D25220RJCS	DRALORIK
72	4	"R134,R135,R274,R304"	51R1	D1151R1FCS	ROEDER-STEIN
73	3	"R136,R137,R138"	150	D25 150RJCS	ROEDER-STEIN
74	2	"R185,R160"	510	CRCW0603-5100F	DALE
75	2	"R163,R169"	6K8 1%	D25 06K8FCS	ROEDER-STEIN
76	2	"R205,R175"	2R7	D2502R7FCS	ROEDER-STEIN
77	3	"R220,R223,R224"	20	D25020RFCS	DRALORIK
78	4	"R278,R280,R288,R289"	300	xxx	ROEDER-STEIN
79	4	"R301,R302,R306,R307"	0	D25000RFCS	ROEDER-STEIN
80	1	R312	300	CR0805-JW-301	BOURNS
81	1	R315	5K6	D2505K6FCS	ROEDER-STEIN
82	1	R329	3K	D25-03KJ-S	DRALORIK
83	1	R339	172	CRCW0603-1720F	DALE
84	1	R340	21R5	CRCW0603-21R5F	DALE
85	1	R358	243	D25243RFCS	ROEDER-STEIN
86	1	SW1	POWER-ON_RESET	KS12-R23-CQE	C&K

**Table 7-10. PQ2FADS-VR Bill of Materials**

Item	Quantity	Reference	Part	Part Number	Manufacturer
87	1	SW2	ABORT	KS12-R21-CQE	C&K
88	1	SW3	SOFT-RESET	KS12-R22-CQE	C&K
89	1	SW4	E101MD1ABE	E101MD1ABE	C&K
90	1	SW5	SW_DIP-8/SM	90HBW08S	GRAYHIL
91	2	"SW7,SW6"	SW_DIP-4/SM	90HBW04SR	GRAYHIL
95	1	U1	HFBR-5805	HFBR-5805	AGILENT
96	1	U2	S-80828ANMP	S-80828ANMP-EDR-T2	SEIKO
97	2	"U51,U3"	74ACT541	74ACT541DW	ON SEMI-CONDUCTOR
98	2	"U4,U5"	MAX3241ECAI	MAX3241ECAI	
99	1	U6	MIC5209-2.5BS	MIC5209-2.5BS	MICREL
100	1	U7	PDIUSBP11A	PDIUSBP11A	PHILIPS
101	2	"U8,U9"	TG22-3506	TG22-3506ND	HALO
102	6	"U10,U17,U34,U44,U46,U48"	74LCX125	MC74LCX125DT	MOTOROLA
103	1	U11	PM5384-NI	PM5384-NI	PMC SIERRA
104	2	"U13,U12"	DM9161	DM9161E	DAVICOM
105	11	"U14,U15,U21,U23,U24,U28," "U29,U32,U41,U55,U63"	IDT74CBTLV3257PG	IDT74CBTLV3257PG	
106	2	"U57,U16"	IDTQS3VH16233-PV	IDTQS3VH16233-PV	IDT
107	8	"U18,U19,U20,U22,U59,U60," "U61,U62"	IDTQS34XV245Q3	IDTQS34XV245Q3	IDT
108	1	U25	M4A3_192/96	M4A3-192/96-6VC	LATTICE
109	1	U26	MPC8275		
110	1	U27	SN74LVT8980ADWR	SN74LVT8980ADWR	
111	1	U30	MPC9447		
112	1	U31	MIC29500BT	MIC29500-3.3BT	MICREL
113	2	"U64,U33"	MPC2605	MPC2605ZP66	MOTOROLA
114	3	"U35,U39,U40"	74LCX541	MC74LCX541DT	MOTOROLA
115	1	U36	MPC9448	MPC9448FA	MOTOROLA
116	1	U37	M4A5-64/32-VC48	M4A5-64/32-7VC48	VANTIS
117	1	U38	SN74ALVCH162260	SN74ALVCH162260	TI
118	1	U42	EPM3064ATC100-10	EPM3064ATC100-10	ALTERA

### Table 7-10. PQ2FADS-VR Bill of Materials

Item	Quantity	Reference	Part	Part Number	Manufacturer
119	1	U43	74LCX16245	MC74LCX16245DT	ON SEMI-CONDUCTOR
120	1	U45	74LVXZ161284MTD	74LVXZ161284MTD	TI
121	1	U47	AT28HC64B-70JC	AT28HC64B-70JC	ATMEL
122	1	U49	LM317D2T	LM317D2T	ON SEMI-CONDUCTOR
123	1	U50	FLASH_SIMM80	SM73228XG1JHBG0	SMART MODULAR TECHNOLOGIES
124	1	U52	74LCX74D	MC74LCX74D	ON Semiconductor
125	3	"U53,U54,U58"	CY2309ZC-1H	CY2309ZC-1H	Cypress
126	1	U56	PI3B33X257B	PI3B33X257B	Pericom
127	3	"U65,U71,U72"	74LCX16244	MC74LCX16244DT	MOTOROLA
128	3	"U66,U67,U68"	MT48LC4M32B2TG	MT48LC4M32B2TG	MICRON
129	2	"U69,U70"	74ALVT16245	74ALVT16245DL	PHILIPS
130	2	"U73,U74"	74ALVT16373	74ALVT16373DL	PHILIPS
131	1	X1	M216TCN50.00	M216TCN50.00	M-TRON
132	1	X2	M218TCN-19.44MHz	M218TCN-19.44MHz	M-TRON
133	1	X3	M216TCN-48.00MHz	M216TCN-48.00MHz	M-TRON
134	1	X4	66MHz-3V3	M3H16FCD-3V3-66MHz	M-TRON
135	1	X5	40MHz-3V3	M3H16FCD-3V3-40MHz	M-TRON
136	1	X6	1.10933E+12	1.10933E+12	PRECIDIP