

---

# The two\_bitx4ch FPGA Configuration File

The `two_bitx4ch.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](http://www.edt.com/manuals/PCD/ecl.pdf)

## Copyright, Trademarks, Disclaimer

The information in this document is subject to change without notice and does not represent a commitment on the part of Engineering Design Team, Inc. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the agreement.

Engineering Design Team, Inc. ("EDT"), makes no warranties, express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose, regarding the software described in this document ("the software"). EDT does not warrant, guarantee, or make any representations regarding the use or the results of the use of the software in terms of its correctness, accuracy, reliability, currentness, or otherwise. The entire risk as to the results and performance of the software is assumed by you. The exclusion of implied warranties is not permitted by some jurisdictions. The above exclusion may not apply to you.

In no event will EDT, its directors, officers, employees, or agents be liable to you for any consequential, incidental, or indirect damages (including damages for loss of business profits, business interruption, loss of business information, and the like) arising out of the use or inability to use the software even if EDT has been advised of the possibility of such damages. Because some jurisdictions do not allow the exclusion or limitation of liability for consequential or incidental damages, the above limitations may not apply to you. EDT's liability to you for actual damages for any cause whatsoever, and regardless of the form of the action (whether in contract, tort [including negligence], product liability or otherwise), will be limited to \$50 (fifty U.S. dollars).

No part of this manual may be reproduced or transmitted in any form or by any means, electronic or mechanical, without the express written agreement of Engineering Design Team, Inc.

Copyright © 2007 Engineering Design Team, Inc. All rights reserved.

EDT and Engineering Design Team are trademarks of Engineering Design Team, Inc.

Xilinx is a registered trademark of Xilinx, Inc.

**February 21, 2007**

008-02778-00



a HEICO company

## 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 `two_bitx4ch.bit`. Channels 4–7 are used for testing only.

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. ECL (`two_bitx4ch.bit`) Connector Pinout**

Pin	Channel	Signal	Pin	Channel	Signal
1	7	IDV+	35	7	IDV–
2	0	DATI1+	36	0	DATI1–
3	0	DATI0+	37	0	DATI0–
4	0	CLK+	38	0	CLK–
5	0	IDV+	39	0	IDV–
6	1	DATI1+	40	1	DATI1–
7	1	DATI0+	41	1	DATI0–
8	1	CLK+	42	1	CLK–
9	1	IDV+	43	1	IDV–
10	2	DATI1+	44	2	DATI1–
11	2	DATI0+	45	2	DATI0–
12	4	IDV+	46	4	IDV–
13	2	CLK+	47	2	CLK–
14	2	IDV+	48	2	IDV–
15	3	DATI1+	49	3	DATI1–
16	3	DATI0+	50	3	DATI0–
17	5	DATI0+	51	5	DATI0–
18	5	CLK+	52	5	CLK–
19	3	CLK+	53	3	CLK–
20	3	IDV+	54	3	IDV–
21		not used	55		not used
22	4	DATI1+	56	4	DATI1–
23	5	DATI1+	57	5	DATI1–
24	6	IDV+	58	6	IDV–
25	6	CLK+	59	6	CLK–
26	6	DATI0+	60	6	DATI0–
27	4	DATI0+	61	4	DATI0–
28	4	CLK+	62	4	CLK–
29	6	DATI1+	63	6	DATI1–
30	5	IDV+	64	5	IDV–
31	7	DATI1+	65	7	DATI1–
32	7	DATI0+	66	7	DATI0–
33	7	CLK+	67	7	CLK–
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 document entitled [Testing Procedures](#).

You can also test your application by using channels 4–7 to generate a test pattern. Channels 0–3 can then receive the pattern as input using a loopback connector. You can make your own loopback connector or order one from [EDT](#).

To make a loopback connector, wire the channels that you wish to test as described below:

- Wire channel 4 to channel 0.
- Wire channel 5 to channel 1.
- Wire channel 6 to channel 2.
- Wire channel 7 to channel 3.

The example application `ecl_snap.c` contains an example of using the built-in pattern generator to output a test pattern in this way.

## Registers

The following registers are implemented in the firmware `two_bitx4ch.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 Underflow register `SSD16_UNDER` at addresses `0x18` and `0x19` is implemented but not used. It always reads zero.

### Command Register

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

Bit	PCD_	Description
4–7		not used
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.
2–0		not used

### Input Data Valid Enable Register

Size	16-bit
I/O	read-write
Address	0x08 and 0x09
Access	SSD16_ENVALID

Bit	Description
15–4	not used
3–0	Enables IDV for the corresponding channel.

## Interface Configuration Register

Size	8-bit
I/O	read-write
Address	0x0F
Access	PCD_CONFIG

Bit	PCD_	Description
7–4		not used
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 <a href="#">Figure 1</a> for the details of data word structure.
2–1		not used
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 <a href="#">Figure 1</a> 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**

short 1																short 0															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
byte 3								byte 2								byte 1								byte 0							

## Channel Enable Register

Size	16-bit
I/O	read-write
Address	0x10 and 0x11
Access	SSD16_CHEN

Bit	Name	Description
15–4		not used
3–0	CH_ENABLE	A value of one in a bit enables the corresponding channel for DMA.

**Channel Direction Register**

Size	16-bit
I/O	read-write
Address	0x12 and 0x13
Access	SSD16_CHDIR

Bit	Name	Description
15–4		not used
3–0	CH_DIR	A value of zero in a bit enables input for the corresponding DMA channel; a value of one enables output.

**Clock Edge Register**

Size	16-bit
I/O	read-write
Address	0x14 and 0x15
Access	SSD16_CHEDGE

Bit	Name	Description
15–4		not used
3–0	EDGE	For input channels, a value of one in a bit indicates that the corresponding channel latches data on the rising edge of its clock. a value of 0 indicates that it latches data on the falling edge.  For output channels, a value of one indicates that a new bit is output on the rising edge, and a value of zero that it's output on the falling edge.

**Least Significant Bit First Register**

Size	16-bit
I/O	read-write
Address	0x16 and 0x17
Access	SSD16_LSB

Bit	Name	Description
15–4		not used
3–0	LSB_FIRST	When set for a channel, the least significant bit of each 8-bit data byte is the first bit, and the most significant bit is the last. When clear for a channel, the most significant bit of the byte is the first bit.

## Overflow Register

Size	16-bit
I/O	read-write
Address	0x1A and 0x1B
Access	SSD16_OVER

Bit	Name	Description
15–4		not used
3–0	OVERFLOW	A value of 1 in a bit indicates that the corresponding channel's internal FIFO has overflowed since the previous CMD_EN or CHANNEL_ENABLE. Reset by writing a one to the desired channel bit of this register, or by first disabling, then re-enabling, the channel (see the <a href="#">Channel Enable Register</a> ).

## Data Invert Register

Size	16-bit
I/O	read-write
Address	0x1C and 0x1D
Access	SSD16_CHINVERT

Bit	Name	Description
15–4		not used
3–0	CH_INVERT	A value of 1 in a bit indicates that the corresponding channel's data is inverted — a one becomes a zero, and vice-versa.

## Input Data Valid Polarity Register

Size	16-bit
I/O	read-write
Address	0x1E and 0x1F
Access	SSD16_ENINVERT

Bit	Description
15–4	not used
3–0	Determines the active ploarity of the input data valid signal (IDV). When clear, a value of one on IDV indicates valid data; when set, a value of zero on IDV indicates valid data.  For this bit to have an effect, IDV for the channel must be enabled in the <a href="#">Input Data Valid Enable Register</a> .

**PLL Programming Register**

Size	8-bit
I/O	read-write
Address	0x20
Access	EDT_SS_PLL_CTL
Comment	The program <code>set_ss_vco</code> uses this register to program the serial interface of the four PLLs.

Bit	Name	Description
15–4		not used
0	PLL_STROBE	Connected to the strobe inputs of the phase-locked loop.

**PLL 0 Divider Register**

Size	16-bit
I/O	read-write
Address	0x24 and 0x25
Access	EDT_SS_PLL0_CLK
Comment	This register is set by <code>set_ss_vco</code> .

Bit	Name	Description
15–0	PLL0_DIV	A post-scalar divider used to achieve lower frequencies than those at which the PLLs can be programmed. After this division (if any), the clocks are divided by two for an even duty cycle — half the time high, and half low. <code>set_ss_vco</code> takes this into account.  All test generators use PLL0 for the output clock.