Virtex-5 FPGA RocketIO GTX Transceiver Clock Correction Module

XTP037 (v1.0) March 10, 2009

Clock Correction Module Overview

This document assumes familiarity with UG198, *Virtex-5 FPGA RocketIO GTX Transceiver User Guide*.

The clock correction (CC) module is a design that connects to the Virtex®-5 FPGA RocketIO™ GTX transceiver. The design implements clock tolerance compensation using an elastic buffer in FPGA logic. This module must be used when clock tolerance compensation is required and the clock correction sequence is one byte.

The CC module performs clock tolerance compensation by skipping or adding 1 byte clock correction characters from the data stream. This prevents the elastic buffer inside the CC module from overflowing or underflowing when the bit rate is different at the two ends of a link by up to 200 parts per million (PPM). Figure 1 shows a port map of the CC module.

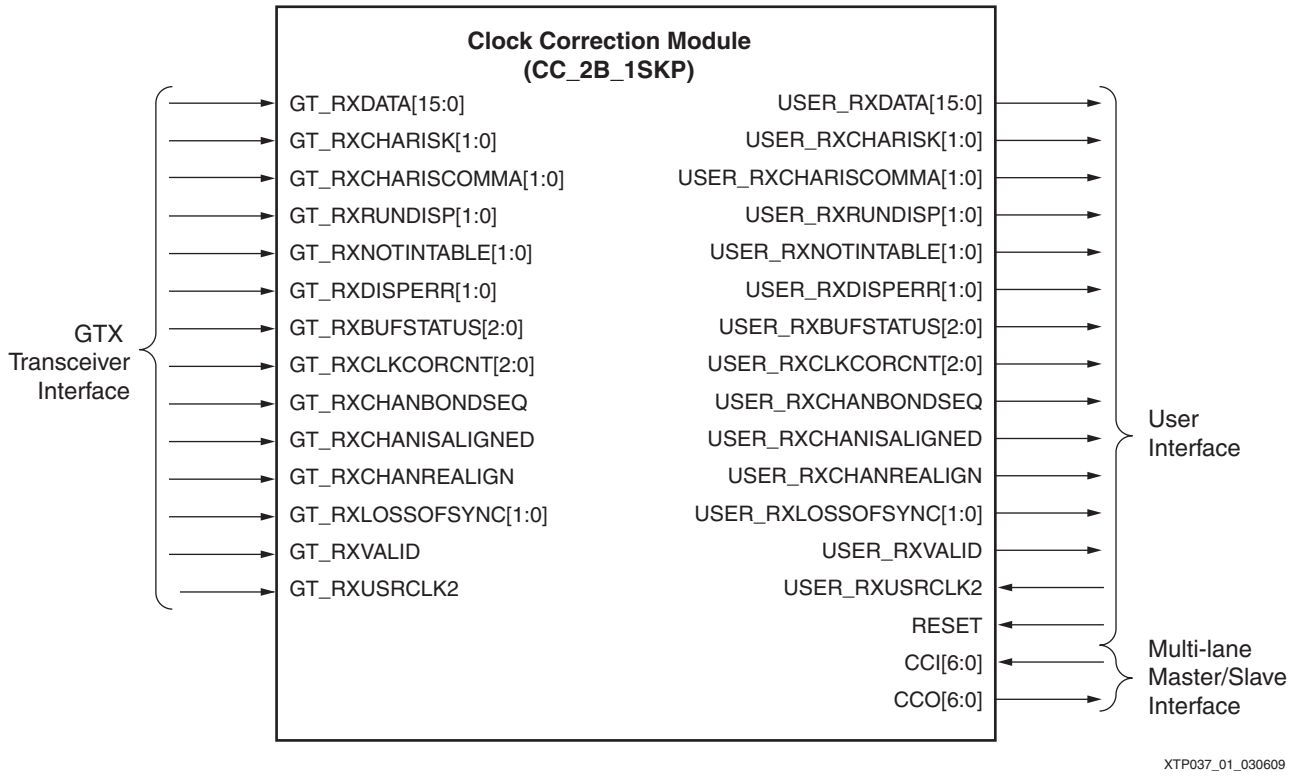

Figure 1: **Port Map of the CC Module**

CC Module Interfaces

The CC module uses three interfaces: the GTX transceiver interface, the user interface, and the multi-lane master/slave interface.

GTX Transceiver Interface

The GTX transceiver interface connects to the data and control signals of the GTX transceiver.

User Interface

The CC module takes incoming data and control signals of the GTX transceiver and performs clock tolerance compensation. Ports beginning with GT_ (for example, GT_RXDATA), are presented to the user interface as ports beginning with USER_ (for example, USER_RXDATA) after being retimed by the CC module. All USER_* ports are synchronous to the CC module USER_RXUSRCLK2 clock domain.

Note: When referring to these ports generically, this document uses the terms GT_* and USER_*.

The USER_RXUSRCLK2 port of the CC module must be connected to the same clock as the user logic. The GT_RXUSRCLK2 and USER_RXUSRCLK2 ports must have the same nominal frequency with up to ±100 PPM difference from nominal. This is the same as an absolute difference of 200 PPM between the two clock rates.

The RESET port must be held High until valid and stable clocks are input to the GT_RXUSRCLK2 and USER_RXUSRCLK2 ports.

Multi-Lane Master/Slave Interface

This interface consists of CCI and CCO. The use models and connectivity for this interface are described in the "Use Models" section.

Parameter/ Generic List

Table 1 lists the type and bit widths for each parameter used by the CC module interface. The description column summarizes the parameter function.

Table 1: **Parameter/Generic List**

Parameter	Type	Width (Bits)	Description
CC_CHAR	Bit Vector	8	Clock correction character (also known as skip character). The default value is `0x1C`.
ALIGN_CHAR	Bit Vector	8	Channel bonding character in the 8B domain used to detect alignment between the master and slave of a channel bonded interface when ALIGN_PARALLEL_CHECK is `1`. ALIGN_CHAR is ignored when ALIGN_PARALLEL_CHECK is set to 0. The default value is `0x7C`.

Table 1: **Parameter/Generic List *(Cont'd)***

Parameter	Type	Width (Bits)	Description
ALIGN_PARALLEL_CHECK	Bit	1	Enables parallel data alignment check between master and slave CC modules of a channel bonded interface. Alignment check is performed in each slave on parallel data (USER_RXDATA). The default value is `1` (enabled). Value = `1` (enabled): • A parallel data alignment check is performed on slave. USER_RXCHANISALIGNED is asserted when GT_RXCHANISALIGNED is asserted and alignment is detected between master and slave. • ALIGN_CHAR is used as channel bonding character. Value = `0` (disabled): • The parallel data alignment check is disabled. USER_RXCHANISALIGNED is driven by GT_RXCHANISALIGNED through a retiming circuit.
USE_AUTORECOVER	Bit	1	This parameter must be set to `1`. The default value is `1`.
CHAN_BOND_MODE	String		Channel bond mode. Set to the same value as the GTX transceiver CHAN_BOND_MODE attribute. The default string is OFF. Possible values: • OFF: Single-lane interface • MASTER: Multi-lane interface master • SLAVE: Multi-lane interface slave
FIFO_ALMOST_EMPTY_OFFSET(1)	Integer		Sets the almost empty threshold of the FIFO. The default value is 5. This is the minimum value allowed.
FIFO_ALMOST_FULL_OFFSET(1)	Integer		Sets the almost full threshold of the FIFO as measured from a full condition of 511. The default value is 498. This is the maximum value allowed.

Notes:

1. The default values of FIFO_ALMOST_EMPTY_OFFSET and FIFO_ALMOST_FULL_OFFSET were used during hardware validation of the CC module. Changing these values is not recommended without proper analysis and further testing.

Port List

Table 2 lists the direction, bit width, and clock domain used by the CC module interface.

Table 2: **Port List**

Port	Direction	Width (Bits)	Clock Domain	Description
GT_RXDATA	Input	16	GT_RXUSRCLK2	GTX transceiver side RXDATA.
GT_RXCHARISK	Input	2	GT_RXUSRCLK2	GTX transceiver side RXCHARISK.
GT_RXCHARISCOMMA	Input	2	GT_RXUSRCLK2	GTX transceiver side RXCHARISCOMMA.
GT_RXRUNDISP	Input	2	GT_RXUSRCLK2	GTX transceiver side RXRUNDISP.
GT_RXNOTINTABLE	Input	2	GT_RXUSRCLK2	GTX transceiver side RXNOTINTABLE.
GT_RXDISPERR	Input	2	GT_RXUSRCLK2	GTX transceiver side RXDISPERR.
GT_RXBUFSTATUS	Input	3	GT_RXUSRCLK2	GTX transceiver side RXBUFSTATUS.

Table 2: **Port List *(Cont'd)***

Port	Direction	Width (Bits)	Clock Domain	Description
GT_RXCLKCORCNT	Input	3	GT_RXUSRCLK2	GTX transceiver side RXCLKCORCNT. Because clock correction must be disabled when using the CC module, the status from the GTX transceiver side always indicates `000`.
GT_RXCHANBONDSEQ	Input	1	GT_RXUSRCLK2	GTX transceiver side RXCHANBONDSEQ.
GT_RXCHANISALIGNED	Input	1	GT_RXUSRCLK2	GTX transceiver side RXCHANISALIGNED.
GT_RXCHANREALIGN	Input	1	GT_RXUSRCLK2	GTX transceiver side RXCHANREALIGN.
GT_RXLOSSOFSYNC	Input	2	GT_RXUSRCLK2	GTX transceiver side RXLOSSOFSYNC.
GT_RXVALID	Input	1	GT_RXUSRCLK2	GTX transceiver side RXVALID.
GT_RXUSRCLK2	Input	1	-	Connect to GTX RXUSRCLK2. See "CC Module Clocking" section for more details.
USER_RXDATA	Output	16	USER_RXUSRCLK2	User side RXDATA. GT_RXDATA is passed through to this port.
USER_RXCHARISK	Output	2	USER_RXUSRCLK2	User side RXCHARISK. GT_RXCHARISK is passed through to this port.
USER_RXCHARISCOMMA	Output	2	USER_RXUSRCLK2	User side RXCHARISCOMMA. GT_RXCHARISCOMMA is passed through to this port.
USER_RXRUNDISP	Output	2	USER_RXUSRCLK2	User side RXRUNDISP. GT_RXRUNDISP is passed through to this port.
USER_RXNOTINTABLE	Output	2	USER_RXUSRCLK2	User side RXNOTINTABLE. GT_RXNOTINTABLE is passed through to this port.
USER_RXDISPERR	Output	2	USER_RXUSRCLK2	User side RXDISPERR. GT_RXDISPERR is passed through to this port.
USER_RXBUFSTATUS	Output	3	USER_RXUSRCLK2	User side RXBUFSTATUS. Values output on this port indicate these status conditions: • `000`: Nominal condition • `100`: GTX RX elastic buffer error • `101`: CC module buffer underflow • `110`: CC module buffer overflow
USER_RXCLKCORCNT	Output	3	USER_RXUSRCLK2	User side RXCLKCORCNT. Values output on this port indicate these status conditions: • `000`: No clock correction • `001`: 1 clock correction sequence was skipped • `111`: 1 clock correction sequence was added
USER_RXCHANBONDSEQ	Output	1	USER_RXUSRCLK2	User side RXCHANBONDSEQ. GT_RXCHANBONDSEQ is passed through to this port. This port might be asserted for one extra cycle.
USER_RXCHANISALIGNED	Output	1	USER_RXUSRCLK2	User side RXCHANISALIGNED. GT_RXCHANISALIGNED is passed through to this port.
USER_RXCHANREALIGN	Output	1	USER_RXUSRCLK2	User side RXCHANREALIGN. GT_RXCHANREALIGN is passed through to this port. This port might be asserted one cycle early or late.

Table 2: **Port List *(Cont'd)***

Port	Direction	Width (Bits)	Clock Domain	Description
USER_RXLOSSOFSYNC	Output	2	USER_RXUSRCLK2	User side RXLOSSOFSYNC. GT_RXLOSSOFSYNC is passed through to this port. This port might change one cycle early or late.
USER_RXVALID	Output	1	USER_RXUSRCLK2	User side RXVALID. GT_RXVALID is passed through to this port. This port might be asserted one cycle early or late.
USER_RXUSRCLK2	Input	1	-	Must be connected to the clock used by the user logic. See "CC Module Clocking" section for more details.
RESET	Input	1	Asynchronous	System reset (active High). • Master CC module (CHAN_BOND_MODE is set to MASTER or OFF). It must be held High until valid clocks are input to both the GT_RXUSRCLK2 and USER_RXUSRCLK2 ports. • Slave CC module (CHAN_BOND_MODE set to SLAVE). Can be tied Low or driven from the same reset signal as master clock correction module.
CCI	Input	7	-	CC module control. • Master CC module - connect to 0000000. • Slave CC module - connect to CCO port of the master clock correction module.
CCO	Output	7	-	CC module control. • Master CC module - connect to CCI ports of all slave CC modules. ♦ Bit 0: FIFO almost empty pulse. ♦ Bit 1: Reset synchronized to USER_RXUSRCLK2. ♦ Bit 2: FIFO almost full pulse. ♦ Bit 3: Reset synchronized to GT_RXUSRCLK2. ♦ Bit 4: Channel bonding character detected in the lower byte. ♦ Bit 5: Channel bonding character detected in the upper byte. ♦ Bit 6: FIFO almost empty. • Slave CC module - leave floating.

Software Requirements

The CC module has been verified to simulate, synthesize, and function properly with the following tools:

- Simulation: ModelSim SE 6.4c
- Synthesis: XST 10.1.3; Synplify Pro 9.6.2
- Implementation: ISE® Design Suite version 10.1.3

Use Models

This section details these three different use models of the CC module:

- CC module clocking
- CC_2B_1SKP Single-Lane
- CC_2B_1SKP Multi-Lane with the GTX RX channel bonding enabled

The CC module should be treated as if it were part of the physical coding sublayer (PCS) of the transceiver. The RESET port of the CC module must be asserted whenever the RXRESET or RXBUFRESET ports of the GTX transceiver are asserted. User logic must connect to the USER_* versions of the CC module ports. The GT_* ports are reserved for connection between the GTX_DUAL tile and the CC module.

The clock correction feature of the GTX transceiver must be disabled when using the CC_2B_1SKP module. To disable the clock correction feature in the GTX transceiver, set the attributes as described in Table 3.

Table 3: **Attribute Settings to Disable the Clock Correction Feature of GTX Transceivers**

Attribute	Value
CLK_CORRECT_USE_0/1	FALSE
CLK_COR_SEQ_1_1_0/1	10'b0100000000
CLK_COR_SEQ_1_ENABLE_0/1	4'b1111
CLK_COR_SEQ_2_1_0 /1	10'b0100000000
CLK_COR_SEQ_2_ENABLE_0/1	4'b1111

To instantiate the CC module in user design:

- Disable clock correction in the transceiver using the attribute settings in Table 3.
- Set the CHAN_BOND_MODE parameter to match the value of the attached GTX_DUAL CHAN_BOND_MODE attribute.
- Drive the RESET port High on the CC module (CHAN_BOND_MODE is set to either MASTER or OFF) until valid clocks are input to both the GT_RXUSRCLK2 and USER_RXUSRCLK2 ports. Examples of conditions for which RESET must be asserted High include:
  - When RESETDONE of the GTX transceiver is Low.
  - When the LOCK of a DCM/PLL is Low (if a DCM or PLL is used to generate RXUSRCLK2).
  - When the RXRESET or RXBUFRESET ports of the GTX transceiver are asserted.
- If channel bonding is required:
  - Connect the CCO port of the master CC module to the CCI port of all slave CC modules of the same interface.
  - Tie the CCI port of the master CC module to 0000000.
  - Leave the CCO port of the slave CC module(s) floating.
  - For the slave CC module(s), the RESET port is not used. The RESET port can be left floating. The RESET port on slave modules is completely ignored, so connecting it will not cause harm.
- If channel bonding is not required:
  - Connect the CCI port to 0000000.
  - Leave the CCO port floating.

- Any signal from the GTX transceiver that does not go through the CC module must be synchronized to the CC module USER_RXUSRCLK2 clock domain, which is the same domain on which the user logic is clocked.

CC Module Clocking

Table 4 shows the clocking requirements of the GTX transceiver and the CC module for a 2 byte data width.

The clocking structure is different than when the GTX_DUAL tile provides the clock correction functionality. When the CC module is used, the GTX_DUAL tile clock correction feature must be disabled, and the RXUSRCLK and RXUSRCLK2 ports must be derived from RXRECCLK so that the internal GTX receiver buffer does not underflow or overflow.

The CC module GT_RXUSRCLK2 port must be connected to the GTX RXUSRCLK2 port. The CC module USER_RXUSRCLK2 port must be connected to the clock source that drives the user logic.

Table 4: **USRCLK Connections**

Module	Port	Description
CC Module	GT_RXUSRCLK2	Connects to GTX RXRECCLK.
	USER_RXUSRCLK2	User clock of frequency bit rate/20. In most systems, this port is driven by the same clock as the GTX TXUSRCLK2.
GTX Receiver	RXUSRCLK2	Connects to GTX RXRECCLK.
	RXUSRCLK	Connects to GTX RXRECCLK.

Single-Lane Use Model

Figure 2 shows the connections when the CC_2B_1SKP module is used in a single-lane design.

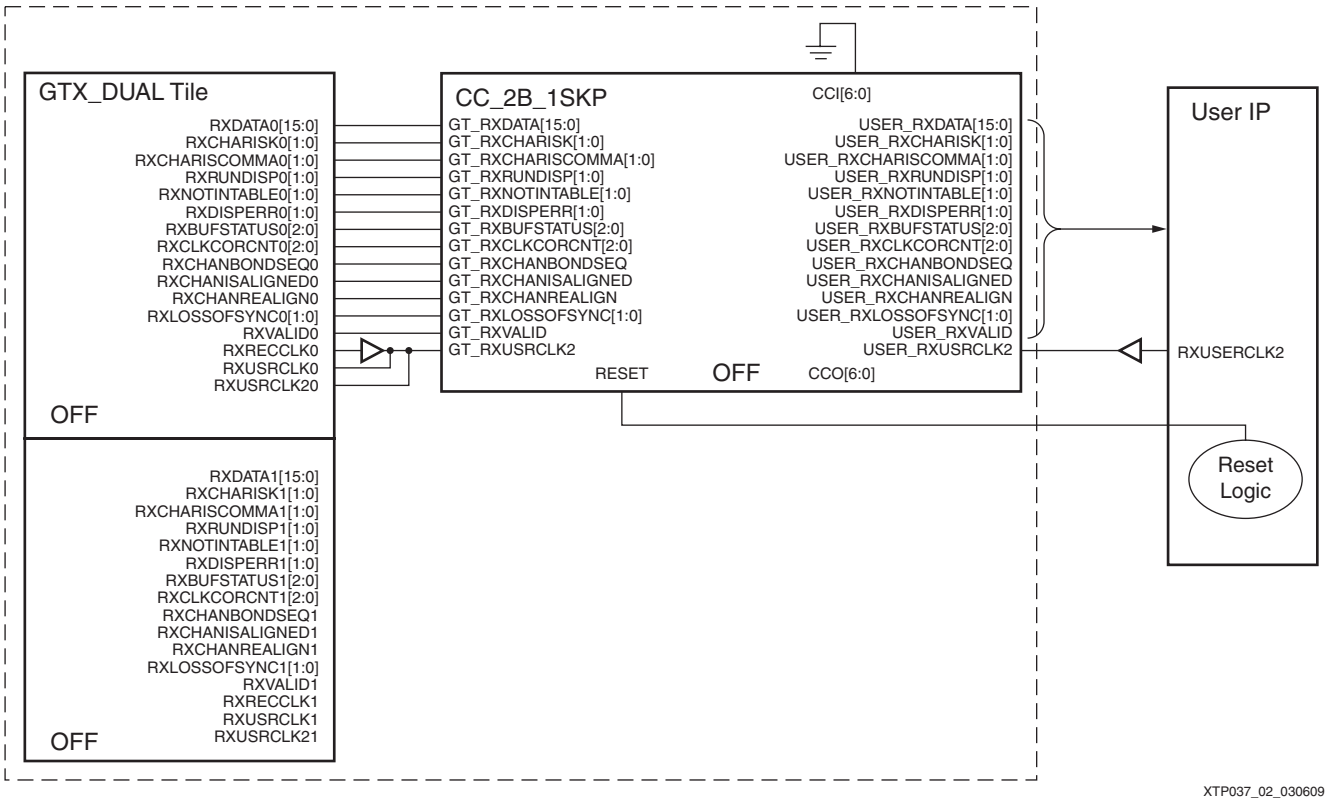

Figure 2: **CC_2B_1SKP Single-Lane Use Model**

Multi-Lane Use Model

Figure 3 shows the connections when the CC_2B_1SKP module is used in a multi-lane design with GTX receiver channel bonding enabled.

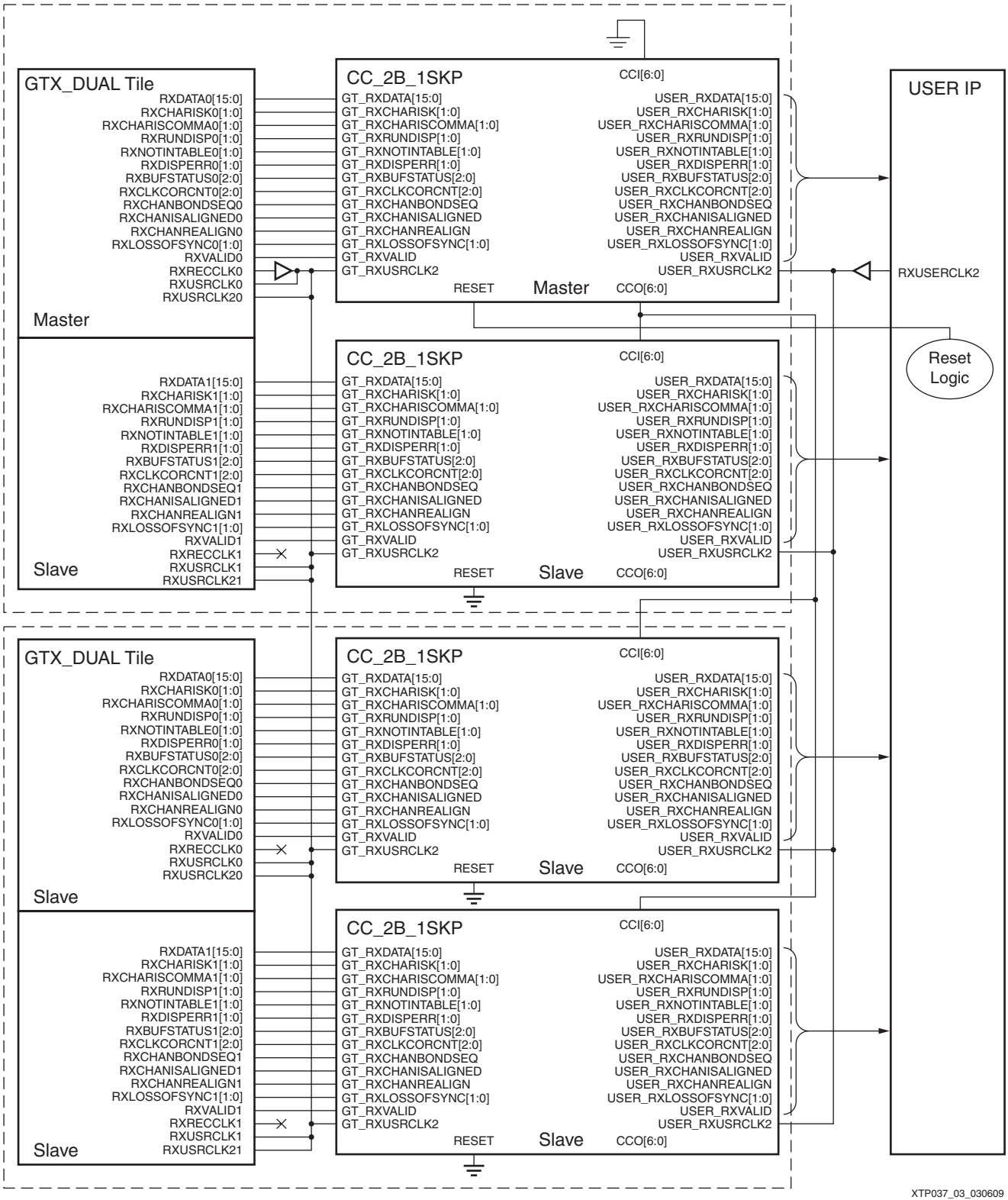

Figure 3: **CC_2B_1SKP Multi-Lane Use Model**

Resource Utilization

Table 5 shows the resource utilization of the CC module as reported by the XST synthesis tool, version 10.1.3.

Table 5: **CC_2B_1SKP Resource Utilization**

CHAN_BOND_MODE	ALIGN_PARALLEL_CHECK	FFs	LUTs	Block RAMs	BUFG or BUFR
OFF	1	230	136	1(1)	1(2)
	0				
MASTER	1	239	154		
	0	237	146		
SLAVE	1	227	152		
	0	224	142		

Notes:

1. Number of FIFO18_36 components. Block RAMs are fundamentally 36 Kb in size. Each FIFO18_36 uses half of the available 36 Kb. The other half can be used as an 18 Kb Block RAM.
2. One global clock resource (BUFR or BUFG) is required per interface. An interface is defined as one or more GTX transceiver bonded together to make up a single-lane or multi-lane interface. Examples are the LogiCORE™ IP XAUI core and the Serial RapidIO solution.

Design Files

The `xtp037.zip` file contains the source code for the clock correction module:

https://secure.xilinx.com/webreg/clickthrough.do?cid=115636

Revision History

The following table shows the revision history for this document.

Date	Version	Description of Revisions
03/10/09	1.0	Initial Xilinx release.

Notice of Disclaimer