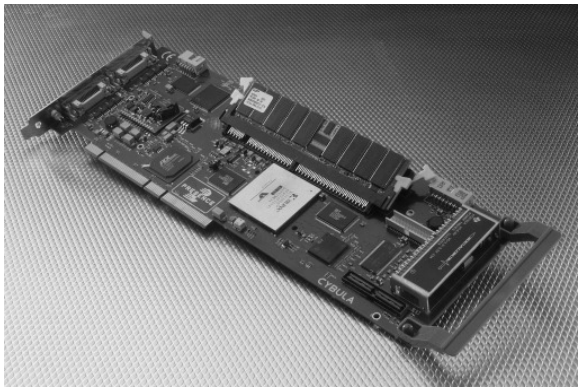
THE UNIVERSITY of York

PRESENCE-2 PCI64-NP DATA SHEET

PRELIMINARY DATA

JAN 2007 Rev A4



Features

- PCI v2.2 compliant 64-bit universal PCI card
- PCI Bus master capability will full Block / Scatter-Gather transfer DMA features
- 3 or 6 million system gate Virtex-II FPGA
- 300 MHz 'C6203 DSP offering up to 2400 MIPS and 600 million MACs/s
- High-speed 300 M-byte/s 32-bit FPGA-DSP link
- Two SDRAM DIMM sockets supporting up to 4 G-bytes of SDRAM
- 32 M-byte external DSP memory operating at 150 MHz
- Two independent 1 M-byte synchronous ZBT memories
- Mezzanine expansion slot for video and audio stream processing
- Compatible 23-way Sundance header for fast off-board digital I/O
- Two independent 4-bit bi-directional LVDS channels for high-speed board-to-board / Chassis-to-chassis communication
- FPGA reconfigurable 'on-the-fly' via PCI bus
- JTAG Emulator/RTDX support for rapid DSP code development
- Embedded DSP code loaded via PCI bus

Typical Applications

- Data Mining
- Pattern Matching
- Recognition System
- Video/Audio Processing
- IP/ASIC Prototyping

Ordering Information

Model Number	Description
PCI64-NP(3)	3 million system-gate FPGA fitted (Xilinx Virtex-II XC2V3000-5)
PCI64-NP(6)	6 million system-gate FPGA fitted (Xilinx Virtex-II XC2V6000-5)

Support

Web site : www.cs.york.ac.uk/arch/neural/hardware/presence-2/support

Description

The PCI64-NP product is part of the PRESENCE technology family developed by the University of York, UK. PRESENCE-2 is the latest architecture, currently only available for PCI bus based systems in the form of the PCI64-NP board.

PCI64-NP is a full-length 64-bit PCI card. It is targeted at the developer looking for an embedded single card solution to pattern matching, rapid search and match processing, complex codec implementation or any other high-performance applications requiring a large on-board memory resource, an embedded fast Digital Signal Processor (DSP) and a large Field Programmable Gate Array (FPGA).

Refer to the simplified block diagram shown in Figure 1.

The PCI Bridge device, shown in the diagram, provides a high-bandwidth gateway between the FPGA device and the PCI bus. The device equips the FPGA with the mechanism to send and receive single cycle or burst data to/from the PCI bus. The device, on power-up, enables the Host system to map the FPGA into PCI memory space for Target (slave) data transfers. The bridge device also enables the FPGA to act as PCI bus Initiator (master). The PCI Bridge supports DMA activity between the PCI bus and FPGA, enabling fast read/write block transfers of data from/to any resource accessible by the FPGA and the PCI bus. For example, it is possible to initiate DMA from Host system memory (PC/Workstation) to one of the SDRAM DIMMs on the card, or stream data from the card's DSP to a VGA adaptor PCI card.

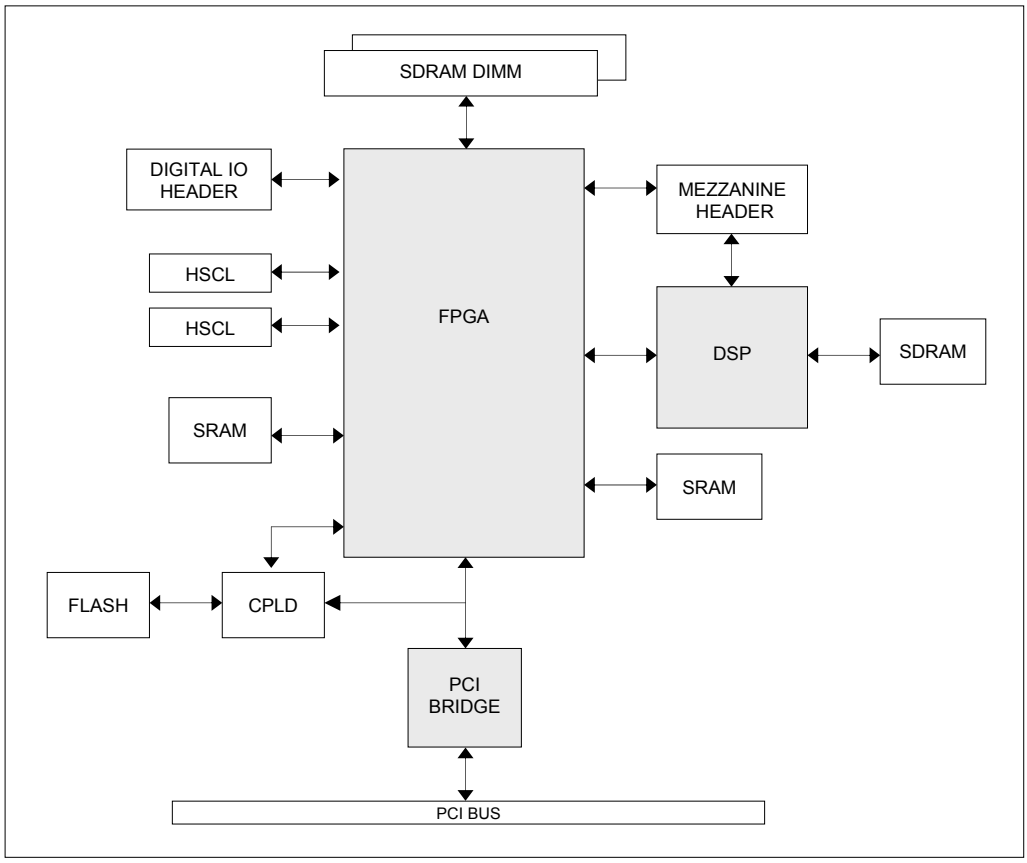


Figure 1. Simplified Block Diagram

The FPGA, as indicated in Figure 1, is given access to all on-board resources. Each resource is independently routed, enabling parallel operation in the FPGA. This architecture is ideal for exploring novel solutions since all the I/O interfacing and bus switching/muxing etc. is done in the FPGA.

FPGA configuration is accomplished by first loading the bitstream into the Flash memory. At power-up, or after a system reset, the FPGA is configured from Flash with the pre-loaded bitstream. This loading process is controlled by the CPLD. The FPGA can also be configured directly from the PCI bus. Alternatively, the dedicated FPGA JTAG Header (not shown in Figure 1) can be used to program the device using a Xilinx MultiLinx cable or other third-party JTAG programmer.

The FPGA has access to a 23-way Digital I/O Header. Its pin-out is compatible with Sundance's SDB header but may be used for any user-defined purpose, for example as a signal (net) breakout header for real-time FPGA debug.

The DSP has the ability of accessing FPGA I/O resources on the card using its 32-bit Expansion Bus (XBUS). This bus is routed directly to the FPGA. With a suitably designed XBUS interface in the FPGA, the DSP is able to manipulate and control the PCI Bridge device and initiate PCI bus cycles (as bus master) under DSP program control. The DSP's XBUS supports synchronous 32-bit read/write FIFO transfers. After configuring and enabling one of six dedicated DSP DMA engines, the developer is able to use the XBUS to transfer large blocks of data at high bandwidth between the DSP and FPGA.

The DSP is supplied with 32 M-bytes of SDRAM via its External Memory Interface (EMIF) for additional data/program memory.

Loading a DSP boot program can be achieved in two ways. During code development, the user will prefer to use the dedicated JTAG Emulator port (not shown in Figure 1) for connection to a Texas Instruments Emulator pod. With the emulator connected, the developer is able to load code and manipulate the DSP as required. As an embedded DSP target however, the boot code must be loaded from the Host system driver via the PCI bus. To do this, the FPGA must act as DSP Host Port Interface (HPI) bus master on the XBUS. A suitable HPI interface in the FPGA is required. The free STARTUP FPGA bitstream (available for download from the Support web site) provides a basic HPI interface, enabling the developer to quickly up-load DSP code. Alternatively, the developer may design his/her own interface.

Both the DSP and FPGA have access to the on-board Mezzanine Header. The header facilitates the installation of purpose-built daughter boards for such tasks as video capture, Ethernet Serialize/Deserialize, etc. Twenty-one user-defined FPGA I/O signals are routed to the header. All three DSP high bit-rate Multi-channel Buffered Serial Ports (McBSPs) are wired, plus DSP on-chip 32-bit timer control signals.

Two 168-pin DIMM sockets are wired to the FPGA to support the installation of two 72-bit registered 3.3V SDRAM DIMMs. The two DIMM sockets share a 72-bit data bus and common control signals.

The two independent High Speed Communication Link (HSCL) channels offer the card the capability of fast off-board communication with other cards/systems. Each HSCL channel is capable of 4-bit duplex communication, with (de)serialization/clocking provided by the FPGA.

Two 9 M-bit Zero-Bus-Turnaround (ZBT) synchronous SRAM memories are independently routed to the FPGA, enabling the memories to operate in parallel.

Field Programmable Gate Array (FPGA)

A high gate-count Xilinx Virtex-II FPGA is placed at the heart of the PCI64-NP board. All on-board resources, including the DSP, are independently routed to the FPGA. This 'star' PCB wiring configuration allows the developer to choose and implement connection topologies within the FPGA to best achieve desired operating functionality and bandwidth performance.

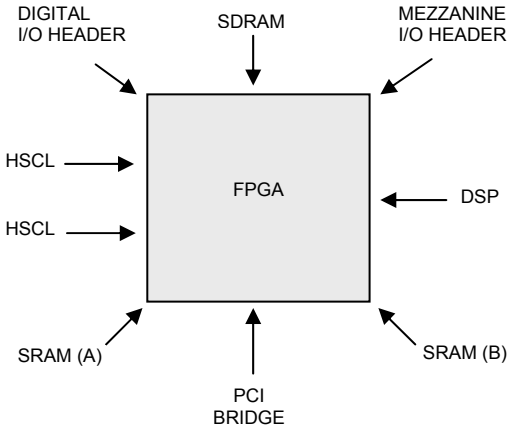


Figure 2: 'Star' configuration of routed board resources, with the FPGA placed at the centre.

For each resource shown in Figure 2, pin-locking information for the respective group of FPGA interface port signals is detailed in the User Constraints File named PCI64-NP.ucf, downloadable from the Support web site. For an explanation of specific signal (port) names listed in the UCF file, see the relevant I/O interface section in this datasheet.

WARNING: it is <u>essential,</u> when developing your own resource interfaces, that you use the PCI64-NP.ucf file during FPGA design implementation.

FPGA Configuration

WARNING: incorrect configuration of the FPGA may result in permanent damage of the device and other connected devices on the board. It is highly recommended that you read the User Manual before implementing your FPGA design.

The two BOARD STATUS LEDs are used to indicate success or failure in configuring the FPGA. When successful, the green LED illuminates. On fail, the red LED will flash continuously. Note: when in JTAG configuration mode, the red LED will flash and the green LED will not illuminate. See Table 1.

All FPGA I/O pins are pulled high during configuration.

BOARD STATUS LEDS		
GREEN LED	**RED LED**	**INDICATED STATUS**
off	off	CPLD/Power failure
off	flashing	Flash mode: FPGA configuration failure JTAG mode: indicates FPGA in JTAG configuration mode
on	off	Flash mode: FPGA configuration success JTAG mode. N/A

Table 1: BOARD STATUS LEDs. The table lists each supported condition.

Configuration From Flash

A Flash memory device on the card is used to configure the FPGA on power-up and after a system reset. It is pre-programmed with STARTUP during manufacture. STARTUP is an FPGA configuration bitstream that provides the user with the minimum functionality to achieve the following basic FPGA operations:

a) access to CPLD registers;
b) ability to re-program Flash memory via the CPLD with FPGA configuration PROM file;
c) load and boot the DSP using XBUS HPI cycles;
d) enable control of the FPGA STATUS LED for debug.

The latest version of STARTUP is available for free download from the Support web site. It is available in two PROM file formats, EXO (Motorola EXORMAX) and MCS (Intel MCS-86), for programming the Flash device.

STARTUP uses the 'Lite' version of the Simplified I/O Resources Layer (SIRL) IP Core. (Information about SIRL can be found on the Support web site. The full version of SIRL implements back-end interfacing to all FPGA connected devices and presents an abstracted and simplified front-end, to aid in rapid FPGA development, saving the developer considerable time and effort.)

To enable FPGA configuration from Flash memory, a 2 mm jumper must be installed on header J3. See the PCI64-NP Layout drawing in the Appendix for the location of J3. Note: header J2 must be left without a jumper fitted. See Figure 3.

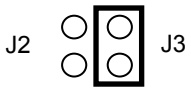


Figure 3: Jumper position to enable FPGA configuration from Flash memory.

Configuration via JTAG Boundary Scan

The FPGA JTAG Header enables the developer to configure the FPGA using third-party JTAG Boundary Scan programming tools. STARTUP is available in bitstream format for free download. To configure via JTAG, the developer must use Xilinx ISE 4.2i or later IMPACT software and a Xilinx MultiLinx cable connected to J1. See PCI64-NP Layout drawing for location of the header. Figure 4 details the pin-out of J1 for connection of the JTAG cable.

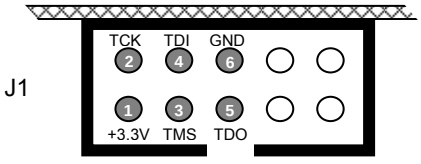


Figure 4: FPGA JTAG Header, J1. Pins viewed from edge of board.

DNC = Do Not Connect.
+3.3V pin fused at 2A.

2	TCK	4	TDI	6	GND	8	DNC	10	DNC
1	+3.3V	3	TMS	5	TDO	7	DNC	9	DNC

CAUTION: ensure the board is powered-down before connecting the MultiLinx cable; check connections before applying power.

Note: the boundary scan chain includes the CPLD device. This device should be ignored when configuring the FPGA.

A 2 mm jumper must be installed on header J2 to enable FPGA configuration via the JTAG header. See PCI64-NP Layout drawing for location of J2. Note, header J3 must be left without a jumper fitted - see Figure 5.

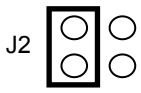


Figure 5: Jumper position for FPGA configuration via FPGA JTAG Header, J1.

'On-the Fly' Configuration

It is possible to configure the FPGA from the Host, directly via the PCI Bridge. This has the advantage of not having to program the Flash device, nor having to use a JTAG programming cable.

This feature will be detailed in later issues of the datasheet. Contact Support for further information.

FPGA Clock Sources

A 66.666 MHz clock source is supplied to the FPGA. This clock is synchronized to the local clock used by the PCI Bridge device for Local Bus signalling. Synchronizing the two devices enables the use of the Local Bus to transfer data between the PCI Bridge and FPGA at up to 266 million bytes/s.

For development purposes, an un-populated second clock source is available to the FPGA. To implement, the developer must fit an appropriate SMD clock generator in position U37 on the board. See the PCI64-NP Layout drawing for location of U37. Suitable oscillator device types are given in Table 3 below.

The 66.666 MHz clock source is wired to the FPGA's GCLK2P global clock input pin while the un-populated (free) clock source is input on GCLK0P global clock pin. View the PCI64NP.UCF file for complete pin-lock information, downloadable from the Support web site.

Using its internal DCMs, the FPGA is able to synthesis other frequencies as required, based on the 66.666 MHz input clock source.

Table 2: FPGA clock source specification

Frequency	Frequency Stability	Cycle-Cycle Jitter*	Skew from PCI Bridge Clock
66.666 MHz	+/- 25 ppm	+/- 150 ps max	< 500 ps

* Note: the jitter specification may be affected by aggressive noise on the +3.3V supply.

Table 3: Recommended SMD Crystal Oscillators types

Manufacturer	Product Type*
CTS	CB3LV-5C
M-TRON	M316FCN

*Note: the tri-state input control pin is pulled-up +3.3V on the board via a 10K resistor.

Table 4: FPGA global clock sources

Clock Source	FPGA Pin Name	UCF Net Name
66.666 MHz	GCKL2P	FPGA_CLK
U37	GCLK0P	USR_CLK

Note: both clocks enter the South East (SE) quadrant on the FPGA's floorplan.

FPGA Power Dissipation

It is recommended that a heatsink should be fitted to the FPGA when the device is likely to dissipate significant heat; i.e. when more than 3 Watts of power is dissipated at an ambient temperature, T_A (within the equipment enclosure), of +50ºC, or 5 Watts dissipated at 25ºC T_A. See Table 5.

The requirement to fit a heatsink is calculated on worst-case ambient temperature within the equipment enclosure and the expected FPGA power dissipation. The values in Table 5 are based on a T_j max of 85ºC in order to maintain applicability of published FPGA device speed files. In addition, normal air convection within the enclosure is assumed; equipment with chassis fans fitted may enable the FPGA to safely dissipate more power.

The location of the two-pin +12V power connector, J7, is shown on the PCI64-NP Layout drawing in the Appendix.

Table 5: Requirement to install a BGA heatsink (guidance only)

Heatsink required	
T_A = 25ºC	T_A = 50ºC
P_D > 5 Watts	P_D > 3 Watts

Table 6: Recommended Heatsink

Manufacturer	Part No.	Rated ºC/W	Max FPGA P_D @ T_A = 50ºC
AAVID Thermalloy *(Special connector required. For stock, contact Support)*	11-5608-13	3.3	12.5 Watts

FPGA Temperature Sensor

The FPGA's internal temperature sensor diode is routed to a MAX1617A device to enable the user to monitor FPGA die temperature. The MAX1617A's SMBus and Alarm interrupt pins are routed to the CPLD. Programming the device and reading temperature data etc. is accomplished by accessing the FPGA Temp Sensor Control and CPLD Status registers in the CPLD. See the CPLD section in the datasheet.

FPGA Debug

Four ways exist on the card to debug and verifying the loaded FPGA design:

- route test signals (probes) to the Digital I/O header (enables real-time debug)
- map test signals into PCI Bridge device's local bus address space (supports near real-time debug)
- route single test probe to the FPGA STATUS LED
- implement and use Xilinx's ChipScope system (supports real-time debug) – connect to FPGA JTAG header.

PCI Bridge

The 64-bit PCI Bridge, a PLX Technology PCI9656 device, is an extremely flexible and powerful PCI-to-local bus and local bus-to-PCI bridge adaptor. For technical information regarding this device, read the PCI64-NP User Manual and download the PCI 9656BA Data Book available from PLX Technology's web site www.plxtech.com.

The PCI Bridge supports full PCI bus master capability, enabling Block or Scatter-Gather transfer DMA to/from Host system resources. The DSP, with a suitable interface in the FPGA, is capable of controlling the PCI9656 device to act as PCI bus Initiator (master), enabling single-cycle, block or DMA type data transfer.

The default STARTUP FPGA configuration provides a simple PCI controller interface in the FPGA to support single cycle read/write transfers on the local bus for PCI target only cycles. For full master capability and to support DMA block transfers, the developer must design his/her own FPGA interface. The User Manual contains more complete information to aid the developer. Alternatively, the full version of SIRL may be implemented. When developing your own interface in the FPGA design, use DCI 50 Ohm source terminations as indicated in the PCI64-NP.ucf file. Figure 6 indicates the signals routed between the PCI9656 and FPGA devices.

The local bus clock signal, LCLK, is sourced from the same PLL used to supply the FPGA with its 66.666 MHz global clock. This ensures the local bus signals are synchronized with the FPGA's PCI controller interface.

The rich set of signals routed between the PLX9656 and the FPGA, together with the PCI9656 Local Configuration Registers, offers the developer a wide choice in implementing PCI Target and PCI Initiator solutions.

PCI bus Connector

- PCI Bus M66EN pin not grounded for 66 MHz PCI clock
- PCI bus INTA pin used for all PCI64-NP sourced PCI interrupts
- 25-Watt power requirement specified on the two PRSNT pins (PRSNT1# grounded). (Note: 0805 SMT 0R pull-down resistors are used to ground these pins - R67 and R66 for PRSNT1# and PRSNT2# respectively. The developer may fit/remove R66 and/or R67 for other values on the PRSTN pins, if necessary.)

PCI9656 Hardwired Configuration

- Power Management system not implemented
- C Mode Local Bus operation

- Data Parity bits on Local Bus not implemented

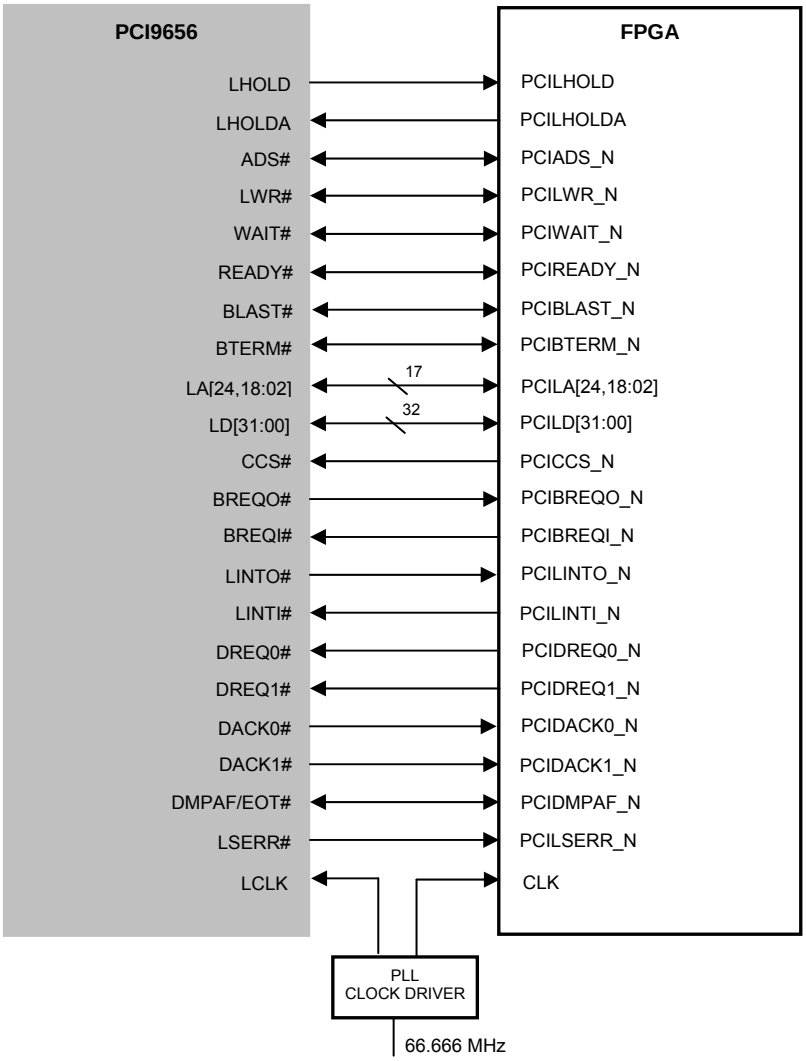


Figure 6: Signals routed between the PCI9656 device and FPGA. Note, the FPGA port signal names shown are reflected in the PCI64-NP.ucf file.

PULL-UP	PULL-DOWN
LA29	BREQI
	BREQ0
	LHOLDA
	LBE[3:0] – not shown

Table 7: List of pull-up / pull-down resistors added on the board.

Note: the local bus is configured for 32-bit data only.

Serial EEPROM

The PCI Bridge device is configured from an on-board 2k-bit serial EEPROM device. Configuration occurs after power-up and/or system (PCI) reset. The card is supplied with the EEPROM pre-programmed to enable slave access to the FPGA. The latest version of the program is available in an ASCII file STARTUP.EEPROM, downloadable form the Support web site.

Vital Product Data

64 bytes of the serial EEPROM memory are available to store user-defined VPD data. VPD data can be written and read as per PCI Local Bus Specification Rev 2.2, via the PCI9656 device's two PCI configuration space VPD Capability Structure support registers, VPD Address and VPD Data. Note: an offset of 80h must be added to the VPD Address field in order to correctly access VPD storage in EEPROM (a limitation of the PCI9656 device), i.e. VPD Address is offset from 80h, not 00h.

PCI Configuration Space

The Vendor ID, Device ID, Revision ID, Class Code, Subsystem ID and Subsystem Vendor ID register values are programmable via the serial EEPROM. The default values are given in Table 8. For a complete description of all configuration space registers available for the PCI Bridge device, refer to the PCI9656 Data Book.

The PCI Configuration Space BARs are programmed with values based on data stored in EEPROM. The default STARTUP.EEPROM contents in EEPROM configure the PCI9656 device to map the FPGA and CPLD into PCI address space for Target access only – see Figure 7. For details regarding STARTUP.EEPROM, visit the Support web site.

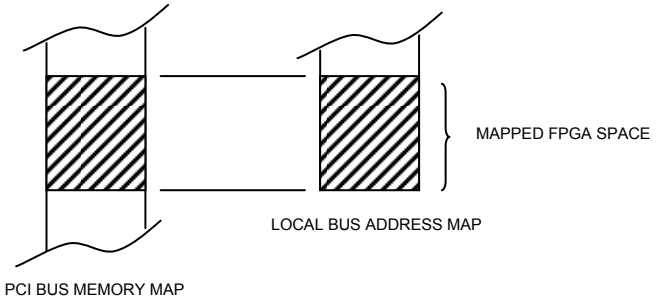


Figure 7: Default EEPROM contents STARTUP.EEPROM maps FPGA into PCI Memory Space.

Table 8: Default values for the pertinent PCI Configuration Space registers

Vendor ID	Device ID	Revision ID	Class Code		Subsystem ID	Subsystem Vendor ID
			Sub Class	Base Class		
10B5h	9656h	BAh	80h	06h	9656h	10B5h

DSP

The DSP implemented on the board is a Texas Instruments TMS320C6203B device. The data sheet and application notes for the DSP can be obtained from Texas Instruments' web site www.ti.com. The TMS320C6203B is a fixed-point DSP with six ALUs, two 16-bit integer multipliers, 7M-bits of on-chip SRAM and four independent (plus two auxiliary) DMA controllers. Its 32-bit External Memory Interface (EMIF) is configured on the card to access SDRAM. Its 32-bit Expansion Bus (XBUS) links the DSP to the FGPA. See Figure 8. The addition of buffered serial ports (McBSPs) enables the DSP to maintain three independent serial data streams to/from the Mezzanine header.

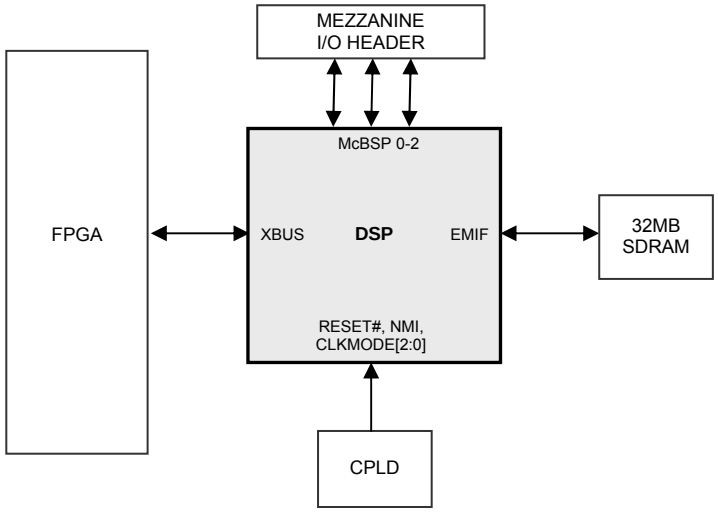


Figure 8: Board connectivity of the three independent DSP I/O interfaces.

Note: the CPLD device controls the reset state of the DSP and also determines the DSP's CPU clock frequency.

The diagram in Figure 9 shows all the signals routed on the board between the DSP and FPGA. The full XBUS is wired to the FPGA, plus HPI specific signals. In addition, all four external interrupt pins EXT_INT[7:4] are sourced from the FPGA, with the INUM bus and IACK signals. The DMAC[3:0] and Timer0 TIN/TOUT pins are also routed. The two clock signals XFCLK and XCLKIN complete the interconnection shown.

When developing your own FPGA XBUS interface, use a DCI 50 Ohm source termination on the XBUS signals, as indicated in the PCI64-NP.ucf file.

32M-byte SDRAM

Two 16 M-byte SDRAM devices are wired to the DSP's EMIF interface, fully populating both CE2 and CE3 EMIF address spaces with SDRAM memory. These devices are organizes as 1 M by 32-bits by 4 banks and can support the DSP running at 300 MHz (150 MHz EMIF bus speed).

The 32-bit data bus is wired in 16-bit lanes, supporting 16-bit and 32-bit integer EMIF data transfers only. An 8-bit write transfer is not supported and an attempt would result in an erroneous 16-bit write.

In order to correctly setup the SDRAM interface in the DSP, refer to the Support web site for recommended register values for the following EMIF control registers: GBLCTL, CECTL2, CECTL3, SDCTL and SDTIM.

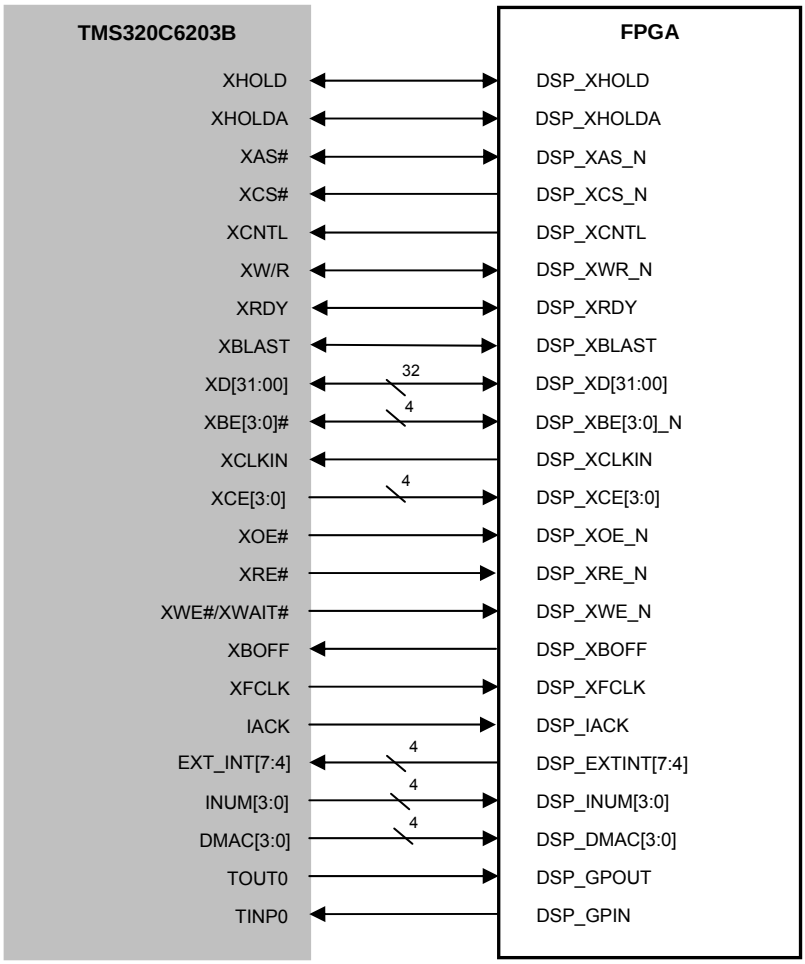


Figure 9: Interconnectivity implemented between the DSP and FPGA. Note: the FPGA port signal names shown are reflected in the PCI64-NP.ucf file.

DSP RESET and NMI

The DSP's RESET# and NMI interrupt inputs are sourced from the CPLD. These signals are latched, and maintain the level written in the relevant CPLD control register. See the separate section describing the CPLD. Sourcing from the CPD enables the DSP to continue functioning, if required, during FPGA (re)configuration.

DSP CPU Clock Frequency Selection

The three CLKMODE input signals, used by the DSP to select the clock source multiplier, are driven by the CPLD. This enables the user to determine the DSP's operating frequency from within Host application software, as required. See the separate section regarding the CPLD. The CPLD's CLKMODE[2:0] output signals, on power-up and/or system reset, are driven with the value 011b (default), selecting the x8 clock multiplier in the DSP. This results in a 266 MHz DSP instruction cycle frequency.

The DSP's CLKIN pin is driven from a crystal oscillator i.c. See Table 9. The DSP is independent of the FPGA's clock source.

Table 9: DSP's CLKIN oscillator i.c. specification

Frequency	Frequency Stability
33.333 MHz	+/- 25 ppm

DSP Boot Mode and Configuration

The TMS320C6203B is configured by placing a specific bit pattern on the DSP's XD pins whilst de-asserting the device's RESET# signal. The board does not pull-up or pull-down any XD pin directly. Rather, it allows the developer to use the

FPGA to accomplish this task, either by directly driving the signals or implementing a pull-up/pull-down in the appropriate IOB. The PCI64-NP.ucf file gives recommended IOB pull-up/pull-down constraints on the relevant FPGA net; the resultant DSP configuration is summarised in Table 10.

Table 10: Recommended bit field values for the XD pins during DSP boot configuration phase

XD Bit Field	Value	Description
BOOTMODE	00111b	HPI boot process, MAP1 memory map, address 0 internal
LEND	1	Little Endian mode
FMOD	0	Glue logic required in FPGA for all XBUS FIFO reads
XARB	0	Internal XBUS arbiter disabled
HMOD	1	External host interface (FPGA) is in sync master/slave mode
RWPOL	0	XW/R is active high for writes
BLPOL	0	XBLAST is active low
MTYPE XCEO	101b	32-bit FIFO interface for XCE0
MTPYE XCE1	101b	32-bit FIFO interface for XCE1
MTPYE XCE2	101b	32-bit FIFO interface for XCE2
MTPYE XCE3	101b	32-bit FIFO interface for XCE3

DSP Loader

Embedded DSP executable code may be loaded from the Host via the PCI bus with an appropriate FPGA implementation. With the DSP configured as detailed in Table 10, the FPGA will access the DSP's HPI registers to up-load the DSP code. Once loaded, the FPGA releases the DSP's CPU by setting the DSPINT bit in the XBISA Register. The DSP will then begin execution from internal address 0. Refer to Texas Instruments TMS320C62x DSP Expansion Bus (XBUS) Reference Guide (Lit. No. SPRU579) for detailed information regarding the HPI.

SIRL includes an HPI interface for boot loading the DSP. See the Support web site for more information.

DSP Debug

Two approaches exist to debug the DSP; use of the board's DSP JTAG Emulator header and access to the DSP's internal memory/registers via its Host Port Interface.

Debug via DSP JTAG Emulator Header

During development, it is often more convenient to use the DSP's JTAG port to up-load executable code from within Texas Instrument's Code Composer Studio IDE (CCS). This approach enables the use of the CCS' integrated debug tools and implementation of JTAG RTDX channels.

The DSP's JTAG port is configured on the board for emulation mode. The JTAG pins are routed to the DSP JTAG Emulator header, P3. The pin-out for the header is shown in Figure 10. The header is compatible with Spectrum Digital's XDS510PP Plus JTAG Emulator. See the PCI64-NP Layout diagram in the Appendix for the location of P3.

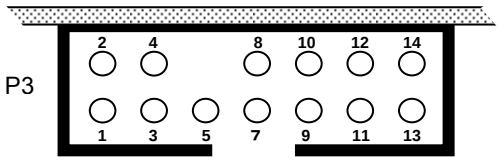


Figure 10: DSP JTAG Emulator header, P3. Pins viewed from edge of board.

Note: pin 6 is missing to enable connection of keyed emulator receptacle.

+3.3V pin is fused at 2A.

2	TRST#	4	GND	6	KEY	8	GND	10	GND	12	GND	14	EMU1
1	TMS	3	TDI	5	TDO	7	TDO	9	TCK_RET	11	TCK	13	EMU0

CAUTION: ensure power is removed from the board before connecting the Emulator cable.

Debug via Host Port Interface

With a suitable FPGA design, it is possible to use the DSP's HPI to enable the Host to read/write to DSP internal memory and manipulate DSP control registers. STARTUP, in addition to SIRL, provides the necessary FPGA interface. See the Support website for more information.

Mezzanine Header

A Mezzanine header is provided to facilitate an easy expansion of the board's capabilities. It is supplied with signals from both the FPGA and DSP devices to enable support of typical data acquisition functions and data streaming. Figure 11 illustrates the routed interconnect to these devices. A full pin-out is given in Table 11.

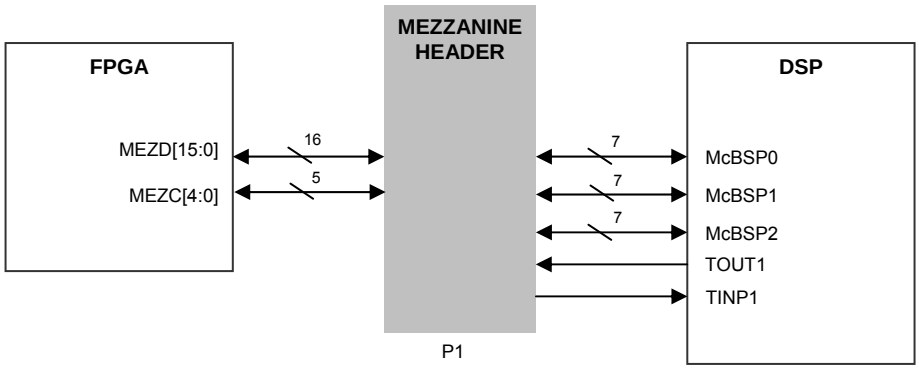
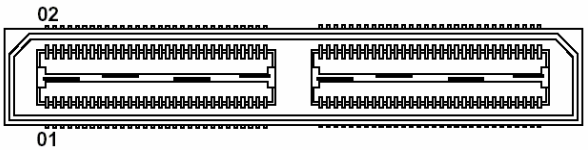


Figure 11: FPGA and DSP signals routed to the Mezzanine Header, P1.

Note: all McBSP signals are pull-up to +3.3V on the board, via a 10K resistor.

Three d.c. voltages, +3.3V and +/- 12V, are supplied to the header and are individually fused on-board. Table 12 lists the fuse ratings for each supply.

Several 3 mm mounting holes exist on the board, specifically for mounting daughter cards using the Mezzanine header.



P1 Header type: Samtec Part No. QSH-060-01-L-D-A

1	DSP_CLKR1	13	GND	25	GND	37	GND	49	N/C
2	DSP_FSR1	14	DSP_FSR2	26	DSP_FX0	38	DSP_TOUT1	50	N/C
3	DSP_CLKS1	15	DSP_FSX2	27	DSP_DR0	39	DSP_CLKR0	51	GND
4	DSP_CLKX1	16	GND	28	GND	40	GND	52	GND
5	GND	17	GND	29	GND	41	GND	53	N/C
6	DSP_DR1	18	DSP_CLKX2	30	DSP_CLKS0	42	GND	54	N/C
7	DSP_DR2	19	DSP_CLKR2	31	DSP_FSR0	43	N/C	55	N/C
8	DSP_DX1	20	GND	32	GND	44	N/C	56	N/C
9	GND	21	GND	33	GND	45	N/C	57	N/C
10	DSP_FXS1	22	DSP_DX0	34	DSP_TINP1	46	N/C	58	N/C
11	DSP_DX2	23	DSP_CLKS2	35	DSP_CLKR0	47	N/C	59	N/C
12	GND	24	GND	36	GND	48	N/C	60	N/C
61	MEZD0	73	MEZD10	85	MEZC2	97	GND	109	GND
62	MEZD1	74	MEZD11	86	MEZC3	98	GND	110	+3.3VMEZ
63	MEZD2	75	MEZD12	87	GND	99	GND	111	GND
64	MEZD3	76	MEZD13	88	GND	100	GND	112	+3.3VMEZ
65	MEZD4	77	MEZD14	89	MEZC4	101	GND	113	GND
66	MEZD5	78	MEZD15	90	GND	102	GND	114	+3.3VMEZ
67	MEZD6	79	GND	91	GND	103	+12VMEZ	115	GND
68	MEZD7	80	GND	92	GND	104	GND	116	+3.3VMEZ
69	GND	81	MEZC0	93	GND	105	+12VMEZ	117	-12VMEZ
70	GND	82	MEZC1	94	GND	106	+3.3VMEZ	118	+3.3VMEZ
71	MEZD8	83	GND	95	GND	107	GND	119	-12VMEZ
72	MEZD9	84	GND	96	GND	108	+3.3VMEZ	120	+3.3VMEZ

Table 11: Pin-out for the Mezzanine Header, P1.

Note: the centre pins are connected to GND (not numbered in the Table).

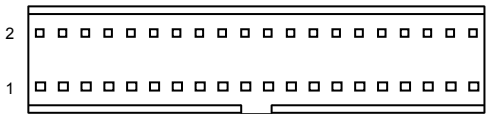
The MEZC and MEZD bus signal names are reflected in the PCI64-NP.ucf file.

Table 12: Fuse ratings for Mezzanine header voltage supplies

+3.3V	+12V	-12V
2 A (max)	375 mA (max)	375 mA (max)

Digital I/O Header

A 23-way digital I/O header is provided to facilitate connection between the FPGA and Sundance products that support the Sundance Data Bus (SDB) header specification. The pin-out is tabulated in Table 13. All signals are routed directly to the FPGA.



P2 Header type: ODU Part No. 515.568.035.040.000

Table 13: Pin-out of the Digital I/O Header

1	SUN_CLK	9	SUN_D3	17	SUN_D7	25	SUN_D11	33	SUN_D15
2	GND	10	GND	18	GND	26	GND	34	GND
3	SUN_D0	11	SUN_D4	19	SUN_D8	27	SUN_D12	35	SUN_USERDEF0
4	GND	12	GND	20	GND	28	GND	36	SUN_DIR0
5	SUN_D1	13	SUN_D5	21	SUN_D9	29	SUN_D13	37	SUN_WEN
6	GND	14	GND	22	GND	30	GND	38	SUN_DIR2/REQ
7	SUN_D2	15	SUN_D6	23	SUN_D10	31	SUN_D14	39	SUN_USERDEF1
8	GND	16	GND	24	GND	32	GND	40	SUN_DIR1/ACK

The signal names in Table 13 are reflected in the PCI64-NP.ucf file.

The header may be used for other purposes, such as the connection of a break-out box for real-time debugging purposes, useful during FPGA development. A DCI 50 Ohm source termination is available in all Digital I/O header FPGA IOBs, if required. See PCI64-NP.ucf file.

SRAM

Two 9 M-bit Zero-Bus-Turnaround (ZBT) synchronous SRAM devices, IDT part IDT71V65603, are independently routed to the FPGA, as shown in Figure 12. Write operations are 18-bit or 36-bit only, as indicated.

The devices are configured for Linear Burst Sequence mode of operation. Also, with ADV/LD# input signal wired to GND, a new address is always loaded.

To save power, the SRAM devices may be put to sleep by asserting the ZZ input signal, sourced from the CPLD. The signal is mapped into the CPLD's System Control Register. See the separate CPLD section for programming information.

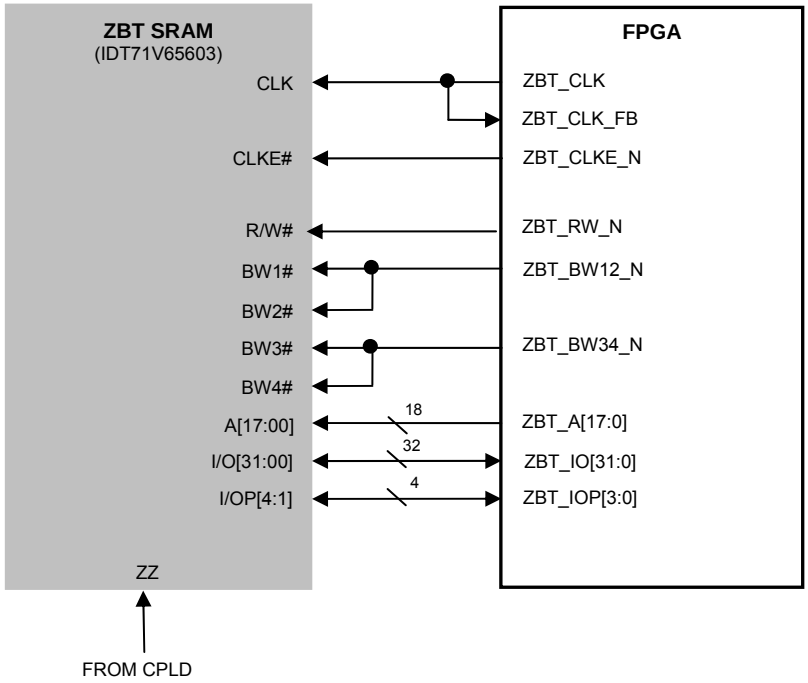


Figure 12: Connection between the ZBT SRAM device and FPGA. The routing for both ZBT SRAMs is identical.

Note: the FPGA port signal names shown are reflected in the PCI64-NP.ucf file; ZBTA and ZBTB replace ZBT prefix for the two devices, e.g. ZBTA_CLK and ZBTB_CLK.

Note: FPGA DCI source termination is not available for driving ZBT SRAM signals and must not be instantiated for either ZBT SDRAM device interface.

The full SIRL IP Core includes the necessary interface to operate these devices.

SDRAM DIMMs

Two 168-pin DIMM sockets enable the installation of up to two 3.3V SDRAM DIMM modules. Installing registered DIMMs is recommended rather than un-buffered types. Registered SDRAM offer higher performance and are more easily incorporated into a synchronous FPGA design implementation.

Only low-profile DIMMs of 30.48 mm (1.200 inches) or less in height should be installed. Note: a module installed in DIMM 2 socket will extend beyond the standard PCI card height. See the Mechanical Specification section. Ensure there is sufficient clearance in the target enclosure before installation, otherwise use DIMM 1 socket only.

If possible, non-stacked DIMM types should be used. This will offer maximum clearance between installed adjacent PCI cards.

SDRAM DIMM sizes up to 2 G-byte are supported, effectively offering 4 G-bytes of SDRAM memory on a single PCI64-NP card.

Table 14: Suggested SDRAM DIMM Modules

Manufacturer	Part No.	168-pin DIMM Type (all low-profile, 3.3V PC133 registered)	Size (M-Byte)
Samsung	M390S6450CTU-C7A	non-stacked	512
	M390S2950MTU-C75	non-stacked	1024
	M390S2858DTU-C7A	stacked	1024
	M390S5658MTU-C75	stacked	2048
Infineon	HYS72V64601GR-7.5	non-stacked	512
	HYS72V128520GR-7.5	stacked	1024
SMART	SM572284578EY3R01	stacked	1024
	SM572564578E83R01	stacked	2048

The routing between the FPGA and the two DIMM sockets is shown in Figures 13 and 14. Note: all common DIMM signals share their respective FPGA I/O pin connection, except for the CLK0 source and CS[3:0]# input.

A DCI 50 Ohm source termination is available in all FPGA SDRAM interface IOBs. When developing an interface in the FPGA design, use DCI source termination when driving I/O signals, as indicated in the PCI64-NP.ucf file.

The full SIRL IP Core includes a configurable generic SDRAM interface, compatible with leading manufacturers' SDRAM DIMM products. The core also includes an SPD interface.

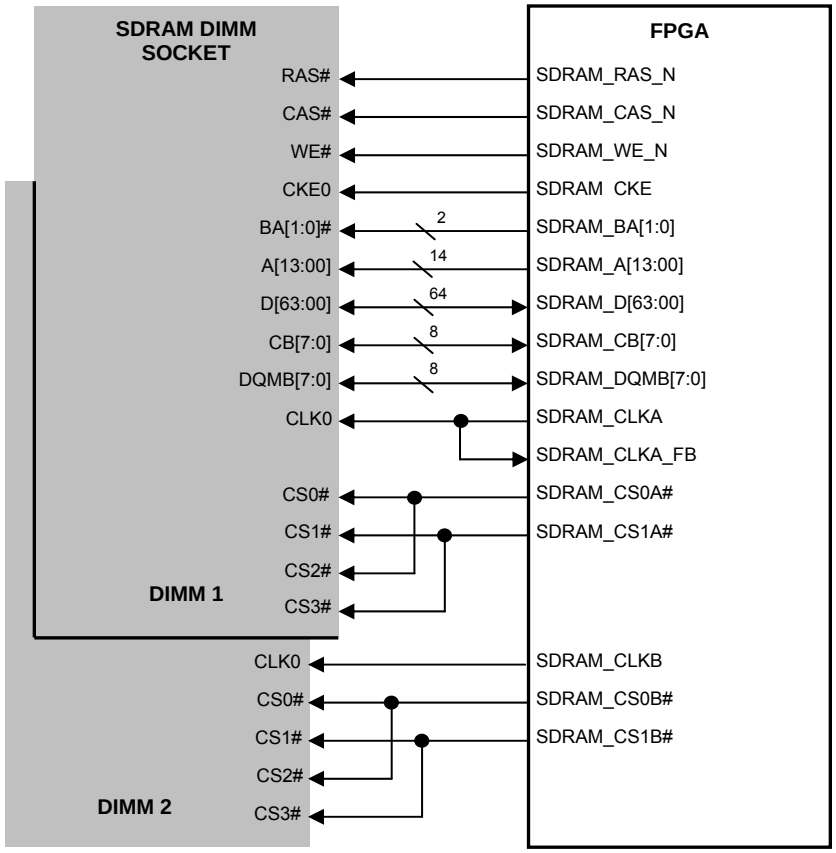


Figure 13: Connection between the two on-board SDRAM DIMM sockets and the FPGA.

Note: the FPGA port signal names shown are reflected in the PCI64-NP.ucf file.

Serial Presence Detect

The standard two-wire SPD is supported on the card, as shown in Figure 14. DIMMs 1 and 2 are given the hard-wired identity 000b and 001b respectively.

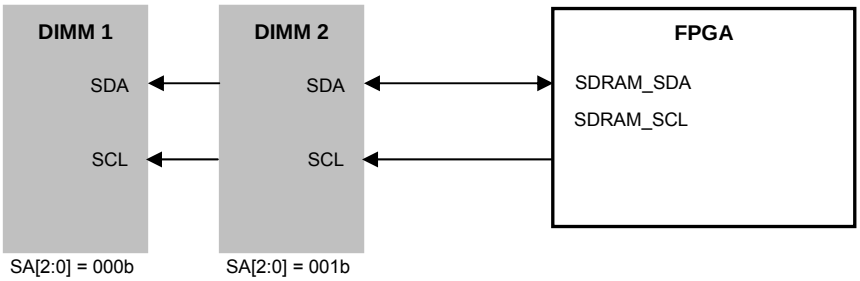


Figure 14: Implementation of Serial Presence Detect.

Note: the FPGA port signal names shown are reflected in the PCI64-NP.ucf file.

CPLD

A CPLD device on the board provides the mechanism for Flash memory programming and for FPGA boot configuration on power-up. It contains several control registers that are accessible to the Host via the FPGA.

The routed connections between the CPLD and FPGA are shown in Figure 15. This interface facilitates access to the CPLD registers.

The STARTUP FPGA configuration bitstream provides the user with a full and easy to use interface to the CPLD. However, the developer may implement his/her own design using the information contained in this datasheet.

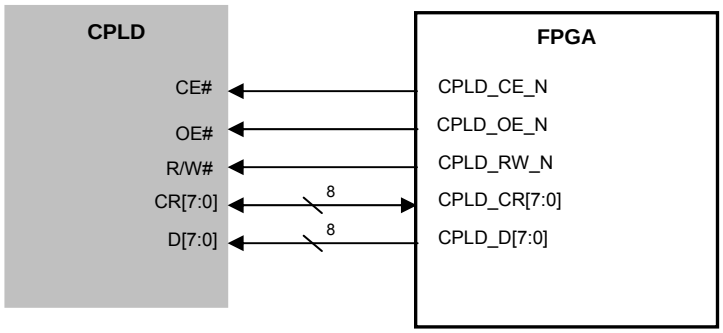


Figure 15: Connection between the CPLD and FPGA.

Note: the FPGA port signal names shown are reflected in the PCI64-NP.ucf file.

Two 8-bit buses are used to write data to the CPLD registers and Flash memory, the Control bus (CR[7:0]) and the Data Bus (D[7:0]). To write to the CPLD's Control Register, CE# and R/W# must be low and OE# high. A description of the bits in the Control Register is given in Figure 16. The indirectly referenced registers (System Control and FPGA Temp Sensor Control registers) are shown in Figures 17 and 18 respectively. A Status register is provided in the CPLD and is read by asserting the CE# and OE# pins, with R/W# held high. Figure 19 indicates the status bits available.

Reading Flash memory is achieved by setting the Read Flash Mode bit in the Control Register. Subsequent Flash read requests will result in Flash data being placed on the CR bus at consecutive byte addresses, starting at address 0. Clearing this mode is achieved by setting the Control Register's Reset Flash Interface Logic bit.

A write to Flash memory is accomplished by placing write data on the D bus, whilst setting the Program Flash bit in the Control Register. The first write occurs at byte address 0. Each write address is automatically incremented on successive writes. This mode is cleared by setting the Reset Flash Interface Logic bit in the Control Register. When programming the Flash, after each write the Flash Busy and CPLD busy status bits must be read to check if the Flash has completed its write cycle. A Subsequent write to Flash must not occur until the CPLD Busy bit is low and the Flash Busy# bit high.

Setting the Erase Flash bit in the Control Register will cause the entire Flash memory to be erased. Read the Flash Busy# and CPLD Busy bits in the Status Register to check if the erase operation has completed.

For CPLD read and write timing information, see Figures 20 to 22. Note: the CPLD is synchronized to the FPGA's clock source at half clock rate; the timing data shown assumes the FPGA CPLD interface is a synchronous implementation using the default FPGA clock.

CPLD Control Registers

Figure 16: Control Register. The CR bus is used to write to this register. Note: only bit 6 is latched.

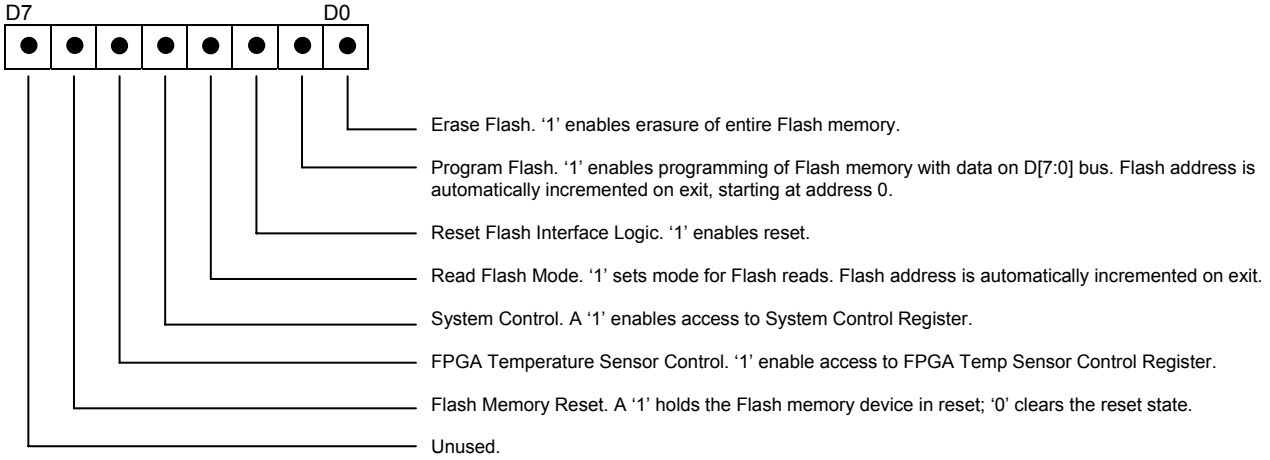


Figure 17: System Control Register. The D[7:0] bus accesses this register. Note: to access the System Control Register, bit CR[4] must be simultaneously set.

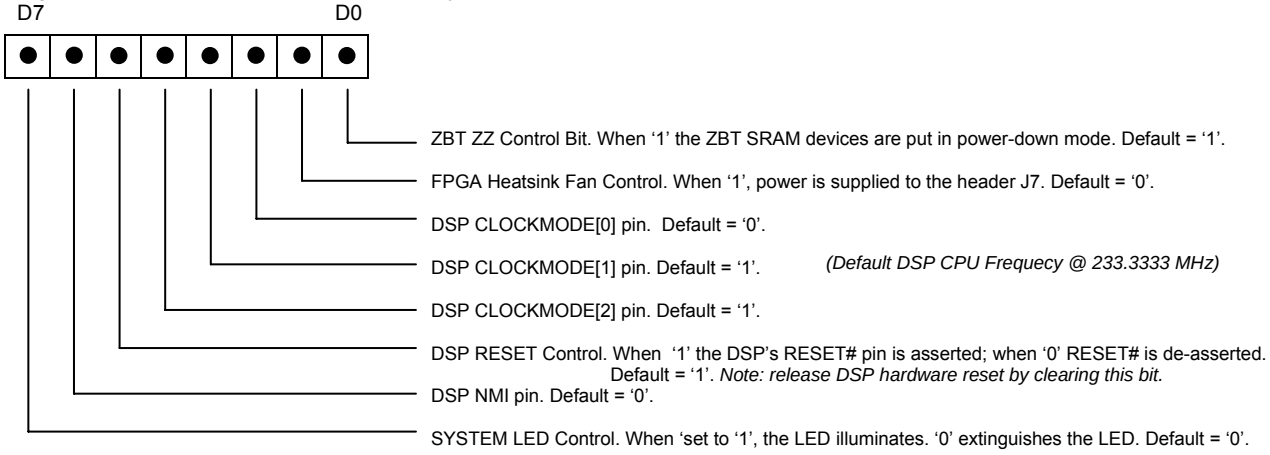


Figure 18: FPGA Temp Sensor Control Register. The D[7:0] bus accesses this register. Note: to access the FPGA Temp Sensor Control Register, bit CR[5] must be set. All bits are latched.

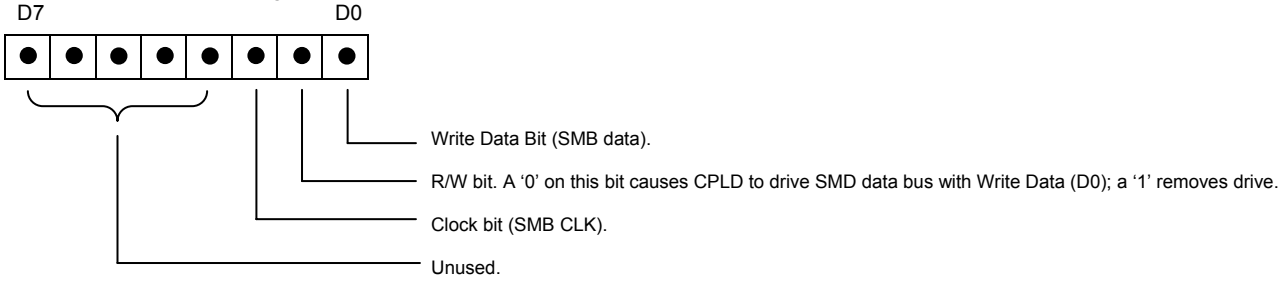


Figure 19: CPLD Status Register. Status bits are read on the CR bus during a CPLD read cycle.

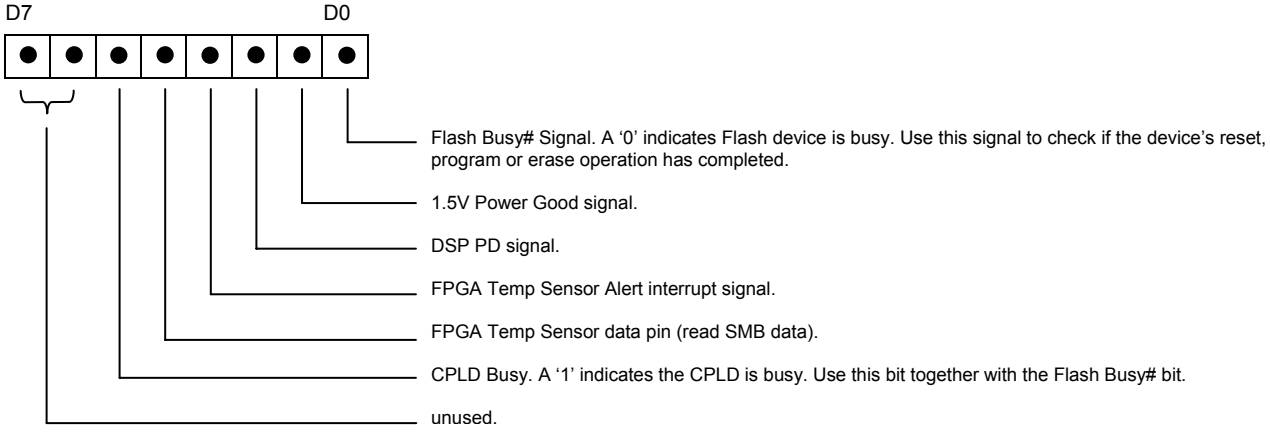
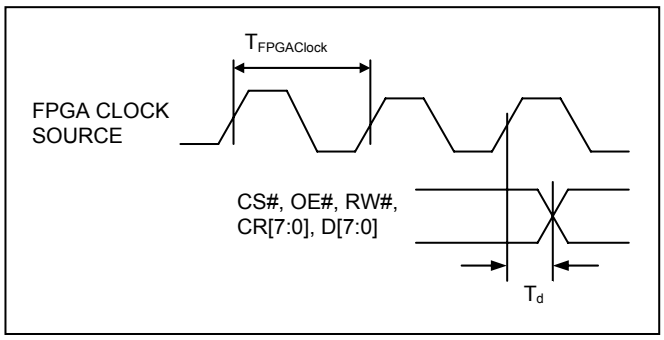
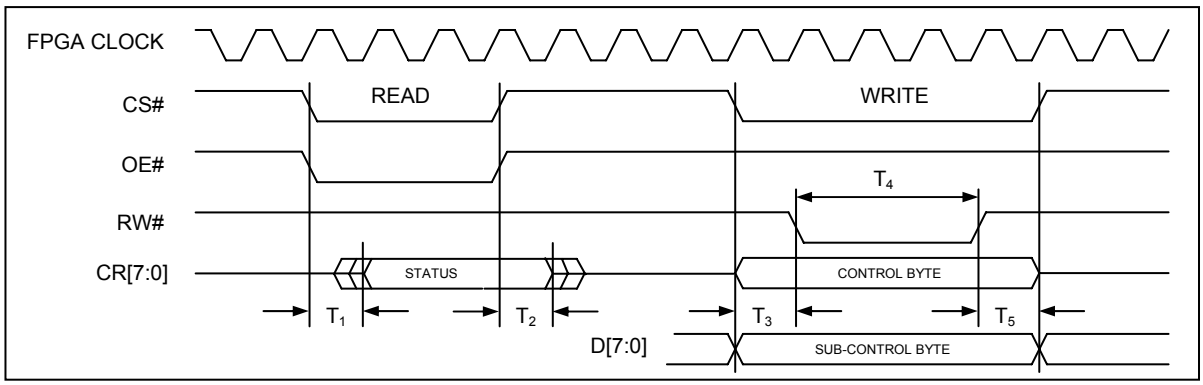


Figure 20: FPGA Clock and Output Signal Delay Timing Diagram



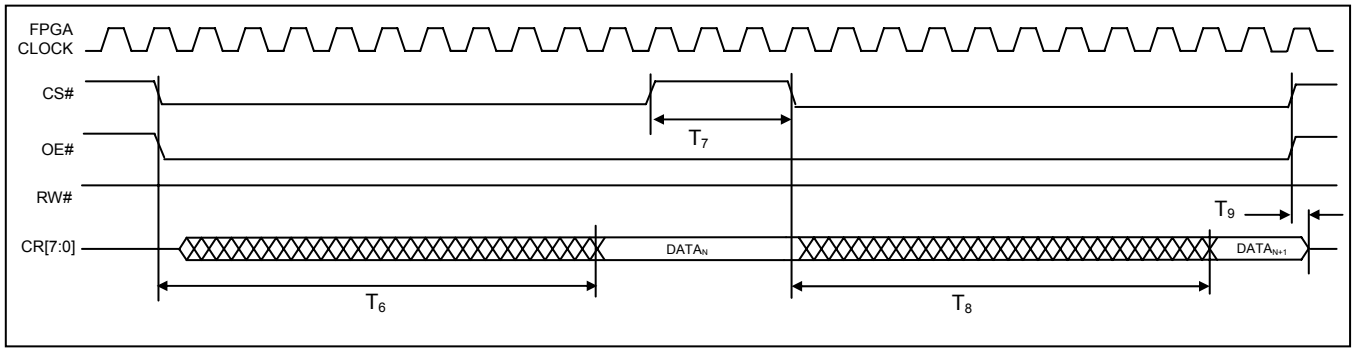
Symbol	Min	Typ	Max
$T_{FPGAClock}$	-	15.2588 ns	-
T_d	1 ns	-	10 ns

Figure 21: CPLD Status Read and Control Write Timing Diagram



Symbol	Min	Max	Symbol	Min	Max
T_1	-	12 ns	$T_{3,T5}$	1 * $T_{FPGAClock}$	-
T_2	0 ns	12 ns	T_4	3 * $T_{FPGAClock}$	-

Figure 22: CPLD Flash Read Timing Diagram



Symbol	Min	Max	Symbol	Min	Max
T_6, T_8	-	145 ns	T_9	0 ns	12 ns
T_7	3 * $T_{FPGAClock}$	-			

Table 15: Simplified Truth Table

CE#	OE#	R/W#	CR[7:0]	D[7:0]	Action
H	X	X	ZZh	XXh	none
L	L	H	read data	XXh	Status read / Flash read
L	H	L	command	command/write data	Command write

High-Speed Communication Link (HSCL)

Two independent high-speed multi-bit transmit/receive communication links HSCL-A and HSCL-B, implemented on the board, facilitate point-to-point connection with other PCI64-NP cards.

Sixteen National Semiconductor DS90LV001 devices provide the necessary LVDS buffering for eight receive channels, sourced from the two HSCL port connectors, and eight transmit channels driven by the FPGA. See Figure 23. The data sheet for the DS90LV001 device is available for download from www.national.com or from Support.

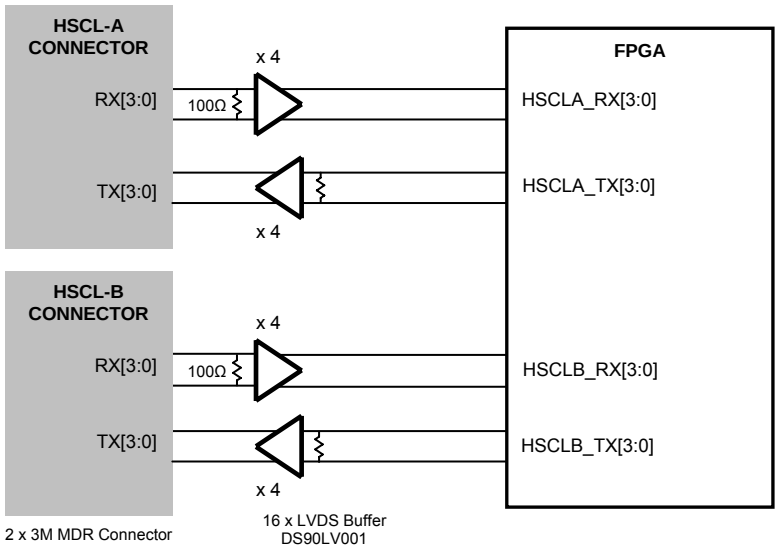


Figure 23: Connection of LVDS buffers routed to the HSCL connectors and the FPGA.

Note: the FPGA I/O LVDS differential pair names shown are reflected in the PCI64-NP.ucf file.

The connector used for the HSCL port s is a 3M Mini D Ribbon (MDR) receptacle. The pin-out for the two connectors is identical and is listed in Table 16.

A compatible cable type for use with the HSCL ports is given in Table 17. Note: the channels are crossed in the cable so that the TX[i] channel is wired to the corresponding RX[i] channel, e.g. TX0 differential pair is wired to the RX0 pair.

HSCL-A TX3 and HSCL-B TX0 transmit channels sourced from the FPGA are designated as clock signals. These signals can be used to clock data on the remaining three RX channels in each RX group. To help overcome signal skew problems, the two corresponding RX clock channels are delayed on the board, increasing data valid time before the clock signal transitions. Skew and clock delay data is given in Table 18.

Note: the HSCL-A or HSCL-B port must connect to the port with the same name in the remote card if TX0 and TX3 are used as clock sources, as described above.

Table 16: Pin-out for the HSCL-A and HSCL-B 3M Mini D Ribbon (MDR) Connectors

1 RX3+	6 RX0-	11 SHIELD	16 SHIELD	21 TX0-	26 TX3+
2 SHIELD	7 -	12 TX2+	17 RX1-	22 SHIELD	
3 RX2-	8 SHIELD	13 TX3-	18 RX0+	23 TX1+	
4 RX1+	9 TX0+	14 RX3-	19 SHIELD	24 TX2-	
5 SHIELD	10 TX1-	15 RX2+	20 TX3+	25 SHIELD	

Note: shields with pin numbers 8,11, 22, 25 are connected directly to ground on the board; the remaining shields are AC coupled to ground. The outer shield is connected to chassis ground via the I/O bracket.

Table 17: Compatible cable type to fit HSCL port connector

Manufacturer	Description	Length	Part No.
3M	26-pin MDR cable, F26-0 pinout, 8 shielded differential pairs, with outer shield	1 m	14526-EZ5B-100-02C

Table 18: Timing skew between channels in each RX and TX group due to PCB routing. Routed Clock delay is also specified.

HSCL	TX Group Max Skew (ps)*	RX Group Max Skew (ps)*	CLK Delay (ns)
HSCL-A	10	100	1.100 +/- 0.1
HSCL-B	10	100	1.045 +/- 0.1

* figure does not include skew due to LVDS buffers (see DS90LV001 datasheet)

The SIRL IP Core contains an interface for implementing the HSCL links.

Electrical Specification

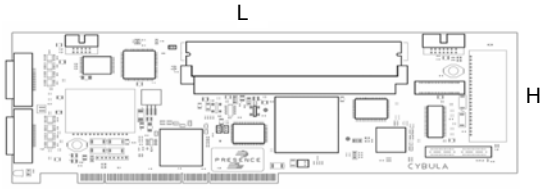
Required Supply Voltages:

Supply Voltage	Description	Max Current
5 V d.c.	Supply used to generate on-board +1.5V, +2.5V and +3.3V voltages	5A*
+12 V d.c.	Used by Heatsink Fan and mezzanine card (if fitted)	100 mA
-12 V d.c.	Used by mezzanine card (if fitted)	-100 mA

*Note: maximum total power consumption must not exceed 25 Watts. If an installed mezzanine card consumes power from the +/-12V supplies, the maximum 5V current draw must be derated.

Mechanical Specification

Board dimensions:



Dimension	Comment	Value (mm)
L	Retainer bracket not fitted (not shown above)	312
H	DIMM 1 installed only, with low-profile module	107
H	DIMM 2 (and DIMM 1) installed, with low-profile module	120

Environmental Specification

Parameter	Operating	Storage
Temperature	5ºC to +50ºC	-40ºC to +65ºC
Humidity	10% - 90% non-condensing	

Regulatory Body Approvals/Compliance

Description	Country	Compliance
Electromagnetic Compatibility	USA	FCC Part 15, Class A and B
	Europe	EN55022, EN61000-4

Appendix

Figure A1: PCI64-NP LAYOUT