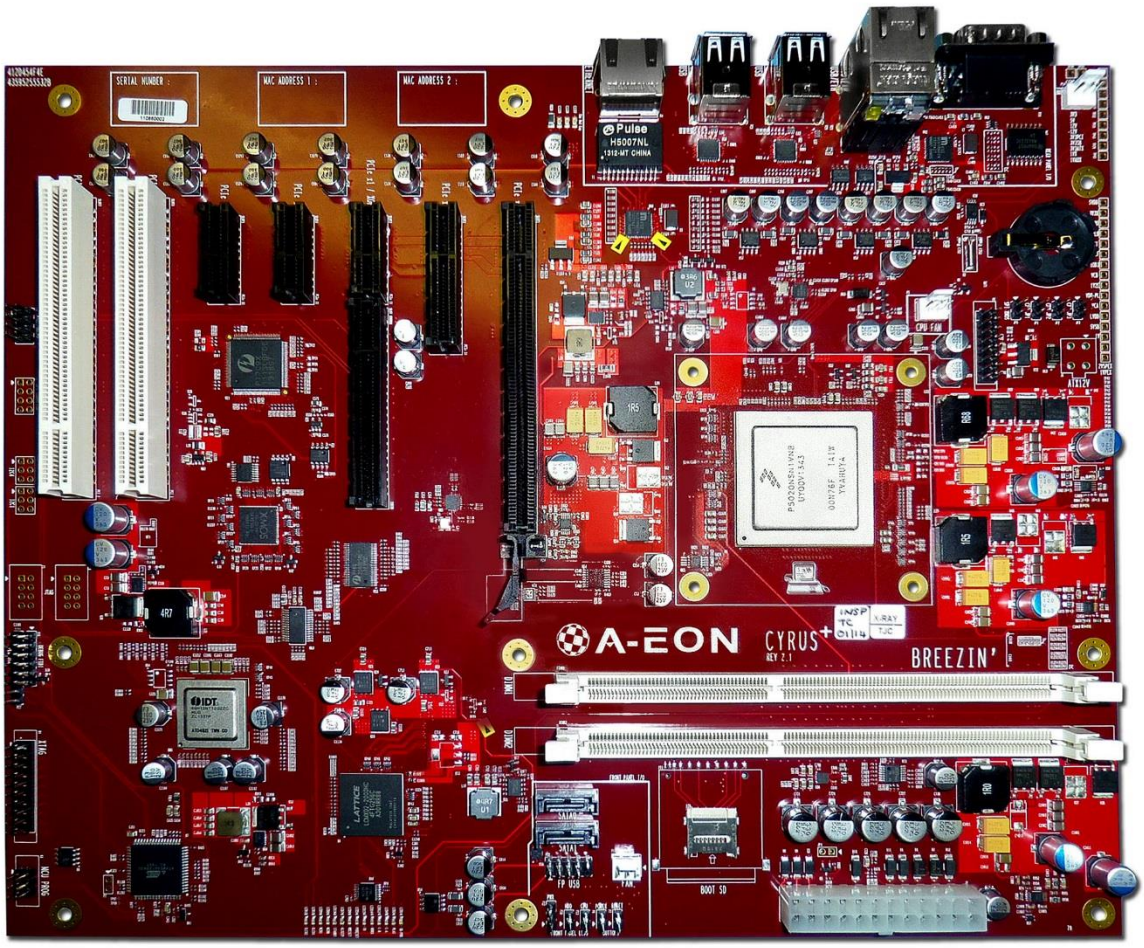

Revison 2.1 Cyrus+ Motherboard

CYRUS PLUS MOTHERBOARD TECHNICAL REFERENCE MANUAL VERSION 1.1.1 AMIGAONE X5000

Issue	Author	Approved	Date
1	James Felix	Marcin Jankowski	16/05/2014
1.1	James Felix	Marcin Jankowski	10/06/2014
1.1.1	Lyle Hazlewood	R.T.Dickinson	16/07/2014

1 CONTENTS

FIGURES

TABLES

2 INTRODUCTION

The Cyrus Plus motherboard combines a high performance Freescale QorIQ CPU with powerful and flexible I/O features to deliver the ultimate desktop platform for AmigaOS users.

This manual contains hardware and software reference information to assist with installation, configuration and low level programming of Cyrus Plus.

2.1 TECHNICAL SUPPORT

For technical support, please contact your reseller.

2.2 ABBREVIATIONS

Acronym	Description
PCIe	PCI Express
PSU	Power Supply Unit
CPU	Central Processing Unit
Hot-Plug	Remove or insert connection/cable whilst power is on
RTC	Real time clock
OD	Open Drain
PU	Pulled-Up
PD	Pulled-Down
RO	Read only
RW	Read write
BCD	Binary-coded decimal
ACPI	Advanced Configuration and Power Interface
SD	Secure Digital
ASCII	American Standard Code for Information Interchange

3 ARCHITECTURE

Cyrus Plus's architecture is shown in Figure 1 below:

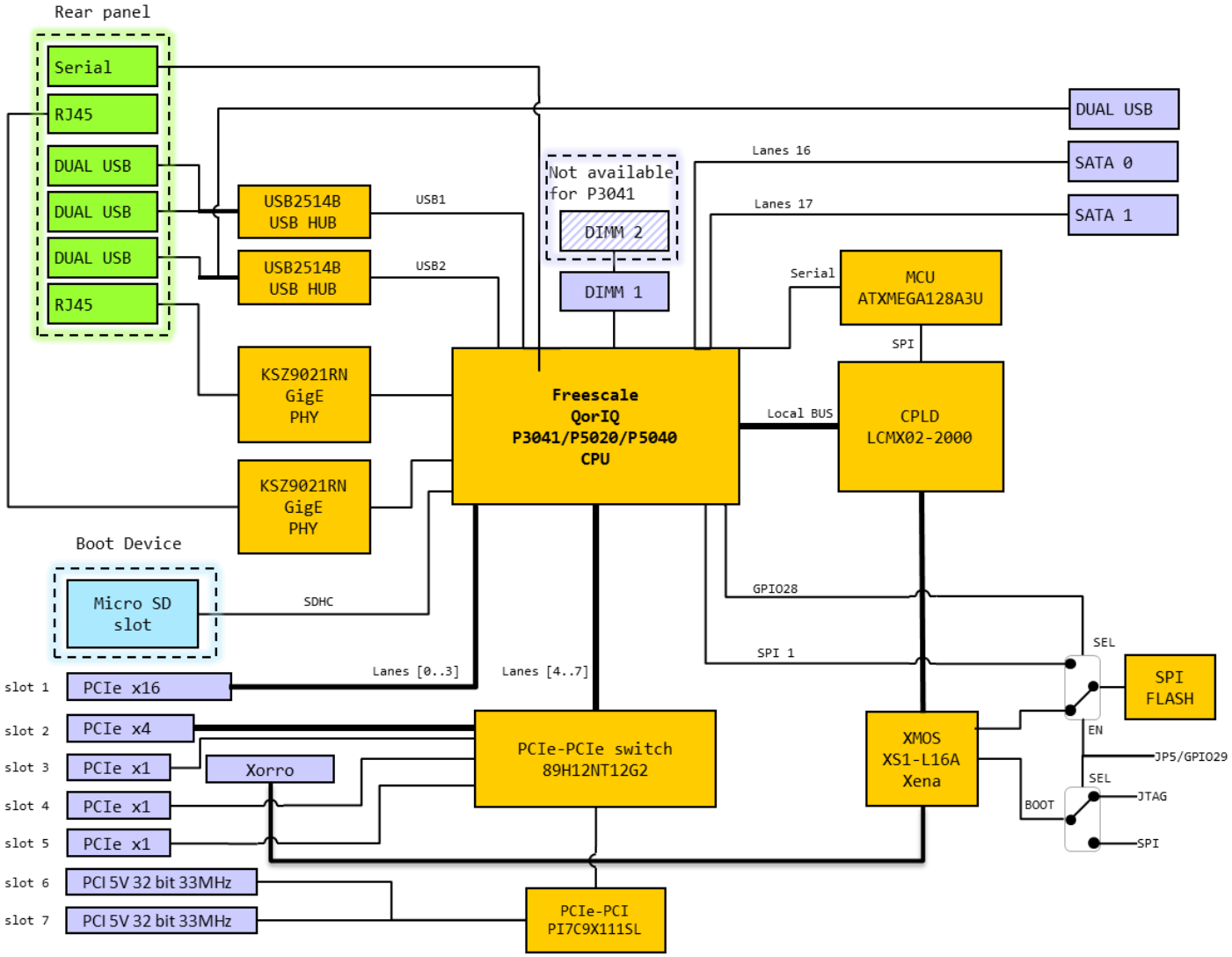

Figure 1: Cyrus Plus Block Diagram

3.1 CPU

The CPU on Cyrus Plus is a Freescale QorIQ Power Architecture P series processor. There are 3 options of CPU, these are the P3041, P5020 and P5040.

3.1.1 P3041

This CPU combines four 1.5 GHz 32-bit e500mc core with a 128KB L2 cache, a single DDR3 memory controller (1333MT/s) and 18 SerDes channels.

The Power Architecture e500mc cores adhere to most of the Power ISA v2.06 for more information on the e500mc check the Freescale website.

3.1.2 P5020

This CPU combines two 2 GHz 64-bit e5500 core with a 512KB L2 cache, dual DDR3 memory controller (1333MT/s) and 18 SerDes channels.

The Power Architecture e5500 cores adhere to most of the Power ISA v2.06 for more information on the e5500 check the Freescale website.

3.1.3 P5040

This CPU combines four 2.2 GHz 64-bit e5500 core with a 512KB L2 cache, dual DDR3 memory controller (1600MT/s) and 20 SerDes channels.

The Power Architecture e5500 cores adhere to most of the Power ISA v2.06 for more information on the e5500 check the Freescale website.

3.2 MAIN MEMORY

The P5020 and P5040 variants have two memory controllers, the P3041 variant has one memory controller. Each CPU memory controller is connected to a standard DDR3 DIMM slot.

For further details, see section 5.

3.3 ETHERNET PHY

The two Micrel KSZ9021RN Gigabit Ethernet PHYs use the RGMII protocol.

The PHYs adaptors use two LEDs to indicate the link speed connection as shown in Table 1. The on LED blinks when there is activity on the port.

Speed	LED1 (right)	LED2 (left)
1000 link	Off	On
100 link	On	Off
10 link	On	On

Table 1: Ethernet link speed

3.4 XENA

An XMOS XS1-L16A-128 “Software Defined Silicon” (SDS) device is provided to support simple, high performance I/O.

3.5 CPLD

The CPLD provides glue logic and control registers. It also provides a fast mailbox and data interface between the CPU and the XENA device.

For further details on the CPLD, see section 7.1 and for the XENA see section 6.5.

3.6 BOOT SD CARD

The Cyrus Plus motherboard is booted from a micro SD card. This needs to be fitted in P29 for the Cyrus Plus motherboard to boot with a valid BIOS in the first 1258 blocks. For more information see section 9.

3.7 BIOS

A micro SD card provided will hold BIOS code.

The BIOS code is maintained by Hyperion.

4 CPU

This section provides programmer visible details of CPU hardware implementation.

4.1 SERDES LANES

The SerDes lanes are connected as shown in Table 2 below:

Lane	Connection
0	PCIe slot 1 lane 0
1	PCIe slot 1 lane 1
2	PCIe slot 1 lane 2
3	PCIe slot 1 lane 3
4	PCIe switch lane 0
5	PCIe switch lane 1
6	PCIe switch lane 2
7	PCIe switch lane 3
8	Not used.
9	Not used.
10	Not used.
11	Not used.
12	Not used.
13	Not used.
14	Not used.
15	Not used.
16	SATA 0
17	SATA 1
18	Not used
19	Not used

Table 2: CPU SerDes Lane Assignments

Notes:
Lane 18-19 are only available on the P5040.
Unused ports are left un-connected.

4.2 UARTS

The CPU provides two UARTs, one for external RS232 communication and one for MCU supervisor interface.

The UART 0 signals are available on a DB9 connector, P16. This is a 5 wire RS232 interface with RTS and CTS, **U-Boot does not used hardware flow control.**

UART 1 is connected to the MCU to get temperature and voltage readings. For further details on the supervisor interface see section 8.1.

4.3 GPIOS

The CPU provides 32 general purpose I/Os (GPIOs) and 14 are used. For details of how these are wired, see Table 3 below.

GPIO line	Signal Name	Direction	Connection	Notes
GPIO0	Jumper 1	in	CPU_GPIO0	0 = fitted, PU
GPIO1	Jumper 2	in	CPU_GPIO1	0 = fitted, PU
GPIO2	HARD_RESET#	out	CPU_MCU_GPIO0	OD
GPIO3	POWER_OFF#	out	CPU_MCU_GPIO1	OD
GPIO4	CPU_LED	out	CPU LED	PU
GPIO5	HDD_LED	out	HDD ACTIVITY LED	PU
GPIO6	CPLD GPIO 0	in	CPU_CPLD_GPIO0	PU
GPIO7	CPLD GPIO 1	in	CPU_CPLD_GPIO1	PU
GPIO18	CPLD GPIO 2	in	CPU_CPLD_GPIO2	PU
GPIO19	VGA_ BIOS_EN#	in	CPU_GPIO9	0 = fitted, PU
GPIO20	-	in	CPU_GPIO10	PU
GPIO27	RTC_MFP	in	None	PU
GPIO28	EN_XMOS_SPI#	in	EN_XSPI_n	PU 0 = fitted
GPIO29	Program XMOS SPI	out	PROG_XSPI	PD

Table 3: CPU GPIOs

Notes:
A '#' suffix denotes an active-low signal.

4.4 EXTERNAL INTERRUPTS

P5020 External Interrupt	Connection	Setup
IRQ0#	unused	interrupt input, level sensitive, active low
IRQ1#	Ethernet PHY #2	interrupt input, level sensitive, active low
IRQ2#	XMOS	interrupt input, level sensitive, active low
IRQ3#	Ethernet PHY #1	interrupt input, level sensitive, active low
IRQ4#	CPLD	interrupt input, level sensitive, active low
IRQ5#	MCU	interrupt input, level sensitive, active low
IRQ6#	unused	interrupt input, level sensitive, active low
IRQ7#	unused	interrupt input, level sensitive, active low
IRQ8#	unused	interrupt input, level sensitive, active low
IRQ9#	unused	interrupt input, level sensitive, active low
IRQ10	Jumper 5, JP5	GPIO, input, PU
IRQ11	XMOS SPI data steering mux	GPIO, output, PD

Table 4: CPU External interrupts

4.5 I^2C CONTROLLER

The CPU has 4 I^2C controllers the use of each controller is shown below in Table 5.

I^2C BUS	Devices attached
1	NXID and MAC address
2	DDR3 DIMMS
3	SMBUS for PCIe devices
4	RTC

Table 5: I^2C CPU List

4.6 SERIAL TERMINAL

For serial communications, on a PC it is recommended to use TeraTerm. The serial port control must be configured as follows:

- 115200 Baud
- 8 bit data
- No Parity
- 1 Stop bit
- No Flow Control

5 DDR3 DIMMS

Cyrus Plus uses standard 1.5V DDR3 DIMMs, 1 socket for P3041 and 2 sockets for P5020/P5040.

The board has been qualified with unbuffered non-ECC DIMMs. For the latest information on recommended DIMM module types, please contact your reseller.

It is recommended to use DIMMs in matched pairs.

5.1 SIZE

The total physical maximum size of memory that the memory controllers can address is 64GB, however the practical memory size limit will depend on software.

5.2 SPEED

The maximum speed supported by the memory controllers is DDR3-1333 for the P3041 and P5020 variants and DDR3-1600 for the P5040 variant. Faster memory may be fitted but this speed limit will apply.

5.3 SERIAL PRESENCE DETECT

The Serial Presence Detect (SPD) addresses of the 2 DIMM sockets is are as follows:

Socket	SA2	SA1	SA0	Address
DIMM1	0	0	1	0x51
DIMM2	0	1	0	0x52

Table 6: SIMM SPD Addresses

6 XMOS SUBSYSTEM

Cyrus Plus includes direct support for XMOS “SDS” (Software Defined Silicon) technology. A dedicated XMOS device, designated “Xena” here, is provided on-board.

Xena is connected to both the main CPU and a custom expansion slot (“Xorro”), which is mechanically aligned with a conventional PCI Express x1 slot.

Xorro expansion cards may be enhanced by adding connectivity to the PCI Express bus. Alternatively, if the Xorro slot is not required, the PCI Express slot connector can be used for conventional PCI Express x1 add-in cards.

This section provides essential details of Cyrus Plus’s XMOS subsystem, and should be read in conjunction with relevant XMOS documentation.

6.1 BLOCK DIAGRAM

Figure 2 shows how Xena is connected to the main processor, the CPLD, the XTAG debug header and the Xorro slot.

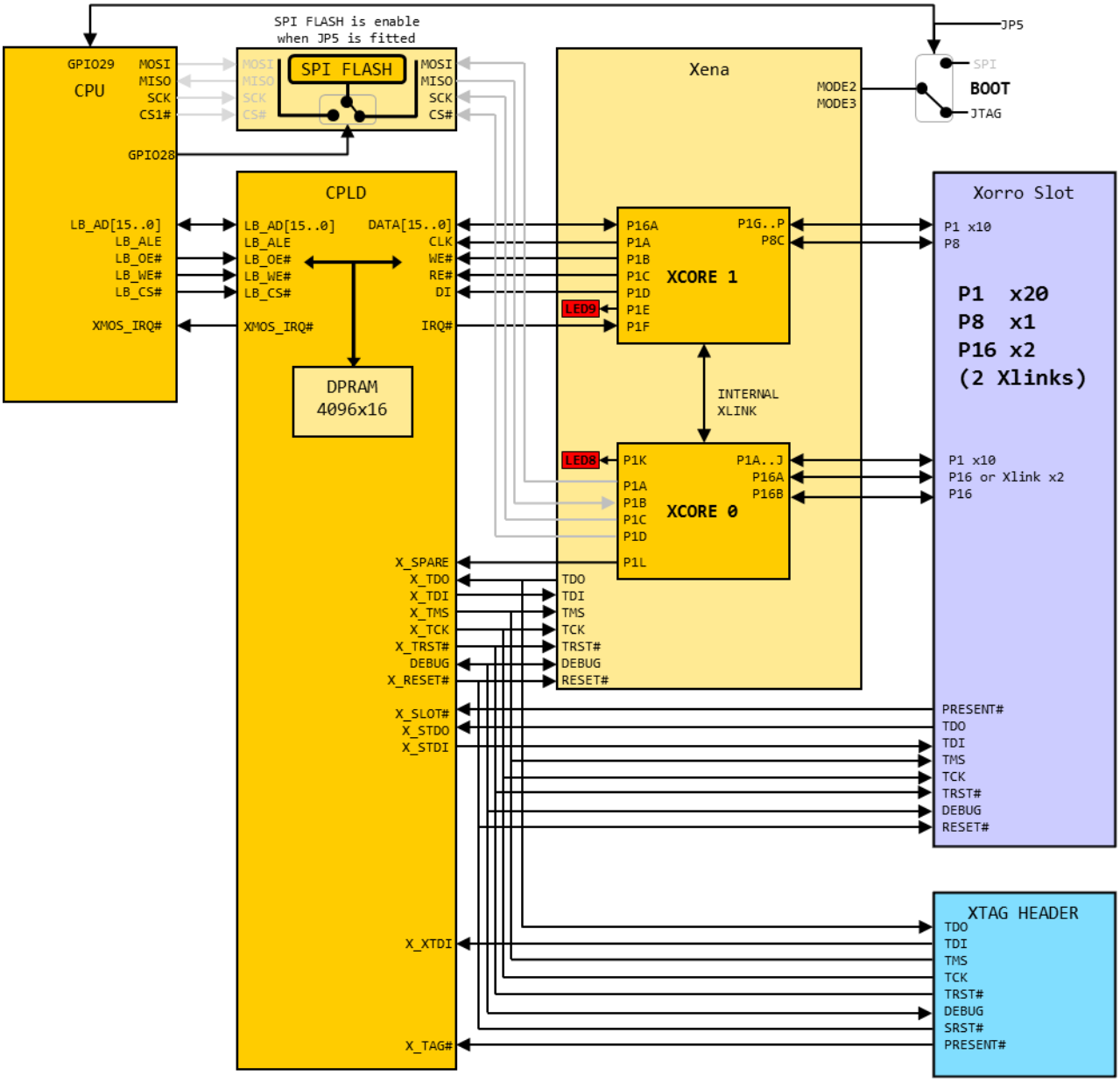

Figure 2: XMOS Subsystem Block Diagram

6.2 XENA CONNECTORS

The XENA connectors are displayed below in Figure 3.

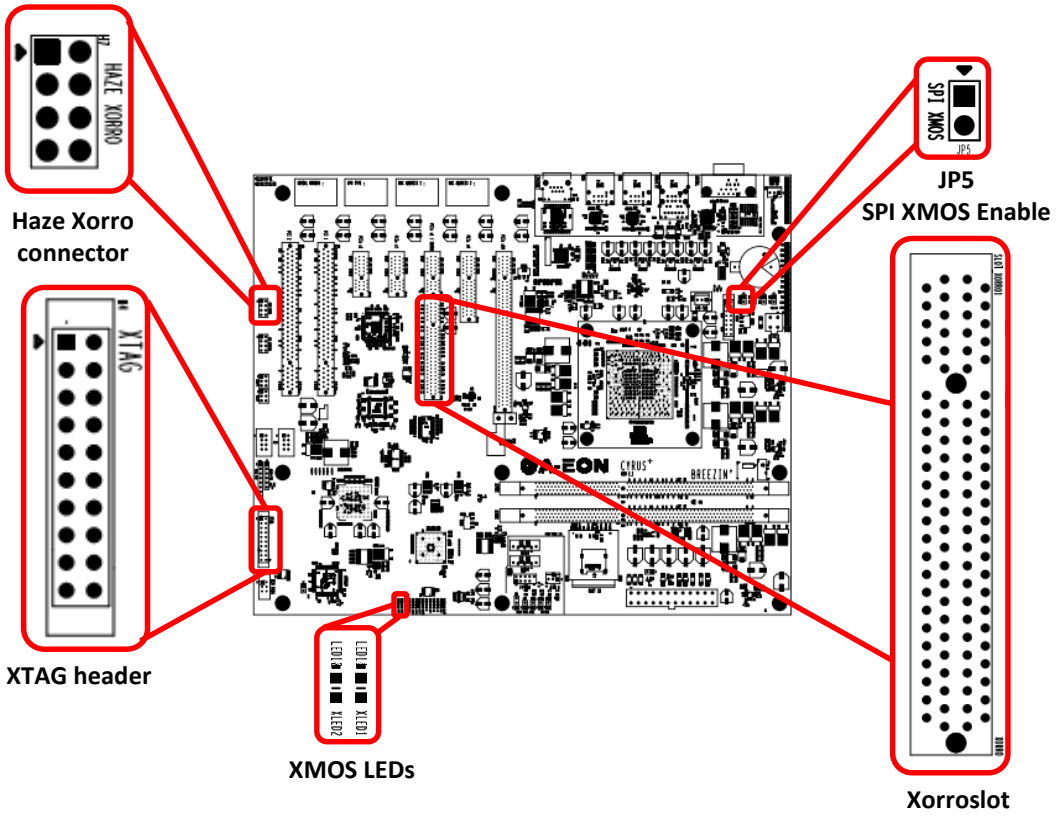

Figure 3: XENA connector positions

6.3 XMOS DEVICE TYPE AND CLOCKING

Xena is a 500MHz, dual-core XS1-L16A (formerly XS1-L2), in a 124-pin QFN package. It is clocked from a 25MHz oscillator, and its PLL is configured for x20 operation i.e. a core clock speed of 500MHz.

6.4 BOOTSTRAPPING AND DEBUG

Bootstrapping and debug of Xena is accomplished via its reset, JTAG and debug signals. These are connected to the CPLD, and may be controlled and sampled via the CPU local bus registers that are implemented within it. For details of these, see section 7.

Normally, software running on the main CPU will only see the Xena chip on this interface. If a Xorro card is fitted in the slot, and this asserts the slot PRESENT# signal, the CPLD will route the JTAG chain through the Xorro card, so that any devices on it will appear before Xena (Xorro's TDO connects to Xena's TDI).

Xena's control and debug signals are also connected to a header to allow the use of an XMOS XTAG debugger. If one is connected, the CPLD will float most of its pins, allowing the XTAG to take over. It will, however, still provide automatic routing of the TDI/TDO signal chain through a Xorro card, if required, so that the XTAG can control both Xena and Xorro together.

The addition for Cyrus Plus is for an SPI option to boot for the Xena. The SPI device can be programmed via the CPU when PROG_XSPI (GPIO28) is driven low. This allows for the Xena to be booted at startup without any need for JTAG programming.

6.5 CPU COMMS

As shown in Figure 2, a number of ports from Xcore 1 are connected to the CPU via the CPLD via a mailbox interface with a shared dual port RAM. An interrupt is generated when the CPU has written data to the RAM for the Xena to read data.

Signal	Description	I/O	Xena pin
DATA[0..15]	16 bit data	bidirectional	XS1_PORT_16A
CLK	Bus clock/strobe	output	XS1_PORT_1A
WE#	Write Enable	output	XS1_PORT_1B
RE#	Read Enable	output	XS1_PORT_1C
DI	Data/Index, Data=1 index=0	output	XS1_PORT_1D
IRQ#	Interrupt	input	XS1_PORT_1F

Table 7: XMOS CPLD pin connections

The intention is that a thread on Xcore 1 should be programmed for the mailbox protocol using the pin assignment in Table 7. The protocol uses indirect addressing in that the address of the area of the memory to be accessed is first written to the index register (using DI low to select the index register), then reading or writing the data with DI driven high. Note that the index register will auto increment when reading or writing to the RAM. When finished writing data to RAM the code should write to the MBX2C register to generate an interrupt to the CPU. Example code for this will be provided.

The CPU communications to the Xena is similar to the CPLD to Xena, but the index or data registers are directly memory mapped. The index address is 0x0 and the data address is 0x8000. For example to read the SIG1 value, from the CPU you write 0x0 at address 0x0 and then read the data in 0x8000.

The memory map for Xena to the CPLD is shown below in Table 8.

Address (hex)	Register name	Description	Read/Write
0x0000	SIG1	signature value 1 (0xDEAD)	RO
0x0001	SIG2	signature value 2 (0xBEEF)	RO
0x0002	HWREV	Hardware revision	RO
0x0005	MBC2X	CPU to XMOS mailbox	RW
0x0006	MBX2C	XMOS to CPU mailbox	RO
0x8000-0x8FFF	RAM	Dual port RAM , 16 bits wide, 4kbytes	RW
0x0030	XSCR1	XMOS Scratch register	RW
0x0031	XSCR2	XMOS Scratch register	RW

Table 8: XMOS CPLD memory map

6.6 XORRO SLOT

The Xorro slot connector is physically a PCI Express x8 (98 pin) card edge connector. **Xorro cards are not compatible with PCI Express x8 cards.**

The pinout of the Xorro slot connector is provided in section 10.5, together with signal descriptions.

6.7 LEDS

A pair of simple LEDs is provided for diagnostic purposes. These are connected to Xcore 0 (port P1K) and Xcore 1 (port P1E), and are illuminated when driven low.

6.8 SPARE PORT LINE

The spare port line (Xcore 0 port P1L) is connected to the CPLD. Its use is reserved and it should be tri-stated.

6.9 PCU

Xena's PCU (Power Control Unit) is not used.

6.10 HAZE XORRO HEADER

There is an additional header to the Xena to allow for direct connection between the Xena and CPU via the serial port. The pin out of the Haze Xorro connector, H7 is shown in Table 9**.**

Pin	Signal	Signal	Pin
1	CPU RX0	X0D1 (P1B0)	2
3	CPU RX0	X0D10 (P1C0)	4
5	CPU TX0	X0D11 (P1D0)	6
7	CPU TX0	X0D12 (P1E0)	8

Table 9: Haze Xorro pinout

Warning: When using the Haze Xorro header make sure that you have configured the Xena IOs so that it is not driving outputs onto the CPU output signals.

7 CPLD

The CPLD is connect to the CPU via the local bus and allows for the high speed interface between the CPU and the XMOS. There are other read only registers which include the CPU fan speed.

7.1 CPU COMMS

The interface for the CPU to the CPLD is similar to the XMOS mail box, the address of the register to be accessed should first be written to the index register at address (0x0). The data can then be read and written using the data address (0x8000). For example to check the speed of the CPU fan TACHO you would write 0x10 into address 0x0, then read data from address 0x8000.

For more details on the Xena link protocol referrer to section 6.5.

The memory map for the CPU to the CPLD is shown below in Table 10.

Address (hex)	Register name	Description	Read/Write
0x0000	SIG1	signature value 1 (0xDEAD)	RO
0x0001	SIG2	signature value 2 (0xBEEF)	RO
0x0002	HWREV	Hardware revision	RO
0x0004	IA_LBALIVE	Local bus is alive returns 0x1	RO
0x0005	MBC2X	CPU to XMOS mailbox	RW
0x0006	MBX2C	XMOS to CPU mailbox	RO
0x000C	XDEBUG	XMOS Control see Table 11 for pin assignments	-
0x000D	XJTAG	XMOS JTAG pins see Table 12 for pin assignments	-
0x0010	FAN_TACHO[1]	Fan speed in Revolution per seconds	RO
0x0011	VID_CA_CB[2]	Core voltage value VID value	RO
0x0012	VID_PL[2]	Platform voltage VID value	RO
0x0013	PEX_SIG	PCIe wake and present signals	RO
0x0018	-	Reserved for internal use	-
0x0019	-	Reserved for internal use	-
0x001A	-	Reserved for internal use	-
0x0020	ETH2STATE	Number of Ethernet PHY fitted, [0x1=2 0x0=1]	RO
0x0021	DATE_LB	CPLD build date lower byte, see 7.1.4 for build format	RO
0x0022	DATE_UB	CPLD build date upper byte, see 7.1.4 for build format	RO
0x0023	TIME_LB	CPLD build time lower byte, see 7.1.4 for build format	RO
0x0024	TIME_UB	CPLD build time upper byte, see 7.1.4 for build format	RO
0x0030	SCR1	Scratch register	RW
0x0031	SCR2	Scratch register	RW
0x8000	RAM	Dual port RAM base address , 16 bits wide, 4kbytes	RW

Table 10: CPLD Local bus memory map

Notes:

1. **The FAN_TACHO signal should be read multiple times to get rid of metastability.**
2. **VID values are relevant to P5040 boards only.**

7.1.1 CPLD XMOS DEBUG REGISTER

Bit	Signal Name	Read/Write
0	RESET#	RW
1	DEBUG in	RO
2	DEBUG out	RW
3	DEBUG OE	RW
4		
5		
6	Slot detect	RO
7	XTAG detect	RO

Table 11: CPLD register XMOS Debug pin map

7.1.2 CPLD XMOS JTAG REGISTER

Bit	Signal Name	Read/Write
0	TCK	RW
1	TMS	RW
2	TDI	RW
3	TDO	RO
4		
5		
6		
7	TRST#	RW

Table 12: CPLD register XMOS JTAG pin map

7.1.3 CPLD PCIE REGISTER

Bit	Signal Name	Read/Write
0	PCIe x16 PRSNT#	RO
1	PCIe x4 PRSNT#	RO
2	PCIe x1 PRSNT0#	RO
3	PCIe x1 PRSNT1#	RO
4	PCIe x1 PRSNT2#	RO
5	PCIe x1 PRSNT3#	RO
6		
7		
8	PCIe WAKE#	RO

Table 13: CPLD register PCIe signals map

7.1.4 CPLD BUILD FORMAT

The format of the CPLD build time and date are stored in a 32-bit value using BCD, on for build date and one for build time. The date is store as YYYYMMDD and the time is store as 00HHMMSS.

8 MCU

The MCU is a supervisor for the Cyrus Plus motherboard and provides voltage and temperature monitoring for the CPU.

8.1 SUPERVISOR INTERFACE

The supervisor interface is like an ACPI and is connected over serial port 1 to the CPU. The connection should be setup using:

- 38400 Baud
- 8 bit data
- No Parity
- 1 Stop bit
- No Flow Control

Each packet has a start and end character. The CPU can tell the MCU to turn off the power, get temperatures and get voltages.

Commands can be pipelined as the serial interface is interrupt driven, and responses contain the command it is responding to.

All MCU to CPU messages start with '$' and end with a new line character, ASCII 0x0A.
All CPU to MCU messages start with '#' and end with a new line character, ASCII 0x0A.

8.1.1 POWER BUTTON

When the power button is pressed, a 1ms low pulse is generated on IRQ4# to the CPU. The CPU can use the interrupt to cleanly shut down the OS and the power supplies via the ACPI serial interface.

Holding the power button for greater than 5s will force the power off.

8.1.2 SHUT POWER DOWN

To shut down the power of the Cyrus Plus motherboard from the CPU, the '**s**' command is used.
Example:

CPU command	#s
MCU Returns	$s

8.1.3 TEMPERATURES READINGS

To read the temperature readings, the '**t**' command is used.

Returns the temperatures in the following format:
$t<sign>HH...<sign>HH

Where HH is the ASCII hex value of the temperature, and <sign> is either '+' or '-'.

There are three temperature available to read on the Cyrus Plus motherboard. The temperatures given are returned in this order:

1. PCB temperature
2. CPU temperature
3. PCIe switch temperature

Example:

CPU command	`#t`
MCU Returns	`$t+20+38+4B`

Represents
+32°C for the PCB temperature
+56°C for the CPU temperature
+75°C for the PCIe switch temperature

8.1.4 VOLTAGES

To read all the measured voltages, the '**v**' command is used.

Returns the voltages in the following format:
$vXXYY...XXYY

Where XX represents the whole number of volts as ASCII hex, YY represents the number of 10mV units as ASCII hex.

The voltages are sent in the order:

1. CPLD, 3.3V
2. Xena/Xorro 3.3V
3. Xena/Xorro 1.0V
4. PCIe switch, 1.0V
5. Xena 1.0V
6. 3.3V
7. 2.5V
8. Ethernet 1.2V
9. Platform, 1.0V
10. Core A, 1.0V for P3041 1.1V for P5020, 1.1V-1.2V for P5040
11. Core B, 1.0V for P3041 1.1V for P5020, 1.1V-1.2V for P5040
12. DDR3 IO, 1.5V
13. Serdes, 1.8V

Example for the above default values:

CPU command	#v
MCU Returns	`$v031E031E01000100031E023201140100010A010A013201500041`

8.1.5 CPU FAN SPEED

To read the CPU fan speed, the '**f**' command is used. This returns the fan speed (in RPM) and the fan duty cycle (0 -> 255, where 0 is off and 255 is full on).

Returns the fan status in the following format:
$fXXYYYY

Where XX is the ASCII hex value of the PWM (0x00 to 0xFE) YYYY is the ASCII hex value of the RPM, MSB first.

Example:

CPU command	#f
MCU Returns	$fC010E6

Represents
Fan PWM, 0xC0, 192 decimal
Fan RPM, 0x10E6, 4326 decimal

8.2 DEBUG SERIAL TERMINAL

The MCU also provides serial debug interface for status reporting of the read voltages and temperature rails. This uses a 6 pin FTDI USB-TTL cable pinout (P18), see Figure 4 for location.

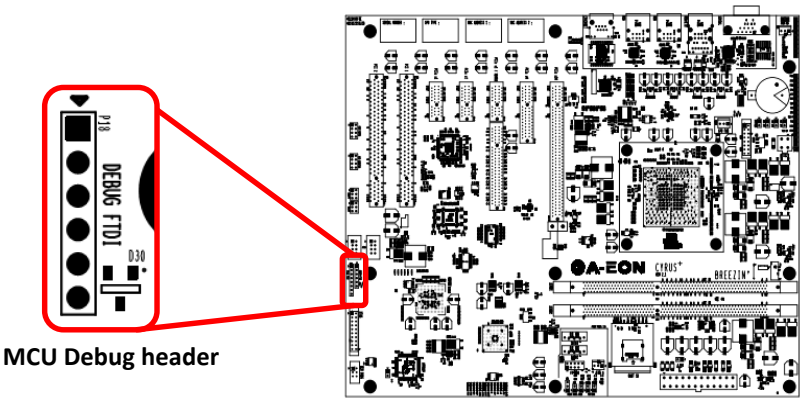

Figure 4: MCU Debug header

The pinout of the Debug serial terminal is give below in Table 14.

Pin	MCU Connection	MCU Direction
1	Ground	-
4	MCU RX	In
5	MCU TX	Out

Table 14: MCU serial pinout

Notes:
Pins 2-4 and 6 are unconnected.

To set up a serial communications on a PC, it is recommended to use TeraTerm. The serial port must be configured as follows:

- 38400 Baud
- 8 bit data
- No Parity
- 1 Stop bit
- No Flow Control

9 BOOT

This section contains specific Cyrus Plus boot information on the micro SD card and U-Boot.

9.1 MICRO SD CARD

The micro SD card contains all the U-Boot data. This is required to boot the system to U-Boot. The first 629kB or 1258 blocks contain the boot loader. The boot loader data should not be edited when accessing the SD card otherwise the system will cease to boot. The structure of the boot loader blocks of the micro SD card is shown in Table 15 below.

Function	Start block	Block count	End block
PBL image	8	1088	1096
Environment	1097	16	1113
Ethernet	1130	128	1258

Table 15: SD boot loader blocks

9.2 U-BOOT

Cyrus Plus uses a standard version of U-Boot configured for the Cyrus Plus hardware. For further reference of the U-Boot commands check out U-Boot website.

There are specific environment settings which need to be configured for the system work correctly, these are shown in Table 16. These should not be edited as they will affect the functionality of Cyrus Plus Motherboard.

Environment name	Environment value
baudrate	115200
bootargs	sole=ttyS0,115200 root=/dev/sda1
consoledev	ttyS0
hwconfig	fsl_ddr:ctlr_intlv=cacheline,bank_intlv=cs0_cs1;usb1:dr_mode=host,phy_type=utmi; usb2:dr_mode=host,phy_type=utmi
stdin	serial,usbkbd
stdout	serial,vga

Table 16: U-Boot critical settings

There are specific environment settings for booting Amiga OS and these are listed in Table 17 below.

Environment name	Environment value
aosautoboot	if run aosusbboot; then echo OK; else sata init; if run aossata0boot; then echo OK; echo if run aossata1boot; then echo OK; else run aosnetboot; fi; fi; fi
bootmenu_0	OS4 Auto Boot=run aosautoboot
bootmenu_1	USB Boot=run aosusbboot
bootmenu_2	Net Boot=run aosnetboot
bootmenu_3	SATA 0 Boot=sata init; run aossata1boot
bootmenu_4	SATA 1 Boot=sata init; run aossata1boot

Table 17: U-Boot Amiga OS boot setting

10 CONNECTOR, JUMPER AND LEDS

10.1 SWITCHES

Headers are provided for front panel power and reset switches/buttons of the momentary, normally open type.

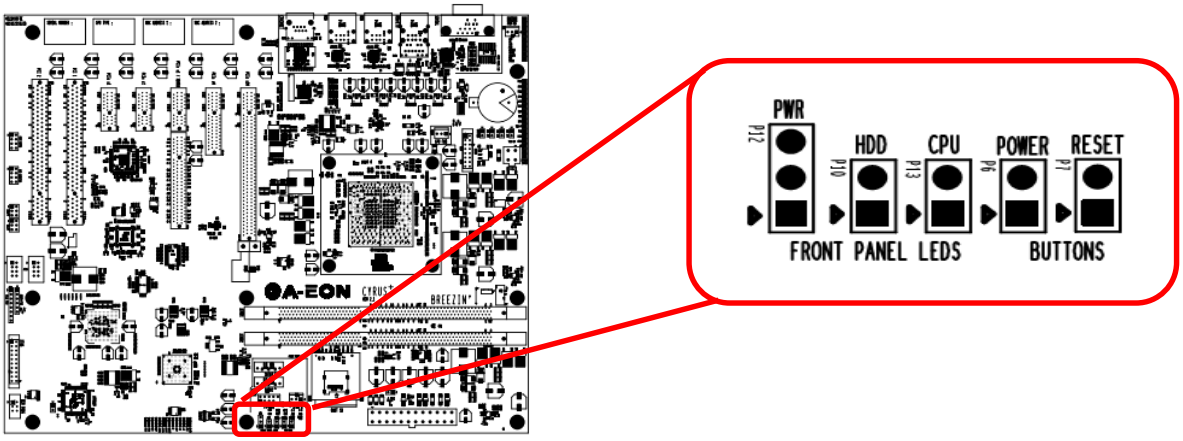

Figure 5: Front Panel LED and switches

P6 (labelled POWER) is for the power button. P7 (RESET) is for the reset button. For P6 pin 1 is grounded and pin 2 is pulled up to 3.3V. For P7 pin 2 is grounded and pin 1 is pulled up to 3.3V.

10.2 JUMPERS

Jumpers are provided to select boot configuration options, there position on the motherboard as shown in Figure 6.

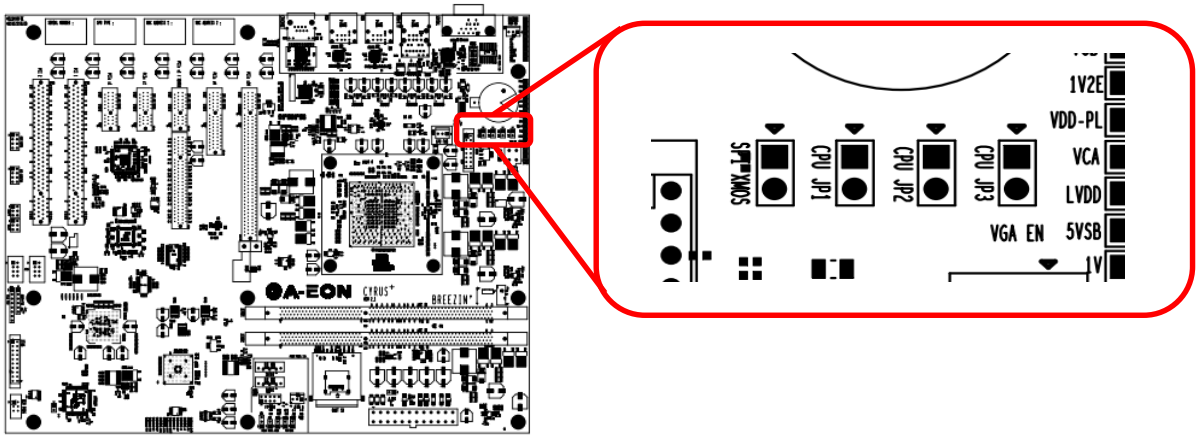

Figure 6: Cyrus Jumpers

Jumper	Description
JP1	Controls the state of CPU GPIO0 which is pulled up. Fitting a jumper causes the signal to be grounded.
JP2	Controls the state of CPU GPIO1 which is pulled up. Fitting a jumper causes the signal to be grounded.
JP3	Controls VGA enable, 1 - U-Boot serial, 0 - U-Boot VGA if available.

Table 18: Jumpers

10.3 LEDS

Cyrus provides 12 on-board LEDs and headers for 3 off board LEDs. Their location on the motherboard is shown in Figure 5 and Figure 7.

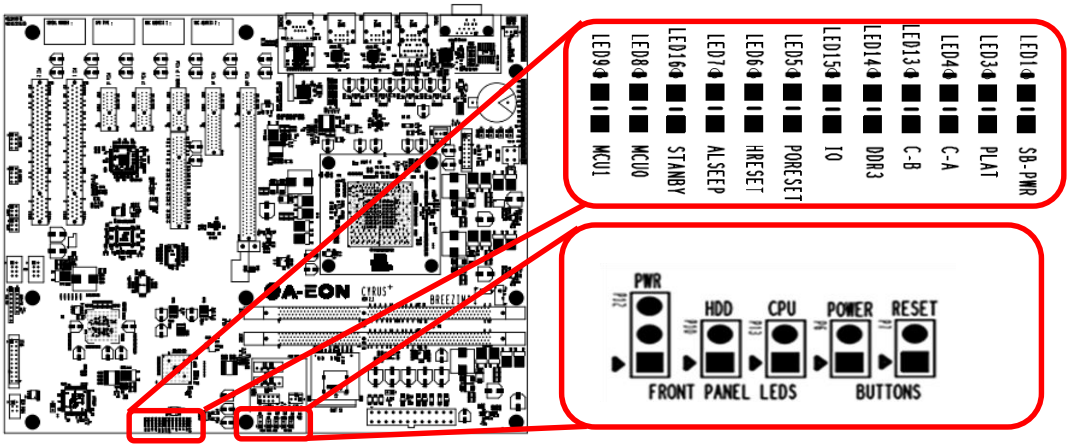

Figure 7: Cyrus Plus LEDs

Reference	Type	Description
LED1	0603, green	Standby power indicator
LED3	0603, red	CPU Platform voltage power good (CPLD controlled)
LED4	0603, red	CPU Core A voltage power good (CPLD controlled)
LED13	0603, red	CPU Core B voltage power good (CPLD controlled)
LED14	0603, red	DDR 3 voltages power good (CPLD controlled)
LED15	0603, red	IO voltages power good (CPLD controlled)
LED5	0603, red	PORESET state indicator (CPLD controlled)
LED6	0603, red	HRESET state indicator (CPLD controlled)
LED7	0603, red	ASEELP state indicator (CPLD controlled)
LED16	0603, red	STANDBY state indicator (CPLD controlled)
LED8	0603, red	MCU temperature Warning LED
LED9	0603, red	MCU Error LED
P12	0.1" header	Power on (pins 1 & 2 = anode, pin 3 = cathode)
P10	0.1" header	SATA activity, CPU GPIO5 (pin 1 = anode, pin 2 = cathode)
P13	0.1" header	CPU GPIO4 (pin 1 = anode, pin 2 = cathode)

Table 19: LEDs

Notes:

The LED header drivers are of the constant current (20mA) type and are suitable for driving LEDs with a forward voltage of between 2 and 12V (they will directly drive any standard LED assuming it is rated for 20mA or more). P12 is pinned out as follows: 1,2 = +/anode, 3 = -/cathode. P10 and P13 are pinned out as follows: 1 = +/anode, 2 = -/cathode. Pin 1 is marked by an arrow in each case.

10.4 PCIE AND PCI SLOTS

The pinout of the PCIe slots 1 shown in Table 20. The pinout for slot 2 is shown in Table 21.The pinout of slots 3, 4 and 5 is shown in Table 22. The pinout of slots 6 and 7 is shown in Table 23. The position of the connector for the PCIe and PCI slots is shown in Figure 8.

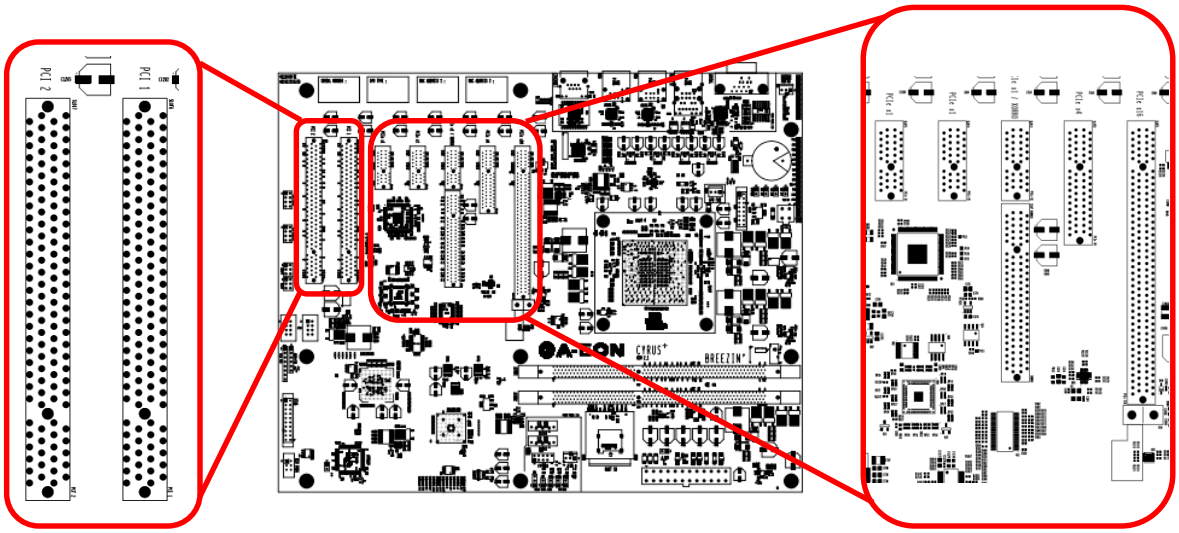

Figure 8: PCIe and PCI slots

pin	row A	row B	pin	row A	row B
1	PRSNT1#	+12V	42	GND	PETn6
2	+12V	+12V	43	PERp6	GND
3	+12V	+12V	44	PERn6	GND
4	GND	GND	45	GND	PETp7
5	TCK	SMCLK	46	GND	PETn7
6	TDI	SMDAT	47	PERp7	GND
7	TDO	GND	48	PERn7	PRSNT2#
8	TMS	+3.3V	49	GND	GND
9	+3.3V	TRST#	50	RSVD	PETp8
10	+3.3V	3.3Vaux	51	GND	PETn8
11	PERST#	WAKE#	52	PERp8	GND
12	GND	RSVD	53	PERn8	GND
13	REFCLK+	GND	54	GND	PETp9
14	REFCLK-	PETp0	55	GND	PETn9
15	GND	PETn0	56	PERp9	GND
16	PERp0	GND	57	PERn9	GND
17	PERn0	PRSNT2#	58	GND	PETp10
18	GND	GND	59	GND	PETn10
19	RSVD	PETp1	60	PERp10	GND
20	GND	PETn1	61	PERn10	GND
21	PERp1	GND	62	GND	PETp11
22	PERn1	GND	63	GND	PETn11
23	GND	PETp2	64	PERp11	GND
24	GND	PETn2	65	PERn11	GND
25	PERp2	GND	66	GND	PETp12
26	PERn2	GND	67	GND	PETn12
27	GND	PETp3	68	PERp12	GND
28	GND	PETn3	69	PERn12	GND
29	PERp3	GND	70	GND	PETp13
30	PERn3	RSVD	71	GND	PETn13
31	GND	PRSNT2#	72	PERp13	GND
32	RSVD	GND	73	PERn13	GND
33	RSVD	PETp4	74	GND	PETp14
34	GND	PETn4	75	GND	PETn14
35	PERp4	GND	76	PERp14	GND
36	PERn4	GND	77	PERn14	GND
37	GND	PETp5	78	GND	PETp15
38	GND	PETn5	79	GND	PETn15
39	PERp5	GND	80	PERp15	GND
40	PERn5	GND	81	PERn15	PRSNT2#
41	GND	PETp6	82	GND	RSVD

Table 20: PCIe x16 Slots Pinout

Note: Slot 1 lanes 4-15 are always no connect.

pin	row A	row B
1	PRSNT1#	+12V
2	+12V	+12V
3	+12V	+12V
4	GND	GND
5	TCK	SMCLK
6	TDI	SMDAT
7	TDO	GND
8	TMS	+3.3V
9	+3.3V	TRST#
10	+3.3V	3.3Vaux
11	PERST#	WAKE#
12	GND	RSVD
13	REFCLK+	GND
14	REFCLK-	PETp0
15	GND	PETn0
16	PERp0	GND
17	PERn0	PRSNT2#
18	GND	GND
19	RSVD	PETp1
20	GND	PETn1
21	PERp1	GND
22	PERn1	GND
23	GND	PETp2
24	GND	PETn2
25	PERp2	GND
26	PERn2	GND
27	GND	PETp3
28	GND	PETn3
29	PERp3	GND
30	PERn3	RSVD
31	GND	PRSNT2#
32	RSVD	GND

Table 21: PCIe x4 Slots Pinout

pin	Row A	Row B
1	PRSNT1#	+12V
2	+12V	+12V
3	+12V	+12V
4	GND	GND
5	TCK	SMCLK
6	TDI	SMDAT
7	TDO	GND
8	TMS	+3.3V
9	+3.3V	TRST#
10	+3.3V	3.3VAUX
11	PERST#	WAKE#
12	GND	RSVD
13	REFCLK+	GND
14	REFCLK-	PETp0
15	GND	PETn0
16	PERp0	GND
17	PERn0	PRSNT2#
18	GND	GND

Table 22: PCIe x1 Slots Pinout

Pin	Row A	Row B	Pin	Row A	Row B
1	TRST#	-12V	32	AD[16]	AD[17]
2	+12V	TCK	33	+3.3V	C/BE[2]#
3	TMS	GND	34	FRAME#	GND
4	TDI	TDO	35	GND	IRDY#
5	+5V	+5V	36	TRDY#	+3.3V
6	INTA#	+5V	37	GND	DEVSEL#
7	INTC#	INTB#	38	STOP#	GND
8	+5V	INTD#	39	+3.3V	LOCK#
9	RSVD	PRSNT1#	40	RSVD	PERR#
10	+5V	RSVD	41	RSVD	+3.3V
11	RSVD	PRSNT2#	42	GND	SERR#
12	GND	GND	43	PAR	+3.3V
13	GND	GND	44	AD[15]	C/BE[1]#
14	3.3VAUX	RSVD	45	+3.3V	AD[14]
15	RST#	GND	46	AD[13]	GND
16	+5V	CLK	47	AD[11]	AD[12]
17	GNT#	GND	48	GND	AD[10]
18	GND	REQ#	49	AD[09]	GND
19	PME#	+5V	50	-KEY-	-KEY-
20	AD[30]	AD[31]	51	-KEY-	-KEY-
21	+3.3V	AD[29]	52	C/BE[0]#	AD[08]
22	AD[28]	GND	53	+3.3V	AD[07]
23	AD[26]	AD[27]	54	AD[06]	+3.3V
24	GND	AD[25]	55	AD[04]	AD[05]
25	AD[24]	+3.3V	56	GND	AD[03]
26	IDSEL	C/BE[3]#	57	AD[02]	GND
27	+3.3V	AD[23]	58	AD[00]	AD[01]
28	AD[22]	GND	59	+5V	+5V
29	AD[20]	AD[21]	60	REQ64#	ACK64#
30	GND	AD[19]	61	+5V	+5V
31	AD[18]	+3.3V	62	+5V	+5V

Table 23: PCI Slots Pinout

10.5 XENA CONNECTORS

The pinout of the Xorro slot is shown in Table 24 and Table 25 below. See Table 26 for signal descriptions. The pinout of the XTAG (XMOS JTAG) header is shown in Table 27.

			Ports				Links	
Pin	**Signal**	**Xcore**	**1 bit**	**4 bit**	**8 bit**	**16 bit**	**5 bit**	**2 bit**
A1	PRESENT#							
A2	3.3V							
A3	GND							
A4	CLK							
A5	GND							
A6	DEBUG							
A7	RESET#							
A8	*reserved*							
A9	3.3V							
A10	GND							
A11	5V							
A12	GND							
A13	X0D1	0	P1B0				XLA4out	
A14	X0D11	0	P1D0					
A15	X0D13	0	P1F0				XLB4out	
A16	GND							
A17	X0D14	0		P4C0	P8B0	P16A8	XLB3out	
A18	X0D15	0		P4C1	P8B1	P16A9	XLB2out	
A19	X0D16	0		P4D0	P8B2	P16A10	XLB1out	XLB1out
A20	X0D17	0		P4D1	P8B3	P16A11	XLB0out	XLB0out
A21	X0D18	0		P4D2	P8B4	P16A12	XLB0in	XLB0in
A22	X0D19	0		P4D3	P8B5	P16A13	XLB1in	XLB1in
A23	X0D20	0		P4C2	P8B6	P16A14	XLB2in	
A24	X0D21	0		P4C3	P8B7	P16A15	XLB3IN	
A25	GND							
A26	X0D23	0	P1H0					
A27	X0D25	0	P1J0					
A28	GND							
A29	X0D36	0	P1M0		P8D0	P16B8		
A30	X0D37	0	P1N0		P8D1	P16B9		
A31	X0D38	0	P1I0		P8D2	P16B10		
A32	X0D39	0	P1P0		P8D3	P16B11		
A33	X0D40	0			P8D4	P16B12		
A34	X0D41	0			P8D5	P16B13		
A35	X0D42	0			P8D6	P16B14		
A36	X0D43	0			P8D7	P16B15		
A37	GND							
A38	X1D23	1	P1H0					
A39	X1D25	1	P1J0					
A40	GND							
A41	X1D30	1		P4F2	P8C4	P16B4		
A42	X1D31	1		P4F3	P8C5	P16B5		
A43	X1D32	1		P4E2	P8C6	P16B6		
A44	X1D33	1		P4E3	P8C7	P16B7		
A45	GND							
A46	X1D35	1	P1L0					
A47	X1D37	1	P1N0		P8D1	P16B9		
A48	X1D39	1	P1P0		P8D3	P16B11		
A49	GND							

Table 24: Xorro Slot Pinout - A row

			Ports				**Links**	

Pin	Signal	Xcore	1 bit	4 bit	Pin	Signal	Xcore	1 bit
B1	GND							
B2	3.3V							
B3	TRST#							
B4	GND							
B5	TCK							
B6	TMS							
B7	TDI							
B8	TDO							
B9	3.3V							
B10	GND							
B11	5V							
B12	GND							
B13	X0D0	0	P1A0					
B14	X0D10	0	P1C0				XLA4in	
B15	X0D12	0	P1E0					
B16	GND							
B17	X0D2	0		P4A0	P8A0	P16A0	XLA3out	
B18	X0D3	0		P4A1	P8A1	P16A1	XLA2out	
B19	X0D4	0		P4B0	P8A2	P16A2	XLA1out	XLA1out
B20	X0D5	0		P4B1	P8A3	P16A3	XLA0out	XLA0out
B21	X0D6	0		P4B2	P8A4	P16A4	XLA0in	XLA0in
B22	X0D7	0		P4B3	P8A5	P16A5	XLA1in	XLA1in
B23	X0D8	0		P4A2	P8A6	P16A6	XLA2in	
B24	X0D9	0		P4A3	P8A7	P16A7	XLA3IN	
B25	GND							
B26	X0D22	0	P1G0				XLB4in	
B27	X0D24	0	P1I0					
B28	GND							
B29	X0D26	0		P4E0	P8C0	P16B0		
B30	X0D27	0		P4E1	P8C1	P16B1		
B31	X0D28	0		P4F0	P8C2	P16B2		
B32	X0D29	0		P4F1	P8C3	P16B3		
B33	X0D30	0		P4F2	P8C4	P16B4		
B34	X0D31	0		P4F3	P8C5	P16B5		
B35	X0D32	0		P4E2	P8C6	P16B6		
B36	X0D33	0		P4E3	P8C7	P16B7		
B37	GND							
B38	X1D22	1	P1G0				XLB4in	
B39	X1D24	1	P1I0					
B40	GND							
B41	X1D26	1		P4E0	P8C0	P16B0		
B42	X1D27	1		P4E1	P8C1	P16B1		
B43	X1D28	1		P4F0	P8C2	P16B2		
B44	X1D29	1		P4F1	P8C3	P16B3		
B45	GND							
B46	X1D34	1	P1K0					
B47	X1D36	1	P1M0		P8D0	P16B8		
B48	X1D38	1	P1I0		P8D2	P16B10		
B49	GND							

Table 25: Xorro Slot Pinout - B row

Signal Name(s)	Direction[1]	Description	Notes for Card Design
XnDn	I/O	XMOS port/link signals	
PRESENT#	O	card presence detect	connect to ground
RESET#	I	Reset, as used by Xena	
CLK	I	25MHz clock, as used by Xena	buffer if used
DEBUG	I/O	XMOS debug signal, as used by Xena	
TRST#	I	JTAG test reset	
TCK	I	JTAG test clock	
TMS	I	JTAG test mode select	
TDI	I	JTAG test data in	see note 2
TDO	O	JTAG test data out	see note 2

Table 26: Xorro Slot Signal Descriptions

Notes:

3. **Signal direction is with respect to the Xorro card. "I" signifies a signal driven from the motherboard to the card.**
4. **Cards that do not provide any JTAG devices should connect TDI to TDO, and leave other JTAG signals unconnected.**

Pin	Signal	Signal	Pin
1	n/c	n/c	2
3	TRST#	GND	4
5	XTDI	n/c	6
7	TMS	GND	8
9	TCK	n/c	10
11	DEBUG	GND	12
13	TDO	n/c	14
15	RESET#	GND	16
17	n/c	n/c	18
19	n/c	XTAG#	20

Table 27: XTAG Connector Pinout

Notes:

1. **The XTAG# signal is wired to ground on the XTAG debugger and is used to sense its presence by Cyrus Plus (it is pulled up to 3.3V).**

10.6 PROGRAMMING HEADERS

The locations for the programing header are located below in Figure 9.

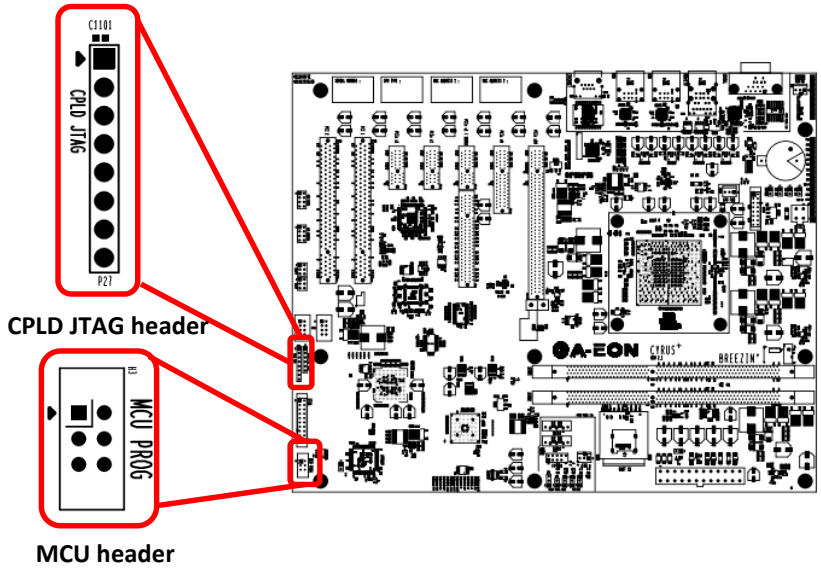

Figure 9: Programming Headers

10.6.1 CPLD

The pinout for H2 (labelled PLD JTAG) is shown in Table 28 below:

Pin	Signal
1	3.3VSB
2	TDO
3	TDI
4	n/c
5	n/c
6	TMS
7	Ground
8	TCK

Table 28: CPLD JTAG Header

10.6.2 MCU

The pinout for H3 (labelled MCU PROG) is shown in Table 28 below:

Pin	Signal
1	PDI DATA
2	3.3VSB
3	n/c
4	n/c
5	PDI CLK
6	Ground

Table 29: MCU Programming Header