PXECORE

User's Guide

Rev 0.4, 2 February 1999

Table of contents

Table of contents

1. Introduction

The CORE module is a tiny, high-performance embedded computer, designed to serve as a building block in electronic and communication products. The card has been extremely optimized in terms of cost, performance and level of integration. Special effort has been made to allow easy mechanical connection of the module to the target card.

The PXECORE design addresses two opposing issues. A building block module such as the PXECORE must be cost effective for volume production. On the other hand, each application has different requirements in terms of performance, memory and peripherals needed. This flexibility usually increases the cost of the product. CompuLab offers the perfect solution to this problem. The PXECORE embedded peripheral offers the flexibility of selecting the most suitable configuration without paying for extra hardware. Peripherals may be selected from a rich library of I/O ports, UART, USRT, DMA and more. To ease the development stage, a more powerful combination of peripherals may be selected, and then removed when production starts. This guarantees that the product will not carry the surplus cost of development components.

The CORE module has all the components needed to run an operating system independent of the surrounding environment. It is supplied with MON960. CompuLab has a BSP for VXWORKS and can supply BSP for other RTOS.

Key Product Features

- **Intel 80960Rx processor @ 33..100 MHz**
- **4-32 MB of Dynamic RAM**
  - High performance 1-0-0-0.... DRAM controller
  - Up to 4KB burst length
- **Up to 16 MB of FLASH ROM**
  - On board re-programmable
  - 4 MB to 8 MB, 32-bit, 0 wait states SyncFlash
  - 2 MB to 4 MB, 8-bit, 0 wait states SyncFlash
  - 2 MB to 16MB 32 bit StrataFlash
- **Complete PCI bus host interface**
  - PCI arbiter controls 4 external PCI agents
  - PCI clock source
  - All card resources are accessible from PCI
  - 2 DMA engines
- **PCI Bridge**
- **Fast Ethernet controller**
  - 100Base-Tx RJ45 connector
  - MII connector
- **Dual UART**
  - RS232 or TTL level option for EMI suppression
- **JTAG interface**
- **Powerful external CPU bus interface**
  - QUIET and FAST bus interface modes
  - Guarded functionality of CORE local components
  - Direct interface to external components
  - Programmable Chip Select lines for external devices
- **Large variety of mounting options**
  - Miniature 50 mil interface connectors
  - User selectable mounting height up to 17 mm.

2. Overview

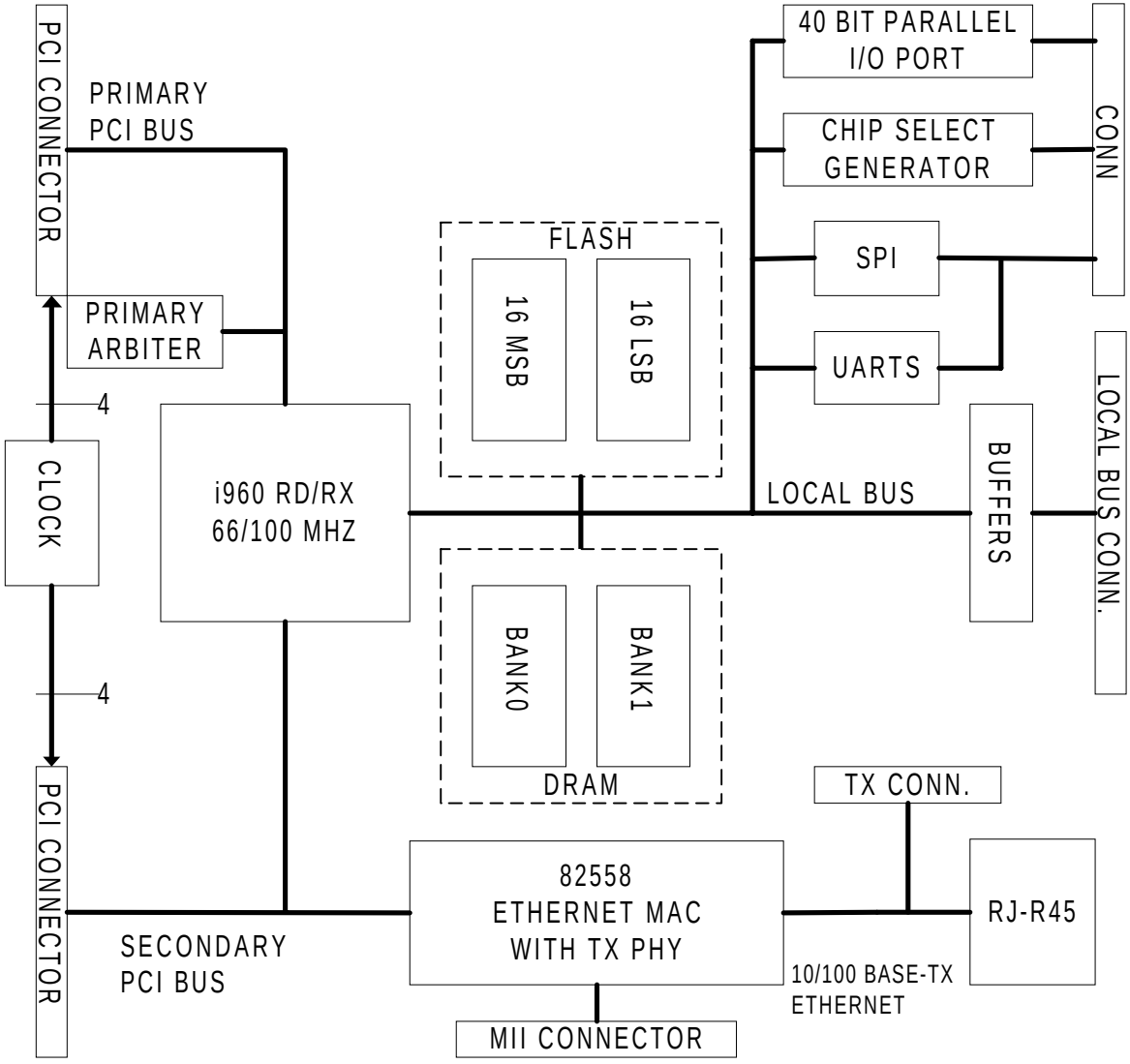


Figure 1- PXECORE block diagram

The PXECORE is based on Intel's 80960Rx RISC CPU. The CPU has an internal PCI bridge and a memory controller. The board has an on-board 10/100 Ethernet controller (Intel 82558). Also on board are 2 DRAM banks of 32 bit, and 1-2 Flash chips. The board also has 2 RS-232 serial channels, and 50 programable I/O pins.

PCI

The i960Rx PCI bridge has an arbiter for its secondary PCI bus. The PXECORE has a PCI arbiter which is connected to the primary PCI. It is intended for stand-alone applications in which the PXECORE manages all PCI bus resources. This arbiter may be disabled. To complete its stand-alone capability, the PXECORE supplies 8 PCI clock outputs to drive all other PCI devices. The clock source can be from an on-board oscillator, or driven from an external source.

Memory

The DRAM controller is programmable, and may be tuned for several types of DRAM memories (but not SDRAM). DRAM performance is 1-0-0-0.....-0 for burst access. The DRAM controller will burst data to/from the PCI until it reaches the DRAM page boundary. Two DRAM banks are available.
FLASH memory has a selection of several devices. The 28F016XS sync FLASH gives the best performance. The StrataFlash provides more space. FLASH is configured as an 8 or 32 bit device.

Local Bus and Peripherals

The PXECORE uses a 10-40K gate FPGA for all local bus peripherals and functions. In its basic configuration it includes 1 or 2 UART's, 4 programmable chip select lines and some I/O lines. Additional peripherals may be added. The number of UART's may increase to 8 (or more in some cases) and Chip select lines and I/O lines may be mixed. The total number of I/O lines is 50.
The PXECORE buffers the local bus into the interface connectors. The external interface may either use the latched address bus or the multiplexed address/data lines and de-multiplex them using the ADS/ALE lines. External bus masters are supported using the Hold / Holda lines.

Fast Ethernet

The 82558 Intel's fast Ethernet controller is used for the Ethernet link. The device is connected to the secondary PCI bus. Two physical line interfaces are supported. The first uses the built-in TX interface which is attached to an RJ-45 connector. The second alternative is to use the standard MII connection and drive an external PHY device such as T4 or FX.

Table 1. PXECORE component options

Component	Option	Data bus width	Performance (MB/s) @ 33 MHz	Notes
DRAM	4 MB 16 MB 32 MB	323232	133133133	1
FLASH	2 MB 4 MB 8 MB	83232	206644	2
CPU	RD66 RN100	32	-	3
UART	1 TO 8	8	-	4
SPI (USRT)	1 TO 8	8	-	4
User Defined features	Variety of controllers and I/O embedded in FPGA			

Component	Option	Data bus width	Performance (MB/s) @ 33 MHz	Notes
I/O ports	4 -50 bit	-	-	4
Ethernet	TX or MII	PCI 32		

Notes:

1. DRAM performance is specified for long bursts, as is typical of block transfer operations.
2. FLASH performance is specified for typical code fetch operations, as performed by the CPU. The 2 MB and 4 MB performance indicators are based on SyncFLASH.
3. For details see Intel's *'i960 Processors and Related Products'*.
4. *CompuLab* provides a variety of serial and parallel interfaces embedded in the PXECORE on-board FPGA. Some options are mutually exclusive. Call *CompuLa*b's technical support for details.

3. Getting started

3.1. Jumper configuration

The PXECORE has a set of jumpers that enables control of the external board interface and some internal functions. Before powering up and connecting the PXECORE, you should configure it to meet your system's needs. There are two major configurations:

A. PXECORE is the PCI clock and arbitration source, Reset is an output. In this case note that E5 and E9 jumpers are dismounted and E8's position is 1-2.

B. PXECORE PCI arbiter is disabled (float), and the PCI clock is fed into the PXECORE. Reset is an input. In this case E5 and E9 jumpers should be mounted, and E8's position is 2-3. We use this configuration when the PXECORE is stacked into a PC with a PCI bus. (Using the PCREVA evaluation board as a carrier.)

Details of each connector and jumper are found in the "Connectors and Jumpers" Appendix at the end of this manual.

3.2. Attaching the serial cable

PXECORE has a 3 pin header for Tx/Rx RS-232 communication. Header ref is P1. The evaluation kit of the PXECORE contains one serial cable. The other end of the serial cable may be connected to any DB9 serial port. The connected port is UART B.

When using the PCREVA, you may connect the serial ports through the PCREVA DB9 connectors. PCREVA P6 (upper) is UART A, P7 (lower) is UART B (default for MON960).

3.3. Checking out the board

PXECORE comes with the Intel MON960 monitor program. It communicates with a host through the serial ports.

- Start up a terminal emulation program with the following communication setup: Baud rate between 9600 and 115200, 8 data bits, Parity - NONE, 1 stop bit, flow control - NONE, terminal emulation - TTY (terminal emulation is not critical).
- Power up the PXECORE. Watch the FAIL LED. It should turn off after a few seconds. Watch the ALARM LED - it should be off.

At this time, if both the FAIL and the ALARM LED's are off, the PXECORE has passed its initial startup stage and is waiting for an input from the serial port.

- Hit the <ENTER> key a few times (x 8 or more). This enables the MON960 to sense the host's baud rate and synchronize with it. You should see the MON960's startup message.
- Optional - test the memory subsystem.
    - Type PO <ENTER>. MON960 replies with a menu of tests. Select memory tests (single or infinite).
    - Enter start address C0008000
    - Enter block size 3f8000 for 4 MB DRAM or 7f8000 for 8 MB DRAM.
    - Inspect the test results.

3.4. Downloading and executing programs

Programs written for the PXECORE can be downloaded and debugged with either of two programs. The first is MON960:

- Type DO <ENTER> at the monitor prompt.
- Using the terminal emulation program, send (upload) the file using the XMODEM protocol.
- When the upload is done, MON960 shows the start address. Type G <ENTER> to run the program.

GDB960:
Usage of GDB960 does not require a terminal emulation program. GDB960 performs all communication with the MON960. GDB960 can be operated in command line mode or (preferred) in a GUI-like debugger.
Command line mode:

- Reset the PXECORE
- Enter the GDB960 command line, specifying the file to download.
- When download has finished, type r <ENTER> at the GDB960 command line.

GUI mode:

- Run the program found in: Start->Programs->Ctools->GDB960V
- Select the Target Connect method (select serial or PCI)
- Use the menus to download and symbolically debug your program.

GDB960 is more complicated than MON960 and has slightly faster downloads compared with terminal emulations with MON960.

3.5. Compiling and linking your own code

CompuLab provides a demo program with makefile, compiler and linker switch files. Refer to the programmer reference for in-depth information.

3.6. Recommended literature

- Core Module Embedded Peripherals User's guide - a *CompuLab* document.
- i960 Processors and Related Products. Intel order number 272084
- i960 Rx Microprocessor User's Manual. Intel order number 272736
- FLASH memory (Volume 1). Intel order number 210830
- 16-Mbit FLASH Product Family User's Manual. Intel order number 297372
- Intel 82558 data sheet
- PCI Local Bus Specification Revision 2.1
- National Semiconductor or Startech data sheet for the 16C1450 UART.

3.7. Revision History

Table 2. Revision History

Revision	Date	Comments
0	03/May/98	Beta release.
0.1	28/Oct/98	Review the connectors section.
0.2	02/Jan/99	Change memory map .Do some cleanup.
0.3	19/Jan/99	Add mechanicals drawings, add I/O ports section
0.4	02/Feb/99	Change memory map

4. Motherboard Design considerations

This section aims to help the hardware engineer with interfacing the PXECORE module.
The following are recommended guidelines.

4.1. Connecting to the PCI bus

4.1.1. Number of agents to connect

PXECORE is designed to support up to four external PCI devices. This limitation is imposed by the PXECORE arbiter. The PXECORE arbiter may be expanded at the expense of other I/O lines. The maximum number of PCI agents should be calculated as follows:

1. Allow a maximum of 10 devices on a PCI segment.
2. An edge connector is treated as a load.
3. PXECORE is 1 (one) PCI load.

Example A:
Q - A system with PXECORE + 1 PCI IC on the motherboard. How many edge connector slots can be added?
A - PXECORE = 1, PCI IC = 1.
10 - (1 + 1) = 2 * slot
slots = 4

Example B:
Q - how many PCI IC's can be directly (without slots) attached to the PXECORE ?
A- PXECORE = 1, PCI IC = 1.
10 - 2 = IC = 8
In this case, you will have to ask for arbiter customization, or disable the PXECORE arbiter, and design your own 9 device arbiter (8 external + PXECORE's PCI9080).
(Also see the clock section).

Note: The above examples refer to the PRIMARY PCI port of the i960Rx. When connecting to the SECONDARY PCI port, you should take into account 2 loads: the CPU PCI port and the 82558 Ethernet controller.

4.1.2. PCI clock scheme

The PXECORE supplies four PCI clock outputs per pci bus segment. The trace lengths should be kept equal on all outputs.

- The PCI clock trace length should not exceed TBD cm in length.
- Do not place more than one load per clock output. If you must: 1. Make the segment between the shared loads as short as possible, and 2. Select the lowest possible capacitance combination from the loads.
- The PXECORE has series termination resistors of 22 ohm on-board for each PCI clock output. Do not

use parallel terminations. Call *CompuLab*'s technical support for help if your clock signal is not properly terminated. Put a GND test point near the clock pin of every PCI device. Use it when measuring the clock's signal quality.

- Remember that the PCI allows a maximum of 2 ns clock skew between devices!

4.1.2.1. If your system needs more clock sources than the PXECORE can supply

In this case you will probably use a multi-output, low skew buffer to drive the PCI clock.

- Trace length should be kept equal on all outputs. The PXECORE internal path is TBD cm long.
- Do not use the PXECORE clock outputs, to avoid double skew (your clock driver's skew + PXECORE's internal PLL skew).
- The E9 jumper should be assembled. This tells the PXECORE to use the external PCI clock source and not its on-board oscillator.
- The PXECORE PCI clock input is P2-55.
- Signal quality of the input PCI clock should be verified. Call *CompuLab*'s technical support for details.

4.1.3. PCI Arbiter

The i960Rx supplies a PCI arbiter for its secondary port. In cases when an arbiter for the primary port is needed, the on-board PCI arbiter can be used. The PXECORE PCI arbiter supports four external PCI devices. Arbitration takes place in parallel to the bus activity, and there is no cycle penalty because of arbitration. Arbitration is fair - it implements a rotating priority algorithm[1].

The PXECORE arbiter may be disabled either by software or hardware.

- By hardware - assemble the E5 jumper
- By software - clear bit 4 of the FPGA control register (0xE0000004). This bit is set after reset.

The disable action overrides the enable. This means that in order to use the arbiter, the E5 jumper must be removed and bit 4 of the FPGA control register must be set to 1.

NOTE: When using an external arbiter, the PXECORE must be included in the arbitration scheme. The relevant signals are: -CORE_REQ, and -CORE_GNT.

[1] - The PCI arbiter may be customized. Contact *CompuLab*'s technical support for details.

4.1.4. PCI Board layout

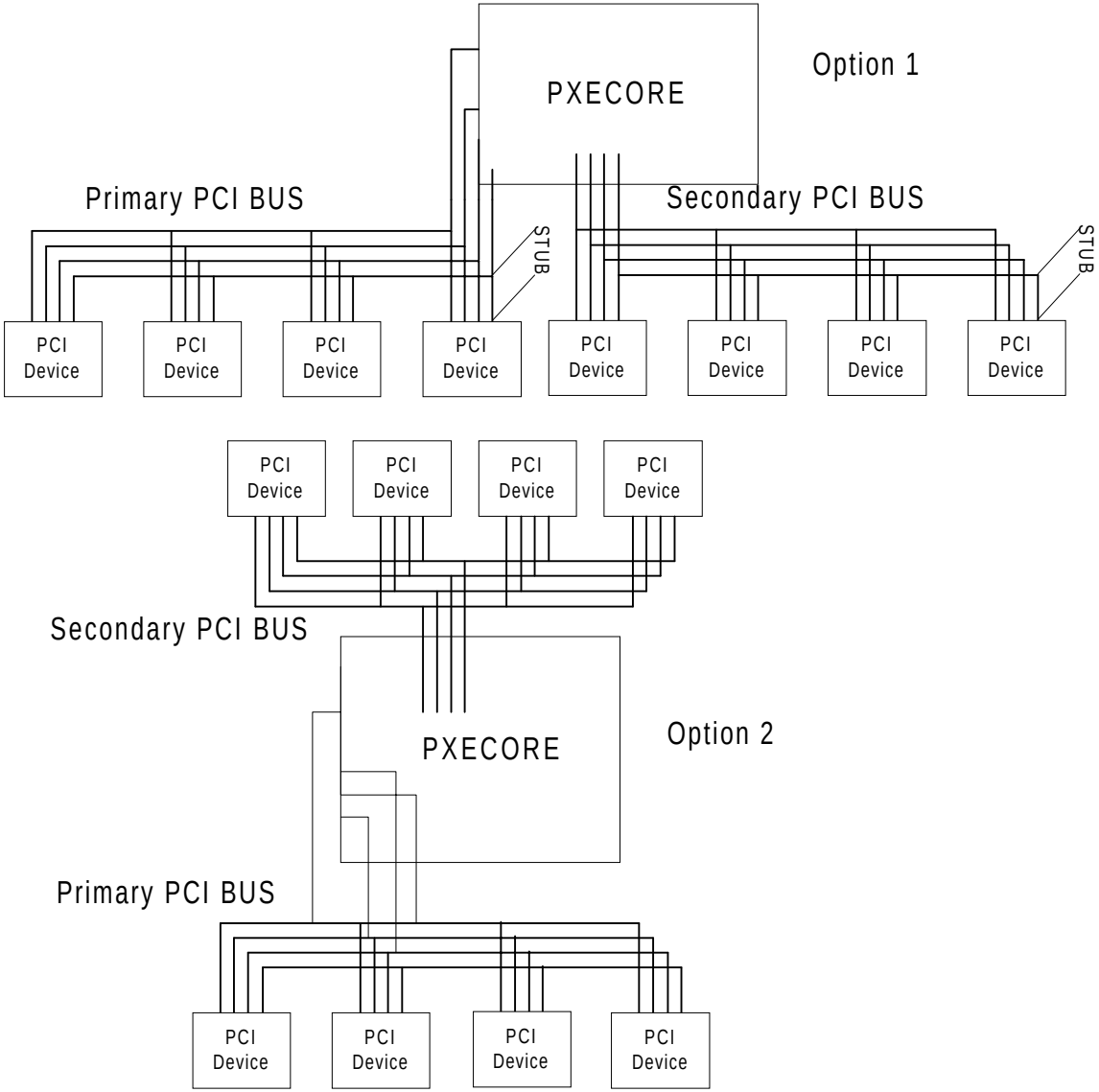


Figure 2 - PXECORE layout options

Above are two layout examples, both of which follow the guidelines:

- Stubs must be kept as short as possible.
- Bus length should be about TBD cm (or less).
- All PCI devices should be placed close to each other.

4.1.5. Miscellaneous PCI signals

4.1.5.1. CORE_IDSEL signal

The IDSEL is needed to access the configuration space of a PCI device.
The PXECORE has an IDSEL signal, connected to the primary PCI IDSEL line. It has an internal 10K pull down. When the PXECORE is responsible for the PCI bus configuration, there is no need for the IDSEL signal, because the CPU has access to the configuration registers as a local bus device. When another PCI master is responsible for the PCI configuration, the IDSEL must be connected.

4.1.5.2. -P_INTA and -P_INTB signals

These are open drain interrupt outputs of the primary PCI. They can be used to interrupt another processor. A 10K pullup is mounted on-board.

4.2. The local bus interface

The local bus interface includes:

- A set of standard buffered i960Rx signals
- Chip select controls
- General purpose I/O lines

Two buffering methods can be used:

1. Quiet - the card's control logic filters out all CPU accesses which do not belong to the target system. This mode reduces the amount of bus traffic and the resulting EMI.
2. Fast - optimized for performance, the buffers are always open.

Quiet /Fast mode is controlled by a bit in the control register.

4.2.1. Signal description

Key:

(M) : PXECORE is the local bus master
(S) : PXECORE is a slave to an external bus master
- : Active low signal
I/O : Input/Output, bi-directional signal
I : Input
O: output
OC, OD: open collector, Open Drain

Table 3. CPU signals

Signal	Dir	Buf	Description
AD[31..0]	I/O	+	CPU muxed Address / Data bus
ADDR[31..2]	O (M) I (S)	+	CPU address lines (latched)
-RDY	I (M) OD (S)	-	Ready. An input when the PXECORE is a bus master. An output when the PXECORE is a slave to an external device. -RDY has an internal pull-up.
W/-R	O (M) I (S)	+	Write / Read
-BLAST	O (M) I (S)	-	Burst last.
-BE[0..3]	O (M) I (S)	+	Byte enable
-ADS	O (M) I (S)	+	Address strobe
HOLD	I	-	Hold request

Signal	Dir	Buf	Description
HOLDA	O	-	Hold acknowledge

Table 4. Interrupts, All lines have an internal pull up.

Signal	Dir	Buf	Description
-INT0..-INT4	I	-	CPU interrupts 0..4.
-INT5	I, OD	-	General purpose interrupt. Used internally as a shared interrupt. The internal interrupt output is open drain. Do not use this line when using the on-board Ethernet controller.
-NMI	I	-	Non-maskable interrupt

Table 5. I/O signals

Signal	Dir	Buf	Description
-CE[0..4]	O	-	Active low chip selects which are programmable.
LED[3..0]	O	-	General purpose outputs, connected to the on-board LED's.
PA_[0..7]	I/O	-	8 bit I/O port. Each bit can be configured as in or out.
PB_[0..7]	I/O	-	8 bit I/O port. Each bit can be configured as in or out.
PC_[0..7]	I/O	-	8 bit I/O port. Each bit can be configured as in or out.
PD_[0..7]	I/O	-	8 bit I/O port. Each bit can be configured as in or out.
PE_[0..7]	I/O	-	8 bit I/O port. Each bit can be configured as in or out.

Note: Ports PA..PE can be used when optional P8 is assembled.

4.2.2. QUIET vs. Fast modes

The external bus interface works in Quiet or Fast mode. In QUIET mode, the card's control logic filters out all CPU accesses which do not belong to the target system, thus greatly reducing the amount of bus traffic and the resulting EMI. External buffers are enabled only if the CPU accesses the external memory address. In FAST mode, the control logic constantly enables the buffers in order to optimize timing. The selection of QUIET or FAST mode is made by a bit in the card's control logic.

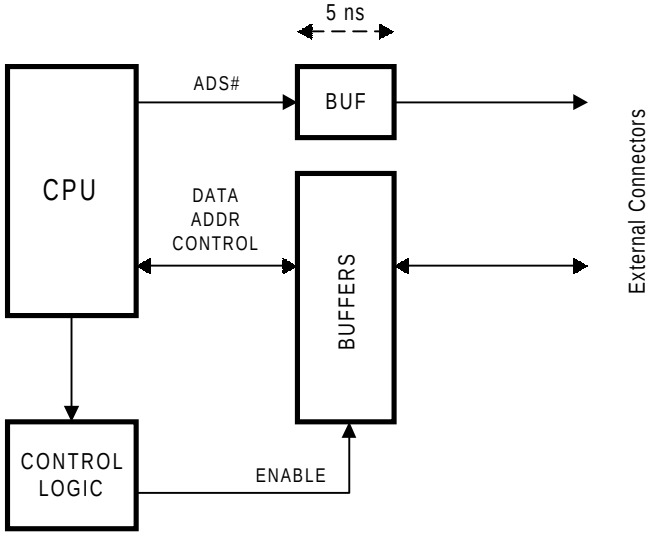


Figure 3 - External Interface Block Diagram

4.2.3. External Interface Timing

External interface timing is closely related to CPU timing. Refer to the i80960H data sheet for CPU timing information. External timing is calculated by combining the CPU timing parameter with the external buffer delay:

For CPU output signals: [CPU clock to output delay] + 5 ns (buffer delay)
For CPU input signals: [CPU input to clock setup time] + 5 ns (buffer delay)

The only exception to this rule is the timing of CPU output signals at the first clock of the bus cycle, in QUIET mode. In this case, timing is determined by the delay of the address decoding logic, which is performed by the PFGA, in order to enable the buffer's output.

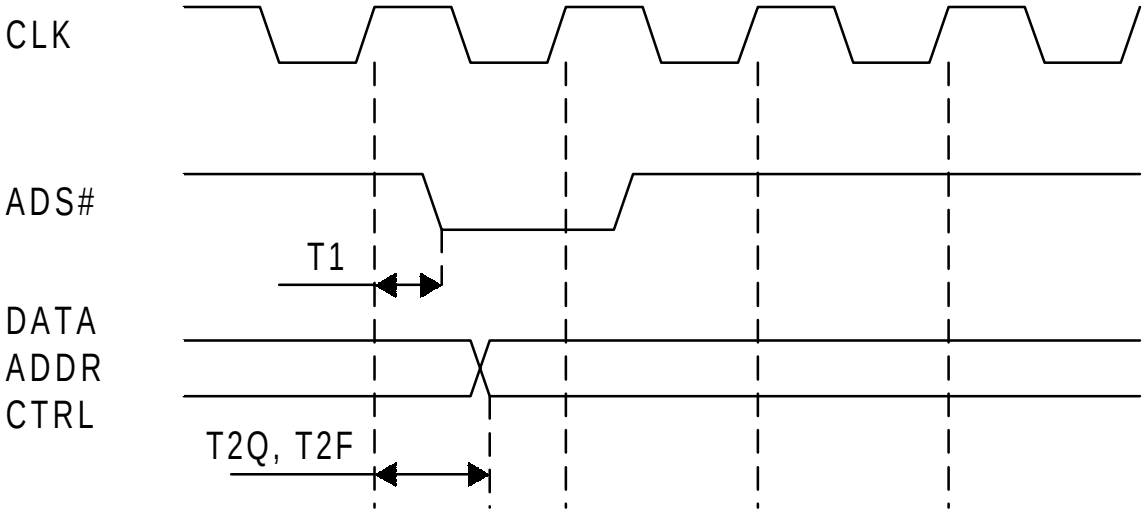


Figure 4 - QUIET and FAST modes' Timing

Table 6. External Interface Timing

Parameter	Max (ns)	Description
T1	18	CLK to ADS# delay
T2Q	32	QUIET mode - CLK to any CPU output, such as ADR[31..2], BE[3..0]#, etc.
T2F	18	FAST mode - CLK to any CPU output, such as ADR[31..2], BE[3..0]#, etc.

4.3. Ethernet signals

The PXECORE supports two methods of Ethernet connection

- 100Base-TX - direct connection to a twisted pair cable. An RJ-45 connector is found on the PXECORE. If the RJ-45 location is not adequate for your mechanical needs, optional connector P7 may lead the TX/RX signal pair to an off-board RJ-45 connector[2].
- MII - if a different physical connection such as T4 or FX is needed, the MII signals found on optional connector P8 may be used.
- P8 also outputs 3 LED indicators: 10/100 speed, link, activity.

[2] - Please note that this method requires controlled impedance layout, and a good understanding of high speed layout techniques.

Table 7. 100Base-TX Ethernet signals

Signal	Dir	Description
Tx+ / Tx-	O	Transmit pair
Rx+ / Rx-	I	Receive pair
SHLD	-	Ground shield

Table 8. MII Ethernet signals[3]

Signal	Dir	Description
RXD[0..3]	I	Receive data nibble
RXCLK	I	25 MHz receive data clock
RXDV	I	Receive data valid
RXER	I	Receive error
CRS	I	Receive carrier sense
COL	I	Collision detected
TXD[0..3]	O	Transmit data nibble
TX_EN	O	Transmit enable
TX_ER	O	Transmit an Error
TX_CLK	I	25 MHz transmit data clock
MDC	O	Management clock
MDIO	I/O	Management data

Table 9. Miscellaneous Ethernet signals

Signal	Dir	Description
SPD100	O	LED output, high when line speed is 100.
-LINKLED	O	LED output, low when there is a link
-ACTLED	O	LED output, low when there is activity on the line

[3] - Refer to IEEE 802.3u, clause 22 for an in-depth description of MII signals.

4.4. General purpose signals

RS-232 signals. These signals may be configured as TTL or RS-232. UART channel B has an on-board connector P1 (See Appendix A - connectors and jumpers).

Table 10. RS-232 signals

Signal	Dir	Buf	Description
UTXD-A	O		Channel A - Transmit data
URXD-A	I		Channel A - Receive data
URTS-A	O		Channel A - Request to send
UCTS-A	I		Channel A - Clear to send
UDTR-A	O		Channel A - Data terminal ready
UDSR-A	I		Channel A - Data set ready
UCD-A	I		Channel A - Carrier detect
URI-A	I		Channel A - Ring indicator
UTXD-B	O		Channel B - Transmit data
URXD-B	I		Channel B - Receive data

Table 11. JTAG signals

Signal	Dir	Description
TCK	I	JTAG signal, has a 10K internal pullup. May be left open.
TMS	I	JTAG signal, has a 10K internal pullup. May be left open.
TDI	I	JTAG signal, has a 10K internal pullup. May be left open.
TRST	I	JTAG signal, may be left open. Has a 10K pullup to the internal active low reset. When reset is active, TRST is low. When reset is not active (high level), TRST is high.
TDO	O	JTAG signal.
-ICELOCK	I	i960Rx proprietary signal. Used for JTAG debuffers.

5. Software Reference

5.1. Memory map

Some of the peripherals in the following table are optional and are board configuration dependant.

Table 12. Memory map

Device	Range	Width	Burst
FLASH memory[4]	FE00 0000 - FEFF FFFF	8 / 32	Yes
DRAM memory	A000 0000 - A0FF FFFF	32	Yes
UART B	E000 0000 - E000 001C	8	No
UART A	E000 0020 - E000 003C	8	No
LED register	E100 0000 -	8	No
Control register	E100 0004 -	8	No
Status register	E100 0008 -	8	No
EE register	E100 000C -	8	No
Interrupt source reg	E100 0018 -	8	No
Revision register	E100 001C -	8	No
I/O ports	E100 0020 - B100 003f	8/16/32	No
Timer	E100 0040 - B100 0044	8	No
CE0 chip select	Programmable to any address in the range of : 0000 4000 - 7FFF FFFF	32	Yes
CE1 chip select		32	Yes
CE2 chip select		32	Yes
CE3 chip select		32	Yes
CE4 chip select		32	Yes

[4] - See paragraph 5.2 for exact mapping of the FLASH memory

5.2. FLASH memory

FLASH memory has the following options:

- 8 or 32 bits wide
- Divided into 2 logical banks: Bank0 and Bank1. Each bank is 4 Mb.
- Can use synchronous (28F016XS) or asynchronous (28F016SA / 28F032SA) devices.
- Can use 2Mx8 (28F016XS/SA) or 4Mx8 (28F032SA) devices.
- Can use Intel's Strata Flash devices.

When using the 8 bit configuration, only Bank1 is in use.

5.2.1. Using the 28F016XS synchronous FLASH

28F016XS performance greatly surpasses that of other asynchronous FLASH devices. This improvement does however require some housekeeping.. (In order to understand the following paragraphs, you should acquaint yourself with the 28F016 data sheet).

At power up, the 28F016XS wakes up in SFI=4. This means that 4 clock cycles will pass before data will be available from the FLASH. The desired SFI is 2, but a program which runs from FLASH cannot program the FLASH. To accommodate the change of SFI, the following steps are taken at boot time:

1. FLASH starts with SFI = 4, CPU region register PMCON15 is initialized to Burst=0, and Control register bit 7 equals 0.
2. Copy the program that changes SFI to DRAM, and execute it.
3. Control register bit 7 is set to 1.
4. Program SFI value to 2, return to calling FLASH routine.

At this point FLASH runs faster, but the CPU cannot burst read from it. The final step is to:

5. Change PMCON14-15 to Burst = 1.

Note: The PXECORE is supplied with sources of MON960 including the above procedure.

5.2.2. Programming the FLASH memory

While *CompuLab* supplies sources and programming examples, you may elect to write your own driver. Please note the following rules.

- When programming a 32 bit Flash memory, the programming words and data **MUST** be 32 bit. You cannot program the upper 16 bit and leave the lower 16 bit intact and vice versa. If, for example, the opcode for "Clear Status Register" is 0x50, then the opcode for 32 bit FLASH would be 0x00500050.
- Always identify the FLASH type before programming: "XS", "016SA", and "032SA" have different ID's .
- Check E7 (Write Protect Jumper) before programming. If it's assembled, you won't be able to program or erase the FLASH.
- SFI timing is related only to the "XS" type of FLASH.
- The value of SFI is not relevant to write operations.

5.3. UART interface

PXECORE basic configuration offers 1 or 2 UART's. The UART's are compatible with 16C1450, except for the RHR register (LSB of baud rate divisor) which is not implemented. This means that the available baud rates are between 115,200 and 600[5]. The baud rate divisor's clock source is 1.8432 MHz.
UART registers are allocated every 4 bytes, i.e. 0,4,8,..
The PXECORE can offer more UART's with the following limitations: UART interface will be TTL level and not RS232 level. The UART lines will use the general purpose I/O lines.

5.4. Fast Ethernet Interface

The PXECORE uses Intel's 82558 as its Ethernet controller. It is connected to the secondary PCI bus.
The 82558 PCI IDSEL pin is connected to the PCI S_AD30 line.
The 82558 interrupt output is connected to INT5.
Ethernet connection has 3 options:

- STP/UTP through RJ45 using J1.
- STP/UTP through P7, when the location of P7 does not suit the mechanical demands.
- MII using P8.

[5] - Please contact *CompuLab* if you need a baud rate lower than 600.

5.5. PXECORE internal registers

5.5.1. LED Register

The PXECORE has a general purpose 4 bit output port. All 4 bits are connected to LED's.

Table 13. LED register

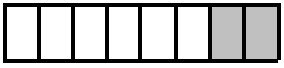
LED register, 0xE1000000

Bit	Default	R/W	Description
0..3	0	R/W	Output only port, also connected to the outside world.
4..7	----	RO	Reserved

5.5.2. Control Register

Table 14. Control register

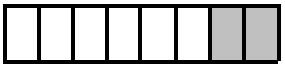
CONTROL register, 0xE1000004



Bit	Default	R/W	Description
0..1	----	RO	Reserved
2	0	R/W	Local Arbiter Priority mode[6]: 0 - Rotating Priority 1 - Fixed
3	0	R/W	Programmable chip select unit write enable[5] 0 - Disable modifications of the chip select unit 1 - Enable modifications of the chip select unit
4	1	R/W	PCI arbiter enable. When 0, the Primary PCI arbiter is disabled and its GNT outputs are in 3-state. When 1, the Primary PCI arbiter is enabled.

[6] - Detailed descriptions of the 'chip select unit' and 'Local Arbiter' is found in the "Embedded Peripherals Reference Guide" document.

CONTROL register, 0xE1000004



Bit	Default	R/W	Description
5	0	R/W	External bus mode. 0 - QUIET 1 - FAST
6	0	R/W	Watch dog. 0 - Disabled 1 - Enabled. After the watch dog is enabled, it can no longer be disabled. This bit must be toggled every second in order to prevent watch dog reset.
7	0	R/W	FLASH SFI mode. 0 - SFI = 4 1 - SFI = 2 Changing this bit does not change the FLASH SFI. It just tells the FLASH controller to which SFI the FLASH is currently programmed.

5.5.2.1. Reset and Watch dog control

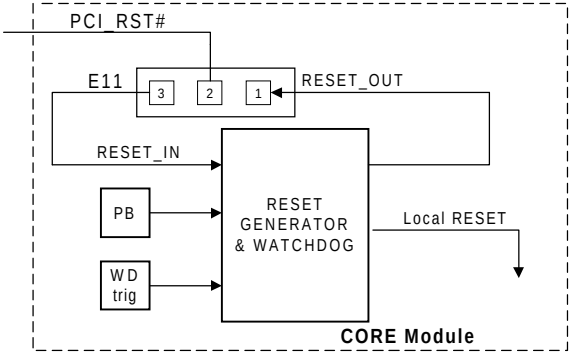


Figure 5 - Reset and watch dog configuration

The PXECORE can be reset from any of the following sources:

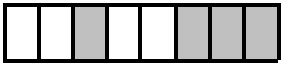
- Power-up reset
- On-board Push button
- Watch dog
- External source (when PCI_RST# is configured as an input)

To activate the watch dog, bit 6 of the CONTROL register should be set to 1. After writing 1 to this register, it should be written with a 1,0 pattern every subsequent second. Failure to do so will activate the watch dog and reset the PXECORE. After being activated, the watch dog cannot be disabled. Only a reset can clear the watch dog assertion.

5.5.3. Status register

Table 15. Status register

Status register, 0xE1000008

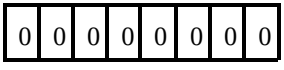


Bit	Default	R/W	Description
0	0	RO	Reserved
1	0	RO	Reserved
2	0	RO	Reserved
3	----	RO	PCI arbiter exists if this bit is set to 1
4	----	RO	FLASH RY/BY output.
5	----	RO	Reserved
6	----	RO	Status of E1 jumper 0 - Jumper inserted 1 - Jumper is out
7	----	RO	Status of E5 jumper 0 - Jumper inserted 1 - Jumper is out

5.5.4. Revision register

Table 16. Revision Register

Revision register, 0xE100001C



Bit	Default	R/W	Description
0..7	0x00	RO	Shows the current revision of the chip

5.5.5. I/O ports

The PXECORE offers up to 5 parallel ports of 8-bits. Each bit can be programmed as input or output.
As seen in the figure on the right, writing “1" to the control register will enable the output buffer, and turn the port into an output port. After reset, the control register is cleared. This guaranties that the ports will always wake up in input mode.

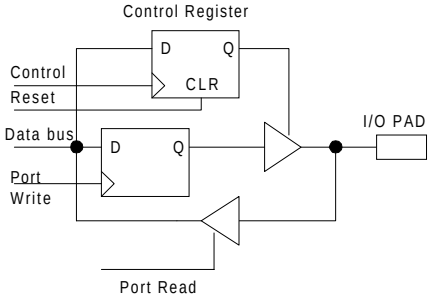


Figure 6 - I/O port internal structure

Ports A/B/C/D can be accessed as 4 x 8 bit ports, as 2 x 16 bit ports (A+B and C+D), or as one 32 bit port. Bit order in the ports is as follows:

Table 17. I/O port bit order

Access mode	D[31:24]	D[23:16]	D[15:8]	D[7:0]
32 bit	PORT D	PORT C	PORT B	PORT A
16 bit - first port			PORT B	PORT A
16 bit - second port			PORT D	PORT C

Table 18. I/O port address map (32 bit access)

32 bit Access mode		
PORT	**R/W Access**	**Control Register**
PORT ABCD	E100 0020	E100 0024

Table 19. I/O port address map (16 bit access)

16 bit Access mode		
PORT	**R/W Access**	**Control Register**
PORT AB	E100 0020	E100 0024
PORT BC	E100 0022	E100 0026

Table 20. I/O port address map (8 bit access)

8 bit access mode		
PORT	**R/W Access**	**Control Register**
PORT A	E100 0020	E100 0024
PORT B	E100 0021	E100 0025
PORT C	E100 0022	E100 0026
PORT D	E100 0023	E100 0027
PORT E	E100 0028	E100 002C

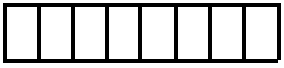
5.5.6. The Timer

The timer has 2 registers: The delay register which holds the desired delay and the count register which counts from 0 to the value of the delay register. When the count register equals the delay register, the Timer interrupt is set and the count register restarts from 0.
Writing a new value to the delay register also restarts the count register. Both registers can be read. A write to the count register will not change its content, but will clear the pending interrupt flag.
The timer clock's tick occurs every 512 micro seconds.

Note: The timer is configuration specific device. It is not included in the basic PXECORE configuration. The i960Rx has 2 internal timers that may be used instead.

Table 21. Timer delay register

Timer delay register, 0xE1000040



Bit	Default	R/W	Description
0..7	0x00	R/W	Timer delay. Provides a timed interrupt in the range of 1953..8 interrupts per second.

Table 22. Timer count register

Timer count register, 0xE1000044

Bit	Default	R/W	Description
0..7	0x00	RO	The count register's current value may be read at any time. Writing any value to this register will clear the pending timer interrupt.

5.5.7. The DMA

The DMA is described in detail in the “Embedded Peripherals Reference Guide”.

5.6. DRAM memory

The PXECORE has DRAM configurations in the range of 4..32MB.
The DRAM is mapped by the i960Rx memory controller. Initialization is done by MON960.
It is recommended to split the DRAM into two logical regions - cache able and non-cache able. This will allow sharing of memory with other PCI bus masters.
DRAM performance is programmed for 1-0-0-0...0 access. PCI bus masters or DMA can burst data in blocks of up to 4 KBytes.

5.7. Interrupt summary

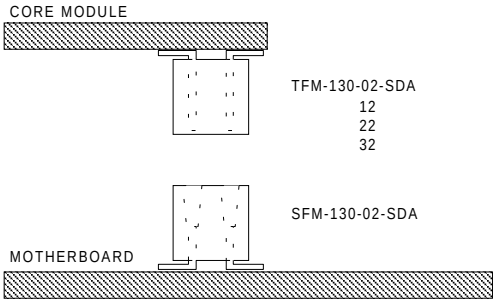
Table 23. Interrupt summary

Interrupt	Function
0	General purpose, can be used as PCI INTA# input
1	General purpose, can be used as PCI INTB# input
2	General purpose, can be used as PCI INTC# input
3	General purpose, can be used as PCI INTD# input
4	General purpose, or 82558 Ethernet controller
5	FPGA interrupt. Can be configured for Timer, DMA or watchdog interrupt
6	UART A interrupt
7	UART B interrupt (UART B interrupt can be connected to the -NMI input)
-NMI	General purpose, or UART B

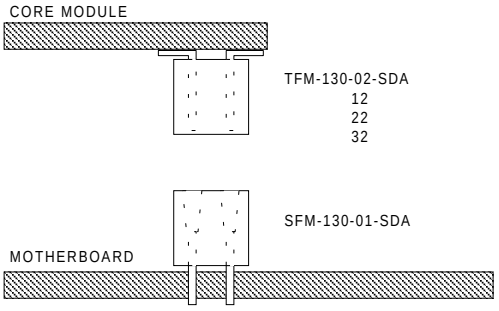
6. Mechanical connection

The PXECORE is supplied with several types of compatible connectors, each addressing a different objective:

- Variable height connectors support 5.97 to 11.43 mm spacing between the motherboard and the CORE module.
- Shrouded, low insertion force connectors for convenient prototyping.
- Through-hole and SMD connectors, according to the customer's assembly facility's specifications.



OPTION #1
SHROUDED CONNECTORS, SMD FEMALE MOTHERBOARD



OPTION #2
SHROUDED CONNECTORS, TH FEMALE MOTHERBOARD

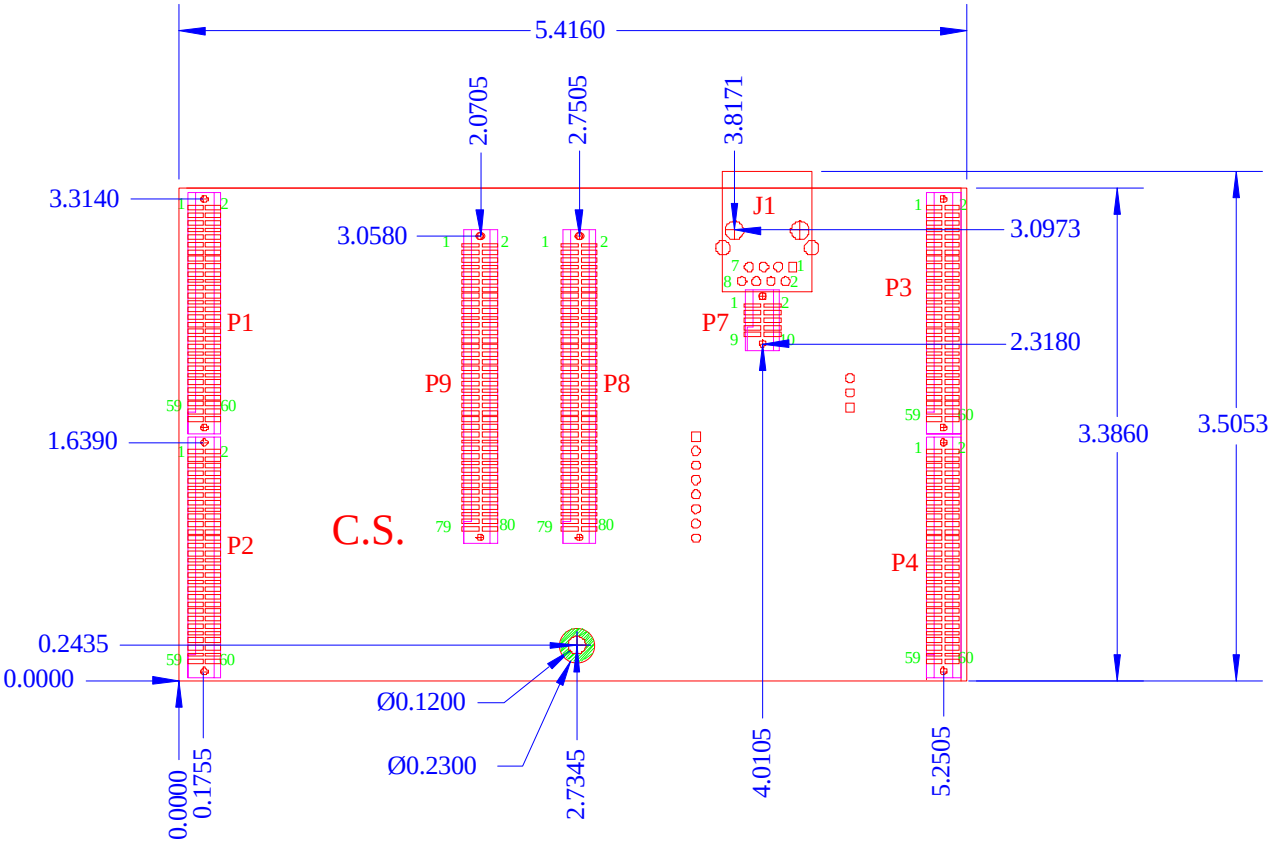
Figure 7 - Mechanical options 1 & 2

Mating motherboard connectors are available from SAMTEC (TH and SMT), and AMP (SMT only). The above part number are Samtec's

Table 24. Mechanical heights

Option	Mated Height (mm)
2	5.97
12	7.75
22	9.53
32	11.43

PXECORE Mechanical Dimensions



CompuLab supplies DXF and DWG files with all required mechanical information.

APPENDIX A - Connectors and Jumpers

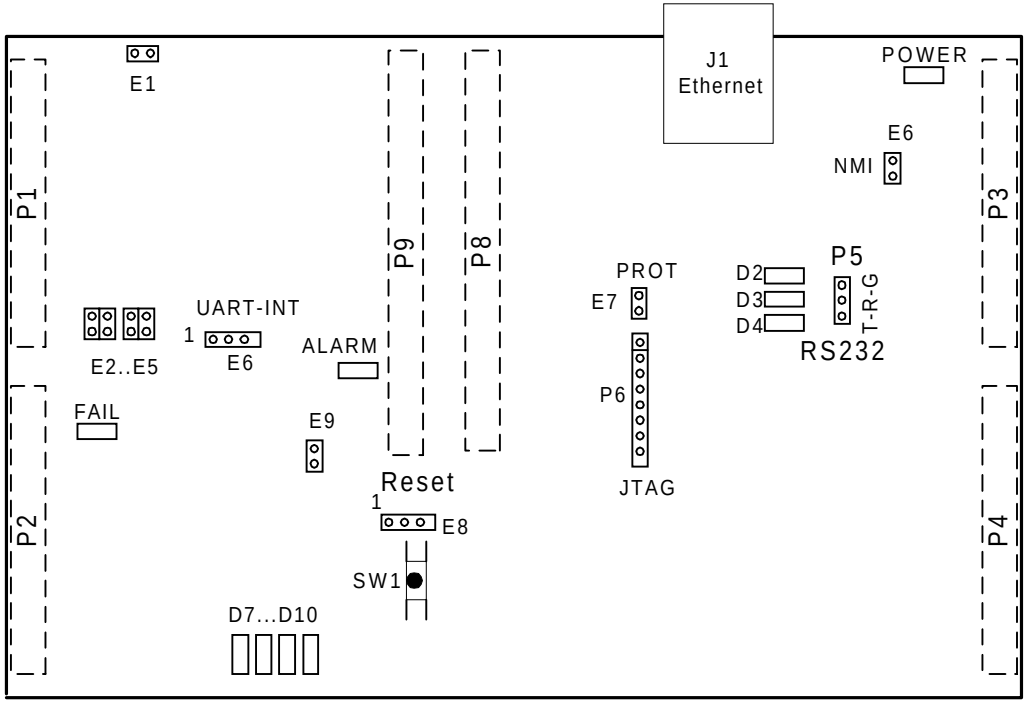


Figure 9 - PXECORE jumpers and LED's

(**Bold** indicates default)

Table 24. Jumper description

Jumper	Description
E1	General purpose. Seen by the status register.
E2	Connected to the CPU WIDTH1(RETRY). Enables CPU boot configuration.
E3	Connected to the CPU D/C# (RstMode). Enables CPU boot configuration.
E4	NMI connector - shortening this jumper will cause an NMI
E5	Float the PCI arbiter outputs when shorted. Seen by the status register.
E6	UART A interrupt destination **1-2** UART interrupt is connected to CPU INT7# 2-3 UART interrupt is connected to CPU NMI#

Jumper	Description
E8	Reset direction. **1-2** Shorted: PCIRESET# (P2-32) is an output. 2-3 Shorted: PCIRESET# is an input, allowing an external device to reset the PXECORE
E7	FLASH protection. When inserted, disables FLASH memory write / erase.
E9	**In** - PCI clock source is external, driven from P2-55 (PCICLKIN) Out - PCI clock source is internal. The Source is an on-board 33 MHz oscillator.

Table 25. LED's

LED	Description
ALARM	Alarm is on at Reset and when the FLASH memory is in slow mode.
POWER	Connected to the 3.3V supply.
FAIL	Connected to the CPU FAIL output.
D7..D10	General purpose. Connected to a 4 bit I/O port.
D2	Ethernet Activity LED
D3	Ethernet speed 10/100 LED
D4	Ethernet link LED

Table 26. P1 and P2 connectors

P1			
Pin	**Name**	**Pin**	**Name**
1	VCC	2	ADDR2
3	ADDR4	4	ADDR3
5	GND	6	ADDR5
7	ADDR7	8	ADDR6
9	GND	10	ADDR8
11	ADDR10	12	ADDR9
13	GND	14	ADDR11
15	ADDR13	16	ADDR12
17	GND	18	ADDR14

P2			
Pin	**Name**	**Pin**	**Name**
1	VCC	2	-PCILOCK
3	-PCIDEVSEL	4	-PCIPERR
5	GND	6	PCIPAR
7	-PCISERR	8	-PCICBE1
9	GND	10	PCIAD15
11	PCIAD14	12	PCIAD13
13	GND	14	PCIAD12
15	PCIAD11	16	PCIAD10
17	GND	18	PCIAD9

P1			
Pin	**Name**	**Pin**	**Name**
19	ADDR16	20	ADDR15
21	GND	22	ADDR17
23	ADDR19	24	ADDR18
25	GND	26	ADDR20
27	ADDR22	28	ADDR21
29	VCC	30	ADDR23
31	PCIAD31	32	PCIAD30
33	GND	34	PCIAD29
35	PCIAD28	36	PCIAD27
37	GND	38	PCIAD26
39	PCIAD25	40	PCIAD24
41	GND	42	-PCICBE3
43	PCIAD23	44	PCIAD22
45	GND	46	PCIAD21
47	PCIAD20	48	PCIAD19
49	GND	50	PCIAD18
51	PCIAD17	52	PCIAD16
53	GND	54	-PCICBE2
55	-PCIFRAME	56	-PCITRDY
57	GND	58	-PCIIRDY
59	VCC	60	-PCISTOP

P2			
Pin	**Name**	**Pin**	**Name**
19	PCIAD8	20	PCIAD7
21	GND	22	PCIAD6
23	PCIAD5	24	PCIAD4
25	GND	26	PCIAD3
27	PCIAD2	28	PCIAD1
29	VCC	30	PCIAD0
31	-PCICBE0	32	-PCIRST
33	GND	34	GND
35	GND	36	PCICLK0
37	GND	38	PCICLK1
39	GND	40	PCICLK2
41	GND	42	PCICLK3
43	GND	44	GND
45	-REQ3	46	-GNT3
47	GND	48	-REQ2
49	-GNT2	50	-REQ1
51	GND	52	-GNT1
53	-REQ0	54	-GNT0
55	PCICLKIN	56	-CORE_REQ
57	GND	58	-CORE_GNT
59	VCC	60	CORE_IDSEL

Table 27. P3 and P4 connectors

P3			
Pin	**Name**	**Pin**	**Name**
1	LED0	2	VCC
3	LED1	4	-CE0
5	LED2	6	GND
7	LED3	8	-CE1
9	-INT0	10	GND
11	-INT1	12	-CE2
13	-INT2	14	GND
15	-INT3	16	-CE3
17	-INT4	18	GND
19	-INT5	20	-CE4
21	-NMI	22	GND
23	UTXD-A	24	
25	URXD-A	26	GND
27	URTS-A	28	
29	UCTS-A	30	VCC
31	UDTR-A	32	-P_INTA
33	UDSR-A	34	GND
35	UCD-A	36	-P_INTB
37	URI-A	38	GND
39	URXD-B	40	TCK
41	UTXD-B	42	GND
43	HOLD	44	TMS
45	HOLDA	46	GND
47	-ADS	48	TDI
49	-BE0	50	GND
51	-BE1	52	TDO
53	-BE2	54	GND
55	-BE3	56	TRST
57	-BLAST	58	GND

P4			
Pin	**Name**	**Pin**	**Name**
1	-RDY	2	VCC
3		4	-ICELOCK
5		6	GND
7	ADDR24	8	ADDR25
9	ADDR26	10	GND
11	ADDR27	12	ADDR28
13	ADDR29	14	GND
15	ADDR30	16	ADDR31
17	DATA0	18	GND
19	DATA2	20	DATA1
21	DATA3	22	GND
23	DATA5	24	DATA4
25	DATA6	26	GND
27	DATA8	28	DATA7
29	DATA9	30	VCC
31	DATA11	32	DATA10
33	DATA12	34	GND
35	DATA14	36	DATA13
37	DATA15	38	GND
39	DATA17	40	DATA16
41	DATA18	42	GND
43	DATA20	44	DATA19
45	DATA21	46	GND
47	DATA23	48	DATA22
49	DATA24	50	GND
51	DATA26	52	DATA25
53	DATA27	54	GND
55	DATA29	56	DATA28
57	DATA30	58	GND

P3			
Pin	**Name**	**Pin**	**Name**
59	W/-R	60	VCC

P4			
Pin	**Name**	**Pin**	**Name**
59	DATA31	60	VCC

Table 28. P8 MII and I/O Connector

P8			
Pin	**Name**	**Pin**	**Name**
1	GND	2	GND
3	RXD0	4	RXD1
5	RXD2	6	RXD3
7	RXCLK	8	RXDV
9	GND	10	RXER
11	COL	12	CRS
13	TXEN	14	GND
15	TXD0	16	TXD1
17	TXD2	18	TXD3
19	TXCLK	20	MDIO
21	GND	22	MDC
23	SPD100	24	-LINKLED
25	-ACTLED	26	GND
27	PA_0	28	PA_1
29	PA_2	30	PA_3
31	PA_4	32	PA_5
33	PA_6	34	PA_7
35	GND	36	GND
37	PB_0	38	PB_1
39	PB_2	40	PB_3

P8			
Pin	**Name**	**Pin**	**Name**
41	PB_4	42	PB_5
43	PB_6	44	PB_7
45	PC_0	46	PC_1
47	PC_2	48	PC_3
49	PC_4	50	PC_5
51	PC_6	52	PC_7
53	GND	54	GND
55	PD_0	56	PD_1
57	PD_2	58	PD_3
59	PD_4	60	PD_5
61	PD_6	62	PD_7
63	PE_0	64	PE_1
65	PE_2	66	PE_3
67	PE_4	68	PE_5
69	PE_6	70	PE_7
71	GND	72	GND
73	----	74	----
75	----	76	----
77	----	78	----
79	GND	80	GND

Table 29. P9 Secondary PCI Connector

P9			
Pin	**Name**	**Pin**	**Name**
1	GND	2	GND
3	+3.3V	4	+3.3V
5	S_AD0	6	S_AD1
7	S_AD2	8	S_AD3
9	S_AD4	10	S_AD5
11	S_AD6	12	S_AD7
13	GND	14	-S_CBE0
15	S_AD8	16	S_AD9
17	S_AD10	18	S_AD11
19	S_AD12	20	S_AD13
21	S_AD14	22	S_AD15
23	-S_CBE1	24	GND
25	S_AD16	26	S_AD17
27	S_AD18	28	S_AD19
29	S_AD20	30	S_AD21
31	S_AD22	32	S_AD23
33	GND	34	-S_CBE2
35	S_AD24	36	S_AD25
37	S_AD26	38	S_AD27
39	S_AD28	40	S_AD29

P9			
Pin	**Name**	**Pin**	**Name**
41	S_AD30	42	S_AD31
43	-S_CBE3	44	GND
45	+3.3V	46	+3.3V
47	GND	48	GND
49	S_PCICLK2	50	S_PCICLK1
51	S_PCICLK4	52	S_PCICLK3
53	GND	54	GND
55	-S_DEVSEL	56	-S_FRAME
57	-S_IRDY	58	-S_TRDY
59	-S_STOP	60	-S_LOCK
61	-S_PAR	62	-S_PERR
63	GND	64	-S_SERR
65	-S_REQ1	66	-S_GNT1
67	-S_REQ2	68	-S_GNT2
69	-S_REQ3	70	-S_GNT3
71	-S_REQ4	72	-S_GNT4
73	----	74	----
75	+3.3V	76	+3.3V
77	-S_RST	78	+3.3V
79	GND	80	GND

Table 30. P5 UART connector

P5	Description
T	Transmit data
R	Receive data
G	GND

Table 31. P7 JTAG connector

P7	Description
1	TCK
2	TMS
3	TDI
4	TRST
5	TDO
6	-ICELOCK (non-standard JTAG signal, used in debug)
7	VCC (5V)
8	GND

APPENDIX B - The PCREVA Evaluation Board

The PCREVA is targeted for evaluation of the PXECORE in a PCI-PC environment. *CompuLab* supplies software drivers for WIN95/WIN98 that enable communication with the PXECORE. We recommend the use of KRF-TECH's WIN-Driver for easy development of device drivers. (http://www.krf-tech.com).

B.1. Jumper Configuration

Before plugging the PXECORE (assembled on PCREVA) into the PC, it should be configured as follows:

B.1.1. Jumpers to assemble:

- E5, Disable the on-board PCI arbiter (make it float).
- E8 2-3, Make the -PCIRST signal an input.
- E9, PCI clock source is external.

B.2. Connectors

B.2.1. PXECORE connectors

The PXECORE P1..P4 connects directly to the PCREVA P1..P4.
PCREVA does not support P8 and P9 of the PXECORE.

B.2.2. RS232 Connectors

The PCREVA has two male DB-9's for RS-232 communication with the PXECORE.

Table 32. Connection of RS-232 to the PCREVA

PXECORE pin	Signal name	PCREVA connector & pin
UART A		
--	GND	P2-5
P3-23	UTXD-A	P2-3
P3-25	URXD-A	P2-2
P3-27	URTS-A	P2-7
P3-29	UCTS-A	P2-8
P3-31	UDTR-A	P2-4
P3-33	UDSR-A	P2-6
P3-35	UCD-A	P2-1
P3-37	URI-A	P2-9
UART B		
--	GND	P7-5

PXECORE pin	Signal name	PCREVA connector & pin
P3-39	URXD-B	P7-2
P3-41	UTXD-B	P7-3

B.2.3. JTAG Connector

The PCREVA has two options for routing the PXECORE JTAG signals. The first is a P5 JTAG connector (which is compatible with the ALTERA Bit/Byte blaster). The second option is to route the JTAG signals to the PCI edge connector. Jumper block E1..E5 switches between these two options. Position 1-2 in the jumper block directs the JTAG signal to P5, while position 2-3 directs it to the PCI edge connector.

Table 33. PCREVA JTAG connection

PXECORE pin	Signal name	Jumper	P5	PCI Edge Connector
48	TDI	E1	9	A4
52	TDO	E2	3	B4
40	TCK	E3	1	B2
44	TMS	E4	5	A3
56	TRST	E5	7	A1
--	GND	--	2	--
--	GND	--	10	--
--	VCC	--	4	--

B.2.4. External power connector

Table 34. P8 Power connector

P8 - Pin	Signal Name
1	+12V
2	GND
3	GND
4	VCC

B.3. The NMI switch

The PCREVA has a pushbutton tactile switch SW1, which is connected to the PXECORE -NMI input (P3-21). Pressing this switch will generate an NMI (Non Maskable Interrupt).

B.4. Local bus headers

The PCREVA provides six headers with most of the local bus buffered signals. These may be connected to a logic analyser or to a prototype board. Headers have two rows of 10 pins with 100 mil spacing.

Table 35. PCREVA Local bus headers

P11					P14			
PIN	**SIGNAL**	**PIN**	**SIGNAL**		**PIN**	**SIGNAL**	**PIN**	**SIGNAL**
10	GND	20	GND		10	GND	20	GND
11		21			11		21	
12		22			12	CPUA2	22	CPUA3
13	CPUA4	23	CPUA5		13	CPUA6	23	CPUA7
14	CPUA8	24	CPUA9		14	CPUA10	24	CPUA11
15	CPUA12	25	CPUA13		15	CPUA14	25	CPUA15
16	CPUA16	26	CPUA17		16	CPUA18	26	CPUA19
17	CPUA20	27	CPUA21		17	CPUA22	27	CPUA23
18	LED0	28	LED1		18	LED2	28	LED3
19	GND	29	GND		19	GND	29	GND

P12

PIN	SIGNAL	PIN	SIGNAL
10	GND	20	GND
11	-CE0	21	-CE1
12		22	
13	HOLD	23	
14	-BE0	24	-BE1
15		25	
16	RLWH	26	-RDY
17	CPUA24	27	CPUA25
18	CPUA28	28	CPUA29
19	GND	29	GND

P13

PIN	SIGNAL	PIN	SIGNAL
10	GND	20	GND
11	-CE2	21	-CE3
12	-NMI	22	-CE4
13	HOLDA	23	-ADS
14	-BE2	24	-BE3
15		25	-BLAST
16	-BTERM	26	CPUCLK
17	CPUA26	27	CPUA27
18	CPUA30	28	CPUA31
19	GND	29	GND

P10

PIN	SIGNAL	PIN	SIGNAL
10	GND	20	GND
11	CPUD0	21	CPUD1
12	CPUD4	22	CPUD5
13	CPUD8	23	CPUD9
14	CPUD12	24	CPUD13
15	CPUD16	25	CPUD17
16	CPUD20	26	CPUD21
17	CPUD24	27	CPUD25
18	CPUD28	28	CPUD29
19	GND	29	GND

P9

PIN	SIGNAL	PIN	SIGNAL
10	GND	20	GND
11	CPUD2	21	CPUD3
12	CPUD6	22	CPUD7
13	CPUD10	23	CPUD11
14	CPUD14	24	CPUD15
15	CPUD18	25	CPUD19
16	CPUD22	26	CPUD23
17	CPUD26	27	CPUD27
18	CPUD30	28	CPUD31
19	GND	29	GND

B.5. The PCREVA schematic

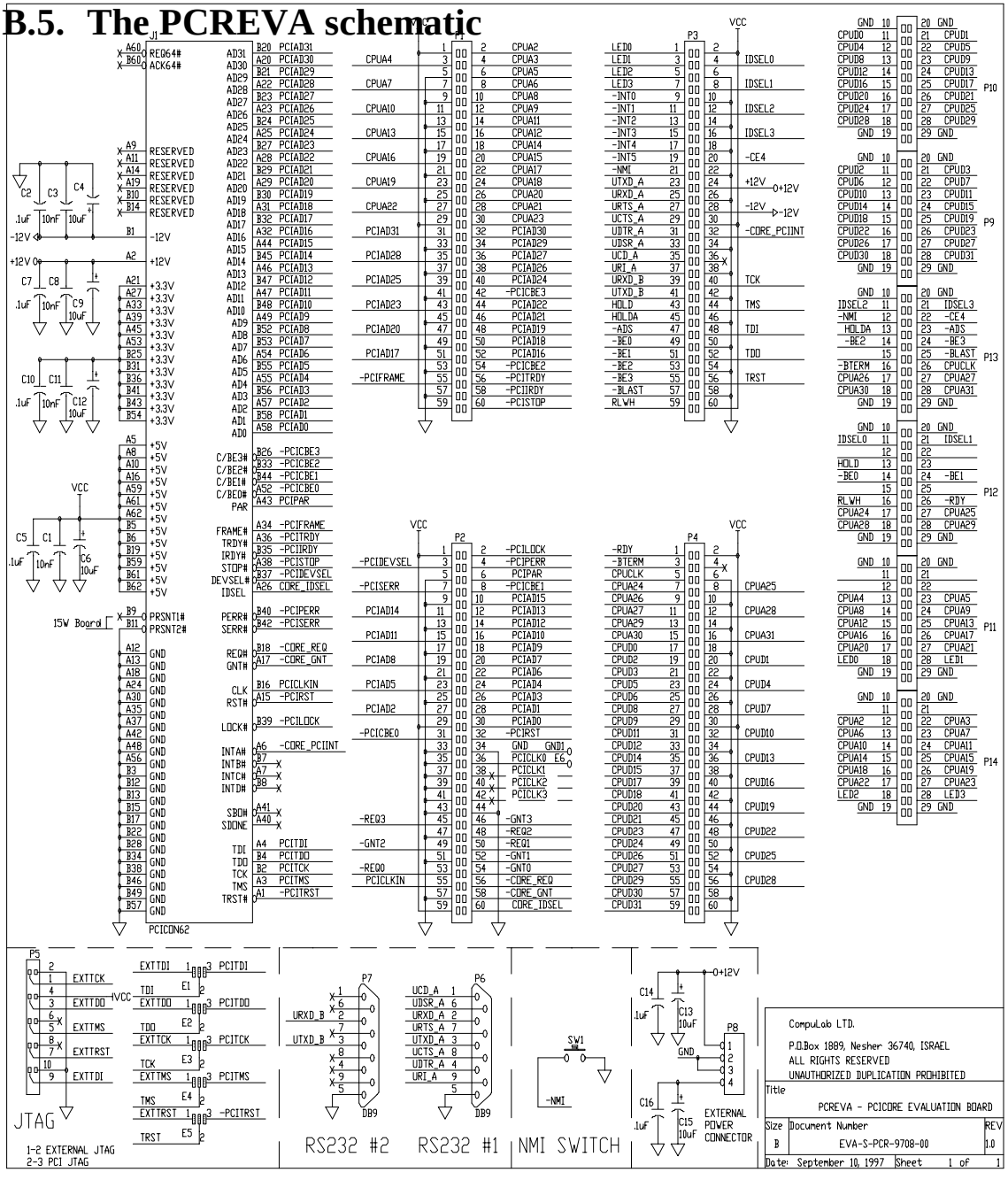


Figure 10- The PCREVA schematic