XAPP234 (v1.1) March 15, 2000

Virtex SelectLink Communications Channel

Author: John Logue

Summary

Systems with two or more FPGAs often require high-bandwidth data paths between devices. As the clock period and switching times of digital circuits become shorter, straightforward methods of transferring data between devices are often inadequate. At high frequencies, signal propagation delay and reflections that occur in conductors just a few centimeters long must be taken into account. The SelectLink™ communications channel utilizes special features of the Virtex™ series of FPGAs including Delay Locked Loops (DLL), block SelectRAM+™ memory, and the SelectI/O™ interface, to create a system to move large amounts of data between FPGAs at very high speeds. A code generation tool available at www.xilinx.com allows logic designers everywhere to instantly create customized SelectLink Verilog source code. The modules are easily instantiated in the designers top level code for a complete system solution.

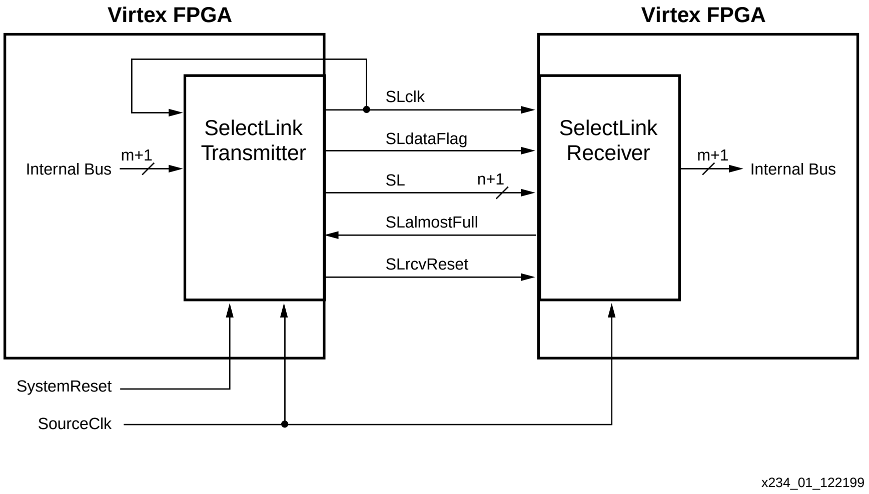


Figure 1: **SelectLink System Level**

Introduction

As the internal speed of ICs continually increases, external buses can easily be the bottleneck that limits system performance. High-speed external bus ports using new signal standards and protocols are found on state of the art memories, processors, and other integrated circuits. These new techniques increase bandwidth without resorting to very wide external buses requiring multiple internal cycles for a single transfer. Figure 1 is a high level SelectLink block diagram.

The SelectLink channel creates a high-bandwidth Virtex-to-Virtex path using similar techniques, and it does this with resources that are standard on all Virtex FPGAs. Since it uses existing programmable resources, the SelectLink feature can be configured to meet the individual needs of each design, while leaving the remaining resources for other needs. The SelectLink channel has the following notable features:

Transmission Line Device Interface

The SelectLink physical connection is a transmission line with a controlled impedance, rather than just a voltage transfer medium. The SelectLink channel employs clock-forwarding, a technique in which the transmit clock is sent to the receiver along with the data. This clock is used at the receiver to recover the data. Since the transmit clock remains in phase with the data as both propagate to the receiver, very high frequency data can be recovered at the receiver without errors due to set-up or hold violations.

Unidirectional

Data is sent in one direction only on a single SelectLink channel, from one transmitter to one receiver. Two channels may be used for bi-directional operation.

Double Data Rate

Two bits of data are carried on each data line during one period of the transmit clock.

Configurable Bus Widths

The width of the internal FIFO ports and the width of the external inter-chip bus can be configured to match the design requirements.

Data Buffering

FIFO data buffers are included in both the transmitter and receiver modules. The depth of the FIFOs is configurable.

Data Flow Control

Flow control is built-in to the SelectLink channel. Inside the transmitter FPGA, the user sees a FIFO write port with a Full flag. Inside the receiver FPGA, the user sees a FIFO read port with an Empty flag. To the user, these ports appear to be write and read ports of the same FIFO.

Data Funneling and Expansion

Because data is transferred at twice the clock rate, the external SelectLink bus need never be wider than 1/2 the width of the internal FIFO ports. Designers can save pins and interconnects by selecting an even narrower external bus, if it can handle the average data rate required by the application. Each data word is automatically broken down into separate smaller vectors in the transmitter and reassembled into the original width in the receiver.

Fully Synchronous User Clocks

The user clocks in the transmitter and receiver FPGAs are in phase with each other. By using standard single data rate transfers, the FPGAs could, for example, both synchronously communicate with a third IC placed between them.

Online Source Code Generation

A code generation engine available at www.xilinx.com relieves the designer of the time consuming and error-prone task of manually editing each HDL module to match their requirements. After the customized SelectLink source code is downloaded, it can easily be merged with the user's design by making just a few connections at the top hierarchy level.

The SelectLink bus is especially suited to applications that stream large amounts of data from one FPGA to another. Because of latency (the delay between the time data is written at the transmitter and the time it is available at the receiver), it is less effective for applications that require a response after sending just a few bytes.

Figure 2 is a conceptual diagram of the SelectLink communications channel, depicting the "virtual" FIFO that the user sees between the FPGAs.

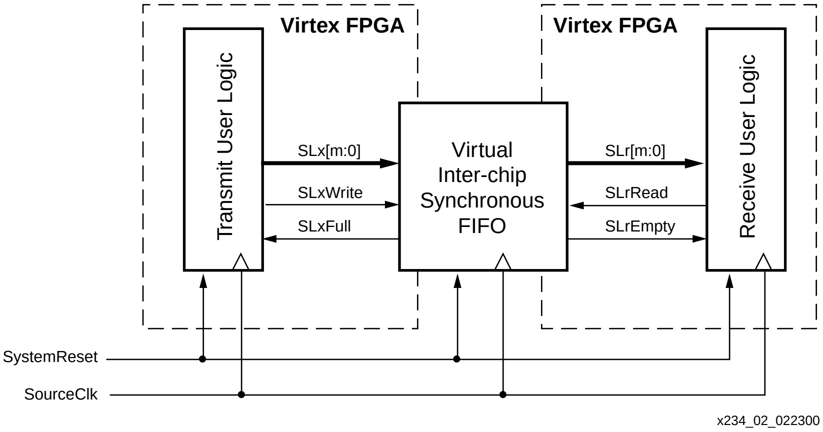


Figure 2: **SelectLink Conceptual Diagram**

External Bus

Figure 1 includes all of the required external bus signals. These signals are broken into two groups, External System (Table 1), and External Interface (Table 2).

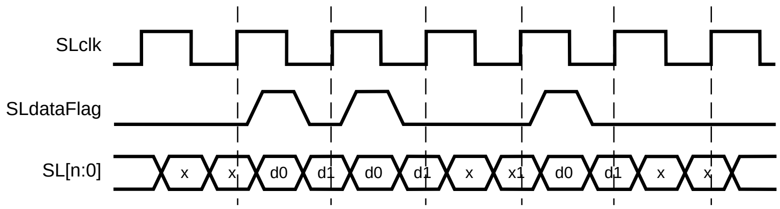
Table 1: **SelectLink External System Signals**

Signal Name	Direction	FPGA	Description
System Reset	Into FPGA	Transmit	SystemReset asynchronously initializes all SelectLink interfaces and user logic. When SystemReset is active, all SelectLink DLLs are held in reset. SystemReset feeds only the transmitter chip, but it is propagated through to the receiver chip via signal SLrcvReset.
SourceCLK	Into FPGA	Both	SourceClk is the only system clock source needed for a SelectLink DLL. SelectLink signal SLclk is sourced from this signal, but it is in phase with SourceClk only at the transmitter. By the time it reaches the receiver, the phase has shifted. Both the transmitter and receiver internal user clocks are in phase with SourceClk (and thus with each other), provided it reaches the respective FPGA clock input pins at the same time. That is, the propagation times from the source must be matched. (If the application doesn't require that the user clocks be in phase, the equal propagation requirement can be waived.)
SLclkFB	Into FPGA	Transmit	An IBUFG pin driven by signal SLclk. This DLL feedback loop is needed to establish the correct timing between SelectLink clock and data. SLclk should be assigned to a pin near the IBUFG pin, so this connection is short. See Table 2.

Table 2: **SelectLink External Interface Signals**

Signal Name	Direction	Description
SLclk	X → R	SLclk is the data clock. The frequency of SLclk is 1/2 the data rate of each bit of SL. The high-going edge of SLclk leads the switching point on SLdataFlag and SL[n:0] to ensure adequate hold time at the receiver. SLclk has a duty cycle of approximately 50/50, but the falling edge is not used at the receiver. The high and low times of SLclk are referred to as Phase 0 and Phase 1, respectively. SLclk is fed back to an IBUFG pin on the transmitter FPGA. This is needed to establish the correct timing between SLclk and SL[n:0].
SL[n:0]	X → R	SL is the double data rate bus. Data switches about one ns after the high going edge of SLclk, and 1/2 period later. Each datum on SL is referred to as a parcel.
SLdataFlag	X → R	When SLdataFlag is true during Phase 0, it means that a parcel is being transferred during this phase, and a parcel will be transferred on the following Phase 1. When SLdataFlag is false during Phase 0, it means that this is an idle cycle; i.e., no data is transmitted during either phase. SLdataFlag has the same timing as SL. SLdataFlag is always false during Phase 1.
SLalmostFull	X ← R	SLalmostFull is an asynchronous flow control signal that goes true when the receive FIFO is nearly full. The transmitter will send data only when this signal is false. SLalmostFull is also used during reset initialization to tell the transmitter when the receiver is ready for operation.
SLrcvReset	X → R	SLrcvReset is an asynchronous signal that is active until the DLL that generates SLclk is Locked. The purpose of this signal is to hold the receiver DLLs in reset until signal SLclk is stable.

Figure 3 shows the relative timing of SLclk, SLdataFlag, and SL[n:0]. When SLdataFlag is active during Phase 0, valid data is transmitted during the corresponding Phase 0 and the following Phase 1. When SLdataFlag is inactive during Phase 0, no data is transmitted during either phase, and the SL bus levels are "don't care".



x234_03_091799

Figure 3: **External Timing**

The SelectLink feature is not limited to a particular electrical interface (physical layer). The code generation engine available at www.xilinx.com allows the designer to chose any of the SelectI/O standards. (See XAPP133 for information on the various I/O standards supported by Virtex devices.)

When the transmitter and receiver FPGAs are on the same printed circuit board, a high-speed single-ended standard, such as SSTL3 Class II, is a good choice. For transmission between PCBs, the LVDS differential standard (available in the Virtex-E families) is recommended, because it provides the best noise immunity.

Whatever physical layer is used, the propagation time from the transmitter output pin to the receiver input pin must match each other for the following signals (with a maximum tolerance of 500 ps):

SLclk SLdataFlag SL[n:0]

The propagation delay of the physical layer does not affect the data rate. The physical connection may be any length, provided skew tolerance and signal integrity are maintained. (As discussed later in this document, propagation delay may increase the latency.)

Two of the External Interface signals, SLrcvReset and SLalmostFull, are asynchronous and switch at a relatively low rate. Any physical connection that maintains signal integrity is adequate for these signals.

Module Hierarchy

Internal to the FPGAs, the SelectLink channel is implemented in Verilog modules as follows:

Xchip - Transmitter FPGA

SLXtc - Main Transmit module

Xclk - Transmit clock module

Rchip - Receiver FPGA

SLRtc - Main Receive module

Rclk - Receive clock module

Figure 4 shows how these modules connect to each other, and how they connect to the user's top-level modules. Figures 5 and 6 represent modules SLXtc and SLRtc, respectively. These diagrams show the primary data and control paths, but they are not complete schematic diagrams. The reader is referred to the Verilog listing for design details.

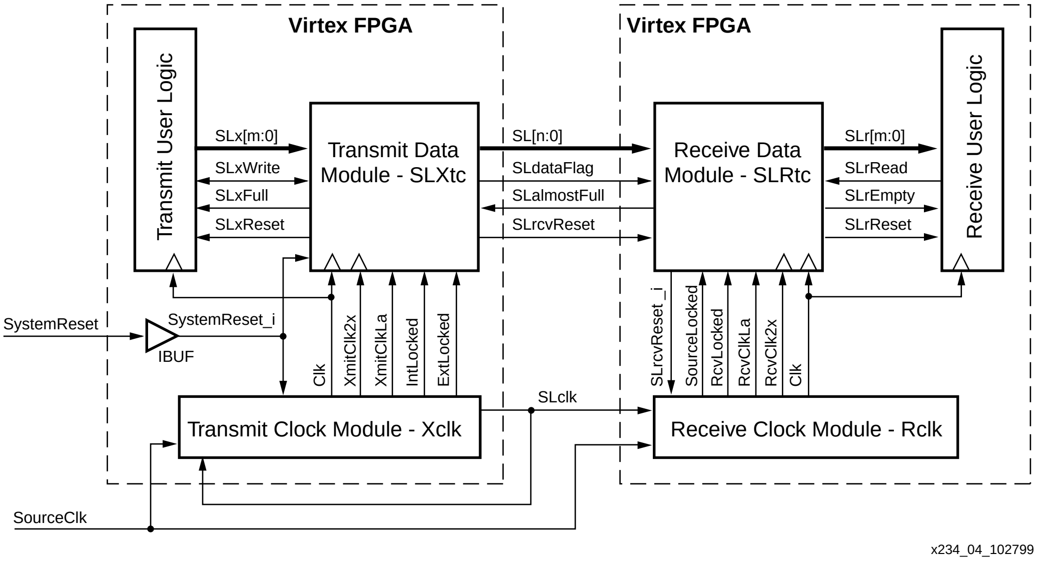


Figure 4: **SelectLink Top-Level Modules and Interconnects**

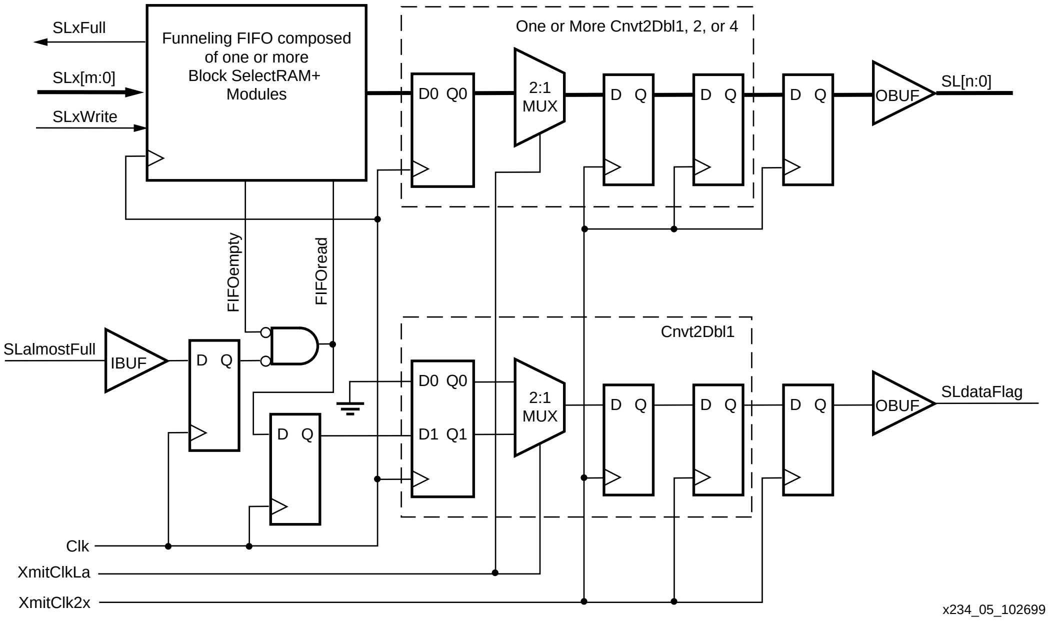


Figure 5: **Module SLXtc**

As shown in Figure 5, dual-port block RAM is used in the transmit module to create a FIFO that may have write and read busses that differ in width. This is an efficient way to "funnel" data to a narrower width.

Words read from the FIFO are converted to double data rate with module Cnvt2DblN, where N is 1, 2, or 4. N is the width of the module output bus. For example, module Cnvt2Dbl4 converts an 8-bit byte to a double data rate 4-bit nibble by multiplexing the lower and upper nibbles, in that order, at the 2x clock rate. Multiple copies of the same Cnvt2DblN module are used to span the full width of the bus. The largest module width that evenly divides into the full width is used.

Whenever the FIFO is not empty, and SLalmostFull is not true, a word is read from the FIFO and written to the SelectLink channel. The SLdataFlag signal is also converted to double data rate by clocking it through a Cnvt2Dbl1 module. This ensures that SLdataFlag and SL[n:0] will be aligned on the external bus.

Module SLRtc reverses the funneling and data rate conversions performed by module SLXtc. However, it is also necessary to phase shift the data, which has been delayed by the data channel path, so that it again aligns with signal Clk.

Figure 6 depicts how SL[n:0] is converted from double to single rate with module Cnvt2SingleN, where N is 1, 2, or 4. N is the width of the module input bus. For example, module Cnvt2Single4 converts a double rate 4-bit nibble to a single rate 8-bit byte by sampling the lower and upper nibbles, in that order, at the 2x clock rate and loading them into a byte-wide register at the 1x clock rate. Multiple copies of the same Cnvt2SingleN module are used to span the full width of the bus. The largest module width that evenly divides into the full width is used.

Signal SLdataFlag is also converted to a 1x rate with a Cnvt2Single1 module. Only the lower (first) bit is used.

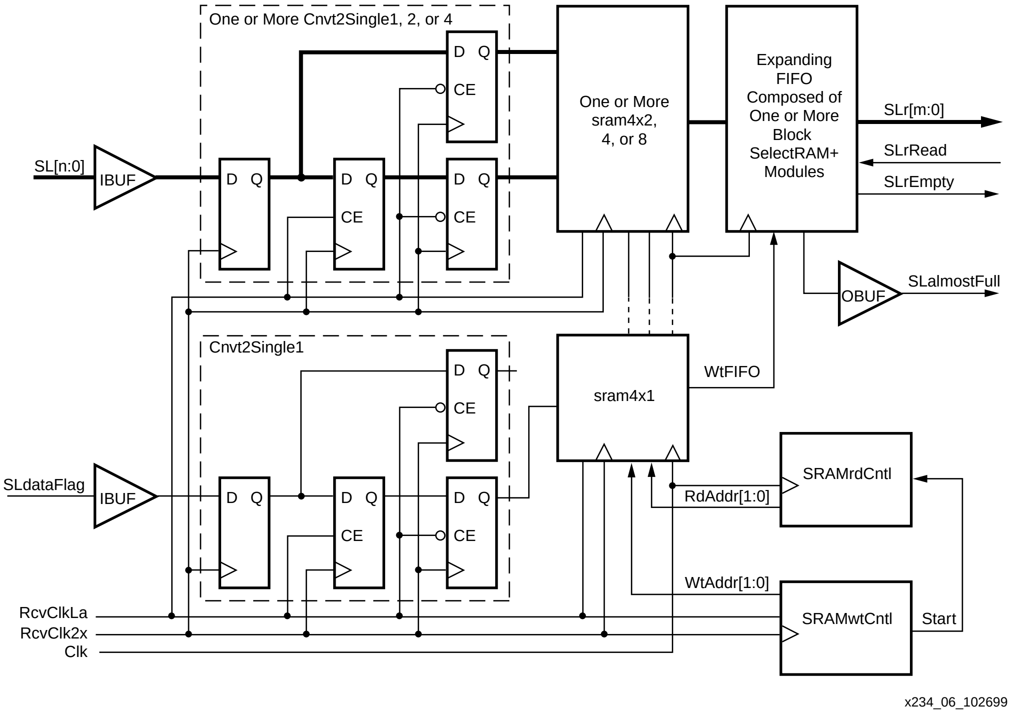


Figure 6: **Module SLRtc**

A four word deep FIFO, created with CLB D flip-flops, is used to phase shift the data. Module sram4xN, where N is 2, 4, or 8, provide the storage elements for these FIFOs. N is the width of the write and read ports, in bits. Multiple copies of the same sram4xN module are used to span the full width of the bus. The largest module width that evenly divides into the full width is used. This same four word deep FIFO includes an sram4x1 module for the SLdataFlag signal.

The four deep FIFO corrects the phase shift by writing synchronously to the receive clock, and reading synchronously to Clk. At start up (the trailing edge of reset), reading is delayed from writing so that data read is guaranteed stable no matter when the FIFO is written. After start up, this FIFO is written and read continuously, whether or not bus SL[n:0] holds valid data.

After the data has been phase corrected, it is buffered and expanded, if necessary, to the full user bus width with a FIFO implemented with dual port block RAM. A word is written to this FIFO whenever SLdataFlag read from the 4x1 RAM is true. The AlmostFull signal from the FIFO drives an OBUF to create signal SLalmostFull. AlmostFull goes true when there are 32 or fewer empty cells in the FIFO. (This is 32 cells at the output port; it might be a multiple of 32 cells at the input port.)

Internal FIFO Busses

The various signals in the internal transmit and receive interfaces are defined in Table 3 and Table 4, respectively. In the Direction columns, "SL" is the SelectLink channel, and "UL" means User Logic

.

Table 3: **SelectLink Internal User Interface Transmit Signals**

Signal Name	Direction	Description
SLxReset	UL ← SL	This signal will be active when SystemReset is active, and it will remain active after SystemReset goes inactive until all four SelectLink DLLs are locked. The user must connect this signal to asynchronously reset all control logic that interfaces to the SelectLink channel. It may also be used to reset any other user logic, at the designer's discretion.
SLxFull	UL ← SL	This is the transmit FIFO full flag. The user can write data to the FIFO on the next high-going edge of Clk, provided this signal is false.
SLx[m:0]	UL → SL	This is the transmit FIFO write data bus. When SLxWrite is active, SLx[m:0] must be valid and setup to the high-going edge of Clk.
SLxWrite	UL → SL	This signal indicates that the user wants to write SLx[m:0] to the SelectLink receiver on the next high-going edge of Clk. SLxWrite must be setup to the high-going edge of Clk. This signal should only be activated when SLxFull is false. It will have no effect if it is activated when SLxFull is true.

Table 4: **SelectLink Internal User Interface Receive Signals**

Signal Name	Direction	Description
SLrReset	SL → UL	This signal will be active when SystemReset is active, and it will stay active after SystemReset goes inactive until all four SelectLink DLLs are locked. The user must connect this signal to asynchronously reset all control logic that interfaces to the SelectLink channel. It may also be used to reset any other user logic, at the designer's discretion.
SLrEmpty	SL → UL	This is the receiver FIFO empty flag. The user can read data from the FIFO on the next high-going edge of Clk, provided this signal is false.
SLr[m:0]	SL → UL	This is the receive FIFO read data bus. When SLxRead is active, the next available FIFO data word will appear on this bus in the cycle after the next high-going edge of Clk. It will remain on this bus until SLrRead is activated again.
SLrRead	SL ← UL	This signal indicates that the user wants to read a word from the receive FIFO on the next high-going edge of Clk. SLrRead must be setup to the high-going edge of Clk. This signal should only be activated when SLrEmpty is false. It will have no effect if it is activated when SLrEmpty is true.

Figure 7 shows the timing of the interface signals relative to Clk. This diagram shows a single word written and read, but a word can be written and read every clock cycle, provided SLxFull and SLrEmpty, respectively, are false.

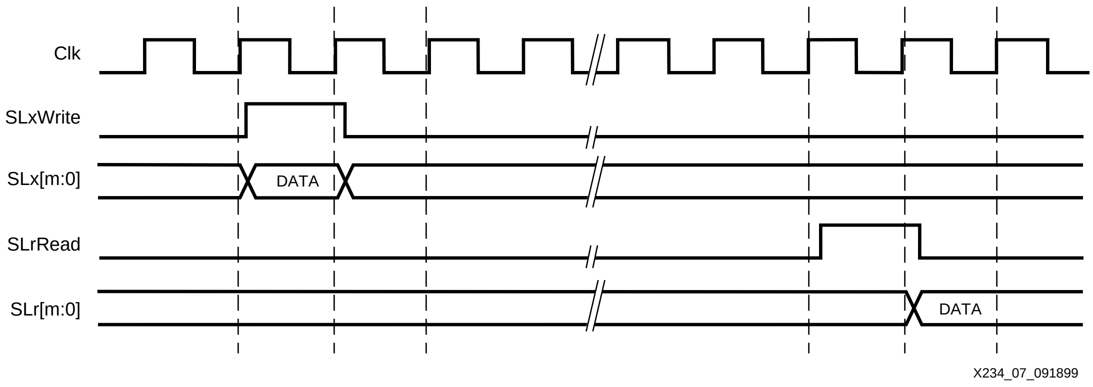


Figure 7: **User Interface Write and Read Timing**

DLL lock status signals are available at the Xchip and Rchip hierarchy levels, as shown in Table 5 and Table 6. These status signals may be used or not used, at the designer's discretion.

Table 5: **SelectLink Internal Transmitter Status Signals**

Signal Name	Direction	Description
ExtLocked	From SelectLink	This signal goes true after the trailing edge of SystemReset when the DLL that generates signal SLclk is Locked.
IntLocked	From SelectLink	This signal goes true after the trailing edge of SystemReset when the transmitter DLL that generates signal Clk is Locked.

Table 6: **SelectLink Internal Receiver Status Signals**

Signal Name	Direction	Description
RcvLocked	From SelectLink	This signal goes true after the trailing edge of SLrcvReset when the DLL that de-skews signal SLclk is Locked.
SourceLocked	From SelectLink	This signal goes true after the trailing edge of SLrcvReset when the receiver DLL that generates signal Clk is Locked.

Data Rate

The data transfer rate is a function of the frequency of SLclk and the width (in bits) of bus SL[n:0]:

Data Rate = 2 x f_{SLclk} x SLextWidth bits/second

where f_{SLclk} is the frequency of SLclk, and SLextWidth is the width,

in bits, of bus SL[n:0].

Using Table 7 as a starting point, the designer can choose the required bus width and device speed grade. Table 7 lists the maximum SelectLink clock frequency that can be expected for various Virtex speed grades. While exact performance is determined only by implementing a complete design, this table may be used as a guideline.

For best performance, it is a good idea to assign the SelectLink bus pins and FIFO block RAM locations (see the section on "Synthesis, Place, and Route" on page 14). The SelectLink channel is designed in such a way that wider buses do not degrade performance significantly. However, performance is reduced if the number of block RAMs is so great that more than one RAM column is needed.

Table 7: **Maximum SelectLink Clock Frequencies**

Family (Speed Grade)	Maximum f_{SLclk} (MHz)	Mbit/sec/pin
Virtex (-4)	70	140
Virtex (-5)	85	170
Virtex (-6)	100	200
Virtex-E (-6)	133	266
Virtex-E (-7)	155.5	311

Latency

SelectLink latency is defined as the number of periods between the time a datum is written on the SLx bus and the time it appears on the SLr bus (assuming it is read immediately when SLrEmpty goes false). Latency is a function of the ratio of the internal and external bus widths, and the propagation time of the external bus.

Table 8 shows the latency as a function of bus width ratios when the external bus propagation time is less than one SLclk period. If the bus propagation time exceeds one period, add one to the value in Table 8 for each additional period, or portion thereof.

Table 8: **Latency as a Function of Bus Width Ratio**

Internal Width/External Width	Latency in SLclk Periods
2/1	11
4/1	12
8/1	14
16/1	18
32/1	26

FPGA Resource Requirements

All SelectLink configurations use clock resources as shown in Table 9. The user logic of many applications will need only signal Clk, which is available at the top level of the hierarchy in both the transmitter and receiver FPGAs. However, the remaining clock resources in the FPGAs are available to the application, as needed.

Table 9: **SelectLink Clock Resource Requirements**

Resource	Transmitter	Receiver
IBUFG	2	2
DLL	2	2
BUFG	2	2

Figures 8 and 9 show details of clock generation in modules Xclk and Rclk, respectively. Notice that a 2x clock is generated in the transmitter to create the double data rate stream, and a 2x clock is generated in the receiver to recover the double data rate stream. The DLL LOCKED status signals from all DLLs are available to the user at the top level of the design hierarchy.

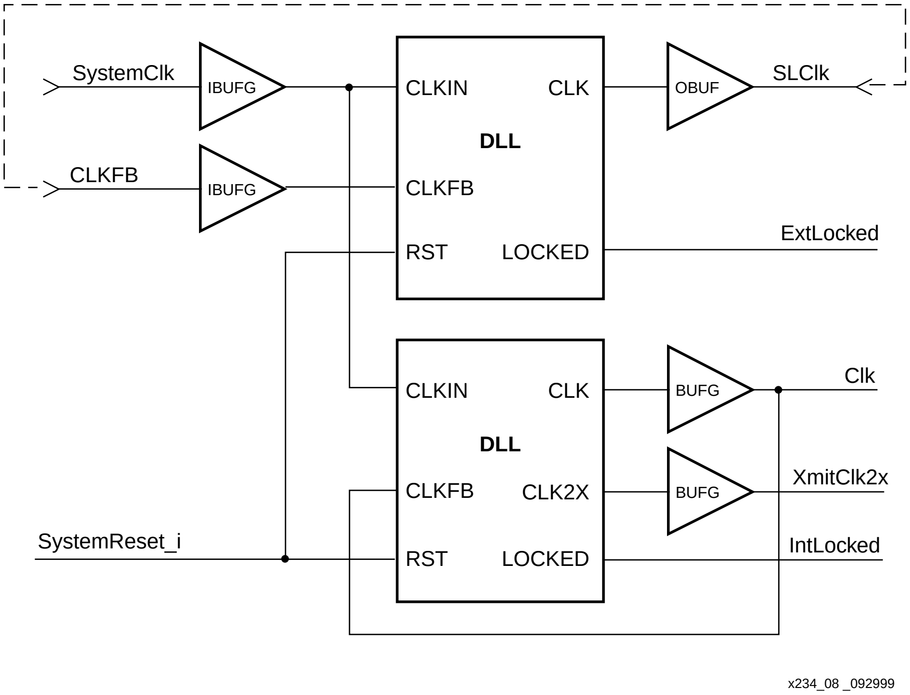


Figure 8: **Xclk Detail**

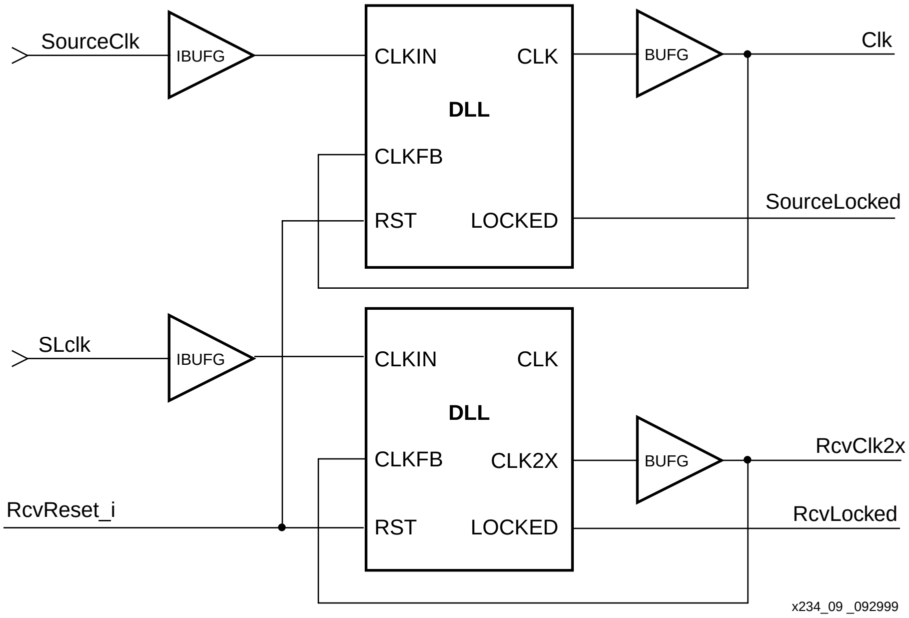


Figure 9: **Rclk Detail**

Signal Clk is effectively the same signal in both the transmit and receive FPGA (provided the propagation time from SourceClk to the FPGAs is matched). It is possible to implement a bi-directional communication link using two SelectLink channels. When using two SelectLink channels, the DLL, IBUFG, and BUFG that generate Clk in module Rclk may be deleted in both FPGAs, with all Clk loads sourced from module Xclk.

Modules Xclk and Rclk also create a "Logic Accessible Clock"; signals XmitClkLa and RcvClkLa, respectively. These signals are synchronously daisy-chained throughout the transmit and receive data modules (SLXtc and SLRtc), and they are used to qualify the 2x clocks whenever a 1x clock (CLK0 or CLK180) is needed.

CLB and block RAM requirements are configuration dependent. Resource requirements for a few typical configurations are shown in Table 10.

Table 10: **CLB and Block RAM Requirements**

Internal Bus Width	External Bus Width	Block RAMs in Xmtr & Rcvr	FIFO Depth in Xmtr & Rcvr	Number of Slices in Xmtr	Number of Slices in Rcvr
16	4	1	256	57	121
16	8	1	256	68	166
32	8	2	256	72	181
32	16	2	256	93	292
64	32	4	256	147	488
64	32	8	512	150	491

Code Generation Engine

A SelectLink Verilog source code generator is available at www.xilinx.com to dynamically generate customized code.

The user defines a specific SelectLink system by entering 3 numbers:

SLintWidth - The width in bits of the internal FIFO user data buses in the transmitter and receiver.

SLextWidth - The width in bits of the external double data rate bus (SL[n:0]).

NumBlks - Number of SelectRAM+ blocks used in each FIFO (transmitter & receiver)

These three parameters are interdependent, and must satisfy these two rules:

SLintWidth/NumBlks = 1, 2, 4, 8, or 16

(2 x SLextWidth)/NumBlks = 1, 2, 4, 8, or 16

Also, the parameters values must fall within these ranges:

(2 <= SLintWidth <= 512)

(1 <= SLextWidth <= 256)

(1 <= NumBlks <= 64)

The depth of each FIFO = (4096 x NumBlks)/SLintWidth

The user is also asked to specify the desired SelectI/O standard for SelectLink drivers and receivers.

After the code is downloaded and stored it can then be simulated. The web page instructs the user to “save as” a .txt file.

The top three levels of Verilog hierarchy include these modules:

BDtest - Board Level Verilog Test Bench

Xchip - SelectLink Transmit Chip

SLXtc - Main Transmit Module

Xclk - Transmit Clock Module

XTU - Transmit Test Unit.

Rchip - SelectLink Receive Chip

SLRtc - Main Receive Module

Rclk - Receive Clock Module

RTU - Receive Test Unit

An alpha-numeric suffix that defines configuration is attached to the base module name of each module that is generated. The suffix elements are separated by underscores. For example, the suffix in module name Xchip_SSTL3_II_32_4_2_0 indicates that this transmitter chip module uses SSTL3 Class II drivers and receivers, the internal bus is 32 bits wide, the external bus is four bits wide, two block RAM modules are used in the FIFO, and the FIFO Full flag goes true when there are zero empty cells in the FIFO. Since the correct modules are automatically generated, the designer generally does not need to be concerned about the module name suffixes.

To facilitate simulation, a board level test bench, module BDtest, is included. A Transmit Test Unit (XTU) module and a Receive Test Unit (RTU) module are also included (see Figure 10). The XTU and RTU are effectively placeholders for the user logic, and they are examples of how the user logic should be connected to the SelectLink modules.

To test a SelectLink system, the XTU writes pseudo random data words. Signal SLxWrite is also generated in a pseudo random manner. Similarly, the RTU generates signal SLrRead in a pseudo random manner, and checks the data by comparing it to data generated in the same way that the write data was generated.

To thoroughly verify a SelectLink system, it is desirable to alternately create both Full and Empty conditions. This is done with two asynchronous "sideband" signals, XmitFull and RcvEmpty, which are used to speed up or slow down the write and read rates to reverse the current Full or Empty condition.

Simulation of the generated code is recommended before replacing the XTU and RTU modules with user defined test units.

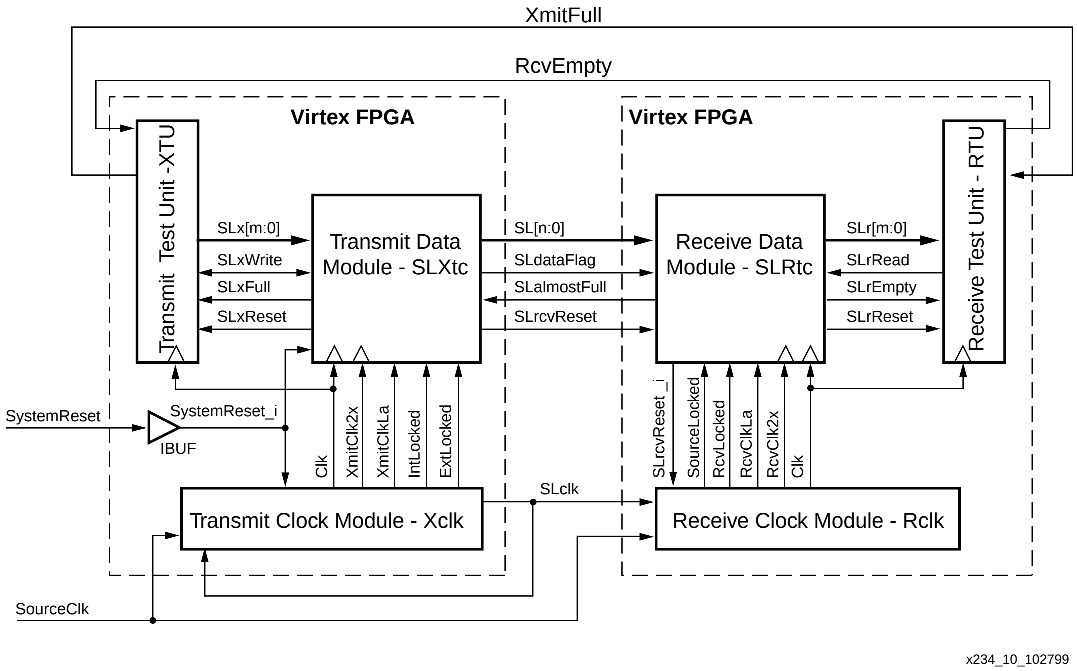


Figure 10: **SelectLink Top-Level Modules and Interconnects With Test Units**

Bus Widths That Are Not Powers of Two

- If the bus width is not a power of two, a large number of block RAM modules may be needed to meet the SelectLink configuration rules. For example, an 18/9 internal/external bus width system requires 9 block RAM modules in each FIFO, with a FIFO depth of 2048 words. If this depth is needed, this is a good solution. On the other hand, if fewer block RAM modules are available and the FIFO depth requirements are less, portions of 16/8 and 2/1 configurations in the SLXtc and SLRtc modules can be combined to create an 18/9 system that used only two block RAM modules, and has a FIFO depth of 256 words. The method and rules for combining configurations are:

- The latency of the configurations that are combined must be the same, so the ratio of the internal and external bus widths must be the same.
- Retain all the control logic of the configuration that has the shallowest FIFOs. This becomes the primary configuration. Copy the logic labeled "DATA PATH" from SLXtc and SLRtc of the other configuration(s) into the primary configuration.
- The FIFO full and empty flags must be sourced from the FIFO in the primary configuration.
- Ensure that the XmitClkLa and RcvClkLa daisy chains are connected properly.
- Renumber data bus bits in Xchip, SLXtc, Rchip, and SLRtc to represent contiguous buses.
- Rename instances as necessary to avoid duplicates.

Synthesis, Place, and Route

Xilinx Alliance or Foundation software may be used to synthesized, place, and route a design after it is merged with the SelectLink code. Example Xchip and Rchip constraint files are included with the Verilog source code. These examples include the recommended pin and block RAM location constraints. Internal flip-flops that are connected to input or output pins must be packed into the IOBs. This can be done in the map command with the “-pr b” option.

Conclusion

As system clock speeds continue to rise, transferring data between FPGAs can present a significant engineering challenge. The SelectLink communication channel is a flexible, powerful, and easy to use system designed to meet this challenge.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
12/21/99	1.0	Initial Xilinx release.
3/15/00	1.1	Updated to new format.