XAPP606 (v1.1) December 20, 2001

XGMII Using the DDR Registers, DCM, and SelectI/O Features in Virtex-II Devices

Author: Martin Rhodes

Summary

The DDR, DCM, and SelectI/O™ features of the Virtex™-II architecture make it ideal for use in applications such as the IEEE Draft P802.3ae/D3.1, 10-Gigabit Media Independent Interface (XGMII). The Digital Clock Manager (DCM) provides the Delay Locked Loop (DLL) and Digital Phase Shift (DFS) functions. The Input/Output Blocks (IOBs) provide both input and output Double-Data Rate (DDR) registers. The SelectI/O feature provides the High-Speed Transceiver Logic Class I (HSTL_I) bus standard required for XGMII. This application note describes an interface design to XGMII. This reference design is fully synthesizable, has a flexible pinout and achieves the 156.25 MHz DDR (312.5 MHz switching) performance with automatic place and route tools.

Introduction to XGMII

The purpose of the XGMII is to provide a simple, inexpensive, and easy-to-implement optional interconnection between the Media Access Control (MAC) sublayer and the Physical layer (PHY) of 10-Gigabit ethernet. The interface provides two separate 32-bit data paths (TXD<31:0>, RXD<31:0>) each with 4-bit data delimiters (TXC<3:0>, RXC<3:0>) which are synchronous to their respective clock (TX_CLK, RX_CLK) which operate at 156.25 MHz ± 0.01%. XGMII supports full duplex operation only as illustrated in Figure 1. All signals use the HSTL_I bus standard; this is a general purpose high-speed 1.5 V standard, requiring a differential amplifier at the input and a push-pull driver on the output.

The XGMII tracks are only designed to be a few centimeters in length (approximately 7 cm) since routing imperfections will soon degrade the high frequency signals. The interface is intended to link separate ICs, placed in close proximity on the same PCB.

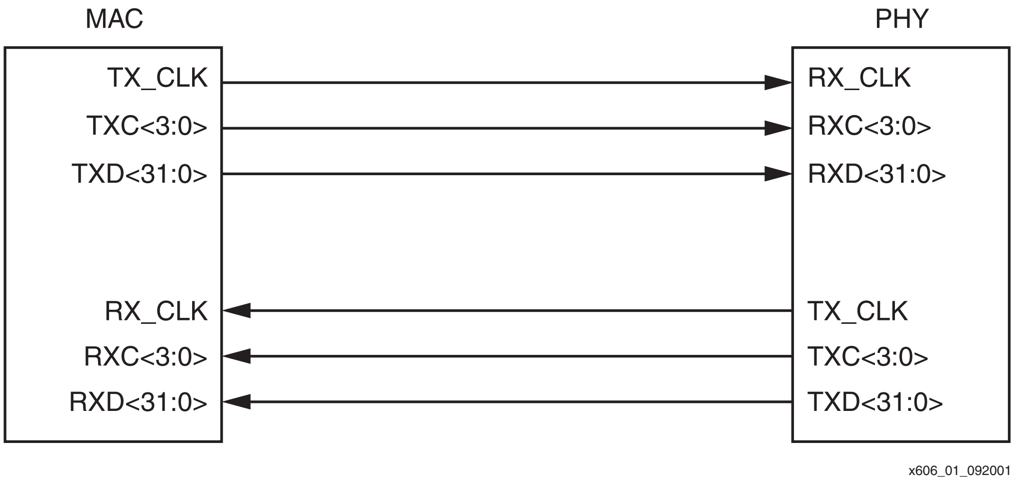
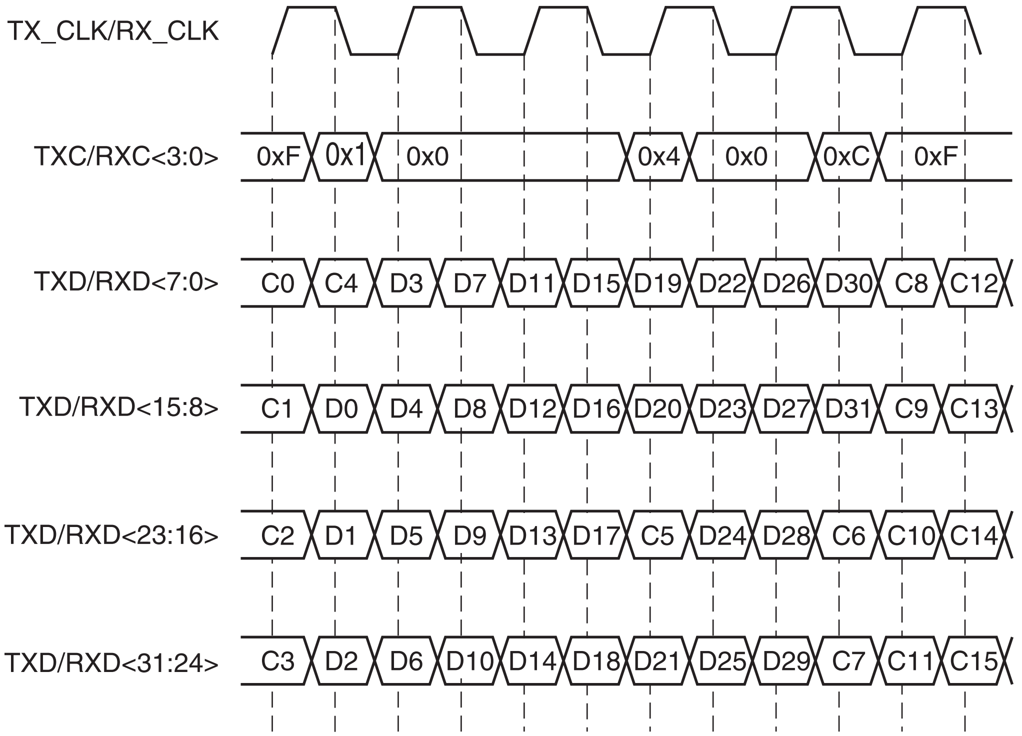


Figure 1: **Full Duplex Operation of XGMII**

XGMII Signal Definition

TXD<31:0> and RXD<31:0> are each grouped into four byte lanes. TXC<3:0> and RXC<3:0> are the data delimiters for these four byte lanes and separate frame data bytes from control characters. Note that TXC<0> / RXC<0> maps to TXD<7:0> / RXD<7:0>, whereas TXC<3> / RXC<3> maps to TXD<31:24> / RXD<31:24>.

All signals are synchronous to either the clocks TX_CLK or RX_CLK, and data transfers take place on both clock edges to support the double data rate. This is illustrated in Figure 2.


TX_CLK/RX_CLK
TXC/RXC<3:0>
0xF 0x1 0x0 0x4 0x0 0xC 0xF
TXD/RXD<7:0>
C0 C4 D3 D7 D11 D15 D19 D22 D26 D30 C8 C12
TXD/RXD<15:8>
C1 D0 D4 D8 D12 D16 D20 D23 D27 D31 C9 C13
TXD/RXD<23:16>
C2 D1 D5 D9 D13 D17 C5 D24 D28 C6 C10 C14
TXD/RXD<31:24>
C3 D2 D6 D10 D14 D18 D21 D25 D29 C7 C11 C15

Where C = Control Character, D = Frame Data Byte

x606_02_092001

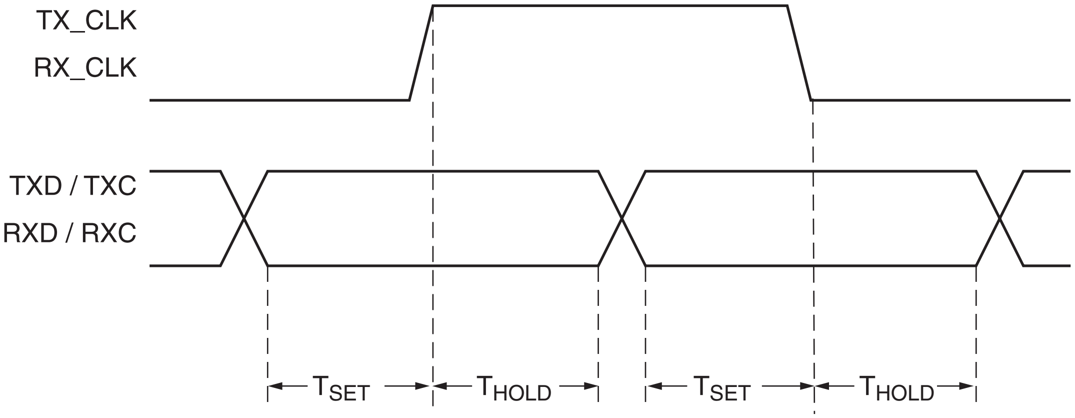
Figure 2: **Timing Relationship of XGMII Signals**

XGMII Timing Parameters

Together, Table 1 and Figure 3 illustrate the XGMII timing parameters. For the transmitter, TXD<31:0> and TXC<3:0> must be driven by the device to provide these limits. A receiver must be able to accept RXD<31:0> and RXC<3:0> at the input to the device within these limits.

Table 1: **TX_CLK and RX_CLK Timing Parameters**

Symbol	Transmitter	Receiver	Units
T_{SET}	960	480	ps
T_{HOLD}	960	480	ps



x606_03_092201

Figure 3: **TX_CLK and RX_CLK Timing Parameters**

Using the XGMII Reference Design

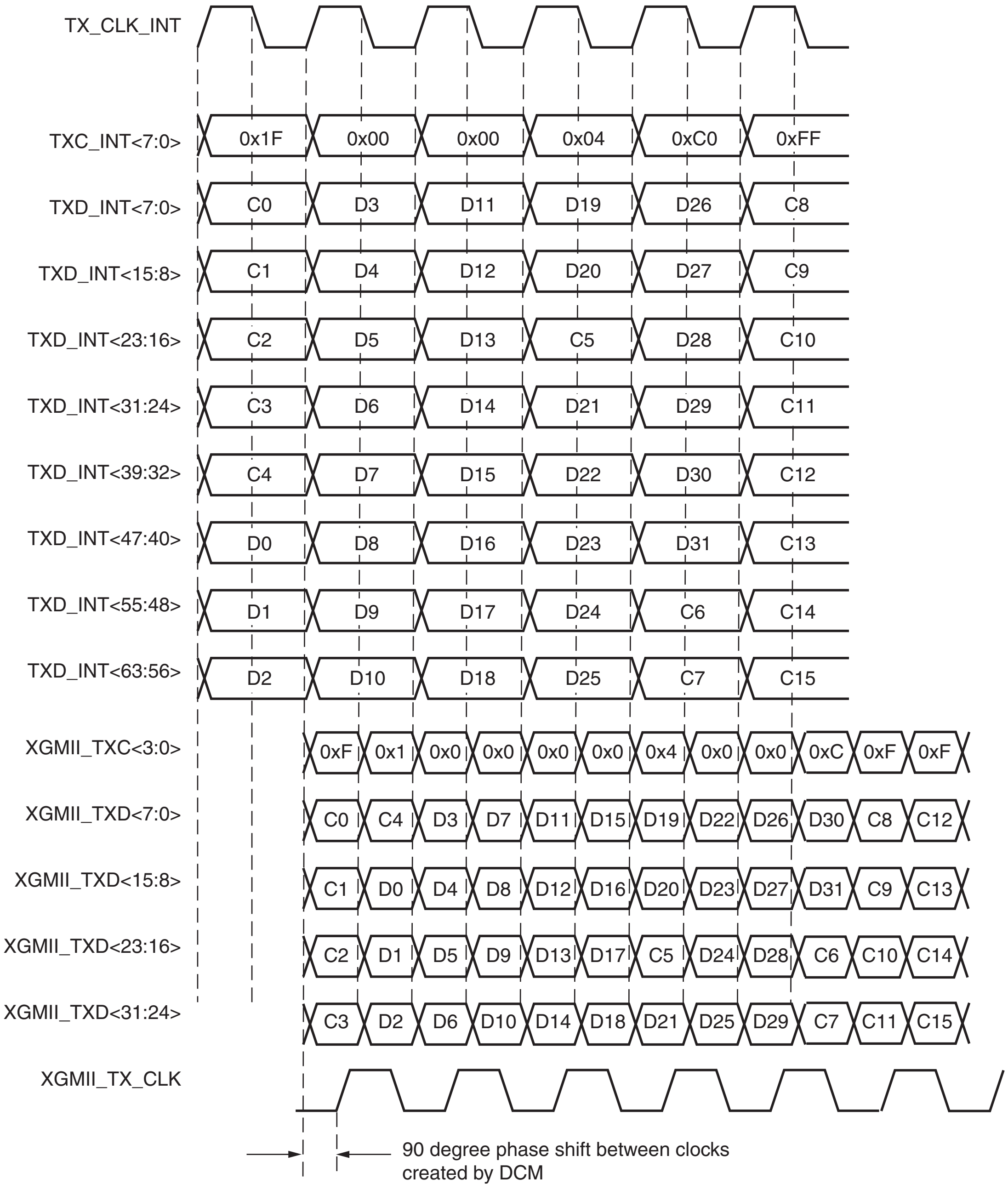
The XGMII reference design is available for VHDL users on the Xilinx FTP site at ftp://ftp.xilinx.com/pub/applications/xapp/xapp606.zip. The reference design implements the XGMII interface described above. These signals are connected to external pads of the Virtex-II device using SelectI/O features to drive and receive at the HSTL_I bus standard. All other inputs and outputs of the reference design are designed to interface to internal logic: this may be to the reconciliation sublayer of a 10-Gigabit ethernet MAC, to the 10-Gigabit ethernet PHY sublayers, or to some other bridging logic. This reference design is therefore kept as simple and general purpose as possible. It simply performs the DDR (where the XGMII bus is effectively running at 312.5 MHz) to Single Data Rate (SDR) conversion (where the bus is running at 156.25 MHz). This is achieved by doubling the size of the internal busses to maintain the XGMII 10-Gigabit rate. DCM is used on both the TX_CLK and RX_CLK domains to meet the XGMII timing parameters. Table 2 lists the connections to the reference design.

Table 2: **XGMII Reference Design Port Definition**

Port Name	Connection	Description
XGMII_TX_CLK	External Output	XGMII TX_CLK (XGMII transmitter clock)
XGMII_RX_CLK	External Input	XGMII RX_CLK (XGMII receiver clock)
XGMII_TXD<31:0>	External Output	XGMII TXD<31:0> (XGMII transmitter data)
XGMII_TXC<3:0>	External Output	XGMII TXC<3:0> (XGMII transmitter data delimiter)
XGMII_RXD<31:0>	External Input	XGMII RXD<31:0> (XGMII receiver data)
XGMII_RXC<3:0>	External Input	XGMII RXC<3:0> (XGMII receiver data delimiter)
RESET	Internal Input	Asynchronous reset for flip-flops
TX_CLK_REF	Internal Input	Transmitter clock reference from which XGMII_TX_CLK and TX_CLK_INT are derived using DCM. This must be of a frequency of 156.25 MHz ± 0.01%
TX_CLK_INT	Internal Output	Internal transmitter clock used to clock all transmitter logic
RX_CLK_INT	Internal Output	Internal receiver clock used to clock all receiver logic
TXD_INT<63:0>	Internal Input	Transmitter data, single data rate, to be output across XGMII_TXD<31:0>
TXC_INT<7:0>	Internal Input	Transmitter data delimiters, single data rate, to be output across XGMII_TXC<3:0>
RXD_INT<63:0>	Internal Output	Receiver data, single data rate, recovered from XGMII_RXD<31:0>
RXC_INT<7:0>	Internal Output	Receiver data delimiters, single data rate, recovered from XGMII_RXC<3:0>
TX_DCM_LOCK	Internal Output	The locked signal from the DCM used to derive the transmitter clocks
TX_DCM_RESET	Internal Input	To manually reset the DCM used to derive the transmitter clocks
RX_DCM_LOCK	Internal Output	The locked signal from the DCM used to derive the receiver clock
RX_DCM_RESET	Internal Input	To manually reset the DCM used to derive the receiver clock

Using the XGMII Transmitter

Figure 4 shows the relationship between the internal transmitter signals which must be driven by the user, and the XGMII. The pipeline delay shown is accurate, as is the byte mapping between single and double data rate buses. Therefore the order in which bits are transmitted for words of txd_int<63:0> is txd_int<31:0> firstly, and txd<63:32> finally. Also note that txc_int<0> associates with txd_int<7:0>, whereas txc_int<7> associates with txd_int<63:56>. The pipeline delay incurred through the transmitter is necessary for timing and will be explained in the implementation section.



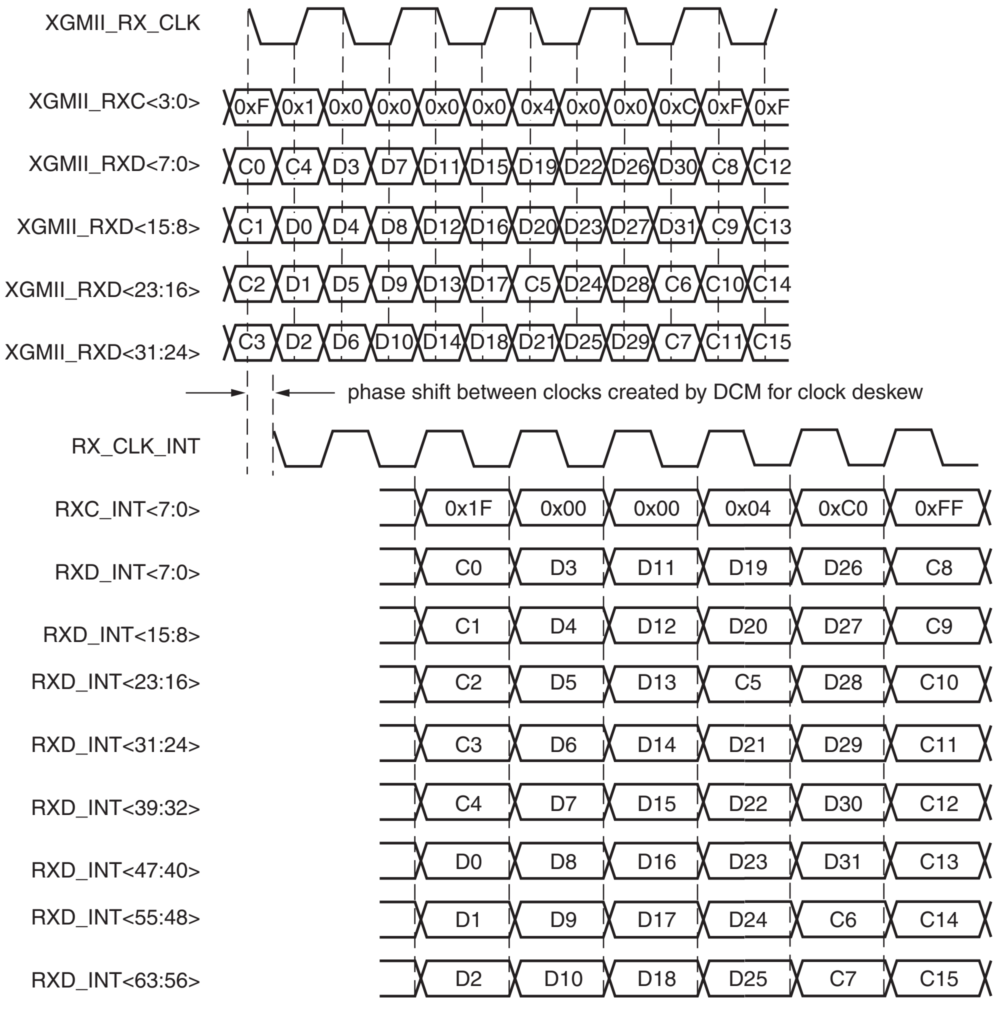
Where C = Control Character, D = Frame Data Byte

x606_04_092201

Figure 4: **Using the XGMII Transmitter**

Using the XGMII Receiver

Figure 5 shows the relationship between the XGMII and the internal receiver signals which are provided to the user. The pipeline delay shown is accurate, as is the byte mapping between single and double data rate buses. Therefore the order in which bits are received for words of rxd_int<63:0> is rxd_int<31:0> firstly, and rxd<63:32> finally. It is important to note that rxc_int<0> associates with rxd_int<7:0>, whereas rxc_int<7> associates with rxd_int<63:56>. The pipeline delay incurred through the receiver is necessary for timing and will be explained in the implementation section.


XGMII_RX_CLK
XGMII_RXC<3:0> 0xF 0x1 0x0 0x0 0x0 0x0 0x4 0x0 0x0 0xC 0xF 0xF
XGMII_RXD<7:0> C0 C4 D3 D7 D11 D15 D19 D22 D26 D30 C8 C12
XGMII_RXD<15:8> C1 D0 D4 D8 D12 D16 D20 D23 D27 D31 C9 C13
XGMII_RXD<23:16> C2 D1 D5 D9 D13 D17 C5 D24 D28 C6 C10 C14
XGMII_RXD<31:24> C3 D2 D6 D10 D14 D18 D21 D25 D29 C7 C11 C15
phase shift between clocks created by DCM for clock deskew
RX_CLK_INT
RXC_INT<7:0> 0x1F 0x00 0x00 0x04 0xC0 0xFF
RXD_INT<7:0> C0 D3 D11 D19 D26 C8
RXD_INT<15:8> C1 D4 D12 D20 D27 C9
RXD_INT<23:16> C2 D5 D13 C5 D28 C10
RXD_INT<31:24> C3 D6 D14 D21 D29 C11
RXD_INT<39:32> C4 D7 D15 D22 D30 C12
RXD_INT<47:40> D0 D8 D16 D23 D31 C13
RXD_INT<55:48> D1 D9 D17 D24 C6 C14
RXD_INT<63:56> D2 D10 D18 D25 C7 C15

Where C = Control Character, D = Frame Data Byte

x606_05_092201

Figure 5: **Using the XGMII Receiver**

XGMII Transmitter Implementation

Figure 6 illustrates the fundamental principles of the transmitter design. Illustrated is clock production for TX_CLK_INT and XGMII_TX_CLK. Also shown, as an example, is the production of XGMII_TXD<0>. Identical logic is used to produce XGMII_TXD<31:1> and XGMII_TXC<3:0>. Refer to Figure 6 for all descriptions throughout this section.

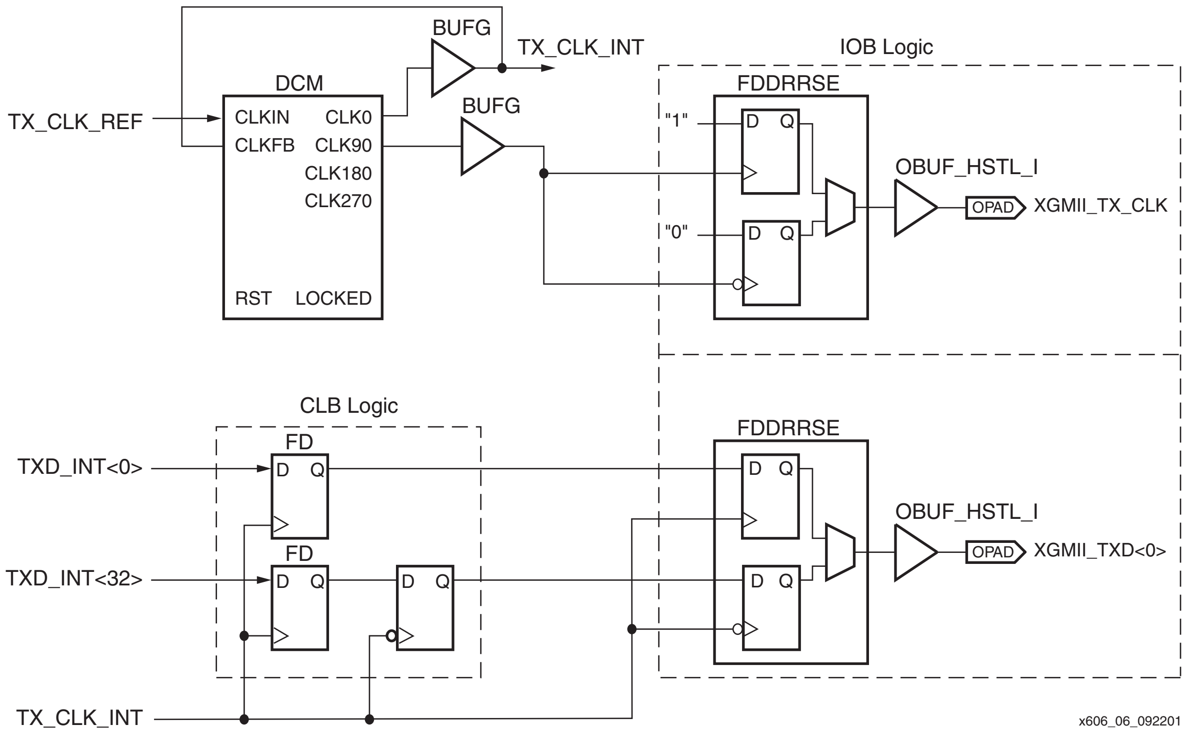


Figure 6: **XGMII Transmitter Implementation**

XGMII Transmitter Clock Production

Virtex-II DCM provides a convenient solution to generate the phase differing clocks required. TX_CLK_REF must be provided to the design. This must be of frequency 156.25 MHz ± 0.01% to satisfy the XGMII specification. This clock is fed into a DCM and from this reference two 50/50 duty-cycle corrected clocks are created with a relative phase difference of 90 degrees. These two clocks are each fed through a BUFG primitive which drive their clock onto one of the sixteen global clock networks. These provide low-skew clock distribution to, potentially, all parts of the device. Therefore, at any point in the device, the phase relationship between these two clocks is effectively maintained. Please refer to the Virtex-II User Guide for a description of Global Clock Networks.

Of these two clocks described, one is given the name TX_CLK_INT as shown in Figure 6, and should be used as the internal transmitter clock for all related logic.

The second clock is routed only to one of the IOBs where it is used to derive XGMII_TX_CLK. Instead of simply driving this clock onto a pad using the OBUF_HSTL_I buffer to create XGMII_TX_CLK, this clock instead is used to clock the output DDR registers (FDDRRSE) of that IOB. By placing logic 1 and logic 0 at the two inputs to this DDR register, the resultant output of this is to toggle between the logic 1 and logic 0 at DDR (on every clock edge). This produces a clock which will have exactly the same delay incurred by the output data bits XGMII_TXD<31:0> and XGMII_TXC<3:0>, further maintaining the phase relationship between the two clocks derived from the DCM. This is illustrated in Figure 6, which shows that both XGMII_TX_CLK and XGMII_TXD<0> have equivalent IOB logic.

XGMII_TX_CLK is completed by using the SelectI/O resource to drive the clock through an HSTL_I output buffer and onto a pad of the device.

XGMII Data Transmission

As illustrated in Figure 4, both TXD_INT<63:32> and TXD_INT<31:0> are transmitted on XGMII_TXD<31:0> using alternate clock edges of TX_CLK_INT. Similarly TXC_INT<7:4> and TXC_INT<3:0> are transmitted on XGMII_TXC<3:0>. Consequently XGMII_TXD<0> is constructed from TXD_INT<32> and TXD_INT<0>, whereas XGMII_TXC<0> is constructed from TXC_INT<4> and TXC_INT<0>.

The logic provided by the reference design for XGMII transmission is now described and illustrated in Figure 6 for XGMII_TXD<0> only. Identical logic is used to produce XGMII_TXD<31:1> and XGMII_TXC<3:0>.

TXD_INT<0> and TXD_INT<32> are firstly registered on the rising edge of TX_CLK_INT. TXD_INT<32> is immediately registered again on the falling edge. This creates a half period delay constraint of 3.2 ns from this rising edge flip-flop to the falling edge flip flop for this path, which the PAR tools are easily capable of meeting. This has the advantage of giving a whole clock period (6.4 ns) of routing time from CLB logic to IOB logic since rising-edge flip-flops are routed to rising-edge flip-flops whilst falling edge flip-flops are routed to falling edge flip-flops. This provides maximum flexibility for the PAR tools since the CLB logic can now be placed well away from the chosen IOBs.

At the IOB, the dedicated DDR registers are used to clock both TXD_INT<32> and TXD_INT<0> onto XGMII_TXD<0> on alternate clock edges of TX_CLK_INT. XGMII_TXD<0> is completed by using SelectI/O to drive the clock through an HSTL_I output buffer and onto a pad of the device.

Since the IOB logic used is identical to that used for the production of XGMII_TX_CLK, the relationship between transitions of XGMII_TXD<0> and XGMII_TX_CLK edges is maintained at 90 degrees. This ensures that XGMII_TXD<0> is "source-centred" (providing nominally 1.6 ns of both setup and hold time) with respect to XGMII_TX_CLK. This easily surpasses the timing parameters set in Table 1.

XGMII Receiver Implementation

Figure 7 illustrates the fundamental principles of the receiver design. Illustrated is clock production for RX_CLK_INT. Also shown, as an example, is the production of RXD_INT<32> and RXD_INT<0> from XGMII_RXD<0>. Identical logic is used to produce RXD_INT<63:33>, RXD_INT<31:1> and RXC_INT<7:0>. Refer to Figure 7 for all descriptions throughout this section.

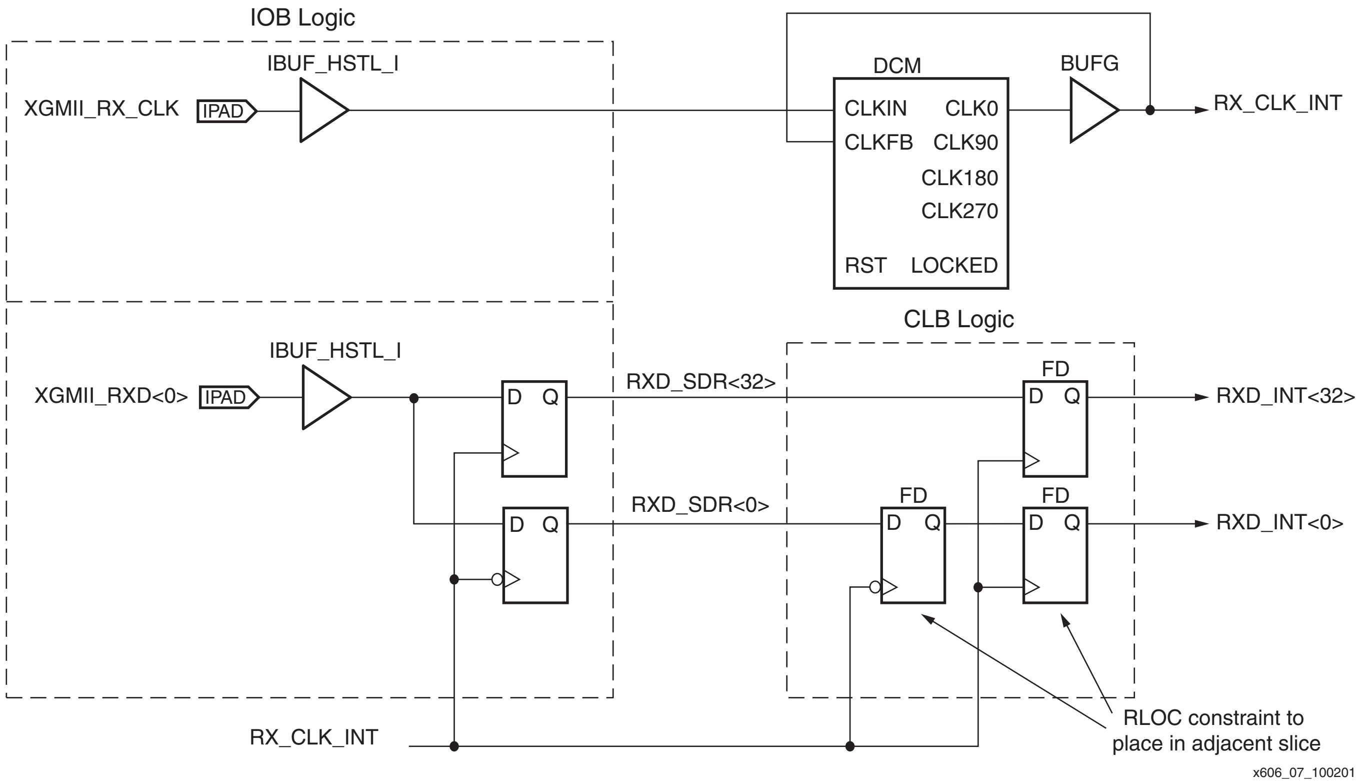


Figure 7: **XGMII Receiver Implementation**

XGMII Receiver Clock Production

XGMII_RX_CLK is provided by the XGMII. This must firstly pass through the SelectI/O input buffer to receive the HSTL_I bus standard. The clock signal is then immediately routed to a DCM.

The purpose of this DCM is to deskew the clock driven onto the global clock network. Without this DCM, the clock would simply be routed through the IBUF_HSTL_I followed by the BUFG which drives the clock onto the global clock matrix. This would result in a large routing delay when compared to the data inputs; these data bits are only routed from the IBUF_HSTL_I buffer to the input of the DDR registers, all within single IOBs (see Figure 7 for XGMII_RXD<0> example). This results in a loss of the delicate timing relationship between the input clock and the input data which was present at the device pads. By using a DCM in this manner, RX_CLK_INT is compared with XGMII_RX_CLK from the IOB. The function of the DCM is to control the phase shift between the rising edges of these clocks, counteracting the large timing delay caused by the global clock network. This restores the timing relationship at the IOB input DDR registers to match those present at the device pads. To accurately achieve this timing relationship, the DCM is set to use the fixed phase shift mode (refer to the User Guide). This skews RX_CLK_INT so that its edges fall in the center of the XGMII input data.

It is important to note that the 50/50 duty-cycle correction functionality for this DCM must not be used. The timing relationship between both clock edges of XGMII_RX_CLK relative to XGMII_RXD<31:0> / XGMII_RXC<3:0> must be maintained.

XGMII Data Reception

As illustrated in Figure 5, both RXD_INT<63:32> and RXD_INT<31:0> are received from XGMII_RXD<31:0> using alternate clock edges of RX_CLK_INT. Similarly RXC_INT<7:4> and RXD_INT<3:0> are received from XGMII_RXC<3:0>. Consequently both RXD_INT<32> and RXD_INT<0> are obtained from XGMII_RXD<0>, whereas RXC_INT<4> and RXC_INT<0> are obtained from XGMII_RXC<0>.

The logic provided by the reference design for XGMII reception is now described and illustrated in Figure 7 for XGMII_RXD<0> only. Identical logic is used to receive XGMII_RXD<31:1> and XGMII_RXC<3:0>.

The XGMII_RXD<0> is input from the XGMII. This must firstly pass through the SelectI/O input buffer to receive the HSTL_I bus standard. The signal is then immediately routed to the DDR input registers which are also present in the IOB. These clock the data on alternative edges as illustrated to create RXD_SDR<0> and RXD_SDR<32>. These are in turn routed to CLB flip-flops. Similar to the transmitter logic, these signals are allowed a whole clock period (6.4 ns) to be routed from IOB to CLB since rising-edge flip-flops are routed to rising-edge flip-flops whilst falling edge flip-flops are routed to falling edge flip-flops. Again this provides maximum flexibility for the PAR tools since the CLB logic can now be placed well away from the chosen IOBs.

Finally the falling edge CLB flip-flop is reclocked on the rising edge so that the design provides registered rising-edge (internal) outputs for all signals. Since the DCM must not use its duty-cycle correction functionality, there is no guarantee that this path, from falling-edge flip-flop to rising-edge flip-flop, has a half clock period (3.2 ns) of routing constraint. The worst case duty-cycle is illustrated in Figure 8 and is shown to give, at worst case, a 960 ps delay constraint for this path. This still satisfies the setup and hold parameters of Table 1. Consequently the design places an RLOC constraint on these rising-edge to falling-edge flip-flops to locate them in adjacent slices. This guides the PAR tools to reliably meet this 960 ps delay constraint.

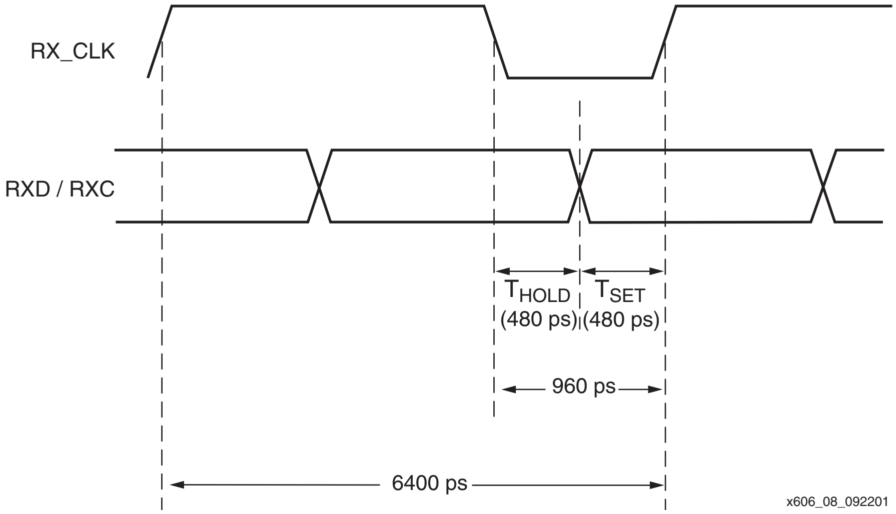


Figure 8: **Worst case Duty-Cycle for XGMII_RX_CLK / RX_CLK_INT**

Pin Location Considerations

The reference design allows for a flexible pinout and the exact pin location of the XGMII is left for the PCB designer. In doing this codes of practice and device restrictions must be followed.

Every Virtex-II device has eight separate I/O banks. Each I/O bank has output drive source voltage pads (V_{CCO}) which must be connected to the same external voltage reference. For XGMII this must be 1.5 V. This forces all I/O pads within the bank to operate at this voltage level.

I/O standards, including HSTL, that use input differential amplifiers require voltage reference inputs (V_{REF}). These are automatically configured by the place and route tools onto predefined pins (see the User Guide for all devices and packages). Approximately one of every twelve I/O pins within an I/O bank will be configured as a V_{REF} pin. For XGMII, which uses HSTL_I, all V_{REF} pins must be connected externally to 0.75 V.

To avoid ground bounce, the number of simultaneously switching outputs per power / ground pair must not exceed the device package limits. These are listed, for all device packages, in the Virtex-II User Guide.

I/Os should be grouped into separate clock domains. XGMII contains two of these; XGMII_RXD<31:0> and XGMII_RXC<3:0> which are synchronous to XGMII_RX_CLK; XGMII_TXD<31:0> and XGMII_TXC<3:0> which are synchronous to XGMII_TX_CLK. It is recommended that these be separated into separate I/O Banks. Unused I/Os in these banks, if tied to ground, will help reduce jitter by providing a low impedance path for ground currents.

Table 3: **Device Utilization / Performance**

IOBs	Slices	GCLKs	DCMs
74	144	3	2

The design will easily exceed the 312.5 MHz DDR in all Virtex-II speed grades. However, to meet the 960 ps falling-edge to rising-edge paths for the worst case receiver clock duty cycle (see Figure 8), a -4 part or faster must be selected.

References

IEEE Draft P802.3ae/D3.1 (specifically Clause 46) http://www.ieee802.org/3/index.html

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/23/01	1.0	Initial Xilinx release.
12/20/01	1.1	Revised speed grade requirement in **Pin Location Considerations**