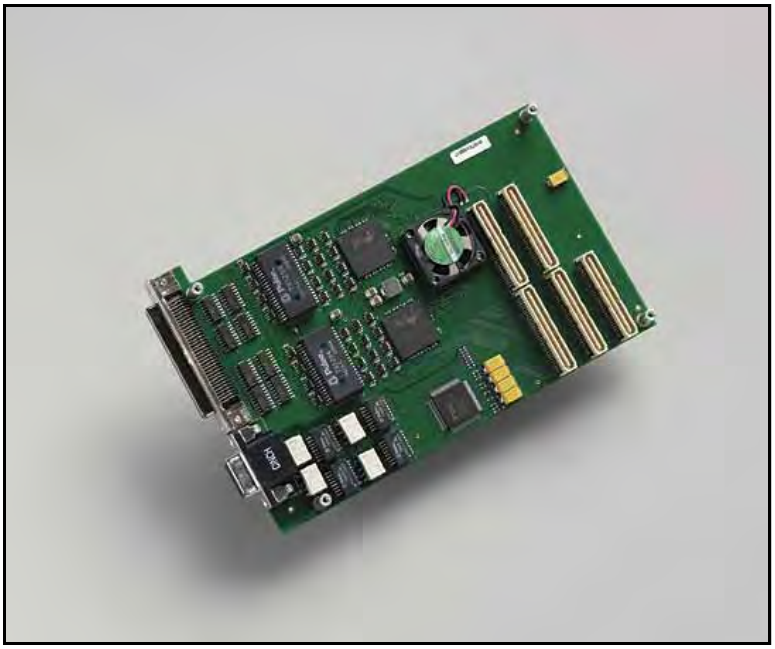
User's Guide

Combo, Combo 2, or Combo 3

Mezzanine Board



Interface for multiple signal types for use with an EDT main board

Rev. 2010 April 19

Parhelia B.V.
www.parhelia-bv.com
+31(0)10 741 00 28

Engineering Design Team (EDT), Inc.
1400 NW Compton Drive, Suite 315
Beaverton, OR 97006
p 503-690-1234 / 800-435-4320
f 503-690-1243
www.edt.com

Terms of Use Agreement

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., 20230, 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

Combo, Combo 2, or Combo 3

Overview

The Combo family consists of three mezzanine boards: Combo, Combo 2, or Combo 3. Each can be combined with an EDT main board for DMA and other resources; an EDT Time Distribution auxiliary board for time code input; or both. For user's guides to main and auxiliary boards, see Related Resources below.

Each Combo family mezzanine board has three types of physical interfaces:

- sixteen E1 or T1 (DS1) interfaces – referred to as Level 1 signals in this guide;
- four E3 or T3 (DS3) interfaces – referred to as Level 3 signals in this guide; and
- eight general-purpose differential input / output (I/O) interfaces.

E1 and E3 are European telecommunication standards; T1 and T3 (also called DS1 and DS3) are the similar North American equivalents. To find telecommunications standards, see Related Resources below.

NOTE Unlike the telecommunications system, a Combo family board does not provide duplex capability on each of its interfaces. If duplex capability is required, only eight Level 1 interfaces and two Level 3 interfaces are usable due to connector limitations.

In the registers in the configured FPGA on the board, the 28 serial interfaces are mapped to the 16 DMA channels as either input or output. Unformatted data is received into, or transmitted from, memory buffers maintained in system memory. For more information, see Registers on page 19.

You can also switch between input and output:

- For Combo 3, use the registers to access the FPGA on the main board (see Registers on page 19).
- For Combo or Combo 2, use the jumpers (see Pinouts on page 16).

Related Resources

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

EDT Resources

Resource	*Documentation*	*Web link*
• Combo family specifications (datasheet)	PDF	www.edt.com/mezz_combo.html
• EDT Main Board User's Guide	PDF	www.edt.com/manuals/PCD/main_boards.pdf
• Time Distribution User's Guide	PDF	www.edt.com/manuals/PCD/time-dist.pdf
• Application Programming Interface	HTML and PDF	www.edt.com/api
• Installation packages (software): Windows, Linux, Solaris, Mac	Software download	www.edt.com/software.html

Standards

Description	*Pertains to*	*Documentation*	*Web link*
• E1/T1 through E3/T3	Combo family:		www.itu.int/home
	Waveforms / traits	ITU-T G.703	"
	Framing (E1, E2, E3)	ITU-T G.704, 742, 751	"
	Loss of signal	ITU-T G.775	"
	Jitter tolerance	ITU-T G.823, 824	"
• T1 and T3	Combo family	ANSI T1.102-1933	www.t1.org
• Negative ECL	Combo, Combo 3	Fairchild 100398	www.alldatasheet.net/datasheet-pdf/pdf/49973/FAIRCHILD/100398.html
• LVDS	Combo 2, Combo 3	ANSI TIA / EIA RS644	www.datasheetarchive.com/datasheet-pdf/022/DSA00391558.html
• RS422	Combo 2, Combo 3	ANSI TIA / EIA RS422	www.datasheetarchive.com/datasheet-pdf/022/DSA00391558.html
• Generic Requirements	Combo family	Telcordia GR-499-CORE	www.telcordia.com/services/genericreq/digest/latest.html#_GR-499

Parts

Description	*Part Number*	*Manufacturer*	*Web link*
• PLL oscillators	ICS370-01	IDT, Inc. (formerly Integrated Circuit Systems)	www.globalspec.com/datasheets/3623/DigiKey/DB0A94E8-8006-467B-8125-A14A3FD72A54
• E1 / T1 LIUs (Combo and Combo 2)	LXT384	Intel Corp.	www.datasheetarchive.com/datasheet-pdf/025/DSA00449266.html
• E1 / T1 LIUs (Combo 3)	LXT3108	Intel Corp.	www.datasheetarchive.com/datasheet-pdf/09/DSA00155141.html
• E3 / T3 LIUs (Combo, Combo 2)	DS3150T or	Maxim Integrated Products	www.maxim-ic.com/pst/run.mvp?query=DS3150T
	TDK78P2241	Teridian Semiconductor Corp. (formerly TDK)	www.datasheet.org.uk/78P2241-datasheet.html
• E3 / T3 LIUs (Combo 3)	TDK78P2344JAT	"	www.datasheet.org.uk/78P2344JAT-datasheet.html

Level 1 (E1/T1) and Level 3 (E3/T3) Signals

Level 1 and Level 3 signals are used for telecommunications. A typical telecommunication connection has four wires providing full duplex capability (two wires for a receiving signal and two for a transmit signal).

All Level 1 and Level 3 signals use a return-to-zero code (over time, there's no DC offset to the signal) to transmit data in a transformer-coupled environment. A typical voice telecommunication signal uses time-division multiplexing – a method of putting multiple data streams into one signal by separating the signal into many segments of very short duration. Each individual data stream is reassembled at the receiving end based on an embedded frame signal to carry many voice calls on one of these higher-level signals.

For example, the T3 interface multiplexes 28 T1 interfaces; with each T1 interface multiplexing 24 64-Kbps channels. (The standardized 64 Kbps channel is based on the bandwidth required for a voice conversation.) The four Level 1 wires typically are a pair of twisted copper wires, but can be coaxial cable. Level 3 wires are coaxial cable.

Line Interface Units (LIUs)

Each Combo family board includes line interface units (LIUs) that receive return-to-zero coded signals, extract the clock, and decode the original data, as shown in Table 1. (For part information, see Related Resources on page 2.)

Table 1. Line Interface Units

Signal	Bit Rate	RZ Code	Combo LIU	Combo 2 LIU	Combo 3 LIU
T1	1.544 Mb	B8ZS	Intel LXT384	Intel LXT384	Intel LXT3108
E1	2.048 Mb	HDB3	Intel LXT384	Intel LXT384	Intel LXT3108
E3	34.368 Mb	HDB3	TDK 78P2241 or Dallas DS3150T	TDK 78P2241 or Dallas DS3150T	TDK 78P2344(JAT)
T3	44.736 Mb	B3ZS	TDK 78P2241 or Dallas DS3150T	TDK 78P2241 or Dallas DS3150T	TDK 78P2344(JAT)

NOTE All boards are available with an option of a 75-ohm termination for the E1 signals. When this option is installed, the board will no longer meet T1 termination specifications.

Differential I/O Signals

Combo boards implement sixteen general-purpose differential I/O signals. Each differential signal is carried over two wires. These signals can be inputs or outputs — controlled by the FPGA — in groups of four.

The FPGA configuration files provided by EDT configure these signals as eight synchronous serial interfaces. Each interface uses one differential signal as a clock and another as data.

Table 2 lists the supported differential I/O standards.

Table 2. Supported Differential I/O Standards

Board	Standard	Supporting Document
Combo and Combo 3	Negative ECL	Fairchild 100398 data sheet
Combo 2 and Combo 3 LVDS	LVDS	TIA / EIA RS644
Combo 2 and Combo 3 RS422	RS422	TIA / EIA RS422

Circuit Protection

Combo family boards do not support lightning or power line short protection. The signals provide typical industrial electrostatic discharge protection. Supply external protection when cabled to external building wiring or in situations where shorts to power wires are possible.

Signal Grounding

The twisted pair T1 and E1 inputs and outputs are transformer-isolated, and no DC or AC ground coupling is provided. The coaxial T3, E3 and 75-ohm E1 options provide an AC coupling from the coaxial shield to the logic ground. Some standards suggest a DC connection to logic ground. This is possible on the boards, but generally is not a good idea as ground levels between equipment can vary by several volts.

Installation

To install the Combo family boards on Windows systems:

1. Power up the host, if necessary.
2. Install the Pcd driver software on the host as described on the EDT installation disk jacket.
3. Power down the host.
4. Plug the board into the PCI bus connector according to host computer manufacturer's instructions.
5. Connect the required cables to the 16-pin and 68-pin connectors, and connect the external device(s) to the cables as necessary. If a cable backshell does not permit both cables to be connected at once, you can change the backshell to one that does.

To install the Combo family boards on Unix-based systems:

1. Power down the host, if necessary.
2. Plug the board into the PCI bus connector according to host computer manufacturer's instructions.
3. Connect the required cables to the 16-pin and 68-pin connectors, and connect the external device(s) to the cables as necessary. If a cable backshell does not permit both cables to be connected at once, you can change the backshell to one that does.
4. Power up the host.
5. Install the Pcd driver software on the host as described on the EDT installation disk jacket.

Included Files

This section lists the EDT files that are included with each Combo family mezzanine board.

FPGA Configuration Files

EDT provides the FPGA configuration files below (files ending in `.bit` are FPGA configuration files).

Combo and Combo 2

`combo16io.bit` Enables acquisition of unframed or raw data from 16 of the LIUs or differential interfaces. The 16 differential signals are treated as eight simple clock/data interfaces. Data for each output can be sourced by any of the 16 DMA channels not used for input.

`combo16in.bit` Enables acquisition of unframed or raw data from 16 of the LIUs or differential interfaces. The 16 differential signals are treated as eight simple clock/data interfaces. Fifteen-bit pseudorandom (prbs15) data is generated for each of the outputs. This file is a subset of `combo16io.bit`. It is included for legacy applications and for single-board loopback testing.

`combo_pdh_demuxin.bit`
Enables framing and demultiplexing of E1 / E3 signals into five DMA channels. The differential inputs serve as eight differential clock-data pairs on eight DMA channels.

Combo 3

`c3_16io.bit` Enables acquisition of unframed or raw data from 16 of the LIUs or differential interfaces. The 16 differential signals are treated as eight simple clock/data interfaces. Data for each output can be sourced by any of the 16 DMA channels not used for input.

`c3_16in.bit` Enables acquisition of unframed or raw data from 16 of the LIUs or differential interfaces. The 16 differential signals are treated as eight simple clock/data interfaces. Fifteen-bit pseudorandom (prbs15) data is generated for each of the outputs. This file is a subset of `c3_16io.bit`. It is included for legacy applications and for single-board loopback testing.

`combo3_pdh_demuxin.bit`
Enables framing and demultiplexing of E3 and E1 signals into five DMA channels. The differential inputs serve as eight differential clock-data pairs on eight DMA channels.

Example and Utility Programs

EDT provides the example and utility programs below.

General

`ss_time_lib.c` EDT Time functions to use the onboard clock.

`ss_time_lib.h` Header file for the EDT Time functions.

`edt_ss_time.c` Example program to show use of the EDT Time functions.

`set_ss_vco.c` Program that sets the clock frequency of the specified PLL on the PCI SS, PCI GS, or PCIe8 LX / FX, which the Combo family of mezzanine boards can use as an output frequency.

`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.

`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.

Combo and Combo 2

`demux`	Example script to show use of demultiplexing. It loads the appropriate configuration file and sets up the registers for E1 and E3 inputs. You can then use `simple_getdata` with the appropriate channel number to get the resulting data.

Combo 3

`c3_demux`	Example script to show use of demultiplexing. It loads the appropriate configuration file and sets up the registers for E1 and E3 inputs. You can then use `simple_getdata` with the appropriate channel number to get the resulting data.
`c3_set_e1.c`	Configures the Level 1 LIUs for E1 operation.
`c3_set_e3.c`	Configures the Level 3 LIU for E3 operation.

Initialization Files

EDT provides the initialization files below (for details, see Automatic Software Configuration on page 11).

Combo and Combo 2

`combo_t1.cfg`	Configures the Combo and Combo 2 boards for T1 operation.
`combo_t3.cfg`	Configures the Combo and Combo 2 boards for T3 operation.
`combo_e1.cfg`	Configures the Combo and Combo 2 boards for E1 operation.
`combo_e3.cfg`	Configures the Combo and Combo 2 boards for E3 operation.
`combo_diff.cfg`	Configures the Combo and Combo 2 boards for differential operation.

Combo 3

`combo3_t1.cfg`	Configures the Combo 3 board for T1 operation.
`combo3_t3.cfg`	Configures the Combo 3 board for T3 operation.
`combo3_e1.cfg`	Configures the Combo 3 board for E1 operation.
`combo3_e3.cfg`	Configures the Combo 3 board for E3 operation.
`combo3_diff.cfg`	Configures the Combo 3 board for differential operation.

Test Files

EDT provides the test files listed below (for details, see Testing on page 11).

General

`sslooptest`	The application that performs the loopback test.
`sslooptest.c`	C source for the loopback test application.
`pciss1test.bit`	FPGA configuration file to support bandwidth test for a PCI FPGA loaded with `pciss1.bit` or `pciss4.bit`.
`pciss16test.bit`	FPGA configuration file to support bandwidth test for a PCI FPGA loaded with `pciss16.bit`.
`check_gap_ss.c`	Program to load the appropriate FPGA configuration file and perform the host bandwidth test.

`genprbs15.c`	Generates DMA data for a 15-bit pseudorandom bit sequence in selected channels, up to sixteen at once.
`chkprbs15.c`	Checks DMA data against a 15-bit pseudorandom bit sequence in selected channels, up to sixteen at once.

Combo and Combo 2

`e3prbs`	Outputs generated E3 test data to cable. (Windows: `.bat`)
`e3chk`	Intputs and tests generated E3 test data to cable. (Windows: `.bat`)
`e3chk.pdb`	Command file to configure registers for E3 test; used by test script.
`e1prbs`	Outputs generated E1 test data to cable. (Windows: `.bat`)
`e1chk`	Intputs and tests generated E1 test data to cable. (Windows: `.bat`)
`e1chk.pdb`	Command file to configure registers for E1 test; used by test script.
`t1prbs`	Outputs generated T1 test data to cable. (Windows: `.bat`.)
`t1chk`	Intputs and tests generated T1 test data to cable. (Windows: `.bat`)
`t1prbs.cfg`	Command file to configure output registers for T1 test; used by test script.
`t1chk.cfg`	Command file to configure input registers for T1 test; used by test script.
`diffprbs`	Outputs generated differential signal test data to cable. (Windows: `.bat`)
`diffchk`	Intputs and tests generated differential signal test data to cable. (Windows: `.bat`)
`diffchk.pdb`	Command file to configure registers for differential signal test; used by test script.

Combo 3

`c3_set_e3.c`	C source for the application that sets up the E3/T3 line interfaces; used by the test scripts.
`c3_e3prbs`	Outputs generated E3 test data to cableI. (Windows: `.bat`)
`c3_e3chk`	Intputs and tests generated E3 test data to cable. (Windows: `.bat`)
`c3_e3prbs.pdb`	Command file to configure registers for E3 test output; used by test script.
`c3_e3chk.pdb`	Command file to configure registers for E3 test input; used by test script.
`c3_set_e1.c`	C source for the application that sets up the E1/T1 line interfaces; used by the test scripts.
`c3_e1prbs`	Outputs generated E1 test data to cable. (Windows: `.bat`)
`c3_e1chk`	Intputs and tests generated E1 test data to cable. (Windows: `.bat`)
`c3_e3prbs.pdb`	Command file to configure registers for E1 test output; used by test script.
`c3_e3chk.pdb`	Command file to configure registers for E1 test input; used by test script.
`c3_diffprbs`	Outputs generated differential signal test data to cable. (Windows: `.bat`)
`c3_diffchk`	Intputs and tests generated differential signal test data to cable. (Windows: `.bat`)
`c3_diffchk.pdb`	Command file to configure registers for differential signal test; used by test script.

Building Applications

Executable files and Pcd source files are at the top level of the EDT PCD distribution directory. Therefore, if you need to rebuild an application, run `make` at the top-level 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. If you wish to use `gcc`, contact `tech@edt.com`.

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

Configuring the Board

On a Combo family mezzanine board, the FPGA configuration file configures how the Level 1, Level 3, and differential I/O interfaces work with each other and with the main board. The included FPGA configuration files are of two types:

- Those used to select, collect, and send raw data only (see Selecting and Collecting Raw Data below); and
- Those that can also synchronize with properly framed E3 signals to collect and send framed or un-framed E1 data (see Collecting Framed E1 Data below).

With either type, configure the FPGA on the main board with the configuration file `pciss16.bit` or `pcigs16.bit`. This file supports sixteen simultaneous DMA channels. Detailed instructions are provided in Loading and Upgrading the Firmware on page 10.

Selecting and Collecting Raw Data

The configuration files `combo16io.bit` and `c3_16io.bit` have a total of 28 possible serial interfaces (sixteen for Level 1, four for Level 3, and eight for differential I/O). Each of these 28 interfaces can be either input or output.

Registers in the configured FPGA map the 28 serial interfaces to the sixteen DMA channels as either an input or an output. Unformatted data is received from, or transmitted to, memory buffers maintained in system memory.

Collecting Framed E1 Data

Your Combo family board includes the demultiplexing configuration files `combo_pdh_demuxin.bit` and `combo3_pdh_demuxin.bit`. These files collect framed or unframed E1 data from the E3 inputs and the E1 inputs, plus raw serial data from the differential inputs. The data from each E3 or all sixteen E1 inputs is presented in a single DMA channel. In this way, all eighty possible E1 inputs (16 from each E3 plus 16 E1s) require only five DMA channels.

A single E3 is a multiplexed collection of sixteen E1 signals: each E3 contains four E2-level signals, and each E2 contains four E1 signals. Though the E2 level does have a physical implementation specification, it usually exists only in this virtual state. Because each E1 signal can have a different timing source, the E3 / E2 structure must accommodate E1 data shifting relative to the E3 timing. (For the ITU standards that describe the E3, E2 and E1 framing, see Related Resources on page 2.)

If framing is disabled (using the register 0x4B Enable E3 Framing), then the board collects raw E3 data. If framing is enabled, then data is collected only when a properly framed E2 signal can be found within the E3 signal. If E2 framing is found, then the E1 data can be collected whether the E1 signal is framed or not. In

the register 0x4E Enable E1 Framed Data, if the E1 frame enable bit for a given E3 signal is set, then only framed E1 signals are collected out of that E3 signal.

A framed E1 signal is 256 bits every 125 μseconds. This 256-bit frame is divided into 32 eight-bit channels, or time slots These channels are referred to as channel 0 – 31. Channel 0 is always used for framing. A fixed sequence of patterns in channel 0 is used to determine the frame boundaries. A pattern in channel 16 is sometimes used for to determine a multiframe containing additional signaling. EDT configuration files frame to the basic pattern; if necessary, you can write software to decode multiframe information from channels 16 and 0.

The example scripts `demux` (for Combo and Combo 2) and `c3_demux` (for Combo 3) show how to configure the board and set up the registers for multiplexing.

DMA channels are assigned as shown in Table 3.

Table 3. DMA Channel Usage

Channel	Data Source
0-7	Unformatted differential inputs 0-7, respectively.
8-11	Unformatted E3 data, unframed E1, or framed E1 data from E3 inputs 0-3, respectively.
12	Unframed or framed E1 data from the sixteen E1 inputs.
13-15	Not used.

Data from 16 E1 signals is placed in a single DMA stream with a timestamp, E1 number, E1 status, and channel data. E1 data in a DMA channel is formatted in ten 32-bit word blocks (see Table 4).

Table 4. E1 Data Format

	Byte 3	Byte 2	Byte 1	Byte 0
Word	**bit 31**			**bit 0**
0	timestamp (in seconds)			
1	timestamp (in fractions of seconds)		frame status	E1 number
2	channel 3	channel 2	channel 1	channel 0
3	channel 7	channel 6	channel 5	channel 4
4	channel 11	channel 10	channel 9	channel 8
5	channel 15	channel 14	channel 13	channel 12
6	channel 19	channel 18	channel 17	channel 16
7	channel 23	channel 22	channel 21	channel 20
8	channel 27	channel 26	channel 25	channel 24
9	channel 31	channel 30	channel 29	channel 28

The format of the timestamp is described in EDT Time on page 13.

Table 5 describes the frame status and E1 number bits.

Table 5. Frame Status and E1 Number Bits

Bit	Name	Description
11	E1_FRAMED	Set when the E1 signal frame alignment is valid.
10	E1_ALIGNED	Set when a frame pattern has been detected for potential byte alignment.

Table 5. Frame Status and E1 Number Bits

Bit	Name	Description
9–5		Reserved.
4	E1_ODDFRM	Set when the following E1 frame is the odd frame of a frame pair. Cleared when the frame is the even frame
3–0	E1_NUMBER	The number 0 – 15 of the E1 signal with which this frame is associated. For an E3 channel, the top two bits are the E2 tributary number and the bottom bits are the E1 tributary within the E2.

Loading and Upgrading the Firmware

As stated earlier in this section, the FPGA configuration file configures how the Level 1, Level 3, and differential I/O interfaces work with each other and with the FPGA on the main board. For a Combo family board to operate correctly, the FPGA PROM on the main board must be loaded with either `pciss16.bit` or `pcigs16.bit`, depending on which main board you are using.

To verify this and, if necessary, load the configuration file on the main board:

1. Navigate to the directory in which you installed the driver (by default, for Unix-based systems, `/opt/EDTpcd`; for Windows-based systems, `\EDT\pcd`).
2. Run `pciload` without arguments to determine the unit number of the main board — by default, 0 — as well as to verify that the host detects your main board, and that it's loaded with 16-channel firmware.
3. If the PROM ID includes the string `pcigs16` or `pciss16`, then 16-channel firmware is already loaded. To verify that the firmware is current, at the prompt, enter:

   ```
   pciload verify
   ```
4. If more than one board is installed on a system, specify the unit number following the -u option:

   ```
   pciload -u unit number verify
   ```

   This compares the date and revision numbers of the PROM configuration file with the date and revision number of the PCI FPGA configuration file on the main board. If these match, there is no need to upgrade the firmware. If they differ, and if you've also been experiencing difficulties, upgrade the firmware as described in Loading and Upgrading the Firmware on page 10.
5. If 16-channel firmware is not already loaded, load it with the appropriate command – either...

   ```
   pciload -u unit number pciss16
   ```
   ...or...
   ```
   pciload -u unit number pcigs16
   ```
6. At the prompt, press Enter to confirm the loading operation.
7. Shut down the operating system and cycle Unix power on the host computer, as the Combo family re-loads firmware from flash PROM only during power-up.

If the results of `pciload verify` indicate that it is necessary (as specified on step 3 on page 10), upgrade the firmware as follows:

1. At the prompt, enter:

   pciload update
2. Shut down the operating system and cycle power on the host computer, as the Combo family reloads firmware from flash PROM only during power-up.
3. If necessary, load the selected Combo family configuration file...

   `bitload -u` *unit number* `filename.bit`

...substituting the appropriate configuration file name for *filename*. (See Included Files on page 4 for a list of the provided configuration files and their descriptions.)

For other `pciload` options, run:

```
pciload -h
```

Automatic Software Configuration

The utility `initpcd` loads the initialization files, programs the registers, sets the clocks (if necessary), and gets the Combo family mezzanine board ready to perform DMA. This utility takes, as an argument, an initialization file, and then automatically runs the pertinent command (or commands) of those discussed in Loading and Upgrading the Firmware on page 10.

If you use `initpcd` to initialize the Combo family, your application can concern itself solely with application-specific operations; it will therefore omit Combo family-specific operations and be portable to software written for other EDT boards that peform DMA.

To initialize the Combo family, enter...

`initpcd -f filename`

...replacing *filename* with the appropriate initialization file for your board (see Initialization Files on page 6).

For example:

`initpcd -f combo_t3.cfg`

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

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.

NOTE The main board's PCI FPGA, when shipped, is configured with either `pciss16.bit` or `pcigs16.bit`. However, custom applications may require a different configuration; contact EDT for details.

Testing

Your Combo family board can be tested via two methods:

- The internal loopback method tests input and output on the same board by generating data within the FPGA and then looping inputs to outputs.
- The board-to-board method uses two boards – one as output and one as input to test the connector pins as well and, with generated test data, the entire circuit.

The files used for these tests are described in the above sections entitled Test Files on page 6. Test data is generated in prbs15 format (contact us for prbs7 or prbs23 format).

NOTE If you wish to conduct external loopback or board-to-board tests, contact EDT for information on constructing the loopback connecter and test cable.

Internal Loopback

The loopback test determines the board configuration, loads the appropriate FPGA configuration file, generates test data, and tests the board and its components with no loopback connector or external device connected. To perform this test:

1. Leave the board in the host computer with the mezzanine board (if any) attached, but disconnect any external device and its cable.
2. In a command window, enter...

   `sslooptest -u` *unit_number*

   The test outcome varies depending on the main board and mezzanine board installed. Errors are redirected to the file `sslooptest.err` in the current directory; if no such file exists, the test completed without errors.

   Loopback test output for a functional board contains such lines as:

   `Total errs=0 bufs=4000; Channel errs(NNNNxxxxxxxxxxxx) bufs(YYYYxxxxxxxxxxxx)`

   `Total errs` shows the error count so far. `bufs` shows the number of buffers tested. The sixteen characters after `Channel errs` show the absence (`N`) or presence (`Y`) of a data error in a specific channel (0–15); an `x` indicates a channel is not in use.

   Similarly, a `Y` after `Channel... bufs` shows a buffer in use; an `x`, that the corresponding channel is not in use. An `N` indicates that DMA is not occurring in a specific channel.
3. After test completion, reconfigure the board with `initpcd` (or your own application) to disable loopback.

NOTE Since the loopback test overwrites the configuration file in the UI FPGA, you must reconfigure the board before you can use it with your application again.

4. Reconnect the board to the external device.

Board-to-Board

This test is performed one way for Combo and Combo 2, and another way for Combo 3.

CAUTION Testing an ECL board with a board using any other type of differential interface will damage your equipment. Otherwise, you can test one kind of Combo board with another kind, as long as both boards use the same differential interfaces: LVDS with LVDS, RS422 with RS422, ECL with ECL.

For Combo and Combo 2 – See Test Files on page 6. To test between two boards:

1. Set up one board for input and one for output (see Pinouts on page 16).
2. Connect the boards with board-to-board cables (a standard 68-pin SCSI cable for the 68-pin connectors and a straight-through 15-pin cable for the E3 connectors). The boards can reside on different hosts.
3. At a command prompt on the host for the output board, to test E3 signals, enter:

   e3`prbs -u` *unit number*

   Replace *e3* with `e1`, `t1`, or `diff` to test E1, T1, or differential signals, respectively.
4. At a command prompt on the host for the input board, enter:

   e3`chk -u` *unit number*

   Again, replace *e3* with `e1`, `t1`, or `diff` to test E1, T1, or differential signals, respectively.

   The output appears as shown in Internal Loopback on page 12.
5. Stop the test when required using the operating system interrupt.
6. If desired, after the test has completed, reconfigure the output board to serve as input and the input board to serve as output and repeat the test.

7. After both tests have completed, reconnect the board to the external device.

For Combo 3 – See Test Files on page 6. Follow the steps above, except:

- Omit the steps that require you to configure the board for input or output with jumpers.
- Prefix the commands in steps 3 and 4 with the characters `c3_`.

EDT Time

Certain EDT FPGA configuration files incorporate time stamps associated with the acquired data. In general, the time can be set from an external time source, or by software synchronization with the host system time. However, the files `combo_pdh_demuxin` and `combo3_pdh_demuxin` are designed to be synchronized with the system time.

The EDT Time functions use Unix time, which counts the seconds from the start of January 1, 1970, with periodic adjustments to match the rotation of the earth (UTC time). EDT Time functions represent time as a 64-bit value in which the most significant 32 bits equal the least significant 32 bits of a 64-bit representation of Unix time. If you need to track the most significant 32 bits of Unix time (which increment once every 136 years), your application must implement this functionality.

The least significant 32 bits of EDT Time represent fractions of a second. A single increment in this number is approximately 233 picoseconds. This level of accuracy is impractical and unnecessary for most applications, so the number of significant bits your application uses can vary as needed. The `combo_pdh_demuxin` and `combo3_pdh_demuxin` configuration files implement twenty bits, for a time stamp accuracy of $1/2^{20}$ of a second, or approximately 954 nanoseconds.

A 32-bit operating system representation of Unix time keeps the second counter in a single signed 32-bit integer. The 32-bit second counter in EDT Time will be compatible with this representation of time until the year 2036, when the 32-bit Unix time will wrap around to 1904.

Adjustments

Because EDT Time is kept on the board, it is based on a crystal oscillator, which is subject to initial accuracy errors as well as temperature and aging errors. To compensate for these errors, the time circuits include an adjustment counter. This counter adds or holds the least significant bit of the fractional second counter on each overflow.

For host system-based time, the EDT software routines make adjustments as the application notes divergence from the system time. (For hardware time code-based systems, adjustment is automatic.) Due to such adjustments, the least significant bit sometimes repeats or jumps by two, so the relative accuracy of the clock is, in the best case, plus or minus one least significant bit (about one μsecond for `combo_pdh_demuxin` and `combo3_pdh_demuxin`). The absolute accuracy of the agreement between the system clock and EDT Time depends on the operating system and your system's response time. Linux can maintain 20 to 50-μsec agreement; Windows is more erratic.

EDT software implements these adjustments gradually to prevent large jumps in time between events, particularly negative jumps where the relative time is important. If the time is significantly wrong, such as at startup, your application can set it directly.

Software Functions

Software functions for EDT Time include:

- setting the board time to system time as Unix time;
- retrieving the 64-bit time value; and
- adjusting for the errors between system time and EDT Time.

The clock on the EDT board can be adjusted to compensate for the drift between board time and system time, as well as adjusted to converge back to the desired system time without time values ever decreasing.

Also, functions are provided to create a monitoring thread that periodically samples the error between EDT Time and system time, then adjusts the board time accordingly.

EDT Time starts automatically as soon as the FPGA configuration file is loaded. For complete function documentation, see the link to the EDT API under Related Resources on page 2.

Table 6 lists some of the most useful functions.

Table 6. EDT Time Software Functions - Abridged List

Purpose	Function
To set the time to current system time:	`edt_sstm_set_to_sys`
To retrieve the current time:	`edt_ss_timestamp`
To get the current error between EDT Time and system time:	`edt_sstm_measure_drift`
To measure the drift between EDT Time and system time:	`edt_sstm_sys_error`
To calculate the current error and revert to system time gradually:	`edt_sstm_iterate_adjust`
To create and start an adjustment thread:	`edt_sstm_launch_adjuster`

It doesn't matter which channel an application opens, as there's only one clock per board. Below is a simple example that sets the board time, then launches an adjustment thread that samples every five minutes.

```
edt_p = edt_open(EDT_INTERFACE, unit);

edt_sstm_set_to_sys(edt_p);

adjuster = edt_sstm_launch_adjuster(edt_p,
      300, // check every 5 minutes
      20, / # of iterations of adjustment as error gets smaller
      10, // each iteration should take 10 secs.
      200,  // maximum 200 microsecond error allowed
      20,  // try to get within 20 microseconds
      0    // loop indefinitely
      );

// for this example just go to sleep
while (1)
      edt_msleep(300000);
```

The sample program provided, `edt_ss_time.c`, implements the above code. To run it, enter:

```
 edt_ss_time -T -L 300 20 200
```

It also exercises the other EDT Time functions.

Programmable Oscillators and set_ss_vco

Each Combo family board uses four programmable oscillators on the main board (PCI SS, PCI GS, or PCIe8 LX / FX). These oscillators, called PLL0 through PLL3 in this guide, allow you to:

- select among three reference frequencies, and
- set up to four different output frequencies.

The reference frequency can be set to:

- 40 MHz (with an accuracy of 100 ppm); or
- 10.3681 MHz (with an accuracy of 25 ppm); or
- the PCI clock frequency.

All EDT-supplied configuration files use a reference frequency of 10.3681 MHz, but your custom configuration file can use one of the two other options, if necessary.

If you use an EDT-supplied configuration file, or have set the reference frequency to 10.3681 MHz (and implemented other features as in EDT-supplied configuration files), you can use the application `set_ss_vco` to set the output frequency for a given PLL. This application sets the internal registers of the specified PLL to achieve the desired frequency, or to get as close as possible, given the limitations of the chip.

To do so, invoke the application with the -F flag, followed by the frequency in Hz, and then followed by the desired PLL number (0–3), thus:

```
set_ss_vco -F frequency PLL_number
```

For example, to set PLL0 to 200 Hz, enter:

```
set_ss_vco -F 200 0
```

The application `set_ss_vco` accepts frequencies in the range of 168 Hz – 100 MHz to produce an even (half low, half high) duty cycle. Frequencies in the range of 100 – 160 MHz are possible, but the duty cycle will be uneven.

An optional `-d` command-line flag may get closer to the desired frequency, but sacrifices the even duty cycle. The specific amount of time the cycle is high versus low varies accroding to the requested frequency.

The application `set_ss_vco` can take several other command-line options, including a `-u` followed by the unit number, in the case of a system with multiple PCI SS, PCI GS, or PCIe8 LX / FX boards. For a complete list of command-line options, invoke the program with `-h` (for Help).

Instead of specifying a frequency with the `-F` flag, `set_ss_vco` can accept a number in the range of 0–5 to specify one of six preprogrammed telecommunications frequencies, as shown in Table 7.

Table 7. Preprogrammed set_ss_vco Frequencies

Argument	Telecommunications Standard	Frequency (MHz)
0	T1	1.544
1	E1	2.048
2	T2	6.312
3	E2	8.448
4	E3	34.368
5	T3	44.736

Thus, to set PLL2 to the E3 frequency, enter:

```
set_ss_vco 5 2
```

By default, EDT-supplied configuration files wire the PLLs to the LIUs as shown in Table 8.

Table 8. Default PLL-LIU Connctions

PLL	FPGA Configuration File	LIU

0 and 3	`combo16io.bit`
	`c3_16io.bit`
	`combo16in.bit`
	`c3_16in.bit`
		Level 1 LIUs (can be both T1 and E1 at the same time)

0	`combo_pdh_demuxin.bit` `combo3_pdh_demuxin.bit`	Level 1 LIU
3	`combo_pdh_demuxin.bit` `combo3_pdh_demuxin.bit`	Internal clock; not user-settable
1	all	Level 3 LIU (either T3 or E3)
2	all	differential

Pinouts

For Combo and Combo 2 boards, the function of the differential I/O and the LVDS input depends on which configuration file is loaded and whether the telecommunications interfaces are jumpered as input or output.

The differential I/O and the E1 / T1 signal levels depend on the type of board: Combo = ECL; Combo 2 = RS422 = RS422; and Combo 2 LVDS = LVDS. Table 9 shows connections made by EDT configuration files in the 68-pin connector for Level 1 and differential signals.

Table 9. Pinout for 68-pin Connector (Level 1 & Differential Signals) – part 1 of 2

			Combo / Combo 2: E1 / T1 Channel		Combo 3	
+ Pins	– Pins	Signal Type	RX (Input)	TX (Output)	RX & TX	Function
1	35	E1 or T1	0	8	0	
2	36	differential I/O				DATA1
3	37	differential I/O				DATA7
4	38	differential I/O				CLOCK1
5	39	E1 or T1	1	9	1	
6	40	differential I/O				DATA2
7	41	differential I/O				CLOCK7
8	42	differential I/O				CLOCK2
9	43	E1 or T1	3	11	3	
10	44	differential I/O				DATA3
11	45	differential I/O				DATA8
12	46	E1 or T1	9	1	9	
13	47	differential I/O				CLOCK3
14	48	E1 or T1	5	13	5	
15	49	differential I/O				DATA4
16	50	differential I/O				CLOCK8
17	51	E1 or T1	12	4	12	
18	52	E1 or T1	4	12	4	
19	53	differential I/O				CLOCK4

Table 9. Pinout for 68-pin Connector (Level 1 & Differential Signals) – part 2 of 2

			Combo / Combo 2: E1 / T1 Channel		Combo 3	
+ Pins	– Pins	Signal Type	RX (Input)	TX (Output)	RX & TX	Function
20	54	E1 or T1	7	15	7	
21	55	LVDS differential input				none
22	56	differential I/O				DATA5
23	57	E1 or T1	2	10	2	
24	58	E1 or T1	11	3	11	
25	59	differential I/O				CLOCK6
26	60	E1 or T1	10	2	10	
27	61	E1 or T1	8	0	8	
28	62	differential I/O				CLK5
29	63	differential I/O				DATA6
30	64	E1 or T1	13	5	13	
31	65	E1 or T1	6	14	6	
32	66	E1 or T1	14	6	14	
33	67	E1 or T1	15	7	15	
34	68	ground				ground

Table 10 shows connections made by EDT configuration files in the 15-pin connector for Level 3 signals.

Table 10. Connector Pinout for 15-pin Connector (Level 3 Signals)

				Combo / Combo 2: E3 or T3 Channel		Combo 3
+ Pins	– Pins	Signal Type	Line (back of board)	RX (Input)	TX (Output)	RX & TX
1	2	E3 or T3	A	0	2	0
3	4	E3 or T3	B	1	3	1
5	6	E3 or T3	C	2	0	2
7	8	E3 or T3	D	3	1	3

Pins 9, 11, 13, and 15 are ground; pins 10, 12, and 14 are not used.

Selecting Input or Output

For Combo 3, you change interfaces from input or output by using the registers to access the FPGA on the main board (see Registers on page 19). For Combo and Combo 2, however, you do so by using jumpers.

Each of the E3/T3 and E1/T1 (20 total) interfaces on a Combo or Combo 2 board has an input (receive, RX) and an output (transmit, TX). Each signal requires two wires; these two wires create a signal pair. The connectors have only enough pins for half of these wires, so each channel must be selected as transmit or receive by setting a pair of jumpers.

To determine which jumpers to set for a given channel:

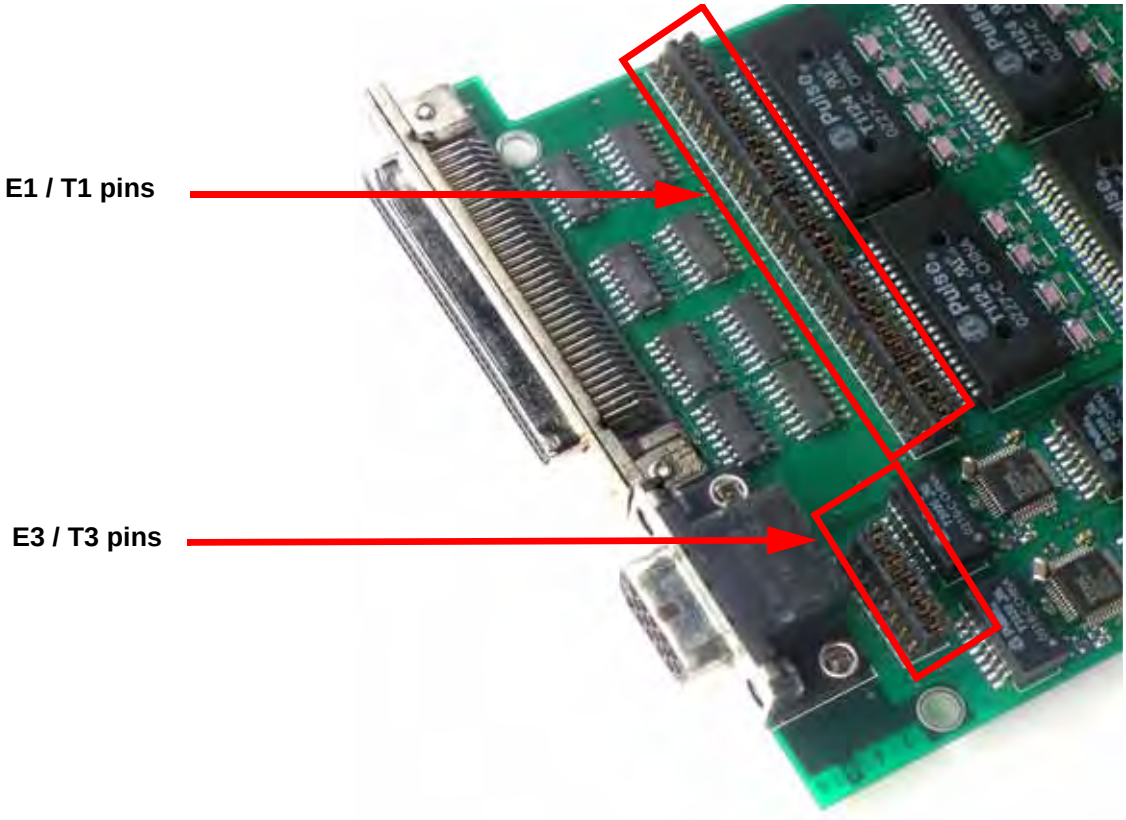
1. Find the line number (for E1/T1) or letter (for E3/T3) to the left of the jumper block that matches the input (RX) column in Table 9 above. The legend to the right of the jumper block indicates which receive / transmit pair is connected to the line pair.
2. Holding the board so that the connectors are on the left, as in Figure 1:
   - — Set the two jumpers left to middle for receive.
   - — Set the two jumpers middle to right for transmit

Figure 1 shows the back of a Combo or Combo 2 board, with pin locations and legends called out.

Figure 1. Combo and Combo 2 – Pin Locations and Legends

Figure 2 shows the front of a Combo or Combo 2 board, with all channels set to transmit.

Figure 2. Combo and Combo 2 – Channels Set to TX



Registers

This section contains a register summary, followed by the registers, for the Combo family boards.

NOTE For Combo and Combo 2, 0x14-15 Channel Edge is implemented but not used, and these additional registers are implemented but not used when using the `combo16in.bit` configuration file: Function; Data Path; Status; Status Polarity; Direction Control; and Channel Direction (automatically detected from the direction of the DMA by the logic in `combo16io.bit` or `c3_16io.bit`)

Table 11 lists all the registers, and shows which boards and which configuration files use each one.

Table 11. Register Summary – part 1 of 2

Register	Used by board...	Used by configuration file...
0x00 Command	Combo family	all
0x05 Site ID	Combo family	all
0x0F Configuration 1	Combo family	all
0x10-11 Channel Enable	Combo family	all; only bits 0–12 are relevant for `combo_pdh_demuxin.bit` and `combo3_pdh_demuxin.bit`
0x14-15 Channel Edge	Combo family	`combo16io.bit` `c3_16io.bit`
0x16-17 Least Significant Bit First	Combo family	all
0x18-19 Underflow	Combo family	`combo16io.bit` `c3_16io.bit`
0x1A-1B Overflow	Combo family	`combo16io.bit` `c3_16io.bit`
0x20 PLL Programming	Combo family	all, but ignored in `combo_pdh_demuxin.bit` and `combo3_pdh_demuxin.bit`
0x21 Output Clock Select	Combo family	`combo16io.bit` `c3_16io.bit`
0x22 Differential Direction	Combo family	all, but ignored in `combo_pdh_demuxin.bit` and `combo3_pdh_demuxin.bit`
0x23 Differential Channel Edge	Combo family	all
0x24-25 PLL 0 Divider	Combo family	all
0x26-27 PLL 1 Divider	Combo family	all
0x28-29 PLL 2 Divider	Combo family	all
0x2A-2B PLL 3 Divider	Combo family	all
0x2E E3 / T3 Control 0	Combo, Combo 2	`combo16io.bit`
0x2F E3 / T3 Control 1	Combo, Combo 2	`combo16io.bit`
0x3A Enable E1 / T1 Data 7–0	Combo family	`combo16io.bit` `c3_16io.bit`
0x48 E3 / T3 Status	Combo family	`combo16io.bit` `c3_16io.bit`
0x48 E3 / T3 Status	Combo 3	`c3_16io.bit`

Table 11. Register Summary – part 2 of 2

Register	Used by board...	Used by configuration file...
0x48 E3 / T3 Status	Combo family	combo16io.bit c3_16io.bit
0x49 E3 / T3 Control 2	Combo, Combo 2	combo16io.bit
0x4A Enable E3 / T3 Data and Clock	Combo family	combo16io.bit c3_16io.bit
0x4B Enable E3 Framing	Combo family	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x4C Combo 3 E3 / T3 Control	Combo 3	c3_16io.bit
0x4D Select E2 and E1 Frame Status	Combo family	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x4E Enable E1 Framed Data	Combo family	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x4F E3 Frame Status	Combo family	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x50-53 E2 Frame Status	Combo family	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x54-57 E1 Frame Status	Combo family	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x60-6F Data Source Select	Combo family	combo16io.bit c3_16io.bit
0x7C-7D FPGA Design ID	–	combo16io.bit c3_16io.bit combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x7E FPGA Revision ID	Combo family	combo16io.bit c3_16io.bit
0x7F Board ID	Combo family	all combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x88 Set Time	–	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x8C Time Adjust	–	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0x8F Time Control	–	combo_pdh_demuxin.bit combo3_pdh_demuxin.bit
0xC0-C3 Intel LXT3108 E1 / T1 LIU Control (Combo 3)	Combo 3	c3_16io.bit combo3_pdh_demuxin.bit c3_16in.bit
0xC0-DF Intel LXT384 E1 / T1 Channels 0–7 Control (Combo and Combo 2)	Combo, Combo 2	combo16io.bit combo_pdh_demuxin.bit combo16in.bit
0xE0-0xFF Intel LXT384 E1 / T1 Channels 8–15 Control (Combo and Combo 2)	Combo, Combo 2	combo16io.bit combo_pdh_demuxin.bit combo16in.bit

0x00 Command

Access / Notes: PCD_CMD / 8-bit read-write

Bit	Name	Description
7-4	[no name]	Not used.
3	CMD_EN	Set this bit, and enable the required channels in 0x10-11 Channel Enable, for DMA to occur. When clear, resets all channels, flushes the FIFOs, and clears all under- and overflow bit.
2-0	[no name]	Not used.

0x05 Site ID

Access / Notes: EDT_SITEID / 8-bit read-only

Bit	Name	Description
7-0	[no name]	Read a fixed value. EDT Site ID is 0xFF; your configuration files can return different values.

0x0F Configuration 1

Access / Notes: PCD_CONFIG / 8-bit read-write

This register and 0x16-17 Least Significant Bit First can affect the order of data transfer.

Bit	Name	Description
7-4	[no name]	Not used.
3	SSWAP	Swaps the order of the two 16-bit short words in one 32-bit data word, so that *short 2* is transferred before *short 1*; does not change the order of the bits within each short. See Figure 3.
2-1	[no name]	Not used.
0	BSWAP	Swaps the order of bytes 1 and 2, and also bytes 3 and 4, in a 32-bit data word, so that the bytes are transferred in the order 2, 1, 4, 3; does not change the order of the bits within each byte. See Figure 3.

Figure 3 shows the structure of a 32-bit data word.

Figure 3. Data Word Structure

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

0x10-11 Channel Enable

Access / Notes: SSD16_CHEN / 16-bit read-write

Bit	Name	Description
15-0	CH_ENABLE[15-0]	Set to 1 to enable the corresponding DMA channel as either input or output. The `combo_e1_demux.bit` and `combo3_e1_demux.bit` files implement only bits 12–0.

0x14-15 Channel Edge

Access / Notes: SSD16_CHEDGE / 16-bit read-only

Bit	Name	Description
15-0	CH_EDGE[15-0]	Set to 1 to set the corresponding DMA channel to enable receiving on the rising edge of the clock signal.

0x16-17 Least Significant Bit First

Access / Notes: SSD16_LSB / 16-bit read-write

This register and 0x0F Configuration 1 can affect the order of data transfer.

Bit	Name	Description
15-0	LSB_FIRST	When set for a channel, the least significant bit of the 32-bit data word is the first bit, and the most significant bit is the last. When clear for a channel, the most significant bit of a 32-bit word is the first bit. When 32 bits are received, the word is written to host memory.

0x18-19 Underflow

Access / Notes: SSD16_UNDER / 16-bit read-only

Bit	Name	Description
15-0	UNDERFLOW	A value of 1 in a bit indicates that the corresponding channel's internal FIFO has underflowed since the previous CMD_EN or CHANNEL_ENABLE. Reset by first disabling, then re-enabling, the channel (see 0x10-11 Channel Enable).

0x1A-1B Overflow

Access / Notes: SSD16_OVER / 16-bit read-only

Bit	Name	Description
15-0	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 first disabling, then re-enabling, the channel (see 0x10-11 Channel Enable).

0x20 PLL Programming

Access / Notes: EDT_SS_PLL_CTRL / 8-bit read-write

Used by `set_ss_vco` to program the serial interface of the four phase-locked loops (PLLs).

Bit	Name	Description
7	PLL_SCLK	Connected to all four PLL serial clock inputs.
6	PLL_DATA	Connected to all four PLL serial data inputs.
5-4	[no name]	Not used.
3-0	PLL_STROBE	Connected to the strobe inputs of PLLs 3 to 0, respectively.

0x21 Output Clock Select

Access / Notes: EDT_SS_PLL_SEL / 8-bit read-write

Select the source for the transmit clock used by each interface. The external clock is an LVDS input from pins 21 and 55 of the 68-pin connector.

Bit	Name	Description
7-3	DIFF_CLK	These bits select the transmit clock source for all eight differential clock outputs, as below.

Bit 7	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Clock Source*
0	0	0	0	0	PLL 1 (default)
0	0	0	0	1	E1/T1 receive clock 0
0	0	0	1	0	E1/T1 receive clock 1
0	0	0	1	1	E1/T1 receive clock 2
0	0	1	0	0	E1/T1 receive clock 3
0	0	1	0	1	E1/T1 receive clock 4
0	0	1	1	0	E1/T1 receive clock 5
0	0	1	1	1	E1/T1 receive clock 6
0	1	0	0	0	E1/T1 receive clock 7
0	1	0	0	1	E1/T1 receive clock 8
0	1	0	1	0	E1/T1 receive clock 9
0	1	0	1	1	E1/T1 receive clock 10
0	1	1	0	0	E1/T1 receive clock 11
0	1	1	0	1	E1/T1 receive clock 12
0	1	1	1	0	E1/T1 receive clock 13
0	1	1	1	1	E1/T1 receive clock 14
1	0	0	0	0	E1/T1 receive clock 15
1	0	0	0	1	E3/T3 receive clock 0
1	0	0	1	0	E3/T3 receive clock 1
1	0	0	1	1	E3/T3 receive clock 2
1	0	1	0	0	E3/T3 receive clock 3
1	0	1	0	1	external clock

2	DIFF_CLK	Select the source for the transmit clock on E3/T3 interfaces 0–3. The default, 0, selects the source as PLL 2; 1 selects the external clock.
1	T3E3_CLK	Select the source for the transmit clock on E1/T1 interfaces 8–15. The default, 0, selects the source as PLL 3; 1 selects the external clock.
0	T1E1_1_CLK	Select the source for the transmit clock on E1/T1 interfaces 0–7. The default, 0, selects the source as PLL 0; 1 selects the external clock.

0x22 Differential Direction

Access / Notes: EDT_SS_DIFFLOOP / 8-bit read-write

Bit	Name	Description
7-4	[no name]	Not used.
3	CH7_6_OUT	Enables the differential drivers for channels 6 and 7.
2	CH5_4_OUT	Enables the differential drivers for channels 4 and 5.
1	CH3_2_OUT	Enables the differential drivers for channels 3 and 2.
0	CH1_0_OUT	Enables the differential drivers for channels 0 and 1.

0x23 Differential Channel Edge

Access / Notes: COMBO_CHEDGE / 16-bit read-write

Set by `set_ss_vco`.

Bit	Name	Description
7-0	EDGE[7-0]	Set corresponding DMA channel to enable receiving on rising clock edge.

0x24-25 PLL 0 Divider

Access / Notes: EDT_SS_PLL0_DIV / 16-bit read-write

Set by `set_ss_vco`. PLL0 sets the clock for the E1/T1 channel 15–8 LIUs.

Bit	Name	Description
15-0	PLL0DIV	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 (`set_ss_vco` takes this into account).

0x26-27 PLL 1 Divider

Access / Notes: EDT_SS_PLL1_DIV / 16-bit read-write

Set by `set_ss_vco`. PLL1 sets the clock for the differential channels.

Bit	Name	Description
15-0	PLL1DIV	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 (`set_ss_vco` takes this into account).

0x28-29 PLL 2 Divider

Access / Notes: EDT_SS_PLL2_DIV / 16-bit read-write

Set by `set_ss_vco`. PLL2 is set for Level 3 (E3 or T3) signal operation.

Bit	Name	Description
15-0	PLL2DIV	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 (`set_ss_vco` takes this into account).

0x2A-2B PLL 3 Divider

Access / Notes: EDT_SS_PLL3_DIV / 16-bit read-write

Set by `set_ss_vco`. PLL3 sets the clock for the E1/T1 channel 7-0 LIUs.

Bit	Name	Description
15-0	PLL3DIV	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 (`set_ss_vco` takes this into account).

0x2E E3 / T3 Control 0

Access / Notes: COMBO_E3T3_CTL / 8-bit read-write

Controls the E3/T3 LIUs used on the Combo and Combo 2 boards. For part information, see Related Resources on page 2.

Bit	Name	Description
7	CHAN1_LBO	Line build out in T3 only. Set for cables less than or equal to 225 feet long; clear for cables more than 225 feet long.
6	CHAN1_RLPBK	Sets loopback receive data to transmit.
5	CHAN1_LPBKA	Set to 0, in addition to LPBKB, for zero analog loopback mode on channel 1.
4	CHAN1_LPBKB	Set to 0, in addition to LPBKA, for zero analog loopback mode on channel 1.
3	CHAN0_LBO	Line build out in T3 only. Set for cables less than or equal to 225 feet long; clear for cables more than 225 feet long.
2	CHAN0_RLPBK	Sets loopback receive data to transmit.
1	CHAN0_LPBKA	Set to 0, in addition to LPBKB, for zero analog loopback mode on channel 0.
0	CHAN0_LPBKB	Set to 0, in addition to LPBKA, for zero analog loopback mode on channel 0.

0x2F E3 / T3 Control 1

Access / Notes: COMBO_E3T3_CTL / 8-bit read-write

Controls the E3/T3 LIUs used on the Combo and Combo 2 boards. For part information, see Related Resources on page 2.

Bit	Name	Description
7	CHAN3_LBO	Line build out in T3 only. Set for cables less than or equal to 225 feet long; clear for cables more than 225 feet long.
6	CHAN3_RLPBK	Sets loopback receive data to transmit.
5	CHAN3_LPBKA	Set to 0, in addition to LPBKB, for zero analog loopback mode on channel 3.
4	CHAN3_LPBKB	Set to 0, in addition to LPBKA, for zero analog loopback mode on channel 3.
3	CHAN2_LBO	Line build out in T3 only. Set for cables less than or equal to 225 feet long; clear for cables more than 225 feet long.
2	CHAN2_RLPBK	Sets loopback receive data to transmit.
1	CHAN2_LPBKA	Set to 0, in addition to LPBKB, for zero analog loopback mode on channel 2.
0	CHAN2_LPBKB	Set to 0, in addition to LPBKA, for zero analog loopback mode on channel 2.

0x3A Enable E1 / T1 Data 7–0

Access / Notes: COMBO_E1T1_CHAN_ENA / 8-bit read-write

Bit	Name	Description
7-0	EN_DATA_[7-0]	For Combo and Combo 2 boards. Set to enable the data for the corresponding E1/T1 channel. (Disabling the data can reduce the amount of noise on output.

0x3B Enable E1 / T1 Data 15–8

Access / Notes: COMBO_E1T1_CHAN_ENA / 8-bit read-write

Bit	Name	Description
7-0	EN_DATA_[15-8]	Combo and Combo 2: Set to enable the data for the corresponding E1/T1 channel. (Disabling the data can reduce the amount of noise on output.

0x3C-3D Combo 3 E1 / T1 Control

Access / Notes: COMBO_E1T1_DIR_CTL / 8-bit read-write

Channels 0-7 are controlled by 0x3C, and channels 8-15 by 0x3D.

Combo 3: You can use this register to specify whether an E1/T1 interface external pin will function as input or output. The TDK 7802344JAT E1 / T1 LIU is programmed via a serial interface. Bits in this register are used by the program `c3_set_e3.c` to set the LIU for E3. (Combo and Combo 2 boards use jumpers to set the signal direction.)

Bit	Name	Description
7-0	EN_TX1	Combo 3: Specify the signal direction of the external signal pair of the corresponding channel; 0 (the default) sets the channel to receive; 1 sets the channel to transmit.

0x48 E3 / T3 Status

Access / Notes: COMBO_E3T3_STAT / 8-bit read-only

Bit	Name	Description
7-5	[no name]	Not used.
4	C3_SPO	Combo 3 only. Reads the status of the TDK 78P2344JAT LIU serial programming interface serial output. Program this using 0x4B Enable E3 Framing.
3-0	T3E3_LOS	Reads the status pin on the E3 / T3 LIU. Combo 3: This is the TDK 78P2344JAT interrupt pin. Combo, Combo 2: This is the LOS signal for the LIU. The EDT Combo 3 setup program `c3_set_e3.c` sets the TDK 78P2344JAT interrupt for LOS.

0x49 E3 / T3 Control 2

Access / Notes: COMBO_E3T3_CTL / 8-bit read-only

Combo and Combo 2: Controls the E3/T3 LIUs (for part information, see Related Resources on page 2). These bits globally control all four channels. Bits 0 and 1 can be set for STS-1 operation, but this is not tested. For E3 operation, set PLL2 to 34.368 MHz. For T3 (DS3) operation, set PLL2 to 44.74 MHz. .

Bit	Name	Description
7-3	[no name]	Not used.
2	DISABLE_DEC	Set to disable internal HDB3 (E3) or B3ZS (T3) decoder. Disabling the decoder with EDT-supplied configuration files produces invalid data.
1	DS3	Set for T3 (DS3) operation; clear for E3 operation.
0	NOT_E3	Set for T3 (DS3) operation; clear for E3 operation.

0x4A Enable E3 / T3 Data and Clock

Access / Notes: COMBO_E3T3_CLK_ENA / 8-bit read-write

Bit	Name	Description
7	EN_DATA3	Set to 1 to enable transmit data for T3 or E3 channel 3.
6	EN_CLK3	Set to 1 to enable transmit clock for T3 or E3 channel 3.
5	EN_DATA2	Set to 1 to enable transmit data for T3 or E3 channel 2.
4	EN_CLK2	Set to 1 to enable transmit clock for T3 or E3 channel 2.
3	EN_DATA1	Set to 1 to enable transmit data for T3 or E3 channel 1.
2	EN_CLK1	Set to 1 to enable transmit clock for T3 or E3 channel 1.
1	EN_DATA0	Set to 1 to enable transmit data for T3 or E3 channel 0.
0	EN_CLK0	Set to 1 to enable transmit clock for T3 or E3 channel 0.

0x4B Enable E3 Framing

Access / Notes: COMBO_E3T3_FRAMING / 8-bit read-write

Bit	Name	Description
7-4	[no name]	Reserved.
3-0	EN_E3_FRAME	Set to enable E3 framing on the specified E3 input. Clear to collect unframed data on the specified E3 input.

0x4C Combo 3 E3 / T3 Control

Access / Notes: COMBO_E3T3_DIR_CTL / 8-bit read-write

Combo 3: Using this register, you can specify that an E3 / T3 interface external pin on the 3 board functions as either input or output. The TDK 78P2344JAT E3/T3 LIU is programmed via a serial interface. Bits in this register are used by the `c3_set_e3.c` program to set the LIU for E3 operation. (Combo and Combo 2 boards use jumpers to set the signal direction.)

Bit	Name	Description
7	LIU3_SCS	Controls the LIU programming serial select input.
6	LIU3_SCK	Controls the LIU programming serial clock input.
5	LIU3_SD1	Controls the LIU programming serial data input.
4	LIU3_POR_L	Clear to place the TDK 78P2344JAT power on reset low (reset state); set to enable the LIU.
3-0	EN_TX3	Specify the signal direction of the external signal pair of the corresponding channel: 0 (the default) sets the channel to receive; 1 sets the channel to transmit.

0x4D Select E2 and E1 Frame Status

Access / Notes: COMBO_E2E1_FRM_STAT / 8-bit read-write

Bit	Name	Description
7-6	[no name]	Reserved.
5-4	SELECT_E2	Specify which E2 status from a given E3 input is reflected in 0x50-53 E2 Frame Status.
3-0	SELECT_E1	Specify which E1 status from a given E3 input is reflected in 0x54-57 E1 Frame Status.

0x4E Enable E1 Framed Data

Access / Notes: COMBO_E1_FRM_ENA / 8-bit read-write

- Bit 4 enables framed E1 signals in the channel made up by all the individual E1 inputs.
- Bits 0-3 enable framed E1 signals contained in the respective E3 input.

Bit	Name	Description
7-5		Reserved.
4-0		Set to acquire only framed E1 data on the specified channel. Clear to collect framed and unframed data on all E1 signals of the channel.

0x4F E3 Frame Status

Access / Notes: COMBO_E3_FRM_STAT / 8-bit read-only

- Bit 4 enables framed E1s in the channel made up by all the individual E1 inputs.
- Bits 0–3 enable framed E1 signals contained in the respective E3 input.

Bit	Name	Description
7-4	E3_AI	Set when the corresponding E3 input is framed and the alarm indicator bit is set.
3-0	E3_FRAMED	Set when the corresponding E3 input is receiving a framed E3 signal.

0x50-53 E2 Frame Status

Access / Notes: COMBO_E2_FRM_STAT0 through COMBO_E2_FRM_STAT3 / 8-bit read-only

Addresses 0x50 through 0x53 read E3 channels 0–3, respectively.

Reflects status of the E2 signal specified in bits 4–5 of 0x4D Select E2 and E1 Frame Status.

Bit	Name	Description
7-3	[no name]	Reserved.
2	REMOTE_AI	Set if the remote alarm indicator bit is set for the specified E2 signal; clear if not.
1	ALIGNED	Set if the board has detected a possible byte alignment in the associated E2 signal; clear if not.
0	FRAMED	Set if the associated E2 signal is framed; clear if not

0x54-57 E1 Frame Status

Access / Notes: COMBO_E1_FRM_STAT0 through COMBO_E1_FRM_STAT3 / 8-bit read-only

Addresses 0x54 through 0x57 read E3 channels 0–3, respectively.

Reflects status of the E1 signal specified in bits 0–3 of 0x4D Select E2 and E1 Frame Status.

Bit	Name	Description
7-2	[no name]	Reserved.
1	ALIGNED	Set if the board has detected a possible byte alignment in the associated E1 signal; clear if not.
0	FRAMED	Set if the associated E1 signal is framed; clear if not.

0x60-6F Data Source Select

Access / Notes: COMBO_DMA_MAP0 through COMBO_DMA_MAPF / 8-bit read-write

Used to select the source of data for each of the input DMA channels.

- Register 0x60 selects the source for DMA channel 0, 0x61 for DMA channel 1, and so on.
- Each register holds a five-bit code that selects the data input/output.

Bit	Name	Description
7-5	[no name]	Not used.
4-0	SEL_CODE	Enter the appropriate code: 0x00–0x0F = E1/T1 0–15, respectively. 0x10–0x17 = Differential data 0–7, respectively. 0x18–0x1B = E3/T3 0–3, respectively.

0x7C-7D FPGA Design ID

Access / Notes: EDT_FPGA_DESIGN_ID / 16-bit read-only

Value assigned to an FPGA configuration file by the development site. EDT configuration files read 0xFF from 0x05 Site ID.

Bit	Name	Description
15-0	DESIGN_ID	**File Design ID** `combo16in.bit` not yet implemented `c3_16in.bit` not yet implemented `combo16io.bit` not yet implemented `c3_16io.bit` not yet implemented `combo_pdh_demuxin.bit` 0x0100 `combo3_pdh_demuxin.bit` 0x0100

NOTE: Not all versions of these configuration files have design IDs.

0x7E FPGA Revision ID

Access / Notes: EDT_FPGA_REVISION / 8-bit read-write

Arbitrary string of up to 64 characters that identifies the configuration file, revision, date, and some basic DMA information, such as the number of required channels, and the number actually used.

Bit	Name	Description
15-0	REV_ID	- Write a character ASCII number 0x00 to 0x3F. - Read a character from the Revision ID string.

0x7F Board ID

Access / Notes: EDT_BOARDID / 8-bit

Used to identify EDT mezzanine boards. A value of 0x2 in the lowest four bits indicates an extended board ID, hard-wired into a nonvolatile complex programmable logic device (CPLD). The `extbdid` application seeks the identifier in the board ID register; if it finds a value of 0x2, then it seeks the extended board ID from the CPLD instead.

Bit	RW	Name	Description
7-4	RW	[no name]	Used by `extbdid.exe`.
3-0	RW	BOARD_ID	See EDT Board ID and Extended Board ID table (below).

Table 12. EDT Board ID and Extended Board ID (CPLD)

Board ID Register, Bits 3–0	Extended Board ID	Board Name	Detail
0 0 0 0 0x0	–	RS422	–
0 0 0 1 0x1	–	LVDS	–
0 0 1 0 0x2	–	Reserved	For extended board IDs (below).
– – – – –	0x0A	SRXL	–
– – – – –	0x10	16TE3	–
– – – – –	0x11	OC192	–
– – – – –	0x12	3x3G	–
– – – – –	0x13	MSDV	–
– – – – –	0x14	SRXL2 (rev01 & 02)	Contact EDT to exchange for later revision.
– – – – –	0x15	Net10G	–
– – – – –	0x16	DRX	–
– – – – –	0x17	DDSP	–
– – – – –	0x18	SRXL2 (rev03+)	For the IDM + LBM option.
– – – – –	0x19	SRXL2 (rev03+)	For the IDM + IDM option.
– – – – –	0x1A	SRXL2 (rev03+)	For the IMM + IMM option.
– – – – –	0x1B	SRXL2 (rev03+)	For the IMM + LBM option.
– – – – –	0x1C	SRXL2 (rev03+)	For the IDM + IMM option.
– – – – –	0x1D	DRX16	For the IDX + IDX option.
– – – – –	0x1E	OCM2P7G	–
0 0 1 1 0x3	–	Reserved	–
0 1 0 0 0x4	–	SSE	–
0 1 0 1 0x5	–	HRC	For E4, STS3, STM1 / OC3 I/O.
0 1 1 0 0x6	–	OCM	–
0 1 1 1 0x7	–	Combo 2	For LVDS I/O.
1 0 0 0 0x8	–	ECL/LVDS-E/RS422-E	For ECL, LVDS, RS422, E1/T1 I/O.
1 0 0 1 0x9	–	TLK1501	–
1 0 1 0 0xA	–	Reserved	–
1 0 1 1 0xB	–	Combo 3	For RS422 I/O.
1 1 0 0 0xC	–	Combo 3	For LVDS I/O.
1 1 0 1 0xD	–	Combo 3	For ECL I/O.
1 1 1 0 0xE	–	Combo 2	For RS422 I/O.
1 1 1 1 0xF	–	Combo	For ECL I/O.

0x80 Capture Time

Access / Notes: EDT_TIME_SNAPSHOT / 64-bit read-only

Reads EDT Time as of the last time the latch time bit was set in 0x8F Time Control.

Bit	Name	Description
63–32	SEC_TIME	Elapsed time in seconds.
31–12	FRAC_TIME	Elapsed time in fractions of a second. Each increment is $1/2^{20}$ second.
11–0	[no name]	Always zero; the minimum time resolution is therefore $1/2^{20}$ second.

0x88 Set Time

Access / Notes: EDT_TIME_SET / 32-bit read-write

Reads EDT Time as of the last time the latch time bit was set in 0x8F Time Control.

Bit	Name	Description
31-0	SET_TIME	Write either a 32-bit start time for the second counter, or a 24-bit adjustment time. The value is written into the correct register by setting the appropriate bit in 0x8F Time Control.

0x8C Time Adjust

Access / Notes: EDT_TIME_ADJUST / 24-bit read-only

Reads EDT Time as of the last time the latch time bit was set in 0x8F Time Control.

Bit	Name	Description
23-0	ADJ_TIME	Read the current adjustment time. Adjustment time is explained in EDT Time on page 13; detailed function descriptions are available at www.edt.com/api. Requires that bit 4 be set in 0x8F Time Control.

0x8F Time Control

Access / Notes: EDT_TIME_CTL / 8-bit read-write

Bit	Name	Description
7-6	[no name]	Reserved.
5	ADJUST_PLUS	Set to increment fractional seconds by 2 when the adjustment timer expires. Clear to skip an increment when the adjustment timer expires. Requires that bit 4 be set.
4	ADJUST_EN	Set to enable time adjustment; clear if no adjustment is required.
3	[no name]	Reserved.
2	SET_ADJUST	Set to transfer the contents of 0x88 Set Time into 0x8C Time Adjust.
1	CAP_TIME	Set to capture a 64-bit time value into 0x80 Capture Time.
0	SET_SEC	Set to transfer the contents of 0x88 Set Time into the seconds counter.

0xC0-C3 Intel LXT3108 E1 / T1 LIU Control (Combo 3)

Access / Notes: [no access name] / 8-bit read-write

The Combo 3 has two Intel LXT3108 E1/T1 LIUs, each of which connects to eight E1/T1 channels. The LXT3108 has many programming registers for each channel and cannot be mapped into the same address space, as the LXT384 on the Combo and Combo 2 can be. The `c3_16io.bit` configuration file has four registers to command these two LXT3108s, adding another level of indirection.

The included EDT program file `c3_set_e1.c` sets the LXT3108 for E1 short-haul operation (the same as the LXT384 default) and demonstrates how to program the LXT3108 registers. For details, see the link to the LXT3108 datasheet under Related Resources on page 2.

- 0xC0: This register is programmed with the register address presented to both LIUs.
- 0xC1: Bit 0 of this register controls both LXT3108 power-on reset pin. A value of 0 resets the LXT3108s; a value of 1 enables normal operation.
- 0xC2: Data written to 0xC2 is written to the LXT3108 for channels 7–0, using the address in register 0xC0; data read from 0xC2 is read from the same LXT3108.
- 0xC3: This register functions the same as 0xC2, but for channels 15–8 instead of 7–0.

0xC0-DF Intel LXT384 E1 / T1 Channels 0–7 Control (Combo and Combo 2)

Access / Notes: [no access name] / 8-bit read-write

The LXT384 data sheet describes the contents of the 32 registers that control E1/T1 channels 0–7. (For part information, see Related Resources on page 2.)

The power-up default for these registers operates in E1 mode with jitter attenuation disabled. The oscillator PLL3 must be set for 2.048 MHz for E1 operation. You can use register 0xD0 to set an individual channel's transmit pulse shape for T1. For T1 operation, PLL3 must be set to 1.536 MHz.

Register 0xD0 is written with the channel number (0–7).
Register 0xD1 is written as shown below.

T1 Cable Length Compensation

Code	*Mode*	*Cable length*	*Code*	*Mode*	*Cable length*
0x7	T1	534-655 feet	0x6	T1	400–533 feet
0x5	T1	267–399 feet	0x4	T1	134–266 feet
0x3	T1	0–133 feet	0x0	E1	any

0xE0-0xFF Intel LXT384 E1 / T1 Channels 8–15 Control (Combo and Combo 2)

Access / Notes: [no access name] / 8-bit read-write

Identical to 0xC0-DF above, but for E1/T1 channels 8–15 (instead of 0–7). PLL0 sets the MCLK pin for this chip.

Revision Log

Below is a history of modifications to this guide.

Date	By	Pp	Detail	Release
20100419	PH,DB	28	Corrected name of bits 7-4 to “E3_AI” in 0x4F register.	yes
20100415	PH,TL	16-17	Revised table headings & corrected text typos (not data-significant).	yes
20100408	PH PH,MM PH,DB	All 26-29 19 26	Updated formatting and standardized text. Changed access names for registers 0x48, 49, and 54-57. Changed “16-pin connector” to “15-pin connector” in table. Changed “8-0” to “7-0” in register 0x3C-3D.	yes
20090224	PH	All	Changed “Xilinx” to “FPGA.”	next
20090224	PH	All	Changed “T1/E1” to “E1/T1” and “E3/T3” to / ”T3/E3” to match syntax in register pages (as in “Access = COMBO_E3T3_CTL”).	next
20090224	PH	20	Added Figure 2, showing front of board with all channels set to TX	yes
20070500	LW	All	Added information about Combo 3.	yes
20060000	LW	All	Created guide for new product (Combo).	yes