The pcda8 FPGA Configuration File

The `pcda8.bit` FPGA configuration file is firmware intended for the UI Xilinx on the PCI SS/GS main board when used with the ECL mezzanine board. This document describes the connector pinout and registers that it defines. Information on the mezzanine board itself can be found in:

PCI SS/GS ECL User's Guide www.edt.com/manuals/PCD/ecl.pdf

Copyright, Trademarks, Disclaimer

February 27, 2007
008-02776-00

Connector Pinouts

Table 1 describes the connection from the PCI SS/GS board to the connector, with the ECL mezzanine board, when loaded with with pcda8.bit.

The board uses a high-density 68-pin SCSI-type I/O connector (Tyco part number 787169-7), with a straight-shielded backshell (Tyco part number 750752-1). You can use a typical SCSI cable (Tyco part number 749621-7) if your equipment has a SCSI connector.

NOTE Do not connect your own circuits to the unused pins, as they may be internally connected.

Table 1. pcda8.bit Connector Pinout

Pin	Signal	Pin	Signal
1	TXT+	35	TXT–
2	DATI0+	36	DATI0–
3	DATI1+	37	DATI1–
4	DATI2+	38	DATI2–
5	DATI3+	39	DATI3–
6	DATI4+	40	DATI4–
7	DATI5+	41	DATI5–
8	DATI6+	42	DATI6–
9	DATI7+	43	DATI7–
10	DATO0+	44	DATO0–
11	DATO1+	45	DATO1–
12	STAT3+	46	STAT3–
13	DATO2+	47	DATO2–
14	DATO3+	48	DATO3–
15	DATO4+	49	DATO4–
16	DATO5+	50	DATO5–
17	FUNCT1+	51	FUNCT1–
18	FUNCT2+	52	FUNCT2–
19	DATO6+	53	DATO6–
20	DATO7+	54	DATO7–
21	SPARE0+	55	SPARE0–
22	STAT0+	56	STAT0–
23	FUNCT0+	57	FUNCT0–
24	spare	58	spare
25	DNR+	59	DNR–
26	IDV+	60	IDV–
27	STAT1+	61	STAT1–
28	STAT2+	62	STAT2–
29	RXT+	63	RXT–
30	FUNCT3+	64	FUNCT3–
31	SENDT+	65	SENDT–
32	ODV+	66	ODV–
33	BNR+	67	BNR–
34	ground	68	ground

Loopback Connector for Testing

The ECL mezzanine board and the PCI SS/GS main boards come with testing files to allow you to conduct a loopback test of the board. Files are listed in the section entitled Included Files in your board manual.

Testing instructions are provided in the file entitled Testing Procedures.

You can also test your application using a loopback connector, which you can make yourself, following the instructions below, or order from EDT.

To make a loopback connector:

1. Wire data inputs to data outputs.
2. Wire TXT to RXT.
3. Wire DNR to BNR
4. Wire IDV to ODV.
5. Wire STAT to FUNCT.
6. Leave SENDT unconnected.

Registers

These registers are implemented in the firmware pcda8.bit.

Applications can access the ECL registers through the DMA library routines especially edt_reg_read() and edt_reg_write(), using the symbolic names listed under "Access" for each register.

The following registers are implemented but not used:

- Direction Control registers at address 0x06 and 0x0
- LED Control at address 0x30

Command Register

Size	8-bit
I/O	read-write
Address	0x00
Access	PCD_CMD

Bit	Name	Description
4–7	STAT_INT_EN	A value of one enables the corresponding STAT bit to cause an interrupt when it is asserted.
3	ENABLE	Set to one to enable the ECL interface. This bit is set after the direction is chosen and typically after the first DMA buffer is ready. To reset direction or flags, toggle this bit. To flush the DMA FIFOs, clear then set this bit.
2	INV	When set, inverts the bits in each byte, so that ones become zeroes and zeroes become ones.
1	FORCEBNR	A value of 1indicates that the board is not ready.
0	DIR	A value of 1 indicates that data is coming in to the ECL. A value of 0 indicates that data is going out from the ECL.

Data Path Status Register

Size 8-bit

I/O read-only

Address 0x01

Access PCD_DATA_PATH_STAT

Bit	Name	Description
7	SP_IN	Reads the state of the spare input (pins 24 and 58).
6	INFFAFULL	If set, input FIFO is almost full.
4–5	INFFULL	If set, input FIFO is full.
3	OVERFLOW	This bit is asserted when the input FIFO is full and the IDV signal is high. Reset this bit with the ENABLE bit in the Command Register on page 4.
2	UNDERFLOW	If the DNR signal is low and the ODV signal goes low because the output FIFO runs out of data, then this bit is asserted and remains so throughout the data transfer. Reset this bit with the ENABLE bit in the command register.
1		Reserved; always reads zero.
0	OF_NOT_EMP	If this bit is set, the output FIFO is not empty.

Funct Register

Size 8-bit

I/O read-write

Address 0x02

Access PCD_FUNCT

Bit	Name	Description
7	PLLCLK	Set to enable the PLL output clock to be used as the TXT clock. Clear to allow the output clock to be selected according to the SELRXT bit (bit 1) in the Interface Configuration Register.
6–4		not used
0–3	FUNCT	Sets the state of the user-definable FUNCT outputs.

Stat Register

Size 8-bit

I/O read-only

Address 0x03

Access PCD_STAT

Bit	Name	Description
7–4	STAT_INT	Interrupt bits for the status bits. To cause a PCI Bus interrupt, use the EDT DMA library routine `edt_set_event_func`, which uses the following bits: If the following conditions are both true, then the corresponding bit of these four can be asserted to cause a PCI Bus interrupt: • The device interrupt is enabled using the STAT_INT_ENA bit in the Stat Polarity Register. • The corresponding bit is asserted in the Command Register (one of bits 7–4, named STAT_INT_EN). The PCI Bus interrupt is then caused when the corresponding STAT signal is asserted according to the polarity specified in the Stat Polarity Register. To reset the interrupt, disable and re-enable the appropriate STAT_INT_EN bit in the Command Register.
3–0	STAT	The state of user-definable STAT input signals as last sampled by the RXT clock signal.

Stat Polarity Register

Size 8-bit

I/O read-write

Address 0x04

Access PCD_STAT_POLARITY

Bit	Name	Description
7–5		not used
4	STAT_INT_ENA	Provides global enable or disable for all interrupt bits in Stat Register on page 6, allowing the driver to disable and re-enable them in one operation, without altering the state of the Stat register. This bit is used mainly by the driver to disable the Stat interrupts to determine which other interrupts are pending. A value of 1 enables the interrupts.
3–0	POLARITY	A value of 0 indicates that a change from 0 to 1 from one clock cycle to the next causes an interrupt in bits 7–4 of the Stat Register on page 6, if the corresponding STAT_INT_EN bit is also enabled in the Command Register on page 4. A value of 1 causes the same event when the STAT_INT bit changes from 1 to 0 from one clock cycle to the next.

Programmed I/O High Register

Size	8-bit
I/O	read-write
Address	0x09
Access	PCD_PIO_OUTHI

Bit	Description
7–0	Outputs the data byte. Write to this register with a byte of data to output. When the byte is output, an ODV signal (Output Data Valid) is also output for one TXT clock cycle. Read the programmed I/O directly from I/O devices. INV (bit 2 of the Command Register) and REVBITS (bit 7 of the Interface Configuration Register) have no effect on the input.

Interface Configuration Register

Size 8-bit

I/O read-write

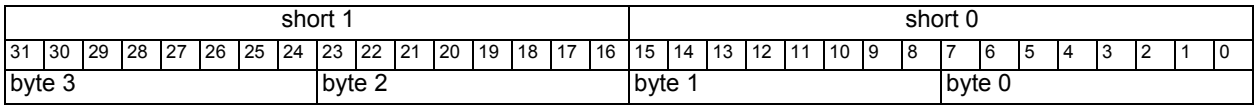
Address 0x0F

Access PCD_CONFIG

Bit	Name	Description
7	REVBITS	Reverses the order of the bits within each byte.
6	PIOEN	Enables programmed I/O. A value of 1 translates DMA channel buffers and enables the Programmed I/O Low Register and the Programmed I/O High Register. Write the desired 16-bit word, the low eight bits first to the Programmed I/O Low Register, and then the high eight bits to the Programmed I/O High Register. When the Programmed I/O High Register is written to, the firmware generates an ODV pulse in mid-clock, to enable the device to latch the data.
5	SETDNR	Set this bit to stop transfer to the device, as if the device had set DNR.
4	DED	Disable output delay. If set, may cause ODV transitioning on DMA start and underflows.
3	SHORTSWAP	Set to 1 if the host computer writes the first 16-bit word on bits 16–31 of the PCI data bus (bigendian format) instead of bits 0–15 as defined in the PCI Bus specification. See Figure 1 for the details of data word structure.
2	EN_DNR	Set to enable DNR output.
1	SELRXT	Set to use the RXT input clock as the output clock; clear to use the 40 MHz internal oscillator as the output clock. If PLLCLK (bit 7) is set in the Funct Register, this bit is ignored.
0	BYTESWAP	A value of 1 swaps the order of bytes in a 16-bit word of data coming in from the data source. See Figure 1 for the details of data word structure.

Figure 1 shows the structure of a 32-bit data word, with no swapping in effect. With SHORTSWAP set, short 0 appears before short 1. With BYTESWAP set, byte 2 appears before byte 3, and byte 0 before byte 1. With both set, byte 0 appears first, followed by byte 1, byte 2, and finally byte 3.

Figure 1. Data Word Structure Without Swapping



PLL Programming Register

Size 8-bit

I/O read-write

Address 0x20

Access EDT_SS_PLL_CTL

Comment The program `set_ss_vco` uses this register to program the serial interface of the PLL.

Bit	Name	Description
7	PLL_SCLK	Connected to the PLL serial clock input.
6	PLL_DATA	Connected to the PLL serial data input.
5–4		not used
3–0	PLL_STROBE	Connected to the strobe inputs of PLL 0 (PCI CDa) or PLLs 0–3 (PCI SS/GS).

PLL Divider Register

Size 16-bit

I/O read-only

Address 0x24, 0x25

Access EDT_SS_PLL0_CLK

Comment The program `set_ss_vco` sets this register.

Bit	Description
15–0	Programmable post-scalar divider to set PLL frequency, to achieve lower frequencies than those to which you can program the PLL directly. After this division, the clocks are divided by two to even the duty cycle.

NOTE For application portability, addresses 0x26 and 0x27 also program PLL0.

Output Data Valid Delay Register

Size 8-bit

I/O read-write

Address 0x28

Access ODV_DELAY

Bit	Description
7–0	Set the number of 8-bit words by which to delay output. The specified number of outgoing words accumulate in the FIFO, reducing or eliminating ODV transitioning on startup.

Output State Machine Register

Size 8-bit

I/O read only

Address 0x29

Access PCD_OUTPUT_STATE

Bit	Description
7–6	Not used; always read zero.
5–0	Reads the state of the PCD8 output state machine, for testing and debugging.

Main Board FPGA Configuration File Design ID Register

Size 16-bit

I/O read only

Address 0x7C, 0x7D

Access PCD_DESIGN_ID

Bit	Description
15–0	A sixteen-bit number assigned by the organization that produced the FPGA configuration file loaded in the main board UI Xilinx. (EDT uses the top eight bits only.) The design ID for `pcda8.bit` is 0x0200.

Main Board Configuration File Version String Register

Use this register to read the FPGA configuration file version string from ROM. Write the ROM address to the register and read the ASCII data from the same register. The version string is a maximum of 64 bytes long, so only the first six bits of the address are significant.

Size 8-bit

I/O read-write

Address 0x7E

Access MAIN_BITFILE_VERSION

Bit	Name	Description
7–0	ID_ADD_DATA	Write an address to read ROM contents. Result is *mainBoard_mezzBoard_bitfileName version.revision mm/dd/yyyy* (*number of DMA channels used*, *number of DMA channels required by the PCI Xilinx*). The date given is the date the FPGA configuration file was created. Placeholders in italics are replaced by actual values — for example, `cd1_none_pcda8 1.1 10/08/2006 (1,1)`.

Board ID Register

Size	8-bit
I/O	read-write
Address	0x7F
Access	EDT_BOARDID
Comment	Returns a unique four-bit code corresponding to the mezzanine board installed. A value of 2 indicates an extended board ID. To read an extended board ID code, use the application `extbdid.exe` or the EDT DMA library routine `edt_get_boardID`.

Bit	Name	Description
7–5		used by `extbdid.exe`
4		not used; always set
3–0	BOARD_ID	The ID code of the installed mezzanine board:
		12 3x3G
		11 OC192
		10 16TE3
		F Combo I/O, ECL
		E Combo II I/O, RS-422
		D Combo III I/O, ECL
		C Combo III I/O, LVDS
		B Combo III I/O, RS-422
		A SRXL (with Graychips)
		9 TLK1501 I/O
		8 ECL I/O
		7 Combo II I/O, LVDS
		6 OCM
		5 HRC for E4, STM-1, OC3
		4–2 reserved
		1 LVDS I/O
		0 RS-422 I/O