XAPP622 (v1.2) July 2, 2002

644-MHz SDR LVDS Transmitter/Receiver

Author: Ed McGettigan

Summary

This application note describes single data rate (SDR) transmitter and receiver interfaces operating at up to 644 MHz, using 17 Low-Voltage Differential Signaling (LVDS) pairs (one clock and 16 data channels), implemented in a Virtex™-II FPGA. The accompanying reference design files include an example implementation targeting a Virtex-II XC2V3000-FF1152 -5 speed grade device. The design is implemented as an EDIF netlist with embedded location and routing constraints, Verilog simulation, and synthesis template files.

Introduction

An SDR interface is defined as having only one single positive and negative transition of the clock with respect to the data as shown in Figure 1. Thus, if the data rate is 500 Mb/s the clock frequency would be 500 MHz. SDR LVDS interfaces can be found in a variety of standard products in both the telecom and datacom, markets such as the XSBI 16-bit interface used in 10 gigabit Ethernet systems.

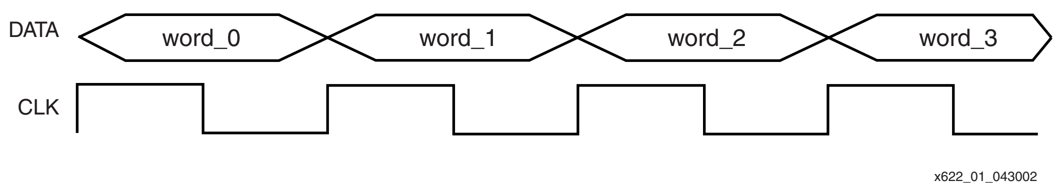

Figure 1: **SDR Clock and Data Interface**

At SDR clock frequencies below the maximum operating frequency (420 MHz) of the Virtex-II Digital Clock Manager (DCM), implementing a single data rate design can be easily accomplished using standard design techniques. This application note describes a method of implementing an SDR interface at clock frequencies higher than the maximum operating frequency of the DCM, without exceeding the AC timing specifications of the Virtex-II devices.

Figure 2 illustrates the overall system configuration, showing a full duplex SDR link between a Virtex-II device and another device with an SDR interface. The Virtex-II device requires a reference clock with either LVDS or LVPECL differential outputs operating at the SDR frequency to generate the transmit clock from the Virtex-II device. Figure 2 shows a discrete clock source operating at the SDR frequency. In some systems, the other device may receive a quad-date rate (QDR) clock frequency and provide an SDR reference clock back to the Virtex-II device.

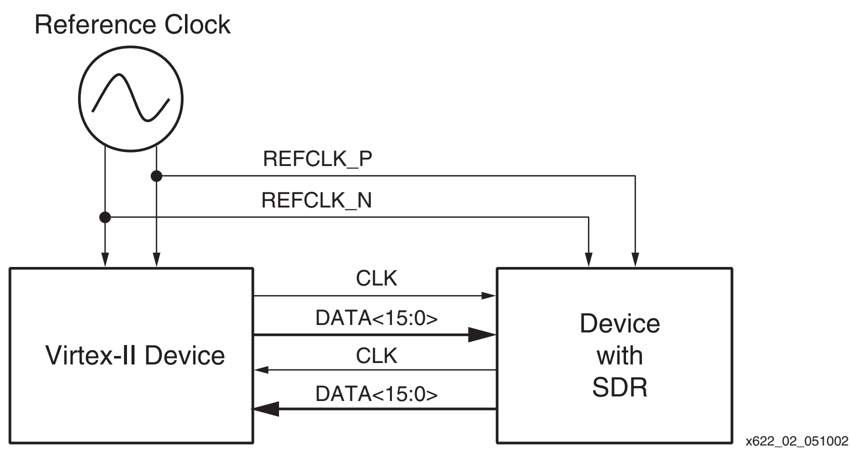

Figure 2: **Typical SDR Link System**

Virtex-II Implementation

Figure 3 shows a Virtex-II SDR interface comprised of three modules, TX_CLOCK, TX_SDR_16D_4TO1, and RX_SDR_16D_4TO1 and one hard macro, DIFF_LVDS_IBUFG. Details on each module and the hard macro are available in the following sections.

Multiple transmitters and receivers can be implemented in the same Virtex-II FPGA. When multiple instances are needed, only the RX_SDR_16D_4TO1 and TX_SDR_16D_4TO1 modules are replicated, saving valuable global clock resources by not replicating the TX_CLOCK module.

The TX_CLOCK module provides the transmitter with two active High SDR global clocks, representing the positive and negative edge transitions of the reference clock (REFCLK) and a quad-rate global clock (QDR) for the system data path logic.

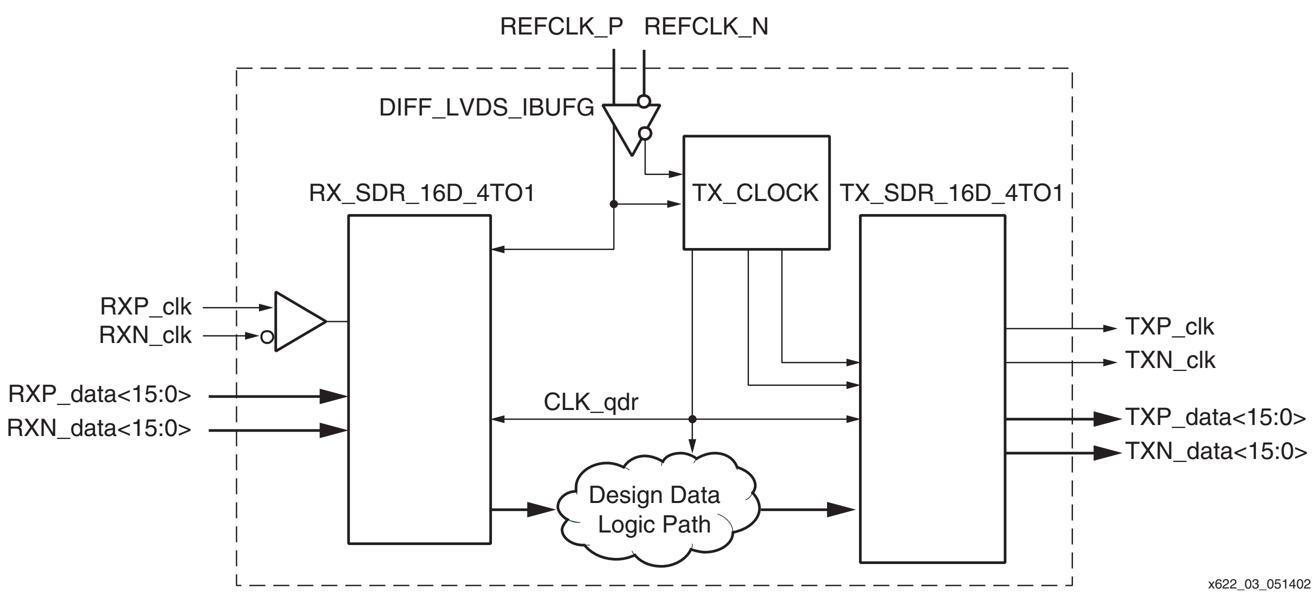

Figure 3: **Virtex-II SDR Interface**

Transmitter Clocks

The TX_CLOCK module has been designed to provide minimal distortion to the duty cycle of the input reference clock to be compatible with industry standards requiring a maximum of ± 5% duty-cycle difference. This is achieved by using a differential input clock buffer, with differential outputs implemented as a Xilinx hard macro named LVDS_DIFF_IBUFG (shown in Figure 4). This macro creates an active High edge during the High transition of the reference clock, as well as an active High edge during the Low transition. The outputs from this cell are connected to two global buffers and a DCM (Figure 5). Implementing the clocking network in this manner eliminates clock skew differences that are due to rising versus falling edges as they propagate through the device. Table 1 shows TX_CLOCK module pin definitions.

Table 1: **TX_CLOCK Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	REFCLK_P REFCLK_N	Differential SDR clock input from LVDS_DIFF_IBUFG hard Macro
	RST	Active High Reset signal
Output	CLK_sdr_p CLK_sdr_n	Active High positive edge global SDR clock Active High negative edge global SDR clock
	CLK_qdr	Active High global QDR clock

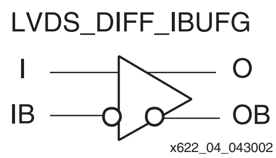

Figure 4: **LVDS_DIFF_IBUFG Hard Macro**

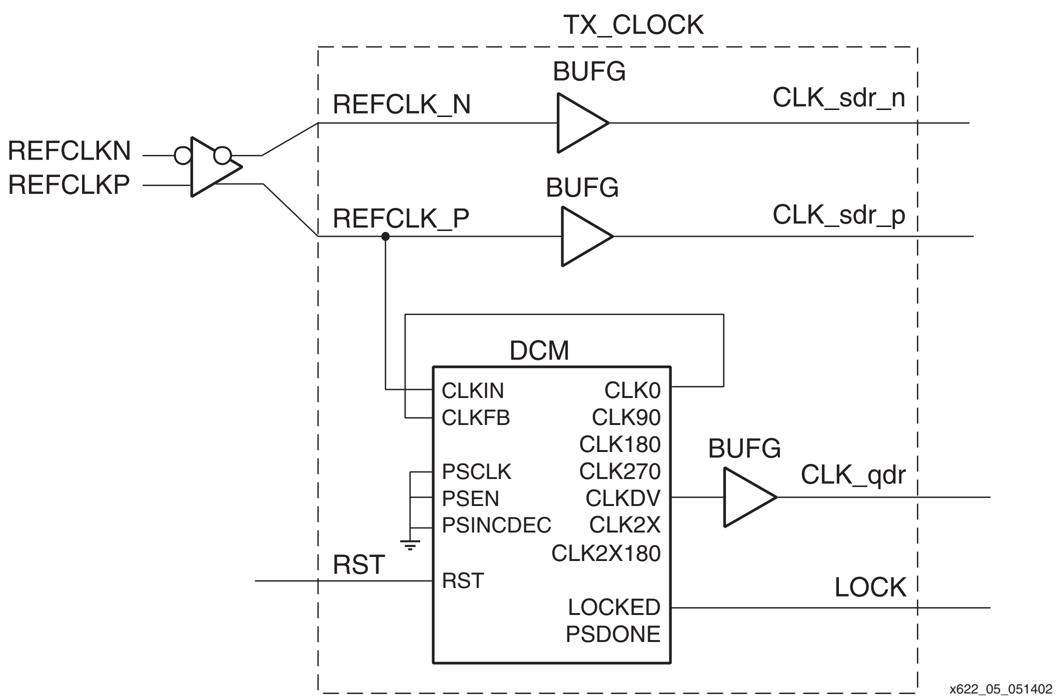

Figure 5: **TX_CLOCK Module**

In addition to the two active High SDR global clocks, a DCM is used to create a QDR global clock. A feature of the Virtex-II DCM clock input pin (CLKIN) allows dividing the input by two before the signal is applied to the digital delay lines in the DCM without additional jitter or delay. To enable this feature, the attribute CLKIN_DIVIDE_BY_2 should be set to TRUE on the DCM cell. This feature enables the DCM to operate at half of the SDR frequency and below the maximum AC timing specification. Figure 6 shows a composite waveform of all clock signals.

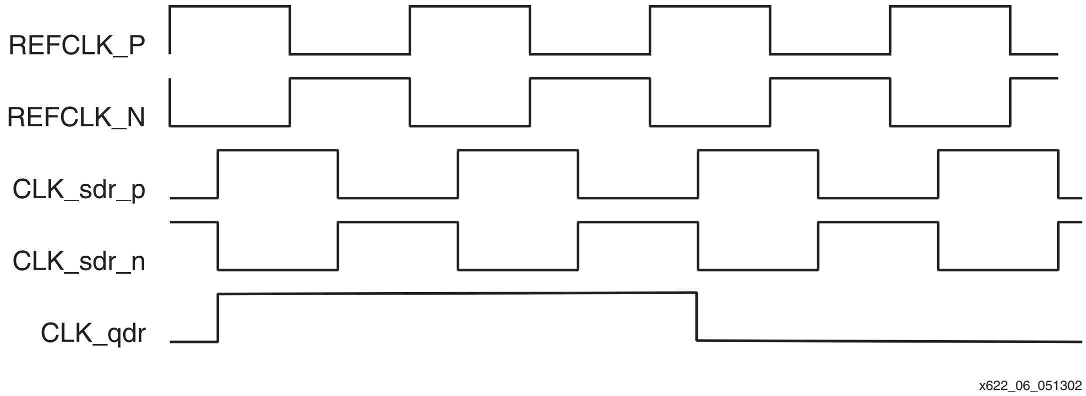

Figure 6: **TX_CLOCK Output Waveforms**

Transmitter with 4-to-1 Serializer (TX_SDR_16D_4TO1)

The transmitter (TX_SDR_16D_4TO1) is composed of two different types of output modules, OUTSTAGE_DATA for the data channels and OUTSTAGE_CLK for the clock output. Both of these are discussed in detail in the following sections. Table 2 contains the module pin descriptions.

Table 2: **TX_SDR_16D_4TO1 Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	DATA<63:0>	Data presented at the 64-bit data input bus is transmitted to the receiver in LSW to MSW order. E.g., <15:0> <31:16> <47:32> <63:48>
	CLK_sdr_p CLK_sdr_n	Active High positive edge global SDR clock Active High negative edge global SDR clock
	CLK_qdr	Active High global QDR clock
	RST	Active High reset
Output	TXP_data<15:0> TXN_data<15:0>	These signals comprise the differential pairs which form the 16-bit transmit bus. All data transferred over this bus is synchronous with the transmit clock.
	TXP_clk TXN_clk	TXP_clk and TXN_clk form the differential, source-synchronous clock signal.

Figure 7 shows the transmitter block diagram. The only module receiving the active High negative edge SDR global clock is the OUTSTAGE_CLK module. The final register stage of the data channels and the clock are located in the IOBs and clocked with the CLK_sdr_p global clock, providing minimal channel-to-channel skew for optimal performance. The CLK_sdr_n global clock is used only to generate the falling edge of the transmitted clock, providing minimal distortion of the original reference clock duty cycle.

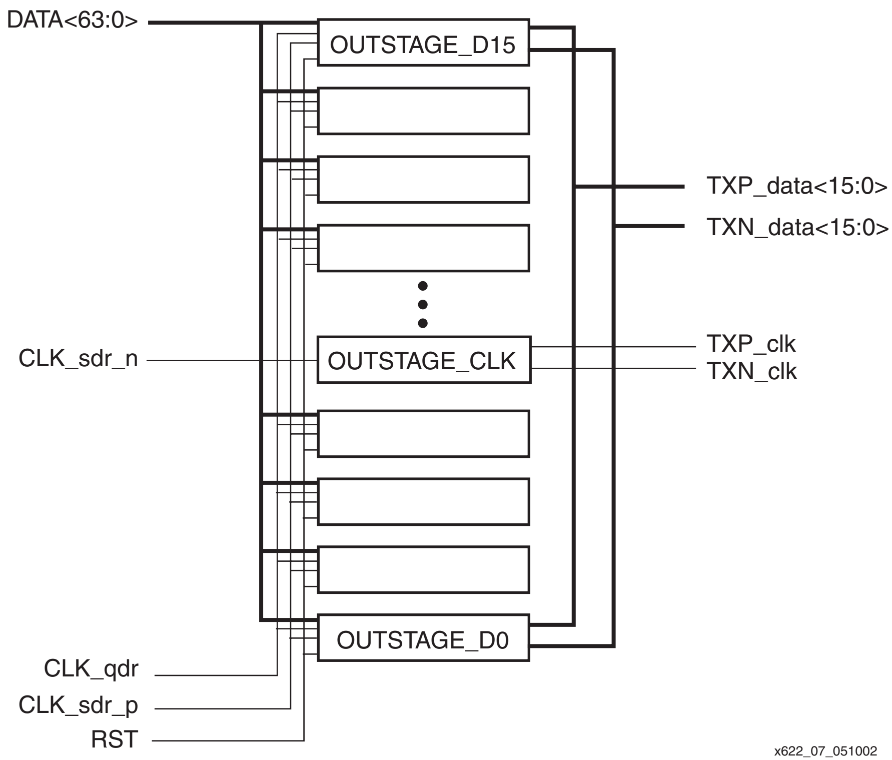

Figure 7: **TX_SDR_16D_4TO1 Block Diagram**

Transmitter Data Output Channel (OUTSTAGE_DATA)

The transmitter output data channel (OUTSTAGE_DATA) module (Figure 8) performs as a 4 to 1 serializer. The four bits of data from the QDR global clock domain are pipelined once for fast timing closure and then fed to a 4-bit parallel-to-serial converter before transmission. Table 3 shows the module pin definitions.

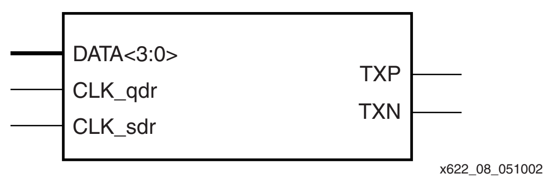

Figure 8: **OUTSTAGE_DATA Module**

Table 3: **OUTSTAGE_DATA Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	DATA<3:0>	Slice of data for this transmitter
	CLK_qdr	Active High QDR global clock
	CLK_sdr	Active High positive edge global SDR clock.
Output	TXP, TXN	LVDS output data (single channel)

The loading of the data from the QDR to the SDR global clock domains is handled with a load-pulse circuit in each OUTSTAGE_DATA module. The construction of the load circuit allows for ample settling time of the DATA between the two clock domains. Figure 9 shows a timing waveform of the transmitted data with respect to the QDR and SDR clock domains.

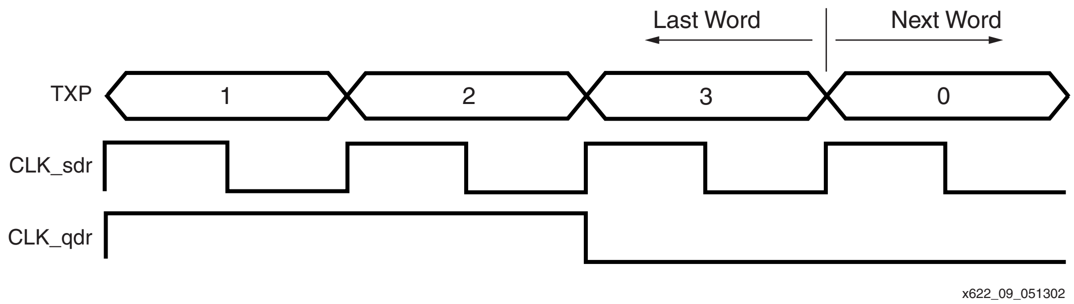

Figure 9: **OUTSTAGE_DATA Output Waveforms**

The timing of this module is very critical and must be explicitly placed in the design. The module includes directed routing constraints to ensure that timing is met every time the design is implemented. The only placement location constraint necessary is the RLOC_ORIGIN for each data channel.

The design files associated with this application note are targeted for the right side of the device in banks 2 and 3. Each OUTSTAGE_DATA module consumes two full CLBs adjacent to the IOBs, with the RLOC_ORIGIN set to three less than the number of CLB slice columns in the device and on an even row. Figure 10 is an example of two data channels in an XC2V3000-FF1152 device. In this example, the following placement constraints would be added to the UCF file for the design.

```
INST    "U_tx/OUTSTAGE_D1"  RLOC_ORIGIN = X108Y66 ;
NET     "TXP_data<1>"       LOC = U8;
NET     "TXN_data<1>"       LOC = U9;
INST    "U_tx/OUTSTAGE_D2"  RLOC_ORIGIN = X108Y68;
NET     "TXP_data<2>"       LOC = U6;
NET     "TXN_data<2>"       LOC = T6;
```

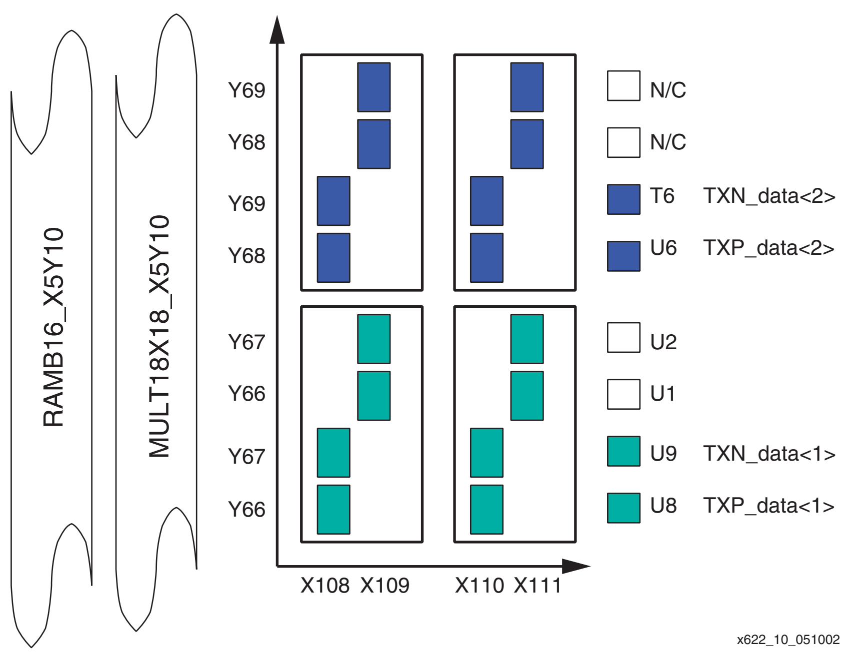

Figure 10: **OUTSTAGE_DATA Floorplan**

Transmitter Clock Output Channel (OUTSTAGE_CLK)

The transmitter clock output channel (OUTSTAGE_CLK) module regenerates the SDR clock waveform synchronous to data channels. The module in Figure 11 is composed of a DDR output flip-flop with the D0 input tied to a logic 1 and the D1 input tied to a logic 0. Table 4 contains the module pin descriptions.

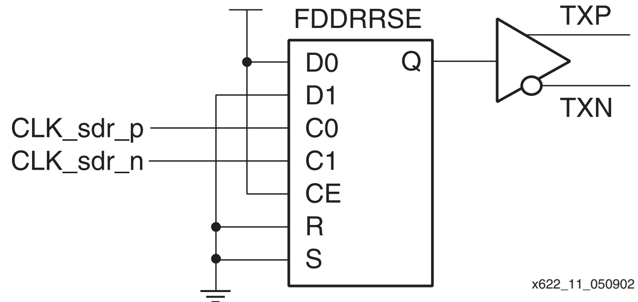

Figure 11: **OUTSTAGE_CLK Module**

Table 4: **OUTSTAGE_CLK Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	CLK_sdr_p	Active High positive edge global SDR clock.
	CLK_sdr_n	Active High negative edge global SDR clock.
Output	TXP, TXN	LVDS output clock

Receiver with 4-to-1 Serializer (RX_SDR_16D_4TO1)

The receiver (RX_SDR_16D_4TO1) module has a high-speed receiver (HSRX_16D_4TO1) and a FIFO for crossing the clock domain boundary between the received clock and the internal system clock. Both of these are discussed in detail in the following sections. Table 5 contains the module pin descriptions. Figure 12 shows the connections between the HSRX and FIFO modules.

Table 5: **Receiver Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	RXP_data<15:0> RXN_data<15:0>	These signals comprise the differential pairs which form the 16-bit receiver input bus.
	REFCLK	Receiver reference SDR clock
	RX_sync	Receiver phase synchronization input
	SYSCLK	System global clock for read FIFO
	RE	Active High FIFO read enable
	RST	Active High receiver reset
Output	DATA<63:0>	64-bit bus of received data out of FIFO, synchronous to SYSCLK
	BUFSTAT<3:0>	FIFO buffer Full status
	READY	Active High indicates receiver is ready to receive valid data
	CLK_qdr	QDR of received clock (not on global clock)

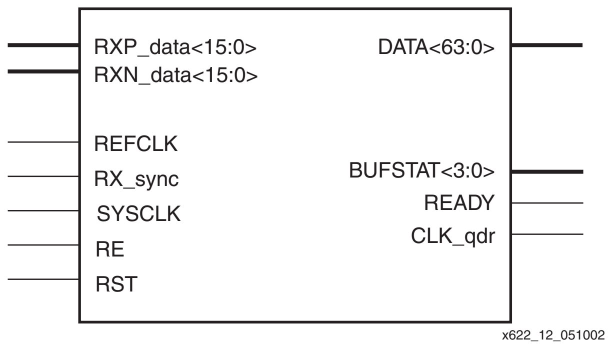

Figure 12: **HSRX_16D_4TO1 Module**

The HSRX_16D_4TO1 module includes a DCM cell with a dynamic phase training circuit to optimally center the receiver clock in the center of the data valid window. This capability leads to two different clock strategies that may be implemented using the REFCLK and RX_sync inputs on the HSRX_16D_4TO1 module. In Figure 13, the REFCLK input is sourced from the same differential input that is used for the transmitter interface and the RX_sync input is connected to the RXP_clk and RXN_clk pins from the SDR interface. This topology has the benefits of using the original reference clock as the DCM clock input source and being able to co-locate the receiver SDR clock input pins with the receiver SDR data channels. This circuit should only be used when the REFCLK and receiver SDR clock frequencies are guaranteed to be identical.

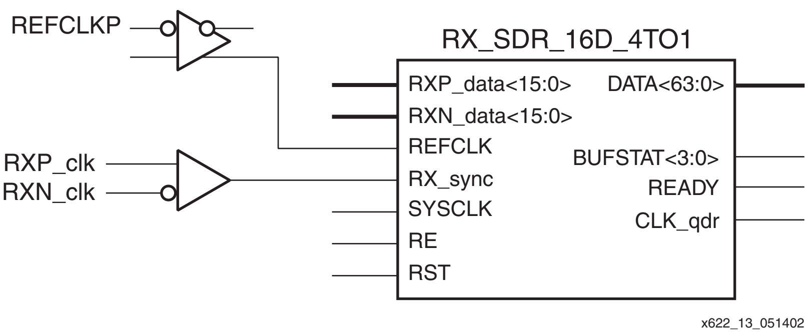

Figure 13: **Receiver Clocking with REFCLK**

In Figure 14, the REFCLK and RX_sync inputs are both connected to the RXP_clk and RXN_clk pins from the SDR interface. This topology has the benefit of using the RX clock as the DCM clock input source eliminating any concerns about frequency drift at the expense of not being able to co-locate the receiver SDR clock input pins with the receiver SDR data channels.

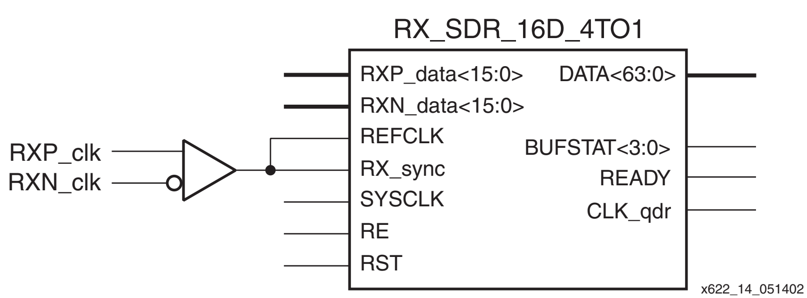

Figure 14: **Receiver Clocking with Received Clock**

The receiver block diagram is shown in Figure 15. The receiver is composed of two modules. Both of these is discussed in detail in the following sections. Moving from left to right, first is the HSRX_16D_4TO1 module. The HSRX_16D_4TO1 module is the high-speed LVDS receiver. It passes unaligned data to the FIFO block for queuing to the system clock domain.

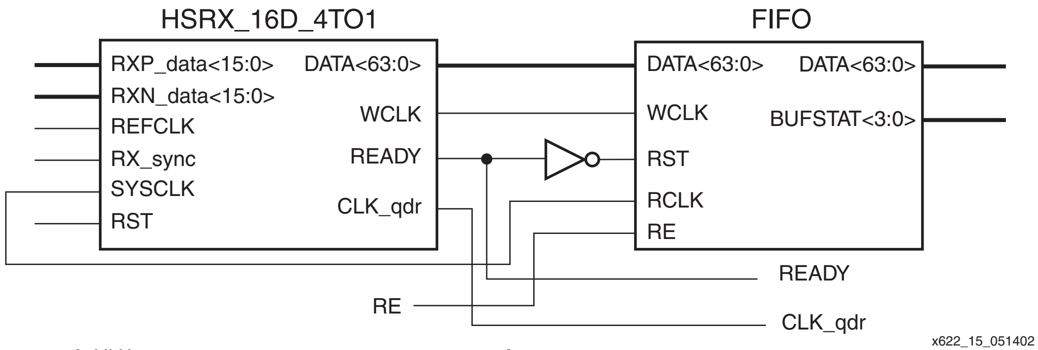

Figure 15: **Receiver Block Diagram**

The CLK_qdr output is provided for use as a system clock if needed, but would typically not be used. When the CLK_qdr output is used a global clock buffer (BUFG) must be inserted before connecting the signal to additional logic.

The control of the FIFO interface is user definable. A typical control system waits until the buffer is near full (i.e., BUFSTAT<3:0> = `1101`) before starting to read and then continues to read until the buffer is near empty (i.e., BUFSTAT<3:0> = `0010`). If the SYSCLK frequencies is identical to CLK_qdr frequency the FIFO would never go empty after the read operation has started. The following verilog code fragment shows an example control circuit.

```
//
// RX FIFO control logic
//
always @ (posedge SYSCLK)
begin
  if (RST_i == 1'b1 || BUFSTAT < 4'b0010 )
     READ_ENABLE = 1'b0;
  else if (READY == 1'b1 && BUFSTAT == 4'b1101 )
     READ_ENABLE = 1'b1;
end
```

High Speed Receiver (HSRX_16D_4TO1)

The 16 data channel high speed receiver (HSRX_16D_4TO1) is composed of 16 deserialization modules (QDR_REG) and one clock phase synchronization module (CLKGEN). Both of these are discussed in detail in following sections.

The module port descriptions are shown in Table 6 and Figure 16 shows the HSRX block diagram.

Table 6: **HSRX Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	RXP_data<15:0> RXN_data<15:0>	LVDS data pins
	REFCLK	Receive Reference SDR clock
	RX_sync	Receiver phase synchronize input
	SYSCLK	System clock
Output	DATA<63:0>	Unaligned data bits
	WCLK	FIFO write clock derived from REFCLK
	READY	Active High ready status
	CLK_qdr	QDR of reference clock (not on global clock)

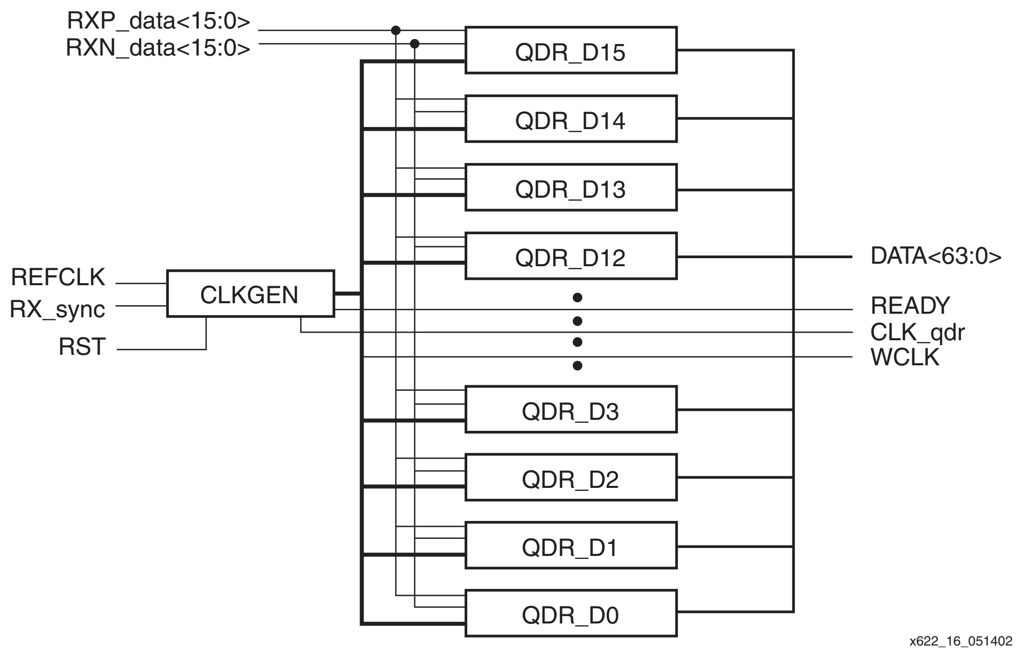

Figure 16: **HSRX_16D_4TO1 Block Diagram**

Quad Data Rate Register (QDR_REG)

The quad data rate register provides 4 to 1 deserialization of the input SDR data using a tree of DDR registers including the DDR input registers in the IOB. Each of the DDR registers is clocked by a global clock (CLK_ddr) at half of the SDR frequency. CLK_ddr is derived from the SDR reference clock and phase aligned with the receiver clock for optimal performance. The module pin descriptions are shown in Figure 17 and listed in Table 7.

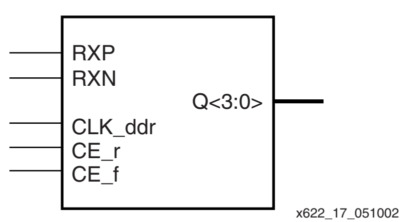

Figure 17: **QDR_REG Module**

Table 7: **QDR_REG Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	RXP RXN	LVDS data
	CLK_ddr	DDR global clock
	CE_r CE_f	Clock enable for rising edge Clock enable for negative edge
Output	Q<3:0>	Received data

The receiver design files associated with this application note are currently targeted for the left side of the device in banks 6 and 7. Each QDR_REG module will consume half of the resources in the 2-1/2 CLBs adjacent to the IOBs allowing two data channels to be placed in the same row. The RLOC_ORIGIN attribute must be set to the first CLB location and on an even row. Figure 18 shows an example placement with two data channels in a XC2V3000-FF1152 device. In this case, the following placement constraints would be added to the UCF file for the design.

```
INST  "U_rx/QDR_D8"      RLOC_ORIGIN = X0Y80;
NET   "RXP_data<8>"      LOC = R30;
NET   "RXN_data<8>"      LOC = P30;
INST  "U_rx/QDR_D9"      RLOC_ORIGIN = X0Y82;
NET   "RXP_data<9>"      LOC = R25;
NET   "RXN_data<9>"      LOC = P25;
```

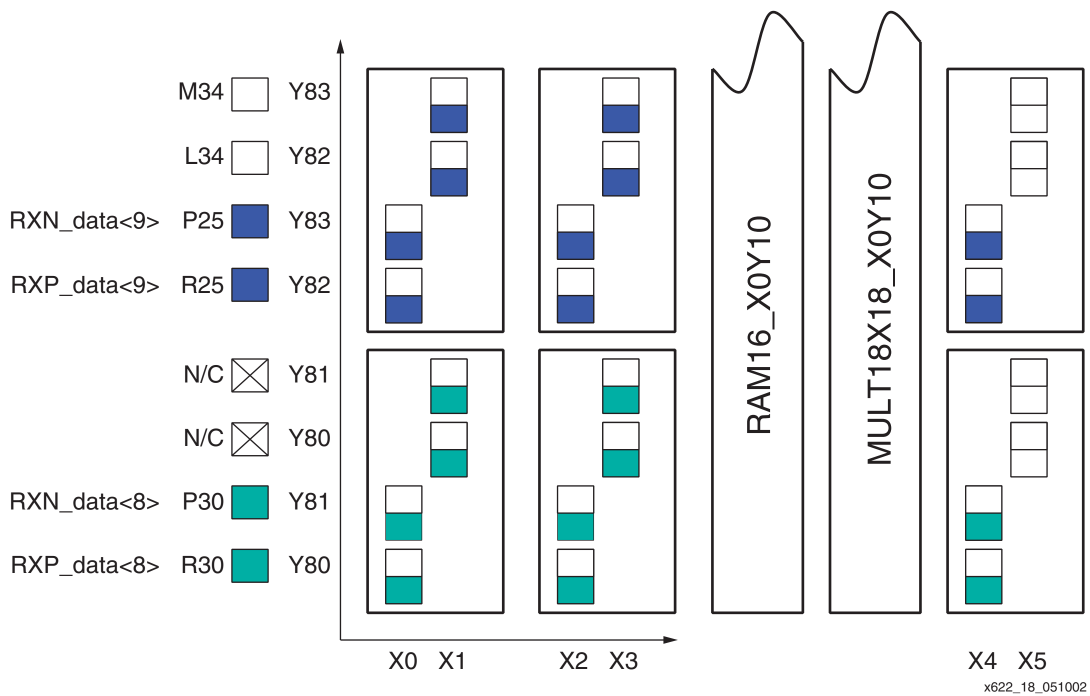

Figure 18: **Receive Data Channel Floorplan**

Receiver Clock Phase Alignment (CLKGEN)

The receiver interface implements a clock phase alignment circuit that provides optimal performance by aligning the reference clock to the center of the data valid window using the variable phase shifting capabilities of the Virtex-II DCM. The CLKGEN module port descriptions are graphically shown in Figure 19 and listed in Table 8.

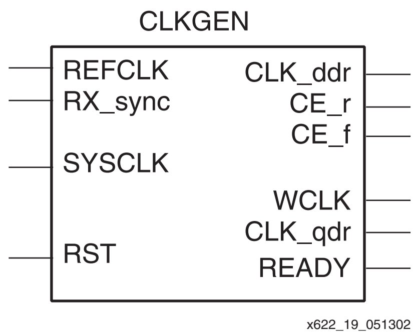

Figure 19: **CLKGEN Module**

Table 8: **CLKGEN Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	REFCLK	Reference SDR clock
	RX_sync	Receiver phase synchronization input
	RST	Reset
	SYSCLK	System clock for phase control alignment logic
Output	CLK_ddr	Half receive clock
	CLK_qdr	Quarter receive clock
	CE_r, CE_f	DDR clock enable signals
	WCLK	FIFO Write clock
	READY	Active High clock ready status

Figure 20 shows the block diagram of the CLKGEN module. The DCM in this instance also uses the CLKIN_DIVIDE_BY_2 attribute to reduce the actual clock input frequency to half of the reference SDR clock frequency. The output of the DCM (CLK0) is connected to a BUFG providing a DDR global clock to the receiver data channels. The RX_sync input is connected to the data pins of DDR registers located in the IOB and the CLK_ddr global clock is used as the clock input. This circuit is identical to the data. The outputs of the DDR input registers are in turn connected to registers in the fabric before passing to the phase synchronization logic.

Figure 21 shows waveforms of the receiver data patterns, RX_sync, REFCLK, and three states of the CLK_ddr signal. Since the RX_sync is actually an SDR clock signal, it has twice as many transitions as the data channels. When the CLKGEN module has been reset the DCM will lock on the REFCLK signal and will start with a phase adjustment of –254 (one full negative clock period).

After the DCM has locked, the phase adjust block will increment the phase value until the RX_sync DDR registers are correctly receiving a steady logic 0 from both outputs of the DDR registers. The phase increment will continue until a steady logic 1 is received from both outputs of the DDR registers. This is the start of the data valid window and the start of a window counter.

The phase will continue to increment until a steady logic 0 is received and will then continue to be incremented until the logic 0 pattern is broken, indicating the end of the data valid window and stopping the window counter. The phase is then decremented until the counter value is half of the window counter value.

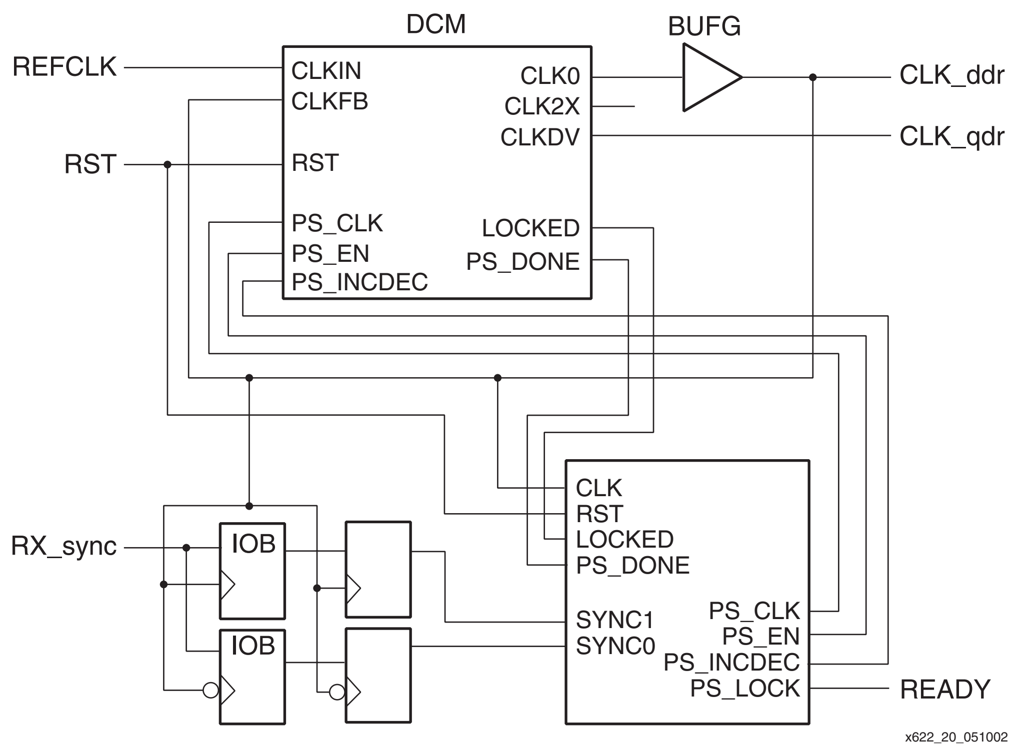

Figure 20: **CLKGEN Block Diagram**

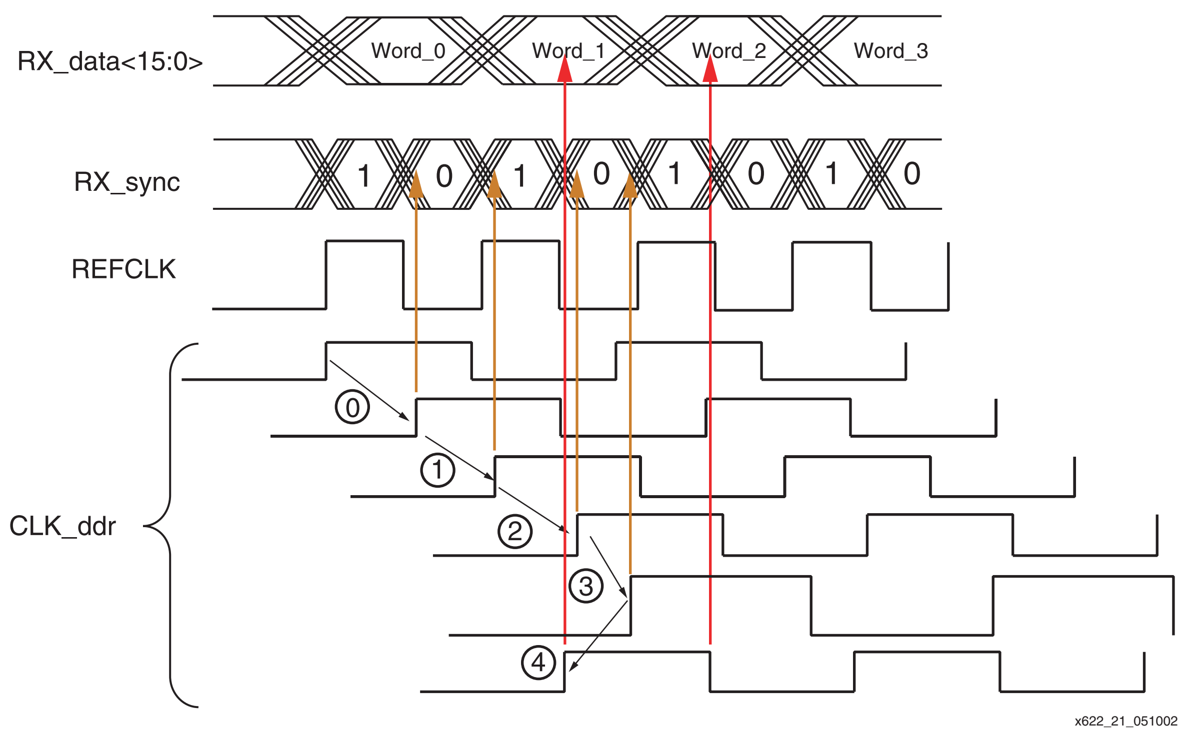

Figure 21: **Clock Phase Synchronization**

Receiver FIFO (FIFO)

A 512 deep, 64-bit FIFO is used for crossing between the receiver and system clock domains. The buffer status bus (BUFSTAT) shows the number of blocks of sixteen 64-bit words left in the FIFO and is intended for use with high water to low water FIFO control system.

The active High RST signal clears both read and write counters. Figure 22 shows the FIFO module, and Figure 23 shows the FIFO block diagram. The module pin descriptions are listed in Table 10.

The control of the FIFO interface is user definable. A typical control system waits until the buffer is near full (i.e., BUFSTAT<3:0> = 1101) before starting to read and then continues to read until the buffer is near empty (i.e., BUFSTAT<3:0> = 0010). If the SYSCLK frequency is identical to CLK_qdr frequency the FIFO would never go empty after the read operation has started. The following verilog code fragment shows an example control circuit.

```
//
// RX FIFO control logic
//
always @ (posedge SYSCLK)
begin
  if (RST_i == 1'b1 || BUFSTAT < 4'b0010 )
     READ_ENABLE = 1'b0;
  else if (READY == 1'b1 && BUFSTAT == 4'b1101 )
     READ_ENABLE = 1'b1;
end
```

The FIFO write data is organized to accept four data channels per block SelectRAM that is used with the following mapping:

RX_data<3:0>	FIFOBLK0
RX_data<7:4>	FIFOBLK1
RX_data<11:8>	FIFOBLK2
RX_data<15:12>	FIFOBLK3

The block SelectRAM must be placed in the location closest to the IOB ring using a LOC constraint in the NCF file. In the placement example shown in Figure 18,

```
INST U_fifo/FIFOBLK1    LOC = RAMB16_X0Y10;
```

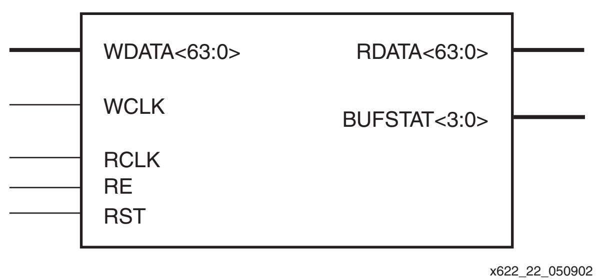

Figure 22: **FIFO Module**

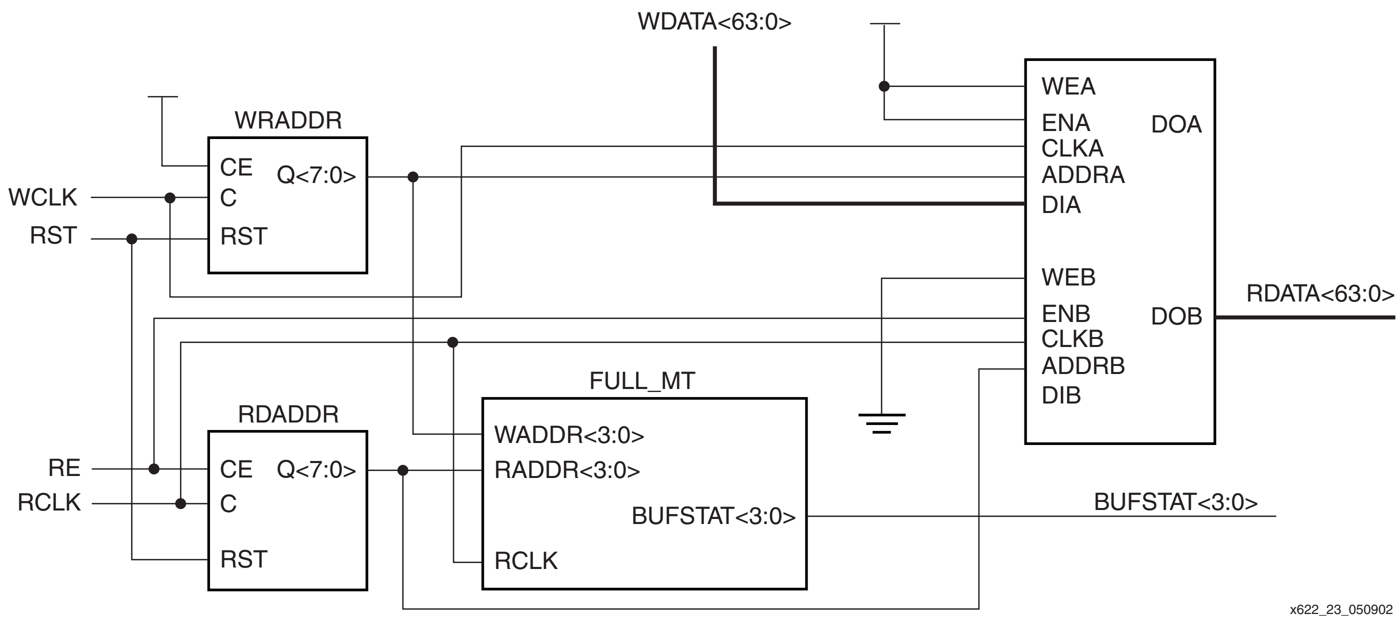

Figure 23: **FIFO Block Diagram**

Table 9: **FIFO Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	WDATA<63:0>	Write data bus
	WCLK	Write CLK
	RCLK	Read CLK (system clock)
	RE	Active High Read enable
	RST	Active High FIFO reset
Output	RDATA<63:0>	Read data bus
	BUFSTAT<3:0>	Buffer status

FULL_MT Overview

Figure 24 shows the FULL_MT module, a simple Full versus Empty generator. The FULL_MT module synchronizes the Write address to the Read clock domain. The buffer status is synchronous to the Read clock (WCLK). Figure 25 shows the FULL_MT block diagram. The module pin descriptions are listed in Table 10.

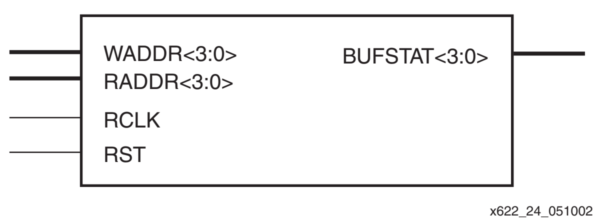

Figure 24: **FULL_MT Module**

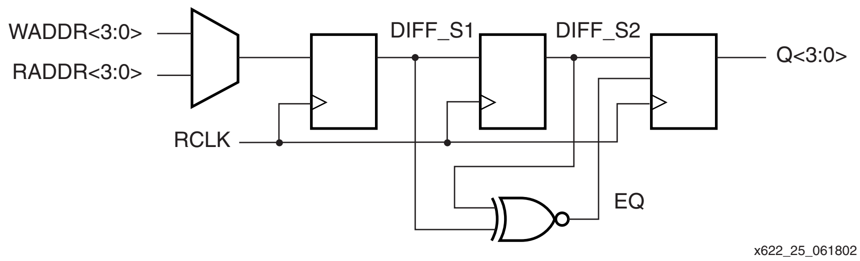

Figure 25: **FULL_MT Block Diagram**

Table 10: **FULL_MT Module Pin Definitions**

I/O Type	Module Pin Name	Definition
Input	WADDR<3:0>	Top 4 bits of Write address bus
	RADDR<3:0>	Top 4 bits of Read address bus
	RCLK	Read CLK
	RST	Reset
Output	BUFSTAT<3:0>	Buffer status

PCB Design Considerations

The generated differential data and clock signals need to be routed very carefully on the PCB and the trace lengths strictly controlled. Ideally, all the signals should have the same delay. If the clock trace has a different delay, the receiver DCM can correct it; as a minimum, however, all the data and the frame signals should be matched to within a few tens of picoseconds for the best circuit operation. The physical characteristics of the traces and the PCB are discussed more fully in XAPP233.

Due to the low pulse widths of the high speed frequency of the differential clock signals the mean voltage levels of the TXP and TXN signals may diverge if the duty cycles are more than 2% different above 500 MHz. This effect is easily compensated for by using an 0.1µF capacitor and a 1K Ω resistor to create an AC termination and DC bias circuit as shown in Figure 26. If this circuit or an equivalent is not added to the PCB design the receiver may not receive valid LVDS input levels.

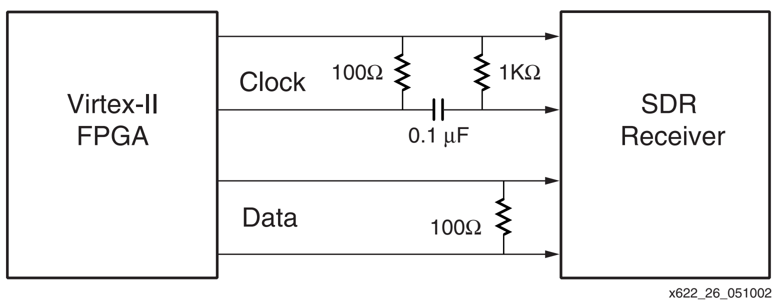

Figure 26: **AC Termination and DC Bias Circuit**

Reference Design

The Verilog and EDIF design files for the implementations discuss in this application note are available on the Xilinx FTP site at:

ftp://ftp.xilinx.com/pub/applications/xapp/xapp622.zip

The readme.txt file includes further implementation details.

Conclusion

The Virtex-II device in a -5 speed grade can implement single data rate 16-bit LVDS data transmission and reception, at up to 644 MHz.

Appendix A

Using the Differential Clock Macro (LVDS_DIFF_IBUFG)

The following procedure should be followed to effectively use the LVDS_DIFF_IBUFG macro for clock inputs.

- Instantiate the macro on the top most level of hierarchy in the design. Failing to do this results in a PAR error.
- Constrain the macro to a clock input pin location. The pin location constraint must be the LVDS positive input pin. Table 11 lists, by package, the possible locations to place the macro.
- Connect the two outputs of the macro to two global buffers (BUFGs.) For a given pin location, the positive and negative terminals must be constrained to the correct BUFG to ensure an optimal route to the clock tree.

Example

This example uses the macro in the top LVDS pair of bank 4 in an FG256 device package.

Verilog Code

```
LVDS_DIFF_IBUFG IB_refclk
(.I( REFCLKP ), .IB( REFCLKN ), .O( REFCLK_in_p ), .OB(REFCLK_in_n ));
BUFG  BG_sdr_p ( .I( REFCLK_in_p ) , .O(  CLK_sdr_p  ) );
BUFG  BG_sdr_n ( .I( REFCLK_in_n ) , .O(  CLK_sdr_n  ) );
```

Given this Verilog code, a UCF file with the constrained values shown in Table 11 must be written with the following lines:

```
INST  IB_refclk/PADS  LOC = P9;
# This constraint places the macro in pins P9, N9, where P9 is + LVDS pin
INST  BG_sdr_p        LOC = BUFGMUX2P;
# This constraint is for the positive output of the macro
INST  BG_sdr_n        LOC = BUFGMUX3S;
# This constraint is for the negative output of the macro
```

Table 11: **Macro Constraints**

Package Type	Bank Number	Possible Macro Locations Top Pair: Pos*, Neg Bottom Pair: Pos*, Neg	BUFGMUX Locations (By Pin Pair) Pos*, Neg
CS144	Bank 0	B6*, C6	6S*, 7P
		D7*, A6	4S*, 5P
	Bank 1	B7*, A7	2S*, 3P
		B8*, A8	0S*, 1P
	Bank 4	M8*, L8	2P*, 3S
		K7*, N8	0P*, 1S
	Bank 5	M7*, N7	6P*, 7S
		M6*, N6	4P*, 5S

Table 11: **Macro Constraints *(Continued)***

Package Type	Bank Number	Possible Macro Locations Top Pair: Pos*, Neg Bottom Pair: Pos*, Neg	BUFGMUX Locations (By Pin Pair) Pos*, Neg
FG256	Bank 0	C8*, D8	6S*, 7P
		A8*, B8	4S*, 5P
	Bank 1	B9*, A9	2S*, 3P
		D9*, C9	0S*, 1P
	Bank 4	P9*, N9	2P*, 3S
		T9*, R9	0P*, 1S
	Bank 5	R8*, T8	6P*, 7S
		N8*, P8	4P*, 5S
FG456	Bank 0	C11*, D11	6S*, 7P
		A11*, B11	4S*, 5P
	Bank 1	F13*, F12	2S*, 3P
		D12*, E12	0S*, 1P
	Bank 4	Y12*, W12	2P*, 3S
		AB12*, AA12	0P*, 1S
	Bank 5	Y11*, AA11	6P*, 7S
		V11*, W11	4P*, 5S
FG676	Bank 0	E13*, F13	6S*, 7P
		C13*, D13	4S*, 5P
	Bank 1	H15*, H14	2S*, 3P
		F14*, G14	0S*, 1P
	Bank 4	AB14*, AA14	2P*, 3S
		AD14*, AC14	0P*, 1S
	Bank 5	AB13*, AC13	6P*, 7S
		Y13*, AA13	4P*, 5S
BG575	Bank 0	F12*, G12	6S*, 7P
		H12*, H11	4S*, 5P
	Bank 1	A14*, A13	2S*, 3P
		C13*, B13	0S*, 1P
	Bank 4	W13*, V13	2P*, 3S
		U13*, U14	0P*, 1S
	Bank 5	AD11*, AD12	6P*, 7S
		AB12*, AC12	4P*, 5S

Table 11: **Macro Constraints *(Continued)***

Package Type	Bank Number	Possible Macro Locations Top Pair: Pos*, Neg Bottom Pair: Pos*, Neg	BUFGMUX Locations (By Pin Pair) Pos*, Neg
BG728	Bank 0	C14*, C13	6S*, 7P
		E14*, F14	4S*, 5P
	Bank 1	H14*, G14	2S*, 3P
		B15*, A15	0S*, 1P
	Bank 4	AG15*, AF15	2P*, 3S
		AA14*, Y14	0P*, 1S
	Bank 5	AB14*, AC14	6P*, 7S
		AF13*, AG13	4P*, 5S
FF896	Bank 0	G16*, H16	6S*, 7P
		C16*, C17	4S*, 5P
	Bank 1	C14*, C15	2S*, 3P
		F14*, F15	0S*, 1P
	Bank 4	AE15*, AD15	2P*, 3S
		AH15*, AH14	0P*, 1S
	Bank 5	AH17*, AH16	6P*, 7S
		AD16*, AE16	4P*, 5S
FF1152	Bank 0	J18*, K18	6S*, 7P
		E18*, E19	4S*, 5P
	Bank 1	E16*, E17	2S*, 3P
		H16*, H17	0S*, 1P
	Bank 4	AG17*, AF17	2P*, 3S
		AK17*, AK16	0P*, 1S
	Bank 5	AK19*, AK18	6P*, 7S
		AF18*, AG18	4P*, 5S
FF1517	Bank 0	J20*, H20	6S*, 7P
		D21*,C21	4S*, 5P
	Bank 1	F20*, F19	2S*, 3P
		H18*, H19	0S*, 1P
	Bank 4	AM20*, AL20	2P*, 3S
		AT19*, AU19	0P*, 1S
	Bank 5	AP20*, AP21	6P*, 7S
		AN22*, AN21	4P*, 5S

Table 11: **Macro Constraints *(Continued)***

Package Type	Bank Number	Possible Macro Locations Top Pair: Pos*, Neg Bottom Pair: Pos*, Neg	BUFGMUX Locations (By Pin Pair) Pos*, Neg
BF957	Bank 0	E16*, E17	6S*, 7P
		A17*, A18	4S*, 5P
	Bank 1	C15*, C16	2S*, 3P
		H15*, H16	0S*, 1P
	Bank 4	AL15*, AL14	2P*, 3S
		AJ15*, AH15	0P*, 1S
	Bank 5	AH17*, AJ16	6P*, 7S
		AD17*, AD16	4P*, 5S

Notes:

1. The "*" symbol denotes the positive LVDS pin

SFI-4 and XSBI Compatibility Checklist

Virtex-II devices are compatible to the SFI-4 Implementation Agreement revision 1.0 and XSBI specifications in the IEEE P802.3ae draft 4.1. The Table 12 and Table 13 summarize the numerical analysis of the specifications versus Virtex-II devices.

Table 12: **TX Timing Comparisons of SFI-4 versus LVDS SDR Design in Virtex-II Devices**

Description	SFI-4 Value	XSBI Value	LVDS SDR Design using a Virtex-II Device Value
Clock Period	1/(622.08 MHz)	1/(644.53 MHz)	1/(622.08 MHz) for SFI-4 1/(644.53 MHz) for XSBI
Duty cycle = High CLK Pulse Width Divided by Clock Period	40/60	40/60	45/55
20-80% rise, Fall Times[1]	100 – 250 ps	100 – 250 ps	400 ps
Data Invalid Window With Respect to Clock Edge[2]	400 ps	400 ps	242 ps

Notes:

1. Although the Virtex-II rise and fall times exceeds the SFI-4 document, based upon interoperability tests, Virtex-II devices are compatible with SFI-4 devices.
2. Data invalid window includes total system jitter, clock skew, and package skew. TX = (Jitter + $T_{CKSSKEW}$ + $T_{PKGSKEW}$).

Table 13: **RX Timing Comparisons of SFI-4 versus LVDS SDR Design in Virtex-II Devices**

Description	SFI-4 Value	XSBI Value	LVDS SDR Design using a Virtex-II Device Value
Clock Period	1/(622.08 MHz)	1/(644.53 MHz)	1/(622.08 MHz) SFI-4 1/(644.53 MHz) XSBI
Duty Cycle = High CLK Pulse Width Divided by Clock Period	45/55	45/55	45/55
20-80% rise, Fall Times[1]	100 – 300 ps	100 – 300 ps	400 ps
Total Setup and Hold Time. Defines Data Valid Window with Respect to Clock at Framer Pin	Setup = 300 ps Hold = 300 ps Data Valid Window = 600ps	Data Valid Window = 600 ps	590 ps[2]

Notes:

1. Although the rise and fall times of Virtex-II device exceeds that of the SFI-4 document, based upon interoperability tests, Virtex-II is compatible with SFI-4 devices.
2. Data valid window includes sampling error, clock skew, and package skew.
   RX = (T_{SAMP} + T_{CKSKEW} + $T_{PKGSKEW}$).
   T_{SAMP} is known to be 500 ps (as specified in the Virtex-II source-synchronous data sheet).
   $T_{CKSSKEW}$ is less than 50 ps (based on analysis of the SDR design's clock distribution). These two numbers are fixed. Careful choice of pin locations, or by deskewing the package on the PCB, allows the inter-pin skews to be as low as 40 ps.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
05/17/02	1.0	Initial Xilinx release.
05/30/02	1.1	Revised Table 2.
07/02/02	1.2	Revised Figure 25 and added **Appendix A**.