| | Document # | Date effective |
|---|---|---|
| | | September 20, 2011 |
| | Author(s)<br>Ryan Herbst | Supersedes |
| | Revision 1.00 | |

Document Title

# Pretty Good Protocol Version 2 - Design Specification

## CHANGE HISTORY LOG

| Revision | Effective Date | Description of Changes |
|---|---|---|
| 1.00 | 09/20/2011 | Initial Version |
| | | |
| | | |
| | | |
| | | |

# Table of Contents

# 1. OVERVIEW

The Pretty Good Protocol Version 2 (PGP2) is a VHDL module which facilitates the bi-directional transmission of frame based messages over a two-wire physical link with a user selected data rate (commonly 3.125Gbs). The following features are supported by the PGP module:

- Multiple physical layer (PHY) device support
    - Xilinx Virtex4 MGT
    - Xilinx Virtex5 GTP & GTX
    - Wrappers & clock generation blocks included for supported interfaces
- Physical Layer Features
    - Automatic receive signal inversion detection
    - Multi-channel bonding for up to 4 channels
    - Endpoint version detection
    - Each link direction is independent allowing unidirectional links, disparate lane widths and or link speeds.
- 4 individual virtual channels (VC)
    - Two modes of VC interface arbitration
        - Interleave mode with equal bandwidth allocation
        - Non-interleave mode for frame at a time transport across link
    - Per VC indication of remote FIFO full & FIFO almost full status
- Cell based frame transmission
    - Allows for unlimited frame size
    - Support for stall and continue of frame transmission

    - CRC protection on a cell by cell basis for error detector
    - Lost and out of order cell detection
- Support for sideband 8-bit opcode transmission with limited latency
- Support for 8-bit sideband data transmission
- Add on applications
    - Registers controller for remote register access over PGP2
        - Single and block register access supported
    - Command controller for event generation over PGP2
    - Upstream and downstream buffers with clock rate adaptation

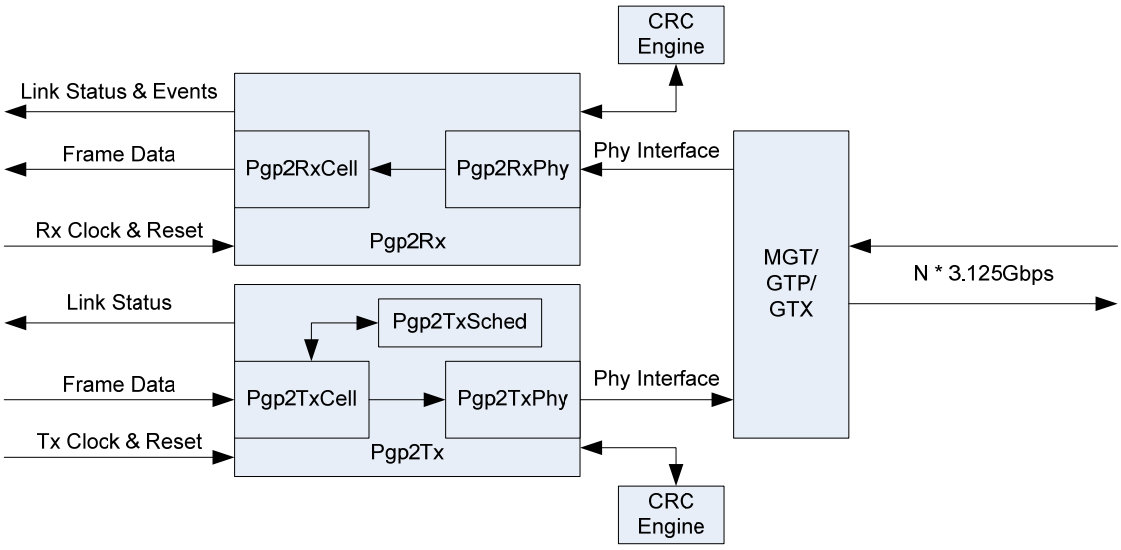
The basic operation of the PGP2 is to transport data frames from one end of a link to the other. Frames are provided by external user logic over a 16-bit FIFO like interface (16-bits per physical link). The user logic is allowed to pass frames of unlimited size and can pause frame transmission at any point for an arbitrary amount of time.

Frames received from the user interface are divided into individual cells. The cell size used can be configured at compile time. Information within the cell will identify the location of the start and end of the frame transmitted. Transmission of cells from the four virtual channels may be interleaved in order to ensure equal allocation of bandwidth between virtual channels. In applications where the receiver needs to receive complete frames one at a time a non-interleave mode can be chosen at compile time.

In addition to transporting frames across a link the PGP also transports two buffer threshold bits for each of the four virtual channels supported. The use of these two bits is entirely up to the user logic.

The PGP2 interface also supports and 8-bit user defined value which is passed across the PGP2 link. This 8-bit value is updated with each cell transmitted and can be used to pass low speed state information across the link (remote error, reset control, etc).

Additionally an 8-bit user defined opcode can be transmitted across the link. When an opcode transmission is requested by the user logic the state of the PGP2 cell transmitter is paused and the requested opcode is inserted into the serial stream with minimal latency. Similarly on the remote end of the link this opcode is presented as soon as it is received. A possible use of this interface is to generate a trigger and run start control signal at the remote end of the link. The latency and jitter of this signal can by adjusted by the operating mode and buffering of the SERDES.




PGP2 Overview Block Diagram

The source code for the PGP2 core block can be found in the “rtl/core” sub directory of the PGP2 distribution.

## 1.1 Getting PGP2

PGP2 currently resides in the research engineering group's subversion respository. The package can be checked out with the following command:

svn checkout file:///afs/slac/g/reseng/svn/repos/pgp2/trunk pgp2

The repository contains the following directory structure:

- rtl - VHDL source code
    - applications - PGP2 based applications blocks.
    - core - PGP2 core code.
    - gtp - Support blocks and wrappers for using PGP2 with a Xilinx GTP transceiver.
    - gtx - Support blocks and wrappers for using PGP2 with a Xilinx GTX transceiver.
    - mgt - Support blocks and wrappers for using PGP2 with a Xilinx MGT transceiver.
    - rce - VHDL source code for the PGP2 integration into the RCE.
- sim - Directory for various simulations used during development.
- xil_cores - Directory containing Xilinx cores (generated with coregen)

In each sub-directory under rtl you will find a VHDL package that contains the component declaration for each component contained within its sub-directory.

The xil_cores directory contains cores used by blocks in the gtp, gtx, mgt, rce and applications directory. This path must be passed to the Xilinx “ngdbuild” application so it can find the netlists needed to synthesize various modules. The core PGP2 code does not make use of any Xilinx cores.

# 2. EXTERNAL INTERFACES

Since the transmit and receive blocks operate independently their external interfaces are described separately below

## 2.1 Transmitter Block

### 2.1.1 Transmitter Generics

The following generics configure the operation of the PGP2 transmitter.

| Generic | Description |
|---|---|
| TxLaneCount | The number of lanes to support. This sets the width of the user interface as well as the width of the SERDES interface. Supported values are 1, 2, 3 & 4. The values set in the transmitter and receiver sides must match. |
| VcInterleave | Controls the interleave mode of the PGP2 transmitter. Setting this to 1 will allows frames on the four virtual channels to be interleaved. A setting of 0 will grant access to a single virtual channel until its frame is completed. |
| PayloadCntTop | Sets the MSB of the cell payload size counter. This defines the maximum size of a cell which is transmitted over the physical link. This can be used to increase link efficiency in synchronous applications. The default value is 7 (255 words). The values set in the transmitter and receiver sides must match. |

Configuration Signals

### 2.1.2 Transmit Clock & Reset

The following table describes the clock and reset signals used by the PGP2 transmit block.

| Signal | Width | Direction | Description |
|---|---|---|---|
| pgpTxClock | 1 | Input | Transmit clock. This block must be derived from the reference clock used for the SERDES device. Any frequency can be used with the common being 156.25Mhz. |
| pgpTxReset | 1 | Input | Transmit reset. This reset shall be synchronized to the pgpTxClock and asserted for at least 2 clock periods. Forces all internal logic to be reset and the physical interface to be re-linked. |
| pgpTxFlush | 1 | Input | Clear the internal state of the transmitter. This signal clears the frame state of the PGP transmitter. This allows the link state to be cleared without causing the physical link |

| | | | |
|---|---|---|---|
| | | | to go down. This is usually used when the user logic is reset (clearing buffers). |

Clock & Reset Signals

### 2.1.3 Transmit Status & Sideband Signals

The following table describes the clock status and sideband signals supported by the PGP2 transmit block.

| Signal | Width | Direction | Description |
|---|---|---|---|
| PgpTxLinkReady | 1 | Output | Asserted when the local side has completed it clock initialization and is transmitting a valid data stream. |
| PgpTxOpCodeEn | 1 | Input | This signal is asserted by the user logic to insert an OpCode sequence in the transmitted data stream. This transmission of an OpCode sequence stalls the current transmission of data for a single clock. This signal is asserted for a single clock cycle with multiple OpCode transmissions separated by at least 2 x 2^(PayloadCntTop+1) clock cycles. |
| PgpTxOpCode | 8 | Input | User defined OpCode to transmit over the physical link. Sampled when PgpTxOpCodeEn is asserted. Please note that these signals are transmitted outside of the CRC protected portion of the cell. Link errors can cause false OpCode detection. |
| pgpLocLinkReady | 1 | Input | Local receiver link status to transmit to remote end. This signal is passed over the link to be used as a link status signal to the remote end. This signal is not used in unidirectional link applications. |
| pgpLocData | 8 | Input | User defined data to pass to the remote end of the link. These signals are sampled each time a new cell is generated and updated on the remote side when the cell is received. Please note that these signals are transmitted outside of the CRC protected portion of the cell. Link errors can cause false data transmission. |

Transmit Status & Sideband Signals

### 2.1.4 Frame Transmit Interface

The following table identifies the signals which are used by the user logic to pass transmit frames to the PGP2. Each of the four virtual channels has its own set of signals. The N value in the following list is replaced with the virtual channel number, 0-3.

| Signal | Width | Direction | Description |
|---|---|---|---|

| | | | |
|---|---|---|---|
| vcNFrameTxValid | 1 | Input | Signal from user logic indicating data is ready on the input signals. Data is transferred when this signal is asserted at the same time as vcNFrameTxReady. |
| vcNFrameTxReady | 1 | Output | Signal from PGP indicating it is ready to accept data from the user logic. Data is transferred when this signal is asserted at the same time as vcNFrameTxValid. |
| vcNFrameTxSOF | 1 | Input | Signal asserted by user logic to indicate the passed data is the first of a frame. |
| vcNFrameTxEOF | 1 | Input | Signal asserted by user logic to indicate the passed data is the last of the frame. |
| vcNFrameTxEOFE | 1 | Input | Signal asserted with EOF when the user logic wishes to indicate that an error exists within the passed frame data. |
| vcNFrameTxData[M-1:0] | M | Input | Frame data passed by the user logic for transmission. M = TxLaneCnt * 16. |
| vcNLocBuffAFull | 1 | Input | Local receive buffer almost full buffer status. Used by user buffers on the remote end of the link to control frame transmission. Not used in unidirectional links. Updated with each cell transmission. |
| vcNLocBuffFull | 1 | Input | Local receive buffer full status indication. Used by user buffers on the remote end of the link to control frame transmission. Not used in unidirectional links. Updated with each cell transmission. |

Frame Transmit Signals

This interface consists of a data bus with signals to indicate start of frame (vcNTxFrameSOF), end of frame (vcNTxFrameEOF) and end of frame with error (vcTxFrameEOFE).

Handshaking between the user logic and the core consists of two signals, vcNTxFrameValid asserted by the user logic & the vcNTxFrameReady signal asserted by the core. The valid signal is asserted by the user logic when it is ready to transfer a word to the core. The ready signal is asserted by the core logic when it is ready to accept a word from the user logic. Frame data is transferred between the two blocks when valid & ready are asserted at the same time. The user logic must be able to deal with the core de-asserting ready at any point and for arbitrary periods of time. Similarly the core must accept that the user logic can de-assert valid at any time.

The PGP2 will transfer the requested frames regardless of the buffer status at the remote end of the link. It is up to the user logic to monitor the buffer status of the remote receiver to determine if it should start or pause frame transmission.

### 2.1.5 Transmit Physical Interface Signals

The PGP2 is designed with a generic interface allowing it to be adapted to multiple PHY types. A wrapper is used to adapt these generic symbols to the interface required by the PHY device. The following list describes the signals used to interface the PGP2 to a PHY device.

| Signal | Width | Direction | Description |
|---|---|---|---|
| phyTxData[M-1:0] | M | Output | Parallel data or K character to be transmitted by the PHY. M = TxLaneCnt * 16. |
| phyTxDataK[N-1:0] | N | Output | Indicates that the data to be transmitted is a K character. N = TxLaneCnt * 2. |
| phyTxReady | 1 | Input | Asserted when the physical interface initialization is complete and it is ready to transmit data & control characters. |

Physical Interface Device Signals

### 2.1.6 Transmit CRC Interface

The external transmit CRC engine is used by the PGP2 core to add a CRC protection to the cells being transferred to the remote end of the link. An external CRC engine is used to allow the design to take advantage of cores which may exist in the implementation target.

The following signals make up the Transmit CRC Interface.

| Signal | Width | Direction | Description |
|---|---|---|---|
| crcTxIn[M-1:0] | M | Output | Data being passed to the CRC engine by the PGP. M = TxLaneCnt * 16. |
| crcTxInit | 1 | Output | Signal to indicate the data passed is the first set of data in a CRC envelope. |
| crcTxValid | 1 | Output | Signal to indicate to the CRC engine that it should update the CRC on this clock. |
| crcTxOut[31:0] | 32 | Input | 32-bit CRC value generated by the CRC logic. This value is sampled 4 clocks after the last data value is passed to the CRC engine. |

Transmit CRC Interface

## 2.2 Receiver Block

### 2.2.1 Receiver Generics

The following generics configure the operation of the PGP2 receiver.

| Generic | Description |
|---|---|
| RxLaneCount | The number of lanes to support. This sets the width of the user interface as well as the width of the SERDES interface. Supported values are 1, 2, 3 & 4. The values set in the transmitter and receiver sides must match. |
| EnShortCells | This is a legacy configuration value meant to avoid short cells being passed to the RCE DMA engine. When this control is set to 1 non EOF cells which are shorter than the max cell size (defined by PayloadTopCnt) are marked as EOFE frames. This control is normally set to 0. |
| PayloadCntTop | Sets the MSB of the cell payload size counter. This defines the maximum size of a cell which is received over the physical link. This can be used to increase link efficiency in synchronous applications. The default value is 7 (255 words). The values set in the transmitter and receiver sides must match. |

Configuration Signals

### 2.2.2 Receiver Clock & Reset

The following table describes the clock and reset signals used by the PGP2 receiver block.

| Signal | Width | Direction | Description |
|---|---|---|---|
| pgpRxClock | 1 | Input | Receive clock. Defines the clock which is used to clock data out of the receive SERDES and to the user logic. The frequency of this signal must be derived from the receiver reference clock. Any frequency can be used with the common being 156.25Mhz. |
| pgpRxReset | 1 | Input | Receive reset. This reset shall be synchronized to the pgpTxClock and asserted for at least 2 clock periods. Forces all internal logic to be reset and the physical interface to be re-linked. |
| pgpRxFlush | 1 | Input | Clear the internal state of the receiver. This signal clears the frame state of the PGP receiver. This allows the link state to be cleared without causing the physical link to go down. This is usually used when the user logic is reset (clearing buffers). |

Clock & Reset Signals

### 2.2.3 Receiver Status & Sideband Signals

The following table describes the clock status and sideband signals supported by the PGP2 receiver block.

| Signal | Width | Direction | Description |
|---|---|---|---|
| PgpRxLinkReady | 1 | Output | Asserted when the local side has successfully linked to the remote transmitter. |
| PgpRxOpCodeEn | 1 | Output | This signal is asserted when the local receiver has received an OpCode sequence over the serial link. |
| PgpRxOpCode | 8 | Output | User defined OpCode received over the physical link. Valid when PgpRxOpCodeEn is asserted. Please note that these signals are transmitted outside of the CRC protected portion of the cell. Link errors can cause false OpCode detection. |
| pgpRemLinkReady | 1 | Output | Remote receiver link status received from the remote end. This signal is passed over the link to be used to monitor the link status of the remote end of the link. This signal is not used in unidirectional link applications. |
| pgpRemData | 8 | Output | User defined data received from the remote end of the link. These signals are updated each time a new cell is received. Please note that these signals are transmitted outside of the CRC protected portion of the cell. Link errors can cause false data transmission. |
| pgpRxCellError | 1 | Output | Indication of a cell error detection. This signal is asserted for one clock when a cell error (CRC fail, out of order, missing cell, etc) has been detected. Normally used to increment an error counter. |
| pgpRxLinkDown | 1 | Output | Indication of a link down event. This signal is asserted for a single clock cycle when the local link state transitions from ‘1’ to ‘0’. Normally used to increment an error counter. |
| pgpRxLinkError | 1 | Output | Indication of a link error event. This signal is asserted for a single clock cycle when a 8B/10B disparity error or decode error is detected. Normally used to increment an error counter. |

Receiver Status & Sideband Signals

### 2.2.4 Frame Receive Interface

The following table identifies the signals which are used to receive frames from the PGP2 core. Similar to the transmit interface, each of the four virtual channels has its own set of signals. The N value in the following list is replaced with the virtual channel number, 0-3.

| Signal | Width | Direction | Description |
|---|---|---|---|
| vcNFrameRxValid | 1 | Output | Signal to user logic indicating data is ready on the output signals. |
| vcFrameRxSOF | 1 | Output | Signal asserted by the PGP to indicate the passed data is the first of a frame. Common to all four virtual channels. |

| | | | |
|---|---|---|---|
| vcFrameRxEOF | 1 | Output | Signal asserted by the PGP to indicate the passed data is the last of the frame. Common to all four virtual channels. |
| vcFrameRxEOFE | 1 | Output | Signal asserted with EOF when a frame has been received with a known error. This known error is either detected within the PGP2 or indicated by the user logic on the remote end of the link. Common to all four virtual channels. |
| vcFrameRxData[M-1:0] | M | Output | Receive frame data. M = RxLaneCnt * 16. |
| vcNRemBuffAFull | 1 | Output | Remote receive buffer almost full buffer status. Not used in unidirectional links. Updated with each cell reception. |
| vcNRemBuffFull | 1 | Output | Remote receive buffer full status indication. Not used in unidirectional links. Updated with each cell reception. |

Frame Receive Signals

### 2.2.5 Receiver CRC Interface

The external receive CRC engine is used by the PGP2 core to check the CRC value of cells received from the remote end of the link. An external CRC engine is used to allow the design to take advantage of cores which may exist in the implementation target.

The following signals make up the Receiver CRC Interface.

| Signal | Width | Direction | Description |
|---|---|---|---|
| crcRxIn[M-1:0] | M | Output | Data being passed to the CRC engine by the PGP2. M = RxLaneCnt * 16. |
| crcRxInit | 1 | Output | Signal to indicate the data passed is the first set of data in a CRC envelope. |
| crcRxValid | 1 | Output | Signal to indicate to the CRC engine that it should update the CRC on this clock. |
| crcRxWidth | 1 | Output | Signal to indicate which portion of the passed data should be used for CRC generation. Used when RxLaneCnt is 4. Set to '1' to indicate 64-bits, set to '0' to indicate 32-bits. |
| crcRxOut[31:0] | 32 | Input | 32-bit CRC value generated by the CRC logic. This value is sampled 4 clocks after the last data value is passed to the CRC engine. |

Receive CRC Interface

### 2.2.6 Receiver Physical Interface Signals

The PGP2 is designed with a generic interface allowing it to be adapted to multiple PHY types. A

wrapper is used to adapt these generic symbols to the interface required by the PHY device. The following list describes the signals used to interface the PGP2 to a PHY device.

| Signal | Width | Direction | Description |
|---|---|---|---|
| phyRxData[M-1:0] | M | Input | Parallel data or K character received by the PHY. M = RxLaneCnt * 16. |
| phyRxDataK[N-1:0] | N | Input | Indicates that the data received is a K character. N = RxLaneCnt * 2. |
| phyRxReady | 1 | Input | Asserted when the physical interface initialization is complete and it is ready to receive data & control characters. |
| phyRxPolarity[N-1:0] | 1 | Output | Used to invert the receiver. This is asserted when inverted data is detected on the link. N = RxLaneCnt. |
| phyRxDispErr[N-1:0] | 1 | Input | Asserted when the 8B/10B receiver detects a disparity error. N = RxLaneCnt * 2. |
| phyRxDecErr[N-1:0] | 1 | Input | Asserted when the 8B/10B receiver detects a decode error. N = RxLaneCnt * 2. |
| phyRxInit | 1 | Output | Used by the PGP2 receiver to force the SERDES receiver to re-initialize. |

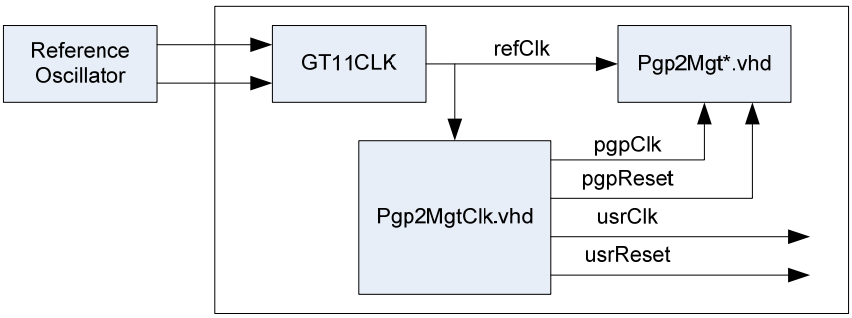
Physical Interface Device Signals

# 3. IMPLEMENTATION WRAPPERS

The PGP2 core serves as a generic core which can interface to a number of different serial wrappers. At the present time wrappers exist for the serial interfaces provided in the Xilinx Virtex 4 & 5 families. The following sections provide an overview of the serial wrappers currently supported.

## 3.1 Virtex 4 MGT

The SERDES in the Virtex 4 FPGA is the MGT. The following diagram shows an example PGP2 implementation in a Virtex 4 FPGA.




Virtex 4 MGT Implementation

The reference clock for a Xilinx Virtex 4 MGT comes in on a pin pair. To use the clock reference you must instantiate a GT11CLK module from the Virtex 4 libraries. The output of this block is a mgtRefClk signal which is then connected to one of the two reference clock ports on the Pgp2Mgtxx block described below. The mgtRefClkN port to use is chosen using the RefClkSel generic. The mgtRefClk signal is also routed to the Pgp2MgtClk block which generated the pgpClk signal which is used by the PGP2 core. The Pgp2MgtClk block also generates a user clock which has a frequency defined by the UserFxDiv and UserFxMult generics.

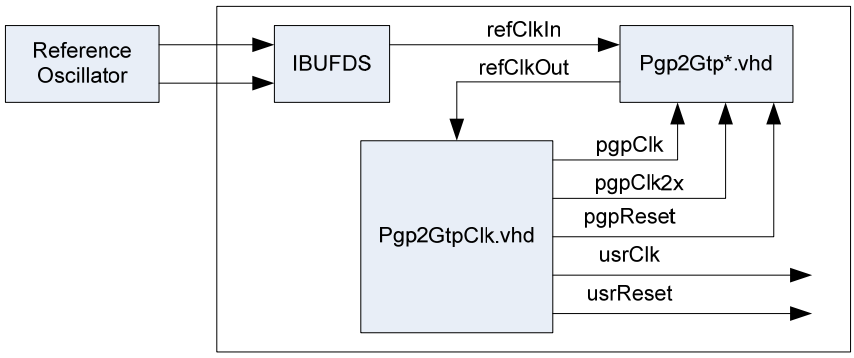
The following are the VHDL files used with the Virtex 4 FPGA. These are found in the “rtl/mgt” subdirectory of the PGP2 distribution.

| File | Usage |
|---|---|
| Pgp2MgtClk.vhd | Clock generation module for PGP2. This block is usually instantiated at the top level. |
| Pgp2MgtRxRst.vhd | MGT reset controller for the receive direction. Instantiated within the Pgp2Mgt* blocks. |
| Pgp2MgtTxRst.vhd | MGT reset controller for the transmit direction. Instantiated within the Pgp2Mgt* blocks. |
| Pgp2Mgt16.vhd | MGT single lane wrapper. This block implements a single lane using a single MGT. The second half of the instantiated MGT is unused. |
| Pgp2Mgt32.vhd | MGT dual lane wrapper. This block implements a dual lane, 32-bit wide, PGP using both halves of an MGT core. |
| Pgp2Mgt64.vhd | MGT quad lane wrapper. This block implements a quad lane, 64-bit wide, PGP using both halves of two MGT cores. |

Virtex 4 MGT Support Files

### 3.2 Virtex 5 GTP

The Virtex 5 FPGA has two different types of SERDES depending on which type is being used. The lower speed SERDES is the GTP. The following diagram shows an example PGP2 implementation in a Virtex 5 using a GTP transceiver.




Virtex 5 GTP Implementation

The reference clock for a Xilinx Virtex 5 GTP comes in on a pin pair. The incoming clock reference

is converted from LVDS to single ended using a IBUFDS module. The resulting single ended signal is then routed directly to the GTPDUAL module instantiated within the Pgp2Gtpxx block. The GTPDUAL module then outputs a copy of this reference clock which is then connected to the Pgp2GtpClk block. This block outputs the pgpClk and pgpClk2x signals which are used by the PGP2 core. The Pgp2GtpClk block also generates a user clock which has a frequency defined by the UserFxDiv and UserFxMult generics.

To use the clock reference you must instantiate a GT11CLK module from the Virtex 4 libraries. The output of this block is a mgtRefClk signal which is then connected to one of the two reference clock ports on the Pgp2Mgtxx block described below. The mgtRefClkN port to use is chosen using the RefClkSel generic.

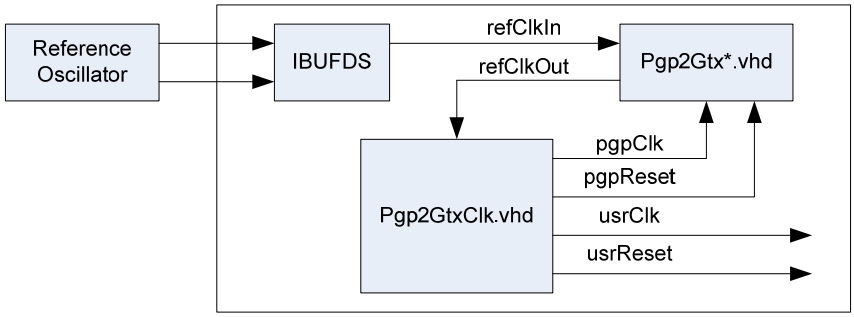
The following are the VHDL files used with the Virtex 5 GTP. These are found in the "rtl/gtp" subdirectory of the PGP2 distribution.

| File | Usage |
|---|---|
| Pgp2GtpClk.vhd | Clock generation module for PGP2. This block is usually instantiated at the top level. |
| Pgp2GtpRxRst.vhd | GTP reset controller for the receive direction. Instantiated within the Pgp2Gtp* blocks. |
| Pgp2GtpTxRst.vhd | GTP reset controller for the transmit direction. Instantiated within the Pgp2Gtp* blocks. |
| Pgp2Gtp16.vhd | GTP single lane wrapper. This block implements a single lane using one half (port A) of a single GTP core. |
| Pgp2Gtp32.vhd | GTP dual lane wrapper. This block implements a dual lane, 32-bit wide, PGP using both halves of a GTP core. |
| Pgp2GtpDual.vhd | GTP dual channel single lane wrapper. This block instantiates two separate single lane PGP2 interfaces using both halves of a GTP core. |

Virtex 5 GTP Support Files

### 3.3 Virtex 5 GTX

The higher speed SERDES in the Virtex 5 FPGA is the GTX. The following diagram shows an example PGP2 implementation in a Virtex 5 using a GTX transceiver.




Virtex 5 GTX Implementation

The reference clock for a Xilinx Virtex 5 GTX comes in on a pin pair. The incoming clock reference is converted from LVDS to single ended using a IBUFDS module. The resulting single ended signal is then routed directly to the GTPDUAL module instantiated within the Pgp2Gtpxx block. The GTPDUAL module then outputs a copy of this reference clock which is then connected to the Pgp2GtpClk block. This block outputs the pgpClk signal which is used by the PGP2 core. The Pgp2GtpClk block also generates a user clock which has a frequency defined by the UserFxDiv and UserFxMult generics.

The following are the VHDL files used with the Virtex 5 GTX. These are found in the “rtl/gtx” subdirectory of the PGP2 distribution.

| File | Usage |
|---|---|
| Pgp2GtxClk.vhd | Clock generation module for PGP2. This block is usually instantiated at the top level. |
| Pgp2GtxRxRst.vhd | GTX reset controller for the receive direction. Instantiated within the Pgp2Gtx* blocks. |
| Pgp2GtxTxRst.vhd | GTX reset controller for the transmit direction. Instantiated within the Pgp2Gtx* blocks. |
| Pgp2Gtx16.vhd | GTX single lane wrapper. This block implements a single lane using one half (port A) of a single GTX core. |
| Pgp2GtxDual.vhd | GTX dual channel single lane wrapper. This block instantiates two separate single lane |

| | |
|---|---|
| | PGP2 interfaces using both halves of a GTX core. |
| Pgp2Gtx16B.vhd | GTX single lane wrapper. This block implements a single lane using one half (port B) of a single GTX core. |

Virtex 5 GTX Support Files

### 3.4 RCE Support Files

The files contained in the “rtl/rce” subdirectory of the PGP2 distribution are used to implement the PGP2 core in a generation 1 SLAC RCE. This use of these files is not described in this document.

# 4. PGP2 APPLICATIONS

The “rtl/applications” directory contains a number of add on application blocks which are designed as add-ons to the PGP2 core. The PGP2 application modules use Xilinx cores located in the “xil_cores” directory of the PGP2 distribution.

## 4.1 Register Controller

The PGP2 Register Controller module (Pgp2RegSlave.vhd) is a block which is used to perform register read & write transactions over a PGP2 link. The module is designed to interface directly to a VC interface port on the PGP2 core. The following transactions are supported using the PGP2 Register Controller:

- Single register write transaction
- Single register read transaction
- Block register write transaction
- Block register read transaction
- Single register bit clear transaction
- Single register bit set transaction

### 4.1.1 Hardware Interface

The user interface to the PGP2 Register Controller consists of the following signals.

| Signal | Width | Direction | Description |
|---|---|---|---|
| locClk | 1 | Input | User interface clock. This clock is used to clock the user interface signals. |
| locReset | 1 | Input | User interface synchronous reset. Synchronous to the locClk and asserted for at least 2 clock periods. |
| regInp | 1 | Output | Signal to indicate that a register operation is currently in progress. This signal is asserted for the entire duration of a multiple transaction operation such as a block read, a block write, a bit set or a bit clear. |

| | | | |
|---|---|---|---|
| regReq | 1 | Output | Signal to request a register transaction to the user logic. This signal is asserted at the start of a transaction and remains asserted until the individual register transaction has completed. |
| regOp | 1 | Output | Signal to indicate the type of register transaction is being requested. This signal is set to '1' for register writes and '0' for register reads. |
| regAck | 1 | Input | Signal from the user logic to indicate the conclusion of the register transaction. For writes this signal is asserted when the register is updated. For reads this signal is asserted when the data has been presented on the regDataIn bus. This signal allows the regAddr, regReq, regOp and regDataOut signals to be heavily pipelined. The register controller will not attempt another transaction until this signal is de-asserted following the de-assertion of the regReq. |
| regFail | 1 | Input | This signal is asserted to indicate that the user logic has failed to complete the register transaction. This signal is asserted along with regAck. |
| regAddr[23:0] | 24 | Output | Register address bus. This bus is valid when regReq is asserted. |
| regDataOut[31:0] | 32 | Output | Register data output bus. This bus is valid when regReq is asserted. |
| regDataIn[31:0] | 32 | Input | Register data input bus. The user logic should present read data on this bus when regAck is asserted. |

PGP2 Register Controller Interface Signals

A register transaction is initiated by the Register Controller block asserting the regReq and regOp signals while the regAddr & regDataOut busses contain valid data. These signals are held valid until the Register Controller detects that the regAck signal has been asserted by the user logic. At this time the regDataIn bus is sampled for read transactions and the regReq signal is de-asserted. The Register Controller will then wait for the user logic to de-assert the regAck signal. If the user logic does not assert the regAck signal within 2^24 clock cycles the register controller with de-assert the regReq signal and indicate a timeout error to the requesting software.

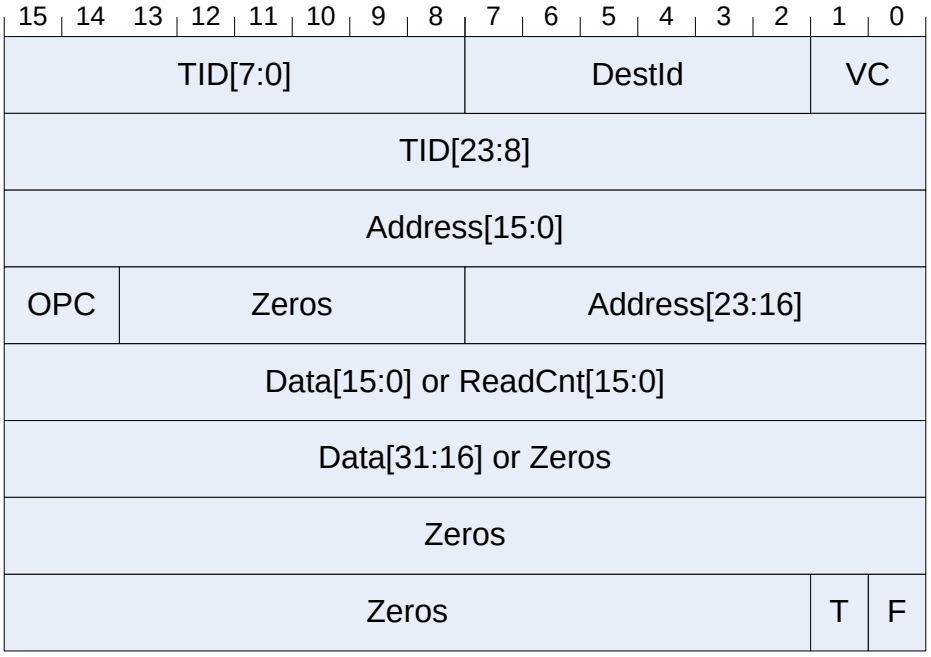
Some Register Controller operations require a number of transactions on the user interface. These include the bit set, bit clear, block read and block write transactions. Each of these transactions requires two or more separate read and write transactions on the user interface. Each of these individual transactions appears as a separate regReq assertion/de-assertion cycle on the user bus. The only difference is that the regInp signal remains asserted until the entire operation has completed.

A single generic "FifoType" is supported by the Register Controller is used to determine if a Virtex

4 or Virtex 5 FIFO is instantiated in the Register Controller. The supported values for this generic are “V4” or “V5”. The Register Controller contains a 1024 entry receive FIFO and a 1024 entry transmit FIFO.

### 4.1.2 Software Interface

The following image shows the frame structure of a register request/response transaction.




Register Controller Frame Structure

The Register Controller Frame Structure is a minimum of 16-bytes but can grow larger when block read and write transactions are generated. The following fields are present in the register transaction request and response frames:

- TID[23:0] – Transaction ID field. This is a software defined field that is echoed back to software in the response frame unmodified.
- DestId – This is the destination ID field. This field is currently not used by the register controller but is defined for future use. This field is echoed back in the response frame

unmodified..

- VC – This field is the virtual channel ID. This field is not used by the register controller and is echoed back in the response frame.
- OPC – Transaction OpCode. The following are the supported OpCode definitions:
    - 0x0 – Read transaction (single or block)
    - 0x1 – Write transaction (single or block)
    - 0x2 – Bit set transaction
    - 0x3 – Bit clear transaction
- Address – 24-bit register address for transaction. Treated as a base address for block read and write transactions.
- Data – 32-bit data value for write. For block write transactions this 32-bit value is repeated once for each location in the block transfer. The return frame for write transfers will contain a copy of the write data in the request frame. For read transactions the data field in the downstream frame will contain a zero based count (0=1, 1=2, etc) of the number of registers locations to read. A single location read frame will contain a value of 0. The return frame for read transactions will contain a 32-bit value for each requested read location.
- F – Fail bit. This bit is set when the user logic asserts the regFail signal.
- T – Timeout bit. This bit is set when the register controller times out waiting for the user logic to assert the regAck signal.

### 4.2 Command Controller

The Command Controller block (Pgp2CmdSlave.vhd) is used to send a command from software over the PGP2 link. This command results in a single clock assertion of the cmdEn signal associated with an 8-bit opCode which is user defined.

### 4.2.1 Hardware Interface

The following table defines the hardware interface to the Command Controller.

| Signal | Width | Direction | Description |
|---|---|---|---|
| locClk | 1 | Input | User interface clock. This clock is used to clock the user interface signals. |
| locReset | 1 | Input | User interface synchronous reset. Synchronous to the locClk and asserted for at least 2 clock periods. |
| cmdEn | 1 | Output | Signal to indicate that a command is being generated. Asserted for a single clock period along with the cmdOpCode and cmdCtxOut signals. |
| cmdOpCode[7:0] | 8 | Output | Signal containing an 8-bit user defined 8-bit opCode value. |
| cmdCtxOut[23:0] | 24 | Output | Optional 24-bit user defined context value which can be used by software to track transactions. Some applications may include this value in a possible return data frame as a result of the command transaction. |

PGP2 Command Controller Interface Signals

The following generics are supported by the PGP2 Command Controller.

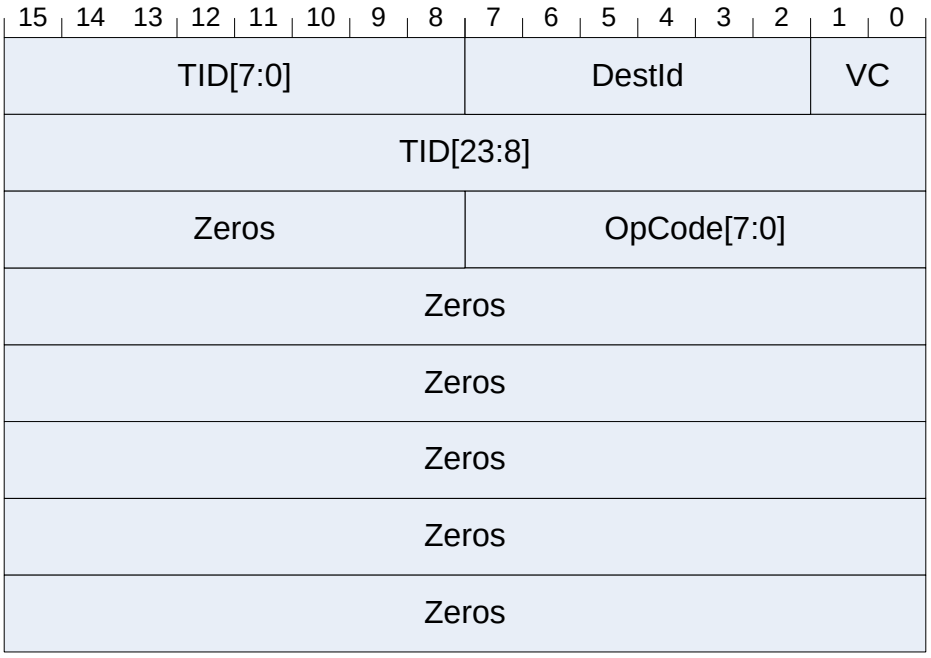
| Generic | Description |
|---|---|
| DestId | Destination ID to which this Command Controller will respond. The destination ID is included in the Command Controller header. The use of this field allows multiple command controller modules to be connected to a single PGP2 virtual channel. |
| DestMask | Mask used when matching the DestId generic to the destination ID field received in the Command Controller header. |
| FifoType | Generic to determine the FIFO type instantiated in the Register Controller. Supported values for this generic are “V4” for Virtex 4 or “V5” for Virtex 5. |

PGP2 Command Controller Generics

The Register Controller contains a 1024 entry receive FIFO.

### 4.2.2 Software Interface

The following image shows the frame structure of a command request transaction.




Command Controller Frame Structure

The Command Controller Frame Structure is a fixed size of 16-bytes. The following fields are present in the register transaction request and response frames:

- TID[23:0] – Transaction ID field. This is a software defined field. The user may decide to include this value in any return data frame that results from the command.
- DestId – This is the destination ID field. This field is used when multiple command receivers are placed on a given virtual channel interface.
- OpCode – This 8-bit value is passed to the user logic along with command strobe. This is a user defined field.

### 4.3 Downstream Data Buffer

The downstream data buffer is a generic block which serves as an asynchronous buffer placed between the receive interface of the PGP2 block and the user logic. This block adapts the receive data to the user clock while generating the flow control signals to the PGP2 core.

The following table defines the user hardware interface to the downstream data buffer.

| Signal | Width | Direction | Description |
|---|---|---|---|
| locClk | 1 | Input | User interface clock. This clock is used to clock the user interface signals. |
| locReset | 1 | Input | User interface synchronous reset. Synchronous to the locClk and asserted for at least 2 clock periods. |
| frameRxValid | 1 | Output | Signal to indicate data is ready on the user interface. |
| frameRxReady | 1 | Input | Signal from the user logic to indicate a data value can be transferred between the downstream buffer and the user logic. |
| frameRxSOF | 1 | Output | Signal indicating start of frame. |
| frameRxEOF | 1 | Output | Signal indicating end of frame. |
| frameRxEOFE | 1 | Output | Signal indicating end of frame with error. |
| frameRxData[15:0] | 16 | Output | Receive data bus. |

PGP2 Downstream Buffer Signals

A single generic “FifoType” is supported by the downstream buffer is used to determine if a Virtex 4 or Virtex 5 FIFO is instantiated in the downstream buffer. The supported values for this generic are “V4” or “V5”. The downstream buffer contains a 1024 entry receive FIFO.

### 4.4 Upstream Data Buffer

The upstream data buffer is a generic block which serves as an asynchronous buffer placed between the transmit interface of the PGP2 block and the user logic. This block adapts the receive data to the user clock while responding to the flow control signals from the PGP2 core.

The following table defines the user hardware interface to the upstream data buffer.

| Signal | Width | Direction | Description |
|---|---|---|---|
| locClk | 1 | Input | User interface clock. This clock is used to clock the user interface signals. |
| locReset | 1 | Input | User interface synchronous reset. Synchronous to the locClk and asserted for at least 2 clock periods. |
| frameTxValid | 1 | Input | Signal to indicate data is valid to be written to the upstream data buffer. |
| frameTxSOF | 1 | Input | Signal indicating start of frame. |

| | | | |
|---|---|---|---|
| frameTxEOF | 1 | Input | Signal indicating end of frame. |
| frameTxEOFE | 1 | Input | Signal indicating end of frame with error. |
| frameTxData[15:0] | 16 | Input | Receive data bus. |
| frameTxAFull | 1 | Output | Signal to indicate that the local buffer is starting to fill up. The user logic should stop presenting data within 4 clocks of this signal being asserted. |

PGP2 Upstream Buffer Signals

A single generic “FifoType” is supported by the downstream buffer is used to determine if a Virtex 4 or Virtex 5 FIFO is instantiated in the downstream buffer. The supported values for this generic are “V4” or “V5”. The downstream buffer contains a 1024 entry receive FIFO.

# 5. PGP2 PROTOCOL

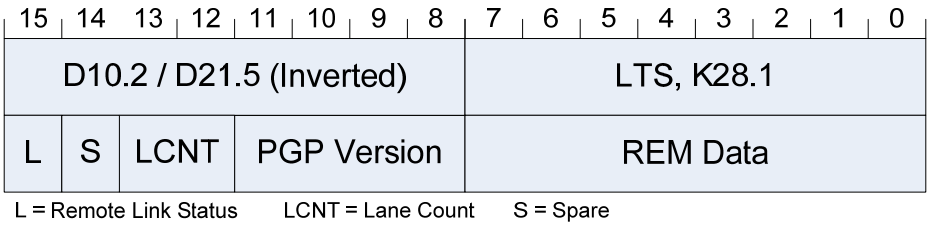
The serial link between the transmitter and receiver is a continuous stream of control characters and payload data. The primary data type sent over the link is the cell structure containing payload data. The space between each cell data structure consists of one empty clock, following by two ordered sets. The first ordered set following the empty clock is either a clock compensation ordered set or a link alignment ordered set. The second ordered set is always the link initialization ordered set.

The link efficiency is determined by how much real data can be transmitted for each block of user frame data that is transmitted. With a PayloadCntTop value of 7 (default) 256 clocks of real data can be transmitted in each cell. The non user data for each block of real data consists of the following:

- Link Initialization Ordered Set – 2 clocks
- Link Alignment or Clock Compensation Ordered Set – 2 clocks
- One clock of empty data
- Cell structure overhead – 4 clocks

This results in an efficiency of 256/(256+9) = 96.6%

## 5.1 Link Initialization Ordered Set



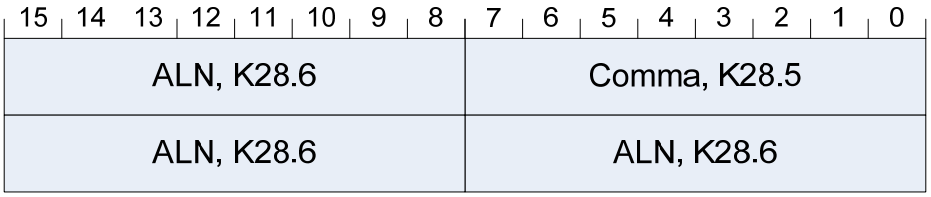

PGP2 Link Initialization Ordered Set

The Link Initialization Ordered Set is transmitted regularly by the PGP2 transmitter. The purpose of this ordered set is to detect link and to detect receive link inversion. This ordered set is also used to transport link data. The following fields are contained in the Link Initialization Ordered Set.

- LTS, K28.1 – K character for link initialization.
- D10.2 / D21.5 – Data character for link inversion detection.
- RemData – 8-Bit user data transported over the PGP2 link.
- PgpVersion – PGP Version field to ensure compatibility of the endpoints.
- LCNT – Field containing the lane count for channel bonding.
- S – Spare bit for future use.
- L – Remote link status.

The Link Initialization Ordered set is transmitted between each cell transmission.

## 5.2 Link Alignment Ordered Set

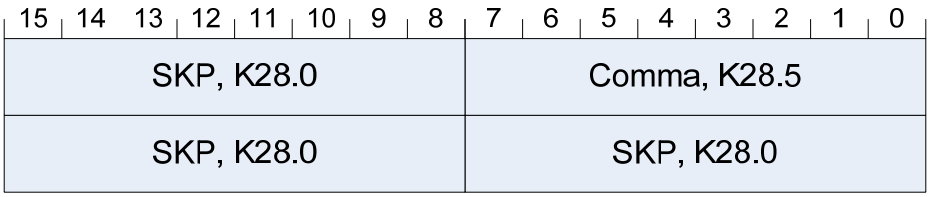



PGP2 Link Alignment Ordered Set

The Link Alignment Ordered Set is transmitted regularly by the PGP2 transmitter. The purpose of this ordered set is to align multiple lanes when the channel bonding feature of the PGP2 core is being used.

The Link Initialization Ordered set is transmitted between every other cell transmission.

## 5.3 Clock Compensation Ordered Set

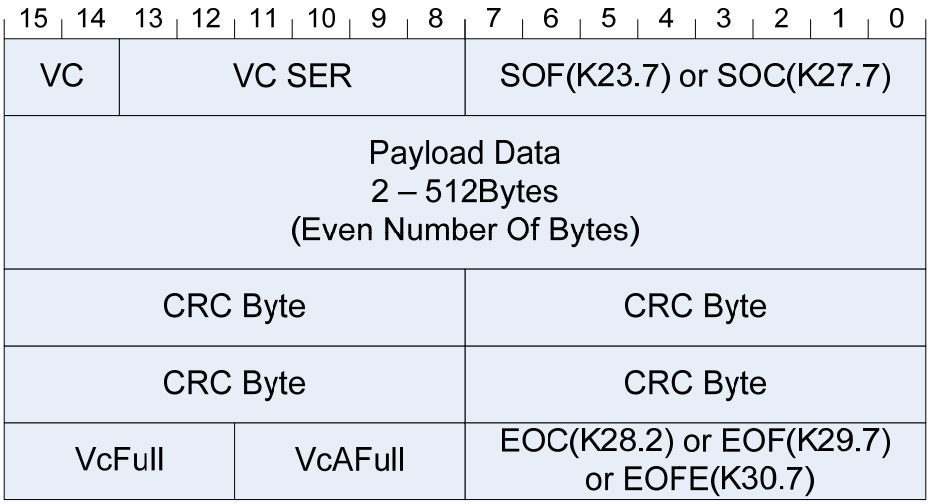


PGP2 Clock Compensation Ordered Set

The Clock Compensation Ordered Set is transmitted regularly by the PGP2 transmitter. The purpose of this ordered set is to adjust the receive buffer to adapt for differences between the transmit and receive oscillator frequencies.

The Clock Compensation Ordered set is transmitted between every other cell transmission.

## 5.4 Cell Data



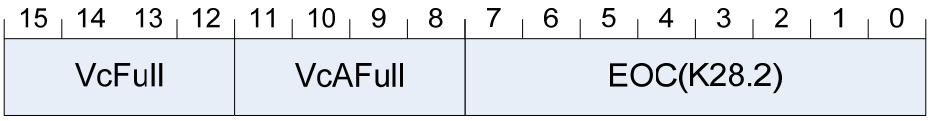

PGP2 Cell Data Structure

The PGP2 cell structure is used to transport frame data across the PGP2 link. The cell contains a header, a payload and footer. The header contains a K-character indicating SOF or SOC, a virtual channel number and a cell serial number. The footer contains a CRC value, flow control signals and an EOC, EOF or EOFE control character. The following non-payload data fields are contained in the cell structure.

- SOF or SOC – K-chacter which indicates if this is a normal cell (SOC) or the start of a new frame (SOF).
- VcSer – Per virtual channel cell serial number. This is used to detect lost cells.

- Vc – Vritual channel identifier.
- CRC Bytes – A 32-bit cell data CRC protection value. The CRC protected portion of the frame includes the cell header.
- VcFull – Flow control full flags, one per virtual channel.
- VcAFull – Flow control almost full flags, one per virtual channel.
- EOC, EOF or EOFE – K-character indicating a normal cell (EOC), a cell indicating end of frame (EOF), a cell indicating end of frame with error (EOFE).

## 5.5 Empty Cell Structure




PGP2 Empty Cell Data Structure

The empty cell structure is generated in place of a normal payload cell when user data is not read to be transmitted. The use of empty cells ensures that the flow control status bits are regularly updated even in the absence of transmitted data.