User’s Guide

DRX16

Mezzanine Board

Dual receiver with 16-bit A/D
for use with an EDT main board

Rev. 2010 March 31

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

DRX16 Mezzanine Board

Overview

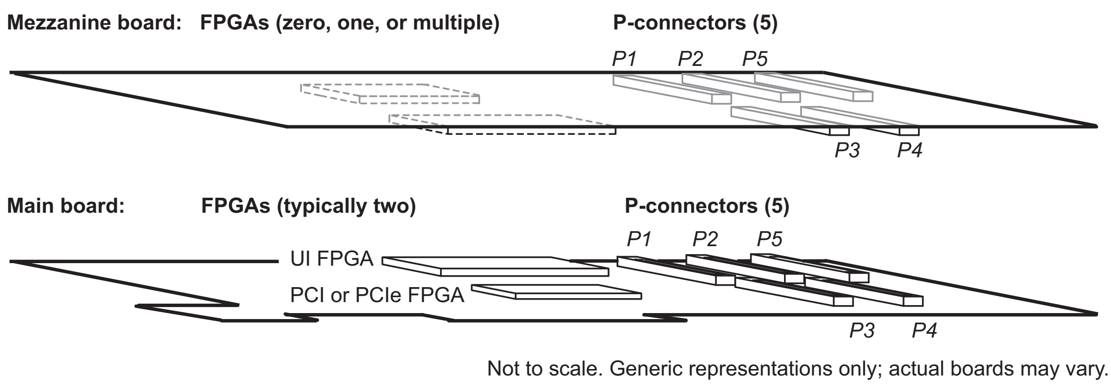
The DRX16 is a dual receiver mezzanine board with 16-bit A/D that digitizes and captures intermediate frequency (IF) signals. For specifications, see Related Resources on page 2.

The DRX16 is paired with an EDT main board (PCIe8 LX / FX) for high-speed DMA and other resources. The board pair provides the field-programmable gate arrays (FPGAs) below.

FPGAs: Mezzanine Board + Main Board

In general, an EDT board pair has P-connectors (to link the two boards) and FPGAs, as shown in Figure 1.

Figure 1. Generic EDT Board Pair



In specific, your DRX16 board pair has the following FPGAs.

- The mezzanine board (DRX16) has one user-programmable FPGA – the *DRX16 FPGA*. To load this FPGA, see Installation on page 3.
- The main board (PCIe8 LX / FX) has two FPGAs:
  - — The *UI FPGA* links the DRX16 FPGA to the PCIe FPGA. To load this FPGA, use the EDT Main Board User's Guide (see link under Related Resources, below).
  - — The *PCI / PCIe FPGA* communicates with the host computer over the PCI or PCIe bus and implements the DMA engine, which transfers data between the board and the host. At powerup, this FPGA automatically loads the correct firmware from the main board's flash ROM.

Related Resources

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

Resource	*EDT webpage*
DRX16 specifications (datasheet)	www.edt.com (on DRX16 product page)
EDT Main Board User's Guide	www.edt.com/manuals/PCD/main_boards.pdf
Time Distribution User's Guide	www.edt.com/manuals/PCD/time-dist.pdf
Application Programming Interface	www.edt.com/api
Installation packages (software downloads):	www.edt.com/software.html
Windows, Linux, Solaris, Mac OS	

The third-party resources below can provide specifications for parts used on the DRX16.

Manufacturer	*Part Number*	*Manufacturer's website*
Silicon Labs	Si5368	www.silabs.com
Texas Instruments	ADS5485	www.ti.com
"	MSP430F7222	"
"	PGA870	"
Xilinx	XC6VLX240T	www.xilinx.com

Installation

EDT provides installation packages for all supported operating systems (Windows, Linux, Solaris, Mac OS). These packages are provided on the EDT installation disk that ships with every EDT product.

However, to prevent installation package version issues, EDT recommends going to the EDT website and doing one of the following:

- For a new application, download the latest package.
- For an existing application, use the same package that was used to build it (from your own or EDT's archives), or recompile / relink the application with the latest installation package download.

In either case, to find the installation package you need (either the latest one or an archived version), see Related Resources on page 2.

Included Files

When installation is complete, open the top-level `EDT/PCD` directory to find the subdirectories and other resources that pertain to the DRX16.

- Relevant subdirectories include `bitfiles`, `flash`, `pci_config`, and `srxl`.
- Relevant other resources include applications and utilities.

bitfiles

This subdirectory contains FPGA configuration files for each user-programmable FPGA available on your board pair (DRX16 + PCIe8 LX / FX). For proper FPGA configuration, load the appropriate files below (see Loading the Firmware on page 5).

For the DRX16 mezzanine board: In `bitfiles`, find the `XC6VLX240T` subdirectory. That subdirectory contains this DRX16 FPGA configuration file:

`drx16_top_lx2.bit`	Configures the DRX16 for dual IF capture.

For the PCIe8 LX / FX main board: In `bitfiles`, find the correct subdirectory for the FPGA you ordered (`XC5VLX110T / 220T / 330T`). That subdirectory contains this UI FPGA configuration file:

`drx16_2in.bit`	Configures the PCIe8 LX / FX UI FPGA for use with the DRX16

flash

This subdirectory relates to the PCIe FPGA on the main board. The main board ships from EDT with the correct configuration file below already preloaded in flash ROM, and loads automatically at powerup.

In `flash`, find the `ep2sgx30d` subdirectory. That subdirectory contains this PCIe FPGA configuration file:

`pe8lx16.bit`	Configures the PCIe FPGA on the PCIe8 LX / FX main board.

pci_config

This subdirectory contains `.cfg` files for software initialization and other purposes.

NOTE EDT does not currently provide `.cfg` files for the DRX16. However, the following discussion of these files may prove helpful to you.

The `.cfg` files are editable text files that run like scripts to configure EDT boards and prepare the boards to perform DMA. The commands in these files are defined in a C application named `initpcd`. When you invoke `initpcd`, you specify which `.cfg` file to use with the `-f` flag.

A typical `.cfg` file loads an FPGA configuration file into the UI FPGA on the main board, and then sets up various registers to prepare for DMA transfers. Some `.cfg` files may load an FPGA configuration file into an FPGA residing on the mezzanine board.

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: drx16_2in.bit
command_reg: 0x08
run_command: mezzload
```

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 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`, contact EDT.

srxl

This subdirectory contains source files for the DRX16 library functions and debugger application.

`lib_drx16.c`	C source for routines called by the example applications. These routines in turn call routines in `libedt.c`, EDT's API library (see Related Resources on page 2).
`lib_drx16.h`	A header file for `lib_drx16.c`.
`drx16_debugger.c`	C source for the `drx16_debugger` application.
`srxl2_fft.c`	The C source for the `fft` application.

Applications and Utilities

In addition to the above subdirectories, the top-level `EDT/PCD` directory contains applications and utilities that you can use for board initialization and configuration, register access, and testing (in many cases, C source is included to give you a starting point for writing your own applications). Some commonly useful applications and utilities are listed below; for the complete list, see the `README` file.

`bitload`	Loads the configuration files for the UI FPGA on the main board.
`chkprbs15`	Checks DMA data against a 15-bit pseudorandom bit sequence in the specified channels.
`extbdid`	Displays the ID and revision number of the mezzanine board installed.
`initpcd`	Initializes and configures the mezzanine board.
`mezzload`	Loads the configuration files for the main board UI FPGA and the mezzanine board FPGA.
`pciload`	Shows currently installed EDT boards, outputs the date and revision number of the firmware in the PROM, and can be used to update the firmware in the PROM.

`pdb`	Enables interactive reading and writing of the registers on the main board UI FPGA and the mezzanine board FPGA.
`simple_getdata`	Provides an example of several DMA-related operations, including reading data from the connector interface, writing the data to a file, and measuring input rate.
`drx16_debugger`	Exercises most of the mezzanine board functions and provides debugging aids. The example application `drx16_debugger` provides a way to: - read and write the DRX16 and UI FPGA registers; - access and set values on the DRX16 devices; - capture data from channel 0 or channel 1; and - perform complex Fast Fourier Transforms (FFT) on the captured data. To run the example application, at the Pcd Utilities prompt, enter: `drx16_debugger` At the DRX16 debugger prompt, enter the `h` command for a list of all the usage options and their descriptions.
`srxl2_fft.c`	Performs a Fast Fourier Transform (FFT) on both real and imaginary components of the captured data.

Building Applications

Executables and PCD source files are located in the `EDT PCD` directory. Therefore, if you need to rebuild an application, run `make` in this top-level directory.

Table 1 shows platform-specific details (namely, the recommended compiler and the location of the library and the debugger application) for each major platform.

Table 1. DRX16 platform-specific details

	Recommended compiler	Library & debugger location
Windows	Install Microsoft Visual C compiler (or contact EDT to use `gcc`)	`srxl` directory
Solaris	Install Sun Workshop C compiler (or contact EDT to use `gcc`)	`./srxl/` subdirectory
Linux	Use `gcc` compiler (typically included with Linux installation)	`./srxl/` subdirectory

After building an application, enter the `--help` command line option to see a list of usage options and descriptions.

Loading the Firmware

For FPGA configuration, follow the instructions below, replacing italic terms (such as *`unit_number`*) with the appropriate values.

1. Run `pciload` to determine the unit number of the PCIe8 LX / FX main board (by default, 0), and to verify that the host detects the main board and that the main board is loaded with the appropriate firmware.
2. If the PROM ID includes the string `pcie8LX16`, then 16-channel firmware is already loaded. If not, load the appropriate firmware with the appropriate command:

   `pciload -u` *`unit_number firmware_file`*

   (where *`firmware_file`* is `pe8lx16`)
3. At the prompt, press Enter to confirm the loading operation.
4. Power-cycle the system.

5. Load the correct configuration files for the main board UI FPGA and the mezzanine board FPGA:

   `mezzload -u` *`unit_number`* `-b` *`mezzanine_board_bitfile`* `-B` *`main_board_UI_FPGA`*

   For example, a PCIe8 LX / FX main board with the DRX16 option loaded in unit 0 would be:

   ```
   mezzload -u 0 -b drx16_top_lx.bit -B drx16_2in.bit
   ```

6. Watch for the onscreen message which verifies that the files are loaded.

Getting the Board ID

After you install the DRX16 and load the firmware, you'll need to get the extended board ID and revision information. To do so, see 0x7F Board ID.

Board Architecture and IDX Modules

This section covers the architecture of the DRX16 mezzanine board and its two IDX modules.

Figure 2 shows the key components of the DRX16, including:

- two independent channels, each providing one IF direct module (IDX), one optional programmable gain amplifier (PGA870), and one 16-bit 200 MSPS analog-to-digital converter (ADC, ADS5485).
- one user-programmable FPGA (XC6VLX240T).
- one user-programmable any-rate clock multiplier (Si5368).
- four additional ports – one reference input, one reference output, one sample clock I/O, and one time code input.

Each channel receives, samples, and digitizes the IF signals, which then are captured in the user-programmable DRX16 FPGA. From there, the digitized data can be routed to the UI FPGA on the main board – either for more processing, or for DMA transfer to the host computer.

The DRX16 requires a 10 MHz reference as a timebase. This timebase, which can be selected from either the reference input connector or the internal 10 MHz TCXO reference clock on the board, is available via the reference output connector.

The sample clock I/O connector can be used either for sample clock output, or as the timebase for the Si5368. The sample clock for each channel can be programmed individually from 10 to 200 MHz.

The time code input accepts IRIG-B or one-pulse-per-second (1pps) format, for timestamping of the data.

Figure 2. DRX16 Mezzanine Board – Key Components

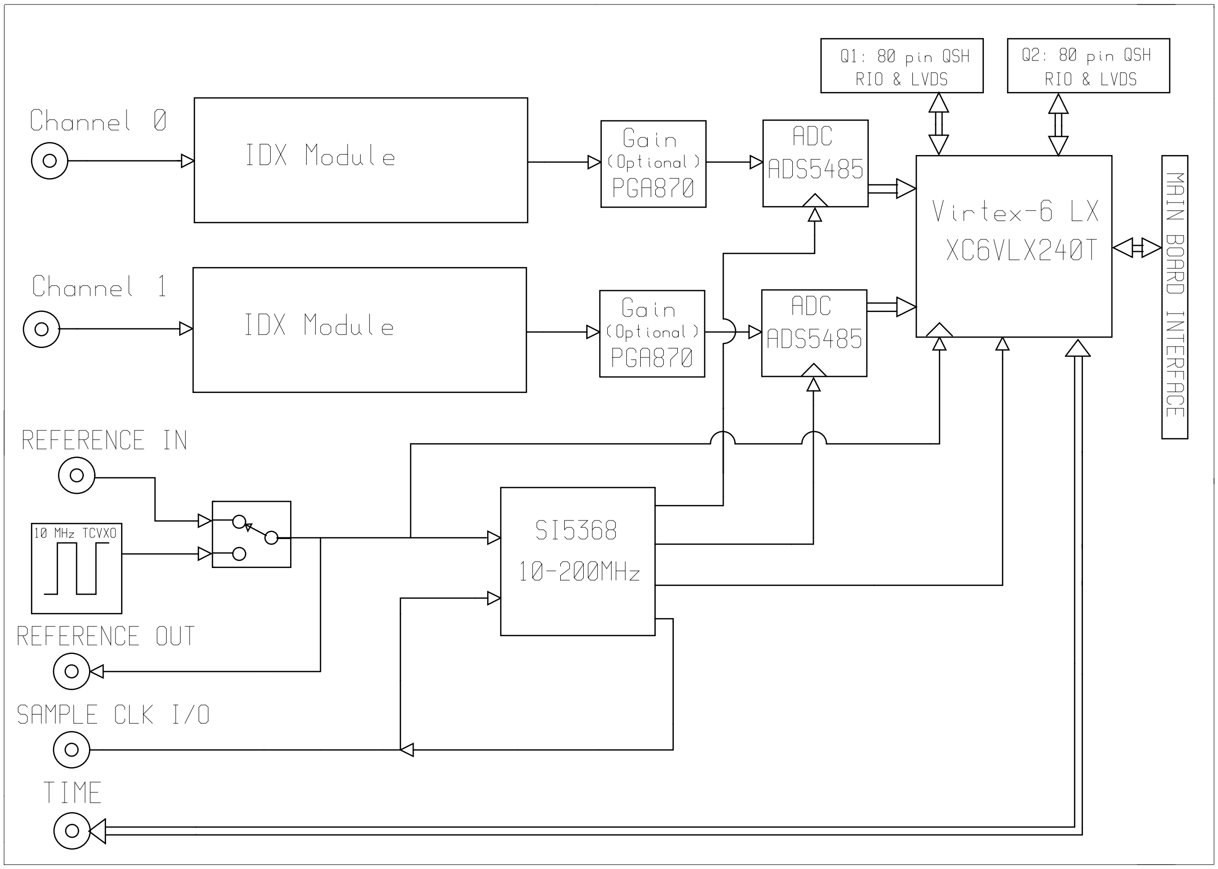


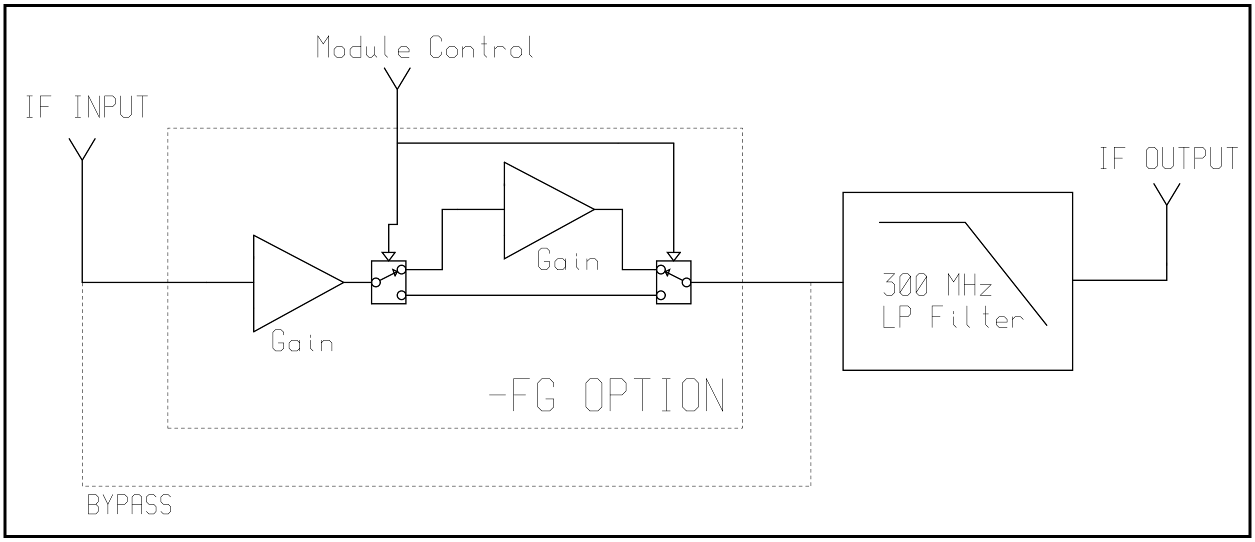
Figure 3 shows the IDX configuration on the board. In the standard configuration, each IDX accepts IF frequencies up to 200 MHz and has no gain. This configuration has a 300 MHz low-pass filter to eliminate high frequency signals from aliasing into the desired bandwidth.

IDX options include a fixed gain option, and either 50- or 70-ohm input signaling; these options must be specified at the time of ordering. The bandwidth of a specific IF depends on external filtering (not provided).

If the DRX16 was ordered with a PGA870, the IDX output signal is connected to the PGA870. Otherwise, the signal is sampled directly by the ADC on the DRX16.

For dynamic range with the various ordering options, follow the link under Related Resources on page 2 to see the IDRX16 datasheet.

Figure 3. DRX16 Mezzanine Board – IF Direct Module (IDX)



To set up this module:

1. Turn on channel power; see 0x50 Board Control.
2. Select either the onboard 10-MHz TCXO reference or an external reference; see 0x50 Board Control.
3. Program the sample clock; see Sample Clock on page 11.
4. Program the gain on the IDX module (if you ordered the gain option); see 0x50 Board Control.
5. Program the gain on the PGA870 (if you ordered the PGA870); see Programming the DRX16 Devices on page 10.
6. Acquire a signal; see Acquiring a Signal and Testing DMA Transfer on page 12.

General Board Control

On the DRX16, you can control capabilities related to power, reference clock, and time code. The locations of the components that support these capabilities are shown in Figure 5 on page 19.

Power

On the DRX16, when both channels and all other components are fully powered up and running, the power requirements may exceed PCIe specifications for power per slot. If you have this problem, try these possible solutions:

- Connect the auxiliary power cable on the board to any 12-volt power source to boost power.
- If you're using only one module channel, power down the other module channel; see 0x50 Board Control.

Reference Clock

The DRX16 requires a 10 MHz signal from a reference clock, to which all other clocks on the board are phase-locked. For this reference clock, you can select either the internal TCXO or an external reference via bit 7 (EXT_REF_SEL) in 0x50 Board Control.

The selected timebase is available via the 10 MHz output connector.

NOTE When switching between internal and external reference clocks, check the MMCM_LOCKED status bit in 0x58 Clock Control. If it is not locked, then perform MMCM_RESET. This clock manager creates an internal clock, which transfers data between the DRX16 FPGA and the UI FPGA.

Time Code

For time code input, the DRX16 uses the same three registers as does the EDT Time Distribution board. These three registers appear in both user's guides, but with different addresses, as shown below.

Table 2. Serial Peripheral Interface (SPI) Registers for Time Code Input

	DRX16 board	Time Distribution board
SPI Data	0x5C	0x60
SPI Status and Control	0x5D	0x61
SPI Strobe	0x5E	0x62

For details on the time code circuitry and the 7-pin Lemo pinout, see Related Resources on page 2 for a link to the Time Distribution User's Guide.

Programming the DRX16 Devices

For each channel (channel 0 and channel 1) on the DRX16, you'll need to program certain components by reading and writing the appropriate registers in the FPGA. These components are:

- the gain adjustments (if the board is ordered with this option); and
- the sample clock synthesizer.

Gain

If the gain option is ordered, each channel provides a programmable gain amplifier (PGA870). For each channel, the gain can be adjusted from -11.5 dB to +20 dB, in increments of 0.5 dB. The gain is set by a 6-bit control word, using the following equation...

`Gain = 20 dB+ 0.5 dB x (N - 63)`

...where *N* is the decimal equivalent of the 6-bit binary number.

For example:

- To select the lowest gain (-11.5 dB), set the control word to 000000.
- To select the highest gain (+20 dB), set the control word to 111111.

To set the gain, follow the steps below, using 0x55 PGA Control – Channel 0 and 0x56 PGA Control – Channel 1.

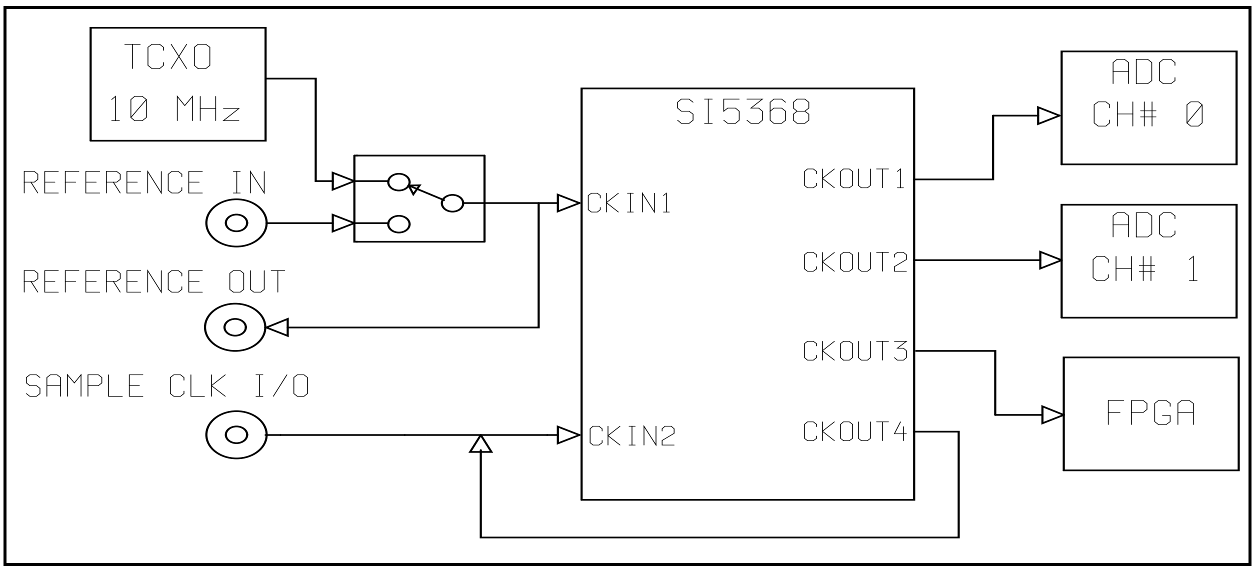
1. For each channel, write the gain control word and set the latch:
   — In bits 5–0 (CH0_B or CH1_B), write the gain control word;
   — In bit 7 (CH0_LATCH or CH1_LATCH), set the latch.
2. For each channel, to update the gain in the PGA870, clear the latch in bit 7 (CH0_LATCH or CH1_LATCH).

Sample Clock

For each channel, the module's output is sampled and digitized using an ADS5485 16-bit ADC.

For each ADC, the sample clock rate is generated to any frequency from 10 to 200 MHz by a low-jitter precision clock multiplier (Si5368 – see Related Resources on page 2). Figure 4 shows the connections.

Figure 4. Sample Clock Connectors



The sample clock is set by writing and reading the appropriate registers in the Si5368.

NOTE Writing register 0x53 or 0x54 will cause the hardware to serialize the data and transfer it to the Si5368; these serial transfers are slower than the register programming interface. Therefore, to prevent errors in the serial datastream, wait for bit 4 (XFER_BUSY) in 0x51 Sample Clock Control to clear before performing the next read or write to any Si5368 register.

To write an Si5368 register:

1. Write the address of the Si5368 target register to 0x52 Sample Clock Write Address.
2. Write the data to 0x53 Sample Clock Write Data.
3. Wait for bit 4 (XFER_BUSY) in 0x51 Sample Clock Control to clear before performing the next read or write to any Si5368 register.

To read an Si5368 register:

1. Write the address of the Si5368 target register to 0x54 Sample Clock Read.
2. Wait for bit 4 (XFER_BUSY) in 0x51 Sample Clock Control to clear.
3. Read the data from 0x54 Sample Clock Read.

Acquiring a Signal and Testing DMA Transfer

For acquiring a signal, the EDT-provided FPGA configuration files enable continuous capture and transfer of data, via DMA, to the host computer.

For example application code that you can use to acquire data, see the `drx16_debugger` source file.

To start acquiring a signal with the DRX16 and the PCIe8 LX / FX main board, follow the steps below.

1. Clear and then set bit 3 (CMD_EN) in 0x10 Channel Enable Low – DMA Channels 7–0 to reset and enable DMA FIFOs in the UI FPGA.
2. Make the appropriate EDT driver call(s) to start DMA on the main board.
3. Set the appropriate CH_ENABLE bit (0 or 1) in 0x10 Channel Enable Low – DMA Channels 7–0.

To test DMA transfer, the DRX16 can automatically generate a PRBS15 test pattern, which in turn can be used with the `checkprbs15` application software to verify the DMA is working.

To enable the PRBS15 test pattern, set bit 0 (PRBS_EN) in 0x57 M2U DMA.

The PRBS test pattern is on DMA channels 0 and 1 – so when the test pattern is enabled, you cannot acquire data from the module on either channel of the DRX16.

Registers

This section shows the DRX16 registers, arranged by hexadecimal number.

0x00 Command

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

Bits	Name	Description
7-4	[no name]	Not used.
3	CMD_EN	Set to enable the required DMA channels in 0x10 Channel Enable Low – DMA Channels 7–0 and 0x11 Channel Enable High – DMA Channels 15–8. Clear to reset all DMA channels, flush the FIFOs, and clear all under- and overflow bits.
2-0	[no name]	Not used.

0x10 Channel Enable Low – DMA Channels 7–0

Access / Notes: SSD16_CHENL / 8-bit read-write

Bits	Name	Description
7-0	CH_ENABLE	Set to enable corresponding DMA channel; clear to reset.

0x11 Channel Enable High – DMA Channels 15–8

Access / Notes: SSD16_CHENH / 8-bit read-write

Bits	Name	Description
15-8	CH_ENABLE	Set to enable corresponding DMA channel; clear to reset.

0x50 Board Control

Access / Notes: DRX16_AD_BOARD_CTRL / 8-bit read-write

Bits	Name	Description
7	CH0_IDX_CTRL	For channel 0 IDX gain: Set for 46 dB; clear for 23 dB.
6	CH1_IDX_CTRL	For channel 1 IDX gain: Set for 46 dB; clear for 23 dB.
5	CH0_DITHER	For channel 0 ADC dither: Set to enable; clear to disable.
4	CH1_DITHER	For channel 1 ADC dither: Set to enable; clear to disable.
3	CH0_ENABLE	For channel 0 power: Set to enable; clear to disable.
2	CH1_ENABLE	For channel 1 power: Set to enable; clear to disable.
1	REF_SEL	For reference: Set for external; clear for internal.
0	RESET	To perform global reset: Set, then clear.

0x51 Sample Clock Control

Access / Notes: DRX16_SCLK_CTRL / 8-bit read-write

Bits	Name	Description
7	LOL	Read only; if set, the Si5368 has not locked to the sample clock.
6	LOS_REF	Read only; if set, the Si5368 is not receiving a signal from the reference clock.
5	LOS_SCLK	Read only; if set, the Si5368 is not receiving a signal from the external sample clock input.
4	XFER_BUSY	Read only. The firmware sets this bit when the sample clock serializer is busy; wait until it is clear before starting any new read or write to the Si5368.
3–1	[no name]	Not used.
0	SCLK_RESET	To reset the Si5368: Set, then clear.

0x52 Sample Clock Write Address

Access / Notes: DRX16_SCLK_ADDR / 8-bit read-write

The address of the Si5368 register to which you wish to write.

0x53 Sample Clock Write Data

Access / Notes: DRX16_SCLK_DATA / 8-bit read-write

The data that you wish to write to the Si5368 register specified by the address in 0x52 Sample Clock Write Address.

0x54 Sample Clock Read

Access / Notes: DRX16_SCLK_READ / 8-bit read-write

Write the address of the desired Si5368 register. Wait for bit 4 (XFER_BUSY) in 0x51 Sample Clock Control to clear; then read the data from the requested register.

0x55 PGA Control – Channel 0

Access / Notes: DRX16_PGA_CTRL_CH0 / 8-bit read-write

Bits	Name	Description
7	CH0_LATCH	For channel 0, to latch gain control word: Set, then clear.
6	[no name]	Not used.
5–0	CH0_B	For channel 0: Gain control word.

0x56 PGA Control – Channel 1

Access / Notes: DRX16_PGA_CTRL_CH1 / 8-bit read-write.

Bits	Name	Description
7	CH1_LATCH	For channel 1, to latch gain control word: Set, then clear.
6	[no name]	Not used.
5–0	CH1_B	For channel 1: Gain control word.

0x57 M2U DMA

Access / Notes: DRX16_M2U_DMA / 8-bit read-write.

Bits	Name	Description
7–1	[no name]	Not used.
0	PRBS_EN	Enable PRBS data on DMA channels 0 and 1.

0x58 Clock Control

Access / Notes: DRX16_CLK_CTRL / 8-bit read-write

After switching between internal and external reference, check the locked status below and reset if not locked.

Bits	Name	Description
7	MMCM_LOCKED	Read only. If set, clock manager is locked.
6–1	[no name]	Not used.
0	MMCM_RESET	To reset clock manager: Set, then clear.

0x59 Idelay Control

Access / Notes: DRX16_IDELAY_CTRL / 8-bit read-write

Bits	Name	Description
7–5	[no name]	Not used.
4	CH1_IDLY_RESET	For channel 1 IDELAY on ADC data input: Set, then clear to reset.
3–1	[no name]	Not used.
0	CH0_IDLY_RESET	For channel 0 IDELAY on ADC data input: Set, then clear to reset.

0x5B Time Control

Access / Notes: DRX16_TIME_CTRL / 8-bit read-write

Bits	Name	Description
7–2	[no name]	Not used.
1	RELATIVE_TIME_RESET	Set, then clear to reset relative time.
0	MASTER_ENABLE	Set if board is master; clear if slave.

0x5C SPI Data

Access / Notes: DRX16_SPI_DATA / 8-bit read-write

When read, bits read from the input FIFO; when written, bits write to the output FIFO.

0x5D SPI Status and Control

Access / Notes: DRX16_SPI_STAT_CTRL / 8-bit read-write

Bits	Name	Description
7	CHIP_ENABLE	Detects which main board is connected to the DRX16 master header. A value of one indicates the master header; otherwise reads zero.
6	IRIGB_ PULSE_IN	When set, indicates that the one-pulse-per-second signal has been detected from the incoming IRIG signal, and it has been processed by, and passed through, the MSP430.
5	MODEM_ PULSE_IN	When set, indicates that the one-pulse-per-second signal has been detected from the satellite modem and passed directly through, without any processing by the MSP430.
4	[no name]	Not used.; reads as zero.
3	FIFO_OUT_ EMPTY	When set, indicates the output FIFO is empty.
2	FIFO_OUT_ FULL	When set, indicates the output FIFO is full.
1	FIFO_IN_ EMPTY	When set, indicates the input FIFO is empty.
0	When written: RESET When read: FIFO_IN_OV	On write: toggle this bit to reset the SPI data path. On read: when set, indicates the input FIFO has overflowed. Data may be lost.

0x5E SPI Strobe

Access / Notes: DRX16_SPI_STROBE / 8-bit read-write

Write (any value) to this register to advance the input FIFO.

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 3. EDT Board ID and Extended Board ID (CPLD) – part 1 of 2

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.

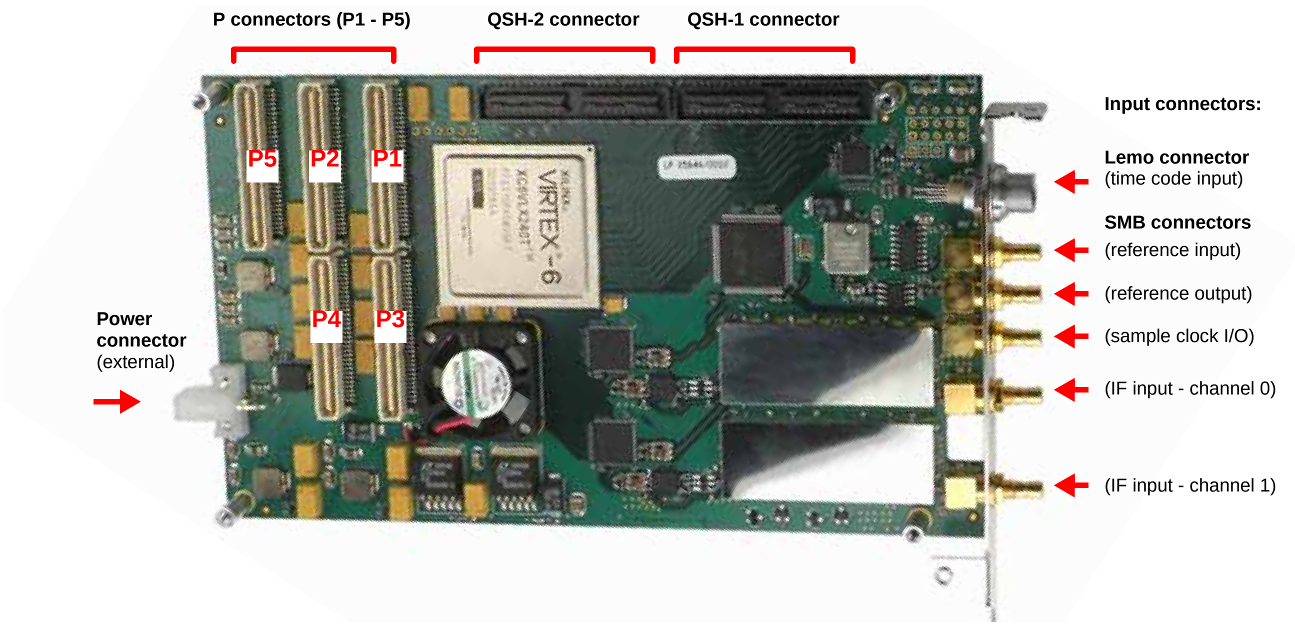
Table 3. EDT Board ID and Extended Board ID (CPLD) – part 2 of 2

Board ID Register, Bits 3–0	Extended Board ID	Board Name	Detail
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.

Connectors

Figure 5, with its accompanying notes, identifies the connectors on the DRX16.

Figure 5. DRX16 Connectors and Pins



5 P-connectors	These five 64-pin CMC-type connectors (P1 – P5) link the DRX16 to the main board.
6 connectors	These six connectors described below. One Lemo connector for time code input. Five SMB connectors – one for each of the following: - Reference input. - Reference output. - Sample clock I/O. - IF input on channel 0. - IF input on channel 1.
1 power connector	This connector is for the external power source.
2 QSH connectors	These 80-pin QSH connectors can be used to debug, or to link up to two other DRX16 boards in a system. Each QSH connector provides: - 4 channels of RocketIO transceivers; - 13 LVDS RX pairs; - 13 LVDS TX pairs; - 2 LVDS RX clock pairs; and - 2 LVDS TX clock pairs.

Pinouts

This section contains a series of tables showing the pinouts from the DRX16 FPGA to other devices:

Tables 4 through 8 cover the pinouts from the DRX16 FPGA to the P-connectors. For these tables, the meaning of each function is shown below.

If the function says...	...the meaning is...
+5 V	These pins are the 5-volt supply.
+12 V	These pins are the 12-volt supply.
BD ID*x*[0-3]	These pins are for mezzanine board identification (the lowest four bits of 0x7F Board ID).
extbdid...	These pins are the interface for the extended board ID.
free	These pins have wires connecting the DRX16 FPGA to the UI FPGA on the main board, so this signal can be accessed by your firmware.
FPGA...	These pins are dedicated for programming the DRX16 FPGA.
ground	These pins are the ground connection.
unused	These pins do not have wires connecting the DRX16 FPGA to any outside device, so the signals cannot be accessed.

Tables 9 through 12 show the pinouts from the DRX16 FPGA to the other devices on the board.

Table 4. DRX16 FPGA to P1 Connector

P1 Pin	DRX16 FPGA Pin	Function	P1 Pin	DRX16 FPGA Pin	Function
01	L8	FPGA PROG_B	02	–	unused
03	–	ground	04	AK8	free
05	AL8	free	06	AK9	free
07	–	BD ID0	08	–	+5 V
09	AL9	free	10	AM10	free
11	–	ground	12	AL10	free
13	AM11	free	14	–	ground
15	–	ground	16	AM12	free
17	AN9	free	18	–	+5 V
19	–	unused	20	AL13	free
21	AP9	free	22	AK14	free
23	AN10	free	24	–	ground
25	–	ground	26	AL15	free
27	AP10	free	28	AM17	free
29	AP11	free	30	–	+5 V
31	–	unused	32	AK19	free
33	AN12	free	34	–	ground
35	–	ground	36	AL19	free
37	AP12	free	38	–	+5 V
39	–	ground	40	AL20	free
41	AN13	free	42	AM21	free
43	AN14	free	44	–	ground
45	–	unused	46	AM23	free
47	AP14	free	48	AK24	free
49	AN15	free	50	–	+5 V
51	–	ground	52	AM25	free
53	AP15	free	54	AM26	free
55	AP16	free	56	–	ground
57	–	unused	58	AL28	free
59	AN17	free	60	AK29	free
61	AP17	free	62	–	+5 V
63	–	ground	64	L23	free

Table 5. DRX16 FPGA to P2 Connector

P2 Pin	DRX16 FPGA Pin	Function	P2 Pin	DRX16 FPGA Pin	Function
01	–	+12 V	02	AH8	free
03	AJ9	free	04	AJ10	free
05	AK11	free	06	–	ground
07	–	ground	08	AL11	free
09	AK13	free	10	AK12	free
11	–	BD ID1	12	–	unused
13	AJ14	free	14	–	BD ID2
15	–	unused	16	–	BD ID3
17	AM13	free	18	–	ground
19	AL14	free	20	AJ15	free
21	–	ground	22	AL16	free
23	AM15	free	24	–	unused
25	AM16	free	26	AK17	free
27	–	unused	28	AK18	free
29	AL18	free	30	–	ground
31	AM18	free	32	AJ19	free
33	–	ground	34	AJ20	free
35	AM20	free	36	–	unused
37	–	ground	38	AL21	free
39	AM22	free	40	–	ground
41	–	unused	42	AK22	free
43	AL23	free	44	–	ground
45	P8	FPGA INIT	46	J9	free
47	–	ground	48	AK23	free
49	AL24	free	50	–	unused
51	AL25	free	52	AJ24	free
53	–	unused	54	AJ25	free
55	AM27	free	56	–	ground
57	AM28	free	58	–	unused
59	–	ground	60	AK26	free
61	AL29	free	62	–	unused
63	–	ground	64	AN18	free

Table 6. DRX16 FPGA to P3 Connector

P3 Pin	DRX16 FPGA Pin	Function	P3 Pin	DRX16 FPGA Pin	Function
01	AN20	free	02	–	ground
03	–	ground	04	AP19	free
05	AP20	free	06	AN19	free
07	AP21	free	08	–	ground
09	–	unused	10	AP32	free
11	AN22	free	12	AN33	free
13	AP22	free	14	–	ground
15	–	ground	16	K8	FPGA CCLK
17	AN23	free	18	AM33	free
19	AN24	free	20	–	ground
21	–	unused	22	AL33	free
23	AP24	free	24	AL34	free
25	AN25	free	26	–	ground
27	–	ground	28	AK32	free
29	AP25	free	30	AK33	free
31	AP26	free	32	–	ground
33	–	ground	34	AK34	free
35	AN27	free	36	AJ32	free
37	AP27	free	38	–	ground
39	–	unused	40	AJ34	free
41	AN28	free	42	AH33	free
43	AP29	free	44	–	ground
45	–	ground	46	AH34	free
47	AN30	free	48	AH28	free
49	AP30	free	50	–	ground
51	–	ground	52	AG30	free
53	AF29	free	54	AF30	free
55	AD30	free	56	–	ground
57	–	unused	58	AE28	free
59	AD31	free	60	AD27	free
61	AC32	free	62	–	ground
63	–	ground	64	AD29	free

Table 7. DRX16 FPGA to P4 Connector

P4 Pin	DRX16 FPGA Pin	Function	P4 Pin	DRX16 FPGA Pin	Function
01	–	+12 V	02	AK27	free
03	AM30	free	04	AK28	free
05	AL30	free	06	–	ground
07	AP31	free	08	AN29	free
09	AN31	free	10	AJ29	free
11	–	ground	12	AJ30	free
13	AP33	free	14	AM31	free
15	AM32	free	16	–	ground
17	AN34	free	18	AL31	free
19	AH29	free	20	AK31	free
21	–	ground	22	AJ31	free
23	AH32	free	24	AH30	free
25	AH27	free	26	–	ground
27	AF24	free	28	AG25	free
29	AF25	free	30	AG27	free
31	–	ground	32	AG28	free
33	AF26	free	34	AG26	free
35	AE22	free	36	–	ground
37	R8	FPGA DONE	38	AE26	free
39	AE27	free	40	AF28	free
41	–	ground	42	AD25	free
43	AD26	free	44	AE29	free
45	AE31	free	46	–	ground
47	AE33	free	48	AE32	free
49	AE34	free	50	AF20	free
51	–	ground	52	AG21	free
53	AD32	free	54	AF21	free
55	AC33	free	56	–	ground
57	AC34	free	58	AE21	free
59	AE23	free	60	AD21	free
61	–	ground	62	AE24	free
63	AG31	free	64	AD24	free

Table 8. DRX16 FPGA to P5 Connector

P5 Pin	DRX16 FPGA Pin	Function	P5 Pin	DRX16 FPGA Pin	Function
01	–	+12 V	02	AG8	free
03	AD9	free	04	AH9	free
05	AH10	free	06	–	ground
07	AE9	free	08	AJ11	free
09	AF9	free	10	AJ12	free
11	–	ground	12	AH12	free
13	AG10	free	14	AH13	free
15	AG11	free	16	–	ground
17	AG12	free	18	AH14	free
19	AG13	free	20	AH15	free
21	–	ground	22	AK16	free
23	AF14	free	24	AJ17	free
25	AG15	free	26	–	ground
27	AJ16	free	28	AH18	free
29	AH17	free	30	AH19	free
31	–	ground	32	AH20	free
33	AG18	free	34	AK21	free
35	AF19	free	36	–	ground
37	AG20	free	38	AJ22	free
39	AJ21	free	40	–	extbdid CLK
41	–	ground	42	–	extbdid DATA
43	AH22	free	44	AH23	free
45	AG22	free	46	–	ground
47	AG23	free	48	AH24	free
49	AF23	free	50	AH25	free
51	–	ground	52	AJ26	free
53	AF11	free	54	AF10	free
55	AE12	free	56	–	ground
57	AE14	free	58	AF13	free
59	H8	FPGA DIN	60	AJ27	free
61	–	ground	62	AG16	free
63	AF18	free	64	–	unused

Table 9. DRX16 FPGA to QSH-1 Connector

QSH-1 Pin	DRX16 FPGA Pin	Function	QSH-1 Pin	DRX16 FPGA Pin	Function
01	–	RX_REFCLK +	02	–	TX_REFCLK +
03	–	RX_REFCLK –	04	–	TX_REFCLK –
05	N3	RX_RIO [0]+	06	M1	TX_RIO [0]+
07	N4	RX_RIO [0]–	08	M2	TX_RIO [0]–
09	L3	RX_RIO [1]+	10	K1	TX_RIO [1]+
11	L4	RX_RIO [1]–	12	K2	TX_RIO [1]–
13	K5	RX_RIO [2]+	14	H1	TX_RIO [2]+
15	K6	RX_RIO [2]–	16	H2	TX_RIO [2]–
17	J3	RX_RIO [3]+	18	F1	TX_RIO [3]+
19	J4	RX_RIO [3]–	20	F2	TX_RIO [3]–
21	D11	RX_CLK0 +	22	L15	TX_CLK0 +
23	E11	RX_CLK0 –	24	L14	TX_CLK0 –
25	F14	RX_DATA [0]+	26	H15	TX_DATA [0]+
27	E14	RX_DATA [0]–	28	J15	TX_DATA [0]–
29	H12	RX_DATA [1]+	30	G18	TX_DATA [1]+
31	J12	RX_DATA [1]–	32	H18	TX_DATA [1]–
33	G12	RX_DATA [2]+	34	H17	TX_DATA [2]+
35	H13	RX_DATA [2]–	36	G17	TX_DATA [2]–
37	G13	RX_DATA [3]+	38	J17	TX_DATA [3]+
39	H14	RX_DATA [3]–	40	J16	TX_DATA [3]–
41	H10	RX_DATA [4]+	42	F16	TX_DATA [4]+
43	G10	RX_DATA [4]–	44	G16	TX_DATA [4]–
45	A11	RX_DATA [5]+	46	E16	TX_DATA [5]+
47	B11	RX_DATA [5]–	48	D16	TX_DATA [5]–
49	G11	RX_DATA [6]+	50	E18	TX_DATA [6]+
51	F11	RX_DATA [6]–	52	D17	TX_DATA [6]–
53	K13	RX_CLK [1]+	54	K16	TX_CLK [1]+
55	K12	RX_CLK [1]–	56	L16	TX_CLK [1]–
57	A13	RX_DATA [7]+	58	A16	TX_DATA [7]+
59	A14	RX_DATA [7]–	60	B16	TX_DATA [7]–
61	E13	RX_DATA [8]+	62	G15	TX_DATA [8]+
63	F13	RX_DATA [8]–	64	F15	TX_DATA [8]–
65	C13	RX_DATA [9]+	66	A15	TX_DATA [9]+
67	C12	RX_DATA [9]–	68	B15	TX_DATA [9]–
69	D12	RX_DATA [10]+	70	F18	TX_DATA [10]+
71	E12	RX_DATA [10]–	72	E17	TX_DATA [10]–
73	D14	RX_DATA [11]+	74	D15	TX_DATA [11]+
75	C14	RX_DATA [11]–	76	C15	TX_DATA [11]–
77	B12	RX_DATA [12]+	78	C17	TX_DATA [12]+
79	B13	RX_DATA [12]–	80	B17	TX_DATA [12]–

Table 10. DRX16 FPGA to QSH-2 Connector

QSH-2 Pin	DRX16 FPGA Pin	Function	QSH-2 Pin	DRX16 FPGA Pin	Function
01	–	RX_REFCLK +	02	–	TX_REFCLK +
03	–	RX_REFCLK –	04	–	TX_REFCLK –
05	G3	RX_RIO [0]+	06	D1	TX_RIO [0]+
07	G4	RX_RIO [0]–	08	D2	TX_RIO [0]–
09	E3	RX_RIO [1]+	10	C3	TX_RIO [1]+
11	E4	RX_RIO [1]–	12	C4	TX_RIO [1]–
13	D5	RX_RIO [2]+	14	B1	TX_RIO [2]+
15	D6	RX_RIO [2]–	16	B2	TX_RIO [2]–
17	B5	RX_RIO [3]+	18	A3	TX_RIO [3]+
19	B6	RX_RIO [3]–	20	A4	TX_RIO [3]–
21	H28	RX_CLK0 +	22	B20	TX_CLK0 +
23	H29	RX_CLK0 –	24	C19	TX_CLK0 –
25	D24	RX_DATA [0]+	26	B18	TX_DATA [0]+
27	E24	RX_DATA [0]–	28	C18	TX_DATA [0]–
29	C24	RX_DATA [1]+	30	A18	TX_DATA [1]+
31	C25	RX_DATA [1]–	32	A19	TX_DATA [1]–
33	D25	RX_DATA [2]+	34	E19	TX_DATA [2]+
35	D26	RX_DATA [2]–	36	D19	TX_DATA [2]–
37	B25	RX_DATA [3]+	38	C20	TX_DATA [3]+
39	A25	RX_DATA [3]–	40	D20	TX_DATA [3]–
41	B26	RX_DATA [4]+	42	A20	TX_DATA [4]+
43	A26	RX_DATA [4]–	44	A21	TX_DATA [4]–
45	E26	RX_DATA [5]+	46	D21	TX_DATA [5]+
47	F26	RX_DATA [5]–	48	E21	TX_DATA [5]–
49	D27	RX_DATA [6]+	50	B21	TX_DATA [6]+
51	E27	RX_DATA [6]–	52	B22	TX_DATA [6]–
53	B31	RX_CLK [1]+	54	F21	TX_CLK [1]+
55	A31	RX_CLK [1]–	56	G20	TX_CLK [1]–
57	B27	RX_DATA [7]+	58	F19	TX_DATA [7]+
59	C27	RX_DATA [7]–	60	F20	TX_DATA [7]–
61	C28	RX_DATA [8]+	62	G21	TX_DATA [8]+
63	B28	RX_DATA [8]–	64	G22	TX_DATA [8]–
65	A28	RX_DATA [9]+	66	E22	TX_DATA [9]+
67	A29	RX_DATA [9]–	68	E23	TX_DATA [9]–
69	C29	RX_DATA [10]+	70	C22	TX_DATA [10]+
71	D29	RX_DATA [10]–	72	D22	TX_DATA [10]–
73	C30	RX_DATA [11]+	74	B23	TX_DATA [11]+
75	D30	RX_DATA [11]–	76	C23	TX_DATA [11]–
77	A30	RX_DATA [12]+	78	A23	TX_DATA [12]+
79	B30	RX_DATA [12]–	80	A24	TX_DATA [12]–

Table 11. DRX16 FPGA to ADC

DRX16 FPGA Pin	ADC Channel 0 Function	DRX16 FPGA Pin	ADC Channel 1 Function
R33	Data in 0+	K32	Data in 0+
T30	Data in 1+	K34	Data in 1+
T33	Data in 2+	L33	Data in 2+
U33	Data in 3+	M31	Data in 3+
V32	Data in 4+	N33	Data in 4+
W31	Data in 5+	N32	Data in 5+
Y33	Data in 6+	N34	Data in 6+
Y32	Data in 7+	R31	Data in 7+
R34	Data in 0-	K31	Data in 0-
T31	Data in 1-	L34	Data in 1-
T34	Data in 2-	M32	Data in 2-
U32	Data in 3-	L31	Data in 3-
V33	Data in 4-	M33	Data in 4-
W32	Data in 5-	P32	Data in 5-
Y34	Data in 6-	P34	Data in 6-
Y31	Data in 7-	R32	Data in 7-
V34	Data clock +	N28	Data clock +
W34	Data clock -	N29	Data clock -
B33	Dither	A33	Dither

Table 12. DRX16 FPGA to Other Local Devices

Function	DRX16 FPGA Pin	Function	DRX16 FPGA Pin
PGA870 – Channel 0		***PGA870 – Channel 1***	
Gain	D32	Gain	F30
Latch	D31	Latch	G30
B5	F31	B5	H34
B4	F33	B4	H33
B3	F34	B3	H32
B2	E31	B2	G33
B1	E32	B1	G32
B0	E33	B0	G31
Reference Clock		***LED***	
Reference clock into FPGA	K24	LED 1	C32
External or internal reference clock selector	C33	LED 0	B32
Board Control		***Sample Clock***	
Enable (channel 0)	B34	Loss of 10 MHz signal	J34
Enable (channel 1)	C34	Loss of external sample clock	J32
Module control (channel 0)	D34	SPI serial data out	J31
Module control (channel 1)	E34	SPI clock	J29
		SPI serial data in	J27
Time Code		SPI select	J30
SPI serial data out to MSP430	L24	Reset signal	H30
SPI serial data in from MSP430	F24	Loss of lock	J26
SPI serial data clock to MSP430	F23		
SPI serial enable to MSP430	G23		
One-pulse-per-second input to FPGA	J25		
IRIG-B comparator input to FPGA	H25		

Revision Log

Below is a history of additions, deletions, corrections, revisions, and rewrites / restructures to this guide.

Date / By	Initiator/s	Pp.	Content	Pp.	Cosmetic	Posted
20100331 PH	B. O'Neal / PH	All	Created (based on SRXL2).			