ML628 IBERT Getting Started Guide (ISE 13.2)

UG806 (v2.0) July 14, 2011





© Copyright 2011 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners. DISCLAIMER

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials, or to advise you of any corrections or update. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/20/11	1.0	Initial Xilinx release.
07/14/11	2.0	Revised links, software references, and figures containing screen captures to reflect ISE software v13.2.

Table of Contents

Revision History	
apter 1: ML628 IBERT Getting Started Guide	
Overview	
Requirements	
Setting Up the ML628 Board	
Extracting the Project Files	
Running the GTH IBERT Demonstration	
Connecting the GTH Transceivers and Reference Clocks	
Attach the GTH Quad Connector	
GTH Transceiver Clock Connections	
GTH TX/RX Loopback Connections	
Configuring the FPGA	
Setting Up the ChipScope Pro Software	
Starting the SuperClock-2 Module	
Viewing GTH Transceiver Operation	
In Case of RX Bit Errors	
DC Blocks	
Closing the IBERT Demonstration	
More Information	
Running the GTX IBERT Demonstration	
Connecting the GTX Transceivers and Reference Clocks	•••••
Attach the GTX Quad Connector	
GTX Transceiver Clock Connections	
GTX TX/RX Loopback Connections.	
Configuring the FPGA	
Setting Up the ChipScope Pro Software	
Starting the SuperClock-2 Module	
Viewing GTX Transceiver Operation	
Closing the IBERT Demonstration	
More Information	
SuperClock-2 Frequency Table	
IBERT Demonstration Designs	
Source Directories and Files	
IBERT Design Files	
example_ibert_v6_q1xx_top.bit	
example_implement_ibert_v6_q1xx.prj	
example_implement_ibert_v6_q1xx.xst	
i2c_sclk2_control.ngc	
ibert_v6_q1xx.ngc	
icon_v6_1.ngc	
vio_v6_si84_so78.ngc	
implement.bat	
example_ibert_v6_q1xx.v	
i2c_sclk2_control_bb.v	
ibert_v6_qxxx_top.ucf	

vio_sclk2_control.v	. 36
Creating the GTH IBERT Core	. 36
Creating the GTX IBERT Core	. 43
Warranty	. 51



Chapter 1

ML628 IBERT Getting Started Guide

Overview

This document provides a procedure for setting up the ML628 Virtex®-6 FPGA GTX and GTH Transceiver Characterization Board to run the Integrated Bit Error Ratio Test (IBERT) demonstration. The designs that are required to run the IBERT demonstration are stored in a CompactFlash (CF) memory card that is provided with the ML628 board. The demonstration shows the capabilities of the Virtex-6 XC6VHX380T FPGA GTX and GTH transceivers.

The ML628 board is described in detail in <u>UG771</u>, *ML628 Virtex-6 FPGA GTX and GTH Transceiver Characterization Board User Guide*.

The IBERT demonstrations operate one GTH or GTX Quad at a time. Separate procedures are provided for GTH and GTX transceiver evaluation. The GTH procedure consists of:

- 1. Setting Up the ML628 Board.
- 2. Extracting the Project Files.
- 3. Connecting the GTH Transceivers and Reference Clocks.
- 4. Configuring the FPGA.
- 5. Setting Up the ChipScope Pro Software.
- 6. Viewing GTH Transceiver Operation.
- 7. Closing the IBERT Demonstration.

The GTX procedure consists of:

- 1. Setting Up the ML628 Board.
- 2. Extracting the Project Files.
- 3. Connecting the GTX Transceivers and Reference Clocks.
- 4. Configuring the FPGA.
- 5. Setting Up the ChipScope Pro Software.
- 6. Viewing GTX Transceiver Operation.
- 7. Closing the IBERT Demonstration.

Requirements

The hardware and software required to run the GTX or GTH IBERT demonstrations are:

- ML628 Virtex-6 FPGA GTH and GTX Transceiver Characterization Board including:
 - Two CF cards containing the IBERT demonstration designs (*IBERT #1* and *IBERT #2*)
 - One Samtec BullsEye cable
 - Eight SMA female-to-female (F-F) adapters
 - Eight SMA (female-to-male) DC Blocks (optional)
 - GTH transceiver power supply module (installed on board)
 - GTX transceiver power supply module (installed on board)
 - SuperClock-2 module (installed on board)
 - Active BGA Heatsink (installed on FPGA)
 - 12V DC power adapter
- JTAG cable:
 - Platform Cable USB (Xilinx part number DLC9, DLC9G, or DLC9LP) or,
 - Platform Cable USB II (Xilinx part number DLC10) or,
 - Parallel Cable IV (Xilinx part number PC4)
- Host PC with:
 - CF card reader or PCMCIA adapter card for CF cards
 - USB ports
- Xilinx[®] ChipScope[™] Pro software, version 13.2 or higher. Software is available at: <u>http://www.xilinx.com/chipscopepro</u>

The hardware and software required to rebuild the IBERT demonstration designs are:

- Xilinx ISE® Design Suite version 13.2 or higher
- PC with a version of the Windows operating system supported by Xilinx ISE Design Suite
- ML628 IBERT demonstration design source files (provided online as collection rdf0117_13-2.zip) at:

http://www.xilinx.com/products/boards/ml628/reference_designs.htm

Setting Up the ML628 Board

This section describes how to set up the ML628 Virtex-6 FPGA GTH and GTX Transceiver Characterization Board.

Caution! The ML628 board can be damaged by electrostatic discharge (ESD). Follow standard ESD prevention measures when handling the board.

When the ML628 board ships from the factory, it is configured for the GTH and GTX IBERT demonstrations described in this document. If the board has been re-configured it must be returned to the default set-up before running the IBERT demonstrations.

1. Move all jumpers to their default positions. The default jumper positions are listed in <u>UG771</u>, *ML628 Virtex-6 FPGA GTX and GTH Transceiver Characterization Board User Guide*.

- 2. Install the GTX and GTH transceiver power modules:
 - a. Plug the GTX transceiver power module into connectors J34 and J179.
 - b. Plug the GTH transceiver power module into connectors J6 and J197.
- 3. Install the SuperClock-2 module:
 - a. Align the three metal standoffs on the bottom side of the module with the three mounting holes in the CLOCK MODULE interface of the ML628 board.
 - b. Using three 4-40 x 0.25 inch screws, firmly screw down the module from the bottom of the ML628 board.
 - c. On the SuperClock-2 module, place a jumper across pins 2–3 (2V5) of the CONTROL VOLTAGE header, J18.

Extracting the Project Files

The ChipScope Pro Software .cpj project files for the IBERT demonstrations are located in ml628_cpj.zip on the CF card labeled *IBERT* #1. They are also available online along with .bit files for all 16 designs (as collection rdf0116_13-2.zip) at:

http://www.xilinx.com/products/boards/ml628/reference_designs.htm

ml628_cpj.zip contains two project files:ml628_gth.cpj and ml628_gtx.cpj. These files are used to load pre-saved MGT/IBERT and SuperClock-2 module control settings for the GTH and GTX demonstrations.

To copy the files from the CompactFlash memory card:

1. Connect the CompactFlash memory card to the host computer.

Note: The CompactFlash memory card can be plugged into a host PC's PCMCIA interface using a PCMCIA adapter card.

- 2. Locate the file ml628_cpj.zip on the Compact Flash memory card.
- 3. Unzip the files to a working directory on the host computer.

Running the GTH IBERT Demonstration

The GTH IBERT demonstration operates one GTH Quad at a time. This section describes how to test GTH Quad 117. The remaining GTH Quads are tested following a similar series of steps. The GTX IBERT demonstration is described in Running the GTX IBERT Demonstration, page 20.

Connecting the GTH Transceivers and Reference Clocks

Figure 1-1 shows the locations for GTH transceiver Quads 106, 107, 108, 116, 117, and 118 on the ML628 board.

Note: Figure 1-1 is for reference only and might not reflect the current revision of the board.

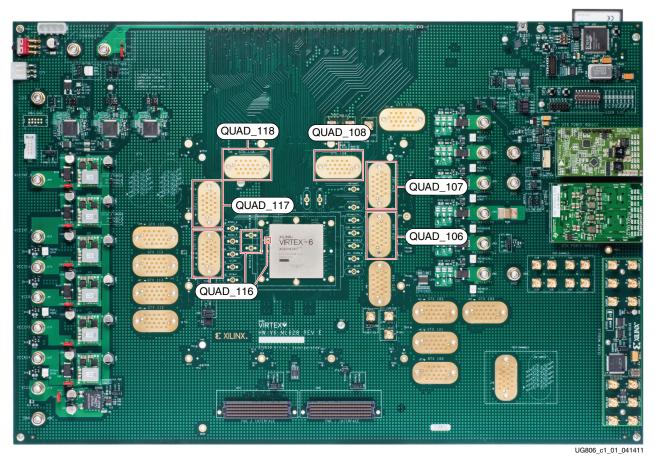


Figure 1-1: GTH Quad Locations

With the exception of GTH Quad 116, all GTH transceiver pins and reference clock pins are routed from the FPGA to a connector pad which interfaces with Samtec BullsEye connectors. Figure 1-2 **A** shows the connector pad. Figure 1-2 **B** shows the connector pinout.

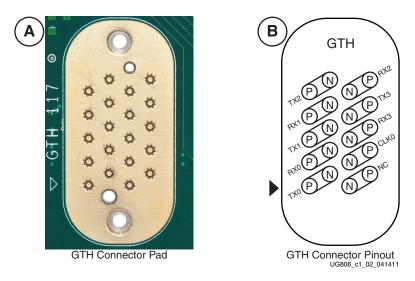


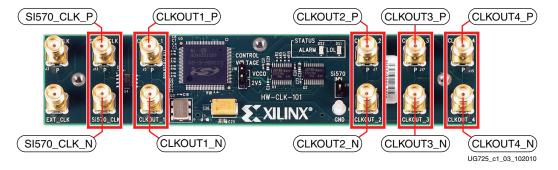
Figure 1-2: A – GTH Connector Pad. B – GTH Connector Pinout

www.xilinx.com

Transmitter pairs TX0 and TX1 on Quad 116 are not connected to the BullsEye connector pads. Refer to <u>UG771</u>, *ML628 Virtex-6 FPGA GTX and GTH Transceiver Characterization* Board User Guide for details on accessing these signals.

The SuperClock-2 module provides LVDS clock outputs for the GTH and GTX transceiver reference clocks in the IBERT demonstrations. Figure 1-3 shows the locations of the differential clock SMA connectors on the clock module which can be connected to the reference clock cables. The four SMA pairs labeled CLKOUT provide LVDS clock outputs from the Si5368 clock multiplier/jitter attenuator device on the clock module. The SMA pair labeled Si570_CLK provides LVPECL clock output from the Si570 programmable oscillator on the clock module. For the GTH IBERT demonstration, the output clock frequencies are preset to 174.690 MHz. For more information regarding the SuperClock-2 module, refer to UG770, HW-CLK-101-SCLK2 SuperClock-2 Module User Guide.

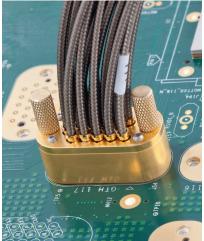
Note: The image in Figure 1-3 is for reference only and might not reflect the current revision of the board.





Attach the GTH Quad Connector

Attach the Samtec BullsEye connector to the connector pad for GTH Quad 117 (Figure 1-4), aligning the two indexing pins on the bottom of the connector with the guide holes on the board. Hold the connector flush with the board and fasten it by tightening the two captive screws.



UG806_c1_04_022411

Figure 1-4: BullsEye Connector Attached to Quad 117

www.xilinx.com

GTH Transceiver Clock Connections

Refer to Figure 1-2, page 8 to identify the P and N coax cables that are connected to the reference clock (CLK0). Connect these cables to the SuperClock-2 Module as follows:

- CLK0_P coax cable \rightarrow SMA connector J5 (CLKOUT1_P on the SuperClock-2 Module)
- CLK0_N coax cable \rightarrow SMA connector J6 (CLKOUT1_N on the SuperClock-2 Module)

Note: Any one of the five differential outputs from the SuperClock-2 Module can be used to source the GTH reference clock. CLKOUT1_P and CLKOUT1_N are used here as an example.

GTH TX/RX Loopback Connections

Refer to Figure 1-2, page 8 to identify the P and N coax cables that are connected to the four receivers (RX0, RX1, RX2 and RX3) and the four transmitters (TX0, TX1, TX2 and TX3). Use eight SMA female-to-female (F-F) adapters (Figure 1-5) to connect the transmit and receive cables as shown in Figure 1-6 and as detailed below:

- $TX0_P \rightarrow SMA F-F Adapter \rightarrow RX0_P$
- $TX0_N \rightarrow SMA F-F Adapter \rightarrow RX0_N$
- $TX1_P \rightarrow SMA F-F Adapter \rightarrow RX1_P$
- $TX1_N \rightarrow SMA F-F Adapter \rightarrow RX1_N$
- $TX2_P \rightarrow SMA F-F Adapter \rightarrow RX2_P$
- TX2_N \rightarrow SMA F-F Adapter \rightarrow RX2_N
- $TX3_P \rightarrow SMA F-F Adapter \rightarrow RX3_P$
- TX3_N \rightarrow SMA F-F Adapter \rightarrow RX3_N



Figure 1-5: **SMA F-F Adapter**



UG806_c1_06_022811

Figure 1-6: TX-To-RX Loopback Connection Example

Figure 1-7 shows the ML628 board with the cable connections required for the Quad 117 GTH IBERT demonstration.

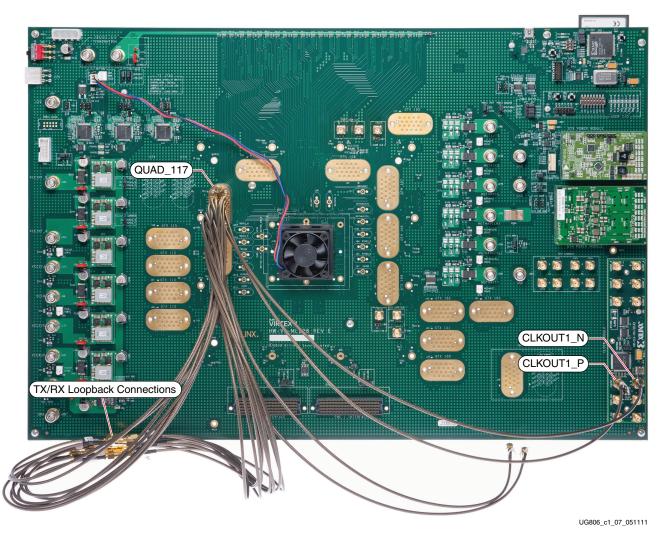


Figure 1-7: Cable Connections for Quad 117 GTH IBERT Demonstration

Configuring the FPGA

This section describes how to configure the FPGA using the CF cards included with the board. The FPGA can also be configured through ChipScope Pro or iMPACT software using the .bit files which are available online (as collection rdf0116_13-2.zip) at:

http://www.xilinx.com/products/boards/ml628/reference_designs.htm

To configure from the CF card:

- 1. Insert the CF card labeled *IBERT* #1 into the CF card reader slot located on the bottom-side (upper-right corner) of the board.
- 2. Plug the 12V output from the power adapter into connector J122.
- 3. Connect the programming cable to the host computer. Any of these cables can be used:
 - Platform Cable USB-II (DLC10)
 - Platform Cable USB (DLC9, DLC9G or DLC9LP)

• Parallel Cable IV (PC4)

Connect the ribbon cable between the programming cable and the ML628 board at JTAG connector J1.

4. Select the GTH IBERT demonstration with the System Ace[™] controller CFG ADDRESS switch, SW3. The setting on this 3-bit DIP switch (Figure 1-8) selects the file used to configure the FPGA. A switch is in the ON position if it switched towards the ON text printed on the DIP; otherwise, it is OFF. For the Quad 117 GTH IBERT demonstration, set: ADR2 = ON, ADR1 = OFF, and ADR0 = OFF.

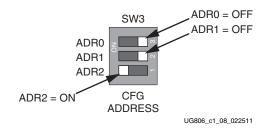


Figure 1-8: Configuration Address DIP Switch (SW3)

There is one IBERT demonstration design for each GTH and GTX Quad on the ML628 board, for a total of 16 designs. The designs are organized and stored on the two CF cards (*IBERT #1* and *IBERT #2*) as shown in Table 1-1.

CF Card	Demonstration Design	ADR2	ADR1	ADR0
	GTH Quad 106	OFF	OFF	OFF
	GTH Quad 107	OFF	OFF	ON
	GTH Quad 108	OFF	ON	OFF
IBERT #1	GTH Quad 116	OFF	ON	ON
IDENI #1	GTH Quad 117	ON	OFF	OFF
	GTH Quad 118	ON	OFF	ON
	GTX Quad 100	ON	ON	OFF
	GTX Quad 101	ON	ON	ON
	GTX Quad 102	OFF	OFF	OFF
	GTX Quad 103	OFF	OFF	ON
	GTX Quad 104	OFF	ON	OFF
IBERT #2	GTX Quad 105	OFF	ON	ON
IDENI #2	GTX Quad 112	ON	OFF	OFF
	GTX Quad 113	ON	OFF	ON
	GTX Quad 114	ON	ON	OFF
	GTX Quad 115	ON	ON	ON

Table 1-1: CF Card Contents and Configuration Addresses

5. Place the main power switch SW1 to the ON position.

Setting Up the ChipScope Pro Software

- 1. Start the ChipScope Pro analyzer tool on the host computer and select **File** \rightarrow **Open Project**.
- 2. When the Project window opens, navigate to the directory where the ChipScope software project files (.cpj) were extracted. Select ml628_gth.cpj and click **Open**.

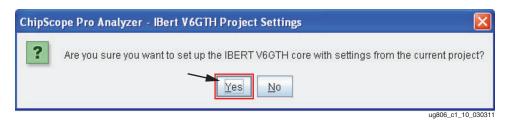
The .cpj file loads pre-saved project settings for the demonstration including MGT/ IBERT and clock module control parameters. For more information regarding MGT/ IBERT settings, refer to <u>www.xilinx.com/support/documentation/sw_manuals/</u> <u>xilinx13_2/chipscope_pro_sw_cores_ug029.pdf</u>, UG029 - *ChipScope Pro Software Cores*.

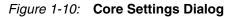
3. Click the **Open Cable** button (Figure 1-9).

ChipScope Pro Analyzer [ne ChipScope Pro Analyzer [ne	
New Project JTAG Chain	ChipScope Pro
	UG806 c1 09 02251

Figure 1-9: Open Cable Button

4. When the dialog opens asking to set up the core with the settings from the current project, click **Yes** (Figure 1-10).





Note: After completing step 4, the IBERT Console will open and display **No Link** on all four lanes (Figure 1-11) and an error message will appear in the status panel at the bottom of the GUI (Figure 1-12). This is due to a limitation in this revision of the GTH IBERT core. At this point the IBERT core is loaded and running, but the outputs of the Si5368 on the SuperClock-2 module

www.xilinx.com

are not enabled. Normally, a graceful re-start can be achieved by enabling the reference clock and resetting the core. However, the issue is that the signal from the QUAD Reset button does not reach the reset logic in the core. The workaround is to reload the core which is described later in this procedure. This limitation will be fixed in a later release.

Note: Unlike the Si5368, the Si570 on the SuperClock-2 module is an always-on clock source. As such, this problem will not be observed if this procedure is run with the GTH reference clock inputs connected to Si570 outputs on the SuperClock-2 module (Si570_CLK_P and Si570_CLK_N).

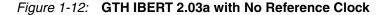
Note: This is a limitation of the GTH IBERT core only. It does not exist in the GTX IBERT core.

GT/BERT Settings DRP	Settings Port Settings Swe	ep Test Settings		
	GTHE1_QUAD_X1Y1_0	GTHE1_QUAD_X1Y1_1	GTHE1_QUAD_X1Y1_2	GTHE1_QUAD_X1Y1_3
MGT Settings				
- DFETAPOVRD				
– MGT Alias	QUAD117_0	QUAD117_1	QUAD117_2	QUAD117_3
- DFETAP1	0	0	0	0
 Tile Location 	GTHE1_QUAD_X1Y1	GTHE1_QUAD_X1Y1	GTHE1_QUAD_X1Y1	GTHE1_QUAD_X1Y1
- DFETAP2	0	0	0	0
- MGT Link Status	No Link	No Link	No Link	No Link
- DFETAP3	0	0	0	0
 PLL Status 	NOT LOCKED	NOT LOCKED	NOT LOCKED	NOT LOCKED
- AGCOVRD				
– Loopback Mode	None 🗸	None	None 💌	None
- AGC	0	0	0	0
- QUAD Reset	Reset	Reset	Reset	Reset
 TX Error Inject 	Inject	Inject	Inject	Inject
 TX Diff Output Swing 	450 mV 💌	450 mV 💌	450 mV 💌	450 mV
– TX Pre-Emphasis	0 💌	0	0	0
– TX Post-Emphasis	0 💌	0	0	0
RX Equalization	0.15 dB 💌	0.15 dB 💌	0.15 dB 💌	0.15 dB
BERT Settings				
– TX Data Pattern	PRBS 31-bit 💌	PRBS 31-bit 💌	PRBS 31-bit	PRBS 31-bit
– RX Data Pattern	PRBS 31-bit	PRBS 31-bit 💌	PRBS 31-bit	PRBS 31-bit
 RX Bit Error Ratio 	1	1	1	1
 RX Received Bit Count 	0.000E000	0.000E000	0.000E000	0.000E000
 RX Bit Error Count 	0.000E000	0.000E000	0.000E000	0.000E000
BERT Reset	Reset	Reset	Reset	Reset
Clocking Settings				

Figure 1-11: GTH IBERT 2.03a with No Reference Clock

INFO: ChipScope Pro Analyzer Version: 13.2 O.61xb (Build 13200.11.158.813)
COMMAND: open_project "C:ldataWL628_Production\"ise\"ibert_demo\ml628_gth.cpj"
COMMAND: open_cable
INFO: Started ChipScope host (localhost:50001)
INFO: Successfully opened connection to server: localhost50001 (localhost/127.0.0.1)
INFO: Trying to open Xilinx Platform USB Cable on port USB2
INFO: Successfully opened Xilinx Platform USB Cable
INFO: Cable: Platform Cable USB II, Port: USB21, Speed: 3 MHz
INFO: Found 2 Core Units in the JTAG device Chain.
ERROR - Device 1 Unit 1 0: Either system clock or reference clock for GTHET_QUAD_X1Y1_0 is not running: JTAG Scan Rate is set to "Manual Scan".
ERROR - Device 1 Unit 1_0: Either system clock or reference clock for GTHE1_QUAD_X1Y1_0 is not running: JTAG Scan Rate is set to "Manual Scan".

ug806_c1_12_062111



Starting the SuperClock-2 Module

The IBERT demonstration designs use an integrated ChipScope Pro software VIO core to control the clocks on the SuperClock-2 module. The SuperClock-2 module features two clock-source components: 1) An always-on Si570 crystal oscillator and, 2) an Si5386 jitter-attenuating clock multiplier. Outputs from either device can be used to drive the transceiver reference clocks. To start the SuperClock-2 Module:

1. In the Project Panel, double-click **VIO Console** (Figure 1-13).

Project: ml628_gth	
JTAG Chain	
- DEV:0 MyDevice0 (Syste	m_ACE_CF)
- DEV:1 MyDevice1 (XC6V	HX380T)
 System Monitor Con: 	sole
- UNIT:1_0 MyIBERT Y	/6 GTH1_0 (IBERT V6 GTH)
BERT Console	
P UNIT:0 SCLK2 Cont	rol (VIO)
VIO Console	
	UG806_c1_13_041411

Figure 1-13: Project Panel - VIO Console (GTH)

2. The clock sources on the SuperClock-2 module are controlled from the VIO Console. Click on the **Si5368 Start** button (Figure 1-14) to enable the clock output.

Note: The ROM address values for the Si5360 and Si570 devices (i.e., Si5368 ROM Addr and Si570 ROM Addr) are preset to 44 to produce an output frequency of 174.69 MHz. Entering a different ROM address changes the reference clock(s) frequency. The complete list of pre-programmed SuperClock-2 frequencies and their associated ROM addresses is provided in Table 1-2, page 31.

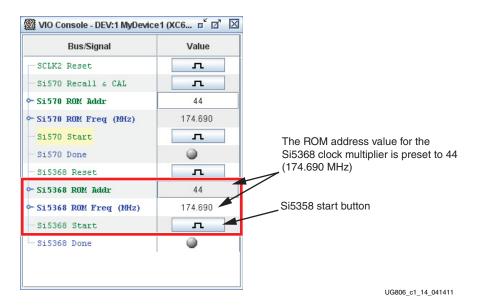


Figure 1-14: Si5368 Address, Frequency and Start Button

Note: Perform step 3 through step 8 as part of the GTH IBERT workaround described previously in this procedure (see notes included with step 4, page 13).

- 3. Close the ChipScope application by selecting **File** \rightarrow **Exit**. Do not save the changes when prompted. Do not power down the board.
- 4. Press and release the System ACE controller reset button (SW2) on the ML628 board to reload the GTH IBERT demonstration design.
- 5. Restart the ChipScope application and select **File** \rightarrow **Open** Project.
- 6. When the Open Project window appears, select ml628_gth.cpj and click **Open**.
- 7. Click the **Open Cable** button (Figure 1-15).

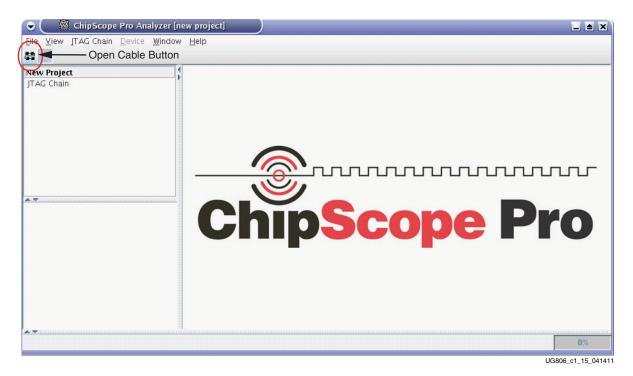


Figure 1-15: Open Cable Button

8. When the dialog appears asking to set up the core with the settings from the current project, click **Yes** (Figure 1-16).

Note: Restarting the SuperClock-2 module is not required. The Si5368 clock outputs will be enabled and running at the correct frequency.



Figure 1-16: IBERT V6GTH Project Settings

9. In the project panel, double-click **IBERT Console** (Figure 1-17).



Figure 1-17: Project Panel - IBERT Console (GTH)

10. At the top of the ChipScope Pro analyzer window, click the **Reset All** button (Figure 1-18).

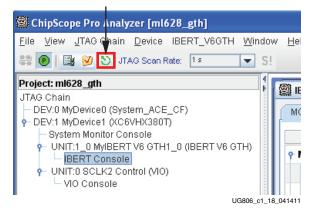


Figure 1-18: Reset All Button

11. When prompted "Are you sure you want to reset all Channels and Counters?" Click **Yes**.

Viewing GTH Transceiver Operation

After completing step 11 in Starting the SuperClock-2 Module, the IBERT demonstration is configured and running. The status and test settings are displayed on the **MGT/IBERT Settings** tab in the IBERT Console shown in Figure 1-19.

Note the line rate, TX differential output swing, and RX bit error count:

- The line rate for all four GTX transceivers is 11.18 Gps (see **MGT Link Status** in Figure 1-19).
- The GTX transmitter differential output swing is preset to 800 mV.
- Verify that there are no bit errors. If the count is not zero, see In Case of RX Bit Errors.

GT/BERT Settings DRP S	Settings Port Settings Swe	ep Test Settings		
	GTHE1_QUAD_X1Y1_0	GTHE1_QUAD_X1Y1_1	GTHE1_QUAD_X1Y1_2	GTHE1_QUAD_X1Y1_3
MGT Settings				
- DFETAPOVRD				
– MGT Alias	QUAD117_0	QUAD117_1	QUAD117_2	QUAD117_3
- DFETAP1	4	4	3	4
 Tile Location 	GTHE1_QUAD_X1Y1	GTHE1_QUAD_X1Y1	GTHE1_QUAD_X1Y1	GTHE1_QUAD_X1Y1
- DFETAP2	18	19	17	19
MGT Link Status	11.18 Gbps	11.18 Gbps	11.18 Gbps	11.18 Gbps
- DFETAP3	1	0	1	1
 PLL Status 	LOCKED	LOCKED	LOCKED	LOCKED
AGCOVRD				
– Loopback Mode	None	None 💌	None 💌	None
- AGC	0	0	0	1
- QUAD Reset	Reset	Reset	Reset	Reset
 TX Error Inject 	Inject	Inject	Inject	Inject
TX Diff Output Swing	800 mV 💌	800 mV 🗸 🔻	800 mV	800 mV
– TX Pre-Emphasis	1	1	1	1
– TX Post-Emphasis	10 💌	10 💌	10 💌	10 💌
RX Equalization	4.15 dB	4.15 dB	4.15 dB	4.15 dB
BERT Settings				
– TX Data Pattern	PRBS 31-bit	PRBS 31-bit 💌	PRBS 31-bit 💌	PRBS 31-bit
- RX Data Pattern	PRBS 31-bit	PRBS 31-bit 💌	PRBS 31-bit	PRBS 31-bit
 RX Bit Error Ratio 	1.087E-012	1.087E-012	1.087E-012	1.087E-012
 RX Received Bit Count 	9.202E011	9.202E011	9.202E011	9.202E011
RX Bit Error Count	0.000E000	0.000E000	0.000E000	0.000E000
- BERT Reset	Reset	Reset	Reset	Reset

ug806_c1_19_062111

Figure 1-19: GTH IBERT Console

In Case of RX Bit Errors

If after the **Reset All** button (Figure 1-18) has been pressed the **RX Bit Error Count** for one or more Quads displays a non-zero value, click the respective BERT **Reset** button (Figure 1-20) to zero the count.

- TX Data Pattern	PRBS 31-bit	PRBS 31-bit 💌	PRBS 31-bit 🗨	PRBS 31-bit
– RX Data Pattern	PRBS 31-bit	PRBS 31-bit	PRBS 31-bit	PRBS 31-bit
RX Bit Error Ratio	4.309E-010	1.657E-011	3.315E-011	1.657E-011
- RX Received Bit Count	6.034E010	0.0042010	6.034E010	6.034E010
- RX Bit Error Count	2.600E001	0.000E000	2.000E000	0.000E000
BERT Reset	Reset	Reset	Reset	Reset

UG806_c1_20_041411

Figure 1-20: Resetting the RX Bit Error Count

If RX bit errors persist after clicking the BERT **Reset** buttons, additional tuning of the transceivers might be required and/or one or more DC blocks might need to be installed in line with the loopback cables. See DC Blocks for installation instructions.

DC Blocks

The GTH receiver analog front end (AFE) does not support DC coupling (see UG371, *Virtex-6 FPGA GTH Transceivers User Guide* for details). For this reason, a DC component in the signal can result in bit errors being observed by the receiver. Sixteen in-line DC blocks are included with the ML628 board and are provided to filter the DC component in the signal path. To use the DC block in loopback, screw one end into to the SMA F-F adapter and then connect the RX and TX cables to each end of the union (Figure 1-21). Install a DC block to both the P and N signals of each lane where bit errors are observed.



Figure 1-21: Loopback Connection with DC Block

Closing the IBERT Demonstration

To stop the IBERT demonstration:

- 1. Close the ChipScope application by selecting $File \rightarrow Exit$. *Note:* Do not save changes to the project.
- 2. Place the main power switch SW1 in the off position.

More Information

Additional information on the ChipScope Pro software and GTH IBERT core can be found in:

- www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ chipscope_pro_sw_cores_ug029.pdf, UG029 - ChipScope Pro Software Cores
- <u>http://www.xilinx.com/support/documentation/ip_documentation/</u> <u>chipscope_ibert_virtex6_gth.pdf</u>, DS755 - *ChipScope Integrated Bit Error Ratio Test* (*IBERT*) for Virtex-6 GTH.

Running the GTX IBERT Demonstration

The GTX IBERT demonstration operates one GTX Quad at a time. This section describes how to test GTX Quad 100. The remaining GTX Quads are tested following a similar series of steps. The GTH IBERT demonstration is described in Running the GTH IBERT Demonstration, page 7.

Connecting the GTX Transceivers and Reference Clocks

Figure 1-22 shows the locations for GTX transceiver Quads 100, 101, 102, 103, 104, 105, 112, 113, 114, and 115 on the ML628 board.

Note: Figure 1-22 is for reference only and might not reflect the current revision of the board.

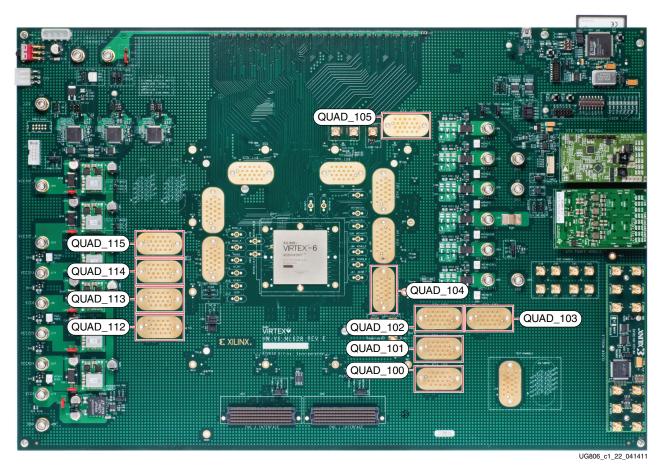


Figure 1-22: GTX Quad Locations

All GTX transceiver pins and reference clock pins are routed from the FPGA to a connector pad which interfaces with Samtec BullsEye connectors. Figure 1-23 A shows the connector pad. Figure 1-23 B shows the connector pinout.

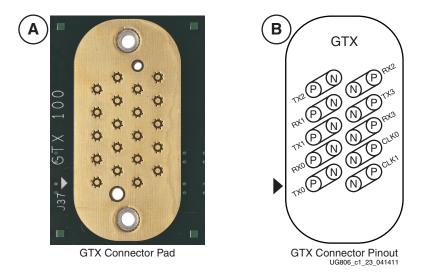


Figure 1-23: A – GTX Connector Pad. B – GTX Connector Pinout

www.xilinx.com

The SuperClock-2 module provides LVDS clock outputs for the GTX and GTH transceiver reference clocks in the IBERT demonstrations. Figure 1-3, page 9 shows the locations of the differential clock SMA connectors on the clock module which can be connected to the reference clock cables. The four SMA pairs labeled CLKOUT provide LVDS clock outputs from the Si5368 clock multiplier/jitter attenuator device on the clock module. The SMA pair labeled Si570_CLK provides LVPECL clock output from the Si570 programmable oscillator on the clock module. For the GTX IBERT demonstration, the output clock frequencies are preset to 162.500 MHz. For more information regarding the SuperClock-2 module, refer to UG770, HW-CLK-101-SCLK2 SuperClock-2 Module User Guide.

Attach the GTX Quad Connector

Attach the Samtec BullsEye connector to GTX Quad 100 (Figure 1-24), aligning the two indexing pins on the bottom of the connector with the guide holes on the board. Hold the connector flush with the board and fasten it by tightening the two captive screws.



Figure 1-24: BullsEye Connector Attached to Quad 100

GTX Transceiver Clock Connections

Refer to Figure 1-22, page 21 to identify the P and N coax cables that are connected to the two reference clock inputs (CLK0 and CLK1). Connect these cables to the SuperClock-2 Module as follows:

- CLK0_P coax cable \rightarrow SMA connector J5 (CLKOUT1_P) on the SuperClock-2 Module
- CLK0_N coax cable \rightarrow SMA connector J6 (CLKOUT1_N) on the SuperClock-2 Module
- CLK1_P coax cable \rightarrow SMA connector J7 (CLKOUT2_P) on the SuperClock-2 Module
- CLK1_N coax cable \rightarrow SMA connector J8 (CLKOUT2_N) on the SuperClock-2 Module

Note: Any one of the five differential outputs from the SuperClock-2 Module can be used to source the GTX reference clock. CLKOUT1_P, CLKOUT1_N, CLKOUT2_P, and CLKOUT2_N are used here as an example.

GTX TX/RX Loopback Connections

Refer to Figure 1-22, page 21 to identify the P and N coax cables that are connected to the four receivers (RX0, RX1, RX2 and RX3) and the four transmitters (TX0, TX1, TX2 and TX3). Use eight SMA female-to-female (F-F) adapters (Figure 1-25), to connect the transmit and receive cables as shown in Figure 1-26 and detailed below:

- $TX0_P \rightarrow SMA F-F Adapter \rightarrow RX0_P$
- $TX0_N \rightarrow SMA F-F Adapter \rightarrow RX0_N$
- $TX1_P \rightarrow SMA F-F Adapter \rightarrow RX1_P$

- TX1_N \rightarrow SMA F-F Adapter \rightarrow RX1_N
- $TX2_P \rightarrow SMA F-F Adapter \rightarrow RX2_P$
- $TX2_N \rightarrow SMA F-F Adapter \rightarrow RX2_N$
- $TX3_P \rightarrow SMA F-F Adapter \rightarrow RX3_P$
- TX3_N \rightarrow SMA F-F Adapter \rightarrow RX3_N

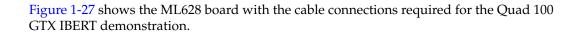


Figure 1-25: SMA F-F Adapter



UG806_c1_26_041411

Figure 1-26: TX-To-RX Loopback Connection Example



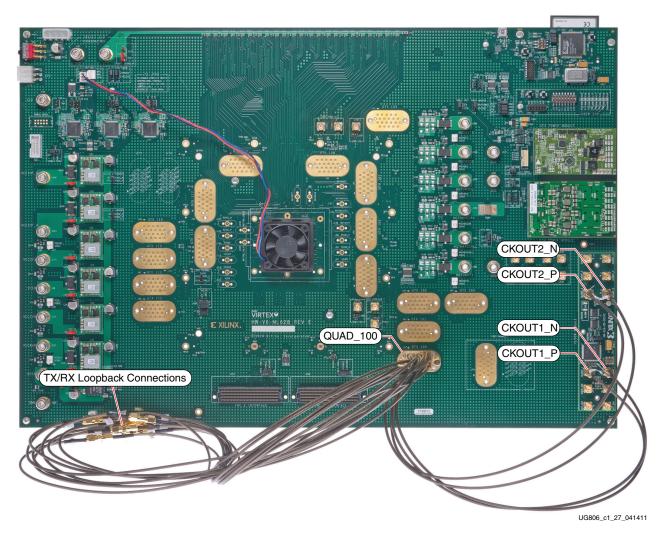


Figure 1-27: Cable Connections for Quad 100 GTX IBERT Demonstration

Configuring the FPGA

This section describes how to configure the FPGA using the CF cards included with the board. The FPGA can also be configured through ChipScope Pro or iMPACT software using the .bit files which are available online (as collection rdf0116_13-2.zip) at:

http://www.xilinx.com/products/boards/ml628/reference_designs.htm

To configure from the CF card:

- 1. Insert the CF card labeled *IBERT* #2 into the CF card reader slot located on the bottom-side (upper-right corner) of the board.
- 2. Plug the 12V output from the power adapter into connector J122.
- 3. Connect the programming cable to the host computer. Any of these cables can be used:
 - Platform Cable USB-II (DLC10)
 - Platform Cable USB (DLC9, DLC9G or DLC9LP)

Parallel Cable IV (PC4)

Connect the ribbon cable between the programming cable and the ML628 board at JTAG connector J1.

4. Select the GTX IBERT demonstration with the System Ace controller CFG ADDRESS switch, SW3. The setting on this 3-bit DIP switch (Figure 1-28) selects the file used to configure the FPGA. A switch is in the ON position if it switched towards the ON text printed on the DIP; otherwise, it is OFF. For the Quad 100 GTH IBERT demonstration, set: ADR2 = ON, ADR1 = ON, and ADR0 = OFF.

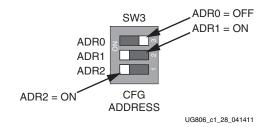


Figure 1-28: Configuration Address DIP Switch (SW3)

There is one IBERT demonstration design for each GTX and GTH Quad on the ML628 board, for a total of 16 designs. The designs are organized and stored on the two CF cards (*IBERT #1* and *IBERT #2*) as shown in Table 1-1, page 12.

5. Place the main power switch SW1 to the ON position.

Setting Up the ChipScope Pro Software

- Start the ChipScope Pro analyzer tools on the host computer and select File → Open Project.
- 2. When the Project window opens, navigate to the directory where the ChipScope software project files (.cpj) were extracted. Select ml628 gtx.cpj and click **Open**.

Note: The .cpj file loads pre-saved project settings for the demonstration including MGT/IBERT and clock module control parameters. For more information regarding MGT/IBERT settings, refer to www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ www.support/documentation/sw_manuals/xilinx13_2/ www.support/documentation/sw_manuals/xilinx13_2/ www.support/documentation/sw_manuals/xilinx13_2/ www.support/documentation/sw_manuals/xilinx13_2/ <a href="http://www.support.com/support/documentation/sw_manuals/support/documentation/sw_manuals/support/support.com/support

3. Click the **Open Cable** button (Figure 1-29).

🕞 🤇 📓 ChipScope Pro Analyzer [ne	w project]			
Eile View ITAG Chain Device Window	Help			
Copen Cable Button				
Néw Project JTAG Chain				
		21		
	Ch	IPS (cope	Pro
				UG806 c1 29 04141

Figure 1-29: Open Cable Button

4. When the dialog appears asking to set up the core with the settings from the current project, click **Yes** (Figure 1-30).

ChipSco	ope Pro Analyzer - IBert V6GTX Project Settings 🛛 🛛 🔀
?	Project settings do not match current core! Do you want to set up the IBERT V6GTX core with settings from the current project?
	ug806_c1_30_041411



Starting the SuperClock-2 Module

The IBERT demonstration designs use an integrated ChipScope Pro software VIO core to control the clocks on the SuperClock-2 module. The SuperClock-2 module features two clock-source components: 1) An always-on Si570 crystal oscillator and, 2) an Si5386 jitter-attenuating clock multiplier. Outputs from either device can be used to drive the transceiver reference clocks. To start the SuperClock-2 Module:

1. In the Project Panel, double-click **VIO Console** (Figure 1-31).

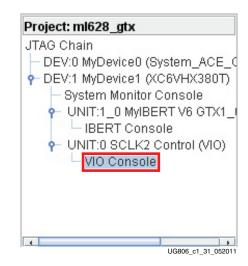


Figure 1-31: Project Panel - VIO Console (GTX)

2. The clock sources on the SuperClock-2 module are controlled from the VIO Console. Click on the **Si5368 Start** button (Figure 1-32) to enable the clock output.

Note: The ROM address values for the Si5360 and Si570 devices (i.e., Si5368 ROM Addr and Si570 ROM Addr) are preset to 2 to produce an output frequency of 162.500 MHz. Entering a different ROM address changes the reference clock(s) frequency. The complete list of pre-programmed SuperClock-2 frequencies and their associated ROM addresses is provided in Table 1-2, page 31.

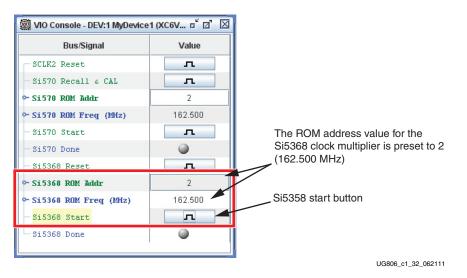


Figure 1-32: Si5368 Address, Frequency and Start Button

3. In the project panel, double-click IBERT Console (Figure 1-33).

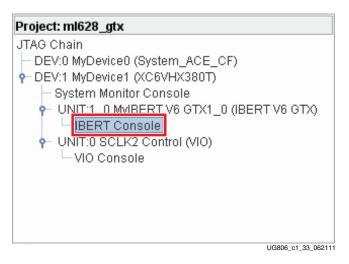


Figure 1-33: Project Panel - IBERT Console (GTX)

4. At the top of the ChipScope Pro analyzer window, click the **Reset All** button (Figure 1-34).

🗟 ChipScope Pro A	nalyzer [ml628_gtx]
Eile View JTAG 🖣	nain <u>D</u> evice IBERT <u>V</u> 6GTX <u>W</u> indow <u>H</u> elp JTAG Scan Rate: 1≤ ▼ S!
Project: ml628_gtx JTAG Chain — DEV:0 MyDevice0	

Figure 1-34: Reset All Button

5. When prompted "Are you sure you want to reset all Channels and Counters?" Click **Yes**.

Viewing GTX Transceiver Operation

After completing step 5 in Starting the SuperClock-2 Module, the IBERT demonstration is configured and running. The status and test settings are displayed on the **MGT/IBERT Settings** tab in the IBERT Console shown in Figure 1-35.

Note the line rate, TX differential output swing, and RX bit error count:

- The line rate for all four GTX transceivers is 6.5 Gps (see **MGT Link Status** in Figure 1-35).
- The GTX transmitter differential output swing is preset to 590 mV.
- Verify that there are no bit errors.

IGT/BERT Settings DRP	Settings Port Settings	Swee	ep Test Settings						
	GTX_X0Y0		GTX_X0Y1		GTX_X0Y2		GTX_X0Y3		
MGT Settings									
- TX Pre-Emphasis	TX Pre-Emphasis 0.400 dB (0011)		0.400 dB (0011)		0.400 dB (0011)		0.400 dB (0011)		
- DFETAPOVRD	~		V		v		~		
- MGT Alias	GTX0_0	GTX0_0		GTX1_0		GTX2_0			
– TX Post-Emphasis	0.000 dB (00000)	•	0.000 dB (00000)		0.000 dB (00000)		0.000 dB (00000)		
- DFETAP1	0	-	0		0		0		
- Tile Location	GTX_X0Y0		GTX_X0Y1		GTX_X0Y2		GTX_X0Y3		
- RX Polarity Invert					_				
- DFETAP2	0	-	0		0		0		
- MGT Link Status	6.5 Gbps		6.5 Gbps		6.5 Gbps		6.5 Gbps		
- RX AC Coupling Enable	2		2		v		V		
- MGT Edit Line Rate	6.5 Gbps		6.5 Gbps		6.5 Gbps		6.5 Gbps		
- RX Termination Voltage	MGTAVTT *	-	MGTAVTT *	-	MGTAVTT *	-	MGTAVTT *		
- TX PLL Status	LOCKED		LOCKED		LOCKED		LOCKED		
 RX Equalization 	6	-	6	-	6	-	6		
- RX PLL Status	LOCKED		LOCKED		LOCKED		LOCKED		
- DFEEYEDACMON	51.6 mV		64.5 mV		77.4 mV		71.0 mV		
- Loopback Mode	None	-	None	•	None	-	None		
– Channel Reset	Reset		Reset		Reset		Reset		
- TX Polarity Invert									
- TX Error Inject	Inject		Inject		Inject		Inject		
- TX Diff Output Swing	590 mV (0110)	•	590 mV (0110)	•	590 mV (0110)	-	590 mV (0110)		
- DFETAP3	0	•	0	•	0	-	0		
- DFETAP4	0	-	0	-	0	-	0		
- RX Sampling Point	760	.598 UI	76	0.598 UI	76	0.598 UI	76	0.59	
BERT Settings									
– TX Data Pattern	PRBS 31-bit	-	PRBS 31-bit	-	PRBS 31-bit	-	PRBS 31-bit		
- RX Data Pattern	PRBS 31-bit	-	PRBS 31-bit	-	PRBS 31-bit	-	PRBS 31-bit		
 RX Bit Error Ratio 	1.829E-012		1.833E-012		1.837E-012		1.841E-012		
- RX Received Bit Count	5.467E011		5.455E011		5.442E011		5.432E011		
- RX Bit Error Count	0.000E000		0.000E000		0.000E000		0.000E000		
BERT Reset		Reset		Reset		Reset			

ug806_c1_35_041411



Additional information on the ChipScope Pro software and IBERT core can be found in:

- <a>www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ chipscope_pro_sw_cores_ug029.pdf, UG029 - ChipScope Pro Software Cores
- <u>http://www.xilinx.com/support/documentation/ip_documentation/</u> <u>chipscope_ibert_virtex6_gth.pdf</u>, DS755 - *ChipScope Integrated Bit Error Ratio Test* (*IBERT*) for Virtex-6 GTH.

Closing the IBERT Demonstration

To stop the IBERT demonstration:

- 1. Close the ChipScope application by selecting $File \rightarrow Exit$. *Note:* Do not save changes to the project.
- 2. Place the main power switch SW1 in the off position.

More Information

Additional information on the ChipScope Pro software and GTH IBERT core can be found in:

- <a>www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ chipscope_pro_sw_cores_ug029.pdf, UG029 - ChipScope Pro Software Cores
- <u>http://www.xilinx.com/support/documentation/ip_documentation/</u> <u>chipscope_ibert_virtex6_gtx.pdf</u>, DS732 - *ChipScope Integrated Bit Error Ratio Test* (*IBERT*) for Virtex-6 GTX.

SuperClock-2 Frequency Table

Table 1-2 lists the addresses for the frequencies that are programmed into the SuperClock-2 read-only-memory (ROM).

Table 1-2: Si570 and Si5368 Frequency Table

Address	Protocol	Frequency (MHz)	Address	Protocol	Frequency (MHz)	Address	Protocol	Frequency (MHz)
0	100GE/40GE/ 10GE	161.130	30	OBSAI	307.200	60	XAUI	156.250
1	Aurora	81.250	31	OBSAI	614.400	61	XAUI	312.500
2	Aurora	162.500	32	OC-48	19.440	62	XAUI	625.000
3	Aurora	325.000	33	OC-48	77.760	63	Generic	66.667
4	Aurora	650.000	34	OC-48	155.520	64	Generic	133.333
5	CE111	173.370	35	OC-48	311.040	65	Generic	166.667
6	CPRI	61.440	36	OC-48	622.080	66	Generic	266.667
7	CPRI	122.880	37	OTU-1	166.629	67	Generic	333.333
8	CPRI	153.630	38	OTU-1	333.257	68	Generic	533.333
9	CPRI	245.760	39	OTU-1	666.514	69	Generic	644.000
10	CPRI	491.520	40	OTU-1	666.750	70	Generic	666.667
11	Display Port	67.500	41	OTU-2	167.330	71	Generic	205.000
12	Display Port	81.000	42	OTU-2	669.310	72	Generic	210.000
13	Display Port	135.000	43	OTU-3	168.050	73	Generic	215.000
14	Display Port	162.000	44	OTU-4	174.690	74	Generic	220.000
15	Fibrechannel	106.250	45	PCIe	100.000	75	Generic	225.000
16	Fibrechannel	212.500	46	PCIe	125.000	76	Generic	230.000
17	Fibrechannel	425.000	47	PCIe	250.000	77	Generic	235.000
18	GigE	62.500	48	SATA	75.000	78	Generic	240.000
19	GigE	125.000	49	SATA	150.000	79	Generic	245.000
20	GigE	250.000	50	SATA	300.000	80	Generic	250.000
21	GigE	500.000	51	SATA	600.000	81	Generic	255.000
22	GPON	187.500	52	SDI	74.250	82	Generic	260.000
23	Interlaken	132.813	53	SDI	148.500	83	Generic	265.000
24	Interlaken	195.313	54	SDI	297.000	84	Generic	270.000
25	Interlaken	265.625	55	SDI	594.000	85	Generic	275.000
26	Interlaken	390.625	56	SMPTE435M	167.063	86	Generic	280.000
27	Interlaken	531.250	57	SMPTE435M	334.125	87	Generic	285.000
28	OBSAI	76.800	58	SMPTE435M	668.250	88	Generic	290.000
29	OBSAI	153.600	59	XAUI	78.125	89	Generic	295.000

www.xilinx.com

Address	Protocol	Frequency (MHz)	Address	Protocol	Frequency (MHz)	Address	Protocol	Frequency (MHz)
90	Generic	300.000	103	Generic	365.000	116	Generic	430.000
91	Generic	305.000	104	Generic	370.000	117	Generic	435.000
92	Generic	310.000	105	Generic	375.000	118	Generic	440.000
93	Generic	315.000	106	Generic	380.000	119	Generic	445.000
94	Generic	320.000	107	Generic	385.000	120	Generic	450.000
95	Generic	325.000	108	Generic	390.000	121	Generic	455.000
96	Generic	330.000	109	Generic	395.000	122	Generic	460.000
97	Generic	335.000	110	Generic	400.000	123	Generic	465.000
98	Generic	340.000	111	Generic	405.000	124	Generic	470.000
99	Generic	345.000	112	Generic	410.000	125	Generic	475.000
100	Generic	350.000	113	Generic	415.000	126	Generic	480.000
101	Generic	355.000	114	Generic	420.000	127	Generic	485.000
102	Generic	360.000	115	Generic	425.000			

Table 1-2: Si570 and Si5368 Frequency Table (Cont'd)

IBERT Demonstration Designs

This section provides an overview of the source files used to generate the IBERT demonstrations provided with ML628 board.

To rebuild the designs shown here, you must have an installation of ISE Design Suite version 13.2 or higher.

Source Directories and Files

The file rdf0116_13-2.zip contains the source files for 16 individual designs (one for each GTH and GTX Quad on the ML628 board). The .zip file is located at:

http://www.xilinx.com/products/boards/ml628/reference_designs.htm.

Each design is saved in a separate directory:

ML628_gth_q106/	ML628_gtx_q100/
ML628_gth_q107/	ML628_gtx_q101/
ML628_gth_q108/	ML628_gtx_q102/
ML628_gth_q116/	ML628_gtx_q103/
ML628_gth_q117/	ML628_gtx_q104/
ML628_gth_q118/	ML628_gtx_q105/
	ML628_gtx_q112/
	ML628_gtx_q113/
	ML628_gtx_q114/
	ML628_gtx_q115/

The directory structures for the GTH and GTX designs are identical. For example:

```
ML628_gth_q117/
 par/
   example_ibert_v6_q117_top.bit
   example implement ibert v6 q117.prj
   example_implement_ibert_v6_q117.xst
   i2c_sclk2_control.ngc
   ibert_v6_q117.ngc
   icon_v6_1.ngc
   implement.bat
   vio v6 si84 so78.ngc
  src/
   chipscope.v
   example_ibert_v6_q117.v
   i2c_sclk2_control_bb.v
   ibert_v6_q117_top.ucf
   vio sclk2 control.v
```

Has the same structure as:

```
ML628_gtx_q100/
 par/
   example_ibert_v6_q100_top.bit
   example implement ibert v6 q100.prj
   example_implement_ibert_v6_q100.xst
   i2c sclk2 control.ngc
   ibert_v6_q100.ngc
   icon_v6_1.ngc
   implement.bat
   vio_v6_si84_so78.ngc
 src/
   chipscope.v
   example ibert v6 q100.v
   i2c_sclk2_control_bb.v
   ibert v6 q100 top.ucf
   vio_sclk2_control.v
```

The par directory contains the project, input and pre-compiled .ngc files required to build the demonstration. The .bit configuration file is also in the par directory.

IBERT Design Files

The IBERT design files are described in this section.

example_ibert_v6_q1xx_top.bit

The example_ibert_v6_q1xx_top.bit file is the FPGA bitstream (configuration) file for the IBERT demonstration. This file can be used to program the FPGA directly using ChipScope or iMPACT and a JTAG download cable.

example_implement_ibert_v6_q1xx.prj

The example_implement_ibert_v6_q1xx.prj project file is used with the Xilinx Synthesis Technology (xst) synthesis application to provide a list of files associated with the design. The .prj file contains the language, library name (e.g., "work") and the design files. This .prj file is referenced in the .xst file. For additional details on this file, see:

• <u>http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/xst.pdf</u>, UG627 - XST User Guide for Virtex-4, Virtex-5, Spartan-3, and Newer CPLD Devices

example_implement_ibert_v6_q1xx.xst

The example_implement_ibert_v6_q1xx.xst file contains the arguments that are passed to the xst synthesis application when the application is run in command line (i.e., script) mode.

For details on the arguments used in this file, see:

• <u>http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/xst.pdf</u>, UG627 - XST User Guide for Virtex-4, Virtex-5, Spartan-3, and Newer CPLD Devices

i2c_sclk2_control.ngc

The i2c_sclk2_control.ngc file is a binary implementation netlist file containing the logic and constraints required for the FPGA to control the SuperClock-2 module over an I²C interface. The SuperClock-2 frequency table is also contained in this module. A black box interface (i2c_sclk2_control_bb.v) is provided for synthesis, but the underlying HDL source code is not provided in the design package.

ibert_v6_q1xx.ngc

The ibert_v6_q1xx.ngc file is a binary implementation netlist file containing the logic and constraints required to implement the ChipScope IBERT core in an FPGA. ibert_v6_q1xx.ngc is created using the ISE Design Suite CORE GeneratorTM. In the ML628 IBERT demonstration designs, each IBERT core is configured to support a single GTH or GTX Quad. For example, ibert_v6_q117.ngc is the GTH IBERT core for Quad 117. The flows for building the GTH and GTX IBERT cores for the demonstration designs are provided in Creating the GTH IBERT Core, page 36 and Creating the GTX IBERT Core, page 43

For additional information on ChipScope IBERT cores, refer to following documents:

- <u>www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/</u> <u>chipscope_pro_sw_cores_ug029.pdf</u>, UG029 - ChipScope Pro Software Cores
- <u>http://www.xilinx.com/support/documentation/ip_documentation/</u> <u>chipscope_ibert_virtex6_gth.pdf</u>, DS755 - *ChipScope Integrated Bit Error Ratio Test* (*IBERT*) for Virtex-6 GTH.
- <u>http://www.xilinx.com/support/documentation/ip_documentation/</u> <u>chipscope_ibert_virtex6_gtx.pdf</u>, DS732 - ChipScope Integrated Bit Error Ratio Test (IBERT) for Virtex-6 GTX.

icon_v6_1.ngc

The icon_v6_1.ngc file is a binary implementation netlist file containing the logic and constraints required to implement the ChipScope Integrated Control (ICON) core in an FPGA. This file is created using the ISE Design Suite CORE Generator. In the ML628 IBERT demonstration designs, the ICON core is configured with a single control port which connects to the SuperClock-2 VIO core vio_v6_si84_so78.ngc. The ICON core is required to control the SuperClock-2 VIO core from the ChipScope software.

For additional details on the ChipScope ICON core, refer to:

 www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ chipscope_pro_sw_cores_ug029.pdf, UG029 - ChipScope Pro Software Cores

vio_v6_si84_so78.ngc

The vio_v6_si84_so78.ngc file is a binary implementation netlist file containing the logic and constraints required to implement the ChipScope Virtual Input/Output (VIO) core in an FPGA. vio_v6_si84_so78.ngc is created using the ISE Design Suite CORE Generator. In the ML628 IBERT demonstration designs, a VIO core is required to control and receive status from SuperClock-2. The VIO core is configured with 84 synchronous inputs and 78 synchronous outputs, and communicates with ChipScope through an ICON core (icon_v6_1.ngc).

For additional details on the ChipScope VIO core, refer to:

 www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/ chipscope_pro_sw_cores_ug029.pdf, UG029 - ChipScope Pro Software Cores

implement.bat

The implement.bat file is a Windows (DOS) batch file used to build the IBERT demonstration design. To use this batch, open a DOS shell:

- 1. From the Windows desktop, click the Start button and select Run...
- 2. When the Run dialog box appears, enter **cmd** in the **Open** field.
- 3. Click **OK** to open the DOS shell.

From the DOS shell, navigate to the directory containing the implement.bat file. To run the build, pass the Quad number as an argument to the batch. For example, to build the IBERT demonstration in the ML628_gth_q117 directory, enter **implement 117** on the command line.

The batch file checks that the IBERT core (e.g., ibert_v6_q117.ngc) is present in the current directory before starting the build. The batch file also creates a results directory where, upon successful completion of the build, the .bit file is placed.

The src directory contains the Verilog HDL source code for the design. An example of the design hierarchy is shown here:

```
example_ibert_v6_q117.v
icon_v6_1.ngc
vio_sclk2_control.v
i2c_sclk2_control_bb.v
```

The following is a description of each source file:

example_ibert_v6_q1xx.v

The example_ibert_v6_q1xx.v file is the top-level source file for the IBERT demonstration. example_ibert_v6_q1xx.v is a modified version of the example code that is automatically created by the CORE Generator when the IBERT core is generated (see Creating the GTH IBERT Core, page 36 and Creating the GTX IBERT Core, page 43). All of the top-level modules and cores for the design are instantiated in this file.

i2c_sclk2_control_bb.v

The i2c_sclk2_control_bb.v file is the "black box" definition for the SuperClock-2 control module. It is a black box because it provides only an interface to i2c_sclk2_control.ngc, but no source code is present at this level. This module is instantiated in vio_sclk2_control.v.

ibert_v6_qxxx_top.ucf

The ibert_v6_qxxx_top.ucf file is the user constraints file (UCF) for the demonstration. ibert_v6_qxxx_top.ucf is the example .ucf created by the CORE Generator during the IBERT core generation, but is modified to include the system clock, I²C and SuperClock-2 control pin mapping.

For additional details on the user constrains file, refer to:

http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_2/cgd.pdf, UG625 - Constraints Guide.

vio_sclk2_control.v

The vio_sclk2_control.v file provides the interface between the ChipScope Virtual IO (VIO) and the SuperClock-2 control module, i2c_sclk2_control.ngc. For this reason, the ChipScope VIO core (vio_v6_si84_so78.ngc) and SuperClock-2 control module (i2c_sclk2_control.ngc) are instantiated here.

Creating the GTH IBERT Core

This section provides a procedure to create a single Quad GTH IBERT core using CORE Generator software. The procedure assumes Quad 117 and the OTU4 protocol (11.18 Gb/s line rate), but cores for any of the GTH Quads with any supported line rate can be created following the same series of steps.

For more details on generating IBERT cores, refer to <u>www.xilinx.com/support/</u> <u>documentation/sw_manuals/xilinx13_2/chipscope_pro_sw_cores_ug029.pdf</u>, UG029 - *ChipScope Pro Software Cores*

- 1. Start the CORE Generator tool from either the ISE Project Navigator window or a command line:
 - From the Project Navigator window, select: **Tools** → **Core Generator...**
 - From a command line, enter: coregen
- 2. In the Core Generator window, click the New Project icon (highlighted in Figure 1-36).

Eile	¥iew <u>H</u> elp							
	🆻 🗟 🔲 Core g	Generator Help	N2					
INT SEASE	P Catalog View by Function A Name Automotive & In Basic Elements Communication of Debug & Verifice Digital Signal Pro Example Cores Communication of Communication of Communication of Debug & Verifice Communication of Communication o	& Networking ation ocessing	₽ × Version AXI4 Status License	LogiCRE There is no You may browse to cores until you op	Xilinx C project open. the IP Catalog but you ten or create a project.	will not be	 	
	Math Functions Memories & Store Standard Bus Int P Test Video & Image F	rage Elements terfaces		Copyright (c) 1995-2	2010 Xilinx, Inc. All right	ts reserved.		
				Console				đ
				Welcome to Xilinx C Help system initializ				
	۱		•					
S	Search IP Catalog:		Clear	Search Console		Find	<u>S</u> ave	<u>C</u> lear
	All IP versions	_	y IP compatible with chosen part	Information	Warnings S Errors			

Figure 1-36: Open New Project Icon, CORE Generator Project Window

- 3. When the the New Project dialog window opens (not shown), name the project and click **Save**.
- 4. In the Project Options window, click on **Part** and select the parameters listed here:
 - Family: Virtex6
 - Device: **xc6vhx380t**
 - Package: **ff1923**
 - Speed Grade: -2

Figure 1-37 shows the correct settings.

Project Options			? 🔀
Part Generation Advanced	Part Select the part for Family Device	your project: Virtex6 xc6vhx380t	~
	Package Speed Grade	ff1923 -2 K Cancel Apply	
			UG806 c1 37

Figure 1-37: CORE Generator Project Options (Part Options)

- 5. Next in the Project Options window, click on **Generation** and select the parameters listed here:
 - Design Entry: Verilog
 - Vendor: Other
 - Netlist Bus Format: **B<n:m>**
 - Preferred Simulation Model: Structural
 - ASY Symbol File: unchecked

Figure 1-38 shows the correct settings.

🍕 Project Options		? 🔀
Generation Auvanceu	Flow Design Entry Custom Output Products Please refer to the online help for in models using compxlib and using .VB	Verilog formation about compiling behavioral EO (Verilog) templates.
	- Flow Settings Vendor Netlist Bus Format	Other V B <n:m></n:m>
	Simulation Files Preferred Simulation Model Behavioral Structural None Other Output Products ASY Symbol File	Preferred Language VHDL Verilog
	OK Car	ncel Apply Help

UG806_c1_38_041411

Figure 1-38: CORE Generator Project Options (Generation Options)

- 6. Click **OK** to close the Project Options window.
- 7. In the IP Catalog pane of the CORE Generator window (Figure 1-39) select:
 Debug & Verification →

 $\textbf{ChipScope Pro} \rightarrow \textbf{}$

IBERT Virtex6 GTH (ChipScope Pro - IBERT) 2.03.a

Project View Manage	IP Help								
2 🗐 🗐 🖓 🔊	CORE Generator Help 🕅	94							
IP Catalog			8 ×						
View by Function View	by Name			PE	TB	FRTV	rtex6 G	TH	0
Name	A 1	Version A	XI4 AXI4-Stre	LogiCKRE	(Show Prot
Automotive & Indus		ICI SIGIT H	ATT HATT SUC		(Chip	Scope	Pro - I	BERI)	
BaseIP	circa:							-	
🗄 岁 Basic Elements				This core is supp	orted at status P	roduction by	your chosen pa	rt.	
😨 📂 Communication & Ne									
😑 📂 Debug & Verification	(Informati					
😑 📂 ChipScope Pro				Core type:		GTH (ChipScop	e Pro - IBERT)		
		1.04.a 2.01.a		Version: Core Summary:	2.03.a	Due Caules Tak	egrated Bit Error	Datie Tester 6	
		2.01.a 2.00.a		Cure Summary:	Virtex6 GTH.	Pro Series Inc	egrateu bit Error	Ratio lester in	Jr.
		2 02 a		1111					
IBERT Virte	x6 GTH (ChipScope Pro - IBERT) 2	2.03.a		Supported Fa	amilies				
VIBER I VITE	x6 GTX (Chip5cope Pro - IBERT) 2	2.05.a		Current Proj	act Options				
		1.05.a							
		1.00.a	~	Actions					
< TLA (ChinSc	one Pro - Integrated Logic Analyzer) 1	1 04 a	>			10021 2044			
Search IP Catalog:			Clear	The following act		e for this core:			
				Customize ar	id Generate	*****			
All IP versions	Only I	P compatible	e with chosen part	Console					
Project IP			8×	Help system initial					
Instance Name 👻	Core Name	Version	Last Generat 🛆	The IP Catalog ha					
Vibert_v6_q100 IBER	RT Virtex6 GTX (ChipScope Pro - IBERT)	2.05.a	2011-02-25 at	CoreGen has not CoreGen has been					
📢 ibert_v6_q101 IBEF	RT Virtex6 GTX (ChipScope Pro - IBERT)	2.05.a	2011-02-25 at	- 'Y:\O.40e\rtf\cc			Amix repositorie	51	
	RT Virtex6 GTX (ChipScope Pro - IBERT)		2011-02-25 at	The IP Catalog ha	s been reloaded				
	RT Virtex6 GTX (ChipScope Pro - IBERT)		2011-02-25 at	Opening project f	ile C:\data\ML62	8_Production\i	se\ibert_demo\c	pregen\ml628_	ibert.cgp.
	RT Virtex6 GTX (ChipScope Pro - IBERT) PT Virtex6 GTV (ChipScope Pro - IBERT)	2.05.a	2011-02-25 at	-					
<		. 115 8	2	Search Console			Find	Save	Cle
Search Project IP:			Clear	Information	Narnings	8 Errors			
						1			

Figure 1-39: Select IBERT Core

- 8. Click on the Customize and Generate link under the Actions heading (Figure 1-39)
- 9. After a few seconds page 1 of the IP customization window will appear. Enter the information shown here and in Figure 1-40, then click Next:
 - Component Name: ibert_v6_q117
 - Generate Bitstream: unchecked
 - Add RXUSERCLK probe: unchecked
 - GTH Naming Style: MGT m n
 - MGT Column: Right
 - Frequency: **25** MHz
 - Pin Location: UNASSIGNED
 - Pin Input Standard: LVCMOS25

lodi	8×		TREPT	COTU	
	<u>^</u>	LOGICERE	IBERT Vir	tex6 GTH	
IBERT_SYSCLOCK		109.01	ChipScope P	ro - IBERT)	2.0
Q0 REFCLK IN				···· ·	2.0
X0 Y0_GTH0_RX_IN		Component Name	ibert v6_q11	7	
X0Y0_GTH1_RX_IN		System Design			
	→X0Y0_GTH1_USERCLK_OL	System Design			
X0Y0_6TH2_RX_IN		🔲 Generate Bitstrea	am		
		Add RXUSERCLK	probe		
X0Y0_GTH3_RX_IN	→X0Y0_GTH3_TX_OUT	GTH Naming Style	MGTm n 🗸 e	x. MGT0_113 / MGTREFCLK_113	
	→X0Y0_GTH3_USERCLK_OL	a construction of the second			
Q1_REFCLK_IN		MGT Column	Right 🗸 M	1GT0_116-MGT3_118	
X0Y1_GTH0_RX_IN	→X0Y1_GTH0_TX_OUT	System Clock			
	→ X0Y1_GTHD_USERCLK_OL				
X0Y1_GTH1_RK_IN	→X0Y1_GTH1_TX_OUT	Frequency	25	MHz	
	→X0Y1_GTH1_USERCLK_OL	Pin Location	UNASSIGNED		
X0Y1_GTH2_RX_IN>	→X0Y1_GTH2_TX_OUT				
	→X0Y1_GTH2_USERCLK_OL	Pin Input Standard	LVCM0525	~	
X0Y1_GTH3_RX_IN>	→X0Y1_GTH3_TX_OUT				
	>X0Y1_GTH3_USERCLK_OL				
Q2_REFCLK_IN>					
X0Y2_GTH0_RX_IN	→ X0 Y2_GTH0_TX_OUT				
	→X0Y2_GTH0_USERCLK_OL				
X0Y2_GTH1_RX_IN	→X0Y2_GTH1_TX_OUT				
X0Y2_0TH2_RX_IN	→X0Y2_GTH2_TX_OUT				
	> XD Y2_GTH2_USERCLK_OL				
X0Y2_GTH3_RX_IN	→X0Y2_GTH3_TX_OUT				
	→X0Y2_GTH3_USERCLK_OL				
I_REFCLK_IN_RIGHT					
X1Y0_GTH0_RX_IN	>X1Y0_GTH0_TX_OUT				

Figure 1-40: CORE Generator - IBERT GTH Customization - Page 1

- 10. Enter the information shown here and in Figure 1-41, then click **Next**:
 - Protocol: OTU4
 - Refclk (MHz): **174.69**
 - GTH count: **4**

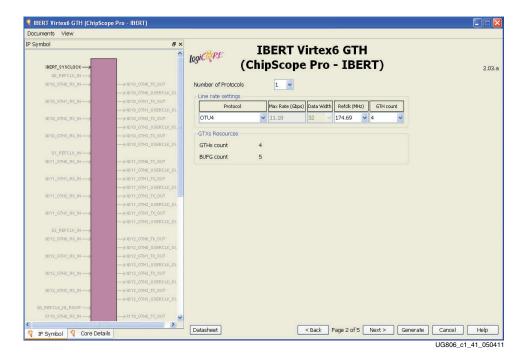


Figure 1-41: CORE Generator - IBERT GTH Customization - Page 2

- 11. Enter the information shown here and in Figure 1-42, then click **Next**:
 - Select Protocol for Quad117: OTU4 / 11.18 Gps

mbol	đ ×		ERT Virtex6 GTH	
		(Chin	Scope Pro - IBERT)	
00 REFCLK IN		(emp	beoperito ibility	2.1
X0Y0_GTH0_RX_IN		Assign GTHs to a protocol		
	>XDYD_GTHD_USERCLK_OL			
10 YO_OTH1_RX_IN		GTH	Protocol Selected	
	>XDY0_GTH1_USERCLK_OL	Location		
X0Y0_GTH2_RX_IN	→X0Y0_GTH2_TX_OUT	Select Protocol for Quad116	None	~
	→ XDY0_GTH2_USERCLK_OL =	MGT0_116	None	~
X0Y0_GTH3_RX_IN	→X0Y0_GTH3_TX_OUT	MGT1 116	None	
	→ X0 Y0_GTH3_USERCLK_OL			<u> </u>
01_REFCLK_IN		MGT2_116	None	×
X0Y1_GTH0_RX_IN	→ X0Y1_GTH0_TX_OUT	MGT3_116	None	*
	→ X0Y1_GTH0_USERCLK_OL	Select Protocol for Ouad117	OTU4 / 11.18 Gbps	~
X0Y1_GTH1_RX_IN	→ XDY1_GTH1_TX_OUT			
10111 CT10 CY 10	>XBY1_6TH1_USERCLK_OL	MGT0_117	OTU4 / 11.18 Gbps	Y
XDY1_GTH2_RX_IN	→ XDY1_GTH2_TX_OUT = → XDY1_GTH2_USERCLK_OL	MGT1_117	OTU4 / 11.18 Gbps	~
XDY1_GTH3_RX_IN	→ XDY1_GTH3_TX_OUT	MGT2_117	OTU4 / 11.18 Gbps	*
AD TIL OTHOUND IN CITY	→ XDY1_GTH3_USERCLK_OL	MGT3_117	OTU4 / 11.18 Gbps	~
Q2_REFCLK_IN	Anni Cours Corriger			
X0Y2_GTHD_RX_IN		Select Protocol for Quad118	None	×
		MGT0_118	None	*
XDY2_GTH1_RX_IN		MGT1_118	None	~
	→ XDY2_GTH1_USERCLK_OL	MGT2_118	None	~
X0Y2_0TH2_RX_IN	→XBY2_GTH2_TX_OUT	MGT3_118	None	~
	→ XDY2_GTH2_USERCLK_OU			
XDY2_GTH3_RX_IN	→XDY2_GTH3_TX_OUT		OTU4	
	→ X0Y2_GTH3_USERCLK_OL	Overall Totals(GTs):	4 of 4	
D_REFCLK_IN_RIGHT				
X1Y0_GTHD_RX_IN	→X1Y0_GTH0_TX_OUT			

Figure 1-42: CORE Generator - IBERT GTH Customization - Page 3

- 12. Enter the information shown here and in Figure 1-43, then click Next:
 - Quad_117: MGTREFCLK 117

IBERT_SYSCLOCK	₽×	^{logiC≷RE} (C	IBERT Vi hipScope				2.03
Q0_REFCLK_IN	→ X0Y0_GTH0_TX_OUT → X0Y0_GTH0_USERCLK_OL	GTH	Protocol	Linerate (Gbps)	Refclk MHz	Refclk Source	
X0 Y0_GTH1_RX_IN	→XDY0_GTH1_TX_OUT	Quad_117	0TU4	11.18	174.69	MGTREFCLK 117	~
	→>X0Y0_GTH1_USERCLK_OL		1 010-100-000				
X0Y0_GTH2_RX_IN>	→ X0Y0_GTH2_TX_OUT	MGT0_117	OTU4	11.18	174.69	MGTREFCLK 117	2
	>XDY0_GTH2_USERCLK_OL =	MGT1_117	OTU4	11.18	174.69	MGTREFCLK 117	~
X0Y0_GTH3_RX_IN	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	MGT2_117	OTU4	11.18	174.69	MGTREFCLK 117	~
Q1_REFCLK_IN	yangons_overcer_oc	MGT3 117	OTU4	11.18	174.69	MGTREFCLK 117	~
X0 Y1_GTH0_RX_IN	→X0Y1_GTH0_TX_OUT		0.01	11.10	17 1105		
	>X0Y1_GTH0_USERCLK_OL						
X0Y1_GTH1_RX_IN	>X0Y1_GTH1_TX_OUT						
	>X0Y1_GTH1_USERCLK_OL						
X0Y1_GTH2_RX_IN	>X0Y1_GTH2_TX_OUT						
	>>DY1 GTH2 USERCLK OL						
X0Y1 GTH3 RX IN							
	>X0Y1_GTH3_USERCLK_OL						
02 REFCLK IN							
X0Y2_GTH0_RX_IN	>X0Y2_GTH0_TX_OUT						
	>X0Y2_GTH0_USERCLK_OL						
X0Y2_GTH1_RX_IN	>XDY2_GTH1_TX_OUT						
	>>DY2_GTH1_USERCLK_OL						
XBY2_GTH2_RX_IN	>>DY2_GTH2_TK_OUT						
	>X0Y2_GTH2_USERCLK_OL						
	>X0Y2_GTH3_TX_OUT						
VAV2 GTH2 BY IN	2701%_0110_0/_001						
X0Y2_GTH3_RX_IN>	NAME OF THE DESCRIPTION						
X0 Y2_GTH3_RX_IN	→X0Y2_GTH3_USERCLK_OU						

Figure 1-43: CORE Generator - IBERT GTH Customization - Page 4

- 13. Verify the information shown here and in Figure 1-44, then click **Generate**.
 - Production Silicon (v2.1): Selected

ments View				
lodi	₽×	TRE	RT Virtex6 GTH	
_		Indicert		
IBERT_SYSCLOCK		Chips	Scope Pro - IBERT)	2.0
Q0_REFCLK_IN		•	· · · · · · · · · · · · · · · · · · ·	
X0Y0_GTH0_RX_IN	→ X0 Y0_GTH0_TX_OUT	IBERT Design Summary		
	→ X0 Y0_GTH0_USERCLK_OL	Component Name :	ibert_v6_q117	
X0Y0_GTH1_RX_IN	→X0Y0_GTH1_TX_OUT	Number of Protocols :	1	
	→ X0 Y0_0TH1_USERCLK_OL	System Clock Source :	External (Pin: UNASSIGNED)	
X0 Y0_GTH2_RX_IN	→ X0 Y0_GTH2_TX_OUT	No. of Arritry and		
X0 Y0_GTH3_RX_IN	→ X0 Y0_GTH2_USERCLK_OL ■ → X0 Y0_GTH3_TX_OUT	System Clock Frequency :	25 MHz	
2010_01H3_KA_H4	→ X0 Y0_GTH3_USERCLK_OL	BUFG count :	6	
Q1_REFCLK_IN	91010_0110_00ENCE/_00	GTH count :	4	
X0Y1_GTH0_RX_IN	→ X0 Y1_GTH0_TX_OUT	MMCM count :	0	
X0Y1_GTH1_RX_IN	→X0Y1_GTH1_TX_OUT	Refclk sources :	1	
	→X0Y1_GTH1_USERCLK_OL	Silicon Version		
X0Y1_GTH2_RX_IN	→X0Y1_GTH2_TX_OUT	Production Silicon (v2.1)	 Early Silicon (v2.0) 	
	>X0Y1_GTH2_USERCLK_OL	0	0 (()	
X0Y1_0TH3_RX_IN	→X0Y1_6TH3_TX_OUT			
	→ X0 Y1_0TH3_USERCLK_0L			
Q2_REFCLK_IN				
X0Y2_GTH0_RX_IN	>X0Y2_GTH0_TX_OUT			
X0Y2_GTH1_RX_IN	>X0Y2_GTH0_USERCLK_OU >X0Y2_GTH1_TX_OUT			
1011_0110_01	>X0Y2_GTH1_USERCLK_OL			
X0 Y2_GTH2_RX_IN	>10Y2_OTH2_TX_OUT			
X0Y2_GTH3_RX_IN	→X0Y2_GTH3_TX_OUT			
	→ X0 Y2_GTH3_USERCLK_OL			
_REFCLK_IN_RIGHT				
X1 Y0_GTH0_RX_IN	X1 Y0_6TH0_TX_OUT			
	>	Datasheet	< Back Page 5 of 5 Next >	Generate Cancel Help

Figure 1-44: CORE Generator - IBERT GTH Customization - Page 5

- 14. The generation process will take a few minutes. When complete, a *Readme* window will appear (Figure 1-45). Review the information presented and locate the following files:
 - ibert_v6_q117.ngc
 - example_ibert_v6_q117.v
 - example_ibert_v6_q117_top.ucf

Compare the .v and .ucf files generated here with the identically named source files provided with the ML628 board (see Source Directories and Files, page 32) for details on how the SuperClock-2 control module is integrated and the system clock is connected.

4	Readme ibert_v6_q117	×
	The following files were generated for 'bert_v6_g117' in directory C: (ML628_Production\ise\ibert_demo\coregen_O_40d\	
	ibert_v6_q117\example_design\example_ibert_v6_q117.v: Unisim Verilog file containing the information required to simulate the module.	
	<pre>ibert_v6_q117\example_design\ibert_v6_q117_top.ucf: Please see the core data sheet.</pre>	
	ibert_v6_q117\example_design\sysclock_div_ver.v: Unisim Verilog file containing the information required to simulate the module.	
	ibert_v6_q117_flist.txt: Text file listing all of the output files produced when a customized core was generated in the CORE Generator.	
	<pre>ibert_v6_q117\implement\example_ibert_v6_q117.prj: Please see the core data sheet.</pre>	
	ibert_v6_q117\implement\example_ibert_v6_q117.xst: Please see the core data sheet.	
	ibert_v6_q117\implement\example_implement_ibert_v6_q117.prj: Please see the core data sheet.	
	ibert_v6_q117\implement\example_implement_ibert_v6_q117.xst: Please see the core data sheet.	
	ibert_v6_q117\implement\implement.bat: Please see the core data sheet.	
	ibert_v6_q117\implement\implement.sh: Please see the core data sheet.	
	ibert_v6_q117.gise: ISE Project Navigator support file. This is a generated file and should not be edited directly.	
	ibert_v6_q117.ngc: Binary Xilinx implementation netlist file containing the information required to implement the	
	Close Help]:
	UG806 c1 45 04	141

Figure 1-45: CORE Generator - Readme

Creating the GTX IBERT Core

This section provides a procedure to create a single Quad GTX IBERT core using CORE Generator software. The procedure assumes Quad 100 and 6.5 Gb/s line rate, but cores for any of the GTX Quads with any supported line rate can be created following the same series of steps.

For more details on generating IBERT cores, refer to <u>www.xilinx.com/support/</u> <u>documentation/sw_manuals/xilinx13_2/chipscope_pro_sw_cores_ug029.pdf</u>, UG029 - *ChipScope Pro Software Cores*

- 1. Start the CORE Generator tool from either the ISE Project Navigator window or a command line:
 - From the Project Navigator window, select: **Tools** → **Core Generator...**
 - From a command line, enter: coregen
- 2. In the Core Generator window, click the New Project icon (highlighted in Figure 1-36).

Eile	¥iew ∐elp → 🗐 🔲 CORE C	Generator Help	N2							
14	P Catalog	View by <u>N</u> ame		₽×	TE	Vilian		C		
24 .	Name	Δ	Version AXI4 Sta	atus License	LogiCXRE	Xilinx C	URE	Ger	nerato	r
	Communication					project open.				
5	 ▶ ▶	ation				he IP Catalog but you en or create a project.		able to	generate any	ý
E	Example Contor Farmer Contor For Fore Contor Math Functions Memories & Sto Standard Bus In Test Video & Image F	rage Elements terfaces			Copyright (c) 1995-2	010 Xilinx, Inc. All right	ts reserved.			
					Console					8
					Welcome to Xilinx C Help system initializ					
-	Search IP Catalog:			Clear	Search C <u>o</u> nsole		Find		Save	Clear
	search in caralog.			Cica		 Warnings 🔇 Errors	-			

Figure 1-46: Open New Project Icon, CORE Generator Project Window

- 3. When the the New Project dialog window opens (not shown), name the project and click **Save**.
- 4. In the Project Options window, click on **Part** and select the parameters listed here:
 - Family: Virtex6
 - Device: **xc6vhx380t**
 - Package: ff1923
 - Speed Grade: -2

Figure 1-37 shows the correct settings.

Project Options			? 🔀
Part Generation Advanced	Part Select the part for your pr Family Virtex6 Device xc6vhx Package ff1923	19 A	~
	Speed Grade -2	Cancel Apply	
			UG806 c1 47 041411

Figure 1-47: CORE Generator Project Options (Part Options)

- 5. Next in the Project Options window, click on **Generation** and select the parameters listed here:
 - Design Entry: Verilog
 - Vendor: Other
 - Netlist Bus Format: **B<n:m>**
 - Preferred Simulation Model: Structural
 - ASY Symbol File: unchecked

Figure 1-38 shows the correct settings.

🕴 Project Options		? 🛛
Dart Generation Auvanceu	Flow Design Entry Custom Output Products Please refer to the online help for in models using compxlib and using .VE	Verilog formation about compiling behavioral (Verilog) templates.
	Flow Settings	
	Vendor	Other 💌
	Netlist Bus Format	B <n:m> ♥</n:m>
	Simulation Files Preferred Simulation Model O Behavioral O Structural None	Preferred Language
	Other Output Products ASY Symbol File	
	OK Can	cel Apply Help

UG806_c1_48_041411

Figure 1-48: CORE Generator Project Options (Generation Options)

- 6. Click **OK** to close the Project Options window.
- 7. In the IP Catalog pane of the CORE Generator window (Figure 1-49) select:
 Debug & Verification →

 $\textbf{ChipScope Pro} \rightarrow \\$

IBERT Virtex6 GTX (ChipScope Pro - IBERT) 2.05.a

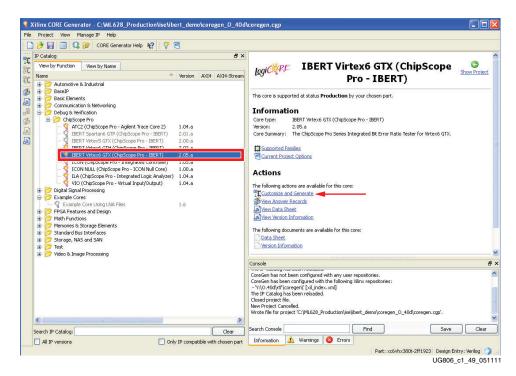


Figure 1-49: Select IBERT Core

- 8. Click on the **Customize and Generate** link under the **Actions** heading (Figure 1-49)
- 9. After a few seconds page 1 of the IP customization window will appear. Enter the information shown here and in Figure 1-50, then click Next:
 - Component Name: ibert_v6_q100
 - Generate Bitstream: unchecked
 - Add RXRECCLK probe: unchecked
 - GTX Naming Style: MGT m n
 - GTX Column: Left
 - Use External clock source: checked
 - Frequency: 25 MHz
 - Pin Location: UNASSIGNED
 - Pin Input Standard: LVCMOS25

IBERT Virtex6 GTX (CF Documents View					
IP Symbol	ē ×		DEDT Vistave C	TV	
		Indicate	[BERT Virtex6 G]		
IBERT_SYSCLOCK		(C	ipScope Pro - IB	ERT)	2.05.a
00_REFCLK0_IN		•			2.00.0
QD_REFCLK1_IN	_	Component Name	ibert v6 g100		
X0_Y0_RX_IN	→ X0_Y0_TX_OUT				
	>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>				
30_Y1_RX_IN>	→X0_Y1_TX_OUT	Custom Davies			
	>X0_Y1_RECCLK_OUT	System Design			
X0_Y2_RX_IN>	→>>0_Y2_TX_0UT	🔲 Generate Bitstream			
	>>0_Y2_RECCLK_OUT	Add RXRECCLK prob			
X0_Y3_RX_IN>	→X0_Y3_TX_0UT	GTX Naming Style	STmini 🔍 ex. MGT0_113 / MGTREFC	CI KO 113	
	→ X0_Y3_RECCLK_OUT	and the second s		50.0_110	
01_REFCLK0_IN		GTX Column	ft MGT0_100-MGT3_105		
Q1_REFCLK1_IN		System Clock			
X0_Y4_RX_IN>	→ XD_Y4_TX_OUT	Use External clock s			
	→ X0_Y4_RECCLK_OUT	Se External Clock s			
X0_Y6_RX_IN>	→ X0_Y5_TX_OUT → X0_Y5_RECCLK_OUT	Frequency	25 MH	łz	
X0_Y6_RX_IN	→ X0_Y6_TX_OUT	Pin Location	UNASSIGNED		
ND_10_KA_IR	>X0_Y6_RECCLK_OUT				
X0_Y7_RX_IN	→ x0_Y7_TX_OUT	Pin Input Standard	LVCMOS25		
	>X0_Y7_RECCLK_OUT				
02_REFCLI0_IN					
02_REFCLK1_IN					
30_Y8_RX_IN	>XD_Y8_TX_OUT				
30_Y9_RX_IN	→ X0_Y9_TX_OUT				
	>X0_Y9_RECCLK_OUT				
X0_Y10_RX_IN	→ X0_Y10_TX_OUT				
,	→X0 Y10 RECCLK OU M				
🐧 IP Symbol 🍕 Core	Details	Datasheet	< Back Page 1 of	f6 Next > Gener	rate Cancel Help

Figure 1-50: CORE Generator - IBERT GTX Customization - Page 1

- 10. Enter the information shown here and in Figure 1-51, then click **Next**:
 - Max Rate (Gps): **6.5**
 - Refclk (MHz): **162.50**
 - GTH count: **4**

cuments View				
Symbol	8 ×		IBERT Virtex6 GTX	
		LogiCXRE		
IBERT_SYSCLOCK		²⁰ (C	hipScope Pro - IBERT)	2.05.
Q0_REFCLK0_IN		-	·	
Q0_REFCLK1_IN		Number of Protocols	1 🗸	
X0_Y0_RX_IN>	→X0_Y0_TX_OUT	-Line rate settings		
		Protocol	Max Rate (Gbps) Data Width Refclk (MHz) GTX count	1
X0_Y1_RX_IN>	→X0_Y1_TX_OUT			
	>X0_Y1_RECCLK_OUT	Name Protocol	✓ 6.5 20 ✓ 162.50 ✓ 4	
X0_Y2_RX_IN>	→X0_Y2_TX_OUT	Custom_1		
	>X0_Y2_RECCLK_OUT			
X0_Y3_RX_IN>	→X0_Y3_TX_OUT	GTXs Resources		
	→X0_Y3_RECCLK_OUT	GTXs count	4	
Q1_REFCLK0_IN				
Q1_REFCLK1_IN		BUFG count	8	
X0_Y4_RX_IN	→X0_Y4_TX_OUT			
	>X0_Y4_RECCLK_OUT			
X0_Y5_RX_IN	→X0_Y5_TX_OUT			
	→X0_Y5_RECCLK_OUT			
X0_Y6_RX_IN	→X0_Y6_TX_OUT			
	→X0_Y6_RECCLK_OUT			
X0_Y7_RX_IN	→X0_Y7_TX_OUT			
	>X0_Y7_RECCLK_OUT			
02_REFCLK0_IN				
02_REFCLK1_IN				
XD_Y8_RX_IN	→X0_Y8_TX_OUT			
	→X0_Y8_RECCLK_OUT			
X0_Y9_RX_IN	→X0_Y9_TX_OUT			
	→X0_Y9_RECCLK_OUT			
XD_Y10_RX_IN	→>X0_Y10_TX_OUT			
m	→X0 Y10 RECCLK OU"			
IP Symbol 🍕 Core	Details	Datasheet	< Back Page 2 of 5 Next >	Generate Cancel Help

Figure 1-51: CORE Generator - IBERT GTX Customization - Page 2

- 11. Enter the information shown here and in Figure 1-52, then click Next:
 - MGT0_110: CUSTOM1 / 6.5 Gps
 - MGT1_110: CUSTOM1 / 6.5 Gps
 - MGT2_110: CUSTOM1 / 6.5 Gps
 - MGT3_110: CUSTOM1 / 6.5 Gps

🂐 IBERT Virtex6 GTX (Chip	Scope Pro - IBERT)		
Documents View			
IP Symbol	8 ×		
IBERT_SYSCLOCK			IBERT Virtex6 GTX ChipScope Pro - IBERT) 205.a
Q0_REFCLK1_IN	E	Assign GTXs to a proto	tocol
X0_Y0_RX_IN → X0_Y1_RX_IN →	→X0_Y0_TX_OUT →X0_Y0_RECCLK_OUT →X0_Y1_TX_OUT	GTX Location	Protocol Selected Region
20_11_02_14	>X0_Y1_RECCLK_OUT	MGT0_100	Custom 1 / 6.5 Gbps V Bottom
X0_Y2_RX_IN	→X0_Y2_TX_OUT	MGT1_100	Custom 1 / 6.5 Gbps
	>X0_Y2_RECCLK_OUT	MGT2_100	Custom 1 / 6.5 Gbps
X0_Y3_RX_IN	→X0_Y3_TX_OUT	MGT3_100	Custom 1 / 6.5 Gbps
	→ X0_Y3_RECCLK_OUT		None v Bottom
01_REFCLK0_IN		 MGT1 101	None v Bottom
X0_Y4_RX_IN>	>X0_Y4_TX_OUT	MGT2_101	None v Bottom
	>X0_Y4_RECCLK_OUT	MGT3_101	None V Bottom
X0_Y5_RX_IN>	→X0_Y5_TX_OUT	MGT0_102	None Statum
	→ X0_Y6_RECCLK_OUT		
X0_Y6_RX_IN>	→X0_Y6_TX_OUT →X0_Y6_RECCLK_OUT	MGT1_102	None V Bottom
0_Y7_RX_IN	>X0_Y7_TX_OUT	MGT2_102	None V Bottom
	>X0_Y7_RECCLK_OUT	MGT3_102	None Bottom
02_REFCLK0_IN>			Custom_1
Q2_REFCLK1_IN>		Page Totals (GTs):	4 of 4
X0_Y8_RX_IN>	→X0_Y8_TX_OUT	Overall Totals(GTs):	4 of 4
10_Y9_RX_IN	→ X0_Y8_RECCLK_OUT → X0_Y9_TX_OUT		
10_10_0A_IN	>X0_Y9_RECCLK_OUT		
X0_Y10_RX_IN			
	→X0 Y10 RECCLK OU Z		
< 🕴 🏹 IP Symbol 🍕 Core D	etails	Datasheet	< Back Page 3 of 6 Next > Generate Cancel Help
			UG806 c1 52 04141



- 12. Leave page 4 unchanged and click **Next**.
- 13. Enter the information shown here and in Figure 1-52, then click Next:
 - MGT0_100: **MGTREFCLK0 100**
 - MGT1_100: **MGTREFCLK0 100**
 - MGT2_100: MGTREFCLK1 100
 - MGT3_100: MGTREFCLK1 100

IBERT_SYSCLOCK		logiCKRE	IBERT Vi ChipScope				2.05
Q0_REFCLK1_IN	=	GTX	Protocol	Linerate	Refclk	Refclk	
X0_Y0_RX_IN	→X0_Y0_TX_OUT	Location		(Gbps)	MHz	Source	
	→X0_Y0_RECCLK_OUT	MGT0_100	Custom_1	6.5	162.50	MGTREFCLK0 100	~
X0_Y1_RX_IN →	→X0_Y1_TX_OUT →X0 Y1 RECCLK OUT	MGT1 100	Custom 1	6.5	162.50	MGTREFCLK0 100	~
X0_Y2_RX_IN	→x0_Y2_TX_OUT	MGT2_100	Custom_1	6.5	162.50	MGTREFCLK1 100	~
				10000		_	
X0_Y3_RX_IN	→X0_Y3_TX_OUT	MGT3_100	Custom_1	6.5	162.50	MGTREFCLK1 100	*
01_REFCLK1_IN	→ x0_Y4_TX_OUT → x0_Y4_RECCLK_OUT						
X0_Y5_RX_IN	→ ₩_Y8_TK_OUT → ₩_Y8_RECCLK_OUT → ₩_Y8_TK_OUT → ₩_Y8_RECCLK_OUT						
	→ X0_Y6_RECCLK_OUT → X0_Y6_TX_OUT						
X0_Y6_RX_IN	→X0_Y6_RECCLK_OUT →X0_Y6_TK_OUT →X0_Y6_RECCLK_OUT →X0_Y6_RECCLK_OUT						
$30_{Y6}RX_{IN} \longrightarrow$ $30_{Y7}RX_{IN} \longrightarrow$ $02_{REFCLY0_{IN}} \longrightarrow$ $02_{REFCLY0_{IN}} \longrightarrow$	>>>>>>>>>>>>>						
$X0_Y0_RX_IN \longrightarrow$ $X0_Y7_RX_IN \longrightarrow$ $02_REFCLK0_IN \longrightarrow$	→X0_Y6_RECCLK_OUT →X0_Y6_TK_OUT →X0_Y6_RECCLK_OUT →X0_Y6_RECCLK_OUT						
$30_{Y6}RX_{IN} \longrightarrow$ $30_{Y7}RX_{IN} \longrightarrow$ $02_{REFCLY0_{IN}} \longrightarrow$ $02_{REFCLY0_{IN}} \longrightarrow$	→ 30_Y5_RECCLV_0UT → 30_Y6_TV_0UT → 30_Y6_TV_0UT → 30_Y7_TV_0UT → 30_Y7_RECCLV_0UT → 30_Y8_TV_0UT						



14. Verify the information shown in Figure 1-54, then click Generate.

Symbol	ē ×	TDE	RT Virtex6 GTX	
		Indicoke		
IBERT_SYSCLOCK		(Chips	Scope Pro - IBERT)	2.0
Q0_REFCLK0_IN				
Q0_REFCLK1_IN	E	IBERT Design Summary		
X0_Y0_RX_IN	→x0_Y0_TX_OUT	Component Name :	ibert_v6_q100	
	→)0_Y0_RECCLK_OUT	Number of Protocols :	1	
X0_Y1_RX_IN	→x0_Y1_TX_OUT	System Clock Source :	External (Pin: UNASSIGNED)	
X0 Y2 RX IN	→ X0_Y1_RECCLK_OUT → X0_Y2_TX_OUT	System Clock Frequency :	25 MHz	
201120000 m	>X0_Y2_RECCLK_OUT			
X0_Y3_RX_IN	→x0_Y3_TX_OUT	BUFG count :	9	
	>X0_Y3_RECCLK_OUT	GTX count :	4	
Q1_REFCLK0_IN		MMCM count :	0	
Q1_REFCLK1_IN		Refclk sources :	2	
X0_Y4_RX_IN	→X0_Y4_TX_OUT	Reicik sources .	2	
	→X0_Y4_RECCLK_OUT			
X0_Y5_RX_IN	→ X0_Y5_TX_OUT			
	→ X0_Y6_RECCLK_OUT			
X0_Y6_RX_IN	→ X0_Y6_TX_OUT			
30_17_RX_IN	→ X0_Y6_RECCLK_OUT → X0_Y7_TX_OUT			
AD_IT_BALIN	>X0_Y7_RECCLK_OUT			
02_REFCLK0_IN>	,			
02_REFCLK1_IN				
30_Y8_RX_IN	→X0_Y8_TX_OUT			
X0_Y9_RX_IN	→X0_Y9_TX_OUT			
	→ X0_Y9_RECCLK_OUT			
X0_Y10_RX_IN	→ X0_Y10_TX_0UT			
	→X0 Y10 RECCLK OU Y			

Figure 1-54: **CORE Generator - IBERT GTX Customization - Page 6**

15. The generation process will take a few minutes. When complete, a *Readme* window will appear (Figure 1-55). Review the information presented and locate the following files:

- ibert_v6_q100.ngc
- example_ibert_v6_q100.v
- example_ibert_v6_q100_top.ucf

Compare the .v and .ucf files generated here with the identically named source files provided with the ML628 board (see Source Directories and Files, page 32) for details on how the SuperClock-2 control module is integrated and the system clock is connected.

<pre>% Readme ibert_v6_q100</pre>	? 🛛
The following files were generated for 'bert_v6_q100' in directory C: \ML628_Production\ise\ibert_demo\coregen_0_40d\	^
ibert_v6_q100\example_design\example_ibert_v6_q100.v: Unisim Verilog file containing information required to simulate the module.	the
<pre>ibert_v6_q100\example_design\ibert_v6_q100_top.ucf: Please see the core data sheet.</pre>	
ibert_v6_q100_flist.txt: Text file listing all of the output files produced when a customized core to in the CORE Generator.	was generated
ibert_v6_q100\implement\chipscope_icon.xco: CORE Generator input file containing the pa used to regenerate a core.	arameters
ibert_v6_q100\implement\coregen.cgp: Please see the core data sheet.	
ibert_v6_q100\implement\example_implement_ibert_v6_q100.prj: Please see the core	e data sheet.
<pre>ibert_v6_q100\implement\example_implement_ibert_v6_q100.xst: Please see the cor</pre>	e data sheet. 📃
ibert_v6_q100\implement\implement.bat: Please see the core data sheet.	
<pre>ibert_v6_q100\implement\implement.sh: Please see the core data sheet.</pre>	
ibert_v6_q100.gise: ISE Project Navigator support file. This is a generated file and should not be directly.	edited
ibert_v6_q100.ngc: Binary Xilinx implementation netlist file containing the information required to module in a Xilinx (R) FPGA.	implement the
4 . e 100	<u>×</u>
Close	Help
	UG806_c1_55_041411

Figure 1-55: CORE Generator - Readme

Warranty

THIS LIMITED WARRANTY applies solely to standard hardware development boards and standard hardware programming cables manufactured by or on behalf of Xilinx ("Development Systems"). Subject to the limitations herein, Xilinx warrants that Development Systems, when delivered by Xilinx or its authorized distributor, for ninety (90) days following the delivery date, will be free from defects in material and workmanship and will substantially conform to Xilinx publicly available specifications for such products in effect at the time of delivery. This limited warranty excludes: (i) engineering samples or beta versions of Development Systems (which are provided "AS IS" without warranty); (ii) design defects or errors known as "errata"; (iii) Development Systems procured through unauthorized third parties; and (iv) Development Systems that have been subject to misuse, mishandling, accident, alteration, neglect, unauthorized repair or installation. Furthermore, this limited warranty shall not apply to the use of covered products in an application or environment that is not within Xilinx specifications or in the event of any act, error, neglect or default of Customer. For any breach by Xilinx of this limited warranty, the exclusive remedy of Customer and the sole liability of Xilinx shall be, at the option of Xilinx, to replace or repair the affected products, or to refund to Customer the price of the affected products. The availability of replacement products is subject to product discontinuation policies at Xilinx. Customer may not return product without first obtaining a customer return material authorization (RMA) number from Xilinx.

THE WARRANTIES SET FORTH HEREIN ARE EXCLUSIVE. XILINX DISCLAIMS ALL OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT, AND ANY WARRANTY THAT MAY ARISE FROM COURSE OF DEALING, COURSE OF PERFORMANCE, OR USAGE OF TRADE. (2008.10)



Do not throw Xilinx products marked with the "crossed out wheelie bin" in the trash. Directive 2002/96/EC on waste electrical and electronic equipment (WEEE) requires the separate collection of WEEE. Your cooperation is essential in ensuring the proper management of WEEE and the protection of the environment and human health from potential effects arising from the presence of hazardous substances in WEEE. Return the marked products to Xilinx for proper disposal. Further information and instructions for free-of-charge return available at: <u>http://www.xilinx.com/ehs/weee.htm</u>.

