Virtex-II SelectLink Communications Channel

XAPP263 (v1.0) July 16, 2002

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 Virtex™-II SelectLink communications channel utilizes special features of the Virtex-II series of FPGAs, including Digital Clock Managers (DCMs), 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 or VHDL source code. The modules are easily instantiated in the designer's top-level code for a complete system solution.

Note: This application also works on Virtex-II Pro™ devices.

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 that require multiple internal cycles for a single transfer. Figure 1 is a high-level SelectLink block diagram.
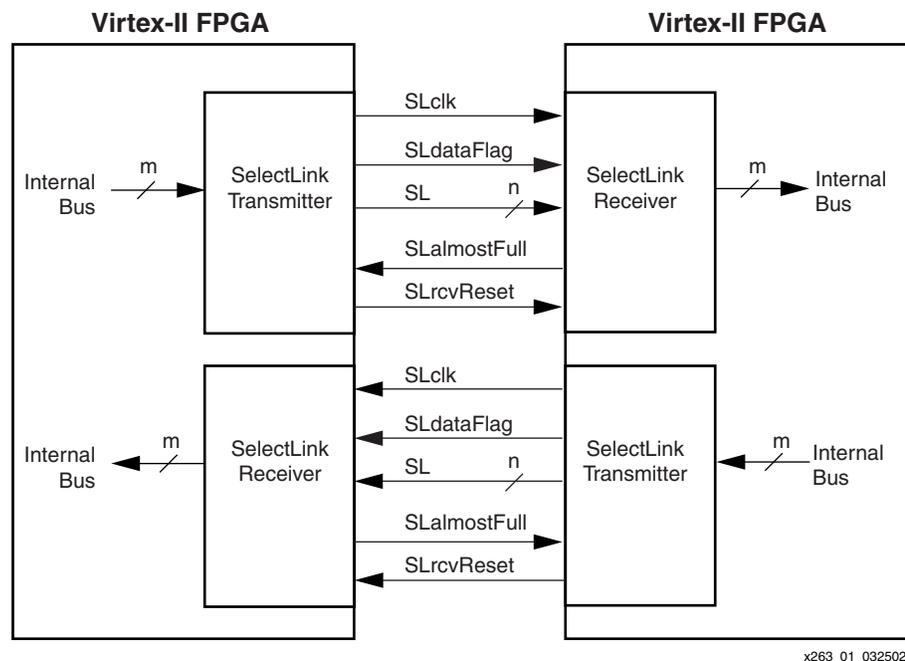
Figure 1: **SelectLink System Level**

The SelectLink channel creates a high-bandwidth FPGA-to-FPGA path using similar techniques, and it does this with resources that are standard on all Virtex-II 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 Virtex-II SelectLink channel has the following notable features.

On-Line Source Code Generation Wizard

The on-line code generator at **www.xilinx.com** employs a "wizard" style user interface to configure a SelectLink system that matches the requirements of the user's application. A series of forms are presented. Each form is customized based on information from the previous forms. The user can navigate forward or backward through the forms. When the process is complete, a block diagram of the configured system is displayed, followed by either Verilog or VHDL source code.

After the customized SelectLink source code is saved as a local file, it can easily be merged with the user's design by making just a few connections at the top hierarchy level.

Selectable Hardware Description Language

Either Verilog or VHDL can be selected as the HDL representation of the SelectLink design.

Selectable Synthesis Tool

Any of the available synthesis tools can be selected during the configuration process. This information is used to correctly configure certain primitive attributes that are, in some cases, synthesis tool dependent.

Selectable Bus Bit Numbering Direction

Bit 0 can be configured as either the least significant or the most significant bit on all internal and external buses.

Performance/Resources Tradeoff

Four performance/resources tradeoff options are available. Note that these tradeoffs can also affect data transfer latency; see **Latency** below.

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. Additionally, Virtex-II SelectLink includes Adaptive Data Slicing (ADS) circuitry. See **Adaptive Data Slicing** below.

Unidirectional or Bidirectional

Data is sent in one direction only on a single SelectLink channel, from one transmitter to one receiver. If bidirectional operation is needed, two channels can be selected when the code generator is configured.

Double Data Rate (DDR)

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

Adaptive Data Slicing

Each time the SelectLink system is reset, the initialization process automatically positions the receiver data clock in the center of the data eye. All signals that carry sampled data (SL[bus],

SLdataFlag, and optional SLpar[bus]) are included in this process. The Variable Phase Shift feature of the receiver DCM is used to position the clock edges.

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. (Since the FIFOs in both the transmitter and receiver use "Almost Full" rather than "Full" flags, the usable FIFO depth may be as low as 7/8 (87.5%) of the total memory depth listed in the configuration menu.)

Data Flow Control

Flow control is built into 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.

Optional Parity Bus

Byte parity can be optionally included for the SelectLink data path. When parity is selected, the internal data bus width (in bits) must be evenly divisible by 8. Parity generation and checking is done by the user. (Actually, this is just one additional bit path between the transmitter and receiver for each 8 bits in the normal data path. These extra bit paths can be used for any purpose, not just parity.)

Data Funneling and Expansion

Because data is transferred at twice the clock rate, the external SelectLink bus is never 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. The width of the external data bus affects latency. See **Latency** below.

Independent User Clocks

The user clocks in the transmitter and receiver FPGAs are independent of the SelectLink bus clock and each other. If this flexibility is not needed, the internal SelectLink clock buffer can also drive the user logic in the transmitter and/or the receiver.

System-Level Overview

The SelectLink bus is especially suited to applications that stream large amounts of data from one FPGA to another. Because of latency (i.e., 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. The optional parity buses are not shown in this figure.
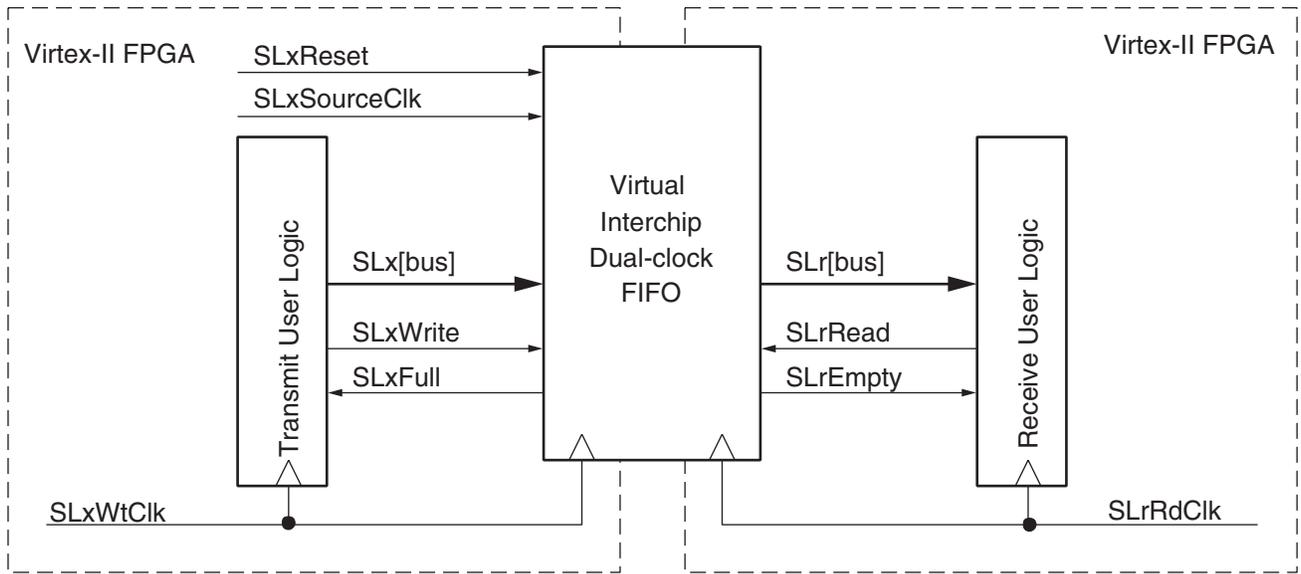
x263_02_032502

Figure 2: **SelectLink Conceptual Diagram**

External Bus

Table 1 includes all of the external SelectLink signals that connect the transmitter FPGA to the receiver FPGA. All of the signal names begin with "SL". When a bidirectional system is configured, the code generator inserts a number right after "SL" in the signal names. A 0 is inserted in transmitter signal names and a 1 is inserted in receiver signal names. The numbers make the signal names unique, which is a requirement when the code is synthesized. If more than one transmitter or receiver is used on a single chip, the 0 can be manually replaced with successive even numbers (2, 4,...), and the 1 can be replaced with successive odd numbers (3, 5,...) for the additional channels.

Table 1: **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. 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.
SL[bus]	X → R	SL is the double data rate bus. Data switches on the rising and falling edge of SLclk. Each datum on SL is referred to as a parcel. The width of this bus is configurable.
Slpar[bus] (optional)	X → R	This optional parity bus carries the DDR parity bits associated with the DDR bytes of SL. The width of Slpar is the width of SL divided by 8, and rounded up to the next integer, if it is not an integer.

Table 1: **SelectLink External Interface Signals** *(Continued)*

Signal Name	Direction	Description
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 sends 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 DCM that generates SLclk is Locked. The purpose of this signal is to hold the receiver DCM in reset until signal SLclk is stable.

Figure 3 shows the relative timing of SLclk, SLdataFlag, and SL[bus]. 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."

x263_03_032502
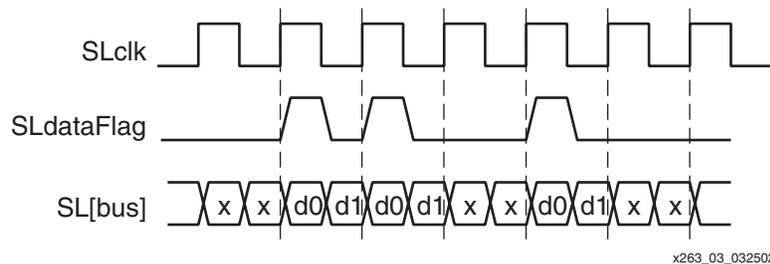
Figure 3: **External Timing**

The SelectLink feature is not limited to a particular electrical interface (physical layer). The code generation engine available at **www.xilinx.com** (in conjunction with constraint information, in some cases), allows the designer to chose any of the SelectI/O standards. (See the *Virtex-II Platform FPGA Users Guide* for information on the various I/O standards supported by Virtex-II 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 moderate data rates (to achieve the maximum data rate, a differential standard is needed). For transmission between PCBs, the LVDS differential standard is recommended, because it provides the best noise immunity.

Whatever physical layer is used, the propagation times from the transmitter output pin to the receiver input pin must match each other for the following signals (with a maximum skew tolerance of 10% of SLclk period):

        SLdataFlag   SL[bus]   Slpar[bus]

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 can 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 FPGA, the SelectLink channel is implemented in Verilog modules or VHDL entities as follows:

SLXRlc - Tranceiver; present when bidirectional operation is selected

- SLXlc - Main Transmitter Module; present when bidirectional or transmitter only selected.

- SLRlc - Main Receiver Module; present when bidirectional or receiver only selected.

There are a number of submodules in the hierarchy below SLXlc and SLRlc. The exact module set is determined by the configuration that is selected.

Internal FIFO Buses

The various signals in the internal transmit and receive interfaces are defined in Table 2 and Table 3, respectively. In the Direction columns, "SL" is the SelectLink channel and "UL" means User Logic.

Table 2: **SelectLink Internal User Interface Transmit Signals**

Signal Name	Direction	Description
SLxReset	UL→ SL	This signal is typically driven by the system reset. It resets the entire SelectLink channel, including the receiver (via external signal SLrcvReset).
SLxSourceClk	UL → SL	This signal feeds the DCM that is the source for SLclk. (If an FPGA contains multiple SelectLink transmitters, the same DCM and BUFG can be used for all of them.)
SLxWtClk	UL → SL	This is the user write clock. It must be driven by a global clock driver (e.g., BUFG). This clock is the timing reference for all the following signals in this table.
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 SLxWtClk, provided this signal is false. When SlxReset is activated, SlxFull goes High until the channel has been initialized and is ready for use.
SLx[bus]	UL → SL	This is the transmit FIFO write data bus. When SLxWrite is active, SLx[bus] must be valid and setup to the high-going edge of SLxWtClk. The width of this bus is configurable. If parity is selected, the width (in bits) must be evenly divisible by 8.
SLxPar[bus]	UL → SL	This optional parity bus has the same timing requirements as SLx[bus]. The width of this bus is equal to the width of SLx[bus] divided by 8.
SLxWrite	UL → SL	This signal indicates that the user wants to write SLx[bus] to the SelectLink receiver on the next high-going edge of Clk. SLxWrite must be setup to the high-going edge of SLxWtClk. This signal should only be activated when SLxFull is false. It will have no effect if it is activated when SLxFull is true.

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

Signal Name	Direction	Description
SLrRdClk	UL → SL	This is the user read clock. It must be driven by a global clock driver (e.g., BUFG). This clock is the timing reference for all the following signals in this table.
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 SLrRdClk, provided this signal is false. This signal goes true when SelectLink is reset, and it stays true until initialization is complete, and the first datum is available in the receiver FIFO.
SLr[bus]	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 SLrRdClk. It will remain on this bus until SLrRead is activated again. The width of this bus is the same as SLx[bus].
SLrPar[bus]	SL → UL	This optional parity bus has the same timing requirements as SLr[bus]. The width of this bus is the same as SLxPar[bus].
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 SLrRdClk. SLrRead must be setup to the high-going edge of SLrRdClk. This signal should only be activated when SLrEmpty is false. It will have no effect if it is activated when SLrEmpty is true.

Figure 4 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.

x263_04_032502
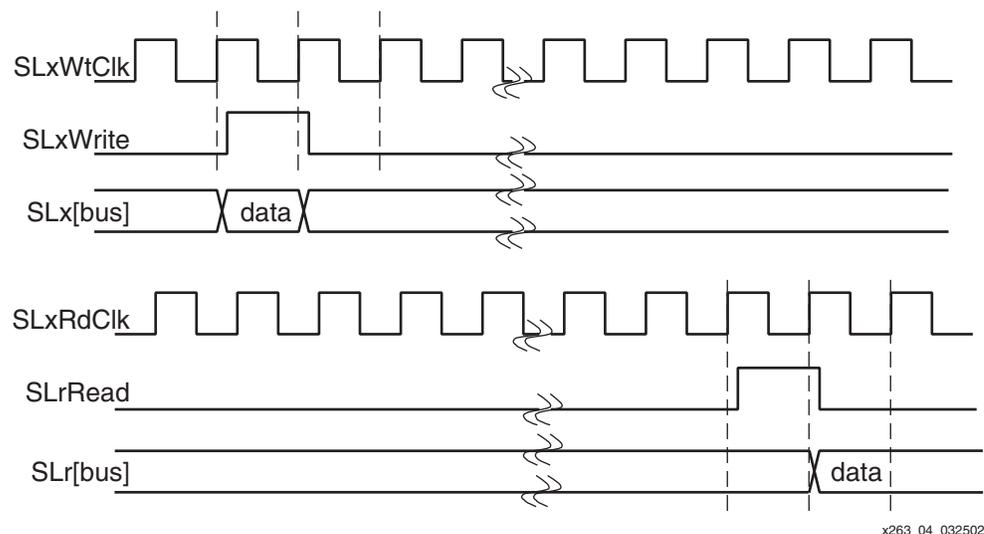
Figure 4: **User Interface Write and Read Timing**

Data Rate

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

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[bus].

Using Table 4 as a starting point, the designer can choose the required bus width and device speed grade. Table 4 lists the maximum SelectLink clock frequency that can be expected for various Virtex-II speed grades, using the highest performance option (3 BUFGs, extra logic) and an LVDS external bus. While exact performance is determined only by implementing a complete design, this table may be used as a guideline.

Table 4: **Maximum SelectLink Clock Frequencies**

Family (Speed Grade)	Maximum f_{SLclk} (MHz)	Mbit/sec/pin
Virtex-II (-4)	255	510
Virtex-II (-5)	300	600
Virtex-II (-6)	335	670

A SelectLink system with the following configuration was verified in hardware:

- Hardware Description Language: Verilog

- Synthesis Tool: Synplicity

- FIFO Parity: Yes

- Bus Bit Numbering: [MSB:0]

- Performance/Resources Tradeoff: 3 BUFGs, extra logic

- I/O Standard: LVDS 3.3V

- Internal Data Bus Width: 16

- External Data Bus Width: 8

- SelectRAM Blocks: 1 each for transmitter and receiver

- FPGA: XC2V1000-FG456-5

This system, which used three independent clock sources for SlxSourceClk, SLxWtClk, and SLrRdClk, ran reliably up to f_{SLclk} = 310 MHz at nominal voltages and room temperature.

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**). 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.

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 following:

1. The ratio of the internal and external bus widths.

2. Whether or not an "extra logic" choice is selected when configuring 'Performance/Resources Tradeoff".

3. The propagation time of the external bus.

4. The instantaneous phase difference between the rising edges of SLxWtClk and SLclk when the datum is written and the instantaneous phase difference between the rising

edges of SLrRdClk and SLclk when the datum is read. This affects latency only if the FIFO Empty flag switches from true to false in either or both of the FIFOs. On average, the latency will increase by 1.5 SLclk periods (3.0 SLclk periods when "extra logic" is selected) when the Transmitter FIFO goes from empty to not empty. On average, the latency will increase by 1.5 SLrRdclk periods when the Receiver FIFO goes from empty to not empty.

Table 5 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 5 for each additional period, or portion thereof. The data in this table was generated with SLxSourceClk = SLxWtClk = SLrRdClk, i.e., the phase difference between the clocks is fixed.

Table 5: **Latency as a Function of Bus Width Ratio and "Extra Logic"**

Internal Width/ External Width	Latency in SLclk Periods Without "Extra Logic"	Latency in SLclk Periods With "Extra Logic"
2/1	8	16
4/1	9	16
8/1	11	18
16/1	15	22
32/1	23	30
64/1	39	46

Code Generation Engine

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

The interface is in the form of a "wizard," a series of forms that are customized based on the previous entries. "Previous" and "Next" buttons can be used to navigate backward or forward, respectively, through the forms. After all the configuration information is entered, a block diagram of the configured system is displayed. If the block diagram appears correct, the user can click on the "Finish" button to download the custom Verilog or VHDL source code.

Virtex-II SelectLink is an inter-FPGA communication channel, so it requires two or more Virtex-II FPGAs in a system. However, the on-line code generator always creates code for a single chip. If the chip is receiving one channel only, then "Receiver only" would be chosen on the form. Similarly, if the chip is transmitting one channel only, "Transmitter only" would be chosen.

If the chip needs both a transmitter and receiver, then select "Both." Typically, this is because a bidirectional channel is needed to communicate to another Virtex-II FPGA. However, you could chose this option if you are receiving from one Virtex-II and transmitting to another in a system with three FPGAs.

The code generator automatically generates final code for all of the above scenarios. For more complicated architectures, such as two bidirectional channels to communicate with two other FPGAs, some manual code editing is required. See the **External Bus** section for more information on manual code modifications.

Although the on-line code generator always creates code for a single chip, the same code might be applicable to more than one chip. For example, the code that creates a bidirectional channel between two chips can be used in both chips.

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

An alphanumeric suffix that defines configuration is attached to the base module name of each module (or entity, if VHDL) that is generated. The suffix elements are separated by underscores. For example, the suffix in entity name dpRAMv2_h_16_32_2 indicates that this is

a Virtex-II dual-port RAM written in VHDL format, with a 16-bit wide Port A and a 32-bit wide Port B, and it uses two Block SRAM modules. Since the correct modules are automatically generated, the designer generally does not need to be concerned about the module name suffixes.

Synthesis, Place, and Route

Xilinx Alliance, Foundation, or BaseX software can be used to synthesize, place, and route a design after it is merged with the SelectLink code. Internal SelectLink 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.

For best performance, the registers that directly interface to the SelectLink bus IOBs must be placed adjacent to the corresponding IOB in both the transmitter and receiver. This is because the transfer time at this point is limited, in some cases, to only 1/2 the SLclk period. The names of these instances will vary somewhat depending on the HDL and the synthesis tool that is used.

The following UCF constraints are an example from a Verilog design that had an 8-bit wide SL[bus] and was synthesized with XST.

```
# transmitter placement constraints
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_1   LOC=SLICE_X0Y17;
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_3   LOC=SLICE_X0Y19;
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_5   LOC=SLICE_X0Y20;
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_7   LOC=SLICE_X0Y21;
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_9   LOC=SLICE_X0Y23;
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_11  LOC=SLICE_X0Y24;
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_13  LOC=SLICE_X0Y25;
INST  SLTU_xcvr_xmtr_FIFOdataOutPL_15  LOC=SLICE_X0Y27;
INST  SLTU_xcvr_xmtr_DataFlagPL_1      LOC=SLICE_X0Y31;

# receiver placement constraints
INST  SLTU_xcvr_rcvr_DataSDR_1         LOC=SLICE_X111Y36;
INST  SLTU_xcvr_rcvr_DataSDR_3         LOC=SLICE_X111Y35;
INST  SLTU_xcvr_rcvr_DataSDR_5         LOC=SLICE_X111Y33;
INST  SLTU_xcvr_rcvr_DataSDR_7         LOC=SLICE_X111Y32;
INST  SLTU_xcvr_rcvr_DataSDR_9         LOC=SLICE_X111Y31;
INST  SLTU_xcvr_rcvr_DataSDR_11        LOC=SLICE_X111Y29;
INST  SLTU_xcvr_rcvr_DataSDR_13        LOC=SLICE_X111Y28;
INST  SLTU_xcvr_rcvr_DataSDR_15        LOC=SLICE_X111Y27;
INST  SLTU_xcvr_rcvr_SLdataFlagSDR_1   LOC=SLICE_X111Y23;
```

Conclusion

As system clock speeds continue to rise, transferring data between FPGAs can present a significant engineering challenge. The Virtex-II 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
07/16/02	1.0	Initial Xilinx Release