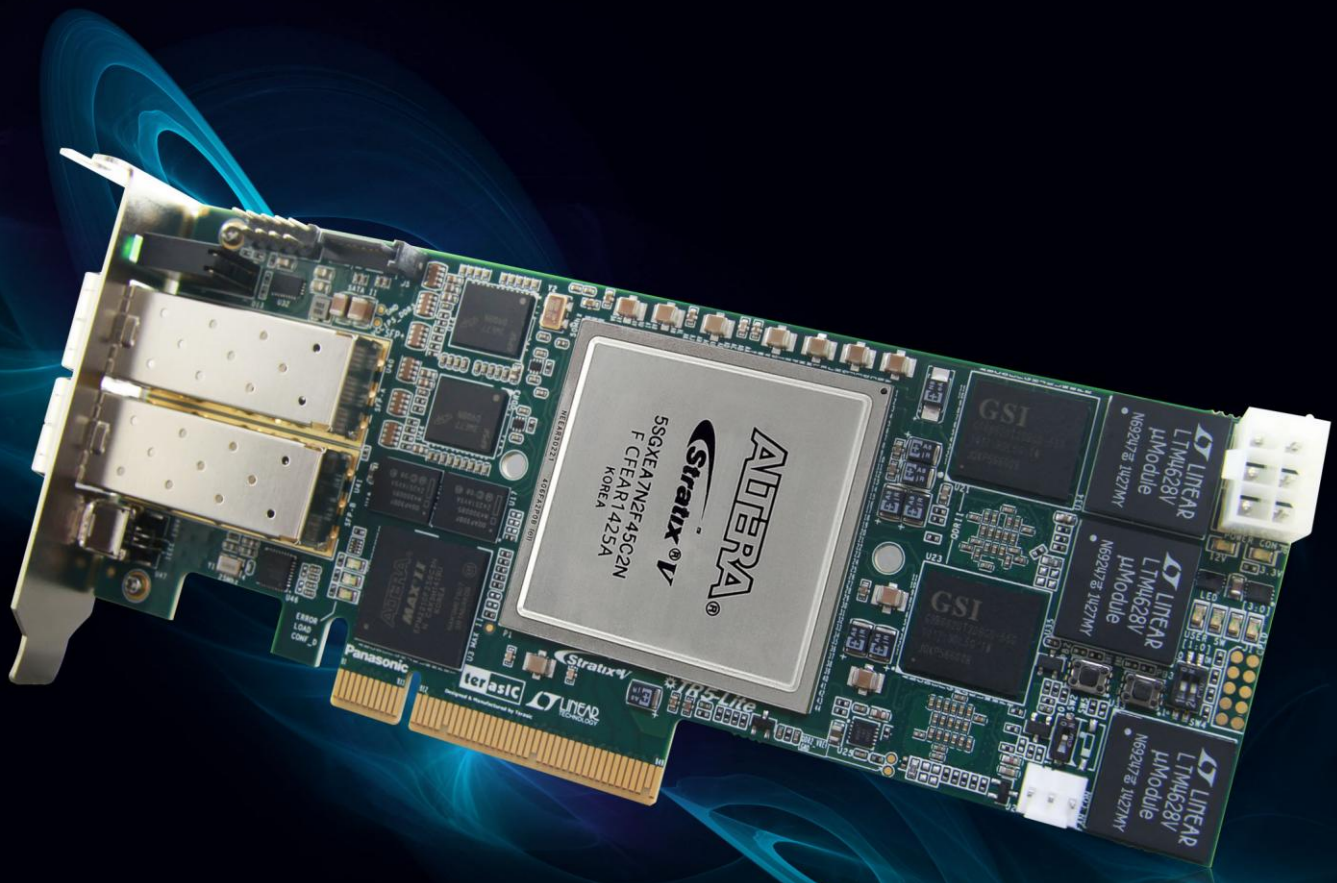
TR5-Lite

FPGA Development Kit

User Manual



terasIC
www.terasic.com

ALTERA

CONTENTS

Chapter 1

Overview

This chapter provides an overview of the TR5-Lite Development Board and installation guide.

1.1 General Description

The Terasic TR5-Lite Stratix V GX FPGA Development Kit provides the ideal hardware solution for designs that demand high bandwidth, advanced memory interfacing, and power efficiency in a convenient half-height, half-length form-factor package. Designed for the most demanding high-end applications, the TR5-Lite is empowered with the top-of-the-line Altera Stratix V GX, delivering the best system-level integration and flexibility in the industry.

The Stratix® V GX FPGA features integrated transceivers that transfer at a maximum of 12.5 Gbps, allowing the TR5-Lite to be fully compliant with version 3.0 of the PCI Express standard, as well as allowing an ultra low-latency, straight connections to dual external 10G SFP+ modules. Not relying on an external PHY will accelerate mainstream development of network applications enabling customers to deploy designs for a broad range of high-speed connectivity applications. Matched with two independent banks of DDR3 RAM, four independent banks of QDRII, and flash memory, the TR5-Lite fully delivers in all high-bandwidth applications such as high frequency trading, data acquisition, network processing, and signal processing.

It is highly recommended that users read the *TR5-Lite Getting Started Guide.pdf* before using the TR5-Lite board.

1.2 Key Features

The following hardware is implemented on the TR5-Lite board:

- FPGA
  - Altera Stratix® V GX FPGA (5SGXEA7N2F45C2)

- FPGA Configuration
  - JTAG header for FPGA programming
  - Fast passive parallel (FPPx32) configuration via MAX II CPLD and flash memory

- General user input/output:
  - 4 LEDs
  - 2 push-buttons
  - 2-position DIP switch

- On-Board Clock
  - 50MHz Oscillator
  - Programmable oscillators Si570 and CDCM61004

- Memory
  - DDR3 SDRAM
  - QDRII+ SRAM
  - FLASH

- Communication Ports
  - Two SFP+ connectors
  - One Serial ATA (SATA II) host port
  - PCI Express (PCIe) x8 edge connector
  - One RS422 transevier with 1394 connector

- System Monitor and Control
  - Temperature sensor
  - Fan control

- Power
  - PCI Express 6-pin power connector, 12V DC Input
  - PCI Express edge connector power

- Mechanical Specification
  - PCI Express half-height and half-length

1.3 Block Diagram

Figure 1-1 shows the block diagram of the TR5-Lite board. To provide maximum flexibility for the users, all key components are connected with the Stratix V GX FPGA device. Thus, users can configure the FPGA to implement any system design.

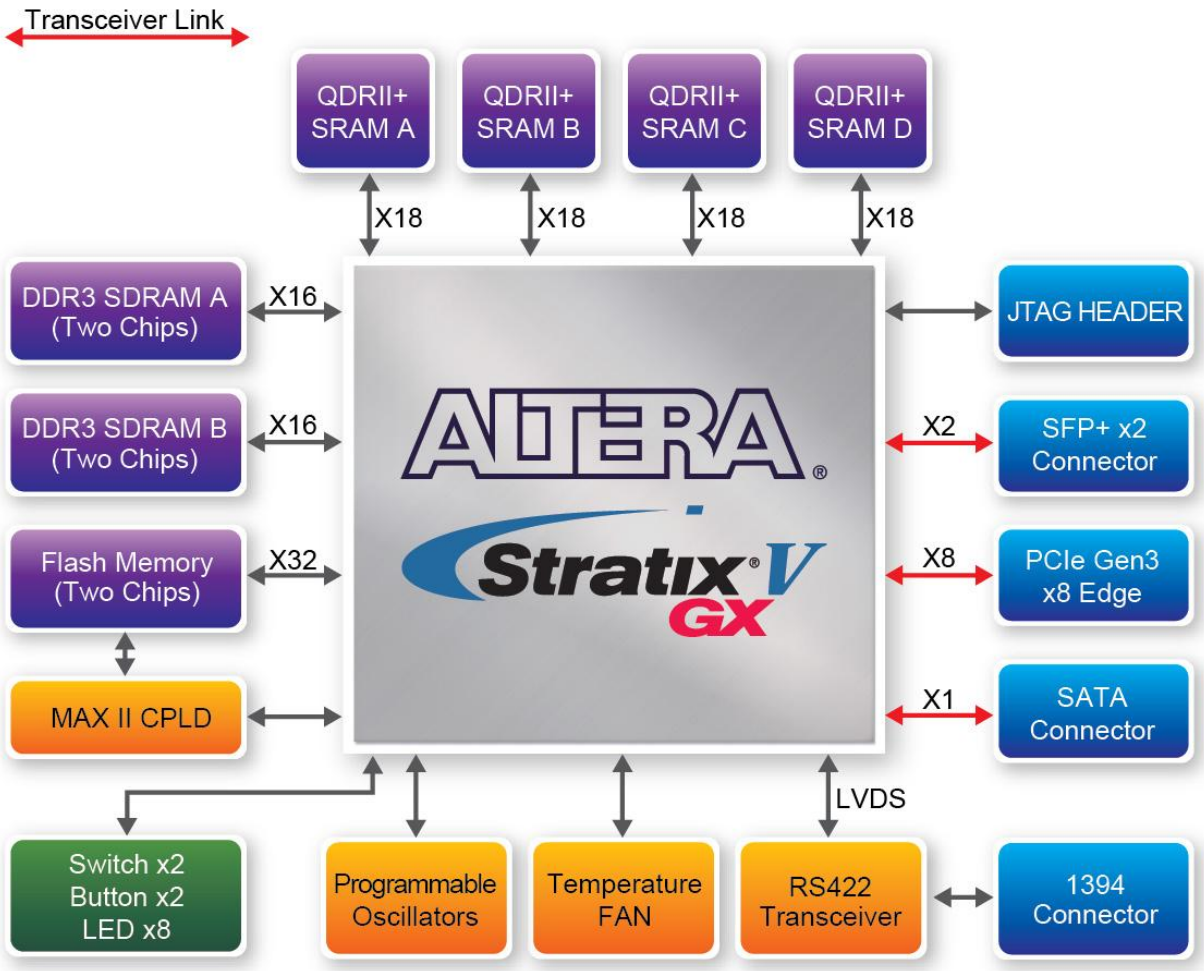


Figure 1-1 Block diagram of the TR5-Lite board

Below is more detailed information regarding the blocks in Figure 1-1.

Stratix V GX FPGA

- 5SGXEA7N2F45C2
- 622,000 logic elements (LEs)
- 50-Mbits embedded memory
- 48 transceivers (12.5Gbps)
- 512 18-bit x 18-bit multipliers
- 256 27-bit x 27-bit DSP blocks
- 2 PCI Express hard IP blocks
- 840 user I/Os
- 210 full-duplex LVDS channels
- 28 phase locked loops (PLLs)

JTAG Header and FPGA Configuration

- On-board JTAG header for use with the Quartus II Programmer
- MAXII CPLD EPM2210 System Controller and Fast Passive Parallel (FPP) configuration

Memory devices

- 32MB QDRII+ SRAM
- 2GB DDR3 SDRAM
- 256MB FLASH

General user I/O

- 4 user controllable LEDs
- 2 user push buttons
- 2 user DIP switches

On-Board Clock

- 50MHz oscillator
- Programming PLL providing clock for 10G SFP+ transceiver
- Programming PLL providing clock for SATA or 1G SFP+ transceiver

One Serial ATA ports

- SATA 3.0 standard 6Gbps signaling rate

Two SFP+ ports

- Two SFP+ connector (10 Gbps+)

PCI Express x8 edge connector

- Support PCIe Gen1/2/3
- Connection established with PC motherboard with x8 or x16 PCI Express slot

Power Source

- PCI Express 6-pin DC 12V power
- PCI Express edge connector power

1.4 Installation Setup

The TR5-Lite board can be functional standalone or installed on a host PC. External USB-Blaster is required to configure the board through the on-board JTAG Header.

Switch Setup

The MSEL[0:4] switches should be in the ON position(MSEL[0:4]=00010), as shown in **Figure 1-2.**

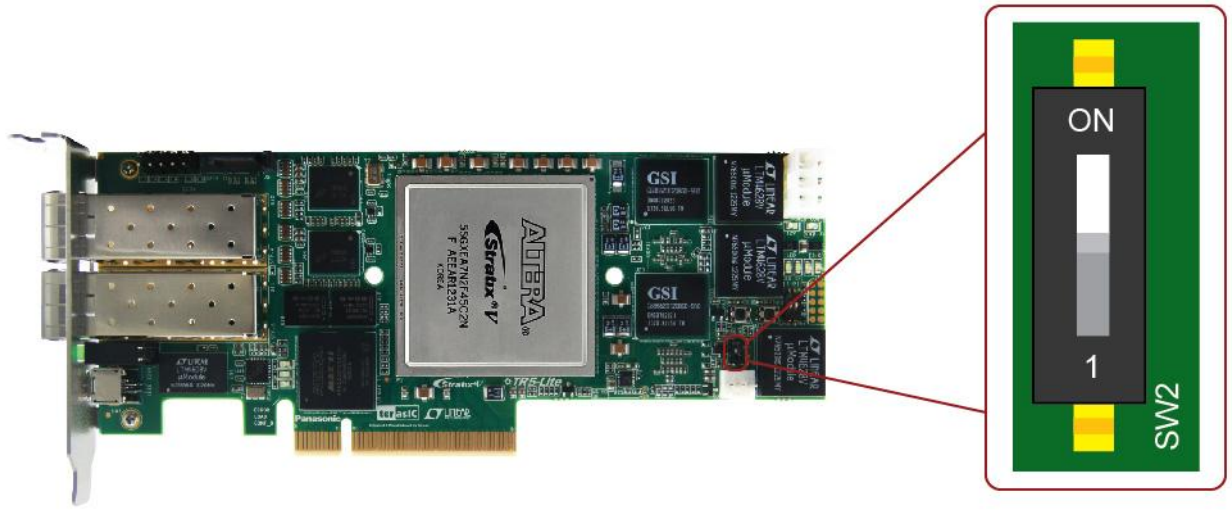


Figure 1-2 MSEL Default Configuration

SW1 is set to low for loading the default factory FPGA configuration. For more details, please see **Figure 1-3**.

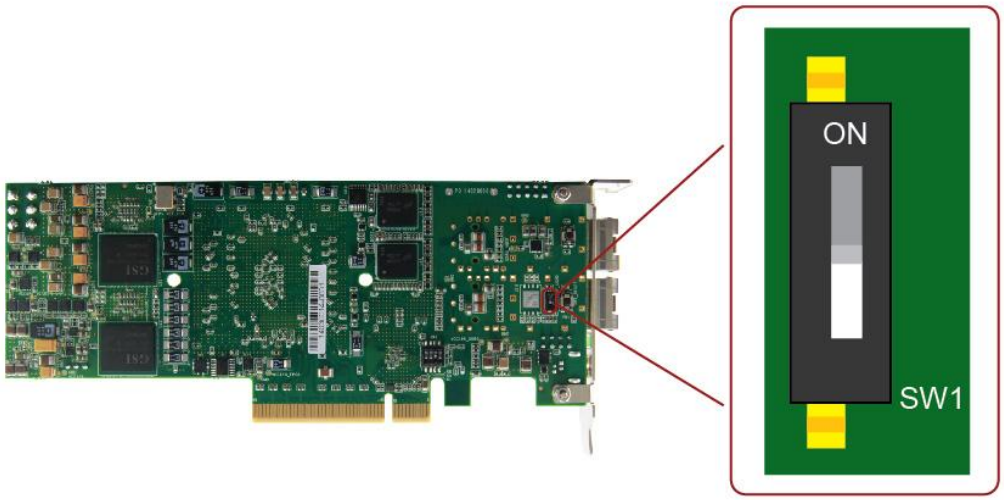


Figure 1-3 Factory Default Configuration Setting

USB-Blaster Setup

An external USB-Blaster (or USB-Blaster II) is required to configure the TR5-Lite board. **Figure 1-4** shows how connect Terasic USB-Blaster II to the JTAG header (J3) on the TR5-Lite. Terasic USB-Blaster II is recommended because it provides a faster downloading speed than the traditional USB-Blaster. Please note the red edge in the flat cable should be connected to the first pin in JTAG header.

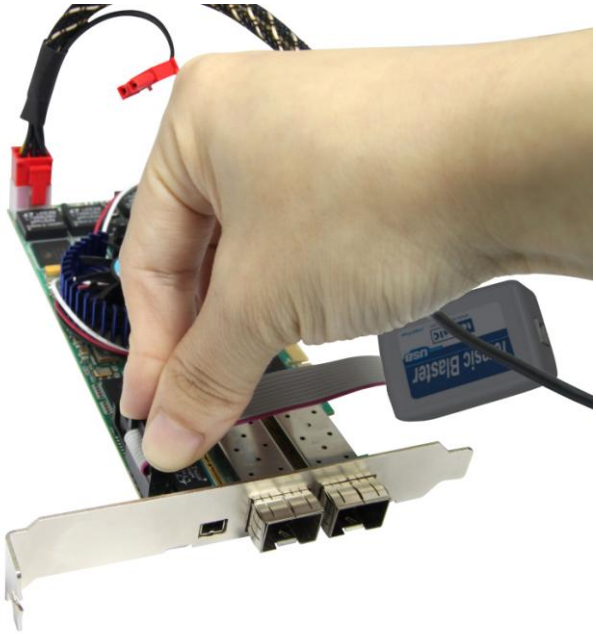


Figure 1-4 Connect the Terasic USB-Blaster II to the JTAG header (J3)

Power Setup

The TR5-Lite power is provided from both the 6-pin DC 12V power connector and PCI Express edge connector. In standalone-mode, only the 6-pin DC 12V power input is required. **Figure 1-5** shows the installation of connecting PCIe 12V DC power source to the 6-pin PCIe power connector on the TR5-Lite board.

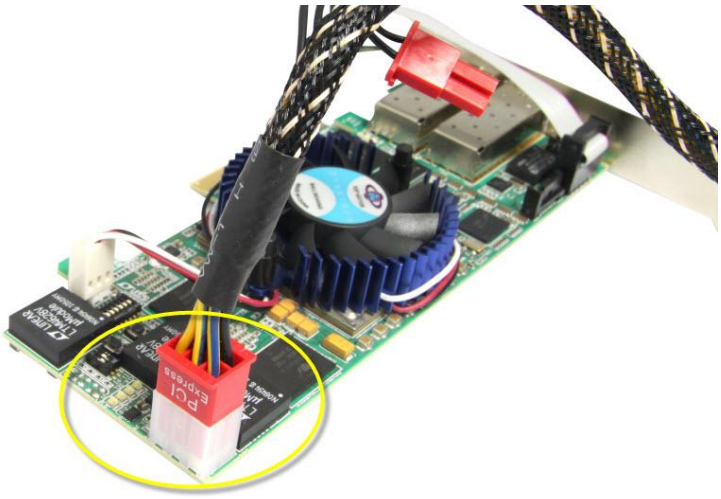


Figure 1-5 Installation of 12V DC for TR5-Lite through the 6-Pin PCIe Power Connector

When TR5-Lite is installed on a PC, the 6-pin DC 12 V power input also is **required** even if there is a power provided through the PCI Express edge connector, as shown in **Figure 1-6**. The 12V DC input can come from the PC power supply if it supports 6-pin PCIe power source. Without the 12V DC power from the 6-pin power connector, the TR5-Lite may be damaged due to insufficient power from the PCIe edge connector when many FPGA resources are used.

Figure 1-6 TR5-Lite Installed on PC

Quartus II Setup

64-bit Quartus 11.1 SP1 or later is strongly recommended. When compiling with Stratix V FPGA with 32-bit Quartus, an out of memory error may occur due to the memory limitation of operating system on the host computer.

The Quartus II golden top project for TR5-Lite is available on the TR5-Lite System CD location **Demonstrations/TR5_Lite_Golden_Top**. The project includes complete pin assignments, so users can develop their projects based on this golden top project regardless of the assignment details.

Chapter 2

Board Components

This chapter introduces all the important components on the TR5-Lite.

2.1 Board Overview

Figure 2-1 and Figure 2-2 is the top and bottom view of the TR5-Lite development board. It depicts the layout of the board and indicates the location of the connectors and key components. Users can refer to this figure for relative location of the connectors and key components.

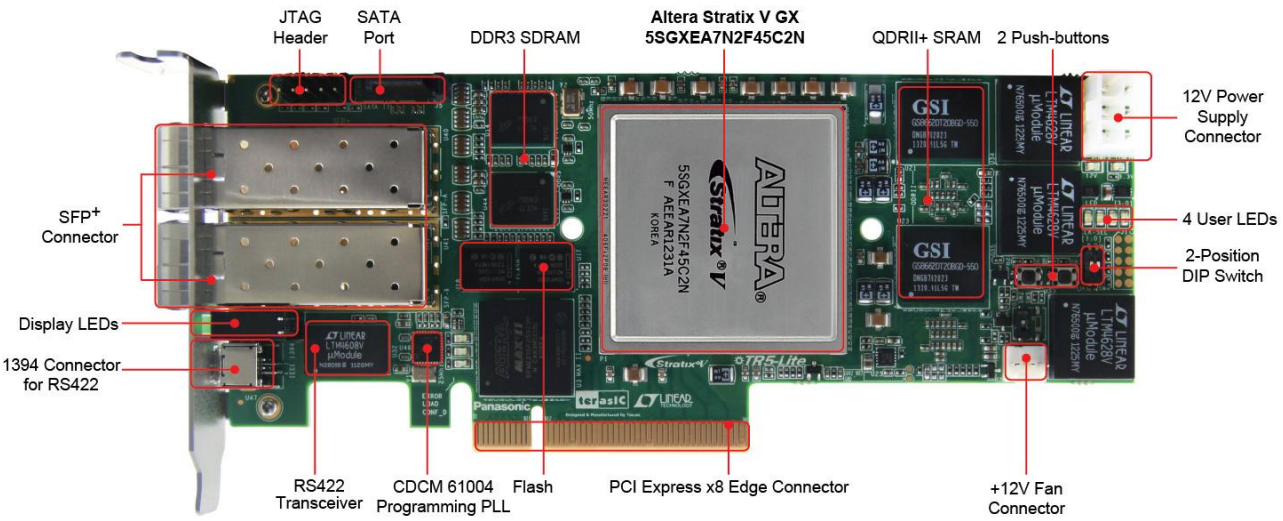


Figure 2-1 The TR5-Lite Board (Top)

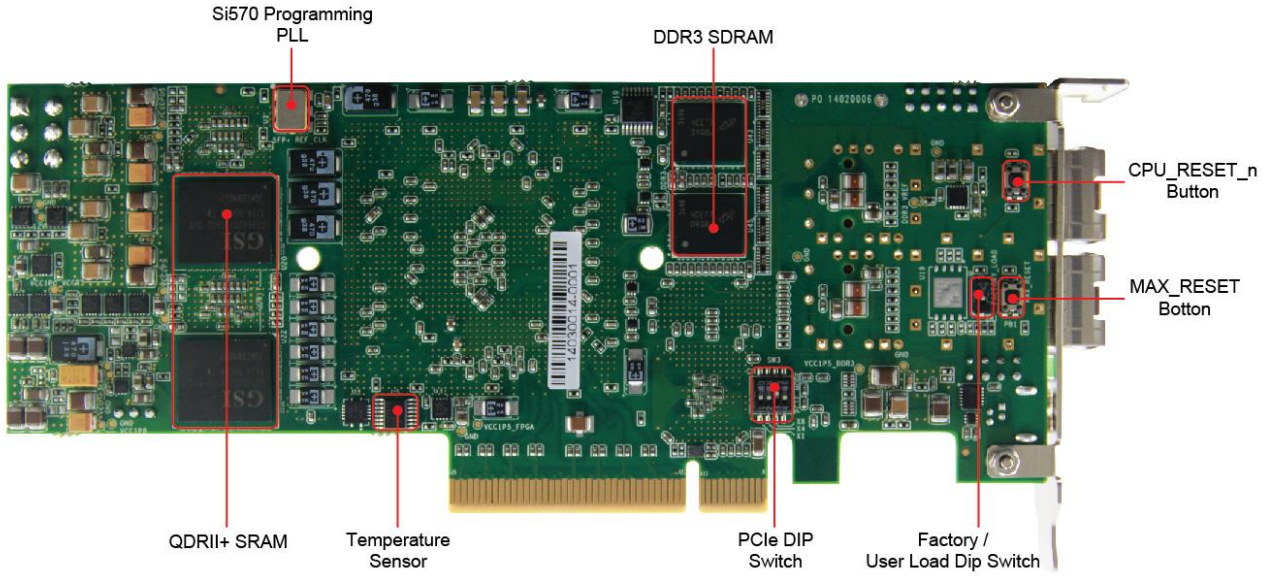


Figure 2-2 The TR5-Lite Board (Bottom)

2.2 Configuration, Status and Setup

■ Configure

The TR5-Lite board supports two configuration methods for the Stratix V FPGA:

- External USB-Blaster for configuring the FPGA using the external USB-Blaster.
- Flash memory configuration of the FPGA using stored images from the flash memory on power-up.

For programming by USB-Blaster, the USB-Blaster should connect to the JTAG header (J3) on the TR5-Lite board. The following procedures show how to download a configuration bit stream into the Stratix V GX FPGA:

- Make sure that power is provided to the TR5-Lite board
- Connect your PC to the TR5-Lite board using a USB-Blaster (or USB-Blaster II) module and make sure the USB-Blaster driver is installed on PC.
- Launch Quartus II programmer and make sure the USB-Blaster (or USB-Blaster II) is detected.
- In Quartus II Programmer, add the configuration bit stream file (.sof), check the associated "Program/Configure" item, and click "Start" to start FPGA programming.

■ Status LED

The TR5-Lite development board includes board-specific status LEDs to indicate board status. Please refer to **Table 2-1** for the description of the LED indicator.

Table 2-1 Status LED

Board Reference	*LED Name*	*Description*
D12	12-V Power	Illuminates when 12-V power is active.
D11	3.3-V Power	Illuminates when 3.3-V power is active.
D1	CONF DONE	Illuminates when the FPGA is successfully configured. Driven by the MAX II CPLD EPM2210 System Controller.
D2	Loading	Illuminates when the MAX II CPLD EPM2210 System Controller is actively configuring the FPGA. Driven by the MAX II CPLD EPM2210 System Controller with the Embedded Blaster CPLD.
D3	Error	Illuminates when the MAX II CPLD EPM2210 System Controller fails to configure the FPGA. Driven by the MAX II CPLD EPM2210 System Controller.

■ Setup PCI Express Control DIP switch

The PCI Express Control DIP switch (SW3) is provided to enable or disable different configurations of the PCIe Connector. **Table 2-2** lists the switch controls and description.

Table 2-2 SW3 PCIe Control DIP Switch

Board Reference	*Signal Name*	*Description*	*Default*
SW3.1	PCIE_PRSNT2n_x1	On : Enable x1 presence detect Off: Disable x1 presence detect	Off
SW3.2	PCIE_PRSNT2n_x4	On : Enable x4 presence detect Off: Disable x4 presence detect	Off
SW3.3	PCIE_PRSNT2n_x8	On : Enable x8 presence detect Off: Disable x8 presence detect	On

■ Setup Configure Mode Control DIP switch

The Configure Mode Control DIP switch (SW2) is provided to specify the configuration mode of the FPGA. Because currently only one mode is supported, please set all positions to OFF as shown in **Figure 2-3**.

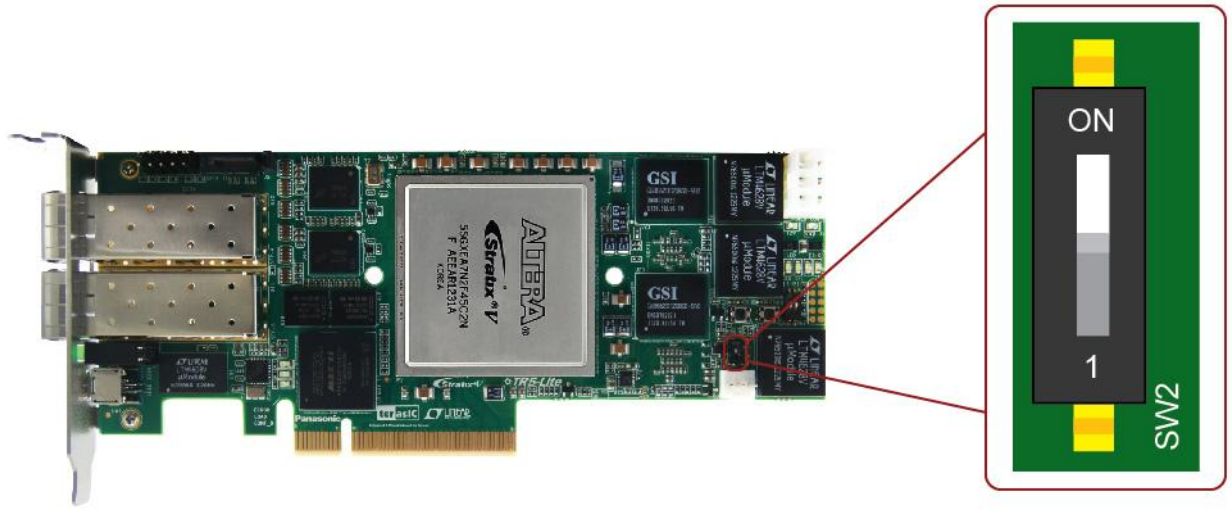


Figure 2-3 6-Position DIP switch for Configure Mode

■ Selecting Default Factory FPGA Configuration or User-defined Configuration

Users can select loading from default factory hardware or user-defined hardware through the use of SW1. The settings for the configurations are shown in **Table 2-3.**

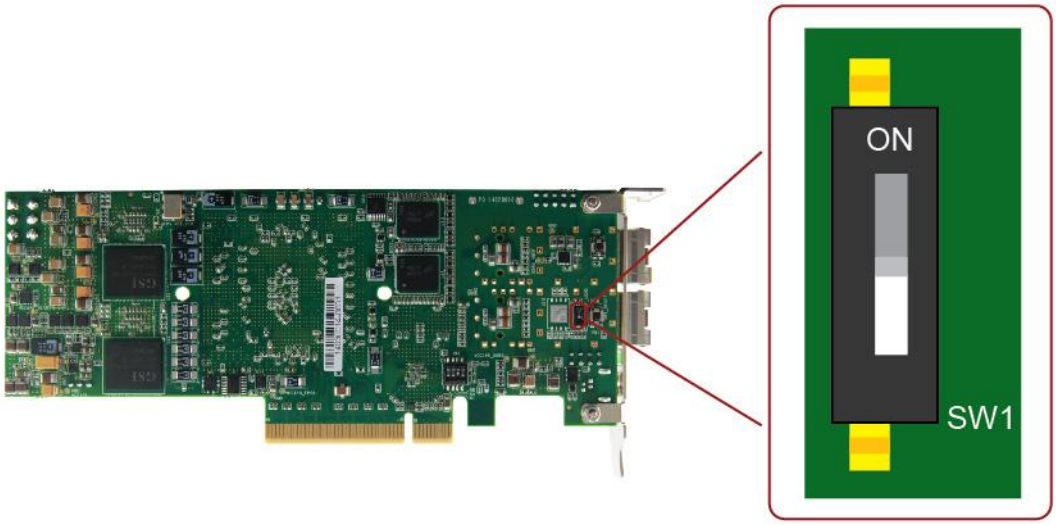


Figure 2-4 Position Dip Switch for FPGA Configuration

Table 2-3 SW1 FPGA Configuration Settings

Board Reference	*Description*	*Default*
SW1	On : User-defined Configuration Off: Factory Default Configuration	Off

2.3 General User Input/Output

This section describes the user I/O interface to the FPGA.

■ User Defined Push-buttons

The TR5-Lite board includes two user defined push-buttons that allow users to interact with the Stratix V GX device. Each push-button provides a high logic level or a low logic level when it is not pressed or pressed, respectively. Table 2-4 lists the board references, signal names and their corresponding Stratix V GX device pin numbers.

Table 2-4 Push-button Pin Assignments, Schematic Signal Names, and Functions

Board Reference	*Schematic Signal Name*	*Description*	*I/O Standard*	*Stratix V GX Pin Number*
PB2	BUTTON0	High Logic Level when the button is not pressed	2.5-V	PIN_A35
PB3	BUTTON1		2.5-V	PIN_A34

■ User-Defined DIP Switch

There is one 2-position DIP switch (SW4) on the TR5-Lite board to provide additional FPGA input control. Each switch is connected directly to a pin of the Stratix V GX FPGA. For 2-position DIP switch, when a switch is in the ON position, it provides a low logic level to the FPGA, as shown in Figure 2-5.

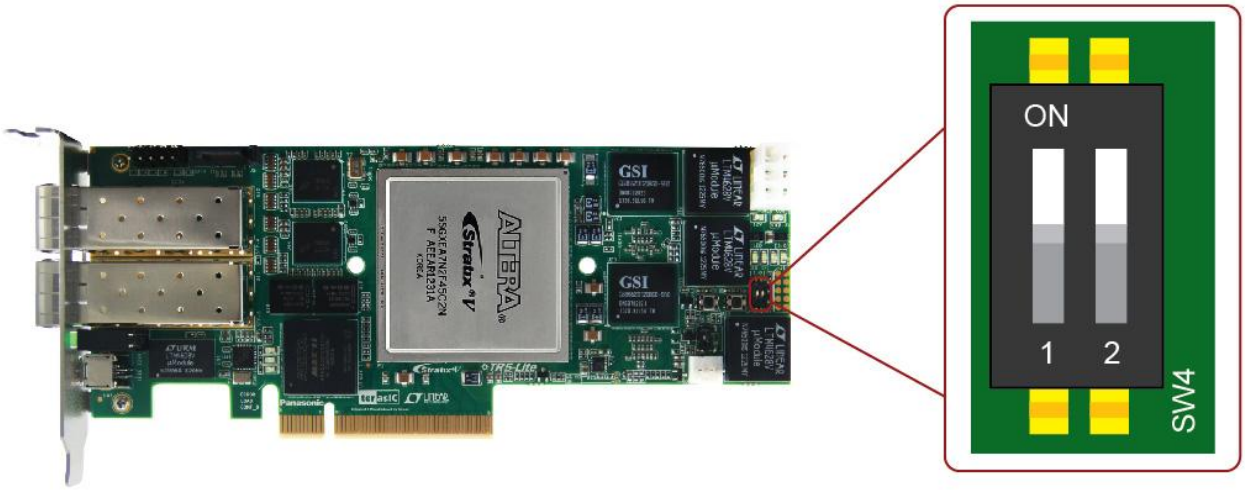


Figure 2-5 2-Position DIP switches

Table 2-5 lists the signal names and their corresponding Stratix V GX device pin numbers.

Table 2-5 DIP Switch Pin Assignments, Schematic Signal Names, and Functions

Board Reference	*Schematic Signal Name*	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
SW4	SLIDE_SW0	When the switch is in the ON position, a logic 0 is selected.	2.5-V	PIN_E33
SW4	SLIDE_SW1		2.5-V	PIN_D33

■ User-Defined LEDs

The TR5-Lite board consists of 4 user-controllable LEDs to allow status and debugging signals to be driven to the LEDs from the designs loaded into the Stratix V GX device. Each LED is driven directly by the Stratix V GX FPGA. The LED is turned on or off when the associated pins are driven to a low or high logic level, respectively. A list of the pin names on the FPGA that are connected to the LEDs is given in **Table 2-6**.

Table 2-6 User LEDs Pin Assignments, Schematic Signal Names, and Functions

Board Reference	*Schematic Signal Name*	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
D3	LED0	Driving a logic 0 on the I/O port turns the LED ON. Driving a logic 1 on the I/O port turns the LED OFF.	2.5-V	PIN_C33
D4	LED1		2.5-V	PIN_B32
D5	LED2		2.5-V	PIN_C34
D6	LED3		2.5-V	PIN_B34

2.4 Temperature Sensor and Fan Control

The TR5-Lite is equipped with a temperature sensor, MAX1619, which provides temperature sensing and over-temperature alert. These functions are accomplished by connecting the temperature sensor to the internal temperature sensing diode of the Stratix V GX device. The temperature status and alarm threshold registers of the temperature sensor can be programmed by a two-wire SMBus, which is connected to the Stratix V GX FPGA. In addition, the 7-bit POR slave address for this sensor is set to '0011000b'.

An optional 3-pin +12V fan located on J2 of the TR5-Lite board is intended to reduce the temperature of the FPGA. Users can control the fan to turn on/off depending on the measured system temperature. The FAN is turned on when the FAN_CTRL pins are driven to a high logic level.

The pin assignments for the associated interface are listed in **Table 2-7**.

Table 2-7 Temperature Sensor Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
TEMPDIODEp	Positive pin of temperature diode in Stratix V	2.5-V	PIN_P6

TEMPDIODEn	Negative pin of temperature diode in Stratix V	2.5-V	PIN_EP7
TEMP_CLK	SMBus clock	2.5-V	PIN_F35
TEMP_DATAT	SMBus data	2.5-V	PIN_E35
TEMP_OVERT_n	SMBus alert (interrupt)	2.5-V	PIN_H35
TEMP_INT_n	SMBus alert (interrupt)	2.5-V	PIN_G34
FAN_CTRL	Fan control	2.5-V	PIN_F34

2.5 Clock Circuit

The development board includes one 50 MHz and two programmable oscillators. **Figure 2-6** shows the default frequencies of on-board all external clocks going to the Stratix V GX FPGA. The figures also show an off-board external clock from PCI Express Host to the FPGA.

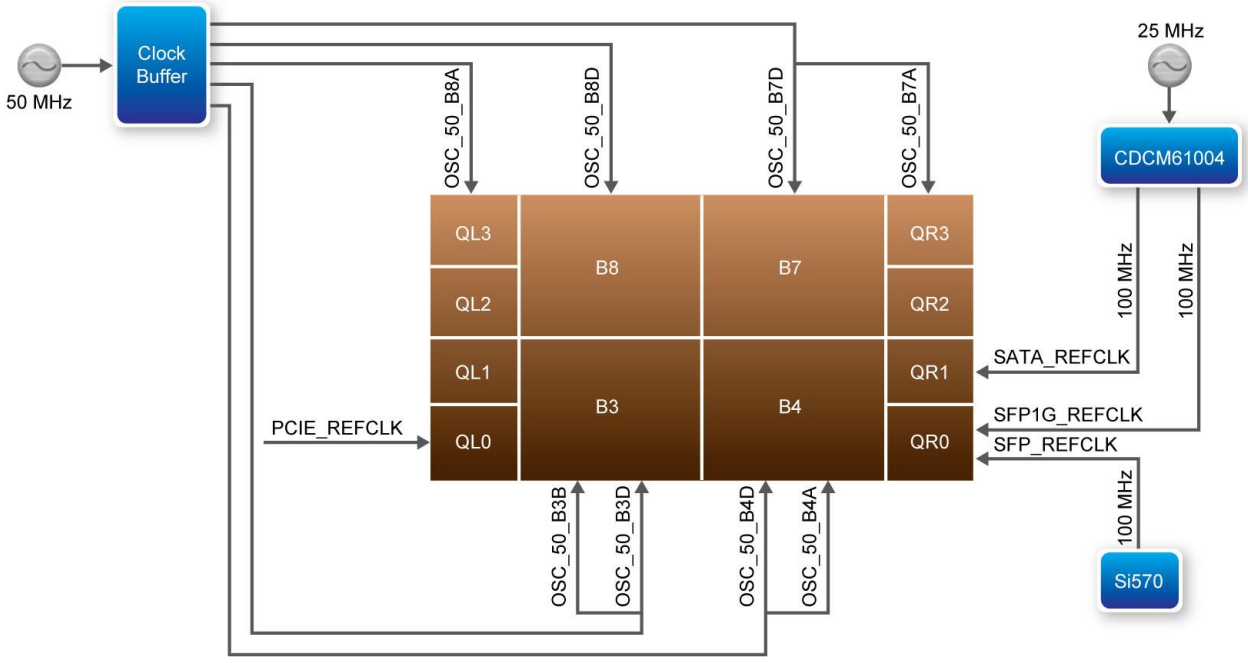


Figure 2-6 Clock circuit of the TR5-Lite

A clock buffer is used to duplicate the 50 MHz oscillator, so each bank of FPGA I/O bank 3/4/7/8 has two clock inputs. The two programming oscillators are low-jitter oscillators which are used to provide special and high quality clock signals for high-speed transceivers.

Table 2-8 lists the clock source, signal names, default frequency and their corresponding Stratix V GX device pin numbers.

Table 2-8 Clock Source, Signal Name, Default Frequency, Pin Assignments and Functions

Source	*Schematic Signal Name*	*Default Frequency*	*I/O Standard*	*Stratix V GX Pin Number*	*Application*
Y2	OSC_50_B3B	50.0 MHz	2.5-V	PIN_AW35	
	OSC_50_B3D		2.5-V	PIN_BC28	
	OSC_50_B4A		1.5-V	PIN_AP10	
	OSC_50_B4D		1.5-V	PIN_AY18	
	OSC_50_B7A		1.8-V	PIN_M9	
	OSC_50_B7D		1.8-V	PIN_J18	
	OSC_50_B8A		1.8-V	PIN_R36	
	OSC_50_B8D		1.8-V	PIN_R25	
U2	SFP_REFCLK _p	100.0 MHz	LVDS	PIN_AK7	10G SFP+
U46	SFP1G_REFCLK_p	100.0 MHz	LVDS	PIN_AH6	1G SFP+
U46	SATA_REFCLK_p	100.0 MHz	LVDS	PIN_AF7	SATA
J4	PCIE_REFCLK_p	From Host	LVDS	PIN_AK38	PCI Express

2.6 RS-422 Serial Port

The RS422 is designed to perform communication between boards, allowing a transmission speed of up to 20 Mbps. Figure 2-7 shows the RS-422 block diagram of the development board. The full-duplex LTC28255 is used to translate the RS-422 signal, and the 1394 is used as an external connector for the RS-422 signal.

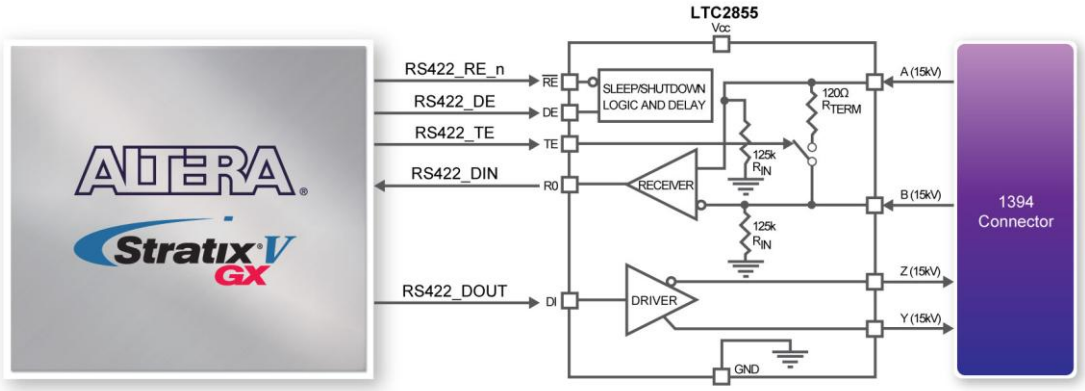


Figure 2-7 Block Diagram of RS-422

Table 2-9 lists the RS-422 pin assignments, signal names and functions.

Table 2-9 RS-422 Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix V GX Pin Number*
RS422_DE	Driver Enable. A high on DE enables the driver. A low input will force the driver outputs into a high impedance state.	2.5-V	PIN_AU23
RS422_DIN	Receiver Output. The data is send to FPGA.		PIN_AR24
RS422_DOUT	Driver Input. The data is sent from FPGA.		PIN_AV23
RS422_RE_n	Receiver Enable. A low enables the receiver. A high input forces the receiver output into a high impedance state.		PIN_AL24
RS422_TE	Internal Termination Resistance Enable. A high input will connect a termination resistor (120Ω typical) between pins A and B.		PIN_AL23

2.7 FLASH Memory

The development board has two 1Gb CFI-compatible synchronous flash devices for non-volatile

storage of FPGA configuration data, user application data, and user code space.

Each interface has a 16-bit data bus and the two devices combined allow for x32 FPP configuration. This device is part of the shared flash and MAX (FM) bus, which connects to the flash memory and MAX II CPLD (EPM2210) System Controller. **Figure 2-8** shows the connections between the Flash, MAX and Stratix V GX FPGA.

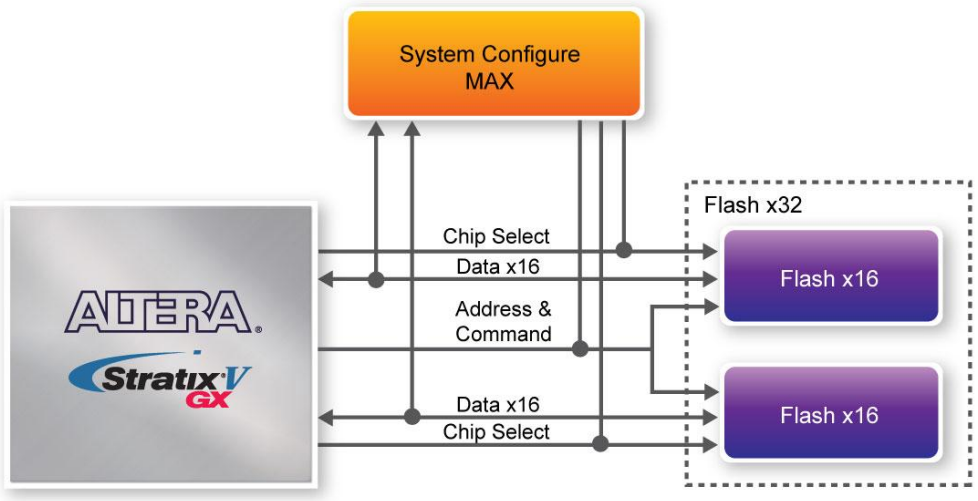


Figure 2-8 Connection between the Flash, Max and Stratix V GX FPG

Table 2-10 lists the flash pin assignments, signal names, and functions.

Table 2-10 Flash Memory Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
FSM_A0	Address bus	2.5-V	PIN_AG33
FSM_A1	Address bus	2.5-V	PIN_AN31
FSM_A2	Address bus	2.5-V	PIN_AP31
FSM_A3	Address bus	2.5-V	PIN_AH33
FSM_A4	Address bus	2.5-V	PIN_AG32
FSM_A5	Address bus	2.5-V	PIN_AF32
FSM_A6	Address bus	2.5-V	PIN_AV34
FSM_A7	Address bus	2.5-V	PIN_AM31
FSM_A8	Address bus	2.5-V	PIN_AP33
FSM_A9	Address bus	2.5-V	PIN_AF34
FSM_A10	Address bus	2.5-V	PIN_AR31

FSM_A11	Address bus	2.5-V	PIN_AE34
FSM_A12	Address bus	2.5-V	PIN_AW33
FSM_A13	Address bus	2.5-V	PIN_AN33
FSM_A14	Address bus	2.5-V	PIN_AJ32
FSM_A15	Address bus	2.5-V	PIN_AR32
FSM_A16	Address bus	2.5-V	PIN_AE30
FSM_A17	Address bus	2.5-V	PIN_AE29
FSM_A18	Address bus	2.5-V	PIN_AK32
FSM_A19	Address bus	2.5-V	PIN_AJ33
FSM_A20	Address bus	2.5-V	PIN_AE32
FSM_A21	Address bus	2.5-V	PIN_AE31
FSM_A22	Address bus	2.5-V	PIN_AK33
FSM_A23	Address bus	2.5-V	PIN_AD32
FSM_A24	Address bus	2.5-V	PIN_AD33
FSM_A25	Address bus	2.5-V	PIN_AM32
FSM_A26	Address bus	2.5-V	PIN_AF31
FSM_D0	Data bus	2.5-V	PIN_AF28
FSM_D1	Data bus	2.5-V	PIN_AG30
FSM_D2	Data bus	2.5-V	PIN_AG25
FSM_D3	Data bus	2.5-V	PIN_AK29
FSM_D4	Data bus	2.5-V	PIN_BA29
FSM_D5	Data bus	2.5-V	PIN_AG26
FSM_D6	Data bus	2.5-V	PIN_AG27
FSM_D7	Data bus	2.5-V	PIN_AE28
FSM_D8	Data bus	2.5-V	PIN_AG29
FSM_D9	Data bus	2.5-V	PIN_AK30
FSM_D10	Data bus	2.5-V	PIN_AG28
FSM_D11	Data bus	2.5-V	PIN_AF29
FSM_D12	Data bus	2.5-V	PIN_AJ29
FSM_D13	Data bus	2.5-V	PIN_AE27
FSM_D14	Data bus	2.5-V	PIN_AP28
FSM_D15	Data bus	2.5-V	PIN_BD31
FSM_D16	Data bus	2.5-V	PIN_BC31
FSM_D17	Data bus	2.5-V	PIN_BB32
FSM_D18	Data bus	2.5-V	PIN_BB30
FSM_D19	Data bus	2.5-V	PIN_BA31
FSM_D20	Data bus	2.5-V	PIN_AW30
FSM_D21	Data bus	2.5-V	PIN_BA30
FSM_D22	Data bus	2.5-V	PIN_AL29
FSM_D23	Data bus	2.5-V	PIN_AR29

FSM_D24	Data bus	2.5-V	PIN_BC32
FSM_D25	Data bus	2.5-V	PIN_AM29
FSM_D26	Data bus	2.5-V	PIN_BD32
FSM_D27	Data bus	2.5-V	PIN_AY30
FSM_D28	Data bus	2.5-V	PIN_AY31
FSM_D29	Data bus	2.5-V	PIN_AN28
FSM_D30	Data bus	2.5-V	PIN_AL30
FSM_D31	Data bus	2.5-V	PIN_AL31
FLASH_CLK	Clock	2.5-V	PIN_AU31
FLASH_RESET_n	Reset	2.5-V	PIN_AV31
FLASH_CE_n[0]	Chip enable of of flash-0	2.5-V	PIN_AJ31
FLASH_CE_n[1]	Chip enable of of flash-1	2.5-V	PIN_AW32
FLASH_OE_n	Output enable	2.5-V	PIN_AU30
FLASH_WE_n	Write enable	2.5-V	PIN_AH30
FLASH_ADV_n	Address valid	2.5-V	PIN_AT29
FLASH_RDY_BSY_n[0]	Ready of flash-0	2.5-V	PIN_AJ30
FLASH_RDY_BSY_n[0]	Ready of flash-1	2.5-V	PIN_AV32

2.8 DDR3 SDRAM

The development board supports two independent banks of DDR3 SDRAM which totals 2GB in memory. Each bank comprises of two x8 DDR3 devices. The DDR3 signals are connected to the vertical I/O banks on the bottom edge of the FPGA. The DDR3 devices shipped with this board are running at 667 MHz, for a total theoretical bandwidth of over 42.68 Gbps. **Figure 2-9** shows the connections between the DDR3 and Stratix V GX FPGA.

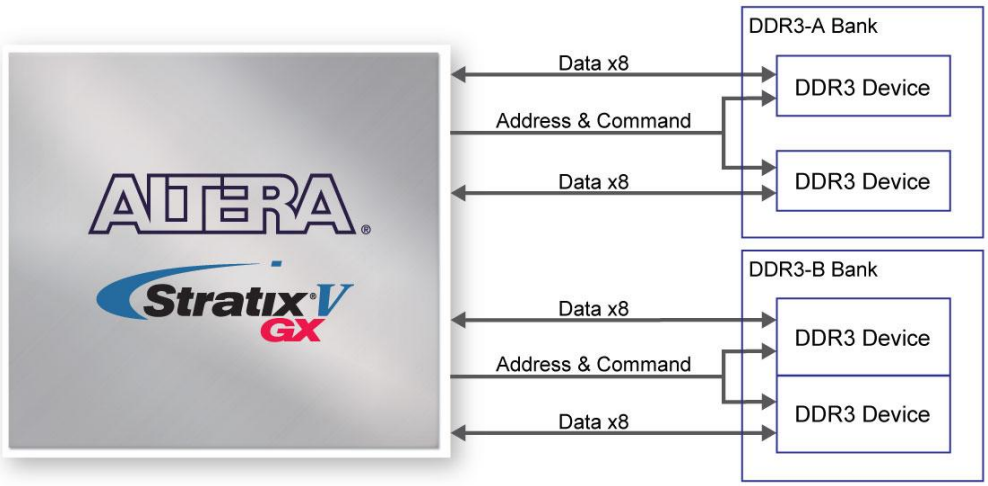


Figure 2-9 Connection between the DDR3 and Stratix V GX FPGA

The pin assignments for DDR3 Bank-A and Bank-B are listed in **Table 2-11** and **Table 2-12**, in respectively.

Table 2-11 DDR3-A Bank Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix V GX Pin Number*
DDR3A_DQ0	Data [0]	SSTL-15 Class I	PIN_AG9
DDR3A_DQ1	Data [1]	SSTL-15 Class I	PIN_AJ10
DDR3A_DQ2	Data [2]	SSTL-15 Class I	PIN_AL11
DDR3A_DQ3	Data [3]	SSTL-15 Class I	PIN_AJ11
DDR3A_DQ4	Data [4]	SSTL-15 Class I	PIN_AG11
DDR3A_DQ5	Data [5]	SSTL-15 Class I	PIN_AH10
DDR3A_DQ6	Data [6]	SSTL-15 Class I	PIN_AG10
DDR3A_DQ7	Data [7]	SSTL-15 Class I	PIN_AL12
DDR3A_DQ8	Data [8]	SSTL-15 Class I	PIN_AN12
DDR3A_DQ9	Data [9]	SSTL-15 Class I	PIN_AV10
DDR3A_DQ10	Data [10]	SSTL-15 Class I	PIN_AR13
DDR3A_DQ11	Data [11]	SSTL-15 Class I	PIN_AR12
DDR3A_DQ12	Data [12]	SSTL-15 Class I	PIN_AM13
DDR3A_DQ13	Data [13]	SSTL-15 Class I	PIN_AP12
DDR3A_DQ14	Data [14]	SSTL-15 Class I	PIN_AP13
DDR3A_DQ15	Data [15]	SSTL-15 Class I	PIN_AU9
DDR3A_QDS0	Data Strobe p[0]	Differential 1.5-V SSTL Class I	PIN_AG12
DDR3A_QDS_n0	Data Strobe n[0]	Differential 1.5-V SSTL Class I	PIN_AH12
DDR3A_QDS1	Data Strobe p[1]	Differential 1.5-V SSTL Class I	PIN_AU10
DDR3A_QDS_n1	Data Strobe n[1]	Differential 1.5-V SSTL Class I	PIN_AV11

DDR3A_DM0	Data Mask [0]	SSTL-15 Class I	PIN_AJ12
DDR3A_DM0	Data Mask [1]	SSTL-15 Class I	PIN_AU11
DDR3A_A0	Address [0]	SSTL-15 Class I	PIN_AW10
DDR3A_A1	Address [1]	SSTL-15 Class I	PIN_AK15
DDR3A_A2	Address [2]	SSTL-15 Class I	PIN_AJ14
DDR3A_A3	Address [3]	SSTL-15 Class I	PIN_AT12
DDR3A_A4	Address [4]	SSTL-15 Class I	PIN_AE18
DDR3A_A5	Address [5]	SSTL-15 Class I	PIN_AF16
DDR3A_A6	Address [6]	SSTL-15 Class I	PIN_AE16
DDR3A_A7	Address [7]	SSTL-15 Class I	PIN_Ah15
DDR3A_A8	Address [8]	SSTL-15 Class I	PIN_AE17
DDR3A_A9	Address [9]	SSTL-15 Class I	PIN_AK18
DDR3A_A10	Address [10]	SSTL-15 Class I	PIN_BC14
DDR3A_A11	Address [11]	SSTL-15 Class I	PIN_AG14
DDR3A_A12	Address [12]	SSTL-15 Class I	PIN_AY15
DDR3A_A13	Address [13]	SSTL-15 Class I	PIN_AG15
DDR3A_A14	Address [14]	SSTL-15 Class I	PIN_AF11
DDR3A_A15	Address [15]	SSTL-15 Class I	PIN_BD16
DDR3A_RAS_n	Row Address Strobe	SSTL-15 Class I	PIN_BC13
DDR3A_CAS_n	Column Address Strobe	SSTL-15 Class I	PIN_BB11
DDR3A_BA0	Bank Address [0]	SSTL-15 Class I	PIN_BC16
DDR3A_BA1	Bank Address [1]	SSTL-15 Class I	PIN_AK12
DDR3A_BA2	Bank Address [2]	SSTL-15 Class I	PIN_AY13
DDR3A_CK	Clock p0	Differential 1.5-V SSTL Class I	PIN_AG16
DDR3A_CK_n	Clock n0	Differential 1.5-V SSTL Class I	PIN_AF17
DDR3A_CKE0	Clock Enable pin 0	SSTL-18 Class I	PIN_BD14
DDR3A_CKE1	Clock Enable pin 1	SSTL-18 Class I	PIN_AY10
DDR3A_ODT0	On Die Termination[0]	SSTL-15 Class I	PIN_BA13
DDR3A_ODT1	On Die Termination[1]	SSTL-15 Class I	PIN_BD13
DDR3A_WE_n	Write Enable	SSTL-15 Class I	PIN_BB14
DDR3A_CS_n0	Chip Select [0]	SSTL-15 Class I	PIN_BB15
DDR3A_CS_n1	Chip Select [1]	SSTL-15 Class I	PIN_BA15
DDR3A_RESET_n	Chip Reset	SSTL-15 Class I	PIN_AE15

Table 2-12 DDR3-B Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
DDR3B_DQ0	Data [0]	SSTL-15 Class I	PIN_AN20
DDR3B_DQ1	Data [1]	SSTL-15 Class I	PIN_AL18
DDR3B_DQ2	Data [2]	SSTL-15 Class I	PIN_AN19
DDR3B_DQ3	Data [3]	SSTL-15 Class I	PIN_AP19

DDR3B_DQ4	Data [4]	SSTL-15 Class I	PIN_AW19
DDR3B_DQ5	Data [5]	SSTL-15 Class I	PIN_AM19
DDR3B_DQ6	Data [6]	SSTL-15 Class I	PIN_AL19
DDR3B_DQ7	Data [7]	SSTL-15 Class I	PIN_AU18
DDR3B_DQ8	Data [8]	SSTL-15 Class I	PIN_AM17
DDR3B_DQ9	Data [9]	SSTL-15 Class I	PIN_AJ16
DDR3B_DQ10	Data [10]	SSTL-15 Class I	PIN_AL17
DDR3B_DQ11	Data [11]	SSTL-15 Class I	PIN_AG18
DDR3B_DQ12	Data [12]	SSTL-15 Class I	PIN_AJ17
DDR3B_DQ13	Data [13]	SSTL-15 Class I	PIN_AG17
DDR3B_DQ14	Data [14]	SSTL-15 Class I	PIN_AK17
DDR3B_DQ15	Data [15]	SSTL-15 Class I	PIN_AJ15
DDR3B_DQS0	Data Strobe p[0]	Differential 1.5-V SSTL Class I	PIN_AP18
DDR3B_DQS_n0	Data Strobe n[0]	Differential 1.5-V SSTL Class I	PIN_AR19
DDR3B_DQS1	Data Strobe p[1]	Differential 1.5-V SSTL Class I	PIN_AH18
DDR3B_DQS_n1	Data Strobe n[1]	Differential 1.5-V SSTL Class I	PIN_AH19
DDR3B_DM0	Data Mask [0]	SSTL-15 Class I	PIN_AV19
DDR3B_DM1	Data Mask [1]	SSTL-15 Class I	PIN_AJ18
DDR3B_A0	Address [0]	SSTL-15 Class I	PIN_AV13
DDR3B_A1	Address [1]	SSTL-15 Class I	PIN_AT14
DDR3B_A2	Address [2]	SSTL-15 Class I	PIN_AT17
DDR3B_A3	Address [3]	SSTL-15 Class I	PIN_AU12
DDR3B_A4	Address [4]	SSTL-15 Class I	PIN_AL15
DDR3B_A5	Address [5]	SSTL-15 Class I	PIN_AU14
DDR3B_A6	Address [6]	SSTL-15 Class I	PIN_AL16
DDR3B_A7	Address [7]	SSTL-15 Class I	PIN_AR17
DDR3B_A8	Address [8]	SSTL-15 Class I	PIN_AN17
DDR3B_A9	Address [9]	SSTL-15 Class I	PIN_AP15
DDR3B_A10	Address [10]	SSTL-15 Class I	PIN_BC17
DDR3B_A11	Address [11]	SSTL-15 Class I	PIN_AR15
DDR3B_A12	Address [12]	SSTL-15 Class I	PIN_AY19
DDR3B_A13	Address [13]	SSTL-15 Class I	PIN_AR14
DDR3B_A14	Address [14]	SSTL-15 Class I	PIN_AM16
DDR3B_A15	Address [15]	SSTL-15 Class I	PIN_AT15
DDR3B_RAS_n	Row Address Strobe	SSTL-15 Class I	PIN_BA16
DDR3B_CAS_n	Column Address Strobe	SSTL-15 Class I	PIN_AP16
DDR3B_BA0	Bank Address [0]	SSTL-15 Class I	PIN_AW13
DDR3B_BA1	Bank Address [1]	SSTL-15 Class I	PIN_AU13
DDR3B_BA2	Bank Address [2]	SSTL-15 Class I	PIN_AV14
DDR3B_CK	Clock p0	Differential 1.5-V SSTL Class I	PIN_BC19
DDR3B_CK_n	Clock n0	Differential 1.5-V SSTL Class I	PIN_BD19
DDR3B_CKE0	Clock Enable pin 0	SSTL-15 Class I	PIN_BD17

DDR3B_CKE1	Clock Enable pin 1	SSTL-15 Class I	PIN_AH13
DDR3B_ODT0	On Die Termination[0]	SSTL-15 Class I	PIN_AR16
DDR3B_ODT1	On Die Termination[1]	SSTL-15 Class I	PIN_AR18
DDR3B_WE_n	Write Enable	SSTL-15 Class I	PIN_AU15
DDR3B_CS_n0	Chip Select [0]	SSTL-15 Class I	PIN_BA19
DDR3B_CS_n1	Chip Select [1]	SSTL-15 Class I	PIN_AW14
DDR3B_RESET_n	Chip Reset	SSTL-15 Class I	PIN_AN15

2.9 QDRII+ SRAM

The development board supports four independent QDRII+ SRAM memory devices for very-high speed and low-latency memory access. Each of QDRII+ has a x18 interface, providing addressing to a device of up to a 8MB (not include parity bits). The QDRII+ has separate read and write data ports with DDR signaling at up to 550 MHz .

Table 2-13, Table 2-14, Table 2-15 and Table 2-16 lists the QDRII+ SRAM Bank A, B, C and D pin assignments, signal names relative to the Stratix I GX device, in respectively.

Table 2-13 QDRII+ SRAM A Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
QDRIIA_A0	Address bus[0]	1.8-V HSTL Class I	PIN_R15
QDRIIA_A1	Address bus[1]	1.8-V HSTL Class I	PIN_J13
QDRIIA_A2	Address bus[2]	1.8-V HSTL Class I	PIN_U15
QDRIIA_A3	Address bus[3]	1.8-V HSTL Class I	PIN_W14
QDRIIA_A4	Address bus[4]	1.8-V HSTL Class I	PIN_V13
QDRIIA_A5	Address bus[5]	1.8-V HSTL Class I	PIN_W16
QDRIIA_A6	Address bus[6]	1.8-V HSTL Class I	PIN_G14
QDRIIA_A7	Address bus[7]	1.8-V HSTL Class I	PIN_G13
QDRIIA_A8	Address bus[8]	1.8-V HSTL Class I	PIN_H13
QDRIIA_A9	Address bus[9]	1.8-V HSTL Class I	PIN_H15
QDRIIA_A10	Address bus[10]	1.8-V HSTL Class I	PIN_H14
QDRIIA_A11	Address bus[11]	1.8-V HSTL Class I	PIN_K13
QDRIIA_A12	Address bus[12]	1.8-V HSTL Class I	PIN_K14
QDRIIA_A13	Address bus[13]	1.8-V HSTL Class I	PIN_L15
QDRIIA_A14	Address bus[14]	1.8-V HSTL Class I	PIN_J16
QDRIIA_A15	Address bus[15]	1.8-V HSTL Class I	PIN_K16
QDRIIA_A16	Address bus[16]	1.8-V HSTL Class I	PIN_F13

QDRIIA_A17	Address bus[17]	1.8-V HSTL Class I	PIN_Y17
QDRIIA_A18	Address bus[18]	1.8-V HSTL Class I	PIN_T15
QDRIIA_A19	Address bus[19]	1.8-V HSTL Class I	PIN_Y16
QDRIIA_A20	Address bus[20]	1.8-V HSTL Class I	PIN_W17
QDRIIA_D0	Write data bus[0]	1.8-V HSTL Class I	PIN_A11
QDRIIA_D1	Write data bus[1]	1.8-V HSTL Class I	PIN_A10
QDRIIA_D2	Write data bus[2]	1.8-V HSTL Class I	PIN_B11
QDRIIA_D3	Write data bus[3]	1.8-V HSTL Class I	PIN_C12
QDRIIA_D4	Write data bus[4]	1.8-V HSTL Class I	PIN_T13
QDRIIA_D5	Write data bus[5]	1.8-V HSTL Class I	PIN_U12
QDRIIA_D6	Write data bus[6]	1.8-V HSTL Class I	PIN_T14
QDRIIA_D7	Write data bus[7]	1.8-V HSTL Class I	PIN_U11
QDRIIA_D8	Write data bus[8]	1.8-V HSTL Class I	PIN_U14
QDRIIA_D9	Write data bus[9]	1.8-V HSTL Class I	PIN_E12
QDRIIA_D10	Write data bus[10]	1.8-V HSTL Class I	PIN_E11
QDRIIA_D11	Write data bus[11]	1.8-V HSTL Class I	PIN_D12
QDRIIA_D12	Write data bus[12]	1.8-V HSTL Class I	PIN_M13
QDRIIA_D13	Write data bus[13]	1.8-V HSTL Class I	PIN_D11
QDRIIA_D14	Write data bus[14]	1.8-V HSTL Class I	PIN_N14
QDRIIA_D15	Write data bus[15]	1.8-V HSTL Class I	PIN_P13
QDRIIA_D16	Write data bus[16]	1.8-V HSTL Class I	PIN_G11
QDRIIA_D17	Write data bus[17]	1.8-V HSTL Class I	PIN_C10
QDRIIA_Q0	Read Data bus[0]	1.8-V HSTL Class I	PIN_M11
QDRIIA_Q1	Read Data bus[1]	1.8-V HSTL Class I	PIN_N11
QDRIIA_Q2	Read Data bus[2]	1.8-V HSTL Class I	PIN_V9
QDRIIA_Q3	Read Data bus[3]	1.8-V HSTL Class I	PIN_V10
QDRIIA_Q4	Read Data bus[4]	1.8-V HSTL Class I	PIN_T11
QDRIIA_Q5	Read Data bus[5]	1.8-V HSTL Class I	PIN_U9
QDRIIA_Q6	Read Data bus[6]	1.8-V HSTL Class I	PIN_T9
QDRIIA_Q7	Read Data bus[7]	1.8-V HSTL Class I	PIN_R10
QDRIIA_Q8	Read Data bus[8]	1.8-V HSTL Class I	PIN_T10
QDRIIA_Q9	Read Data bus[9]	1.8-V HSTL Class I	PIN_L11
QDRIIA_Q10	Read Data bus[10]	1.8-V HSTL Class I	PIN_M12
QDRIIA_Q11	Read Data bus[11]	1.8-V HSTL Class I	PIN_L12
QDRIIA_Q12	Read Data bus[12]	1.8-V HSTL Class I	PIN_H10
QDRIIA_Q13	Read Data bus[13]	1.8-V HSTL Class I	PIN_J10
QDRIIA_Q14	Read Data bus[14]	1.8-V HSTL Class I	PIN_H11
QDRIIA_Q15	Read Data bus[15]	1.8-V HSTL Class I	PIN_H12
QDRIIA_Q16	Read Data bus[16]	1.8-V HSTL Class I	PIN_P12
QDRIIA_Q17	Read Data bus[17]	1.8-V HSTL Class I	PIN_R12
QDRIIA_BWS_n0	Byte Write select[0]	1.8-V HSTL Class I	PIN_R13
QDRIIA_BWS_n1	Byte Write select[1]	1.8-V HSTL Class I	PIN_P14

QDRIIA_K_P	Clock P	Differentail 1.8-V HSTL Class I	PIN_V12
QDRIIA_K_N	Clock N	Differentail 1.8-V HSTL Class I	PIN_V11
QDRIIA_CQ_P	Echo clock P	1.8-V HSTL Class I	PIN_T12
QDRIIA_CQ_N	Echo clock N	1.8-V HSTL Class I	PIN_K11
QDRIIA_RPS_n	Report Select	1.8-V HSTL Class I	PIN_K12
QDRIIA_WPS_n	Write Port Select	1.8-V HSTL Class I	PIN_F11
QDRIIA_DOFF_n	DLL enable	1.8-V HSTL Class I	PIN_B10
QDRIIA_ODT ?		1.8-V HSTL Class I	PIN_F10
QDRII_QVLD?		1.8-V HSTL Class I	PIN_G10

Table 2-14 QDRII+ SRAM B Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
QDRIIB_A0	Address bus[0]	1.8-V HSTL Class I	PIN_F22
QDRIIB_A1	Address bus[1]	1.8-V HSTL Class I	PIN_A14
QDRIIB_A2	Address bus[2]	1.8-V HSTL Class I	PIN_F23
QDRIIB_A3	Address bus[3]	1.8-V HSTL Class I	PIN_E21
QDRIIB_A4	Address bus[4]	1.8-V HSTL Class I	PIN_G23
QDRIIB_A5	Address bus[5]	1.8-V HSTL Class I	PIN_C21
QDRIIB_A6	Address bus[6]	1.8-V HSTL Class I	PIN_D20
QDRIIB_A7	Address bus[7]	1.8-V HSTL Class I	PIN_A20
QDRIIB_A8	Address bus[8]	1.8-V HSTL Class I	PIN_D21
QDRIIB_A9	Address bus[9]	1.8-V HSTL Class I	PIN_E20
QDRIIB_A10	Address bus[10]	1.8-V HSTL Class I	PIN_B20
QDRIIB_A11	Address bus[11]	1.8-V HSTL Class I	PIN_A22
QDRIIB_A12	Address bus[12]	1.8-V HSTL Class I	PIN_F21
QDRIIB_A13	Address bus[13]	1.8-V HSTL Class I	PIN_A16
QDRIIB_A14	Address bus[14]	1.8-V HSTL Class I	PIN_B16
QDRIIB_A15	Address bus[15]	1.8-V HSTL Class I	PIN_D17
QDRIIB_A16	Address bus[16]	1.8-V HSTL Class I	PIN_E17
QDRIIB_A17	Address bus[17]	1.8-V HSTL Class I	PIN_B22
QDRIIB_A18	Address bus[18]	1.8-V HSTL Class I	PIN_E24
QDRIIB_A19	Address bus[19]	1.8-V HSTL Class I	PIN_C22
QDRIIB_A20	Address bus[20]	1.8-V HSTL Class I	PIN_D24
QDRIIB_D0	Write data bus[0]	1.8-V HSTL Class I	PIN_N20
QDRIIB_D1	Write data bus[1]	1.8-V HSTL Class I	PIN_P20
QDRIIB_D2	Write data bus[2]	1.8-V HSTL Class I	PIN_B19
QDRIIB_D3	Write data bus[3]	1.8-V HSTL Class I	PIN_L20
QDRIIB_D4	Write data bus[4]	1.8-V HSTL Class I	PIN_E18
QDRIIB_D5	Write data bus[5]	1.8-V HSTL Class I	PIN_D18

QDRIIB_D6	Write data bus[6]	1.8-V HSTL Class I	PIN_H19
QDRIIB_D7	Write data bus[7]	1.8-V HSTL Class I	PIN_G19
QDRIIB_D8	Write data bus[8]	1.8-V HSTL Class I	PIN_F19
QDRIIB_D9	Write data bus[9]	1.8-V HSTL Class I	PIN_N19
QDRIIB_D10	Write data bus[10]	1.8-V HSTL Class I	PIN_T18
QDRIIB_D11	Write data bus[11]	1.8-V HSTL Class I	PIN_T19
QDRIIB_D12	Write data bus[12]	1.8-V HSTL Class I	PIN_V18
QDRIIB_D13	Write data bus[13]	1.8-V HSTL Class I	PIN_U18
QDRIIB_D14	Write data bus[14]	1.8-V HSTL Class I	PIN_T20
QDRIIB_D15	Write data bus[15]	1.8-V HSTL Class I	PIN_K19
QDRIIB_D16	Write data bus[16]	1.8-V HSTL Class I	PIN_B17
QDRIIB_D17	Write data bus[17]	1.8-V HSTL Class I	PIN_W18
QDRIIB_Q0	Read Data bus[0]	1.8-V HSTL Class I	PIN_V17
QDRIIB_Q1	Read Data bus[1]	1.8-V HSTL Class I	PIN_U17
QDRIIB_Q2	Read Data bus[2]	1.8-V HSTL Class I	PIN_T17
QDRIIB_Q3	Read Data bus[3]	1.8-V HSTL Class I	PIN_T16
QDRIIB_Q4	Read Data bus[4]	1.8-V HSTL Class I	PIN_R16
QDRIIB_Q5	Read Data bus[5]	1.8-V HSTL Class I	PIN_M14
QDRIIB_Q6	Read Data bus[6]	1.8-V HSTL Class I	PIN_M15
QDRIIB_Q7	Read Data bus[7]	1.8-V HSTL Class I	PIN_N16
QDRIIB_Q8	Read Data bus[8]	1.8-V HSTL Class I	PIN_P16
QDRIIB_Q9	Read Data bus[9]	1.8-V HSTL Class I	PIN_C13
QDRIIB_Q10	Read Data bus[10]	1.8-V HSTL Class I	PIN_J15
QDRIIB_Q11	Read Data bus[11]	1.8-V HSTL Class I	PIN_D14
QDRIIB_Q12	Read Data bus[12]	1.8-V HSTL Class I	PIN_A13
QDRIIB_Q13	Read Data bus[13]	1.8-V HSTL Class I	PIN_B13
QDRIIB_Q14	Read Data bus[14]	1.8-V HSTL Class I	PIN_D15
QDRIIB_Q15	Read Data bus[15]	1.8-V HSTL Class I	PIN_F14
QDRIIB_Q16	Read Data bus[16]	1.8-V HSTL Class I	PIN_E14
QDRIIB_Q17	Read Data bus[17]	1.8-V HSTL Class I	PIN_L14
QDRIIB_BWS_n0	Byte Write select[0]	1.8-V HSTL Class I	PIN_J19
QDRIIB_BWS_n1	Byte Write select[1]	1.8-V HSTL Class I	PIN_A19
QDRIIB_K_p	Clock P	Differential 1.8-V HSTL Class I	PIN_C19
QDRIIB_K_n	Clock N	Differential 1.8-V HSTL Class I	PIN_C18
QDRIIB_CQ_p	Echo clock P	1.8-V HSTL Class I	PIN_K15
QDRIIB_CQ_n	Echo clock N	1.8-V HSTL Class I	PIN_E15
QDRIIB_RPS_n	Report Select	1.8-V HSTL Class I	PIN_M20
QDRIIB_WPS_n	Write Port Select	1.8-V HSTL Class I	PIN_A17
QDRIIB_DOFF_n	PLL Turn Off	1.8-V HSTL Class I	PIN_B14
QDRIIB_ODT	On-Die Termination Input	1.8-V HSTL Class I	PIN_C15
QDRIIB_QVLD	Valid Output Indicator	1.8-V HSTL Class I	PIN_C16

Table 2-15 QDRII+ SRAM C Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
QDRIIC_A0	Address bus[0]	1.8-V HSTL Class I	PIN_J27
QDRIIC_A1	Address bus[1]	1.8-V HSTL Class I	PIN_H28
QDRIIC_A2	Address bus[2]	1.8-V HSTL Class I	PIN_A26
QDRIIC_A3	Address bus[3]	1.8-V HSTL Class I	PIN_D27
QDRIIC_A4	Address bus[4]	1.8-V HSTL Class I	PIN_C28
QDRIIC_A5	Address bus[5]	1.8-V HSTL Class I	PIN_C27
QDRIIC_A6	Address bus[6]	1.8-V HSTL Class I	PIN_D29
QDRIIC_A7	Address bus[7]	1.8-V HSTL Class I	PIN_F26
QDRIIC_A8	Address bus[8]	1.8-V HSTL Class I	PIN_F28
QDRIIC_A9	Address bus[9]	1.8-V HSTL Class I	PIN_A29
QDRIIC_A10	Address bus[10]	1.8-V HSTL Class I	PIN_B28
QDRIIC_A11	Address bus[11]	1.8-V HSTL Class I	PIN_G28
QDRIIC_A12	Address bus[12]	1.8-V HSTL Class I	PIN_H26
QDRIIC_A13	Address bus[13]	1.8-V HSTL Class I	PIN_F29
QDRIIC_A14	Address bus[14]	1.8-V HSTL Class I	PIN_G29
QDRIIC_A15	Address bus[15]	1.8-V HSTL Class I	PIN_H29
QDRIIC_A16	Address bus[16]	1.8-V HSTL Class I	PIN_H27
QDRIIC_A17	Address bus[17]	1.8-V HSTL Class I	PIN_E29
QDRIIC_A18	Address bus[18]	1.8-V HSTL Class I	PIN_B26
QDRIIC_A19	Address bus[19]	1.8-V HSTL Class I	PIN_E27
QDRIIC_A20	Address bus[20]	1.8-V HSTL Class I	PIN_A28
QDRIIC_D0	Write data bus[0]	1.8-V HSTL Class I	PIN_V19
QDRIIC_D1	Write data bus[1]	1.8-V HSTL Class I	PIN_F20
QDRIIC_D2	Write data bus[2]	1.8-V HSTL Class I	PIN_H20
QDRIIC_D3	Write data bus[3]	1.8-V HSTL Class I	PIN_R22
QDRIIC_D4	Write data bus[4]	1.8-V HSTL Class I	PIN_K20
QDRIIC_D5	Write data bus[5]	1.8-V HSTL Class I	PIN_H22
QDRIIC_D6	Write data bus[6]	1.8-V HSTL Class I	PIN_K22
QDRIIC_D7	Write data bus[7]	1.8-V HSTL Class I	PIN_J21
QDRIIC_D8	Write data bus[8]	1.8-V HSTL Class I	PIN_J22
QDRIIC_D9	Write data bus[9]	1.8-V HSTL Class I	PIN_R21
QDRIIC_D10	Write data bus[10]	1.8-V HSTL Class I	PIN_U21
QDRIIC_D11	Write data bus[11]	1.8-V HSTL Class I	PIN_V21
QDRIIC_D12	Write data bus[12]	1.8-V HSTL Class I	PIN_U20
QDRIIC_D13	Write data bus[13]	1.8-V HSTL Class I	PIN_V20
QDRIIC_D14	Write data bus[14]	1.8-V HSTL Class I	PIN_H21
QDRIIC_D15	Write data bus[15]	1.8-V HSTL Class I	PIN_G22

QDRIIC_D16	Write data bus[16]	1.8-V HSTL Class I	PIN_G20
QDRIIC_D17	Write data bus[17]	1.8-V HSTL Class I	PIN_T21
QDRIIC_Q0	Read Data bus[0]	1.8-V HSTL Class I	PIN_U24
QDRIIC_Q1	Read Data bus[1]	1.8-V HSTL Class I	PIN_P24
QDRIIC_Q2	Read Data bus[2]	1.8-V HSTL Class I	PIN_M23
QDRIIC_Q3	Read Data bus[3]	1.8-V HSTL Class I	PIN_U23
QDRIIC_Q4	Read Data bus[4]	1.8-V HSTL Class I	PIN_T23
QDRIIC_Q5	Read Data bus[5]	1.8-V HSTL Class I	PIN_T24
QDRIIC_Q6	Read Data bus[6]	1.8-V HSTL Class I	PIN_P23
QDRIIC_Q7	Read Data bus[7]	1.8-V HSTL Class I	PIN_R24
QDRIIC_Q8	Read Data bus[8]	1.8-V HSTL Class I	PIN_N23
QDRIIC_Q9	Read Data bus[9]	1.8-V HSTL Class I	PIN_J24
QDRIIC_Q10	Read Data bus[10]	1.8-V HSTL Class I	PIN_H24
QDRIIC_Q11	Read Data bus[11]	1.8-V HSTL Class I	PIN_F24
QDRIIC_Q12	Read Data bus[12]	1.8-V HSTL Class I	PIN_J25
QDRIIC_Q13	Read Data bus[13]	1.8-V HSTL Class I	PIN_H23
QDRIIC_Q14	Read Data bus[14]	1.8-V HSTL Class I	PIN_L24
QDRIIC_Q15	Read Data bus[15]	1.8-V HSTL Class I	PIN_F25
QDRIIC_Q16	Read Data bus[16]	1.8-V HSTL Class I	PIN_K24
QDRIIC_Q17	Read Data bus[17]	1.8-V HSTL Class I	PIN_K25
QDRIIC_BWS_n0	Byte Write select[0]	1.8-V HSTL Class I	PIN_L21
QDRIIC_BWS_n1	Byte Write select[1]	1.8-V HSTL Class I	PIN_P21
QDRIIC_K_p	Clock P	Differential 1.8-V HSTL Class I	PIN_N22
QDRIIC_K_n	Clock N	Differential 1.8-V HSTL Class I	PIN_M22
QDRIIC_CQ_p	Echo clock P	1.8-V HSTL Class I	PIN_L23
QDRIIC_CQ_n	Echo clock N	1.8-V HSTL Class I	PIN_K26
QDRIIC_RPS_n	Report Select	1.8-V HSTL Class I	PIN_K21
QDRIIC_WPS_n	Write Port Select	1.8-V HSTL Class I	PIN_T22
QDRIIC_DOFF_n	PLL Turn Off	1.8-V HSTL Class I	PIN_E26
QDRIIC_ODT	On-Die Termination Input	1.8-V HSTL Class I	PIN_G25
QDRIIC_QVLD	Valid Output Indicator	1.8-V HSTL Class I	PIN_H25

Table 2-16 QDRII+ SRAM D Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix IV GX Pin Number*
QDRIID_A0	Address bus[0]	1.8-V HSTL Class I	PIN_U26
QDRIID_A1	Address bus[1]	1.8-V HSTL Class I	PIN_T27

QDRIID_A2	Address bus[2]	1.8-V HSTL Class I	PIN_P26
QDRIID_A3	Address bus[3]	1.8-V HSTL Class I	PIN_V27
QDRIID_A4	Address bus[4]	1.8-V HSTL Class I	PIN_P28
QDRIID_A5	Address bus[5]	1.8-V HSTL Class I	PIN_T26
QDRIID_A6	Address bus[6]	1.8-V HSTL Class I	PIN_M28
QDRIID_A7	Address bus[7]	1.8-V HSTL Class I	PIN_P29
QDRIID_A8	Address bus[8]	1.8-V HSTL Class I	PIN_T25
QDRIID_A9	Address bus[9]	1.8-V HSTL Class I	PIN_N26
QDRIID_A10	Address bus[10]	1.8-V HSTL Class I	PIN_M27
QDRIID_A11	Address bus[11]	1.8-V HSTL Class I	PIN_R27
QDRIID_A12	Address bus[12]	1.8-V HSTL Class I	PIN_U27
QDRIID_A13	Address bus[13]	1.8-V HSTL Class I	PIN_L26
QDRIID_A14	Address bus[14]	1.8-V HSTL Class I	PIN_K27
QDRIID_A15	Address bus[15]	1.8-V HSTL Class I	PIN_L27
QDRIID_A16	Address bus[16]	1.8-V HSTL Class I	PIN_N28
QDRIID_A17	Address bus[17]	1.8-V HSTL Class I	PIN_V26
QDRIID_A18	Address bus[18]	1.8-V HSTL Class I	PIN_P27
QDRIID_A19	Address bus[19]	1.8-V HSTL Class I	PIN_V25
QDRIID_A20	Address bus[20]	1.8-V HSTL Class I	PIN_D26
QDRIID_D0	Write data bus[0]	1.8-V HSTL Class I	PIN_H31
QDRIID_D1	Write data bus[1]	1.8-V HSTL Class I	PIN_H30
QDRIID_D2	Write data bus[2]	1.8-V HSTL Class I	PIN_H32
QDRIID_D3	Write data bus[3]	1.8-V HSTL Class I	PIN_R31
QDRIID_D4	Write data bus[4]	1.8-V HSTL Class I	PIN_T30
QDRIID_D5	Write data bus[5]	1.8-V HSTL Class I	PIN_V31
QDRIID_D6	Write data bus[6]	1.8-V HSTL Class I	PIN_W32
QDRIID_D7	Write data bus[7]	1.8-V HSTL Class I	PIN_Y32
QDRIID_D8	Write data bus[8]	1.8-V HSTL Class I	PIN_W31
QDRIID_D9	Write data bus[9]	1.8-V HSTL Class I	PIN_M30
QDRIID_D10	Write data bus[10]	1.8-V HSTL Class I	PIN_P31
QDRIID_D11	Write data bus[11]	1.8-V HSTL Class I	PIN_N29
QDRIID_D12	Write data bus[12]	1.8-V HSTL Class I	PIN_L30
QDRIID_D13	Write data bus[13]	1.8-V HSTL Class I	PIN_L29
QDRIID_D14	Write data bus[14]	1.8-V HSTL Class I	PIN_J30
QDRIID_D15	Write data bus[15]	1.8-V HSTL Class I	PIN_G32
QDRIID_D16	Write data bus[16]	1.8-V HSTL Class I	PIN_F32
QDRIID_D17	Write data bus[17]	1.8-V HSTL Class I	PIN_T29
QDRIID_Q0	Read Data bus[0]	1.8-V HSTL Class I	PIN_B29
QDRIID_Q1	Read Data bus[1]	1.8-V HSTL Class I	PIN_E30
QDRIID_Q2	Read Data bus[2]	1.8-V HSTL Class I	PIN_R28
QDRIID_Q3	Read Data bus[3]	1.8-V HSTL Class I	PIN_W29
QDRIID_Q4	Read Data bus[4]	1.8-V HSTL Class I	PIN_V29

QDRIID_Q5	Read Data bus[5]	1.8-V HSTL Class I	PIN_Y30
QDRIID_Q6	Read Data bus[6]	1.8-V HSTL Class I	PIN_Y29
QDRIID_Q7	Read Data bus[7]	1.8-V HSTL Class I	PIN_U29
QDRIID_Q8	Read Data bus[8]	1.8-V HSTL Class I	PIN_V30
QDRIID_Q9	Read Data bus[9]	1.8-V HSTL Class I	PIN_C30
QDRIID_Q10	Read Data bus[10]	1.8-V HSTL Class I	PIN_Y27
QDRIID_Q11	Read Data bus[11]	1.8-V HSTL Class I	PIN_D30
QDRIID_Q12	Read Data bus[12]	1.8-V HSTL Class I	PIN_Y28
QDRIID_Q13	Read Data bus[13]	1.8-V HSTL Class I	PIN_W28
QDRIID_Q14	Read Data bus[14]	1.8-V HSTL Class I	PIN_V28
QDRIID_Q15	Read Data bus[15]	1.8-V HSTL Class I	PIN_B31
QDRIID_Q16	Read Data bus[16]	1.8-V HSTL Class I	PIN_A31
QDRIID_Q17	Read Data bus[17]	1.8-V HSTL Class I	PIN_A32
QDRIID_BWS_n0	Byte Write select[0]	1.8-V HSTL Class I	PIN_T31
QDRIID_BWS_n1	Byte Write select[1]	1.8-V HSTL Class I	PIN_R30
QDRIID_K_p	Clock P	Differential 1.8-V HSTL Class I	PIN_K30
QDRIID_K_n	Clock N	Differential 1.8-V HSTL Class I	PIN_K29
QDRIID_CQ_p	Echo clock P	1.8-V HSTL Class I	PIN_T28
QDRIID_CQ_n	Echo clock N	1.8-V HSTL Class I	PIN_C31
QDRIID_RPS_n	Report Select	1.8-V HSTL Class I	PIN_U30
QDRIID_WPS_n	Write Port Select	1.8-V HSTL Class I	PIN_P30
QDRIID_DOFF_n	PLL Turn Off	1.8-V HSTL Class I	PIN_G31
QDRIID_ODT	On-Die Termination Input	1.8-V HSTL Class I	PIN_D32
QDRIID_QVLD	Valid Output Indicator	1.8-V HSTL Class I	PIN_E32

2.10 SPF+

The development board has two independent 10G SFP+ connectors that use one transceiver channel each from the Stratix V GX FPGA device. These modules take in serial data from the Stratix V GX FPGA device and transform them to optical signals. The board includes cage assemblies for the SFP+ connectors. **Figure 2-10** shows the connections between the SFP+ and Stratix V GX FPGA.

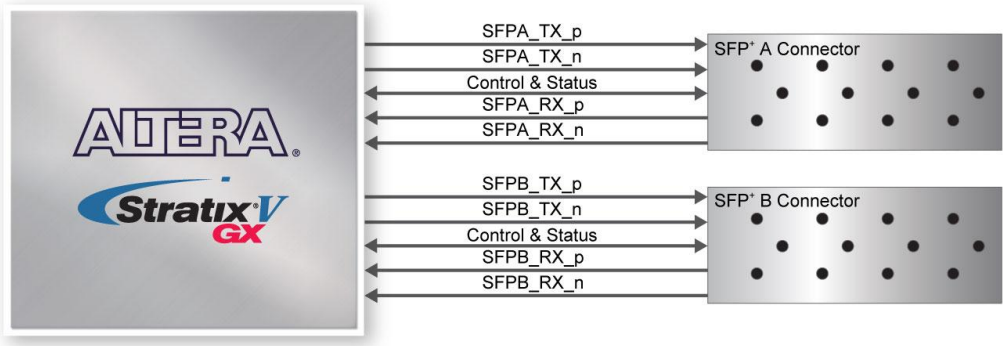


Figure 2-10 Connection between the SFP+ and Stratix V GX FPGA

Table 2-17 and **Table 2-18** list ists the QSF+ A and B pin assignments and signal names relative to the Stratix V GX device.

Table 2-17 SFP+ A Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix V GX Pin Number*
SFPA_TX_p	Transmiter data	1.4-V PCML	PIN_AY6
SFPA_TX_n	Transmiter data	1.4-V PCML	PIN_AY5
SFPA_RX_p	Receiver data	1.4-V PCML	PIN_BB2
SFPA_RX_n	Receiver data	1.4-V PCML	PIN_BB1
SPFA_LOS	Signal loss indicator	2.5V	PIN_AW24
SPFA_MOD0_PRSNT_n	Module present	2.5V	PIN_BA25
SPFA_MOD1_SCL	Serial 2-wire clock	2.5V	PIN_BA24
SPFA_MOD2_SDA	Serial 2-wire data	2.5V	PIN_BB26
SPFA_RATSEL0	Rate select 0	2.5V	PIN_AY25
SPFA_RATSEL1	Rate select 1	2.5V	PIN_AV25

SPFA_TXDISABLE	Turns off and disables the transmitter output	2.5V	PIN_BC26
SPFA_TXFAULT	Transmitter fault	2.5V	PIN_BD26

Table 2-18 SFP+ B Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix V GX Pin Number*
SFPB_TX_p	Transmiter data	1.4-V PCML	PIN_AV6
SFPB_TX_n	Transmiter data	1.4-V PCML	PIN_AV5
SFPB_RX_p	Receiver data	1.4-V PCML	PIN_BA4
SFPB_RX_n	Receiver data	1.4-V PCML	PIN_BA3
SPFB_LOS	Signal loss indicator	2.5V	PIN_AJ23
SPFB_MOD0_PRSNT_n	Module present	2.5V	PIN_AK23
SPFB_MOD1_SCL	Serial 2-wire clock	2.5V	PIN_AR25
SPFB_MOD2_SDA	Serial 2-wire data	2.5V	PIN_AP24
SPFB_RATSEL0	Rate select 0	2.5V	PIN_AJ22
SPFB_RATSEL1	Rate select 1	2.5V	PIN_AH22
SPFB_TXDISABLE	Turns off and disables the transmitter output	2.5V	PIN_AU25
SPFB_TXFAULT	Transmitter fault	2.5V	PIN_AU24

2.11 PCI Express

The TR5-Lite development board is designed to fit entirely into a PC motherboard with x8 or x16 PCI Express slot. Utilizing built-in transceivers on a Stratix V GX device, it is able to provide a fully integrated PCI Express-compliant solution for multi-lane (x1, x4, and x8) applications. With the PCI Express hard IP block incorporated in the Stratix V GX device, it will allow users to implement simple and fast protocol, as well as saving logic resources for logic application. **Figure 2-11** presents the pin connection established between the Stratix V GX and PCI Express.

The PCI Express interface supports complete PCI Express Gen1 at 2.5Gbps/lane, Gen2 at 5.0Gbps/lane, and Gen3 at 8.0Gbps/lane protocol stack solution compliant to PCI Express base specification 3.0 that includes PHY-MAC, Data Link, and transaction layer circuitry embedded in PCI Express hard IP blocks.

The power of the board can be sourced entirely from the PCI Express edge connector when installed into a PC motherboard. It is strongly recommended that users connect the PCIe external power connector to 6-pin 12V DC power connector in the TR5-Lite to avoid FPGA damage due to

insufficient power. The PCIE_REFCLK_p signal is a differential input that is driven from the PC motherboard on this board through the PCIe edge connector. A DIP switch (SW3) is connected to the PCI Express to allow different configurations to enable a x1, x4, or x8 PCIe.

Table 2-19 summarizes the PCI Express pin assignments of the signal names relative to the Stratix V GX FPGA.

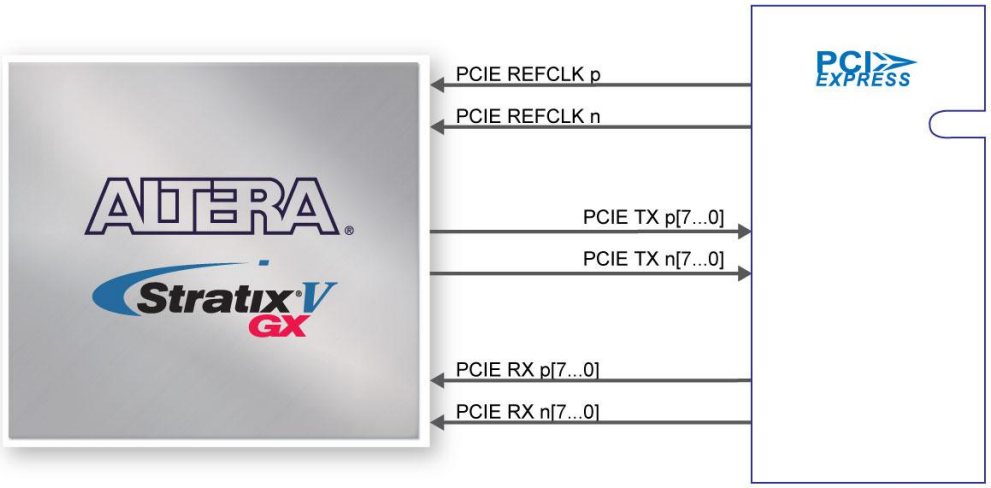


Figure 2-11 PCI Express pin connection

Table 2-19 PCI Express Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix V GX Pin Number*
PCIE_TX_p0	Add-in card transmit bus	1.4-V PCML	PIN_AY39
PCIE_TX_n0	Add-in card transmit bus	1.4-V PCML	PIN_AY40
PCIE_TX_p1	Add-in card transmit bus	1.4-V PCML	PIN_AV39
PCIE_TX_n1	Add-in card transmit bus	1.4-V PCML	PIN_AV40
PCIE_TX_p2	Add-in card transmit bus	1.4-V PCML	PIN_AT39
PCIE_TX_n2	Add-in card transmit bus	1.4-V PCML	PIN_AT40
PCIE_TX_p3	Add-in card transmit bus	1.4-V PCML	PIN_AU41
PCIE_TX_n3	Add-in card transmit bus	1.4-V PCML	PIN_AU42
PCIE_TX_p4	Add-in card transmit bus	1.4-V PCML	PIN_AN41
PCIE_TX_n4	Add-in card transmit bus	1.4-V PCML	PIN_AN42
PCIE_TX_p5	Add-in card transmit bus	1.4-V PCML	PIN_AL41
PCIE_TX_n5	Add-in card transmit bus	1.4-V PCML	PIN_AL42
PCIE_TX_p6	Add-in card transmit bus	1.4-V PCML	PIN_AJ41
PCIE_TX_n6	Add-in card transmit bus	1.4-V PCML	PIN_AJ42
PCIE_TX_p7	Add-in card transmit bus	1.4-V PCML	PIN_AG41

PCIE_TX_n7	Add-in card transmit bus	1.4-V PCML	PIN_AG42
PCIE_RX_p0	Add-in card receive bus	1.4-V PCML	PIN_BB43
PCIE_RX_n0	Add-in card receive bus	1.4-V PCML	PIN_BB44
PCIE_RX_p1	Add-in card receive bus	1.4-V PCML	PIN_BA41
PCIE_RX_n1	Add-in card receive bus	1.4-V PCML	PIN_BA42
PCIE_RX_p2	Add-in card receive bus	1.4-V PCML	PIN_AW41
PCIE_RX_n2	Add-in card receive bus	1.4-V PCML	PIN_AW42
PCIE_RX_p3	Add-in card receive bus	1.4-V PCML	PIN_AY43
PCIE_RX_n3	Add-in card receive bus	1.4-V PCML	PIN_AY44
PCIE_RX_p4	Add-in card receive bus	1.4-V PCML	PIN_AT43
PCIE_RX_n4	Add-in card receive bus	1.4-V PCML	PIN_AT44
PCIE_RX_p5	Add-in card receive bus	1.4-V PCML	PIN_AP43
PCIE_RX_n5	Add-in card receive bus	1.4-V PCML	PIN_AP44
PCIE_RX_p6	Add-in card receive bus	1.4-V PCML	PIN_AM43
PCIE_RX_n6	Add-in card receive bus	1.4-V PCML	PIN_AM44
PCIE_RX_p7	Add-in card receive bus	1.4-V PCML	PIN_AK43
PCIE_RX_n7	Add-in card receive bus	1.4-V PCML	PIN_AK44
PCIE_REFCLK_p	Motherboard reference clock	HCSL	PIN_AK38
PCIE_REFCLK_n	Motherboard reference clock	HCSL	PIN_AK39
PCIE_PERST_n	Reset	2.5-V	PIN_BB39
PCIE_SMBCLK	SMB clock	2.5-V	PIN_AW37
PCIE_SMBDAT	SMB data	2.5-V	PIN_AV37
PCIE_WAKE_n	Wake signal	2.5-V	PIN_AY37
PCIE_PRSNT1n	Hot plug detect	-	-
PCIE_PRSNT2n_x1	Hot plug detect x1 PCIe slot enabled using SW3 dip switch	-	-
PCIE_PRSNT2n_x4	Hot plug detect x4 PCIe slot enabled using SW3 dip switch	-	-
PCIE_PRSNT2n_x8	Hot plug detect x8 PCIe slot enabled using SW3 dip switch	-	-

2.12 SATA

One Serial ATA (SATA) host port is available on the TR5-Lite development board which is computer bus standard with the primary function of transferring data between the motherboard and mass storage devices (such as hard drives, optical drives, and solid-state disks). Supporting a storage interface is just one of many different applications an FPGA can be used in storage appliances. The Stratix V GX device can bridge different protocols such as bridging simple bus I/Os

like PCI Express (PCIe) to SATA. The SATA interface supports SATA 3.0 standard with connection speed of 6 Gbps based on the Stratix V GX device with integrated transceivers compliant to SATA electrical standards.

Table 2-20 lists the SATA pin assignments, signal names and functions.

Table 2-20 Serial ATA Pin Assignments, Schematic Signal Names, and Functions

Schematic Signal Name	*Description*	*I/O Standard*	*Stratix V GX Pin Number*
SATA_HOST_TX_p0	Differential transmit data output before DC blocking capacitor	1.4-V PCML	PIN_AL4
SATA_HOST_TX_n0	Differential transmit data output before DC blocking capacitor	1.4-V PCML	PIN_AL3
SATA_HOST_RX_p0	Differential receive data input after DC blocking capacitor	1.4-V PCML	PIN_AP2
SATA_HOST_RX_n0	Differential receive data input after DC blocking capacitor	1.4-V PCML	PIN_AP1
SATA_REFCLK_ p	Reference Clock	HCSL	PIN_AF7
SATA_REFCLK_ n	Reference Clock	HCSL	PIN_AF6

Chapter 3

System Builder

This chapter describes how users can create a custom design project on the TR5-Lite board by using the Software Tools – TR5-Lite System Builder.

3.1 Introduction

The TR5-Lite System Builder is a Windows based software utility, designed to assist users to create a Quartus II project for the TR5-Lite board within minutes. The generated Quartus II project files include:

- Quartus II Project File (.qpf)
- Quartus II Setting File (.qsf)
- Top-Level Design File (.v)
- External PLL Contorller (.v)
- Synopsis Design Constraints file (.sdc)
- Pin Assignment Document (.htm)

The TR5-Lite System Builder not only can generate the files above, but can also provide error-checking rules to handle situation that are prone to errors. The common mistakes that users encounter are the following:

- Board damaged for wrong pin/bank voltage assignment.
- Board malfunction caused by wrong device connections or missing pin counts for connected ends.
- Performance dropped because of improper pin assignments

3.2 General Design Flow

This section will introduce the general design flow to build a project for the TR5-Lite board via the TR5-Lite System Builder. The general design flow is illustrated in the **Figure 3-1**.

Users should launch TR5-Lite System Builder and create a new project according to their design requirements. When users complete the settings, the TR5-Lite System Builder will generate two major files which include top-level design file (.v) and the Quartus II setting file (.qsf).

The top-level design file contains top-level verilog wrapper for users to add their own design/logic. The Quartus II setting file contains information such as FPGA device type, top-level pin assignment, and I/O standard for each user-defined I/O pin.

Finally, Quartus II programmer must be used to download SOF file to TR5-Lite board using JTAG interface.

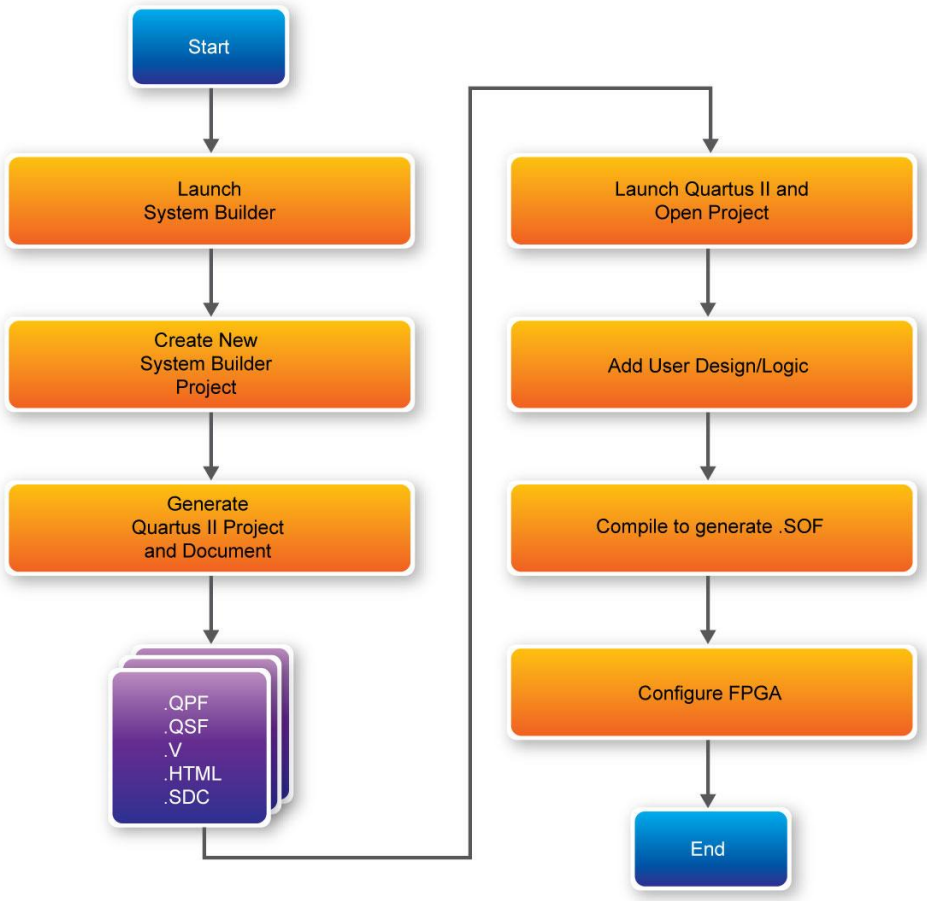


Figure 3-1 The general design flow of building a design

3.3 Using TR5-Lite System Builder

This section provides the detail procedures on how the TR5-Lite System Builder is used.

■ Install and launch the TR5-Lite System Builder

The TR5-Lite System Builder is located in the directory: "**Tools\SystemBuilder**" in the TR5-Lite System CD. Users can copy the whole folder to a host computer without installing the utility. Before using the TR5-Lite System Builder, execute the **TR5_Lite_SystemBuilder.exe** on the host computer as appears in **Figure 3-2**.

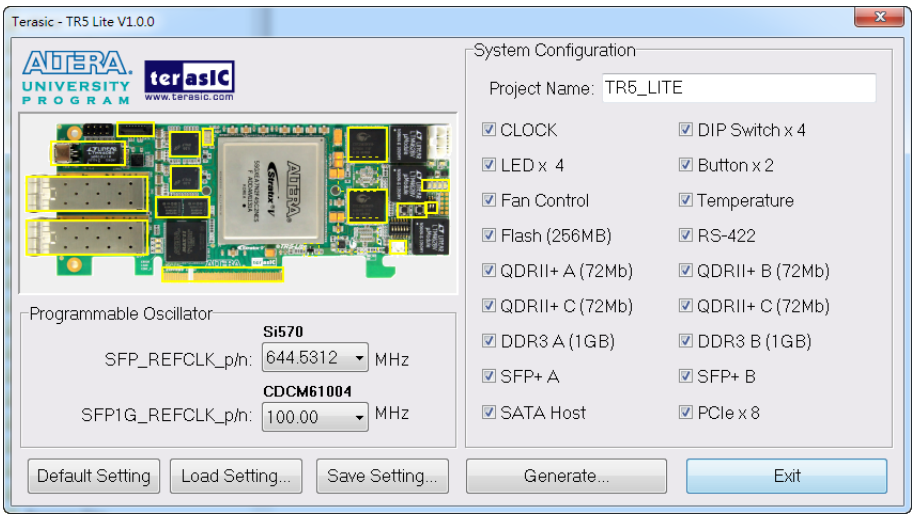


Figure 3-2 The TR5-Lite System Builder window

■ Select Board Type and Input Project Name

Select the target board type and input project name as show in **Figure 3-3**.

- Project Name: Specify the project name as it is automatically assigned to the name of the top-level design entity.

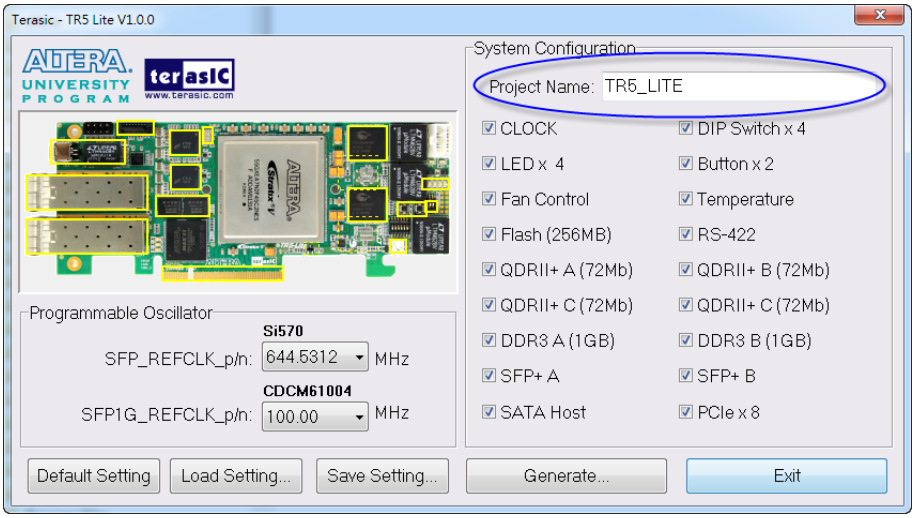


Figure 3-3 The TR5-Lite Project Name

■ System Configuration

Under System Configuration users are given the flexibility of enabling their choice of components on the TR5-Lite as shown in **Figure 3-4**. Each component of the TR5-Lite is listed where users can

enable or disable a component according to their design by simply marking a check or removing the check in the field provided. If the component is enabled, the TR5-Lite System Builder will automatically generate the associated pin assignments including the pin name, pin location, pin direction, and I/O standards.

Note. The pin assignments for some components for e.g. DDR3 and SFP+ require associated controller codes in the Quartus project otherwise Quartus will result in compilation errors. Therefore, do not select them if they are not necessary in your design. To use the DDR3 controller, please refer to the DDR3 SDRAM demonstration in Chapter 6.

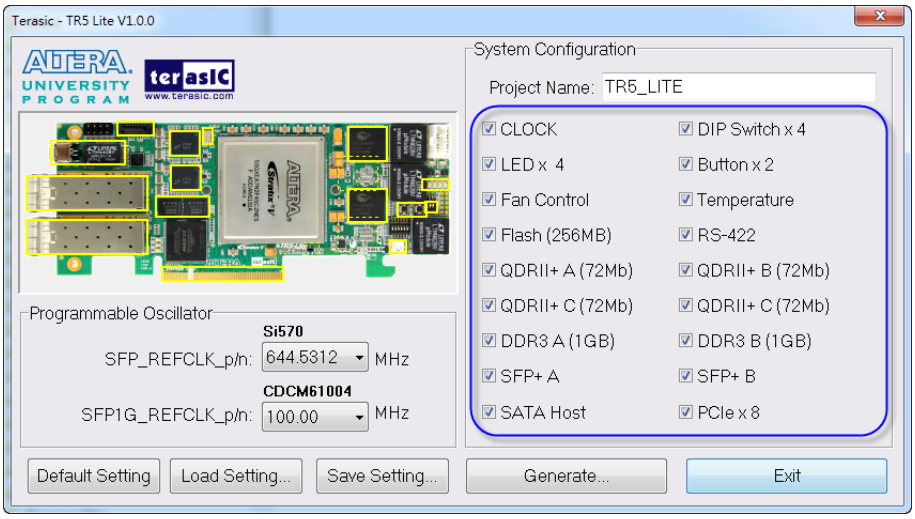


Figure 3-4 System Configuration Group

■ Programmable Oscillator

There are two external oscillators on-board that provide reference clocks for the following signals SFP_REFCLK, SFP1G_REFCLK and SATA_REFCLK. To use these oscillators, users can select the desired frequency on the Programmable Oscillator group, as show in **Figure 3-5**. SPF+ or SATA should be checked before users can start to specify the desired frequency in the Programmable Oscillator group.

As the Quartus project is created, System Builder automatically generates the associated controller according to users' desired frequency in verilog which facilitates users' implementation as no additional control code is required to configure the programmable oscillator.

Note. If users need to dynamically change the frequency, they would need to modify the generated

control code themselves.

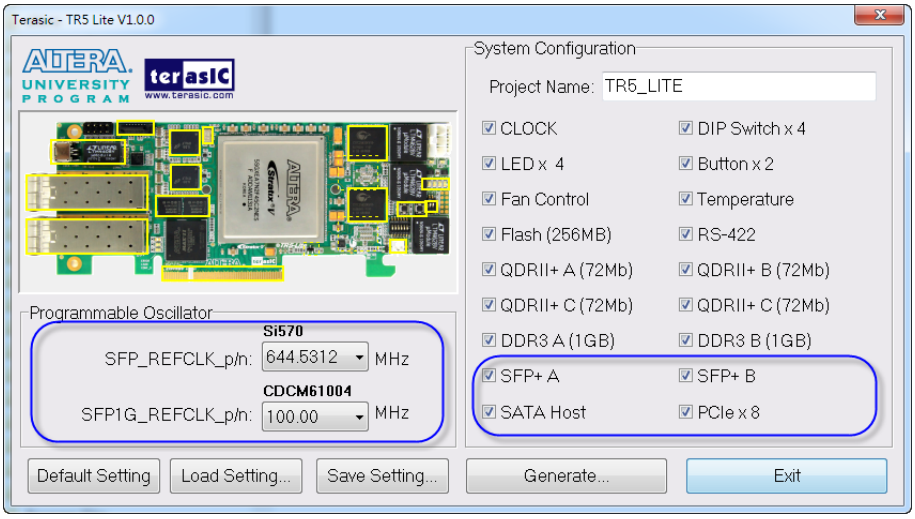


Figure 3-5 External Programmable Oscillators

■ Project Setting Management

The TR5-Lite System Builder also provides functions to restore default setting, loading a setting, and saving users' board configuration file shown in **Figure 3-6**. Users can save the current board configuration information into a .cfg file and load it to the TR5-Lite System Builder.

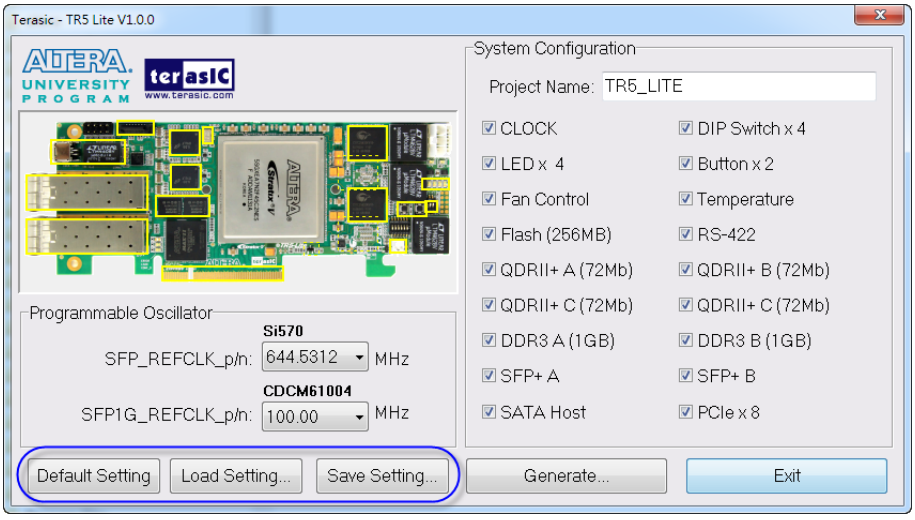


Figure 3-6 Project Settings

■ Project Generation

When users press the **Generate** button, the TR5-Lite System Builder will generate the corresponding Quartus II files and documents as listed in the **Table 3-1** in the directory specified by the user.

Table 3-1 The files generated by TR5-Lite System Builder

No.	Filename	Description
1	<Project name>.v	Top level verilog file for Quartus II
2	si570_controller.v(*)	SI570 External Oscillator controller IP
3	<Project name>.qpf	Quartus II Project File
4	<Project name>.qsf	Quartus II Setting File
5	<Project name>.sdc	Synopsis Design Constraints file for Quartus II
6	<Project name>.htm	Pin Assignment Document

(*) The SI570 Controller includes seven files: si570_controller.v, initial_config.v, clock_divider.v, edge_detector.v, i2c_reg_controller.v, i2c_controller.v and i2c_bus_controller.v.

Users can use Quartus II software to add custom logic into the project and compile the project to generate the SRAM Object File (.sof).

For SI570, the Controller will be instantiated in the Quartus II top-level file as listed below:

```
//=======================================================
//  Configure SI570 as 644.5312 MHz ====================
//=======================================================

si570_controller si570_controller_inst(
    .iCLK(OSC_50_B3B), // system   clock 50mhz
    .iRST_n(BUTTON[0]), // system reset;
    .iFREQ_MODE(3'b110),
    .I2C_CLK(CLOCK_SCL),
    .I2C_DATA(CLOCK_SDA),
    .oController_Ready()
);
```

For CDCM61004, the Controller will be instantiated in the Quartus II top-level file as listed below:

```
//=======================================================
//  Configure CDCM61004 as 100.00 MHz ====================
//=======================================================

assign CLK_CE = 1'b1;
assign CLK_OD = 3'b101;
assign CLK_PR = 2'b00;
assign CLK_OS = 2'b01; // LVDS
assign CLK_RST_n = BUTTON[0];
```

If dynamic configuration for the oscillator is required, users need to modify the code according to users' desired behavior.

Chapter 4

Flash Programming

As you develop your own project using the Altera tools, you can program the flash memory device so that your own design loads from flash memory into the FPGA on power up. This chapter will describe how to use Altera Quartus II Programmer Tool to program the common flash interface (CFI) flash memory device on the TR5-Lite. The Stratix V GX FPGA development board ships with the CFI flash device preprogrammed with a default factory FPGA configuration for running the Parallel Flash Loader design example.

4.1 CFI Flash Memory Map

Table 4-1 shows the default memory contents of two interlaced 1Gb (128MB) CFI flash device. Each flash device has a 16-bit data bus and the two combined flash devices allow for a 32-bit flash memory interface. For the factory default code to run correctly and update designs in the user memory, this memory map must not be altered.

Table 4-1 Flash Memory Map (Byte Address)

Block Description	*Size(KB)*	*Address Range*
PFL option bits	**64**	**0x00030000 – 0x0003FFFF**
Factory hardware	**33,280**	**0x00040000 – 0x020BFFFF**
User hardware	**33,280**	**0x020C0000 – 0x0413FFFF**
Factory software	**8,192**	**0x04140000 – 0x0493FFFF**
User software and data	**187,136**	**0x04940000 – 0x0FFFFFFF**

For user application, user hardware must be stored with start address **0x020C0000**, and the user's software is suggested to be stored with start address **0x04940000**. The NIOS II EDS tool **nios-2-flash-programmer** is used for programming the flash. Before programming, users need to

translate their Quartus .sof and NIOS II .elf files into the .flash which is used by the **nios-2-flash-programmer**. For .sof to .flash translation, NIOS II EDS tool **sof2flsh** can be used. For the .elf to .flash translation, NIOS II EDS tool **elf2flash** can be used. For convenience, the System CD contains a batch file for file translation and flash programming with users given .sof and .elf file.

4.2 FPGA Configure Operation

Here is the procedure to enable FPGA configuration from Flash:

1. Please make sure the FPGA configuration data has been stored in the CFI flash.
2. Set the FPGA configuration mode to FPPx32 mode by setting SW5 MSEL[0:4] as 00010
3. Specify the configuration of the FPGA using the default Factory Configuration or User Configuration by setting SW1 according to Table
4. Power on the TR5-Lite board or press MAX_RST button if board is already powered on
5. When configuration is completed, the green Configure Done LED will light. If there is error, the red Configure Error LED will light.

4.3 Flash Programming with Users Design

Users can program the flash memory device so that users' own design loads from flash memory into the FPGA on power up. For convenience, the translation and programming batch files are available on the Demonstrations/TR5_Lite_Hello/flash_programming_batch folder in the System CD. There folder contains five files as shown in **Table 4-2**

Table 4-2 Flash Memory Map (Byte Address)

Files Name	*Description*
TR5_LITE_PFL.sof	**Parallel Flash Loader Design**
flash_program_ub2.bat	**Top batch file to download TR5_LITE_PFL.sof and launch batch flash_program_bashrc_ub2**
flash_program_bashrc_ub2	**Translate .sof and .elf into .flash and programming flash with the generated .flash file**
TR5_LITE_golden_top.sof	**Hardware design file for Hello Demo**

HELLO_NIOS.elf	**Software design file for Hello Demo**

To apply the batch file to users' .sof and .elf file, users can change the .sof and .elf filename in the **flash_program_bashrc_ub2** file as shown in **Figure 4-1**.

```
#conver to .flash
sof2flash --input=TR5_LITE_golden_top.sof --output=flash_hw.flash --offset=0x20C0000 --pfl
elf2flash --base=0x0 --end=0x0FFFFFFF --reset=0x04940000 --input=HELLO_NIOS.elf --output=fl
```

Figure 4-1 Change to usrs' .sof and .elf filename

If your design does not contain a NIOS II processor, users can add "#" to comment (disable) the elf2flash and nios-flash-programmer commands in the **flash_program_bashrc_ub2** file as shown in **Figure 4-2**.

```
#conver to .flash
sof2flash --input=TR5_LITE_golden_top.sof --output=flash_hw.flash --offset=0x20C0000
elf2flash --base=0x0 --end=0x0FFFFFFF --reset=0x04940000 --input=HELLO_NIOS.elf --out


#Programming with .flash
nios2-flash-programmer --base=0x0 flash_hw.flash
nios2-flash-programmer --base=0x0 flash_sw.flash
```

Figure 4-2 Disable .elf translation and programming

If your design includes a NIOS II processor and the NIOS II program is stored on external memory, users must to perform following items so the NIOS II program can be boot from flash successfully:

1. QSYS should include a Flash controller for the CFI Flash on the development board. Please ensure that the base address of the controller is 0x00, as shown in **Figure 4-3**.
2. In NIOS II processor options, select FLASH as reset vector memory and specify 0x04940000 as reset vector, as shown in **Figure 4-4**.

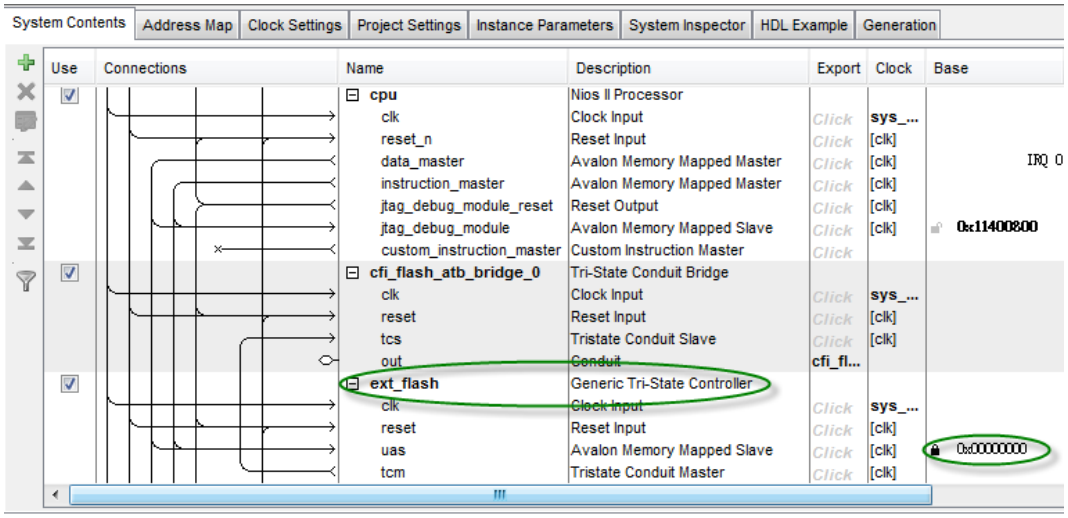


Figure 4-3 Flash Controller Settings in QSYS

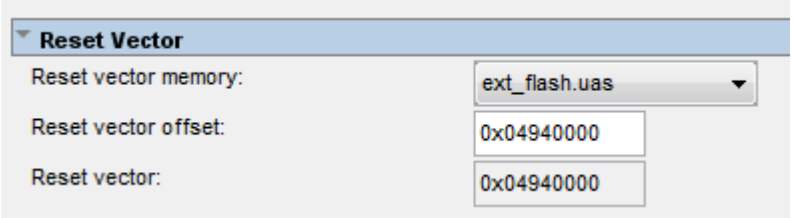


Figure 4-4 Reset Vector Settings for NIOS II Processor

For implementation detail, users can refer the Hello example located in the CD folder:

Demonstrations/TR5_Lite_Hello

4.4 Restore Factory Settings

This section describes how to restore the original factory contents to the flash memory device on the FPGA development board. Perform the following instructions:

1. Make sure the Nios II EDS and USB-Blaster II driver are installed.

2. Make sure the TR5-Lite board and PC are connected with USB-Blaster II.
3. Power on the TR5-Lite.
4. Copy the "Demonstrations/TR5_Lite_PFL/flash_programming_batch" folder under the CD to your PC's local drive.
5. Execute the batch file flash_program_ub2.bat to start flash programming.
6. Power off the TR5-Lite.
7. Set FPGA configure mode as FPPx32 Mode by setting SW5 MSEL[0:4] to 00010.
8. Specify configuration of the FPGA to Factory Hardware by setting SW1 to low.
9. Power on the TR5-Lite, and the Configure Done LED should light.

Except for programming the Flash with the default code TR5_Lite_PFL, the batch file also writes PFL (Parallel Flash Loader) Option Bits data into the address 0x30000. The option bits data specifies 0x20C0000 as start address of your hardware design.

The NIOS II EDS tool **nios-2-flash-programmer** programs the Flash based on the Parallel Flasher Loader design in the FPGA. The Parallel Flash Loader design is included in the default code TR5_Lite_PFL and the source code is available in the folder Demonstrations/TR5_Lite_PFL in System CD.

Chapter 5

Programmable Oscillator

This chapter describes how to program the two programmable oscillators Si570 and CDCM61004 on the TR4 board. Also, RTL code based and Nios based reference design are explained in the chapter. The source codes of these examples are all available on the TR5-Lite System CD.

5.1 Overview

This section describes how to program Si570- and CDCM61004. For detail programming information, please refer to their datasheets which are available on the TR5-Lite System CD.

■ Si570

The Si570 utilizes Silicon Laboratories advanced DSPLL® circuitry to provide a low-jitter clock at any frequency. The Si570 are user-programmable to any output frequency from 10 to 945 MHz and select frequencies to 1400 MHz with < 1ppb resolution. The device is programmed via an I2C serial interface. The differential clock output of the Si570 directly connects to dedicated reference clock input of the Stratix V GX transceiver for SFP+ channels. Many applications can be implemented using this function. For example, the 10G Ethernet application can be designed onto this board by feeding a necessary clock frequency of 644.53125MHz or 322.265625MHz from the Si570.

Figure 5-1 shows the block diagram of SI570 device. Users can modify the value of the three registers RFREQ, HS_DIV, and N1 to generate the desired output frequency.

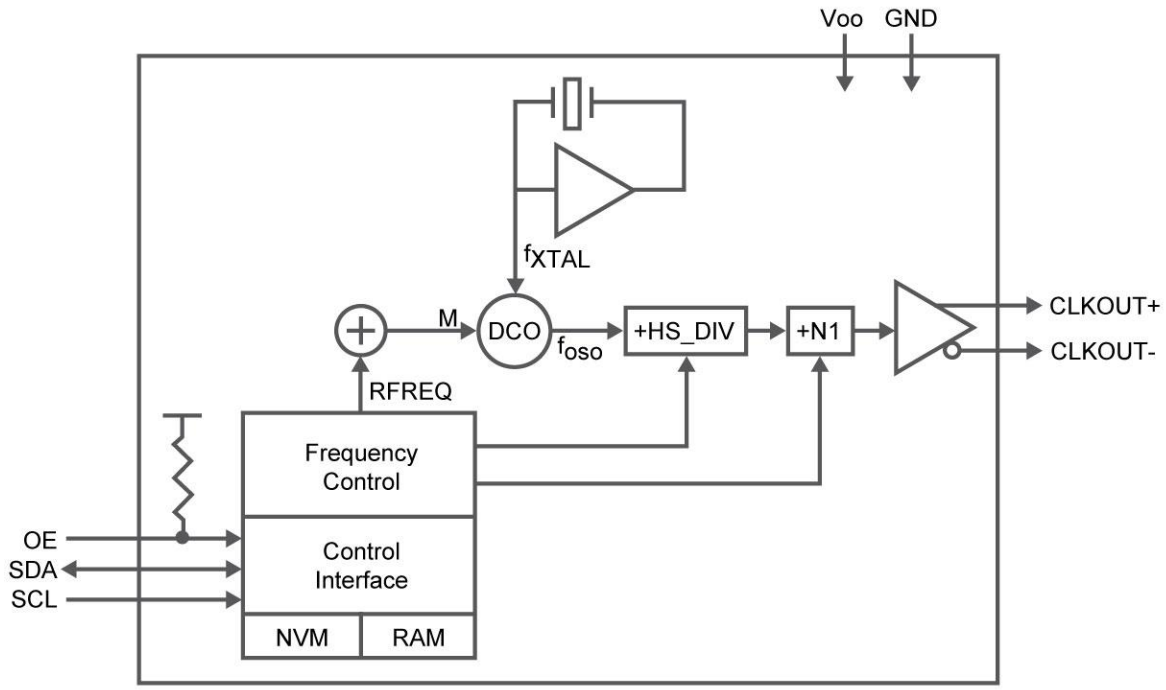


Figure 5-1 Si570 Block diagram

The output frequency is calculated using the following equation:

$$f_{out} = \frac{f_{DCO}}{\text{Output Dividers}} = \frac{f_{XTAL} \times RFREQ}{HSDIV \times N1}$$

When Si570 is powered on, the default output frequency is 100 MHz. Users can program the output frequency through the I2C interface using the following procedure.

10. Freeze the DCO (bit 4 of Register 137).
11. Write the new frequency configuration (RFREQ,HSDIV, and N1) to Register 7 – 12.
12. Unfreeze the DCO and assert the NewFreq bit (bit 6 of Register 135).

The I2C address of Si570 is zero and it supports fast mode operation whose transfer rate is up to 400 kbps. **Table 5-1** shows the register table for Si570.

Table 5-1 Si570 Register Table

Register	*Name*	*Bit 7*	*Bit 6*	*Bit 5*	*Bit 4*	*Bit 3*	*Bit 2*	*Bit 1*	*Bit 0*
7	High Speed/N1 Dividers	HS_DIV[2:0]			N1[6:2]				
8	Reference Frequrncy	N1[1:0]		RFREQ[37:32]					
9	Reference Frequrncy	RFREQ[31:24]							
10	Reference Frequrncy	RFREQ[23:16]							
11	Reference Frequrncy	RFREQ[15:8]							
12	Reference Frequrncy	RFREQ[7:0]							
135	Reference Frequrncy	RST_REG	NewFreq	Freeze M	Freeze VCADC				RECALL
137	Reference Frequrncy				Freeze DCO				

Table 5-2 lists the register settings for some common used frequency.

Table 5-2 Si570 Register Table

Output Frequency (MHz)	*HS_DIV*	*HS_DIV Register Setting*	*NI*	*NI Register Setting*	*REF_CLK Register Setting*
100	9	101	6	0000101	02F40135A9(hex)
125	11	111	4	0000011	0302013B65(hex)
156.25	9	101	4	0000011	0313814290(hex)
250	11	111	2	0000001	0302013B65(hex)
312.5	9	101	2	0000001	0313814290(hex)
322.265625	4	000	4	0000011	02D1E127AF(hex)
644.53125	4	000	2	0000001	02D1E127AF(hex)

■ CDCM61004

The TR5-Lite includes another programmable PLL CDCM61004. The CDCM61004 supports output frequency range from 43.75 MHz to 683.264 MHz. It provides a parallel interface for selecting a desired output frequency. The Stratix V GX FPGA's IOs connect to the interface directly. The differential clock outputs of the CDCM61004 are designed for SFP+ and SATA applications on TR5-Lite board.

When CDCM61004 is powered on, the default output frequency is 100 MHZ. Users can change the output frequency by the following control pins:

1. PR0 and PR1
2. OD0, OD1, and OD2
3. RSTN
4. CE
5. OS0 and OS1

The following table lists the frequency which CDCM61004 can generate in TR5-Lite.

PRESCALLR DIVIDER	*FEEDBACK DIVIDER*	*OUTPUT DEVIDER*	*OUTPUT FREQUENCY(MHz)*	*APPLICATION*
4	**20**	**8**	**62.5**	**GigE**
3	**24**	**8**	**75**	**SATA**
3	**24**	**6**	**100**	**PCI Express**
4	**20**	**4**	**125**	**GigE**
3	**24**	**4**	**150**	**SATA**
3	**25**	**4**	**156.25**	**10 GigE**
5	**15**	**2**	**187.5**	**12 GigE**
3	**24**	**3**	**200**	**PCI Express**
4	**20**	**2**	**250**	**GigE**
4	**20**	**2**	**312.5**	**XGMII**
3	**25**	**1**	**625**	**10 GigE**

The both values of PRESCALER DIVIDER and FEEDBACK DIVIDER can be specified by the PR0 and PR1 control pins according to the following table:

CONTROL INPUTS		PRESCALER DIVIDER	FEEDBACK DIVIDER
PR1	PR0		
0	0	3	24
0	1	5	15
1	0	3	25
1	1	4	20

The value of OUTPUT DIVIDER can be specified by the OD0, OD1 and OD2 control pins according to the following table:

CONTROL INPUTS			OUTPUT DIVIDER
OD2	OD1	OD0	
0	0	0	1
0	0	1	2
0	1	0	3
0	1	1	4
1	0	0	Reserved
1	0	1	6
1	1	0	Reserved
1	1	1	8

After specifying the desired output frequency in the parallel interface, developers must assert the output enable pin CE and control the RSTN pin to generate a rising signal to start the PLL Recalibration process. In the TR5-Lite, the required output type is LVDS, so always set OS0 and SO1 to 0 and 1, respectively.

5.2 Si570 Example by RTL

In this section we will demonstrate how to use the Terasic SI570 Controller implemented in Verilog to control the SI570 programmable oscillator on the TR5-Lite board. This controller IP can configure the SI570 to output a clock with a specific frequency via I2C interface. For demonstration, the output clock is used to implement a counter where the MSB is used to drive an LED, so the user can get the result from the frequency of the LED blinking. We will also introduce the port declarations and associated parameter settings of this IP. **Figure 5-2** shows the block diagram of this demonstration.

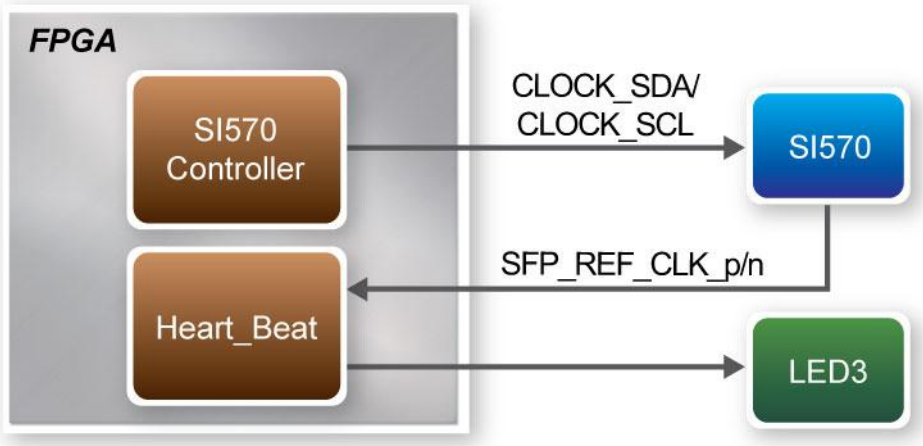


Figure 5-2 Block Diagram of this Demonstration

■ Block Diagrams of SI570 Controller IP

The block diagram of the SI570 controller is shown on **Figure 5-3.** Shown here are four blocks named i2c_reg_controller, i2c_bus_controller, clock_divider and initial_config in SI570 controller IP. Firstly, the i2c_reg_controller will generate an associated SI570 register value for the i2c_bus_controller based on user-desired frequency. Once i2c_bus_controller receives this data, it will transfer these settings to SI570 via serial clock and data bus using I2C protocol. The registers in SI570 will be configured and output the user-desired frequency.

Secondly, the clock_divider block will divide system clock (50Mhz) into 97.6Khz which is used as I2C interface clock of i2c_bus_controller. Finally, the initial_config block will generate a control signal to drive i2c_reg_controller which allows the SI570 controller to configure SI570 based on default settings.

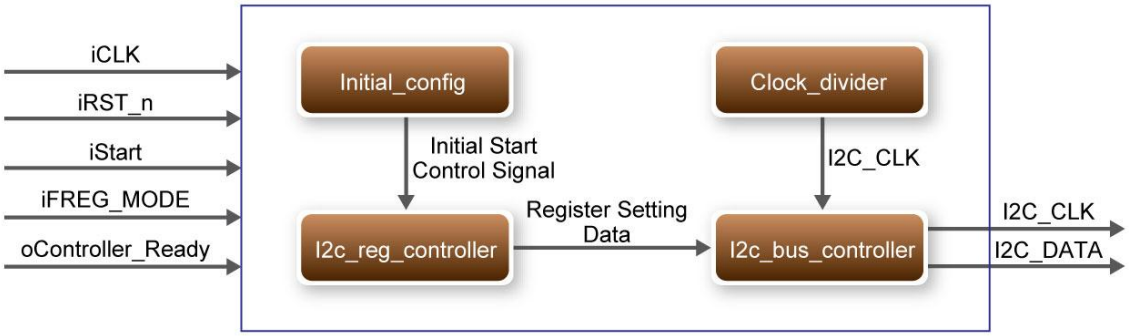


Figure 5-3 Block Diagram of SI570 Controller IP

■ Using SI570 Controller IP

Table 5-3 lists the instruction ports of SI570 Controller IP

Table 5-3 SI570 Controller Instruction Ports

Port	*Direction*	*Description*
iCLK	input	System Clock (50Mhz)
iRST_n	input	Synchronous Reset (0: Module Reset, 1: Normal)
iStart	input	Start to Configure（positive edge trigger）
iFREQ_MODE	input	Setting SI570 Output Frequency Value
oController_Ready	output	SI570 Configuration status (0: Configuration in Progress, 1: Configuration Complete)
I2C_DATA	inout	I2C Serial Data to/from SI570
I2C_CLK	output	I2C Serial Clock to SI570

To use SI570 Controller, the first thing users need to determine is the desired output frequency in advance. The Si570 controller provides six optional clock frequencies. These options can be set through an input port named “iFREQ_MODE” in SI570 controller. The specified settings with corresponding frequencies are listed in Table 5-4. For example, setting “iFREQ_MODE” as 3’b110 will configure SI570 to output 655.53 MHz clock.

Table 5-4 SI570 Controller Frequency Setting

iFREQ MODE Setting	*SI570 Clock Frequency(MHz)*
3'b000	100
3'b001	125

3'b010	156.25
3'b011	250
3'b100	312.25
3'b101	322.26
3'b110	644.53125
3'b111	100

When the output clock frequency is decided, the next thing users need to do is to enable the controller to configure SI570. Before sending enable signal to SI570 controller, users need to monitor an output port named “oController_Ready”. This port indicates if SI570 controller is ready to be configured or not. If it is ready, logic high will be outputted and the user needs to send a high level logic to “iStart” port to enable the SI570 Controller as shown in **Figure 5-4**. During SI570 configuring, the logic level of “oController_Ready” is low; when it rises to high again that means the user can configure another frequency value.

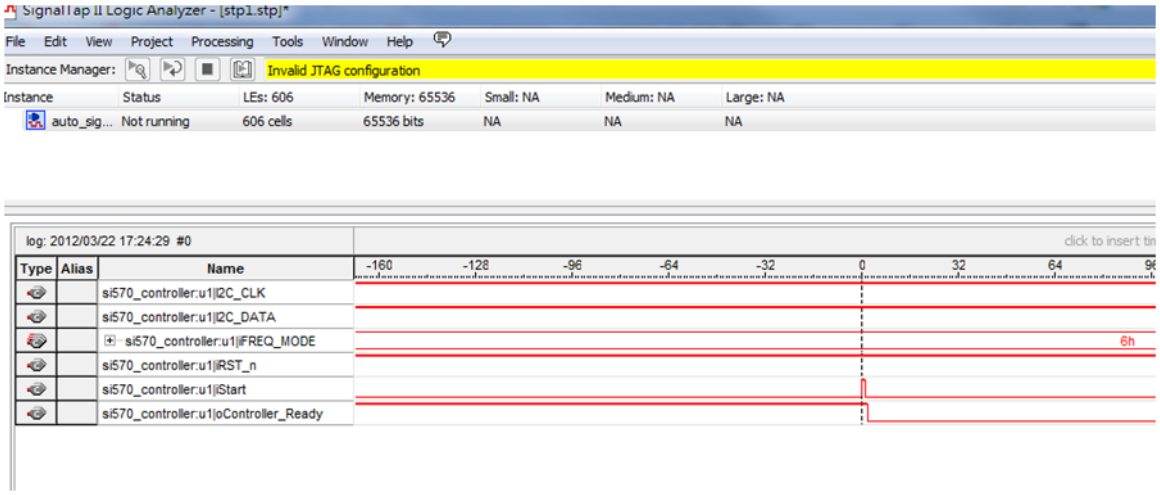


Figure 5-4 Timing Waveform of SI570 Controller

■ Modify Clock Parameter For Your Own Frequency

If all the six clock frequencies are not user desired, user can perform the following steps to modify SI570 controller.

1. Open i2c_reg_controller.v

2. Locate the Verilog code shown below:

```
always @(*)
   begin
      case(iFREQ_MODE)
         3'h0 :     //100Mhz
            begin
               new_hs_div = 4'b0101 ;
               new_n1 = 8'b0000_1010 ;
               fdco = 28'h004_E200 ;
            end
         3'h1 :     //125Mhz
            begin
               new_hs_div = 4'b0101 ;
               new_n1 = 8'b0000_1000 ;
               fdco = 28'h004_E200 ;
            end
         3'h2 :     //156.25Mhz
            begin
               new_hs_div = 4'b0100 ;
               new_n1 = 8'b0000_1000 ;
               fdco = 28'h004_E200 ;
            end
         3'h3 :     //250Mhz
            begin
               new_hs_div = 4'b0101 ;
               new_n1 = 8'b0000_0100 ;
               fdco = 28'h004_E200 ;
            end
         3'h4 :     //312.5Mhz
            begin
               new_hs_div = 4'b0100 ;
               new_n1 = 8'b0000_0100 ;
               fdco = 28'h004_E200 ;
            end
         3'h5 :     //322.265625Mhz
            begin
```

```
                new_hs_div = 4'b0100 ;
                new_n1 = 8'b0000_0100 ;
                fdco = 28'h005_0910 ;
            end
        3'h6 :      //644.53125Mhz
            begin
                new_hs_div = 4'b0100 ;
                new_n1 = 8'b0000_0010 ;
                fdco = 28'h005_0910 ;
            end
        default :      //100Mhz
            begin
                new_hs_div = 4'b0101 ;
                new_n1 = 8'b0000_1010 ;
                fdco = 28'h004_E200 ;
            end
    endcase
end
```

Users can get a desired frequency output from si570 by modifying these three parameters : **new_hs_div** ,**new_n1** and **fdco**.

Detailed calculation method is in following equation:

fdco = output frequency * new_hs_div * new_n1 * 64

There are three constraints for the equation:

1. ***4850 < output fequency * new_hs_div * new_n1 < 5600***
2. ***4 <= new_hs_div <= 11***
3. ***1 <= new_n1 < =128***

For example, you want to get a 133.5 mhz clock, then

fdco = 133.5 x 4 x 10 x 64 = 341760d = 0x53700

Find a mode in this RTL code section and modify these three parameters,as shown below:

```
new_hs_div = 3'b100 ;
new_n1 = 4'b1010 ;
fdco = 23'h05_3700 ;
```

In addition, Silicon Lab also provide the corresponding calculation tool.

Users can refer to the Programmable Oscillator tool (See **Figure 5-5)** mentioned in below link to calculate the values of new_hs_div and new_n1, then, the fdco value can be calcuted with above ftdo equation.

http://www.silabs.com/products/clocksoscillators/oscillators/Pages/oscillator-software-development-tools.aspx

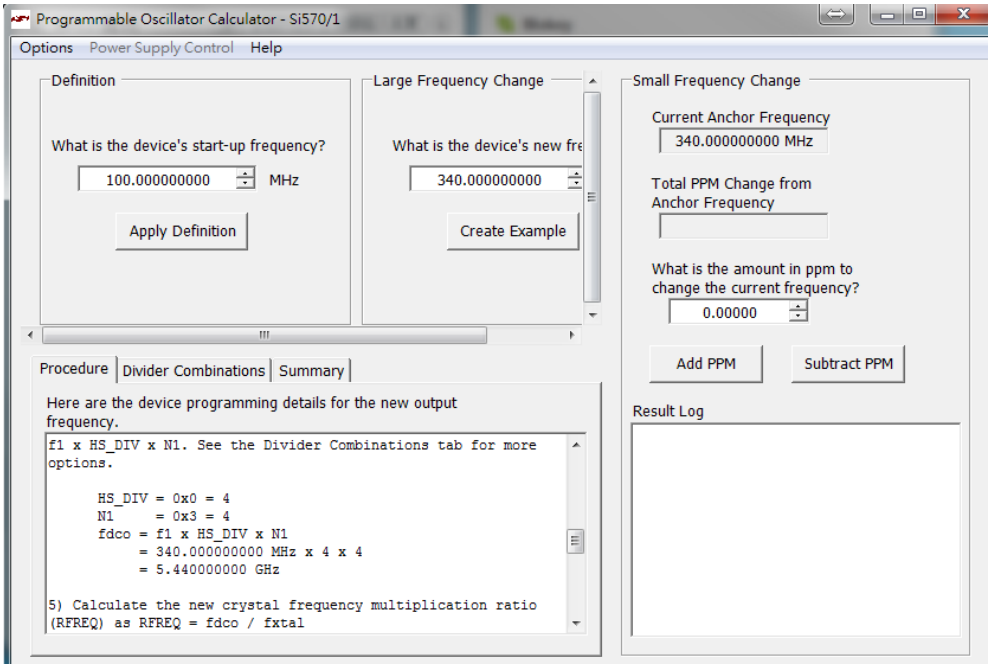


Figure 5-5 Programmable Oscillator Calculator tool

In addition, if the user doesn't want Si570 controller to configure Si570 as soon as the FPGA configuration finishes, users can change settings in Si570_controller.v, shown below.

```
initial_config initial_config(

.iCLK(iCLK), // system      clock 50mhz
.iRST_n(iRST_n), // system reset
.oINITIAL_START(initial_start),
.iINITIAL_ENABLE(1'b1),
);
```

Changing the setting from ".iINITIAL_ENABLE(1'b1) " to ".iINITIAL_ENABLE(1'b0)" will disable the initialization function of Si570 Controller.

■ Design Tools

- Quartus II 13.1

■ Demonstration Source Code

- Project directory: TR5_LITE_SI570_Controller
- Bit stream used: TR5_LITE_SI570_Controller.sof
- Demonstration Batch File : test.bat (For USB-Blaster)/ test_ub2.bat (For USB-Blaster II)
- Demo Batch File Folder: TR5_LITE_SI570_Controller \demo_batch

The demo batch file folders include the following files:

- Batch File: test.bat (For USB-Blaster)/ test_ub2.bat (For USB-Blaster II)

- FPGA Configuration File: TR5_LITE_SI570_CONTROLLER.sof

■ Demonstration Setup

- Make sure Quartus II is installed on your PC.
- Connect the USB Blaster cable to the TR5-Lite board and host PC. Install the USB Blaster driver if necessary.

- Power on the TR5-Lite board.
- Execute the demo batch file “test.bat (or test_ub2.bat)” under the batch file folder, TR5_LITE_SI570_CONTROLLER\demo_batch
- Press **BUTTON1** to configure the SI570.
- Observe LED3 status.

5.3 Si570 and CDCM Programming by Nios II

This demonstration shows how to use the Nios II processor to program both programmable oscillators Si570 and CDCM on TR5-Lite Board. The demonstration also includes a function to monitor system temperature with the on-board temperature sensor.

■ System Block Diagram

Figure 5-6 shows the system block diagram of this demonstration. The system requires a 50 MHz clock provided from the board. The three peripheral temperature sensor, Si570, and CDCM61004 are all controlled by Nios II through the PIO controller. The temperature sensor and external PLL Si570 are controlled through I2C interface. The Nios II program toggles the PIO controller to implement the I2C protocol. The CDCM 61004 is programmed through the PIO directly. The Nios II program is running in the on-chip memory.

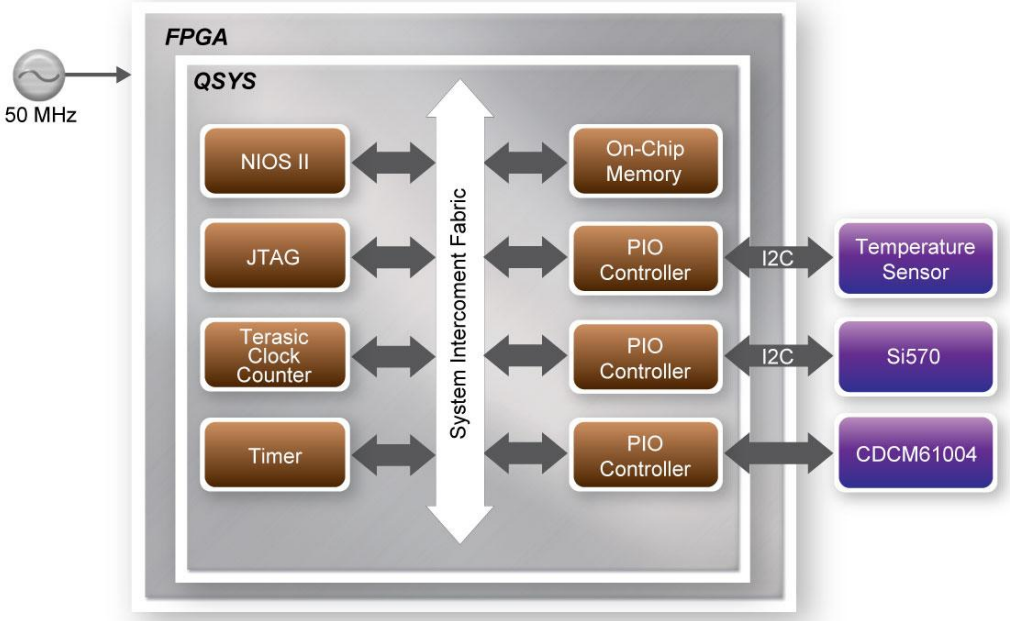


Figure 5-6 Block diagram of the Nios II Basic Demonstration

The program provides a menu in nios-terminal, as shown in **Figure 5-7** to provide an interactive interface. With the menu, users can perform the test for the temperatures sensor and external PLL. Note, pressing 'ENTER' should be followed with the choice number.

```
======= TR5-Lite Demo Program =======
[0] Temperature
[1] CDCM61004
[2] Si570
Input your chioce:
```

Figure 5-7 Menu of Demo Program

In temperature test, the program will display local temperature and remote temperature. The remote temperature is the FPGA temperature, and the local temperature the board temperature where the temperature sensor located.

In the external PLL programming test, the program will program the PLL first, then use TERASIC QSYS custom CLOCK_COUNTER IP to count the clock count in a specified period to check whether the output frequency is changed as demanded. To avoid a Quartus II compilation error, dummy transceiver controllers are created to receive the clock from the external PLL. Users can

ignore the functionality of the transceiver controller in the demonstration.

For CDMC61004 programming, users must trigger the CLK_RST_n to notify the chip to perform PLL recalibration. For Si570 programming, please note the device I2C address is 0x00. Also, before configure the output frequency, users must to Freeze the DCO (bit 4 of Register 137) first. After configure the output frequency, users must Un-freeze the DCO and assert the NewFreq bit (bit 7 of Register 135).

■ Design Tools

- Quartus II 11.1 SP2
- Nios II Eclipse 11.1 SP2

■ Demonstration Source Code

- Quartus II Project directory: Nios_BASIC_DEMO
- Nios II Eclipse: Nios_BASIC_DEMO\Software

■ Nios II IDE Project Compilation

- Before you attempt to compile the reference design under Nios II Eclipse, make sure the project is cleaned first by clicking on 'Clean' in the 'Project' menu of Nios II Eclipse.

■ Demonstration Batch File

Demo Batch File Folder: *Nios_BASIC_DEMO\demo_batch*

The demo batch file includes following files:

- Batch File for USB-Blaster (II): test.bat, test_bashrc (test_ub2.bat, test_bashrc_ub2)
- FPGA Configure File: TR5_LITE_gondlen_top.*sof*
- Nios II Program: Nios_DEMO.*elf*

■ Demonstration Setup

- Make sure Quartus II and Nios II are installed on your PC.

- Power on the TR5-Lite board.
- Use the USB Blaster to connect your PC and the TR5-Lite board and install USB Blaster driver if necessary.
- Execute the demo batch file "*test.bat*"(or "*test_ub2.bat*" for USB-Blaster II) under the batch file folder, *Nios_BASIC_DEMO\demo_batch*
- After the Nios II program is downloaded and executed successfully, a prompt message will be displayed in nios2-terminal.
- For temperature test, please input key '0' and press 'Enter' in the nios-terminal, , as shown in **Figure 5-8**.
- For programming PLL CDCD61004 test, please input key '1' and press 'Enter' in the nios-terminal first, then select the desired output frequence , as shown in **Figure 5-9**.
- For programming PLL Si570 test, please input key '2' and press 'Enter' in the nios-terminal first, then select the desired output frequency , as shown in **Figure 5-10**.

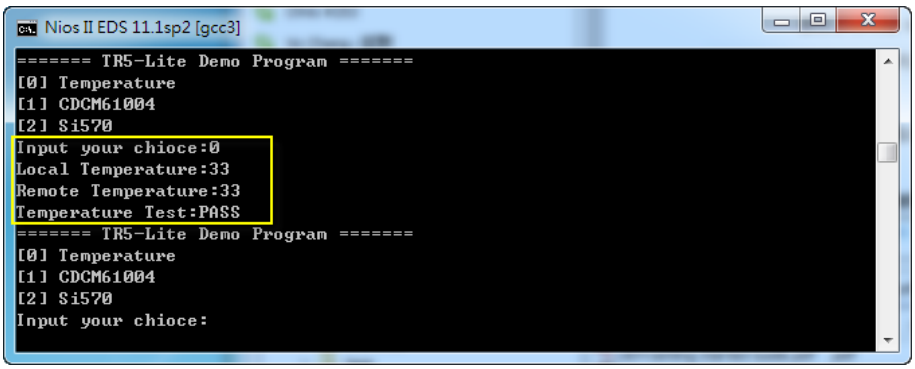


Figure 5-8 Temperature Demo

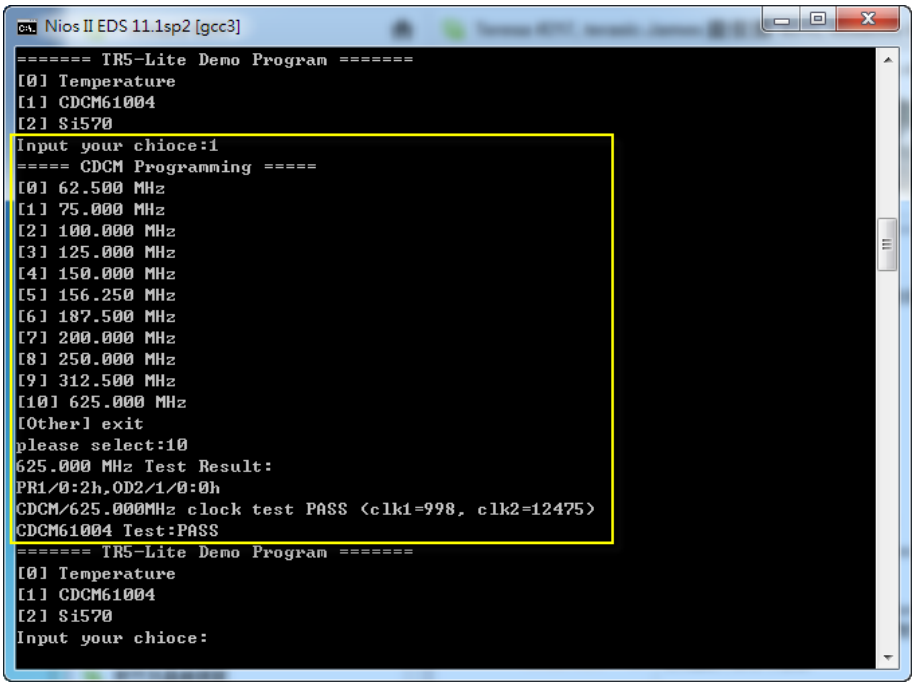


Figure 5-9 CDCM 61004 Demo

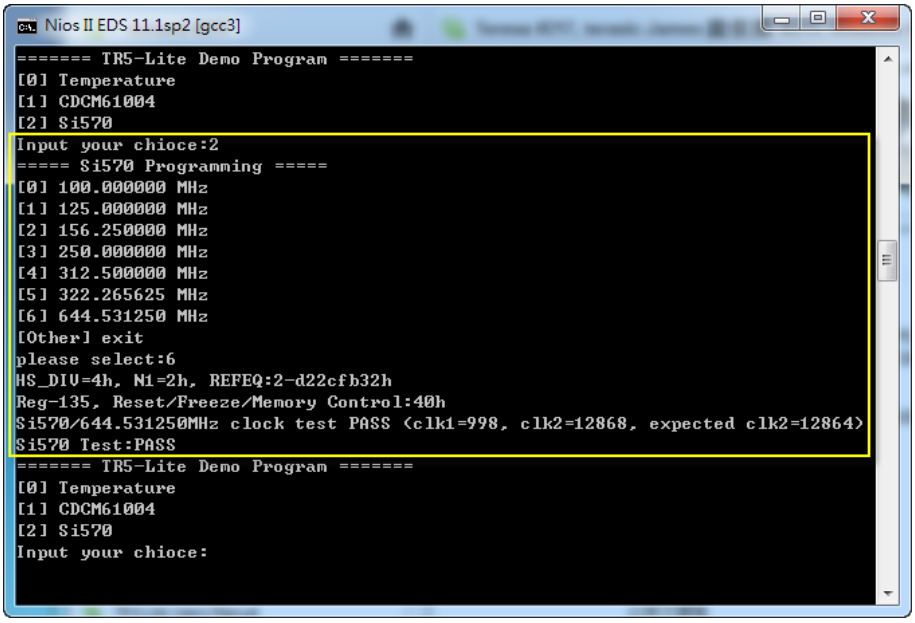


Figure 5-10 Si570 Demo

Chapter 6

Memory Reference Design

The TR5-Lite development board includes two kinds of high-speed memories:

- DDR3 SDRAM: two independent banks, update to 667 MHz
- QDRII+ SRAM: four independent banks, update to 550 MHz

This chapter will show three examples which use the Altera Memory IP to perform memory test functions. The source codes of these examples are all available on the TR5-Lite System CD. These three examples are:

- QDRII+ SRAM Test: Full test of the four banks of QDRII+ SRAM
- DD3 SDRAM Test: Random test of the two banks of DDR3 SDRAM.
- DDR3 SDRAM Test by Nios II: Full test of one bank of DDR3 SDRAM with Nios II

Note. 64-Bit Quartus 11.1 SP1 or later is strongly recommended for compiling these projects.

6.1 QDRII+ SRAM Test

QDR II/QDR II+ SRAM devices enable you to maximize memory bandwidth with separate read and write ports. The memory architecture features separate read and write ports operating twice per clock cycle to deliver a total of four data transfers per cycle. The resulting performance increase is particularly valuable in bandwidth-intensive and low-latency applications.

This demonstration utilizes four QDRII+ SRAMs on the TR5-Lite board. It describes how to use Altera's "QDRII and QDRII+ SRAM Controller with UniPHY" IP to implement a memory test function. In the design, the four QDRII controllers share the same PLL/DLL/OCT due to limited DLL numbers in the FPGA.

■ Function Block Diagram

Figure 6-1 shows the function block diagram of the demonstration. The four QDRII+ SRAM controllers are configured as a 72Mb controller. The QDRII+ SRAM IP generates a 550MHz clock as memory clock and a half-rate system clock, 275MHz, for the controllers.

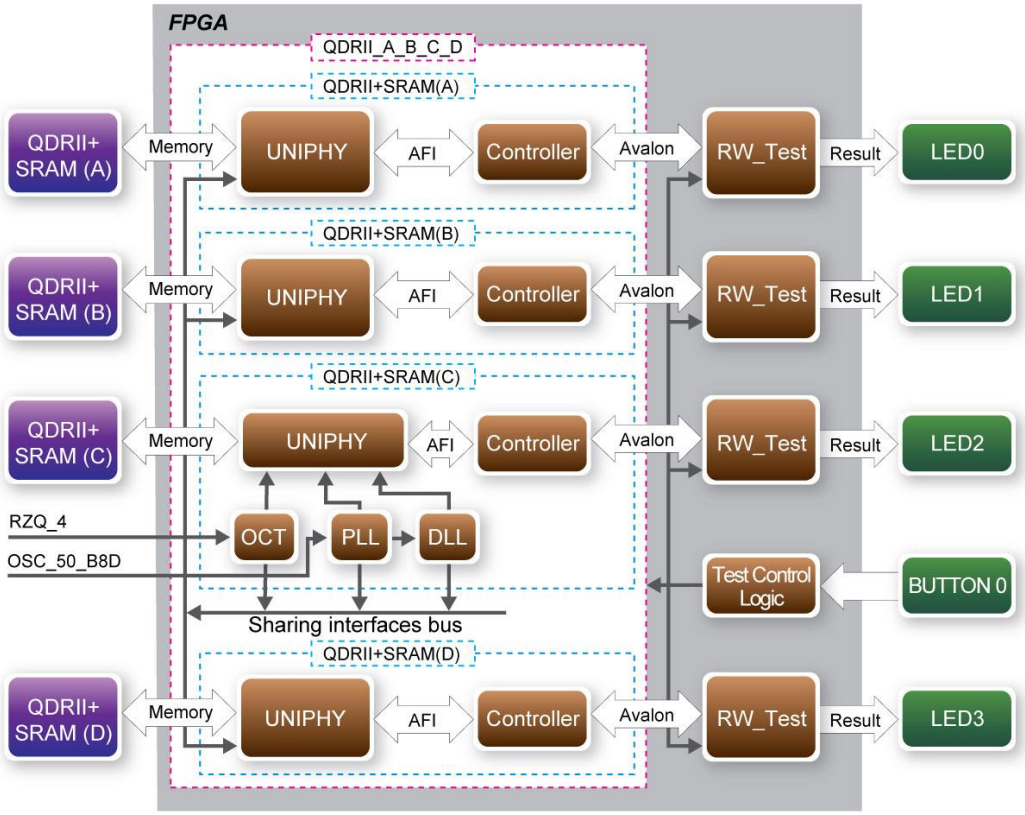


Figure 6-1 Function Block Diagram of the QDRII+ SRAM x4 Demonstration

In this demonstration, four QDRII+ SRAM controllers are sharing the FPGA resources (OCT, PLL, and DLL), and the QDRII+ SRAM (C) is configured as the master to share the resource to the other three slave QDRII+ SRAM (A/B/D). **RW_test** modules read and write the entire memory space of each QDRII+ SRAM through the Avalon interface of each controller. In this project, the Avalon bus read/write test module will first write the entire memory and then compare the read back data with the regenerated data (the same sequence as the write data). Test control signals for four QDRII+ SRAMs will generate from BUTTON0 and four LEDs will indicate the test results of four QDRII+ SRAMs.

■ Altera QDRII and QDRII+ SRAM Controller with UniPHY

To use Altera QDRII+ SRAM controller, users need to perform the following steps in order:

1. Create correct pin assignments for QDRII+.
2. Setup correct parameters in QDRII+ SRAM controller dialog.
3. Perform "Analysis and Synthesis" by clicking Quartus menu: Process→Start→Start Analysis & Synthesis.
4. Run the TCL files generated by QDRII+ IP by clicking Quartus menu: Tools→TCL Scripts…

■ Design Tools

- Quartus II v13.1

■ Demonstration Source Code

- Project directory: TR5_Lite_QDRIIx4_Test
- Bit stream used: TR5_Lite_QDRIIx4_Test.sof

■ Demonstration Batch File

Demo Batch File Folder: *TR5_Lite_QDRIIx4_Test\demo_batch*

The demo batch files include the followings:

- Batch file for USB-Blaster: TR5_Lite_QDRIIx4_Test.bat,
- Batch file for USB-Blaster II: TR5_Lite_QDRIIx4_Test_ub2.bat,
- FPGA configuration file: TR5_Lite_QDRIIx4_Test.sof

■ Demonstration Setup

- Make sure Quartus II is installed on your PC.
- Connect the USB Blaster cable (or USB-Blaster II cable) to the TR5_Lite board and host PC. Install the USB-Blaster driver if necessary.

- Power on the TR5_Lite board.
- Execute the demo batch file "*TR5_Lite_QDRIIx4_Test.bat*" (or "*TR5_Lite_QDRIIx4_Test_ub2.bat*" if use USB Blaster II) under the batch file folder, *TR5_Lite_QDRIIx4_Test\demo_batch*.
- Press **BUTTON0** of the TR5-Lite board to start the verification process. When **BUTTON0** is held down, all the **LEDs** will be turned off. All **LEDs** should turn back on to indicate test passes upon the release of **BUTTON0**.
- If any LED is not lit up after releasing **BUTTON0**, it indicates the corresponding QDRII+ SRAM test has failed. Table 6-1 lists the matchup for the four LEDs.
- Press **BUTTON0** again to regenerate the test control signals for a repeat test.

Table 6-1 LED Indicators

NAME	*Description*
LED0	**QDRII+ SRAM(A) test result**
LED1	**QDRII+ SRAM(B) test result**
LED2	**QDRII+ SRAM(C) test result**
LED3	**QDRII+ SRAM(D) test result**

6.2 DDR3 SDRAM Test

This demonstration presents a memory test function on the two banks of DDR3-SDRAM on the TR5-Lite. The memory size of each DDR3 SDRAM bank is 1GB.

Function Block Diagram

Figure 6-2 shows the function block diagram of this demonstration. There are two DDR3 SDRAM controllers. One is the master controller which shares resources with a slave controller. The shared resources include delay-locked loops (DLLs), phase-locked loops (PLLs), and on-chip termination (OCT). The controller uses 50 MHz as a reference clock, generates one 666.667 MHz clock as memory clock, and generates one quarter-rate system clock 166.666 MHz for the controller itself.

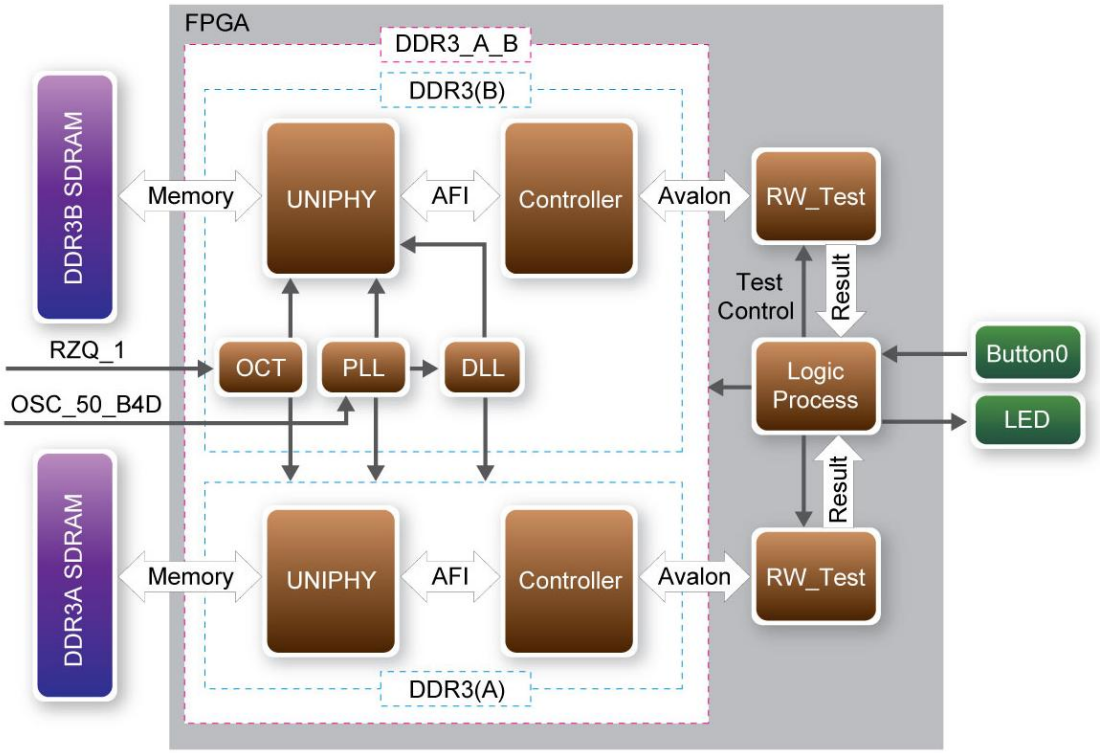


Figure 6-2 Block Diagram of the DDR3 SDRAM (1G) x2 Demonstration

RW_test modules read and write the entire memory space of each DDR3 through the Avalon interface of each controller. In this project, the Avalon bus read/write test module will first write the entire memory and then compare the read back data with the regenerated data (the same sequence as the write data). BUTTON0 will trigger test control signals for the two DDR3, and the LEDs will indicate the test results according to **Table 6-2**.

Altera DDR3 SDRAM Controller with UniPHY

To use the Altera DDR3 controller, users need to perform three major steps:

1. Create correct pin assignments for the DDR3.
2. Setup correct parameters in DDR3 controller dialog.
3. Perform "Analysis and Synthesis" by selecting from the Quartus II menu: Process→Start→Start Analysis & Synthesis.
4. Run the TCL files generated by DDR3 IP by selecting from the Quartus II menu:

Tools→TCL Scripts…

Design Tools

- 64-Bit Quartus 13.1

Demonstration Source Code

- Project directory: TR5_Lite_DDR3x2_Test
- Bit stream used: TR5_Lite_DDR3x2_Test.sof

Demonstration Batch File

Demo Batch File Folder: *TR5_Lite_DDR3x2_Test \demo_batch*

The demo batch file includes following files:

- Batch File: TR5_Lite_DDR3x2_Test.bat
- Batch file for USB-Blaster II: TR5_Lite_DDR3x2_Test_ub2.bat,
- FPGA Configure File: TR5_Lite_DDR3x2_Test.sof

Demonstration Setup

- Make sure Quartus II is installed on your PC.
- Connect the USB Blaster cable (or USB-Blaster II cable) to the TR5-Lite board and host PC. Install the USB Blaster driver if necessary.
- Power on the TR5-Lite board.
- Execute the demo batch file "TR5_Lite_DDR3x2_Test.bat" (or "*TR5_Lite_DDR3x2_Test_ub2.bat*" if use USB Blaster II) under the batch file folder, TR5_Lite_DDR3x2_Test \demo_batch.
- Press **BUTTON0** on the **TR5-Lite** board to start the verification process. When **BUTTON0** is pressed, all the **LEDs** (**LED** [3:0]) should turn on. At the instant of releasing **BUTTON0**, **LED1**, **LED2**, **LED3** should start blinking. After approximately 13 seconds, **LED1** and **LED2** should stop blinking and stay on to indicate that the DDR3 (A) and DDR3 (B) have passed the test, respectively. Table 4-2 lists the **LED** indicators.
- If **LED3** is not blinking, it means the 50MHz clock source is not working.

- If **LED1** or **LED2** do not start blinking after releasing BUTTON0, it indicates local_init_done or local_cal_success of the corresponding DDR3 failed.
- If **LED1** or **LED2** fail to remain on after 13 seconds, the corresponding DDR3 test has failed.
- Press **BUTTON0** again to regenerate the test control signals for a repeat test.

Table 6-2 LED Indicators

NAME	*Description*
LED0	**Reset**
LED1	**If lit, DDR3 (A) test pass**
LED2	**If lit, DDR3 (B) test pass**
LED3	**Blinks**

6.3 DDR3 SDRAM Test by Nios II

Many applications use a high performance RAM, such as a DDR3 SDRAM, to provide temporary storage. In this demonstration hardware and software designs are provided to illustrate how to perform DDR3 memory access in QSYS. We describe how the Altera's "DDR3 SDRAM Controller with UniPHY" IP is used to access a DDR3-SDRAM, and how the Nios II processor is used to read and write the SDRAM for hardware verification. The DDR3 SDRAM controller handles the complex aspects of using DDR3 SDRAM by initializing the memory devices, managing SDRAM banks, and keeping the devices refreshed at appropriate intervals.

■ System Block Diagram

Figure 6-3 shows the system block diagram of this demonstration. The system requires a 50 MHz clock provided from the board. The DDR3 controller is configured as a 1 GB DDR3-666.667 controller. The DDR3 IP generates one 666.667 MHz clock as SDRAM's data clock and one quarter-rate system clock 666.667/4=166.666 MHz for those host controllers, e.g. Nios II processor, accessing the SDRAM. In the QSYS, Nios II and the On-Chip Memory are designed running with the 166.666 MHz clock, and the Nios II program is running in the on-chip memory.

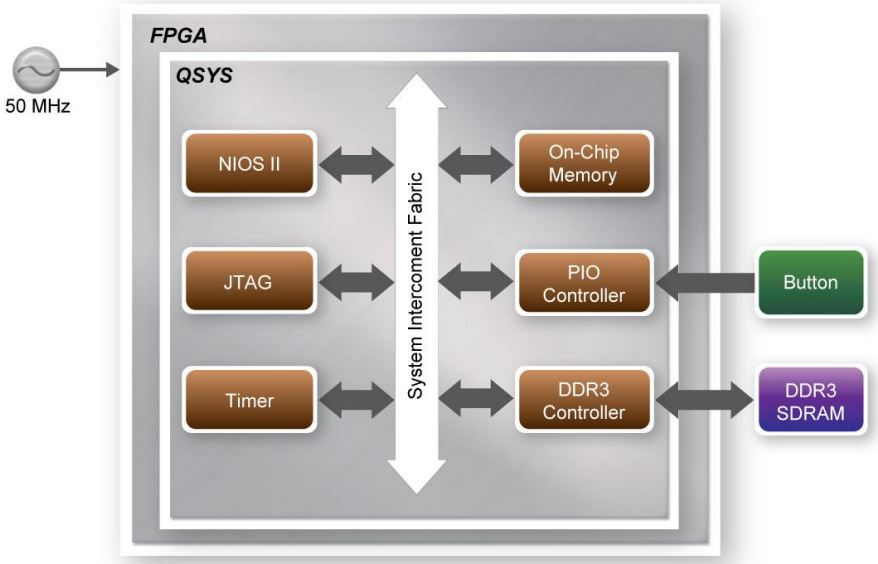


Figure 6-3 Block diagram of the DDR3 Basic Demonstration

The system flow is controlled by a Nios II program. First, the Nios II program writes test patterns into the whole 1 GB of SDRAM. Then, it calls Nios II system function, alt_dache_flush_all, to make sure all data has been written to SDRAM. Finally, it reads data from SDRAM for data verification. The program will show progress in JTAG-Terminal when writing/reading data to/from the SDRAM. When verification process is completed, the result is displayed in the JTAG-Terminal.

■ Altera DDR3 SDRAM Controller with UniPHY

To use Altera DDR3 controller, users need to perform the four major steps:

1. Create correct pin assignments for DDR3.
2. Setup correct parameters in DDR3 controller dialog.
3. Perform "Analysis and Synthesis" by clicking Quartus menu: Process→Start→Start Analysis & Synthesis.
4. Run the TCL files generated by DDR3 IP by clicking Quartus menu: Tools→TCL Scripts…

■ Quartus II Project

The Quartus II project is designed to only access DDR3-A or DDR3-B at same time due to the address space limitation of Nios II. Users can change the accessed memory target at Quartus compile time by defining the constant USE_DDR3_A for DDR3-A or constant USE_DDR3_B for DDR3-B bank. After the constant is defined, please perform Analysis and Synthesis and then run the TCL files generated by DDR3 IP before starting Quartus II compilation.

■ Design Tools

- Quartus II 13.1
- Nios II Eclipse 13.1

■ Demonstration Source Code

- Quartus Project directory: Nios_DDR3
- Nios II Eclipse: Nios_DDR3\Software

■ Nios II Project Compilation

Before you attempt to compile the reference design under Nios II Eclipse, make sure the project is cleaned first by clicking 'Clean' from the 'Project' menu of Nios II Eclipse.

■ Demonstration Batch File

Demo Batch File Folder:

Nios_DDR3\demo_batch\DDR3_A_667HZ or
Nios_DDR3\demo_batch\DDR3_B_667MHZ

The demo batch file includes following files:

- Batch File for USB-Blaseter (II): test.bat, test_bashrc (test_ub2.bat, test_bashrc_ub2)
- FPGA Configure File: TR5_LITE_gondlen_top.*sof*
- Nios II Program: TEST_DDR3.*elf*

■ Demonstration Setup

- Make sure Quartus II and Nios II are installed on your PC.
- Power on the TR5-Lite board.
- Use USB Blaster to connect PC and the TR5-Lite board and install USB Blaster driver if necessary.
- Execute the demo batch file "*test.bat*"(or "*test_ub2.bat*" for USB-Blaster II) under the batch file folder, *Nios_DDR3\demo_batch\DDR3_A_667MHZ* or *Nios_DDR3\demo_batch\DDR3_B_667MHZ*
- After Nios II program is downloaded and executed successfully, a prompt message will be displayed in nios2-terminal.
- Press **Button1~Button0** of the TR5-Lite board to start SDRAM verify process. Press **Button0** for continued test and press any to terminate the continued test**.**
- The program will display progressing and result information, as shown in **Figure 6-4**.

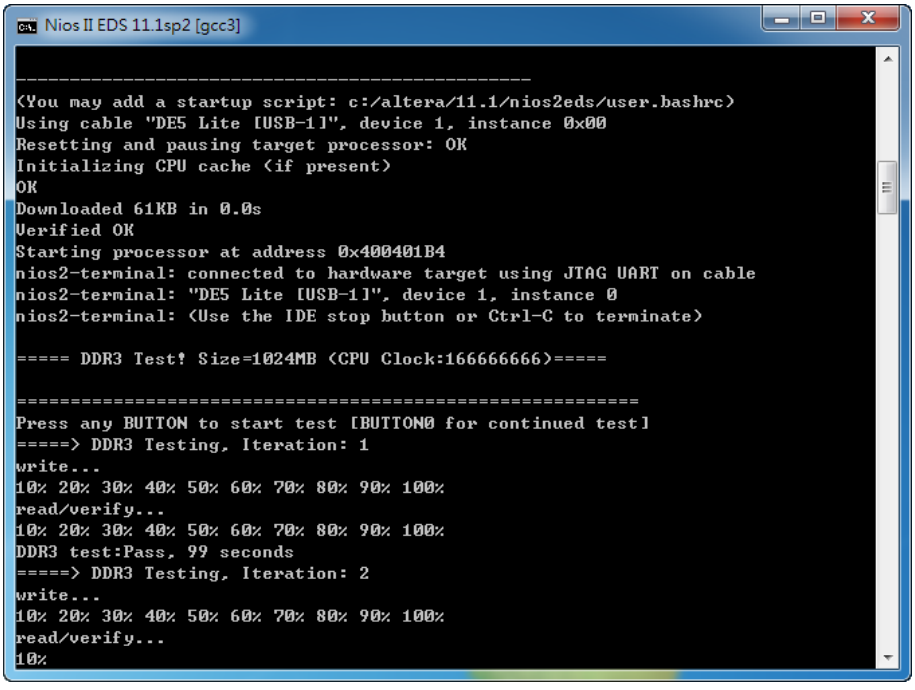


Figure 6-4 Display Progress and Result Information for the DDR3 Demonstration

Chapter 7

Transceiver Verification

This chapter describes how to verify the FPGA transceivers using the test code provided in the DE5-Net system CD.

7.1 Test Code

The transceiver test code verifies the transceiver channels through an external loopback method. The following transceiver channels can be verified with different data rates:

- **10.3125 Gbps:** SPF-A, SPF-B
- **6.0 Gbps:** SATA Host
- **8.0 Gbps:** PCIe Channel 0~7

7.2 Loopback Fixture

To enable an external loopback of transceiver channels, specific loopback fixtures are required. Some fixtures may be proprietary to Terasic.

For SFP+ loopback, optical SFP+ loopback fixtures are required. **Figure 7-1** shows the optical SFP+ loopback fixture.

Figure 7-1 Optical SFP+ Loopback Fixture

.

Figure 7-2 shows the SATA loopback fixture.

Figure 7-2 SATA Loopback Fixture

Figure 7-3 shows the Terasic PCIe loopback fixture.

Figure 7-3 PCIe Loopback Fixture

Figure 7-4 shows the FPGA board with all transceiver loopback fixtures installed.

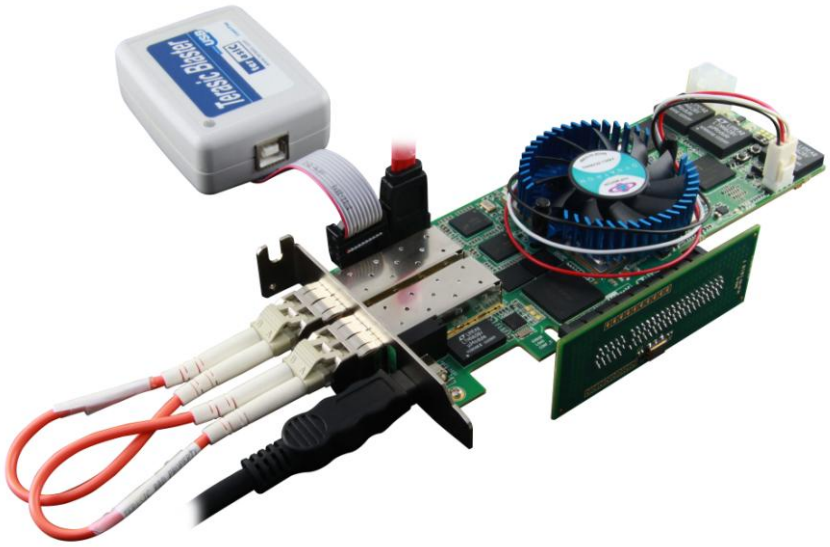


Figure 7-4 Transceiver Loopback Fixtures Installed

7.3 Testing

The transceiver test code is available in the folder System CD\Tool\Transceiver_Test. Here are the procedures to perform transceiver channel test:

1. Copy **Transceiver_Test** folder to your local disk.
2. Ensure that the FPGA board is NOT powered on.
3. Plug-in the SPF+ loopback fixtures if the SPF+ transceivers will be tested.
4. Plug-in the SATA loopback fixtures if the SATA transceivers will be tested.
5. Plug-in the PCIe loopback fixture if PCIe transceivers will be tested. Also, make sure PCIe Mode SW7 is switched to x8 mode.
6. Connect your FPGA board to your PC with an USB-Blastger cable.
7. Power on the FPGA board
8. Execute 'xcvr_test.bat" in the **Transceiver_Test** folder under your local disk.
9. The batch file will download .sof and .elf files, and start the test immediately. The test result is shown in the Nios-Terminal, as shown in **Figure 7-5.**
10. To terminate the test, press one of the BUTTON0~1 buttons on the FPGA board. The loopback test will terminate, and the test summary will be shown in the Nios-Terminal, as shown in **Figure 7-5**.

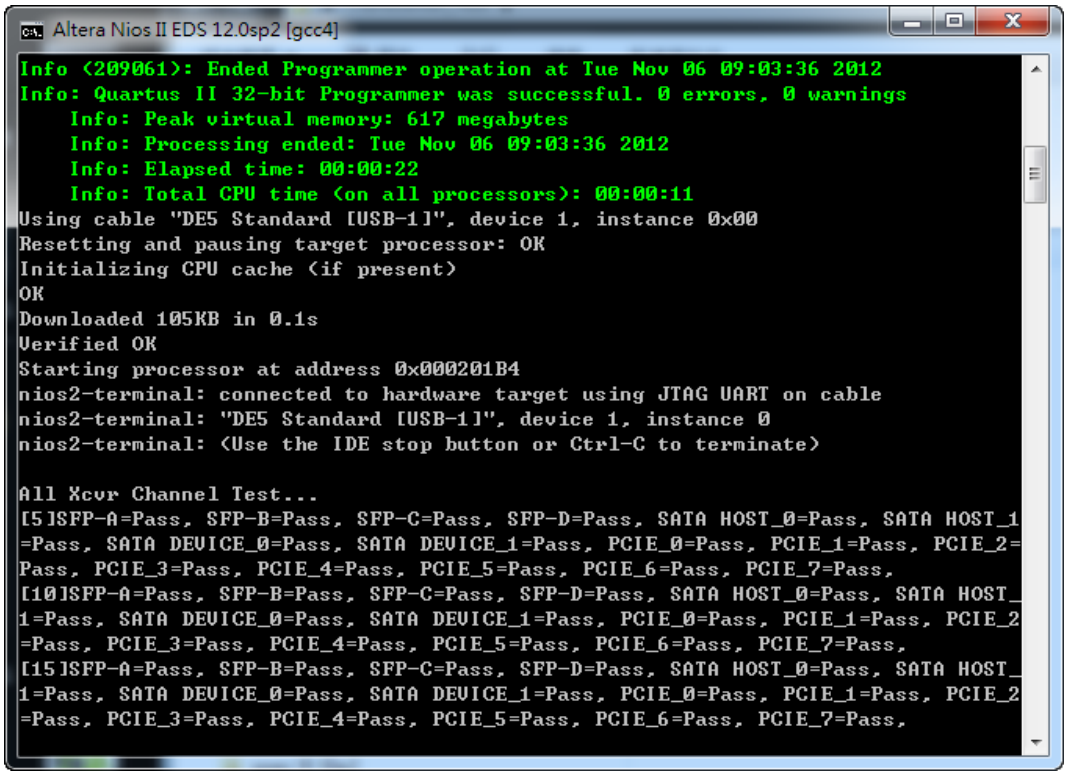


Figure 7-5 Transceiver Loopback Test in Progress

Additional Information

Getting Help

Here are the addresses where you can get help if you encounter problems:

- Terasic Technologies
  9F., No.176, Sec.2, Gongdao 5th Rd,
  East Dist, HsinChu City, 30070. Taiwan, 30070
  Email: support@terasic.com
  Web: www.terasic.com
  TR5-Lite Web: tr5-lite.terasic.com

Revision History

Date	*Version*	*Changes*
2012.4	First publication	
2014.10	V1.1	*Update section 5.2 for modifying si570 function*
2017.4	V1.2	*Modify PCIE demo*
2018.6	V1.3	*Remove PCIE chapter, PCIe demo and manual changed to standalone CD*