

Virtex-II Pro Array Functional Description

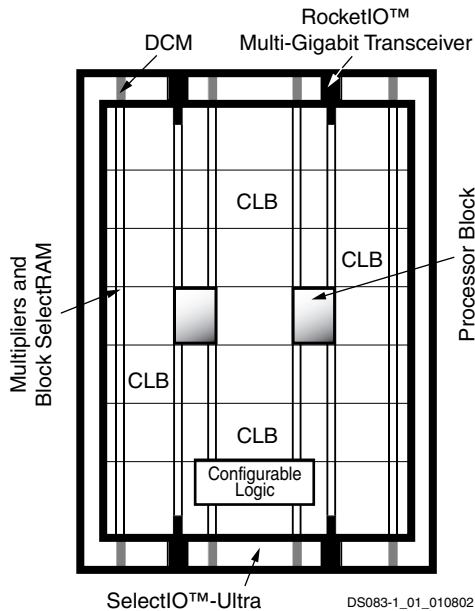


Figure 1: Virtex-II Pro Generic Architecture Overview

This module describes the following Virtex-II Pro functional components, as shown in Figure 1:

- Embedded RocketIO™ Multi-Gigabit Transceiver (MGT)
- Processor block containing embedded IBM® PowerPC™ 405 RISC CPU (PPC405) core and integration circuitry.
- FPGA fabric based on Virtex-II architecture.

For a description of PPC405 embedded core programming models and internal core operations, refer to the [PPC405 User Manual](#) and the [PPC405 Processor Block Manual](#). For detailed RocketIO transceiver digital/analog design considerations, refer to [RocketIO Transceiver User Guide](#).

All of the documents above, as well as a complete listing and description of Xilinx-developed Intellectual Property cores for Virtex-II Pro, are available on the Xilinx website at www.xilinx.com/virtex2pro.

Virtex-II Pro Compared to Virtex-II Devices

Virtex-II Pro devices are built on the Virtex-II FPGA architecture. Most FPGA features are identical to Virtex-II devices. Differences are described below:

- The Virtex-II Pro FPGA family is the first to incorporate embedded PPC405 cores and RocketIO MGTs.
- V_{CCAUX} , the auxiliary supply voltage, is 2.5V instead of 3.3V as for Virtex-II devices. Advanced processing at 0.13 μ m has resulted in a smaller die, faster speed, and lower power consumption.
- Virtex-II Pro devices are neither bitstream-compatible nor pin-compatible with Virtex-II devices. However, Virtex-II designs can be compiled into Virtex-II Pro devices.
- SSTL3, AGP-2X/AGP, LVPECL_33, LVDS_33, and LVDS_33 standards are not supported.
- The open-drain output pin TDO does not have an internal pullup resistor.

Functional Description: RocketIO Multi-Gigabit Transceiver (MGT)

This section summarizes the features of the RocketIO multi-gigabit transceiver. For an in-depth discussion of the RocketIO MGT, including digital and analog design considerations, refer to the [RocketIO Transceiver User Guide](#).

Overview

The embedded RocketIO multi-gigabit transceiver is based on Mindspeed's SkyRail™ technology. Up to twenty-four transceivers are available. The transceiver is designed to operate at any baud rate in the range of 622 Mb/s to 3.125 Gb/s per channel. This includes specific baud rates used by various standards as listed in Table 1.

Table 1: Protocols Supported by RocketIO Transceiver

Protocol	Channels (Lanes)	I/O Baud Rate (Gb/s)	Reference Clock Rate (MHz)
Fibre Channel	1	1.06	53
		2.12	106
		3.1875 ⁽¹⁾	159.375
Gigabit Ethernet	1	1.25	62.5
10Gbit Ethernet	4	3.125	156.25
Infiniband	1, 4, 12	2.5	125
Aurora	1, 2, 3, 4, ...	0.840 – 3.125	42.00 – 156.25
Custom Protocol	1, 2, 3, 4, ...	up to 3.125	up to 156.25

Notes:

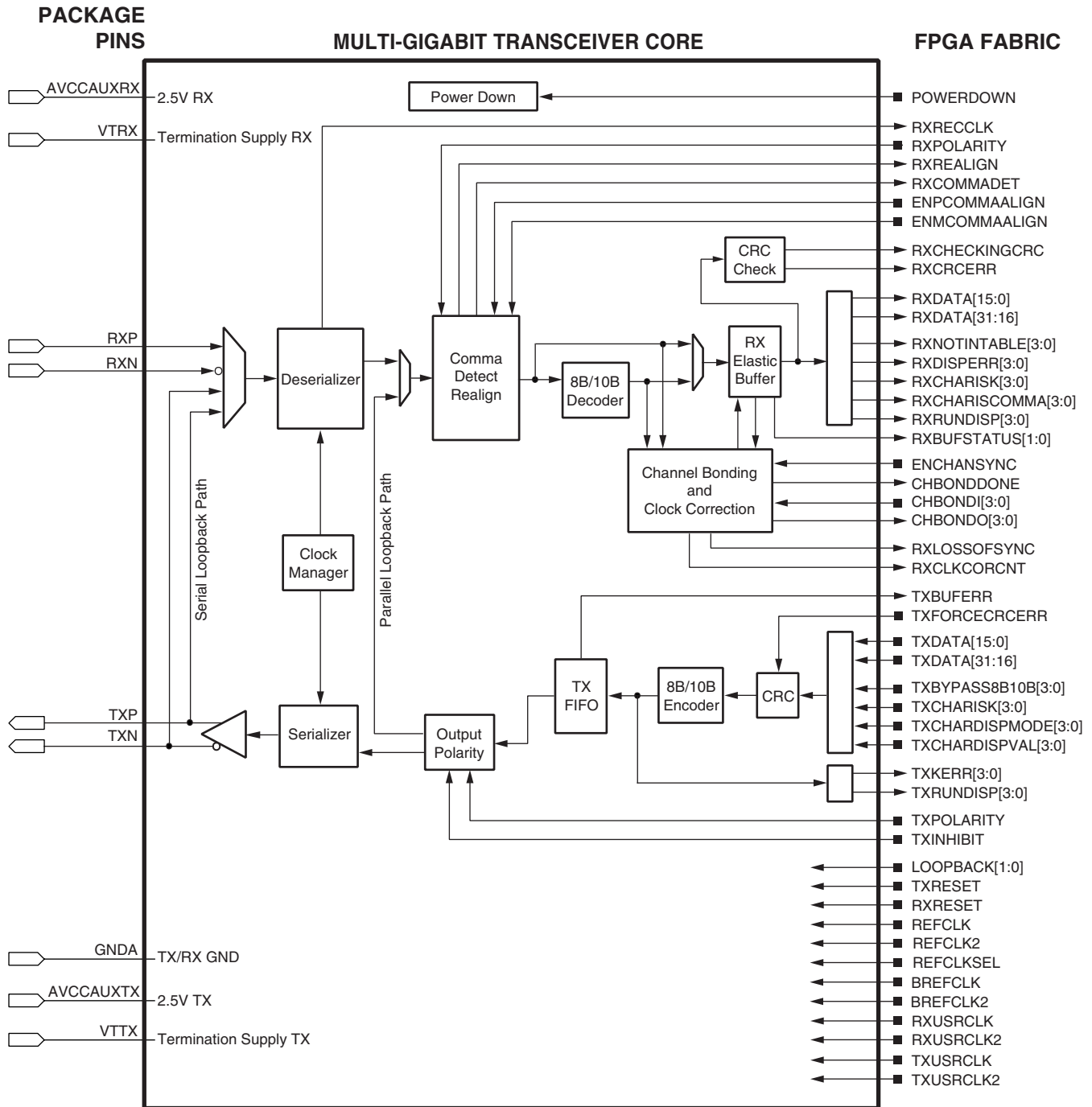
1. Virtex-II Pro MGT can support the 10G Fibre Channel data rates of 3.1875Gb/s across 6" of standard FR-4 PCB and one connector (Molex 74441 or equivalent) with a bit error rate of 10^{-12} or better.

The serial bit rate need not be configured in the transceiver, as the operating frequency is implied by the received data and reference clock applied.

The RocketIO transceiver consists of the Physical Media Attachment (PMA) and Physical Coding Sublayer (PCS). The PMA contains the serializer and deserializer. The PCS contains the bypassable 8B/10B encoder/ decoder, elastic

buffers, and Cyclic Redundancy Check (CRC) units. The encoder and decoder handle the 8B/10B coding scheme. The elastic buffers support the clock correction (rate matching) and channel bonding features. The CRC units perform CRC generation and checking.

Figure 2 shows a high-level block diagram of the RocketIO transceiver and its FPGA interface signals.



DS083-2_04_090402

Figure 2: RocketIO Transceiver Block Diagram

Clock Synthesizer

Synchronous serial data reception is facilitated by a clock/data recovery circuit. This circuit uses a fully monolithic Phase Lock Loop (PLL), which does not require any external components. The clock/data recovery circuit extracts both phase and frequency from the incoming data stream. The recovered clock is presented on output RXRECCLK at 1/20 of the serial received data rate.

The gigabit transceiver multiplies the reference frequency provided on the reference clock input (REFCLK) by 20. The multiplication of the clock is achieved by using a fully monolithic PLL that does not require any external components.

No fixed phase relationship is assumed between REFCLK, RXRECCLK, and/or any other clock that is not tied to either of these clocks. When the 4-byte or 1-byte receiver data path is used, RXUSRCLK and RXUSRCLK2 have different frequencies, and each edge of the slower clock is aligned to a falling edge of the faster clock. The same relationships apply to TXUSRCLK and TXUSRCLK2.

Clock and Data Recovery

The clock/data recovery (CDR) circuits will lock to the reference clock automatically if the data is not present. For proper operation, the frequency of the reference clock must be within ± 100 ppm of the nominal frequency.

It is critical to keep power supply noise low in order to minimize common and differential noise modes into the clock/data recovery circuitry. Refer to the *RocketIO Transceiver User Guide* for more details.

Transmitter

FPGA Transmit Interface

The FPGA can send either one, two, or four characters of data to the transmitter. Each character can be either 8 bits or 10 bits wide. If 8-bit data is applied, the additional inputs become control signals for the 8B/10B encoder. When the 8B/10B encoder is bypassed, the 10-bit character order is generated as follows:

```
TXCHARDISPMODE[0]    (first bit transmitted)
TXCHARDISPVAL[0]
TXDATA[7:0]          (last bit transmitted is TXDATA[0])
```

8B/10B Encoder

A bypassable 8B/10B encoder is included. The encoder uses the same 256 data characters and 12 control characters that are used for Gigabit Ethernet, Fibre Channel, and InfiniBand.

The encoder accepts 8 bits of data along with a K-character signal for a total of 9 bits per character applied, and generates a 10 bit character for transmission. If the K-character signal is High, the data is encoded into one of the twelve possible K-characters available in the 8B/10B code. If the K-character input is Low, the 8 bits are encoded

as standard data. If the K-character input is High, and a user applies other than one of the twelve possible combinations, TXKERR indicates the error.

Disparity Control

The 8B/10B encoder is initialized with a negative running disparity. Unique control allows forcing the current running disparity state.

TXRUNDISP signals its current running disparity. This may be useful in those cases where there is a need to manipulate the initial running disparity value.

Bits TXCHARDISPMODE and TXCHARDISPVAL control the generation of running disparity before each byte.

For example, the transceiver can generate the sequence

```
K28.5+ K28.5+ K28.5- K28.5-
```

or

```
K28.5- K28.5- K28.5+ K28.5+
```

by specifying inverted running disparity for the second and fourth bytes.

Transmit FIFO

Proper operation of the circuit is only possible if the FPGA clock (TXUSRCLK) is frequency-locked to the reference clock (REFCLK). Phase variations up to one clock cycle are allowable. The FIFO has a depth of four. Overflow or underflow conditions are detected and signaled at the interface. Bypassing of this FIFO is programmable.

Serializer

The multi-gigabit transceiver multiplies the reference frequency provided on the reference clock input (REFCLK) by 20. Clock multiplication is achieved by using a fully monolithic PLL requiring no external components. Data is converted from parallel to serial format and transmitted on the TXP and TXN differential outputs.

The electrical connection of TXP and TXN can be interchanged through configuration. This option can be controlled by an input (TXPOLARITY) at the FPGA transmitter interface. This facilitates recovery from situations where printed circuit board traces have been reversed.

Transmit Termination

On-chip termination is provided at the transmitter, eliminating the need for external termination. Programmable options exist for 50 Ω (default) and 75 Ω termination.

Pre-Emphasis Circuit and Swing Control

Four selectable levels of pre-emphasis (10% [default], 20%, 25%, and 33%) are available. Optimizing this setting allows the transceiver to drive various distances of PCB or cable at the maximum baud rate.

The programmable output swing control can adjust the differential output level between 400 mV and 800 mV in four increments of 100 mV.

Receiver

Deserializer

The RocketIO transceiver accepts serial differential data on its RXP and RXN inputs. The clock/data recovery circuit extracts the clock and retimes incoming data to this clock. It uses a fully monolithic PLL requiring no external components. The clock/data recovery circuitry extracts both phase and frequency from the incoming data stream. The recovered clock is presented on output RXRECCLK at 1/20 of the received serial data rate.

The receiver is capable of handling either transition-rich 8B/10B streams or scrambled streams, and can withstand a string of up to 75 non-transitioning bits without an error.

Word alignment is dependent on the state of comma detect bits. If comma detect is enabled, the transceiver recognizes up to two 10-bit preprogrammed characters. Upon detection of the character or characters, the comma detect output is driven high and the data is synchronously aligned. If a comma is detected and the data is aligned, no further alignment alteration takes place. If a comma is received and realignment is necessary, the data is realigned and an indication is given at the receiver interface. The realignment indicator is a distinct output.

The transceiver continuously monitors the data for the presence of the 10-bit character(s). Upon each occurrence of a 10-bit character, the data is checked for word alignment. If comma detect is disabled, the data is not aligned to any particular pattern. The programmable option allows a user to align data on comma+, comma–, both, or a unique user-defined and programmed sequence.

The receiver can be configured to reverse the RXP and RXN inputs. This can be useful in the event that printed circuit board traces have been reversed.

Receiver Termination

On-chip termination is provided at the receiver, eliminating the need for external termination. The receiver includes programmable on-chip termination circuitry for 50Ω (default) or 75Ω impedance.

8B/10B Decoder

An optional 8B/10B decoder is included. A programmable option allows the decoder to be bypassed. When the 8B/10B decoder is bypassed, the 10-bit character order is, for example,

```
RXCHARISK[0]           (first bit received)
RXRUNDISP[0]
RXDATA[7:0]           (last bit received is RXDATA[0])
```

The decoder uses the same table that is used for Gigabit Ethernet, Fibre Channel, and InfiniBand. In addition to decoding all data and K-characters, the decoder has several extra features. The decoder separately detects both “disparity errors” and “out-of-band” errors. A disparity error is the reception of 10-bit character that exists within the

8B/10B table but has an incorrect disparity. An out-of-band error is the reception of a 10-bit character that does not exist within the 8B/10B table. It is possible to obtain an out-of-band error without having a disparity error. The proper disparity is always computed for both legal and illegal characters. The current running disparity is available at the RXRUNDISP signal.

The 8B/10B decoder performs a unique operation if out-of-band data is detected. If out-of-band data is detected, the decoder signals the error and passes the illegal 10-bits through and places them on the outputs. This can be used for debugging purposes if desired.

The decoder also signals the reception of one of the 12 valid K-characters. In addition, a programmable comma detect is included. The comma detect signal registers a comma on the receipt of any comma+, comma–, or both. Since the comma is defined as a 7-bit character, this includes several out-of-band characters. Another option allows the decoder to detect only the three defined commas (K28.1, K28.5, and K28.7) as comma+, comma–, or both. In total, there are six possible options, three for valid commas and three for “any comma.”

Note that all bytes (1, 2, or 4) at the RX FPGA interface each have their own individual 8B/10B indicators (K-character, disparity error, out-of-band error, current running disparity, and comma detect).

Loopback

In order to facilitate testing without having the need to either apply patterns or measure data at GHz rates, two programmable loop-back features are available.

One option, serial loopback, places the gigabit transceiver into a state where transmit data is directly fed back to the receiver. An important point to note is that the feedback path is at the output pads of the transmitter. This tests the entirety of the transmitter and receiver.

The second option, parallel loopback, checks the digital circuitry. When parallel loopback is enabled, the serial loopback path is disabled. However, the transmitter outputs remain active, and data can be transmitted. If TXINHIBIT is asserted, TXP is forced to 0 until TXINHIBIT is de-asserted.

Elastic and Transmitter Buffers

Both the transmitter and the receiver include buffers (FIFOs) in the datapath. This section gives the reasons for including the buffers and outlines their operation.

Receiver Buffer

The receiver buffer is required for two reasons:

- *Clock correction* to accommodate the slight difference in frequency between the recovered clock RXRECCLK and the internal FPGA user clock RXUSRCLK
- *Channel bonding* to allow realignment of the input stream to ensure proper alignment of data being read through multiple transceivers

The receiver uses an *elastic buffer*, where "elastic" refers to the ability to modify the read pointer for clock correction and channel bonding.

Clock Correction

Clock RXRECCLK (the recovered clock) reflects the data rate of the incoming data. Clock RXUSRCLK defines the rate at which the FPGA fabric consumes the data. Ideally, these rates are identical. However, since the clocks typically have different sources, one of the clocks will be faster than the other. The receiver buffer accommodates this difference between the clock rates. See **Figure 3**.

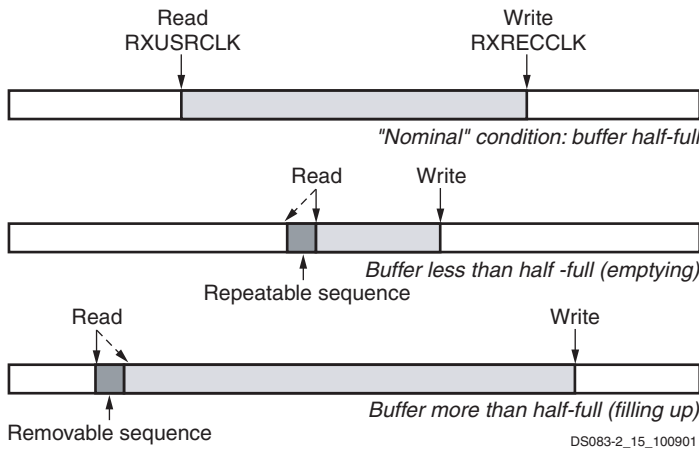


Figure 3: Clock Correction in Receiver

Nominally, the buffer is always half full. This is shown in the top buffer, **Figure 3**, where the shaded area represents buffered data not yet read. Received data is inserted via the write pointer under control of RXRECCLK. The FPGA fabric reads data via the read pointer under control of RXUSRCLK. The half full/half empty condition of the buffer gives a cushion for the differing clock rates. This operation continues indefinitely, regardless of whether or not "meaningful" data is being received. When there is no meaningful data to be received, the incoming data will consist of IDLE characters or other padding.

If RXUSRCLK is faster than RXRECCLK, the buffer becomes more empty over time. The clock correction logic corrects for this by decrementing the read pointer to reread a repeatable byte sequence. This is shown in the middle buffer, **Figure 3**, where the solid read pointer decrements to the value represented by the dashed pointer. By decrementing the read pointer instead of incrementing it in the usual fashion, the buffer is partially refilled. The transceiver design will repeat a single repeatable byte sequence when necessary to refill a buffer. If the byte sequence length is greater than one, and if attribute CLK_COR_REPEAT_WAIT is 0, then the transceiver may repeat the same sequence multiple times until the buffer is refilled to the desired extent.

Similarly, if RXUSRCLK is slower than RXRECCLK, the buffer will fill up over time. The clock correction logic corrects for this by incrementing the read pointer to skip over a

removable byte sequence that need not appear in the final FPGA fabric byte stream. This is shown in the bottom buffer, **Figure 3**, where the solid read pointer increments to the value represented by the dashed pointer. This accelerates the emptying of the buffer, preventing its overflow. The transceiver design will skip a single byte sequence when necessary to partially empty a buffer. If attribute CLK_COR_REPEAT_WAIT is 0, the transceiver may also skip two consecutive removable byte sequences in one step to further empty the buffer when necessary.

These operations require the clock correction logic to recognize a byte sequence that can be freely repeated or omitted in the incoming data stream. This sequence is generally an IDLE sequence, or other sequence comprised of special values that occur in the gaps separating packets of meaningful data. These gaps are required to occur sufficiently often to facilitate the timely execution of clock correction.

Channel Bonding

Some gigabit I/O standards such as Infiniband specify the use of multiple transceivers in parallel for even higher data rates. Words of data are split into bytes, with each byte sent over a separate channel (transceiver). See **Figure 4**.

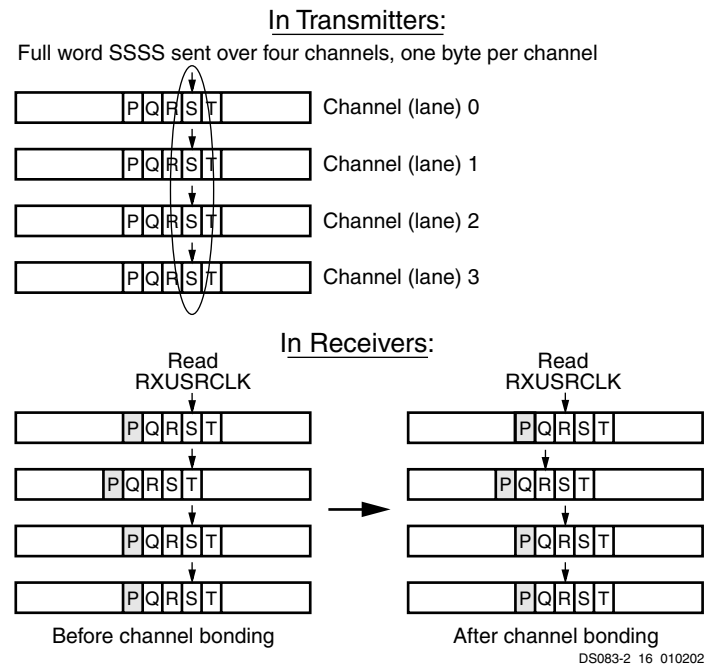


Figure 4: Channel Bonding (Alignment)

The top half of the figure shows the transmission of words split across four transceivers (channels or lanes). PPPP, QQQQ, RRRR, SSSS, and TTTT represent words sent over the four channels.

The bottom-left portion of **Figure 4** shows the initial situation in the FPGA's receivers at the other end of the channels. Due to variations in transmission delay—especially if the channels are routed through repeaters—the FPGA fabric might not correctly assemble the bytes into complete

words. The bottom-left illustration shows the incorrect assembly of data words PQPP, QRQQ, RSRR, and so forth.

To support correction of this misalignment, the data stream includes special byte sequences that define corresponding points in the several channels. In the bottom half of [Figure 4](#), the shaded "P" bytes represent these special characters. Each receiver recognizes the "P" channel bonding character, and remembers its location in the buffer. At some point, one transceiver designated as the master instructs all the transceivers to align to the channel bonding character "P" (or to some location relative to the channel bonding character).

After this operation, words transmitted to the FPGA fabric are properly aligned: RRRR, SSSS, TTTT, and so forth, as shown in the bottom-right portion of [Figure 4](#). To ensure that the channels remain properly aligned following the channel bonding operation, the master transceiver must also control the clock correction operations described in the previous section for all channel-bonded transceivers.

Transmitter Buffer

The transmitter's buffer write pointer (TXUSRCLK) is frequency-locked to its read pointer (REFCLK). Therefore, clock correction and channel bonding are not required. The purpose of the transmitter's buffer is to accommodate a phase difference between TXUSRCLK and REFCLK. A simple FIFO suffices for this purpose. A FIFO depth of four will permit reliable operation with simple detection of overflow or underflow, which could occur if the clocks are not frequency-locked.

CRC

The RocketIO transceiver CRC logic supports the 32-bit invariant CRC calculation used by Infiniband, FibreChannel, and Gigabit Ethernet.

On the transmitter side, the CRC logic recognizes where the CRC bytes should be inserted and replaces four placeholder bytes at the tail of a data packet with the computed CRC. For Gigabit Ethernet and FibreChannel, transmitter CRC may adjust certain trailing bytes to generate the required running disparity at the end of the packet.

On the receiver side, the CRC logic verifies the received CRC value, supporting the same standards as above.

The CRC logic also supports a user mode, with a simple data packet structure beginning and ending with user-defined SOP and EOP characters.

Configuration

This section outlines functions that can be selected or controlled by configuration. Xilinx implementation software supports 16 transceiver primitives, as shown in [Table 2](#).

Each of the primitives in [Table 2](#) defines default values for the configuration attributes, allowing some number of them to be modified by the user. Refer to the *RocketIO Transceiver User Guide* for more details.

Table 2: Supported RocketIO Transceiver Protocol Primitives

GT_CUSTOM	Fully customizable by user
GT_FIBRE_CHAN_1	Fibre Channel, 1-byte data path
GT_FIBRE_CHAN_2	Fibre Channel, 2-byte data path
GT_FIBRE_CHAN_4	Fibre Channel, 4-byte data path
GT_ETHERNET_1	Gigabit Ethernet, 1-byte data path
GT_ETHERNET_2	Gigabit Ethernet, 2-byte data path
GT_ETHERNET_4	Gigabit Ethernet, 4-byte data path
GT_XAUI_1	10-gigabit Ethernet, 1-byte data path
GT_XAUI_2	10-gigabit Ethernet, 2-byte data path
GT_XAUI_4	10-gigabit Ethernet, 4-byte data path
GT_INFINIBAND_1	Infiniband, 1-byte data path
GT_INFINIBAND_2	Infiniband, 2-byte data path
GT_INFINIBAND_4	Infiniband, 4-byte data path
GT_AURORA_1 ⁽¹⁾	1-byte data path
GT_AURORA_2 ⁽¹⁾	2-byte data path
GT_AURORA_4 ⁽¹⁾	4-byte data path

Notes:

1. For more information on the Aurora protocol, visit <http://www.xilinx.com>.

Reset

The receiver and transmitter have their own synchronous reset inputs. The transmitter reset recenters the transmission FIFO, and resets all transmitter registers and the 8B/10B decoder. The receiver reset recenters the receiver elastic buffer, and resets all receiver registers and the 8B/10B encoder. Neither reset has any effect on the PLLs.

Power

All RocketIO transceivers in the FPGA, whether instantiated in the design or not, must be connected to power and ground. Unused transceivers can be powered by any 2.5V source, and passive filtering is not required.

Power Down

The Power Down module is controlled by the transceiver's POWERDOWN input pin. The Power Down pin on the FPGA package has no effect on the transceiver.

Power Sequencing

Although applying power in a random order does not damage the device, it is recommended to apply power in the following sequence to minimize power-on current:

1. Apply FPGA fabric power supplies (V_{CCINT} and V_{CCAUX}) in any order.
2. Apply AVCCAUXRX.
3. Apply AVCCAUTX, V_{TTX} , and V_{TRX} in any order.

Functional Description: Processor Block

This section briefly describes the interfaces and components of the Processor Block. The subsequent section, [Functional Description: Embedded PowerPC 405 Core](#) beginning on [page 9](#), offers a summary of major PPC405 core features. For an in-depth discussion on both the Processor Block and PPC405, see [PPC405 User Manual](#) and [PPC405 Processor Block Manual](#), available on the Xilinx website at <http://www.xilinx.com>.

Processor Block Overview

Figure 5 shows the internal architecture of the Processor Block.

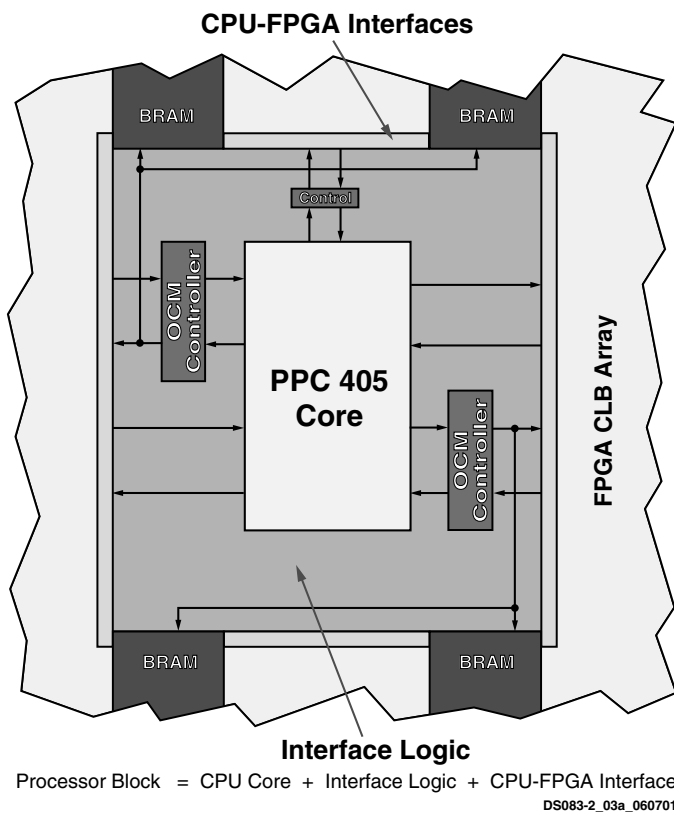


Figure 5: Processor Block Architecture

Within the Virtex-II Pro Processor Block, there are four components:

- Embedded IBM PowerPC 405-D5 RISC CPU core
- On-Chip Memory (OCM) controllers and interfaces
- Clock/control interface logic
- CPU-FPGA Interfaces

Embedded PowerPC 405 RISC Core

The PowerPC 405D5 core is a 0.13 μm implementation of the IBM PowerPC 405D4 core. The advanced process technology enables the embedded PowerPC 405 (PPC405) core to operate at 300+ MHz while maintaining low power

consumption. Specially designed interface logic integrates the core with the surrounding CLBs, block RAMs, and general routing resources. Up to four Processor Blocks can be available in a single Virtex-II Pro device.

The embedded PPC405 core implements the PowerPC User Instruction Set Architecture (UISA), user-level registers, programming model, data types, and addressing modes for 32-bit fixed-point operations. 64-bit operations, auxiliary processor operations, and floating-point operations are trapped and can be emulated in software.

Most of the PPC405 core features are compatible with the specifications for the PowerPC Virtual Environment Architecture (VEA) and Operating Environment Architecture (OEA). They also provide a number of optimizations and extensions to the lower layers of the PowerPC Architecture. The full architecture of the PPC405 is defined by the PowerPC Embedded Environment and PowerPC UISA documentation, available from IBM.

On-Chip Memory (OCM) Controllers

Introduction

The OCM controllers serve as dedicated interfaces between the block RAMs in the FPGA fabric (see [18 Kb Block SelectRAM+ Resources](#), [page 32](#)) and OCM signals available on the embedded PPC405 core. The OCM signals on the PPC405 core are designed to provide very quick access to a fixed amount of instruction and data memory space. The OCM controller provides an interface to both the 64-bit Instruction-Side Block RAM (ISBRAM) and the 32-bit Data-Side Block RAM (DSBRAM). The designer can choose to implement:

- ISBRAM only
- DSBRAM only
- Both ISBRAM and DSBRAM
- No ISBRAM and no DSBRAM

One of OCM's primary advantages is that it guarantees a fixed latency of execution for a higher level of determinism. Additionally, it reduces cache pollution and thrashing, since the cache remains available for caching code from other memory resources.

Typical applications for DSOCM include scratch-pad memory, as well as use of the dual-port feature of block RAM to enable bidirectional data transfer between processor and FPGA. Typical applications for ISOCM include storage of interrupt service routines.

Functional Features

Common Features

- Separate Instruction and Data memory interface between processor core and BRAMs in FPGA
- Dedicated interface to Device Control Register (DCR) bus for ISOCM and DSOCM

- Single-cycle and multi-cycle mode option for I-side and D-side interfaces
- Single cycle = one CPU clock cycle; multi-cycle = minimum of two and maximum of eight CPU clock cycles
- FPGA configurable DCR addresses within DSOCM and ISOCM.
- Independent 16 MB logical memory space available within PPC405 memory map for each of the DSOCM and ISOCM. The number of block RAMs in the device might limit the maximum amount of OCM supported.
- Maximum of 64K and 128K bytes addressable from DSOCM and ISOCM interfaces, respectively, using address outputs from OCM directly without additional decoding logic.

Data-Side OCM (DSOCM)

- 32-bit Data Read bus and 32-bit Data Write bus
- Byte write access to DSBRAM support
- Second port of dual port DSBRAM is available to read/write from an FPGA interface
- 22-bit address to DSBRAM port
- 8-bit DCR Registers: DSCNTL, DSARC
- Three alternatives to write into DSBRAM: BRAM initialization, CPU, FPGA H/W using second port

Instruction-Side OCM (ISOCM)

The ISOCM interface contains a 64-bit read only port, for instruction fetches, and a 32-bit write only port, to initialize or test the ISBRAM. When implementing the read only port, the user must deassert the write port inputs. The preferred method of initializing the ISBRAM is through the configuration bitstream.

- 64-bit Data Read Only bus (two instructions per cycle)
- 32-bit Data Write Only bus (through DCR)
- Separate 21-bit address to ISBRAM
- 8-bit DCR Registers: ISCNTL, ISARC
- 32-bit DCR Registers: ISINIT, ISFILL
- Two alternatives to write into ISBRAM: BRAM initialization, DCR and write instruction

Clock/Control Interface Logic

The clock/control interface logic provides proper initialization and connections for PPC405 clock/power management, resets, PLB cycle control, and OCM interfaces. It also couples user signals between the FPGA fabric and the embedded PPC405 CPU core.

The processor clock connectivity is similar to CLB clock pins. It can connect either to global clock nets or general routing resources. Therefore the processor clock source can come from DCM, CLB, or user package pin.

CPU-FPGA Interfaces

All Processor Block user pins link up with the general FPGA routing resources through the CPU-FPGA interface. Therefore processor signals have the same routability as other non-Processor Block user signals. Longlines and hex lines travel across the Processor Block both vertically and horizontally, allowing signals to route through the Processor Block.

Processor Local Bus (PLB) Interfaces

The PPC405 core accesses high-speed system resources through PLB interfaces on the instruction and data cache controllers. The PLB interfaces provide separate 32-bit address/64-bit data buses for the instruction and data sides.

The cache controllers are both PLB masters. PLB arbiters are implemented in the FPGA fabric and are available as soft IP cores.

Device Control Register (DCR) Bus Interface

The device control register (DCR) bus has 10 bits of address space for components external to the PPC405 core. Using the DCR bus to manage status and configuration registers reduces PLB traffic and improves system integrity. System resources on the DCR bus are protected or isolated from wayward code since the DCR bus is not part of the system memory map.

External Interrupt Controller (EIC) Interface

Two level-sensitive user interrupt pins (critical and non-critical) are available. They can be either driven by user defined logic or Xilinx soft interrupt controller IP core outside the Processor Block.

Clock/Power Management (CPM) Interface

The CPM interface supports several methods of clock distribution and power management. Three modes of operation that reduce power consumption below the normal operational level are available.

Reset Interface

There are three user reset input pins (core, chip, and system) and three user reset output pins for different levels of reset, if required.

Debug Interface

Debugging interfaces on the embedded PPC405 core, consisting of the JTAG and Trace ports, offer access to resources internal to the core and assist in software development. The JTAG port provides basic JTAG chip testing functionality as well as the ability for external debug tools to gain control of the processor for debug purposes. The Trace port furnishes programmers with a mechanism for acquiring instruction execution traces.

The JTAG port complies with IEEE Std 1149.1, which defines a test access port (TAP) and boundary scan architecture. Extensions to the JTAG interface provide

debuggers with processor control that includes stopping, starting, and stepping the PPC405 core. These extensions are compliant with the IEEE 1149.1 specifications for vendor-specific extensions.

The Trace port provides instruction execution trace information to an external trace tool. The PPC405 core is capable of back trace and forward trace. Back trace is the tracing of instructions prior to a debug event while forward trace is the tracing of instructions after a debug event.

The processor JTAG port and the FPGA JTAG port can be accessed independently, or the two can be programmatically linked together and accessed via the dedicated FPGA JTAG pins.

For detailed information on the PPC405 JTAG interface, please refer to the "JTAG Interface" section of the [PPC405 Processor Block Manual](#).

CoreConnect™ Bus Architecture

The Processor Block is compatible with the CoreConnect™ bus architecture. Any CoreConnect compliant cores including Xilinx soft IP can integrate with the Processor Block through this high-performance bus architecture implemented on FPGA fabric.

The CoreConnect architecture provides three buses for interconnecting Processor Blocks, Xilinx soft IP, third party IP, and custom logic, as shown in **Figure 6**:

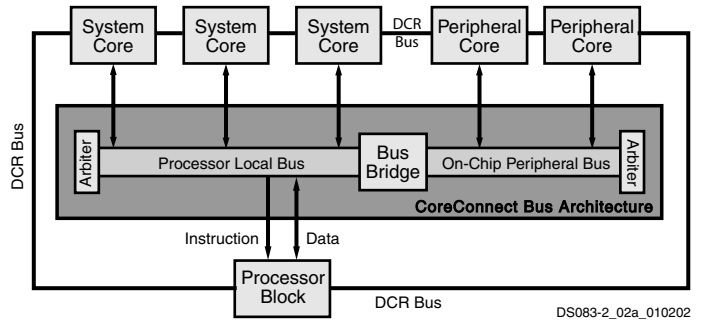


Figure 6: CoreConnect Block Diagram

- Processor Local Bus (PLB)
- On-Chip Peripheral Bus (OPB)
- Device Control Register (DCR) bus

High-performance peripherals connect to the high-bandwidth, low-latency PLB. Slower peripheral cores connect to the OPB, which reduces traffic on the PLB, resulting in greater overall system performance.

For more information, refer to:

http://www-3.ibm.com/chips/techlib/techlib.nfs/product/families/CoreConnect_Bus_Architecture/

Functional Description: Embedded PowerPC 405 Core

This section offers a brief overview of the various functional blocks shown in **Figure 7**.

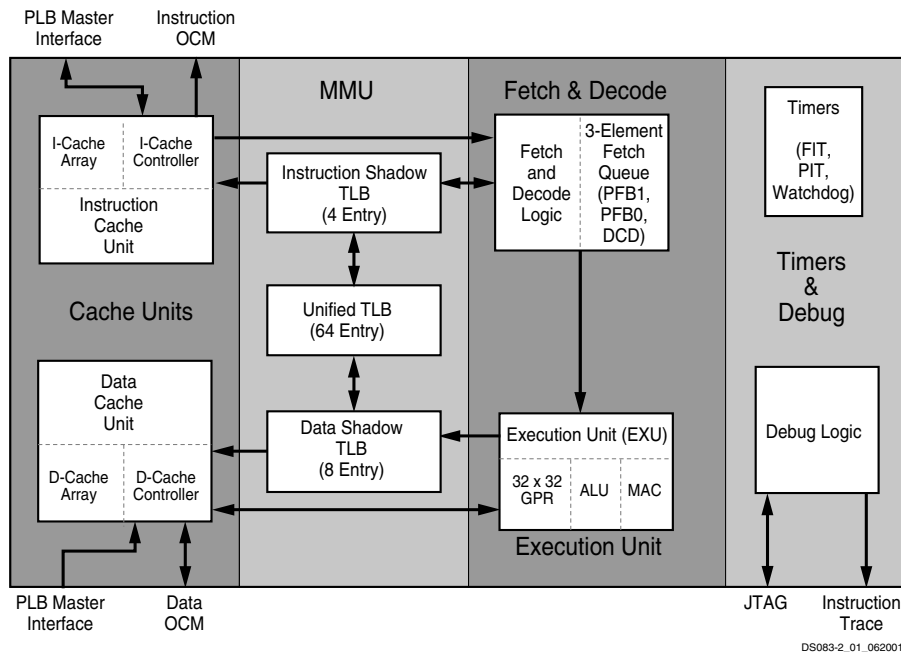


Figure 7: Embedded PPC405 Core Block Diagram

Embedded PPC405 Core

The embedded PPC405 core is a 32-bit Harvard architecture processor. [Figure 7](#) illustrates its functional blocks:

- Cache units
- Memory Management unit
- Fetch Decode unit
- Execution unit
- Timers
- Debug logic unit

It operates on instructions in a five stage pipeline consisting of a fetch, decode, execute, write-back, and load write-back stage. Most instructions execute in a single cycle, including loads and stores.

Instruction and Data Cache

The embedded PPC405 core provides an instruction cache unit (ICU) and a data cache unit (DCU) that allow concurrent accesses and minimize pipeline stalls. The instruction and data cache array are 16 KB each. Both cache units are two-way set associative. Each way is organized into 256 lines of 32 bytes (eight words). The instruction set provides a rich assortment of cache control instructions, including instructions to read tag information and data arrays.

The PPC405 core accesses external memory through the instruction (ICU) and data cache units (DCU). The cache units each include a 64-bit PLB master interface, cache arrays, and a cache controller. The ICU and DCU handle cache misses as requests over the PLB to another PLB device such as an external bus interface unit. Cache hits are handled as single cycle memory accesses to the instruction and data caches.

Instruction Cache Unit (ICU)

The ICU provides one or two instructions per cycle to the instruction queue over a 64-bit bus. A line buffer (built into the output of the array for manufacturing test) enables the ICU to be accessed only once for every four instructions, to reduce power consumption by the array.

The ICU can forward any or all of the four or eight words of a line fill to the EXU to minimize pipeline stalls caused by cache misses. The ICU aborts speculative fetches abandoned by the EXU, eliminating unnecessary line fills and enabling the ICU to handle the next EXU fetch. Aborting abandoned requests also eliminates unnecessary external bus activity, thereby increasing external bus utilization.

Data Cache Unit (DCU)

The DCU transfers one, two, three, four, or eight bytes per cycle, depending on the number of byte enables presented by the CPU. The DCU contains a single-element command and store data queue to reduce pipeline stalls; this queue enables the DCU to independently process load/store and cache control instructions. Dynamic PLB request prioritization reduces pipeline stalls even further. When the DCU is

busy with a low-priority request while a subsequent storage operation requested by the CPU is stalled; the DCU automatically increases the priority of the current request to the PLB.

The DCU provides additional features that allow the programmer to tailor its performance for a given application. The DCU can function in write-back or write-through mode, as controlled by the Data Cache Write-through Register (DCWR) or the Translation Look-aside Buffer (TLB); the cache controller can be tuned for a balance of performance and memory coherency. Write-on-allocate, controlled by the store word on allocate (SWOA) field of the Core Configuration Register 0 (CCR0), can inhibit line fills caused by store misses, to further reduce potential pipeline stalls and unwanted external bus traffic.

Fetch and Decode Logic

The fetch/decode logic maintains a steady flow of instructions to the execution unit by placing up to two instructions in the fetch queue. The fetch queue consists of three buffers: pre-fetch buffer 1 (PFB1), pre-fetch buffer 0 (PFB0), and decode (DCD). The fetch logic ensures that instructions proceed directly to decode when the queue is empty.

Static branch prediction as implemented on the PPC405 core takes advantage of some standard statistical properties of code. Branches with negative address displacement are by default assumed taken. Branches that do not test the condition or count registers are also predicted as taken. The PPC405 core bases branch prediction upon these default conditions when a branch is not resolved and speculatively fetches along the predicted path. The default prediction can be overridden by software at assembly or compile time.

Branches are examined in the decode and pre-fetch buffer 0 fetch queue stages. Two branch instructions can be handled simultaneously. If the branch in decode is not taken, the fetch logic fetches along the predicted path of the branch instruction in pre-fetch buffer 0. If the branch in decode is taken, the fetch logic ignores the branch instruction in pre-fetch buffer 0.

Execution Unit

The embedded PPC405 core has a single issue execution unit (EXU) containing the register file, arithmetic logic unit (ALU), and the multiply-accumulate (MAC) unit. The execution unit performs all 32-bit PowerPC integer instructions in hardware.

The register file is comprised of thirty-two 32-bit general purpose registers (GPR), which are accessed with three read ports and two write ports. During the decode stage, data is read out of the GPRs and fed to the execution unit. Likewise, during the write-back stage, results are written to the GPR. The use of the five ports on the register file enables either a load or a store operation to execute in parallel with an ALU operation.

Memory Management Unit (MMU)

The embedded PPC405 core has a 4 GB address space, which is presented as a flat address space.

The MMU provides address translation, protection functions, and storage attribute control for embedded applications. The MMU supports demand-paged virtual memory and other management schemes that require precise control of logical-to-physical address mapping and flexible memory protection. Working with appropriate system-level software, the MMU provides the following functions:

- Translation of the 4 GB effective address space into physical addresses
- Independent enabling of instruction and data translation/protection
- Page-level access control using the translation mechanism
- Software control of page replacement strategy
- Additional control over protection using zones
- Storage attributes for cache policy and speculative memory access control

The MMU can be disabled under software control. If the MMU is not used, the PPC405 core provides other storage control mechanisms.

Translation Look-Aside Buffer (TLB)

The Translation Look-Aside Buffer (TLB) is the hardware resource that controls translation and protection. It consists of 64 entries, each specifying a page to be translated. The TLB is fully associative; a given page entry can be placed anywhere in the TLB. The translation function of the MMU occurs pre-cache. Cache tags and indexing use physical addresses.

Software manages the establishment and replacement of TLB entries. This gives system software significant flexibility in implementing a custom page replacement strategy. For example, to reduce TLB thrashing or translation delays, software can reserve several TLB entries in the TLB for globally accessible static mappings. The instruction set provides several instructions used to manage TLB entries. These instructions are privileged and require the software to be executing in supervisor state. Additional TLB instructions are provided to move TLB entry fields to and from GPRs.

The MMU divides logical storage into pages. Eight page sizes (1 KB, 4 KB, 16 KB, 64 KB, 256 KB, 1 MB, 4 MB, and 16 MB) are simultaneously supported, such that, at any given time, the TLB can contain entries for any combination of page sizes. In order for a logical to physical translation to exist, a valid entry for the page containing the logical address must be in the TLB. Addresses for which no TLB entry exists cause TLB-Miss exceptions.

To improve performance, four instruction-side and eight data-side TLB entries are kept in shadow arrays. The shadow arrays allow single-cycle address translation and

also help to avoid TLB contention between load/store and instruction fetch operations. Hardware manages the replacement and invalidation of shadow-TLB entries; no system software action is required.

Memory Protection

When address translation is enabled, the translation mechanism provides a basic level of protection.

The Zone Protection Register (ZPR) enables the system software to override the TLB access controls. For example, the ZPR provides a way to deny read access to application programs. The ZPR can be used to classify storage by type; access by type can be changed without manipulating individual TLB entries.

The PowerPC Architecture provides WIU0GE (write-back / write-through, cacheability, user-defined 0, guarded, endian) storage attributes that control memory accesses, using bits in the TLB or, when address translation is disabled, storage attribute control registers.

When address translation is enabled, storage attribute control bits in the TLB control the storage attributes associated with the current page. When address translation is disabled, bits in each storage attribute control register control the storage attributes associated with storage regions. Each storage attribute control register contains 32 fields. Each field sets the associated storage attribute for a 128 MB memory region.

Timers

The embedded PPC405 core contains a 64-bit time base and three timers, as shown in [Figure 8](#):

- Programmable Interval Timer (PIT)
- Fixed Interval Timer (FIT)
- Watchdog Timer (WDT)

The time base counter increments either by an internal signal equal to the CPU clock rate or by a separate external timer clock signal. No interrupts are generated when the time base rolls over. The three timers are synchronous with the time base.

The PIT is a 32-bit register that decrements at the same rate as the time base is incremented. The user loads the PIT register with a value to create the desired delay. When the register reaches zero, the timer stops decrementing and generates a PIT interrupt. Optionally, the PIT can be programmed to auto-reload the last value written to the PIT register, after which the PIT continues to decrement.

The FIT generates periodic interrupts based on one of four selectable bits in the time base. When the selected bit changes from 0 to 1, the PPC405 core generates a FIT interrupt.

The WDT provides a periodic critical-class interrupt based on a selected bit in the time base. This interrupt can be used for system error recovery in the event of software or system

lockups. Users may select one of four time periods for the interval and the type of reset generated if the WDT expires twice without an intervening clear from software. If enabled, the watchdog timer generates a reset unless an exception handler updates the WDT status bit before the timer has completed two of the selected timer intervals.

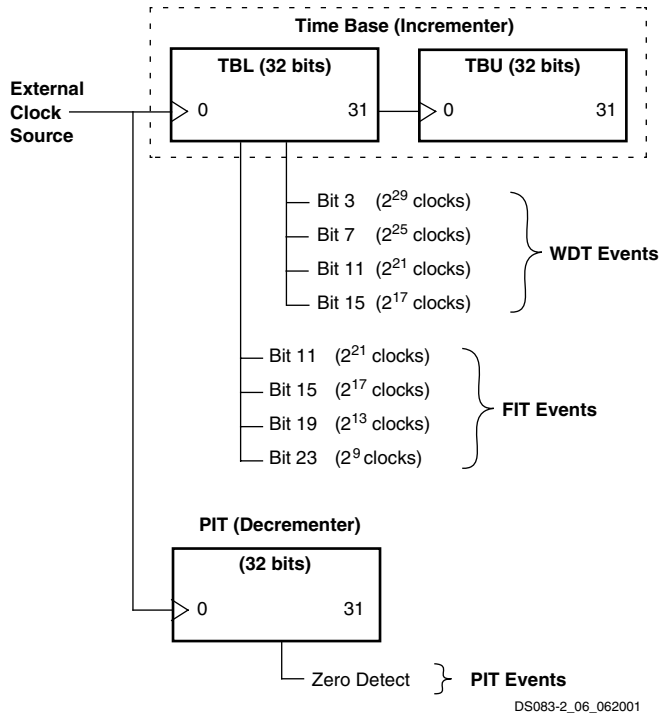


Figure 8: Relationship of Timer Facilities to Base Clock

Interrupts

The PPC405 provides an interface to an interrupt controller that is logically outside the PPC405 core. This controller combines the asynchronous interrupt inputs and presents them to the embedded core as a single interrupt signal. The sources of asynchronous interrupts are external signals, the JTAG/debug unit, and any implemented peripherals.

Debug Logic

All architected resources on the embedded PPC405 core can be accessed through the debug logic. Upon a debug event, the PPC405 core provides debug information to an external debug tool. Three different types of tools are supported depending on the debug mode: ROM monitors, JTAG debuggers, and instruction trace tools.

In internal debug mode, a debug event enables exception-handling software at a dedicated interrupt vector to take over the CPU core and communicate with a debug tool. The debug tool has read-write access to all registers and can set hardware or software breakpoints. ROM monitors typically use the internal debug mode.

In external debug mode, the CPU core enters stop state (stops instruction execution) when a debug event occurs. This mode offers a debug tool read-write access to all registers in the PPC405 core. Once the CPU core is in stop state, the debug tool can start the CPU core, step an instruction, freeze the timers, or set hardware or software breakpoints. In addition to CPU core control, the debug logic is capable of writing instructions into the instruction cache, eliminating the need for external memory during initial board bring-up. Communication to a debug tool using external debug mode is through the JTAG port.

Debug wait mode offers the same functionality as external debug mode with one exception. In debug wait mode, the CPU core goes into wait state instead of stop state after a debug event. Wait state is identical to stop state until an interrupt occurs. In wait state, the PPC405 core can vector to an exception handler, service an interrupt and return to wait state. This mode is particularly useful when debugging real time control systems.

Real-time trace debug mode is always enabled. The debug logic continuously broadcasts instruction trace information to the trace port. When a debug event occurs, the debug logic signals an external debug tool to save instruction trace information before and after the event. The number of instructions traced depends on the trace tool.

Debug events signal the debug logic to stop the CPU core, put the CPU core in debug wait state, cause a debug exception or save instruction trace information.

Big Endian and Little Endian Support

The embedded PPC405 core supports big endian or little endian byte ordering for instructions stored in external memory. Since the PowerPC architecture is big endian internally, the ICU rearranges the instructions stored as little endian into the big endian format. Therefore, the instruction cache always contains instructions in big endian format so that the byte ordering is correct for the execution unit. This feature allows the 405 core to be used in systems designed to function in a little endian environment.

Functional Description: FPGA

Input/Output Blocks (IOBs)

Virtex-II Pro I/O blocks (IOBs) are provided in groups of two or four on the perimeter of each device. Each IOB can be used as input and/or output for single-ended I/Os. Two IOBs can be used as a differential pair. A differential pair is always connected to the same switch matrix, as shown in **Figure 9**. IOB blocks are designed for high-performance I/O, supporting 22 single-ended standards, as well as differential signaling with LVDS, LDT, bus LVDS, and LVPECL.

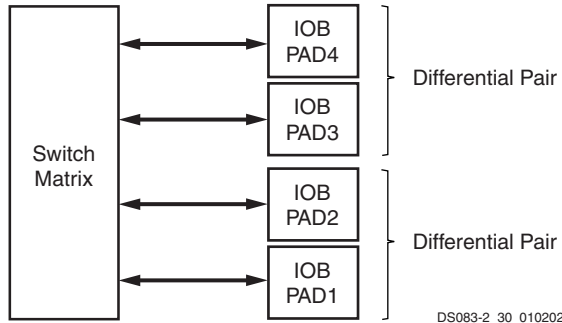


Figure 9: Virtex-II Pro Input/Output Tile

Note: Differential I/Os must use the same clock.

Supported I/O Standards

Virtex-II Pro IOB blocks feature SelectIO-Ultra inputs and outputs that support a wide variety of I/O signaling standards. In addition to the internal supply voltage ($V_{CCINT} = 1.5V$), output driver supply voltage (V_{CCO}) is dependent on the I/O standard (see **Table 3** and **Table 4**). An auxiliary supply voltage ($V_{CCAUX} = 2.5V$) is required, regardless of the I/O standard used. For exact supply voltage absolute maximum ratings, see [Virtex-II Pro™ Platform FPGAs: DC and Switching Characteristics \(Module 3\)](#).

Table 3: Supported Single-Ended I/O Standards

I/O Standard	Output V_{CCO}	Input V_{CCO}	Input V_{REF}	Board Termination Voltage (V_{TT})
LVTTL	3.3	3.3	N/A	N/A
LVC MOS33	3.3	3.3	N/A	N/A
LVC MOS25	2.5	2.5	N/A	N/A
LVC MOS18	1.8	1.8	N/A	N/A
LVC MOS15	1.5	1.5	N/A	N/A
PCI33_3	Note (1)	Note (1)	N/A	N/A
PCI66_3	Note (1)	Note (1)	N/A	N/A
PCI-X	Note (1)	Note (1)	N/A	N/A
GTL	Note (2)	Note (2)	0.8	1.2

Table 3: Supported Single-Ended I/O Standards

I/O Standard	Output V_{CCO}	Input V_{CCO}	Input V_{REF}	Board Termination Voltage (V_{TT})
GTL	Note (2)	Note (2)	1.0	1.5
HSTL_I	1.5	N/A	0.75	0.75
HSTL_II	1.5	N/A	0.75	0.75
HSTL_III	1.5	N/A	0.9	1.5
HSTL_IV	1.5	N/A	0.9	1.5
HSTL_I_18	1.8	N/A	0.9	0.9
HSTL_II_18	1.8	N/A	0.9	0.9
HSTL_III_18	1.8	N/A	1.08	1.8
HSTL_IV_18	1.8	N/A	1.08	1.8
SSTL2_I	2.5	N/A	1.25	1.25
SSTL2_II	2.5	N/A	1.25	1.25
SSTL18_I ⁽³⁾	1.8	N/A	0.9	0.9
SSTL18_II	1.8	N/A	0.9	0.9

Notes:

- For PCI and PCI-X standards, refer to [XAPP653](#).
- V_{CCO} of GTL or GTLP should not be lower than the termination voltage or the voltage seen at the I/O pad.
- SSTL18_I is not a JEDEC-supported standard.

Table 4: Supported Differential Signal I/O Standards

I/O Standard	Output V_{CCO}	Input V_{CCO}	Input V_{REF}	Output V_{OD}
LDT_25	2.5	N/A	N/A	0.500 – 0.740
LVDS_25 ⁽¹⁾	2.5	N/A	N/A	0.250 – 0.400
LVDS EXT_25 ⁽¹⁾	2.5	N/A	N/A	0.330 – 0.700
BLVDS_25	2.5	N/A	N/A	0.250 – 0.450
ULVDS_25	2.5	N/A	N/A	0.500 – 0.740
LVPECL_25	2.5	N/A	N/A	0.345 – 1.185

Notes:

- LVDCI_XX is LVCMOS output controlled impedance buffers, matching all or half of the reference resistors.

All of the user IOBs have fixed-clamp diodes to V_{CCO} and to ground. The IOBs are not compatible or compliant with 5V I/O standards (not 5V-tolerant).

Table 5 lists supported I/O standards with Digitally Controlled Impedance. See **Digitally Controlled Impedance (DCI)**, page 20.

Table 5: Supported DCI I/O Standards

I/O Standard	Output V _{CC0}	Input V _{CC0}	Input V _{REF}	Termination Type
LVDCI_33 ⁽¹⁾	3.3	3.3	N/A	Series
LVDCI_25	2.5	2.5	N/A	Series
LVDCI_DV2_25	2.5	2.5	N/A	Series
LVDCI_18	1.8	1.8	N/A	Series
LVDCI_DV2_18	1.8	1.8	N/A	Series
LVDCI_15	1.5	1.5	N/A	Series
LVDCI_DV2_15	1.5	1.5	N/A	Series
GTL_DCI	1.2	1.2	0.8	Single
GTL_P_DCI	1.5	1.5	1.0	Single
HSTL_I_DCI	1.5	1.5	0.75	Split
HSTL_II_DCI	1.5	1.5	0.75	Split
HSTL_III_DCI	1.5	1.5	0.9	Single
HSTL_IV_DCI	1.5	1.5	0.9	Single
HSTL_I_DCI_18	1.8	1.8	0.9	Split
HSTL_II_DCI_18	1.8	1.8	0.9	Split
HSTL_III_DCI_18	1.8	1.8	1.08	Single
HSTL_IV_DCI_18	1.8	1.8	1.08	Single
SSTL2_I_DCI ⁽²⁾	2.5	2.5	1.25	Split
SSTL2_II_DCI ⁽²⁾	2.5	2.5	1.25	Split
SSTL18_I ⁽³⁾	1.8	1.8	0.9	Split
SSTL18_II	1.8	1.8	0.9	Split
LVDS_25_DCI	N/A	2.5	N/A	Split
LVDS_EXT_25_DCI	N/A	2.5	N/A	Split

Notes:

1. LVDCI_XX is LVCMOS output controlled impedance buffers, matching all or half of the reference resistors.
2. These are SSTL compatible.
3. SSTL18_I is not a JEDEC-supported standard.

Logic Resources

IOB blocks include six storage elements, as shown in Figure 10.

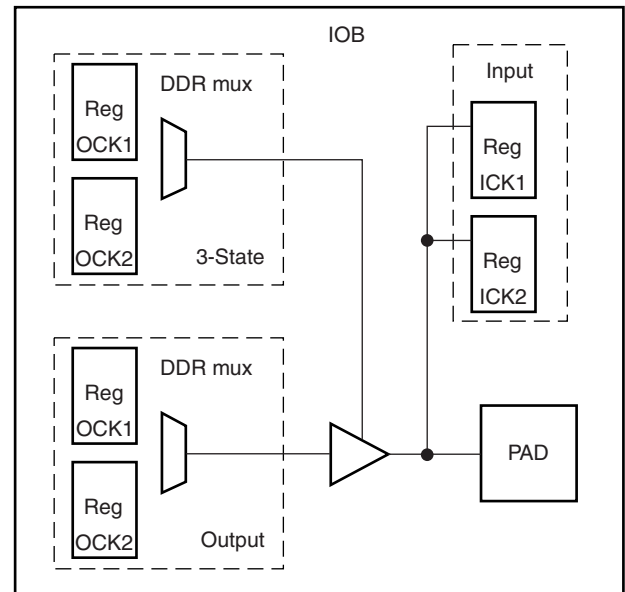
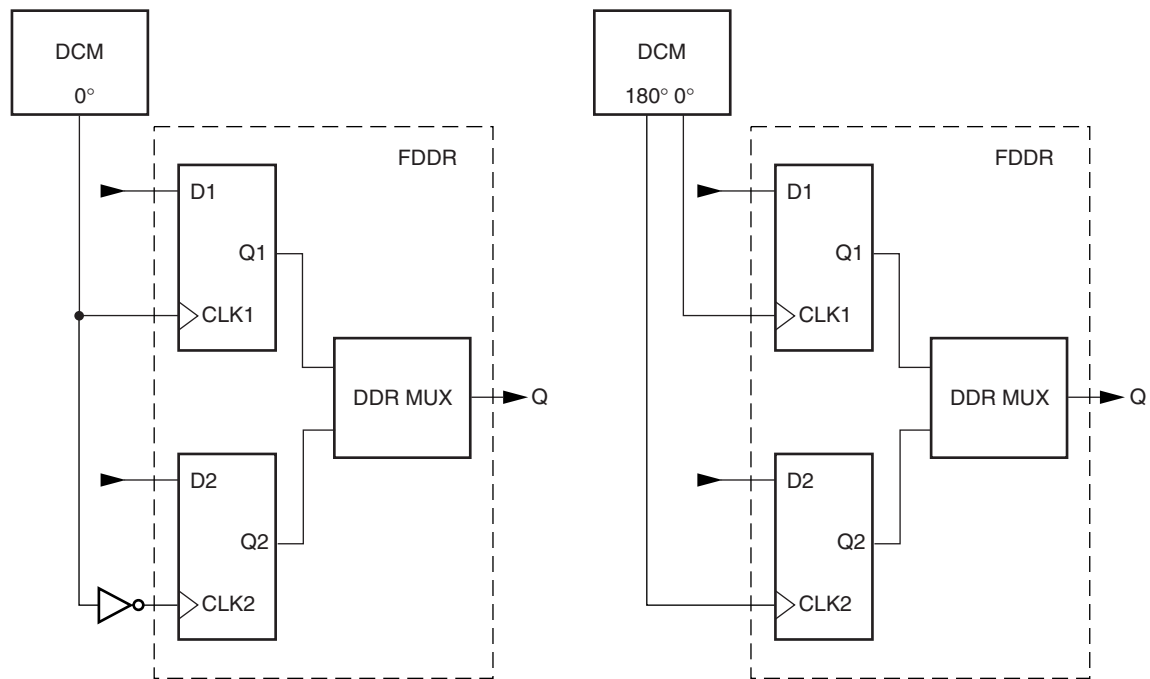


Figure 10: Virtex-II Pro IOB Block

Each storage element can be configured either as an edge-triggered D-type flip-flop or as a level-sensitive latch. On the input, output, and 3-state path, one or two DDR registers can be used.

Double data rate is directly accomplished by the two registers on each path, clocked by the rising edges (or falling edges) from two different clock nets. The two clock signals are generated by the DCM and must be 180 degrees out of phase, as shown in Figure 11. There are two input, output, and 3-state data signals, each being alternately clocked out.



DS083-2_26_122001

Figure 11: Double Data Rate Registers

This DDR mechanism can be used to mirror a copy of the clock on the output. This is useful for propagating a clock along the data that has an identical delay. It is also useful for multiple clock generation, where there is a unique clock driver for every clock load. Virtex-II Pro devices can produce many copies of a clock with very little skew.

Each group of two registers has a clock enable signal (ICE for the input registers, OCE for the output registers, and TCE for the 3-state registers). The clock enable signals are active High by default. If left unconnected, the clock enable for that storage element defaults to the active state.

Each IOB block has common synchronous or asynchronous set and reset (SR and REV signals).

SR forces the storage element into the state specified by the SRHIGH or SRLOW attribute. SRHIGH forces a logic 1. SRLOW forces a logic "0". When SR is used, a second input (REV) forces the storage element into the opposite state. The reset condition predominates over the set condition. The initial state after configuration or global initialization state is defined by a separate INIT0 and INIT1 attribute. By default, the SRLOW attribute forces INIT0, and the SRHIGH attribute forces INIT1.

For each storage element, the SRHIGH, SRLOW, INIT0, and INIT1 attributes are independent. Synchronous or asynchronous set / reset is consistent in an IOB block.

All the control signals have independent polarity. Any inverter placed on a control input is automatically absorbed.

Each register or latch, independent of all other registers or latches, can be configured as follows:

- No set or reset
- Synchronous set
- Synchronous reset
- Synchronous set and reset
- Asynchronous set (preset)
- Asynchronous reset (clear)
- Asynchronous set and reset (preset and clear)

The synchronous reset overrides a set, and an asynchronous clear overrides a preset.

Refer to [Figure 12](#).

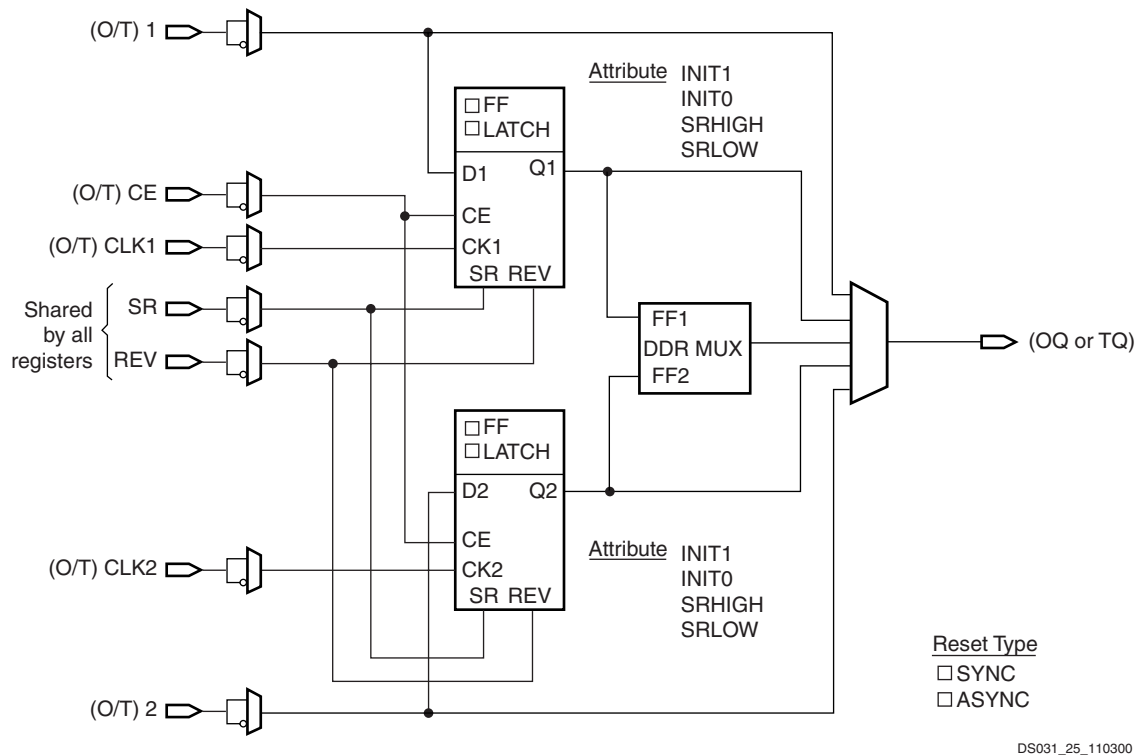


Figure 12: Register / Latch Configuration in an IOB Block

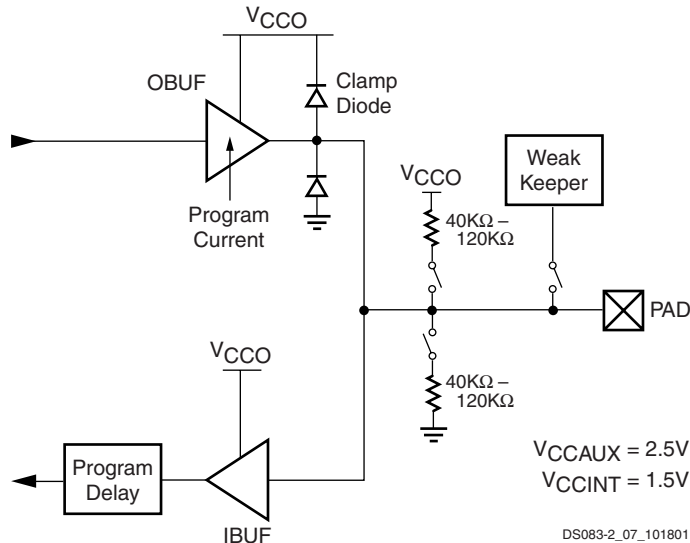


Figure 13: LVTTL, LVCMOS, or PCI SelectIO-Ultra Standard

Input/Output Individual Options

Each device pad has optional pull-up/pull-down resistors and weak-keeper circuit in the LVTTL, LVCMOS, and PCI SelectIO-Ultra configurations, as illustrated in **Figure 13**. Values of the optional pull-up and pull-down resistors fall within a range of 40 KΩ to 120 KΩ when $V_{CCO} = 2.5V$ (from 2.38V to 2.63V only). The clamp diodes are always present, even when power is not.

The optional weak-keeper circuit is connected to each output. When selected, the circuit monitors the voltage on the pad and weakly drives the pin High or Low. If the pin is connected to a multiple-source signal, the weak-keeper holds the signal in its last state if all drivers are disabled. Maintaining a valid logic level in this way eliminates bus chatter. An enabled pull-up or pull-down overrides the weak-keeper circuit.

LVCMOS25 sinks and sources current up to 24 mA. The current is programmable (see **Table 6**). Drive strength and slew rate controls for each output driver minimize bus transients. For LVDCI and LVDCI_DV2 standards, drive strength and slew rate controls are not available.

Table 6: LVCMOS Programmable Currents (Sink and Source)

SelectIO-Ultra	Programmable Current (Worst-Case Guaranteed Minimum)						
LVTTTL	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	24 mA
LVCMOS33	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	24 mA
LVCMOS25	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	24 mA
LVCMOS18	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	n/a
LVCMOS15	2 mA	4 mA	6 mA	8 mA	12 mA	16 mA	n/a

Figure 14 shows the SSTL2, SSTL18, and HSTL configurations. HSTL can sink current up to 48 mA. (HSTL IV)

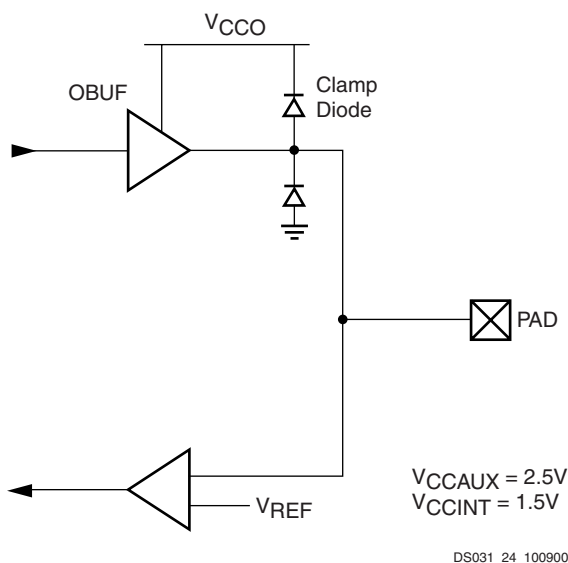


Figure 14: SSTL or HSTL SelectIO-Ultra Standards

All pads are protected against damage from electrostatic discharge (ESD) and from over-voltage transients. Virtex-II Pro uses two memory cells to control the configuration of an I/O as an input. This is to reduce the probability of an I/O configured as an input from flipping to an output when subjected to a single event upset (SEU) in space applications.

Prior to configuration, all outputs not involved in configuration are forced into their high-impedance state. The pull-down resistors and the weak-keeper circuits are inactive. The dedicated pin HSWAP_EN controls the pull-up resistors prior to configuration. By default, HSWAP_EN is set High, which disables the pull-up resistors on user I/O pins. When HSWAP_EN is set Low, the pull-up resistors are activated on user I/O pins.

All Virtex-II Pro IOBs (except RocketIO transceiver pins) support IEEE 1149.1 and IEEE 1532 compatible boundary scan testing.

Input Path

The Virtex-II Pro IOB input path routes input signals directly to internal logic and / or through an optional input flip-flop or latch, or through the DDR input registers. An optional delay element at the D-input of the storage element eliminates pad-to-pad hold time. The delay is matched to the internal clock-distribution delay of the Virtex-II Pro device, and when used, assures that the pad-to-pad hold time is zero.

Each input buffer can be configured to conform to any of the low-voltage signaling standards supported. In some of these standards the input buffer utilizes a user-supplied threshold voltage, V_{REF} . The need to supply V_{REF} imposes constraints on which standards can be used in the same bank. See I/O banking description.

Output Path

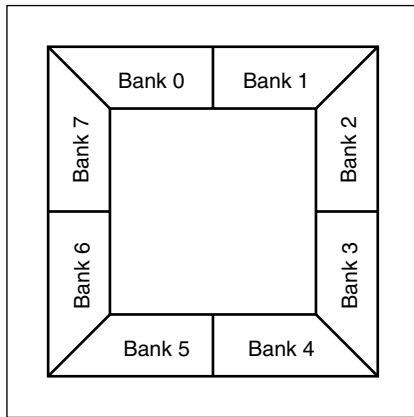
The output path includes a 3-state output buffer that drives the output signal onto the pad. The output and / or the 3-state signal can be routed to the buffer directly from the internal logic or through an output / 3-state flip-flop or latch, or through the DDR output / 3-state registers.

Each output driver can be individually programmed for a wide range of low-voltage signaling standards. In most signaling standards, the output High voltage depends on an externally supplied V_{CCO} voltage. The need to supply V_{CCO} imposes constraints on which standards can be used in the same bank. See I/O banking description.

I/O Banking

Some of the I/O standards described above require V_{CCO} and V_{REF} voltages. These voltages are externally supplied and connected to device pins that serve groups of IOB blocks, called banks. Consequently, restrictions exist about which I/O standards can be combined within a given bank.

Eight I/O banks result from dividing each edge of the FPGA into two banks, as shown in Figure 15 and Figure 16. Each bank has multiple V_{CCO} pins, all of which must be connected to the same voltage. This voltage is determined by the output standards in use.

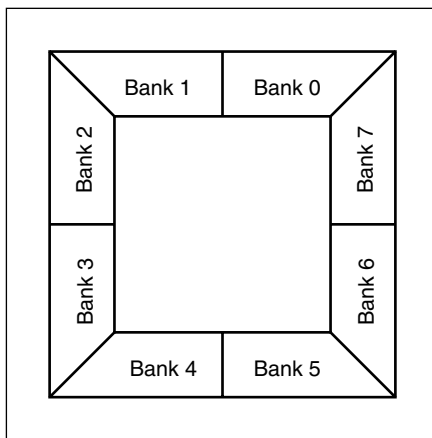


ug002_c2_014_112900

Figure 15: I/O Banks: Wire-Bond Packages (FG) Top View

Within a bank, output standards can be mixed only if they use the same V_{CCO} . Compatible standards are shown in Table 7. GTL and GTLP appear under all voltages because their open-drain outputs do not depend on V_{CCO} .

Some input standards require a user-supplied threshold voltage, V_{REF} . In this case, certain user-I/O pins are automatically configured as inputs for the V_{REF} voltage. Approximately one in six of the I/O pins in the bank assume this role. Table 8 lists compatible input standards.



ds031_66_112900

Figure 16: I/O Banks: Flip-Chip Packages (FF) Top View

V_{REF} pins within a bank are interconnected internally, and consequently only one V_{REF} voltage can be used within each bank. However, for correct operation, all V_{REF} pins in the bank must be connected to the external reference voltage source.

The V_{CCO} and the V_{REF} pins for each bank appear in the device pinout tables. Within a given package, the number of V_{REF} and V_{CCO} pins can vary depending on the size of device. In larger devices, more I/O pins convert to V_{REF} pins. Since these are always a superset of the V_{REF} pins used for smaller devices, it is possible to design a PCB that permits migration to a larger device if necessary.

Table 7: Compatible Output Standards

V_{CCO}	Compatible Standards
3.3V ⁽¹⁾	LVTTTL, LVCMOS33, PCI/PCI-X, LVDCI_33
2.5V	SSTL2 (I & II), LVCMOS25, GTL, GTLP, LVDS_25, LVDSEXT_25, LVDS_25_DCI, LVDSEXT_25_DCI, LVDCI_25, LVDCI_DV2_25, SSTL2_DCI (I & II), LDT, ULVDS, BLVDS, LVPECL_25
1.8V	HSTL (I, II, III, & IV), HSTL_DCI (I,II, III & IV), HSTL18 (I, II, III, & IV), HSTL18_DCI (I,II, III & IV), LVCMOS18, GTL, GTLP, LVDCI_18, LVDCI_DV2_18, SSTL18_I, SSTL18_II, SSTL18_I_DCI, SSTL18_II_DCI
1.5V	HSTL (I, II, III, & IV), HSTL_DCI (I,II, III & IV), LVCMOS15, GTL, GTLP, LVDCI_15, LVDCI_DV2_15, GTLP_DCI
1.2V	GTL_DCI

Notes:

1. See application note [XAPP653](#) for detailed information.

All V_{REF} pins for the largest device anticipated must be connected to the V_{REF} voltage and not used for I/O. In smaller devices, some V_{CCO} pins used in larger devices do not connect within the package. These unconnected pins can be left unconnected externally, or, if necessary, they can be connected to the V_{CCO} voltage to permit migration to a larger device.

Table 8: Compatible Input Standards

V _{REF}	V _{CCO}				
	3.3V	2.5V	1.8V	1.5V	1.2V
No V _{REF}	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25 ⁽²⁾	LVC MOS25, LVDCI_25, LVDCI_DV2_25, LVDS_25, LVDSEXT_25, LVPECL_25, LVDS_25_DCI, LVDSEXT_25_DCI, LDT, BLVDS, ULVDS_25	LVC MOS18, LVDCI_18, LVDCI_DV2_18, LVDS_25, LDT, ULVDS_25	LVC MOS15, LVDCI_15, LVDCI_DV2_15, LDT, ULVDS_25	
1.5V	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVC MOS25, LVDCI_25, LVDCI_DV2_25, LVDS_25, LVDSEXT_25, LVPECL_25, LVDS_25_DCI, LVDSEXT_25_DCI, LDT, BLVDS, ULVDS_25	LVC MOS18, LVDCI_18, LVDCI_DV2_18, LVDS_25, LDT, ULVDS_25	LVC MOS15, LVDCI_15, LVDCI_DV2_15, LDT, ULVDS_25	
1.25V	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVC MOS25, LVDCI_25, LVDCI_DV2_25, LVDS_25, LVDSEXT_25, LVPECL_25, LVDS_25_DCI, LVDSEXT_25_DCI, LDT, BLVDS, ULVDS_25, SSTL2_I_DCI, SSTL2_II_DCI	LVC MOS18, LVDCI_18, LVDCI_DV2_18, LVDS_25, LDT, ULVDS_25	LVC MOS15, LVDCI_15, LVDCI_DV2_15, LDT, ULVDS_25	
	SSTL2_I, SSTL2_II	SSTL2_I, SSTL2_II			
1.1V	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25, HSTL18_DCI_III, HSTL18_DCI_IV		
	HSTL18_III, HSTL18_IV	HSTL18_III, HSTL18_IV	HSTL18_III, HSTL18_IV	HSTL18_III, HSTL18_IV	
1.0V	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVC MOS25, LVDCI_25, LVDCI_DV2_25, LVDS_25, LVDSEXT_25, LVPECL_25, LVDS_25_DCI, LVDSEXT_25_DCI, LDT, BLVDS, ULVDS_25	LVC MOS18, LVDCI_18, LVDCI_DV2_18, LVDS_25, LDT, ULVDS_25	LVC MOS15, LVDCI_15, LVDCI_DV2_15, GTLP_DCI, LDT, ULVDS_25	
	GTLP	GTLP	GTLP	GTLP	GTLP
0.9V	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVC MOS25, LVDCI_25, LVDCI_DV2_25, LVDS_25, LVDSEXT_25, LVPECL_25, LVDS_25_DCI, LVDSEXT_25_DCI, LDT, BLVDS, ULVDS_25	LVC MOS18, LVDCI_18, LVDCI_DV2_18, LVDS_25, LDT, ULVDS_25, SSTL18_I_DCI, SSTL18_II_DCI, HSTL_III_DCI, HSTL_IV_DCI, HSTL18_DCI_I, HSTL18_DCI_II	LVC MOS15, LVDCI_15, LVDCI_DV2_15, LDT, ULVDS_25, HSTL_III_DCI, HSTL_IV_DCI	
	SSTL18_I, SSTL18_II, HSTL_III, HSTL_IV, HSTL18_I, HSTL18_II	SSTL18_I, SSTL18_II, HSTL_III, HSTL_IV, HSTL18_I, HSTL18_II	SSTL18_I, SSTL18_II, HSTL_III, HSTL_IV, HSTL18_I, HSTL18_II	HSTL_III, HSTL_IV	
0.8V	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVC MOS25, LVDCI_25, LVDCI_DV2_25, LVDS_25, LVDSEXT_25, LVPECL_25, LVDS_25_DCI, LVDSEXT_25_DCI, LDT, BLVDS, ULVDS_25	LVC MOS18, LVDCI_18, LVDCI_DV2_18, LVDS_25, LDT, ULVDS_25	LVC MOS15, LVDCI_15, LVDCI_DV2_15, LDT, ULVDS_25	GTL_DCI
	GTL	GTL	GTL	GTL	GTL
0.75V	LVTTTL, LVCMOS33, LVDCI_33, PCI33_3, PCI66_3, PCI-X, LVDS_25, LVDSEXT_25, LDT, BLVDS, ULVDS_25	LVC MOS25, LVDCI_25, LVDCI_DV2_25, LVDS_25, LVDSEXT_25, LVPECL_25, LVDS_25_DCI, LVDSEXT_25_DCI, LDT, BLVDS, ULVDS_25	LVC MOS18, LVDCI_18, LVDCI_DV2_18, LVDS_25, LDT, ULVDS_25	LVC MOS15, LVDCI_15, LVDCI_DV2_15, LDT, ULVDS_25, HSTL_I_DCI, HSTL_II_DCI	
	HSTL_I, HSTL_II	HSTL_I, HSTL_II	HSTL_I, HSTL_II	HSTL_I, HSTL_II	

Notes:

1. V_{REF}-controlled inputs are completely independent of V_{CCO}-controlled inputs. Therefore, V_{REF}-controlled inputs can be placed in banks with V_{CCO}-controlled inputs and outputs of different voltages, as long as V_{CCO} is not below the supply voltage of the particular standard.
2. All non-DCI differential inputs are V_{CCAUX} controlled. This makes them (Inputs Only) very flexible in terms of banking rules.
3. It is important to ensure that the input DC levels are within V_{CCO} + 0.5V, because all user I/Os have clamp diodes connected to V_{CCO}.

Digitally Controlled Impedance (DCI)

Today's chip output signals with fast edge rates require termination to prevent reflections and maintain signal integrity. High pin count packages (especially ball grid arrays) can not accommodate external termination resistors.

Virtex-II Pro XCITE DCI provides controlled impedance drivers and on-chip termination for single-ended and differential I/Os. This eliminates the need for external resistors and improves signal integrity. The DCI feature can be used on any IOB by selecting one of the DCI I/O standards.

When applied to inputs, DCI provides input parallel termination. When applied to outputs, DCI provides controlled impedance drivers (series termination) or output parallel termination.

DCI operates independently on each I/O bank. When a DCI I/O standard is used in a particular I/O bank, external reference resistors must be connected to two dual-function pins on the bank. These resistors, voltage reference of N transistor (VRN) and the voltage reference of P transistor (VRP) are shown in **Figure 17**.

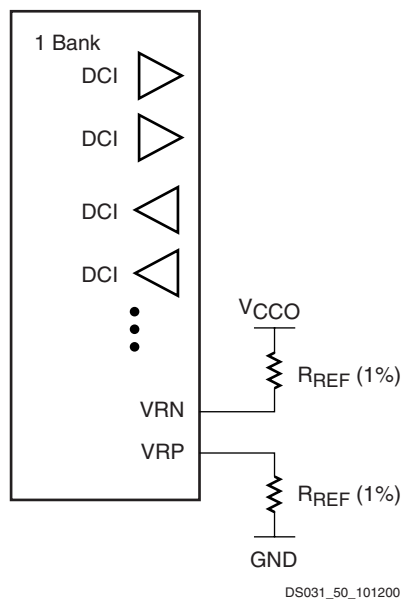


Figure 17: DCI in a Virtex-II Pro Bank

When used with a terminated I/O standard, the value of the resistors are specified by the standard (typically 50Ω). When used with a controlled impedance driver, the resistors set the output impedance of the driver within the specified range (20Ω to 100Ω). For all series and parallel terminations listed in **Table 9** and **Table 10**, the reference resistors must have the same value for any given bank. One percent resistors are recommended.

The DCI system adjusts the I/O impedance to match the two external reference resistors, or half of the reference resistors, and compensates for impedance changes due to voltage and/or temperature fluctuations. The adjustment is done by turning parallel transistors in the IOB on or off.

Controlled Impedance Drivers (Series Termination)

DCI can be used to provide a buffer with a controlled output impedance. It is desirable for this output impedance to match the transmission line impedance (Z_0). Virtex-II Pro input buffers also support LVDCI and LVDCI_DV2 I/O standards.

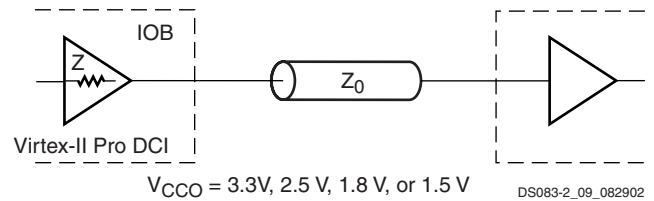


Figure 18: Internal Series Termination

Table 9: SelectIO-Ultra Controlled Impedance Buffers

V _{CCO}	DCI	DCI Half Impedance
3.3V	LVDCI_33	N/A
2.5V	LVDCI_25	LVDCI_DV2_25
1.8V	LVDCI_18	LVDCI_DV2_18
1.5V	LVDCI_15	LVDCI_DV2_15

Controlled Impedance Terminations (Parallel Termination)

DCI also provides on-chip termination for SSTL2, SSTL18, HSTL (Class I, II, III, or IV), LVDS_25, LVDSEXT_25, and GTL/GTLP receivers or transmitters on bidirectional lines. **Table 10** and **Table 11** list the on-chip parallel terminations available in Virtex-II Pro devices. V_{CCO} must be set according to **Table 5**. There is a V_{CCO} requirement for GTL_DCI and GTLP_DCI, due to the on-chip termination resistor.

Table 10: SelectIO-Ultra Buffers With On-Chip Parallel Termination

I/O Standard	External Termination	On-Chip Termination
SSTL2 Class I	SSTL2_I	SSTL2_I_DCI ⁽¹⁾
SSTL2 Class II	SSTL2_II	SSTL2_II_DCI ⁽¹⁾
SSTL18 Class I	SSTL18_I	SSTL18_I_DCI
SSTL18 Class II	SSTL18_II	SSTL18_II_DCI
HSTL Class I	HSTL_I	HSTL_I_DCI
	HSTL_I_18	HSTL_I_DCI_18
HSTL Class II	HSTL_II	HSTL_II_DCI
	HSTL_II_18	HSTL_II_DCI_18
HSTL Class III	HSTL_III	HSTL_III_DCI
	HSTL_III_18	HSTL_III_DCI_18
HSTL Class IV	HSTL_IV	HSTL_IV_DCI
	HSTL_IV_18	HSTL_IV_DCI_18
GTL	GTL	GTL_DCI
GTLP	GTLP	GTLP_DCI

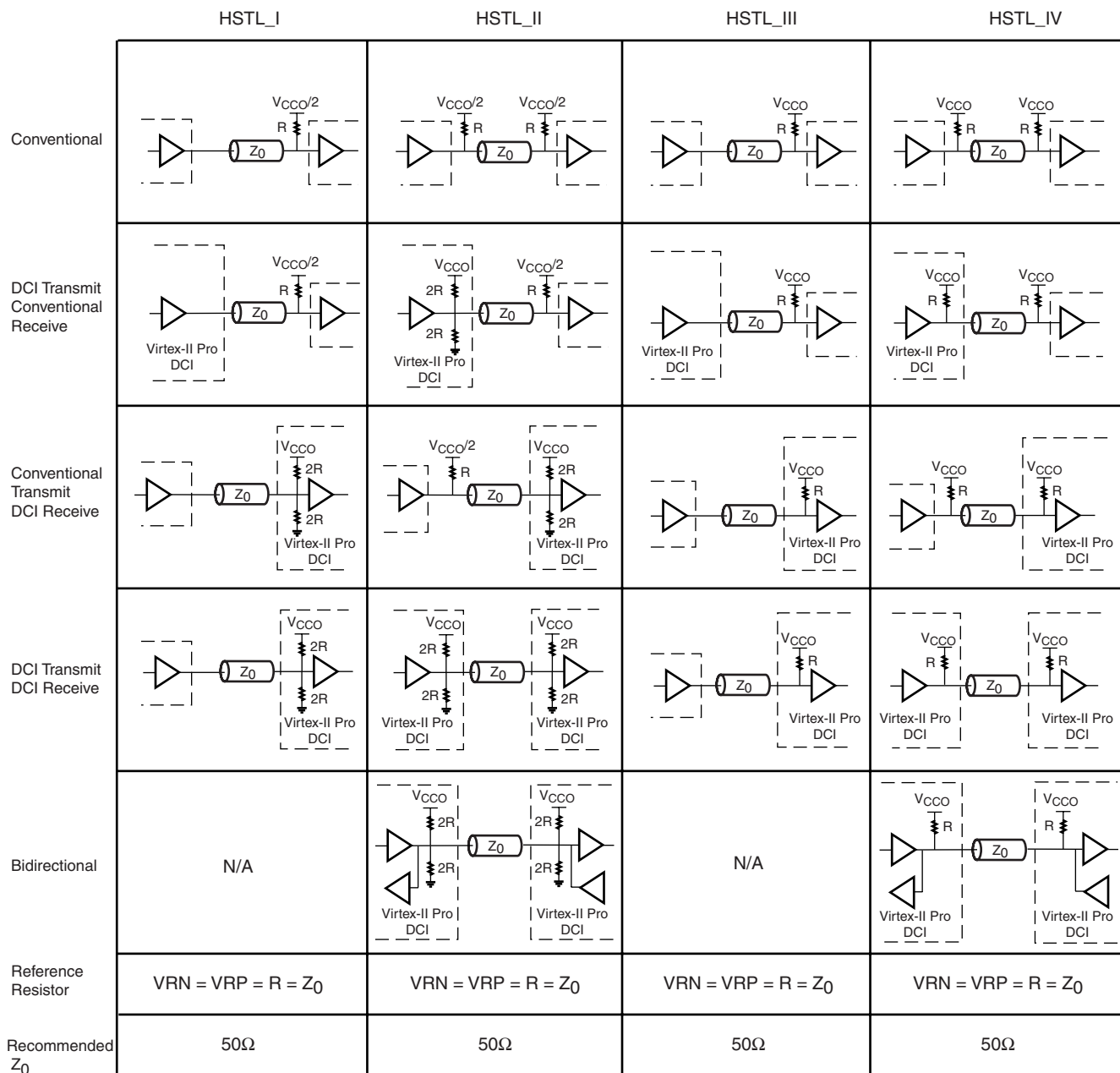
Notes:

1. SSTL compatible.

Table 11: SelectIO-Ultra Differential Buffers With On-Chip Termination

I/O Standard	External Termination	On-Chip Termination
LVDS	LVDS_25	LVDS_25_DCI
LVDSSEXT	LVDSSEXT_25	LVDSSEXT_25_DCI

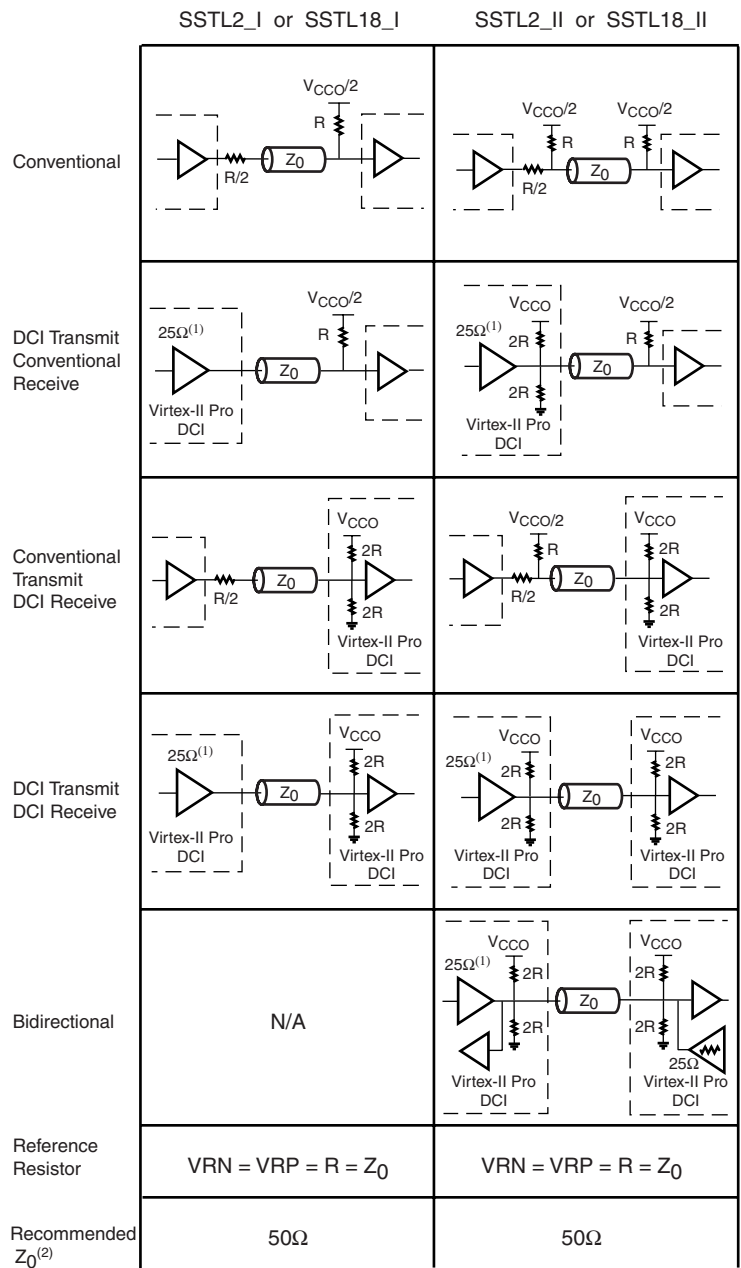
Figure 19 provides examples illustrating the use of the HSTL_I_DCI, HSTL_II_DCI, HSTL_III_DCI, and HSTL_IV_DCI I/O standards. For a complete list, see the *Virtex-II Pro Platform FPGA User Guide*.



DS083-2_65a_082102

Figure 19: HSTL DCI Usage Examples

Figure 20 provides examples illustrating the use of the SSSL2_I_DCI, SSSL2_II_DCI, SSSL18_I_DCI, and SSSL18_II_DCI I/O standards. For a complete list, see the *Virtex-II Pro Platform FPGA User Guide*.



DS083-2_65b_011603

Notes:

1. The SSSL-compatible 25Ω series resistor is accounted for in the DCI buffer, and it is not DCI controlled.
2. Z_0 is the recommended PCB trace impedance.

Figure 20: SSSL DCI Usage Examples

Figure 21 provides examples illustrating the use of the LVDS_25_DCI and LVDSEXT_25_DCI I/O standards. For a complete list, see the *Virtex-II Pro Platform FPGA User Guide*.

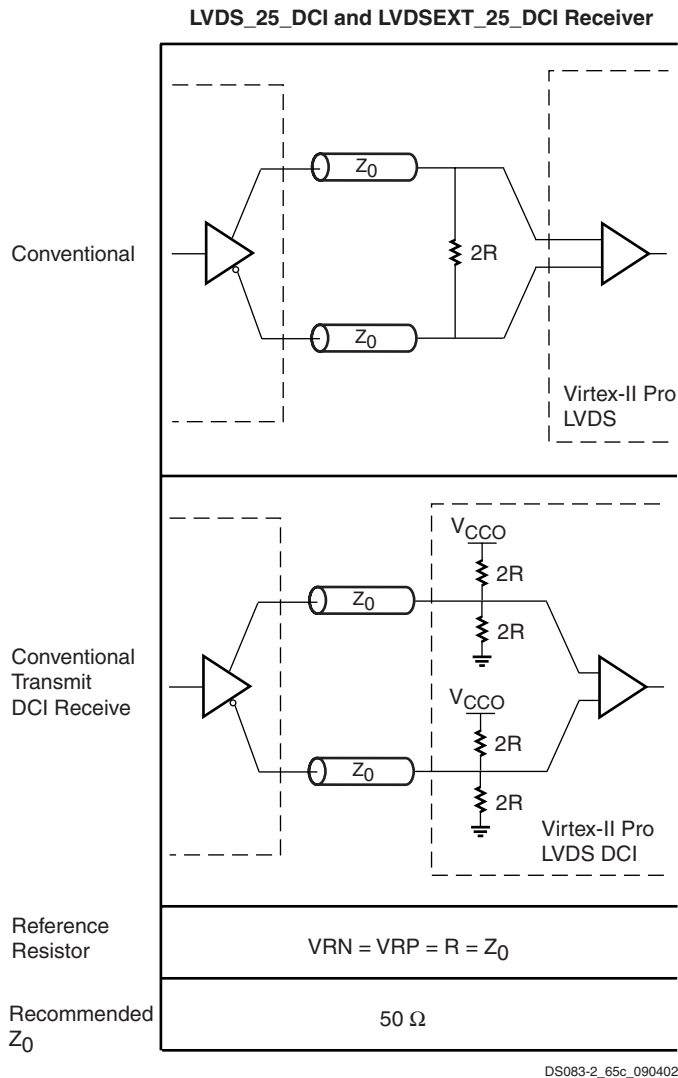


Figure 21: LVDS DCI Usage Examples

Configurable Logic Blocks (CLBs)

The Virtex-II Pro configurable logic blocks (CLB) are organized in an array and are used to build combinatorial and synchronous logic designs. Each CLB element is tied to a switch matrix to access the general routing matrix, as shown in Figure 22.

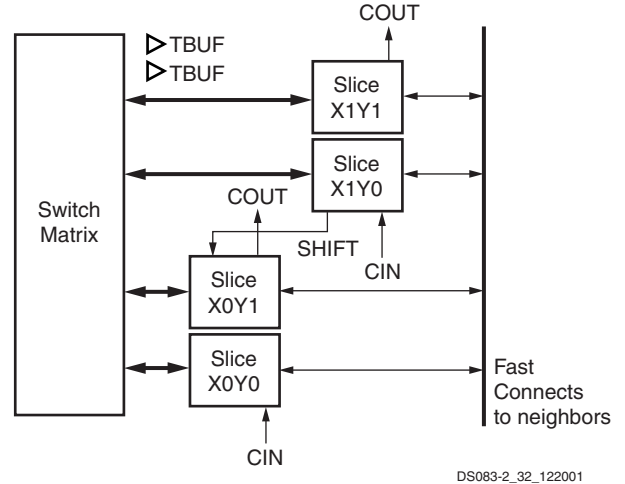


Figure 22: Virtex-II Pro CLB Element

A CLB element comprises 4 similar slices, with fast local feedback within the CLB. The four slices are split in two columns of two slices with two independent carry logic chains and one common shift chain.

Slice Description

Each slice includes two 4-input function generators, carry logic, arithmetic logic gates, wide function multiplexers and two storage elements. As shown in Figure 23, each 4-input function generator is programmable as a 4-input LUT, 16 bits of distributed SelectRAM+ memory, or a 16-bit variable-tap shift register element.

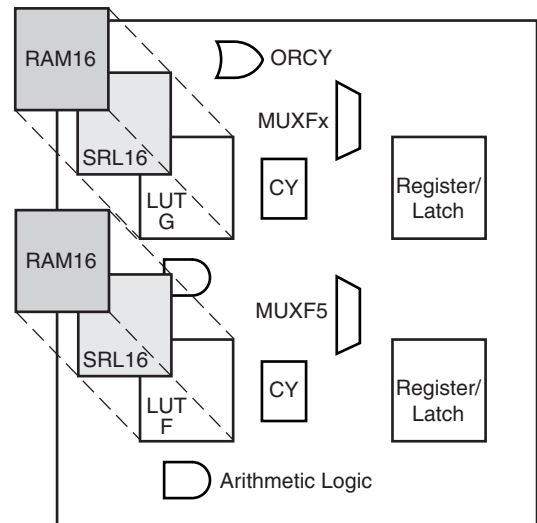
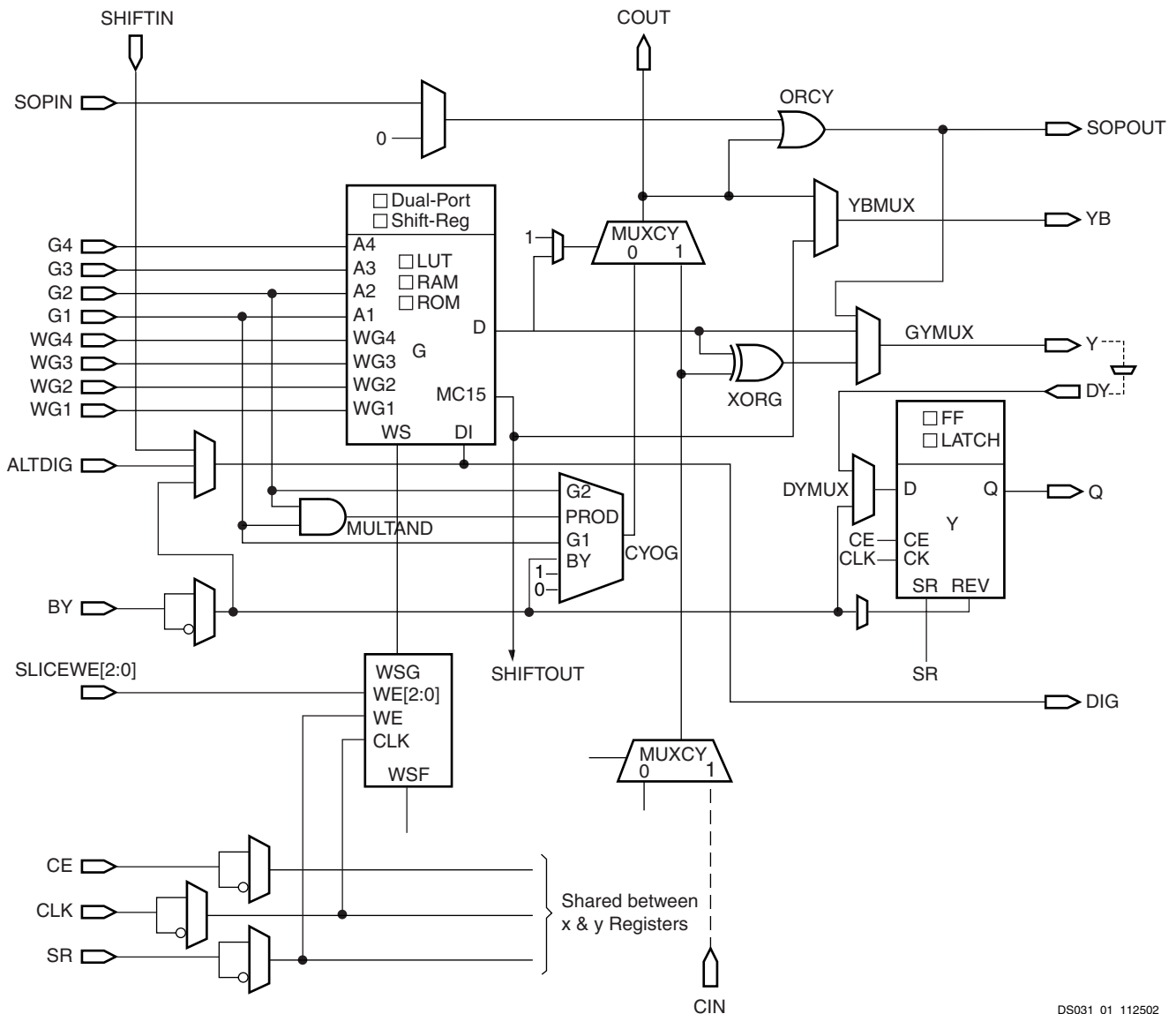


Figure 23: Virtex-II Pro Slice Configuration

The output from the function generator in each slice drives both the slice output and the D input of the storage element. **Figure 24** shows a more detailed view of a single slice.



DS031_01_112502

Figure 24: Virtex-II Pro Slice (Top Half)

Configurations

Look-Up Table

Virtex-II Pro function generators are implemented as 4-input look-up tables (LUTs). Four independent inputs are provided to each of the two function generators in a slice (F and G). These function generators are each capable of implementing any arbitrarily defined boolean function of four inputs. The propagation delay is therefore independent of the function implemented. Signals from the function generators can exit the slice (X or Y output), can input the XOR dedicated gate (see arithmetic logic), or input the carry-logic multiplexer (see fast look-ahead carry logic), or feed the D input of the storage element, or go to the MUXF5 (not shown in **Figure 24**).

In addition to the basic LUTs, the Virtex-II Pro slice contains logic (MUXF5 and MUXFX multiplexers) that combines function generators to provide any function of five, six, seven, or eight inputs. The MUXFX is either MUXF6, MUXF7, or MUXF8 according to the slice considered in the CLB. Selected functions up to nine inputs (MUXF5 multiplexer) can be implemented in one slice. The MUXFX can also be a MUXF6, MUXF7, or MUXF8 multiplexer to map any function of six, seven, or eight inputs and selected wide logic functions.

Register/Latch

The storage elements in a Virtex-II Pro slice can be configured either as edge-triggered D-type flip-flops or as level-sensitive latches. The D input can be directly driven by

the X or Y output via the DX or DY input, or by the slice inputs bypassing the function generators via the BX or BY input. The clock enable signal (CE) is active High by default. If left unconnected, the clock enable for that storage element defaults to the active state.

In addition to clock (CK) and clock enable (CE) signals, each slice has set and reset signals (SR and BY slice inputs). SR forces the storage element into the state specified by the attribute SRHIGH or SRLow. SRHIGH forces a logic 1 when SR is asserted. SRLow forces a logic 0. When SR is used, an optional second input (BY) forces the storage element into the opposite state via the REV pin. The reset condition is predominant over the set condition. (See [Figure 25.](#))

The initial state after configuration or global initial state is defined by a separate INIT0 and INIT1 attribute. By default, setting the SRLow attribute sets INIT0, and setting the SRHIGH attribute sets INIT1.

For each slice, set and reset can be set to be synchronous or asynchronous. Virtex-II Pro devices also have the ability to set INIT0 and INIT1 independent of SRHIGH and SRLow.

The control signals clock (CLK), clock enable (CE) and set/reset (SR) are common to both storage elements in one slice. All of the control signals have independent polarity. Any inverter placed on a control input is automatically absorbed.

- No set or reset
- Synchronous set
- Synchronous reset
- Synchronous set and reset
- Asynchronous set (preset)
- Asynchronous reset (clear)
- Asynchronous set and reset (preset and clear)

The synchronous reset has precedence over a set, and an asynchronous clear has precedence over a preset.

Distributed SelectRAM+ Memory

Each function generator (LUT) can implement a 16 x 1-bit synchronous RAM resource called a distributed SelectRAM+ element. SelectRAM+ elements are configurable within a CLB to implement the following:

- Single-Port 16 x 8-bit RAM
- Single-Port 32 x 4-bit RAM
- Single-Port 64 x 2-bit RAM
- Single-Port 128 x 1-bit RAM
- Dual-Port 16 x 4-bit RAM
- Dual-Port 32 x 2-bit RAM
- Dual-Port 64 x 1-bit RAM

Distributed SelectRAM+ memory modules are synchronous (write) resources. The combinatorial read access time is extremely fast, while the synchronous write simplifies high-speed designs. A synchronous read can be implemented with a storage element in the same slice. The distributed SelectRAM+ memory and the storage element share the same clock input. A Write Enable (WE) input is active High, and is driven by the SR input.

[Table 12](#) shows the number of LUTs (2 per slice) occupied by each distributed SelectRAM+ configuration.

Table 12: Distributed SelectRAM+ Configurations

RAM	Number of LUTs
16 x 1S	1
16 x 1D	2
32 x 1S	2
32 x 1D	4
64 x 1S	4
64 x 1D	8
128 x 1S	8

Notes:

1. S = single-port configuration; D = dual-port configuration

For single-port configurations, distributed SelectRAM+ memory has one address port for synchronous writes and asynchronous reads.

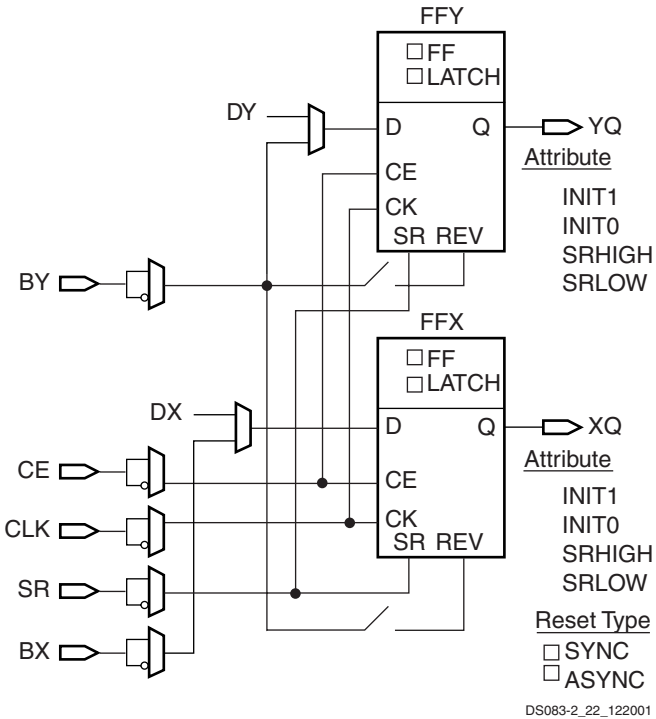


Figure 25: Register / Latch Configuration in a Slice

The set and reset functionality of a register or a latch can be configured as follows:

For dual-port configurations, distributed SelectRAM+ memory has one port for synchronous writes and asynchronous reads and another port for asynchronous reads. The function generator (LUT) has separated read address inputs (A1, A2, A3, A4) and write address inputs (WG1/WF1, WG2/WF2, WG3/WF3, WG4/WF4).

In single-port mode, read and write addresses share the same address bus. In dual-port mode, one function generator (R/W port) is connected with shared read and write addresses. The second function generator has the A inputs (read) connected to the second read-only port address and the W inputs (write) shared with the first read/write port address.

Figure 26, Figure 27, and Figure 28 illustrate various example configurations.

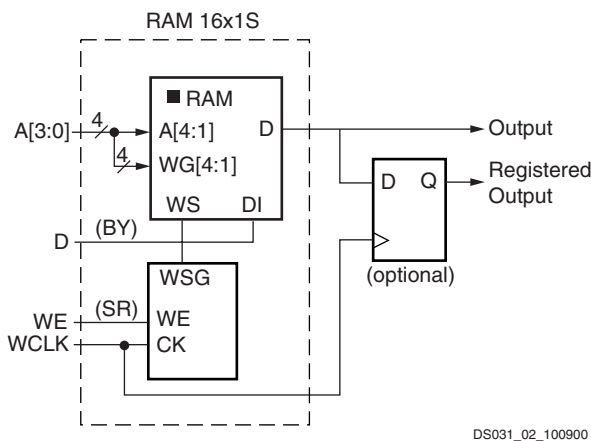


Figure 26: Distributed SelectRAM+ (RAM16x1S)

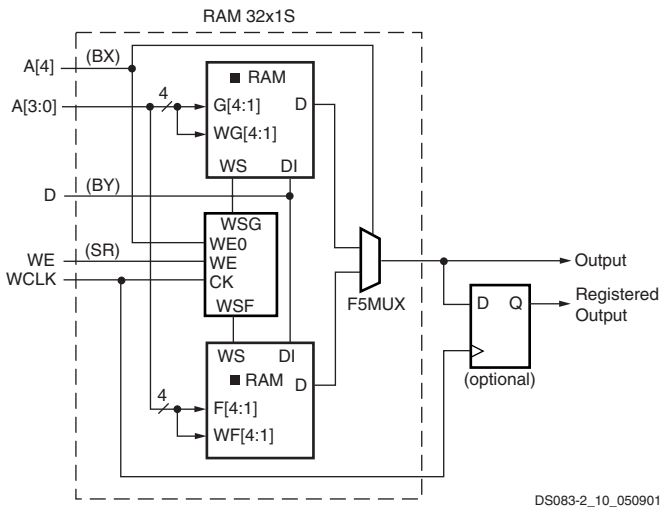


Figure 27: Single-Port Distributed SelectRAM+ (RAM32x1S)

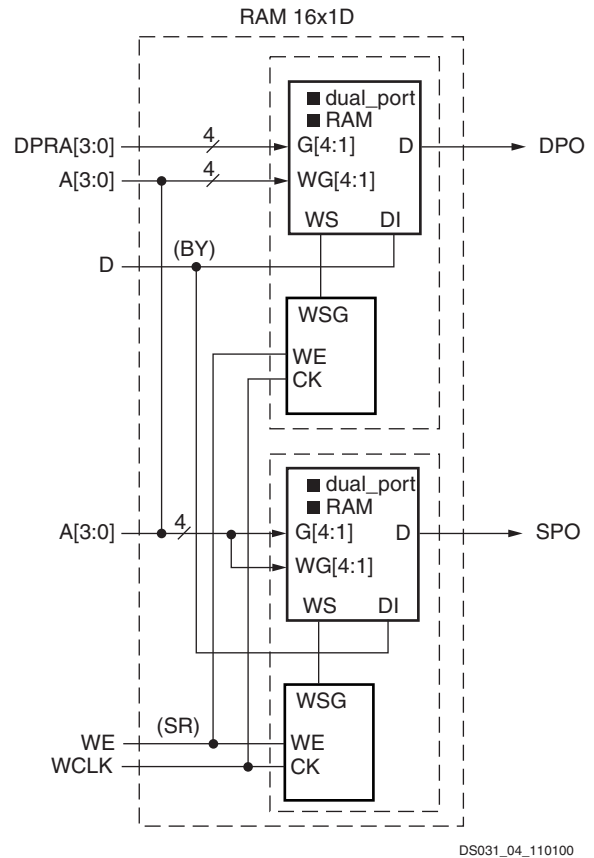


Figure 28: Dual-Port Distributed SelectRAM+ (RAM16x1D)

Similar to the RAM configuration, each function generator (LUT) can implement a 16 x 1-bit ROM. Five configurations are available: ROM16x1, ROM32x1, ROM64x1, ROM128x1, and ROM256x1. The ROM elements are cascadable to implement wider or/and deeper ROM. ROM contents are loaded at configuration. Table 13 shows the number of LUTs occupied by each configuration.

Table 13: ROM Configuration

ROM	Number of LUTs
16 x 1	1
32 x 1	2
64 x 1	4
128 x 1	8 (1 CLB)
256 x 1	16 (2 CLBs)

Shift Registers

Each function generator can also be configured as a 16-bit shift register. The write operation is synchronous with a clock input (CLK) and an optional clock enable, as shown in Figure 29. A dynamic read access is performed through the 4-bit address bus, A[3:0]. The configurable 16-bit shift register cannot be set or reset. The read is asynchronous; however, the storage element or flip-flop is available to implement a synchronous read. Any of the 16 bits can be read out asynchronously by varying the address. The storage element should always be used with a constant address. For example, when building an 8-bit shift register and configuring the addresses to point to the 7th bit, the 8th bit can be the flip-flop. The overall system performance is improved by using the superior clock-to-out of the flip-flops.

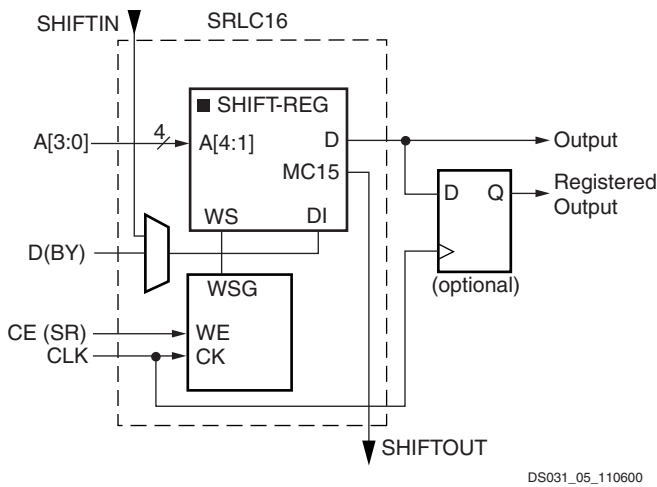


Figure 29: Shift Register Configurations

An additional dedicated connection between shift registers allows connecting the last bit of one shift register to the first bit of the next, without using the ordinary LUT output. (See Figure 30.) Longer shift registers can be built with dynamic access to any bit in the chain. The shift register chaining and the MUXF5, MUXF6, and MUXF7 multiplexers allow up to a 128-bit shift register with addressable access to be implemented in one CLB.

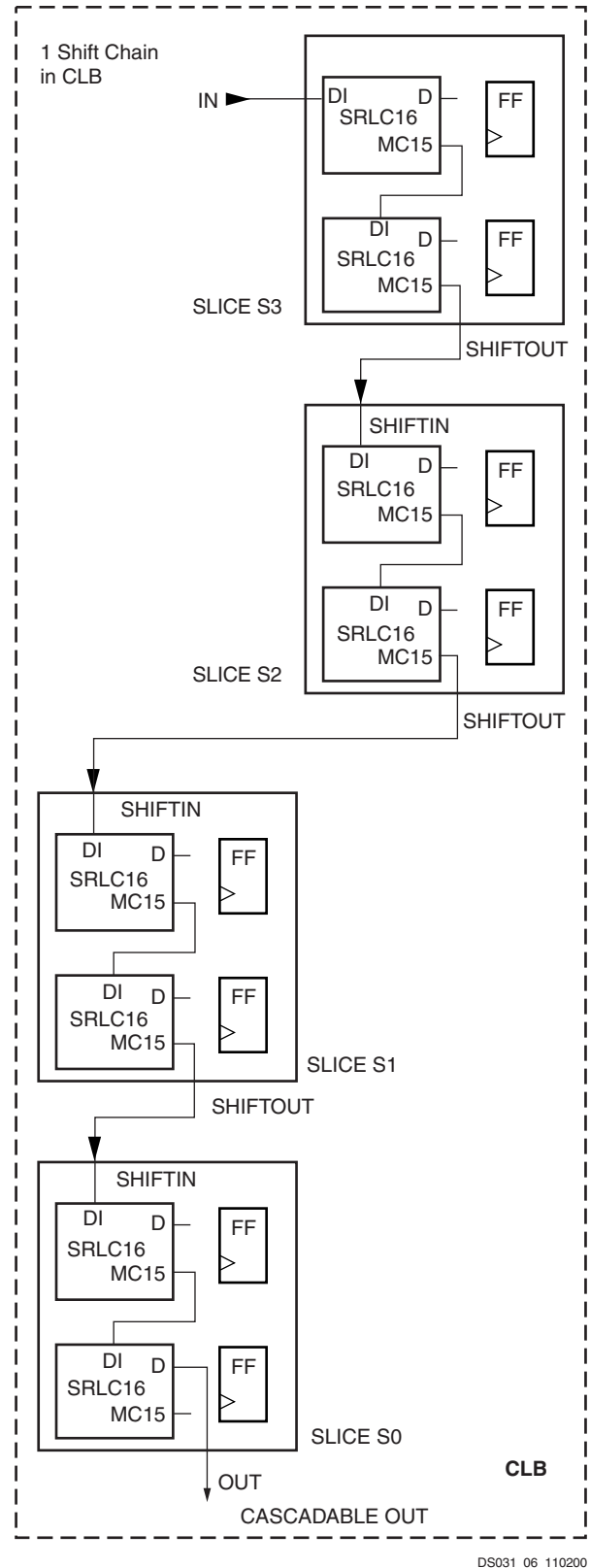


Figure 30: Cascadable Shift Register

Multiplexers

Virtex-II Pro function generators and associated multiplexers can implement the following:

- 4:1 multiplexer in one slice
- 8:1 multiplexer in two slices
- 16:1 multiplexer in one CLB element (4 slices)
- 32:1 multiplexer in two CLB elements (8 slices)

Each Virtex-II Pro slice has one MUXF5 multiplexer and one MUXFX multiplexer. The MUXFX multiplexer implements the MUXF6, MUXF7, or MUXF8, as shown in **Figure 31**. Each CLB element has two MUXF6 multiplexers, one MUXF7 multiplexer and one MUXF8 multiplexer. Examples of multiplexers are shown in the *Virtex-II Pro Platform FPGA User Guide*. Any LUT can implement a 2:1 multiplexer.

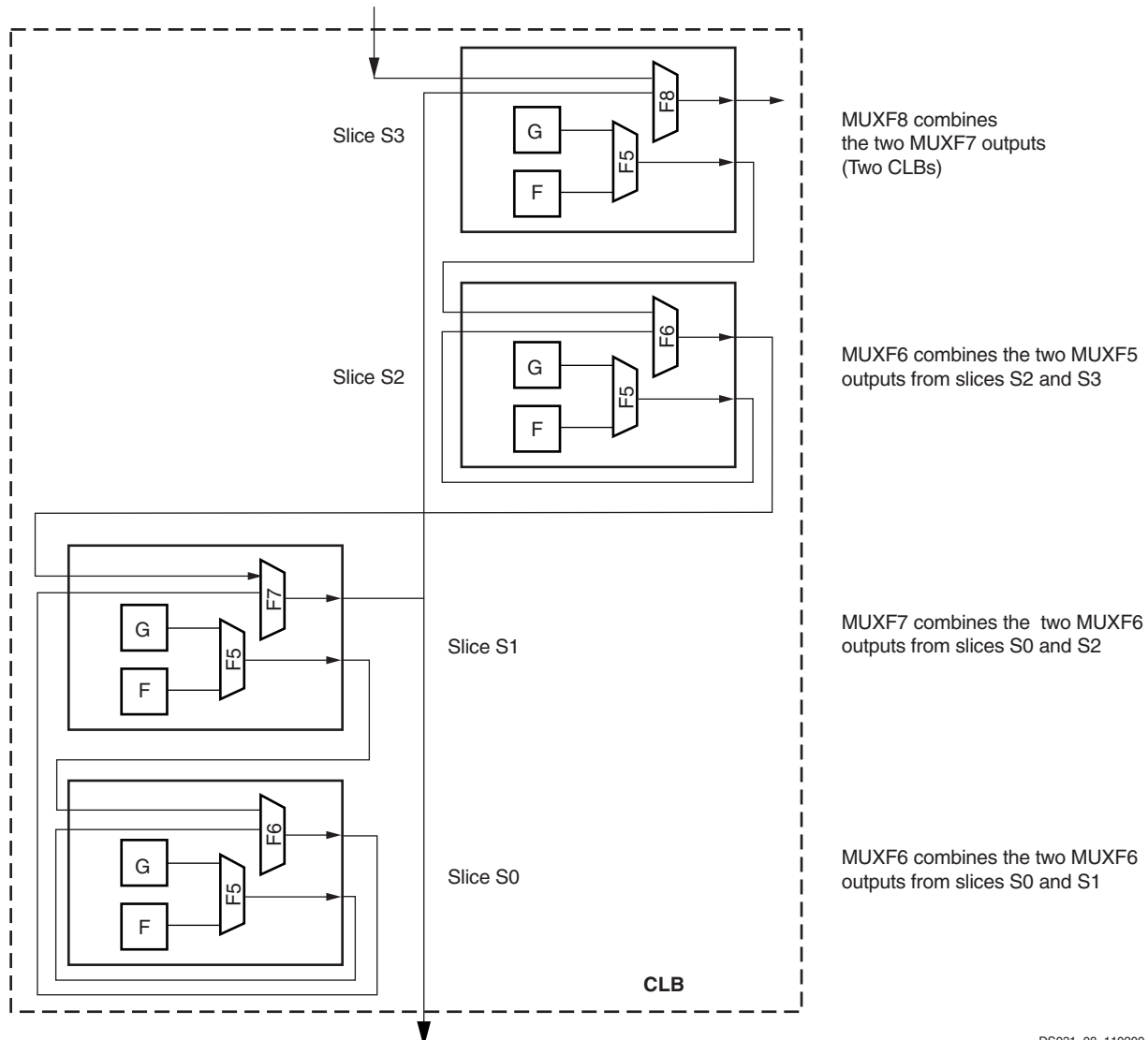


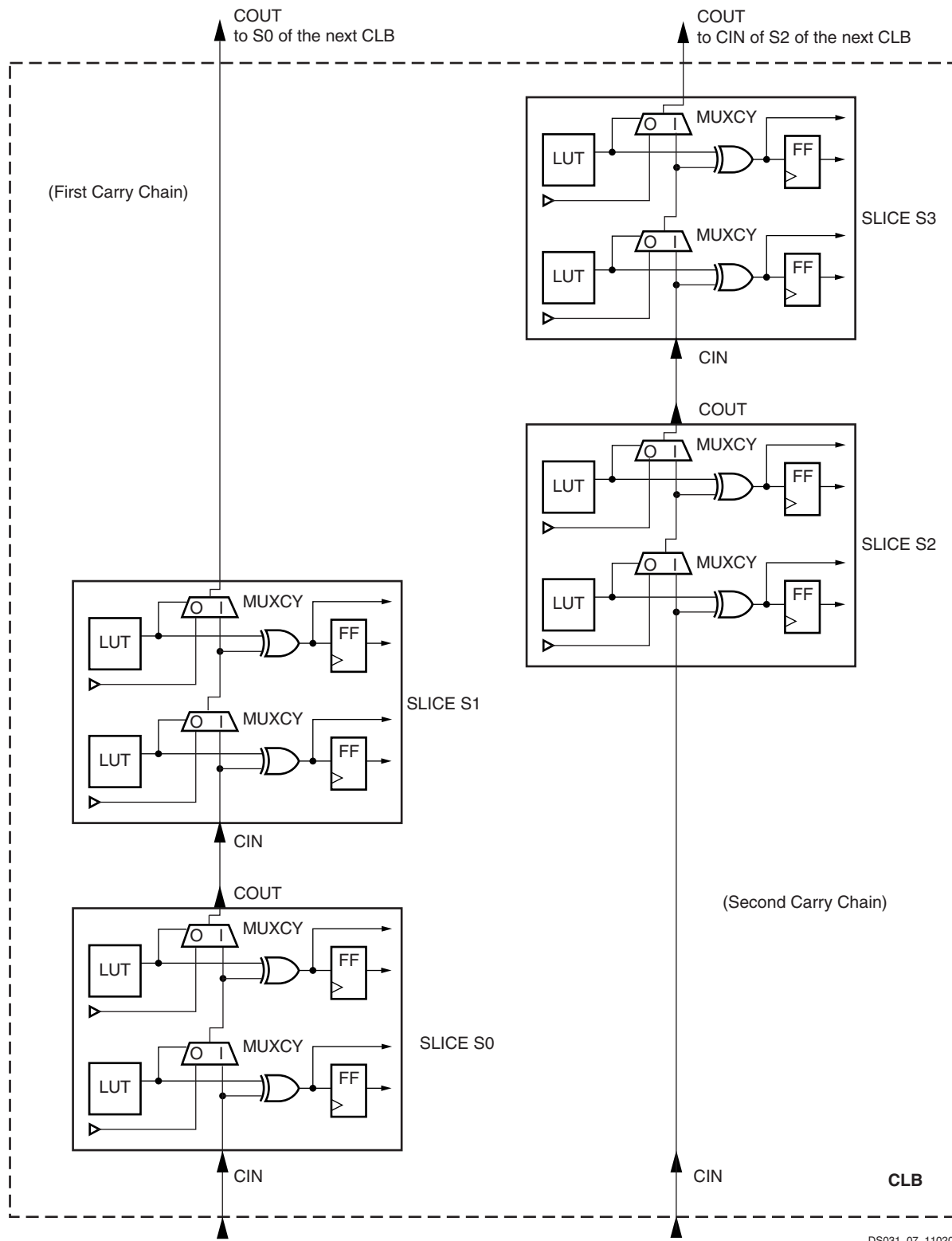
Figure 31: MUXF5 and MUXFX multiplexers

DS031_08_110200

Fast Lookahead Carry Logic

Dedicated carry logic provides fast arithmetic addition and subtraction. The Virtex-II Pro CLB has two separate carry chains, as shown in the **Figure 32**.

The height of the carry chains is two bits per slice. The carry chain in the Virtex-II Pro device is running upward. The dedicated carry path and carry multiplexer (MUXCY) can also be used to cascade function generators for implementing wide logic functions.



DS031_07_110200

Figure 32: Fast Carry Logic Path

Arithmetic Logic

The arithmetic logic includes an XOR gate that allows a 2-bit full adder to be implemented within a slice. In addition,

a dedicated AND (MULT_AND) gate (shown in Figure 24) improves the efficiency of multiplier implementation.

Sum of Products

Each Virtex-II Pro slice has a dedicated OR gate named ORCY, ORing together outputs from the slices carryout and the ORCY from an adjacent slice. The ORCY gate with the dedicated Sum of Products (SOP) chain are designed for

implementing large, flexible SOP chains. One input of each ORCY is connected through the fast SOP chain to the output of the previous ORCY in the same slice row. The second input is connected to the output of the top MUXCY in the same slice, as shown in **Figure 33**.

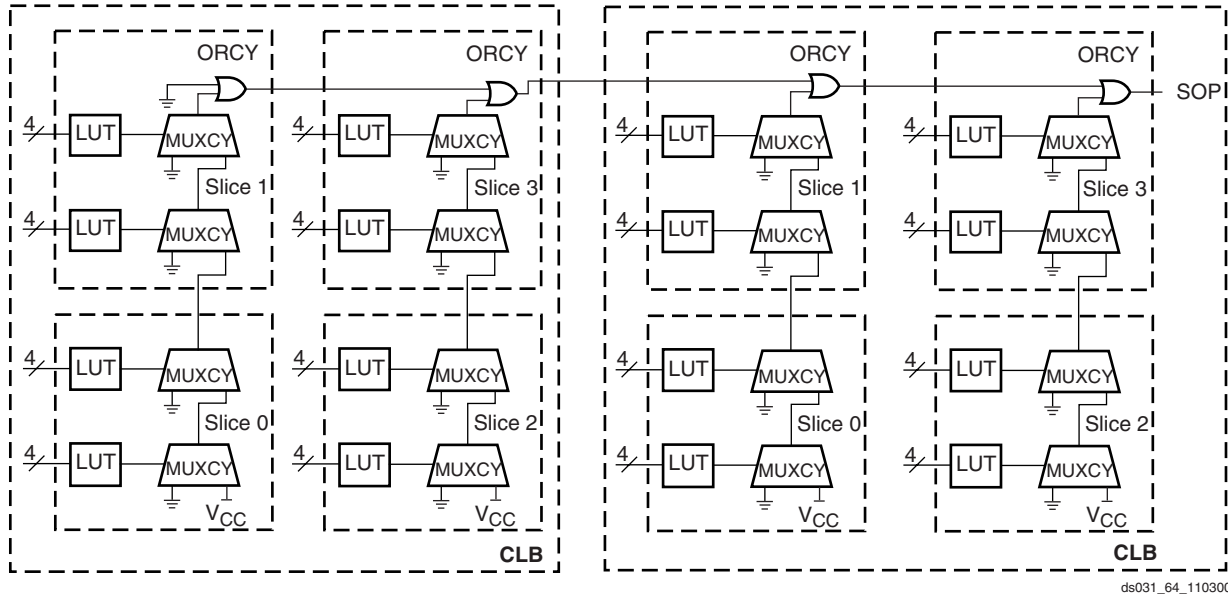


Figure 33: Horizontal Cascade Chain

LUTs and MUXCYs can implement large AND gates or other combinational logic functions. **Figure 34** illustrates

LUT and MUXCY resources configured as a 16-input AND gate.

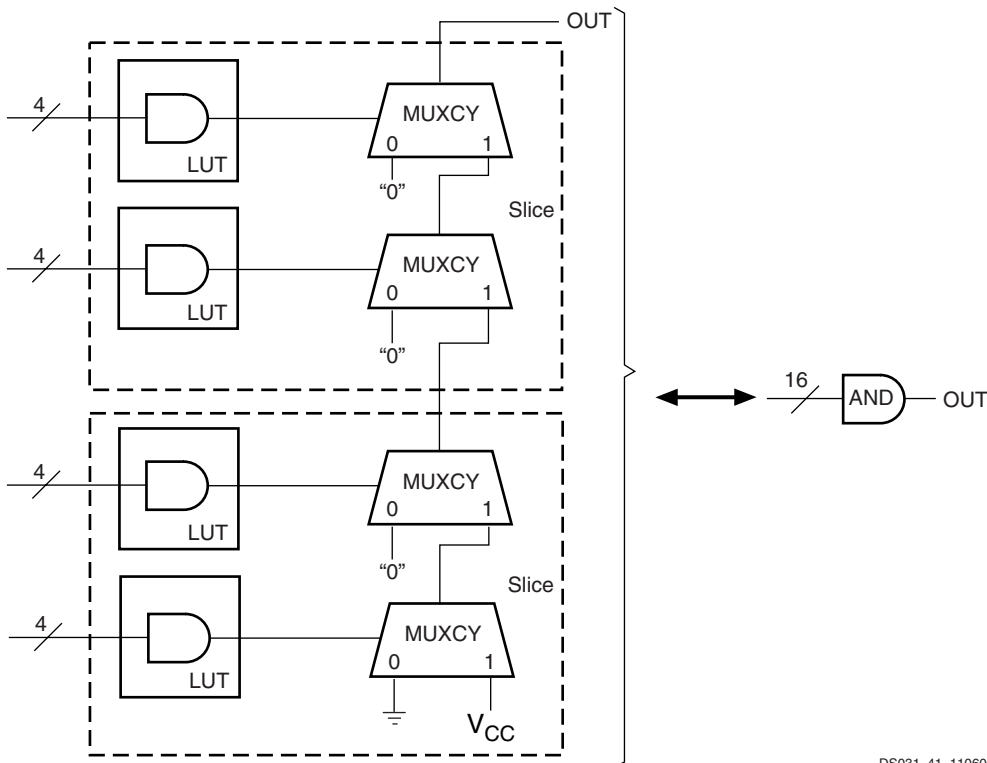


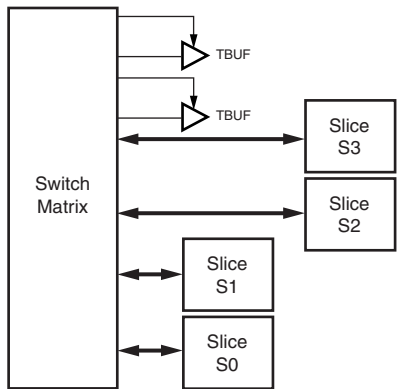
Figure 34: Wide-Input AND Gate (16 Inputs)

3-State Buffers

Introduction

Each Virtex-II Pro CLB contains two 3-state drivers (TBUFs) that can drive on-chip buses. Each 3-state buffer has its own 3-state control pin and its own input pin.

Each of the four slices have access to the two 3-state buffers through the switch matrix, as shown in Figure 35. TBUFs in neighboring CLBs can access slice outputs by direct connects. The outputs of the 3-state buffers drive horizontal routing resources used to implement 3-state buses.



DS031_37_060700

Figure 35: Virtex-II Pro 3-State Buffers

The 3-state buffer logic is implemented using AND-OR logic rather than 3-state drivers, so that timing is more predictable and less load dependant especially with larger devices.

Locations / Organization

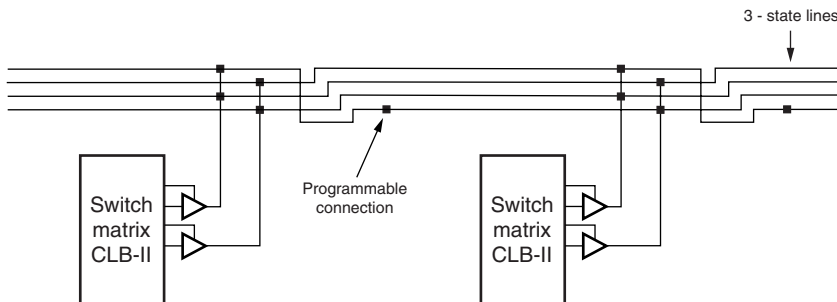
Four horizontal routing resources per CLB are provided for on-chip 3-state buses. Each 3-state buffer has access alternately to two horizontal lines, which can be partitioned as shown in Figure 36. The switch matrices corresponding to SelectRAM+ memory and multiplier or I/O blocks are skipped.

Number of 3-State Buffers

Table 14 shows the number of 3-state buffers available in each Virtex-II Pro device. The number of 3-state buffers is twice the number of CLB elements.

Table 14: Virtex-II Pro 3-State Buffers

Device	3-State Buffers per Row	Total Number of 3-State Buffers
XC2VP2	44	704
XC2VP4	44	1,760
XC2VP7	68	2,720
XC2VP20	92	5,152
XC2VP30	92	6,848
XC2VP40	116	9,696
XC2VP50	140	11,808
XC2VP70	164	16,544
XC2VP100	188	22,048
XC2VP125	212	27,808



DS031_09_032700

Figure 36: 3-State Buffer Connection to Horizontal Lines

CLB/Slice Configurations

Table 15 summarizes the logic resources in one CLB. All of the CLBs are identical and each CLB or slice can be imple-

mented in one of the configurations listed. Table 16 shows the available resources in all CLBs.

Table 15: Logic Resources in One CLB

Slices	LUTs	Flip-Flops	MULT_ANDs	Arithmetic & Carry-Chains	SOP Chains	Distributed SelectRAM+	Shift Registers	TBUF
4	8	8	8	2	2	128 bits	128 bits	2

Table 16: Virtex-II Pro Logic Resources Available in All CLBs

Device	CLB Array: Row x Column	Number of Slices	Number of LUTs	Max Distributed SelectRAM+ or Shift Register (bits)	Number of Flip-Flops	Number of Carry Chains ⁽¹⁾	Number of SOP Chains ⁽¹⁾
XC2VP2	16 x 22	1,408	2,816	45,056	2,816	44	32
XC2VP4	40 x 22	3,008	6,016	96,256	6,016	44	80
XC2VP7	40 x 34	4,928	9,856	157,696	9,856	68	80
XC2VP20	56 x 46	9,280	18,560	296,960	18,560	92	112
XC2VP30	80 x 46	13,696	27,392	438,272	27,392	92	160
XC2VP40	88 x 58	19,392	38,784	620,544	38,784	116	176
XC2VP50	88 x 70	23,616	47,232	755,712	47,232	140	176
XC2VP70	104 x 82	33,088	66,176	1,058,816	66,176	164	208
XC2VP100	120 x 94	44,096	88,192	1,411,072	88,192	188	240
XC2VP125	136 x 106	55,616	111,232	1,779,712	111,232	212	272

Notes:

1. The carry-chains and SOP chains can be split or cascaded.

18 Kb Block SelectRAM+ Resources

Introduction

Virtex-II Pro devices incorporate large amounts of 18 Kb block SelectRAM+ resources. These complement the distributed SelectRAM+ resources that provide shallow RAM structures implemented in CLBs. Each Virtex-II Pro block SelectRAM+ resource is an 18 Kb true dual-port RAM with two independently clocked and independently controlled synchronous ports that access a common storage area. Both ports are functionally identical. CLK, EN, WE, and SSR polarities are defined through configuration.

Each port has the following types of inputs: Clock and Clock Enable, Write Enable, Set/Reset, and Address, as well as separate Data/parity data inputs (for write) and Data/parity data outputs (for read).

Operation is synchronous; the block SelectRAM+ behaves like a register. Control, address and data inputs must (and need only) be valid during the set-up time window prior to a rising (or falling, a configuration option) clock edge. Data outputs change as a result of the same clock edge.

Configuration

Virtex-II Pro block SelectRAM+ supports various configurations, including single- and dual-port RAM and various

data/address aspect ratios. Supported memory configurations for single- and dual-port modes are shown in [Table 17](#).

Table 17: Dual- and Single-Port Configurations

16K x 1 bit	2K x 9 bits
8K x 2 bits	1K x 18 bits
4K x 4 bits	512 x 36 bits

Single-Port Configuration

As a single-port RAM, the block SelectRAM+ has access to the 18 Kb memory locations in any of the 2K x 9-bit, 1K x 18-bit, or 512 x 36-bit configurations and to 16 Kb memory locations in any of the 16K x 1-bit, 8K x 2-bit, or 4K x 4-bit configurations. The advantage of the 9-bit, 18-bit and 36-bit widths is the ability to store a parity bit for each eight bits. Parity bits must be generated or checked externally in user logic. In such cases, the width is viewed as 8 + 1, 16 + 2, or 32 + 4. These extra parity bits are stored and behave exactly as the other bits, including the timing parameters. Video applications can use the 9-bit ratio of Virtex-II Pro block SelectRAM+ memory to advantage.

Each block SelectRAM+ cell is a fully synchronous memory as illustrated in [Figure 37](#). Input data bus and output data bus widths are identical.

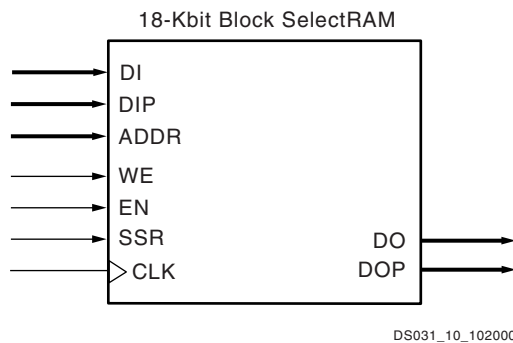


Figure 37: 18 Kb Block SelectRAM+ Memory in Single-Port Mode

Dual-Port Configuration

As a dual-port RAM, each port of block SelectRAM+ has access to a common 18 Kb memory resource. These are fully synchronous ports with independent control signals for each port. The data widths of the two ports can be configured independently, providing built-in bus-width conversion.

Table 18 illustrates the different configurations available on ports A and B.

If both ports are configured in either 2K x 9-bit, 1K x 18-bit, or 512 x 36-bit configurations, the 18 Kb block is accessible from port A or B. If both ports are configured in either 16K x 1-bit, 8K x 2-bit, or 4K x 4-bit configurations, the 16 K-bit block is accessible from Port A or Port B. All other configurations result in one port having access to an 18 Kb memory block and the other port having access to a 16 K-bit subset of the memory block equal to 16 Kbs.

Table 18: Dual-Port Mode Configurations

Port A	16K x 1	16K x 1	16K x 1	16K x 1	16K x 1	16K x 1
Port B	16K x 1	8K x 2	4K x 4	2K x 9	1K x 18	512 x 36
Port A	8K x 2	8K x 2	8K x 2	8K x 2	8K x 2	
Port B	8K x 2	4K x 4	2K x 9	1K x 18	512 x 36	
Port A	4K x 4	4K x 4	4K x 4	4K x 4		
Port B	4K x 4	2K x 9	1K x 18	512 x 36		
Port A	2K x 9	2K x 9	2K x 9			
Port B	2K x 9	1K x 18	512 x 36			
Port A	1K x 18	1K x 18				
Port B	1K x 18	512 x 36				
Port A	512 x 36					
Port B	512 x 36					

Each block SelectRAM+ cell is a fully synchronous memory, as illustrated in **Figure 38**. The two ports have independent inputs and outputs and are independently clocked.

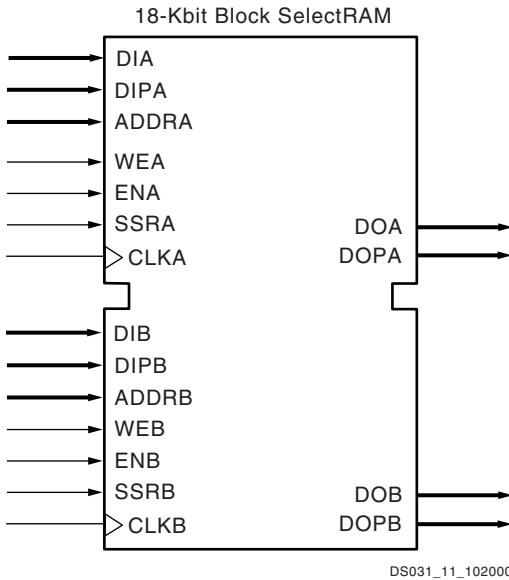


Figure 38: 18 Kb Block SelectRAM+ in Dual-Port Mode

Port Aspect Ratios

Table 19 shows the depth and the width aspect ratios for the 18 Kb block SelectRAM+ resource. Virtex-II Pro block SelectRAM+ also includes dedicated routing resources to provide an efficient interface with CLBs, block SelectRAM+, and multipliers.

Table 19: 18 Kb Block SelectRAM+ Port Aspect Ratio

Width	Depth	Address Bus	Data Bus	Parity Bus
1	16,384	ADDR[13:0]	DATA[0]	N/A
2	8,192	ADDR[12:0]	DATA[1:0]	N/A
4	4,096	ADDR[11:0]	DATA[3:0]	N/A
9	2,048	ADDR[10:0]	DATA[7:0]	Parity[0]
18	1,024	ADDR[9:0]	DATA[15:0]	Parity[1:0]
36	512	ADDR[8:0]	DATA[31:0]	Parity[3:0]

Read/Write Operations

The Virtex-II Pro block SelectRAM+ read operation is fully synchronous. An address is presented, and the read operation is enabled by control signal EN A or EN B. Then, depending on clock polarity, a rising or falling clock edge causes the stored data to be loaded into output registers.

The write operation is also fully synchronous. Data and address are presented, and the write operation is enabled by control signals WEA and WEB in addition to EN A or EN B. Then, again depending on the clock input mode, a ris-

ing or falling clock edge causes the data to be loaded into the memory cell addressed.

A write operation performs a simultaneous read operation. Three different options are available, selected by configuration:

1. WRITE_FIRST

The WRITE_FIRST option is a transparent mode. The same clock edge that writes the data input (DI) into the memory also transfers DI into the output registers DO, as shown in **Figure 39**.

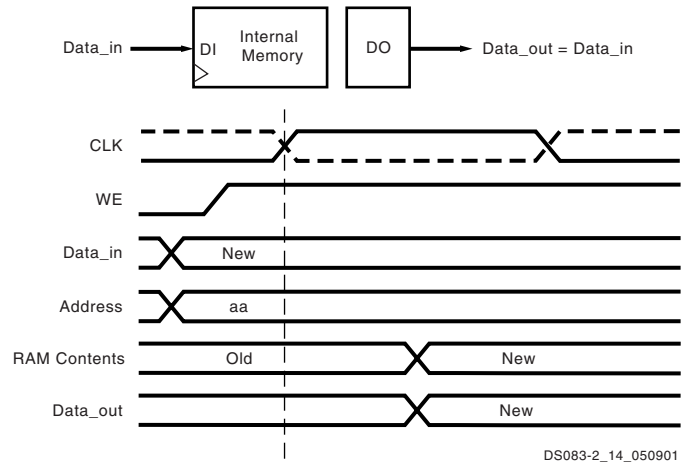


Figure 39: WRITE_FIRST Mode

2. READ_FIRST

The READ_FIRST option is a read-before-write mode.

The same clock edge that writes data input (DI) into the memory also transfers the prior content of the memory cell addressed into the data output registers DO, as shown in **Figure 40**.

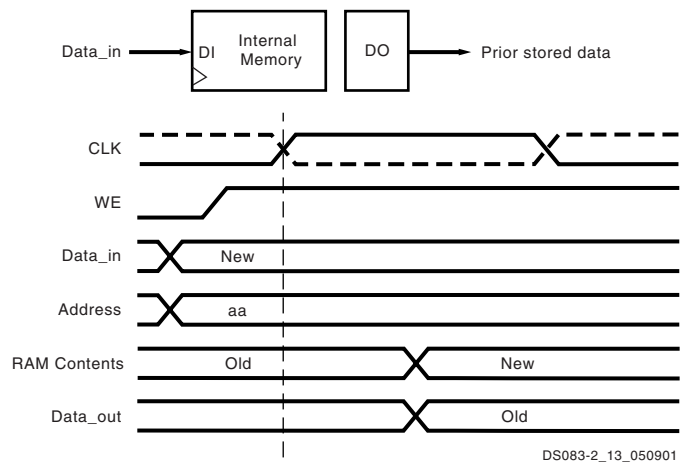


Figure 40: READ_FIRST Mode

3. NO_CHANGE

The NO_CHANGE option maintains the content of the output registers, regardless of the write operation. The clock edge during the write mode has no effect on the content of the data output register DO. When the port is configured as NO_CHANGE, only a read operation loads a new value in the output register DO, as shown in Figure 41.

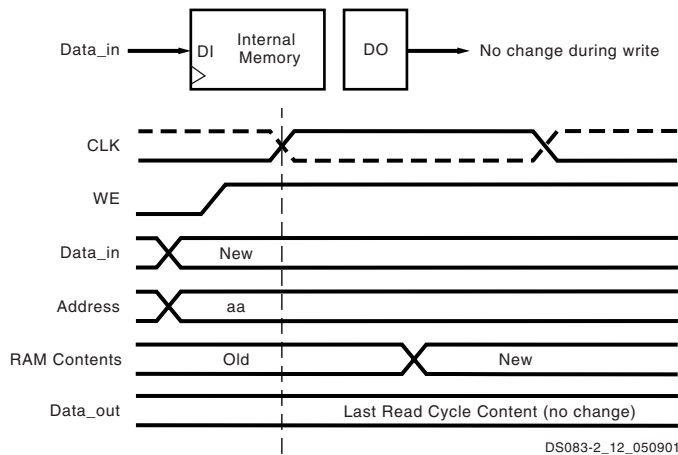


Figure 41: NO_CHANGE Mode

Control Pins and Attributes

Virtex-II Pro SelectRAM+ memory has two independent ports with the control signals described in Table 20. All control inputs including the clock have an optional inversion.

Table 20: Control Functions

Control Signal	Function
CLK	Read and Write Clock
EN	Enable affects Read, Write, Set, Reset
WE	Write Enable
SSR	Set DO register to SRVAL (attribute)

Initial memory content is determined by the INIT_x attributes. Separate attributes determine the output register value after device configuration (INIT) and SSR is asserted (SRVAL). Both attributes (INIT_B and SRVAL) are available for each port when a block SelectRAM+ resource is configured as dual-port RAM.

Total Amount of SelectRAM+ Memory

Virtex-II Pro SelectRAM+ memory blocks are organized in multiple columns. The number of blocks per column depends on the row size, the number of Processor Blocks, and the number of RocketIO transceivers.

Table 21 shows the number of columns as well as the total amount of block SelectRAM+ memory available for each Virtex-II Pro device. The 18 Kb SelectRAM+ blocks are cascadable to implement deeper or wider single- or dual-port memory resources.

Table 21: Virtex-II Pro SelectRAM+ Memory Available

Device	Columns	Total SelectRAM+ Memory		
		Blocks	in Kb	in Bits
XC2VP2	4	12	216	221,184
XC2VP4	4	28	504	516,096
XC2VP7	6	44	792	811,008
XC2VP20	8	88	1,584	1,622,016
XC2VP30	8	136	2,448	2,506,752
XC2VP40	10	192	3,456	3,538,944
XC2VP50	12	232	4,176	4,276,224
XC2VP70	14	328	5,904	6,045,696
XC2VP100	16	444	7,992	8,183,808
XC2VP125	18	556	10,008	10,248,192

Figure 42 shows the layout of the block RAM columns in the XC2VP4 device.

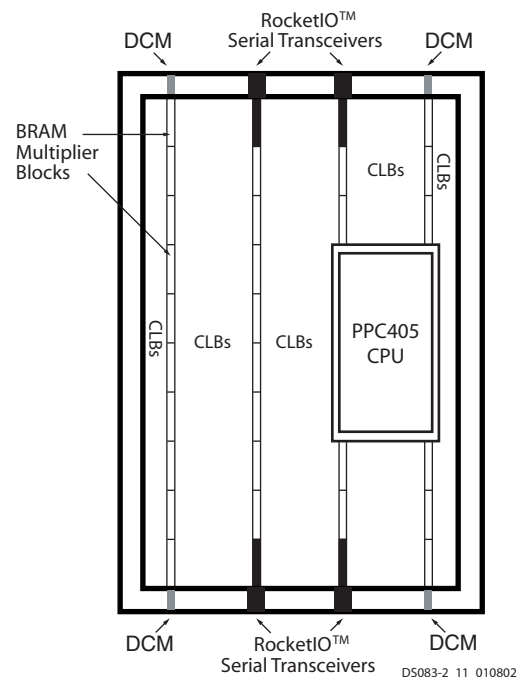


Figure 42: XC2VP4 Block RAM Column Layout

18-Bit x 18-Bit Multipliers

Introduction

A Virtex-II Pro multiplier block is an 18-bit by 18-bit 2's complement signed multiplier. Virtex-II Pro devices incorporate many embedded multiplier blocks. These multipliers can be associated with an 18 Kb block SelectRAM+ resource or can be used independently. They are optimized for high-speed operations and have a lower power consumption compared to an 18-bit x 18-bit multiplier in slices.

Each SelectRAM+ memory and multiplier block is tied to four switch matrices, as shown in Figure 43.

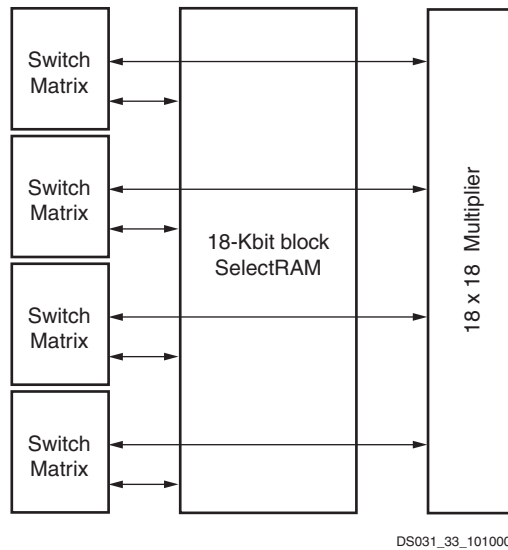


Figure 43: SelectRAM+ and Multiplier Blocks

Association With Block SelectRAM+ Memory

The interconnect is designed to allow SelectRAM+ memory and multiplier blocks to be used at the same time, but some interconnect is shared between the SelectRAM+ and the multiplier. Thus, SelectRAM+ memory can be used only up to 18 bits wide when the multiplier is used, because the multiplier shares inputs with the upper data bits of the SelectRAM+ memory.

This sharing of the interconnect is optimized for an 18-bit-wide block SelectRAM+ resource feeding the multiplier. The use of SelectRAM+ memory and the multiplier with an accumulator in LUTs allows for implementation of a digital signal processor (DSP) multiplier-accumulator (MAC) function, which is commonly used in finite and infinite impulse response (FIR and IIR) digital filters.

Configuration

The multiplier block is an 18-bit by 18-bit signed multiplier (2's complement). Both A and B are 18-bit-wide inputs, and the output is 36 bits. Figure 44 shows a multiplier block.

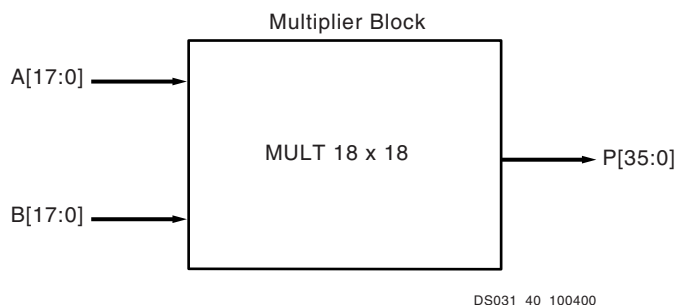


Figure 44: Multiplier Block

Locations / Organization

Multiplier organization is identical to the 18 Kb SelectRAM+ organization, because each multiplier is associated with an 18 Kb block SelectRAM+ resource.

Table 22: Multiplier Resources

Device	Columns	Total Multipliers
XC2VP2	4	12
XC2VP4	4	28
XC2VP7	6	44
XC2VP20	8	88
XC2VP30	8	136
XC2VP40	10	192
XC2VP50	12	232
XC2VP70	14	328
XC2VP100	16	444
XC2VP125	18	556

In addition to the built-in multiplier blocks, the CLB elements have dedicated logic to implement efficient multipliers in logic. (Refer to **Configurable Logic Blocks (CLBs)**, page 23).

Global Clock Multiplexer Buffers

Virtex-II Pro devices have 16 clock input pins that can also be used as regular user I/Os. Eight clock pads center on both the top edge and the bottom edge of the device, as illustrated in Figure 45.

The global clock multiplexer buffer represents the input to dedicated low-skew clock tree distribution in Virtex-II Pro devices. Like the clock pads, eight global clock multiplexer buffers are on the top edge of the device and eight are on the bottom edge.

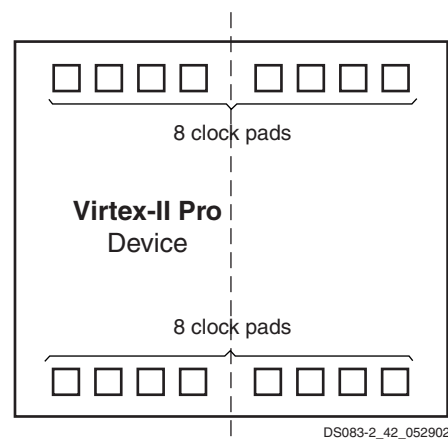


Figure 45: Virtex-II Pro Clock Pads

Each global clock multiplexer buffer can be driven either by the clock pad to distribute a clock directly to the device, or by the Digital Clock Manager (DCM), discussed in **Digital Clock Manager (DCM)**, page 39. Each global clock multi-

plexer buffer can also be driven by local interconnects. The DCM has clock output(s) that can be connected to global clock multiplexer buffer inputs, as shown in **Figure 46**.

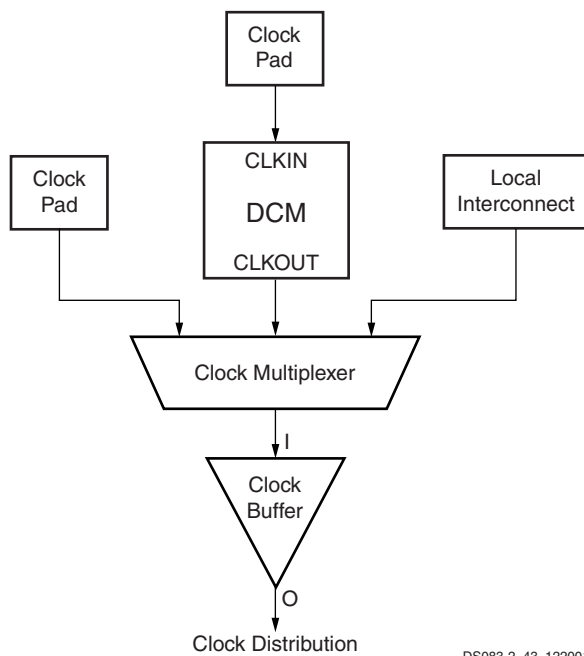


Figure 46: Virtex-II Pro Clock Multiplexer Buffer Configuration

Global clock buffers are used to distribute the clock to some or all synchronous logic elements (such as registers in CLBs and IOBs, and SelectRAM+ blocks).

Eight global clocks can be used in each quadrant of the Virtex-II Pro device. Designers should consider the clock distribution detail of the device prior to pin-locking and floor-planning. (See the *Virtex-II Pro Platform FPGA User Guide*.)

Figure 47 shows clock distribution in Virtex-II Pro devices.

In each quadrant, up to eight clocks are organized in clock rows. A clock row supports up to 16 CLB rows (eight up and eight down).

To reduce power consumption, any unused clock branches remain static.

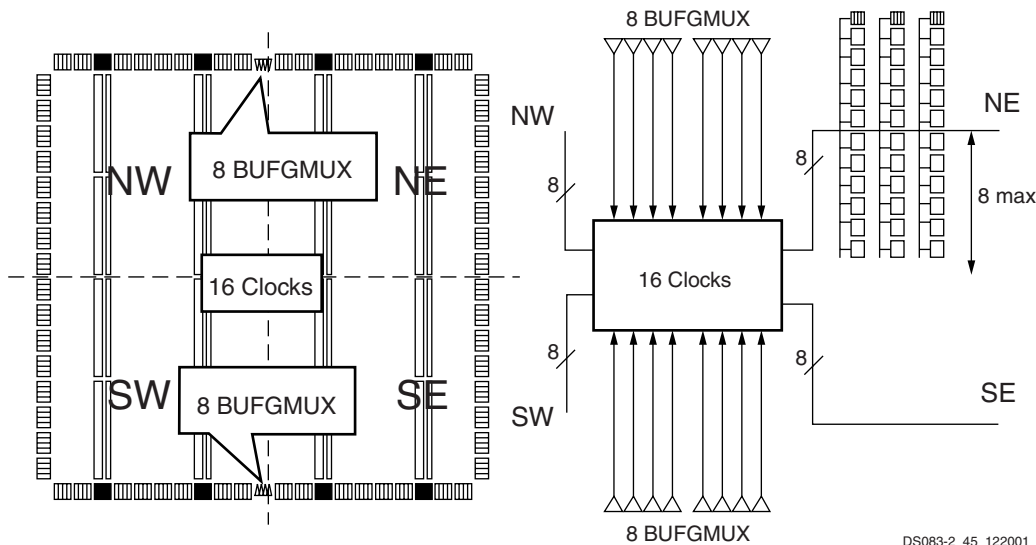


Figure 47: Virtex-II Pro Clock Distribution

Global clocks are driven by dedicated clock buffers (BUFG), which can also be used to gate the clock (BUFGCE) or to multiplex between two independent clock inputs (BUFGMUX).

The most common configuration option of this element is as a buffer. A BUFG function in this (global buffer) mode, is shown in **Figure 48**.

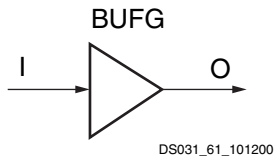


Figure 48: Virtex-II Pro BUFG Function

The Virtex-II Pro global clock buffer BUFG can also be configured as a clock enable/disable circuit (**Figure 49**), as well as a two-input clock multiplexer (**Figure 50**). A functional description of these two options is provided below. Each of them can be used in either of two modes, selected by configuration: rising clock edge or falling clock edge.

This section describes the rising clock edge option. For the opposite option, falling clock edge, just change all "rising" references to "falling" and all "High" references to "Low", except for the description of the CE and S levels. The rising clock edge option uses the BUFGCE and BUFGMUX primitives. The falling clock edge option uses the BUFGCE_1 and BUFGMUX_1 primitives.

BUFGCE

If the CE input is active (High) prior to the incoming rising clock edge, this Low-to-High-to-Low clock pulse passes through the clock buffer. Any level change of CE during the incoming clock High time has no effect.

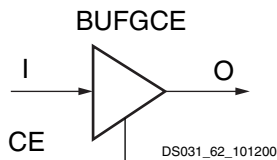


Figure 49: Virtex-II Pro BUFGCE Function

If the CE input is inactive (Low) prior to the incoming rising clock edge, the following clock pulse does not pass through the clock buffer, and the output stays Low. Any level change of CE during the incoming clock High time has no effect. CE must not change during a short setup window just prior to the rising clock edge on the BUFGCE input I. Violating this setup time requirement can result in an undefined runt pulse output.

BUFGMUX

BUFGMUX can switch between two unrelated, even asynchronous clocks. Basically, a Low on S selects the I₀ input, a High on S selects the I₁ input. Switching from one clock to the other is done in such a way that the output High and Low

time is never shorter than the shortest High or Low time of either input clock. As long as the presently selected clock is High, any level change of S has no effect.

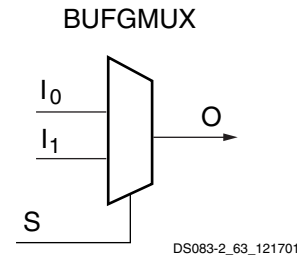


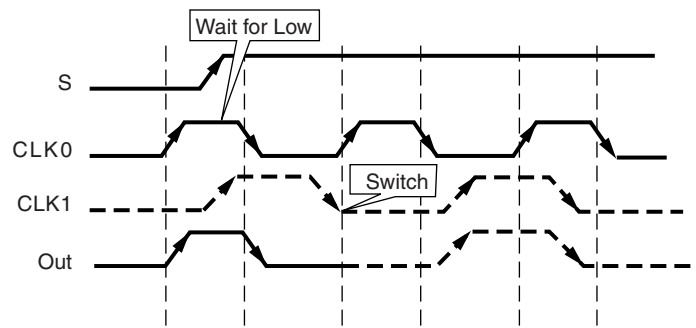
Figure 50: Virtex-II Pro BUFGMUX Function

If the presently selected clock is Low while S changes, or if it goes Low after S has changed, the output is kept Low until the other ("to-be-selected") clock has made a transition from High to Low. At that instant, the new clock starts driving the output.

The two clock inputs can be asynchronous with regard to each other, and the S input can change at any time, except for a short setup time prior to the rising edge of the presently selected clock; that is, prior to the rising edge of the BUFGMUX output O. Violating this setup time requirement can result in an undefined runt pulse output.

All Virtex-II Pro devices have 16 global clock multiplexer buffers.

Figure 51 shows a switchover from CLK0 to CLK1.



DS083-2_46_121701

Figure 51: Clock Multiplexer Waveform Diagram

- The current clock is CLK0.
- S is activated **High (setup is required before the next negative CLK0 edge)**High.
- If CLK0 is currently High, the multiplexer waits for **the next negative edge**CLK0 to go Low.
- Once CLK0 is Low, the multiplexer output stays **Low,** Low until CLK1 **goes** transitions High to Low.
- When CLK1 transitions from High to Low, the output switches to CLK1.
- No glitches or short pulses can appear on the output.

Digital Clock Manager (DCM)

The Virtex-II Pro DCM offers a wide range of powerful clock management features.

- **Clock De-skew:** The DCM generates new system clocks (either internally or externally to the FPGA), which are phase-aligned to the input clock, thus eliminating clock distribution delays.
- **Frequency Synthesis:** The DCM generates a wide range of output clock frequencies, performing very flexible clock multiplication and division.
- **Phase Shifting:** The DCM provides both coarse phase shifting and fine-grained phase shifting with dynamic phase shift control.

The DCM utilizes fully digital delay lines allowing robust high-precision control of clock phase and frequency. It also utilizes fully digital feedback systems, operating dynamically to compensate for temperature and voltage variations during operation.

Up to four of the nine DCM clock outputs can drive inputs to global clock buffers or global clock multiplexer buffers simultaneously (see [Figure 52](#)). All DCM clock outputs can simultaneously drive general routing resources, including routes to output buffers.

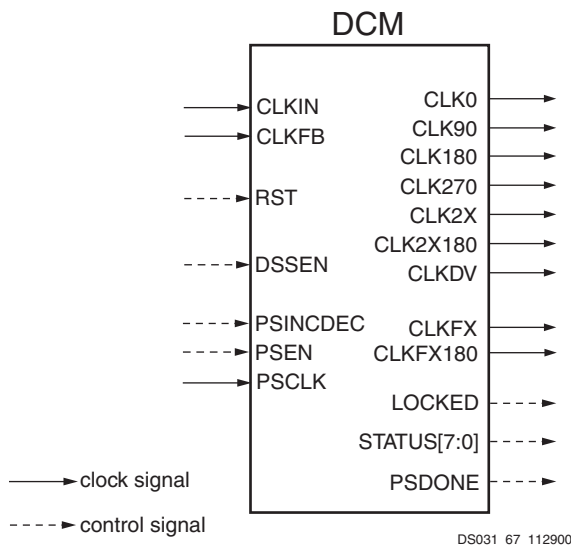


Figure 52: Digital Clock Manager

The DCM can be configured to delay the completion of the Virtex-II Pro configuration process until after the DCM has achieved lock. This guarantees that the chip does not begin operating until after the system clocks generated by the DCM have stabilized.

The DCM has the following general control signals:

- **RST** input pin: resets the entire DCM
- **LOCKED** output pin: asserted High when all enabled DCM circuits have locked.
- **STATUS** output pins (active High): shown in [Table 23](#).

Table 23: DCM Status Pins

Status Pin	Function
0	Phase Shift Overflow
1	CLKIN Stopped
2	CLKFX Stopped
3	N/A
4	N/A
5	N/A
6	N/A
7	N/A

Clock De-skew

The DCM de-skews the output clocks relative to the input clock by automatically adjusting a digital delay line. Additional delay is introduced so that clock edges arrive at internal registers and block RAMs simultaneously with the clock edges arriving at the input clock pad. Alternatively, external clocks, which are also de-skewed relative to the input clock, can be generated for board-level routing. All DCM output clocks are phase-aligned to CLK0 and, therefore, are also phase-aligned to the input clock.

To achieve clock de-skew, the CLKFB input must be connected, and its source must be either CLK0 or CLK2X. Note that CLKFB must always be connected, unless only the CLKFX or CLKFX180 outputs are used and de-skew is not required.

Frequency Synthesis

The DCM provides flexible methods for generating new clock frequencies. Each method has a different operating frequency range and different AC characteristics. The CLK2X and CLK2X180 outputs double the clock frequency. The CLKDV output creates divided output clocks with division options of 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5, 5.5, 6, 6.5, 7, 7.5, 8, 9, 10, 11, 12, 13, 14, 15, and 16.

The CLKFX and CLKFX180 outputs can be used to produce clocks at the following frequency:

$$FREQ_{CLKFX} = (M/D) \cdot FREQ_{CLKIN}$$

where M and D are two integers. Specifications for M and D are provided under **DCM Timing Parameters** in [Data Sheet Module 3](#). By default, $M = 4$ and $D = 1$, which results in a clock output frequency four times faster than the clock input frequency (CLKIN).

CLK2X180 is phase shifted 180 degrees relative to CLK2X. CLKFX180 is phase shifted 180 degrees relative to CLKFX. All frequency synthesis outputs automatically have 50/50 duty cycles, with the exception of the CLKDV output when performing a non-integer divide in high-frequency mode. See [Table 24](#) for more details.

Note that CLK2X and CLK2X180 are not available in high-frequency mode.

Table 24: CLKDV Duty Cycle for Non-integer Divides

CLKDV_DIVIDE	Duty Cycle
1.5	1/3
2.5	2/5
3.5	3/7
4.5	4/9
5.5	5/11
6.5	6/13
7.5	7/15

Phase Shifting

The DCM provides additional control over clock skew through either coarse or fine-grained phase shifting. The CLK0, CLK90, CLK180, and CLK270 outputs are each phase shifted by ¼ of the input clock period relative to each other, providing coarse phase control. Note that CLK90 and CLK270 are not available in high-frequency mode.

Fine-phase adjustment affects all nine DCM output clocks. When activated, the phase shift between the rising edges of CLKIN and CLKFB is a specified fraction of the input clock period.

In variable mode, the PHASE_SHIFT value can also be dynamically incremented or decremented as determined by PSINCDEC synchronously to PSCLK, when the PSEN input is active. Figure 53 illustrates the effects of fine-phase shifting. For more information on DCM features, see the *Virtex-II Pro Platform FPGA User Guide*.

Table 25 lists fine-phase shifting control pins, when used in variable mode.

Table 25: Fine Phase Shifting Control Pins

Control Pin	Direction	Function
PSINCDEC	In	Increment or decrement
PSEN	In	Enable ± phase shift
PSCLK	In	Clock for phase shift
PSDONE	Out	Active when completed

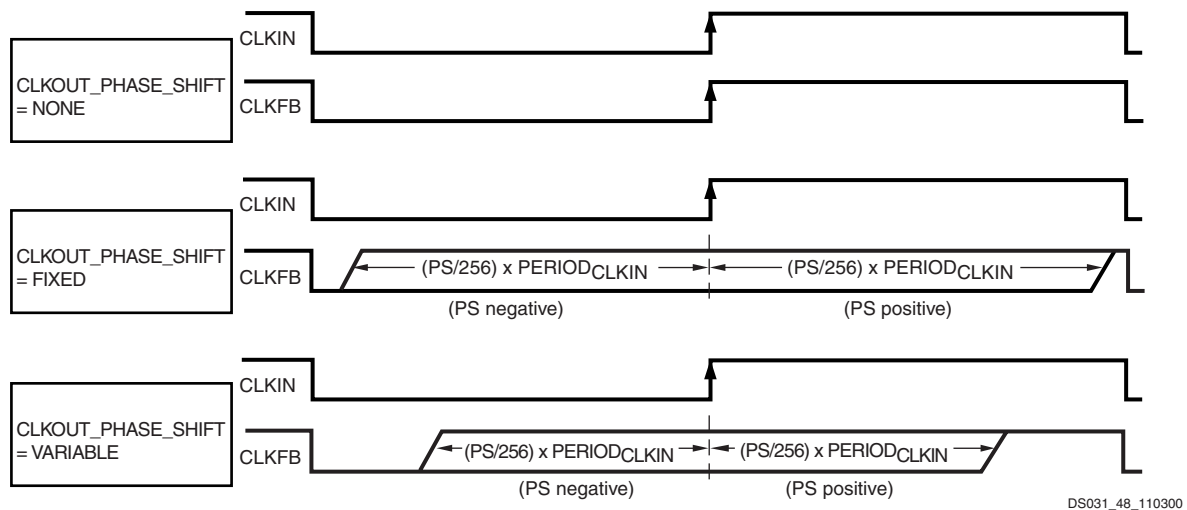


Figure 53: Fine-Phase Shifting Effects

Two separate components of the phase shift range must be understood:

- PHASE_SHIFT attribute range
- FINE_SHIFT_RANGE DCM timing parameter range

The PHASE_SHIFT attribute is the numerator in the following equation:

$$\text{Phase Shift (ns)} = (\text{PHASE_SHIFT}/256) * \text{PERIOD}_{\text{CLKIN}}$$

The full range of this attribute is always -255 to +255, but its practical range varies with CLKIN frequency, as constrained by the FINE_SHIFT_RANGE component, which represents the total delay achievable by the phase shift delay line. Total

delay is a function of the number of delay taps used in the circuit. Across process, voltage, and temperature, this absolute range is guaranteed to be as specified under **DCM Timing Parameters** in [Data Sheet Module 3](#).

Absolute range (fixed mode) = ± FINE_SHIFT_RANGE

Absolute range (variable mode) = ± FINE_SHIFT_RANGE/2

The reason for the difference between fixed and variable modes is as follows. For variable mode to allow symmetric, dynamic sweeps from -255/256 to +255/256, the DCM sets the "zero phase skew" point as the middle of the delay line, thus dividing the total delay line range in half. In fixed mode, since the PHASE_SHIFT value never changes after configu-

ration, the entire delay line is available for insertion into either the CLKIN or CLKFB path (to create either positive or negative skew).

Taking both of these components into consideration, the following are some usage examples:

- If $PERIOD_{CLKIN} = 2 * FINE_SHIFT_RANGE$, then PHASE_SHIFT in fixed mode is limited to ± 128 , and in variable mode it is limited to ± 64 .
- If $PERIOD_{CLKIN} = FINE_SHIFT_RANGE$, then PHASE_SHIFT in fixed mode is limited to ± 255 , and in variable mode it is limited to ± 128 .

Table 26: DCM Frequency Ranges

Output Clock	Low-Frequency Mode		High-Frequency Mode	
	CLKIN Input	CLK Output	CLKIN Input	CLK Output
CLK0, CLK180	CLKIN_FREQ_DLL_LF	CLKOUT_FREQ_1X_LF	CLKIN_FREQ_DLL_HF	CLKOUT_FREQ_1X_HF
CLK90, CLK270	CLKIN_FREQ_DLL_LF	CLKOUT_FREQ_1X_LF	NA	NA
CLK2X, CLK2X180	CLKIN_FREQ_DLL_LF	CLKOUT_FREQ_2X_LF	NA	NA
CLKDV	CLKIN_FREQ_DLL_LF	CLKOUT_FREQ_DV_LF	CLKIN_FREQ_DLL_HF	CLKOUT_FREQ_DV_HF
CLKFX, CLKFX180	CLKIN_FREQ_FX_LF	CLKOUT_FREQ_FX_LF	CLKIN_FREQ_FX_HF	CLKOUT_FREQ_FX_HF

CLK90, and CLK270 outputs are not available in high-frequency mode.

High or low-frequency mode is selected by an attribute.

Routing

DCM and MGT Locations/Organization

Virtex-II Pro DCMs and serial transceivers (MGTS) are placed on the top and bottom of each block RAM and multiplier column in some combination, as shown in Table 27. The number of DCMs and RocketIO transceivers total twice the number of block RAM columns in the device. Refer to Figure 42, page 35 for an illustration of this in the XC2VP4 device.

Place-and-route software takes advantage of this regular array to deliver optimum system performance and fast compile times. The segmented routing resources are essential to guarantee IP cores portability and to efficiently handle an incremental design flow that is based on modular implementations. Total design time is reduced due to fewer and shorter design iterations.

Hierarchical Routing Resources

Most Virtex-II Pro signals are routed using the global routing resources, which are located in horizontal and vertical routing channels between each switch matrix.

As shown in Figure 54, page 42, Virtex-II Pro has fully buffered programmable interconnections, with a number of resources counted between any two adjacent switch matrix

- If $PERIOD_{CLKIN} \leq 0.5 * FINE_SHIFT_RANGE$, then PHASE_SHIFT is limited to ± 255 in either mode.

Operating Modes

The frequency ranges of DCM input and output clocks depend on the operating mode specified, either low-frequency mode or high-frequency mode, according to Table 26. For actual values, see [Virtex-II Pro Switching Characteristics \(Module 3\)](#). The CLK2X, CLK2X180,

rows or columns. Fanout has minimal impact on the performance of each net.

Table 27: DCM Organization

Device	Block RAM Columns	DCMs	MGTS
XC2VP2	4	4	4
XC2VP4	4	4	4
XC2VP7	6	4	8
XC2VP20	8	8	8
XC2VP30	8	8	8
XC2VP40	10	8	12
XC2VP50	12	8	16
XC2VP70	14	8	20
XC2VP100	16	12	20
XC2VP125	18	12	24

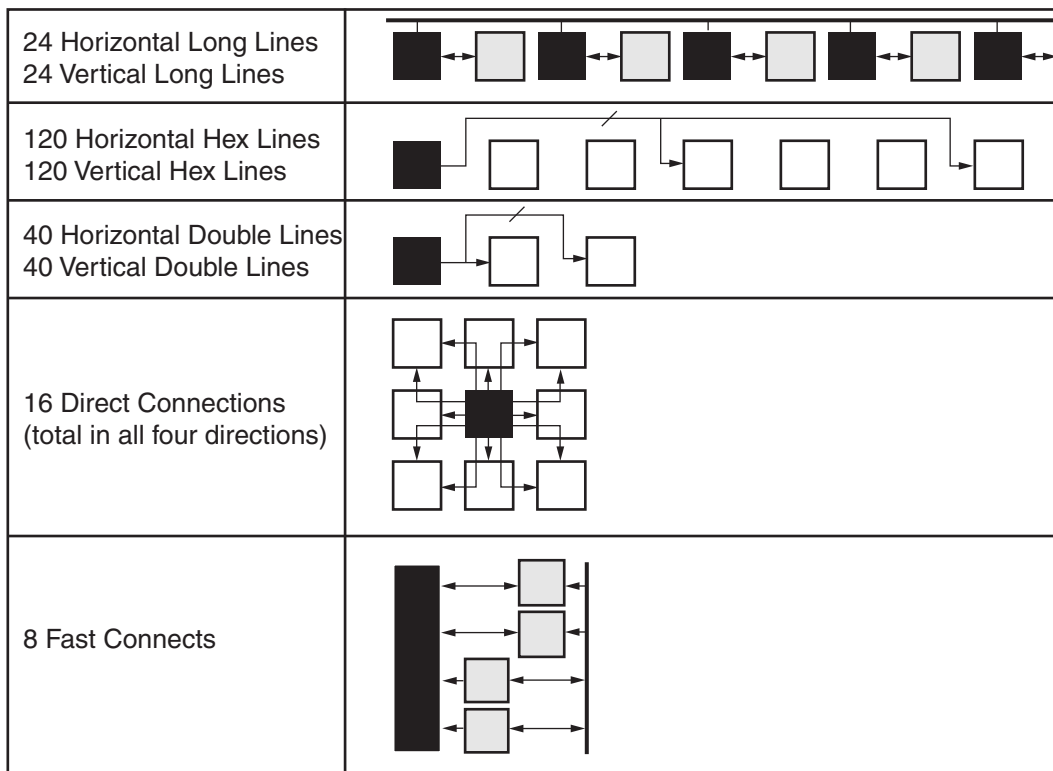
- The long lines are bidirectional wires that distribute signals across the device. Vertical and horizontal long lines span the full height and width of the device.
- The hex lines route signals to every third or sixth block away in all four directions. Organized in a staggered pattern, hex lines can only be driven from one end. Hex-line signals can be accessed either at the endpoints or at the midpoint (three blocks from the source).

- The double lines route signals to every first or second block away in all four directions. Organized in a staggered pattern, double lines can be driven only at their endpoints. Double-line signals can be accessed either at the endpoints or at the midpoint (one block from the source).
- The direct connect lines route signals to neighboring blocks: vertically, horizontally, and diagonally.
- The fast connect lines are the internal CLB local interconnections from LUT outputs to LUT inputs.
- Horizontal routing resources are provided for on-chip 3-state buses. Four partitionable bus lines are provided per CLB row, permitting multiple buses within a row. (See **3-State Buffers**, page 31.)
- Two dedicated carry-chain resources per slice column (two per CLB column) propagate carry-chain MUXCY output signals vertically to the adjacent slice. (See **CLB/Slice Configurations**, page 31.)
- One dedicated SOP chain per slice row (two per CLB row) propagate ORCY output logic signals horizontally to the adjacent slice. (See **Sum of Products**, page 30.)
- One dedicated shift-chain per CLB connects the output of LUTs in shift-register mode to the input of the next LUT in shift-register mode (vertically) inside the CLB. (See **Shift Registers**, page 27.)

Dedicated Routing

In addition to the global and local routing resources, dedicated signals are available.

- There are eight global clock nets per quadrant. (See **Global Clock Multiplexer Buffers**, page 36.)



DS031_60_110200

Figure 54: Hierarchical Routing Resources

Configuration

Virtex-II Pro devices are configured by loading application specific configuration data into the internal configuration memory. Configuration is carried out using a subset of the device pins, some of which are dedicated, while others can be re-used as general purpose inputs and outputs once configuration is complete.

Depending on the system design, several configuration modes are supported, selectable via mode pins. The mode pins M2, M1 and M0 are dedicated pins. An additional pin,

HSWAP_EN is used in conjunction with the mode pins to select whether user I/O pins have pull-ups during configuration. By default, HSWAP_EN is tied High (internal pull-up) which shuts off the pull-ups on the user I/O pins during configuration. When HSWAP_EN is tied Low, user I/Os have pull-ups during configuration. Other dedicated pins are CCLK (the configuration clock pin), DONE, PROG_B, and the boundary-scan pins: TDI, TDO, TMS, and TCK. (The TDO pin is open-drain and does not have an internal pullup resistor.) Depending on the configuration mode chosen, CCLK can be an output generated by the FPGA, or an input

accepting an externally generated clock. The configuration pins and boundary scan pins are independent of the V_{CCO} . The auxiliary power supply (V_{CCAUX}) of 2.5V is used for these pins. All configuration pins are LVCMOS25 12mA. See [Virtex-II Pro Switching Characteristics \(Module 3\)](#).

Configuration Modes

A "persist" option is available which can be used to force the configuration pins to retain their configuration function even after device configuration is complete. If the persist option is not selected then the configuration pins with the exception of CCLK, PROG_B, and DONE can be used as user I/O in normal operation. The persist option does not apply to the boundary-scan related pins. The persist feature is valuable in applications which employ partial reconfiguration or reconfiguration on the fly.

Virtex-II Pro supports the following five configuration modes:

- **Slave-Serial Mode**
- **Master-Serial Mode**
- **Slave SelectMAP Mode**
- **Master SelectMAP Mode**
- **Boundary-Scan (JTAG, IEEE 1532) Mode**

Refer to [Table 28, page 44](#).

A detailed description of configuration modes is provided in the *Virtex-II Pro Platform FPGA User Guide*.

Slave-Serial Mode

In slave-serial mode, the FPGA receives configuration data in bit-serial form from a serial PROM or other serial source of configuration data. The CCLK pin on the FPGA is an input in this mode. The serial bitstream must be setup at the DIN input pin a short time before each rising edge of the externally generated CCLK.

Multiple FPGAs can be daisy-chained for configuration from a single source. After a particular FPGA has been configured, the data for the next device is routed internally to the DOUT pin. The data on the DOUT pin changes on the rising edge of CCLK.

Slave-serial mode is selected by applying [111] to the mode pins (M2, M1, M0). A weak pull-up on the mode pins makes slave serial the default mode if the pins are left unconnected.

Master-Serial Mode

In master-serial mode, the CCLK pin is an output pin. It is the Virtex-II Pro FPGA device that drives the configuration clock on the CCLK pin to a Xilinx Serial PROM which in turn feeds bit-serial data to the DIN input. The FPGA accepts this data on each rising CCLK edge. After the FPGA has been loaded, the data for the next device in a daisy-chain is presented on the DOUT pin after the rising CCLK edge.

The interface is identical to slave serial except that an internal oscillator is used to generate the configuration clock (CCLK). A wide range of frequencies can be selected for CCLK which always starts at a slow default frequency. Configuration bits then switch CCLK to a higher frequency for the remainder of the configuration.

Slave SelectMAP Mode

The SelectMAP mode is the fastest configuration option. Byte-wide data is written into the Virtex-II Pro FPGA device with a BUSY flag controlling the flow of data. An external data source provides a byte stream, CCLK, an active Low Chip Select (CS_B) signal and a Write signal (RDWR_B). If BUSY is asserted (High) by the FPGA, the data must be held until BUSY goes Low. Data can also be read using the SelectMAP mode. If RDWR_B is asserted, configuration data is read out of the FPGA as part of a readback operation.

After configuration, the pins of the SelectMAP port can be used as additional user I/O. Alternatively, the port can be retained to permit high-speed 8-bit readback using the persist option.

Multiple Virtex-II Pro FPGAs can be configured using the SelectMAP mode, and be made to start-up simultaneously. To configure multiple devices in this way, wire the individual CCLK, Data, RDWR_B, and BUSY pins of all the devices in parallel. The individual devices are loaded separately by deasserting the CS_B pin of each device in turn and writing the appropriate data.

Master SelectMAP Mode

This mode is a master version of the SelectMAP mode. The device is configured byte-wide on a CCLK supplied by the Virtex-II Pro FPGA device. Timing is similar to the Slave SerialMAP mode except that CCLK is supplied by the Virtex-II Pro FPGA.

Boundary-Scan (JTAG, IEEE 1532) Mode

In boundary-scan mode, dedicated pins are used for configuring the Virtex-II Pro device. The configuration is done entirely through the IEEE 1149.1 Test Access Port (TAP). Virtex-II Pro device configuration using Boundary scan is compliant with IEEE 1149.1-1993 standard and the new IEEE 1532 standard for In-System Configurable (ISC) devices. The IEEE 1532 standard is backward compliant with the IEEE 1149.1-1993 TAP and state machine. The IEEE Standard 1532 for In-System Configurable (ISC) devices is intended to be programmed, reprogrammed, or tested on the board via a physical and logical protocol. Configuration through the boundary-scan port is always available, independent of the mode selection. Selecting the boundary-scan mode simply turns off the other modes.

Table 28: Virtex-II Pro Configuration Mode Pin Settings

Configuration Mode ⁽¹⁾	M2	M1	M0	CCLK Direction	Data Width	Serial D _{OUT} ⁽²⁾
Master Serial	0	0	0	Out	1	Yes
Slave Serial	1	1	1	In	1	Yes
Master SelectMAP	0	1	1	Out	8	No
Slave SelectMAP	1	1	0	In	8	No
Boundary Scan	1	0	1	N/A	1	No

Notes:

1. The HSWAP_EN pin controls the pullups. Setting M2, M1, and M0 selects the configuration mode, while the HSWAP_EN pin controls whether or not the pullups are used.
2. Daisy chaining is possible only in modes where Serial D_{OUT} is used. For example, in SelectMAP modes, the first device does NOT support daisy chaining of downstream devices.

Table 29 lists the total number of bits required to configure each device.

Table 29: Virtex-II Pro Bitstream Lengths

Device	Number of Configuration Bits
XC2VP2	1,305,440
XC2VP4	3,006,560
XC2VP7	4,485,472
XC2VP20	8,214,624
XC2VP30	11,589,984
XC2VP40	15,868,256
XC2VP50	19,021,408
XC2VP70	26,099,040
XC2VP100	34,292,832
XC2VP125	43,602,784

Configuration Sequence

The configuration of Virtex-II Pro devices is a three-phase process. First, the configuration memory is cleared. Next, configuration data is loaded into the memory, and finally, the logic is activated by a start-up process.

Configuration is automatically initiated on power-up unless it is delayed by the user. The INIT_B pin can be held Low using an open-drain driver. An open-drain is required since INIT_B is a bidirectional open-drain pin that is held Low by a Virtex-II Pro FPGA device while the configuration memory is being cleared. Extending the time that the pin is Low causes the configuration sequencer to wait. Thus, configuration is delayed by preventing entry into the phase where data is loaded.

The configuration process can also be initiated by asserting the PROG_B pin. The end of the memory-clearing phase is signaled by the INIT_B pin going High, and the completion

of the entire process is signaled by the DONE pin going High. The Global Set/Reset (GSR) signal is pulsed after the last frame of configuration data is written but before the start-up sequence. The GSR signal resets all flip-flops on the device.

The default start-up sequence is that one CCLK cycle after DONE goes High, the global 3-state signal (GTS) is released. This permits device outputs to turn on as necessary. One CCLK cycle later, the Global Write Enable (GWE) signal is released. This permits the internal storage elements to begin changing state in response to the logic and the user clock.

The relative timing of these events can be changed via configuration options in software. In addition, the GTS and GWE events can be made dependent on the DONE pins of multiple devices all going High, forcing the devices to start synchronously. The sequence can also be paused at any stage, until lock has been achieved on any or all DCMs, as well as DCI.

Readback

In this mode, configuration data from the Virtex-II Pro FPGA device can be read back. Readback is supported only in the SelectMAP (master and slave) and Boundary Scan mode.

Along with the configuration data, it is possible to read back the contents of all registers, distributed SelectRAM+, and block RAM resources. This capability is used for real-time debugging. For more detailed configuration information, see the *Virtex-II Pro Platform FPGA User Guide*.

Bitstream Encryption

Virtex-II Pro devices have an on-chip decryptor using one or two sets of three keys for triple-key Data Encryption Standard (DES) operation. Xilinx software tools offer an optional encryption of the configuration data (bitstream) with a triple-key DES determined by the designer.

The keys are stored in the FPGA by JTAG instruction and retained by a battery connected to the V_{BATT} pin, when the device is not powered. Virtex-II Pro devices can be config-

ured with the corresponding encrypted bitstream, using any of the configuration modes described previously.

A detailed description of how to use bitstream encryption is provided in the *Virtex-II Pro Platform FPGA User Guide*. Your local FAE can also provide specific information on this feature.

Partial Reconfiguration

Partial reconfiguration of Virtex-II Pro devices can be accomplished in either Slave SelectMAP mode or Boundary-Scan mode. Instead of resetting the chip and doing a full configuration, new data is loaded into a specified area of

the chip, while the rest of the chip remains in operation. Data is loaded on a column basis, with the smallest load unit being a configuration “frame” of the bitstream (device size dependent).

Partial reconfiguration is useful for applications that require different designs to be loaded into the same area of a chip, or that require the ability to change portions of a design without having to reset or reconfigure the entire chip.

For more information on Partial Reconfiguration in Virtex-II Pro devices, please refer to Xilinx Application Note [XAPP290](#), *Two Flows for Partial Reconfiguration*.

Revision History

This section records the change history for this module of the data sheet.

Date	Version	Revision
01/31/02	1.0	Initial Xilinx release.
06/13/02	2.0	New Virtex-II Pro family members. New timing parameters per speedsfile v1.62 .
09/03/02	2.1	<ul style="list-style-type: none"> Revised Reset and Power sections. Updated Table 8, which lists compatible input standards. Added Figure 19, Figure 20, and Figure 21, which provide examples illustrating the use of I/O standards.
09/27/02	2.2	<ul style="list-style-type: none"> In section Overview, corrected max number of MGTs from 16 to 24. In section Input/Output Blocks (IOBs), added references to XAPP653 regarding implementation of 3.3V I/O standards.
11/20/02	2.3	<ul style="list-style-type: none"> Table 3: Add rows for LVTTTL, LVCMOS33, and PCI-X. Table 8: Added LVTTTL and LVCMOS33 to compatible 3.3V cells. Table 29: Correct bitstream lengths.
12/03/02	2.4	<ul style="list-style-type: none"> Added mention of LVTTTL and PCI with respect to SelectIO-Ultra configurations. See section Input/Output Individual Options and Figure 13.
01/20/03	2.5	<ul style="list-style-type: none"> Added qualification to features vs. Virtex-II (open-drain output pin TDO does not have internal pullup resistor) Table 7: Added HSTL18 (I, II, III, & IV) and HSTL18_DCI (I,II, III & IV) to 1.8V V_{CC0} row. Table 8: Numerous revisions.

Virtex-II Pro Data Sheet

The Virtex-II Pro Data Sheet contains the following modules:

- [Virtex-II Pro™ Platform FPGAs: Introduction and Overview \(Module 1\)](#)
- [Virtex-II Pro™ Platform FPGAs: Detailed Description \(Module 2\)](#)
- [Virtex-II Pro™ Platform FPGAs: DC and Switching Characteristics \(Module 3\)](#)
- [Virtex-II Pro™ Platform FPGAs: Pinout Information \(Module 4\)](#)

