ML623 IBERT Getting Started Guide (ISE 13.1)

UG725 (v4.0) May 11, 2011

XILINX®

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/22/10	1.0	Initial Xilinx release.
10/25/10	2.0	Revised Requirements, page 5 to include equipment and software required for regenerating IBERT designs. Added URL for online access to design files in Extracting the IBERT Demonstration Files, page 6. Revised Figure 1-16, Figure 1-17, and Figure 1-18 to reflect ChipScope™ Pro software v12.1. Added Regenerating IBERT Designs, page 21 through page 35.
01/19/11	3.0	Revised document to reflect ChipScope™ Pro software v12.3.
01/28/11	3.0.1	Revised cover title. Was: "ML623 IBERT Getting Started Guide." Is: "ML623 IBERT Getting Started Guide (ISE 12.3)."
05/11/11	4.0	Revised links, software references, and figures containing screen captures to reflect ISE software v13.1. Updated the Si570 and Si5368 addresses and frequencies in Table 1-3, page 19.

Table of Contents

XILINX®

Chapter 1

ML623 IBERT Getting Started Guide

Overview

This document provides a procedure for setting up the ML623 Virtex®-6 FPGA GTX 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 the CompactFlash memory card that is provided with the ML623 board. The demonstration shows the capabilities of the Virtex-6 XC6VLX240T FPGA GTX transceivers.

The IBERT demonstration operates one GTX Quad at a time. This procedure describes the steps to test a .single Quad, Q112. The remaining four Quads are tested following the same series of steps. The procedure consists of:

1. Extracting the IBERT Demonstration Files.
2. Setting Up the ML623 Board.
3. Connecting the GTX Transceivers and Reference Clocks.
4. Configuring the FPGA.
5. Setting Up the ChipScope Pro Analyzer Tool.
6. Running the IBERT Demonstration.
7. Repeating the IBERT Demonstration for the Remaining GTX Quads.

The ML623 board is described in detail in UG724, *ML623 Virtex-6 FPGA GTX Transceiver Characterization Board User Guide*.

Requirements

The equipment and software required to run the IBERT demonstration are:

- ML623 Virtex-6 FPGA GTX Transceiver Characterization Board including:
  - 12V DC power adapter
  - CompactFlash memory card containing the IBERT demonstration designs
  - GTX transceiver power supply module (installed on ML623 board)
  - SuperClock-2 module (installed on ML623 board)
  - 12 SMA to SMA Cables
- One of these JTAG cables:
  - Platform Cable USB-II (DLC10)
  - Parallel IV Cable (PC4)
- Host PC or Linux system, with USB ports

- Xilinx® ChipScope™ Pro software, version 13.1 or higher.
  Software is available at: http://www.xilinx.com/chipscopepro

The additional equipment and software required to regenerate the designs are:

- Linux system with Xilinx ISE® Design Suite v13.1 already installed
- ML623 IBERT design source files (provided online as collection `rdf0097_13-1.zip`) at:

  http://www.xilinx.com/products/boards/ml623/reference_designs.htm

Running the IBERT Demonstration

Extracting the IBERT Demonstration Files

The ChipScope Pro Software `.cpj` project files for the IBERT demonstration are located on the CompactFlash memory card that is provided with the ML623 board. They are also located online along with `.bit` files for all five designs (as collection `rdf0080_13-1.zip`) at:

http://www.xilinx.com/products/boards/ml623/reference_designs.htm

The `.cpj` files are used to load pre-saved MGT/IBERT and clock module control settings for the demonstration. These files must be copied to a working directory on the host computer. 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 `ml623.zip` on the Compact Flash memory card. The ZIP file content is similar to the files shown in Figure 1-1.

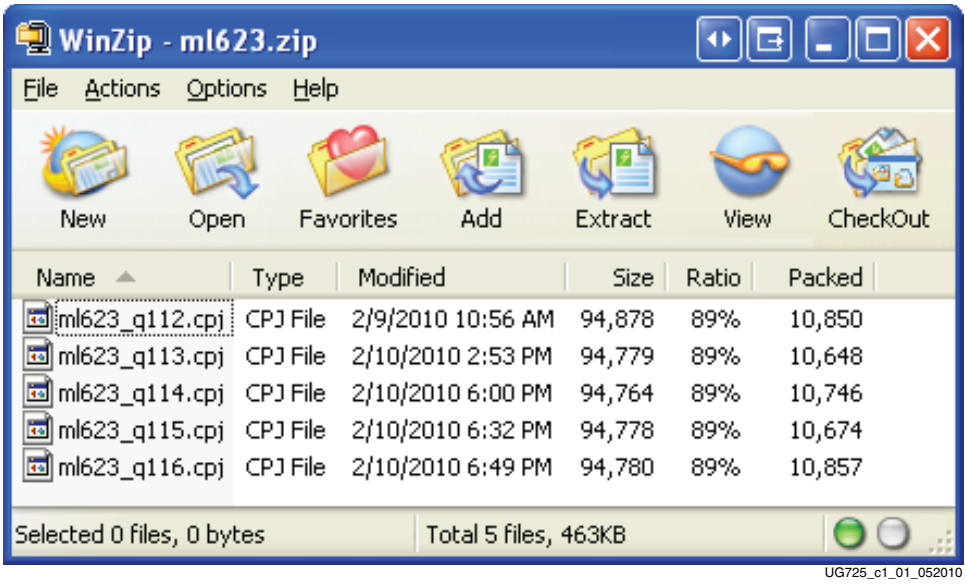


Figure 1-1: **ChipScope Project Files Included in the ml623.zip File**

3. Unzip the files to a working directory on the host computer.

Setting Up the ML623 Board

Caution! The ML623 board can be damaged by electrostatic discharge (ESD). Follow standard ESD prevention measures when handling the board.

To set up the ML623 board:

1. Install the GTX transceiver power module:
   a. Plug the module into connectors J34 and J179.
   b. Remove DCPS ENABLE jumpers at J184 and J185 located on the ML623 board.
2. Verify the four SYSACE JTAG ENABLE jumpers are installed at locations J22, J23, J195, and J196 on the ML623 board.
3. Place a jumper across pins 1–2 of the JTAG FMC BYPASS header at J162.
4. Enable the 200 MHz LVDS system clock by placing two jumpers (P, N) across pins 1–3 and pins 2–4 of J188.
5. Verify there is a 30 MHz oscillator in the SYSTEM ACE CLK oscillator socket at location X1 on the ML623 board.
6. Enable the System ACE™ controller clock by placing the jumper on J4 to the ON position.
7. Insert the CompactFlash memory card into the CF card connector (U24) located on the underside of the ML623 board.
8. 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 ML623 board.
   b. Using three 4-40 x 0.25 inch screws, firmly screw down the module from the bottom of the ML623 board.
   c. On the SuperClock-2 module, place a jumper across pins 2–3 (2V5) of the CONTROL VOLTAGE header, J18.

Connecting the GTX Transceivers and Reference Clocks

All GTX transceiver pins are connected to differential SMA connector pairs. The GTX transceivers are grouped into five sets of four (referred to as Quads) which share two differential reference clock pin pairs. Figure 1-2 shows the SMA locations for the GTX transceiver Quads (Q112 through Q116) and their associated reference clocks (112 Clocks through 116 Clocks).

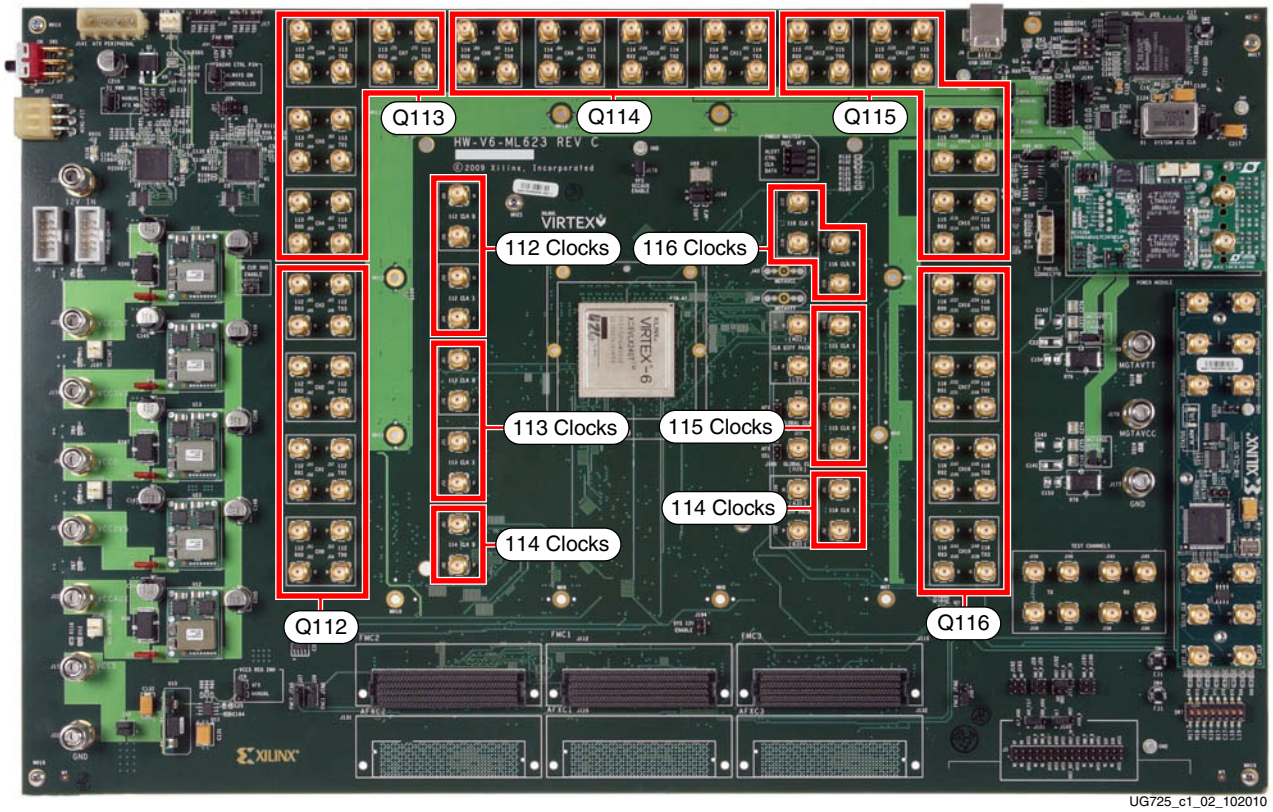


Figure 1-2: **GTX Transceiver and Reference Clock SMA Locations**

Note: The image in Figure 1-2 is for reference only and might not reflect the current revision of the board.

The SuperClock-2 module provides LVDS clock outputs for the GTX transceiver reference clocks in the IBERT demonstration. Figure 1-3 shows the location of the differential clock SMA connectors on the clock module which may be connected to the GTX transceiver reference clock SMAs on the ML623 board. The four SMA pairs labeled "CLKOUT" provide LVDS clock outputs from the Si5378 clock multiplier/jitter attenuator device on the clock module. The SMA pair labeled "Si570_CLK" provides LVDS clock output from the Si570 programmable oscillator on the clock module. For the IBERT demonstration, the output clock frequencies from both devices are preset to 162.5 MHz. For more information regarding the SuperClock-2 module, refer to UG770, *HW-CLK-101-SCLK2 SuperClock-2 Module User Guide*.

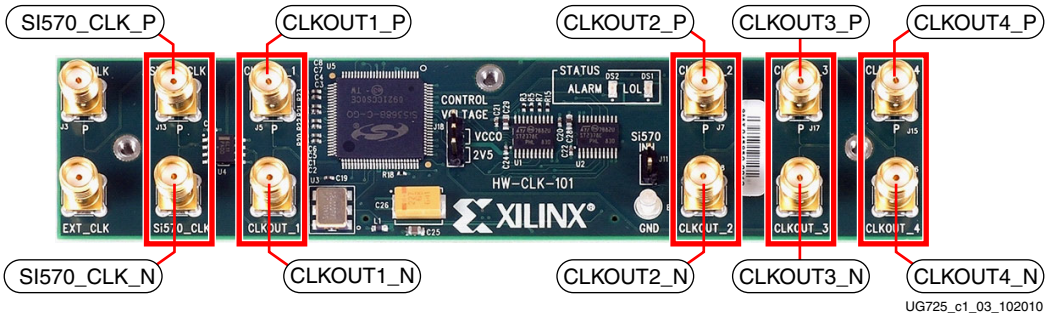


Figure 1-3: **SuperClock-2 Module Output Clock SMA Locations**

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

GTX Transceiver Clock Connections

Refer to Table 1-1 and use four SMA cables to connect the output clock SMAs from the SuperClock-2 module to the reference clock SMAs of GTX Quad Q112 on the ML623 board. In other words, for each row in Table 1-1, connect the source SMA with its corresponding destination SMA. For example, connect CKOUT1_P (J5) to 112_REFCLK0_P (J59).

Note: Any one of the five differential output SMA clocks from the clock module can be used to source either REFCLK0_P|N or REFCLK1_P|N on the ML623 board. Output clocks from the Si5368 device, specifically CKOUT1_P|N and CKOUT2_P|N, are described here and throughout this document as an example.

Table 1-1: **Q112 Reference Clock Connections**

Source		Destination	
SuperClock-2 Module		ML623 Board	
Net Name	SMA Connector	Net Name	SMA Connector
CKOUT1_P	J5	112_REFCLK0_P	J59
CKOUT1_N	J6	112_REFCLK0_N	J60
CKOUT2_P	J7	112_REFCLK1_P	J49
CKOUT2_N	J8	112_REFCLK1_N	J50

See Appendix A for a complete listing of reference clock SMA connections for the remaining GTX Quads.

GTX TX/RX Connections

Refer to Table 1-2 and use eight SMA cables to connect the transmitter SMAs to the receiver SMAs in GTX Quad Q112. In other words, for each row in Table 1-2, connect the transmitter SMA with its corresponding receiver SMA.

For example, connect 112_TX0_P (J53) to 112_RX0_P (J51) on the ML623 board.

Table 1-2: **Q112 TX/RX Connections**

Transmitter		Receiver	
Net Name	SMA Connector	Net Name	SMA Connector
112_TX0_P	J53	112_RX0_P	J51
112_TX0_N	J54	112_RX0_N	J52
112_TX1_P	J57	112_RX1_P	J55
112_TX1_N	J58	112_RX1_N	J56
112_TX2_P	J43	112_RX2_P	J41
112_TX2_N	J44	112_RX2_N	J42
112_TX3_P	J47	112_RX3_P	J45
112_TX3_N	J48	112_RX3_N	J46

See Appendix A for a complete listing of TX/RX SMA connections for the remaining GTX Quads.

The final SMA cable connections for Q112 are shown in Figure 1-4.

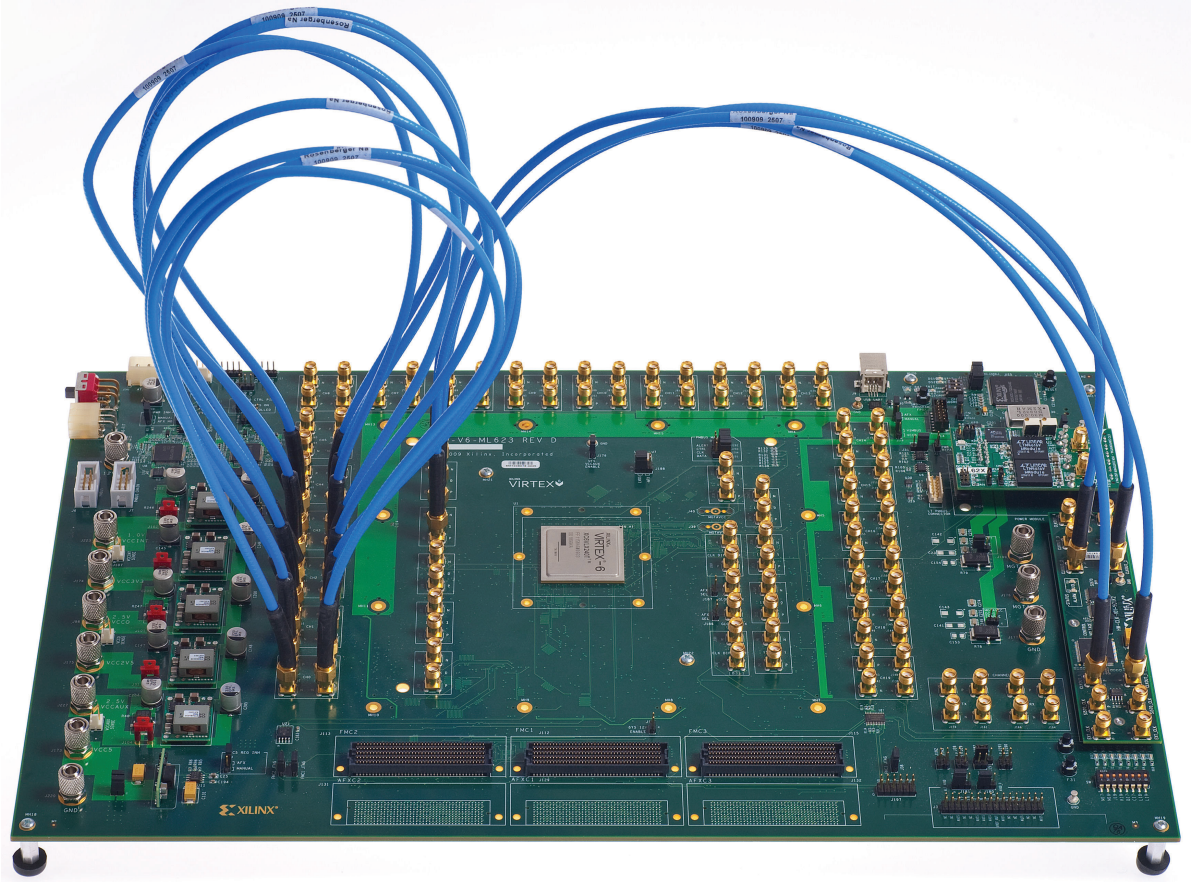


Figure 1-4: **SMA Cable Connections for Q112 Transceivers and Clocks**

Configuring the FPGA

The following set of instructions describe how to configure the FPGA using the CompactFlash memory card included with the board. The FPGA may also be configured through ChipScope analyzer or iMPACT using the `.bit` files located online (as collection `rdf0080_13-1.zip`) at:

http://www.xilinx.com/products/boards/ml623/reference_designs.htm

To configure from the CompactFlash memory card:

1. Plug the 12V output from the power adapter into connector J122.
2. Connect the ML623 board to the host computer. Either of these cables may be used for this connection:
   - Platform Cable USB-II (DLC10)
   - Parallel IV Cable (PC4)

   Connect one end of the cable to the host computer. Connect the other end to the download cable connector (J1) on the ML623 board.

3. Set the System Ace Controller Configuration Address switch, SW3. The setting on this 3-bit DIP switch (shown in Figure 1-5) determines which bitstream stored in the CompactFlash card configures the FPGA. There are five IBERT demonstration designs on the CompactFlash card, one for each of the five GTX Quads available on the ML623 board. Note that Figure 1-5 shows an example setting of ADR[2:0] = `100`. For Q112, the configuration address is `000`.

   Table 1-3 lists the SW3 DIP switch settings for each of five Quads available on the ML623 board.

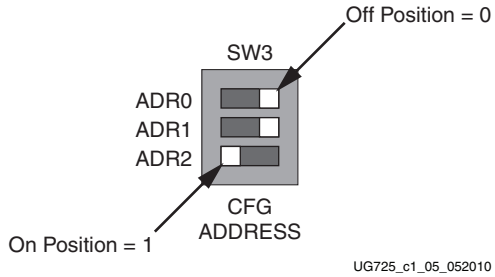


Figure 1-5: **Configuration Address DIP Switch (SW3)**

Table 1-3: **DIP Switch SW3 Configuration Addresses**

Demonstration Design	ADR2	ADR1	ADR0
Q112	0	0	0
Q113	0	0	1
Q114	0	1	0
Q115	0	1	1
Q116	1	0	0

4. Apply power to the board by placing SW1 in the ON position. After a few seconds, the FPGA is configured and the Done LED (DS6) lights.

Setting Up the ChipScope Pro Analyzer Tool

1. Open the ChipScope Pro analyzer tool and select **File** → **Open Project**.

2. When the Open Project window appears, navigate to the location on the host computer where the `.cpj` project files were extracted, select `ml623_q112.cpj` and click **Open** (Figure 1-6).

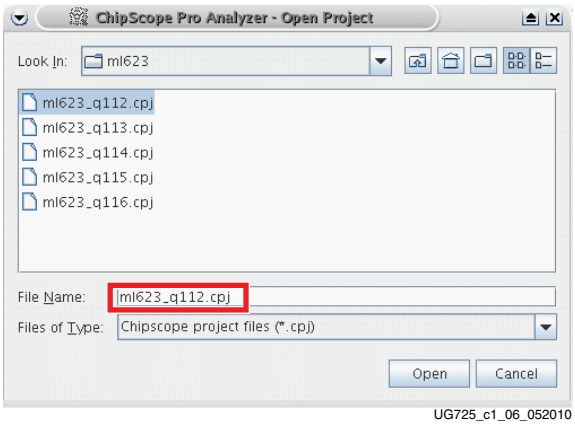


Figure 1-6: **Open Project Window**

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 UG029, *ChipScope Pro Software and Cores User Guide*.

3. Click the **Open Cable** button (Figure 1-7).

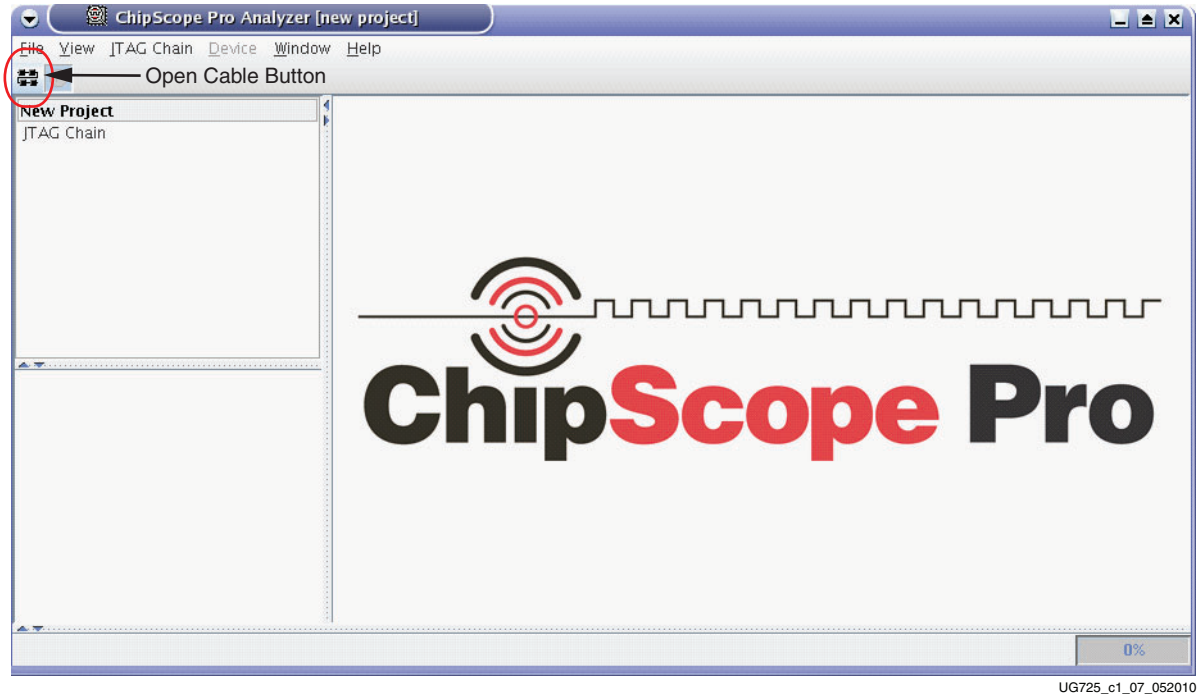


Figure 1-7: **Open Cable Button**

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

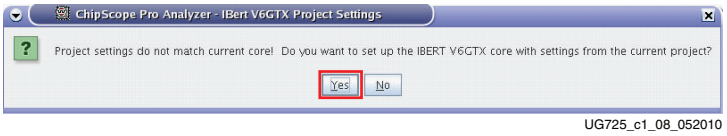


Figure 1-8: **Core Settings Dialog Box**

5. When the project panel opens, verify the JTAG chain shows the devices listed in Figure 1-9.

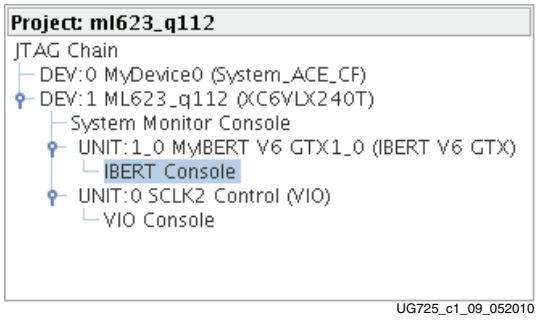


Figure 1-9: **Project Panel**

Starting the Clock Module

The IBERT demonstration design uses a ChipScope Pro VIO core to control the clocks on the SuperClock-2 module. The SuperClock-2 module features two clock-source components: An always-on Si570 crystal oscillator and an Si5368 jitter-attenuating clock multiplier. The IBERT demonstration uses the output from either device to clock the GTX transceivers.

1. In the project panel, double-click **VIO Console** (Figure 1-10).

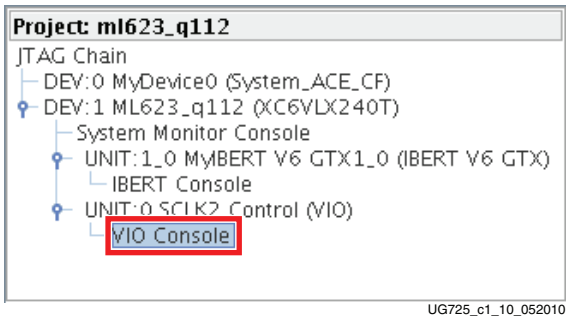


Figure 1-10: **VIO Console Selection**

2. Having selected the VIO Console, the clock source(s) for the GTX transceivers can be initialized. Do one or both of the following:

   a. If using the Si5368 device to source the GTX transceiver clocks (e.g. as described in Table 1-1, page 9), initialize the Si5378 device. Click the **Si5368 Start** button (Figure 1-11). A transition arrow flashes ON/OFF to the right of **Si5368 Done** when the command is complete.

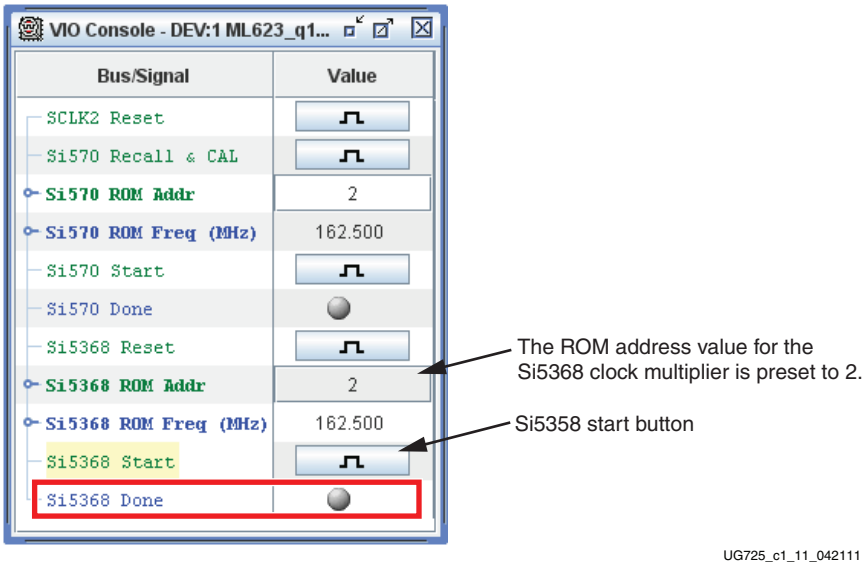


Figure 1-11: **VIO Console, Si5368 Initialization**

   b. If using the Si570 crystal to source the GTX transceiver clocks, click the Si570 Start button (Figure 1-12). A transition arrow flashes ON/OFF to the right of **Si570 Done** when the command is complete.

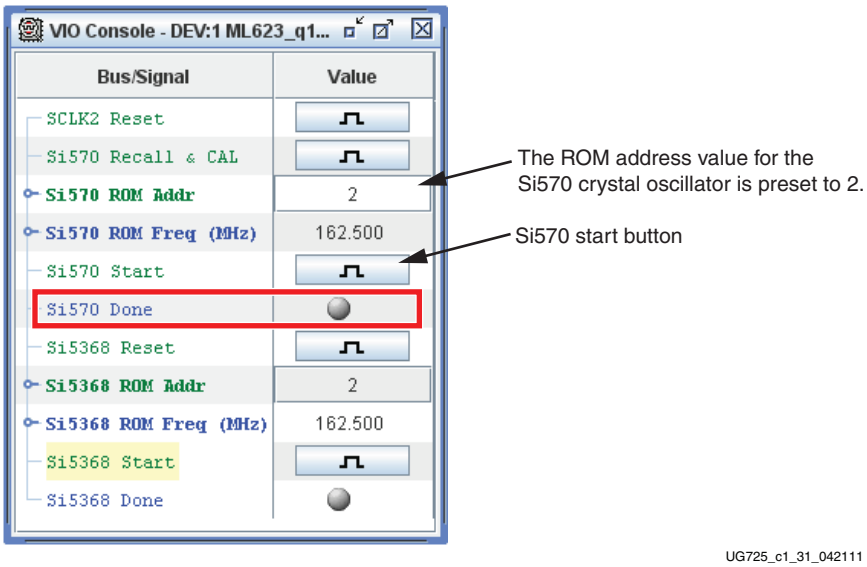


Figure 1-12: **VIO Console, Si570 Initialization**

Note: The ROM address value for the Si5368 and Si570 devices is preset to 2 to produce an output frequency of 162.5 MHz. Typing in a different address changes the frequency of the GTX

transceiver reference clocks. A complete list of frequency options and their associated ROM addresses is provided in Table 1-4, page 20.

3. In the project panel, double-click **IBERT Console** (Figure 1-13).

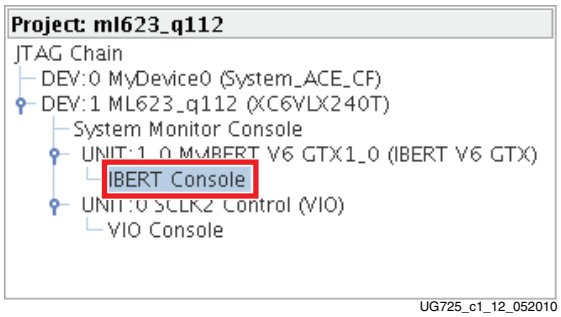


Figure 1-13: **IBERT Console Selection**

4. At the top of the ChipScope Pro analyzer window, click the **Reset All** button (Figure 1-14).

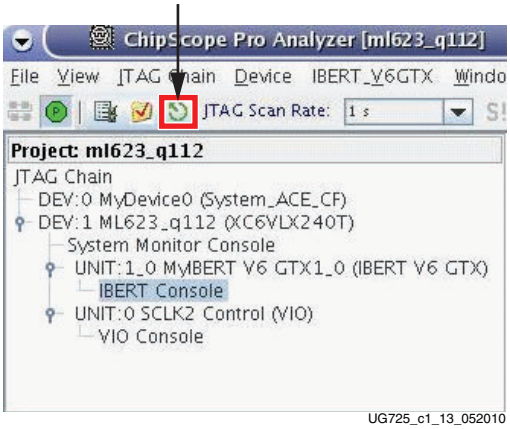


Figure 1-14: **Reset All Button**

5. When the confirmation dialog box opens, click **Yes** (Figure 1-15).

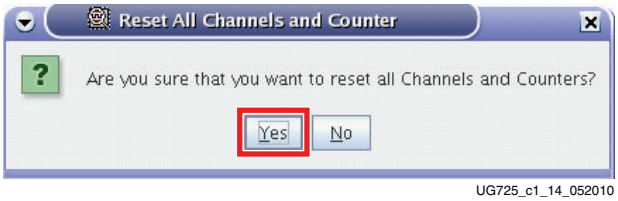


Figure 1-15: **Reset Confirmation Dialog Box**

Running the IBERT Demonstration

After completing step 5 in Starting the Clock Module, the IBERT demonstration is configured and running as indicated by the **MGT/IBERT Settings** tab within the IBERT Console.

Viewing GTX Transceiver Operation

1. Note the line rate is 6.5 Gb/s for all four GTX transceivers (MGT Link Status in Figure 1-16).

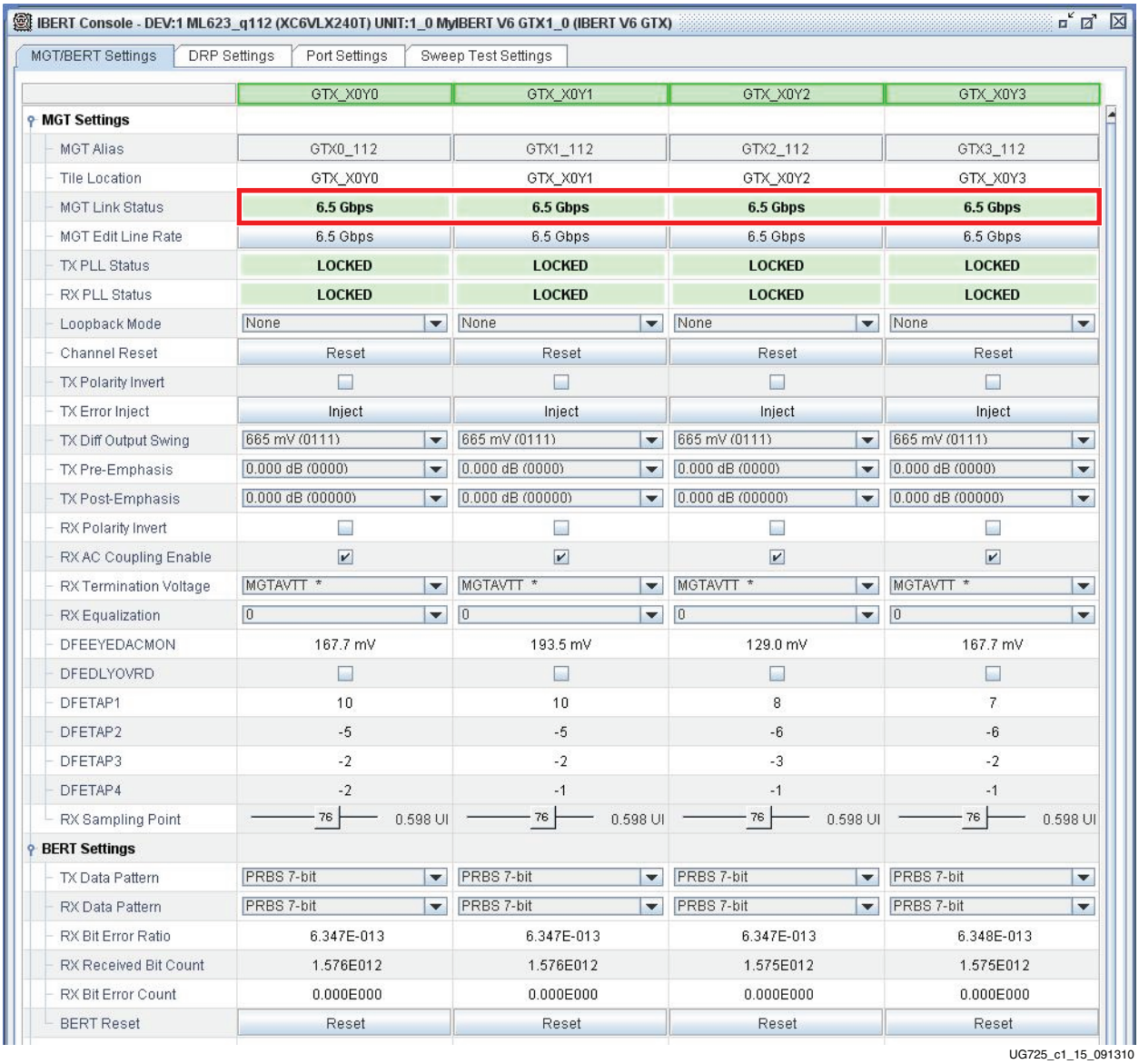


Figure 1-16: **GTX Transceiver Link Status**

2. Note the GTX transmitter differential output swing is preset to 665 mV (`0111`) as shown in Figure 1-17.

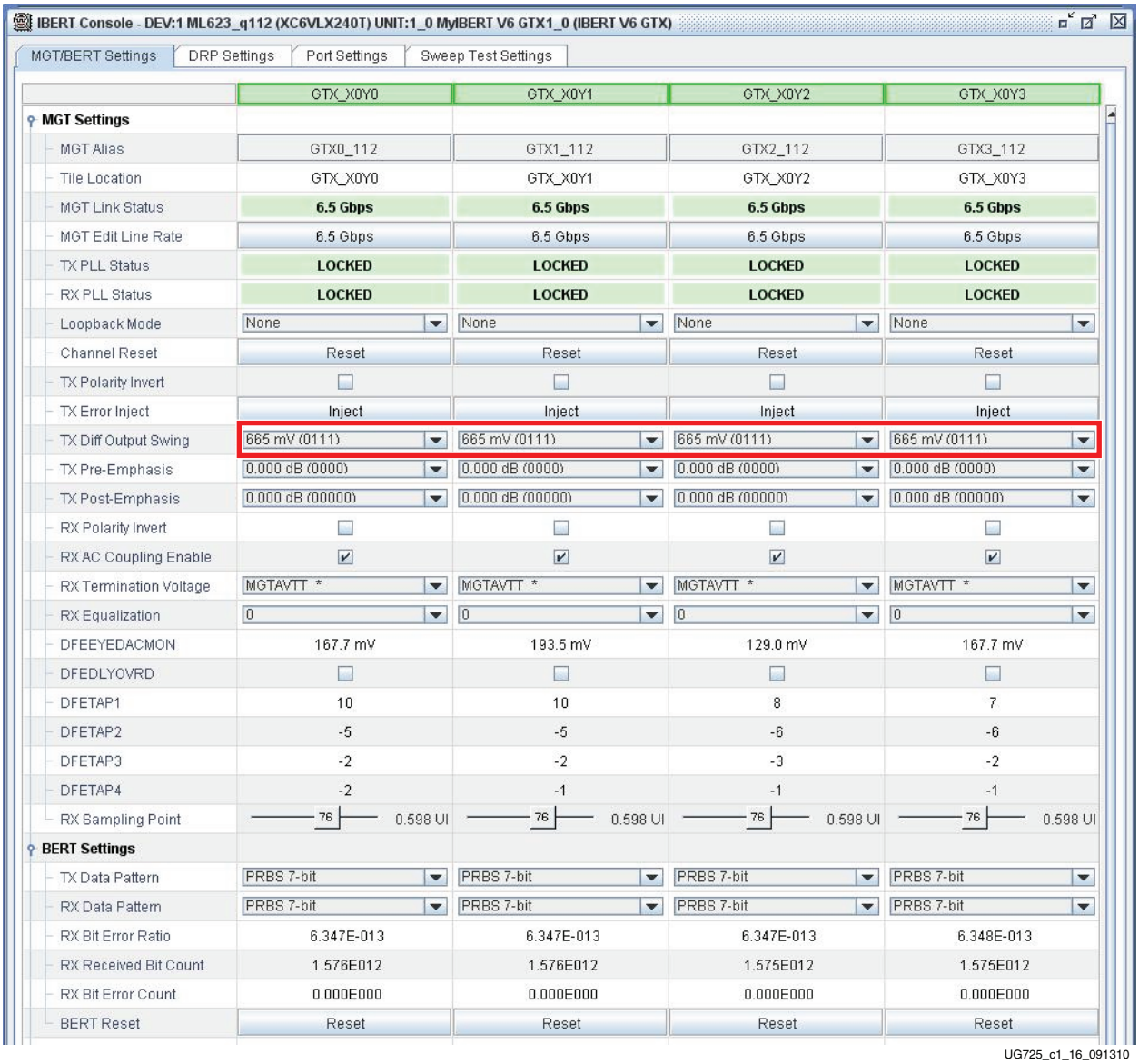


Figure 1-17: **GTX Transceiver TX Differential Output Swing**

3. Note that there are no bit errors as indicated by the RX Bit Error Count as shown in Figure 1-18.

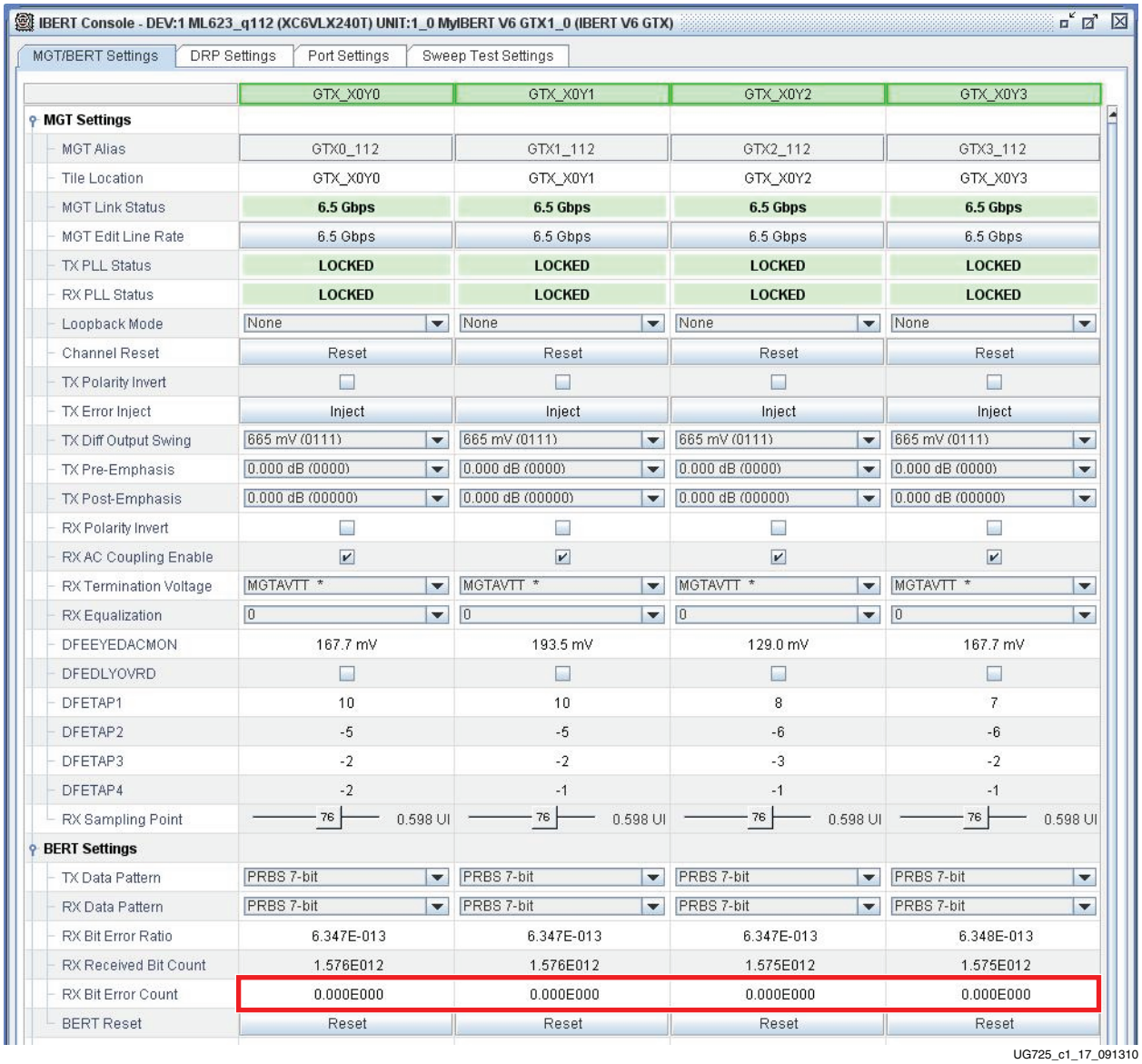


Figure 1-18: **RX Bit Error Count**

Stopping the IBERT Demonstration

To stop the IBERT demonstration:

1. Close the ChipScope Pro analyzer tool.

   Note: Do not save changes to the project.

2. Remove power to the ML623 board by placing SW1 in the OFF position.
3. Remove the SMA cables from the ML623 board.

Repeating the IBERT Demonstration for the Remaining GTX Quads

Start at Connecting the GTX Transceivers and Reference Clocks, page 7 and repeat the procedure for the remaining GTX Quads. Appendix A lists the connections for Q113, Q114, Q115, and Q116.

Frequency Table

Table 1-4 lists the addresses of the output frequencies of the Si570 and Si5360 programmable clock sources.

Table 1-4: **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

Table 1-4: **Si570 and Si5368 Frequency Table *(Cont'd)***

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			

Regenerating IBERT Designs

Source File Overview

The file `rdf0097_13-1.zip` contains the source files for five individual designs (one for each GTX Quad on the ML623 board). The `.zip` file is located at:

http://www.xilinx.com/products/boards/ml623/reference_designs.htm.

To set up the source files:

1. Download `rdf0097_13-1.zip` to a working directory on the Linux System.
2. Unzip the files to the working directory.

   The project directory file structure for each Quad is as follows:

```
ML623_q11x/
  par/
    chipscope_icon.xco
    coregen.cgp
    example_implement_ibert_v6_q11x.prj
    example_implement_ibert_v6_q11x.xst
    i2c_sclk2_control.ngc
    ibert_v6_q112.ngc
    icon_v6_1.ngc
    implement.sh
    vio_v6_si84_so78.ngc
  src/
    chipscope.v
    example_ibert_v6_q11x.v
    i2c_sclk2_control_bb.v
    ibert_sclk2_link.dat
```

```
ibert_v6_q11x_top.ucf
sysclk_i2c_sclk2_pins.ucf
vio_sclk2_control.v
```

IBERT Design IP Components

The IBERT design IP consists of three main components:

- **example_ibert_v6_q11x**
  A four-channel IBERT core utilizing two reference clocks.
- **vio_sclk2_control**
  A ChipScope Pro virtual I/O controller core for the SuperClock-2 module.
- **icon_v6_1**
  Single-channel Integrated Controller (ICON) core for Virtex-6 devices.

Note: `ibert_v6_q11x` uses the BSCAN USER1 scan chain, `icon_v6_1` uses the BSCAN USER2 scan chain.

An example design hierarchy is:

```
example_ibert_v6_q112.v
  icon_v6_1.ngc
  vio_sclk2_control.v
    i2c_sclk2_control_bb.v
```

example_ibert_v6_q11x Module

Each Quad has its own individual `example_ibert_v6_q11x.v` module generated by the Xilinx CORE Generator™ v13.1 (using the IBERT Virtex-6 GTX core, v2.05.a) without the **Implement Design** option selected. The module features four GTX lanes (one lane equals: TXP, TXN, RXP, RXN), two reference clock inputs (REFCLK0, REFCLK1), and a 25 MHz system clock.

vio_sclk2_control Module

The `vio_sclk2_control.v` module provides a VIO core for controlling the SuperClock-2 module through the ChipScope Pro software. The `vio_sclk2_control.v` module features 84 synchronous inputs (14 free) and 78 synchronous outputs (12 free). No logic exists in this level because `vio_sclk2_control.v` is only a wrapper. The `i2c_sclk2_control` module instantiated at this level is a black-box HDL module and is provided as an ISE software v11.4 NGC file.

CLK50

The IBERT design uses a 25 MHz system clock to match the IBERT requirements. Using the same clock, the I^2C interface runs at half its target clock frequency of 50 MHz with no impact on the functionality or performance of the design.

Design Notes

All files are built using ISE Design Suite, v13.1. The ML623 IBERT design uses a new methodology to combine an IBERT from the CORE Generator software with user logic.

The `vio_sclk2_control` module is configured with fixed values to reduce user error:

- **`sclk_out[3]` - Si5368 RESET_B pin**
  Connected to Logic 1 to avoid accidental reset of the Si5368 jitter-attenuating clock multiplier on the SuperClock-2 module.
- **`pca0_ctrl[5:0]`**
  Set to `0x05`. Enables the SuperClock-2 module on the I^2C bus only.
- **`si570_idcode`**
  Set to the idcode of the Si570 crystal oscillator on the SuperClock-2 module (`0x55`).

Recreating IBERT Module with CORE Generator

This procedure describes the steps to recreate the IBERT module for GTX transceiver Quad Q112. IBERT modules for the remaining four Quads can be recreated following the same series of steps.

To recreate the IBERT module from CORE Generator, follow these steps on a Linux system on which ISE Design Suite v13.1 is installed.

1. Open a command window.
2. In the command window, navigate to the top-level directory where the IBERT source files are located. Source File Overview is described on page 21.
3. Open up CORE Generator by executing the following command:

   ```
   % coregen
   ```

4. When the Core Generator window appears on screen, click the **New Project** icon (highlighted in Figure 1-19).

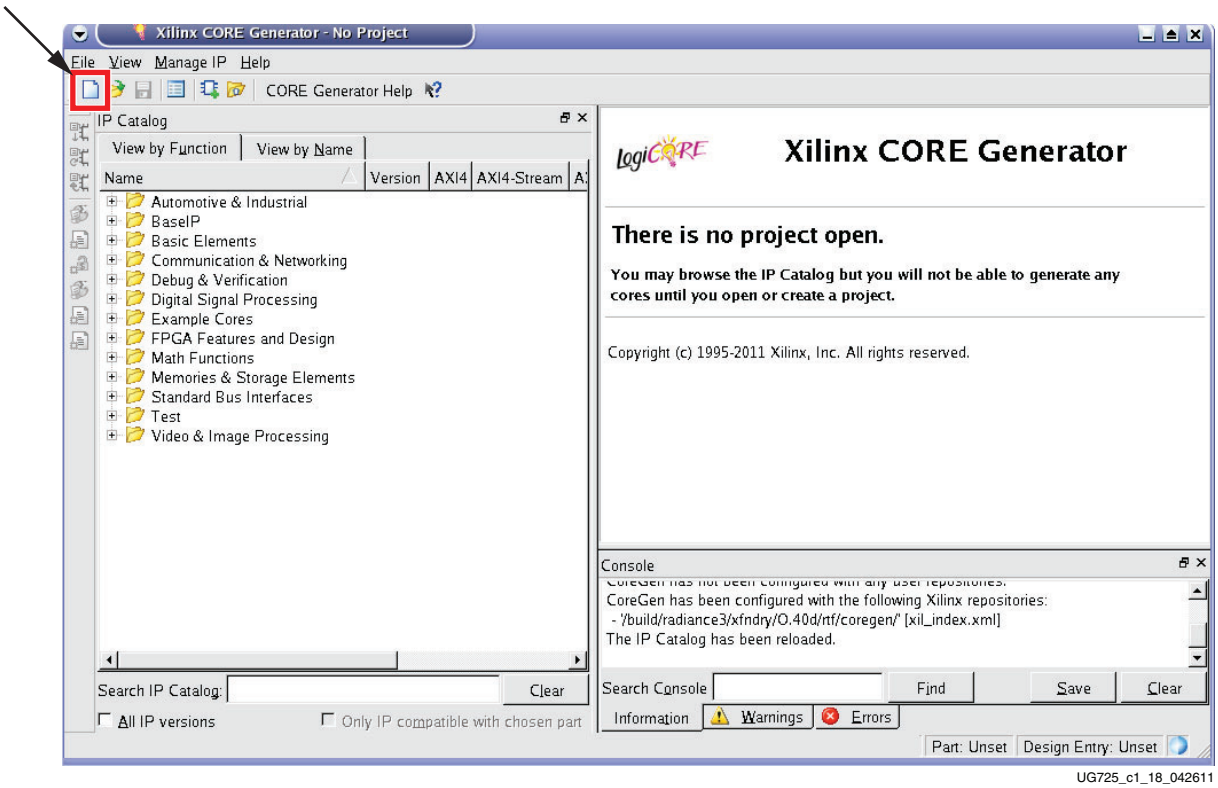


Figure 1-19: **Open New Project**

5. Name the project `coregen_q112.cgp` and click **Save** (Figure 1-20).

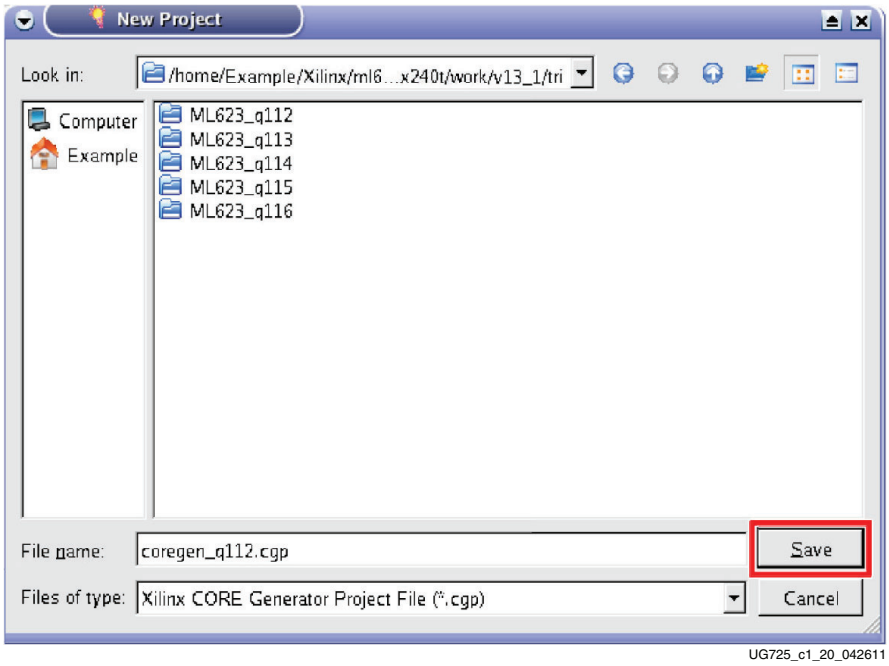


Figure 1-20: **Save New Project**

6. In the Project Options window, under **Part,** select the parameters listed here:
   - Family: **Virtex6**
   - Device: **xc6vlx240t**
   - Package: **ff1156**
   - Speed Grade: **-2**

   Figure 1-21 shows the correct settings.

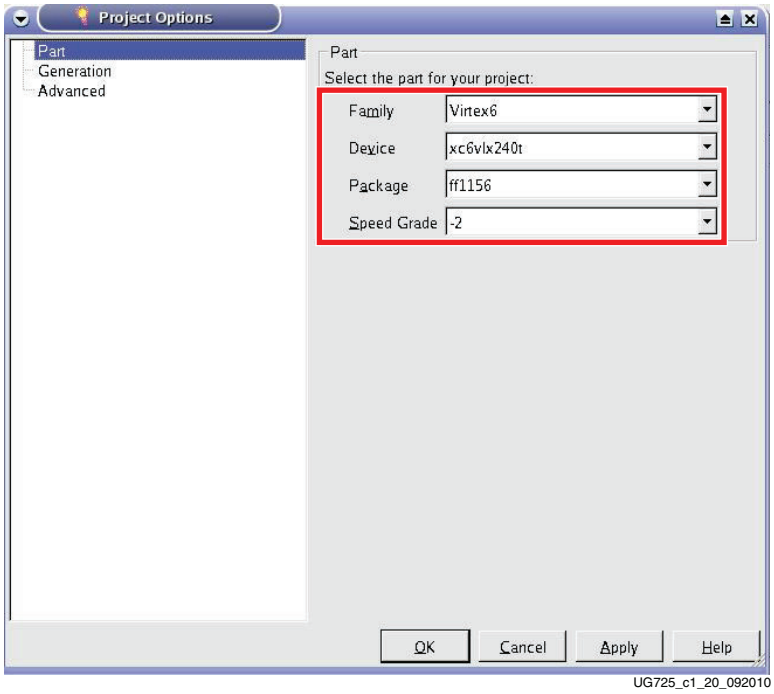


Figure 1-21: **Part Options**

7. In the Project Options window, click **Generation** and select **Verilog** for **Design Entry**, select **Structural** for **Preferred Simulation Model**, and uncheck the box for **ASY Symbol File**. Leave the other settings unchanged. Figure 1-22 shows the correct settings.

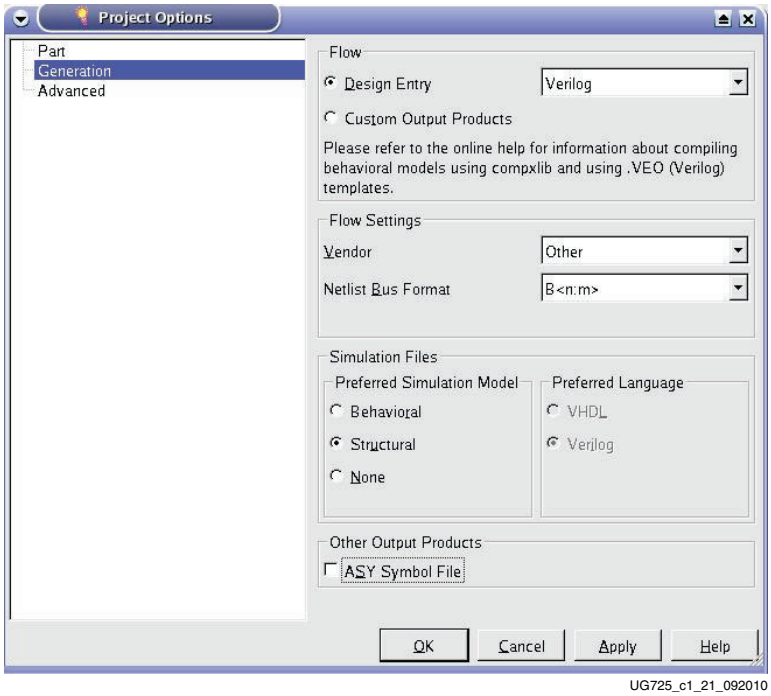


Figure 1-22: **Generation Options**

8. In the Project Options window, under **Advanced**, leave all settings unchanged. Figure 1-23 shows the correct settings.

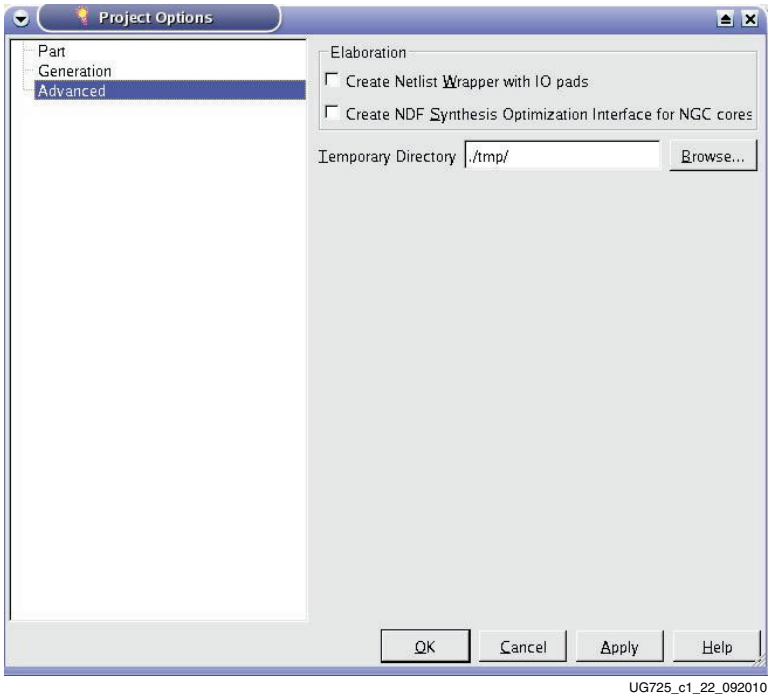


Figure 1-23: **Advanced Options**

9. Click **OK** to close the Project Options window.
10. In the Xilinx Core Generator window under **IP Catalog** select:

    Debug & Verification →

    ChipScope Pro →

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

Double-click the selected core as shown in Figure 1-24.

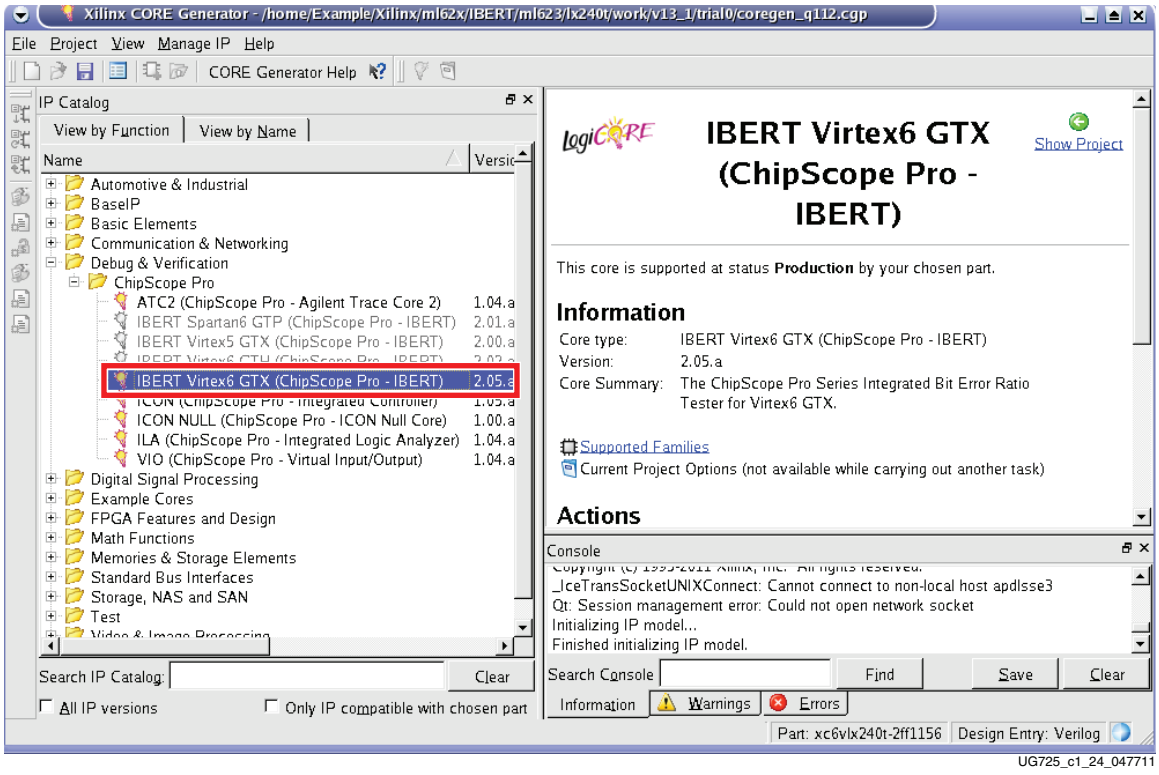


Figure 1-24: **Select IP Core**

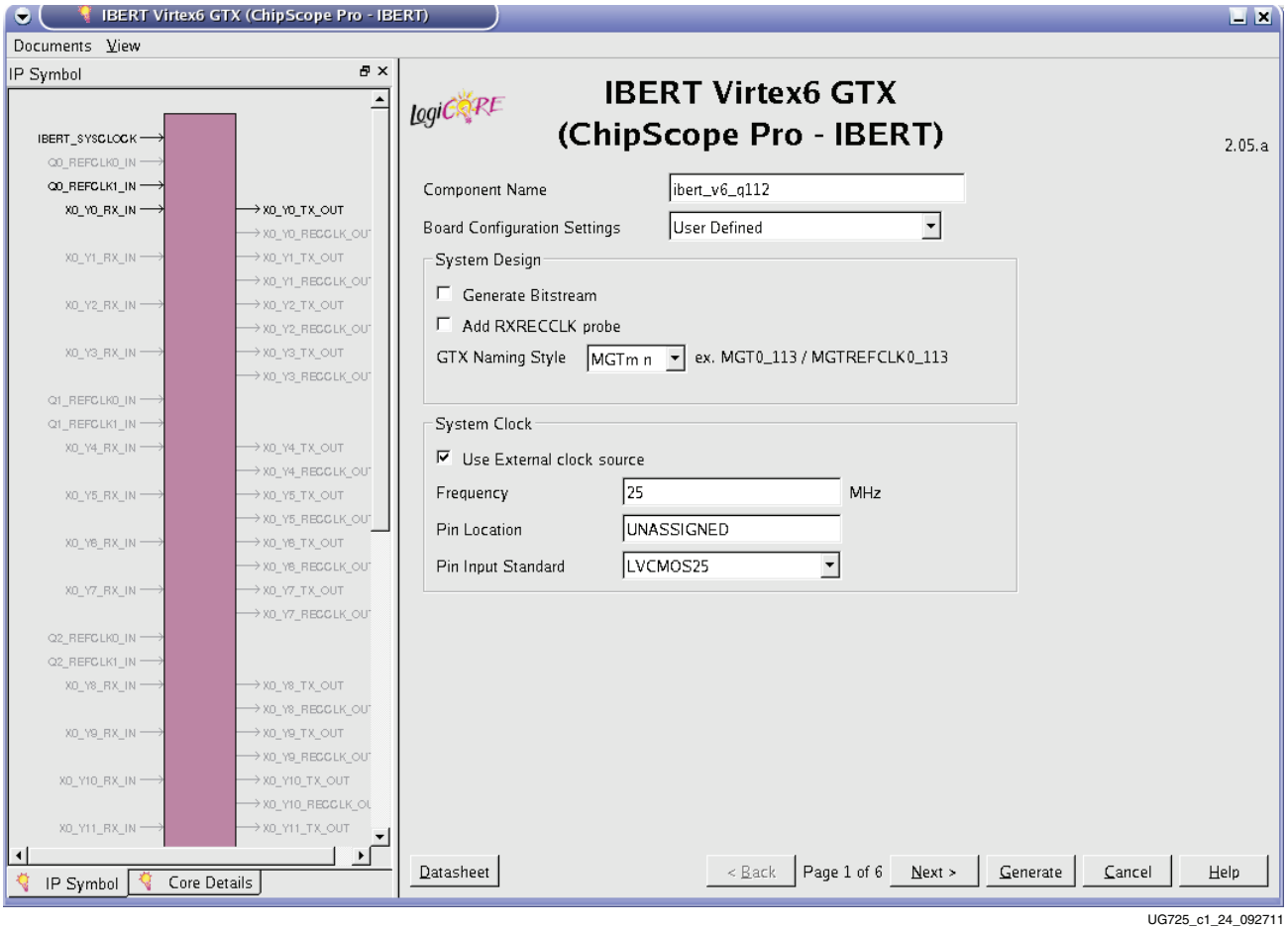
11. After page 1 of the IP customization window appears, for "Component Name" type in **ibert_v6_q112**, under System Design uncheck the box for "Generate Bitstream", and change the "GTX Naming Style" to **MGTm_n**.

    Also select **Use External Clock Source** with these parameters:

    - Frequency: **25 MHz**
    - Location: **UNASSIGNED**
    - Input Standard: **LVCMOS25**

    Figure 1-25 shows the correct settings.

    After entering the changes to page 1, click **Next >** to continue to page 2.



UG725_c1_24_092711

Figure 1-25: **IP Customization, Page 1**

12. After page 2 of the IP customization window appears, edit the fields using these values:
    - Max Rate (Gbps): **6.5**
    - REFCLK (MHz): **162.50**
    - GTX count: **4**

    Figure 1-26 shows the correct settings.

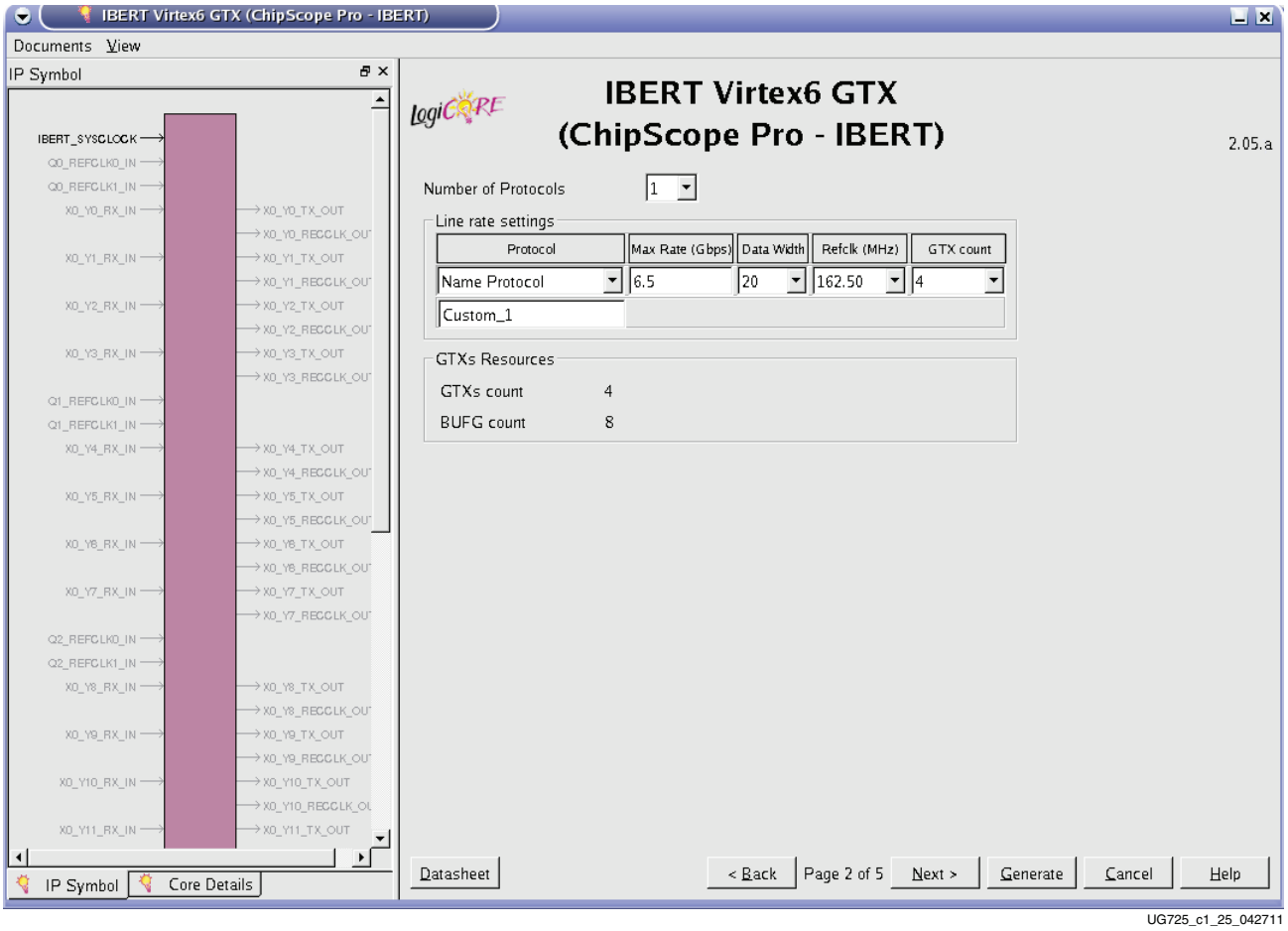


Figure 1-26: **IP Customization, Page 2**

After entering the changes to page 2, click **Next >** to continue to page 3.

13. After page 3 of the IP customization window appears, under "Protocol Selected" select "Custom 1 / 6.5 Gbps" for all four transceivers in GTX quad Q112. Figure 1-27 shows the correct settings.

    Click **Next >** to continue to page 4.

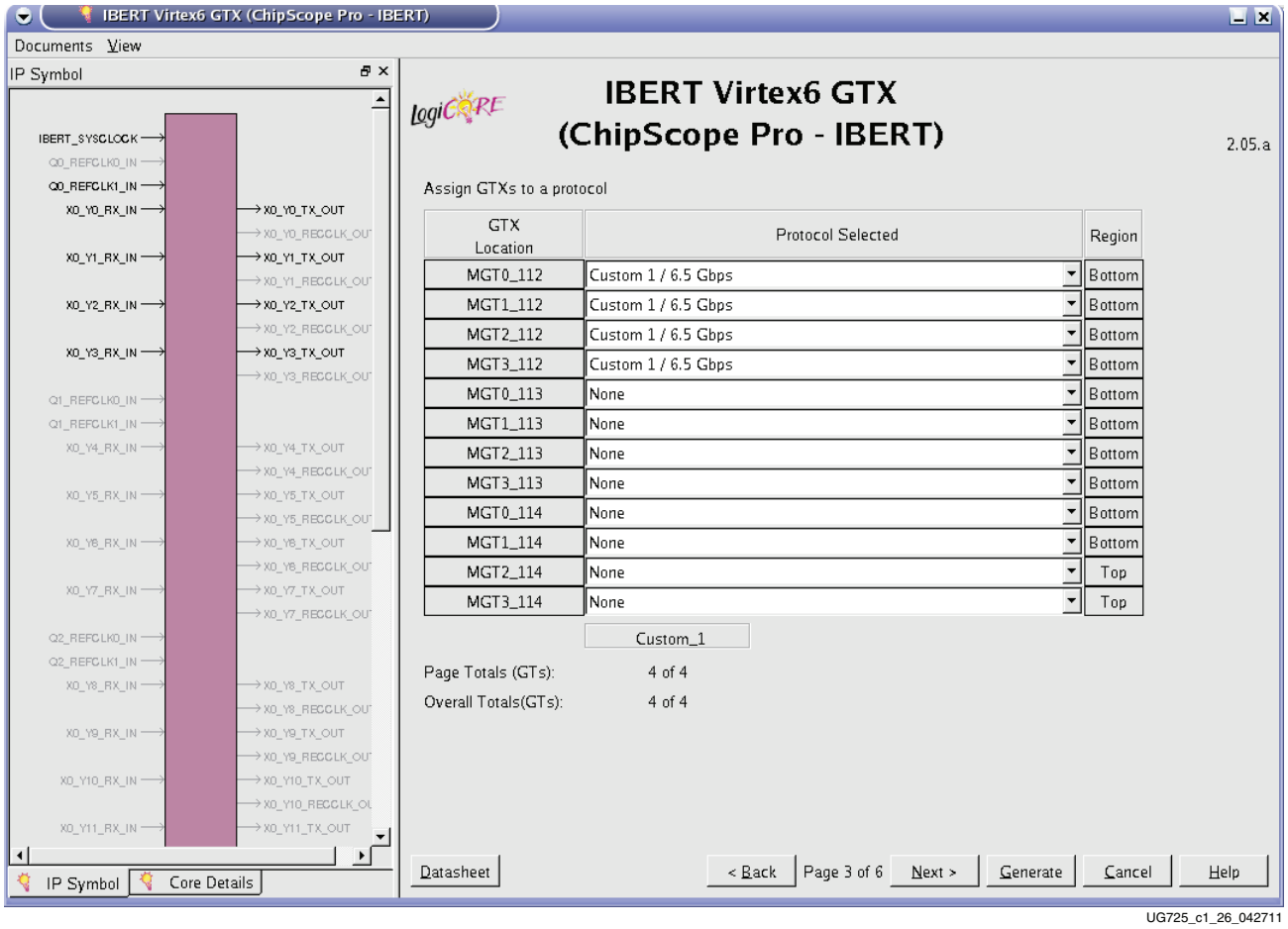


Figure 1-27: **IP Customization, Page 3**

14. Leave page 4 settings as they are. Figure 1-28 shows the correct settings. After entering the changes to page 4, Click **Next >**to continue to page 5.

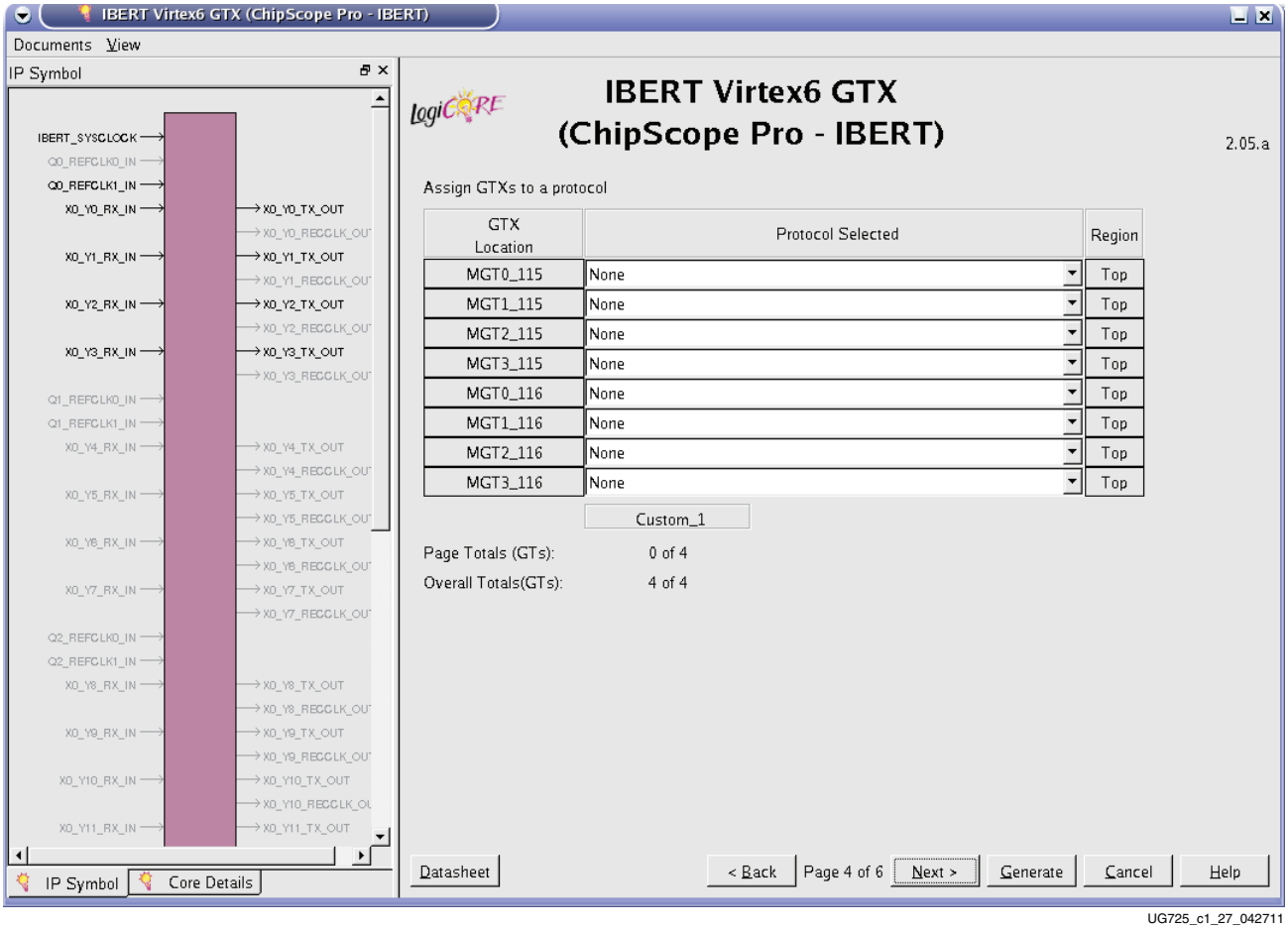


Figure 1-28: **IP Customization, Page 4**

15. After page 5 of the IP customization window appears, select the dedicated reference clocks for the GTX quad of interest (i.e., for the Q112 quad select the 112 reference clocks). For the top two transceivers (MGT0/1) select MGTREFCLK0, for the bottom two transceivers (MGT2/3) select MGTREFCLK1 (Figure 1-29).

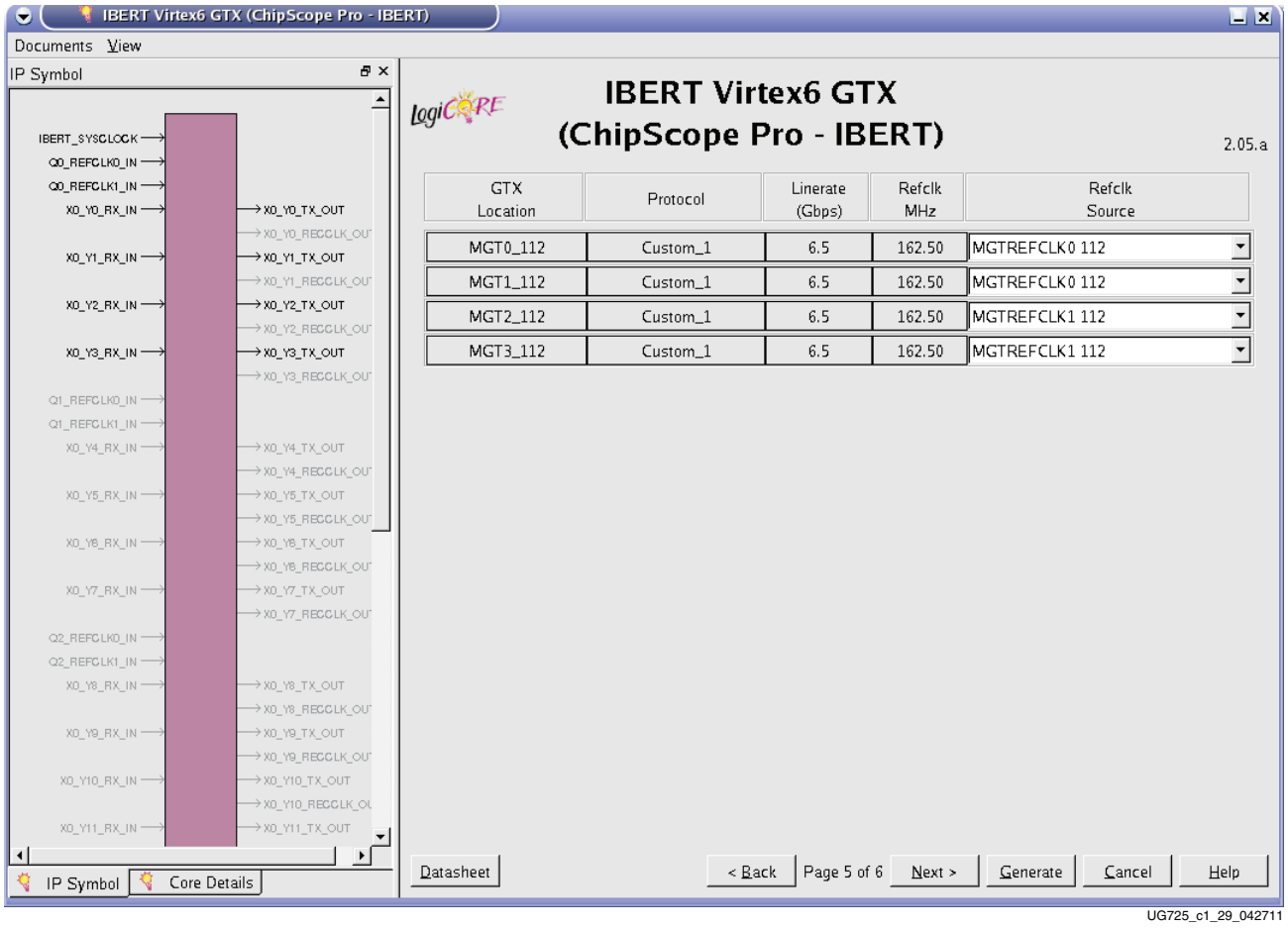


Figure 1-29: **IP Customization, Page 5**

16. After page 6 of the IP customization window appears, review the IBERT Design Summary and click **Generate** (Figure 1-30).

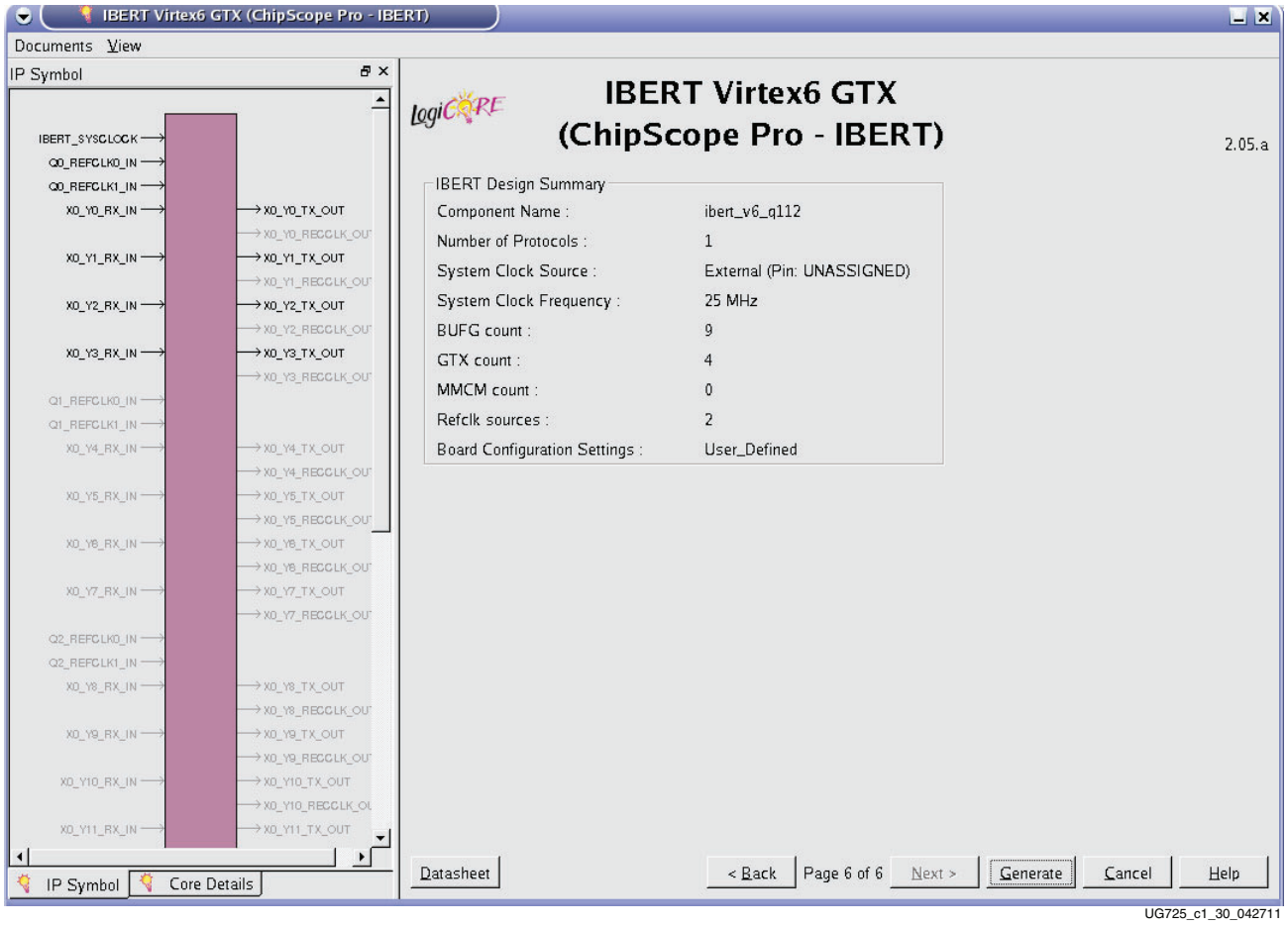


Figure 1-30: **IBERT Design Summary**

17. A readme window for the **ibert_v6_q112** core opens after core generation completes (Figure 1-31). Review the list of files created and click **Close** when finished. Close the Core Generator application as well.

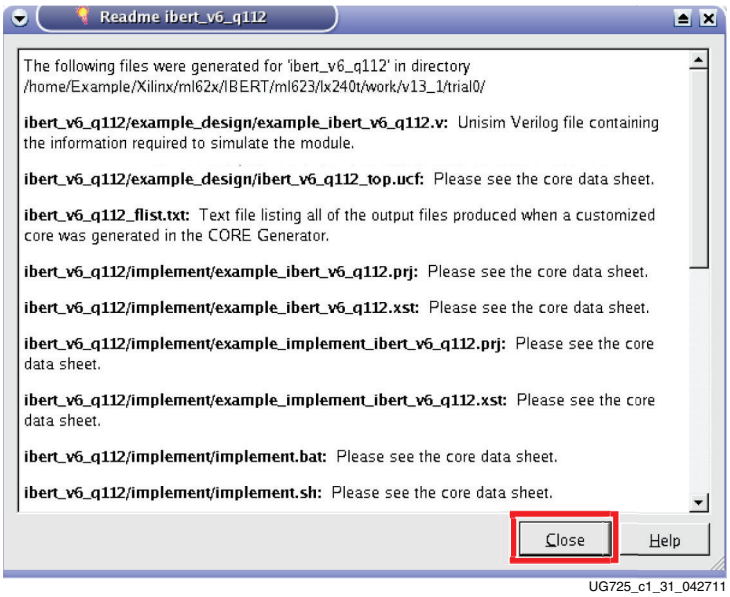


Figure 1-31: **IBERT Core Readme**

18. After the module has been compiled, copy the `ibert_v6_q112.ngc` file to the working directory. In a Linux command window, execute the following command from the top-level directory:

    % cp ibert_v6_q112.ngc ML623_q112/par

19. The `ibert_v6_q112_top.ucf` file needs to be supplemented with system clock, I2C, and SuperClock-2 pin locations. This can most easily be done by concatenating the aforementioned file with the `sysclock_i2c_sclk2_pins.ucf` file provided. Within a Linux command window, and once again from the top-level directory, execute the following command:

    % cat ibert_v6_q112/example_design/ibert_v6_q112_top.ucf ML623_q112/src/sysclk_i2c_sclk2_pins.ucf > ibert_v6_q112/new.ucf

20. Copy the new file over to the source directory:

    % cp ibert_v6_q112/new.ucf ML623_q112/src/ibert_v6_q112_top.ucf

21. Using a text editor, open the `ibert_v6_q112_top.ucf` file created in the previous step and delete the line:

    ```
    NET "IBERT_SYSCLOCK_P_IPAD" IOSTANDARD = LVCMOS25;
    ```

22. Copy the top-level module to the `ML623_q112/src` directory by executing the command:

    % cp ibert_v6_q112/example_design/example_ibert_v6_q112.v ML623_q112/src

23. Using a text editor, open the `example_ibert_v6_q112.v` file in the `src` directory from the previous step.

24. At the very top of the file, locate the very last output declaration for the IBERT module (“`output X0Y0_RXRECCLK_N_OPAD`”), add a comma to the end of this line, and hit **Enter** to create a new line immediately below.

25. At the start of the new line, add the following text to assign user ports for the SuperClock-2 module and the I2C bus:

```
//User Ports
output  [5:0] SCLK_OUT,
input   [8:0] SCLK_IN,
inout         I2C_SDA,
inout         I2C_SCL
```

26. At the end of the local wire declarations, enter a new line and add the following user signal and MMCM input/output declarations:

```
//User Signals
wire   [5:0]  sclk_out_w;
wire   [8:0]  sclk_in_w;
wire   [35:0] control1;
//MMCM inputs & outputs
wire sysclk_i;
wire mmcm_fb;
wire sysclk200_m;
wire sysclk200_g;
wire sysclk25_m;
```

27. Open the file `ibert_sclk2_link.dat`, select all content inside the file, copy the selection, and paste everything in a new line immediately above the IBERT Core Wrapper Instance within the `example_ibert_v6_q112.v` file.
28. Locate and delete the line: `assign ibert_sysclock = IBERT_SYSCLOCK_P_IPAD;`
29. The module is now ready to be used in your design. Change into the par directory by typing the following command:

    % cd ML623_q112/par

30. Run the implement script by typing the following:

    % implement.sh

31. The resulting bitstream, `example_ibert_v6_q112_top.bit`, will be located in the `ML623_q112/par/results` directory.

References

UG029, *ChipScope Pro Software and Cores User Guide*

UG724, *ML623 Virtex-6 FPGA GTX Transceiver Characterization Board User Guide*.

UG770, *HW-CLK-101-SCLK2 SuperClock-2 Module User Guide*

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: http:\\www.xilinx.com\ehs\weee.htm.

XILINX®

Appendix A

SMA Connections for Quads Q113-Q116

Connecting the GTX Transceivers and Reference Clocks, page 7 describes how to connect the GTX transceivers and reference clocks to run the IBERT demonstration for Quad Q1 To run the IBERT demonstration for Quad Q113, Q114, Q115, or Q116, follow the procedure from the start of the aforementioned section substituting the connections for the Quad of interest as provided by the tables listed in Q113, Q114, Q115, or Q116.

Q113

Use the connections in Table A-1 and Table A-2 to run the IBERT demonstration with `ml623_q113.cpj`.

Table A-1: **Q113 Reference Clock Connections**

Source		Destination	
SuperClock-2 Module		ML623 Board	
Net Name	SMA Connector	Net Name	SMA Connector
CKOUT1_P	J5	113_REFCLK0_P	J70
CKOUT1_N	J6	113_REFCLK0_N	J61
CKOUT2_P	J7	113_REFCLK1_P	J72
CKOUT2_N	J8	113_REFCLK1_N	J71

Table A-2: **Q113 TX/RX Connections**

Transmitter		Receiver	
Net Name	SMA Connector	Net Name	SMA Connector
113_TX0_P	J67	113_RX0_P	J68
113_TX0_N	J66	113_RX0_N	J69
113_TX1_P	J63	113_RX1_P	J65
113_TX1_N	J62	113_RX1_N	J64
113_TX2_P	J77	113_RX2_P	J79
113_TX2_N	J76	113_RX2_N	J78
113_TX3_P	J74	113_RX3_P	J80
113_TX3_N	J73	113_RX3_N	J75

Q114

Use the connections in Table A-3 and Table A-4 to run the IBERT demonstration with `ml623_q114.cpj`.

Table A-3: **Q114 Reference Clock Connections**

Source		Destination	
SuperClock-2 Module		ML623 Board	
Net Name	SMA Connector	Net Name	SMA Connector
CKOUT1_P	J5	114_REFCLK0_P	J90
CKOUT1_N	J6	114_REFCLK0_N	J81
CKOUT2_P	J7	114_REFCLK1_P	J92
CKOUT2_N	J8	114_REFCLK1_N	J91

Table A-4: **Q114 TX/RX Connections**

Transmitter		Receiver	
Net Name	SMA Connector	Net Name	SMA Connector
114_TX0_P	J87	114_RX0_P	J88
114_TX0_N	J86	114_RX0_N	J89
114_TX1_P	J83	114_RX1_P	J85
114_TX1_N	J82	114_RX1_N	J84
114_TX2_P	J99	114_RX2_P	J103
114_TX2_N	J97	114_RX2_N	J100
114_TX3_P	J94	114_RX3_P	J96
114_TX3_N	J93	114_RX3_N	J95

Q115

Use the connections in Table A-5 and Table A-6 to run the IBERT demonstration with `ml623_q115.cpj`.

Table A-5: **Q115 Reference Clock Connections**

Source		Destination	
SuperClock-2 Module		ML623 Board	
Net Name	SMA Connector	Net Name	SMA Connector
CKOUT1_P	J5	115_REFCLK0_P	J125
CKOUT1_N	J6	115_REFCLK0_N	J124
CKOUT2_P	J7	115_REFCLK1_P	J123
CKOUT2_N	J8	115_REFCLK1_N	J106

Table A-6: **Q115 TX/RX Connections**

Transmitter		Receiver	
Net Name	SMA Connector	Net Name	SMA Connector
115_TX0_P	J134	115_RX0_P	J136
115_TX0_N	J133	115_RX0_N	J135
115_TX1_P	J127	115_RX1_P	J130
115_TX1_N	J126	115_RX1_N	J128
115_TX2_P	J118	115_RX2_P	J120
115_TX2_N	J117	115_RX2_N	J121
115_TX3_P	J111	115_RX3_P	J116
115_TX3_N	J110	115_RX3_N	J114

Q116

Use the connections in Table A-7 and Table A-8 to run the IBERT demonstration with `ml623_q116.cpj`.

Table A-7: **Q116 Reference Clock Connections**

Source		Destination	
SuperClock-2 Module		ML623 Board	
Net Name	SMA Connector	Net Name	SMA Connector
CKOUT1_P	J5	116_REFCLK0_P	J156
CKOUT1_N	J6	116_REFCLK0_N	J148
CKOUT2_P	J7	116_REFCLK1_P	J138
CKOUT2_N	J8	116_REFCLK1_N	J137

Table A-8: **Q116 TX/RX Connections**

Transmitter		Receiver	
Net Name	SMA Connector	Net Name	SMA Connector
116_TX0_P	J154	116_RX0_P	J157
116_TX0_N	J153	116_RX0_N	J155
116_TX1_P	J150	116_RX1_P	J152
116_TX1_N	J149	116_RX1_N	J151
116_TX2_P	J145	116_RX2_P	J147
116_TX2_N	J144	116_RX2_N	J146
116_TX3_P	J140	116_RX3_P	J143
116_TX3_N	J139	116_RX3_N	J142