MIPS® SEAD™-3 Board User's Manual

Document Number: MD00682
Revision 01.03
July 1, 2010

MIPS Technologies, Inc.
955 East Arques Avenue
Sunnyvale, CA 94085-4521

Template: nB1.03, Built with tags: 2B

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

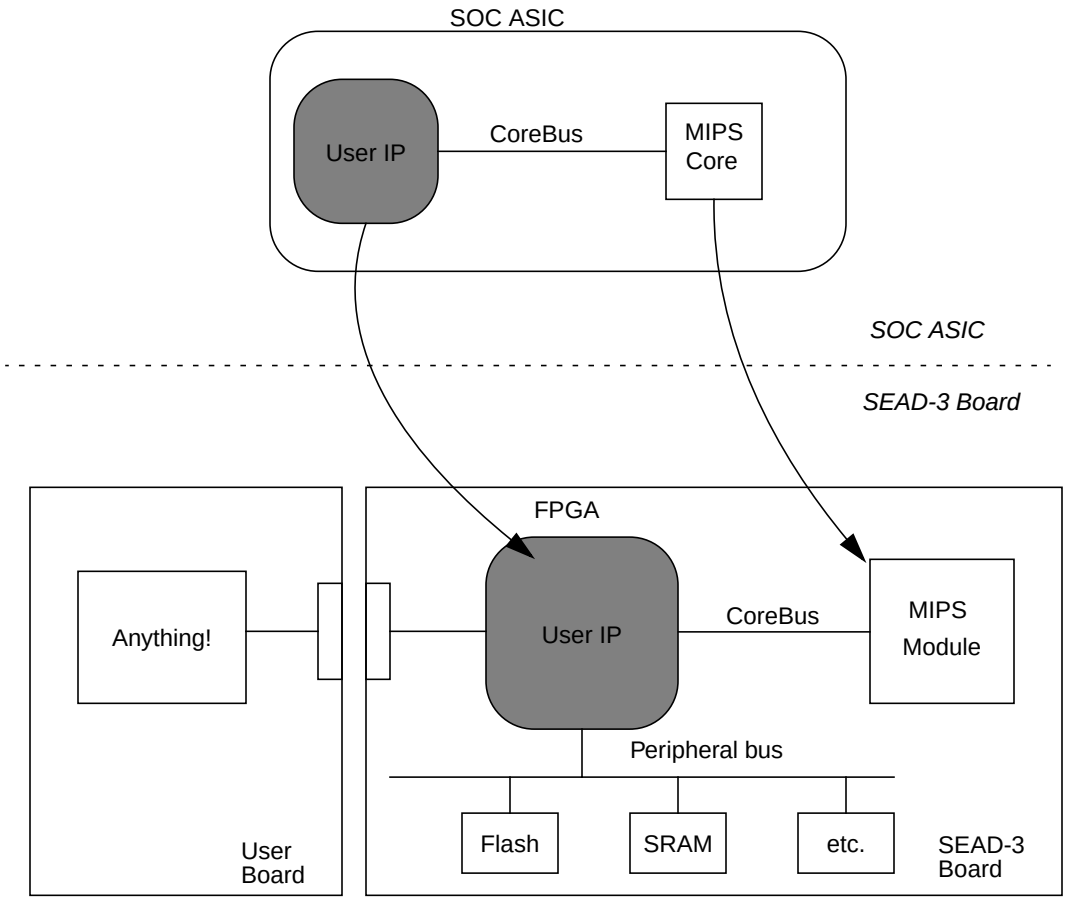
This document describes the MIPS® *S*OC *E*valuation *A*nd *D*evelopment (SEAD) 3 board. The SEAD™-3 board is basically a playground where users can develop and verify their hardware and IP that includes a MIPS CPU core.

The SEAD-3 board contains a CoreBus Connector that accepts a small PCB with a MIPS 4K®, M14K®, 5K®, 24K®, or 34K® core. The CPU interfaces to a large Xilinx FPGA which is uncommitted and available for user IP. The board is shipped with a version of MIPS Basic RTL (BRTL) in the FPGA. Subject to licensing, the user can obtain this RTL as example code to use. There are third-party RTL modules in the BRTL, which MIPS is not allowed to distribute in RTL form, although MIPS will provide details on how users can obtain their own licences.

The SEAD-3 board is the ideal solution for customers who are designing a System On a Chip (SOC) ASIC which includes a MIPS CPU core. An SOC design will typically contain the CPU, bus controllers, and other IP. The IP can be synthesized to the FPGA and verified together with the CPU core at a speed which is beyond what any emulator or simulator can provide. The user may extend board functionality by designing a custom PCB that interfaces to SEAD-3 through a number of SEAD-3 Expansion Connectors. The board also enables software engineers to begin code development and testing before the final ASIC is available. So instead of having a sequential development of ASIC and software, the process can be speeded up by performing the tasks in parallel.

Figure 1.1 below illustrates how an SOC ASIC with CPU and user IP can be mapped to the SEAD-3 board.

Figure 1.1 SOC ASIC to SEAD™-3 Board Mapping



Both versions of the SEAD-3 board, A80209 and A80211, are pre-equipped with a number of resources including I2C, SPI, GPIO, ADC, MicroSD slot, Serial and USB UART, Ethernet controller, and SRAM and Flash memory. The DDR2 Controller and USB 2.0 PHY are only available on A80209.

There are either 238 or 356 FPGA signals (depending on the FPGA type) on the SEAD-3 Expansion Connectors which are available for the user board interface. See Table 16.1 and Table 16.3.

There are 264 FPGA signals on the SEAD-3 CoreBus Connector used for the FPGA_Module-3. See Table 10.1.

1.1 SEAD™-3 Board at a Glance

- The SEAD-3 CoreBus Connector accepts a CPU module with a MIPS CPU core. The CPU can either be synthesized in an FPGA variant, or be a Lead Vehicle variant in bond-out mode and with the core interface connected to the FPGA. Alternatively, if the core and system controller are housed in the on-board FPGA, the connector can be used for a custom add-on board.
- The SEAD-3 CoreBus Connector signals are defined in CORE_B[264:1]. This bus can be defined as an EC, OCP, or AHB bus. Alternatively, the user may define his own custom bus.
- A80209 has a Xilinx XCVLX110 FPGA with 800 I/Os and 155,000 LUTs.

- A80211 has a Xilinx XCVLX50 FPGA with 560 I/Os and 46,000 LUTs.
- 32 MBytes of boot Flash containing FPGA boot code and boot software, 8 bits wide.
- Fast and easy programming of boot Flash using USB at 12 MBit/s.
- 32 MBytes of uncommitted Flash memory, 32 bits wide.
- 4 MBytes of uncommitted SRAM, 32 bits wide.
- DDR II - PC2700 533MHz 200-pin SODIMM socket – for A80209 only.
- Two DIP switches for board configuration:
  - SW1: Table 7.1, Table 7.2, and Table 7.3
  - SW2: Table 13.1
- Two uncommitted 8-position DIP switches on peripheral bus for user-defined purposes.
- 2-line 16-character alphanumeric LCD display.
- 16 LEDs attached to peripheral bus - Table 14.1.
- Two standard 16550 UARTs including a level translator and one DB9 connector, and an RS232 to USB converter that provides a USB serial port.
- SEAD-3 Expansion Connector to user-defined PCB with approximately 238 undefined I/Os, connected to the FPGA. A second connector provides the peripheral bus and Xilinx programming interface for an additional two FPGAs.
- Jumper-selectable source for FPGA configuration (Table 12.1):
  - Download from PC or using a Xilinx cable (DLCx) through JTAG chain
  - Automatic boot from boot Flash
- HP Logic Analyzer probes can be connected to the SEAD-3 Expansion Connector via a small (optional) adaptor board.
- EJTAG / PDtrace / iFlowtrace connectors for connecting directly to an internal CPU, enabling the use of debugging tools ([7], [9], [10]). These connectors are also available on the FPGA-based FPGA_Module-3 ([7]).

Figure 1.2 shows a block diagram of the board with all the major busses.

Figure 1.2 SEAD™-3 Board Block Diagram

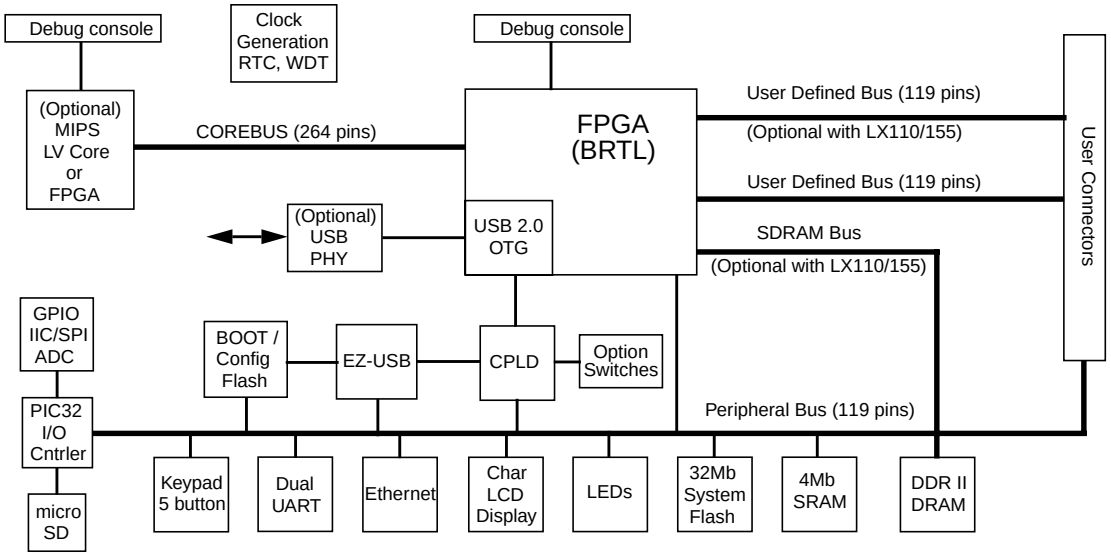


Table 1.1 describes the busses shown in Figure 1.2.

Table 1.1 Busses on SEAD™-3 Board

Name	Number of Signals	Description
CoreBus	264	The CoreBus bus is the signal interconnect between the module and the BRTL FPGA.The signals are not bus-specific, and several busses can share the signals. The primary bus support is for EC, AHB lite, and OCP. For users incorporating the CPU with the BRTL, this connector can be used for other peripherals. This bus is always connected to the FPGA (J14).
Peripheral Bus	78	This is the main peripheral bus used to connect the peripheral devices. The timing of this bus can be varied either by software or by changing the default values in the BRTL build. However, most of the peripherals on this bus have minimum and maximum cycle timings. The bus contains 32 data signals (PI_D[31:0]), 25 address bits (PI_A[24:0]), and control signals. See Chapter 9, “Peripheral Bus” on page 49 for details.
SEAD-3 Expansion Connector	128 / 238	Two connectors carry uncommitted signals from the FPGA. Depending on the board variant, you will have access to either 128 (LX50) or 238 (LX110) signals (A3,A4,A6).
DRAM bus	101	The DDR II DRAM is connected to dedicated pins on the FPGA (LX110 version only). Check Chapter 8, “DRAM Interface” on page 47 for details.

Table 1.1 Busses on SEAD™-3 Board (Continued)

Name	Number of Signals	Description
EJTAG iFlowtrace PDTrace	7 5 28	These buses carry EJTAG, iFlowtrace, and PDTrace signals for debugging of the CPU. The EJTAG connector is a standard 2x7 pin 100 mil header which can be connected to an EJTAG probe. One pin is intentionally removed for polarization. For more information on EJTAG. See Ref[7]. The iFlowtrace connector is a 2x5 pin 100-mil header which can be connected to an iFlowtrace probe. See Ref[9]. The PDTrace connector is a 38-pin Mictor connector which can be connected to a Trace probe. See Ref[10].

Chapter 2

Memory Map Decode

The X-bus controller BIU (which we will also refer to as the “X-bus controller” in the following) has a BIU interface and four X-bus master ports. The address bus and write data bus originating from the X-bus controller are actually shared between the X-bus targets. However, each of the four targets has its own read data bus, in order to avoid tri-state busses.

2.1 First-level Address Mapping

The X-bus controller contains a fixed (hardcoded) first-level address mapping, which maps the physical address space from the CPU to the X-bus ports. An X-bus may be used for *data transfers* (typically memories) and/or *register accesses*. For X-busses with both types of functions, the X-bus controller decodes two segments. The address mapping is shown in Table 2.1.

Table 2.1 SEAD™-3 X-bus Controller First-level Address Map

Base Address		Size	Device	Function
0x0000	0000	256 MByte	DRAM Memory	TARGET#0 - Data
0x1000	0000	176 MByte	DRAM Memory	TARGET#0 - Data
0x1B00	0000	1 MByte	DRAM	TARGET#0 - Registers
0x1B10	0000	1 MByte	CFG/GIC	TARGET#1 - Registers
0x1B20	0000	1 MByte	USB-HS 2.0 OTG	TARGET#2 - Registers
0x1B30	0000	1 MByte	Reserved	Reserved
0x1B40	0000	12 MBytes	Reserved	Reserved
0x1C00	0000	64 MBytes	Memory / IO	TARGET#3

The X-bus controller decodes only address bits 28:20 in order to map the accesses to the correct target ports. Since the BIU interface address bits 35:29 for EC bus, 31:29 for AHB and OCP, are not included in the address decoding, the address segment 0x0.0000.0000-0x0.1FFF.FFFF will be “mirrored” to 0x0.2000.0000-0x0.3FFF.FFFF, and 0x0.4000.0000-0x0.5FFF.FFFF, etc.

Although the physical address bus width on the EC interface is 36 bits, we will only show 32-bit addresses in the rest of this document, because the upper 4 bits are not decoded and should always be zero.

Table 2.2 Target Register Decode

TARGET#0 Register Decode				
0x1B00	0040	64 bytes	SD_SPDCNF	Presence Detect Configuration
0x1B00	0048	64 bytes	SD_SPADDR	Presence Detect Read Address register
0x1B00	0050	64 bytes	SD_SPADDR	Presence Detect Read Data register
TARGET#1 Register Decode				
0x1B10	0110	4 bytes	SEAD3 Configuration Register	
0x1B1C	0000	128 Kbytes	GIC Base Address	GIC Interrupt Controller
0x1B1C	0000	32 Kbytes	Shared Section	Base + offset - 0x0000 to 0x7fff
0x1B1C	0000	16 Kbytes	VPE-Local Section	Base + offset - 0x8000 to 0xbfff
0x1B1C	0000	16 Kbytes	VPE-Other Section	Base + offset - 0xc000 to 0xffff
0x1B1D	0000	64 Kbytes	UserMode Visible Section	Base + offset - 0x10000 to 0x1ffff
TARGET#2 Register Decode				
0x1B20	0000	512 bytes	USB 2.0 HS Controller base address	
0x1B20	0000	256 bytes	USB 2.0 HS Controller	Identification Registers, 0x000 to 0x00fc
0x1B20	0100	64 bytes	USB 2.0 HS Controller	Capability Registers, 0x0100 to 0x0124
0x1B20	0140	192 bytes	USB 2.0 HS Controller	Operational Registers, 0x0140 to 0x01fc
TARGET#3 Register Decode				
0x1F00	0010	16 bytes	PI_TIMSRAM	SRAM timing parameters
0x1F00	0020	16 bytes	PI_TIMOTHER	Timing parameters for other external peripherals
0x1F00	0040	8 bytes	PI_NMISTATUS	Interrupt latch status register
0x1F00	0048	8 bytes	PI_NMIACK	NMI interrupt acknowledge register
0x1F00	0050	16 bytes	PI_SWRESET	SW board reset register
0x1F00	0060	16 bytes	PI_PIC32_USB_STATUS	PIC32, USB Status register
0x1F00	0070	16 bytes	PI_SOFTENDIAN	Soft endian register

2.2 Second-level Address Mapping

The TARGET#3 device is responsible for subdecoding in the 0x1C000.0000 - 0x1FFF.FFFF address space.

See Table 2.3 for the memory map.

Table 2.3 SEAD™-3 Physical Memory Map as Decoded by Target#3

Base Address		Size	Device	Function
0x1C00	0000	32 MBytes	System FLASH	TARGET#3 - Data
0x1E00	0000	4 MBytes	System SRAM	TARGET#3 - Data

Table 2.3 SEAD™-3 Physical Memory Map as Decoded by Target#3 (Continued)

Base Address		Size	Device	Function
0x1E00	0000	4 MBytes	M14K SRAM Base Address Configuration Register	When used, SRAM must be mapped to Address 0x0000-0000
0x1E40	0000	4 MBytes	Optional SRAM	TARGET#3 - Data
0x1E80	0000	8 MBytes	Reserved	TARGET#3 - Data
0x1F00	0000	512 Bytes	FPGA Internal Registers	TARGET#3 - Data
0x1F00	0200	8 Bytes	P-SWITCH	TARGET#3 - Data
0x1F00	0208	8 Bytes	F-SWITCH	TARGET#3 - Data
0x1F00	0210	8 Bytes	P-LED	TARGET#3 - Data
0x1F00	0218	8 Bytes	F_LED	TARGET#3 - Data
0x1F00	0220	8 Bytes	NEWSC - Live	TARGET#3 - Data
0x1F00	0228	8 Bytes	NEWSC - Registered	TARGET#3 - Data
0x1F00	0230	16 Bytes	NEWSC - Control	TARGET#3 - Data
0x1F00	0240	192 Bytes	Reserved	TARGET#3 - Data
0x1F00	0300	256 Bytes	Reserved	TARGET#3 - Data
0x1F00	0400	128Bytes	LCD Display	TARGET#3 - Data
0x1F00	0480	128Bytes	Device Reset	TARGET#3 - Data
0x1F00	0500	256 Bytes	Reserved	TARGET#3 - Data
0x1F00	0600	256 Bytes	PIC32 Registers	TARGET#3 - Data
0x1F00	0700	256 Bytes	Reserved	TARGET#3 - Data
0x1F00	0800	256 Bytes	UART CH 0	TARGET#3 - Data
0x1F00	0900	256 Bytes	UART CH 1	TARGET#3 - Data
0x1F00	0A00	62 KBytes	Reserved	Reserved
0x1F01	0000	64 KBytes	Ethernet Controller	TARGET#3 - Data
0x1F02	0000	3968 Kbytes	Reserved	Reserved
0x1F40	0000	4 MBytes	User Expansion	TARGET#3 - Data
0x1F80	0000	2 MBytes	Reserved	Reserved for FPGA
0x1FA0	0000	2 MBytes	Boot Flash Extension	Reserved for System S/W
0x1FC0	0000	4 MBytes	Boot Flash	System Software
0x1FC0	0010	4 Bytes	Revision Register	Overlays Boot Flash

Note: Address 0x1FC0.0010 is “special”, in the sense that it is overridden and does *not* decode to an address in the SW-EPROM, but rather to register address REVISION. This is done to ensure future compatibility—the System ROM monitor uses the REVISION register to identify the hardware platform and configure its drivers accordingly.

2.3 Uncached Access of Registers

To avoid potential cache coherency problems, all registers (e.g. configuration registers internal to Basic RTL modules and registers in the peripheral bus devices) must be accessed in uncached mode. If the program runs in kernel mode, the registers can be accessed via the kseg1 mapping, since kseg1 is always noncacheable. In the rest of this document, only 32-bit physical register addresses are provided; a 32-bit physical address can be converted to a 32- bit

kseg1 address by OR'ing it with 0xA000.0000. For example, if peripheral bus controller register PI_TIMSRAM has physical address 0x1f00.0010, the virtual kseg1 address that kernel mode programs should use, is (0x1F00.0010 | 0xA000.0000) = 0xBF00.0010.

2.4 Accesses to Illegal/Reserved Addresses

If the CPU attempts to access any of the above reserved areas, the X-bus controller will map those accesses to X-bus Target #2 as type "Data". This target's normal function is "Register" only, so it is easy for this target to detect these illegal accesses (and it simplified the implementation of the X-bus controller). The X-bus targets will generally signal "read bus error" to the master, in case of illegal read accesses. The master will forward any read bus error to the CPU, which will take an exception. In order to make sure that any illegal write access is noticed, all the X-bus targets are required to issue a "write access error pulse" to some extra NMI logic in the peripheral bus controller when they detect an illegal write access. The NMI logic will then cause the CPU to take an NMI exception whenever a "write access error pulse" is detected from any of the targets.

2.5 BRTL Register Definitions

Table 2.4 SEAD3_CFG Register Field Descriptions (Address = 0x1B10,0110)

Fields		Description	Access
Name	Bits		
0	31:5	Must be written as zero; returns zero on read.	RO
USB_PRESENT	4	0 = Not present, no USB support 1 = Present, USB-HS 2.0 Controller Interface	RO
DDR2_PRESENT	3	0 = Not present, no dram support and sram will be mapped to address 0x0 1 = Present, DRAM interface is DDR2	RO
SRAM_SIZE	2	0 = SRAM size is 4MB, FPGA_OPT switch in OFF position 1 = SRAM size is 8MB, FPGA_OPT switch in ON position, Only available for Rev03 boards	RO
GIC_PRESENT	1	0 = Not present. Interrupts are directly mapped. 1 = Present. Interrupt Controller is GIC.	RO
ADDRESS_0X0_DEVICE	0	0 = DRAM mapped to address 0x0, SRAM_MAP_ZERO switch in OFF position 1 = SRAM mapped to address 0x0, SRAM_MAP_ZERO switch in ON position	RO

Table 2.5 PI_PIC32_USB_STATUS Register Field Descriptions (Address = 0x1F00,0060)

Fields		Description	Read / Write
Name	Bits		
0	31:4	Must be written as zero; return zero on read.	
GPIOB_INT	3	0 = Not active 1 = Active	RO
GPIOA_INT	2	0 = Not active 1 = Active	RO
SPI_ INT	1	0 = Not active 1 = Active	RO

Table 2.5 PI_PIC32_USB_STATUS Register Field Descriptions (Address = 0x1F00,0060)

Fields		Description	Read / Write
Name	Bits		
IO_RDY	0	0 = Not ready 1 = Ready	RO

Table 2.6 PI_SOFTENDIAN Register Field Descriptions (Address = 0x1F00,0070)

Fields		Description	Read / Write
Name	Bits		
SOFT_ENDIAN EIC CONTROL	31	0 = Disable 1 = Enable	R/W
SEAD3_CFG_PRESENT	30	0 = Not present 1 = Present	RO
0	29:3	Must be written as zero; returns zero on read.	RO
EIC_MODE	2	0 = Disable 1 = Enable	R/W
DONE	1	Set on write to bit 0. Cleared on write to PI_SWRESET register.	RO
ENDIANESS	0	0 = Little Endian 1 = Big Endian	R/W

Chapter 3

CPLD Controller

The following features are implemented in the CPLD:

- Power Controller, controls power to the board
- Module Voltage Control, controls voltage level to modules on the board
- Board Controller, controls state of the board
- Reset and Chip Selects to various devices on the board
- Periodic Pulse Generator
- LCD Interface controller
- Switch registers and LED registers, drivers

3.1 Power Controller

Board power is controlled by using the NMI push button. Push NMI button to Power ON the board,.To turn off power, push and hold NMI button for at least 7 seconds. Power Controller states are OFF, OFF2ON, ON, ON2OFF.

3.2 Module Voltage Control

CPLD will map VID codes from the Module board to Voltage Enable pins as shown in Table 3.1.

Table 3.1 Module Voltage Control Truth Table

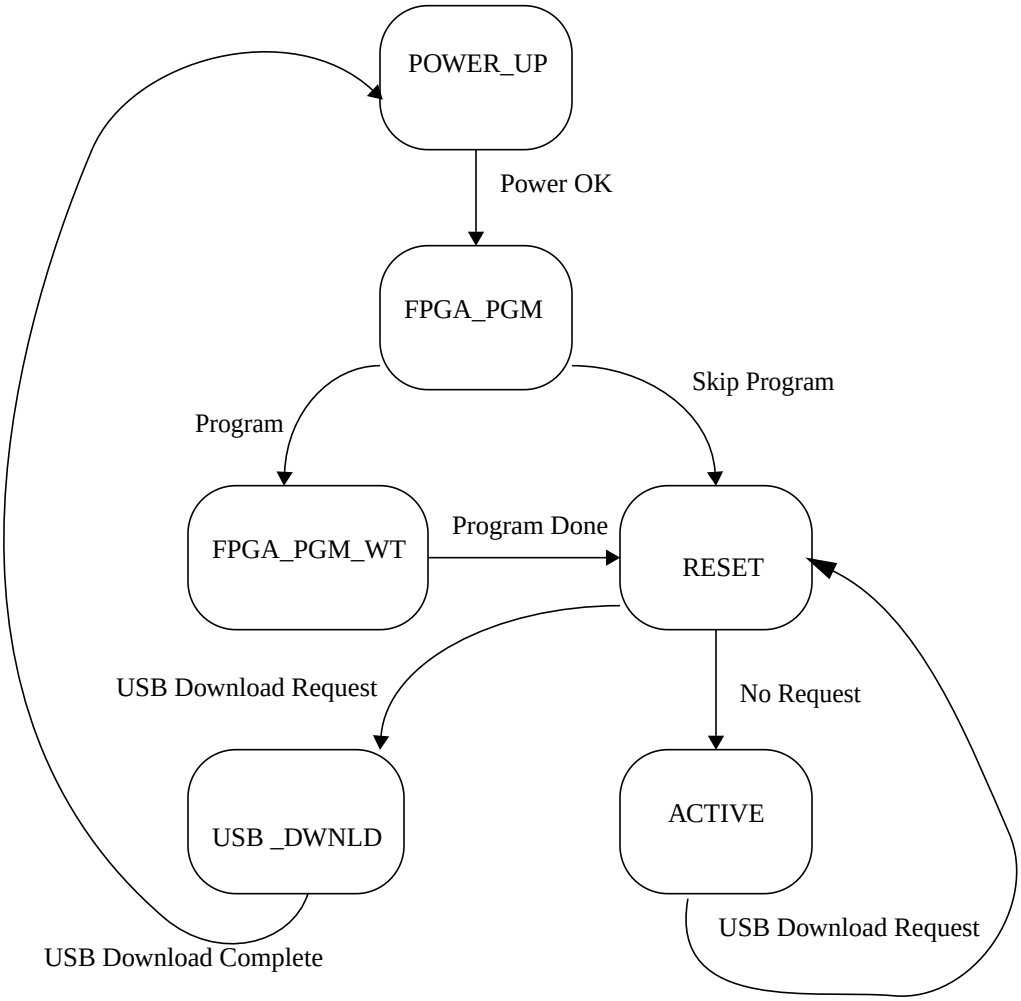
Inputs								Outputs							
Function	module_present_n	VID [5]	VID [4]	VID [3]	VID [2]	VID [1]	VID [0]	dxv_ inhibit	vlXvYen[1] 2V5	1V8	1V5	1V2	1V1	1V0	0V9
No Module	1	X[2]	X	X	X	X	X	1	0	0	0	0	0	0	0
Inhibit	0	X	1	1	1	1	1	1	0	0	0	0	0	0	0
Unused codes	0	?[3]	?	?	?	?	?	1	0	0	0	0	0	0	0
2.5V	0	1	1	1	0	1	0	0	1	0	0	0	0	0	0
1.8V	0	1	0	0	1	0	1	0	0	1	0	0	0	0	0
1.5V	0	1	0	1	0	1	1	0	0	0	1	0	0	0	0
1.2V	0	0	1	1	1	0	1	0	0	0	0	1	0	0	0
1.1V.	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0
1.0V	0	0	0	0	0	0	1	0	0	0	0	0		1	0
0.9V	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1
0.85V	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

1. vlXvYen Outputs, where XvY is 2v5, 1v8, 1v5, 1v2, 1v1, 1v0, 0v9
2. X = Don’t care
3. ? = Undriven

3.3 Board Controller

The board controller controls the state of SEAD-3 board. The SEAD-3 Board will be in one of the states at any given time. The controller states are POWER_UP, FPGA_PROGRAM, FPGA_PROGRAM_WAIT, RESET, USB_DOWNLOAD, ACTIVE. Figure 3.1 describes the flow of board state transitions. The SEAD-3 board can be used for normal operations when the controller state is ACTIVE. In the ACTIVE state, the controller monitors requests from the USB download port and processes the requests immediately as shown in the flow diagram. In USB_DOWNLOAD, FPGA_PROGRAM and FPGA_PROGRAM _WAIT states, the peripheral bus is controlled by the Cypress EZ-USB Controller. CPLD drives the selects on CBT isolation switches to give control to the USB bus. The FPGA gains control of the peripheral bus when the board controller returns to the ACTIVE state.

Figure 3.1 SEAD™-3 Board State Transitions



3.4 Reset and Chip Selects

When board is in RESET state, it is held in RESET for 36ms and board becomes ACTIVE if there are no active Reset inputs or USB download request as shown in Fig 3.1. CPLD drives reset pin of FPGAs, Peripheral Bus, PIC32, Ethernet Controller and other devices on Peripheral bus. PIC32 reset is software programmable, see Section 3.8.9 Device Reset Register

CPLD also drives Chip Selects of FPGAs, devices on Peripheral Bus like bootflash, flash, sram, LCD, ethernet controller and PIC32

3.5 Periodic Pulse Generator

CPLD provides access to a periodic pulse generator. Period is fixed to 20 ms, bit 1 in CPLD Status register (see Section 3.8.8 "2-Line 16-Character Alphanumeric LCD ASCII Display Registers") will be high for 10ms and low for 10ms, with a 50% duty cycle.

3.6 LCD Interface Controller

The LCD Module 32610A on the SEAD-3 board is a slower device compared to the operational speed of the CPU. The LCD Interface Controller in CPLD acts a synchronizer and ensures that the CPU does not overflow the display buffer. Table 3.2 shows the Interface Controller functions. Fig 3.2 illustrates state transitions of the LCD controller.

Table 3.2 LCD Interface Functional Table

Function	USB/FPGA Drivers					LCD Bus			
	A4	A3	Read	Write	Data D7-D0	RS	R/W	E	Data D7-D0
LCD write control	0	0	1	0	D[1]	0	0	*[2]	D
LCD write data	0	1	1	0	D	1	0	*	D
LCD Read control D->DR[3]	0	0	0	1	X[4]	0	1	*	D
LCD Read Data D->DR	0	1	0	1	X	1	1	*	D
Illegal, invalid write from USB	1	X	1	0	X	X	0	X	D
All other combinations are also invalid and ignored									

1. D = Data on bus
2. * = Strobe to generate LCD cycle
3. DR = LCD data register in CPLD
4. X = Don't care

Software needs to read the status in LCD Status and LCD contents. Because it is not possible to retrieve the data from the LCD quickly enough for the software, data reads will be done in 2 stages. Software will issue a read to the LCD control/data register, LCD Interface controller in CPLD will generate a read on the LCD bus with RS set appropriately and the data will be captured and written to CPLD LCD Data register (DR). Software should read CPLD LCD Data register (DR) to retrieve LCD control/data value after LCD Interface Controller completes the read cycle. Bit 0 in CPLD LCD Status register will be zero when LCD Interface Controller is IDLE. Software should issue an LCD read/write cycle only when the LCD Interface Controller is IDLE. The LCD Status and Data register in CPLD can be read at any time

3.7 Switches and LEDs

There are registers allocated in the CPLD to sample switches. The switch registers are F-SWITCH, P-SWITCH, NEWSC-LIVE, NEWSC-REG, and NEWSC-CTRL.

The CPLD's P-LED and F-LED registers drive the LEDs on the board.

The switch and LED registers are CPU-accessible, Refer to the Peripheral Bus Register in the *SEAD Basic RTL User's Manual* for a description. Software can only read a switch register, with the exception of NEWSC-CTRL, and software can either read or write an LED register.

3.8 CPLD Registers

3.8.1 P-SWITCH Register

Name: PSWITCH
Address: 0x1F00.0200
Access: RO
Reset Value: N/A

This register allows software to monitor the state of the 8 bit P-SWITCH (S3) on the peripheral bus.

Table 3.3 PSWITCH Register

Bits	Field name	Function	Initial Value
31:8	Reserved		N/A
7:0	VAL	8 P-SWITCH bits (physical switches are numbered 8 to 1). 0: OFF/Open 1: ON/Closed	N/A

3.8.2 F-SWITCH Register

Name: FSWITCH
Address: 0x1F00.0208
Access: RO
Reset Value: N/A

This register allows software to monitor the state of the 8 bit F-SWITCH (S4) on the peripheral bus.

Table 3.4 FSWITCH Register

Bits	Field name	Function	Initial Value
31:8	Reserved		N/A
7:0	VAL	8 F-SWITCH bits (physical switches are numbered 8 to 1). 0: OFF/Open 1: ON/Closed	N/A

3.8.3 P-LED Register

Name: PLED
Address: 0x1F00.0210
Access: R/W
Reset Value: 0x00000000

The PLED register allows software to program the state of the 8 P-LED bits on the peripheral bus.

Table 3.5 PLED Register

Bits	Field name	Function	Initial Value
31:8	Reserved		N/A
7:0	VAL	8 bits corresponding to the 8 P-LED bits. 0: Off 1: On	0x0 (all P-LED bits off)

3.8.4 F-LED Register

Name: FLED
Address: 0x1F00.0218
Access: R/W
Reset Value: 0x00000000

The FLED register allows software to program the state of the 8 F-LED bits on the peripheral bus.

Table 3.6 FLED Register

Bits	Field name	Function	Initial Value
31:8	Reserved		N/A
7:0	VAL	8 bits corresponding to the 8 F-LED bits. 0: Off 1: On	0x0 (all P-LED bits off)

3.8.5 NEWSC-LIVE Register

Name: NEWSC-LIVE
Address: 0x1F00.0220
Access: RO
Reset Value: 0

The NEWSC-LIVE register allows software to read the present state of NEWSC switch. This is an 8-bit wide register.

Table 3.7 NEWSC-LIVE Register

Bits	Field name	Function	Initial Value
7:5	Reserved		0
4	sw_cpld_n	Switch position North	0
3	sw_cpld_e	Switch position East	0
2	sw_cpld_w	Switch position West	0
1	sw_cpld_s	Switch position South	0
0	sw_cpld_c	Switch position Center	0

3.8.6 NEWSC-REG Register

Name: NEWSC-REG
Address: 0x1F00.0228
Access: RO
Reset Value: 0

The NEWSC-REG register allows software to read the state of NEWSC switch after debounce. This is an 8-bit wide register.

Table 3.8 NEWSC-REG Register

Bits	Field name	Function	Initial Value
7:5	Reserved		0
4	sw_cpld_n	Switch position North after debounce	0
3	sw_cpld_e	Switch position East after debounce	0
2	sw_cpld_w	Switch position West after debounce	0
1	sw_cpld_s	Switch position South after debounce	0
0	sw_cpld_c	Switch position Center after debounce	0

3.8.7 NEWSC-CTRL Register

Name: NEWSC-CTRL
Address: 0x1F00.0230
Access: RW
Reset Value: N/A

The NEWSC-CTRL register allows software to switch debounce time. This is an 8-bit wide register.

Table 3.9 NEWSC-CTRL Register

Bits	Field name	Function	Initial Value
7:3	Reserved		N/A
2:0	DEBOUNCE	This field indicates debounce count used in reading NEWSC switch values	0x6

3.8.8 2-Line 16-Character Alphanumeric LCD ASCII Display Registers

ASCII Address Base: 0x1F00.0400

The registers are 8 bits wide. and are used to display characters. LCD Status and Data register reside in the CPLD.

Table 3.10 ASCII Display Registers

Name	Offset Address	Access	Function
LCD Read/Write Control	0x0000.0000	R/W	LCD read/write control register
LCD Read/Write Data	0x0000.0008	R/W	LCD read/write data register
CPLD LCD Status	0x0000.0010	RO	8-bit register. Bit 7 reflects LED data port bit 7, bit 1 is a 10ms pulse used by the Linux kernel to calculate CPU frequency, bit 0 is the BUSY bit: when set to 1, it indicates that the LCD controller is busy processing a read/write transaction
CPLD LCD Data	0x0000.0018	RO	ASCII character in position 3

See the documentation from HP for additional information on how to program the HDSP-2532 ASCII display.

3.8.9 Device Reset Register

Name: PI_DEVRST
Address: 0x1F00.0480
Access: WO
Reset Value: 0x0

The CPU/USB can reset pic32 by writing 0x01 to the PI_DEVRST register, and pic32 can be brought out of reset by writing 0x0 to the PI_DEVRST register.

Table 3.11 PI_DEVRST register

Bits	Field name	Function	Initial Value
31:1	Reserved		N/A
0	PIC32_RST	0 = assert PIC32 reset 1= deassert PIC32 reset	0

Chapter 4

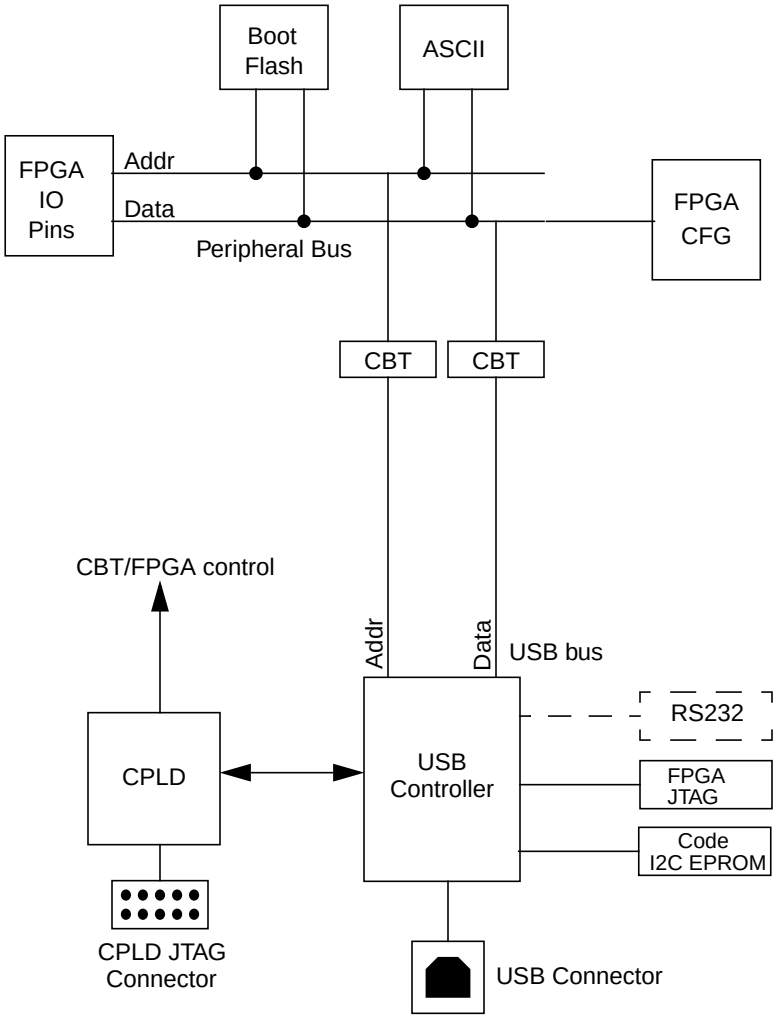
USB Download

The USB Download feature of the SEAD-3 is used to program the boot Flash containing the boot software (6 MBytes) and the FPGA configuration. There is sufficient storage for one LX155 and two LX330 size FPGAs (26 MBytes). This section describes the board hardware involved in USB download and the format used.

4.1 USB Hardware

The USB controller on the board is an EZ-USB device from Cypress Semiconductor. Firmware for the device is located in an on-board serial EPROM. The figure below illustrates the components on the board that are used in USB download.

Figure 4.1 USB Download



The boot Flash and the LCD display, which are both on the peripheral bus, are the main targets of USB accesses. The peripheral bus is normally connected to the FPGA so the controller in the peripheral can access the bus. However, when USB download is enabled, the bus is disconnected from the FPGA by tri-stating, and the CBT switches connect the USB controller to the peripheral bus. To boot the FPGA, the board must be configured with the CBT switches OFF. Data will flow from the boot Flash through to the FPGA.

The USB controller has its own private serial port. This is only intended for debug of the USB firmware, so in most cases the user can ignore this port.

The CPLD controls the reset signals, based on the FPGA mode of operation.

Table 4.1 FPGA Configuration Modes

Mode	Comment
USB download	In this mode, the USB controller takes the role as master on the peripheral bus. It can access the ASCII display and the boot Flash.
FPGA config	This is the mode in which the FPGA is configured. In this mode, the USB controller reads data from the system Flash and writes it to the configuration data port of the FPGA. This state is entered after a reset or after a USB download. NOTE: We recommend the USB download for configuring the FPGAs, because the config data is stored in flash and will be written on power-up.
Normal operation	The FPGA is in control of the peripheral bus.

When the board powers on, it enters a reset state. The board will remain in reset until the FPGA is successfully configured, either by data from the boot Flash, or by cable download.

When the USB controller detects a file download, it enters the USB download state, resets the board, and keeps the board in reset until all data has been downloaded. Following the download, if the FPGA is configured for Flash boot, the FPGA is re-configured.

Data sent to the board will have a start command (!R) indicating the start of data and a stop command indicating the end of data (>#DL_DONE). Note that after the start command is received, the board will remain in the USB download state until the end command is received. So if the data download is aborted, the board will stay in the download state.

4.2 USB Download Format

The SEAD-3 board presents itself to the USB host as a bi-directional printer device. As a device in the printer class, the SEAD-3 board can use existing printer drivers (for example, in Linux and Windows) to access the board. In addition to the control endpoint, the board supports one bidirectional high-speed (12 Mbit/s) bulk endpoint:

Table 4.2 USB Endpoints on SEAD-3 Board

Endpoint #	Direction (seen from host)	Type
0	IN	Control
0	OUT	Control
2	IN	Bulk
2	OUT	Bulk

Endpoint 0 is the standard control endpoint and is used, for example, as device descriptors and stall/un-stall endpoints. Endpoint 0 supports all standard requests defined by the USB 1.1 standard and the additional requests defined for printer-class devices.

Endpoint 2 is a bidirectional bulk endpoint used for data transfer. The host will use the bulk-out pipe to send data to the board. The use of the bulk-in pipe is optional, as described below.

4.2.1 Sending Data to the Board

Data is sent to the board through the bulk-out pipe. The exact method used to access the bulk-out pipe depends on the operating system. For Linux, the user can issue a command similar to:

```
cat xx.fl > /dev/usblp0
```

where xx.fl contains the data to send to the board and "`/dev/usblp0`" is the device interface for the bulk pipe.

For Windows the user must open the file in the Wordpad editor and "print" it to the port representing the SEAD-3 board. We suggest that all `.fl` files are associated with the Wordpad editor to assure that this editor is used to open the files.

See Ref [1] for more information on driver installation for Windows and Linux.

The file sent to the board is a pure text file containing ASCII characters. Keep the number of characters per line below 40 characters to avoid line splitting when Windows assumes it prints to a true printer. All lines must be terminated by line feed (LF) or carriage return/line feed (CR/LF). The file contents is case insensitive.

The boot Flash device is organized in sectors of 128 Kbytes. "Erase" and "Set Lock Bit" commands operate on exactly one sector, this being the sector currently addressed. After the last block of 16 words in a sector are written into flash, the address counter has advanced to the next sector. This implies that a Set Address (@) to the sector has to be executed before a Set Lock Bit command (!S) can be issued.

The file to be loaded into the Flash via USB contains 3 types of elements: Commands, data and separators, as described in Table 4.3.

Table 4.3 Download File Contents

Type	Description
Command	A command is build from an opcode and in some cases and argument, see next table.
Separator	Separators are used to separate commands and/or data. One or more of the following are valid separators: space, tab, LF or CR-LF.
Data	A 32-bit value like 11223344. Data must appear in blocks of 16 starting on a 16-word boundary. The boot Flash is 8 bits wide, and the 32 bits are stored in Big Endian format, so the value 11223344 is stored with 11 at the lowest address and 44 at the highest address.

A number of opcodes are used to control code download and Flash memory handling, as described in Table 4.4.

Table 4.4 Download Commands

Opcode	Meaning	Argument
@	Sets current writing/erasing address to SEAD-3 Board physical memory map format. Addresses must be on 16-word boundaries.	32-bit address, 8 characters
!R	Reset download system and enter download mode.	No
!E	Erase the current Flash sector (128 KB).	No
!C	Clear all Flash lock bits.	No
!S	Set current Flash sector lock bit.	No
#	Comment (rest of line).	A string of ASCII values between 0x20 and 0x7f except the '!' character. The controller continuously looks for the '!' character to get in sync if some error occurs (!R is a reset). So don't use this character in comments and display strings.
>	Print command (shows next 8 characters in ASCII display, the command needs exactly 8 non-white space characters). Please note that display strings starting with # will be interpreted as commands as well, as specified in Table 4.5.	A string of exactly 8 ASCII values between 0x20 and 0x7f, except the '!' character.
data	Data has to be in blocks of 16 words, without interruption of any Comments (#) and Print Commands (>).	No

Display commands in which the string starts with # will be displayed, but will also be interpreted as commands according to Table 4.5.

Table 4.5 Special Display Commands

Display string	Meaning
#DL_DONE	This string brings the board out of USB download mode. This display string should be the last line in all download files.
#GET_MAN	This string will place the board in a mode where a small manual can be retrieved by reading the USB port.
#<others>	Reserved.

Example of code download format:

```
#Example
!R
@1fc00000
!E
12345678 23456789 3456789A 456789AB
56789ABC 6789ABCD 789ABCDE 89ABCDEF
9ABCDEF0 ABCDEF01 BCDEF012 CDEF0123
DEF01234 EF012345 F0123456 01234567
# always 16 words in a block
>#DL_DONE
```

The example will reset the download system, erase the sector starting at 1fc00.0000, and write 16 words starting at this address. Finally, the board will return to normal operation due to the >#DL_DONE display command.

If an error should occur during Flash download, the ASCII display will show an error message, as described in Table 4.6.

Table 4.6 Flash Download Error Messages

Message	Meaning
Ill cmd	Illegal command received, e.g. “R” is received (not !R)
Ill !cmd	Illegal “!” command received, e.g. “!A” is received
Ill hex	Illegal hex received (in data or addr), e.g. “ABCDEFGH” both “G” and “H” is illegal characters
Hex exp	Hex expected (always data blocks of 16 words). Happens if e.g. a comment (#) is received in the middle of a block of 16 words.
Era susp	Block erase suspended
Err era	Error in block erasure or clear lock-bits
Err prog	Error in programming or set block lock-bits
Low volt	Low programming voltage detected
Lock det	Master lock-bit, Block lock-bit or RP# lock detected

If an error occurs, the USB controller ignores all data until the next !R command. A !R command will always reset the download system, regardless of state, even if it occurs in the middle of a data stream.

The following commands will always bring the board out of download mode, regardless of the previous state:

```
# Get in sync.
!R
# Back to normal operation
>#DL_DONE
```

The boot Flash device is 32 MBytes (see Table 4.7). In programming mode, the USB controller can access all 32 Mbytes. The USB controller uses 32-Kbyte pages together with a page address register to access the full 32 Mbytes.

Table 4.7 Boot Flash Layout (USB Controller)

Table 1.

Area	Size	Usage
1e00.0000 -> 1e7f.ffff	8 MBytes	Contains FPGA configuration.
1e80.0000 -> 1eff.ffff	8 MBytes	Contains FPGA configuration
1f00.0000 -> 1f7f.ffff	8 MBytes	Contains FPGA configuration
1f80.0000 -> 1fff.ffff	8 MBytes	Contains FPGA configuration and boot SW, e.g., YAMON.

When the CPU has control of the bus only the top 8 MBytes can be seen, as listed in Table 4.8.

Table 4.8 Boot Flash Layout (FPGA)

Area	Size	Usage
1f80.0000 -> 1f9f.ffff	2 MBytes	Contains FPGA configuration. **DO NOT OVERWRITE.**
1fa0.0000 -> 1fff.ffff	6 MBytes	Contains boot SW, e.g., YAMON.

Chapter 5

EJTAG / PDTrace / iFlowtrace

Depending on the use model of the SEAD-3 board, the CPU may either be placed on the small CPU module, or if the debug connectors are required, it can be placed on the FPGA_module-3 by loading the CPU along with the BRTL into the SEAD-3 FPGA.

There are three debug connectors on SEAD-3:

- EJTAG
- PDTrace
- iFlowtrace

On the EJTAG connector (J21), one pin is intentionally removed (#12) from the connector to assure correct orientation with the probe header. The connector signals are routed directly to the EJTAG interface on the CPU core. The only exception is the EJTAG_RST_N signal, which is routed to the CPLD where it can issue a board reset. See Reference [7] for additional information on EJTAG.

Table 5.1 EJTAG Connector (J21)

Pin	Signal	Description
1	EJTRSTN	Test Reset Input
3	EJTDI	Test Data Input
5	EJTDO	Test Data Output
7	EJTMS	Test Mode Select Input
9	EJTCK	Test Clock Input
11	EJRST_N	System Reset
13	EJDINT	Debug Interrupt
2,4,6,8,10	GND	Ground
14	VIO	Voltage Sense for I/O
12	NC	NO Pin

The iFlowtrace connector (J20) is a 10- pin 0.1” IDC header connected directly to the FPGA. See Ref [9] for additional information on iFlowtrace.

Table 5.2 iFlowtrace Connector (J20)

Pin	Signal	Description
1	IF_TRCLK	Trace Clock
3	IF_TRDATA0	Trace Data 0
5	IF_TRDATA1	Trace Data1

Table 5.2 iFlowtrace Connector (J20)

Pin	Signal	Description
7	IF_TRDATA2	Trace Data2
9	IF_TRDATA3	Trace Data 3
2,4,6,8,10	GND	GROUND

The PDtrace connector (J22) is a 38-pin Mictor connector connected directly to the FPGA. See Ref [10] for additional information on PDTrace.

Table 5.3 PDTrace Connector (J22)

Pin	Signal	Description
3	TR_PROBE_EN	Trace Probe Enable
5	TR_CLK	Trace Clk
7	TR_DATA15	Trace Data 15
9	TR_DATA14	Trace Data 14
11	TR_DATA13	Trace Data 13
13	TR_DATA12	Trace Data 12
15	TR_DATA11	Trace Data 11
17	TR_DATA10	Trace Data 10
19	TR_DATA9	Trace Data 9
21	TR_DATA8	Trace Data 8
23	TR_DATA7	Trace Data 7
25	TR_DATA6	Trace Data 6
27	TR_DATA5	Trace Data 5
29	TR_DATA4	Trace Data 4
31	TR_DATA3	Trace Data 3
33	TR_DATA2	Trace Data 2
35	TR_DATA1	Trace Data 1
37	TR_DATA0	Trace Data 0
2	NC	No Connect
4	EJ_VIO	Voltage Sense for I/O
6	TR_CLK	Trace Clk
8	TR_TCK	Trace Clock Input
10	TR_TMS	Trace Test Mode Select
12	TR_TDI	Trace Test Data Input
14	TR_TDO	Trace Test Data Output
16	TR_TRSTN	Trace Reset Input
18	TR_RSTN	System Reset
20	TR_DINT	Trace Debug Interrupt
22	TR_DM	

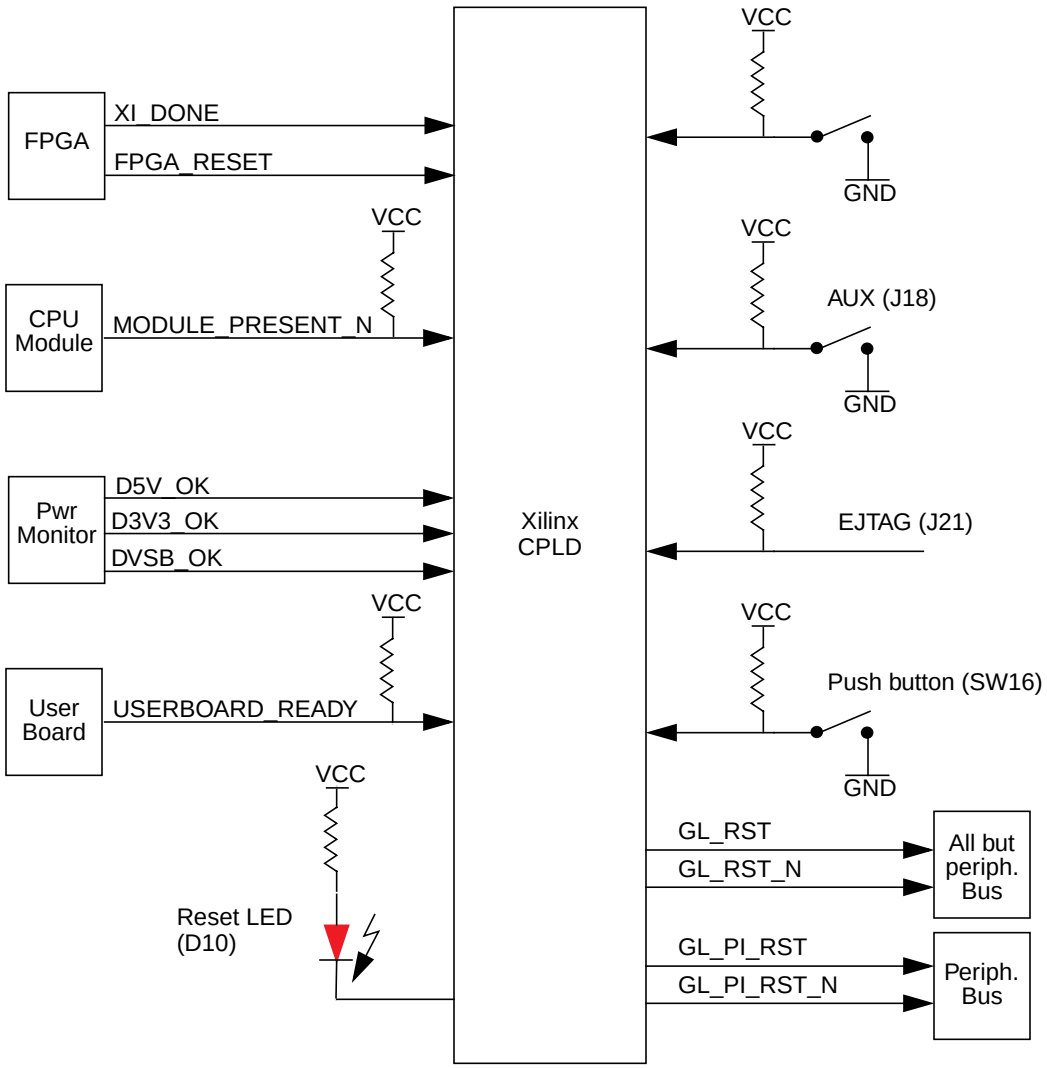
Table 5.3 PDTrace Connector (J22) (Continued)

Pin	Signal	Description
24,26,28,30,32,34	NC	No Connect
36	TR_TRIGOUT	Trace Trigger Out
38	TR_TRIGIN	Trace Trigger In

Reset Operation

The reset signals on the board are generated by a Xilinx CPLD from a number of input signals, as shown in Figure 6.1. A mono-stable circuit guarantees that reset is asserted at least 35 ms.

Figure 6.1 Reset Conditions



All the input signals except MODULE_PRESENT_N can cause a reset. The intention of MODULE_PRESENT_N was to keep the board in reset if no CPU module was present, but this signal is intensionally ignored in the EPLD to allow implementation of the CPU in the FPGA.

The EPLD generates four reset output signals; GL_RST, GL_PI_RST (active high) and GL_RST_N, GL_PI_RST_N (active low). The GL_PI_RST/GL_PI_RST_N signals are used to reset units on the peripheral bus, while the other two reset signals are used for the rest (user board, CPU, FPGA etc.). The two sets of reset allows the USB controller to access the peripheral bus for download and in the meantime keep the rest of the board in reset.

The reset state is observed on the red LED D10. When “ON”, GL_RST/GL_RST_N is asserted. Table 6.1 summaries the reset conditions and contains references to the schematic sheets where the reset signals are generated.

Table 6.1 Reset Conditions

Unit	Reset when:	Signal	Description
FPGA	Low	XI_DONE[2:0] (DONE pins of the FPGAs)	The FPGAs assert their DONE signals high when the FPGA is successfully configured. The FPGA validates the CRC on the configuration data stream and if it fails it will not assert DONE. The combined DONE state can be observed on LED D8 (CONF. DONE).
	High	FPGA_RESET	This signal is driven by a normal IO pin on the FPGA. It allow the user to implement circuitry that can reset the board.
CPU Module	High	MODULE_PRESENT_N	This signal is driven low by the CPU module when it is present and ready for reset to be removed. The EPLD ignores this signal to allow implementation of a CPU in the FPGA.
Pwr. Monitor	Low	D5V_OK	SIgnal asserted when the 5V supply is within +-5% range
	Low	D3V3_OK	Signal asserted when the 3.3V supply is within +-5% range
	Low	DVSB_OK	Signal asserted when 3V3 and 5V standby voltage is within +-5% range.
User Board	Low	USERBOARD_READY	If the user board contains circuit that must boot or otherwise prepare itself before reset is removed then it can use the USERBOARD_READY to indicate when it is ready.
AUX	Low	AUX_RESET_N	This is a jumper that is available for user purpose. A reset will occur when the two pins in the jumper are shorted. Connect to an emulator or similar to achieve remote reset.
EJTAG	Low	EJTAG_RST_N	Signal is generated by the EJTAG probe when connected to the probe header J21.
Push Button	Low	ATX_RESET_N	A reset is generated when the user pushes the SW16 button at the front of the board.

When power is initially turned on to the board, it will be in the reset state. Note that if the board is setup to download FPGA configuration via cable, then the board will remain in reset until download is complete. Otherwise the FPGA will boot from the boot Flash.

Chapter 7

Clocking

The board has three clock synthesizers. The following clocks are routed to the FPGA:

- Synthesizer 0: (Selectable)
    - GL_LV_CLK0: Clock output
    - GL_A3_LV_CLK0: Clock output
- Synthesizer 1: (Selectable)
    - GL_LV_CLK1: Clock output
    - GL_A3_LV_CLK1: Clock output
    - PIC_CLK_20MHZ: 20 MHz fixed frequency clock to drive the PIC32
- Synthesizer 2: (Fixed)
    - USBPHY_CLK_24MHZ:24 MHz clock to drive the USB Phy
    - USB_CLK_24MHZ:24 MHz clock to drive the EZUSB Microcontroller
    - ETH_CLK_25MHZ:25 MHz clock to drive the Ethernet controller
    - GL_CLK_24_576:24.576 MHz clock (Not currently used)
    - GL_LV_CLK_P:P-Side of the 200MHz differential clock for the DRAM controller
    - GL_LV_CLK_N: N-Side of the 200MHz differential clock for the DRAM controller
    - Module Clock:
    - GL_MOD_LV_CLK: Clock source for the FPGA_Module-3. It can be derived from either Synthesizer 0 or Synthesizer 1

The clock source for GL_MOD_LV_CLK is controlled by a clock selection buffer which selects from an output of Sysnthesiser 0 or 1, using SW1[8].

The GL_MOD_LV_CLK output can also be disabled using SW1[7].

The three synthesizers have a daisy-chained clock source derived initially from a 25MHz oscillator, which is fed into Synthesizer 1 and then buffered and fed to Synthesizer2 and again to Synthesizer 3.

All clocks generated are 3.3V except the differential clocks, which are 2.5V.

Table 7.1 GL_MOD_LV_CLK - Selection Table

SW1-[8]	SW1-[7]	Description
on	off	Synthesizer 0 range
off	off	Synthesizer 1 range
x	on	Output disabled

Table 7.2 Synthesizer 0 - Switch Settings

SW1-[6]	SW1-[5]	SW1-[4]	Power
on	on	on	200.00 MHz
on	on	off	187.50 MHz
on	off	on	175.00 MHz
on	off	off	166.66 MHz
off	on	on	150.00 MHz
off	on	off	133.33 MHz
off	off	on	125.00 MHZ
off	off	off	100.00 MHz

Table 7.3 Synthesizer 1 - Switch Settings

SW1-[3]	SW1-[2]	SW1-[1]	Power
on	on	on	133.33 MHz
on	on	off	125.00 MHz
on	off	on	100.00 MHz
on	off	off	83.33 MHz
off	on	on	75.00 MHz
off	on	off	66.66 MHz
off	off	on	50.00 MHZ
off	off	off	40.00 MHz

Chapter 8

DRAM Interface

The DRAM socket on the board accepts a standard DDR II -533 1.8V DDR II DRAM module. DRAM is not available on the LX50-equipped boards The controller is built into the FPGA.

The DRAM interface signals on the FPGA are:

- MC_CTRL: DRAM control pins. See Table 8.1 below.
- MC_DQ[63:0], MC_DQMB[7:0]: DRAM data pins.
- MC_A[13:0], MC_BA[2:0]: DRAM address.

The MC_CTRL pin group on the FPGA contains the following members:

Table 8.1 SD_CTRL Group

Member Name
MC_RAS_N
MC_CAS_N
MC_WE0_N
MC_CS[1:]_B
MC_CKE[1:0]
MC_CLK[1:0]_P
MC_CLK[1:0]_N
MC_DDT[1:0]
MC_SDA
MC_SCL

NOTE: Currently we only support single-sided DRAM modules with a maximum size of 512 MB. Only 432MB is accessible by software.

Chapter 9

Peripheral Bus

The peripheral bus on the board is used primarily to access the on-board resources such as UARTs, Flash SRAM, etc. Use the bus to access the on-board resources only, and ignore the signals on the user connector.

- Use the bus to access on-board resources and resources on the user board. When required, signal members can be added to the bus using the uncommitted FPGA pins. The user can change the timing and use different timing for the on-board resources and resources on the user board.

- When the EzUSB controller has the bus, it is used to program the FPGA and to display status messages on the LCD display.

The resistors on the peripheral bus assure that all units are disabled when not in use. The peripheral bus is a 3.3V bus and is **NOT** 5V-tolerant.

The CBT block are controlled by the CPLD in conjunction with the EzUSB controller, as shown in Table 9.1.

Table 9.1 Peripheral Bus Switch Configuration

Mode	USB2PI_EN_N	USB2PI_EN
Peripheral bus is disconnected from FPGA. Pins are available to the USB controller.	Lo	Hi
Peripheral bus is connected to FPGA.	Hi	Lo

The following signals are part of PI_CTRL group on the FPGA:

Table 9.2 PI_CTRL[11:0] Group

Member name	Function
PI_SEL[4:0]	Unit select. These 5 bits define the unit to access according to the list below. PI_SEL must be stable while one or more of the signals PI_RD_N, PI_CS_N or PI_WE_N are asserted. 00000: System Flash 00001: Ethernet controller 00010: SRAM Bank low 00011: SRAM Bank high 00100: UART 0 00101: UART 1 00110: LCD Display 00111: PIC Microcontroller 01000: LED blocks (P & F) 01001: DIP switches (P & F) 01010: Revision code 01011: Boot Flash 01100: NEWSC switches 01101: Unused, Reserved 01110: Unused, Reserved 01111: Unused, Reserved 11000: USER board
PI_RD_N	Read strobe.
PI_CS_N	Common chip select.
PI_BE_N[3:0]	Write enables. PI_BE_N[3] corresponds to PI_D[31:24] PI_BE_N[2] corresponds to PI_D[23:16] PI_BE_N[1] corresponds to PI_D[15:8] PI_BE_N[0] corresponds to PI_D[7:0]
PI_UART0_INT_N	Interrupt from UART0
PI_UART1_INT_N	Interrupt from UART1

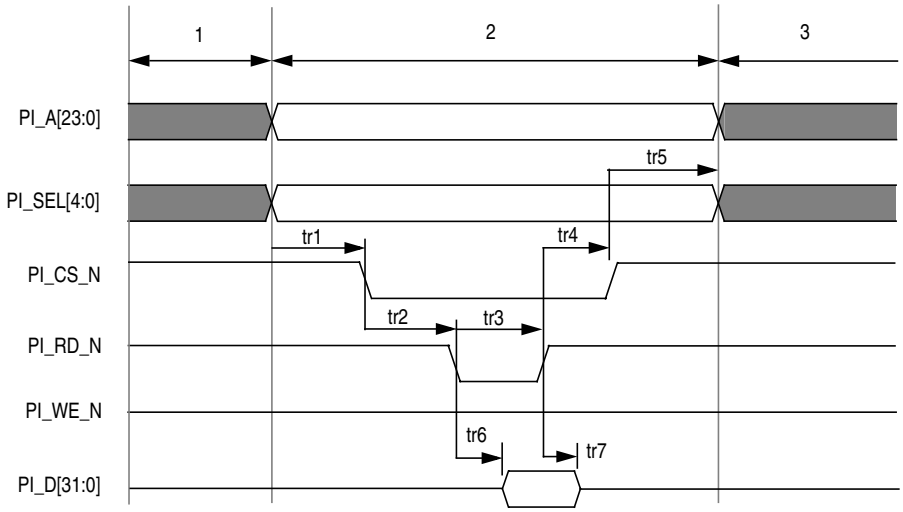
9.1 Access Timing

The peripheral bus is a simple asynchronous bus with no external ready for access termination. All on-board resources are asynchronous devices. The user may extend the bus to the user board and add additional signals such as acknowledge/ready as required. The user may even add synchronous devices, because all clock inputs to the FPGA are available on the user connector as well.

9.1.1 Read Access

The basic peripheral bus read access pattern is shown in Figure 9.1.

Figure 9.1 Read Access



The events are:

- Setup a valid address and unit select vector (PI_A[13:0], PI_SEL[4:0]).
- Assert chip select PI_CS_N. PI_CS_N is used together with PI_SEL[4:0] to generate a chip select to the addressed unit. For units not using chip, select PI_SEL[4:0] is used with PI_RD_N to generate a unit specific read strobe.
- Then assert the read strobe PI_RD_N, and the addressed unit will drive data on the peripheral data bus PI_D[31:0].
- The controller must sample read data when PI_RD_N is de-asserted.
- After PI_RD_N is de-asserted, the controller may de-assert PI_CS_N and finally when PI_CS_N is de-asserted the controller can change PI_A[13:0] and PI_SEL[4:0].

Safe values for the parameters in the figure above are listed in Table 9.3. This is by no means the ideal timing for all units on the bus. Some units are fast like SRAM, while others are slow like the boot Flash. However, the values below, which represent a kind of overall worst-case timing, will guarantee timing for all units.

Table 9.3 Safe Read Timing Parameters

Symbol	Signals	Description	min	max	Unit
tr1	PI_A[23:0], PI_SEL[4:0], PI_CS_N	Setup time of address and unit select vector before chip select assertion.	5	-	ns
tr2	PI_CS_N, PI_RD_N	Chip select asserted to read asserted.	5	-	ns
tr3	PI_RD_N	Read strobe pulse width.	200	-	ns
tr4	PI_RD_N, PI_CS_N	Chip select hold after read strobe de-assertion.	5	-	ns

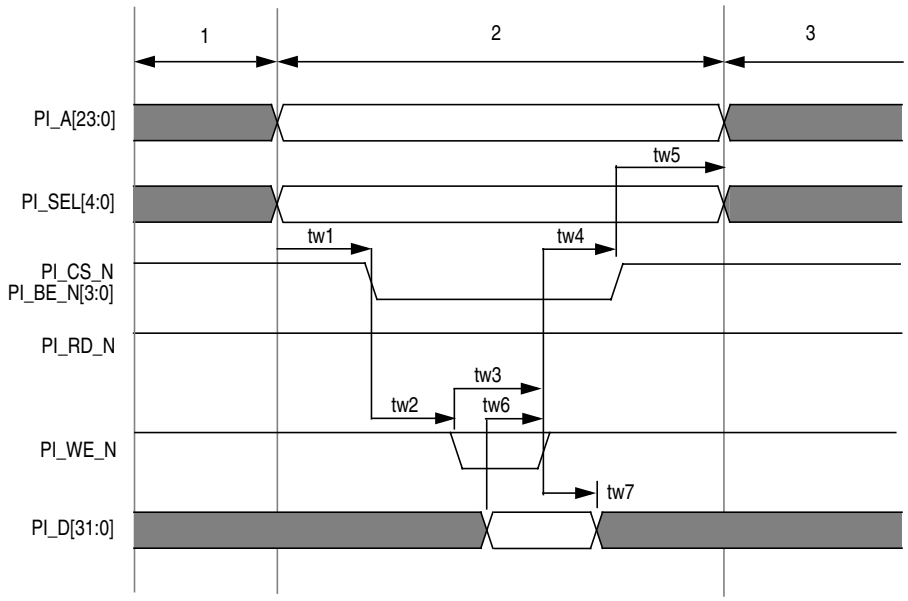
Table 9.3 Safe Read Timing Parameters (Continued)

Symbol	Signals	Description	min	max	Unit
tr5	PI_A[23:0], PI_SEL[4:0], PI_CS_N	Hold time of address and unit select vector after chip select de-assertion.	3	-	ns
tr6	PI_RD_N, PI_D[31:0]	Read data valid after read strobe assertion.	-	170	ns
tr7	PI_RD_N, PI_D[31:0]	Read data hold after read strobe de-assertion.	0	-	ns

9.1.2 Write Access

Figure 9.2 shows the basic access pattern for a write operation to the on-board resources.

Figure 9.2 Write Access



The events are:

- Setup a valid address and unit select vector (PI_A[13:0], PI_SEL[4:0]).
- Assert chip select PI_CS_N. PI_CS_N is used together with PI_SEL to generate a chip select to the addressed unit. For units not using chip select PI_SEL[3:0] is used with PI_WE_N to generate a unit specific write strobe.
- Then assert the write strobe(s) PI_BE_N[3:0]. Valid write data must meet the setup and hold time with respect to the rising edge of the write strobe(s). All combinations are valid even the case where no write strobes are asserted.
- After PI_BE_N[3:0] is de-asserted, the controller may de-assert PI_CS_N and finally when PI_CS_N is de-asserted the controller can change PI_A[13:0] and PI_SEL[4:0].

Safe values for the parameters in the figure above are listed below. Again, these are very conservative values guaranteed to work for all units.

Table 9.4 Write Timing Parameters

Symbol	Signals	Description	min	max	Unit
tw1	PI_A[23:0], PI_SEL[4:0], PI_CS_N	Setup time of address and unit select vector before chip select assertion	5	-	ns
tw2	PI_CS_N, PI_BE_N[3:0]	Chip select asserted to write strobe(s) asserted	5	-	ns
tw3	PI_BE_N[3:0]	Write strobe(s) pulse width	100	-	ns
tw4	PI_BE_N[3:0], PI_CS_N	Chip select hold after write strobe(s) deassertion	5	-	ns
tw5	PI_A[23:0], PI_SEL[4:0], PI_CS_N	Hold time of address and unit select vector after chip select de-assertion	5	-	ns
tw6	PI_D[31:0], PI_BE_N[3:0]	Write data setup to write strobe deassertion	20	-	ns
tw7	PI_D[31:0], PI_BE_N[3:0]	Write data hold after write strobe deassertion	10	-	ns

9.2 Peripheral Bus Resources

The table below summaries the characteristics of the peripheral bus resources. The user should obtain data sheets from the vendor for detailed information. The “PI_SEL[4:0]” column is the value used to access the unit on the peripheral bus. The “Access” column shows the allowed operations (RW or RO).

Table 9.5 Peripheral Bus Resources

Resource	PI_SEL[4:0]	Access Type	Description
System Flash	00000	RW	The Flash blocks contains two chips, and each device is 16 bits wide. Each is connected to one half of the 32-bit bus. Only 32-bit accesses are allowed. The address bits used are PI_A[24:2]. The size of the Flash block is 32 MBytes.
Ethernet Controller	00001	RW	Ethernet Command / Data interface.
SRAM Low	00010	RW	Each of the two SRAM blocks contains two chips in parallel for a 32-bit data width. Writes of 0, 1, 2, 3, or 4 bytes are allowed. The address bits used are PI_A[23:2].
SRAM High	00011		
UART0	00100	RW	There is a dual 16C2550 UART. The UART uses address bits PI_A[4:2], i.e., addresses are word aligned. The UART uses PI_D[7:0] for data. One channel of the UART is connected to a 9-way Dtype via a level RS232 level shifter. The other channel is connected to an FT232RL controller to provide USB to Serial conversion.
UART1	00101		

Table 9.5 Peripheral Bus Resources (Continued)

Resource	PI_SEL[4:0]	Access Type	Description
LCD Display	00110	RW	The display is driven by a state machine in the CPLD. This allows the complex timing for the display to be handled in hardware.
PIC32 IO Controller	01111	RW	Command / Data interface to the PIC32 IO subsystem.
LEDs F_LED = D20-D27 P_LED = D12-D19	01000	RW	The Bar LEDs consist of two bank of 8 LEDs. They are driven by the CPLD, which contains programmable registers that allow the user to switch each LED to an on or off state. After reset, the register is cleared and all LEDs are off.
DIP Switches F_SWITCH = SW3 P_SWITCH = SW4	01001	RO	The DIP switches are connected to the CPLD which contains registers to allow the user to read the state of the switches. All switch lines are pulled low when the switch is open. A closed switch will result in the user reading a “1”.
Revision code	01010	RO	The revision code is read through another register within the CPLD. Bits 7:4 indicates the board ID, which is 0001 for SEAD-3. Bit 3:0 indicates the board revision, staring with 0000.
Boot Flash	01011	RW	The boot Flash contains the boot SW and the FPGA configuration, as described in Section 4.2.1, "Sending Data to the Board". The device is organized as 32M x 8 bits. There are 3 FPGA config blocks and 1 boot S/W block.
NEWSC North = SW5 East = SW8 West = SW6 South = SW11 Center = SW7	01100	RW	Keypad switches
Reserved	01101	RW	Unused, Reserved
Reserved	01110	RW	Unused, Reserved
Reserved	01111	RW	Unused, Reserved
User Expansion	11000	R/W	This PSEL is reserved for User I/O added via the SEAD-3 Expansion Connector.

Chapter 10

SEAD™-3 CoreBus Connector

The CoreBus interface defined in Table 10.1 is the interface between the CPU and the FPGA. The SEAD-3 board is designed to accept a module with a variety of CPU cores.

The bus is not defined as any particular bus type, since the user may define any of the 264 signals to be any signal of a desired bus, as long as both ends match.

MIPS has defined three bus types for various MIPS cores: EC bus, OCP bus, and AHB Lite.

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B2	E29	A3
CORE_B3	F26	A4
CORE_B4	G31	A5
CORE_B5	H29	A7
CORE_B6	H27	A8
CORE_B7	K24	A9
CORE_B8	K27	A10
CORE_B9	L26	A12
CORE_B10	M27	A13
CORE_B11	P29	A14
CORE_B12	P24	A15
CORE_B13	R24	A17
CORE_B14	T24	A18
CORE_B15	U25	A19
CORE_B16	AF8	A20
CORE_B17	AC9	A22
CORE_B18	W31	A23
CORE_B19	Y28	A24
CORE_B20	AA28	A25
CORE_B21	AA31	A27
CORE_B22	AC28	A28
CORE_B23	AC29	A29
CORE_B24	AD30	A30

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference (Continued)

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B25	AJ27	A32
CORE_B26	AG31	A33
CORE_B27	AG30	A34
CORE_B28	AC24	A35
CORE_B29	AG25	A37
CORE_B30	AH32	A38
CORE_B1	D34	A2
CORE_B31	AK32	A39
CORE_B32	AP32	A40
CORE_B33	C34	B1
CORE_B34	F29	B2
CORE_B35	E27	B4
CORE_B36	G26	B5
CORE_B37	G30	B6
CORE_B38	J26	B7
CORE_B39	K32	B9
CORE_B40	K28	B10
CORE_B41	M26	B11
CORE_B42	M28	B12
CORE_B43	P30	B14
CORE_B44	P27	B15
CORE_B45	T34	B16
CORE_B46	T25	B17
CORE_B47	U26	B19
CORE_B48	AE8	B20
CORE_B49	AD9	B21
CORE_B50	W30	B22
CORE_B51	Y31	B24
CORE_B52	AA24	B25
CORE_B53	AA26	B26
CORE_B54	AC27	B27
CORE_B55	AC30	B29
CORE_B56	AE31	B30
CORE_B57	AK26	B31
CORE_B58	AF31	B32

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference (Continued)

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B59	AH29	B34
CORE_B60	AC25	B35
CORE_B61	AF24	B36
CORE_B62	AG32	B37
CORE_B63	AJ32	B39
CORE_B64	AN32	B40
CORE_B65	D32	C2
CORE_B66	F30	C3
CORE_B67	E26	C4
CORE_B68	G25	C5
CORE_B69	H30	C7
CORE_B70	J27	C8
CORE_B71	K33	C9
CORE_B72	K26	C10
CORE_B73	M25	C12
CORE_B74	N27	C13
CORE_B75	P31	C14
CORE_B76	P26	C15
CORE_B77	R27	C17
CORE_B78	T26	C18
CORE_B79	U28	C19
CORE_B80	AJ9	C20
CORE_B81	AE11	C22
CORE_B82	AD7	C23
CORE_B83	W27	C24
CORE_B84	Y27	C25
CORE_B85	Y24	C27
CORE_B86	AA25	C28
CORE_B87	AB27	C29
CORE_B88	AC33	C30
CORE_B89	AD31	C32
CORE_B90	AF28	C33
CORE_B91	AJ29	C34
CORE_B92	AH30	C35
CORE_B93	AE26	C37

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference (Continued)

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B94	AH25	C38
CORE_B95	AK33	C39
CORE_B96	AN33	C40
CORE_B97	C32	D1
CORE_B98	F34	D2
CORE_B99	F28	D4
CORE_B100	G28	D5
CORE_B101	H24	D6
CORE_B102	J25	D7
CORE_B103	K34	D9
CORE_B104	L25	D10
CORE_B105	N29	D11
CORE_B106	N32	D12
CORE_B107	M31	D14
CORE_B108	P25	D15
CORE_B109	R26	D16
CORE_B110	T29	D17
CORE_B111	U27	D19
CORE_B112	AH9	D20
CORE_B113	AD11	D21
CORE_B114	AE7	D22
CORE_B115	W26	D24
CORE_B116	Y26	D25
CORE_B117	Y29	D26
CORE_B118	AA30	D27
CORE_B119	AB25	D29
CORE_B120	AC32	D30
CORE_B121	AD24	D31
CORE_B122	AE28	D32
CORE_B123	AK29	D34
CORE_B124	AJ30	D35
CORE_B125	AE27	D36
CORE_B126	AJ25	D37
CORE_B127	AK34	D39
CORE_B128	AN34	D40

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference (Continued)

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B129	V28	E40
CORE_B130	AB11	F24
CORE_B131	AB10	F25
CORE_B132	AB8	F26
CORE_B133	V24	F36
CORE_B134	V25	F37
CORE_B135	V30	F39
CORE_B136	V27	F40
CORE_B137	C33	G2
CORE_B138	E33	G3
CORE_B139	E28	G4
CORE_B140	G27	G5
CORE_B141	H25	G7
CORE_B142	J24	G8
CORE_B143	K29	G9
CORE_B144	L24	G10
CORE_B145	M30	G12
CORE_B146	N34	G13
CORE_B147	N30	G14
CORE_B148	R33	G15
CORE_B149	R31	G17
CORE_B150	T28	G18
CORE_B151	U30	G19
CORE_B152	AF9	G20
CORE_B153	AF11	G22
CORE_B154	AC7	G23
CORE_B155	W25	G24
CORE_B156	W29	G25
CORE_B157	Y34	G27
CORE_B158	AA29	G28
CORE_B159	AB28	G29
CORE_B160	AD34	G30
CORE_B161	AD25	G32
CORE_B162	AH28	G33
CORE_B163	AK31	G34

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference (Continued)

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B164	AF30	G35
CORE_B165	AF26	G37
CORE_B166	AJ26	G38
CORE_B167	AH33	G39
CORE_B168	AM32	G40
CORE_B169	B33	H1
CORE_B170	E32	H2
CORE_B171	F25	H4
CORE_B172	H28	H5
CORE_B173	H34	H6
CORE_B174	J29	H7
CORE_B175	K31	H9
CORE_B176	L28	H10
CORE_B177	L29	H11
CORE_B178	N33	H12
CORE_B179	N25	H14
CORE_B180	R34	H15
CORE_B181	R29	H16
CORE_B182	T30	H17
CORE_B183	U32	H19
CORE_B184	AE9	H20
CORE_B185	AG11	H21
CORE_B186	AC8	H22
CORE_B187	V34	H24
CORE_B188	W24	H25
CORE_B189	Y32	H26
CORE_B190	AA33	H27
CORE_B191	AB26	H29
CORE_B192	AC34	H30
CORE_B193	AD26	H31
CORE_B194	AG28	H32
CORE_B195	AJ31	H34
CORE_B196	AF29	H35
CORE_B197	AF25	H36
CORE_B198	AH27	H37

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference (Continued)

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B199	AG33	H39
CORE_B200	AM33	H40
CORE_B201	A33	J2
CORE_B202	E34	J3
CORE_B203	E31	J4
CORE_B204	G33	J5
CORE_B205	H33	J7
CORE_B206	J34	J8
CORE_B207	J31	J9
CORE_B208	L33	J10
CORE_B209	L31	J12
CORE_B210	M32	J13
CORE_B211	N24	J14
CORE_B212	P32	J15
CORE_B213	R28	J17
CORE_B214	T31	J18
CORE_B215	U31	J19
CORE_B216	AD10	J20
CORE_B217	AG10	J22
CORE_B218	AB7	J23
CORE_B219	V32	J24
CORE_B220	W32	J25
CORE_B221	Y33	J27
CORE_B222	AB32	J28
CORE_B223	AB30	J29
CORE_B224	AD27	J30
CORE_B225	AE33	J32
CORE_B226	AE34	J33
CORE_B227	AD29	J34
CORE_B228	AJ34	J35
CORE_B229	AG26	J37
CORE_B230	AK27	J38
CORE_B231	AE32	J39
CORE_B232	AL33	J40
CORE_B233	B32	K1

Table 10.1 SEAD™-3 CoreBus Signal Cross Reference (Continued)

Signal Name	FPGA (U35)	SEAD-3 CoreBus Connector (J14)
CORE_B234	F33	K2
CORE_B235	F31	K4
CORE_B236	G32	K5
CORE_B237	H32	K6
CORE_B238	J32	K7
CORE_B239	J30	K9
CORE_B240	L34	K10
CORE_B241	L30	K11
CORE_B242	M33	K12
CORE_B243	N28	K14
CORE_B244	P34	K15
CORE_B245	R32	K16
CORE_B246	T33	K17
CORE_B247	U33	K19
CORE_B248	AC10	K20
CORE_B249	AF10	K21
CORE_B250	V29	K22
CORE_B251	V33	K24
CORE_B252	W34	K25
CORE_B253	AA34	K26
CORE_B254	AB33	K27
CORE_B255	AB31	K29
CORE_B256	AE24	K30
CORE_B257	AF33	K31
CORE_B258	AF34	K32
CORE_B259	AE29	K34
CORE_B260	AH34	K35
CORE_B261	AG27	K36
CORE_B262	AK28	K37
CORE_B263	AD32	K39
CORE_B264	AL34	K40

Chapter 11

Connectors

All connectors on the board are listed in Table 11.1. On the PCB, Pin 1 on connectors is marked with a “1” or a dot. The “Sheet” column lists the diagram sheet instantiating the connector.

Table 11.1 Connectors

Ref	PCB Label	Type	Description
J10		ATX PSU	Connects to a standard ATX power supply. A power supply with standby current supply of minimum 720 mA is required (1A/1.5A peak recommended) for the 5V standby voltage. See Table 11.2.
J11		3 pin 0.1”Fan connector	Connector for a 3-wire **5Volt** fan Pin 1 = Fan Tach Pin 2 = 5.0V Pin 3 = GND
A3, A4, A6		Expansion connectors, 120 pin Samtec connectors	These are the connectors that facilitate add-on boards. The A6 pin is pre-defined. A3 and A4 have undefined pins connected to the FPGA. Note: Depending on the board variant, not all of the pins on A4 are available.
J1		USB type “Mini-B” connector	Connects to Cypress EZ-USB Controller for USB download to update Bitfiles and board Firmware.
J2		USB type “Mini-B” connector	Hi-Speed USB 2.0, only available on A00-R00209.
J5		3x18 pin 0.1” header	This connector contains all the I/O from the PIC32 I/O controller. See Table 11.5.
P1		RJ45 Ethernet connector	Standard 10/100 Ethernet Port
J7		9-way D-type connector	RS232 COMM Port 1
J6		USB type “Mini-B” connector	RS232 COMM Port 0 (with built-in FT232 driver to provide fast USB connection)
J4	USB DBG	3-pin 0.1” header	Provides a simple 3-wire connection for access to the serial port in the USB controller. This is only intended for debug of the USB controller firmware, so normally this connector is not used. See Table 11.6.

Table 11.1 Connectors (Continued)

Ref	PCB Label	Type	Description
J3	PIC DBG	3-pin 0.1” header	Provides a simple 3-wire connection for access to the serial port in the PIC32 controller. This is only intended for debug of the PIC32 controller firmware, so normally this connector is not used. See Table 11.6.
J8	PIC DEBUG	6-pin 0.1” dual row Header	This connector allows the user to connect a Microchip ICE probe to the PIC32 controller for firmware development. See Table 11.4.
J13		3-pin Plug header (Green)	Provide auxiliary power for add-on board: Pin 1 = 5V Pin 2 = GND Pin 3 = 12V These pins are directly connected to the ATX_PSU connector. See Table 11.3.
J15	CPLD JTAG	14-pin 2mm Boxed Header	Connects to Xilinx DLCx cable to allow the CPLD to be programmed. The CPLD is programmed before the board is delivered, and normally it should not need to be re-programmed. See Table 11.8.
J17	FPGA JTAG	14-pin 2mm Boxed Header	Connects to Xilinx DLCx cable to allow access to the FPGA/PIC JTAG chain. The PIC is programmed before the board is delivered using the PIC Debug header. The FPGA re-programming is usually achieved via the USB download mechanism. However, this connector does allow the user JTAG to access both devices. Note: See description of jumper J16. See Table 11.7.
U46		200-pin DDR II DRAM socket	Accepts a standard DDR II - 533 1.8V DRAM module.
J21	EJTAG	14-pin 0.1” header	EJTAG connector. See Table 5.1.
J20	iFlowtrace™	10-pin 0.1” Boxed Header	iFlowtrace™ Connector. See Table 5.2.
J22	PDtrace™	38-Pin Mictor	PDtrace™ Connector. See Table 5.3.
J14		400-position BGA	Interface to SEAD-3 CoreBus Connector. See Table 10.1.
P2		14-pin 0.1” header	Connects to the LCD module

Table 11.2 J10 - PSU Connector

Pin	Name	Description
1,2,11	3.3V	3.3V supply to board
3,5,7,13,15,16,17	GND	Connects to ground plane
4.6.19.20	5V	5V supply to board

Table 11.2 J10 - PSU Connector (Continued)

Pin	Name	Description
8	ATXOK	Asserted when power is stable from the PSU. Currently not used on the board, but connects to Altera PLD
9	5V_SB	5V standby power. Always present when the PSU is connected to the main power-supply outlet. Used on SEAD-3 to turn on power when the ON/OFF button is pushed
10	12V	12V supply to board
12	-12V	Unused on board but available on user connector
14	PS_ONN	The PSU will be ON when this signal is low
18	-5V	Unused

Table 11.3 J13 - Aux Power Outlet

Pin	Name	Description
1	VCC5V0	5 Volt
2	GND	Ground
3	VCC12V0	12 Volt

Table 11.4 J8 - PIC32 Debug Header

Pin	Name	Description	Notes
1	R_PIC_RST_N	PIC32 Master Reset.	
2	VCC3V3	3.3V	
3	GND	GND.	
4	ADC_CH0	PIC32 pin PDG1	NOTE: The PDC1/PDG1 are dual use pins on the PIC32. ADC_CH0 (J5 pinC10) and ADC_CH1 (J5 pin C11) must not be connected at the same time as the ICS probe.
5	ADC_CH1	PIC32 PGC1	
6		No Connect.	

Table 11.5 PIC32 IO COnnector (J5)

Pin	Function	Pin	Function	Pin	Function
A1	GPIO0	B1	VCC3V3	C1	GND
A2	GPIO1	B2	GND	C2	SP_CLK
A3	GPIO2	B3	GND	C3	SP_MISO
A4	GPIO3	B4	GND	C4	SP_MOSI
A5	GPIO4	B5	GND	C5	SP_CS0

Table 11.5 PIC32 IO COnnector (J5) (Continued)

Pin	Function	Pin	Function	Pin	Function
A6	GPIO5	B6	GND	C6	SP_CS1
A7	GPIO6	B7	GND	C7	SP_CS2
A8	GPIO7	B8	GND	C8	SP_CS3
A9	GPIO8	B9	GND	C9	GND
A10	GPIO9	B10	GND	C10	ADC_CH0
A11	GPIO10	B11	GND	C11	ADC_CH1
A12	GPIO11	B12	GND	C12	ADC_CH2
A13	GPIO12	B13	GND	C13	ADC_CH3
A14	GPIO13	B14	GND	C14	ADC_CH4
A15	GPIO14	B15	GND	C15	ADC_CH5
A16	GPIO15	B16	GND	C16	ADC)CH6
A17	IIC_SCL	B17	GND	C17	ADC_CH7
A18	IIC_SDA	B18	VCC3V3	C18	GND

Table 11.6 J3 and J4 - Serial Port Pinouts.

Pin	Name	Direction
1	TXD	Output
2	GND	
3	RXD	Input

Table 11.7 J17 - FPGA JTAG Connector

Pin	Name	Description
2	VCC 3.3Volt	Supply voltage to the download cable
1,3,5,7,9,11,13	GND	Ground
4	USB_TMS	FPGA JTAG mode select
6	USB_TCK	FPGA JTAG clock
8	USB_TDO	FPGA JTAG data out
10	USB_TDI	FPGA JTAG data in
12	NC	
14	NC	

Table 11.8 J15 - CPLD JTAG Connector

Pin	Name	Description
2	VCC 3.3Volt	Supply voltage to the download cable
1,3,5,7,9,11,13	GND	Ground
4	CPLD_TMS	CPLD JTAG mode select
6	CPLD_TCK	CLPD JTAG clock
8	CPLD_TDO	CPLD JTAG data out
10	CPLD_TDI	CPLD JTAG data in
12	NC	
14	NC	

Chapter 12

Jumpers

Jumpers settings are listed in Table 12.1. All jumpers are standard 0.1” pitch. It is worthwhile getting some spares, in case you lose any. Pin 1 is marked with “1”. On all jumpers, pin numbering is cross-wise, i.e., the end pins are 1 and 2 (this is not always the case on other connectors).

Table 12.1 Jumpers

Ref	Pins	Position	Description
J19	10 -pin 0.1” header	1-2 3-4 5-6 7-8 9-10	Fit jumpers as shown in Positions 1-2, 3-4 to route IO power to the CPU module. Fit jumpers as shown in Positions 7-8, 9-10 to route CORE power to the CPU module. Positions 5-6 are unused.
J9		2 pin 0.1” header	Network AUTO DIX, enable network cable auto switching for uplink. In = Disable Out = Enable Default = Enable
J16	2-pin 0.1” header	1-2	Fit jumpers as shown in the “Position” column to close the FPGA JTAG chain (normal operation). (Default is fitted) If further devices are added to the chain, J16 must be left open.
J18	AUX RESET	2 pin 0.1” header	Connects to an auxiliary reset. Connecting pin 1 to pin 2 (GND) will cause a reset. (Default is OPEN)
J12	10 -pin 2mm header	1-2 3-4 5-6, 7-8, 9-10 = Reserved	Fit jumpers as shown in the “Position” column to route the FPGA temperature sensor to the temperature monitor IC (normal operation). (Default is fitted)

Chapter 13

Switches

Switches are listed in Table 13.1, together with their functions.

Table 13.1 Switches

Ref	Type	Position	Description
SW9	Push-button		NMI/Power ON button. In a benchtop environment, this button will bring the ATX power supply out of standby. It can also be used to generate an NMI to the CPU, for example, to shut down the PSU again.
SW10	Push-button		Reset button.
SW1	8-way DIP	SW1[1], SW1[2], SW1[3]	Clock Frequency select. Refer to Table 7.2
		SW1[4], SW1[5], SW1[6]	Clock Frequency select. Refer to Table 7.3
		SW1[7], SW1[8]	Module clock control. Refer to Table 7.1
SW2	4-way DIP	SW2[1]	SW_BIGEND_P: When Off: Set the CPU to Little Endian mode. (Default) When On: Set the CPU to Big Endian mode.
		SW2[2]	SW_XI_PGM_MODE: When Off: Enable EzUSB controller access to the FPGA JTAG chain (Default) When On: Disable EzUSB controller access to the FPGA JTAG chain
		SW2[3]	SW_SRAM_MAP_ZERO: When Off: MAPS SRAM @ address 0x1E00,0000 - 0x1E3f,ffff When On: Maps SRAM @ address 0x0000 0000 - 0x003f,ffff
		SW2[4]	SW_FPGA_OPTION: (currently unused) This switch is directly connected to the FPGA.
SW3	8-way DIP		This switch, marked “F_SWITCH” on the PCB, provides a value which can be read via the peripheral bus when PI_SEL[4:0]=01001. SW3-1 maps to PI_D[0], SW3-2 maps to PI_D[1], and so forth.
SW4	8-way DIP		This switch, marked “P_SWITCH” on the PCB, provides a value which can be read via the peripheral bus when PI_SEL[4:0]=01001. SW4-1 maps to PI_D[0], SW4-2 maps to PI_D[1], and so forth.

Table 13.1 Switches (Continued)

Ref	Type	Position	Description
SW5, SW6, SW7, SW8, SW11	Push-buttons		These push-buttons provide a simple 5 button keypad; North, South, East, West and Center. This allows user to add simple interactive controls. SW6 and SW8 are also used for Serial port boot up selection. When SW6 is held down during a reset, YAMON will select tty1 RS232 J7 as the default port. When SW8 is held down during a reset, YAMON will select tty0 USB to Serial J67 as the default port.

Chapter 14

LEDs

Table 14.1 lists all the LEDs and their purpose.

Table 14.1 LEDs

LED	Label	Type	Description
D1	Activity	Green LED	Indicates MicroSD card activity
D2	FDPLX	Yellow LED	Ethernet Full Duplex
D3	PHY_INIT_DONE	Green LED	DDRII Phy initialized
D4	ERROR	Red LED	DDRII PHY Error
D5	D5V_OK	Green LED	LED will be ON when the 5V power is within +-5% of the nominal value.
D6	D3V3_OK	Green LED	LED will be ON when the 3V3 power is within +-5% of the nominal voltage.
D7	VCC5V0	Green LED	LED will be ON when the board is power up.
D8	CONF_DONE	Green LED	Turned on when the FPGA is successfully configured, either from cable or boot Flash.
D9	USB_DL	Green LED	This LED is turned on when USB download is active.
D10	RESET	Red LED	This LED will be ON while the board is in the reset state.
D11	VCC5V0STBY	Blue LED	LED will be ON when the standby power is available from the ATX power supply. This will be the case when the power supply is connected to the main outlet.
D12	PLD0	Green LED	P-LED - 0
D13	PLD1	Green LED	P-LED - 1
D14	PLD2	Green LED	P-LED - 2
D15	PLD3	Green LED	P-LED - 3
D16	PLD4	Green LED	P-LED - 4
D17	PLD5	Green LED	P-LED - 5
D18	PLD6	Green LED	P-LED - 6
D19	PLD7	Green LED	P-LED - 7
D20	FLD0	Green LED	F-LED - 0
D21	FLD1	Green LED	F-LED - 1

Table 14.1 LEDs (Continued)

LED	Label	Type	Description
D22	FLD2	Green LED	F-LED - 2
D23	FLD3	Green LED	F-LED - 3
D24	FLD4	Green LED	F-LED - 4
D25	FLD5	Green LED	F-LED - 5
D26	FLD6	Green LED	F-LED - 6
D27	FDL7	Green LED	F-LED - 7

Chapter 15

Test Points

Table 15.1 lists all the SEAD-3 board test points.

Table 15.1 Test Points

Ref.	Signal Name	Description
TP1	GND	GND
TP2	VCC5V0	5V voltage supply
TP3	VCC3V3	3V3 voltage supply
TP4	GND	GND
TP5	VCC3V3STBY	3V3 standby voltage.
TP6	CLK_A0	Unused clock from synthesizer 0
TP7	GL_A3_LV_CLK0	Expansion connector clock
TP8	GL_LV_CCLK0	Clock source for the Module clock selector
TP9	CLK_B0	Unused clock from synthesizer 1
TP10	PIC_CLK_20MHZ	PIC32 controller clock
TP11	GL_A3_LV_CLK1	Expansion connector clock
TP12	GL_LV_CLK1	Synthesizer 1 clock
TP13	CLK_B1	Unused clock
TP14	USBPHY_CLK_24MHZ	Hi-Speed USB PHY clock
TP15	ETH_CLK_25MHZ	Ethernet controller clock
TP17	USB_CLK_24MHZ	EZUSB Controller clock
TP18	GL_LV_CCLK1	Clock source for the Module clock selector from synthesizer 1, based on Table 7.2
TP19	GL_LV_CLK0	Clock source for the Module clock selector from synthesizer 0, based on Table 7.1
TP20	GL_MOD_LV_CLK	FPGA_Module-3 clock
TP25	GND	GND
TP26	VCC2V5	2V5 voltage supply

Table 15.1 Test Points (Continued)

Ref.	Signal Name	Description
TP30	VCC1V8	DDRII DRAM Supply
TP31	VCCINT	Core voltage for the FPGA
TP34	VTTDDR	DDRII DRAM termination supply
TP35	GND	GND
TP36	VTTVREF	DDRII DRAM volatge reference

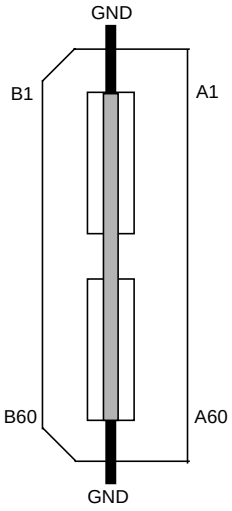
Chapter 16

Building a User Board

One of the advantages of SEAD™-3 is that the user can extend functionality by connecting a user-designed board to the SEAD-3 Epansion Connectors. There are between 128 and 238 completely uncommitted signals (depending on the FPGA version) between the expansion connectors and the FPGA. Furthermore, the entire peripheral bus is available, and the user can use this bus as is, or modified with added functionality.

The SEAD-3 board interfaces to the user board through three, high-density connectors located on either side of the CPLD and to the north of the FPGA. The connectors on the SEAD-3 board are from Samtec, Order Number QTH-060-05-L-D-A.

Figure 16.1 Front Panel Connector



Expansion Connector

Samtec	Expansion Connector
01	A1
02	B1
03	A2
04	B2
...	...
117	A59
118	B59
119	A60
120	B60

Pin Mapping Samtec - Expansion Connector

The pin list for the SEAD-3 board is shown in Table 16.1 through Table 16.3. The color codes used in the pin tables are shown below.

	CLK_N
	CLK_P
	VREF
	GCLK

Table 16.1 A3 Connector Pins

Signal Name	FPGA		Pin Polarity		FPGA	Signal Name
A3_B(1)	L3	B1	N	A1	F5	A3_A(1)
A3_B(2)	K3	B2	P	A2	G5	A3_A(2)
A3_B(3)	G1	B3	N	A3	H5	A3_A(3)
A3_B(4)	F1	B4	P	A4	J5	A3_A(4)
A3_B(5)	G2	B5	N	A5	F4	A3_A(5)
A3_B(6)	G3	B6	P	A6	E4	A3_A(6)
A3_B(7)	E1	B7	N	A7	M6	A3_A(7)
A3_B(8)	E2	B8	P	A8	L6	A3_A(8)
A3_B(9)	F3	B9	N	A9	J4	A3_A(9)
A3_B(10)	E3	B10	P	A10	H4	A3_A(10)
A3_B(11)	D1	B11	N	A11	L5	A3_A(11)
A3_B(12)	D2	B12	P	A12	M5	A3_A(12)
A3_B(13)	B2	B13	N	A13	L4	A3_A(13)
A3_B(14)	C3	B14	P	A14	K4	A3_A(14)
A3_B(15)	B1	B15	N	A15	N5	A3_A(15)
A3_B(16)	C2	B16	P	A16	N4	A3_A(16)
A3_B(17)	H3	B17	N	A17	T11	A3_A(17)
A3_B(18)	H2	B18	P	A18	U11	A3_A(18)
A3_B(19)	N3	B19	N	A19	T9	A3_A(19)
A3_B(20)	M3	B20	P	A20	U10	A3_A(20)
A3_B(21)	M2	B21	N	A21	T10	A3_A(21)
A3_B(22)	N2	B22	P	A22	R11	A3_A(22)
A3_B(23)	R3	B23	N	A23	T8	A3_A(23)
A3_B(24)	P2	B24	P	A24	U8	A3_A(24)
A3_B(25)	R1	B25	N	A25	R7	A3_A(25)
A3_B(26)	T1	B26	P	A26	R6	A3_A(26)
A3_B(27)	R2	B27	N	A27	T6	A3_A(27)
A3_B(28)	P1	B28	P	A28	U7	A3_A(28)
A3_B(29)	U2	B29	N	A29	T5	A3_A(29)
A3_B(30)	U1	B30	P	A30	U6	A3_A(30)
A3_B(31)	T3	B31	N	A31	P5	A3_A(31)
A3_B(32)	U3	B32	P	A32	P6	A3_A(32)
A3_B(33)	L9	B33	N	A33	Y9	A3_A(33)
A3_B(34)	K9	B34	P	A34	W9	A3_A(34)
A3_B(35)	F8	B35	N	A35	V10	A3_A(35)
A3_B(36)	F9	B36	P	A36	W10	A3_A(36)

Table 16.1 A3 Connector Pins (Continued)

Signal Name	FPGA		Pin Polarity		FPGA	Signal Name
A3_B(37)	M10	B37	N	A37	V8	A3_A(37)
A3_B(38)	M11	B38	P	A38	V9	A3_A(38)
A3_B(39)	L10	B39	N	A39	V5	A3_A(39)
A3_B(40)	L11	B40	P	A40	W5	A3_A(40)
A3_B(41)	E8	B41	N	A41	V7	A3_A(41)
A3_B(42)	E7	B42	P	A42	W7	A3_A(42)
A3_B(43)	E9	B43	N	A43	Y8	A3_A(43)
A3_B(44)	F10	B44	P	A44	Y7	A3_A(44)
A3_B(45)	J9	B45	N	A45	W6	A3_A(45)
A3_B(46)	J10	B46	P	A46	Y6	A3_A(46)
A3_B(47)	G10	B47	N	A47	AA5	A3_A(47)
A3_B(48)	H10	B48	P	A48	AA6	A3_A(48)
A3_B(49)	K8	B49	N	A49	AG6	A3_A(49)
A3_B(50)	L8	B50	P	A50	AF6	A3_A(50)
A3_B(51)	J7	B51	N	A51	AJ4	A3_A(51)
A3_B(52)	K7	B52	P	A52	AH4	A3_A(52)
A3_B(53)	M7	B53	N	A53	AF4	A3_A(53)
A3_B(54)	M8	B54	P	A54	AE4	A3_A(54)
A3_B(55)	G7	B55	N	A55	AE6	A3_A(55)
A3_B(56)	G6	B56	P	A56	AF5	A3_A(56)
A3_B(57)	N7	B57	N	A57	AD6	A3_A(57)
A3_B(58)	P7	B58	P	A58	AD5	A3_A(58)
A3_B(59)	N19	B59	N	A59	AC5	A3_A(59)
VDD		B60	P	A60	AB6	A3_A(60)

Table 16.2 A4 Connector Pins

Signal Name	Bank 15 FPGA		Pin Polarity	Bank 13 FPGA		Signal Name
A4_B(1)	AK7	B1	N	A1	AN4	A4_A(1)
A4_B(2)	AK6	B2	P	A2	AN5	A4_A(2)
A4_B(3)	AG7	B3	N	A3	AL4	A4_A(3)
A4_B(4)	AH7	B4	P	A4	AM5	A4_A(4)
A4_B(5)	AJ7	B5	N	A5	AP4	A4_A(5)
A4_B(6)	AJ6	B6	P	A6	AP5	A4_A(6)
A4_B(7)	AJ10	B7	N	A7	AL9	A4_A(7)

Table 16.2 A4 Connector Pins (Continued)

Signal Name	Bank 15 FPGA		Pin Polarity	Bank 13 FPGA		Signal Name
A4_B(8)	AH10	B8	P	A8	AL8	A4_A(8)
A4_B(9)	AG8	B9	N	A9	AM6	A4_A(9)
A4_B(10)	AH8	B10	P	A10	AN7	A4_A(10)
A4_B(11)	AK9	B11	N	A11	AM7	A4_A(11)
A4_B(12)	AK8	B12	P	A12	AM8	A4_A(12)
A4_B(13)	AM30	B13	N	A13	AN8	A4_A(13)
A4_B(14)	AN30	B14	P	A14	AN9	A4_A(14)
A4_B(15)	AL31	B15	N	A15	AL10	A4_A(15)
A4_B(16)	AM31	B16	P	A16	AM10	A4_A(16)
A4_B(17)	AL24	B17	N	A17	an15	A4_A(17)
A4_B(18)	AL25	B18	P	A18	AP15	A4_A(18)
A4_B(19)	AM20	B19	N	A19	AN14	A4_A(19)
A4_B(20)	AM21	B20	P	A20	AP14	A4_A(20)
A4_B(21)	AP20	B21	N	A21	AM16	A4_A(21)
A4_B(22)	AN20	B22	P	A22	AM15	A4_A(22)
A4_B(23)	AM23	B23	N	A23	AP12	A4_A(23)
A4_B(24)	AM23	B24	P	A24	AP11	A4_A(24)
A4_B(25)	AM22	B25	N	A25	AM13	A4_A(25)
A4_B(26)	AN22	B26	P	A26	AN13	A4_A(26)
A4_B(27)	AM18	B27	N	A27	AP17	A4_A(27)
A4_B(28)	AN18	B28	P	A28	AP16	A4_A(28)
A4_B(29)	AP21	B29	N	A29	AM17	A4_A(29)
A4_B(30)	AP22	B30	P	A30	AN17	A4_A(30)
A4_B(31)	AP19	B31	N	A31	AM11	A4_A(31)
A4_B(32)	AN19	B32	P	A32	AN10	A4_A(32)
A4_B(33)	AL26	B33	N	A33	AH13	A4_A(33)
A4_B(34)	AM26	B34	P	A34	AH14	A4_A(34)
A4_B(35)	AL30	B35	N	A35	AJ24	A4_A(35)
A4_B(36)	AL29	B36	P	A36	AH24	A4_A(36)
A4_B(37)	AP24	B37	N	A37	AH15	A4_A(37)
A4_B(38)	AN24	B38	P	A38	AJ14	A4_A(38)
A4_B(39)	AP25	B39	N	A39	AJ12	A4_A(39)
A4_B(40)	AP26	B40	P	A40	AJ11	A4_A(40)
A4_B(41)	AP6	B41	N	A41	AJ21	A4_A(41)
A4_B(42)	AP7	B42	P	A42	AJ22	A4_A(42)
A4_B(43)	AL14	B43	N	A43	AK21	A4_A(43)

Table 16.2 A4 Connector Pins (Continued)

Signal Name	Bank 15 FPGA		Pin Polarity	Bank 13 FPGA		Signal Name
A4_B(44)	AL13	B44	P	A44	AJ20	A4_A(44)
A4_B(45)	AL11	B45	N	A45	AJ15	A4_A(45)
A4_B(46)	AK11	B46	P	A46	AK14	A4_A(46)
A4_B(47)	AN12	B47	N	A47	AK23	A4_A(47)
A4_B(48)	AM12	B48	P	A48	AK22	A4_A(48)
A4_B(49)	AP10	B49	N	A49	AL16	A4_A(49)
A4_B(50)	AP9	B50	P	A50	AL15	A4_A(50)
A4_B(51)	AL6	B51	N	A51	AK12	A4_A(51)
A4_B(52)	AL5	B52	P	A52	AK13	A4_A(52)
A4_B(53)	AH23	B53	N	A53	AH18	A4_A(53)
A4_B(54)	AH22	B54	P	A54	AG18	A4_A(54)
A4_B(55)	AH17	B55	N	A55	AL19	A4_A(55)
A4_B(56)	AJ16	B56	P	A56	AL18	A4_A(56)
A4_B(57)	AK18	B57	N	A57	AL21	A4_A(57)
A4_B(58)	AK17	B58	P	A58	AL20	A4_A(58)
A4_B(59)	AK16	B59	N	A59	AK24	A4_A(59)
VDD		B60	P	A60	AL23	A4_A(60)

Table 16.3 A6 Connector Pins

Signal Name				Signal Name
PI_LOC_D8	B1		A1	PI_LOC_A0
PI_LOC_D9	B2		A2	PI_LOC_A1
PI_LOC_D10	B3		A3	PI_LOC_A2
PI_LOC_D11	B4		A4	PI_LOC_A3
PI_LOC_D12	B5		A5	PI_LOC_A4
PI_LOC_D13	B6		A6	PI_LOC_A5
PI_LOC_D14	B7		A7	PI_LOC_A6
PI_LOC_D15	B8		A8	PI_LOC_A7
PI_LOC_D16	B9		A9	PI_LOC_A8
PI_LOC_D17	B10		A10	PI_LOC_A9
PI_LOC_D18	B11		A11	PI_LOC_A10
PI_LOC_D19	B12		A12	PI_LOC_A11
PI_LOC_D20	B13		A13	PI_LOC_A12
PI_LOC_D21	B14		A14	PI_LOC_A13
PI_LOC_D22	B15		A15	PI_LOC_A14

Table 16.3 A6 Connector Pins (Continued)

Signal Name				Signal Name
PI_LOC_D23	B16		A16	PI_LOC_A15
PI_LOC_D24	B17		A17	PI_LOC_A16
PI_LOC_D25	B18		A18	PI_LOC_A17
PI_LOC_D26	B19		A19	PI_LOC_A18
PI_LOC_D27	B20		A20	PI_LOC_A19
PI_LOC_D28	B21		A21	PI_LOC_A20
PI_LOC_D29	B22		A22	PI_LOC_A21
PI_LOC_D30	B23		A23	PI_LOC_A22
PI_LOC_D31	B24		A24	PI_LOC_A23
PI_LOC_RD_N	B25		A25	PI_LOC_A24
PI_LOC_WE_N	B26		A26	PI_LOC_SEL0
PI_LOC_CS_N	B27		A27	PI_LOC_SEL1
GPIO0	B28		A28	PI_LOC_SEL2
GPIO1	B29		A29	PI_LOC_SEL3
GPIO2	B30		A30	PI_LOC_SEL4
GPIO3	B31		A31	PI_LOC_BE_N0
GPIO4	B32		A32	PI_LOC_BE_N1
GPIO5	B33		A33	PI_LOC_BE_N2
GCLK_FPGA_A	B34		A34	PI_LOC_BE_N3
GPIO6	B35		A35	PI_LOC_USER_INT
GPIO7	B36		A36	SP_MOSI
GPIO8	B37		A37	SP_MISO
GPIO9	B38		A38	SP_SCK
GPIO10	B39		A39	SP_CS2
GPIO11	B40		A40	SP_CS3
GPIO12	B41		A41	PI_LOC_EXTRN
GPIO13	B42		A42	USERBOARD_READY
GPIO14	B43		A43	XI_DONEO
GPIO15	B44		A44	XI_DONE1
IIC_SCL	B45		A45	XI_CS0_N
IIC_SDA	B46		A46	XI_CS1_N
SP_CS0	B47		A47	XI_WRITE_N
SP_CS1	B48		A48	XI_INIT
CLK1_RTRN	B49		A49	XI_CCLK
GL_A3_LV_CLK0	B50		A50	XI_PROGRAM_N
JT_TMS	B51		A51	XI_BUSY
JT_TCK	B52		A52	PI_LOC_D7

Table 16.3 A6 Connector Pins (Continued)

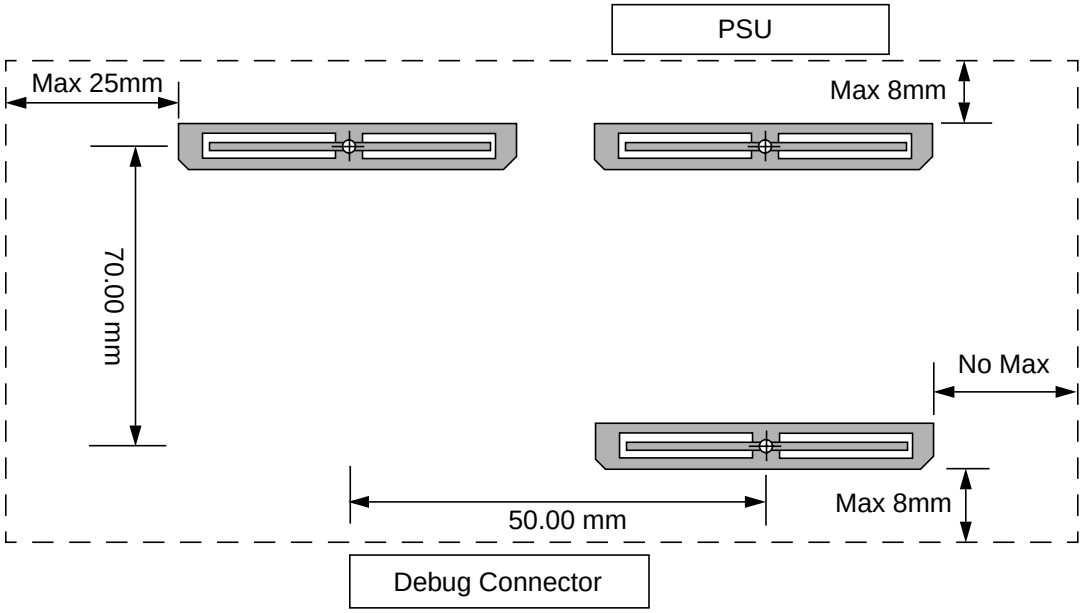
Signal Name				Signal Name
JT_TDO	B53		A53	PI_LOC_D6
XI_TDO	B54		A54	PI_LOC_D5
FPGA_CLK_P	B55		A55	PI_LOC_D4
FPGA_CLK_N	B56		A56	PI_LOC_D3
GL_RST_N	B57		A57	PI_LOC_D2
NC	B58		A58	PI_LOC_D1
CLK0_RTRN	B59		A59	PI_LOC_D0
VDD	B60		A60	GL_A3_LV_CLK1

16.1 SEAD™-3 Expansion Connector Layout and Mating

The user board should have three mating connectors for stability, even when not all signals are required. See Figure 16.2.

The Samtec Order Number for the connectors is QSH-060-01-L-D-A.

Figure 16.2 User Board Connectors and Mating



16.2 User Board Power

The user board receives 3.3V power from the SEAD-3 through the user connectors. Each connector has 1 power pin. Power should be limited to 1A per pin. There is a secondary power socket, J13, which can provide a 5-volt and 12-volt supply directly from the ATX connector. See Table 11.3.

Appendix A

References

1 SEAD™-3 Board Getting Started
MIPS Document: MD00687

2 MIPS® SEAD™-3 Basic RTL User’s Manual
MIPS Document: MD00693

3 MIPS® SEAD™-3 Basic RTL Reference Manual
MIPS Document: MD00692

4 SEAD™-3 IO Processor User’s Manual
MIPS Document: MD00630

5 SEAD™-3 Board Schematics
MIPS Document: MD00648

6 MIPS® Global Interrupt Controller User's Manual
MIPS Document: MD00695

7 YAMON™ User’s Manual
MIPS Document: MD00008

8 EJTAG Specification
MIPS Document: MD00047

9 iFlowtrace Specification
MIPS Document: MD00526

10 PDTrace Specification
MIPS Document: MD00136

11 FPGA Module 3 Schematics
MIPS Document: MD00663

Appendix B

Revision History

Change bars (vertical lines) in the margins of this document indicate significant changes in the document since its last release. Change bars are removed for changes that are more than one revision old.

This document may refer to Architecture specifications (for example, instruction set descriptions and EJTAG register definitions), and change bars in these sections indicate changes since the previous version of the relevant Architecture document.

Revision	Date	Comments
01.00	November 24, 2009	Initial release
01.01	February 19, 2010	Fix error in Table 2.6
01.02	March 10, 2010	• Add description of CPLD registers • Add new fields to SEAD-3_CFG register • Add Periodoc Pulse Generator to CPLD
01.03	July 1, 2010	• Add description of expansion connector.