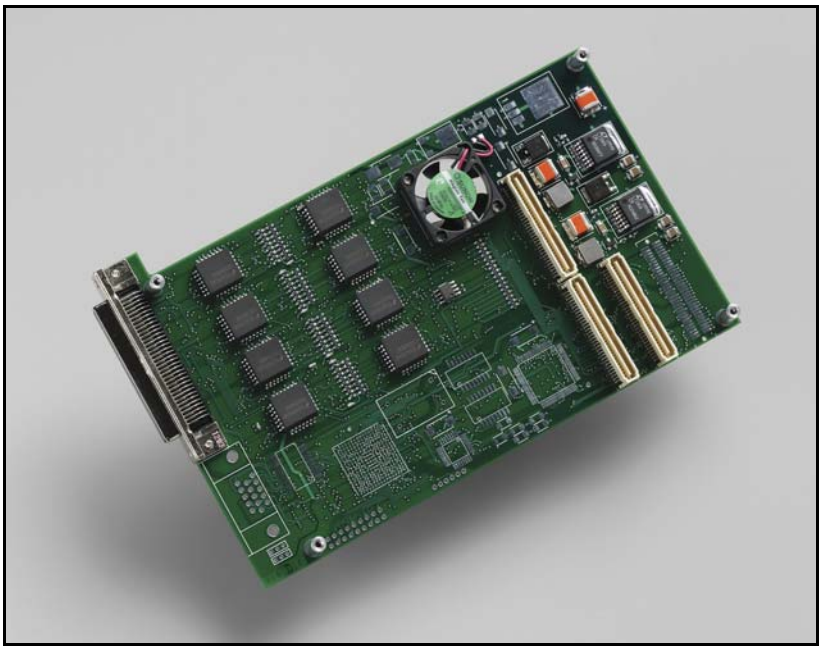
edt
a HEICO company

User's Guide

ECL Mezzanine Board



For PCI / PCI Express

Rev. 2013 March 07
Doc. 008-02773-02

EDT | Engineering Design Team, Inc.
3423 NE John Olsen Ave.
Hillsboro, OR 97124
U.S.A.
Tel: +1-503-690-1234 | Toll free (in U.S.A.): 800-435-4320
Fax: +1-503-690-1243
www.edt.com

TERMS OF USE

Definitions. This agreement, between Engineering Design Team, Inc. ("Seller") and the user or distributor ("Buyer"), covers the use and distribution of the following items provided by Seller: a) the binary and all provided source code for any and all device drivers, software libraries, utilities, and example applications (collectively, "Software"); b) the binary and all provided source code for any and all configurable or programmable devices (collectively, "Firmware"); and c) the computer boards and all other physical components (collectively, "Hardware"). Software, Firmware, and Hardware are collectively referred to as "Products." This agreement also covers Seller's published Limited Warranty ("Warranty") and all other published manuals and product information in physical, electronic, or any other form ("Documentation").

License. Seller grants Buyer the right to use or distribute Seller's Software and Firmware Products solely to enable Seller's Hardware Products. Seller's Software and Firmware must be used on the same computer as Seller's Hardware. Seller's Products and Documentation are furnished under, and may be used only in accordance with, the terms of this agreement. By using or distributing Seller's Products and Documentation, Buyer agrees to the terms of this agreement, as well as any additional agreements (such as a nondisclosure agreement) between Buyer and Seller.

Export Restrictions. Buyer will not permit Seller's Software, Firmware, or Hardware to be sent to, or used in, any other country except in compliance with applicable U.S. laws and regulations. For clarification or advice on such laws and regulations, Buyer should contact: **U.S. Department of Commerce, Export Division, Washington, D.C., U.S.A.**

Limitation of Rights. Seller grants Buyer a royalty-free right to modify, reproduce, and distribute executable files using the Seller's Software and Firmware, provided that: a) the source code and executable files will be used only with Seller's Hardware; b) Buyer agrees to indemnify, hold harmless, and defend Seller from and against any claims or lawsuits, including attorneys' fees, that arise or result from the use or distribution of Buyer's products containing Seller's Products. Seller's Hardware may not be copied or recreated in any form or by any means without Seller's express written consent.

No Liability for Consequential Damages. In no event will Seller, its directors, officers, employees, or agents be liable to Buyer for any consequential, incidental, or indirect damages (including damages for business interruptions, loss of business profits or information, and the like) arising out of the use or inability to use the Products, even if Seller 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 Buyer. Seller's liability to Buyer for actual damages for any cause whatsoever, and regardless of the form of the action (whether in contract, product liability, tort including negligence, or otherwise) will be limited to fifty U.S. dollars ($50.00).

Limited Hardware Warranty. Seller warrants that the Hardware it manufactures and sells shall be free of defects in materials and workmanship for a period of 12 months from date of shipment to initial Buyer. This warranty does not apply to any product that is misused, abused, repaired, or otherwise modified by Buyer or others. Seller's sole obligation for breach of this warranty shall be to repair or replace (F.O.B. Seller's plant, Beaverton, Oregon, USA) any goods that are found to be non-conforming or defective as specified by Buyer within 30 days of discovery of any defect. Buyer shall bear all installation and transportation expenses, and all other incidental expenses and damages.

Limitation of Liability. *In no event shall Seller be liable for any type of special consequential, incidental, or penal damages, whether such damages arise from, or are a result of, breach of contract, warranty, tort (including negligence), strict liability, or otherwise.* All references to damages herein shall include, but not be limited to: loss of profit or revenue; loss of use of the goods or associated equipment; costs of substitute goods, equipment, or facilities; downtime costs; or claims for damages. Seller shall not be liable for any loss, claim, expense, or damage caused by, contributed to, or arising out of the acts or omissions of Buyer, whether negligent or otherwise.

No Other Warranties. Seller makes no other warranties, express or implied, including without limitation the implied warranties of merchantability and fitness for a particular purpose, regarding Seller's Products or Documentation. Seller does not warrant, guarantee, or make any representations regarding the use or the results of the use of the Products or Documentation or their correctness, accuracy, reliability, currentness, or otherwise. All risk related to the results and performance of the Products and Documentation is assumed by Buyer. The exclusion of implied warranties is not permitted by some jurisdictions. The above exclusion may not apply to Buyer.

Disclaimer. Seller's Products and Documentation, including this document, are subject to change without notice. Documentation does not represent a commitment from Seller.

Contents

ECL Mezzanine Board
Overview 5
Related Resources 5
About the Software and Firmware 6
FPGA Configuration Files 6
ECL-specific Example Applications 6
Software Initialization Files 6
The PCD Device Driver 7
FPGA Configuration Files 7
Software Initialization Files 7
Sample Applications and Utilities 8
Sample Applications 8
Utility Files 8
Testing Files 9
Building Applications 9
Configuring the ECL 9
Checking the PCI FPGA Firmware 9
Loading the UI FPGA Firmware and Configuring the ECL 10
Using Custom FPGA Configuration Files 11
Revision Log 12

ECL Mezzanine Board

Overview

The ECL mezzanine board connects to an EDT main board to provide 32 differential ECL interface signals. Each group of four signals can be either inputs or outputs. The function of these signals depends on the FPGA configuration file loaded into the main board.

The ECL is bidirectional, but the internal input and output data paths are shared. Loopback tests, therefore, can test its data paths only from the host to the main board connector pins.

The ECL can also be built to use RS422 or LVDS signal levels — order the RS422-E or LVDS-E option. This may seem odd, but it allows you to use FPGA configuration files for the ECL without modification in RS422 or LVDS systems.

The ECL can also be built using combinations of signal levels in groups of four; contact EDT for details.

A list of the firmware files provided is available in About the Software and Firmware on page 6, and instructions for loading them are provided in Configuring the ECL on page 9.

Related Resources

The resources below may be helpful or necessary for your applications.

EDT Resources

Description	*Detail*	*Web link*
• Documentation for each particular product	Datasheets and user's guides	www.edt.com (find product page)
• Application programming interface (API)	HTML and PDF versions	" (Product Documentation)
• Installation packages (Windows, Linux, Mac)	Software / firmware downloads	" (Product Documentation)
• FPGA configuration files for this product	At www.edt.com/manuals...	...look for user's guides for .bit files... eclssd16.bit two_bitx4ch.bit pcda8.bit

Third-Party Standards / Specifications

Description	*Pertains to*	*Documentation*	*Web link*
• PCI / PCIe	PCI / PCIe bus	PCI Special Interest Group (PCI SIG)	www.pcisig.com

About the Software and Firmware

The following ECL-specific files are included...

FPGA Configuration Files

`pciss16.bit`	PCI FPGA configuration file for the PCI SS main board, and for use with the UI FPGA configuration file `eclssd16.bit`.
`pcigs16.bit`	PCI FPGA configuration file for the PCI GS main board, and for use with the UI FPGA configuration file eclssd16.bit.
`eclssd16.bit`	UI FPGA configuration file for 16-channel synchronous serial input/output.

and:

`pciss4.bit`	PCI FPGA configuration file for the PCI SS main board, and use with the UI FPGA configuration file `two_bitx4ch.bit`.
`pcigs4.bit`	PCI FPGA configuration file for the PCI GS main board, and use with the UI FPGA configuration file `two_bitx4ch.bit`.
`two_bitx4ch.bit`	UI FPGA configuration file for two-bit, four-channel synchronous parallel input.

and:

`pciss1.bit`	PCI FPGA configuration file for the PCI SS main board, and for use with the UI FPGA configuration file `pcda8.bit`.
`pcigs1.bit`	PCI FPGA configuration file for the PCI GS main board, and for use with the UI FPGA configuration file `pcda8.bit`.
`pcda8.bit`	UI FPGA configuration file for eight-channel synchronous parallel input/output.

ECL-specific Example Applications

`ecl_snap`	An example application that generates a test pattern from the dedicated output channels 4–7 and receives them as input in channels 0–3.
`ecl_snap.c`	C source for the example application.

Software Initialization Files

Sample software initialization files for all board configurations are in the `pcd_config` subdirectory of the distribution directory.

Software initialization files are editable text files that you can customize for your own applications.

`eclssd16.cfg`	Software initialization file for use with `eclssd16.bit`.
`two_bitx4ch.cfg`	Software initialization file for use with `two_bitx4ch.bit`.
`pcda8.cfg`	Software initialization file for use with `pcda8.bit`.

The firmware file names you see in the EDT distribution do not match the file names given above because PCI Bus slots come in two varieties: those supplying 3 V power, and those supplying 5 V power. Different

firmware is required for the two kinds of slots, but the correct firmware file is chosen automatically when you run `pciload` or any other EDT-supplied firmware loading utility.

For example, you may see files named `cda16_3v.bit` and `cda16_5v.bit`, but the correct argument to supply to load the firmware is `cda16.bit`.

In some cases, you may also see additional firmware files incorporating changes required for various board revisions, or files with the same name in different subdirectories. You need not be concerned with any of these variations of name or path, however. In all cases, the names given above are the correct arguments to supply to the firmware-loading utilities.

The PCD Device Driver

The PCD device driver is the software running on the host that allows the host operating system to communicate with the ECL. The driver is loaded into the kernel upon installation, and thereafter runs as a kernel module. The driver name and subdirectory is specific to each supported operating system; the installation script handles those details for you, automatically installing the correct device driver in the correct operating system-specific manner.

FPGA Configuration Files

FPGA configuration files define the firmware required for the PCI FPGA and the UI FPGA. The PCI FPGA firmware files are in the `flash` subdirectory of the EDT top-level distribution directory. UI FPGA firmware files are in the `bitfiles` subdirectory of the EDT top-level distribution directory.

Each FPGA must be loaded with the firmware specific to the chosen interface, and the firmware in one FPGA must be compatible with the firmware in the other. By default, the correct FPGA configuration file for the PCI FPGA is loaded atthe factory. However, you'll need to load the required FPGA configuration file for the UI FPGA yourself.

The firmware files specific to your ECL are listed at the beginning of this section. Instructions for loading them are provided in Configuring the ECL.

Software Initialization Files

Software initialization files (having the extension `.cfg`) are editable text files that run like scripts to configure EDT boards so that they are ready to perform DMA. The commands in a software initialization file are defined in a C application named `initpcd`. When you invoke `initpcd`, you specify which software initialization file to use with the `-f` flag.

A typical software initialization file loads an FPGA configuration file into the UI FPGA and sets up various registers to prepare the board for DMA transfers. Some software initialization files may also load an FPGA configuration file into an FPGA residing on the mezzanine board.

A variety of software initialization files are included with the EDT software, at least one of which is customized for each main board or main board / mezzanine board combination — that is, each FPGA configuration file has a matching software initialization file. Software initialization files are located in the `pcd_config` subdirectory of the EDT top-level distribution directory. The software initialization files specific to your ECL are listed at the beginning of this section. Instructions for their use are provided in Configuring the ECL.

Commands defined in `initpcd` and typically found in software initialization files allow for specific FPGA configuration files to be loaded (for example, `bitfile:`), write specified hexadecimal values to specified registers (for example, `command_reg:`), enable or disable byte-swapping or short-swapping to accommodate different operating systems' requirements for bit ordering (for example, `byteswap:`), or invoke arbitrary commands (for example, `run_command:`). For example:

```
bitfile: ssd16io.bit
command_reg: 0x08
byteswap: 1
run_command: set_ss_vco -F 1000000 2
```

For complete usage details, enter `initpcd --help`.

C source for `initpcd` is included so that you can add your own commands, if you wish. You can then edit your own software initialization file to use your new commands and specify that `initpcd` use your new file when configuring your board. If you would like us to include your new software initialization commands in subsequent releases of `initpcd`, send mail to `tech@edt.com`.

Sample Applications and Utilities

Along with the driver, the FPGA configuration files, and the software initialization files, the software CD includes a number of applications and utilities that you can use to initialize and configure the board, access registers, or test the board. For many of these applications and utiilities, C source is also provided, so that you can use them as starting points to write your own applications. The most commonly useful are described below; see the README file for the complete list.

NOTE Software is updated regularly; the latest versions are available on our website at www.edt.com/software.html. We encourage you to use the latest versions for new installations. For existing applications, upgrade only if you have a specific reason to do so.

Sample Applications

`rd16`	Performs simple multichannel ring buffer input.
`wr16`	Performs simple multichannel ring buffer output.
`simple_read`	Performs DMA input without using ring buffers. Data is therefore subject to interruptions, depending on system performance.
`simple_write`	Performs DMA output without using ring buffers. Data is therefore subject to interruptions, depending on system performance.
`simple_getdata`	Serves as an example of a variety of DMA-related operations, including reading the data from the connector interface and writing it to a file, as well as measuring input rate.
`simple_putdata`	Serves as an example of a variety of DMA-related operations, including reading data from a file and writing it out to the connector interface.
`test_timeout`	Under normal operation, timeouts cancel DMA transfers. This application exemplifies giving notification when a timeout occurs, without canceling DMA
`set_ss_vco`	A utility for programming the output clock or clocks on the ECL to specific frequencies used by the UI FPGA for input and output.

Utility Files

`initpcd`	A utility for initializing and configuring the ECL.
`pdb`	Utility application that enables interactive reading and writing of the PCI SS/GS UI FPGA registers.

Testing Files

A variety of files — C source, executables, and FPGA configuration files — are available to test the boards. Their uses are described in the documents listed under the heading Testing Procedures. They include at least:

`sslooptest`	Tests most PCI SS- and PCI GS-based boards. Determines the board model and selects the loopback test to run, then runs it.
`xtest`	Tests the PCI CD and CDa boards, and the single-channel DMA interface for the PCI SS and PCI GS main boards.

Building Applications

Executable and PCD source files are at the top level of the EDT PCD driver distribution directory. If you need to rebuild an application, therefore, run `make` in this directory.

Windows and Solaris users must install a C compiler. For Windows, we recommend the Microsoft Visual C compiler; for Solaris, the Sun WorkShop C compiler. Linux users can use the `gcc` compiler typically included with your Linux installation. If Solaris or Windows users wish to use `gcc`, contact `tech@edt.com`.

After you've built an application, use the `--help` command line option for a list of usage options and descriptions.

Configuring the ECL

For the ECL to operate as you require, it must be loaded with the appropriate FPGA configuration files for both FPGAs. The PCI FPGA is loaded from flash ROM, which is shipped from the factory already loaded with the appropriate FPGA configuration file; however, you must load the UI FPGA yourself.

Before loading the UI FPGA, however, you may wish to check the firmware in the PCI FPGA to ensure that it is correct and up-to-date.

Checking the PCI FPGA Firmware

When upgrading to a new device driver, or switching to a FPGA configuration file with special functionality, you may also need to reprogram the PCI interface flash PROM using `pciload`.

The following procedure applies to standard firmware only. If you are running a custom firmware file and need to update it, first get a custom firmware configuation file from EDT.

NOTE The presence of a newer version of the firmware with a new driver doesn't necessarily mean that the firmware must be updated; if a package contains a mandatory upgrade, it is prominently stated in the README file.

On UNIX systems, `pciload` is an application in the installation directory `/opt/EDTpcd`.

On Windows systems, double-click the Pcd Utilities icon to bring up a command shell in the installation directory `\EDT\Pcd`.

On Macintosh systems, `pciload` is an application in the installation directory `/Applications/EDT/pcd`.

To see currently installed and recognized EDT boards and drivers, enter:

```
pciload
```

The program outputs the date and revision number of the firmware in the PROM.

To compare the PCI FPGA firmware in the package with the one already loaded on the board, enter:

```
pciload verify
```

The program compares the firmware in the PROM against the firmware file in the installation directory. If they match, there's no need to upgrade the firmware. If they differ, you'll see error messages. This does not necessarily indicate a problem; if your application is operating correctly, you may not need to upgrade the firmware.

If you wish to update the standard firmware, enter:

```
pciload update
```

1. To upgrade or switch to a custom firmware file, enter:

   ```
   pciload firmware_filename
   ```

   replacing *firmware_filename* with the name of the PCI FPGA configuration file, with or without the `.bit` file extension.

NOTE If the host computer holds more than one board, you can specify the correct board to load with the optional *unit_number* argument (by default, 0 for the first or only board in a host):

```
pciload -u unit_number filename
```

2. At the prompt, press **Enter** to confirm the loading operation. (If the file date is older than the PROM ID date, you may need to press **Enter** twice.)

The board reloads the firmware from the PROM only during power-up, so after running `pciload`, the old firmware remains in the PCI FPGA until the system has power-cycled.

NOTE Updating the firmware requires cycling power, not simply rebooting.

For a list of all `pciload` options, enter:

```
pciload --help
```

Loading the UI FPGA Firmware and Configuring the ECL

The utility `initpcd` loads the UI FPGA configuration files, programs the registers, sets the clocks (if necessary), and gets the ECL mezzanine board ready to perform DMA. This utility takes, as an argument, a software initialization file, and then automatically runs the pertinent commands.

If you use `initpcd` to configure the ECL, your application can concern itself solely with performing DMA and other application-specific operations; it will therefore omit ECL-specific operations and be portable to other EDT boards that peform DMA.

To configure the ECL, enter:

```
initpcd -u unit_number -f pcd_config/filename.cfg
```

replacing `unit_number` with the number of the board (by default, 0), and replacing *filename* with one of the initialization files listed in About the Software and Firmware; for example:

```
initpcd -f two_bitx4ch.cfg
```

NOTE Software initialization files are editable text files. If the files provided don't meet your needs, copy and modify the one that's closest to your required configuration, then run `initpcd` with your new file.

Using Custom FPGA Configuration Files

You can substitute your own FPGA configuration file, if necessary. If you wish to develop your own VHDL design, contact EDT. When you're done, be sure to create a new software initialization file for your new firmware file and update the `pcd_config` directory to include it.

NOTE The example application `extdbdid`, or the EDT DMA library calls `edt_get_board_description` and `edt_print_board_description`, can also determine which firmware is loaded into the PCI and UI FPGAs. These procedures, or the example application, also tell you which mezzanine boards are installed in a system, their revision numbers, and other useful information.

Revision Log

Below is a history of modifications to this guide.

Date	By	Rev.	Pp	Detail
20130307	PH	02	1, 11	• Updated product photo on first page (title page). • Added Revision Log on last page (this page).
20130307	PH	02	All	• Updated formatting and copyright and warranty information. • Repaginated to use continuous arabic numerals from title page to end.
20070500	LW	01	All	• Created this guide for ECL product.