RC24596 (W0807-021) July 2, 2008
Electrical Engineering

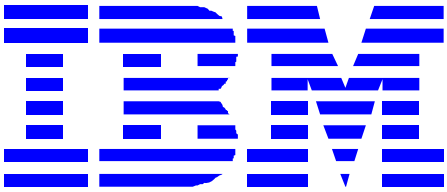
IBM Research Report

Application Note:
FPGA to IBM Power Processor Interface Setup

Ibrahim Ouda
IBM Systems &Technology Group
3605 Highway 52 N
Rochester, MN 55901-1407

Kai Schleupen
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich



Application Note
FPGA to IBM Power Processor Interface Setup

Document Name: PowerXCell_FlexIO_GTP_app_note_rel1.0
Date: June 27, 2008

Owner: Ibrahim Ouda
Prepared by: Ibrahim Ouda and Kai Schleupen

IBM System&Technology Group and
IBM Research

Sites: Rochester, MN and
Yorktown Heights, NY
USA

Printed in the United States of America in June 2008

Table of Contents

1 Abstract

This application note describes the setup within the Xilinx FPGA tools to be able to establish a high-bandwidth interface between an IBM Power Processor and the latest Xilinx FPGAs via an interface implementation from Rambus Inc. It has been currently successfully evaluated and tested on a system prototyping platform up to a data rate of 3Gbit/sec per lane and can be expanded to multiple bytes. It represents one of the fastest coherent/non-coherent processor to FPGA interfaces available in the industry.

2 Introductions

2.1 FlexIO™ [1] - The Industry's Fastest Processor Bus Interface

The Rambus® FlexIO family of processor bus interfaces closes the critical I/O to logic speed performance gap and represents a significant breakthrough in parallel chip-to-chip interface technology. Now this technology is available on FPGAs. This application note describes how to configure Xilinx® Virtex®-5 FPGA to interface to FlexIO.

FlexIO offers data rates of 400MHz to 8 GHz which is up to ten times faster than today's best-of-class processor busses, while reducing overall package, board, and system costs. Optimized for intra-board environments, it provides an ideal low-latency and low-power solution for high-volume, low-cost applications, including processor (e.g. Cell Broadband Engine™ [2] and IBM's PowerXCell™ [3] 8i Processor), chipset and network chip connections for a broad range of applications like High-Performance Computing. For ASIC implementations, the FlexIO processor bus is optionally backwards compatible to existing LVDS-based standards, including HyperTransport™ [4], SPI-4 and RapidIO™ [5], allowing easy integration into next generation products, while providing a path to higher levels of performance. The FPGA implementation of the FlexIO processor bus interface provides a complete physical layer solution for both standards based and proprietary applications. Rambus provides directly both the physical layer and logic layer IP for ASIC applications.

In addition to breakthrough bandwidth, FlexIO processor bus interfaces provide flexibility, simplicity and savings in system design through three innovative building blocks: FlexPhase™ circuit technology is a major departure from traditional circuit technologies. It enables precise, per bit, on-chip alignment of data and clock, eliminating the need for PCB trace length matching and PCB timing constraints. This results in logic systems that are simpler, more compact, lower in cost, and capable of multi-GHz data rates. DRSL (Differential Rambus Signaling Levels) with LVDS (Low Voltage Differential Signaling) enables backward compatibility. DRSL is a differential signaling technology available in bi-directional and uni-directional versions that offers a high-performance, low power, cost-effective solution for transmitting data on and off chip. DRSL uses a variable signal swing as low as 200mV, enabling low-power operation, reduced electromagnetic radiation (EMR) and scalability to lower process voltages in the future. Variable Date Rate (VDR) delivers data rates of 1-to-10 times the speed of the clock, supporting a wide range of system clocks and data rates ranging from 400 MHz - 8 GHz.

The FlexIO family of high performance processor bus interfaces extends the versatility of Rambus'

1 FlexIO is a trademark of Rambus Inc.
2 Cell Broadband Engine is a registered trademark of Sony Computer Entertainment, Inc
3 PowerXCell is a registered trademark of IBM Corp
4 Hypertransport is a registered trademark of the HyperTransport Technology Consortium
5 RapidIO is a registered trademark of the RapidIO Trade Association

logic interface solutions and complements the Rambus(tm) high-performance serial link interface family. FlexIO interfaces are ideal for intra-board applications with shorter-distance (up to 15") and lower-distortion interconnects. In these applications, FlexIO processor busses provide lower power and lower latency than traditional serial link solutions

2.2 Xilinx

Xilinx® Virtex®-5 multi-platform FPGAs offer built-in serial connectivity with two varieties of transceivers. RocketIO™ [6] GTP transceivers found on Virtex-5 LXT and SXT FPGA platform devices support line rates up to 3.75Gpbs with the industry's lowest power consumption. These transceivers support a variety of standards, such as PCIe™ [7], RapidIO®[8], Gigabit Ethernet, XAUI™ [9], InfiniBandt™[10], Fibre Channel™ [11], SATA, HD-SDI, SONET, Interlaken, and SFI-5.

For protocols requiring higher line rates, including future FlexIO interfaces, Virtex-5 FXT FPGA platform devices offer RocketIO GTX transceivers which deliver power-efficient connectivity up to 6.5Gbps. Cross platform pin compatibility eases design upgrades from GTP to GTX transceivers.

The RocketIO GTP transceiver is highly configurable and tightly integrated with the programmable logic resources of the FPGA. Key features include:

- Line rates from 100 Mb/s to 3.75 Gb/s
- Less than 100 mW typical power consumption at 3.2 Gbps
- Current Mode Logic (CML) serial drivers/buffers with configurable termination, voltage swing, and coupling
- Programmable TX pre-emphasis and RX equalization for optimized signal integrity
- Optional built-in PCS features, such as 8B/10B encoding, comma alignment, channel bonding, and clock correction
- CRC generation and checking
- User dynamic reconfiguration using secondary configuration bus
- Spread-spectrum clocking support
- Built-in PRBS Generators and Checkers

GTP transceivers are placed as dual transceiver GTP_DUAL tiles in Virtex-5 LXT and SXT platform devices. This configuration allows two transceivers to share a single PLL with the TX and RX functions of both, reducing size and power consumption.

6 RocketIO is a registered trademark of Xilinx, Inc.
7 PCIe is a registered trademark of PCI-SIG Corp.
8 RapidIO is a regsistered trademark of the RapidIO Trade Association
9 XAUI is a trademark of the 10Gigabit Ethernet Alliance XAUI Interoperability Group
10 Infiniband is a registered trademark of systemI/O Inc.
11 Fibre Channel is trademark of the Fibre Channel Industry Association

Documentation available from Xilinx:

For more information on Virtex-5 FPGAs visit www.xilinx.com/virtex5. There you will find links to product overview videos, technical documentation, design tools. and more.

The first-time RocketIO transceiver user is recommended to read *High-Speed Serial I/O Made Simple*, which discusses high-speed serial transceiver technology and its applications. A free download is available at http://www.xilinx.com/publications/books/serialio/serialio-book.pdf

Virtex-5 family overview: http://www.xilinx.com/support/documentation/data_sheets/ds100.pdf

RocketIO GTP user guide: http://www.xilinx.com/support/documentation/user_guides/ug196.pdf

RocketIO GTX user guide: http://www.xilinx.com/support/documentation/user_guides/ug198.pdf

2.3 IBM Hardware used for bring-up of FlexIO™ to RocketIO™ GTP link

IBM developed a special prototype blade, called Chameleon Cell Blade, as a test platform for exploratory research and development of direct high-speed processor interfaces, accelerators, architectural studies and testing processor cores in a real firmware/software environment.

Illustration 1 shows a picture of the Chameleon Cell Blade prototype. Key components of the blade are the Cell Broadband Engine™, XDR memory, DDR2 DIMM, and in this version two Xilinx Virtex-4™ FPGAs. The prototype is designed to fit a single-wide blade form factor. FPGA#1 interacts directly with the Cell Broadband Engine™ and is the main device for running exploratory logic. FPGA#2 establishes the system IO and therefore acts as a reconfigurable southbridge. This prototype blade uses a special high-speed expansion card connector to provide PCIe connectivity and routing one single byte of FlexIO™ from the Cell Broadband Engine™ to a high speed expansion card (HSEC) for experimental extensions. One of the extensions designed by IBM is a HSEC with a Xilinx FPGA. The same FlexIO™ to RocketIO™ GTP interface exists on IBM's PowerXCell™ 8i processor as well and was setup the same way.

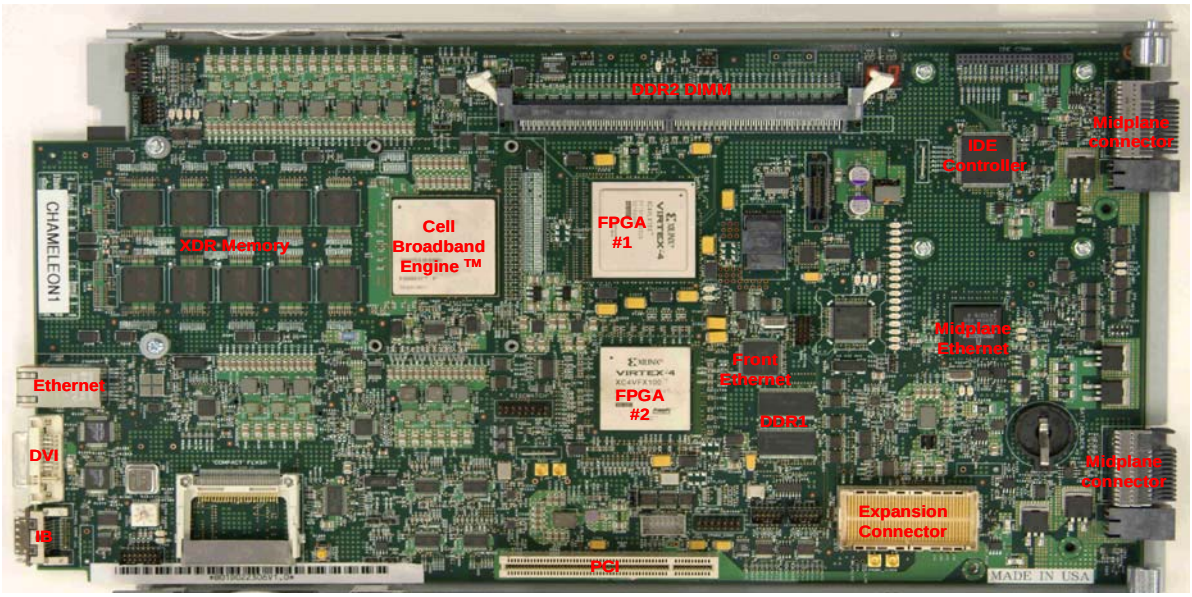


Illustration 1: Photograph of the Chameleon Cell Blade V1.0

Illustration 2 shows a simplified block diagram of the bring-up platform for the setup, testing and evaluation of the FlexIO™ to FPGA GTP interface whose implementation is described the the following chapters.

Illustration 3 shows a photograph of this special HSEC, called MagicIO HSEC, which contains as a key component: a Virtex-5™ LX110T Xilinx FPGA. The FPGA provides versatility to the expansion card function. This HSEC was part of the setup establishing a high-speed interface between the Cell Broadband Engine™ and the FPGA's GTPs. The average trace length on the board (Chameleon and

HSEC combined) for this link is about 16” routed on a standard FR4 PCB.

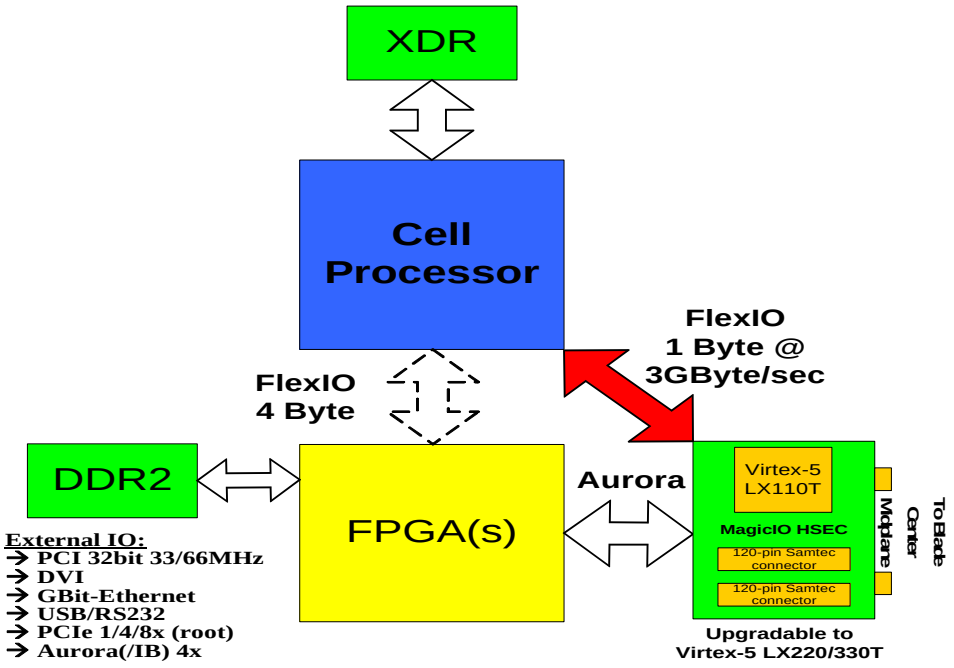


Illustration 2: Simplified block diagram of Chameleon Cell Blade with MagicIO HSEC

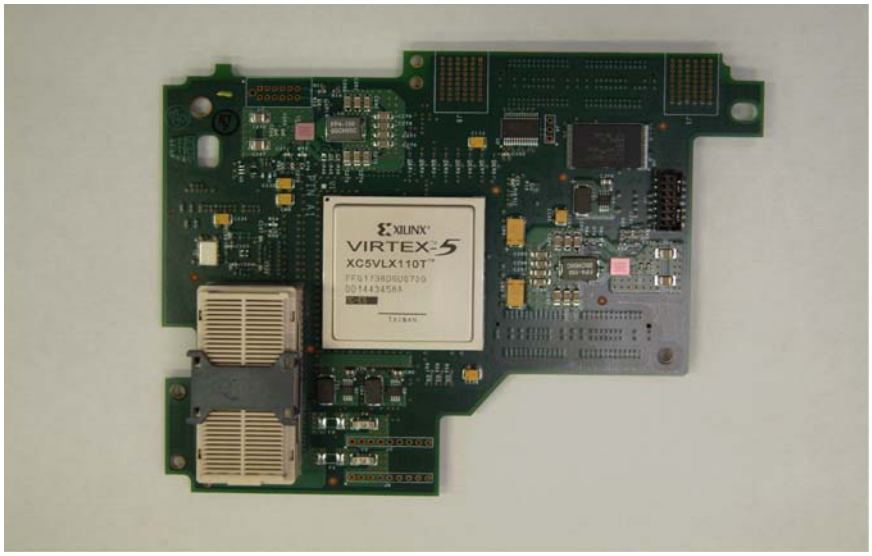


Illustration 3: Photo of the MagicIO HSEC

3 RocketIO GTP configuration

3.1 RocketIO GTP Transceiver Wizard Screen Shots

The following are the screen shots from the RocketIO GTP Transceiver Wizard for the setup used in the prototype hardware for the link. In the prototype implementation GTPs 4 – 7 are used. For board specific issue, the dedicated GTP clock pins were not used for this implementation. Using the GTP clock pins will improve the clocking structure and clock quality. The clocking implementation is described in a later section.

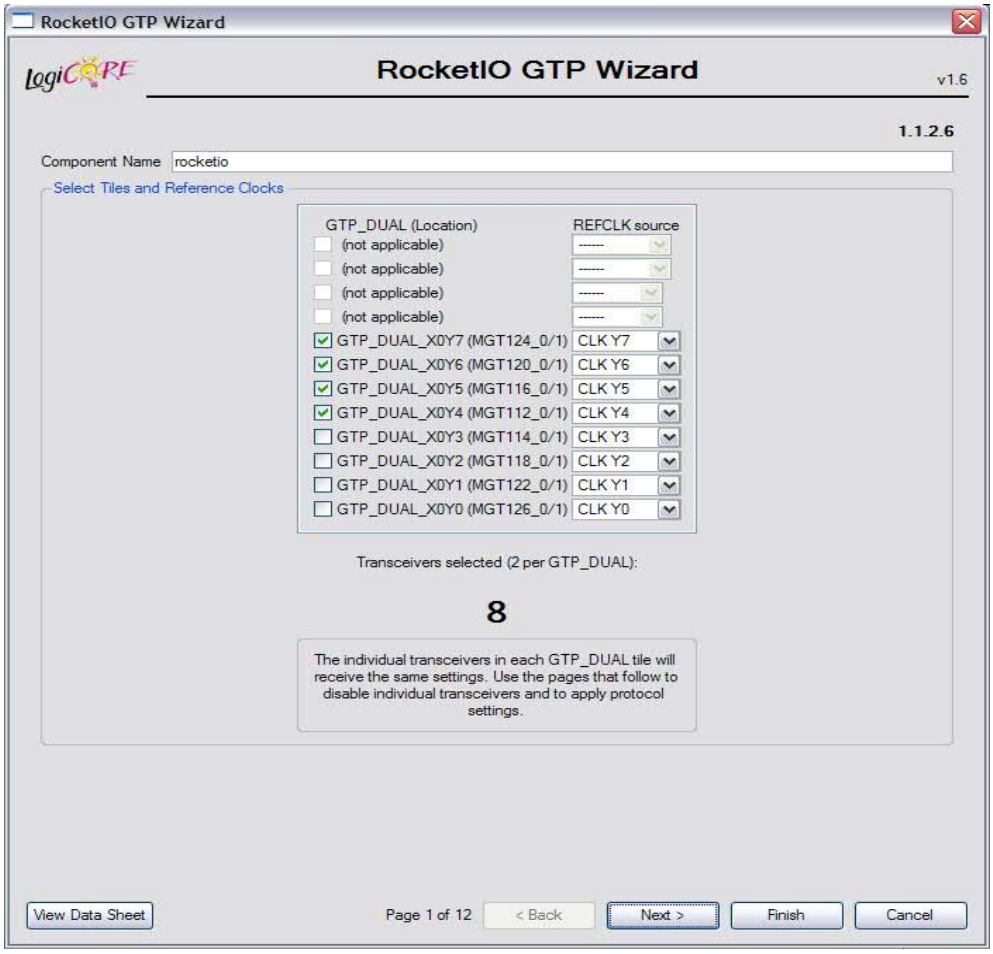


Illustration 4: RocketIO GTP Wizard screen 1 of 12

GTP_DUAL_X0Y4, GTP_DUAL_X0Y5, GTP_DUAL_X0Y6 and GTP_DUAL_X0Y7 are selected.

Select the proper REFCLK source if using the GTP clock pins. In this implementation the default value is selected.

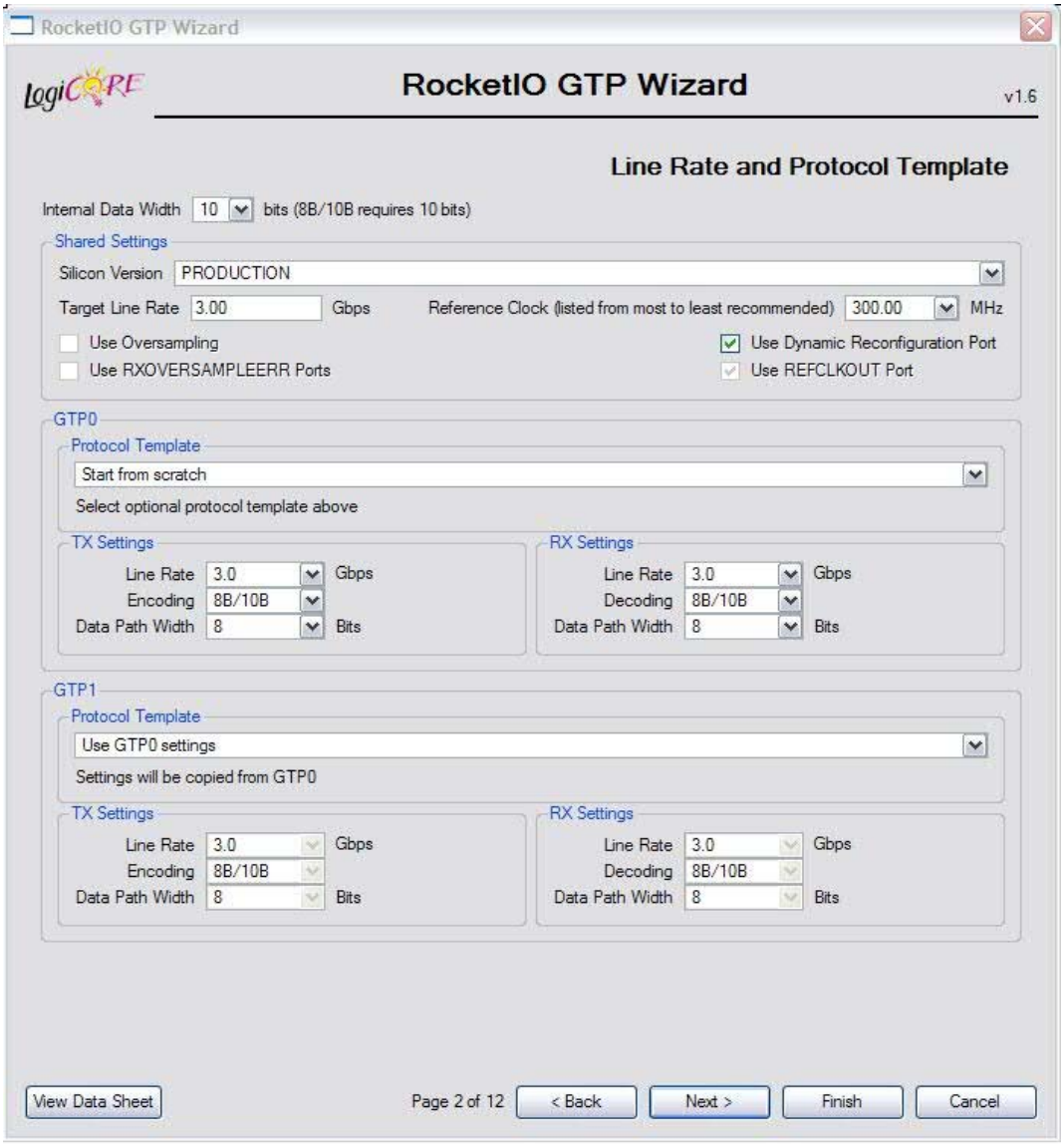


Illustration 5: RocketIO GTP Wizard screen 2 of 12

Using internal Data Width of 10 bits. Target line rate is set to 3.00 Gbps. Reference clock is set to 300 MHz. To be able to modify GTP internal settings through the GTP registers, select the option to “Use Dynamic Reconfiguration Port”. Select the option to “Start from Scratch for the GTP Protocol Template. Set the TX Settings to use 3Gbps line rate, 8B/10B Encoding and 8 bits for the data path width. The FlexIO link does not use any encoding but the the 8B/10B selection is needed to get the proper internal data width. The internal 8B/10B circuit will be bypassed and not used. The RX settings

are the same as the TX settings. For GTP1 Protocol Template, select the option to use GTP0 settings since both GTPs are identical. The 8B/10B selection above along with the data path width of 8 bits for the TX and RX will enable the user to connect the 10 bits manually outside of the RocketIO GTP wrapper. This method was used originally due to a Wizard issue which caused bad bit connection inside the RocketIO GTP wrapper. An alternative way is to use no encoding and 10 bits for data path width and the Wizard will make the proper connection internally.

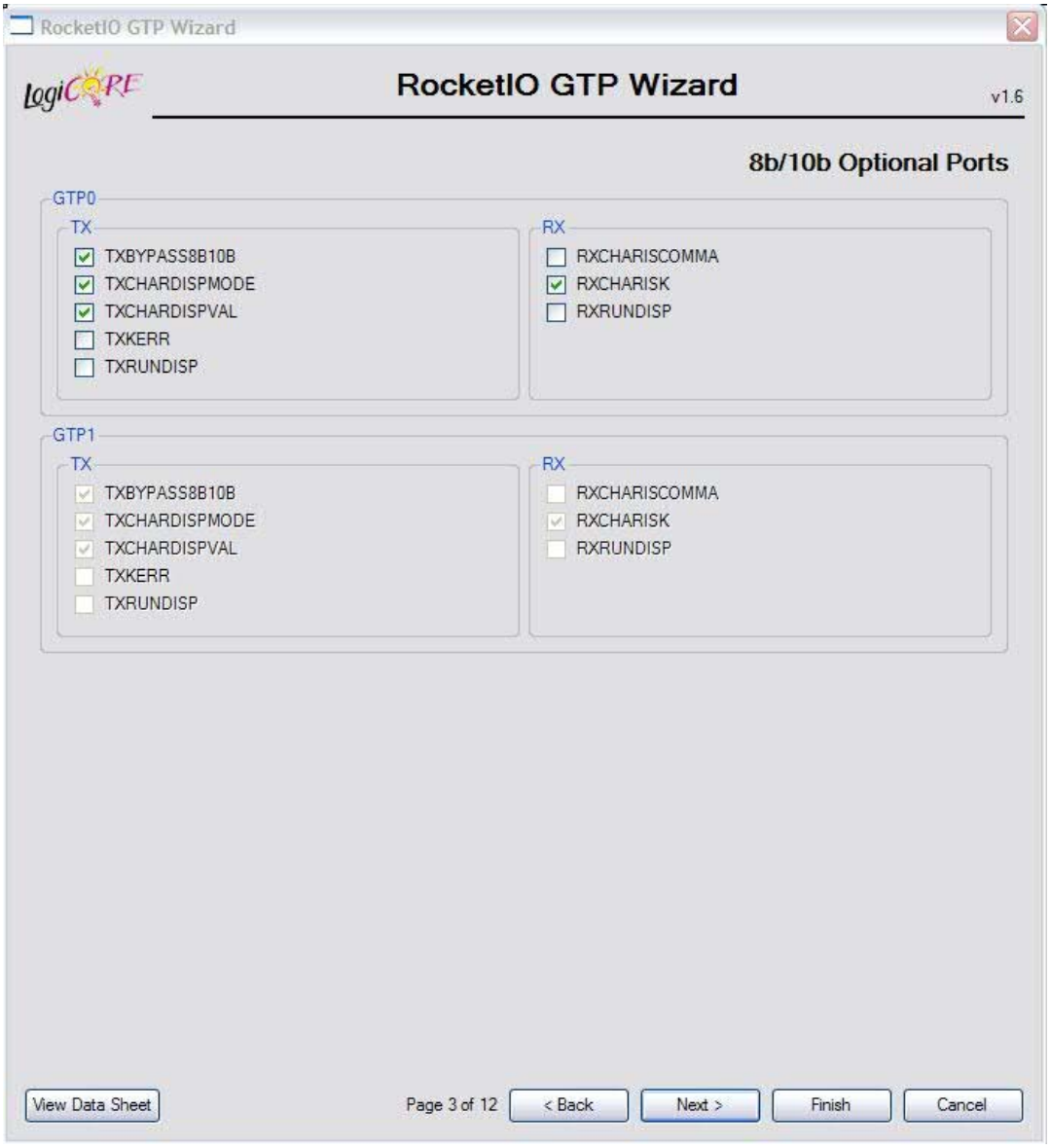


Illustration 6: RocketIO GTP Wizard screen 3 of 12

For GTP TX settings, select the following ports:

- TXBYPASS8B10B. This is used to bypass the internal 8B/10B circuit.
- TXCHARDISPMODE. This will be used for bit 9 of TX link 10 bit data.
- TXCHARDISPVAL. This will be used for bit 8 of TX link 10 bit data.

For GTP RX settings, select RXCHARISK port which will be used for bit 8 of RX link data. Bit 9 of the TX link data will use RXDISPERR port which is available but not selectable from the RocketIO GTP Wizard.

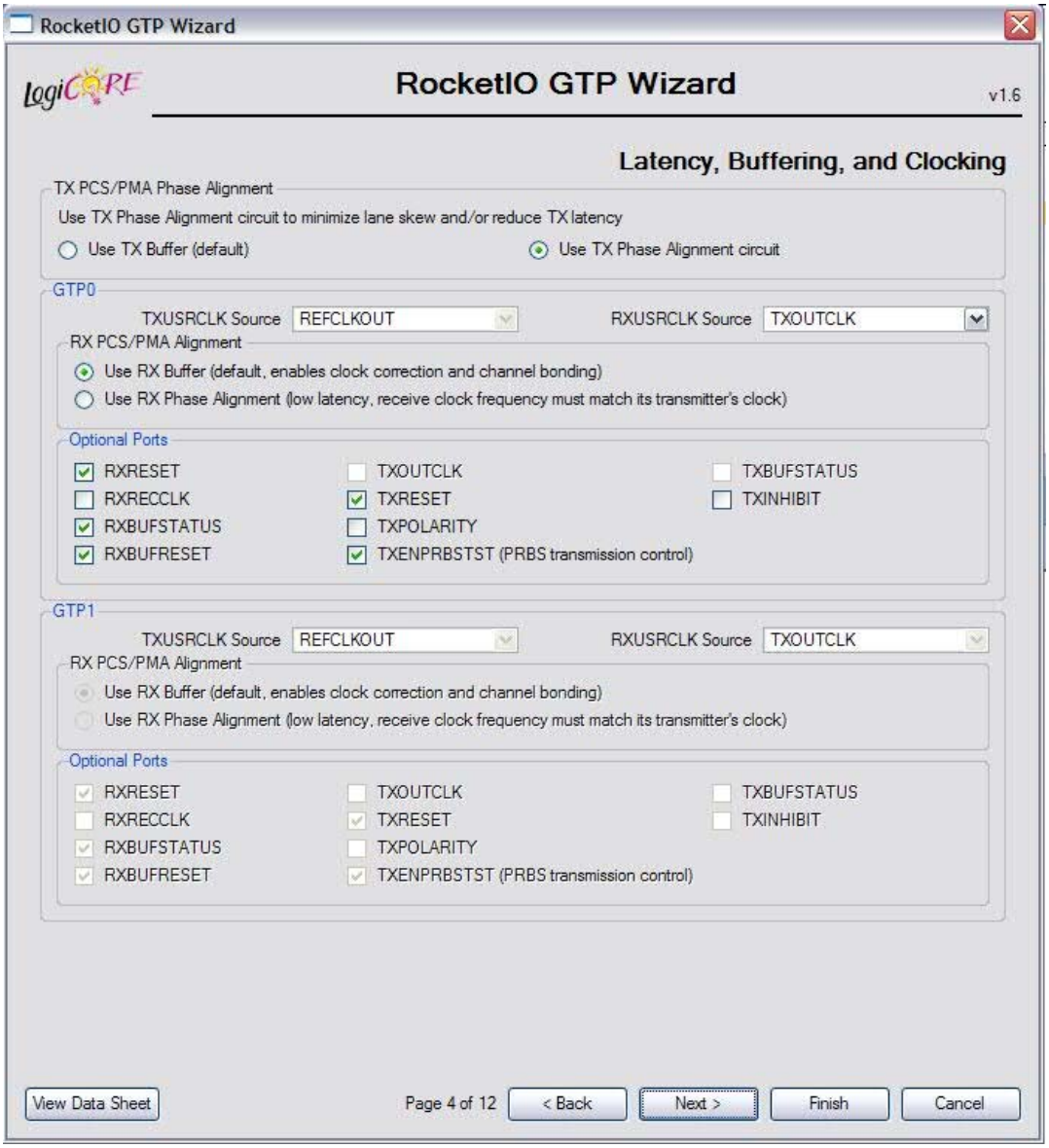


Illustration 7: RocketIO GTP Wizard screen 4 of 12

The above selections are needed if the 8B/10B selection is made in the previous page. If no encoding is selected and 10 bit data path width, this page is not applicable.

For TX PCS/PMA Phase Alignment, select the option to use TX phase alignment circuit. For the RX PCS/PMA Alignment, select the option to use RX buffer. Select the following optional ports:

- RXRESET. Can be used to reset the RX block.
- RXBUFSTATUS. Used for debug in the initial implementation.
- RXBUFRESET. Used to reset the RX buffer. This is optional but recommended.
- TXRESET. Can be used to reset the TX block.
- TXENPRBSTST. Needed to enable PRBS pattern to be sent to the FlexIO during link phase calibration.

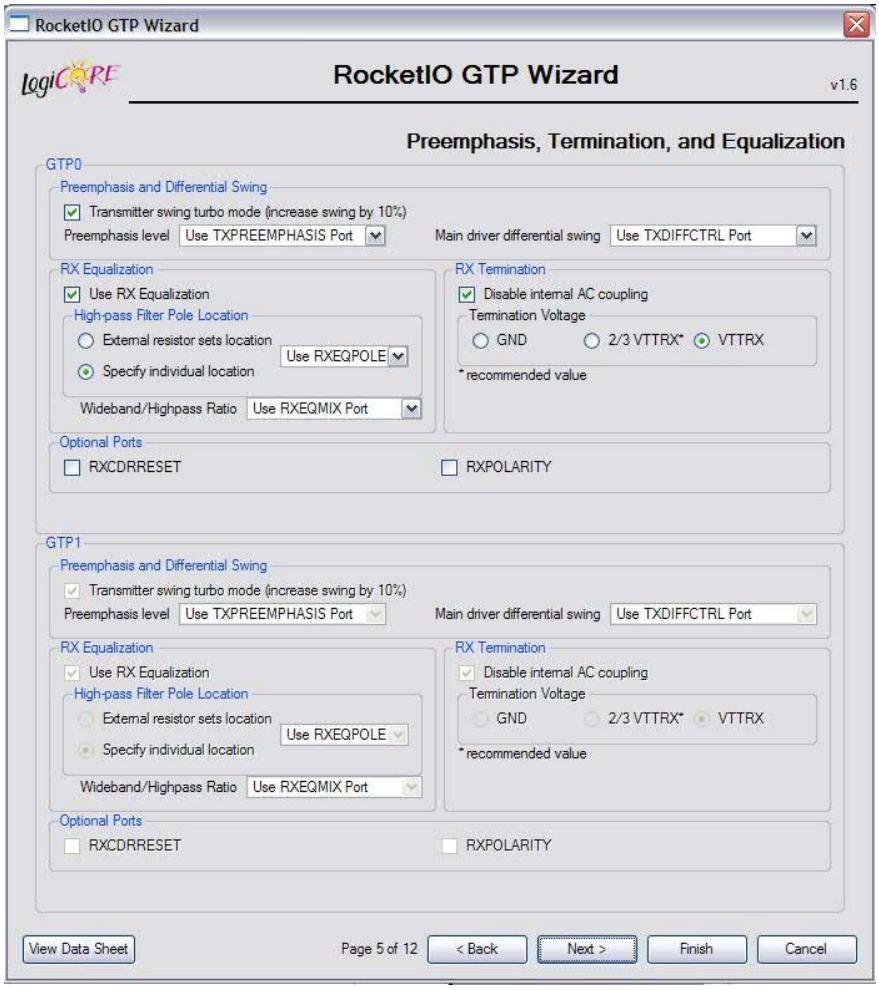


Illustration 8: RocketIO GTP Wizard screen 5 of 12

Select “Transmitter swing turbo” mode. To be able to easily modify the pre-emphasis level, select the option to “use TXPREEMPHASIS port”. Select to use the TXDIFFCTRL port for the main driver differential swing.

Select RX Equalization options as shown and select the option to disable the internal AC coupling under “RX Termination”.

Illustration 9: RocketIO GTP Wizard screen 6 of 12

The only modification to this screen is to select the option to use PRBS detector with PRBS error threshold of 1. This will be used during the GTP receive side phase calibration step.

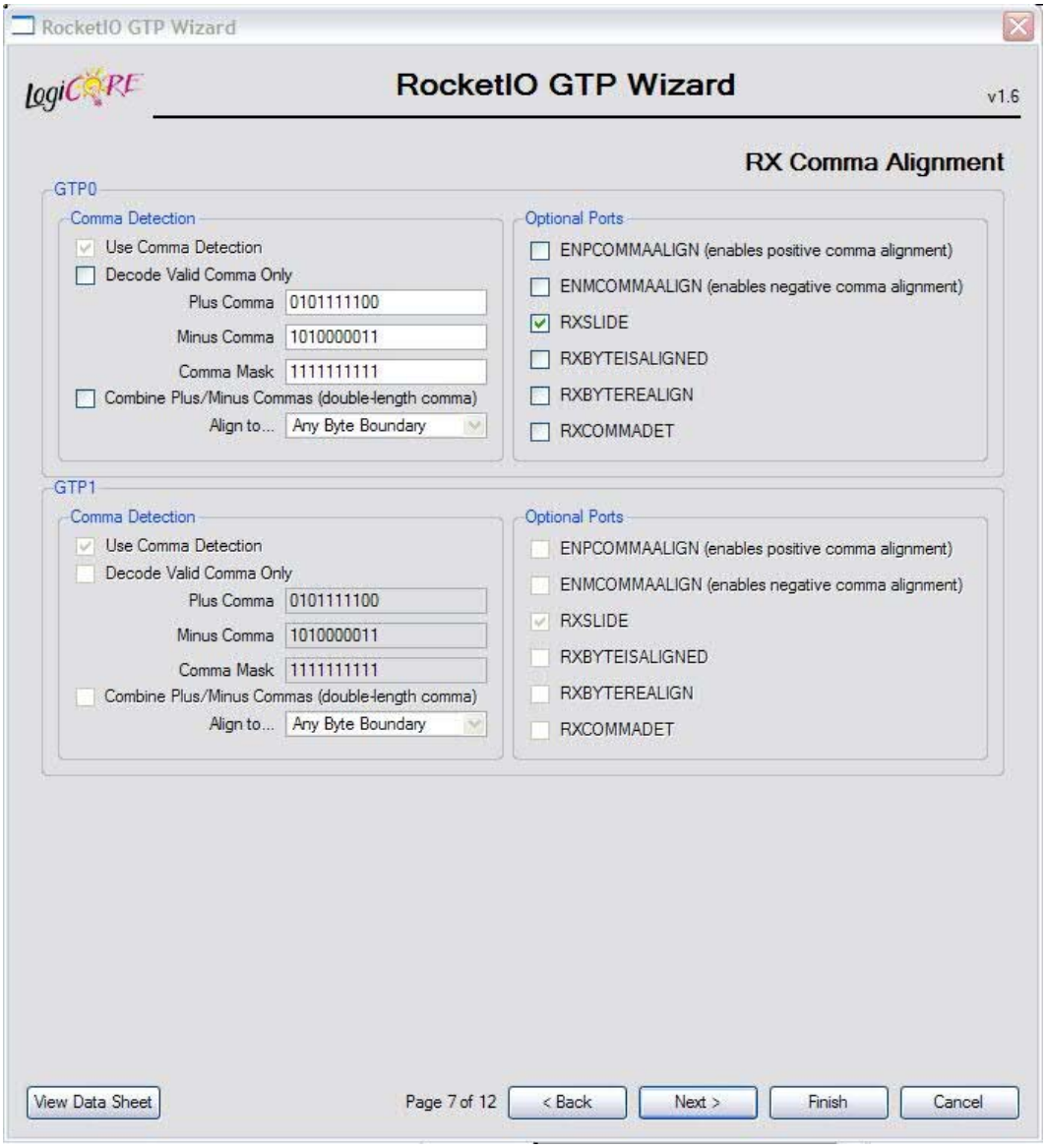


Illustration 10: RocketIO GTP Wizard screen 7 of 12

To be able to use the slide function which is used during link parallel calibration to align all data lanes, select the option to use Comma detection. This will allow the selection of the RXSLIDE as an optional port. The Comma detection circuit will be bypassed.

For Comma detection circuit bypass, ensure that the following ports are tied to ground (0):

RXENMCOMMAALIGN

RXENPCOMMAALIGN

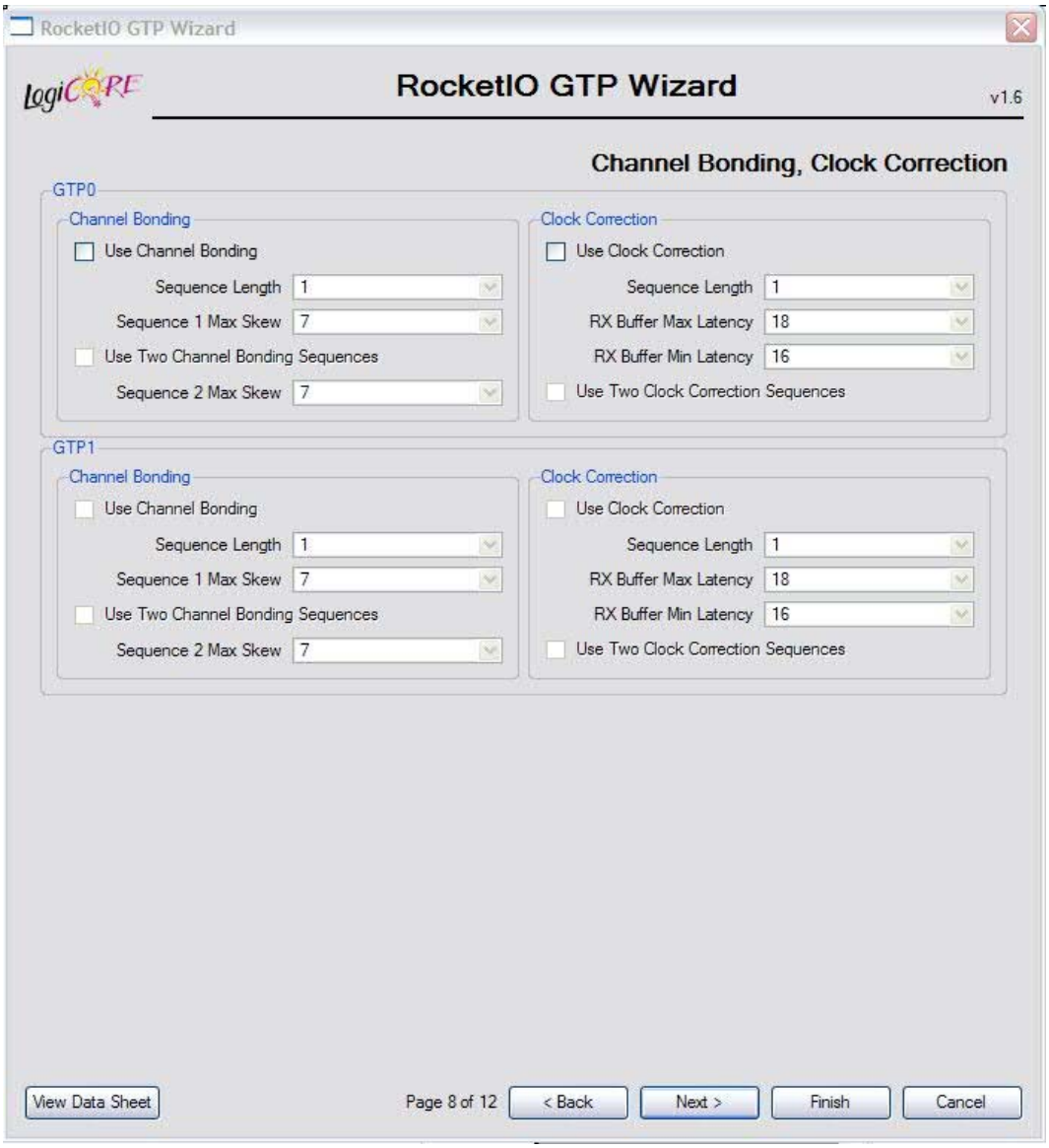


Illustration 11: RocketIO GTP Wizard screen 8 of 12

No changes are needed on this screen.

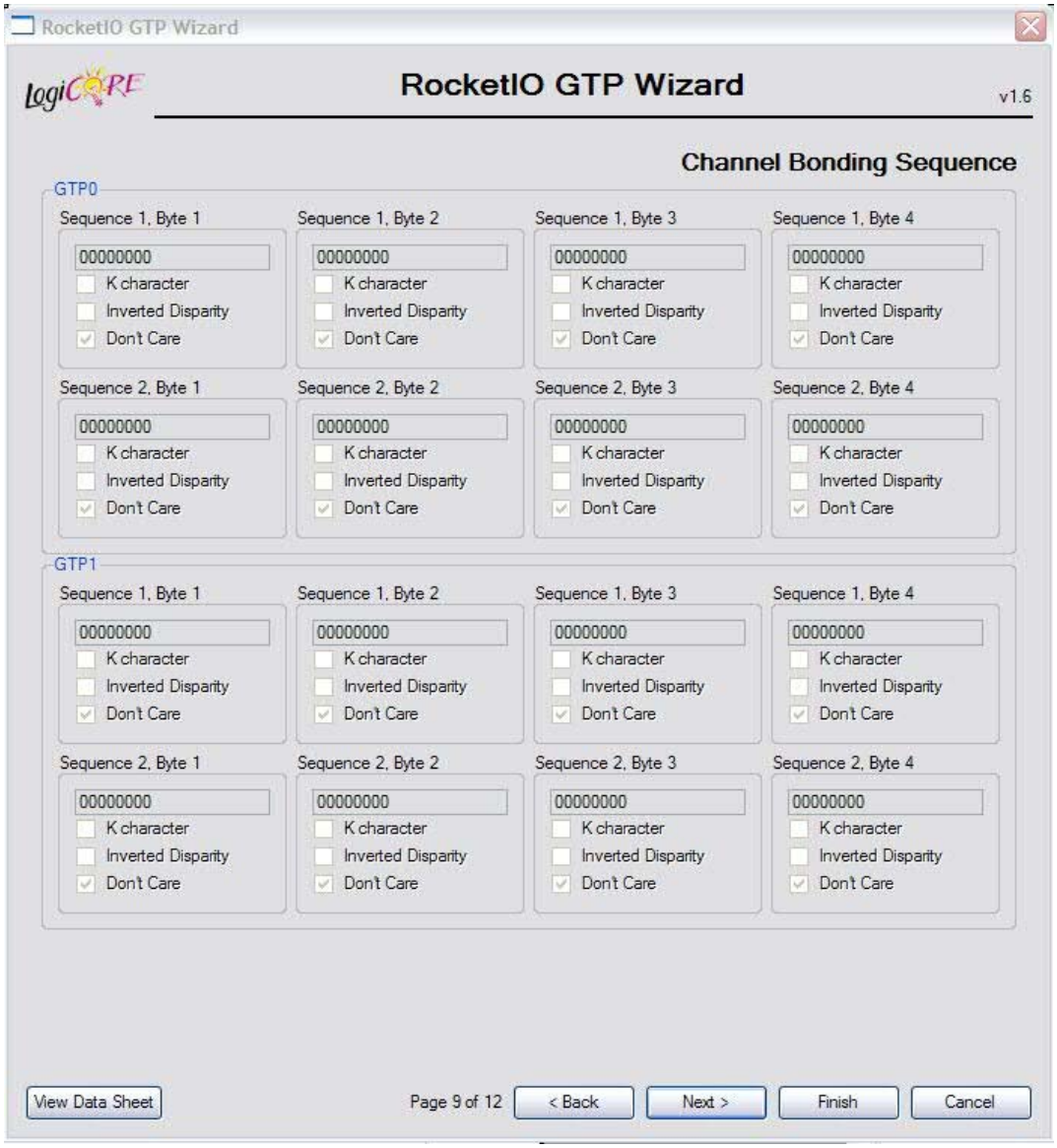


Illustration 12: RocketIO GTP Wizard screen 9 of 12

No changes are needed on this screen.

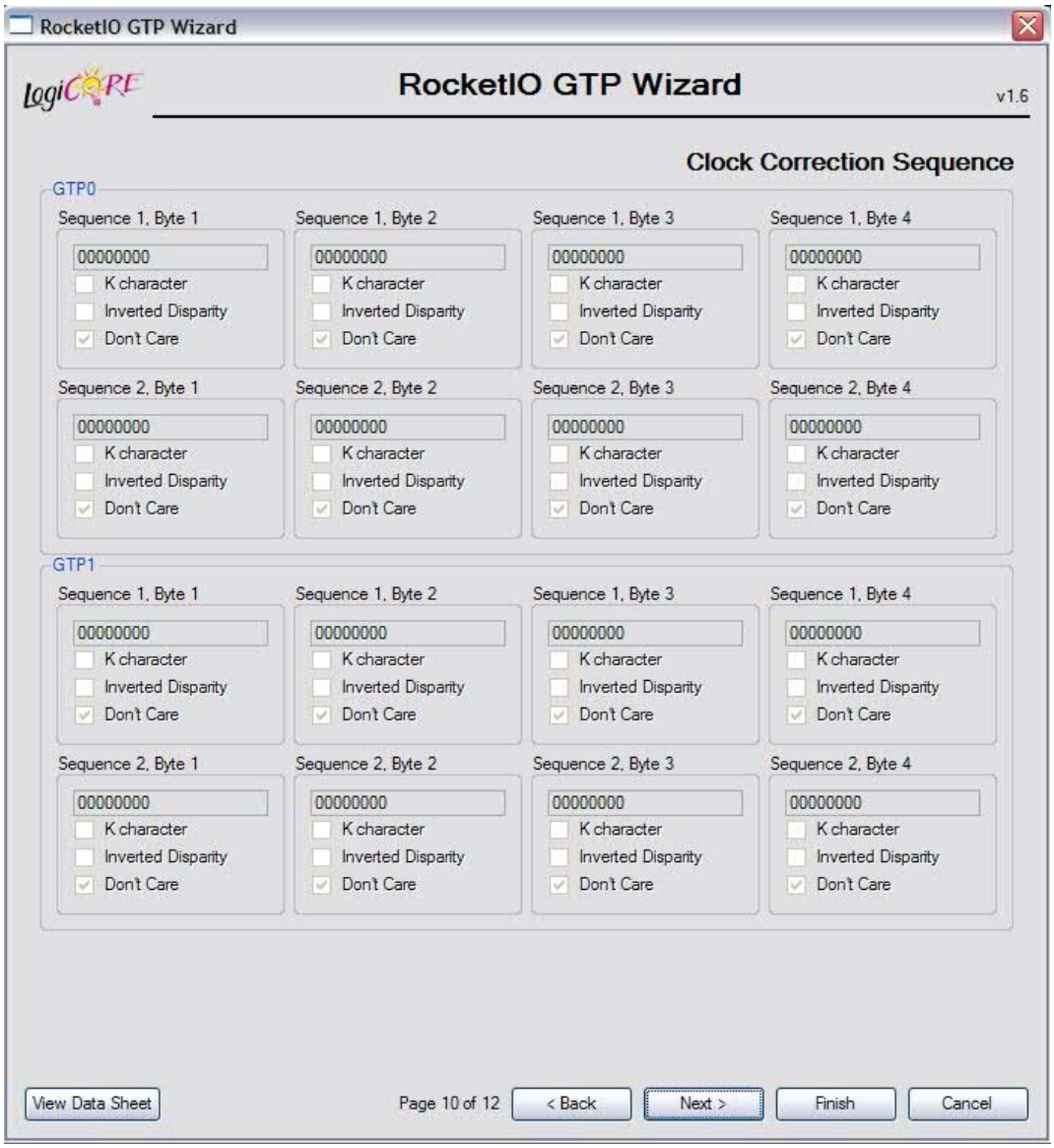


Illustration 13: RocketIO GTP Wizard screen 10 of 12

No changes are needed on this screen.

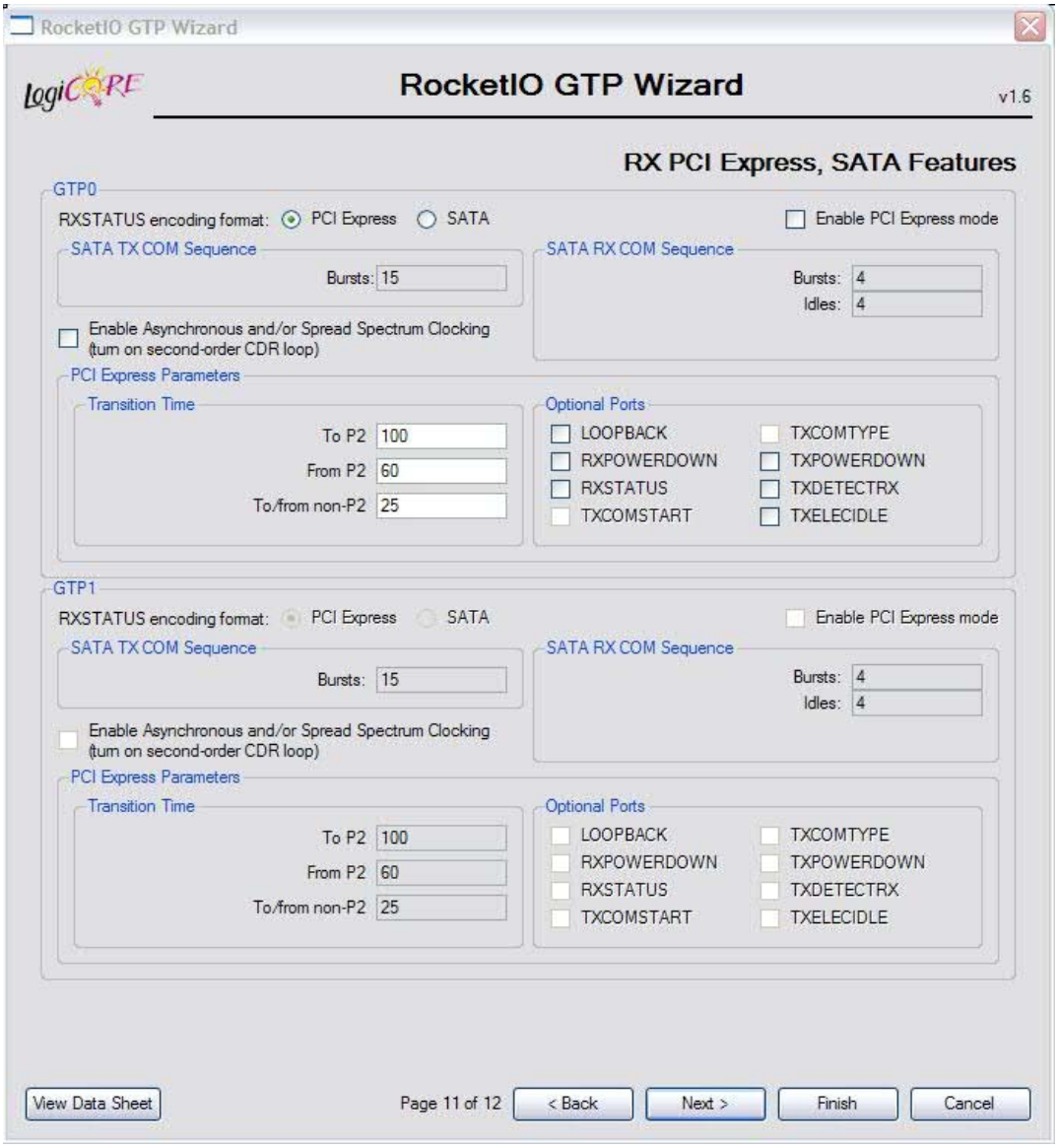


Illustration 14: RocketIO GTP Wizard screen 11 of 12

No changes are needed on this screen.

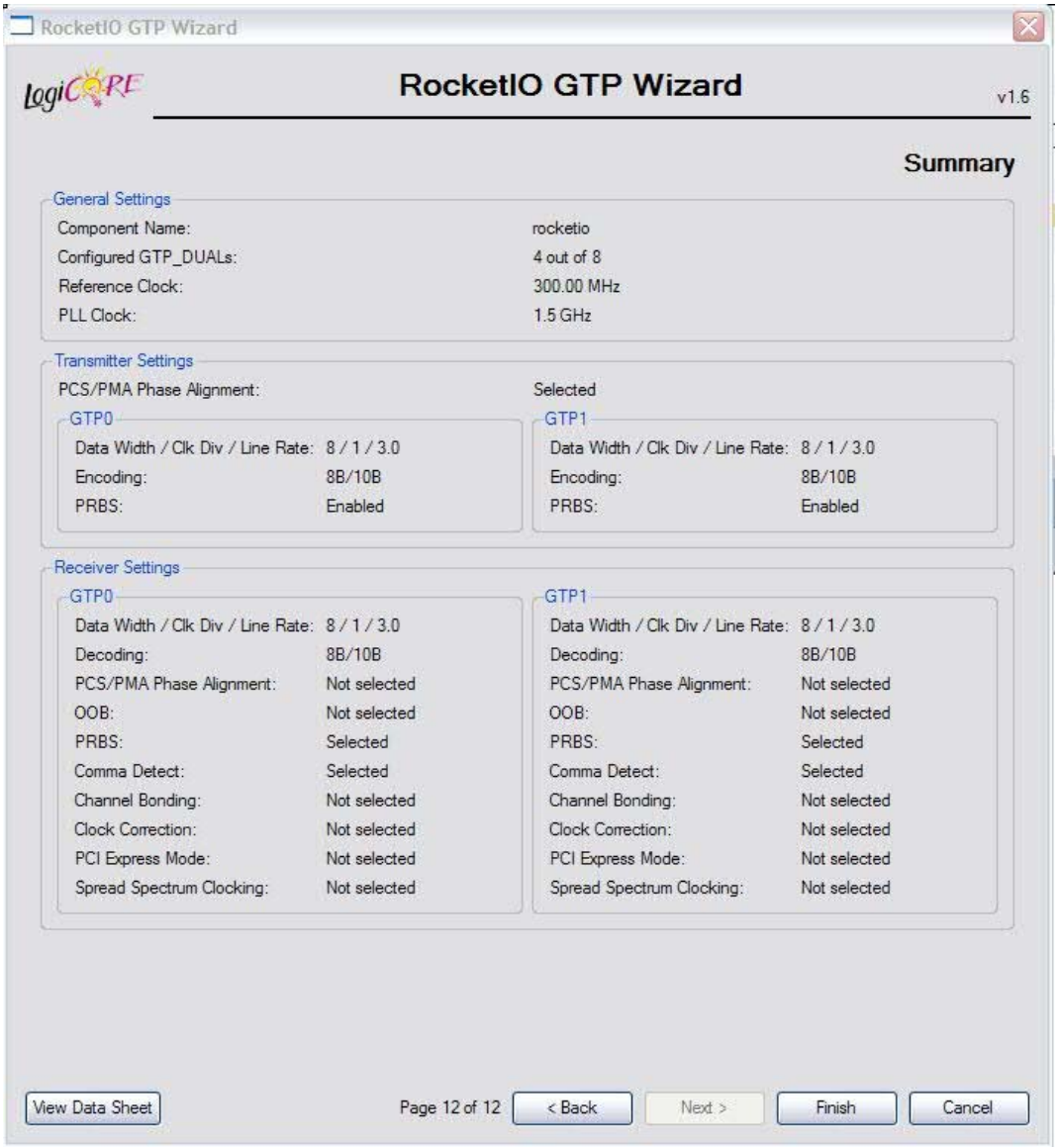


Illustration 15: RocketIO GTP Wizard screen 12 of 12

This is just informational summary screen.

3.2 Manual Edit To Source File

The GTP setup file generated for this implementation is named RocketIO which can be defined in the first screen of the RocketIO GTP Wizard. After using the RocketIO GTP Wizard to generate the GTP, an edit to the source file is needed to ensure that the 8B10B circuit in the RX block in bypassed. This is only needed if you selected to use 8B/10B encoding and 8 bit data path width. If no encoding is selected, no farther VHDL modification is needed. VHDL is used for this implementation. Under the src directory, edit the file called rocketio_tile.vhd.

In rocketio_tile.vhd:

Change the following 2 lines:

```
RXDEC8B10BUSE0      =>   tied_to_vcc_i,
RXDEC8B10BUSE1      =>   tied_to_vcc_i,
```

To:

```
RXDEC8B10BUSE0      =>   tied_to_ground_i,
RXDEC8B10BUSE1      =>   tied_to_ground_i,
```

Again to bypass the Comma detection circuit, the following ports need to be tied to '0':

```
RXENMCOMMAALIGN0      =>   tied_to_ground_i,
RXENMCOMMAALIGN1      =>   tied_to_ground_i,
RXENPCOMMAALIGN0      =>   tied_to_ground_i,
RXENPCOMMAALIGN1      =>   tied_to_ground_i,
```

3.3 Clocking

The clock names are with respect to the FlexIO in the Cell Broadband Engine or PowerXCell 8i processor chip. TXCLK is the clock going from FlexIO to RocketIO. RXCLK is the clock going from RocketIO to FlexIO. This is one implementation of the clocking that was used for the prototype hardware to test the link functionality and other implementations are also possible. Due to prototype card design, the GTP dedicated clock pins were not used in this implementation. For better and cleaner clocks into the GTP, the dedicated clock pins should be used. For better RXCLK to RXDATA tracking over a range of temperature and voltage variation, an extra GTP should be used for the RXCLK instead of an OBUFDS to send the clock to the Cell Broadband Engine or PowerXCell 8i processor. In this implementation, a 2 byte link is planned which will consume all available GTPs on the FPGA. This implementation worked well for our link design and proper cooling for both FPGA and Cell Broadband Engine or PowerXCell 8i processor.

The 375MHz clock out of the PLL (ACLK) is needed to convert the 10 bit data out of the GTP to 8 bits for each lane.

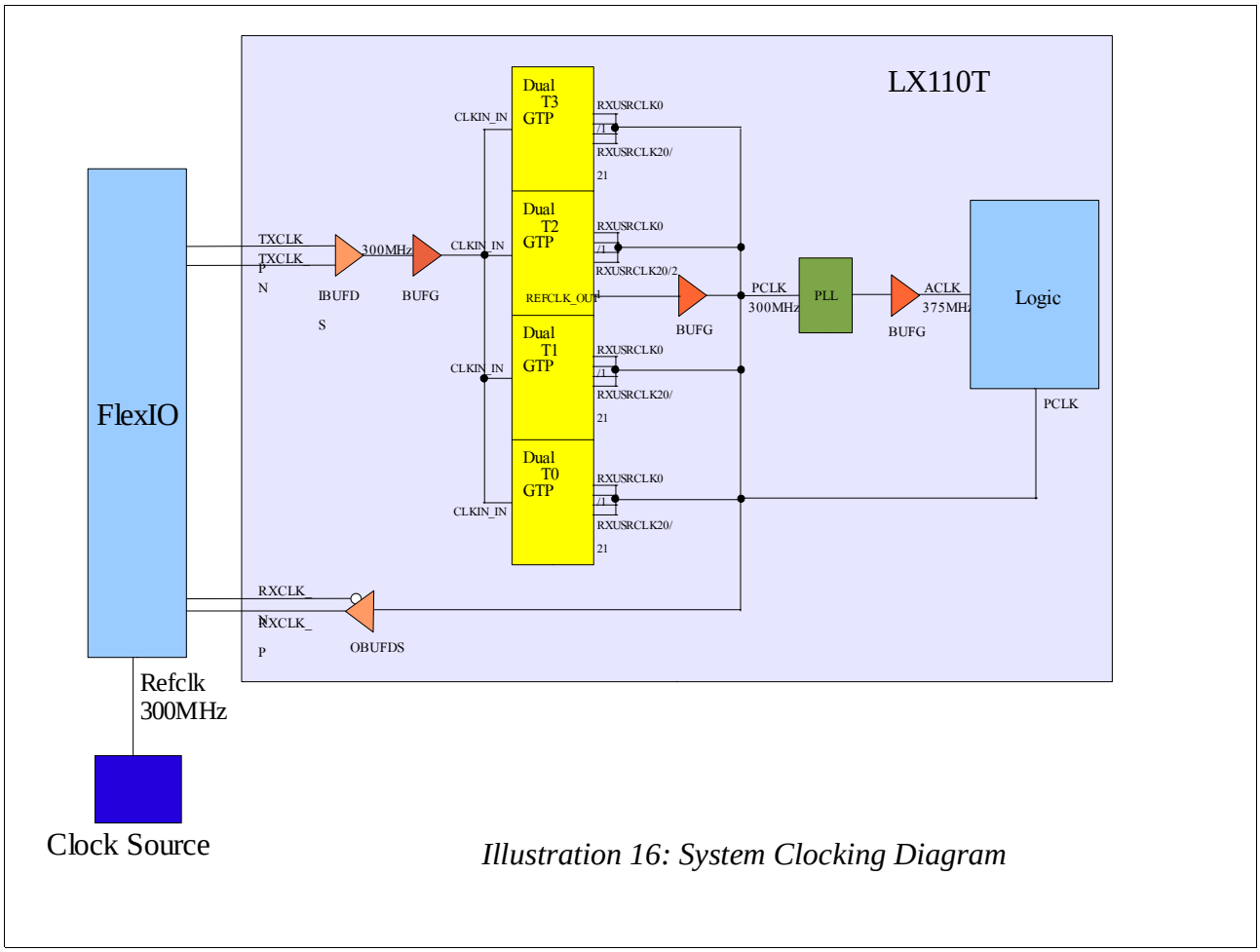


Illustration 16: System Clocking Diagram

4 TX Logic

This section describes the transmit logic. The following figure shows the TX path for one data lane. This logic is replicated 8 times to implement the 1 byte link to the FlexIO.

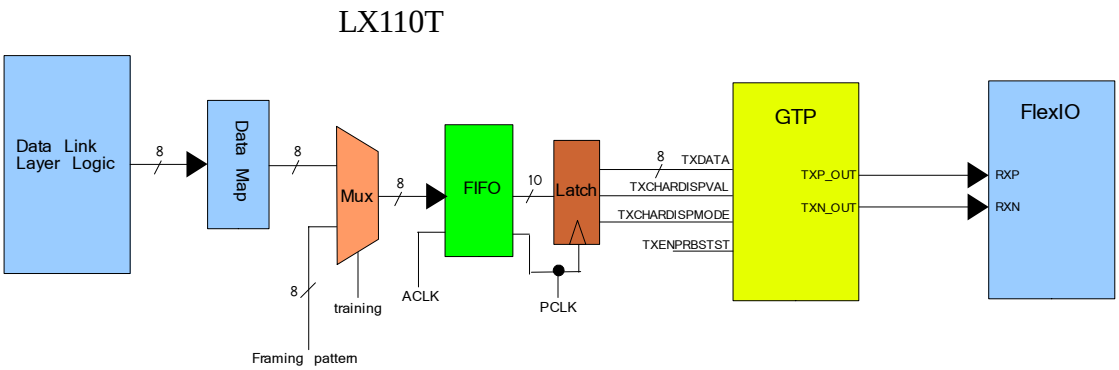


Illustration 17: TX Path Block Diagram

4.1 Data Link Layer Logic

This block refers to the higher level protocol logic. It usually deals with inter-byte alignment for multi-byte links, link CRC, data processing, etc. This layer implementation requires information beyond the scope of this application note.

4.2 Data Map

The Cell Broadband Engines's FlexIO is a 2:1 implementation. This requires a specific data bit mapping for each data lane. This mapping is described in the FlexIO data sheet This block maps the data bits to the required FlexIO data mapping.

4.3 Mux

During link parallel and levelization training steps, a pattern of 1 followed by 63 zeros is required. These training steps are explained in later sections. The multiplexer is controlled by a register bit to enable training. When training is enabled, the framing pattern is sent on each data lane. This framing pattern is used by the FlexIO receive logic to perform the required link training steps.

4.4 FIFO

The small FIFO is used to convert 8 bits of data at 375MHz to 10 bits at 300MHz. The write pointer is controlled by the ACLK and the read pointer is controlled by the PCLK. The pointer separation is set to ensure that the write data is stable before it is read out.

4.5 GTP

The 10 bits of data are connected to the GTP as follows:

Bits 7:0 connected to TXDATA(7:0)

Bit 8 connected to TXCHARDISPVAL

Bit 9 connected to TXCHARDISPMODE

During link phase calibration step, the GTP will be enabled to transmit PRBS-7 pattern which will be used by the FlexIO receive logic to perform the required training.

5 RX Logic

This section describes the receive logic. The following figure shows the RX path for one data lane. This logic is replicated 8 times to implement the 1 byte link from the FlexIO.

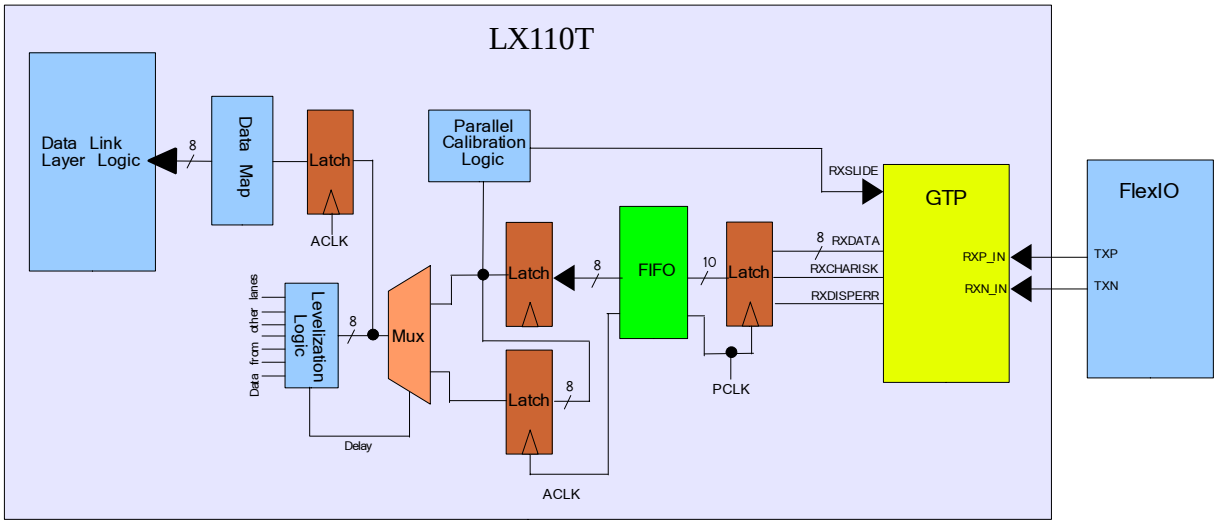


Illustration 18: RX Path Block Diagram

5.1 GTP

The RocketIO GTP block receives 10 bit serial data from FlexIO. The output data from GTP is connected as follows:

Bits 7:0 output from RXDATA(7:0)

Bit 8 output from RXCHARISK

Bit 9 output from RXDISPERR

The internal PRBS-7 checker circuit is used in the phase alignment step of link training which is described in chapter 6.

5.2 FIFO

The small FIFO is used to convert 10 bits of data at 300MHz to 8 bits at 375MHz. The write pointer is controlled by the PCLK and the read pointer is controlled by the ACLK. The pointer separation is set to ensure that the write data is stable before it is read out.

5.3 Parallel Calibration

Parallel calibration is done to synchronize the receiver's deserializer to the transmitters serializer. The transmit end which is the FlexIO in this case is sending framing pattern consisting of 1 followed by 63 zeros where the one is bit 0 at the serializer. This framing pattern is used to align the receive data to have the '1' bit in bit 0 position. The following algorithm is used during parallel calibration:

1. Find the framing pattern (the '1' bit in any position of the 8 bit pattern out of the receive FIFO).
2. Check if it is a bit 0 position. If it is, the calibration is done. If not, continue to the next step.
3. If the '1' bit is not in bit 0 position, issue a slide request to the GTP. This slide request will shift the receive data by one bit position.
4. Wait for 64 cycles which ensures that the slide request is completed and the shifted data is received at this logic.
5. Repeat steps 2, 3 and 4 until the parallel calibration is complete.
6. Send a status bit to a status register to indicate the completion of the parallel calibration.

5.4 Levelization Logic

As a result of the parallel calibration step and the skew difference between the bit lanes, the receive data could be shifted by one cycle from one data lane to another. The levelization calibration step will align all the data lanes on the same cycle. For this step of the link calibration, the same framing pattern that is used for parallel calibration is being sent from the FlexIO transmit logic. The following algorithm is used for levelization calibration:

1. Find the framing pulse (which is bit 0 = 1 after parallel calibration) on any data lane.
2. On the framing cycle, check if all the data lanes have the framing pulse. If yes, the calibration is complete.
3. If framing pulse is not present on all the data lanes, each data lane checks itself for the framing pulse. If the framing pulse is available in this cycle, the delay control is raised to delay the data by 1 cycle. Once the delay control is set to high it will stay high until power down or system reset is asserted. If the framing pulse is not present on this cycle then the delay control will stay at 0 to indicate no delay needed since the framing pulse will be present on the next cycle.
4. After step 3, the next framing pulse will have all data lanes aligned. Send calibration complete to status register.

6 Link Training

The FlexIO link requires training after system power on. The training is done once and no further adjustments will be needed while the link is operational (no dynamic calibration). The link training includes 4 levels:

1. Phase calibration.
2. Parallel calibration.
3. Levelization calibration.
4. Inter-Byte alignment for multiple byte links. This step of link alignment is application specific and beyond the scope of this document.

The link training is controlled by software. Registers on both sides of the link are setup and status registers are checked. This implementation includes an indirect register access into the GTP address space. To access the GTP registers, software writes a command (read/write), address, tile number and write data into a register and the hardware will drive the GTP Dynamic Reconfiguration PORT (DRP) to complete the register access. The DRP is described in Xilinx documentation “Virtex-5 FPGA Configuration User Guide”. The GTP registers and address map are described in Xilinx documentation “Virtex-5 RocketIO GTP Transceiver User Guide”.

The 2 GTP registers that require access for the link training are PMA_RX_CFG_0 and PMA_RX_CFG_1.

The training starts by software setting all the required FlexIO registers and enables the the transmit and receive blocks. Software also checks for PLL status to ensure stable clocks.

6.1 Phase Calibration

This step of the link training will align the data for optimal sampling point (clock edge centered on the data eye). PRBS-7 pattern is sent from the transmit side to the receive side to complete this part of calibration.

1. The GTP is enabled to send PRBS-7 pattern to the FlexIO.
2. FlexIO receive side is setup to start the phase calibration.
3. Reads FlexIO status register which includes the pass/fail status.
4. Disable GTP TX PRBS-7.
5. The FlexIO transmit side is enabled to send PRBS-7 pattern to the GTP.
6. The GTP RX Clock Data Recovery (CDR) circuit is setup by default to Oversampling mode. In this mode the the GTP will find the optimal clock to data settings.
7. Enable the GTP RX to receive PRBS-7 pattern.
8. Wait for the CDR to find the optimal setting.

9. Check GTP RX status for PRBS-7 errors. If no errors occurred on all data lanes, then the phase calibration is complete.
10. Switch the CDR to Lock-to-reference mode (where the optimal setting will be fixed). This is done via the DRP to access PMA_RX_CFG register. The switch to lock-to-reference mode is done in 2 steps for each bit lane using a read modify write sequence.
11. The default PMA_RX_CFG value is 25'h09F0088'.
12. Read-modify-write to set bit 7 to 0.
13. Read-modify-write to set bit 3 to 0.
14. PMA_RX_CFG value should now be 25'h09F0000' .
15. Check GTP RX status for errors.
16. Set GTP RX to stop PRBS-7 checking.
17. Disable FlexIO TX PRBS-7.

6.2 Parallel Calibration

This step of link training will synchronize the receive deserializer with the transmit serializer. The repeating framing sequence of one followed by 63 zeros is used for this part of link training. The following steps are done for both sides of the link (GTP -> FlexIO and FlexIO -> GTP):

1. The transmit side is setup to send the framing sequence.
2. The receive side is setup to start parallel calibration.
3. Check parallel calibration status.
4. Disable the receive side parallel calibration.
5. The transmit will keep sending the framing sequence as it is needed for the next step of link calibration.

6.3 Levelization Calibration

This step of link training will align all data lanes. The same framing sequence used in parallel calibration step is used for this step. The following steps are done for both side of the link (GTP -> FlexIO and FlexIO -> GTP):

1. The receive side is setup to start levelization calibration.
2. Check Levelization calibration status.
3. Disable the receive side levelization calibration.
4. Disable the transmit side framing sequence sending and switch is to normal mode.

7 GTP Electrical Characteristics Settings

The RocketIO driver and receiver settings are board design specific. The following settings are used for this implementation and are included here for reference. The FlexIO data sheet should be reviewed and setting adjusted for any specific design. Also FlexIO has different settings to tune the driver and receiver for the specific implementation. The settings in the RocketIO are done per bit and different bits can be tuned differently for the best operation. For this implementation, all the bits are set to the same values.

- RXEQMIX = “00”
- RXEQPOLE = “1001”
- TXDIFFCTRL = “100”
- TXPREEMPHASIS = “111”.
- TXENPMAPHASEALIGN = '1'
- TXPMASETPHASE = '1'

It is recommended to use TX PMACLK Phase-Alignment Procedure as described in the GTP User Guide rather than hard-wiring TXPMASETPHASE to 1. This will result in lower TX clock jitter in the GTP.

8 Conclusion

This application note details the hardware and software steps needed to establish a 3GByte/sec communication channel between Cell/BE's or PowerXCell 8i processor's FlexIO interface and a LX110T Virtex-5 FPGA's RocketIO™ GTP's. The evaluation setup supported a maximum 3.25Gbit/sec datarate per transceiver on the GTPs.With the newly introduced Virtex-5 FXT series each transceiver could support the maximum datarate of 5Gbit/sec of Cell Broadband Engine or PowerXCell 8i processor

This link is currently being expanded to two bytes and in the future to multiple bytes as long as there are sufficient number of GTPs are available. The two-byte interface in the works to be implemented on two new prototypes, one using Cell Broadband Engine chip as processor and the other using PowerXCell 8i processor enabling link datarates at 6Gbyte/sec and beyond in each direction. The link can be operated either in coherent more or non-coherent mode. This represents one of the fastest processor-FPGA interfaces available today and can be used as a fabric to network multiple multi-core processors together. It opens up opportunities to tie FPGAs close to a Power processor at low latency for mission critical applications.

9 Contact information for the application note

Ibrahim A. Ouda
IBM System & Technology Group
Rochester, MN, USA
Technology Development / STI development
Phone 507-253-6667
E-mail ouda@us.ibm.com

Kai R. Schleupen
IBM Research
Yorktown Heights, NY, USA
Next-Generation Computing Prototyping
Phone 914-945-1618
E-mail kais@us.ibm.com