THE DATASHEET OF

EP4SGX290KF40C3NAB

Stratix IV Device Handbook

Volume 1

101 Innovation Drive
San Jose, CA 95134
www.altera.com

SIV5V1-4.0

Contents

Chapter 3. TriMatrix Embedded Memory Blocks in Stratix IV Devices

Chapter 4. DSP Blocks in Stratix IV Devices

Chapter 5. Clock Networks and PLLs in Stratix IV Devices

Section II. I/O Interfaces

Chapter 6. I/O Features in Stratix IV Devices

Chapter 7. External Memory Interfaces in Stratix IV Devices

Chapter 8. High-Speed Differential I/O Interfaces and DPA in Stratix IV Devices

Section III. System Integration

Chapter 9. Hot Socketing and Power-On Reset in Stratix IV Devices

Chapter 10. Configuration, Design Security, and Remote System Upgrades in Stratix IV Devices

Chapter 11. SEU Mitigation in Stratix IV Devices

Chapter 12. JTAG Boundary-Scan Testing in Stratix IV Devices

Chapter 13. Power Management in Stratix IV Devices

Chapter Revision Dates

The chapters in this book, *Stratix IV Device Handbook Volume 1*, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1 Stratix IV Device Family Overview
Revised: *November 2009*
Part Number: *SIV51001-3.0*

Chapter 2 Logic Array Blocks and Adaptive Logic Modules in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51002-3.0*

Chapter 3 TriMatrix Embedded Memory Blocks in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51003-3.0*

Chapter 4 DSP Blocks in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51004-3.0*

Chapter 5 Clock Networks and PLLs in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51005-3.0*

Chapter 6 I/O Features in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51006-3.0*

Chapter 7 External Memory Interfaces in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51007-3.0*

Chapter 8 High-Speed Differential I/O Interfaces and DPA in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51008-3.0*

Chapter 9 Hot Socketing and Power-On Reset in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51009-3.0*

Chapter 10 Configuration, Design Security, and Remote System Upgrades in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51010-3.0*

Chapter 11 SEU Mitigation in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51011-3.0*

Chapter 12 JTAG Boundary-Scan Testing in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51012-3.0*

Chapter 13 Power Management in Stratix IV Devices
Revised: *November 2009*
Part Number: *SIV51013-3.0*

About this Handbook

This handbook provides comprehensive information about the Altera® Stratix® IV family of devices.

How to Contact Altera

For the most up-to-date information about Altera products, see the following table.

Contact *(Note 1)*	Contact Method	Address
Technical support	Website	www.altera.com/support
Technical training	Website	www.altera.com/training
	Email	custrain@altera.com
Product literature	Website	www.altera.com/literature
Non-technical support (General)	Email	nacomp@altera.com
(Software Licensing)	Email	authorization@altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

The following table shows the typographic conventions that this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicates command names, dialog box titles, dialog box options, and other GUI labels. For example, **Save As** dialog box. For GUI elements, capitalization matches the GUI.
bold type	Indicates directory names, project names, disk drive names, file names, file name extensions, dialog box options, software utility names, and other GUI labels. For example, **\qdesigns** directory, **d:** drive, and **chiptrip.gdf**.
Italic Type with Initial Capital Letters	Indicates document titles. For example, *AN 519: Stratix IV Design Guidelines.*
Italic type	Indicates variables. For example, *n* + 1. Variable names are enclosed in angle brackets (< >). For example, *<file name>* and *<project name>*.**pof**.
Initial Capital Letters	Indicates keyboard keys and menu names. For example, Delete key and the Options menu.
"Subheading Title"	Quotation marks indicate references to sections within a document and titles of Quartus II Help topics. For example, "Typographic Conventions."

Visual Cue	Meaning
Courier type	Indicates signal, port, register, bit, block, and primitive names. For example, `data1`, `tdi`, and `input`. Active-low signals are denoted by suffix n. For example, `resetn`. Indicates command line commands and anything that must be typed exactly as it appears. For example, `c:\qdesigns\tutorial\chiptrip.gdf`. Also indicates sections of an actual file, such as a Report File, references to parts of files (for example, the AHDL keyword `SUBDESIGN`), and logic function names (for example, `TRI`).
1., 2., 3., and a., b., c., and so on.	Numbered steps indicate a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■ ■	Bullets indicate a list of items when the sequence of the items is not important.
	The hand points to information that requires special attention.
	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
	A warning calls attention to a condition or possible situation that can cause you injury.
↵	The angled arrow instructs you to press **Enter**.
	The feet direct you to more information about a particular topic.

This section provides a complete overview of all features relating to the Stratix® IV device family, which is the most architecturally advanced, high-performance, low-power FPGA in the market place. This section includes the following chapters:

- Chapter 1, Stratix IV Device Family Overview
- Chapter 2, Logic Array Blocks and Adaptive Logic Modules in Stratix IV Devices
- Chapter 3, TriMatrix Embedded Memory Blocks in Stratix IV Devices
- Chapter 4, DSP Blocks in Stratix IV Devices
- Chapter 5, Clock Networks and PLLs in Stratix IV Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

1. Stratix IV Device Family Overview

SIV51001-3.0

Altera® Stratix® IV FPGAs deliver a breakthrough level of system bandwidth and power efficiency for high-end applications, allowing you to innovate without compromise. Stratix IV FPGAs are based on the Taiwan Semiconductor Manufacturing Company (TSMC) 40-nm process technology and surpass all other high-end FPGAs, with the highest logic density, most transceivers, and lowest power requirements.

The Stratix IV device family contains three optimized variants to meet different application needs:

- Stratix IV E (Enhanced) FPGAs—up to 813,050 logic elements (LEs), 33,294 Kbits RAM, and 1,288 18 × 18 bit multipliers
- Stratix IV GX transceiver FPGAs—up to 531,200 LEs, 27,376 Kbits RAM, 1,288 18 × 18-bit multipliers, and 48 full-duplex CDR-based transceivers at up to 8.5 Gbps
- Stratix IV GT FPGAs—up to 531,200 LEs, 27,376 Kbits RAM, 1,288 18 × 18-bit multipliers, and 48 full-duplex clock data recovery (CDR)-based transceivers at up to 11.3 Gbps

The complete Altera high-end solution includes the lowest risk, lowest total cost path to volume using HardCopy® IV ASICs for all the family variants, a comprehensive portfolio of application solutions customized for end-markets, and the industry leading Quartus® II software for increasing productivity and performance.

This chapter contains the following sections:

Feature Summary

The following list summarizes the Stratix IV device family features:

- Up to 48 full-duplex CDR-based transceivers in Stratix IV GX and GT devices supporting data rates up to 8.5 Gbps and 11.3 Gbps, respectively
- Dedicated circuitry to support physical layer functionality for popular serial protocols, such as PCI-Express (PIPE) Gen1 and Gen2, Gigabit Ethernet, Serial RapidIO, SONET/SDH, XAUI/HiGig, (OIF) CEI-6G, SD/HD/3G-SDI, Fibre Channel, SFI-5, and Interlaken
- Complete PCI Express (PIPE) protocol solution with embedded PCI Express hard IP blocks that implement PHY-MAC layer, Data Link layer, and Transaction layer functionality

For more information, refer to the *PCI Express Compiler User Guide*.

- Programmable transmitter pre-emphasis and receiver equalization circuitry to compensate for frequency-dependent losses in the physical medium
- Typical physical medium attachment (PMA) power consumption of 100 mW at 3.125 Gbps and 135 mW at 6.375 Gbps per channel
- 72,600 to 813,050 equivalent LEs per device
- 7,370 to 33,294 Kbits of enhanced TriMatrix memory consisting of three RAM block sizes to implement true dual-port memory and FIFO buffers
- High-speed DSP blocks configurable as 9 × 9-bit, 12 × 12-bit, 18 × 18-bit, and 36 × 36-bit full-precision multipliers at up to 600 MHz
- Up to 16 global clocks (GCLK), 88 regional clocks (RCLK), and 132 periphery clocks (PCLK) per device
- Programmable power technology that minimizes power while maximizing device performance
- Up to 1,120 user I/O pins arranged in 24 modular I/O banks that support a wide range of single-ended and differential I/O standards
- Support for high-speed external memory interfaces including DDR, DDR2, DDR3 SDRAM, RLDRAM II, QDR II, and QDR II+ SRAM on up to 24 modular I/O banks
- High-speed LVDS I/O support with serializer/deserializer (SERDES), dynamic phase alignment (DPA), and soft-CDR circuitry at data rates up to 1.6 Gbps
- Support for source-synchronous bus standards, including SGMII, Gigabit Ethernet, SPI-4 Phase 2 (POS-PHY Level 4), SFI-4.1, XSBI, UTOPIA IV, NPSI, and CSIX-L1
- Pinouts for Stratix IV E devices designed to allow migration of designs from Stratix III to Stratix IV E with minimal PCB impact

Stratix IV GX Devices

Stratix IV GX devices provide up to 48 CDR-based transceiver channels per device:

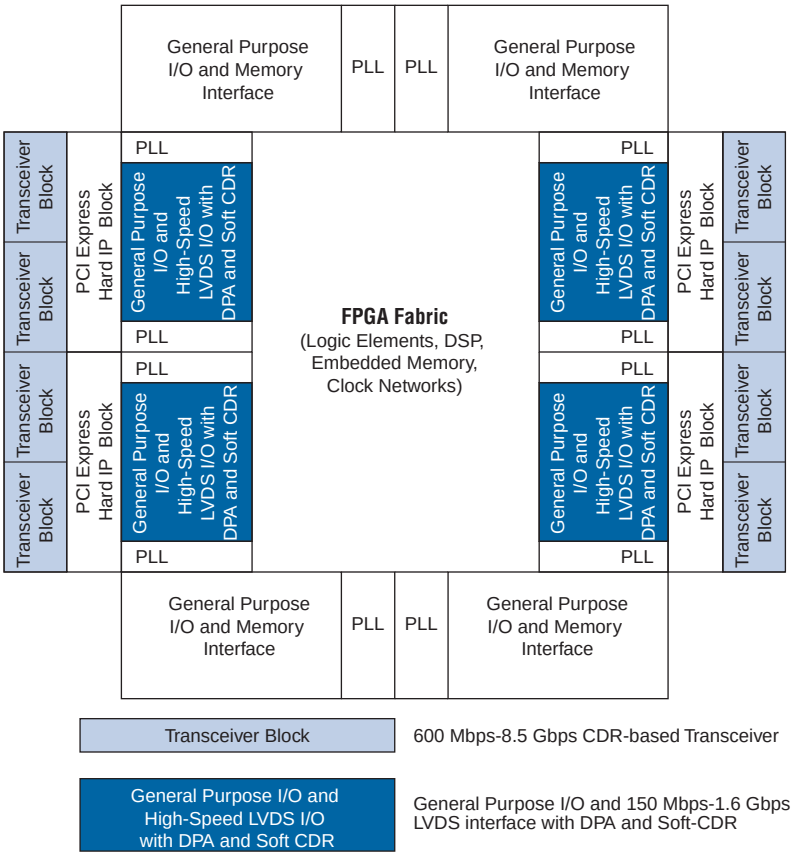
- Thirty-two out of the 48 transceiver channels have dedicated physical coding sublayer (PCS) and physical medium attachment (PMA) circuitry and support data rates between 600 Mbps and 8.5 Gbps
- The remaining sixteen transceiver channels have dedicated PMA-only circuitry and support data rates between 600 Mbps and 6.5 Gbps

The actual number of transceiver channels per device varies with device selection. For more information about the exact transceiver count in each device, refer to Table 1–1 on page 1–11.

For more information about transceiver architecture, refer to the *Stratix IV Transceiver Architecture* chapter.

Figure 1–1 shows a high-level Stratix IV GX chip view.

Figure 1–1. Stratix IV GX Chip View *(Note 1)*



Note to Figure 1–1:

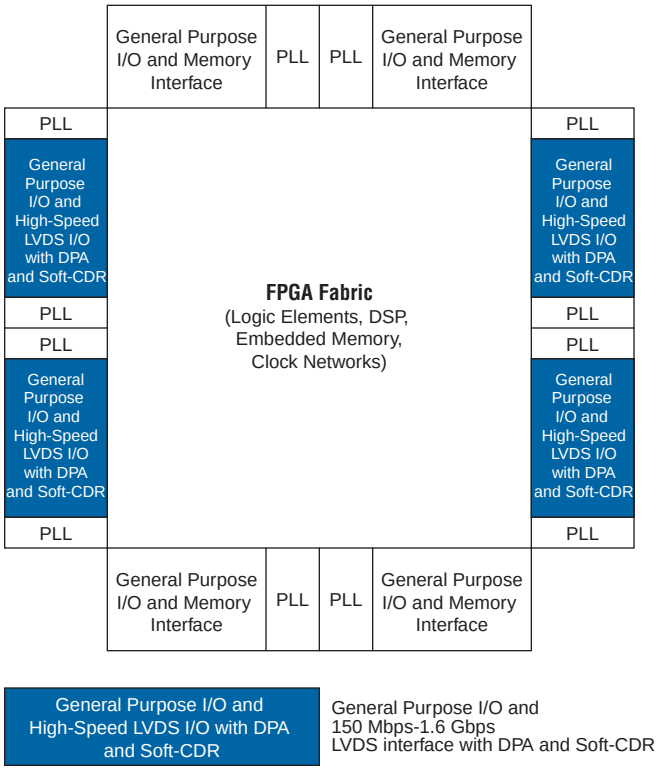
(1) Resource counts vary with device selection, package selection, or both.

Stratix IV E Device

Stratix IV E devices provide an excellent solution for applications that do not require high-speed CDR-based transceivers, but are logic, user I/O, or memory intensive.

Figure 1–2 shows a high-level Stratix IV E chip view.

Figure 1–2. Stratix IV E Chip View *(Note 1)*



Note to Figure 1–2:

(1) Resource counts vary with device selection, package selection, or both.

Stratix IV GT Devices

Stratix IV GT devices provide up to 48 CDR-based transceiver channels per device:

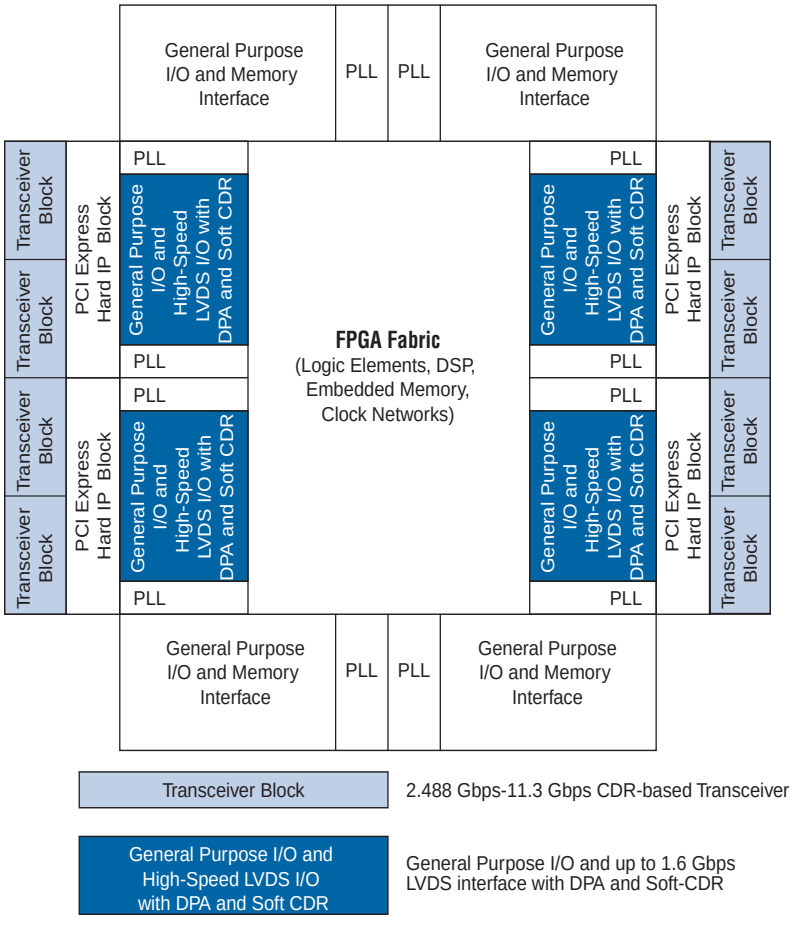
- Thirty-two out of the 48 transceiver channels have dedicated PCS and PMA circuitry and support data rates between 2.488 Gbps and 11.3 Gbps
- The remaining sixteen transceiver channels have dedicated PMA-only circuitry and support data rates between 2.488 Gbps and 6.5 Gbps

The actual number of transceiver channels per device varies with device selection. For more information about the exact transceiver count in each device, refer to Table 1–10 on page 1–20.

For more information about transceiver architecture, refer to the *Stratix IV Transceiver Architecture* chapter.

Figure 1–3 shows a high-level Stratix IV GT chip view.

Figure 1–3. Stratix IV GT Chip View *(Note 1)*



Note to Figure 1–3:

(1) Resource counts vary with device selection, package selection, or both.

For information about Stratix IV GT devices, refer to the *Stratix IV Transceiver Architecture* chapter.

Architecture Features

The Stratix IV device family features are divided into high-speed transceiver features and FPGA fabric and I/O features.

The high-speed transceiver features apply only to Stratix IV GX and Stratix IV GT devices.

High-Speed Transceiver Features

Stratix IV GX and Stratix IV GT high-speed transceiver features include:

Highest Aggregate Data Bandwidth

- Up to 48 full-duplex transceiver channels supporting data rates up to 8.5 Gbps in Stratix IV GX devices and up to 11.3 Gbps in Stratix IV GT devices.

Wide Range of Protocol Support

Physical layer support for the following serial protocols:

- Stratix IV GX—PCI Express (PIPE) Gen1 and Gen2, Gigabit Ethernet, Serial RapidIO, SONET/SDH, XAUI/HiGig, (OIF) CEI-6G, SD/HD/3G-SDI, Fibre Channel, SFI-5, GPON, SAS/SATA, HyperTransport 1.0 and 3.0, and Interlaken
- Stratix IV GT—40G/100G Ethernet, SFI-S, Interlaken, SFI-5.1, Serial RapidIO, SONET/SDH, XAUI/HiGig, (OIF) CEI-6G, 3G-SDI, and Fibre Channel
- Extremely flexible and easy-to-configure transceiver data path to implement proprietary protocols
- PCI Express (PIPE) Support
  - Complete PCI Express (PIPE) Gen1 and Gen2 protocol stack solution compliant to PCI Express base specification 2.0 that includes PHY-MAC, Data Link, and transaction layer circuitry embedded in PCI Express hard IP blocks

    For more information, refer to the *PCI Express Compiler User Guide*.

  - Root complex and end-point applications
  - ×1, ×4, and ×8 lane configurations
  - PIPE 2.0-compliant interface
  - Embedded circuitry to switch between Gen1 and Gen2 data rates
  - Built-in circuitry for electrical idle generation and detection, receiver detect, power state transitions, lane reversal, and polarity inversion
  - 8B/10B encoder and decoder, receiver synchronization state machine, and ± 300 parts per million (ppm) clock compensation circuitry
  - Transaction layer support for up to two virtual channels (VCs)

- XAUI/HiGig Support
  - Compliant to IEEE802.3ae specification
  - Embedded state machine circuitry to convert XGMII idle code groups (||I||) to and from idle ordered sets (||A||, ||K||, ||R||) at the transmitter and receiver, respectively
  - 8B/10B encoder and decoder, receiver synchronization state machine, lane deskew, and ± 100 ppm clock compensation circuitry
- Gigabit Ethernet Support
  - Compliant to IEEE802.3-2005 specification
  - Automatic idle ordered set (/I1/, /I2/) generation at the transmitter, depending on the current running disparity
  - 8B/10B encoder and decoder, receiver synchronization state machine, and ± 100 ppm clock compensation circuitry
- Support for other protocol features such as MSB-to-LSB transmission in SONET/SDH configuration and spread-spectrum clocking in PCI Express (PIPE) configurations

Diagnostic Features

- Serial loopback from the transmitter serializer to the receiver CDR for transceiver PCS and PMA diagnostics
- Reverse serial loopback pre- and post-CDR to transmitter buffer for physical link diagnostics
- Loopback master and slave capability in PCI Express hard IP blocks

For more information, refer to the *PCI Express Compiler User Guide*.

Signal Integrity

Stratix IV devices simplify the challenge of signal integrity through a number of chip, package, and board-level enhancements to enable efficient high-speed data transfer into and out of the device. These enhancements include:

- Programmable 3-tap transmitter pre-emphasis with up to 8192 pre-emphasis levels to compensate for pre-cursor and post-cursor inter-symbol interference (ISI)
- Up to 900% boost capability on the first pre-emphasis post-tap
- User-controlled and adaptive 4-stage receiver equalization with up to 16 dB of high-frequency gain
- On-die power supply regulators for transmitter and receiver phase-locked loop (PLL) charge pump and voltage controlled oscillator (VCO) for superior noise immunity
- On-package and on-chip power supply decoupling to satisfy transient current requirements at higher frequencies, thereby reducing the need for on-board decoupling capacitors
- Calibration circuitry for transmitter and receiver on-chip termination (OCT) resistors

FPGA Fabric and I/O Features

The Stratix IV FPGA fabric and I/O features include:

Device Core Features

- Up to 531,200 LEs in Stratix IV GX and Stratix IV GT devices and up to 813,050 LEs in Stratix IV E devices, efficiently packed in unique and innovative adaptive logic modules (ALMs)
- Ten ALMs per logic array block (LAB) deliver faster performance, improved logic utilization, and optimized routing
- Programmable power technology, including a variety of process, circuit, and architecture optimizations and innovations
- Programmable power technology available to select power-driven compilation options for reduced static power consumption

Embedded Memory

- TriMatrix embedded memory architecture provides three different memory block sizes to efficiently address the needs of diversified FPGA designs:
  - 640-bit MLAB
  - 9-Kbit M9K
  - 144-Kbit M144K
- Up to 33,294 Kbit of embedded memory operating at up to 600 MHz
- Each memory block is independently configurable to be a single- or dual-port RAM, FIFO, ROM, or shift register

Digital Signal Processing (DSP) Blocks

- Flexible DSP blocks configurable as 9 × 9-bit, 12 × 12-bit, 18 × 18-bit, and 36 × 36-bit full-precision multipliers at up to 600 MHz with rounding and saturation capabilities
- Faster operation due to fully pipelined architecture and built-in addition, subtraction, and accumulation units to combine multiplication results
- Optimally designed to support advanced features such as adaptive filtering, barrel shifters, and finite and infinite impulse response (FIR and IIR) filters

Clock Networks

- Up to 16 GCLKs and 88 RCLKs optimally routed to meet the maximum performance of 800 MHz
- Up to 112 and 132 PCLKs in Stratix IV GX and Stratix IV E devices, respectively
- Up to 66 (16 GCLK + 22 RCLK + 28 PCLK) per device quadrant in Stratix IV GX and Stratix IV GT devices
- Up to 71 (16 GCLK + 22 RCLK + 33 PCLK) per device quadrant in Stratix IV E devices

PLLs

- Three to 12 PLLs per device supporting spread-spectrum input tracking, programmable bandwidth, clock switchover, dynamic reconfiguration, and delay compensation
- On-chip PLL power supply regulators to minimize noise coupling

I/O Features

- Sixteen to 24 modular I/O banks per device with 24 to 48 I/Os per bank designed and packaged for optimal simultaneous switching noise (SSN) performance and migration capability
- Support for a wide range of industry I/O standards, including single-ended (LVTTL/CMOS/PCI/PCIX), differential (LVDS/mini-LVDS/RSDS), voltage-referenced single-ended and differential (SSTL/HSTL Class I/II) I/O standards
- On-chip series (R_S) and on-chip parallel (R_T) termination with auto-calibration for single-ended I/Os and on-chip differential (R_D) termination for differential I/Os
- Programmable output drive strength, slew rate control, bus hold, and weak pull-up capability for single-ended I/Os
- User I/O:GND:V_{CC} ratio of 8:1:1 to reduce loop inductance in the package—PCB interface
- Programmable transmitter differential output voltage (V_{OD}) and pre-emphasis for high-speed LVDS I/O

High-Speed Differential I/O with DPA and Soft-CDR

- Dedicated circuitry on the left and right sides of the device to support differential links at data rates from 150 Mbps to 1.6 Gbps
- Up to 98 differential SERDES in Stratix IV GX devices, up to 132 differential SERDES in Stratix IV E devices, and up to 47 differential SERDES in Stratix IV GT devices
- DPA circuitry at the receiver automatically compensates for channel-to-channel and channel-to-clock skew in source synchronous interfaces
- Soft-CDR circuitry at the receiver allows implementation of asynchronous serial interfaces with embedded clocks at up to 1.6 Gbps data rate (SGMII and Gigabit Ethernet)

External Memory Interfaces

- Support for existing and emerging memory interface standards such as DDR SDRAM, DDR2 SDRAM, DDR3 SDRAM, QDRII SRAM, QDRII+ SRAM, and RLDRAM II
- DDR3 up to 1,067 Mbps/533 MHz
- Programmable DQ group widths of 4 to 36 bits (includes parity bits)
- Dynamic OCT, trace mismatch compensation, read-write leveling, and half-rate register capabilities provide a robust external memory interface solution

System Integration

- All Stratix IV devices support hot socketing
- Four configuration modes:
  - Passive Serial (PS)
  - Fast Passive Parallel (FPP)
  - Fast Active Serial (FAS)
  - JTAG configuration
- Ability to perform remote system upgrades
- 256-bit advanced encryption standard (AES) encryption of configuration bits protects your design against copying, reverse engineering, and tampering
- Built-in soft error detection for configuration RAM cells

For more information about how to connect the PLL, external memory interfaces, I/O, high speed differential I/O, power, and the JTAG pins to PCB, refer to the *Stratix IV GX and Stratix IV E Device Family Pin Connection Guidelines* and the *Stratix IV GT Device Family Pin Connection Guidelines*.

Table 1–1 lists the Stratix IV GX device features.

Table 1–1. Stratix IV GX Device Features (Part 1 of 2)

Feature	EP4SGX70		EP4SGX110			EP4SGX180			EP4SGX230			EP4SGX290					EP4SGX360			
Package Option	F780	F1152	F780	F1152		F780	F1152	F1517	F780	F1152	F1517	F780	F1152	F1517	F1760	F1932	F780	F1152	F1517	F1760
ALMs	29,040		42,240			70,300			91,200			116,480					141,440			
LEs	72,600		105,600			175,750			228,000			291,200					353,600			
0.6 Gbps-8.5 Gbps Transceivers (PMA + PCS) *(1)*	—	16	—	—	16	—	—	16	24	—	—	16	24	—	—	16	24	24	32	—	—	16	24	24
0.6 Gbps-6.5 Gbps Transceivers (PMA +PCS) *(1)*	8	—	8	16	—	8	16	—	—	8	16	—	—	16	16	—	—	—	—	16	16	—	—	—
PMA-only CMU Channels (0.6 Gbps-6.5 Gbps)	—	8	—	—	8	—	—	8	12	—	—	8	12	—	—	8	12	12	16	—	—	8	12	12
PCI Express hard IP Blocks	1	2	1	2		1	2		1	2		2				4	2							
High-Speed LVDS SERDES (up to 1.6 Gbps) *(4)*	28	56	28	28	56	28	44	88	28	44	88	—	44	88	88	98	—	44	88	88
SPI-4.2 Links	1		1			1	2	4	1	2	4	—	2	4			—	2	4	

Table 1–1. Stratix IV GX Device Features (Part 2 of 2)

Feature	EP4SGX70		EP4SGX110			EP4SGX180				EP4SGX230				EP4SGX290						EP4SGX360				
Package Option	**F780**	**F1152**	**F780**	**F1152**		**F780**	**F1152**		**F1517**	**F780**	**F1152**		**F1517**	**F780**	**F1152**		**F1517**	**F1760**	**F1932**	**F780**	**F1152**		**F1517**	**F1760**
M9K Blocks (256 × 36 bits)	462		660			950				1,235				936						1,248				
M144K Blocks (2048 × 72 bits)	16		16			20				22				36						48				
Total Memory (MLAB+M9K+ M144K) Kbits	7,370		9,564			13,627				17,133				17,248						22,564				
Embedded Multipliers 18 × 18 *(2)*	384		512			920				1,288				832						1,040				
PLLs	3	4	3	4		3	6		8	3	6		8	4	6		8	12	12	4	6		8	12
User I/Os *(3)*	372	488	372	372	488	372	564	564	744	372	564	564	744	289	564	564	744	880	920	289	564	564	744	88
Speed Grade (fastest to slowest)	–2×, –3, –4	–2, –3, –4	–2×, –3, –4	–2×, –3, –4	–2, –3, –4	–2×, –3, –4	–2×, –3, –4	–2, –3, –4	–2, –3, –4	–2×, –3, –4	–2×, –3, –4	–2, –3, –4	–2, –3, –4	–2×, –3, –4	–2×, –3, –4	–2, –3, –4	–2, –3, –4	–2, –3, –4	–2, –3, –4	–2×, –3, –4	–2×, –3, –4	–2, –3, –4	–2, –3, –4	–2 –3 –4

Notes to Table 1–1:

(1) The total number of transceivers is divided equally between the left and right side of each device, except for the devices in the F780 package. These devices have eight on the right side of the device.

(2) Four multiplier adder mode.

(3) The user I/Os count from pin-out files includes all general purpose I/O, dedicated clock pins, and dual purpose configuration pins. Transceiver pins and dedicated config pin count.

(4) Total pairs of High-Speed LVDS SERDES take the lowest channel count of R_X/T_X.

Table 1–2 summarizes the Stratix IV GX device package options.

Table 1–2. Stratix IV GX Device Package Options *(Note 1)*

Device	F780 (29 mm × 29 mm) *(5)*	F1152 (35 mm × 35 mm) *(5)*	F1152 (35 mm × 35 mm) *(4)*, *(6)*	F1517 (40 mm × 40 mm) *(4)*, *(6)*	F1760 (42.5 mm × 42.5 mm)
EP4SGX70	DF29	—	HF35	—	—
EP4SGX110	DF29	FF35	HF35	—	—
EP4SGX180	DF29	FF35	HF35	KF40	—
EP4SGX230	DF29	FF35	HF35	KF40	—
EP4SGX290	FH29 *(2)*	FF35	HF35	KF40	KF43
EP4SGX360	FH29 *(2)*	FF35	HF35	KF40	KF43
EP4SGX530	—	—	HH35 *(3)*	KH40 *(3)*	KF43

Notes to Table 1–2:

(1) Device packages in the same column and marked under the same arrow sign have vertical migration capability.

(2) The 780-pin EP4SGX290 and EP4SGX360 devices are available only in 33 mm × 33 mm Hybrid flip chip package.

(3) The 1152-pin and 1517-pin EP4SGX530 devices are available only in 42.5 mm × 42.5 mm Hybrid flip chip packages.

(4) When migrating between hybrid and flip chip packages, there is an additional keep-out area. For more information, refer to the *Altera Device Package Information Dat*

(5) Devices listed in this column are available in –2×, –3, and –4 speed grades. These devices do not have on-package decoupling capacitors.

(6) Devices listed in this column are available in –2, –3, and –4 speed grades. These devices have on-package decoupling capacitors. For more information about on-packa each device, refer to Table 1–3.

Table 1–3 lists the Stratix IV GX device on-package decoupling information. On-package decoupling re on-board or PCB decoupling capacitors by satisfying the transient current requirements at higher frequ *Delivery Network* design tool for Stratix IV devices accounts for the on-package decoupling and reflects requirements for PCB decoupling capacitors.

Table 1–3. Stratix IV GX Device On-Package Decoupling Information *(Note 1)* (Part 1 of 2)

Ordering Information		V_{CC}	V_{CCIO}	V_{CCL_GXB}	$V_{CCA\_L/R}$
EP4SGX70	HF35	2× 1uF + 2× 470nF	10nF per bank *(2)*	100nF per transceiver block	100nF
EP4SGX110	HF35	2× 1uF + 2× 470nF	10nF per bank *(2)*	100nF per transceiver block	100nF
EP4SGX180	HF35 KF40	2× 1uF + 2× 470nF	10nF per bank *(2)*	100nF per transceiver block	100nF

Table 1–3. Stratix IV GX Device On-Package Decoupling Information *(Note 1)* (Part 2 of 2)

Ordering Information		V_{CC}	V_{CCIO}	V_{CCL_GXB}	$V_{CCA\_L/R}$
EP4SGX230	HF35 KF40	2×1 uF + 2×470 nF	10 nF per bank *(2)*	100 nF per transceiver block	100 nF
EP4SGX290	HF35 KF40 KF43 NF45	4×1 uF + 4×470 nF	10 nF per bank *(2)*	100 nF per transceiver block	100nF
EP4SGX360	HF35 KF40 KF43 NF45	4×1 uF + 4×470 nF	10 nF per bank *(2)*	100 nF per transceiver block	100 nF
EP4SGX530	HH35 KH40 KF43 NF45	4×1 uF + 4×470 nF	10 nF per bank *(2)*	100 nF per transceiver block	100 nF

Notes to Table 1–3:

(1) Table 1–3 is referring to production devices on-package decoupling. For more information about decoupling design of engineering sample (ES) devices, contact Altera

(2) For I/O banks 3(*), 4(*), 7(*), and 8(*) only. There is no OPD for I/O bank 1(*), 2(*), 5(*), and 6(*).

Table 1–4 lists the Stratix IV E device features.

Table 1–4. Stratix IV E Device Features

Feature	EP4SE230	EP4SE360		EP4SE530			EP4SE820		
Package Pin Count	780	780	1152	1152	1517	1760	1152	1517	1760
ALMs	91,200	141,440		212,480			325,220		
LEs	228,000	353,600		531,200			813,050		
High-Speed LVDS SERDES (up to 1.6 Gbps) *(3)*	56	56	88	88	112	112	88	112	132
SPI-4.2 Links	3	3	4	4	6		4	6	6
M9K Blocks (256 × 36 bits)	1,235	1,248		1,280			1610		
M144K Blocks (2048 × 72 bits)	22	48		64			60		
Total Memory (MLAB+M9K+ M144K) Kbits	17,133	22,564		27,376			33,294		
Embedded Multipliers (18 × 18) *(1)*	1,288	1,040		1,024			960		
PLLs	4	4	8	8	12	12	8	12	12
User I/Os *(2)*	488	488	744	744	976	976	744 *(4)*	976 *(4)*	1120 *(4)*
Speed Grade (fastest to slowest)	–2, –3, –4	–2, –3, –4	–2, –3, –4	–2, –3, –4	–2, –3, –4	–2, –3, –4	–3, –4	–3, –4	–3, –4

Notes to Table 1–4:

(1) Four multiplier adder mode.

(2) The user I/O count from the pin-out files include all general purpose I/Os, dedicated clock pins, and dual purpose configuration pins. Transceiver pins and dedicated configuration pins are not included in the pin count.

(3) Total pairs of high-speed LVDS SERDES take the lowest channel count of R_X/T_X.

(4) This data is preliminary.

Table 1–5 summarizes the Stratix IV E device package options.

Table 1–5. Stratix IV E Device Package Options *(Note 1)*

Device	F780 (29 mm × 29 mm) *(4)*, *(5)*	F1152 (35 mm × 35 mm) *(4)*, *(6)*	F1517 (40 mm × 40 mm) *(6)*	F1760 (42.5 mm × 42.5 mm) *(6)*
EP4SE230	F29	—	—	—
EP4SE360	H29 *(2)*	F35	—	—
EP4SE530	—	H35 *(3)*	H40 *(3)*	F43
EP4SE820	—	H35 *(3)*	H40 *(3)*	F43

Notes to Table 1–5:

(1) Device packages in the same column and marked under the same arrow sign have vertical migration capability.
(2) The 780-pin EP4SE360 device is available only in the 33 mm × 33 mm hybrid flip chip package.
(3) The 1152-pin and 1517-pin for EP4SE530 and EP4SE820 devices are available only in the 42.5 mm × 42.5 mm hybrid flip chip package.
(4) When migrating between hybrid and flip chip packages, there is an additional keep-out area. For more information, refer to the *Altera Device Package Information Data Sheet*.
(5) Devices listed in this column do not have on-package decoupling capacitors.
(6) Devices listed in this column have on-package decoupling capacitors. For more information about on-package decoupling capacitor value for each device, refer to Table 1–6.

Table 1–6 lists the Stratix IV E on-package decoupling information. On-package decoupling reduces the need for on-board or PCB decoupling capacitors by satisfying the transient current requirements at higher frequencies. The *Power Delivery Network* design tool for Stratix IV devices accounts for the on-package decoupling and reflects the reduced requirements for PCB decoupling capacitors.

Table 1–6. Stratix IV E Device On-Package Decoupling Information *(Note 1)*

Ordering Information		V_{CC}	V_{CCIO}
EP4SE360	F35	4×1 uF + 4×470 nF	10 nF per bank
EP4SE530	H35 H40 F43	4×1 uF + 4×470 nF	10 nF per bank
EP4SE820	H35 H40 F43	4×1 uF + 4×470 nF	10 nF per bank

Note to Table 1–6:

(1) Table 1–6 is referring to production devices on-package decoupling. For more information about decoupling design of engineering sample (ES) devices, contact Altera Technical Support.

Table 1–7 lists the Stratix IV GT device features.

Table 1–7. Stratix IV GT Device Features (Part 1 of 2)

Feature	EP4S40G2	EP4S40G5	EP4S100G2	EP4S100G3	EP4S100G4	EP4S100G5	
Package Pin Count	1517	1517	1517	1932	1932	1517	1932
ALMs	91,200	212,480	91,200	116,480	141,440	212,480	
LEs	228,000	531,200	228,000	291,200	353,600	531,200	
Total Transceiver Channels	36	36	36	48	48	36	48

Table 1–7. Stratix IV GT Device Features (Part 2 of 2)

Feature	EP4S40G2	EP4S40G5	EP4S100G2	EP4S100G3	EP4S100G4	EP4S100G5	
10G Transceiver Channels (2.488 Gbps-11.3 Gbps with PMA + PCS)	12	12	24	24	24	24	32
8G Transceiver Channels (2.488 Gbps- 8.5 Gbps with PMS + PCS) *(3)*	12	12	0	8	8	0	0
PMA-only CMU Channels (2.488 Gbps - 6.5 Gbps)	12	12	12	16	16	12	16
PCI Express hard IP Blocks	2	2	2	4	4	2	4
High-Speed LVDS SERDES (up to 1.6 Gbps) *(4)*	46	46	46	47	47	46	47
SP1-4.2 Links	2	2	2	2	2	2	2
M9K Blocks (256 × 72 bits)	1,235	1,280	1,235	936	1,248	1,280	
M144K Blocks (2048 × 72 bits)	22	64	22	36	48	64	
Total Memory (MLAB + M9K + M144K) Kbits	17,133	27,376	17,133	17,248	22,564	27,376	
Embedded Multipliers 18 × 18 *(1)*	1,288	1,024	1,288	832	1,024	1,024	
PLLs	8	8	8	12	12	8	12
User I/Os *(2)*, *(5)*	654	654	654	781	781	654	781
Speed Grade (fastest to slowest)	–1, –2, –3	–1, –2, –3	–1, –2, –3	–1, –2, –3	–1, –2, –3	–1, –2, –3	–1, –2, –3

Notes to Table 1–7:

(1) Four multiplier adder mode.

(2) The user I/O count from the pin-out files include all general purpose I/Os, dedicated clock pins, and dual purpose configuration pins. Transceiver pins and dedicated configuration pins are not included in the pin count.

(3) You can configure all 10G transceiver channels as 8G transceiver channels. For example, the EP4S40G2F40 device has twenty-four 8G transceiver channels and the EP4S100G5F45 device has thirty-two 8G transceiver channels.

(4) Total pairs of high-speed LVDS SERDES take the lowest channel count of R_X/T_X.

(5) This data is preliminary.

Table 1–8 summarizes the resource counts for the Stratix IV GT devices.

Table 1–8. Stratix IV GT Device Package Options *(Note 1)*

Device	1517 Pin (40 mm × 40 mm) *(3)*	1932 Pin (45 mm × 45 mm)
Stratix IV GT 40 G Devices		
EP4S40G2	F40	—
EP4S40G5	H40 *(2)*	—
Stratix IV GT 100 G Devices		
EP4S100G2	F40	—
EP4S100G3	—	F45
EP4S100G4	—	F45
EP4S100G5	H40 *(2)*	F45

Notes to Table 1–8:

(1) Devices under the same arrow sign have vertical migration capability.

(2) EP4S40G5 and EP4S100G5 devices with 1517 pin-count are only available in 42.5-mm x 42.5-mm hybrid flip chip packages.

(3) When migrating between hybrid and flip chip packages, there is an additional keep-out area. For more information, refer to the *Altera Device Package Information Data Sheet*.

Table 1–9 lists the Stratix IV GT on-package decoupling information. On-package decoupling reduces the need for on-board or PCB decoupling capacitors by satisfying the transient current requirements at higher frequencies. The *Power Delivery Network* design tool for Stratix IV devices accounts for the on-package decoupling and reflects the reduced requirements for PCB decoupling capacitors.

Table 1–9. Stratix IV GT Device On-Package Decoupling Information *(Note 1)*

Ordering Information	V_{CC}	V_{CCIO}	V_{CCL_GXB}	$V_{CCA\_L/R}$	$V_{CCT\_L/R}$	$V_{CCR\_L/R}$
EP4S40G2F40 EP4S100G2F40	2× 1 uF + 2× 470 nF	10 nF per bank *(2)*	100 nF per transceiver block	100 nF	100 nF	100 nF
EP4S100G3F45	4× 1 uF + 4× 470 nF	10 nF per bank *(2)*	100 nF per transceiver block	100 nF	100 nF	100 nF
EP4S100G4F45						
EP4S40G5H40						
EP4S100G5H40						
EP4S100G5F45						

Notes to Table 1–9:

(1) Table 1–9 is referring to production devices on-package decoupling. For more information about decoupling design of engineering sample (ES) devices, contact Altera Technical Support.

(2) For I/O banks 3(*), 4(*), 7(*), and 8(*) only. There is no OPD for I/O bank 1(*), 2(*), 5(*), and 6(*).

Integrated Software Platform

The Quartus II software provides an integrated environment for HDL and schematic design entry, compilation and logic synthesis, full simulation and advanced timing analysis, SignalTap II logic analyzer, and device configuration of Stratix IV designs. The Quartus II software provides the MegaWizard™ Plug-In Manager user interface to generate different functional blocks, such as memory, PLL, and digital signal processing logic. For transceivers, the Quartus II software provides the ALTGX MegaWizard Plug-In Manager interface that guides you through configuration of the transceiver based on your application requirements.

The Stratix IV GX and GT transceivers allow you to implement low-power and reliable high-speed serial interface applications with its fully reconfigurable hardware, optimal signal integrity, and integrated Quartus II software platform.

For more information about the Quartus II software features, refer to the *Quartus II Handbook*.

Ordering Information

This section describes the Stratix IV E, GT, and GX devices ordering information. Figure 1–4 describes the ordering codes for Stratix IV GX and E devices.

Figure 1–4. Stratix IV GX and E Device Packaging Ordering Information

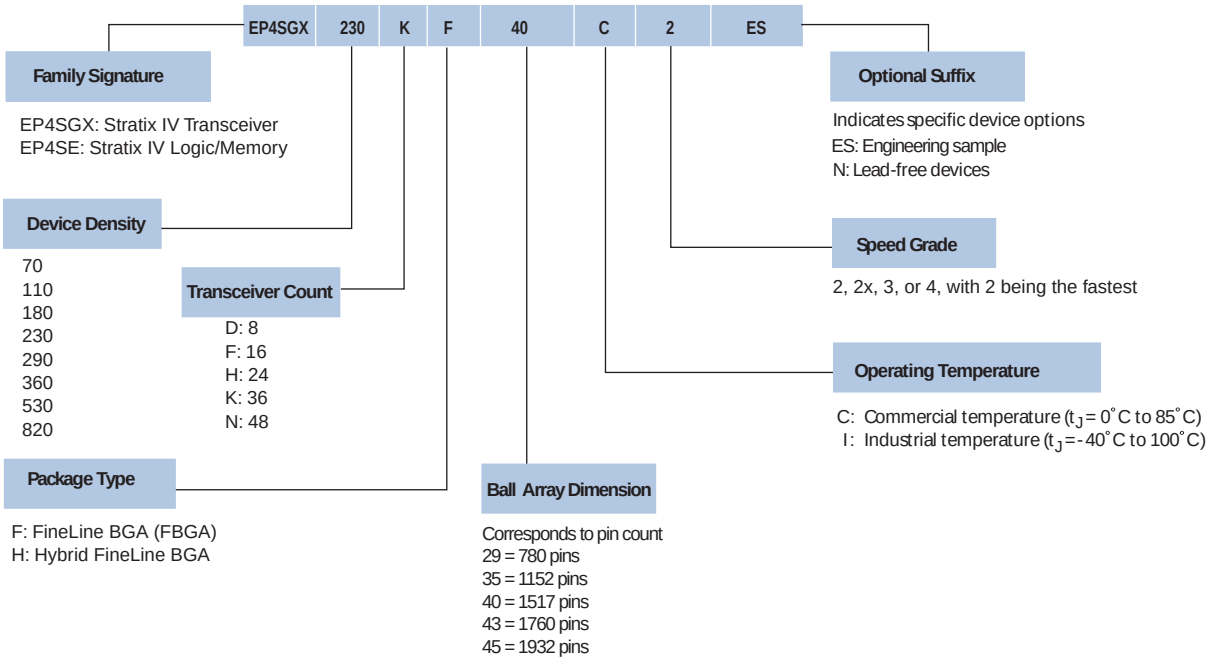
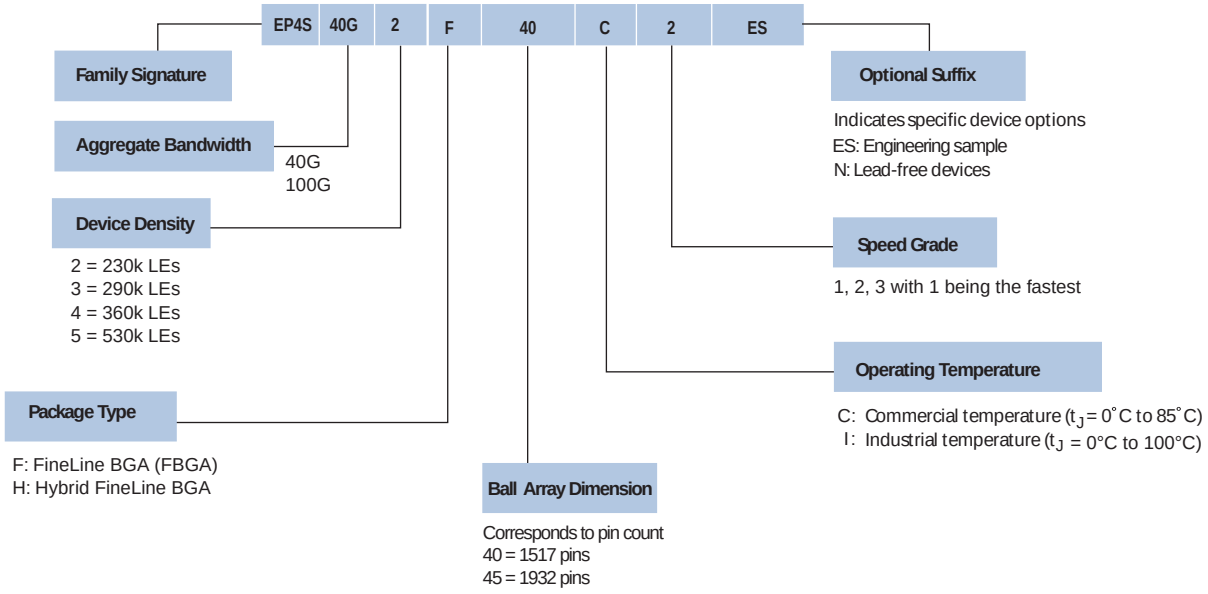


Figure 1–5 describes the ordering codes for Stratix IV GT devices.

Figure 1–5. Stratix IV GT Device Packaging Ordering Information



Document Revision History

Table 1–10 shows the revision history for this chapter.

Table 1–10. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated the "Stratix IV Device Family Overview", "Feature Summary", "Stratix IV GT Devices", "High-Speed Transceiver Features", "FPGA Fabric and I/O Features", "Highest Aggregate Data Bandwidth", "System Integration", and "Integrated Software Platform" sections. ■ Added Table 1–3, Table 1–6, and Table 1–9. ■ Updated Table 1–1, Table 1–2, Table 1–4, Table 1–5, Table 1–7, and Table 1–8. ■ Updated Figure 1–3, Figure 1–4, and Figure 1–5. ■ Minor text edits.	—
June 2009 v2.4	■ Updated Table 1–1. ■ Minor text edits.	—
April 2009 v2.3	■ Added Table 1–5, Table 1–6, and Figure 1–3. ■ Updated Figure 1–5. ■ Updated Table 1–1, Table 1–2, Table 1–3, and Table 1–4. ■ Updated "Introduction", "Feature Summary", "Stratix IV GX Devices", "Stratix IV GT Devices", "Architecture Features", and "FPGA Fabric and I/O Features"	—

Table 1–10. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
March 2009 v2.2	■ Updated “Feature Summary”, “Stratix IV GX Devices”, “Stratix IV E Device”, “Stratix IV GT Devices”, “Signal Integrity” ■ Removed Tables 1-5 and 1-6 ■ Updated Figure 1–4	—
March 2009 v2.1	■ Updated “Introduction”, “Feature Summary”, “Stratix IV Device Diagnostic Features”, “Signal Integrity”, “Clock Networks”,“High-Speed Differential I/O with DPA and Soft-CDR”, “System Integration”, and “Ordering Information” sections. ■ Added “Stratix IV GT 100G Devices” and “Stratix IV GT 100G Transceiver Bandwidth” sections. ■ Updated Table 1–1, Table 1–2, Table 1–3, and Table 1–4. ■ Added Table 1–5 and Table 1–6. ■ Updated Figure 1–3 and Figure 1–4. ■ Added Figure 1–5. ■ Removed “Referenced Documents” section.	—
November 2008 v2.0	■ Updated “Feature Summary” on page 1–1. ■ Updated “Stratix IV Device Diagnostic Features” on page 1–7. ■ Updated “FPGA Fabric and I/O Features” on page 1–8. ■ Updated Table 1–1. ■ Updated Table 1–2. ■ Updated “Table 1–5 shows the total number of transceivers available in the Stratix IV GT Device.” on page 1–15.	—
July 2008 v1.1	Revised “Introduction”.	—
May 2008 v1.0	Initial Release.	—

2. Logic Array Blocks and Adaptive Logic Modules in Stratix IV Devices

SIV51002-3.0

This chapter describes the features of the LABs in the Stratix IV core fabric. LABs are made up of ALMs you can configure to implement logic functions, arithmetic functions, and register functions.

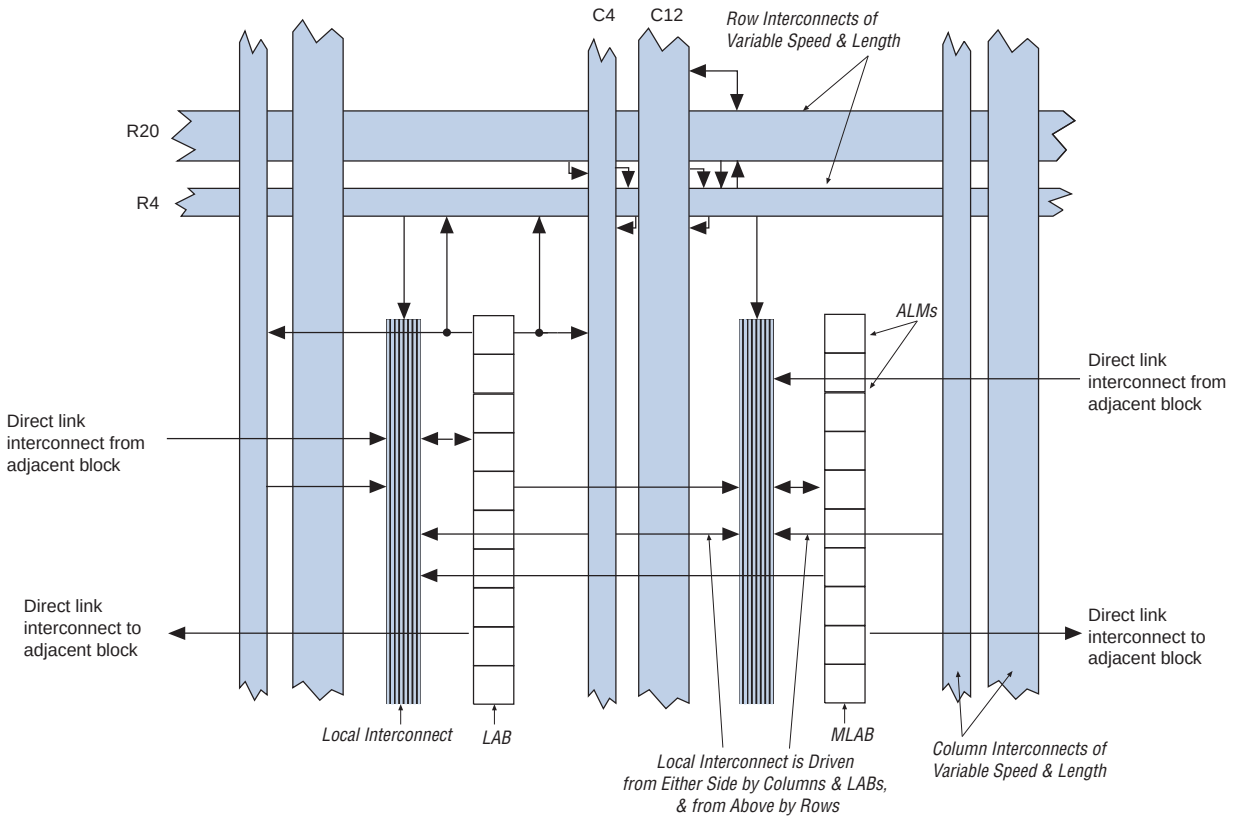
Logic array blocks (LABs) and adaptive logic modules (ALMs) are the basic building blocks of the Stratix® IV device. You can use these to configure logic functions, arithmetic functions, and register functions. The ALM provides advanced features with efficient logic utilization and is completely backward-compatible.

This chapter contains the following sections:

Logic Array Blocks

Each LAB consists of ten ALMs, various carry chains, shared arithmetic chains, LAB control signals, local interconnect, and register chain connection lines. The local interconnect transfers signals between ALMs in the same LAB. The direct link interconnect allows the LAB to drive into the local interconnect of its left and right neighbors. Register chain connections transfer the output of the ALM register to the adjacent ALM register in the LAB. The Quartus® II Compiler places associated logic in the LAB or adjacent LABs, allowing the use of local, shared arithmetic chain, and register chain connections for performance and area efficiency. Figure 2–1 shows the Stratix IV LAB structure and the LAB interconnects.

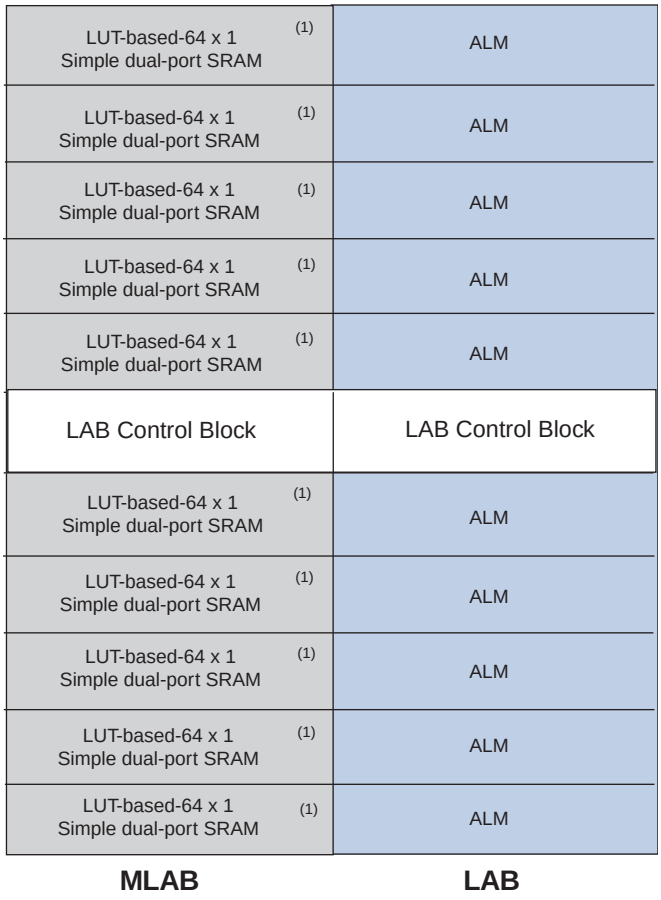
Figure 2–1. Stratix IV LAB Structure



The LAB of the Stratix IV device has a derivative called memory LAB (MLAB), which adds look-up table (LUT)-based SRAM capability to the LAB, as shown in Figure 2–2. The MLAB supports a maximum of 640 bits of simple dual-port static random access memory (SRAM). You can configure each ALM in an MLAB as either a 64 × 1 or a 32 × 2 block, resulting in a configuration of either a 64 × 10 or a 32 × 20 simple dual-port SRAM block. MLAB and LAB blocks always coexist as pairs in all Stratix IV families. MLAB is a superset of the LAB and includes all LAB features.

The MLAB is described in detail in the *TriMatrix Embedded Memory Blocks in Stratix IV Devices* chapter.

Figure 2–2. Stratix IV LAB and MLAB Structure



Note to Figure 2–2:

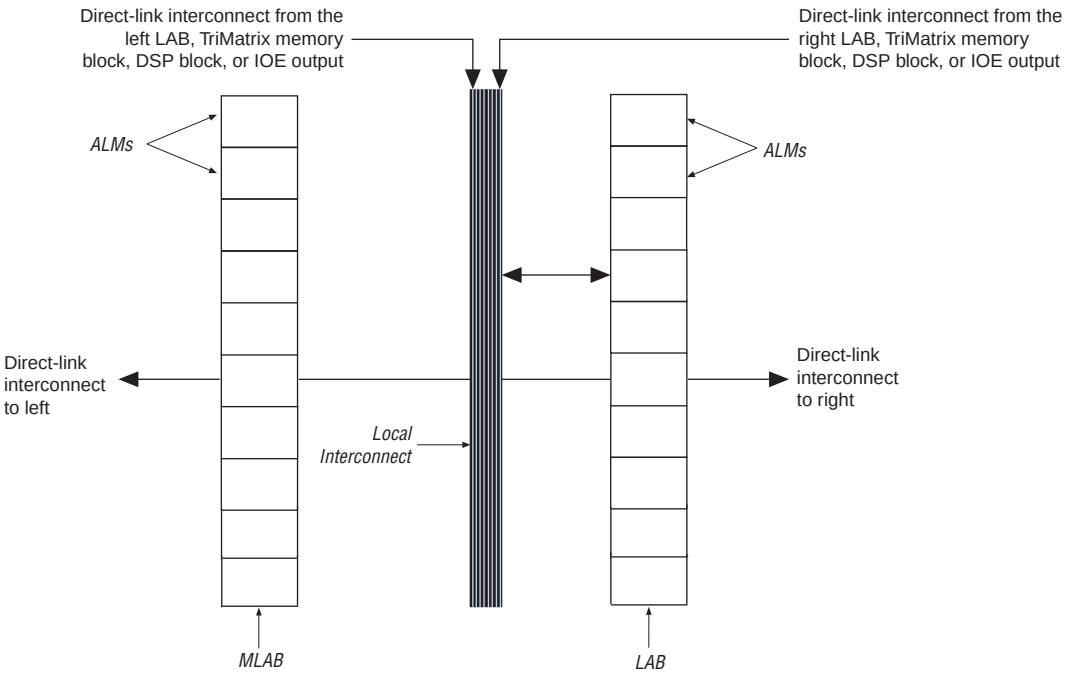
(1) You can use the MLAB ALM as a regular LAB ALM or configure it as a dual-port SRAM, as shown.

LAB Interconnects

The LAB local interconnect can drive ALMs in the same LAB. It is driven by column and row interconnects and ALM outputs in the same LAB. Neighboring LABs/MLABs, M9K RAM blocks, M144K blocks, or DSP blocks from the left or right can also drive the LAB's local interconnect through the direct link connection. The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility. Each LAB can drive 30 ALMs through fast-local and direct-link interconnects.

Figure 2–3 shows the direct-link connection.

Figure 2–3. Direct-Link Connection



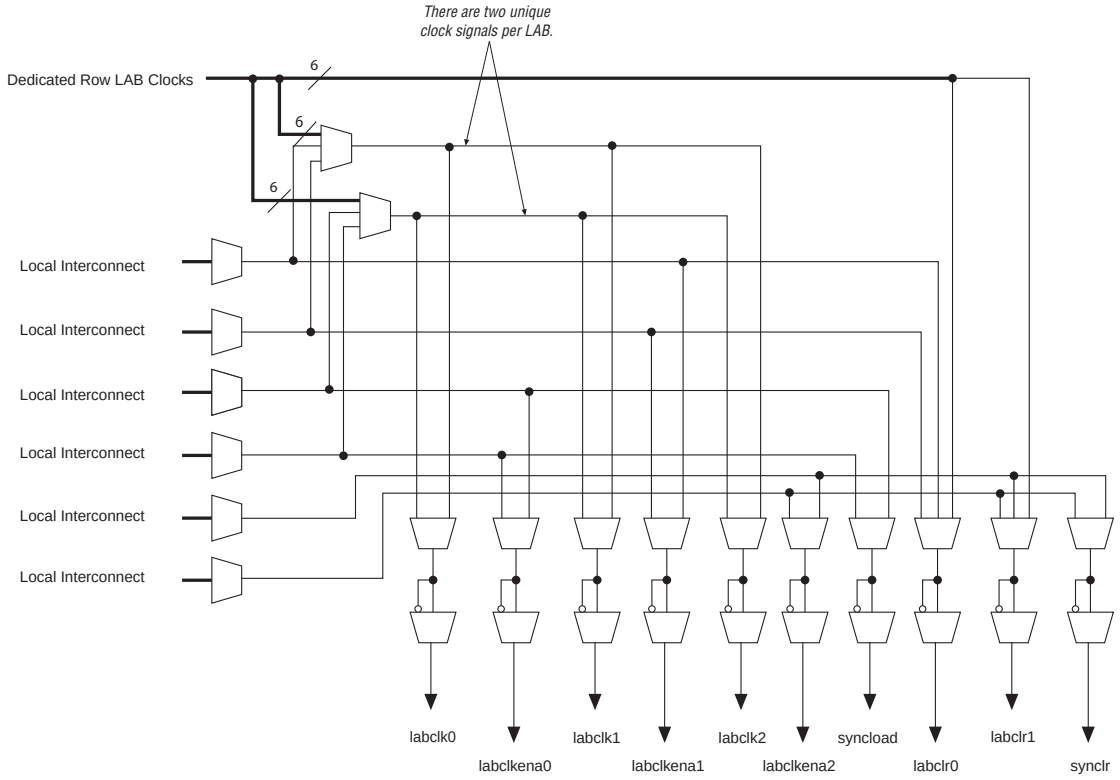
LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its ALMs. Control signals include three clocks, three clock enables, two asynchronous clears, a synchronous clear, and synchronous load control signals. This gives a maximum of 10 control signals at a time. Although you generally use synchronous-load and clear signals when implementing counters, you can also use them with other functions.

Each LAB has two unique clock sources and three clock enable signals, as shown in Figure 2–4. The LAB control block can generate up to three clocks using two clock sources and three clock enable signals. Each LAB's clock and clock enable signals are linked. For example, any ALM in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. If the LAB uses both the rising and falling edges of a clock, it also uses two LAB-wide clock signals. De-asserting the clock enable signal turns off the corresponding LAB-wide clock.

The LAB row clocks [5..0] and LAB local interconnects generate the LAB-wide control signals. The MultiTrack interconnect's inherent low skew allows clock and control signal distribution in addition to data.

Figure 2–4. LAB-Wide Control Signals



Adaptive Logic Modules

The ALM is the basic building block of logic in the Stratix IV architecture. It provides advanced features with efficient logic utilization. Each ALM contains a variety of LUT-based resources that can be divided between two combinational adaptive LUTs (ALUTs) and two registers. With up to eight inputs for the two combinational ALUTs, one ALM can implement various combinations of two functions. This adaptability allows an ALM to be completely backward-compatible with four-input LUT architectures. One ALM can also implement any function with up to six inputs and certain seven-input functions.

In addition to the adaptive LUT-based resources, each ALM contains two programmable registers, two dedicated full adders, a carry chain, a shared arithmetic chain, and a register chain. Through these dedicated resources, an ALM can efficiently implement various arithmetic functions and shift registers. Each ALM drives all types of interconnects: local, row, column, carry chain, shared arithmetic chain, register chain, and direct link. Figure 2–5 shows a high-level block diagram of the Stratix IV ALM.

Figure 2–5. High-Level Block Diagram of the Stratix IV ALM

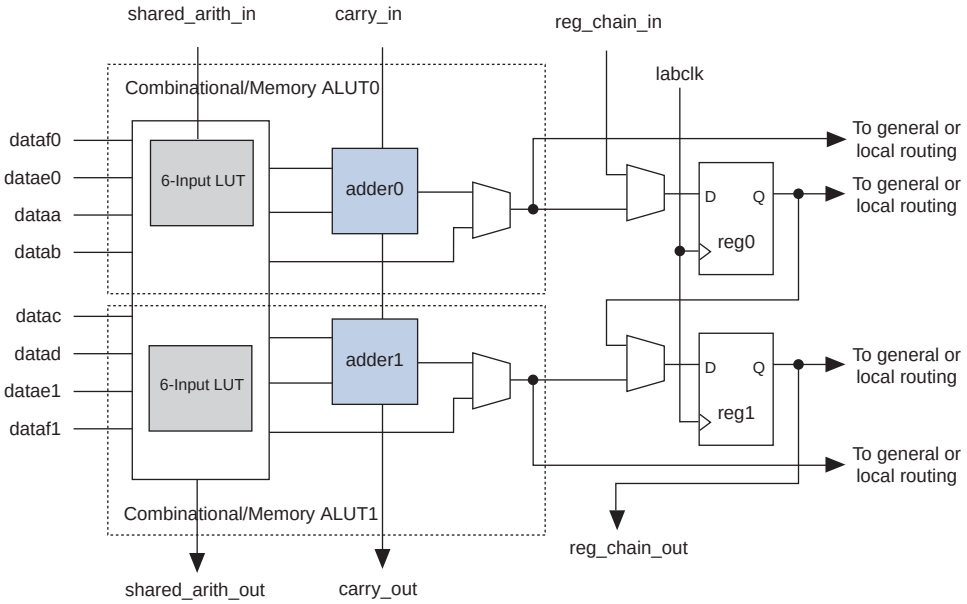
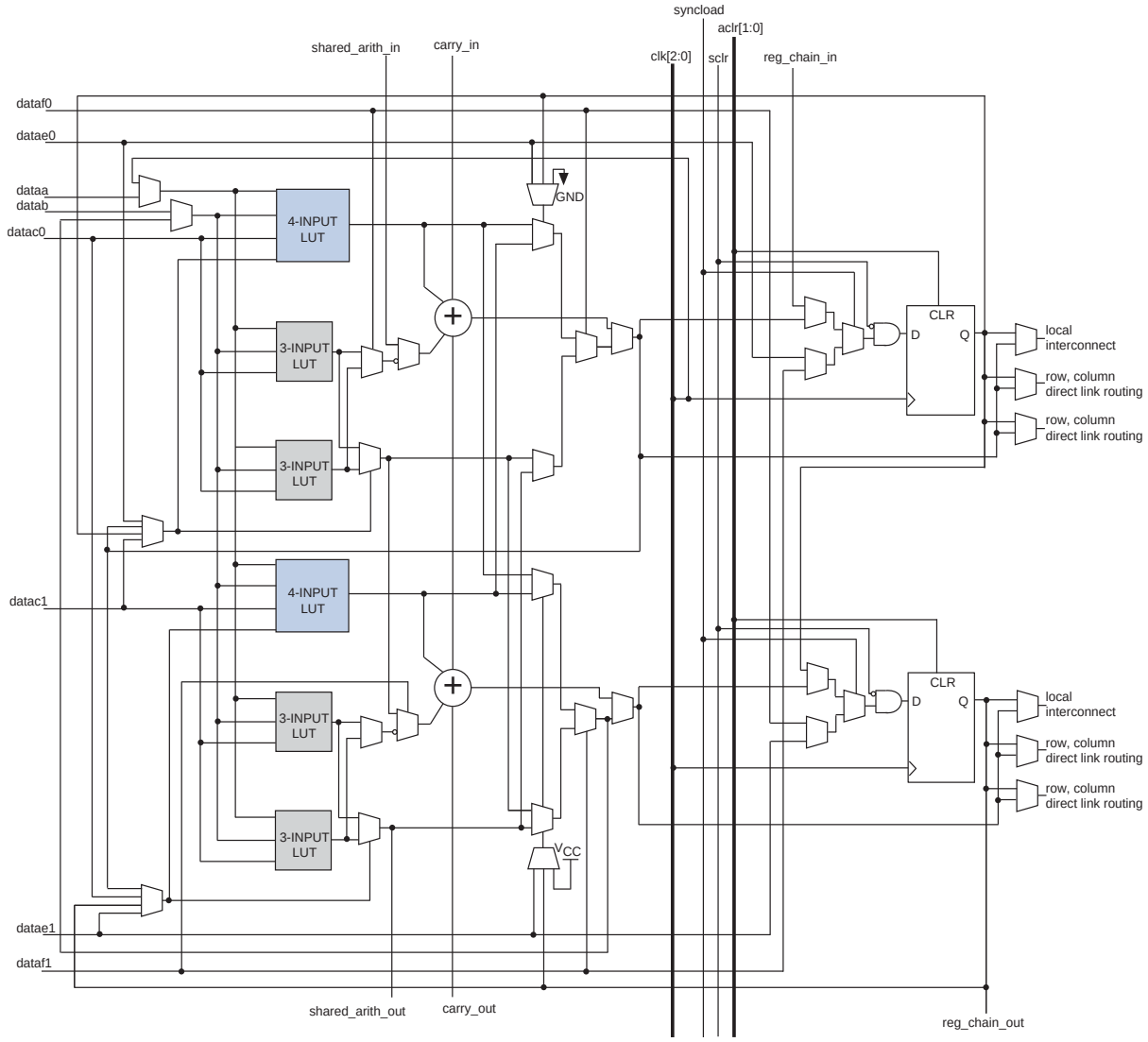


Figure 2–6 shows a detailed view of all the connections in an ALM.

Figure 2–6. Stratix IV ALM Connection Details



One ALM contains two programmable registers. Each register has data, clock, clock enable, synchronous and asynchronous clear, and synchronous load and clear inputs. Global signals, general-purpose I/O pins, or any internal logic can drive the register's clock and clear-control signals. Either general-purpose I/O pins or internal logic can drive the clock enable. For combinational functions, the register is bypassed and the output of the LUT drives directly to the outputs of an ALM.

Each ALM has two sets of outputs that drive the local, row, and column routing resources. The LUT, adder, or register outputs can drive these output drivers (refer to Figure 2–6). For each set of output drivers, two ALM outputs can drive column, row, or direct-link routing connections. One of these ALM outputs can also drive local interconnect resources. This allows the LUT or adder to drive one output while the register drives another output.

This feature, called register packing, improves device utilization because the device can use the register and the combinational logic for unrelated functions. Another special packing mode allows the register output to feed back into the LUT of the same ALM so that the register is packed with its own fan-out LUT. This provides another mechanism for improved fitting. The ALM can also drive out registered and unregistered versions of the LUT or adder output.

ALM Operating Modes

The Stratix IV ALM operates in one of the following modes:

- Normal
- Extended LUT
- Arithmetic
- Shared Arithmetic
- LUT-Register

Each mode uses ALM resources differently. In each mode, eleven available inputs to an ALM—the eight data inputs from the LAB local interconnect, carry-in from the previous ALM or LAB, the shared arithmetic chain connection from the previous ALM or LAB, and the register chain connection—are directed to different destinations to implement the desired logic function. LAB-wide signals provide clock, asynchronous clear, synchronous clear, synchronous load, and clock enable control for the register. These LAB-wide signals are available in all ALM modes.

For more information on the LAB-wide control signals, refer to "LAB Control Signals" on page 2–4.

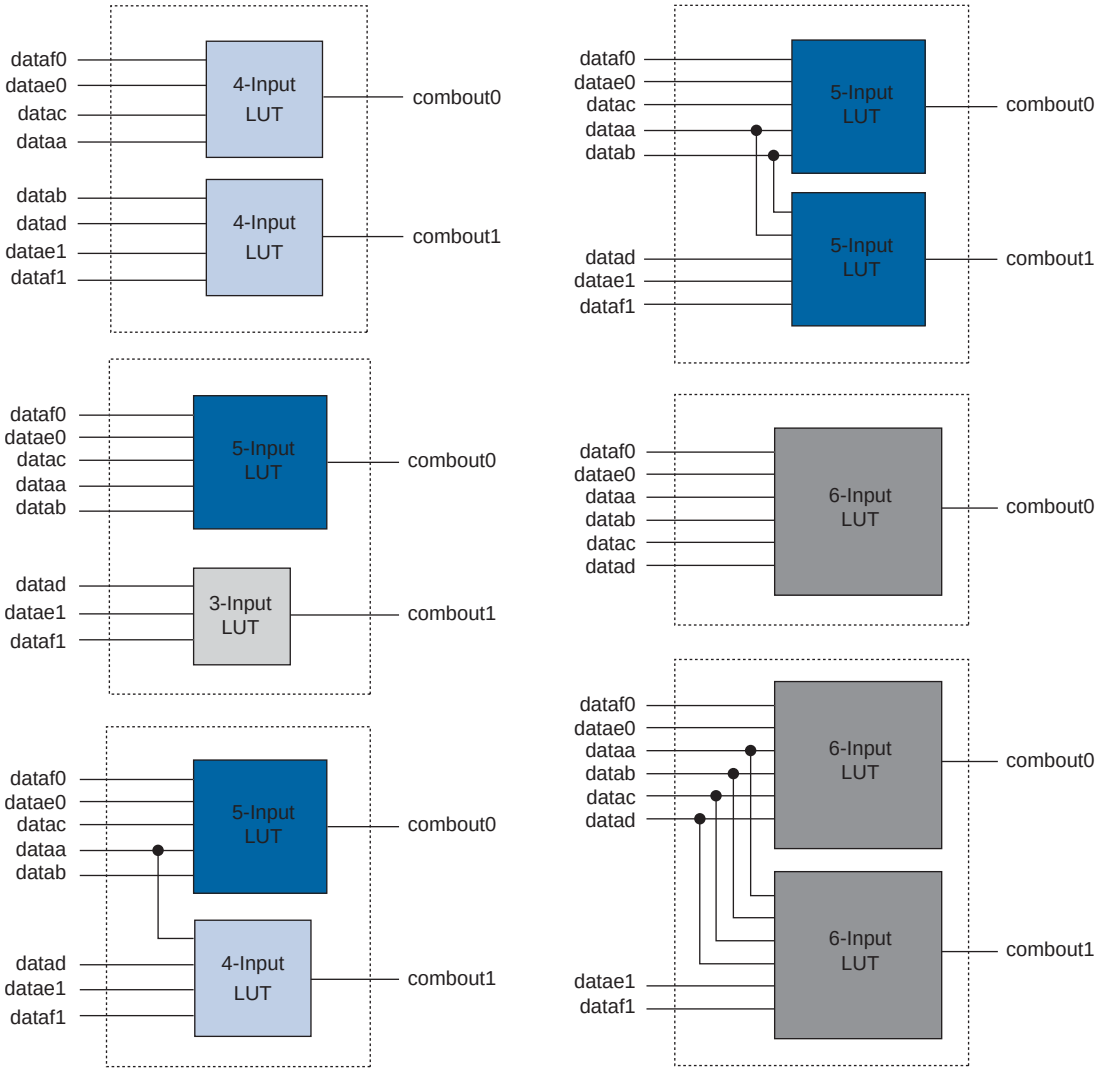
The Quartus II software and supported third-party synthesis tools, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions, automatically choose the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions.

Normal Mode

Normal mode is suitable for general logic applications and combinational functions. In this mode, up to eight data inputs from the LAB local interconnect are inputs to the combinational logic. Normal mode allows two functions to be implemented in one Stratix IV ALM, or a single function of up to six inputs. The ALM can support certain combinations of completely independent functions and various combinations of functions that have common inputs.

Figure 2–7 shows the supported LUT combinations in normal mode.

Figure 2–7. ALM in Normal Mode *(Note 1)*



Note to Figure 2–7:

(1) Combinations of functions with fewer inputs than those shown are also supported. For example, combinations of functions with the following number of inputs are supported: 4 and 3, 3 and 3, 3 and 2, and 5 and 2.

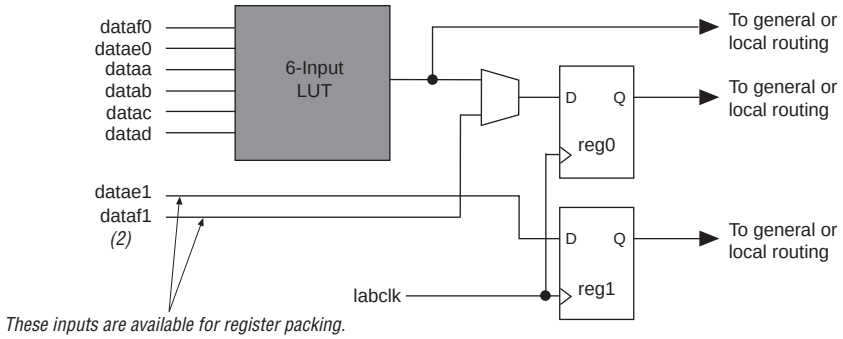
Normal mode provides complete backward-compatibility with four-input LUT architectures.

For the packing of 2 five-input functions into one ALM, the functions must have at least two common inputs. The common inputs are `dataa` and `datab`. The combination of a four-input function with a five-input function requires one common input (either `dataa` or `datab`).

In the case of implementing 2 six-input functions in one ALM, four inputs must be shared and the combinational function must be the same. In a sparsely used device, functions that could be placed in one ALM may be implemented in separate ALMs by the Quartus II software to achieve the best possible performance. As a device begins to fill up, the Quartus II software automatically utilizes the full potential of the Stratix IV ALM. The Quartus II Compiler automatically searches for functions using common inputs or completely independent functions to be placed in one ALM to make efficient use of device resources. In addition, you can manually control resource usage by setting location assignments.

Any six-input function can be implemented using inputs `dataa`, `datab`, `datac`, `datad`, and either `datae0` and `dataf0` or `datae1` and `dataf1`. If `datae0` and `dataf0` are utilized, the output is driven to `register0`, and/or `register0` is bypassed and the data drives out to the interconnect using the top set of output drivers (refer to Figure 2–8). If `datae1` and `dataf1` are used, the output either drives to `register1` or bypasses `register1` and drives to the interconnect using the bottom set of output drivers. The Quartus II Compiler automatically selects the inputs to the LUT. ALMs in normal mode support register packing.

Figure 2–8. Input Function in Normal Mode *(Note 1)*



Notes to Figure 2–8:

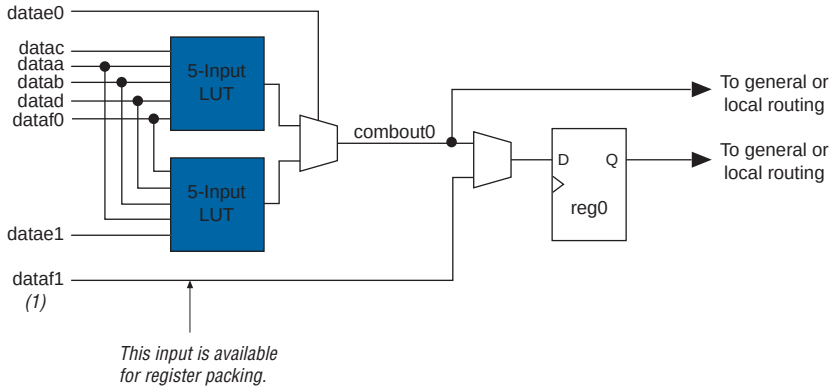
(1) If `datae1` and `dataf1` are used as inputs to a six-input function, `datae0` and `dataf0` are available for register packing.
(2) The `dataf1` input is available for register packing only if the six-input function is unregistered.

Extended LUT Mode

Use extended LUT mode to implement a specific set of seven-input functions. The set must be a 2-to-1 multiplexer fed by two arbitrary five-input functions sharing four inputs. Figure 2–9 shows the template of supported seven-input functions using extended LUT mode. In this mode, if the seven-input function is unregistered, the unused eighth input is available for register packing.

Functions that fit into the template shown in Figure 2–9 occur naturally in designs. These functions often appear in designs as "if-else" statements in Verilog HDL or VHDL code.

Figure 2–9. Template for Supported Seven-Input Functions in Extended LUT Mode



Note to Figure 2–9:

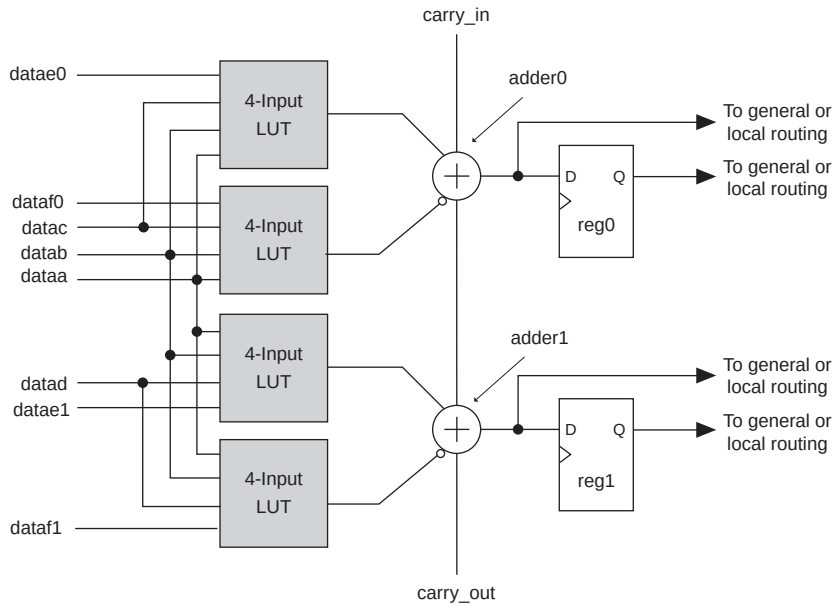
(1) If the seven-input function is unregistered, the unused eighth input is available for register packing. The second register, `reg1`, is not available.

Arithmetic Mode

Arithmetic mode is ideal for implementing adders, counters, accumulators, wide parity functions, and comparators. The ALM in arithmetic mode uses two sets of 2 four-input LUTs along with two dedicated full adders. The dedicated adders allow the LUTs to be available to perform pre-adder logic; therefore, each adder can add the output of 2 four-input functions.

The four LUTs share `dataa` and `datab` inputs. As shown in Figure 2–10, the carry-in signal feeds to `adder0` and the carry-out from `adder0` feeds to the carry-in of `adder1`. The carry-out from `adder1` drives to `adder0` of the next ALM in the LAB. ALMs in arithmetic mode can drive out registered and/or unregistered versions of the adder outputs.

Figure 2–10. ALM in Arithmetic Mode



While operating in arithmetic mode, the ALM can support simultaneous use of the adder's carry output along with combinational logic outputs. In this operation, adder output is ignored. Using the adder with combinational logic output provides resource savings of up to 50% for functions that can use this ability.

Arithmetic mode also offers clock enable, counter enable, synchronous up/down control, add/subtract control, synchronous clear, and synchronous load. The LAB local interconnect data inputs generate the clock enable, counter enable, synchronous up/down, and add/subtract control signals. These control signals are good candidates for the inputs that are shared between the four LUTs in the ALM. The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. These signals can also be individually disabled or enabled per register. The Quartus II software automatically places any registers that are not used by the counter into other LABs.

Carry Chain

The carry chain provides a fast carry function between the dedicated adders in arithmetic or shared-arithmetic mode. The two-bit carry select feature in Stratix IV devices halves the propagation delay of carry chains within the ALM. Carry chains can begin in either the first ALM or the fifth ALM in the LAB. The final carry-out signal is routed to the ALM, where it is fed to local, row, or column interconnects.

The Quartus II Compiler automatically creates carry-chain logic during design processing, or you can create it manually during design entry. Parameterized functions such as LPM functions automatically take advantage of carry chains for the appropriate functions.

The Quartus II Compiler creates carry chains longer than 20 (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically, allowing fast horizontal connections to TriMatrix memory and DSP blocks. A carry chain can continue as far as a full column.

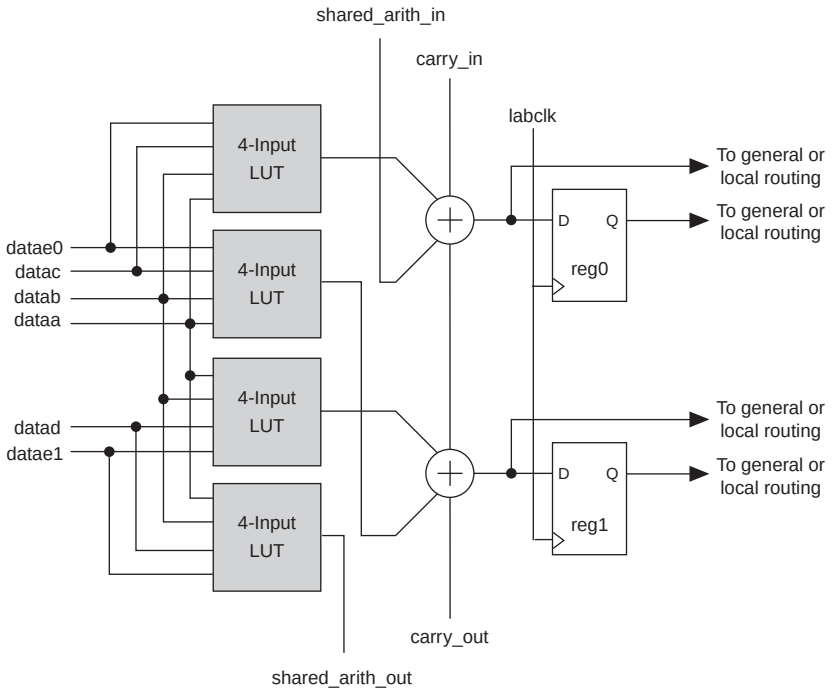
To avoid routing congestion in one small area of the device when a high fan-in arithmetic function is implemented, the LAB can support carry chains that only utilize either the top half or bottom half of the LAB before connecting to the next LAB. This leaves the other half of the ALMs in the LAB available for implementing narrower fan-in functions in normal mode. Carry chains that use the top five ALMs in the first LAB carry into the top half of the ALMs in the next LAB within the column. Carry chains that use the bottom five ALMs in the first LAB carry into the bottom half of the ALMs in the next LAB within the column. In every alternate LAB column, the top half can be bypassed; in the other MLAB columns, the bottom half can be bypassed.

For more information about carry-chain interconnects, refer to "ALM Interconnects" on page 2–17.

Shared Arithmetic Mode

In shared arithmetic mode, the ALM can implement a three-input add within the ALM. In this mode, the ALM is configured with 4 four-input LUTs. Each LUT either computes the sum of three inputs or the carry of three inputs. The output of the carry computation is fed to the next adder (either to `adder1` in the same ALM or to `adder0` of the next ALM in the LAB) using a dedicated connection called the shared arithmetic chain. This shared arithmetic chain can significantly improve the performance of an adder tree by reducing the number of summation stages required to implement an adder tree. Figure 2–11 shows the ALM using this feature.

Figure 2–11. ALM in Shared Arithmetic Mode



You can find adder trees in many different applications. For example, the summation of the partial products in a logic-based multiplier can be implemented in a tree structure. Another example is a correlator function that can use a large adder tree to sum filtered data samples in a given time frame to recover or de-spread data that was transmitted utilizing spread-spectrum technology.

Shared Arithmetic Chain

The shared arithmetic chain available in enhanced arithmetic mode allows the ALM to implement a three-input add. This significantly reduces the resources necessary to implement large adder trees or correlator functions.

The shared arithmetic chains can begin in either the first or sixth ALM in the LAB. The Quartus II Compiler creates shared arithmetic chains longer than 20 (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long shared arithmetic chain runs vertically, allowing fast horizontal connections to the TriMatrix memory and DSP blocks. A shared arithmetic chain can continue as far as a full column.

Similar to the carry chains, the top and bottom halves of shared arithmetic chains in alternate LAB columns can be bypassed. This capability allows the shared arithmetic chain to cascade through half of the ALMs in an LAB while leaving the other half available for narrower fan-in functionality. Every other LAB column is top-half by-passable, while the other LAB columns are bottom-half by-passable.

For more information on shared arithmetic chain interconnect, refer to "ALM Interconnects" on page 2–17.

LUT-Register Mode

LUT-Register mode allows third-register capability within an ALM. Two internal feedback loops allow combinational `ALUT1` to implement the master latch and combinational `ALUT0` to implement the slave latch needed for the third register. The LUT register shares its clock, clock enable, and asynchronous clear sources with the top dedicated register. Figure 2–12 shows the register constructed using two combinational blocks within the ALM.

Figure 2–12. LUT Register from Two Combinational Blocks

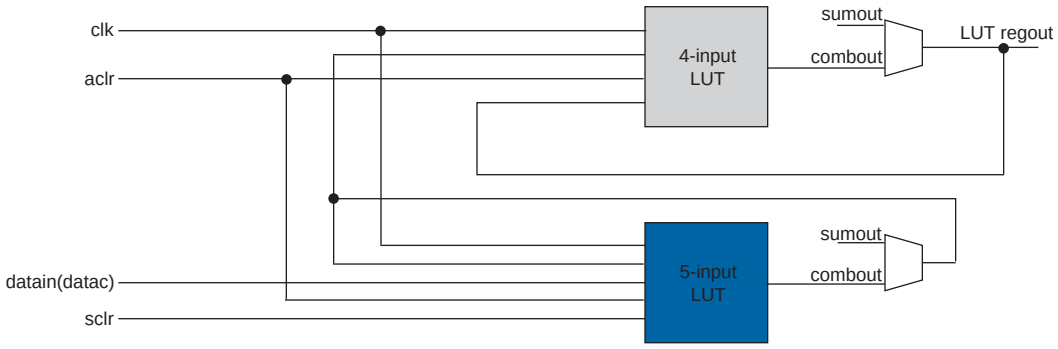
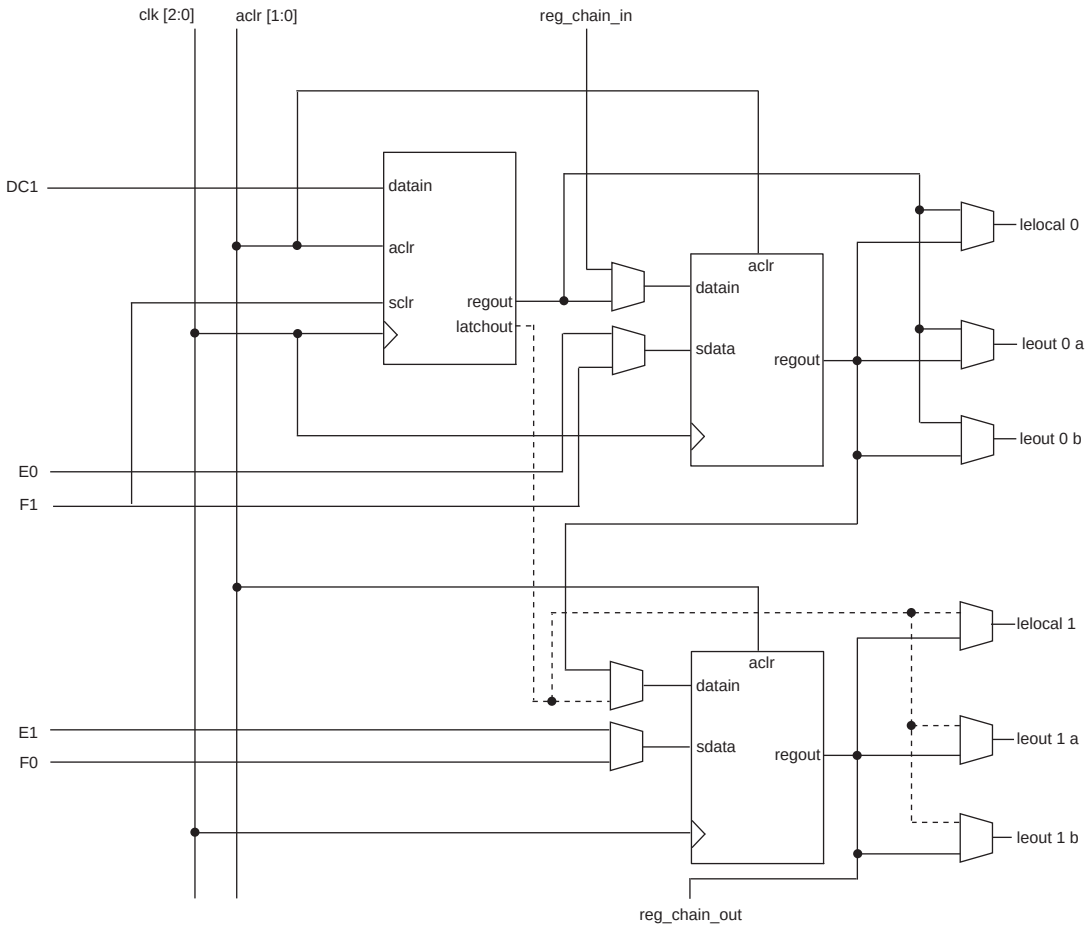


Figure 2–13 shows the ALM in LUT-register mode.

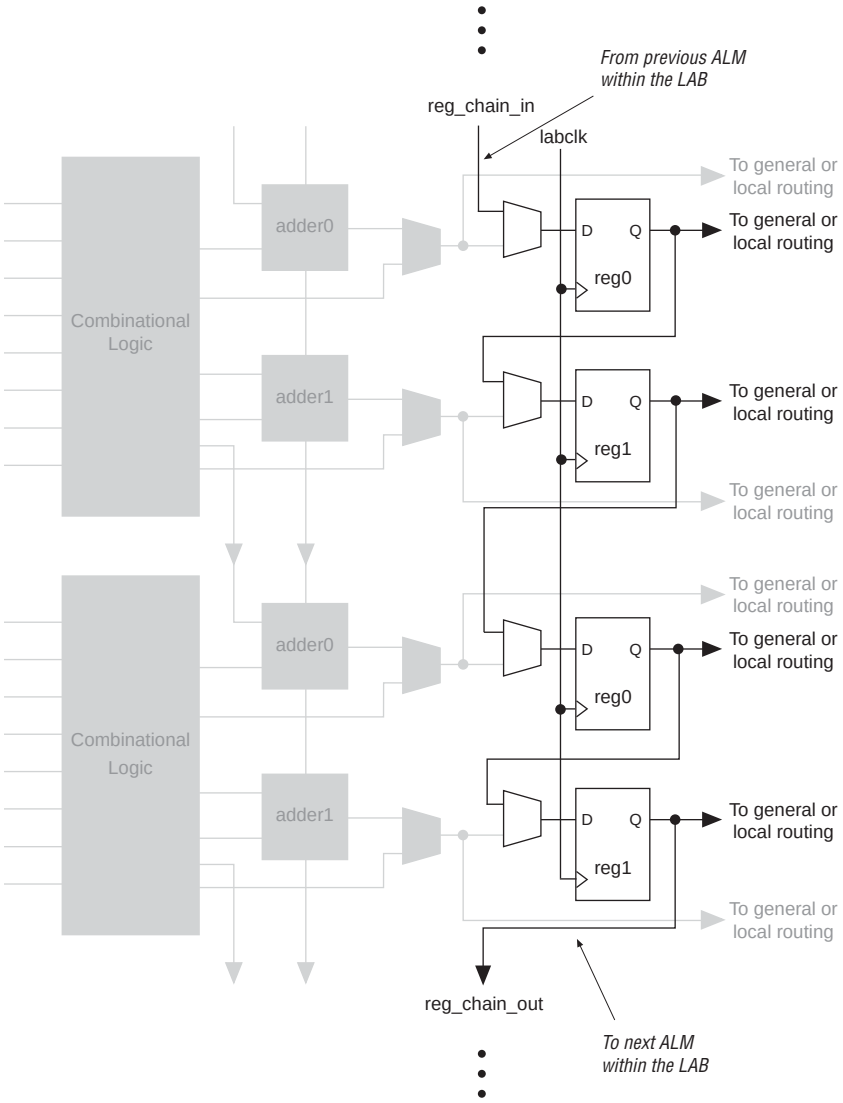
Figure 2–13. ALM in LUT-Register Mode with Three-Register Capability



Register Chain

In addition to general routing outputs, ALMs in the LAB have register-chain outputs. Register-chain routing allows registers in the same LAB to be cascaded together. The register-chain interconnect allows the LAB to use LUTs for a single combinational function and the registers to be used for an unrelated shift-register implementation. These resources speed up connections between ALMs while saving local interconnect resources (refer to Figure 2–14). The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance.

Figure 2–14. Register Chain within the LAB *(Note 1)*



Note to Figure 2–14:

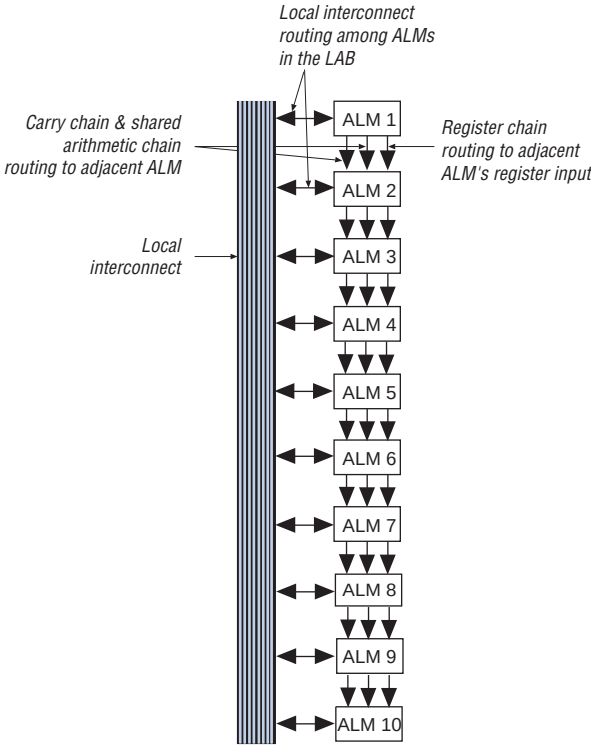
(1) You can use the combinational or adder logic to implement an unrelated, un-registered function.

For more information about register chain interconnect, refer to "ALM Interconnects" on page 2–17.

ALM Interconnects

There are three dedicated paths between ALMs: register cascade, carry chain, and shared arithmetic chain. Stratix IV devices include an enhanced interconnect structure in LABs for routing shared arithmetic chains and carry chains for efficient arithmetic functions. The register chain connection allows the register output of one ALM to connect directly to the register input of the next ALM in the LAB for fast shift registers. These ALM-to-ALM connections bypass the local interconnect. The Quartus II compiler automatically takes advantage of these resources to improve utilization and performance. Figure 2–15 shows the shared arithmetic chain, carry chain, and register chain interconnects.

Figure 2–15. Shared Arithmetic Chain, Carry Chain, and Register Chain Interconnects



Clear and Preset Logic Control

LAB-wide signals control the logic for the register's clear signal. The ALM directly supports an asynchronous clear function. You can achieve the register preset through the Quartus II software's **NOT-gate push-back logic** option. Each LAB supports up to two clears.

Stratix IV devices provide a device-wide reset pin (DEV_CLRn) that resets all the registers in the device. An option set before compilation in the Quartus II software controls this pin. This device-wide reset overrides all other control signals.

LAB Power Management Techniques

The following techniques are used to manage static and dynamic power consumption within the LAB:

- To save AC power, the Quartus II software forces all adder inputs low when ALM adders are not in use.
- Stratix IV LABs operate in high-performance mode or low-power mode. The Quartus II software automatically chooses the appropriate mode for the LAB, based on the design, to optimize speed versus leakage trade-offs.
- Clocks represent a significant portion of dynamic power consumption due to their high switching activity and long paths. The LAB clock that distributes a clock signal to registers within an LAB is a significant contributor to overall clock power consumption. Each LAB's clock and clock enable signal are linked. For example, a combinational ALUT or register in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. To disable an LAB-wide clock power consumption without disabling the entire clock tree, use the LAB-wide clock enable to gate the LAB-wide clock. The Quartus II software automatically promotes register-level clock enable signals to the LAB-level. All registers within the LAB that share a common clock and clock enable are controlled by a shared, gated clock. To take advantage of these clock enables, use a clock-enable construct in your HDL code for the registered logic.

For more information about implementing static and dynamic power consumption within the LAB, refer to the *Power Optimization* chapter in volume 2 of the *Quartus II Handbook*.

Document Revision History

Table 2–1 shows the revision history for this chapter. .

Table 2–1. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated graphics. ■ Minor text edits.	—
June 2009 v2.2	■ Removed the Conclusion section. ■ Added introductory sentences to improve search ability. ■ Minor text edits.	—
March 2009 v2.1	Removed "Referenced Documents" section.	—
November 2008 v2.0	■ Updated Figure 2–6. ■ Made minor editorial changes.	—
May 2008 v1.0	Initial release.	—

3. TriMatrix Embedded Memory Blocks in Stratix IV Devices

SIV51003-3.0

This chapter describes the TriMatrix embedded memory blocks in Stratix® IV devices. TriMatrix embedded memory blocks provide three different sizes of embedded SRAM to efficiently address the needs of Stratix IV FPGA designs. TriMatrix memory includes 640-bit memory logic array blocks (MLABs), 9-Kbit M9K blocks, and 144-Kbit M144K blocks. MLABs have been optimized to implement filter delay lines, small FIFO buffers, and shift registers. You can use the M9K blocks for general purpose memory applications and the M144K blocks for processor code storage, packet buffering, and video frame buffering.

You can independently configure each embedded memory block to be a single- or dual-port RAM, FIFO buffer, ROM, or shift register using the Quartus® II MegaWizard™ Plug-In Manager. You can stitch together multiple blocks of the same type to produce larger memories with minimal timing penalty. TriMatrix memory provides up to 31,491 Kbits of embedded SRAM at up to 600 MHz operation.

This chapter contains the following sections:

Overview

Table 3–1 lists the features supported by the three sizes of TriMatrix memory.

Table 3–1. Summary of TriMatrix Memory Features (Part 1 of 2)

Feature	MLABs	M9K Blocks	M144K Blocks
Maximum performance	600 MHz	600 MHz	540 MHz
Total RAM bits (including parity bits)	640	9216	147,456
Configurations (depth × width)	64 × 8 64 × 9 64 × 10 32 × 16 32 × 18 32 × 20	8K × 1 4K × 2 2K × 4 1K × 8 1K × 9 512 × 16 512 × 18 256 × 32 256 × 36	16K × 8 16K × 9 8K × 16 8K × 18 4K × 32 4K × 36 2K × 64 2K × 72
Parity bits	✓	✓	✓
Byte enable	✓	✓	✓
Packed mode	—	✓	✓

Table 3–1. Summary of TriMatrix Memory Features (Part 2 of 2)

Feature	MLABs	M9K Blocks	M144K Blocks
Address clock enable	✓	✓	✓
Single-port memory	✓	✓	✓
Simple dual-port memory	✓	✓	✓
True dual-port memory	—	✓	✓
Embedded shift register	✓	✓	✓
ROM	✓	✓	✓
FIFO buffer	✓	✓	✓
Simple dual-port mixed width support	—	✓	✓
True dual-port mixed width support	—	✓	✓
Memory Initialization File (**.mif**)	✓	✓	✓
Mixed-clock mode	✓	✓	✓
Power-up condition	Outputs cleared if registered, otherwise reads memory contents	Outputs cleared	Outputs cleared
Register clears	Output registers	Output registers	Output registers
Write/Read operation triggering	Write: Falling clock edges Read: Rising clock edges	Write and Read: Rising clock edges	Write and Read: Rising clock edges
Same-port read-during-write	Outputs set to “don’t care”	Outputs set to “old data” or “new data”	Outputs set to “old data” or “new data”
Mixed-port read-during-write	Outputs set to “old data” or “don’t care”	Outputs set to “old data” or “don’t care”	Outputs set to “old data” or “don’t care”
ECC Support	Soft IP support using the Quartus II software	Soft IP support using the Quartus II software	Built-in support in ×64-wide SDP mode or soft IP support using the Quartus II software

Table 3–2 lists the capacity and distribution of the TriMatrix memory blocks in each Stratix IV family member.

Table 3–2. TriMatrix Memory Capacity and Distribution in Stratix IV Devices ***(Part 1 of 2)***

Device	MLABs	M9K Blocks	M144K Blocks	Total Dedicated RAM Bits (Dedicated Memory Blocks Only) (Kb)	Total RAM Bits (Including MLABs) (Kb)
EP4SE230	4560	1235	22	14,283	17,133
EP4SE360	7072	1248	48	18,144	22,564
EP4SE530	10,624	1280	64	20,736	27,376
EP4SE820	16,261	1610	60	23,130	33,294
EP4SGX70	1452	462	16	6462	7370
EP4SGX110	2112	660	16	8244	9564
EP4SGX180	3515	950	20	11,430	13,627
EP4SGX230	4560	1235	22	14,283	17,133

Table 3–2. TriMatrix Memory Capacity and Distribution in Stratix IV Devices *(Part 2 of 2)*

Device	MLABs	M9K Blocks	M144K Blocks	Total Dedicated RAM Bits (Dedicated Memory Blocks Only) (Kb)	Total RAM Bits (Including MLABs) (Kb)
EP4SGX290	5824	936	36	13,608	17,248
EP4SGX360	7072	1248	48	18,144	22,564
EP4SGX530	10,624	1280	64	20,736	27,376
EP4S40G2	4560	1235	22	14,283	17,133
EP4S40G5	10,624	1280	64	20,736	27,376
EP4S100G2	4560	1235	22	14,283	17,133
EP4S100G3	5824	936	36	13,608	17,248
EP4S100G4	7072	1248	48	18,144	22,564
EP4S100G5	10624	1280	64	20,736	27,376

TriMatrix Memory Block Types

While the M9K and M144K memory blocks are dedicated resources, the MLABs are dual-purpose blocks. They can be configured as regular logic array blocks (LABs) or as MLABs. Ten adaptive logic modules (ALMs) make up one MLAB. Each ALM in an MLAB can be configured as either a 64 × 1 or a 32 × 2 block, resulting in a 64 × 10 or 32 × 20 simple dual-port SRAM block in a single MLAB.

Parity Bit Support

All TriMatrix memory blocks have built-in parity-bit support. The ninth bit associated with each byte can store a parity bit or serve as an additional data bit. No parity function is actually performed on the ninth bit.

Byte Enable Support

All TriMatrix memory blocks support byte enables that mask the input data so that only specific bytes of data are written. The unwritten bytes retain the previously written values. The write enable (`wren`) signals, along with the byte enable (`byteena`) signals, control the RAM blocks' write operations.

The default value for the byte enable signals is high (enabled), in which case writing is controlled only by the write enable signals. The byte enable registers have no clear port. When using parity bits on the M9K and M144K blocks, the byte enable controls all nine bits (eight bits of data plus one parity bit). When using parity bits on the MLAB, the byte-enable controls all 10 bits in the widest mode.

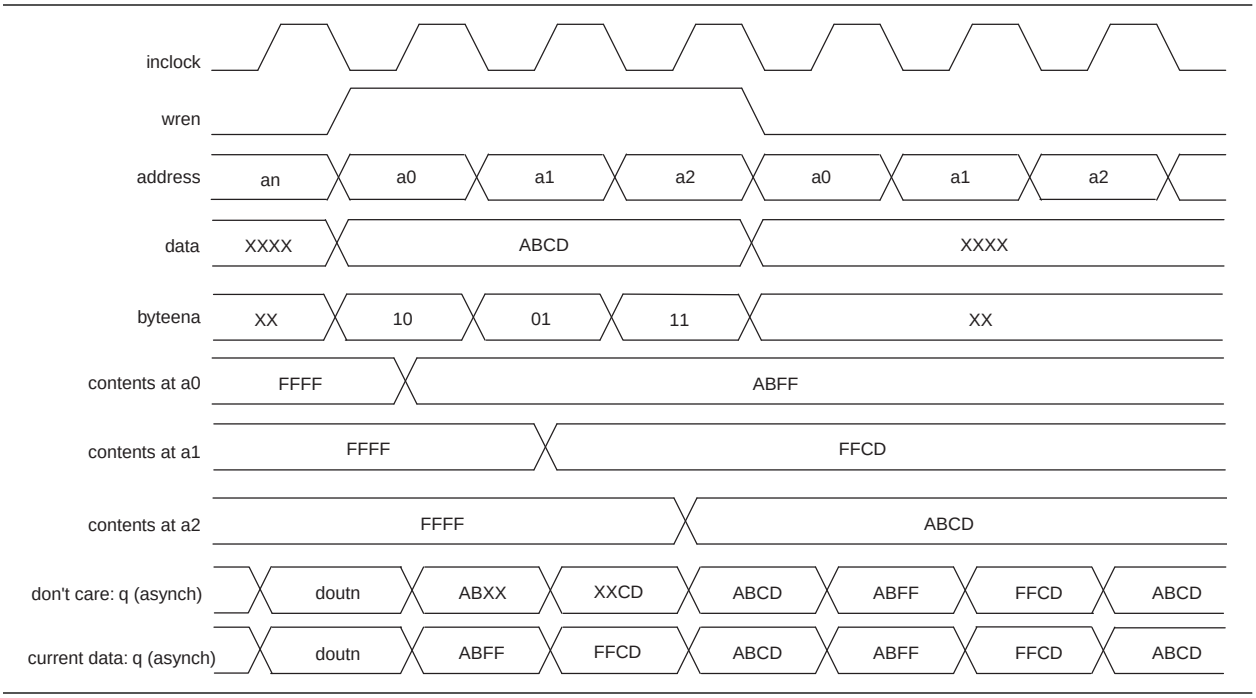
Byte enables operate in a one-hot fashion, with the LSB of the `byteena` signal corresponding to the LSB of the data bus. For example, if you use a RAM block in ×18 mode, with `byteena = 01`, `data[8..0]` is enabled, and `data[17..9]` is disabled. Similarly, if `byteena = 11`, both `data[8..0]` and `data[17..9]` are enabled. Byte enables are active high.

You cannot use the byte enable feature when using the error correction coding (ECC) feature on M144K blocks.

Figure 3–1 shows how the write enable (wren) and byte enable (byteena) signals control the operations of the RAM blocks.

When a byte-enable bit is de-asserted during a write cycle, the corresponding data byte output can appear as either a "don't care" value or the current data at that location. The output value for the masked byte is controllable using the Quartus II software. When a byte-enable bit is asserted during a write cycle, the corresponding data byte output also depends on the setting chosen in the Quartus II software.

Figure 3–1. Byte Enable Functional Waveform



Packed Mode Support

Stratix IV M9K and M144K blocks support packed mode. The packed mode feature packs two independent single-port RAMs into one memory block. The Quartus II software automatically implements packed mode where appropriate by placing the physical RAM block into true dual-port mode and using the MSB of the address to distinguish between the two logical RAMs. The size of each independent single-port RAM must not exceed half of the target block size.

Address Clock Enable Support

All Stratix IV memory blocks support address clock enable, which holds the previous address value for as long as the signal is enabled (`addressstall` = `1`). When the memory blocks are configured in dual-port mode, each port has its own independent address clock enable. The default value for the address clock enable signals is low (disabled).

Figure 3–2 shows an address clock enable block diagram. The address clock enable is referred to by the port name `addressstall`.

Figure 3–2. Address Clock Enable

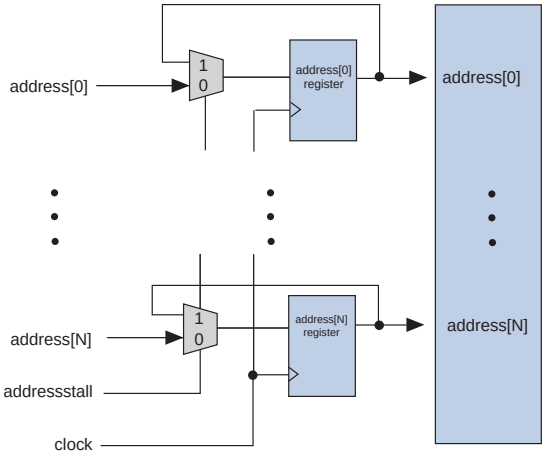


Figure 3–3 shows the address clock enable waveform during the read cycle.

Figure 3–3. Address Clock Enable During Read Cycle Waveform

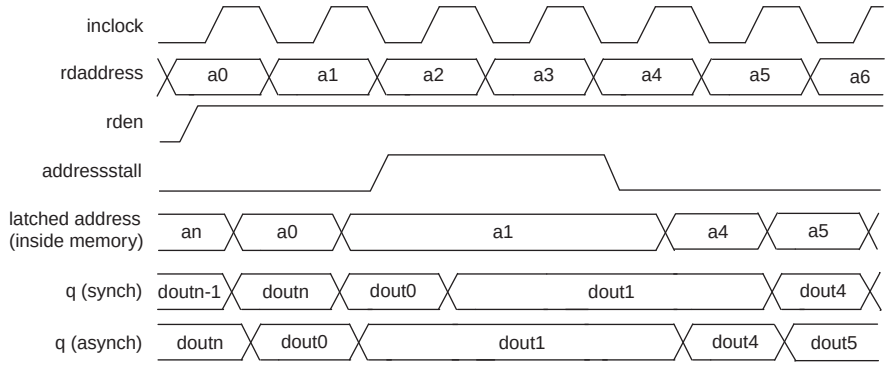
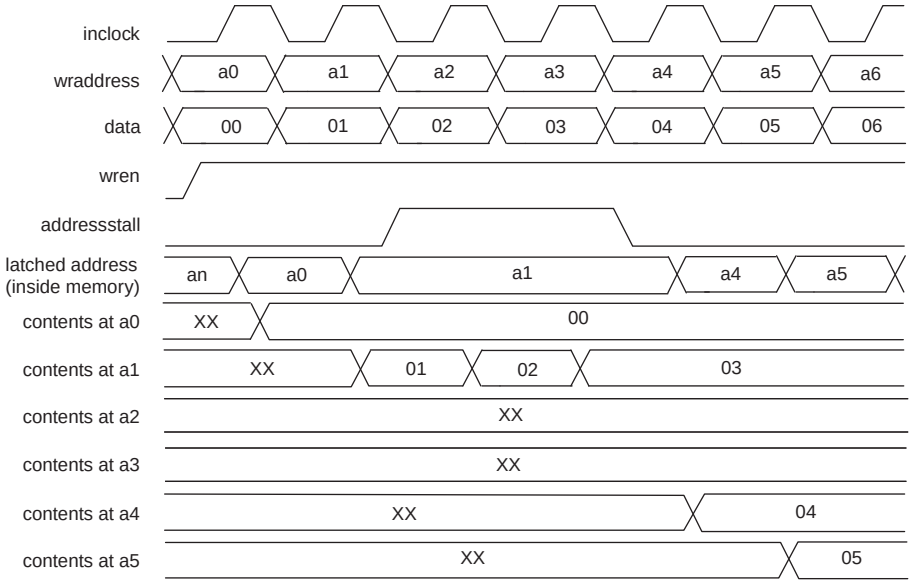


Figure 3–4 shows the address clock enable waveform during the write cycle.

Figure 3–4. Address Clock Enable During the Write Cycle Waveform



Mixed Width Support

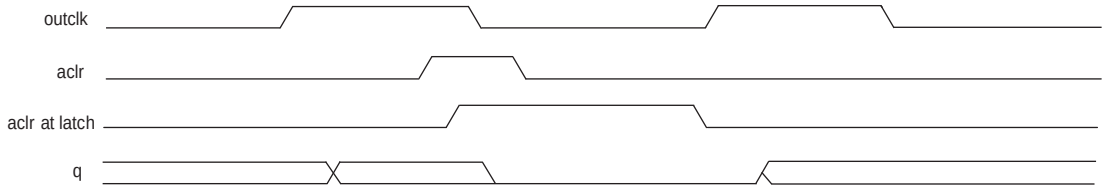
M9K and M144K memory blocks support mixed data widths inherently. MLABs can support mixed data widths through emulation using the Quartus II software. When using simple dual-port, true dual-port, or FIFO modes, mixed width support allows you to read and write different data widths to a memory block. For more information about the different widths supported per memory mode, refer to "Memory Modes" on page 3–8.

MLABs do not support mixed-width FIFO mode.

Asynchronous Clear

Stratix IV TriMatrix memory blocks support asynchronous clears on output latches and output registers. Therefore, if your RAM is not using output registers, you can still clear the RAM outputs using the output latch asynchronous clear. Figure 3–5 shows a waveform of the output latch asynchronous clear function.

Figure 3–5. Output Latch Asynchronous Clear Waveform



You can selectively enable asynchronous clears per logical memory using the Quartus II RAM MegaWizard Plug-In Manager.

For more information, refer to the *RAM Megafunction User Guide*.

Error Correction Code (ECC) Support

Stratix IV M144K blocks have built-in support for error correction code (ECC) when in ×64-wide simple dual-port mode. ECC allows you to detect and correct data errors in the memory array. The M144K blocks have a single-error-correction double-error-detection (SECDED) implementation. SECDED can detect and fix a single bit error in a 64-bit word, or detect two bit errors in a 64-bit word. It cannot detect three or more errors.

The M144K ECC status is communicated using a three-bit status flag `eccstatus[2..0]`. The status flag can be either registered or unregistered. When registered, it uses the same clock and asynchronous clear signals as the output registers. When unregistered, it cannot be asynchronously cleared.

Table 3–3 lists the truth table for the ECC status flags.

Table 3–3. Truth Table for ECC Status Flags

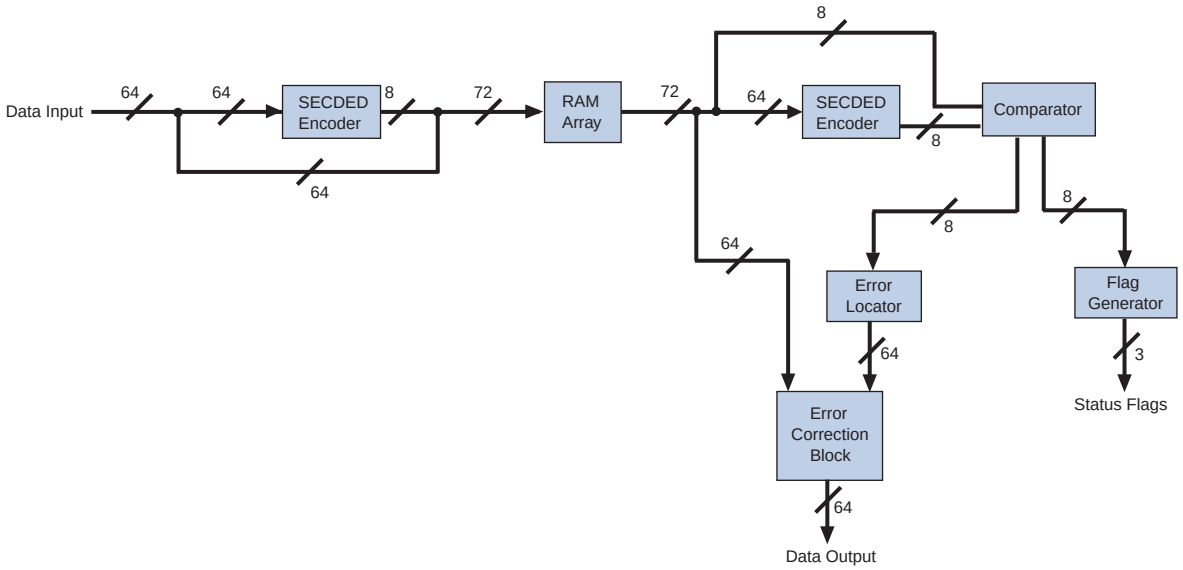
Status	eccstatus[2]	eccstatus[1]	eccstatus[0]
No error	0	0	0
Single error and fixed	0	1	1
Double error and no fix	1	0	1
Illegal	0	0	1
Illegal	0	1	0
Illegal	1	0	0
Illegal	1	1	X

You cannot use the byte enable feature when ECC is engaged.

Read during write “old data mode” is not supported when ECC is engaged.

Figure 3–6 shows a diagram of the ECC block of the M144K block.

Figure 3–6. ECC Block Diagram of the M144K Block



Memory Modes

Stratix IV TriMatrix memory blocks allow you to implement fully synchronous SRAM memory in multiple modes of operation. M9K and M144K blocks do not support asynchronous memory (unregistered inputs). MLABs support asynchronous (flow-through) read operations.

Depending on which TriMatrix memory block you target, you can use the following modes:

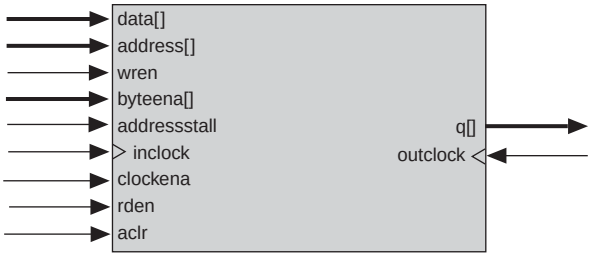
- Single-port
- Simple dual-port
- True dual-port
- Shift-register
- ROM
- FIFO

When using the memory blocks in ROM, single-port, simple dual-port, or true dual-port mode, you can corrupt the memory contents if you violate the setup or hold-time on any of the memory block input registers. This applies to both read and write operations.

Single-Port RAM

All TriMatrix memory blocks support single-port mode. Single-port mode allows you to do either one-read or one-write operation at a time. Simultaneous reads and writes are not supported in single-port mode. Figure 3–7 shows the single-port RAM configuration.

Figure 3–7. Single-Port RAM *(Note 1)*



Note to Figure 3–7:

(1) You can implement two single-port memory blocks in a single M9K or M144K block. For more information, refer to "Packed Mode Support" on page 3–5.

During a write operation, RAM output behavior is configurable. If you use the read-enable signal and perform a write operation with the read enable de-activated, the RAM outputs retain the values they held during the most recent active read enable. If you activate read enable during a write operation, or if you are not using the read-enable signal at all, the RAM outputs either show the "new data" being written, the "old data" at that address, or a "don't care" value. To choose the desired behavior, set the read-during-write behavior to either **new data**, **old data**, or **don't care** in the RAM MegaWizard Plug-In Manager in the Quartus II software. For more information, refer to "Read-During-Write Behavior" on page 3–19.

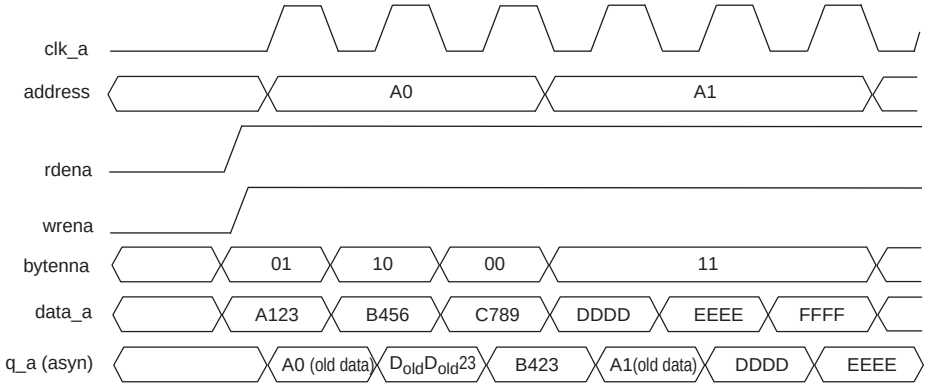
Table 3–4 lists the possible port width configurations for TriMatrix memory blocks in single-port mode.

Table 3–4. Port Width Configurations for MLABs, M9K, and M144K Blocks (Single-Port Mode)

	MLABs	M9K Blocks	M144K Blocks
Port Width Configurations	64 × 8 64 × 9 64 × 10 32 × 16 32 × 18 32 × 20	8K × 1 4K × 2 2K × 4 1K × 8 1K × 9 512 × 16 512 × 18 256 × 32 256 × 36	16K × 8 16K × 9 8K × 16 8K × 18 4K × 32 4K × 36 2K × 64 2K × 72

Figure 3–8 shows timing waveforms for read and write operations in single-port mode with unregistered outputs. Registering the RAM's outputs simply delays the q output by one clock cycle.

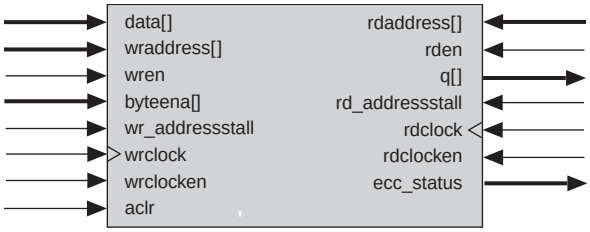
Figure 3–8. Timing Waveform for Read-Write Operations (Single-Port Mode)



Simple Dual-Port Mode

All TriMatrix memory blocks support simple dual-port mode. Simple dual-port mode allows you to perform one-read and one-write operation to different locations at the same time. Write operation happens on port A; read operation happens on port B. Figure 3–9 shows a simple dual-port configuration.

Figure 3–9. Stratix IV Simple Dual-Port Memory *(Note 1)*



Note to Figure 3–9:

(1) Simple dual-port RAM supports input/output clock mode in addition to read/write clock mode.

Simple dual-port mode supports different read and write data widths (mixed-width support). Table 3–5 lists the mixed width configurations for M9K blocks in simple dual-port mode. MLABs do not have native support for mixed-width operation. The Quartus II software implements mixed-width memories in MLABs by using more than one MLAB.

Table 3–5. M9K Block Mixed-Width Configurations (Simple Dual-Port Mode) (Part 1 of 2)

Read Port	Write Port								
	8K × 1	4K × 2	2K × 4	1K × 8	512 × 16	256 × 32	1K × 9	512 × 18	256 × 36
8K × 1	✓	✓	✓	✓	✓	✓	—	—	—
4K × 2	✓	✓	✓	✓	✓	✓	—	—	—
2K × 4	✓	✓	✓	✓	✓	✓	—	—	—

Table 3–5. M9K Block Mixed-Width Configurations (Simple Dual-Port Mode) (Part 2 of 2)

Read Port	Write Port								
	8K × 1	4K × 2	2K × 4	1K × 8	512 × 16	256 × 32	1K × 9	512 × 18	256 × 36
1K × 8	✓	✓	✓	✓	✓	✓	—	—	—
512 × 16	✓	✓	✓	✓	✓	✓	—	—	—
256 × 32	✓	✓	✓	✓	✓	✓	—	—	—
1K × 9	—	—	—	—	—	—	✓	✓	✓
512 × 18	—	—	—	—	—	—	✓	✓	✓
256 × 36	—	—	—	—	—	—	✓	✓	✓

Table 3–6 lists the mixed-width configurations for M144K blocks in simple dual-port mode.

Table 3–6. M144K Block Mixed-Width Configurations (Simple Dual-Port Mode)

Read Port	Write Port							
	16K × 8	8K × 16	4K × 32	2K × 64	16K × 9	8K × 18	4K × 36	2K × 72
16K × 8	✓	✓	✓	✓	—	—	—	—
8K × 16	✓	✓	✓	✓	—	—	—	—
4K × 32	✓	✓	✓	✓	—	—	—	—
2K × 64	✓	✓	✓	✓	—	—	—	—
16K × 9	—	—	—	—	✓	✓	✓	✓
8K × 18	—	—	—	—	✓	✓	✓	✓
4K × 36	—	—	—	—	✓	✓	✓	✓
2K × 72	—	—	—	—	✓	✓	✓	✓

In simple dual-port mode, M9K and M144K blocks support separate write-enable and read-enable signals. You can save power by keeping the read-enable signal low (inactive) when not reading. Read-during-write operations to the same address can either output a "don't care" value or "old data" value. To choose the desired behavior, set the read-during-write behavior to either **don't care** or **old data** in the RAM MegaWizard Plug-In Manager in the Quartus II software. For more information, refer to "Read-During-Write Behavior" on page 3–19.

MLABs only support a write-enable signal. Read-during-write behavior for the MLABs can be either "don't care" or "old data". The available choices depend on the configuration of the MLAB.

Figure 3–10 shows timing waveforms for read and write operations in simple dual-port mode with unregistered outputs. Registering the RAM outputs simply delays the q output by one clock cycle.

Figure 3–10. Simple Dual-Port Timing Waveforms

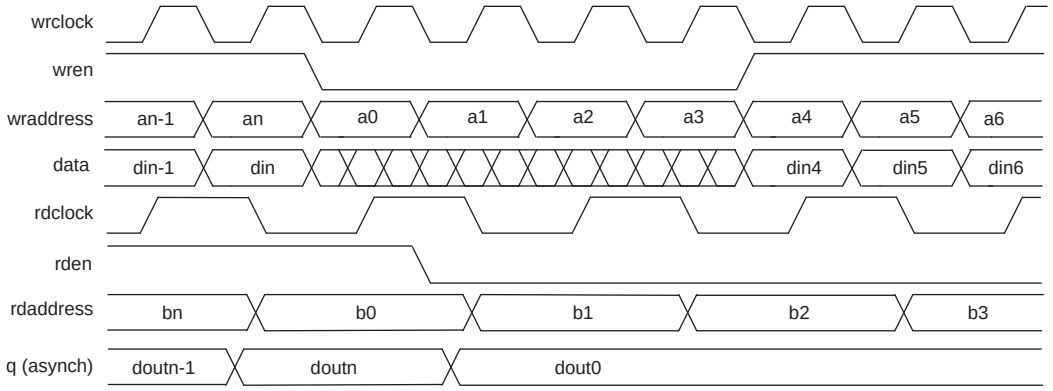
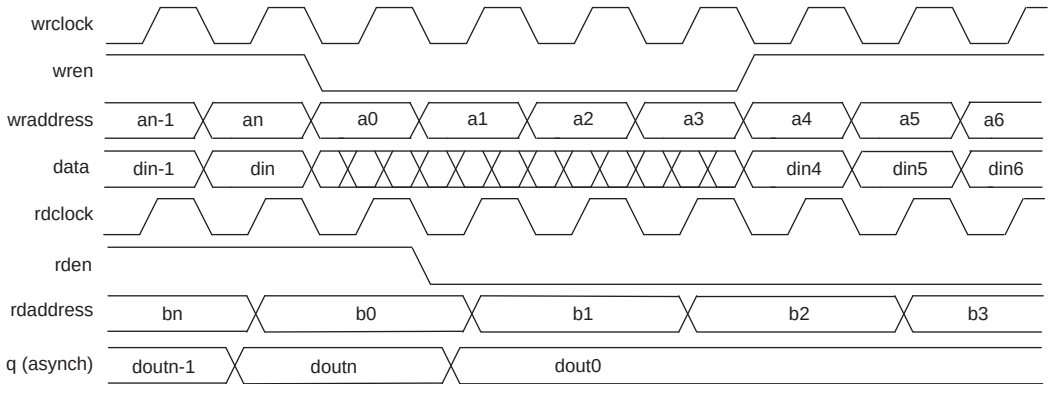


Figure 3–11 shows timing waveforms for read and write operations in mixed-port mode with unregistered outputs.

Figure 3–11. Mixed-Port Read-During-Write Timing Waveforms

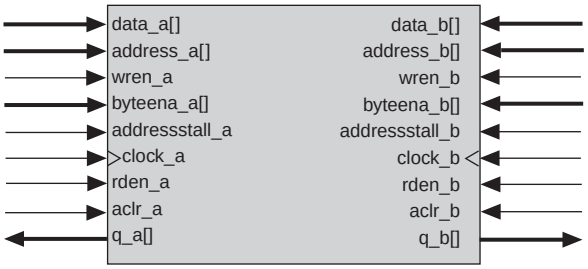


True Dual-Port Mode

Stratix IV M9K and M144K blocks support true dual-port mode. Sometimes called bi-directional dual-port, this mode allows you to perform any combination of two port operations: two reads, two writes, or one read and one write at two different clock frequencies.

Figure 3–12 shows the true dual-port RAM configuration.

Figure 3–12. Stratix IV True Dual-Port Memory *(Note 1)*



Note to Figure 3–12:

(1) True dual-port memory supports input/output clock mode in addition to independent clock mode.

The widest bit configuration of the M9K and M144K blocks in true dual-port mode is as follows:

- 512 × 16-bit (×18-bit with parity) (M9K)
- 4K × 32-bit (×36-bit with parity) (M144K)

Wider configurations are unavailable because the number of output drivers is equivalent to the maximum bit width of the respective memory block. Because true dual-port RAM has outputs on two ports, its maximum width equals half of the total number of output drivers. Table 3–7 lists the possible M9K block mixed-port width configurations in true dual-port mode.

Table 3–7. M9K Block Mixed-Width Configuration (True Dual-Port Mode)

Read Port	Write Port						
	8K × 1	4K × 2	2K × 4	1K × 8	512 × 16	1K × 9	512 × 18
8K × 1	✓	✓	✓	✓	✓	—	—
4K × 2	✓	✓	✓	✓	✓	—	—
2K × 4	✓	✓	✓	✓	✓	—	—
1K × 8	✓	✓	✓	✓	✓	—	—
512 × 16	✓	✓	✓	✓	✓	—	—
1K × 9	—	—	—	—	—	✓	✓
512 × 18	—	—	—	—	—	✓	✓

Table 3–8 lists the possible M144K block mixed-port width configurations in true dual-port mode.

Table 3–8. M144K Block Mixed-Width Configurations (True Dual-Port Mode) (Part 1 of 2)

Read Port	Write Port					
	16K × 8	8K × 16	4K × 32	16K × 9	8K × 18	4K × 36
16K × 8	✓	✓	✓	—	—	—
8K × 16	✓	✓	✓	—	—	—
4K × 32	✓	✓	✓	—	—	—

Table 3–8. M144K Block Mixed-Width Configurations (True Dual-Port Mode) (Part 2 of 2)

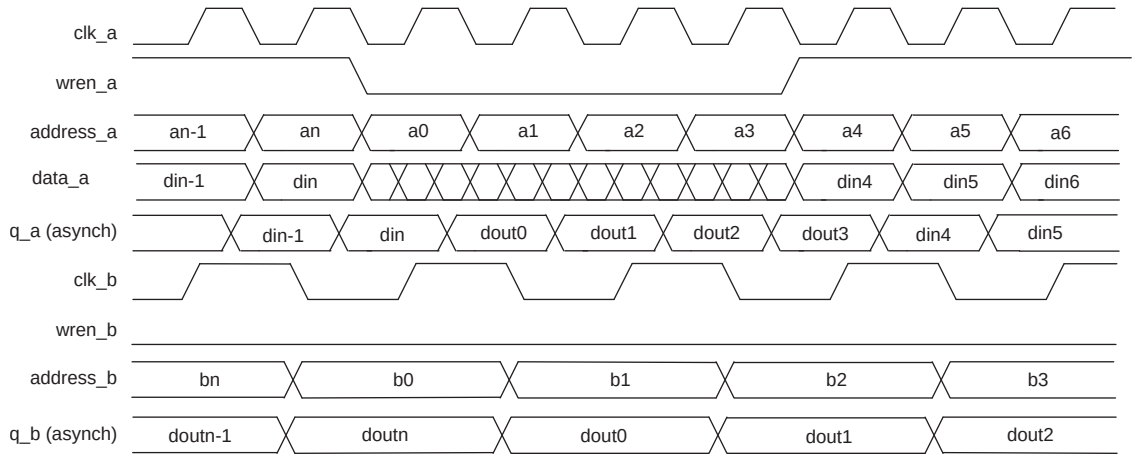
Read Port	Write Port					
	16K × 8	8K × 16	4K × 32	16K × 9	8K × 18	4K × 36
16K × 9	—	—	—	✓	✓	✓
8K × 18	—	—	—	✓	✓	✓
4K × 36	—	—	—	✓	✓	✓

In true dual-port mode, M9K and M144K blocks support separate write-enable and read-enable signals. You can save power by keeping the read-enable signal low (inactive) when not reading. Read-during-write operations to the same address can either output "new data" at that location or "old data". To choose the desired behavior, set the read-during-write behavior to either **new data** or **old data** in the RAM MegaWizard Plug-In Manager in the Quartus II software. For more information, refer to "Read-During-Write Behavior" on page 3–19.

In true dual-port mode, you can access any memory location at any time from either port. When accessing the same memory location from both ports, you must avoid possible write conflicts. A write conflict happens when you attempt to write to the same address location from both ports at the same time. This results in unknown data being stored to that address location. No conflict resolution circuitry is built into the Stratix IV TriMatrix memory blocks. You must handle address conflicts external to the RAM block.

Figure 3–13 shows true dual-port timing waveforms for the write operation at port A and read operation at port B, with the **Read-During-Write** behavior set to **new data**. Registering the RAM's outputs simply delays the q outputs by one clock cycle.

Figure 3–13. True Dual-Port Timing Waveform



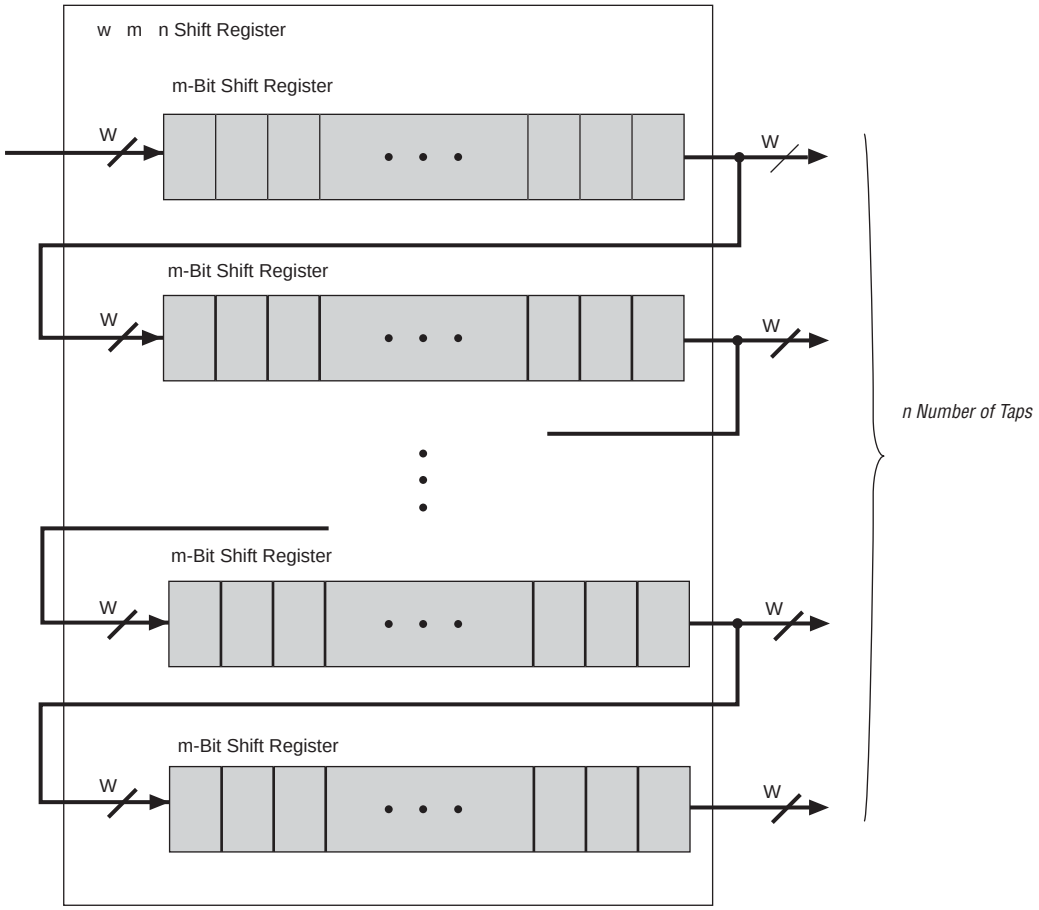
Shift-Register Mode

All Stratix IV memory blocks support shift register mode. Embedded memory block configurations can implement shift registers for digital signal processing (DSP) applications, such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto- and cross-correlation functions. These and other DSP applications require local data storage, traditionally implemented with standard flipflops that quickly exhaust many logic cells for large shift registers. A more efficient alternative is to use embedded memory as a shift-register block, which saves logic cell and routing resources.

The size of a shift register ($w \times m \times n$) is determined by the input data width (w), the length of the taps (m), and the number of taps (n). You can cascade memory blocks to implement larger shift registers.

Figure 3–14 shows the TriMatrix memory block in shift-register mode.

Figure 3–14. Shift-Register Memory Configuration



ROM Mode

All Stratix IV TriMatrix memory blocks support ROM mode. A **.mif** file initializes the ROM contents of these blocks. The address lines of the ROM are registered on M9K and M144K blocks, but can be unregistered on MLABs. The outputs can be registered or unregistered. Output registers can be asynchronously cleared. The ROM read operation is identical to the read operation in the single-port RAM configuration.

FIFO Mode

All TriMatrix memory blocks support FIFO mode. MLABs are ideal for designs with many small, shallow FIFO buffers. To implement FIFO buffers in your design, use the Quartus II software FIFO MegaWizard Plug-In Manager. Both single- and dual-clock (asynchronous) FIFO buffers are supported.

For more information about implementing FIFO buffers, refer to the *Single- and Dual-Clock FIFO Megafunctions User Guide*.

MLABs do not support mixed-width FIFO mode.

Clocking Modes

Stratix IV TriMatrix memory blocks support the following clocking modes:

- Independent
- Input/output
- Read/write
- Single clock

Violating the setup or hold time on the memory block address registers could corrupt memory contents. This applies to both read and write operations.

Table 3–9 lists the clocking mode versus memory mode support matrix.

Table 3–9. TriMatrix Memory Clock Modes

Clocking Mode	True Dual-Port Mode	Simple Dual-Port Mode	Single-Port Mode	ROM Mode	FIFO Mode
Independent	✓	—	—	✓	—
Input/output	✓	✓	✓	✓	—
Read/write	—	✓	—	—	✓
Single clock	✓	✓	✓	✓	✓

Independent Clock Mode

Stratix IV TriMatrix memory blocks can implement independent clock mode for true dual-port memories. In this mode, a separate clock is available for each port (clock A and clock B). Clock A controls all registers on the port A side; clock B controls all registers on the port B side. Each port also supports independent clock enables for both port A and port B registers, respectively. Asynchronous clears are supported only for output latches and output registers on both ports.

Input/Output Clock Mode

Stratix IV TriMatrix memory blocks can implement input/output clock mode for true dual-port and simple dual-port memories. In this mode, an input clock controls all registers related to the data input to the memory block including data, address, byte enables, read enables, and write enables. An output clock controls the data output registers. Asynchronous clears are available on output latches and output registers only.

Read/Write Clock Mode

Stratix IV TriMatrix memory blocks can implement read/write clock mode for simple dual-port memories. In this mode, a write clock controls the data-input, write-address, and write-enable registers. Similarly, a read clock controls the data-output, read-address, and read-enable registers. The memory blocks support independent clock enables for both the read and write clocks. Asynchronous clears are available on data output latches and registers only.

When using read/write clock mode, if you perform a simultaneous read/write to the same address location, the output read data is unknown. If you require the output data to be a known value, use either single-clock mode or input/output clock mode and choose the appropriate read-during-write behavior in the Megawizard Plug-In Manager.

Single Clock Mode

Stratix IV TriMatrix memory blocks can implement single-clock mode for true dual-port, simple dual-port, and single-port memories. In this mode, a single clock, together with a clock enable, is used to control all registers of the memory block. Asynchronous clears are available on output latches and output registers only.

Design Considerations

This section describes guidelines for designing with TriMatrix memory blocks.

Selecting TriMatrix Memory Blocks

The Quartus II software automatically partitions user-defined memory into embedded memory blocks by taking into account both speed and size constraints placed on your design. For example, the Quartus II software may spread memory out across multiple memory blocks when resources are available to increase the performance of the design. You can manually assign memory to a specific block size using the RAM MegaWizard Plug-In Manager.

MLABs can implement single-port SRAM through emulation using the Quartus II software. Emulation results in minimal additional logic resources being used. Because of the dual-purpose architecture of the MLAB, it only has data input registers and output registers in the block. MLABs gain input address registers and additional data output registers from ALMs.

For more information about register packing, refer to the *Logic Array Blocks and Adaptive Logic Modules in Stratix IV Devices* chapter.

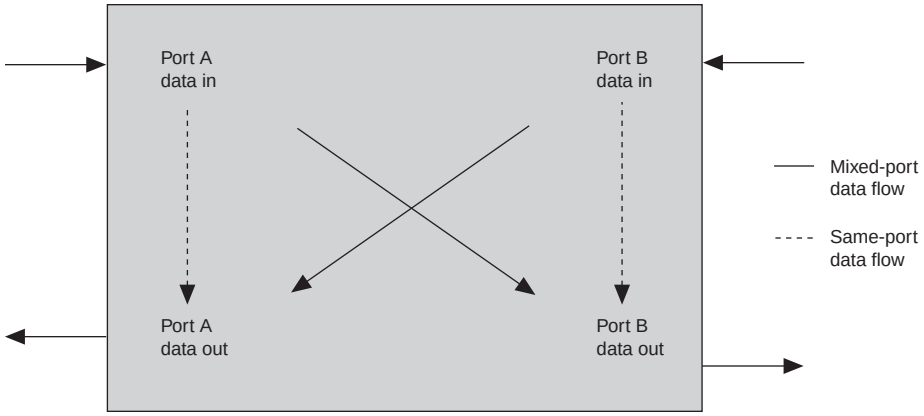
Conflict Resolution

When using memory blocks in true dual-port mode, it is possible to attempt two write operations to the same memory location (address). Because no conflict resolution circuitry is built into the memory blocks, this results in unknown data being written to that location. Therefore, you must implement conflict resolution logic external to the memory block to avoid address conflicts.

Read-During-Write Behavior

You can customize the read-during-write behavior of the Stratix IV TriMatrix memory blocks to suit your design needs. Two types of read-during-write operations are available: same port and mixed port. Figure 3–15 shows the difference between the two types.

Figure 3–15. Stratix IV Read-During-Write Data Flow



Same-Port Read-During-Write Mode

This mode applies to either a single-port RAM or the same port of a true dual-port RAM. In same-port read-during-write mode, three output choices are available: new data mode (or flow-through), old data mode, or don't care mode. In new data mode, the new data is available on the rising edge of the same clock cycle on which it was written. In old data mode, the RAM outputs reflect the old data at that address before the write operation proceeds. In don't care mode, the RAM outputs don't care values for a read-during-write operation.

Figure 3–16 shows sample functional waveforms of same-port read-during-write behavior in new data mode.

Figure 3–16. Same Port Read-During Write: New Data Mode

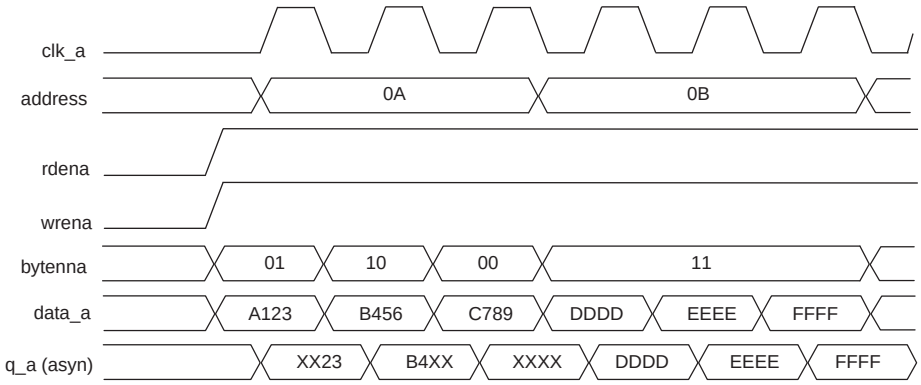
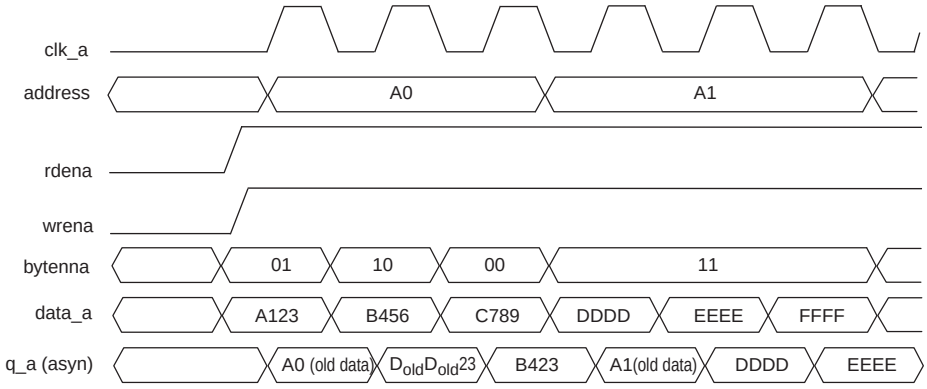


Figure 3–17 shows sample functional waveforms of same-port read-during-write behavior in old data mode.

Figure 3–17. Same Port Read-During-Write: Old Data Mode



Mixed-Port Read-During-Write Mode

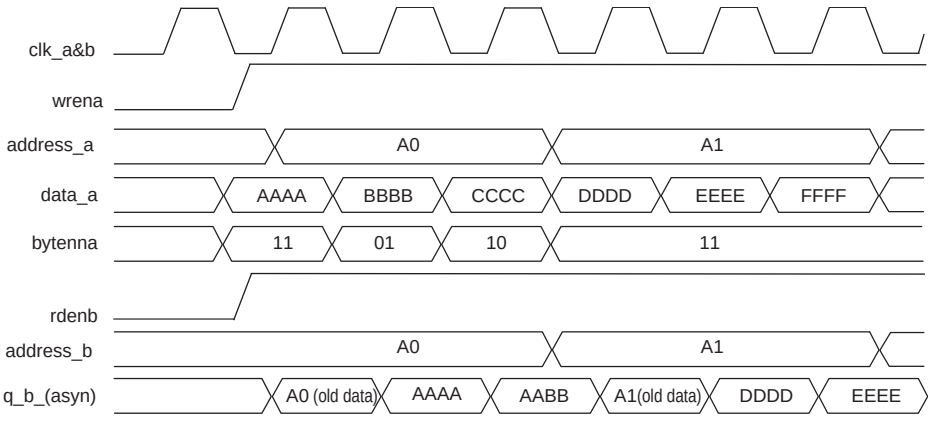
This mode applies to a RAM in simple or true dual-port mode that has one port reading from and the other port writing to the same address location with the same clock.

In this mode, you also have two output choices: "old data" or "don't care". In old data mode, a read-during-write operation to different ports causes the RAM outputs to reflect the "old data" at that address location. In don't care mode, the same operation results in a "don't care" or "unknown" value on the RAM outputs.

Read-during-write behavior is controlled using the RAM MegaWizard Plug-In Manager. For more information, refer to the *RAM Megafunction User Guide*.

Figure 3–18 shows a sample functional waveform of mixed-port read-during-write behavior for old data mode. In don't care mode, "old data" is simply replaced with "don't cares".

Figure 3–18. Mixed Port Read-During-Write: Old Data Mode



Mixed-port read-during-write is not supported when two different clocks are used in a dual-port RAM. The output value is unknown during a dual-clock mixed-port read-during-write operation.

Power-Up Conditions and Memory Initialization

M9K and M144K memory block outputs power up to zero (cleared), regardless of whether the output registers are used or bypassed. MLABs power up to zero if output registers are used and power up reading the memory contents if output registers are not used. You must take this into consideration when designing logic that might evaluate the initial power-up values of the MLAB memory block. For Stratix IV devices, the Quartus II software initializes the RAM cells to zero unless there is a **.mif** file specified.

All memory blocks support initialization using a **.mif** file. You can create **.mif** files in the Quartus II software and specify their use with the RAM MegaWizard Plug-In Manager when instantiating a memory in your design. Even if a memory is pre-initialized (for example, using a **.mif** file), it still powers up with its outputs cleared.

For more information about **.mif** files, refer to the *RAM Megafunction User Guide* and the *Quartus II Handbook*.

Power Management

Stratix IV memory block clock-enables allow you to control clocking of each memory block to reduce AC power consumption. Use the read-enable signal to ensure that read operations only occur when you need them to. If your design does not need read-during-write, you can reduce your power consumption by de-asserting the read-enable signal during write operations, or any period when no memory operations occur.

The Quartus II software automatically places any unused memory blocks in low-power mode to reduce static power.

Document Revision History

Table 3–10 lists the revision history for this chapter.

Table 3–10. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated Table 3–2. ■ Updated the "Simple Dual-Port Mode" section. ■ Minor text edits. ■ Updated graphics.	—
June 2009 v2.3	■ Updated Table 3–1 and Figure 3–2. ■ Updated the "Introduction", "Byte Enable Support", "Mixed Width Support", "Asynchronous Clear", "Single-Port RAM", "Simple Dual-Port Mode", "True Dual-Port Mode", "FIFO Mode", and "Read/Write Clock Mode" sections. ■ Added introductory sentences to improve search ability. ■ Removed the Conclusion section. ■ Minor text edits.	—
April 2009 v2.2	■ Updated Table 3–2.	—
March 2009 v2.1	■ Updated Table 3–2. ■ Removed "Referenced Documents" section.	—
November 2008 v2.0	Updated "Power-Up Conditions and Memory Initialization" on page 3–20	—
May 2008 v1.0	Initial Release.	—

4. DSP Blocks in Stratix IV Devices

SIV51004-3.0

This chapter describes how the Stratix® IV device digital signal processing (DSP) blocks are optimized to support DSP applications requiring high data throughput, such as finite impulse response (FIR) filters, infinite impulse response (IIR) filters, fast Fourier transform (FFT) functions, and encoders. You can configure the DSP blocks to implement one of several operational modes to suit your application. The built-in shift register chain, multipliers, and adders/subtractors minimize the amount of external logic to implement these functions, resulting in efficient resource utilization and improved performance and data throughput for DSP applications.

Many complex systems, such as WiMAX, 3GPP WCDMA, high-performance computing (HPC), voice over Internet protocol (VoIP), H.264 video compression, medical imaging, and HDTV use sophisticated digital signal processing techniques, which typically require a large number of mathematical computations. Stratix IV devices are ideally suited for these tasks because the DSP blocks consist of a combination of dedicated elements that perform multiplication, addition, subtraction, accumulation, summation, and dynamic shift operations.

Along with the high-performance Stratix IV soft logic fabric and TriMatrix memory structures, you can configure DSP blocks to build sophisticated fixed-point and floating-point arithmetic functions. These can be manipulated easily to implement common, larger computationally intensive subsystems such as FIR filters, complex FIR filters, IIR filters, FFT functions, and discrete cosine transform (DCT) functions.

This chapter contains the following sections:

Stratix IV DSP Block Overview

Each Stratix IV device has two to seven columns of DSP blocks that implement multiplication, multiply-add, multiply-accumulate (MAC), and dynamic shift functions efficiently. Architectural highlights of the Stratix IV DSP block include:

- High-performance, power optimized, fully registered, and pipelined multiplication operations
- Natively supported 9-bit, 12-bit, 18-bit, and 36-bit wordlengths
- Natively supported 18-bit complex multiplications
- Efficiently supported floating-point arithmetic formats (24-bit for single precision and 53-bit for double precision)
- Signed and unsigned input support
- Built-in addition, subtraction, and accumulation units to combine multiplication results efficiently
- Cascading 18-bit input bus to form tap-delay line for filtering applications
- Cascading 44-bit output bus to propagate output results from one block to the next block without external logic support
- Rich and flexible arithmetic rounding and saturation units
- Efficient barrel shifter support
- Loopback capability to support adaptive filtering

Table 4–1 shows the number of DSP blocks for the Stratix IV device family.

Table 4–1. Number of DSP Blocks in Stratix IV Devices (Part 1 of 2)

Family	Device	DSP Blocks	Independent Input and Output Multiplication Operators					High-Precision Multiplier Adder Mode	Four Multiplier Adder Mode
			9 × 9 Multipliers	12 × 12 Multipliers	18 × 18 Multipliers	18 × 18 Complex	36 × 36 Multipliers	18 × 36 Multipliers	18 × 18 Multipliers
Stratix IV E	EP4SE230	161	1288	966	644	322	322	644	1288
	EP4SE360	130	1040	780	520	260	260	520	1040
	EP4SE530	128	1024	768	512	256	256	512	1024
	EP4SE820	120	960	720	480	240	240	480	960
Stratix IV GX	EP4SGX70	48	384	288	192	96	96	192	384
	EP4SGX110	64	512	384	256	128	128	256	512
	EP4SGX180	115	920	690	460	230	230	460	920
	EP4SGX230	161	1288	966	644	322	322	644	1288
	EP4SGX290	104	832	624	416	208	208	416	832
	EP4SGX360 *(1)*	130	1040	780	520	260	260	520	1040
	EP4SGX360 *(2)*	128	1024	768	512	256	256	512	1024
	EP4SGX530	128	1024	768	512	256	256	512	1024

Table 4–1. Number of DSP Blocks in Stratix IV Devices (Part 2 of 2)

Family	Device	DSP Blocks	Independent Input and Output Multiplication Operators					High-Precision Multiplier Adder Mode	Four Multiplier Adder Mode
			9 × 9 Multipliers	12 × 12 Multipliers	18 × 18 Multipliers	18 × 18 Complex	36 × 36 Multipliers	18 × 36 Multipliers	18 × 18 Multipliers
Stratix IV GT	EP4S40G2	161	1288	966	644	322	322	644	1288
	EP4S40G5	128	1024	768	512	256	256	512	1024
	EP4S100G2	161	1288	966	644	322	322	644	1288
	EP4S100G3	104	832	624	416	208	208	416	832
	EP4S100G4	128	1024	768	512	256	256	512	1024
	EP4S100G5	128	1024	768	512	256	256	512	1024

Notes to Table 4–1:

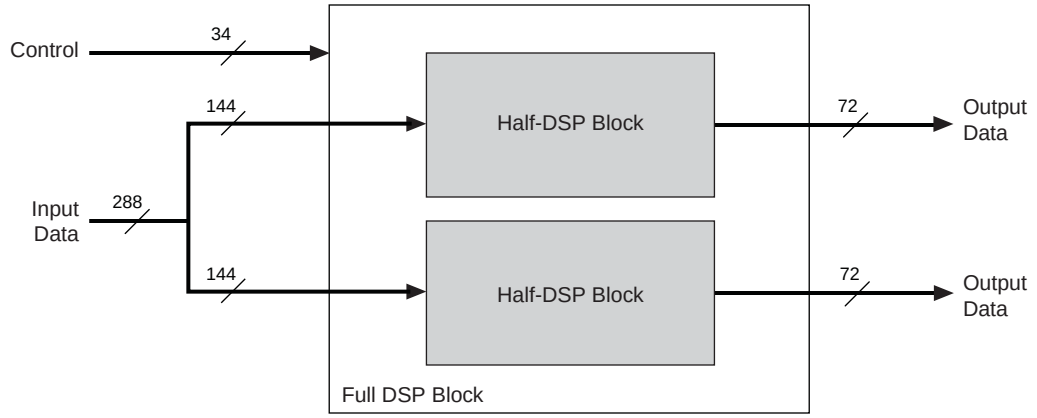
(1) This is applicable for all packages in EP4SGX360 except F1932.

(2) This is applicable for EP4SGX360F1932 only.

Table 4–1 shows that the largest Stratix IV DSP-centric device provides up to 1288 18 × 18 multiplier functionality in the 36 × 36, complex 18 × 18, and summation modes.

Each DSP block occupies four LABs in height and can be divided further into two half blocks that share some common clock signals, but are for all common purposes identical in functionality. Figure 4–1 shows the layout of each DSP block.

Figure 4–1. Overview of DSP Block Signals



Stratix IV Simplified DSP Operation

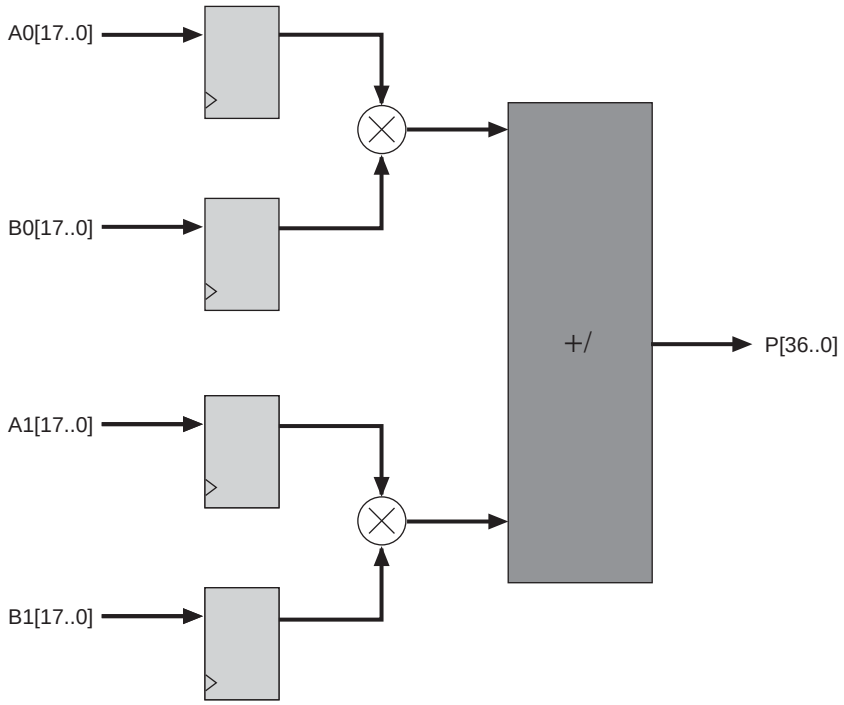
In Stratix IV devices, the fundamental building block is a pair of 18 × 18-bit multipliers followed by a first-stage 37-bit addition/subtraction unit, as shown in Equation 4–1 and Figure 4–2.

All signed numbers, input, and output data are represented in 2's-complement format only.

Equation 4–1. Multiplier Equation

$$P[36..0] = A_0[17..0] \times B_0[17..0] \pm A_1[17..0] \times B_1[17..0]$$

Figure 4–2. Basic Two-Multiplier Adder Building Block



The structure shown in Figure 4–2 is useful for building more complex structures, such as complex multipliers and 36 × 36 multipliers, as described in later sections.

Each Stratix IV DSP block contains four two-multiplier adder units (2 two-multiplier adder units per half block). Therefore, there are eight 18 × 18 multiplier functionalities per DSP block.

Following the two-multiplier adder units are the pipeline registers, the second-stage adders, and an output register stage. You can configure the second-stage adders to provide the alternative functions per half block, as shown in Equation 4–2 and Equation 4–3.

Equation 4–2. Four-Multiplier Adder Equation

$$Z[37..0] = P_0[36..0] + P_1[36..0]$$

Equation 4–3. Four-Multiplier Adder Equation (44-Bit Accumulation)

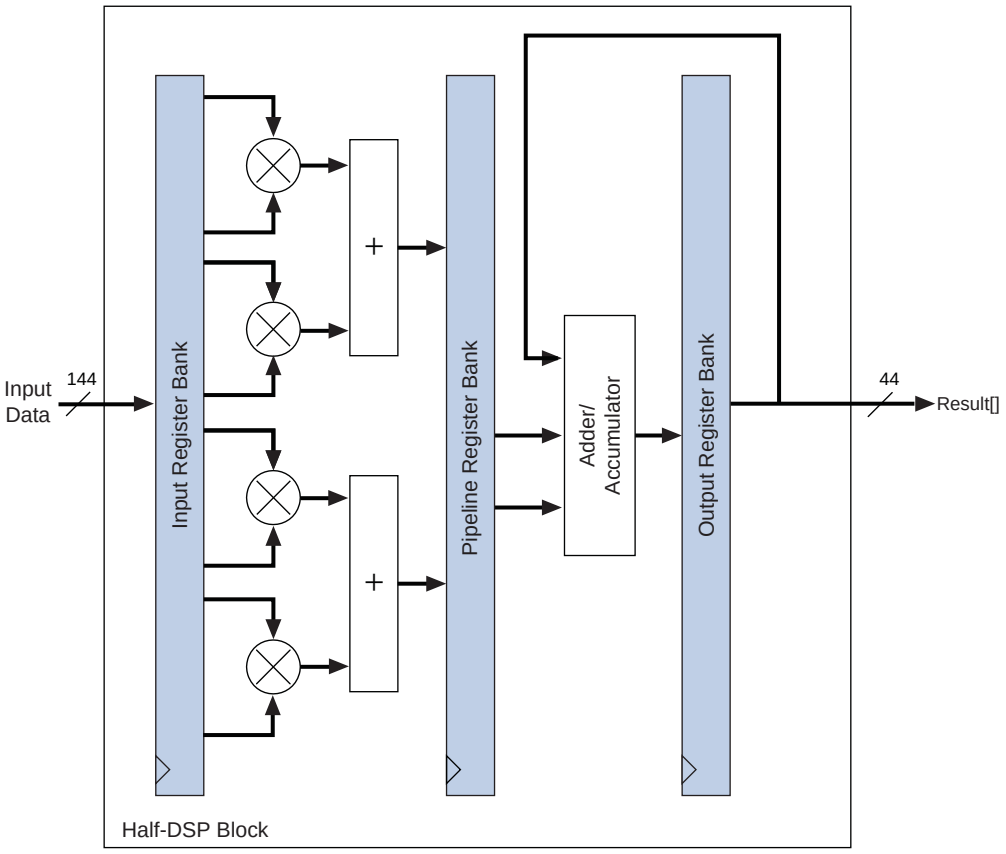
$$W_n[43..0] = W_{n-1}[43..0] \pm Z_n[37..0]$$

In these equations, *n* denotes sample time and *P[36..0]* denotes the result from the two-multiplier adder units.

Equation 4–2 provides a sum of four 18 × 18-bit multiplication operations (four-multiplier adder). Equation 4–3 provides a four 18 × 18-bit multiplication operation but with a maximum 44-bit accumulation capability by feeding the output of the unit back to itself, as shown in Figure 4–3.

Depending on the mode you select, you can bypass all register stages except accumulation and loopback mode. In these two modes, one set of register must be enabled. If the register is not enabled, an infinite loop occurs.

Figure 4–3. Four-Multiplier Adder and Accumulation Capability

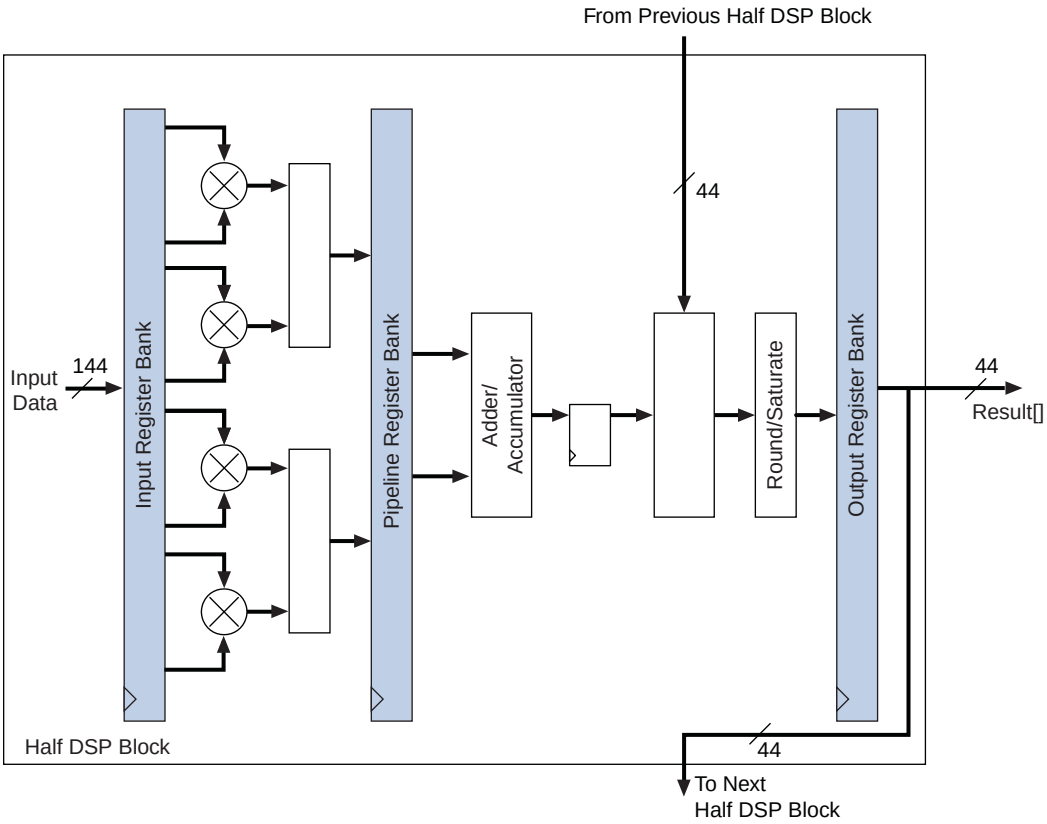


To support commonly found FIR-like structures efficiently, a major addition to the DSP block in Stratix IV devices is the ability to propagate the result of one half block to the next half block completely within the DSP block without additional soft logic overhead. This is achieved by the inclusion of a dedicated addition unit and routing that adds the 44-bit result of a previous half block with the 44-bit result of the current block. The 44-bit result is either fed to the next half block or out of the DSP block using the output register stage, as shown in Figure 4–4. Detailed examples are described in later sections.

The combination of a fast, low-latency four-multiplier adder unit and the "chained cascade" capability of the output chaining adder provides the optimal FIR and vector multiplication capability.

To support single-channel type FIR filters efficiently, you can configure one of the multiplier input's registers to form a tap delay line input, saving resources and providing higher system performance.

Figure 4–4. Output Cascading Feature for FIR Structures

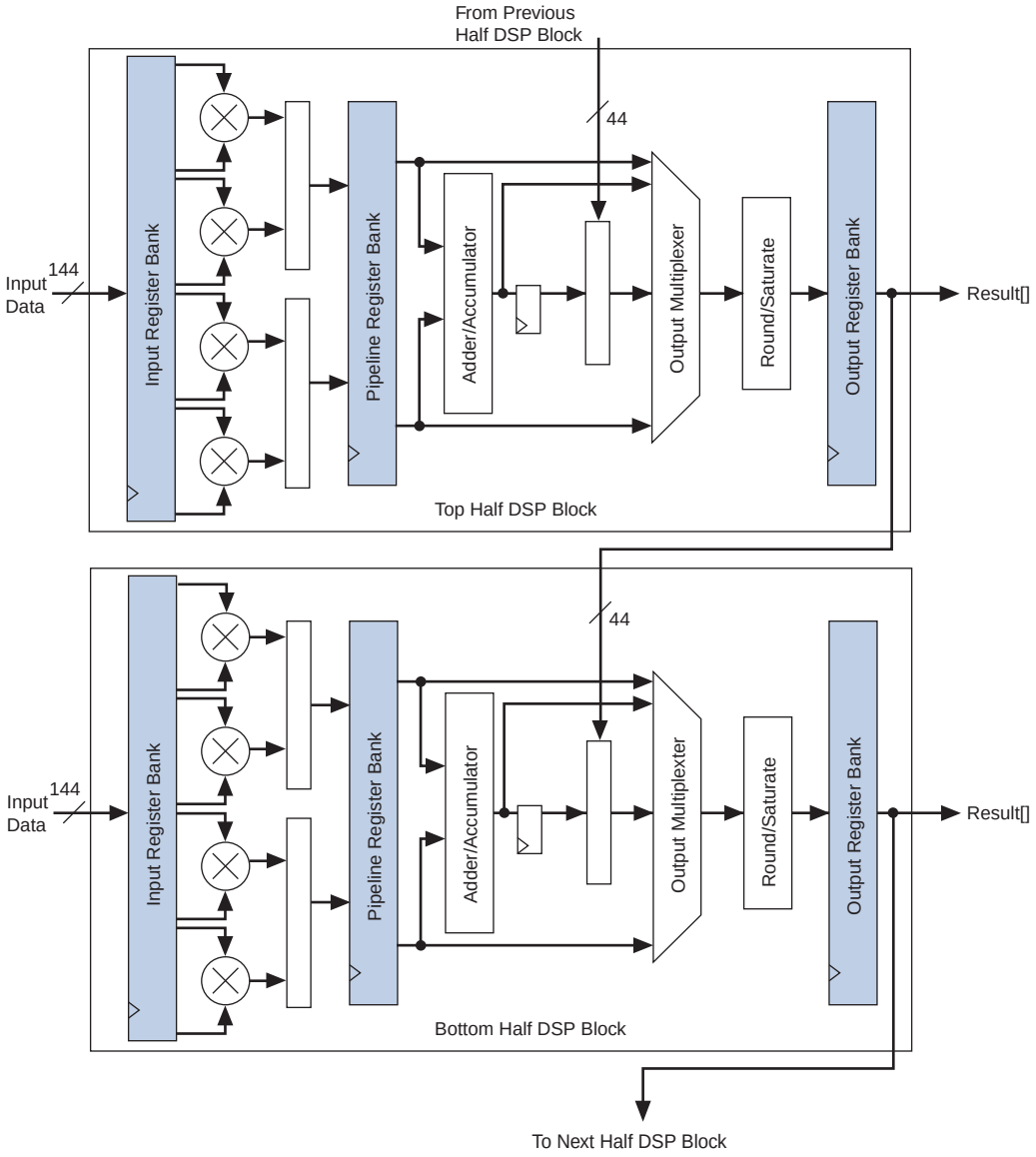


Also shown in Figure 4–4 is the optional rounding and saturation unit (RSU). This unit provides a rich set of commonly found arithmetic rounding and saturation functions used in signal processing.

In addition to the independent multipliers and sum modes, you can use DSP blocks to perform shift operations. DSP blocks can dynamically switch between logical shift left/right, arithmetic shift left/right, and rotation operation in one clock cycle.

Figure 4–5 shows a top-level view of the Stratix IV DSP block.
Figure 4–6 on page 4–9 shows a more detailed top-level view of the DSP block.

Figure 4–5. Stratix IV Full DSP Block



Stratix IV Operational Modes Overview

You can use each Stratix IV DSP block in one of five basic operational modes. Table 4–2 shows the five basic operational modes and the number of multipliers that you can implement within a single DSP block, depending on the mode.

Table 4–2. Stratix IV DSP Block Operation Modes

Mode	Multiplier in Width	# of Mults	# per Block	Signed or Unsigned	RND, SAT	In Shift Register	Chainout Adder	1st Stage Add/Sub	2nd Stage Add/Acc
Independent Multiplier	9 bits	1	8	Both	No	No	No	—	—
	12 bits	1	6	Both	No	No	No	—	—
	18 bits	1	4	Both	Yes	Yes	No	—	—
	36 bits	1	2	Both	No	No	No	—	—
	Double	1	2	Both	No	No	No	—	—
Two-Multiplier Adder *(1)*	18 bits	2	4	Signed *(4)*	Yes	No	No	Both	—
Four-Multiplier Adder	18 bits	4	2	Both	Yes	Yes	Yes	Both	Add Only
Multiply Accumulate	18 bits	4	2	Both	Yes	Yes	Yes	Both	Both
Shift *(2)*	36 bits *(3)*	1	2	Both	No	No	—	—	—
High Precision Multiplier Adder	18×36	2	2	Both	No	No	No	—	Add Only

Notes to Table 4–2:

(1) This mode also supports loopback mode. In loopback mode, the number of loopback multipliers per DSP block is two. You can use the remaining multipliers in regular two-multiplier adder mode.

(2) Dynamic shift mode supports arithmetic shift left, arithmetic shift right, logical shift left, logical shift right, and rotation operation.

(3) Dynamic shift mode operates on a 32-bit input vector but the multiplier width is configured as 36 bits.

(4) Unsigned value is also supported but you must ensure that the result can be contained within 36 bits.

The DSP block consists of two identical halves (the top half and bottom half). Each half has four 18 × 18 multipliers.

The Quartus® II software includes megafunctions used to control the mode of operation of the multipliers. After making the appropriate parameter settings using the megafunction's MegaWizard™ Plug-In Manager, the Quartus II software automatically configures the DSP block.

Stratix IV DSP blocks can operate in different modes simultaneously. Each half block is fully independent except for the sharing of the three `clock`, `ena`, and `aclr` signals. For example, you can break down a single DSP block to operate a 9 × 9 multiplier in one half block and an 18 × 18 two-multiplier adder in the other half block. This increases DSP block resource efficiency and allows you to implement more multipliers within a Stratix IV device. The Quartus II software automatically places multipliers that can share the same DSP block resources within the same block.

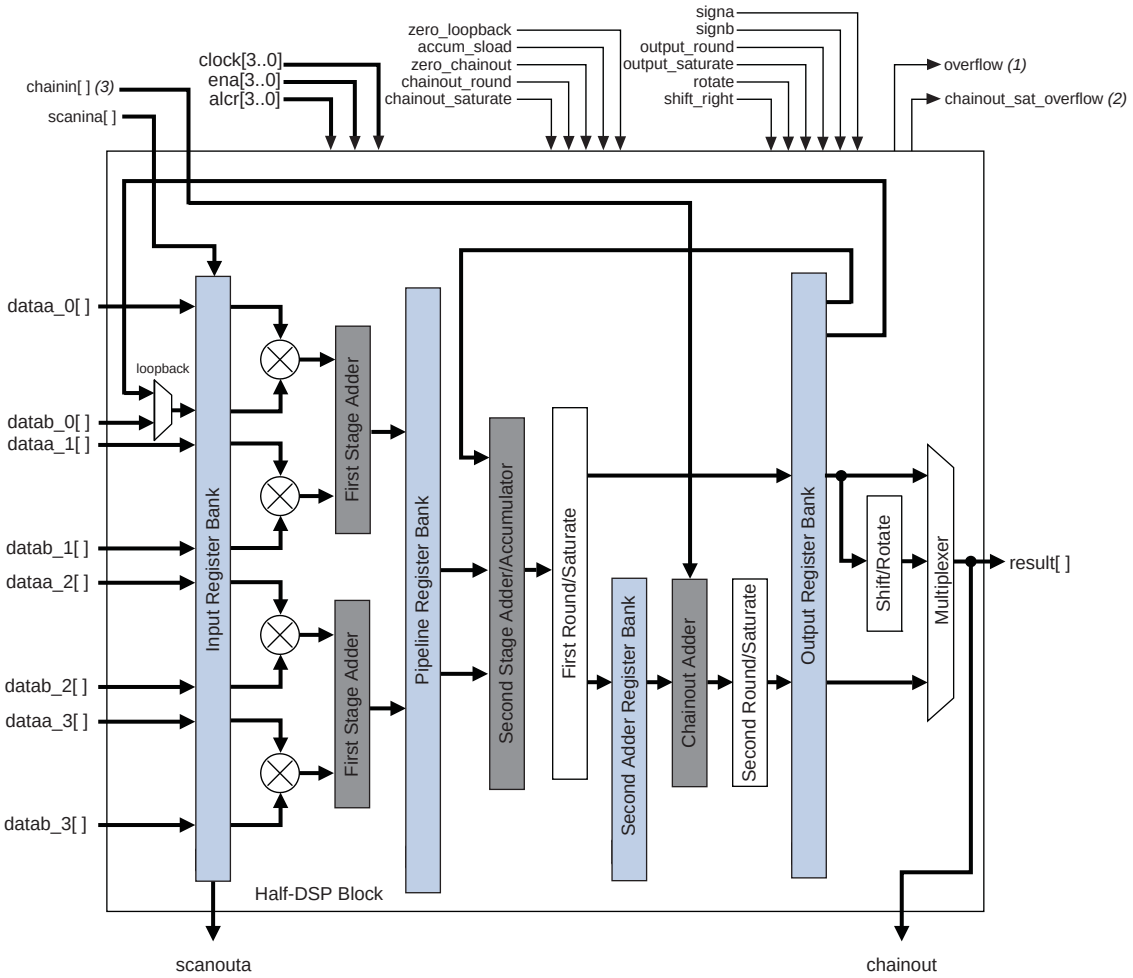
Stratix IV DSP Block Resource Descriptions

The DSP block consists of the following elements:

- Input register bank
- Four two-multiplier adders
- Pipeline register bank
- Two second-stage adders
- Four rounding and saturation logic units
- Second adder register and output register bank

Figure 4–6 shows a detailed overall architecture of the top half of the DSP block. Table 4–9 on page 4–33 shows a list of DSP block dynamic signals.

Figure 4–6. Half DSP Block Architecture



Notes to Figure 4–6:

(1) Block output for accumulator overflow and saturate overflow.
(2) Block output for saturation overflow of chainout.
(3) The chainin port must only be connected to chainout of the previous DSP blocks and must not be connected to general routings.

Input Registers

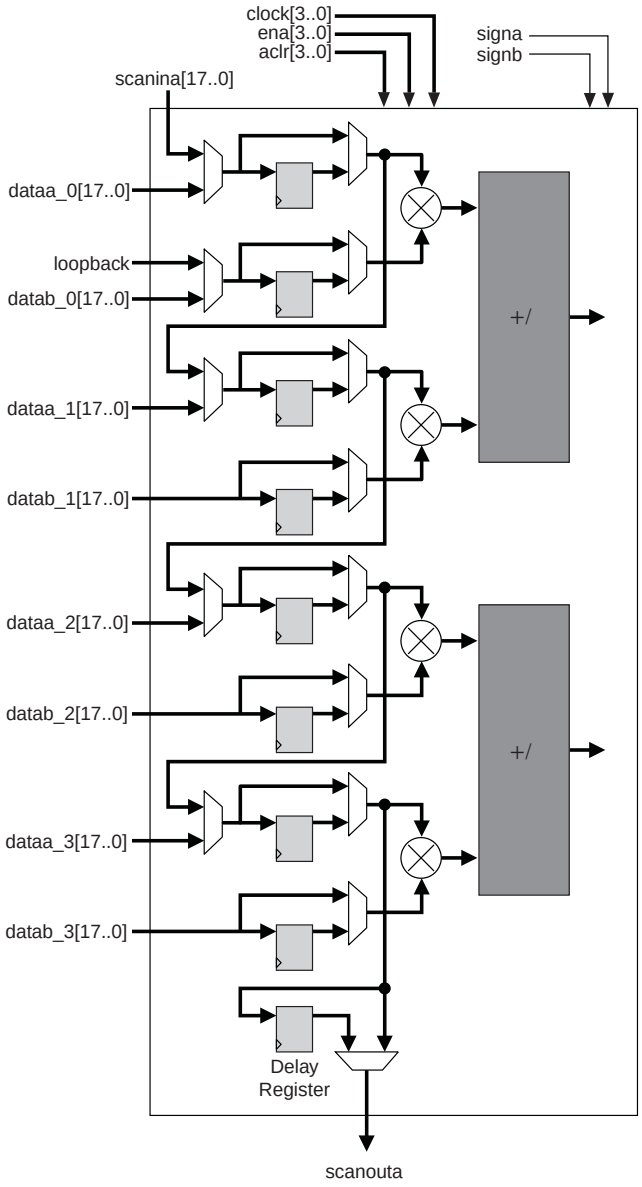
All of the DSP block registers are triggered by the positive edge of the clock signal and are cleared upon power up. Each multiplier operand can feed an input register or go directly to the multiplier, bypassing the input registers. The following DSP block signals control the input registers within the DSP block:

- `clock[3..0]`
- `ena[3..0]`
- `aclr[3..0]`

Every DSP block has nine 18-bit data input register banks per half DSP block. Every half DSP block has the option to use the eight data register banks as inputs to the four multipliers. The special ninth register bank is a delay register required by modes that use both the cascade and chainout features of the DSP block. Use the ninth register bank to balance the latency requirements when using the chained cascade feature.

A feature of the input register bank is to support a tap delay line. Therefore, the top leg of the multiplier input (A) can be driven from general routing or from the cascade chain, as shown in Figure 4–7. Table 4–9 on page 4–33 lists the DSP block dynamic signals.

Figure 4–7. Input Register of a Half DSP Block



At compile time, you must select whether the A-input comes from general routing or from the cascade chain. In cascade mode, the dedicated shift outputs from one multiplier block and directly feeds the input registers of the adjacent multiplier below it (within the same half DSP block) or the first multiplier in the next half DSP block, to form an 8-tap shift register chain per DSP Block. The DSP block can increase the length of the shift register chain by cascading to the lower DSP blocks. The dedicated shift register chain spans a single column, but you can implement longer shift register chains requiring multiple columns using the regular FPGA routing resources.

Shift registers are useful in DSP functions such as FIR filters. When implementing 18 × 18 or smaller width multipliers, you do not need external logic to create the shift register chain because the input shift registers are internal to the DSP block. This implementation significantly reduces the logical element (LE) resources required, avoids routing congestion, and results in predictable timing.

The first multiplier in every half DSP block (top- and bottom-half) in Stratix IV devices has a multiplexer for the first multiplier B-input (lower-leg input) register to select between general routing and loopback, as shown in Figure 4–6 on page 4–9. In loopback mode, the most significant 18-bit registered outputs are connected as feedback to the multiplier input of the first top multiplier in each half DSP block. Loopback modes are used by recursive filters where the previous output is needed to compute the current output.

Loopback mode is described in "Two-Multiplier Adder Sum Mode" on page 4–22.

Table 4–3 lists input register modes for the DSP block.

Table 4–3. Input Register Modes

Register Input Mode *(1)*	9 × 9	12 × 12	18 × 18	36 × 36	Double
Parallel input	✓	✓	✓	✓	✓
Shift register input *(2)*	—	—	✓	—	—
Loopback input *(3)*	—	—	✓	—	—

Notes to Table 4–3:

(1) Multiplier operand input wordlengths are statically configured at compile time.

(2) Available only on the A-operand.

(3) Only one loopback input is allowed per half block. For more information, refer to Figure 4–15 on page 4–23.

Multiplier and First-Stage Adder

The multiplier stage natively supports 9 × 9, 12 × 12, 18 × 18, or 36 × 36 multipliers. Other wordlengths are padded up to the nearest appropriate native wordlength; for example, 16 × 16 would be padded up to use 18 × 18. For more information, refer to "Independent Multiplier Modes" on page 4–15. Depending on the data width of the multiplier, a single DSP block can perform many multiplications in parallel.

Each multiplier operand can be a unique signed or unsigned number. Two dynamic signals, `signa` and `signb`, control the representation of each operand, respectively. A `logic 1` value on the `signa/signb` signal indicates that `data A/data B` is a signed number; a `logic 0` value indicates an unsigned number. Table 4–4 shows the sign of the multiplication result for the various operand sign representations. The result of the multiplication is signed if any one of the operands is a signed value.

Table 4–4. Multiplier Sign Representation

Data A (signa Value)	Data B (signb Value)	Result
Unsigned (logic 0)	Unsigned (logic 0)	Unsigned
Unsigned (logic 0)	Signed (logic 1)	Signed
Signed (logic 1)	Unsigned (logic 0)	Signed
Signed (logic 1)	Signed (logic 1)	Signed

Each half block has its own `signa` and `signb` signal. Therefore, all of the `data A` inputs feeding the same half DSP block must have the same sign representation. Similarly, all of the `data B` inputs feeding the same half DSP block must have the same sign representation. The multiplier offers full precision regardless of the sign representation in all operational modes except for full precision 18 × 18 loopback and two-multiplier adder modes. For more information, refer to "Two-Multiplier Adder Sum Mode" on page 4–22.

By default, when the `signa` and `signb` signals are unused, the Quartus II software sets the multiplier to perform unsigned multiplication.

Figure 4–6 on page 4–9 shows that the outputs of the multipliers are the only outputs that can feed into the first-stage adder. There are four first-stage adders in a DSP block (two adders per half DSP block). The first-stage adder block has the ability to perform addition and subtraction. The control signal for addition or subtraction is static and has to be configured upon compile time. The first-stage adders are used by the sum modes to compute the sum of two multipliers, 18 × 18-complex multipliers, and to perform the first stage of a 36 × 36 multiply and shift operations.

Depending on your specifications, the output of the first-stage adder has the option to feed into the pipeline registers, second-stage adder, rounding and saturation unit, or output registers.

Pipeline Register Stage

Figure 4–6 on page 4–9 shows that the output from the first-stage adder can either feed or bypass the pipeline registers. Pipeline registers increase the DSP block's maximum performance (at the expense of extra cycles of latency), especially when using the subsequent DSP block stages. Pipeline registers split up the long signal path between the input registers/multiplier/first-stage adder and the second-stage adder/round-and-saturation/output registers, creating two shorter paths.

Second-Stage Adder

There are four individual 44-bit second-stage adders per DSP block (two adders per half DSP block). You can configure the second-stage adders as follows:

- The final stage of a 36-bit multiplier
- A sum of four (18 × 18)
- An accumulator (44-bits maximum)
- A chained output summation (44-bits maximum)

You can use the chained-output adder at the same time as a second-level adder in chained output summation mode.

The output of the second-stage adder has the option to go into the rounding and saturation logic unit or the output register.

You cannot use the second-stage adder independently from the multiplier and first-stage adder.

Rounding and Saturation Stage

The rounding and saturation logic units are located at the output of the 44-bit second-stage adder (the rounding logic unit followed by the saturation logic unit). There are two rounding and saturation logic units per half DSP block. The input to the rounding and saturation logic unit can come from one of the following stages:

- Output of the multiplier (independent multiply mode in 18 × 18)
- Output of the first-stage adder (two-multiplier adder)
- Output of the pipeline registers
- Output of the second-stage adder (four-multiplier adder and multiply-accumulate mode in 18 × 18)

These stages are described in "Stratix IV Operational Mode Descriptions" on page 4–15.

The rounding and saturation logic unit is controlled by the dynamic rounding and saturate signals, respectively. A `logic 1` value on the rounding and/or saturate signals enables the rounding and/or saturate logic unit, respectively.

You can use the rounding and saturation logic units together or independently.

Second Adder and Output Registers

The second adder register and output register banks are two banks of 44-bit registers that you can combine to form larger 72-bit banks to support 36 × 36 output results.

The outputs of the different stages in the Stratix IV devices are routed to the output registers through an output selection unit. Depending on the operational mode of the DSP block, the output selection unit selects whether the outputs of the DSP blocks comes from the outputs of the multiplier block, first-stage adder, pipeline registers, second-stage adder, or the rounding and saturation logic unit. The output selection unit is set automatically by the software, based on the DSP block operational mode you specified, and has the option to either drive or bypass the output registers. The exception is when you use the block in shift mode, in which case you dynamically control the output-select multiplexer directly.

When the DSP block is configured in chained cascaded output mode, both of the second-stage adders are used. Use the first one for performing a four-multiplier adder; use the second for the chainout adder.

The outputs of the four-multiplier adder are routed to the second-stage adder registers before they enter the chainout adder. The output of the chainout adder goes to the regular output register bank. Depending on the configuration, you can route the chainout results to the input of the next half block's chainout adder input or to the general fabric (functioning as regular output registers). For more information, refer to "Stratix IV Operational Mode Descriptions" on page 4–15.

The second-stage and output registers are triggered by the positive edge of the clock signal and are cleared on power up. The following DSP block signals control the output registers within the DSP block:

- `clock[3..0]`
- `ena[3..0]`
- `aclr[3..0]`

Stratix IV Operational Mode Descriptions

This section contains an explanation of different operational modes in Stratix IV devices.

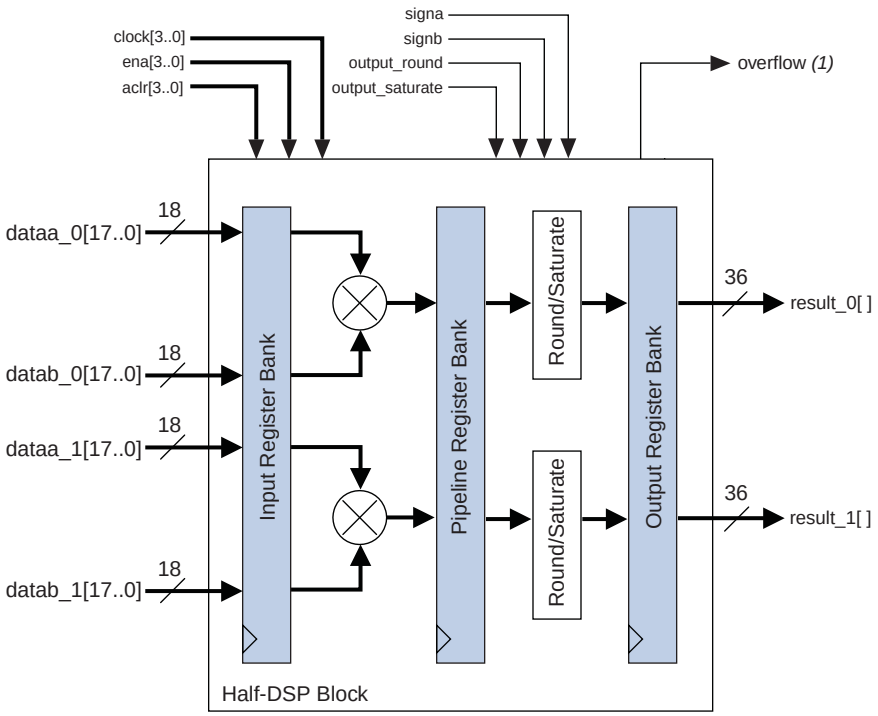
Independent Multiplier Modes

In independent input and output multiplier mode, the DSP block performs individual multiplication operations for general-purpose multipliers.

9-, 12-, and 18-Bit Multiplier

You can configure each DSP block multiplier for 9-, 12-, or 18-bit multiplication. A single DSP block can support up to eight individual 9 × 9 multipliers, six individual 12 × 12 multipliers, or four individual 18 × 18 multipliers. For operand widths up to 9 bits, a 9 × 9 multiplier is implemented. For operand widths from 10 to 12 bits, a 12 × 12 multiplier is implemented, and for operand widths from 13 to 18 bits, an 18 × 18 multiplier is implemented. This is done by the Quartus II software by zero-padding the LSBs. Figure 4–8, Figure 4–9, and Figure 4–10 show the DSP block in the independent multiplier operation. Table 4–9 on page 4–33 lists the dynamic signals for the DSP block.

Figure 4–8. 18-Bit Independent Multiplier Mode Shown for a Half DSP Block



Note to Figure 4–8:

(1) Block output for accumulator overflow and saturate overflow.

Figure 4–9. 12-Bit Independent Multiplier Mode Shown for a Half DSP Block

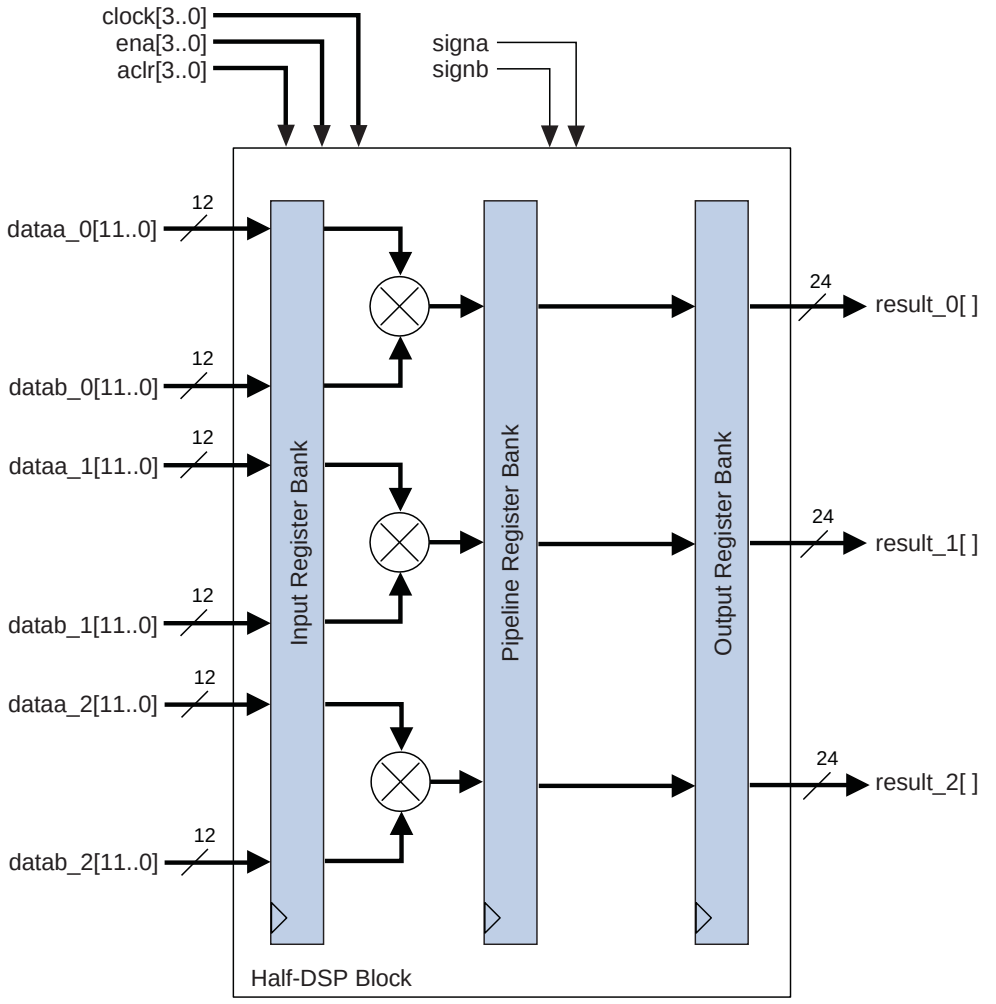
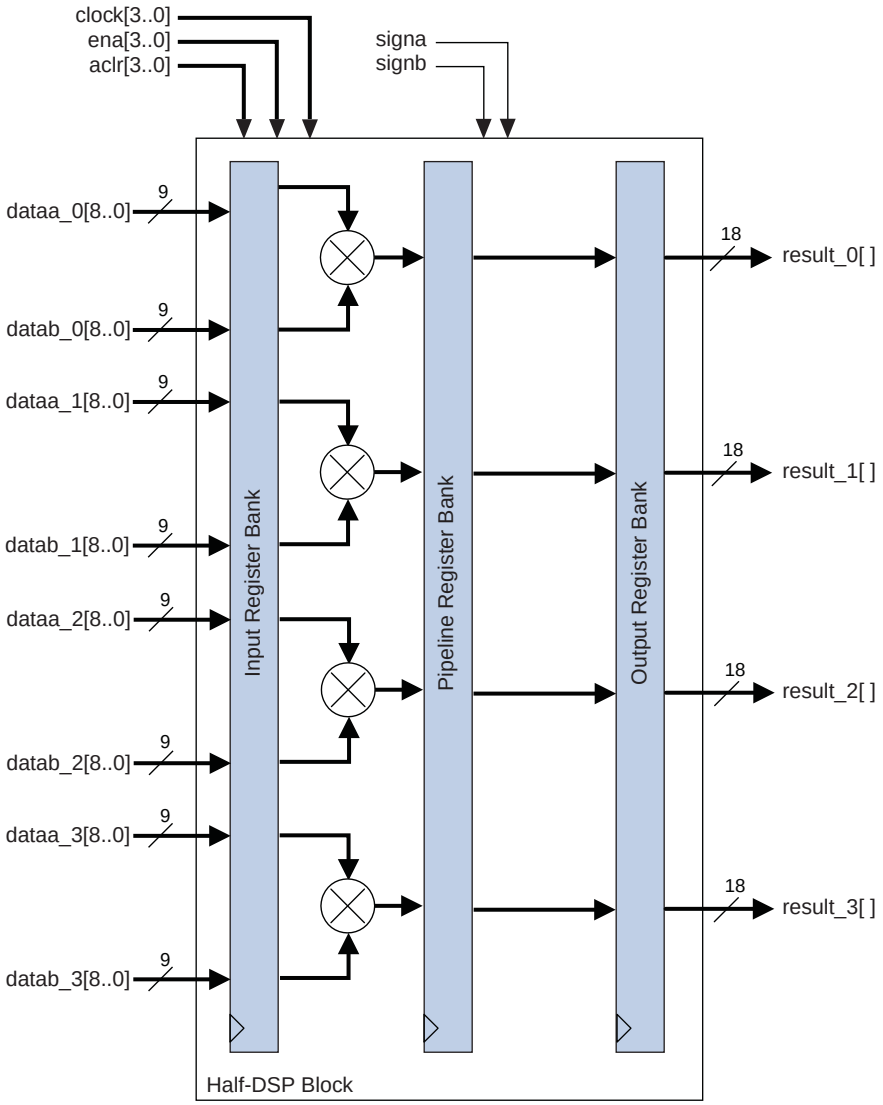


Figure 4–10. 9-Bit Independent Multiplier Mode Shown for a Half Block



The multiplier operands can accept signed integers, unsigned integers, or a combination of both. You can change the `signa` and `signb` signals dynamically and can register the signals in the DSP block. Additionally, the multiplier inputs and results can be registered independently. You can use the pipeline registers within the DSP block to pipeline the multiplier result, increasing the performance of the DSP block.

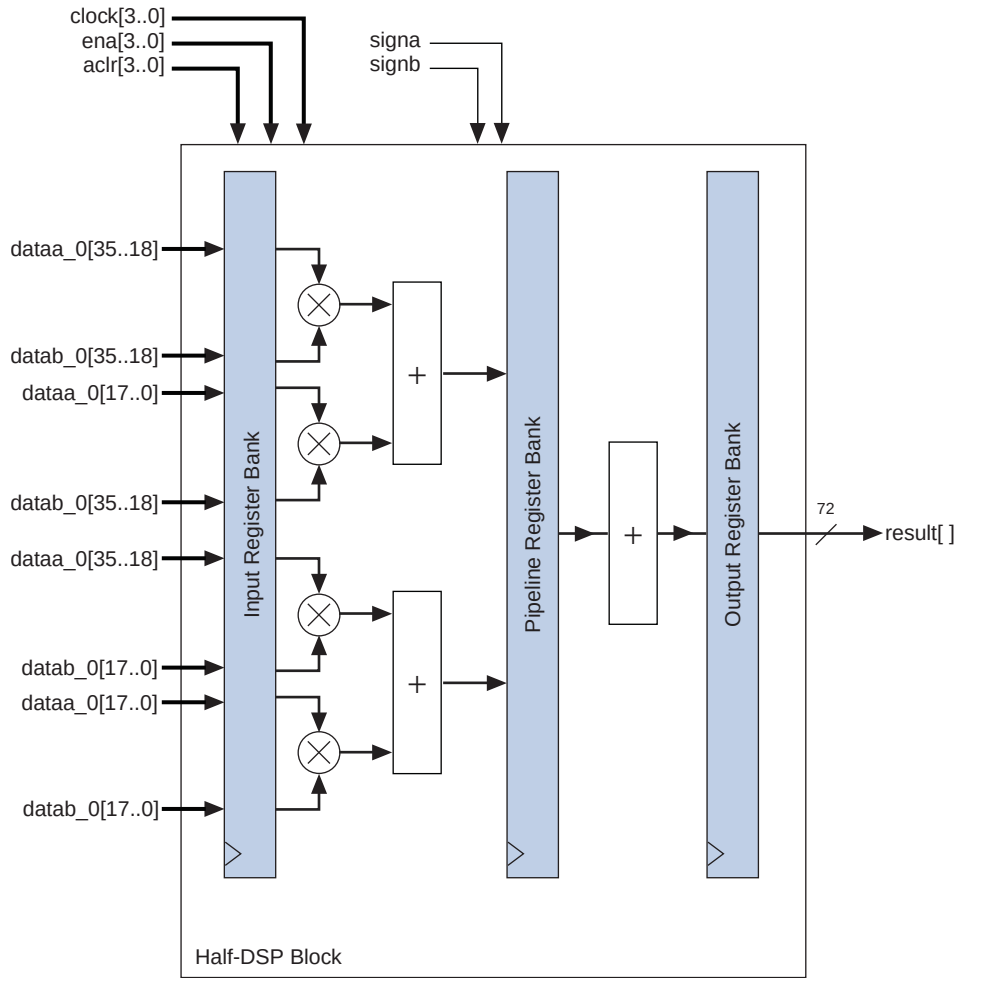
The rounding and saturation logic unit is supported for 18-bit independent multiplier mode only.

36-Bit Multiplier

You can efficiently construct a 36 × 36 multiplier using four 18 × 18 multipliers. This simplification fits conveniently into one half DSP block and is implemented in the DSP block automatically by selecting 36 × 36 mode. Stratix IV devices can have up to two 36-bit multipliers per DSP block (one 36-bit multiplier per half DSP block). The 36-bit multiplier is also under the independent multiplier mode but uses the entire half DSP block, including the dedicated hardware logic after the pipeline registers to implement the 36 × 36 bit multiplication operation, as shown in Figure 4–11.

The 36-bit multiplier is useful for applications requiring more than 18-bit precision; for example, for the mantissa multiplication portion of single precision and extended single precision floating-point arithmetic applications.

Figure 4–11. 36-Bit Independent Multiplier Mode Shown for a Half DSP Block



Double Multiplier

You can configure the Stratix IV DSP block to efficiently support a signed or unsigned 54 × 54-bit multiplier that is required to compute the mantissa portion of an IEEE double-precision floating point multiplication. You can build a 54 × 54-bit multiplier using basic 18 × 18 multipliers, shifters, and adders. In order to efficiently utilize the Stratix IV DSP block's built-in shifters and adders, a special double mode (partial 54 × 54 multiplier) is available that is a slight modification to the basic 36 × 36 multiplier mode, as shown in Figure 4–12 and Figure 4–13.

Figure 4–12. Double Mode Shown for a Half DSP Block

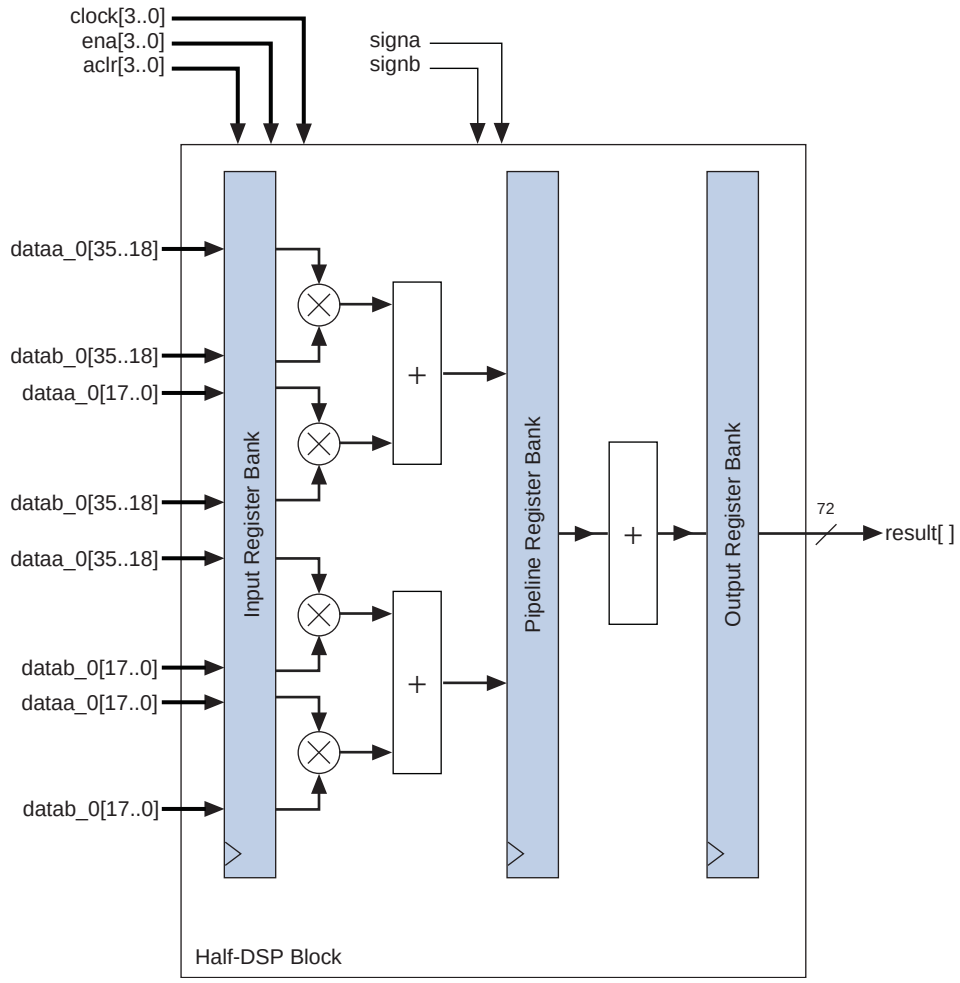
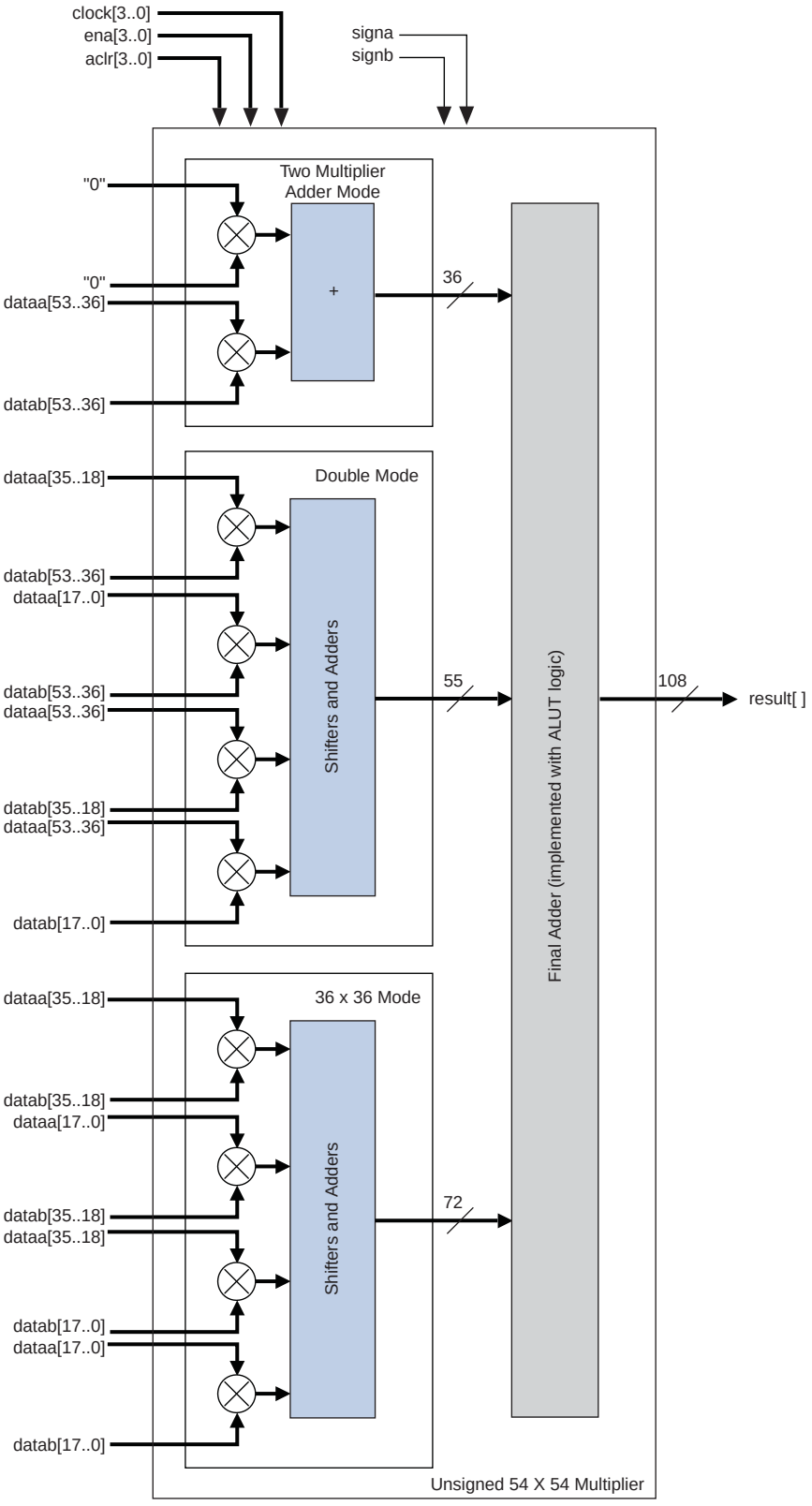


Figure 4–13. Unsigned 54 × 54 Multiplier for a Half-DSP Block



Two-Multiplier Adder Sum Mode

In the two-multiplier adder configuration, the DSP block can implement four 18-bit two-multiplier adders (2 two-multiplier adders per half DSP block). You can configure the adders to take the sum or difference of two multiplier outputs. You must select summation or subtraction at compile time. The two-multiplier adder function is useful for applications such as FFTs, complex FIR, and IIR filters. Figure 4–14 shows the DSP block configured in two-multiplier adder mode.

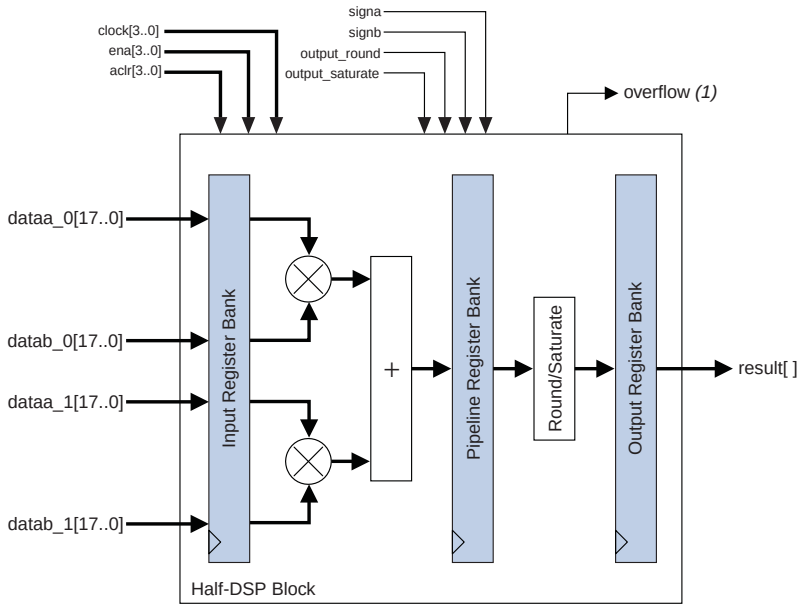
Loopback mode is the other sub-feature of the two-multiplier adder mode. Figure 4–15 shows the DSP block configured in the loopback mode. This mode takes the 36-bit summation result of the two multipliers and feeds back the most significant 18-bits to the input. The lower 18-bits are discarded. You have the option to disable or zero-out the loopback data by using the dynamic `zero_loopback` signal. A `logic 1` value on the `zero_loopback` signal selects the `zeroed` data or disables the looped back data, while a `logic 0` selects the looped back data.

You must select the option to use loopback mode or the general two-multiplier adder mode at compile time.

For two-multiplier adder mode, if all the inputs are full 18-bit and unsigned, the result requires 37 bits. As the output data width in two-multiplier adder mode is limited to 36 bits, this 37-bit output requirement is not allowed. Any other combination that does not violate the 36-bit maximum result is permitted; for example, two 16 × 16 signed two-multiplier adders is valid.

Two-multiplier adder mode supports the rounding and saturation logic unit. You can use the pipeline registers and output registers within the DSP block to pipeline the multiplier-adder result, increasing the performance of the DSP block.

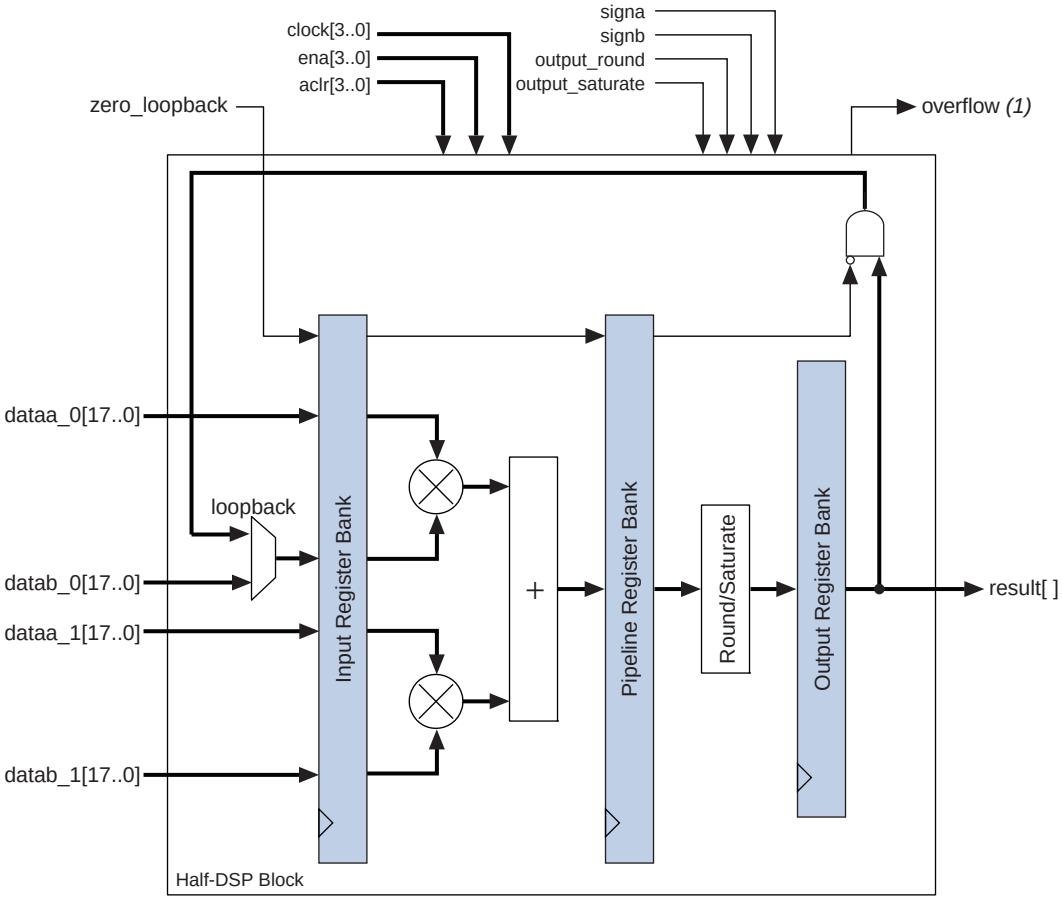
Figure 4–14. Two-Multiplier Adder Mode Shown for a Half DSP Block



Note to Figure 4–14:

(1) Block output for accumulator overflow and saturate overflow.

Figure 4–15. Loopback Mode for a Half DSP Block



Note to Figure 4–15:

(1) Block output for accumulator overflow and saturate overflow.

18 x 18 Complex Multiply

You can configure the DSP block to implement complex multipliers using two-multiplier adder mode. A single half DSP block can implement one 18-bit complex multiplier.

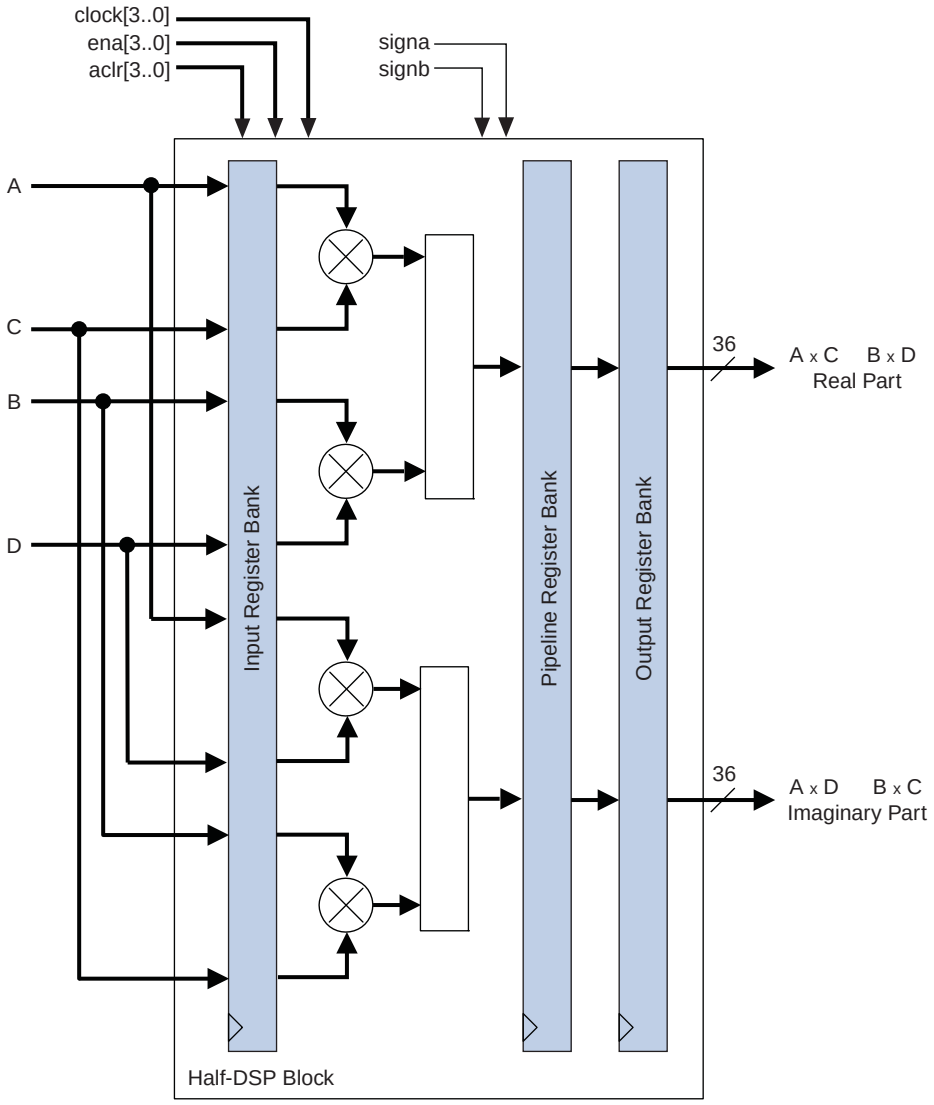
Equation 4–4 shows a complex multiplication.

Equation 4–4. Complex Multiplication Equation

$$(a + jb) \times (c + jd) = ((a \times c) - (b \times d)) + j((a \times d) + (b \times c))$$

To implement this complex multiplication within the DSP block, the real part ((a × c) – (b × d)) is implemented using two multipliers feeding one subtractor block while the imaginary part ((a × d) + (b × c)) is implemented using another two multipliers feeding an adder block. Figure 4–16 shows an 18-bit complex multiplication. This mode automatically assumes all inputs are using signed numbers.

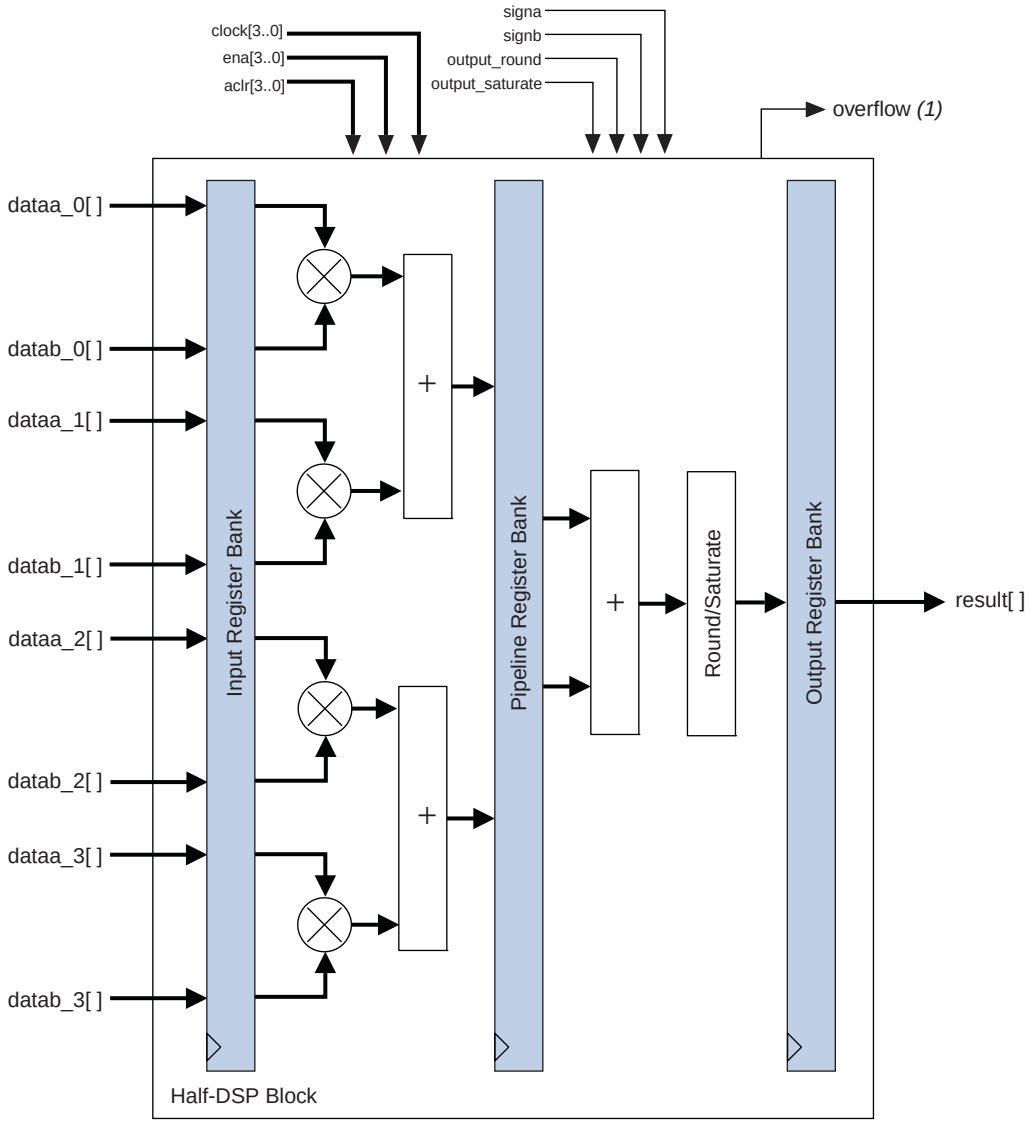
Figure 4–16. Complex Multiplier Using Two-Multiplier Adder Mode



Four-Multiplier Adder

In the four-multiplier adder configuration shown in Figure 4–17, the DSP block can implement two four-multiplier adders (one four-multiplier adder per half DSP block). These modes are useful for implementing one-dimensional and two-dimensional filtering applications. The four-multiplier adder is performed in two addition stages. The outputs of two of the four multipliers are initially summed in the two first-stage adder blocks. The results of these two adder blocks are then summed in the second-stage adder block to produce the final four-multiplier adder result, as shown by Equation 4–2 on page 4–4 and Equation 4–3 on page 4–4.

Figure 4–17. Four-Multiplier Adder Mode Shown for a Half DSP Block



Note to Figure 4–17:

(1) Block output for accumulator overflow and saturate overflow.

Four-multiplier adder mode supports the rounding and saturation logic unit. You can use the pipeline registers and output registers within the DSP block to pipeline the multiplier-adder result, increasing the performance of the DSP block.

High-Precision Multiplier Adder Mode

In the high-precision multiplier adder configuration, shown in Figure 4–18, the DSP block can implement 2 two-multiplier adders, with multiplier precision of 18 x 36 (one two-multiplier adder per half DSP block). This mode is useful in filtering or FFT applications where a data path greater than 18 bits is required, yet 18 bits is sufficient for the coefficient precision. This can occur where the data has a high dynamic range. If the coefficients are fixed, as in FFT and most filter applications, the precision of 18 bits provide a dynamic range over 100 dB, if the largest coefficient is normalized to the maximum 18-bit representation.

In these situations, the data path can be up to 36 bits, allowing sufficient capacity for bit growth or gain changes in the signal source without loss of precision. This mode is also extremely useful in single precision block floating point applications.

The high-precision multiplier adder is performed in two stages. The 18 × 36 multiply is divided into two 18 × 18 multipliers. The multiplier with the LSB of the data source is performed unsigned, while the multiplier with the MSB of the data source can be signed or unsigned. The latter multiplier has its result left shifted by 18 bits prior to the first adder stage, creating an effective 18 x 36 multiplier. The results of these two adder blocks are then summed in the second stage adder block to produce the final result:

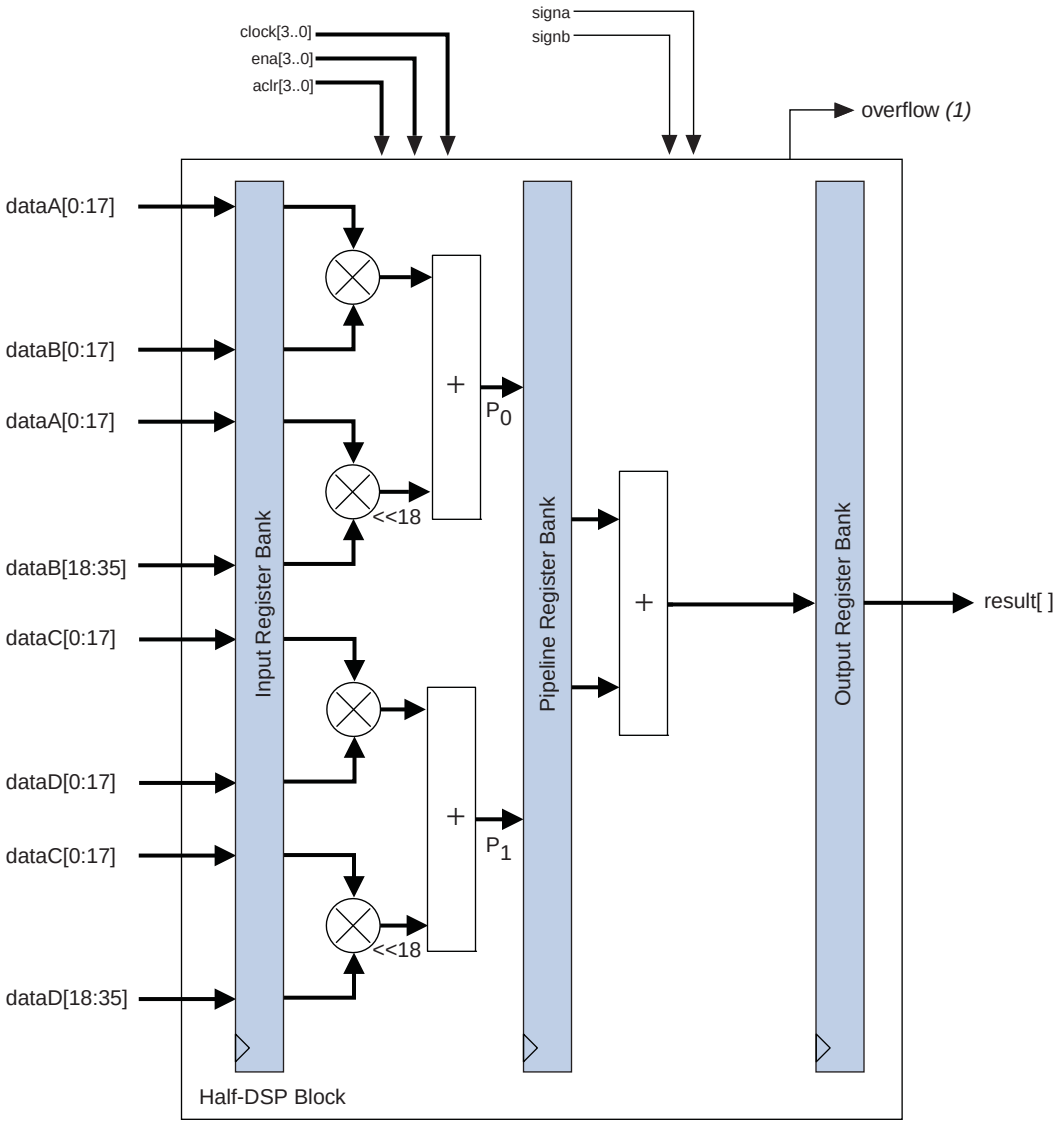
$Z[54..0] = P_0[53..0] + P_1[53..0]$

where:

$P_0 = A[17..0] \times B[35..0]$

$P_1 = C[17..0] \times D[35..0]$

Figure 4–18. High-Precision Multiplier Adder Configuration



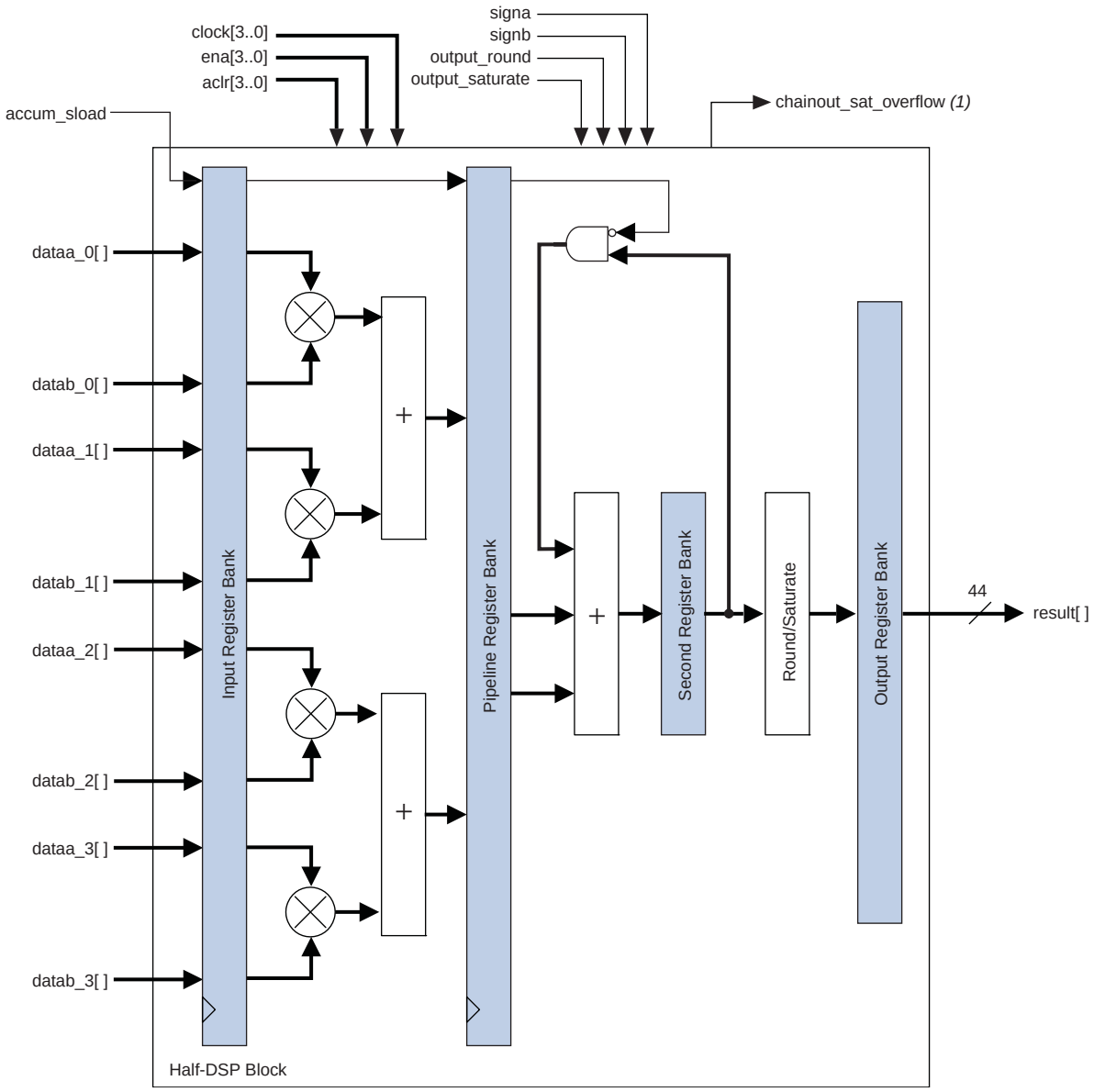
Note to Figure 4–18:

(1) Block output for accumulator overflow and saturate overflow.

Multiply Accumulate Mode

In multiply accumulate mode, the second-stage adder is configured as a 44-bit accumulator or subtractor. The output of the DSP block is looped back to the second-stage adder and added or subtracted with the two outputs of the first-stage adder block according to Equation 4–3 on page 4–4. Figure 4–19 shows the DSP block configured to operate in multiply accumulate mode.

Figure 4–19. Multiply Accumulate Mode Shown for a Half DSP Block



Note to Figure 4–19:

(1) Block output for saturation overflow of chainout.

A single DSP block can implement up to two independent 44-bit accumulators.

Use the dynamic `accum_sload` control signal to clear the accumulation. A `logic 1` value on the `accum_sload` signal synchronously loads the accumulator with the multiplier result only, while a `logic 0` enables accumulation by adding or subtracting the output of the DSP block (accumulator feedback) to the output of the multiplier and first-stage adder.

You must configure the control signal for the accumulator and subtractor is static at compile time.

This mode supports the rounding and saturation logic unit because it is configured as an 18-bit multiplier accumulator. You can use the pipeline registers and output registers within the DSP block to increase the performance of the DSP block.

Shift Modes

Stratix IV devices support the following shift modes for 32-bit input only:

- Arithmetic shift left, `ASL[N]`
- Arithmetic shift right, `ASR[32-N]`
- Logical shift left, `LSL[N]`
- Logical shift right, `LSR[32-N]`
- 32-bit rotator or barrel shifter, `ROT[N]`

You can switch shift mode between these modes using the dynamic rotate and shift control signals.

You can use shift mode in a Stratix IV device by using a soft embedded processor such as Nios® II to perform the dynamic shift and rotate operation. Figure 4–20 shows the shift mode configuration.

Shift mode makes use of the available multipliers to logically or arithmetically shift left, right, or rotate the desired 32-bit data. You can configure the DSP block similar to the independent 36-bit multiplier mode to perform the shift mode operations.

Arithmetic shift right requires a signed input vector. During an arithmetic shift right, the sign is extended to fill the MSB of the 32-bit vector. The logical shift right uses an unsigned input vector. During a logical shift right, zeros are padded in the MSBs, shifting the 32-bit vector to the right. The barrel shifter uses unsigned input vector and implements a rotation function on a 32-bit word length.

Two control signals, `rotate` and `shift_right`, together with the `signa` and `signb` signals, determine the shifting operation. Table 4–5 shows examples of shift operations.

Figure 4–20. Shift Operation Mode Shown for a Half DSP Block

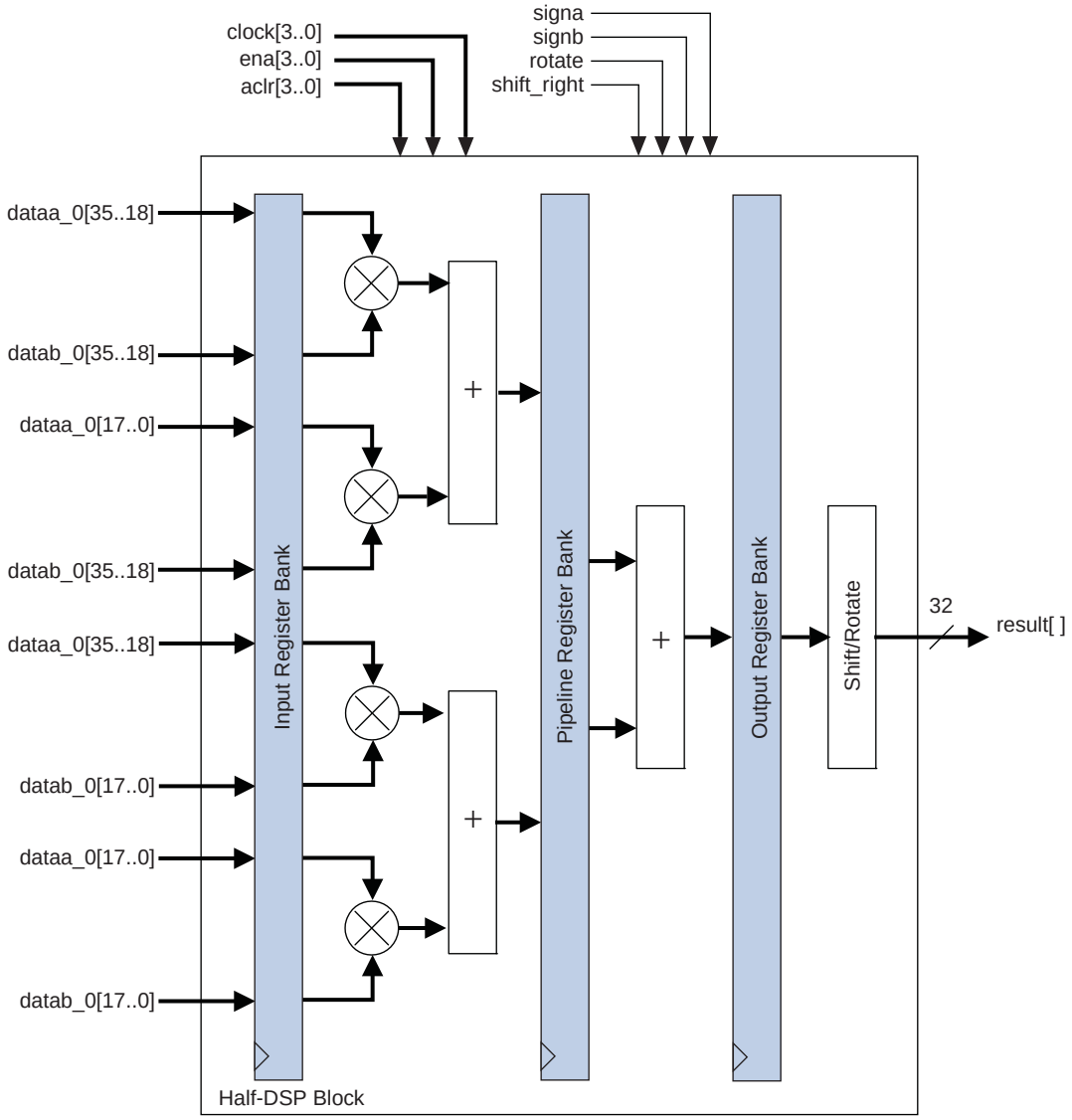


Table 4–5. Examples of Shift Operations

Example	Signa	Signb	Shift	Rotate	A-input	B-input	Result
Logical Shift Left `LSL[N]`	Unsigned	Unsigned	0	0	0xAABBCCDD	0x0000100	0xBBCCDD00
Logical Shift Right `LSR[32-N]`	Unsigned	Unsigned	1	0	0xAABBCCDD	0x0000100	0x000000AA
Arithmetic Shift Left `ASL[N]`	Signed	Unsigned	0	0	0xAABBCCDD	0x0000100	0xBBCCDD00
Arithmetic Shift Right `ASR[32-N]`	Signed	Unsigned	1	0	0xAABBCCDD	0x0000100	0xFFFFFFAA
Rotation `ROT[N]`	Unsigned	Unsigned	0	1	0xAABBCCDD	0x0000100	0xBBCCDDAA

Rounding and Saturation Mode

Rounding and saturation functions are often required in DSP arithmetic. Use rounding to limit bit growth and its side effects; use saturation to reduce overflow and underflow side effects.

Two rounding modes are supported in Stratix IV devices:

- Round-to-nearest-integer mode
- Round-to-nearest-even mode

You must select one of these two options at compile time.

Round-to-nearest-integer provides the biased rounding support and is the simplest form of rounding commonly used in DSP arithmetic. The round-to-nearest-even method provides unbiased rounding support and is used where DC offsets are a concern. Table 4–6 shows how round-to-nearest-even works.

Table 4–7 shows examples of the difference between the two modes. In this example, a 6-bit input is rounded to 4 bits. Table 4–7 shows the main difference between the two rounding options is when the residue bits are exactly halfway between its nearest two integers and the LSB is zero (even).

Table 4–6. Example of Round-To-Nearest-Even Mode

6- to 4-bits Rounding	Odd/Even (Integer)	Fractional	Add to Integer	Result
010111	x	> 0.5 (11)	1	0110
001101	x	< 0.5 (01)	0	0011
001010	Even (0010)	= 0.5 (10)	0	0010
001110	Odd (0011)	= 0.5 (10)	1	0100
110111	x	> 0.5 (11)	1	1110
101101	x	< 0.5 (01)	0	1011
110110	Odd (1101)	= 0.5 (10)	1	1110
110010	Even (1100)	= 0.5 (10)	0	1100

Table 4–7. Comparison of Round-to-Nearest-Integer and Round-to-Nearest-Even

Round-To-Nearest-Integer	Round-To-Nearest-Even
010111 ⇨ 0110	010111 ⇨ 0110
001101 ⇨ 0011	001101 ⇨ 0011
001010 ⇨ 0011	001010 ⇨ 0010
001110 ⇨ 0100	001110 ⇨ 0100
110111 ⇨ 1110	110111 ⇨ 1110
101101 ⇨ 1011	101101 ⇨ 1011
110110 ⇨ 1110	110110 ⇨ 1110
110010 ⇨ 1101	110010 ⇨ 1100

Two saturation modes are supported in Stratix IV:

- Asymmetric saturation mode
- Symmetric saturation mode

You must select one of the two options at compile time.

In 2's-complement format, the maximum negative number that can be represented is $-2^{(n-1)}$, while the maximum positive number is $2^{(n-1)} - 1$. Symmetrical saturation limits the maximum negative number to $-2^{(n-1)} + 1$. For example, for 32 bits:

- Asymmetric 32-bit saturation: Max = 0x7FFFFFFF, Min = 0x80000000
- Symmetric 32-bit saturation: Max = 0x7FFFFFFF, Min = 0x80000001

Table 4–8 shows how saturation works. In this example, a 44-bit input is saturated to 36-bits.

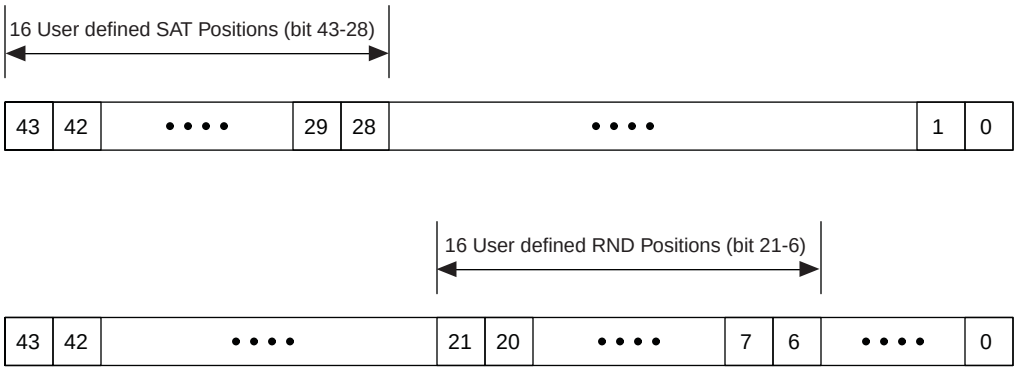
Table 4–8. Examples of Saturation

44- to 36-Bits Saturation	Symmetric SAT Result	Asymmetric SAT Result
5926AC01342h	7FFFFFFFh	7FFFFFFFh
ADA38D2210h	800000001h	800000000h

Stratix IV devices have up to 16 configurable bit positions out of the 44-bit bus ([43:0]) for the rounding and saturate logic unit providing higher flexibility. These 16-bit positions are located at bits [21:6] for rounding and [43:28] for saturation, as shown in Figure 4–21.

You must select the 16 configurable bit positions at compile time.

Figure 4–21. Rounding and Saturation Locations



For symmetric saturation, the RND bit position is also used to determine where the LSP for the saturated data is located.

You can use the rounding and saturation function described above in regular supported multiplication operations, as specified in Table 4–2 on page 4–8. However, for accumulation type operations, use the following convention:

The functionality of the round logic unit is in the format of:

Result = RND[S(A × B)], when used for an accumulation type of operation.

Likewise, the functionality of the saturation logic unit is in the format of:

Result = SAT[S(A × B)], when used for an accumulation type of operation.

If both the rounding and saturation logic units are used for an accumulation type of operation, the format is:

Result = SAT[RND[S(A × B)]]

DSP Block Control Signals

The Stratix IV DSP block is configured using a set of static and dynamic signals. You can configure the DSP block dynamic signals and can be set to toggle or not at run time. Table 4–9 lists the dynamic signals for the DSP block.

Table 4–9. DSP Block Dynamic Signals (Part 1 of 2)

Signal Name	Function	Count
■ signa ■ signb	Signed/unsigned control for all multipliers and adders. ■ signa for "multiplicand" input bus to dataa[17:0] each multiplier. ■ signb for "multiplier" input bus datab[17:0] to each multiplier. ■ signa = 1, signb = 1 for signed-signed multiplication ■ signa = 1, signb = 0 for signed-unsigned multiplication ■ signa = 0, signb = 1 for unsigned-signed multiplication ■ signa = 0, signb = 0 for unsigned-unsigned multiplication	2
output_round	Round control for first stage round and saturation block. ■ output_round = 1 for rounding on multiply output ■ output_round = 0 for normal multiply output	1
chainout_round	Round control for second stage round and saturation block. chainout_round = 1 for rounding multiply output chainout_round = 0 for normal multiply output	1
output_saturate	Saturation control for first stage round and saturation block for Q-format multiply. If both rounding and saturation is enabled, saturation is done on the rounded result. output_saturate = 1 for saturation support output_saturate = 0 for no saturation support	1
chainout_saturate	Saturation control for second stage round and saturation block for Q-format multiply. If both rounding and saturation is enabled, saturation is done on the rounded result. chainout_saturate = 1 for saturation support chainout_saturate = 0 for no saturation support	1

Table 4–9. DSP Block Dynamic Signals (Part 2 of 2)

Signal Name	Function	Count
accum_sload	Dynamically specifies whether the accumulator value is zero. accum_sload = 0, accumulation input is from the output registers accum_sload = 1, accumulation input is set to zero	1
zero_chainout	Dynamically specifies whether the chainout value is zero.	1
zero_loopback	Dynamically specifies whether the loopback value is zero.	1
rotate	rotate = 1, rotation feature is enabled	1
shift_right	shift_right = 1, shift right feature is enabled	1
Total Signals per Half Block		**11**
clock0 clock1 clock2 clock3	DSP-block-wide clock signals.	4
ena0 ena1 ena2 ena3	Input and Pipeline Register enable signals.	4
aclr0 aclr1 aclr2 aclr3	DSP block-wide asynchronous clear signals (active low).	4
Total Count per Full Block		**34**

Software Support

Altera provides two distinct methods for implementing various modes of the DSP block in a design: instantiation and inference. Both methods use the following Quartus II megafunctions:

- lpm_mult
- altmult_add
- altmult_accum
- altfp_mult

To use the DSP block, instantiate the megafunctions in the Quartus II software. Alternatively, with inference, you can create an HDL design and synthesize it using a third-party synthesis tool (such as LeonardoSpectrum™, Synplify, or Quartus II Native Synthesis) that infers the appropriate megafunction by recognizing multipliers, multiplier adders, multiplier accumulators, and shift functions. Using either method, the Quartus II software maps the functionality to the DSP blocks during compilation.

For instructions about using these megafunctions and the MegaWizard Plug-In Manager, refer to the *Quartus II Software Help*.

For more information, refer to the "Synthesis" section in volume 1 of the *Quartus II Development Software Handbook*.

Document Revision History

Table 4–10 shows the revision history for this chapter.

Table 4–10. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated Table 4–1. ■ Updated "Stratix IV Simplified DSP Operation" section. ■ Updated graphics. ■ Minor text edits.	—
June 2009 v2.3	■ Added an introductory paragraph to increase search ability. ■ Removed the Conclusion section.	—
April 2009 v2.2	■ Updated Table 4–1.	—
March 2009 v2.1	■ Updated Table 4–1. ■ Removed "Referenced Documents" section.	—
November 2008 v2.0	■ Updated Table 4–2. ■ Updated Figure 4–16. ■ Updated Figure 4–18.	—
May 2008 v1.0	Initial Release.	—

5. Clock Networks and PLLs in Stratix IV Devices

SIV51005-3.0

This chapter describes the hierarchical clock networks and phase-locked loops (PLLs) with advanced features in Stratix® IV devices. It includes details about the ability to reconfigure the PLL counter clock frequency and phase shift in real time, allowing you to sweep PLL output frequencies and dynamically adjust the output clock phase shift.

The Quartus® II software enables the PLLs and their features without external devices. The following sections describe the Stratix IV clock networks and PLLs in detail:

Clock Networks in Stratix IV Devices

The global clock networks (GCLKs), regional clock networks (RCLKs), and periphery clock networks (PCLKs) available in Stratix IV devices are organized into hierarchical clock structures that provide up to 236 unique clock domains (16 GCLKs + 88 RCLKs + 132 PCLKs) within the Stratix IV device and allow up to 71 unique GCLK, RCLK, and PCLK clock sources (16 GCLKs + 22 RCLKs + 33 PCLKs) per device quadrant. Table 5–1 lists the clock resources available in Stratix IV devices.

Table 5–1. Clock Resources in Stratix IV Devices

Clock Resource	Number of Resources Available	Source of Clock Resource
Clock input pins	32 Single-ended (16 Differential)	CLK[0..15]p and CLK[0..15]n pins
GCLK networks	16	CLK[0..15]p and CLK[0..15]n pins, PLL clock outputs, and logic array
RCLK networks	64/88 *(1)*	CLK[0..15]p and CLK[0..15]n pins, PLL clock outputs, and logic array
PCLK networks	56/88/112/132 (33 per device quadrant) *(2)*	DPA clock outputs, PLD-transceiver interface clocks, horizontal I/O pins, and logic array
GCLKs/RCLKs per quadrant	32/38 *(3)*	16 GCLKs + 16 RCLKs/ 16 GCLKs + 22 RCLKs
GCLKs/RCLKs per device	80/104 *(4)*	16 GCLKs + 64 RCLKs/ 16 GCLKs + 88 RCLKs

Notes to Table 5–1:

(1) There are 64 RCLKs in the EP4S40G2, EP4S100G2, EP4SE230, EP4SGX70, EP4SGX110, EP4SGX180, and EP4SGX230 devices. There are 88 RCLKs in the EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5, EP4SE360, EP4SE530, EP4SE820, EP4SGX290, EP4SGX360, and EP4SGX530 devices.

(2) There are 56 PCLKs in the EP4SGX70, and EP4SGX110 devices. There are 88 PCLKs in the EP4S40G2, EP4S100G2, EP4SE230, EP4SE360, EP4SGX180, EP4SGX230, EP4SGX290, and EP4SGX360 devices. There are 112 PCLKs in the EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5, EP4SE530 and EP4SGX530 devices. There are 132 PCLKs in the EP4SE820 device.

(3) There are 32 GCLKs/RCLKs per quadrant in the EP4S40G2, EP4S100G2, EP4SE230, EP4SGX70, EP4SGX110, EP4SGX180, and EP4SGX230 devices. There are 38 GCLKs/RCLKs per quadrant in the EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5, EP4SE360, EP4SE530, EP4SE820, EP4SGX290, EP4SGX360, and EP4SGX530 devices.

(4) There are 80 GCLKs/RCLKs per entire device in the EP4S40G2, EP4S100G2, EP4SE230, EP4SGX70, EP4SGX110, EP4SGX180, and EP4SGX230 devices. There are 104 GCLKs/RCLKS per entire device in the EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5, EP4SE360, EP4SE530, EP4SE820, EP4SGX290, EP4SGX360, and EP4SGX530 devices.

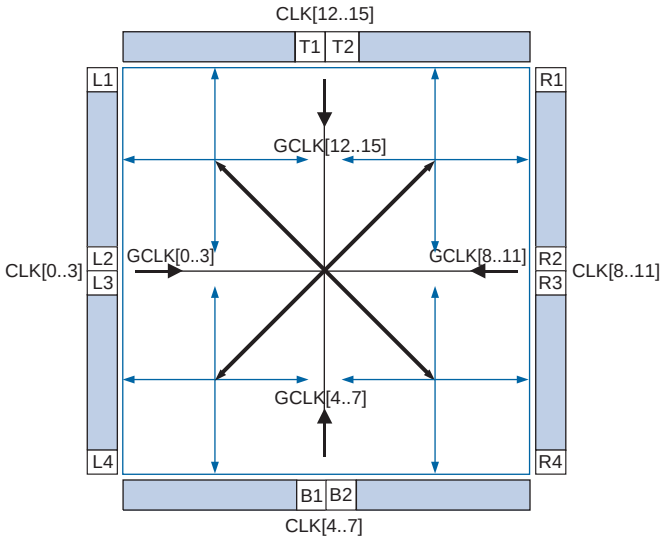
Stratix IV devices have up to 32 dedicated single-ended clock pins or 16 dedicated differential clock pins (CLK[0..15]p and CLK[0..15]n) that can drive either the GCLK or RCLK networks. These clock pins are arranged on the four sides of the Stratix IV device, as shown in Figure 5–1 through Figure 5–4 on page 5–4.

For more information about how to connect the clock input pins, refer to the *Stratix IV Device Family Pin Connection Guidelines*.

Global Clock Networks

Stratix IV devices provide up to 16 GCLKs that can drive throughout the device, serving as low-skew clock sources for functional blocks such as adaptive logic modules (ALMs), digital signal processing (DSP) blocks, TriMatrix memory blocks, and PLLs. Stratix IV device I/O elements (IOEs) and internal logic can also drive GCLKs to create internally generated global clocks and other high fan-out control signals; for example, synchronous or asynchronous clears and clock enables. Figure 5–1 shows the CLK pins and PLLs that can drive the GCLK networks in Stratix IV devices.

Figure 5–1. GCLK Networks



Regional Clock Networks

RCLK networks only pertain to the quadrant they drive into. RCLK networks provide the lowest clock delay and skew for logic contained within a single device quadrant. The Stratix IV device IOEs and internal logic within a given quadrant can also drive RCLKs to create internally generated regional clocks and other high fan-out control signals; for example, synchronous or asynchronous clears and clock enables.

Figure 5–2 through Figure 5–4 on page 5–4 show the CLK pins and PLLs that can drive the RCLK networks in Stratix IV devices.

Figure 5–2. RCLK Networks (EP4SE230, EP4SGX70, and EP4SGX110 Devices)

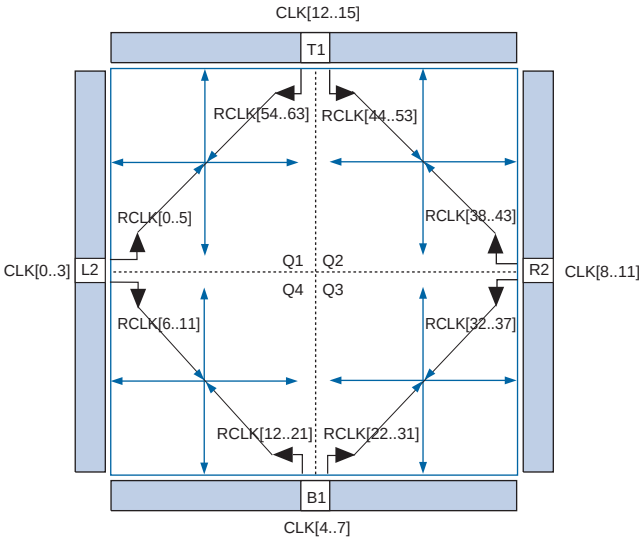


Figure 5–3. RCLK Networks (EP4S40G2, EP4S100G2, EP4SGX180, and EP4SGX230 Devices)

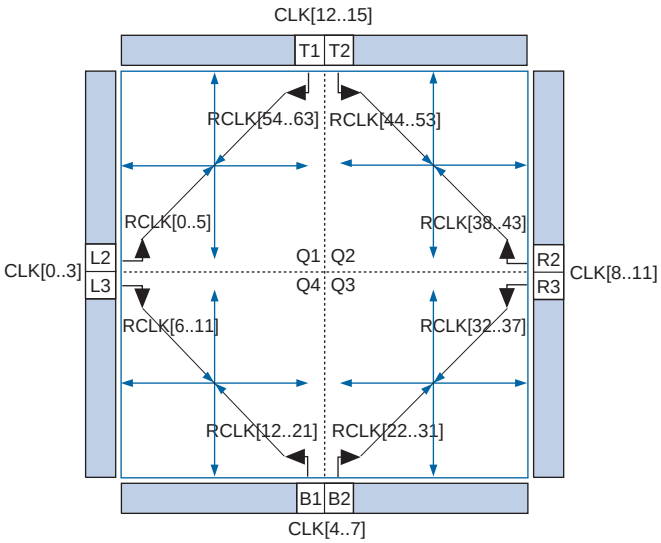
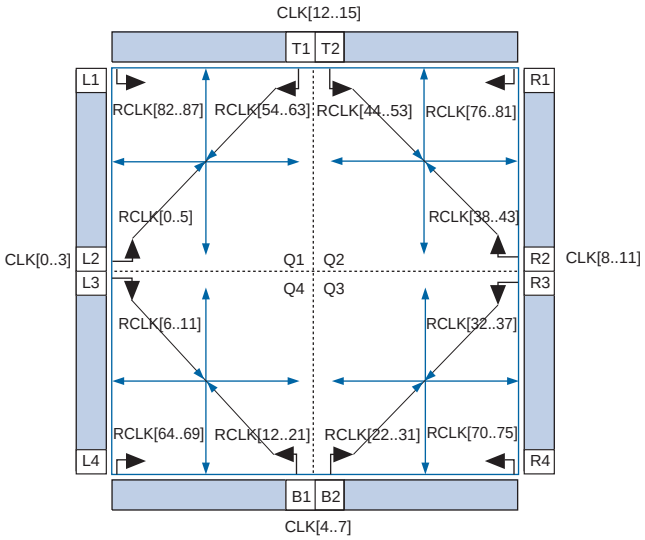


Figure 5–4. RCLK Networks (EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5, EP4SE360, EP4SE530, EP4SE820, EP4SGX290, EP4SGX360, and EP4SGX530 Devices) *(Note 1)*, *(2)*



Notes to Figure 5–4:

(1) The corner `RCLK[64..87]` can only be fed by their respective corner PLL outputs. For more details about connectivity, refer to Table 5–6 on page 5–12.

(2) EP4S40G5 and EP4SE360 devices have up to 8 PLLs. For more details about PLL availability, refer to Table 5–7 on page 5–18.

Periphery Clock Networks

PCLK networks shown in Figure 5–5 to Figure 5–8 on page 5–6 are collections of individual clock networks driven from the periphery of the Stratix IV device. Clock outputs from the dynamic phase aligner (DPA) block, programmable logic device (PLD)-transceiver interface clocks, horizontal I/O pins, and internal logic can drive the PCLK networks.

PCLKs have higher skew when compared with GCLK and RCLK networks. You can use PCLKs for general purpose routing to drive signals into and out of the Stratix IV device.

Legal clock sources for PCLK networks are clock outputs from the DPA block, PLD-transceiver interface clocks, horizontal I/O pins, and internal logic.

Figure 5–5. PCLK Networks (EP4SGX70 and EP4SGX110 Devices)

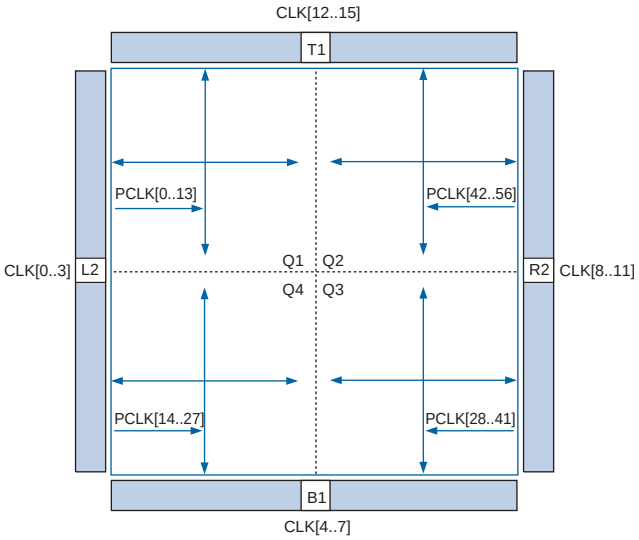
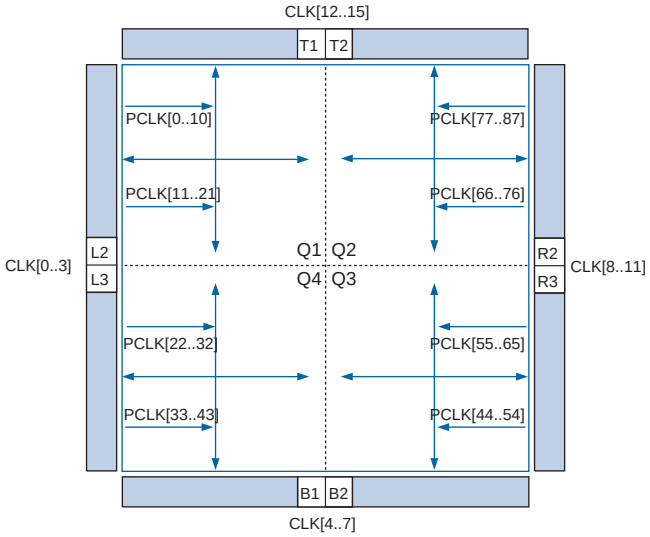


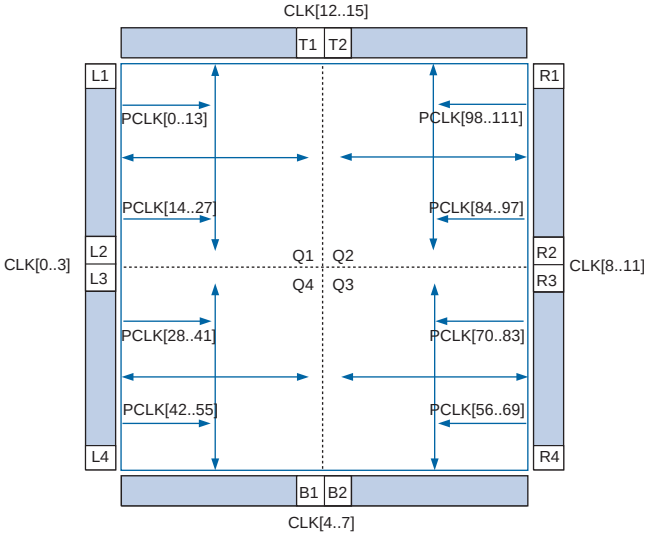
Figure 5–6. PCLK Networks (EP4S40G2, EP4S100G2, EP4SE230, EP4SE360, EP4SGX180, EP4SGX230, EP4SGX290, and EP4SGX360 Devices) *(Note 1)*



Note to Figure 5–6:

(1) EP4SE230 device has 4 PLLs. EP4SGX290 and EP4SGX360 devices have up to 12 PLLs. For more details about PLL availability, refer to Table 5–7 on page 5–18.

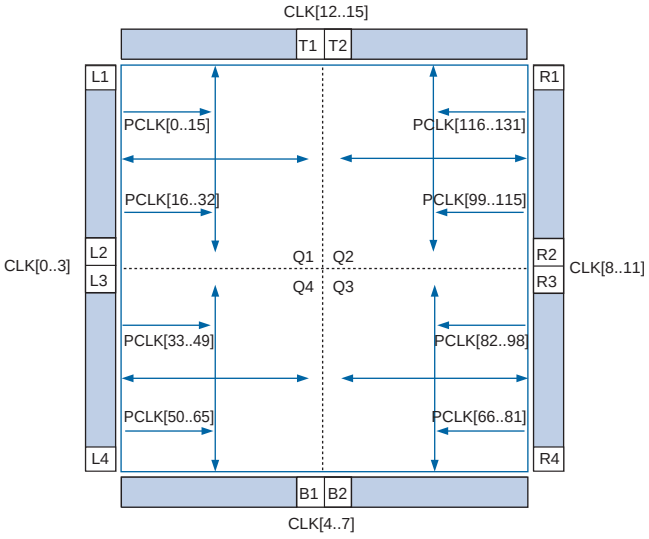
Figure 5–7. PCLK Networks (EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5, EP4SE530, and EP4SGX530 Devices) *(Note 1)*



Note to Figure 5–7:

(1) EP4S40G5 device has 8 PLLs. For more details about PLL availability, refer to Table 5–7 on page 5–18.

Figure 5–8. PCLK Networks (EP4SE820 Device)

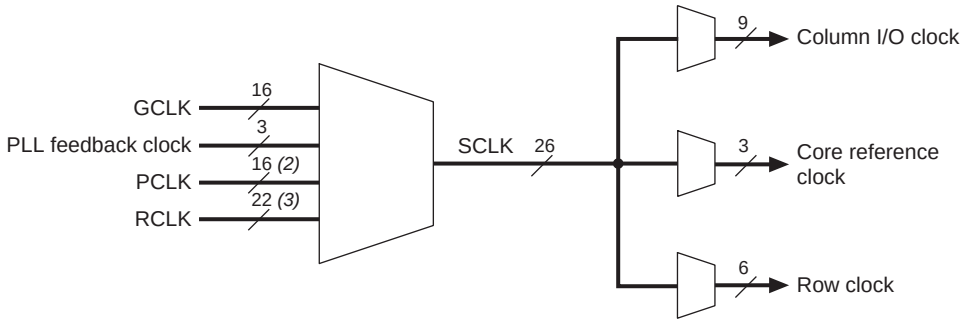


Clock Sources Per Region

There are 26 section clock (SCLK) networks available in each spine clock that can drive 6 row clocks in each logic array block (LAB) row, 9 column I/O clocks, and 3 core reference clocks. The SCLKs are the clock resources to the core functional blocks, PLLs, and I/O interfaces of the device. Figure 5–9 shows that the SCLKs can be driven by the GCLK, RCLK, PCLK, or the PLL feedback clock networks in each spine clock.

A spine clock is another layer of routing below the GCLKs, RCLKs, and PCLKs before each clock is connected to the clock routing for each LAB row. The settings for spine clocks are transparent to all users. The Quartus II software automatically routes the spine clock based on the GCLK, RCLK, and PCLKs.

Figure 5–9. Hierarchical Clock Networks per Spine Clock *(Note 1)*



Notes to Figure 5–9:

(1) The GCLK, RCLK, PCLK, and PLL feedback clocks share the same routing to the SCLKs. The total number of clock resources must not exceed the SCLK limits in each region to ensure successful design fitting in the Quartus II software.

(2) There are up to 16 PCLKs that can drive the SCLKs in each spine clock in the largest device.

(3) There are up to 22 RCLKs that can drive the SCLKs in each spine clock in the largest device.

Clock Regions

Stratix IV devices provide up to 104 distinct clock domains (16 GCLKs + 88 RCLKs) in the entire device. You can use these clock resources to form the following types of clock regions:

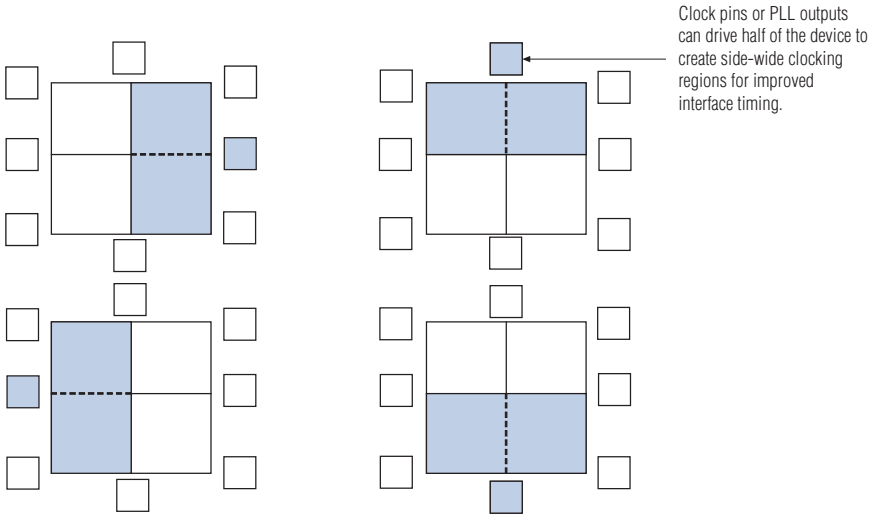
- Entire device
- Regional
- Dual-regional

To form the entire device clock region, a source (not necessarily a clock signal) drives a GCLK network that can be routed through the entire device. This clock region has the maximum delay when compared with other clock regions, but allows the signal to reach every destination within the device. This is a good option for routing global reset and clear signals or routing clocks throughout the device.

To form a RCLK region, a source drives a single quadrant of the device. This clock region provides the lowest skew within a quadrant and is a good option if all the destinations are within a single device quadrant.

To form a dual-regional clock region, a single source (a clock pin or PLL output) generates a dual-regional clock by driving two RCLK networks (one from each quadrant). This technique allows destinations across two device quadrants to use the same low-skew clock. The routing of this signal on an entire side has approximately the same delay as a RCLK region. Internal logic can also drive a dual-regional clock network. Corner PLL outputs only span one quadrant, they cannot generate a dual-regional clock network. Figure 5–10 shows the dual-regional clock region.

Figure 5–10. Stratix IV Dual-Regional Clock Region



Clock Network Sources

In Stratix IV devices, clock input pins, PLL outputs, and internal logic can drive the GCLK and RCLK networks. For the connectivity between dedicated pins CLK[0..15] and the GCLK and RCLK networks, refer to Table 5–2 and Table 5–3 on page 5–10.

Dedicated Clock Input Pins

CLK pins can be either differential clocks or single-ended clocks. Stratix IV devices support 16 differential clock inputs or 32 single-ended clock inputs. You can also use dedicated clock input pins CLK[15..0] for high fan-out control signals such as asynchronous clears, presets, and clock enables for protocol signals such as TRDY and IRDY for PCI through GCLK or RCLK networks.

LABs

You can drive each GCLK and RCLK network using LAB-routing to enable internal logic to drive a high fan-out, low-skew signal.

Stratix IV PLLs cannot be driven by internally generated GCLKs or RCLKs. The input clock to the PLL has to come from dedicated clock input pins or pin/PLL-fed GCLKs or RCLKs.

PLL Clock Outputs

Stratix IV PLLs can drive both GCLK and RCLK networks, as described in Table 5–5 on page 5–11 and Table 5–6 on page 5–12.

Table 5–2 lists the connection between the dedicated clock input pins and GCLKs.

Table 5–2. Clock Input Pin Connectivity to the GCLK Networks (Part 1 of 2)

Clock Resources	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
GCLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—
GCLK4	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK5	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK6	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK7	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK8	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK9	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK10	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK11	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK12	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓
GCLK13	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓

Table 5–2. Clock Input Pin Connectivity to the GCLK Networks (Part 2 of 2)

Clock Resources	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
GCLK14	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓
GCLK15	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓

Table 5–3 lists the connectivity between the dedicated clock input pins and RCLKs in Stratix IV devices. A given clock input pin can drive two adjacent RCLK networks to create a dual-regional clock network.

Table 5–3. Clock Input Pin Connectivity to the RCLK Networks

Clock Resource	CLK (p/n Pins)															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
RCLK [0, 4, 6, 10]	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [1, 5, 7, 11]	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [2, 8]	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [3, 9]	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [12, 16, 20, 22, 26, 30]	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
RCLK [13, 17, 21, 23, 27, 31]	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
RCLK [14, 18, 24, 28]	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
RCLK [15, 19, 25, 29]	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK [32, 36, 38, 42]	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
RCLK [33, 37, 39, 43]	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
RCLK [34, 40]	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—
RCLK [35, 41]	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
RCLK [44, 48, 52, 54, 58, 62]	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—
RCLK [45, 49, 53, 55, 59, 63]	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—
RCLK [46, 50, 56, 60]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—
RCLK [47, 51, 57, 61]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓

Clock Input Connections to the PLLs

Table 5–4 lists the dedicated clock input pin connectivity to Stratix IV PLLs.

Table 5–4. Stratix IV Device PLLs and PLL Clock Pin Drivers

Dedicated Clock Input Pin CLK (p/n Pins)	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
CLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—
CLK4	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK5	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK6	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK7	—	—	—	—	✓	✓	—	—	—	—	—	—
CLK8	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK9	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK10	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK11	—	—	—	—	—	—	✓	✓	✓	✓	—	—
CLK12	—	—	—	—	—	—	—	—	—	—	✓	✓
CLK13	—	—	—	—	—	—	—	—	—	—	✓	✓
CLK14	—	—	—	—	—	—	—	—	—	—	✓	✓
CLK15	—	—	—	—	—	—	—	—	—	—	✓	✓

Clock Output Connections

PLLs in Stratix IV devices can drive up to 20 RCLK networks and four GCLK networks. For Stratix IV PLL connectivity to GCLK networks, refer to Table 5–5. The Quartus II software automatically assigns PLL clock outputs to RCLK and GCLK networks.

Table 5–5 lists how the PLL clock outputs connect to the GCLK networks.

Table 5–5. Stratix IV PLL Connectivity to the GCLK Networks (Part 1 of 2)

Clock Network	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
GCLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK4	—	—	—	—	✓	✓	—	—	—	—	—	—
GCLK5	—	—	—	—	✓	✓	—	—	—	—	—	—
GCLK6	—	—	—	—	✓	✓	—	—	—	—	—	—
GCLK7	—	—	—	—	✓	✓	—	—	—	—	—	—

Table 5–5. Stratix IV PLL Connectivity to the GCLK Networks (Part 2 of 2)

Clock Network	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
GCLK8	—	—	—	—	—	—	✓	✓	✓	✓	—	—
GCLK9	—	—	—	—	—	—	✓	✓	✓	✓	—	—
GCLK10	—	—	—	—	—	—	✓	✓	✓	✓	—	—
GCLK11	—	—	—	—	—	—	✓	✓	✓	✓	—	—
GCLK12	—	—	—	—	—	—	—	—	—	—	✓	✓
GCLK13	—	—	—	—	—	—	—	—	—	—	✓	✓
GCLK14	—	—	—	—	—	—	—	—	—	—	✓	✓
GCLK15	—	—	—	—	—	—	—	—	—	—	✓	✓

Table 5–6 lists how the PLL clock outputs connect to the RCLK networks.

Table 5–6. Stratix IV RCLK Outputs From the PLL Clock Outputs

Clock Resource	PLL Number											
	L1	L2	L3	L4	B1	B2	R1	R2	R3	R4	T1	T2
RCLK[0..11]	—	✓	✓	—	—	—	—	—	—	—	—	—
RCLK[12..31]	—	—	—	—	✓	✓	—	—	—	—	—	—
RCLK[32..43]	—	—	—	—	—	—	—	✓	✓	—	—	—
RCLK[44..63]	—	—	—	—	—	—	—	—	—	—	✓	✓
RCLK[64..69]	—	—	—	✓	—	—	—	—	—	—	—	—
RCLK[70..75]	—	—	—	—	—	—	—	—	—	✓	—	—
RCLK[76..81]	—	—	—	—	—	—	✓	—	—	—	—	—
RCLK[82..87]	✓	—	—	—	—	—	—	—	—	—	—	—

Clock Control Block

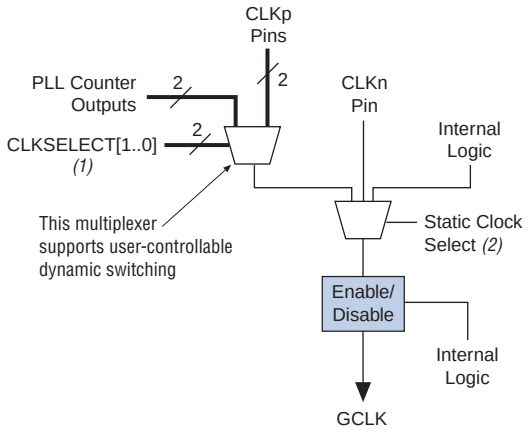
Every GCLK and RCLK network has its own clock control block. The control block provides the following features:

- Clock source selection (dynamic selection for GCLKs)
- Global clock multiplexing
- Clock power down (static or dynamic clock enable or disable)

Figure 5–11 and Figure 5–12 show the GCLK and RCLK select blocks, respectively.

You can select the clock source for the GCLK select block either statically or dynamically. You can statically select the clock source using a setting in the Quartus II software or you can dynamically select the clock source using internal logic to drive the multiplexer-select inputs. When selecting the clock source dynamically, you can select either PLL outputs (such as `C0` or `C1`) or a combination of clock pins or PLL outputs.

Figure 5–11. Stratix IV GCLK Control Block



Notes to Figure 5–11:

(1) When the device is operating in user mode, you can dynamically control the clock select signals through internal logic.

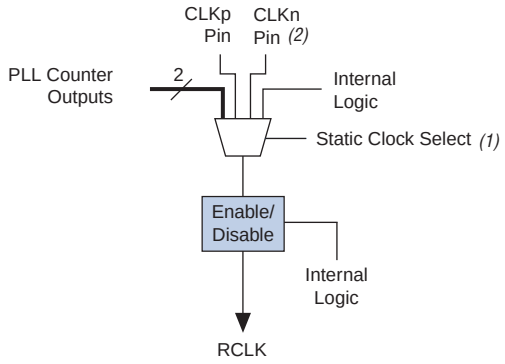
(2) When the device is operation in user mode, you can only set the clock select signals through a configuration file (SRAM object file [**.sof**] or programmer object file [**.pof**]) and cannot be dynamically controlled.

The mapping between the input clock pins, PLL counter outputs, and clock control block inputs is as follows:

- `inclk[0]` and `inclk[1]`—can be fed by any of the four dedicated clock pins on the same side of the Stratix IV device
- `inclk[2]`—can be fed by PLL counters C0 and C2 from the two center PLLs on the same side of the Stratix IV device
- `inclk[3]`—can be fed by PLL counters C1 and C3 from the two center PLLs on the same side of the Stratix IV device

The corner PLLs (L1, L4, R1, and R4) and the corresponding clock input pins (`PLL_L1_CLK` and so forth) do not support dynamic selection for the GCLK network. The clock source selection for the GCLK and RCLK networks from the corner PLLs (L1, L4, R1, and R4) and the corresponding clock input pins (`PLL_L1_CLK` and so forth) is controlled statically using configuration bit settings in the configuration file (**.sof** or **.pof**) generated by the Quartus II software.

Figure 5–12. RCLK Control Block



Notes to Figure 5–12:

(1) When the device is operation in user mode, you can only set the clock select signals through a configuration file (**.sof** or **.pof**) and cannot be dynamically controlled.

(2) The CLKn pin is not a dedicated clock input when used as a single-ended PLL clock input.

You can only control the clock source selection for the RCLK select block statically using configuration bit settings in the configuration file (**.sof** or **.pof**) generated by the Quartus II software.

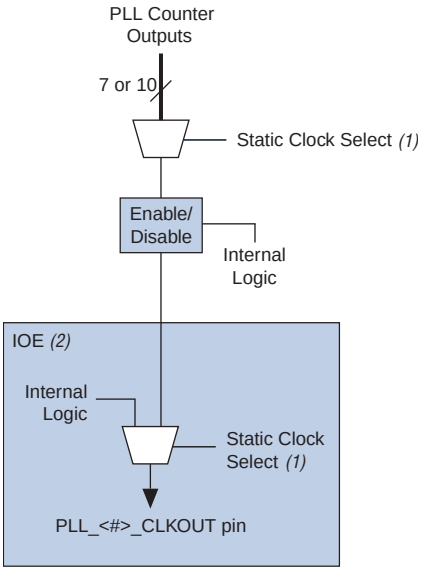
You can power down the Stratix IV clock networks using both static and dynamic approaches. When a clock network is powered down, all the logic fed by the clock network is in off-state, thereby reducing the overall power consumption of the device. The unused GCLK and RCLK networks are automatically powered down through configuration bit settings in the configuration file (**.sof** or **.pof**) generated by the Quartus II software. The dynamic clock enable or disable feature allows the internal logic to control power-up or power-down synchronously on the GCLK and RCLK networks, including dual-regional clock regions. This function is independent of the PLL and is applied directly on the clock network, as shown in Figure 5–11 and Figure 5–12.

You can set the input clock sources and the `clkena` signals for the GCLK and RCLK network multiplexers through the Quartus II software using the ALTCLKCTRL megafunction. You can also enable or disable the dedicated external clock output pins using the ALTCLKCTRL megafunction. Figure 5–13 shows the external PLL output clock control block.

When using the ALTCLKCTRL megafunction to implement dynamic clock source selection, the inputs from the clock pins feed the `inclk[0..1]` ports of the multiplexer, while the PLL outputs feed the `inclk[2..3]` ports. You can choose from among these inputs using the `CLKSELECT[1..0]` signal.

For more information, refer to the *ALTCLKCTRL Megafunction User Guide*.

Figure 5–13. Stratix IV External PLL Output Clock Control Block



Notes to Figure 5–13:

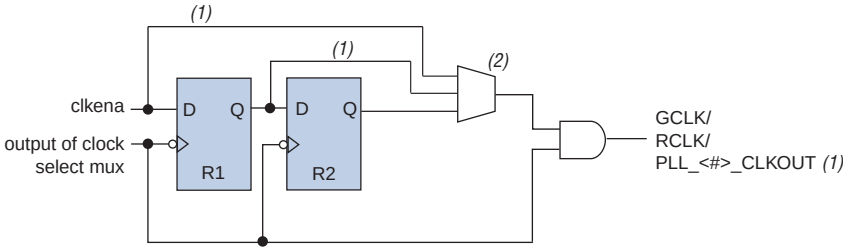
(1) When the device is operation in user mode, you can only set the clock select signals through a configuration file (**.sof** or **.pof**) and cannot be dynamically controlled.

(2) The clock control block feeds to a multiplexer within the PLL_<#>_CLKOUT pin's IOE. The PLL_<#>_CLKOUT pin is a dual-purpose pin. Therefore, this multiplexer selects either an internal signal or the output of the clock control block.

Clock Enable Signals

Figure 5–14 shows how the clock enable and disable circuit of the clock control block is implemented in Stratix IV devices.

Figure 5–14. clkena Implementation



Notes to Figure 5–14:

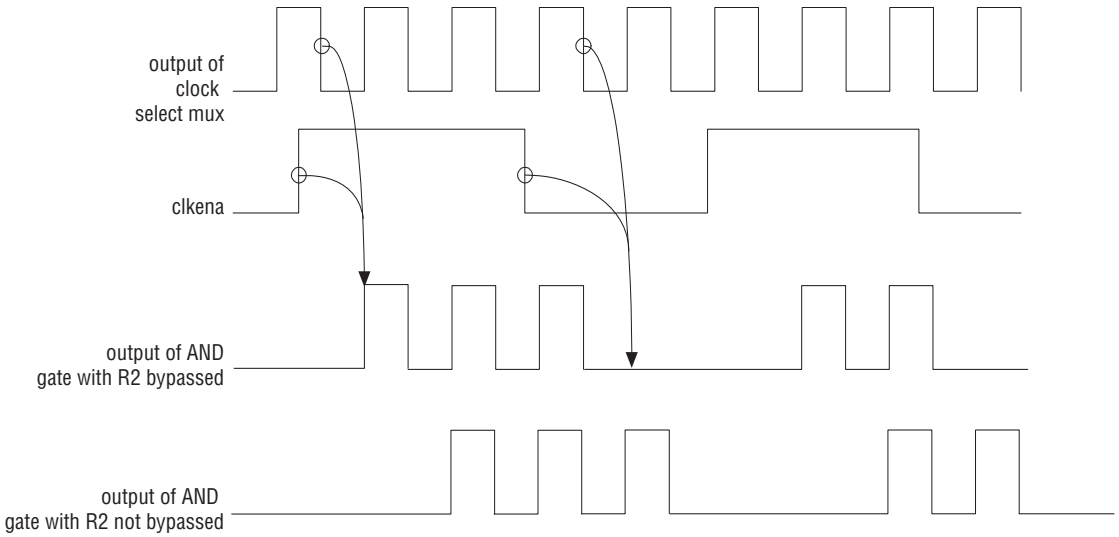
(1) The R1 and R2 bypass paths are not available for the PLL external clock outputs.

(2) The select line is statically controlled by a bit setting in the configuration file (**.sof** or **.pof**).

In Stratix IV devices, the clkena signals are supported at the clock network level instead of at the PLL output counter level. This allows you to gate off the clock even when you are not using a PLL. You can also use the clkena signals to control the dedicated external clocks from the PLLs. Figure 5–15 shows a waveform example for a clock output enable. clkena is synchronous to the falling edge of the clock output.

Stratix IV devices also have an additional metastability register that aids in asynchronous enable and disable of the GCLK and RCLK networks. You can optionally bypass this register in the Quartus II software.

Figure 5–15. clkena Signals *(Note 1)*



Note to Figure 5–15:

(1) You can use the clkena signals to enable or disable the GCLK and RCLK networks or the PLL_<#>_CLKOUT pins.

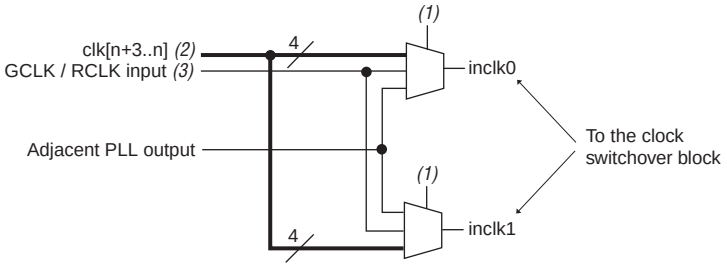
The PLL can remain locked independent of the clkena signals because the loop-related counters are not affected. This feature is useful for applications that require a low-power or sleep mode. The clkena signal can also disable clock outputs if the system is not tolerant of frequency over-shoot during resynchronization.

Clock Source Control for PLLs

The clock input to Stratix IV PLLs comes from clock input multiplexers. The clock multiplexer inputs come from dedicated clock input pins, PLLs through the GCLK and RCLK networks, or from dedicated connections between adjacent top/bottom and left/right PLLs. The clock input sources to top/bottom and left/right PLLs (L2, L3, T1, T2, B1, B2, R2, and R3) are shown in Figure 5–16; the corresponding clock input sources to left and right PLLs (L1, L4, R1, and R4) are shown in Figure 5–17.

The multiplexer select lines are only set in the configuration file (**.sof** or **.pof**). After programmed, this block cannot be changed without loading a new configuration file (**.sof** or **.pof**). The Quartus II software automatically sets the multiplexer select signals depending on the clock sources selected in the design.

Figure 5–16. Clock Input Multiplexer Logic for L2, L3, T1, T2, B1, B2, R2, and R3 PLLs



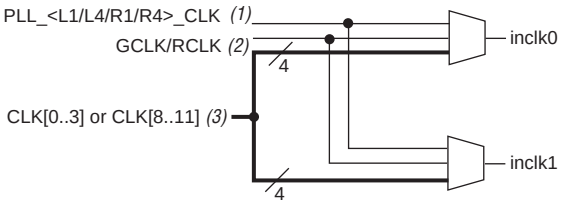
Notes to Figure 5–16:

(1) When the device is operating in user mode, input clock multiplexing is controlled through a configuration file (**.sof** or **.pof**) only and cannot be dynamically controlled.

(2) n=0 for L2 and L3 PLLs; n=4 for B1 and B2 PLLs; n=8 for R2 and R3 PLLs, and n=12 for T1 and T2 PLLs.

(3) You can drive the GCLK or RCLK input using an output from another PLL, a pin-driven GCLK or RCLK, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated GCLK or RCLK. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

Figure 5–17. Clock Input Multiplexer Logic for L1, L4, R1, and R4 PLLs



Notes to Figure 5–17:

(1) Dedicated clock input pins to the PLLs are L1, L4, R1, and R4, respectively. For example, `PLL_L1_CLK` is the dedicated clock input for `PLL_L1`.

(2) You can drive the GCLK or RCLK input using an output from another PLL, a pin-driven GCLK or RCLK, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated GCLK or RCLK. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

(3) The center clock pins can feed the corner PLLs on the same side directly through a dedicated path. However, these paths may not be fully compensated.

Cascading PLLs

You can cascade the left/right and top/bottom PLLs through the GCLK and RCLK networks. In addition, where two left/right or top/bottom PLLs exist next to each other, there is a direct connection between them that does not require the GCLK or RCLK network. Using this path reduces clock jitter when cascading PLLs.

Stratix IV GX devices allow cascading the left and right PLLs to transceiver PLLs (CMU PLLs and receiver CDRs).

For more information, refer to the "FPGA Fabric PLLs -Transceiver PLLs Cascading" section in the *Stratix IV Transceiver Clocking* chapter.

When cascading PLLs in Stratix IV devices, the source (upstream) PLL must have a low-bandwidth setting while the destination (downstream) PLL must have a high-bandwidth setting. Ensure that there is no overlap of the bandwidth ranges of the two PLLs.

For more information about PLL cascading in external memory interfaces designs, refer to the *External Memory PHY Interface Megafunction User Guide (ALTMEMPHY)*.

PLLs in Stratix IV Devices

Stratix IV devices offer up to 12 PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces. The nomenclature for the PLLs follows their geographical location in the device floor plan. The PLLs that reside on the top and bottom sides of the device are named PLL_T1, PLL_T2, PLL_B1 and PLL_B2; the PLLs that reside on the left and right sides of the device are named PLL_L1, PLL_L2, PLL_L3, PLL_L4, PLL_R1, PLL_R2, PLL_R3, and PLL_R4.

Table 5–7 lists the number of PLLs available in the Stratix IV device family.

Table 5–7. Stratix IV Device PLL Availability (Part 1 of 2)

Device	Package	L1	L2	L3	L4	T1	T2	B1	B2	R1	R2	R3	R4
EP4S40G2	F1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP4S40G5	H1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP4S100G2	F1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP4S100G3	F1932	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP4S100G4	F1932	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP4S100G5	H1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
	F1932	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP4SE230	F780	—	✓	—	—	✓	—	✓	—	—	✓	—	—
EP4SE360	H780	—	✓	—	—	✓	—	✓	—	—	✓	—	—
	F1152	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP4SE530	H1152	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
	H1517	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F1760	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP4SE820	H1152	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
	H1517	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F1760	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP4SGX70	F780	—	✓	—	—	✓	—	✓	—	—	—	—	—
	F1152	—	✓	—	—	✓	—	✓	—	—	✓	—	—
EP4SGX110	F780	—	✓	—	—	✓	—	✓	—	—	—	—	—
	F1152	—	✓	—	—	✓	—	✓	—	—	✓	—	—
EP4SGX180	F780	—	✓	—	—	✓	—	✓	—	—	—	—	—
	F1152	—	✓	—	—	✓	✓	✓	✓	—	✓	—	—
	F1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
EP4SGX230	F780	—	✓	—	—	✓	—	✓	—	—	—	—	—
	F1152	—	✓	—	—	✓	✓	✓	✓	—	✓	—	—
	F1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—

Table 5–7. Stratix IV Device PLL Availability (Part 2 of 2)

Device	Package	L1	L2	L3	L4	T1	T2	B1	B2	R1	R2	R3	R4
EP4SGX290	H780	—	—	—	—	✓	✓	✓	✓	—	—	—	—
	F1152	—	✓	—	—	✓	✓	✓	✓	—	✓	—	—
	F1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
	F1760	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F1932	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP4SGX360	H780	—	—	—	—	✓	✓	✓	✓	—	—	—	—
	F1152	—	✓	—	—	✓	✓	✓	✓	—	✓	—	—
	F1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
	F1760	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F1932	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
EP4SGX530	H1152	—	✓	—	—	✓	✓	✓	✓	—	✓	—	—
	H1517	—	✓	✓	—	✓	✓	✓	✓	—	✓	✓	—
	F1760	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	F1932	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

All Stratix IV PLLs have the same core analog structure with only minor differences in the features that are supported. Table 5–8 lists the features of top/bottom and left/right PLLs in Stratix IV devices.

Table 5–8. Stratix IV PLL Features (Part 1 of 2)

Feature	Stratix IV Top/Bottom PLLs	Stratix IV Left/Right PLLs
C (output) counters	10	7
M, N, C counter sizes	1 to 512	1 to 512
Dedicated clock outputs	6 single-ended or 4 single-ended and 1 differential pair	2 single-ended or 1 differential pair
Clock input pins	4 single-ended or 2 differential pin pairs	4 single-ended or 2 differential pin pairs
External feedback input pin	Single-ended or differential	Single-ended only
Spread-spectrum input clock tracking	Yes *(1)*	Yes *(1)*
PLL cascading	Through GCLK and RCLK and a dedicated path between adjacent PLLs	Through GCLK and RCLK and dedicated path between adjacent PLLs *(2)*
Compensation modes	All except LVDS clock network compensation	All except external feedback mode when using differential I/Os
PLL drives LVDSCLK and LOADEN	No	Yes
VCO output drives the DPA clock	No	Yes
Phase shift resolution	Down to 96.125 ps *(3)*	Down to 96.125 ps *(3)*
Programmable duty cycle	Yes	Yes
Output counter cascading	Yes	Yes

Table 5–8. Stratix IV PLL Features (Part 2 of 2)

Feature	Stratix IV Top/Bottom PLLs	Stratix IV Left/Right PLLs
Input clock switchover	Yes	Yes

Notes to Table 5–8:

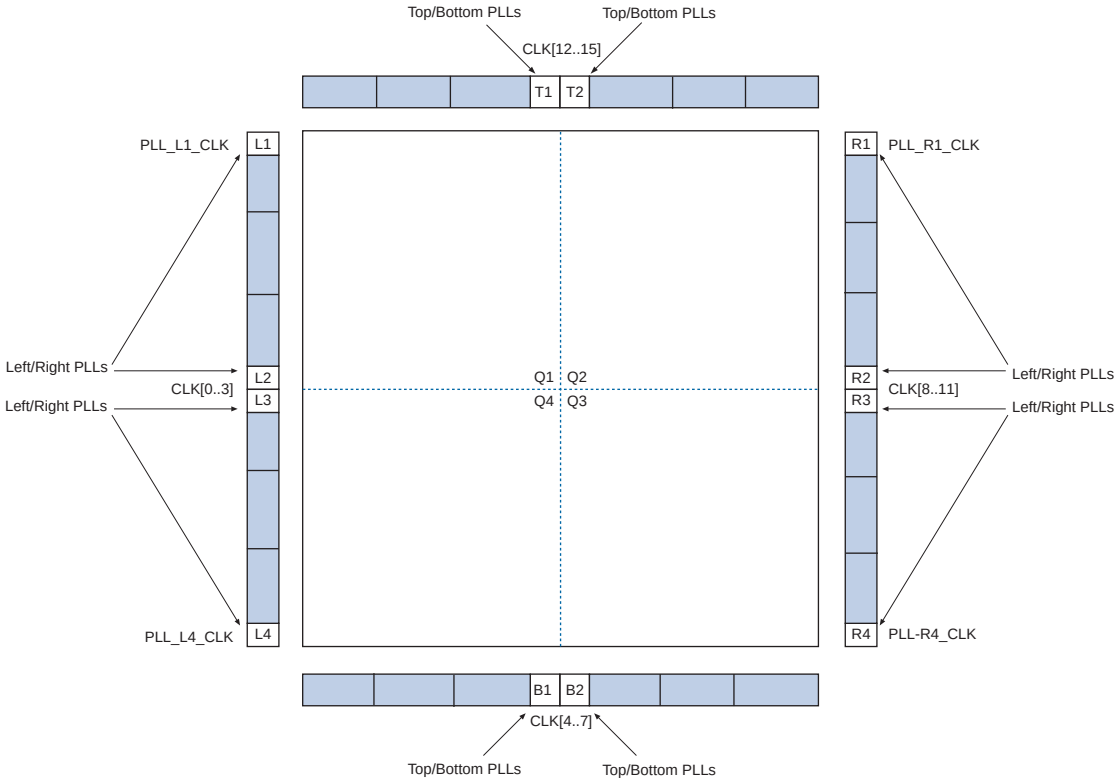
(1) Provided input clock jitter is within input jitter tolerance specifications.

(2) The dedicated path between adjacent PLLs is not available on L1, L4, R1, and R4 PLLs.

(3) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by eight. For degree increments, the Stratix IV device can shift all output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the frequency and divide parameters.

Figure 5–18 shows the location of PLLs in Stratix IV devices.

Figure 5–18. Stratix IV PLL Locations



Stratix IV PLL Hardware Overview

Stratix IV devices contain up to 12 PLLs with advanced clock management features. The goal of a PLL is to synchronize the phase and frequency of an internal or external clock to an input reference clock. There are a number of components that comprise a PLL to achieve this phase alignment.

Stratix IV PLLs align the rising edge of the input reference clock to a feedback clock using the phase-frequency detector (PFD). The falling edges are determined by the duty-cycle specifications. The PFD produces an up or down signal that determines whether the VCO needs to operate at a higher or lower frequency. The output of the PFD feeds the charge pump and loop filter, which produces a control voltage for

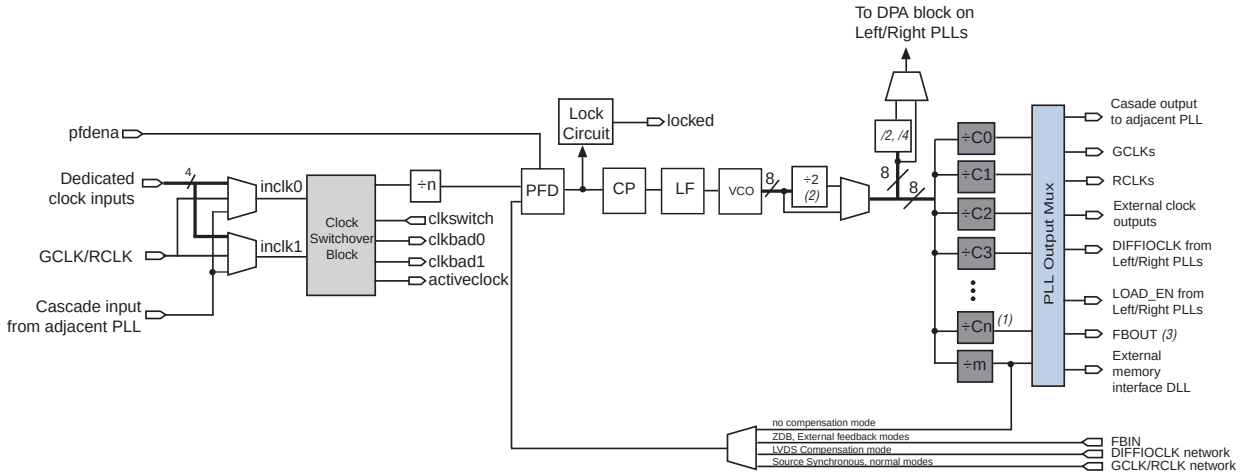
setting the VCO frequency. If the PFD produces an up signal, the VCO frequency increases. A down signal decreases the VCO frequency. The PFD outputs these up and down signals to a charge pump. If the charge pump receives an up signal, current is driven into the loop filter. Conversely, if the charge pump receives a down signal, current is drawn from the loop filter.

The loop filter converts these up and down signals to a voltage that is used to bias the VCO. The loop filter also removes glitches from the charge pump and prevents voltage over-shoot, which filters the jitter on the VCO. The voltage from the loop filter determines how fast the VCO operates. A divide counter (`m`) is inserted in the feedback loop to increase the VCO frequency above the input reference frequency. VCO frequency (f_{VCO}) is equal to (`m`) times the input reference clock (f_{REF}). The input reference clock (f_{REF}) to the PFD is equal to the input clock (f_{IN}) divided by the pre-scale counter (`N`). Therefore, the feedback clock (f_{FB}) applied to one input of the PFD is locked to the f_{REF} that is applied to the other input of the PFD.

The VCO output from the left and right PLLs can feed seven post-scale counters (`C[0..6]`), while the corresponding VCO output from the top and bottom PLLs can feed ten post-scale counters (`C[0..9]`). These post-scale counters allow a number of harmonically related frequencies to be produced by the PLL.

Figure 5–19 shows a simplified block diagram of the major components of the Stratix IV PLL.

Figure 5–19. Stratix IV PLL Block Diagram



Notes to Figure 5–19:

(1) The number of post-scale counters is seven for left and right PLLs and ten for top and bottom PLLs.
(2) This is the VCO post-scale counter `K`.
(3) The `FBOUT` port is fed by the `M` counter in Stratix IV PLLs.

You can drive the GCLK or RCLK inputs using an output from another PLL, a pin-driven GCLK or RCLK, or through a clock control block provided the clock control block is fed by an output from another PLL or a pin-driven dedicated GCLK or RCLK. An internally generated global signal or general purpose I/O pin cannot drive the PLL.

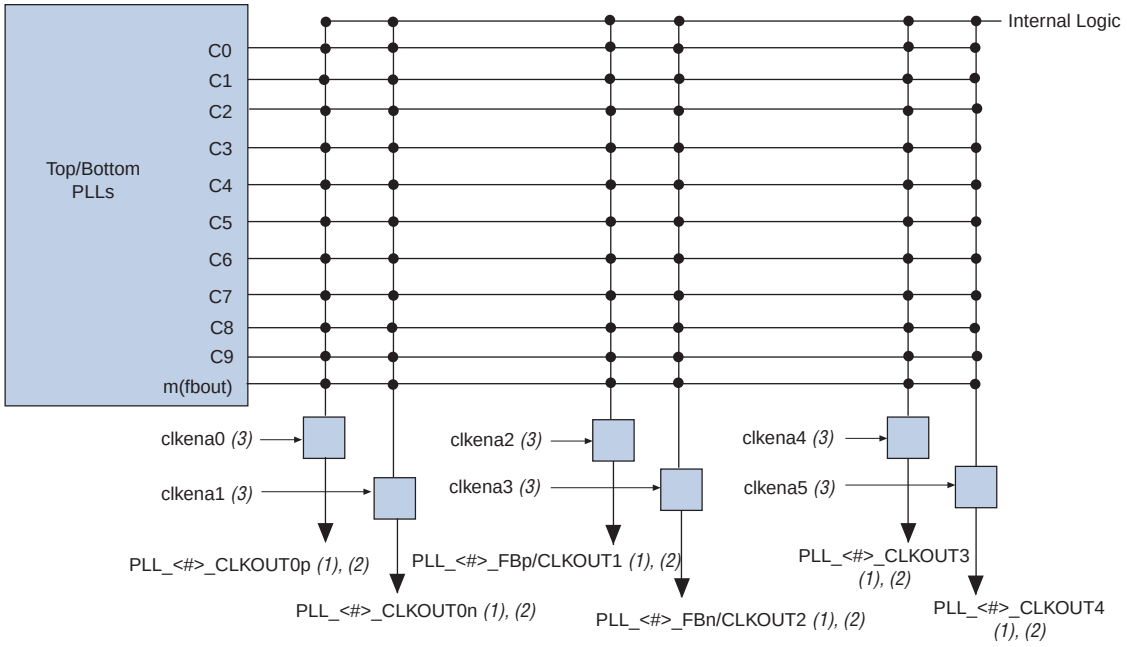
PLL Clock I/O Pins

Each top and bottom PLL supports six clock I/O pins, organized as three pairs of pins:

- 1st pair—two single-ended I/O or one differential I/O
- 2nd pair—two single-ended I/O or one differential external feedback input (FBp/FBn)
- 3rd pair—two single-ended I/O or one differential input

Figure 5–20 shows the clock I/O pins associated with the top and bottom PLLs.

Figure 5–20. External Clock Outputs for Top and Bottom PLLs



Notes to Figure 5–20:

(1) You can feed these clock output pins using any one of the C[9..0], m counters.

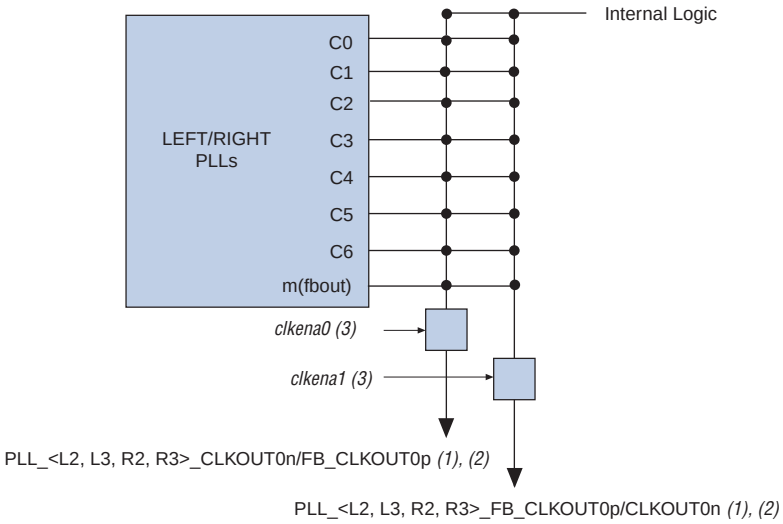
(2) The CLKOUT0p and CLKOUT0n pins can be either single-ended or differential clock outputs. The CLKOUT1 and CLKOUT2 pins are dual-purpose I/O pins that you can use as two single-ended outputs or one differential external feedback input pin. The CLKOUT3 and CLKOUT4 pins are two single-ended output pins.

(3) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

Any of the output counters (C[9..0] on the top and bottom PLLs and C[6..0] on the left and right PLLs) or the M counter can feed the dedicated external clock outputs, as shown in Figure 5–20 and Figure 5–21. Therefore, one counter or frequency can drive all output pins available from a given PLL.

Each left and right PLL supports two clock I/O pins, configured as either two single-ended I/Os or one differential I/O pair. When using both pins as single-ended I/Os, one of them can be the clock output while the other pin is the external feedback input (FB) pin. Therefore, for single-ended I/O standards, the left and right PLLs only support external feedback mode.

Figure 5–21. External Clock Outputs for Left and Right PLLs



Notes to Figure 5–21:

(1) You can feed these clock output pins using any one of the `C[6..0]`, m counters.

(2) The `CLKOUT0p` and `CLKOUT0n` pins are dual-purpose I/O pins that you can use as two single-ended outputs or one single-ended output and one external feedback input pin.

(3) These external clock enable signals are available only when using the ALTCLKCTRL megafunction.

Each pin of a single-ended output pair can either be in-phase or 180° out-of-phase. The Quartus II software places the NOT gate in the design into the IOE to implement the 180° phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins (in the top and bottom banks) as well as LVDS, LVPECL, differential High-Speed Transceiver Logic (HSTL), and differential SSTL.

To determine which I/O standards are supported by the PLL clock input and output pins, refer to the *I/O Features in Stratix IV Devices* chapter.

Stratix IV PLLs can also drive out to any regular I/O pin through the GCLK or RCLK network. You can also use the external clock output pins as user I/O pins if you do not need external PLL clocking.

PLL Control Signals

You can use the three signals, pfdena, areset, and locked, to observe and control PLL operation and resynchronization.

pfdena

Use the pfdena signal to maintain the most recent locked frequency so your system has time to store its current settings before shutting down. The pfdena signal controls the PFD output with a programmable gate. If you disable PFD, the VCO operates at its most recent set value of control voltage and frequency, with some long-term drift to a lower frequency. The PLL continues running even if it goes out-of-lock or the input clock is disabled. You can use either your own control signal or the control signals available from the clock switchover circuit (activeclock, clkbad[0], or clkbad[1]) to control pfdena.

areset

The areset signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals. When areset is driven high, the PLL counters reset, clearing the PLL output and placing the PLL out-of-lock. The VCO is then set back to its nominal setting. When areset is driven low again, the PLL resynchronizes to its input as it re-locks.

You must assert the areset signal every time the PLL loses lock to guarantee the correct phase relationship between the PLL input and output clocks. You can set up the PLL to automatically reset (self reset) upon a loss-of-lock condition using the Quartus II MegaWizard™ Plug-In Manager. You must include the areset signal in designs if either of the following conditions is true:

- PLL reconfiguration or clock switchover is enabled in the design
- Phase relationships between the PLL input and output clocks need to be maintained after a loss-of-lock condition

If the input clock to the PLL is not toggling or is unstable after power up, assert the areset signal after the input clock is stable and within specifications.

locked

The locked signal output of the PLL indicates that the PLL has locked onto the reference clock and the PLL clock outputs are operating at the desired phase and frequency set in the Quartus II MegaWizard Plug-In Manager. The lock detection circuit provides a signal to the core logic that gives an indication when the feedback clock has locked onto the reference clock both in phase and frequency.

Altera recommends using the areset and locked signals in your designs to control and observe the status of your PLL.

Clock Feedback Modes

Stratix IV PLLs support up to six different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle. Table 5–9 lists the clock feedback modes supported by the Stratix IV device PLLs.

Table 5–9. Clock Feedback Mode Availability

Clock Feedback Mode	Availability	
	Top/Bottom PLLs	Left/Right PLLs
Source-synchronous	Yes	Yes
No-compensation	Yes	Yes
Normal	Yes	Yes
Zero-delay buffer (ZDB)	Yes	Yes
External feedback	Yes	Yes *(2)*
LVDS compensation	No	Yes

Notes to Table 5–9:

(1) The high-bandwidth PLL setting is not supported in the external feedback mode.

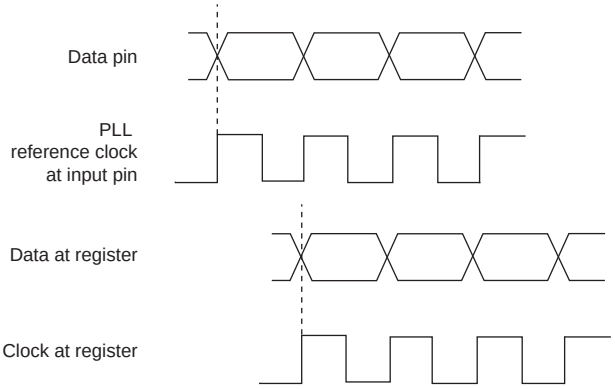
(2) External feedback mode is supported for single-ended inputs and outputs only on the left and right PLLs.

The input and output delays are fully compensated by a PLL only when using the dedicated clock input pins associated with a given PLL as the clock source. For example, when using `PLL_T1` in normal mode, the clock delays from the input pin to the PLL clock output-to-destination register are fully compensated, provided the clock input pin is one of the following four pins: `CLK12`, `CLK13`, `CLK14`, or `CLK15`. When an RCLK or GCLK network drives the PLL, the input and output delays may not be fully compensated in the Quartus II software. Another example is when you configure `PLL_T2` in zero-delay buffer mode and the PLL input is driven by a dedicated clock input pin, a fully compensated clock path results in zero-delay between the clock input and one of the output clocks from the PLL. If the PLL input is instead fed by a non-dedicated input (using the GCLK network), the output clock may not be perfectly aligned with the input clock.

Source Synchronous Mode

If data and clock arrive at the same time on the input pins, the same phase relationship is maintained at the clock and data ports of any IOE input register. Figure 5–22 shows an example waveform of the clock and data in this mode. Altera recommends source synchronous mode for source-synchronous data transfers. Data and clock signals at the IOE experience similar buffer delays as long as you use the same I/O standard.

Figure 5–22. Phase Relationship Between Clock and Data in Source-Synchronous Mode



Source-synchronous mode compensates for the delay of the clock network used plus any difference in the delay between these two paths:

- Data pin to the IOE register input
- Clock input pin to the PLL PFD input

The Stratix IV PLL can compensate multiple pad-to-input-register paths, such as a data bus when it is set to use source-synchronous compensation mode. You can use the "PLL Compensation" assignment in the Quartus II software Assignment Editor to select which input pins are used as the PLL compensation targets. You can include your entire data bus, provided the input registers are clocked by the same output of a source-synchronous-compensated PLL. In order for the clock delay to be properly compensated, all of the input pins must be on the same side of the device. The PLL compensates for the input pin with the longest pad-to-register delay among all input pins in the compensated bus.

If you do not make the "PLL Compensation" assignment, the Quartus II software automatically selects all of the pins driven by the compensated output of the PLL as the compensation target.

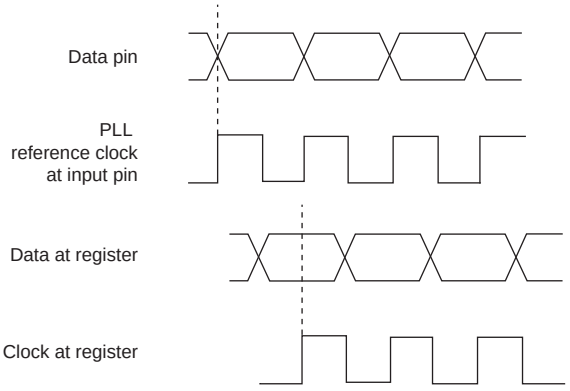
Source-Synchronous Mode for LVDS Compensation

The goal of source-synchronous mode is to maintain the same data and clock timing relationship seen at the pins of the internal serializer/deserializer (SERDES) capture register, except that the clock is inverted (180° phase shift). Thus, source-synchronous mode ideally compensates for the delay of the LVDS clock network plus any difference in delay between these two paths:

- Data pin-to-SERDES capture register
- Clock input pin-to-SERDES capture register. In addition, the output counter must provide the 180° phase shift

Figure 5–23 shows an example waveform of the clock and data in LVDS mode.

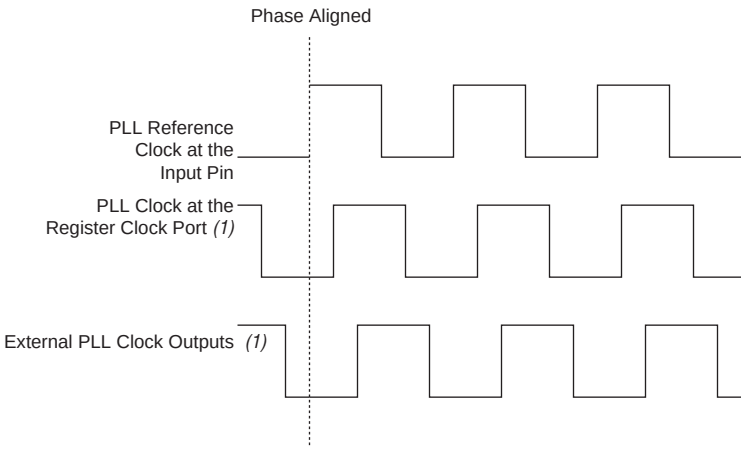
Figure 5–23. Phase Relationship Between the Clock and Data in LVDS Mode



No-Compensation Mode

In no-compensation mode, the PLL does not compensate for any clock networks. This mode provides better jitter performance because the clock feedback into the PFD passes through less circuitry. Both the PLL internal- and external-clock outputs are phase-shifted with respect to the PLL clock input. Figure 5–24 shows an example waveform of the PLL clocks' phase relationship in no-compensation mode.

Figure 5–24. Phase Relationship Between the PLL Clocks in No Compensation Mode



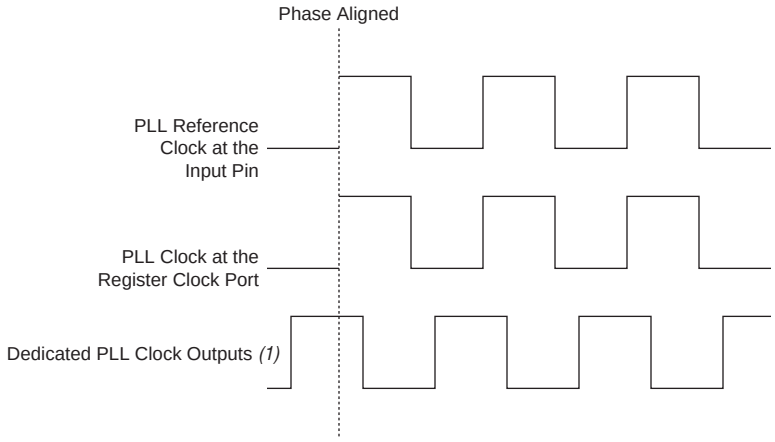
Note to Figure 5–24

(1) The PLL clock outputs lags the PLL input clocks depending on routine delays.

Normal Mode

An internal clock in normal mode is phase-aligned to the input clock pin. The external clock-output pin has a phase delay relative to the clock input pin if connected in this mode. The Quartus II software timing analyzer reports any phase difference between the two. In normal mode, the delay introduced by the GCLK or RCLK network is fully compensated. Figure 5–25 shows an example waveform of the PLL clocks' phase relationship in normal mode.

Figure 5–25. Phase Relationship Between the PLL Clocks in Normal Mode



Note to Figure 5–25:

(1) The external clock output can lead or lag the PLL internal clock signals.

Zero-Delay Buffer Mode

In ZDB mode, the external clock output pin is phase-aligned with the clock input pin for zero-delay through the device. When using this mode, you must use the same I/O standard on the input clocks and output clocks in order to guarantee clock alignment at the input and output pins. ZDB mode is supported on all Stratix IV PLLs.

When using Stratix IV PLLs in ZDB mode, along with single-ended I/O standards, to ensure phase alignment between the CLK pin and the external clock output (`CLKOUT`) pin, you must instantiate a bi-directional I/O pin in the design to serve as the feedback path connecting the `FBOUT` and `FBIN` ports of the PLL. The PLL uses this bi-directional I/O pin to mimic, and compensate for, the output delay from the clock output port of the PLL to the external clock output pin. Figure 5–26 shows ZDB mode in Stratix IV PLLs. When using ZDB mode, you cannot use differential I/O standards on the PLL clock input or output pins.

The bi-directional I/O pin that you instantiate in your design must always be assigned a single-ended I/O standard.

When using ZDB mode, to avoid signal reflection, do not place board traces on the bi-directional I/O pin.

Figure 5–26. ZDB Mode in Stratix IV PLLs

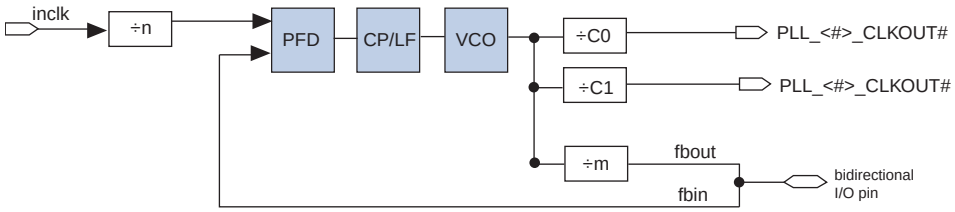
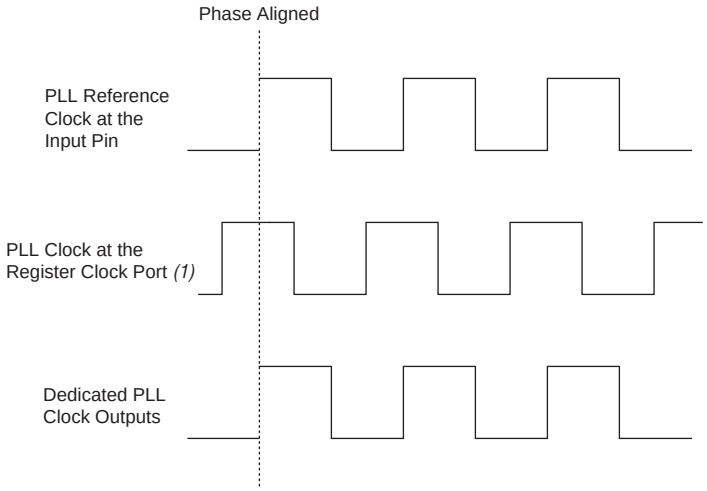


Figure 5–27 shows an example waveform of the PLL clocks' phase relationship in ZDB mode.

Figure 5–27. Phase Relationship Between the PLL Clocks in ZDB Mode



Note to Figure 5–27:

(1) The internal PLL clock output can lead or lag the external PLL clock outputs.

External Feedback Mode

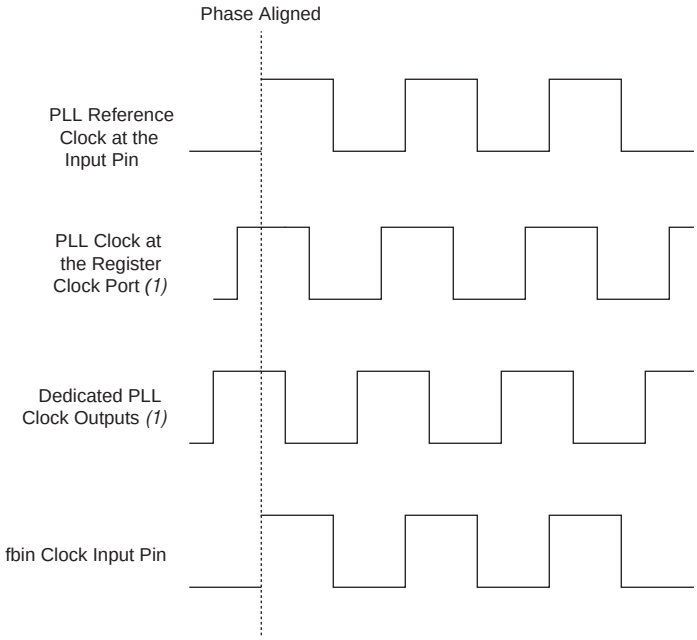
In external feedback mode, the external feedback input pin (`fbin`) is phase-aligned with the clock input pin, as shown in Figure 5–28. Aligning these clocks allows you to remove clock delay and skew between devices. This mode is supported on all Stratix IV PLLs.

In external feedback mode, the output of the `M` counter (`FBOUT`) feeds back to the PLL `fbin` input (using a trace on the board) becoming part of the feedback loop. Also, use one of the dual-purpose external clock outputs as the `fbin` input pin in this mode.

When using external feedback mode, you must use the same I/O standard on the input clock, feedback input, and output clocks. Left and right PLLs support this mode when using single-ended I/O standards only.

Figure 5–28 shows an example waveform of the phase relationship between the PLL clocks in external feedback mode.

Figure 5–28. Phase Relationship Between the PLL Clocks in External Feedback Mode

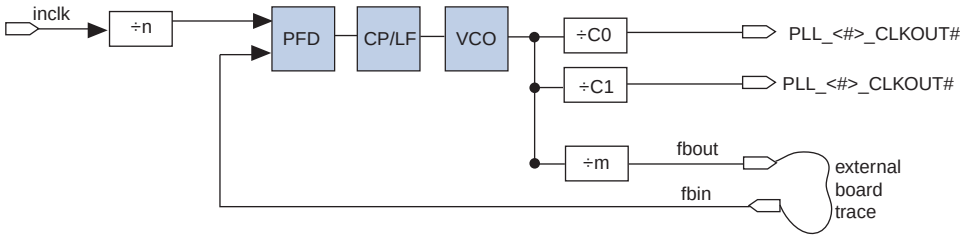


Note to Figure 5–28:

(1) The PLL clock outputs can lead or lag the `fbin` clock input.

Figure 5–29 shows external feedback mode implementation in Stratix IV devices.

Figure 5–29. External Feedback Mode in Stratix IV Devices



Clock Multiplication and Division

Each Stratix IV PLL provides clock synthesis for PLL output ports using $M/(N$* post-scale counter) scaling factors. The input clock is divided by a pre-scale factor, n, and is then multiplied by the *m* feedback factor. The control loop drives the VCO to match f_{in} (M/N). Each output port has a unique post-scale counter that divides down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO is set to the least common multiple of the output frequencies that meets its frequency specifications. For example, if the output frequencies required from one PLL are 33 and 66 MHz, the Quartus II software sets the VCO to 660 MHz (the least common multiple of 33 and 66 MHz within the VCO range). Then the post-scale counters scale down the VCO frequency for each output port.

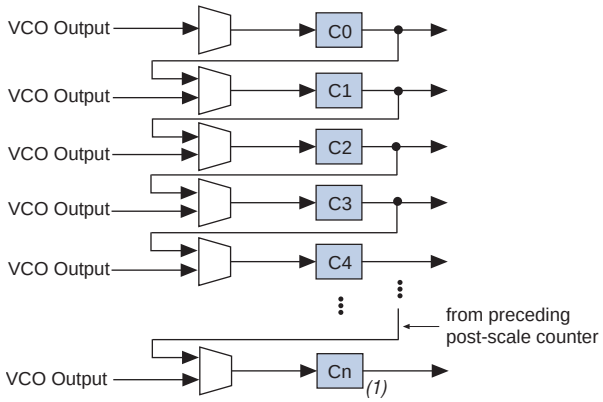
Each PLL has one pre-scale counter, n, and one multiply counter, m, with a range of 1 to 512 for both m and n. The n counter does not use duty-cycle control because the only purpose of this counter is to calculate frequency division. There are seven generic post-scale counters per left or right PLL and ten post-scale counters per top or bottom PLL that can feed the GCLKs, RCLKs, or external clock outputs. These post-scale counters range from 1 to 512 with a 50% duty cycle setting. The high- and low-count values for each counter range from 1 to 256. The sum of the high- and low-count values chosen for a design selects the divide value for a given counter.

The Quartus II software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTPLL megafunction.

Post-Scale Counter Cascading

Stratix IV PLLs support post-scale counter cascading to create counters larger than 512. This is automatically implemented in the Quartus II software by feeding the output of one C counter into the input of the next C counter, as shown in Figure 5–30.

Figure 5–30. Counter Cascading



Note to Figure 5–30:

(1) N = 6 or N = 9

When cascading post-scale counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings. For example, if C0 = 40 and C1 = 20, the cascaded value is C0 × C1 = 800.

Post-scale counter cascading is set in the configuration file. You cannot set this using PLL reconfiguration.

Programmable Duty Cycle

The programmable duty cycle allows PLLs to generate clock outputs with a variable duty cycle. This feature is supported on the PLL post-scale counters. The duty-cycle setting is achieved by a low and high time-count setting for the post-scale counters. To determine the duty cycle choices, the Quartus II software uses the frequency input and the required multiply or divide rate. The post-scale counter value determines the precision of the duty cycle. The precision is defined as 50% divided by the post-scale counter value. For example, if the `C0` counter is 10, steps of 5% are possible for duty-cycle choices from 5% to 90%.

If the PLL is in external feedback mode, set the duty cycle for the counter driving the `fbin` pin to 50%. Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

Programmable Phase Shift

Use phase shift to implement a robust solution for clock delays in Stratix IV devices. Implement phase shift by using a combination of the VCO phase output and the counter starting time. A combination of VCO phase output and counter starting time is the most accurate method of inserting delays because it is only based on counter settings, which are independent of process, voltage, and temperature (PVT).

You can phase-shift the output clocks from the Stratix IV PLLs in either of these two resolutions:

- Fine resolution using VCO phase taps
- Coarse resolution using counter starting time

Implement fine-resolution phase shifts by allowing any of the output counters (`C[n..0]`) or the `m` counter to use any of the eight phases of the VCO as the reference clock. This allows you to adjust the delay time with a fine resolution. Equation 5–1 shows the minimum delay time that you can insert using this method.

Equation 5–1. Fine-Resolution Phase Shift

$$\Phi_{fine} = \frac{1}{8} T_{VCO} = \frac{1}{8 f_{VCO}} = \frac{N}{8 M f_{REF}}$$

where f_{REF} is the input reference clock frequency.

For example, if f_{REF} is 100 MHz, N is 1, and M is 8, then f_{VCO} is 800 MHz and Φ_{fine} equals 156.25 ps. This phase shift is defined by the PLL operating frequency, which is governed by the reference clock frequency and the counter settings.

Equation 5–2 shows the coarse-resolution phase shifts are implemented by delaying the start of the counters for a predetermined number of counter clocks.

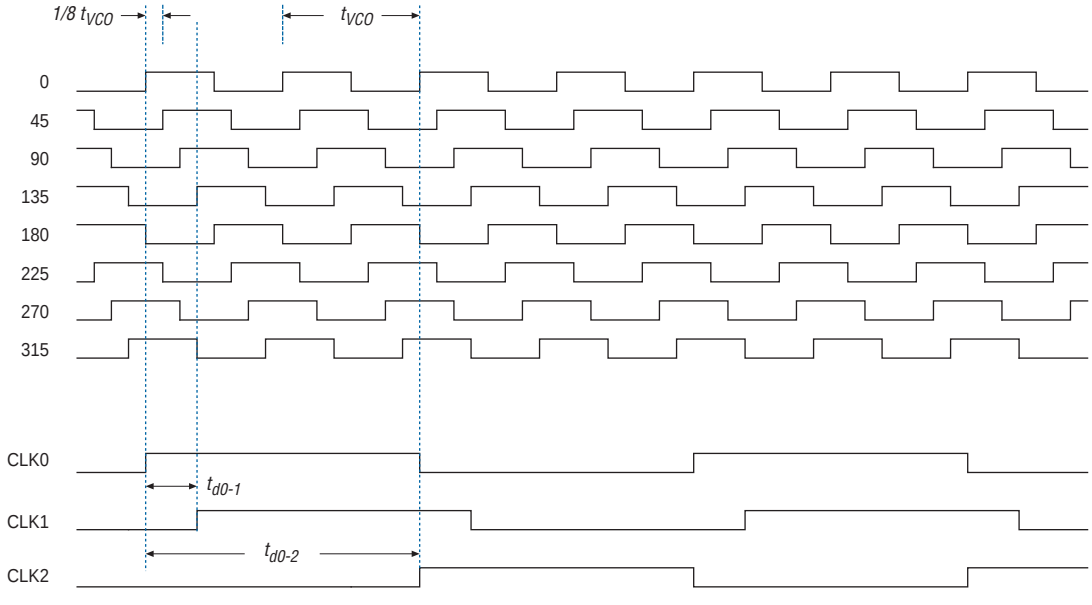
Equation 5–2. Coarse-Resolution Phase Shift

$$\Phi_{coarse} = \frac{C-1}{f_{VCO}} = \frac{(C-1)N}{Mf_{REF}}$$

where C is the count value set for the counter delay time (this is the initial setting in the "PLL usage" section of the compilation report in the Quartus II software). If the initial value is 1, C – 1 = 0° phase shift.

Figure 5–31 shows an example of phase-shift insertion with fine resolution using the VCO phase-taps method. The eight phases from the VCO are shown and labeled for reference. For this example, `CLK0` is based on the `0phase` from the VCO and has the C value for the counter set to one. The `CLK1` signal is divided by four, two VCO clocks for high time and two VCO clocks for low time. `CLK1` is based on the 135° phase tap from the VCO and also has the C value for the counter set to one. In this case, the two clocks are offset by 3 Φ_{FINE}. `CLK2` is based on the `0phase` from the VCO but has the C value for the counter set to **three**. This arrangement creates a delay of 2 Φ_{COARSE} (two complete VCO periods).

Figure 5–31. Delay Insertion Using VCO Phase Output and Counter Delay Time



You can use coarse- and fine-phase shifts to implement clock delays in Stratix IV devices.

Stratix IV devices support dynamic phase-shifting of VCO phase taps only. You can reconfigure the phase shift any number of times. Each phase shift takes about one `SCANCLK` cycle, allowing you to implement large phase shifts quickly.

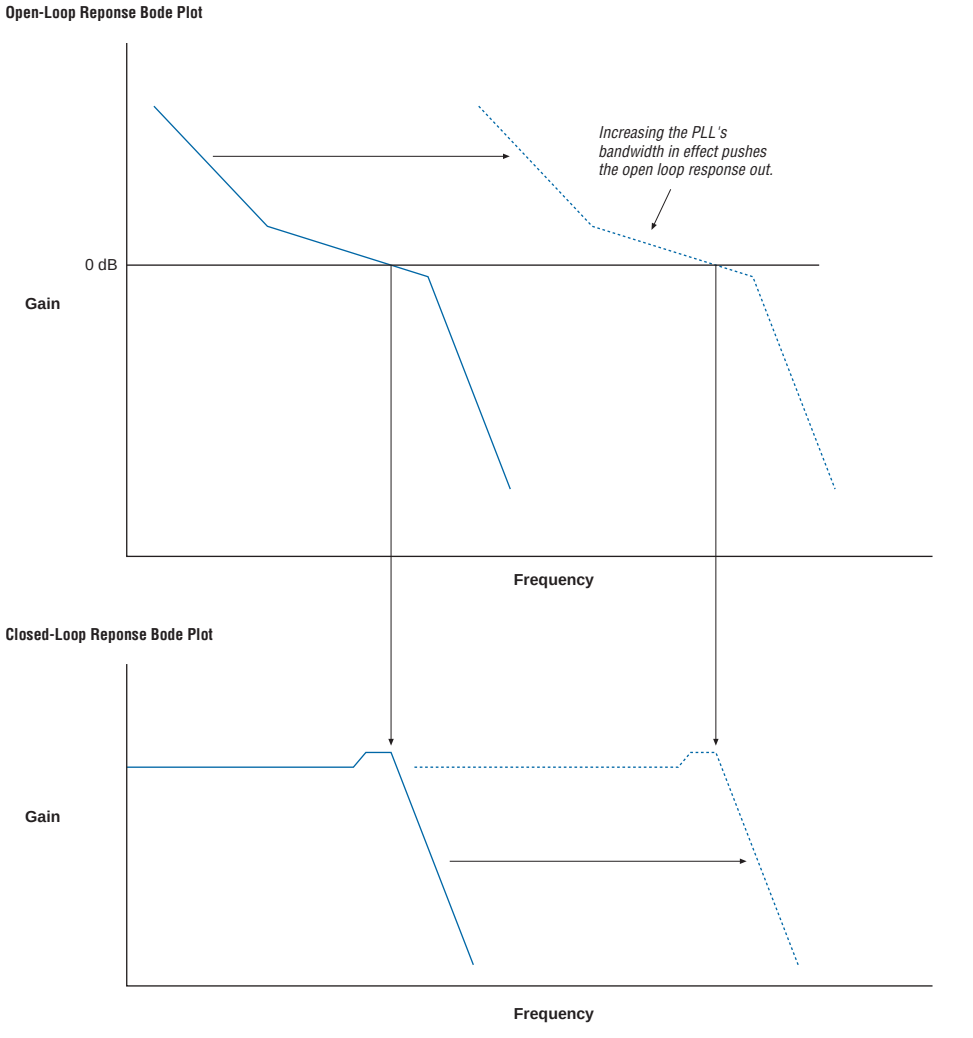
Programmable Bandwidth

Stratix IV PLLs provide advanced control of the PLL bandwidth using the PLL loop's programmable characteristics, including loop filter and charge pump.

Background

PLL bandwidth is the measure of the PLL's ability to track the input clock and its associated jitter. The closed-loop gain 3 dB frequency in the PLL determines PLL bandwidth. Bandwidth is approximately the unity gain point for open loop PLL response. As Figure 5–32 shows, these points correspond to approximately the same frequency. Stratix IV PLLs provide three bandwidth settings—low, medium (default), and high.

Figure 5–32. Open- and Closed-Loop Response Bode Plots



A high-bandwidth PLL provides a fast lock time and tracks jitter on the reference clock source, passing it through to the PLL output. A low-bandwidth PLL filters out reference clock jitter but increases lock time. Stratix IV PLLs allow you to control the bandwidth over a finite range to customize the PLL characteristics for a particular application. The programmable bandwidth feature in Stratix IV PLLs benefits applications requiring clock switchover.

A high-bandwidth PLL can benefit a system that must accept a spread-spectrum clock signal. Stratix IV PLLs can track a spread-spectrum clock by using a high-bandwidth setting. Using a low-bandwidth setting in this case could cause the PLL to filter out the jitter on the input clock.

A low-bandwidth PLL can benefit a system using clock switchover. When clock switchover occurs, the PLL input temporarily stops. A low-bandwidth PLL reacts more slowly to changes on its input clock and takes longer to drift to a lower frequency (caused by the input stopping) than a high-bandwidth PLL.

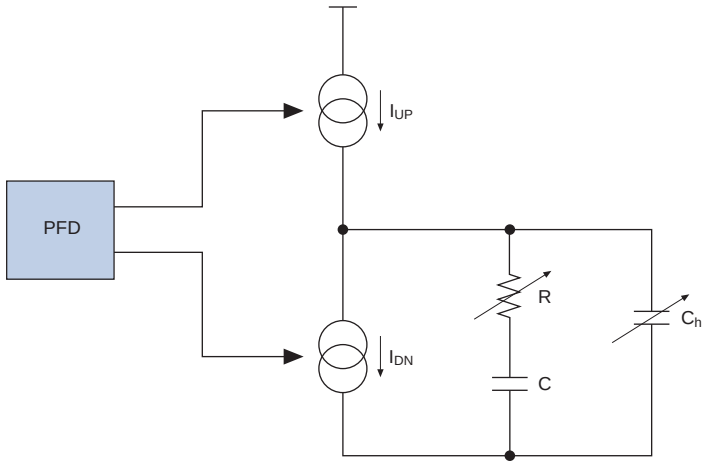
Implementation

Traditionally, external components such as the VCO or loop filter control a PLL's bandwidth. Most loop filters consist of passive components such as resistors and capacitors that take up unnecessary board space and increase cost. With Stratix IV PLLs, all the components are contained within the device to increase performance and decrease cost.

When you specify the bandwidth setting (low, medium, or high) in the ALTPLL MegaWizard Plug-in Manager, the Quartus II software automatically sets the corresponding charge pump and loop filter (`Icp`, `R`, `C`) values to achieve the desired bandwidth range.

Figure 5–33 shows the loop filter and components that you can set using the Quartus II software. The components are the loop filter resistor, R, the high frequency capacitor, C_h, and the charge pump current, I_{UP} or I_{DN}.

Figure 5–33. Loop Filter Programmable Components



Spread-Spectrum Tracking

Stratix IV devices can accept a spread-spectrum input with typical modulation frequencies. However, the device cannot automatically detect that the input is a spread-spectrum signal. Instead, the input signal looks like deterministic jitter at the input of the PLL. Stratix IV PLLs can track a spread-spectrum input clock as long as it is within the input-jitter tolerance specifications. Stratix IV devices cannot internally generate spread-spectrum clocks.

Clock Switchover

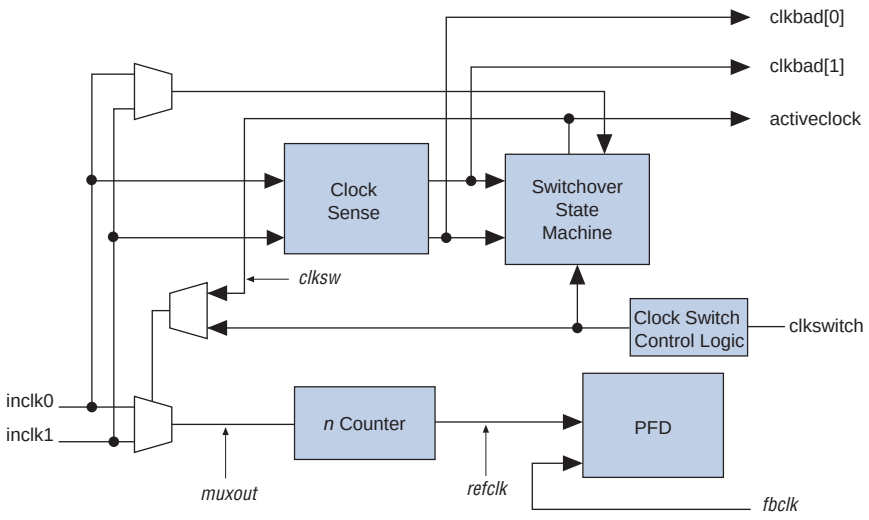
The clock switchover feature allows the PLL to switch between two reference input clocks. Use this feature for clock redundancy or for a dual-clock domain application such as in a system that turns on the redundant clock if the previous clock stops running. The design can perform clock switchover automatically when the clock is no longer toggling or based on a user control signal, `clkswitch`.

The following clock switchover modes are supported in Stratix IV PLLs:

- Automatic switchover—The clock sense circuit monitors the current reference clock and if it stops toggling, automatically switches to the other `inclk0` or `inclk1` clock.
- Manual clock switchover—Clock switchover is controlled using the `clkswitch` signal. When the `clkswitch` signal goes from logic low to logic high, and stays high for at least three clock cycles, the reference clock to the PLL is switched from `inclk0` to `inclk1`, or vice-versa.
- Automatic switchover with manual override—This mode combines automatic switchover and manual clock switchover. When the `clkswitch` signal goes high, it overrides the automatic clock switchover mode.

Stratix IV PLLs support a fully configurable clock switchover capability. Figure 5–34 shows a block diagram of the automatic switchover circuit built into the PLL. When the current reference clock is not present, the clock sense block automatically switches to the backup clock for PLL reference. The clock switchover circuit also sends out three status signals—`clkbad[0]`, `clkbad[1]`, and `activeclock`—from the PLL to implement a custom switchover circuit in the logic array. You can select a clock source as the backup clock by connecting it to the `inclk1` port of the PLL in your design.

Figure 5–34. Automatic Clock Switchover Circuit Block Diagram



Automatic Clock Switchover

Use the switchover circuitry to automatically switch between `inclk0` and `inclk1` when the current reference clock to the PLL stops toggling. For example, in applications that require a redundant clock with the same frequency as the reference clock, the switchover state machine generates a signal (`clksw`) that controls the multiplexer select input, as shown in Figure 5–34. In this case, `inclk1` becomes the reference clock for the PLL. When using automatic switchover mode, you can switch back and forth between `inclk0` and `inclk1` any number of times when one of the two clocks fails and the other clock is available.

When using automatic clock switchover mode, the following requirements must be satisfied:

- Both clock inputs must be running
- The period of the two clock inputs can differ by no more than 100% (2×)

If the current clock input stops toggling while the other clock is also not toggling, switchover is not initiated and the `clkbad[0..1]` signals are not valid. Also, if both clock inputs are not the same frequency, but their period difference is within 100%, the clock sense block detects when a clock stops toggling, but the PLL may lose lock after the switchover is completed and needs time to re-lock.

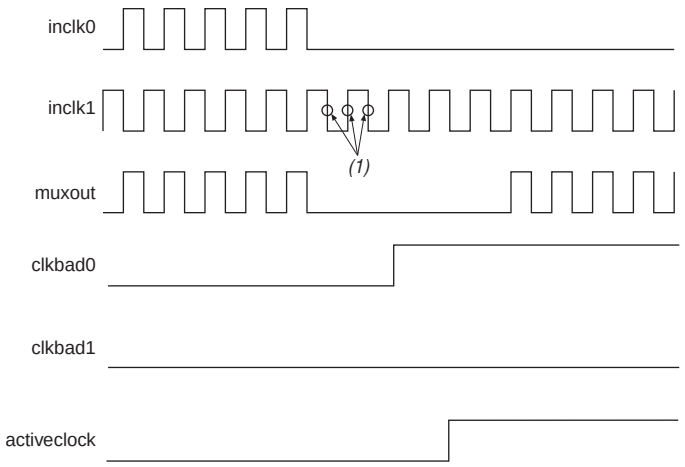
Altera recommends resetting the PLL using the `areset` signal to maintain the phase relationships between the PLL input and output clocks when using clock switchover.

In automatic switchover mode, the `clkbad[0]` and `clkbad[1]` signals indicate the status of the two clock inputs. When they are asserted, the clock sense block has detected that the corresponding clock input has stopped toggling. These two signals are not valid if the frequency difference between `inclk0` and `inclk1` is greater than 20%.

The activeclock signal indicates which of the two clock inputs (inclk0 or inclk1) is being selected as the reference clock to the PLL. When the frequency difference between the two clock inputs is more than 20%, the activeclock signal is the only valid status signal.

Figure 5–35 shows an example waveform of the switchover feature when using automatic switchover mode. In this example, the inclk0 signal is stuck low. After the inclk0 signal is stuck at low for approximately two clock cycles, the clock sense circuitry drives the clkbad[0] signal high. Also, because the reference clock signal is not toggling, the switchover state machine controls the multiplexer through the clkswitch signal to switch to the backup clock, inclk1.

Figure 5–35. Automatic Switchover Upon Loss of Clock Detection



Note to Figure 5–35:

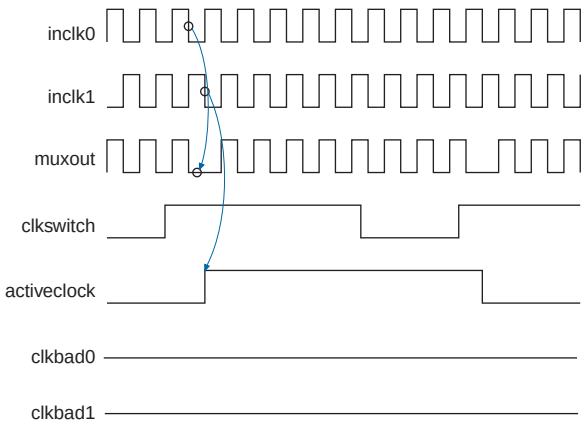
(1) Switchover is enabled on the falling edge of inclk0 or inclk1, depending on which clock is available. In this figure, switchover is enabled on the falling edge of inclk1.

Manual Override

In automatic switchover with manual override mode, you can use the clkswitch input for user- or system-controlled switch conditions. You can use this mode for same-frequency switchover, or to switch between inputs of different frequencies. For example, if inclk0 is 66 MHz and inclk1 is 200 MHz, you must control switchover using clkswitch because the automatic clock-sense circuitry cannot monitor clock input (inclk0 and inclk1) frequencies with a frequency difference of more than 100% (2×). This feature is useful when the clock sources originate from multiple cards on the backplane, requiring a system-controlled switchover between the frequencies of operation. You must choose the backup clock frequency and set the m, n, c, and k counters accordingly so the VCO operates within the recommended operating frequency range of 600 to 1,600 MHz. The ALTPLL MegaWizard Plug-in Manager notifies you if a given combination of inclk0 and inclk1 frequencies cannot meet this requirement.

Figure 5–36 shows an example of a waveform showing the switchover feature when controlled by clkswitch. In this case, both clock sources are functional and inclk0 is selected as the reference clock; clkswitch goes high, which starts the switchover sequence. On the falling edge of inclk0, the counter's reference clock, muxout, is gated off to prevent clock glitching. On the falling edge of inclk1, the reference clock multiplexer switches from inclk0 to inclk1 as the PLL reference and the activeclock signal changes to indicate which clock is currently feeding the PLL.

Figure 5–36. Clock Switchover Using the clkswitch (Manual) Control *(Note 1)*



Note to Figure 5–36:

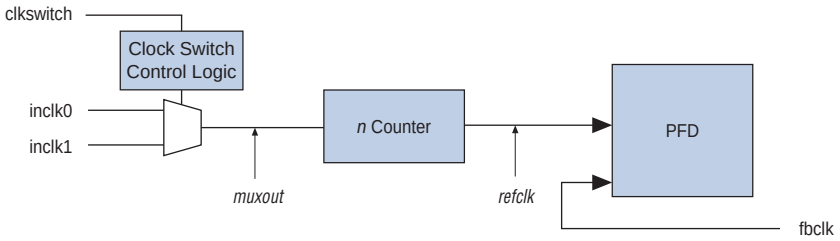
(1) To initiate a manual clock switchover event, both inclk0 and inclk1 must be running when the clkswitch signal goes high.

In automatic override with manual switchover mode, the activeclock signal mirrors the clkswitch signal. As both clocks are still functional during the manual switch, neither clkbad signal goes high. Because the switchover circuit is positive-edge sensitive, the falling edge of the clkswitch signal does not cause the circuit to switch back from inclk1 to inclk0. When the clkswitch signal goes high again, the process repeats. clkswitch and automatic switch only work if the clock being switched to is available. If the clock is not available, the state machine waits until the clock is available.

Manual Clock Switchover

In manual clock switchover mode, the clkswitch signal controls whether inclk0 or inclk1 is selected as the input clock to the PLL. By default, inclk0 is selected. A low-to-high transition on clkswitch and clkswitch being held high for at least three inclk cycles initiates a clock switchover event. You must bring clkswitch back low again in order to perform another switchover event in the future. If you do not require another switchover event in the future, you can leave clkswitch in a logic high state after the initial switch. Pulsing clkswitch high for at least three inclk cycles performs another switchover event. If inclk0 and inclk1 are different frequencies and are always running, the clkswitch minimum high time must be greater than or equal to three of the slower frequency inclk0 or inclk1 cycles. Figure 5–37 shows a block diagram of the manual switchover circuit.

Figure 5–37. Manual Clock Switchover Circuitry in Stratix IV PLLs



For more information about PLL software support in the Quartus II software, refer to the *ALTPLL Megafunction User Guide*.

Guidelines

When implementing clock switchover in Stratix IV PLLs, use the following guidelines:

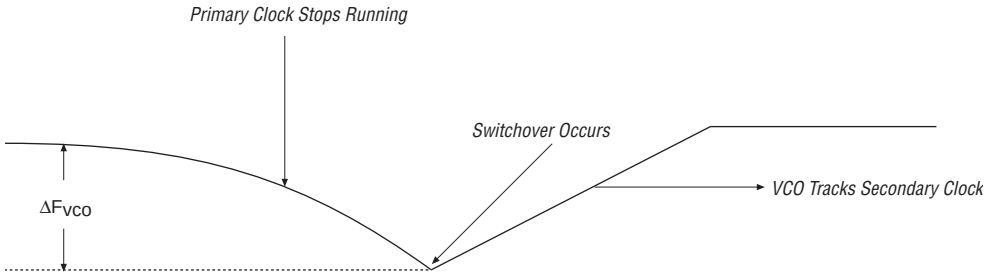
- Automatic clock switchover requires that the `inclk0` and `inclk1` frequencies be within 100% (2×) of each other. Failing to meet this requirement causes the `clkbad[0]` and `clkbad[1]` signals to not function properly.
- When using manual clock switchover, the difference between `inclk0` and `inclk1` can be more than 100% (2×). However, differences in frequency, phase, or both, of the two clock sources will likely cause the PLL to lose lock. Resetting the PLL ensures that the correct phase relationships are maintained between input and output clocks.

  Both `inclk0` and `inclk1` must be running when the `clkswitch` signal goes high to initiate the manual clock switchover event. Failing to meet this requirement causes the clock switchover to not function properly.

- Applications that require a clock switchover feature and a small frequency drift must use a low-bandwidth PLL. The low-bandwidth PLL reacts more slowly than a high-bandwidth PLL to reference input clock changes. When switchover happens, a low-bandwidth PLL propagates the stopping of the clock to the output more slowly than a high-bandwidth PLL. However, be aware that the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The exact amount of time it takes for the PLL to re-lock depends on the PLL configuration.
- The phase relationship between the input clock to the PLL and the output clock from the PLL is important in your design. Assert `areset` for at least 10 ns after performing a clock switchover. Wait for the locked signal to go high and be stable before re-enabling the output clocks from the PLL.

- Figure 5–38 shows how the VCO frequency gradually decreases when the current clock is lost and then increases as the VCO locks on to the backup clock.

Figure 5–38. VCO Switchover Operating Frequency



- Disable the system during clock switchover if it is not tolerant of frequency variations during the PLL resynchronization period. You can use the `clkbad[0]` and `clkbad[1]` status signals to turn off the PFD (`PFDENA` = `0`) so the VCO maintains its most recent frequency. You can also use the state machine to switch over to the secondary clock. When the PFD is re-enabled, output clock-enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. When the lock indication is stable, the system can re-enable the output clocks.

PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In Stratix IV PLLs, you can reconfigure both the counter settings and phase-shift the PLL output clock in real time. You can also change the charge pump and loop-filter components, which dynamically affects PLL bandwidth. You can use these PLL components to update the output-clock frequency and PLL bandwidth and to phase-shift in real time, without reconfiguring the entire Stratix IV device.

The ability to reconfigure the PLL in real time is useful in applications that operate at multiple frequencies. It is also useful in prototyping environments, allowing you to sweep PLL output frequencies and adjust the output-clock phase dynamically. For instance, a system generating test patterns is required to generate and transmit patterns at 75 or 150 MHz, depending on the requirements of the device under test. Reconfiguring the PLL components in real time allows you to switch between two such output frequencies within a few microseconds. You can also use this feature to adjust clock-to-out (`tCO`) delays in real time by changing the PLL output clock phase shift. This approach eliminates the need to regenerate a configuration file with the new PLL settings.

PLL Reconfiguration Hardware Implementation

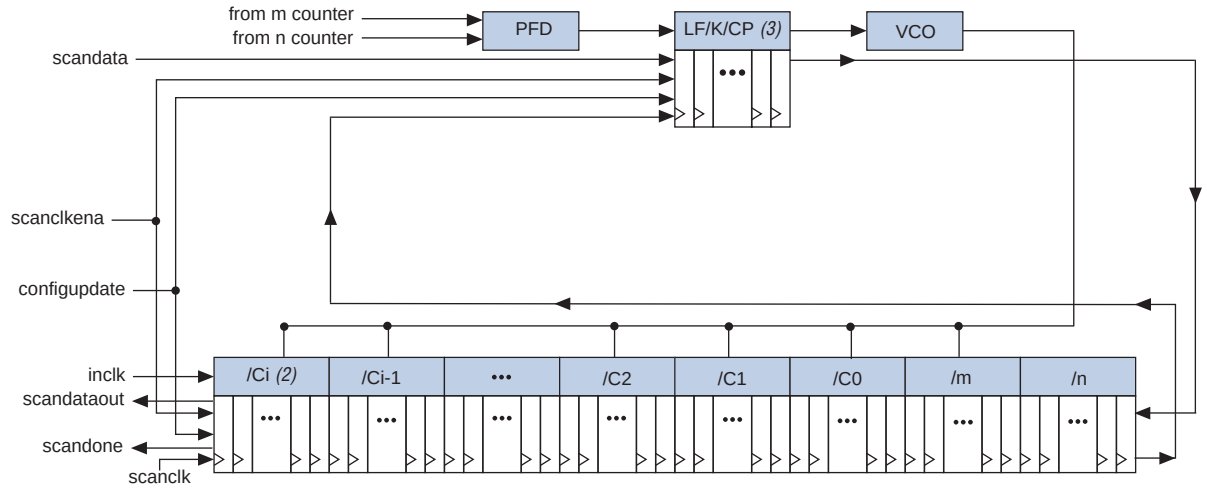
The following PLL components are reconfigurable in real time:

- Pre-scale counter (`n`)
- Feedback counter (`m`)

- Post-scale output counters (C0 - C9)
- Post VCO Divider (K)
- Dynamically adjust the charge-pump current (Icp) and loop-filter components (R, C) to facilitate reconfiguration of the PLL bandwidth

Figure 5–39 shows how you can dynamically adjust the PLL counter settings by shifting their new settings into a serial shift-register chain or scan chain. Serial data is input to the scan chain using the scandata port. Shift registers are clocked by scanclk. The maximum scanclk frequency is 100 MHz. Serial data is shifted through the scan chain as long as the scanclkena signal stays asserted. After the last bit of data is clocked, asserting the configupdate signal for at least one scanclk clock cycle causes the PLL configuration bits to be synchronously updated with the data in the scan registers.

Figure 5–39. PLL Reconfiguration Scan Chain *(Note 1)*



Notes to Figure 5–39:

(1) Stratix IV left and right PLLs support C0 - C6 counters.

(2) i = 6 or i = 9.

(3) This figure shows the corresponding scan register for the K counter in between the scan registers for the charge pump and loop filter. The K counter is physically located after the VCO.

The counter settings are updated synchronously to the clock frequency of the individual counters. Therefore, all counters are not updated simultaneously.

Table 5–10 lists how these signals can be driven by the PLD logic array or I/O pins.

Table 5–10. Real-Time PLL Reconfiguration Ports

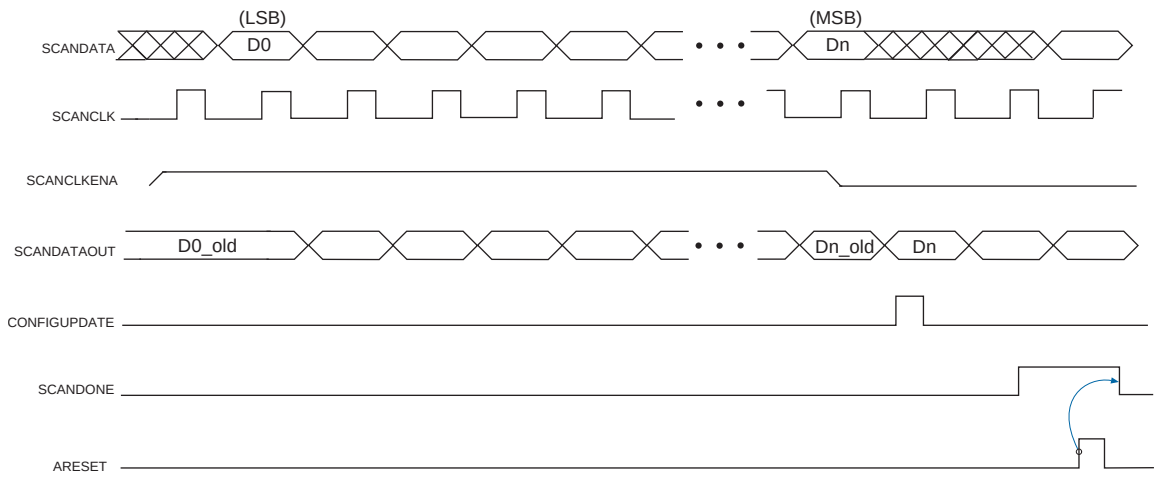
PLL Port Name	Description	Source	Destination
scandata	Serial input data stream to scan chain.	Logic array or I/O pin	PLL reconfiguration circuit
scanclk	Serial clock input signal. This clock can be free running.	GCLK, RCLK or I/O pins	PLL reconfiguration circuit
scanclkena	Enables scanclk and allows the scandata to be loaded in the scan chain. Active high.	Logic array or I/O pin	PLL reconfiguration circuit
configupdate	Writes the data in the scan chain to the PLL. Active high.	Logic array or I/O pin	PLL reconfiguration circuit
scandone	Indicates when the PLL has finished reprogramming. A rising edge indicates the PLL has begun reprogramming. A falling edge indicates the PLL has finished reprogramming.	PLL reconfiguration circuit	Logic array or I/O pins
scandataout	Used to output the contents of the scan chain.	PLL reconfiguration circuit	Logic array or I/O pins

To reconfigure the PLL counters, perform the following steps:

1. The scanclkena signal is asserted at least one scanclk cycle prior to shifting in the first bit of scandata (D0).
2. Serial data (scandata) is shifted into the scan chain on the second rising edge of scanclk.
3. After all 234 bits (top and bottom PLLs) or 180 bits (left and right PLLs) have been scanned into the scan chain, the scanclkena signal is de-asserted to prevent inadvertent shifting of bits in the scan chain.
4. The configupdate signal is asserted for one scanclk cycle to update the PLL counters with the contents of the scan chain.
5. The scandone signal goes high, indicating the PLL is being reconfigured. A falling edge indicates the PLL counters have been updated with new settings.
6. Reset the PLL using the areset signal if you make any changes to the M, N, or post-scale output C counters or to the Icp, R, or C settings.
7. You can repeat steps 1-5 to reconfigure the PLL any number of times.

Figure 5–40 shows a functional simulation of the PLL reconfiguration feature.

Figure 5–40. PLL Reconfiguration Waveform



When you reconfigure the counter clock frequency, you cannot reconfigure the corresponding counter phase shift settings using the same interface. Instead, reconfigure the phase shifts in real time using the dynamic phase shift reconfiguration interface. If you reconfigure the counter frequency, but wish to keep the same non-zero phase shift setting (for example, 90°) on the clock output, you must reconfigure the phase shift immediately after reconfiguring the counter clock frequency.

Post-Scale Counters (C0 to C9)

You can reconfigure the multiply or divide values and duty cycle of post-scale counters in real time. Each counter has an 8-bit high-time setting and an 8-bit low-time setting. The duty cycle is the ratio of output high- or low-time to the total cycle time, which is the sum of the two. Additionally, these counters have two control bits, `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.

When the `rbypass` bit is set to 1, it bypasses the counter, resulting in a divide by 1. When the `rbypass` bit is set to 0, the high- and low-time counters are added to compute the effective division of the VCO output frequency. For example, if the post-scale divide factor is 10, the high- and low-count values can be set to 5 and 5, respectively, to achieve a 50% - 50% duty cycle. The PLL implements this duty cycle by transitioning the output clock from high to low on the rising edge of the VCO output clock. However, a 4 and 6 setting for the high- and low-count values, respectively, produces an output clock with a 40% - 60% duty cycle.

The `rselodd` bit indicates an odd divide factor for the VCO output frequency along with a 50% duty cycle. For example, if the post-scale divide factor is 3, the high- and low-time count values could be set to 2 and 1, respectively, to achieve this division. This implies a 67% - 33% duty cycle. If you need a 50% - 50% duty cycle, you can set the `rselodd` control bit to 1 to achieve this duty cycle despite an odd division factor. The PLL implements this duty cycle by transitioning the output clock from high to low on a falling edge of the VCO output clock. When you set `rselodd` = 1, you subtract 0.5 cycles from the high time and you add 0.5 cycles to the low time. For example:

- High-time count = 2 cycles
- Low-time count = 1 cycle
- `rselodd` = 1 effectively equals:
  - High-time count = 1.5 cycles
  - Low-time count = 1.5 cycles
  - Duty cycle = (1.5/3) % high-time count and (1.5/3) % low-time count

Scan Chain Description

The length of the scan chain varies for different Stratix IV PLLs. The top and bottom PLLs have ten post-scale counters and a 234-bit scan chain, while the left and right PLLs have seven post-scale counters and a 180-bit scan chain. Table 5–11 lists the number of bits for each component of a Stratix IV PLL.

Table 5–11. Top and Bottom PLL Reprogramming Bits (Part 1 of 2)

Block Name	Number of Bits		Total
	Counter	Other *(1)*	
C9 *(2)*	16	2	18
C8	16	2	18
C7	16	2	18
C6 *(3)*	16	2	18
C5	16	2	18
C4	16	2	18
C3	16	2	18
C2	16	2	18
C1	16	2	18
C0	16	2	18
M	16	2	18
N	16	2	18
Charge Pump Current	0	3	3
VCO Post-Scale divider (K)	1	0	1
Loop Filter Capacitor *(4)*	0	2	2
Loop Filter Resistor	0	5	5
Unused CP/LF	0	7	7

Table 5–11. Top and Bottom PLL Reprogramming Bits (Part 2 of 2)

Block Name	Number of Bits		Total
	Counter	Other *(1)*	
Total number of bits	—	—	234

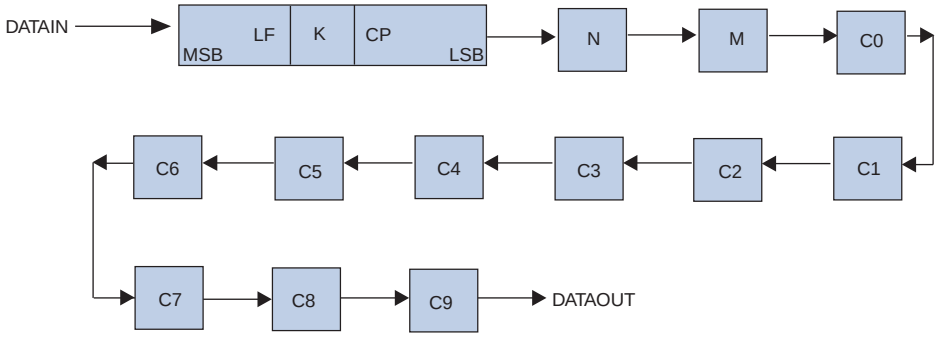
Notes to Table 5–11:

(1) Includes two control bits, `rbypass`, for bypassing the counter, and `rselodd`, to select the output clock duty cycle.
(2) The LSB for the C9 low-count value is the first bit shifted into the scan chain for the top and bottom PLLs.
(3) The LSB for the C6 low-count value is the first bit shifted into the scan chain for the left and right PLLs.
(4) The MSB for the loop filter is the last bit shifted into the scan chain.

Table 5–11 lists the scan chain order of PLL components for the top and bottom PLLs, which have 10 post-scale counters. The order of bits is the same for the left and right PLLs, but the reconfiguration bits start with the C6 post-scale counter.

Figure 5–41 shows the scan-chain order of PLL components for the top and bottom PLLs.

Figure 5–41. Scan-Chain Order of PLL Components for Top and Bottom PLLs *(Note 1)*

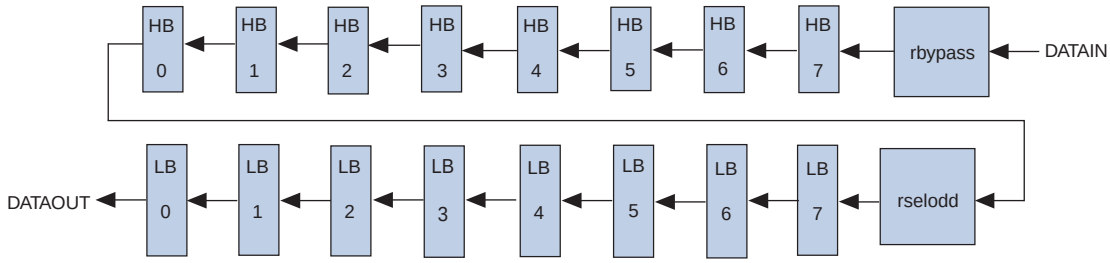


Note to Figure 5–41:

(1) Left and right PLLs have the same scan-chain order. The post-scale counters end at C6.

Figure 5–42 shows the scan-chain bit-order sequence for post-scale counters in all Stratix IV PLLs.

Figure 5–42. Scan-Chain Bit-Order Sequence for Post-Scale Counters in Stratix IV PLLs



Charge Pump and Loop Filter

You can reconfigure the charge-pump and loop-filter settings to update the PLL bandwidth in real time.

Table 5–12 lists the possible settings for charge pump current (Icp) values for Stratix IV PLLs.

Table 5–12. Charge Pump Current Bit Settings

CP[2]	CP[1]	CP[0]	Decimal Value for Setting
0	0	0	0
0	0	1	1
0	1	1	3
1	1	1	7

Table 5–13 lists the possible settings for loop-filter resistor (R) values for Stratix IV PLLs.

Table 5–13. Loop-Filter Resistor Bit Settings

LFR[4]	LFR[3]	LFR[2]	LFR[1]	LFR[0]	Decimal Value for Setting
0	0	0	0	0	0
0	0	0	1	1	3
0	0	1	0	0	4
0	1	0	0	0	8
1	0	0	0	0	16
1	0	0	1	1	19
1	0	1	0	0	20
1	1	0	0	0	24
1	1	0	1	1	27
1	1	1	0	0	28
1	1	1	1	0	30

Table 5–14 lists the possible settings for loop-filter capacitor (C) values for Stratix IV PLLs.

Table 5–14. Loop-Filter Capacitor Bit Settings

LFC[1]	LFC[0]	Decimal Value for Setting
0	0	0
0	1	1
1	1	3

Bypassing PLL

Bypassing a PLL counter results in a multiply (m counter) or a divide (n and C0 to C9 counters) factor of one.

Table 5–15 lists the settings for bypassing the counters in Stratix IV PLLs.

Table 5–15. PLL Counter Settings

PLL Scan Chain Bits [0..8] Settings									
LSB *(2)*								**MSB** *(1)*	**Description**
X	X	X	X	X	X	X	X	1 *(3)*	PLL counter bypassed
X	X	X	X	X	X	X	X	0 *(3)*	PLL counter not bypassed because bit 8 (MSB) is set to 0

Notes to Table 5–15:

(1) Most significant bit (MSB).
(2) Least significant bit (LSB).
(3) Counter-bypass bit.

To bypass any of the PLL counters, set the bypass bit to **1**. The values on the other bits are ignored. To bypass the VCO post-scale counter (K), set the corresponding bit to 0.

Dynamic Phase-Shifting

The dynamic phase-shifting feature allows the output phases of individual PLL outputs to be dynamically adjusted relative to each other and to the reference clock, without the need to send serial data through the scan chain of the corresponding PLL. This feature simplifies the interface and allows you to quickly adjust the clock-to-out (t_{CO}) delays by changing the output clock phase-shift in real time. This adjustment is achieved by incrementing or decrementing the VCO phase-tap selection to a given C counter or to the M counter. The phase is shifted by 1/8 of the VCO frequency at a time. The output clocks are active during this phase-reconfiguration process.

Table 5–16 lists the control signals that are used for dynamic phase-shifting.

Table 5–16. Dynamic Phase-Shifting Control Signals (Part 1 of 2)

Signal Name	Description	Source	Destination
PHASECOUNTERSELECT[3..0]	Counter select. Four bits decoded to select either the M or one of the C counters for phase adjustment. One address maps to select all C counters. This signal is registered in the PLL on the rising edge of SCANCLK.	Logic array or I/O pins	PLL reconfiguration circuit
PHASEUPDOWN	Selects dynamic phase shift direction; 1 = UP; 0 = DOWN. Signal is registered in the PLL on the rising edge of SCANCLK.	Logic array or I/O pin	PLL reconfiguration circuit
PHASESTEP	Logic high enables dynamic phase shifting.	Logic array or I/O pin	PLL reconfiguration circuit

Table 5–16. Dynamic Phase-Shifting Control Signals (Part 2 of 2)

Signal Name	Description	Source	Destination
SCANCLK	Free running clock from the core used in combination with PHASESTEP to enable and disable dynamic phase shifting. Shared with SCANCLK for dynamic reconfiguration.	GCLK, RCLK or I/O pin	PLL reconfiguration circuit
PHASEDONE	When asserted, this indicates to core-logic that the phase adjustment is complete and the PLL is ready to act on a possible second adjustment pulse. Asserts based on internal PLL timing. De-asserts on the rising edge of SCANCLK.	PLL reconfiguration circuit	Logic array or I/O pins

Table 5–17 lists the PLL counter selection based on the corresponding PHASECOUNTERSELECT setting.

Table 5–17. Phase Counter Select Mapping

PHASECOUNTERSELECT[3]	[2]	[1]	[0]	Selects
0	0	0	0	All Output Counters
0	0	0	1	M Counter
0	0	1	0	C0 Counter
0	0	1	1	C1 Counter
0	1	0	0	C2 Counter
0	1	0	1	C3 Counter
0	1	1	0	C4 Counter
0	1	1	1	C5 Counter
1	0	0	0	C6 Counter
1	0	0	1	C7 Counter
1	0	1	0	C8 Counter
1	0	1	1	C9 Counter

To perform one dynamic phase-shift, follow these steps:

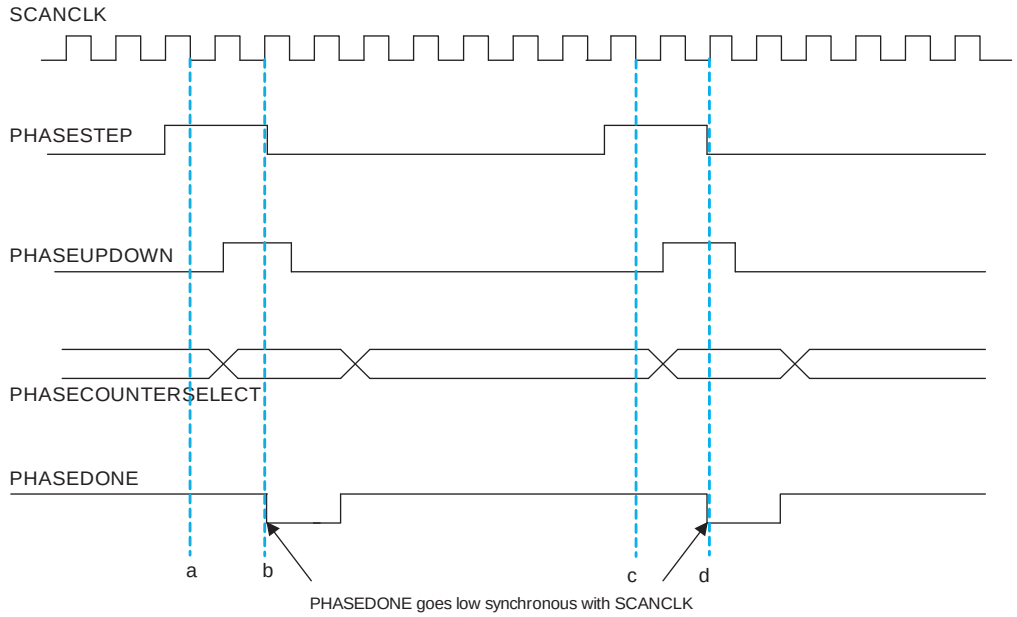
1. Set PHASEUPDOWN and PHASECOUNTERSELECT as required.
2. Assert PHASESTEP. Each PHASESTEP pulse enables one phase shift. The PHASESTEP pulses must be at least one scanclk cycle apart.
3. Wait for PHASEDONE to go low.
4. De-assert PHASESTEP.
5. Wait for PHASEDONE to go high.
6. Repeat steps 1-5 as many times as required to perform multiple phase-shifts.

All signals are synchronous to SCANCLK and are latched on the SCANCLK edges and must meet tsu/th requirements with respect to SCANCLK edges.

You can repeat dynamic phase-shifting indefinitely. For example, in a design where the VCO frequency is set to 1000 MHz and the output clock frequency is 100 MHz, performing 40 dynamic phase shifts (each one yields 125 ps phase shift) results in shifting the output clock by 180°, which is a phase shift of 5 ns.

The PHASESTEP signal is latched on the negative edge of SCANCLK. In Figure 5–43, this is shown by the second SCANCLK falling edge. PHASESTEP must stay high for at least two SCANCLK cycles. On the second SCANCLK rising edge after PHASESTEP is latched (the fourth SCANCLK rising edge in Figure 5–43), the values of PHASEUPDOWN and PHASECOUNTERSELECT are latched and the PLL starts dynamic phase-shifting for the specified counter(s) and in the indicated direction. On the fourth SCANCLK rising edge, PHASEDONE goes from high to low and remains low until the PLL finishes dynamic phase-shifting. You can perform another dynamic phase shift after the PHASEDONE signal goes from low to high.

Figure 5–43. Dynamic Phase Shifting Waveform



Depending on the VCO and SCANCLK frequencies, PHASEDONE low time may be greater than or less than one SCANCLK cycle. The maximum time for reconfiguring phase shift dynamically is to be determined based on device characterization.

After PHASEDONE goes from low to high, you can perform another dynamic phase shift.

For information about the ALTPLL_RECONFIG MegaWizard Plug-In Manager, refer to the *ALTPLL_RECONFIG Megafunction User Guide*.

PLL Specifications

For information about PLL timing specifications, refer to the *DC and Switching Characteristics of Stratix IV Devices* chapter.

Chapter Revision History

Table 5–18 lists the revision history for this chapter.

Table 5–18. Chapter Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated Table 5–1 and Table 5–7. ■ Updated "Clock Networks in Stratix IV Devices", "Periphery Clock Networks", and "Cascading PLLs" sections. ■ Added Figure 5–5, Figure 5–6, Figure 5–7, Figure 5–8, and Figure 5–9. ■ Added "Clock Sources Per Region" section. ■ Updated Figure 5–40. ■ Removed EP4SE110, EP4SE290, and EP4SE680 devices. ■ Added EP4S40G2, EP4S100G2, EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5, and EP4SE820 devices.	—
June 2009 v2.3	■ Updated Table 5–7. ■ Updated the "PLL Reconfiguration Hardware Implementation" and "Zero-Delay Buffer Mode" sections. ■ Added introductory sentences to improve search ability. ■ Removed the Conclusion section. ■ Minor text edits.	—
April 2009 v2.2	■ Updated Table 5–1 and Table 5–7. ■ Updated Figure 5–3 and Figure 5–4. ■ Updated the "Periphery Clock Networks" section.	—
March 2009 v2.1	■ Updated Table 5–7. ■ Updated Figure 5–34. ■ Updated "Guidelines" section. ■ Removed "Referenced Documents" section.	—

Table 5–18. Chapter Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
November 2008 v2.0	■ Updated Table 5–7. ■ Updated Note 1 of Figure 5–10. ■ Updated Figure 5–15. ■ Updated Figure 5–20. ■ Added Figure 5–21. ■ Made minor editorial changes.	—
May 2008 v1.0	Initial Release.	—

Section II. I/O Interfaces

This section provides information on Stratix® IV device I/O features, external memory interfaces, and high-speed differential interfaces with DPA. This section includes the following chapters:

- Chapter 6, I/O Features in Stratix IV Devices
- Chapter 7, External Memory Interfaces in Stratix IV Devices
- Chapter 8, High-Speed Differential I/O Interfaces and DPA in Stratix IV Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

6. I/O Features in Stratix IV Devices

SIV51006-3.0

This chapter describes how Stratix® IV devices provide I/O capabilities that allow you to work in compliance with current and emerging I/O standards and requirements. With these device features, you can reduce board design interface costs and increase development flexibility.

Altera® Stratix IV FPGAs deliver a breakthrough level of system bandwidth and power efficiency for high-end applications, allowing you to innovate without compromise. Stratix IV I/Os are specifically designed for ease-of-use and rapid system integration while simultaneously providing the high bandwidth required to maximize internal logic capabilities and produce system-level performance.

Stratix IV device I/O capability far exceeds the I/O bandwidth available from previous generation FPGAs. Independent modular I/O banks with a common bank structure for vertical migration lend efficiency and flexibility to the high-speed I/O.

Package and die enhancements with dynamic termination and output control provide best-in-class signal integrity. Numerous I/O features assist high-speed data transfer into and out of the device, including:

- Up to 32 full-duplex clock data recovery (CDR)-based transceivers supporting data rates between 600 Mbps and 8.5 Gbps
- Dedicated circuitry to support physical layer functionality for popular serial protocols, such as PCI Express (PIPE) Gen1 and Gen2, Gigabit Ethernet, Serial RapidIO, SONET/SDH, XAUI/HiGig, (OIF) CEI-6G, SD/HD/3G-SDI, Fibre Channel, SFI-5, and Interlaken
- Complete PCI Express (PIPE) protocol solution with embedded PCI Express hard IP blocks that implement PHY-MAC layer, data link layer, and transaction layer functionality
- Single-ended, non-voltage-referenced, and voltage-referenced I/O standards
- Low-voltage differential signaling (LVDS), reduced swing differential signaling (RSDS), mini-LVDS, high-speed transceiver logic (HSTL), and SSTL
- Single data rate (SDR) and half data rate (HDR—half frequency and twice data width of SDR) input and output options
- Up to 132 full duplex 1.6 Gbps true LVDS channels (132 Tx + 132 Rx) on the row I/O banks
- Hard dynamic phase alignment (DPA) block with serializer/deserializer (SERDES)
- Deskew, read and write leveling, and clock-domain crossing functionality
- Programmable output current strength
- Programmable slew rate
- Programmable delay
- Programmable bus-hold circuit
- Programmable pull-up resistor

- Open-drain output
- Serial, parallel, and dynamic on-chip termination (OCT)
- Differential OCT
- Programmable pre-emphasis
- Programmable equalization
- Programmable differential output voltage (V_{OD})

This chapter contains the following sections:

I/O Standards Support

Stratix IV devices support a wide range of industry I/O standards. Table 6–1 lists the I/O standards Stratix IV devices support, as well as the typical applications. These devices support V_{CCIO} voltage levels of 3.0, 2.5, 1.8, 1.5, and 1.2 V.

Table 6–1. Stratix IV I/O Standards and Applications (Part 1 of 2)

I/O Standard	Application
3.3-V LVTTL/LVCMOS *(1)*, *(2)*	General purpose
2.5-V LVCMOS	General purpose
1.8-V LVCMOS	General purpose
1.5-V LVCMOS	General purpose
1.2-V LVCMOS	General purpose
3.0-V PCI/PCI-X	PC and embedded system
SSTL-2 Class I and II	DDR SDRAM
SSTL-18 Class I and II	DDR2 SDRAM
SSTL-15 Class I and II	DDR3 SDRAM
HSTL-18 Class I and II	QDRII/RLDRAM II
HSTL-15 Class I and II	QDRII/QDRII+/RLDRAM II
HSTL-12 Class I and II	General purpose
Differential SSTL-2 Class I and II	DDR SDRAM
Differential SSTL-18 Class I and II	DDR2 SDRAM
Differential SSTL-15 Class I and II	DDR3 SDRAM
Differential HSTL-18 Class I and II	Clock interfaces
Differential HSTL-15 Class I and II	Clock interfaces

Table 6–1. Stratix IV I/O Standards and Applications (Part 2 of 2)

I/O Standard	Application
Differential HSTL-12 Class I and II	Clock interfaces
LVDS	High-speed communications
RSDS	Flat panel display
mini-LVDS	Flat panel display
LVPECL	Video graphics and clock distribution

Notes to Table 6–1:

(1) The 3.3-V LVTTL/LVCMOS standard is supported using V_{CCIO} at 3.0 V.

(2) For more information about the 3.3-V LVTTL/LVCMOS standard supported in Stratix IV devices, refer to "3.3-V I/O Interface" on page 6–19.

For more information about transceiver supported I/O standards, refer to the *Stratix IV Transceiver Architecture* chapter.

I/O Standards and Voltage Levels

Stratix IV devices support a wide range of industry I/O standards, including single-ended, voltage-referenced single-ended, and differential I/O standards.

Table 6–2 lists the supported I/O standards and typical values for input and output V_{CCIO}, V_{CCPD}, V_{REF}, and board V_{TT}.

Table 6–2. Stratix IV I/O Standards and Voltage Levels *(Note 1)*, *(2)* (Part 1 of 3)

I/O Standard	Standard Support	V_{CCIO} (V)				V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation		Output Operation				
		Column I/O Banks	Row I/O Banks	Column I/O Banks	Row I/O Banks			
3.3-V LVTTL	JESD8-B	3.0/2.5	3.0/2.5	3.0	3.0	3.0	—	—
3.3-V LVCMOS *(3)*	JESD8-B	3.0/2.5	3.0/2.5	3.0	3.0	3.0	—	—
2.5-V LVCMOS	JESD8-5	3.0/2.5	3.0/2.5	2.5	2.5	2.5	—	—
1.8-V LVCMOS	JESD8-7	1.8/1.5	1.8/1.5	1.8	1.8	2.5	—	—
1.5-V LVCMOS	JESD8-11	1.8/1.5	1.8/1.5	1.5	1.5	2.5	—	—
1.2-V LVCMOS	JESD8-12	1.2	1.2	1.2	1.2	2.5	—	—
3.0-V PCI	PCI Rev 2.1	3.0	3.0	3.0	3.0	3.0	—	—
3.0-V PCI-X	PCI-X Rev 1.0	3.0	3.0	3.0	3.0	3.0	—	—
SSTL-2 Class I	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	1.25	1.25
SSTL-2 Class II	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	1.25	1.25
SSTL-18 Class I	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90
SSTL-18 Class II	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90
SSTL-15 Class I	—	*(2)*	*(2)*	1.5	1.5	2.5	0.75	0.75
SSTL-15 Class II	—	*(2)*	*(2)*	1.5	—	2.5	0.75	0.75
HSTL-18 Class I	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90
HSTL-18 Class II	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	0.90	0.90

Table 6–2. Stratix IV I/O Standards and Voltage Levels *(Note 1)*, *(2)* (Part 2 of 3)

I/O Standard	Standard Support	V_{CCIO} (V)				V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation		Output Operation				
		Column I/O Banks	Row I/O Banks	Column I/O Banks	Row I/O Banks			
HSTL-15 Class I	JESD8-6	*(2)*	*(2)*	1.5	1.5	2.5	0.75	0.75
HSTL-15 Class II	JESD8-6	*(2)*	*(2)*	1.5	—	2.5	0.75	0.75
HSTL-12 Class I	JESD8-16A	*(2)*	*(2)*	1.2	1.2	2.5	0.6	0.6
HSTL-12 Class II	JESD8-16A	*(2)*	*(2)*	1.2	—	2.5	0.6	0.6
Differential SSTL-2 Class I	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	—	1.25
Differential SSTL-2 Class II	JESD8-9B	*(2)*	*(2)*	2.5	2.5	2.5	—	1.25
Differential SSTL-18 Class I	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential SSTL-18 Class II	JESD8-15	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential SSTL-15 Class I	—	*(2)*	*(2)*	1.5	1.5	2.5	—	0.75
Differential SSTL-15 Class II	—	*(2)*	*(2)*	1.5	—	2.5	—	0.75
Differential HSTL-18 Class I	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential HSTL-18 Class II	JESD8-6	*(2)*	*(2)*	1.8	1.8	2.5	—	0.90
Differential HSTL-15 Class I	JESD8-6	*(2)*	*(2)*	1.5	1.5	2.5	—	0.75
Differential HSTL-15 Class II	JESD8-6	*(2)*	*(2)*	1.5	—	2.5	—	0.75
Differential HSTL-12 Class I	JESD8-16A	*(2)*	*(2)*	1.2	1.2	2.5	—	0.60
Differential HSTL-12 Class II	JESD8-16A	*(2)*	*(2)*	1.2	—	2.5	—	0.60
LVDS *(4)*, *(5)*	ANSI/TIA/EIA-644	*(2)*	*(2)*	2.5	2.5	2.5	—	—
RSDS *(6)*, *(7)*	—	*(2)*	*(2)*	2.5	2.5	2.5	—	—
mini-LVDS *(6)*, *(7)*	—	*(2)*	*(2)*	2.5	2.5	2.5	—	—

Table 6–2. Stratix IV I/O Standards and Voltage Levels *(Note 1)*, *(2)* (Part 3 of 3)

I/O Standard	Standard Support	V_{CCIO} (V)				V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
		Input Operation		Output Operation				
		Column I/O Banks	Row I/O Banks	Column I/O Banks	Row I/O Banks			
LVPECL	—	*(4)*	2.5	—	—	2.5	—	—

Notes to Table 6–2:

(1) V_{CCPD} is either 2.5 or 3.0 V. For V_{CCIO} = 3.0 V, V_{CCPD} = 3.0 V. For V_{CCIO} = 2.5 V or less, V_{CCPD} = 2.5 V.

(2) Single-ended HSTL/SSTL, differential SSTL/HSTL, and LVDS input buffers are powered by V_{CCPD}. Row I/O banks support both true differential input buffers and true differential output buffers. Column I/O banks support true differential input buffers, but not true differential output buffers. I/O pins are organized in pairs to support differential standards. Column I/O differential HSTL and SSTL inputs use LVDS differential input buffers without on-chip R_D support.

(3) For more information about the 3.3-V LVTTL/LVCMOS standard supported in Stratix IV devices, refer to "3.3-V I/O Interface" on page 6–19.

(4) Column I/O banks support LVPECL I/O standards for input clock operation. Clock inputs on column I/O are powered by $V_{CCCLKIN}$ when configured as differential clock inputs. They are powered by V_{CCIO} when configured as single-ended clock inputs. Differential clock inputs in row I/O are powered by V_{CCPD}.

(5) Column and row I/O banks support LVDS outputs using two single-ended output buffers, an external one-resistor (LVDS_E_1R), and a three-resistor (LVDS_E_3R) network.

(6) Row I/O banks support RSDS and mini-LVDS I/O standards using true LVDS output buffer without a resistor network.

(7) Column and row I/O banks support RSDS and mini-LVDS I/O standards using two single-ended output buffers with one-resistor (RSDS_E_1R and mini-LVDS_E_1R) and three-resistor (RSDS_E_3R and mini-LVDS_E_3R) networks.

For more information about electrical characteristics of each I/O standard, refer to the *DC and Switching Characteristics* chapter.

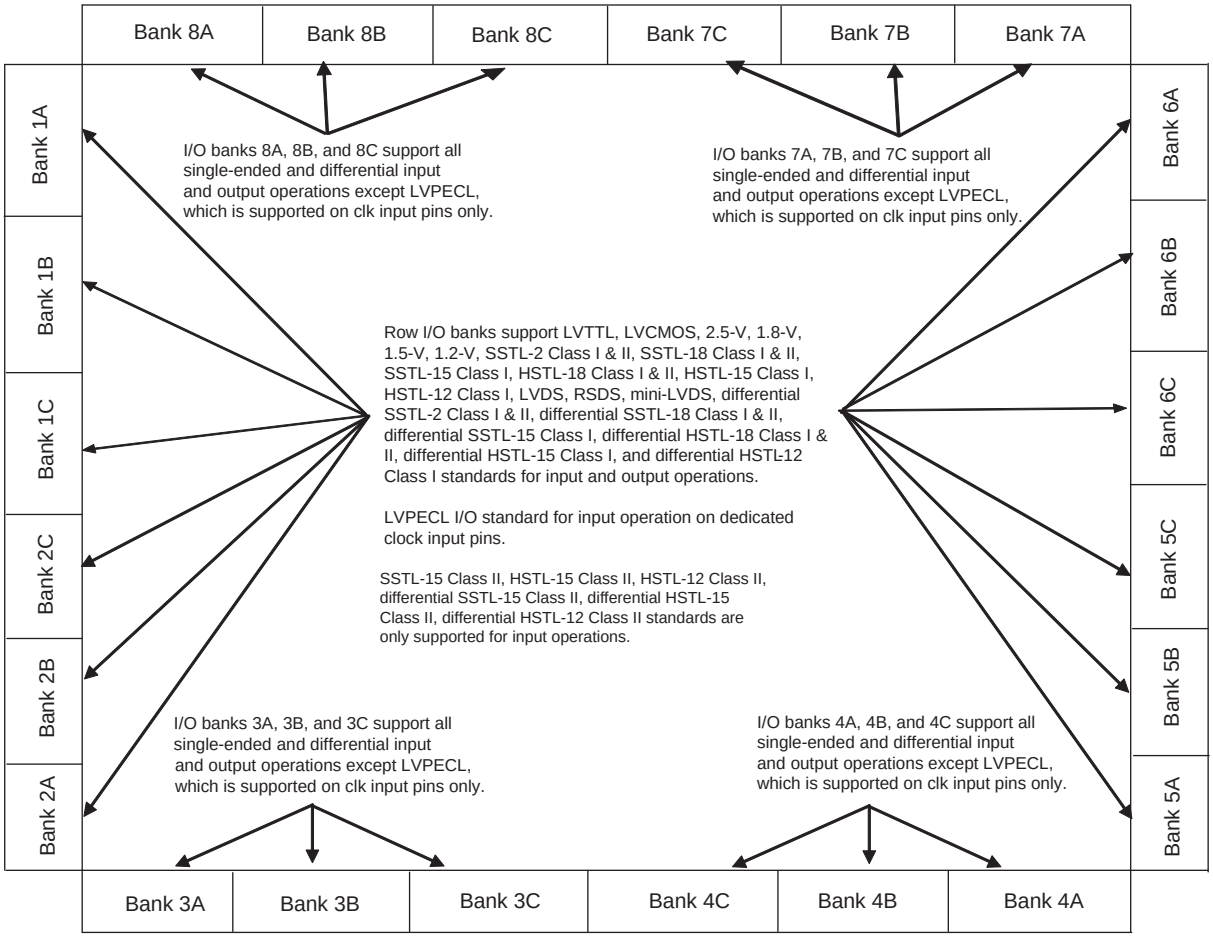
I/O Banks

Stratix IV devices contain up to 24 I/O banks, as shown in Figure 6–1 and Figure 6–2. The row I/O banks contain true differential input and output buffers and dedicated circuitry to support differential standards at speeds up to 1.6 Gbps.

Each I/O bank in Stratix IV devices can support high-performance external memory interfaces with dedicated circuitry. The I/O pins are organized in pairs to support differential standards. Each I/O pin pair can support both differential input and output buffers. The only exceptions are the `clk[1,3,8,10]`, `PLL_L[1,4]_clk`, and `PLL_R[1,4]_clk` pins, which support differential input operations only.

For the number of channels available for the LVDS I/O standard, refer to the *High-Speed Differential I/O Interface with DPA* chapter. For more information about transceiver-bank-related features, refer to the *Stratix IV Transceiver Architecture* chapter.

Figure 6–1. Stratix IV E Devices I/O Banks *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*, *(6)*, *(7)*, *(8)*



Notes to Figure 6–1:

(1) Differential HSTL and SSTL outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted.

(2) Column I/O differential HSTL and SSTL inputs use LVDS differential input buffers without differential OCT support.

(3) Column I/O supports LVDS outputs using single-ended buffers and external resistor networks.

(4) Column I/O supports PCI/PCI-X with on-chip clamp diode. Row I/O supports PCI/PCI-X with external clamp diode.

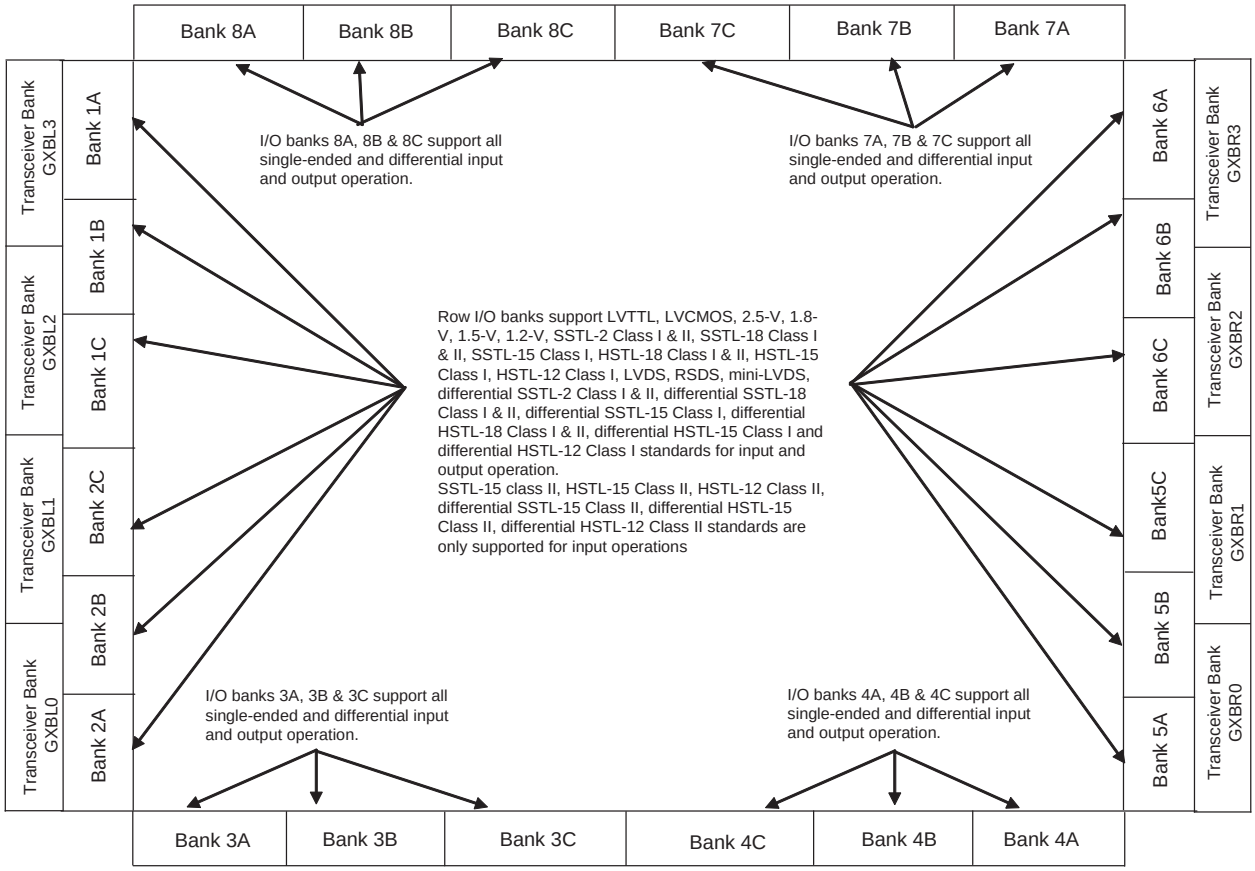
(5) Clock inputs on column I/O are powered by $V_{CCCLKIN}$ when configured as differential clock inputs. They are powered by V_{CCIO} when configured as single-ended clock inputs. All outputs use the corresponding bank V_{CCIO}.

(6) Row I/O supports the true LVDS output buffer.

(7) Column and row I/O banks support LVPECL standards for input clock operation.

(8) Figure 6–1 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

Figure 6–2. Stratix IV GX Devices I/O Banks *(Note 1), (2), (3), (4), (5), (6), (7), (8)*



Notes to Figure 6–2:

(1) Differential HSTL and SSTL outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted.

(2) Column I/O differential HSTL and SSTL inputs use LVDS differential input buffers without differential OCT support.

(3) Column I/O supports LVDS outputs using single-ended buffers and external resistor networks.

(4) Column I/O supports PCI/PCI-X with on-chip clamp diode. Row I/O supports PCI/PCI-X with external clamp diode.

(5) Clock inputs on column I/O are powered by $V_{CCCLKIN}$ when configured as differential clock inputs. They are powered by V_{CCIO} when configured as single-ended clock inputs. All outputs use the corresponding bank V_{CCIO}.

(6) Row I/O supports the true LVDS output buffer.

(7) Column and row I/O banks support LVPECL standards for input clock operation.

(8) Figure 6–2 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only.

Modular I/O Banks

The I/O pins in Stratix IV devices are arranged in groups called modular I/O banks. Depending on device densities, the number of Stratix IV device I/O banks range from 16 to 24. The number of I/O pins on each bank is 24, 32, 36, 40, or 48. Figure 6–4 through Figure 6–16 show the number of I/O pins available in each I/O bank.

In Stratix IV devices, the maximum number of I/O banks per side is either four or six, depending on the device density. When migrating between devices with a different number of I/O banks per side, it is the middle or "B" bank that is removed or inserted. For example, when moving from a 24-bank device to a 16-bank device, the banks that are dropped are "B" banks, namely: 1B, 2B, 3B, 4B, 5B, 6B, 7B, and 8B. Similarly, when moving from a 16-bank device to a 24-bank device, the banks that are added are the same "B" banks.

After migration from a smaller device to a larger device, the bank size increases or remains the same, but never decreases. For example, the number of I/O pins to a bank may increase from 24 to 26, 32, 36, 40, 42, or 48, but will never decrease. This is shown in Figure 6–3.

Figure 6–3. Bank Migration Path with Increasing Device Size

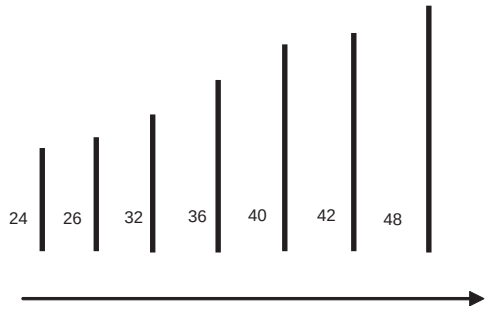


Figure 6–4 through Figure 6–16 show the number of I/O pins and packaging information for different sets of available devices. Figure 6–4 through Figure 6–16 shows the top view of the silicon die that corresponds to a reverse view for flip chip packages. They are graphical representations only.

For Figure 6–4 through Figure 6–16, the pin count includes all general purpose I/Os, dedicated clock pins, and dual purpose configuration pins. Transceiver pins and dedicated configuration pins are not included in the pin count.

Figure 6–4. Number of I/Os in Each Bank in EP4SE230 and EP4SE360 Devices in the 780-Pin FineLine BGA Package

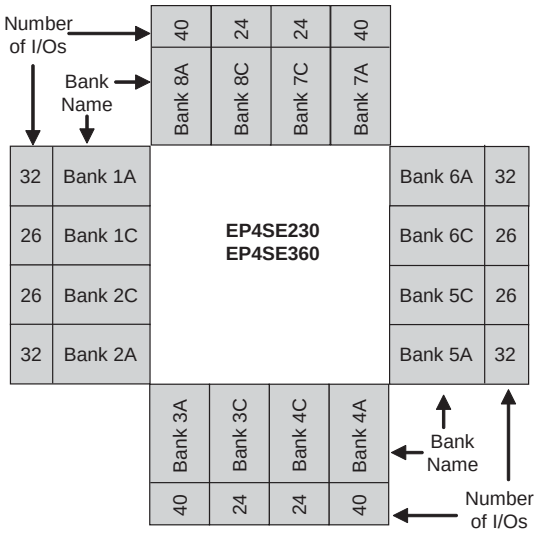


Figure 6–5. Number of I/Os in Each Bank in EP4SE360, EP4SE530, and EP4SE820 Devices in the 1152-Pin FineLine BGA Package

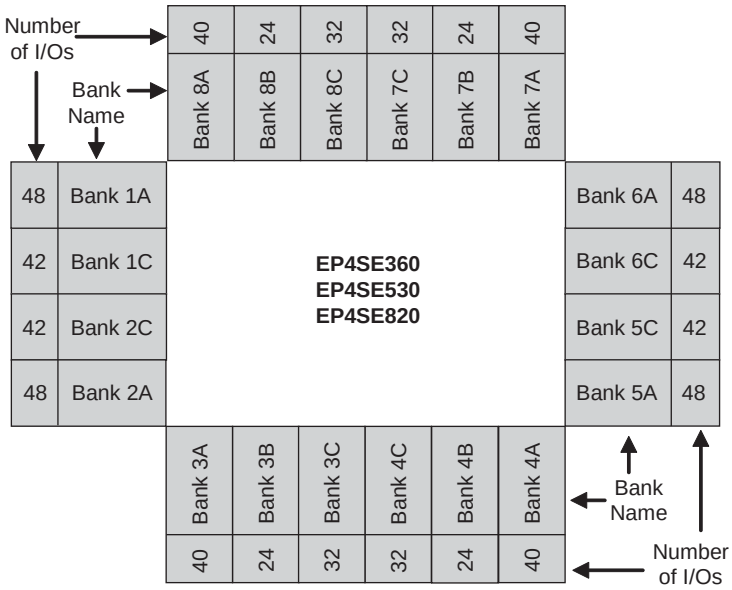


Figure 6–6. Number of I/Os in Each Bank in EP4SE530 and EP4SE820 Devices in the 1517-Pin FineLine BGA Package

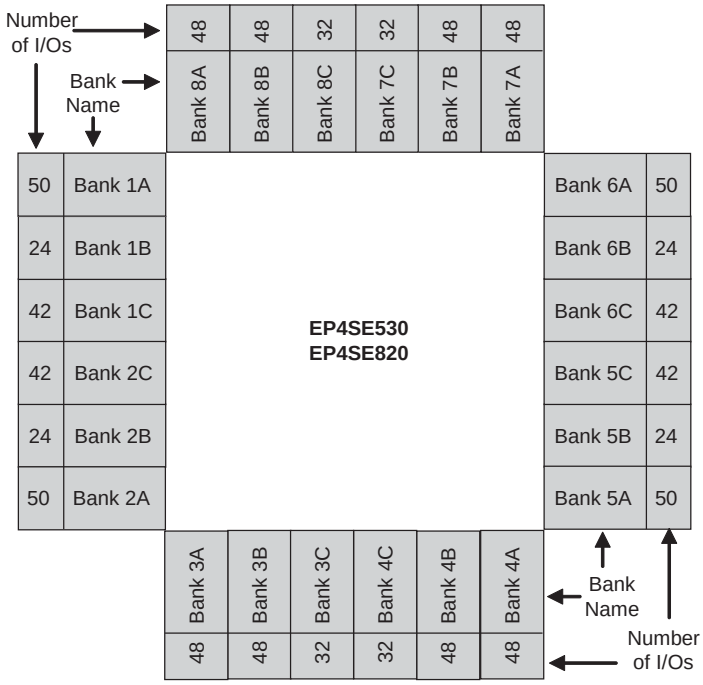


Figure 6–7. Number of I/Os in Each Bank in EP4SE530 and EP4SE820 Devices in the 1760-Pin Fineline BGA Package

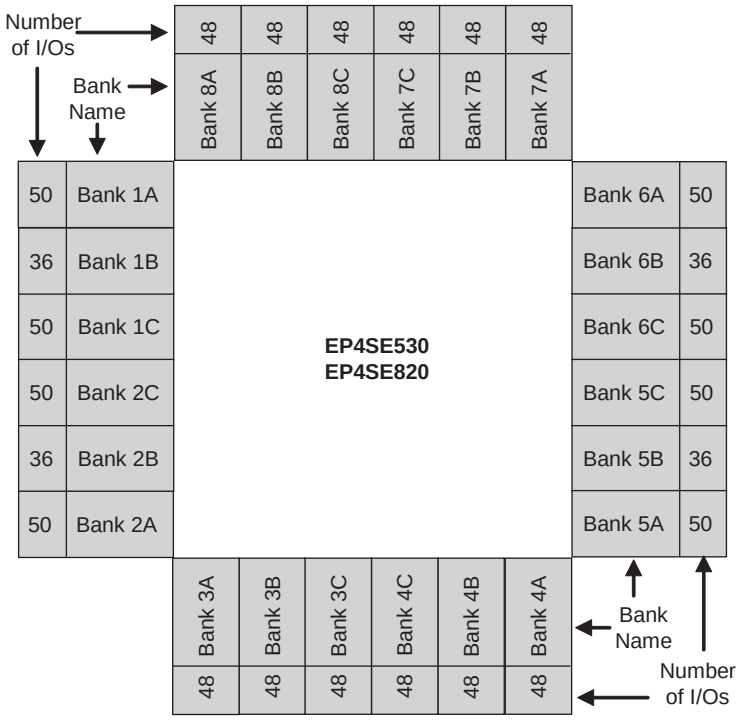


Figure 6–8. Number of I/Os in Each Bank in EP4SGX70, EP4SGX110, EP4SGX180, and EP4SGX230 Devices in the 780-Pin FineLine BGA Package

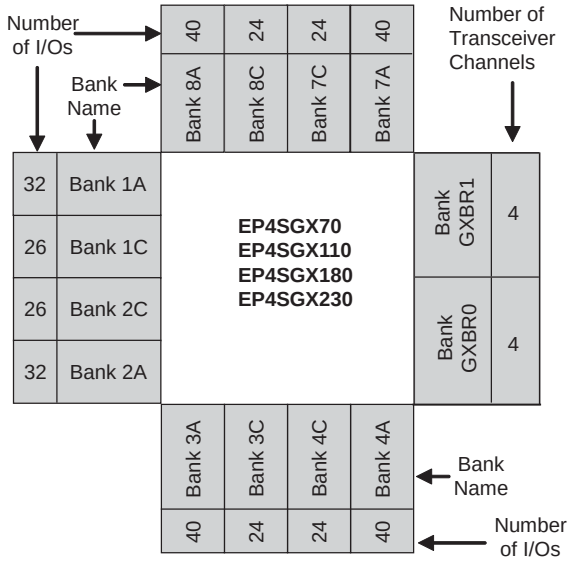


Figure 6–9. Number of I/Os in Each Bank in EP4SGX290 and EP4SGX360 Devices in the 780-Pin FineLine BGA Package

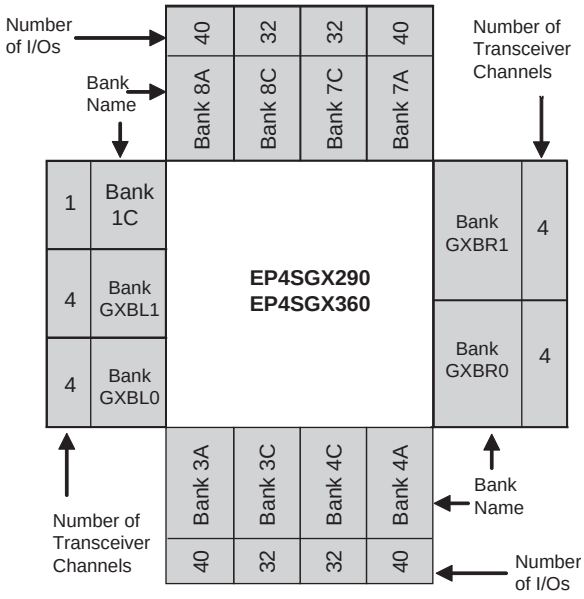


Figure 6–10. Number of I/Os in Each Bank in EP4SGX70 and EP4SGX110 Devices in the 1152-Pin FineLine BGA Package

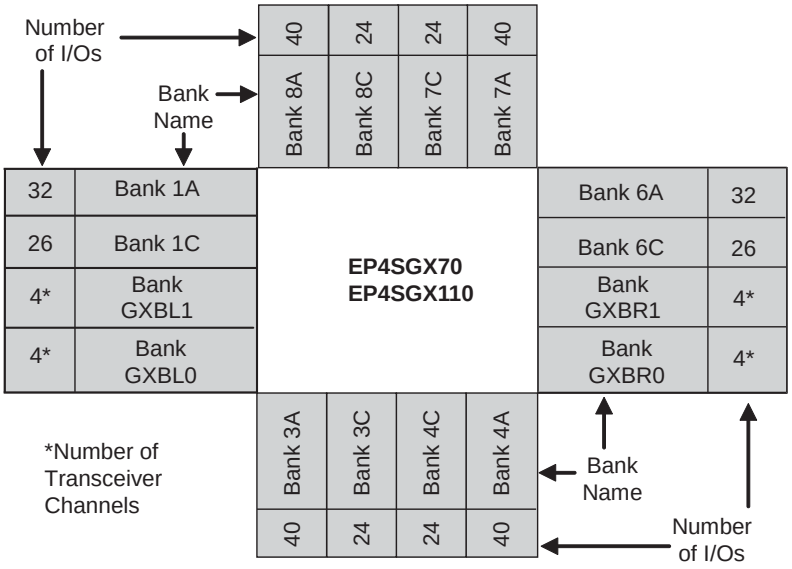
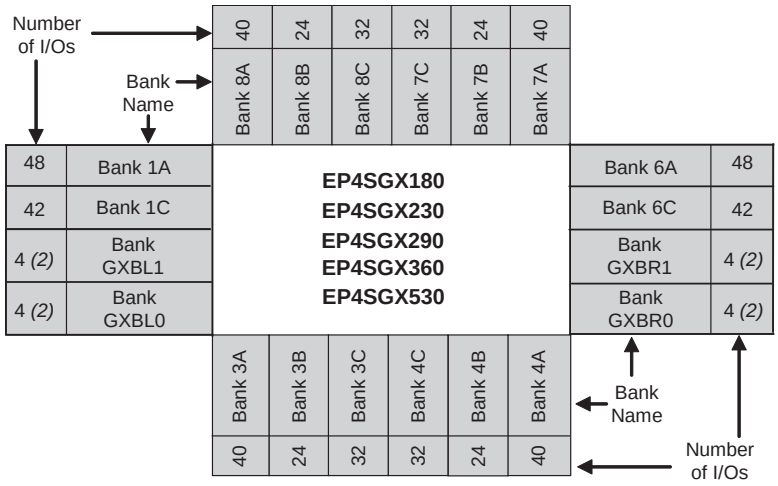


Figure 6–11. Number of I/Os in Each Bank in EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1152-Pin FineLine BGA Package *(Note 1)*, *(2)*

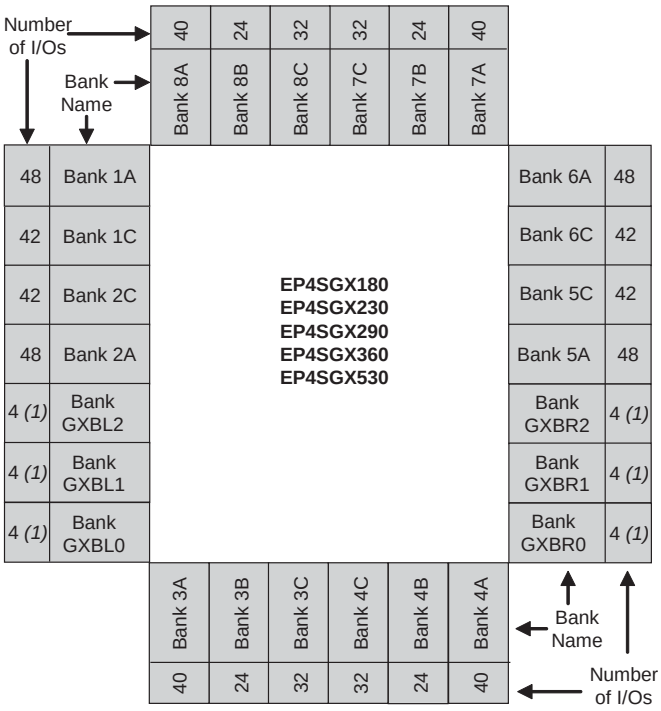


Notes to Figure 6–11:

(1) Except for the EP4SGX530 device, all listed devices have two variants in the F1152 package option—one with no PMA-only transceiver channels and the other with two PMA-only transceiver channels for each transceiver bank. The EP4SGX530 device is only offered with two PMA-only transceiver channels for each transceiver bank in the F1152 package option.

(2) There are two additional PMA-only transceiver channels in each transceiver bank for devices with the PMA-only transceiver package option.

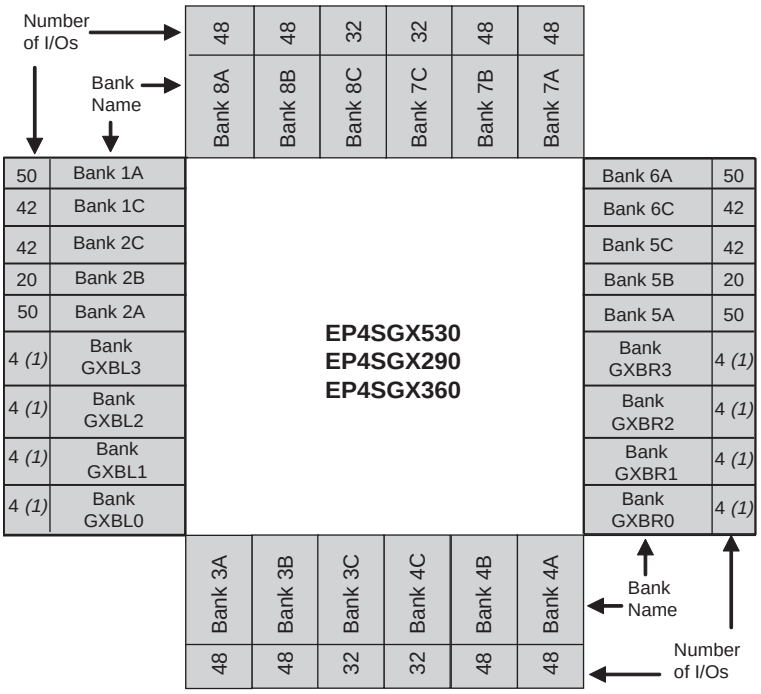
Figure 6–12. Number of I/Os in Each Bank in EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1517-Pin FineLine BGA Package *(Note 1)*



Note to Figure 6–12:

(1) There are two additional PMA-only transceiver channels in each transceiver bank.

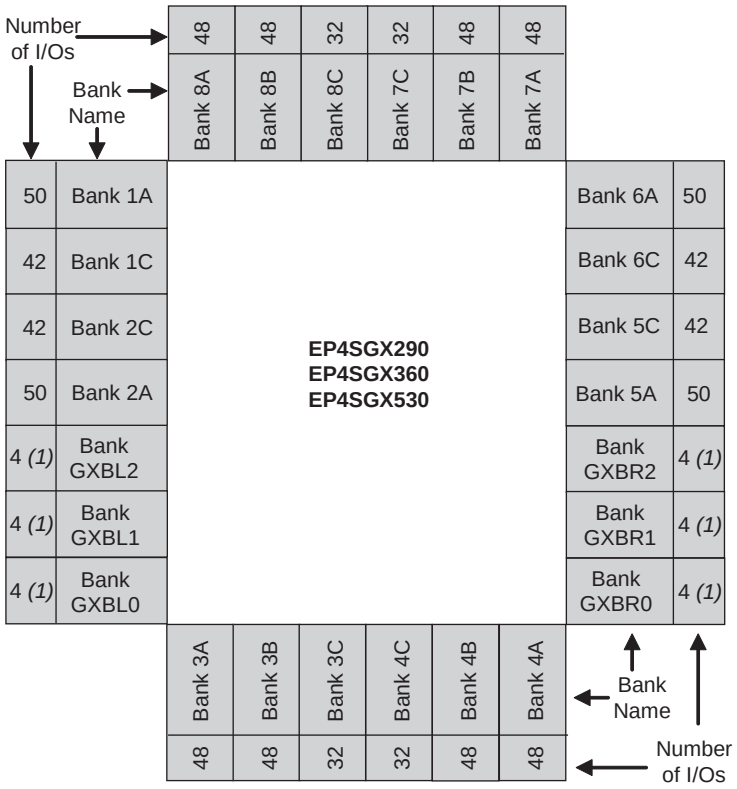
Figure 6–13. Number of I/Os in Each Bank in EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1932-Pin FineLine BGA Package *(Note 1)*



Note to Figure 6–13:

(1) There are two additional PMA-only transceiver channels in each transceiver bank.

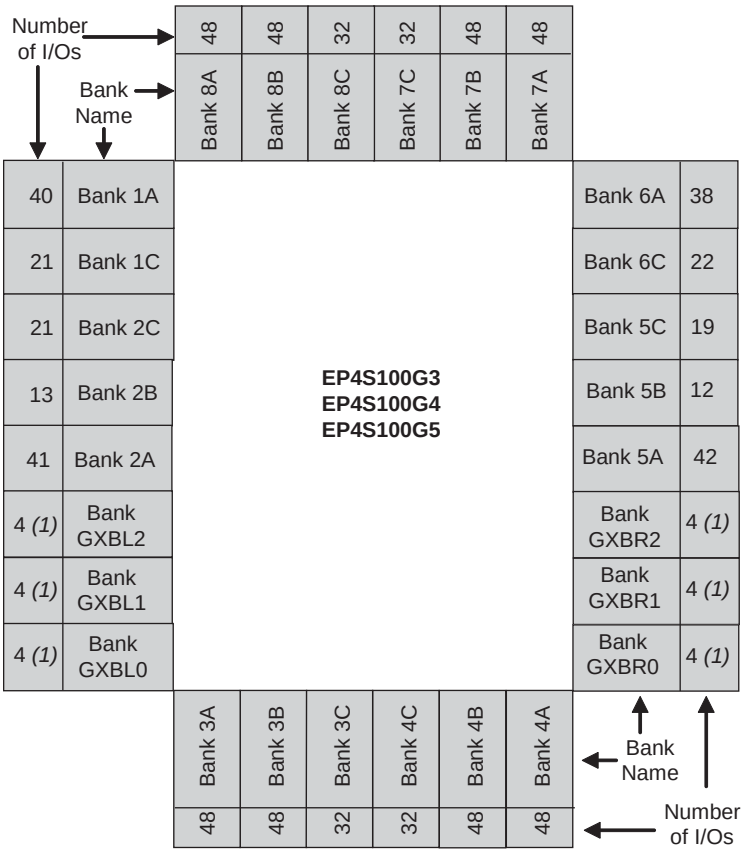
Figure 6–14. Number of I/Os in Each Bank in EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1760-Pin FineLine BGA Package *(Note 1)*



Note to Figure 6–14:

(1) There are two additional PMA-only transceiver channels in each transceiver bank.

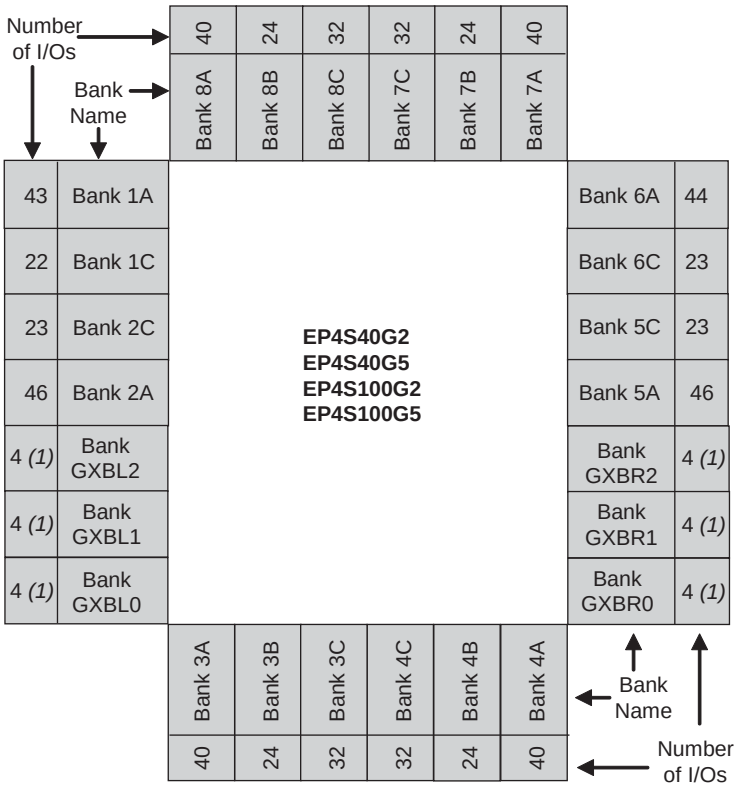
Figure 6–15. Number of I/Os in Each Bank in EP4S100G3, EP4S100G4, and EP4S100G5 Devices in the 1932-Pin FineLine BGA Package *(Note 1)*



Note to Figure 6–15:

(1) There are two additional PMA-only transceiver channels in each transceiver bank.

Figure 6–16. Number of I/Os in Each Bank in EP4S40G2, EP4S40G5, EP4S100G2, and EP4S100G5 Devices in the 1517-Pin FineLine BGA Package *(Note 1)*



Note to Figure 6–16:

(1) There are two additional PMA-only transceiver channels in each transceiver bank.

I/O Structure

The I/O element (IOE) in Stratix IV devices contain a bidirectional I/O buffer and I/O registers to support a complete embedded bidirectional single data rate or DDR transfer. The IOEs are located in I/O blocks around the periphery of the Stratix IV device. There are up to four IOEs per row I/O block and four IOEs per column I/O block. The row IOEs drive row, column, or direct link interconnects. The column IOEs drive column interconnects.

The Stratix IV bidirectional IOE also supports the following features:

- Programmable input delay
- Programmable output-current strength
- Programmable slew rate
- Programmable output delay
- Programmable bus-hold
- Programmable pull-up resistor
- Open-drain output
- On-chip series termination with calibration

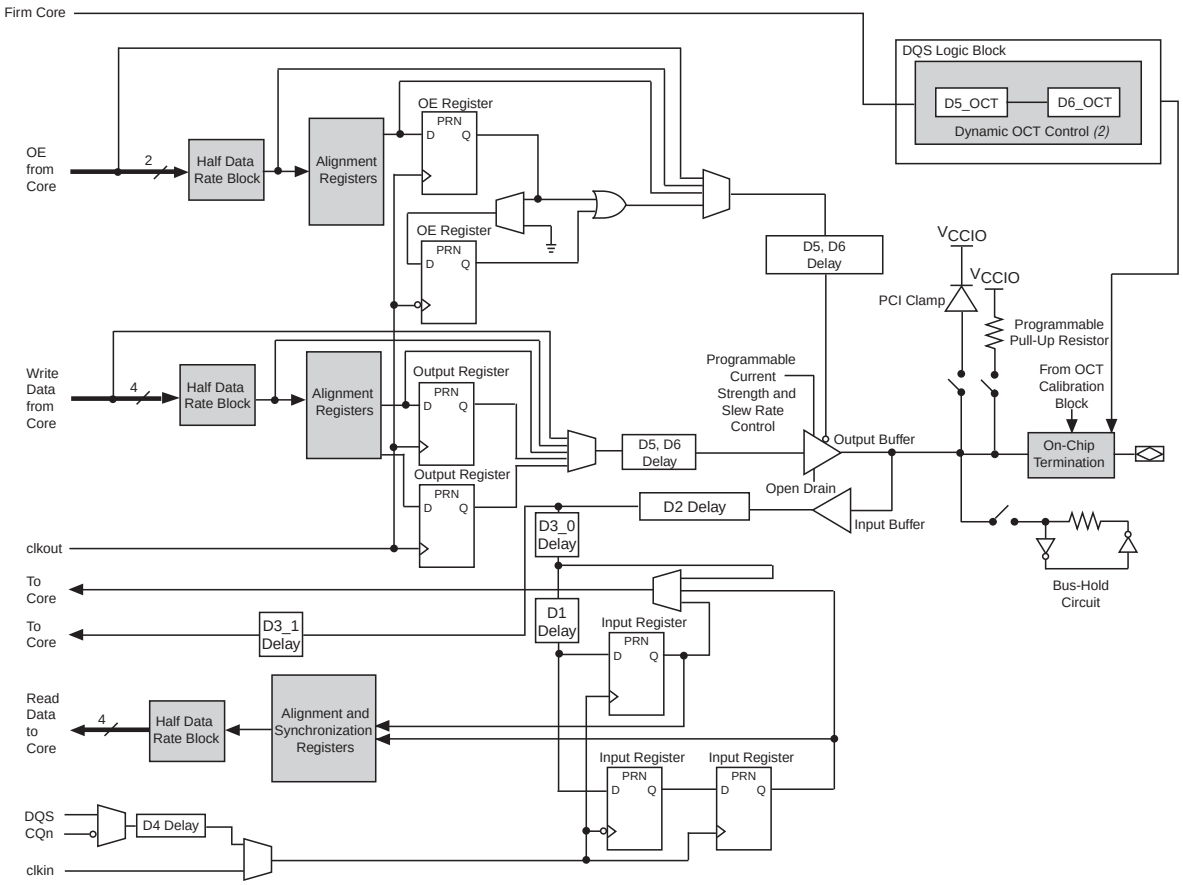
- On-chip series termination without calibration
- On-chip parallel termination with calibration
- On-chip differential termination
- PCI clamping diode

I/O registers are composed of the input path for handling data from the pin to the core, the output path for handling data from the core to the pin, and the output-enable (OE) path for handling the OE signal to the output buffer. These registers allow faster source-synchronous register-to-register transfers and resynchronization. The input path consists of the DDR input registers, alignment and synchronization registers, and HDR. You can bypass each block of the input path.

The output and OE paths are divided into output or OE registers, alignment registers, and HDR blocks. You can bypass each block of the output and OE paths.

Figure 6–17 shows the Stratix IV IOE structure.

Figure 6–17. Stratix IV IOE Structure *(Note 1)*, *(2)*



Notes to Figure 6–17:

(1) The `D3_0` and `D3_1` delays have the same available settings in the Quartus II software.
(2) One dynamic OCT control is available per DQ/DQS group.

For more information about I/O registers and how they are used for memory applications, refer to the *External Memory Interfaces* chapter.

3.3-V I/O Interface

Stratix IV I/O buffers support 3.3-V I/O standards. You can use them as transmitters or receivers in your system. The output high voltage (V_{OH}), output low voltage (V_{OL}), input high voltage (V_{IH}), and input low voltage (V_{IL}) levels meet the 3.3-V I/O standards specifications defined by EIA/JEDEC Standard JESD8-B with margin when the Stratix IV V_{CCIO} voltage is powered by 3.0 V.

To ensure device reliability and proper operation, when interfacing with a 3.3-V I/O system using Stratix IV devices, ensure that you do not violate the absolute maximum ratings of the devices. Altera recommends performing IBIS simulation to determine that the overshoot and undershoot voltages are within the guidelines.

When using the Stratix IV device as a transmitter, you can use slow slew rate and series termination to limit overshoot and undershoot at the I/O pins, but they are not required. Transmission line effects that cause large voltage deviations at the receiver are associated with an impedance mismatch between the driver and the transmission lines. By matching the impedance of the driver to the characteristic impedance of the transmission line, you can significantly reduce overshoot voltage. You can use a series termination resistor placed physically close to the driver to match the total driver impedance to the transmission line impedance. Stratix IV devices support series OCT for all LVTTL and LVCMOS I/O standards in all I/O banks.

When using the Stratix IV device as a receiver, you can use a clamping diode (on-chip or off-chip) to limit overshoot, though this is not required. Stratix IV devices provide an optional on-chip PCI-clamping diode for column I/O pins. You can use this diode to protect the I/O pins against overshoot voltage.

The 3.3-V I/O standard is supported using bank supply voltage (V_{CCIO}) at 3.0 V. In this method, the clamping diode (on-chip or off-chip), when enabled, can sufficiently clamp overshoot voltage to within the DC and AC input voltage specifications. The clamped voltage can be expressed as the sum of the supply voltage (V_{CCIO}) and the diode forward voltage.

For more information about the absolute maximum rating and maximum allowed overshoot during transitions, refer to the *DC and Switching Characteristics* chapter.

External Memory Interfaces

In addition to the I/O registers in each IOE, Stratix IV devices also have dedicated registers and phase-shift circuitry on all I/O banks for interfacing with external memory interfaces.

For more information about external memory interfaces, refer to the *External Memory Interfaces* chapter.

High-Speed Differential I/O with DPA Support

Stratix IV devices have the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer

- Data realignment
- Dynamic phase aligner (DPA)
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs)

For more information about DPA support, refer to the *High-Speed Differential I/O Interfaces with DPA* chapter.

Current Strength

The output buffer for each Stratix IV device I/O pin has a programmable current strength control for certain I/O standards. You can use programmable current strength to mitigate the effects of high signal attenuation due to a long transmission line or a legacy backplane. The LVTTL, LVCMOS, SSTL, and HSTL standards have several levels of current strength that you can control. Table 6–3 lists the programmable current strength for Stratix IV devices.

Table 6–3. Programmable Current Strength *(Note 1), (2)*

I/O Standard	I_{OH} / I_{OL} Current Strength Setting (mA) for Column I/O Pins	I_{OH} / I_{OL} Current Strength Setting (mA) for Row I/O Pins
3.3-V LVTTL	16, 12, 8, 4	12, 8, 4
3.3-V LVCMOS	16, 12, 8, 4	8, 4
2.5-V LVCMOS	16, 12, 8, 4	12, 8, 4
1.8-V LVCMOS	12, 10, 8, 6, 4, 2	8, 6, 4, 2
1.5-V LVCMOS	12, 10, 8, 6, 4, 2	8, 6, 4, 2
1.2-V LVCMOS	8, 6, 4, 2	4, 2
SSTL-2 Class I	12, 10, 8	12, 8
SSTL-2 Class II	16	16
SSTL-18 Class I	12, 10, 8, 6, 4	12, 10, 8, 6, 4
SSTL-18 Class II	16, 8	16, 8
SSTL-15 Class I	12, 10, 8, 6, 4	8, 6, 4
SSTL-15 Class II	16, 8	—
HSTL-18 Class I	12, 10, 8, 6, 4	12, 10, 8, 6, 4
HSTL-18 Class II	16	16
HSTL-15 Class I	12, 10, 8, 6, 4	8, 6, 4
HSTL-15 Class II	16	—
HSTL-12 Class I	12, 10, 8, 6, 4	8, 6, 4
HSTL-12 Class II	16	—

Notes to Table 6–3:

(1) The default setting in the Quartus II software is 50-ΩOCT R_S without calibration for all non-voltage reference and HSTL and SSTL Class I I/O standards. The default setting is 25-ΩOCT R_S without calibration for HSTL and SSTL Class II I/O standards.

(2) The 3.3-V LVTTL and 3.3-V LVCMOS are supported using V_{CCIO} and V_{CCPD} at 3.0 V.

Altera recommends performing IBIS or SPICE simulations to determine the best current strength setting for your specific application.

Slew Rate Control

The output buffer for each Stratix IV device regular- and dual-function I/O pin has a programmable output slew-rate control that you can configure for low-noise or high-speed performance. A faster slew rate provides high-speed transitions for high-performance systems. A slower slew rate can help reduce system noise, but adds a nominal delay to the rising and falling edges. Each I/O pin has an individual slew-rate control, allowing you to specify the slew rate on a pin-by-pin basis.

You cannot use the programmable slew rate feature when using OCT R_S.

The Quartus II software allows four settings for programmable slew rate control—0, 1, 2, and 3—where 0 is slow slew rate and 3 is fast slew rate. Figure 6–4 lists the default slew rate settings from the Quartus II software.

Table 6–4. Default Slew Rate Settings

I/O Standard	Slew Rate Option	Default Slew Rate
1.2-V, 1.5-V, 1.8-V, 2.5-V LVCMOS, and 3.3-V LVTTL/LVCMOS	0, 1, 2, 3	3
SSTL-2, SSTL-18, SSTL-15, HSTL-18, HSTL-15, and HSTL-12	0, 1, 2, 3	3
3.0-V PCI/PCI-X	0, 1, 2, 3	3
LVDS_E_1R, mini-LVDS_E_1R, and RSDS_E_1R	0, 1, 2, 3	3
LVDS_E_3R, mini-LVDS_E_3R, and RSDS_E_3R	0, 1, 2, 3	3

You can use faster slew rates to improve the available timing margin in memory-interface applications or when the output pin has high-capacitive loading.

Altera recommends performing IBIS or SPICE simulations to determine the best slew rate setting for your specific application.

I/O Delay

The following sections describe programmable IOE delay and programmable output buffer delay.

Programmable IOE Delay

The Stratix IV device IOE includes programmable delays, shown in Figure 6–17 on page 6–18, that you can activate to ensure zero hold times, minimize setup times, or increase clock-to-output times. Each pin can have a different input delay from pin-to-input register or a delay from output register-to-output pin values to ensure that the bus has the same delay going into or out of the device. This feature helps read and time margins because it minimizes the uncertainties between signals in the bus.

For more information about programmable IOE delay specifications, refer to the *DC and Switching Characteristics* chapter.

Programmable Output Buffer Delay

Stratix IV devices support delay chains built inside the single-ended output buffer, as shown in Figure 6–17 on page 6–18. The delay chains can independently control the rising and falling edge delays of the output buffer, providing the ability to adjust the output-buffer duty cycle, compensate channel-to-channel skew, reduce simultaneous switching output (SSO) noise by deliberately introducing channel-to-channel skew, and improve high-speed memory-interface timing margins. Stratix IV devices support four levels of output buffer delay settings. The default setting is **No Delay**.

For more information about programmable output buffer delay specifications, refer to the *DC and Switching Characteristics* chapter.

Open-Drain Output

Stratix IV devices provide an optional open-drain output (equivalent to an open collector output) for each I/O pin. When configured as open drain, the logic value of the output is either high-Z or 0. Typically, an external pull-up resistor is required to provide logic high.

Bus Hold

Each Stratix IV device I/O pin provides an optional bus-hold feature. Bus-hold circuitry can weakly hold the signal on an I/O pin at its last-driven state. Because the bus-hold feature holds the last-driven state of the pin until the next input signal is present, you do not need an external pull-up or pull-down resistor to hold a signal level when the bus is tri-stated.

Bus-hold circuitry also pulls non-driven pins away from the input threshold voltage where noise can cause unintended high-frequency switching. You can select this feature individually for each I/O pin. The bus-hold output drives no higher than V_{CCIO} to prevent over-driving signals. If you enable the bus-hold feature, you cannot use the programmable pull-up option. Disable the bus-hold feature if the I/O pin is configured for differential signals.

Bus-hold circuitry uses a resistor with a nominal resistance (R_{BH}) of approximately 7 kΩ to weakly pull the signal level to the last-driven state.

For more information about the specific sustaining current driven through this resistor and the overdrive current used to identify the next-driven input level, refer to the *DC and Switching Characteristics* chapter.

Bus-hold circuitry is active only after configuration. When going into user mode, the bus-hold circuit captures the value on the pin present at the end of configuration.

Pull-Up Resistor

Each Stratix IV device I/O pin provides an optional programmable pull-up resistor during user mode. If you enable this feature for an I/O pin, the pull-up resistor (typically 25 KΩ) weakly holds the I/O to the V_{CCIO} level.

Programmable pull-up resistors are only supported on user I/O pins and are not supported on dedicated configuration pins, JTAG pins, or dedicated clock pins. If you enable the programmable pull-up option, you cannot use the bus-hold feature.

Pre-Emphasis

Stratix IV LVDS transmitters support programmable pre-emphasis to compensate for the frequency dependent attenuation of the transmission line. The Quartus II software allows four settings for programmable pre-emphasis.

For more information about programmable pre-emphasis, refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix IV Devices* chapter.

Differential Output Voltage

Stratix IV LVDS transmitters support programmable V_{OD}. The programmable V_{OD} settings allow you to adjust output eye height to optimize trace length and power consumption. A higher V_{OD} swing improves voltage margins at the receiver end; a smaller V_{OD} swing reduces power consumption. The Quartus II software allows four settings for programmable V_{OD}.

For more information about programmable V_{OD}, refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix IV Devices* chapter.

MultiVolt I/O Interface

The Stratix IV architecture supports the MultiVolt I/O interface feature that allows the Stratix IV devices in all packages to interface with systems of different supply voltages.

You can connect the `VCCIO` pins to a 1.2-, 1.5-, 1.8-, 2.5-, or 3.0-V power supply, depending on the output requirements. The output levels are compatible with systems of the same voltage as the power supply. (For example, when `VCCIO` pins are connected to a 1.5-V power supply, the output levels are compatible with 1.5-V systems.)

For more information about pin connection guidelines, refer to the *Stratix IV Device Family Pin Connection Guidelines*.

The Stratix IV VCCPD power pins must be connected to a 2.5- or 3.0-V power supply. Using these power pins to supply the pre-driver power to the output buffers increases the performance of the output pins. Table 6–5 lists Stratix IV MultiVolt I/O support.

Table 6–5. Stratix IV MultiVolt I/O Support *(Note 1)*

V_{CCIO} (V)	Input Signal (V)						Output Signal (V)					
	1.2	1.5	1.8	2.5	3.0	3.3	1.2	1.5	1.8	2.5	3.0	3.3
1.2	✓	—	—	—	—	—	✓	—	—	—	—	—
1.5	—	✓	✓	—	—	—	—	✓	—	—	—	—
1.8	—	✓	✓	—	—	—	—	—	✓	—	—	—
2.5	—	—	—	✓	✓(2)	✓(2)	—	—	—	✓	—	—
3.0	—	—	—	✓	✓	✓	—	—	—	—	✓	—

Notes to Table 6–5:

(1) The pin current may be slightly higher than the default value. You must verify that the driving device's V_{OL} maximum and V_{OH} minimum voltages do not violate the applicable Stratix IV V_{IL} maximum and V_{IH} minimum voltage specifications.

(2) Altera recommends that you use an external clamping diode on the I/O pins when the input signal is 3.0 V or 3.3 V. You have the option to use an internal clamping diode for column I/O pins.

On-Chip Termination Support and I/O Termination Schemes

Stratix IV devices feature dynamic series and parallel OCT to provide I/O impedance matching and termination capabilities. OCT maintains signal quality, saves board space, and reduces external component costs.

Stratix IV devices support:

- On-chip series (R_S) with calibration
- On-chip series (R_S) without calibration
- Parallel (R_T) with calibration
- Dynamic series termination for single-ended I/O standards
- Parallel termination for single-ended I/O standards
- On-chip differential termination (R_D) for differential LVDS I/O standards

Stratix IV devices support OCT in all I/O banks by selecting one of the OCT I/O standards.

These devices also support OCT R_S and R_T in the same I/O bank for different I/O standards if they use the same V_{CCIO} supply voltage. You can independently configure each I/O in an I/O bank to support OCT R_S, programmable current strength, or OCT R_T.

You cannot configure both OCT R_S and programmable current strength for the same I/O buffer.

A pair of RUP and RDN pins are available in a given I/O bank and are shared for series- and parallel-calibrated termination. The RUP and RDN pins share the same V_{CCIO} and GND, respectively, with the I/O bank where they are located. The RUP and RDN pins are dual-purpose I/Os and function as regular I/Os if you do not use the calibration circuit.

For calibration, connections are as follows:

- The RUP pin is connected to V_{CCIO} through an external 25-Ω ±1% or 50-Ω ±1% resistor for an on-chip series termination value of 25-Ω or 50-Ω, respectively.
- The RDN pin is connected to GND through an external 25-Ω ±1% or 50-Ω ±1% resistor for an on-chip series termination value of 25-Ω or 50-Ω, respectively.

For on-chip parallel termination, connections are as follows:

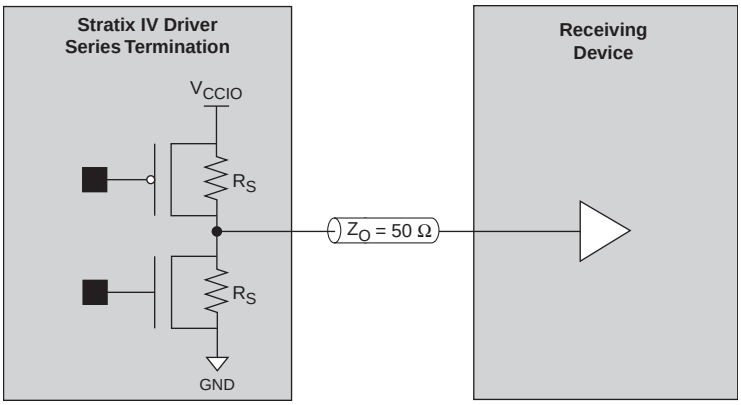
- The RUP pin is connected to V_{CCIO} through an external 50-Ω ±1% resistor.
- The RDN pin is connected to GND through an external 50-Ω ±1% resistor.

On-Chip Series (R_S) Termination Without Calibration

Stratix IV devices support driver-impedance matching to provide the I/O driver with controlled output impedance that closely matches the impedance of the transmission line. As a result, you can significantly reduce reflections. Stratix IV devices support on-chip series termination for single-ended I/O standards (Figure 6–18).

The R_S shown in Figure 6–18 is the intrinsic impedance of the output transistors. The typical R_S values are 25 Ω and 50 Ω. When you select matching impedance, current strength is no longer selectable.

Figure 6–18. On-Chip Series Termination Without Calibration



To use on-chip termination for the SSTL Class I standard, you must select the **50-Ω on-chip series termination** setting, thus eliminating the external 25-Ω R_S (to match the 50-Ω transmission line). For the SSTL Class II standard, you must select the **25-Ω on-chip series termination** setting (to match the 50-Ω transmission line and the near-end external 50-Ω pull-up to V_{TT}).

On-Chip Series Termination with Calibration

Stratix IV devices support on-chip series termination with calibration in all banks. The on-chip series termination calibration circuit compares the total impedance of the I/O buffer to the external 25-Ω ±1% or 50-Ω ±1% resistors connected to the RUP and RDN pins and dynamically enables or disables the transistors until they match.

The R_S shown in Figure 6–19 is the intrinsic impedance of the transistors. Calibration occurs at the end of device configuration. When the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers.

Figure 6–19. On-Chip Series Termination with Calibration

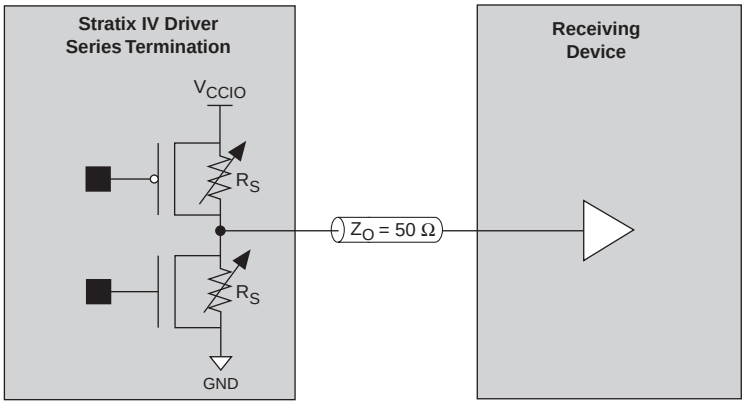


Table 6–6 lists the I/O standards that support on-chip series termination with calibration.

Table 6–6. Selectable I/O Standards for On-Chip Series Termination with and Without Calibration (Part 1 of 2)

I/O Standard	On-Chip Series Termination Setting	
	Row I/O (Ω)	Column I/O (Ω)
3.3-V LVTTL/LVCMOS	50	50
	25	25
2.5-V LVCMOS	50	50
	25	25
1.8-V LVCMOS	50	50
	25	25
1.5-V LVCMOS	50	50
		25
1.2-V LVCMOS	50	50
		25
SSTL-2 Class I	50	50
SSTL-2 Class II	25	25
SSTL-18 Class I	50	50
SSTL-18 Class II	25	25
SSTL-15 Class I	50	50

Table 6–6. Selectable I/O Standards for On-Chip Series Termination with and Without Calibration (Part 2 of 2)

I/O Standard	On-Chip Series Termination Setting	
	Row I/O (Ω)	Column I/O (Ω)
SSTL-15 Class II	—	25
HSTL-18 Class I	50	50
HSTL-18 Class II	25	25
HSTL-15 Class I	50	50
HSTL-15 Class II	—	25
HSTL-12 Class I	50	50
HSTL-12 Class II	—	25

Left-Shift Series Termination Control

Stratix IV devices support left-shift series termination control. You can use left-shift series termination control to get the calibrated OCT R_S with half of the impedance value of the external reference resistors connected to the RUP and RDN pins. This feature is useful in applications that require both 25-Ω and 50-Ω calibrated OCT R_S at the same V_{CCIO}. For example, if your application requires 25-Ω and 50-Ω calibrated OCT R_S for SSTL-2 Class I and Class II I/O standards, you only need one OCT calibration block with 50-Ω external reference resistors.

You can enable the left-shift series termination control feature in the ALTIOBUF megafunction in the Quartus II software. The Quartus II software only allows left-shift series termination control for 25-Ω calibrated OCT R_S with 50-Ω external reference resistors connected to the RUP and RDN pins. You can only use left-shift series termination control for the I/O standards that support 25-Ω calibrated OCT R_S .

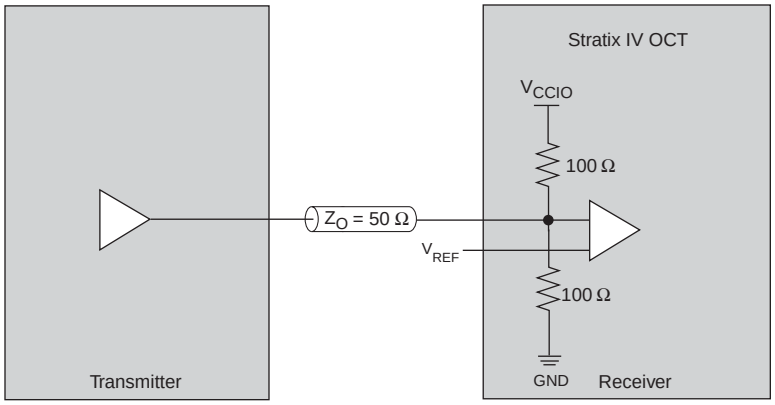
This feature is automatically enabled if you are using a bidirectional I/O with 25-Ω calibrated OCT R_S and 50-Ω parallel OCT.

For more information about how to enable the left-shift series termination feature in the ALTIOBUF megafunction, refer to the *ALTIOBUF Megafunction User Guide*.

On-Chip Parallel Termination with Calibration

Stratix IV devices support on-chip parallel termination with calibration in all banks. On-chip parallel termination with calibration is only supported for input configuration of input and bidirectional pins. Output pin configurations do not support on-chip parallel termination with calibration. Figure 6–20 shows on-chip parallel termination with calibration. When you use parallel OCT, the V_{CCIO} of the bank must match the I/O standard of the pin where the parallel OCT is enabled.

Figure 6–20. On-Chip Parallel Termination with Calibration



The on-chip parallel termination calibration circuit compares the total impedance of the I/O buffer to the external 50-Ω ±1% resistors connected to the RUP and RDN pins and dynamically enables or disables the transistors until they match. Calibration occurs at the end of device configuration. When the calibration circuit finds the correct impedance, it powers down and stops changing the characteristics of the drivers. Table 6–7 lists the I/O standards that support on-chip parallel termination with calibration.

Table 6–7. Selectable I/O Standards with On-Chip Parallel Termination with Calibration

I/O Standard	On-Chip Parallel Termination Setting (Column I/O) (Ω)	On-Chip Parallel Termination Setting (Row I/O) (Ω)
SSTL-2 Class I, II	50	50
SSTL-18 Class I, II	50	50
SSTL-15 Class I, II	50	50
HSTL-18 Class I, II	50	50
HSTL-15 Class I, II	50	50
HSTL-12 Class I, II	50	50
Differential SSTL-2 Class I, II	50	50
Differential SSTL-18 Class I, II	50	50
Differential SSTL-15 Class I, II	50	50
Differential HSTL-18 Class I, II	50	50
Differential HSTL-15 Class I, II	50	50
Differential HSTL-12 Class I, II	50	50

Expanded On-Chip Series Termination with Calibration

OCT calibration circuits always adjust OCT R_S to match the external resistors connected to the RUP and RDN pin; however, it is possible to achieve OCT R_S values other than the 25-Ω and 50-Ω resistors. Theoretically, if you need a different OCT R_S value, you can change the resistance connected to the RUP and RDN pins accordingly. Practically, the OCT R_S range that Stratix IV devices support is limited because of output buffer size and granularity limitations.

The Quartus II software only allows discrete OCT R_S calibration settings of 25, 40, 50, and 60 Ω. You can select the closest discrete value of OCT R_S with calibration settings in the Quartus II software to your system to achieve the closest timing. For example, if you are using 20-Ω OCT R_S with calibration in your system, you can select the **25-Ω OCT R_S with calibration** setting in the Quartus II software to achieve the closest timing.

Table 6–8 lists expanded OCT R_S with calibration supported in Stratix IV devices. Use expanded on-chip series termination with calibration of SSTL and HSTL for impedence matching to improve signal integrity but do not use it to meet the JEDEC standard.

Table 6–8. Selectable I/O Standards with Expanded On-Chip Series Termination with Calibration Range

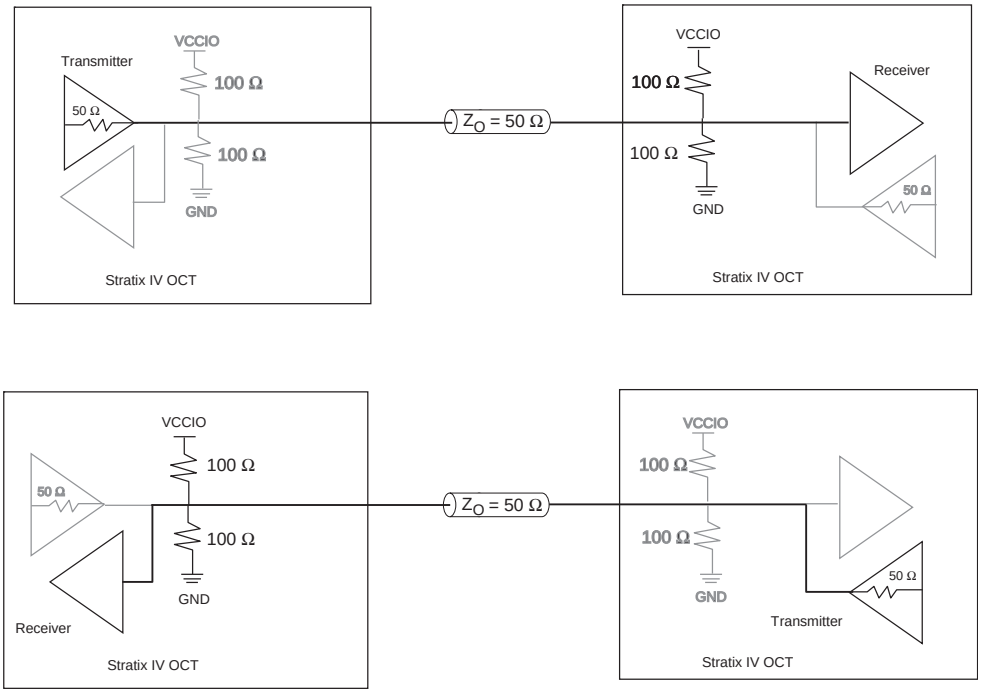
I/O Standard	Expanded OCT R_S Range	
	Row I/O (Ω)	Column I/O (Ω)
3.3-V LVTTL/LVCMOS	20–60	20–60
2.5-V LVTTL/LVCMOS	20–60	20–60
1.8-V LVTTL/LVCMOS	20–60	20–60
1.5-V LVTTL/LVCMOS	40–60	20–60
1.2-V LVTTL/LVCMOS	40–60	20–60
SSTL-2	20–60	20–60
SSTL-18	20–60	20–60
SSTL-15	40–60	20–60
HSTL-18	20–60	20–60
HSTL-15	40–60	20–60
HSTL-12	40–60	20–60

Dynamic On-Chip Termination

Stratix IV devices support on and off dynamic termination, both series and parallel, for a bidirectional I/O in all I/O banks. Figure 6–21 shows the termination schemes supported in Stratix IV devices. Dynamic parallel termination is enabled only when the bidirectional I/O acts as a receiver and is disabled when it acts as a driver. Similarly, dynamic series termination is enabled only when the bidirectional I/O acts as a driver and is disabled when it acts as a receiver. This feature is useful for terminating any high-performance bidirectional path because signal integrity is optimized depending on the direction of the data.

Using dynamic OCT helps save power because device termination is internal instead of external. Termination only switches on during input operation, thus drawing less static power.

Figure 6–21. Dynamic Parallel OCT in Stratix IV Devices

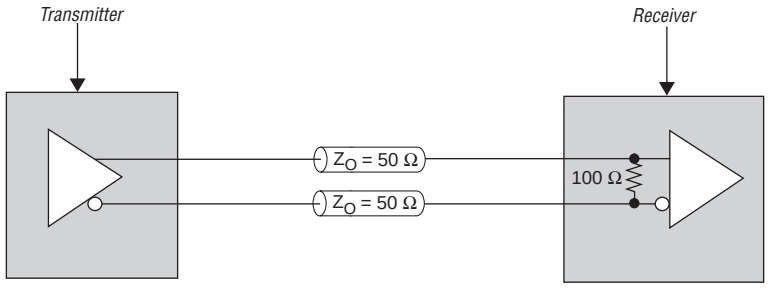


For more information about tolerance specifications for OCT with calibration, refer to the *DC and Switching Characteristics* chapter.

LVDS Input OCT (R_D)

Stratix IV devices support OCT for differential LVDS input buffers with a nominal resistance value of 100 Ω, as shown in Figure 6–22. Differential OCT R_D can be enabled in row I/O banks when both the V_{CCIO} and V_{CCPD} is set to 2.5 V. The column I/O banks do not support OCT R_D. The dedicated clock input pairs `CLK[1,3,8,10][p,n]`, `PLL_L[1,4]_CLK[p,n]`, and `PLL_R[1,4]_CLK[p,n]` on the row I/O banks of Stratix IV devices do not support R_D termination.

Figure 6–22. Differential Input OCT



For more information about differential on-chip termination, refer to the *High Speed Differential I/O Interfaces with DPA* chapter.

OCT Calibration

Stratix IV devices support calibrated on-chip series termination (R_S) and calibrated on-chip parallel termination (R_T) on all I/O pins. You can calibrate the device's I/O bank with any of the OCT calibration blocks available in the device.

OCT Calibration Block Location

Table 6–9 and Table 6–10 list the location of OCT calibration blocks in Stratix IV devices. Table 6–9 lists the OCT calibration blocks in Banks 1A through 4C. Table 6–10 lists the OCT calibration blocks in Banks 5A through 8C. For both tables, the following legend applies:

- "✓" indicates I/O banks with OCT calibration block
- "X" indicates I/O banks without OCT calibration block
- "—" indicates I/O banks that are not available in the device

Table 6–9 and Table 6–10 do not show transceiver banks and transceiver calibration blocks.

Table 6–9. OCT Calibration Block Counts and Placement in Stratix IV Devices (1A through 4C) (Part 1 of 2)

Device	Pin	Number of OCT Blocks	Bank											
			1A	1B	1C	2A	2B	2C	3A	3B	3C	4A	4B	4C
EP4SE230	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
EP4SE360	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
	1152	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4SE530	1152	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1517	10	✓	X	X	✓	X	X	✓	X	✓	✓	X	X
	1760	10	✓	X	X	✓	X	X	✓	X	✓	✓	X	X
EP4SE820	1152	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1517	10	✓	X	X	✓	X	X	✓	X	✓	✓	X	X
	1760	10	✓	X	X	✓	X	X	✓	X	✓	✓	X	X
EP4SGX70	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
EP4SGX110	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
	1152	8	✓	—	X	—	—	—	✓	—	X	✓	—	X
EP4SGX180	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
	1152	8	✓	—	X	—	—	—	✓	X	X	✓	X	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4SGX230	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
	1152	8	✓	—	X	—	—	—	✓	X	X	✓	X	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4SGX290	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
	1152	8	✓	—	X	—	—	—	✓	X	X	✓	X	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1760	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1932	10	✓	X	X	✓	—	X	✓	X	✓	✓	X	X

Table 6–9. OCT Calibration Block Counts and Placement in Stratix IV Devices (1A through 4C) (Part 2 of 2)

Device	Pin	Number of OCT Blocks	Bank											
			1A	1B	1C	2A	2B	2C	3A	3B	3C	4A	4B	4C
EP4SGX360	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
	1152	8	✓	—	X	—	—	—	✓	X	X	✓	X	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1760	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1932	10	✓	X	X	✓	—	X	✓	X	✓	✓	X	X
EP4SGX530	1152	8	✓	—	X	—	—	—	✓	X	✓	✓	X	X
	1517	10	✓	—	X	✓	—	X	✓	X	✓	✓	X	X
	1760	10	✓	—	X	✓	—	X	✓	X	✓	✓	X	X
	1932	10	✓	—	X	✓	X	X	✓	X	✓	✓	X	X
EP4S40G2	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4S40G5	1517	10	✓	—	X	✓	—	X	✓	X	✓	✓	X	X
EP4S100G2	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4S100G3	1932	10	✓	—	X	✓	X	X	✓	X	✓	✓	X	X
EP4S100G4	1932	10	✓	—	X	✓	X	X	✓	X	✓	✓	X	X
EP4S100G5	1517	10	✓	—	X	✓	—	X	✓	X	✓	✓	X	X
	1932	10	✓	—	X	✓	X	X	✓	X	✓	✓	X	X

Table 6–10. OCT Calibration Block Counts and Placement in Stratix IV Devices (5A through 8C) (Part 1 of 2)

Device	Pin	Number of OCT Blocks	Bank											
			5A	5B	5C	6A	6B	6C	7A	7B	7C	8A	8B	8C
EP4SE230	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
EP4SE360	780	8	✓	—	X	✓	—	X	✓	—	X	✓	—	X
	1152	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4SE530	1152	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1517	10	✓	X	X	✓	X	X	✓	X	X	✓	X	✓
	1760	10	✓	X	X	✓	X	X	✓	X	X	✓	X	✓
EP4SE820	1152	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1517	10	✓	X	X	✓	X	X	✓	X	X	✓	X	✓
	1760	10	✓	X	X	✓	X	X	✓	X	X	✓	X	✓
EP4SGX70	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
EP4SGX110	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
	1152	8	—	—	—	✓	—	X	✓	—	X	✓	—	X
EP4SGX180	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
	1152	8	—	—	—	✓	—	X	✓	X	X	✓	✓	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4SGX230	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
	1152	8	—	—	—	✓	—	X	✓	X	X	✓	✓	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X

Table 6–10. OCT Calibration Block Counts and Placement in Stratix IV Devices (5A through 8C) (Part 2 of 2)

Device	Pin	Number of OCT Blocks	Bank											
			5A	5B	5C	6A	6B	6C	7A	7B	7C	8A	8B	8C
EP4SGX290	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
	1152	8	—	—	—	✓	—	X	✓	X	X	✓	X	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1760	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1932	10	✓	—	X	✓	X	X	✓	X	X	✓	X	✓
EP4SGX360	780	8	—	—	—	—	—	—	✓	—	X	✓	—	X
	1152	8	—	—	—	✓	—	X	✓	X	X	✓	X	X
	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1760	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
	1932	10	✓	—	X	✓	X	X	✓	X	X	✓	X	✓
EP4SGX530	1152	8	—	—	—	✓	—	X	✓	X	X	✓	X	✓
	1517	10	✓	—	X	✓	—	X	✓	X	X	✓	X	✓
	1760	10	✓	—	X	✓	—	X	✓	X	X	✓	X	✓
	1932	10	✓	X	X	✓	—	X	✓	X	X	✓	X	✓
EP4S40G2	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4S40G5	1517	10	✓	—	X	✓	—	X	✓	X	X	✓	X	✓
EP4S100G2	1517	8	✓	—	X	✓	—	X	✓	X	X	✓	X	X
EP4S100G3	1932	10	✓	X	X	✓	—	X	✓	X	X	✓	X	✓
EP4S100G4	1932	10	✓	X	X	✓	—	X	✓	X	X	✓	X	✓
EP4S100G5	1517	10	✓	—	X	✓	—	X	✓	X	X	✓	X	✓
	1932	10	✓	X	X	✓	—	X	✓	X	X	✓	X	✓

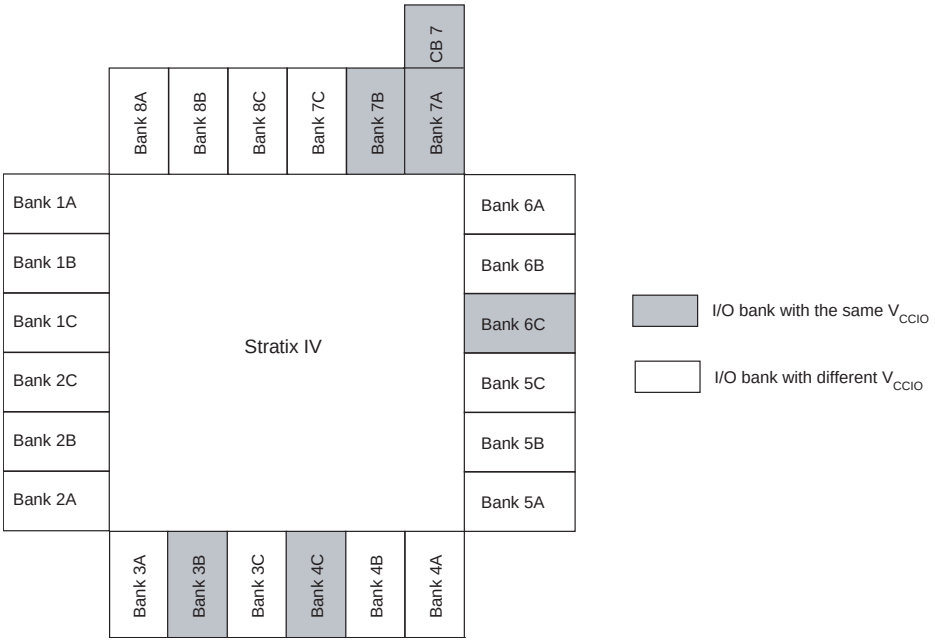
Sharing an OCT Calibration Block on Multiple I/O Banks

An OCT calibration block has the same V_{CCIO} as the I/O bank that contains the block. OCT R_S calibration is supported on all I/O banks with different V_{CCIO} voltage standards, up to the number of available OCT calibration blocks. You can configure the I/O banks to receive calibration codes from any OCT calibration block with the same V_{CCIO}. All I/O banks with the same V_{CCIO} can share one OCT calibration block, even if that particular I/O bank has an OCT calibration block.

For example, Figure 6–23 shows a group of I/O banks that has the same V_{CCIO} voltage. If a group of I/O banks has the same V_{CCIO} voltage, you can use one OCT calibration block to calibrate the group of I/O banks placed around the periphery. Because 3B, 4C, 6C, and 7B have the same V_{CCIO} as bank 7A, you can calibrate all four I/O banks (3B, 4C, 6C, and 7B) with the OCT calibration block located in bank 7A. You can enable this by serially shifting out OCT R_S calibration codes from the OCT calibration block located in bank 7A to the I/O banks located around the periphery.

Figure 6–23 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. It is a graphical representation only. This figure does not show transceiver banks and transceiver calibration blocks.

Figure 6–23. Example of Calibrating Multiple I/O Banks with One Shared OCT Calibration Block



OCT Calibration Block Modes of Operation

Stratix IV devices support OCT R_S and OCT R_T on all I/O banks. The calibration can occur in either power-up or user mode.

Power-Up Mode

In power-up mode, OCT calibration is automatically performed at power up. Calibration codes are shifted to selected I/O buffers before transitioning to user mode.

User Mode

In user mode, the `OCTUSRCLK`, `ENAOCT`, `nCLRUSR`, and `ENASER[9..0]` signals are used to calibrate and serially transfer calibration codes from each OCT calibration block to any I/O. Table 6–11 lists the user-controlled calibration block signal names and their descriptions.

Table 6–11. OCT Calibration Block Ports for User Control

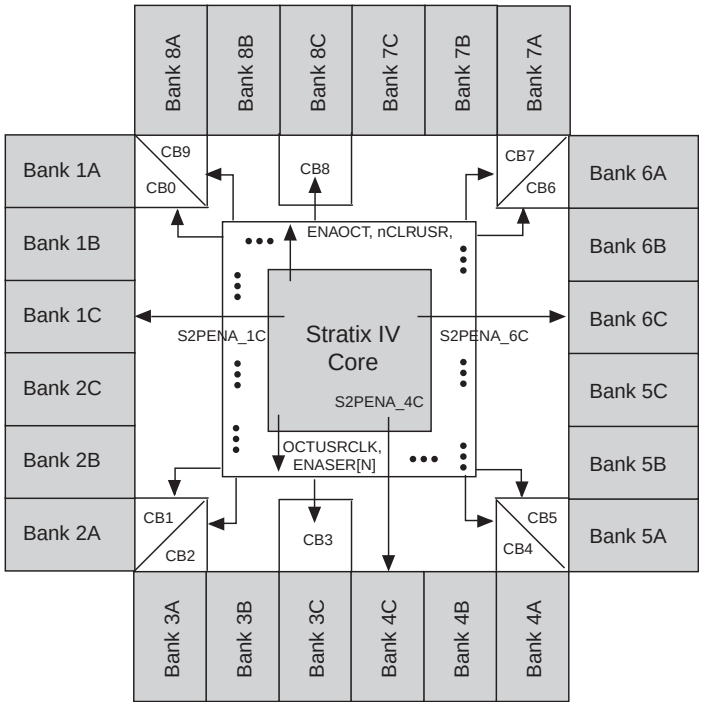
Signal Name	Description
OCTUSRCLK	Clock for OCT block.
ENAOCT	Enable OCT Termination (Generated by user IP).
ENASER[9..0]	When ENOCT = 0, each signal enables the OCT serializer for the corresponding OCT calibration block. When ENAOCT = 1, each signal enables OCT calibration for the corresponding OCT calibration block.
S2PENA_<*bank#*>	Serial-to-parallel load enable per I/O bank.
nCLRUSR	Clear user.

Figure 6–24 shows the flow of the user signal. When ENAOCT is 1, all OCT calibration blocks are in calibration mode; when ENAOCT is 0, all OCT calibration blocks are in serial data transfer mode. The OCTUSRCLK clock frequency must be 20 MHz or less.

You must generate all user signals on the rising edge of OCTUSRCLK.

Figure 6–24 does not show transceiver banks and transceiver calibration blocks.

Figure 6–24. Signals Used for User Mode Calibration



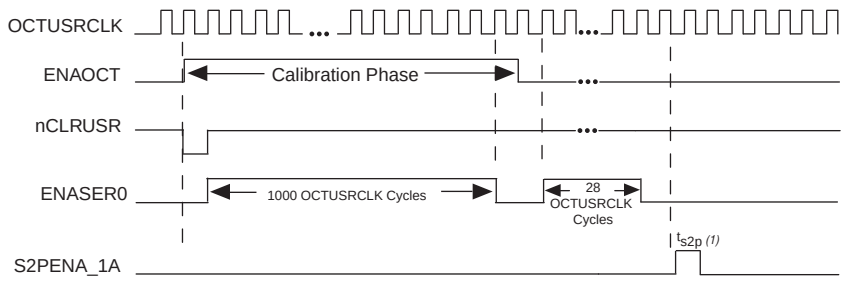
OCT Calibration

Figure 6–25 shows user mode signal-timing waveforms. To calibrate OCT block[N] (where N is a calibration block number), you must assert ENAOCT one cycle before asserting ENASER[N]. Also, nCLRUSR must be set to low for one OCTUSRCLK cycle before the ENASER[N] signal is asserted. Assert the ENASER[N] signals for 1000 OCTUSRCLK cycles to perform OCTRS and OCTRT calibration. You can de-assert ENAOCT one clock cycle after the last ENASER is de-asserted.

Serial Data Transfer

After you complete calibration, you must serially shift out the 28-bit OCT calibration codes (14-bit OCT RS code and 14-bit OCT RT) from each OCT calibration block to the corresponding I/O buffers. Only one OCT calibration block can send out the codes at any time by asserting only one ENASER[N] signal at a time. After you de-assert ENAOCT, wait at least one OCTUSRCLK cycle to enable any ENASER[N] signal to begin serial transfer. To shift the 28-bit code from the OCT calibration block[N], you must assert ENASER[N] for exactly 28 OCTUSRCLK cycles. Between two consecutive asserted ENASER signals, there must be at least one OCTUSRCLK cycle gap. (Figure 6–25).

Figure 6–25. OCT User Mode Signal—Timing Waveform for One OCT Block



Note to Figure 6–25:

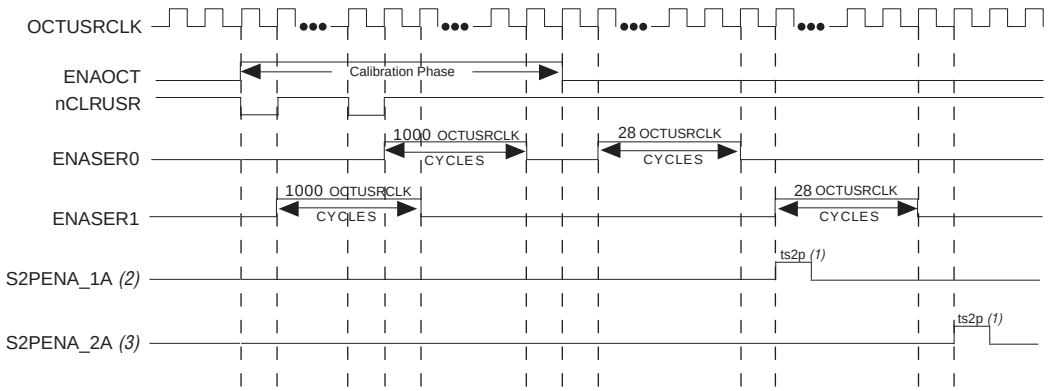
(1) $t_{s2p} \geq 25$ ns.

After calibrated codes are shifted in serially to each I/O bank, the calibrated codes must be converted from serial to parallel format before being used in the I/O buffers. Figure 6–25 shows the S2PENA signals that can be asserted at any time to update the calibration codes in each I/O bank. All I/O banks that received the codes from the same OCT calibration block can have S2PENA asserted at the same time, or at a different time, even while another OCT calibration block is calibrating and serially shifting codes. The S2PENA signal is asserted one OCTUSRCLK cycle after ENASER is de-asserted for at least 25 ns. You cannot use I/Os for transmitting or receiving data when their S2PENA is asserted for parallel codes transfer.

Example of Using Multiple OCT Calibration Blocks

Figure 6–26 shows a signal timing waveform for two OCT calibration blocks doing R_S and R_T calibration. Calibration blocks can start calibrating at different times by asserting the ENASER signals at different times. ENAOCT must remain asserted while any calibration is ongoing. You must set nCLRUSR low for one OCTUSRCLK cycle before each ENASER[N] signal is asserted. In Figure 6–26, when you set nCLRUSR to 0 for the second time to initialize OCT calibration block 0, this does not affect OCT calibration block 1, whose calibration is already in progress.

Figure 6–26. OCT User-Mode Signal Timing Waveform for Two OCT Blocks



Notes to Figure 6–26:

(1) ts2p ≥ 25 ns.

(2) S2PENA_1A is asserted in Bank 1A for calibration block 0.

(3) S2PENA_2A is asserted in Bank 2A for calibration block 1.

RS Calibration

If only R_S calibration is used for an OCT calibration block, its corresponding ENASER signal only needs to be asserted for 240 OCTUSRCLK cycles.

You must assert the ENASER signal for 28 OCTUSRCLK cycles for serial transfer.

Termination Schemes for I/O Standards

The following sections describe the different termination schemes for the I/O standards used in Stratix IV devices.

Single-Ended I/O Standards Termination

Voltage-referenced I/O standards require both an input reference voltage, V_{REF}, and a termination voltage, V_{TT}. The reference voltage of the receiving device tracks the termination voltage of the transmitting device.

Figure 6–27 and Figure 6–28 show the details of SSTL and HSTL I/O termination on Stratix IV devices.

In Stratix IV devices, you cannot use series and parallel OCT simultaneously. For more information, refer to "Dynamic On-Chip Termination" on page 6–29.

Figure 6–27. SSTL I/O Standard Termination

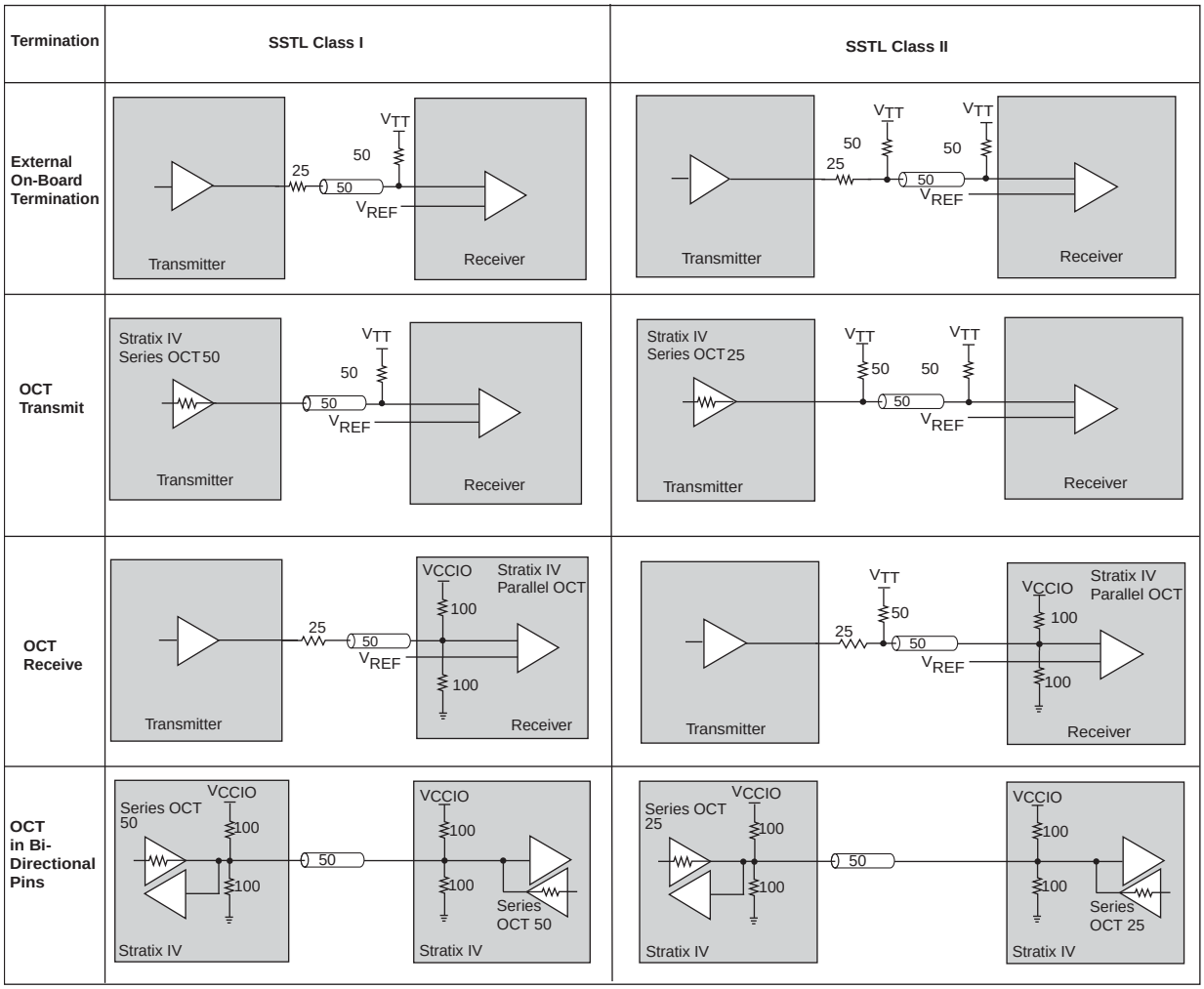
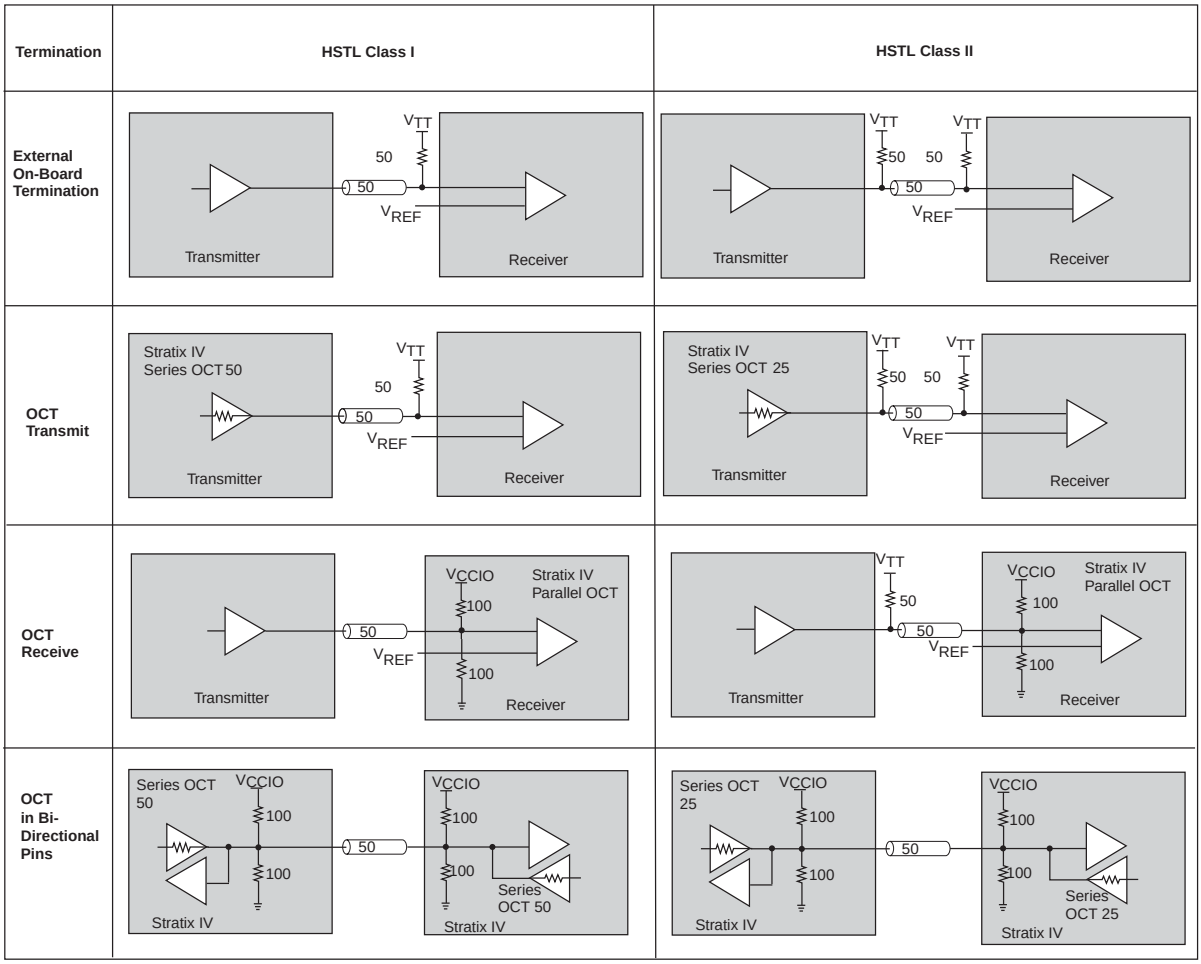


Figure 6–28. HSTL I/O Standard Termination



Differential I/O Standards Termination

Stratix IV devices support differential SSTL-18 and SSTL-2, differential HSTL-18, HSTL-15, HSTL-12, LVDS, LVPECL, RSDS, and mini-LVDS. Figure 6–29 through Figure 6–35 show the details of various differential I/O terminations on these devices.

Differential HSTL and SSTL outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted.

Figure 6–29. Differential SSTL I/O Standard Termination

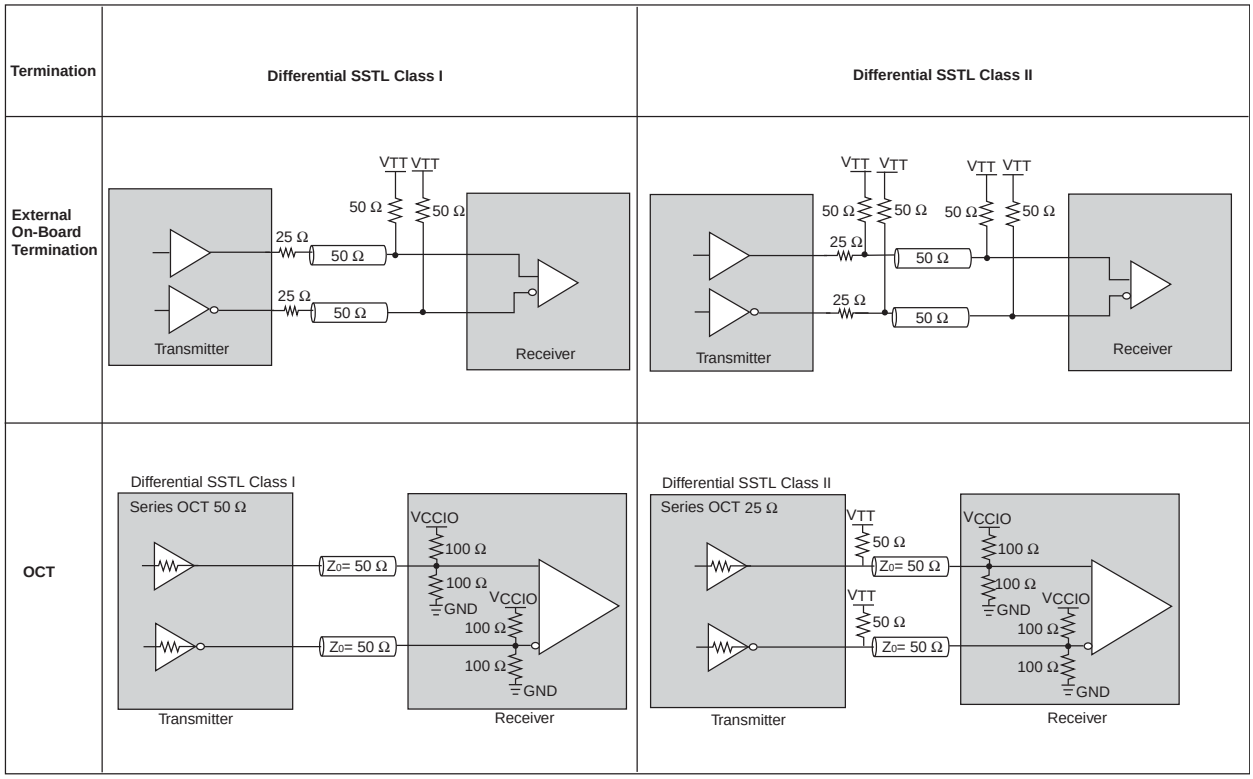
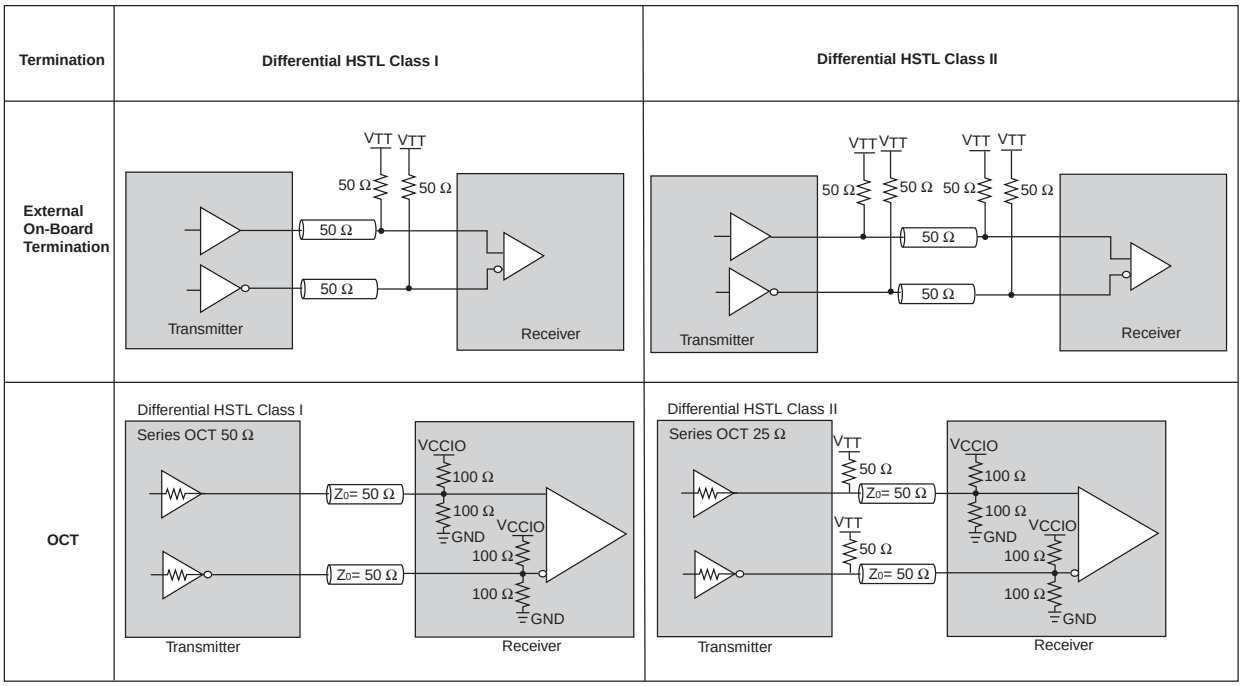


Figure 6–30. Differential HSTL I/O Standard Termination

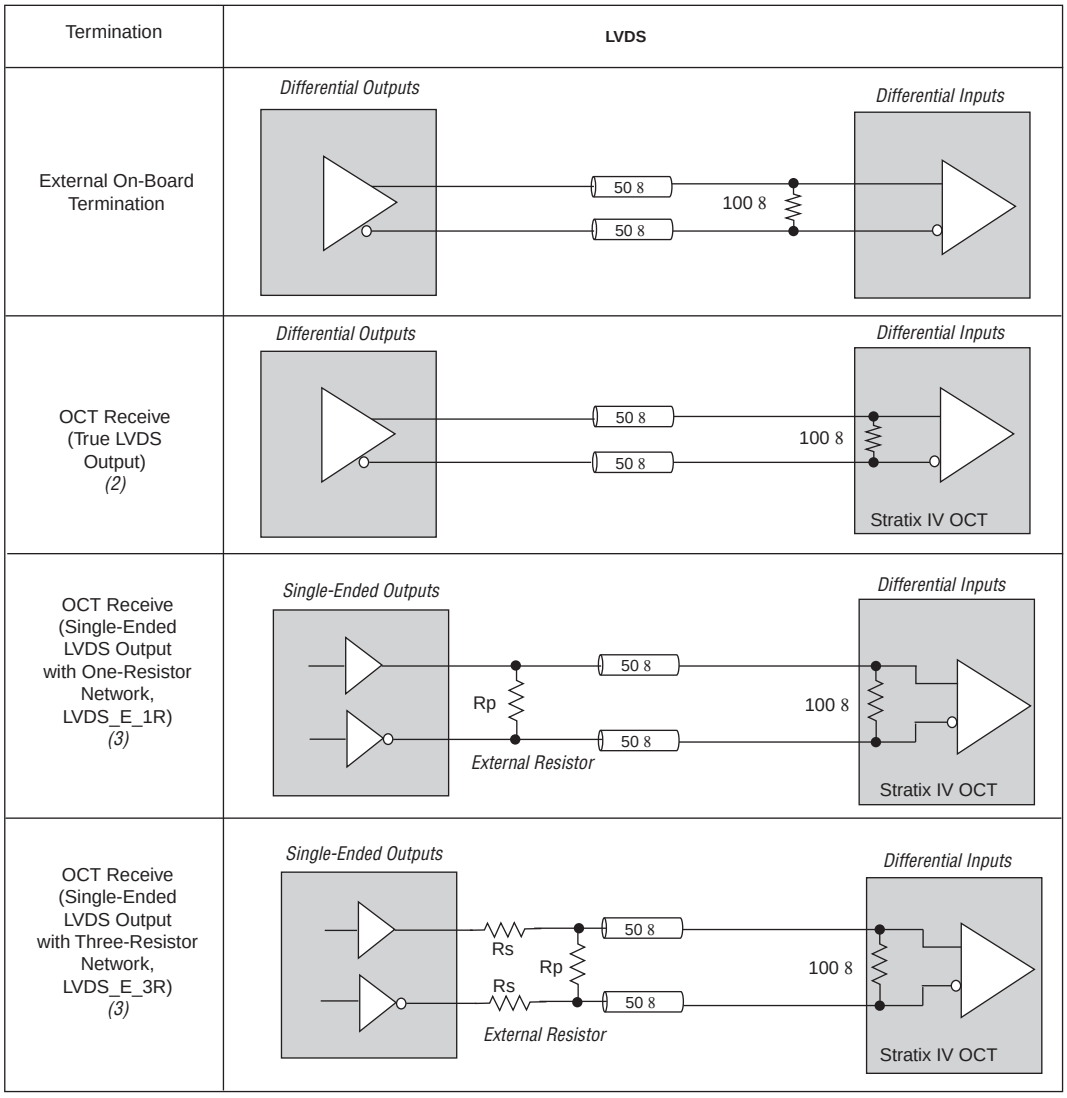


LVDS

The LVDS I/O standard is a differential high-speed, low-voltage swing, low-power, general-purpose I/O interface standard. In Stratix IV devices, the LVDS I/O standard requires a 2.5-V V_{CCIO} level. The LVDS input buffer requires 2.5-V V_{CCPD}. Use this standard in applications requiring high-bandwidth data transfer, such as backplane drivers and clock distribution. LVDS requires a 100-Ωtermination resistor between the two signals at the input buffer. Stratix IV devices provide an optional 100-Ω differential termination resistor in the device using on-chip differential termination.

Figure 6–31 shows LVDS termination. The on-chip differential resistor is only available in the row I/O banks.

Figure 6–31. LVDS I/O Standard Termination *(Note 1)*



Notes to Figure 6–31:

(1) For LVDS output with a three-resistor network, the R_S and R_P values are 120 and 170 Ω respectively. For LVDS output with a one-resistor network, the R_P value is 120 Ω.

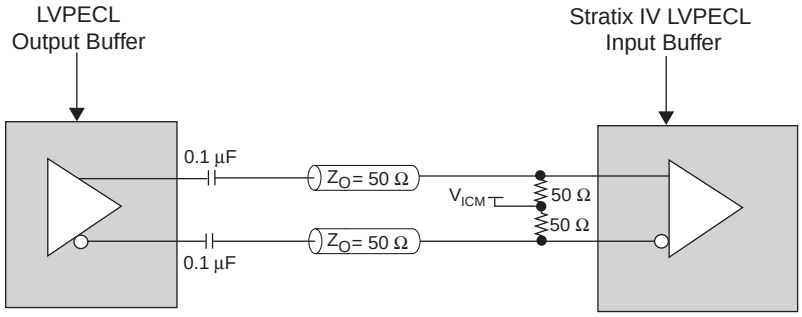
(2) Side I/O banks support true LVDS output buffers.

(3) Column and side I/O banks support LVDS_E_1R and LVDS_E_3R I/O standards using two single-ended output buffers.

Differential LVPECL

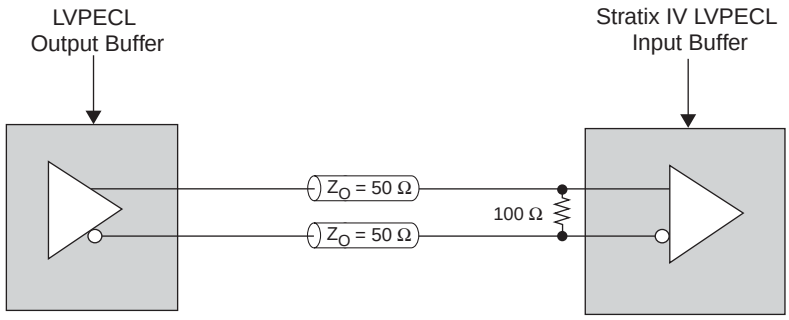
In Stratix IV devices, the LVPECL I/O standard is supported on input clock pins on column and row I/O banks. LVPECL output operation is not supported in Stratix IV devices. LVDS input buffers are used to support LVPECL input operation. AC coupling is required when the LVPECL common-mode voltage of the output buffer is higher than the LVPECL input common-mode voltage. Figure 6–32 shows the AC-coupled termination scheme. The 50-Ω resistors used at the receiver end are external to the device.

Figure 6–32. LVPECL AC-Coupled Termination



DC-coupled LVPECL is supported if the LVPECL output common mode voltage is within the Stratix IV LVPECL input buffer specification (Figure 6–33).

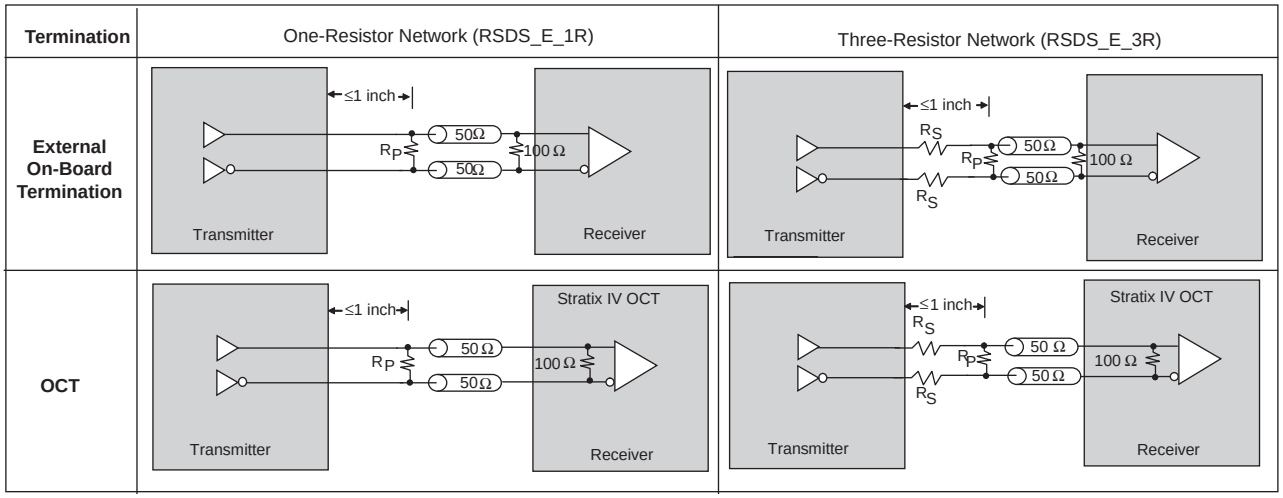
Figure 6–33. LVPECL DC-Coupled Termination



RSDS

Stratix IV devices support the RSDS output standard with data rates up to 230 Mbps using LVDS output buffer types. For transmitters, use two single-ended output buffers with the external one- or three-resistor networks in the column I/O bank, as shown in Figure 6–34. The one-resistor topology is for data rates up to 200 Mbps. The three-resistor topology is for data rates above 200 Mbps. The row I/O banks support RSDS output using true LVDS output buffers without an external resistor network.

Figure 6–34. RSDS I/O Standard Termination *(Note 1)*



Note to Figure 6–34:

(1) The R_S and R_P values are pending characterization.

A resistor network is required to attenuate the LVDS output-voltage swing to meet RSDS specifications. You can modify the three-resistor network values to reduce power or improve noise margin. The resistor values chosen must satisfy Equation 6–1.

Equation 6–1.

$$\frac{R_s \times \frac{R_p}{2}}{R_s + \frac{R_p}{2}} = 50\Omega$$

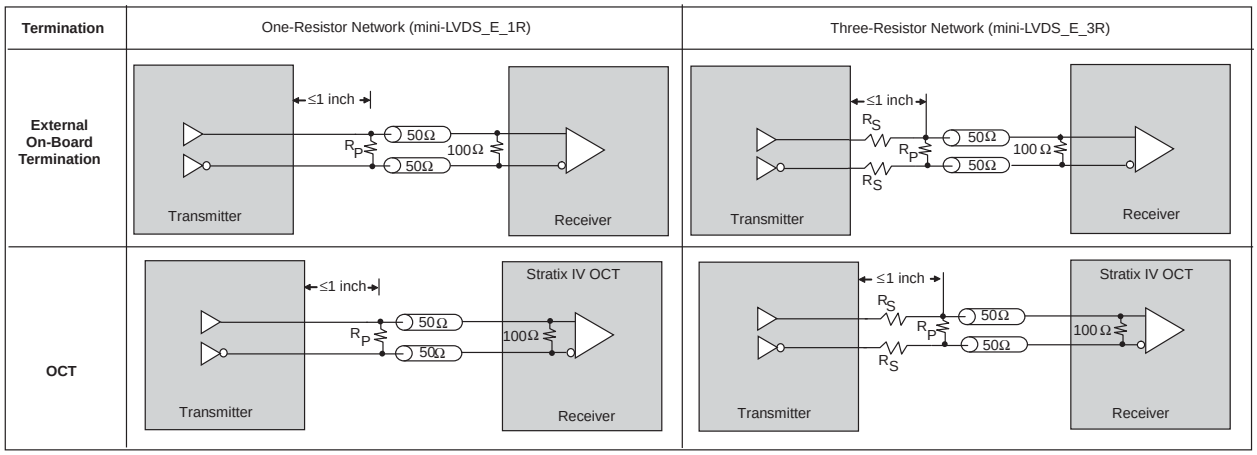
Altera recommends that you perform additional simulations using IBIS models to validate that custom resistor values meet the RSDS requirements.

For more information about the RSDS I/O standard, refer to the *RSDS Specification* from the National Semiconductor website at www.national.com.

Mini-LVDS

Stratix IV devices support the mini-LVDS output standard with data rates up to 340 Mbps using LVDS output buffer types. For transmitters, use two single-ended output buffers with external one- or three-resistor networks, as shown in Figure 6–35. The one-resistor topology is for data rates up to 200 Mbps. The three-resistor topology is for data rates above 200 Mbps. The row I/O banks support mini-LVDS output using true LVDS output buffers without an external resistor network.

Figure 6–35. Mini-LVDS I/O Standard Termination *(Note 1)*



Note to Figure 6–35:

(1) The R_S and R_P values are pending characterization.

A resistor network is required to attenuate the LVDS output voltage swing to meet the mini-LVDS specifications. You can modify the three-resistor network values to reduce power or improve noise margin. The resistor values chosen must satisfy Equation 6–1 on page 6–43.

Altera recommends that you perform additional simulations using IBIS models to validate that custom resistor values meet the RSDS requirements.

For more information about the mini-LVDS I/O standard, see the *mini-LVDS Specification* from the Texas Instruments website at www.ti.com.

Design Considerations

Although Stratix IV devices feature various I/O capabilities for high-performance and high-speed system designs, there are several other design considerations that require your attention to ensure the success of your designs.

I/O Bank Restrictions

Each I/O bank can simultaneously support multiple I/O standards. The following sections provide guidelines for mixing non-voltage-referenced and voltage-referenced I/O standards in Stratix IV devices.

Non-Voltage-Referenced Standards

Each I/O bank of a Stratix IV device has its own `VCCIO` pins and supports only one V_{CCIO}, either 1.2, 1.5, 1.8, 2.5, or 3.0 V. An I/O bank can simultaneously support any number of input signals with different I/O standard assignments if it meets the V_{CCIO} and V_{CCPD} requirement, as shown in Table 6–2 on page 6–3.

For output signals, a single I/O bank supports non-voltage-referenced output signals that are driving at the same voltage as V_{CCIO}. Because an I/O bank can only have one V_{CCIO} value, it can only drive out that one value for non-voltage-referenced signals. For example, an I/O bank with a 2.5-V V_{CCIO} setting can support 2.5-V standard inputs and outputs as well as 3.0-V LVCMOS inputs (but not output or bidirectional pins).

Voltage-Referenced Standards

To accommodate voltage-referenced I/O standards, each Stratix IV device's I/O bank supports multiple VREF pins feeding a common V_{REF} bus. The number of available VREF pins increases as device density increases. If these pins are not used as VREF pins, they cannot be used as generic I/O pins and must be tied to V_{CCIO} or GND. Each bank can only have a single V_{CCIO} voltage level and a single V_{REF} voltage level at a given time.

An I/O bank featuring single-ended or differential standards can support voltage-referenced standards if all voltage-referenced standards use the same V_{REF} setting.

For performance reasons, voltage-referenced input standards use their own V_{CCPD} level as the power source. This feature allows you to place voltage-referenced input signals in an I/O bank with a V_{CCIO} of 2.5 V or below. For example, you can place HSTL-15 input pins in an I/O bank with 2.5-V V_{CCIO}. However, the voltage-referenced input with parallel OCT enabled requires the V_{CCIO} of the I/O bank to match the voltage of the input standard.

Voltage-referenced bidirectional and output signals must be the same as the I/O bank's V_{CCIO} voltage. For example, you can only place SSTL-2 output pins in an I/O bank with a 2.5-V V_{CCIO}.

Mixing Voltage-Referenced and Non-Voltage-Referenced Standards

An I/O bank can support both voltage-referenced and non-voltage-referenced pins by applying each of the rule sets individually. For example, an I/O bank can support SSTL-18 inputs and 1.8-V inputs and outputs with a 1.8-V V_{CCIO} and a 0.9-V V_{REF}. Similarly, an I/O bank can support 1.5-V standards, 1.8-V inputs (but not outputs), and HSTL and HSTL-15 I/O standards with a 1.5-V V_{CCIO} and 0.75-V V_{REF}.

Document Revision History

Table 6–12 shows the revision history for this chapter.

Table 6–12. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated Table 6–2, Table 6–4, Table 6–6, Table 6–9, and Table 6–10. ■ Updated Figure 6–1, Figure 6–2, Figure 6–4, Figure 6–5, Figure 6–6, Figure 6–8, Figure 6–9, Figure 6–10, Figure 6–11, Figure 6–12, Figure 6–13, and Figure 6–31. ■ Added Table 6–8. ■ Added Figure 6–7, Figure 6–14, Figure 6–15, and Figure 6–16. ■ Added "Left-Shift Series Termination Control" and "Expanded On-Chip Series Termination with Calibration" sections. ■ Updated "MultiVolt I/O Interface", "RSDS", "Mini-LVDS", and "Non-Voltage-Referenced Standards" sections. ■ Deleted Figure 6-5: Number of I/Os in Each Bank in EP4SE290 and EP4SE360 in the 1517-Pin FineLine BGA Package. ■ Minor text edits.	—
June 2009 v2.3	■ Added introductory sentences to improve search ability. ■ Removed the Conclusion section.	—
April 2009 v2.2	■ Updated Figure 6–2. ■ Updated Table 6–8 and Table 6–9. ■ Deleted Figure 6-14.	—
March 2009 v2.1	■ Updated Table 6–1, Table 6–2,Table 6–3, Table 6–4, Table 6–6, Table 6–8, and Table 6–9. ■ Updated Figure 6–2, Figure 6–7, Figure 6–8, Figure 6–9, Figure 6–10, Figure 6–11, and Figure 6–12. ■ Added Figure 6–14. ■ Removed Equation 6–2. ■ Removed "Referenced Documents" section.	—
November 2008 v2.0	■ Updated "Modular I/O Banks" on page 6–7. ■ Updated Figure 6–3. ■ Updated Figure 6–21. ■ Made minor editorial changes.	—
May 2008 v1.0	Initial release.	—

7. External Memory Interfaces in Stratix IV Devices

SIV51007-3.0

This chapter describes external memory interfaces available with the Stratix® IV device family and that family's silicon capability to support external memory interfaces. To support the level of system bandwidth achievable with Altera® Stratix IV FPGAs, the devices provide an efficient architecture to quickly and easily fit wide external memory interfaces within their small modular I/O bank structure. The I/Os are designed to provide high-performance support for existing and emerging external double data rate (DDR) memory standards, such as DDR3, DDR2, DDR SDRAM, QDRII+, QDRII SRAM, and RLDRAM II.

Stratix IV I/O elements provide easy-to-use built-in functionality required for a rapid and robust implementation with features such as dynamic calibrated on-chip termination (OCT), trace mismatch compensation, read- and write-leveling circuit for DDR3 SDRAM interfaces, half data rate (HDR) blocks, and 4- to 36-bit programmable DQ group widths.

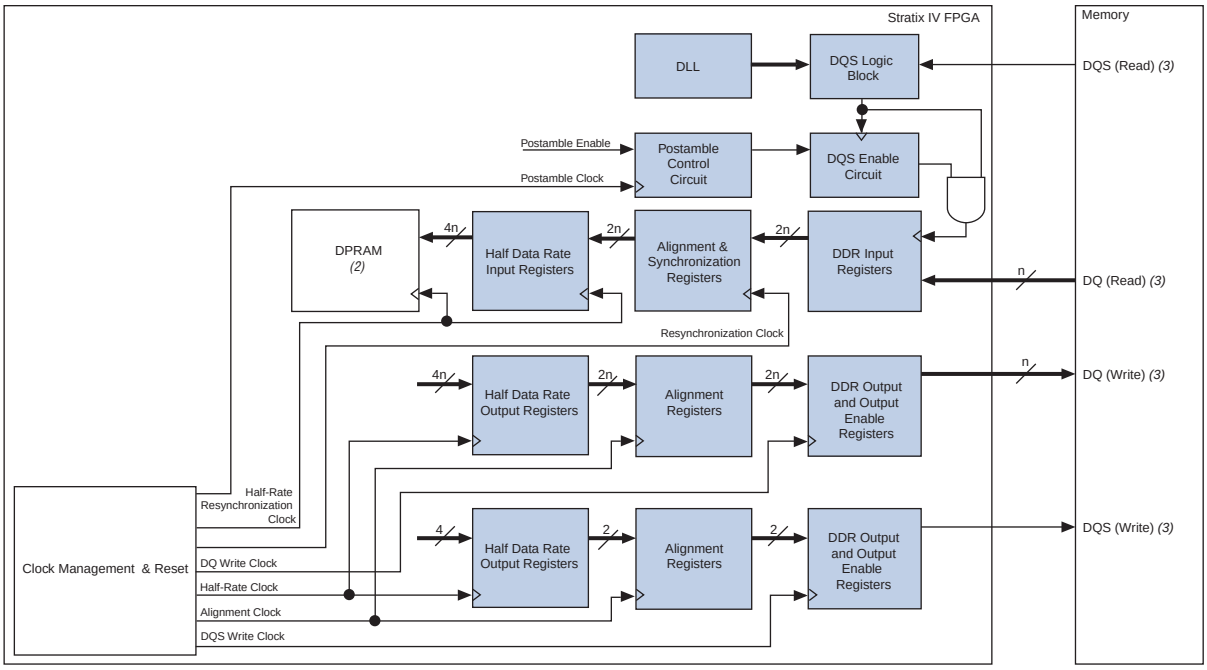
The high-performance memory interface solution is backed-up by a self-calibrating megafunction (ALTMEMPHY), optimized to take advantage of the Stratix IV I/O structure and the TimeQuest Timing Analyzer, which completes the picture by providing the total solution for the highest reliable frequency of operation across process, voltage, and temperature (PVT) variations.

This chapter contains the following sections:

For more information about external memory system performance specifications, board design guidelines, timing analysis, simulation, and debugging information, refer to the *External Memory Interface Handbook*.

Figure 7–1 shows an overview of the memory interface data path that uses all the Stratix IV I/O element (IOE) features.

Figure 7–1. External Memory Interface Data Path Overview *(Note 1)*, *(2)*



Notes to Figure 7–1:

(1) You can bypass each register block.

(2) The blocks used for each memory interface may differ slightly. The shaded blocks are part of the Stratix IV IOE.

(3) These signals may be bidirectional or unidirectional, depending on the memory standard. When bidirectional, the signal is active during both read and write operations.

Memory interfaces use Stratix IV device features such as delay-locked loops (DLLs), dynamic OCT control, read- and write-leveling circuitry, and I/O features such as OCT, programmable input delay chains, programmable output delay, slew rate adjustment, and programmable drive strength.

For more information about I/O features, refer to the *I/O Features in Stratix IV Devices* chapter.

The ALTMEMPHY megafunction instantiates a phase-locked loop (PLL) and PLL reconfiguration logic to adjust the phase shift based on VT variation.

For more information about the Stratix IV PLL, refer to the *Clock Networks and PLLs in Stratix IV Devices* chapter. For more information about the ALTMEMPHY megafunction, refer to the *External Memory PHY Interface Megafunction User Guide (ALTMEMPHY)*.

Memory Interfaces Pin Support

A typical memory interface requires data (D, Q, or DQ), data strobe (DQS/CQ and DQSn/CQn), address, command, and clock pins. Some memory interfaces use data mask (DM, BWSn, or NWSn) pins to enable write masking and QVLD pins to indicate that the read data is ready to be captured. This section describes how Stratix IV devices support all these different pins.

For more information about pin connections, refer to the *Stratix IV Device Family Pin Connection Guidelines*.

Table 7–1 lists the pin connections between a Stratix IV device and an external memory device.

Table 7–1. Stratix IV Memory Interface Pin Utilization (Part 1 of 2)

Pin Description	Memory Standard	Stratix IV Pin Utilization
Read Data	All	DQ
Write Data	All	DQ *(1)*
Parity, DM, BWSn, NWSn, QVLD, ECC	All	DQ *(1)*, *(2)*
Read Data Strobes/Clocks	■ DDR3 SDRAM ■ DDR2 SDRAM (with differential DQS signaling) *(3)* ■ RLDRAM II	Differential DQS/DQSn (also used as a write data clock)
	■ DDR2 SDRAM (with single-ended DQS signaling) *(3)* ■ DDR SDRAM	Single-ended DQS (also used as a write data clock)
	■ QDRII+ SRAM ■ QDRII SRAM	Complementary CQ/CQn
Write Data Clocks	■ QDRII+ SRAM *(4)* ■ QDRII SRAM *(4)* ■ RLDRAM II Separate I/O (SIO)	Any DQS and DQSn pin pairs associated with the DQ groups used for the write data pins *(1)*
	■ RLDRAM II Common I/O (CIO) *(6)*	Any DQ pin with `DIFFOUT` capability within the same DQS/DQ group or adjacent group as the read data (Q) pins, or in the same bank as the address and command pins

Table 7–1. Stratix IV Memory Interface Pin Utilization (Part 2 of 2)

Pin Description	Memory Standard	Stratix IV Pin Utilization
Memory Clocks (for Address and Command) *(5)*	■ DDR3 SDRAM with leveling	Any unused DQ or DQS pins with `DIFFIO_RX` capability for the `mem_clk[0]` and `mem_clk_n[0]` signals
		Any unused DQ or DQS pins with `DIFFOUT` capability for the `mem_clk[n:1]` and `mem_clk_n[n:1]` signals (where n is greater than or equal to 1)
	■ DDR3 SDRAM without leveling ■ DDR2 SDRAM (with differential DQS signaling) *(3)*	Any unused pins with `DIFFIO_RX` capability for the `mem_clk[0]` and `mem_clk_n[0]` signals
		Any unused pins with `DIFFOUT` capability for the `mem_clk[n:1]` and `mem_clk_n[n:1]` signals (where n is greater than or equal to 1)
	■ DDR2 SDRAM (with single-ended DQS signaling) *(3)* ■ DDR SDRAM ■ RLDRAM II	Any `DIFFOUT` pins

Notes to Table 7–1:

(1) If the write data signals are unidirectional, connect them, including the data mask pins, to a separate DQS/DQ group other than the read DQS/DQ group. Connect the write clock to the DQS and DQSn pin-pair associated with that DQS/DQ group. Do not use the CQ and CQn pin-pair as write clocks.

(2) The BWSn, NWSn, and DM pins must be part of the write DQS/DQ group, while parity, QVLD, and ECC pins must be part of the read DQS/DQ group. The ALTMEMPHY megafunction does not support the QVLD pin. However, if your design supports the QVLD pin, the QVLD pin must be part of the read DQS/DQ group as well.

(3) DDR2 SDRAM supports either single-ended or differential DQS signaling.

(4) QDRII+/QDRII SRAM devices use the K/K# clock pin-pair to latch write data, address, and command signals. The clocks must be part of the DQS/DQ group and follow the write data clock rules.

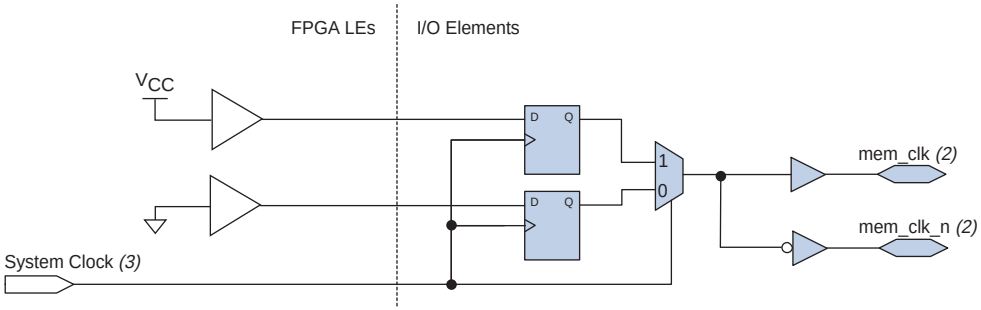
(5) ALTMEMPHY megafunction implementation for a DDR3, DDR2, or DDR SDRAM interface requires that you place all memory clock pin-pairs in a single DQ group of adequate width to minimize skew. For example, DIMMs requiring three memory clock pin-pairs must use a ×4 DQS/DQ group.

(6) When interfacing with RLDRAM II × 36 CIO devices, use two DQ pins in the × 16/× 18 DQS/DQ groups, which are the DQS/DQSn pins in the × 4 or × 8/× 9 DQS/DQ groups for the write data clock.

DDR3, DDR2, DDR SDRAM, and RLDRAM II devices use the CK and CK# signals to capture the address and command signals. Generate these signals to mimic the write-data strobe using Stratix IV DDR I/O registers (DDIOs) to ensure that the timing relationships between the CK/CK# and DQS signals (t_{DQSS}, t_{DSS}, and t_{DSH} in DDR3, DDR2, and DDR SDRAM devices or t_{CKDK} in RLDRAM II devices) are met. QDRII+ and QDRII SRAM devices use the same clock (K/K#) to capture write data, address, and command signals.

Memory clock pins in Stratix IV devices are generated using a DDIO register going to differential output pins (refer to Figure 7–2), marked in the pin table with `DIFFOUT`, `DIFFIO_TX`, or `DIFFIO_RX` prefixes. For more information about which pins to use for memory clock pins, refer to Table 7–1.

Figure 7–2. Memory Clock Generation *(Note 1)*



Notes to Figure 7–2:

(1) For pin location requirements, refer to Table 7–1 on page 7–3.

(2) The `mem_clk[0]` and `mem_clk_n[0]` pins for DDR3, DDR2, and DDR SDRAM interfaces use the I/O input buffer for feedback required by the ALTMEMPHY megafunction for tracking; therefore, use bidirectional I/O buffers for these pins. For memory interfaces using a differential DQS input, the input feedback buffer is configured as differential input. For memory interfaces using a single-ended DQS input, the input buffer is configured as a single-ended input. Using a single-ended input feedback buffer requires that I/O standard's VREF voltage is provided to that I/O bank's VREF pins.

(3) To minimize jitter, regional clock networks are required for memory output clock generation.

Stratix IV devices offer differential input buffers for differential read-data strobe and clock operations. In addition, Stratix IV devices also provide an independent DQS logic block for each CQn pin for complementary read-data strobe and clock operations. In the Stratix IV pin tables, the differential DQS pin pairs are denoted as DQS and DQSn pins, while the complementary CQ signals are denoted as CQ and CQn pins. DQSn and CQn pins are marked separately in the pin table. Each CQn pin connects to a DQS logic block and the shifted CQn signals go to the negative-edge input registers in the DQ IOE registers.

☞ Use differential DQS signaling for DDR2 SDRAM interfaces running at or above 333 MHz.

DQ pins can be bidirectional signals, as in DDR3, DDR2, and DDR SDRAM, and RLDRAM II common I/O (CIO) interfaces, or unidirectional signals, as in QDRII+, QDRII SRAM, and RLDRAM II separate I/O (SIO) devices. Connect the unidirectional read-data signals to Stratix IV DQ pins and the unidirectional write-data signals to a different DQS/DQ group than the read DQS/DQ group. Furthermore, the write clocks must be assigned to the DQS/DQSn pins associated to this write DQS/DQ group. Do not use the CQ/CQn pin-pair for write clocks.

☞ Using a DQS/DQ group for the write-data signals minimizes output skew, allows access to the write-leveling circuitry (for DDR3 SDRAM interfaces), and allows vertical migration. These pins also have access to deskewing circuitry (using programmable delay chains) that can compensate for delay mismatch between signals on the bus.

The DQS and DQ pin locations are fixed in the pin table. Memory interface circuitry is available in every Stratix IV I/O bank that does not support transceivers. All the memory interface pins support the I/O standards required to support DDR3, DDR2, DDR SDRAM, QDRII+, QDRII SRAM, and RLDRAM II devices.

The Stratix IV device family supports DQS and DQ signals with DQ bus modes of ×4, ×8/×9, ×16/×18, or ×32/×36, although not all devices support DQS bus mode ×32/×36. When any of these pins are not used for memory interfacing, you can use them as user I/Os. In addition, you can use any DQSn or CQn pins not used for clocking as DQ (data) pins. Table 7–2 lists pin support per DQS/DQ bus mode, including the DQS/CQ and DQSn/CQn pin pair.

Table 7–2. Stratix IV DQS/DQ Bus Mode Pins

Mode	DQSn Support	CQn Support	Parity or DM (Optional)	QVLD (Optional) *(1)*	Typical Number of Data Pins per Group	Maximum Number of Data Pins per Group *(2)*
×4	Yes	No	No *(6)*	No	4	5
×8/×9 *(3)*	Yes	Yes	Yes	Yes	8 or 9	11
×16/×18 *(4)*	Yes	Yes	Yes	Yes	16 or 18	23
×32/×36 *(5)*	Yes	Yes	Yes	Yes	32 or 36	47

Notes to Table 7–2:

(1) The QVLD pin is not used in the ALTMEMPHY megafunction.

(2) This represents the maximum number of DQ pins (including parity, data mask, and QVLD pins) connected to the DQS bus network with single-ended DQS signaling. When you use differential or complementary DQS signaling, the maximum number of data per group decreases by one. This number may vary per DQS/DQ group in a particular device. Check the pin table for the accurate number per group. For DDR3, DDR2, and DDR interfaces, the number of pins is further reduced for an interface larger than ×8 due to the need of one DQS pin for each ×8/×9 group that is used to form the x16/×18 and ×32/×36 groups.

(3) Two ×4 DQS/DQ groups are stitched to make a ×8/×9 group so there are a total of 12 pins in this group.

(4) Four ×4 DQS/DQ groups are stitched to make a ×16/×18 group.

(5) Eight ×4 DQS/DQ groups are stitched to make a ×32/×36 group.

(6) The DM pin can be supported if differential DQS is not used and the group does not have additional signals.

Table 7–3 lists the number of DQS/DQ groups available per side in each Stratix IV device. For a more detailed listing of the number of DQS/DQ groups available per bank in each Stratix IV device, see Figure 7–3 through Figure 7–17. These figures represent the die-top view of the Stratix IV device.

Table 7–3. Number of DQS/DQ Groups in Stratix IV Devices per Side (Part 1 of 2) *(Note 1)*

Device	Package	Side	×4 *(2)*	×8/×9	×16/×18	×32/×36 *(3)*
EP4SGX70 EP4SGX110 EP4SGX180 EP4SGX230	780-pin FineLine BGA	Left	14	6	2	0
		Top / Bottom	17	8	2	0
		Right	0	0	0	0
EP4SGX290 EP4SGX360	780-pin FineLine BGA	Left / Right	0	0	0	0
		Top / Bottom	18	8	2	0
EP4SE230 EP4SE360	780-pin FineLine BGA	Left / Right	14	6	2	0
		Top / Bottom	17	8	2	0
EP4SGX110	1152-pin FineLine BGA (with 16 transceivers)	Right / Left	7	3	1	0
		Top / Bottom	17	8	2	0

Table 7–3. Number of DQS/DQ Groups in Stratix IV Devices per Side (Part 2 of 2) *(Note 1)*

Device	Package	Side	×4 *(2)*	×8/×9	×16/×18	×32/×36 *(3)*
EP4SGX70 EP4SGX110	1152-pin FineLine BGA (with 24 transceivers)	Right / Left	14	6	2	0
		Top / Bottom	17	8	2	0
EP4SGX180 EP4SGX230 EP4SGX290 EP4SGX360 EP4SGX530	1152-pin FineLine BGA	Right / Left	13	6	2	0
		Top / Bottom	26	12	4	0
EP4SE360 EP4SE530 EP4SE820	1152-pin FineLine BGA	All sides	26	12	4	0
EP4SGX180 EP4SGX230 EP4SGX290 EP4SGX360 EP4SGX530	1517-pin FineLine BGA	All sides	26	12	4	0
EP4SE530 EP4SE820	1517-pin FineLine BGA	Right / Left	34	16	6	0
		Top / Bottom	38	18	8	4
EP4S40G2 EP4S40G5 EP4S100G2 EP4S100G5	1517-pin FineLine BGA	Left	12	3	1	0
		Top / Bottom	26	12	4	0
		Right	11	4	1	0
EP4SGX290 EP4SGX360 EP4SGX530	1760-pin FineLine BGA	Right / Left	26	12	4	0
		Top / Bottom	38	18	8	4
EP4SE530	1760-pin FineLine BGA	Right / Left	34	16	6	0
		Top / Bottom	38	18	8	4
EP4SE820	1760-pin FineLine BGA	Right / Left	40	18	6	0
		Top / Bottom	44	22	10	4
EP4SGX290 EP4SGX360 EP4SGX530	1932-pin FineLine BGA	Right / Left	29	13	4	0
		Top / Bottom	38	18	8	4
EP4S100G3 EP4S100G4 EP4S100G5	1932-pin FineLine BGA	Left	8	2	0	0
		Top / Bottom	38	18	8	4
		Right	7	1	0	0

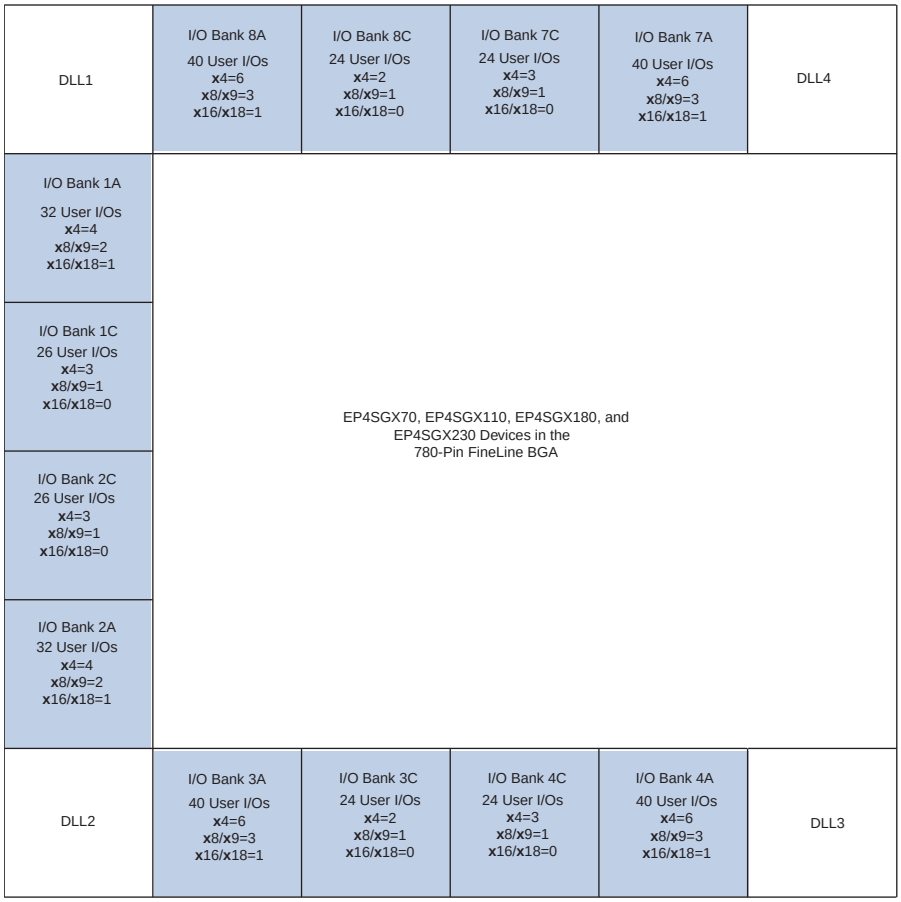
Notes to Table 7–3:

(1) These numbers are preliminary until the devices are available.

(2) Some of the ×4 groups may use R_{UP} and R_{DN} pins. You cannot use these groups if you use the Stratix IV calibrated OCT feature.

(3) To interface with a ×36 QDRII+/QDRII SRAM device in a Stratix IV FPGA that does not support the ×32/×36 DQS/DQ group, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

Figure 7–3. Number of DQS/DQ Groups per Bank in EP4SGX70, EP4SGX110, EP4SGX180, and EP4SGX230 Devices in the 780-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*. *(5)*



Notes to Figure 7–3:

(1) These numbers are preliminary until the devices are available.

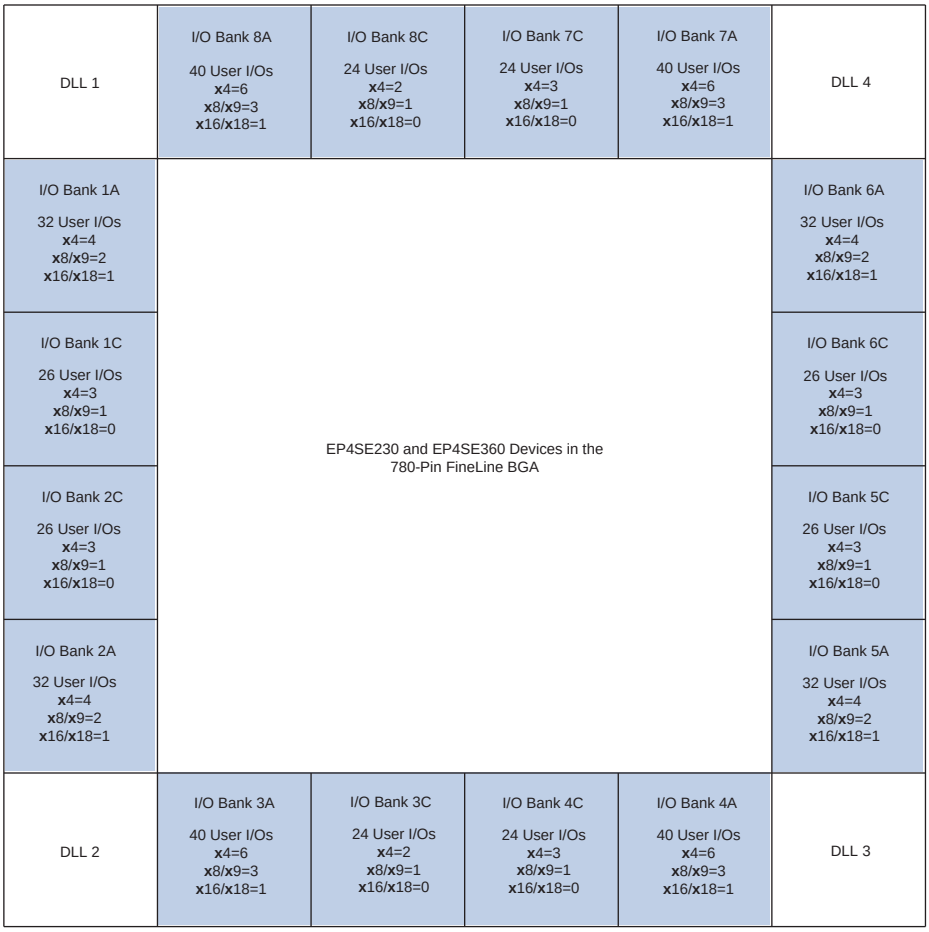
(2) EP4SGX70, EP4SGX110, EP4SGX180, and EP4SGX230 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

(5) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

Figure 7–4. Number of DQS/DQ Groups per Bank in EP4SE230 and EP4SE360 Devices in the 780-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*



Notes to Figure 7–4:

(1) These numbers are preliminary until the devices are available.

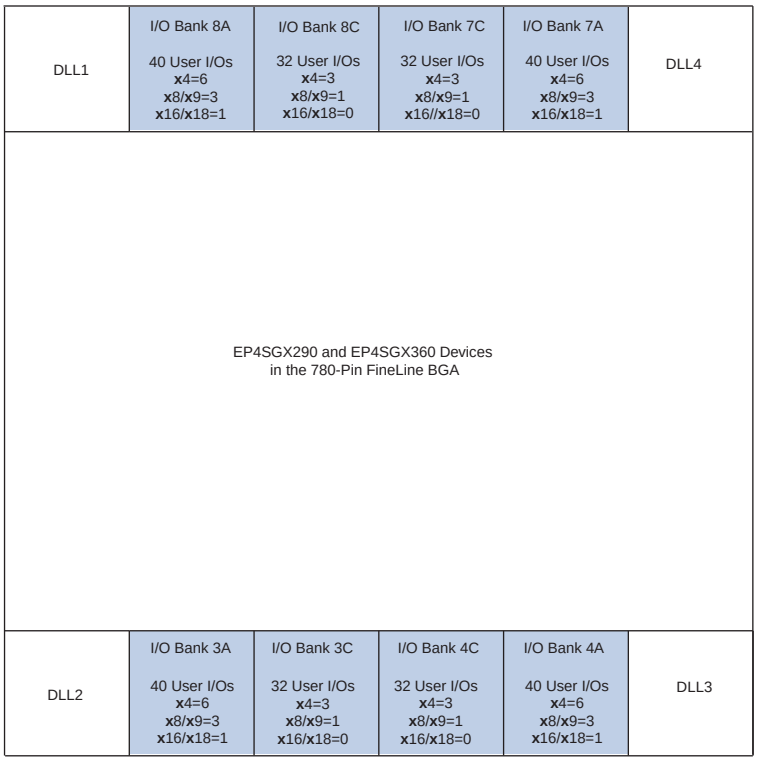
(2) EP4SE230 and EP4SE360 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

(5) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

Figure 7–5. Number of DQS/DQ Groups per Bank in EP4SGX290 and EP4SGX360 Devices in the 780-Pin FineLine BGA Package *(Note 1)*, *(2)*

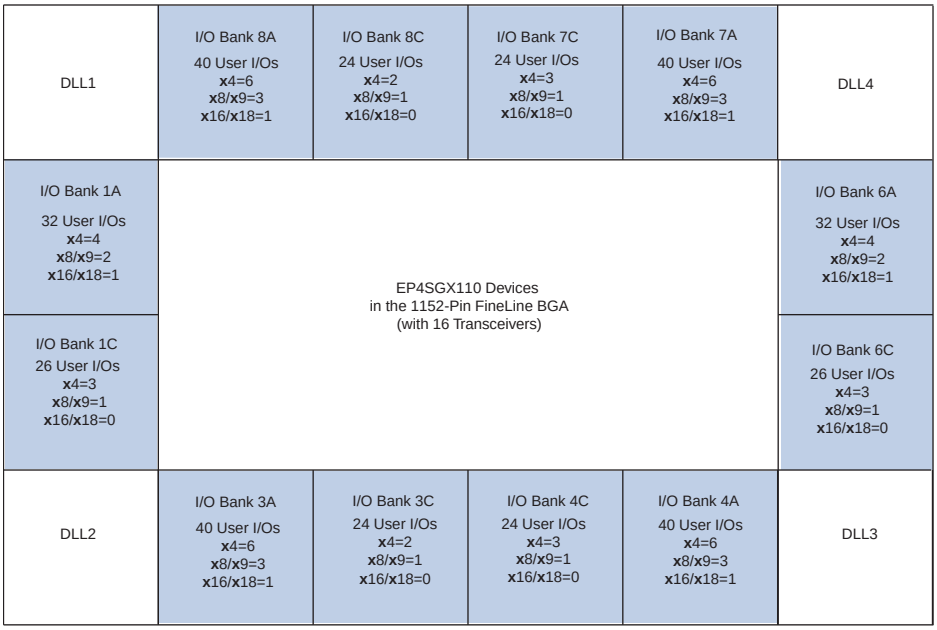


Notes to Figure 7–5:

(1) These numbers are preliminary until the devices are available.

(2) EP4SGX290 and EP4SGX360 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

Figure 7–6. Number of DQS/DQ Groups per Bank in EP4SGX110 Devices with 16 Transceivers in the 1152-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*



Notes to Figure 7–6:

(1) These numbers are preliminary until the devices are available.

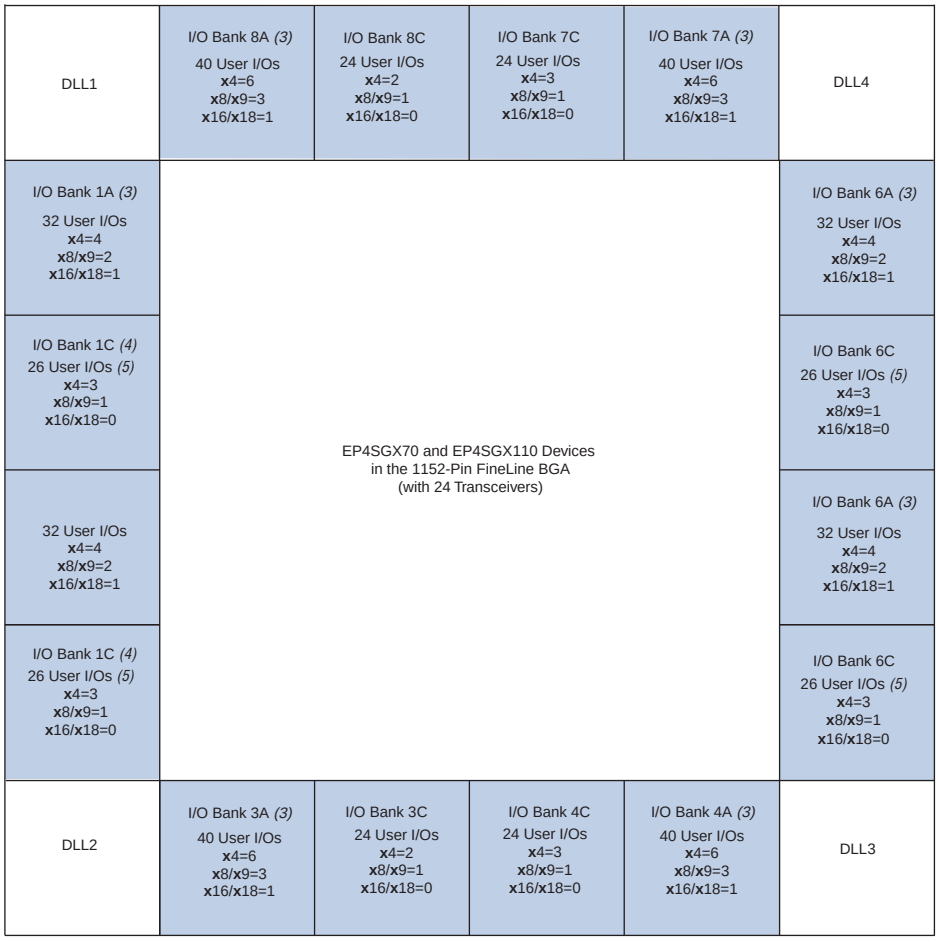
(2) EP4SGX110 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

(5) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

Figure 7–7. Number of DQS/DQ Groups per Bank in EP4SGX70 and EP4SGX110 Devices with 24 Transceivers in the 1152-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*



Notes to Figure 7–7:

(1) These numbers are preliminary until the devices are available.

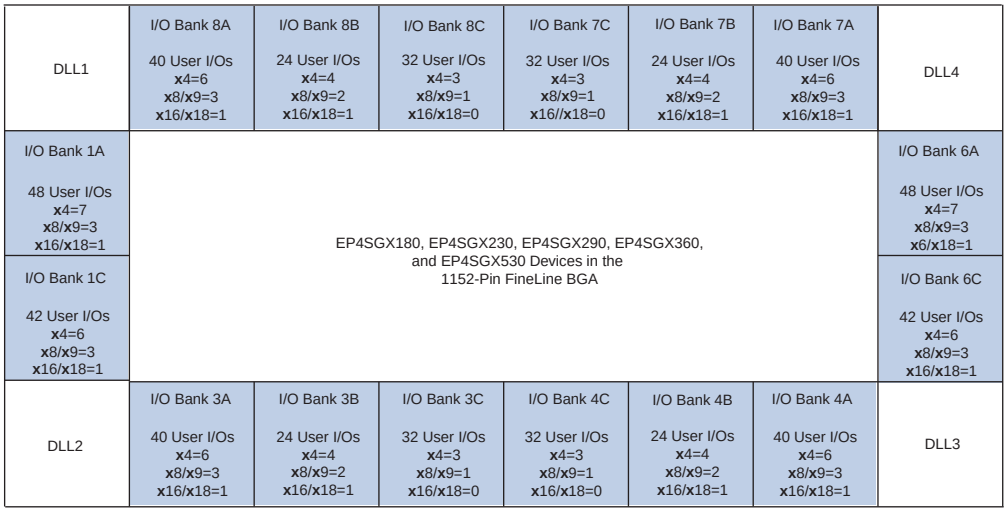
(2) EP4SGX70 and EP4SGX110 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

(5) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–8. Number of DQS/DQ Groups per Bank in EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1152-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*



Notes to Figure 7–8:

(1) These numbers are preliminary until the devices are available.

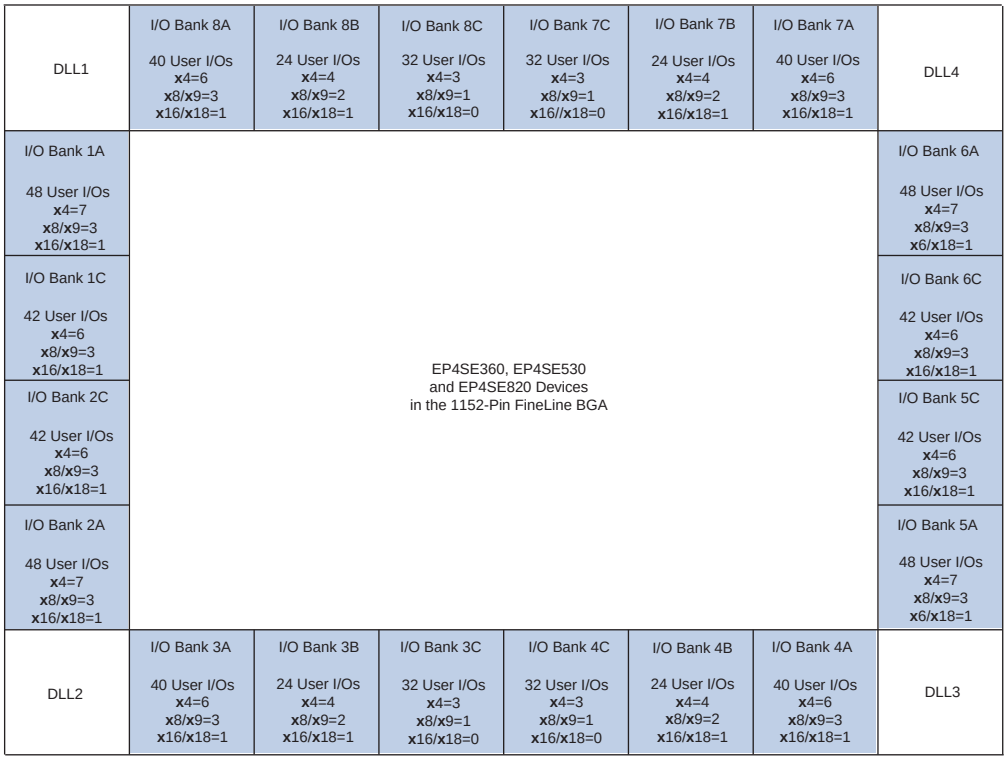
(2) EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

(5) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–9. Number of DQS/DQ Groups per Bank in EP4SE360, EP4SE530, and EPSE820 Devices in the 1152-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*



Notes to Figure 7–9:

(1) These numbers are preliminary until the devices are available.

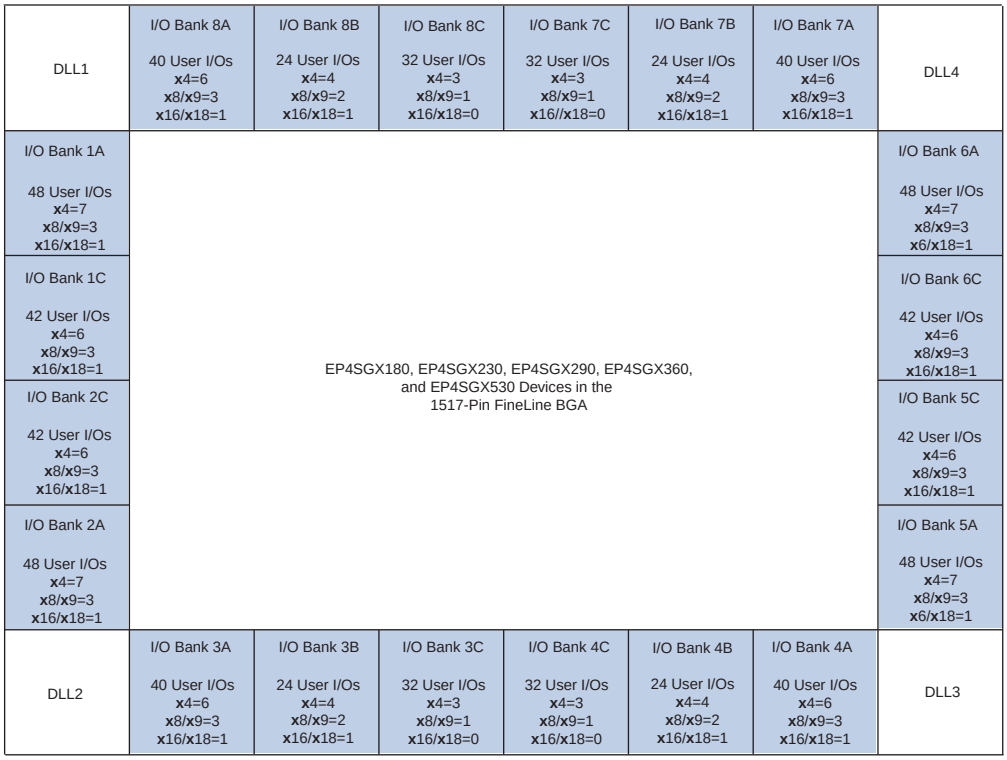
(2) EP4SE360, EP4SE530, and EPSE820 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

(5) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

Figure 7–10. Number of DQS/DQ Groups per Bank in EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1517-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*



Notes to Figure 7–10:

(1) These numbers are preliminary until the devices are available.

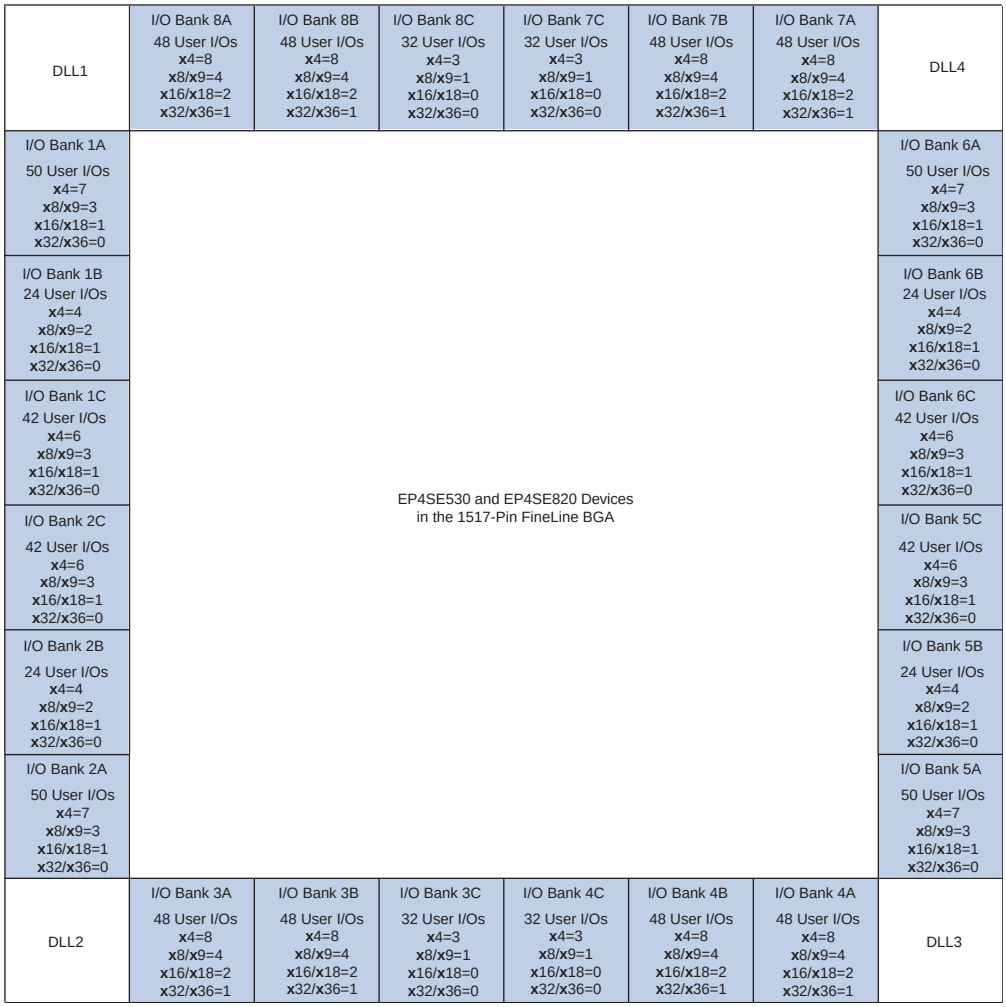
(2) EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 devices do not support ×32/×36 mode. To interface with a ×36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

(5) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–11. Number of DQS/DQ Groups per Bank in EP4SE530 and EP4SE820 Devices in the 1517-pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*



Notes to Figure 7–11:

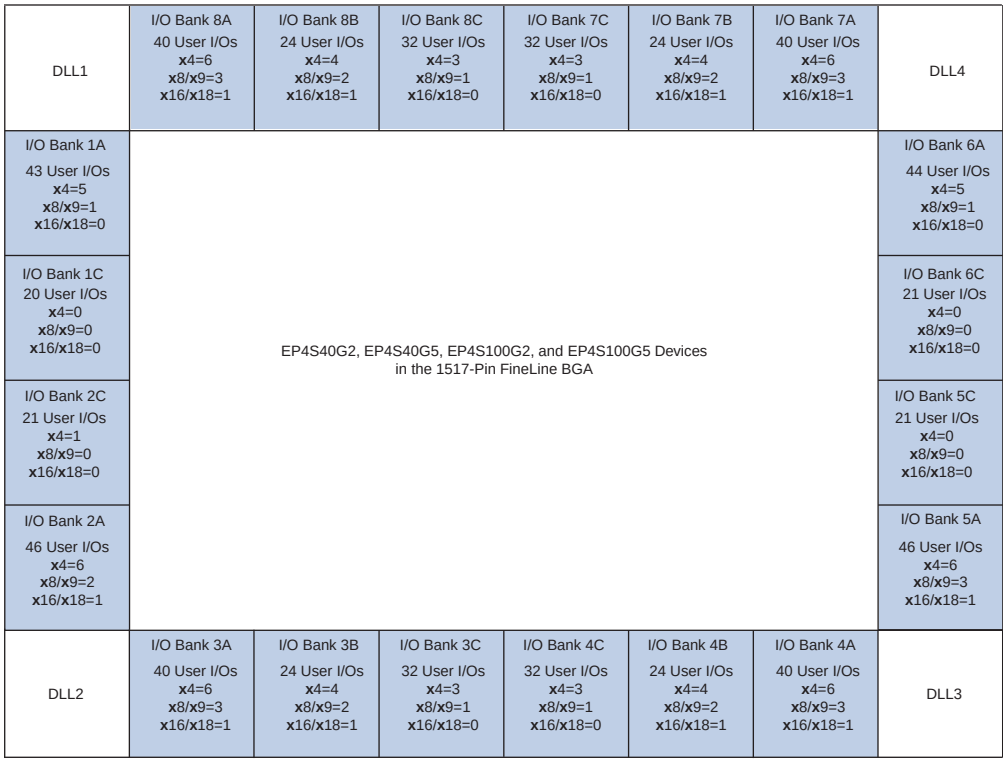
(1) These numbers are preliminary until the devices are available.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(3) All I/O pin counts include dedicated clock inputs and dedicated corner PLL clock inputs that you can use for data inputs.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–12. Number of DQS/DQ Groups per Bank in EP4S40G2, EP4S40G5, EP4S100G2, and EP4S100G5 Devices in the 1517-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*



Notes to Figure 7–12:

(1) These numbers are preliminary until the devices are available.

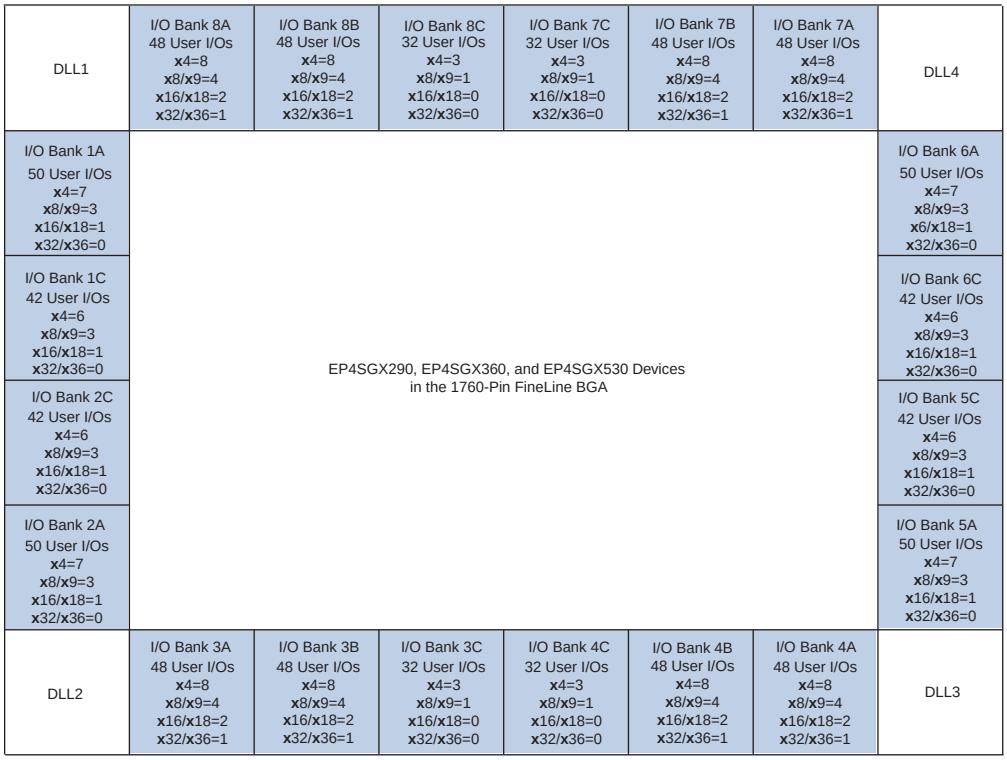
(2) EP4S40G2, EP4S40G5, EP4S100G2, and EP4S100G5 devices do not support ×32/× 36 mode. To interface with a × 36 QDRII+/QDRII SRAM device, refer to "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" on page 7–24.

(3) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(4) All I/O pin counts include dedicated clock inputs that you can use for data inputs.

(5) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a × 4 DQS/DQ group with any of its pin members used for configuration purposes. Make sure that the DQS/DQ groups that you have chosen are not used for configuration as you may lose up to four × 4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–13. Number of DQS/DQ Groups per Bank in EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1760-Pin FineLine BGA Package *(Note 1), (2), (3), (4)*



Notes to Figure 7–13:

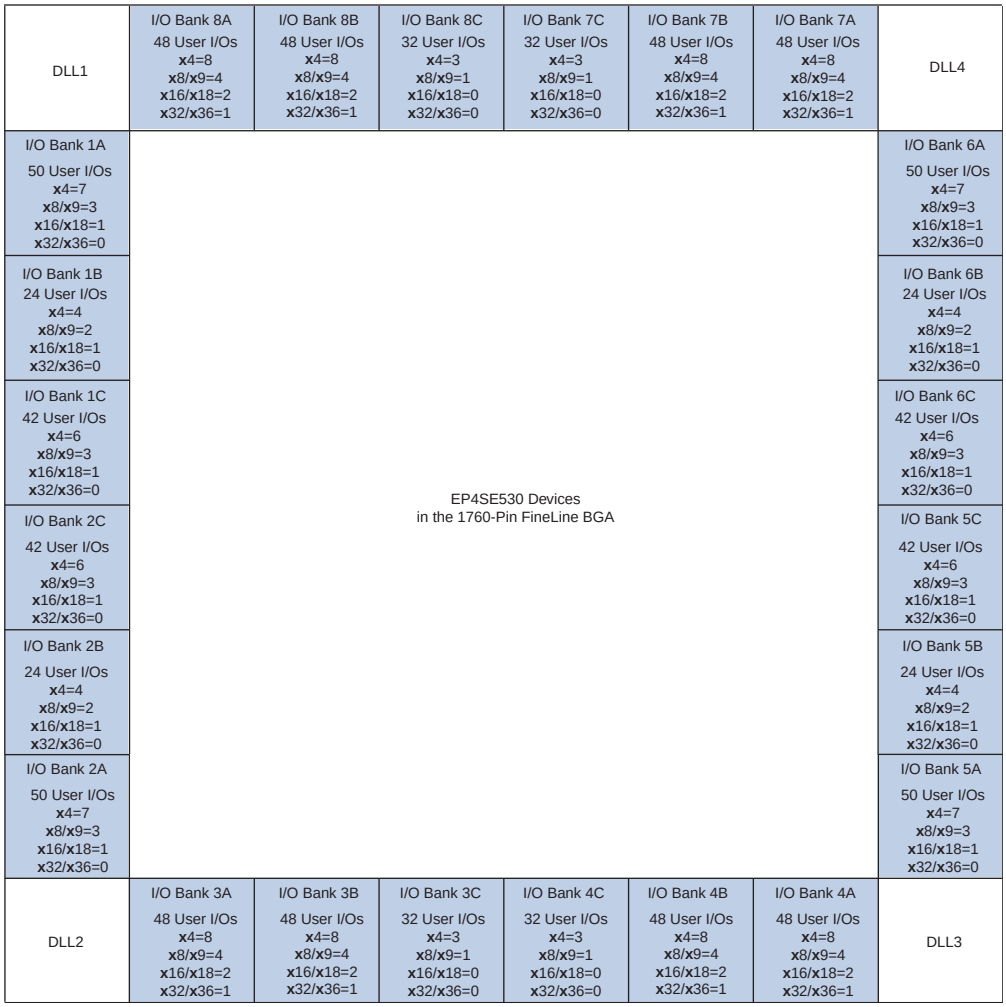
(1) These numbers are preliminary until the devices are available.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group..

(3) All I/O pin counts include dedicated clock inputs and dedicated corner PLL clock inputs that you can use for data inputs.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–14. Number of DQS/DQ Groups per Bank in EP4SE530 Devices in the 1760-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*



Notes to Figure 7–14:

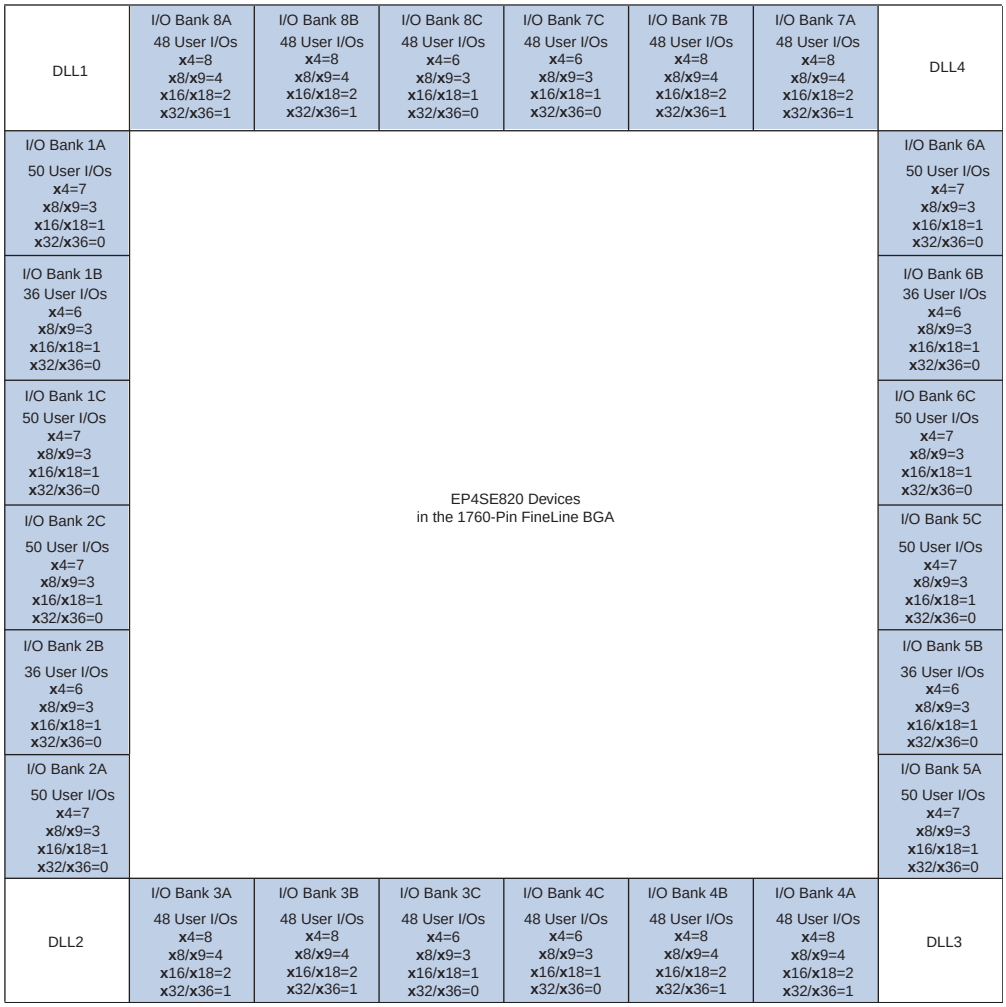
(1) These numbers are preliminary until the devices are available.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(3) All I/O pin counts include dedicated clock inputs and dedicated corner PLL clock inputs that you can use for data inputs.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–15. Number of DQS/DQ Groups per Bank in EP4SE820 Devices in the 1760-pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*



Notes to Figure 7–15:

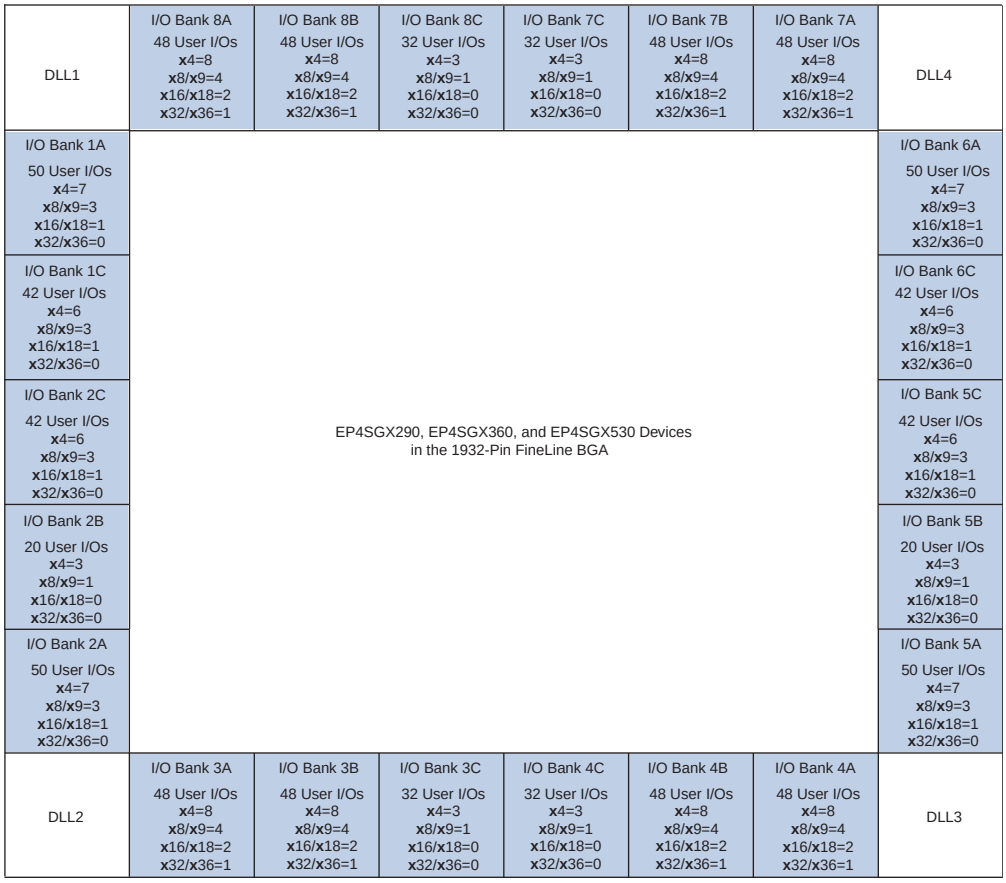
(1) These numbers are preliminary until the devices are available.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(3) All I/O pin counts include dedicated clock inputs and dedicated corner PLL clock inputs that you can use for data inputs.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–16. Number of DQS/DQ Groups per Bank in EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1932-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*



Notes to Figure 7–16:

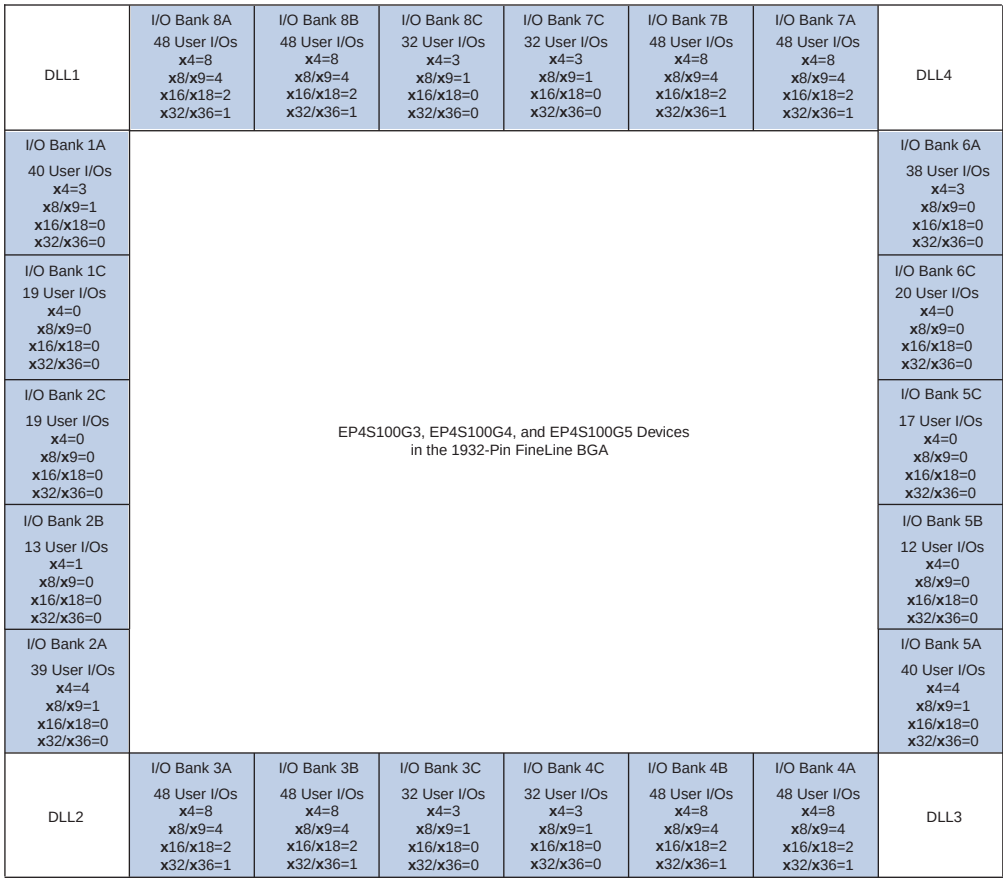
(1) These numbers are preliminary until the devices are available.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(3) All I/O pin counts include dedicated clock inputs and dedicated corner PLL clock inputs that you can use for data inputs.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

Figure 7–17. Number of DQS/DQ Groups per Bank in EP4S100G3, EP4S100G4, and EP4S100G5 Devices in the 1932-Pin FineLine BGA Package *(Note 1)*, *(2)*, *(3)*, *(4)*



Notes to Figure 7–17:

(1) These numbers are preliminary until the devices are available.

(2) You can also use DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins, but you cannot use a ×4 group for memory interfaces if two pins of the ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration. If two pins of a ×4 group are used as R_{UP} and R_{DN} pins for OCT calibration, you can use the ×16/×18 or ×32/×36 groups that include that ×4 group, however there are restrictions on using ×8/×9 groups that include that ×4 group.

(3) All I/O pin counts include dedicated clock inputs and dedicated corner PLL clock inputs that you can use for data inputs.

(4) You can also use some of the DQS/DQ pins in I/O Bank 1C as configuration pins. You cannot use a ×4 DQS/DQ group with any of its pin members used for configuration purposes. Ensure that the DQS/DQ groups that you have chosen are not also used for configuration because you may lose up to four ×4 DQS/DQ groups, depending on your configuration scheme.

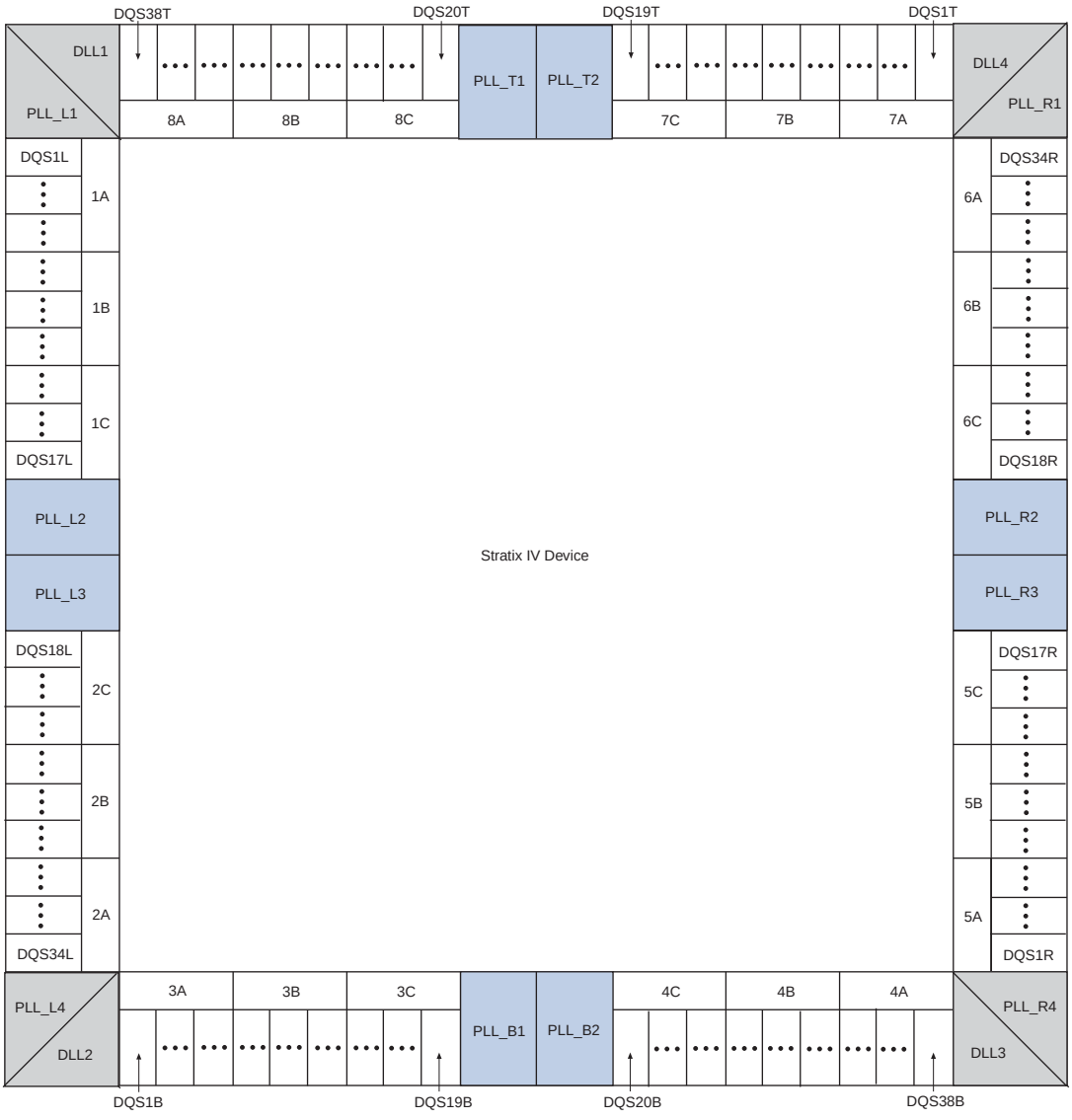
The DQS and DQSn pins are listed in the Stratix IV pin tables as DQSXY and DQSnXY, respectively, where X indicates the DQS/DQ grouping number and Y indicates whether the group is located on the top (T), bottom (B), left (L), or right (R) side of the device. The DQS/DQ pin numbering is based on ×4 mode.

The corresponding DQ pins are marked as DQXY, where X indicates which DQS group the pins belong to and Y indicates whether the group is located on the top (T), bottom (B), left (L), or right (R) side of the device. For example, DQS1L indicates a DQS pin located on the left side of the device. The DQ pins belonging to that group are shown as DQ1L in the pin table. For more information, refer to Figure 7–18.

The parity, DM, BWSn, NWSn, ECC, and QVLD pins are shown as DQ pins in the pin table.

The numbering scheme starts from the top-left side of the device going counter-clockwise in a die-top view. Figure 7–18 shows how the DQS/DQ groups are numbered in a die-top view of the device. The top and bottom sides of the device can contain up to 38 ×4 DQS/DQ groups. The left and right sides of the device can contain up to 34 ×4 DQS/DQ groups.

Figure 7–18. DQS Pins in Stratix IV I/O Banks



Using the R_{UP} and R_{DN} Pins in a DQS/DQ Group Used for Memory Interfaces

You can use the DQS/DQSn pins in some of the ×4 groups as R_{UP} and R_{DN} pins (listed in the pin table). You cannot use a ×4 DQS/DQ group for memory interfaces if any of its pin members are used as R_{UP} and R_{DN} pins for OCT calibration. You may be able to use the ×8/×9 group that includes this ×4 DQS/DQ group, if either of the following applies:

- You are not using DM pins with your differential DQS pins
- You are not using complementary or differential DQS pins

You can use the ×8/×9 group because a DQS/DQ ×8/×9 group actually comprises 12 pins, as the groups are formed by stitching two DQS/DQ groups in ×4 mode with six pins each (refer to Table 7–2 on page 7–6). A typical ×8 memory interface consists of one DQS, one DM, and eight DQ pins that add up to 10 pins. If you choose your pin assignment carefully, you can use the two extra pins for R_{UP} and R_{DN}. In a DDR3 SDRAM interface, you must use differential DQS, which means that you only have one extra pin. In this case, pick different pin locations for the R_{UP} and R_{DN} pins (for example, in the bank that contains the address and command pins).

You cannot use the R_{UP} and R_{DN} pins shared with DQS/DQ group pins when using ×9 QDRII+/QDRII SRAM devices, as the R_{UP} and R_{DN} pins are dual purpose with the CQn pins. In this case, pick different pin locations for R_{UP} and R_{DN} pins to avoid conflict with memory interface pin placement. In this case, you have the choice of placing the R_{UP} and R_{DN} pins in the data-write group or in the same bank as the address and command pins.

There is no restriction of using ×16/×18 or ×32/×36 DQS/DQ groups that include the ×4 groups whose pin members are being used as R_{UP} and R_{DN} pins as there are enough extra pins that can be used as DQS pins.

For ×8, ×16/×18, or ×32/×36 DQS/DQ groups whose members are used for R_{UP} and R_{DN}, you must assign DQS and DQ pins manually. The Quartus® II software might not be able to place DQS and DQ pins without manual pin assignments, resulting in a "no-fit".

Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface

This implementation combines ×16/×18 DQS/DQ groups to interface with a ×36 QDRII+/QDRII SRAM device. The ×36 read data bus uses two ×16/×18 groups while the ×36 write data uses another two ×16/×18 or four ×8/×9 groups. The CQ/CQn signal traces are split on the board trace to connect to two pairs of CQ/CQn pins in the FPGA. This is the only connection on the board that you need to change for this implementation. Other QDRII+/QDRII SRAM interface rules for Stratix IV devices also apply for this implementation.

The ALTMEMPHY megafunction and UniPHY-based external memory interface IPs do not use the QVLD signal, so you can leave the QVLD signal unconnected as in any QDRII+/QDRII SRAM interfaces.

For more information about the ALTMEMPHY megafunction or UniPHY-based IPs, refer to the *External Memory Interface Handbook*.

Rules to Combine Groups

In 780-, 1152-, and some 1517-pin package devices, there is at most one ×16/×18 group per I/O sub-bank. You can combine two ×16/×18 groups from a single side of the device for a ×36 interface.

For devices that do not have four ×16/×18 groups in a single side of the device to form two ×36 groups for read and write data, you can form one ×36 group on one side of the device and another ×36 group on the other side of the device.

For vertical migration with the ×36 emulation implementation, check if migration is possible by enabling device migration in the Quartus II project. The Quartus II software supports the use of four ×8/×9 DQ groups for write data pins and migration of these groups across device density. Table 7–4 lists the possible combinations to use two ×16/×18 DQS/DQ groups to form a ×32/×36 group on Stratix IV devices lacking a native ×32/×36 DQS/DQ group.

Table 7–4. Possible Group Combinations in Stratix IV Devices (Part 1 of 2)

Package	Device Density	I/O Sub-Bank Combinations
780-Pin FineLine BGA	■ EP4SGX70 ■ EP4SGX110 ■ EP4SGX180 ■ EP4SGX230 ■ EP4SGX290 ■ EP4SGX360	3A and 4A, 7A and 8A (bottom and top I/O banks) *(1)*
	■ EP4SE230 ■ EP4SE360	1A and 2A, 5A and 6A (left and right I/O banks) 3A and 4A, 7A and 8A (bottom and top I/O banks) *(1)*
1152-Pin FineLine BGA	■ EP4SGX70 ■ EP4SGX110	3A and 4A, 7A and 8A (bottom and top I/O banks) *(1)*
	■ EP4SGX180 ■ EP4SGX230 ■ EP4SGX290 ■ EP4SGX360 ■ EP4SGX530	1A and 1C, 6A and 6C (left and right I/O banks) 3A and 3B, 4A and 4B (bottom I/O banks) 7A and 7B, 8A and 8B (top I/O banks)
	■ EP4SE360 ■ EP4SE530 ■ EP4SE820	1A and 1C, 2A and 2C (left I/O banks) 3A and 3B, 4A and 4B (bottom I/O banks) 5A and 5C, 6A and 6C (right I/O banks) 7A and 7B, 8A and 8B (top I/O banks)

Table 7–4. Possible Group Combinations in Stratix IV Devices (Part 2 of 2)

Package	Device Density	I/O Sub-Bank Combinations
1517-Pin FineLine BGA	■ EP4SGX180 ■ EP4SGX230 ■ EP4SGX290 ■ EP4SGX360 ■ EP4SGX530	1A and 1C, 2A and 2C (left I/O banks) 3A and 3B, 4A and 4B (bottom I/O banks) 5A and 5C, 6A and 6C (right I/O banks) 7A and 7B, 8A and 8B (top I/O banks)
	■ EP4SE530 *(2)* ■ EP4SE820 *(2)*	1A and 1B, 2A and 2B or 1B and 1C, 2B and 2C (left I/O banks) *(3)* 5A and 5B, 6A and 6B or 5B and 5C, 6B and 6C (right I/O banks) *(3)*
	■ EP4S40G2 ■ EP4S40G5 ■ EP4S100G2 ■ EP4S100G5	3A and 3B, 4A and 4B (bottom I/O banks) 7A and 7B, 8A and 8B (top I/O banks)
1760-Pin FineLine BGA	■ EP4SGX290 ■ EP4SGX360 ■ EP4SGX530	1A and 1C, 2A and 2C (left I/O banks) 3A and 3B, 4A and 4B (bottom I/O banks) 5A and 5C, 6A and 6C (right I/O banks) 7A and 7B, 8A and 8B (top I/O banks)
	■ EP4SE530 *(2)* ■ EP4SE820 *(2)*	1A and 1B, 2A and 2B or 1B and 1C, 2B and 2C (left I/O banks) *(3)* 5A and 5B, 6A and 6B or 5B and 5C, 6B and 6C (right I/O banks) *(3)*
1932-Pin FineLine BGA	■ EP4SGX290 *(2)* ■ EP4SGX360 *(2)* ■ EP4SGX530 *(2)*	1A and 1C, 2A and 2C (left I/O banks) 5A and 5C, 6A and 6C (right I/O banks)

Notes to Table 7–4:

(1) Each side of the device in these packages has four remaining ×8/×9 groups. You can combine them for the write side (only) if you want to keep the ×36 QDRII+/QDRII SRAM interface on one side of the device. You must change the **Memory Interface Data Group** default assignment from the default **18** to **9** in this case.

(2) This device supports ×36 DQS/DQ groups on the top and bottom I/O banks natively.

(3) Although it is possible to combine the ×16/×18 DQS/DQ groups from I/O banks 1A and 1C, 2A and 2C, 5A and 5C, and 6A and 6C, Altera does not recommend this due to the size of the package. Similarly, crossing a bank number (for example, combining groups from I/O banks 6C and 5C) is not supported in this package.

Stratix IV External Memory Interface Features

Stratix IV devices are rich with features that allow robust high-performance external memory interfacing. The ALTMEMPHY megafunction allows you to use these external memory interface features and helps set up the physical interface (PHY) best suited for your system. This section describes each Stratix IV device feature that is used in external memory interfaces from the DQS phase-shift circuitry, DQS logic block, leveling multiplexers, and dynamic OCT control block.

The ALTMEMPHY megafunction and the Altera® memory controller MegaCore® functions can run at half the frequency of the I/O interface of the memory devices to allow better timing management in high-speed memory interfaces. Stratix IV devices have built-in registers in the IOE to convert data from full-rate (the I/O frequency) to half-rate (the controller frequency) and vice versa. You can bypass these registers if your memory controller is not running at half the rate of the I/O frequency. When using the Altera memory controller MegaCore functions, the ALTMEMPHY megafunction is instantiated for you.

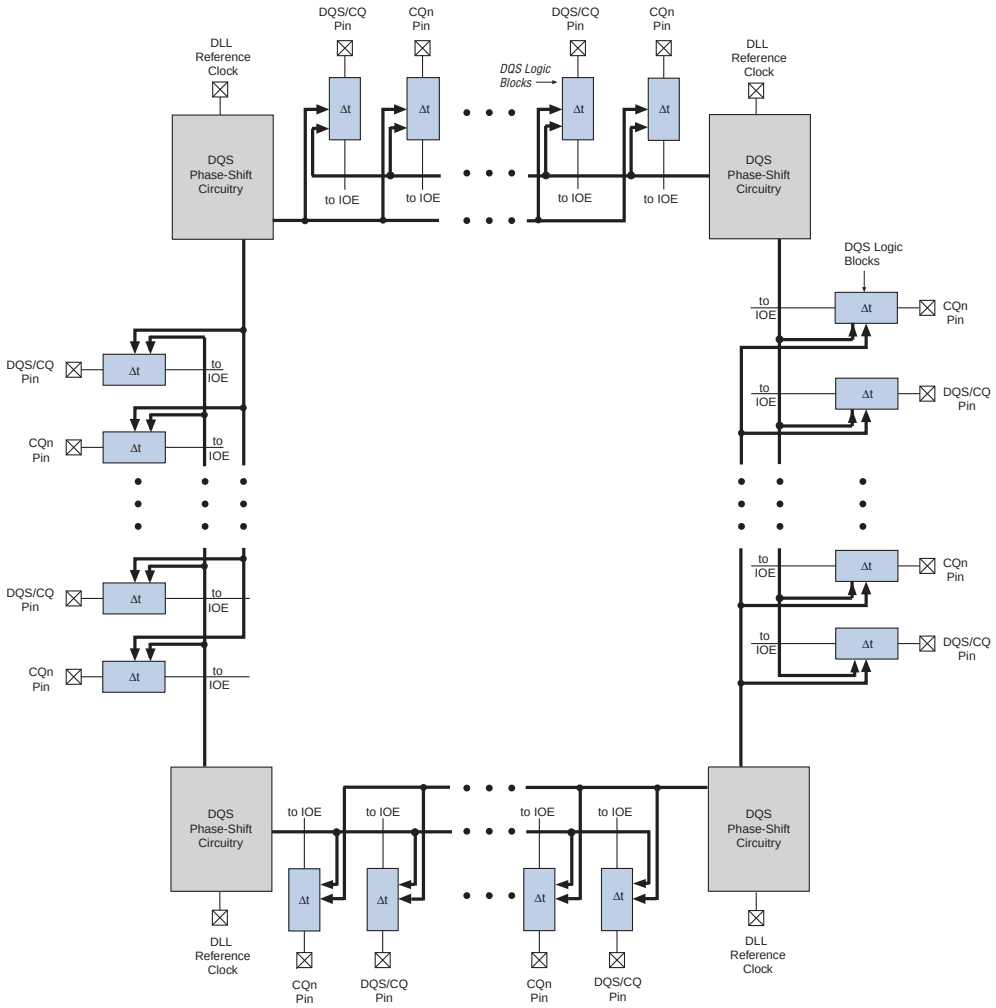
For more information about the ALTMEMPHY megafunction, refer to the *External Memory PHY Interface Megafunction User Guide (ALTMEMPHY)*.

DQS Phase-Shift Circuitry

Stratix IV phase-shift circuitry provides phase shift to the DQS/CQ and CQn pins on read transactions when the DQS/CQ and CQn pins are acting as input clocks or strobes to the FPGA. The DQS phase-shift circuitry consists of DLLs that are shared between multiple DQS pins and the phase-offset module to further fine-tune the DQS phase shift for different sides of the device.

Figure 7–19 shows how the DQS phase-shift circuitry is connected to the DQS/CQ and CQn pins in the device where memory interfaces are supported on all sides of the Stratix IV device.

Figure 7–19. DQS/CQ and CQn Pins and DQS Phase-Shift Circuitry *(Note 1)*, *(2)*



Notes to Figure 7–19:

(1) For possible reference input clock pins for each DLL, refer to "DLL" on page 7–28.

(2) You can configure each DQS/CQ and CQn pin with a phase shift based on one of two possible DLL output settings.

DQS phase-shift circuitry is connected to the DQS logic blocks that control each DQS/CQ or CQn pin. The DQS logic blocks allow the DQS delay settings to be updated concurrently at every DQS/CQ or CQn pin.

DLL

DQS phase-shift circuitry uses a DLL to dynamically control the clock delay needed by the DQS/CQ and CQn pin. The DLL, in turn, uses a frequency reference to dynamically generate control signals for the delay chains in each of the DQS/CQ and CQn pins, allowing it to compensate for PVT variations. The DQS delay settings are Gray-coded to reduce jitter when the DLL updates the settings. The phase-shift circuitry needs 1280 clock cycles to lock and calculate the correct input clock period

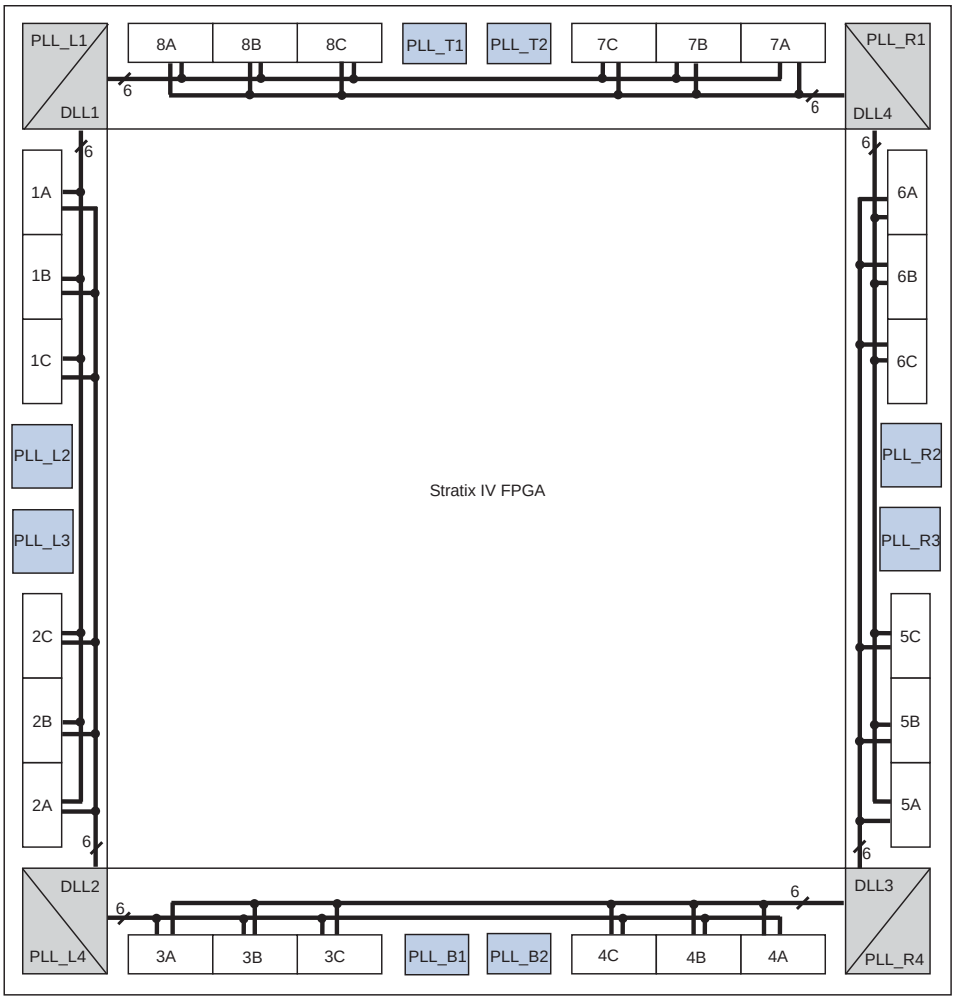
when the DLL is in low jitter mode. Otherwise, only 256 clock cycles are needed. Do not send data during these clock cycles because there is no guarantee that it will be captured properly. As the settings from the DLL may not be stable until this lock period has elapsed, be aware that anything using these settings (including the leveling delay system) may be unstable during this period.

You can still use the DQS phase-shift circuitry for any memory interfaces that are less than 100 MHz. However, the DQS signal may not shift over 2.5 ns. Even if the DQS signal is not shifted exactly to the middle of the DQ valid window, the I/O element should still be able to capture the data in low-frequency applications in which a large amount of timing margin is available.

There are a maximum of four DLLs in a Stratix IV device, located in each corner of the device. These four DLLs support a maximum of four unique frequencies, with each DLL running at one frequency. Each DLL can have two outputs with different phase offsets, which allows one Stratix IV device to have eight different DLL phase shift settings.

Figure 7–20 shows the DLL and I/O bank locations in Stratix IV devices from a die-top view if all sides of the device support external memory interfaces.

Figure 7–20. Stratix IV DLL and I/O Bank Locations (Die-Top View)



The DLL can access the two adjacent sides from its location within the device. For example, DLL1 on the top left of the device can access the top side (I/O banks 7A, 7B, 7C, 8A, 8B, and 8C) and the left side of the device (I/O banks 1A, 1B, 1C, 2A, 2B, and 2C). This means that each I/O bank is accessible by two DLLs, giving more flexibility to create multiple frequencies and multiple-type interfaces. You can have two different interfaces with the same frequency on the two sides adjacent to a DLL, where the DLL controls the DQS delay settings for both interfaces.

Each bank can use settings from either or both DLLs the bank is adjacent to. For example, `DQS1L` can get its phase-shift settings from DLL1, while `DQS2L` can get its phase-shift settings from DLL2. Table 7–5 lists the DLL location and supported I/O banks for Stratix IV devices.

You can only have one memory interface in each I/O sub-bank (such as I/O sub-banks 1A, 1B, and 1C) when you use leveling delay chains. This is because there is only one leveling delay chain per I/O sub-bank.

Table 7–5. DLL Location and Supported I/O Banks

DLL	Location	Accessible I/O Banks *(1)*
DLL1	Top-left corner	1A, 1B, 1C, 2A, 2B, 2C, 7A, 7B, 7C, 8A, 8B, 8C
DLL2	Bottom-left corner	1A, 1B, 1C, 2A, 2B, 2C, 3A, 3B, 3C, 4A, 4B, 4C
DLL3	Bottom-right corner	3A, 3B, 3C, 4A, 4B, 4C, 5A, 5B, 5C, 6A, 6B, 6C
DLL4	Top-right corner	5A, 5B, 5C, 6A, 6B, 6C, 7A, 7B, 7C, 8A, 8B, 8C

Note to Table 7–5:

(1) The DLL can access these I/O banks if they are available for memory interfacing.

The reference clock for each DLL may come from PLL output clocks or any of the two dedicated clock input pins located in either side of the DLL. Table 7–6 through Table 7–17 show the available DLL reference clock input resources for the Stratix IV device family.

When you have a dedicated PLL that only generates the DLL input reference clock, set the PLL mode to **No Compensation** to achieve better performance or the Quartus II software changes it automatically. Because the PLL does not use any other outputs, it does not need to compensate for any clock paths.

Table 7–6. DLL Reference Clock Input for EP4SGX70, EP4SGX110, EP4SGX180, and EP4SGX230 Devices in the 780-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	—	—
DLL3	CLK4P CLK5P CLK6P CLK7P	—	PLL_B1	—	—
DLL4	CLK12P CLK13P CLK14P CLK15P	—	PLL_T1	—	—

Table 7–7. DLL Reference Clock Input for EP4SE230 and EP4SE360 Devices in the 780-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L2	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B1	PLL_R2	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T1	PLL_R2	—

Table 7–8. DLL Reference Clock Input for EP4SGX290 and EP4SGX360 Devices in the 780-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	—	PLL_T1	—	—
DLL2	CLK4P CLK5P CLK6P CLK7P	—	PLL_B1	—	—
DLL3	CLK4P CLK5P CLK6P CLK7P	—	PLL_B2	—	—
DLL4	CLK12P CLK13P CLK14P CLK15P	—	PLL_T2	—	—

Table 7–9. DLL Reference Clock Input for EP4SGX70 and EP4SGX110 Devices in the 1152-Pin FineLine BGA Package (with 24 Transceivers)

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L2	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B1	PLL_R2	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T1	PLL_R2	—

Table 7–10. DLL Reference Clock Input for EP4SGX110 Devices in the 1152-Pin FineLine BGA Package (with 16 Transceivers)

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P	PLL_B1	—	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK10P CLK11P	PLL_B1	—	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK10P CLK11P	PLL_T1	PLL_R2	—

Table 7–11. DLL Reference Clock Input for EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1152-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P	PLL_B1	—	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK10P CLK11P	PLL_B2	—	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK10P CLK11P	PLL_T2	PLL_R2	—

Table 7–12. DLL Reference Clock Input for EP4SE530 and EPSE820 Devices in the 1152-, 1517-, and 1760-Pin FineLine BGA Packages

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L3	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B2	PLL_R3	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T2	PLL_R2	—

Table 7–13. DLL Reference Clock Input for EP4SGX180, EP4SGX230, EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1517-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L3	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B2	PLL_R3	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T2	PLL_R2	—

Table 7–14. DLL Reference Clock Input for EP4S40G2, EP4S40G5, EP4S100G2, and EP4S100G5 Devices in the 1517-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK1P CLK3P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK1P CLK3P	PLL_B1	PLL_L3	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK10P	PLL_B2	PLL_R3	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK10P	PLL_T2	PLL_R2	—

Table 7–15. DLL Reference Clock Input for EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1760-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2	—
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L3	—
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B2	PLL_R3	—
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T2	PLL_R2	—

Table 7–16. DLL Reference Clock Input for EP4SGX290, EP4SGX360, and EP4SGX530 Devices in the 1932-Pin FineLine BGA Package

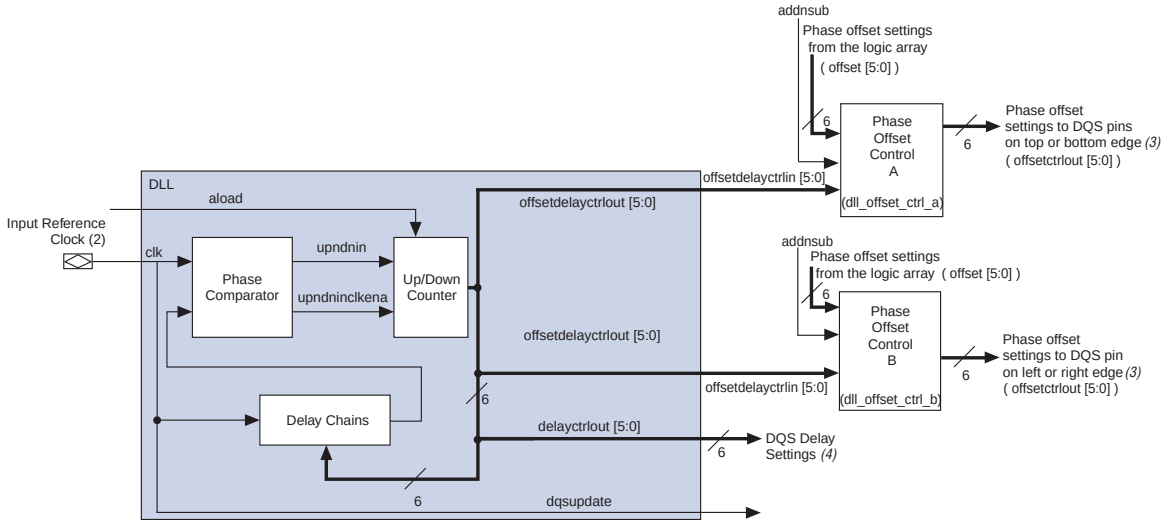
DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	CLK0P CLK1P CLK2P CLK3P	PLL_T1	PLL_L2	PLL_L1
DLL2	CLK4P CLK5P CLK6P CLK7P	CLK0P CLK1P CLK2P CLK3P	PLL_B1	PLL_L3	PLL_L4
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P	PLL_B2	PLL_R3	PLL_R4
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK8P CLK9P CLK10P CLK11P	PLL_T2	PLL_R2	PLL_R1

Table 7–17. DLL Reference Clock Input for EP4S100G3, EP4S100G4, and EP4S100G5 Devices in the 1932-Pin FineLine BGA Package

DLL	CLKIN (Top/Bottom)	CLKIN (Left/Right)	PLL (Top/Bottom)	PLL (Left/Right)	PLL (Corner)
DLL1	CLK12P CLK13P CLK14P CLK15P	—	PLL_T1	PLL_L2	PLL_L1
DLL2	CLK4P CLK5P CLK6P CLK7P	—	PLL_B1	PLL_L3	PLL_L4
DLL3	CLK4P CLK5P CLK6P CLK7P	CLK9P CLK11P	PLL_B2	PLL_R3	PLL_R4
DLL4	CLK12P CLK13P CLK14P CLK15P	CLK9P CLK11P	PLL_T2	PLL_R2	PLL_R1

Figure 7–21 shows a simple block diagram of the DLL. The input reference clock goes into the DLL to a chain of up to 16 delay elements. The phase comparator compares the signal coming out of the end of the delay chain block to the input reference clock. The phase comparator then issues the upndn signal to the Gray-code counter. This signal increments or decrements a six-bit delay setting (DQS delay settings) that increases or decreases the delay through the delay element chain to bring the input reference clock and the signals coming out of the delay element chain in phase.

Figure 7–21. Simplified Diagram of the DQS Phase-Shift Circuitry *(Note 1)*



Notes to Figure 7–21:

(1) All features of the DQS phase-shift circuitry are accessible from the ALTMEMPHY megafunction in the Quartus II software.

(2) The input reference clock for the DQS phase-shift circuitry can come from a PLL output clock or an input clock pin. For more information, refer to Table 7–6 through Table 7–17.

(3) Phase offset settings can only go to the DQS logic blocks.

(4) DQS delay settings can go to the logic array, DQS logic block, and leveling circuitry.

The phase offset control block 'A' is designated as DLLOFFSETCTRL_<coordinate x>_<coordinate y>_N1 and phase offset control block 'B' is designated as DLLOFFSETCTRL_<coordinate x>_<coordinate y>_N2 in the Quartus II assignment.

You can reset the DLL from either the logic array or a user I/O pin. Each time the DLL is reset, you must wait for 1280 clock cycles for the DLL to lock before you can capture the data properly.

Depending on the DLL frequency mode, the DLL can shift the incoming DQS signals by 0°, 22.5°, 30°, 36°, 45°, 60°, 67.5°, 72°, 90°, 108°, 120°, 135°, 144°, or 180°. The shifted DQS signal is then used as the clock for the DQ IOE input registers.

All DQS/CQ and CQn pins, referenced to the same DLL, can have their input signal phase shifted by a different degree amount but all must be referenced at one particular frequency. For example, you can have a 90° phase shift on `DQS1T` and a 60° phase shift on `DQS2T`, referenced from a 200-MHz clock. Not all phase-shift combinations are supported. The phase shifts on the DQS pins referenced by the same DLL must all be a multiple of 22.5° (up to 90°), 30° (up to 120°), 36° (up to 144°), or 45° (up to 180°).

There are seven different frequency modes for the Stratix IV DLL, as shown in Table 7–18. Each frequency mode provides different phase shift selections. In frequency mode 0, 1, 2, and 3, the 6-bit DQS delay settings vary with PVT to implement the phase-shift delay. In frequency modes 4, 5, and 6, only 5 bits of the DQS delay settings vary with PVT to implement the phase-shift delay; the most significant bit of the DQS delay setting is set to 0.

Table 7–18. Stratix IV DLL Frequency Modes

Frequency Mode	Available Phase Shift	Number of Delay Chains
0	22.5, 45, 67.5, 90	16
1	30, 60, 90, 120	12
2	36, 72, 108, 144	10
3	45, 90, 135, 180	8
4	30, 60, 90, 120	12
5	36, 72, 108, 144	10
6	45, 90, 135, 180	8

For the frequency range of each mode, refer to the *DC and Switching Characteristics* chapter.

For 0° shift, the DQS/CQ signal bypasses both the DLL and DQS logic blocks. The Quartus II software automatically sets the DQ input delay chains so that the skew between the DQ and DQS/CQ pin at the DQ IOE registers is negligible when 0° shift is implemented. You can feed the DQS delay settings to the DQS logic block and logic array.

The shifted DQS/CQ signal goes to the DQS bus to clock the IOE input registers of the DQ pins. The signal can also go into the logic array for resynchronization if you are not using IOE resynchronization registers. The shifted CQn signal can only go to the negative-edge input register in the DQ IOE and is only used for QDRII+ and QDRII SRAM interfaces.

Phase Offset Control

Each DLL has two phase-offset modules and can provide two separate DQS delay settings with independent offset, one for the top and bottom I/O bank and one for the left and right I/O bank, so you can fine-tune the DQS phase-shift settings between two different sides of the device. Even though you have independent phase offset control, the frequency of the interface using the same DLL has to be the same. Use the

phase offset control module for making small shifts to the input signal and use the DQS phase-shift circuitry for larger signal shifts. For example, if the DLL only offers a multiple of 30° phase shift, but your interface needs a 67.5° phase shift on the DQS signal, you can use two delay chains in the DQS logic blocks to give you 60° phase shift and use the phase offset control feature to implement the extra 7.5° phase shift.

You can either use a static phase offset or a dynamic phase offset to implement the additional phase shift. The available additional phase shift is implemented in 2s: complement in Gray-code between settings –64 to +63 for frequency mode 0, 1, 2, and 3, and between settings –32 to +31 for frequency modes 4, 5, and 6. An additional bit indicates whether the setting has a positive or negative value. The settings are linear, each phase offset setting adds a delay amount specified in the *DC and Switching Characteristics* chapter. The DQS phase shift is the sum of the DLL delay settings and the user-selected phase offset settings whose top setting is 64 for frequency modes 0, 1, 2, and 3; and 32 for frequency modes 4, 5, and 6, so the actual physical offset setting range is 64 or 32 subtracted by the DQS delay settings from the DLL.

When using this feature, you need to monitor the DQS delay settings to know how many offsets you can add and subtract in the system. Note that the DQS delay settings output by the DLL are also Gray coded.

For example, if the DLL determines that DQS delay settings of 28 is needed to achieve a 30° phase shift in DLL frequency mode 1, you can subtract up to 28 phase offset settings and you can add up to 35 phase offset settings to achieve the optimal delay that you need. However, if the same DQS delay settings of 28 is needed to achieve 30° phase shift in DLL frequency mode 4, you can still subtract up to 28 phase offset settings, but you can only add up to 3 phase offset settings before the DQS delay settings reach their maximum settings because DLL frequency mode 4 only uses 5-bit DLL delay settings.

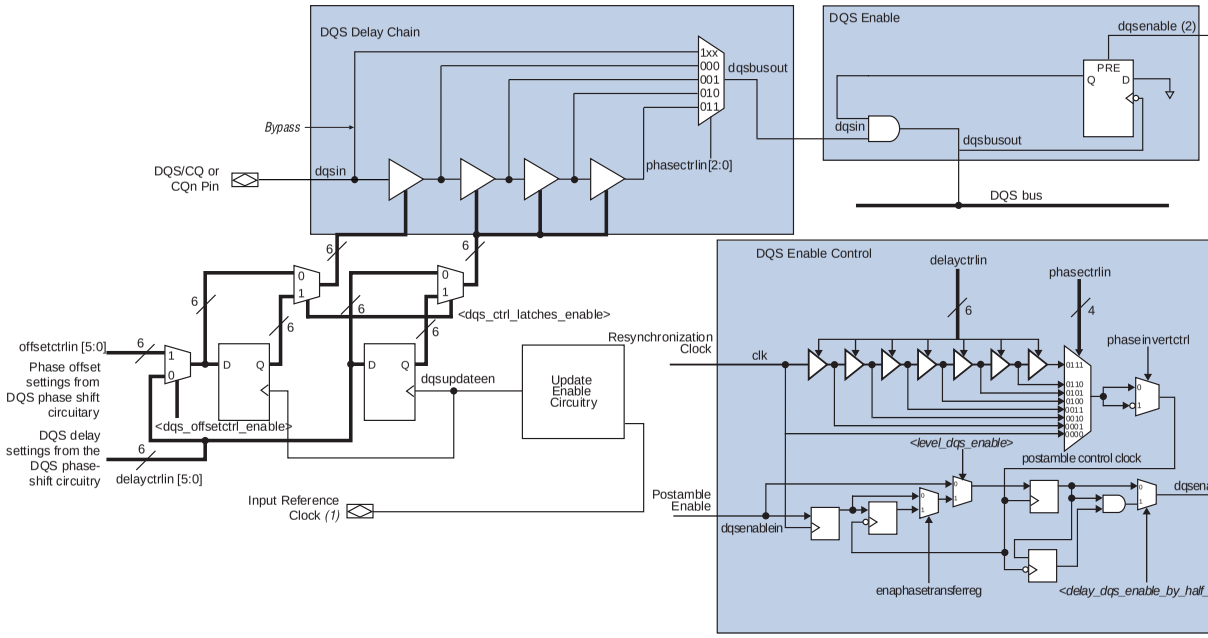
For more information about the value for each step, refer to the *DC and Switching Characteristics* chapter.

When using static phase offset, you can specify the phase offset amount in the ALTMEMPHY megafunction as a positive number for addition or a negative number for subtraction. You can also have a dynamic phase offset that is always added to, subtracted from, or both added to and subtracted from the DLL phase shift. When you always add or subtract, you can dynamically input the phase offset amount into the `dll_offset[5..0]` port. When you want to both add and subtract dynamically, you control the `addnsub` signal in addition to the `dll_offset[5..0]` signals.

DQS Logic Block

Each DQS/CQ and CQn pin is connected to a separate DQS logic block, which consists of the DQS dela circuitry, and DQS postamble circuitry (Figure 7–22).

Figure 7–22. Stratix IV DQS Logic Block



Notes to Figure 7–22:

(1) The input reference clock for the DQS phase-shift circuitry can come from a PLL output clock or an input clock pin. For more information, refer to Table 7–6 through Ta

(2) The dqsenable signal can also come from the Stratix IV FPGA fabric.

DQS Delay Chain

DQS delay chains consist of a set of variable delay elements to allow the input DQS/CQ and CQn signals to be shifted by the amount specified by the DQS phase-shift circuitry or the logic array. There are four delay elements in the DQS delay chain; the first delay chain closest to the DQS/CQ pin can either be shifted by the DQS delay settings or by the sum of the DQS delay setting and the phase-offset setting. The number of delay chains required is transparent because the ALTMEMPHY megafunction automatically sets it when you choose the operating frequency. The DQS delay settings can come from the DQS phase-shift circuitry on either end of the I/O banks or from the logic array.

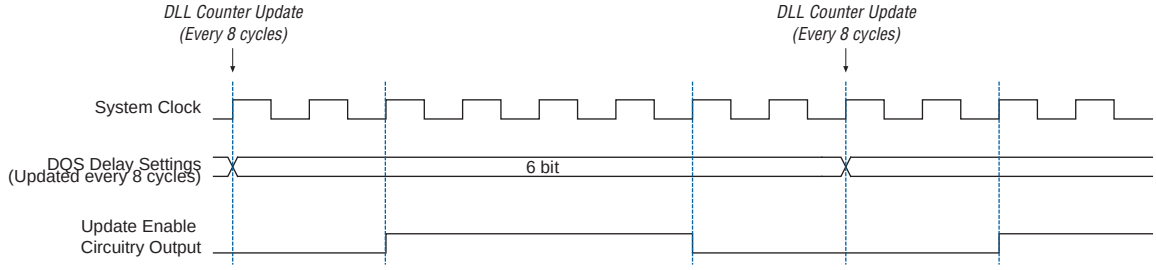
The delay elements in the DQS logic block have the same characteristics as the delay elements in the DLL. When the DLL is not used to control the DQS delay chains, you can input your own Gray-coded 6-bit or 5-bit settings using the `dqs_delayctrlin[5..0]` signals available in the ALTMEMPHY megafunction. These settings control 1, 2, 3, or all 4 delay elements in the DQS delay chains. The ALTMEMPHY megafunction can also dynamically choose the number of DQS delay chains needed for the system. The amount of delay is equal to the sum of the delay element's intrinsic delay and the product of the number of delay steps and the value of the delay steps.

You can also bypass the DQS delay chain to achieve 0° phase shift.

Update Enable Circuitry

Both the DQS delay settings and the phase-offset settings pass through a register before going into the DQS delay chains. The registers are controlled by the update enable circuitry to allow enough time for any changes in the DQS delay setting bits to arrive at all the delay elements. This allows them to be adjusted at the same time. The update enable circuitry enables the registers to allow enough time for the DQS delay settings to travel from the DQS phase-shift circuitry or core logic to all the DQS logic blocks before the next change. It uses the input reference clock or a user clock from the core to generate the update enable output. The ALTMEMPHY megafunction uses this circuit by default. Figure 7–23 shows an example waveform of the update enable circuitry output.

Figure 7–23. DQS Update Enable Waveform



DQS Postamble Circuitry

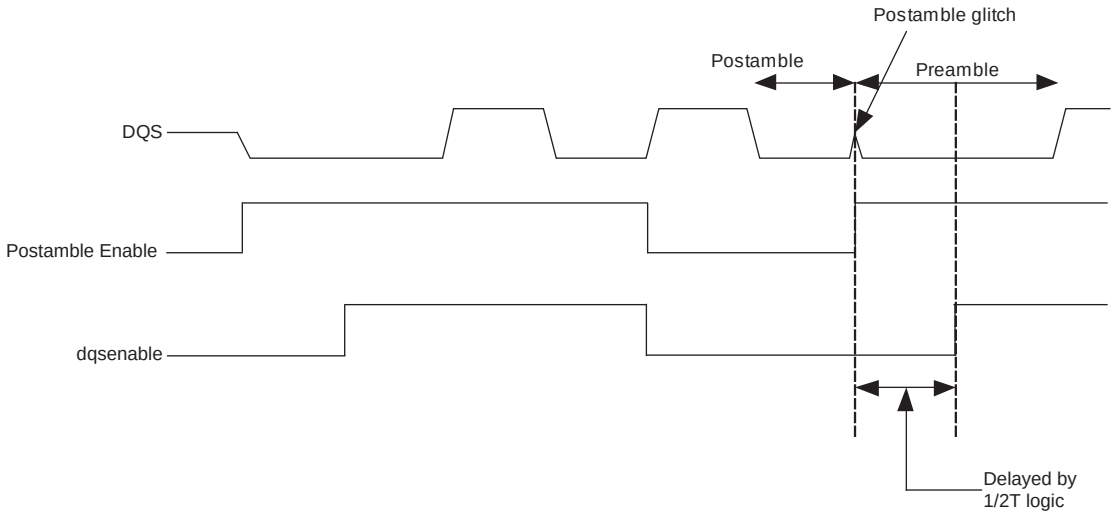
For external memory interfaces that use a bidirectional read strobe such as in DDR3, DDR2, and DDR SDRAM, the DQS signal is low before going to or coming from a high-impedance state. The state in which DQS is low, just after a high-impedance state, is called the preamble; the state in which DQS is low, just before it returns to a high-impedance state, is called the postamble. There are preamble and postamble specifications for both read and write operations in DDR3, DDR2, and DDR SDRAM. The DQS postamble circuitry ensures that data is not lost if there is noise on the DQS line during the end of a read operation that occurs while the DQS is in a postamble state.

Stratix IV devices have dedicated postamble registers that you can control to ground the shifted DQS signal used to clock the DQ input registers at the end of a read operation. This ensures that any glitches on the DQS input signals during the end of a read operation that occurs while the DQS is in a postamble state do not affect the DQ IOE registers.

In addition to the dedicated postamble register, Stratix IV devices also have an HDR block inside the postamble enable circuitry. Use these registers if the controller is running at half the frequency of the I/Os.

Using the HDR block as the first stage capture register in the postamble enable circuitry block is optional. The HDR block is clocked by the half-rate resynchronization clock, which is the output of the I/O clock divider circuit (shown in Figure 7–29 on page 7–47). There is an AND gate after the postamble register outputs that is used to avoid postamble glitches from a previous read burst on a non-consecutive read burst. This scheme allows a half-a-clock cycle latency for `dqsenable` assertion and zero latency for `dqsenable` de-assertion, as shown in Figure 7–24.

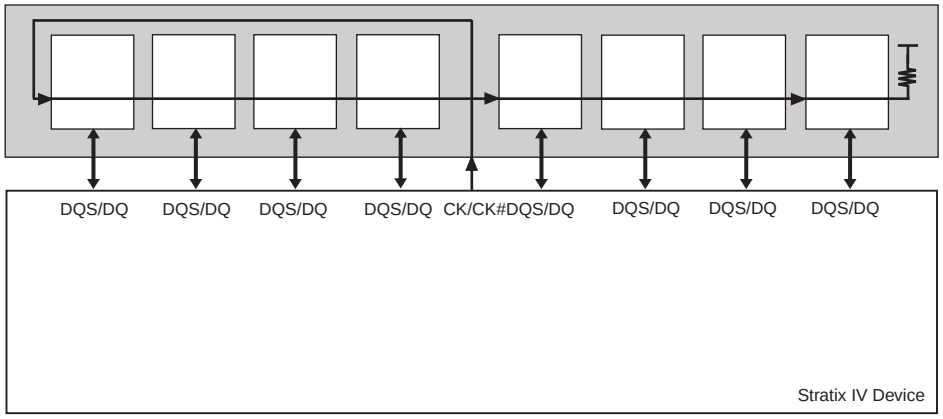
Figure 7–24. Avoiding Glitch on a Non-Consecutive Read Burst Waveform



Leveling Circuitry

DDR3 SDRAM unbuffered modules use a fly-by clock distribution topology for better signal integrity. This means that the CK/CK# signals arrive at each DDR3 SDRAM device in the module at different times. The difference in arrival time between the first DDR3 SDRAM device and the last device on the module can be as long as 1.6 ns. Figure 7–25 shows the clock topology in DDR3 SDRAM unbuffered modules.

Figure 7–25. DDR3 SDRAM Unbuffered Module Clock Topology



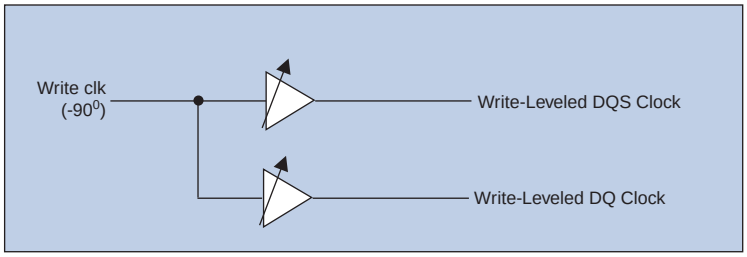
Because the data and read strobe signals are still point-to-point, take special care to ensure that the timing relationship between the CK/CK# and DQS signals (`tDQSS`, `tDSS`, and `tDSH`) during a write is met at every device on the modules. Furthermore, read data coming back into the FPGA from the memory is also staggered in a similar way.

Stratix IV FPGAs have leveling circuitry to address these two situations. There is one leveling circuitry per I/O sub-bank (for example, I/O sub-bank 1A, 1B, and 1C each has one leveling circuitry). These delay chains are PVT-compensated by the same DQS delay settings as the DLL and DQS delay chains.

For frequencies equal to and above 400 MHz, the DLL uses eight delay chains, such that each delay chain generates a 45° delay. The generated clock phases are distributed to every DQS logic block that is available in the I/O sub-bank. The delay chain taps then feeds a multiplexer controlled by the ALTMEMPHY megafunction to select which clock phases are to be used for that ×4 or × 8 DQS group. Each group can use a different tap output from the read-leveling and write-leveling delay chains to compensate for the different CK/CK# delay going into each device on the module.

Figure 7–26 and Figure 7–27 show the Stratix IV write- and read-leveling circuitry.

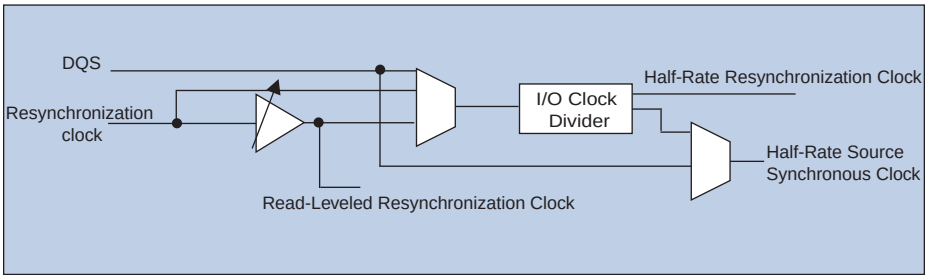
Figure 7–26. Stratix IV Write-Leveling Delay Chains and Multiplexers *(Note 1)*



Note to Figure 7–26:

(1) There is one leveling delay chain per I/O sub-bank (for example, I/O sub-banks 1A, 1B, and 1C). You can only have one memory interface in each I/O sub-bank when you use the leveling delay chain.

Figure 7–27. Stratix IV Read-Leveling Delay Chains and Multiplexers *(Note 1)*



Note to Figure 7–27:

(1) There is one leveling delay chain per I/O sub-bank (for example, I/O sub-banks 1A, 1B, and 1C). You can only have one memory interface in each I/O sub-bank when you use the leveling delay chain.

The –90° write clock of the ALTMEMPHY megafunction feeds the write-leveling circuitry to produce the clock to generate the DQS and DQ signals. During initialization, the ALTMEMPHY megafunction picks the correct write-leveled clock for the DQS and DQ clocks for each DQS/DQ group after sweeping all the available clocks in the write calibration process. The DQ clock output is –90° phase-shifted compared to the DQS clock output.

Similarly, the resynchronization clock feeds the read-leveling circuitry to produce the optimal resynchronization and postamble clock for each DQS/DQ group in the calibration process. The resynchronization and postamble clocks can use different clock outputs from the leveling circuitry. The output from the read-leveling circuitry can also generate the half-rate resynchronization clock that goes to the FPGA fabric.

The ALTMEMPHY megafunction dynamically calibrates the alignment for read and write-leveling during the initialization process.

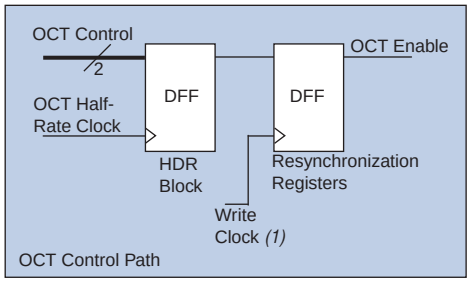
For more information about the ALTMEMPHY megafunction, refer to the *External Memory PHY Interface Megafunction User Guide (ALTMEMPHY)*.

Dynamic On-Chip Termination Control

Figure 7–28 shows the dynamic OCT control block. The block includes all the registers needed to dynamically turn on OCT RT during a read and turn OCT RT off during a write.

For more information about dynamic on-chip termination control, refer to the *I/O Features in Stratix IV Devices* chapter.

Figure 7–28. Stratix IV Dynamic OCT Control Block



Note to Figure 7–28:

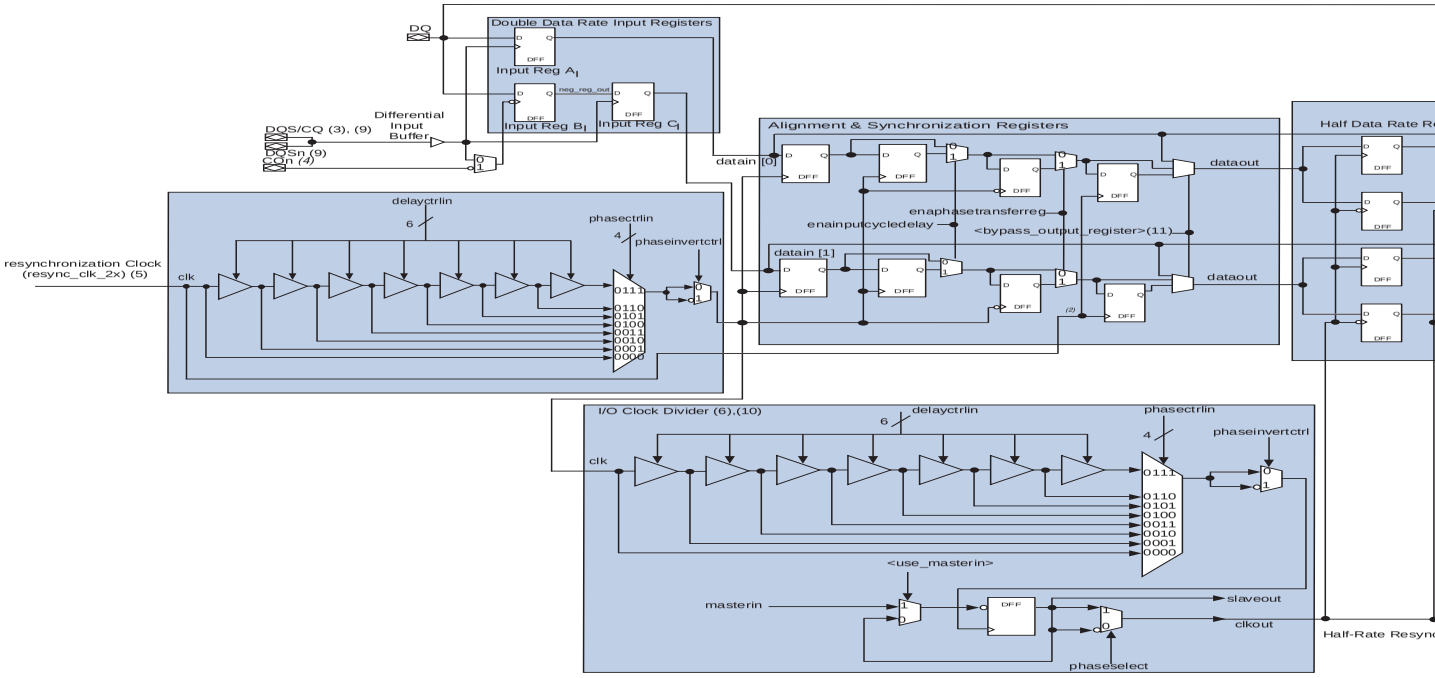
(1) The write clock comes from either the PLL or the write-leveling delay chain.

I/O Element Registers

The IOE registers are expanded to allow source-synchronous systems to have faster register-to-register transfers and resynchronization. Both top and bottom and left and right IOEs have the same capability. Left and right IOEs have extra features to support LVDS data transfer.

Figure 7–29 shows the registers available in the Stratix IV input path. The input path consists of the DDR input registers, resynchronization registers, and HDR block. You can bypass each block of the input path.

Figure 7–29. Stratix IV IOE Input Registers *(Note 1)*



Notes to Figure 7–29:

(1) You can bypass each register block in this path.

(2) This is the 0-phase resynchronization clock (from the read-leveling delay chain).

(3) The input clock can be from the DQS logic block (whether the postamble circuitry is bypassed or not) or from a global clock line.

(4) This input clock comes from the CQn logic block.

(5) This resynchronization clock can come either from the PLL or from the read-leveling delay chain.

(6) The I/O clock divider resides adjacent to the DQS logic block. In addition to the PLL and read-leveled resync clock, the I/O clock divider can also be fed by the DQS bus

(7) The half-rate data and clock signals feed into a dual-port RAM in the FPGA core.

(8) You can dynamically change the `dataoutbypass` signal after configuration to select either the `directin` input or the output from the half data rate register to fee

(9) The DQS and DQSn signals must be inverted for DDR, DDR2, and DDR3 interfaces. When using Altera's memory interface IPs, the DQS and DQSn signals are automati

(10) Each divider feeds up to six pins (from a ×4 DQS group) in the device. To feed wider DQS groups, you must chain multiple clock dividers together by feeding the `slav` `masterin` input of the neighboring pins' divider.

(11) The **bypass_output_register** option allows you to select either the output from the second mux or the output of the fourth alignment/ synchronization register to feed c

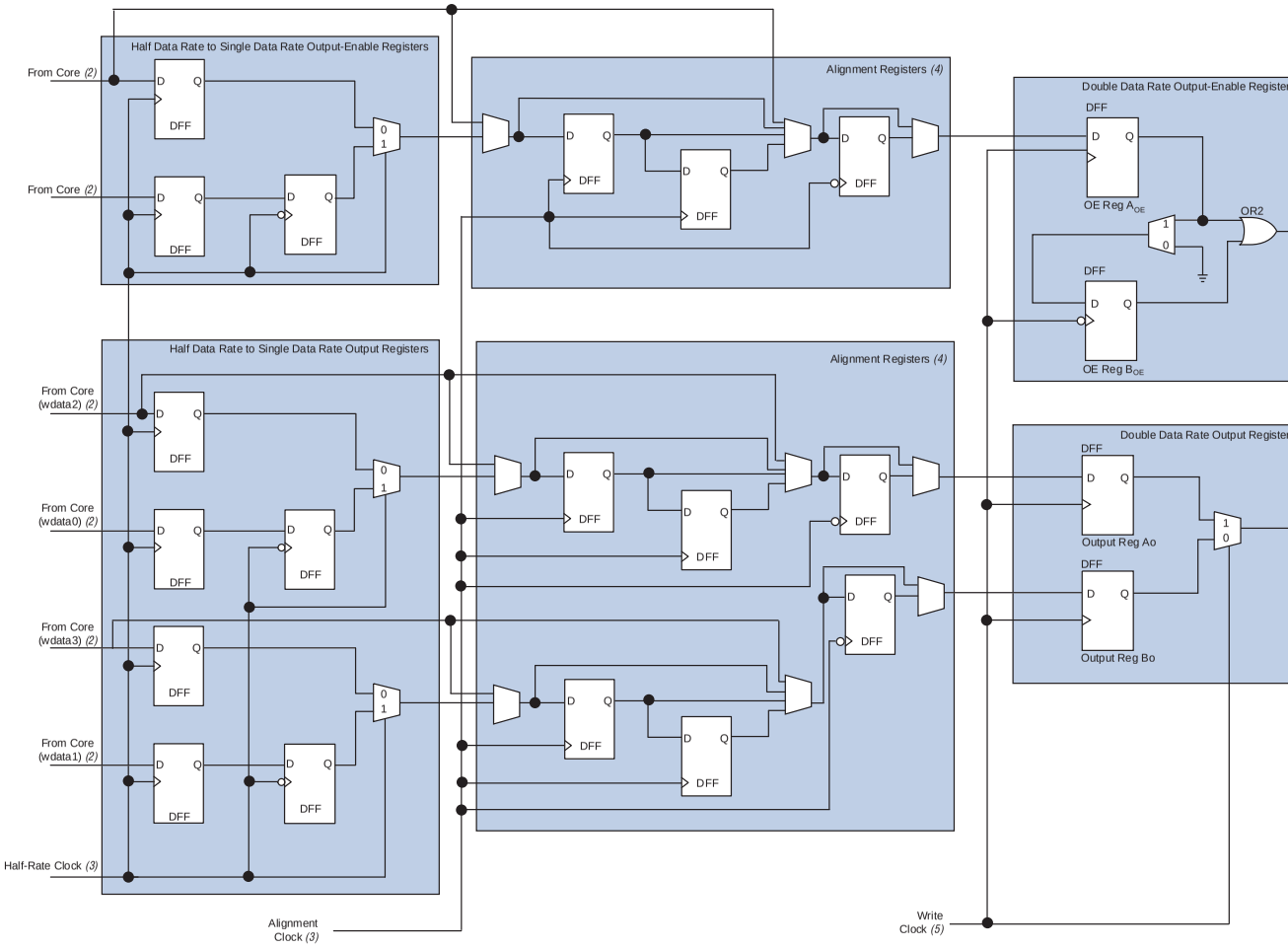
There are three registers in the DDR input registers block. Two registers capture data on the positive and negative edges of the clock, while the third register aligns the captured data. You can choose to use the same clock for the positive edge and negative edge registers, or two complementary clocks (DQS/CQ for the positive-edge register and DQSn/CQn for the negative-edge register). The third register that aligns the captured data uses the same clock as the positive edge registers.

The resynchronization registers consist of up to three levels of registers to resynchronize the data to the system clock domain. These registers are clocked by the resynchronization clock that is either generated by the PLL or the read-leveling delay chain. The outputs of the resynchronization registers can go straight to the core or to the HDR blocks, which are clocked by the divided-down resynchronization clock.

For more information about the read-leveling delay chain, refer to "Leveling Circuitry" on page 7–44.

Figure 7–30 shows the registers available in the Stratix IV output and output-enable paths. The path is divided into the HDR block, resynchronization registers, and output and output-enable registers. The device can bypass each block of the output and output-enable path.

Figure 7–30. Stratix IV IOE Output and Output-Enable Path Registers *(Note 1)*



Notes to Figure 7–30:

(1) You can bypass each register block of the output and output-enable paths.
(2) Data coming from the FPGA core are at half the frequency of the memory interface clock frequency in half-rate mode.
(3) The half-rate clock comes from the PLL, while the alignment clock comes from the write-leveling delay chains.
(4) These registers are only used in DDR3 SDRAM interfaces for write-leveling purposes.
(5) The write clock can come from either the PLL or from the write-leveling delay chain. The DQ write clock and DQS write clock have a 90° offset between them.

The output path is designed to route combinatorial or registered SDR outputs and full-rate or half-rate DDR outputs from the FPGA core. Half-rate data is converted to full-rate using the HDR block, clocked by the half-rate clock from the PLL. The resynchronization registers are also clocked by the same 0° system clock, except in the DDR3 SDRAM interface. In DDR3 SDRAM interfaces, the leveling registers are clocked by the write-leveling clock.

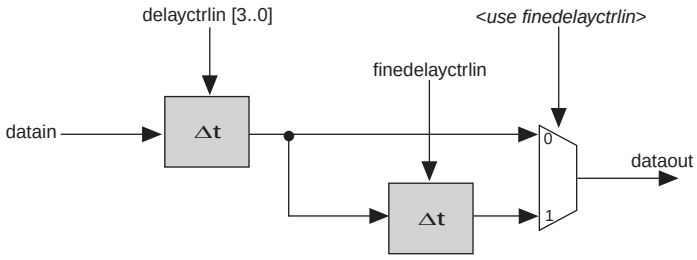
For more information about the write-leveling delay chain, refer to "Leveling Circuitry" on page 7–44.

The output-enable path has a structure similar to the output path. You can have a combinatorial or registered output in SDR applications and you can use half-rate or full-rate operation in DDR applications. Also, the ouput-enable path's resynchronization registers have a structure similar to the output path registers, ensuring that the output-enable path goes through the same delay and latency as the output path.

Delay Chain

Stratix IV devices have run-time adjustable delay chains in the I/O blocks and the DQS logic blocks. You can control the delay chain setting through the I/O or the DQS configuration block output. Figure 7–31 shows the delay chain ports.

Figure 7–31. Delay Chain

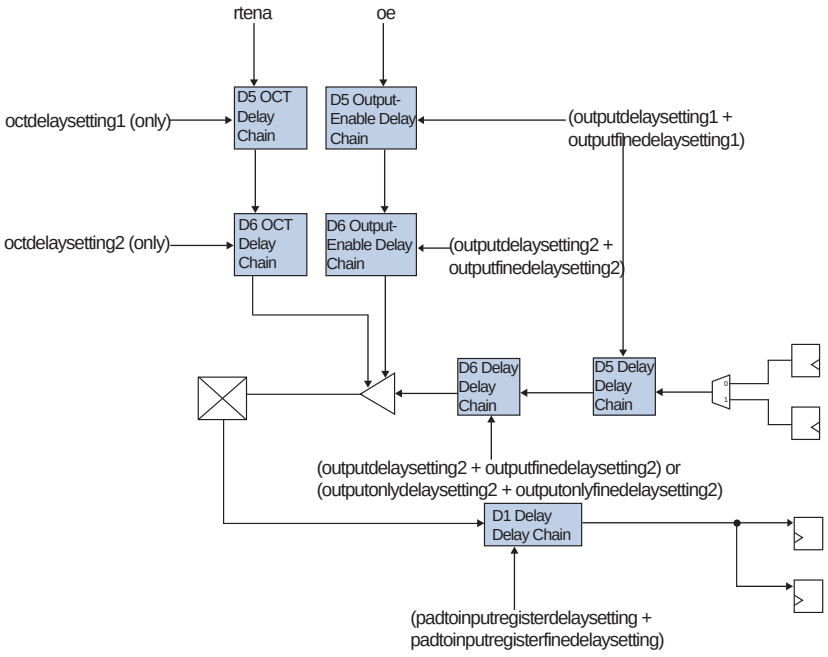


Every I/O block contains the following:

- Two delay chains in series between the output registers and output buffer
- One delay chain between the input buffer and input register
- Two delay chains between the output enable and output buffer
- Two delay chains between the OCT R_T enable control register and output buffer

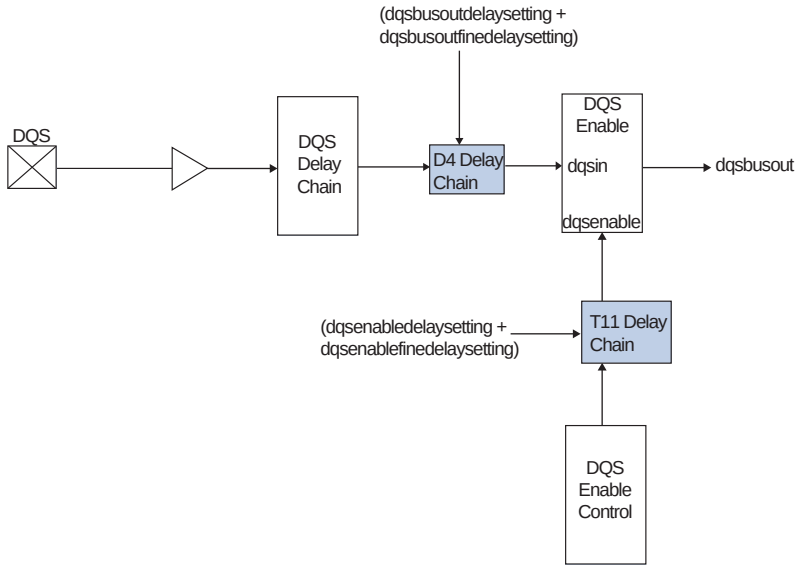
Figure 7–32 shows the delay chains in an I/O block.

Figure 7–32. Delay Chains in an I/O Block



Each DQS logic block contains a delay chain after the `dqsbusout` output and another delay chain before the `dqsenable` input. Figure 7–33 shows the delay chains in the DQS input path.

Figure 7–33. Delay Chains in the DQS Input Path



I/O Configuration Block and DQS Configuration Block

The I/O configuration block and the DQS configuration block are shift registers that you can use to dynamically change the settings of various device configuration bits. The shift registers power-up low. Every I/O pin contains one I/O configuration register, while every DQS pin contains one DQS configuration block in addition to the I/O configuration register. Figure 7–34 shows the I/O configuration block and the DQS configuration block circuitry.

Figure 7–34. I/O Configuration Block and DQS Configuration Block

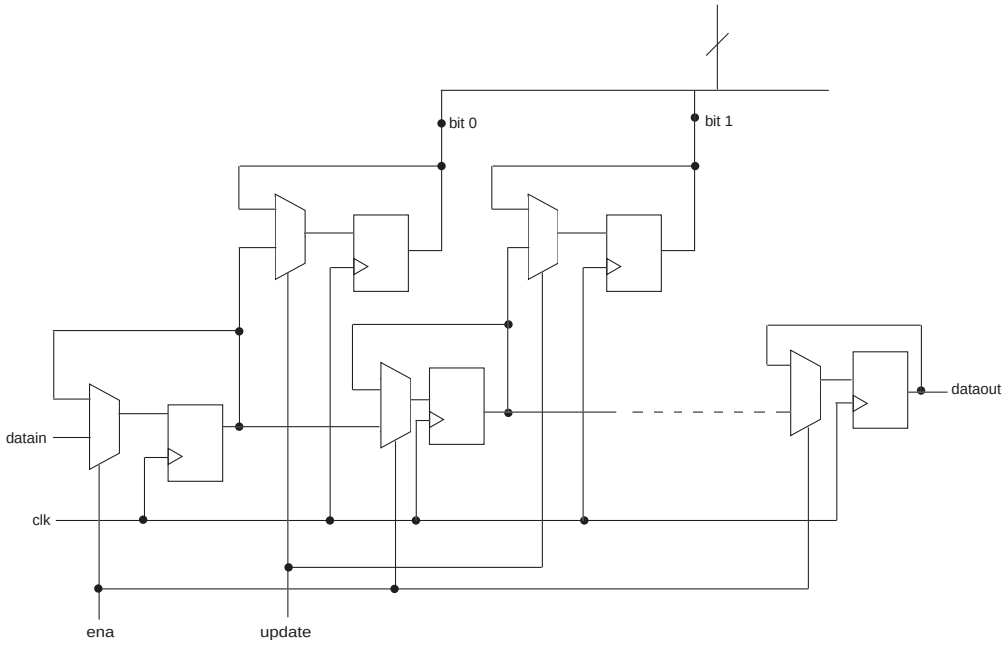


Table 7–19 lists the I/O configuration block bit sequence.

Table 7–19. I/O Configuration Block Bit Sequence

Bit	Bit Name
0..3	outputdelaysetting1[0..3]
4..6	outputdelaysetting2[0..2]
7..10	padtoinputregisterdelaysetting[0..3]
11	outputfinedelaysetting1
12	outputfinedelaysetting2
13	padtoinputregisterfinedelaysetting
14	outputonlyfinedelaysetting2
15..17	outputonlydelaysetting2[2..0]
18	dutycyclecorrectionmode
19..22	dutycyclecorrectionsetting[3..0]

Table 7–20 lists the DQS configuration block bit sequence.

Table 7–20. DQS Configuration Block Bit Sequence

Bit	Bit Name
0..3	dqsbusoutdelaysetting[0..3]
4..6	dqsinputphasesetting[0..2]
7..10	dqsenablectrlphasesetting[0..3]
11..14	dqsoutputphasesetting[0..3]
15..18	dqoutputphasesetting[0..3]
19..22	resyncinputphasesetting[0..3]
23	dividerphasesetting
24	enaoctcycledelaysetting
25	enainputcycledelaysetting
26	enaoutputcycledelaysetting
27..29	dqsenabledelaysetting[0..2]
30..33	octdelaysetting1[0..3]
34..36	octdelaysetting2[0..2]
37	enadataoutbypass
38	enadqsenablephasetransferreg
39	enaoctphasetransferreg
40	enaoutputphasetransferreg
41	enainputphasetransferreg
42	resyncinputphaseinvert
43	dqsenablectrlphaseinvert
44	dqoutputphaseinvert
45	dqsoutputphaseinvert
46	dqsbusoutfinedelaysetting
47	dqsenablefinedelaysetting

Document Revision History

Table 7–21 shows the revision history for this chapter.

Table 7–21. Document Revision History (Part 1 of 2)

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated the "Memory Interfaces Pin Support" and "Combining ×16/×18 DQS/DQ Groups for a ×36 QDRII+/QDRII SRAM Interface" sections. ■ Updated Table 7–1, Table 7–2, Table 7–7, and Table 7–12. ■ Updated Figure 7–3, Figure 7–4, Figure 7–5, Figure 7–6, Figure 7–7, Figure 7–8, Figure 7–9, Figure 7–10, Figure 7–11, Figure 7–13, Figure 7–14, Figure 7–15, and Figure 7–16. ■ Added Figure 7–12 and Figure 7–17. ■ Added Table 7–14, Table 7–17, Table 7–19, and Table 7–20. ■ Added "Delay Chain" and "I/O Configuration Block and DQS Configuration Block" sections. ■ Removed Figure 7-8 and Figure 7-12. ■ Removed Table 7-1, Table 7-2, and Table 7-24. ■ Minor text edits.	—
June 2009 v2.3	■ Updated "Overview" and "Leveling Circuitry". ■ Updated Figure 7–26 and Figure 7–27. ■ Updated Table 7–3. ■ Added introductory sentences to improve search ability. ■ Removed the Conclusion section.	—
April 2009 v2.2	■ Updated Table 7–5, Table 7–6, Table 7–15, and Table 7–17 ■ Removed Figure 7-12, Figure 7-13, and Figure 7-20	—
March 2009 v2.1	■ Updated Table 7–1, Table 7–5, Table 7–8, Table 7–12, Table 7–13, Table 7–14, Table 7–15, and Table 7–17. ■ Replaced Table 7–6. ■ Added Table 7–11 and Table 7–16. ■ Updated Figure 7–3, Figure 7–6, Figure 7–8, Figure 7–9, and Figure 7–11. ■ Added Figure 7–7, Figure 7–11, Figure 7–12, Figure 7–13, and Figure 7–20. ■ Updated "Combining ×16/×18 DQS/DQ Groups for ×36 QDRII+/QDRII SRAM Interface" on page 7–26. ■ Updated "Rules to Combine Groups" on page 7–27. ■ Removed "Referenced Documents" section.	—

Table 7–21. Document Revision History (Part 2 of 2)

Date and Document Version	Changes Made	Summary of Changes
November 2008 v2.0	■ Updated Table 7–1, Table 7–2, Table 7–3, Table 7–4, Table 7–5, and Table 7–6. ■ Added Table 7–7. ■ Updated Figure 7–1. ■ Updated "Combining ×16/×18 DQS/DQ groups for ×36 QDRII+/QDRII SRAM Interface" on page 7–26. ■ Updated "Rules to Combine Groups" on page 7–27. ■ Updated "DQS Phase-Shift Circuitry" on page 7–29. ■ Updated Figure 7–19. ■ Updated Table 7–9, Table 7–10, Table 7–11, Table 7–13, Table 7–13, Table 7–14, Table 7–15, Table 7–15, Table 7–16, and Table 7–18. ■ Updated Figure 7–30. ■ Updated Figure 7–31. ■ Made minor editorial changes.	—
May 2008 v1.0	Initial release.	—

8. High-Speed Differential I/O Interfaces and DPA in Stratix IV Devices

SIV51008-3.0

This chapter describes the significant advantages of the high-speed differential I/O interfaces and the dynamic phase aligner (DPA) over single-ended I/Os and their contribution to the overall system bandwidth achievable with Stratix® IV FPGAs.

The Stratix IV device family consists of the Stratix IV E (Enhanced) devices without high-speed clock data recovery (CDR) based transceivers, Stratix IV GT devices with up to 48 CDR-based transceivers running up to 11.3 Gbps, and Stratix IV GX devices with up to 48 CDR-based transceivers running up to 8.5 Gbps.

The following sections describe the Stratix IV high-speed differential I/O interfaces and DPA in detail:

Overview

All Stratix IV E, GX, and GT devices have built-in serializer/deserializer (SERDES) circuitry that supports high-speed LVDS interfaces at data rates of up to 1.6 Gbps. SERDES circuitry is configurable to support source-synchronous communication protocols such as Utopia, Rapid I/O, XSBI, small form factor interface (SFI), serial peripheral interface (SPI), and asynchronous protocols such as SGMII and Gigabit Ethernet.

All references to Stratix IV devices in this chapter apply to Stratix IV E, GT, and GX devices.

The Stratix IV device family has the following dedicated circuitry for high-speed differential I/O support:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer

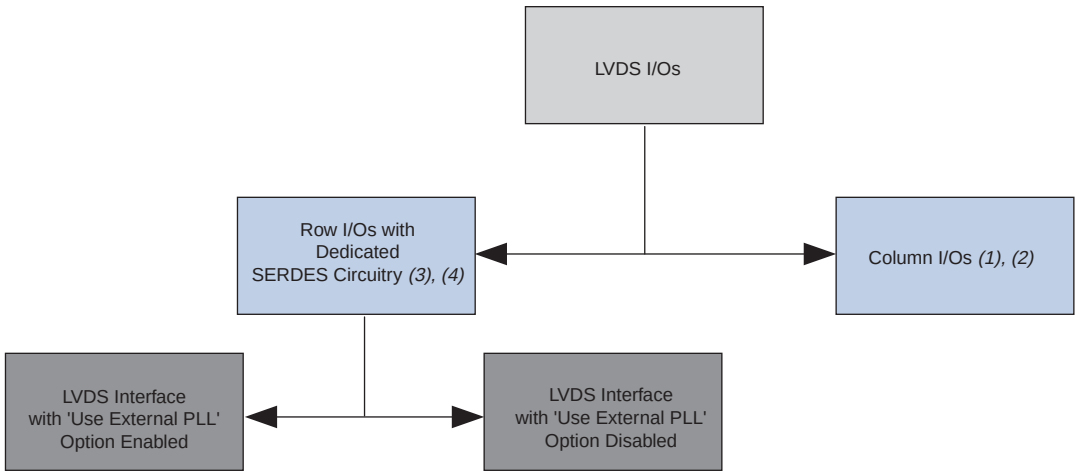
- Data realignment
- DPA
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs) (located on left and right sides of the device)

For high-speed differential interfaces, the Stratix IV device family supports the following differential I/O standards:

- LVDS
- Mini-LVDS
- Reduced swing differential signaling (RSDS)

In the Stratix IV device family, I/Os are divided into row and column I/Os. Figure 8–1 shows I/O bank support for the Stratix IV device family. The row I/Os provide dedicated SERDES circuitry.

Figure 8–1. I/O Bank Support in the Stratix IV Device Family *(Note 1), (2), (3), (4)*



Notes to Figure 8–1:

(1) Column input buffers are true LVDS buffers, but do not support 100-Ω differential on-chip termination.

(2) Column output buffers are single ended and need external termination schemes to support LVDS, mini-LVDS, and RSDS standards. For more information, refer to the *I/O Features in Stratix IV Devices* chapter.

(3) Row input buffers are true LVDS buffers and support 100-Ω differential on-chip termination.

(4) Row output buffers are true LVDS buffers.

The ALTLVDS transmitter and receiver requires various clock and load enable signals from a left or right PLL. The Quartus® II software provides the following two choices when configuring the LVDS SERDES circuitry when using the PLL:

- LVDS interface with the **Use External PLL** option enabled—You control the PLL settings, such as dynamically reconfiguring the PLL to support different data rates, dynamic phase shift, and so on. You must enable the **Use External PLL** option in the ALTLVDS megafunction, using the ALTLVDS MegaWizard™ Plug-in Manager software. You also must instantiate an ALTPLL megafunction to generate the various clocks and load enable signals. For more information, refer to "LVDS Interface with the Use External PLL Option Enabled" on page 8–25.

- LVDS interface with the **Use External PLL** option disabled—The Quartus II software configures the PLL settings automatically. The software is also responsible for generating the various clock and load enable signals based on the input reference clock and data rate selected.

Both choices target the same physical PLL; the only difference is the additional flexibility provided when an LVDS interface has the **Use External PLL** option enabled.

Locations of the I/O Banks

Stratix IV I/Os are divided into 16 to 24 I/O banks. The dedicated circuitry that supports high-speed differential I/Os is located in banks in the right and left side of the device. Figure 8–2 shows a high-level chip overview of the Stratix IV E device.

Figure 8–2. High-Speed Differential I/Os with DPA Locations in Stratix IV E Devices

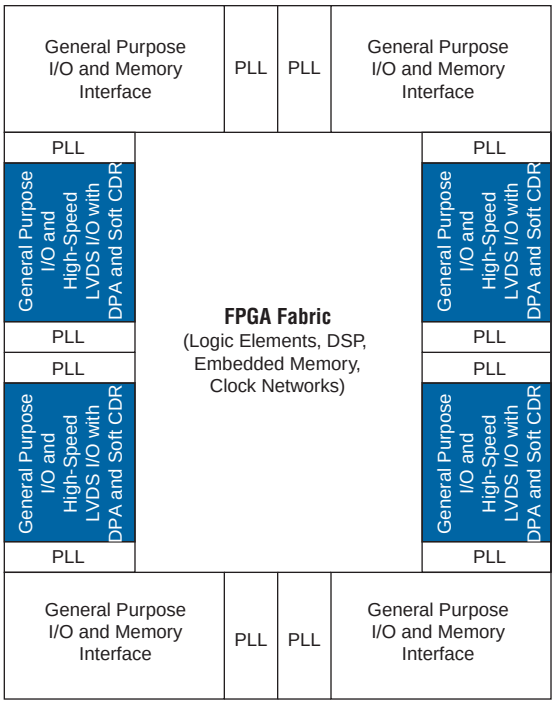
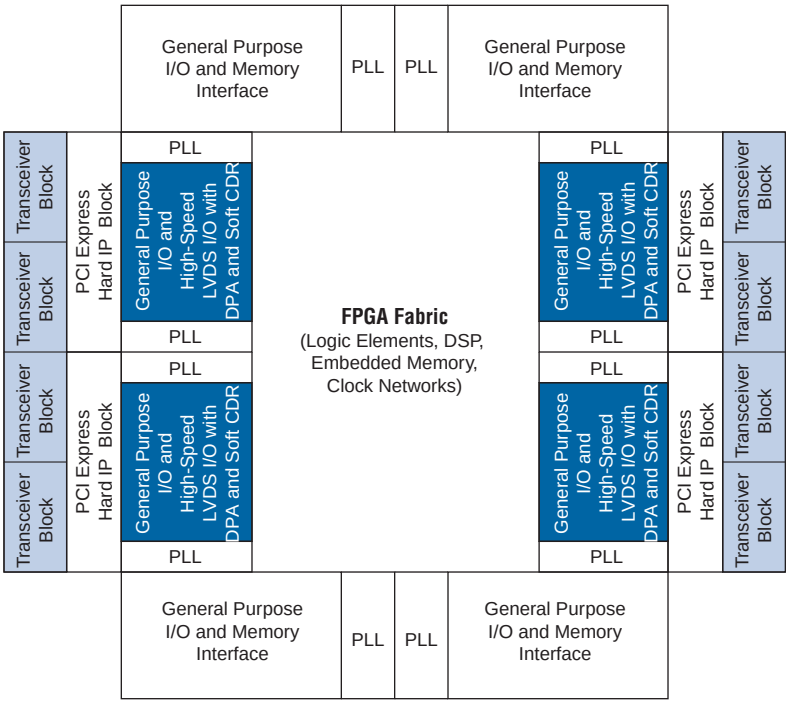


Figure 8–3 shows a high-level chip overview of the Stratix IV GT and GX devices.

Figure 8–3. High-Speed Differential I/Os with DPA Locations in Stratix IV GT and GX Devices



LVDS Channels

The Stratix IV device family supports LVDS on both row and column I/O banks. Row I/Os support true LVDS input with 100-Ω differential input termination (OCT R_D), and true LVDS output buffers. Column I/Os supports true LVDS input buffers without OCT R_D. Alternately, you can configure the row and column LVDS pins as emulated LVDS output buffers that use two single-ended output buffers with an external resistor network to support LVDS, mini-LVDS, and RSDS standards. Dedicated SERDES and DPA circuitries are implemented on the row I/O banks to further enhance LVDS interface performance in the device. For column I/O banks, SERDES is implemented in the core logic because there is no dedicated SERDES circuitry on column I/O banks.

Emulated LVDS output buffers support tri-state capability starting with the Quartus II software version 9.1

Table 8–1 and Table 8–2 list the maximum number of row and column LVDS I/Os supported in Stratix IV E devices. You can design the LVDS I/Os as true LVDS buffers or emulated LVDS buffers, as long as the combination of the two do not exceed the maximum count.

For example, there are a total of 112 LVDS pairs on row I/Os in the 780-pin EP4SE230 device (refer to Table 8–1). You can design up to a maximum of 56 true LVDS input buffers and 56 true LVDS output buffers, or up to a maximum of 112 emulated LVDS output buffers. For the 780-pin EP4SE230 device (refer to Table 8–2), there are a total of 128 LVDS pairs on column I/Os. You can design up to a maximum of 64 true LVDS input buffers and 64 emulated LVDS output buffers, or up to a maximum of 128 emulated LVDS output buffers.

Table 8–1. LVDS Channels Supported in Stratix IV E Device Row I/O Banks *(Note 1)*, *(2)*, *(3)*

Device	780-Pin FineLine BGA	1152-Pin FineLine BGA	1517-Pin FineLine BGA	1760- Pin FineLine BGA
EP4SE230	56 Rx or eTx + 56 Tx or eTx	—	—	—
EP4SE360	56 Rx or eTx + 56 Tx or eTx *(4)*	88 Rx or eTx + 88 Tx or eTx	—	—
EP4SE530	—	88 Rx or eTx + 88 Tx or eTx *(5)*	112 Rx or eTx + 112 Tx or eTx *(6)*	112 Rx or eTx + 112 Tx or eTx
EP4SE820 *(7)*	—	88 Rx or eTx + 88 Tx or eTx	112 Rx or eTx + 112 Tx or eTx	132 Rx or eTx + 132 Tx or eTx

Notes to Table 8–1:

(1) Rx = true LVDS input buffers with OCT R_D, Tx = true LVDS output buffers, eTx = emulated LVDS output buffers (either `LVDS_E_1R` or `LVDS_E_3R`).
(2) The LVDS Rx and Tx channels are equally divided between the left and right sides of the device.
(3) The LVDS channel count does not include dedicated clock input pins.
(4) EP4SE360 devices are offered in the H780 package instead of the F780 package.
(5) EP4SE530 devices are offered in the H1152 package instead of the F1152 package.
(6) EP4SE530 devices are offered in the H1517 package instead of the F1517 package.
(7) The number of LVDS Rx and Tx channels are preliminary.

Table 8–2. LVDS Channels Supported in Stratix IV E Device Column I/O Banks *(Note 1)*, *(2)*, *(3)*

Device	780-Pin FineLine BGA	1152-Pin FineLine BGA	1517-Pin FineLine BGA	1760-Pin FineLine BGA
EP4SE230	64 Rx or eTx + 64 eTx	—	—	—
EP4SE360	64 Rx or eTx + 64 eTx *(4)*	96 Rx or eTx + 96 eTx	—	—
EP4SE530	—	96 Rx or eTx + 96 eTx *(5)*	128 Rx or eTx + 128 eTx *(6)*	128 Rx or eTx + 128 eTx
EP4SE820 *(7)*	—	96 Rx or eTx + 96 eTx	128 Rx or eTx + 128 eTx	144 Rx or eTx + 144 eTx

Notes to Table 8–2:

(1) Rx = true LVDS input buffers without OCT R_D, eTx = emulated LVDS output buffers (either `LVDS_E_1R` or `LVDS_E_3R`).
(2) The LVDS Rx and Tx channels are equally divided between the top and bottom sides of the device.
(3) The LVDS channel count does not include dedicated clock input pins.
(4) EP4SE360 devices are offered in the H780 package instead of the F780 package.
(5) EP4SE530 devices are offered in the H1152 package instead of the F1152 package.
(6) EP4SE530 devices are offered in the H1517 package instead of the F1517 package.
(7) The number of LVDS Rx and Tx channels are preliminary.

Table 8–3 and Table 8–4 list the maximum number of row and column LVDS I/Os supported in Stratix IV GT devices.

Table 8–3. LVDS Channels Supported in Stratix IV GT Device Row I/O Banks *(Note 1)*, *(2)*

Device	1517-pin FineLine BGA	1932-pin FineLine BGA
EP4S40G2	46 Rx or eTx + 73 Tx or eTx	—
EP4S40G5	46 Rx or eTx + 73 Tx or eTx	—
EP4S100G2	46 Rx or eTx + 73 Tx or eTx	—
EP4S100G3	—	47 Rx or eTx + 56 Tx or eTx
EP4S100G4	—	47 Rx or eTx + 56 Tx or eTx
EP4S100G5	46 Rx or eTx + 73 Tx or eTx	47 Rx or eTx + 56 Tx or eTx

Notes to Table 8–3:

(1) Rx = true LVDS input buffers with OCT R_D, eTx = emulated LVDS output buffers (either LVDS_E_1R or LVDS_E_3R).

(2) The LVDS Rx and Tx channel count does not include dedicated clock input pins.

Table 8–4. LVDS Channels Supported in Stratix IV GT Device Column I/O Banks *(Note 1)*, *(2)*

Device	1517-pin FineLine BGA	1932-pin FineLine BGA
EP4S40G2	96 Rx or eTx + 96 eTx	—
EP4S40G5	96 Rx or eTx + 96 eTx	—
EP4S100G2	96 Rx or eTx + 96 eTx	—
EP4S100G3	—	128 Rx or eTx + 128 eTx
EP4S100G4	—	128 Rx or eTx + 128 eTx
EP4S100G5	96 Rx or eTx + 96 eTx	128 Rx or eTx + 128 eTx

Notes to Table 8–4:

(1) Rx = true LVDS input buffers without OCT R_D, eTx = emulated LVDS output buffers (either LVDS_E_1R or LVDS_E_3R).

(2) The LVDS Rx and Tx channel count does not include dedicated clock input pins.

Table 8–5 and Table 8–6 list the maximum number of row and column LVDS I/Os supported in Stratix IV GX devices.

Table 8–5. LVDS Channels Supported in Stratix IV GX Device Row I/O Banks *(Note 1)*, *(2)*, *(3)* (Part 1 of 2)

Device	780-Pin FineLine BGA	1152-Pin FineLine BGA	1152-Pin FineLine BGA *(4)*	1517-Pin FineLine BGA	1760-Pin FineLine BGA	1932-Pin FineLine BGA
EP4SGX70	28 Rx or eTx + 28 Tx or eTx	—	56 Rx or eTx + 56 Tx or eTx	—	—	—
EP4SGX110	28 Rx or eTx + 28 Tx or eTx	28 Rx or eTx + 28 Tx or eTx	56 Rx or eTx + 56 Tx or eTx	—	—	—
EP4SGX180	28 Rx or eTx + 28 Tx or eTx	44 Rx or eTx + 44 Tx or eTx	44 Rx or eTx + 44 Tx or eTx	88 Rx or eTx + 88 Tx or eTx	—	—
EP4SGX230	28 Rx or eTx + 28 Tx or eTx	44 Rx or eTx + 44 Tx or eTx	44 Rx or eTx + 44 Tx or eTx	88 Rx or eTx + 88 Tx or eTx	—	—
EP4SGX290	— *(5)*	44 Rx or eTx + 44 Tx or eTx	44 Rx or eTx + 44 Tx or eTx	88 Rx or eTx + 88 Tx or eTx	88 Rx or eTx + 88 Tx or eTx	98 Rx or eTx + 98 Tx or eTx

Table 8–5. LVDS Channels Supported in Stratix IV GX Device Row I/O Banks *(Note 1)*, *(2)*, *(3)* (Part 2 of 2)

Device	780-Pin FineLine BGA	1152-Pin FineLine BGA	1152-Pin FineLine BGA *(4)*	1517-Pin FineLine BGA	1760-Pin FineLine BGA	1932-Pin FineLine BGA
EP4SGX360	— *(5)*	44 Rx or eTx + 44 Tx or eTx	44 Rx or eTx + 44 Tx or eTx	88 Rx or eTx + 88 Tx or eTx	88 Rx or eTx + 88 Tx or eTx	98 Rx or eTx + 98 Tx or eTx
EP4SGX530	—	—	44 Rx or eTx + 44 Tx or eTx *(6)*	88 Rx or eTx + 88 Tx or eTx *(7)*	88 Rx or eTx + 88 Tx or eTx	98 Rx or eTx + 98 Tx or eTx

Notes to Table 8–5:

(1) Rx = true LVDS input buffers with OCT R_D, Tx = true LVDS output buffers, eTx = emulated LVDS output buffers (either `LVDS_E_1R` or `LVDS_E_3R`).
(2) The LVDS Rx and Tx channels are equally divided between the left and right sides of the device, except for the devices in the 780-pin Fineline BGA. These devices have the LVDS Rx and Tx located on the left side of the device.
(3) The LVDS channel count does not include dedicated clock input pins.
(4) This package supports PMA-only transceiver channels.
(5) EP4SGX290 and EP4SGX360 devices are offered in the H780 package instead of the F780 package.
(6) EP4SGX530 devices are offered in the H1152 package instead of the F1152 package.
(7) EP4SGX530 devices are offered in the H1517 package instead of the F1517 package.

Table 8–6. LVDS Channels Supported in Stratix IV GX Device Column I/O Banks *(Note 1)*, *(2)*, *(3)*

Device	780-Pin FineLine BGA	1152-Pin FineLine BGA	1152-Pin FineLine BGA *(4)*	1517-Pin FineLine BGA	1760-Pin FineLine BGA	1932-Pin FineLine BGA
EP4SGX70	64 Rx or eTx + 64 eTx	—	64 Rx or eTx + 64 eTx	—	—	—
EP4SGX110	64 Rx or eTx + 64 eTx	64 Rx or eTx + 64 eTx	64 Rx or eTx + 64 eTx	—	—	—
EP4SGX180	64 Rx or eTx + 64 eTx	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	—	—
EP4SGX230	64 Rx or eTx + 64 eTx	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	—	—
EP4SGX290	72 Rx or eTx + 72 eTx *(5)*	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	128 Rx or eTx + 128 eTx	128 Rx or eTx + 128 eTx *(8)*
EP4SGX360	72 Rx or eTx + 72 eTx *(5)*	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	96 Rx or eTx + 96 eTx	128 Rx or eTx + 128 eTx	128 Rx or eTx + 128 eTx *(8)*
EP4SGX530	—	—	96 Rx or eTx + 96 eTx *(6)*	96 Rx or eTx + 96 eTx *(7)*	128 Rx or eTx + 128 eTx	128 Rx or eTx + 128 eTx

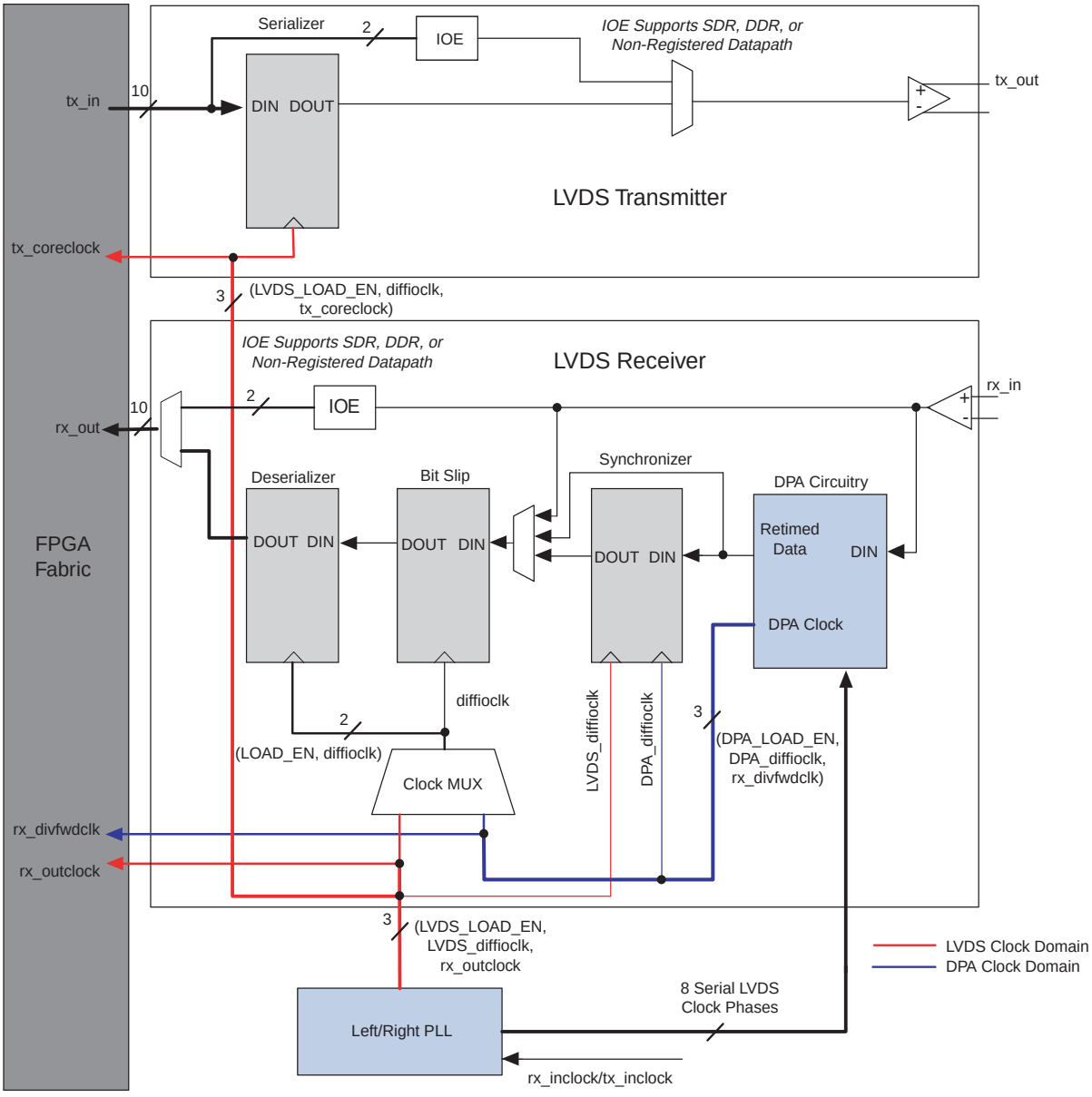
Notes to Table 8–6:

(1) Rx = true LVDS input buffers without OCT R_D, eTx = emulated LVDS output buffers (either `LVDS_E_1R` or `LVDS_E_3R`).
(2) The LVDS Rx and Tx channels are equally divided between the left and right sides of the device.
(3) The LVDS channel count does not include dedicated clock input pins.
(4) This package supports PMA-only transceiver channels.
(5) EP4SGX290 and EP4SGX360 devices are offered in the H780 package instead of the F780 package.
(6) EP4SGX530 devices are offered in the H1152 package instead of the F1152 package.
(7) EP4SGX530 devices are offered in the H1517 package instead of the F1517 package.
(8) The Quartus II software version 9.0 does not support EP4SGX290 and EP4SGX360 devices in the 1932-Pin FineLine BGA package. These devices will be supported in a future release of the Quartus II software.

LVDS SERDES

Figure 8–4 shows a transmitter and receiver block diagram for the LVDS SERDES circuitry in the left and right banks. This diagram shows the interface signals of the transmitter and receiver data path. For more information, refer to "Differential Transmitter" on page 8–11 and "Differential Receiver" on page 8–16.

Figure 8–4. LVDS SERDES *(Note 1), (2), (3)*



Notes to Figure 8–4:

(1) This diagram shows a shared PLL between the transmitter and receiver. If the transmitter and receiver are not sharing the same PLL, the two left and right PLLs are required.

(2) In SDR and DDR modes, the data width is 1 and 2 bits, respectively.

(3) The `tx_in` and `rx_out` ports have a maximum data width of 10 bits.

ALTLVDS Port List

Table 8–7 lists the interface signals for a LVDS transmitter and receiver.

Table 8–7. Port List of the LVDS Interface (ALTLVDS) *(Note 1)*, *(2)*, *(3)* (Part 1 of 3)

Port Name	Input / Output	Description
PLL Signals		
pll_areset	Input	Asynchronous reset to the LVDS transmitter and receiver PLL. The minimum pulse width requirement for this signal is 10 ns.
LVDS Transmitter Interface Signals		
tx_in[ ]	Input	The data bus width per channel is the same as the serialization factor (SF). Input data must be synchronous to the tx_coreclock signal.
tx_inclock	Input	Reference clock input for the transmitter PLL. The allowed frequency range for the reference clock is between 5 MHz and 625 MHz and must satisfy the following condition: 5 Mbps < ([Input Clock Frequency] × [PLL Multiplication Factor]) < 1600 Mbps. The ALTLVDS MegaWizard Plug-In Manager software automatically selects the appropriate PLL multiplication factor based on the data rate and reference clock frequency selection.
tx_enable *(3)*	Input	This port is instantiated only when you select the **Use External PLL** option in the MegaWizard Plug-In Manager software. This input port must be driven by the PLL instantiated though the ALTPLL MegaWizard Plug-In Manager software.
tx_out	Output	LVDS transmitter serial data output port. tx_out is clocked by a serial clock generated by the left and right PLL.
tx_outclock	Output	The frequency of this clock is programmable to be the same as the data rate (up to 717 MHz), half the data rate, or one-fourth the data rate. The phase offset of this clock, with respect to the serial data, is programmable in increments of 45°.
tx_coreclock *(3)*	Output	FPGA fabric-transmitter interface clock. The parallel transmitter data generated in the FPGA fabric must be clocked with this clock. This port is not available when you select the **Use External PLL** option in the MegaWizard Plug-In Manager software. The FPGA fabric-transmitter interface clock must be driven by the PLL instantiated through the ALTPLL MegaWizard Plug-In Manager software.
tx_locked	Output	When high, this signal indicates that the transmitter PLL is locked to the input reference clock.

Table 8–7. Port List of the LVDS Interface (ALTLVDS) *(Note 1)*, *(2)*, *(3)* (Part 2 of 3)

Port Name	Input / Output	Description
LVDS Receiver Interface Signals		
rx_in	Input	LVDS receiver serial data input port.
rx_inclock	Input	Reference clock input for the receiver PLL. The allowed frequency range for the reference clock is between 5 MHz and 625 MHz and must satisfy the following condition: 5 Mbps < ([Input Clock Frequency] × [PLL Multiplication Factor]) < 1600 Mbps. The ALTLVDS MegaWizard Plug-In Manager software automatically selects the appropriate PLL multiplication factor based on the data rate and reference clock frequency selection.
rx_channel_data_align	Input	Edge-sensitive bit-slip control signal. Each rising edge on this signal causes the data re-alignment circuitry to shift the word boundary by one bit. The minimum pulse width requirement is one parallel clock cycle. There is no maximum pulse width requirement.
rx_dpll_hold	Input	When low, the DPA tracks any dynamic phase variations between the clock and data. When high, the DPA holds the last locked phase and does not track any dynamic phase variations between the clock and data. This port is not available in non-DPA mode.
rx_enable *(3)*	Input	This port is instantiated only when you select the **Use External PLL** option in the MegaWizard Plug-In Manager software. This input port must be driven by the PLL instantiated though the ALTPLL MegaWizard Plug-In Manager software.
rx_out[ ]	Output	Receiver parallel data output. The data bus width per channel is the same as the deserialization factor (DF). The output data is synchronous to the rx_outclock signal in non-DPA and DPA modes. It is synchronous to the rx_divfwdclk signal in soft-CDR mode.
rx_outclock	Output	Parallel output clock from the receiver PLL. The parallel data output from the receiver is synchronous to this clock in non-DPA and DPA modes. This port is not available when you select the **Use External PLL** option in the MegaWizard Plug-In Manager software. The FPGA fabric-receiver interface clock must be driven by the PLL instantiated through the ALTPLL MegaWizard Plug-In Manager software.
rx_locked	Output	When high, this signal indicates that the receiver PLL is locked to rx_inclock.
rx_dpa_locked	Output	This signal only indicates an initial DPA lock condition to the optimum phase after power up or reset. This signal is not de-asserted if the DPA selects a new phase out of the eight clock phases to sample the received data. You must not use the rx_dpa_locked signal to determine a DPA loss-of-lock condition.
rx_cda_max	Output	Data re-alignment (bit slip) roll-over signal. When high for one parallel clock cycle, this signal indicates that the user-programmed number of bits for the word boundary to roll-over have been slipped.
rx_divfwdclk	Output	Parallel DPA clock to the FPGA fabric logic array. The parallel receiver output data to the FPGA fabric logic array is synchronous to this clock in soft-CDR mode. This signal is not available in non-DPA and DPA modes.

Table 8–7. Port List of the LVDS Interface (ALTLVDS) *(Note 1)*, *(2)*, *(3)* (Part 3 of 3)

Port Name	Input / Output	Description
dpa_pll_recal	Input	Enable PLL calibration dynamically without resetting the DPA circuitry or the PLL.
dpa_pll_cal_busy	Output	Busy signal that is asserted high when the PLL calibration occurs.
Reset Signals		
rx_reset	Input	Asynchronous reset to the DPA circuitry and FIFO. The minimum pulse width requirement for this reset is one parallel clock cycle. This signal resets DPA and FIFO blocks.
rx_fifo_reset	Input	Asynchronous reset to the FIFO between the DPA and the data realignment circuits. The synchronizer block must be reset after a DPA loses lock condition and the data checker shows corrupted received data. The minimum pulse width requirement for this reset is one parallel clock cycle. This signal resets the FIFO block.
rx_cda_reset	Input	Asynchronous reset to the data realignment circuitry. The minimum pulse width requirement for this reset is one parallel clock cycle. This signal resets the data realignment block.

Notes to Table 8–7:

(1) Unless stated, signals are valid in all three modes (non-DPA, DPA, and soft-CDR) for a single channel.
(2) All reset and control signals are active high.
(3) For more information, refer to "LVDS Interface with the Use External PLL Option Enabled" on page 8–25.

For more information about the LVDS transmitter and receiver settings using ALTLVDS, refer to the *SERDES Transmitter/Receiver (ALTLVDS) Megafunction User Guide*.

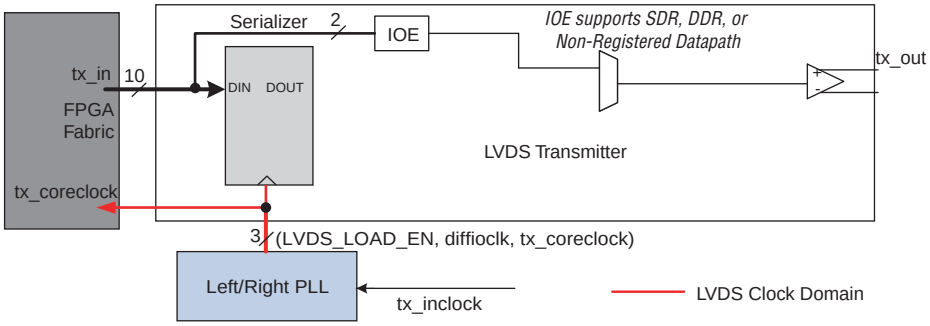
Differential Transmitter

The Stratix IV transmitter has a dedicated circuitry to provide support for LVDS signaling. The dedicated circuitry consists of a differential buffer, a serializer, and left and right PLLs, that can be shared between the transmitter and receiver. The differential buffer can drive out LVDS, mini-LVDS, and RSDS signaling levels. The serializer takes up to 10 bits wide parallel data from the FPGA fabric, clocks it into the load registers, and serializes it using shift registers clocked by the left and right PLL before sending the data to the differential buffer. The MSB of the parallel data is transmitted first.

When using emulated LVDS I/O standards at the differential transmitter, the SERDES circuitry must be implemented in logic cells but not the hard SERDES.

The load enable (LVDS_LOAD_EN) signal and the diffioclk signal (the clock running at serial data rate) generated from the PLL_Lx (left PLL) or PLL_Rx (right PLL) clocks the load and shift registers. You can statically set the serialization factor to ×4, ×6, ×7, ×8, or ×10 using the Quartus II software. The load enable signal is derived from the serialization factor setting. Figure 8–5 shows a block diagram of the Stratix IV transmitter.

Figure 8–5. Stratix IV Transmitter *(Note 1), (2)*



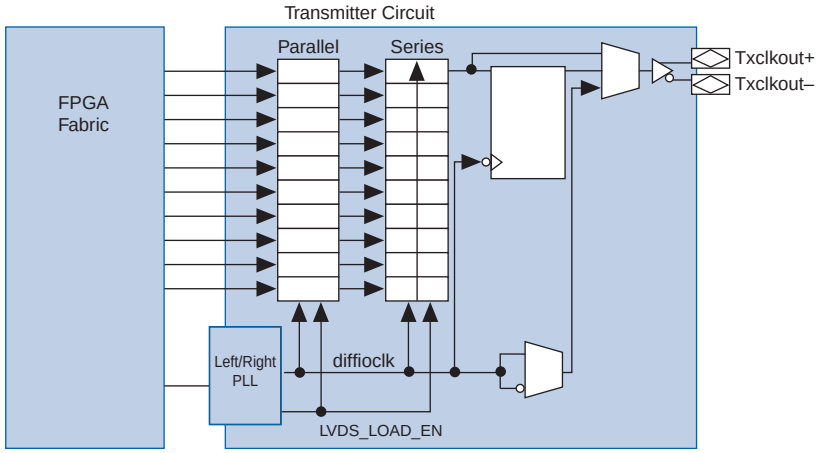
Notes to Figure 8–5:

(1) In SDR and DDR modes, the data width is 1 and 2 bits, respectively.

(2) The `tx_in` port has a maximum data width of 10 bits.

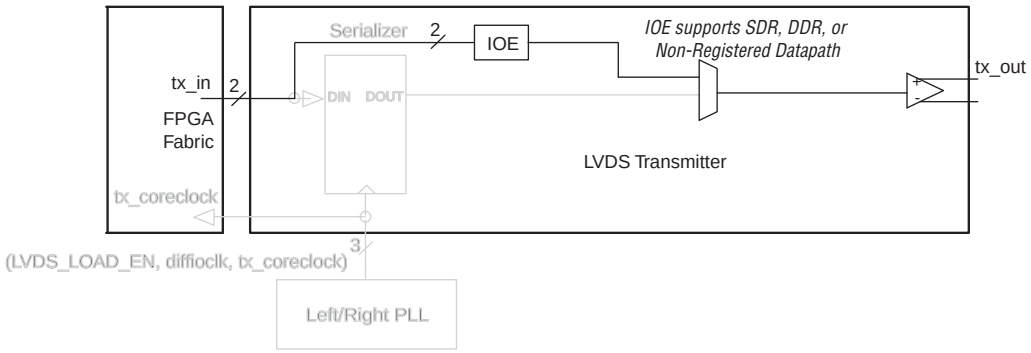
You can configure any Stratix IV transmitter data channel to generate a source-synchronous transmitter clock output. This flexibility allows the placement of the output clock near the data outputs to simplify board layout and reduce clock-to-data skew. Different applications often require specific clock-to-data alignments or specific data-rate-to-clock-rate factors. The transmitter can output a clock signal at the same rate as the data with a maximum frequency of 800 MHz. The output clock can also be divided by a factor of 1, 2, 4, 6, 8, or 10, depending on the serialization factor. You can set the phase of the clock in relation to the data at 0° or 180° (edge or center aligned). The left and right PLLs (`PLL_Lx` and `PLL_Rx`) provide additional support for other phase shifts in 45° increments. These settings are made statically in the Quartus II MegaWizard Plug-In Manager software. Figure 8–6 shows the Stratix IV transmitter in clock output mode. In clock output mode, you can use an LVDS channel as a clock output channel.

Figure 8–6. Stratix IV Transmitter in Clock Output Mode



You can bypass the Stratix IV serializer to support DDR (×2) and SDR (×1) operations to achieve a serialization factor of 2 and 1, respectively. The I/O element (IOE) contains two data output registers that can each operate in either DDR or SDR mode. Figure 8–7 shows the serializer bypass path.

Figure 8–7. Stratix IV Serializer Bypass *(Note 1)*, *(2)*, *(3)*



Notes to Figure 8–7:

(1) All disabled blocks and signals are grayed out.
(2) In DDR mode, `tx_inclock` clocks the IOE register. In SDR mode, data is directly passed through the IOE.
(3) In SDR and DDR modes, the data width to the IOE is 1 and 2 bits, respectively.

Programmable V_{OD} and Programmable Pre-Emphasis

Stratix IV LVDS transmitters support programmable pre-emphasis and programmable V_{OD}. Pre-emphasis increases the amplitude of the high-frequency component of the output signal, and thus helps to compensate for the frequency-dependent attenuation along the transmission line. Figure 8–8 shows the differential LVDS output.

Figure 8–8. Differential V_{OD}

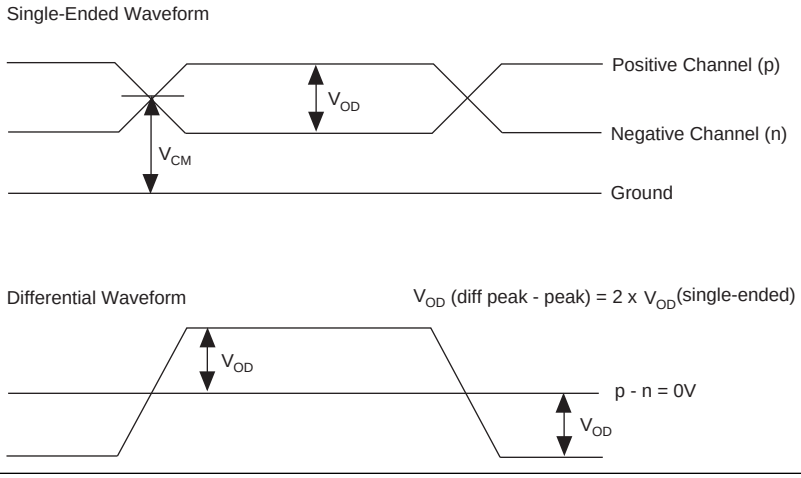
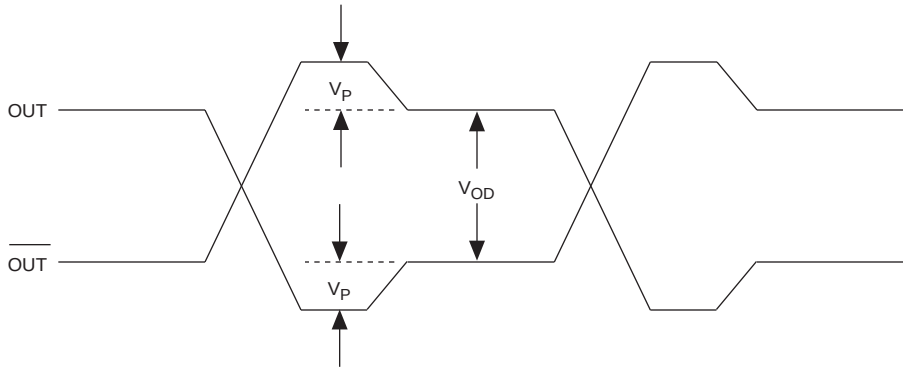


Figure 8–9 shows the LVDS output with pre-emphasis.

Figure 8–9. Programmable Pre-Emphasis *(Note 1)*



Note to Figure 8–9:

(1) V_P— voltage boost from pre-emphasis. V_{OD}— Differential output voltage (peak-peak).

Pre-emphasis is an important feature for high-speed transmission. Without pre-emphasis, the output current is limited by the V_{OD} setting and the output impedance of the driver. At high frequency, the slew rate may not be fast enough to reach full V_{OD} before the next edge, producing pattern-dependent jitter.

With pre-emphasis, the output current is boosted momentarily during switching to increase the output slew rate. The overshoot introduced by the extra current happens only during switching and does not ring, unlike the overshoot caused by signal reflection. The amount of pre-emphasis needed depends on the attenuation of the high-frequency component along the transmission line. The Quartus II software allows four settings for programmable pre-emphasis—zero (**0**), low (**1**), medium (**2**), and high (**3**). The default setting is low.

The V_{OD} is also programmable with four settings: low (**0**), medium low (**1**), medium high (**2**), and high (**3**). The default setting is medium low.

Programmable V_{OD}

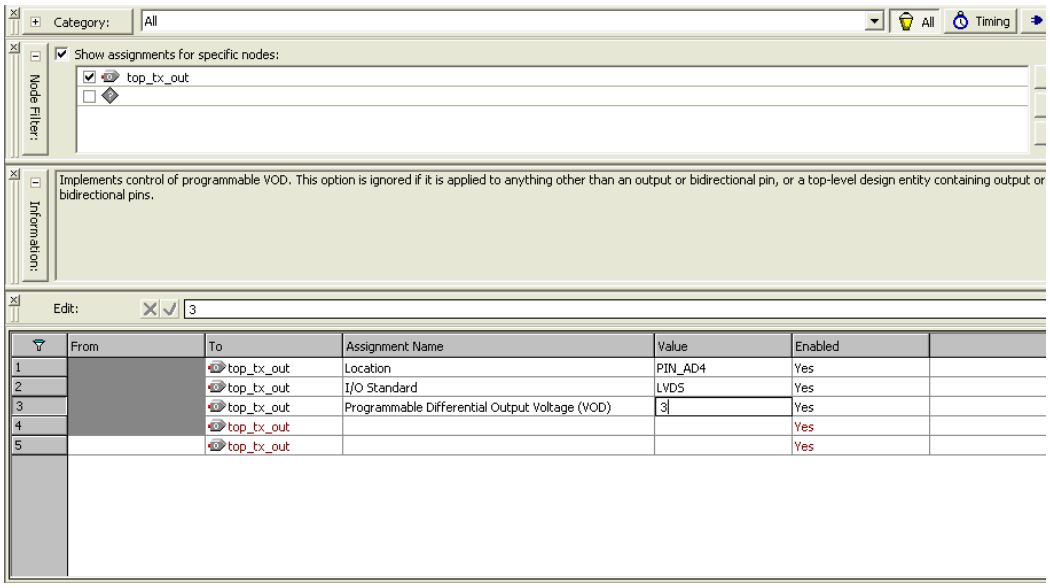
You can statically assign the V_{OD} settings from the Assignment Editor. Table 8–8 lists the assignment name for programmable V_{OD} and its possible values in the Quartus II software Assignment Editor.

Table 8–8. Quartus II Software Assignment Editor

To	`tx_out`
Assignment name	Programmable Differential Output Voltage (V_{OD})
Allowed values	0, 1, 2, 3

Figure 8–10 shows the assignment of programmable V_{OD} for a transmit data output from the Quartus II software Assignment Editor.

Figure 8–10. Quartus II Software Assignment Editor—Programmable V_{OD}



Programmable Pre-Emphasis

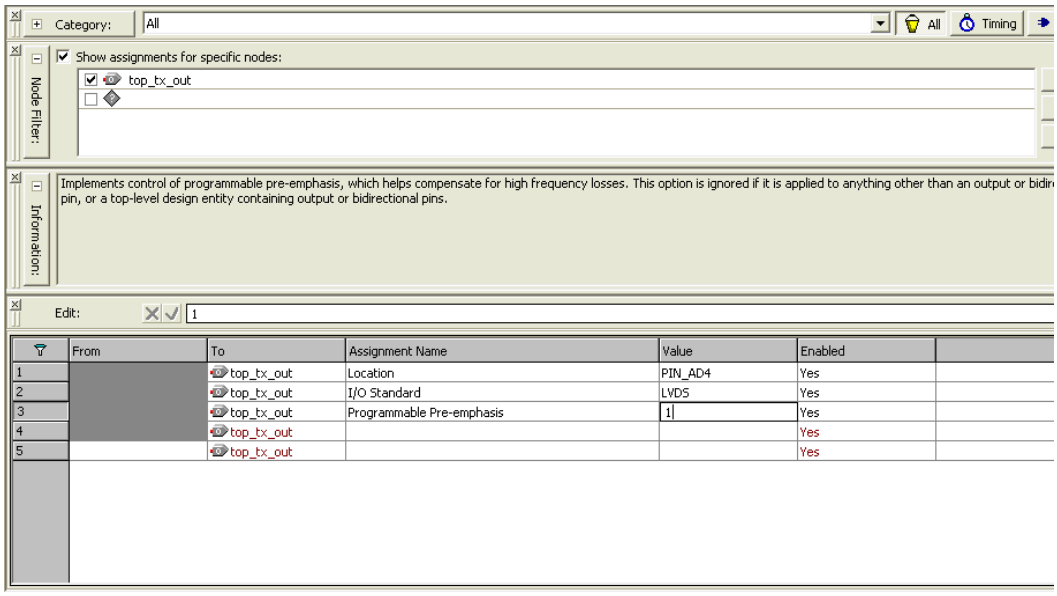
Four different settings are allowed for pre-emphasis from the Assignment Editor for each LVDS output channel. Table 8–9 lists the assignment name and its possible values for programmable pre-emphasis in the Quartus II software Assignment Editor.

Table 8–9. Quartus II Software Assignment Editor

To	tx_out
Assignment name	Programmable Pre-emphasis
Allowed values	0, 1, 2, 3

Figure 8–11 shows the assignment of programmable pre-emphasis for a transmit data output port from the Quartus II software Assignment Editor.

Figure 8–11. Quartus II Software Assignment Editor – Programmable Pre-Emphasis



Differential Receiver

The Stratix IV device family has a dedicated circuitry to receive high-speed differential signals in row I/Os. Figure 8–12 shows the hardware blocks of the Stratix IV receiver. The receiver has a differential buffer and left and right PLLs that can be shared between the transmitter and receiver, a DPA block, a synchronizer, a data realignment block, and a deserializer. The differential buffer can receive LVDS, mini-LVDS, and RSDS signal levels, which are statically set in the Quartus II software Assignment Editor.

The left and right PLL receives the external clock input and generates different phases of the same clock. The DPA block chooses one of the clocks from the left and right PLL and aligns the incoming data on each channel. The synchronizer circuit is a 1 bit wide by 6 bit deep FIFO buffer that compensates for any phase difference between the DPA clock and the data realignment block. If necessary, the user-controlled data realignment circuitry inserts a single bit of latency in the serial bit stream to align to the word boundary. The deserializer includes shift registers and parallel load registers, and sends a maximum of 10 bits to the internal logic.

The Stratix IV device family supports three different receiver modes:

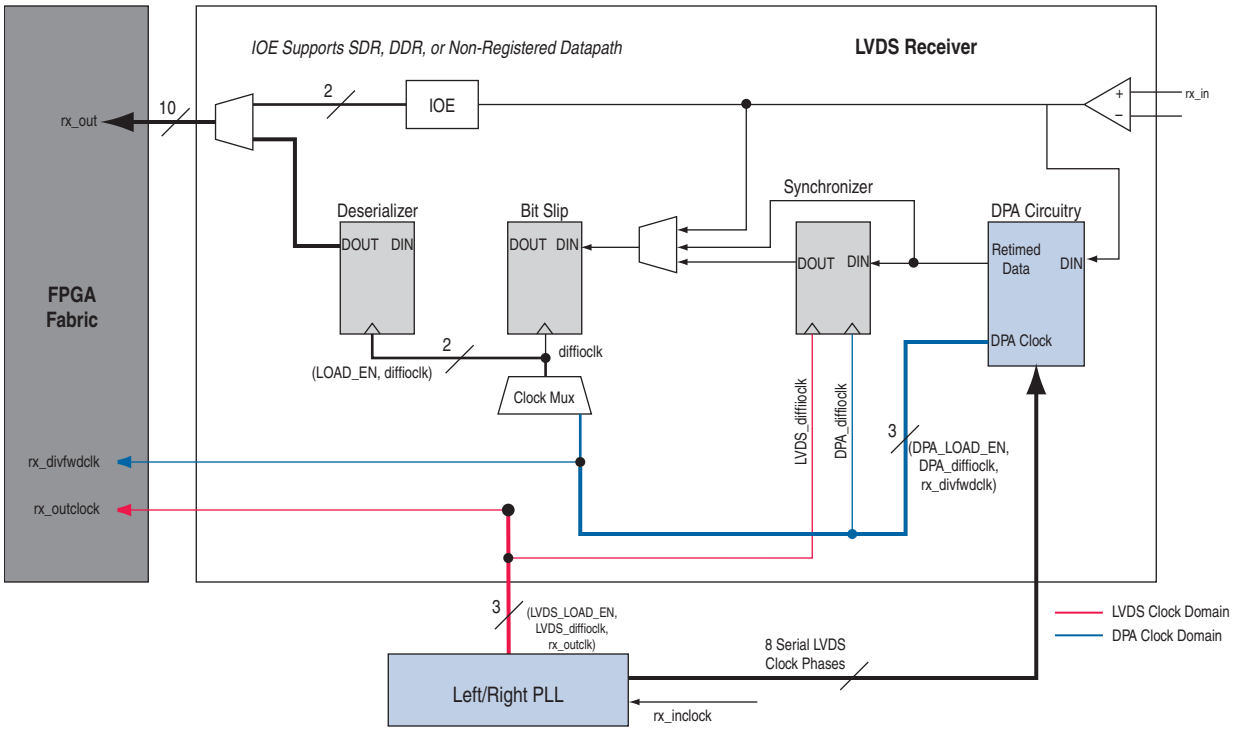
- Non-DPA mode
- DPA mode
- Soft-CDR mode

The physical medium connecting the transmitter and receiver LVDS channels may introduce skew between the serial data and the source-synchronous clock. The instantaneous skew between each LVDS channel and the clock also varies with the jitter on the data and clock signals as seen by the receiver. The three different modes—non-DPA, DPA, and soft-CDR—provide different options to overcome skew between the source synchronous clock (non-DPA, DPA) / reference clock (soft-CDR) and the serial data.

Only non-DPA mode requires manual skew adjustment.

Non-DPA mode allows you to statically select the optimal phase between the source synchronous clock and the received serial data to compensate skew. In DPA mode, the DPA circuitry automatically chooses the best phase to compensate for the skew between the source synchronous clock and the received serial data. Soft-CDR mode provides opportunities for synchronous and asynchronous applications for chip-to-chip and short reach board-to-board applications for SGMII protocols.

Figure 8–12. Receiver Block Diagram *(Note 1)*, *(2)*



Notes to Figure 8–12:

(1) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

(2) The `rx_out` port has a maximum data width of 10 bits.

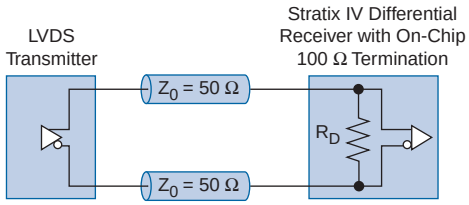
Differential I/O Termination

The Stratix IV device family provides a 100-Ω on-chip differential termination option on each differential receiver channel for LVDS standards. On-chip termination saves board space by eliminating the need to add external resistors on the board. You can enable on-chip termination in the Quartus II software Assignment Editor.

On-chip differential termination is supported on all row I/O pins and dedicated clock input pins (`CLK[0,2,9,11]`). It is not supported for column I/O pins, dedicated clock input pins (`CLK[1,3,8,10]`), or the corner PLL clock inputs.

Figure 8–13 shows device on-chip termination.

Figure 8–13. On-Chip Differential I/O Termination



Receiver Hardware Blocks

The differential receiver has the following hardware blocks:

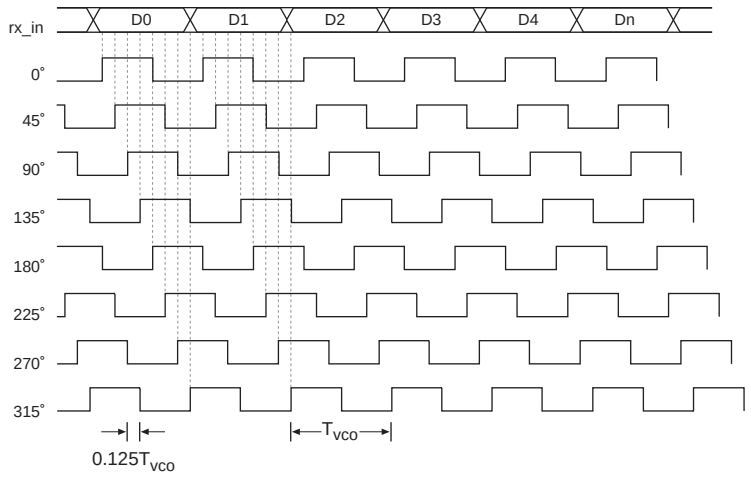
- DPA block
- Synchronizer
- Data realignment (bit slip)
- Deserializer

DPA Block

The DPA block takes in high-speed serial data from the differential input buffer and selects one of the eight phases generated by the left and right PLL to sample the data. The DPA chooses a phase closest to the phase of the serial data. The maximum phase offset between the received data and the selected phase is 1/8 UI, which is the maximum quantization error of the DPA. The eight phases of the clock are equally divided, offering a 45° resolution.

Figure 8–14 shows the possible phase relationships between the DPA clocks and the incoming serial data.

Figure 8–14. DPA Clock Phase to Serial Data Timing Relationship *(Note 1)*



Note to Figure 8–14:

(1) T_{VCO} is defined as the PLL serial clock period.

The DPA block continuously monitors the phase of the incoming serial data and selects a new clock phase if needed. You can prevent the DPA from selecting a new clock phase by asserting the optional RX_DPLL_HOLD port, which is available for each channel.

DPA circuitry does not require a fixed training pattern to lock to the optimum phase out of the eight phases. After reset or power up, DPA circuitry requires transitions on the received data to lock to the optimum phase. An optional output port, RX_DPA_LOCKED, is available to indicate an initial DPA lock condition to the optimum phase after power up or reset. This signal is not de-asserted if the DPA selects a new phase out of the eight clock phases to sample the received data. Do not use the rx_dpa_locked signal to determine a DPA loss-of-lock condition. Use data checkers such as a cyclic redundancy check (CRC) or diagonal interleaved parity (DIP-4) to validate the data.

An independent reset port, RX_RESET, is available to reset the DPA circuitry. DPA circuitry must be retrained after reset.

The DPA block is bypassed in non-DPA mode.

Synchronizer

The synchronizer is a 1 bit wide and 6 bit deep FIFO buffer that compensates for the phase difference between DPA_diffioclk, which is the optimal clock selected by the DPA block, and LVDS_diffioclk, which is produced by the left and right PLL. The synchronizer can only compensate for phase differences, not frequency differences between the data and the receiver's input reference clock.

An optional port, RX_FIFO_RESET, is available to the internal logic to reset the synchronizer. The synchronizer is automatically reset when the DPA first locks to the incoming data. Altera recommends using RX_FIFO_RESET to reset the synchronizer when the DPA signals a loss-of-lock condition and the data checker indicates corrupted received data.

The synchronizer circuit is bypassed in non-DPA and soft-CDR mode.

Data Realignment Block (Bit Slip)

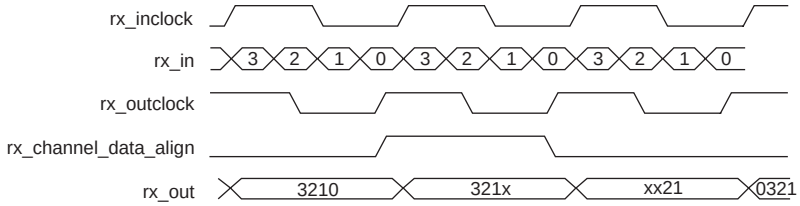
Skew in the transmitted data along with skew added by the link causes channel-to-channel skew on the received serial data streams. If the DPA is enabled, the received data is captured with different clock phases on each channel. This may cause the received data to be misaligned from channel-to-channel. To compensate for this channel-to-channel skew and establish the correct received word boundary at each channel, each receiver channel has a dedicated data realignment circuit that realigns the data by inserting bit latencies into the serial stream.

An optional RX_CHANNEL_DATA_ALIGN port controls the bit insertion of each receiver independently controlled from the internal logic. The data slips one bit on the rising edge of RX_CHANNEL_DATA_ALIGN. The requirements for the RX_CHANNEL_DATA_ALIGN signal include:

- The minimum pulse width is one period of the parallel clock in the logic array.
- The minimum low time between pulses is one period of the parallel clock.
- This is an edge-triggered signal.
- Valid data is available two parallel clock cycles after the rising edge of RX_CHANNEL_DATA_ALIGN.

Figure 8–15 shows receiver output (RX_OUT) after one bit slip pulse with the deserialization factor set to **4**.

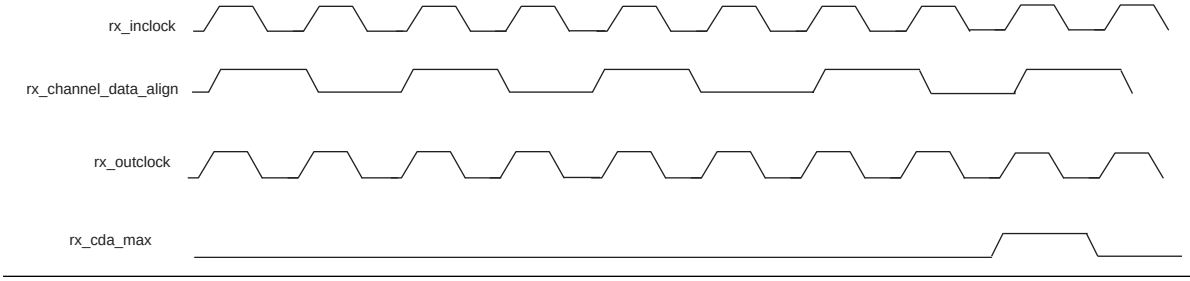
Figure 8–15. Data Realignment Timing



The data realignment circuit can have up to 11 bit-times of insertion before a rollover occurs. The programmable bit rollover point can be from 1 to 11 bit-times, independent of the deserialization factor. The programmable bit rollover point must be set equal to or greater than the deserialization factor, allowing enough depth in the word alignment circuit to slip through a full word. You can set the value of the bit rollover point using the MegaWizard Plug-In Manager software. An optional status port, RX_CDA_MAX, is available to the FPGA fabric from each channel to indicate when the preset rollover point is reached.

Figure 8–16 shows a preset value of four bit-times before rollover occurs. The `rx_cda_max` signal pulses for one `rx_outclock` cycle to indicate that rollover has occurred.

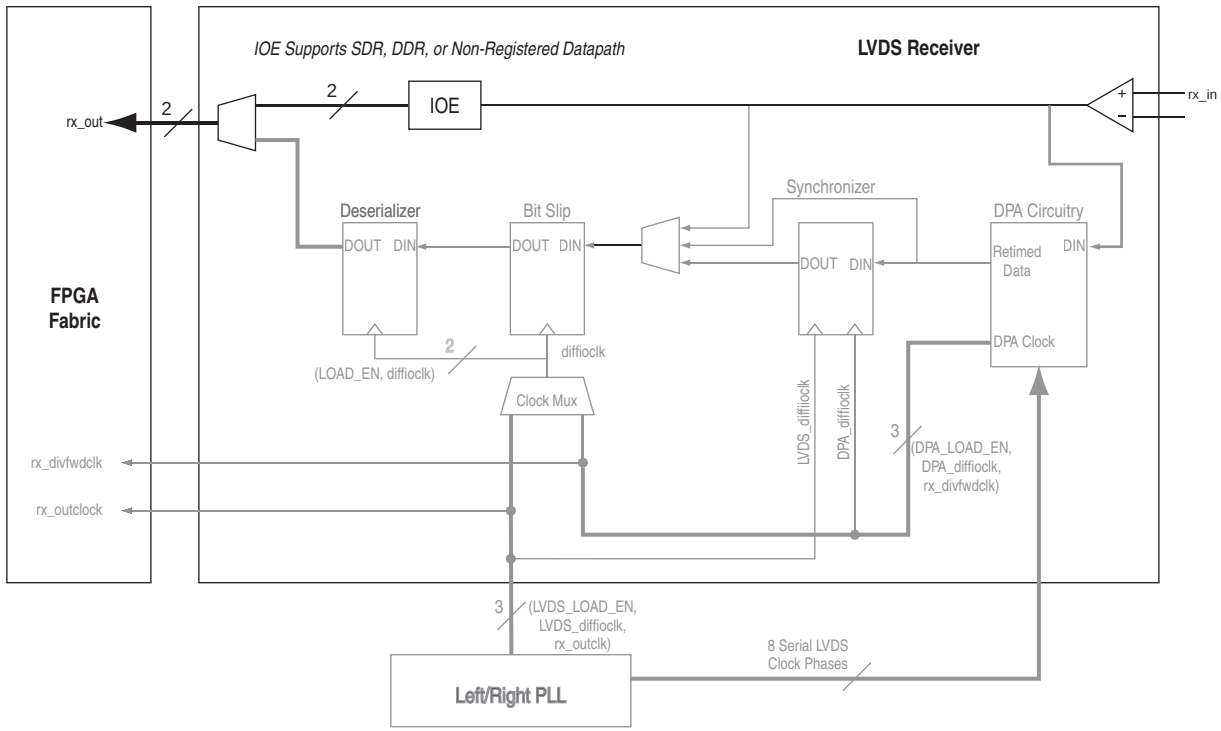
Figure 8–16. Receiver Data Re-alignment Rollover



Deserializer

You can statically set the deserialization factor to 4, 6, 7, 8, or 10 by using the Quartus II software. You can bypass the Stratix IV deserializer in the Quartus II MegaWizard Plug-In Manager software to support DDR (×2) or SDR (×1) operations, as shown Figure 8–17. The DPA and data realignment circuit cannot be used when the deserializer is bypassed. The IOE contains two data input registers that can operate in DDR or SDR mode.

Figure 8–17. Stratix IV Deserializer Bypass *(Note 1)*, *(2)*, *(3)*



Notes to Figure 8–17:

(1) All disabled blocks and signals are grayed out.
(2) In DDR mode, `rx_inclock` clocks the IOE register. In SDR mode, data is directly passed through the IOE.
(3) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

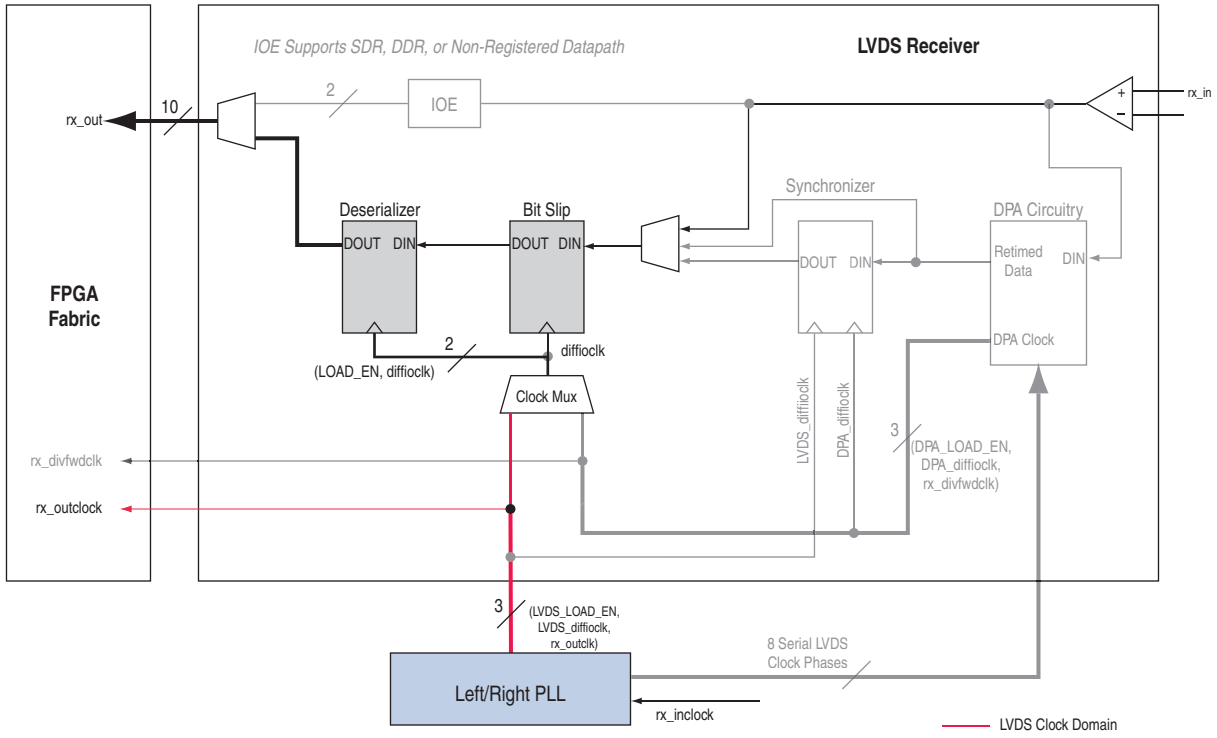
Receiver Data Path Modes

The Stratix IV device family supports three receiver datapath modes—non-DPA mode, DPA mode, and soft-CDR mode.

Non-DPA Mode

Figure 8–18 shows the non-DPA datapath block diagram. In non-DPA mode, the DPA and synchronizer blocks are disabled. Input serial data is registered at the rising or falling edge of the serial `LVDS_diffioclk` clock produced by the left and right PLL. You can select the rising/falling edge option using the ALTLDVS MegaWizard Plug-in Manager software. Both data realignment and deserializer blocks are clocked by the `LVDS_diffioclk` clock, which is generated by the left and right PLL.

Figure 8–18. Receiver Data Path in Non-DPA Mode *(Note 1)*, *(2)*



Notes to Figure 8–18:

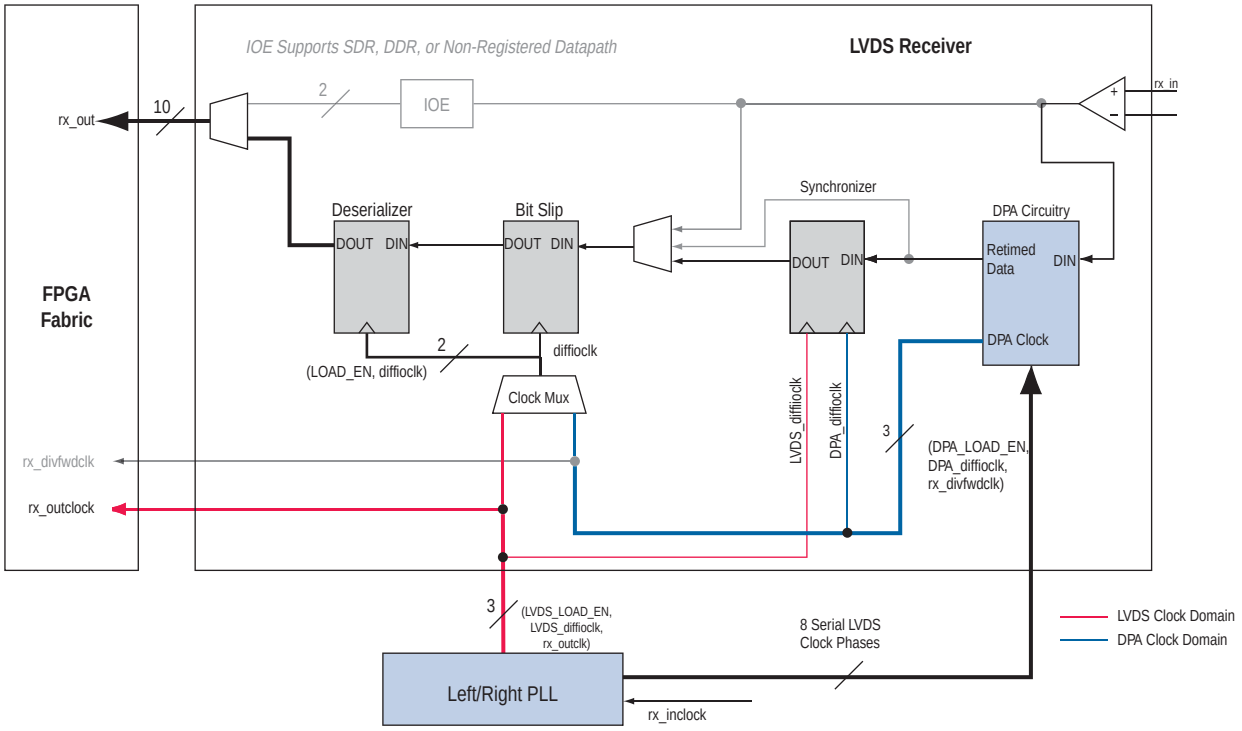
(1) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

(2) The `rx_out` port has a maximum data width of 10 bits.

DPA Mode

Figure 8–19 shows the DPA mode datapath, where all the hardware blocks mentioned in "Receiver Hardware Blocks" on page 8–18 are active. The DPA block chooses the best possible clock (`DPA_diffioclk`) from the eight fast clocks sent by the left and right PLL. This serial `DPA_diffioclk` clock is used for writing the serial data into the synchronizer. A serial `LVDS_diffioclk` clock is used for reading the serial data from the synchronizer. The same `LVDS_diffioclk` clock is used in data realignment and deserializer blocks.

Figure 8–19. Receiver Datapath in DPA Mode *(Note 1)*, *(2)*, *(3)*



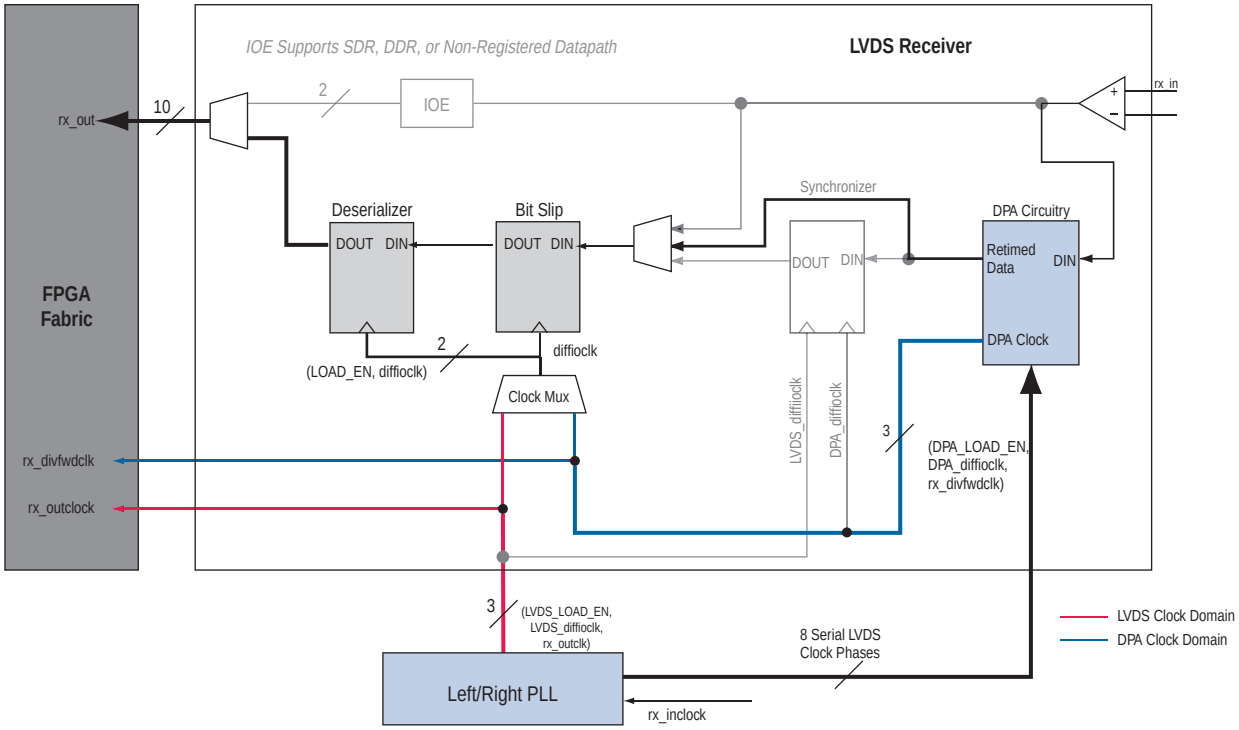
Notes to Figure 8–19:

(1) All disabled blocks and signals are grayed out.
(2) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.
(3) The `rx_out` port has a maximum data width of 10 bits.

Soft-CDR Mode

The Stratix IV LVDS channel offers the soft-CDR mode to support the Gigabit Ethernet and SGMII protocols. A receiver PLL uses the local clock source for reference. Figure 8–20 shows the soft-CDR mode datapath.

Figure 8–20. Receiver Datapath in Soft-CDR Mode *(Note 1), (2), (3)*



Notes to Figure 8–20:

(1) All disabled blocks and signals are grayed out.
(2) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.
(3) The `rx_out` port has a maximum data width of 10 bits.

In soft-CDR mode, the synchronizer block is inactive. The DPA circuitry selects an optimal DPA clock phase to sample the data. Use the selected DPA clock for bit-slip operation and deserialization. The DPA block also forwards the selected DPA clock, divided by the deserialization factor called `rx_divfwdclk`, to the FPGA fabric, along with the deserialized data. This clock signal is put on the periphery clock (PCLK) network. When using soft-CDR mode, the `rx_reset` port must not be asserted after the `rx_dpa_lock` is asserted because the DPA will continuously choose new phase taps from the PLL to track parts per million (PPM) differences between the reference clock and incoming data.

For more information about PCLK networks, refer to the *Clock Networks and PLLs in Stratix IV Devices* chapter.

You can use every LVDS channel in soft-CDR mode and can drive the FPGA fabric using the PCLK network in the Stratix IV device family. The `rx_dpa_locked` signal is not valid in soft-CDR mode, as the DPA continuously changes its phase to track PPM differences between the upstream transmitter and the local receiver input reference clocks. The parallel clock `rx_outclock`, generated by the left and right PLL, is also forwarded to the FPGA fabric.

LVDS Interface with the Use External PLL Option Enabled

The ALTLVDS MegaWizard Plug-In Manager software provides an option for implementing the LVDS interface with the **Use External PLL** option. With this option enabled, you can control the PLL settings, such as dynamically reconfiguring the PLL to support different data rates, dynamic phase shift, and other settings. You also must instantiate an ALTPLL megafunction to generate the various clock and load enable signals.

When you enable the **Use External PLL** option with the ALTLVDS transmitter and receiver, the following signals are required from the ALTPLL megafunction:

- Serial clock input to the SERDES of the ALTLVDS transmitter and receiver
- Load enable to the SERDES of the ALTLVDS transmitter and receiver
- Parallel clock used to clock the transmitter FPGA fabric logic and parallel clock used for the receiver `rx_syncclock` port and receiver FPGA fabric logic
- Asynchronous PLL reset port of the ALTLVDS receiver

As an example, Table 8–10 describes the serial clock output, load enable output, and parallel clock output generated on ports c0, c1, and c2, respectively, along with the locked signal of the ALTPLL instance. You can choose any of the PLL output clock ports to generate the interface clocks.

Table 8–10 lists the signal interface between the output ports of the ALTPLL megafunction and input ports of the ALTLVDS transmitter and receiver.

Table 8–10. Signal Interface Between ALTPLL and ALTLVDS Megafunctions

From the ALTPLL Megafunction	To the ALTLVDS Transmitter	To the ALTLVDS Receiver
Serial clock output (c0)	`tx_inclock` (serial clock input to the transmitter)	`rx_inclock` (serial clock input)
Load enable output (c1)	`tx_enable` (load enable to the transmitter)	`rx_enable` (load enable for the deserializer)
Parallel clock output (c2)	Parallel clock used inside the transmitter core logic in the FPGA fabric	`rx_syncclock` (parallel clock input) and parallel clock used inside the receiver core logic in the FPGA fabric
~(locked)	—	`pll_areset` (asynchronous PLL reset port) *(1)*

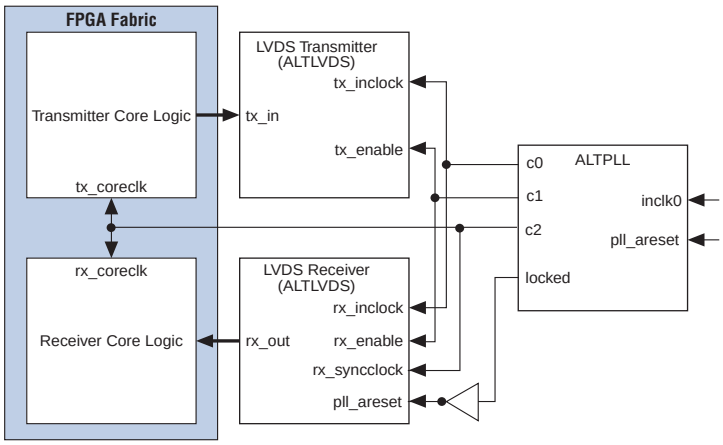
Note to Table 8–10:

(1) The `pll_areset` signal automatically is enabled for the LVDS receiver in external PLL mode. This signal does not exist for LVDS transmitter instantiation when the external PLL option is enabled.

The `rx_syncclock` port is enabled automatically in an LVDS receiver in external PLL mode. The Quartus II compiler errors out if this port is not connected as shown in Figure 8–21.

When generating the ALTPLL megafunction, the **Left/Right PLL** option is configured to set up the PLL in LVDS mode. Figure 8–21 shows the connection between the ALTPLL and ALTLVDS megafunctions.

Figure 8–21. LVDS Interface with the ALTPLL Megafunction *(Note 1)*



Note to Figure 8–21:

(1) Instantiation of `pll_areset` is optional for the ALTPLL instantiation.

Example 8–1 shows how to generate three output clocks using an ALTPLL megafunction.

Example 8–1. Generating Three Output Clocks Using an ALTPLL Megafunction

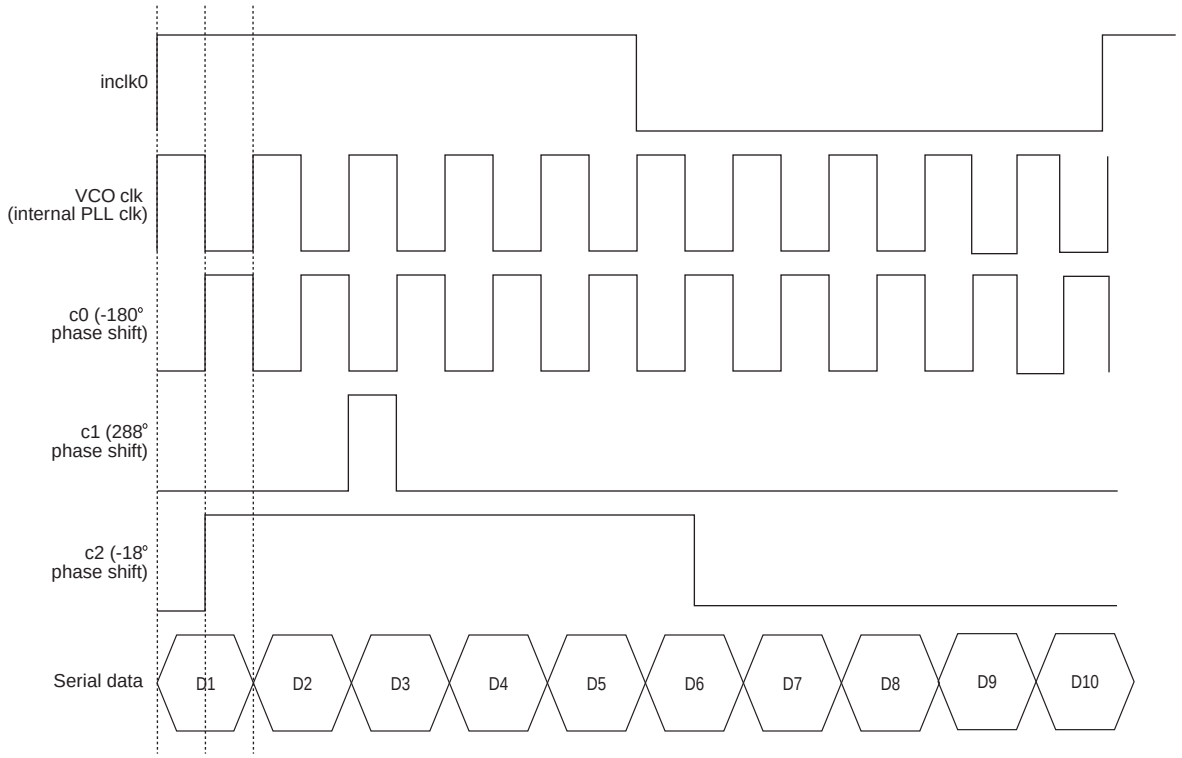
LVDS data rate = 1 Gbps; serialization factor = 10; input reference clock = 100 MHz

The following settings are used when generating the three output clocks using an ALTPLL megafunction. The serial clock must be **1000 MHz** and the parallel clock must be 100 MHz (serial clock divided by the serialization factor):

- c0
  - Frequency = **1000 MHz** (multiplication factor = 10 and division factor = 1)
  - Phase shift = **–180°** with respect to the voltage-controlled oscillator (VCO) clock
  - Duty cycle = **50%**
- c1
  - Frequency = (1000/10) = **100 MHz** (multiplication factor = 1 and division factor = 1)
  - Phase shift = (10 - 2) × 360/10 = **288°** [(deserialization factor - 2)/deserialization factor] × 360°
  - Duty cycle = (100/10) = **10%** (100 divided by the serialization factor)
- c2
  - Frequency = (1000/10) = **100 MHz** (multiplication factor = 1 and division factor = 1)
  - Phase shift = (–180/10) = **–18°** (c0 phase shift divided by the serialization factor)
  - Duty cycle = **50%**

The Equation 8–1 calculations for phase shift assume that the input clock and serial data are edge aligned. Introducing a phase shift of –180° to sampling clock (c0) ensures that the input data is center-aligned with respect to the c0, as shown in Figure 8–22.

Figure 8–22. Phase Relationship for External PLL Interface Signals



Left and Right PLLs (PLL_Lx and PLL_Rx)

The Stratix IV device family contains up to eight left and right PLLs with up to four PLLs located on the left side and four on the right side of the device. The left PLLs can support high-speed differential I/O banks on the left side; the right PLLs can support high-speed differential I/O banks on the right side of the device. The high-speed differential I/O receiver and transmitter channels use these left and right PLLs to generate the parallel clocks (`rx_outclock` and `tx_outclock`) and high-speed clocks (`diffioclk`).

Figure 8–2 on page 8–3 and Figure 8–3 on page 8–4 show the locations of the left and right PLLs for Stratix IV E, GT, and GX devices, respectively. The PLL VCO operates at the clock frequency of the data rate. Clock switchover and dynamic reconfiguration are allowed using the left and right PLL in high-speed differential I/O support mode.

For more information, refer to the *Clock Network and PLLs in Stratix IV Devices* chapter.

Stratix IV Clocking

The left and right PLLs feed into the differential transmitter and receive channels through the LVDS and DPA clock network. The center left and right PLLs can clock the transmitter and receive channels above and below them. The corner left and right PLLs can drive I/Os in the banks adjacent to them.

Figure 8–23 shows center PLL clocking in the Stratix IV device family. For more information about PLL clocking restrictions, refer to "Differential Pin Placement Guidelines" on page 8–37.

Figure 8–23. LVDS/DPA Clocks in the Stratix IV Device Family with Center PLLs

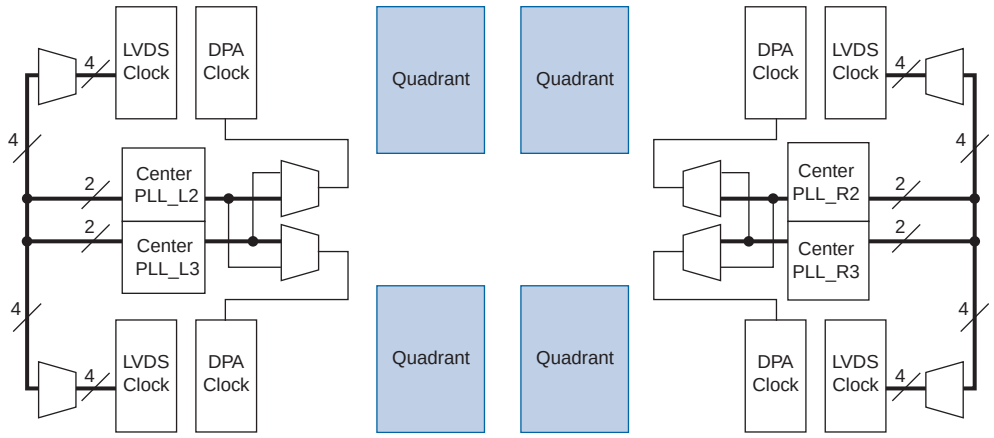
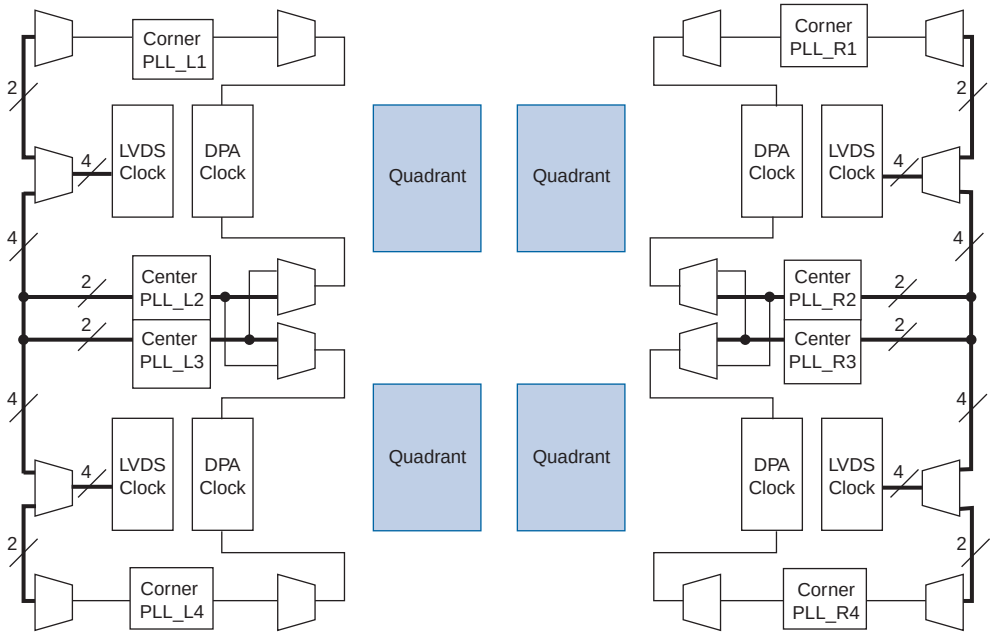


Figure 8–24 shows center and corner PLL clocking in the Stratix IV device family. For more information about PLL clocking restrictions, refer to "Differential Pin Placement Guidelines" on page 8–37.

Figure 8–24. LVDS/DPA Clocks in the Stratix IV Device Family with Center and Corner PLLs



Source-Synchronous Timing Budget

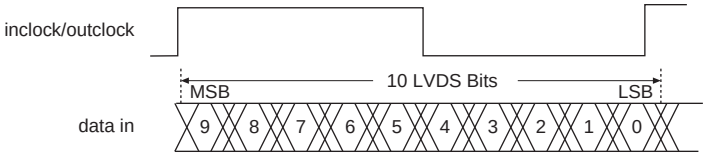
This section describes the timing budget, waveforms, and specifications for source-synchronous signaling in the Stratix IV device family. LVDS I/O standards enable high-speed data transmission. This high data transmission rate results in better overall system performance. To take advantage of fast system performance, it is important to understand how to analyze timing for these high-speed signals. Timing analysis for the differential block is different from traditional synchronous timing analysis techniques.

Instead of focusing on clock-to-output and setup times, source synchronous timing analysis is based on the skew between the data and the clock signals. High-speed differential data transmission requires the use of timing parameters provided by IC vendors and is strongly influenced by board skew, cable skew, and clock jitter. This section defines the source-synchronous differential data orientation timing parameters, the timing budget definitions for the Stratix IV device family, and how to use these timing parameters to determine a design's maximum performance.

Differential Data Orientation

There is a set relationship between an external clock and the incoming data. For operations at 1 Gbps and a serialization factor of 10, the external clock is multiplied by 10. You can set phase-alignment in the PLL to coincide with the sampling window of each data bit. The data is sampled on the falling edge of the multiplied clock. Figure 8–25 shows the data bit orientation of the ×10 mode.

Figure 8–25. Bit Orientation in the Quartus II Software



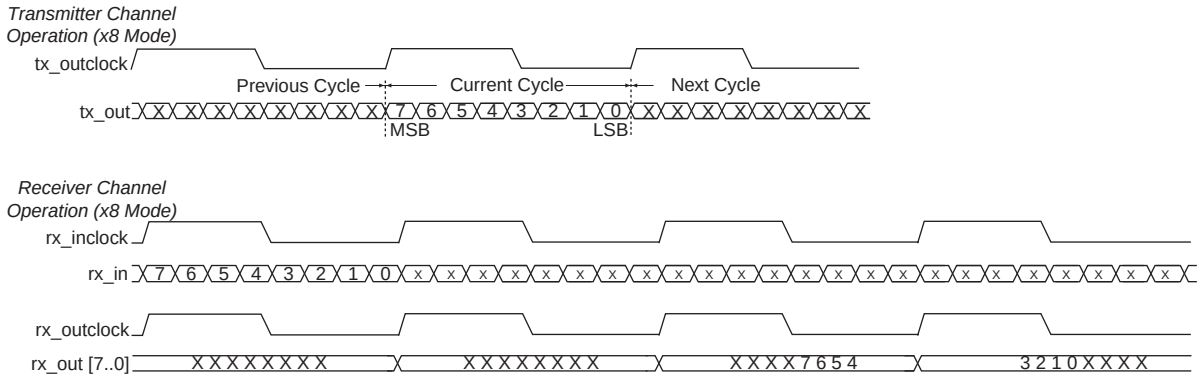
Differential I/O Bit Position

Data synchronization is necessary for successful data transmission at high frequencies. Figure 8–26 shows the data bit orientation for a channel operation. This figure is based on the following:

- Serialization factor equals the clock multiplication factor
- Edge alignment is selected for phase alignment
- Implemented in hard SERDES

For other serialization factors, use the Quartus II software tools to find the bit position within the word. The bit positions after deserialization are listed in Table 8–11.

Figure 8–26. Bit-Order and Word Boundary for One Differential Channel *(Note 1)*



Note to Figure 8–26:

(1) These are only functional waveforms and are not intended to convey timing information.

Table 8–11 lists the conventions for differential bit naming for 18 differential channels. The MSB and LSB positions increase with the number of channels used in a system.

Table 8–11. Differential Bit Naming

Receiver Channel Data Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
1	7	0
2	15	8
3	23	16
4	31	24
5	39	32
6	47	40
7	55	48
8	63	56
9	71	64
10	79	72
11	87	80
12	95	88
13	103	96
14	111	104
15	119	112
16	127	120
17	135	128
18	143	136

Transmitter-Channel-to-Channel Skew

Transmitter-channel-to-channel skew (TCCS) is an important parameter based on the Stratix IV transmitter in a source synchronous differential interface. This parameter is used in receiver skew margin calculation. For more information, refer to "Receiver Skew Margin for Non-DPA Mode" on page 8–32.

TCCS is the difference between the fastest and slowest data output transitions, including the TCO variation and clock skew. For LVDS transmitters, the TimeQuest Timing Analyzer provides a TCCS report, which shows TCCS values for serial output ports.

You can get the TCCS value from the TCCS report (`report_TCCS`) in the Quartus II compilation report under the TimeQuest Timing Analyzer, or from the *DC and Switching Characteristics* chapter.

Receiver Skew Margin for Non-DPA Mode

Changes in system environment, such as temperature, media (cable, connector, or PCB), and loading effect the receiver's setup and hold times; internal skew affects the sampling ability of the receiver.

Different modes of LVDS receivers use different specifications, that can help in deciding the ability to sample the received serial data correctly. In DPA mode, you must use DPA jitter tolerance instead of receiver input skew margin (RSKM).

In non-DPA mode, use TCCS, RSKM, and sampling window (SW) specifications for high-speed source-synchronous differential signals in the receiver data path. The relationship between RSKM, TCCS, and SW can be expressed by the RSKM equation shown in Equation 8–1.

Equation 8–1. RSKM

$$RSKM = \frac{TUI - SW - TCCS}{2}$$

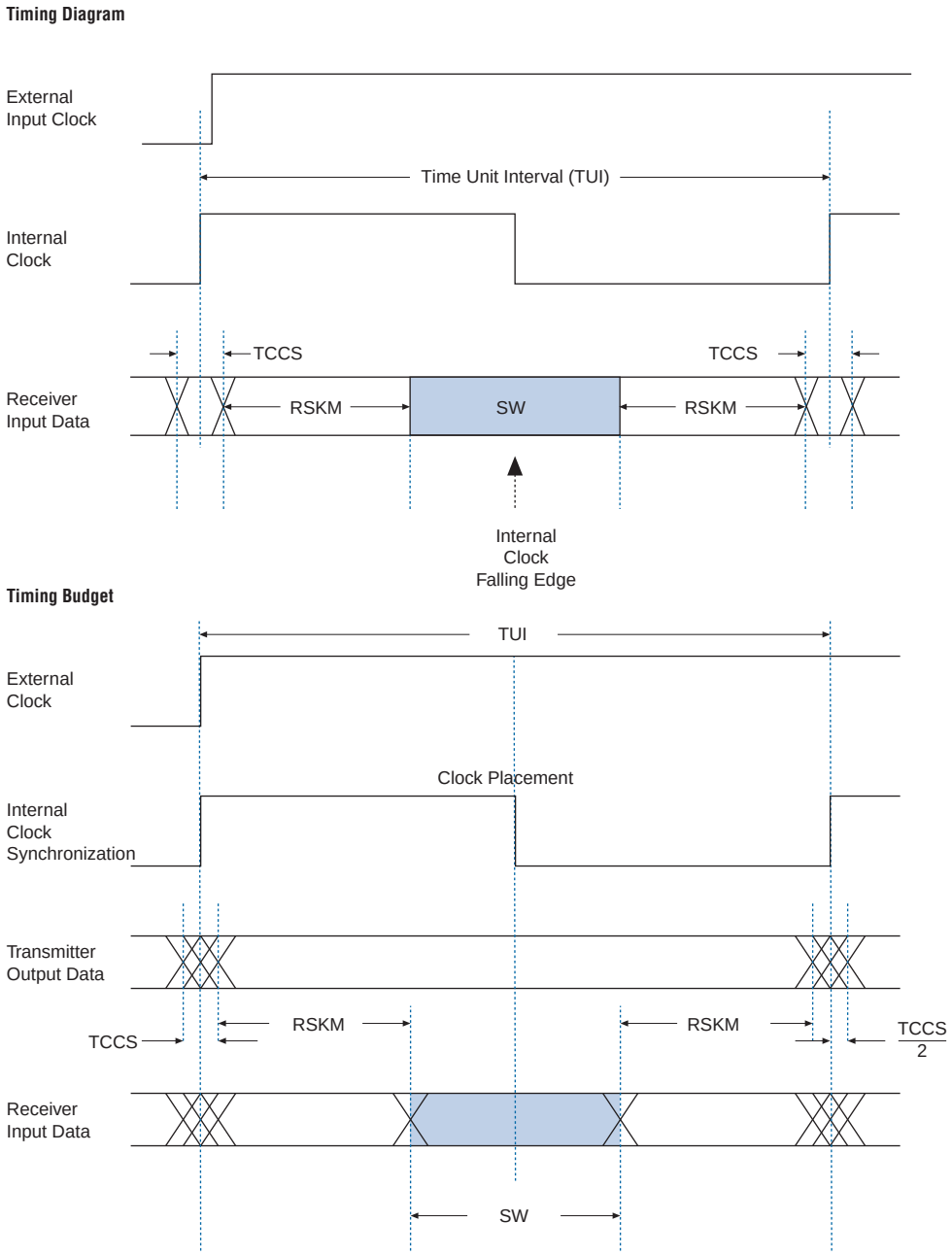
Conventions used for the equation:

- Time unit interval (TUI)—Time period of the serial data.
- RSKM—The timing margin between the receiver's clock input and the data input sampling window.
- SW—The period of time that the input data must be stable to ensure that data is successfully sampled by the LVDS receiver. The SW is a device property and varies with device speed grade.
- TCCS—The timing difference between the fastest and the slowest output edges, including t_{CO} variation and clock skew, across channels driven by the same PLL. The clock is included in the TCCS measurement.

Figure 8–27 shows the relationship between the RSKM, TCCS, and the receiver's SW.

You must calculate the RSKM value to decide whether or not data can be sampled properly by the LVDS receiver with the given data rate and device. A positive RSKM value indicates that the LVDS receiver can sample the data properly, whereas a negative RSKM indicates that it cannot.

Figure 8–27. Differential High-Speed Timing Diagram and Timing Budget for Non-DPA Mode



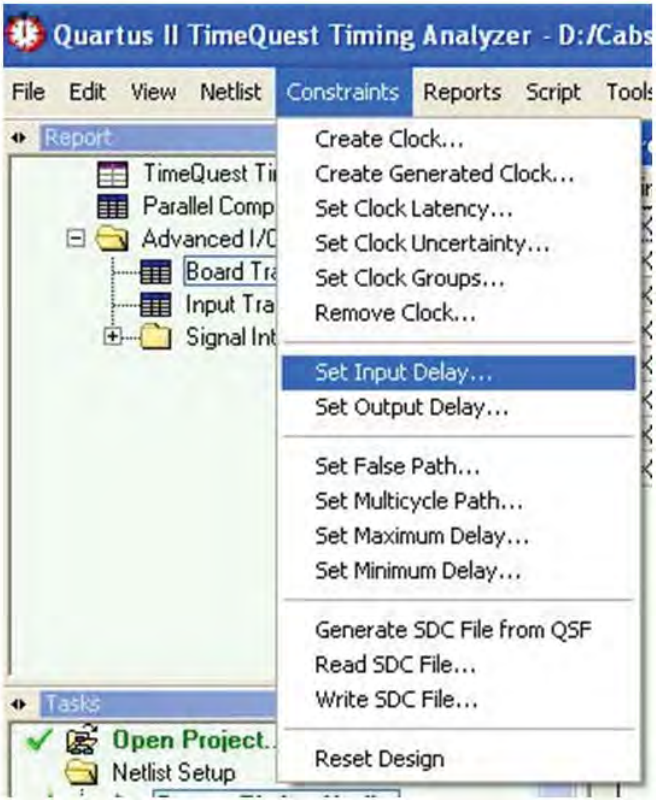
For LVDS receivers, the Quartus II software provides an RSKM report showing the SW, TUI, and RSKM values for non-DPA mode. You can generate the RSKM report by executing the `report_RSKM` command in the TimeQuest Timing Analyzer. You can find the RSKM report in the Quartus II compilation report under the TimeQuest Timing Analyzer section.

In order to obtain the RSKM value, you must assign an appropriate input delay to the LVDS receiver through the TimeQuest Timing Analyzer constraints menu.

For assigning input delay, follow these steps:

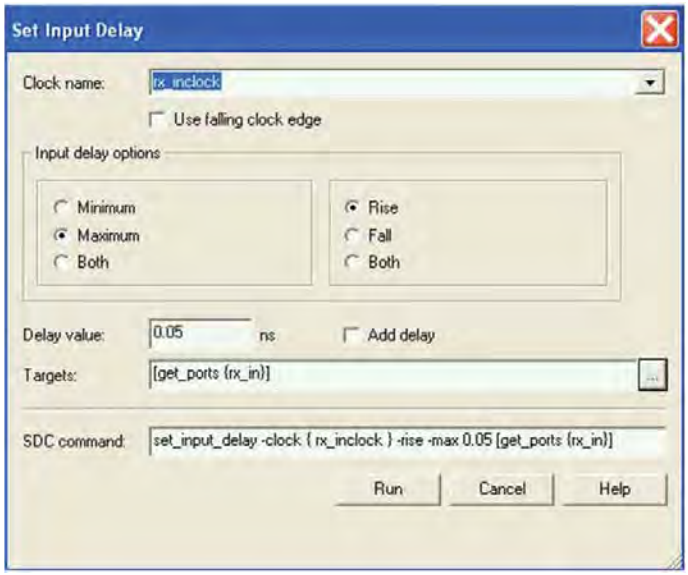
1. The Quartus II TimeQuest Timing Analyzer GUI has many options for setting the constraints and analyzing the design. Figure 8–28 shows various commands on the Constraints menu. For setting input delay, you must select the **Set Input Delay** option.

Figure 8–28. Selection of Constraint Menu in TimeQuest Timing Analyzer



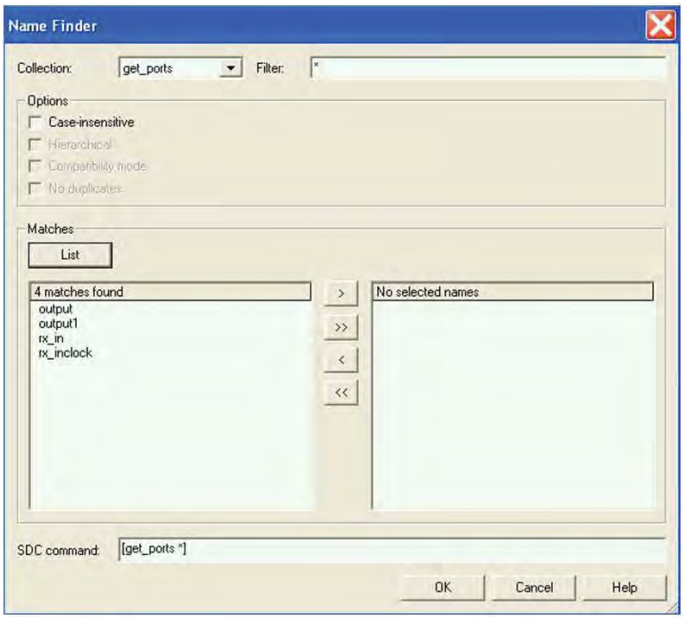
2. Figure 8–29 shows the setting parameters for the **Set Input Delay** option. The clock name must reference the source synchronous clock that feeds the LVDS receiver. Select the desired clock using the pull down menu.

Figure 8–29. Input Time Delay Assignment Through TimeQuest Timing Analyzer



3. Figure 8–30 shows the **Targets** option. You can view a list of all available ports using the **List** option in the **Name Finder** window.

Figure 8–30. Name Finder Window in Set Input Delay Option



4. Select the LVDS receiver serial input ports (from the list) according to the input delay you set. Click **OK**.

5. In the **Set Input Delay** window, set the appropriate values in the **Input Delay Options** section and **Delay** value.
6. Click **Run** to incorporate these values in the TimeQuest Timing Analyzer.
7. Assign the appropriate delay for all the LVDS receiver input ports following these steps. If you have already assigned **Input Delay** and you need to add more delay to that input port, use the **Add Delay** option in the **Set Input Delay** window.

If no input delay is set in the TimeQuest Timing Analyzer, the receiver-channel-to-channel skew (RCCS) defaults to zero. You can also directly set the input delay in a synopsys design constraint file (**.sdc)** using the `set_input_delay` command.

For more information about **.sdc** commands and the TimeQuest Timing Analyzer, refer to the *Quartus II TimeQuest Timing Analyzer* chapter in volume 3 of the *Quartus II Development Software Handbook*.

Example 8–2 shows the RSKM calculation.

Example 8–2. RSKM

Data Rate: 1 Gbps, Board channel-to-channel skew = 200 ps

For Stratix IV devices:

TCCS = **100 ps** (pending characterization)
SW = **300 ps** (pending characterization)

TUI = **1000 ps**
Total RCCS = TCCS + Board channel-to-channel skew= 100 ps + 200 ps
= **300 ps**

RSKM = **TUI - SW - RCCS**
= **1000 ps - 300 ps - 300 ps**
= **400 ps > 0**

Because the RSKM > 0 ps, receiver non-DPA mode must work correctly.

You can also calculate RSKM using the steps described in "Guidelines for DPA-Enabled Differential Channels" on page 8–37.

Differential Pin Placement Guidelines

To ensure proper high-speed operation, differential pin placement guidelines have been established. The Quartus II compiler automatically checks that these guidelines are followed and issues an error message if they are not met.

This section is divided into pin placement guidelines with and without DPA usage because DPA usage adds some constraints on the placement of high-speed differential channels.

DPA-enabled differential channels refer to DPA mode or soft-CDR mode; DPA disabled channels refer to non-DPA mode.

Guidelines for DPA-Enabled Differential Channels

The Stratix IV device family has differential receivers and transmitters in I/O banks on the left and right sides of the device. Each receiver has a dedicated DPA circuit to align the phase of the clock to the data phase of its associated channel. When you use DPA-enabled channels in differential banks, you must adhere to the guidelines listed in the following sections.

DPA-Enabled Channels and Single-Ended I/Os

When you enable a DPA channel in a bank, both single-ended I/Os and differential I/O standards are allowed in the bank.

- Single-ended I/Os are allowed in the same I/O bank, as long as the single-ended I/O standard uses the same V_{CCIO} as the DPA-enabled differential I/O bank.
- Single-ended inputs can be in the same logic array block (LAB) row as a differential channel using the SERDES circuitry.
- DDIO can be placed within the same LAB row as a SERDES differential channel but half rate DDIO (single data rate) output pins cannot be placed within the same LAB row as a receiver SERDES differential channel. The input register must be implemented within the FPGA fabric logic.

DPA-Enabled Channel Driving Distance

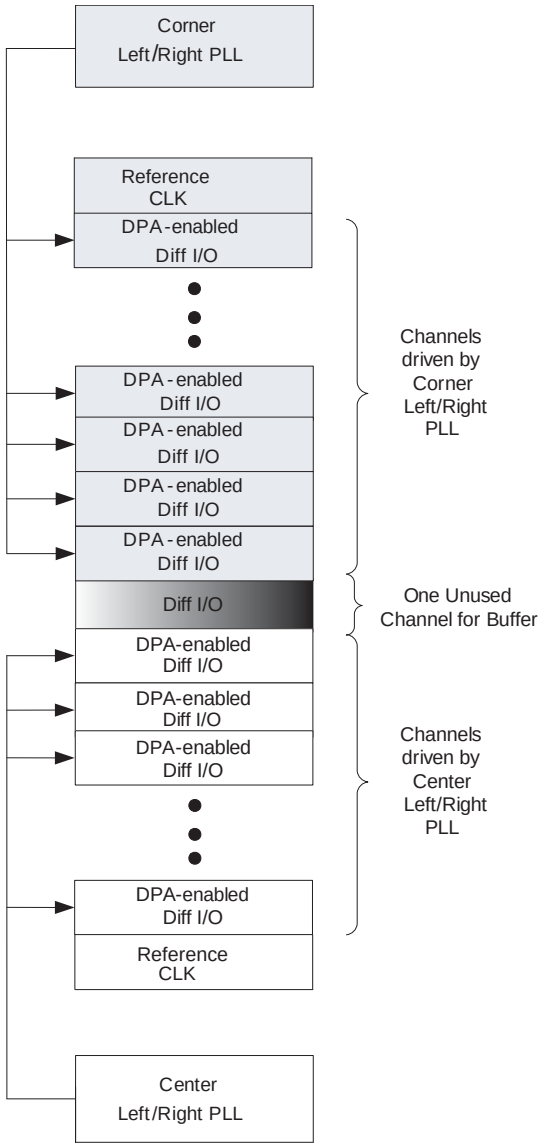
If the number of DPA channels driven by each left and right PLL exceeds 25 LAB rows, Altera recommends implementing data realignment (bit slip) circuitry for all the DPA channels.

Using Corner and Center Left and Right PLLs

If a differential bank is being driven by two left and right PLLs, where the corner left and right PLL is driving one group and the center left and right PLL is driving another group, there must be at least one row of separation between the two groups of DPA-enabled channels (refer to Figure 8–31). The two groups can operate at independent frequencies.

You do not need a separation if a single left and right PLL is driving the DPA-enabled channels as well as DPA-disabled channels.

Figure 8–31. Corner and Center Left and Right PLLs Driving DPA-Enabled Differential I/Os in the Same Bank



Using Both Center Left and Right PLLs

You can use both center left and right PLLs to drive DPA-enabled channels simultaneously, as long as they drive these channels in their adjacent banks only, as shown in Figure 8–32.

If one of the center left and right PLLs drives the top and bottom banks, you cannot use the other center left and right PLL to drive differential channels, as shown in Figure 8–32.

If the top PLL_L2 and PLL_R2 drives DPA-enabled channels in the lower differential bank, the PLL_L3 and PLL_R3 cannot drive DPA-enabled channels in the upper differential banks and vice versa. In other words, the center left and right PLLs cannot drive cross-banks simultaneously, as shown in Figure 8–33.

Figure 8–32. Center Left and Right PLLs Driving DPA-Enabled Differential I/Os

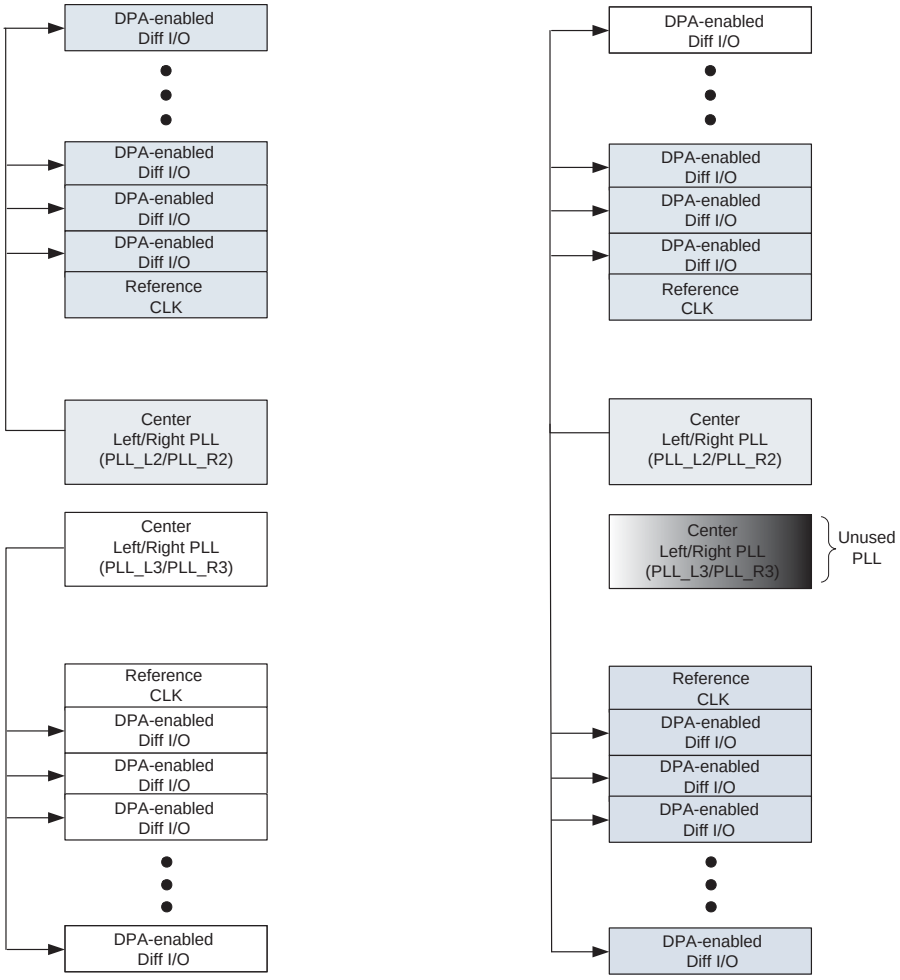
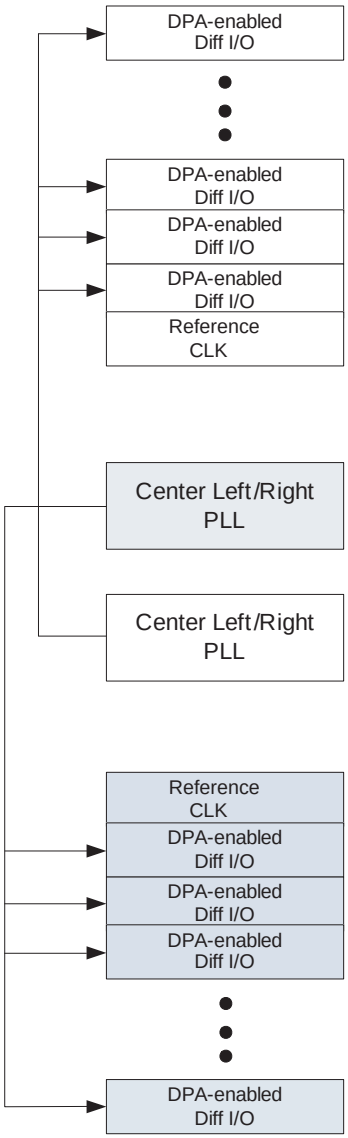


Figure 8–33. Invalid Placement of DPA-Enabled Differential I/Os Driven by Both Center Left and Right PLLs



Guidelines for DPA-Disabled Differential Channels

When you use DPA-disabled channels in the left and right banks of a Stratix IV device, you must adhere to the guidelines in the following sections.

DPA-Disabled Channels and Single-Ended I/Os

The placement rules for DPA-disabled channels and single-ended I/Os are the same as those for DPA-enabled channels and single-ended I/Os.

DPA-Disabled Channel Driving Distance

Each left and right PLL can drive all the DPA-disabled channels in the entire bank.

Using Corner and Center Left and Right PLLs

You can use a corner left and right PLL to drive all transmitter channels and a center left and right PLL to drive all DPA-disabled receiver channels within the same differential bank. In other words, a transmitter channel and a receiver channel in the same LAB row can be driven by two different PLLs, as shown in Figure 8–34.

A corner left and right PLL and a center left and right PLL can drive duplex channels in the same differential bank, as long as the channels driven by each PLL are not interleaved. Separation is not necessary between the group of channels driven by the corner and center left and right PLLs, as shown in Figure 8–34 and Figure 8–35.

Figure 8–34. Corner and Center Left and Right PLLs Driving DPA-Disabled Differential I/Os in the Same Bank

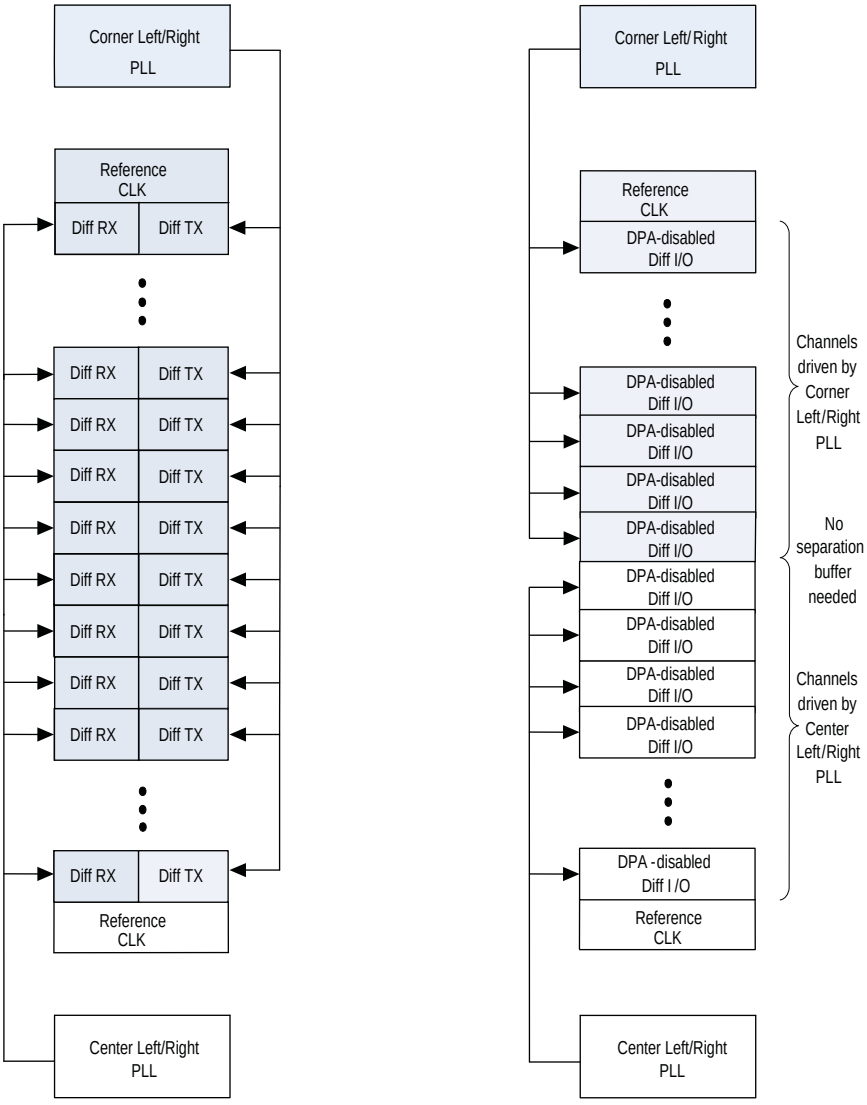
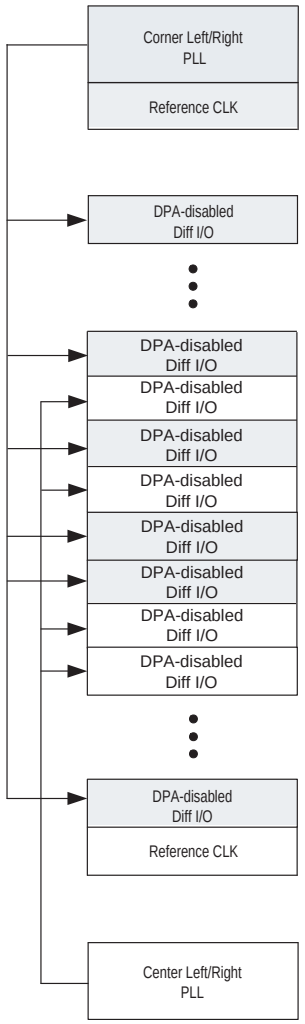


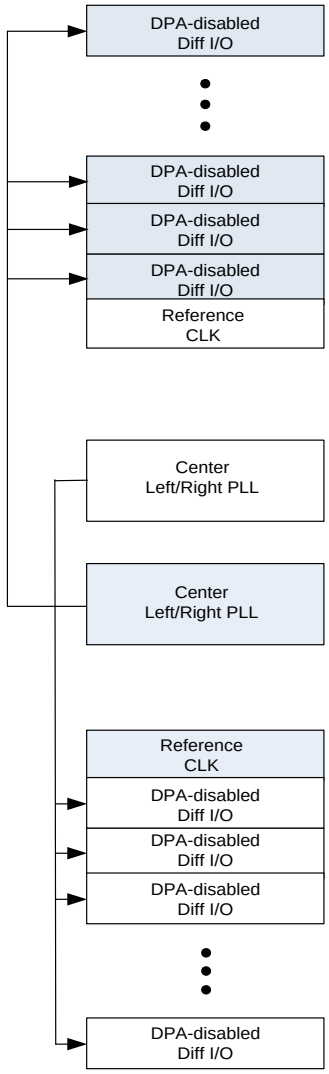
Figure 8–35. Invalid Placement of DPA-Disabled Differential I/Os Due to Interleaving of Channels Driven by the Corner and Center Left and Right PLLs



Using Both Center Left and Right PLLs

You can use both center left and right PLLs simultaneously to drive DPA-disabled channels on upper and lower differential banks. Unlike DPA-enabled channels, the center left and right PLLs can drive cross-banks. For example, the upper-center left and right PLL can drive the lower differential bank at the same time the lower center left and right PLL is driving the upper differential bank, and vice versa, as shown in Figure 8–36.

Figure 8–36. Both Center Left and Right PLLs Driving Cross-Bank DPA-Disabled Channels Simultaneously



Chapter Revision History

Table 8–12 lists the revision history for this chapter.

Table 8–12. Chapter Revision History

Date and Document Version	Changes Made	Changes Made
November 2009 v3.0	■ Changed "dedicated LVDS" to "true LVDS". ■ Removed EP4SE110, EP4SE290, and EP4SE680 devices. ■ Added EP4SE820 and Stratix IV GT devices. ■ Updated "LVDS Channels", "Differential Transmitter", "Soft-CDR Mode", and "DPA-Enabled Channels and Single-Ended I/Os" sections. ■ Updated Table 8–1, Table 8–2, Table 8–5, and Table 8–6. ■ Added Table 8–3 and Table 8–4. ■ Updated Example 8–1. ■ Updated Figure 8–22. ■ Minor text edits.	—
June 2009 v2.3	■ Added an introductory paragraph to increase search ability. ■ Minor text edits.	—
April 2009 v2.2	■ Updated "Introduction". ■ Updated Figure 8–3. ■ Removed Table 8-5 and Table 8-6.	—
March 2009 v2.1	■ Updated "Introduction", "Stratix IV LVDS Channels", "Stratix IV Differential Transmitter", "Differential I/O Termination", and "Dynamic Phase Alignment (DPA) Block" sections. ■ Updated Table 8–1, Table 8–2, Table 8–3, Table 8–4, and Table 8–7. ■ Added Table 8–5 and Table 8–6. ■ Updated Figure 8–2. ■ Removed "Referenced Documents" section.	—
November 2008 v2.0	■ Updated Figure 8–2, Figure 8–3, Figure 8–21, Figure 8–34. ■ Removed Figure 8–31. ■ Updated Table 8–1, Table 8–10. ■ Updated "Differential Pin Placement Guidelines" section.	—
May 2008 v1.0	Initial Release.	—

Section III. System Integration

This section includes the following chapters:

- Chapter 9, Hot Socketing and Power-On Reset in Stratix IV Devices
- Chapter 10, Configuration, Design Security, and Remote System Upgrades in Stratix IV Devices
- Chapter 11, SEU Mitigation in Stratix IV Devices
- Chapter 12, JTAG Boundary-Scan Testing in Stratix IV Devices
- Chapter 13, Power Management in Stratix IV Devices

Revision History

Refer to each chapter for its own specific revision history. For information on when each chapter was updated, refer to the Chapter Revision Dates section, which appears in the full handbook.

9. Hot Socketing and Power-On Reset in Stratix IV Devices

SIV51009-3.0

This chapter describes information about hot-socketing specifications, power-on reset (POR) requirements, and their implementation in Stratix® IV devices.

Stratix IV devices offer hot socketing, also known as hot plug-in or hot swap, and power sequencing support without the use of external devices. You can insert or remove a Stratix IV device or a board in a system during system operation without causing undesirable effects to the running system bus or board that is inserted into the system.

The hot-socketing feature also removes some of the difficulty when you use Stratix IV devices on PCBs that contain a mixture of 3.0-, 2.5-, 1.8-, 1.5-, and 1.2-V devices. With the Stratix IV hot-socketing feature, you no longer have to ensure a proper power-up sequence for each device on the board.

The Stratix IV hot-socketing feature provides:

- Board or device insertion and removal without external components or board manipulation
- Support for any power-up sequence
- I/O buffers non-intrusive to system buses during hot insertion

This section also describes POR circuitry in Stratix IV devices. POR circuitry keeps the devices in the reset state until the power supply outputs are within operating range.

This chapter contains the following sections:

Stratix IV Hot-Socketing Specifications

Stratix IV devices are hot-socketing compliant without the need for external components or special design requirements. Hot-socketing support in Stratix IV devices has the following advantages:

- You can drive the device before power-up without damaging it.
- I/O pins remain tri-stated during power-up. The device does not drive out before or during power-up, thereby not affecting other buses in operation.
- You can insert a Stratix IV device into or remove it from a powered-up system board without damaging the system board or interfering with its operation.

Stratix IV Devices can be Driven Before Power Up

You can drive signals into I/O pins, dedicated input pins, and dedicated clock pins of Stratix IV devices before or during power up or power down without damaging the device. Stratix IV devices support power up or power down of the V_{CCIO}, V_{CC}, V_{CCPGM}, and V_{CCPD} power supplies in any sequence in order to simplify system-level design.

I/O Pins Remain Tri-Stated During Power Up

A device that does not support hot socketing can interrupt system operation or cause contention by driving out before or during power up. In a hot-socketing situation, the Stratix IV device's output buffers are turned off during system power up or power down. Also, the Stratix IV device does not drive out until the device is configured and working within the recommended operating conditions.

Insertion or Removal of a Stratix IV Device from a Powered-Up System

Devices that do not support hot socketing can short power supplies when powered up through the device signal pins. This irregular power up can damage both the driving and driven devices and can disrupt card power up.

You can insert a Stratix IV device into or remove it from a powered-up system board without damaging the system board or interfering with its operation.

You can power up or power down the V_{CCIO}, V_{CC}, V_{CCPGM}, and V_{CCPD} supplies in any sequence. Individual power supply ramp-up and ramp-down rates range from 50 µs to 100 ms. During hot socketing, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

For more information about the hot-socketing specification, refer to the *DC and Switching Characteristics* and the *Hot-Socketing and Power Sequencing Feature and Testing for Altera Devices* white paper.

A possible concern regarding hot socketing is the potential for "latch-up." Nevertheless, Stratix IV devices are immune to latch-up when hot socketing. Latch-up can occur when electrical subsystems are hot socketed into an active system. During hot socketing, the signal pins can be connected and driven by the active system before the power supply can provide current to the device's power and ground planes. This condition can lead to latch-up and cause a low-impedance path from power to ground within the device. As a result, the device draws a large amount of current, possibly causing electrical damage.

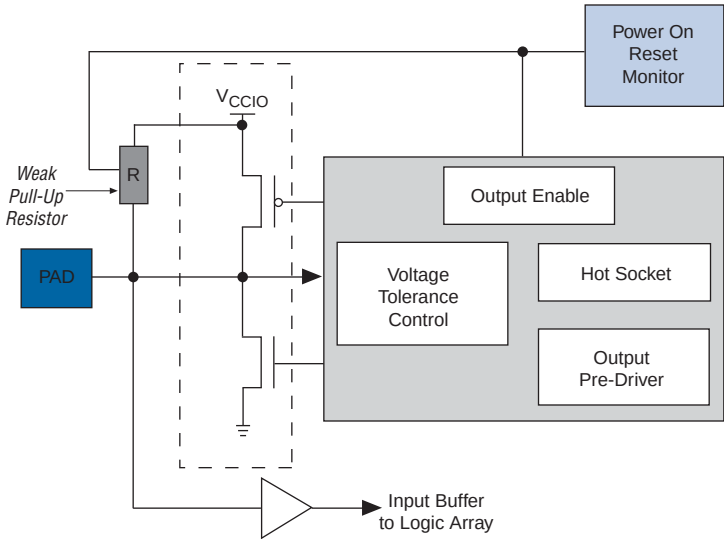
Hot-Socketing Feature Implementation in Stratix IV Devices

The hot-socketing feature turns off the output buffer during power up and power down of the V_{CC}, V_{CCIO}, V_{CCPGM}, or V_{CCPD} power supplies. The hot-socketing circuitry generates an internal `HOTSCKT` signal when the V_{CC}, V_{CCIO}, V_{CCPGM}, or V_{CCPD} power supplies are below the threshold voltage. Hot-socketing circuitry is designed to prevent excess I/O leakage during power up. When the voltage ramps up very slowly, it is still relatively low, even after the POR signal is released and the configuration is completed. The `CONF_DONE`, `nCEO`, and `nSTATUS` pins fail to

respond, as the output buffer cannot flip from the state set by the hot-socketing circuit at this low voltage. Therefore, the hot-socketing circuitry has been removed from these configuration pins to make sure that they are able to operate during configuration. Thus, it is expected behavior for these pins to drive out during power-up and power-down sequences.

Figure 9–1 shows the Stratix IV device's I/O pin circuitry.

Figure 9–1. Hot-Socketing Circuitry for Stratix IV Devices



The POR circuit monitors the voltage level of the power supplies (V_{CC}, V_{CCPT}, V_{CCPD}, and V_{CCPGM}) and keeps the I/O pins tri-stated until the device is in user mode. The weak pull-up resistor (R) in the Stratix IV input/output element (IOE) keeps the I/O pins from floating. The 3.0-V tolerance control circuit permits the I/O pins to be driven by 3.0 V before the V_{CCIO}, V_{CC}, V_{CCPD}, and/or V_{CCPGM} supplies are powered. It also prevents the I/O pins from driving out when the device is not in user mode.

Altera uses GND as a reference for hot-socketing operations and I/O buffer designs. To ensure proper operation, you must connect the GND between boards before connecting the power supplies. This prevents the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled up GND could otherwise cause an out-of-specification I/O voltage or current condition with the Altera device.

Power-On Reset Circuitry

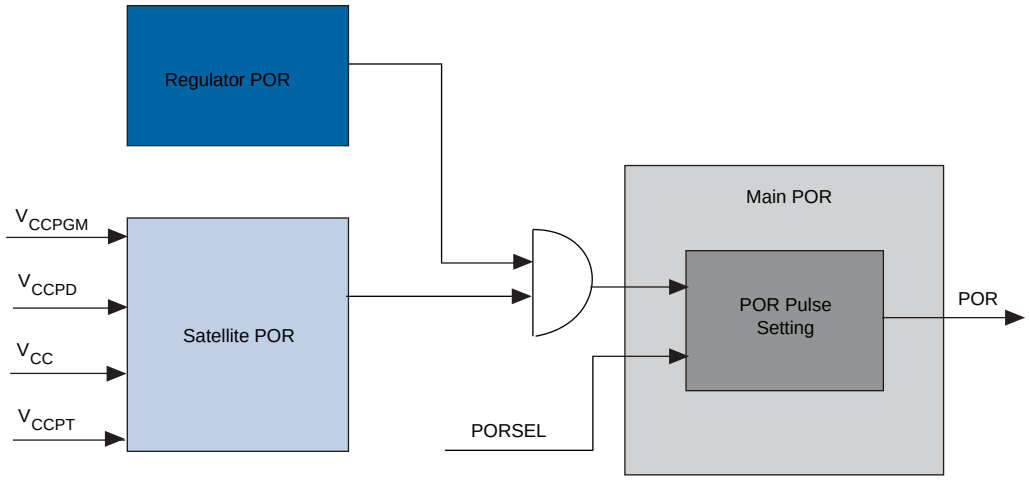
When power is applied to a Stratix IV device, a POR event occurs if the power supply reaches the recommended operating range within the maximum power supply ramp time (t_{RAMP}). If t_{RAMP} is not met, the device I/O pins and programming registers remain tri-stated, during which device configuration could fail. The maximum t_{RAMP} for Stratix IV devices is 100 ms; the minimum t_{RAMP} is 50 µs. Stratix IV devices provide a dedicated input pin (PORSEL) to select a POR delay time during power up. When the PORSEL pin is connected to GND, the POR delay time is 100 to 300 ms. When the PORSEL pin is set to high, the POR delay time is 4 to 12 ms.

The POR block consists of a regulator POR, satellite POR, and main POR to check the power supply levels for proper device configuration. The satellite POR monitors the V_{CCPD} and V_{CCPGM} power supplies that are used in the I/O buffers and for device programming. The satellite POR also monitors the V_{CC} and V_{CCPT} power supplies that are used in the device core. The POR block also checks for functionality of I/O level shifters powered by V_{CCPD} and V_{CCPGM} during power-up mode. The main POR waits for satellite POR and the regulator POR to release the POR signal. Until the release of the POR signal, the device configuration cannot start.

The internal configuration memory supply that is used during device configuration is checked by the regulator POR block and is gated in the main POR block for the final POR trip. Figure 9–2 shows a simplified diagram of the POR block.

All configuration-related dedicated and dual function I/O pins must be powered by V_{CCPGM}.

Figure 9–2. Simplified POR Diagram for Stratix IV Devices



Power-On Reset Specifications

The POR circuit monitors the power supplies listed in Table 9–1.

Table 9–1. Power Supplies Monitored by the POR Circuitry

Power Supply	Description	Setting (V)
V_{CC}	Core and periphery power supply	0.9
V_{CCPT}	Programmable power technology power supply	1.5
V_{CCPD}	I/O pre-driver power supply	2.5, 3.0
V_{CCPGM}	Configuration pins power supply	1.8, 2.5, 3.0

The POR circuit does not monitor the power supplies listed in Table 9–2.

Table 9–2. Power Supplies Not Monitored by the POR Circuitry

Power Supply	Description	Setting (V)
V_{CCIO}	I/O power supply	1.2, 1.5, 1.8, 2.5, 3.0
V_{CCA_PLL}	PLL analog global power supply	2.5
V_{CCD_PLL}	PLL digital power supply	0.9
V_{CC_CLKIN}	PLL differential clock input power supply (top and bottom I/O banks only)	2.5
V_{CCBAT}	Battery back-up power supply for design security volatile key storage	3.0

V_{CCIO}, V_{CCA_PLL}, V_{CCD_PLL}, V_{CC_CLKIN}, and V_{CCBAT} are not monitored by POR and have no affect on the device configuration.

The POR specification is designed to ensure that all the circuits in the Stratix IV device are at certain known states during power up.

The POR signal pulse width is programmable using the PORSEL input pin. When PORSEL is set to low, the POR signal pulse width is set to **100 ms**. When the PORSEL is set to high, the POR signal pulse width is set to **12 ms**.

For more information about the POR specification, refer to the *DC and Switching Characteristics*.

Document Revision History

Table 9–3 shows the revision history for this chapter.

Table 9–3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated graphics. ■ Minor text edits.	—
June 2009 v2.2	■ Updated Table 9–2. ■ Added introductory sentences to improve search ability. ■ Removed the Conclusion section. ■ Minor text edits.	—
March 2009 v2.1	■ Changed all “Stratix IV E” to “Stratix IV”. ■ Updated “Stratix IV Hot-Socketing Specifications” and “Hot-Socketing Feature Implementation in Stratix IV Devices” sections. ■ Updated Figure 9–2. ■ Removed “Referenced Documents” section.	—
November 2008 v2.0	■ Updated “Hot-Socketing Feature Implementation in Stratix IV Devices” on page 9–2. ■ Updated “Power-On Reset Circuitry” on page 9–4. ■ Updated Table 9–1. ■ Made minor editorial changes.	—
July 2008 v1.1	Revised “Introduction”.	—
May 2008 v1.0	Initial release.	—

10. Configuration, Design Security, and Remote System Upgrades in Stratix IV Devices

SIV51010-3.0

This chapter describes the configuration, design security, and remote system upgrades in Stratix® IV devices. Stratix IV devices provide configuration data decompression to save configuration memory space and time. They also provide a built-in design security feature that protects your designs against IP theft and tampering of your configuration files.

Stratix IV devices also offer remote system upgrade capability so that you can upgrade your system in real-time through any network. This helps to deliver feature enhancements and bug fixes and provides error detection, recovery, and status information to ensure reliable reconfiguration.

Overview

This chapter contains information about Stratix IV—supported configuration schemes, instructions about how to execute the required configuration schemes, and the necessary pin settings.

Stratix IV devices use SRAM cells to store configuration data. As SRAM is volatile, you must download configuration data to the Stratix IV device each time the device powers up. You can configure Stratix IV devices using one of four configuration schemes:

- Fast passive parallel (FPP)
- Fast active serial (AS)
- Passive serial (PS)
- Joint Test Action Group (JTAG)

All configuration schemes use either an external controller (for example, a MAX® II device or microprocessor), a configuration device, or a download cable. For more information, refer to "Configuration Features" on page 10–3.

This chapter includes the following sections:

- "Dedicated Remote System Upgrade Circuitry" on page 10–52
- "Quartus II Software Support" on page 10–57
- "Design Security" on page 10–58

Configuration Devices

Altera® serial configuration devices support a single-device and multi-device configuration solution for Stratix IV devices and are used in the fast AS configuration scheme. Serial configuration devices offer a low-cost, low pin-count configuration solution.

For information about serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook*.

All minimum timing information in this chapter covers the entire Stratix IV family. Some devices may work at less than the minimum timing stated in this handbook due to process variation.

Configuration Schemes

Select the configuration scheme by driving the Stratix IV device MSEL pins either high or low, as shown in Table 10–1. The MSEL input buffers are powered by the V_{CC} power supply. Altera recommends you hardwire the MSEL[] pins to V_{CCPGM} and GND. The MSEL[2..0] pins have 5-kΩ internal pull-down resistors that are always active. During power-on reset (POR) and during reconfiguration, the MSEL pins must be at V_{IL} and V_{IH} levels of V_{CCPGM} voltage to be considered logic low and logic high.

To avoid problems with detecting an incorrect configuration scheme, hardwire the MSEL[] pins to V_{CCPGM} and GND without pull-up or pull-down resistors. Do not drive the MSEL[] pins by a microprocessor or another device.

Table 10–1. Stratix IV Configuration Schemes

Configuration Scheme	MSEL2	MSEL1	MSEL0
Fast passive parallel	0	0	0
Passive serial	0	1	0
Fast AS (40 MHz) *(1)*	0	1	1
Remote system upgrade fast AS (40 MHz) *(1)*	0	1	1
FPP with design security feature and/or decompression enabled *(2)*	0	0	1
JTAG-based configuration *(4)*	*(3)*	*(3)*	*(3)*

Notes to Table 10–1:

(1) Stratix IV devices only support fast AS configuration. You must use either EPCS64 or EPCS128 devices to configure a Stratix IV device in fast AS mode.

(2) These modes are only supported when using a MAX II device or a microprocessor with flash memory for configuration. In these modes, the host system must output a DCLK that is ×4 the data rate.

(3) Do not leave the MSEL pins floating, connect them to V_{CCPGM} or GND. These pins support the non-JTAG configuration scheme used in production. If you only use the JTAG configuration, connect the MSEL pins to GND.

(4) The JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored. The JTAG-based configuration does not support the design security or decompression features.

Table 10–2 lists the uncompressed raw binary file (**.rbf**) configuration file sizes for Stratix IV devices.

Table 10–2. Stratix IV Uncompressed Raw Binary File (**.rbf**) Sizes *(Note 1)*

Device	Data Size (Bits)
EP4SE230	94, 600, 000
EP4SE360	128, 400, 000
EP4SE530	171, 800, 000
EP4SE820	241, 700, 000
EP4SGX70	47, 900, 000
EP4SGX110	47, 900, 000
EP4SGX180	94, 600, 000
EP4SGX230	94, 600, 000
EP4SGX290	128, 400, 000
EP4SGX360	128, 400, 000
EP4SGX530	171, 800, 000
EP4S40G2	94, 600, 000
EP4S40G5	171, 800, 000
EP4S100G2	94, 600, 000
EP4S100G3	171, 800, 000
EP4S100G4	171, 800, 000
EP4S100G5	171, 800, 000

Note to Table 10–2:

(1) These values are preliminary.

Use the data in Table 10–2 to estimate the file size before design compilation. Different configuration file formats; for example, a hexidecimal (**.hex**) or tabular text file (**.ttf**) format, have different file sizes. Refer to the Quartus® II software for the different types of configuration file and file sizes. However, for any specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you are using compression, the file size can vary after each compilation because the compression ratio is dependent on the design.

For more information about setting device configuration options or creating configuration files, refer to the *Device Configuration Options* and *Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

Configuration Features

Stratix IV devices offer design security, decompression, and remote system upgrade features. Design security using configuration bitstream encryption is available in Stratix IV devices, which protects your designs. Stratix IV devices can receive a compressed configuration bitstream and decompress this data in real-time, reducing storage requirements and configuration time. You can make real-time system upgrades from remote locations of your Stratix IV designs with the remote system upgrade feature.

Table 10–3 summarizes which configuration features you can use in each configuration scheme.

Table 10–3. Stratix IV Configuration Features

Configuration Scheme	Configuration Method	Decompression	Design Security	Remote System Upgrade
FPP	MAX II device or a microprocessor with flash memory	✓ *(1)*	✓ *(1)*	—
Fast AS	Serial configuration device	✓	✓	✓
PS	MAX II device or a microprocessor with flash memory	✓	✓	—
	Download cable	✓	✓	—
JTAG	MAX II device or a microprocessor with flash memory	—	—	—
	Download cable	—	—	—

Note to Table 10–3:

(1) In these modes, the host system must send a DCLK that is ×4 the data rate.

You can also refer to the following:

- For more information about the configuration data decompression feature, refer to "Configuration Data Decompression" on page 10–44.
- For more information about the remote system upgrade feature, refer to "Remote System Upgrades" on page 10–46.
- For more information about the design security feature, refer to "Design Security" on page 10–58.

If your system already contains a common flash interface (CFI) flash memory, you can use it for Stratix IV device configuration storage as well. The MAX II parallel flash loader (PFL) feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface and provides the logic to control configuration from the flash memory device to the Stratix IV device. Both PS and FPP configuration modes are supported using this PFL feature.

For more information about PFL, refer to *AN 386: Using the MAX II Parallel Flash Loader with the Quartus II Software*.

For more information about programming Altera serial configuration devices, refer to "Programming Serial Configuration Devices" on page 10–21.

Power-On Reset Circuit

The POR circuit keeps the entire system in reset until the power supply voltage levels have stabilized on power-up. After power-up, the device does not release nSTATUS until V_{CCPT}, V_{CC}, V_{CCPD}, and V_{CCPGM} are above the device's POR trip point. On power down, brown-out occurs if the V_{CC} or V_{CCPT} ramps down below the POR trip point and if V_{CC}, V_{CCPD}, or V_{CCPGM} drops below the threshold level of the hot socket circuitry.

In Stratix IV devices, a pin-selectable option (PORSEL) is provided that allows you to select between the standard POR time or fast POR time. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms, which has a lower power-ramp rate. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms.

V_{CCPGM} Pins

Stratix IV devices have a power supply, V_{CCPGM}, for all the dedicated configuration pins and dual function pins. The supported configuration voltage is 1.8, 2.5, and 3.0 V. Stratix IV devices do not support 1.5 V configuration.

Use the V_{CCPGM} pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bidirectional pins, and some of the dual functional pins that you use for configuration. With V_{CCPGM}, the configuration input buffers do not have to share power lines with the regular I/O buffer in Stratix IV devices.

The operating voltage for the configuration input pin is independent of the I/O banks power supply V_{CCIO} during configuration. Therefore, Stratix IV devices do not need configuration voltage constraints on V_{CCIO}.

V_{CCPD} Pins

Stratix IV devices have a dedicated programming power supply, V_{CCPD}, which must be connected to 3.0 V/2.5 V to power the I/O pre-drivers and JTAG I/O pins (TCK, TMS, TDI, TDO, and TRST).

V_{CCPGM} and V_{CCPD} must ramp up from 0 V to the desired voltage level within 100 ms when PORSEL is low or 4 ms when PORSEL is high. If these supplies are not ramped up within this specified time, your Stratix IV device will not configure successfully. If your system cannot ramp up the power supplies within 100 ms or 4 ms, you must hold nCONFIG low until all the power supplies are stable.

V_{CCPD} must be greater than or equal to V_{CCIO} of the same bank. If V_{CCIO} of the bank is set to 3.0 V, V_{CCPD} must be powered up to 3.0 V. If the V_{CCIO} of the bank is powered to 2.5 V or lower, V_{CCPD} must be powered up to 2.5 V.

For more information about configuration pins power supply, refer to "Device Configuration Pins" on page 10–37.

Fast Passive Parallel Configuration

Fast passive parallel configuration in Stratix IV devices is designed to meet the continuously increasing demand for faster configuration times. Stratix IV devices are designed with the capability of receiving byte-wide configuration data per clock cycle.

You can perform FPP configuration of Stratix IV devices using an intelligent host, such as a MAX II device or a microprocessor.

FPP Configuration Using a MAX II Device as an External Host

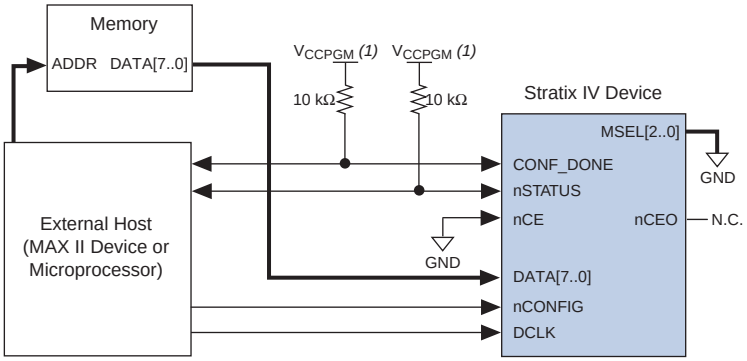
FPP configuration using compression and an external host provides the fastest method to configure Stratix IV devices. In this configuration scheme, you can use a MAX II device as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** format. When using the MAX II device as an intelligent host, a design that controls the configuration process, such as fetching the data from flash memory and sending it to the device, must be stored in the MAX II device.

If you are using the Stratix IV decompression and/or design security features, the external host must be able to send a DCLK frequency that is ×4 the data rate.

The ×4 DCLK signal does not require an additional pin and is sent on the DCLK pin. The maximum DCLK frequency is 125 MHz, which results in a maximum data rate of 250 Mbps. If you are not using the Stratix IV decompression or design security features, the data rate is ×8 the DCLK frequency.

Figure 10–1 shows the configuration interface connections between the Stratix IV device and a MAX II device for single device configuration.

Figure 10–1. Single Device FPP Configuration Using an External Host



Note to Figure 10–1:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix IV device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends that you power up all configuration system I/Os with V_{CCPGM}.

After power-up, the Stratix IV device goes through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power up and configuration, the user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in the reset stage. To initiate configuration, the MAX II device must drive the nCONFIG pin from low to high.

To begin the configuration process, you must fully power V_{CCPT}, V_{CC}, V_{CCPD}, and V_{CCPGM} of the banks where the configuration pins reside to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the MAX II device places the configuration data one byte at a time on the DATA[7..0] pins.

Stratix IV devices receive configuration data on the DATA[7..0] pins and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. If you are using the Stratix IV decompression and/or design security features, configuration data is latched on the rising edge of every fourth DCLK cycle. After the configuration data is latched in, it is processed during the following three DCLK cycles. Therefore, you can only stop DCLK after three clock cycles after the last data is latched into the Stratix IV Devices.

Data is continuously clocked into the target device until CONF_DONE goes high. The CONF_DONE pin goes high one byte early in FPP modes. The last byte is required for AS and PS modes. After the device has received the next-to-last byte of the configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

In Stratix IV devices, the initialization clock source is either the internal oscillator or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix IV device provides itself with enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You can also synchronize initialization of multiple devices or delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock (CLKUSR)** option in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. Supplying a clock on CLKUSR does not affect the configuration process. The CONF_DONE pin goes high one byte early in FPP modes. The last byte is required for AS and PS modes. After the CONF_DONE pin transitions high, CLKUSR is enabled after the time specified at t_{CD2CU}. After this time period elapses, Stratix IV devices require 8,532 clock cycles to initialize properly and enter user mode. Stratix IV devices support a CLKUSR f_{MAX} of 125 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. This **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it is high because of an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. After the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low.

When initialization is complete, the INIT_DONE pin is released and pulled high. The MAX II device must be able to detect this low-to-high transition, which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user-mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed bitstream in FPP.

To ensure DCLK and DATA[7..0] are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[7..0] pins are available as user I/O pins after configuration. When you select the FPP scheme as a default in the Quartus II software, these I/O pins are tri-stated in user mode. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If you need to stop DCLK, it can only be stopped:

- three clock cycles after the last data byte was latched into the Stratix IV device when you use the decompression and/or design security features.
- two clock cycles after the last data byte was latched into the Stratix IV device when you do not use the Stratix IV decompression and/or design security features.

By stopping DCLK, the configuration circuit allows enough clock cycles to process the last byte of latched configuration data. When the clock restarts, the MAX II device must provide data on the DATA[7..0] pins prior to sending the first DCLK rising edge.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box) is turned on, the device releases nSTATUS after a reset time-out period (a maximum of 500 µs). After nSTATUS is released and pulled high by a pull-up resistor, the MAX II device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

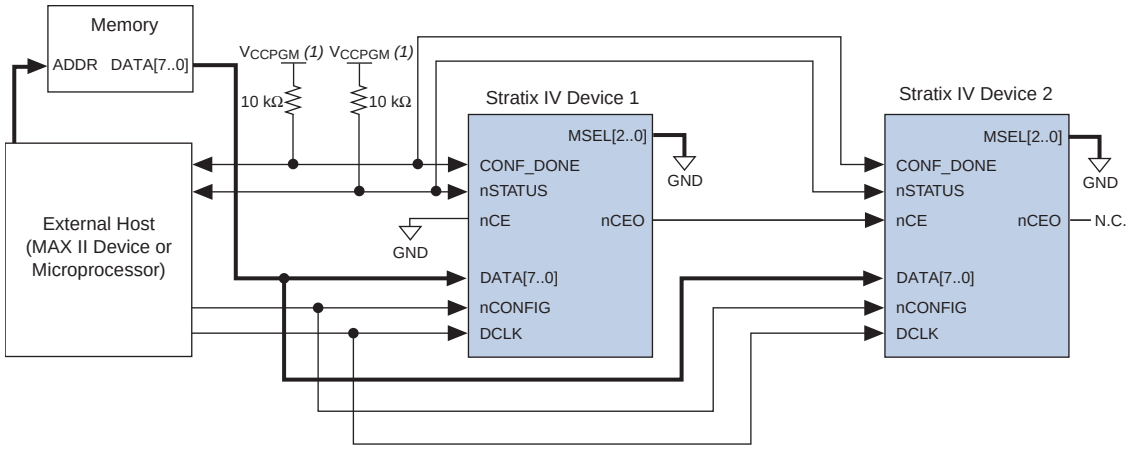
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The MAX II device must monitor the CONF_DONE pin to detect errors and determine when programming completes. If all the configuration data is sent, but the CONF_DONE or INIT_DONE signals have not gone high, the MAX II device reconfigures the target device.

If you use the optional CLKUSR pin and nCONFIG is pulled low to restart the configuration during device initialization, ensure that CLKUSR continues toggling during the time nSTATUS is low (a maximum of 500 µs).

When the device is in user mode, initiating reconfiguration is done by transitioning the nCONFIG pin low-to-high. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. After nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 10–2 shows how to configure multiple Stratix IV devices using a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the devices are cascaded for multi-device configuration.

Figure 10–2. Multi-Device FPP Configuration Using an External Host



Note to Figure 10–2:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O standard on the device and the external host. Altera recommends you power up all configuration system I/Os with V_{CCPGM}.

In a multi-device FPP configuration, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration within one clock cycle; therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. The configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

All nSTATUS and CONF_DONE pins are tied together; if any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

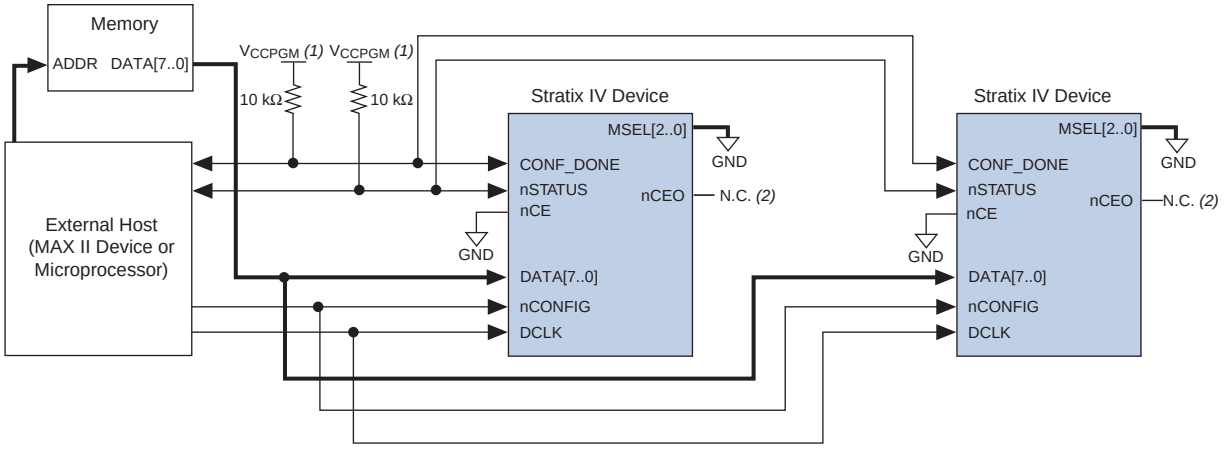
If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (a maximum of 500 µs). After all nSTATUS pins are released and pulled high, the MAX II device tries to reconfigure the chain without pulsing nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

In a multi-device FPP configuration chain, all Stratix IV devices in the chain must either enable or disable the decompression and/or design security features. You cannot selectively enable the decompression and/or design security features for each device in the chain because of the DATA and DCLK relationship. If the chain contains devices that do not support design security, use a serial configuration scheme.

If a system has multiple devices that contain the same configuration data, tie all device nCE inputs to GND and leave the nCEO pins floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices start and complete configuration at the same time.

Figure 10–3 shows a multi-device FPP configuration when both Stratix IV devices are receiving the same configuration data.

Figure 10–3. Multiple-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 10–3:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends you power up all configuration system I/Os with V_{CCPGM}.

(2) The nCEO pins of both Stratix IV devices are left unconnected when configuring the same configuration data into multiple devices.

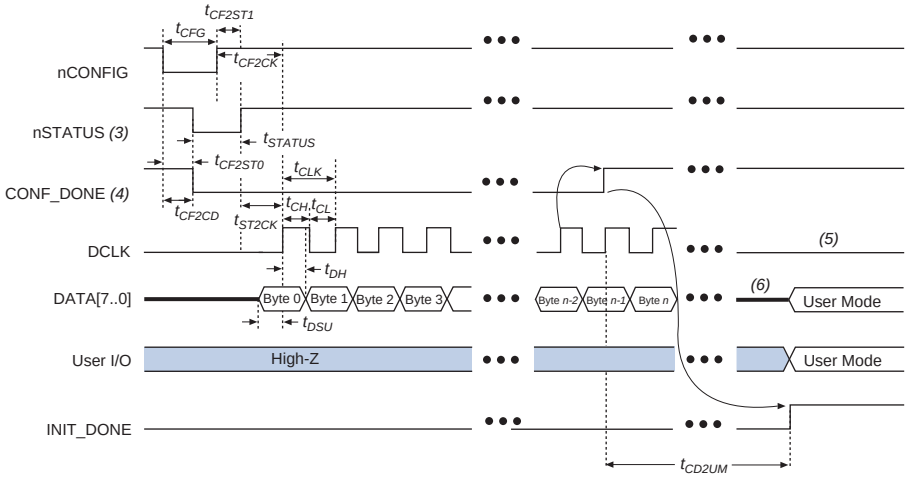
You can use a single configuration chain to configure Stratix IV devices with other Altera devices that support FPP configuration, such as other types of Stratix devices. To ensure that all devices in the chain complete configuration at the same time, or that an error flagged by one device initiates reconfiguration in all devices, tie all of the device CONF_DONE and nSTATUS pins together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* in volume 2 of the *Configuration Handbook*.

FPP Configuration Timing

Figure 10–4 shows the timing waveform for FPP configuration when using a MAX II device as an external host. This waveform shows the timing when you have not enabled the decompression and design security features.

Figure 10–4. FPP Configuration Timing Waveform *(Note 1)*, *(2)*



Notes to Figure 10–4:

(1) Use this timing waveform when you have not enabled the decompression and design security features.

(2) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) After power-up, the Stratix IV device holds nSTATUS low for the time of the POR delay.

(4) After power-up, before and during configuration, CONF_DONE is low.

(5) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(6) DATA[7..0] are available as user I/O pins after configuration except for some exceptions on Stratix IV GT. The state of these pins depends on the dual-purpose pin settings.

Table 10–4 lists the timing parameters for Stratix IV devices for FPP configuration when you have not enabled the decompression and design security features.

Table 10–4. FPP Timing Parameters for Stratix IV Devices (Part 1 of 2) *(Note 1)*, *(2)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	800	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	10	500 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	500 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	500	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	4	—	ns

Table 10–4. FPP Timing Parameters for Stratix IV Devices (Part 2 of 2) *(Note 1), (2)*

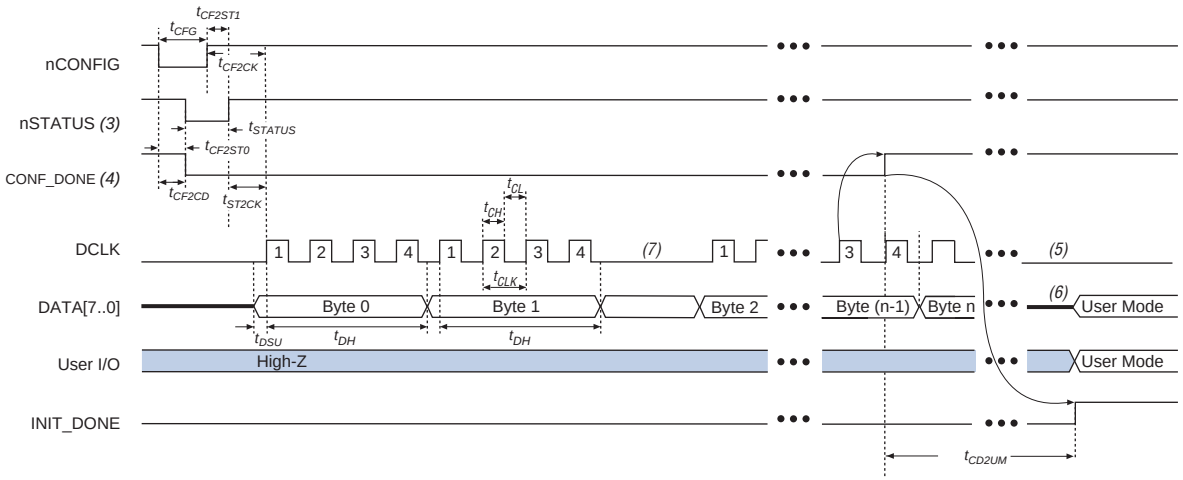
Symbol	Parameter	Minimum	Maximum	Units
t_{DH}	Data hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time *(5)*	3.2	—	ns
t_{CL}	DCLK low time *(5)*	3.2	—	ns
t_{CLK}	DCLK period *(5)*	8	—	ns
f_{MAX}	DCLK frequency	—	125	MHz
t_R	Input rise time	—	40	ns
t	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	55	150	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (8532 × CLKUSR period)	—	—

Notes to Table 10–4:

(1) This information is preliminary.
(2) Use these timing parameters when you have not enabled the decompression and design security features.
(3) You can obtain this value if you do not delay the configuration by extending the nCONFIG or nSTATUS low pulse width.
(4) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for starting the device.
(5) Adding up t_{CH} and t_{CL} equals to t_{CLK}. When t_{CH} is 3.2 ns (min), t_{CL} must be 4.8 ns and vice versa.

Figure 10–5 shows the timing waveform for FPP configuration when using a MAX II device as an external host. This waveform shows the timing when you have enabled the decompression and/or design security features.

Figure 10–5. FPP Configuration Timing Waveform with Decompression or Design Security Feature Enabled *(Note 1)*, *(2)*



Notes to Figure 10–5:

(1) Use this timing waveform when you have enabled the decompression and/or design security features.

(2) The beginning of this waveform shows the device in user-mode. In user-mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) After power-up, the Stratix IV device holds nSTATUS low for the time of the POR delay.

(4) After power-up, before and during configuration, CONF_DONE is low.

(5) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(6) DATA[7..0] are available as user I/O pins after configuration except for some exceptions on Stratix IV GT. The state of these pins depends on the dual-purpose pin settings.

(7) If needed, you can pause DCLK by holding it low. When DCLK restarts, the external host must provide data on the DATA[7..0] pins prior to sending the first DCLK rising edge.

Table 10–5 lists the timing parameters for Stratix IV devices for FPP configuration when you enable the decompression and/or the design security features.

Table 10–5. FPP Timing Parameters for Stratix IV Devices with the Decompression and/or Design Security Features Enabled *(Note 1)*, *(2)* (Part 1 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	800	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	10	500 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	500 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	500	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	4	—	ns
t_{DH}	Data hold time after rising edge on DCLK	24	—	ns
t_{CH}	DCLK high time *(5)*	3.2	—	ns

Table 10–5. FPP Timing Parameters for Stratix IV Devices with the Decompression and/or Design Security Features Enabled *(Note 1)*, *(2)* (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{CL}	DCLK low time *(5)*	3.2	—	ns
t_{CLK}	DCLK period *(5)*	8	—	ns
f_{MAX}	DCLK frequency	—	125	MHz
t_{DATA}	Data rate	—	250	Mbps
t_R	Input rise time	—	40	ns
t	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	55	150	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on *(4)*	t_{CD2CU} + (8532 × CLKUSR period)	—	—

Notes to Table 10–5:

(1) This information is preliminary.
(2) Use these timing parameters when you use the decompression and/or design security features.
(3) You can obtain this value if you do not delay the configuration by extending the nCONFIG or nSTATUS low pulse width.
(4) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for starting the device.
(5) Adding up t_{CH} and t_{CL} equals to t_{CLK}. When t_{CH} is 3.2 ns (min), t_{CL} must be 4.8 ns and vice versa.

For more information about device configuration options and how to create configuration files, refer to the *Device Configuration Options* and *Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

FPP Configuration Using a Microprocessor

In this configuration scheme, a microprocessor can control the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device.

All information in "FPP Configuration Using a MAX II Device as an External Host" on page 10–6 is also applicable when using a microprocessor as an external host. Refer to this section for all configuration and timing information.

Fast Active Serial Configuration (Serial Configuration Devices)

In the fast AS configuration scheme, Stratix IV devices are configured using a serial configuration device. These configuration devices are low-cost devices with non-volatile memory that feature a simple four-pin interface and a small form factor.

The largest serial configuration device currently supports 128 MBits of configuration bitstream. Use the Stratix IV decompression features or select an FPP or PS configuration scheme for EP4SE360, EP4SGX290, EP4S40G5, EP4S100G3 and larger devices.

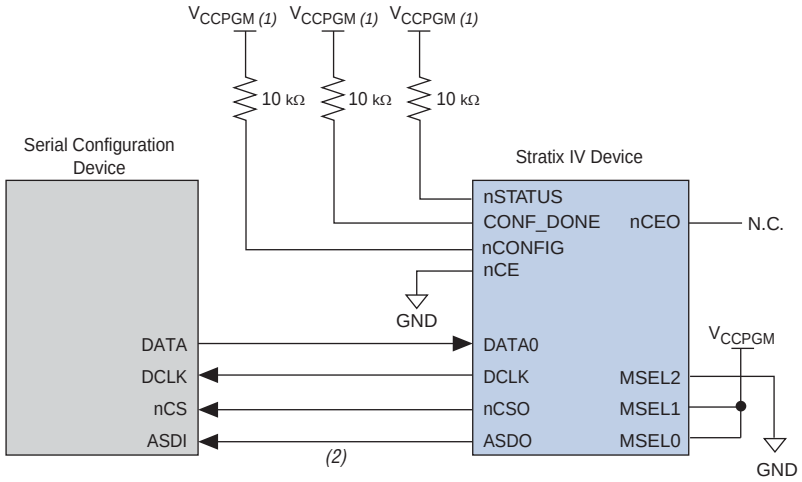
For more information about serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

Serial configuration devices provide a serial interface to access configuration data. During device configuration, Stratix IV devices read configuration data using the serial interface, decompress data if necessary, and configure their SRAM cells. This scheme is referred to as the AS configuration scheme because the Stratix IV device controls the configuration interface. This scheme contrasts with the PS configuration scheme where the configuration device controls the interface.

The Stratix IV decompression and design security features are fully available when configuring your Stratix IV device using fast AS mode.

Serial configuration devices have a four-pin interface: serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and an active-low chip select (nCS). This four-pin interface connects to Stratix IV device pins, as shown in Figure 10–6.

Figure 10–6. Single Device Fast AS Configuration



Notes to Figure 10–6:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.

(2) Stratix IV devices use the ASDO-to-ASDI path to control the configuration device.

You can power the EPCS serial configuration device with 3.0 V when you configure the Stratix IV FPGA using Active Serial (AS) configuration mode. This is feasible because the power supply to the EPCS device ranges between 2.7 V and 3.6 V. You do not need a dedicated 3.3 V power supply to power the EPCS device. The EPCS device and the VCCPGM pins on the Stratix IV device may share the same 3.0 V power supply.

Upon power-up, the Stratix IV devices go through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS and CONF_DONE low, and tri-states all user I/O pins. After the device successfully exits POR, all the user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors, which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. After POR, the Stratix IV device releases nSTATUS, which is pulled high by an external 10-kΩ pull-up resistor and enters configuration mode.

To begin configuration, power the V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} voltages (for the banks where the configuration pins reside) to the appropriate voltage levels.

The serial clock (DCLK) generated by the Stratix IV device controls the entire configuration cycle and provides timing for the serial interface. Stratix IV devices use an internal oscillator to generate DCLK. Using the MSEL[] pins, you can select to use a 40 MHz oscillator.

In fast AS configuration schemes, Stratix IV devices drive out control signals on the falling edge of DCLK. The serial configuration device responds to the instructions by driving out configuration data on the falling edge of DCLK. Then the data is latched into the Stratix IV device on the following falling edge of DCLK.

In configuration mode, Stratix IV devices enable the serial configuration device by driving the nCSO output pin low, which connects to the chip select (nCS) pin of the configuration device. The Stratix IV device uses the serial clock (DCLK) and serial data output (ASDO) pins to send operation commands and/or read address signals to the serial configuration device. The configuration device provides data on its serial data output (DATA) pin, which connects to the DATA0 input of the Stratix IV devices.

After all the configuration bits are received by the Stratix IV device, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ resistor. Initialization begins only after the CONF_DONE signal reaches a logic high level. All AS configuration pins (DATA0, DCLK, nCSO, and ASDO) have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and are driven high by the weak internal pull-up resistors. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

In Stratix IV devices, the initialization clock source is either the internal oscillator or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix IV device provides itself with enough clock cycles for proper initialization. You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. When you select the **Enable user supplied start-up clock** option, the CLKUSR pin is the initialization clock source. Supplying a clock on CLKUSR does not affect the configuration process. After all configuration data is accepted and CONF_DONE goes high, CLKUSR is enabled after four clock cycles of DCLK. After this time period elapses, Stratix IV devices require 8,532 clock cycles to initialize properly and enter user mode. Stratix IV devices support a CLKUSR f_{MAX} of 125 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it is high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. After the option bit to enable INIT_DONE is programmed into the

device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high. This low-to-high transition signals that the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

If an error occurs during configuration, Stratix IV devices assert the nSTATUS signal low, indicating a data frame error, and the CONF_DONE signal stays low. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Stratix IV device resets the configuration device by pulsing nCSO, releases nSTATUS after a reset time-out period (a maximum of 500 µs), and retries configuration. If this option is turned off, the system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration.

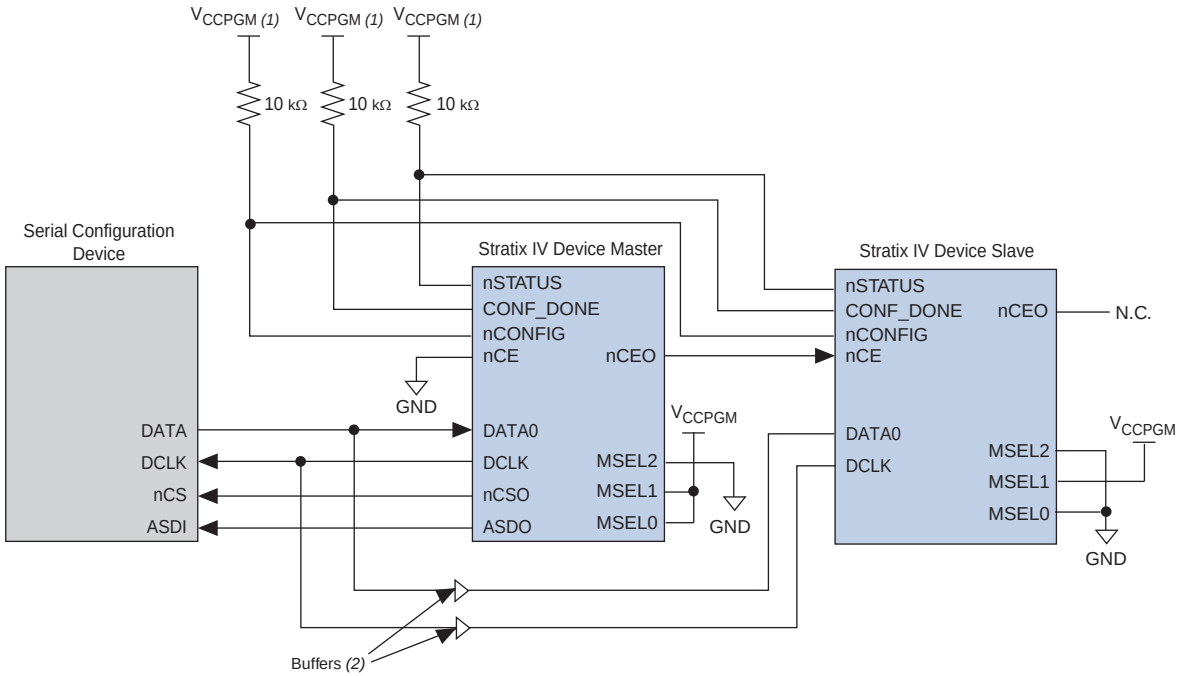
When the Stratix IV device is in user mode, you can initiate reconfiguration by pulling the nCONFIG pin low. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. After nCONFIG returns to a logic high level and nSTATUS is released by the Stratix IV device, reconfiguration begins.

You can configure multiple Stratix IV devices using a single serial configuration device. You can cascade multiple Stratix IV devices using the chip-enable (nCE) and chip-enable-out (nCEO) pins. The first device in the chain must have its nCE pin connected to GND. You must connect its nCEO pin to the nCE pin of the next device in the chain. When the first device captures all of its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. You must leave the nCEO pin of the last device unconnected. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA0 pins of each device in the chain are connected (refer to Figure 10–7).

The first Stratix IV device in the chain is the configuration master and controls configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Stratix IV devices are configuration slaves. You must connect their MSEL pins to select the PS configuration scheme. Any other Altera device that supports PS configuration can also be part of the chain as a configuration slave.

Figure 10–7 shows the pin connections for the multi-device fast AS configuration.

Figure 10–7. Multi-Device Fast AS Configuration



Notes to Figure 10–7:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.

(2) Connect the repeater buffers between the Stratix IV master and slave device(s) for DATA[0] and DCLK. This is to prevent potential signal integrity and clock skew problems.

As shown in Figure 10–7, the nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors. These pins are open-drain bidirectional pins on the devices. When the first device asserts nCEO (after receiving all of its configuration data), it releases its CONF_DONE pin. But the subsequent devices in the chain keep this shared CONF_DONE line low until they have received their configuration data. When all target devices in the chain have received their configuration data and have released CONF_DONE, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode.

If an error occurs at any point during configuration, the nSTATUS line is driven low by the failing device. If you enable the **Auto-restart configuration after error** option, reconfiguration of the entire chain begins after a reset time-out period (a maximum of 500 µs). If you did not enable the **Auto-restart configuration after error** option, the external system must monitor nSTATUS for errors and then pulse nCONFIG low to restart configuration. The external system can pulse nCONFIG if it is under system control rather than tied to V_{CCGPM}.

While you can cascade Stratix IV devices, you cannot cascade or chain together serial configuration devices.

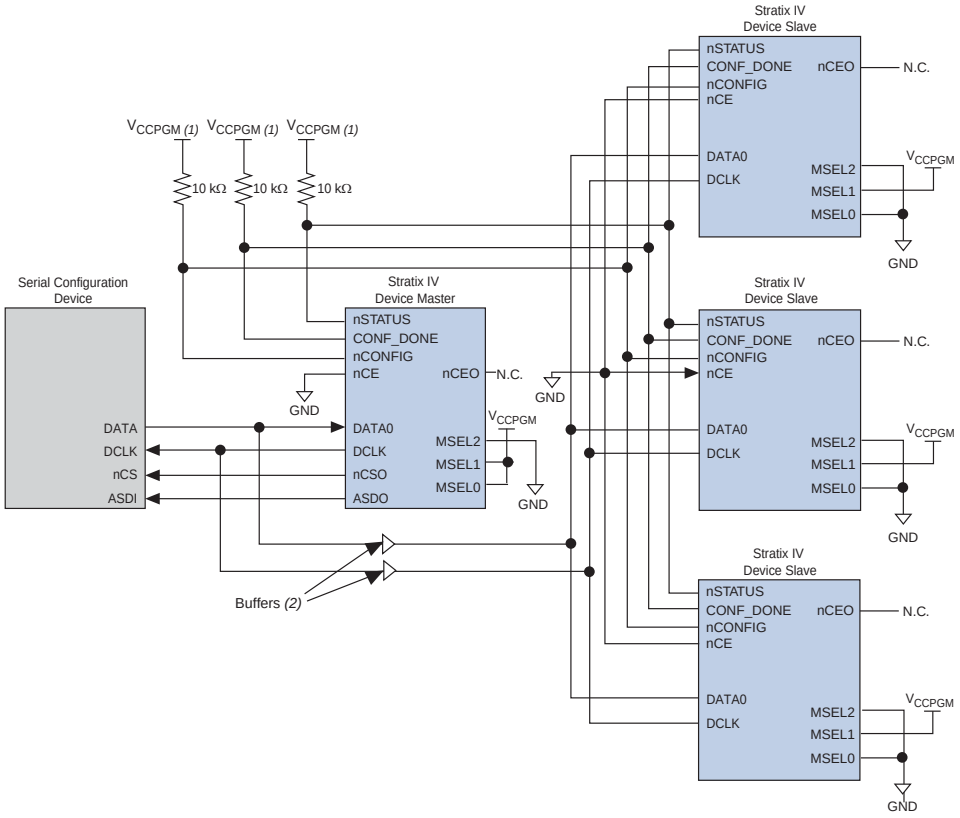
If the configuration bitstream size exceeds the capacity of a serial configuration device, you must select a larger configuration device and/or enable the compression feature. When configuring multiple devices, the size of the bitstream is the sum of the individual device configuration bitstreams.

A system may have multiple devices that contain the same configuration data. In active serial chains, you can implement this by storing one copy of the **.sof** in the serial configuration device. The same copy of the **.sof** configures the master Stratix IV device and all remaining slave devices concurrently. All Stratix IV devices must be the same density and package.

To configure four identical Stratix IV devices with the same **.sof**, you can set up the chain as shown in Figure 10–8. The first device is the master device and its MSEL pins must be set to select AS configuration. The other three slave devices are set up for concurrent configuration and their MSEL pins must be set to select PS configuration. The nCE input pins from the master and slave are connected to GND, and the DATA and DCLK pins connect in parallel to all four devices. During the configuration cycle, the master device reads its configuration data from the serial configuration device and transmits the configuration data to all three slave devices, configuring all of them simultaneously.

Figure 10–8 shows the multi-device fast AS configuration when the devices receive the same data using a single **.sof**.

Figure 10–8. Multi-Device Fast AS Configuration When the Devices Receive the Same Data Using a Single **.sof**



Notes to Figure 10–8:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.

(2) Connect the repeater buffers between the Stratix IV master and slave device(s) for `DATA[0]` and `DCLK`. This is to prevent potential signal integrity and clock skew problems.

Estimating Active Serial Configuration Time

Active serial configuration time is dominated by the time it takes to transfer data from the serial configuration device to the Stratix IV device. This serial interface is clocked by the Stratix IV `DCLK` output (generated from an internal oscillator) and must be set to **40 MHz (25 ns)**.Therefore, the minimum configuration time estimate for an EP4SE230 device (94, 600, 000 bits of uncompressed data) is:

RBF Size × (minimum `DCLK` period / 1 bit per `DCLK` cycle) = estimated minimum configuration time

94, 600, 000 bits × (25 ns / 1 bit) = 2365 ms

The calculation above is based on a preliminary uncompressed **.rbf** size. The final **.rbf** size will be available after the Quartus II software is able to generate the **.rbf**.

Enabling compression reduces the amount of configuration data that is transmitted to the Stratix IV device, which also reduces configuration time. On average, compression reduces configuration time, depending on the design.

Programming Serial Configuration Devices

Serial configuration devices are non-volatile, flash-memory-based devices. You can program these devices in-system using the USB-Blaster™, EthernetBlaster™, or ByteBlaster™ II download cable. Alternatively, you can program them using the Altera programming unit (APU), supported third-party programmers, or a microprocessor with the SRunner software driver.

You can perform in-system programming of serial configuration devices using the conventional AS programming interface or the JTAG interface solution.

Because serial configuration devices do not support the JTAG interface, the conventional method to program them is using the AS programming interface. The configuration data used to program serial configuration devices is downloaded using programming hardware.

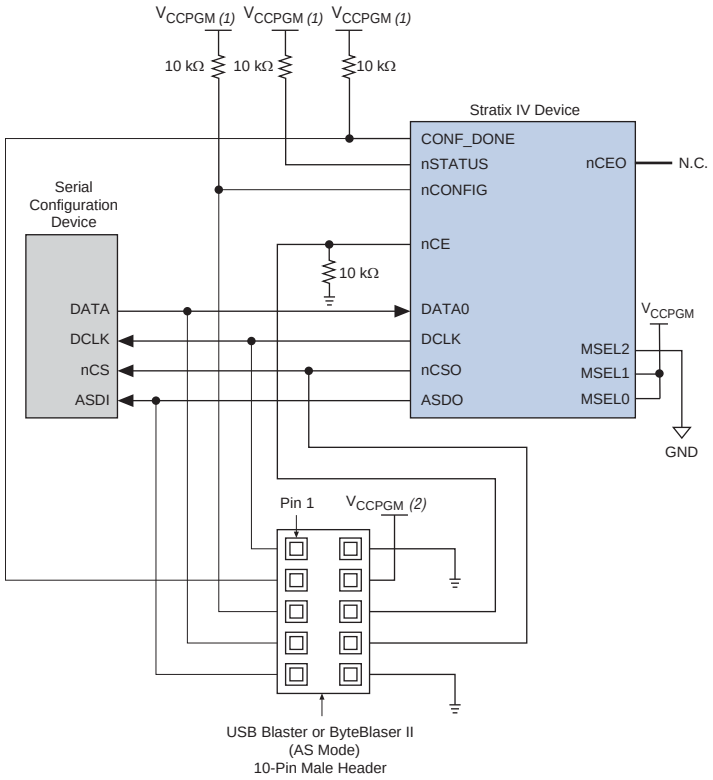
During in-system programming, the download cable disables device access to the AS interface by driving the `nCE` pin high. Stratix IV devices are also held in reset by a low level on `nCONFIG`. After programming is complete, the download cable releases `nCE` and `nCONFIG`, allowing the pull-down and pull-up resistors to drive GND and V_{CCPGM}, respectively. Figure 10–9 shows the download cable connections for the serial configuration device.

Altera has developed Serial FlashLoader (SFL), an in-system programming solution for serial configuration devices using the JTAG interface. This solution requires the Stratix IV device to be a bridge between the JTAG interface and the serial configuration device.

For more information about SFL, refer to *AN 370: Using the Serial FlashLoader with Quartus II Software*.

For more information about the USB-Blaster download cable, refer to the *USB-Blaster Download Cable User Guide*. For more information about the ByteBlaster II cable, refer to the *ByteBlaster II Download Cable User Guide*. For more information about the EthernetBlaster download cable, refer to the *EthernetBlaster Communications Cable User Guide*.

Figure 10–9. In-System Programming of Serial Configuration Devices



Notes to Figure 10–9:

(1) Connect these pull-up resistors to V_{CCPGM} at a 3.0-V supply.

(2) Power up the USB-ByteBlaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

You can program serial configuration devices with the Quartus II software using the Altera programming hardware and the appropriate configuration device programming adapter.

In production environments, you can program serial configuration devices using multiple methods. You can use Altera programming hardware or other third-party programming hardware to program blank serial configuration devices before they are mounted on PCBs. Alternatively, you can use an on-board microprocessor to program the serial configuration device in-system using C-based software drivers provided by Altera.

You can program a serial configuration device in-system by an external microprocessor using SRunner. SRunner is a software driver developed for embedded serial configuration device programming, which can be easily customized to fit in different embedded systems. SRunner is able to read raw programming data (.**rpd**) and write to serial configuration devices. The serial configuration device programming time using SRunner is comparable to the programming time with the Quartus II software.

For more information about SRunner, refer to *AN 418: SRunner: An Embedded Solution for EPCS Programming* and the source code on the Altera website at www.altera.com.

For more information about programming serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

Passive Serial Configuration

You can program PS configuration for Stratix IV devices using an intelligent host, such as a MAX II device or microprocessor with flash memory, or a download cable. In the PS scheme, an external host (a MAX II device, embedded processor, or host PC) controls configuration. Configuration data is clocked into the target Stratix IV device using the DATA0 pin at each rising edge of DCLK.

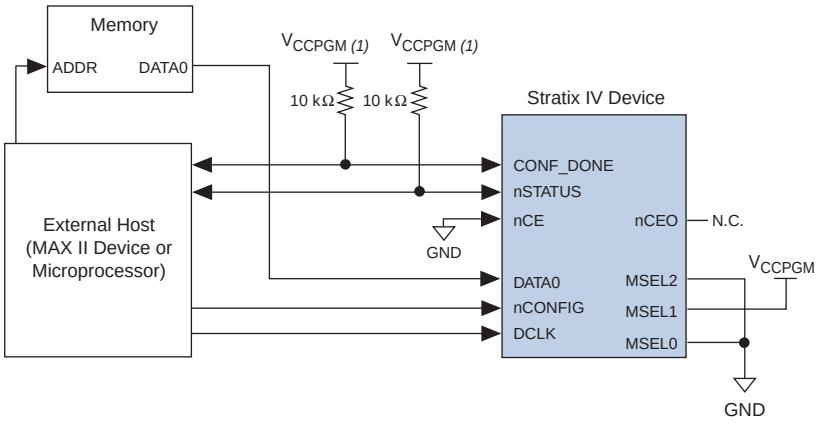
The Stratix IV decompression and design security features are fully available when configuring your Stratix IV device using PS mode.

PS Configuration Using a MAX II Device as an External Host

In this configuration scheme, you can use a MAX II device as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** format.

Figure 10–10 shows the configuration interface connections between a Stratix IV device and a MAX II device for single device configuration.

Figure 10–10. Single Device PS Configuration Using an External Host



Note to Figure 10–10:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix IV device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends you power up all configuration system I/Os with V_{CCPGM}.

After power-up, Stratix IV devices go through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors that are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration, the MAX II device must generate a low-to-high transition on the nCONFIG pin.

V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} of the banks where the configuration pins reside must be fully powered to the appropriate voltage levels to begin the configuration process.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the MAX II device places the configuration data one bit at a time on the DATA0 pin. If you are using configuration data in **.rbf**, **.hex**, or **.ttf** format, you must send the LSB of each data byte first. For example, if the **.rbf** contains the byte sequence 02 1B EE 01 FA, the serial bitstream you must transmit to the device is 0100-0000 1101-1000 0111-0111 1000-0000 0101-1111.

The Stratix IV device receives configuration data on the DATA0 pin and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. Data is continuously clocked into the target device until CONF_DONE goes high. After the device has received all configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

In Stratix IV devices, the initialization clock source is either the internal oscillator or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix IV device provides itself with enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you supply a clock on CLKUSR, it will not affect the configuration process. After all configuration data has been accepted and CONF_DONE goes high, CLKUSR is enabled after the time specified at t_{CD2CU}. After this time period elapses, Stratix IV devices require 8,532 clock cycles to initialize properly and enter user mode. Stratix IV devices support a CLKUSR f_{MAX} of 125 MHz.

An optional INIT_DONE pin is available that signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it is high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. After the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high. The MAX II device must be able to detect this low-to-high transition that signals the device has entered user mode. When initialization is complete, the device enters user mode. In user-mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed bitstream in PS.

To ensure DCLK and DATA0 are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When you chose the PS scheme as a default in the Quartus II software, this I/O pin is tri-stated in user mode and must be driven by the MAX II device. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause the configuration by halting DCLK for an indefinite amount of time.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Stratix IV device releases nSTATUS after a reset time-out period (a maximum of 500 µs). After nSTATUS is released and pulled high by a pull-up resistor, the MAX II device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

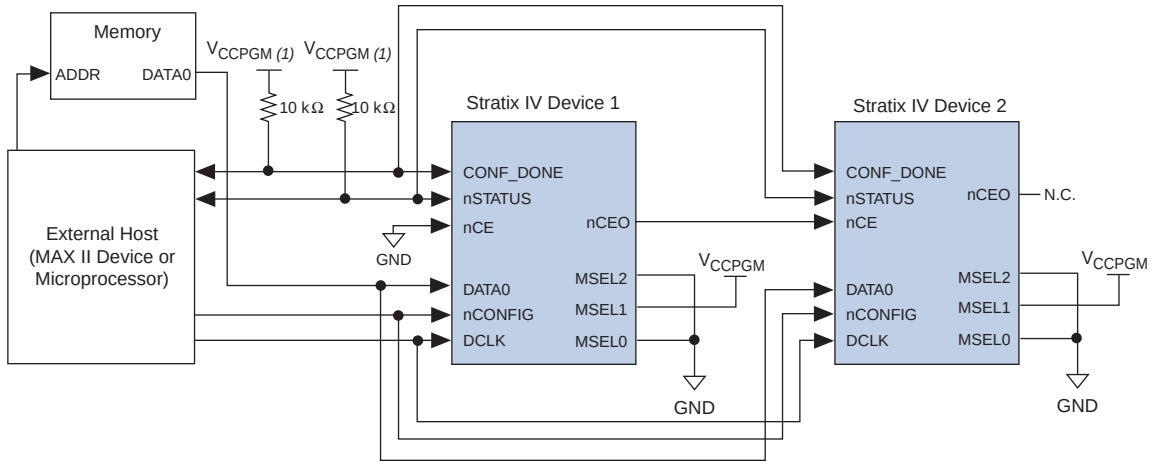
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The CONF_DONE pin must be monitored by the MAX II device to detect errors and determine when programming completes. If all configuration data is sent, but CONF_DONE or INIT_DONE have not gone high, the MAX II device must reconfigure the target device.

If you use the optional CLKUSR pin and nCONFIG is pulled low to restart configuration during device initialization, you must ensure that CLKUSR continues toggling during the time nSTATUS is low (a maximum of 500 µs).

When the device is in user-mode, you can initiate a reconfiguration by transitioning the nCONFIG pin low-to-high. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Once nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 10–11 shows how to configure multiple devices using a MAX II device. This circuit is similar to the PS configuration circuit for a single device, except the Stratix IV devices are cascaded for multi-device configuration.

Figure 10–11. Multi-Device PS Configuration Using an External Host



Note to Figure 10–11:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends you power up all configuration system I/Os with V_{CCPGM}.

In multi-device PS configuration, the first device's nCE pin is connected to GND, while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration within one clock cycle. Therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

Because all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

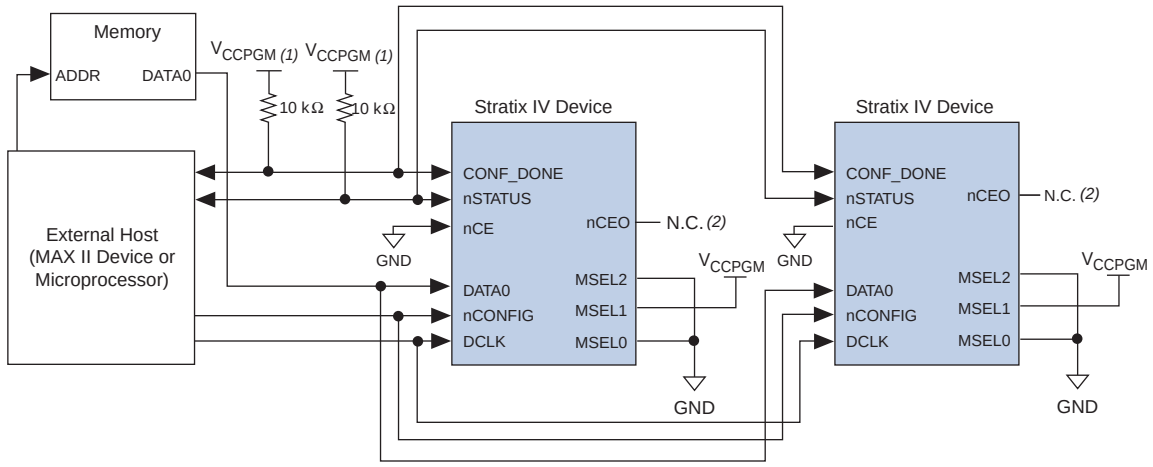
If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (a maximum of 500 µs). After all nSTATUS pins are released and pulled high, the MAX II device can try to reconfigure the chain without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

In your system, you can have multiple devices that contain the same configuration data. To support this configuration scheme, all device nCE inputs are tied to GND, while the nCEO pins are left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain.

Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices start and complete configuration at the same time.

Figure 10–12 shows multi-device PS configuration when both Stratix IV devices are receiving the same configuration data.

Figure 10–12. Multiple-Device PS Configuration When Both Devices Receive the Same Data



Notes to Figure 10–12:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends you power up all configuration system I/Os with V_{CCPGM}.

(2) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

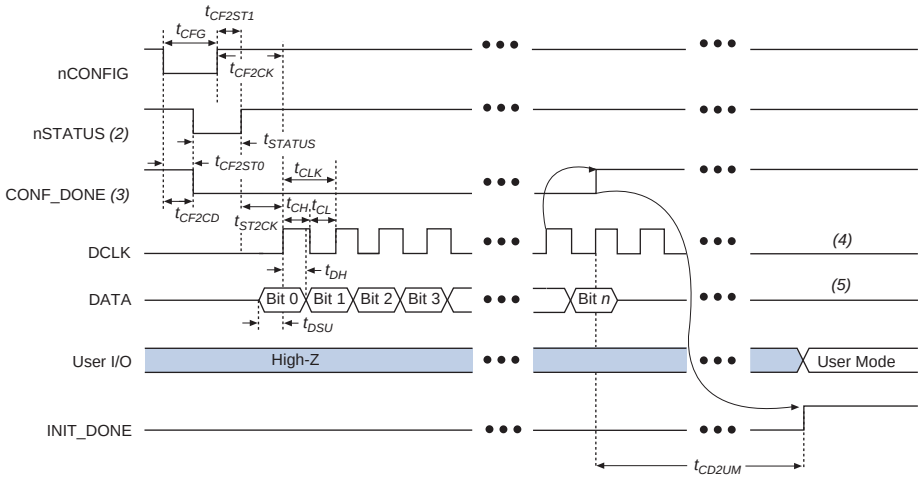
You can use a single configuration chain to configure Stratix IV devices with other Altera devices. To ensure that all devices in the chain complete configuration at the same time, or that an error flagged by one device initiates reconfiguration in all devices, all of the device CONF_DONE and nSTATUS pins must be tied together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

PS Configuration Timing

Figure 10–13 shows the timing waveform for PS configuration when using a MAX II device as an external host.

Figure 10–13. PS Configuration Timing Waveform *(Note 1)*



Notes to Figure 10–13:

(1) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.
(2) After power-up, the Stratix IV device holds nSTATUS low for the time of the POR delay.
(3) After power-up, before and during configuration, CONF_DONE is low.
(4) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.
(5) DATA[0] is available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.

Table 10–6 lists the timing parameters for Stratix IV devices for PS configuration.

Table 10–6. PS Timing Parameters for Stratix IV Devices (Part 1 of 2) *(Note 1)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	800	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	10	500 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	500 *(2)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	500	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	4	—	ns
t_{DH}	Data hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time *(4)*	3.2	—	ns
t_{CL}	DCLK low time *(4)*	3.2	—	ns
t_{CLK}	DCLK period *(4)*	8	—	ns
f_{MAX}	DCLK frequency	—	125	MHz
t_R	Input rise time	—	40	ns

Table 10–6. PS Timing Parameters for Stratix IV Devices (Part 2 of 2) *(Note 1)*

Symbol	Parameter	Minimum	Maximum	Units
t_F	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(3)*	55	150	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (8532 CLKUSR period)	—	—

Notes to Table 10–6:

(1) This information is preliminary.
(2) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.
(3) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.
(4) Adding up t_{CH} and t_{CL} equals to t_{CLK}. When t_{CH} is 3.2 ns (min), t_{CL} must be 4.8 ns and vice versa.

Device configuration options and how to create configuration files are described in the *Device Configuration Options* and *Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

PS Configuration Using a Microprocessor

In this PS configuration scheme, a microprocessor controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device.

For more information about configuration and timing information, refer to "PS Configuration Using a MAX II Device as an External Host" on page 10–23. This section is also applicable when using a microprocessor as an external host.

PS Configuration Using a Download Cable

In this section, the generic term "download cable" includes the Altera USB-Blaster universal serial bus (USB) port download cable, MasterBlaster serial/USB communications cable, ByteBlaster II parallel port download cable, ByteBlasterMV parallel port download cable, and EthernetBlaster download cable.

In a PS configuration with a download cable, an intelligent host (such as a PC) transfers data from a storage device to the device using the USB Blaster, MasterBlaster, ByteBlaster II, EthernetBlaster, or ByteBlasterMV cable.

After power-up, Stratix IV devices go through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors, which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration in this scheme, the download cable generates a low-to-high transition on the nCONFIG pin.

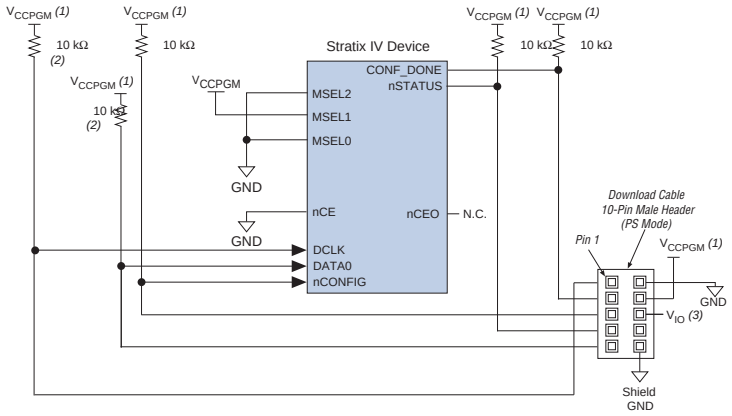
To begin configuration, power the V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} voltages (for the banks where the configuration pins reside) to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. The programming hardware or download cable then places the configuration data one bit at a time on the device's DATA0 pin. The configuration data is clocked into the target device until CONF_DONE goes high. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

When using a download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs. Additionally, the **Enable user-supplied start-up clock** (**CLKUSR**) option has no affect on the device initialization because this option is disabled in the **.sof** when programming the device using the Quartus II programmer and download cable. Therefore, if you turn on the CLKUSR option, you do not need to provide a clock on CLKUSR when you are configuring the device with the Quartus II programmer and a download cable.

Figure 10–14 shows PS configuration for Stratix IV devices using a USB Blaster, EthernetBlaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV cable.

Figure 10–14. PS Configuration Using a USB Blaster, EthernetBlaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV Cable



Notes to Figure 10–14:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCPGM}) as the USB Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This ensures that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, you do not need the pull-up resistors on DATA0 and DCLK.

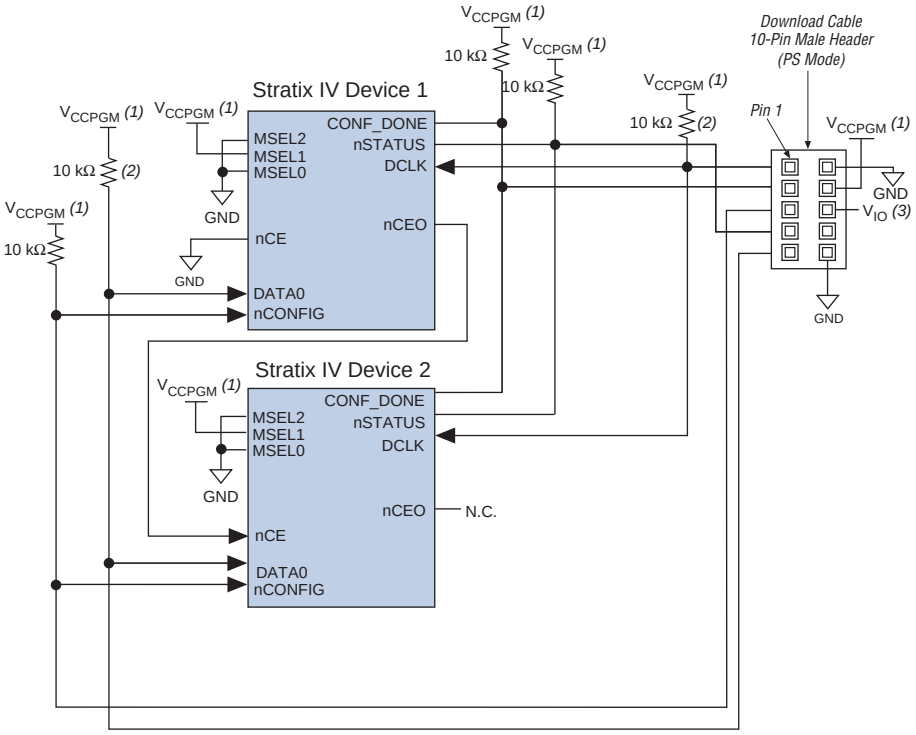
(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPGM}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cable, this pin is a no connect.

You can use a download cable to configure multiple Stratix IV devices by connecting each device's nCEO pin to the subsequent device's nCE pin. The first device's nCE pin is connected to GND, while its nCEO pin is connected to the nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Because all CONF_DONE pins are tied together, all devices in the chain initialize and enter user mode at the same time.

In addition, because the nSTATUS pins are tied together, the entire chain halts configuration if any device detects an error. The **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart the configuration in the Quartus II software when an error occurs.

Figure 10–15 shows how to configure multiple Stratix IV devices with a download cable.

Figure 10–15. Multi-Device PS Configuration Using a USB Blaster, EthernetBlaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV Cable



Notes to Figure 10–15:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCPGM}) as the USB Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This is to ensure that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, you do not need the pull-up resistors on DATA0 and DCLK.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPGM}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cables, this pin is a no connect.

For more information about how to use the USB Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV cables, refer to the following user guides:

- *USB-Blaster Download Cable User Guide*
- *MasterBlaster Serial/USB Communications Cable User Guide*
- *ByteBlaster II Download Cable User Guide*
- *ByteBlasterMV Download Cable User Guide*
- *EthernetBlaster Communications Cable User Guide*

JTAG Configuration

JTAG has developed a specification for boundary-scan testing. This boundary-scan test (BST) architecture offers the capability to efficiently test components on PCBs with tight lead spacing. The BST architecture can test pin connections without using physical test probes and capture functional data while a device is operating normally. You can also use JTAG circuitry to shift configuration data into the device. The Quartus II software automatically generates **.sof**s that you can use for JTAG configuration with a download cable in the Quartus II software programmer.

For more information about JTAG boundary-scan testing and commands available using Stratix IV devices, refer to the following documents:

- *JTAG Boundary Scan Testing* chapter
- *Programming Support for Jam STAPL Language*

Stratix IV devices are designed such that JTAG instructions have precedence over any device configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration of Stratix IV devices during PS configuration, PS configuration is terminated and JTAG configuration begins.

You cannot use the Stratix IV decompression or design security features if you are configuring your Stratix IV device when using JTAG-based configuration.

A device operating in JTAG mode uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS`, and `TRST` pins have weak internal pull-up resistors (typically 25 kΩ). The JTAG output pin `TDO` and all JTAG input pins are powered by 2.5-V/3.0-V V_{CCPD}. All the JTAG pins only support the LVTTL I/O standard.

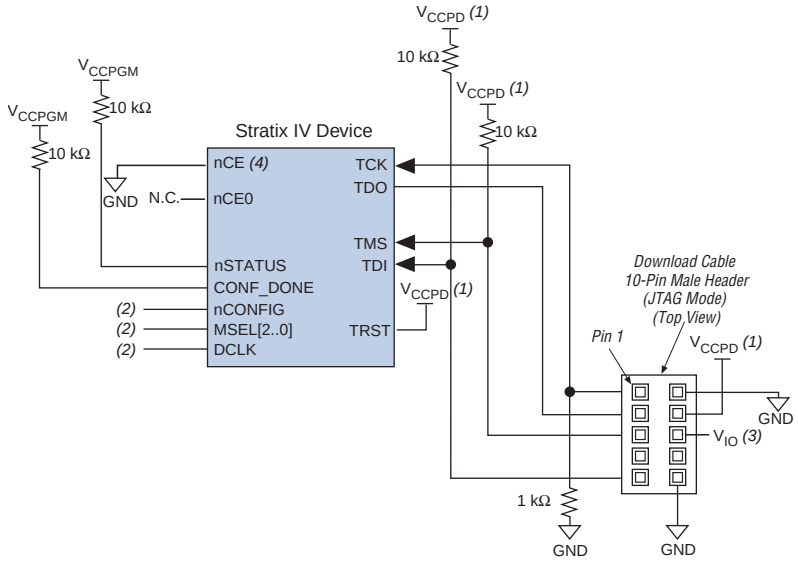
All user I/O pins are tri-stated during JTAG configuration.

The `TDO` output is powered by the V_{CCPD} power supply of I/O bank 1A. For more information about how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing* chapter.

During JTAG configuration, you can download data to the device on the PCB through the USB Blaster, MasterBlaster, ByteBlaster II, EthernetBlaster, or ByteBlasterMV download cable. Configuring devices through a cable is similar to programming devices in-system, except you must connect the TRST pin to V_{CCPD}. This ensures that the TAP controller is not reset.

Figure 10–16 shows JTAG configuration of a single Stratix IV device when using a download cable.

Figure 10–16. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 10–16:

(1) Connect the pull-up resistor to the same supply voltage as the USB Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable. The voltage supply can be connected to the V_{CCPD} of the device.

(2) Connect the nCONFIG and MSEL[2..0] pins to support a non-JTAG configuration scheme. If you only use the JTAG configuration, connect nCONFIG to V_{CCPGM} and MSEL[2..0] to GND. Pull DCLK either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPD}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cable, this pin is a no connect.

(4) You must connect nCE to GND or driven low for successful JTAG configuration.

To configure a single device in a JTAG chain, the programming software places all other devices in bypass mode. In bypass mode, devices pass programming data from the TDI pin to the TDO pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the TDO pin one clock cycle later.

The Quartus II software verifies successful JTAG configuration upon completion. At the end of configuration, the software checks the state of CONF_DONE through the JTAG port. When the Quartus II software generates a JAM file (**.jam**) for a multi-device chain, it contains instructions so that all the devices in the chain are initialized at the same time. If CONF_DONE is not high, the Quartus II software indicates that configuration has failed. If CONF_DONE is high, the software indicates that configuration was successful. After the configuration bitstream is transmitted serially using the JTAG TDI port, the TCK port is clocked an additional 1,094 cycles to perform device initialization.

Stratix IV devices have dedicated JTAG pins that always function as JTAG pins. Not only can you perform JTAG testing on Stratix IV devices before and after, but also during configuration. While other device families do not support JTAG testing during configuration, Stratix IV devices support the bypass, ID code, and sample instructions during configuration without interrupting configuration. All other JTAG instructions may only be issued by first interrupting configuration and reprogramming the I/O pins using the CONFIG_IO instruction.

The CONFIG_IO instruction allows I/O buffers to be configured using the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix IV device or waiting for a configuration device to complete configuration. After configuration has been interrupted and JTAG testing is complete, you must reconfigure the part using JTAG (PULSE_CONFIG instruction) or by pulsing nCONFIG low.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Stratix IV devices do not affect JTAG boundary-scan or programming operations. Toggling these pins does not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration for Stratix IV devices, consider the dedicated configuration pins. Table 10–7 lists how these pins are connected during JTAG configuration.

Table 10–7. Dedicated Configuration Pin Connections During JTAG Configuration

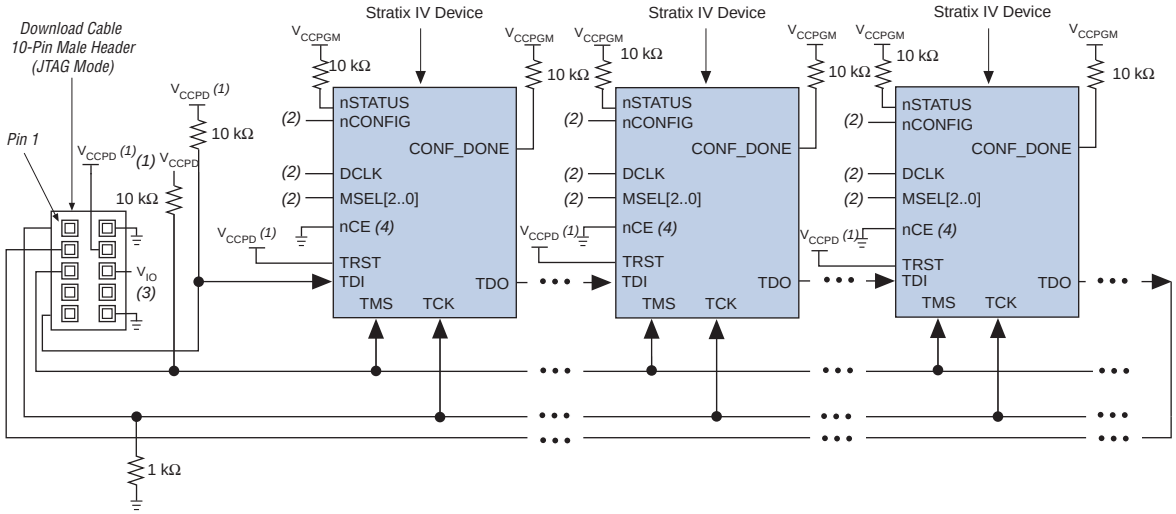
Signal	Description
nCE	On all Stratix IV devices in the chain, nCE must be driven low by connecting it to ground, pulling it low using a resistor, or driving it by some control circuitry. For devices that are also in multi-device FPP, AS, or PS configuration chains, the nCE pins must be connected to GND during JTAG configuration or JTAG must be configured in the same order as the configuration chain.
nCEO	On all Stratix IV devices in the chain, you can leave nCEO floating or connected to the nCE of the next device.
MSEL	Do not leave these pins floating. These pins support whichever non-JTAG configuration is used in production. If you only use JTAG configuration, tie these pins to GND.
nCONFIG	Driven high by connecting to V_{CCPGM}, pulling up using a resistor, or driven high by some control circuitry.
nSTATUS	Pull to V_{CCPGM} using a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin must be pulled up to V_{CCPGM} individually.
CONF_DONE	Pull to V_{CCPGM} using a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin must be pulled up to V_{CCPGM} individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration.
DCLK	Do not leave DCLK floating. Drive low or high, whichever is more convenient on your board.

When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer.

JTAG-chain device programming is ideal when the system contains multiple devices, or when testing your system using JTAG BST circuitry.

Figure 10–17 shows a multi-device JTAG configuration when using a download cable.

Figure 10–17. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 10–17:

(1) Connect the pull-up resistor to the same supply voltage as the USB Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable. Connect the voltage supply to V_{CCPD} of the device.

(2) Connect the nCONFIG and MSEL[2..0] pins to support a non-JTAG configuration scheme. If you only use JTAG configuration, connect nCONFIG to V_{CCPGM} and MSEL[2..0] to GND. Pull DCLK either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPD}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cables, this pin is a no connect.

(4) You must connect nCE to GND or drive it low for successful JTAG configuration.

You must connect the nCE pin to GND or drive it low during JTAG configuration. In multi-device FPP, AS, and PS configuration chains, the first device's nCE pin is connected to GND, while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. In addition, the CONF_DONE and nSTATUS signals are all shared in multi-device FPP, AS, or PS configuration chains so the devices can enter user mode at the same time after configuration is complete. When the CONF_DONE and nSTATUS signals are shared among all the devices, you must configure every device when JTAG configuration is performed.

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 10–17, where each of the CONF_DONE and nSTATUS signals are isolated, so that each device can enter user mode individually.

After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, ensure the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the nCEO of the previous device drives the nCE of the next device low when it has successfully been JTAG configured.

You can place other Altera devices that have JTAG support in the same JTAG chain for device programming and configuration.

JTAG configuration support is enhanced and allows more than 17 Stratix IV devices to be cascaded in a JTAG chain.

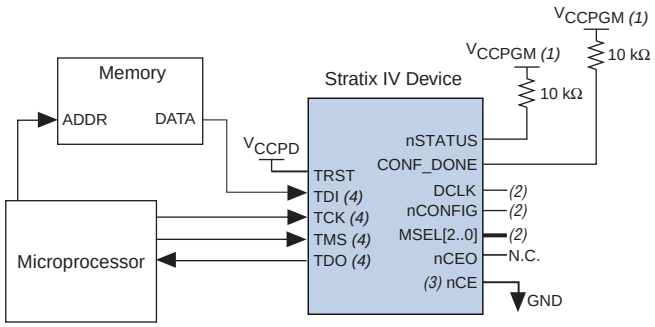
For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera Device Chains* chapter in volume 2 of the *Configuration Handbook*.

You can configure Stratix IV devices using multiple configuration schemes on the same board. Combining JTAG configuration with AS configuration on your board is useful in the prototyping environment because it allows multiple methods to configure your FPGA.

For more information about combining JTAG configuration with other configuration schemes, refer to the *Combining Different Configuration Schemes* chapter in volume 2 of the *Configuration Handbook*.

Figure 10–18 shows JTAG configuration of a Stratix IV device using a microprocessor.

Figure 10–18. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 10–18:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device.

(2) Connect the nCONFIG and MSEL[2..0] pins to support a non-JTAG configuration scheme. If you use only the JTAG configuration, connect nCONFIG to V_{CCGPM} and MSEL[2..0] to GND. Pull DCLK either high or low, whichever is convenient on your board.

(3) Connect nCE to GND or drive it low for successful JTAG configuration.

(4) The microprocessor must use the same I/O standard as V_{CCPD} to drive the JTAG pins.

Jam STAPL

Jam™ STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface. Jam STAPL is a freely licensed open standard.

The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 122: Using Jam STAPL for ISP and ICR via an Embedded Processor*. To download the Jam Player, visit the Altera website at www.altera.com.

Device Configuration Pins

The following tables list the connections and functionality of all the configuration-related pins on Stratix IV devices. Table 10–8 lists the Stratix IV configuration pins and their power supply.

Table 10–8. Stratix IV Configuration Pin Summary (Part 1 of 2) *(Note 1)*

Description	Input/Output	Dedicated	Powered By	Configuration Mode
TDI	Input	Yes	V_{CCPD}	JTAG
TMS	Input	Yes	V_{CCPD}	JTAG
TCK	Input	Yes	V_{CCPD}	JTAG
TRST	Input	Yes	V_{CCPD}	JTAG
TDO	Output	Yes	V_{CCPD}	JTAG
CRC_ERROR	Output	—	Pull-up	Optional, all modes
DATA0	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	All modes except JTAG
DATA[7..1]	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	FPP
INIT_DONE	Output	—	Pull-up	Optional, all modes
CLKUSR	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	Optional
nSTATUS	Bidirectional	Yes	V_{CCPGM}/Pull-up	All modes
nCE	Input	Yes	V_{CCPGM}	All modes
CONF_DONE	Bidirectional	Yes	V_{CCPGM}/Pull-up	All modes
nCONFIG	Input	Yes	V_{CCPGM}	All modes
PORSEL	Input	Yes	V_{CC} *(2)*	All modes
ASDO	Output	Yes	V_{CCPGM}	AS
nCSO	Output	Yes	V_{CCPGM}	AS
DCLK	Input	Yes	V_{CCPGM}	PS, FPP
	Output	Yes	V_{CCPGM}	AS
nIO_PULLUP	Input	Yes	V_{CC} *(2)*	All modes
nCEO	Output	Yes	V_{CCPGM}	All modes

Table 10–8. Stratix IV Configuration Pin Summary (Part 2 of 2) *(Note 1)*

Description	Input/Output	Dedicated	Powered By	Configuration Mode
MSEL[2..0]	Input	Yes	V_{CC} *(2)*	All modes

Notes to Table 10–8:

(1) The total number of pins is 29. The total number of dedicated pins is 18.

(2) Although MSEL[2..0], PORSEL, and nIO_PULLUP are powered up by V_{CC}, Altera recommends you connect these pins to V_{CCPGM} or GND directly without using a pull-up or pull-down resistor.

(3) These pins are powered up by V_{CCPGM} during configuration. These pins are powered up by V_{CCIO} if they are used as regular I/O in user mode.

Table 10–9 lists the dedicated configuration pins. You must connect these pins properly on your board for successful configuration. Some of these pins may not be required for your configuration schemes.

Table 10–9. Dedicated Configuration Pins on the Stratix IV Device (Part 1 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
VCCPGM	N/A	All	Power	Dedicated power pin. Use this pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bidirectional pins, and some of the dual functional pins that are used for configuration. You must connect this pin to 1.8, 2.5, or 3.0 V. V_{CCPGM} must ramp-up from 0 V to V_{CCPGM} within 100 ms when PORSEL is low or 4 ms when PORSEL is high. If V_{CCPGM} is not ramped up within this specified time, your Stratix IV device will not configure successfully. If your system does not allow a V_{CCPGM} ramp-up within 100 ms or 4 ms, you must hold nCONFIG low until all power supplies are stable.
VCCPD	N/A	All	Power	Dedicated power pin. Use this pin to power the I/O pre-drivers, JTAG input and output pins, and design security circuitry. You must connect this pin to 2.5 V or 3.0 V, depending on the I/O standards selected. For the 3.0-V I/O standard, V_{CCPD} = 3.0 V. For the 2.5 V or below I/O standards, V_{CCPD} = 2.5 V. V_{CCPD} must ramp-up from 0 V to 2.5 V / 3.0 V within 100 ms when PORSEL is low or 4 ms when PORSEL is high. If V_{CCPD} is not ramped up within this specified time, your Stratix IV device will not configure successfully. If your system does not allow a V_{CCPD} to ramp-up time within 100 ms or 4 ms, you must hold nCONFIG low until all power supplies are stable.
PORSEL	N/A	All	Input	Dedicated input that selects between a standard POR time or a fast POR time. A logic low selects a standard POR time setting of 100 ms < T_{POR} < 300 ms and a logic high selects a fast POR time setting of 4 ms < T_{POR} < 12 ms. The PORSEL input buffer is powered by V_{CC} and has an internal 5-kΩ pull-down resistor that is always active. Tie the PORSEL pin directly to V_{CCPGM} or GND.

Table 10–9. Dedicated Configuration Pins on the Stratix IV Device (Part 2 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nIO_PULLUP	N/A	All	Input	Dedicated input that chooses whether the internal pull-up resistors on the user I/O pins and dual-purpose I/O pins (nCSO, nASDO, DATA[7..0], CLKUSR, and INIT_DONE) are on or off before and during configuration. A logic high turns off the weak internal pull-up resistors; a logic low turns them on. The nIO-PULLUP input buffer is powered by V_{CC} and has an internal 5-kΩ pull-down resistor that is always active. Tie the nIO-PULLUP directly to V_{CCPGM} or GND.
MSEL[2..0]	N/A	All	Input	Three-bit configuration input that sets the Stratix IV device configuration scheme. For the appropriate connections, refer to Table 10–1 on page 10–2 . You must hardwire these pins to V_{CCPGM} or GND. The MSEL[2..0] pins have internal 5-kΩ pull-down resistors that are always active.
nCONFIG	N/A	All	Input	Configuration control input. Pulling this pin low during user-mode causes the device to lose its configuration data, enter a reset state, and tri-state all I/O pins. Returning this pin to a logic high level initiates a reconfiguration. Configuration is possible only if this pin is high, except in JTAG programming mode, when nCONFIG is ignored.
nSTATUS	N/A	All	Bidirectional open-drain	The device drives nSTATUS low immediately after power-up and releases it after the POR time. During user mode and regular configuration, this pin is pulled high by an external 10-kΩ resistor. This pin, when driven low by the Stratix IV device, indicates that the device has encountered an error during configuration. ■ Status output—If an error occurs during configuration, nSTATUS is pulled low by the target device. ■ Status input—If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. Driving nSTATUS low after configuration and initialization does not affect the configured device. If you use a configuration device, driving nSTATUS low causes the configuration device to attempt to configure the device, but because the device ignores transitions on nSTATUS in user mode, the device does not reconfigure. To initiate a reconfiguration, nCONFIG must be pulled low.

Table 10–9. Dedicated Configuration Pins on the Stratix IV Device (Part 3 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nSTATUS (continued)	—	—	—	If V_{CCPGM} is not fully powered up, the following could occur: ■ V_{CCPGM} is powered high enough for the nSTATUS buffer to function properly and nSTATUS is driven low. When V_{CCPGM} is ramped up, POR trips and nSTATUS is released after POR expires. ■ V_{CCPGM} is not powered high enough for the nSTATUS buffer to function properly. In this situation, nSTATUS might appear logic high, triggering a configuration attempt that would fail because POR did not yet trip. When V_{CCPD} is powered up, nSTATUS is pulled low because POR did not yet trip. When POR trips after V_{CCPGM} is powered up, nSTATUS is released and pulled high. At that point, reconfiguration is triggered and the device is configured.
CONF_DONE	N/A	All	Bidirectional open-drain	Status output. The target device drives the CONF_DONE pin low before and during configuration. After all the configuration data is received without error and the initialization cycle starts, the target device releases CONF_DONE. Status input. After all the data is received and CONF_DONE goes high, the target device initializes and enters user mode. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize. Driving CONF_DONE low after configuration and initialization does not affect the configured device.
nCE	N/A	All	Input	Active-low chip enable. The nCE pin activates the device with a low signal to allow configuration. The nCE pin must be held low during configuration, initialization, and user mode. In single device configuration, it must be tied low. In multi-device configuration, nCE of the first device is tied low, while its nCEO pin is connected to nCE of the next device in the chain. The nCE pin must also be held low for successful JTAG programming of the device.
nCEO	N/A	All	Output	Output that drives low when device configuration is complete. In single device configuration, this pin is left floating. In multi-device configuration, this pin feeds the next device's nCE pin. The nCEO of the last device in the chain is left floating. The nCEO pin is powered by V_{CCPGM}.
ASDO	N/A	AS	Output	Control signal from the Stratix IV device to the serial configuration device in AS mode used to read out configuration data. In AS mode, ASDO has an internal pull-up resistor that is always active.
nCSO	N/A	AS	Output	Output control signal from the Stratix IV device to the serial configuration device in AS mode that enables the configuration device. In AS mode, nCSO has an internal pull-up resistor that is always active.

Table 10–9. Dedicated Configuration Pins on the Stratix IV Device (Part 4 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DCLK	N/A	Synchronous configuration schemes (PS, FPP, AS)	Input (PS, FPP) Output (AS)	In PS and FPP configuration, DCLK is the clock input used to clock data from an external source into the target device. Data is latched into the device on the rising edge of DCLK. In AS mode, DCLK is an output from the Stratix IV device that provides timing for the configuration interface. In AS mode, DCLK has an internal pull-up resistor (typically 25 kΩ) that is always active. After configuration, this pin is tri-stated. In schemes that use a configuration device, DCLK is driven low after configuration is done. In schemes that use a control host, DCLK must be driven either high or low, whichever is more convenient. Toggling this pin after configuration does not affect the configured device.
DATA0	N/A in AS mode. I/O in PS or FPP mode.	PS, FPP, AS	Input	Data input. In serial configuration modes, bit-wide configuration data is presented to the target device on the DATA0 pin. In AS mode, DATA0 has an internal pull-up resistor that is always active. After PS or FPP configuration, DATA0 is available as a user I/O pin. The state of this pin depends on the **Dual-Purpose Pin** settings.
DATA[7..1]	I/O	Parallel configuration schemes (FPP)	Inputs	Data inputs. Byte-wide configuration data is presented to the target device on DATA[7..0]. In serial configuration schemes, they function as user I/O pins during configuration, which means they are tri-stated. After FPP configuration, DATA[7..1] are available as user I/O pins. The state of these pins depends on the **Dual-Purpose Pin** settings.

Table 10–10 lists the optional configuration pins. If these optional configuration pins are not enabled in the Quartus II software, they are available as general-purpose user I/O pins. Therefore, during configuration, these pins function as user I/O pins and are tri-stated with weak pull-up resistors.

Table 10–10. Optional Configuration Pins

Pin Name	User Mode	Pin Type	Description
CLKUSR	N/A if option is on. I/O if option is off.	Input	Optional user-supplied clock input synchronizes the initialization of one or more devices. Enable this pin by turning on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software.
INIT_DONE	N/A if option is on. I/O if option is off.	Output open-drain	Use as a status pin to indicate when the device has initialized and is in user mode. When nCONFIG is low and during the beginning of configuration, the INIT_DONE pin is tri-stated and pulled high due to an external 10-kΩ pull-up resistor. After the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization is complete, the INIT_DONE pin is released and pulled high and the device enters user mode. Thus, the monitoring circuitry must be able to detect a low-to-high transition. Enable this pin by turning on the **Enable INIT_DONE output** option in the Quartus II software.
DEV_OE	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all tri-states on the device. When this pin is driven low, all I/O pins are tri-stated. When this pin is driven high, all I/O pins behave as programmed. Enable this pin by turning on the **Enable device-wide output enable** (**DEV_OE**) option in the Quartus II software.
DEV_CLRn	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all clears on all device registers. When this pin is driven low, all registers are cleared. When this pin is driven high, all registers behave as programmed. Enable this pin by turning on the **Enable device-wide reset** (**DEV_CLRn**) option in the Quartus II software.

Table 10–11 lists the dedicated JTAG pins. JTAG pins must be kept stable before and during configuration to prevent accidental loading of JTAG instructions. The TDI, TMS, and TRST pins have weak internal pull-up resistors, while TCK has a weak internal pull-down resistor (typically 25 kΩ). If you plan to use the SignalTap® embedded logic array analyzer, you must connect the JTAG pins of the Stratix IV device to a JTAG header on your board.

Table 10–11. Dedicated JTAG Pins

Pin Name	User Mode	Pin Type	Description
TDI	N/A	Test data input	Serial input pin for instructions as well as test and programming data. Data is shifted on the rising edge of TCK. The TDI pin is powered by the 2.5-V/3.0-V V_{CCPD} supply. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting this pin to logic high using a 1-kΩ resistor.
TDO	N/A	Test data output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. The TDO pin is powered by V_{CCPD}. For recommendations about connecting a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing* chapter. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by leaving this pin unconnected.
TMS	N/A	Test mode select	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. TMS is evaluated on the rising edge of TCK. Therefore, you must set up TMS before the rising edge of TCK. Transitions within the state machine occur on the falling edge of TCK after the signal is applied to TMS. The TMS pin is powered by 2.5-V/3.0-V V_{CCPD}. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting this pin to logic high using a 1-kΩ resistor.
TCK	N/A	Test clock input	Clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. The TCK pin is powered by the 2.5-V/3.0-V V_{CCPD} supply. It is expected that the clock input waveform have a nominal 50% duty cycle. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting TCK to GND.
TRST	N/A	Test reset input (optional)	Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to IEEE Std. 1149.1. The TRST pin is powered by the 2.5-V/3.0-V V_{CCPD} supply. Hold TMS at 1 or keep TCK static while TRST is changed from 0 to 1. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting the TRST pin to GND.

For more information about the pin connection recommendations, refer to the *Stratix IV GX Device Family Pin Connection Guidelines*.

Configuration Data Decompression

Stratix IV devices support configuration data decompression, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other memory and transmit this compressed bitstream to Stratix IV devices. During configuration, the Stratix IV device decompresses the bitstream in real time and programs its SRAM cells.

Preliminary data indicates that compression typically reduces the configuration bitstream size by 35 to 55% based on the designs used.

Stratix IV devices support decompression in the FPP (when using a MAX II device or microprocessor + flash), fast AS, and PS configuration schemes. The Stratix IV decompression feature is not available in the JTAG configuration scheme.

In PS mode, use the Stratix IV decompression feature because sending compressed configuration data reduces configuration time.

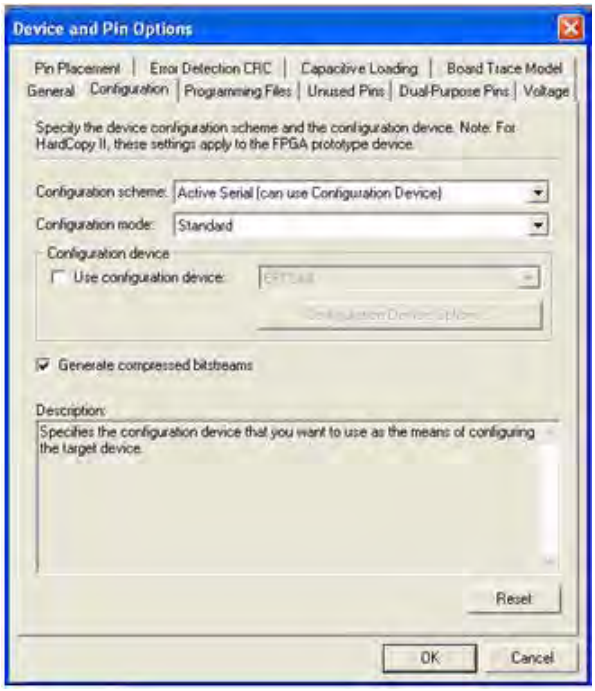
When you enable compression, the Quartus II software generates configuration files with compressed configuration data. This compressed file reduces the storage requirements in the configuration device or flash memory, and decreases the time needed to transmit the bitstream to the Stratix IV device. The time required by a Stratix IV device to decompress a configuration file is less than the time needed to transmit the configuration data to the device.

There are two ways to enable compression for Stratix IV bitstreams: before design compilation (in the Compiler Settings menu) and after design compilation (in the **Convert Programming Files** window).

To enable compression in the project's Compiler Settings menu, perform the following steps:

1. On the Assignments menu, click **Device** to bring up the **Settings** dialog box.
2. After selecting your Stratix IV device, open the **Device and Pin Options** window.
3. In the **Configuration** settings tab, turn on **Generate compressed bitstreams** (as shown in Figure 10–19).

Figure 10–19. Enabling Compression for Stratix IV Bitstreams in Compiler Settings



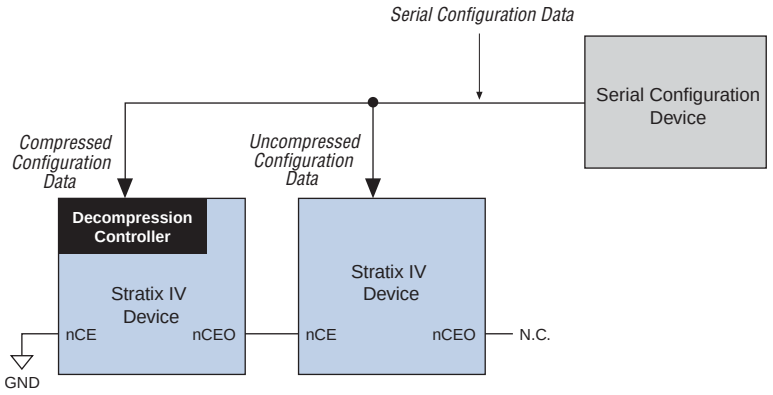
You can also enable compression when creating programming files from the **Convert Programming Files** window. To do this, perform the following steps:

1. On the File menu, click **Convert Programming Files**.
2. Select the programming file type (**.pof**, **.sram**, **.hex**, **.rbf**, or **.ttf**).
3. For **.pof** output files, select a configuration device.
4. In the **Input files to convert** box, select **SOF Data**.
5. Select **Add File** and add a Stratix IV device **.sof** file.
6. Select the name of the file you added to the **SOF Data** area and click **Properties**.
7. Check the **Compression** check box.

When multiple Stratix IV devices are cascaded, you can selectively enable the compression feature for each device in the chain if you are using a serial configuration scheme. Figure 10–20 shows a chain of two Stratix IV devices. The first Stratix IV device has compression enabled; therefore, receives a compressed bitstream from the configuration device. The second Stratix IV device has the compression feature disabled and receives uncompressed data.

In a multi-device FPP configuration chain (with a MAX II device or microprocessor + flash), all Stratix IV devices in the chain must either enable or disable the decompression feature. You cannot selectively enable the compression feature for each device in the chain because of the DATA and DCLK relationship.

Figure 10–20. Compressed and Uncompressed Configuration Data in the Same Configuration File



You can generate programming files for this setup by clicking **Convert Programming Files** on the File menu in the Quartus II software.

Remote System Upgrades

This section describes the functionality and implementation of the dedicated remote system upgrade circuitry. It also defines several concepts related to remote system upgrade, including factory configuration, application configuration, remote update mode, and user watchdog timer. Additionally, this section provides design guidelines for implementing remote system upgrades with the supported configuration schemes.

System designers sometimes face challenges such as shortened design cycles, evolving standards, and system deployments in remote locations. Stratix IV devices help overcome these challenges with their inherent reprogrammability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduce time-to-market, extend product life, and avoid system downtime.

Stratix IV devices feature dedicated remote system upgrade circuitry. Soft logic (either the Nios® II embedded processor or user logic) implemented in a Stratix IV device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information.

Remote system upgrade is supported in fast AS Stratix IV configuration schemes. You can also implement remote system upgrade in conjunction with advanced Stratix IV features such as real-time decompression of configuration data and design security using the advanced encryption standard (AES) for secure and efficient field upgrades. The largest serial configuration device currently supports 128 Mbits of configuration bitstream.

Stratix IV devices only support remote system upgrade in the single device fast AS configuration scheme. Because the largest serial configuration device currently supports 128 Mbits of configuration bitstream, the remote system upgrade feature is not supported in EP4SGX290, EP4SE360, and larger devices.

The remote system upgrade feature is not supported in a multi-device chain.

Functional Description

The dedicated remote system upgrade circuitry in Stratix IV devices manages remote configuration and provides error detection, recovery, and status information. User logic or a Nios II processor implemented in the Stratix IV device logic array provides access to the remote configuration data source and an interface to the system's configuration memory.

Stratix IV devices have remote system upgrade processes that involve the following steps:

1. A Nios II processor (or user logic) implemented in the Stratix IV device logic array receives new configuration data from a remote location. The connection to the remote source uses a communication protocol such as the transmission control protocol/Internet protocol (TCP/IP), peripheral component interconnect (PCI), user datagram protocol (UDP), universal asynchronous receiver/transmitter (UART), or a proprietary interface.
2. The Nios II processor (or user logic) stores this new configuration data in non-volatile configuration memory.
3. The Nios II processor (or user logic) initiates a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any error(s) that might occur during or after the reconfiguration cycle and provides error status information to the user design.

Figure 10–21 shows the steps required for performing remote configuration updates. (The numbers in Figure 10–21 coincide with the steps just mentioned.)

Figure 10–21. Functional Diagram of Stratix IV Remote System Upgrade

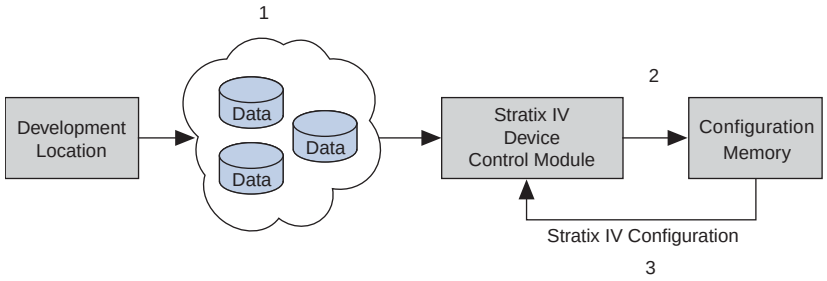
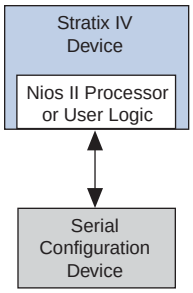


Figure 10–22 shows a block diagram for implementing a remote system upgrade with the Stratix IV fast AS configuration scheme.

Figure 10–22. Remote System Upgrade Block Diagram for Stratix IV Fast AS Configuration Scheme



You must set the mode select pins (MSEL[2..0]) to fast AS mode to use remote system upgrade in your system. Table 10–12 lists the MSEL pin settings for Stratix IV devices in standard configuration mode and remote system upgrade mode. The following sections describe remote update of the remote system upgrade mode.

For more information about standard configuration schemes supported in Stratix IV devices, refer to "Configuration Schemes" on page 10–2.

Table 10–12. Stratix IV Remote System Upgrade Modes

Configuration Scheme	MSEL[2..0]	Remote System Upgrade Mode
Fast AS (40 MHz)	011	Standard
	011	Remote update *(1)*

Note to Table 10–12:

(1) EPCS64 and EPCS128 serial configuration devices support a DCLK up to 40 MHz. For more information, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

When using fast AS mode, you must select remote update mode in the Quartus II software and insert the ALTREMOTE_UPDATE megafunction to access the circuitry. For more information, refer to "ALTREMOTE_UPDATE Megafunction" on page 10–58.

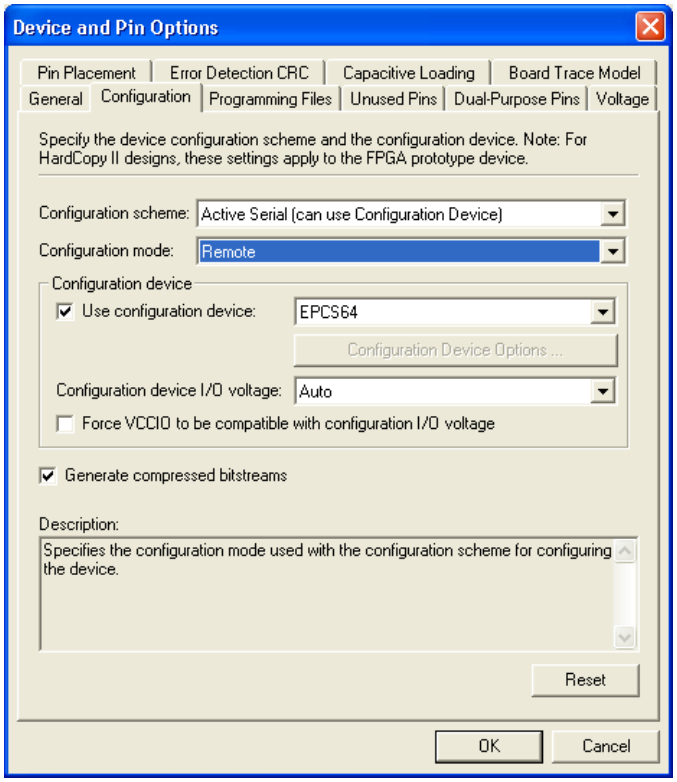
Enabling Remote Update

You can enable remote update for Stratix IV devices in the Quartus II software before design compilation (in the Compiler Settings menu). In remote update mode, the **auto-restart configuration after error** option is always enabled. To enable remote update in the project's compiler settings, perform the following steps in the Quartus II software:

1. On the Assignment menu, click **Device**. The **Settings** dialog box appears.
2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
3. Click the **Configuration** tab.
4. From the **Configuration scheme** list, select **Active Serial** (you can also use **Configuration Device**) (Figure 10–23).

5. From the **Configuration Mode** list, select **Remote** (Figure 10–23).
6. Click **OK**.
7. In the **Settings** dialog box, click **OK**.

Figure 10–23. Enabling Remote Update for Stratix IV Devices in the Compiler Settings Menu



Configuration Image Types

When performing a remote system upgrade, Stratix IV device configuration bitstreams are classified as factory configuration images or application configuration images. An image, also referred to as a configuration, is a design loaded into the Stratix IV device that performs certain user-defined functions.

Each Stratix IV device in your system requires one factory image or the addition of one or more application images. The factory image is a user-defined fall-back, or safe configuration, and is responsible for administering remote updates in conjunction with the dedicated circuitry. Application images implement user-defined functionality in the target Stratix IV device. You may include the default application image functionality in the factory image.

A remote system upgrade involves storing a new application configuration image or updating an existing one using the remote communication interface. After an application configuration image is stored or updated remotely, the user design in the Stratix IV device initiates a reconfiguration cycle with the new image. Any errors during or after this cycle are detected by the dedicated remote system upgrade

circuitry and cause the device to automatically revert to the factory image. The factory image then performs error processing and recovery. The factory configuration is written to the serial configuration device only once by the system manufacturer and must not be remotely updated. On the other hand, application configurations may be remotely updated in the system. Both images can initiate system reconfiguration.

Remote System Upgrade Mode

Remote system upgrade has one mode of operation—remote update mode. Remote update mode allows you to determine the functionality of your system upon power-up and offers several features.

Remote Update Mode

In remote update mode, Stratix IV devices load the factory configuration image after power up. The user-defined factory configuration determines which application configuration is to be loaded and triggers a reconfiguration cycle. The factory configuration may also contain application logic.

When used with serial configuration devices, remote update mode allows an application configuration to start at any flash sector boundary. For example, this translates to a maximum of 128 sectors in the EPCS64 device and 32 sectors in the EPCS16 device, where the minimum size of each page is 512 KBits. Altera recommends not using the same page in the serial configuration devices for two images. Additionally, remote update mode features a user watchdog timer that determines the validity of an application configuration.

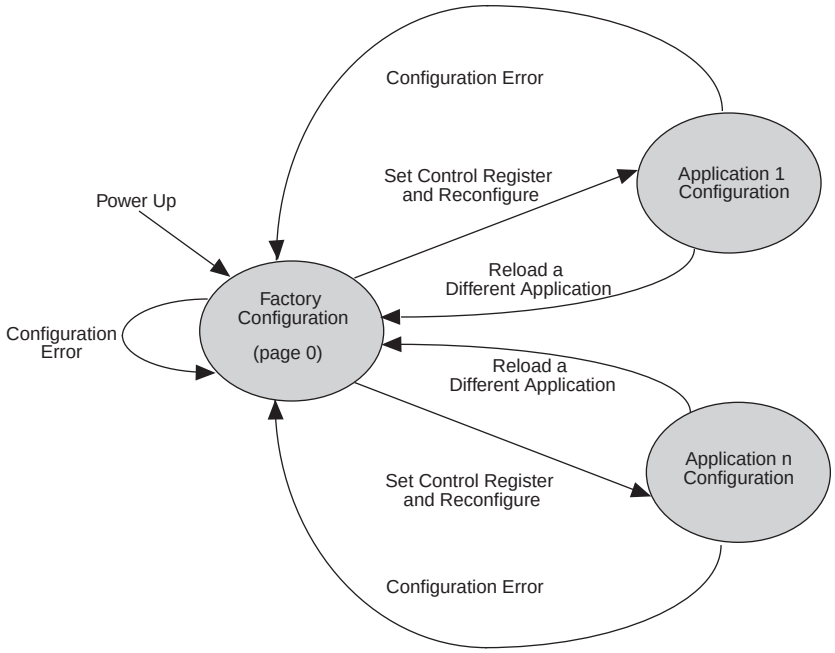
When a Stratix IV device is first powered up in remote update mode, it loads the factory configuration located at page zero (page registers `PGM[23:0] = 24'b0`). Always store the factory configuration image for your system at page address zero. This corresponds to the start address location 0×000000 in the serial configuration device.

The factory image is user-designed and contains soft logic to:

- Process any errors based on status information from the dedicated remote system upgrade circuitry
- Communicate with the remote host and receive new application configurations and store this new configuration data in the local non-volatile memory device
- Determine which application configuration is to be loaded into the Stratix IV device
- Enable or disable the user watchdog timer and load its time-out value (optional)
- Instruct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle

Figure 10–24 shows the transitions between the factory and application configurations in remote update mode.

Figure 10–24. Transitions between Configurations in Remote Update Mode



After power up or a configuration error, the factory configuration logic is loaded automatically. The factory configuration also must specify whether to enable the user watchdog timer for the application configuration and if enabled, to include the timer setting information.

The user watchdog timer ensures that the application configuration is valid and functional. The timer must be continually reset within a specific amount of time during user mode operation of an application configuration. Only valid application configurations contain the logic to reset the timer in user mode. This timer reset logic must be part of a user-designed hardware and/or software health monitoring signal that indicates error-free system operation. If the timer is not reset in a specific amount of time; for example, the user application configuration detects a functional problem or if the system hangs, the dedicated circuitry updates the remote system upgrade status register, triggering the loading of the factory configuration.

The user watchdog timer is automatically disabled for factory configurations. For more information about the user watchdog timer, refer to "User Watchdog Timer" on page 10–57.

If there is an error while loading the application configuration, the cause of the reconfiguration is written by the dedicated circuitry to the remote system upgrade status register. Actions that cause the remote system upgrade status register to be written are:

- nSTATUS driven low externally
- Internal CRC error

- User watchdog timer time-out
- A configuration reset (logic array nCONFIG signal or external nCONFIG pin assertion to low)

Stratix IV devices automatically load the factory configuration located at page address zero. This user-designed factory configuration can read the remote system upgrade status register to determine the reason for the reconfiguration. The factory configuration then takes appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

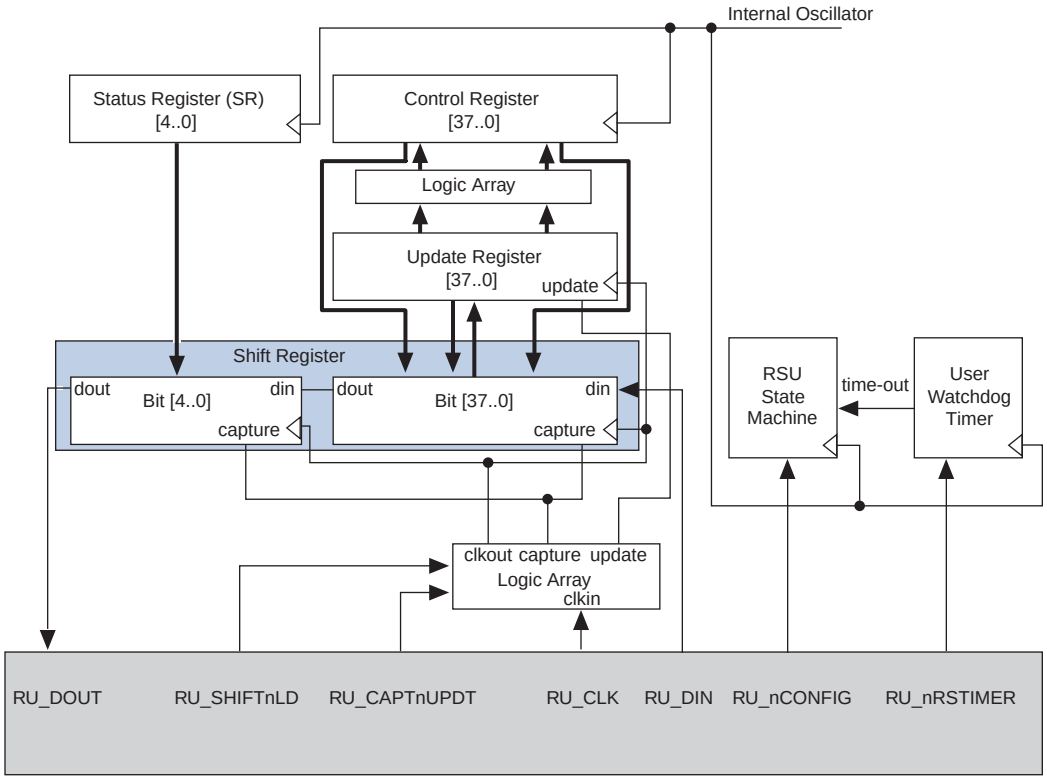
When Stratix IV devices successfully load the application configuration, they enter into user mode. In user mode, the soft logic (Nios II processor or state machine and the remote communication interface) assists the Stratix IV device in determining when a remote system update is arriving. When a remote system update arrives, the soft logic receives the incoming data, writes it to the configuration memory device, and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register and control register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and initiates system reconfiguration.

Dedicated Remote System Upgrade Circuitry

This section describes the implementation of the Stratix IV remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces to the user-defined factory and application configurations implemented in the Stratix IV device logic array to provide the complete remote configuration solution. The remote system upgrade circuitry contains the remote system upgrade registers, a watchdog timer, and a state machine that controls those components.

Figure 10–25 shows the data path for the remote system upgrade block.

Figure 10–25. Remote System Upgrade Circuit Data Path *(Note 1)*



Note to Figure 10–25:

(1) The RU_DOUT, RU_SHIFTnLD, RU_CAPTnUPDT, RU_CLK, RU_DIN, RU_nCONFIG, and RU_nRSTIMER signals are internally controlled by the ALTREMOTE_UPDATE megafunction.

Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that store the page addresses, watchdog timer settings, and status information. Table 10–13 lists these registers.

Table 10–13. Remote System Upgrade Registers (Part 1 of 2)

Register	Description
Shift register	This register is accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic.
Control register	This register contains the current page address, user watchdog timer settings, and one bit specifying whether the current configuration is a factory configuration or an application configuration. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is initiated, the contents of the update register are written into the control register.

Table 10–13. Remote System Upgrade Registers (Part 2 of 2)

Register	Description
Update register	This register contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a capture in a factory configuration, this register is read into the shift register.
Status register	This register is written to by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

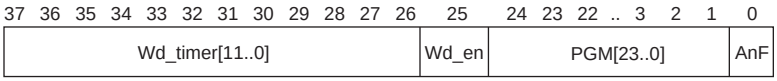
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (RU_CLK).

Remote System Upgrade Control Register

The remote system upgrade control register stores the application configuration page address and user watchdog timer settings. The control register functionality depends on the remote system upgrade mode selection. In remote update mode, the control register page address bits are set to all zeros (`24'b0 = 0×000000`) at power up to load the factory configuration. A factory configuration in remote update mode has write access to this register.

Figure 10–26 and Table 10–14 specify the control register bit positions. In the figure, the numbers show the bit position of a setting within a register. For example, bit number 25 is the enable bit for the watchdog timer.

Figure 10–26. Remote System Upgrade Control Register



The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix IV device is the factory configuration or an application configuration. This bit is set low by the remote system upgrade circuitry when an error condition causes a fall-back to the factory configuration. When the AnF bit is high, the control register access is limited to read operations. When the AnF bit is low, the register allows write operations and disables the watchdog timer.

In remote update mode, the factory configuration design sets this bit high (1'b1) when updating the contents of the update register with the application page address and watchdog timer settings.

Table 10–14 lists the remote system upgrade control register contents.

Table 10–14. Remote System Upgrade Control Register Contents

Control Register Bit	Remote System Upgrade Mode	Value *(2)*	Definition
AnF *(1)*	Remote update	1'b0	Application not factory
PGM[23..0]	Remote update	24'b0×000000	AS configuration start address (StAdd[23..0])
Wd_en	Remote update	1'b0	User watchdog timer enable bit
Wd_timer[11..0]	Remote update	12'b000000000000	User watchdog time-out value (most significant 12 bits of 29-bit count value: {Wd_timer[11..0], 17'b0})

Notes to Table 10–14:

(1) In remote update mode, the remote configuration block does not update the AnF bit automatically (you can update it manually).

(2) This is the default value of the control register bit.

Remote System Upgrade Status Register

The remote system upgrade status register specifies the reconfiguration trigger condition. The various trigger and error conditions include:

- Cyclic redundancy check (CRC) error during application configuration
- nSTATUS assertion by an external device due to an error
- Stratix IV device logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- External configuration reset (nCONFIG) assertion
- User watchdog timer time-out

Figure 10–27 and Table 10–15 specify the contents of the status register. The numbers in the figure show the bit positions within a 5-bit register.

Figure 10–27. Remote System Upgrade Status Register

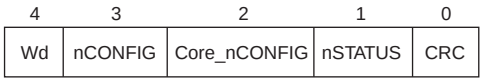


Table 10–15. Remote System Upgrade Status Register Contents (Part 1 of 2)

Status Register Bit	Definition	POR Reset Value
CRC (from the configuration)	CRC error caused reconfiguration	1 bit '0'
nSTATUS	nSTATUS caused reconfiguration	1 bit '0'
CORE_nCONFIG *(1)*	Device logic array caused reconfiguration	1 bit '0'
nCONFIG	nCONFIG caused reconfiguration	1 bit '0'

Table 10–15. Remote System Upgrade Status Register Contents (Part 2 of 2)

Status Register Bit	Definition	POR Reset Value
wd	Watchdog timer caused reconfiguration	1 bit '0'

Note to Table 10–15:

(1) Logic array reconfiguration forces the system to load the application configuration data into the Stratix IV device. This occurs after the factory configuration specifies the appropriate application configuration page address by updating the update register.

Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (refer to Table 10–13 on page 10–53). While both registers can only be updated when the device is loaded with a factory configuration image, the update register writes are controlled by the user logic; the control register writes are controlled by the remote system upgrade state machine.

In factory configurations, the user logic sends the AnF bit (set high), the page address, and the watchdog timer settings for the next application configuration bit to the update register. When the logic array configuration reset (RU_nCONFIG) goes low, the remote system upgrade state machine updates the control register with the contents of the update register and initiates system reconfiguration from the new application page.

To ensure successful reconfiguration between the pages, assert the RU_nCONFIG signal for a minimum of 250 ns. This is equivalent to strobing the reconfiguration input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

In the event of an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (page zero or page one, based on the mode and error condition) by setting the control register accordingly. Table 10–16 lists the contents of the control register after such an event occurs for all possible error or trigger conditions.

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition but before the factory configuration is loaded.

Table 10–16. Control Register Contents after an Error or Reconfiguration Trigger Condition

Reconfiguration Error/Trigger	Control Register Setting Remote Update
nCONFIG reset	All bits are 0
nSTATUS error	All bits are 0
CORE triggered reconfiguration	Update register
CRC error	All bits are 0
wd time out	All bits are 0

Capture operations during factory configuration access the contents of the update register. This feature is used by the user logic to verify that the page address and watchdog timer settings were written correctly. Read operations in application configurations access the contents of the control register. This information is used by the user logic in the application configuration.

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from stalling the device indefinitely. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Stratix IV device.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29 bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The granularity of the timer setting is 2^{15} cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator. Table 10–17 lists the operating range of the 10-MHz internal oscillator.

Table 10–17. 10-MHz Internal Oscillator Specifications *(Note 1)*

Minimum	Typical	Maximum	Units
4.3	5.3	10	MHz

Note to Table 10–17:

(1) These values are preliminary.

The user watchdog timer begins counting once the application configuration enters device user mode. This timer must be periodically reloaded or reset by the application configuration before the timer expires by asserting `RU_nRSTIMER`. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (`Wd`) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

To allow remote system upgrade dedicated circuitry to reset the watchdog timer, you must assert the `RU_nRSTIMER` signal active for a minimum of 250 ns. This is equivalent to strobing the `reset_timer` input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

The user watchdog timer is not enabled during the configuration cycle of the device. Errors during configuration are detected by the CRC engine. Also, the timer is disabled for factory configurations. Functional errors should not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

The user watchdog timer is disabled in factory configurations and during the configuration cycle of the application configuration. It is enabled after the application configuration enters user mode.

Quartus II Software Support

The Quartus II software provides the flexibility to include the remote system upgrade interface between the Stratix IV device logic array and the dedicated circuitry, generate configuration files for production, and allows remote programming of the system configuration memory.

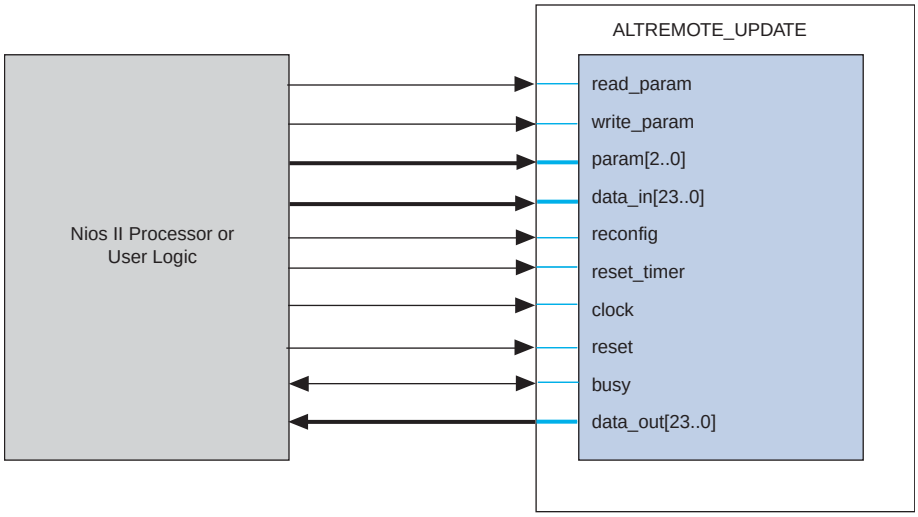
The ALTREMOTE_UPDATE megafunction is the implementation option in the Quartus II software that you use for the interface between the remote system upgrade circuitry and the device logic array interface. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

ALTREMOTE_UPDATE Megafunction

The ALTREMOTE_UPDATE megafunction provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read and write protocol in the Stratix IV device logic. This implementation is suitable for designs that implement the factory configuration functions using a Nios II processor or user logic in the device.

Figure 10–28 shows the interface signals between the ALTREMOTE_UPDATE megafunction and Nios II processor or user logic.

Figure 10–28. Interface Signals between the ALTREMOTE_UPDATE Megafunction and the Nios II Processor



For more information about the ALTREMOTE_UPDATE megafunction and the description of ports listed in Figure 10–28, refer to the *ALTREMOTE_UPDATE Megafunction User Guide*.

Design Security

This section provides an overview of the design security feature and its implementation on Stratix IV devices using the advanced encryption standard (AES). It also covers the new security modes available in Stratix IV devices.

As Stratix IV devices continue play a role in larger and more critical designs in competitive commercial and military environments, it is increasingly important to protect the designs from copying, reverse engineering, and tampering.

Stratix IV devices address these concerns with both volatile and non-volatile security feature support. Stratix IV devices have the ability to decrypt configuration bitstreams using the AES algorithm, an industry-standard encryption algorithm that is FIPS-197 certified. Stratix IV devices have a design security feature that utilizes a 256-bit security key.

Stratix IV devices store configuration data in SRAM configuration cells during device operation. Because SRAM is volatile, the SRAM cells must be loaded with configuration data each time the device powers up. It is possible to intercept configuration data when it is being transmitted from the memory source (flash memory or a configuration device) to the device. The intercepted configuration data could then be used to configure another device.

When using the Stratix IV design security feature, the security key is stored in the Stratix IV device. Depending on the security mode, you can configure the Stratix IV device using a configuration file that is encrypted with the same key, or for board testing, configured with a normal configuration file.

The design security feature is available when configuring Stratix IV devices using FPP configuration mode with an external host (such as a MAX II device or microprocessor), or when using fast AS or PS configuration schemes. The design security feature is also available in remote update with fast AS configuration mode. The design security feature is not available when you are configuring your Stratix IV device using JTAG-based configuration. For more information, refer to "Supported Configuration Schemes" on page 10–63.

When using a serial configuration scheme such as PS or fast AS, configuration time is the same whether or not you enable the design security feature. If the FPP scheme is used with the design security or decompression feature, a ×4 DCLK is required. This results in a slower configuration time when compared with the configuration time of a Stratix IV device that has neither the design security nor the decompression feature enabled.

Stratix IV Security Protection

Stratix IV device designs are protected from copying, reverse engineering, and tampering using configuration bitstream encryption.

Security Against Copying

The security key is securely stored in the Stratix IV device and cannot be read out through any interfaces. In addition, as configuration file read-back is not supported in Stratix IV devices, the design information cannot be copied.

Security Against Reverse Engineering

Reverse engineering from an encrypted configuration file is very difficult and time consuming because the Stratix IV configuration file formats are proprietary and the file contains millions of bits which require specific decryption. Reverse engineering the Stratix IV device is just as difficult because the device is manufactured on the most advanced 40-nm process technology.

Security Against Tampering

The non-volatile keys are one-time programmable. After the Tamper Protection bit is set in the key programming file generated by the Quartus II software, the Stratix IV device can only be configured with configuration files encrypted with the same key.

AES Decryption Block

The main purpose of the AES decryption block is to decrypt the configuration bitstream prior to entering data decompression or configuration.

Prior to receiving encrypted data, you must enter and store the 256-bit security key in the device. You can choose between a non-volatile security key and a volatile security key with battery backup.

The security key is scrambled prior to storing it in the key storage to make it more difficult for anyone to retrieve the stored key using de-capsulation of the device.

Flexible Security Key Storage

Stratix IV devices support two types of security key programming—volatile and non-volatile keys. Table 10–18 lists the differences between volatile keys and non-volatile keys.

Table 10–18. Security Key Options

Options	Volatile Key	Non-Volatile Key
Key programmability	Reprogrammable and erasable	One-time programmable
External battery	Required	Not required
Key programming method *(1)*	On-board	On and off board
Design protection	Secure against copying and reverse engineering	Secure against copying and reverse engineering. Tamper resistant if tamper protection bit is set.

Note to Table 10–18:

(1) Key programming is carried out using the JTAG interface.

You can program the non-volatile key to the Stratix IV device without an external battery. Also, there are no additional requirements to any of the Stratix IV power supply inputs.

V_{CCBAT} is a dedicated power supply for volatile key storage and not shared with other on-chip power supplies, such as V_{CCIO} or V_{CC}. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition.

After power-up, you must wait 300 ms (PORSEL = 0) or 12 ms (PORSEL = 1) before beginning key programming to ensure that V_{CCBAT} is at full rail.

For more information about how to calculate the key retention time of the battery used for volatile key storage, refer to the *Stratix IV PowerPlay Early Power Estimator*.

For more information about battery specifications, refer to the *DC and Switching Characteristics* chapter.

For more information about the V_{CCBAT} pin connection recommendations, refer to the *Stratix IV GX Device Family Pin Connection Guidelines*.

Stratix IV Design Security Solution

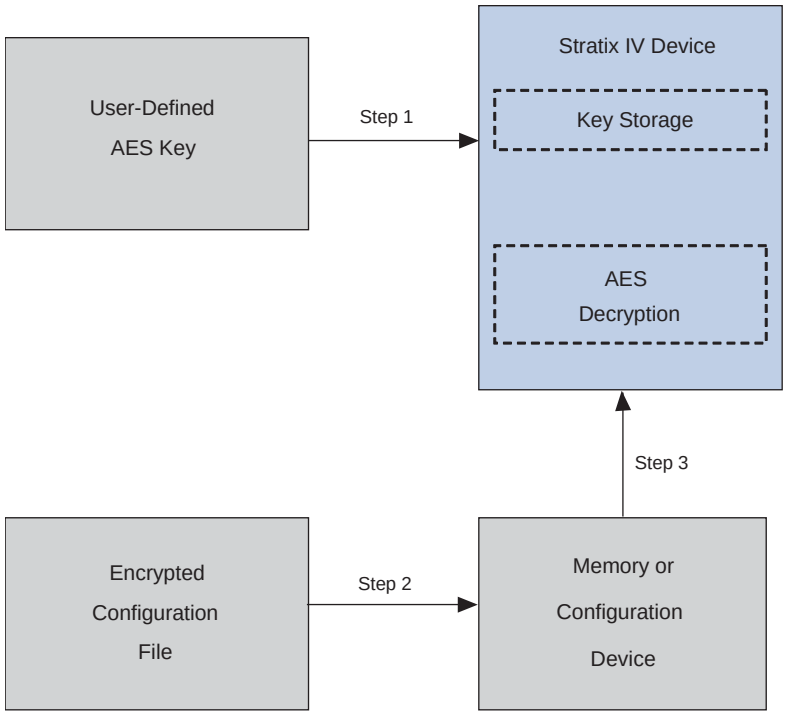
Stratix IV devices are SRAM-based devices. To provide design security, Stratix IV devices require a 256-bit security key for configuration bitstream encryption.

You can carry out secure configuration in the following steps, as shown in Figure 10–29:

1. Program the security key into the Stratix IV device.
2. Program the user-defined 256-bit AES keys to the Stratix IV device through the JTAG interface.
3. Encrypt the configuration file and store it in the external memory.
4. Encrypt the configuration file with the same 256-bit keys used to program the Stratix IV device. Encryption of the configuration file is done using the Quartus II software. The encrypted configuration file is then loaded into the external memory, such as a configuration or flash device.
5. Configure the Stratix IV device.

At system power-up, the external memory device sends the encrypted configuration file to the Stratix IV device.

Figure 10–29. Design Security *(Note 1)*



Note to Figure 10–29:

(1) Step 1, Step 2, and Step 3 correspond to the procedure described in "Design Security" on page 10–58.

Security Modes Available

The following security modes are available on the Stratix IV device:

Volatile Key

Secure operation with volatile key programmed and required external battery: this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board-level testing only.

Non-Volatile Key

Secure operation with one time programmable (OTP) security key programmed: this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board level testing only.

Non-Volatile Key with Tamper Protection Bit Set

Secure operation in tamper resistant mode with OTP security key programmed: only encrypted configuration bitstreams are allowed to configure the device. Tamper protection disables JTAG configuration with unencrypted configuration bitstream.

Enabling the tamper protection bit disables test mode in Stratix IV devices. This process is irreversible and prevents Altera from conducting carry-out failure analysis if test mode is disabled. Contact Altera Technical Support to enable the tamper protection bit.

No Key Operation

Only unencrypted configuration bitstreams are allowed to configure the device.

Table 10–19 lists the different security modes and configuration bitstream supported for each mode.

Table 10–19. Security Modes Supported

Mode *(1)*	Function	Configuration File
Volatile key	Secure	Encrypted
	Board-level testing	Unencrypted
Non-volatile key	Secure	Encrypted
	Board-level testing	Unencrypted
Non-volatile key with tamper protection bit set	Secure (tamper resistant) *(2)*	Encrypted

Notes to Table 10–19:

(1) In No key operation, only the unencrypted configuration file is supported.

(2) The tamper protection bit setting does not prevent the device from being reconfigured.

Supported Configuration Schemes

The Stratix IV device supports only selected configuration schemes, depending on the security mode you select when you encrypt the Stratix IV device.

Figure 10–30 shows the restrictions of each security mode when encrypting Stratix IV devices.

Figure 10–30. Stratix IV Security Modes—Sequence and Restrictions

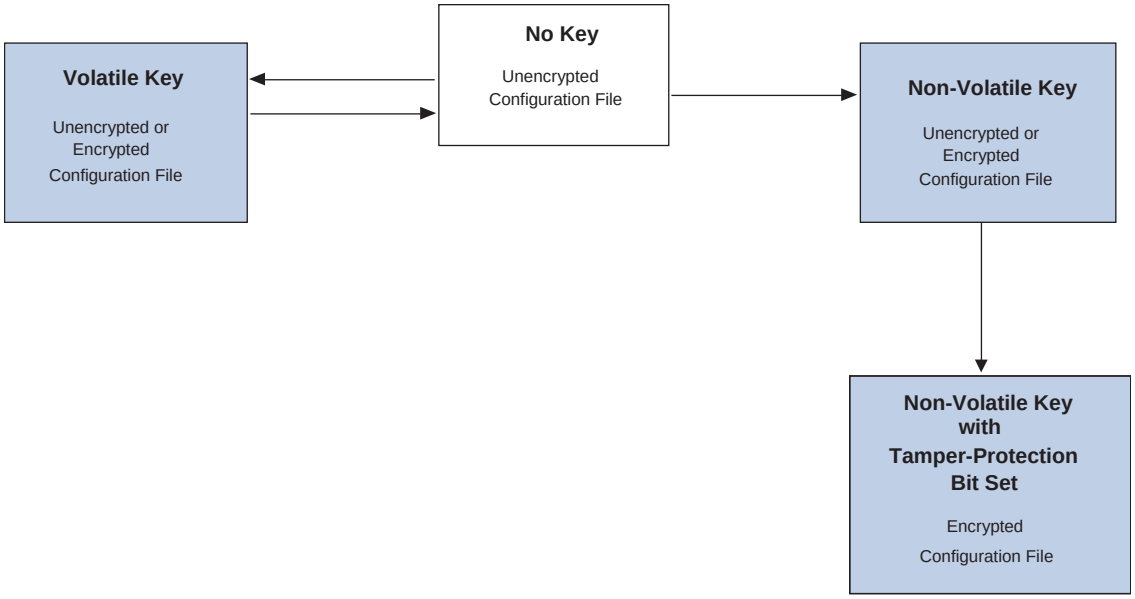


Table 10–20 lists the configuration modes allowed in each of the security modes.

Table 10–20. Allowed Configuration Modes for Various Security Modes (Part 1 of 2) *(Note 1)*

Security Mode	Configuration File	Allowed Configuration Modes
No key	Unencrypted	All configuration modes that do not engage the design security feature.
Secure with volatile key	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)
Board-level testing with volatile key	Unencrypted	All configuration modes that do not engage the design security feature.
Secure with non-volatile key	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)
Board-level testing with non-volatile key	Unencrypted	All configuration modes that do not engage the design security feature.

Table 10–20. Allowed Configuration Modes for Various Security Modes (Part 2 of 2) *(Note 1)*

Security Mode	Configuration File	Allowed Configuration Modes
Secure in tamper resistant mode using non-volatile key with tamper protection set	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)

Note to Table 10–20:

(1) There is no impact to the configuration time required when compared with unencrypted configuration modes except FPP with AES (and/or decompression), which requires a DCLK that is ×4 the data rate.

You can use the design security feature with other configuration features, such as compression and remote system upgrade features. When you use compression with the design security feature, the configuration file is first compressed and then encrypted using the Quartus II software. During configuration, the Stratix IV device first decrypts and then decompresses the configuration file.

Document Revision History

Table 10–21 shows the revision history for this chapter.

Table 10–21. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated Table 10–1 and Table 10–2. ■ Updated the "FPP Configuration Using a MAX II Device as an External Host","Fast Active Serial Configuration (Serial Configuration Devices)", "Device Configuration Pins", "Remote System Upgrades", "Remote System Upgrade Mode", "Estimating Active Serial Configuration Time", "Remote System Upgrade State Machine", and "User Watchdog Timer" sections. ■ Removed Table 10-4, Table 10-7, Table 10-8, and Table 10-25. ■ Minor text edits.	—
June 2009 v2.3	■ Updated the "VCCPD Pins", "FPP Configuration Using a MAX II Device as an External Host", "Estimating Active Serial Configuration Time", "Fast Active Serial Configuration (Serial Configuration Devices)", "Remote System Upgrades", "PS Configuration Using a MAX II Device as an External Host", and "PS Configuration Using a Download Cable" sections. ■ Updated Table 10–3, Table 10–13 and Table 10–2. ■ Added introductory sentences to improve search ability. ■ Removed the Conclusion section. ■ Minor text edits.	—
April 2009 v2.2	■ Updated Table 10–2.	—
March 2009 v2.1	■ Updated Table 10–1, Table 10–2, and Table 10–9. ■ Removed "Referenced Documents" section.	—

Table 10–21. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2008 v2.0	■ Updated "Fast Active Serial Configuration (Serial Configuration Devices)" and "JTAG Configuration" sections. ■ Updated Figure 10–4, Figure 10–5, Figure 10–6, and Figure 10–13. ■ Updated Table 10–2 and Table 10–13.	Medium update.
May 2008 v1.0	Initial release.	—

11. SEU Mitigation in Stratix IV Devices

SIV51011-3.0

This chapter describes how to use the error detection cyclical redundancy check (CRC) feature when a Stratix® IV device is in user mode and recovers from CRC errors. The purpose of the error detection CRC feature in the Stratix IV device is to detect a flip in any of the configuration random access memory (CRAM) bits in Stratix IV devices due to a soft error. With the error detection circuitry, you can continuously verify the integrity of the configuration CRAM bits.

In critical applications such as avionics, telecommunications, system control, and military applications, it is important to be able to do the following:

- Confirm that the configuration data stored in a Stratix IV device is correct
- Alert the system to the occurrence of a configuration error

The error detection feature is enhanced in the Stratix IV device family. Similar to Stratix III devices, the error detection and recovery time for single-event upset (SEU) in Stratix IV devices is reduced when compared with Stratix II devices.

For more information about test methodology for enhanced error detection in Stratix IV devices, refer to *AN 539: Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices*.

Dedicated circuitry is built into Stratix IV devices and consists of a CRC error detection feature that optionally checks for SEUs continuously and automatically.

For Stratix IV devices, the error detection CRC feature is provided in the Quartus® II software version 8.0 and onwards.

Using error detection CRC for the Stratix IV device family has no impact on fitting or performance of your device.

This chapter contains the following sections:

Error Detection Fundamentals

Error detection determines whether the data received is corrupted during transmission. To accomplish this, the transmitter uses a function to calculate a checksum value for the data and appends the checksum to the original data frame. The receiver uses the same calculation methodology to generate a checksum for the received data frame and compares the received checksum to the transmitted checksum. If the two checksum values are equal, the received data frame is correct and no data corruption occurred during transmission or storage.

The error detection CRC feature uses the same concept. When Stratix IV devices are configured successfully and are in user mode, the error detection CRC feature ensures the integrity of the configuration data.

There are two CRC error checks. One CRC error check always runs during configuration and a second optional CRC error check runs in the background in user mode. Both CRC error checks use the same CRC polynomial but different error detection implementations. For more information, refer to "Configuration Error Detection" on page 11–2 and "User Mode Error Detection" on page 11–2.

Configuration Error Detection

In configuration mode, a frame-based CRC is stored within the configuration data and contains the CRC value for each data frame.

During configuration, the Stratix IV device calculates the CRC value based on the frame of data that is received and compares it against the frame CRC value in the data stream. Configuration continues until either the device detects an error or configuration is completed.

In Stratix IV devices, the CRC value is calculated during the configuration stage. A parallel CRC engine generates 16 CRC check bits per frame and then stores them in CRAM. The CRAM chain used for storing the CRC check bits is 16 bits wide and its length is equal to the number of frames in the device.

User Mode Error Detection

Stratix IV devices have built-in error detection circuitry to detect data corruption by soft errors in the CRAM cells. This feature allows all CRAM contents to be read and verified to match a configuration-computed CRC value. Soft errors are changes in a CRAM bit state due to an ionizing particle.

The error detection capability continuously computes the CRC of the configured CRAM bits and compares it with the pre-calculated CRC. If the CRCs match, there is no error in the current configuration CRAM bits. The process of error detection continues until the device is reset (by setting `nCONFIG` low).

If you enable the **CRC error detection** option in the Quartus II software, after the device transitions into user mode, the error detection process is enabled. The internal 100 MHz configuration oscillator is divided down by a factor of two to 256 (at powers of two) to be used as the clock source during the error detection process. You must set the clock divide factor in the Quartus II software.

A single 16-bit CRC calculation is done on a per-frame basis. After it has finished the CRC calculation for a frame, the resulting 16-bit signature is `hex 0000` if there are no CRAM bit errors detected in a frame by the error detection circuitry and the output signal `CRC_ERROR` is `0`. If a CRAM bit error is detected by the circuitry within a frame in the device, the resulting signature is non-zero. This causes the CRC engine to start searching for the error bit location.

Error detection in Stratix IV devices calculates CRC check bits for each frame and pulls the `CRC_ERROR` pin high when it detects bit errors in the chip. Within a frame, it can detect all single-bit, double-bit, and three-bit errors. The probability of more than three CRAM bits being flipped by an SEU event is very low. In general, for all error patterns the probability of detection is 99.998%.

The CRC engine reports the bit location and determines the type of error for all single-bit errors and over 99.641% of double-adjacent errors. The probability of other error patterns is very low and report of the location of bit flips is not guaranteed by the CRC engine.

You can also read-out the error bit location through the JTAG and the core interface. Shift these bits out through either the `SHIFT_EDERROR_REG` JTAG instruction or the core interface, before the CRC detects the next error in another frame. If the next frame also has an error, you must shift these bits out within the amount of time of one frame CRC verification. You can choose to extend this time interval by slowing down the error detection clock frequency, but this slows down the error recovery time for the SEU event. For the minimum update interval for Stratix IV devices, refer to Table 11–6 on page 11–9. If these bits are not shifted out before the next error location is found, the previous error location and error message is overwritten by the new information. The CRC circuit continues to run, and if an error is detected, you must decide whether to complete a reconfiguration or to ignore the CRC error.

The error detection logic continues to calculate the `CRC_ERROR` and 16-bit signatures for the next frame of data regardless if any error has occurred in the current frame or not. You need to monitor these signals and take the appropriate actions if a soft error occurs.

The error detection circuitry in Stratix IV devices uses a 16-bit CRC-ANSI standard (16-bit polynomial) as the CRC generator.

The computed 16-bit CRC signature for each frame is stored in the registers within the core. The total storage register size is 16 (the number of bits per frame) × the number of frames.

The Stratix IV device error detection feature does not check memory blocks and I/O buffers. Thus, the `CRC_ERROR` signal might stay solid high or low depending on the error status of the previously checked CRAM frame. The I/O buffers are not verified during error detection because these bits use flipflops as storage elements that are more resistant to soft errors when compared with CRAM cells. The support parity bits of MLAB, M9K, and M144K are used to check the contents of the memory blocks for any errors. The M144K TriMatrix memory block has a built-in error correction code block that is able to check and correct the errors in the block.

For more information, refer to the *TriMatrix Embedded Memory Blocks in Stratix IV Devices* chapter.

A JTAG instruction, EDERROR_INJECT, is provided to test the capability of the error detection block. This instruction is able to change the content of the 21-bit JTAG fault injection register that is used for error injection in Stratix IV devices, enabling the testing of the error detection block.

You can only execute the EDERROR_INJECT JTAG instruction when the device is in user mode.

Table 11–1 lists the description of the EDERROR_INJECT JTAG instruction.

Table 11–1. EDERROR_INJECT JTAG Instruction

JTAG Instruction	Instruction Code	Description
EDERROR_INJECT	00 0001 0101	This instruction controls the 21-bit JTAG fault injection register, which is used for error injection.

You can create a Jam™ file (**.jam**) to automate the testing and verification process. This allows you to verify the CRC functionality in-system, on-the-fly, without having to reconfigure the device. You can then switch to the CRC circuit to check for real errors induced by an SEU.

You can introduce a single-error or double-errors adjacent to each other to the configuration memory. This provides an extra way to facilitate design verification and system fault tolerance characterization. Use the JTAG fault injection register with the EDERROR_INJECT instruction to flip the readback bits. The Stratix IV device is then forced into error test mode.

The content of the JTAG fault injection register is not loaded into the fault injection register during the processing of the last and first frame. It is only loaded at the end of this period.

You can only introduce error injection in the first data frame, but you can monitor the error information at any time. For more information about the JTAG fault injection register and fault injection register, refer to "Error Detection Registers" on page 11–6.

Table 11–2 lists how the fault injection register is implemented and describes error injection.

Table 11–2. Fault Injection Register

Bit	Bit[20..19]			Bit[18..8]	Bit[7..0]
Description	Error Type			Byte Location of the Injected Error	Error Byte Value
Content	Error Type *(1)*		Error injection type	Depicts the location of the injected error in the first data frame.	Depicts the location of the bit error and corresponds to the error injection type selection.
	Bit[20]	Bit[19]			
	0	1	Single-byte error injection		
	1	0	Double-adjacent byte error injection		
	0	0	No error injection		

Note to Table 11–2:

(1) Bit[20] and Bit[19] cannot both be set to 1 as this is not a valid selection. The error detection circuitry decodes this as no error injection.

After the test completes, Altera recommends that you reconfigure the device.

Automated Single-Event Upset Detection

Stratix IV devices offer on-chip circuitry for automated checking of SEU detection. Some applications that require the device to operate error-free in high-neutron flux environments require periodic checks to ensure continued data integrity. The error detection CRC feature ensures data reliability and is one of the best options for mitigating SEU.

You can implement the error detection CRC feature with existing circuitry in Stratix IV devices, eliminating the need for external logic. The CRC_ERROR pin reports a soft error when the configuration CRAM data is corrupted. You must decide whether to reconfigure the device or to ignore the error.

Error Detection Pin Description

Depending on the type of error detection feature you choose, you must use different error detection pins to monitor the data during user mode.

CRC_ERROR Pin

Table 11–3 describes the CRC_ERROR pin.

Table 11–3. CRC_ERROR Pin Description

Pin Name	Pin Type	Description
CRC_ERROR	I/O, output open-drain	Active-high signal indicates that the error detection circuit has detected errors in the configuration CRAM bits. This pin is optional and is used when the error detection CRC circuit is enabled. When the error detection CRC circuit is disabled, it is a user I/O pin. The CRC error output, when using the `stratixiv_crcblock` WYSIWYG function, is a dedicated path to the CRC_ERROR pin. To use the CRC_ERROR pin, you can either tie this pin to V_{CCPGM} through a 10k Ω resistor or, depending on the input voltage specification of the system receiving the signal, you can tie this pin to a different pull-up voltage.

The WYSIWYG function performs optimization on the Verilog Quartus Mapping (VQM) netlist within the Quartus II software.

For more information about the `stratixiv_crcblock` WYSIWYG function, refer to the *AN 539: Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices*.

For more information about the CRC_ERROR pin for Stratix IV devices, refer to *Device Pin-Outs* on the Altera website.

Error Detection Block

You can enable the Stratix IV device error detection block in the Quartus II software (refer to "Software Support" on page 11–10). This block contains the logic necessary to calculate the 16-bit CRC signature for the configuration CRAM bits in the device.

The CRC circuit continues running even if an error occurs. When a soft error occurs, the device sets the CRC_ERROR pin high. Two types of CRC detection checks the configuration bits:

- CRAM error checking ability (16-bit CRC), which occurs during user mode to be used by the CRC_ERROR pin.
  - For each frame of data, the pre-calculated 16-bit CRC enters the CRC circuit at the end of the frame data and determines whether there is an error or not.
  - If an error occurs, the search engine starts to find the location of the error.
  - The error messages are shifted out through the JTAG instruction or core interface logics while the error detection block continues running.
  - The JTAG interface reads out the 16-bit CRC result for the first frame and also shifts the 16-bit CRC bits to the 16-bit CRC storage registers for test purposes.
  - Single error, double errors, or double-errors adjacent to each other are deliberately introduced to configuration memory for testing and design verification.
- 16-bit CRC that is embedded in every configuration data frame.
  - During configuration, after a frame of data is loaded into the Stratix IV device, the pre-computed CRC is shifted into the CRC circuitry.
  - At the same time, the CRC value for the data frame shifted-in is calculated. If the pre-computed CRC and calculated CRC values do not match, nSTATUS is set low. Every data frame has a 16-bit CRC; therefore, there are many 16-bit CRC values for the whole configuration bitstream. Every device has different lengths of configuration data frame.

The "Error Detection Block" section describes the 16-bit CRC only when the device is in user mode.

Error Detection Registers

There is one set of 16-bit registers in the error detection circuitry that stores the computed CRC signature. A non-zero value on the syndrome register causes the CRC_ERROR pin to be set high.

Figure 11–1 shows the error detection circuitry, syndrome registers, and error injection block.

Figure 11–1. Error Detection Block Diagram

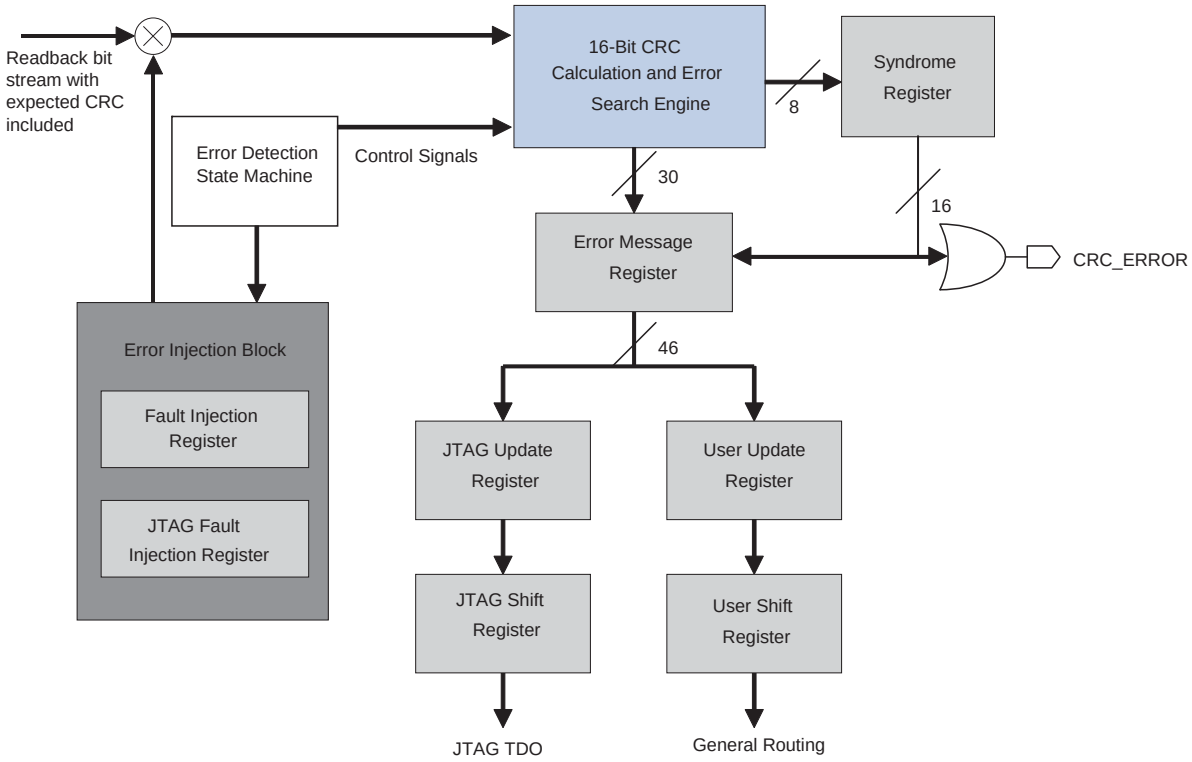


Table 11–4 defines the registers shown in Figure 11–1.

Table 11–4. Error Detection Registers (Part 1 of 2)

Register	Description
Syndrome Register	This register contains the CRC signature of the current frame through the error detection verification cycle. The CRC_ERROR signal is derived from the contents of this register.
Error Message Register	This 46-bit register contains information on the error type, location of the error, and the actual syndrome. The types of errors and location reported are single- and double-adjacent bit errors. The location bits for other types of errors are not identified by the error message register. The content of the register can be shifted out through the SHIFT_EDERROR_REG JTAG instruction or to the core through the core interface.
JTAG Update Register	This register is automatically updated with the contents of the error message register one cycle after the 46-bit register content is validated. It includes a clock enable that must be asserted prior to being sampled into the JTAG shift register. This requirement ensures that the JTAG update register is not being written into by the contents of the error message register at exactly the same time that the JTAG shift register is reading its contents.

Table 11–4. Error Detection Registers (Part 2 of 2)

Register	Description
User Update Register	This register is automatically updated with the contents of the Error Message Register, one cycle after the 46-bit register content is validated. It includes a clock enable that must be asserted prior to being sampled into the User Shift Register. This requirement ensures that the User Update Register is not being written into by the contents of the Error Message Register at exactly the same time that the User Shift Register is reading its contents.
JTAG Shift Register	This register is accessible by the JTAG interface and allows the contents of the JTAG Update Register to be sampled and read by the JTAG instruction SHIFT_EDERROR_REG.
User Shift Register	This register is accessible by the core logic and allows the contents of the User Update Register to be sampled and read by user logic.
JTAG Fault Injection Register	This 21-bit register is fully controlled by the JTAG instruction EDERROR_INJECT. This register holds the information of the error injection that you want in the bitstream.
Fault Injection Register	The content of the JTAG Fault Injection Register is loaded into this 21-bit register when it is being updated.

Error Detection Timing

When you enable the CRC feature through the Quartus II software, the device automatically activates the CRC process upon entering user mode, after configuration, and after initialization is complete.

If an error is detected within a frame, CRC_ERROR is driven high at the end of the error location search, after the error message register is updated. At the end of this cycle, the CRC_ERROR pin is pulled low for a minimum of 32 clock cycles. If the next frame contains an error, the CRC_ERROR is driven high again after the error message register is overwritten by the new value. You can start to unload the error message on each rising edge of the CRC_ERROR pin. The error detection runs until the device is reset.

The error detection circuitry runs off an internal configuration oscillator with a divisor that sets the maximum frequency. Table 11–5 lists the minimum and maximum error detection frequencies based on the best performance of the internal configuration oscillator.

Table 11–5. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (n)
Stratix IV	100 MHz / 2^n	50 MHz	390 kHz	1, 2, 3, 4, 5, 6, 7, 8

You can set a lower clock frequency by specifying a division factor in the Quartus II software (refer to "Software Support" on page 11–10). The divisor is a power of two, in which n is between 1 and 8. The divisor ranges from 2 through 256. Refer to Equation 11–1.

Equation 11–1.

$$\text{error detection frequency} = \frac{100\text{MHz}}{2^n}$$

The error detection frequency reflects the frequency of the error detection process for a frame because the CRC calculation in the Stratix IV device is done on a per-frame basis.

You must monitor the error message to avoid missing information in the error message register. The error message register is updated whenever an error or errors occur. The minimum interval time between each update for the error message register depends on the device and the error detection clock frequency.

Table 11–6 lists the estimated minimum interval time between each update for the error message register for Stratix IV devices.

Table 11–6. Minimum Update Interval for Error Message Register *(Note 1)*

Device	Timing Interval (µs)
EP4SGX70	15
EP4SGX110	15
EP4SGX180	21
EP4SGX230	21
EP4SGX290	23
EP4SGX360	23
EP4SGX530	28
EP4SE230	21
EP4SE360	23
EP4SE530	28
EP4SE820	35
EP4S40G2	21
EP4S40G5	28
EP4S100G2	21
EP4S100G3	28
EP4S100G4	28
EP4S100G5	28

Note to Table 11–6:

(1) These timing numbers are preliminary.

CRC calculation time for the error detection circuitry to check from the first until the last frame depends on the device and the error detection clock frequency.

Table 11–7 lists the estimated time for each CRC calculation with minimum and maximum clock frequencies for Stratix IV devices. The minimum CRC calculation time is calculated by using the maximum error detection frequency with a divisor factor of one, while the maximum CRC calculation time is calculated by using the minimum error detection frequency with a divisor factor of eight.

Table 11–7. CRC Calculation Time *(Note 1)*

Device	Minimum Time (ms)	Maximum Time (s)
EP4SGX70	111	30.90
EP4SGX110	111	30.90
EP4SGX180	225	62.44
EP4SGX230	225	62.44
EP4SGX290	296	82.05
EP4SGX360	296	82.05
EP4SGX530	398	110.38
EP4SE230	225	62.44
EP4SE360	296	82.05
EP4SE530	398	110.38
EP4SE820	577	160.00
EP4S40G2	225	62.44
EP4S40G5	398	110.38
EP4S100G2	225	62.44
EP4S100G3	398	110.38
EP4S100G4	398	110.38
EP4S100G5	398	110.38

Note to Table 11–7:

(1) These timing numbers are preliminary.

Software Support

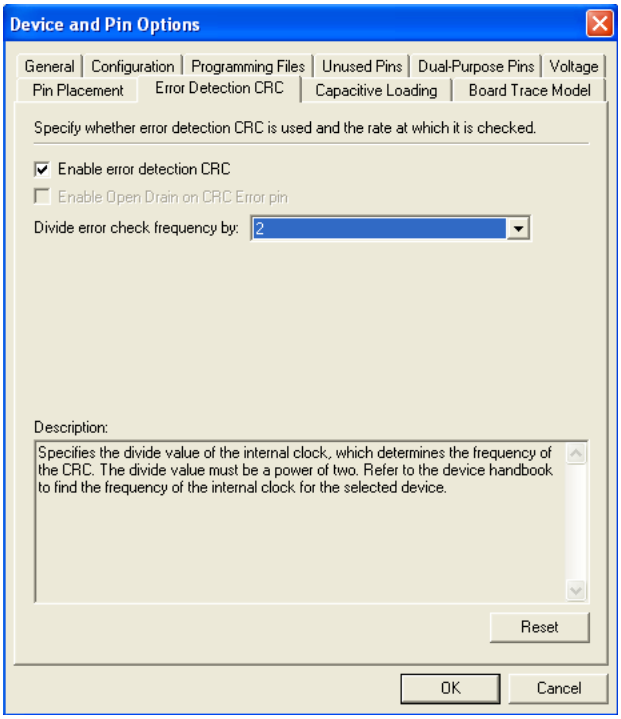
The Quartus II software version 8.0 and onwards supports the error detection CRC feature for Stratix IV devices. Enabling this feature generates the CRC_ERROR output to the optional dual purpose CRC_ERROR pin.

The error detection CRC feature is controlled by the **Device and Pin Options** dialog box in the Quartus II software.

To enable the error detection feature using CRC, perform the following steps:

1. Open the Quartus II software and load a project using a Stratix IV device.
2. On the Assignments menu, click **Settings**. The **Settings** dialog box is shown.
3. In the **Category** list, select **Device**. The **Device** page is shown.
4. Click **Device and Pin Options**. The **Device and Pin Options** dialog box is shown (refer to Figure 11–2).
5. In the **Device and Pin Options** dialog box, click the **Error Detection CRC** tab.
6. Turn on **Enable error detection CRC** (Figure 11–2).

Figure 11–2. Enabling the Error Detection CRC Feature in the Quartus II Software



7. In the **Divide error check frequency by** pull-down list, enter a valid divisor as listed in Table 11–5 on page 11–8.

The divide value divides the frequency of the configuration oscillator output clock that clocks the CRC circuitry.

8. Click **OK**.

Recovering From CRC Errors

The system that the Stratix IV device resides in must control the device reconfiguration. After detecting an error on the CRC_ERROR pin, strobing the nCONFIG signal low directs the system to perform the reconfiguration at a time when it is safe for the system to reconfigure the device.

When the data bit is rewritten with the correct value by reconfiguring the device, the device functions correctly.

While soft errors are uncommon in Altera devices, certain high-reliability applications require a design to account for these errors.

Document Revision History

Table 11–8 lists the revision history for this chapter.

Table 11–8. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated Table 11–3, Table 11–5, Table 11–6, and Table 11–7. ■ Updated the "CRC_ERROR Pin" section. ■ Minor text edits.	—
June 2009 v2.3	■ Added an introductory paragraph to increase search ability. ■ Removed the Conclusion section. ■ Minor text edits.	—
April 2009 v2.2	■ Updated Table 11–6 and Table 11–7.	—
March 2009 v2.1	■ Updated "Error Detection Timing" section. ■ Updated Table 11–6. ■ Added Table 11–7. ■ Removed "Critical Error Detection", "Critical Error Pin", and "Referenced Documents" sections.	—
November 2008 v2.0	Minor text edits.	—
May 2008 | v1.0	Initial Release.	—

12. JTAG Boundary-Scan Testing in Stratix IV Devices

SIV51012-3.0

The IEEE Std. 1149.1 boundary-scan test (BST) circuitry available in Stratix® IV devices provides a cost-effective and efficient way to test systems that contain devices with tight lead spacing. Circuit boards with Altera and other IEEE Std. 1149.1-compliant devices can use EXTEST, SAMPLE/PRELOAD, and BYPASS modes to create serial patterns that internally test the pin connections between devices and check device operation.

This chapter describes how to use the IEEE Std. 1149.1 BST circuitry in Stratix IV devices. The features are similar to Stratix III devices, unless stated otherwise in this document.

This chapter contains the following sections:

BST Architecture

A device operating in IEEE Std. 1149.1 BST mode uses four required pins, TDI, TDO, TMS, TCK, and one optional pin, TRST. The TCK pin has an internal weak pull-down resistor, while the TDI, TMS, and TRST pins have internal weak pull-up resistors. The TDO output pin and all the JTAG input pins are powered by the 2.5-V/3.0-V V_{CCPD} supply of I/O bank 1A. All user I/O pins are tri-stated during JTAG configuration.

For more information about the description and functionality of all JTAG pins, registers used by the IEEE Std. 1149.1 BST circuitry, and the test access port (TAP) controller, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

BST Operation Control

Table 12–1 lists the boundary-scan register length for Stratix IV devices.

Table 12–1. Stratix IV Devices Boundary-Scan Register Length (Part 1 of 2)

Device	Boundary-Scan Register Length
EP4SGX70	1506
EP4SGX110	1506
EP4SGX180	2274
EP4SGX230	2274
EP4SGX290 *(1)*	2682
EP4SGX360 *(1)*	2682

Table 12–1. Stratix IV Devices Boundary-Scan Register Length (Part 2 of 2)

Device	Boundary-Scan Register Length
EP4SGX530	2970
EP4SE230	2274
EP4SE360	2682
EP4SE530	2970
EP4SE820	3402
EP4S40G2	2274
EP4S40G5	2970
EP4S100G2	2274
EP4S100G3	2970
EP4S100G4	2970
EP4S100G5	2970

Note to Table 12–1:

(1) For the F1932 package of EP4SGX290 and EP4SGX360 devices, the boundary-scan register length is 2970.

Table 12–2 lists the IDCODE JTAG instruction information for Stratix IV devices.

Table 12–2. Stratix IV Devices IDCODE JTAG Instruction Information (Part 1 of 2)

Device	IDCODE (32 Bits) *(1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) *(2)*
EP4SGX70	0000	0010 0100 0010 0000	000 0110 1110	1
EP4SGX110	0000	0010 0100 0000 0000	000 0110 1110	1
EP4SGX180	0000	0010 0100 0010 0001	000 0110 1110	1
EP4SGX230	0000	0010 0100 0000 1001	000 0110 1110	1
EP4SGX290 *(3)*	0000	0010 0100 0010 0010	000 0110 1110	1
EP4SGX290 *(4)*	0000	0010 0100 0100 0011	000 0110 1110	1
EP4SGX360 *(3)*	0000	0010 0100 0000 0010	000 0110 1110	1
EP4SGX360 *(4)*	0000	0010 0100 1000 0011	000 0110 1110	1
EP4SGX530	0000	0010 0100 0000 0011	000 0110 1110	1
EP4SE230	0000	0010 0100 0001 0001	000 0110 1110	1
EP4SE360	0000	0010 0100 0001 0010	000 0110 1110	1
EP4SE530	0000	0010 0100 0001 0011	000 0110 1110	1
EP4SE820	0000	0010 0100 0000 0100	000 0110 1110	1
EP4S40G2 *(5)*	0000	0010 0100 0100 0001	000 0110 1110	1
EP4S40G5 *(6)*	0000	0010 0100 0010 0011	000 0110 1110	1
EP4S100G2 *(5)*	0000	0010 0100 0100 0001	000 0110 1110	1
EP4S100G3	0000	0010 0100 1010 0011	000 0110 1110	1
EP4S100G4	0000	0010 0100 0110 0011	000 0110 1110	1

Table 12–2. Stratix IV Devices IDCODE JTAG Instruction Information (Part 2 of 2)

Device	IDCODE (32 Bits) *(1)*			
	Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) *(2)*
EP4S100G5 *(6)*	0000	0010 0100 0010 0011	000 0110 1110	1

Notes to Table 12–2:

(1) The MSB is on the left.
(2) The LSB of the IDCODE JTAG instruction is always 1.
(3) The IDCODE JTAG instruction is applicable for all packages except F1932.
(4) The IDCODE JTAG instruction is applicable for package F1932 only.
(5) For the ES1 device, the IDCODE JTAG instruction is the same as the IDCODE JTAG instruction of EP4SGX230.
(6) For the ES1 device, the IDCODE JTAG instruction is the same as the IDCODE JTAG instruction of EP4SGX530.

To correctly read the IDCODE JTAG instruction, you must issue the IDCODE JTAG instruction after initialization, which is signaled by nSTATUS going high.

For more information about the following topics, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*:

- JTAG instruction codes with descriptions
- TAP controller state-machine
- Timing requirements for IEEE Std. 1149.1 signals
- Instruction mode
- Mandatory JTAG instructions (SAMPLE/PRELOAD, EXTEST, and BYPASS)
- Optional JTAG instructions (IDCODE, USERCODE, CLAMP, and HIGHZ)

I/O Voltage Support in a JTAG Chain

The JTAG chain supports several devices. However, you must use caution if the chain contains devices that have different V_{CCIO} levels.

For more information, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

BST Circuitry

The IEEE Std. 1149.1 BST circuitry is enabled upon device power-up. You can perform BST on Stratix IV devices before, during, and after configuration. Stratix IV devices support BYPASS, IDCODE, and SAMPLE JTAG instructions during configuration without interrupting configuration. To send all other JTAG instructions, you must interrupt configuration using the CONFIG_IO JTAG instruction.

For more information, refer to *AN 39: IEEE Std. 1149.1 (JTAG) Boundary-Scan Testing in Altera Devices*.

For more information about using the CONFIG_IO JTAG instruction for dynamic I/O buffer configuration, considerations when performing BST for configured devices, and JTAG pin connections to mask-out the BST circuitry, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*.

For more information about using the IEEE Std.1149.1 circuitry for device configuration, refer to the *Configuration, Design Security, Remote System Upgrades* chapter.

If you must perform BST for configured devices, you must use the Quartus II software version 8.1 and onwards to generate the design-specific boundary-scan description language (BSDL) files. For the procedure to generate post-configured BSDL files using the Quartus II software, refer to the *BSDL Files Generation in Quartus II* on the Altera website.

BSDL Support

BSDL, a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.1 BST-capable device that can be tested.

For more information about BSDL files for IEEE Std. 1149.1-compliant Stratix IV devices, refer to the *Stratix IV BSDL Files* on the Altera website.

BSDL files for IEEE std. 1149.1-compliant Stratix IV devices can also be generated using the Quartus II software version 8.1 and onwards. For more information about the procedure to generate BSDL files using the Quartus II software, refer to the *BSDL Files Generation in Quartus II* on the Altera website.

Document Revision History

Table 12–3 shows the revision history for this chapter.

Table 12–3. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009, v3.0	■ Updated Table 12–1 and Table 12–2. ■ Minor text edits.	—
June 2009, v2.3	■ Added an introductory paragraph to increase search ability. ■ Removed the Conclusion section. ■ Minor text edits.	—
April 2009 v2.2	■ Updated Table 12–1.	—
March 2009 v2.1	■ Updated Table 12–1 and Table 12–2. ■ Removed “Referenced Documents” section.	—
November 2008 v2.0	Minor text edits.	—
May 2008 v1.0	Initial Release.	—

13. Power Management in Stratix IV Devices

SIV51013-3.0

This chapter describes power management in Stratix® IV devices. Stratix IV devices offer programmable power technology options for low-power operation. You can use these features, along with speed grade choices, in different permutations to give the best power and performance combination. For thermal management, use the Stratix IV internal temperature sensing device (TSD) with built-in analog-to-digital converter (ADC) circuitry or external TSD with an external temperature sensor to easily incorporate this feature in your designs. Being able to monitor the junction temperature of the device at any time also allows you the ability to control air flow to the device and save power for the whole system.

Overview

This chapter describes how Stratix IV FPGAs deliver a breakthrough level of system bandwidth and power efficiency for high-end applications, allowing you to innovate without compromise. Advanced power management techniques are used in Stratix IV devices to enable both density and performance increases, while simultaneously reducing power dissipation.

The total power of an FPGA includes static power and dynamic power. Static power is the power consumed by the FPGA when it is configured but no clocks are operating. Dynamic power is comprised of switching power when the device is configured and running. Dynamic power is calculated using the equation shown in Equation 13–1.

Equation 13–1. Dynamic Power Equation

$$P = \frac{1}{2} CV^2 \times \text{frequency}$$

Note to Equation 13–1:

(1) P=power; C=load capacitance; V=supply voltage level.

Equation 13–1 shows that frequency is design-dependent. However, voltage can be varied to lower dynamic power consumption by the square value of the voltage difference. Stratix IV devices minimize static and dynamic power with advanced process optimizations and programmable power technology. These technologies enable Stratix IV designs to optimally meet design-specific performance requirements with the lowest possible power.

The Quartus® II software optimizes all designs with Stratix IV power technology to ensure performance is met at the lowest power consumption. This automatic process allows you to concentrate on the functionality of the design instead of the power consumption of the design.

Power consumption also affects thermal management. Stratix IV devices offer a TSD feature, which can self-monitor the device junction temperature and can be used with external circuitry for activities such as controlling air flow to the Stratix IV FPGA.

This chapter contains the following sections:

Stratix IV Power Technology

The following sections describe Stratix IV programmable power technology.

Programmable Power Technology

Stratix IV devices offer the ability to configure portions of the core, called tiles, for high-speed or low-power mode of operation performed by the Quartus II software without user intervention. Setting a tile to high-speed or low-power mode is accomplished with on-chip circuitry and does not require extra power supplies brought into the Stratix IV device. In a design compilation, the Quartus II software determines whether a tile must be in high-speed or low-power mode based on the timing constraints of the design.

For more information about how the Quartus II software uses programmable power technology when compiling a design, refer to *AN 514: Power Optimization in Stratix IV Devices*.

A Stratix IV tile can consist of the following:

- Memory logic array block (MLAB)/logic array block (LAB) pairs with routing to the pair
- MLAB/LAB pairs with routing to the pair and to adjacent digital signal processing (DSP)/memory block routing
- TriMatrix memory blocks
- DSP blocks

All blocks and routing associated with the tile share the same setting of either high speed or low power. By default, tiles that include DSP blocks or memory blocks are set to high-speed mode for optimum performance. Unused DSP blocks and memory blocks are set to low-power mode to minimize static power. Clock networks do not support programmable power technology.

With programmable power technology, faster speed grade FPGAs may require less power because there are fewer high-speed MLAB and LAB pairs, when compared with slower speed grade FPGAs. The slower speed grade device may need to use more high-speed MLAB and LAB pairs to meet performance requirements, while the faster speed grade device can meet performance requirements with MLAB and LAB pairs in low-power mode.

The Quartus II software can set unused device resources in the design to low-power mode to reduce static and dynamic power. The Quartus II software can set the following resources to low power when they are not used in the design:

- LABs and MLABs
- TriMatrix memory blocks
- DSP blocks

If the PLL is instantiated in the design, asserting the `areset` pin high keeps the PLL in low-power mode.

Table 13–1 lists the available Stratix IV programmable power capabilities. Speed grade considerations can add to the permutations to give you flexibility in designing your system.

Table 13–1. Stratix IV Programmable Power Capabilities

Feature	Programmable Power Technology
LAB	Yes
Routing	Yes
Memory Blocks	Fixed setting *(1)*
DSP Blocks	Fixed setting *(1)*
Global Clock Networks	No

Note to Table 13–1:

(1) Tiles with DSP blocks and memory blocks that are used in the design are always set to high-speed mode. By default, unused DSP blocks and memory blocks are set to low-power mode.

Stratix IV External Power Supply Requirements

This section describes the different external power supplies needed to power Stratix IV devices. You can supply some of the power supply pins with the same external power supply, provided they need the same voltage level.

For power supply pin connection guidelines and power regulator sharing, refer to the *Stratix IV GX Device Family Pin Connection Guidelines*.

For each Altera recommended power supply's operating conditions, refer to the *DC and Switching Characteristics* chapter.

Temperature Sensing Diode

The Stratix IV TSD uses the characteristics of a PN junction diode to determine die temperature. Knowing the junction temperature is crucial for thermal management. Historically, junction temperature is calculated using ambient or case temperature, junction-to-ambient (ja) or junction to-case (jc) thermal resistance, and device power consumption. Stratix IV devices can either monitor its die temperature with the internal TSD with built-in ADC circuitry or the external TSD with an external temperature sensor. This allows you to control the air flow to the device.

You can use the Stratix IV internal TSD in two different modes of operations—power-up mode and user mode. For power-up mode, the internal TSD reads the die's temperature during configuration if the ALTTEMP_SENSE megafunction is enabled in your design. The ALTTEMP_SENSE megafunction allows temperature sensing during device user mode by asserting the `clken` signal to the internal TSD circuitry. To reduce device static power, you can disable the internal TSD with built-in ADC circuitry when not in use.

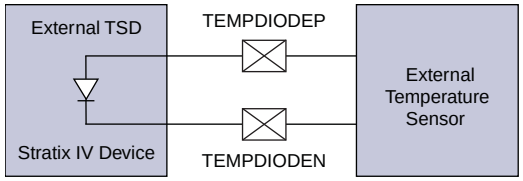
For more information about using the ALTTEMP_SENSE megafunction, refer to the *ALTTEMP_SENSE Megafunction User Guide*.

The external temperature sensor steers bias current through the Stratix IV external TSD, measuring forward voltage and converting this reading to temperature in the form of an 8-bit signed number (7 bits plus sign). The 8-bit output represents the junction temperature of the Stratix IV device and can be used for intelligent power management.

External Pin Connections

Stratix IV external TSD requires two pins for voltage reference. Figure 13–1 shows how you can connect the external TSD with an external temperature sensor device. As an example, external temperature sensing devices, such as MAX1619, MAX1617A, MAX6627, and ADT 7411, can be connected to the two external TSD pins for temperature reading.

Figure 13–1. Stratix IV TSD External Pin Connections



For more information about the external TSD specification, refer to the *DC and Switching Characteristics* chapter.

The TSD is a very sensitive circuit that can be influenced by noise coupled from other traces on the board and possibly within the device package itself, depending on your device usage. The interfacing device registers' temperature is based on millivolts (mV) of difference, as seen at the external TSD pins. Switching the I/O near the TSD pins can affect the temperature reading. Altera recommends taking temperature readings during periods of inactivity in the device or use the internal TSD with built-in ADC circuitry.

The following are board connection guidelines for the TSD external pin connections:

- The maximum trace lengths for the TEMPDIODEP/TEMPDIODEN traces must be less than eight inches.
- Route both traces in parallel and place them close to each other with grounded guard tracks on each side.
- Altera recommends 10-mils width and space for both traces.

- Route traces through a minimum number of vias and crossunders to minimize the thermocouple effects.
- Ensure that the number of vias are the same on both traces.
- Ensure both traces are approximately the same length.
- Avoid coupling with toggling signals (for example, clocks and I/O) by having the GND plane between the diode traces and the high frequency signals.
- For high-frequency noise filtering, place an external capacitor (close to the external chip) between the TEMPDIODEP/TEMPDIODEN trace.
- For Maxim devices, use an external capacitor between 2200 pF to 3300 pF.
- Place a 0.1 uF bypass capacitor close to the external device.

For device specification and connection guidelines, refer to the external temperature sensor device data sheet from the device manufacturer.

Document Revision History

Table 13–2 shows the revision history for this chapter.

Table 13–2. Document Revision History

Date and Document Version	Changes Made	Summary of Changes
November 2009 v3.0	■ Updated the "Temperature Sensing Diode" and "External Pin Connections" sections. ■ Updated Equation 13–1. ■ Removed Table 13-2: Stratix IV External Power Supply Pins. ■ Minor text edits.	—
June 2009 v2.2	■ Updated the "External Pin Connections" section. ■ Added an introductory paragraph to increase search ability. ■ Removed the Conclusion section.	—
March 2009 v2.1	■ Updated "Temperature Sensing Diode" and "External Pin Connections" sections. ■ Updated Figure 13–1. ■ Removed "Referenced Documents" section.	—
November 2008 v2.0	Minor text edits.	—
May 2008 v1.0	Initial Release.	—

Looking for pricing, stock, or lifecycle information?

Click below to explore more details on WIN SOURCE:

- View EP4SGX290KF40C3NAB on WIN SOURCE
- Intel Information

Optimize Your Supply Chain with WIN SOURCE Solutions

- Global Sourcing Solution
- Obsolete Management
- Cost Control Management
- Shortage Management
- Alternative Solution
- Excess Inventory Management