DS100DF410EVK, DS110DF410EVK and DS125DF410EVM Evaluation Board Software Installation, Setup, and Operating Guide

User's Guide

Literature Number: SNLU126A
February 2013–Revised May 2013

DS100DF410EVK, DS110DF410EVK and DS125DF410EVM Evaluation Board Software Installation, Setup, and Operating Guide

The DS100DF410EVK, DS110DF410EVK and DS125DF410EVM evaluation board allows the user to examine the advanced signal conditioning capabilities of the quad retimer products using SMA connectors for the high speed signal. The board connects to a PC using a USB port and the Analog LauchPAD software GUI interface is used to control the device.

All references to the DS110DF410EVK in the document should be taken or apply to the device installed on the evaluation board. The document applies to all the following devices that can be installed on the board: DS100DF410, DS110DF410, DS125DF410.

Topic **Page**

1	**Features**	3
2	**Software Installation and Configuration**	5
3	**Running ALP and Configuring the Device Registers**	7
4	**EEPROM and Register Map Informations**	34
5	**Bill of Materials:**	51
6	**Schematic:**	53
7	**Board Layout:**	56

1 Features

- Each channel independently locks to 10.3125 Gbps (DS100DF410), 8.5 to 11.3 Gbps (DS110DF410) , 9.8 - 12.5 Gbps (DS125DF410) and sub-multiples of the data rates
- Lock time operation (typically under 15 ms)
- Low latency (~300 ps)
- Adaptive equalization up to 34 dB boost at 5 Gbps GHz
- Adjustable transmit VOD : 600 to 1300 mVp-p
- Adjustable transmit de-emphasis to -12 dB
- Typical Power Dissipation (EQ+DFE+CDR+DE): 180 mW / channel
- Programmable output polarity inversion
- Input signal detection, CDR lock detection/indicator
- On-chip Eye Monitor (EOM), PRBS generator
- Single 2.5 V ±5% power supply
- SMBus/EEPROM configuration modes
- Operating temperature range of -40 to 85°C
- RHS 48-pin, 7 mm x 7 mm package

1.1 Applications

Front port SFF 8431 (SFP+) optical and direct attach copper

Backplane reach extension, data retimer

Ethernet: 10GbE, 1GbE

Fibre-Channel, Infiniband and other protocols supports

CPRI: Line bit rate options 3–7

Interlaken: All lane bit rates

1.2 Ordering Information

EVM ID	DEVICE ID	DEVICE PACKAGE	PACKAGE TYPE
DS100DF410EVK/NOPB	DS100DF410SQ/NOPB	RHS-48	QFN
DS110DF410EVK/NOPB	DS110DF410SQ/NOPB	RHS-48	QFN
DS125DF410EVM	DS125DF410SQ/NOPB	RHS-48	QFN

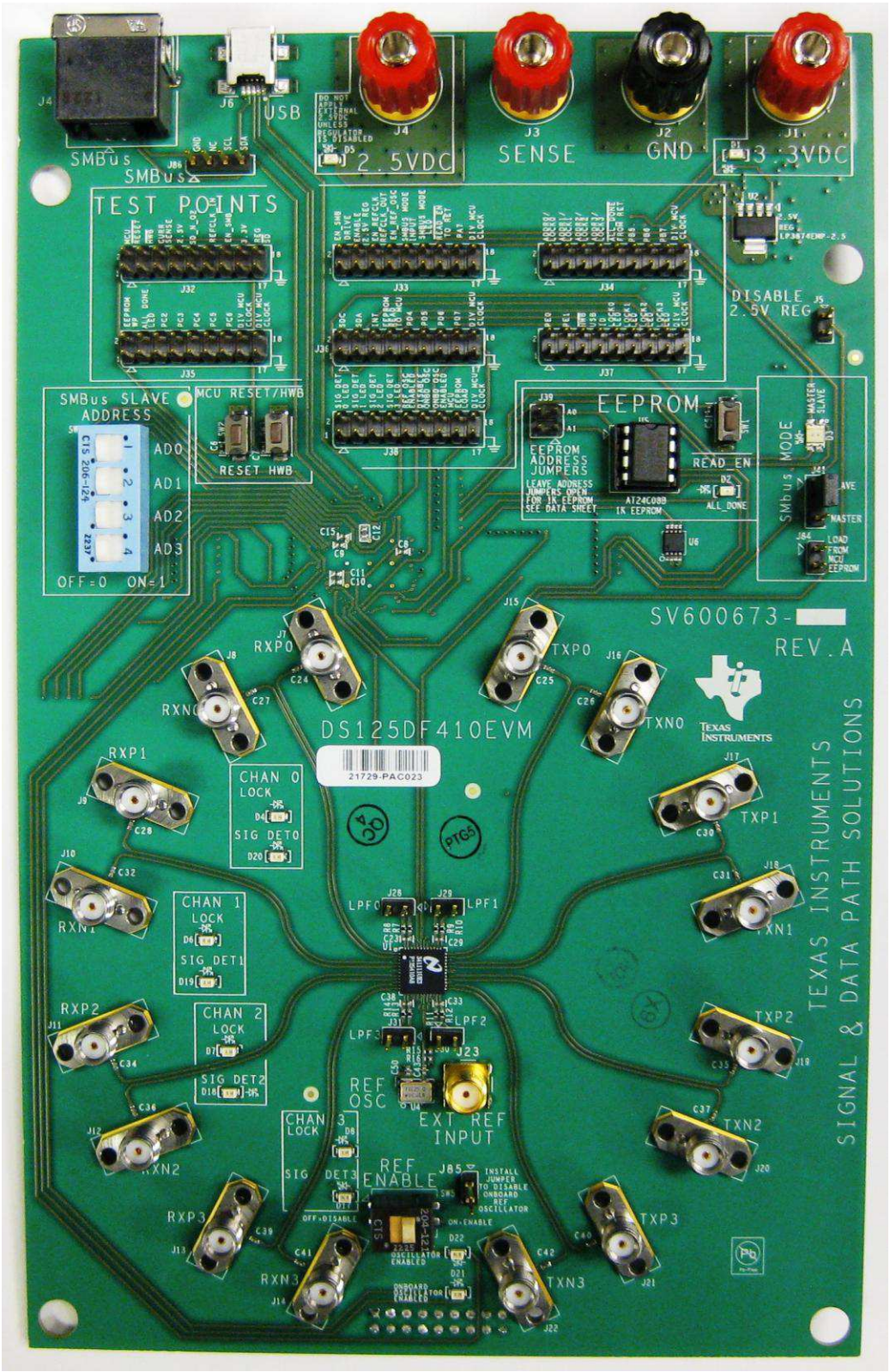

Figure 1. DS100DF410EVK, DS110DF410EVK, DS125DF410EVM Top View

2 Software Installation and Configuration

The Analog LaunchPAD software may be revised to add new features and to correct bugs. This document applies to all Analog LaunchPAD software versions 1.28.1021 and later. Wherever the revision number occurs in this document you may take it to refer to the version of the software you are installing and using, as long as that version is 1.28.1021 or later.

Texas Instruments Analog LaunchPAD (ALP) software includes a device profile for the DS110DF410EVK. This profile is designed to control the DS110DF410EVK retimer evaluation board, which includes an integrated microcontroller.

There are several steps for preparing the ALP software for first use.

1. Install the ALP software using the setup.exe program provided.
2. Connect the DS110DF410EVK retimer evaluation board to a USB port on the computer and install the driver for the ALP Nano (the integrated microcontroller on the board).
3. Connect the signal cables in the system.
4. Run ALP software and configure the DS110DF410 retimer as required.

All the screen images in this document were taken from a computer running Windows XP. The driver for the ALP Nano board does not work under versions of Windows later than XP including Windows Vista and Windows 7, because the driver model in Windows is different for these operating systems. Texas Instruments recommends that you install the software on a PC running Windows XP.

If the PC only have Windows 7, then there is an option to install Windows XP Mode which will have a virtual PC with Windows XP.

1. To do this, the first step is to goto the website: http://www.microsoft.com/windows/virtual-pc/download.aspx and download and install Microsoft Windows XP mode for Windows 7.
2. Follow all of the installation steps listed on Microsoft's website. The download file is 500 Mb, so it may take from 15 to 30 mins to complete.
3. After the installation is completed, may need to restart the PC. Launch Microsoft XP Mode by clicking on Start > All Programs > Windows Virtual PC > Windows XP Mode.
4. Once Windows XP virtual mode is running, install the ALP software by running the file: setup.exe.

2.1 Installing the ALP Software

First install the ALP software using the setup.exe program provided. The setup.exe program may be run from the CD provided or downloaded onto your computer. If you copy it, remember to put it in a location where it can be found again, since many software installation programs are named setup.exe.

To run the setup program, just double click on it.

The Welcome screen for the setup.exe program for Analog LaunchPAD v 1.29.0009 first screen will appear first. The current version number is 1.29.0009.

Click Next and the software license agreement screen will appear.

Click the radio button labeled "I accept the agreement" and then click Next.

The installation destination screen will appear next. Accept the default location which is in your Program Files directory in the National Semiconductor Corp subdirectory. A new folder will be created for this new version of Analog LaunchPAD, but the older versions will not be deleted. The current default destination directory under the National Semiconductor Corp subdirectory is Analog LaunchPAD v1.29.0009.

Click Next and the start menu folder selection screen will appear next. Again, accept the default location, which is in a program group called National Semiconductor Corp.

The additional tasks window will appear, Select the "Create a desktop icon" options and click Next. The ready to install screen will appear, if the selected options are okay, click Install. The installation will start. It usually takes less than a minute and there will be continuous progress indications. When the installation is complete you will see the completion screen.

Don't start Analog LaunchPAD (ALP) yet. We need to be sure that the ALP Nano drivers are installed. Uncheck the check box and click Finish. If the installation created a shortcut on your desktop, you can start ALP later by double-clicking the shortcut. If not, ALP may be launched from the start menu by clicking Programs and finding the correct program group.

2.2 Connecting the DS110DF410EVK Board and Installing the Driver

The basic operation is to install a driver for the user program (NSC ALP Nano Atmel) and, optionally, one for the bootloader on the microcontroller (AT90USB1287) integrated onto the DS110DF410EVK board. It is necessary to be sure that the correct program is running on the microcontroller when the drivers are installed in order to associate the correct driver with each program on the microcontroller. This section will describe the sequence for installing the NSC ALP Nano Atmel driver.

2.2.1 Installing and checking the ALP Nano driver.

You will be plugging the DS110DF410EVK board USB connector is at the top left of the board picture label as J6 into your computer's USB port. It is usually best to use a USB port that connects directly to the motherboard of your computer. Introducing additional USB hubs between the motherboard and the DS110DF410EVK board may cause problems with the USB driver.

The DS110DF410EVK board requires an external 3.3V power supply. The supply terminals are banana jack binding posts. These are at the top left corner of the DS110DF10EVK board. Notice that there are four power supply terminals. In normal operation, only the 3.3V DC supply need be connected, between J1 (3.3V DC) and J2 (ground). To power up the board, connect a 3.3V DC power supply to the power supply terminals. An onboard 2.5V DC regulator supplies the 2.5V DC power to the DS110DF410 retimer. This regulator is under the control of the onboard microcontroller and will be activated by the microcontroller when it is correctly programmed. The board is shipped with the microcontroller already programmed, so the 2.5V DC indicator light should come on when the board is first powered up. If the 2.5V DC LED is flashing, or is not illuminated, the power supply voltage or supply clamping current may be set too low. Try increasing the power supply voltage to 3.4V DC. The microcontroller can tolerate supply voltages up to 5V DC. If the power supply is current limiting, increase the power supply current limit setting. In normal operation, with all channels active, the DS110DF410EVK board will draw about 500 mA from a 3.3V DC supply. Make sure the power supply is set to supply at least this much current. A supply current limit setting of at 750 mA is recommended. If the 2.5V DC LED is flashing, the microcontroller may not respond properly to USB requests. This will prevent you from installing the driver, so if this occurs correct it as described above before proceeding.

A jumper is required on the SMBus Mode header (J41) in the Slave position for proper operation. If this jumper is not installed, the microcontroller may not respond properly to USB requests, which will prevent you from installing the driver. If the header is installed, the SMBus mode indicator LED should light up green. If the header is not installed, or is installed in master mode, the SMBus mode indicator will light up red. Master mode is not currently implemented on this board, so the jumper must be set for Slave mode for proper operation. If the jumper is not set correctly, set it for Slave mode and insure that the SMBus mode LED lights up green before proceeding.

Once the DS110DF410EVK board is properly powered up with the USB connected to the PC, and the Slave mode jumper is installed, Windows should prompt you to install a driver for the NSC ALP Nano Atmel. If you get an "Unknown Device" prompt you may need to uninstall the current driver and reinstall it. If you get the prompt to install a driver for the NSC ALP Nano Atmel, install the driver from the Analog LaunchPAD "driver" directory. If it ask for windows update to search for the software dirver, click the radio button next to "No, not this time" and then click Next. Click the radio button next to "Install from a list or specific location (Advanced)" and then click Next. The default setting is "Search for the best driver in these locations". Click the check box next to "Include this location in the search:". Browse to the Analog LaunchPAD v1.29.0009\Drivers directory and click Next. After the drivers have been installed once, they do not need to be installed again, even if a new version of Analog LaunchPAD is installed. If there is a warning that says the driver has not passed the windows logo testing for compatibility with windows, click Continue Anyway. The driver will install and you will see a screen that shows the driver installation complete screen for installation of the ALP Nano driver, click Finish. It is possible that the installation will

prompt you for another file, NSCNanoUSB.sys. If it does, this file is in the same Drivers directory mentioned above. Tell the installation program to search in this directory and it should find the file. After clicking Finish, go to the start menu and click Control Panel, then System, then Hardware, then Hardware Manager to see that there is a "NSC ALP Nano Atmel" device in the Universal Serial Bus Controllers node. When this is done, you have a driver for the user program, NSC ALP Nano Atmel.

2.3 Connecting the Signal Cables

The DS110DF410EVK board features four pairs of input and four pairs of output SMA connectors. The SMA connectors are vertical connectors which are held to the board by screws inserted from the back side of the board. As a result, the SMA connectors on the DS110DF410EVK board are comparatively robust. Nevertheless, SMA connectors are precision microwave connectors, and they can be damaged by improper connections. Use caution when connecting the signal cables to the boards as the SMA connectors can be broken loose if excessive torque is applied. Use a torque wrench and do not torque the connectors to more than 7-10 inch-pounds (the recommended torque for SMA connectors).

The connectors are arranged in pairs and are labeled, for example, RXP0 and RXN0 for the positive and negative input connectors for channel 0. The retimed output for the data stream presented at connectors RXP0 and RXN0, for example, will be output on connectors TXP0 and TXN0. For Channel 1, the inputs are RXP1 and RXN1, the outputs are TXP1 and TXN1. For Channel 2, the inputs are RXP2 and RXN2, the outputs are TXP2 and TXN2. For Channel 3, the inputs are RXP3 and RXN3, the outputs are TXP3 and TXN3.

3 Running ALP and Configuring the Device Registers

When the driver setup is complete, the DS110DF410EVK board has been powered on, and the DS110DF410EVK board has been connected to the host computer by a USB cable, you can start Analog LaunchPAD by double-clicking its icon on the desktop or by selecting it from the appropriate program group. Analog LaunchPAD will recognize the DS110DF410EVK profile and will start the appropriate GUI. When Analog LaunchPAD starts it will display a splash screen which includes the Analog LaunchPAD version number, followed by the main screen for Analog LaunchPAD. Note that at the left side of the screen, under Devices, an ALP Nano USB device is shown. Under this device, which represents the on-board microcontroller running the correct firmware, is the DS110DF410EVK device.

Click on DS110DF410EVK and the Graphical User Interface (GUI) will start. The GUI consists of eight different notebook tabs, each of which controls a different feature of the DS110DF410 retimer. The main tabbed notebook which appears just after the DS110DF410EVK label has been clicked.

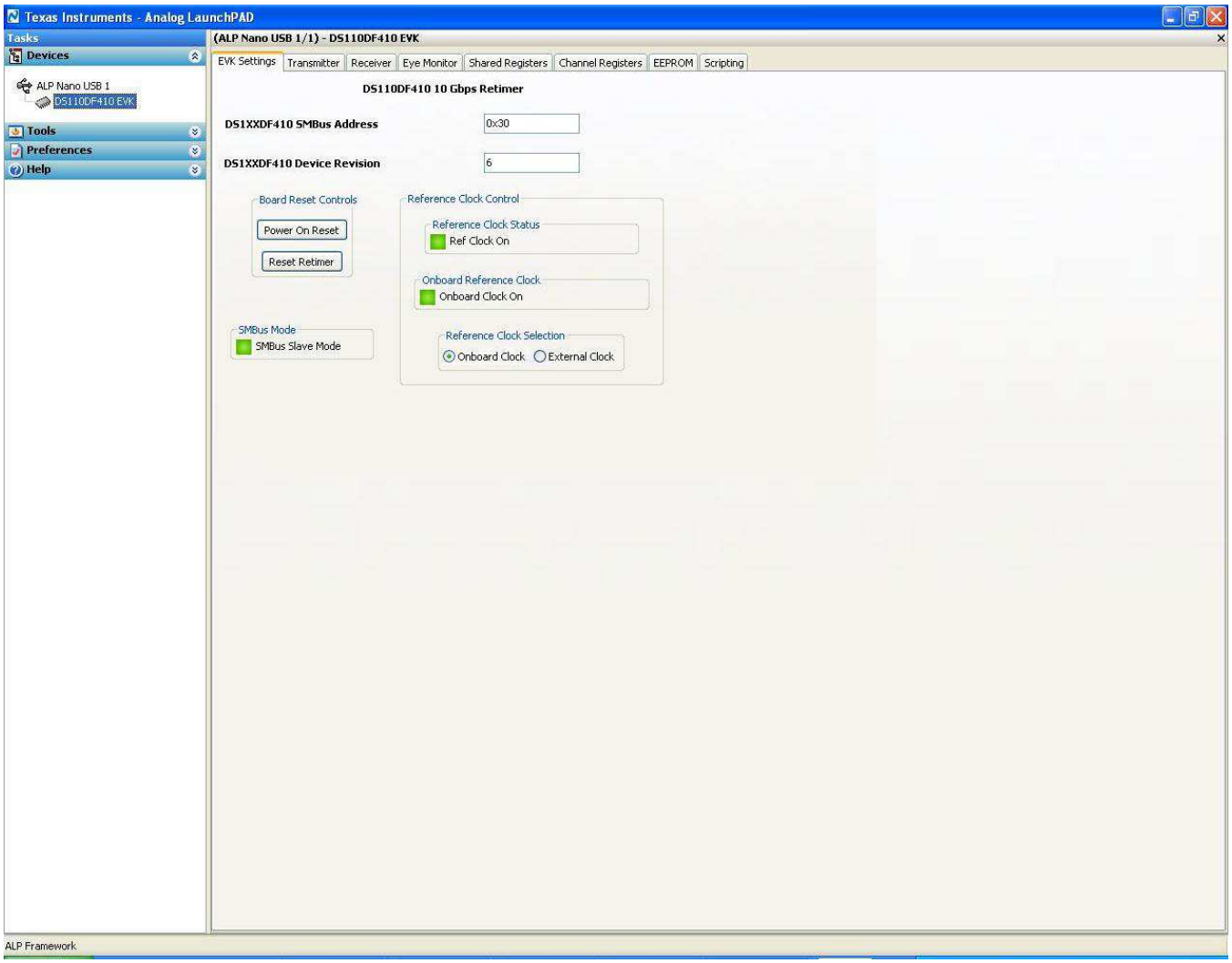

Figure 2. EVK Settings Tab Screen

3.1 EVK Settings Tab

The EVK Settings tab is visible when the GUI starts is the EVK Settings tab. The EVK Settings tab contains information that applies to the DS110DF410EVK itself, and to the overall operation of the DS110DF410 retimer. At the top of this screen is a text field showing the DS110DF410 SMBus address. This address is set by the switches on the DS110DF410EVK board. Every setting of the SMBus address setting switches corresponds to a valid SMBus address for the DS110DF410 retimer. The default address, with all the switches off, is 0x30. This is the SMBus Write address for the DS110DF410 retimer.

When the DS110DF410 retimer powers up, it reads the value on its SMBus address lines. These lines are later re-used as the lock indicator outputs of the DS110DF410 retimer. The microcontroller on the board controls the 2.5V DC onboard regulator, so it controls when the DS110DF410 powers up. Since it controls the power on sequence, it can also read the SMBus address switches. It records the SMBus address setting when the retimer powers up and reports it to the GUI. The DS110DF410 SMBus address cannot be set from the GUI. This control is read-only.

If a different SMBus address is desired, set the switches (SW4) on the board to the desired setting and press the RESET button. This causes the microcontroller to start the DS110DF410 up again and re-read the SMBus address straps. The SMBus address switches set the SMBus Write address for the DS110DF410 according to Table 1. The address that will be displayed on the EVK Settings tab is the SMBus write address. The microcontroller will automatically address the DS110DF410 retimer using its configured write address and read address as necessary, which is transparent to the user.

Table 1. SMBus address switch settings and DS110DF410 SMBus addresses.

AD3	AD2	AD2	AD0	DS110DF410 Write Address (Hex)	DS110DF410 Read Address (Hex)
0	0	0	0	0x30	0x31
0	0	0	1	0x32	0x33
0	0	1	0	0x34	0x35
0	0	1	1	0x36	0x37
0	1	0	0	0x38	0x39
0	1	0	1	0x3A	0x3B
0	1	1	0	0x3C	0x3D
0	1	1	1	0x3E	0x3F
1	0	0	0	0x40	0x41
1	0	0	1	0x42	0x43
1	0	1	0	0x44	0x45
1	0	1	1	0x46	0x47
1	1	0	0	0x48	0x49
1	1	0	1	0x4A	0x4B
1	1	1	0	0x4C	0x4D
1	1	1	1	0x4E	0x4F

There is normally no reason to change the factory default SMBus write address of 0x30. However, if it is desired to change this address in order to test the device operation with some other SMBus write address, the DS110DF410 retimer on the DS110DF410EVK board can be configured to any address using the address switches and the DS110DF410EVK board will detect the SMBus address properly.

The second line on the EVK Settings screen is a text field showing the DS110DF410 device revision. This is read from the DS110DF410 on the board and displayed on the EVK Settings screen. Since the DS110DF410 is permanently mounted to the DS110DF410EVK board, this revision will never change. This control is read-only.

A control group on this screen labeled Board Reset Control permits the user to perform two different types of resets of the retimer. The Power On Reset button in this control group can perform a power on reset of the retimer, which momentarily turns off the 2.5V DC internal regulator. This resets the DS110DF410 retimer. The microcontroller can be configured to automatically reload a set of non-default register settings into the DS110DF410 retimer on power up, or to leave the DS110DF410 retimer in its factory default state. This operation will be described in a later section of this document.

The Reset Retimer button in the Board Reset Control control group performs a different type of reset on the DS110DF410 retimer. It resets all registers back to their factory defaults and resets all the state machines and CDRs for all channels. This type of reset can be performed if desired to reset the DS110DF410 back to its factory settings.

Note that both of these controls reset only the DS110DF410 retimer on the DS110DF410EVK board. They do not reset the microcontroller, so the Analog LaunchPAD GUI remains active. On the other hand, the RESET switch on the DS110DF410EVK resets the microcontroller as well as the DS110DF410 retimer. This requires the Analog LaunchPAD GUI to restart. If you press the RESET switch on the board while Analog LaunchPAD is active, the GUI will return to the main screen and you will have to restart the GUI again.

The next control group on the EVK Settings panel is the Reference Clock Control group. This group consists of two indicators and a radio box.The first indicator, labeled Reference Clock Status indicates whether the reference clock on the DS110DF410EVK board was enabled when the retimer was started. The reference clock is enabled using the REF ENABLE switch on the board.

The reference clock status is read by the DS110DF410 when it is powered on, at which time the microcontroller also reads the status. This is the status reported on the Analog LaunchPAD EVK Settings tab. If the reference clock switch is operated after the DS110DF410 has been powered up, the reference clock status will not change until the DS110DF410 is reset by a power on reset. When this occurs, the indicators will show that the reference clock is active.

The onboard reference oscillator may be disabled by clicking the radio button labeled External Clock in the Reference Clock Oscillator control group. This is useful if you want to use and external reference. The J23 is a SMA connector for this external reference clock. If Analog LaunchPAD is not running, the onboard reference clock can still be disabled by installing a jumper to ground on the header pin labeled as J85. When a jumper is installed, the onboard reference oscillator is disabled regardless of the state of the reference oscillator enable switch. If the reference oscillator enable switch is turned on, the DS110DF410 will expect and will use a 25 MHz reference signal. The 25 MHz signal should be supplied by an external source to J23 if this jumper is installed.

The EVK Settings screen of the GUI also includes an indicator for Slave mode. As noted above, Master mode is not currently functional on the DS110DF410EVK board, so the LED indicator should always be green. The indicator reads the state of the SMBus mode jumper on the board.

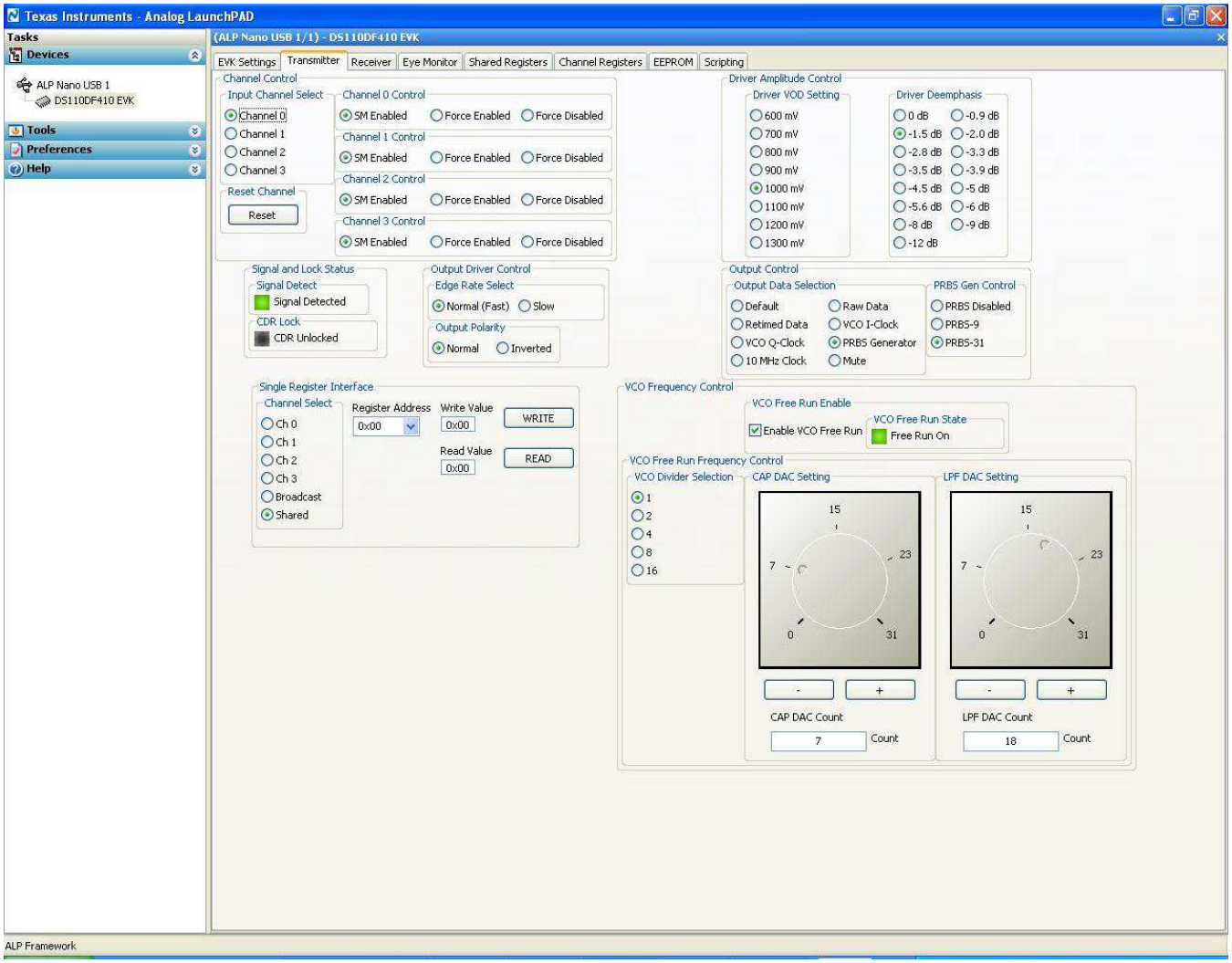

Figure 3. Transmitter Tab Screen

3.2 Transmitter Tab

The Transmitter tab includes controls and indicators that relate to the output driver functions of the DS110DF410 retimer. Note that this screen can be bigger than the monitor used here, so horizontal and vertical scroll bars are displayed.

At the top left corner of the screen is the Channel Control group. This includes a radio box labeled Input Channel Select. The DS110DF410 retimer has four independent signal channels. The channel selected in this radio box is the one which will be controlled by the other controls on this panel. There are channel selection controls on other panels, all of which operate the same way as the one on the Transmitter tab. The channel selection controls on each panel are independent. A different channel can be selected on this panel than the one selected on other panels. If you navigate away from the Transmitter panel to another panel, and then return to the Transmitter panel, the previous channel selection on the Transmitter panel will be retained.

Changes made to the settings on a channel using the controls on this panel are retained until the DS110DF410 retimer is reset or powered on and off. When the Input Channel Select radio button selection is changed, the controls on this panel will be updated to reflect the register settings for the newly selected channel. Any settings in the DS110DF410 retimer that have been changed will be retained, and will be reflected again when the original channel radio button is again selected.

The Channel Control group also includes a button labeled Reset Channel. When this button is pressed, the Clock/Data Recovery state machine (CDR) in the DS110DF410 retimer is reset. Any register settings that have changed will be retained, but the CDR will be reset and will go through its data acquisition sequence. This is usually not necessary for the controls on the Transmitter tab. If the single-register interface, described later, is used to change a register setting that requires a CDR reset (such as the equalizer adaptation mode), then the Reset Channel button on the Transmitter tab may be used to reset the CDR. The Reset Channel button applies only to the channel selected in the Input Channel Select radio box. If a complete reset of the DS110DF410 retimer is required, use one of the reset controls on the EVK Settings tab.

Finally, the Channel Control group includes radio boxes that control the enable state of all four channels. The default setting is SM Enabled. When this radio button is selected for a particular channel, that channel will be enabled under state machine control. In general, this means that the channel will be enabled when a valid signal is present at its inputs, and disabled when no such signal is present.

Each channel may also be force enabled using the Force Enabled radio buttons in the Channel Control group. When a channel is force enabled, its signal detect register is overridden. This causes the state machine to enable the channel. The signal detect light will turn green for this channel even though no signal is really present. When a channel is force enabled, the power supply current will increase since the circuitry associated with this channel now becomes active.

The controls in the Channel Control group may also be used to force disable a channel even when a valid signal is present at its input. Note that in this case the signal detect and lock detect indicators are off (black) even though a valid signal is present at the input to the channel. When the channel is force disabled, the power supply current decreases because the circuitry associated with that channel is powered off. The output for that channel is also muted. The channel can be returned to normal operation by selecting the radio button labeled SM Enabled.

The signal and lock detect status indicators reflect the status of the signal at the input to the channel selected in the Input Channel Select radio box. The Signal Detect indicator will turn green when a signal is present at the input with sufficient amplitude to activate the signal detect circuitry in the DS110DF410. The CDR Lock indicator will turn green when the CDR state machine detects phase lock to the input signal.

The DS110DF410 retimer will usually lock automatically to any input signal in its lock range. When the DS110DF410 retimer is locked, the lock indicator on the board will also turn green. The lock indicators LED are located near the input connectors for each channel.

The signal detect can be monitored with an oscilloscope or logic analyzer on pins PF0-PF3 of header J38. Pins PF0-PF3 reflect the signal detect state of channels 0-3 respectively. The voltage at each pin goes high when a valid signal is detected on the corresponding channel. The signal detect monitor pins are only active when the DS110DF410EVK Analog LaunchPAD GUI is running.

The Output Driver Control group contains two radio boxes. The first is labeled Edge Rate Select. By selecting the Slow option in this radio box, the edge rate for driver of the selected channel will be reduced. This can be useful to reduce EMI and crosstalk where edge rate is not critical. Normally, the output driver should always be operated in Normal (Fast) mode unless there is a system-related reason for slowing down the edge rate.

The second control in this group is the Output Polarity control. Many systems using differential transmission do not include the capability to adjust to an accidental polarity inversion. The DS110DF410 retimer can invert the polarity of the signals passing through it. This can correct for connection errors in the system. Normally the DS110DF410 should be operated in the Normal polarity mode unless there is a system-related reason for inverting the polarity.

The Driver Amplitude Control group is at the right hand top of the Transmitter panel. There are two radio boxes in this control group. The first is the Driver VOD Setting radio box. This radio box allows the selection of the nominal output differential voltage for the selected channel. The actual observed output voltage is also affected by the setting in the second radio box, labeled Driver De-emphasis. This selects the amplitude of the de-emphasis applied to the output signal for the selected channel. Where there is a long, lossy channel after the DS110DF410 retimer (for example, a long cable), increasing the de-emphasis setting provides an optimized waveform for transmission through the lossy media channel. The settings in the “Driver VOD Setting” and “Driver Deemphasis” radio boxes apply to the currently selected channel only. They are retained until the DS110DF410 retimer is fully reset, either by a power on reset or a device reset using the controls on the EVK Settings tab.

Just beneath the Driver Amplitude Control group is the Output Control group. This group includes radio boxes labeled Output Data Selection and PRBS Gen Control. The Output Data Selection radio box sets the output data multiplexer for the currently selected channel. The settings in the radio box and the output they select are shown in Table 2.

Table 2. Output Data Selected by the Output Data Selection Radio Buttons

Output Data Selection Radio Button	Output Data Selected
Default	Data output is selected automatically by the DS110DF410 retimer state machine. When the DS110DF410 retimer is locked, the output is the retimed data. When the DS110DF410 is not locked, the output is muted.
Retimed Data	Data output is the output of the CDR. In normal operation, this is the same as selecting Default.
VCO Q-Clock	Quadrature phase clock output of the VCO for the selected channel. If the DS110DF410 is locked on the selected channel, the quadrature phase clock will be synchronous with the incoming data, but 90° out of phase.
10 MHz Clock	Internal 10 MHz clock. This clock is not synchronous to any input. It is used to clock the internal digital circuitry. Its frequency is nominally 10 MHz, but the frequency accuracy is not specified.
Raw Data	Data output is the output of the equalizers, sampled after both the Continuous Time Linear Equalizer (CTLE) and Decision Feedback Equalizer (DFE). This data output is sampled before the CDR, but it is driven by the output driver circuitry. This data output is useful for evaluating the performance of the equalizers.
VCO I-Clock	In-phase clock output of the VCO for the selected channel. If the DS110DF410 is locked on the selected channel, the in-phase clock will be synchronous with the incoming data.
PRBS Generator	Data output of the internal PRBS signal generator. If the DS110DF410 is locked on the selected channel, the PRBS generator output will be synchronous with the incoming data.
Mute	Zero differential voltage output. The output is off.

The PRBS Gen Control radio box controls the digital portion of the internal PRBS signal generator. The PRBS signal generator can be disabled, which is the default, or it can be set to generate either a PRBS-9 or PRBS-31 pattern. This pattern will be independent of the data input to the selected channel, but it will be synchronous to it if the CDR is locked. In other words, if PRBS-31 is selected, and if the Output Data Selection radio box control for PRBS Generator is selected, then the output data stream for the selected channel will be a standard PRBS-31 pattern no matter what the input data stream is. The output data stream will, however, by synchronous with the input data stream.

In order for the DS110DF410 retimer to output a PRBS pattern, it is necessary both to enable the PRBS generator using the controls in the PRBS Gen Control radio box, and to select the output of the PRBS generator using the controls in the Output Data Selection radio box.

It is also possible for the DS110DF410 to produce a free-running clock or a PRBS signal that is not synchronous to any signal at its outputs. This is enabled using the controls in the VCO Frequency Control group. Note that the VCO frequency control is disabled by default. When the Enable VCO Free Run check box is unchecked, as shown, the VCO frequency is controlled by the CDR. The output of the DS110DF410, if any, is synchronous with the input signal on the selected channel when the Enable VCO Free Run check box is unchecked.

To set the VCO for free-running operation, first check the Enable VCO Free Run check box. Note that the CAP DAC Setting and LPF DAC Setting knobs and the VCO Divider Selection radio box will now be enabled. Also the VCO Free Run State indicator is now green, indicating that VCO free-run mode is active.

The VCO Divider Selection control indicates that the current divide ratio is 16, which is the default. In order to output a 10.3 Gbps signal, as an example, the divide ratio must be set to 1. When this is set, the output signal is clocked from the fundamental frequency of the VCO.

The frequency of the VCO in free-running mode is set by the CAP DAC Setting and LPF DAC Setting knobs. The CAP DAC Setting knob selects the coarse VCO range by tuning the VCO tank circuit in a stepwise fashion, switching additional capacitors in or out of the circuit. The LPF DAC Setting knob sets the VCO control voltage. The combination of the values of these two controls sets the output frequency of the VCO when it is in free-running mode. The setting of the VCO Divider Selection control sets the ratio of the VCO frequency to the output frequency in free-running mode.

The exact output frequency will vary from part to part and even from channel to channel within the same part. The output frequency is not locked to any precise reference, so it is dependent upon the exact values of the components inside the retimer VCO, on the temperature, on the supply voltage, and on other factors. In general, a CAP DAC count of about 9 and an LPF DAC count of about 19, with a divider ratio of 1, will produce a VCO frequency of approximately 10.3 GHz. As noted, however, this will vary from device to device and even from channel to channel. To determine the VCO output frequency, set the Output Data Selection to VCO I-Clock and look at the output frequency on a spectrum analyzer to measure the VCO I-Clock output.

Since the output in this mode is not locked to a stable reference, the output frequency is not very stable. The output amplitude is measured at -22 dBm, which corresponds to about 50 mV peak-to-peak. The output driver of the device is designed for 10 Gbps signals, which means that it is optimized for half that frequency, or about 5 GHz. The fact that the amplitude is much lower than the 600 mV setting for the output driver is due to (1) the fact that the output frequency is much higher than the design frequency and (2) the fact that the output driver setting is for a differential output, but the measurement is made on only one of the two single-ended outputs making up the differential output. When the output amplitude is set to its maximum, 1.3 V differential nominal, the output power of the VCO I-Clock is measured at about -7 dBm, or 280 mV peak-to-peak, single ended. This will vary from part to part and from channel to channel.

Note that when the DS110DF10 VCO is set to free-running mode, the CDR will not lock to the incoming signal. The lock light on the board will not illuminate and the CDR Lock indicator on the Transmitter panel will indicate that the CDR is unlocked. This is because the VCO is free-running and is not phase locked to the incoming signal. If you uncheck the Enable VCO Free Run check box, the CDR will lock to the incoming signal again.

When the VCO is set to free-running mode, several registers in the DS110DF410 are set to non-default values. All these register settings are required in order to make the VCO free run, and it is possible to overwrite some of the required register settings, leaving the GUI in an unknown state. If this occurs, the VCO Free Run State indicators will turn red, indicating a bad VCO free run state. To get the VCO out of this state, just click the Enable VCO Free Run check box one or more times. This will put the VCO back into a known good state, either with free-run enabled or disabled.

The other control on the Transmitter panel is the Single Register Interface control. This control allows you to read or write a single register in the DS110DF410 retimer. The control includes a radio box labeled Channel Select. This radio box selects either the shared register set, which is the set of registers that controls the overall operation of the DS110DF410 retimer, each channel register set individually, or a broadcast setting which enables you to write to all channels at once. The contents of the register may be changed by entering the desired data in hex in the text field next to the Write button, and then clicking the Write button. Note that when the Broadcast register set is selected, the read operation is disabled because the channels may have different register contents. It does not make sense to read a register when the Broadcast register set is selected.

Writing a register with the Single Register Interface control is the operation that can cause the VCO free-run control to go to a bad state, so it is recommended that you not write a register using this control unless you are sure of what the effects will be. This control is duplicated on both the Receiver and Eye Monitor tabs, and it works the same on all three tabs. The channel register set selected in this control can be different from the selection in the Channel Control group.

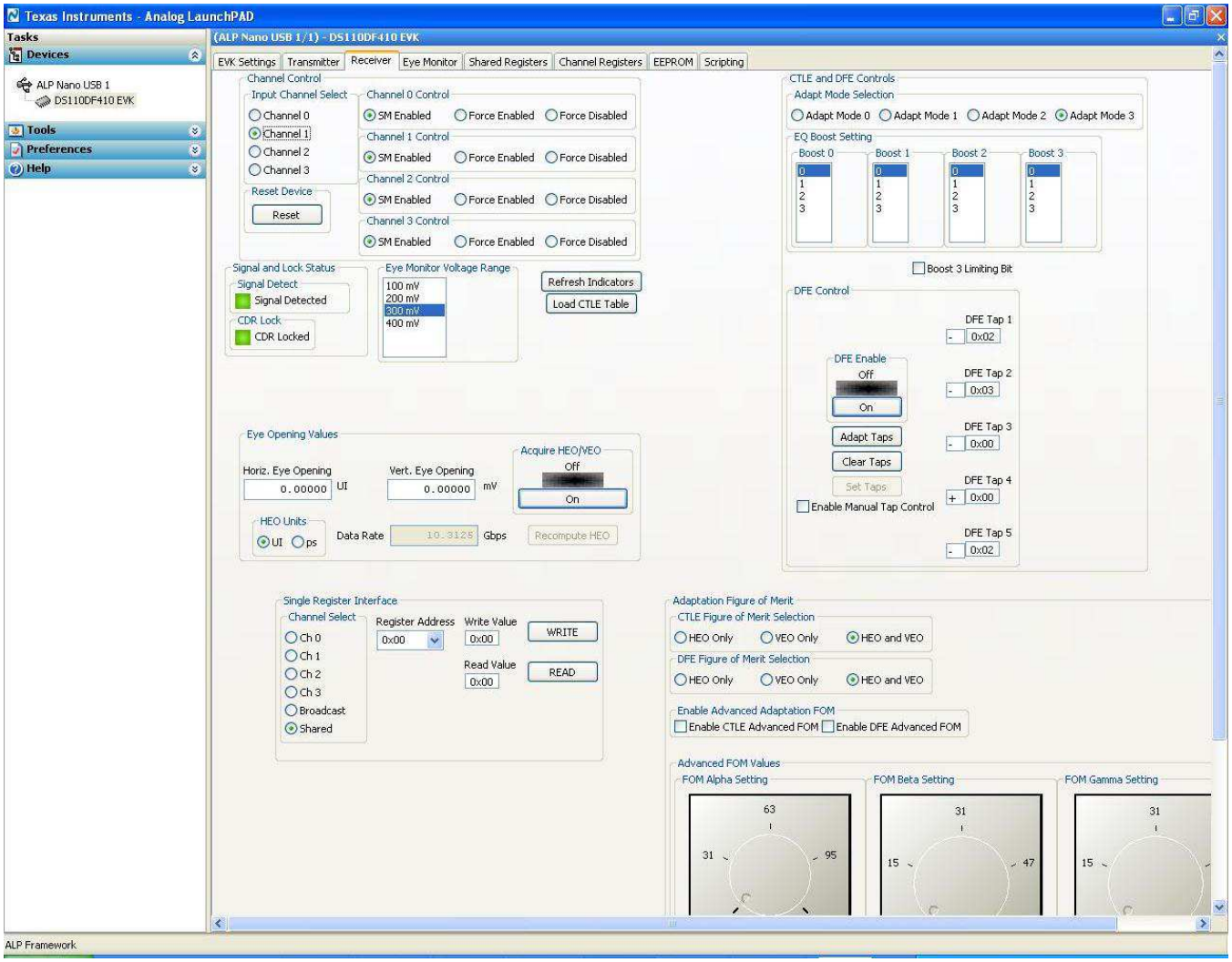

Figure 4. Receiver Tab Screen

3.3 Receiver Tab

The Receiver tab includes controls and indicators that relate to the input equalization functions of the DS110DF410 retimer. Note that this screen can be bigger than the monitor used here, so horizontal and vertical scroll bars are displayed.

At the top left corner of the screen is the Channel Control group. This control is identical to the one on the Transmitter screen and functions the same way. The channel selected in this radio box is the one which will be controlled by the other controls on this panel. The channel selection controls on each panel are independent. A different channel can be selected on this panel than the one selected on other panels. If you navigate away from the Receiver panel to another panel, and then return to the Receiver panel, the previous channel selection on the Receiver panel will be retained.

Changes made to the settings on a channel using the controls on this panel are retained until the DS110DF410 retimer is reset or powered on and off. When the Input Channel Select radio button selection is changed, the controls on this panel will be updated to reflect the register settings for the newly selected channel. Any settings in the DS110DF410 retimer that have been changed will be retained, and will be reflected again when the original channel radio button is again selected.

The signal and lock detect status indicators on the Receiver tab work exactly the same as those described previously for the Transmitter tab.

There is also a Single Register Interface control on the Receiver tab. This control works exactly the same way as the one described earlier on the Transmitter tab.

Just beneath the Channel Control group on the Receiver tab is the Eye Monitor Voltage Range control. This controls the voltage range for measurement using the internal eye monitor. The internal eye monitor provides a figure of merit for equalizer adaptation, so its voltage range setting can affect the equalizer performance. Generally, when the equalizers adapt to the incoming signal, the state machine sets the eye monitor voltage range to provide an optimum measurement. However, on the Receiver tab, you can set it if you like. This will affect the vertical eye opening as measured by the Eye Opening Values control on the Receiver tab.

At the top right of the Receiver tab is the CTLE and DFE Controls group. This control group controls the operation of the Continuous Time Linear Equalizer (CTLE) and the Decision Feedback Equalizer (DFE) in the DS110DF410 retimer. When the DS110DF410 detects a valid signal at its inputs, or when its CDR state machine is reset (that's what the Reset button on the Receiver tab does), it begins to try to lock to the incoming signal. Part of this lock process is automatic adaptation of the CTLE, the DFE, or both. The DS110DF410 retimer can also be set to adapt neither equalizer, but just to try to acquire lock using its phase-locked loop. The Adapt Mode Selection radio button at the top of this control group sets the adaptation mode the DS110DF410 will use in trying to lock to an incoming signal. There are four available adaptation modes, Adapt Mode 0 through 3.

When the DS110DF410 retimer is in Adapt Mode 0, it will not change its current CTLE and DFE settings as it tries to acquire phase lock to the incoming signal. The default settings for the CTLE boost registers and the DFE tap registers are all zero, so if the DS110DF410 retimer has been reset to its default state the equalizers will all be set to their minimum values. This mode is useful primarily for troubleshooting.

When the DS110DF410 retimer is in Adapt Mode 1, it will adapt the CTLE to an optimum value as it acquires lock. The optimum value is the value of the CTLE coefficients that (1) maximizes the figure of merit for adaptation and (2) is in the CTLE coefficients table. The figure of merit which is currently in use can be controlled by the user. This will be explained shortly. The CTLE settings that the DS110DF410 tries to use to maximize the figure of merit are contained in a CTLE coefficients table stored in a contiguous set of channel registers. These settings can be changed so that the DS110DF410 tries a different set of CTLE boost settings if desired. This is also explained in this section.

In Adapt Mode 1, the DFE is not used. The DFE coefficients will be left at the default value of 0. In Adapt Modes 2 and 3, both the CTLE and the DFE are used for adaptation.

In Adapt Mode 2, the CTLE is first adapted until an optimum eye opening is obtained with the DFE coefficients forced to 0. The DFE is then adapted and the DFE coefficients will change if a DFE setting that improves the eye opening is found. Finally the CTLE is adapted again with the new DFE settings, and the CTLE settings will change if a better eye opening can be found. This three step process tends to produce CTLE boost settings that are larger and DFE tap values that are smaller than does Adapt Mode 3.

In Adapt Mode 3, the sequence is almost the same. The difference is that instead of initially adapting the CTLE to obtain an optimum eye opening, the CTLE is adapted until the DS110DF410 retimer just declares phase lock. This may occur at a much lower CTLE boost setting than the optimum. Once phase lock is attained, the DFE is adapted to further optimize the eye opening, after which the CTLE is once again adapted with the new DFE values. In this adapt mode, the DFE tap values are generally greater in magnitude than for Adapt Mode 2 and the CTLE boost values are generally smaller. Adapt Mode 3 may provide superior performance in the presence of a large crosstalk interferer.

The EQ Boost Setting controls are below the Adapt Mode Select radio box. There are four stages of cascaded CTLE boost in the DS110DF410 retimer. The high-pass filter function of each stage is variable by the CTLE boost setting for that stage. Once the automatic adaptation is complete, the CTLE boost may be changed by clicking the desired settings in these list controls. You should note, however, that if a change to the CTLE boost causes the DS110DF410 retimer to drop out of lock, the CDR lock state machine will take over and will reset the CTLE boost settings to relock to the incoming signal (unless the DS110DF410 retimer is in Adapt Mode 0).

CTLE boost stage 0 is the first stage encountered by the signal, followed by stages 1, 2, and 3. In general, setting the CTLE so that more of the gain is in the first stage (stage 0) will reduce the noise propagated through the CTLE and will result in lower random jitter. This is not a universal rule, however, just a rule of thumb.

Also in general, the overall CTLE boost frequency response is more or less the same for all combinations of CTLE boost settings that sum up to the same value. For example, a CTLE boost setting, given as (Stage 0 boost, Stage 1 boost, Stage 2 boost, Stage 3 boost), of (2, 2, 0, 0) will produce a CTLE boost frequency response almost the same as a setting of (1, 1, 1, 1). Again, this is not a universal rule. There are slight differences in the frequency responses produced by such different boost settings. In general, however, you can determine comparatively how much CTLE boost is being applied by summing the boost settings of all four stages.

Below the EQ Boost Setting control group is a checkbox labeled Boost 3 Limiting Bit. The final boost stage, stage 3, can be set to be a limiting amplifier with relatively flat gain over frequency by checking this checkbox. For some channels this can provide improved performance, but generally it is better to leave this checkbox in its default, unchecked state.

Below this checkbox is the DFE Control group. The DS110DF410 retimer features a five-tap Decision Feedback Equalizer (DFE). The summing point for the DFE is after the CTLE and just before the comparator that decides whether the current bit is a one or a zero. The DFE Control group contains text controls that show the current values of the DFE taps. Tap 1 (the first tap, the tap that adds back to the current bit the previously-received bit delayed from the current bit by one bit time), has a magnitude range from 0x00 to 0x1F. The other taps each have a magnitude range from 0x00 to 0x0F. All taps can be subtracted at the summing point (sign is “-“) or added at the summing point (sign is “+”).

After adaptation, the text controls show the current values of the various DFE taps. The same controls can be used to manually set the DFE taps by first checking the checkbox labeled Enable Manual Tap Control, which is checked by default, and then entering the desired tap values (within the ranges described above) for each DFE tap. The tap values are applied when the button labeled Set Taps is clicked. The button labeled Clear Taps sets all the DFE tap values to 0x00.

The button labeled Adapt Taps will, when clicked, cause the DS110DF410 retimer to attempt to re-adapt the DFE tap values, starting from the current tap values, to find a better setting that optimizes the eye opening figure of merit. This operation works in all adapt modes. If a better set of DFE tap values is not found, the DFE tap values will not change. If, during adaptation, the DS110DF410 retimer loses lock, the CTLE values may be changed by the state machine in order to reacquire lock.

Finally, in this control group, there is a button labeled DFE Enable. This button will enable or disable the DFE. The DFE is enabled by default, but if the tap values are all 0x00, the default settings, it does not do anything. Clearing the taps and disabling the DFE will have the exact same effect on the waveform at the comparator summing junction.

Below the CTLE and DFE Controls group is a control group labeled Adaptation Figure of Merit. The controls in this group are used to fine tune the CTLE and DFE adaptation sequence for channels that are very lossy or for systems with large crosstalk interference components.

In all modes of operation, adaptive equalization, both of the CTLE and the DFE, operates by maximizing a defined figure of merit based upon measurements of the horizontal and vertical eye opening (HEO and VEO) at the CDR comparator summing point. The measurements are made by circuitry in the DS110DF410 retimer which samples the incoming data after the equalization, but before the CDR.

The default optimization algorithm maximizes both the HEO and the VEO. This is accomplished by taking the smaller of the two measured eye openings as a figure of merit and attempting to increase both in a stair-step fashion until no further improvement is possible. The first two radio boxes in this control group allow you to specify that optimization will be based on HEO alone, on VEO alone, or, the default setting, on both HEO and VEO. Whether this improves the performance of the adaptation is dependent upon the system and the channel. In general, however, a HEO-only figure of merit for adaptation may yield slightly better adaptation for very lossy channels or for systems with significant crosstalk interference.

The DS110DF410 retimer is, by design, almost immune to crosstalk. The channels within the device are independent and the design rules followed in the design of the DS110DF410 retimer insure that signals input on one channel do not propagate to the other channels. If there is crosstalk in the system, it occurs outside the DS110DF410 retimer: either in the connectors or in coupling between adjacent channels. The DFE in the DS110DF410 retimer can discriminate against this kind of crosstalk, but there are limits to the amount of crosstalk that can, even in principle, be corrected. For systems exhibiting high levels of crosstalk interference, some experimentation will be required to determine the optimum settings for the DS110DF410 retimer.

The two top radio boxes in this control group allow independent specification of the form of the figure of merit for the CTLE and the DFE. In high-crosstalk systems, the DFE figure of merit is the more important parameter.

If additional fine-tuning of the adaptation algorithm is desired, the Advanced Adaptation FOM may be used. This is enabled by the check boxes in the Enable Advanced Adaptation FOM control group. The CTLE and DFE advanced adaptation figures of merit may be enabled independently by the check boxes in this group, but a single set of figure of merit coefficients applies to both.

The Advanced Adaptation Figure of Merit is computed from the same HEO and VEO data as the standard adaptation figure of merit. The algorithm used, however, is different. For the Advanced Adaptation Figure of Merit, the figure of merit at each adaptation step is computed according to Equation (1) below.

$$FOM = (HEO-\beta)\times\alpha+(VEO-\gamma)\times(1-\alpha)$$

The range of measured HEO and VEO is 0 to 63, so this is also the range of β and γ. Clearly when one of these is set to a large value, it offsets the measurement of HEO and VEO. The HEO and VEO values used in the computation are unsigned, so the adjusted HEO and VEO values are always greater than or equal to zero. The value of α is scaled by 1/127 so that a value of 127 for α completely removes the effects of VEO and a value of 0 for α completely removes the effects of HEO.

The values for α, β, and γ are set by the knobs below the Enable Advanced Adaptation FOM check boxes.

The Advanced Adaptation Figure of Merit is designed as a last resort for systems with extremely lossy channels or very high levels of crosstalk. For such systems, trial-and-error experimentation with the Advanced Adaptation Figure of Merit may result in settings that improve bit error rate, or may lead to further insights on how to improve the performance of the system. In normal operation, the Advanced Adaptation Figure of Merit is not needed and should not be enabled. For almost all systems, the default adaptation figure of merit will provide the best BER performance. It is possible to improve the adaptation performance by loading a non-default CTLE table into the DS110DF410 retimer. The procedure for loading such a table is controlled by the controls on this tab and will be described shortly.

Just to the right of the Eye Monitor Voltage Range control are two standalone buttons. The first is labeled Refresh Indicators. All this button does is read the register settings that control the indicators on the Receiver panel and refresh the status of the indicators. Normally this is not necessary, but if the Single Register Interface is used from the Receiver panel the indicators may not reflect the current state of the DS110DF410 retimer. If you suspect that this may be the case, click the Refresh Indicators button and the indicators will be updated to reflect the current state of the DS110DF410 retimer.

The second button, labeled Load CTLE Table, is used to load a non-default CTLE table as mentioned above.

When the DS110DF410 starts to adapt the CTLE, either to acquire lock or to optimize the eye-opening figure of merit, it steps through a defined set of CTLE settings. These settings have been designed to provide monotonically-increasing CTLE boost for many channels. They are optimized for backplane channels, either stripline or microstrip, on a printed circuit board substrate. Channels fabricated on FR-4 and its derivatives can be effectively equalized by the default CTLE table, as can channels fabricated on lower-loss RF substrates.

For systems where the channel consists of a cable, however, the default CTLE table may not provide optimum equalization. This is because the loss characteristics of a cable as a function of frequency are different from those of a backplane channel. For these types of systems it may be desirable to load a new, non-default, set of CTLE settings through which the DS110DF410 retimer will step during equalization.

In order to load such a table, click the Load CTLE Table button. This will cause a file selection window to appear. The CTLE table files are simple text files which can be created or modified using any text editor. The default extension for the CTLE table files is ".eqx", but All Files is also a valid setting for the file type in the file selection window. Below is an example of the format of the CTLE table file.

[DS110DF410]

StartIndex=0

CTLESettingsList=[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,3,12,13,111,23,33,1111,233,333,2222,1333,2333,3333]

The first line in the file, [DS110DF410], is a tag that indicates to the Analog LaunchPAD software that this is a valid CTLE table file. Following this is an optional value StartIndex. This is the index into to CTLE table settings array at which the supplied CTLE settings list should be inserted. This can be useful if you only want to replace part of the CTLE table. In the case shown, the StartIndex is set to 0, meaning that the settings provided in the CTLE settings list will be inserted starting at the beginning of the table.

The CTLESettingsList is in a Python list format. In fact, in the Analog LaunchPAD program, when the CTLE settings list is read in it is treated as a Python list. Note that the CTLESettingsList is comma-delimited and is enclosed in square brackets. It can contain white space, but it cannot contain line feeds, carriage returns, or continuation characters. It must be typed in on a single line in the file.

Note also that the list consists of non-negative integers. Note that the integers range in length from 1 to 4 digits and that no individual digit is greater than 3. These integers are interpreted by the Analog LaunchPAD program as follows.

The "ones" digit is the setting, from 0 to 3, of CTLE stage 0. The "tens" digit is the setting, also from 0 to 3, of CTLE stage 1. The "hundreds" digit is the setting from 0 to 3 of CTLE state 2, and the "thousands" digit is the setting from 0 to 3 of CTLE stage 3. Digits which are not supplied are assumed to be 0. In the file shown, the first 16 settings are all 0. The DS110DF410 retimer will try all these settings in turn, but since they are all the same no improvement in the eye opening will be apparent.

Negative numbers, non-integers, integers with more than four digits, and integers containing digits greater than 3 are all badly-formed values and will be ignored by the Analog LaunchPAD program. In other words, the CTLE table entry for a badly-formed value will not be changed but will remain at the default setting.

The CTLE table is 32 entries long. List elements at indices greater than 32 will be ignored when the CTLE table setting file is read in.

As soon as a new, valid CTLE settings table file has been selected, the selected channel CTLE table values will be overwritten by the values from the file. When this is complete, if the CDR is reset (using the Reset button on the Receiver tab), the CTLE will be adapted to one of the new values in the revised CTLE table.

The CableEqSettings.eqx file is included in the Analog LaunchPAD software distribution. As noted, the user may create a new CTLE settings table with any text editor. If is recommended, however, that the user verify that the new CTLE settings table provides a monotonic, or almost monotonic, increase in equalizer boost with increasing index, as the adaptation algorithm assumes this to be the case.

As noted, many of the controls on the Receiver tab are used to determine the adaptation algorithms and figures of merit used by the DS110DF410 retimer to optimize the eye opening at the input to the CDR comparator. Also present on the Receiver tab is an indicator that can tell you how you are doing in optimizing the equalization.

When the Acquire HEO/VEO button is pressed, the DS110DF410 retimer measures the horizontal and vertical eye openings at the input to its comparator. These are displayed in UI or ps (for the horizontal eye opening) and in mV (for the vertical eye opening). These values represent the maximum excursion from the center of the incoming signal eye for which the offset comparator produces the same result as the main comparator. These values are peak-to-peak.

As the equalization transfer function approaches the inverse of the channel loss transfer function, the eye will begin to open up. Both the horizontal and vertical eye openings will, in general, get bigger. This is how you know the DS110DF410 has adapted to the correct settings. The eye will be bigger than with any other combination of CTLE and DFE settings.

It should be noted that the point within the circuitry where the vertical and horizontal eye openings are measured is not accessible from outside the device. Even if the output is selected to be Raw Data on the Transmitter tab, the signal still goes through the output driver before arriving at the pins of the DS110DF410 retimer. The measurements obtained from the Eye Opening Values control group on the Receiver tab should be used only as a comparative measurement to determine how well the DS110DF410 has adapted to the incoming signal. It will not be possible to directly compare this to any signal measured external to the DS110DF410 retimer.

The controls on the Receiver tab of the DS110DF410 GUI can be used to fine-tune the adaptation figure of merit, to manually control the equalizer settings, to load a new CTLE adaptation table, and to measure the horizontal and vertical eye opening at the input to the CDR comparator. The DS110DF410 retimer also has to capacity to measure the full internal eye diagram at the same point, the input to the CDR comparator. This is accomplished using the controls on the Eye Monitor tab, to be described next.

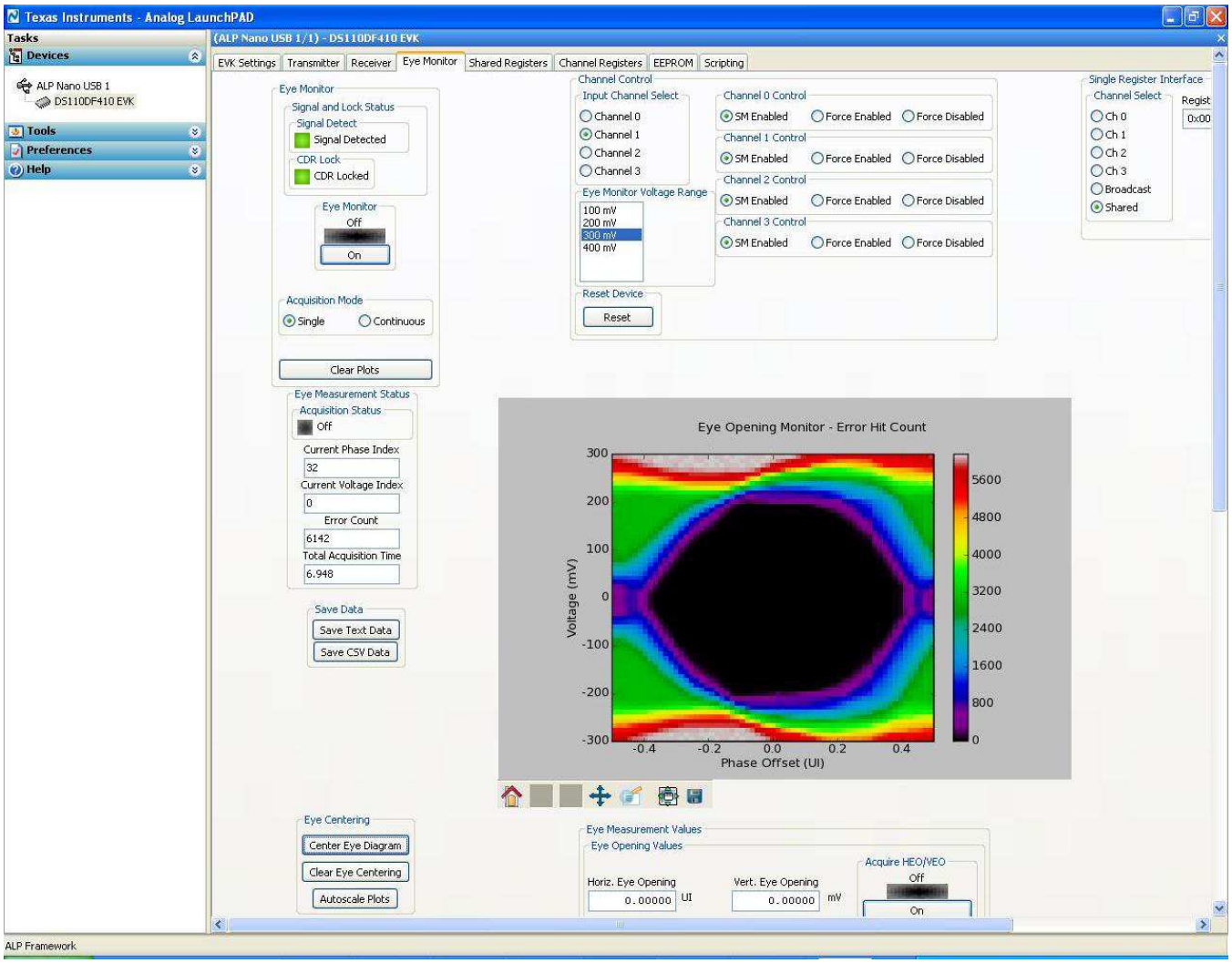

Figure 5. Eye Monitor Tab - Upper Screen

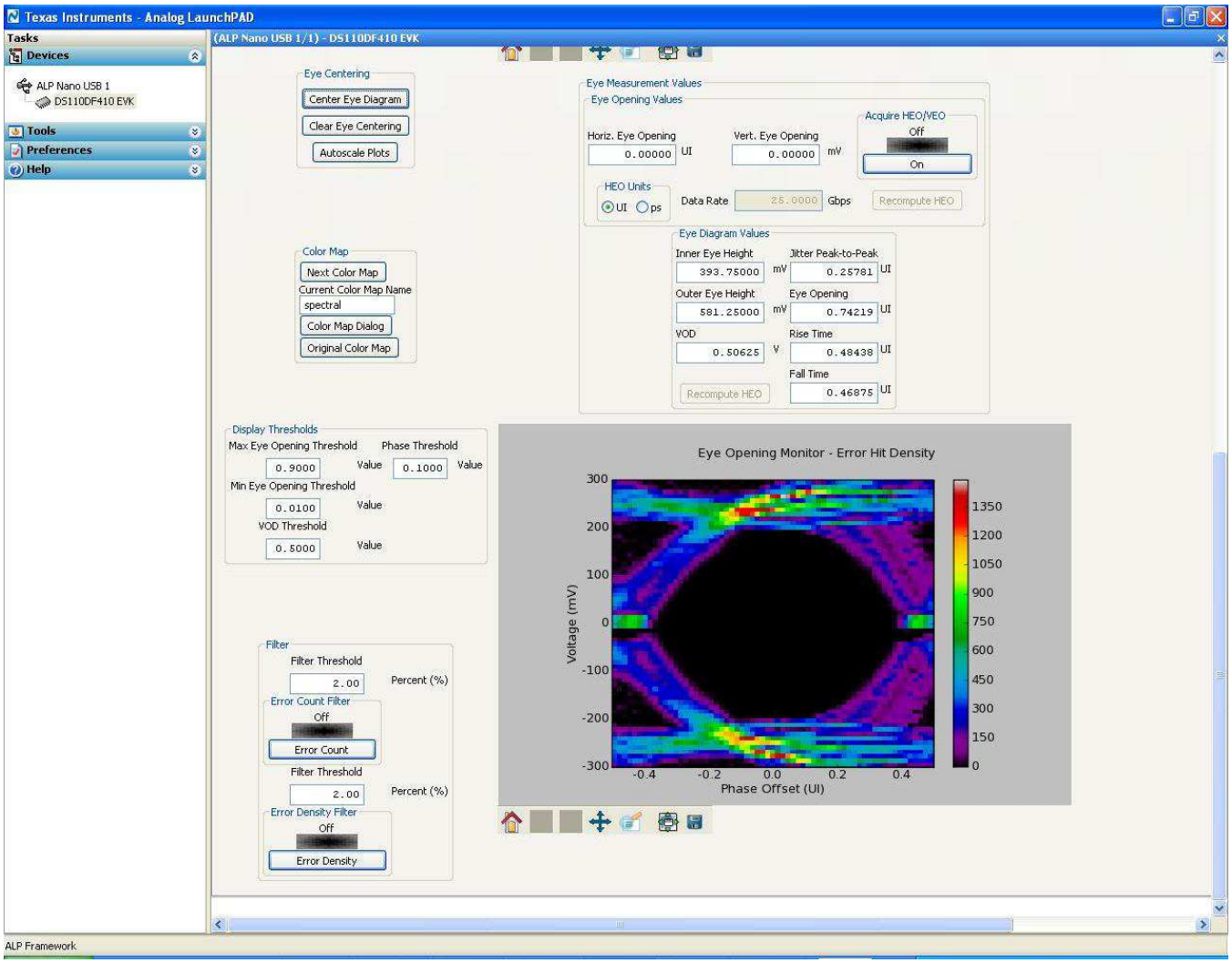

Figure 6. Eye Monitor Tab - Lower Screen

3.4 Eye Monitor Tab

The Eye Monitor tab of the GUI provides an intuitive, control-based screen for measuring the internal eye inside the retimer at the input to the CDR comparator. Many of the controls on the Eye Monitor screen are identical to those discussed in conjunction with the other screens in the GUI. The Eye Monitor screen, for example, includes a single-register interface control, located at the top right of the screen, which is identical in form and operation to those found on the Transmitter and Receiver tabs.

The Eye Monitor tab also includes a Channel Control group, which functions identically to the Channel Control groups on the Transmitter and Receiver tabs. The Channel Control group on the Eye Monitor tab also includes the Eye Monitor Voltage Range control, which functions identically to the one on the Receiver tab. All the controls on the Eye Monitor tab, except the Single Register Interface control, operate on the channel selected in the Channel Control group.

The Eye Monitor tab also includes signal and lock detect indicators, which work identically to those on the Transmitter and Receiver tabs. On the Eye Monitor tab, these indicators are grouped into the Eye Monitor control group.

The Eye Monitor tab also includes a control for obtaining the HEO and VEO values from the DS110DF410 retimer. This control is identical to the one on the Receiver tab. On the Eye Monitor tab, this control is grouped into the Eye Measurement Values control group, immediately beneath the eye diagram hit count values plot.

The other controls on the Eye Monitor tab are designed for acquisition and display of the internal eye diagram. In addition to the signal and lock detect indicators, this control group includes an Eye Monitor on/off button, an Acquisition Mode radio box, and a Clear Plots button.

When the Eye Monitor button is clicked, the DS110DF410 measures its internal eye at the input to the CDR comparator. The DS110DF410 performs this measurement autonomously when commanded to do so via the SMBus, and it reports the data back to the microcontroller on the EVK board over the SMBus when it is done. When the microcontroller gets the data, it sends it back to the Analog LaunchPAD GUI for plotting.

If the Acquisition Mode radio button is set to Single, then after the eye has been acquired the Eye Monitor button state goes back to Off. The eye acquisition can be started again by clicking the Eye Monitor button again.

If the Acquisition Mode radio button is set to Continuous, the DS110DF410 retimer is retriggered to acquire the eye diagram again immediately upon completion of each eye diagram measurement. The plot is updated twice per acquisition. To cause the DS110DF410 retimer to stop acquiring eye diagram data, either set the Acquisition Mode radio button back to Single or click the Eye Monitor button to turn it off.

When the Clear Plots button is clicked, the eye diagram and density diagram plots are erased. When new data is acquired, it will be plotted in place of the data that was erased.

The DS110DF410 retimer features, in addition to its primary data comparator which decides if a given bit is a one or a zero, a secondary comparator which is used to acquire the internal eye measurement. The reference voltage for the secondary comparator can be shifted away from the nominal 0V (really approximately mid-rail, since the DS110DF410 operates from a single 2.5V supply). The clock sampling point can also be shifted away from 0 (nominally the center of the eye) to offset values of ± ½ Unit Interval (UI). Shifting the reference voltage and the sampling point of the secondary comparator generates the eye diagram data plotted on the “Eye Monitor” tab.

As the reference voltage is offset from zero (which the DS110DF410 retimer does automatically), at some point the primary comparator and the offset comparator will disagree about whether a bit is a one or a zero. This is counted as a bit error – the assumption being that the primary comparator is always right. This assumption is valid if the CDR is locked and the equalizers are set to approximately the correct settings. The CDR must be locked for the eye measurement to be performed, and the equalizers must be approximately correct for the DS110DF410 retimer state machine to declare that it is locked. So the assumption regarding the primary comparator is valid for most cases of interest.

When it acquires the eye diagram data, the DS110DF410 steps through the clock phase and reference voltage offset settings and accumulates errors for a period of time at each setting. The dwell time for each offset setting is configurable by changing the value of register 0x2A. The default setting for this register is 0x30. The maximum accumulation time is obtained by setting the value of this register to 0xFF.

The eye plot is a plot of error count versus phase offset in UI and amplitude offset in mV. The black area in the middle of the plot is the eye opening, where no errors were detected. Note that the eye is not centered at 0.0 UI phase offset. This is because the electrical lengths of the paths between the two detectors in the DS110DF410 retimer are not precisely integer multiples of 1 UI. There is a control on the “Eye Monitor” tab that post-processes the data and offsets the eye so that its opening is centered at 0.0 UI phase offset which is in eye centering control group.

For some pathological eye patterns, the eye centering algorithm will not work. The eye centering algorithm depends upon finding an open area which may not be centered in phase offset but which is centered in amplitude offset. If the eye centering algorithm does not find this it may apply an incorrect offset to the eye diagram values. If this occurs, the eye diagram centering can be cleared and the original view restored by clicking the Clear Eye Centering button.

The eye diagram plots are auto scale by default, but if old data is retained in the eye diagram plot the scaling may be corrupted. If this occurs, click the Autoscale Plots button.

While we are on the subject of the appearance of the eye diagram plot, there is a control on the Eye Monitor tab to set the color map used for plotting. The button labeled Next Color Map allows you to cycle through a long set of standard color maps. The Color Map Dialog button allows fine-tuning of the color map segment endpoints. And the Original Color Map button resets the color map to the default, Spectral. In most instances, the default color map provides the most visually-pleasing display of the eye diagram. However, other color maps, and color map adjustment, are available on the Eye Monitor tab if desired.

The Eye Measurement Status control group indicates whether the eye opening measurement is currently active or not. It also displays the current phase index which is always either 0 or 32. A value of 0 indicates that the eye measurement acquisition has completed, and a value of 32 indicates that the eye measurement acquisition is in progress. The current voltage index is always 0 since the displays are only updated at the end of a voltage offset sweep. The error count is the raw error count at the currently displayed phase and voltage indices. It will generally be a large number since the voltage offset is always 0, meaning the largest negative voltage offset at the bottom of the plot.

The total acquisition time is the measured time between the initiation of the eye measurement sequence and plotting the last update. Measuring an entire eye diagram, 64 phase steps by 64 voltage steps, generally takes a little under 7 seconds.

The Save Data control group contains two buttons. The first Save Text Data, saves the data from the current eye diagram plot into a text file in a format so that it can be easily re-loaded into the eye diagram plot window when the Analog LaunchPAD program is in demo mode. This text file is human-readable. The Analog LaunchPAD software distribution includes an example of this text file in the .\Profiles\EOM directory. The file is called EyeDiagramValues.txt.

The second button in this control group, Save CSV Data, saves the eye diagram data in the familiar CSV format so that it can be loaded into a spreadsheet. A picture of the same eye diagram data is loaded into a spreadsheet. The values in the eye opening are all zeros.

The controls located beneath the eye diagram plot are used to save, pan, and zoom the eye diagram plot. If you hover your mouse cursor over these controls, tool tips will appear that describe the functions of each control. Various graphical file formats are available for saving the eye diagram plots but not all of them are compatible with Windows XP. The .png file format generally works best for saving the eye diagram.

Directly beneath the eye diagram plot is a group of indicators labeled Eye Measurement Values. The indicators in the box labeled Eye Opening Values display the values measured by the internal eye opening measurement circuitry in the DS110DF410 retimer. These indicators work the same way as the corresponding indicators on the Receiver tab.

The indicators in the box labeled Eye Diagram Values display values computed from the full eye diagram measurement. The computation of these values is affected by the thresholds set in the control group shown in the Display Thresholds group. The Max Eye Opening Threshold and Min Eye Opening Threshold controls determine the ratio of the eye opening error count to the maximum error count for the eye diagram plot. These controls affect the computed value of the inner and outer eye height.

The VOD Threshold control affects the computation of the VOD. For the example shown here, the VOD is measured as the difference between the positive and negative offsets for which the error count is ½ the maximum error count recorded in the eye diagram data.

The Phase Threshold control sets the error count ratio where the edges of the eye diagram are considered to be. This affects the horizontal eye opening measurement and the jitter measurement.

The default values of these parameters are suitable for many general applications, but they can be adjusted to suit the needs of a particular application if desired.

There is a second graphical display on the Eye Monitor tab. This is the Error Hit Density plot. This plot is derived from the same data as the Error Hit Count plot, but instead of the raw number of errors at each phase and voltage offset, this plot shows the difference between the error count at the current voltage offset and the error count at the previous voltage offset. Locations on the plot where the value is high represent voltage offsets (and phase offsets) at which the number of errors is increasing quickly. These are at the edges of the eye diagram. This plot can provide additional insight into the character of the eye diagram inside the DS110DF410.

In the vicinity of the eye crossing, you will see non-zero values plotted at large voltage increments. These are artifacts caused by the fact that the raw error count values are all very large in those regions so the difference may be greater than 0 just by random variations in the number of eye samples accumulated at those offsets. These artifacts may be reduced by using the eye diagram filter controls. When the filters are active, differences smaller than the specified percentage differences between samples at adjacent voltage offsets are zeroed out. This reduces the artificial speckle near the eye boundary.

As noted, the eye diagram values can be downloaded into a CSV file for further processing if desired. The density diagram features the same panning, zooming, and saving controls as the error hit count diagram.

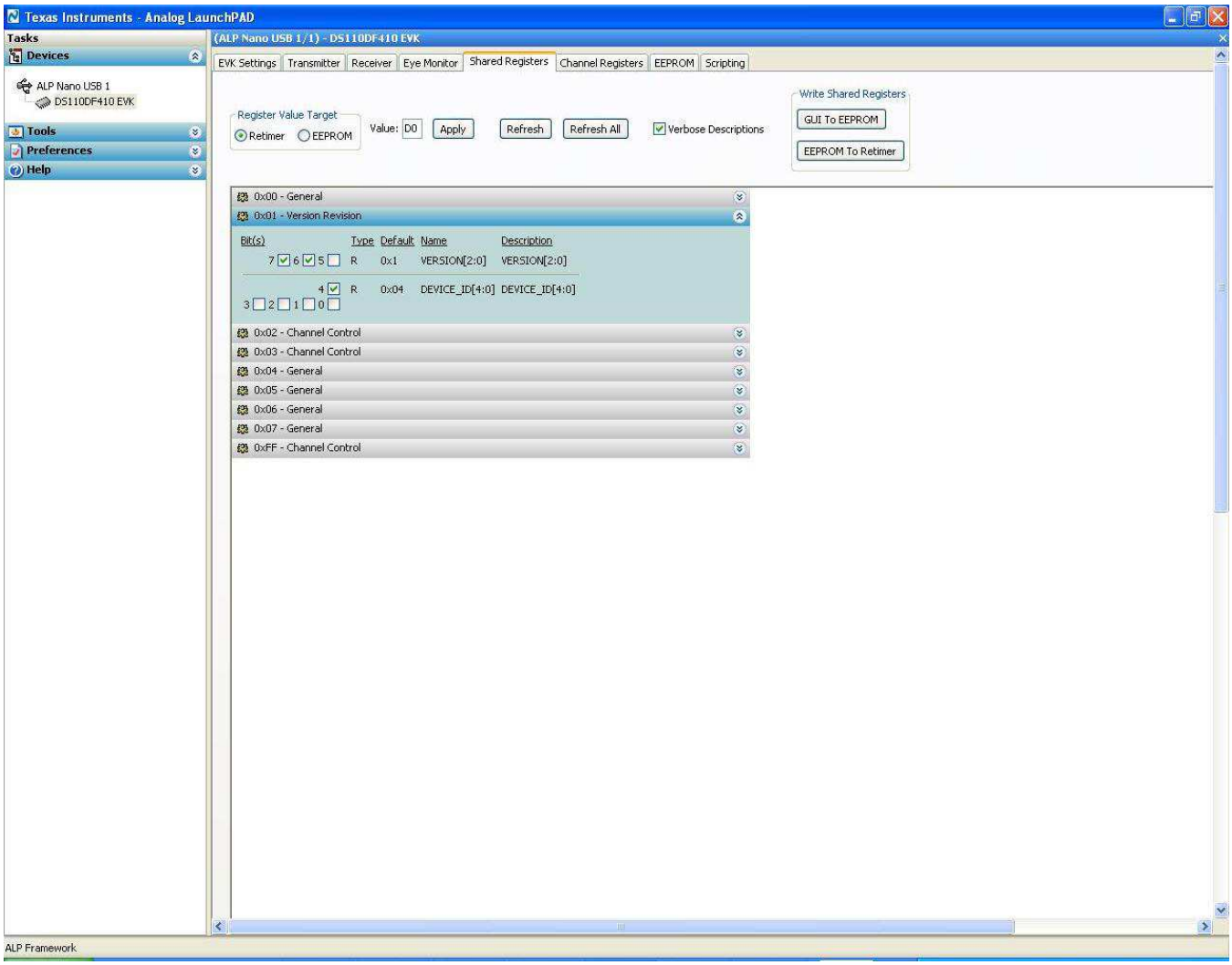

Figure 7. Shared Registers Tab Screen

3.5 Shared Registers Tab

The Shared Registers tab of the GUI provides direct access to the shared register set in the DS110DF410 retimer.

Note that there are only a few shared registers in the DS110DF410 retimer. These are registers that control the operation of the entire device, rather than just a single channel, and the Channel Control register, which is used to select one or all device channels as a register target.

When we click the arrow next to one of the device registers, a panel opens showing the details of the register. The register panels on the Shared Register tab and the Channel Register tab work identically.

Note that the values for the individual bits in the eight bit register are shown as check boxes. A checked box means the value in the register is 1, an unchecked box means it is 0. Note also that the value of the register is shown in the text box labeled Value in hex. In this case, the value shown is 0xD0.

The radio box labeled Register Value Target determines where the values displayed in the register value panels come from. The register value target is the Retimer. This means that the register value displayed is the value in the DS110DF410 retimer's register 0x01. There is, however another possible register target setting.

When the register target is set to EEPROM, the value displayed is the value stored in the EEPROM on the microcontroller on the board. The microcontroller maintains an image of the register map of the DS110DF410 retimer. The values in the microcontroller EEPROM can be the same as those in the device, or they can be different. The microcontroller EEPROM register set can be used in several different ways.

First, for either the retimer or the microcontroller EEPROM, writeable registers can be changed by changing the state of the check boxes on the register panel display. For the panel shown, the values in register 0x01 are not writeable. This is not surprising since they indicate the device ID and revision. But let's look at register 0x02, in this register, bits 5 and 4 are both set to 1. This means that channel 3 is set to control the GPIO. The GUI sets this automatically as the GPIO is used to detect the signal detect. However, if we want to change this value, we just uncheck one of the check boxes, such as the one for bit 4, and click Apply. This writes the new value to the DS110DF410 retimer.

The value of this register in the retimer has now been modified to reflect the value entered on the panel. However, if we select EEPROM as the register target, without doing anything else, we will see the value change.

The value does not affect the operation of the DS110DF410 retimer. This value is just stored in the microcontroller EEPROM. To cause this value to take effect, we have to write it into the register in the DS110DF410 retimer. The operations with the EEPROM will be described in this section.

First, though, let us look at the buttons at the top of the Shared Registers tab. The Apply button has already been discussed. This button writes the value shown in the currently selected register pane either to the DS110DF410 retimer (if the Register Value Target radio box selection is Retimer) or to the microcontroller EEPROM (if the Register Value Target radio box selection is EEPROM).

The Refresh button reads the current contents of the selected register either from the DS110DF410 retimer or the microcontroller EEPROM. The Register Value Target radio box selects which of these the register value is read from.

The Refresh All button reads all the shared registers and updates all the panes in the GUI whether they are open or not. Again, the Register Value Target radio box selects which of these the register values are read from.

Unchecking the Verbose Descriptions checkbox will display less information about each register bit in the register panes. There is usually no reason to uncheck this checkbox.

The control group labeled Write Shared Registers is used to write the entire shared register set for either register target. If the GUI to EEPROM button is clicked, the entire shared register set is read from the GUI or from the DS110DF410 retimer and written to the microcontroller EEPROM. If a pane is open, the register values are read from the GUI. The reason for this is that these values may have been changed by the user, but they may not have been applied to the retimer. In any case, what gets stored in the microcontroller EEPROM is the value displayed on the GUI.

If a pane is not open, the value stored in the EEPROM is the value in the DS110DF410 retimer register.

Clicking the GUI to EEPROM button permits you to store an image of the DS110DF410 retimer's shared registers in the microcontroller EEPROM. It also permits you to store different values for a given register if desired.

Clicking the EEPROM to Retimer button writes the shared register set stored in the microcontroller EEPROM to the DS110DF410 retimer on the EVK board. This allows you to store a set of register values in the EEPROM and then write them into the retimer with a single click whenever you want. It is also possible to write the values from the microcontroller EEPROM into the DS110DF410 retimer on power up without even starting the Analog LaunchPAD. This will be discussed in conjunction with the Channel Registers tab.

At the far right-hand side of the Shared Registers tab are three more buttons, labeled Display, Load, and Save.

Clicking the Display button brings up a window. This window displays the contents of the entire shared register set, either from the DS110DF410 retimer or from the microcontroller EEPROM, depending upon the Register Target selection. The window is dismissed by clicking the OK button at the bottom of the window.

The Save button brings up a save file dialog. This dialog allows you to save the current set of register values into a text file with the special extension ".nrd". The first element on each line of the file is the register page, which will always be 0 for the DS110DF410 retimer. The next element is the register address as a hex number, and the third element is the register value.

Finally, just as you can save a set of register values from the Shared Registers tab, you can load in a saved set using the Load button. This button opens up a file dialog, which allows you to select a previously-saved register settings file. After the file is selected, it is immediately loaded into either the DS110DF410 retimer or the microcontroller EEPROM, depending upon the Register Target setting.

The shared registers control some aspects of the overall operation of the DS110DF410 retimer, but the primary control of the device is through the channel register sets. These will be discussed next.

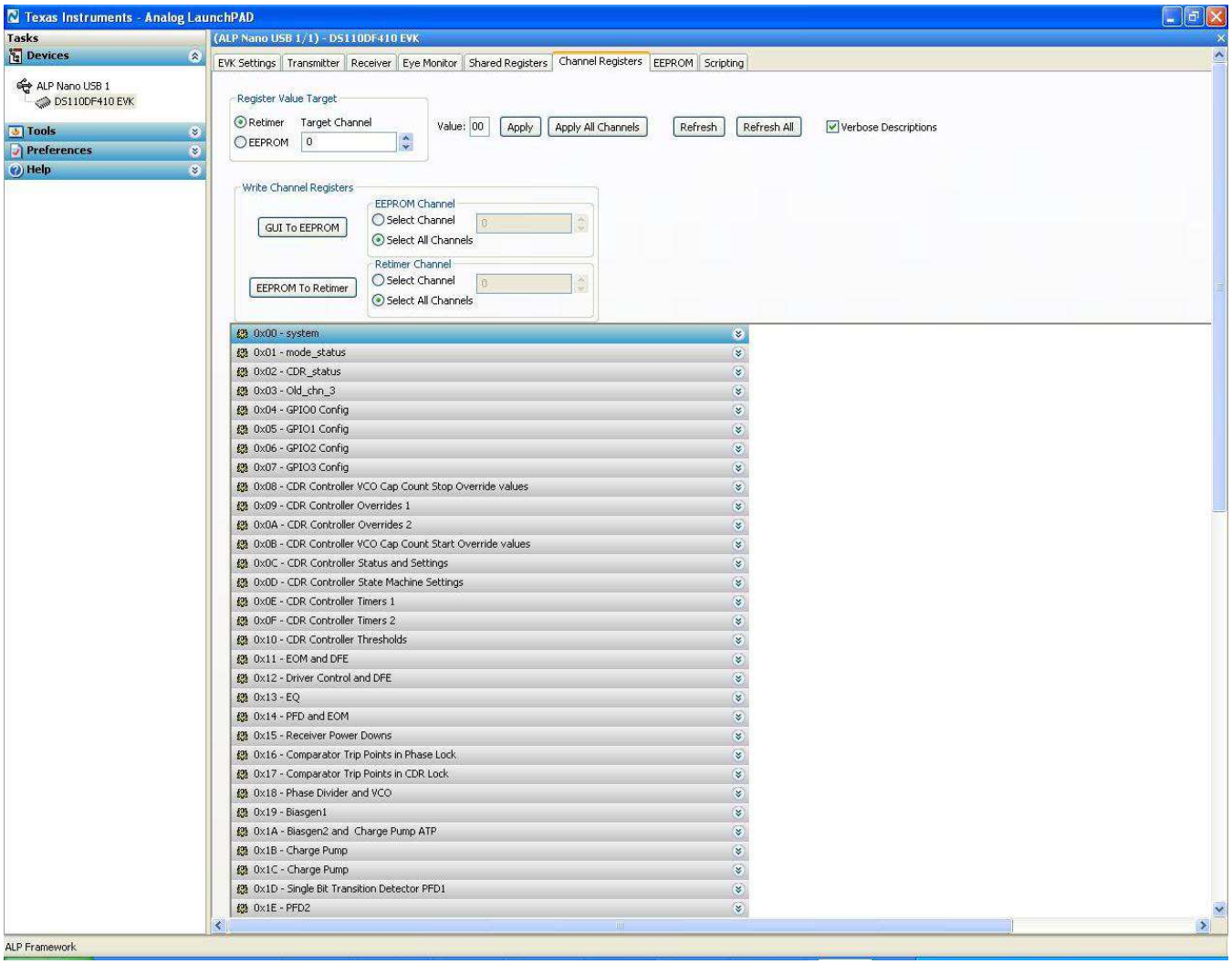

Figure 8. Channel Registers Tab Screen

3.6 Channel Registers Tab

The Channel Registers tab is very similar to the Shared Registers tab. It highlights a feature of the GUI toolkit used for the Analog LaunchPAD software, wxPython. The tabbed notebook control, which is the underlying control for all the tabs we have discussed in this document, automatically resizes its tabs so that they are all the same size as the largest one displayed. So when the Eye Monitor tab is displayed, which is the largest of the seven tabs in the DS110DF410EVK GUI, all the other tabs are resized to display at the same size. This may cause some confusion since the sizes of the screens are seen to change and horizontal and vertical scroll bars may sometimes be present and sometimes not. This is a feature of the GUI toolkit. When a scrollbar is needed, there will be one on the screen. When it is not needed, you can be sure that all the controls are visible and accessible without it.

The Channel Registers tab includes some controls that have already been discussed in conjunction with the Shared Registers tab. The Display, Load, and Save buttons on the Channel Registers tab work exactly the same as their counterparts on the Shared Registers tab. In the case of the Channel Registers tab, the register set is the set for one channel. However any or all channels can be loaded with a set of registers saved in a ".nrd" file. Different files can be used to set the different channels to different settings.

The Apply, Refresh, Refresh All, and Verbose Descriptions controls work exactly the same way on the Channel Registers tab as they do on the Shared Registers tab. Note, however, that the register target setting control is slightly different for the Channel Registers tab.

The Register Value Target control on the Channel Registers tab includes both radio buttons to select Retimer or EEPROM, and a spin control to select a channel. The channel selected is the channel from which register values will be read when a pane is opened on the Channel Registers tab, and it is also the channel to which the register values will be written when you click the Apply button. It is also possible to write the values in the current register pane to all channels of the retimer by clicking the Apply All Channels button. This button writes the values in the GUI for a single register to all channels either of the DS110DF410 retimer or the microcontroller EEPROM, depending upon the register target selected. This is a quick way to copy a register setting from one channel to all the other channels of the DS110DF410 retimer.

The Write Channel Registers control group on the Channel Registers tab works essentially the same as the Write Shared Registers control group on the Shared Registers tab. The only difference is that the values in the GUI, which, remember, are read from a single channel of the device or the EEPROM and may be modified by the user, can be written either to a single channel (not necessarily the one they came from) or to all the channels of the EEPROM with a single click. If the Select All Channels radio button is selected, and the GUI to EEPROM button is clicked, all channels of the microcontroller EEPROM register map will be written with the same values. If the Select Channel radio box is selected, then all the values from the GUI will be written to a single channel of the microcontroller EEPROM register map.

This can be used to copy a set of register settings from one channel in the EEPROM or in the DS110DF410 retimer to another channel in the EEPROM. To do this, select the channel from which the register settings are to be read in the Register Value Target control. Change any register values desired manually by opening the register pane and setting the desired bits. Then select the channel in the EEPROM to be written in the Write Channel Registers control group's EEPROM Channel selection control. Note that this does not have to be the same channel from which the register values were read.

After the channel is selected, or all channels are selected, click the GUI to EEPROM button. This writes the values from the Register Value Target channel to the EEPROM Channel.

Separate register images are maintained in the microcontroller EEPROM for the shared register set and for each channel register set. The instructions above describe how to save a given set of channel registers in the microcontroller EEPROM. The EEPROM to Retimer button allows you to write the values from the microcontroller EEPROM to any channel, or to all channels, of the retimer, but this works a little differently than does the GUI to EEPROM write control.

If you select a single channel in the Write Channel Registers control group's Retimer Channel control, then click the EEPROM to Retimer button, the EEPROM register data for the selected channel will be written into the same selected channel in the DS110DF410 retimer. If you select All Channels in the Retimer Channel control, then the EEPROM data for each channel will be written into the corresponding channel in the DS110DF410 retimer. The register settings for each EEPROM channel may be different. If it is, the data in the DS110DF410 retimer will also be different after the EEPROM to Retimer button is clicked.

It should be emphasized that changing individual register bits from the Shared Registers and Channel Registers tabs can produce unexpected results. Some register settings interact with each other in ways that may not be immediately apparent. You should not change the register settings from these tabs unless you are certain of what you are changing. A better method is to use the controls on the Transmitter, Receiver, and Eye Monitor tabs to configure the DS110DF410 retimer as desired, and then use the EEPROM controls on the Shared Registers and Channel Registers tabs to save the settings into the microcontroller EEPROM where they can be easily reloaded into the DS110DF410 retimer. Here is an example of how that would work.

As you may have already realized, the DS110DF410EVK board can be used as a crosstalk generator for system testing. It can produce either PRBS-9 or PRBS-31 pseudo-random bit sequences at various bit rates without the need for an input signal. It can also lock to an incoming clock to produce these bit sequences in a way that is synchronous to the clock. Suppose you want to set up all the channels of the DS110DF410EVK to produce free-running PRBS-31 data at about 10.3 Gbps whenever the board powers up. These are the steps to accomplish this.

1. Select any channel on the Transmitter tab. Let's say we select channel 0. Use the Input Channel Select control.
2. Check the Enable VCO Free Run check box.
3. Set the VCO Divider Selection to 1.
4. Set the CAP DAC Setting knob to 9 and the LPF DAC Setting knob to 19. You may need to experiment with the exact values of these settings to get the desired output data rate. Use a spectrum analyzer to look at the VCO I-Clock and adjust the LPF DAC Setting and the CAP DAC Setting to get the desired output frequency.
5. Set the PRBS Gen Control to PRBS-31.
6. Set the Output Data Selection control to PRBS Generator.
7. Set the Driver VOD Setting and Driver De-emphasis to the desired values. You may need to look at the output on an oscilloscope to set these values as desired.

At this point, channel 0 of the DS110DF410 retimer should be generating a free-running PRBS-31 pattern at a data rate of approximately 10.3 Gbps with the desired output VOD and de-emphasis settings.

Now select the Shared Registers tab. Write the GUI values to the EEPROM using the Write Shared Registers control.

Now select the Channel Registers tab. Select the Register Value Target to be the Retimer and select the Target Channel to be channel 0. Remember, this is the channel we just configured for PRBS-31 generation.

Now make sure that in the Write Channel Registers control group, the EEPROM Channel selection is set to Select All Channels. Once you have that, click the GUI to EEPROM button. This will write the values from channel 0 of the DS110DF410 retimer, which we just set up using the steps above, into all the register channels of the microcontroller EEPROM. This operation may take 30 seconds or so since the EEPROM write operation in the microcontroller is slow. Wait until the GUI to EEPROM button returns to its "unpressed" state before proceeding.

Now you can do two things. From the Channel Registers tab you can write from the EEPROM to the retimer. Insure that the Retimer Channel control in the Write Channel Registers control group is set to Select All Channels. Click the EEPROM to Retimer button to write the values from the microcontroller EEPROM into all the channels of the retimer. The settings for all the channels in the microcontroller EEPROM are identical because we wrote the same register set to all the EEPROM channel register sets.

Now all the retimer channels should be generating free-running PRBS-31 signals at about 10.3 Gbps.

The other way to make this happen is to install a jumper on pin PF7 on header J38 on the DS110DF410EVK board. Jumper between pins PF7 and the ground pin on the row beneath it to cause the microcontroller to automatically load the values stored it its EEPROM into the DS110DF410 retimer every time the board is powered up. Make sure that a jumper is also installed on J41 in the Slave position. The current firmware on the DS110DF410EVK board does not support Master SMBus mode, so this jumper should always be set to the Slave position.

After you have followed the steps above and installed the jumper, the DS110DF410 board will power up every time with all four channels generating free-running PRBS-31 data at about 10.3 Gbps. It is not necessary to start ALP or to connect the USB cable. The microcontroller will load the correct register settings automatically. If you want the DS110DF410 to power up in the default state instead, just remove the jumper on PF7.

After the DS110DF410EVK board powers up, whether or not the microcontroller has loaded register settings into the DS110DF410 retimer from the EEPROM, you can start Analog LaunchPAD and connect the USB cable to the DS110DF410EVK board to change any settings you want. You can also rewrite the information in the microcontroller EEPROM. If you don't change the register settings in the EEPROM, the board will once again power up in the previously-stored state when the jumper is replaced.

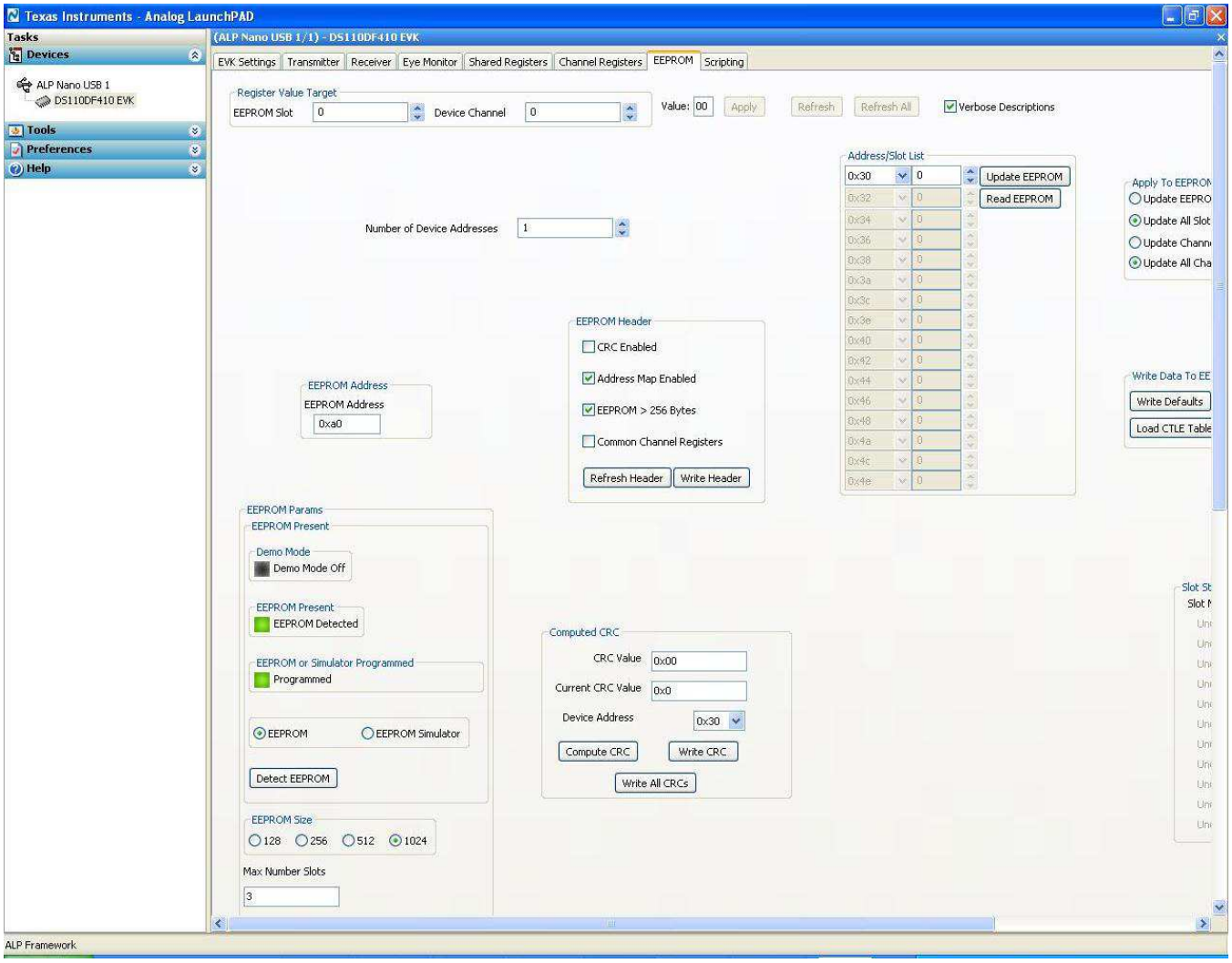

Figure 9. EEPROM Tab - Upper Screen

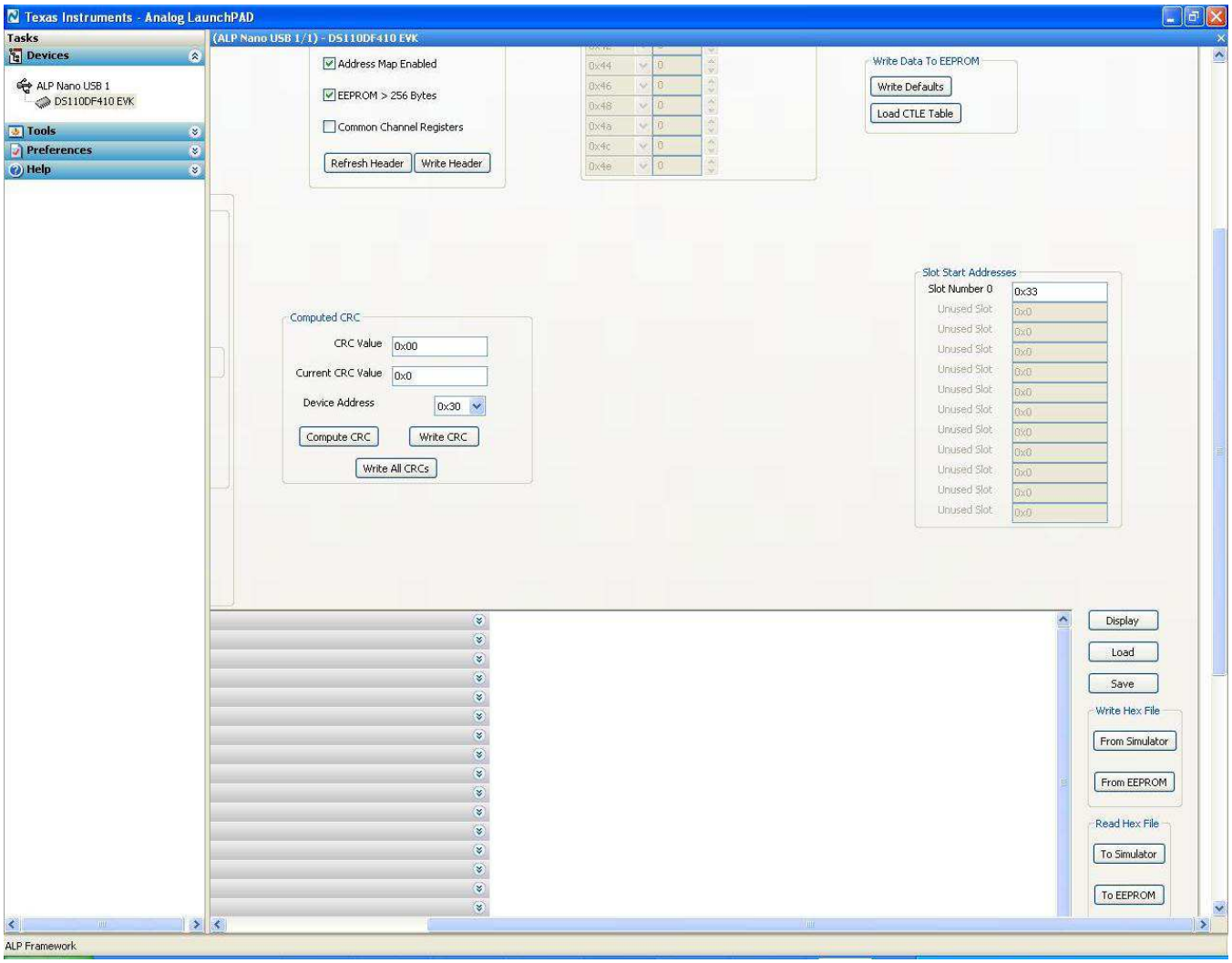

Figure 10. EEPROM Tab - Lower Screen

3.7 EEPROM Tab

The EEPROM tab provides the software and hardware for in-system programming of the EEPROM for the DS110DF410 configuration. The data to be programmed into the EEPROM is encoded in an Intel-format hex file. The software tool can create hex files and load them into an EEPROM.

Analog LaunchPAD is configured using a text file called alp.ini which resides in the main program directory, usually c:\Program Files\National Semiconductor Corp\Analog LaunchPAD v1.2x.xxxx, where the last string is the version of Analog LaunchPAD installed on your computer. The EEPROM GUI is enabled by adding strings to the alp.ini file like this.

[DS110DF410]

EnableEEPROM=Yes

The tag [DS110DF410] specifies that the following string applies to the DS110DF410 family retimers profiles, and the string EnableEEPROM=Yes indicates that the EEPROM GUI is to be displayed.

Let's look at the various controls and displays on this window. To begin with, there is a control group labeled "EEPROM Params". This control group includes an indicator for Analog LaunchPAD's demo mode. This is the mode in which Analog LaunchPAD runs if there is no board attached to it.

There are also indicators for the presence of an EEPROM and for whether or not the EEPROM is programmed. EEPROM present is detected by the software by attempting to write a value into a memory location on the EEPROM and then read it back. If this is successful, the EEPROM is present. The same mechanism is used to detect the EEPROM size. The software writes a value to sequentially higher memory addresses in the EEPROM and attempts to read it back. If this is successful, the EEPROM must be at least as big as the memory address the software wrote and attempted to read back.

The EEPROM is assumed to be programmed if the byte at EEPROM address 0x01 is 0x00. This byte is not used by the retimer, so the software programs it to 0x00 to indicate that the EEPROM in the system has been programmed.

A radio button selects either the EEPROM, if one is detected, or the EEPROM GUI simulator. The EEPROM GUI simulator is just an area of memory that mimics an EEPROM. The EEPROM GUI simulator is the only selection available when Analog LaunchPAD is operating in demo mode. The EEPROM GUI simulator is also available even when an EEPROM is attached to the SMBus, however. The simulator can act like a scratch pad. Once an EEPROM image with the desired settings is programmed into the simulator, a hex file can be produced directly from the simulator which can be used to program an EEPROM.

When the EEPROM GUI simulator is in use, the EEPROM size to be simulated can be selected by the “EEPROM Size” radio buttons. When a real EEPROM is detected, these radio buttons show the detected size of the EEPROM.

EEPROM detection occurs when the EEPROM GUI is first displayed. However, if an EEPROM is connected to the SMBus after the EEPROM GUI is displayed, the “Detect EEPROM” button will initiate the EEPROM detection sequence again.

Finally, in this control group, there is an indicator for the maximum number of data slots that will fit in the selected (or detected) EEPROM. This depends upon the header configuration, as described previously. The maximum number of EEPROM slots is computed and displayed in this control group for convenience.

Several controls are used to set the header parameters in the EEPROM or the simulator. When an EEPROM is detected, the values in its headers are used to populate these controls.

First, there is a spin control box for the number of device addresses in the EEPROM. This value is encoded in the 0th byte of the EEPROM. It is not used by the retimer, but the software uses it to determine how many address headers should be displayed. This control enables the controls in the Address/Slot List control group.

The Address/Slot List control group shows how the retimer SMBus addresses are mapped into the address headers. If only one data slot 0, is used, then the slot number should be se to 0. All the EEPROM addresses are mapped to this data slot 0.

In this control group, the “Update EEPROM” button writes the address headers to the EEPROM or the simulator (depending upon the selection in the EEPROM Params control group). The “Read EEPROM” button reads the header information from the EEPROM or the simulator.

The “EEPROM Header” control group specifies the contents of byte 0 of the EEPROM or the simulator as previously described. The “Refresh Header” button reads the current contents of byte 0 and sets the header controls. The “Write Header” button writes the current values shown on the header controls into the EEPROM or the simulator.

Finally, the “EEPROM Address” indicator is set to 0xa0. This is the required EEPROM address for the DS110DF410 family retimers.

The data in the headers is used to populate the “Slot Start Addresses” indicators. If there is only one slot used in the EEPROM, slot 0, so this is the only slot for which the starting address is defined.

The "Register Value Target" is used to write a single value to a single EEPROM slot and channel. When the common channel registers are selected, writing to any channel really writes to channel 0, since this is the only channel in the register set. The register address to be written must be selected using the fold panels in the lower left of the GUI.

The highlighted register in the fold panel is the active selected register. The state of the check boxes in this register panel show the current contents of the slot and channel register selected using the "Register Value Target" control. To write a new value to a selected slot and channel, use the check boxes in this fold panel to set the desired bits and then click the “Apply” button. To read the contents of this slot and channel, use the “Refresh” and “Refresh All” buttons. “Refresh” just reads the currently selected register. “Refresh All” reads all the registers and sets the contents of all the fold panels according to the contents of the selected EEPROM slot and channel.

After changing the register contents using the fold panels, you must click the “Apply” button to save the changes to the EEPROM or the simulator. If you highlight another register without clicking the “Apply” button, the changes you have made to the first selected register will be lost.

If “Common Channel Registers” is selected, it is only necessary to write one channel of each slot. If not, each channel of each slot must be updated separately.

Note that there is a “Value” text field next to the slot and channel target controls. The desired value for the register can be entered in this field in hex. This is useful if more than one slot or channel is to be updated, because the value that appears in both this box and in the fold panel is the value currently contained in that slot or channel. If you update a slot or channel and then change the register target, the “Value” box will be updated with the contents of the newly-selected register target. But you can easily enter the same value into the newly-selected register target by just typing it into the “Value” box and hitting the “Apply” button.

The “Refresh” button reads the current contents of the selected register target and applies them to the “Value” box and the selected fold panel. The “Refresh All” button reads all the memory locations in the currently selected slot and channel (the register value target) and refreshes the value in all the fold panels, even those not currently selected.

It is also possible to update all the information in a given EEPROM slot and channel, or all slots and all channels at once, using the “Apply to EEPROM” control group. With this control you can select as a target either all slots or a selected slot, and either all channels or a selected channel. When you click the “Update All” or “Update Slot” button, all the memory locations in the EEPROM corresponding to the selected slot and channel, or all slots, or all channels, are written from the current contents of the fold panels. Remember that these contents are always the same as the contents of the currently selected EEPROM slot and channel except for the selected fold panel, which may be different. This control group is mostly useful for copying information from one EEPROM slot to another.

The second control group is the “Write Data To EEPROM” control group. The “Write Defaults” button in this control group allows you to write all the slots and all the channels in the EEPROM to the factory defaults. This is useful for initial setup of the EEPROM. Changes can be made using the default settings as a starting point.

The “Load CTLE Table” button loads a CTLE table, in the .eqx format described in the Recevier Tab section can be used to load into all EEPROM slots and channels in a single click. Normally loading a new CTLE table would require modifying 32 registers for each slot and channel. This button loads all the CTLE table values in a single operation.

Once the EEPROM slot information has been written, the “Slot Start Address” indicator shows the starting address of each register data slot in the EEPROM. This is mainly useful as a sanity check to insure that the slots start at the expected offsets.

The read, write, and display control groups on the EEPROM tab have the following function. The “Display” button displays a pop-up window showing the entire contents of all the EEPROM locations for the currently-selected register slot and channel. The “Load” button loads the fold panels for the currently selected slot and channel from a .nrd file, which is the standard text file format used for storing register contents. The “Save” button writes a .nrd file containing the current register contents to the disk. Note that neither the “Load” nor the “Save” button changes the contents of the EEPROM. You have to either apply the values or update a slot or slots to change the EEPROM.

The “Write Hex File” control group writes a standard Intel format hex file from either the current contents of the EEPROM or the GUI simulator. This file can be used to program another EEPROM identically to the current one, using either Analog LaunchPAD itself or another EEPROM programmer that uses Intel format .hex files.

The “Read Hex File” control group loads an Intel format hex file either into the GUI simulator or directly into the EEPROM. If the EEPROM is selected, this operation may take several seconds. A progress bar window will pop up showing the progress of the operation.

If a hex file which was not created using Analog LaunchPAD is read into the EEPROM or the simulator, the software will try to interpret the slot start addresses and usage from the EEPROM or simulator contents. This is not foolproof. In particular, the software may not know how many devices are to be programmed from the EEPROM or what their SMBus addresses are. The software will attempt to determine this, but it may not be completely correct.

If changes are to be made and recorded into a new hex file, or written to the EEPROM, the headers should be written first. Write the 0th byte EEPROM header first, then the number of device addresses, and then the device addresses and slot assignments. After that, check to see that the slot start addresses make sense.

This should not be necessary for hex file that have been generated using Analog LaunchPAD because the slot locations are fixed in these EEPROM files.

The Computed CRC control group can compute the CRC using the same algorithm as the DS110DF410 uses. When the CRC is enabled, the software writes this value to the EEPROM in the correct location.

The CRC value is different for each device address, since it is based on the address map values. The controls display the current CRC value in the EEPROM or the GUI simulator, and the computed EEPROM value based on the current contents of the EEPROM or simulator data slot. When you click the “Compute CRC” button, the software re-computes the CRC. When you then click the “Write CRC” button, it writes the computed CRC to the correct location for the selected device address.

The “Write All CRCs” button does what its label implies. It computes and writes the CRCs for all the device addresses enabled in the EEPROM.

If the CRCs are enabled in the header, the displayed current CRC value should match the computed CRC value. If it does not, the DS110DF410 will try to read a valid data set from the EEPROM using the CRC to check it, and it will continue to retry forever to read a valid data set. The only way to get it out of this operation is to put the DS110DF410 in slave mode or to pull the $\overline{\text{READ\_EN}}$ pin high. This will prevent the DS110DF410 from trying to read from the EEPROM.

The CRC computation is the standard CRC-8 computation. This is described in this Wikipedia page: http://en.wikipedia.org/wiki/CRC-8

3.7.1 Intel Format Hex Files

Below is an example hex file listing.

```
:20000000730010000000003300003300007F0000CB000000000000000000000000000000AD
:200020000000000000000000000000000000000000000000780082893693A2181800A8F406
:200040006D230C91C500001FF3F9439CC204621F8F972E0880004104100200A000C30C10CB
:200060005430182422220A81194A32C00100108183FFFFFFFE42CE42CFFE8000000000000EE
:200080007800828936 93A218180020F46D230C91C500001FF3F9411CC204621F8F972E0831
:2000A0008000410410020 0A000C30C10543018242220A81194A32C00100108183FFFFFFF5F
:2000C000E42CE42CFFE8000000000007800828936 93A2181800A8F46D230C91C500001F4E
:2000E000F3F9439CC204621F8F972E0880004104100200A000C30C10543018242220A81181
:2001000094A32C00100108183FFFFFFFE42CE42CFFE8000000000000000083C93693A20849
:20012000180060F46D230C91C500001300394000C104621F8F81014A80004104100200A0BD
:2001400000C30C10543018242220A81194A32C3215D75A5D756665A940000000002800007C
:20016000000000000000083C93693A208180060F46D230C91C500001300394000C104621F90
:200180008F81014A80004104100200A000C30C10543018242220A81194A32C3215D75A5DBB
:2001A000756665A940000000002800000000000000083C93693A208180060F46D230C9196
:2001C000C500001300394000C104621F8F81014A80004104100200A000C30C105430182417
:2001E0002220A81194A32C3215D75A5D756665A940000000002800000000000000083C92F
:20020000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFE
:20022000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDE
:20024000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBE
:20026000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9E
:20028000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7E
```

```
:2002A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5E
:2002C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3E
:2002E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1E
:20030000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD
:20032000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFDD
:20034000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFBD
:20036000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF9D
:20038000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF7D
:2003A000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF5D
:2003C000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF3D
:2003E000FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF1D
:00000001FF
```

In this listing, the first character on each line, a colon (":"), is required. The next two characters ("20") form a hex digit indicating how many bytes are contained on the line. For this file, each line contains 32 bytes (0x20) of data. The next four characters are the starting address of the data on the current line in hex. For example, the starting address of the data on the first line is 0x0000, or 0. On the second line, the starting address is 0x0020, or 32. The next two characters are a required data type. For the DS110DF410 hex files, these are always 00. The next 64 characters on each line are the data in hex. Look at the first line in this hex file. The first data byte is 0x73. This is the 0th byte header for the DS110DF410. This data indicates that CRCs are not enabled, the address maps are enabled, the EEPROM is greater than 256 bytes, and the common channel registers are enabled. The number of devices is 3, which the software interprets as four devices being programmed from this EEPROM (one more than the number of devices in the hex file). The last two characters on each line are a checksum for each line. This is computed by taking the least significant byte of the two's complement of all the byte values on the line except the first character (the colon) and the checksum byte itself. To compute the checksum, add all the other byte values on the line, take only the least-significant byte of the result, and subtract it from 0x100, and then take the least significant byte of the result if necessary. This will only be necessary if the least-significant byte of the sum of the byte values is 0x00. The checksum is computed for each line and is checked by the Analog LaunchPAD software to insure that the hex file is valid.

4 EEPROM and Register Map Informations

The family of quad retimers can be configured on power up using an external EEPROM to set the retimer to non-default operational settings.

This document describes the usage of the external EEPROM to configure the DS110DF410 family quad retimers. It lists the EEPROMs which are supported, illustrates the memory mapping of these EEPROMs, and describes how to program a device configuration into the EEPROM.

4.1 Supported EEPROM

The quad retimers are designed to read a register configuration on power up from an external EEPROM autonomously. When it is configured in SMBus Master Mode, the DS110DF410 family retimer takes command of the SMBus on power up on reads its configuration from the external EEPROM. The retimer is designed to support a particular family of external EEPROMs. It expects the addressing scheme of these EEPROMs to match what it is designed to support. It also expects the data in the EEPROM to match its internal EEPROM data scheme. We will first discuss the addressing scheme of the EEPROM and give examples of some EEPROMs that can be supported.

The DS110DF410 family quad retimers expect the base SMBus write address of the EEPROM to be 0xa0. SMBus addresses are sometimes understood as seven-bit values which are left-shifted by 1 bit and bitwise or-ed with a READ/$\overline{\text{WRITE}}$ bit. In this nomenclature, the SMBus address of the EEPROM is 0x50. When the EEPROM is addressed for a read operation, the address that is sent over the SMBus is 0xa1. When the EEPROM is addressed for a write operation, the address that is sent over the SMBus is 0xa0. This is what is meant by the statement that the base write address of the EEPROM must be 0xa0.

The retimer immediately reads its configuration from the external EEPROM on power up when it is in SMBus master mode. The SMBus address of the EEPROM is fixed in the retimer and cannot be changed. This yields the first requirement on the external EEPROM:

- The base SMBus write address of the external EEPROM must be 0xa0.

The retimer uses an eight-bit memory location addressing scheme for reading the information from the EEPROM. That is, when the retimer attempts to read a memory address in the external EEPROM, it first sends the SMBus write address of the EEPROM, then the eight-bit memory address. It then sends the SMBus read address of the EEPROM and allows the EEPROM to write its data to the SMBus , which the retimer then reads. This is the standard way the SMBus operates for reading memory from an EEPROM.

Clearly, since the memory address is eight bits, the maximum memory address is 0xff or 255. This restricts the address space to 256 bytes. However, the retimer can address a larger address space than this. The maximum address space the retimer can address is 1024 bytes. This is the second requirement on the external EEPROM.

- The size of the external EEPROM must be between 128 bytes and 1024 bytes.

To address memory locations in the external EEPROM with addresses > 255, the retimer uses one or two Least Significant Bits (LSBs) of the EEPROM SMBus address as page bits. For an external EEPROM with 512 bytes two memory pages are required. The retimer uses one bit of the SMBus address as a page bit.

For an external EEPROM with 1024 bytes, four memory pages are required. The retimer uses two bits of the SMBus address as page bits in this case.

Using a 1024 byte EEPROM as an example, if the retimer is to read the EEPROM memory contents at memory address 127 (0x7f), then it first sends the base write address of the EEPROM over the SMBus. This is a one-byte value, 0xa0. It then sends the memory address, 127, over the SMBus. This is a one-byte value 0x7f.

The retimer then sends the base read address of the EEPROM over the SMBus. This is a one-byte value, 0xa1. The retimer then releases the SMBus and the EEPROM writes the data from memory location 127 to the SMBus and the retimer acknowledges receipt of the one-byte value.

Now consider the case where the retimer is to read the contents of memory location 639 (0x27f). The memory address, 639, is too big to be contained in an eight-bit value. So the retimer uses the two LSBs of the SMBus address as page bits.

The retimer sends a write address of 0xa4 over the SMBus. The EEPROM interprets this as its base SMBus write address (0xa0) bitwise or-ed with a two-bit page code of 2. The retimer then sends the same memory address byte as in the previous example, 127, over the SMBus. The EEPROM interprets this as a request for the data at memory location 639 (0x27f).

The retimer then sends a read address of 0xa5 over the SMBus. Again, the EEPROM interprets this as its base SMBus read address (0xa1) bitwise or-ed with a two-bit page code of 2. The EEPROM responds by sending the data at memory location 639 (0x27f) over the SMBus. This is the third requirement on the external EEPROM.

- The external EEPROM must support paging by using the one or two LSBs of the SMBus address as page bits.

Some other fairly obvious requirements for the EEPROM are its I/O voltage capability and its SMBus clock speed.

- The external EEPROM must support 2.5V to 3.3 V SMBus I/O voltages
- The external EEPROM must support 400 kHz SMBus clock speed.

A family of EEPROMs that meets all these requirements is the Atmel AT24C01/2/4/8B family. A data sheet for these EEPROMs can be found at the following URL: http://www.atmel.com/Images/doc8517.pdf

4.2 EEPROM Memory Usage

Conceptually, the EEPROM is divided into three subsections for the purposes of storing configurations for the DS110DF410 family of retimers.

Table 3. EEPROM Memory Subsections

EEPROM Subsection	EEPROM Subsection Name	Starting Address	Subsection Length (bytes)	Required?	Comments
1	Base Header	0	3	Yes	Always present, this header tells the retimer how to interpret the rest of the EEPROM data

Table 3. EEPROM Memory Subsections (continued)

EEPROM Subsection	EEPROM Subsection Name	Starting Address	Subsection Length (bytes)	Required?	Comments
2	Address Map Headers	3	2 – 48	No	Base Header indicates whether the address map headers are used. Location of each address map header is fixed for a given retimer SMBus address.
3	Register Data Slots	Variable	76-77 or 298-299 per slot, multiple slots allowed	Yes	This is where the configuration data for the retimer is stored. A register data slot can be used to configure one or more retimers depending upon the contents of the address map headers

4.2.1 Base Header

The base header must always be present in the EEPROM. It is always stored in memory locations 0-2. The contents of each byte in the base header are described below.

4.2.1.1 Byte 0

The very first byte in the EEPROM must contain byte 0 of the base header. The contents of this byte are described in Table 4.

Table 4. Byte 0 Bit Definitions

Bit Number	Bit Name	Meaning
7	CRC_EN	When this bit is set, CRCs are enabled. If this bit is set, the CRC field in the address header or at the end of the register data slot must match the CRC computed internally by the retimer. If it does not match, the configuration is not loaded.
6	ADDR_MAP_EN	When this bit is set, the EEPROM address headers are used. If this bit is set, each retimer on the SMBus looks for an address map header at a location determined by that retimer's SMBus address. If it is not set, each retimer looks for its configuration at a specific EEPROM starting address, again determined by that retimer's SMBus address.
5	EEPROM_GT_256	When this bit is set, the EEPROM is assumed by the retimer to be larger than 256 bytes. If the retimer needs to address memory locations in the EEPROM at addresses greater than 255, it uses the paging scheme described in EEPROM Memory Usage above.
4	COMMON_CHAN	When this bit is set, the retimer assumes that only one set of channel register information and one set of shared register information is present in each register data slot. It configures all four of its channels according to this channel register information. If this bit is not set, the retimer assumes that each channel has a different set of information in each register data slot. If this bit is set, the register data slot length is 76-77 bytes, depending upon whether the address maps are enabled. If this bit is cleared, the register data slot length is 298-299 bytes, again depending upon whether the address maps are enabled.
3:0	DEVICE_COUNT[3:0]	This field is not used by the retimer, but it is useful to designate the number of address map headers present in the EEPROM.

An unprogrammed EEPROM will generally contain 0xff in each memory location. This includes byte 0, the retimer interprets this as "CRC Enabled". The retimer has no way to know that the EEPROM does not contain a valid configuration, so the retimer will try to read its configuration data from the EEPROM and will compute a checksum. It will compare this checksum to the checksum byte in the EEPROM (which will be 0xff), and the comparison will fail.

If the retimer is set to SMBus master mode, meaning the SMBUS_EN pin is floating, and if the $\overline{\text{READ\_EN}}$ pin is pulled low, the retimer will attempt to read its configuration from the EEPROM. If the EEPROM is not programmed, then, as described above, the retimer will attempt to compute a checksum and compare it to the data it reads in, and it will fail. When this happens the retimer will continue to hold the SMBus as it attempts to read a valid configuration from the EEPROM. The retimer will continue to try to read a valid configuration and will never set its $\overline{\text{ALL\_DONE}}$ pin low. This causes the SMBus to hang up and the retimer cannot be configured.

If an unprogrammed EEPROM is to be installed in the system, make sure that there is a provision for putting the retimer into SMBus slave mode for initial EEPROM programming. A jumper that can be installed to pull the SMBUS_EN pin to ground is recommended.

4.2.1.2 Byte 1

Byte 1 is reserved. This is not used by the retimer. The value of this byte is not important. Normally an unprogrammed EEPROM will have 0xff in all its memory locations. This byte can be set to something other than 0xff to flag that the EEPROM has been programmed.

4.2.1.3 Byte 2

Byte 2 is the maximum EEPROM burst size in bytes, from 0 to 255. Most EEPROMs will support a burst read operation. The Atmel AT24C01/2/4/8B family of EEPROMs, for example, will continue to present data from sequential memory locations as long as each byte is acknowledged and the master does not generate a STOP condition on the SMBus.

A value of 16 (0x10) in this byte will work for all supported EEPROMs and provides for fast reading of the configuration from the EEPROM.

4.2.2 Address Map Headers

The address map headers are only assumed by the retimer to be present if bit 6 of byte 0, the first byte of the base header, is set. If the address map headers are not present, then the register data slots are assumed by the retimer to start at EEPROM memory location 3.

4.2.2.1 Address Map Header Memory Locations

If the address map headers are present, as indicated by bit 6 of byte 0, then each retimer computes the starting memory location of its address map header (not its register data slot) as follows.

The size of each address map header, in bytes, is either 2 or 3, depending upon whether the EEPROM size is greater than 256 bytes, as indicated by bit 5 of byte 0. If the EEPROM size is less than or equal to 256 bytes, then each address map header is 2 bytes in length. If the EEPROM size is greater than 256 bytes, then each address map header is 3 bytes in length.

We will designate the length of the address map header, either 2 or 3, as N_{Addr_Map}.

Note that the actual EEPROM size need not match the value of bit 5 of byte 0. This byte just tells the retimer how big each address map header is and whether or not to use paged addressing for EEPROM addresses greater than 255. If the size does not match the setting of this bit, however, it is easy to see that the retimer might try to address non-existent memory locations and would therefore read nonsense data.

The retimer computes the starting memory location for its address map based upon its (the retimer's) SMBus address. We will designate the starting memory location for the address map for a retimer as $ADDR_{Map\_Start}$.

The retimer first determines its SMBus address index, I_{SMB_Addr}. This is the index into the array of permissible SMBus write addresses for the retimer. The retimer can be configured to use SMBus write addresses in the range of 0x30 to 0x4e. The indexing is straightforward.

The relationship between the retimer SMBus address, the SMBus address index, and the address map start memory location is shown in Table 5.

Table 5. Retimer SMBus Addresses, SMBus Address Indices, and Address Map Start Locations

Retimer SMBus Write Address	SMBus Address Index ISMB_Addr	Starting Address Map Memory Location ADDRMap_Start when EEPROM size ≤ 256	Starting Address Map Memory Location ADDRMap_Start when EEPROM size > 256
0x30	0	3	3
0x32	1	5	6
0x34	2	7	9

Table 5. Retimer SMBus Addresses, SMBus Address Indices, and Address Map Start Locations (continued)

Retimer SMBus Write Address	SMBus Address Index ISMB_Addr	Starting Address Map Memory Location ADDRMap_Start when EEPROM size ≤ 256	Starting Address Map Memory Location ADDRMap_Start when EEPROM size > 256
0x36	3	9	12
0x38	4	11	15
0x3a	5	13	18
0x3c	6	15	21
0x3e	7	17	24
0x40	8	19	27
0x42	9	21	30
0x44	10	23	33
0x46	11	25	36
0x48	12	27	39
0x4a	13	29	42
0x4c	14	31	45
0x4e	15	33	48

This table gives the fixed addresses in the EEPROM where the retimer will look for its address map depending upon the retimer's SMBus address and the size of the EEPROM. There are a few things to note about this operation.

1. If the address maps are not enabled, (bit 6 of byte 0 is 1'b0), then the retimer will not look for an address map. It will, instead, compute a starting address for its register data in the EEPROM and it will look there for its register data.
2. It is not necessary for all the address maps to be present. If the only retimer in the system reading from an EEPROM has an SMBus address of 0x30, for example, then only the first address map needs to be present in the EEPROM. EEPROM memory locations from 5 or 6 (depending upon the EEPROM size) to the end of the EEPROM memory space can be used for register data.
3. The address map locations are fixed for a given retimer SMBus address. For example, if the only retimer in the system has an SMBus address of 0x43, then it will still look for its address map data starting at memory location 33 or 48, depending upon the EEPROM size. In this case, the first EEPROM memory location that can be used for register data is 35 or 51. Here the data in memory locations 3-32 or 3-47 in the EEPROM are not used for address maps, but they cannot be used for register data, either.

4.2.2.2 Address Map Header Contents

If the address maps are present, they are each 2 or 3 bytes in length. The address maps start at the EEPROM memory locations shown in Table 5. If the EEPROM size is greater than 256 bytes the address map headers are 3 bytes long. If the EEPROM size is less than or equal to 256 bytes the address map headers are 2 bytes long.

The contents of each address map header are as shown in Table 6. Each address map header starts at the EEPROM memory location given by ADDRMap_Start for the retimer's SMBus address and spans either 2 or 3 bytes starting from there.

The first byte of the address map header is the Cyclic Redundancy Check (CRC) for this address map. The CRC is computed from all the bytes read by the retimer from the EEPROM except the CRC byte itself. The computation of the CRC, both within the retimer and by the external software, uses a standard algorithm, CRC-8. The computation of the CRC is covered in [insert cross reference].

The second byte of the address map header is the Least Significant Byte (LSB) of the EEPROM address that is the start of the register data for this address map header. The register data that begins at the start location in the address map should be valid register data. Otherwise the retimer will read from that memory location and will be configured incorrectly.

Table 6. Address Map Header Contents

EEPROM ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0
ADDRMap_Start	CRC[7]	CRC[6]	CRC[5]	CRC[4]	CRC[3]	CRC[2]	CRC[1]	CRC[0]
ADDRMap_Start + 1	EE ADDR LSB[7]	EE ADDR LSB[6]	EE ADDR LSB[5]	EE ADDR LSB[4]	EE ADDR LSB[3]	EE ADDR LSB[2]	EE ADDR LSB[1]	EE ADDR LSB[0]
ADDRMap_Start + 2						EE ADDR MSB[2] (If EEPROM > 256 bytes)	EE ADDR MSB[1] (If EEPROM > 256 bytes)	EE ADDR MSB[0] (If EEPROM > 256 bytes)

Table 7. Address Map Header Example

	0x_0	0x_1	0x_2	0x_3	0x_4	0x_5	0x_6	0x_7	0x_8	0x_9	0x_a	0x_b	0x_c	0x_d	0x_e	0x_f
0x00_	0xe0	0x00	0x10	0xeo	0x33	0x00	0xe0	0x33	0x00	0xa8	0x5d	0x01	0xe0	0x33	0x00	0x04
0x01_	0x87	0x02	0x04	0x87	0x02	0xa8	0x5d	0x01	0xa8	0x5d	0x01	0xa8	0x5d	0x01	0xeo	0x33
0x02_	0x00	0xe0	0x33	0x00	0xa8	0x5d	0x01	0xa8	0x5d	0x01	0xeo	0x33	0x00	0xeo	0x33	0x00
0X03_	0xa8	0x5d	0x01													

The third byte of the address map header, if the EEPROM size is greater than 256 bytes, is the Most Significant Byte (MSB) of the EEPROM address that is the start of the register data for this address map header. To compute the EEPROM address that is the start of the register data for a particular address map header, take the MSB from the address map header and left shift it by 8 bits, then add it to the LSB from the address map header.

Suppose the data in the first 51 bytes of the EEPROM is as shown in Table 7. In this table, the EEPROM memory address is given by the two most significant hex digits in the left-hand column and the least significant hex digit in the top row. For example, in this table, the contents of EEPROM memory location 0 is 0xe0. Here's what that means.

Remember that byte 0 of the EEPROM is the first byte of the base header. Since the value is 0xe0, bits 7, 6, and 5 are set, and all the rest of the bits in this byte are cleared. Referring to Table 4, we see that this means that the CRCs are enabled, the address map headers are present, and the EEPROM is larger than 256 bytes. The device count in the lower four bits is set to 0, but this is unused. The common channel bit is not set, meaning that the EEPROM contains information to configure each channel of each retimer separately.

The second byte of the base header, at EEPROM memory address 1, contains 0x00. Remember that this is not used, but since it contains 0x00 we can be confident that this EEPROM has at least been partially programmed. If the EEPROM were not programmed, this byte would probably be 0xff.

The third byte of the base header, at EEPROM memory address 2, contains 0x10, or decimal 16. This is the burst count. When the retimer reads from the EEPROM, it will not attempt to read more than 16 bytes in a single burst.

Since the EEPROM size is greater than 256 bytes (because bit 5 of byte 0 is set), each address map is three bytes long. So, starting with byte 0x003, consider the data in Table 7 in groups of three bytes each. The first such group is given by [0xe0, 0x33, 0x00].

This group of three bytes is the first address map in the EEPROM. This is the address map that will be read at power up by a retimer with address 0x30, if there is one in the system.

The first byte of this address map is 0xe0. This is the CRC for this address map. When the retimer reads the data from the EEPROM, it will independently compute the CRC for the data it reads. The CRC computed by the retimer must match the CRC in the EEPROM or the data read from the EEPROM will be ignored. The CRCs are enabled because bit 7 of byte 0 is set. If this bit were cleared, the CRC values would be ignored by the retimer, and the retimer would read the data from the EEPROM and use it no matter what value was contained in the first byte of the address map.

The second byte of this address map is 0x33 and the third byte is 0x00. Taken together, these two bytes yield the starting address for the register data for this address map. The starting address for the data is 0x033, or decimal 51. Note that this is the first byte in the EEPROM following all the address map data in Table 7. The data does not have to be arranged this way, but this is the most efficient use of the data space in the EEPROM.

The set of register data pointed to by this address map begins at EEPROM address 0x33, or decimal 51. Let's introduce some nomenclature at this point. Let's call a set of register data in the EEPROM a "slot". This terminology is easy to visualize. Consider writing down a set of register configuration data in a book. We can do this for several different register configurations, producing several different books. Obviously we only need one book for each register configuration no matter how many times we might reuse that configuration.

Now take the books with register configurations that we have written down and insert them into various locations, or "slots", in a bookshelf. This illustrates what we mean by "slots". To refer to a given register configuration, written down in one of these books, we have only to indicate its position, or "slot", on the bookshelf. We can make a list that tells each retimer in the system which "slot" to look in for its register configuration. This list is the address maps, and the register data corresponds to the books in the "slots".

Using this nomenclature, the data in Table 7 tells the retimer with SMBus address 0x30 to look in a slot beginning at EEPROM address 0x33 for its configuration data.

The next set of three bytes, that is, the next address map, begins at EEPROM address 0x006. This second group of three bytes is the same as the first one, [0xe0, 0x33, 0x00]. This second set of three bytes tells the retimer with SMBus address 0x32 to look in a slot beginning at EEPROM address 0x33 for its configuration data. That means that the retimer with address 0x32 will be configured exactly the same as the retimer with address 0x30.

Note that the CRCs for these two address maps are the same, 0xe0. The retimers with SMBus addresses 0x30 and 0x32 will read exactly the same set of data from the EEPROM. They will read this data from different locations because their address map locations are different, but the contents of the two address maps are the same. And the two address maps point to the same EEPROM slot. So these two retimers will compute the same CRC when they read the data from the EEPROM, and so the comparison values in the first bytes of their address maps are the same.

Now the next set of three bytes, beginning at EEPROM address 0x009, is different. These three memory locations contain [0xa8, 0x5d, 0x01]. These three bytes make up the address map for the retimer with SMBus address 0x36.

The CRC for this retimer is 0xa8, which is different from the value of 0xe0 in the first two address maps. This is because the retimer with address 0x36 will read different bytes from the EEPROM than the first two retimers. At the very least, the starting address for the EEPROM slot that contains the configuration data for this retimer is different, and that is part of the data the retimer reads from the EEPROM. Presumably at least some of the register contents are also different, or else we would probably use the same EEPROM slot to program this retimer as we used to program the first two.

The starting address for the data slot for this retimer is given by the second and third bytes of the address map. These bytes indicate a starting address in the EEPROM of 0x15d, or 349 decimal. Note that this is the starting address of the first data slot, 51, plus the length of a register configuration when the common channel bit is not set, 298. Again, it is not necessary to start the second data slot immediately after the first in this way, but this is the most efficient way to use the EEPROM memory space.

It would not be a good idea to start the second data slot before the end of the first one. The retimer has no way to know about the overlap, so it would happily read its configuration data starting from whatever EEPROM memory address was contained in its address map, and it is unlikely that this would produce the desired configurations in the retimers. In practice, a set of address maps like those shown in Table 7 yield the most efficient use of the EEPROM memory space.

The next set of three bytes is the address map for the retimer with SMBus address 0x38. This address map is the same as the first two. The retimer with SMBus address 0x38 will be configured the same as the retimers with SMBus addresses 0x30 and 0x32.

This address mapping scheme in the EEPROM allows for maximum flexibility in configuring the retimers. A retimer with any SMBus address can read its configuration from any data slot in the EEPROM.

The next set of three bytes, beginning at address 0x00f, is different from any we have yet seen. This set of three bytes is [0x04, 0x87, 0x02]. This is the address map for the retimer with SMBus address 0x3a.

The first byte in this address map, 0x04, is the CRC for this retimer. It is different from the other CRCs because the data this retimer will read from the EEPROM is different.

The second two bytes in this address map are the starting address of the EEPROM data slot. These two bytes give us a starting address of 0x287 or 647 decimal. Note that this is the starting address of the previous slot, 349, plus the length of the register configuration, 298. Once again, this is the most efficient way to arrange the data in the EEPROM.

Note that there is not room for another data slot even if the EEPROM size is 1 Kbyte. The next slot would have to start at address 945 decimal, and there is not enough room left in the EEPROM after that address for another 298-byte register configuration. So the maximum number of EEPROM slots available in the largest supported EEPROM is 3 if the retimer channels are to be set up differently. If the retimer channels are to be set up identically, indicated by setting bit 4 of byte 0 in the EEPROM, the common channel bit, then there is enough room in a 1 Kbyte EEPROM for 12 data slots.

Looking at the rest of Table 7, we can see that the rest of the address map headers all point to one of the three data slots already referenced. With this EEPROM header information we can configure up to 16 retimers, each with a unique SMBus address. There will be only three different retimer configurations applied to these 16 retimers, however.

4.2.2.3 EEPROM Configuration without Address Map Headers

At this point it should be clear how the address map headers work. If bit 6 of byte 0 in the EEPROM is cleared, then the address map headers are not used. Instead, each retimer computes a unique start address in the EEPROM for its data slot.

First the retimer computes the length of the data slot, N_{Data_Slot}. This is based on COMMON_CHAN bit in byte 0 of the EEPROM. The length of the data slot is given by the following truth Table 8.

Table 8. EEPROM Data Slot Size N_{Data_Slot} versus EEPROM Byte 0 Bits

CRD_EN / COMMON_CHAN	0	1
0	299	299
1	77	77

If the CRC_EN bit is set, then the CRC for the data slot is the last byte of the register data slot. Note that when the address maps are present, the CRC is in the address map. Even if the CRCs are not enabled, the CRC byte is assumed to be present. It is just not used if the CRCs are not enabled.

Once the retimer has computed the data slot size, it computes a unique data slot start address based upon its SMBus address index, I_{SMB_Addr}, as shown in Table 5. Each retimer computes its data slot start address, $ADDR_{Data\_Start}$, as follows.

$$ADDR_{Data\_Start} = 3 + (I_{SMB\_Addr} \times N_{Data\_Slot})$$

For example, if the COMMON_CHAN bit is set, then $N_{Data\_Slot} = 77$. A retimer with SMBus address 0x34 has an SMBus address index, I_{SMB_Addr}, of 3. This retimer would compute its data slot start address $ADDR_{Data\_Start}$ as follows.

$$ADDR_{Data\_Start} = 3 + (3 \times 77) = 234$$

This is the EEPROM memory location where the retimer would begin looking for its configuration data. It would compare the CRC it computed with the CRC byte it finds at memory location 311. If the CRC_EN bit is not set, then the contents of this memory location are ignored.

Note that since the EEPROM can contain only three data slots if the COMMON_CHAN bit is not set, only the retimers at addresses 0x30, 0x32, and 0x34 can be configured with a 1 Kbyte EEPROM if the address maps are not used. If the COMMON_CHAN bit is set, then the EEPROM can contain as many as 12 data slots. In this case, retimers with addresses 0x30 to 0x46 can be configured with a 1 Kbyte EEPROM if the address maps are not used. Retimers with SMBus write addresses of 0x48 to 0x4e cannot be configured from the EEPROM if the address maps are not used.

4.2.3 Register Data Slots

We have so far described how the retimer knows where to find its configuration data. We will now describe what the retimer configuration data consists of and how it is organized.

4.2.3.1 Bit Mapping of the Register Data

The operation of a DS110DF410 quad retimer can be customized for specific applications by changing some of the default operational parameters of the device. This is accomplished by writing desired values into registers in the device over the SMBus.

When the DS110DF410 quad retimer is configured for SMBus slave mode operation, the system controller writes data into the retimer's registers by sending a register address, which is one byte, followed by a byte of register data. This sets all the bits of a one-byte register in the retimer. To save area and power in the retimer, some of the registers are arranged in bit-fields, which may not be related to one another. Some of the bits in some registers configure one operational parameter while other bits in the same register configure another.

Often the user desires to configure some of the bits in a register while leaving the rest at their current values. When this is required, the normal procedure is to read the entire register over the SMBus, change only the bits that are required, and then write the entire register back to the retimer over the SMBus. The retimer can be configured as desired in SMBus slave mode using this procedure.

In SMBus master mode, a similar configuration can be achieved. In SMBus master mode, the retimer reads its configuration autonomously from an external EEPROM. "Reading its configuration" really means reading and setting the contents of some of the registers in the retimer, just as is done in SMBus slave mode. In SMBus master mode the registers to be set are defined in advance, and their contents are read from the EEPROM prior to beginning any mission-mode operation of the retimer.

Not all of the registers in the retimer are configured from the EEPROM in SMBus master mode. For example, some of the register bits and bit fields in the retimer are read-only. They report the status of various circuit blocks within the retimer. It would not make sense to try to set the values of these bits, as the retimer would simply ignore this.

Some of the register bits and bit fields in the retimer are reserved for test and troubleshooting and should not be changed by the user under normal conditions. These bits are not included in the EEPROM register set because they should be left at their default values for almost all applications.

Since only a subset of the retimer register bits are to be configured from the EEPROM, the EEPROM register data set is designed to configure these bits as efficiently as possible. This means that there is not a one-to-one mapping of retimer registers to EEPROM memory locations. Register bits in the retimer which are sequential are also sequential in the EEPROM. To the extent possible, register bits that are contiguous in the retimer are also contiguous in the EEPROM. However, the register bits are packed into the EEPROM in the minimum possible space, which means that register bits that are located in the same register in the retimer may not be located in the same register in the EEPROM, although they will always be in the same sequence.

4.2.3.2 Register Data Slot Organization

The organization of the register data within an EEPROM register data slot can be described generically as follows:

1. Slot_Start
2. Channel_0_Data
3. Optional_Channel_1_Data
4. Optional_Channel_2_Data
5. Optional_Channel_3_Data
6. Shared_Register_Data
7. CRC_Byte

In this description, Slot_Start simply refers to the EEPROM memory address where the register data slot begins. The first register data slot begins at a Slot_Start address of 0x33, or decimal 51.

Channel_0_Data is a set of EEPROM memory locations that configure the channel registers for channel 0 of the retimer. The contents of any channel register set are shown in Table 9.

The channel data sets for the other channels are optional because if the COMMON_CHAN bit is set, these channel data sets are not needed and are not present. If the COMMON_CHAN bit is not set, all four sets of channel data are present and are used. The structure of all the channel data sets is the same, although the register contents may be different.

The Shared_Register_Data set is a two-byte field that configures some parameters in the retimer's shared register set. These two bytes are always present. They either occur after the first set of channel register data if the COMMON_CHAN bit is set, or after the fourth set of channel register data if the COMMON_CHAN bit is not set.

The CRC byte is always present when the address maps are not enabled, but the CRCs may or may not be enabled. However, even if the CRCs are enabled, the CRC is contained in the address map header when it is used. So this byte will often not be present.

Each channel register data set is 74 bytes long. The contents of the channel register data set are described in Table 9.

The first column in Table 9, "Address", contains the offset in hexadecimal from the beginning of the data for the channel under consideration. For channel 0, this is the offset from the beginning of the EEPROM data slot. For the other channels, this is modified by the length of the channel register sets that appear before this one. That is, for channel 1 this is the offset from the beginning of the EEPROM data slot plus the length of the channel 0 register set, 74 bytes.

The second column contains a simple name for the EEPROM channel register. The only real information in this name is the address offset in decimal, corresponding to the address offset in hexadecimal in column 1.

The third column designates which bits in the EEPROM channel register are described by the "Field" name in column 4 and the "Default Value" in column 5.

The "Field" column is the most descriptive. For each bit or bit field in the EEPROM channel register set, the "Field" column contains the address of the retimer channel register which the bit or bit field configures, the bit indices that are configured, and a descriptive string that describes what the retimer channel register bit or bit field configures in the retimer.

As an example, consider the first register in the channel register set at address offset 0x00, called CFGBYT_0_BITDESC. This register consists of four bit fields called reg_03[b+1:b]_eq_BSTN[1:0]. All four of these bit fields configure bits in retimer channel register 0x03. For each bit field, the starting bit in retimer channel register 0x03 is b+1 and the ending bit is b. Each bit field configures one of the CTLE boost stages in the retimer, boost stage N. Each boost stage is configured by a two-bit value, so the description of the retimer channel register targeted by each bit field is eq_BSTN[1:0].

For this register, there is a one-to-one correspondence between the EEPROM memory location bits and the retimer register bits. The memory location at offset 0 in the EEPROM (which is a different EEPROM memory location for each data slot start and retimer channel) is read directly into register 0x03 for the corresponding channel of the retimer. Referring to the retimer data sheet, we see that the description for channel register 0x03 is as shown in Table 10.

As an example of an EEPROM memory location that does not contain a one-to-one configuration for a retimer register, consider the second location in the channel register set at address offset 0x01, called CFGBYT_1_BITDESC. This register consists of a bit field called reg_08[4:0]_cdr_cap_dac_start0[4:0] and three individual bits. The first bit field maps to retimer channel register 0x08, bits 4:0. Note that in the EEPROM memory location, this bit field is contained in bits 7:3. When the retimer reads its configuration from the EEPROM, bits 7:3 in this memory location are read into bits 4:0 in retimer channel register 0x08.

The three individual bits all map to retimer register 0x09. Bits 2, 1, and 0 of the EEPROM memory location at offset 1 map to bits 7, 6, and 5 of register 0x09 in the retimer. These bits all configure different settings in the retimer.

Finally, consider an example where a bit field in the retimer channel register set is broken across two memory locations in the EEPROM. The byte at EEPROM offset 3, called CFGBYT_3_BITDESC, contains 5 individual bits that configure retimer channel register 0x0a, bits 4, 3, 2, 1, and 0. These bits all configure independent settings in the retimer.

Bits 2:0 of this EEPROM memory location map to retimer channel register 0x0b, bits 4:2. These are the three most-significant bits of channel register bit field reg_0B[4:0]_cdr_cap_dac_start1[4:0]. Only the bits reg_0B[4:2]_cdr_cap_dac_start1[4:2] are contained in the EEPROM memory location at offset 3.

The next byte in the EEPROM memory map at offset 4, called CFGBYT_3_BITDESC, contains the remainder of this bit field. Bits 7:6 of this EEPROM memory location contain bits 1:0 of this bit field, referred to as reg_0B[1:0]_cdr_cap_dac_start1[1:0]. When the retimer reads its configuration from the EEPROM, bits 4:2 of retimer channel register 0x0b are set from bits 2:0 of the EEPROM memory location at offset 3 and bits 1:0 of retimer channel register 0x0b are set from bits 7:6 of the EEPROM memory location at offset 4.

Table 9. Channel Register Data Set

Address	Register Name	Bit(s)	Default Value	Field
0x00	CFGBYT_0_BITDESC	7:6	0x0	reg_03[7:6]_eq_BST0[1:0]
		5:4	0x0	reg_03[5:4]_eq_BST1[1:0]
		3:2	0x0	reg_03[3:2]_eq_BST2[1:0]
		1:0	0x0	reg_03[1:0]_eq_BST3[1:0]
0x01	CFGBYT_1_BITDESC	7:3	0x0	reg_08[4:0]_cdr_cap_dac_start0[4:0]
		2	0x0	reg_09[7]_reg_divsel_vco_cap_ov
		1	0x0	reg_09[6]_reg_set_cp_lvl_lpf_ov
		0	0x0	reg_09[5]_reg_bypass_pfd_ov
0x02	CFGBYT_2_BITDESC	7	0x0	reg_09[4]_reg_en_fd_pd_vco_pdiq_ov
		6	0x0	reg_09[3]_reg_en_pd_cp_ov
		5	0x0	reg_09[2]_reg_divsel_ov
		4	0x0	reg_09[1]_reg_en_fld_ov
		3	0x0	reg_09[0]_reg_pfd_lock_mode_sm
		2	0x0	reg_0A[7]_reg_sbt_en
		1	0x0	reg_0A[6]_reg_en_idac_pd_cp_ov_AND_reg_en_idac_fd_cp_ov
		0	0x0	reg_0A[5]_reg_dac_lpf_high_phase_ov_AND_reg_dac_lpf_low_phase_ov
0x03	CFGBYT_3_BITDESC	7	0x1	reg_0A[4]_reg_en150_lpf_ov
		6	0x0	reg_0A[3]_reg_cdr_reset_ov
		5	0x0	reg_0A[2]_reg_cdr_reset_sm
		4	0x0	reg_0A[1]_reg_cdr_lock_ov
		3	0x0	reg_0A[0]_reg_cdr_lock
		2:0	0x3	reg_0B[4:2]_cdr_cap_dac_start1[4:2]
0x04	CFGBYT_4_BITDESC	7:6	0x3	reg_0B[1:0]_cdr_cap_dac_start1[1:0]
		5	0x0	reg_0C[2]_reg_EN_FORCE_EXCEPTION_FSM
		4	0x0	reg_0D[5]_PRBS_PATT_SHIFT_EN
		3:2	0x2	reg_0E[7:6]_reg_timer_1ms_sm[1:0]
		1:0	0x1	reg_0E[5:4]_reg_timer_600us_sm[2:1]
0x05	CFGBYT_5_BITDESC	7	0x0	reg_0E[3]_reg_timer_600us_sm[0]
		6:4	0x3	reg_0E[2:0]_reg_timer_200us_sm[2:0]
		3:1	0x3	reg_0F[7:5]_reg_timer_charge_lpf[11:9]
		0	0x0	reg_0F[4]_timer_cdr_unlock[2]
0x06	CFGBYT_6_BITDESC	7:6	0x2	reg_0F[3:2]_timer_cdr_unlock[1:0]
		5:4	0x1	reg_0F[1:0]_reg_timer_lock_check[1:0]
		3:1	0x1	reg_10[7:5]_false_lock_detector_threshold[2:0]
		0	0x1	reg_10[4]_reg_hd_threshold_sm[4]
0x07	CFGBYT_7_BITDESC	7:4	0xA	reg_10[3:0]_reg_hd_threshold_sm[3:0]
		3:2	0x0	reg_11[7:6]_eom_sel_vrange[1:0]

Table 9. Channel Register Data Set (continued)

Address	Register Name	Bit(s)	Default Value	Field
		1	0x1	reg_11[5]_eom_PD
		0	0x0	reg_11[3]_dfe_tap2_pol
0x08	CFGBYT_8_BITDESC	7	0x0	reg_11[2]_dfe_tap3_pol
		6	0x0	reg_11[1]_dfe_tap4_pol
		5	0x0	reg_11[0]_dfe_tap5_pol
		4	0x0	reg_12[7]_dfe_tap1_pol
		3	0x1	reg_12[5]_dfe_sel_neg_gm
		2:0	0x0	reg_12[4:2]_dfe_wt1[4:2]
0x09	CFGBYT_9_BITDESC	7:6	0x0	reg_12[1:0]_dfe_wt1[1:0]
		5	0x0	reg_13[6]_eq_PD_SD
		4	0x1	reg_13[5]_eq_mute_z
		3	0x1	reg_13[4]_eq_en_dc_off
		2	0x0	reg_13[3]_eq_PD_EQ
		1	0x0	reg_13[2]_eq_BST3[2]
		0	0x0	reg_13[1]_eq_pd_cm
0x0A	CFGBYT_10_BITDESC	7	0x0	reg_13[0]_reg_vco_bypass
		6	0x0	reg_14[7]_eq_sd_preset
		5	0x0	reg_14[6]_eq_sd_reset
		4:3	0x0	reg_14[5:4]_eq_refa_sel[1:0]
		2:1	0x0	reg_14[3:2]_eq_refd_sel[1:0]
		0	0x0	reg_15[7]_dfe_force_enable
0x0B	CFGBYT_11_BITDESC	7	0x0	reg_15[6]_drv_dem_range
		6	0x1	reg_15[5]_comp_en_hyst
		5	0x1	reg_15[4]_comp_en
		4	0x0	reg_15[3]_drv_PD
		3:1	0x0	reg_15[2:0]_drv_dem[2:0]
		0	0x0	reg_16[7]_reg_dac_lpf_high_phase[3]
0x0C	CFGBYT_12_BITDESC	7:5	0x7	reg_16[6:4]_reg_dac_lpf_high_phase[2:0]
		4:1	0xA	reg_16[3:0]_reg_dac_lpf_low_phase[3:0]
		0	0x0	reg_17[7]_reg_dac_lpf_high_lock[3]
0x0D	CFGBYT_13_BITDESC	7:5	0x3	reg_17[6:4]_reg_dac_lpf_high_lock[2:0]
		4:1	0x6	reg_17[3:0]_reg_dac_lpf_low_lock[3:0]
		0	0x1	reg_18[6]_pdiq_sel_div[2]
0x0E	CFGBYT_14_BITDESC	7:6	0x0	reg_18[5:4]_pdiq_sel_div[1:0]
		5:0	0x23	reg_19[5:0]_bg_sel_ptat[5:0]
0x0F	CFGBYT_15_BITDESC	7:6	0x0	reg_1A[7:6]_bg_sel_rph[1:0]
		5:4	0x0	reg_1A[5:4]_bg_sel_rpp[1:0]
		3	0x1	reg_1B[1]_cp_en_cp_pd
		2	0x1	reg_1B[0]_cp_en_cp_fd
		1:0	0x0	reg_1C[7:6]_cp_en_idac_pd[2:1]
0x10	CFGBYT_16_BITDESC	7	0x1	reg_1C[5]_cp_en_idac_pd[0]
		6:4	0x1	reg_1C[4:2]_cp_en_idac_fd[2:0]
		3	0x0	reg_1C[1]_pdiq_PD
		2	0x0	reg_1C[0]_vco_PD
		1	0x0	reg_1D[7]_sbt_en
		0	0x1	reg_1E[7]_pfd_sel_data_mux[2]
0x11	CFGBYT_17_BITDESC	7:6	0x3	reg_1E[6:5]_pfd_sel_data_mux[1:0]

Table 9. Channel Register Data Set (continued)

Address	Register Name	Bit(s)	Default Value	Field
		5	0x0	reg_1E[3]_dfe_PD
		4	0x0	reg_1E[2]_pfd_PD_pd
		3	0x0	reg_1E[1]_pfd_EN_fld
		2	0x1	reg_1E[0]_pfd_en_fd
		1	0x0	reg_1F[7]_drv_sel_inv
		0	0x1	reg_1F[6]_lpf_en[150]
0x12	CFGBYT_18_BITDESC	7:4	0x0	reg_20[7:4]_dfe_wt5[3:0]
		3:0	0x0	reg_20[3:0]_dfe_wt4[3:0]
0x13	CFGBYT_19_BITDESC	7:4	0x0	reg_21[7:4]_dfe_wt3[3:0]
		3:0	0x0	reg_21[3:0]_dfe_wt2[3:0]
0x14	CFGBYT_20_BITDESC	7	0x0	reg_22[7]_eom_ov
		6	0x0	reg_22[6]_SPARE
		5	0x0	reg_23[7]_eo_get_heo_veo_ov
		4	0x1	reg_23[6]_dfe_ov
		3:0	0x3	reg_2A[7:4]_eom_timer_thr[7:4]
0x15	CFGBYT_21_BITDESC	7:4	0x0	reg_2A[3:0]_eom_timer_thr[3:0]
		3:2	0x0	reg_2B[5:4]_reg_timer_10ms_sm[1:0]
		1:0	0x0	reg_2B[3:2]_eom_min_req_hits[3:2]
0x16	CFGBYT_22_BITDESC	7:6	0x0	reg_2B[1:0]_eom_min_req_hits[1:0]
		5	0x1	reg_2C[6]_veo_scale
		4:3	0x3	reg_2C[5:4]_dfe_sm_fom[1:0]
		2:0	0x1	reg_2C[3:1]_dfe_adapt_counter[3:1]
0x17	CFGBYT_23_BITDESC	7	0x0	reg_2C[0]_dfe_adapt_counter[0]
		6	0x1	reg_2D[7]_drv_sel_scp
		5	0x0	reg_2D[6]_sd_en_fast_oob
		4	0x0	reg_2D[5]_sd_ref_high
		3	0x0	reg_2D[4]_sd_gain
		2	0x0	reg_2D[3]_reg_eq_bst_ov
		1:0	0x0	reg_2D[2:1]_drv_sel_vod[2:1]
0x18	CFGBYT_24_BITDESC	7	0x0	reg_2D[0]_drv_sel_vod[0]
		6	0x0	reg_2E[5]_reg_vod_ov
		5	0x0	reg_2E[2]_reg_dem_ov
		4:3	0x0	reg_2F[7:6]_RATE[1:0]
		2:1	0x0	reg_2F[5:4]_SUBRATE[1:0]
		0	0x0	reg_2F[3]_index_ov
0x19	CFGBYT_25_BITDESC	7	0x1	reg_2F[2]_en_ppm_check
		6	0x1	reg_2F[1]_en_fld_check
		5	0x0	reg_30[3]_prbs_en_dig_clk
		4:3	0x0	reg_30[1:0]_prbs_pattern_sel[1:0]
		2	0x0	reg_31[7]_eq_dfe_sm
		1:0	0x1	reg_31[6:5]_adapt_mode[1:0]
0x1A	CFGBYT_26_BITDESC	7:6	0x0	reg_31[4:3]_eq_sm_fom[1:0]
		5:2	0x1	reg_32[7:4]_heo_int_thresh[3:0]
		1:0	0x0	reg_32[3:2]_veo_int_thresh[3:2]
0x1B	CFGBYT_27_BITDESC		0x1	reg_32[1:0]_veo_int_thresh[1:0]
			0x8	reg_33[7:4]_heo_thresh[3:0]
			0x2	reg_33[3:2]_veo_thresh[3:2]

Table 9. Channel Register Data Set (continued)

Address	Register Name	Bit(s)	Default Value	Field
0x1C	CFGBYT_28_BITDESC	7:6	0x0	reg_33[1:0]_veo_thresh[1:0]
		5	0x0	reg_34[6]_low_power_mode_disable
		4:3	0x3	reg_34[5:4]_lock_counter[1:0]
		2:0	0x7	reg_34[3:1]_dfe_max_tap_2_5[3:1]
0x1D	CFGBYT_29_BITDESC	7	0x1	reg_34[0]_dfe_max_tap_2_5[0]
		6:5	0x0	reg_35[7:6]_data_lock_ppm[1:0]
		4	0x0	reg_35[5]_get_ppm_error
		3:0	0xF	reg_35[4:1]_dfe_max_tap_1[4:1]
0x1E	CFGBYT_30_BITDESC	7	0x1	reg_35[0]_dfe_max_tap_1[0]
		6	0x0	reg_36[7]_enable_manual_adaptation
		5	0x0	reg_36[6]_heo_veo_int_enable
		4:3	0x0	reg_36[5:4]_ref_mode[1:0]
		2	0x0	reg_36[2]_mr_cdr_cap_dac_rng_ov
		1:0	0x1	reg_36[1:0]_mr_cdr_cap_dac_rng[1:0]
0x1F	CFGBYT_31_BITDESC	7:6	0x0	reg_39[6:5]_mr_eom_rate[1:0]
		5:1	0x0	reg_39[6:5]_mr_eom_rate[1:0]
		0	0x1	reg_3A[7]_fixed_eq_BST0[1]
0x20	CFGBYT_32_BITDESC	7	0x0	reg_3A[6]_fixed_eq_BST0[0]
		6:5	0x2	reg_3A[5:4]_fixed_eq_BST1[1:0]
		4:3	0x1	reg_3A[3:2]_fixed_eq_BST2[1:0]
		2:1	0x1	reg_3A[1:0]_fixed_eq_BST3[1:0]
		0	0x0	reg_3D[7]_SPARE
0x21	CFGBYT_33_BITDESC	7	0x1	reg_3E[7]_HEO_VEO_LOCKMON_EN
		6	0x0	reg_3F[7]_SPARE
		5:4	0x0	reg_40[7:6]_EQ_array_index_0_BST0[1:0]
		3:2	0x0	reg_40[5:4]_EQ_array_index_0_BST1[1:0]
		1:0	0x0	reg_40[3:2]_EQ_array_index_0_BST2[1:0]
0x22	CFGBYT_34_BITDESC	7:6	0x0	reg_40[1:0]_EQ_array_index_0_BST3[1:0]
		5:4	0x0	reg_41[7:6]_EQ_array_index_1_BST0[1:0]
		3:2	0x0	reg_41[5:4]_EQ_array_index_1_BST1[1:0]
		1:0	0x0	reg_41[3:2]_EQ_array_index_1_BST2[1:0]
0x23	CFGBYT_35_BITDESC	7:6	0x1	reg_41[1:0]_EQ_array_index_1_BST3[1:0]
		5:4	0x0	reg_42[7:6]_EQ_array_index_2_BST0[1:0]
		3:2	0x0	reg_42[5:4]_EQ_array_index_2_BST1[1:0]
		1:0	0x1	reg_42[3:2]_EQ_array_index_2_BST2[1:0]
0x24	CFGBYT_36_BITDESC	7:6	0x0	reg_42[1:0]_EQ_array_index_2_BST3[1:0]
		5:4	0x0	reg_43[7:6]_EQ_array_index_3_BST0[1:0]
		3:2	0x1	reg_43[5:4]_EQ_array_index_3_BST1[1:0]
		1:0	0x0	reg_43[3:2]_EQ_array_index_3_BST2[1:0]
0x25	CFGBYT_37_BITDESC	7:6	0x0	reg_43[1:0]_EQ_array_index_3_BST3[1:0]
		5:4	0x1	reg_44[7:6]_EQ_array_index_4_BST0[1:0]
		3:2	0x0	reg_44[5:4]_EQ_array_index_4_BST1[1:0]
		1:0	0x0	reg_44[3:2]_EQ_array_index_4_BST2[1:0]
0x26	CFGBYT_38_BITDESC	7:6	0x0	reg_44[1:0]_EQ_array_index_4_BST3[1:0]
		5:4	0x0	reg_45[7:6]_EQ_array_index_5_BST0[1:0]
		3:2	0x0	reg_45[5:4]_EQ_array_index_5_BST1[1:0]
		1:0	0x2	reg_45[3:2]_EQ_array_index_5_BST2[1:0]

Table 9. Channel Register Data Set (continued)

Address	Register Name	Bit(s)	Default Value	Field
0x27	CFGBYT_39_BITDESC	7:6	0x0	reg_45[1:0]_EQ_array_index_5_BST3[1:0]
		5:4	0x0	reg_46[7:6]_EQ_array_index_6_BST0[1:0]
		3:2	0x0	reg_46[5:4]_EQ_array_index_6_BST1[1:0]
		1:0	0x0	reg_46[3:2]_EQ_array_index_6_BST2[1:0]
0x28	CFGBYT_40_BITDESC	7:6	0x2	reg_46[1:0]_EQ_array_index_6_BST3[1:0]
		5:4	0x2	reg_47[7:6]_EQ_array_index_7_BST0[1:0]
		3:2	0x0	reg_47[5:4]_EQ_array_index_7_BST1[1:0]
		1:0	0x0	reg_47[3:2]_EQ_array_index_7_BST2[1:0]
0x29	CFGBYT_41_BITDESC	7:6	0x0	reg_47[1:0]_EQ_array_index_7_BST3[1:0]
		5:4	0x0	reg_48[7:6]_EQ_array_index_8_BST0[1:0]
		3:2	0x0	reg_48[5:4]_EQ_array_index_8_BST1[1:0]
		1:0	0x0	reg_48[3:2]_EQ_array_index_8_BST2[1:0]
0x2A	CFGBYT_42_BITDESC	7:6	0x3	reg_48[1:0]_EQ_array_index_8_BST3[1:0]
		5:4	0x0	reg_49[7:6]_EQ_array_index_9_BST0[1:0]
		3:2	0x0	reg_49[5:4]_EQ_array_index_9_BST1[1:0]
		1:0	0x3	reg_49[3:2]_EQ_array_index_9_BST2[1:0]
0x2B	CFGBYT_43_BITDESC	7:6	0x0	reg_49[1:0]_EQ_array_index_9_BST3[1:0]
		5:4	0x0	reg_4A[7:6]_EQ_array_index_10_BST0[1:0]
		3:2	0x3	reg_4A[5:4]_EQ_array_index_10_BST1[1:0]
		1:0	0x0	reg_4A[3:2]_EQ_array_index_10_BST2[1:0]
0x2C	CFGBYT_44_BITDESC	7:6	0x0	reg_4A[1:0]_EQ_array_index_10_BST3[1:0]
		5:4	0x1	reg_4B[7:6]_EQ_array_index_11_BST0[1:0]
		3:2	0x0	reg_4B[5:4]_EQ_array_index_11_BST1[1:0]
		1:0	0x0	reg_4B[3:2]_EQ_array_index_11_BST2[1:0]
0x2D	CFGBYT_45_BITDESC	7:6	0x1	reg_4B[1:0]_EQ_array_index_11_BST3[1:0]
		5:4	0x1	reg_4C[7:6]_EQ_array_index_12_BST0[1:0]
		3:2	0x1	reg_4C[5:4]_EQ_array_index_12_BST1[1:0]
		1:0	0x0	reg_4C[3:2]_EQ_array_index_12_BST2[1:0]
0x2E	CFGBYT_46_BITDESC	7:6	0x0	reg_4C[1:0]_EQ_array_index_12_BST3[1:0]
		5:4	0x3	reg_4D[7:6]_EQ_array_index_13_BST0[1:0]
		3:2	0x0	reg_4D[5:4]_EQ_array_index_13_BST1[1:0]
		1:0	0x0	reg_4D[3:2]_EQ_array_index_13_BST2[1:0]
0x2F	CFGBYT_47_BITDESC	7:6	0x0	reg_4D[1:0]_EQ_array_index_13_BST3[1:0]
		5:4	0x1	reg_4E[7:6]_EQ_array_index_14_BST0[1:0]
		3:2	0x2	reg_4E[5:4]_EQ_array_index_14_BST1[1:0]
		1:0	0x0	reg_4E[3:2]_EQ_array_index_14_BST2[1:0]
0x30	CFGBYT_48_BITDESC	7:6	0x0	reg_4E[1:0]_EQ_array_index_14_BST3[1:0]
		5:4	0x2	reg_4F[7:6]_EQ_array_index_15_BST0[1:0]
		3:2	0x1	reg_4F[5:4]_EQ_array_index_15_BST1[1:0]
		1:0	0x0	reg_4F[3:2]_EQ_array_index_15_BST2[1:0]
0x31	CFGBYT_49_BITDESC	7:6	0x0	reg_4F[1:0]_EQ_array_index_15_BST3[1:0]
		5:4	0x2	reg_50[7:6]_EQ_array_index_16_BST0[1:0]
		3:2	0x0	reg_50[5:4]_EQ_array_index_16_BST1[1:0]
		1:0	0x2	reg_50[3:2]_EQ_array_index_16_BST2[1:0]
0x32	CFGBYT_50_BITDESC	7:6	0x0	reg_50[1:0]_EQ_array_index_16_BST3[1:0]
		5:4	0x2	reg_51[7:6]_EQ_array_index_17_BST0[1:0]
		3:2	0x0	reg_51[5:4]_EQ_array_index_17_BST1[1:0]

Table 9. Channel Register Data Set (continued)

Address	Register Name	Bit(s)	Default Value	Field
		1:0	0x0	reg_51[3:2]_EQ_array_index_17_BST2[1:0]
0x33	CFGBYT_51_BITDESC	7:6	0x2	reg_51[1:0]_EQ_array_index_17_BST3[1:0]
		5:4	0x2	reg_52[7:6]_EQ_array_index_18_BST0[1:0]
		3:2	0x2	reg_52[5:4]_EQ_array_index_18_BST1[1:0]
		1:0	0x0	reg_52[3:2]_EQ_array_index_18_BST2[1:0]
0x34	CFGBYT_52_BITDESC	7:6	0x0	reg_52[1:0]_EQ_array_index_18_BST3[1:0]
		5:4	0x1	reg_53[7:6]_EQ_array_index_19_BST0[1:0]
		3:2	0x0	reg_53[5:4]_EQ_array_index_19_BST1[1:0]
		1:0	0x1	reg_53[3:2]_EQ_array_index_19_BST2[1:0]
0x35	CFGBYT_53_BITDESC	7:6	0x2	reg_53[1:0]_EQ_array_index_19_BST3[1:0]
		5:4	0x1	reg_54[7:6]_EQ_array_index_20_BST0[1:0]
		3:2	0x1	reg_54[5:4]_EQ_array_index_20_BST1[1:0]
		1:0	0x0	reg_54[3:2]_EQ_array_index_20_BST2[1:0]
0x36	CFGBYT_54_BITDESC	7:6	0x2	reg_54[1:0]_EQ_array_index_20_BST3[1:0]
		5:4	0x2	reg_55[7:6]_EQ_array_index_21_BST0[1:0]
		3:2	0x0	reg_55[5:4]_EQ_array_index_21_BST1[1:0]
		1:0	0x3	reg_55[3:2]_EQ_array_index_21_BST2[1:0]
0x37	CFGBYT_55_BITDESC	7:6	0x0	reg_55[1:0]_EQ_array_index_21_BST3[1:0]
		5:4	0x2	reg_56[7:6]_EQ_array_index_22_BST0[1:0]
		3:2	0x3	reg_56[5:4]_EQ_array_index_22_BST1[1:0]
		1:0	0x0	reg_56[3:2]_EQ_array_index_22_BST2[1:0]
0x38	CFGBYT_56_BITDESC	7:6	0x0	reg_56[1:0]_EQ_array_index_22_BST3[1:0]
		5:4	0x3	reg_57[7:6]_EQ_array_index_23_BST0[1:0]
		3:2	0x0	reg_57[5:4]_EQ_array_index_23_BST1[1:0]
		1:0	0x2	reg_57[3:2]_EQ_array_index_23_BST2[1:0]
0x39	CFGBYT_57_BITDESC	7:6	0x0	reg_57[1:0]_EQ_array_index_23_BST3[1:0]
		5:4	0x1	reg_58[7:6]_EQ_array_index_24_BST0[1:0]
		3:2	0x1	reg_58[5:4]_EQ_array_index_24_BST1[1:0]
		1:0	0x1	reg_58[3:2]_EQ_array_index_24_BST2[1:0]
0x3A	CFGBYT_58_BITDESC	7:6	0x3	reg_58[1:0]_EQ_array_index_24_BST3[1:0]
		5:4	0x1	reg_59[7:6]_EQ_array_index_25_BST0[1:0]
		3:2	0x1	reg_59[5:4]_EQ_array_index_25_BST1[1:0]
		1:0	0x3	reg_59[3:2]_EQ_array_index_25_BST2[1:0]
0x3B	CFGBYT_59_BITDESC	7:6	0x1	reg_59[1:0]_EQ_array_index_25_BST3[1:0]
		5:4	0x1	reg_5A[7:6]_EQ_array_index_26_BST0[1:0]
		3:2	0x2	reg_5A[5:4]_EQ_array_index_26_BST1[1:0]
		1:0	0x2	reg_5A[3:2]_EQ_array_index_26_BST2[1:0]
0x3C	CFGBYT_60_BITDESC	7:6	0x1	reg_5A[1:0]_EQ_array_index_26_BST3[1:0]
		5:4	0x1	reg_5B[7:6]_EQ_array_index_27_BST0[1:0]
		3:2	0x3	reg_5B[5:4]_EQ_array_index_27_BST1[1:0]
		1:0	0x1	reg_5B[3:2]_EQ_array_index_27_BST2[1:0]
0x3D	CFGBYT_61_BITDESC	7:6	0x1	reg_5B[1:0]_EQ_array_index_27_BST3[1:0]
		5:4	0x3	reg_5C[7:6]_EQ_array_index_28_BST0[1:0]
		3:2	0x1	reg_5C[5:4]_EQ_array_index_28_BST1[1:0]
		1:0	0x1	reg_5C[3:2]_EQ_array_index_28_BST2[1:0]
0x3E	CFGBYT_62_BITDESC	7:6	0x1	reg_5C[1:0]_EQ_array_index_28_BST3[1:0]
		5:4	0x2	reg_5D[7:6]_EQ_array_index_29_BST0[1:0]

Table 9. Channel Register Data Set (continued)

Address	Register Name	Bit(s)	Default Value	Field
		3:2	0x1	reg_5D[5:4]_EQ_array_index_29_BST1[1:0]
		1:0	0x2	reg_5D[3:2]_EQ_array_index_29_BST2[1:0]
0x3F	CFGBYT_63_BITDESC	7:6	0x1	reg_5D[1:0]_EQ_array_index_29_BST3[1:0]
		5:4	0x2	reg_5E[7:6]_EQ_array_index_30_BST0[1:0]
		3:2	0x1	reg_5E[5:4]_EQ_array_index_30_BST1[1:0]
		1:0	0x1	reg_5E[3:2]_EQ_array_index_30_BST2[1:0]
0x40	CFGBYT_64_BITDESC	7:6	0x2	reg_5E[1:0]_EQ_array_index_30_BST3[1:0]
		5:4	0x2	reg_5F[7:6]_EQ_array_index_31_BST0[1:0]
		3:2	0x2	reg_5F[5:4]_EQ_array_index_31_BST1[1:0]
		1:0	0x1	reg_5F[3:2]_EQ_array_index_31_BST2[1:0]
0x41	CFGBYT_65_BITDESC	7:6	0x1	reg_5F[1:0]_EQ_array_index_31_BST3[1:0]
		5:0	0x0	reg_60[7:2]_grp0_ov_cnt[7:2]
0x42	CFGBYT_66_BITDESC	7:6	0x0	reg_60[1:0]_grp0_ov_cnt[1:0]
		5	0x0	reg_61[7]_cnt_dlta_ov0
		4:0	0x0	reg_61[6:2]_grp0_ov_cnt[14:10]
0x43	CFGBYT_67_BITDESC	7:6	0x0	reg_61[1:0]_grp0_ov_cnt[9:8]
		5:0	0x0	reg_62[7:2]_grp1_ov_cnt[7:2]
0x44	CFGBYT_68_BITDESC	7:6	0x0	reg_62[1:0]_grp1_ov_cnt[1:0]
		5	0x0	reg_63[7]_cnt_dlta_ov1
		4:0	0x0	reg_63[6:2]_grp1_ov_cnt[14:10]
0x45	CFGBYT_69_BITDESC	7:6	0x0	reg_63[1:0]_grp1_ov_cnt[9:8]
		5:2	0x0	reg_64[7:4]_grp1_ov_dlta[3:0]
		1:0	0x0	reg_64[3:2]_grp0_ov_dlta[3:2]
0x46	CFGBYT_70_BITDESC	7:6	0x0	reg_64[1:0]_grp0_ov_dlta[1:0]
		5:2	0xA	reg_69[3:0]_hv_lckmon_cnt_ms[3:0]
		1:0	0x0	reg_6B[7:6]_fom_a[7:6]
0x47	CFGBYT_71_BITDESC	7:2	0x0	reg_6B[5:0]_fom_a[5:0]
		1:0	0x0	reg_6C[7:6]_fom_b[7:6]
0x48	CFGBYT_72_BITDESC	7:2	0x0	reg_6C[5:0]_fom_b[5:0]
		1:0	0x0	reg_6D[7:6]_fom_c[7:6]
0x49	CFGBYT_73_BITDESC	7:2	0x0	reg_6D[5:0]_fom_c[5:0]
		1	0x0	reg_6E[7]_en_new_fom_ctle
		0	0x0	reg_6E[6]_en_new_fom_dfe

Table 10. Retimer Channel Register 0x03 Description

Address (Hex)	Bits	Default Value (Hex)	Mode	Field Name	Description
0x03	7:6	0x0	R/W	eq_BST0[1:0]	CTLE Boost Stage 0 <1:0>
	5:4	0x0	R/W	eq_BST1[1:0]	CTLE Boost Stage 1 <1:0>
	3:2	0x0	R/W	eq_BST2[1:0]	CTLE Boost Stage 2 <1:0>
	1:0	0x0	R/W	eq_BST3[1:0]	CTLE Boost Stage 3 <1:0>

5 Bill of Materials:

Table 11. Bill of Materials:

Item	Qty	Reference	Digikey PN	Manufacture PN	Descriptions
1	2	CR1,CR2	F2594CT-ND	PGB1010603MR	SUPPRESSOR ESD 24VDC 0603 SMD
2	4	C1,C12,C18,C21	311-1357-1-ND	CC0603ZRY5V6BB105	CAP CERAMIC 1UF 10V Y5V 0603
3	16	C2,C8,C9,C10,C11,C15,C19,C22,C44,C45,C46,C47,C48,C49,C50,C51	311-1047-1-ND	CC0402ZRY5V7BB104	CAP .10UF 16V CERAMIC Y5V 0402
4	5	C3,C4,C16,C17,C20	478-3281-1-ND	TAJP106M010RNJ	CAP TANTALUM 10UF 10V 20% SMD
5	3	C5,C6,C7	311-1353-1-ND	CC0402ZRY5V6BB224	CAP CERAMIC .22UF 10V Y5V 0402
6	2	C13,C14	478-5126-1-ND	04025A150FAT2A	CAP CER 15PF 50V NP0 0402
7	4	C23,C29,C33,C38	587-2476-1-ND	LMK105B7223KV-F	CAP CER 22000PF 10V X7R 10% 0402
8	18	C24,C25,C26,C27,C28,C30,C31,C32,C34,C35,C36,C37,C39,C40,C41,C42,C52,C53	445-4986-1-ND	C1005X5R1A224M	CAP CER 0.22UF 10V 20% X5R 0402
9	1	C43	445-5000-1-ND	C1005X6S0J105K	CAP CER 1.0UF 6.3V X6S 0402
10	2	D1,D5	475-2691-1-ND	LS M67K-J2L1-1-Z	LED MINI TOPLED RED 630NM SMD
11	11	D2,D4,D6,D7,D8,D17,D18,D19,D20,D21,D22	475-2750-1-ND	LP M67K-E2G1-25-Z	LED MINI TOPLED GREEN 560NM SMD
12	1	D3	160-1409-1-ND	LTST-C155KGJRKT	LED GREEN/RED BICOLOR 1210 SMD
13	3	J1,J3,J4	7006K-ND	7006	POST BINDING ECON NYLON-INS RED
14	1	J2	7007K-ND	7007	POST BINDING ECON NYLON-INS BLK
15	7	J5,J28,J29,J30,J31,J84,J85	A26543-ND	87224-2	CONN HEADER VERT .100 2POS 15AU
16	1	J6	H2959CT-ND	UX60-MB-5ST	CONN RECEPT MINI USB2.0 5POS.
17	20	J7,J8,J9,J10,J11,J12,J13,J14,J15,J16,J17,J18,J19,J20,J21,J22,J80,J81,J82,J83	WM5535-ND	73251-1850	CONN JACK SMA FLANGE MOUNT GOLD
18	1	J23	ARFX1231-ND	901-144-8RFX	CONN SMA RECEPTACLE STRAIGHT PCB
19	7	J32,J33,J34,J35,J36,J37,J38	A34269-09-ND	9-146256-0-09	CONN HDR BRKWAY .100 18POS VERT
20	1	J39	A26567-ND	87227-2	CONN HEADER VERT .100 4POS 15AU
21	1	J40	SAM1008-09-ND	BCS-109-L-D-TE	CONN RCPT 18POS .100 DUAL VERT
22	1	J41	A26545-ND	87224-3	CONN HEADER VERT .100 3POS 15AU
23	1	J42	4-1761206-1-ND	4-1761206-1	CONN RCPT 4POS R/A SDL GOLD
24	1	J86	A26547-ND	87224-4	CONN HEADER VERT .100 4POS 15AU
25	1	Q2	SI6925ADQ-T1-GE3TR-ND	SI6925ADQ-T1-GE3	MOSFET DL N-CH 20V 3.9A 8-TSSOP
26	2	RN1,RN6	858-668A2001BLF	668A2001BLF	Resistor Networks & Arrays 2K .1% 16PIN THINFILM DIP

Table 11. Bill of Materials: (continued)

Item	Qty	Reference	Digikey PN	Manufacture PN	Descriptions
27	2	RN2,RN3	858-668A1001DLF7	668A1001DLF7	Resistor Networks & Arrays 1K .1% 16PIN THINFILM DIP
28	2	RN4,RN7	652-4816P-T1LF-390	4816P-T01-391LF	Resistor Networks & Arrays 390ohm 2% 16Pin SMT
29	2	R1,R2	P22JCT-ND	ERJ-2GEJ220X	RES 22 OHM 1/10W 5% 0402 SMD
30	10	R3,R4,R7,R8,R9,R10,R11,R12,R13,R14	P0.0JCT-ND	ERJ-2GE0R00X	RES 0.0 OHM 1/10W 0402 SMD
31	1	R5	P.10AKCT-ND	ERJ-2BSFR10X	RESISTOR .10 OHM 1/8W 1% 0402
32	1	R6	P300JCT-ND	ERJ-2GEJ301X	RES 300 OHM 1/10W 5% 0402 SMD
33	2	R15,R16	541-100YCT-ND	CRCW0402100RFKEDHP	RES 100 OHM .125W 1% 0402 SMD
34	3	SW1,SW2,SW3	ADTSM31NV-ND	ADTSM31NV	SWITCH TACT SPST 12VDC 160GF
35	1	SW4	CT206124-ND	206-124	SWITCH SPDT GOLD
36	1	SW5	CT204121ST-ND	204-121ST	SWITCH DIP SPDT 1POS SMT STDPRO
37	1	U1		DS100DF410SQ/NOPB or DS110DF410SQ/NOPB or DS125DF410SQ/NOPB	Quad Retimer with CTLE, DFE
38	1	U2	LP3874EMP-2.5CT-ND	LP3874EMP-2.5/NOPB	IC REG LDO 0.8A 2.5V SOT223-5
39	1	U3	AT90USB1287-MU-ND	AT90USB1287-MU	IC AVR MCU 128K 64QFN
40	1	U4	887-1442-1-ND	7C-25.000MCB-T	OSCILLATOR 25.000 MHZ 2.5V SMD
41	1	U5	AT24C08B-PU-ND	AT24C08B-PU	IC EEPROM 8KBIT 1MHZ 8DIP
42	1	U6	296-21917-2-ND	TS3A4741DGKR	IC SWITCH DUAL SPST 8MSOP
43	1	Y1	535-10630-1-ND	ABM3-8.000MHZ-D2Y-T	CRYSTAL 8.000 MHZ 18PF SMD
44	1	SOCKET for line item 41 (U5)	A24802-ND	2-641260-4	CONN IC SOCKET 8 POS DIP 15AU
45	40	Screw		91772A052	18-8 Stainless Steel Pan Head Phillips Machine Screw

6 Schematic:

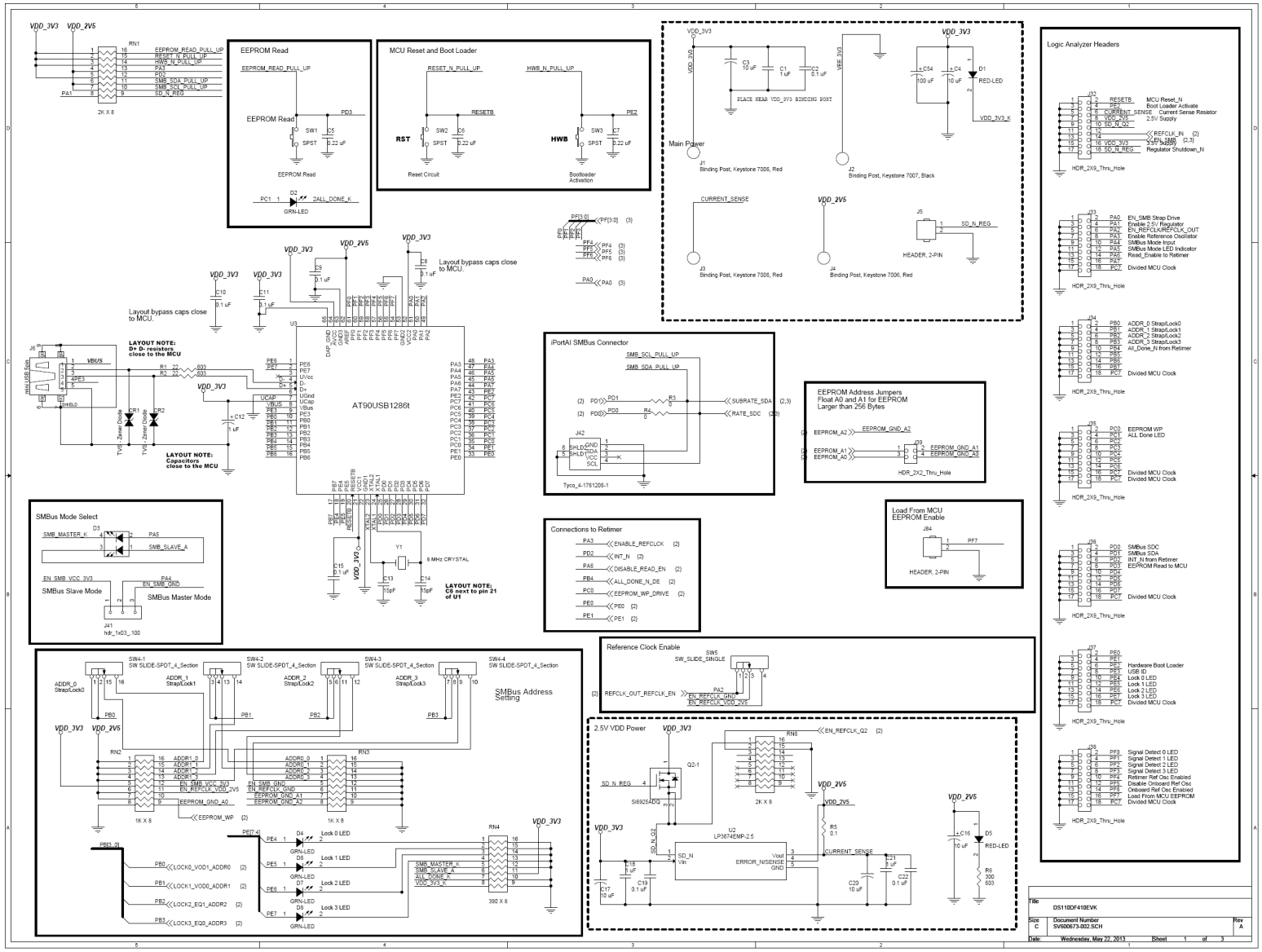

Figure 11. DS100DF410EVK, DS110DF410EVK and DS125DF410EVM Schematic Page 1

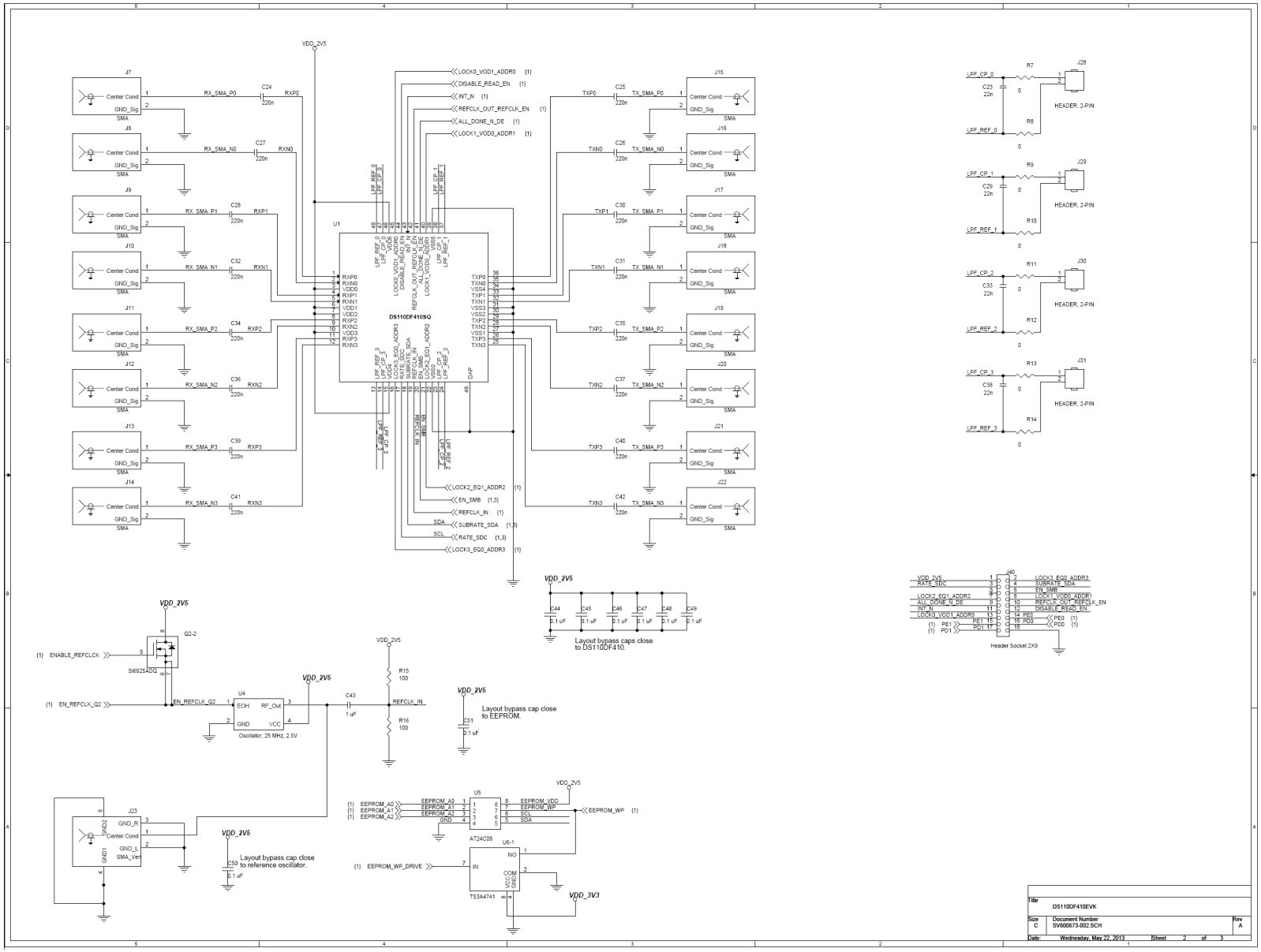

Figure 12. DS100DF410EVK, DS110DF410EVK and DS125DF410EVM Schematic Page 2

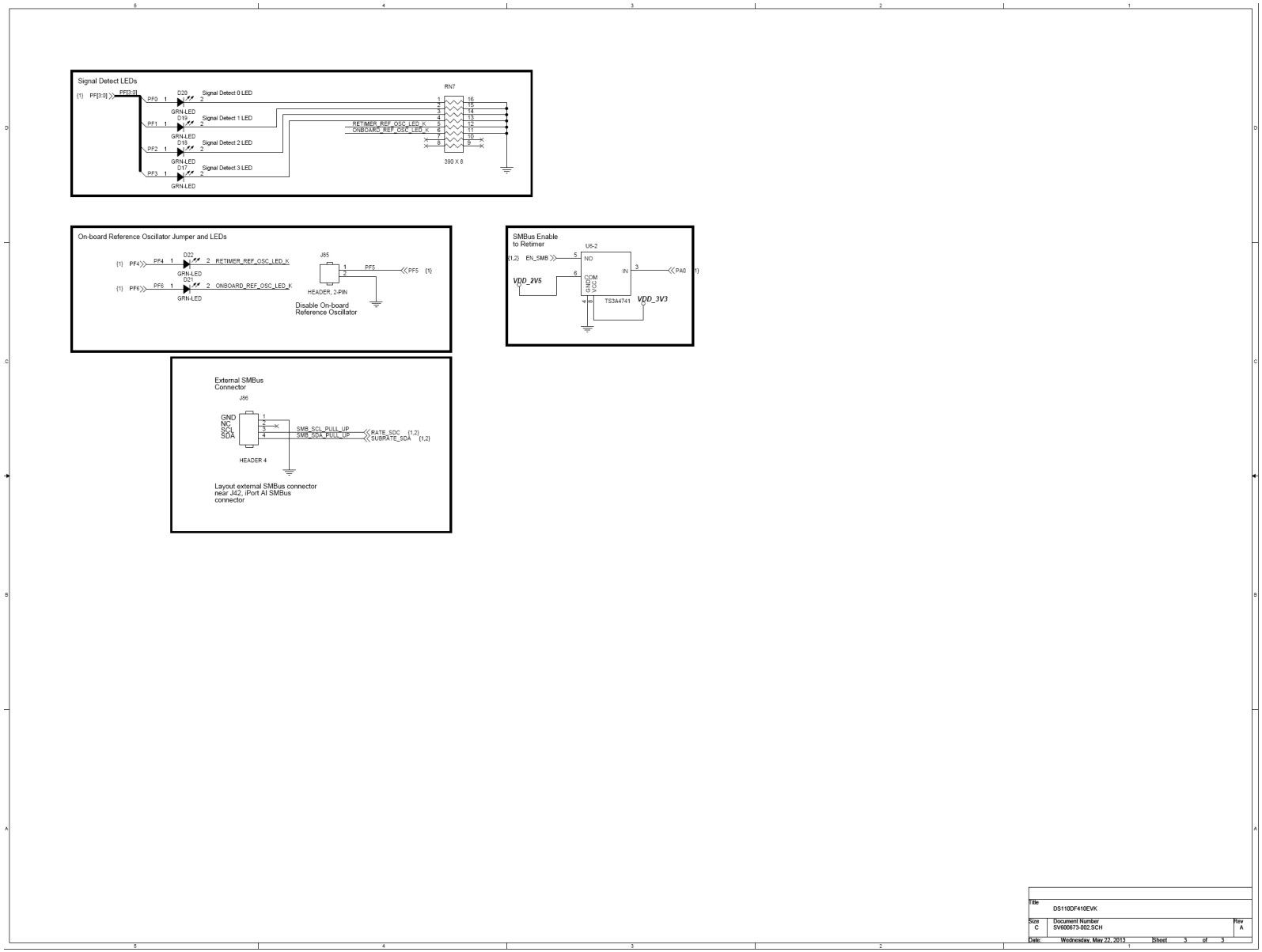

Figure 13. DS100DF410EVK, DS110DF410EVK and DS125DF410EVM Schematic Page 3

7 Board Layout:

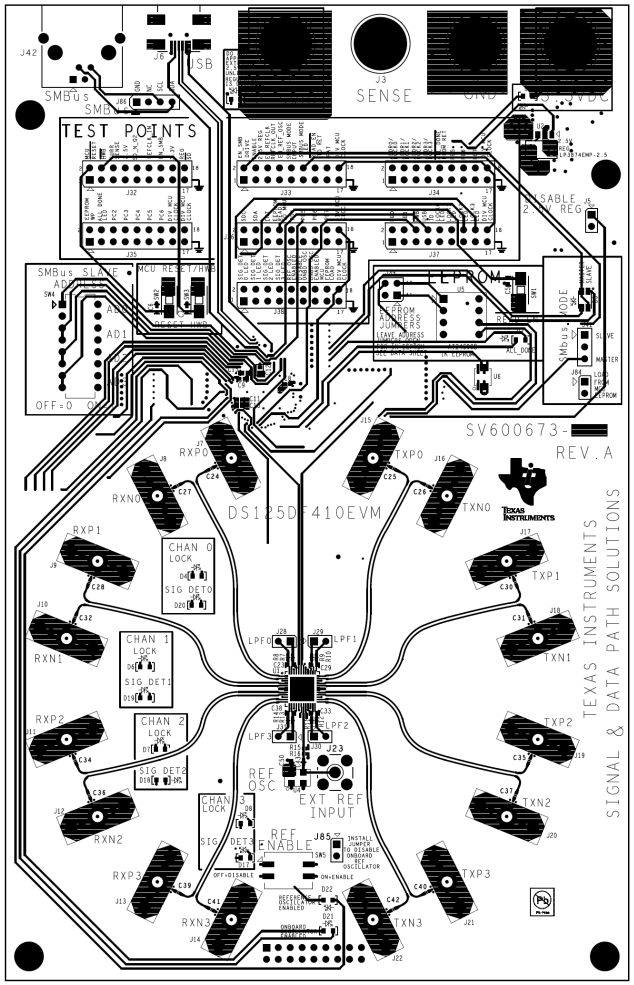

Figure 14. Top Layer

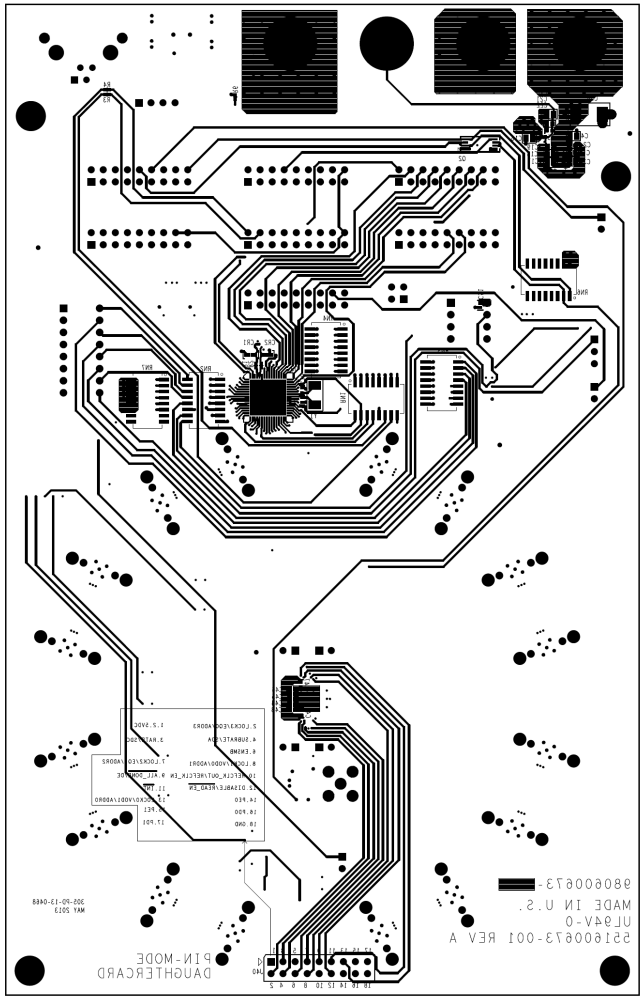

Figure 15. Bottom Layer

EVALUATION BOARD/KIT/MODULE (EVM) ADDITIONAL TERMS

Texas Instruments (TI) provides the enclosed Evaluation Board/Kit/Module (EVM) under the following conditions:

The user assumes all responsibility and liability for proper and safe handling of the goods. Further, the user indemnifies TI from all claims arising from the handling or use of the goods.

Should this evaluation board/kit not meet the specifications indicated in the User's Guide, the board/kit may be returned within 30 days from the date of delivery for a full refund. THE FOREGOING LIMITED WARRANTY IS THE EXCLUSIVE WARRANTY MADE BY SELLER TO BUYER AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESSED, IMPLIED, OR STATUTORY, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. EXCEPT TO THE EXTENT OF THE INDEMNITY SET FORTH ABOVE, NEITHER PARTY SHALL BE LIABLE TO THE OTHER FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES.

Please read the User's Guide and, specifically, the Warnings and Restrictions notice in the User's Guide prior to handling the product. This notice contains important safety information about temperatures and voltages. For additional information on TI's environmental and/or safety programs, please visit www.ti.com/esh or contact TI.

No license is granted under any patent right or other intellectual property right of TI covering or relating to any machine, process, or combination in which such TI products or services might be or are used. TI currently deals with a variety of customers for products, and therefore our arrangement with the user is not exclusive. TI assumes no liability for applications assistance, customer product design, software performance, or infringement of patents or services described herein.

REGULATORY COMPLIANCE INFORMATION

As noted in the EVM User's Guide and/or EVM itself, this EVM and/or accompanying hardware may or may not be subject to the Federal Communications Commission (FCC) and Industry Canada (IC) rules.

For EVMs **not** subject to the above rules, this evaluation board/kit/module is intended for use for ENGINEERING DEVELOPMENT, DEMONSTRATION OR EVALUATION PURPOSES ONLY and is not considered by TI to be a finished end product fit for general consumer use. It generates, uses, and can radiate radio frequency energy and has not been tested for compliance with the limits of computing devices pursuant to part 15 of FCC or ICES-003 rules, which are designed to provide reasonable protection against radio frequency interference. Operation of the equipment may cause interference with radio communications, in which case the user at his own expense will be required to take whatever measures may be required to correct this interference.

General Statement for EVMs including a radio

User Power/Frequency Use Obligations: This radio is intended for development/professional use only in legally allocated frequency and power limits. Any use of radio frequencies and/or power availability of this EVM and its development application(s) must comply with local laws governing radio spectrum allocation and power limits for this evaluation module. It is the user's sole responsibility to only operate this radio in legally acceptable frequency space and within legally mandated power limitations. Any exceptions to this are strictly prohibited and unauthorized by Texas Instruments unless user has obtained appropriate experimental/development licenses from local regulatory authorities, which is responsibility of user including its acceptable authorization.

For EVMs annotated as FCC – FEDERAL COMMUNICATIONS COMMISSION Part 15 Compliant

Caution

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

FCC Interference Statement for Class A EVM devices

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

FCC Interference Statement for Class B EVM devices

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna.
- Increase the separation between the equipment and receiver.
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.
- Consult the dealer or an experienced radio/TV technician for help.

For EVMs annotated as IC – INDUSTRY CANADA Compliant

This Class A or B digital apparatus complies with Canadian ICES-003.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

Concerning EVMs including radio transmitters

This device complies with Industry Canada licence-exempt RSS standard(s). Operation is subject to the following two conditions: (1) this device may not cause interference, and (2) this device must accept any interference, including interference that may cause undesired operation of the device.

Concerning EVMs including detachable antennas

Under Industry Canada regulations, this radio transmitter may only operate using an antenna of a type and maximum (or lesser) gain approved for the transmitter by Industry Canada. To reduce potential radio interference to other users, the antenna type and its gain should be so chosen that the equivalent isotropically radiated power (e.i.r.p.) is not more than that necessary for successful communication.

This radio transmitter has been approved by Industry Canada to operate with the antenna types listed in the user guide with the maximum permissible gain and required antenna impedance for each antenna type indicated. Antenna types not included in this list, having a gain greater than the maximum gain indicated for that type, are strictly prohibited for use with this device.

Cet appareil numérique de la classe A ou B est conforme à la norme NMB-003 du Canada.

Les changements ou les modifications pas expressément approuvés par la partie responsable de la conformité ont pu vider l'autorité de l'utilisateur pour actionner l'équipement.

Concernant les EVMs avec appareils radio

Le présent appareil est conforme aux CNR d'Industrie Canada applicables aux appareils radio exempts de licence. L'exploitation est autorisée aux deux conditions suivantes : (1) l'appareil ne doit pas produire de brouillage, et (2) l'utilisateur de l'appareil doit accepter tout brouillage radioélectrique subi, même si le brouillage est susceptible d'en compromettre le fonctionnement.

Concernant les EVMs avec antennes détachables

Conformément à la réglementation d'Industrie Canada, le présent émetteur radio peut fonctionner avec une antenne d'un type et d'un gain maximal (ou inférieur) approuvé pour l'émetteur par Industrie Canada. Dans le but de réduire les risques de brouillage radioélectrique à l'intention des autres utilisateurs, il faut choisir le type d'antenne et son gain de sorte que la puissance isotrope rayonnée équivalente (p.i.r.e.) ne dépasse pas l'intensité nécessaire à l'établissement d'une communication satisfaisante.

Le présent émetteur radio a été approuvé par Industrie Canada pour fonctionner avec les types d'antenne énumérés dans le manuel d'usage et ayant un gain admissible maximal et l'impédance requise pour chaque type d'antenne. Les types d'antenne non inclus dans cette liste, ou dont le gain est supérieur au gain maximal indiqué, sont strictement interdits pour l'exploitation de l'émetteur.

【Important Notice for Users of this Product in Japan】

This development kit is NOT certified as Confirming to Technical Regulations of Radio Law of Japan

If you use this product in Japan, you are required by Radio Law of Japan to follow the instructions below with respect to this product:

1. Use this product in a shielded room or any other test facility as defined in the notification #173 issued by Ministry of Internal Affairs and Communications on March 28, 2006, based on Sub-section 1.1 of Article 6 of the Ministry's Rule for Enforcement of Radio Law of Japan,
2. Use this product only after you obtained the license of Test Radio Station as provided in Radio Law of Japan with respect to this product, or
3. Use of this product only after you obtained the Technical Regulations Conformity Certification as provided in Radio Law of Japan with respect to this product. Also, please do not transfer this product, unless you give the same notice above to the transferee. Please note that if you could not follow the instructions above, you will be subject to penalties of Radio Law of Japan.

Texas Instruments Japan Limited
(address) 24-1, Nishi-Shinjuku 6 chome, Shinjuku-ku, Tokyo, Japan

http://www.tij.co.jp

【ご使用にあたっての注】

本開発キットは技術基準適合証明を受けておりません。

本製品のご使用に際しては、電波法遵守のため、以下のいずれかの措置を取っていただく必要がありますのでご注意ください。

1. 電波法施行規則第6条第1項第1号に基づく平成18年3月28日総務省告示第173号で定められた電波暗室等の試験設備でご使用いただく。
2. 実験局の免許を取得後ご使用いただく。
3. 技術基準適合証明を取得後ご使用いただく。

なお、本製品は、上記の「ご使用にあたっての注意」を譲渡先、移転先に通知しない限り、譲渡、移転できないものとします。

上記を遵守頂けない場合は、電波法の罰則が適用される可能性があることをご留意ください。

日本テキサス・インスツルメンツ株式会社
東京都新宿区西新宿６丁目２４番１号
西新宿三井ビル
http://www.tij.co.jp

EVALUATION BOARD/KIT/MODULE (EVM)
WARNINGS, RESTRICTIONS AND DISCLAIMERS

For Feasibility Evaluation Only, in Laboratory/Development Environments. Unless otherwise indicated, this EVM is not a finished electrical equipment and not intended for consumer use. It is intended solely for use for preliminary feasibility evaluation in laboratory/development environments by technically qualified electronics experts who are familiar with the dangers and application risks associated with handling electrical mechanical components, systems and subsystems. It should not be used as all or part of a finished end product.

Your Sole Responsibility and Risk. You acknowledge, represent and agree that:

1. You have unique knowledge concerning Federal, State and local regulatory requirements (including but not limited to Food and Drug Administration regulations, if applicable) which relate to your products and which relate to your use (and/or that of your employees, affiliates, contractors or designees) of the EVM for evaluation, testing and other purposes.
2. You have full and exclusive responsibility to assure the safety and compliance of your products with all such laws and other applicable regulatory requirements, and also to assure the safety of any activities to be conducted by you and/or your employees, affiliates, contractors or designees, using the EVM. Further, you are responsible to assure that any interfaces (electronic and/or mechanical) between the EVM and any human body are designed with suitable isolation and means to safely limit accessible leakage currents to minimize the risk of electrical shock hazard.
3. You will employ reasonable safeguards to ensure that your use of the EVM will not result in any property damage, injury or death, even if the EVM should fail to perform as described or expected.
4. You will take care of proper disposal and recycling of the EVM's electronic components and packing materials.

Certain Instructions. It is important to operate this EVM within TI's recommended specifications and environmental considerations per the user guidelines. Exceeding the specified EVM ratings (including but not limited to input and output voltage, current, power, and environmental ranges) may cause property damage, personal injury or death. If there are questions concerning these ratings please contact a TI field representative prior to connecting interface electronics including input power and intended loads. Any loads applied outside of the specified output range may result in unintended and/or inaccurate operation and/or possible permanent damage to the EVM and/or interface electronics. Please consult the EVM User's Guide prior to connecting any load to the EVM output. If there is uncertainty as to the load specification, please contact a TI field representative. During normal operation, some circuit components may have case temperatures greater than 60°C as long as the input and output are maintained at a normal ambient operating temperature. These components include but are not limited to linear regulators, switching transistors, pass transistors, and current sense resistors which can be identified using the EVM schematic located in the EVM User's Guide. When placing measurement probes near these devices during normal operation, please be aware that these devices may be very warm to the touch. As with all electronic evaluation tools, only qualified personnel knowledgeable in electronic measurement and diagnostics normally found in development environments should use these EVMs.

Agreement to Defend, Indemnify and Hold Harmless. You agree to defend, indemnify and hold TI, its licensors and their representatives harmless from and against any and all claims, damages, losses, expenses, costs and liabilities (collectively, "Claims") arising out of or in connection with any use of the EVM that is not in accordance with the terms of the agreement. This obligation shall apply whether Claims arise under law of tort or contract or any other legal theory, and even if the EVM fails to perform as described or expected.

Safety-Critical or Life-Critical Applications. If you intend to evaluate the components for possible use in safety critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, such as devices which are classified as FDA Class III or similar classification, then you must specifically notify TI of such intent and enter into a separate Assurance and Indemnity Agreement.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as “components”) are sold subject to TI’s terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI’s terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers’ products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers’ products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI’s goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or “enhanced plastic” are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have ***not*** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.

Products		**Applications**	
Audio	www.ti.com/audio	Automotive and Transportation	www.ti.com/automotive
Amplifiers	amplifier.ti.com	Communications and Telecom	www.ti.com/communications
Data Converters	dataconverter.ti.com	Computers and Peripherals	www.ti.com/computers
DLP® Products	www.dlp.com	Consumer Electronics	www.ti.com/consumer-apps
DSP	dsp.ti.com	Energy and Lighting	www.ti.com/energy
Clocks and Timers	www.ti.com/clocks	Industrial	www.ti.com/industrial
Interface	interface.ti.com	Medical	www.ti.com/medical
Logic	logic.ti.com	Security	www.ti.com/security
Power Mgmt	power.ti.com	Space, Avionics and Defense	www.ti.com/space-avionics-defense
Microcontrollers	microcontroller.ti.com	Video and Imaging	www.ti.com/video
RFID	www.ti-rfid.com		
OMAP Applications Processors	www.ti.com/omap	**TI E2E Community**	e2e.ti.com
Wireless Connectivity	www.ti.com/wirelessconnectivity		

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265