LogiCORE IP Fibre Channel v3.5

DS270 April 19, 2010 **Product Specification**

Introduction

The LogiCORE™ IP Fibre Channel (FC) core provides a flexible core for use in any non-loop FC port and can run at 1, 2, and 4 Gbps. The FC core includes credit management features as well as the FC (old) Port State Machine for link initialization.

Features

- Optional single or dual-speed FC core running at 1 Gbps (1062.5 Mb), 2 Gbps (2125 Mb), 4 Gbps (4250 Mb), 1/2 Gbps, negotiable; or 2/4 Gbps, negotiable.
- Common internal core clock frequency maintained at 53.125 or 106.25 MHz, dependent on communication rate.
- Designed to ANSI INCITS X3-230-1994 (R1999), X3-297-1997 (R2002), X3-303-1998 FC-PH and T11-FC-FS (v1.9) specifications.
- Supports Class 1, 2, 3, and F frames, as well as Class 4 frames in the Virtex®-4 and Virtex-5 families.
- Port-independent implementation supports underlying functionality for all non-arbitrated loop port types: N, F, E, and B.
- 32-bit client interface for maximum flexibility when interfacing to back-end applications.
- Optional generic management interface to access configuration registers and statistics.
- Supports CRC checking on received frames and optional CRC checking or insertion for transmitted frames.
- Uses Virtex-4 FPGA RocketIO™ Multi- Gigabit Transceivers (MGTs) or Virtex-5 FPGA RocketIO GTP transceivers for the serial interface.
- HDL wrapper provided with netlist includes IOBs, MGTs, and resetting and clocking circuitry to provide maximum flexibility for integrating the core into user designs; the wrapper also facilitates resource sharing across multiple cores.

LogiCORE IP Facts	
Core Specifics	
Supported Families[1]	Virtex-4[2], Virtex-5
Speed Grades	Virtex-4 -10/-11 speed grade (4VFX20 or larger)[3] Virtex-5 -2 speed grade (Any LXT/SXT part)
Performance	1.0625 Gbps, 2.125 Gbps, 4.250 Gbps
Core Resources	
Slices	1198-2674[4] or 798-1464[5]
LUTs	1370-3525[4] or 1120-2722 [5]
FFs	1068-1991[4,5]
DCMs	1-2[4,5]
BUFGs	3-8[4,5]
RocketIO transceivers	1-2[4,5]
Provided with Core	
Documentation	Product Specification Getting Started Guide, User Guide
Design File Formats	NGC netlist, scripts, HDL wrapper, demo test bench
Constraints File	User Constraints File (UCF)
Design Tools Requirements	
Supported HDL	VHDL, Verilog
Synthesis	XST 12.1
Xilinx® Tools	ISE® software v12.1
Simulation Tools[6]	Mentor Graphics ModelSim v6.5c and above Cadence Incisive Enterprise Simulator (IES) v9.2 and above
Core Highlights	
FC_PH Compliant	Hardware Verified

1. For the complete list of supported devices, see the 12.1 release notes for this core.
2. Virtex-4 FX FPGA solutions require the latest silicon stepping and are pending hardware validation. See Device Support for more information.
3. -11 for 4 Gbps with Statistics or Multispeed 2/4 Gbps with Statistics.
4. Can change with configuration. See Table 9 and Table 10.
5. Virtex-5 FPGA slices and LUTs are different from previous families. See Table 10.
6. Virtex-5 devices require a Verilog LRM-IEEE 1364-2005 encryption-compliant simulator. For VHDL simulation, a mixed HDL license is required.

Discontinued IP

© 2004-2010 Xilinx, Inc. XILINX, the Xilinx logo, Virtex, Spartan, ISE and other designated brands included herein are trademarks of Xilinx in the United States and other countries. The PowerPC name and logo are registered trademarks of IBM Corp. and used under license. All other trademarks are the property of their respective owners.

Features (Continued)

- Optional buffer-to-buffer credit management support
- Optional statistics-gathering block
- Optional speed-negotiation block
- Optional programmable parity checking of transmit data
- Designed for use in a non-arbitrated loop topology, with higher-level port and class-specific functions provided by user modules
- Available under the SignOnce IP Site License

Device Support

Note: Speeds greater than 2 Gbps are supported only in Virtex-4 devices.

The Fibre Channel core is designed to work with the RocketIO MGT transceivers for the Virtex-4 FX family and RocketIO GTP transceivers for the Virtex-5 LXT/SXT families. The Xilinx CORE Generator™ software restricts generating the core to those devices with sufficient resources for the example design.

Speed Grade

The speed grade for the core is determined by the selected configuration. For Virtex-4 devices, 4 Gbps operation is only supported in –11 parts, while all other configurations can be targeted at –10 parts. For Virtex-5 devices, all valid configurations can be targeted at the -2 speed grade parts.

Overview

Figure 1 illustrates the FC core as part of the Fibre Channel architecture. The highlighted sections indicate the blocks supported by the core: FC-0, FC-1 and part of FC-2. FC-1 and FC-2 functionality supported by the core includes the FC Port State Machine (PSM) and optional buffer-to-buffer credit management, with automatic BB_SCx credit recovery. Figure 2 displays the top-level block diagram for the single-speed implementation of the core.

Discontinued IP

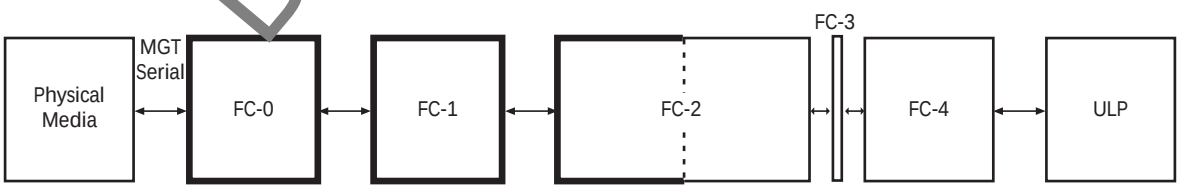


Figure 1: **Fibre Channel System Level Block Diagram**

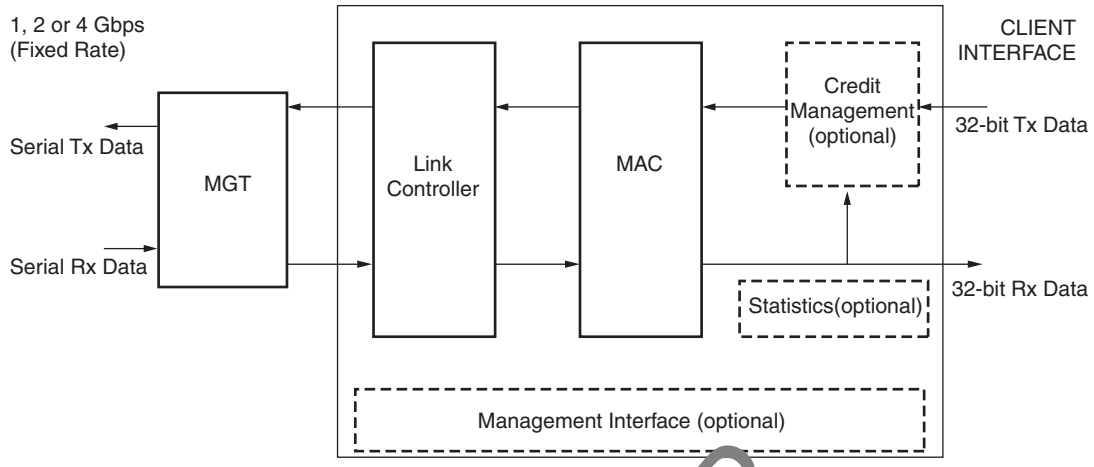


Figure 2: **Single-speed Architecture**

Applications

The FC core is designed to be used in a non-arbitrated loop topology with higher-level port and class-specific functions provided by user modules.

Figure 3 shows an example of other modules that may be required to implement an FC port: Node (N), Fabric (F), Extension (E), or Bridge (B). Firmware is a consideration, and the diagram shows how an entire FC port may be implemented on a single Virtex-4 device using the embedded PowerPC® 405 processor. The backend includes frame buffers, FIFOs, DMA, and other features.

The FC core can function in any non-loop scenario. The ability of the dual-speed core to transmit and receive at independent speeds (1/2 or 2/4 Gbps) allows the optional implementation of *FC-FS* Section 28 (speed negotiation).

Discontinued IP

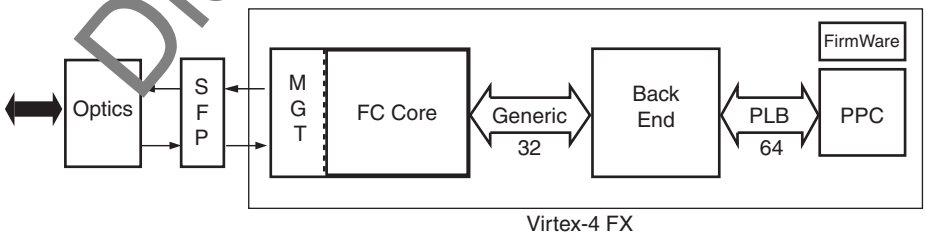


Figure 3: **Example Application Block Diagram**

Core Architecture

Figure 2 displays the internal architecture of the single-speed FC core, and Figure 4 displays the architecture of the dual-speed core.

The block diagrams show the major functional blocks of the FC core:

- RocketIO Transceiver(s)
- Client Interface
- Management Interface (optional)
- Credit Management (optional)
- MAC
- Link Controller
- Statistics Module (optional)
- Speed Negotiation (optional)

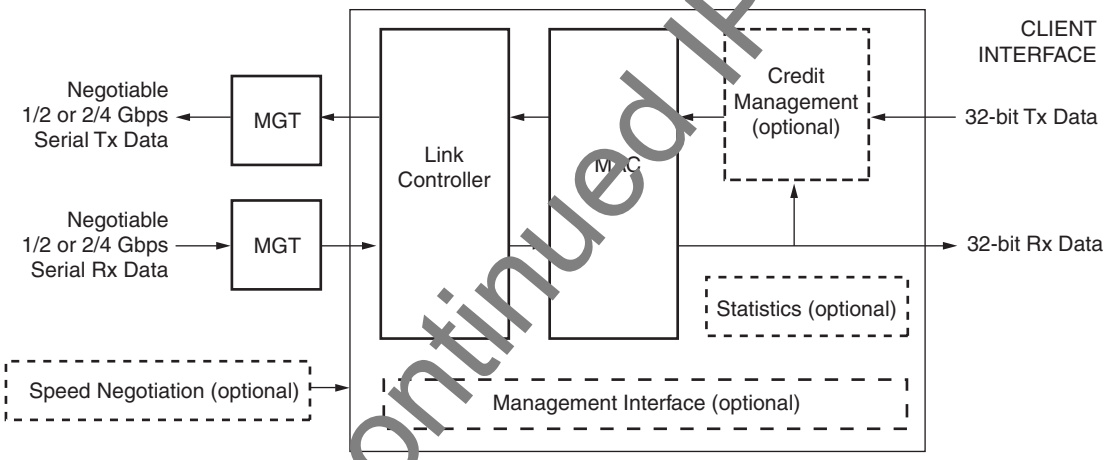


Figure 4: **Dual-speed Architecture**

Discontinued IP

RocketIO Transceivers

The FC core uses one or two of the device-specific RocketIO Transceivers to provide the 1, 2, and 4 Gbps connectivity required by the interface. The core also makes use of the 8B/10B encoder/decoder, CRC generator/checker, and the receiver elastic buffer in the Virtex-4 FPGA MGTs and Virtex-5 FPGA GTP transceivers, allowing the core to run in a single clock domain resulting in the simplest back-end design. A single transceiver can be used for single-speed modes for all Virtex-4 and Virtex-5 FPGA configurations.

Client Interface

The internal interface for the core is 32-bits wide, allowing FC data to be analyzed one word at a time. This 32-bit data path is preserved through to the Client Interface to ensure maximum flexibility in back-end interfacing. The Client Interface contains a set of additional signals required to support the main data traffic. The core clock is fixed at either 53.125 MHz or 106.25 MHz, depending on channel operation. At the 1 Gbps rate in single-speed and multi-speed 1/2 Gbps configuration and 2 Gbps rate in multi-speed 2/4 Gbps configuration, `clienttxdataread` and `clientrxdatavalid` oscillate high and low on successive clock cycles to throttle the data throughput.

Management Interface (Optional)

Configuration of the core and access to the statistics block can be provided through the optional Management (Host) Interface, a 32-bit processor-neutral data pathway independent of the FC data pathway. When the Management Interface is omitted, configuration of the core can still be made using a configuration vector, and statistics may still be collected outside the core using a statistics vector. The Management Interface is synchronous to the core clock.

Credit Management (Optional)

The optional Credit Management block provides simple buffer-to-buffer credit management, keeping track of `r_rdy` and `sof` primitives received and sent and supporting BB_SCx-based credit recovery as defined in *FC-FS* Section 18.5.11.

MAC

The MAC block provides the main functionality of the core and consists of the Port State Machine (PSM) as well as the framing control and checking of the data. Designed to *FC-FS* Section 7.

Link Controller

The Link block provides the word alignment and synchronization of the incoming data and provides CRC generation and checking on outgoing data. Designed to *FC-FS* Sections 5 and 6 (equivalent to *FC-PH* Sections 11 and 12).

Statistics (optional)

The optional Statistics block collects and stores statistical information in memory in the core. This needs to be polled regularly to avoid the counters saturating. Statistics may still be collected outside of the core using the statistics vector when this optional block is not included in the core.

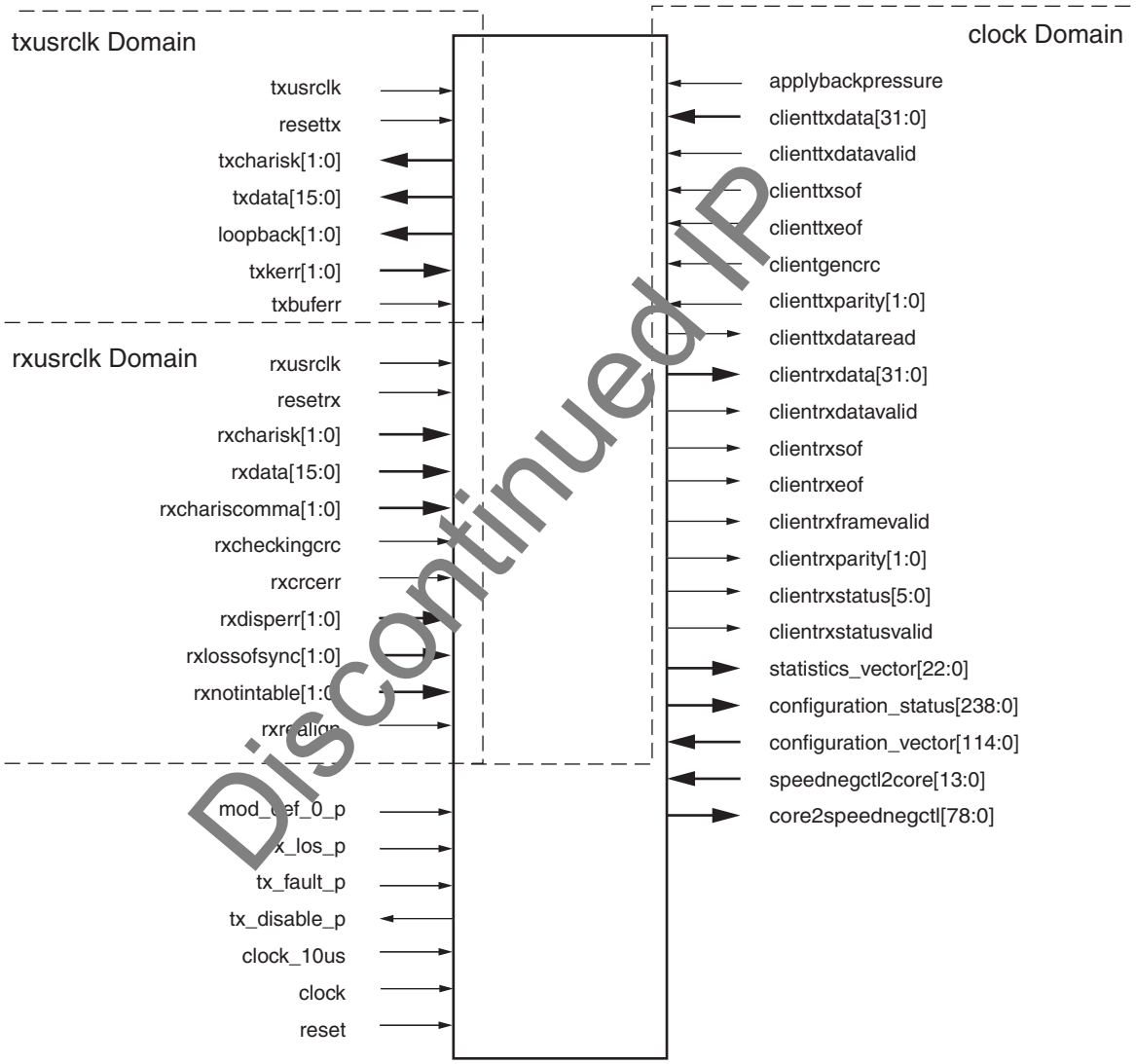
Speed Negotiation (Optional)

The optional Speed Negotiation block provides the ability for a dual-speed core to implement the *FC-FS* Section 28 speed negotiation algorithm.

Discontinued IP

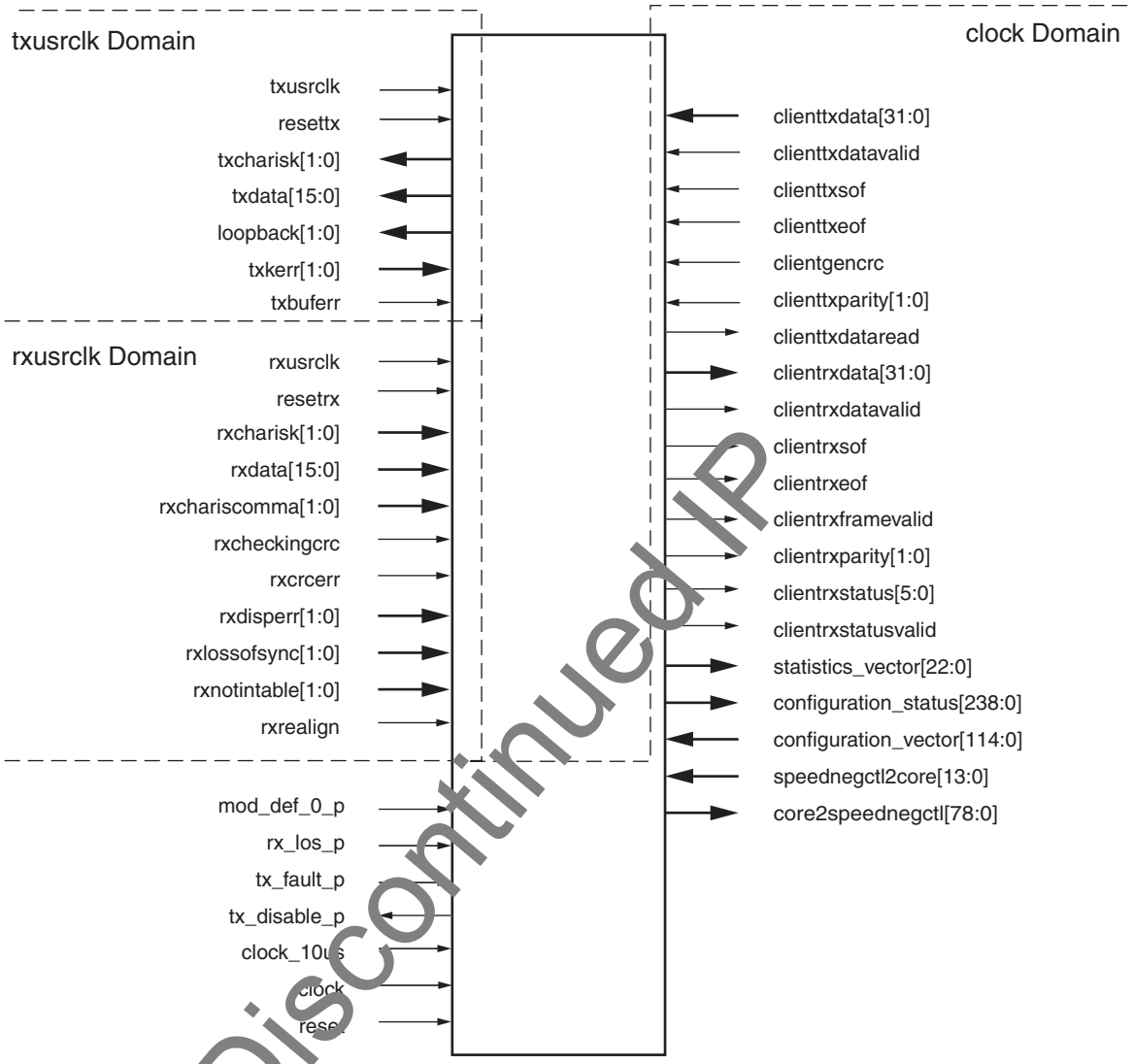
Interface Descriptions

Figures 5 through 10 display pinouts for the various core configuration options. The `c_speed` and `c_has_stats` options make no real difference to the pinout and for this reason are shown with and without the Management Interface, credit management, and speed negotiation blocks. The signals are split into their respective clock domains. In some implementations, some of these clock domains may be the same as each other, and no re-clocking is required between them. For information about clocking schemes for Virtex-4 and Virtex-5 devices, see Chapter 6, "Design Considerations," in the *Fibre Channel User Guide*.



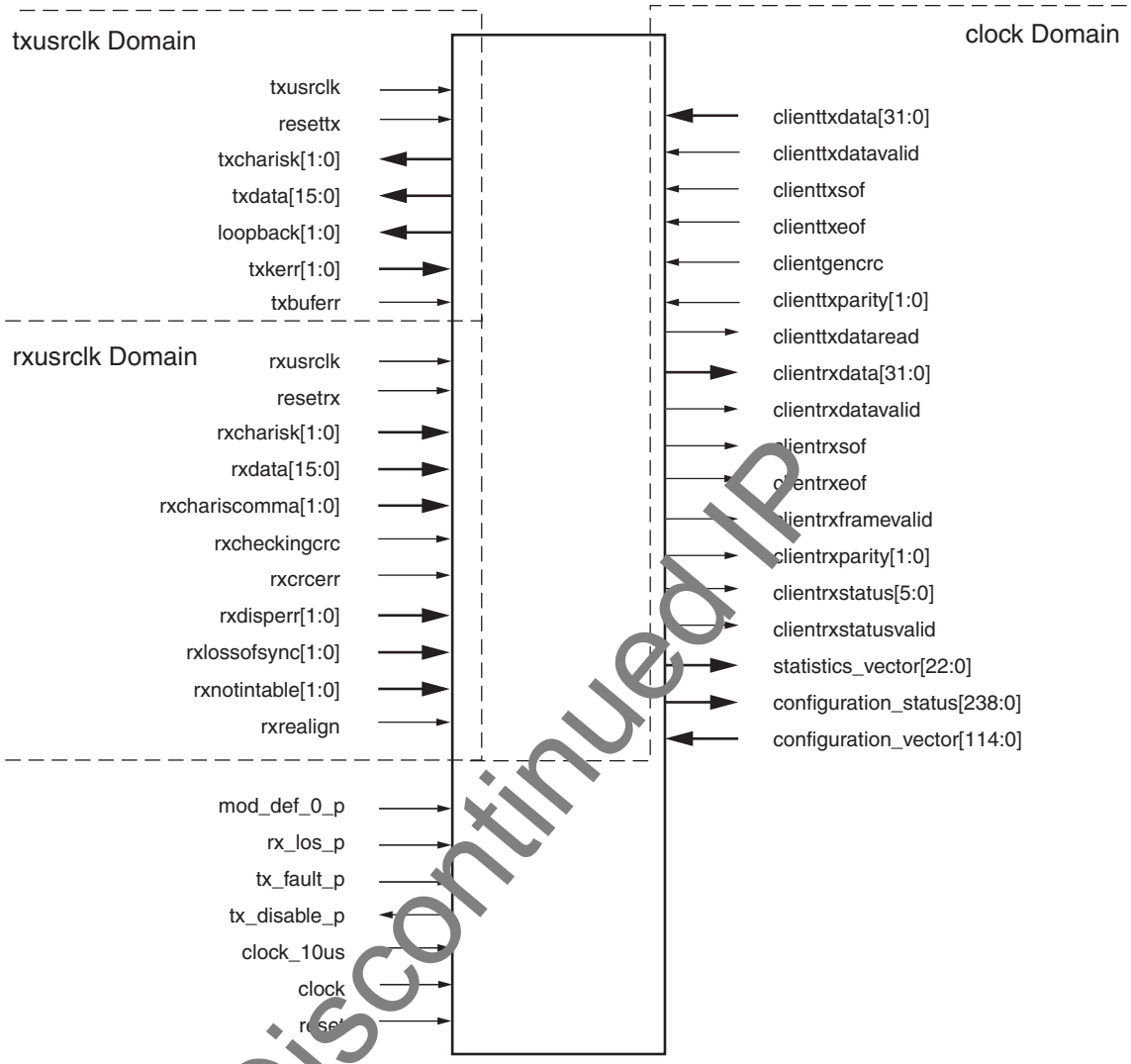
Discontinued IP

Figure 5: **Pinout Without Management Interface**



Discontinued IP

Figure 6: **Pinout Without Management Interface: No Credit Block**



Discontinued IP

Figure 7: **Pinout Without Management Interface: No Credit Block or Speed Negotiation**

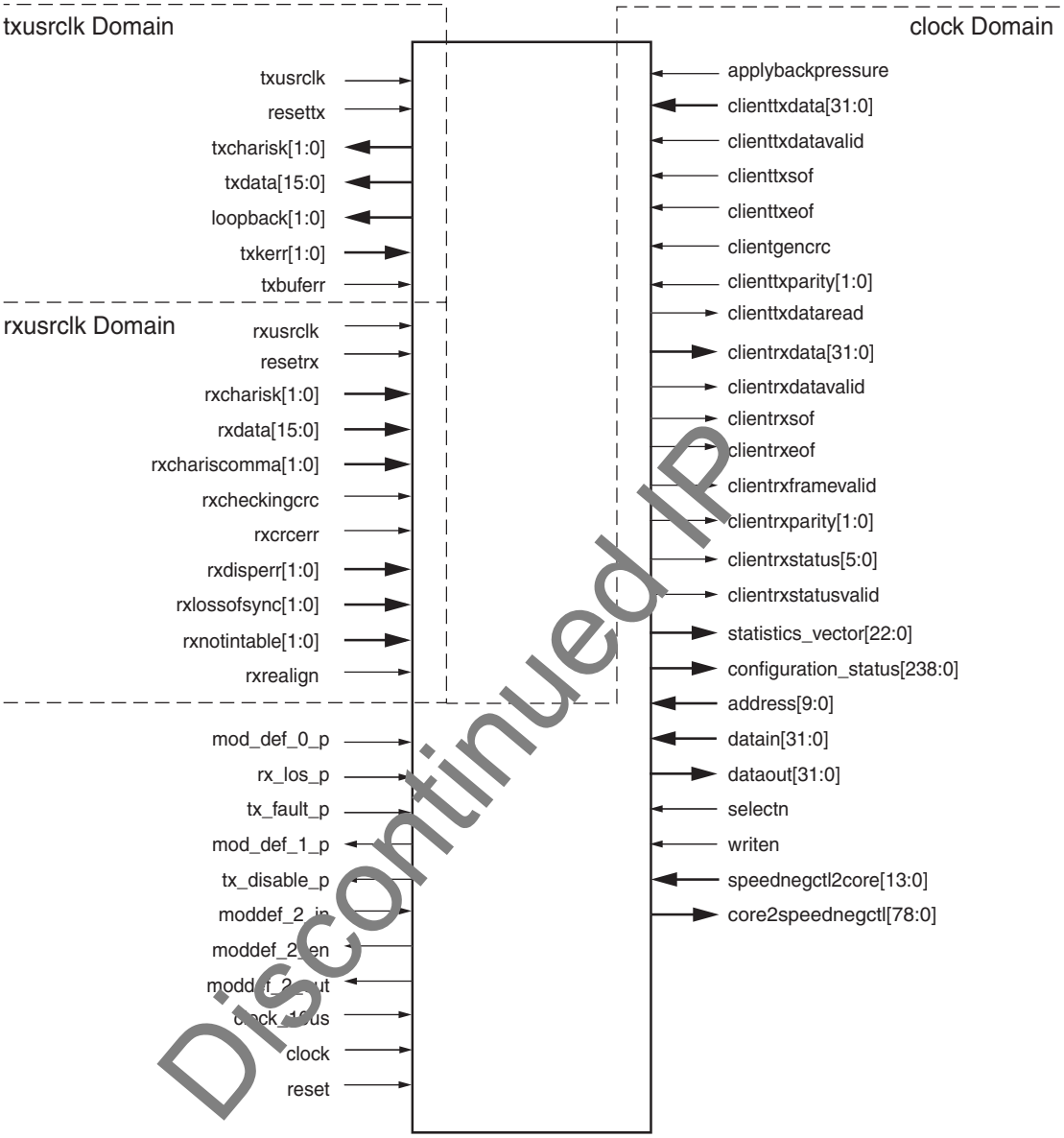


Figure 8: **Pinout With Management Interface**

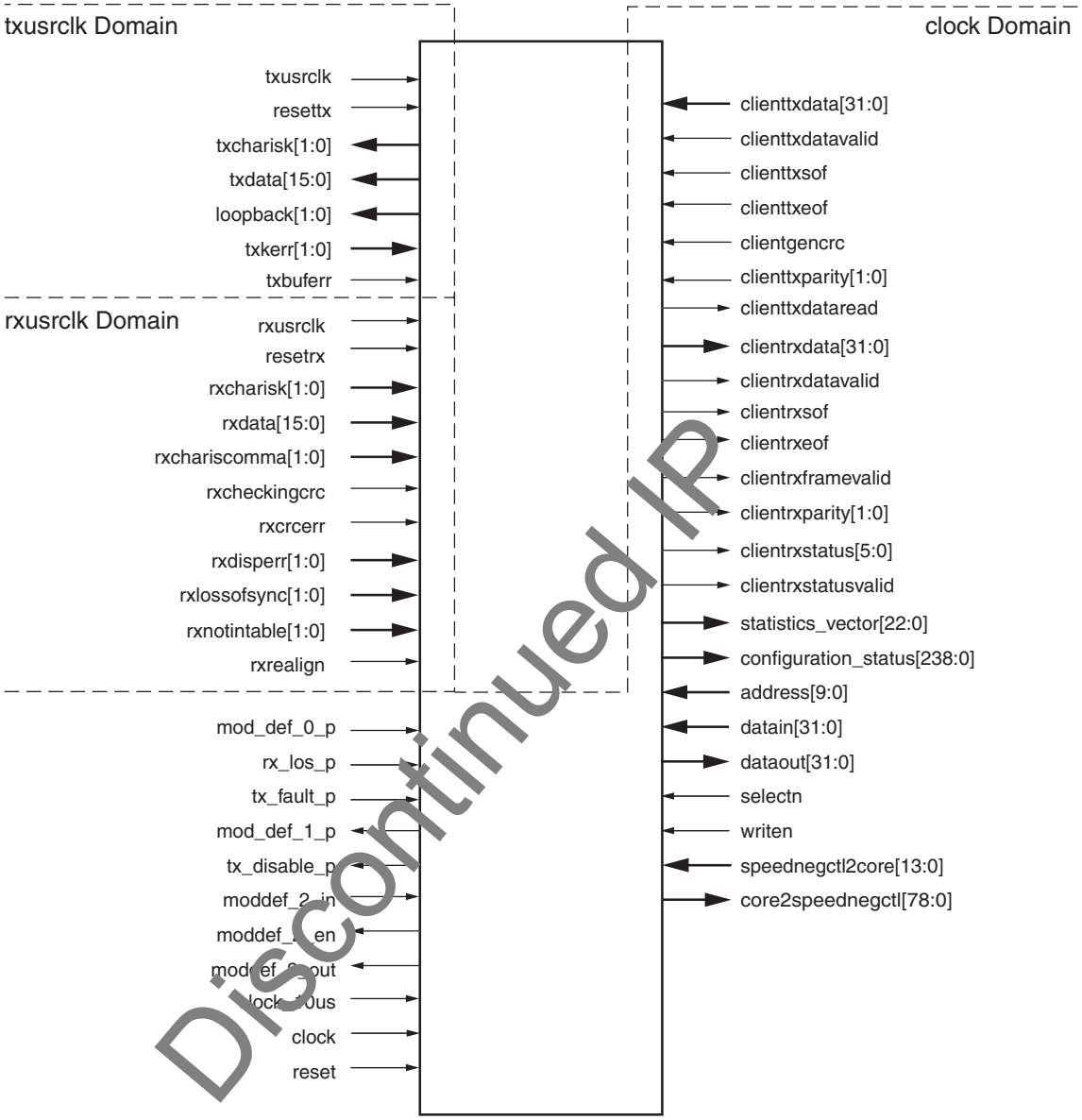
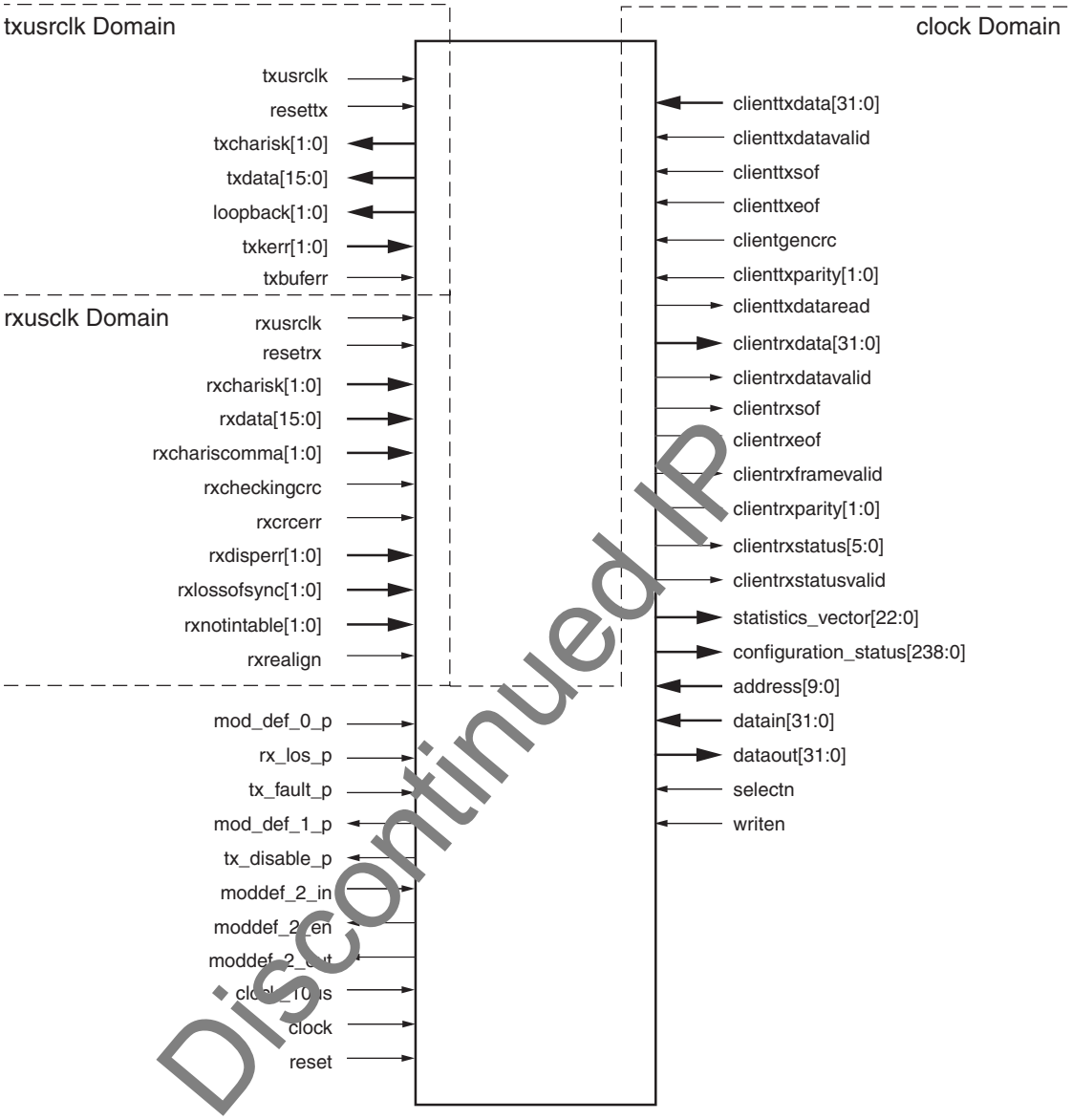


Figure 9: **Pinout With Management Interface: No Credit Block**



Discontinued IP

Figure 10: **Pinout With Management Interface: No Credit Block or Speed Negotiation**

Client Side Interface Signal Definition

Table 1 defines the client-side interface signals of the FC core. For detailed information about signal definitions, see the *Fibre Channel User Guide*.

Table 1: **Client Interface Signal Pins**

Signal[1]	Direction	Description
applybackpressure	Input	Apply back pressure to the attached FC device. Stops R_RDYs being transmitted if high.
clienttxdata[31:0]	Input	Transmit Data coming from Client.
clienttxdatavalid	Input	'1' when the clienttxdata word is valid. Used to detect Client Underflow.
clienttxsof	Input	'1' when the clienttxdata word is an SOF.
clienttxeof	Input	'1' when the clienttxdata word is an EOF.
clientgencrc	Input	'1' when a CRC is generated for this frame by the core.
clienttxparity[1:0]	Input	Parity vector for clienttxdata, one bit per byte-pair.
clienttxdataread	Output	'1' when the clientxdatabus has been read.
clientrxdata[31:0]	Output	Receive Data going to Client.
clientrxdatavalid	Output	When '1' the clientrxdata is valid.
clientrxsof	Output	When '1' the clientrxdata word is an SOF.
clientrxeof	Output	When '1,' the clientrxdata word is an EOF.
clientrxframevalid	Output	'1' when the current data is within an FC frame.
clientrxparity[1:0]	Output	Parity vector for clientrxdata, one bit per byte-pair.
clientrxstatus[5:0]	Output	Indicates the status of the FC frame. See Table 8.
clientrxstatusvalid	Output	'1' when clientrxdata is valid.

1. All signals in this table should be synchronous to clock and active high unless otherwise noted.

Table 2 defines the Management Interface and support signals. These signals are used by the client to configure the FC core and to read the status of configuration bits and statistics counters.

Table 2: **Management Interface Signal Pins**

Signal[1]	Direction	Description
address[9:0]	Input	Management Address bus
datain[31:0]	Input	Management Data bus in
selectn	Input	Management Enable signal - Active Low
writen	Input	Indicates the type of access '1' - Read '0' - Write
dataout[31:0]	Output	Management Data bus out
statistics_vector[22:0]	Output	Statistics Increment vector
configuration_status[238:0]	Output	Configuration register status vector
configuration_vector[114:0]	Input	Alternative configuration loading vector

1. All signals in this table should be synchronous to clock and active high unless otherwise noted.

Discontinued IP

Table 3 defines the system-level signals for clocking and resetting.

Table 3: **Clock and Reset Signals**

Signal	Direction	Description
clock_10us	Input	10µs clock signal for various timers - synchronous to clock
clock	Input	Core clock - 53.125/106.25 MHz
txusrclk	Input	Transmit clock - 53.125/106.25/212.5 MHz
rxusrclk	Input	Receive clock - 53.125/106.25/212.5 MHz
reset	Input	Synchronous reset for clock
resettx	Input	Synchronous reset for txusrclk
resetrx	Input	Synchronous reset for rxusrclk

Other clocks are derived within the example design from this master clock. In Virtex-4 and Virtex-5 devices, a 212.5 MHz clock must be supplied for all configurations. Again, other clocks are derived from this master clock in the example design.

Speed Negotiation Signal Definition

Table 4 defines the speed negotiation signals, used by the core and speed negotiation controller to run the speed negotiation algorithm as defined in *FC-FS* Section 28. For detailed information, see the *Fibre Channel User Guide*.

Table 4: **Speed Negotiation Interface Pinout**

Signal[1]	Direction	Description
speednegctl2core[13:0]	Input	Speed negotiation control to core vector
core2speednegctl[78:0]	Output	Core to speed negotiation control vector

1. All signals in this table should be synchronous to clock and active high unless otherwise noted.

Discontinued IP

Physical Interface Signal Definition

Table 5 defines the RocketIO MGT and GTP transceiver signals of the FC core. These signals attach to the one or two RocketIO transceivers, as illustrated in the example design included with the core. See the *Fibre Channel User Guide* for more information.

Table 5: **RocketIO Transceiver Interface Pinout**

Signal	Direction	Description
txcharisk[1:0]	Output	Indicates which byte lanes have a K Character.
txdata[15:0]	Output	Transmit 16-bit Word Data.
loopback[1:0]	Output	Selects the two loopback test modes. Bit 1 is for serial loopback and bit 0 is for internal parallel loopback. Set to '1' to select loopback mode, '0' for normal operation.
txkerr[1:0]	Input	'1' indicates that a byte lane has a K-character to be transmitted which is not a valid K-character.
txbuferr	Input	Provides status of the transmission FIFO. If '1,' an overflow/underflow has occurred. When this bit is '1,' it can only be reset by asserting resettx.
rxcharisk[1:0]	Input	Indicates which byte lanes have a K Character.
rxdata[15:0]	Input	Receive 16-bit Word Data.
rxchariscomma[1:0]	Input	Similar to rxcharisk except that the data is a comma.
rxcheckingcrc	Input	CRC status for the receiver. '1' indicates that the receiver has recognized the end of a data packet.
rxcrcerr	Input	'1' indicates that the CRC code is incorrect when asserted high.
rxdisperr[1:0]	Input	'1' indicates that a disparity error has occurred on the serial line.
rxlossofsync[1:0]	Input	Indicates the state of the FSM: Bit 1 = Loss of sync (active high) Bit 0 = Resync state (active high).
rxnotintable[1:0]	Input	Status of encoded data - Indicates which byte lane(s) contain an invalid character.
rxrealign	Input	Signal from the PMA denoting that the byte alignment with the serial data stream changed due to a comma detection. '1' when alignment occurs.

Discontinued IP

Table 6 describes the physical media interface of the FC core. These signals are typically connected to an external optical module. An example is the Finisar FTRJ8519P1xNL module.

Table 6: **Physical Media Interface Signal Pinout**

Signal	Direction	Description
mod_def_0_p	Input	Module Definition 0: '0' - module is present External I/O Pad.
rx_los_p	Input	Loss of Signal External I/O Pad.
tx_fault_p	Input	Indicates the optical interface has a transmit fault External I/O Pad.
mod_def_1_p	Output	Module Definition 1: Serial Clock External I/O Pad.
tx_disable_p	Output	Disables the transmit laser when active External I/O Pad.
moddef_2_in[1]	Input	Module Definition 2: Serial Data In.
moddef_2_en[1]	Output	Module Definition 2: Serial Data Out Tri-state Enable.
moddef_2_out [1]	Output	Module Definition 2: Serial Data Out.

1. A tristate buffer is provided in the example design code to create the single MOD_DEF_2 signal required.

Discontinued IP

Functional Description

Client Interface

The data pathway is 32 bits wide in both transmit and receive directions and is synchronous to the core clock at 53.125/106.25 MHz. The Client Interface is the same for both transmitter and the receiver with a constant core clock frequency no matter what speed of operation. The difference between the speeds of operations is only apparent at the Client Interface: the frequency at which the core can accept new data to transmit or provide new data it has received. Table 7 defines the timing diagram abbreviations.

Table 7: **Timing Diagram Abbreviations**

Abbreviation	Definition
SOF	Start of frame
H(0-5)	Header word
D(0-n)	Data word
$P_1P_0P_x$	Valid 16- bit parity bits
CRC	Cyclic redundancy check sequence
EOF	End Of Frame
X	Don't care/unknown
xF4	Hex value

Discontinued IP

Transmitter

Normal Frame Transmission

The signal `clienttxsof` is raised at the start of the frame, and lowered once `clienttxdataread` is raised to indicate that the core has accepted the data for reading (after a single clock-tick in the timing example, although this may vary). `clienttxeof` is raised at the same time as the EOF word is present on `clienttxdata`. The `clienttxdatavalid` signal is raised concurrently with `clienttxsof` and held high for the duration of the frame. The signal `clientgencrc` is held high throughout the frame transfer to indicate that the CRC is not being included with the frame. You must provide valid SOF and EOF words to the core, as these are not checked by the core. The timing of a normal 2 Gbps or 4 Gbps frame transmission across the Client Interface is shown in Figure 11. The only difference between single-speed transmission at 2 Gbps and 4 Gbps are the clock frequencies.

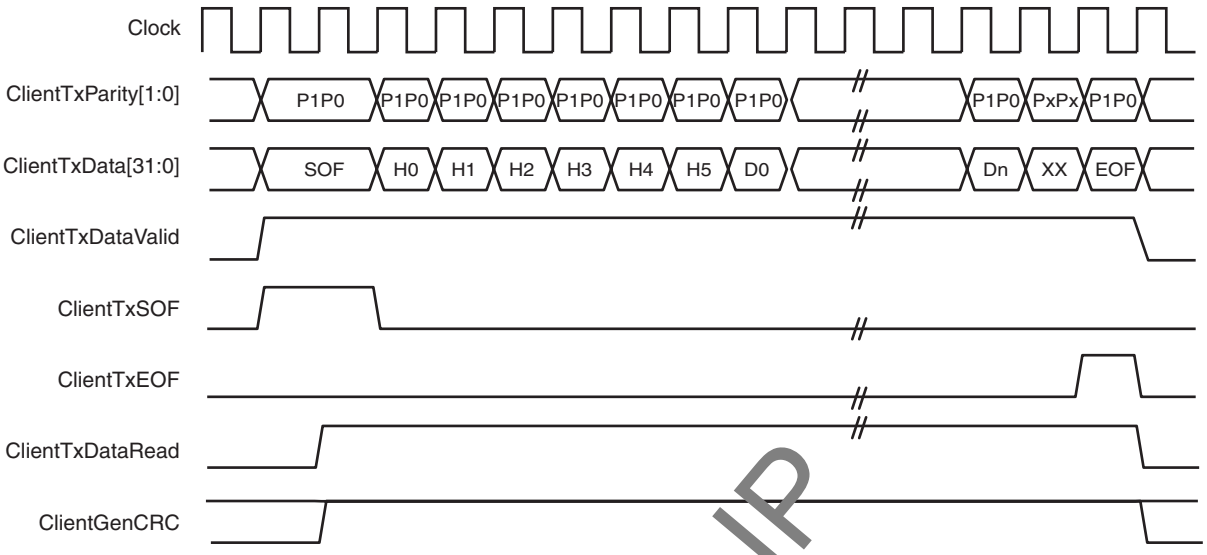


Figure 11: **Normal 2 Gbps/4 Gbps Frame Transmission Across Client Interface**

The timing of a 1 Gbps frame transmission across the Client Interface is shown in Figure 12. Note how new data is accepted every other clock by the `clienttxdataread` signal toggling every clock tick. This results in half the transfer rate compared to the single-speed 2 Gbps version. This also applies to the multi-speed 2/4 Gbps configuration running at 2 Gbps.

Discontinued IP

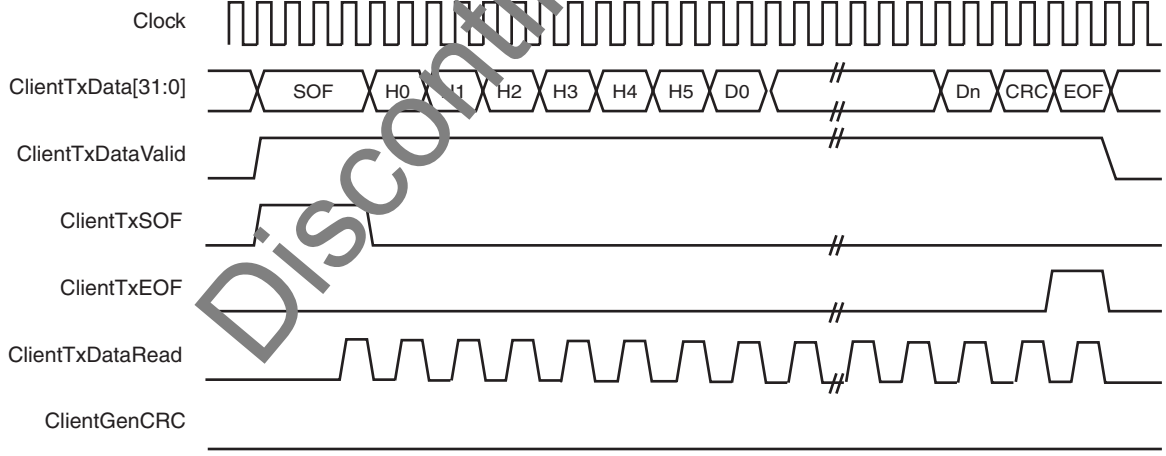


Figure 12: **Normal 1 Gbps Frame Transmission Across Client Interface**

See "CRC Generation" on page 18 for more information about the use of `clientgencrc`. The only other difference between the 1 Gbps timing diagrams and the single-speed 2 Gbps timing diagrams is the oscillation of `clienttxdataread`. For this reason, all further timing diagrams are based on single-speed 2 Gbps operation, from which the 1 Gbps timing results can be inferred.

TX Parity

If parity checking is not disabled, the core is required to receive 16-bit parity information with each word of data. There are two parity bits; the least significant bit relates to the parity of the least significant 16 bits in `clienttxdata`, and the most significant bit relates to the parity of the most significant 16 bits in `clienttxdata`. Figure 11 illustrates an example of the use of parity; the use of parity is not shown in further timing diagrams.

To create the correct TX Parity, use the formulae:

```
txparity(0) = not(xor(clienttxdata[15..0]))
txparity(1) = not(xor(clienttxdata[31:16]))
```

CRC Generation

It is possible for the core to generate the CRC for the frame transmission. To enable this mode, the `clientgencrc` signal needs to be asserted along with `clienttxdatavalid`. One word (with valid parity bits if TX Parity checking is enabled) still needs to be transferred to the core as a placeholder for the CRC. The value of this word is ignored and overwritten with the calculated CRC by the core before transmitting on the physical interface. Figure 11 also illustrates this operation.

Client Underflow

Figure 13 displays an abnormal frame transmission with a delay in new data after H2 for one clock cycle. This frame will be invalidated by the core, which inserts a EOFni code, and is the responsibility of the client to schedule retransmission. As a general rule, `clienttxdatavalid` should be held high throughout the Transmit frame.

After a new `clienttxsof` is transferred to the core, a new frame transmission starts as normal.

Discontinued IP

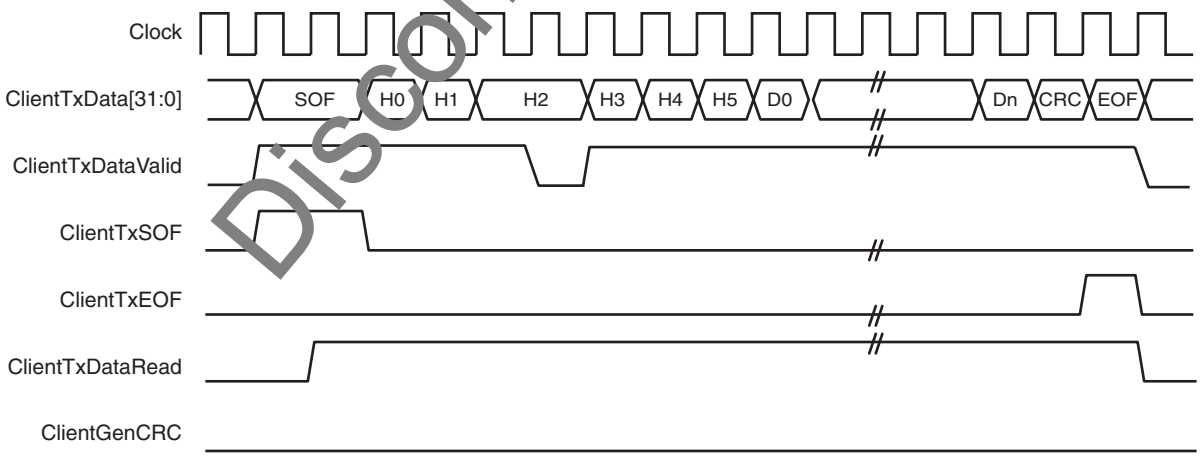


Figure 13: **Underrun 2 Gbps Frame Transmission Across Client Interface**

Receiver

Normal Frame Reception

`clientrxsof` is raised at the start of the frame, while the SOF word appears on `clientrxdata`. `clientrxeof` is raised at the same time as the EOF word is available on `clientrxdata` at the end of the frame. In 2 Gbps operation, `clientrxdatavalid` is raised with the SOF and held high for the duration of the frame, including the EOF. In 1 Gbps operation and multi-speed 2/4 Gbps running at 2 Gbps, `clientrxdatavalid` toggles every clock cycle, and the data changes every other clock cycle so that data appears on the Client Interface at the correct rate. Figure 14 illustrates the timing of a normal single-speed 2 Gbps frame reception across the Client Interface.

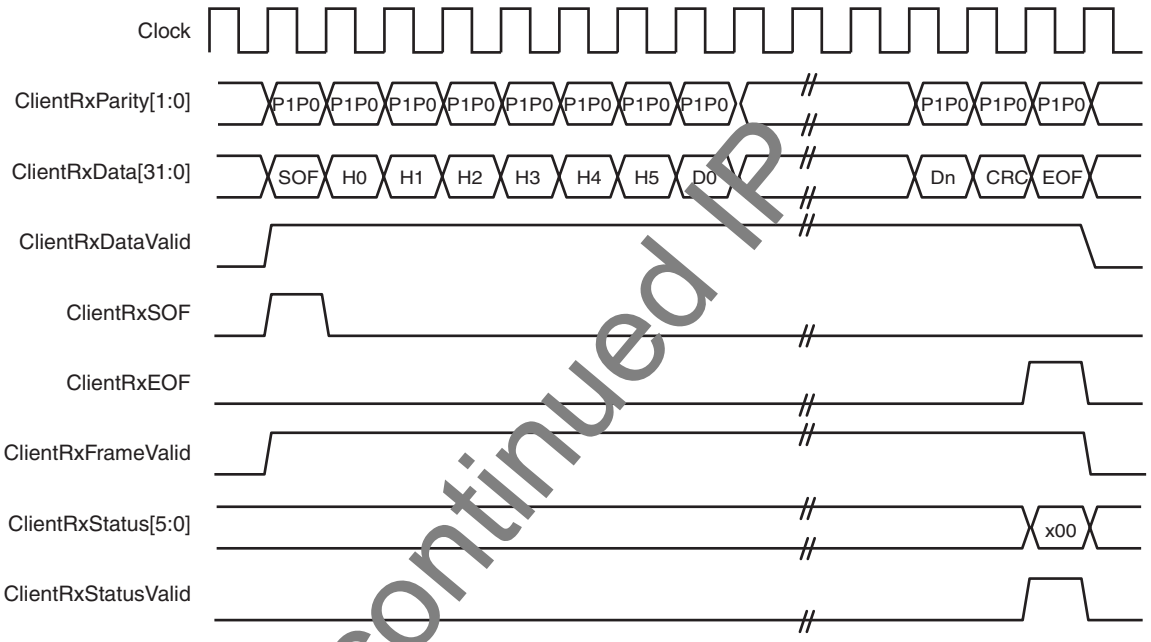


Figure 14: **Normal 2/4 Gbps Frame Reception Across Client Interface**

The core performs some analysis on the frame as it is being received and provides information on the frame to the client on the `clientrxstatus` vector. Table 8 defines this vector.

Table 8: **ClientRxStatus Definition**

ClientRxStatus	Description
Bit 0	CRC Error
Bit 1	Illegal Transmission Word
Bit 2	Undersized Frame Error
Bit 3	Oversized Frame Error
Bit 4	Invalid EOF
Bit 5	Frame received while not in Active State

`clientrxstatusvalid` validates this vector and also indicates the termination of the frame, even when the EOF is missing or late (see Figures 16, 17, and 18).

Discontinued IP

clientrxframevalid is asserted high with clientrxsof and stays high until the signal clientrxstatusvalid is asserted and indicates the framing of the received frame.

Whether or not parity checking on the Transmit path is disabled, the core provides parity information with each word of RX data. There are two parity bits; the least significant bit relates to the parity of the least significant 16 bits in clientrxdata, and the most significant bit relates to the parity of the most significant 16 bits in clientrxdata. Figure 14 illustrates an example of the use of parity; the use of parity is not shown in further timing diagrams.

Figure 15 illustrates the timing of a normal 1 Gbps frame reception across the Client Interface. clientrxdatavalid changes every clock cycle, and clientrxdata changes every other clock cycle. The other control signals align with the data. This is the only difference between 1 Gbps operation and single-speed 2 Gbps operation. All further timing diagrams illustrate 2 Gbps operation, from which 1 Gbps operation can be inferred.

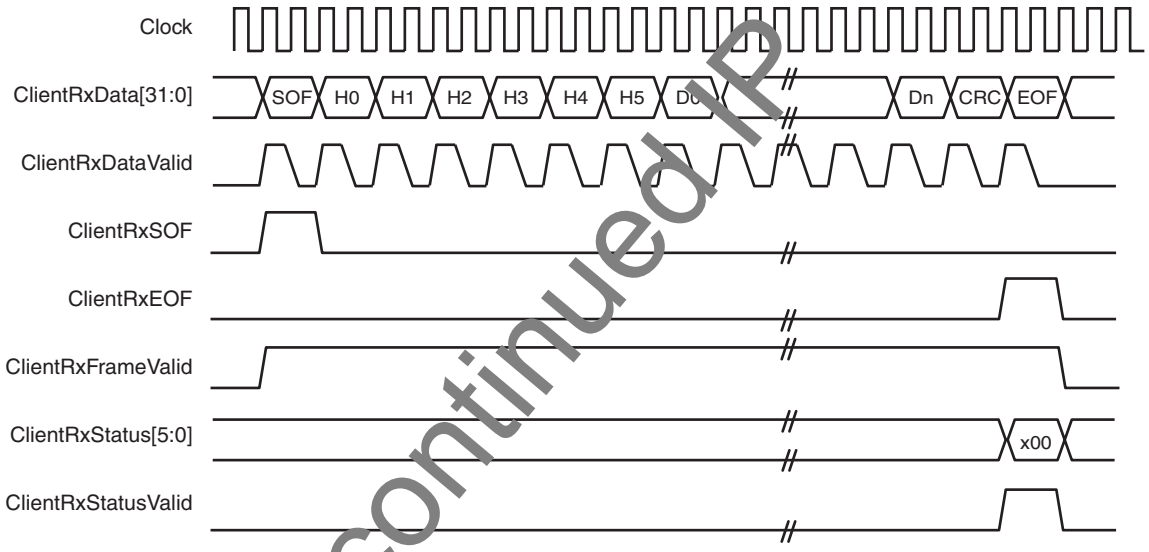


Figure 15: **Normal 1 Gbps Frame Reception Across Client Interface**

Discontinued IP

Abnormal Frame Reception

Under normal circumstances, the clientrxstatusvalid rises coincidentally with clientrxeof to mark the end of the frame. However, there are a number of situations where this operation differs, as described in the following sections.

Maximum Frame Length Exceeded

If the maximum frame length (528 payload words) is exceeded on reception, `clientrxstatusvalid` is raised and `clientrxstatus` identifies the error in the frame, even though this may not be coincident with `clientrxeof`. Data continues to pass to the client until the EOF appears, but `clientrxframevalid` remains low, as illustrated in Figure 16.

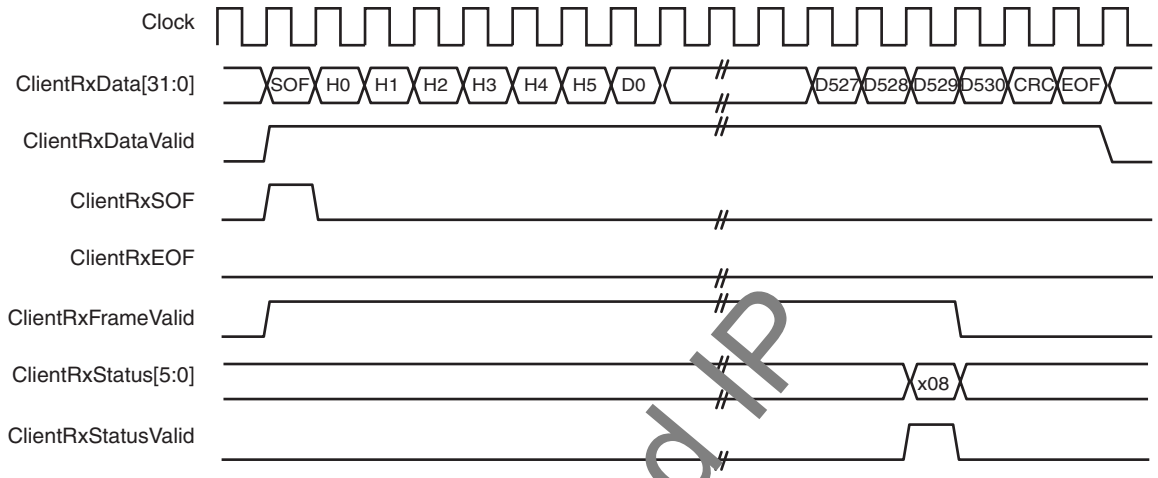


Figure 16: **Abnormal 2 Gbps Frame Reception across Client Interface with Maximum Frame Length Exceeded**

Non-Data Word Received

If a non-data word is received during a frame reception, the frame is invalidated. The core raises `clientrxstatusvalid` at this point and indicates the error in `clientrxstatus`. Figure 17 illustrates an example of this situation with an invalid character received instead of H3. Bad SOF and EOF words are handled differently; a bad SOF is not detected, while a bad EOF is considered missing, as described in the following section.

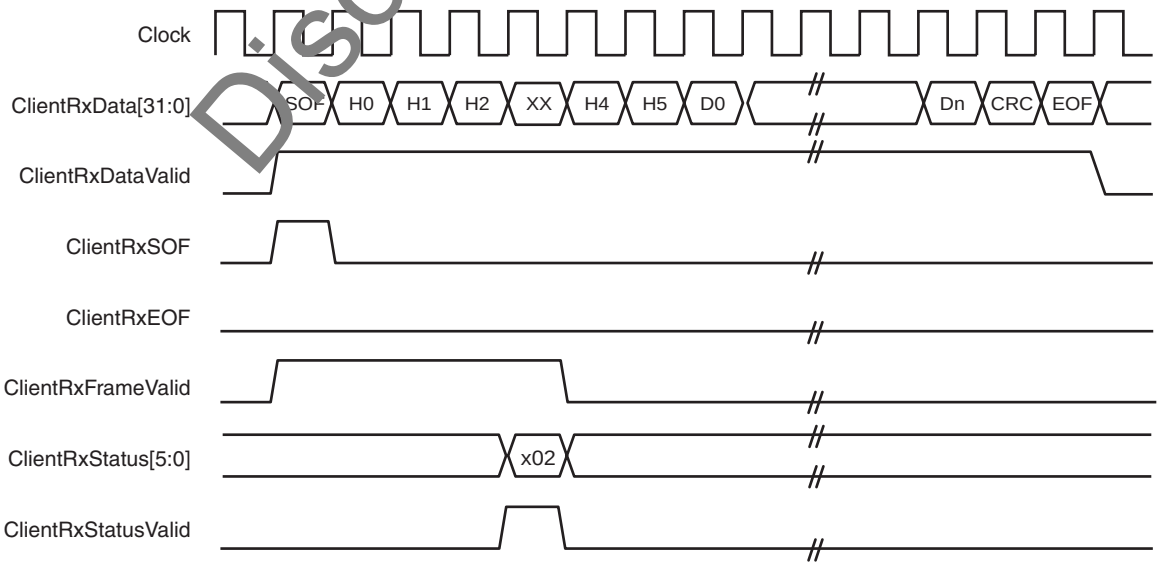


Figure 17: **Abnormal 2 Gbps Frame Reception Across Client Interface with Non-Data Word**

Discontinued IP

Missing/Bad EOF

If a frame is received without an EOF, `clientrxstatusvalid` is raised after the last data word with `clientrxstatus` indicating the error to the client. If the EOF is missing and the interframe gap (IFG) words have also been removed, `clientrxstatusvalid` may be raised at the same time as new `clientrxsof`, or before, to terminate the previous frame with an error and still allow the new frame to be correctly received. In this situation, `clientrxframevalid` remains high between the two frames, as displayed in Figure 18.

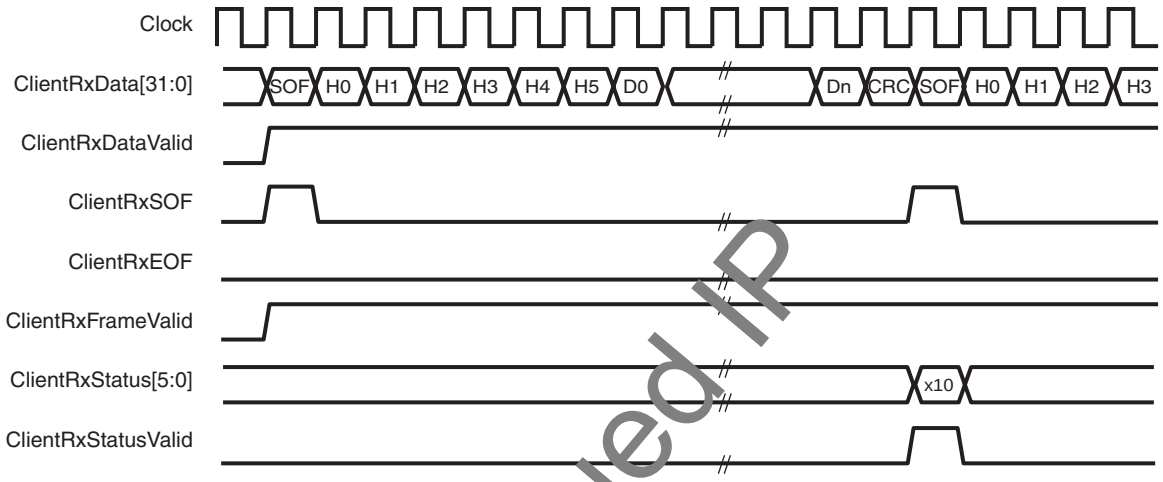


Figure 18: **Abnormal 2 Gbps Frame Reception across Client Interface with Missing EOF and IFG**

Inter Frame Gap

The core default is to enforce the minimum Inter Frame Gap (IFG) of 6 words on the Transmit path, delaying transmission of Client frames as required. `r_rdy` primitives may be inserted into the IFG by the core, with a minimum of two IDLE words before and after each `r_rdy`. The core will operate correctly with the minimum IFG of two words on the Receive path. The values for IFG and `r_rdy` separation are programmable and may be set to a user-defined value using the Management Interface or Configuration Vector. For more information, see Chapter 4, "Designing with the Core" in the *Fibre Channel User Guide*.

Discontinued IP

RocketIO Transceivers

The core uses the device-specific RocketIO transceivers to provide the serial communication, which links with the rest of the core through a 16-bit interface. The device-specific RocketIO transceivers are highly customizable, and can be configured to suit different standards and real-world conditions.

To provide greater flexibility to the user, the core does not integrate the device-specific RocketIO transceivers into the design, rather, it provides the 16-bit interface and associated control signals to link directly with the one or two transceivers.

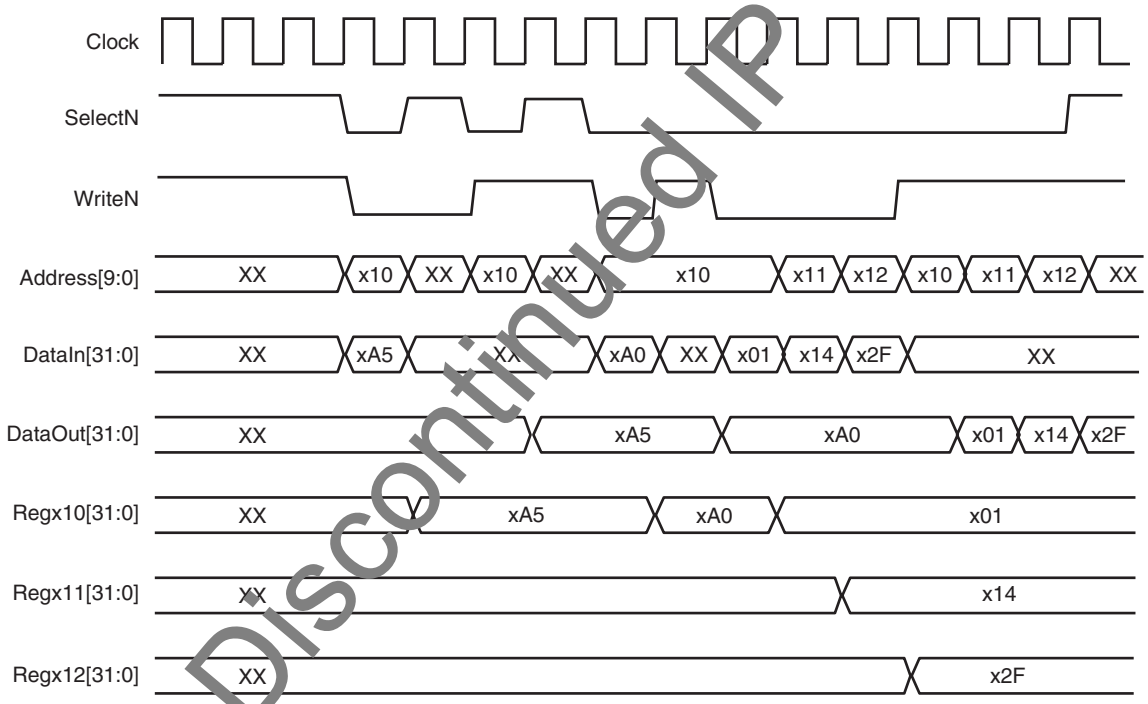
An example of how to connect the transceivers to the core to create a complete solution is provided in the example design provided with the core. For detailed information about the associated signals and the attributes that can be configured for each supported device family, see the following user guides: *Virtex-4 FPGA RocketIO Multi-Gigabit Transceiver User Guide* and the *Virtex-5 FPGA GTP Transceiver User Guide*. In addition, see "Wrapper Files" in Chapter 6 of the *Fibre Channel User Guide*.

Management Interface

The Management Interface is processor-independent with generic address, data, and control signals. It may be used as is, or a simple adapter may be created (not supplied) to interface to common bus interfaces.

Configuration Register Access

Figure 19 illustrates the timing of the Management Interface signals for Configuration Register access. The `selectn` signal starts and stops a given transaction. On the rising edge of the `clock` signal, if the `writen` signal is active (low) and the `selectn` signal is active (low), the data on the `datain` bus is written to the register indicated by the `address` bus. If the `writen` signal is high at the rising edge of the `clock` signal when `selectn` is active, the data in the addressed register is driven onto the `dataout` bus. The register output vectors are included to show exactly when data is written to the registers.



Discontinued IP

Figure 19: **Management Interface Timing Diagram**

Statistics Register Access

Access to the Statistics registers is similar to access to the Configuration registers, however, the Statistics counters are reset after they are read, the output data is only valid for a single clock cycle.

Configuration Vector and Configuration Status Vector

If the Management Interface is not selected during core generation, access to the configuration registers is through a Configuration Vector and a Configuration Status Vector. Figure 20 shows the timing relationship between these vectors and the core. The Configuration Vector settings take effect on the next clock tick. The Configuration Status Vector reflects the current configuration.

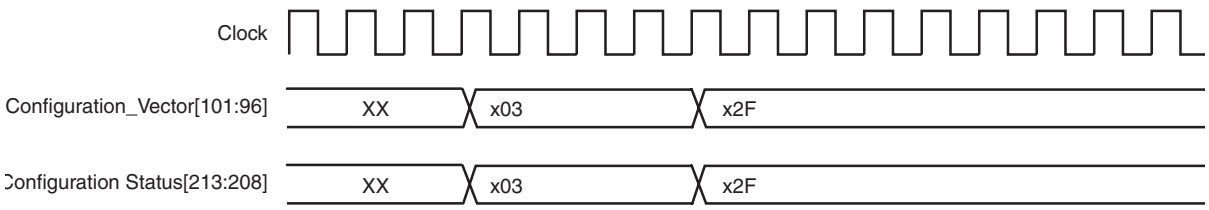


Figure 20: **Use of Configuration Vector and Configuration Status Vector**

Speed Negotiation Vectors

If speed negotiation is selected during core generation, the core is generated with two additional vectors allowing communication between user logic and the speed negotiation block. For more information, see "Speed Negotiation for Multispeed Cores" in Chapter 6 of the *Fibre Channel User Guide*.

Core Verification

The FC core has been verified with extensive simulation, and hardware verification has been completed at both Xilinx and the University of New Hampshire Interoperability Lab (UNH IOL).

Simulation

A highly parameterized test bench was used to test the core at all operating speeds. Tests include:

- Register access
- LOS FSM
- Port State Machine
- Framing
- Statistics gathering

Functional simulation is supported by a structural model generated by the CORE Generator software. See the *Fibre Channel Getting Started Guide* for detailed information.

Hardware Verification

A simple FC B Port design was created around the FC core netlist (with Statistics Gathering and Management Interface). This follows the architecture illustrated in Figure 3, mapped to a 2VP50 device on an ML323 RocketIO transceiver Characterization Board.

This design was subjected to protocol and interoperability testing at the UNH IOL where it passed every FC-PH test at both operating speeds. To request a copy of the UNH test reports, please contact your local Xilinx representative.

Discontinued IP

Software running on the embedded PowerPC 405 processor and the Statistics Gathering block were used together to implement the *FC-FS* Section 28 Speed Negotiation algorithm. This was proven to work correctly when the design was connected to a QLogic 2 Gbps N Port and a QLogic 1 Gbps Switch. Internal testing also showed the core interoperating with Brocade 3200 switches (Firmware v3.0.x and v3.1.x).

Compliance Summary

UNH IOL - FC-PH at 1 Gbps and 2 Gbps.

Device Utilization

Table 9 and 10 provide approximate resource utilization for the block level of the core in varying configurations, plus the DCM and BUFG resource requirements for the example design. Resource utilization is almost independent of the operating speed.

Table 9: **Device Utilization (approximate) Virtex-4 FPGAs**

Parameter Values					Device Resources				
Speed (Gbps)	Management Interface	BBCredit Management	Statistics	Speed Negotiation	Slices	LUTs	FFs	BUFGs	DCMs
1	No	Yes	No	No	1645	2127	1367	4	1
1	No	No	No	No	1209	1379	1177	4	1
1	Yes	No	No	No	1446	1670	1430	4	1
1	Yes	Yes	Yes	No	2317	3044	1927	4	1
1	Yes	No	Yes	No	1830	2226	1723	4	1
1	Yes	Yes	No	No	1931	2487	1634	4	1
2	No	Yes	No	No	1637	2274	1369	4	1
2	No	No	No	No	1198	1523	1179	4	1
2	Yes	Yes	Yes	No	2721	3181	1929	4	1
2	Yes	No	No	No	2298	1808	1432	4	1
2	Yes	Yes	No	No	1918	2624	1636	4	1
2	Yes	No	Yes	No	1813	2364	1725	4	1
4	No	Yes	No	No	1640	2121	1369	4	1
4	No	No	No	No	1201	1370	1179	4	1
4	Yes	No	Yes	No	1822	2222	1725	4	1
4	Yes	Yes	Yes	No	2310	3041	1929	4	1
4	Yes	No	No	No	1437	1665	1432	4	1
4	Yes	Yes	No	No	1927	2484	1636	4	1
1-2	No	No	No	No	1379	1573	1373	6	1
1-2	No	Yes	No	No	1810	2321	1563	6	1
1-2	No	No	No	Yes	1540	1848	1474	6	1

Discontinued IP

Parameter Values					Device Resources				
Speed (Gbps)	Management Interface	BBCredit Management	Statistics	Speed Negotiation	Slices	LUTs	FFs	BUFGs	DCMs
1-2	No	Yes	No	Yes	1979	2596	1664	6	1
1-2	Yes	No	Yes	No	2008	2424	1923	6	1
1-2	Yes	Yes	No	Yes	2257	2968	1935	6	1
1-2	Yes	No	Yes	Yes	2187	2703	2024	6	1
1-2	Yes	No	No	No	1621	1868	1630	6	1
1-2	Yes	Yes	Yes	No	2496	3248	2127	6	1
1-2	Yes	Yes	Yes	Yes	2674	3525	2228	6	1
1-2	Yes	Yes	No	No	2110	2692	1834	6	1
1-2	Yes	No	No	Yes	177[illegible]	2147	1731	6	1
2-4	No	Yes	No	No	1[illegible]98	2304	1555	6	1
2-4	No	No	No	No	1369	1556	1365	6	1
2-4	No	No	No	Yes	1[illegible]31	1831	1466	6	1
2-4	No	Yes	No	Yes	1966	2579	1656	6	1
2-4	Yes	Yes	Yes	[illegible]	2480	3230	2119	6	1
2-4	Yes	No	Yes	[illegible]	1993	2405	1915	6	1
2-4	Yes	No	Yes	Yes	2176	2688	2016	6	1
2-4	Yes	No	No	No	1607	1848	1622	6	1
2-4	Yes	Yes	Yes	Yes	2670	3511	2220	6	1
2-4	Yes	No	No	Yes	1765	2130	1723	6	1
2-4	Yes	Yes	No	No	2097	2673	1826	6	1
2-4	Yes	Yes	No	Yes	2249	2953	1927	6	1

Discontinued IP

Table 10: **Device Utilization (approximate) Virtex-5 FPGAs**

Parameter Values					Device Resources				
Speed (Gbps)	Management Interface	BBCredit Management	Statistics	Speed Negotiation	Slices	LUTs	FFs	BUFGs	DCMs
1	No	Yes	No	No	1026	1729	1271	5	1
1	No	No	No	No	798	1120	1068	5	1
1	Yes	No	No	No	1025	1342	1332	5	1
1	Yes	No	Yes	No	983	1833	1617	5	1
1	Yes	Yes	Yes	No	1389	2491	1821	5	1
1	Yes	Yes	No	No	1165	2000	1536	5	1
2	No	No	No	No	804	1124	1072	5	1
2	No	Yes	No	No	987	1728	1273	5	1
2	Yes	Yes	No	No	1143	2000	1538	5	1
2	Yes	No	No	No	982	1341	1334	5	1
2	Yes	Yes	Yes	No	1433	2495	1822	5	1
2	Yes	No	Yes	No	1008	1832	1618	5	1
1-2	No	No	No	Yes	961	1380	1237	9	1
1-2	No	Yes	No	No	1076	1816	1337	9	1
1-2	No	Yes	No	Yes	1171	1996	1438	9	1
1-2	No	No	No	No	890	1205	1136	9	1
1-2	Yes	No	No	No	1053	1426	1402	9	1
1-2	Yes	No	Yes	No	1115	1917	1686	9	1
1-2	Yes	Yes	Yes	No	1325	2542	1890	9	1
1-2	Yes	No	No	Yes	1018	1606	1503	9	1
1-2	Yes	No	Yes	Yes	1257	2097	1787	9	1
1-2	Yes	Yes	No	No	1257	2047	1606	9	1
1-2	Yes	Yes	Yes	Yes	1464	2722	1991	9	1
1-2	Yes	Yes	No	Yes	1320	2227	1707	9	1

Discontinued IP

References

- Virtex-4 FPGA User Guide (UG070)
- Virtex-5 FPGA User Guide (UG190)
- Virtex-4 FPGA RocketIO Multi-Gigabit Transceiver User Guide (UG076)
- Virtex-5 FPGA RocketIO GTP Transceiver User Guide (UG196)
- ANSI INCITS X3-230-1994 (R1999)
- ANSI INCITS X3-297-1997 (R2002)
- ANSI INCITS X3-303-1998 FC-PH
- T11-FC-FS v1.9

Support

Xilinx provides technical support for this product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of this product if implemented in devices not listed in the documentation, or if customized beyond that allowed in the product documentation, or if any changes are made in sections of design marked *DO NOT MODIFY*.

Related Information

Xilinx products are not intended for use in life support appliances, devices, or systems. Use of a Xilinx product in such applications without the written consent of the appropriate Xilinx officer is prohibited.

Ordering Information

This Xilinx LogiCORE IP module is provided under the terms of the Xilinx Core Site License. The core is generated using the Xilinx ISE CORE Generator software, which is a standard component of the Xilinx ISE Design Suite.

For full access to the functionality of this core in simulation and in hardware, you must purchase and install a license for the core. Contact your local Xilinx sales representative for pricing and availability of this and other Xilinx LogiCORE IP modules. Information about additional LogiCORE IP modules can be found on the Xilinx.com Intellectual Property page.

Discontinued IP

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
5/17/04	1.0	First draft in new format.
6/29/04	1.1	Final version for release.
8/25/04	1.11	Updated resource utilization numbers in Table 8, Device Utilization
4/28/05	2.0	Updated core to version 2.0, Xilinx tools v7.1i SP2.
1/11/06	2.1	Update core to version 2.1, Xilinx tools v8.1i.
2/15/07	3.1	Update core to version 3.1, Xilinx tools to v9.1i.
8/8/07	3.2	Update core to version 3.2, Xilinx tools to v9.2i.
3/24/08	3.3	Update core to version 3.3, Xilinx tools to v10.1.
4/24/09	3.4	Update core to version 3.4, Xilinx tools to v11.1.
4/19/10	3.5	Update core to version 3.5, Xilinx tools to v12.1.

Notice of Disclaimer

Xilinx is providing this product documentation, hereinafter "Information," to you "AS IS" with no warranty of any kind, express or implied. Xilinx makes no representation that the Information, or any particular implementation thereof, is free from any claims of infringement. You are responsible for obtaining any rights you may require for any implementation based on the Information. All specifications are subject to change without notice. XILINX EXPRESSLY DISCLAIMS ANY WARRANTY WHATSOEVER WITH RESPECT TO THE ADEQUACY OF THE INFORMATION OR ANY IMPLEMENTATION BASED THEREON, INCLUDING BUT NOT LIMITED TO ANY WARRANTIES OR REPRESENTATIONS THAT THIS IMPLEMENTATION IS FREE FROM CLAIMS OF INFRINGEMENT AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Except as stated herein, none of the Information may be copied, reproduced, distributed, republished, downloaded, displayed, posted, or transmitted in any form or by any means including, but not limited to, electronic, mechanical, photocopying, recording, or otherwise, without the prior written consent of Xilinx.

Discontinued IP