Stratix V Device Handbook

Volume 1: Overview and Datasheet

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

SV5V3-1.3
11.0

Contents

Additional Information

Chapter Revision Dates

The chapters in this document, *Stratix V 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 V Device Family Overview
Revised: *June 2011*
Part Number: *SV51001-1.8*

Chapter 2. DC and Switching Characteristics for Stratix V Devices
Revised: *May 2011*
Part Number: *SV53001-2.0*

1. Stratix V Device Family Overview

SV51001-1.8

This chapter provides an overview of the Stratix® V devices and their features. Many of these devices and features are enabled in the Quartus® II software version 11.0. The remaining devices and features will be enabled in future versions of the Quartus II software.

To find out more about the upcoming Stratix V devices and features, refer to the *Stratix V Upcoming Device Features* document.

Altera's 28-nm Stratix V FPGAs include innovations such as an enhanced core architecture, integrated transceivers up to 28 Gbps, and a unique array of integrated hard intellectual property (IP) blocks. With these innovations, Stratix V FPGAs deliver a new class of application-targeted devices optimized for:

- Bandwidth-centric applications and protocols, including PCI Express® (PCIe®) Gen3
- Data-intensive applications for 40G/100G and beyond
- High-performance, high-precision digital signal processing (DSP) applications

Stratix V devices are available in four variants (GT, GX, GS, and E), each targeted for a different set of applications. For higher volume production, you can prototype with Stratix V FPGAs and use the low-risk, low-cost path to HardCopy® V ASICs.

Stratix V Family Variants

Stratix V GT devices, with both 28-Gbps and 12.5-Gbps transceivers, are optimized for applications that require ultra-high bandwidth and performance in areas such as 40G/100G/400G optical communications systems and optical test systems.

Stratix V GX devices offer up to 66 integrated 14.1-Gbps transceivers supporting backplanes and optical modules. These devices are optimized for high-performance, high-bandwidth applications such as 40G/100G optical transport, packet processing, and traffic management found in wireline, military communications, and network test equipment markets.

Stratix V GS devices have an abundance of variable precision DSP blocks, supporting up to 4,096 18×18 or 2,048 27×27 multipliers. In addition, Stratix V GS devices offer integrated 14.1-Gbps transceivers, which support backplanes and optical modules. These devices are optimized for transceiver-based DSP-centric applications found in wireline, military, broadcast, and high-performance computing markets.

Stratix V E devices offer the highest logic density within the Stratix V family with nearly one million logic elements (LEs) in the largest device. These devices are optimized for applications such as ASIC and system emulation, diagnostic imaging, and instrumentation.

Common to all Stratix V family variants are a rich set of high-performance building blocks, including a redesigned adaptive logic module (ALM), 20 Kbit (M20K) embedded memory blocks, variable precision DSP blocks, and fractional phase-locked loops (PLLs). All of these building blocks are interconnected by Altera's superior multi-track routing architecture and comprehensive fabric clocking network.

Also common to Stratix V devices is the new Embedded HardCopy Block, which is a customizable hard IP block that leverages Altera's unique HardCopy ASIC capabilities. Use the Embedded HardCopy Block for hardening standard or logic-intensive functions, such as interface protocols, application-specific functions, and proprietary custom IP. Incorporating hard IP into the Embedded HardCopy Block frees up valuable core logic resources and reduces overall system power and cost. The Embedded HardCopy Blocks in Stratix M20K Memory Blocks devices include a hard IP instantiation of PCIe Gen 3/2/1 and 40/100GbE.

Stratix V Features Summary

- Technology
  - 28-nm TSMC process technology
  - 0.85-V core voltage
- Low-power serial transceivers
  - 28-Gbps transceivers on Stratix V GT devices
  - Electronic dispersion compensation (EDC) for XFP, SFP+, QSFP, CFP optical module support
  - Adaptive linear and decision feedback equalization
  - 600 Mbps to 14.1 Gbps backplane capability
  - Transmit pre-emphasis and de-emphasis
  - Dynamic reconfiguration of individual channels
  - On-chip instrumentation (EyeQ non-intrusive data eye monitoring)
- General purpose I/Os
  - 1.4-Gbps LVDS
  - 1,066-MHz/1,600-Mbps external memory interface
  - On-chip termination (OCT)
  - 1.2-V to 3.3-V interfacing for all Stratix V devices
- Embedded HardCopy Block
  - PCIe Gen 3/2/1 complete protocol stack, ×1/×2/×4/×8 end point and root port
  - 40G/100G Ethernet physical coding sublayer (PCS)
- Embedded transceiver hard IP
  - Interlaken PCS
  - Gigabit Ethernet (GbE) and XAUI PCS
  - 10G Ethernet PCS
  - Serial RapidIO (SRIO) PCS
  - Common Public Radio Interface (CPRI) PCS
  - Gigabit Passive Optical Networking (GPON) PCS
- Power Management
  - Programmable Power Technology
  - Quartus II integrated PowerPlay Power Analysis
- High-performance core fabric
  - Enhanced ALM with four registers
  - Improved routing architecture reduces congestion and improves compile times
- Embedded memory blocks
  - M20K: 20-Kbit with hard error correction code (ECC)
  - MLAB: 640-bit
- Variable precision DSP blocks
  - Up to 500 MHz performance
  - Natively support signal processing with precision ranging from 9×9 up to 54×54
  - New native 27×27 multiply mode
  - 64-bit accumulator and cascade for systolic finite impulse responses (FIRs)
  - Embedded internal coefficient memory
  - Pre-adder/subtractor improves efficiency
  - Increased number of outputs allows more independent multipliers
- Fractional PLLs
  - Fractional mode with third-order delta-sigma modulation
  - Integer mode
  - Precision clock synthesis, clock delay compensation, and zero delay buffering
- Clock networks
  - 717-MHz fabric clocking
  - Global, quadrant, and peripheral clock networks
  - Unused clock networks can be powered down to reduce dynamic power
- Device Configuration
  - Serial and parallel flash interface
  - Enhanced advanced encryption standard (AES) design security features
  - Tamper protection
  - Partial and dynamic reconfiguration
  - Configuration via Protocol (CvP)
- High-performance packaging
  - Multiple device densities with identical package footprints enables seamless migration between different FPGA densities
  - FBGA packaging with on-package decoupling capacitors
  - Lead and RoHS-compliant lead-free options
- HardCopy V migration

Stratix V Family Plan

Table 1–1 lists the Stratix V GT device features.

Table 1–1. Stratix V GT Device Features

Features	5SGTC5	5SGTC7
Logic Elements (K)	425	622
Registers (K)	642	939
28/12.5-Gbps Transceivers	4/32	4/32
PCIe hard IP Blocks	1	1
Fractional PLLs	24	24
M20K Memory Blocks	2,304	2,560
M20K Memory (MBits)	45	50
Variable Precision Multipliers (18×18)	512	512
Variable Precision Multipliers (27×27)	256	256
DDR3 SDRAM ×72 DIMM Interfaces	4	4
40G/100G PCS hard IP Blocks	Yes	Yes
User I/Os, Full-Duplex LVDS, 28/14.1-Gbps Transceivers		
Package *(1)*, *(2)*, *(3)*	**5SGTC5**	**5SGTC7**
KF40-F1517 *(4)*	600, 150, 4/32	600, 150, 4/32

Notes to Table 1–1:

(1) Packages are flipchip ball grid array (1.0-mm pitch).

(2) Each package row offers pin migration (common board footprint) for all devices in the row.

(3) For full package details, refer to the *Package Information Datasheet for Altera Devices*.

(4) Migration between select Stratix V GT devices and Stratix V GX devices is available. For more information, refer to Table 1–5 on page 1–9.

Table 1–2 lists the Stratix V GX device features.

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

Features	5SGXA3	5SGXA4	5SGXA5	5SGXA7	5SGXA9	5SGXAB	5SGXB5	5SGXB6
Logic Elements (K)	200	300	425	622	840	950	490	597
Registers (K)	302	452	642	939	1,268	1,434	740	901
14.1-Gbps Transceivers	24 or 36	24 or 36	24, 36, or 48	24, 36,or 48	36 or 48	36 or 48	66	66
PCIe hard IP Blocks	1 or 2	1 or 2	1 or 4	1 or 4	1 or 4	1 or 4	1 or 4	1 or 4
Fractional PLLs	24	24	28	28	28	28	24	24
M20K Memory Blocks	800	1,316	2,304	2,560	2,640	2,640	2,100	2,660
M20K Memory (MBits)	16	26	45	50	52	52	41	52
Variable Precision Multipliers (18×18)	376	376	512	512	704	704	798	798
Variable Precision Multipliers (27×27)	188	188	256	256	352	352	399	399
DDR3 SDRAM ×72 DIMM Interfaces	4	4	6	6	6	6	4	4
40G/100G PCS hard IP blocks	No	No	Yes	Yes	No	No	No	No
User I/Os, Full-Duplex LVDS, 14.1-Gbps Transceivers								
Package *(1)*, *(2)*, *(3)*	**5SGXA3**	**5SGXA4**	**5SGXA5**	**5SGXA7**	**5SGXA9**	**5SGXAB**	**5SGXB5**	**5SGXB6**
HH29-H780 *(4)*	264, 66, 24	264, 66, 24	—	—	—	—	—	—
HF35-F1152 *(5)*	552, 138, 24	552, 138, 24	552, 138, 24	552, 138, 24	—	—	—	—
KF35-F1152	432, 108, 36	432, 108, 36	432, 108, 36	432, 108, 36	—	—	—	—
KF40-F1517 *(5)*	624, 156, 36	624, 156, 36	696, 174, 36	696, 174, 36	696, 174, 36	696, 174, 36	—	—
NF40-F1517 *(6)*	—	—	600, 150, 48	600, 150, 48	—	—	—	—
RF40-F1517	—	—	—	—	—	—	432, 108, 66	432, 108, 66
RF43-F1760	—	—	—	—	—	—	600, 150, 66	600, 150, 66

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

Features	5SGXA3	5SGXA4	5SGXA5	5SGXA7	5SGXA9	5SGXAB	5SGXB5	5SGXB6
NF45-F1932 *(5)*	—	—	840, 210, 48	840, 210, 48	840, 210, 48	840, 210, 48	—	—

Notes to Table 1–2:

(1) Packages are flipchip ball grid array (1.0-mm pitch).

(2) LVDS counts are full duplex channels. Each full duplex channel is one transmitter (TX) pair plus one receiver (RX) pair.

(3) Each package row offers pin migration (common circuit board footprint) for all devices in the row.

(4) The 780-pin 5SGXA3 and 5SGXA4 devices are available only in a 33-mm x 33-mm Hybrid flipchip package.

(5) Migration between select Stratix V GX devices and Stratix V GS devices is available. For more information, refer to Table 1–5 on page 1–9.

(6) Migration between select Stratix V GX devices and Stratix V GT devices is available. For more information, refer to Table 1–5 on page 1–9.

Table 1–3 lists the Stratix V GS device features.

Table 1–3. Stratix V GS Device Features

Features	5SGSD2	5SGSD3	5SGSD4	5SGSD5	5SGSD6	5SGSD8
Logic Elements (K)	130	236	332	462	583	703
Registers (K)	196	356	500	696	880	1,060
14.1-Gbps transceivers	12	18	24	36	48	48
PCIe hard IP blocks	1	1	1	1	1 or 2	1 or 2
Fractional PLL	10	12	16	24	28	28
M20K Memory Blocks	450	688	1,062	1,950	2,320	2,688
M20K Memory (MBits)	9	14	22	40	48	55
Variable Precision Multipliers (18×18)	650	1,260	1,892	2,996	3,550	4,096
Variable Precision Multipliers (27×27)	325	630	946	1,498	1,775	2,048
DDR3 SDRAM ×72 DIMM Interfaces	2	2	4	4	7	7
User I/Os, Full-Duplex LVDS, 14.1-Gbps Transceivers						
Package *(1)*, *(2)*, *(3)*	**5SGSD2**	**5SGSD3**	**5SGSD4**	**5SGSD5**	**5SGSD6**	**5SGSD8**
DF23-F484	240, 60, 9	240, 60, 9	—	—	—	—
EF29-F780	400, 100, 12	400, 100, 12	400, 100, 12	—	—	—
GF35/HF35-F1152 *(4)*	—	500, 125, 18	560, 140, 24	560, 140, 24	—	—
KF40-F1517 *(4)*	—	—	700, 175, 36	700, 175, 36	700, 175, 36	700, 175, 36
NF45-F1932 *(4)*	—	—	—	—	900, 225, 48	900, 225, 48

Notes to Table 1–3:

(1) Packages are flipchip ball grid array (1.0-mm pitch).

(2) LVDS counts are full duplex channels. Each full duplex channel is one TX pair plus one RX pair.

(3) Each package row offers pin migration (common circuit board footprint) for all devices in the row.

(4) Migration between select Stratix V GS devices and Stratix V GX devices is available. For more information, refer to Table 1–5 on page 1–9.

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

Table 1–4. Stratix V E Device Features

Features	5SEE9	5SEEB
Logic Elements (K)	840	950
Registers (K)	1,268	1,434
Fractional PLLs	28	28
M20K Memory Blocks	2,640	2,640
M20K Memory (MBits)	52	52
Variable Precision Multipliers (18×18)	704	704
Variable Precision Multipliers (27×27)	352	352
DDR3 SDRAM ×72 DIMM Interfaces	7	7
User I/Os, Full-Duplex LVDS		
Package *(1)*, *(2)*, *(3)*	**5SEE9**	**5SEEB**
H35-F1152 *(4)*	552, 138	552, 138
F40-F1517	696, 174	696, 174
F45-F1932	840, 210	840, 210

Notes to Table 1–4:

(1) Packages are flipchip ball grid array (1.0-mm pitch).

(2) LVDS counts are full duplex channels. Each full duplex channel is one TX pair plus one RX pair.

(3) Each package row offers pin migration (common circuit board footprint) for all devices in the row.

(4) The 1152-pin 5SEE9 and 5SEEB devices are available only in a 42.5-mm x 42.5-mm Hybrid flipchip package.

Each row in Table 1–5 lists which devices allow migration.

Table 1–5. Device Migration List Across All Stratix V Device Variants *(Note 1)*

Package	Stratix V GX								Stratix V GT		Stratix V GS						Stratix V E	
	A3	A4	A5	A7	A9	AB	B5	B6	C5	C7	D2	D3	D4	D5	D6	D8	E9	EB
HH29-H780	✓	✓																
H35-H1152																	✓	✓
DF23-F484											✓	✓						
EF29-F780											✓	✓	✓					
GF35/HF35-F1152 *(2)*	✓	✓	✓	✓								✓	✓	✓				
KF35-F1152	✓	✓	✓	✓														
KF40-F1517	✓	✓	✓	✓	✓	✓							✓	✓	✓	✓		
NF40/KF40-F1517 *(3)*			✓	✓					✓	✓								
RF40-F1517							✓	✓										
F40-F1517																	✓	✓
RF43-F1760							✓	✓										
NF45-F1932			✓	✓	✓	✓									✓	✓		
F45-F1932																	✓	✓

Notes to Table 1–5:

(1) All devices in a given row allow migration.

(2) The 5SGSD3 device is in the GF35 package and has eighteen 14.1-Gbps transceivers. All other devices in this row are in the HF35 package and have twenty-four 14.1-Gbps transceivers.

(3) The 5SGTC5/7 devices in the KF40 package have four 28-Gbps transceivers and thirty-two 12.5-Gbps transceivers. Other devices in this row are in the NF40 package and have forty-eight 14.1-Gbps transceivers.

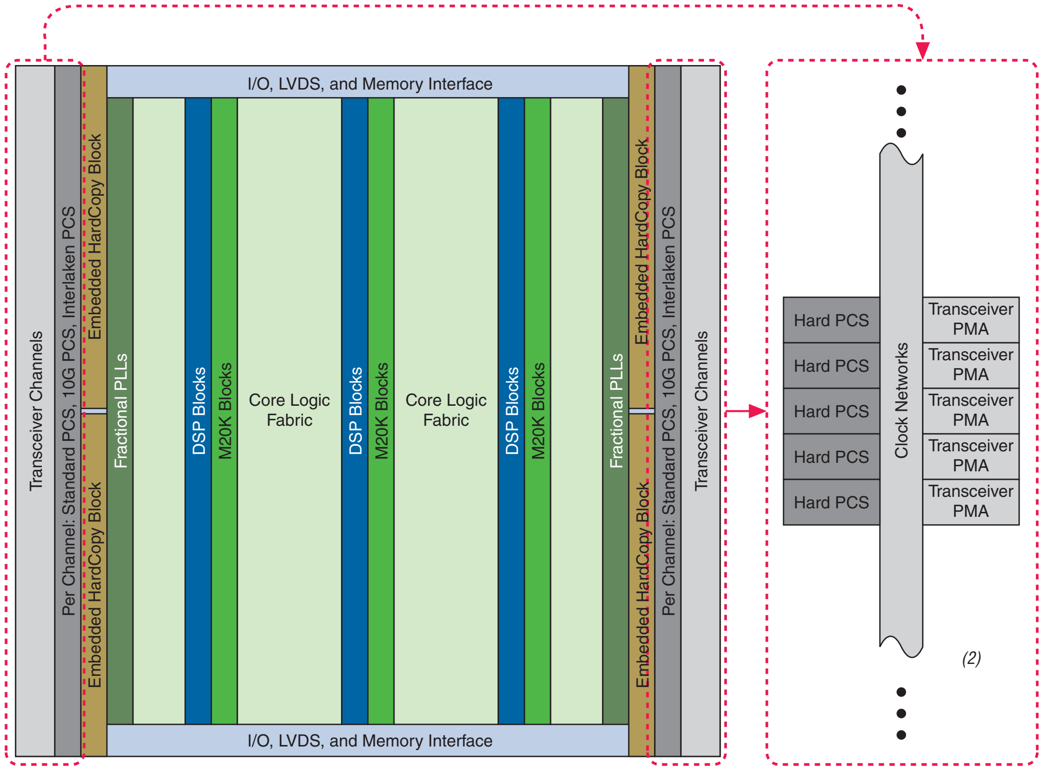
Low-Power Serial Transceivers

Stratix V FPGAs deliver the industry's most flexible transceivers with the highest bandwidth from 600 Mbps to 28 Gbps, low bit error ratio (BER), and low power. Stratix V transceivers have many enhancements to improve flexibility and robustness. These enhancements include robust analog receive clock and data recovery (CDR), advanced pre-emphasis, and equalization for 14.1 Gbps backplanes. In addition, all transceivers are identical with the full featured embedded PCS hard IP to simplify the design, lower the power, and save valuable core resources.

Stratix V transceivers are designed to be compliant with a wide range of standard protocols and data rates, and are equipped with a variety of signal-conditioning features to support backplane, optical module, and chip-to-chip applications.

Stratix V transceivers are located on the left and right sides of the device, as shown in Figure 1–1. They are isolated from the rest of the chip to prevent core and I/O noise from coupling into the transceivers, thereby ensuring optimal signal integrity. The transceiver channels consist of the physical medium attachment (PMA), PCS, and high-speed clock networks. You can also use the unused transceiver PMA channels as additional transmit PLLs. Table 1–6 lists the transceiver PMA features.

Figure 1–1. Stratix V GT/GX/GS Device Chip View ***(Note 1)***



Notes to Figure 1–1:

(1) This figure represents a given variant of a Stratix V device with transceivers. Other variants may have a different floorplan than the one shown here.

(2) You can use the unused transceiver channels as additional transceiver transmit PLLs.

Table 1–6 lists the PMA features for the Stratix V transceivers.

Table 1–6. Transceiver PMA Features

Features	Capability
Backplane support	10GBASE-R, 14.1 Gbps (Stratix V GX/GS devices), 12.5 Gbps (Stratix V GT devices)
Cable driving support	PCIe cable and eSATA applications
Optical module support with EDC	10G Form-factor Pluggable (XFP), Small Form-factor Pluggable (SFP+), Quad Small Form-factor Pluggable (QSFP), CXP, 100G Pluggable (CFP), 100G Form-factor Pluggable
Chip-to-chip support	28 Gbps and 12.5 Gbps (Stratix V GT devices) and 14.1 Gbps (Stratix V GX/GS devices)
Continuous Time Linear Equalization (CTLE)	Receiver 4-stage linear equalization to support high-attenuation channels
Decision Feedback Equalization (DFE)	Receiver 5-tap digital equalizer to minimize losses and crosstalk
Adaptive equalization (ADCE)	Adaptive engine to automatically adjust equalization to compensate for changes over time
PLL-based clock recovery	Superior jitter tolerance versus phase interpolation techniques
Programmable deserialization and word alignment	Flexible deserialization width and configurable word alignment patterns
Transmit equalization (pre-emphasis)	Transmit driver 4-tap pre-emphasis and de-emphasis for protocol compliance under lossy conditions
Ring and logic cell oscillator transmit PLLs	Choice of transmit PLLs per channel, optimized for specific protocols and applications
On-chip instrumentation (EyeQ data-eye monitor)	Allows non-intrusive on-chip monitoring of both width and height of the data eye
Dynamic reconfiguration	Allows reconfiguration of single channels without affecting operation of other channels
Protocol support	Compliance with over 50 industry standard protocols in the range of 600 Mbps to 28 Gbps

The Stratix V core logic connects to the PCS through an 8-, 10-, 16-, 20-, 32-, 40-, 64-, or 66-bit interface, depending on the transceiver data rate and protocol. Stratix V devices contain PCS hard IP to support PCIe Gen 3/2/1, 40G/100G Ethernet, Interlaken, 10GE, XAUI, GbE, SRIO, CPRI, and GPON protocols. All other standard and proprietary protocols are supported through the transceiver PCS hard IP. Table 1–7 lists the transceiver PCS features.

Table 1–7. Transceiver PCS Features (Part 1 of 2)

Protocol	Data Rates (Gbps)	Transmit Data Path	Receiver Data Path
Custom PHY	0.6 to 8.5	Phase compensation FIFO, byte serializer, 8B/10B encoder, bit-slip, and channel bonding	Word aligner, de-skew FIFO, rate match FIFO, 8B/10B decoder, byte deserializer, and byte ordering
Custom 10G PHY	9.98 to 14.1	TX FIFO, gear box, and bit-slip	RX FIFO and gear box
×1, ×4, ×8 PCIe Gen 1/2	2.5 and 5.0	Same as custom PHY plus PIPE 2.0 interface to core logic	Same as custom PHY plus PIPE 2.0 interface to core logic

Table 1–7. Transceiver PCS Features (Part 2 of 2)

Protocol	Data Rates (Gbps)	Transmit Data Path	Receiver Data Path
×1, ×4, ×8 PCIe Gen3	8	Phase compensation FIFO, encoder, scrambler, gear box, and bit slip	Block synchronization, rate match FIFO, decoder, de-scrambler, and phase compensation FIFO
10G Ethernet	10.3125	TX FIFO, 64/66 encoder, scrambler, and gear box	RX FIFO, 64/66 decoder, de-scrambler, block synchronization, and gear box
Interlaken	4.9 to 10.3125	TX FIFO, frame generator, CRC-32 generator, scrambler, disparity generator, and gear box	RX FIFO, frame generator, CRC-32 checker, frame decoder, descrambler, disparity checker, block synchronization, and gearbox
40GBASE-R Ethernet	4 × 10.3125	TX FIFO, 64/66 encoder, scrambler, alignment marker insertion, gearbox, and block striper	RX FIFO, 64/66 decoder, de-scrambler, lane reorder, deskew, alignment marker lock, block synchronization, gear box, and destripper
100GBASE-R Ethernet	10 × 10.3125		
OTN 40 and 100	(4 +1) × 11.3	TX FIFO, channel bonding, and byte serializer	RX FIFO, lane deskew, and byte de-serializer
	(10 +1) × 11.3		
GbE	1.25	Same as custom PHY plus GbE state machine	Same as custom PHY plus GbE state machine
XAUI	3.125 to 4.25	Same as custom PHY plus XAUI state machine for bonding four channels	Same as custom PHY plus XAUI state machine for re-aligning four channels
SRIO	1.25 to 6.25	Same as custom PHY plus SRIO V2.1 compliant ×2 and ×4 channel bonding	Same as custom PHY plus SRIO V2.1-compliant ×2 and ×4 deskew state machine
CPRI	0.6144 to 9.83	Same as Custom PHY plus TX deterministic latency	Same as Custom PHY plus RX deterministic latency
GPON	1.25 and 2.5	Same as custom PHY	Same as custom PHY

PCIe Gen 3/2/1 Hard IP (Embedded HardCopy Block)

Stratix V devices have PCIe hard IP designed for performance, ease-of-use, and increased functionality. The PCIe hard IP consists of the PCS, data link, and transaction layers. It supports Gen 3/2/1 end point and root port up to ×8 lane configurations.

The Stratix V PCIe hard IP operates independently from the core logic, which allows the PCIe link to wake up and complete link training in less than 100 ms while the Stratix V device completes loading the programming file for the rest of the FPGA. It also provides added functionality, which makes it easier to support emerging features such as Single Root I/O Virtualization (SR-IOV) or optional protocol extensions. In addition, the Stratix V device PCIe hard IP has improved end-to-end data path protection using ECC and enables device Configuration via Protocol.

40G and 100G Ethernet Hard IP (Embedded HardCopy Block)

The Stratix V GT, GX, and GS 40G and 100G Ethernet hard IP is standards-compliant and proven. The hard IP includes 40GBASE-R PCS and XLAUI PMA for 40GE, and 100GBASE-R PCS and CAUI PMA for 100GE. The 40G and 100G Ethernet hard IP are scalable because applications requiring multiple 40/100 GbE ports may use a single PLL for the 40/100GBASE-R PCS instantiations to reduce FPGA core and clock resources.

Furthermore, the integrated 10G transceiver simplifies multi-port 40/100GbE system implementation by reducing chip count, board space, and power. Stratix V transceivers interface directly with 40-Gbps QSFP and SFP, and 100-Gbps CFP pluggable modules.

External Memory and General Purpose I/O

Stratix V devices offer high I/O bandwidth with up to seven 72-bit DDR3 SDRAM memory interfaces running at 1,066 MHz/1,600 Mbps and LVDS running at 1.4 Gbps.

Each Stratix V I/O block has a hard FIFO that improves the resynchronization margin as the data is transferred from memory to the FPGA. The hard FIFO also lowers PHY latency, resulting in higher random access performance. General purpose I/Os (GPIOs) include on-chip dynamic termination to reduce the number of external components and minimize reflections. On-package decoupling capacitors suppress noise on the power lines, which reduce noise coupling into the I/Os. Memory banks are isolated to prevent core noise from coupling to the output, thus reducing jitter and providing optimal signal integrity.

The external memory interface block also uses advanced calibration algorithms to compensate for process, voltage and temperature (PVT) variations in the FPGA and external memory components. The advanced algorithms ensure maximum bandwidth and a robust timing margin across all conditions. Stratix V devices also deliver a complete memory solution with the High Performance Memory Controller II (HPMC II) and UniPHY MegaCore® IP that simplify a design for today's advanced memory modules. Table 1–8 lists external memory interface block performance.

Table 1–8. External Memory Interface Performance ***(Note 1)***

Interface	Performance (MHz)
DDR3	1,066
DDR2	533
QDR II	350
QDR II+	550
RLDRAM II	533
RLDRAM III	800

Note to Table 1–8:

(1) The specifications listed in this table are performance targets. For a current achievable performance, use the External Memory Interface Spec Estimator.

Adaptive Logic Module

Stratix V devices use an improved ALM to implement logic functions more efficiently. The Stratix V ALM has eight inputs with a fracturable look-up table (LUT), two dedicated embedded adders, and four dedicated registers.

The Stratix V ALM has the following enhancements:

- Packs 6% more logic when compared with the ALM found in Stratix IV devices
- Implements select 7-input LUT-based functions, all 6-input logic functions, and two independent functions consisting of smaller LUT sizes (such as two independent 4-input LUTs) to optimize core usage
- Adds more registers (four registers per 8-input fracturable LUT). This allows Stratix V devices to maximize core performance at a higher core logic usage and provides easier timing closure for register-rich and heavily pipelined designs.

The Quartus II software leverages the Stratix V ALM logic structure to deliver the highest performance, optimal logic usage, and lowest compile times. The Quartus II software simplifies design re-use because it automatically maps legacy Stratix designs into the new Stratix V ALM architecture.

Clocking

The Stratix V device core clock network is designed to support 717-MHz fabric operations and 1,066-MHz/1,600-Mbps external memory interfaces. The clock network architecture is based on Altera's proven global, quadrant, and peripheral clock structure, which is supported by dedicated clock input pins and fractional clock synthesis PLLs. The Quartus II software identifies all unused sections of the clock network and powers them down, which reduces power consumption.

Fractional PLL

Stratix V devices have up to 32 fractional PLLs that you can use to reduce both the number of oscillators required on the board and the clock pins used in the FPGA by synthesizing multiple clock frequencies from a single reference clock source. In addition, you can use the fractional PLLs for clock network delay compensation, zero-delay buffering, and transmit clocking for transceivers. Fractional PLLs may be individually configured for integer mode or fractional mode with third-order delta-sigma modulation.

Embedded Memory

Stratix V devices contain two types of embedded memory blocks: MLAB (640-bit) and M20K (20-Kbit). MLAB blocks are ideal for wide and shallow memories. M20K blocks are useful for supporting larger memory configurations and include ECC. Both types operate up to 600 MHz and are configurable to be a single- or dual-port RAM, FIFO, ROM, or shift register. These memory blocks are flexible and support a number of memory configurations, as shown in Table 1–9.

Table 1–9. Embedded Memory Block Configuration

MLAB (640 Bits)	M20K (20,480 Bits)
32×20 64×10	512×40 1K×20 2K×10 4K×5 8K×2 16K×1

The Quartus II software simplifies design re-use by automatically mapping memory blocks from legacy Stratix devices into the Stratix V memory architecture.

Variable Precision DSP Block

Stratix V FPGAs feature the industry’s first variable precision DSP block that you can configure to natively support signal processing with precision ranging from 9×9 to 36×36.

You can independently configure each DSP block at compile time as either a dual 18×18 multiply accumulate or a single 27×27 multiply accumulate. With a dedicated 64-bit cascade bus, you can cascade multiple variable precision DSP blocks to implement even higher precision DSP functions efficiently. Table 1–10 lists how different precision is accommodated within a DSP block or by using multiple blocks.

Table 1–10. Variable Precision DSP Block Configurations

Multiplier Size (bits)	DSP Block Resources	Expected Usage
9×9	1/3 of Variable Precision DSP Block	Low precision fixed point
18×18	1/2 of Variable Precision DSP Block	Medium precision fixed point
27×27	1 Variable Precision DSP Block	High precision fixed or single precision floating point
36×36	2 Variable Precision DSP Block	Very high precision fixed point

Complex multiplication is common in DSP algorithms. One of the most popular applications of complex multipliers is the fast Fourier transform (FFT) algorithm. This algorithm has the characteristic of increasing precision requirements on only one side of the multiplier. The variable precision DSP block is designed to support this with a proportional increase in DSP resources with precision growth. Table 1–11 lists complex multiplication with variable precision DSP blocks.

Table 1–11. Complex Multiplication with Variable Precision DSP Blocks

Multiplier Size (bits)	DSP Block Resources	Expected Usage
18×18	2 Variable Precision DSP Blocks	Resource optimized FFTs
18×25	3 Variable Precision DSP Blocks	Accommodate bit growth through FFT stages
18×36	4 Variable Precision DSP Blocks	Highest precision FFT stages
27×27	4 Variable Precision DSP Blocks	Single precision floating point

Additionally, for FFT applications with high dynamic range requirements, only the Altera® FFT MegaCore offers an option of single precision floating point implementation, with the resource usage and performance similar to high-precision fixed point implementations.

Other new features include:

- 64-bit accumulator, the largest in the industry
- Hard pre-adder, available in both 18- and 27-bit modes
- Cascaded output adders for efficient systolic FIR filters
- Internal coefficient register banks
- Enhanced independent multiplier operation
- Efficient support for single- and double-precision floating point arithmetic
- Ability to infer all the DSP block modes through HDL code using the Quartus II design suite.

The variable precision DSP block is ideal for higher bit precision in high-performance DSP applications. At the same time, it can efficiently support the many existing 18-bit DSP applications, such as high definition video processing and remote radio heads. Stratix V FPGAs, with the variable precision DSP block architecture, are the only FPGA family that can efficiently support many different precision levels, up to and including floating point implementations. This flexibility results in increased system performance, reduced power consumption, and reduced architecture constraints on system algorithm designers.

Power Management

Stratix V devices leverage FPGA architectural features and process technology advancements to reduce total power consumption by as much as 30% when compared with Stratix IV devices at the same performance level.

Stratix V devices continue to provide Programmable Power Technology, introduced in earlier generations of Stratix FPGA families. The Quartus II software PowerPlay feature identifies critical timing paths in a design and biases core logic in that path for high performance. The PowerPlay feature also identifies non-critical timing paths and biases core logic in that path for low power instead of high performance. PowerPlay automatically biases core logic to meet performance and optimize power consumption.

Additionally, Stratix V devices have a number of hard IP blocks that not only reduce logic resources but also deliver substantial power savings compared to soft implementations. The list includes PCIe Gen1/Gen2/Gen3, 10G/40G/100G Ethernet, Interlaken PCS, hard I/O FIFOs, and transceivers. Hard IP blocks consume up to 50% less power than equivalent soft implementations.

Stratix V transceivers are also designed for power efficiency. As a result, the transceiver channels consume 50% less power than the previous generation of Stratix FPGAs. The transceiver PMA consumes approximately 90 mW at 6.5 Gbps and 170 mW at 12.5 Gbps.

Incremental Compilation

The Quartus II software incremental compilation feature reduces compilation time by up to 70% and preserves performance to ease timing closure. Incremental compilation supports top-down, bottom-up, and team-based design flows. The incremental compilation feature facilitates modular hierarchical and team-based design flows where different designers compile their respective sections of a design in parallel. Furthermore, different designers or IP providers can develop and optimize different blocks of the design independently, which you can then import into the top-level project.

Enhanced Configuration and Configuration via Protocol

Stratix V device configuration is enhanced for ease-of-use, speed, and cost. Stratix V devices support a new 4-bit bus Active Serial mode (AS×4). AS×4 supports up to a 400 Mbps data rate using small low-cost quad interface Flash devices. This new mode is easy to use and offers an ideal balance between cost and speed. Finally, the Fast Passive Parallel (FPP) interface is enhanced to support 8-, 16-, and 32-bit data widths to meet a wide range of performance and cost goals.

You can configure Stratix V FPGAs using Configuration via Protocol (CvP) with PCIe. CvP with PCIe separates the configuration process into two parts: the PCIe hard IP and periphery and the core logic fabric. CvP uses a much smaller amount of external memory (flash or ROM) because it only has to store the configuration file for the PCIe hard IP and periphery. Also, the 100 ms power-up to active time (for PCIe) is much easier to achieve when only the PCIe hard IP and periphery are loaded. After the PCIe hard IP and periphery are loaded and the root port is booted up, application software running on the root port can send the configuration file for the FPGA fabric across the PCIe link where it is loaded into the FPGA. The FPGA is then fully configured and functional.

Table 1–12 lists the available configuration modes for Stratix V devices.

Table 1–12. Configuration Modes for Stratix V Devices

Mode	Fast or Slow POR	Compression	Encryption	Remote Update	Data Width	Max Clock Rate (MHz)	Max Data Rate (Mbps)
Active Serial	✓	✓	✓	✓	1, 4	100	400
Passive Serial	✓	✓	✓	—	1	125	125
Passive Parallel	✓	✓	✓	*(1)*	8, 16, 32	125	3,000
Configuration via Protocol	—	—	✓	✓	1, 2, 4, 8	—	3,000
Partial Reconfiguration	—	—	✓	✓	16	125	2,000
JTAG	—	—	—	—	1	33	33

Note to Table 1–12:

(1) Remote update support with the Parallel Flash Loader.

Partial Reconfiguration

Partial reconfiguration allows you to reconfigure part of the FPGA while other sections continue to operate. This is required in systems where uptime is critical because it allows you to make updates or adjust functionality without disrupting services. While lowering power and cost, partial reconfiguration also increases the effective logic density by removing the necessity to place the FPGA functions that do not operate simultaneously. Instead, these functions can be stored in external memory and loaded as required. This reduces the size of the FPGA by allowing multiple applications on a single FPGA, saving board space and reducing power.

Up to now, partial reconfiguration solutions have been time-intensive tasks that required you to know all of the intricate FPGA architecture details. Altera simplifies the partial reconfiguration process by building the capability on top of the proven incremental compilation design flow in its Quartus II design software.

Partial reconfiguration is supported through the following configuration options:

- Partial reconfiguration through the FPP ×16 I/O interface
- Configuration via Protocol
- Soft internal core, such as the Nios® II processor.

Automatic Single Event Upset (SEU) Error Detection and Correction

Stratix V devices offer new SEU error detection and correction circuitry that is robust and easy to use. The correction circuitry includes protection for configuration RAM (CRAM) programming bits and user memories. The CRAM is protected by a continuously running cyclical redundancy check (CRC) error detection circuit with integrated ECC that automatically corrects one or two errors and detects higher order multi-bit errors. When more than two errors occur, correction is available through a core programming file reload that provides a complete design refresh while the FPGA continues to operate.

Furthermore, the physical layout of the FPGA is optimized to make the majority of multi-bit upsets appear as independent single- or double-bit errors, which are automatically corrected by the integrated CRAM ECC circuitry. In addition to the CRAM protection in Stratix V devices, the user memories include integrated ECC circuitry and are layout-optimized to enable error detection of 12-bit errors and correction for 8-bit errors.

HardCopy V Devices

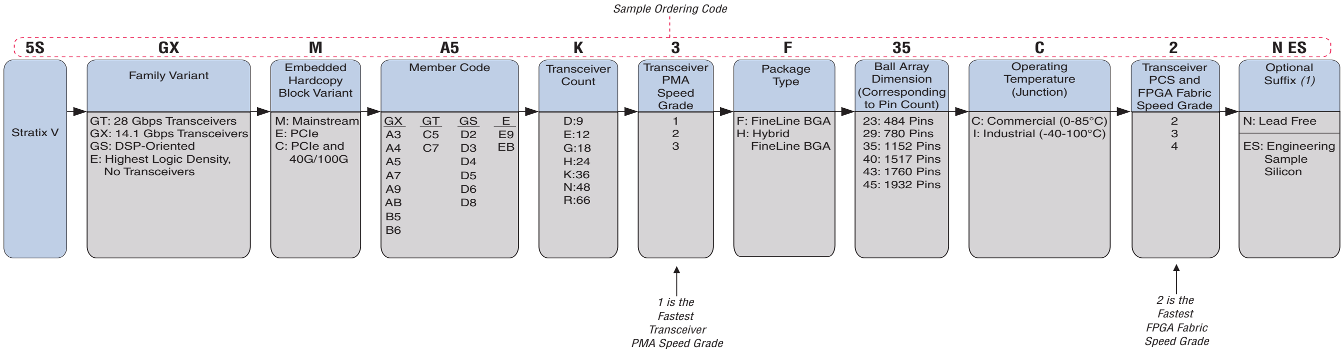
HardCopy V ASICs offer the lowest risk and lowest total cost in ASIC designs with embedded high-speed transceivers. You can prototype and debug with Stratix V FPGAs, then use HardCopy V ASICs for volume production. The proven turnkey process creates a functionally equivalent HardCopy V ASIC with or without embedded transceivers to meet all timing constraints in as little as 12 weeks.

The powerful combination of Stratix V FPGAs and HardCopy V ASICs can help you meet your design requirements. Whether you plan for ASIC production and require the lowest-risk, lowest-cost path from specification to production or require a cost reduction path for your FPGA-based systems, Altera provides the optimal solution for power, performance, and device bandwidth.

Ordering Information

This section describes ordering information for Stratix V GT, GX, GS, and E devices. Figure 1–2 shows the ordering codes for Stratix V devices.

Figure 1–2. Ordering Information for Stratix V Devices



Note to Figure 1–2:

(1) You can select one or both of these options, or you can ignore these options.

Revision History

Table 1–13 lists the revision history for this chapter.

Table 1–13. Revision History

Date	Version	Changes Made
June 2011	1.8	Changed 800 MHz to 1,066 MHz for DDR3 in Table 1–8 and in text.
May 2011	1.7	■ For Stratix V GT devices, changed 14.1 Gbps to 12.5 Gbps. ■ Changed Configuration via PCIe to Configuration via Protocol ■ Updated Table 1–1, Table 1–2, Table 1–3, Table 1–4, Table 1–5, and Table 1–6. ■ Chapter moved to Volume 1.
January 2011	1.6	■ Added Stratix V GS information. ■ Updated tables listing device features. ■ Added device migration information. ■ Updated 12.5-Gbps transceivers to 14.1-Gbps transceivers
December 2010	1.5	Updated Table 1-1.
December 2010	1.4	■ Updated Table 1-1. ■ Updated Figure 1-2. ■ Converted to the new template. ■ Minor text edits.
July 2010	1.3	Updated Table 1–5
July 2010	1.2	■ Updated "Features Summary" on page 1–2 ■ Updated resource counts in Table 1–1 and Table 1–2 ■ Removed "Interlaken PCS Hard IP" and "10G Ethernet Hard IP" ■ Added "40G and 100G Ethernet Hard IP (Embedded HardCopy Block)" on page 1–7 ■ Added information about Configuration via PCIe ■ Added "Partial Reconfiguration" on page 1–12 ■ Added "Ordering Information" on page 1–14
May 2010	1.1	Updated part numbers in Table 1–1 and Table 1–2
April 2010	1.0	Initial release

2. DC and Switching Characteristics for Stratix V Devices

SV53001-2.0

This chapter covers the electrical and switching characteristics for Stratix® V devices. Electrical characteristics include operating conditions and power consumption. Switching characteristics include transceiver specifications, core, and periphery performance. This chapter also describes I/O timing, including programmable I/O element (IOE) delay and programmable output buffer delay.

For information regarding the densities and packages of devices in the Stratix V family, refer to the *Stratix V Device Family Overview* chapter.

Electrical Characteristics

The following sections describe the electrical characteristics of Stratix V devices.

Operating Conditions

When you use Stratix V devices, they are rated according to a set of defined parameters. To maintain the highest possible performance and reliability of Stratix V devices, you must consider the operating requirements described in this chapter.

Stratix V devices are offered in commercial and industrial grades. Commercial devices are offered in –2 (fastest), –3, and –4 speed grades. Industrial devices are offered in –3 and –4 speed grades.

Absolute Maximum Ratings

Absolute maximum ratings define the maximum operating conditions for Stratix V devices. The values are based on experiments conducted with the devices and theoretical modeling of breakdown and damage mechanisms. The functional operation of the device is not implied for these conditions.

CAUTION

Conditions other than those listed in Table 2–1 may cause permanent damage to the device. Additionally, device operation at the absolute maximum ratings for extended periods of time may have adverse effects on the device.

Table 2–1. Absolute Maximum Ratings for Stratix V Devices—Preliminary (Part 1 of 2)

Symbol	Description	Minimum	Maximum	Unit
V_{CC}	Core voltage and periphery circuitry power supply	-0.5	1.35	V
V_{CCPT}	Power supply for programmable power technology	-0.5	1.8	V
V_{CCPGM}	Configuration pins power supply	-0.5	3.75	V
V_{CCAUX}	Auxiliary supply for the programmable power technology	-0.5	3.75	V
V_{CCBAT}	Battery back-up power supply for design security volatile key register	-0.5	3.75	V
V_{CCPD}	I/O pre-driver power supply	-0.5	3.75	V

Table 2–1. Absolute Maximum Ratings for Stratix V Devices—Preliminary (Part 2 of 2)

Symbol	Description	Minimum	Maximum	Unit
V_{CCIO}	I/O power supply	-0.5	3.9	V
V_{CCD_FPLL}	PLL digital power supply	-0.5	3.75	V
V_{CCA_FPLL}	PLL analog power supply	-0.5	3.75	V
V_I	DC input voltage	-0.5	4.0	V
I_{OUT}	DC output current per pin	-25	40	mA
T_J	Operating junction temperature	-55	125	°C
T_{STG}	Storage temperature (No bias)	-65	150	°C

Maximum Allowed Overshoot and Undershoot Voltage

During transitions, input signals may overshoot to the voltage shown in Table 2–2 and undershoot to -2.0 V for input currents less than 100 mA and periods shorter than 20 ns.

Table 2–2 lists the maximum allowed input overshoot voltage and the duration of the overshoot voltage as a percentage of device lifetime. The maximum allowed overshoot duration is specified as a percentage of high time over the lifetime of the device. A DC signal is equivalent to 100% duty cycle. For example, a signal that overshoots to 3.95 V can only be at 3.95 V for ~5% over the lifetime of the device; for a device lifetime of 10 years, this amounts to half a year.

Table 2–2. Maximum Allowed Overshoot During Transitions—Preliminary

Symbol	Description	Condition (V)	Overshoot Duration as % @ T_J = 100°C	Unit
Vi (AC)	AC input voltage	3.8	100	%
		3.85	64	%
		3.9	36	%
		3.95	21	%
		4	12	%
		4.05	7	%
		4.1	4	%
		4.15	2	%
		4.2	1	%

Recommended Operating Conditions

This section lists the functional operating limits for the AC and DC parameters for Stratix V devices. Table 2–3 lists the steady-state voltage and current values expected from Stratix V devices. Power supply ramps must all be strictly monotonic, without plateaus.

Table 2–3. Recommended Operating Conditions for Stratix V Devices—Preliminary

Symbol	Description	Condition	Minimum	Typical	Maximum	Unit
V_{CC}	Core voltage and periphery circuitry power supply	—	0.82	0.85	0.88	V
V_{CCPT}	Power supply for programmable power technology	—	1.45	1.50	1.55	V
V_{CCAUX}	Auxiliary supply for the programmable power technology	—	2.375	2.5	2.625	V
V_{CCPD} *(1)*	I/O pre-driver (3.0 V) power supply	—	2.85	3.0	3.15	V
	I/O pre-driver (2.5 V) power supply	—	2.375	2.5	2.625	V
V_{CCIO}	I/O buffers (3.0 V) power supply	—	2.85	3.0	3.15	V
	I/O buffers (2.5 V) power supply	—	2.375	2.5	2.625	V
	I/O buffers (1.8 V) power supply	—	1.71	1.8	1.89	V
	I/O buffers (1.5 V) power supply	—	1.425	1.5	1.575	V
	I/O buffers (1.35 V) power supply	—	1.283	1.35	1.45	V
	I/O buffers (1.25 V) power supply	—	1.19	1.25	1.31	V
	I/O buffers (1.2 V) power supply	—	1.14	1.2	1.26	V
V_{CCPGM}	Configuration pins (3.0 V) power supply	—	2.85	3.0	3.15	V
	Configuration pins (2.5 V) power supply	—	2.375	2.5	2.625	V
	Configuration pins (1.8 V) power supply	—	1.71	1.8	1.89	V
V_{CCA_FPLL}	PLL analog voltage regulator power supply	—	2.375	2.5	2.625	V
V_{CCD_FPLL}	PLL digital voltage regulator power supply	—	1.45	1.5	1.55	V
V_{CCBAT} *(2)*	Battery back-up power supply (For design security volatile key register)	—	1.2	—	3.0	V
V_I	DC input voltage	—	–0.5	—	3.6	V
V_O	Output voltage	—	0	—	V_{CCIO}	V
T_J	Operating junction temperature	Commercial	0	—	85	°C
		Industrial	–40	—	100	°C
t_{RAMP} *(3)*	Power supply ramp time	Normal POR (PORSEL=0)	200 µs	—	100 ms	—
		Fast POR (PORSEL=1)	200 µs	—	4 ms	—

Notes to Table 2–3:

(1) V_{CCPD} must be 2.5 V when V_{CCIO} is 2.5, 1.8, 1.5, 1.35, 1.25 or 1.2 V. V_{CCPD} must be 3.0 V when V_{CCIO} is 3.0 V.

(2) If you do not use the design security feature in Stratix V devices, connect V_{CCBAT} to a 2.5- or 3.0-V power supply. Stratix V POR circuitry monitors V_{CCBAT}. Stratix V devices will not exit POR if V_{CCBAT} stays at logic low.

(3) Each power supply must reach the recommended operating range within 200 µs.

Table 2–4 lists the transceiver power supply recommended operating conditions for Stratix V GX and GS devices.

Table 2–4. Transceiver Power Supply Operating Conditions for Stratix V GX and GS Devices—Preliminary

Symbol	Description	Minimum	Typical	Maximum	Unit
V_{CCA_GXBL} *(1)*	Transceiver high voltage power (left side)	2.85, 2.375	3.0, 2.5	3.15, 2.625	V
V_{CCA_GXBR} *(1)*	Transceiver high voltage power (right side)				
V_{CCHIP_L}	Transceiver HIP digital power (left side)	0.82	0.85	0.88	V
V_{CCHIP_R}	Transceiver HIP digital power (right side)				
V_{CCHSSI_L}	Transceiver PCS power (left side)	0.82	0.85	0.88	V
V_{CCHSSI_R}	Transceiver PCS power (right side)				
V_{CCR_GXBL} *(2)*	Receiver power (left side)	0.82, 0.95	0.85, 1.0	0.88, 1.05	V
V_{CCR_GXBR} *(2)*	Receiver power (right side)				
V_{CCT_GXBL} *(2)*	Transmitter power (left side)	0.82, 0.95	0.85, 1.0	0.88, 1.05	V
V_{CCT_GXBR} *(2)*	Transmitter power (right side)				
V_{CCH_GXBL}	Transmitter output buffer power (left side)	1.425	1.5	1.575	V
V_{CCH_GXBR}	Transmitter output buffer power (right side)				

Notes to Table 2–4:

(1) This supply must be connected to 3.0 V if the CMU PLL, receiver CDR, or both, are configured at a base data rate > 6.5 Gbps. Up to 6.5 Gbps, you can connect this supply to either 3.0 V or 2.5 V.

(2) This supply must be connected to 1.0 V if the transceiver is configured at a data rate > 6.5 Gbps. Up to 6.5 Gbps, you can connect this supply to either 1.0 V or 0.85 V.

DC Characteristics

This section lists the supply current, I/O pin leakage current, input pin capacitance, on-chip termination tolerance, and hot socketing specifications.

Supply Current

Standby current is the current drawn from the respective power rails used for power budgeting. Use the Excel-based Early Power Estimator (EPE) to get supply current estimates for your design because these currents vary greatly with the resources you use.

For more information about power estimation tools, refer to the *PowerPlay Early Power Estimator User Guide* and the *PowerPlay Power Analysis* chapter in the *Quartus II Handbook*.

I/O Pin Leakage Current

Table 2–5 lists the Stratix V I/O pin leakage current specifications.

Table 2–5. I/O Pin Leakage Current for Stratix V Devices—Preliminary

Symbol	Description	Conditions	Min	Typ	Max	Unit
I_I	Input pin	V_I = 0 V to $V_{CCIOMAX}$	-30	—	30	µA
I_{OZ}	Tri-stated I/O pin	V_O = 0 V to $V_{CCIOMAX}$	-30	—	30	µA

Bus Hold Specifications

Table 2–6 lists the Stratix V device family bus hold specifications.

Table 2–6. Bus Hold Parameters for Stratix V Devices—Preliminary

Parameter	Symbol	Conditions	V_{CCIO}										Unit
			1.2 V		1.5 V		1.8 V		2.5 V		3.0 V		
			Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	
Low sustaining current	I_{SUSL}	$V_{IN} > V_{IL}$ (maximum)	22.5	—	25.0	—	30.0	—	50.0	—	70.0	—	µA
High sustaining current	I_{SUSH}	$V_{IN} < V_{IH}$ (minimum)	-22.5	—	-25.0	—	-30.0	—	-50.0	—	-70.0	—	µA
Low overdrive current	I_{ODL}	$0V < V_{IN} < V_{CCIO}$	—	120	—	160	—	200	—	300	—	500	µA
High overdrive current	I_{ODH}	$0V < V_{IN} < V_{CCIO}$	—	-120	—	-160	—	-200	—	-300	—	-500	µA
Bus-hold trip point	V_{TRIP}	—	0.45	0.95	0.50	1.00	0.68	1.07	0.70	1.70	0.80	2.00	V

On-Chip Termination (OCT) Specifications

If you enable OCT calibration, calibration is automatically performed at power-up for I/Os connected to the calibration block. Table 2–7 lists the Stratix V OCT termination calibration accuracy specifications.

Table 2–7. OCT Calibration Accuracy Specifications for Stratix V Devices—Preliminary (Part 1 of 2) *(Note 1)*

Symbol	Description	Conditions	Calibration Accuracy			Unit
			C2	C3,I3	C4,I4	
25-Ω R_S	Internal series termination with calibration (25-Ω setting)	V_{CCIO} = 3.0, 2.5, 1.8, 1.5, 1.2 V	±15	±15	±15	%
50-Ω R_S	Internal series termination with calibration (50-Ω setting)	V_{CCIO} = 3.0, 2.5, 1.8, 1.5, 1.2 V	±15	±15	±15	%
34-Ω and 40-Ω R_S	Internal series termination with calibration (34-Ω and 40-Ω setting)	V_{CCIO} = 1.5, 1.35, 1.25, 1.2 V	±15	±15	±15	%
48-Ω, 60-Ω, and 80-Ω R_S	Internal series termination with calibration (48-Ω, 60-Ω, and 80-Ω setting)	V_{CCIO} = 1.2 V	±15	±15	±15	%
50-Ω R_T	Internal parallel termination with calibration (50-Ω setting)	V_{CCIO} = 2.5, 1.8, 1.5, 1.2 V	-10 to +40	-10 to +40	-10 to +40	%

Table 2–7. OCT Calibration Accuracy Specifications for Stratix V Devices—Preliminary (Part 2 of 2) *(Note 1)*

Symbol	Description	Conditions	Calibration Accuracy			Unit
			C2	C3,I3	C4,I4	
20-Ω, 30-Ω, 40-Ω,60-Ω, and 120-Ω R_T	Internal parallel termination with calibration (20-Ω, 30-Ω, 40-Ω, 60-Ω, and 120-Ω setting)	V_{CCIO} = 1.5, 1.35, 1.25 V	-10 to +40	-10 to +40	-10 to +40	%
60-Ω and 120-Ω R_T	Internal parallel termination with calibration (60-Ω and 120-Ω setting)	V_{CCIO} = 1.2	-10 to +40	-10 to +40	-10 to +40	%
25-Ω $R_{S\_left\_shift}$	Internal left shift series termination with calibration (25-Ω $R_{S\_left\_shift}$ setting)	V_{CCIO} = 3.0, 2.5, 1.8, 1.5, 1.2 V	±15	±15	±15	%

Note to Table 2–7:

(1) OCT calibration accuracy is valid at the time of calibration only.

Calibration accuracy for the calibrated series and parallel OCTs are applicable at the moment of calibration. When process, voltage, and temperature (PVT) conditions change after calibration, the tolerance may change. Table 2–8 lists the Stratix V OCT without calibration resistance tolerance to PVT changes.

Table 2–8. OCT Without Calibration Resistance Tolerance Specifications for Stratix V Devices—Preliminary ***(Note 1)***

Symbol	Description	Conditions	Resistance Tolerance			Unit
			C2	C3,I3	C4,I4	
25-Ω R_S	Internal series termination without calibration (25-Ω setting)	V_{CCIO} = 3.0 and 2.5 V	±30	±40	±40	%
25-Ω R_S	Internal series termination without calibration (25-Ω setting)	V_{CCIO} = 1.8 and 1.5 V	±30	±40	±40	%
25-Ω R_S	Internal series termination without calibration (25-Ω setting)	V_{CCIO} = 1.2 V	±35	±50	±50	%
50-Ω R_S	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 3.0 and 2.5 V	±30	±40	±40	%
50-Ω R_S	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 1.8 and 1.5 V	±30	±40	±40	%
50-Ω R_S	Internal series termination without calibration (50-Ω setting)	V_{CCIO} = 1.2 V	±35	±50	±50	%
100-Ω R_D	Internal differential termination (100-Ω setting)	V_{CCIO} = 2.5 V	±25	±25	±25	%

Note to Table 2–8:

(1) Pending silicon characterization.

OCT calibration is automatically performed at power-up for OCT-enabled I/Os. Table 2–9 lists the OCT variation with temperature and voltage after power-up calibration. Use Table 2–9 to determine the OCT variation after power-up calibration and Equation 2–1 to determine the OCT variation without re-calibration.

Equation 2–1. OCT Variation Without Re-Calibration for Stratix V Devices—Preliminary *(Note 1)*, *(2)*, *(3)*, *(4)*, *(5)*, *(6)*

$$R_{OCT} = R_{SCAL}\left(1 + \langle\frac{dR}{dT} \times \Delta T\rangle \pm \langle\frac{dR}{dV} \times \Delta V\rangle\right)$$

Notes to Equation 2–1:

(1) The R_{OCT} value calculated from Equation 2–1 shows the range of OCT resistance with the variation of temperature and V_{CCIO}.

(2) R_{SCAL} is the OCT resistance value at power-up.

(3) ∆T is the variation of temperature with respect to the temperature at power-up.

(4) ∆V is the variation of voltage with respect to the V_{CCIO} at power-up.

(5) dR/dT is the percentage change of R_{SCAL} with temperature.

(6) dR/dV is the percentage change of R_{SCAL} with voltage.

Table 2–9 lists the on-chip termination variation after power-up calibration.

Table 2–9. OCT Variation after Power-Up Calibration for Stratix V Devices—Preliminary *(Note 1)*, *(2)*

Symbol	Description	V_{CCIO} (V)	Typical	Unit
dR/dV	OCT variation with voltage without re-calibration	3.0	0.0297	%/mV
		2.5	0.0344	
		1.8	0.0499	
		1.5	0.0744	
		1.2	0.1241	
dR/dT	OCT variation with temperature without re-calibration	3.0	0.189	%/°C
		2.5	0.208	
		1.8	0.266	
		1.5	0.273	
		1.2	0.317	

Notes to Table 2–9:

(1) Valid for a V_{CCIO} range of ±5% and a temperature range of 0° to 85°C.

(2) Pending silicon characterization.

Pin Capacitance

Table 2–10 lists the Stratix V device family pin capacitance.

Table 2–10. Pin Capacitance for Stratix V Devices—Preliminary

Symbol	Description	Typical	Unit
C_{IOTB}	Input capacitance on the top and bottom I/O pins	5.5	pF
C_{IOLR}	Input capacitance on the left and right I/O pins	5.5	pF
C_{OUTFB}	Input capacitance on dual-purpose clock output and feedback pins	5.5	pF

Hot Socketing

Table 2–11 lists the hot socketing specifications for Stratix V devices.

Table 2–11. Hot Socketing Specifications for Stratix V Devices—Preliminary

Symbol	Description	Maximum
$I_{IOPIN\ (DC)}$	DC current per I/O pin	300 μA
$I_{IOPIN\ (AC)}$	AC current per I/O pin	8 mA *(1)*
$I_{XCVR\text{-}TX\ (DC)}$ *(2)*	DC current per transceiver transmitter pin	100 mA
$I_{XCVR\text{-}RX\ (DC)}$ *(2)*	DC current per transceiver receiver pin	50 mA

Notes to Table 2–11:

(1) The I/O ramp rate is 10 ns or more. For ramp rates faster than 10 ns, $|I_{IOPIN}|$ = C dv/dt, in which C is the I/O pin capacitance and dv/dt is the slew rate.

(2) These specifications are preliminary.

Internal Weak Pull-Up Resistor

Table 2–12 lists the weak pull-up resistor values for Stratix V devices.

Table 2–12. Internal Weak Pull-Up Resistor for Stratix V Devices—Preliminary ***(Note 1)*, *(2)***

Symbol	Description	V_{CCIO} Conditions (V) *(3)*	Min	Typ	Max	Unit
R_{PU}	Value of the I/O pin pull-up resistor before and during configuration, as well as user mode if you enable the programmable pull-up resistor option.	3.0 ±5%	—	25	—	kΩ
		2.5 ±5%	—	25	—	kΩ
		1.8 ±5%	—	25	—	kΩ
		1.5 ±5%	—	25	—	kΩ
		1.35 ±5%	—	25	—	kΩ
		1.25 ±5%	—	25	—	kΩ
		1.2 ±5%	—	25	—	kΩ

Notes to Table 2–12:

(1) All I/O pins have an option to enable weak pull-up except the configuration, test, and JTAG pins.

(2) The internal weak pull-down feature is only available for the JTAG `TCK` pin. The typical value for this internal weak pull-down resistor is approximately 25 kΩ.

(3) Pin pull-up resistance values may be lower if an external source drives the pin higher than V_{CCIO}.

I/O Standard Specifications

Table 2–13 through Table 2–18 list the input voltage (V_{IH} and V_{IL}), output voltage (V_{OH} and V_{OL}), and current drive characteristics (I_{OH} and I_{OL}) for various I/O standards supported by Stratix V devices. These tables also show the Stratix V device family I/O standard specifications. The V_{OL} and V_{OH} values are valid at the corresponding I_{OH} and I_{OL}, respectively.

For an explanation of the terms used in Table 2–13 through Table 2–18, refer to "Glossary" on page 2–30.

Table 2–13. Single-Ended I/O Standards for Stratix V Devices—Preliminary

I/O Standard	V_{CCIO} (V)			V_{IL} (V)		V_{IH} (V)		V_{OL} (V)	V_{OH} (V)	I_{OL} (mA)	I_{OH} (mA)
	Min	Typ	Max	Min	Max	Min	Max	Max	Min		
LVTTL	2.85	3	3.15	-0.3	0.8	1.7	3.6	0.4	2.4	2	-2
LVCMOS	2.85	3	3.15	-0.3	0.8	1.7	3.6	0.2	V_{CCIO} - 0.2	0.1	-0.1
2.5 V	2.375	2.5	2.625	-0.3	0.7	1.7	3.6	0.4	2	1	-1
1.8 V	1.71	1.8	1.89	-0.3	0.35 * V_{CCIO}	0.65 * V_{CCIO}	V_{CCIO} + 0.3	0.45	V_{CCIO} - 0.45	2	-2
1.5 V	1.425	1.5	1.575	-0.3	0.35 * V_{CCIO}	0.65 * V_{CCIO}	V_{CCIO} + 0.3	0.25 * V_{CCIO}	0.75 * V_{CCIO}	2	-2
1.2 V	1.14	1.2	1.26	-0.3	0.35 * V_{CCIO}	0.65 * V_{CCIO}	V_{CCIO} + 0.3	0.25 * V_{CCIO}	0.75 * V_{CCIO}	2	-2

Table 2–14. Single-Ended SSTL, HSTL, and HSUL I/O Reference Voltage Specifications for Stratix V Devices—Preliminary (Part 1 of 2)

I/O Standard	V_{CCIO} (V)			V_{REF} (V)			V_{TT} (V)		
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
SSTL-2 Class I, II	2.375	2.5	2.625	0.49 * V_{CCIO}	0.5 * V_{CCIO}	0.51 * V_{CCIO}	V_{REF} - 0.04	V_{REF}	V_{REF} + 0.04
SSTL-18 Class I, II	1.71	1.8	1.89	0.833	0.9	0.969	V_{REF} - 0.04	V_{REF}	V_{REF} + 0.04
SSTL-15 Class I, II	1.425	1.5	1.575	0.49 * V_{CCIO}	0.5 * V_{CCIO}	0.51 * V_{CCIO}	0.49 * V_{CCIO}	0.5 * VCCIO	0.51 * V_{CCIO}
SSTL 135 Class I, II	1.283	1.35	1.418	0.49 * V_{CCIO}	0.5 * V_{CCIO}	0.51 * V_{CCIO}	0.49 * V_{CCIO}	0.5 * V_{CCIO}	0.51 * V_{CCIO}
SSTL 125 Class I, II	1.19	1.25	1.26	0.49 * V_{CCIO}	0.5 * V_{CCIO}	0.51 * V_{CCIO}	0.49 * V_{CCIO}	0.5 * VCCIO	0.51 * V_{CCIO}
SSTL 12 Class I, II	1.14	1.20	1.26	0.49 * V_{CCIO}	0.5 * V_{CCIO}	0.51 * V_{CCIO}	0.49 * V_{CCIO}	0.5 * VCCIO	0.51 * V_{CCIO}
HSTL-18 Class I, II	1.71	1.8	1.89	0.85	0.9	0.95	—	V_{CCIO}/2	—
HSTL-15 Class I, II	1.425	1.5	1.575	0.68	0.75	0.9	—	V_{CCIO}/2	—

Table 2–14. Single-Ended SSTL, HSTL, and HSUL I/O Reference Voltage Specifications for Stratix V Devices—Preliminary (Part 2 of 2)

I/O Standard	V_{CCIO} (V)			V_{REF} (V)			V_{TT} (V)		
	Min	Typ	Max	Min	Typ	Max	Min	Typ	Max
HSTL-12 Class I, II	1.14	1.2	1.26	0.47 * V_{CCIO}	0.5 * V_{CCIO}	0.53 * V_{CCIO}	—	V_{CCIO}/2	—
HSUL-12	1.14	1.2	1.3	0.49 * V_{CCIO}	0.5 * V_{CCIO}	0.51 * V_{CCIO}	—	—	—

Table 2–15. Single-Ended SSTL, HSTL, and HSUL I/O Standards Signal Specifications for Stratix V Devices—Preliminary (Part 1 of 2)

I/O Standard	$V_{IL(DC)}$ (V)		$V_{IH(DC)}$ (V)		$V_{IL(AC)}$ (V)	$V_{IH(AC)}$ (V)	V_{OL} (V)	V_{OH} (V)	I_{ol} (mA)	I_{oh} (mA)
	Min	Max	Min	Max	Max	Min	Max	Min		
SSTL-2 Class I	-0.3	V_{REF} - 0.15	V_{REF} + 0.15	V_{CCIO} + 0.3	V_{REF} - 0.31	V_{REF} + 0.31	V_{TT} - 0.608	V_{TT} + 0.608	8.1	-8.1
SSTL-2 Class II	-0.3	V_{REF} - 0.15	V_{REF} + 0.15	V_{CCIO} + 0.3	V_{REF} - 0.31	V_{REF} + 0.31	V_{TT} - 0.81	V_{TT} + 0.81	16.2	-16.2
SSTL-18 Class I	-0.3	V_{REF} - 0.125	V_{REF} + 0.125	V_{CCIO} + 0.3	V_{REF} - 0.25	V_{REF} + 0.25	V_{TT} - 0.603	V_{TT} + 0.603	6.7	-6.7
SSTL-18 Class II	-0.3	V_{REF} - 0.125	V_{REF} + 0.125	V_{CCIO} + 0.3	V_{REF} - 0.25	V_{REF} + 0.25	0.28	V_{CCIO} - 0.28	13.4	-13.4
SSTL-15 Class I	—	V_{REF} - 0.1	V_{REF} + 0.1	—	V_{REF} - 0.175	V_{REF} + 0.175	0.2 * V_{CCIO}	0.8 * V_{CCIO}	8	-8
SSTL-15 Class II	—	V_{REF} - 0.1	V_{REF} + 0.1	—	V_{REF} - 0.175	V_{REF} + 0.175	0.2 * V_{CCIO}	0.8 * V_{CCIO}	16	-16
SSTL 135 Class I, II	—	V_{REF} - 0.09	V_{REF} + 0.09	—	V_{REF} - 0.16	V_{REF} + 0.16	TBD *(1)*	TBD *(1)*	TBD *(1)*	TBD *(1)*
SSTL 125 Class I, II	—	V_{REF} - 0.85	V_{REF} + 0.85	—	V_{REF} - 0.15	V_{REF} + 0.15	TBD *(1)*	TBD *(1)*	TBD *(1)*	TBD *(1)*
SSTL 12 Class I, II	—	V_{REF} - 0.1	V_{REF} + 0.1	—	V_{REF} - 0.15	V_{REF} + 0.15	TBD *(1)*	TBD *(1)*	TBD *(1)*	TBD *(1)*
HSTL-18 Class I	—	V_{REF} - 0.1	V_{REF} + 0.1	—	V_{REF} - 0.2	V_{REF} + 0.2	0.4	V_{CCIO} - 0.4	8	-8
HSTL-18 Class II	—	V_{REF} - 0.1	V_{REF} + 0.1	—	V_{REF} - 0.2	V_{REF} + 0.2	0.4	V_{CCIO} - 0.4	16	-16
HSTL-15 Class I	—	V_{REF} - 0.1	V_{REF} + 0.1	—	V_{REF} - 0.2	V_{REF} + 0.2	0.4	V_{CCIO} - 0.4	8	-8
HSTL-15 Class II	—	V_{REF} - 0.1	V_{REF} + 0.1	—	V_{REF} - 0.2	V_{REF} + 0.2	0.4	V_{CCIO} - 0.4	16	-16
HSTL-12 Class I	-0.15	V_{REF} - 0.08	V_{REF} + 0.08	V_{CCIO} + 0.15	V_{REF} - 0.15	V_{REF} + 0.15	0.25* V_{CCIO}	0.75* V_{CCIO}	8	-8
HSTL-12 Class II	-0.15	V_{REF} - 0.08	V_{REF} + 0.08	V_{CCIO} + 0.15	V_{REF} - 0.15	V_{REF} + 0.15	0.25* V_{CCIO}	0.75* V_{CCIO}	16	-16

Table 2–15. Single-Ended SSTL, HSTL, and HSUL I/O Standards Signal Specifications for Stratix V Devices—Preliminary (Part 2 of 2)

I/O Standard	$V_{IL(DC)}$ (V)		$V_{IH(DC)}$ (V)		$V_{IL(AC)}$ (V)	$V_{IH(AC)}$ (V)	V_{OL} (V)	V_{OH} (V)	I_{ol} (mA)	I_{oh} (mA)
	Min	Max	Min	Max	Max	Min	Max	Min		
HSUL-12	—	V_{REF} - 0.13	V_{REF} + 0.13	—	V_{REF} - 0.22	V_{REF} + 0.22	0.1* V_{CCIO}	0.9* V_{CCIO}	TBD *(1)*	TBD *(1)*

Note to Table 2–15:

(1) Pending silicon characterization.

Table 2–16. Differential SSTL I/O Standards for Stratix V Devices—Preliminary

I/O Standard	V_{CCIO} (V)			$V_{SWING(DC)}$ (V)		$V_{X(AC)}$ (V)			$V_{SWING(AC)}$ (V)		$V_{OX(AC)}$ (V)		
	Min	Typ	Max	Min	Max	Min	Typ	Max	Min	Max	Min	Typ	Max
SSTL-2 Class I, II	2.375	2.5	2.625	0.3	V_{CCIO} + 0.6	V_{CCIO}/2 - 0.2	—	V_{CCIO}/2 + 0.2	0.62	V_{CCIO} + 0.6	V_{CCIO}/2 - 0.15	—	V_{CCIO}/2 + 0.15
SSTL-18 Class I, II	1.71	1.8	1.89	0.25	V_{CCIO} + 0.6	V_{CCIO}/2 - 0.175	—	V_{CCIO}/2 + 0.175	0.5	V_{CCIO} + 0.6	V_{CCIO}/2 - 0.125	—	V_{CCIO}/2 + 0.125
SSTL-15 Class I, II	1.425	1.5	1.575	0.2	-0.2	-0.15	—	0.15	-0.35	0.35	—	V_{CCIO}/2	—
SSTL 135 Class I, II	1.283	1.35	1.45	0.2	-0.2	V_{REF} -0.135	V_{CCIO}/2	V_{REF} + 0.135	TBD *(1)*	TBD *(1)*	V_{REF} -0.15	—	V_{REF} +0.15
SSTL 125 Class I, II	1.19	1.25	1.31	TBD *(1)*	—	TBD *(1)*	V_{CCIO}/2	TBD *(1)*	TBD *(1)*	—	TBD *(1)*	TBD *(1)*	TBD *(1)*
SSTL 12 Class I, II	1.14	1.2	1.26	TBD *(1)*	—	V_{REF} -0.15	V_{CCIO}/2	V_{REF} + 0.15	-0.30	0.30	TBD *(1)*	TBD *(1)*	TBD *(1)*

Note to Table 2–16:

(1) Pending silicon characterization.

Table 2–17. Differential HSTL and HSUL I/O Standards for Stratix V Devices—Preliminary

I/O Standard	V_{CCIO} (V)			$V_{DIF(DC)}$ (V)		$V_{X(AC)}$ (V)			$V_{CM(DC)}$ (V)			$V_{DIF(AC)}$ (V)	
	Min	Typ	Max	Min	Max	Min	Typ	Max	Min	Typ	Max	Min	Max
HSTL-18 Class I, II	1.71	1.8	1.89	0.2	—	0.78	—	1.12	0.78	—	1.12	0.4	—
HSTL-15 Class I, II	1.425	1.5	1.575	0.2	—	0.68	—	0.9	0.68	—	0.9	0.4	—
HSTL-12 Class I, II	1.14	1.2	1.26	0.16	V_{CCIO} + 0.3	—	0.5* V_{CCIO}	—	0.4* V_{CCIO}	0.5* V_{CCIO}	0.6* V_{CCIO}	0.3	V_{CCIO} + 0.48
HSUL-12	1.14	1.2	1.3	0.26	0.26	0.5*V_{CCIO} - 0.12	0.5* V_{CCIO}	0.5*V_{CCIO} + 0.12	0.4* V_{CCIO}	0.5* V_{CCIO}	0.6* V_{CCIO}	0.44	0.44

Table 2–18. Differential I/O Standard Specifications for Stratix V Devices—Preliminary ***(Note 1)***

I/O Standard	V_{CCIO} (V)			V_{ID} (mV)			$V_{ICM(DC)}$ (V)			V_{OD} (V) *(2)*			V_{OCM} (V) *(2)*		
	Min	Typ	Max	Min	Condition	Max	Min	Condition	Max	Min	Typ	Max	Min	Typ	Max
PCML	Transmitter, receiver, and input reference clock pins of the high-speed transceivers use the PCML I/O standard. For transmitter, receiver, and reference clock I/O pin specifications, refer to Table 2–19 on page 2–14.														
2.5 V LVDS	2.375	2.5	2.625	100	V_{CM} = 1.25 V	—	0.05	—	1.8	0.247	—	0.6	1.125	1.25	1.375
						—	1.05	—	1.55	0.247	—	0.6	1.125	1.25	1.375
RSDS (HIO)	2.375	2.5	2.625	100	V_{CM} = 1.25 V	—	0.3	—	1.4	0.1	0.2	0.6	0.5	1.2	1.4
Mini-LVDS (HIO)	2.375	2.5	2.625	200	—	600	0.4	—	1.325	0.25	—	0.6	1	1.2	1.4
LVPECL	2.375	2.5	2.625	300	—	—	0.6	$D_{MAX} \le$ 700 Mbps	1.8 *(3)*	—	—	—	—	—	—
	2.375	2.5	2.625	300	—	—	1	$D_{MAX} >$ 700 Mbps	1.6 *(3)*	—	—	—	—	—	—

Notes to Table 2–18*:*

(1) The 1.4-V and 1.5-V PCML transceiver I/O standard specifications are described in "Transceiver Performance Specifications" on page 2–14.

(2) RL range: $90 \le RL \le 110\ \Omega$.

(3) For $D_{MAX} > 700$ Mbps, the minimum input voltage is 0.85 V; the maximum input voltage is 1.75 V. For $F_{MAX} \le$ 700 Mbps, the minimum input voltage is 0.45 V; the maximum input voltage is 1.95 V.

Power Consumption

Altera offers two ways to estimate power consumption for a design—the Excel-based Early Power Estimator and the Quartus® II PowerPlay Power Analyzer feature.

You typically use the interactive Excel-based Early Power Estimator before designing the FPGA to get a magnitude estimate of the device power. The Quartus II PowerPlay Power Analyzer provides better quality estimates based on the specifics of the design after you complete place-and-route. The PowerPlay Power Analyzer can apply a combination of user-entered, simulation-derived, and estimated signal activities that, when combined with detailed circuit models, yields very accurate power estimates.

For more information about power estimation tools, refer to the *PowerPlay Early Power Estimator User Guide* and the *PowerPlay Power Analysis* chapters in the *Quartus II Handbook*.

Switching Characteristics

This section provides performance characteristics of the Stratix V core and periphery blocks for commercial grade devices.

These characteristics can be designated as Preliminary or Final.

- Preliminary characteristics are created using simulation results, process data, and other known parameters. The title of these tables show the designation as "Preliminary."
- Final numbers are based on actual silicon characterization and testing. The numbers reflect the actual performance of the device under worst-case silicon process, voltage, and junction temperature conditions. There are no designations on finalized tables.

Transceiver Performance Specifications

This section describes transceiver performance specifications.

Table 2–19 lists the Stratix V GX and GS transceiver specifications.

Table 2–19. Transceiver Specifications for Stratix V GX and GS Devices—Preliminary (Part 1 of 3) *(Note 1)*

Symbol/ Description	Conditions	–1 Commercial Speed Grade			–2 Commercial/Industrial Speed Grade			–3 Commercial/Industrial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Reference Clock											
Supported I/O Standards	1.2 V PCML, 1.4 V PCML, 1.5 V PCML, 2.5 V PCML, Differential LVPECL, LVDS, and HCSL										
Input frequency from `REFCLK` input pins	—	40	—	710	40	—	710	40	—	710	MHz
Duty cycle	—	45	—	55	45	—	55	45	—	55	%
Spread-spectrum modulating clock frequency	PCI Express® (PCIe®)	30	—	33	30	—	33	30	—	33	kHz
Spread-spectrum downspread	PCIe	—	0 to -0.5%	—	—	0 to -0.5%	—	—	0 to -0.5%	—	—
On-chip termination resistors	—	—	100	—	—	100	—	—	100	—	Ω
V_{ICM} (AC coupled)	—	1000/850 *(2)*			1000/850 *(2)*			1000/850 *(2)*			mV
V_{ICM} (DC coupled)	HCSL I/O standard for PCIe reference clock	250	—	550	250	—	550	250	—	550	mV
R_{REF}	—	—	2000 ±1%	—	—	2000 ±1%	—	—	2000 ±1%	—	Ω
Transceiver Clocks											
`fixedclk` clock frequency	PCIe Receiver Detect	—	125	—	—	125	—	—	125	—	MHz

Table 2–19. Transceiver Specifications for Stratix V GX and GS Devices—Preliminary (Part 2 of 3) *(Note 1)*

Symbol/ Description	Conditions	–1 Commercial Speed Grade			–2 Commercial/Industrial Speed Grade			–3 Commercial/Industrial Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Avalon-MM PHY management clock frequency	< 150										MHz
Receiver											
Supported I/O Standards	1.4 V PCML, 1.5 V PCML, 2.5 V PCML, LVPECL, and LVDS										
Data rate (Standard PCS)	—	600	—	8500	600	—	8500	600	—	6500	Mbps
Data rate (10G PCS)	—	2000	—	14100	2000	—	12500	2000	—	8500	Mbps
Absolute V_{MAX} for a receiver pin *(3)*	—	—	—	1.2	—	—	1.2	—	—	1.2	V
Absolute V_{MIN} for a receiver pin	—	-0.4	—	—	-0.4	—	—	-0.4	—	—	V
Maximum peak-to-peak differential input voltage V_{ID} (diff p-p) before device configuration	—	—	—	1.6	—	—	1.6	—	—	1.6	V
Maximum peak-to-peak differential input voltage V_{ID} (diff p-p) after device configuration	V_{CCR_GXB} = 1.0 V and V_{ICM} = 0.65 V	—	—	2.2	—	—	2.2	—	—	2.2	V
	V_{CCR_GXB} = 0.85 V and V_{ICM} = 0.55 V	—	—	2.6	—	—	2.6	—	—	2.6	V
Minimum differential eye opening at receiver serial input pins *(4)*	—	85	—	—	85	—	—	85	—	—	mV
Differential on-chip termination resistors	85–Ω setting	85			85			85			Ω
	100–Ω setting	100			100			100			Ω
	120–Ω setting	120			120			120			Ω
	150-Ω setting	150			150			150			Ω
Programmable equalization	—	—	—	20	—	—	20	—	—	20	dB
Programmable DC gain	DC Gain Setting = 0	—	0	—	—	0	—	—	0	—	dB
	DC Gain Setting = 1	—	2	—	—	2	—	—	2	—	dB
	DC Gain Setting = 2	—	4	—	—	4	—	—	4	—	dB
	DC Gain Setting = 3	—	6	—	—	6	—	—	6	—	dB
	DC Gain Setting = 4	—	8	—	—	8	—	—	8	—	dB

Table 2–19. Transceiver Specifications for Stratix V GX and GS Devices—Preliminary (Part 3 of 3) ***(Note 1)***

Symbol/ Description	Conditions	–1 Commercial Speed Grade			–2 Commercial/Industrial Speed Grade			–3 Commercial/Industrial Speed Grade			Unit
		Min	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	
Transmitter											
Supported I/O Standards	1.4 V and 1.5 V PCML										
Data rate (Standard PCS)	—	600	—	8500	600	—	8500	600	—	6500	Mbps
Data rate (10G PCS)	—	2000	—	14100	2000	—	12500	2000	—	8500	Mbps
V_{OCM}	0.65-V setting	—	650	—	—	650	—	—	650	—	mV
Differential on-chip termination resistors	85-Ω setting	85			85			85			Ω
	100-Ω setting	100			100			100			Ω
	120-Ω setting	120			120			120			Ω
	150-Ω setting	150			150			150			Ω
Transmitter											
Rise time *(5)*	—	30	—	160	30	—	160	30	—	160	ps
Fall time *(5)*	—	30	—	160	30	—	160	30	—	160	ps
CMU PLL											
Supported Data Range	—	600	—	14100	600	—	12500	600	—	8500	Mbps
ATX PLL											
Supported Data Range	VCO post-divider L=1	8000	—	14100	8000	—	12500	8000	—	8500	Mbps
	L=2	4000	—	7050	4000	—	7050	4000	—	7050	Mbps
	L=4	2000	—	3525	2000	—	3525	2000	—	3525	Mbps
Input Reference Clock Frequency *(6)*	—	100	—	875	100	—	875	100	—	875	MHz
Transceiver-FPGA Fabric Interface											
Interface speed	—	25	—	283	25	—	266	25	—	250	MHz

Notes to Table 2–19:

(1) Speed grades shown in Table 2–19 refer to the PMA Speed Grade in the device ordering code. The maximum data rate could be restricted by the Core/PCS speed grade. Contact your Altera Sales Representative for the maximum data rate specifications in each speed grade combination offered. For more information about device ordering codes, refer to the *Stratix V Device Family Overview* chapter.

(2) The reference clock common mode voltage is equal to the V_{CCR_GXB} power supply level.

(3) The device cannot tolerate prolonged operation at this absolute maximum.

(4) The differential eye opening specification at the receiver input pins assumes that **Receiver Equalization** is disabled. If you enable **Receiver Equalization**, the receiver circuitry can tolerate a lower minimum eye opening, depending on the equalization level.

(5) The Quartus II software automatically selects the appropriate slew rate depending on the configured data rate or functional mode.

(6) The input reference clock frequency options depend on the data rate and the device speed grade.

Core Performance Specifications

This section describes the clock tree, phase-locked loop (PLL), digital signal processing (DSP), memory blocks, configuration, and JTAG specifications.

Clock Tree Specifications

Table 2–20 lists the clock tree specifications for Stratix V devices.

Table 2–20. Clock Tree Performance for Stratix V Devices—Preliminary

	Performance			Unit
Symbol	–2 Speed Grade	–3 Speed Grade	–4 Speed Grade	
Global and Regional Clock	717	700	500	MHz
Periphery Clock	550	500	500	MHz

PLL Specifications

Table 2–21 lists the Stratix V PLL specifications when operating in both the commercial junction temperature range (0° to 85°C) and the industrial junction temperature range (-40° to 100°C).

Table 2–21. PLL Specifications for Stratix V Devices (Part 1 of 3)—Preliminary *(Note 1)*

Symbol	Parameter	Min	Typ	Max	Unit
f_{IN}	Input clock frequency (–2 speed grade)	5	—	800 *(2)*	MHz
	Input clock frequency (–3 speed grade)	5	—	700 *(2)*	MHz
	Input clock frequency (–4 speed grade)	5	—	650 *(2)*	MHz
f_{INPFD}	Input frequency to the PFD	5	—	325	MHz
f_{FINPFD}	Fractional Input clock frequency to the PFD	50	—	133	MHz
f_{VCO}	PLL VCO operating range (–2 speed grade)	600	—	1600	MHz
	PLL VCO operating range (–3 speed grade)	600	—	1400	MHz
	PLL VCO operating range (–4 speed grade)	600	—	1300	MHz
$t_{EINDUTY}$	Input clock or external feedback clock input duty cycle	40	—	60	%
f_{OUT}	Output frequency for an internal global or regional clock (–2 speed grade)	—	—	717 *(3)*	MHz
	Output frequency for an internal global or regional clock (–3 speed grade)	—	—	700 *(3)*	MHz
	Output frequency for an internal global or regional clock (–4 speed grade)	—	—	500 *(3)*	MHz
f_{OUT_EXT}	Output frequency for an external clock output (–2 speed grade)	—	—	800 *(3)*	MHz
	Output frequency for an external clock output (–3 speed grade)	—	—	667 *(3)*	MHz
	Output frequency for an external clock output (–4 speed grade)	—	—	533 *(3)*	MHz
$t_{OUTDUTY}$	Duty cycle for an external clock output (when set to **50%**)	45	50	55	%
t_{FCOMP}	External feedback clock compensation time	—	—	10	ns

Table 2–21. PLL Specifications for Stratix V Devices (Part 2 of 3)—Preliminary *(Note 1)*

Symbol	Parameter	Min	Typ	Max	Unit
$t_{CONFIGPHASE}$	Time required to reconfigure phase shift	—	TBD *(1)*	—	—
$f_{DYCONFIGCLK}$	Dynamic Configuration Clock	—	—	100	MHz
t_{LOCK}	Time required to lock from the end-of-device configuration or de-assertion of `areset`	—	—	1	ms
t_{DLOCK}	Time required to lock dynamically (after switchover or reconfiguring any non-post-scale counters/delays)	—	—	1	ms
f_{CLBW}	PLL closed-loop low bandwidth	—	0.3	—	MHz
	PLL closed-loop medium bandwidth	—	1.5	—	MHz
	PLL closed-loop high bandwidth *(8)*	—	4	—	MHz
t_{PLL_PSERR}	Accuracy of PLL phase shift	—	—	±50	ps
t_{ARESET}	Minimum pulse width on the `areset` signal	10	—	—	ns
t_{INCCJ} *(4)*, *(5)*	Input clock cycle-to-cycle jitter ($F_{REF} \geq 100$ MHz)	—	0.15	—	UI (p-p)
	Input clock cycle-to-cycle jitter ($F_{REF} < 100$ MHz)	–750	—	+750	ps (p-p)
t_{OUTPJ_DC} *(6)*	Period Jitter for dedicated clock output ($F_{OUT} \geq 100$ MHz)	—	—	TBD *(1)*	ps (p-p)
	Period Jitter for dedicated clock output ($F_{OUT} < 100$ MHz)	—	—	TBD *(1)*	mUI (p-p)
t_{OUTCCJ_DC} *(6)*	Cycle-to-Cycle Jitter for a dedicated clock output ($F_{OUT} \geq 100$ MHz)	—	—	TBD *(1)*	ps (p-p)
	Cycle-to-Cycle Jitter for a dedicated clock output ($F_{OUT} < 100$ MHz)	—	—	TBD *(1)*	mUI (p-p)
t_{OUTPJ_IO} *(6)*, *(9)*	Period Jitter for a clock output on a regular I/O ($F_{OUT} \geq 100$ MHz)	—	—	TBD *(1)*	ps (p-p)
	Period Jitter for a clock output on a regular I/O ($F_{OUT} < 100$ MHz)	—	—	TBD *(1)*	mUI (p-p)
t_{OUTCCJ_IO} *(6)*, *(9)*	Cycle-to-Cycle Jitter for a clock output on a regular I/O ($F_{OUT} \geq 100$ MHz)	—	—	TBD *(1)*	ps (p-p)
	Cycle-to-Cycle Jitter for a clock output on a regular I/O ($F_{OUT} < 100$ MHz)	—	—	TBD *(1)*	mUI (p-p)
$t_{CASC\_OUTPJ\_DC}$ *(6)*, *(7)*	Period Jitter for a dedicated clock output in cascaded PLLs ($F_{OUT} \geq 100$ MHz)	—	—	TBD *(1)*	ps (p-p)
	Period Jitter for a dedicated clock output in cascaded PLLs ($F_{OUT} < 100$ MHz)	—	—	TBD *(1)*	mUI (p-p)
f_{DRIFT}	Frequency drift after PFDENA is disabled for a duration of 100 µs	—	—	±10	%
dK_{BIT}	Bit number of Delta Sigma Modulator (DSM)	—	24	—	Bits
k_{VALUE}	Numerator of Fraction	—	8388608	—	—

Table 2–21. PLL Specifications for Stratix V Devices (Part 3 of 3)—Preliminary *(Note 1)*

Symbol	Parameter	Min	Typ	Max	Unit
f_{RES}	Resolution of VCO frequency (f_{INPFD} = 100 MHz)	—	5.96	—	Hz

Notes to Table 2–21:

(1) Pending silicon characterization.

(2) This specification is limited in the Quartus II software by the I/O maximum frequency. The maximum I/O frequency is different for each I/O standard.

(3) This specification is limited by the lower of the two: I/O F_{MAX} or F_{OUT} of the PLL.

(4) A high input jitter directly affects the PLL output jitter. To have low PLL output clock jitter, you must provide a clean clock source < 120 ps.

(5) F_{REF} is fIN/N when N = 1.

(6) Peak-to-peak jitter with a probability level of 10^{-12} (14 sigma, 99.99999999974404% confidence level). The output jitter specification applies to the intrinsic jitter of the PLL, when an input jitter of 30 ps is applied. The external memory interface clock output jitter specifications use a different measurement method and are available in Table 2–33 on page 2–28.

(7) The cascaded PLL specification is only applicable with the following condition:
a. Upstream PLL: 0.59Mhz ≤ Upstream PLL BW < 1 MHz
b. Downstream PLL: Downstream PLL BW > 2 MHz

(8) High bandwidth PLL settings are not supported in external feedback mode.

(9) The external memory interface clock output jitter specifications use a different measurement method, which is available in Table 2–31 on page 2–27.

DSP Block Specifications

Table 2–22 lists the Stratix V DSP block performance specifications.

Table 2–22. Block Performance Specifications for Stratix V DSP Devices—Preliminary *(Note 1)* (Part 1 of 2)

Mode	Performance			Unit
	–2 Speed Grade	–3 Speed Grade	–4 Speed Grade	
Modes using One DSP				
Three 9 × 9	600	480	420	MHz
One 18 × 18	600	480	420	MHz
Two partial 18 × 18 (or 16 × 16)	600	480	420	MHz
One 27 × 27	450	360	315	MHz
One 36 × 18	450	360	315	MHz
One sum of two 18 × 18 (One sum of two 16 × 16)	500	400	350	MHz
One sum of square	450	360	315	MHz
One 18 × 18 plus 36 (a × b) + c	500	400	350	MHz
Modes using Two DSPs				
Three 18 × 18	500	400	350	MHz
One sum of four 18 × 18	425	340	298	MHz
One sum of two 27 × 27	425	340	298	MHz
One sum of two 36 × 18	425	340	298	MHz
One complex 18 × 18	500	400	350	MHz
One 36 × 36	400	320	280	MHz
Modes using Three DSPs				
One complex 18 × 25	350	280	245	MHz

Table 2–22. Block Performance Specifications for Stratix V DSP Devices—Preliminary *(Note 1)* (Part 2 of 2)

Mode	Performance			Unit
	–2 Speed Grade	–3 Speed Grade	–4 Speed Grade	
Modes using Four DSPs				
One complex 27 × 27	425	340	298	MHz

Note to Table 2–22:

(1) These numbers are preliminary pending silicon characterization.

Memory Block Specifications

Table 2–23 lists the Stratix V memory block specifications.

Table 2–23. Memory Block Performance Specifications for Stratix V Devices—Preliminary *(Note 1)*, *(2)*, *(3)*

Memory	Mode	Resources Used		Performance			Unit
		ALUTs	Memory	C2 Speed Grade	C3,I3 Speed Grade	C4,I4 Speed Grade	
MLAB	Single port, all supported widths	0	1	600	500	450	MHz
	Simple dual-port, all supported widths	0	1	450	375	300	MHz
	ROM, all supported widths	0	1	600	500	450	MHz
M20K Block	Single-port, all supported widths	0	1	600	500	450	MHz
	Simple dual-port, all supported widths	0	1	600	500	450	MHz
	Simple dual-port with the read-during-write option set to **Old Data**, all supported widths	0	1	525	455	400	MHz
	Simple dual-port with ECC enabled, 512 × 32	0	1	450	400	350	MHz
	Simple dual-port with ECC and optional pipeline registers enabled, 512 × 32	0	1	600	500	450	MHz
	True dual port, all supported widths	0	1	600	500	450	MHz
	ROM, all supported widths	0	1	600	500	450	MHz
	Min Pulse Width (clock high time)	—	—	750	800	850	ps
	Min Pulse Width (clock low time)	—	—	500	625	690	ps

Notes to Table 2–23:

(1) These numbers are preliminary pending silicon characterization.

(2) To achieve the maximum memory block performance, use a memory block clock that comes through global clock routing from an on-chip PLL set to **50%** output duty cycle. Use the Quartus II software to report timing for this and other memory block clocking schemes.

(3) When you use the error detection cyclical redundancy check (CRC) feature, there is no degradation in F_{MAX}.

JTAG Configuration Specifications

Table 2–24 lists the JTAG timing parameters and values for Stratix V devices.

Table 2–24. JTAG Timing Parameters and Values for Stratix V Devices—Preliminary *(Note 1)*

Symbol	Description	Min	Max	Unit
t_{JCP}	TCK clock period	30	—	ns
t_{JCH}	TCK clock high time	14	—	ns
t_{JCL}	TCK clock low time	14	—	ns
$t_{JPSU (TDI)}$	TDI JTAG port setup time	2	—	ns
$t_{JPSU (TMS)}$	TMS JTAG port setup time	3	—	ns
t_{JPH}	JTAG port hold time	5	—	ns
t_{JPCO}	JTAG port clock to output	—	11 *(2)*	ns
t_{JPZX}	JTAG port high impedance to valid output	—	14 *(2)*	ns
t_{JPXZ}	JTAG port valid output to high impedance	—	14 *(2)*	ns

Notes to Table 2–24:

(1) These numbers are preliminary pending silicon characterization.

(2) A 1 ns adder is required for each V_{CCIO} voltage step down from 3.0 V. For example, t_{JPCO} = 12 ns if V_{CCIO} of the TDO I/O bank = 2.5 V, or 13 ns if it equals 1.8 V.

Temperature Sensing Diode Specifications

Table 2–25 lists the specifications for the Stratix V temperature sensing diode.

Table 2–25. External Temperature Sensing Diode Specifications for Stratix V Devices—Preliminary

Description	Min	Typ	Max	Unit
I_{bias}, diode source current	8	—	200	μA
$V_{bias,}$ voltage across diode	0.3	—	0.9	V
Series resistance	—	—	< 5	Ω
Diode ideality factor	—	—	1.030	—

Periphery Performance

This section describes periphery performance, including high-speed I/O and external memory interface.

I/O performance supports several system interfaces, such as the **LVDS** high-speed I/O interface, external memory interface, and the **PCI/PCI-X** bus interface. General-purpose I/O standards such as 3.3-, 2.5-, 1.8-, and 1.5-**LVTTL/LVCMOS** are capable of typical 167 MHz and 1.2 **LVCMOS** at 100 MHz interfacing frequency with 10 pF load.

☞ Actual achievable frequency depends on design- and system-specific factors. You must perform HSPICE/IBIS simulations based on your specific design and system setup to determine the maximum achievable frequency in your system.

High-Speed I/O Specification

Table 2–26 lists high-speed I/O timing for Stratix V devices.

Table 2–26. High-Speed I/O Specifications for Stratix V Devices—Preliminary *(Note 1), (2), (3)* **(Part 1 of 2)**

Symbol	Conditions	–2 Speed Grade			–3 Speed Grade			–4 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
f_{HSCLK_in} (input clock frequency) True Differential I/O Standards	Clock boost factor W = 1 to 40 *(5)*	5	—	717	5	—	625	5	—	525	MHz
f_{HSCLK_in} (input clock frequency) Single Ended I/O Standards *(4)*	Clock boost factor W = 1 to 40 *(5)*	5	—	717	5	—	625	5	—	525	MHz
f_{HSCLK_in} (input clock frequency) Single Ended I/O Standards *(3)*	Clock boost factor W = 1 to 40 *(5)*	5	—	520	5	—	420	5	—	420	MHz
f_{HSCLK_OUT} (output clock frequency)	—	5	—	717 *(6)*	5	—	625 *(6)*	5	—	525 *(6)*	MHz
Transmitter											
True Differential I/O Standards - f_{HSDR} (data rate)	SERDES factor J = 3 to 10 *(10)*	*(7)*	—	1434	*(7)*	—	1250	*(7)*	—	1050	Mbps
	SERDES factor J = 2, Uses DDR Registers	*(7)*	—	*(7)*	*(7)*	—	*(7)*	*(7)*	—	*(7)*	Mbps
	SERDES factor J = 1, Uses SDR Register	*(7)*	—	*(7)*	*(7)*	—	*(7)*	*(7)*	—	*(7)*	Mbps
Emulated Differential I/O Standards with Three External Output Resistor Networks - f_{HSDR} (data rate) *(8)*	SERDES factor J = 4 to 10	*(7)*	—	1100	*(7)*	—	840	*(7)*	—	840	Mbps
$t_{x\ Jitter}$ - True Differential I/O Standards	Total Jitter for Data Rate 600 Mbps - 1.6 Gbps	—	—	160	—	—	160	—	—	160	ps
	Total Jitter for Data Rate < 600 Mbps	—	—	0.1	—	—	0.1	—	—	0.1	UI
$t_{x\ Jitter}$ - Emulated Differential I/O Standards with Three External Output Resistor Network	Total Jitter for Data Rate 600 Mbps - 1.25 Gbps	—	—	300	—	—	300	—	—	325	ps
	Total Jitter for Data Rate < 600 Mbps	—	—	0.2	—	—	0.2	—	—	0.25	UI
t_{DUTY}	Transmitter output clock duty cycle for both True and Emulated Differential I/O Standards	45	50	55	45	50	55	45	50	55	%

Table 2–26. High-Speed I/O Specifications for Stratix V Devices—Preliminary *(Note 1), (2), (3)* **(Part 2 of 2)**

Symbol	Conditions	–2 Speed Grade			–3 Speed Grade			–4 Speed Grade			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
t_{RISE} & t_{FALL}	True Differential I/O Standards	—	—	160	—	—	200	—	—	200	ps
	Emulated Differential I/O Standards with Three External Output Resistor Networks	—	—	250	—	—	250	—	—	300	ps
TCCS	True Differential I/O Standards	—	—	150	—	—	150	—	—	150	ps
	Emulated Differential I/O Standards	—	—	300	—	—	300	—	—	300	ps
Receiver											
True Differential I/O Standards - $f_{HSDRDPA}$ (data rate)	SERDES factor J = 3 to 10	150	—	1434	150	—	1250	150	—	1050	Mbps
f_{HSDR} (data rate)	SERDES factor J = 3 to 10	*(7)*	—	*(9)*	*(7)*	—	*(9)*	*(7)*	—	*(9)*	Mbps
	SERDES factor J = 2, Uses DDR Registers	*(7)*	—	*(7)*	*(7)*	—	*(7)*	*(7)*	—	*(7)*	Mbps
	SERDES factor J = 1, Uses SDR Register	*(7)*	—	*(7)*	*(7)*	—	*(7)*	*(7)*	—	*(7)*	Mbps
DPA Mode											
DPA run length	—	—	—	10000	—	—	10000	—	—	10000	UI
Soft CDR mode											
Soft-CDR PPM tolerance	—	—	—	300	—	—	300	—	—	300	± PPM
Non DPA Mode											
Sampling Window	—	—	—	300	—	—	300	—	—	300	ps

Notes to Table 2–26:

(1) When J = 3 to 10, use the serializer/deserializer (SERDES) block.

(2) When J = 1 or 2, bypass the SERDES block.

(3) This only applies to **LVDS** source synchronous mode.

(4) This only applies to DPA and soft-CDR modes.

(5) Clock Boost Factor (W) is the ratio between the input data rate to the input clock rate.

(6) This is achieved by using the **LVDS** clock network.

(7) The minimum specification depends on the clock source (for example, the PLL and clock pin) and the clock routing resource (global, regional, or local) that you use. The I/O differential buffer and input register do not have a minimum toggle rate.

(8) You must calculate the leftover timing margin in the receiver by performing link timing closure analysis. You must consider the board skew margin, transmitter channel-to-channel skew, and receiver sampling margin to determine the leftover timing margin.

(9) You can estimate the achievable maximum data rate for non-DPA mode by performing link timing closure analysis. You must consider the board skew margin, transmitter delay margin, and receiver sampling margin to determine the maximum data rate supported.

(10) If the receiver with DPA enabled and transmitter are using shared PLLs, the minimum data rate is 150 Mbps.

Figure 2–1 shows the dynamic phase alignment (DPA) lock time specifications with the DPA PLL calibration option enabled.

Figure 2–1. DPA Lock Time Specification with DPA PLL Calibration Enabled

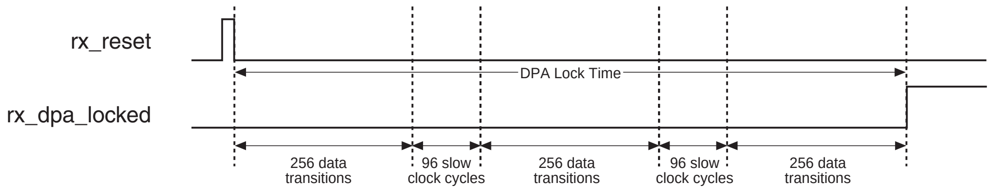


Table 2–27 lists the DPA lock time specifications for Stratix V GX devices.

Table 2–27. DPA Lock Time Specifications for Stratix V GX Devices Only—Preliminary *(Note 1)*, *(2)*, *(3)*

Standard	Training Pattern	Number of Data Transitions in One Repetition of the Training Pattern	Number of Repetitions per 256 Data Transitions *(4)*	Maximum
SPI-4	00000000001111111111	2	128	640 data transitions
Parallel Rapid I/O	00001111	2	128	640 data transitions
	10010000	4	64	640 data transitions
Miscellaneous	10101010	8	32	640 data transitions
	01010101	8	32	640 data transitions

Notes to Table 2–27:

(1) The DPA lock time is for one channel.

(2) One data transition is defined as a 0-to-1 or 1-to-0 transition.

(3) The DPA lock time stated in this table applies to both commercial and industrial grade.

(4) This is the number of repetitions for the stated training pattern to achieve the 256 data transitions.

Figure 2–2 shows the **LVDS** soft-clock data recovery (CDR)/DPA sinusoidal jitter tolerance specification for a data rate equal to or higher than 1.25 Gbps. Table 2–28 lists the **LVDS** soft-CDR/DPA sinusoidal jitter tolerance specification for a data rate equal to or higher than 1.25 Gbps.

Figure 2–2. LVDS Soft-CDR/DPA Sinusoidal Jitter Tolerance Specification for a Data Rate Equal to or Higher Than 1.25 Gbps

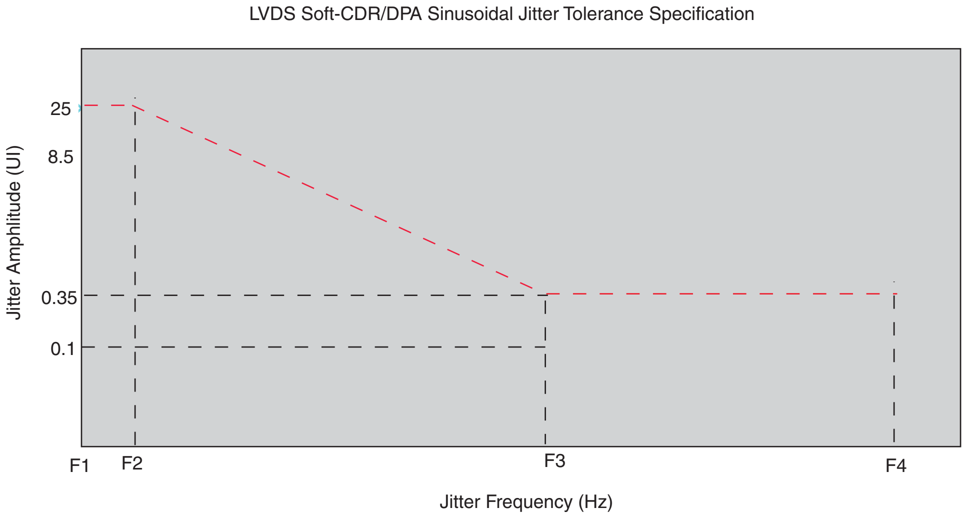
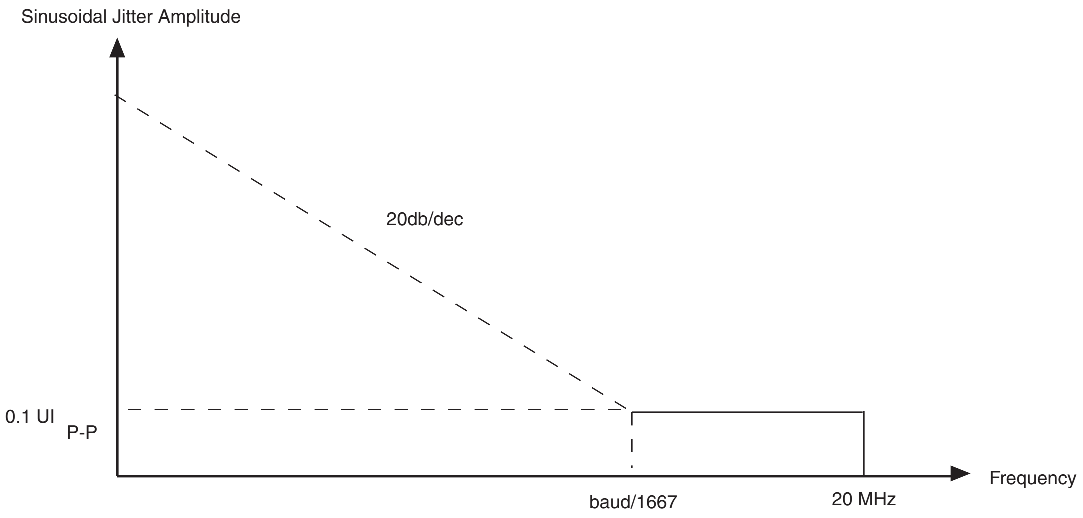


Table 2–28. LVDS Soft-CDR/DPA Sinusoidal Jitter Mask Values for a Data Rate Equal to or Higher than 1.25 Gbps—Preliminary

Jitter Frequency (Hz)		Sinusoidal Jitter (UI)
F1	10,000	25.000
F2	17,565	25.000
F3	1,493,000	0.350
F4	50,000,000	0.350

Figure 2–3 shows the **LVDS** soft-CDR/DPA sinusoidal jitter tolerance specification for a data rate less than 1.25 Gbps.

Figure 2–3. LVDS Soft-CDR/DPA Sinusoidal Jitter Tolerance Specification for a Data Rate Less than 1.25 Gbps



DQ Logic Block and Memory Output Clock Jitter Specifications

Table 2–29 lists the DQS phase offset delay per stage for Stratix V devices.

Table 2–29. DQS Phase Offset Delay Per Setting for Stratix V Devices—Preliminary *(Note 1)*, *(2)*, *(3)*

Speed Grade	Min	Max	Unit
–2	7	13	ps
–3	7	15	ps
–4	7	16	ps

Notes to Table 2–29:

(1) These numbers are preliminary pending silicon characterization.

(2) The typical value equals the average of the minimum and maximum values.

(3) The delay settings are linear with a cumulative delay variation of 40 ps for all speed grades. For example, when using a –2 speed grade and applying a 10-phase offset setting to a 90° phase shift at 400 MHz, the expected average cumulative delay is [625 ps + (10 × 10 ps) ± 20 ps] = 725 ps ± 20 ps.

Table 2–30 lists the DQS phase shift error for Stratix V devices.

Table 2–30. DQS Phase Shift Error Specification for DLL-Delayed Clock (t_{DQS_PSERR}) for Stratix V Devices—Preliminary *(Note 1), (2)*

Number of DQS Delay Buffers	–2 Speed Grade	–3 Speed Grade	–4 Speed Grade	Unit
1	26	28	30	ps
2	52	56	60	ps
3	78	84	90	ps
4	104	112	120	ps

Notes to Table 2–30:

(1) The numbers are preliminary pending silicon characterization.

(2) This error specification is the absolute maximum and minimum error. For example, skew on three DQS delay buffers in a –2 speed grade is ±78 ps or ±39 ps.

Table 2–31 lists the memory output clock jitter specifications for Stratix V devices.

Table 2–31. Memory Output Clock Jitter Specification for Stratix V Devices—Preliminary *(Note 1), (2), (3)*

Parameter	Clock Network	Symbol	–2 Speed Grade		–3 Speed Grade		–4 Speed Grade		Unit
			Min	Max	Min	Max	Min	Max	
Clock period jitter	Regional	$t_{JIT(per)}$	-50	50	-55	55	-55	55	ps
Cycle-to-cycle period jitter	Regional	$t_{JIT(cc)}$	-100	100	-110	110	-110	110	ps
Duty cycle jitter	Regional	$t_{JIT(duty)}$	-50	50	-82.5	82.5	-82.5	82.5	ps
Clock period jitter	Global	$t_{JIT(per)}$	-75	75	-82.5	82.5	-82.5	82.5	ps
Cycle-to-cycle period jitter	Global	$t_{JIT(cc)}$	-150	150	-165	165	-165	165	ps
Duty cycle jitter	Global	$t_{JIT(duty)}$	-75	75	-90	90	-90	90	ps

Notes to Table 2–31:

(1) The numbers are preliminary pending silicon characterization.

(2) The memory output clock jitter measurements are for 200 consecutive clock cycles, as specified in the JEDEC DDR2/DDR3 SDRAM standard.

(3) The clock jitter specification applies to the memory output clock pins generated using differential signal-splitter and DDIO circuits clocked by a PLL output routed on a regional or global clock network as specified. Altera recommends using regional clock networks whenever possible.

OCT Calibration Block Specifications

Table 2–32 lists the OCT calibration block specifications for Stratix V devices.

Table 2–32. OCT Calibration Block Specifications for Stratix V Devices—Preliminary ***(Note 1)***

Symbol	Description	Min	Typ	Max	Unit
OCTUSRCLK	Clock required by the OCT calibration blocks	—	—	20	MHz
T_{OCTCAL}	Number of OCTUSRCLK clock cycles required for OCT R_S/R_T calibration	—	1000	—	Cycles
$T_{OCTSHIFT}$	Number of OCTUSRCLK clock cycles required for the OCT code to shift out	—	32	—	Cycles
T_{RS_RT}	Time required between the `dyn_term_ctrl` and `oe` signal transitions in a bidirectional I/O buffer to dynamically switch between OCT R_S and R_T	—	2.5	—	ns

Note to Table 2–32:

(1) Pending silicon characterization.

Duty Cycle Distortion (DCD) Specifications

Table 2–33 lists the worst-case DCD for Stratix V devices.

Table 2–33. Worst-Case DCD on Stratix V I/O Pins—Preliminary ***(Note 1)***

Symbol	–2 Speed Grade		–3 Speed Grade		–4 Speed Grade		Unit
	Min	Max	Min	Max	Min	Max	
Output Duty Cycle	45	55	45	55	45	55	%

Note to Table 2–33:

(1) The numbers are preliminary pending silicon characterization.

I/O Timing

Altera offers two ways to determine I/O timing—the Excel-based I/O Timing and the Quartus II Timing Analyzer.

Excel-based I/O timing provides pin timing performance for each device density and speed grade. The data is typically used prior to designing the FPGA to get an estimate of the timing budget as part of the link timing analysis. The Quartus II Timing Analyzer provides a more accurate and precise I/O timing data based on the specifics of the design after you complete place-and-route.

You can download the Excel-based I/O Timing spreadsheet from the Stratix V Devices Literature webpage.

Programmable IOE Delay

Table 2–34 lists the Stratix V IOE programmable delay settings.

Table 2–34. IOE Programmable Delay for Stratix V Devices—Preliminary ***(Note 1)***

Parameter *(2)*	Available Settings	Min Offset *(3)*	Fast Model		Slow Model					
			Industrial	Commercial	C2	C3	C4	I3	I4	Unit
D1	63	0	0.471	0.514	0.800	0.843	0.918	0.850	0.924	ns
D2	31	0	0.274	0.274	0.423	0.456	0.501	0.453	0.498	ns
D3	7	0	1.668	1.735	2.830	2.985	3.252	3.007	3.274	ns
D5	63	0	0.493	0.474	0.835	0.882	0.960	0.888	0.966	ns
D6	31	0	0.273	0.258	0.463	0.488	0.532	0.492	0.536	ns

Notes to Table 2–34:

(1) Pending the Quartus II software extraction.

(2) You can set this value in the Quartus II software by selecting **D1**, **D2**, **D3**,**D5**, and **D6** in the **Assignment Name** column.

(3) Minimum offset does not include the intrinsic delay.

Programmable Output Buffer Delay

Table 2–35 lists the delay chain settings that control the rising and falling edge delays of the output buffer. The default delay is 0 ps.

Table 2–35. Programmable Output Buffer Delay for Stratix V Devices—Preliminary ***(Note 1)***, ***(2)***

Symbol	Parameter	Typical	Unit
D_{OUTBUF}	Rising and/or falling edge delay	0 (default)	ps
		50	ps
		100	ps
		150	ps

Notes to Table 2–35:

(1) Pending the Quartus II software extraction.

(2) You can set the programmable output buffer delay in the Quartus II software by setting the **Output Buffer Delay Control** assignment to either positive, negative, or both edges, with the specific values stated here (in ps) for the **Output Buffer Delay** assignment.

Glossary

Table 2–36 lists the glossary for this chapter.

Table 2–36. Glossary (Part 1 of 4)

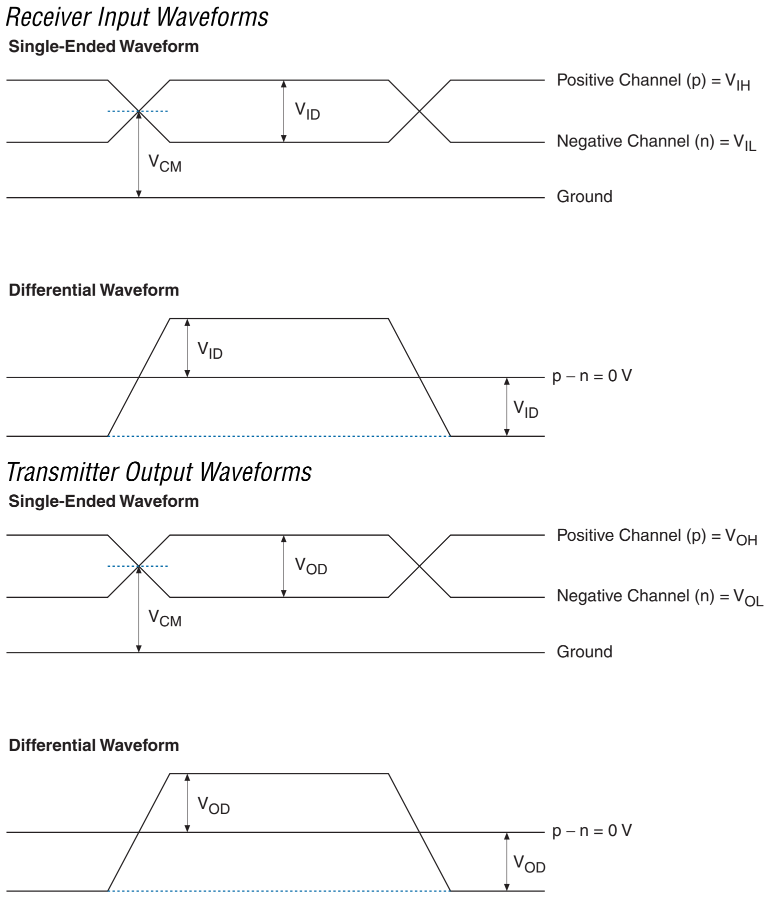
Letter	Subject	Definitions
A B C	—	—
D	Differential I/O Standards	 Receiver Input Waveforms Single-Ended Waveform Positive Channel (p) = V_{IH} V_{ID} Negative Channel (n) = V_{IL} V_{CM} Ground Differential Waveform V_{ID} p − n = 0 V V_{ID} Transmitter Output Waveforms Single-Ended Waveform Positive Channel (p) = V_{OH} V_{OD} Negative Channel (n) = V_{OL} V_{CM} Ground Differential Waveform V_{OD} p − n = 0 V V_{OD}
E	—	—
F	f_{HSCLK}	Left and right PLL input clock frequency.
	f_{HSDR}	High-speed I/O block—Maximum and minimum **LVDS** data transfer rate (f_{HSDR} = 1/TUI), non-DPA.
	$f_{HSDRDPA}$	High-speed I/O block—Maximum and minimum **LVDS** data transfer rate ($f_{HSDRDPA}$ = 1/TUI), DPA.
G H I	—	—

Table 2–36. Glossary (Part 2 of 4)

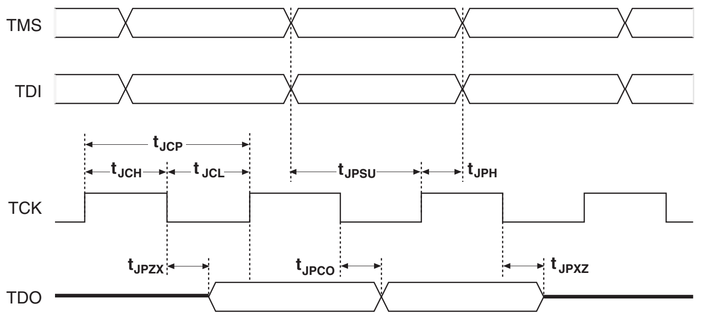
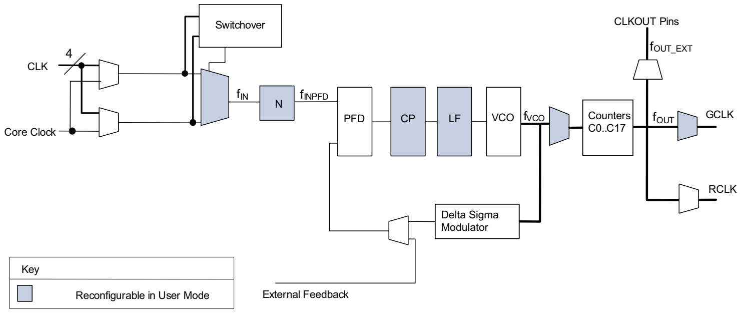
Letter	Subject	Definitions
J	J	High-speed I/O block—Deserialization factor (width of parallel data bus).
	JTAG Timing Specifications	JTAG Timing Specifications:  TMS TDI TCK TDO t_{JCP} t_{JCH} t_{JCL} t_{JPSU} t_{JPH} t_{JPZX} t_{JPCO} t_{JPXZ}
K L M N O	—	—
P	**PLL Specifications**	**Diagram of PLL Specifications** *(1)*  Switchover CLKOUT Pins f_{OUT_EXT} CLK 4 Core Clock f_{IN} N f_{INPFD} PFD CP LF VCO f_{VCO} Counters C0..C17 f_{OUT} GCLK RCLK Delta Sigma Modulator Key Reconfigurable in User Mode External Feedback **Note:** (1) `Core Clock` can only be fed by dedicated clock input pins or PLL outputs.
Q	—	—
R	R_L	Receiver differential input discrete resistor (external to the Stratix V device).

Table 2–36. Glossary (Part 3 of 4)

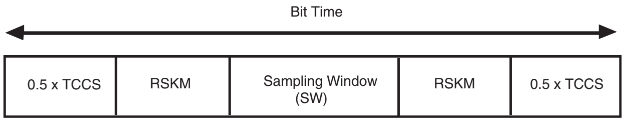
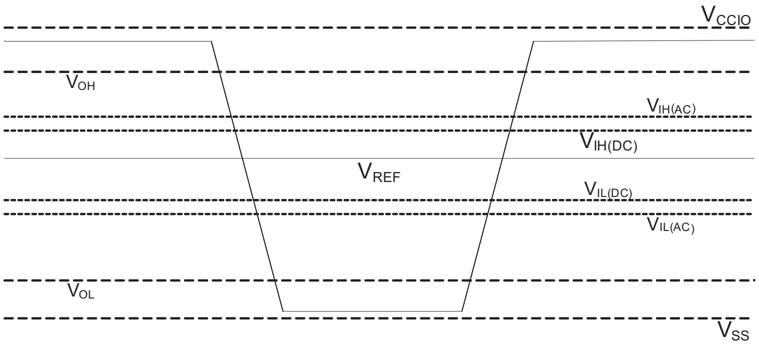
Letter	Subject	Definitions
S	**SW (sampling window)**	Timing Diagram—the period of time during which the data must be valid in order to capture it correctly. The setup and hold times determine the ideal strobe position within the sampling window, as shown:  Bit Time 0.5 x TCCS RSKM Sampling Window (SW) RSKM 0.5 x TCCS
	Single-ended voltage referenced I/O standard	The JEDEC standard for **SSTl** and **HSTL** I/O defines both the AC and DC input signal values. The AC values indicate the voltage levels at which the receiver must meet its timing specifications. The DC values indicate the voltage levels at which the final logic state of the receiver is unambiguously defined. After the receiver input has crossed the AC value, the receiver changes to the new logic state. The new logic state is then maintained as long as the input stays beyond the AC threshold. This approach is intended to provide predictable receiver timing in the presence of input waveform ringing, as shown: *Single-Ended Voltage Referenced I/O Standard*  V_{CCIO} V_{OH} $V_{IH(AC)}$ $V_{IH(DC)}$ V_{REF} $V_{IL(DC)}$ $V_{IL(AC)}$ V_{OL} V_{SS}
T	t_C	High-speed receiver and transmitter input and output clock period.
	TCCS (channel-to-channel-skew)	The timing difference between the fastest and 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 (refer to the *Timing Diagram* figure under **SW** in this table).
	t_{DUTY}	High-speed I/O block—Duty cycle on the high-speed transmitter output clock. **Timing Unit Interval (TUI)** The timing budget allowed for skew, propagation delays, and the data sampling window. (TUI = 1/(receiver input clock frequency multiplication factor) = t_C/w)
	t_{FALL}	Signal high-to-low transition time (80-20%)
	t_{INCCJ}	Cycle-to-cycle jitter tolerance on the PLL clock input.
	t_{OUTPJ_IO}	Period jitter on the general purpose I/O driven by a PLL.
	t_{OUTPJ_DC}	Period jitter on the dedicated clock output driven by a PLL.
	t_{RISE}	Signal low-to-high transition time (20-80%)
U	—	—

Table 2–36. Glossary (Part 4 of 4)

Letter	Subject	Definitions
V	$V_{CM(DC)}$	DC common mode input voltage.
	V_{ICM}	Input common mode voltage—The common mode of the differential signal at the receiver.
	V_{ID}	Input differential voltage swing—The difference in voltage between the positive and complementary conductors of a differential transmission at the receiver.
	$V_{DIF(AC)}$	AC differential input voltage—Minimum AC input differential voltage required for switching.
	$V_{DIF(DC)}$	DC differential input voltage— Minimum DC input differential voltage required for switching.
	V_{IH}	Voltage input high—The minimum positive voltage applied to the input which is accepted by the device as a logic high.
	$V_{IH(AC)}$	High-level AC input voltage
	$V_{IH(DC)}$	High-level DC input voltage
	V_{IL}	Voltage input low—The maximum positive voltage applied to the input which is accepted by the device as a logic low.
	$V_{IL(AC)}$	Low-level AC input voltage
	$V_{IL(DC)}$	Low-level DC input voltage
	V_{OCM}	Output common mode voltage—The common mode of the differential signal at the transmitter.
	V_{OD}	Output differential voltage swing—The difference in voltage between the positive and complementary conductors of a differential transmission at the transmitter.
	V_{SWING}	Differential input voltage
	V_X	Input differential cross point voltage
	V_{OX}	Output differential cross point voltage
W	W	High-speed I/O block—clock boost factor
X Y Z	—	—

Document Revision History

Table 2–37 lists the revision history for this chapter.

Table 2–37. Document Revision History

Date	Version	Changes
May 2011	2.0	■ Updated Table 2–4, Table 2–18, Table 2–19, Table 2–21, Table 2–22, Table 2–23, and Table 2–24. ■ Updated the "DQ Logic Block and Memory Output Clock Jitter Specifications" title. ■ Chapter moved to Volume 1. ■ Minor text edits.
December 2010	1.1	■ Updated Table 1–2, Table 1–4, Table 1–19, and Table 1–23. ■ Converted chapter to the new template. ■ Minor text edits.
July 2010	1.0	Initial release.

Additional Information

This chapter provides additional information about the document and Altera.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (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 to Table:

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

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate 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, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *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** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the 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`. The suffix `n` denotes an active-low signal. 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`).
↵	An angled arrow instructs you to press the Enter key.
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 question mark directs you to a software help system with related information.
👣	The feet direct you to another document or website with related information.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚠ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
✉	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.

Stratix V Device Handbook

Volume 2: Device Interfaces and Integration

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

SV5V1-1.3
11.0

Contents

Section I. Device Core

Chapter 1. Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices

Chapter 2. Memory Blocks in Stratix V Devices

Chapter 3. Variable Precision DSP Blocks in Stratix V Devices

Chapter 4. Clock Networks and PLLs in Stratix V Devices

Section II. I/O Interfaces

Chapter 5. I/O Features in Stratix V Devices

Chapter 6. High-Speed Differential I/O Interfaces and DPA in Stratix V Devices

Chapter 7. External Memory Interfaces in Stratix V Devices

Section III. System Integration

Chapter 8. Hot Socketing and Power-On Reset in Stratix V Devices

Chapter 9. Configuration, Design Security, and Remote System Upgrades in Stratix V Devices

Chapter 10. SEU Mitigation in Stratix V Devices

Chapter 11. JTAG Boundary-Scan Testing in Stratix V Devices

Chapter 12. Power Management in Stratix V Devices

Additional Information

Chapter Revision Dates

The chapters in this document, Stratix V Device Handbook Volume 2: Device Interfaces and Integration, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51002-1.2*

Chapter 2. Memory Blocks in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51003-1.2*

Chapter 3. Variable Precision DSP Blocks in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51004-1.2*

Chapter 4. Clock Networks and PLLs in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51005-1.2*

Chapter 5. I/O Features in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51006-1.3*

Chapter 6. High-Speed Differential I/O Interfaces and DPA in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51007-1.2*

Chapter 7. External Memory Interfaces in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51008-1.2*

Chapter 8. Hot Socketing and Power-On Reset in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51009-1.2*

Chapter 9. Configuration, Design Security, and Remote System Upgrades in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51010-1.3*

Chapter 10. SEU Mitigation in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51011-1.2*

Chapter 11. JTAG Boundary-Scan Testing in Stratix V Devices
Revised: *May 2011*
Part Number: *SV51012-1.2*

Chapter 12. Power Management in Stratix V Devices

Revised: *May 2011*
Part Number: *SV51013-1.2*

Section I. Device Core

This section describes the Stratix® V device family core, which is the most architecturally advanced, high-performance, low-power FPGA in the market place. This section includes the following chapters:

- Chapter 1, Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices
- Chapter 2, Memory Blocks in Stratix V Devices
- Chapter 3, Variable Precision DSP Blocks in Stratix V Devices
- Chapter 4, Clock Networks and PLLs in Stratix V 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. Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices

SV51002-1.2

This chapter describes the features of the logic array blocks (LABs) in the Stratix® V core fabric. LABs are made up of adaptive logic modules (ALMs) that you can configure to implement logic functions, arithmetic functions, and register functions. LABs and ALMs are the basic building blocks of the Stratix V device. ALMs provide advanced features with efficient logic utilization and are completely backward-compatible.

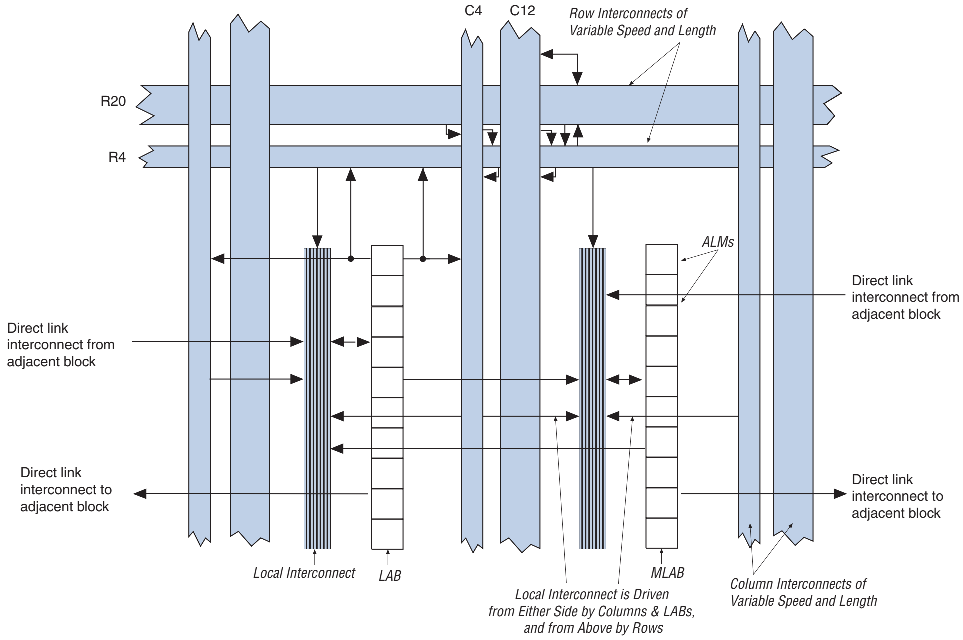
This chapter contains the following sections:

- "Logic Array Blocks" on page 1–1
- "Adaptive Logic Modules" on page 1–4

Logic Array Blocks

Each LAB consists of ten ALMs, various carry chains, shared arithmetic chains, LAB Stratix V control signals, a local interconnect, and register chain connection lines. The local interconnect transfers signals between ALMs in the same LAB. The direct link interconnect enables 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 same LAB or adjacent LABs, allowing the use of local, shared arithmetic chain, and register chain connections for performance and area efficiency. Figure 1–1 shows the Stratix V LAB structure and the LAB interconnects.

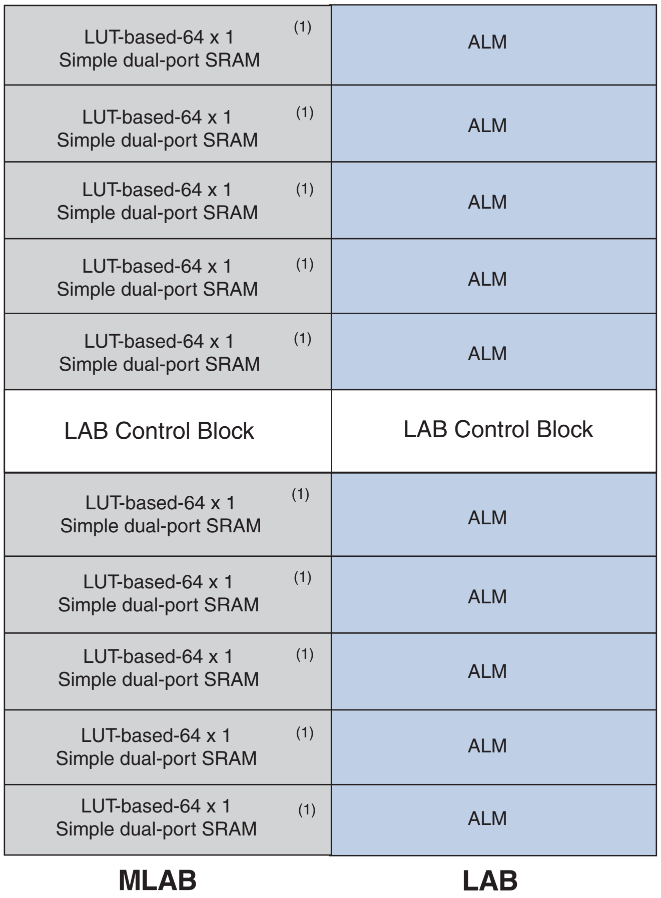
Figure 1–1. LAB Structure for Stratix V Devices


C4
C12
Row Interconnects of Variable Speed and Length
R20
R4
ALMs
Direct link interconnect from adjacent block
Direct link interconnect from adjacent block
Direct link interconnect to adjacent block
Direct link interconnect to adjacent block
Local Interconnect
LAB
MLAB
Local Interconnect is Driven from Either Side by Columns & LABs, and from Above by Rows
Column Interconnects of Variable Speed and Length

The memory LAB (MLAB) is a derivative of the Stratix V LAB. The MLAB adds look-up table (LUT)-based SRAM capability to the LAB, as shown in Figure 1–2. 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 V families. The MLAB is a superset of the LAB and includes all LAB features.

For more information about MLABs, refer to the *Memory Blocks in Stratix V Devices* chapter.

Figure 1–2. LAB and MLAB Structure for Stratix V Devices


LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LAB Control Block
LAB Control Block
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
LUT-based-64 x 1
Simple dual-port SRAM (1)
ALM
MLAB
LAB

Note to Figure 1–2:

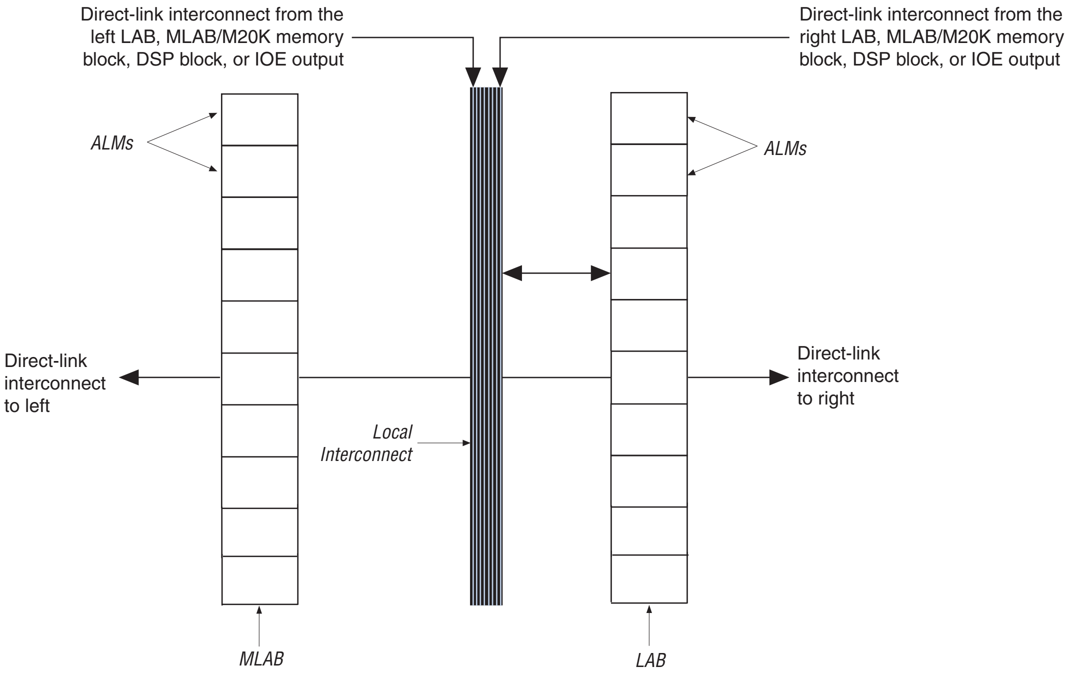
(1) You can use the MLAB ALM as a regular LAB ALM or configure it as a dual-port SRAM, as shown.

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, M20K blocks, or digital signal processing (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 1–3 shows the direct-link connection.

Figure 1–3. Direct-Link Connection


Direct-link interconnect from the left LAB, MLAB/M20K memory block, DSP block, or IOE output
Direct-link interconnect from the right LAB, MLAB/M20K memory block, DSP block, or IOE output
ALMs
ALMs
Direct-link interconnect to left
Direct-link interconnect to right
Local Interconnect
MLAB
LAB

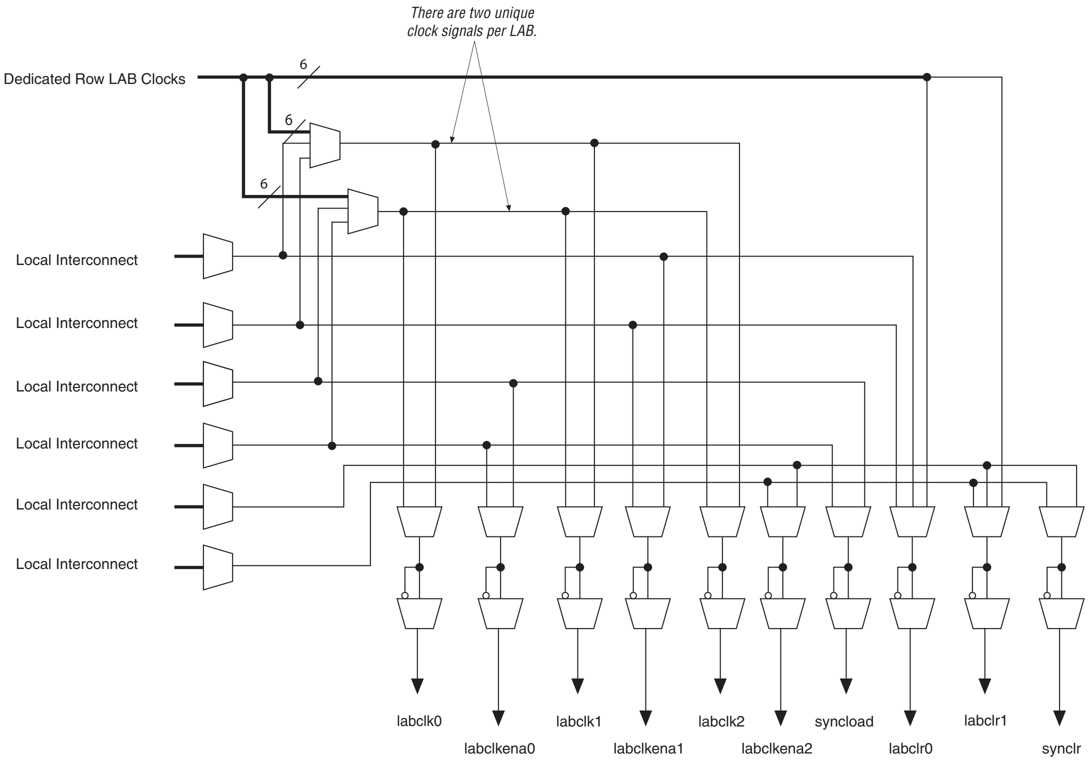
LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its ALMs. The control signals include three clocks, three clock enables, two asynchronous clears, a synchronous clear, and a synchronous load, for 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 1–4. 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. Deasserting 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. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different lengths and speeds used for inter- and intra-design block connectivity.

Figure 1–4. LAB-Wide Control Signals



Adaptive Logic Modules

The ALM is the basic building block of logic in the Stratix V 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 four 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 four 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 1–5 shows a high-level block diagram of the Stratix V ALM.

Figure 1–5. High-Level Block Diagram of the Stratix V ALM

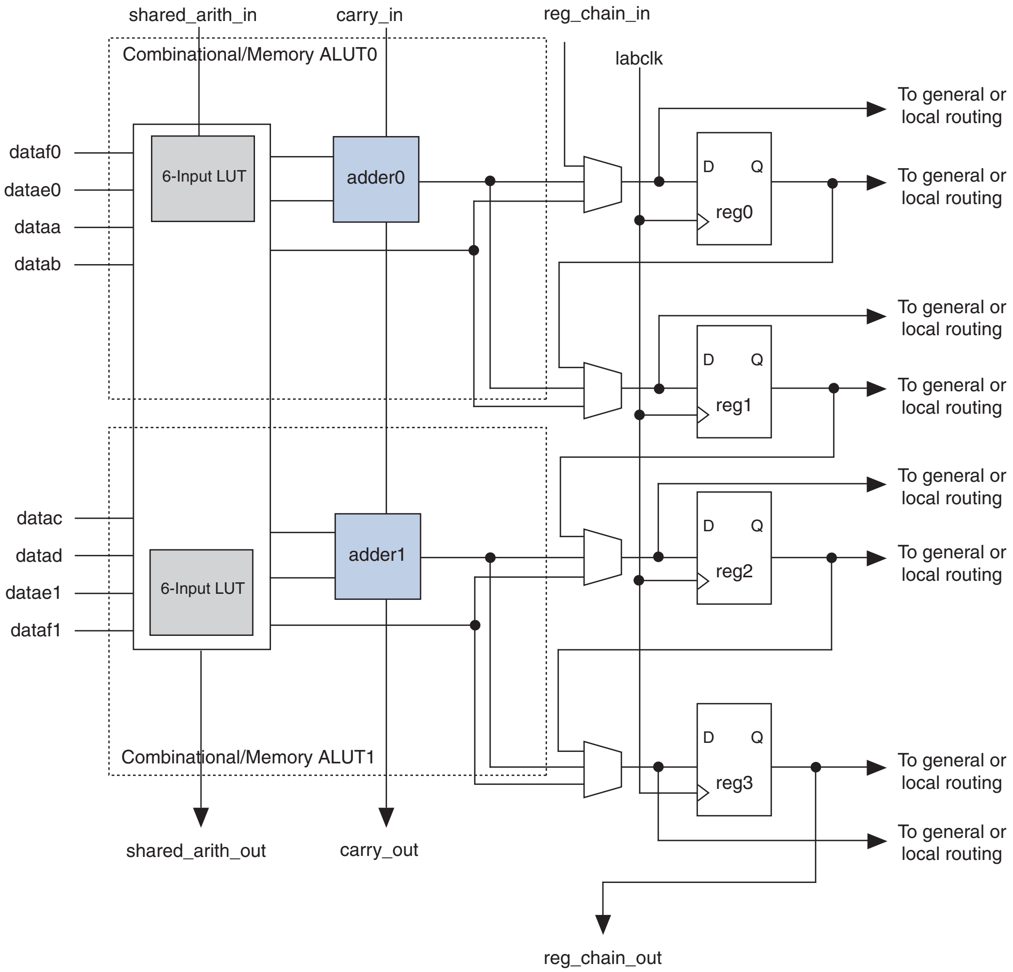
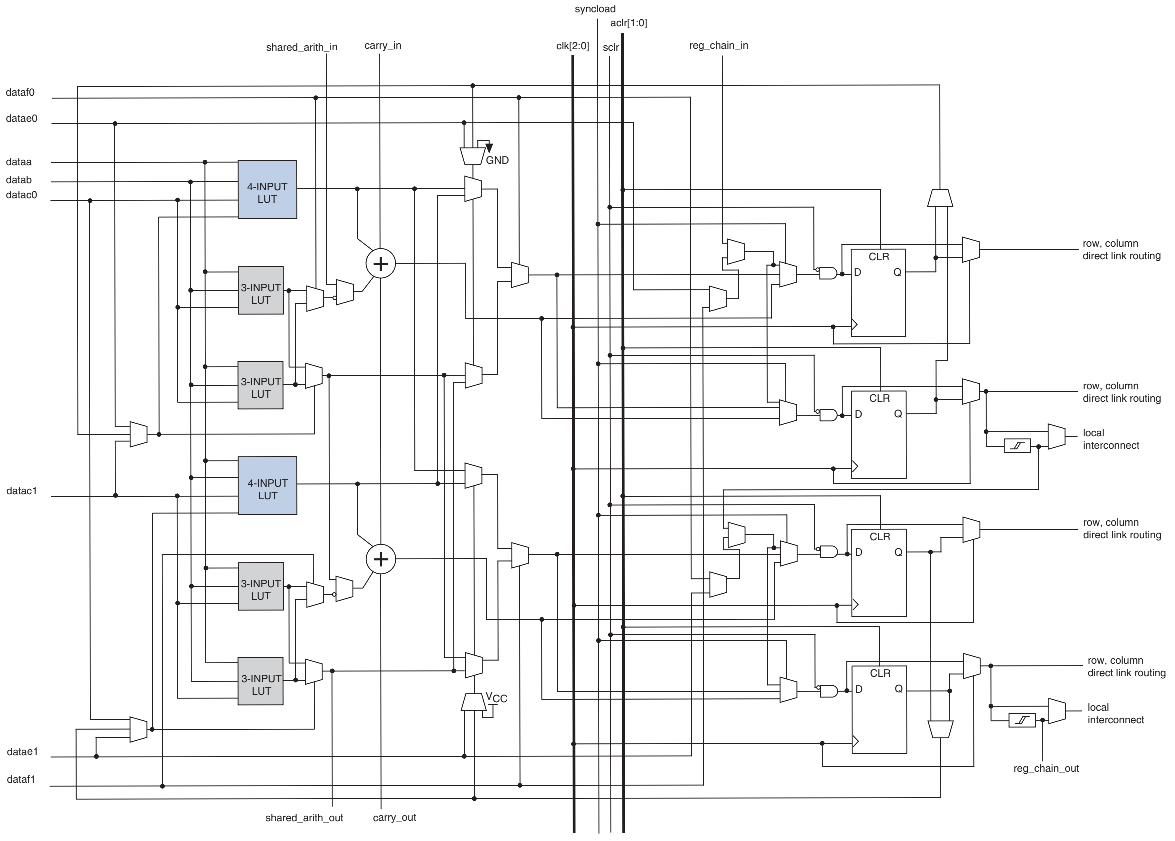

shared_arith_in
carry_in
reg_chain_in
Combinational/Memory ALUT0
labclk
To general or local routing
dataf0
datae0
dataa
datab
6-Input LUT
adder0
D
Q
reg0
To general or local routing
To general or local routing
reg1
To general or local routing
To general or local routing
datac
datad
datae1
dataf1
6-Input LUT
adder1
reg2
To general or local routing
reg3
Combinational/Memory ALUT1
To general or local routing
To general or local routing
shared_arith_out
carry_out
reg_chain_out

Figure 1–6 shows a detailed view of all the connections in an ALM.

Figure 1–6. ALM Connection Details for Stratix V Devices



One ALM contains four 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 1–6). 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 V ALM operates in one of the following modes:

- "Normal Mode" on page 1–7
- "Extended LUT Mode" on page 1–9
- "Arithmetic Mode" on page 1–10
- "Shared Arithmetic Mode" on page 1–11

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 about the LAB-wide control signals, refer to "LAB Control Signals" on page 1–3.

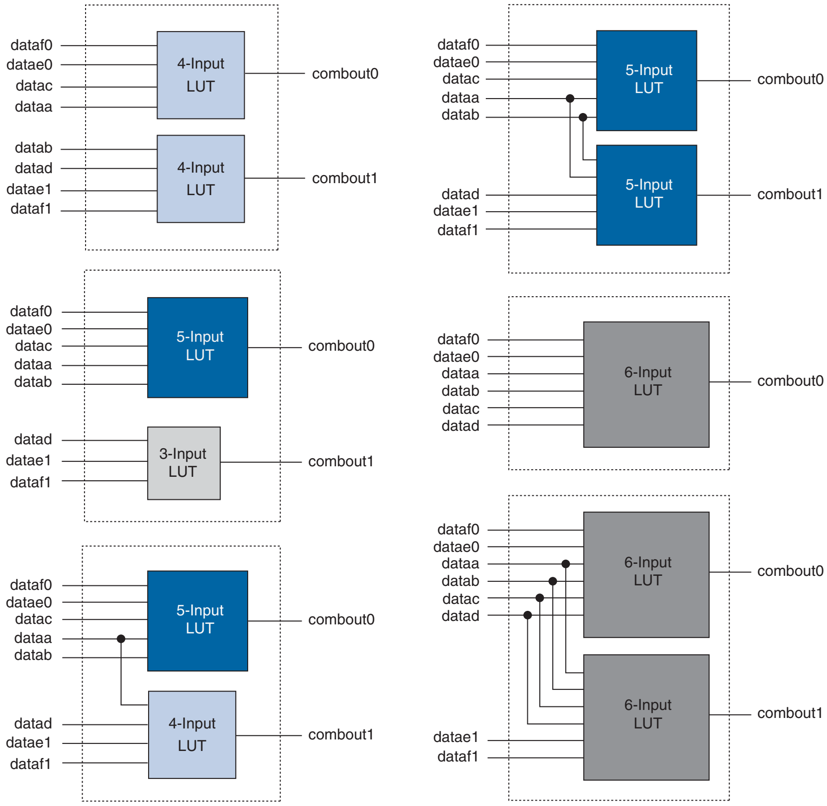
The Quartus II software and supported third-party synthesis tools, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions, automatically choose the appropriate mode for common functions such as counters, adders, subtractors, and arithmetic functions.

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 V 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 1–7 shows the supported LUT combinations in normal mode.

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



Note to Figure 1–7:

(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.

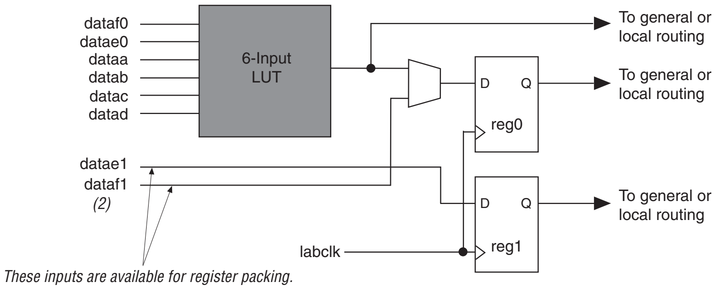
Normal mode provides complete backward-compatibility with four-input LUT architectures.

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 uses the full potential of the Stratix V 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 use by setting location assignments.

You can implement any six-input function using inputs dataa, datab, datac, datad, and either datae0 and dataf0 or datae1 and dataf1. If you use datae0 and dataf0, the output is either driven to register0, or register0 is bypassed, or the output driven to register0 and register0 is bypassed, and the data drives out to the interconnect using the top set of output drivers (Figure 1–8). If you use datae1 and dataf1, 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 1–8. Input Function in Normal Mode ***(Note 1)***


dataf0
datae0
dataa
datab
datac
datad
6-Input LUT
datae1
dataf1
(2)
These inputs are available for register packing.
labclk
D
Q
reg0
D
Q
reg1
To general or local routing
To general or local routing
To general or local routing

Notes to Figure 1–8:

(1) If you use datae1 and dataf1 as inputs to a six-input function, datae0 and dataf0 are available for register packing.

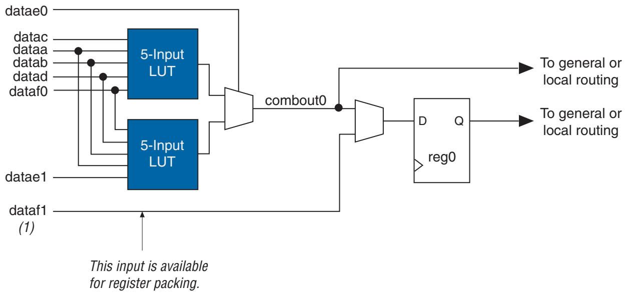
(2) The dataf1 input is available for register packing only if the six-input function is unregistered.

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 1–9 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 1–9 occur naturally in designs. These functions often appear in designs as “if-else” statements in Verilog HDL or VHDL code.

Figure 1–9. Template for Supported Seven-Input Functions in Extended LUT Mode


datae0
datac
dataa
datab
datad
dataf0
5-Input LUT
5-Input LUT
datae1
dataf1
(1)
This input is available for register packing.
combout0
D
Q
reg0
To general or local routing
To general or local routing

Note to Figure 1–9:

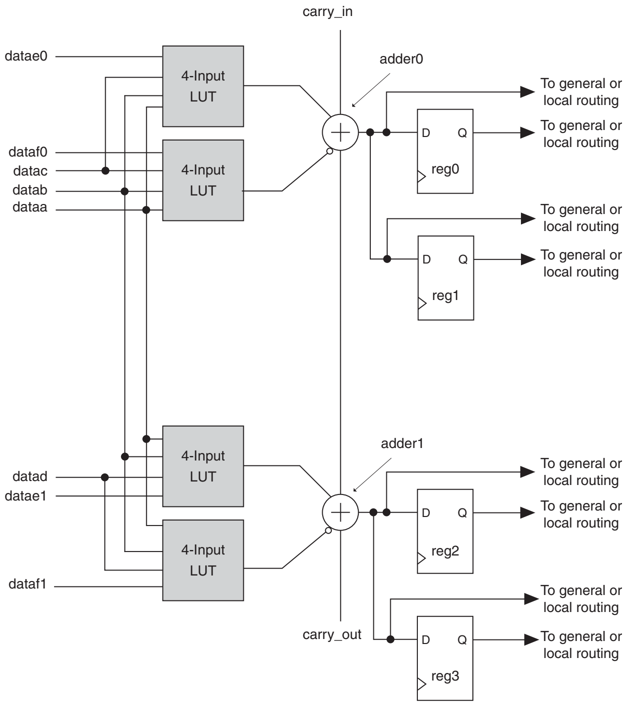
(1) If the seven-input function is unregistered, the unused eighth input is available for register packing. The second register, reg1, is not available.

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 1–10, 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 either registered, unregistered, or registered and unregistered versions of the adder outputs.

Figure 1–10. ALM in Arithmetic Mode


carry_in
datae0
adder0
4-Input LUT
To general or local routing
To general or local routing
D
Q
reg0
dataf0
datac
datab
dataa
4-Input LUT
To general or local routing
D
Q
To general or local routing
reg1
adder1
4-Input LUT
datad
datae1
To general or local routing
D
Q
To general or local routing
reg2
4-Input LUT
dataf1
To general or local routing
carry_out
D
Q
To general or local routing
reg3

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. You can individually disable or enable these signals for each 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 V 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 MLAB/M20K 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 use 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.

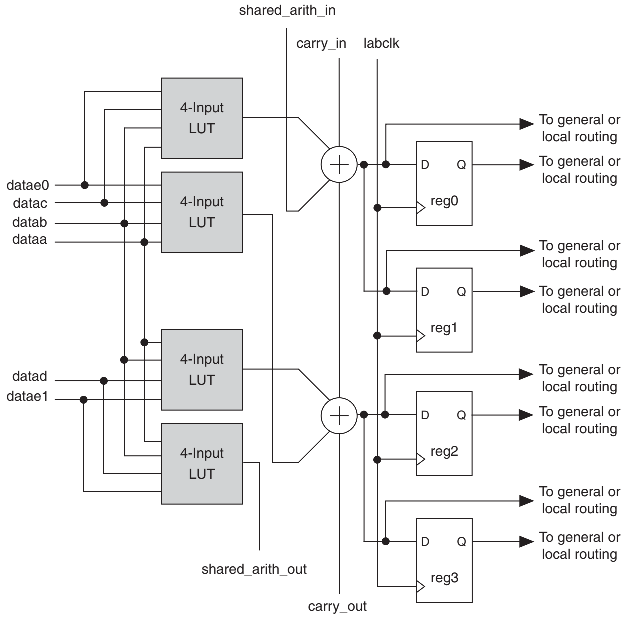
For more information about carry-chain interconnects, refer to "ALM Interconnects" on page 1–14.

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 1–11 shows the ALM using this feature.

Figure 1–11. ALM in Shared Arithmetic Mode



You can find adder trees in many different applications. For example, you can implement the summation of the partial products in a logic-based multiplier 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 despread data that was transmitted using 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 MLAB/M20K memory and DSP blocks. A shared arithmetic chain can continue as far as a full column.

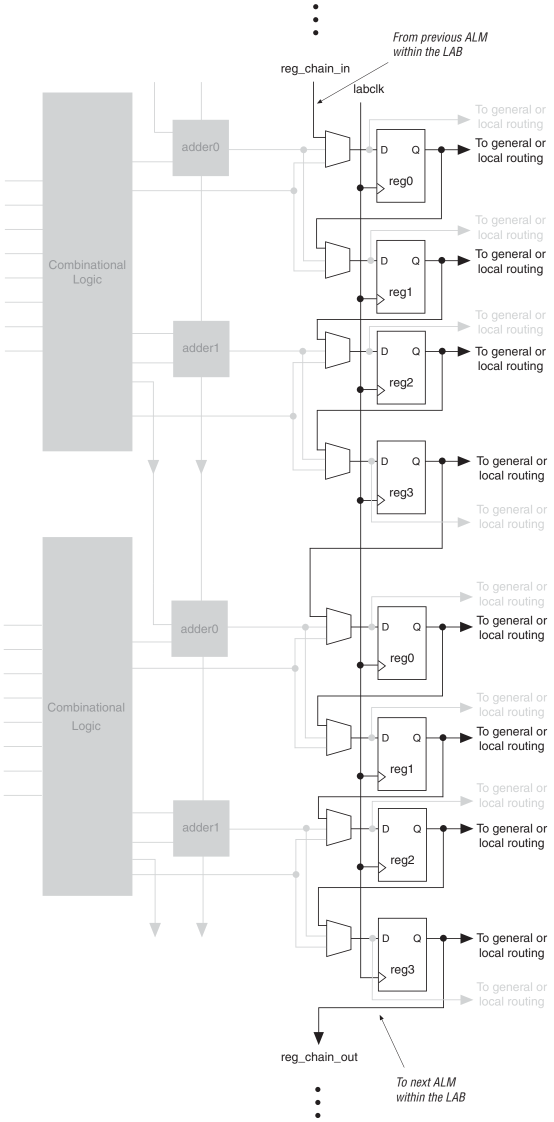
Similar to carry chains, the top and bottom halves of shared arithmetic chains in alternate LAB columns can be bypassed. This capability allows the shared arithmetic chain to cascade through half of the ALMs in a LAB while leaving the other half available for narrower fan-in functionality. The top half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

For more information on shared arithmetic chain interconnect, refer to "ALM Interconnects" on page 1–14.

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 1–12). The Quartus II Compiler automatically takes advantage of these resources to improve utilization and performance.

Figure 1–12. Register Chain within the LAB ***(Note 1)***


From previous ALM within the LAB
reg_chain_in
labclk
Combinational Logic
adder0
adder1
D
Q
reg0
reg1
reg2
reg3
To general or local routing
reg_chain_out
To next ALM within the LAB

Note to Figure 1–12:

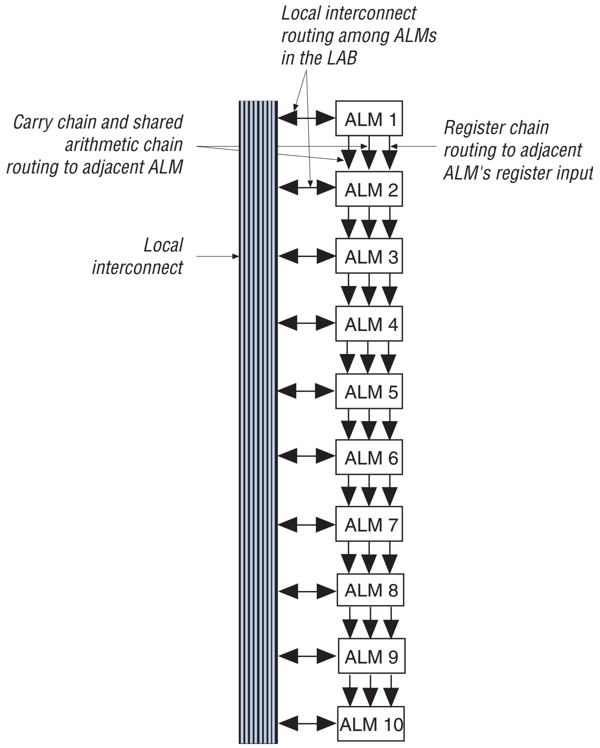
(1) You can use the combinational or adder logic to implement an unrelated, un-registered function.

For more information about the register-chain interconnect, refer to "ALM Interconnects" on page 1–14.

ALM Interconnects

There are three dedicated paths between ALMs—register cascade, carry chain, and shared arithmetic chain. Stratix V 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 1–13 shows the shared arithmetic chain, carry chain, and register chain interconnects.

Figure 1–13. Shared Arithmetic Chain, Carry Chain, and Register Chain Interconnects


Local interconnect routing among ALMs in the LAB
Carry chain and shared arithmetic chain routing to adjacent ALM
Register chain routing to adjacent ALM's register input
Local interconnect
ALM 1
ALM 2
ALM 3
ALM 4
ALM 5
ALM 6
ALM 7
ALM 8
ALM 9
ALM 10

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 V 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 the ALM adders are not in use.
- Stratix V 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 a LAB is a significant contributor to overall clock power consumption. Each LAB's clock and clock enable signals are linked. For example, a combinational ALUT or register in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. To disable a 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 1–1 lists the revision history for this chapter.

Table 1–1. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Figure 1–6. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

2. Memory Blocks in Stratix V Devices

SV51003-1.2

Embedded memory blocks include 640-bit enhanced memory logic array blocks (MLABs) and 20-Kbit M20K blocks. This chapter describes the embedded memory blocks in Stratix® V devices. Embedded memory blocks provide different sizes of embedded SRAM to address the Stratix V device design requirements efficiently. MLABs are optimized to implement shift registers for digital signal processing (DSP) applications, wide shallow FIFO buffers, and filter delay lines. You can use the M20K blocks to support larger memory configurations and include error correction code (ECC).

This chapter contains the following sections:

- "Overview" on page 2–1
- "Memory Modes" on page 2–8
- "Clocking Modes" on page 2–16
- "Design Considerations" on page 2–17

Overview

Table 2–1 lists the features supported by the embedded memory blocks.

Table 2–1. Summary of Memory Features in Stratix V Devices (Part 1 of 2)

Feature	MLABs	M20K
Maximum performance	600 MHz	600 MHz
Total RAM bits (including parity bits)	640	20,480
Configurations (depth x width)	64 x 8 64 x 9 64 x 10 32 x 16 32 x 18 32 x 20	16K x 1 8K x 2 4K x 4 4K x 5 2K x 8 2K x 10 1K x 16 1K x 20 512 x 32 512 x 40
Parity bits	✓	✓
Byte enable	✓	✓
Packed mode	—	✓
Address clock enable	✓	✓
Single-port memory	✓	✓

Table 2–1. Summary of Memory Features in Stratix V Devices (Part 2 of 2)

Feature	MLABs	M20K
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
Register clears	Output registers	Output registers
Write/Read operation triggering	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 **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**
ECC support	Soft IP support using the Quartus® II software	Built-in support in x32-wide simple dual-port mode or soft IP support using the Quartus II software

Table 2–2 lists the capacity and distribution of the embedded memory blocks in each Stratix V device.

Table 2–2. Memory Capacity and Distribution in Stratix V Devices (Part 1 of 2)

Family	Device	MLABs	M20K Blocks	Total Dedicated RAM Bits (M20K Blocks Only) (Mb)	Total RAM Bits (Including LABs) (Mb)
Stratix V GX	5SGXA3	3,776	800	15.6	17.9
	5SGXA4	5,880	1,344	26.3	29.8
	5SGXA5	8,020	2,304	45.0	49.9
	5SGXA7	11,736	2,560	50.0	57.2
	5SGXA9	15,850	2,640	51.6	61.2
	5SGXAB	17,960	2,640	51.6	62.5
	5SGXB5	9,250	2,100	41.0	46.7
	5SGXB6	11,270	2,660	52.0	58.8
Stratix V GT	5SGTC5	8,020	2,304	45.0	49.9
	5SGTC7	11,736	2,560	50.0	57.2
Stratix V GS	5SGSD2	2,450	450	8.8	10.3
	5SGSD3	4,536	686	13.4	16.2
	5SGSD4	6,264	1,062	20.7	24.6
	5SGSD5	8,630	2,014	39.3	44.6
	5SGSD6	11,000	2,320	45.3	52.0
	5SGSD8	13,280	2,624	51.3	59.4

Table 2–2. Memory Capacity and Distribution in Stratix V Devices (Part 2 of 2)

Family	Device	MLABs	M20K Blocks	Total Dedicated RAM Bits (M20K Blocks Only) (Mb)	Total RAM Bits (Including LABs) (Mb)
Stratix V E	5SEE9	15,850	2,640	51.6	61.2
	5SEEB	15,850	2,640	51.6	61.2

Embedded Memory Block Types

M20K memory blocks are dedicated resources. MLABs are dual-purpose blocks. You can configure the MLABs as regular logic array blocks (LABs) or as MLABs. Ten adaptive logic modules (ALMs) make up one MLAB. You can configure each ALM in an MLAB as either a 64 x 1 or a 32 x 2 block, resulting in a 64 x 10 or a 32 x 20 simple dual-port SRAM block in a single MLAB.

Parity Bit Support

On MLABs, 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.

The M20K supports one parity bit per 4-data bits when the data width is 5, 10, 20, or 40. The parity bits for inputs and outputs are bit 4, 9, 14, 19, 24, 29, 34, and 39. When writing or reading with non-parity widths, these bits are skipped. No parity function is performed on bit 4, 9, 14, 19, 24, 29, 34, and 39.

Byte Enable Support

All embedded 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 write operations of the RAM blocks.

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 do not have a clear port. When using parity bits on the M20K blocks, the byte enable controls ten bits (eight bits of data plus two parity bits). When using parity bits on the MLAB, the byte enable controls all ten 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. The byte enables are active high.

Table 2–3 lists the `byteena` controls in the x40 data width.

Table 2–3. byteena Controls in x40 Data Width

byteena[3..0]	Data Bits Written			
1111(default)	`[39:30]`	`[29:20]`	`[19:10]`	`[9:0]`
1000	`[39:30]`	—	—	—
0100	—	`[29:20]`	—	—
0010	—	—	`[19:10]`	—
0001	—	—	—	`[9:0]`

Table 2–4 lists the byteena controls in the x20 data width.

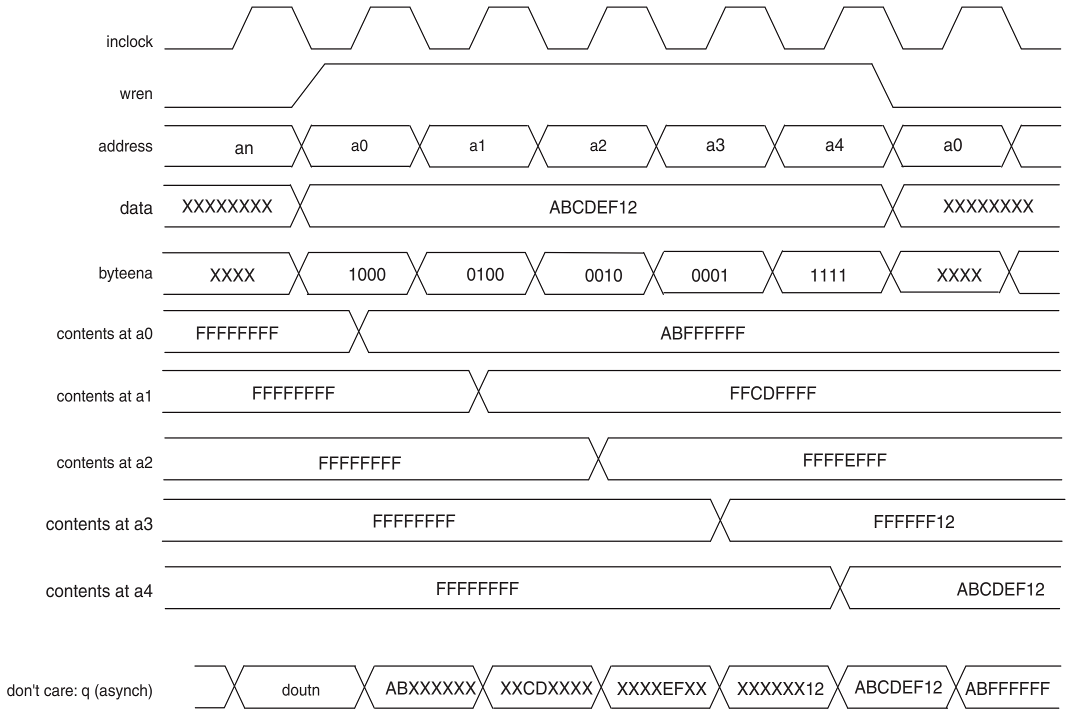
Table 2–4. byteena Controls in x20 Data Width

byteena[1:0]	Data Bits Written	
11(default)	[19:10]	[9:0]
10	[19:10]	—
01	—	[9:0]

If you use the ECC feature on the M20K blocks, you cannot use the byte enable feature.

Figure 2–1 shows how the wren and byteena signals control the operations of the RAM blocks. When a byte-enable bit is deasserted 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 with 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 2–1. Byte Enable Functional Waveform



Packed Mode Support

Stratix V M20K memory 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 in 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

Stratix V embedded 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 signal is low (disabled).

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

Figure 2–2. Address Clock Enable

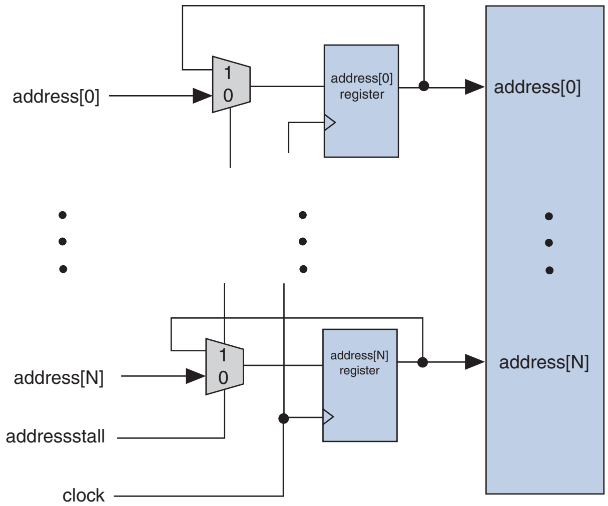


Figure 2–3 shows the address clock enable waveform during the read cycle.

Figure 2–3. Address Clock Enable During Read Cycle Waveform

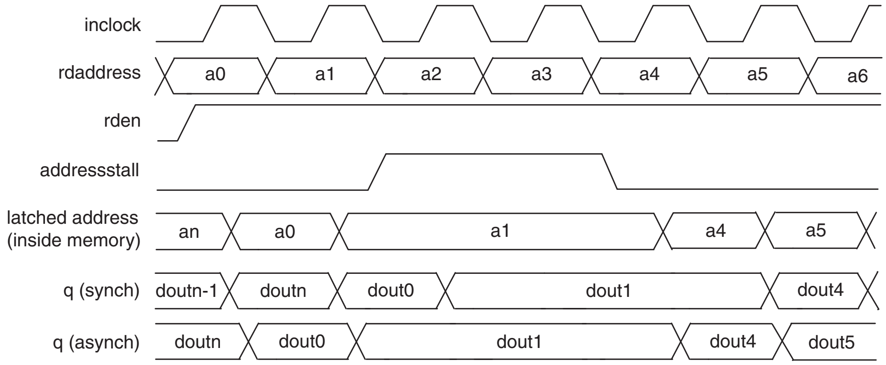
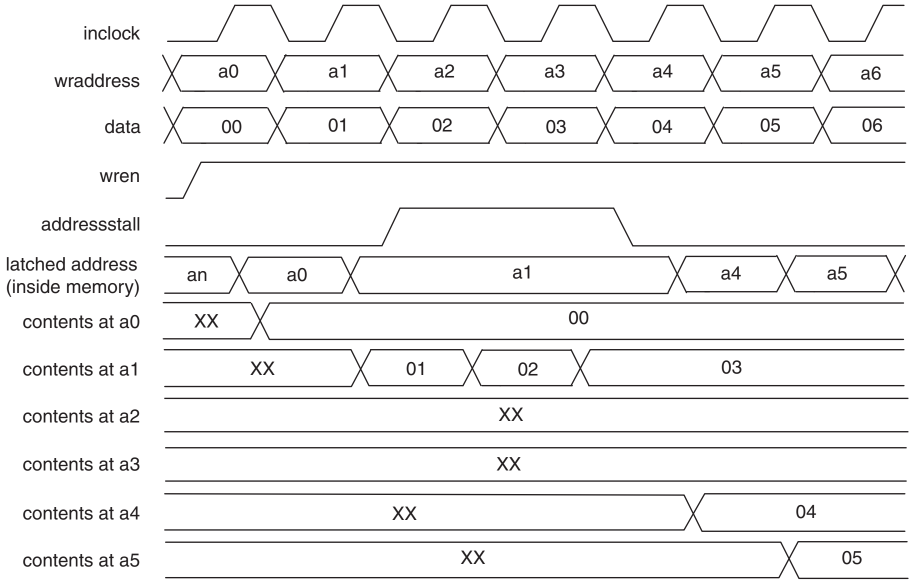


Figure 2–4 shows the address clock enable waveform during the write cycle.

Figure 2–4. Address Clock Enable During the Write Cycle Waveform



Mixed Width Support

M20K memory blocks support mixed data widths inherently. MLABs can support mixed data widths through emulation with 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 2–8.

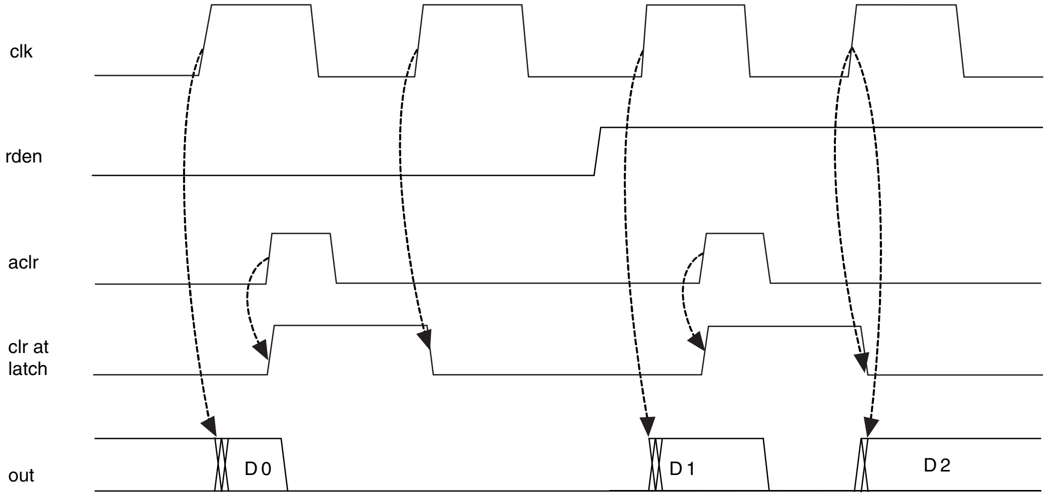
☞ MLABs do not support mixed-width FIFO mode.

Asynchronous Clear

M20K memory blocks support asynchronous clear on output latches and output registers. Therefore, if your RAM does not use output registers, clear the RAM outputs using the output latch asynchronous clear. Because the clear is an asynchronous signal, it is generated at any time. The internal logic extends the clear pulse until the next rising edge of the output clock. When the clear is asserted, the outputs are cleared and stay cleared until the next read cycle.

Figure 2–5 shows the output latch clear in Stratix V devices.

Figure 2–5. Output Latch Clear in Stratix V Devices



Error Correction Code Support

M20K blocks have built-in support for ECC when in x32-wide simple dual-port mode. ECC allows you to detect and correct data errors at the output of the memory. ECC can perform single-error correction, double-adjacent-error correction, and triple-adjacent-error detection in a 32-bit word; however, ECC cannot detect four or more errors.

The M20K runs slower than non-ECC simple-dual port mode when ECC is engaged; however, you can enable optional ECC pipeline registers before the output decoder to achieve the same performance as non-ECC simple-dual port mode at the expense of one cycle of latency.

The M20K ECC status is communicated with two ECC status flag signals—e (error) and ue (uncorrectable error). The status flags are part of the regular output from the memory block. When ECC is engaged, you cannot access two of the parity bits because they are replaced by the ECC status flag.

Table 2–5 lists the truth table for the ECC status flags.

Table 2–5. Truth Table for the ECC Status Flags *(Note 1)*

e (error)	ue (uncorrectable error)	Status
0	0	No error.
0	1	Illegal.
1	0	A correctable error occurred and the error has been corrected at the outputs; however, the memory array has not been updated.
1	1	An uncorrectable error occurred and uncorrectable data appears at the outputs.

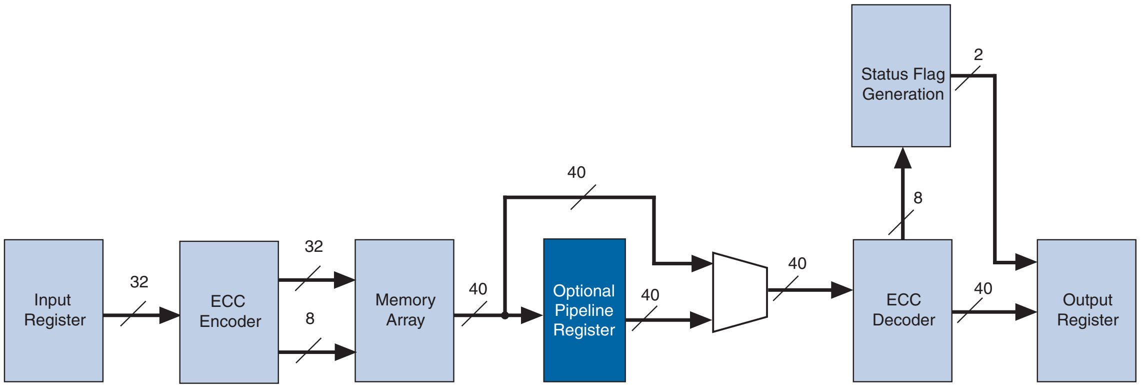
Note to Table 2–5:

(1) `eccstatus[1]` corresponds to `e` and `eccstatus[0]` corresponds to `ue`.

☞ When ECC is engaged, you cannot use the byte enable feature. Also, when ECC is engaged, read-during-write old data mode is not supported.

Figure 2–6 shows a diagram of the ECC block for M20K.

Figure 2–6. ECC Block for M20K



Memory Modes

Stratix V embedded memory blocks allow you to implement fully synchronous SRAM memory in multiple modes of operation. M20K blocks do not support asynchronous memory (unregistered inputs). MLABs support asynchronous (flow-through) read operations.

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

- "Single-Port RAM" on page 2–9
- "Simple Dual-Port Mode" on page 2–10
- "True Dual-Port Mode" on page 2–12 (only supported on M20K)
- "Shift-Register Mode" on page 2–14
- "ROM Mode" on page 2–15
- "FIFO Mode" on page 2–15

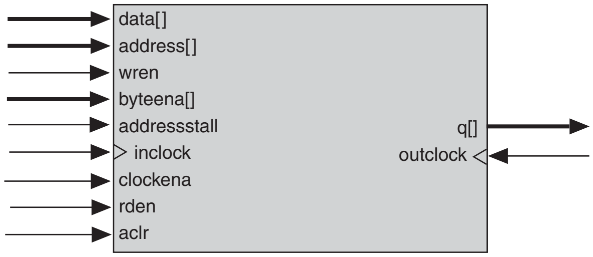
☞ If you use the memory blocks in ROM, single-port, simple dual-port, or true dual-port mode, you can corrupt the memory contents if you violate the setup or hold-time on any of the memory block input registers. This applies to both read and write operations.

Single-Port RAM

All embedded memory blocks support single-port mode. Single-port mode allows you to do either one-read or one-write operation at a time.

Figure 2–7 shows the single-port RAM configuration.

Figure 2–7. Single-Port RAM ***(Note 1)***



Note to Figure 2–7:

(1) You can implement two single-port memory blocks in a single M20K block. For more information, refer to "Packed Mode Support" on page 2–5.

During a write operation, the RAM output behavior is configurable. If you use the read-enable signal and perform a write operation with the read enable deactivated, 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 do not use the read-enable signal at all, the RAM outputs show the "new data" being written.

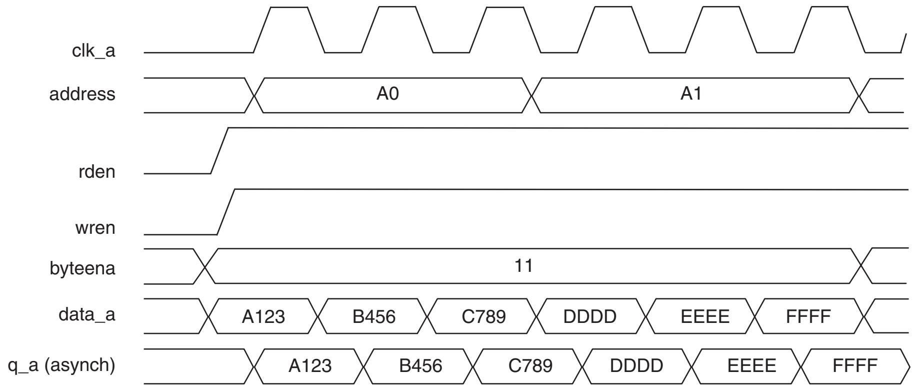
Table 2–6 lists the possible port width configurations for embedded memory blocks in single-port mode.

Table 2–6. Port Width Configurations for MLABs and M20K (Single-Port Mode)

Port Width Configurations	
MLABs	**M20K**
	16K x 1
	8K x 2
64 x 8	4K x 4
64 x 9	4K x 5
64 x 10	2K x 8
32 x 16	2K x 10
32 x 18	1K x 16
32 x 20	1K x 20
	512 x 32
	512 x 40

Figure 2–8 shows timing waveforms for read and write operations in single-port mode with unregistered outputs. Registering the RAM outputs delay the q output by one clock cycle.

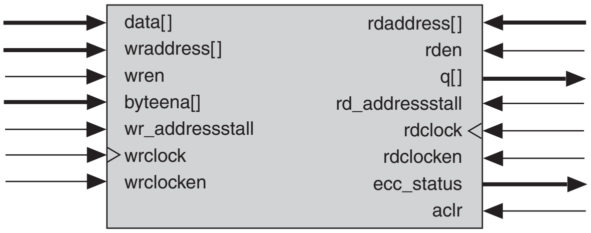
Figure 2–8. Timing Waveform for Read-Write Operations (Single-Port Mode)



Simple Dual-Port Mode

All embedded 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. The write operation happens on port A; the read operation happens on port B. Figure 2–9 shows a simple dual-port configuration.

Figure 2–9. Simple Dual-Port Memory for Stratix V Devices *(Note 1)*



Note to Figure 2–9:

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

Simple dual-port mode supports different read and write data widths (mixed-width support). Table 2–7 lists the mixed width configurations for M20K blocks in simple dual-port mode. MLABs do not have native support for mixed-width operations. The Quartus II software implements mixed-width memories in MLABs with more than one MLAB.

Table 2–7. M20K Block Mixed-Width Configurations (Simple Dual-Port Mode) (Part 1 of 2)

Read Port	Write Port									
	16K x 1	8K x 2	4K x 4	4K x 5	2K x 8	2K x 10	1K x 16	1K x 20	512 x 32	512 x 40
16K x 1	✓	✓	✓	—	✓	—	✓	—	✓	—
8K x 2	✓	✓	✓	—	✓	—	✓	—	✓	—

Table 2–7. M20K Block Mixed-Width Configurations (Simple Dual-Port Mode) (Part 2 of 2)

Read Port	Write Port									
	16K x 1	8K x 2	4K x 4	4K x 5	2K x 8	2K x 10	1K x 16	1K x 20	512 x 32	512 x 40
4K x 4	✓	✓	✓	—	✓	—	✓	—	✓	—
4K x 5	—	—	—	✓	—	✓	—	✓	—	✓
2K x 8	✓	✓	✓	—	✓	—	✓	—	✓	—
2K x 10	—	—	—	✓	—	✓	—	✓	—	✓
1K x 16	✓	✓	✓	—	✓	—	✓	—	✓	—
1K x 20	—	—	—	✓	—	✓	—	✓	—	✓
512 x 32	✓	✓	✓	—	✓	—	✓	—	✓	—
512 x 40	—	—	—	✓	—	✓	—	✓	—	✓

In simple dual-port mode, M20K 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. For more information, refer to “Read-During-Write Behavior” on page 2–17.

MLABs only support a write-enable signal. Read-during-write behavior for the MLABs can be a “new data”, “don’t care” or “old data” value. The available choices depend on the configuration of the MLAB.

Figure 2–10 shows timing waveforms for read and write operations in simple dual-port mode with unregistered outputs. Registering the RAM outputs delay the q output by one clock cycle.

Figure 2–10. Simple Dual-Port Timing Waveforms

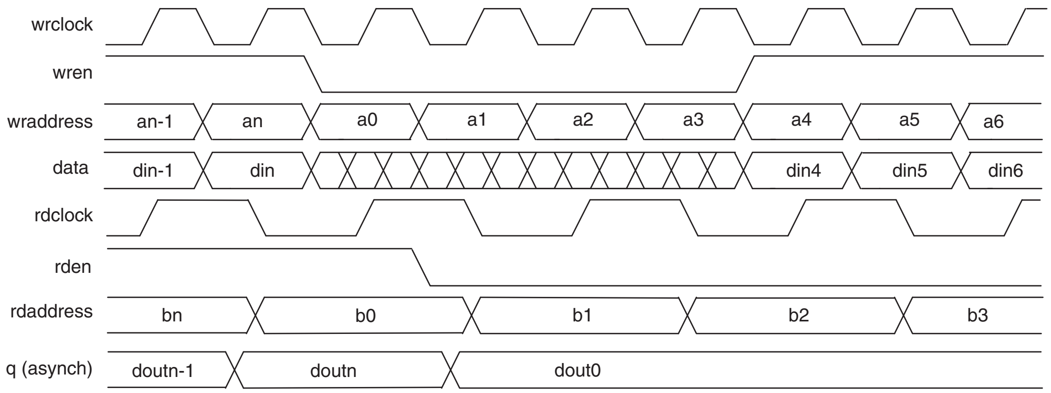
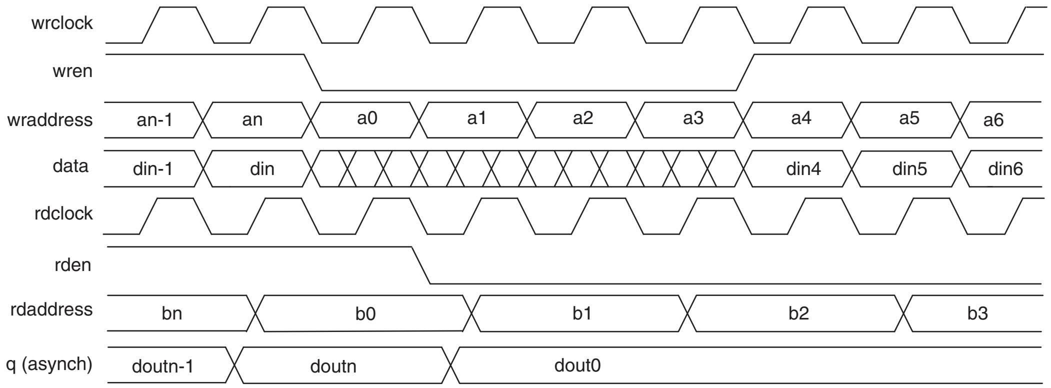


Figure 2–11 shows timing waveforms for read and write operations in mixed-port mode with unregistered outputs.

Figure 2–11. Mixed-Port Read-During-Write Timing Waveforms

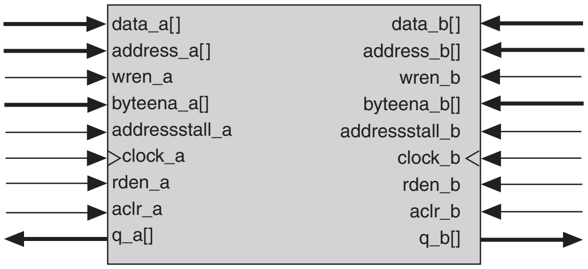


True Dual-Port Mode

Stratix V M20K blocks support true dual-port mode. Sometimes called bidirectional 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 2–12 shows the true dual-port RAM configuration.

Figure 2–12. True-Dual Port Memory for Stratix V Devices



The widest bit configuration of the M20K blocks in true dual-port mode is 1k x 16-bit (x20-bit with parity).

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 2–8 lists the possible M20K block mixed-port width configurations in true dual-port mode.

Table 2–8. M20K Block Mixed-Width Configurations (True Dual-Port Mode)

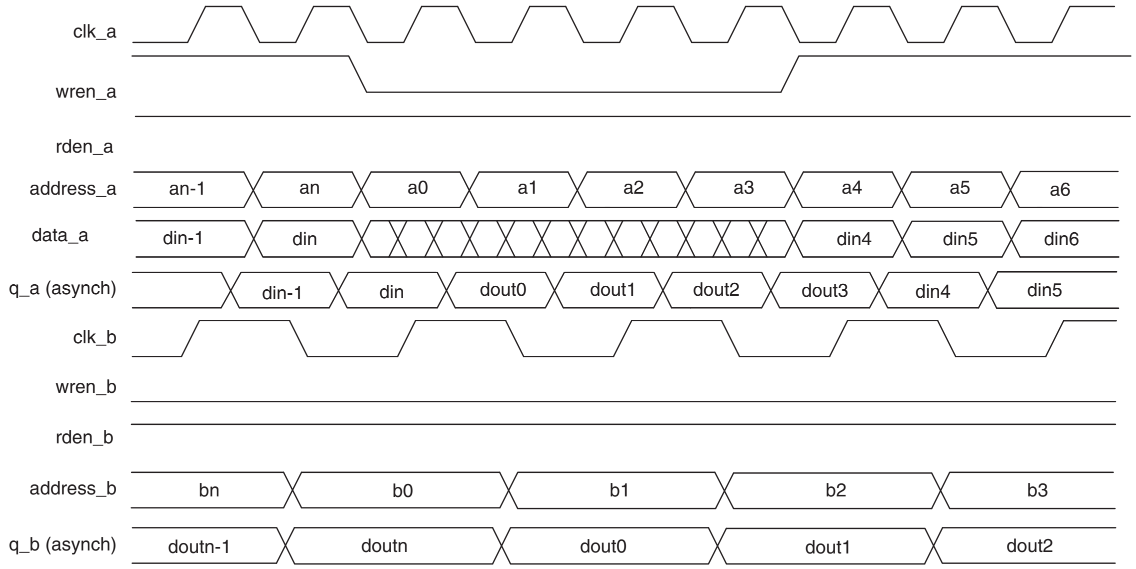
Port A	Port B							
	16K x 1	**8K x 2**	**4K x 4**	**4K x 5**	**2K x 8**	**2K x 10**	**1K x 16**	**1K x 20**
16K x 1	✓	✓	✓	—	✓	—	✓	—
8K x 2	✓	✓	✓	—	✓	—	✓	—
4K x 4	✓	✓	✓	—	✓	—	✓	—
4K x 5	—	—	—	✓	—	✓	—	✓
2K x 8	✓	✓	✓	—	✓	—	✓	—
2K x 10	—	—	—	✓	—	✓	—	✓
1K x 16	✓	✓	✓	—	✓	—	✓	—
1K x 20	—	—	—	✓	—	✓	—	✓

In true dual-port mode, M20K memory 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 2–17.

In true dual-port mode, you can access any memory location at any time from either port. If you are accessing the same memory location from both ports, you must avoid possible write conflicts. A write conflict happens if you are writing 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 V embedded memory blocks. You must resolve address conflicts external to the RAM block.

Figure 2–13 shows true dual-port timing waveforms for the write operation at port A and the read operation at port B, with the Read-During-Write behavior set to **new data**. Registering the RAM outputs delay the q outputs by one clock cycle.

Figure 2–13. True Dual-Port Timing Waveform



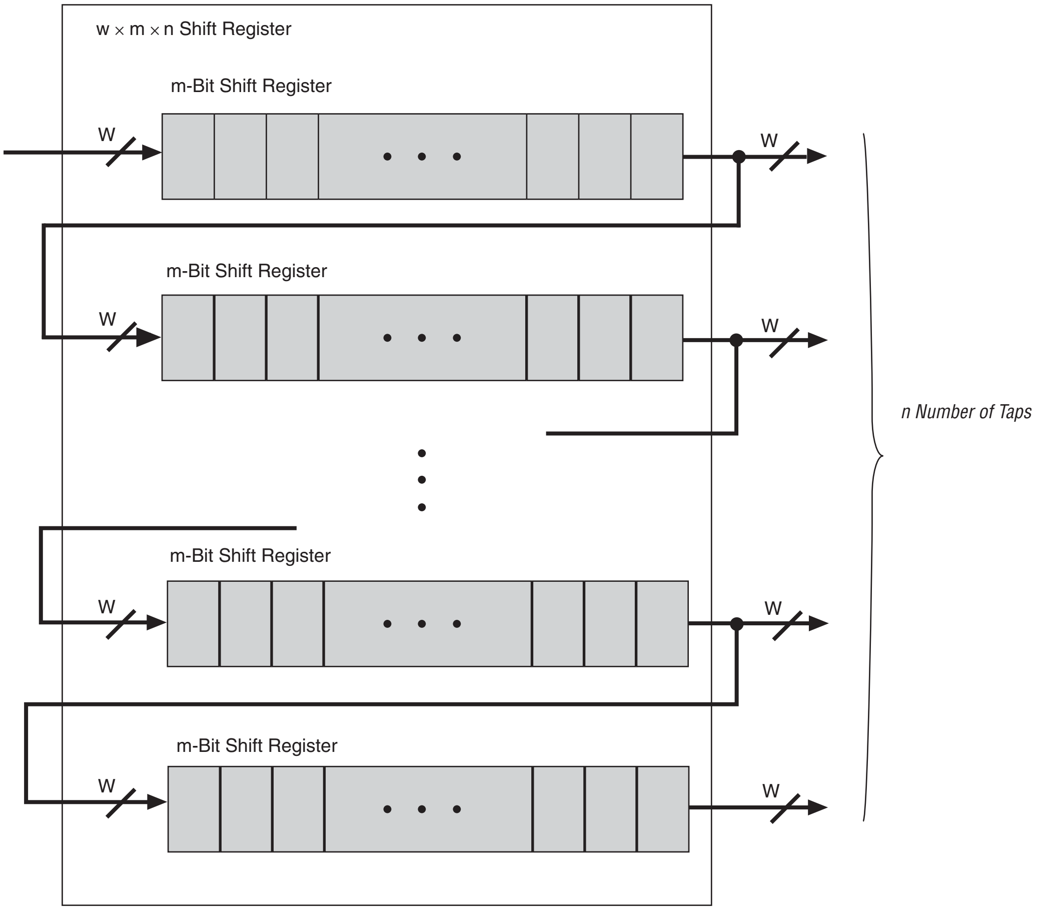
Shift-Register Mode

All Stratix V memory blocks support shift register mode. Embedded memory block configurations can implement shift registers for 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 exhaust many logic cells for large shift registers. Alternatively, you can use embedded memory as a shift-register block to save logic cell and routing resources.

The input data width (w), the length of the taps (m), and the number of taps (n) determine the size of a shift register (w x m x n). You can cascade memory blocks to implement larger shift registers.

Figure 2–14 shows the embedded memory block in shift-register mode.

Figure 2–14. Shift-Register Memory Configuration



ROM Mode

All Stratix V embedded memory blocks support ROM mode. A **.mif** initializes the ROM contents of these blocks. The address lines of the ROM are registered on M20K blocks; however, they 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 Stratix V embedded memory blocks support FIFO mode. MLABs are ideal for designs with many small, shallow FIFO buffers. You can use the FIFO MegaWizard Plug-In Manager to implement FIFO buffers in your design. The FIFO MegaWizard Plug-In Manager supports single- and dual-clock (asynchronous) FIFO buffers.

For more information about implementing FIFO buffers, refer to the *SCFIFO and DCFIFO Megafunctions User Guide*.

MLABs do not support mixed-width FIFO mode.

Clocking Modes

Stratix V embedded memory blocks support the following clocking modes:

- "Independent Clock Mode" on page 2–16
- "Input/Output Clock Mode" on page 2–16
- "Read/Write Clock Mode" on page 2–16
- "Single Clock Mode" on page 2–17

☞ 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 2–9 lists the internal memory clock modes.

Table 2–9. Internal 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 V embedded 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 V embedded 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 V embedded 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 V embedded 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, controls 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 embedded memory blocks.

Selecting Embedded 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 with the Quartus II software. Emulation results in minimal additional logic resources used. Because of the dual-purpose architecture of the MLAB, it only has data input registers and output registers in the block. MLABs gain read address registers from ALMs, while the write address and read data registers are internal to MLAB.

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

Conflict Resolution

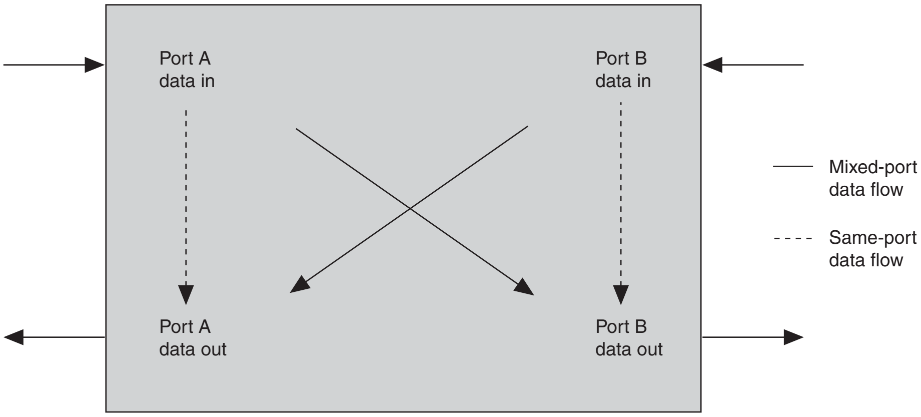
When using memory blocks in true dual-port mode, you can perform two write operations to the same memory location (address). Because there is no conflict resolution circuitry in the memory blocks, you must implement conflict resolution logic external to the memory block to avoid unknown data being written to the address.

Read-During-Write Behavior

You can customize the read-during-write behavior of the Stratix V embedded memory blocks to suit your design requirements. There are two types of read-during-write operations—same port and mixed port.

Figure 2–15 shows the difference between the two types.

Figure 2–15. Read-During-Write Data Flow for Stratix V Devices

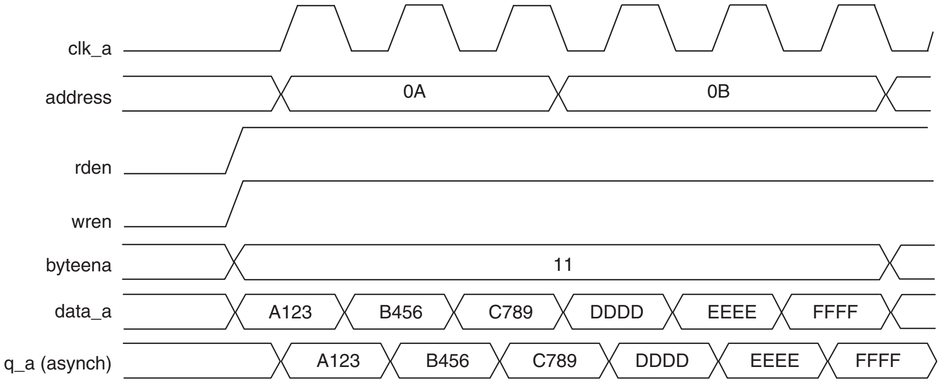


Same-Port Read-During-Write Mode

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, two output choices are available—new data mode (or flow-through) or don't care mode. If the MLAB is selected in same-port read-during-write mode, only the don't care mode is available. In new data mode, the new data is available on the rising edge of the same clock cycle on which it was written. In don't care mode, the RAM outputs "don't care" values for a read-during-write operation.

Figure 2–16 shows sample functional waveforms of same-port read-during-write behavior in new data mode.

Figure 2–16. Same-Port Read-During-Write—New 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" value 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.

The RAM MegaWizard Plug-In Manager controls the read-during-write behavior. For more information, refer to the *Internal Memory (RAM and ROM) Megafunction User Guide.*

Figure 2–17 shows a sample functional waveform of mixed-port read-during-write behavior for old data mode.

Figure 2–17. Mixed-Port Read-During-Write—Old Data Mode

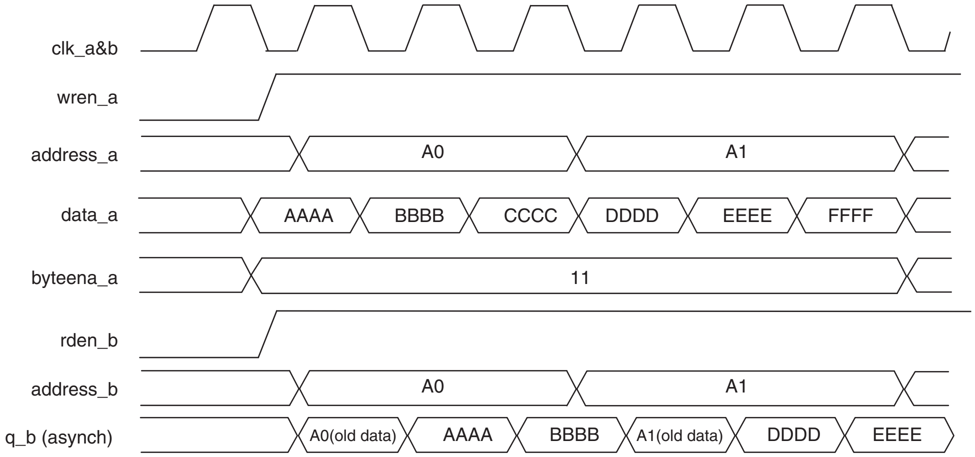
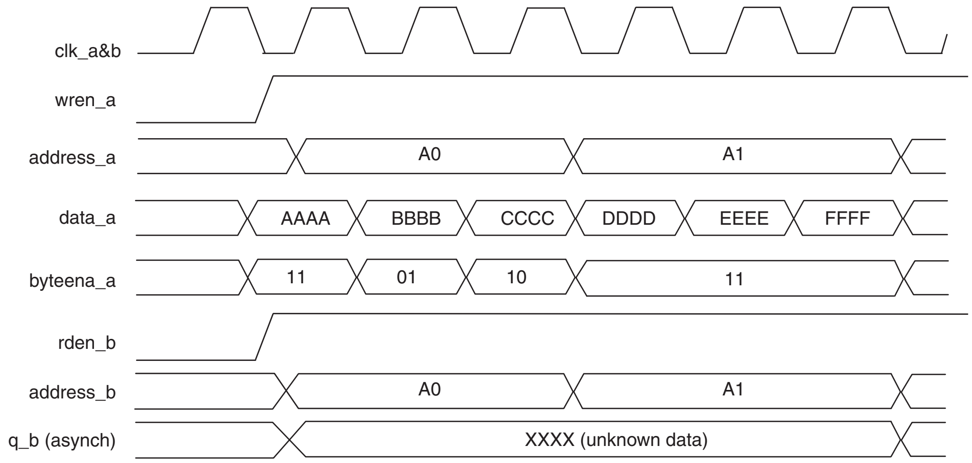


Figure 2–18 shows a sample functional waveform of mixed-port read-during-write behavior for don't care mode.

Figure 2–18. Mixed-Port Read-During-Write—Don't Care Mode



Mixed-port read-during-write is not supported if you use two different clocks 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

The M20K memory block outputs power up to zero (cleared), regardless of whether the output registers are used or bypassed. MLABs power up to zero if the output registers are used and power up reading the memory contents if the output registers are not used. You must consider this when designing logic that might evaluate the initial power-up values of the MLAB memory block. For Stratix V devices, the Quartus II software initializes the RAM cells to zero unless there is a **.mif** specified.

All memory blocks support initialization with a **.mif**. 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**), it still powers up with its outputs cleared.

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

Power Management

Stratix V memory block clock-enables allow you to control the clocking of each memory block to reduce AC power consumption. Use the read-enable signal to ensure that read operations only occur when you require read operations. If your design does not require read-during-write, you can reduce your power consumption by deasserting 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 2–10 lists the revision history for this chapter.

Table 2–10. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 2–1, Table 2–2, and Table 2–5. ■ Updated Figure 2–1 and Figure 2–8. ■ Updated "Read-During-Write Behavior" section. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

3. Variable Precision DSP Blocks in Stratix V Devices

SV51004-1.2

This chapter describes how the variable precision digital signal processing (DSP) blocks in Stratix® V devices are optimized to support higher-bit precision in high-performance DSP applications, such as radar systems that must support higher resolution and multi-antenna architectures, wireless-base station channel cards for MIMO processing, medical and test applications for very high-precision filtering, and fast Fourier transforms (FFTs) functions.

You can configure a variable precision DSP block to implement one of several operational modes to suit your application. The built-in pre-adder, coefficient bank, multipliers, and adder/subtractor minimize the amount of external logic to implement these functions, resulting in efficient resource usage, reduced power consumption, improved performance, and data throughput for DSP applications.

A Stratix V variable precision DSP block maintains backward compatibility, so it can efficiently support existing 18-bit DSP applications, such as high-definition video processing, digital-up and down conversion, and multi-rate filtering.

This chapter contains the following sections:

Variable Precision DSP Block Overview

Each Stratix V variable precision DSP block spans one logic array block (LAB) row height.

The following are the architectural highlights of the Stratix V variable precision DSP block:

- High-performance, power-optimized, and fully registered multiplication operations
- Natively supported 9-bit, 18-bit, and 27-bit word lengths
- Efficiently supported 18 x 25 complex multiplications for FFTs
- Efficiently supported floating-point arithmetic formats
- Built-in addition, subtraction, and 64-bit accumulation units to combine multiplication results efficiently
- Cascading 18-bit and 27-bit input bus to form tap-delay line for filtering applications

- Cascading 64-bit output bus to propagate output results from one block to the next block without external logic support
- Hard pre-adder supported in 18-bit and 27-bit mode for symmetric filters
- Internal coefficient register bank for filter implementation
- Efficiently supported 18-bit or 27-bit systolic finite impulse response (FIR) filters with distributed output adder

Table 3–1 lists the number of multipliers in Stratix V devices.

Table 3–1. Number of Multipliers in Stratix V Devices

Family	Device	Variable Precision DSP Blocks	Independent Input and Output Multiplication Operators					18 × 18 Multiplier Adder Mode	18 × 18 Multiplier Summed with 36-Bit input
			9 × 9 Multipliers	16 × 16 Multipliers	18 × 18 Multipliers with 32-Bit Resolution	27 × 27 Multipliers	36 × 18 Multipliers		
Stratix V GX	5SGXA3	188	564	376	376	188	188	376	188
	5SGXA4	188	564	376	376	188	188	376	188
	5SGXA5	256	768	512	512	256	256	512	256
	5SGXA7	256	768	512	512	256	256	512	256
	5SGXA9	352	1,056	704	704	352	352	704	352
	5SGXAB	352	1,056	704	704	352	352	704	352
	5SGXB5	399	1,197	798	798	399	399	798	399
	5SGXB6	399	1,197	798	798	399	399	798	399
Stratix V GT	5SGTC5	256	768	512	512	256	256	512	256
	5SGTC7	256	768	512	512	256	256	512	256
Stratix V GS	5SGSD2	325	975	650	650	325	325	650	325
	5SGSD3	630	1,890	1,260	1,260	630	630	1,260	630
	5SGSD4	946	2,838	1,892	1,892	946	946	1,892	946
	5SGSD5	1,498	4,494	2,996	2,996	1,498	1,498	2,996	1,498
	5SGSD6	1,775	5,325	3,550	3,550	1,775	1,775	3,550	1,775
	5SGSD8	2,048	6,144	4,096	4,096	2,048	2,048	4,096	2,048
Stratix V E	5SEE9	352	1,056	704	704	352	352	704	352
	5SEEB	352	1,056	704	704	352	352	704	352

Operational Modes Overview

Table 3–2 lists the summary of the operational modes that are supported by Stratix V variable precision DSP blocks.

Table 3–2. Variable Precision DSP Blocks Operational Modes for Stratix V Devices

Variable Precision DSP Block Resources	Operational Mode	Supported Instance	Pre-adder Support	Coefficient Support	Input Cascade support	Chainout Support
1 variable precision DSP block	Independent 9 x 9 Multiplication	3	No	No	No	No
	Independent 16 x 16 Multiplication	2	Yes	Yes	Yes	No
	Independent 18 x 18 Partial Multiplication (32-Bit)	2	Yes	Yes	Yes	No
	Independent 18 x 18 Multiplication	1	Yes	Yes	Yes	No
	Independent 27 x 27 Multiplication	1	Yes	Yes	Yes	Yes
	Independent 36 x 18 Multiplication	1	No	Yes	No	Yes
	Two 18 x 18 Multiplier Adder	1	Yes	Yes	Yes	Yes
	Two 16 x 16 Multiplier Adder	1	Yes	Yes	Yes	Yes
	Sum of 2 Square	1	Yes (1)	No	No	Yes
	18 x 18 Multiplication Summed with 36-Bit Input	1	No	No	No	Yes
2 variable precision DSP blocks	Independent 18 x 18 Multiplication	3	No	No	No	No
	Independent 36 x 36 Multiplication	1	No	No	No	No
	Complex 18 x 18 Multiplication	1	Yes	Yes	Yes	Yes
	Four 18 x 18 Multiplier Adder	1	Yes	Yes	Yes	No
	Two 27 x 27 Multiplier Adder	1	Yes	Yes	Yes	No
	Two 18 x 36 Multiplier Adder	1	No	Yes	No	No
3 variable precision DSP blocks	Complex 18 x 25 Multiplication	1	Yes (1)	No	No	No
4 variable precision DSP blocks	Complex 27 x 27 Multiplication	1	Yes	Yes	Yes	No

Note to Table 3–2:

(1) The pre-adder feature for this mode is automatically enabled.

The Quartus® II software includes megafunctions that you can use to control the operation mode of the multipliers. After making the appropriate parameter settings with the MegaWizard™ Plug-In Manager, the Quartus II software automatically configures the variable precision DSP block.

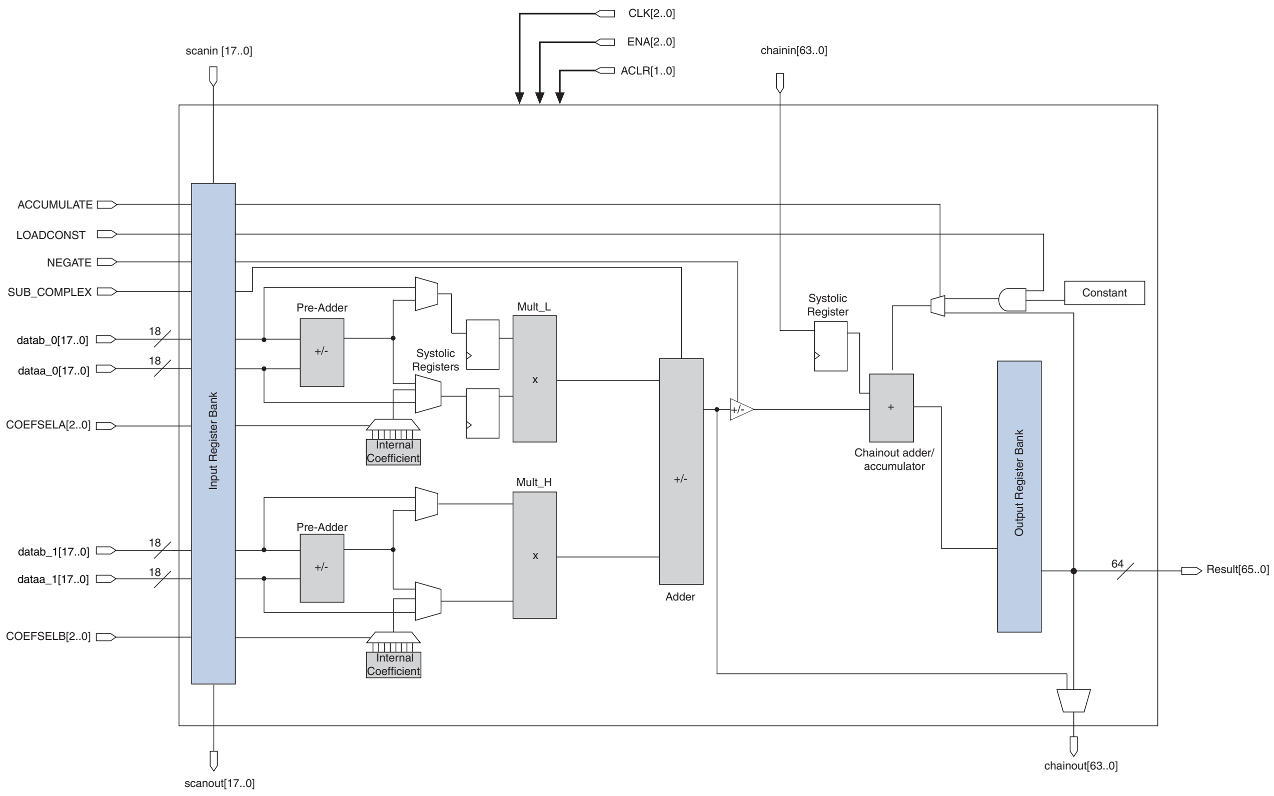
Variable Precision DSP Block Resource Descriptions

The Stratix V variable precision DSP block consists of the following elements:

- Input register bank
- Pre-adder and coefficient select
- Multipliers
- Compressors and accumulator
- Systolic registers
- 64-bit adder and output register bank

Figure 3–1 shows a detailed overall architecture of the Stratix V variable precision DSP block.

Figure 3–1. Variable Precision DSP Block Architecture for Stratix V Devices



Input Registers Bank

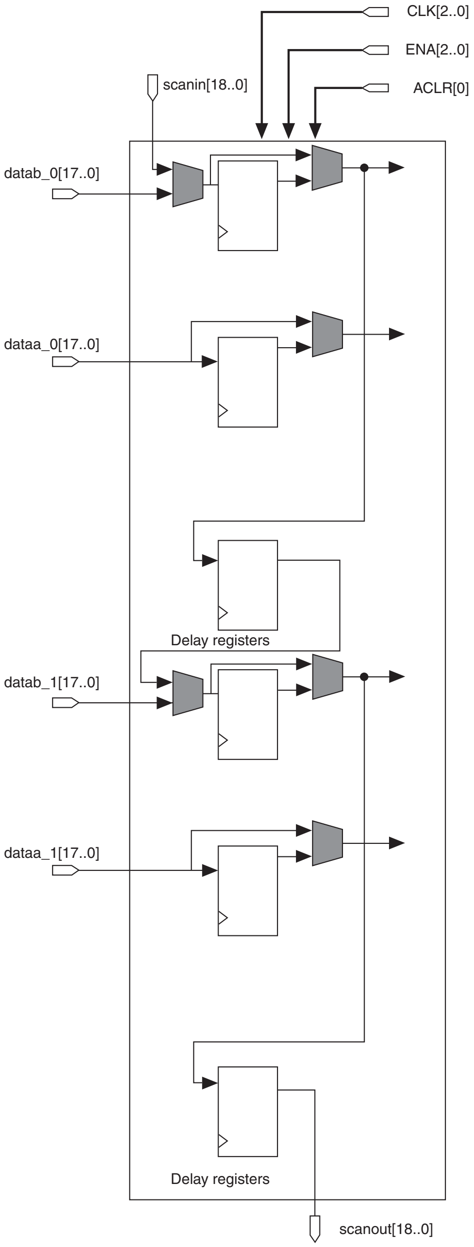
The positive edge of the clock signal triggers all variable precision DSP block registers and clears them after power up. Each multiplier operand can feed an input register or a multiplier directly, bypassing the input registers. The following variable precision DSP block signals control the input registers within the variable precision DSP block:

- `CLK[2..0]`
- `ENA[2..0]`
- `ACLR[0]`

Besides the registers for the data and dynamic control signals, there are also two sets of delay registers in the input register bank. The delay registers are used to balance the latency requirements when both the input cascade and chainout features are used. This is only supported in 18 x 18 mode.

One feature of the input register bank is to support a tap delay line; therefore, you can drive the top leg of the multiplier input (B) from general routing or from the cascade chain, as shown in Figure 3–2 and Figure 3–3. The Stratix V variable precision DSP block supports 18-bit and 27-bit input cascading.

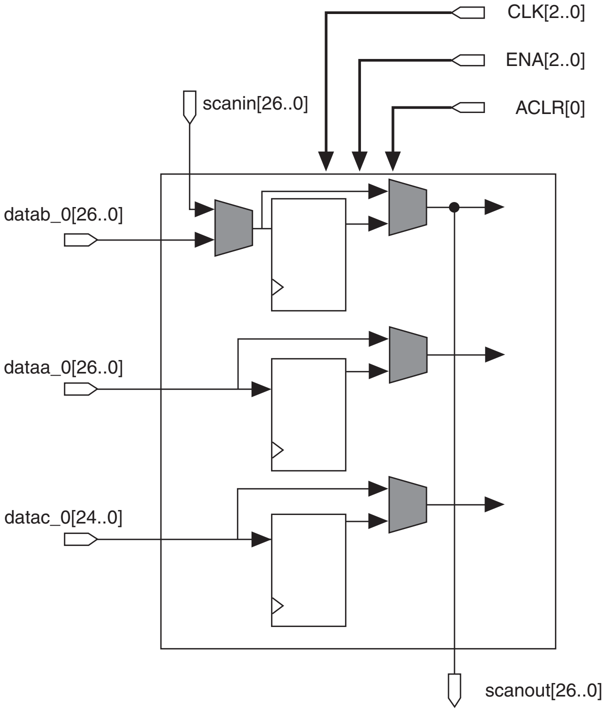
Figure 3–2. Input Register of a Variable Precision DSP Block in 18 x 18 Mode for Stratix V Devices



Note to Figure 3–2:

(1) Figure 3–2 shows only the data registers. Registers for the control signals are not shown.

Figure 3–3. Input Register of a Variable Precision DSP Block in 27 x 27 Mode for Stratix V Devices



Note to Figure 3–3:

(1) Figure 3–3 shows only the data registers. Registers for the control signals are not shown.

Pre-Adder and Coefficient Select

The pre-adder supports both addition and subtraction. There are two 18-bit pre-adders in each variable precision DSP block. You can configure these two pre-adders as two independent 18-bit adders for 18-bit applications, or a 25-bit adder for 27-bit applications.

The Stratix V variable precision DSP block has the flexibility of selecting the multiplicand from either the dynamic input or internal coefficient. The internal coefficient can support up to eight constant coefficients for the multiplicands in 18-bit and 27-bit applications. When you enable the internal coefficient feature, the COEFSELA/COEFSELB are used to control the dynamic selection of the coefficient multiplexer.

In 18-bit applications, you must enable the coefficient feature when the pre-adder feature is enabled. In 27-bit applications, you can use the coefficient feature and pre-adder feature independently. In 27-bit applications with the pre-adder feature enabled, the input data width is restricted to 22 bits if the multiplicand input comes from dynamic input due to input limitations. If the multiplicand input comes from internal coefficient, the data width of the input is 27 bits.

☞ When you enable the pre-adder feature, all input data and multipliers must have the same clock setting.

Multipliers

There are two multipliers (Mult_H and Mult_L) per variable precision DSP block. You can configure these two multipliers to work as a 27 x 27 multiplier, two 18 x 18 multipliers, or three 9 x 9 multipliers, depending on the operational mode. A single variable precision DSP block can perform many multiplications in parallel, depending on the data width of the multiplier. For more information, refer to "Operational Mode Descriptions" on page 3–9.

Adder and Accumulator

The Stratix V variable precision DSP block supports a 64-bit adder and 64-bit accumulator. You can use the 64-bit adder as a full 64-bit adder or several small adders with various sizes, depending on the operational mode.

Table 3–3 lists the functions supported by the accumulator in Stratix V devices.

Table 3–3. Functions Supported by Accumulator in Stratix V Devices

Function	Description
Zeroing	Disables the accumulator.
Preload	Loads an initial value to the accumulator. Only 1 bit of the 64-bit preload value can be "1". It can be used as rounding the DSP result to any position of the 64-bit result.
Accumulation	Adds the current result to the previous accumulate result.
Decimation	This function takes the current result, converts it into two's compliment, and adds it to the previous result.

You can dynamically control the function of the accumulator by three control signals—`NEGATE`, `LOADCONST`, and `ACCUMULATE`. Table 3–4 lists how these dynamic signals control the accumulator functions.

Table 3–4. Dynamic Control Signals for 64-Bit Accumulator for Stratix V Devices

Function	NEGATE	LOADCONST	ACCUMULATE
Zero	0	0	0
Accumulate	0	0	1
Decimate	1	0	1
Preload	0	1	0

Systolic Register

There are two systolic registers per variable precision DSP block. The first systolic register has two 18-bit registers that are used to register the Mult_L's two 18-bit inputs. You must clock these registers with the same clock source as the multiplier inputs. The second systolic register is a 44-bit register that is used to delay the `chainout` output to the next variable precision DSP block. You must clock this register with the same clock source as the output register bank. If the variable precision DSP block is not configured in systolic FIR mode, both systolic registers are bypassed.

Output Register Bank

The positive edge of the clock signal triggers the 64-bit bypassable output register bank and is cleared after power up. The following variable precision DSP block signals control the output register per variable precision DSP block:

- `CLK[2..0]`
- `ENA[2..0]`
- `ACLR[1]`

Operational Mode Descriptions

This section describes how you can configure a Stratix V variable precision DSP block to efficiently support the following operational modes:

- "Independent Multiplier Modes" on page 3–9
- "Independent Complex Multiplier Modes" on page 3–14
- "Multiplier Adder Sum Mode" on page 3–17
- "Sum of Square Mode" on page 3–20
- "18 x 18 Multiplication Summed with 36-Bit Input Mode" on page 3–20
- "Systolic FIR Mode" on page 3–21

Independent Multiplier Modes

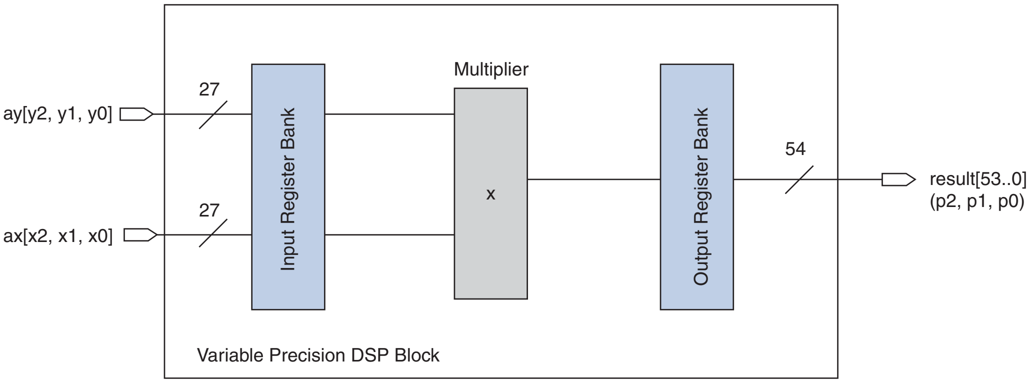
In independent input and output multiplier mode, the variable precision DSP blocks perform individual multiplication operations for general purpose multipliers.

9 x 9, 16 x 16, 18 x18, 27 x 27, and 36 x 18 Multipliers

You can configure each variable precision DSP block multiplier for 9-, 16-, 18-, 27-bit, or 36 x 18 multiplication. A variable precision DSP block can support up to three individual 9 x 9 multipliers, two individual 16 × 16 multipliers, two individual 18 x 18 partial multipliers, one individual 18 x 18 multiplier, one individual 27 x 27 multiplier, or one individual 36 x 18 multiplier. For some operational modes, the unused inputs require zero padding.

Figure 3–4, Figure 3–5, Figure 3–7 on page 3–12, Figure 3–8 on page 3–12, and Figure 3–9 on page 3–13 show the variable precision DSP block in independent multiplier operation mode. Figure 3–6 on page 3–11 shows that two variable precision DSP blocks can support three individual 18 x 18 multipliers.

Figure 3–4. Three 9 x 9 Independent Multiplier Mode for Stratix V Devices ***(Note 1)***



Note toFigure 3–4:

(1) Three pairs of data are packed into the ax and ay ports; result contains three 18-bit products.

Figure 3–5. One 18 x 18 Independent Multiplier Mode with One Variable Precision DSP Block for Stratix V Devices

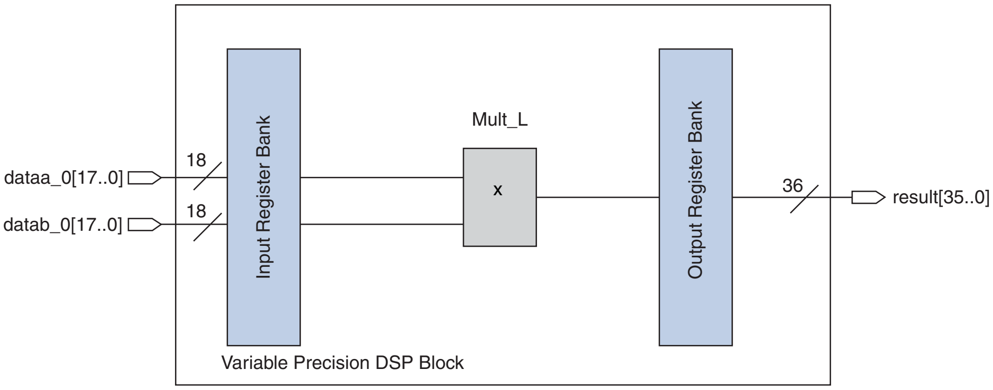


Figure 3–6. Three 18 x 18 Independent Multiplier Mode with Two Variable Precision DSP Blocks for Stratix V Devices

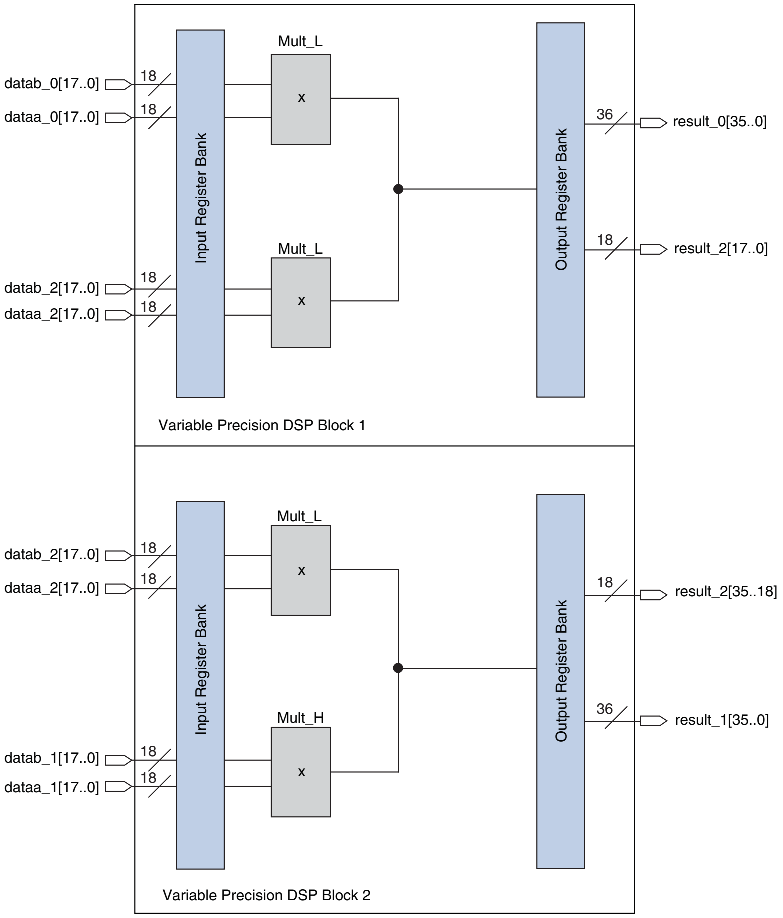
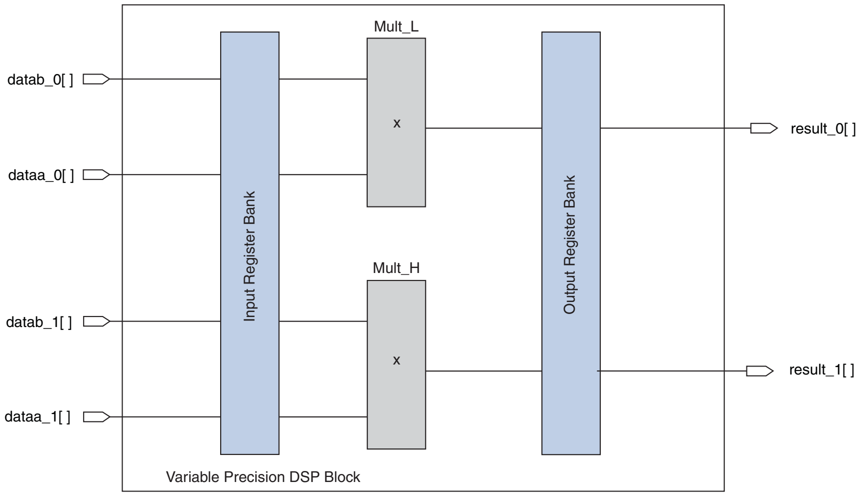


Figure 3–7. Two 16 x 16 Independent Multiplier Mode or Two 18 x 18 Independent Partial Multiplier Mode for Stratix V Devices ***(Note 1)*****,** ***(2)***

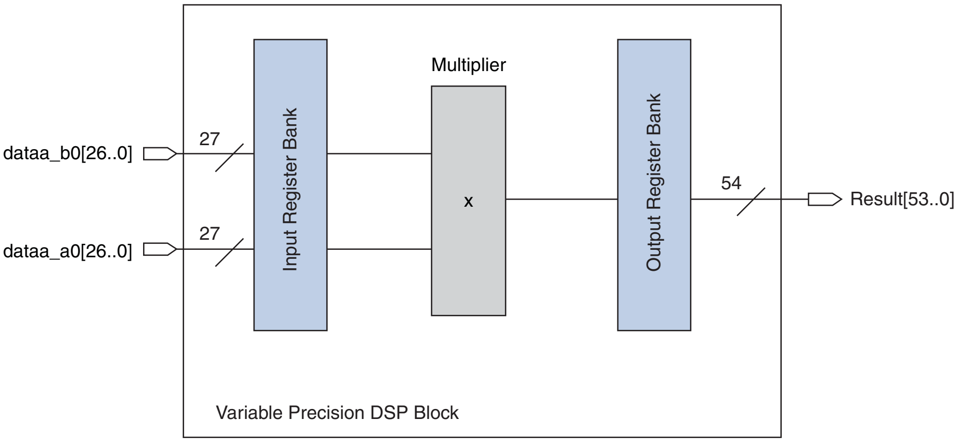


Notes to Figure 3–7:

(1) The inputs for 16-bit independent multiplier mode are `data[15..0]`. The unused input bits require padding with zero.

(2) For 18-bit independent multiplier mode, only 32-LSB output for both multipliers are routed to the output register.

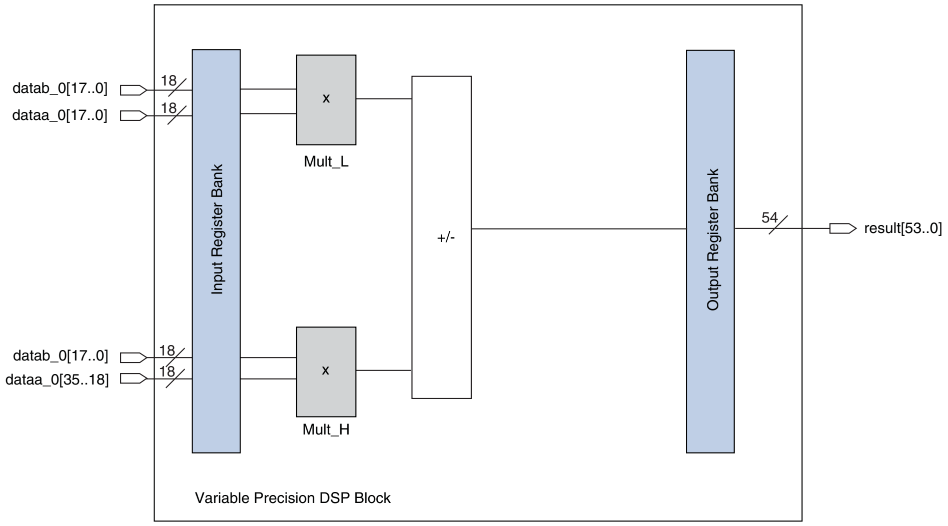
Figure 3–8. One 27 x 27 Independent Multiplier Mode for Stratix V Devices ***(Note 1)***



Note to Figure 3–8:

(1) The result can be up to 64-bits when combined with a chainout adder/accumulator.

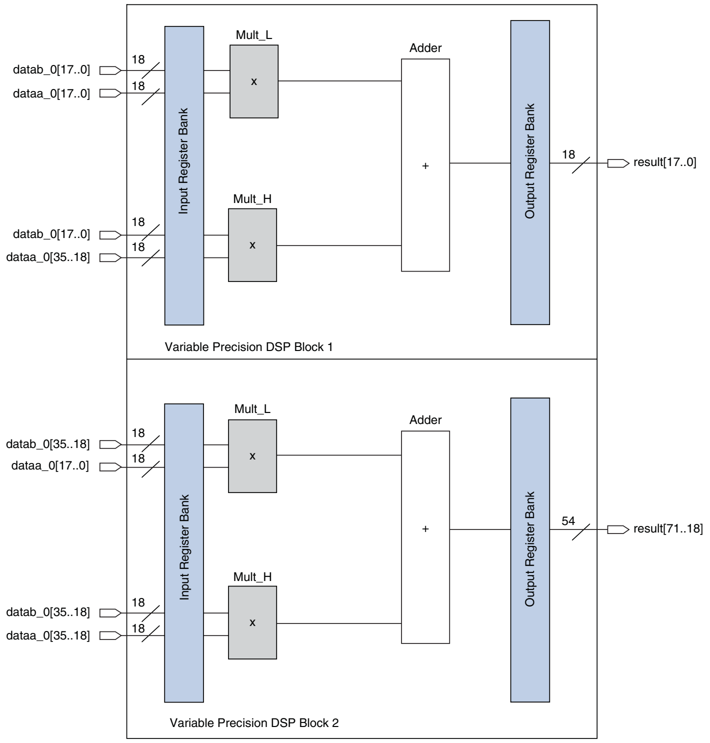
Figure 3–9. One 36 × 18 Independent Multiplier Mode for Stratix V Devices



36-Bit Multiplier

You can efficiently construct an individual 36-bit multiplier with two adjacent Stratix V variable precision DSP blocks. The 36 x 36 multiplication consists of four 18 x 18 multipliers, as shown in Figure 3–10. The 36-bit multiplier is useful for applications requiring more than 18-bit precision; for example, for the mantissa multiplication portion of very high precision fixed-point arithmetic applications.

Figure 3–10. 36-Bit Independent Multiplier Mode with Two Variable Precision DSP Blocks for Stratix V Devices



Independent Complex Multiplier Modes

The Stratix V variable precision DSP block provides the means for a complex multiplication. Equation 3–1 shows a complex multiplication that you can write.

Equation 3–1. Complex Multiplication Equation

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

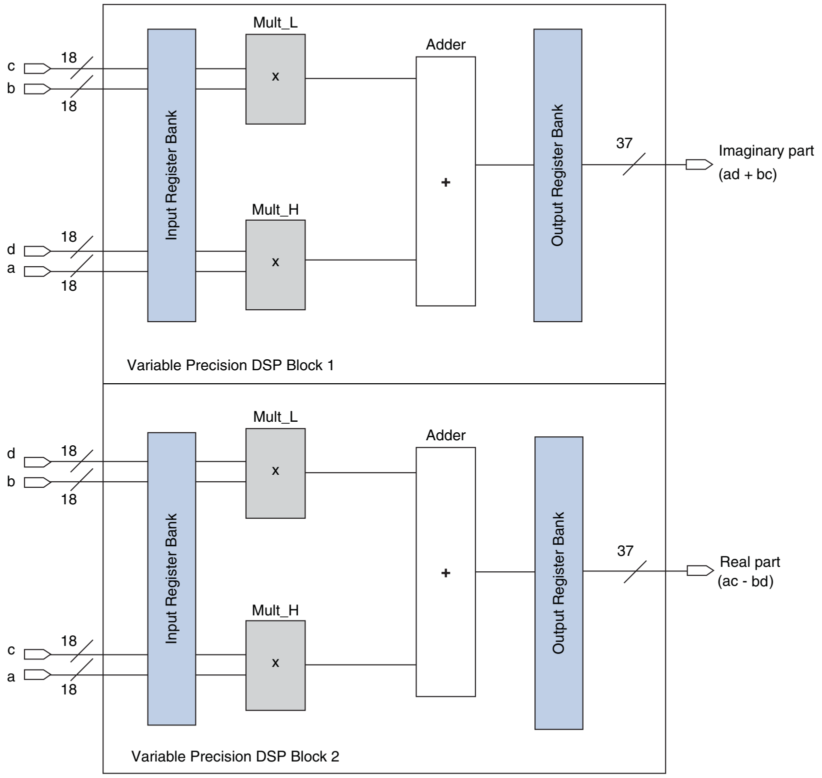
The Stratix V variable precision DSP block can support an 18 x 18 complex multiplier, 18 x 25 complex multiplier, or a 27 x 27 complex multiplier.

18 x 18 Complex Multiplier

For 18 x 18 complex multiplications, you require two variable precision DSP blocks to perform this multiplication mode. You can implement the imaginary part [(a × d) + (b × c)] in the first variable precision DSP block, and you can implement the real part [(a × c) – (b × d)] in the second variable precision DSP block.

Figure 3–11 shows an 18-bit complex multiplication.

Figure 3–11. 18 x 18 Complex Multiplier with Two Variable Precision DSP Blocks for Stratix V Devices



18 x 25 Complex Multiplier

Stratix V devices support an individual 18 x 25 complex multiplication mode. you require three variable precision DSP blocks to implement an individual 18 x 25 complex multiplication mode. The pre-adder feature is automatically enabled for you to implement an individual 18 x 25 complex multiplication mode efficiently.

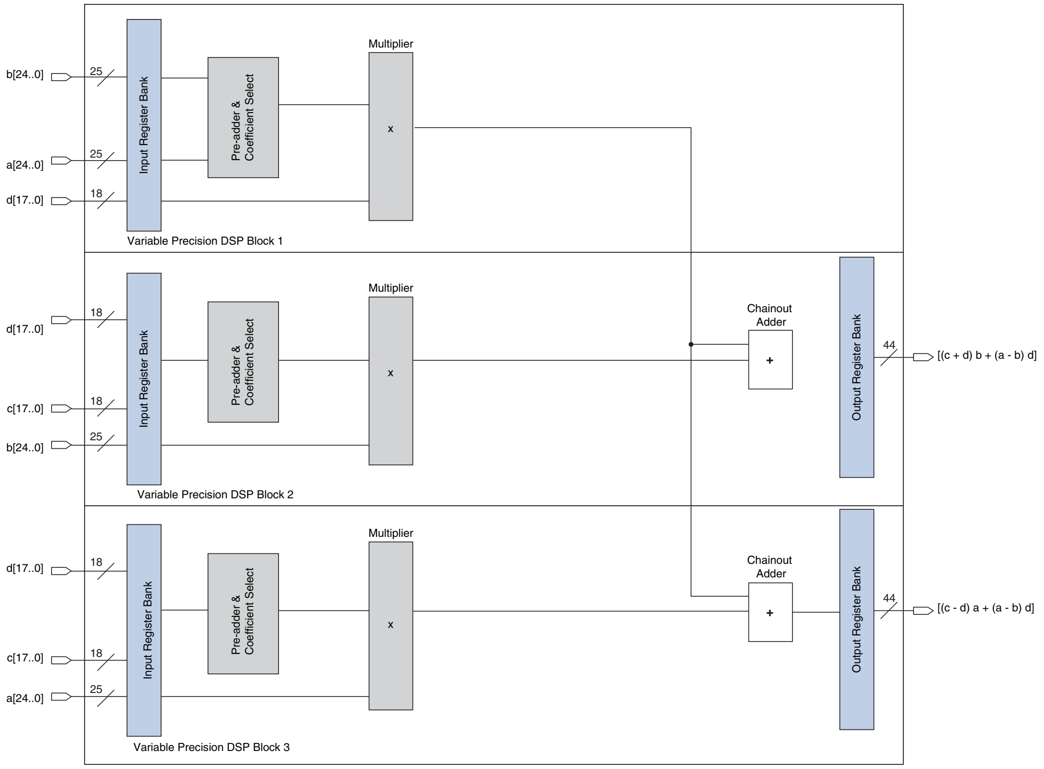
You can write an 18 x 25 complex multiplication with three variable precision DSP blocks, as shown in Equation 3–2.

Equation 3–2. 18 × 25 Complex Multiplication Equation

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

Figure 3–12 shows an 18 x 25 complex multiplication with three variable precision DSP blocks.

Figure 3–12. 18 x 25 Complex Multiplier with Three Variable Precision DSP Blocks for Stratix V Devices

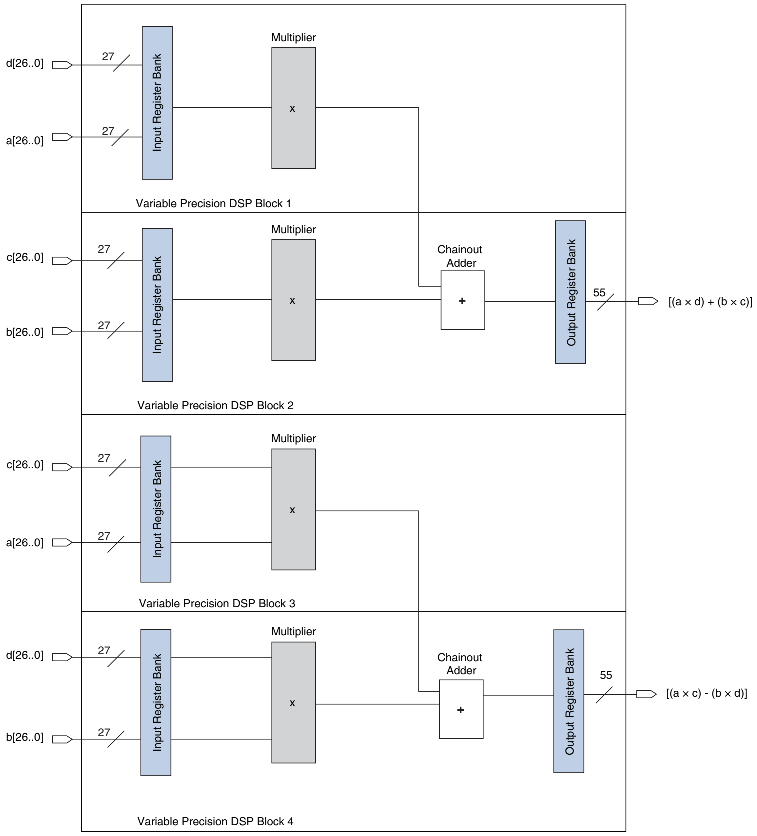


27 x 27 Complex Multiplier

Stratix V devices support an individual 27 x 27 complex multiplication mode. You require four variable precision DSP blocks to implement an individual 27 x 27 complex multiplication mode. You can implement the imaginary part [(a × d) + (b × c)] in the first and second variable precision DSP blocks, and you can implement the real part [(a × c) - (b × d)] in the third and fourth variable precision DSP blocks. You can achieve the difference of two 27 × 27 multiplications by enabling the NEGATE control signal in the fourth variable precision DSP block.

Figure 3–13 shows a 27-bit complex multiplication with four variable precision DSP blocks.

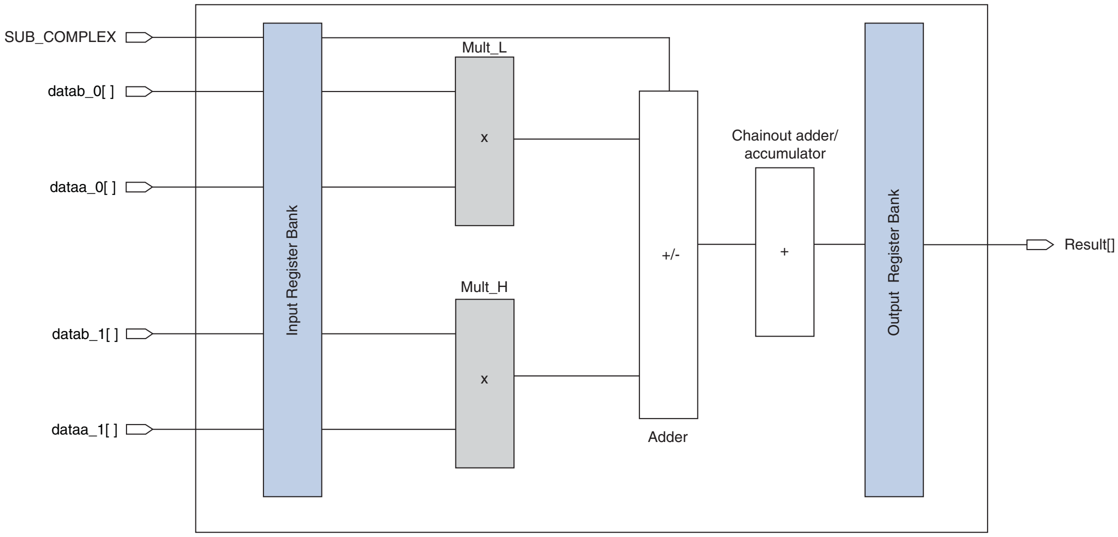
Figure 3–13. 27 × 27 Complex Multiplier with Four Variable Precision Blocks



Multiplier Adder Sum Mode

Stratix V devices support two-multiplier adder sum mode and four-multiplier adder sum mode. For a two-multiplier adder configuration, the Stratix V variable precision DSP blocks can support 16-bit, 18-bit, 27-bit, and 18 × 36 multipliers. You require two variable precision DSP blocks to implement 27-bit and 18 x 36 multiplier adder sum mode. Stratix V devices support one sum of four 18-bit multipliers with two variable precision DSP blocks. Figure 3–14 through Figure 3–17 show the variable precision DSP blocks in the multiplier adder sum mode.

Figure 3–14. One Sum of Two 18 x 18 Multipliers or Two 16 x 16 Multipliers for Stratix V Devices ***(Note 1)*, *(2)***



Notes to Figure 3–14:

(1) For 18-bit multiplier adder sum mode, the input data width is 18 bits and the output data width is 37 bits.

(2) For 16-bit multiplier adder sum mode, the input data width is 16 bits and the unused input bit requires padding with zeroes. The output data width is 33 bits.

Figure 3–15. One Sum of Two 27 x 27 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices

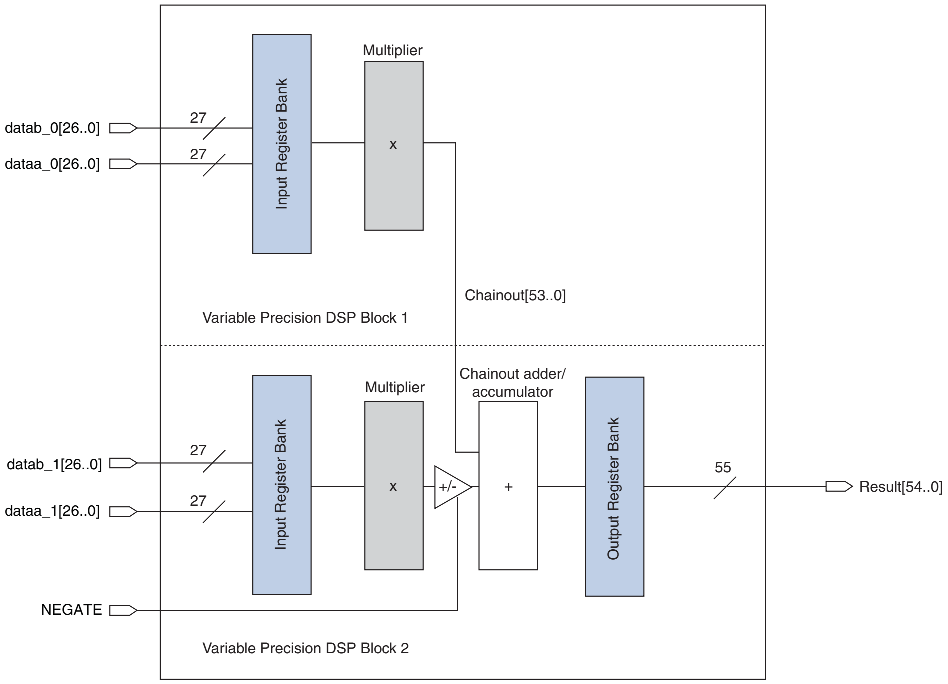


Figure 3–16. One Sum of Two 36 x 18 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices

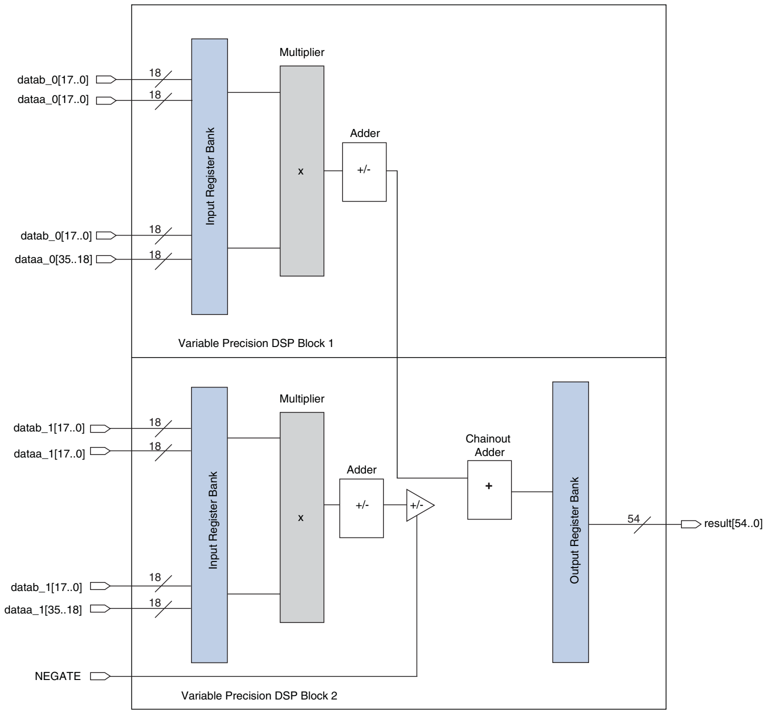
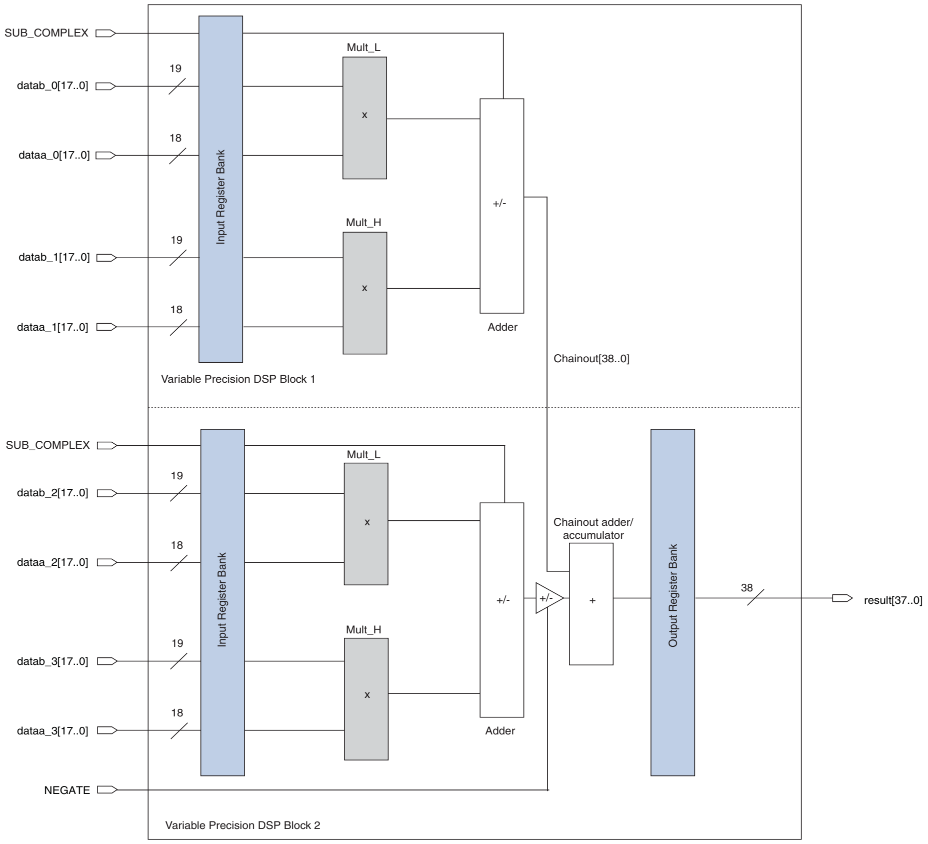


Figure 3–17. One Sum of Four 18 x 18 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices

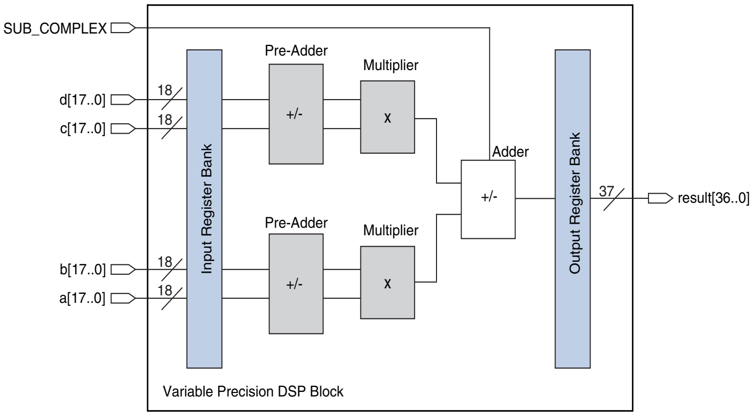


Sum of Square Mode

The Stratix V variable precision DSP block can implement one sum of square mode, $(a \pm b)^{2} \times (c \pm d)^{2}$. You can feed the four 18-bit inputs into the pre-adder block to convert b and d input as two's complement numbers to perform subtraction, if required. You can feed each 18-bit pre-adder block output into both multiplicand and multiplier inputs of an 18 x 18 multiplier to generate a square result.

Figure 3–18 shows the sum of square mode in a variable precision DSP block.

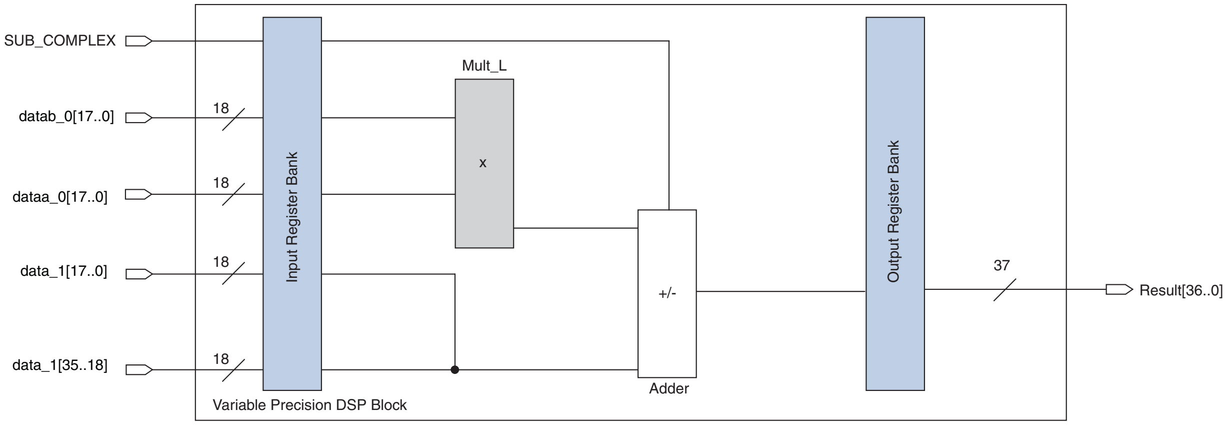
Figure 3–18. One Sum of Square Mode in a Variable Precision DSP Block for Stratix V Devices



18 x 18 Multiplication Summed with 36-Bit Input Mode

Stratix V variable precision DSP blocks support one 18 x 18 multiplication summed to a 36-bit input. You can use Mult_L to provide the input for an 18 x 18 multiplication, whereas Mult_H is bypassed. The `data1[17..0]` and `data1[35..18]` signals are concatenated to produce a 36-bit input. Figure 3–19 shows the 18 x 18 multiplication summed with the 36-bit input mode in a variable precision DSP block.

Figure 3–19. One 18 × 18 Multiplication Summed with 36-Bit Input Mode for Stratix V Devices

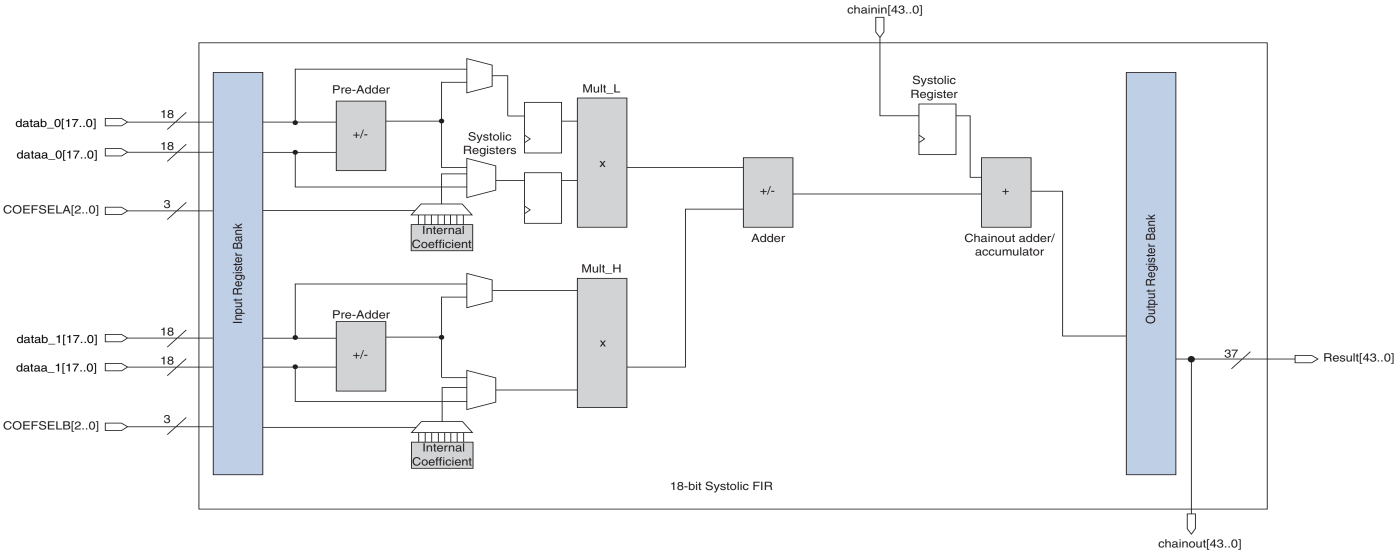


Systolic FIR Mode

Stratix V variable precision DSP blocks support 18-bit or 27-bit systolic FIR structures. Each input of the multiplier can come from three different sets of sources. They can be from two dynamic inputs, one dynamic input and one coefficient input, or one coefficient input and one pre-adder output. You can implement the 27-bit systolic FIR mode using 27-bit multiplier with chainout feature enabled. The chainout output for 18-bit systolic FIR mode is 44-bits whereas for 27-bit systolic FIR mode is 64-bit.

Figure 3–20 shows the 18-bit systolic FIR with two dynamic inputs.

Figure 3–20. 18-bit Systolic FIR Mode with Two Dynamic Inputs for Stratix V Devices



Variable Precision DSP Block Control Signals

The Stratix V variable precision DSP block has a total of 14 dynamic control signal inputs. The variable precision DSP block dynamic signals are user-configurable and can be set to toggle or not at run time.

Table 3–5 on page 3–22 lists the variable precision DSP block dynamic signals. The Stratix V variable precision DSP block supports 18-bit and 27-bit input cascading.

Table 3–5. Variable Precision DSP Block Dynamic Signals for Stratix V Devices

Signal Name	Function	Count
NEGATE	Control the operation of the decimation	1
LOADCONST	Preload an initial value to the accumulator	1
ACCUMULATE	Enable accumulation	1
SUB_COMPLEX	This signal has two functions: ■ Controls add or subtract of the two 18 x 18 multiplier results ■ Controls dynamic switch between 36 x 36 mode and complex 18 x 18	1
COEFSELA COEFSELB	Controls the internal coefficient select multiplexer along with select signals provided through the MSB of each 18-bit data input	2
CLK0 CLK1 CLK2	Variable precision DSP-block-wide clock signals	3
ENA0 ENA1 ENA2	Variable precision DSP-block-wide clock enable signals	3
ACLR0 ACLR1	Variable precision DSP-block-wide asynchronous clear signals	2
Total Count per DSP Block		**14**

Software Support

Altera provides two methods for implementing various modes of the Stratix V variable precision DSP block in a design: Using the Quartus II software and HDL inferring.

The following Quartus II megafunctions are supported for the Stratix V variable precision DSP blocks implementation:

- LPM_MULT
- ALTMULT_ADD
- ALTMULT_ACCUM
- ALTMULT_COMPLEX

Alternatively, you can infer the Stratix V variable precision DSP block from the HDL source code. You can use the HDL templates files that are available in the Quartus II software version 11.0 and later to get the Stratix V variable precision DSP blocks inferred. With either method, the Quartus II software maps the functionality to the variable precision DSP blocks during compilation.

For instructions about using the megafunctions and the templates files, refer to the *Quartus II Software Help*.

Document Revision History

Table 3–6 lists the revision history for this chapter.

Table 3–6. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Updated chapter for Quartus II software 11.0 release. ■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 3–1, Table 3–2, and Table 3–5. ■ Added Table 3–3. ■ Updated all figures in the chapter. ■ Added Figure 3–3. ■ Updated "Software Support" section.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

4. Clock Networks and PLLs in Stratix V Devices

SV51005-1.2

This chapter describes the hierarchical clock networks and phase-locked loops (PLLs) that have advanced features in Stratix® V devices. It includes information about reconfiguring the PLL counter, clock frequency, and phase shift in real time, which allows 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 chapter contains the following sections:

Clock Networks in Stratix V Devices

The global clock networks (GCLKs), regional clock networks (RCLKs), and periphery clock networks (PCLKs) available in Stratix V devices are organized into hierarchical clock structures. The clock networks provide up to 417 unique clock domains (16 GCLKs + 92 RCLKs + 309 PCLKs) within the Stratix V device and allow up to 122 unique GCLK, RCLK, and PCLK clock sources (16 GCLKs + 23 RCLKs + 83 PCLKs) per device quadrant.

Table 4–1 lists the clock resources available in Stratix V devices.

Table 4–1. Clock Resources in Stratix V Devices—Preliminary

Clock Resource	Number of Resources Available	Source of Clock Resource
Clock input pins	48 Single-ended (24 Differential) *(1)* or 56 Single-ended (28 Differential) *(2)*	`CLK[0..23]p` and `CLK[0..23]n` pins *(1)* `CLK[0..27]p` and `CLK[0..27]n` pins *(2)*
GCLK networks	16	`CLK[0..27]p` and `CLK[0..27]n` pins, PLL clock outputs, and logic array
RCLK networks	92	`CLK[0..27]p` and `CLK[0..27]n` pins, PLL clock outputs, and logic array
PCLK networks	228, 282, 306, and 309 *(3)*	DPA clock outputs, PLD-transceiver interface clocks, I/O pins, and logic array
GCLKs and RCLKs per quadrant	39	16 GCLKs + 23 RCLKs
GCLKs and RCLKs per device	108	16 GCLKs + 92 RCLKs

Notes to Table 4–1:

(1) This applies to all Stratix V devices except 5SGSD6, 5SGSD8 devices.

(2) This only applies to 5SGSD6 and 5SGSD8 devices.

(3) There are 228 PCLKs in 5SGXA3 and 5SGXA4 devices, 282 PCLKs in 5SGXB5 and 5SGXB6 devices, 306 PCLKs in 5SGXA5 and 5SGXA7 devices, and 309 PCLKs in 5SGSD6 and 5SGSD8 devices.

Stratix V devices have up to 56 dedicated single-ended clock pins or 28 dedicated differential clock pins (`CLK[0..27]p` and `CLK[0..27]n`) that can drive either the GCLK or RCLK networks. Table 4–2 on page 4–9 and Table 4–3 on page 4–10 list the clock input pins connectivity to the GCLK and RCLK networks, respectively.

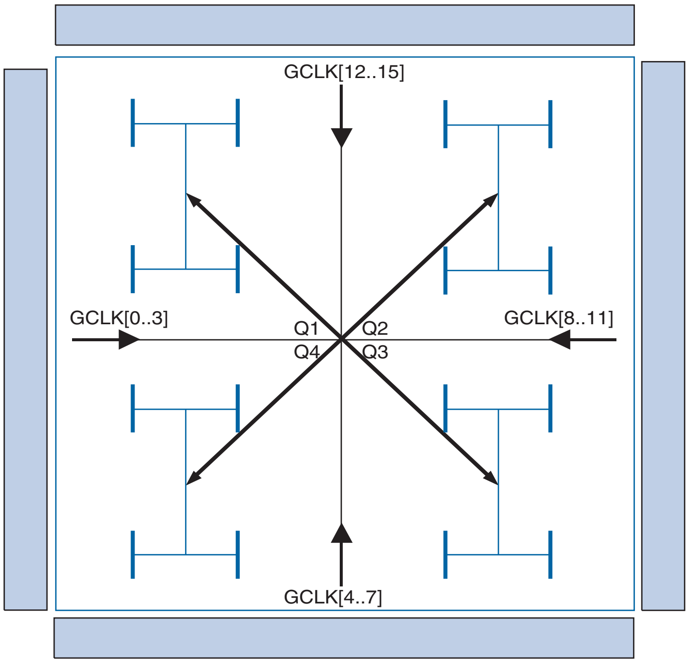
For more information about how to connect the clock input pins, refer to the *Stratix V Device Family Pin Connection Guidelines*.

Global Clock Networks

Stratix V 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, embedded memory blocks, and PLLs. Stratix V 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 4–1 shows the GCLK networks in Stratix V devices.

Figure 4–1. GCLK Networks

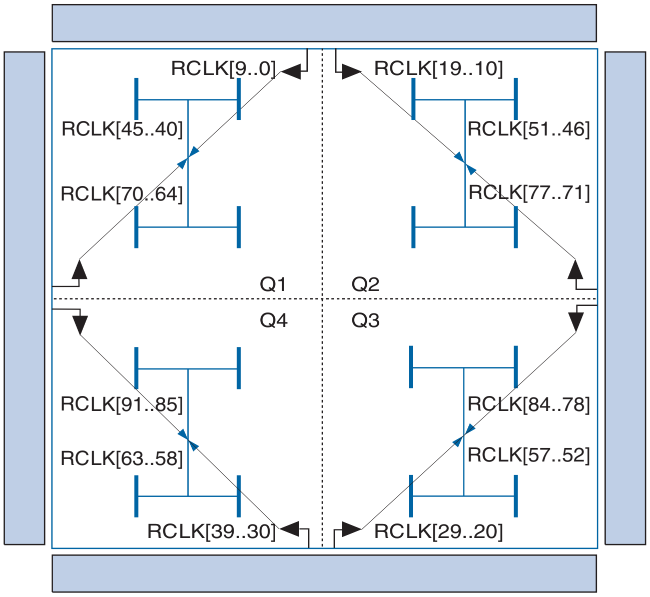


Regional Clock Networks

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 V 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 4–2 shows the RCLK networks in Stratix V devices.

Figure 4–2. RCLK Networks



Periphery Clock Networks

The PCLK networks shown in Figure 4–3 through Figure 4–6 on page 4–5 are collections of individual clock networks driven from the periphery of the Stratix V device. Depending on the routing direction, there are vertical PCLKs from the top and bottom periphery and horizontal PCLKs from the left and right periphery. Clock outputs from the dynamic phase aligner (DPA) block, programmable logic device (PLD)-transceiver interface clocks, 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 V 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 4–3. PCLK Networks—5SGXA3 and 5SGXA4 Devices

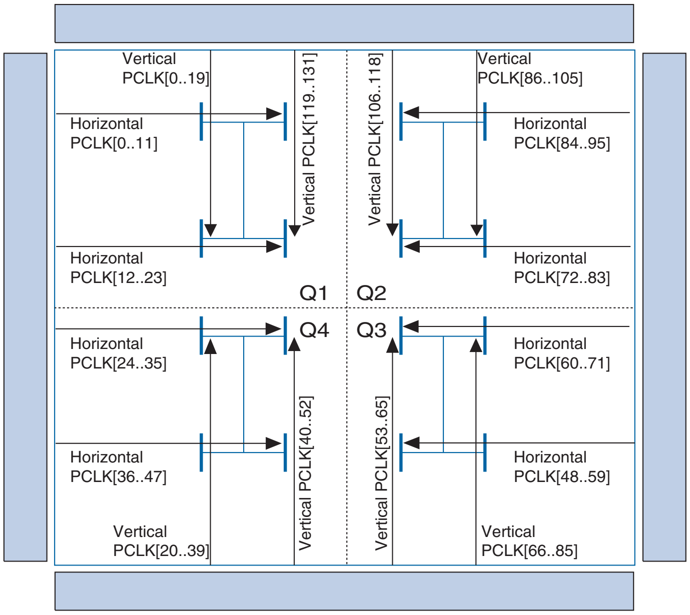


Figure 4–4. PCLK Networks—5SGXB5 and 5SGXB6 Devices

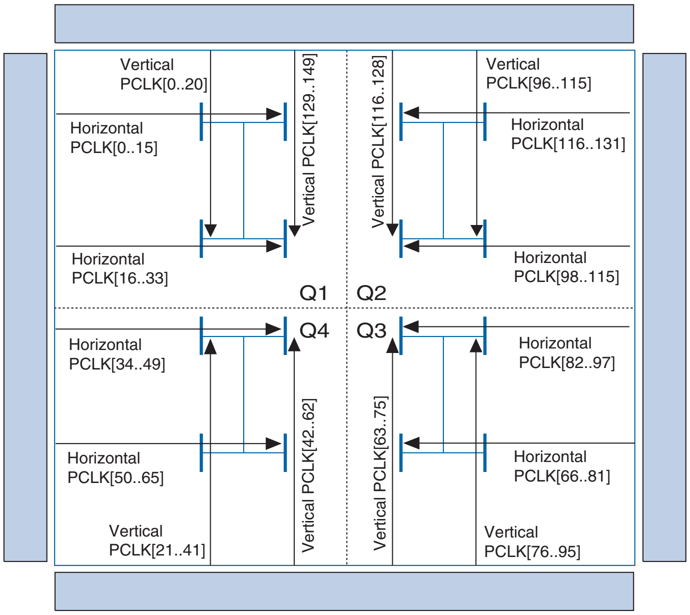


Figure 4–5. PCLK Networks—5SGXA5 and 5SGXA7 Devices

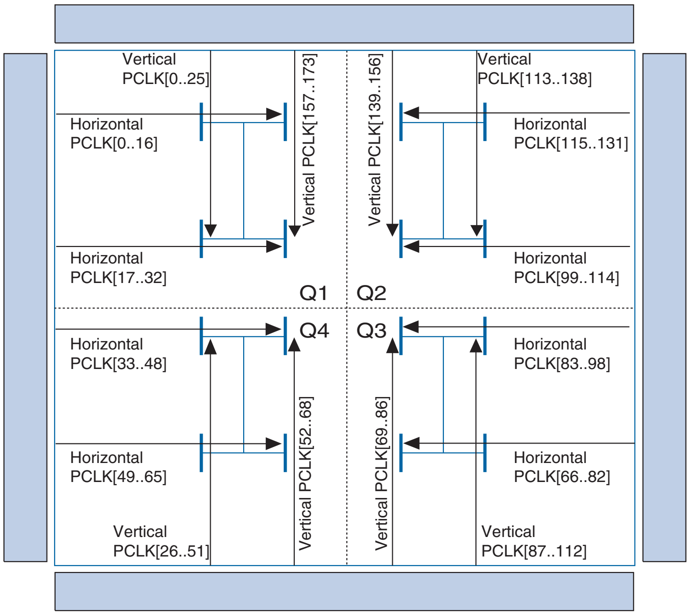
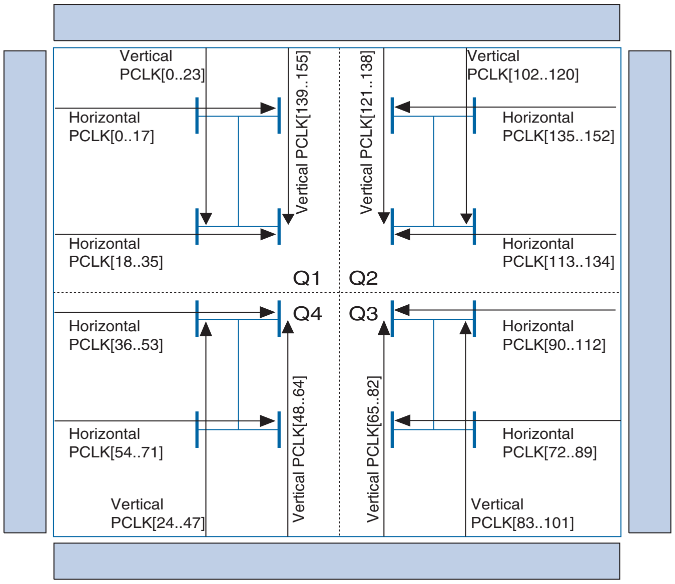


Figure 4–6. PCLK Networks—5SGSD6 and 5SGSD8 Devices

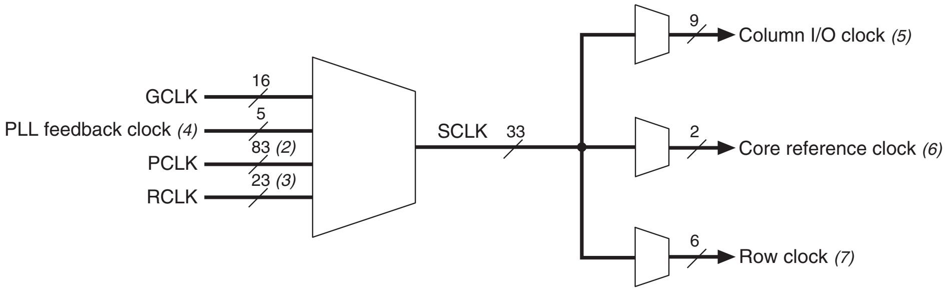


Clock Sources Per Quadrant

There are 33 section clock (SCLK) networks available in each spine clock that can drive six row clocks in each logic array block (LAB) row, nine column I/O clocks, and two core reference clocks. The SCLKs are the clock resources to the core functional blocks, PLLs, and I/O interfaces of the device. Figure 4–7 shows SCLKs driven by the GCLK, RCLK, PCLK, or the PLL feedback clock networks in each spine clock.

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

Figure 4–7. Hierarchical Clock Networks Per Spine Clock *(Note 1)*



Notes to Figure 4–7:

(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 83 PCLKs that can drive the SCLKs in each spine clock in the largest device.

(3) There are up to 23 RCLKs that can drive the SCLKs in each spine clock in the largest device.

(4) The PLL feedback clock is the clock from the PLL that drives into the SCLKs.

(5) The column I/O clock is the clock that drives the column I/O core registers and I/O interfaces.

(6) The core reference clock is the clock that feeds into the PLL as the PLL reference clock.

(7) The row clock is the clock source to the LAB, memory blocks, and row I/O interfaces in the core row.

Clock Regions

Stratix V devices provide the following types of clock regions:

- "Entire Device Clock Region" on page 4–7
- "Regional Clock Region" on page 4–7
- "Dual-Regional Clock Region" on page 4–7

Entire Device Clock Region

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.

Regional Clock Region

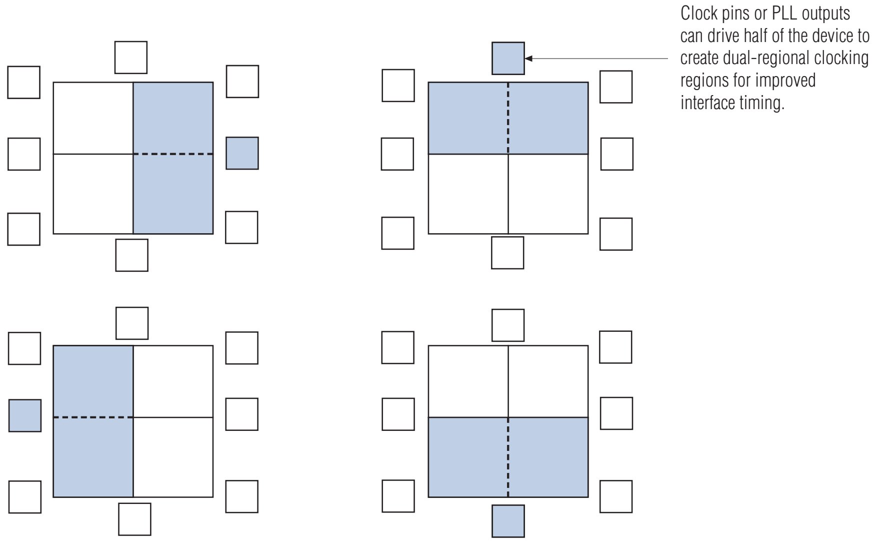
To form a RCLK region, a source drives a single quadrant of the device. This clock region provides the lowest skew within a quadrant and is a good option if all the destinations are within a single device quadrant.

Dual-Regional Clock Region

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 4–8 shows the dual-regional clock region.

Figure 4–8. Dual-Regional Clock Region for Stratix V Devices


Clock pins or PLL outputs can drive half of the device to create dual-regional clocking regions for improved interface timing.

Clock Network Sources

In Stratix V devices, clock input pins, PLL outputs, high-speed serial interface (HSSI) outputs, DPA outputs, and internal logic can drive the GCLK and RCLK networks. For connectivity between the dedicated clock pins, GCLK, and RCLK networks, refer to Table 4–2 and Table 4–3 on page 4–10.

Dedicated Clock Input Pins

`CLK` pins can be either differential clocks or single-ended clocks. Stratix V devices support up to 28 differential clock inputs or 56 single-ended clock inputs. You can also use dedicated clock input pins `CLK[27..0]` for high fan-out control signals such as asynchronous clears, presets, and clock enables for protocol signals through the GCLK or RCLK networks.

Internal Logic

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

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

DPA Outputs

Every DPA generates one PCLK to the core.

HSSI Outputs

Every three HSSI outputs generate a group of six PCLKs to the core.

For more information about DPA and HSSI outputs, refer to the *High-Speed Differential I/O Interfaces with DPA in Stratix V Devices* chapter.

PLL Clock Outputs

Stratix V PLL clock outputs can drive both GCLK and RCLK networks.

Clock Input Pin Connections to GCLK and RCLK Networks

Table 4–2 lists the connection between the dedicated clock input pins and GCLKs.

Table 4–2. Clock Input Pin Connectivity to the GCLK Networks—Preliminary

Clock Resources	CLK (p/n Pins)																											
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
GCLK0	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK1	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK2	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK3	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—
GCLK4	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK5	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK6	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK7	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
GCLK8	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—	—	✓ *(1)*	✓ *(1)*	✓ *(1)*	✓ *(1)*
GCLK9	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—	—	✓ *(1)*	✓ *(1)*	✓ *(1)*	✓ *(1)*
GCLK10	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—	—	✓ *(1)*	✓ *(1)*	✓ *(1)*	✓ *(1)*
GCLK11	—	—	—	—	—	—	—	—	✓	✓	✓	✓	✓	✓	✓	✓	—	—	—	—	—	—	—	—	✓ *(1)*	✓ *(1)*	✓ *(1)*	✓ *(1)*
GCLK12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK13	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK14	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—
GCLK15	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	✓	✓	✓	—	—	—	—	—	—	—	—

Note to Table 4–2:

(1) This is only applicable to 5SGSD6 and 5SGSD8 devices.

Table 4–3 lists the connectivity between the dedicated clock input pins and RCLKs in Stratix V devices. A given clock input pin can drive two adjacent RCLK networks to create a dual-regional clock network.

Table 4–3. Clock Input Pin Connectivity to the RCLK Networks (Part 1 of 3)—Preliminary

Clock Resources	CLK (p/n pins)																											
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
RCLK [58,59,60,61,62, 63,64,68,85,89]	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [58,59,60,61,62, 63,65,69,86,90]	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [58,59,60,61,62, 63,66,70,87,91]	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [58,59,60,61,62, 63,67,88]	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [20,24,28,30,34, 38]	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [21,25,29,31,35, 39]	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [22,26,32,36]	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [23,27,33,37]	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [52,53,54,55,56, 57,71,75,78,82]	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [52,53,54,55,56, 57,72,76,79,83]	—	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
RCLK [52,53,54,55,56, 57,73,77,80,84]	—	—	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 4–3. Clock Input Pin Connectivity to the RCLK Networks (Part 2 of 3)—Preliminary

Clock Resources	CLK (p/n pins)																											
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
`RCLK [52,53,54,55,56,57,74,81]`	—	—	—	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`RCLK [46,47,48,49,50,51,71,75,78,82]`	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`RCLK [46,47,48,49,50,51,72,76,79,83]`	—	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—	—	—
`RCLK [46,47,48,49,50,51,73,77,80,84]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—	—
`RCLK [46,47,48,49,50,51,74,81]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(1)*	—	—	—	—	—	—	—	—	—	—	—	—
`RCLK [0,4,8,10,14,18]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—	—
`RCLK [1,5,9,11,15,19]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—	—
`RCLK [2,6,12,16]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—	—
`RCLK [3,7,13,17]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—	—
`RCLK [40,41,42,43,44,45,64,68,85,89]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
`RCLK [40,41,42,43,44,45,65,69,86,90]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
`RCLK [40,41,42,43,44,45,66,70,87,91]`	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—	—

Table 4–3. Clock Input Pin Connectivity to the RCLK Networks (Part 3 of 3)—Preliminary

Clock Resources	CLK (p/n pins)																											
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
RCLK [40,41,42,43,44,45,67,88]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓	—	—	—	—
RCLK [71,75,78,82]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(2)*	—	—	—
RCLK [72,76,79,83]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(2)*	—	—
RCLK [73,77,80,84]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(2)*	—
RCLK [74,81]	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	✓ *(2)*

Notes to Table 4–3:

(1) This is applicable to all Stratix V devices except 5SGSD6 and 5SGSD8 devices.

(2) This is only applicable to 5SGSD6, 5SGSD8 devices.

Clock Output Connections

For Stratix V PLL connectivity to GCLK and RCLK networks, refer to *PLL Connectivity to GCLK and RCLK Networks for Stratix V Devices*.

Clock Control Block

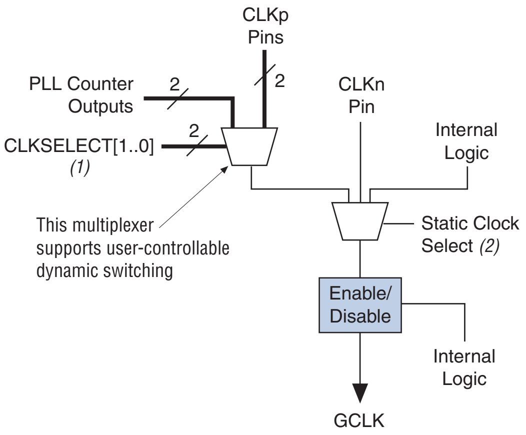
Every GCLK, RCLK, and PCLK network has its own clock control block. The control block provides the following features:

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

Figure 4–9, Figure 4–10, and Figure 4–11 show the GCLK, RCLK, and PCLK control 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 4–9. GCLK Control Block for Stratix V Devices



Notes to Figure 4–9:

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

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

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.

☞ 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.

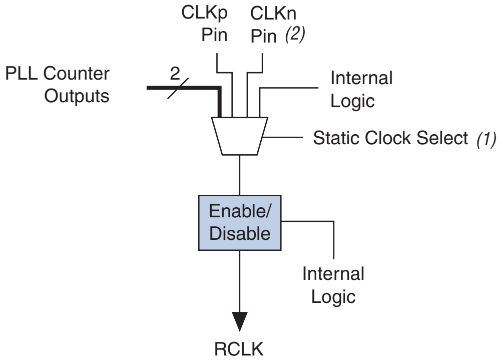
For more information, refer to the *Clock Control Block (ALTCLKCTRL) Megafunction User Guide*.

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 V device
- `inclk[2]`—can be fed by PLL counters C0 and C2 from the two center PLLs on the same side of the Stratix V device
- `inclk[3]`—can be fed by PLL counters C1 and C3 from the two center PLLs on the same side of the Stratix V device

☞ Corner PLLs cannot be used for dynamic clock control selection.

Figure 4–10. RCLK Control Block



Notes to Figure 4–10:

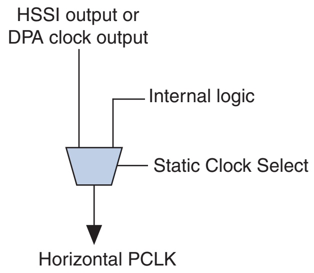
(1) When the device is in user mode, you can only set the clock select signals through a configuration file (**.sof** or **.pof**); they cannot be dynamically controlled.

(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 select the HSSI output or internal logic to drive the HSSI horizontal PCLK control block. Alternatively, you can also use the DPA clock output or internal logic to drive the DPA horizontal PCLK. You can only use the DPA output to generate the vertical PCLK to the core.

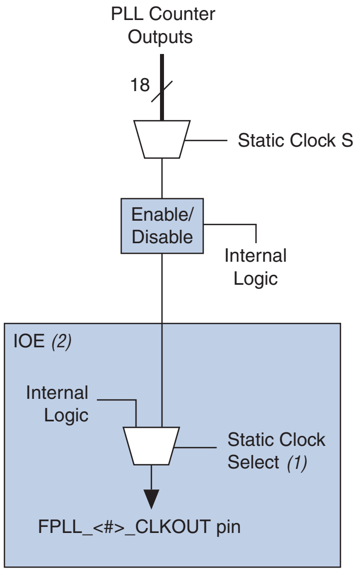
Figure 4–11. Horizontal PCLK Control Block



You can power down the Stratix V GCLK and RCLK 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, RCLK, and PCLK 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. Figure 4–9 and Figure 4–10 show that this function is independent of the PLL and is applied directly on the clock network.

You can enable or disable the dedicated external clock output pins using the ALTCLKCTRL megafunction. Figure 4–12 shows the external PLL output clock control block.

Figure 4–12. External PLL Output Clock Control Block for Stratix V Devices



Notes to Figure 4–12:

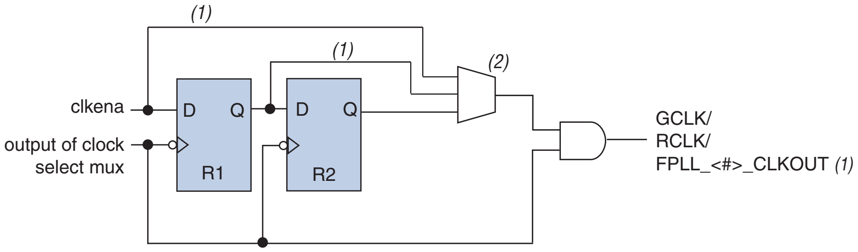
(1) When the device is in user mode, you can only set the clock select signals through a configuration file (**.sof** or **.pof**); they cannot be dynamically controlled.

(2) The clock control block feeds to a multiplexer within the FPLL_<#>_CLKOUT pin's IOE. The FPLL_<#>_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 4–13 shows how the clock enable and disable circuit of the clock control block is implemented in Stratix V devices.

Figure 4–13. clkena Implementation



Notes to Figure 4–13:

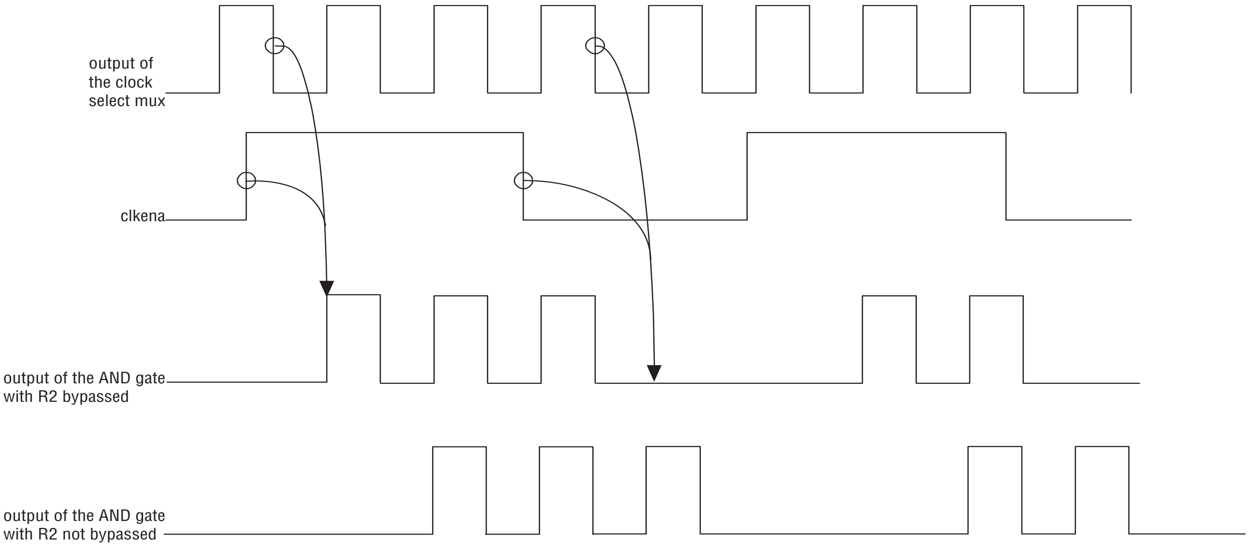
(1) The R1 and R2 bypass paths are not available for the PLL external clock outputs.

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

In Stratix V 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 4–14 shows a waveform example for a clock output enable. clkena is synchronous to the falling edge of the clock output.

Stratix V 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 4–14. clkena Signals ***(Note 1)***


output of the clock select mux
clkena
output of the AND gate with R2 bypassed
output of the AND gate with R2 not bypassed

Note to Figure 4–14:

(1) You can use the clkena signals to enable or disable the GCLK and RCLK networks or the FPLL_<#>_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 overshoot during resynchronization.

Stratix V PLLs

Stratix V device family introduces the fractional PLLs in addition to the existing integer PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces. Each fractional PLL has 18 output counters that supports integer or fractional frequency synthesis.

Stratix V devices offer up to 28 fractional PLLs in the larger densities. All Stratix V fractional PLLs have the same core analog structure and features support.

Table 4–4 lists the features of PLLs in Stratix V devices.

Table 4–4. PLL Features for Stratix V Devices —Preliminary

Feature	Stratix V
Integer PLL	Yes
Fractional PLL	Yes
C output counters	18
M, N, C counter sizes	1 to 512
Dedicated external clock outputs	4 single-ended or 2 single-ended and 1 differential
Clock input pins	4 single-ended or 4 differential
External feedback input pin	Single-ended or differential
Spread-spectrum input clock tracking	Yes (1)
Source synchronous compensation	Yes
Direct compensation	Yes
Normal compensation	Yes
Zero delay buffer (ZDB) compensation	Yes
External feedback compensation	Yes
LVDS compensation	Yes
VCO output drives the DPA clock	Yes
Phase shift resolution	78.125 ps (2)
Programmable duty cycle	Yes

Notes to Table 4–4:

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

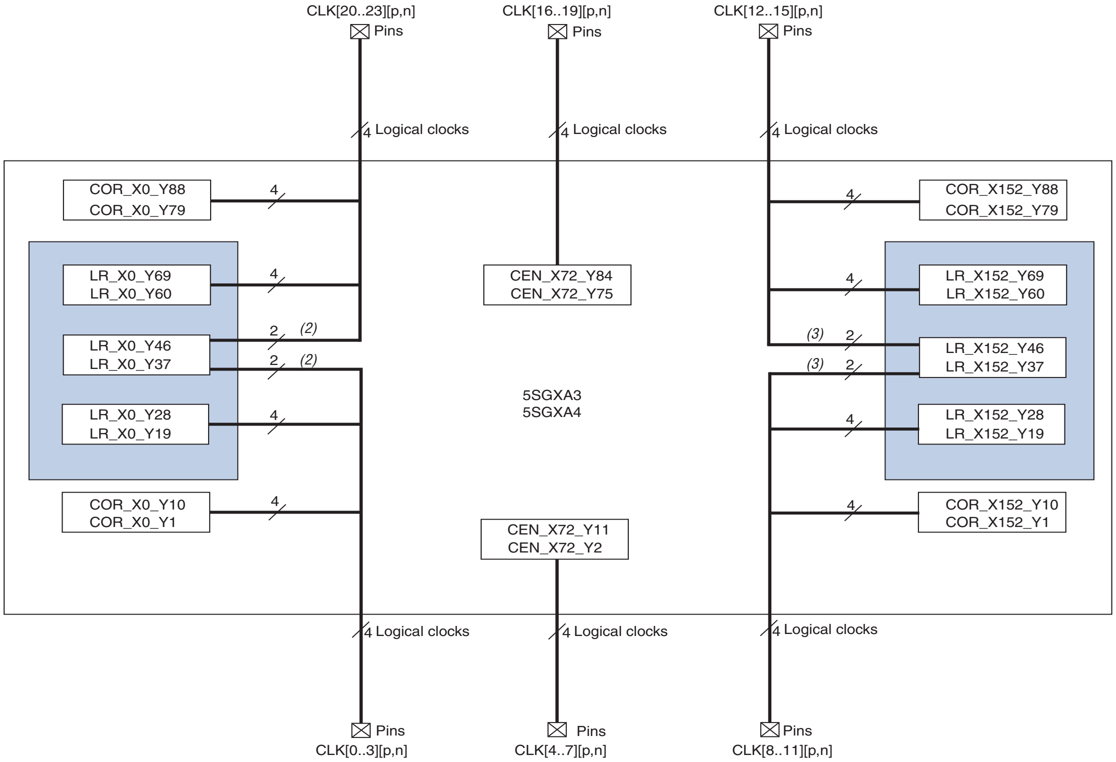
(2) The smallest phase shift is determined by the voltage-controlled oscillator (VCO) period divided by eight. For degree increments, the Stratix V device can shift all output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the frequency and divide parameters.

Figure 4–15 through Figure 4–18 on page 4–22 show the physical locations of the fractional PLLs.

The nomenclature for Stratix V PLLs follow their geographical coordinates in the device floor plan. The PLLs that reside on the center of the device are named CEN_X<#>_Y<#>, the PLLs that reside on the corner of the device are named COR_X<#>_Y<#>, and the PLLs that reside on the left and right sides of the device are named LR_X<#>_Y<#>.

Figure 4–15 shows the PLL locations for 5SGXA3 and 5SGXA4 devices.

Figure 4–15. PLL Locations for 5SGXA3 and 5SGXA4 Devices *(Note 1)*



Notes to Figure 4–15:

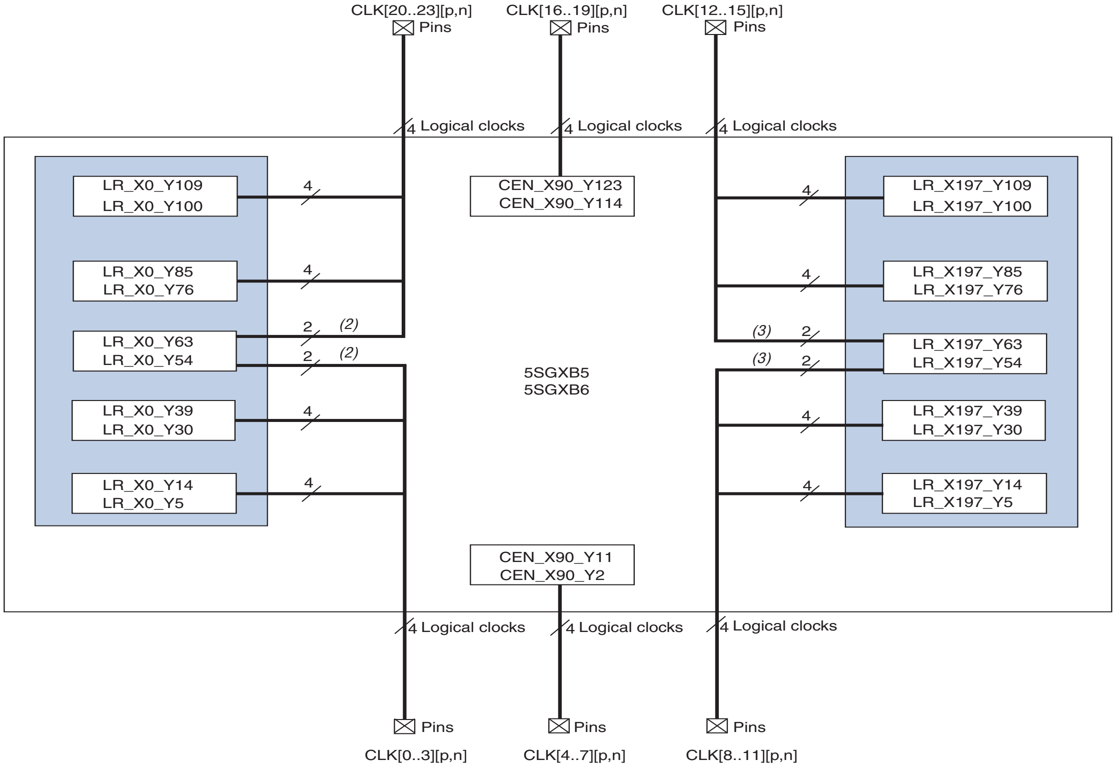
(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

(2) CLK0, CLK1, CLK20, and CLK21 clock pins feed into fractional PLL LR_X0_Y37 and fractional PLL LR_X0_Y46.

(3) CLK8, CLK9, CLK12, and CLK13 clock pins feed into fractional PLL LR_X152_Y37 and fractional PLL LR_X152_Y46.

Figure 4–16 shows the PLL locations for 5SGXB5 and 5SGXB6 devices.

Figure 4–16. PLL Locations for 5SGXB5 and 5SGXB6 Devices ***(Note 1)***



Notes to Figure 4–16:

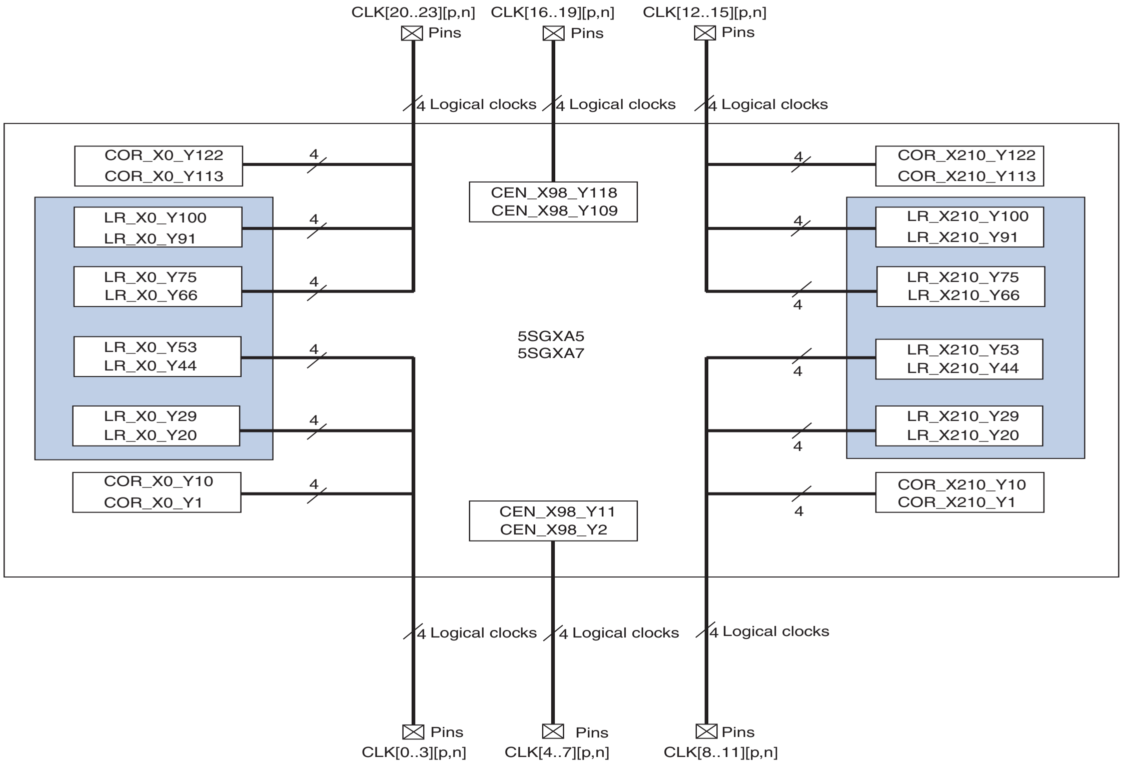
(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

(2) `CLK0`, `CLK1`, `CLK20`, and `CLK21` clock pins feed into fractional PLL `LR_X0_Y54` and fractional PLL `LR_X0_Y63`.

(3) `CLK8`, `CLK9`, `CLK12`, and `CLK13` clock pins feed into fractional PLL `LR_X197_Y54` and fractional PLL `LR_X197_Y63`.

Figure 4–17 shows the PLL locations for 5SGXA5 and 5SGXA7 devices.

Figure 4–17. PLL Locations for 5SGXA5 and 5SGXA7 Devices *(Note 1)*

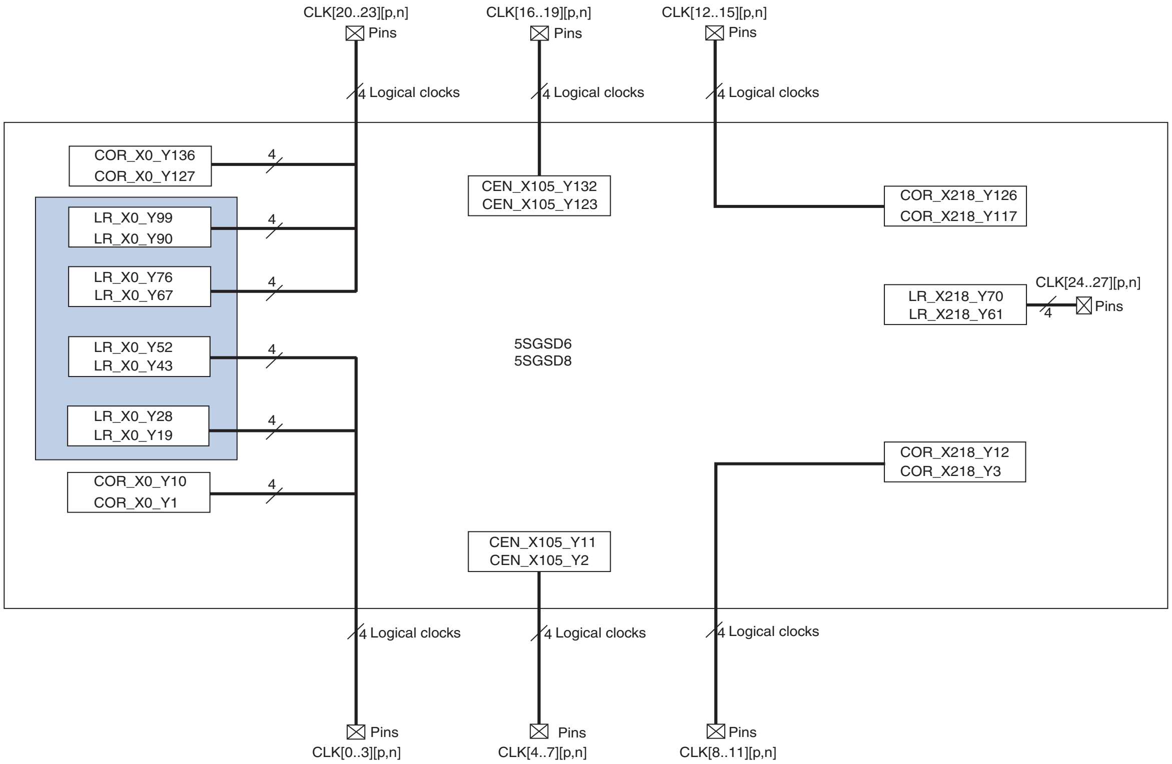


Note to Figure 4–17:

(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

Figure 4–18 shows the PLL locations for 5SGSD6 and 5SGSD8 devices.

Figure 4–18. PLL Locations for 5SGSD6 and 5SGSD8 Devices *(Note 1)*



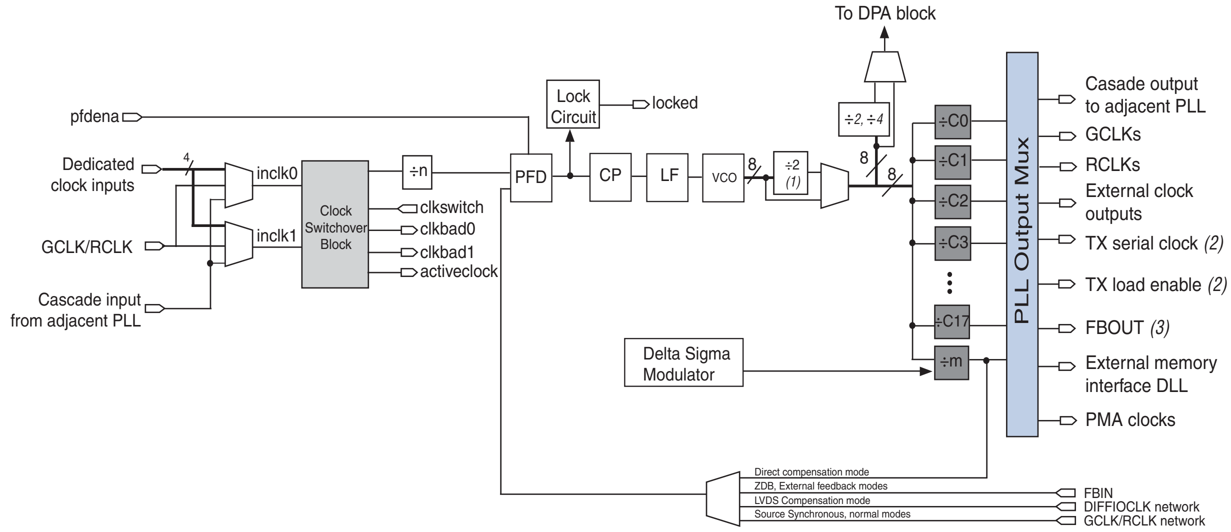
Note to Figure 4–18:

(1) Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II software Chip Planner.

Fractional PLL Architecture

Figure 4–19 shows the high-level block diagram of the Stratix V fractional PLL.

Figure 4–19. Fractional PLL High-Level Block Diagram



Notes to Figure 4–19:

(1) This is the VCO post-scale counter `K`.

(2) Only `C0`, `C2`, `C15`, and `C17` can drive the TX serial clock and `C1`, `C3`, `C14`, and `C16` can drive the TX load enable.

(3) This `FBOUT` port is fed by the `M` counter in the Stratix V PLLs.

Fractional PLL Usage

You can configure the fractional PLL as either an integer or an enhanced fractional mode. One fractional PLL can use up to 18 output counters and all external clock outputs. Two adjacent fractional PLLs share the 18 output counters.

You can use fractional PLLs to reduce the number of oscillators required on the board, as well as to reduce the clock pins used in the FPGA by synthesizing multiple clock frequencies from a single reference clock source. In addition, you can use fractional PLLs for clock network delay compensation, zero-delay buffering, and transmit clocking for transceivers.

PLL External Clock I/O Pins

Two adjacent corner and center fractional PLLs share four dual-purpose clock I/O pins, organized as one of the following combinations:

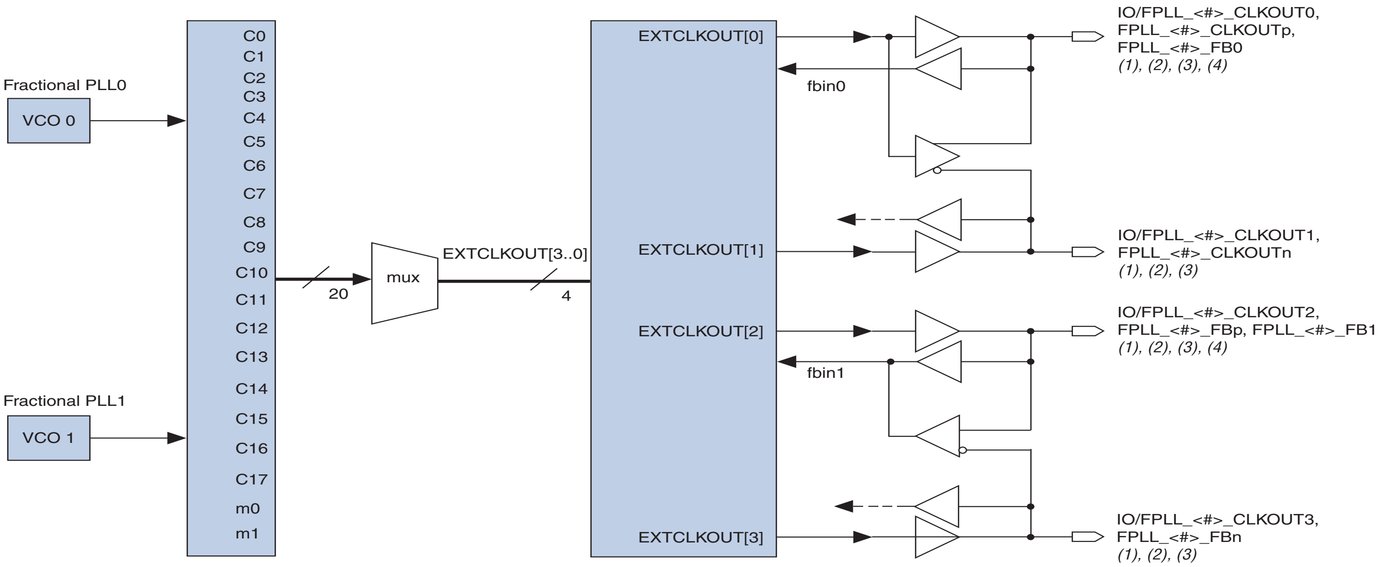
- Four single-ended clock outputs
- Two single-ended outputs and one differential clock output
- Four single-ended clock outputs and two single-ended feedback inputs within the I/O driver feedback for ZDB support
- Two single-ended clock outputs and two single-ended feedback inputs for single-ended External Feedback (EFB) support

- One differential clock output and one differential feedback input for differential EFB support (only one of the two adjacent fractional PLLs can support differential EFB at one time while the other fractional PLL can be used for general-purpose clocking)

All left and right fractional PLLs in Stratix V devices do not support external clock outputs except the right fractional PLLs in 5SGSD6 and 5SGSD8devices.

Figure 4–20 shows the dual-purpose clock I/O pins associated with the PLL for Stratix V devices.

Figure 4–20. Dual-Purpose Clock I/O Pins Associated with PLL for Stratix V Devices



Notes to Figure 4–20:

(1) You can feed these clock output pins using any one of the C[17..0] or m counters. When not used as external clock outputs, these clock output pins can be used as regular user I/Os.

(2) The FPLL_<#>_CLKOUT0, FPLL_<#>_CLKOUT1, FPLL_<#>_CLKOUT2, and FPLL_<#>_CLKOUT3 pins are single-ended clock output pins.

(3) The FPLL_<#>_CLKOUTp and FPLL_<#>_CLKOUTn pins are differential output pins while the FPLL_<#>_FBp and FPLL_<#>_FBn pins are differential feedback input pins to support differential EFB.

(4) The FPLL_<#>_FB0 and FPLL_<#>_FB1 pins are single-ended feedback input pins.

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

Figure 4–20 shows that any of the output counters (C[17..0]) on the PLLs or the M counter can feed the dedicated external clock outputs. Therefore, one counter or frequency can drive all output pins available from a given PLL.

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 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 V Devices* chapter.

Stratix V 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 require external PLL clocking.

PLL Control Signals

You can use the `pfdena`, `areset`, and `locked` signals to control and observe 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) after a loss-of-lock condition using the Quartus II MegaWizard™ Plug-In Manager. You must include the `areset` signal 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 must 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 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

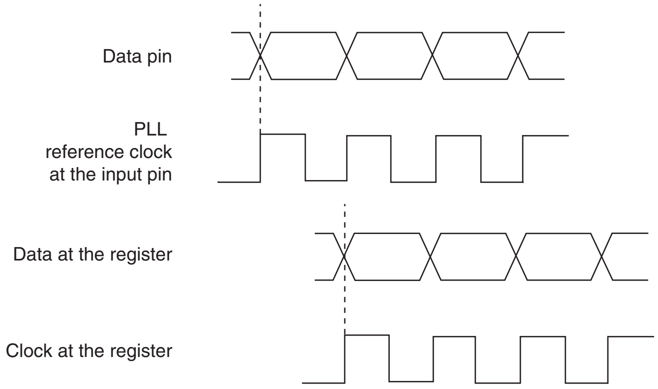
Stratix V PLLs support up to six different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle.

☞ 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. When a RCLK or GCLK network drives the PLL or the PLL is driven by a dedicated clock pin that is not associated with the PLL, the input and output delays may not be fully compensated in the Quartus II software. An example is when you configure a PLL in zero-delay buffer mode and the PLL input is driven by an associated dedicated clock input pin. In this configuration, a fully compensated clock path results in zero delay between the clock input and one of the output clocks from the PLL. However, 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. Refer to Figure 4–15 on page 4–19 through Figure 4–18 on page 4–22 for a mapping of dedicated clock pins to their associated PLLs.

Source Synchronous Mode

If the 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 4–21 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 4–21. 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 V 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.

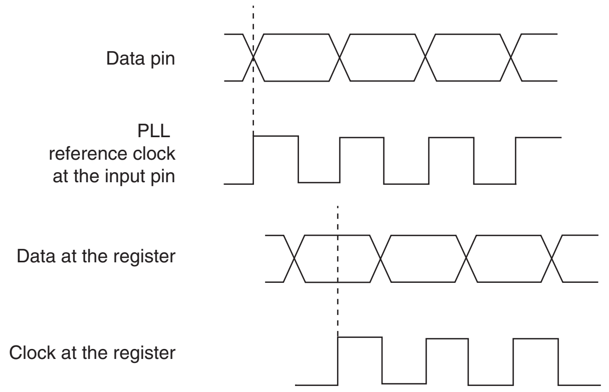
Source Synchronous Mode for LVDS Compensation

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 4–22 shows an example waveform of the clock and data in LVDS mode.

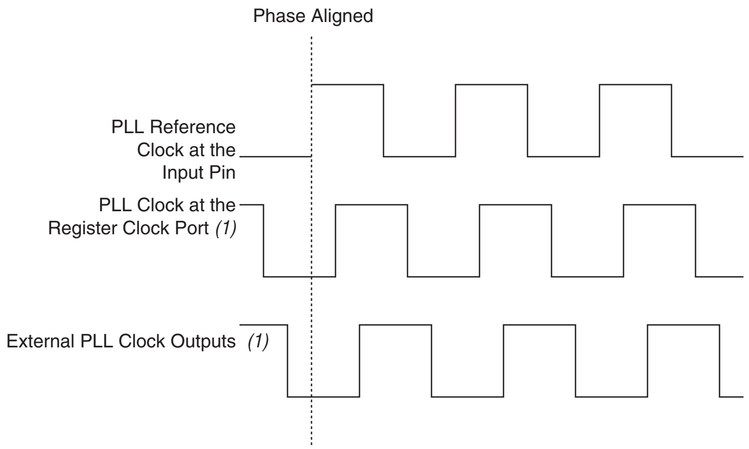
Figure 4–22. Phase Relationship Between the Clock and Data in LVDS Mode



Direct Compensation Mode

In direct 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 4–23 shows an example waveform of the PLL clocks' phase relationship in direct compensation mode.

Figure 4–23. Phase Relationship Between the PLL Clocks in Direct Compensation Mode


Phase Aligned
PLL Reference Clock at the Input Pin
PLL Clock at the Register Clock Port (1)
External PLL Clock Outputs (1)

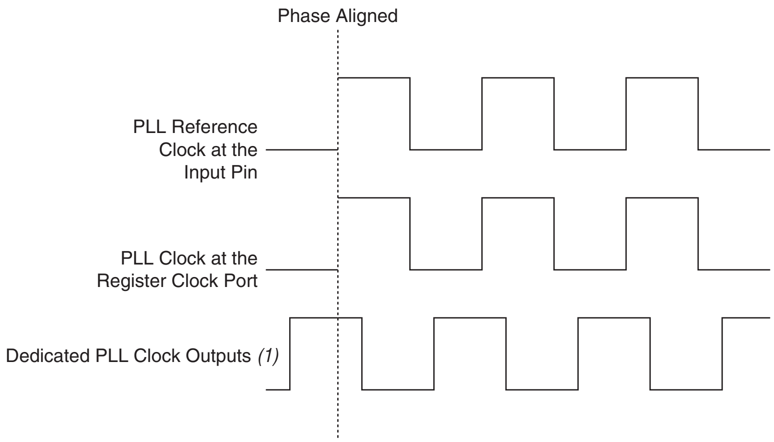
Note to Figure 4–23:

(1) The PLL clock outputs lag 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 4–24 shows an example waveform of the PLL clocks' phase relationship in normal mode.

Figure 4–24. Phase Relationship Between the PLL Clocks in Normal Mode


Phase Aligned
PLL Reference Clock at the Input Pin
PLL Clock at the Register Clock Port
Dedicated PLL Clock Outputs (1)

Note to Figure 4–24:

(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 to guarantee clock alignment at the input and output pins. ZDB mode is supported only on the center and corner PLLs in Stratix V devices.

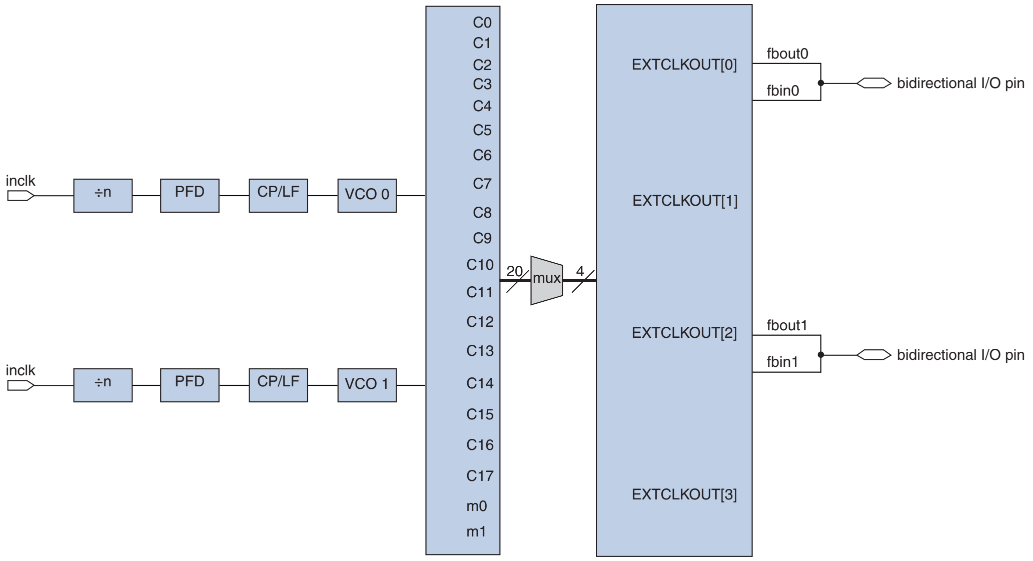
When using Stratix V PLLs in ZDB mode, along with single-ended I/O standards, to ensure phase alignment between the CLK pin and the external clock output (CLKOUT) pin, you must instantiate a bi-directional I/O pin in the design to serve as the feedback path connecting the FBOUT and FBIN ports of the PLL. The PLL uses this bidirectional 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.

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

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

Figure 4–25 shows ZDB mode in Stratix V PLLs. When using ZDB mode, you cannot use differential I/O standards on the PLL clock input or output pins.

Figure 4–25. ZDB Mode in Stratix V PLLs *(Note 1)*

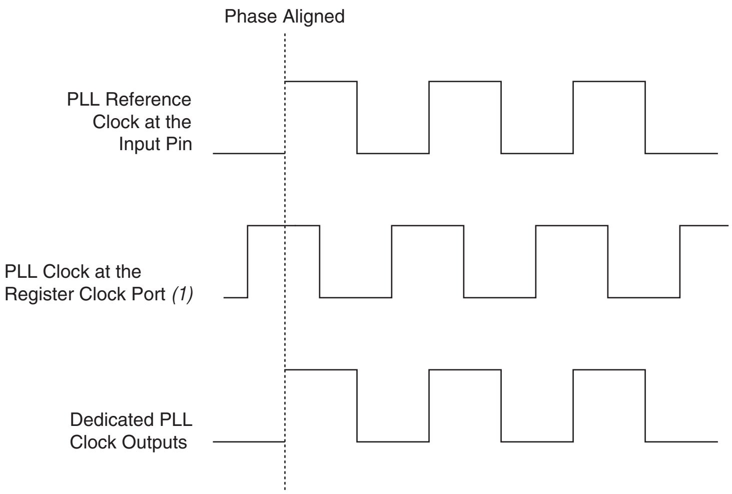


Note to Figure 4–25:

(1) ZDB mode can support up to four single-ended clock outputs. For more information, refer to Figure 4–20 on page 4–24.

Figure 4–26 shows an example waveform of the PLL clocks' phase relationship in ZDB mode.

Figure 4–26. Phase Relationship Between the PLL Clocks in ZDB Mode


Phase Aligned
PLL Reference Clock at the Input Pin
PLL Clock at the Register Clock Port (1)
Dedicated PLL Clock Outputs

Note to Figure 4–26:

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

External Feedback Mode

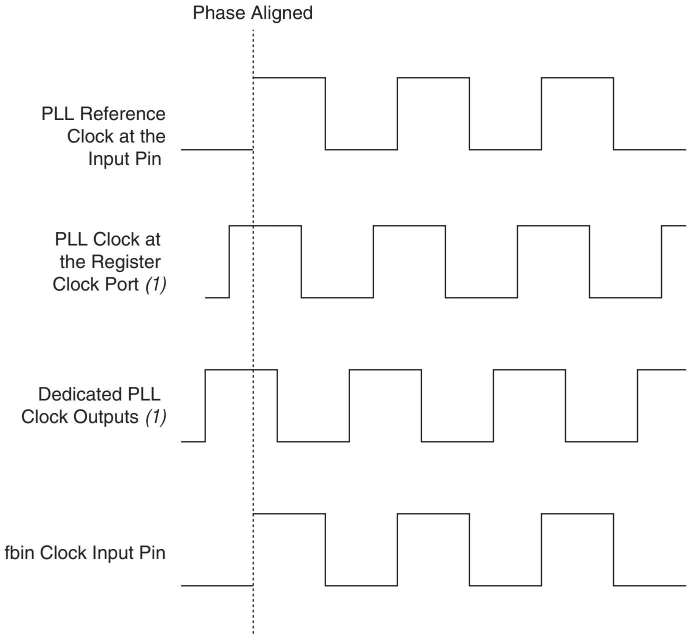
Figure 4–27 shows that in external feedback mode, the external feedback input pin (`fbin`) is phase-aligned with the clock input pin. Aligning these clocks allows you to remove clock delay and skew between devices. EFB mode is supported on only the center and corner PLLs in Stratix V devices.

In external feedback mode, the output of the `M` counter (`FBOUT`) feeds back to the PLL `fbin` input (using a trace on the board) and becomes part of the feedback loop. Also, 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 4–27 shows an example waveform of the phase relationship between the PLL clocks in external feedback mode.

Figure 4–27. Phase Relationship Between the PLL Clocks in External Feedback Mode

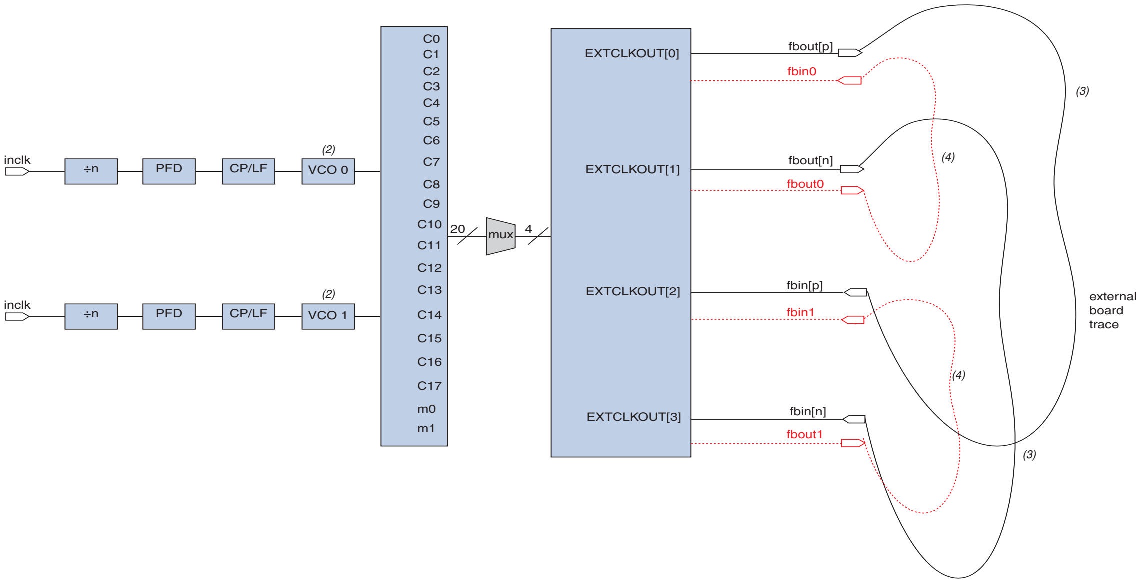

Phase Aligned
PLL Reference Clock at the Input Pin
PLL Clock at the Register Clock Port *(1)*
Dedicated PLL Clock Outputs *(1)*
fbin Clock Input Pin

Note to Figure 4–27:

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

Figure 4–28 shows external feedback mode implementation in Stratix V devices.

Figure 4–28. External Feedback Mode in Stratix V Devices ***(Note 1)***



Notes to Figure 4–28:

(1) EFB mode can support two single-ended or one differential feedback inputs. For more information, refer to Figure 4–20 on page 4–24.

(2) Only one of the two VCOs can support differential EFB mode at one time while you can use the other VCO for general purpose clocking.

(3) External board connection for one differential clock output and one differential feedback input for differential EFB support.

(4) External board connection for two single-ended clock outputs and two single-ended feedback inputs for single-ended EFB support.

Clock Multiplication and Division

Each Stratix V PLL provides clock synthesis for PLL output ports using $M/(N \times \textit{post-scale counter})$ 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. The 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 Altera® PLL megafunction.

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.

Document Revision History

Table 4–5 lists the revision history for this chapter.

Table 4–5. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 4–1. ■ Updated Figure 4–3, Figure 4–4, Figure 4–5, Figure 4–6, Figure 4–15, Figure 4–17, Figure 4–18, Figure 4–20, Figure 4–25, and Figure 4–28. ■ Updated "Zero-Delay Buffer Mode" and "External Feedback Mode" sections. ■ Added "PLL Clock Outputs" section.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

Section II. I/O Interfaces

This section provides information about Stratix® V device I/O features, external memory interfaces, and high-speed differential interfaces with dynamic phase alignment (DPA). This section includes the following chapters:

- Chapter 5, I/O Features in Stratix V Devices
- Chapter 6, High-Speed Differential I/O Interfaces and DPA in Stratix V Devices
- Chapter 7, External Memory Interfaces in Stratix V 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.

5. I/O Features in Stratix V Devices

SV51006-1.3

This chapter describes how Stratix® V 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.

Stratix V 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 enhance system-level performance.

The I/O capability of the Stratix V device family exceeds the I/O bandwidth available in 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/Os.

Package and die enhancements with dynamic termination and output control provide the best signal integrity in its class. Stratix V devices provide I/O features that assist high-speed data transfer into and out of the device, including the following features:

- Up to 1020 general purpose I/Os (GPIOs) and 255 full-duplex true LVDS channels
- True LVDS channels in all I/O banks support SGMII, SPI-4.2, and XSBI applications
- Hard dynamic phase alignment (DPA) and serializer/deserializer (SERDES) support in I/O banks on all sides of the device with DPA
- Single-ended, non-voltage-referenced, and voltage-referenced I/O standards
- **LVDS**, **RSDS**, **mini-LVDS**, **HSTL**, **SSTL**, and **HSUL** I/O standards across all I/O banks
- Double data rate (DDR), single data rate (SDR), and half data rate input and output options
- Ubiquitous I/O support for both row and column I/Os
- Deskew, read and write leveling, and clock-domain crossing functionality for high-performance memory interface
- Programmable output current strength
- Programmable slew rate
- Programmable input and output delays
- Programmable bus-hold circuits
- Programmable pull-up resistors
- Open-drain output
- Dynamic on-chip termination (OCT) for on-chip series (R_S) with and without calibration, and on-chip parallel (R_T) termination with calibration
- Differential (R_D) OCT without calibration

- Programmable pre-emphasis
- Programmable differential output voltage (V_{OD})

☞ The following information is applicable to all Stratix V variants, unless noted otherwise.

I/O Standard Support

Stratix V devices support a wide range of industry I/O standards. Table 5–1 lists the I/O standards for Stratix V devices, as well as the typical applications they support. These devices support V_{CCIO} voltage levels of 3.0, 2.5, 1.8, 1.5, 1.35, 1.25, and 1.2 V.

Table 5–1. Stratix V I/O Standards and Applications (Part 1 of 2)

I/O Standard	Typical Applications
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
SSTL-2 Class I and II	DDR SDRAM
SSTL-18 Class I and II	DDR2 SDRAM
SSTL-15 Class I and II	DDR3 SDRAM
SSTL-15	DDR3 SDRAM
SSTL-135	DDR3L SDRAM
SSTL-125	DDR3U SDRAM
SSTL-12	RLDRAM III
HSTL-18 Class I and II	QDR II/RLDRAM II
HSTL-15 Class I and II	QDR II/QDR II+/RLDRAM II
HSTL-12 Class I and II	General purpose
HSUL-12	LPDDR2 SDRAM
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
Differential HSTL-12 Class I and II	Clock interfaces
Differential SSTL-15	DDR3 SDRAM
Differential SSTL-135	DDR3L SDRAM
Differential SSTL-125	DDR3U SDRAM
Differential SSTL-12	RLDRAM III
Differential HSUL-12	LPDDR2 SDRAM
LVDS	High-speed communications
RSDS	Flat panel display

Table 5–1. Stratix V I/O Standards and Applications (Part 2 of 2)

I/O Standard	Typical Applications
mini-LVDS	Flat panel display
LVPECL	Video graphics and clock distribution

Notes to Table 5–1:

(1) The **3.3-V LVTTL/LVCMOS** I/O standard is supported using V_{CCIO} at 3.0 V.

(2) This I/O standard is only supported in Stratix V GX and GS devices.

I/O Standards and Voltage Levels

Table 5–2 lists the supported I/O standards and typical V_{CCIO}, V_{CCPD}, V_{REF}, and board V_{TT} values for input and output.

Table 5–2. Stratix V I/O Standards and Voltage Levels *(Note 1)* (Part 1 of 2)

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			
3.3-V LVTTL *(2)*	JESD8-B	3.0/2.5	3.0	3.0	—	—
3.3-V LVCMOS *(2)*	JESD8-B	3.0/2.5	3.0	3.0	—	—
2.5-V LVCMOS	JESD8-5	3.0/2.5	2.5	2.5	—	—
1.8-V LVCMOS	JESD8-7	1.8/1.5	1.8	2.5	—	—
1.5-V LVCMOS	JESD8-11	1.8/1.5	1.5	2.5	—	—
1.2-V LVCMOS	JESD8-12	1.2	1.2	2.5	—	—
SSTL-2 Class I	JESD8-9B	*(3)*	2.5	2.5	1.25	1.25
SSTL-2 Class II	JESD8-9B	*(3)*	2.5	2.5	1.25	1.25
SSTL-18 Class I	JESD8-15	*(3)*	1.8	2.5	0.90	0.90
SSTL-18 Class II	JESD8-15	*(3)*	1.8	2.5	0.90	0.90
SSTL-15 Class I	—	*(3)*	1.5	2.5	0.75	0.75
SSTL-15 Class II	—	*(3)*	1.5	2.5	0.75	0.75
SSTL-15	JESD79-3D	*(3)*	1.5	2.5	0.75	*(4)*
SSTL-135	—	*(3)*	1.35	2.5	0.675	*(4)*
SSTL-125	—	*(3)*	1.25	2.5	0.625	*(4)*
SSTL-12	—	*(3)*	1.2	2.5	0.6	*(4)*
HSTL-18 Class I	JESD8-6	*(3)*	1.8	2.5	0.90	0.90
HSTL-18 Class II	JESD8-6	*(3)*	1.8	2.5	0.90	0.90
HSTL-15 Class I	JESD8-6	*(3)*	1.5	2.5	0.75	0.75
HSTL-15 Class II	JESD8-6	*(3)*	1.5	2.5	0.75	0.75
HSTL-12 Class I	JESD8-16A	*(3)*	1.2	2.5	0.6	0.6
HSTL-12 Class II	JESD8-16A	*(3)*	1.2	2.5	0.6	0.6
HSUL-12	—	*(3)*	1.2	2.5	0.6	*(4)*
Differential SSTL-2 Class I	JESD8-9B	*(3)*	2.5	2.5	—	1.25
Differential SSTL-2 Class II	JESD8-9B	*(3)*	2.5	2.5	—	1.25
Differential SSTL-18 Class I	JESD8-15	*(3)*	1.8	2.5	—	0.90

Table 5–2. Stratix V I/O Standards and Voltage Levels *(Note 1)* **(Part 2 of 2)**

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			
Differential SSTL-18 Class II	JESD8-15	*(3)*	1.8	2.5	—	0.90
Differential SSTL-15 Class I	—	*(3)*	1.5	2.5	—	0.75
Differential SSTL-15 Class II	—	*(3)*	1.5	2.5	—	0.75
Differential HSTL-18 Class I	JESD8-6	*(3)*	1.8	2.5	—	0.90
Differential HSTL-18 Class II	JESD8-6	*(3)*	1.8	2.5	—	0.90
Differential HSTL-15 Class I	JESD8-6	*(3)*	1.5	2.5	—	0.75
Differential HSTL-15 Class II	JESD8-6	*(3)*	1.5	2.5	—	0.75
Differential HSTL-12 Class I	JESD8-16A	*(3)*	1.2	2.5	—	0.60
Differential HSTL-12 Class II	JESD8-16A	*(3)*	1.2	2.5	—	0.60
Differential SSTL-15	JESD79-3D	*(3)*	1.5	2.5	—	*(4)*
Differential SSTL-135	—	*(3)*	1.35	2.5	—	*(4)*
Differential SSTL-125	—	*(3)*	1.25	2.5	—	*(4)*
Differential SSTL-12	—	*(3)*	1.2	2.5	—	*(4)*
Differential HSUL-12	—	*(3)*	1.2	2.5	—	*(4)*
LVDS *(6)*, *(7)*	ANSI/TIA/EIA-644	*(3)*	2.5	2.5	—	—
RSDS *(6)*, *(7)*	—	*(3)*	2.5	2.5	—	—
mini-LVDS *(6)*, *(7)*	—	*(3)*	2.5	2.5	—	—
LVPECL	—	*(5)*	—	2.5	—	—

Notes to Table 5–2:

(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) For more information about the **3.3-V LVTTL/LVCMOS** I/O standard supported in Stratix V devices, refer to "3.3-V I/O Interface" on page 5–10.

(3) Single-ended **HSTL**/**SSTL**/**HSUL**, **differential SSTL**/**HSTL**/**HSUL**, and **LVDS** input buffers are powered by V_{CCPD}. **Differential HSTL**, **SSTL**, and **HSUL** outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted. **Differential HSTL**, **SSTL**, and **HSUL** inputs use **LVDS** differential input buffers with R_D support.

(4) Typically this I/O standard does not require board termination.

(5) The **LVPECL** I/O standard is supported for input clock operation. Differential clock input buffers are powered by V_{CCPD}.

(6) All I/O banks support true **LVDS**, **RSDS**, and **mini-LVDS** I/O standards using true **LVDS** output buffers without resistor networks. All I/O banks also support emulated **LVDS**, **RSDS**, and **mini-LVDS** I/O standards using two single-ended output buffers with a three-resistor (LVDS_E_3R, RSDS_E_3R, and mini-LVDS_E_3R) network.

(7) The emulated differential output standard that supports the tri-state feature includes: **LVDS_E_3R**, **RSDS_E_3R**, and **mini_LVDS_E_3R**.

I/O Banks

Figure 5–1 shows the I/O banks in Stratix V devices. All I/O banks in Stratix V devices contain true differential input and output buffers and dedicated circuitry to support differential I/O standards. Each I/O bank in Stratix V devices supports a high-performance external memory interface. The I/O pins are organized in pairs to support differential I/O standards. Each I/O pin pair can support both differential input and output buffers.

Figure 5–1. I/O Banks for Stratix V Devices—Preliminary

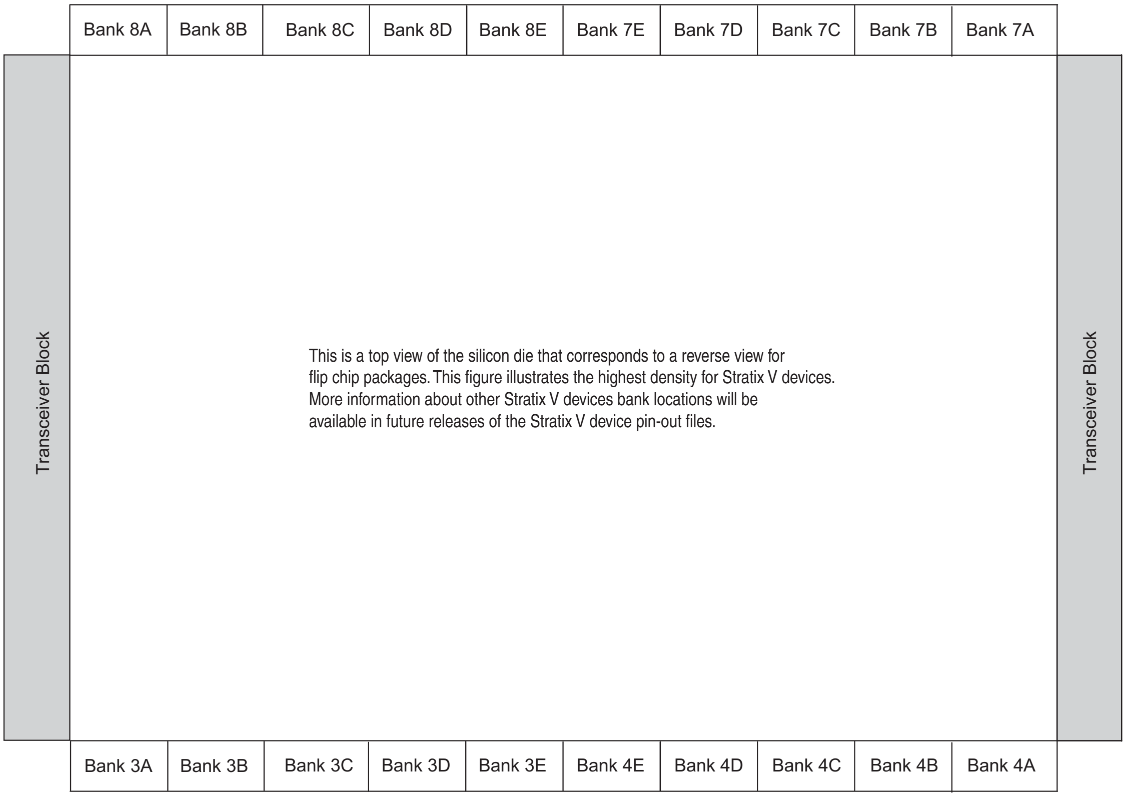

Bank 8A
Bank 8B
Bank 8C
Bank 8D
Bank 8E
Bank 7E
Bank 7D
Bank 7C
Bank 7B
Bank 7A
Transceiver Block
This is a top view of the silicon die that corresponds to a reverse view for flip chip packages. This figure illustrates the highest density for Stratix V devices. More information about other Stratix V devices bank locations will be available in future releases of the Stratix V device pin-out files.
Transceiver Block
Bank 3A
Bank 3B
Bank 3C
Bank 3D
Bank 3E
Bank 4E
Bank 4D
Bank 4C
Bank 4B
Bank 4A

Table 5–3 through Table 5–5 list the GPIO, LVDS, and transceiver channel counts in all Stratix V device packages.

Table 5–3. GPIO, LVDS, and Transceiver Channel Counts for Stratix V GX Devices *(Note 1)*—Preliminary

Device	HF780	HF1152	KF1152	KF1517	NF1517	RF1517	RF1760	NF1932
5SGXA3	264, 66, 24	552, 138, 24	444, 111, 36	624, 156, 36	—	—	—	—
5SGXA4	264, 66, 24	552, 138, 24	444, 111, 36	624, 156, 36	—	—	—	—
5SGXA5	—	552, 138, 24	444, 111, 36	696, 174, 36	600, 150, 48	—	—	840, 210, 48
5SGXA7	—	552, 138, 24	444, 111, 36	696, 174, 36	600, 150, 48	—	—	840, 210, 48
5SGXA9	—	—	—	696, 174, 36	—	—	—	840, 210, 48
5SGXAB	—	—	—	696, 174, 36	—	—	—	840, 210, 48
5SGXB5	—	—	—	—	—	432, 108, 66	600, 150, 66	—
5SGXB6	—	—	—	—	—	432, 108, 66	600, 150, 66	—

Note to Table 5–3:

(1) LVDS and transceiver counts are full duplex channels. Each full duplex channel has one transmitter (TX) pair and one receiver (RX) pair with hard SERDES.

Table 5–4. GPIO, LVDS, and Transceiver Channel Counts for Stratix V GS Devices *(Note 1)*—Preliminary

Device	DF484	EF780	GF1152	HF1152	KF1517	NF1932
5SGSD2	240, 60, 9	396, 100, 12	—	—	—	—
5SGSD3	240, 60, 9	396, 100, 12	492, 125, 18	—	—	—
5SGSD4	—	396, 100, 12	—	552, 140, 24	696, 175, 36	—
5SGSD5	—	—	—	552, 140, 24	696, 175, 36	—
5SGSD6	—	—	—	—	696, 175, 36	900, 225, 48
5SGSD8	—	—	—	—	696, 175, 36	900, 225, 48

Note to Table 5–4:

(1) LVDS and transceiver counts are full duplex channels. Each full duplex channel has one transmitter (TX) pair and one receiver (RX) pair with hard SERDES.

Table 5–5. GPIO, LVDS, and Transceiver Channel Counts for Stratix V GT and Stratix V E Devices *(Note 1)*—Preliminary

Device	F1152	F1517	KF1517	F1932
5SGTC5	—	—	588, 149, 36	—
5SGTC7	—	—	588, 149, 36	—
5SEE9	552, 138, 0	696, 174, 0	—	840, 210, 0
5SEEB	552, 138, 0	696, 174, 0	—	840, 210, 0

Note to Table 5–5:

(1) LVDS and transceiver counts are full duplex channels. Each full duplex channel has one transmitter (TX) pair and one receiver (RX) pair with hard SERDES.

Modular I/O Banks

The I/O pins in Stratix V devices are arranged in groups called modular I/O banks. The number of Stratix V I/O banks in a particular device ranges from 16 to 26, depending on the device density.

Table 5–6 through Table 5–8 list the modular I/O banks for Stratix V devices.

Table 5–6. Modular I/O Banks for Stratix V GX Devices—Preliminary

Package	Device	Bank																				Total
		3A	3B	3C	3D	3E	4A	4B	4C	4D	4E	7A	7B	7C	7D	7E	8A	8B	8C	8D	8E	
HF780	5SGXA3	36	48	—	24	—	24	—	—	24	—	24	—	—	36	—	24	—	—	24	—	264
	5SGXA4	36	48	—	24	—	24	—	—	24	—	24	—	—	36	—	24	—	—	24	—	264
HF1152	5SGXA3	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
	5SGXA4	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
	5SGXA5	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
	5SGXA7	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
KF1152	5SGXA3	36	48	—	24	—	36	48	—	36	—	24	48	48	36	—	24	—	—	36	—	444
	5SGXA4	36	48	—	24	—	36	48	—	36	—	24	48	48	36	—	24	—	—	36	—	444
	5SGXA5	36	48	—	24	—	36	48	—	36	—	24	48	48	36	—	24	—	—	36	—	444
	5SGXA7	36	48	—	24	—	36	48	—	36	—	24	48	48	36	—	24	—	—	36	—	444
KF1517	5SGXA3	36	48	48	24	—	24	48	48	36	—	24	48	48	36	—	24	48	48	36	—	624
	5SGXA4	36	48	48	24	—	24	48	48	36	—	24	48	48	36	—	24	48	48	36	—	624
	5SGXA5	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGXA7	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGXA9	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGXAB	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
HF1517	5SGXA5	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	36	—	48	48	—	600
	5SGXA7	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	36	—	48	48	—	600
RF1517	5SGXB5	36	48	—	—	—	48	48	36	—	—	48	48	36	—	—	36	48	—	—	—	432
	5SGXB6	36	48	—	—	—	48	48	36	—	—	48	48	36	—	—	36	48	—	—	—	432
RF1760	5SGXB5	36	48	48	36	—	48	48	36	—	—	48	48	36	—	—	36	48	48	36	—	600
	5SGXB6	36	48	48	36	—	48	48	36	—	—	48	48	36	—	—	36	48	48	36	—	600
HF1932	5SGXA5	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SGXA7	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SGXA9	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SGXAB	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840

Table 5–7. Modular I/O Banks for Stratix V GS Devices—Preliminary

Package	Device	Bank																				Total
		3A	3B	3C	3D	3E	4A	4B	4C	4D	4E	7A	7B	7C	7D	7E	8A	8B	8C	8D	8E	
DF484	5SGSD2	36	24	—	24	—	24	—	—	24	—	24	—	—	36	—	24	—	—	24	—	240
	5SGSD3	36	24	—	24	—	24	—	—	24	—	24	—	—	36	—	24	—	—	24	—	240
EF780	5SGSD2	36	36	—	24	—	36	36	—	36	—	24	36	36	36	—	24	—	—	36	—	396
	5SGSD3	36	36	—	24	—	36	36	—	36	—	24	36	36	36	—	24	—	—	36	—	396
	5SGSD4	36	36	—	24	—	36	36	—	36	—	24	36	36	36	—	24	—	—	36	—	396
GF1152	5SGSD3	36	48	—	24	—	36	48	—	36	—	24	48	48	36	—	24	48	—	24	—	492
HF1152	5SGSD4	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
	5SGSD5	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	48	—	552
KF1517	5SGSD4	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGSD5	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGSD6	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SGSD8	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
NF1932	5SGSD6	36	48	48	48	48	36	48	48	48	48	36	48	48	48	48	36	48	48	48	36	900
	5SGSD8	36	48	48	48	48	36	48	48	48	48	36	48	48	48	48	36	48	48	48	36	900

Table 5–8. Modular I/O Banks for Stratix V GT and E Devices—Preliminary

Package	Device	Bank																				Total
		3A	3B	3C	3D	3E	4A	4B	4C	4D	4E	7A	7B	7C	7D	7E	8A	8B	8C	8D	8E	
KF1517	5SGTC5	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	24	—	48	48	—	588
	5SGTC7	36	48	48	24	—	24	48	48	24	—	24	48	48	48	—	24	—	48	48	—	588
F1152	5SEE9	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
	5SEEB	36	48	48	24	—	24	48	48	24	—	24	48	48	36	—	24	—	48	24	—	552
F1517	5SEE9	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
	5SEEB	36	48	48	48	—	24	48	48	48	—	24	48	48	48	—	36	48	48	48	—	696
F1932	5SEE9	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840
	5SEEB	36	48	48	48	36	24	48	48	48	36	24	48	48	48	36	36	48	48	48	36	840

I/O Structure

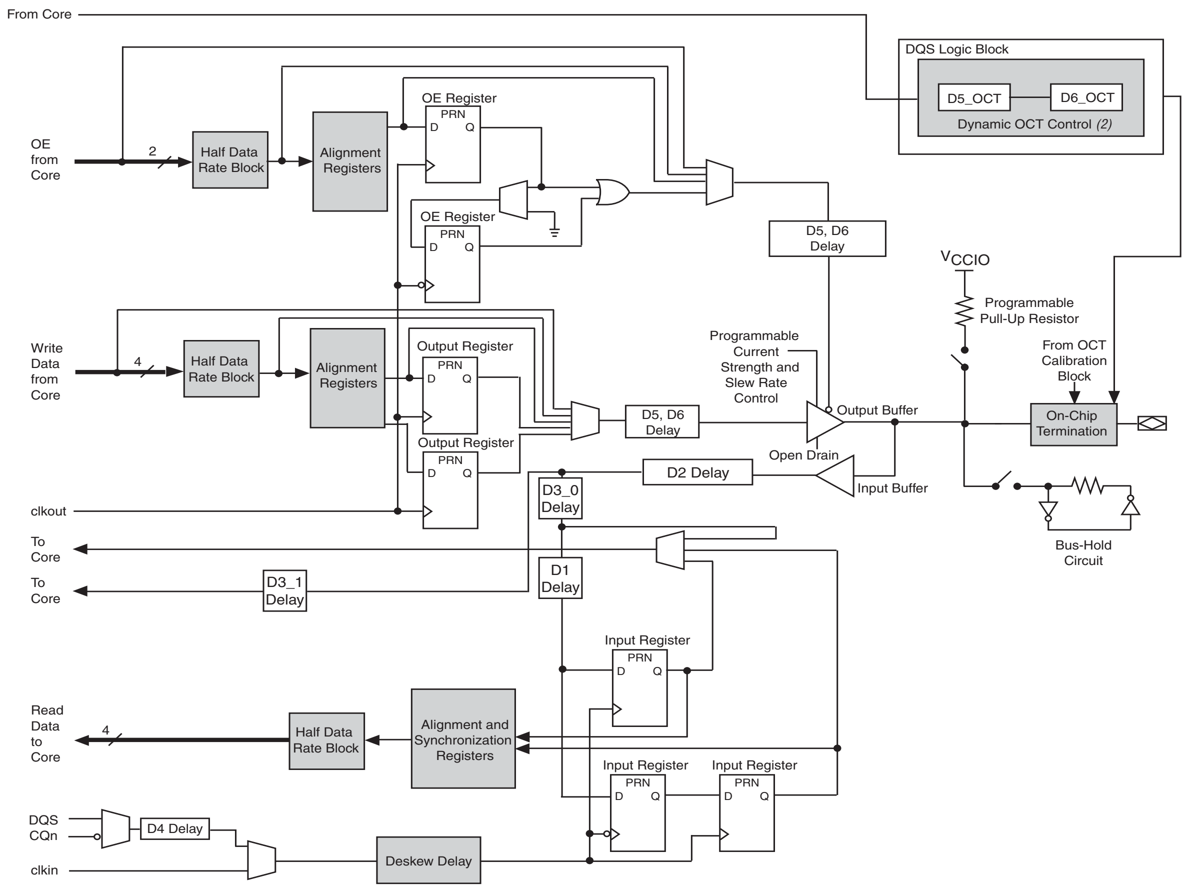
The I/O elements (IOEs) in Stratix V devices contain a bidirectional I/O buffer and I/O registers to support a complete embedded bidirectional SDR or DDR transfer. The IOEs are located in I/O blocks around the periphery of the Stratix V device.

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 half data rate blocks; you can bypass each block in the input path. The input path uses the deskew delay to adjust the input register clock delay across process, voltage, and temperature (PVT) variations.

The output and `OE` paths are divided into the output or `OE` registers, alignment registers, and half data rate blocks. You can bypass each block of the output and `OE` paths.

Figure 5–2 shows the Stratix V IOE structure.

Figure 5–2. IOE Structure for Stratix V Devices *(Note 1)*, *(2)*



Notes to Figure 5–2:

(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.

3.3-V I/O Interface

Stratix V 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 V 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 V devices, do not violate the absolute maximum ratings of the devices. Altera recommends performing IBIS or SPICE simulations to determine that the overshoot and undershoot voltages are within the specifications.

When using a Stratix V device as a transmitter, you can use slow slew rate and series termination to limit overshoot and undershoot at the I/O pins. 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 V devices support R_S OCT for all **LVTTL** and **LVCMOS** I/O standards in all I/O banks.

When using the Stratix V device as a receiver, you can use a clamping diode (off-chip) if it is required to limit the overshoot voltage.

The 3.3-V I/O standard is supported using the bank supply voltage (V_{CCIO}) at 3.0 V and a V_{CCPD} voltage of 3.0 V. In this method, the clamping diode (off-chip) can sufficiently clamp overshoot voltage to within the DC and AC input voltage specifications. The clamped voltage is expressed as the sum of the V_{CCIO} and the diode forward voltage.

External Memory Interfaces

In addition to the I/O registers in each IOE, Stratix V devices also have dedicated registers and phase-shift circuitry on all I/O banks to interface with external memory. Stratix V devices support new I/O standards such as **SSTL-12**, **SSTL-15**, **SSTL-125**, **SSTL-135**, and **HSUL-12**.

High-Speed Differential I/O with DPA Support

Stratix V devices have 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)

Current Strength

The output buffer for each Stratix V device I/O pin has programmable current strength control for certain I/O standards. 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** I/O standards have several levels of current strength that you can control. Table 5–9 lists the programmable current strength settings for Stratix V devices.

Table 5–9. Programmable Current Strength Settings

I/O Standard	I_{OH} / I_{OL} Current Strength Setting	Units
3.3-V LVTTL *(1)*	16, 12, 8, 4	mA
3.3-V LVCMOS *(1)*	16, 12, 8, 4	mA
2.5-V LVCMOS	16, 12, 8, 4	mA
1.8-V LVCMOS	12, 10, 8, 6, 4, 2	mA
1.5-V LVCMOS	12, 10, 8, 6, 4, 2	mA
1.2-V LVCMOS	8, 6, 4, 2	mA
SSTL-2 Class I	12, 10, 8	mA
SSTL-2 Class II	16	mA
SSTL-18 Class I	12, 10, 8, 6, 4	mA
SSTL-18 Class II	16	mA
SSTL-15 Class I	12, 10, 8, 6, 4	mA
SSTL-15 Class II	16	mA
HSTL-18 Class I	12, 10, 8, 6, 4	mA
HSTL-18 Class II	16	mA
HSTL-15 Class I	12, 10, 8, 6, 4	mA
HSTL-15 Class II	16	mA
HSTL-12 Class I	12, 10, 8, 6, 4	mA
HSTL-12 Class II	**16**	mA
SSTL-12	40, 60, 240 *(2)*, *(3)*	Ω
SSTL-15	34, 40, 25, 50 *(2)*, *(3)*	Ω
SSTL-125	34, 40 *(2)*, *(3)*	Ω
SSTL-135	34, 40 *(2)*, *(3)*	Ω
HSUL-12	34, 40, 48, 60, 80 *(2)*, *(3)*	Ω

Notes to Table 5–9:

(1) The **3.3-V LVTTL** and **3.3-V LVCMOS** I/O standards are supported using V_{CCIO} and V_{CCPD} at 3.0 V.

(2) The current strength is represented by the driver impedance value (Ω). Only R_S OCT with calibration is supported.

(3) Pending silicon characterization.

☞ 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 V 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 fast slew rate provides high-speed transitions for high-performance systems. A slow 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 R_S OCT.

The Quartus® II software allows two settings for programmable slew rate control—0 and 1—where 0 is slow slew rate and 1 (default) is fast slew rate. Fast slew rates 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 V device IOE includes programmable delays, shown in Figure 5–2 on page 5–9 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 write timing 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 for Stratix V Devices* chapter.

Programmable Output Buffer Delay

Stratix V devices support delay chains built inside the single-ended output buffer, as shown in Figure 5–2 on page 5–9. 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 V devices support four levels of output buffer delay settings with the default setting of no delay.

For more information about programmable output buffer delay specifications, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Open-Drain Output

Stratix V 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 V 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 require 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 the 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.

The bus-hold circuit is active only after configuration. When going into user mode, the bus-hold circuit captures the value on the pin that is present at the end of configuration.

Pull-Up Resistor

Each Stratix V 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 V LVDS transmitters support programmable pre-emphasis to compensate for the frequency dependent attenuation of the transmission line. The Quartus II software allows two settings for programmable pre-emphasis control—0 and 1—where 0 is disabled and 1 (default) is enabled.

Differential Output Voltage

Stratix V 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} control—0, 1, 2, and 3—where 0 is low, 1 (default) is medium low, 2 is medium high, and 3 is high.

MultiVolt I/O Interface

The Stratix V architecture supports the MultiVolt I/O interface feature that allows Stratix V devices in all packages to interface with systems of different supply voltages.

You can connect the VCCIO pins to a 1.2-, 1.25-, 1.35-, 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.)

The Stratix V 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 5–10 lists Stratix V MultiVolt I/O support.

Table 5–10. Stratix V MultiVolt I/O Support *(Note 1)*

V_{CCIO} (V) *(3)*	Input Signal (V)								Output Signal (V)							
	1.2	1.25	1.35	1.5	1.8	2.5	3.0	3.3	1.2	1.25	1.35	1.5	1.8	2.5	3.0	3.3
1.2	✓	—	—	—	—	—	—	—	✓	—	—	—	—	—	—	—
1.25	—	✓	—	—	—	—	—	—	—	✓	—	—	—	—	—	—
1.35	—	—	✓	—	—	—	—	—	—	—	✓	—	—	—	—	—
1.5	—	—	—	✓	✓	—	—	—	—	—	—	✓	—	—	—	—
1.8	—	—	—	✓	✓	—	—	—	—	—	—	—	✓	—	—	—
2.5	—	—	—	—	—	✓	✓ *(2)*	✓ *(2)*	—	—	—	—	—	✓	—	—
3.0	—	—	—	—	—	✓	✓ *(2)*	✓ *(2)*	—	—	—	—	—	—	✓	✓

Notes to Table 5–10:

(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 V 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.

(3) Each I/O bank of a Stratix V device has its own VCCIO pins and supports only one V_{CCIO}, either 1.2, 1.25, 1.35, 1.5, 1.8, or 3.0 V. The **LVDS** I/O standard is not supported when V_{CCIO} is 3.0 V. The LVDS input operations are supported when V_{CCIO} is 1.2, 1.25, 1.35, 1.5, 1.8, or 2.5 V. The **LVDS** output operations are only supported when V_{CCIO} is 2.5 V.

OCT Support and I/O Termination Schemes

Stratix V devices feature dynamic R_S and R_T OCT to provide I/O impedance matching and termination capabilities. OCT maintains signal quality, saves board space, and reduces external component costs.

Stratix V devices support the following OCT schemes:

- R_S with and without calibration
- R_T with calibration
- Dynamic R_S for single-ended I/O standards
- Dynamic R_T for single-ended I/O standards
- R_D for differential LVDS I/O standards

Stratix V devices support OCT in all I/O banks. You can use R_S and R_T OCT 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 R_S OCT, programmable current strength, or R_T OCT.

☞ You cannot configure both the R_S OCT and the programmable current strength for the same I/O buffer.

The Stratix V OCT calibration process uses the `RZQ` pin that is available in every calibration block in a given I/O bank for series- and parallel-calibrated termination. The `RZQ` pin shares the same V_{CCIO} supply with the I/O bank where it is located. It is a dual-purpose I/O pin and functions as a GPIO if you do not use the calibration circuit. When used for OCT calibration, the `RZQ` pin is connected to GND through an external 100- or 240-Ω reference resistor.

All I/O pins support calibrated R_S OCT, calibrated R_T OCT, and dynamic OCT for bidirectional pins. Dynamic R_T OCT is enabled for a bidirectional pin in receive mode and disabled in transmit mode.

The following connections are required to connect the `RZQ` pin through a reference resistor:

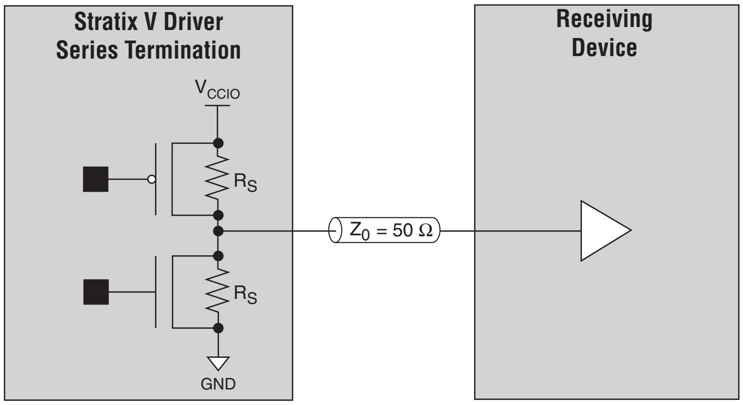
- `RZQ` pin is connected to GND through an external 240-Ω resistor for R_S OCT of 34, 40, 48, 60, and 80 Ω
- `RZQ` pin is connected to GND through an external 240-Ω resistor for R_T OCT of 20, 30, 40, 60, and 120 Ω
- `RZQ` pin is connected to GND through an external 100-Ω resistor for R_S OCT of 25 Ω and 50 Ω
- `RZQ` pin is connected to GND through an external 100-Ω resistor for R_T OCT of 50 Ω

R_S OCT Without Calibration

Stratix V 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 V devices support R_S OCT for single-ended I/O standards (refer to Figure 5–3).

Figure 5–3 shows the R_S as the intrinsic impedance of the output transistors. Typical R_S values are 25 Ω and 50 Ω. When you select matching impedance, current strength is no longer selectable.

Figure 5–3. R_S OCT Without Calibration



To use OCT for the **SSTL Class I** I/O standard, you must select the 50-Ω R_S OCT setting, thus eliminating the external 25-Ω R_S (to match the 50-Ω transmission line). For the **SSTL Class II** I/O standard, you must select the 25-Ω R_S OCT setting (to match the 50-Ω transmission line and the near-end external 50-Ω pull-up to V_{TT}).

R_S OCT with Calibration

Stratix V devices support R_S OCT with calibration in all banks. The R_S OCT calibration circuit compares the total impedance of the I/O buffer to the external 240-Ω or 100-Ω reference resistor connected to the RZQ pin and dynamically enables or disables the transistors until they match.

The R_S shown in Figure 5–4 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 5–4. R_S OCT with Calibration

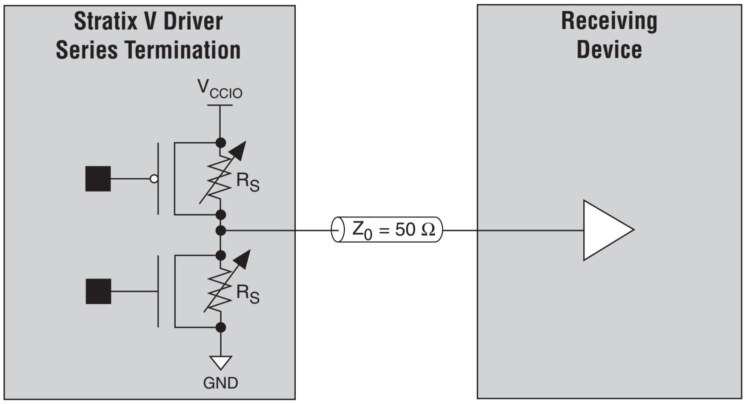


Table 5–11 lists the input and output termination for OCT with and without calibration on different I/O standards.

Table 5–11. Selectable I/O Standards for OCT With and Without Calibration (Part 1 of 2)

I/O standards	Output Termination		Input Termination	V_{REF} (V)	V_{CCIO} (V)
	Uncalibrated R_S OCT Setting, R_S (Ω)	Calibrated R_S OCT Setting, R_S (Ω) *(1)*	R_T OCT Setting, R_T (Ω)		
3.3V LVTTL/LVCMOS	25/50	25/50	—	—	3
2.5-V LVCMOS	25/50	25/50	—	—	2.5
1.8-V LVCMOS	25/50	25/50	—	—	1.8
1.2-V LVCMOS	25/50	25/50	—	—	1.2
SSTL-2 Class I	50	50	50	1.25	2.5
SSTL-2 Class II	25	25	50	1.25	2.5
SSTL-18 Class I	50	50	50	0.9	1.8
SSTL-18 Class II	25	25	50	0.9	1.8
SSTL15 Class I	50	50	50	0.75	1.5
SSTL15 Class II	25	25	50	0.75	1.5
HSTL 1.8 Class I	50	50	50	0.9	1.8
HSTL 1.8 Class II	25	25	50	0.9	1.8
HSTL 1.5 Class I	50	50	50	0.75	1.5
HSTL 1.5 Class II	25	25	50	0.75	1.5
HSTL 1.2 Class I	50	50	50	0.6	1.2
HSTL 1.2 Class II	25	25	50	0.6	1.2
SSTL15	25, 34, 40, 50	25, 34, 40, 50 *(2)*	20, 30, 40, 60, 120 *(2)*	0.75	1.5
SSTL135	34, 40	34, 40 *(2)*	20, 30, 40, 60, 120 *(2)*	0.675	1.35
SSTL125	34, 40	34, 40 *(2)*	20, 30, 40, 60, 120 *(2)*	0.625	1.25
SSTL12	40, 60, 240	40, 60, 240 *(2)*	60, 120 *(2)*	0.6	1.2
HSUL 1.2	34.3, 40, 48, 60, 80	34, 40, 48, 60, 80 *(2)*	*(2)*	0.6	1.2
Differential SSTL-2 Class I	50	50	50	1.25	2.5
Differential SSTL-2 Class II	25	25	50	1.25	2.5
Differential SSTL-18 Class I	50	50	50	0.9	1.8
Differential SSTL-18 Class II	25	25	50	0.9	1.8
Differential SSTL15 Class I	50	50	50	0.75	1.5
Differential SSTL15 Class II	25	25	50	0.75	1.5
Differential HSTL 1.8 Class I	50	50	50	0.9	1.8
Differential HSTL 1.8 Class II	25	25	50	0.9	1.8
Differential HSTL 1.5 Class I	50	50	50	0.75	1.5
Differential HSTL 1.5 Class II	25	25	50	0.75	1.5
Differential HSTL 1.2 Class I	50	50	50	0.6	1.2

Table 5–11. Selectable I/O Standards for OCT With and Without Calibration (Part 2 of 2)

I/O standards	Output Termination		Input Termination	V_{REF} (V)	V_{CCIO} (V)
	Uncalibrated R_S OCT Setting, R_S (Ω)	Calibrated R_S OCT Setting, R_S (Ω) *(1)*	R_T OCT Setting, R_T (Ω)		
Differential HSTL 1.2 Class II	25	25	50	0.6	1.2
Differential SSTL15	—	25, 34, 40, 50 *(2)*	20, 30, 40, 60, 120 *(2)*	—	1.5
Differential SSTL135	—	34, 40 *(2)*	20, 30, 40, 60, 120 *(2)*	—	1.35
Differential SSTL125	—	34, 40 *(2)*	20, 30, 40, 60, 120 *(2)*	—	1.25
Differential SSTL12	—	40, 60, 240 *(2)*	60, 120 *(2)*	—	1.2
Differential HSUL 1.2	—	34, 40, 48, 60, 80 *(2)*	*(2)*	—	1.2

Notes to Table 5–11:

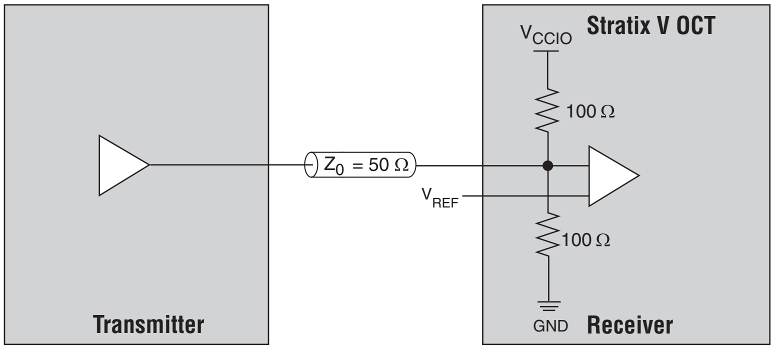
(1) The 25- and 50-Ω driver impedance are calibrated with the RZQ pin connected to a 100-Ω reference resistor to GND.

(2) The calibrated R_S and R_T OCT final values are pending silicon characterization.

R_T OCT with Calibration

Stratix V devices support R_T OCT with calibration in all banks. R_T OCT with calibration is only supported for input configuration of input and bidirectional pins. Output pin configurations do not support R_T OCT with calibration. Figure 5–5 shows R_T OCT with calibration. When you use R_T OCT, the V_{CCIO} of the bank must match the I/O standard of the pin where the R_T OCT is enabled.

Figure 5–5. R_T OCT with Calibration



The R_T OCT calibration circuit compares the total impedance of the I/O buffer to the external 100-Ω or 240-Ω resistors connected to the RZQ pin 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.

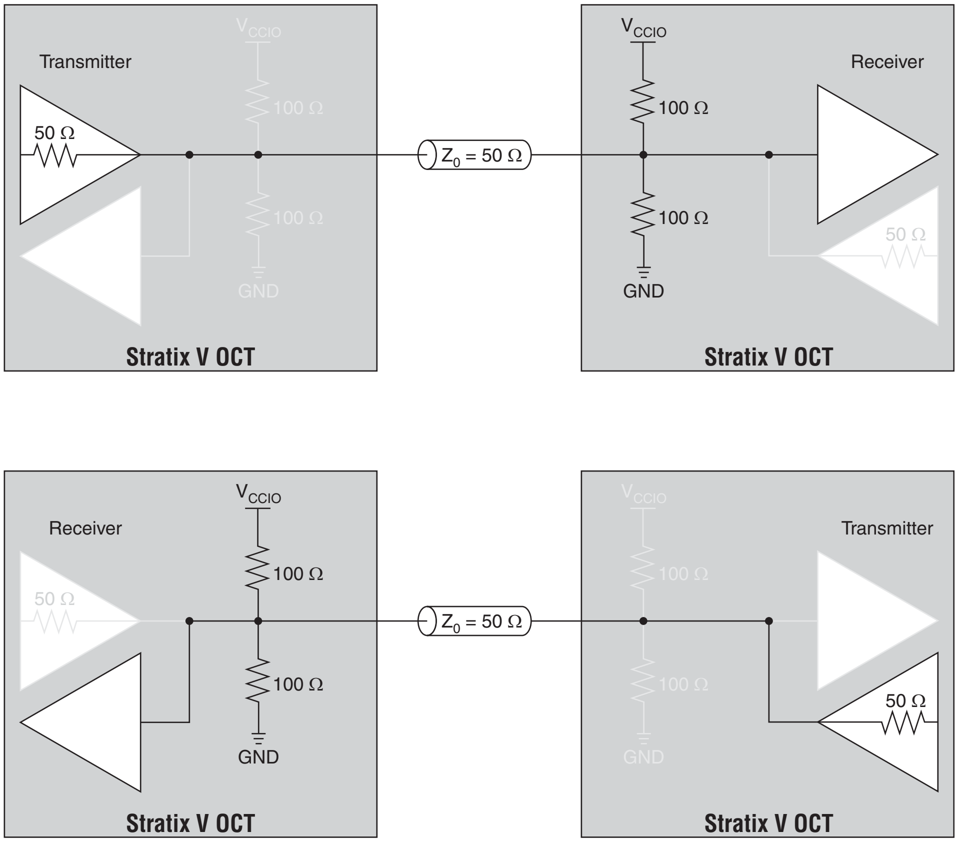
Dynamic OCT

Stratix V devices support dynamic R_S and R_T OCT for bidirectional I/Os in all I/O banks. Figure 5–6 shows the termination schemes supported in Stratix V devices. Dynamic R_T OCT is enabled only when the bidirectional I/O acts as a receiver and is disabled when it acts as a driver. Similarly, dynamic R_S OCT 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.

Altera recommends using the new I/O standards for the DDR3 memory interface with dynamic OCT schemes. These I/O standards save board space by reducing the number of external termination resistors used.

Using dynamic OCT also helps save power because device termination is internal instead of external. Termination only switches on during input operation, thus drawing less static power.

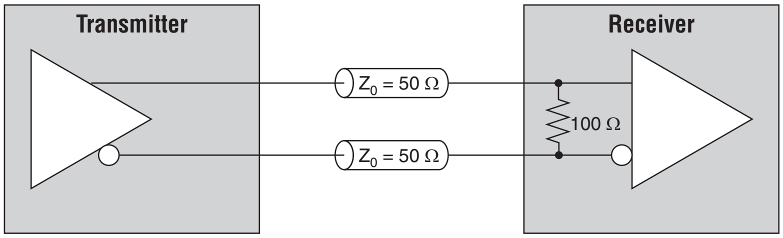
Figure 5–6. Dynamic R_T OCT in Stratix V Devices



LVDS Input R_D OCT

Stratix V devices support OCT for differential LVDS input buffers with a nominal resistance value of 100 Ω, as shown in Figure 5–7. R_D OCT is supported in all I/O banks. You can use R_D OCT when both the V_{CCIO} and V_{CCPD} is set to 2.5 V.

Figure 5–7. Differential Input OCT

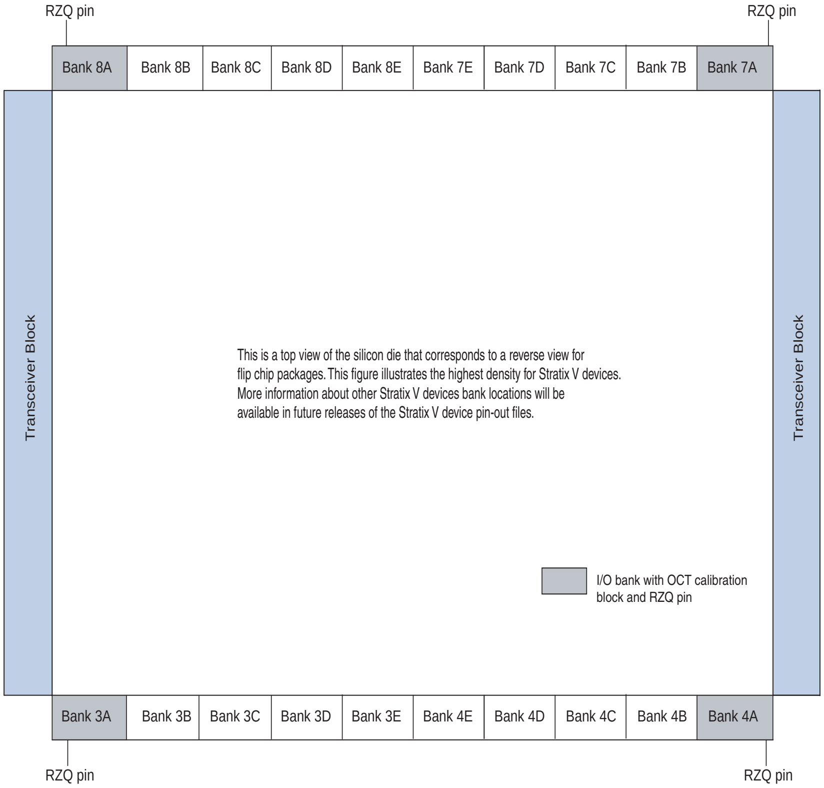


OCT Calibration

Stratix V devices support calibrated R_S and calibrated R_T on all I/O pins with exceptions for dedicated configuration pins. You can calibrate using any of the available four to eight OCT calibration blocks, depending on the density of the device. Each calibration block contains one RZQ pin.

Figure 5–8 shows the location of I/O banks with OCT calibration blocks and RZQ pins.

Figure 5–8. OCT Calibration Block and RZQ Pin Location—Preliminary


RZQ pin
RZQ pin
Bank 8A
Bank 8B
Bank 8C
Bank 8D
Bank 8E
Bank 7E
Bank 7D
Bank 7C
Bank 7B
Bank 7A
Transceiver Block
Transceiver Block
This is a top view of the silicon die that corresponds to a reverse view for flip chip packages. This figure illustrates the highest density for Stratix V devices. More information about other Stratix V devices bank locations will be available in future releases of the Stratix V device pin-out files.
I/O bank with OCT calibration block and RZQ pin
Bank 3A
Bank 3B
Bank 3C
Bank 3D
Bank 3E
Bank 4E
Bank 4D
Bank 4C
Bank 4B
Bank 4A
RZQ pin
RZQ pin

Sharing an OCT Calibration Block on Multiple I/O Banks

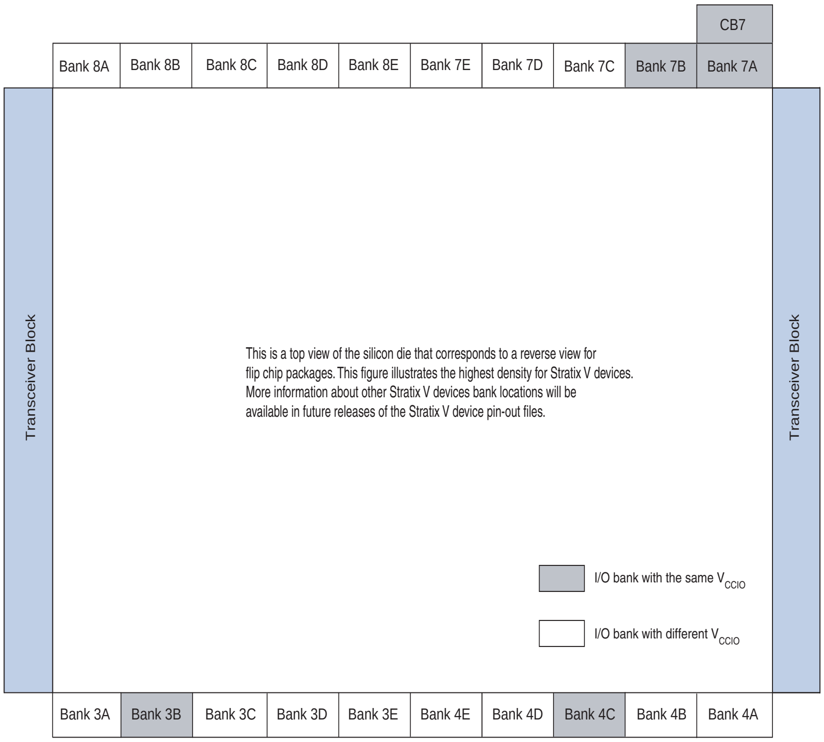
An OCT calibration block has the same V_{CCIO} as the I/O bank that contains the block. OCT calibration is supported on all I/O banks with different V_{CCIO} voltage standards, up to the number of available OCT calibration blocks. You can configure the I/O banks to receive calibration codes from any 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 5–9 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 (CB7) located in bank 7A. You can enable this by serially shifting out the R_S OCT calibration codes from the OCT calibration block located in bank 7A to the I/O banks located around the periphery.

☞ I/O banks that do not contain calibration blocks share calibration blocks with I/O banks that contains calibration blocks.

Figure 5–9 is a top view of the silicon die that corresponds to a reverse view for flip chip packages. This figure does not show transceiver calibration blocks.

Figure 5–9. Example of Calibrating Multiple I/O Banks with One Shared OCT Calibration Block—Preliminary


CB7
Bank 8A
Bank 8B
Bank 8C
Bank 8D
Bank 8E
Bank 7E
Bank 7D
Bank 7C
Bank 7B
Bank 7A
Transceiver Block
Transceiver Block
This is a top view of the silicon die that corresponds to a reverse view for flip chip packages. This figure illustrates the highest density for Stratix V devices. More information about other Stratix V devices bank locations will be available in future releases of the Stratix V device pin-out files.
I/O bank with the same V_CCIO
I/O bank with different V_CCIO
Bank 3A
Bank 3B
Bank 3C
Bank 3D
Bank 3E
Bank 4E
Bank 4D
Bank 4C
Bank 4B
Bank 4A

OCT Calibration Block Modes of Operation

OCT 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 signals are used to calibrate and serially transfer calibration codes from each OCT calibration block to any I/O.

Table 5–12 lists the user-controlled calibration block signal names and their descriptions.

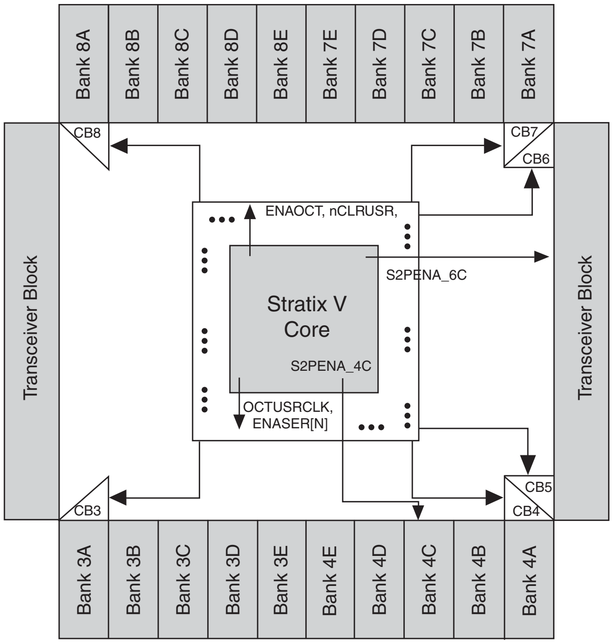
Table 5–12. OCT Calibration Block Ports for User Control

Signal Name	Description
OCTUSRCLK	Clock for OCT block.
ENAOCT	Enable OCT Calibration (Generated by user IP).
ENASER[7..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 5–10 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 the OCTUSRCLK signal.

Figure 5–10. Signals Used for User Mode Calibration



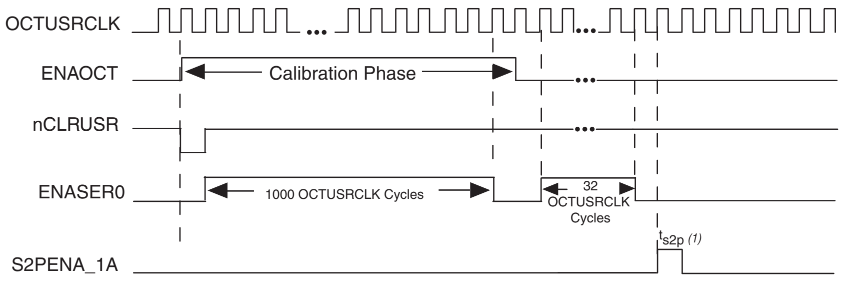
OCT Calibration

Figure 5–11 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 R_S OCT and R_T OCT calibration. You can deassert ENAOCT one clock cycle after the last ENASER is deasserted.

Serial Data Transfer

After you complete calibration, you must serially shift out the 32-bit OCT calibration codes (16-bit R_S OCT and 16-bit R_T OCT) 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 deassert ENAOCT, wait at least one OCTUSRCLK cycle to enable any ENASER[N] signal to begin serial transfer. To shift the 32-bit code from the OCT calibration block[N], you must assert ENASER[N] for exactly 32 OCTUSRCLK cycles. Between two consecutive asserted ENASER signals, there must be at least one OCTUSRCLK cycle gap (refer to Figure 5–11).

Figure 5–11. OCT User Mode Signal—Timing Waveform for One OCT Block



Note to Figure 5–11:

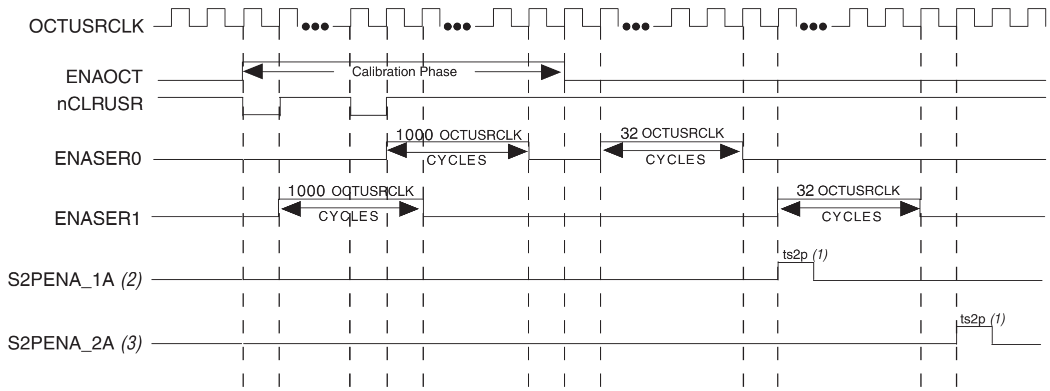
(1) $t_{s2p} \geq 25$ ns.

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 5–11 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 deasserted 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 5–12 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 5–12, 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 5–12. OCT User-Mode Signal Timing Waveform for Two OCT Blocks



Notes to Figure 5–12:

(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.

Termination Schemes for I/O Standards

The following sections describe the different termination schemes for the I/O standards used in Stratix V devices.

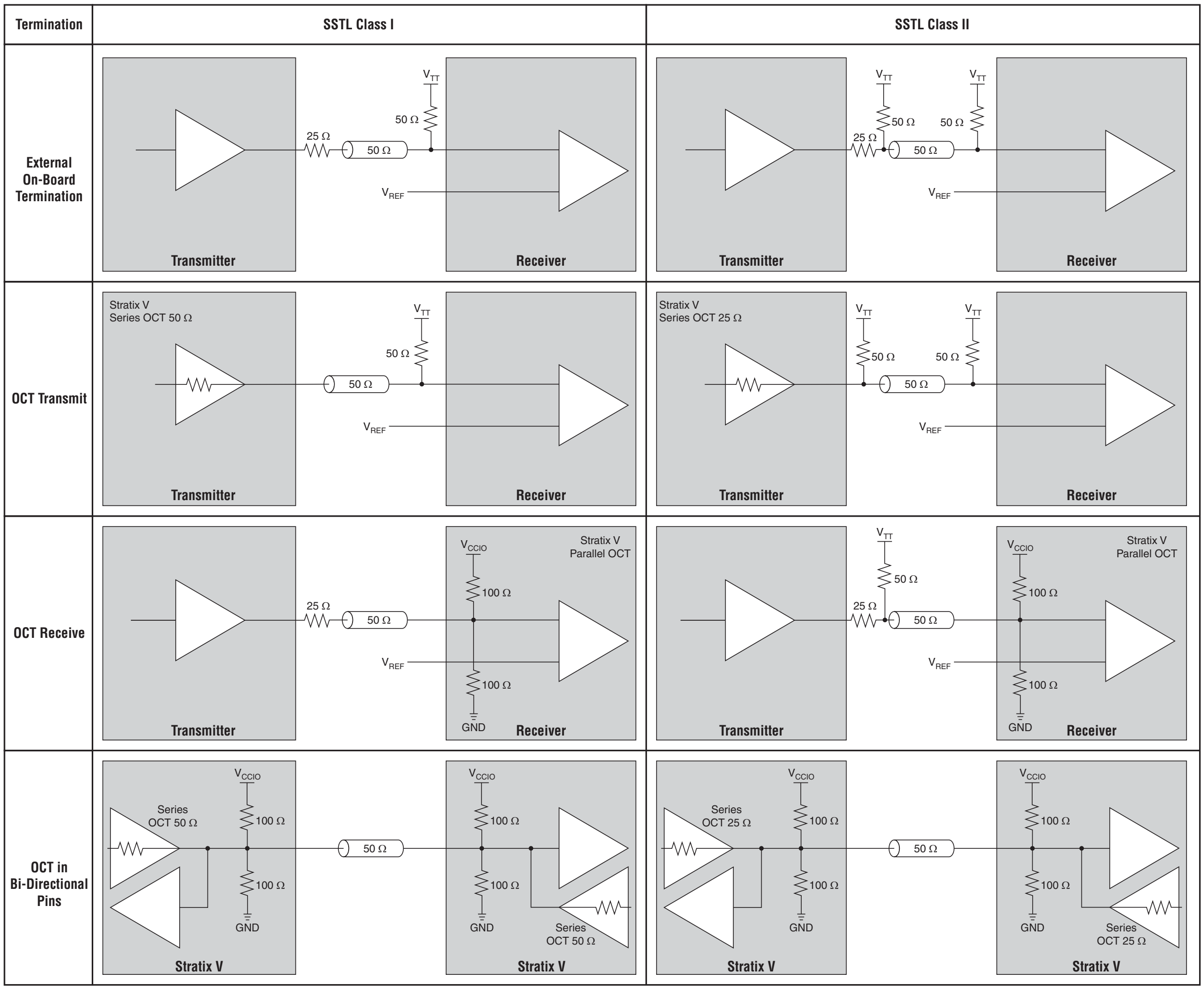
Single-Ended I/O Standards Termination

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.

The supported I/O standards such as **SSTL-15**, **SSTL-135**, **SSTL-125**, and **SSTL-12** typically do not require external board termination. Altera recommends using dynamic OCT with these I/O standards to save board space and cost by reducing the number of external termination resistors used.

Figure 5–13 shows the details of **SSTL** I/O termination on Stratix V devices.

Figure 5–13. SSTL I/O Standard Termination *(Note 1)*

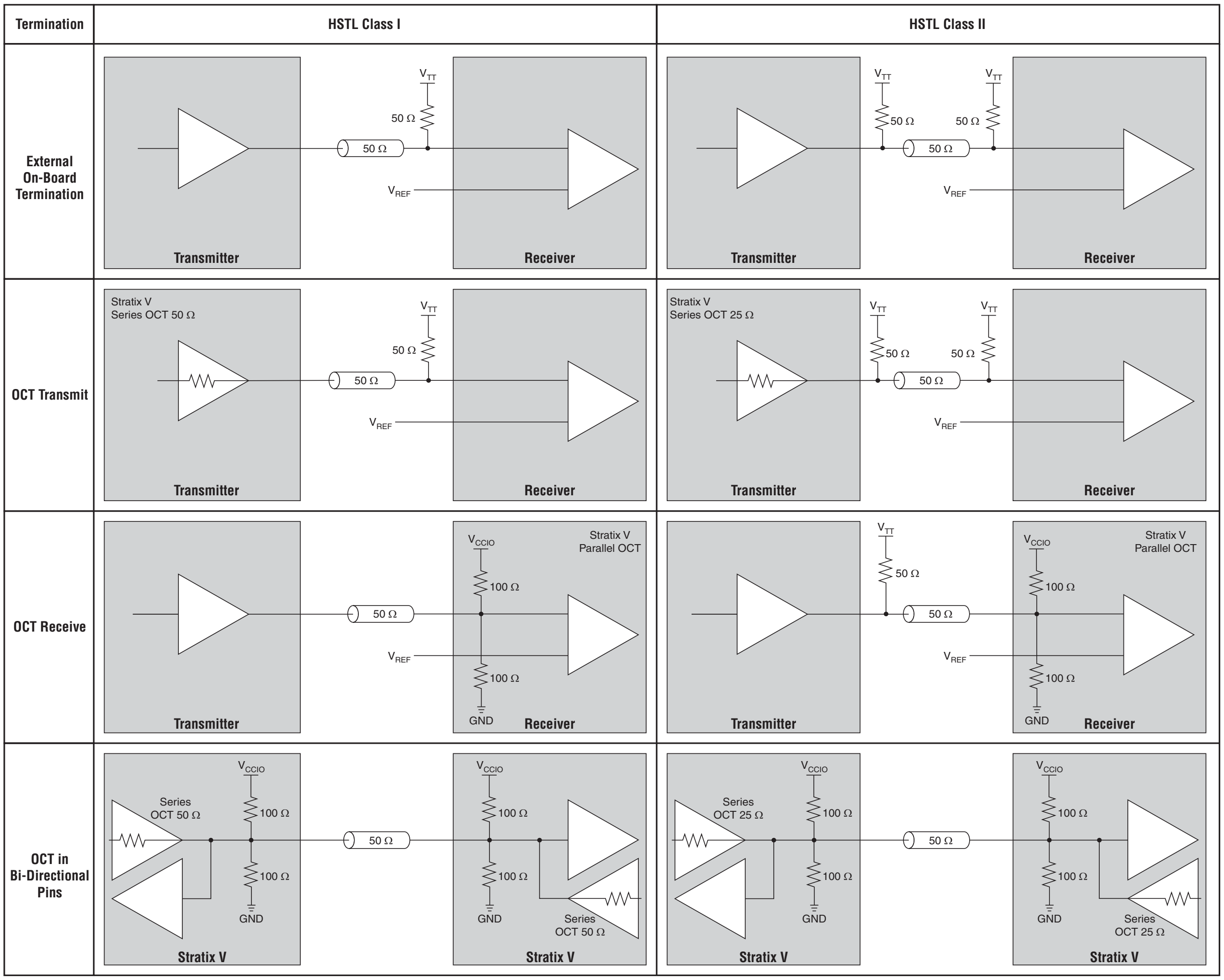


Note to Figure 5–13:

(1) This is not applicable for **SSTL-12**, **SSTL-15**, **SSTL-125,** and **SSTL-135** I/O standards.

Figure 5–14 shows the details of **HSTL** I/O termination on Stratix V devices.

Figure 5–14. HSTL I/O Standard Termination ***(Note 1)***



Note to Figure 5–14:

(1) This is not applicable for **HSUL-12** I/O standard.

You cannot use R_S and R_T OCT simultaneously. For more information, refer to "Dynamic OCT" on page 5–19.

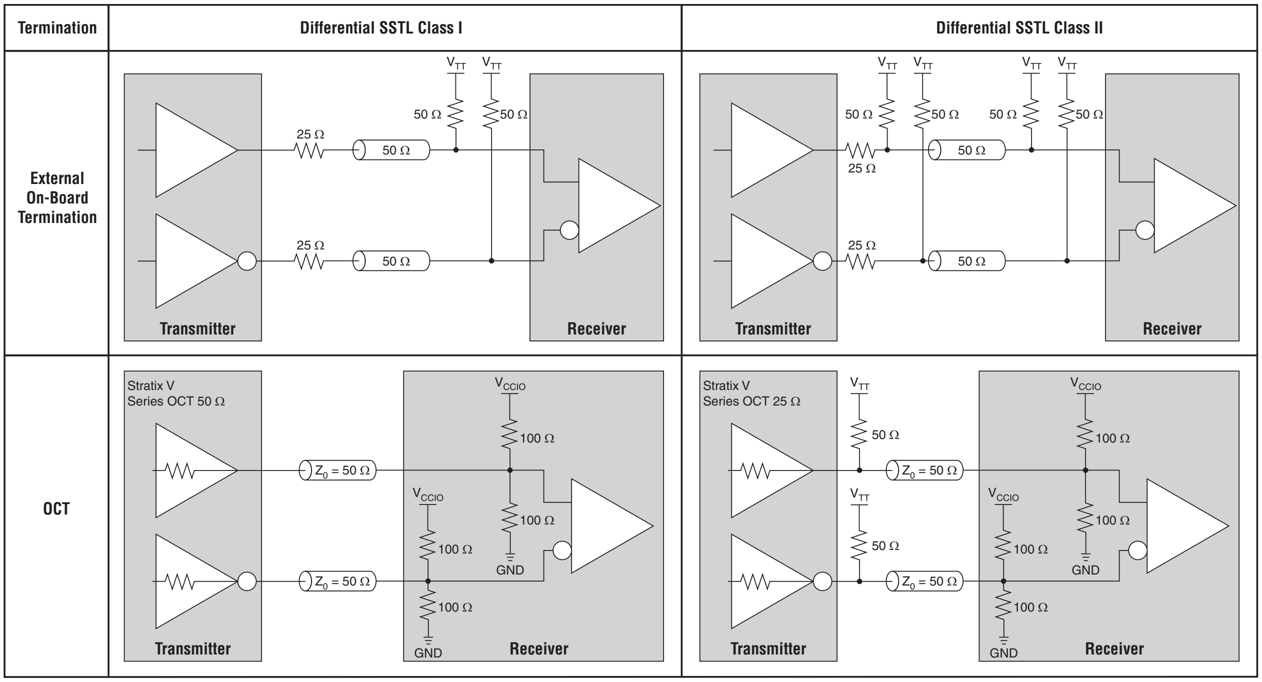
Differential I/O Standards Termination

Stratix V devices support **differential SSTL-18** and **SSTL-2**, **differential HSTL-18**, **HSTL-15**, **HSTL-12**, **LVDS**, **LVPECL**, **RSDS**, and **mini-LVDS** I/O standards. Figure 5–15 through Figure 5–21 show the details of various differential I/O terminations on these devices.

The supported I/O standards such as **differential SSTL-12**, d**ifferential SSTL-15**, **differential SSTL-125**, and **differential SSTL-135** typically do not require external board termination. Altera recommends using these I/O standards with dynamic OCT schemes to save board space and costs by reducing the number of external termination resistors used.

Differential HSTL, **SSTL**, and **HSUL** I/O standard outputs are not true differential outputs. They use two single-ended outputs with the second output programmed as inverted.

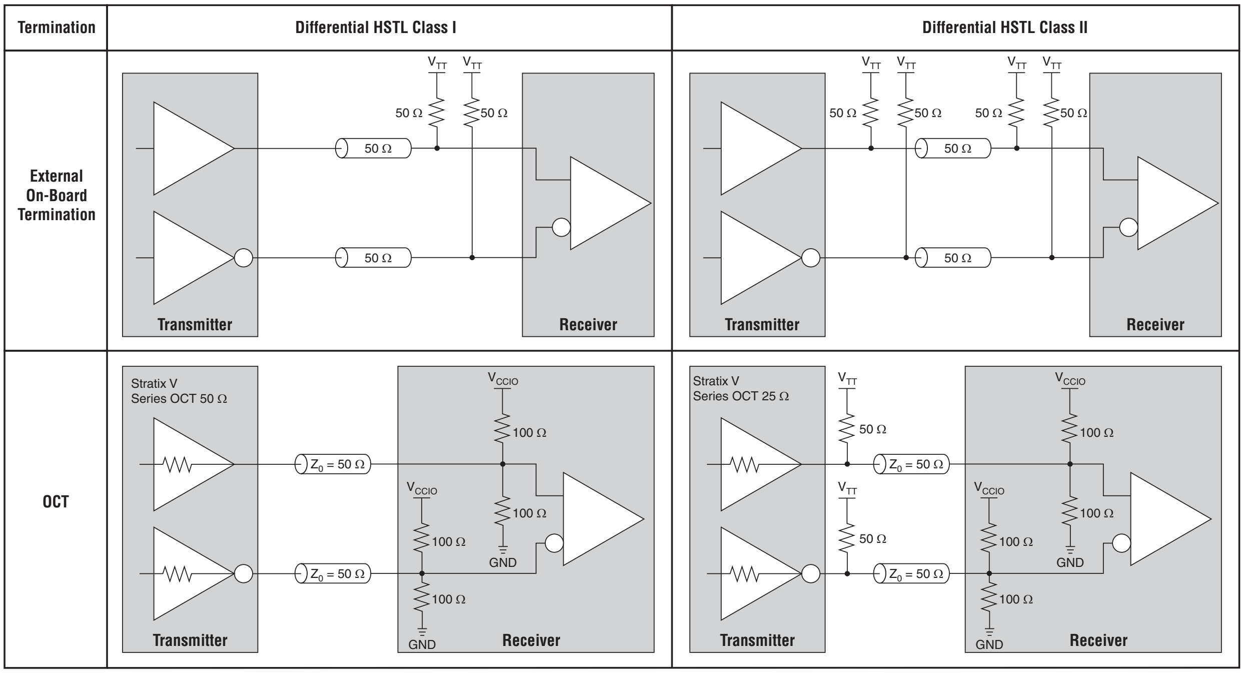
Figure 5–15. Differential SSTL I/O Standard Termination *(Note 1)*



Note to Figure 5–15:

(1) This is not applicable for **differential SSTL-12**, **differential SSTL-15**, **differential SSTL-125**, **differential SSTL-135**, and **differential HSUL-12** I/O standards.

Figure 5–16. Differential HSTL I/O Standard Termination *(Note 1)*


Termination
Differential HSTL Class I
Differential HSTL Class II
External On-Board Termination
V_{TT}
50 Ω
Transmitter
Receiver
OCT
Stratix V Series OCT 50 Ω
Stratix V Series OCT 25 Ω
Z_0 = 50 Ω
V_{CCIO}
100 Ω
GND

Note to Figure 5–16:

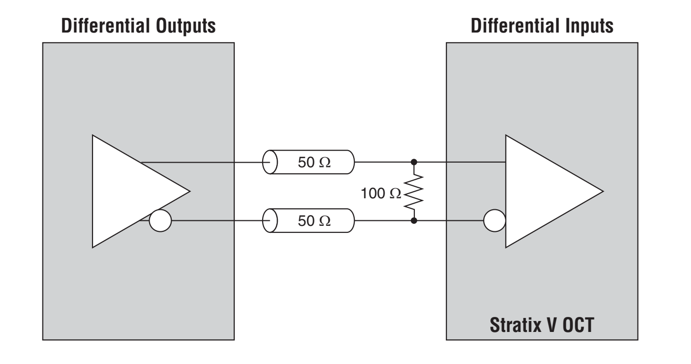
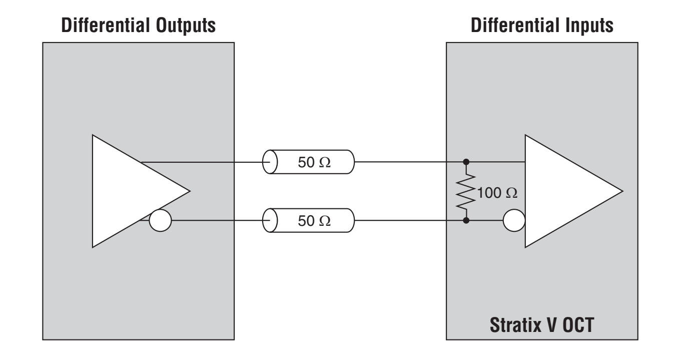
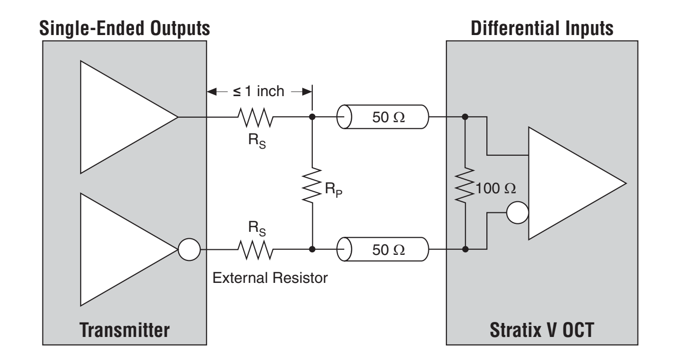
(1) This is not applicable for **differential HSUL-12** I/O standard.

LVDS

In Stratix V devices, the **LVDS** I/O standard requires a 2.5-V V_{CCIO} level. The **LVDS** input buffer requires 2.5-V V_{CCPD}. The **LVDS** receiver requires a 100-Ω termination resistor between the two signals at the input buffer. Stratix V devices provide an optional 100-Ω differential termination resistor in the device using R_D OCT when V_{CCIO} and V_{CCPD} are set to 2.5 V.

Figure 5–17 shows **LVDS** I/O standard termination. The on-chip differential resistor is available in all I/O banks.

Figure 5–17. LVDS I/O Standard Termination *(Note 1)*

Termination	LVDS
External On-Board Termination	
OCT Receiver (True LVDS Output) *(2)*	
OCT Receive (Single-Ended Output with Three-Resistor Network, LVDS_E_3R) *(2)*	

Notes to Figure 5–17:

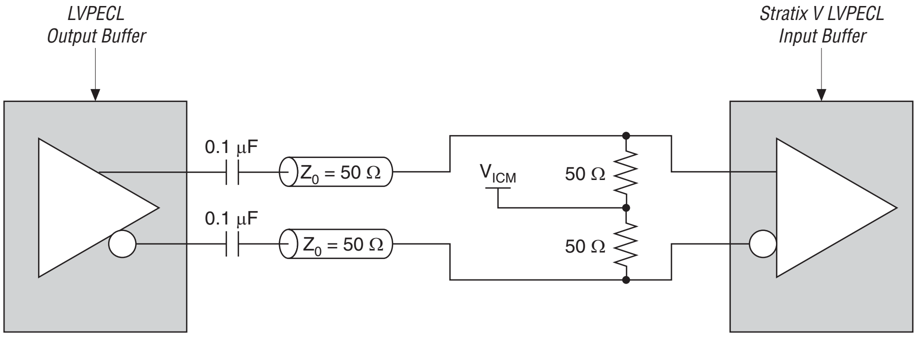
(1) For **LVDS** output with a three-resistor network, the R_S and R_P values are 120 Ω and 170 Ω, respectively. For **LVDS** output with one-resistor network, the R_P value is 120 Ω.

(2) All I/O banks support true LVDS output buffers and emulated **LVDS_E_3R** I/O standards. The emulated **LVDS** output buffers are configured with two single-ended output buffers and can be tri-stated.

Differential LVPECL

In Stratix V devices, the **LVPECL** I/O standard is supported on input clock pins on all the I/O banks. **LVPECL** output operation is not supported in Stratix V 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 not matched to the **LVPECL** input common-mode voltage. Figure 5–18 shows the AC-coupled termination scheme. The 50-Ω resistors used at the receiver end are external to the device.

Figure 5–18. LVPECL AC-Coupled Termination ***(Note 1)***

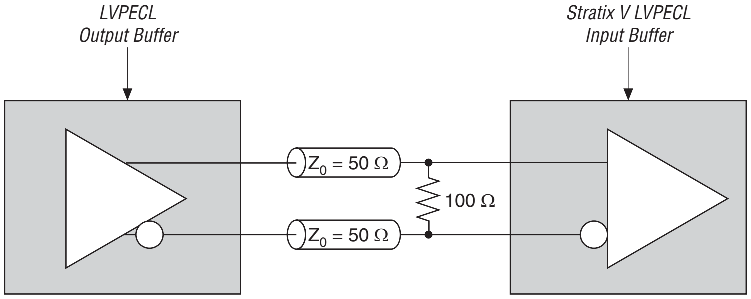


Note to Figure 5–18:

(1) The **LVPECL** AC/DC-coupled termination is applicable only when an Altera FPGA transmitter is used.

DC-coupled **LVPECL** is supported if the **LVPECL** output common mode voltage is within the Stratix V **LVPECL** input buffer specification (refer to Figure 5–19).

Figure 5–19. LVPECL DC-Coupled Termination ***(Note 1)***



Note to Figure 5–19:

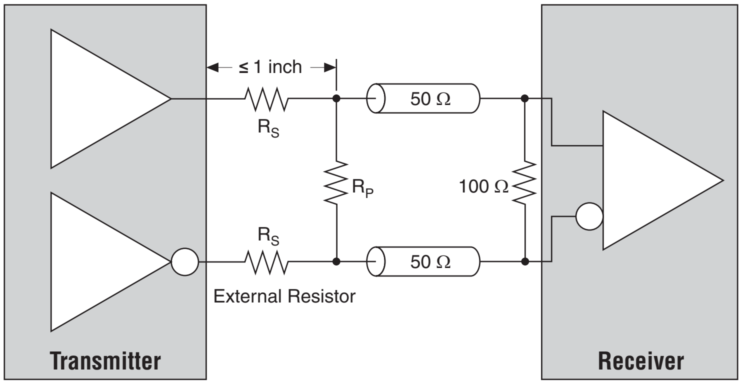
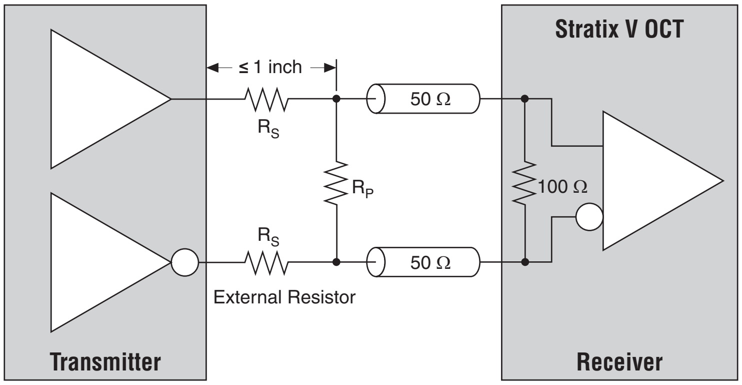
(1) The **LVPECL** AC/DC-coupled termination is applicable only when an Altera FPGA transmitter is used.

RSDS

Stratix V devices support the true **RSDS** output standard with data rates up to 360 Mbps using true **LVDS** output buffer types on all I/O banks. Emulated **RSDS** output buffers use two single-ended output buffers with external three-resistor networks and can be tri-stated. They are available in all I/O banks (refer to Figure 5–20).

Figure 5–20 shows the emulated RSDS I/O standard termination.

Figure 5–20. Emulated RSDS I/O Standard Termination *(Note 1)*

Termination	Three-Resistor Network (RSDS_E_3R)
External On-Board Termination	 ≤ 1 inch R_S R_P R_S 50 Ω 50 Ω 100 Ω External Resistor Transmitter Receiver
OCT	 ≤ 1 inch R_S R_P R_S 50 Ω 50 Ω 100 Ω Stratix V OCT External Resistor Transmitter Receiver

Note to Figure 5–20:

(1) The R_S and R_P values are pending characterization.

A resistor network is required to attenuate the output-voltage swing to meet **RSDS** specifications. You can modify the three-resistor network values to reduce power or improve the noise margin. The resistor values chosen must satisfy Equation 5–1.

Equation 5–1.

$$\frac{R_S \times \frac{R_p}{2}}{R_S + \frac{R_p}{2}} = 50\Omega$$

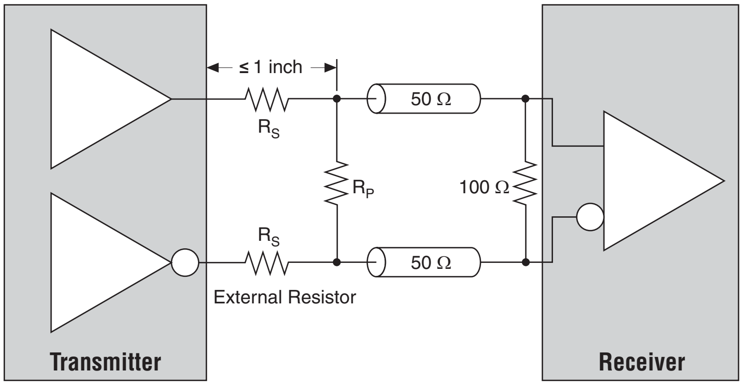
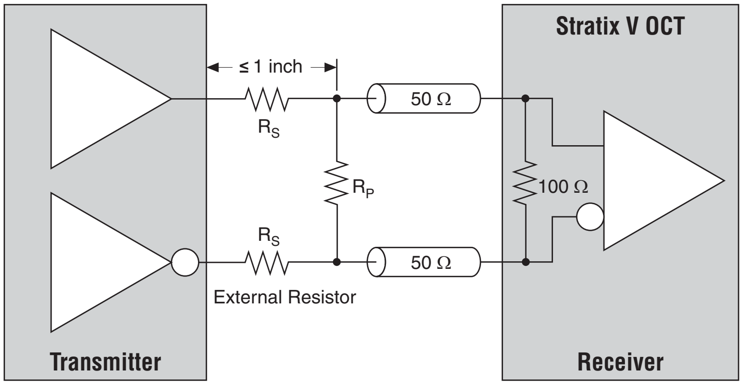
Altera recommends performing additional simulations using IBIS or SPICE models to validate that custom resistor values meet the **RSDS** I/O standard requirements.

For more information about the **RSDS** I/O standard, refer to the *RSDS Specification* available on the National Semiconductor website (www.national.com).

Mini-LVDS

Stratix V devices support the true **mini-LVDS** output standard with data rates up to 400 Mbps using **LVDS** output buffer types on all I/O banks. Emulated **mini-LVDS** output buffers use two single-ended output buffers with external three-resistor networks and can be tri-stated. They are available in all I/O banks (refer to Figure 5–21).

Figure 5–21. Emulated Mini-LVDS I/O Standard Termination ***(Note 1)***

Termination	Three-Resistor Network (RSDS_E_3R)
External On-Board Termination	 ≤ 1 inch R_S R_P R_S 50 Ω 50 Ω 100 Ω External Resistor Transmitter Receiver
OCT	 Stratix V OCT ≤ 1 inch R_S R_P R_S 50 Ω 50 Ω 100 Ω External Resistor Transmitter Receiver

Note to Figure 5–21:

(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 5–1 on page 5–33.

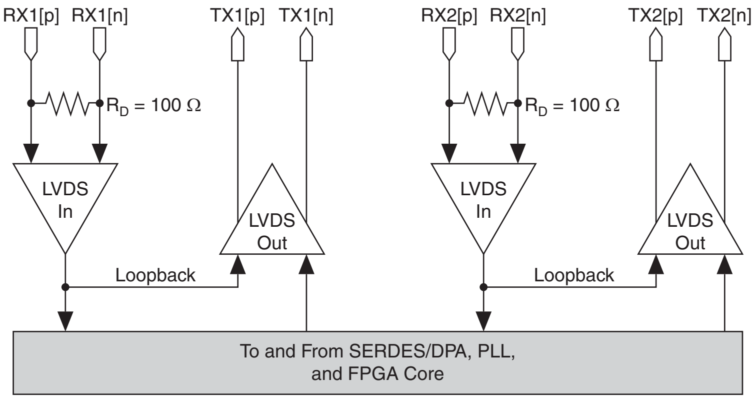
Altera recommends performing simulations using IBIS or SPICE models to validate that custom resistor R_S and R_P values meet the **mini-LVDS** requirements.

LVDS Direct Loopback Mode

LVDS direct loopback mode is available for true **LVDS** driver and receiver pairs in the I/O module that have the same pad group number. Figure 5–22 shows a datapath example of the **LVDS** direct loopback mode for I/O module 1 (RX1 and TX1) and I/O module 2 (RX2 and TX2). The data comes in from the receiver pin through the true **LVDS** input buffer and loops back to the true **LVDS** output buffer. You can use the **LVDS** direct loopback mode to verify the RX and TX buffers by checking the transmitted and received data.

☞ **LVDS** direct loopback mode is not supported for **LVDS** driver and receiver pairs from different I/O modules (for example, between RX1 and TX2).

Figure 5–22. LVDS Direct Loopback Path



To use the loopback mode, you must reset and recompile your existing design. The Quartus II software generates an error if the specified I/O standard and pin direction do not meet the requirements for direct loopback mode.

For an **LVDS** output pair already in use, you can apply the **LVDS** direct loopback mode to override the connection from the device core with the signal from the true differential input buffer in the same I/O module. Use this mode to observe the true **LVDS** input signal of a completed design.

Design Considerations

Although Stratix V 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 V devices.

Non-Voltage-Referenced Standards

Each I/O bank of a Stratix V device has its own VCCIO pins and supports only one V_{CCIO}, either 1.2, 1.25, 1.35, 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 (refer to Table 5–2 on page 5–3).

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 **3.0-V LVCMOS** output or bidirectional pins).

Voltage-Referenced Standards

To accommodate voltage-referenced I/O standards, each Stratix V 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 GPIO 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 R_T OCT enabled requires the V_{CCIO} of the I/O bank to match the voltage of the input standard. R_T OCT cannot be supported for the **HSTL-15** I/O standard when V_{CCIO} is 2.5 V.

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}.

V_{CCPD} Restriction

One VCCPD pin is shared in the particular group of I/O banks. For example, the I/O banks with the same number 7A, 7B, 7C, and 7D form a group that share the same VCCPD pin. This is true for all I/O banks except for I/O banks 3A, 3B, 3C, 3D, and 3E. Banks 3A and 3B form a group with one VCCPD pin while banks 3C, 3D, and 3E form a different group with its own VCCPD pin.

☞ Not all Stratix V device packages have bank 3E.

If one I/O bank is using 3.0-V V_{CCPD}, other I/O banks in the same group must also use 3.0-V V_{CCPD}.

☞ If you are using an output or bidirectional pin with the **3.3 V-LVTTL/LVCMOS** I/O standard, you must adhere to this restriction manually with location assignments.

Document Revision History

Table 5–13 lists the revision history for this chapter.

Table 5–13. Document Revision History

Date	Version	Changes
May 2011	1.3	■ Chapter moved to volume 2 for the 11.0 release. ■ Added Table 5–4, Table 5–5, Table 5–6, Table 5–7, and Table 5–8. ■ Updated "Single-Ended I/O Standards Termination", "Differential I/O Standards Termination", and "V_{CCPD} Restriction" sections. ■ Updated Table 5–3 and Table 5–11. ■ Updated Figure 5–1, Figure 5–8, Figure 5–9, Figure 5–10, Figure 5–17, Figure 5–20, and Figure 5–21. ■ Minor text edits.
January 2011	1.2	Updated Table 5–2.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

6. High-Speed Differential I/O Interfaces and DPA in Stratix V Devices

SV51007-1.2

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® V FPGAs.

The following sections describe the Stratix V high-speed differential I/O interfaces and DPA in detail:

Overview

All Stratix V devices have built-in serializer/deserializer (SERDES) circuitry that supports high-speed LVDS interfaces at data rates of up to 1.434 Gbps. SERDES circuitry is configurable to support source-synchronous communication protocols such as RapidIO™, XSBI, serial peripheral interface (SPI), and asynchronous protocols such as SGMII and Gigabit Ethernet (GbE).

The Stratix V device family has the following dedicated circuitry for high-speed differential I/O support:

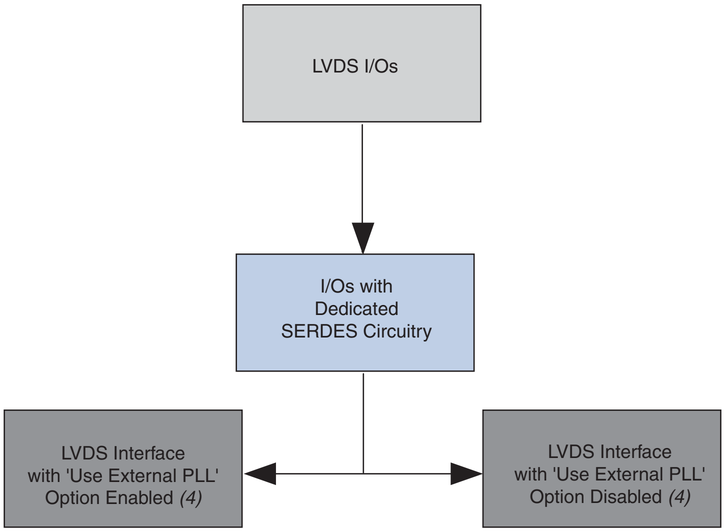
- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment
- DPA
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs)

For high-speed differential interfaces, the Stratix V device family supports the following differential I/O standards:

- **LVDS**
- **Mini-LVDS**
- Reduced swing differential signaling (**RSDS**)

Stratix V devices support ubiquitous I/Os. Row and column I/Os support the same features. Figure 6–1 shows I/O bank support for the Stratix V device family.

Figure 6–1. I/O Bank Support in the Stratix V Device Family *(Note 1)*, *(2)*, *(3)*


LVDS I/Os
I/Os with Dedicated SERDES Circuitry
LVDS Interface with 'Use External PLL' Option Enabled *(4)*
LVDS Interface with 'Use External PLL' Option Disabled *(4)*

Notes to Figure 6–1:

(1) All I/O banks support **LVDS**, **RSDS**, and **mini-LVDS** I/O standards using true LVDS output buffers without resistor networks.

(2) All I/O banks support **RSDS** and **mini-LVDS** I/O standards using two single-ended output buffers with a three-resistor (RSDS_E_3R and mini-LVDS_E_3R) network.

(3) 100-Ω differential input termination (R_D OCT) is supported in all I/O banks.

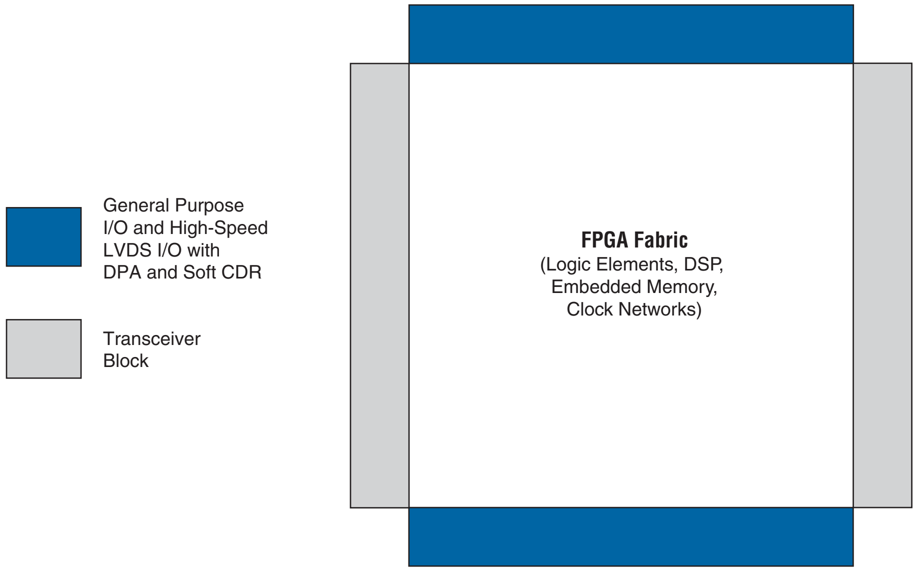
(4) The ALTLVDS **Use External PLL** option will only be available in future Quartus II software releases.

The ALTLVDS transmitter and receiver requires various clock and load enable signals from a fractional PLL. The Quartus® II software configures the PLL settings automatically. It is also responsible for generating the various clock and load enable signals based on the input reference clock and selected data rate.

Locations of the I/O Banks

Stratix V devices contain up to 26 I/O banks. The dedicated circuitry that supports high-speed differential I/Os is located in the top and bottom banks. Figure 6–2 shows the high-level SERDES/DPA location in the Stratix V devices.

Figure 6–2. High-Speed Differential I/Os with DPA Locations for Stratix V Devices


General Purpose I/O and High-Speed LVDS I/O with DPA and Soft CDR
Transceiver Block
FPGA Fabric
(Logic Elements, DSP, Embedded Memory, Clock Networks)

LVDS Channels

The Stratix V device family supports LVDS on all I/O banks. Both row and column I/Os support true LVDS input buffers with R_D OCT and true LVDS output buffers. Alternatively, you can configure the 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. Stratix V devices offer single-ended I/O reference clock support for the LVDS.

Stratix V devices support dedicated SERDES and DPA circuitry. For the supported I/O banks, refer to Figure 6–2.

☞ Emulated differential output buffers support tri-state capability.

Table 6–1 through Table 6–3 list the maximum number of LVDS I/Os supported in Stratix V 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.

Table 6–1. LVDS Channels Supported in Stratix V GX Devices *(Note 1)*, *(2)*—Preliminary

Device	5SGXA3				5SGXA4				5SGXA5				5SGXA7				5SGXA9				5SGXAB				5SGXB5				5SGXB6			
I/O Bank Side	Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom		Top		Bottom	
Clock Region *(3)*	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right	Left	Right
True LVDS Input Buffers, RX	42	39	42	39	42	39	42	39	54	51	54	51	54	51	54	51	47	43	47	43	47	43	47	43	42	39	42	39	42	39	42	39
True LVDS Output Buffers, TX	42	39	42	39	42	39	42	39	54	51	54	51	54	51	54	51	53	49	53	43	53	49	53	43	42	39	42	39	42	39	42	39
Emulated LVDS Output buffers, eTX	84	78	84	78	84	78	84	78	108	102	108	102	108	102	108	102	100	92	100	92	100	92	100	92	84	78	84	78	84	78	84	78

Notes to Table 6–1:

(1) The LVDS channel count does not include dedicated clock input pins.

(2) The number of LVDS receiver (RX) and transmitter (TX) channels are preliminary.

(3) For more information about the clock region, refer to the "Fractional PLLs and Stratix V Clocking" on page 6–22.

Table 6–2. LVDS Channels Supported in Stratix V GS Devices *(Note 1)*, *(2)***—Preliminary**

Device	5SGSD2				5SGSD3				5SGSD4				5SGSD5				5SGSD6						5SGSD8					
I/O Bank Side	**Top**		**Bottom**		**Top**		**Bottom**		**Top**		**Bottom**		**Top**		**Bottom**		**Top**		**Bottom**		**Right**		**Top**		**Bottom**		**Right**	
Clock Region *(3)*	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Top**	**Bottom**	**Left**	**Right**	**Left**	**Right**	**Top**	**Bottom**
True LVDS Input Buffers, RX	42	39	42	39	42	39	42	39	42	39	42	39	42	39	42	39	44	45	38	45	25	26	44	45	38	45	25	26
True LVDS Output Buffers, TX	42	39	42	39	42	39	42	39	42	39	42	39	42	39	42	39	50	47	44	47	31	32	50	47	44	47	31	32
Emulated LVDS Output buffers, eTX	84	78	84	78	84	78	84	78	84	78	84	78	84	78	84	78	94	92	82	92	56	58	94	92	82	92	56	58

Notes to Table 6–2:

(1) The LVDS channel count does not include dedicated clock input pins.

(2) The number of LVDS receiver (RX) and transmitter (TX) channels are preliminary.

(3) For more information about the clock region, refer to the "Fractional PLLs and Stratix V Clocking" on page 6–22.

Table 6–3. LVDS Channels Supported in Stratix V GT and E Devices *(Note 1)*, *(2)***—Preliminary**

Device	5SGTC5				5SGTC7				5SEE9				5SEEB			
I/O Bank Side	**Top**		**Bottom**		**Top**		**Bottom**		**Top**		**Bottom**		**Top**		**Bottom**	
Clock Region *(3)*	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**
True LVDS Input Buffers, RX	54	51	54	51	54	51	54	51	47	43	47	43	47	43	47	43
True LVDS Output Buffers, TX	54	51	54	51	54	51	54	51	53	49	53	43	53	49	53	43
Emulated LVDS Output buffers, eTX	108	102	108	102	108	102	108	102	100	92	100	92	100	92	100	92

Notes to Table 6–3:

(1) The LVDS channel count does not include dedicated clock input pins.

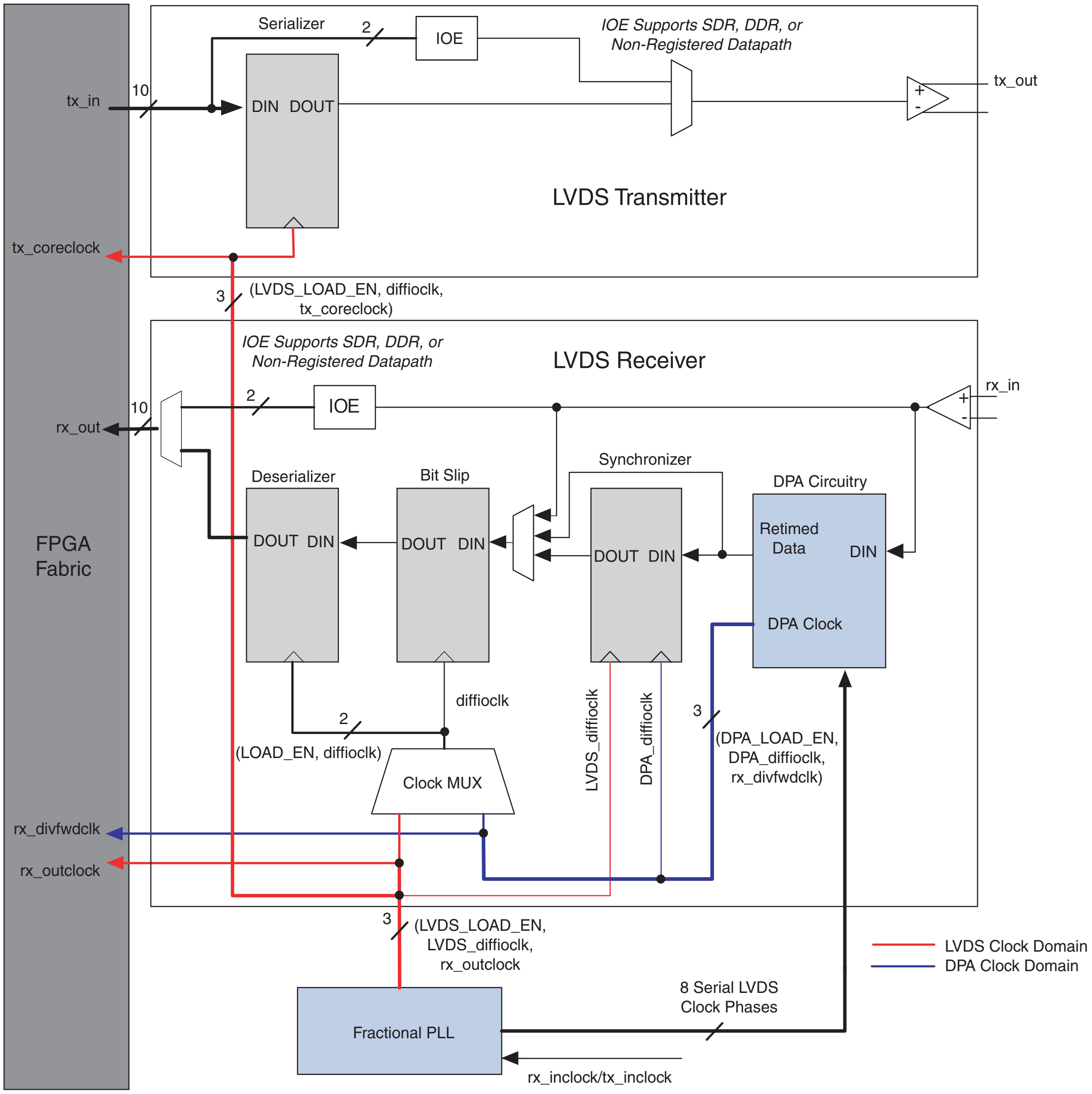
(2) The number of LVDS receiver (RX) and transmitter (TX) channels are preliminary.

(3) For more information about the clock region, refer to the "Fractional PLLs and Stratix V Clocking" on page 6–22.

LVDS SERDES

Figure 6–3 shows a transmitter and receiver block diagram for the LVDS SERDES circuitry. This diagram shows the interface signals of the transmitter and receiver datapath. For more information, refer to "Differential Transmitter" on page 6–7 and "Differential Receiver" on page 6–11.

Figure 6–3. LVDS SERDES ***(Note 1)*****,** ***(2)*****,** ***(3)***



Notes to Figure 6–3:

(1) This diagram shows a shared PLL between the transmitter and receiver. If the transmitter and receiver are not sharing the same PLL, two fractional PLLs are required.

(2) In single data rate (SDR) and double data rate (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.

For more information about the LVDS transmitter and receiver port list and settings using ALTLVDS, refer to the *ALTLVDS Megafunction User Guide*.

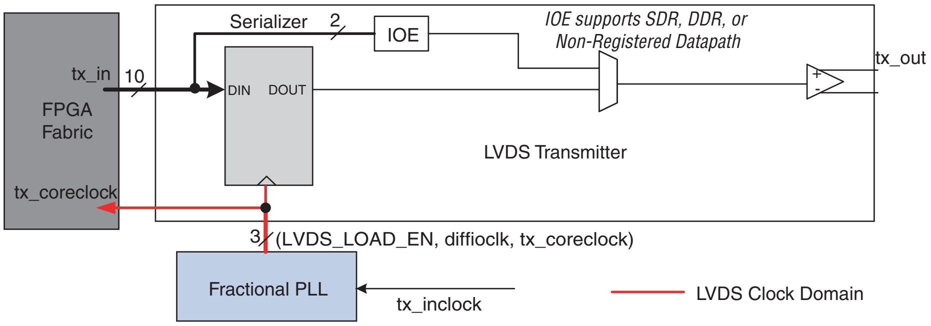
Differential Transmitter

The Stratix V transmitter has dedicated circuitry to provide support for LVDS signaling. The dedicated circuitry consists of a differential buffer, a serializer, and fractional 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 fractional 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 hard SERDES.

The load enable (`LVDS_LOAD_EN`) signal and the `diffioclk` signal (the clock running at serial data rate) generated from the fractional PLL clocks the load and shift registers. You can statically set the serialization factor to x3, x4, x5, x6, x7, x8, x9, or x10 using the Quartus II software. The load enable signal is derived from the serialization factor setting. Figure 6–4 shows a block diagram of the Stratix V transmitter.

Figure 6–4. Stratix V Transmitter ***(Note 1)*****,** ***(2)***



Notes to Figure 6–4:

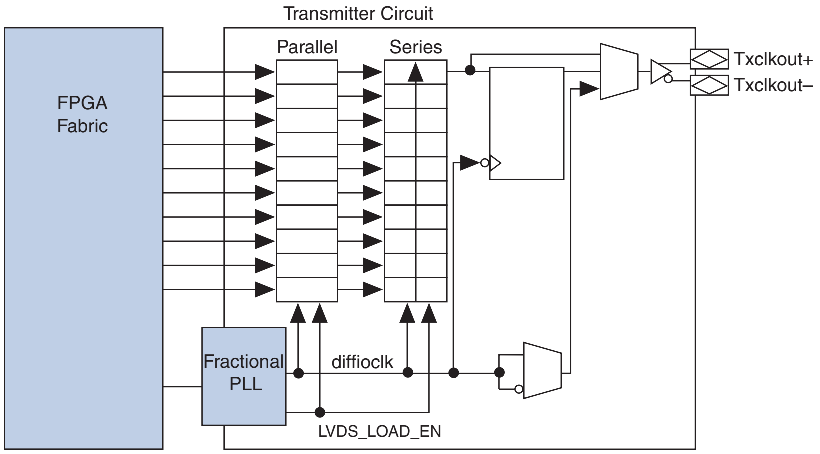
(1) In SDR and DDR modes, the data width is 1 and 2 bits, respectively.

(2) The `tx_in` port has a maximum data width of 10 bits.

You can configure any Stratix V 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 717 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

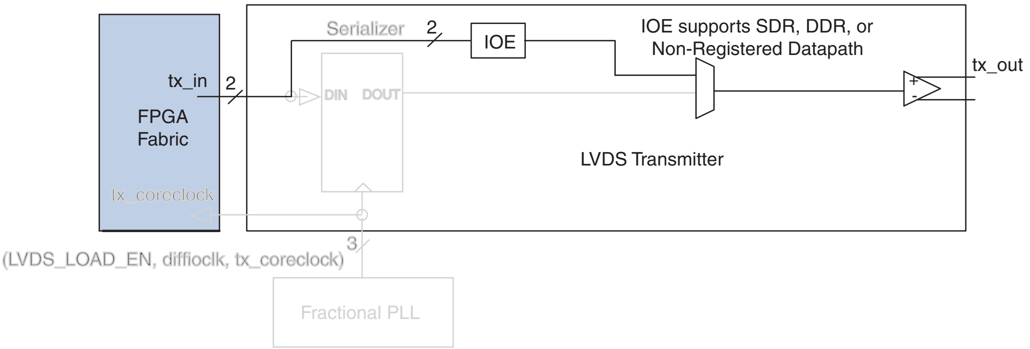
180° (edge or center aligned). The fractional PLLs provide additional support for other phase shifts in 45° increments. These settings are made statically in the Quartus II MegaWizard™ Plug-In Manager. Figure 6–5 shows the Stratix V transmitter in clock output mode. In clock output mode, you can use an LVDS channel as a clock output channel.

Figure 6–5. Stratix V Transmitter in Clock Output Mode



You can bypass the Stratix V serializer to support DDR (x2) and SDR (x1) 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 6–6 shows the serializer bypass path.

Figure 6–6. Stratix V Serializer Bypass *(Note 1)*, *(2)*, *(3)*



Notes to Figure 6–6:

(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 Differential Output Voltage (V_{OD}) and Programmable Pre-Emphasis

Stratix V 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 6–7 shows the differential LVDS output.

Figure 6–7. Differential V_{OD}

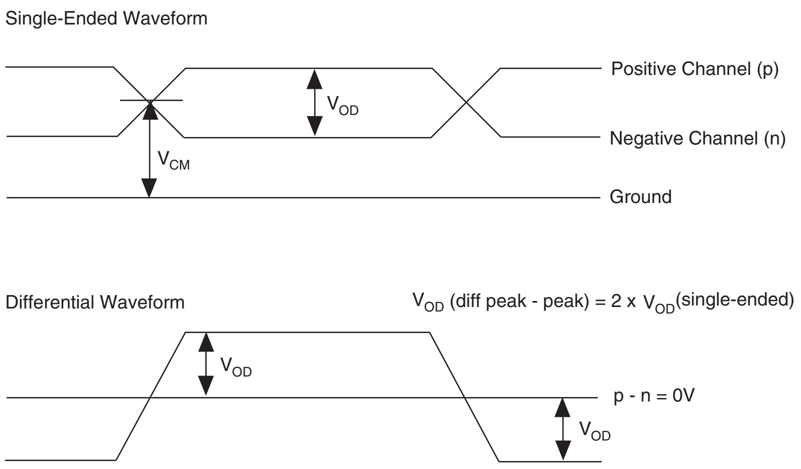
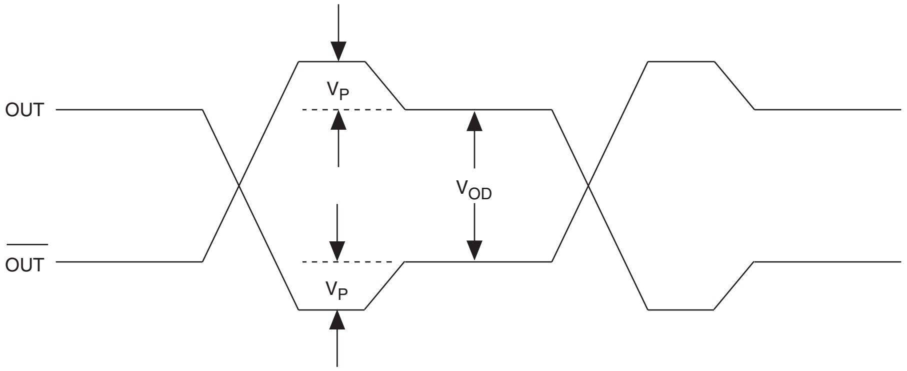


Figure 6–8 shows the LVDS output with pre-emphasis.

Figure 6–8. Programmable Pre-Emphasis *(Note 1)*



Note to Figure 6–8:

(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 two settings for programmable pre-emphasis—disabled (**0**) and enabled (**1**). The default setting is enabled.

The V_{OD} is programmable with four settings: low (**0**), medium low (**1**), medium high (**2**), and high (**3**). The default setting is medium low.

Programmable V_{OD}

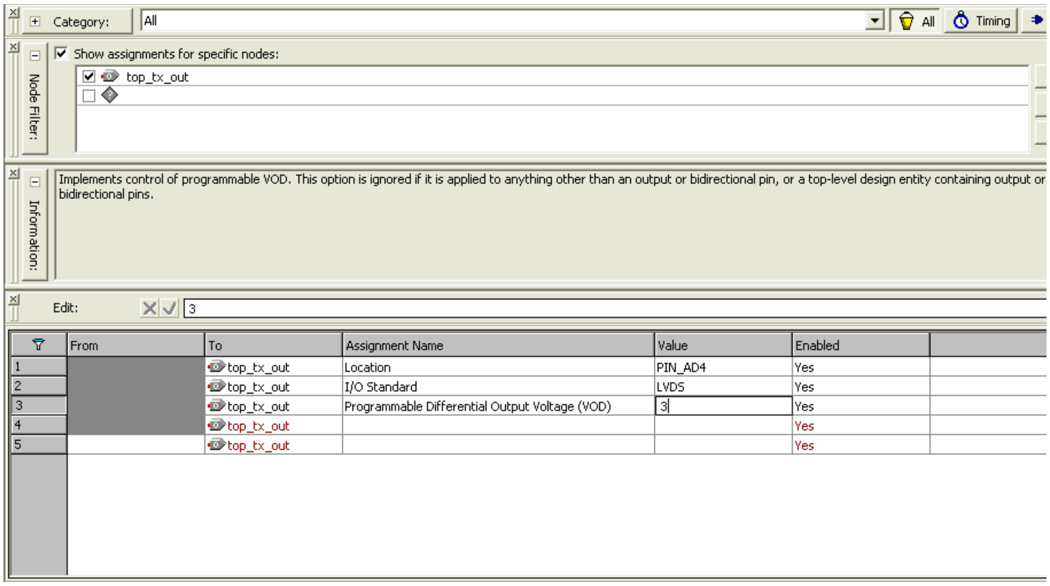
You can statically assign the V_{OD} settings from the Assignment Editor. Table 6–4 lists the assignment name for programmable V_{OD} and its possible values in the Quartus II software Assignment Editor.

Table 6–4. Quartus II Software Assignment Editor

Field	Assignment
To	`tx_out`
Assignment name	Programmable Differential Output Voltage (V_{OD})
Allowed values	0, 1, 2, 3

Figure 6–9 shows the assignment of programmable V_{OD} for a transmit data output from the Quartus II software Assignment Editor.

Figure 6–9. Quartus II Software Assignment Editor—Programmable V_{OD}



Category: All

☑ Show assignments for specific nodes:

☑ top_tx_out
☐

Information: Implements control of programmable VOD. This option is ignored if it is applied to anything other than an output or bidirectional pin, or a top-level design entity containing output or bidirectional pins.

Edit: 3

	From	To	Assignment Name	Value	Enabled
1		top_tx_out	Location	PIN_AD4	Yes
2		top_tx_out	I/O Standard	LVDS	Yes
3		top_tx_out	Programmable Differential Output Voltage (VOD)	3	Yes
4		top_tx_out			Yes
5		top_tx_out			Yes

Programmable Pre-Emphasis

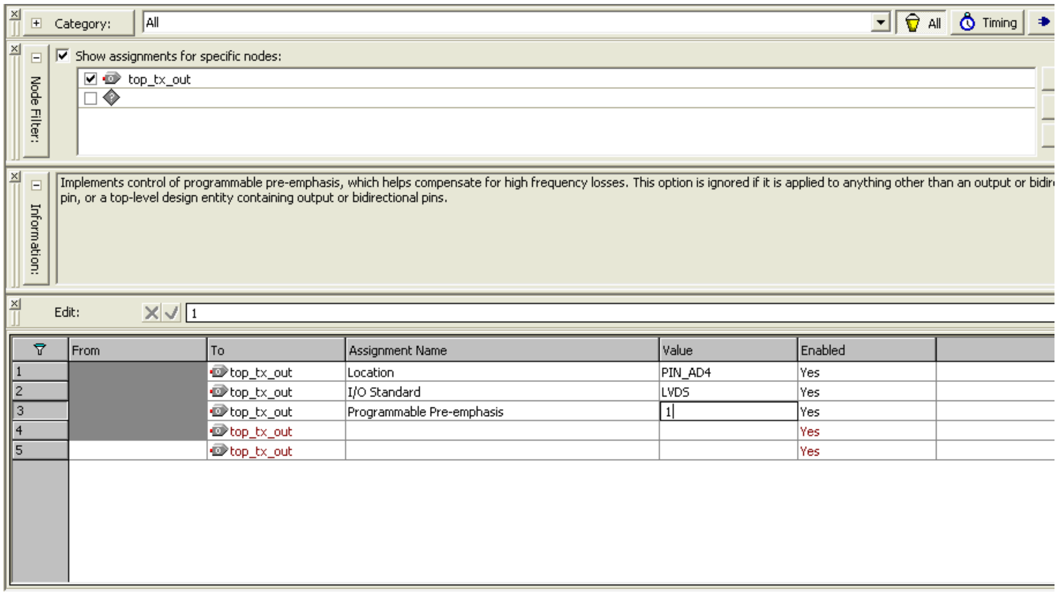
Two different settings are allowed for pre-emphasis from the Assignment Editor for each LVDS output channel. Table 6–5 lists the assignment name and its possible values for programmable pre-emphasis in the Quartus II software Assignment Editor.

Table 6–5. Quartus II Software Assignment Editor

Field	Assignment
To	`tx_out`
Assignment name	Programmable Pre-emphasis
Allowed values	0 (disable) and 1 (enable)

Figure 6–10 shows the assignment of programmable pre-emphasis for a transmit data output port from the Quartus II software Assignment Editor.

Figure 6–10. Quartus II Software Assignment Editor – Programmable Pre-Emphasis



☑ Show assignments for specific nodes:

☑ top_tx_out
□

Implements control of programmable pre-emphasis, which helps compensate for high frequency losses. This option is ignored if it is applied to anything other than an output or bidir
pin, or a top-level design entity containing output or bidirectional pins.

Edit: 1

	From	To	Assignment Name	Value	Enabled
1		top_tx_out	Location	PIN_AD4	Yes
2		top_tx_out	I/O Standard	LVDS	Yes
3		top_tx_out	Programmable Pre-emphasis	1	Yes
4		top_tx_out			Yes
5		top_tx_out			Yes

Differential Receiver

Figure 6–11 shows the hardware blocks of the Stratix V receiver. The receiver has a differential buffer and fractional 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.

For more information, refer to Figure 6–18 on page 6–19 and Figure 6–19 on page 6–20.

The fractional PLL receives the external clock input and generates different phases of the same clock. The DPA block chooses one of the clocks from the fractional 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 V device family supports three different receiver modes:

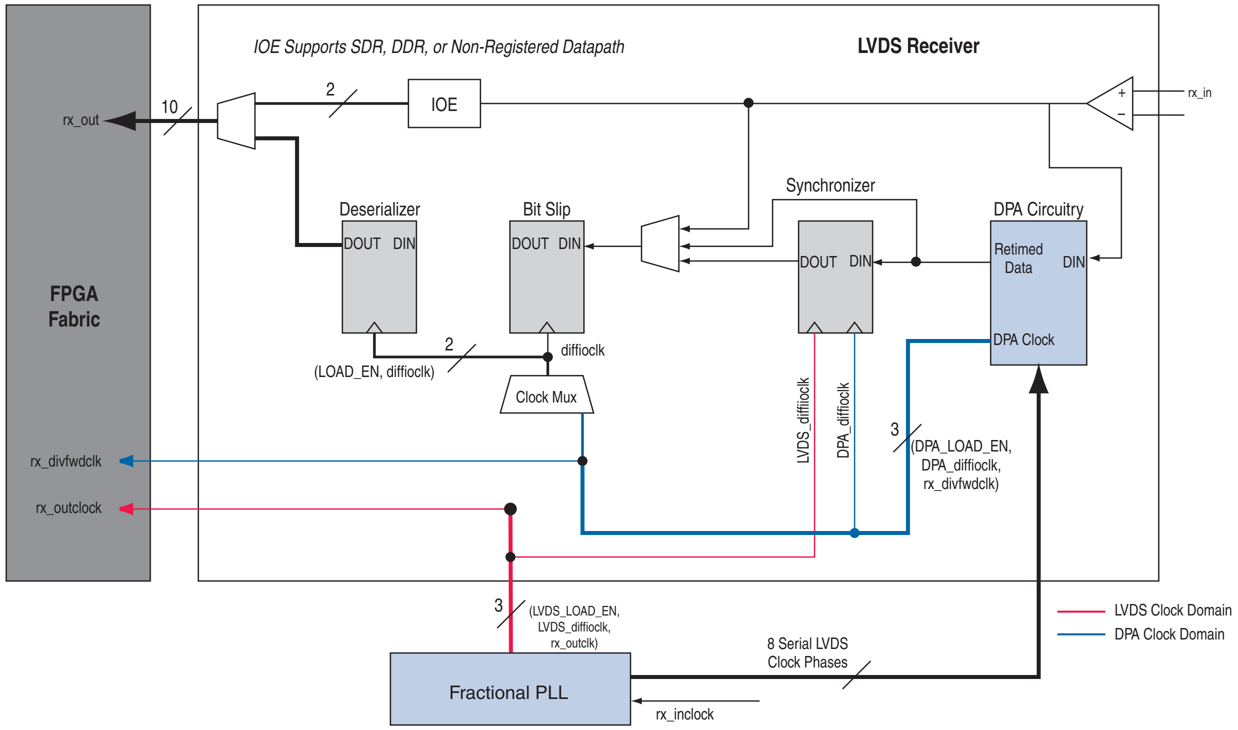
- "Non-DPA Mode" on page 6–18
- "DPA Mode" on page 6–19
- "Soft-CDR Mode" on page 6–20

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 the 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 6–11. Receiver Block Diagram *(Note 1), (2)*



Notes to Figure 6–11:

(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.

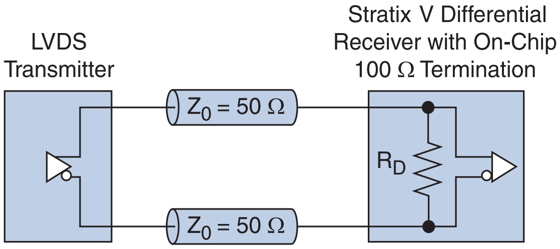
Differential I/O Termination

The Stratix V 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 I/O pins and dedicated clock input pins.

Figure 6–12 shows device on-chip termination.

Figure 6–12. On-Chip Differential I/O Termination



Receiver Hardware Blocks

The differential receiver has the following hardware blocks:

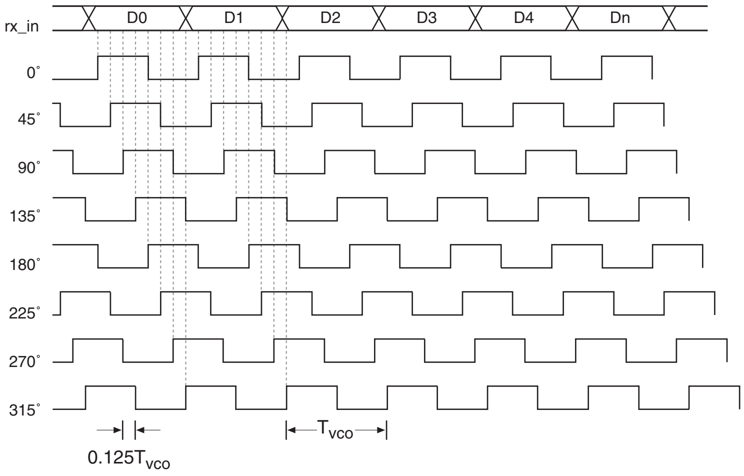
- "DPA Block" on page 6–14
- "Synchronizer" on page 6–15
- "Data Realignment Block (Bit Slip)" on page 6–15
- "Deserializer" on page 6–17

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 fractional PLLs 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 6–13 shows the possible phase relationships between the DPA clocks and the incoming serial data.

Figure 6–13. DPA Clock Phase to Serial Data Timing Relationship *(Note 1)*



Note to Figure 6–13:

(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 deasserted 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 fractional PLLs. 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 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.

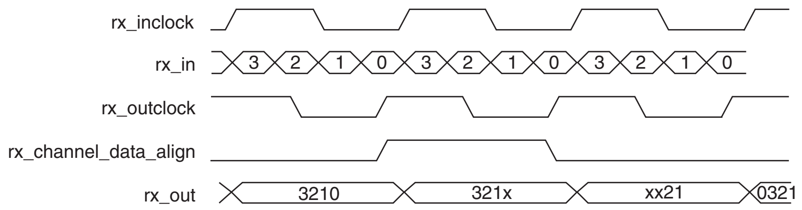
An optional RX_CHANNEL_DATA_ALIGN port controls the bit insertion of each receiver independently controlled from the internal logic. The data slips one bit on the rising edge of RX_CHANNEL_DATA_ALIGN. The requirements for the RX_CHANNEL_DATA_ALIGN signal include:

- 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 6–14 shows receiver output (RX_OUT) after one bit slip pulse with the deserialization factor set to **4**.

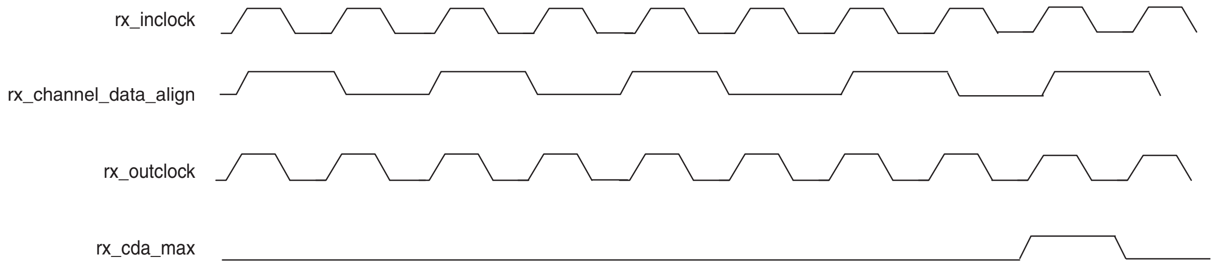
Figure 6–14. Data Realignment Timing



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. 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 6–15 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.

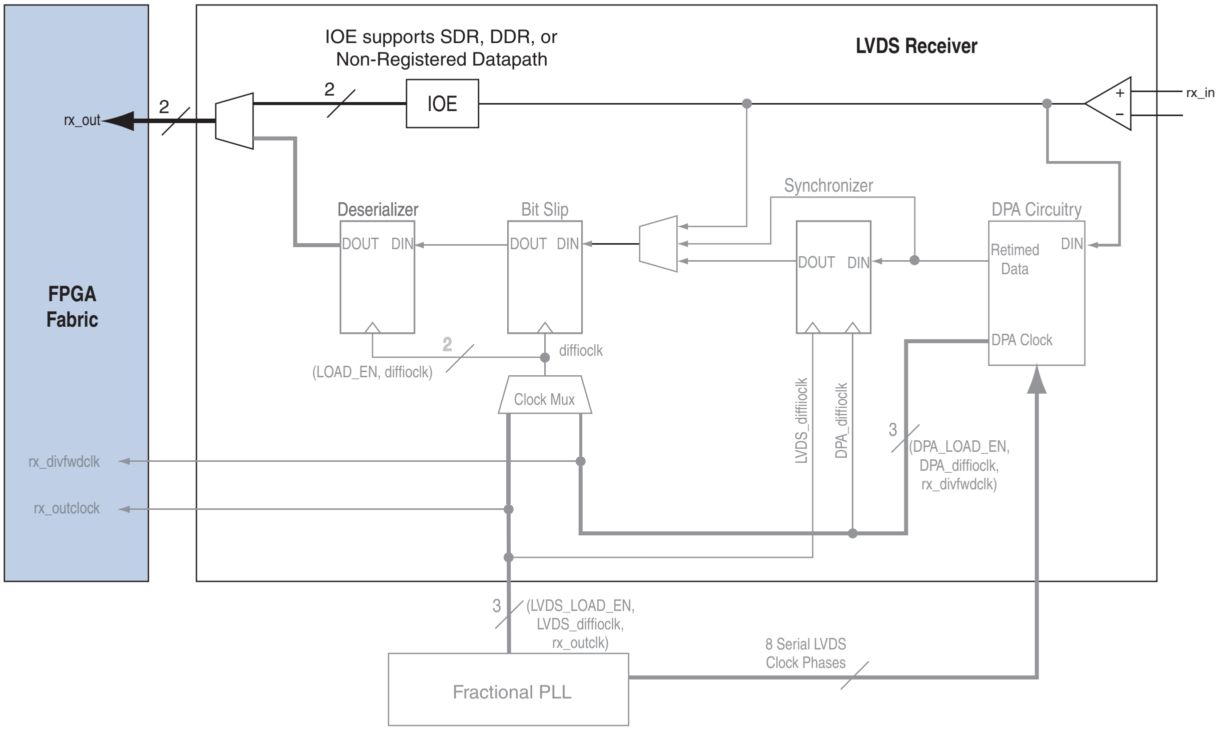
Figure 6–15. Receiver Data Realignment Rollover



Deserializer

You can statically set the deserialization factor to x3, x4, x5, x6, x7, x8, x9, or x10 by using the Quartus II software. You can bypass the Stratix V deserializer in the Quartus II MegaWizard Plug-In Manager to support DDR (x2) or SDR (x1) operations, as shown Figure 6–16. 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 6–16. Stratix V Deserializer Bypass *(Note 1)*, *(2)*, *(3)*



Notes to Figure 6–16:

(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.

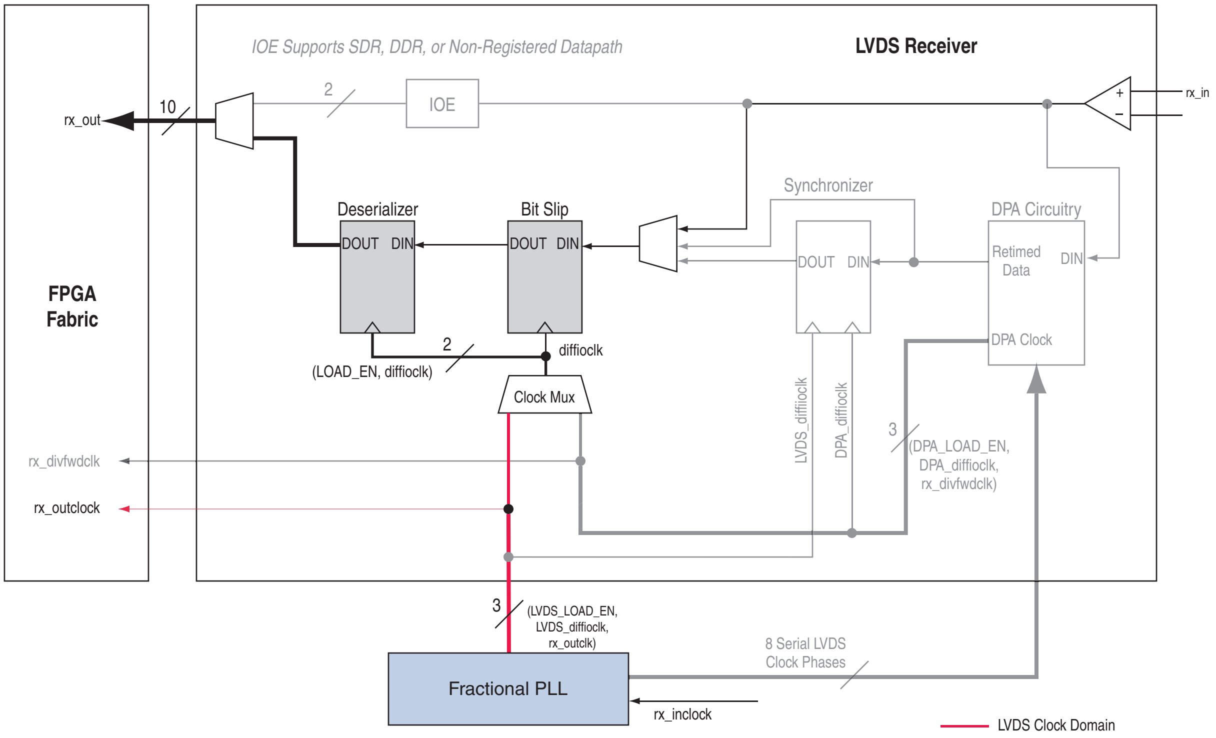
Receiver Datapath Modes

The Stratix V device family supports three receiver datapath modes: non-DPA mode, DPA mode, and soft-CDR mode.

Non-DPA Mode

Figure 6–17 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 PLLs. You can select the rising/falling edge option using the ALTLDVS MegaWizard Plug-In Manager. Both data realignment and deserializer blocks are clocked by the `LVDS_diffioclk` clock, which is generated by the left and right PLLs.

Figure 6–17. Receiver Datapath in Non-DPA Mode *(Note 1)*, *(2)*, *(3)*



Notes to Figure 6–17:

(1) All disabled blocks and signals are grayed out.

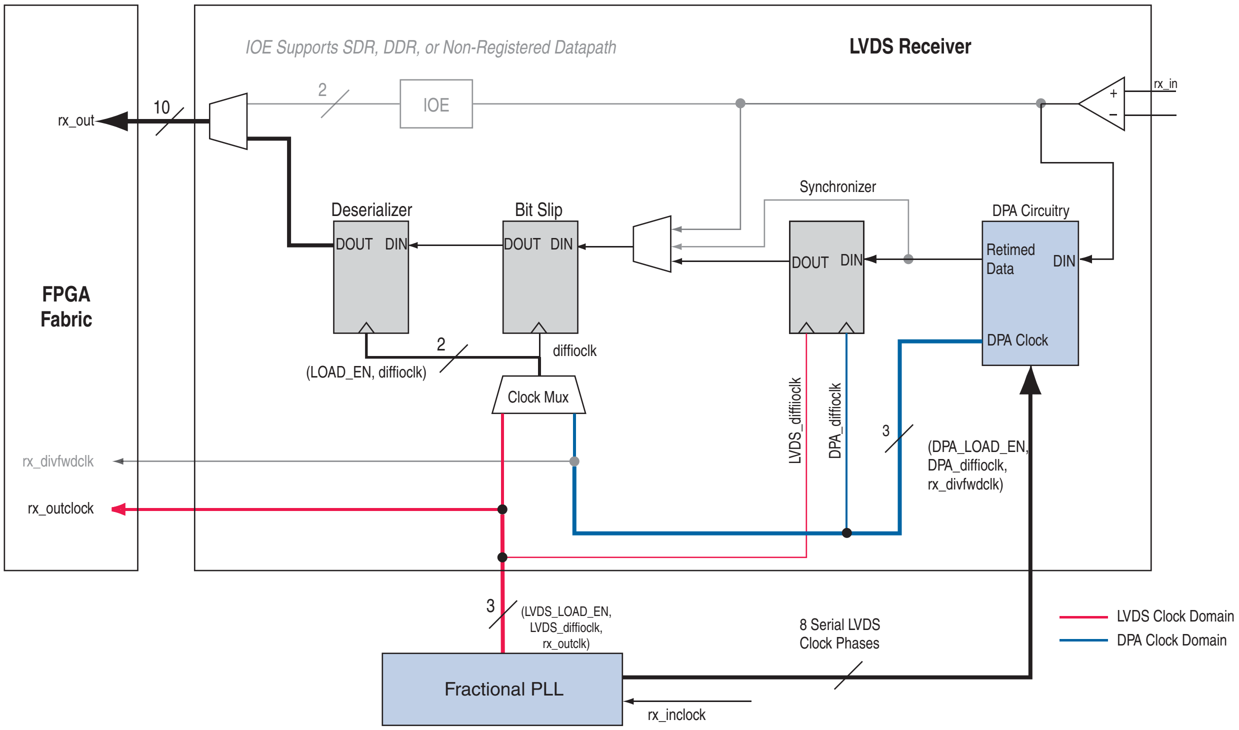
(2) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

(3) The `rx_out` port has a maximum data width of 10 bits.

DPA Mode

Figure 6–18 shows the DPA mode datapath, where all the hardware blocks mentioned in "Receiver Hardware Blocks" on page 6–14 are active. The DPA block chooses the best possible clock (`DPA_diffioclk`) from the eight fast clocks sent by the fractional 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 6–18. Receiver Datapath in DPA Mode *(Note 1), (2), (3)*



Notes to Figure 6–18:

(1) All disabled blocks and signals are grayed out.

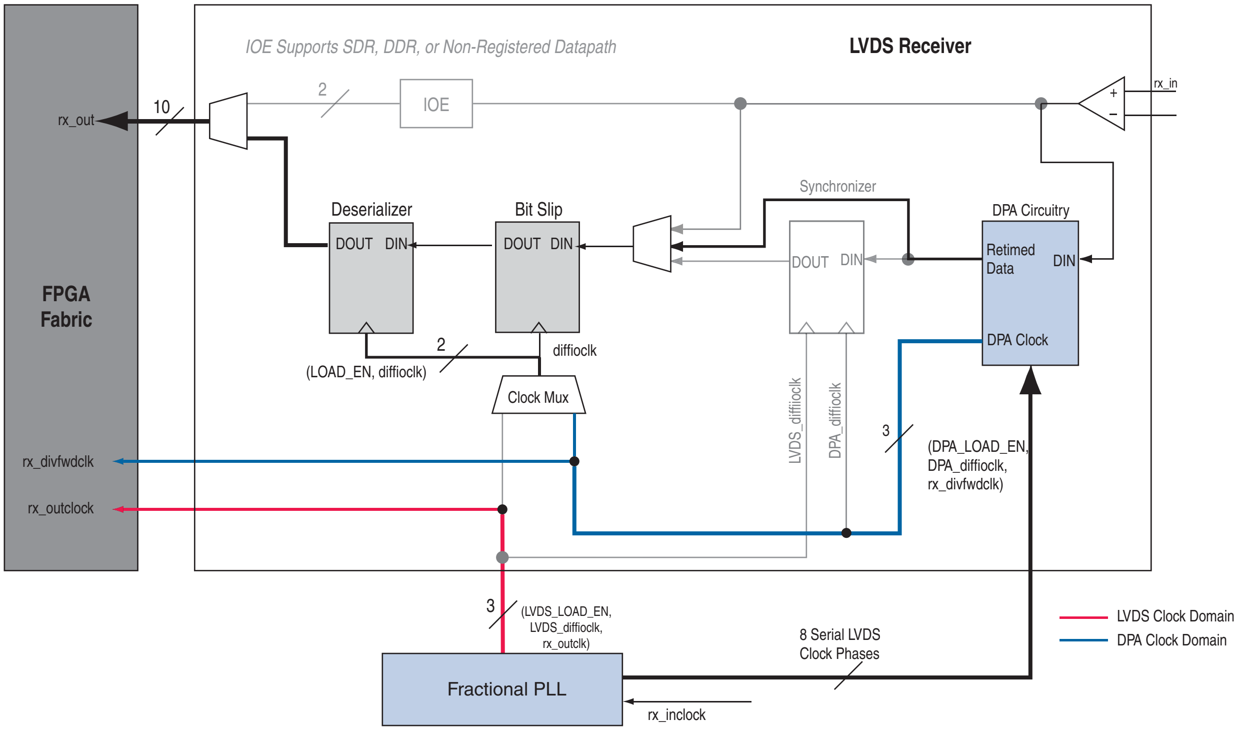
(2) In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

(3) The `rx_out` port has a maximum data width of 10 bits.

Soft-CDR Mode

The Stratix V LVDS channel offers the soft-CDR mode to support the GbE and SGMII protocols. A receiver PLL uses the local clock source for reference. Figure 6–19 shows the soft-CDR mode datapath.

Figure 6–19. Receiver Datapath in Soft-CDR Mode ***(Note 1)*, *(2)*, *(3)***



Notes to Figure 6–19:

(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.

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.

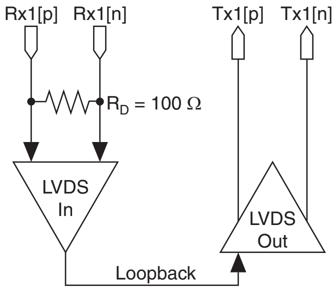
For more information about PCLK networks, refer to the *Clock Networks and PLLs in Stratix V Devices* chapter.

You can use every LVDS channel in soft-CDR mode and drive the FPGA fabric using the PCLK network in the Stratix V device family. The `rx_dpa_locked` signal is not valid in soft-CDR mode because 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 PLLs, is also forwarded to the FPGA fabric.

LVDS Direct Loopback Mode

The Stratix V device family supports direct loopback mode for the LVDS driver and receiver pairs within the same LVDS module only. Figure 6–20 shows the true LVDS input and output buffer from an I/O pair from the same module. LVDS direct loopback mode allows you to verify the LVDS driver and receiver pair by checking the incoming LVDS data from the true LVDS input buffer into the true LVDS output buffer.

Figure 6–20. LVDS Direct Loopback Path ***(Note 1)***



Note to Figure 6–20:

(1) The R_D value is pending characterization.

The Quartus II software allows two option settings for LVDS direct loopback mode in the assignment editor: **On** and **Off**.

This option is only available for true differential I/O standards only.

This option can be applied on the LVDS output pair that is already being used in the design. Turning the LVDS direct loopback mode option to **On** overrides the connection from core with the signal from the true differential input buffer in the same I/O module. You can disable this option after verifying the LVDS driver receiver pair. Turn the LVDS direct loopback mode option to **Off** and recompile your design.

LVDS Interface with the Use External PLL Option Enabled

The ALTLVDS MegaWizard Plug-In Manager 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.

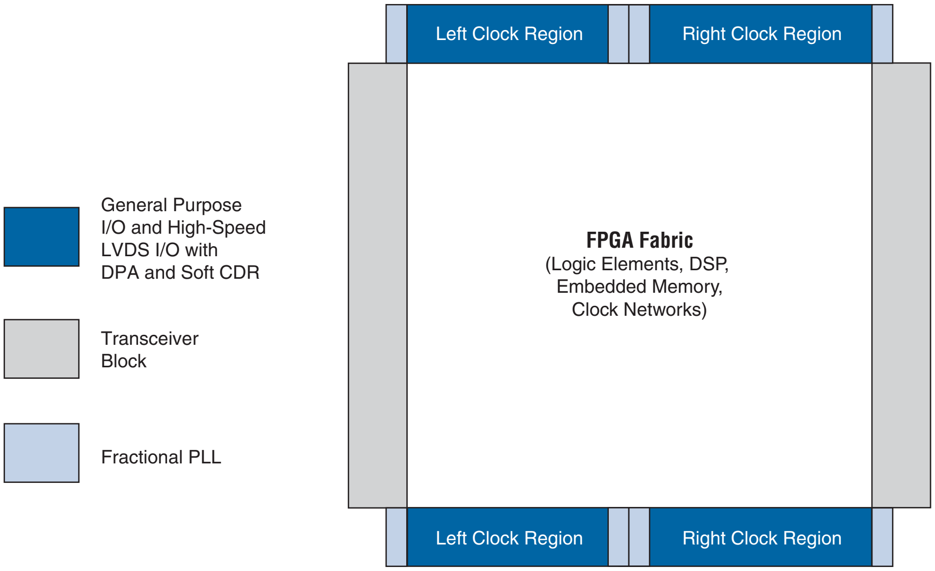
The ALTLVDS **Use External PLL** option will only be available in future Quartus II software releases.

Fractional PLLs and Stratix V Clocking

The Stratix V device family supports fractional PLLs on each side of the device. Figure 6–21 shows the location of the fractional PLLs supported for the high-speed differential I/O receiver and transmitter channels to generate the parallel clocks (`rx_outclock` and `tx_outclock`) and high-speed clocks (`diffioclk`).

For more information about fractional PLLs, refer to the *Clock Network and PLLs in Stratix V Devices* chapter.

Figure 6–21. Fractional PLL Locations for High-Speed Differential I/Os with DPA Locations Stratix V Devices


Left Clock Region
Right Clock Region
General Purpose I/O and High-Speed LVDS I/O with DPA and Soft CDR
Transceiver Block
Fractional PLL
FPGA Fabric
(Logic Elements, DSP, Embedded Memory, Clock Networks)
Left Clock Region
Right Clock Region

The LVDS receiver and driver channels can be driven by the center or corner fractional PLLs. The clock tree network cannot crossover for different I/O regions. For example, the top left corner fractional PLL can not crossover to drive the LVDS receiver and driver channels on the top right I/O bank when the top center fractional PLL is driving any of the LVDS receiver and driver channels on the top left I/O bank.

For more information on the fractional PLL clocking restrictions, refer to the "Differential Pin Placement Guidelines" on page 6–29.

Source-Synchronous Timing Budget

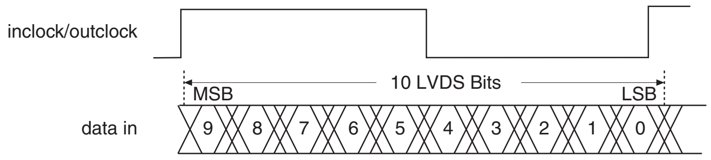
This section describes the timing budget, waveforms, and specifications for source-synchronous signaling in the Stratix V device family. LVDS I/O standards enable high-speed data transmission. This high data transmission rate results in better overall system performance. To take advantage of fast system performance, you must 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 V 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 6–22 shows the data bit orientation of the x10 mode.

Figure 6–22. Bit Orientation in the Quartus II Software



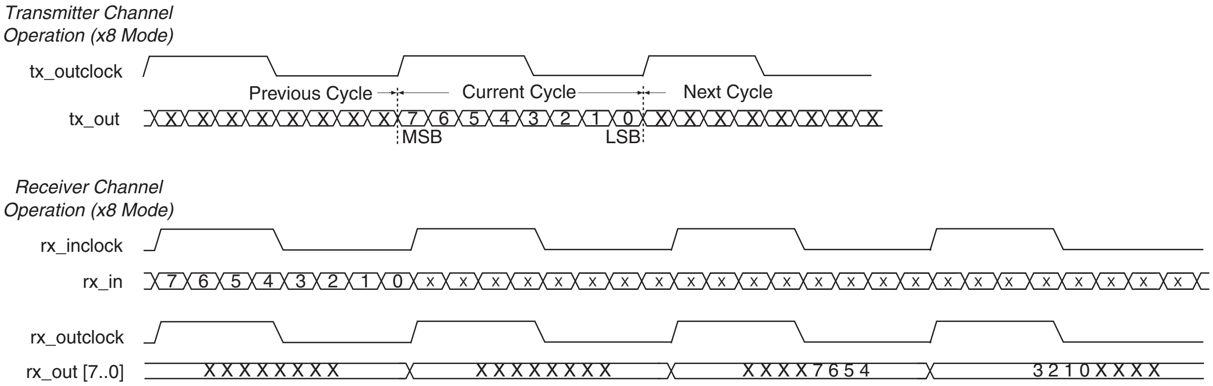
Differential I/O Bit Position

Data synchronization is necessary for successful data transmission at high frequencies. Figure 6–23 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 6–6.

Figure 6–23. Bit-Order and Word Boundary for One Differential Channel *(Note 1)*



Note to Figure 6–23:

(1) These are only functional waveforms and are not intended to convey timing information.

Table 6–6 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 6–6. 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 V 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 6–25.

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 in the TimeQuest Timing Analyzer, or from the *DC and Switching Characteristics for Stratix V Devices* chapter.

Receiver Skew Margin for Non-DPA Mode

Changes in system environment, such as temperature, media (cable, connector, or PCB), and loading affect 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 datapath. The relationship between RSKM, TCCS, and SW is expressed by the RSKM equation shown in Equation 6–1.

Equation 6–1. RSKM

$$\text{RSKM} = \frac{\text{TUI} - \text{SW} - \text{TCCS}}{2}$$

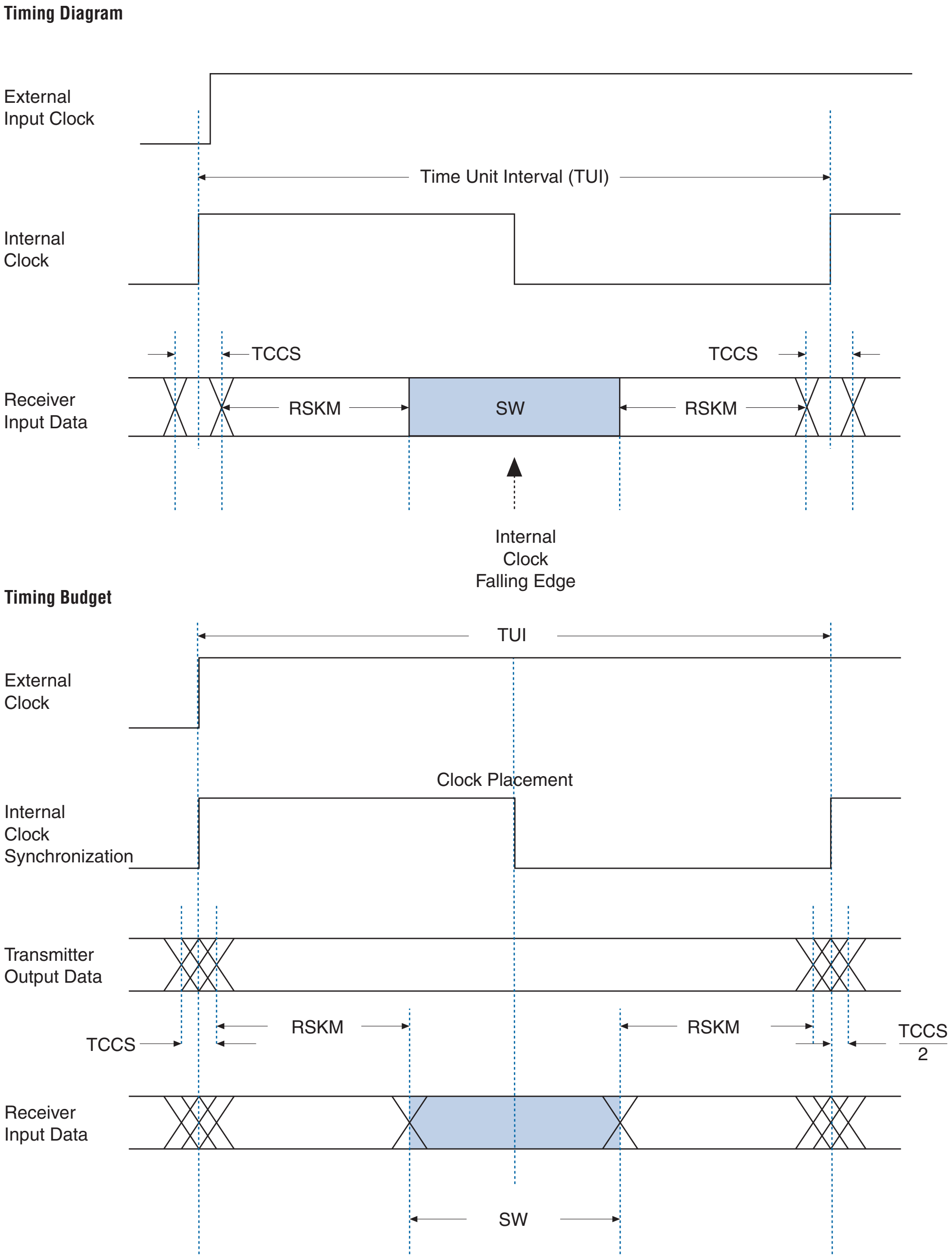
Conventions used for the equation:

- RSKM—the timing margin between the receiver's clock input and the data input sampling window.
- Time unit interval (TUI)—time period of the serial data.
- 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 6–24 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 6–24. Differential High-Speed Timing Diagram and Timing Budget for Non-DPA Mode



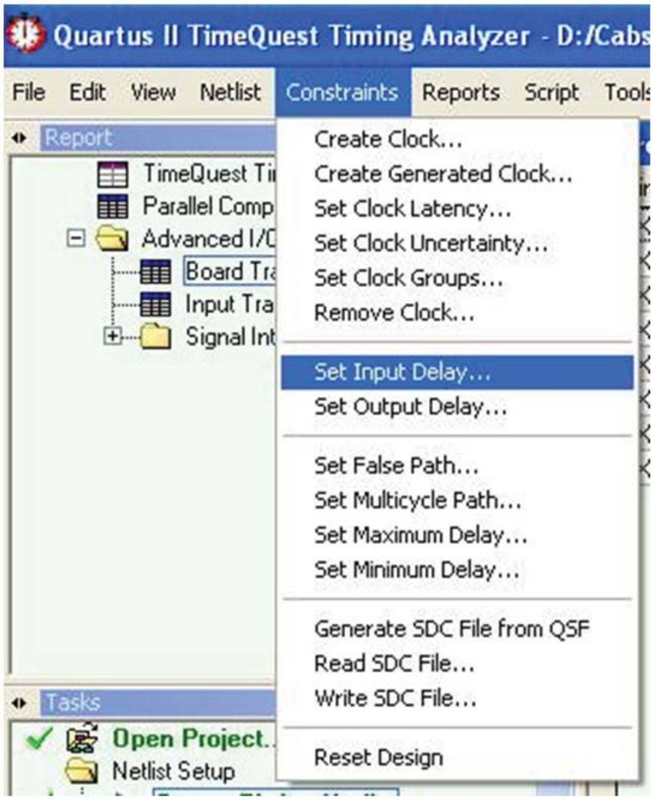
For LVDS receivers, the Quartus II software provides an RSKM report showing the SW, TUI, and RSKM values for non-DPA mode. You can generate the RSKM report by executing the `report_RSKM` command in the TimeQuest Timing Analyzer. You can find the RSKM report in the Quartus II compilation report in the TimeQuest Timing Analyzer section.

To obtain the RSKM value, assign an appropriate input delay to the LVDS receiver through the TimeQuest Timing Analyzer constraints menu.

For assigning an input delay, follow these steps:

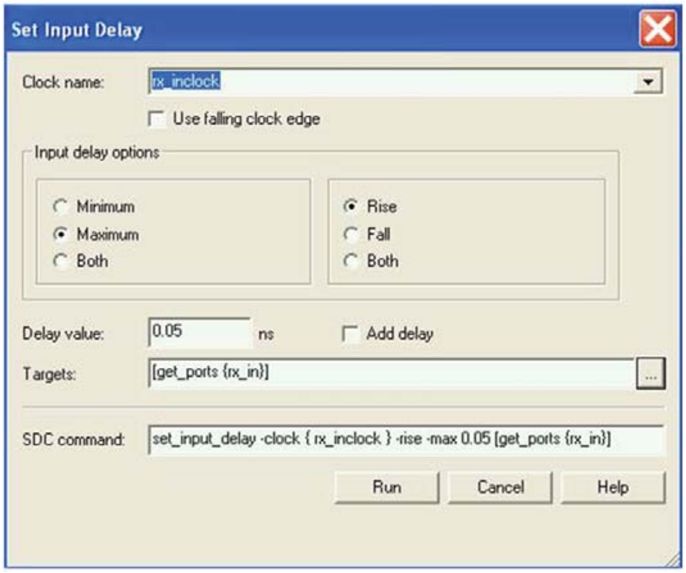
1. The Quartus II TimeQuest Timing Analyzer GUI has many options for setting the constraints and analyzing the design. Figure 6–25 shows various commands on the Constraints menu. For setting an input delay, you must select the **Set Input Delay** option.

Figure 6–25. Selection of Constraints Menu in TimeQuest Timing Analyzer



2. Figure 6–26 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 6–26. Input Time Delay Assignment Through TimeQuest Timing Analyzer



Clock name: rx_inclock

☐ Use falling clock edge

Input delay options

○ Minimum
● Maximum
○ Both

● Rise
○ Fall
○ Both

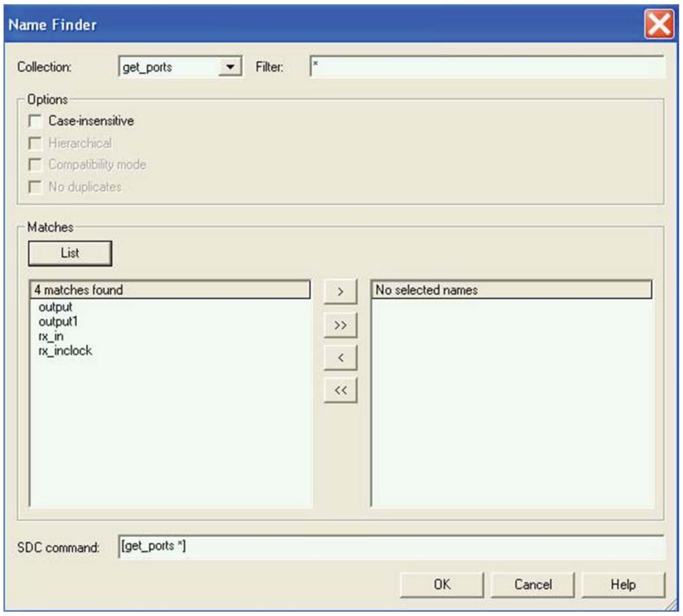
Delay value: 0.05 ns ☐ Add delay

Targets: [get_ports {rx_in}]

SDC command: set_input_delay -clock { rx_inclock } -rise -max 0.05 [get_ports {rx_in}]

3. Click the **Browse** button to the right of the **Targets** option. You can view a list of all available ports using the **List** option in the **Name Finder** window (Figure 6–27).

Figure 6–27. Name Finder Window in Set Input Delay Option



Collection: get_ports Filter: *

Options

☐ Case-insensitive
☐ Hierarchical
☐ Compatibility mode
☐ No duplicates

Matches

4 matches found
output
output1
rx_in
rx_inclock

No selected names

SDC command: [get_ports *]

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 6–1 shows the RSKM calculation.

Example 6–1. RSKM

Data Rate: 1 Gbps, Board channel-to-channel skew = **200 ps**

For Stratix V 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)/2**
= (**1000 ps — 300 ps — 300 ps)/2**
= **200 ps**

```
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 and DPA-Disabled Differential Channels" on page 6–30.

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.

The information in the following sections is preliminary.

Guidelines for DPA-Enabled and DPA-Disabled Differential Channels

The Stratix V device family has differential receivers and transmitters in all I/O blocks. 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, DPA-Disabled 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.

Double data rate I/O (DDIO) output pins can be placed within I/O modules that have the same pad group number as a SERDES differential channel but half rate DDIO (single data rate) output pins cannot be placed within I/O modules that have the same pad group number as a receiver SERDES differential channel. The input register must be implemented within the FPGA fabric logic.

The following lists two DPA restrictions:

- Because there is only a single DPA clock bus, a PLL drives a continuous series of DPA channels.
- To prevent noise mixing, use one row of separation between two groups of DPA channels. The two PLLs operate using different reference clocks (either same base frequency with PPM differences or an entirely different frequencies).

Document Revision History

Table 6–7 lists the revision history for this chapter.

Table 6–7. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Added Table 6–2 and Table 6–3. ■ Updated Table 6–1. ■ Updated Figure 6–2 and Figure 6–21. ■ Updated “Locations of the I/O Banks”, “Programmable Pre-Emphasis”, “Differential Receiver”, “Fractional PLLs and Stratix V Clocking”, and “DPA-Enabled Channels, DPA-Disabled Channels, and Single-Ended I/Os” sections. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

7. External Memory Interfaces in Stratix V Devices

SV51008-1.2

This chapter describes external memory interfaces available with Stratix® V devices, as well as the silicon capabilities of Stratix V devices to support external memory interfaces. Stratix V devices provide an efficient architecture to quickly and easily fit wide external memory interfaces to support the high level of system bandwidth 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 and DDR2 SDRAM, QDR II+ and QDR II SRAM, and RLDRAM II.

Stratix V I/O elements (IOEs) 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 support for DDR3 SDRAM interfaces, read FIFO blocks, and 4- to 36-bit programmable DQ group widths.

The high-performance memory interface solution includes the self-calibrating UniPHY megafunction, which is optimized to take advantage of the Stratix V I/O structure and the Quartus® II software TimeQuest Timing Analyzer, which provides the total solution for the highest reliable frequency of operation across process, voltage, and temperature (PVT) variations.

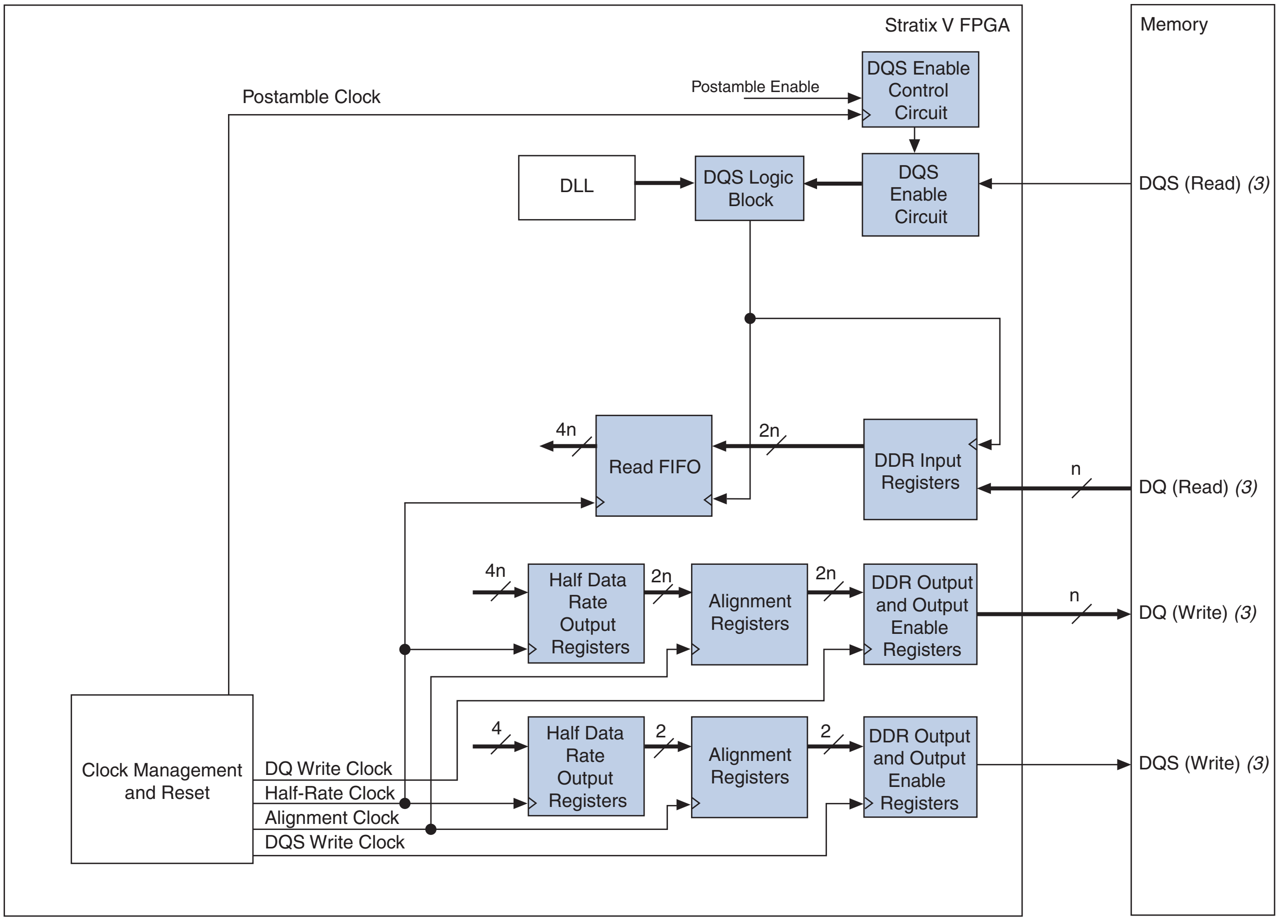
This chapter contains the following sections:

- "Memory Interface Pin Support" on page 7–3
- "Stratix V External Memory Interface Features" on page 7–8

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 datapath that uses all the Stratix V IOE features.

Figure 7–1. External Memory Interface Datapath Overview for Stratix V Devices *(Note 1)*, *(2)*



Notes to Figure 7–1:

(1) You can bypass each register block.

(2) The blocks for each memory interface may differ slightly. The shaded blocks are part of the Stratix V 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 V device features such as delay-locked loops (DLLs), dynamic OCT control, read- and write-leveling circuitry, and I/O features such as programmable I/O delay chains, read FIFO blocks, slew rate adjustment, and programmable drive strength.

For more information about I/O features, refer to the *I/O Features in Stratix V Devices* chapter.

The UniPHY megafunction instantiates a phase-locked loop (PLL) to generate related clocks for the memory interface.

For more information about the Stratix V PLL, refer to the *Clock Networks and PLLs in Stratix V Devices* chapter. For more information about the UniPHY megafunction, refer to *Volume 3: Implementing Altera Memory Interface IP* of the *External Memory Interface Handbook*.

Memory Interface Pin Support

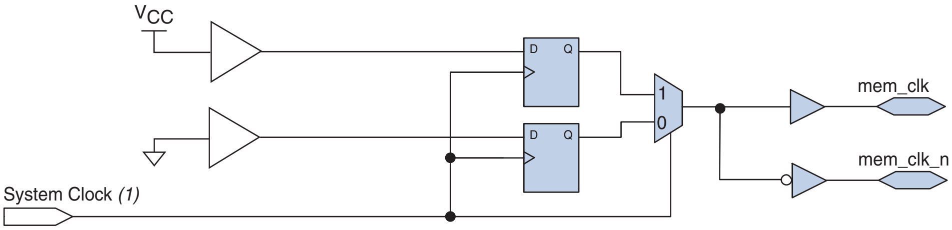
A typical memory interface requires data (D, Q, or DQ), data strobe (DQS/CQ/QK and DQSn/CQn, QK#), address, command, and clock pins. Some memory interfaces use data mask (DM, BWSn, or NWSn) pins to enable write masking and QVLD pins to indicate that the read data is ready to be captured. This section describes how Stratix V devices support all these different pins.

DDR3 and DDR2 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 with Stratix V DDR I/O (DDRIO) registers to ensure that the timing relationships between the CK/CK# and DQS signals (t_{DQSS}, t_{DSS}, and t_{DSH} in DDR3 and DDR2 SDRAM devices or t_{CKDK} in RLDRAM II devices) are met. QDR II+ and QDR II SRAM devices use the same clock (K/K#) to capture write data, address, and command signals.

Memory clock pins in Stratix V devices are generated with a DDRIO 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 and pin location requirements, refer to the *Section I: Device and Pin Planning* in volume 2 of the *External Memory Interface Handbook*.

Figure 7–2. Memory Clock Generation ***(Note 1)***



Notes to Figure 7–2:

(1) To minimize jitter, a dedicated clock network is used for memory output clock generation.

Stratix V devices offer differential input buffers for differential read-data strobe and clock operations. In addition, Stratix V devices also provide an independent DQS logic block for each CQn pin for complementary read-data strobe and clock operations. In the Stratix V pin tables, the differential data strobe/clock pin pairs are denoted as DQS and DQSn pins, while the complementary echo clock signals are denoted as CQ and CQn pins. DQSn and CQn pins are marked separately in the pin tables. Each CQn pin connects to a DQS logic block and the phase-shifted CQn signals go to the negative-half cycle 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 and DDR2 SDRAM, and RLDRAM II common I/O interfaces, or unidirectional signals, as in QDR II+ and QDR II SRAM, and RLDRAM II separate I/O devices. Connect the unidirectional read-data signals to Stratix V DQ pins and the unidirectional write-data signals to a different DQ/DQS group than the read DQ/DQS group. You must assign the write clocks to the DQS/DQSn pins associated to this write DQ/DQS group. Do not use the CQ/CQn pin-pair for write clocks.

☞ Using a DQ/DQS 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 DQ and DQS pin locations are fixed in the pin table. Memory interface circuitry is available in every Stratix V I/O bank that does not support transceivers. All the memory interface pins support the I/O standards required to support DDR3 and DDR2 SDRAM, QDR II+ and QDR II SRAM, and RLDRAM II devices.

Stratix V devices support DQ and DQS signals with DQ bus modes of x4, x8/x9, x16/x18, or x32/x36. If any of these pins are not used for memory interfacing, you can use these pins as user I/Os. In addition, you can use the DQSn or CQn pins that are not used for clocking as DQ pins.

Table 7–1 lists pin support per DQ/DQS bus mode, including the DQS/CQ and DQSn/CQn pin pair.

Table 7–1. DQ/DQS Bus Mode Pins for Stratix V Devices

Mode	DQSn Support	CQn Support	Parity or Data Mask (Optional)	QVLD (Optional) (1)	Typical Number of Data Pins per Group	Maximum Number of Data Pins per Group (2)
x4	Yes	No	No (3)	No	4	5
x8/x9 (4)	Yes	Yes	Yes	Yes	8 or 9	11
x16/x18 (5)	Yes	Yes	Yes	Yes	16 or 18	23
x32/x36 (6)	Yes	Yes	Yes	Yes	32 or 36	47

Notes to Table 7–1:

(1) The QVLD pin is not used in the UniPHY 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. If you use differential or complementary DQS signaling, the maximum number of data per group decreases by one. This number may vary per DQ/DQS group in a particular device. Check the pin table for the exact number per group. For DDR3 and DDR2 interfaces, the number of pins is further reduced for an interface larger than x8 because you require one DQS pin for each x8/x9 group to form the x16/x18 and x32/x36 groups.

(3) If you do not use differential DQS and the group does not have additional signals, the DM pin is supported.

(4) Two x4 DQ/DQS groups are stitched to create a x8/x9 group, so there are a total of 12 pins in this group.

(5) Four x4 DQ/DQS groups are stitched to create a x16/x18 group; so there are a total of 24 pins in this group.

(6) Eight x4 DQ/DQS groups are stitched to create a x32/x36 group, so there are a total of 48 pins in this group.

Table 7–2 lists the number of DQ/DQS groups available per side in each Stratix V device.

Table 7–2. Number of DQ/DQS Groups in Stratix V Devices per Side *(Note 1)*—Preliminary (Part 1 of 2)

Device	Package	Side	x4 *(2)*	x8/x9	x16/x18	x32/x36
5SGXA3 5SGXA4	780-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	26	13	6	1
		Top	18	9	4	0
	1152-pin FineLine BGA (with 24 transceivers)	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	42	21	10	3
	1152-pin FineLine BGA (with 36 transceivers)	Left/Right	0	0	0	0
		Bottom	36	18	8	2
		Top	36	18	8	2
	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	52	26	12	4
		Top	52	26	12	4
5SGXA5 5SGXA7	1152-pin FineLine BGA (with 24 transceivers)	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	42	21	10	3
	1152-pin FineLine BGA (with 36 transceivers)	Left/Right	0	0	0	0
		Bottom	38	19	8	2
		Top	36	18	8	2
	1517-pin FineLine BGA (with 36 transceivers)	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
	1517-pin FineLine BGA (with 48 transceivers)	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	50	25	12	5
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	70	35	16	6
		Top	70	35	16	6
5SGXA9 5SGXAB	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	70	35	16	6
		Top	70	35	16	6

Table 7–2. Number of DQ/DQS Groups in Stratix V Devices per Side *(Note 1)*—Preliminary (Part 2 of 2)

Device	Package	Side	x4 *(2)*	x8/x9	x16/x18	x32/x36
5SGXB5 5SGXB6	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	36	18	8	3
		Top	36	18	8	3
	1760-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	50	25	11	4
		Top	50	25	11	4
5SGSD2 5SGSD3	484-pin FineLine BGA	Left/Right	*(3)*	*(3)*	*(3)*	*(3)*
		Bottom	*(3)*	*(3)*	*(3)*	*(3)*
		Top	*(3)*	*(3)*	*(3)*	*(3)*
	780-pin FineLine BGA	Left/Right	*(3)*	*(3)*	*(3)*	*(3)*
		Bottom	*(3)*	*(3)*	*(3)*	*(3)*
		Top	*(3)*	*(3)*	*(3)*	*(3)*
5SGSD3	1152-pin FineLine BGA	Left/Right	*(3)*	*(3)*	*(3)*	*(3)*
		Bottom	*(3)*	*(3)*	*(3)*	*(3)*
		Top	*(3)*	*(3)*	*(3)*	*(3)*
5SGSD4 5SGSD5	1152-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	42	21	10	3
	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
5SGSD6 5SGSD8	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	46	23	10	3
		Top	40	20	10	4
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	62	31	15	6
		Top	66	33	16	6
5SGTC5 5SGTC7	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	50	25	12	4
		Top	50	25	12	5
5SEE9 5SEEB	1517-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	58	29	14	6
		Top	58	29	14	6
	1932-pin FineLine BGA	Left/Right	0	0	0	0
		Bottom	70	35	16	6
		Top	70	35	16	6

Notes to Table 7–2:

(1) These numbers are preliminary until the devices are available.

(2) Some of the x4 groups are using `RZQ` pins. If you use the Stratix V calibrated OCT feature, you cannot use these groups.

(3) The number is still not available.

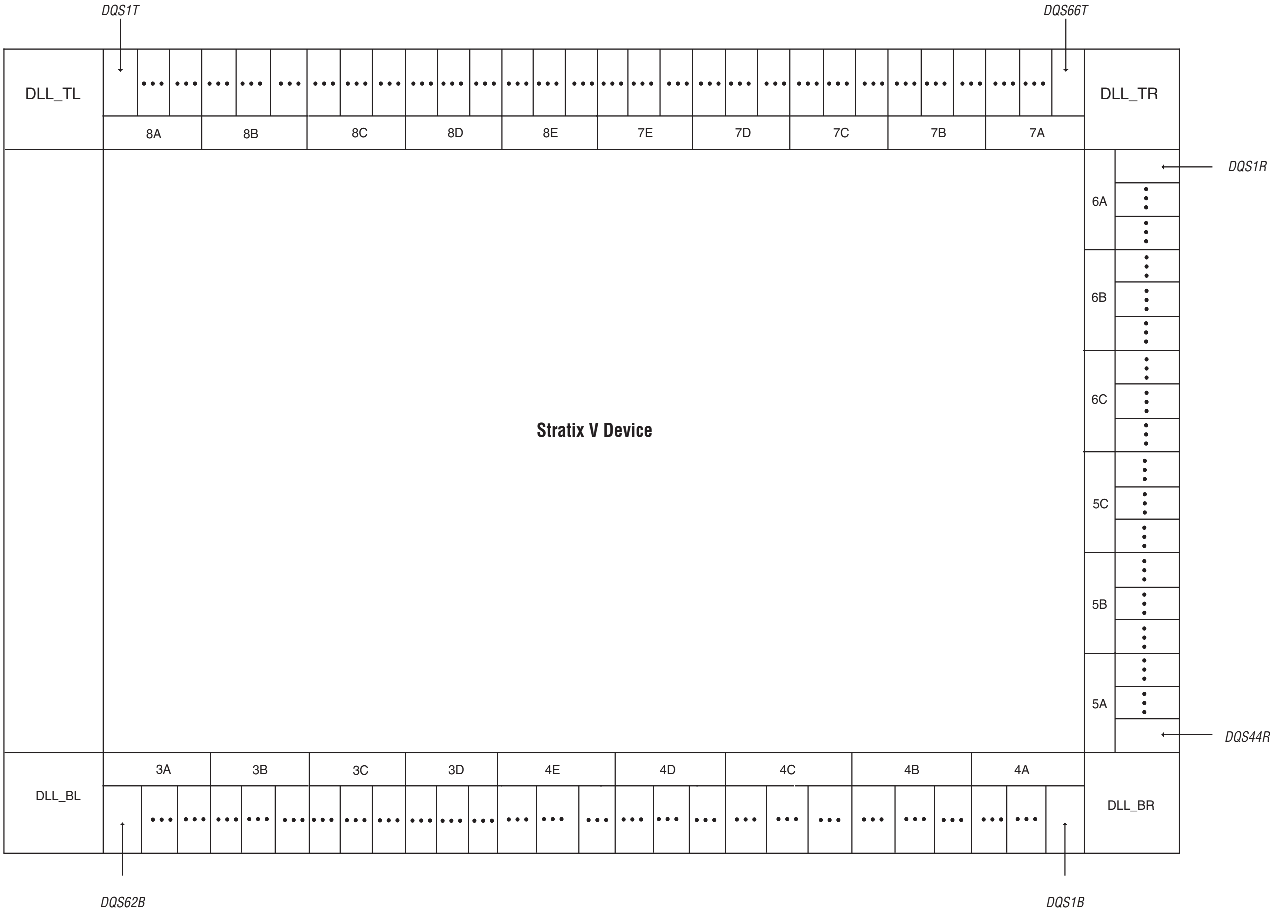
The DQS and DQSn pins are listed in the Stratix V pin tables as `DQSXY` and `DQSnXY`, respectively, where `X` indicates the DQ/DQS grouping number and `Y` indicates whether the group is located on the top (T), bottom (B), left (L), or right (R) side of the device. The DQ/DQS pin numbering is based on x4 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), or right (R) side of the device. For example, `DQS1T` indicates a DQS pin located on the top side of the device. The DQ pins belonging to that group are shown as `DQ1T` in the pin table. For more information, refer to Figure 7–3.

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 corner of the device going counter clockwise in a die-top view. Figure 7–3 shows how the DQ/DQS groups are numbered in a die-top view of the device.

Figure 7–3. DQS Pins in Stratix V I/O Banks



Using the RZQ Pins in a DQ/DQS Group for Memory Interfaces

You can use the DQS/DQSn pins in some of the x4 groups as RZQ pins (listed in the pin table). You cannot use a x4 DQ/DQS group for memory interfaces if any of its DQ pin members are used as RZQ pins for OCT calibration.

There is no restriction on using x8/x9, x16/x18, or x32/x36 DQ/DQS groups that include the x4 groups whose pins are used as RZQ pins because there are enough extra pins that you can use as DQS pins.

☞ You must manually assign DQ and DQS pins for x8, x16/x18, or x32/x36 DQ/DQS groups whose members are used as RZQ pins. The Quartus II software might not be able to place DQ and DQS pins without manual pin assignments, resulting in a “no-fit”.

Stratix V External Memory Interface Features

Stratix V devices have features that allow robust high-performance external memory interfacing. The UniPHY 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 V 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 UniPHY 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 V devices have built-in circuitries in the IOE to convert data from full rate (the I/O frequency) to half rate (the controller frequency) and vice versa. If you use the Altera memory controller MegaCore functions, the UniPHY megafunction is instantiated for you.

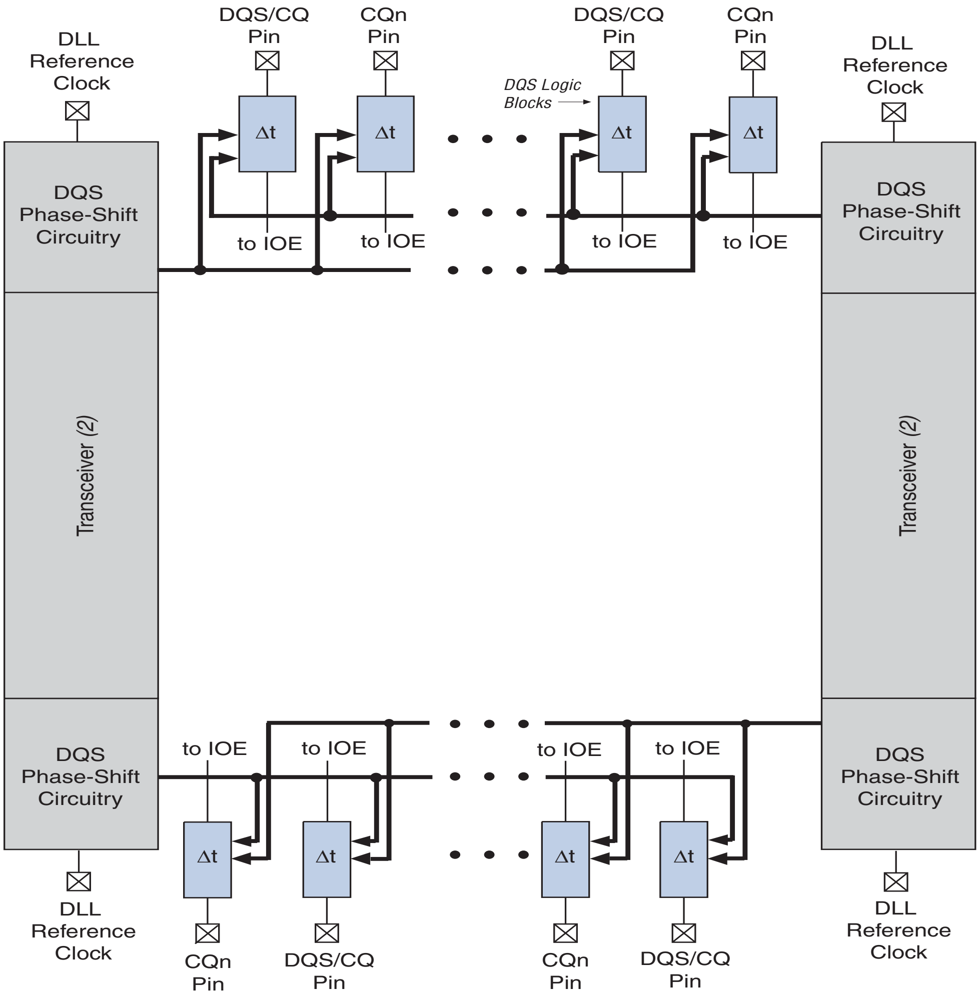
For more information about the UniPHY megafunction, refer to the *Volume 3: Implementing Altera Memory Interface IP* of the *External Memory Interface Handbook*.

DQS Phase-Shift Circuitry

Stratix V phase-shift circuitry provides phase shift to the DQS/CQ and CQn pins on read transactions if the DQS/CQ and CQn pins are acting as input clocks or strobes to the FPGA. 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 7–4 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 V device. For possible reference input clock pins for each DLL, refer to "Delay-Locked Loop" on page 7–10.

Figure 7–4. DQS/CQ and CQn Pins and DQS Phase-Shift Circuitry ***(Note 1)***



Note to Figure 7–4:

(1) You can configure each DQS/CQ and CQn pin with a phase shift based on one of two possible DLL output settings.

(2) This transceiver block is applicable to all Stratix V devices except for Stratix V E devices.

DQS phase-shift circuitry is connected to the DQS logic blocks that control each DQS/CQ or CQn pin. DQS logic blocks allow the DQS delay settings to be updated concurrently at every DQS/CQ or CQn pin.

Delay-Locked Loop

DQS phase-shift circuitry uses a DLL to dynamically control the clock delay required by the DQS/CQ and CQn pin. In turn, the DLL 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 if the DLL updates the settings. If the DLL is in low jitter mode, the phase-shift circuitry requires 2,560 clock cycles to lock and calculate the correct input clock period. Otherwise, only 512 clock cycles are required. Altera recommends not sending 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 with 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 IOE must be able to capture the data in low-frequency applications, where a large amount of timing margin is available.

There are a maximum of four DLLs in a Stratix V 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 V device to have eight different DLL phase shift settings.

The DLL can access the two adjacent sides from its location within the device, if the device has I/O banks on the side. For example, DLL_TR on the top right of the device can access the top side (I/O banks 7A, 7B, 7C, 7D, 7E, 8A, 8B, 8C, 8D and 8E) and the right side of the device (I/O banks 5A, 5B, 5C, 6A, 6B, and 6C), if the I/O banks are available on the device. 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 adjacent DLLs. For example, DQS1R can get its phase-shift settings from DLL_TR, while DQS2R can get its phase-shift settings from DLL_BR.

☞ If you are using leveling delay chains, you can only have one memory interface in each I/O sub-bank (such as I/O sub-banks 5A, 5B, and 5C). This is because there is only one leveling delay chain per I/O sub-bank.

The reference clock for each DLL may come from the PLL output clocks or clock input pins. Table 7–3 through Table 7–6 list the available DLL reference clock input resources for Stratix V devices.

☞ If 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 automatically changes it). Because the PLL does not use any other outputs, it does not have to compensate for any clock paths.

Table 7–3. DLL Reference Clock Input for 5SGXA3 and 5SGXA4 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X70_Y76 CEN_X70_Y31	LR_X0_Y93 LR_X0_Y89	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X70_Y76 CEN_X70_Y71	LR_X152_Y93 LR_X152_Y89	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X0_Y7 CEN_X0_Y2	LR_X152_Y15 LR_X152_Y11	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X0_Y7 CEN_X0_Y2	LR_X0_Y15 LR_X0_Y11	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–4. DLL Reference Clock Input for 5SGXB5 and 5SGXB6 Devices (Part 1 of 2)

DLL	PLL		CLKIN		
	Center	Left/Right	Left	Center	Right
DLL_TL	CEN_X74_Y136 CEN_X74_Y131	LR_X0_Y135 LR_X0_Y131	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X74_Y136 CEN_X74_Y131	LR_X163_Y135 LR_X163_Y131	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P

Table 7–4. DLL Reference Clock Input for 5SGXB5 and 5SGXB6 Devices (Part 2 of 2)

DLL	PLL		CLKIN		
	Center	Left/Right	Left	Center	Right
DLL_BR	CEN_X74_Y7 CEN_X74_Y2	LR_X163_Y25 LR_X163_Y21	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X74_Y7 CEN_X74_Y2	LR_X0_Y25 LR_X0_Y21	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–5. DLL Reference Clock Input for 5SGXA5, 5SGXA7, 5SGTC5, and 5SGTC7 Devices

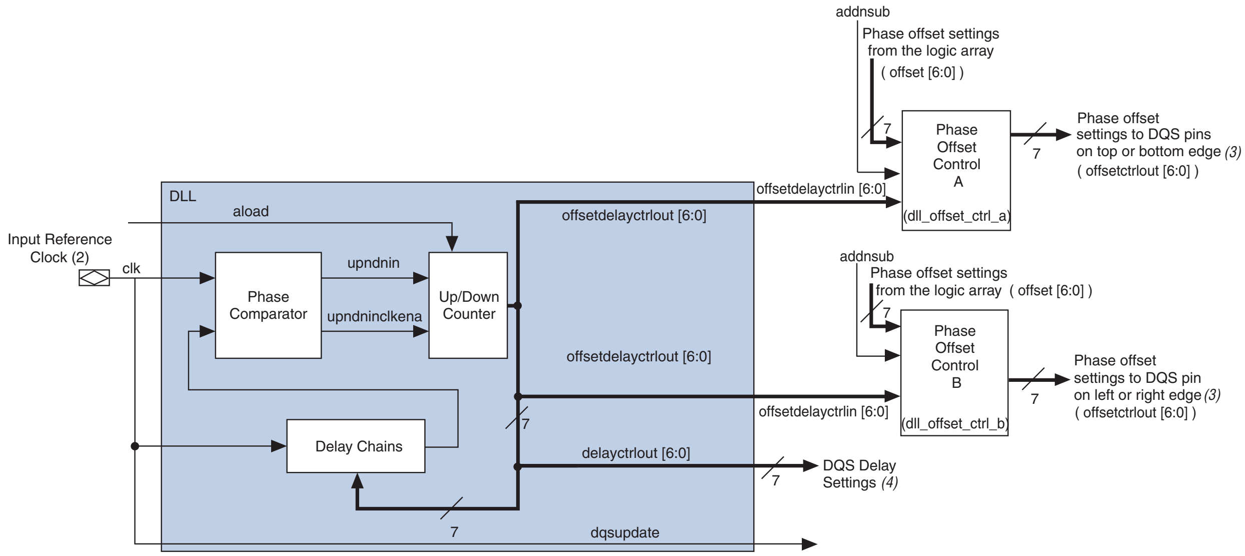
DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X94_Y118 CEN_X94_Y113	LR_X0_Y135 LR_X0_Y131	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X94_Y118 CEN_X94_Y113	LR_X201_Y135 LR_X201_Y131	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P
DLL_BR	CEN_X94_Y7 CEN_X94_Y2	LR_X201_Y15 LR_X201_Y11	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P
DLL_BL	CEN_X94_Y7 CEN_X94_Y2	LR_X0_Y15 LR_X0_Y11	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Table 7–6. DLL Reference Clock Input for 5SGSD6 and 5SGSD8 Devices

DLL	PLL			CLKIN		
	Center	Corner	Right	Left	Center	Right
DLL_TL	CEN_X103_Y124 CEN_X103_Y119	LR_X0_Y141 LR_X0_Y137	—	CLK20P CLK21P CLK22P CLK23P	CLK16P CLK17P CLK18P CLK19P	—
DLL_TR	CEN_X103_Y124 CEN_X103_Y119	LR_X218_Y131 LR_X218_Y127	LR_X218_Y75 LR_X218_Y71	—	CLK16P CLK17P CLK18P CLK19P	CLK12P CLK13P CLK14P CLK15P CLK24P CLK25P CLK26P CLK27P
DLL_BR	CEN_X103_Y7 CEN_X103_Y2	LR_X218_Y17 LR_X218_Y13	LR_X218_Y75 LR_X218_Y71	—	CLK4P CLK5P CLK6P CLK7P	CLK8P CLK9P CLK10P CLK11P CLK24P CLK25P CLK26P CLK27P
DLL_BL	CEN_X103_Y7 CEN_X103_Y2	LR_X0_Y15 LR_X0_Y11	—	CLK0P CLK1P CLK2P CLK3P	CLK4P CLK5P CLK6P CLK7P	—

Figure 7–5 shows a simple block diagram of the DLL. The input reference clock goes into the DLL to a chain of up to eight 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 seven-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 7–5. Simplified Diagram of the DQS Phase-Shift Circuitry ***(Note 1)***



Notes to Figure 7–5:

(1) All features of the DQS phase-shift circuitry are accessible from the UniPHY 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 7–3 through Table 7–6.

(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 read FIFO block.

In the Quartus II assignment, 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`.

You can reset the DLL from either the logic array or a user I/O pin (when 2,560 or 512 clock cycles applies). Each time the DLL is reset, you must wait for 2,560 (low-jitter mode) or 512 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°, 45°, 90°, 135°, 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 45° phase shift on `DQS2T`, referenced from a 200-MHz clock. However, 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 45° (up to 180°).

The seven-bit DQS delay settings from the DLL vary with PVT to implement the phase-shift delay.

For the frequency range of each mode, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

For a 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 if a 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 QDR II+ and QDR II SRAM interfaces.

Phase Offset Control

Each DLL has two phase-offset modules and can provide two separate DQS delay settings with independent offsets, 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 must 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 a 45° phase shift, but your interface must have a 97.5° phase shift on the DQS signal, you can use two delay chains in the DQS logic blocks to give you a 90° phase shift and use the phase offset control feature to implement the extra 7.5° phase shift.

You can use either a static phase offset or a dynamic phase offset to implement the additional phase shift. The available additional phase shift is implemented in 2's complement in Gray-code between settings –128 to +127. An additional bit indicates whether the setting has a positive or negative value. The settings are linear, each phase offset setting adds a specified delay amount. The DQS phase shift is the sum of the DLL delay settings and the user-selected phase offset settings whose top setting is 128, so the actual physical offset setting range is 128 subtracted by the DQS delay settings from the DLL.

☞ When using this feature, you must monitor the DQS delay settings to know how many offsets you can add and subtract in the system. 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 required to achieve a 30° phase shift, you can subtract up to 28 phase offset settings and you can add up to 99 phase offset settings to achieve the optimal delay that you require.

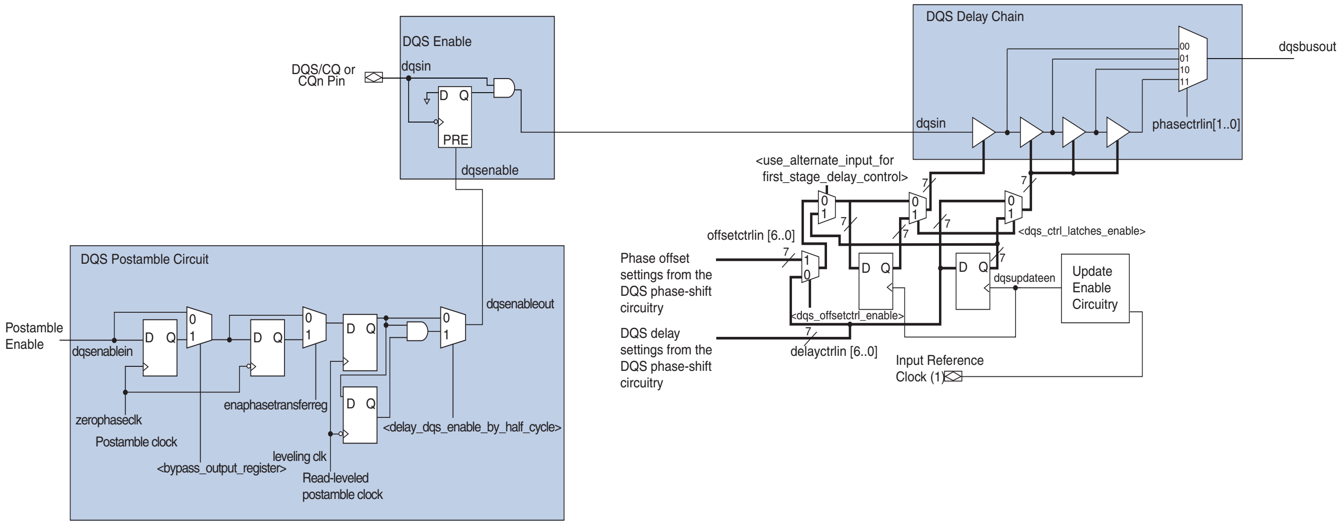
For more information about the value for each step and the specified delay amounts for the phase offset setting, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

When using static phase offset, you can specify the phase offset amount in the UniPHY 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 add or subtract, you can dynamically input the phase offset amount into the `offset[6..0]` port. When you want to both add and subtract dynamically, you control the `addnsub` signal in addition to the `offset[6..0]` signals. The phase offset is not PVT-compensated.

DQS Logic Block

Each DQS/CQ and CQn pin is connected to a separate DQS logic block, which consists of the DQS delay chains, update enable circuitry, and DQS postamble circuitry (Figure 7–6).

Figure 7–6. Stratix V DQS Logic Block



Notes to Figure 7–6:

(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 7–3 on page 7–11 through Table 7–6 on page 7–13.

(2) The `dqsenable` signal can also come from the Stratix V 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 is shifted either 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 UniPHY 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.

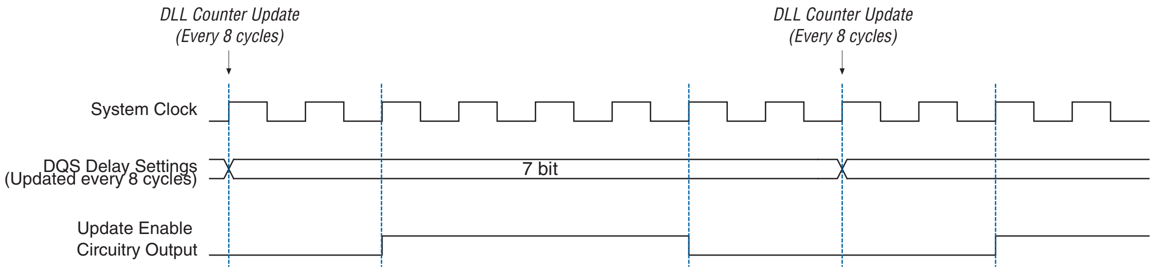
The delay elements in the DQS logic block have the same characteristics as the delay elements in the DLL. When the DLL is not used to control the DQS delay chains, you can input your own Gray-coded 7-bit settings using the `delayctrlin[6..0]` signals available in the UniPHY megafunction. These settings control 1, 2, 3, or all 4 delay elements in the DQS delay chains. The UniPHY megafunction can also dynamically choose the number of DQS delay chains required for the system. The amount of delay is equal to the sum of the intrinsic delay of the delay element 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 a 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 UniPHY megafunction uses this circuit by default. Figure 7–7 shows an example waveform of the update enable circuitry output.

Figure 7–7. DQS Update Enable Waveform



DQS Postamble Circuitry

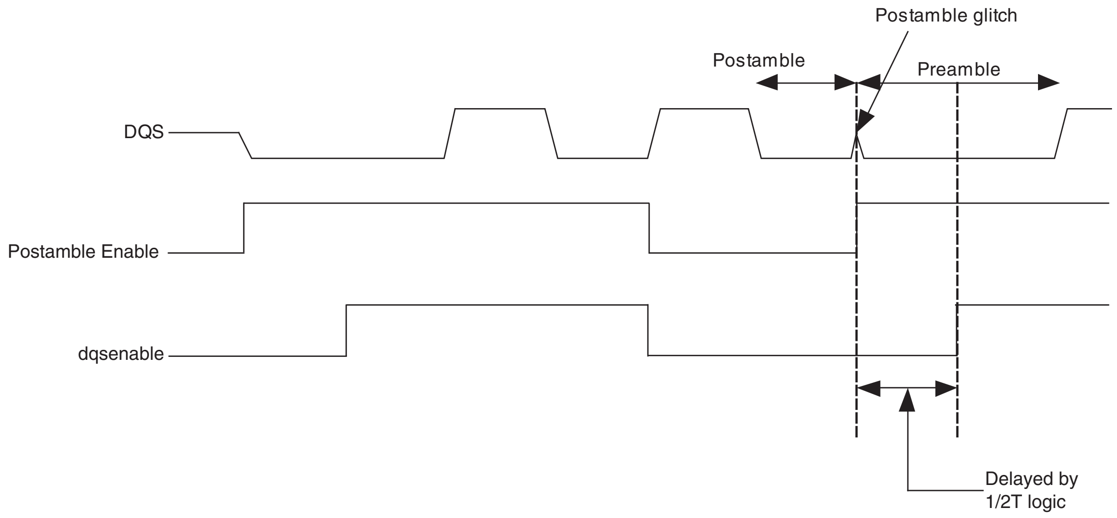
For external memory interfaces that use a bidirectional read strobe (DDR3 and DDR2 SDRAM), the DQS signal is low before going to or coming from a high-impedance state. The state in which DQS is low, just after a high-impedance state, is called the preamble; the state in which DQS is low, just before 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 and DDR2 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 DQS is in a postamble state.

Stratix V 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 DQS is in a postamble state do not affect the DQ IOE registers.

In addition to the dedicated postamble register, Stratix V 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. There is an AND gate after the postamble register outputs to avoid postamble glitches from a previous read burst on a non-consecutive read burst. This scheme allows half-a-clock cycle latency for `dqsenable` assertion and zero latency for `dqsenable` deassertion (refer to Figure 7–8).

Figure 7–8. Avoiding Glitch on a Non-Consecutive Read Burst Waveform

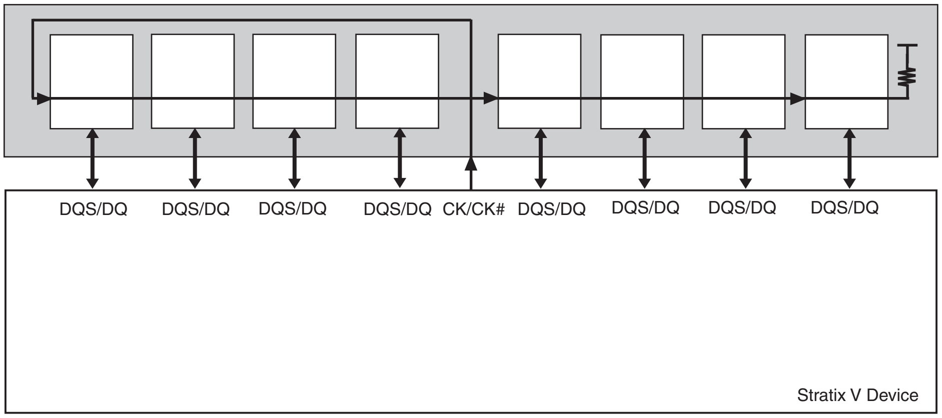


Leveling Circuitry

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 7–9 shows the clock topology in DDR3 SDRAM unbuffered modules.

Figure 7–9. DDR3 SDRAM Unbuffered Module Clock Topology



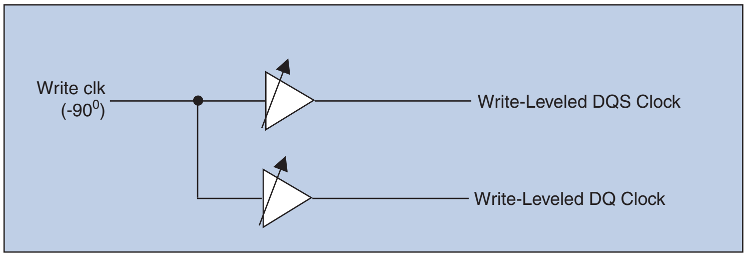
Because the data and read strobe signals are still point-to-point, take special care to ensure that the timing relationship between the CK/CK# and DQS signals (tDQSS, tDSS, and tDSH) during a write is met at every device on the modules. In a similar way, read data coming back into the FPGA from the memory is also staggered.

Stratix V FPGAs have leveling circuitry to address these two situations. There is one leveling circuit 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.

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 feed a multiplexer controlled by the UniPHY megafunction to select which clock phases are to be used for that x4 or x 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 7–10 shows the Stratix V write-leveling circuitry.

Figure 7–10. Write-Leveling Delay Chains and Multiplexers for Stratix V Devices *(Note 1)*



Note to Figure 7–10:

(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 UniPHY megafunction feeds the write-leveling circuitry to produce the clock to generate the DQS and DQ signals. During initialization, the UniPHY megafunction picks the correct write-leveled clock for the DQS and DQ clocks for each DQ/DQS 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.

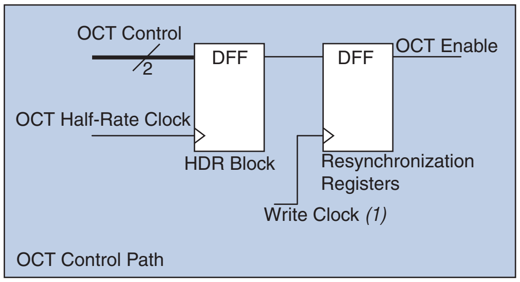
The UniPHY megafunction dynamically calibrates the alignment for read and write leveling during the initialization process.

For more information about the UniPHY megafunction, refer to *Volume 3: Implementing Altera Memory Interface IP* of the *External Memory Interface Handbook*.

Dynamic On-Chip Termination Control

Figure 7–11 shows the dynamic OCT control block.

Figure 7–11. Dynamic OCT Control Block for Stratix V Devices



Note to Figure 7–11:

(1) The write clock comes from either the PLL or the write-leveling delay chain.

The block includes all the registers required to dynamically turn on-chip parallel termination (R_T OCT) on during a read and turn R_T OCT off during a write.

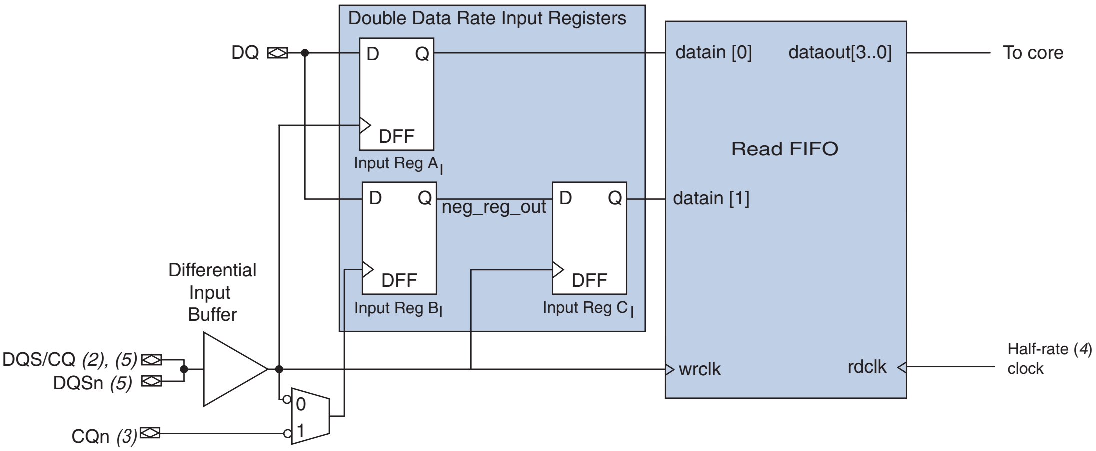
For more information about dynamic OCT control, refer to the *I/O Features in Stratix V Devices* chapter.

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, bottom, and right IOEs have the same capability.

Figure 7–12 shows the registers available in the Stratix V input path. The input path consists of the DDR input registers and the read FIFO block. You can bypass each block of the input path.

Figure 7–12. IOE Input Registers for Stratix V Devices *(Note 1)*



Notes to Figure 7–12:

(1) You can bypass the register or read FIFO block in this path.
(2) The input clock can be from the DQS logic block or from a global clock line.
(3) This input clock comes from the CQn logic block.
(4) This half rate-read clock comes from a PLL through the clock network.
(5) The DQS and DQSn signals must be inverted for DDR3 and DDR2 SDRAM interfaces. When using Altera's memory interface IPs, the DQS and DQSn signals are automatically inverted.

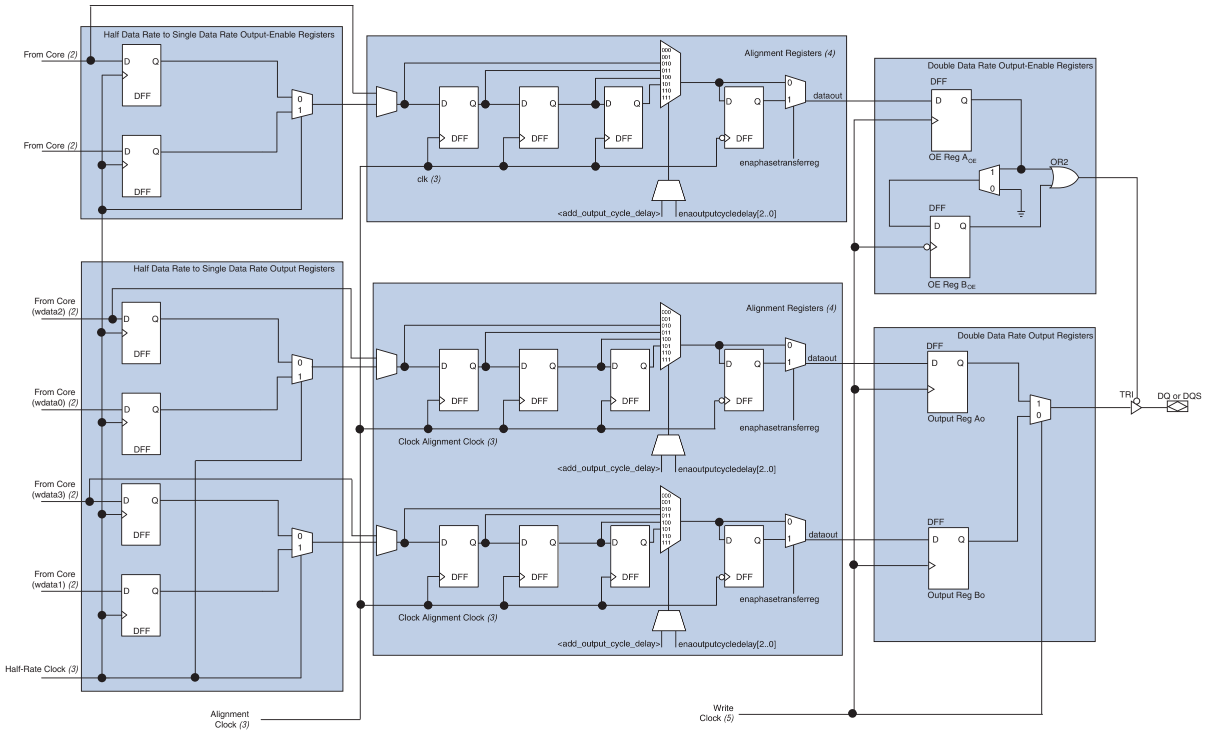
There are three registers in the DDR input registers block. Two registers capture data on the positive and negative edges of the clock while the third register aligns the captured data. You can choose to use the same clock for the positive and negative edge registers or two complementary clocks (DQS/CQ 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 read FIFO block resynchronizes the data to the system clock domain, as well as to lower the data rate to half rate.

For more information about the read-leveling delay chain, refer to "Leveling Circuitry" on page 7–20.

Figure 7–13 shows the registers available in the Stratix V output and output-enable paths. The path is divided into the HDR block, alignment registers, and output and output-enable registers. The device can bypass each block of the output and output-enable path.

Figure 7–13. IOE Output and Output-Enable Path Registers for Stratix V Devices ***(Note 1)***



Notes to Figure 7–13:

(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 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 single data rate (SDR) outputs and full-rate or half-rate DDR outputs from the FPGA core. Half-rate data is converted to full-rate with 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.

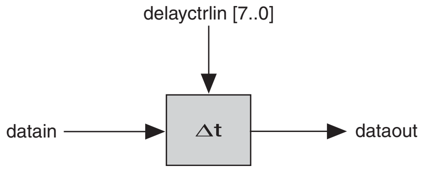
For more information about the write-leveling delay chain, refer to "Leveling Circuitry" on page 7–20.

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 output-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 V 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 7–14 shows the delay chain ports.

Figure 7–14. Delay Chain

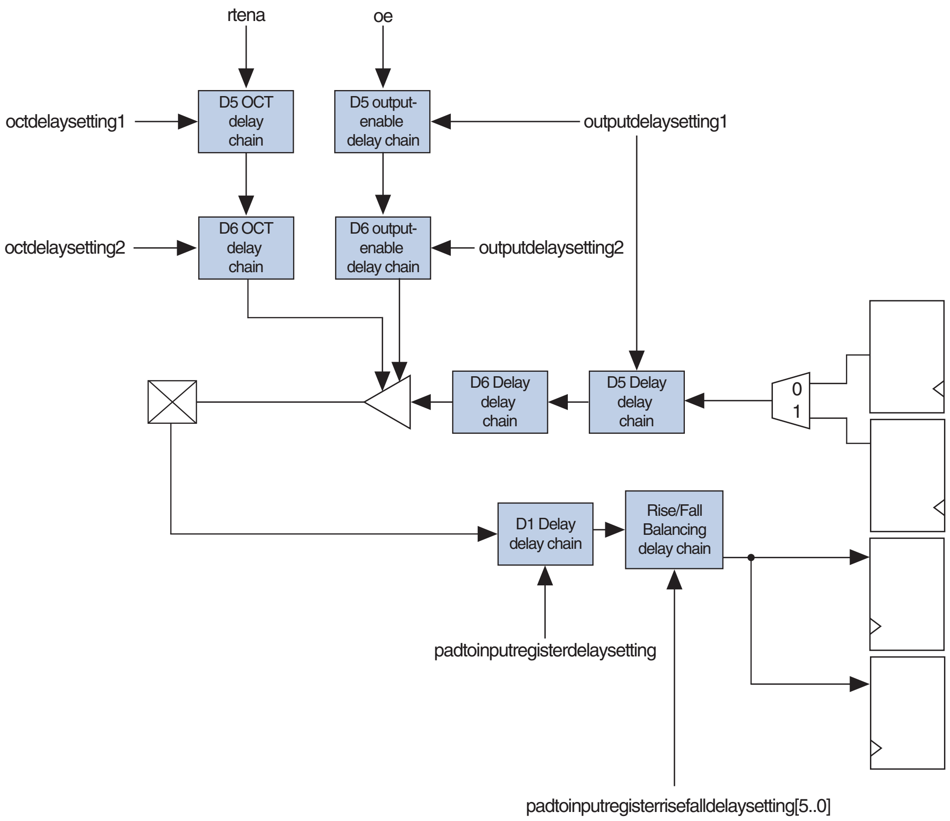


Every I/O block contains the following:

- Two delay chains in series between the output registers and output buffer
- Two delay chains between the input buffer and input register
- Two delay chains between the output enable and output buffer
- Two delay chains between the R_T OCT enable-control register and output buffer

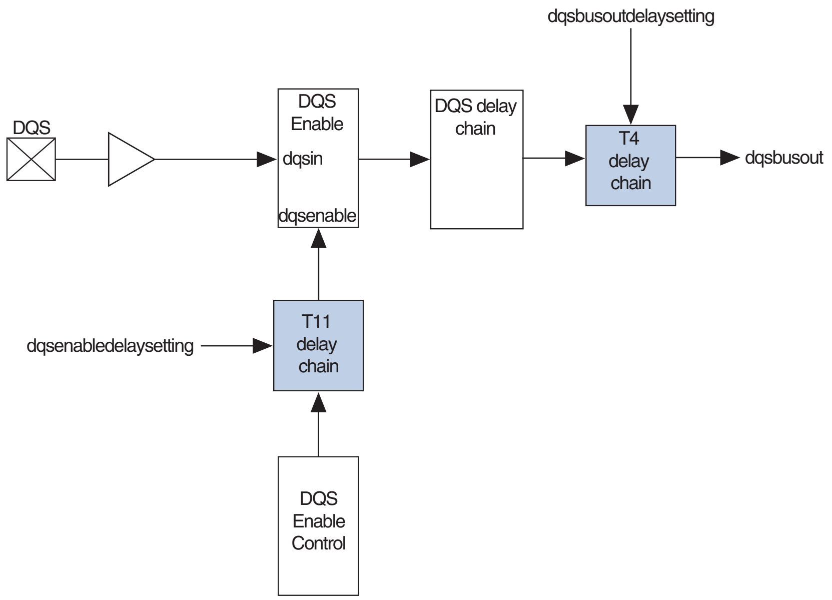
Figure 7–15 shows the delay chains in an I/O block.

Figure 7–15. 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 7–16 shows the delay chains in the DQS input path.

Figure 7–16. 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 7–17 shows the I/O configuration block and the DQS configuration block circuitry.

Figure 7–17. I/O Configuration Block and DQS Configuration Block

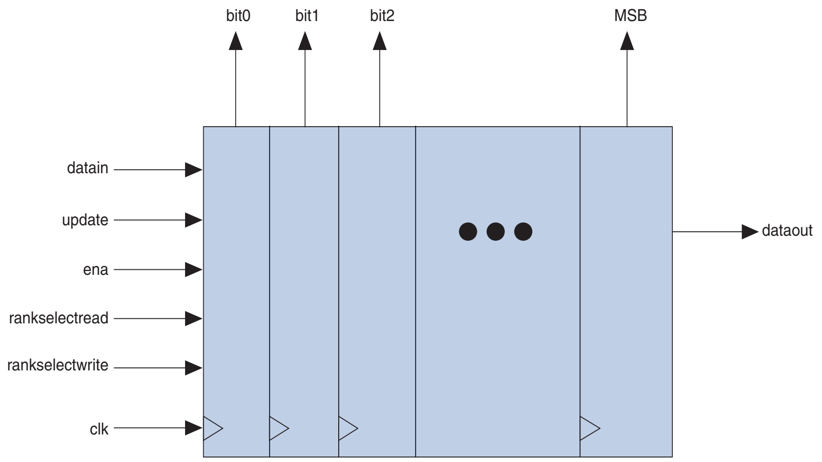


Table 7–7 lists the I/O configuration block bit sequence.

Table 7–7. I/O Configuration Block Bit Sequence

Bit	Bit Name
0..5	padtoinputregisterdelaysetting[0..5]
6..11	padtoinputregisterrisefalldelaysetting[0..5]
12..17	outputdelaysetting1[0..5]
18..23	outputdelaysetting2[0..5]
24..25	inputclkndelaysetting[0..1]
26..27	inputclkdelaysetting[0..1]
28	dutycycledelaymode
29..32	dutycycledelaysetting[0..3]

Table 7–8 lists the DQS configuration block bit sequence.

Table 7–8. DQS Configuration Block Bit Sequence (Part 1 of 2)

Bit	Bit Name
0..5	dqsbusoutdelaysetting[0..5]
6..11	dqsbusoutdelaysetting2[0..5]
12..17	octdelaysetting1[0..5]
18..23	octdelaysetting2[0..5]
24..25	addrphasesetting[0..1]
26	addrpowerdown
27	addrphaseinvert
28..29	dqsoutputphasesetting[0..1]
30	dqsoutputpowerdown
31	dqsoutputphaseinvert
32..33	dqoutputphasesetting[0..1]
34	—
35	dqoutputpowerdown
36	dqoutputphaseinvert
37..38	resyncinputphasesetting[0..1]
39	resyncinputpowerdown
40	resyncinputphaseinvert
41..42	postamblephasesetting[0..1]
43	—
44	postamblepowerdown
45	postamblephaseinvert
46..47	dqs2xoutputphasesetting[0..1]
48	dqs2xoutputpowerdown
49	dqs2xoutputphaseinvert
50..51	dq2xoutputphasesetting[0..1]

Table 7–8. DQS Configuration Block Bit Sequence (Part 2 of 2)

Bit	Bit Name
52	—
53	dq2xoutputpowerdown
54	dq2xoutputphaseinvert
55..56	ck2xoutputphasesetting[0..1]
57	ck2xoutputpowerdown
58	ck2xoutputphaseinvert
59..60	dqoutputzerophasesetting[0..1]
61..62	postamblezerophasesetting[0..1]
63	postamblepowerdown
64	dividerioehratephaseinvert
65	dividerphaseinvert
66..68	enaoctcycledelaysetting[0..2]
69	enaoctphasetransferreg
70..77	dqsdisablendelaysetting[0..7]
78..85	dqsenabledelaysetting[0..7]
86	enadqsenablephasetransferreg
87..88	dqsinputphasesetting[0..1]
89	enadqsphasetransferreg
90	enaoutputphasetransferreg
91..93	enadqscycledelaysetting[0..2]
94..96	enaoutputcycledelaysetting[0..2]
97	enainputcycledelaysetting
98	enainputphasetransferreg
99	—
100	—

Document Revision History

Table 7–9 lists the revision history for this chapter.

Table 7–9. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Figure 7–4, Figure 7–6, Figure 7–13, Figure 7–14, and Figure 7–17. ■ Updated Table 7–2, Table 7–7, and Table 7–8. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

Section III. System Integration

This section provides information about system integration in Stratix® V devices. This section includes the following chapters:

- Chapter 8, Hot Socketing and Power-On Reset in Stratix V Devices
- Chapter 9, Configuration, Design Security, and Remote System Upgrades in Stratix V Devices
- Chapter 10, SEU Mitigation in Stratix V Devices
- Chapter 11, JTAG Boundary-Scan Testing in Stratix V Devices
- Chapter 12, Power Management in Stratix V 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.

8. Hot Socketing and Power-On Reset in Stratix V Devices

SV51009-1.2

This chapter provides information about hot-socketing specifications, power-on reset (POR) requirements, and their implementation in Stratix® V devices.

Stratix V 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 V 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 using Stratix V devices on PCBs that contain a mixture of 3.0-, 2.5-, 1.8-, 1.5-, 1.35-, 1.25-, 1.20-, and 0.85-V devices.

The Stratix V hot-socketing feature provides the following:

- 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 chapter contains the following sections:

- "Stratix V Hot-Socketing Specifications" on page 8–1
- "Hot-Socketing Feature Implementation in Stratix V Devices" on page 8–2
- "Power-On Reset Circuitry" on page 8–3
- "Power-On Reset Specifications" on page 8–5

Stratix V Hot-Socketing Specifications

Stratix V devices are hot-socketing compliant without the need for external components or special design requirements. The advantages of hot-socketing support in Stratix V devices include the following:

- "Stratix V Devices Can Be Driven Before Power Up"
- "I/O Pins Remain Tri-Stated During Power Up"
- "Insertion or Removal of a Stratix V Device from a Powered-Up System"

Stratix V Devices Can Be Driven Before Power Up

You can drive signals into I/O pins, dedicated input pins, and dedicated clock pins of Stratix V devices before or during power up or power down without damaging the device. External input signals to I/O pins of the device do not internally power the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies of the device through internal paths within the device.

I/O Pins Remain Tri-Stated During Power Up

A device that does not support hot socketing can interrupt the system operation or cause contention by driving out before or during power up. In a hot-socketing situation, the Stratix V output buffers are turned off during system power up or power down. Also, the Stratix V device does not drive out until the device is configured and working within the recommended operating conditions.

Insertion or Removal of a Stratix V Device from a Powered-Up System

When powered up through the device signal pins, devices that do not support hot socketing are directly connected to the GND, causing power supplies failure. This irregular power up can damage both the driving and driven devices and can disrupt card power up.

You can insert a Stratix V device into or remove it from a powered-up system board without damaging the system board or interfering with its operation.

The hotsocket circuit monitors the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies. These power supplies can be powered up or powered down in any sequence. During hot socketing, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

You must ensure that all power supplies for Stratix V devices are within the ramp-up and ramp-down rate of 200 µs to 100 ms.

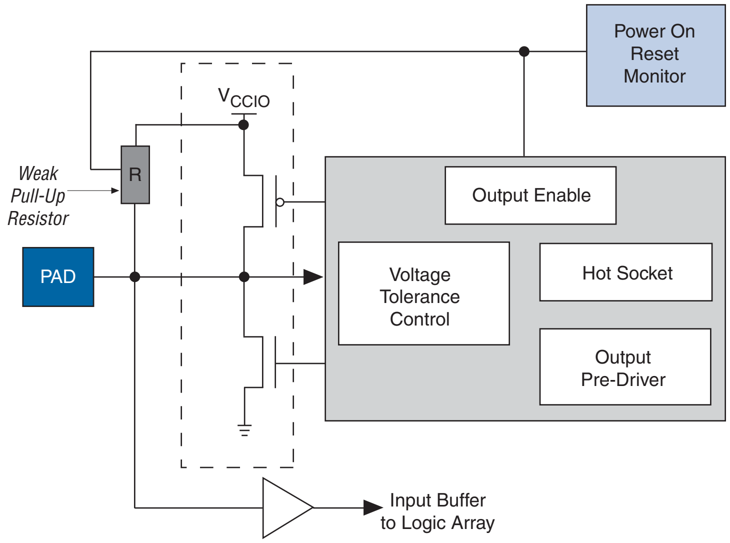
A possible concern regarding hot socketing is the potential for "latch up". When hot socketing, Stratix V devices are immune to latch up. Latch up occurs when electrical subsystems are hot socketed into an active system. During hot socketing, the signal pins can be connected and driven by the active system before the power supply can provide current to the power and GND planes of the device. This condition can lead to latch up and cause a low-impedance path from power to GND within the device. As a result, the device draws a large amount of current, possibly causing electrical damage.

Hot-Socketing Feature Implementation in Stratix V Devices

The hot-socketing feature turns off the output buffer during power up and power down of the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies. When the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies are below the threshold voltage, the hot-socketing circuitry generates an internal HOTSCKT signal. 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 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 is removed from these configuration pins to ensure that they can operate during configuration. Thus, it is an expected behavior for these pins to drive out during power-up and power-down sequences.

Figure 8–1 shows the I/O pin circuitry for Stratix V devices.

Figure 8–1. Hot-Socketing Circuitry for Stratix V Devices



The POR circuit monitors the voltage level of the power supplies (V_{CC}, V_{CCPGM}, V_{CCPD}, V_{CCAUX}, V_{CCBAT}, and V_{CCPT}) and keeps the I/O pins tri-stated until the device is in user mode. The weak pull-up resistor (R) in the Stratix V input/output element (IOE) keeps the I/O pins from floating. The 3.0-V tolerance control circuit permits 3.0 V to drive the I/O pins before the V_{CC}, V_{CCPT}, V_{CCPD}, and V_{CCIO} power supplies are powered and 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 over current condition with the Altera® device.

☞ For the V_{CCAUX} power supply, POR only monitors one of the VCCAUX pins. You must connect all the VCCAUX pins.

Power-On Reset Circuitry

This section describes POR circuitry in Stratix V devices. POR circuitry keeps the devices in the reset state until the power supply outputs are within operating range.

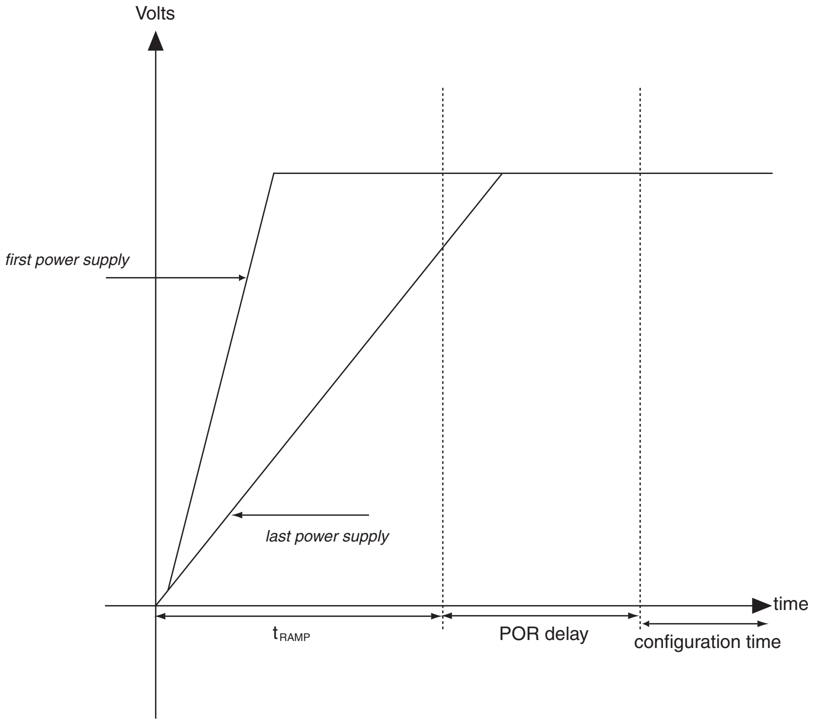
When power is applied to a Stratix V device, a POR event occurs if the power supply reaches the recommended operating range within the maximum power supply ramp time (t_{RAMP}). If t_{RAMP} is not met, the device I/O pins and programming registers remain tri-stated, during which device configuration could fail. The maximum t_{RAMP} for Stratix V devices is 100 ms; the minimum t_{RAMP} is 200 µs. Set the POR delay by selecting the MSEL pins. The standard or fast POR delay mode that is generated internally from the MSEL decoding block determines the POR delay time, as follows:

- For the standard POR delay mode, the POR delay time is 100 to 300 ms.
- For the fast POR delay mode, the POR delay time is 4 to 12 ms.
- The POR maximum pulse width is 12 ms, which leaves enough time after the POR trip for the PCI Express® (PCIe®) to initialize.

For more information about the MSEL pin settings, refer to the *Configuration, Design Security, and Remote System Upgrades in Stratix V Devices* chapter.

Figure 8–2 shows the relationship between t_{RAMP} and POR delay.

Figure 8–2. Relationship between t_{RAMP} and POR Delay



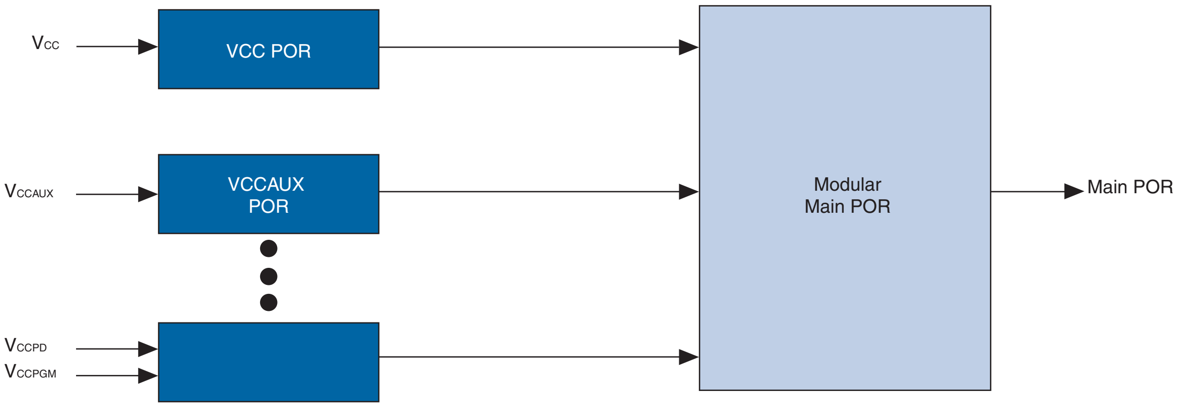
The Stratix V POR implements a modularized design structure to monitor the supplies gating the main POR. Each external power supply being monitored has a base POR, which employs a generic design to detect the corresponding voltage level for proper functionality.

It also checks for functionality of I/O level shifters powered by the V_{CCPD} and V_{CCPGM} power supplies during power-up mode. The main POR waits for all the individual PORs to release the POR signal so that the control block can start programming the device. The Stratix V main POR also employs a modularized structure to enable brown-out detection on a power supply and also to possibly expand the number of power supplies being monitored.

All configuration-related dedicated and dual function I/O pins must be powered by the V_{CCPGM} power supply.

Figure 8–3 shows a simplified POR diagram for Stratix V devices.

Figure 8–3. Simplified POR Diagram for Stratix V Devices



Power-On Reset Specifications

Table 8–1 lists the power supplies that the POR monitors.

Table 8–1. Power Supplies Monitored by the POR Circuitry

Power Supply	Description	Setting (V)
V_{CCAUX}	Auxiliary supply for the programmable power technology	2.5
V_{CCBAT}	Battery back-up power supply for design security volatile key register	3.0–1.2
V_{CC}	Core and periphery power supply	0.85
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

Table 8–2 lists the power supplies that the POR does not monitor.

Table 8–2. Power Supplies Not Monitored by the POR Circuitry

Power Supply	Description	Setting (V)
V_{CCT_GXB}	Transmitter power	0.85, 1.0
V_{CCH_GXB}	Transmitter output buffer power	1.5
V_{CCR_GXB}	Receiver power	0.85, 1.0
V_{CCA_GXB}	Transceiver high voltage power	2.5, 3.0
V_{CCHIP}	Transceiver HIP digital power	0.85
V_{CCIO}	I/O power supply	1.2, 1.25, 1.35, 1.5, 1.8, 2.5, 3.0
V_{CCA_FPLL}	PLL analog global power supply	2.5
V_{CCD_FPLL}	PLL digital power supply	1.5

For more information about the POR specification, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Document Revision History

Table 8–3 lists the revision history for this chapter.

Table 8–3. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated "Power-On Reset Circuitry" section. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

9. Configuration, Design Security, and Remote System Upgrades in Stratix V Devices

SV51010-1.3

This chapter contains information about the Stratix® V supported configuration schemes, instructions about how to execute the required configuration schemes, and all the necessary option pin settings. This chapter also reviews the different ways you can configure your device and explains the design security and remote system upgrade features for the Stratix V devices.

This chapter includes the following sections:

Stratix V devices use SRAM cells to store configuration data. Because SRAM memory is volatile, you must download the configuration data to the Stratix V device each time the device powers up. You can configure Stratix V devices using one of four configuration schemes:

- Fast passive parallel (FPP) (×8, ×16, and ×32)
- Active serial (AS) (×1 and ×4)
- Passive serial (PS)
- 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 about the configuration features, refer to "Configuration Features" on page 9–2.

Configuration Features

Stratix V devices offer decompression, design security, and remote system upgrade features. Stratix V devices can receive a compressed configuration bitstream and decompress this data in real-time, reducing storage requirements and configuration time. Design security using configuration bitstream encryption is available in Stratix V devices, which protects your designs. You can make real-time system upgrades of your Stratix V designs from remote locations with the remote system upgrade feature.

Table 9–1 lists which configuration features you can use in each configuration scheme.

Table 9–1. Configuration Features for Stratix V Devices

Configuration Scheme	Decompression	Design Security	Remote System Upgrade
FPP (×8, ×16, ×32)	✓ *(1)*	✓ *(1)*	—
AS (×1, ×4)	✓	✓	✓
PS	✓	✓	—
JTAG	—	—	—

Note to Table 9–1:

(1) In these configuration schemes, the host system must accommodate a different `DCLK`-to-`DATA[]` ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

Power-On Reset Circuit and Configuration Pins Power Supply

The following sections describe the power-on reset (POR) circuit and the power supply for the configuration pins.

POR Delay Specification

POR delay is defined as the delay between the time when all the power supplies monitored by the POR circuitry reach the minimum recommended operating voltage to the time when the `nSTATUS` is released high and your device is ready to begin configuration.

For more information about the POR delay, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

Table 9–2 lists the fast and standard POR delay specification.

Table 9–2. Fast and Standard POR Delay Specification *(Note 1)*

POR Delay	Minimum	Maximum
Fast	4 ms	12 ms
Standard	100 ms	300 ms

Note to Table 9–2:

(1) You can select the POR delay based on the `MSEL` settings as described in Table 9–4 on page 9–7.

Power-On Reset Circuit

The POR circuit keeps the entire system in reset mode until the power supply voltage levels have stabilized on power-up. After power-up, the device does not release nSTATUS until all the power supplies monitored by the POR circuitry are above the device's POR trip point. On power down, brown-out occurs if any of the power supplies monitored by the POR circuitry drops below the threshold level of the hot-socket circuitry.

For more information about which power supplies are monitored by the POR circuitry, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

V_{CCPGM} Pin

Stratix V devices have a power supply, V_{CCPGM}, for all dedicated configuration pins and dual-purpose pins. The supported configuration voltages are 1.8, 2.5, and 3.0 V.

Use the V_{CCPGM} pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bidirectional pins, and the dual-purpose pins that you use for configuration. The configuration input buffers do not have to share power lines with the regular I/O buffer in Stratix V devices.

The operating voltage for the configuration input pin is independent of the I/O banks power supply, V_{CCIO}, during configuration. Therefore, Stratix V devices do not require configuration voltage constraints on V_{CCIO}.

For more information about the configuration pins connections, refer to the *Stratix V Device Family Pin Connection Guidelines*.

V_{CCPD} Pin

Stratix V devices have a dedicated programming power supply, V_{CCPD}, which must be connected to 3.0 V or 2.5 V to power the I/O pre-drivers and JTAG I/O pins (TCK, TMS, TDI, TDO, and TRST).

☞ V_{CCPD} must be greater than or equal to V_{CCIO}. If V_{CCIO} is set to 3.0 V, V_{CCPD} must be powered up to 3.0 V. If the V_{CCIO} of the bank is set to 2.5 V or lower, V_{CCPD} must be powered up to 2.5 V. This applies for all the banks containing the VCCPD and VCCIO pins.

For more information about the configuration pins power supply, refer to "Device Configuration Pins" on page 9–38.

Configuration Sequence

The following sections describe the general configuration process for the FPP, AS, and PS schemes.

Power Up

To begin the configuration process, you must fully power-up all the power supplies monitored by the POR circuitry to the appropriate voltage levels. The power supplies must ramp-up monotonically within the specified ramp-up time to ensure successful configuration.

☞ All power supplies including the V_{CCPGM} and V_{CCPD} must ramp-up from 0 V to the desired voltage level within the ramp-up time specification. If these supplies are not ramped up within this specified time, your Stratix V device will not configure successfully. If your system cannot ramp-up the power supplies within the specified ramp-up time specification, you must hold nCONFIG low until all the power supplies are stable.

For more information about the ramp-up time specification, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

Reset

After power-up, the Stratix V device goes through a POR. The POR delay depends on the MSEL settings. During POR, the device resets, holds nSTATUS low, clears the configuration RAM bits, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins remain tri-stated until the device is configured.

While nCONFIG is low, the device is in reset. When the device comes out of reset, nCONFIG must be at a logic-high level in order for the device to release the open-drain nSTATUS pin. After nSTATUS is released, it is pulled high by a pull-up resistor and the device is ready to receive configuration data. Before and during configuration, all user I/O pins are 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 the device exits POR, before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

For more information about the POR delay specification, refer to "POR Delay Specification" on page 9–2.

Configuration

Both nCONFIG and nSTATUS must be deasserted at a logic-high level in order for the configuration stage to begin. For the FPP and PS configuration schemes, the device receives configuration data on its DATA pins and the clock source on the DCLK pin. Configuration data is latched into the Stratix V device on the rising edge of DCLK. For the AS configuration scheme, the device receives configuration data on its AS_DATA[] pins and drives the clock source on the DCLK pin. Configuration data is latched into the Stratix V device on the falling edge of DCLK.

After the Stratix V device has received all the configuration data successfully, it releases the `CONF_DONE` pin, which is pulled high by a pull-up resistor. A low-to-high transition on `CONF_DONE` indicates configuration has completed and initialization of the device can begin. For the FPP and PS schemes, `DCLK` must not be left floating at the end of configuration. You must drive them either high or low, whichever is convenient on your board.

☞ For the FPP and PS schemes, there is no maximum `DCLK` period, which means you can stop the configuration by holding the `DCLK` low for an indefinite amount of time. To resume configuration, the external host must provide data on the `DATA[]` pins prior to sending the first `DCLK` rising edge.

Configuration Error

If the **Auto-restart configuration after error** option (available in the Quartus® II software from the **General** panel of the **Device and Pin Options** dialog box) is turned on, the Stratix V device releases the `nSTATUS` pin high after the specified time indicated by `tSTATUS` and retries the configuration. If this option is turned off or if you are using a PS or FPP scheme with an external controller, the system must monitor the `nSTATUS` for errors and sends a low-to-high signal on `nCONFIG` for specified t_{CFG} time to restart the configuration.

Initialization

In Stratix V devices, initialization begins after the `CONF_DONE` goes high. For the FPP and PS configuration schemes, two `DCLK` falling edges are required after the last configuration byte is sent to the Stratix V device to begin the initialization of the device for both uncompressed and compressed configuration data.

The initialization clock source is from the internal oscillator, `CLKUSR`, or `DCLK` pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix V device provides itself with enough clock cycles for proper initialization.

Table 9–3 lists the initialization clock source option, the applicable configuration schemes, and the maximum frequency.

Table 9–3. Initialization Clock Source Option and the Maximum Frequency

Initialization Clock Source	Configuration Schemes	Maximum Frequency	Minimum Number of Clock Cycles *(1)*
Internal Oscillator	AS, PS, FPP	12.5 MHz	17,408 *(3)*
`CLKUSR`	AS, PS, FPP *(2)*	125 MHz	

Notes to Table 9–3:

(1) The minimum number of clock cycles required for device initialization.

(2) To enable `CLKUSR` as the initialization clock source, turn on the **Enable user-supplied start-up clock (`CLKUSR`)** option in the Quartus II software from the **General** panel of the **Device and Pin Options** dialog box.

(3) The number is still preliminary.

☞ If you use the optional `CLKUSR` pin as the initialization clock source and `nCONFIG` is pulled low to restart configuration during device initialization, ensure that `CLKUSR` or `DCLK` continues toggling until `nSTATUS` goes low and goes high again.

CLKUSR provides you with the flexibility to synchronize initialization of multiple devices or to delay initialization. Supplying a clock on the CLKUSR pin during initialization does not affect configuration. After CONF_DONE goes high, CLKUSR or DCLK is enabled after the time specified by t_{CD2CU}. When this time period elapses, Stratix V devices require a minimum number of clock cycles to initialize properly and enter user mode as specified by the t_{CD2UMC} parameter.

User Mode

The Stratix V device enters user mode when the initialization is complete. You can monitor the end of the initialization stage by enabling the optional INIT_DONE pin. If enabled, the low-to-high transition of INIT_DONE indicates the device has completed initialization and entered user mode. In this mode, your design is executed. The user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

☞ At any time during the configuration stage or user mode operation, you can initiate a reconfiguration by setting a low pulse on the nCONFIG pin. The pulse must meet the minimum t_{CFG} low-pulse width. When nCONFIG is pulled low, the nSTATUS and CONF_DONE pins are pulled low and all I/O pins are tri-stated. Configuration begins when the nCONFIG and nSTATUS pins return to a logic-high level.

Configuration Schemes

The following sections describe configuration schemes for Stratix V devices.

MSEL Pin Settings

Select the configuration scheme by driving the Stratix V device MSEL pins either high or low (refer to Table 9–4). The MSEL input buffers are powered by the V_{CCPGM} power supply. Altera recommends hardwiring the MSEL pins to V_{CCPGM} or GND. During POR and during reconfiguration, the MSEL pins must be at the LVTTL V_{IL} and V_{IH} levels to be considered logic low and logic high, respectively.

☞ To avoid problems with detecting an incorrect configuration scheme, hardwire the MSEL pins to V_{CCPGM} or GND without pull-up or pull-down resistors. Do not drive the MSEL pins with a microprocessor or another device.

Table 9–4 lists the configuration schemes for Stratix V devices.

Table 9–4. Configuration Schemes for Stratix V Devices (Part 1 of 2)

Configuration Scheme	Decompression Feature	Design Security Feature	Configuration Voltage Standard (V) (2)	POR Delay (5)	MSEL[4..0]
FPP ×8	Disabled	Disabled	1.8/2.5/3.0	Fast	10100
				Standard	11000
	Disabled	Enabled	1.8/2.5/3.0	Fast	10101
				Standard	11001
	Enabled	Optional (1)	1.8/2.5/3.0	Fast	10110
				Standard	11010
FPP ×16	Disabled	Disabled	1.8/2.5/3.0	Fast	00000
				Standard	00100
	Disabled	Enabled	1.8/2.5/3.0	Fast	00001
				Standard	00101
	Enabled	Optional (1)	1.8/2.5/3.0	Fast	00010
				Standard	00110
FPP ×32	Disabled	Disabled	1.8/2.5/3.0	Fast	01000
				Standard	01100
	Disabled	Enabled	1.8/2.5/3.0	Fast	01001
				Standard	01101
	Enabled	Optional (1)	1.8/2.5/3.0	Fast	01010
				Standard	01110
PS	Optional (1)	Optional (1)	1.8/2.5/3.0	Fast	10000
				Standard	10001
AS (×1, ×4) (3)	Optional (1)	Optional (1)	3.0	Fast	10010
				Standard	10011

Table 9–4. Configuration Schemes for Stratix V Devices (Part 2 of 2)

Configuration Scheme	Decompression Feature	Design Security Feature	Configuration Voltage Standard (V) *(2)*	POR Delay *(5)*	MSEL[4..0]
JTAG-based configuration *(4)*	Disabled	Disabled	—	—	*(6)*

Notes to Table 9–4:

(1) You can select to enable or disable this feature.

(2) The configuration voltage standard applied to the V_{CCPGM} power supply that powers all the configuration pins during configuration.

(3) The AS configuration scheme supports the remote system upgrade feature. For more information about the remote system upgrade feature, refer to "Remote System Upgrades" on page 9–44.

(4) JTAG-based configuration takes precedence over other configuration schemes. This means the MSEL pin settings are ignored. JTAG-based configuration does not support the design security or decompression features.

(5) For POR delay specification, refer to "POR Delay Specification" on page 9–2.

(6) Do not leave the MSEL pins floating. Connect them to V_{CCPGM} or GND. This pin supports the non-JTAG configuration scheme used in production. If you only use the JTAG configuration, Altera recommends connecting the MSEL pins to GND.

Raw Binary File Size

For the POR delay specification, refer to "POR Delay Specification" on page 9–2.

Table 9–5 lists the uncompressed raw binary file (**.rbf**) sizes for Stratix V devices.

Table 9–5. Uncompressed .rbf Sizes for Stratix V Devices ***(Note 1)***

Family	Device	Configuration .rbf Size (bits)
Stratix V GX	5SGXA3	139,255,840
	5SGXA4	139,255,840
	5SGXA5	266,599,584
	5SGXA7	266,599,584
	5SGXA9	387,394,048
	5SGXAB	387,394,048
	5SGXB5	266,798,896
	5SGXB6	266,798,896
Stratix V GT	5SGTC5	266,035,472
	5SGTC7	266,035,472
Stratix V GS	5SGSD2	93,080,448
	5SGSD3	93,080,448
	5SGSD4	209,935,224
	5SGSD5	209,935,224
	5SGSD6	266,798,896
	5SGSD8	266,798,896
Stratix V E	5SEE9	387,394,048
	5SEEB	387,394,048

Note to Table 9–5:

(1) These values are preliminary.

Use the data in Table 9–5 to estimate the file size before design compilation. Different configuration file formats, such as a hexadecimal (**.hex**) or tabular text file (**.ttf**) format, have different file sizes. For the different types of configuration file and file sizes, refer to the Quartus II software. However, for a 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 depends on your 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*.

Fast Passive Parallel Configuration

The FPP configuration using an external host provides the fastest method to configure Stratix V devices. FPP is supported in multiple data widths—8-bits, 16-bits, and 32-bits. You can perform a FPP configuration of Stratix V devices using an external host such as a MAX II device or microprocessor. The external host controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix V device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** formats. Therefore, the design that controls the configuration stages, such as fetching the data from flash memory and sending it to the device, must be stored in the MAX II device or microprocessor.

The Parallel Flash Loader (PFL) feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface. PFL also acts as a controller to read configuration data from the flash memory device and configures the Stratix V device. PFL supports both the PS and FPP configuration schemes.

For more information about the PFL, refer to *Parallel Flash Loader Megafunction User Guide*.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed configuration data in a FPP configuration.

DCLK-to-DATA[] Ratio for FPP configuration

FPP configuration requires a different DCLK-to-DATA[] ratio when you enable the design security, decompression, or both features. Table 9–6 lists the DCLK-to-DATA[] ratio for each combination.

Table 9–6. DCLK-to-DATA[] Ratio ***(Note 1)*** **(Part 1 of 2)**

Configuration Scheme	Decompression	Design Security	DCLK-to-DATA[] Ratio
FPP ×8	Disabled	Disabled	1
	Disabled	Enabled	1
	Enabled	Disabled	2
	Enabled	Enabled	2

Table 9–6. DCLK-to-DATA[] Ratio ***(Note 1)*** **(Part 2 of 2)**

Configuration Scheme	Decompression	Design Security	DCLK-to-DATA[] Ratio
FPP ×16	Disabled	Disabled	1
	Disabled	Enabled	2
	Enabled	Disabled	4
	Enabled	Enabled	4
FPP ×32	Disabled	Disabled	1
	Disabled	Enabled	4
	Enabled	Disabled	8
	Enabled	Enabled	8

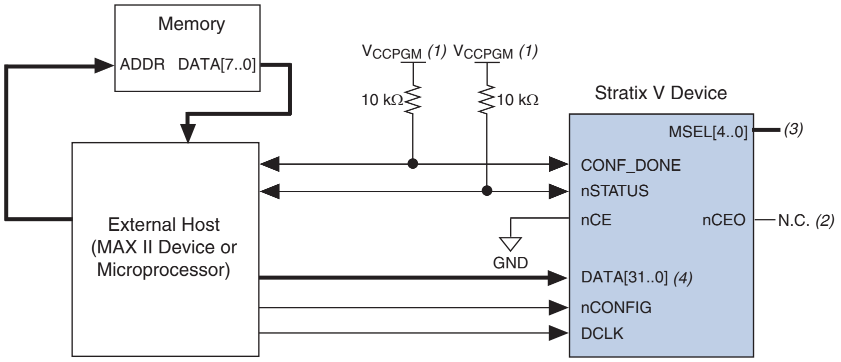
Note to Table 9–6:

(1) Depending on the `DCLK`-to-`DATA[]` ratio, the host must send a `DCLK` frequency that is r times the data rate in bytes per second (Bps), or words per second (Wps). For example, in FPP ×16 when the `DCLK`-to-`DATA[]` ratio is 2, the `DCLK` frequency must be 2 times the data rate in Wps. Stratix V devices use the additional clock cycles to decrypt and decompress the configuration data.

If the `DCLK`-to-`DATA[]` ratio is greater than 1, at the end of configuration, you can only stop the `DCLK` (`DCLK`-to-`DATA[]` ratio – 1) clock cycles after the last data is latched into the Stratix V device.

Figure 9–1 shows the configuration interface connections between the Stratix V device and a MAX II device for single device configuration.

Figure 9–1. Single Device FPP Configuration Using an External Host



Notes to Figure 9–1:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V 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 powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the `nCEO` pin unconnected or use it as a user I/O pin when it does not feed another device's `nCE` pin.

(3) The `MSEL` pin settings vary for different data width, configuration voltage standards, and POR delay. To connect `MSEL`, refer to Table 9–4 on page 9–7.

(4) If you use FPP ×8, use `DATA[7..0]`. If you use FPP ×16, use `DATA[15..0]`.

FPP Multi-Device Configuration

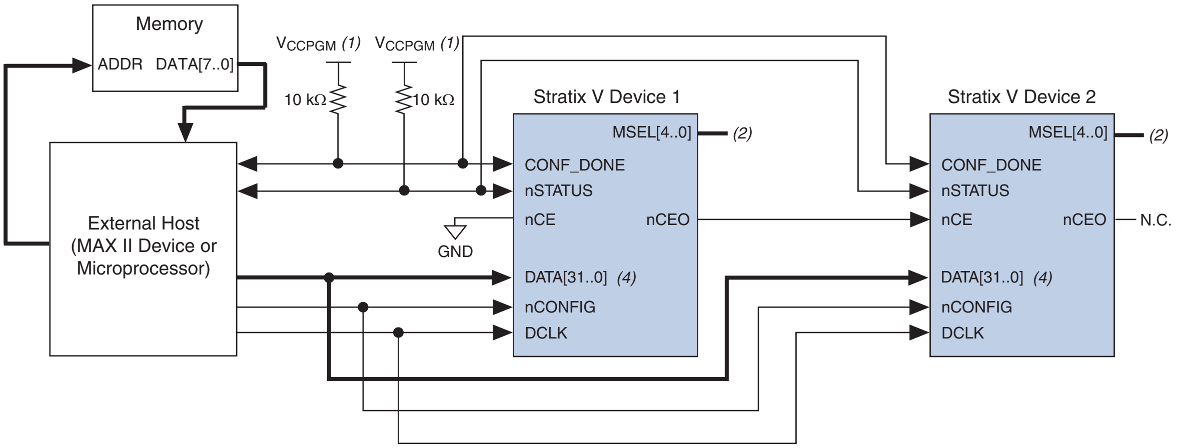
For FPP multi-device configuration, you can configure all devices with different sets of configuration data (multiple SRAM object files [**.sof**s]) or with the same configuration data (single **.sof**). In both cases, the nCONFIG, nSTATUS, DCLK, DATA[], and CONF_DONE pins are connected to every device in the chain. Ensure that the DCLK and data line are buffered for every fourth device. This ensures the signal integrity and prevents clock skew problems.

Because all device's CONF_DONE and nSTATUS pins are tied together, all devices initialize and enter user mode at the same time. If any device detects an error, configuration stops for the entire chain and you must reconfigure all devices. 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.

For FPP multi-device configuration, all devices in the chain must have the same data width. If you are using FPP ×32, all devices in the chain must use FPP ×32 configuration scheme. If you are using FPP ×8, you can use the Stratix V device with other FPGA devices that support FPP ×8.

Figure 9–2 shows how to configure multiple devices using a MAX II device when both devices receive a different set of configuration data (multiple **.sof**s).

Figure 9–2. Multi-Device FPP Configuration Using an External Host When Both Devices Receive a Different Set of Configuration Data



Notes to Figure 9–2:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V 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 powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

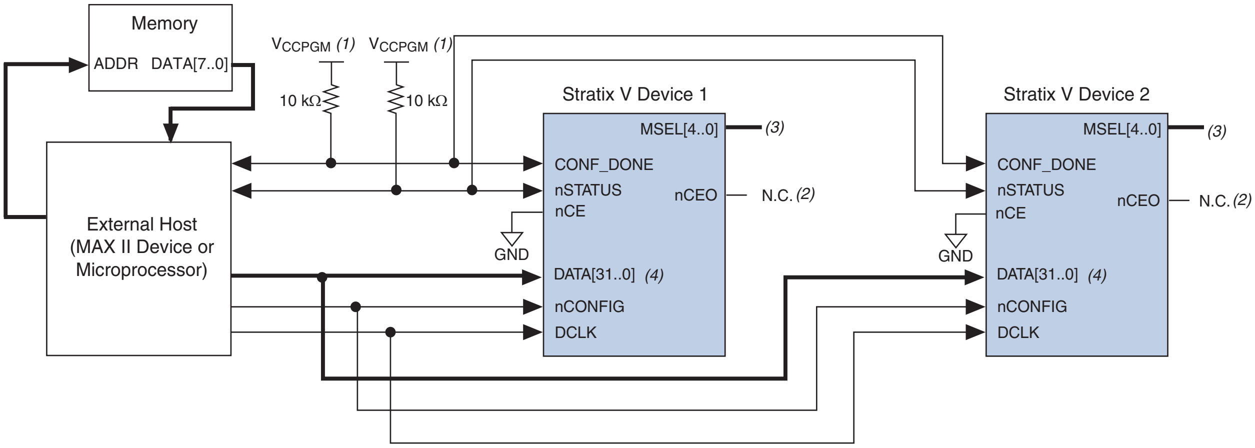
(3) The MSEL pin settings vary for different data width and POR delay. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) If you use FPP ×8, use DATA[7..0]. If you use FPP ×16, use DATA[15..0]. All devices in the chain must have the same data width.

In Figure 9–2, 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 in one clock cycle; therefore, the transfer of data to the second device is transparent to the MAX II device or microprocessor.

Figure 9–3 shows the FPP configuration setup for multiple devices when both Stratix V devices receive the same configuration data (single **.sof**).

Figure 9–3. Multiple Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 9–3:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V 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 powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the `nCEO` pin unconnected or use it as a user I/O pin when it does not feed another device's `nCE` pin.

(3) The `MSEL` pin settings vary for different data width and POR delay. To connect `MSEL`, refer to Table 9–4 on page 9–7.

(4) If you use FPP ×8, use `DATA[7..0]`. If you use FPP ×16, use `DATA[15..0]`. All devices in the chain must have the same data width.

In Figure 9–3, because both nCE pins are tied to GND, both devices in the chain begin and complete the configuration and enter user mode at the same time.

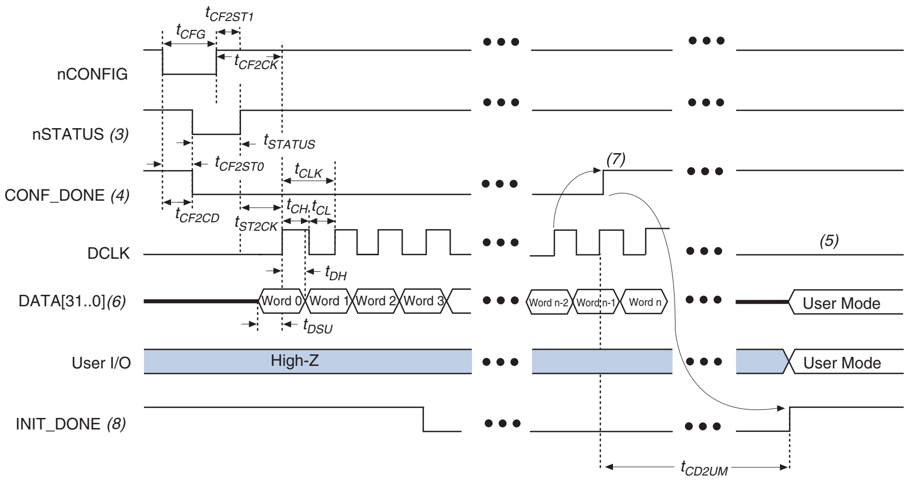
To configure FPP multi-device with a single **.sof**, all Stratix V devices in the chain must be in the same package and density.

FPP Configuration Timing

Figure 9–4 shows the timing waveform for FPP configuration when using a MAX II device as an external host. This waveform shows timing when the DCLK-to-DATA[] ratio is 1.

When you enable the decompression or design security feature, the DCLK-to-DATA[] ratio varies for FPP ×8, FPP ×16, and FPP ×32. For the respective DCLK-to-DATA[] ratio, refer to Table 9–6 on page 9–9.

Figure 9–4. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is 1 ***(Note 1), (2)***



Notes to Figure 9–4:

(1) Use this timing waveform when the DCLK-to-DATA[] ratio is 1.

(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 V 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) For FPP ×16, use DATA[15..0]. For FPP ×8, use DATA[7..0]. DATA[31..0] are available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.

(7) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high when the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(8) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–7 lists the timing parameters for Stratix V devices for FPP configuration when the DCLK-to-DATA[] ratio is 1.

Table 9–7. FPP Timing Parameters for Stratix V Devices *(Note 1)*, *(2)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(4)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	$0.45 \times 1/f_{MAX}$	—	s
t_{CL}	DCLK low time	$0.45 \times 1/f_{MAX}$	—	s
t_{CLK}	DCLK period	$1/f_{MAX}$	—	s
f_{MAX}	DCLK frequency (FPP ×8/×16)	—	125	MHz
	DCLK frequency (FPP ×32)	—	100	MHz
t_R	Input rise time	—	40	ns
t_F	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(5)*	175	437	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} + (17,408 × CLKUSR period) *(6)*	—	—

Notes to Table 9–7:

(1) This information is preliminary.

(2) Use these timing parameters when the decompression and design security features are disabled.

(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

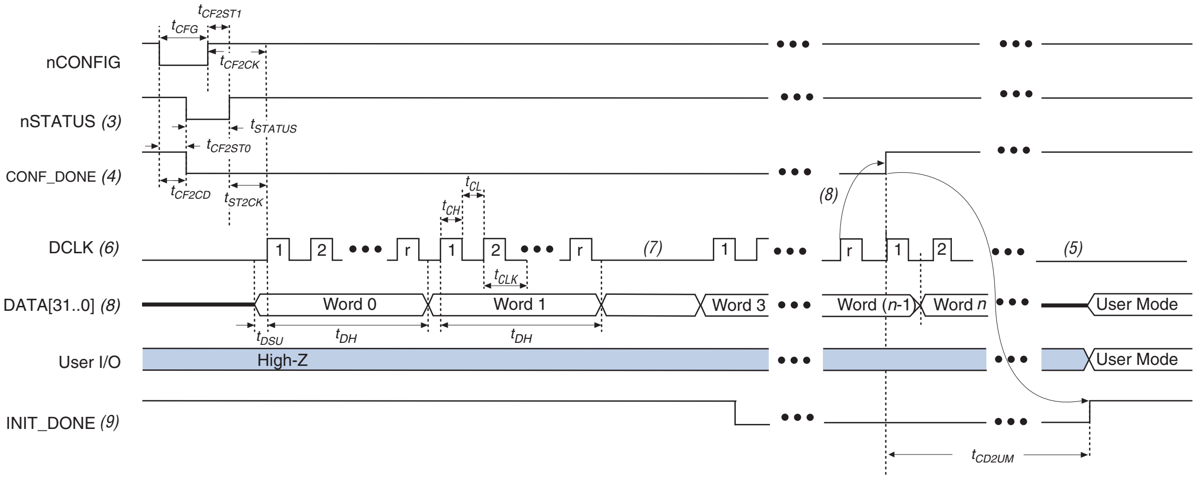
(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.

(5) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for initializing the device.

(6) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

Figure 9–5 shows the timing waveform for FPP configuration when using a MAX II device or microprocessor as an external host. This waveform shows timing when the DCLK-to-DATA[] ratio is more than 1.

Figure 9–5. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is >1 ***(Note 1), (2)***



Notes to Figure 9–5:

(1) Use this timing waveform and parameters when the DCLK-to-DATA[] ratio is >1. To find out the DCLK-to-DATA[] ratio for your system, refer Table 9–6 on page 9–9.

(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 V device holds nSTATUS low for the time as specified by 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) "r" denotes the DCLK-to-DATA[] ratio. For the DCLK-to-DATA[] ratio based on the decompression and the design security feature enable settings, refer to Table 9–6 on page 9–9.

(7) If needed, pause DCLK by holding it low. When DCLK restarts, the external host must provide data on the DATA[31..0] pins prior to sending the first DCLK rising edge.

(8) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high after the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(9) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–8 lists the timing parameters for Stratix V devices for FPP configuration when the DCLK-to-DATA[] ratio is more than 1.

Table 9–8. FPP Timing Parameters for Stratix V Devices When the DCLK-to-DATA[] Ratio is >1 ***(Note 1), (2)*** **(Part 1 of 2)**

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns

Table 9–8. FPP Timing Parameters for Stratix V Devices When the DCLK-to-DATA[] Ratio is >1 *(Note 1)*, *(2)* (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{DH}	DATA[] hold time after rising edge on DCLK	3 × 1/f_{DCLK} *(6)*	—	s
t_{CH}	DCLK high time	0.45 × 1/f_{MAX}	—	s
t_{CL}	DCLK low time	0.45 × 1/f_{MAX}	—	s
t_{CLK}	DCLK period	1/f_{MAX}	—	s
f_{MAX}	DCLK frequency (FPP ×8/×16)	—	125	MHz
	DCLK frequency (FPP ×32)	—	100	MHz
t_R	Input rise time	—	40	ns
t_F	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	175	437	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} + (17,408 × CLKUSR period) *(5)*	—	—

Notes to Table 9–8:

(1) This information is preliminary.

(2) Use these timing parameters when you use the decompression and design security features.

(3) You can obtain this value if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(4) The minimum and maximum numbers apply only if you use the internal oscillator as the clock source for initializing the device.

(5) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

(6) f_{DCLK} is the DCLK frequency the system is operating.

Active Serial Configuration (Serial Configuration Devices)

The AS configuration scheme is supported in 1-bit data wide (AS ×1 mode) or 4-bit data wide (AS ×4 mode). In the AS ×1 mode, the Stratix V devices are configured using a serial configuration device (EPCS). In the AS ×4 mode, a quad-serial configuration device (EPCQ) configures the Stratix V devices. The AS ×4 mode provides four times faster configuration time than the AS ×1 mode.

EPCS and EPCQ are low-cost devices with non-volatile memory that feature a simple four-pin interface or six-pin interface, respectively, and a small form factor. These features make the AS configuration scheme an ideal low-cost configuration solution.

☞ If you wish to gain control of the EPCS pins, hold the nCONFIG pin low and pull the nCE pin high. This causes the device to reset and tri-state the AS configuration pins.

For more information about EPCS and EPCQ devices, refer to the volume 2 of the *Configuration Handbook*.

AS mode supports DCLK frequency up to 100 MHz. You can choose the CLKUSR or internal oscillator as the configuration clock source that drives DCLK. If you use the internal oscillator as the configuration clock source, you can choose a 12.5, 25, 50, or 100 MHz clock from the **Configuration** panel in the **Device and Pins Option** settings.

Table 9–9 lists the DCLK frequency specification in the AS configuration scheme.

Table 9–9. DCLK Frequency Specification in the AS Configuration Scheme *(Note 1), (2)*

Minimum	Typical	Maximum	Unit
5.3	7.9	12.5	MHz
10.6	15.7	25.0	MHz
21.3	31.4	50.0	MHz
42.6	62.9	100.0	MHz

Notes to Table 9–9:

(1) This information is preliminary.

(2) This applies to the DCLK frequency specification when using the internal oscillator as the configuration clock source.

☞ You can choose the internal oscillator or CLKUSR as the DCLK clock source by selecting the option under **Device and Pins Option** settings, **Configuration** panel in the Quartus II software. This sets specific option in the programming file. By default, in AS scheme, Stratix V devices power-up and begin configuration with 12.5 MHz internal oscillator as the DCLK clock source. After reading the option bits from the programming file, Stratix V devices continue using the internal oscillator at 12.5 MHz frequency, switch to a higher internal oscillator clock frequency, or switch to the CLKUSR pin.

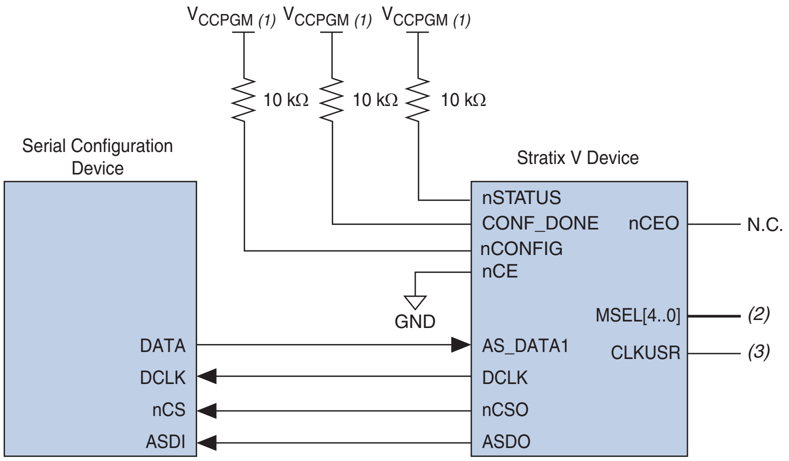
☞ If you choose CLKUSR as configuration clock source, the maximum frequency allowed is 100 MHz.

During device configuration, Stratix V devices read the configuration data using the serial interface, decompress the data if necessary, and configure their SRAM cells. In the AS scheme, the Stratix V device controls the configuration interface. In the PS scheme, the external host (a MAX II device or microprocessor) controls the interface.

☞ You can select between the AS ×1 and AS ×4 settings by selecting the option under **Device and Pins Option** settings, **Configuration** panel in the Quartus II software. This sets a specific option bit in the programming file. By default, in the AS scheme, Stratix V devices power-up and begin configuration as an AS ×1 mode. Upon reading the option bits from the programming file, Stratix V devices either stay as an AS ×1 mode or switch to an AS ×4 mode for the rest of the configuration.

Figure 9–6 shows the single-device configuration setup for an AS ×1 mode.

Figure 9–6. Single Device AS ×1 Mode Configuration



Notes to Figure 9–6:

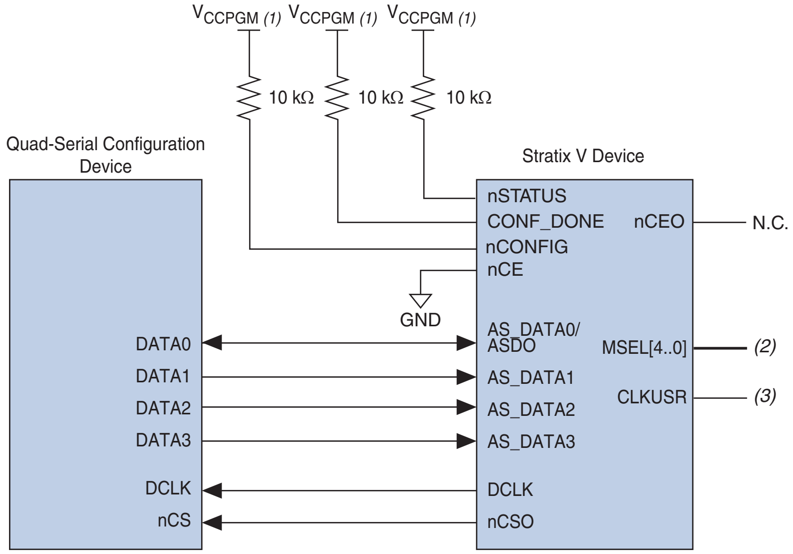
(1) Connect the pull-up resistors to V_{CCPGM} at 3.0-V supply.

(2) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(3) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–7 shows the single-device configuration setup for an AS ×4 mode.

Figure 9–7. Single Device AS ×4 Mode Configuration



Notes to Figure 9–7:

(1) Connect the pull-up resistors to V_{CCPGM} at 3.0-V supply.

(2) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(3) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

The serial clock (DCLK) generated by the Stratix V device controls the entire configuration cycle and provides timing for the serial interface. Stratix V devices use an internal oscillator or external clock (CLKUSR) as the configuration clock source (DCLK). In the AS configuration scheme, Stratix V devices drive out control signals on the falling edge of DCLK and latch in the data on the following falling edge of DCLK.

During configuration, Stratix V devices enable the EPCS or EPCQ by driving the nCSO output pin low, which connects to the chip select (nCS) pin of the EPCS or EPCQ. Stratix V devices use the serial clock (DCLK) and serial data output (ASDO) pins to send operation commands and read address signals to the EPCS or EPCQ. The EPCS or EPCQ provides data on its serial data output (DATA[]) pin, which connects to the AS_DATA[] input of the Stratix V devices.

AS Multi-Device Configuration

For the AS multi-device configuration scheme, you can configure all devices with different sets of configuration data (different **.sof**) or with the same configuration data (same **.sof**). In both cases, the nCONFIG, nSTATUS, DCLK, and data line (AS_DATA1 on the master device and DATA0 on the slave device) and CONF_DONE pins are connected to every device in the chain. Ensure that the DCLK and data line are buffered for every fourth device.

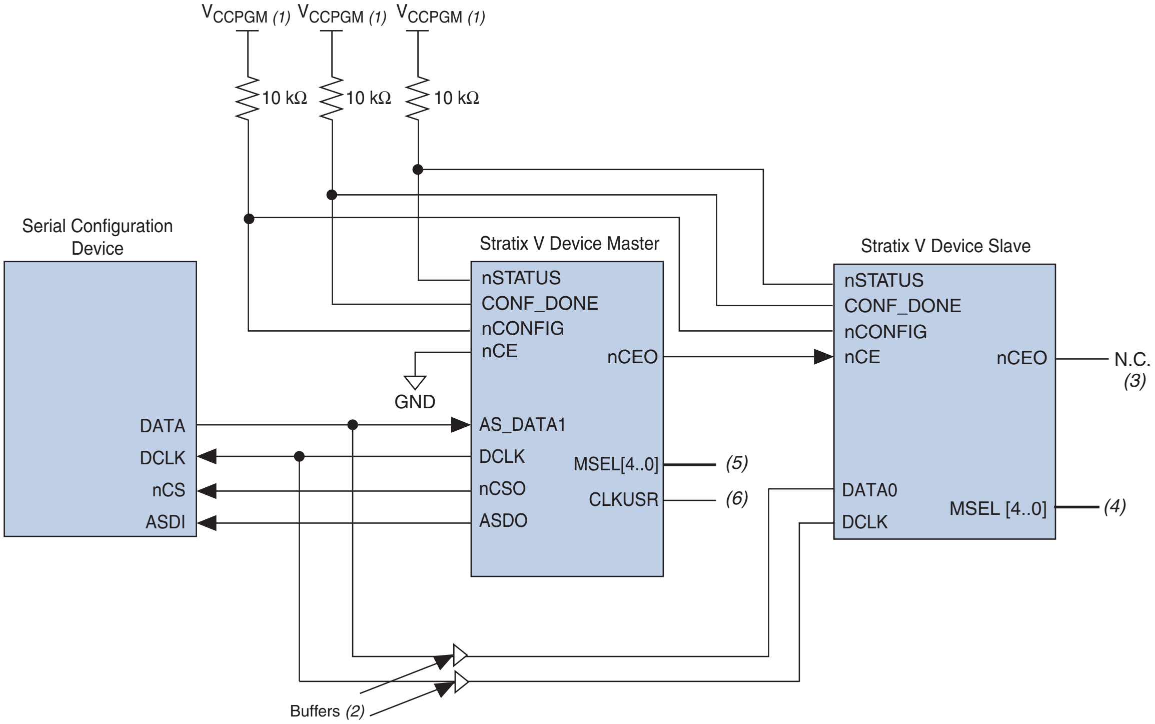
☞ The AS configuration scheme supports multi-device in AS ×1 mode. The AS ×4 mode does not support multi-device configuration setup.

In the AS multi-device configuration, the nSTATUS, nCONFIG, and CONF_DONE pins are tied together. Therefore, all devices initialize and enter user mode at the same time. If any device detects an error, configuration stops for the entire chain and you must reconfigure all devices. 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.

In this configuration scheme, the first Stratix V device in the chain is the configuration master and controls the configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Stratix V devices are configuration slaves. You must connect their MSEL pins to select the PS configuration scheme. Any other Altera® device that supports a PS configuration can also be part of the chain as the configuration slave.

Figure 9–8 shows the multi-device configuration setup for AS ×1 mode when both devices in the chain receive different sets of configuration data (multiple **.sof**s).

Figure 9–8. AS Multi-Device Configuration When Both Devices in the Chain Receive Different Sets of Configuration Data *(Note 1)*, *(2)*



Notes to Figure 9–8:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.

(2) Connect the repeater buffers between the Stratix V master and slave device for AS_DATA1/DATA0 and DCLK.

(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(4) For the appropriate MSEL settings based on POR delay settings, set the slave device MSEL setting to the PS scheme. Refer to Table 9–4 on page 9–7.

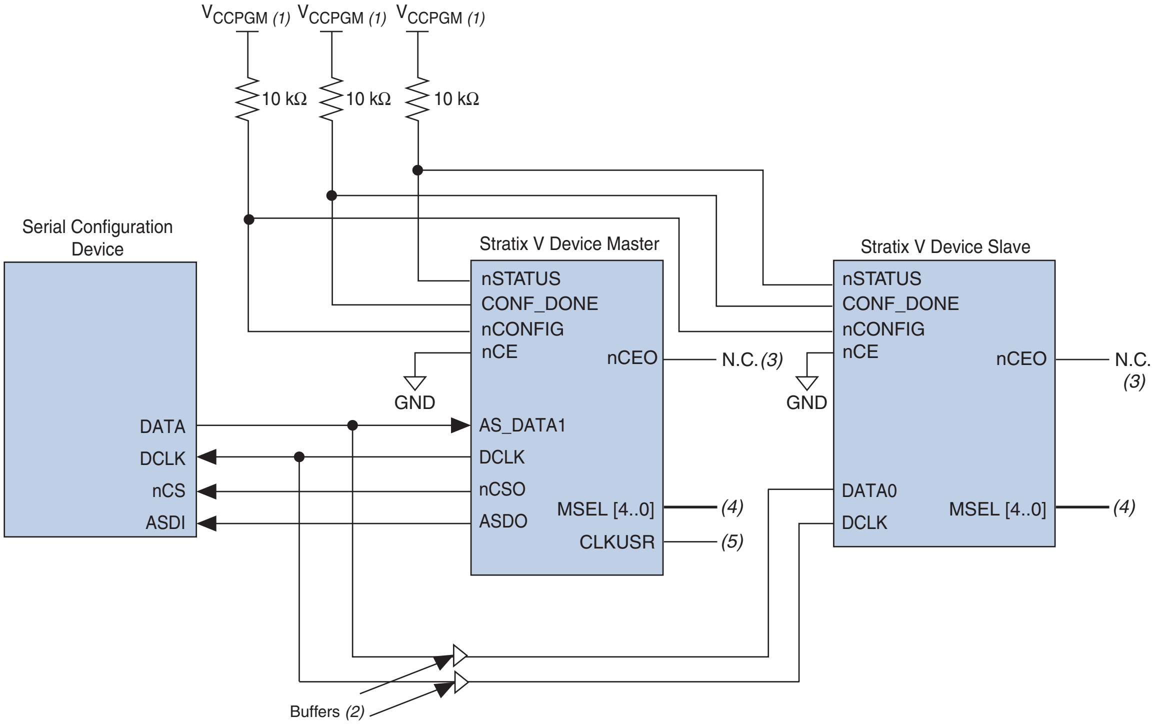
(5) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(6) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

In Figure 9–8, 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 in one clock cycle; therefore, the transfer of data to the second device is transparent to the first device in the chain.

Figure 9–9 shows the multi-device configuration setup for AS ×1 mode when all devices in the chain receive the same set of configuration data (single **.sof**).

Figure 9–9. AS Multi-Device Configuration When the Devices Receive the Same Data Using a Single .sof ***(Note 1)***



Notes to Figure 9–9:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.
(2) Connect the repeater buffers between the Stratix V master and slave device for AS_DATA1/DATA0 and DCLK.
(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.
(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.
(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

AS Connection Guidelines

Table 9–10 lists the board trace length and loading between the supported EPCS or EPCQ and Stratix V devices for single- and multi-device AS configurations.

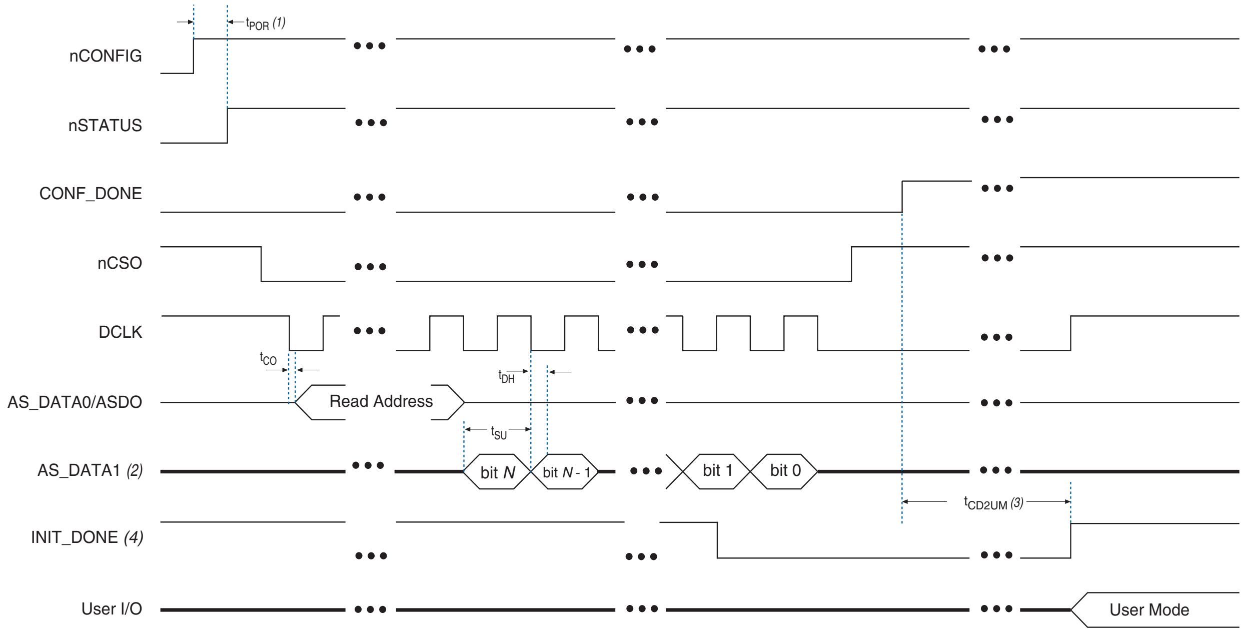
Table 9–10. Maximum Trace Length and Loading for AS ×1/×4 Configurations

Stratix V Device AS Pins	Maximum Board Trace Length from the Stratix V Device to the Serial Configuration Device for 12.5/ 25/ 50 MHz Operation (Inches)	Maximum Board Trace Length from the Stratix V Device to the Serial Configuration Device for 100 MHz Operation (Inches)	Maximum Board Load (pF)
DCLK	10	6	15
DATA[3..0]	10	6	30
nCSO	10	6	30

AS Configuration Timing

Figure 9–10 shows the timing waveform for the AS ×1 mode and AS ×4 mode configuration timing.

Figure 9–10. AS Configuration Timing



Notes to Figure 9–10:

(1) The AS scheme supports standard and fast POR delay (t_{POR}). For t_{POR} delay information, refer to "POR Delay Specification" on page 9–2.

(2) If you are using AS ×4 mode, this signal represents the AS_DATA[3..0] and EPCQ sends in 4-bits of data for each DCLK cycle.

(3) The initialization clock can be from internal oscillator or CLKUSR pin.

(4) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–11 lists the timing parameters for AS ×1 and AS ×4 configurations in Stratix V devices.

Table 9–11. AS Timing Parameters for AS ×1 and AS ×4 Configurations in Stratix V Devices *(Note 1), (2), (3)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CO}	DCLK falling edge to AS_DATA0/ASDO output	—	4	µs
t_{SU}	Data setup time before rising edge on DCLK	1.5	—	ns
t_H	Data hold time after rising edge on DCLK	0	—	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	175	437	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} + (17,408 × CLKUSR period)	—	—

Notes to Table 9–11:

(1) This information is preliminary.

(2) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for initializing the device.

(3) t_{CF2CD}, t_{CF2ST0}, t_{CFG}, t_{STATUS}, and t_{CF2ST1} timing parameters are identical to the timing parameters for PS mode listed in Table 9–12 on page 9–30.

(4) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on this pin, refer to "Initialization" on page 9–5.

Estimating the Active Serial Configuration Time

The AS configuration time is dominated by the time it takes to transfer data from the EPCS to the Stratix V device. This serial interface is clocked by the Stratix V DCLK.

You can estimate the minimum AS ×1 mode configuration time by using the following equation:

.rbf Size × (minimum DCLK period / 1 bit per DCLK cycle) = estimated minimum configuration time.

You can estimate the minimum AS ×4 mode configuration time by using the following equation:

.rbf Size × (minimum DCLK period / 4 bits per DCLK cycle) = estimated minimum configuration time.

Enabling compression reduces the amount of configuration data that is transmitted to the Stratix V device, which also reduces the configuration time. Your configuration time is reduced based on the compression ratio. The compression ratio varies based on the design.

Programming EPCS and EPCQ

EPCS and EPCQ are non-volatile, flash-memory-based devices. You can program these devices in-system using a USB-Blaster™, EthernetBlaster, or ByteBlaster™ II download cable. Alternatively, you can program EPCS or EPCQ device using a microprocessor with the SRunner software driver.

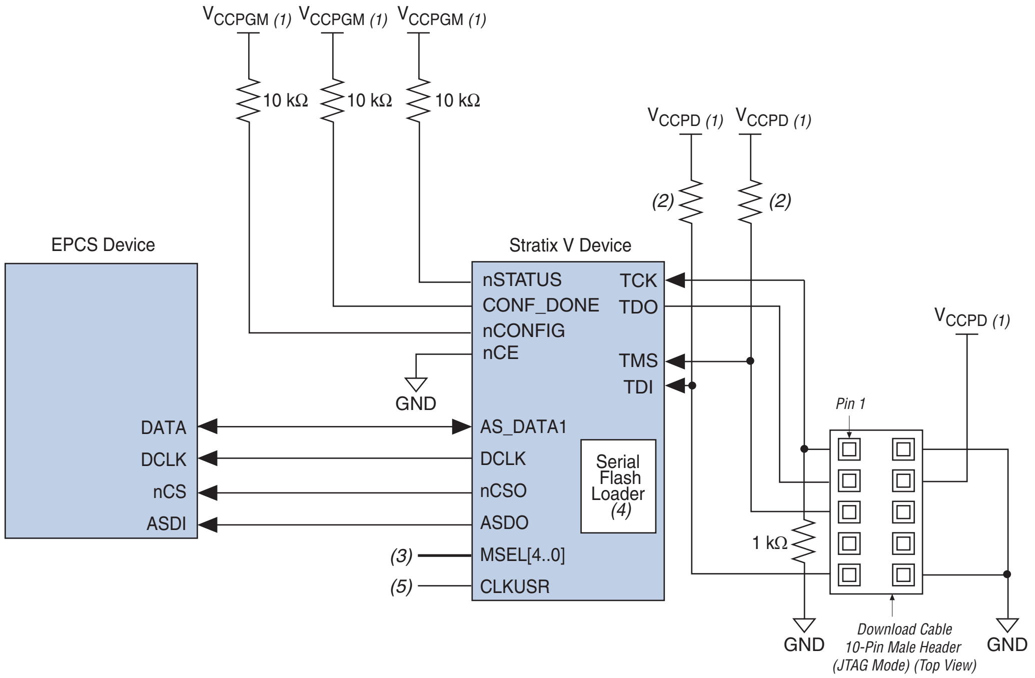
☞ If you are not using Quartus II software or SRunner software for EPCQ 256 programming, put your EPCQ 256 device into four-byte addressing mode before you program and configure your device.

For more information about SRunner software driver, refer to *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming*.

In-system programming offers you an option to program the EPCS or EPCQ device either using an AS programming interface or a JTAG interface. Using the AS programming interface, the configuration data is programmed into the EPCS by the Quartus II software or any supported third-party software. Using the JTAG interface, an Altera IP called Serial Flash Loader (SFL) must be downloaded into the Stratix V device to form a bridge between the JTAG interface and the EPCS or EPCQ device. This allows the EPCS or the EPCQ device to be programmed directly using the JTAG interface.

Figure 9–11 shows the connection setup when programming the EPCS device using the JTAG interface.

Figure 9–11. Connection Setup for Programming the EPCS Device Using the JTAG Interface



Notes to Figure 9–11:

(1) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

(2) Resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

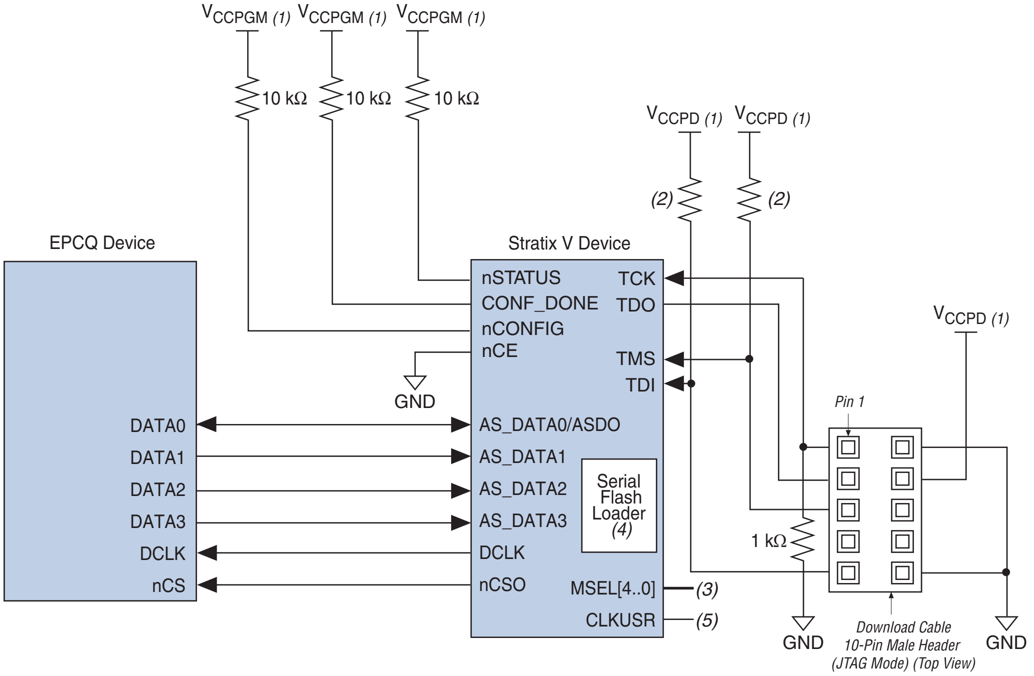
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) Instantiate SFL in your design to form a bridge between the EPCS device and the Stratix V device. For more information about SFL, refer to *AN 370: Using the Serial Flash Loader with the Quartus II Software*.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–12 shows the connection setup when programming the EPCQ device using the JTAG interface.

Figure 9–12. Connection Setup for Programming the EPCQ Device Using the JTAG Interface



Notes to Figure 9–12:

(1) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

(2) Resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

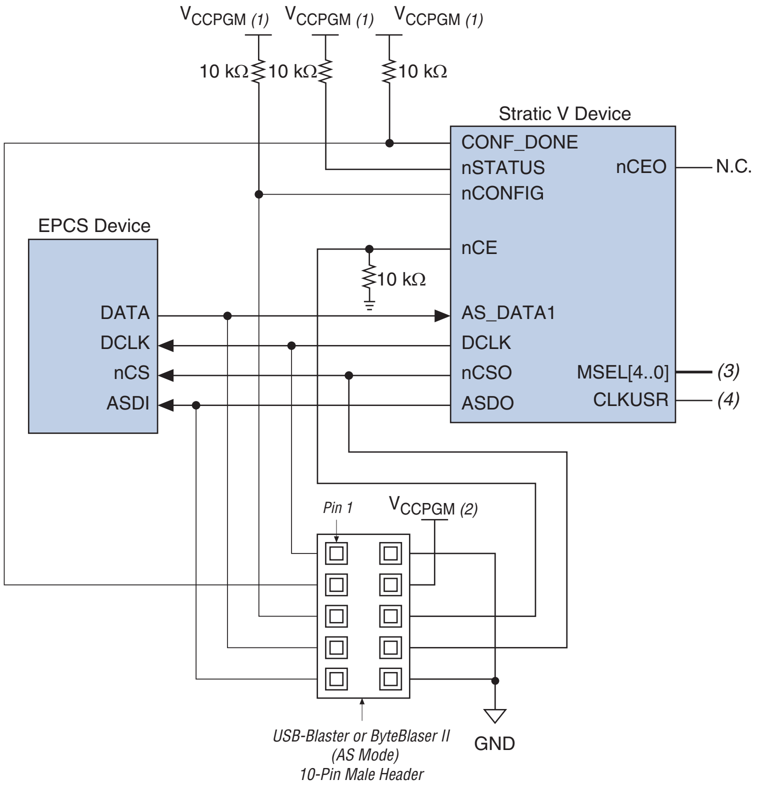
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) Instantiate SFL in your design to form a bridge between the EPCS device and the Stratix V device. For more information about SFL, refer to *AN 370: Using the Serial Flash Loader with the Quartus II Software.*

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–13 shows the connection setup when programming the EPCS device using the AS interface.

Figure 9–13. Connection Setup for Programming the EPCS Device Using the AS Interface



Notes to Figure 9–13:

(1) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

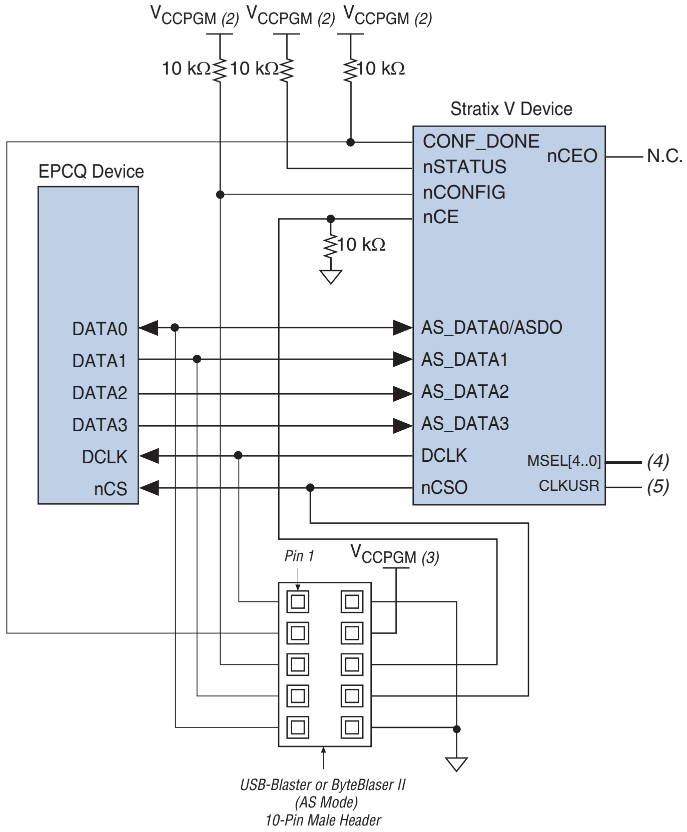
(2) Power up the USB-ByteBlaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–14 shows the connection setup when programming the EPCQ device using the AS interface.

Figure 9–14. Connection Setup for Programming the EPCQ Device Using the AS Interface *(Note 1)*



Notes to Figure 9–14:

(1) Using the AS header, the programmer transmits the operation commands and the configuration bits to the EPCQ device serially on DATA0.This is equivalent to the programming operation for the EPCS device as shown in Figure 9–14.

(2) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

(3) Power up the USB-ByteBlaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

During EPCS and EPCQ programming, the download cable disables the device access to the AS interface by driving the nCE pin high. The nCONFIG line is also pulled low to hold the Stratix V device in reset stage. After programming completes, the download cable releases nCE and nCONFIG, allowing the pull-down and pull-up resistors to drive the pin to GND and V_{CCPGM}, respectively.

☞ During EPCQ programming using the download cable, DATA0 carries the programming data, operation command, and address information from the download cable into the EPCQ device. During EPCQ verification using the download cable, DATA1 carries the programming data back to the download cable.

Passive Serial Configuration

You can perform PS configuration of Stratix V devices using an external host such as a MAX II device, microprocessor, or a host PC. Therefore, the design that controls the configuration stages, such as fetching the data from flash memory and sending it to the device, must be stored in the external host.

The Parallel Flash Loader (PFL) feature in MAX II devices provide an efficient method to program CFI flash memory devices through the JTAG interface. PFL also acts as a controller to read configuration data from the flash memory device and configures the Stratix V device. PFL supports both the PS and FPP configuration schemes.

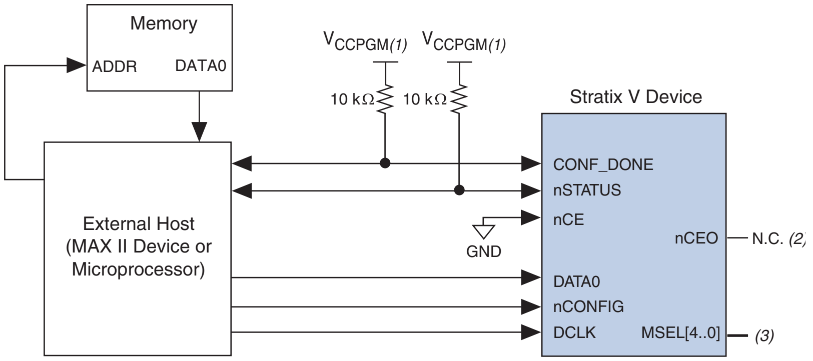
For more information about the PFL, refer to the *Parallel Flash Loader Megafunction User Guide*.

PS Configuration Using a MAX II Device or Microprocessor

The external host (a MAX II device or microprocessor) reads configuration data from storage devices, such as flash memory, and transfers it to Stratix V devices. You can store the configuration data in **.pof**, **.rbf**, **.hex**, or **.ttf** format. 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** file contains the byte sequence 02 1B EE 01 FA, the serial data transmitted to the device must be 0100-0000 1101-1000 0111-0111 1000-0000 0101-1111.

Figure 9–15 shows the configuration interface connections between a Stratix V device and a MAX II device for single device configuration.

Figure 9–15. Single Device PS Configuration Using an External Host



Notes to Figure 9–15:

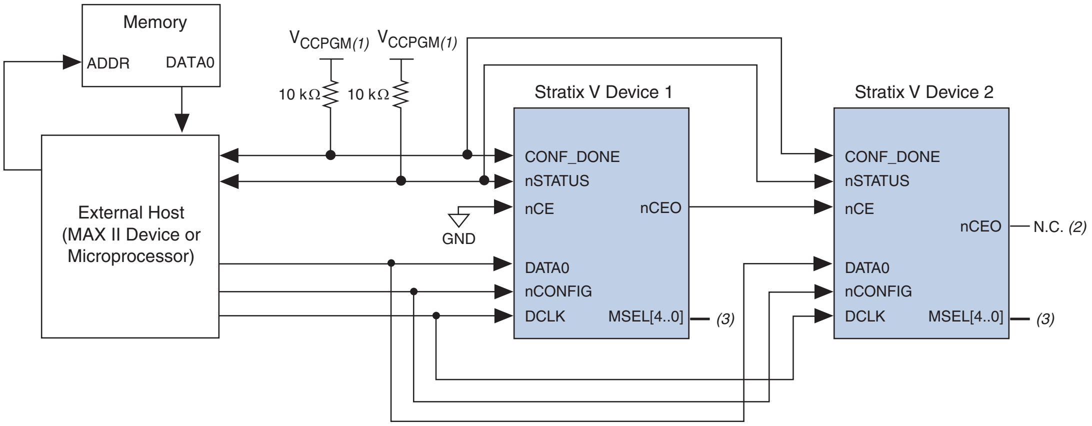
(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the `nCEO` pin unconnected or use it as a user I/O pin when it does not feed another device's `nCE` pin.

(3) The `MSEL` pin settings vary for different configuration voltage standards and POR delays. To connect `MSEL`, refer to Table 9–4 on page 9–7.

Figure 9–16 shows the PS multi-device configuration using an external host when all devices in the chain receive different sets of configuration data (multiple **.sof**s).

Figure 9–16. PS Multi-device Configuration when Both Devices Receive Different Sets of Configuration Data



Notes to Figure 9–16:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V 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 powering up all the configuration system I/Os with V_{CCPGM}.

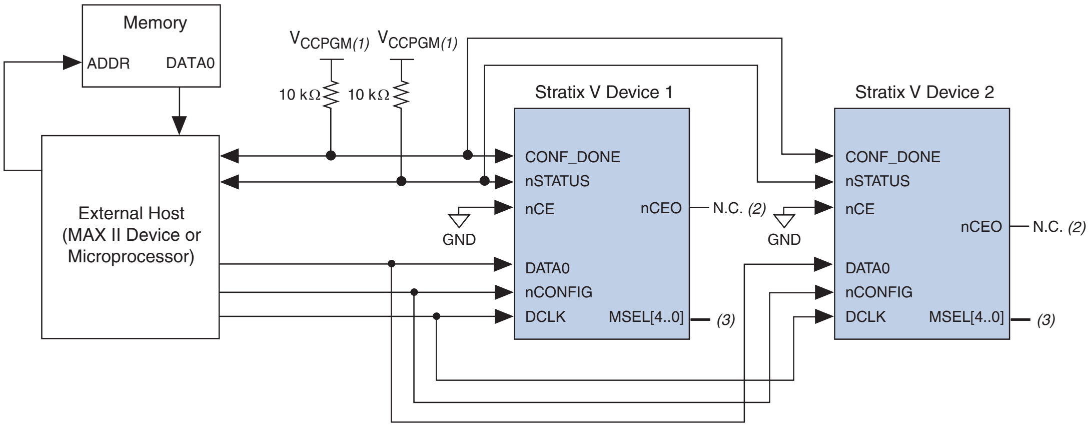
(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

In Figure 9–16, 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 in one clock cycle; therefore, the transfer of data to the second device is transparent to the external host.

Figure 9–17 shows the PS multi-device configuration when all devices receive the same set of configuration data (single **.sof**).

Figure 9–17. PS Multi-device Configuration When Both Devices Receive the Same Set of Configuration Data



Notes to Figure 9–17:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V 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 powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

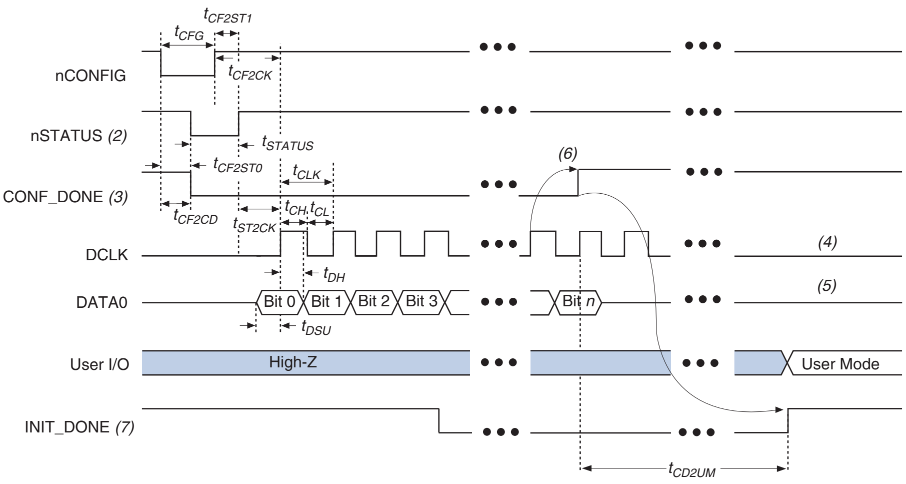
In Figure 9–17, because both nCE pins are tied to GND, both devices in the chain begin and complete the configuration and enter user mode at the same time.

To configure the PS multi-device with a single **.sof**, as shown in Figure 9–17, all Stratix V devices in the chain must be in the same package and density.

PS Configuration Timing

Figure 9–18 shows the timing waveform for a PS configuration when using a MAX II device or microprocessor as an external host.

Figure 9–18. PS Configuration Timing Waveform ***(Note 1)***



Notes to Figure 9–18:

(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 V 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) DATA0 is available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings in the **Device and Pins Option**.

(6) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high after the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(7) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–12 lists the PS configuration timing parameters for Stratix V devices.

Table 9–12. PS Timing Parameters for Stratix V Devices ***(Note 1)*** **(Part 1 of 2)**

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(3)*	µs

Table 9–12. PS Timing Parameters for Stratix V Devices *(Note 1)* **(Part 2 of 2)**

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	$0.45 \times 1/f_{MAX}$	—	s
t_{CL}	DCLK low time	$0.45 \times 1/f_{MAX}$	—	s
t_{CLK}	DCLK period	$1/f_{MAX}$	—	s
f_{MAX}	DCLK frequency	—	125	MHz
t_R	Input rise time	—	40	ns
t_F	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	175	437	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} + (17,408 × CLKUSR period) *(5)*	—	—

Notes to Table 9–12:

(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) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.

(4) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for initializing the device.

(5) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

☞ Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed configuration data in the PS configuration scheme.

PS Configuration Using a Download Cable

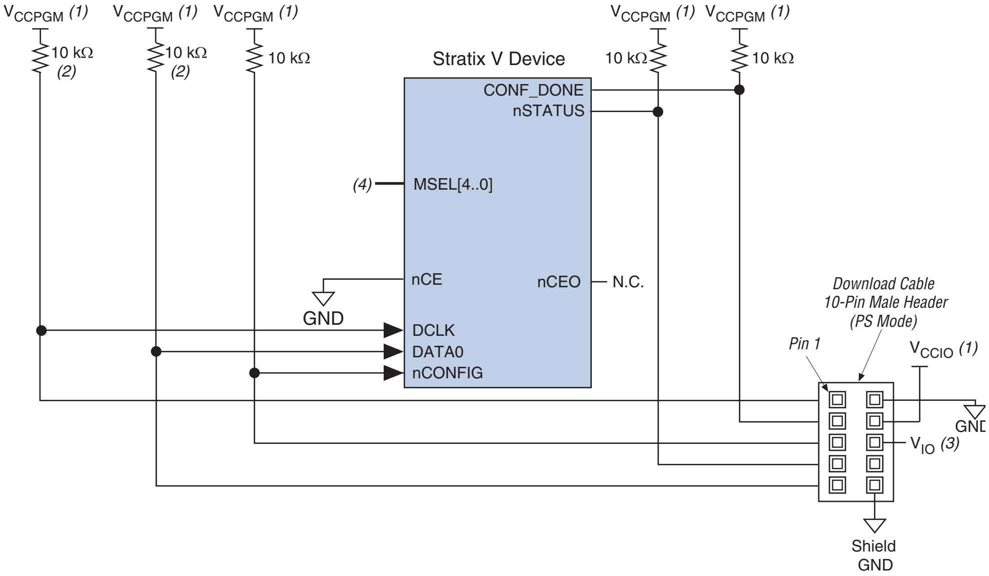
☞ In this section, the generic term "download cable" includes the Altera USB-Blaster universal serial bus (USB) port download cable, ByteBlaster II parallel port download cable, and EthernetBlaster download cable.

In a PS configuration with a download cable, a PC acts as a host to transfer data from a storage device to the Stratix V device using the download cable. During configuration, the programming hardware or download cable 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.

☞ If you turn on the CLKUSR option during PS configuration using a download cable and the Quartus II programmer, you do not have to provide a clock on the CLKUSR pin to initialize your device.

Figure 9–19 shows a PS configuration for Stratix V devices using an Altera download cable.

Figure 9–19. PS Configuration Using an Altera Download Cable



Notes to Figure 9–19:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCIO}) as the USB-Blaster, ByteBlaster II, 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 MAX II device or microprocessor, you do not need the pull-up resistors on DATA0 and DCLK.

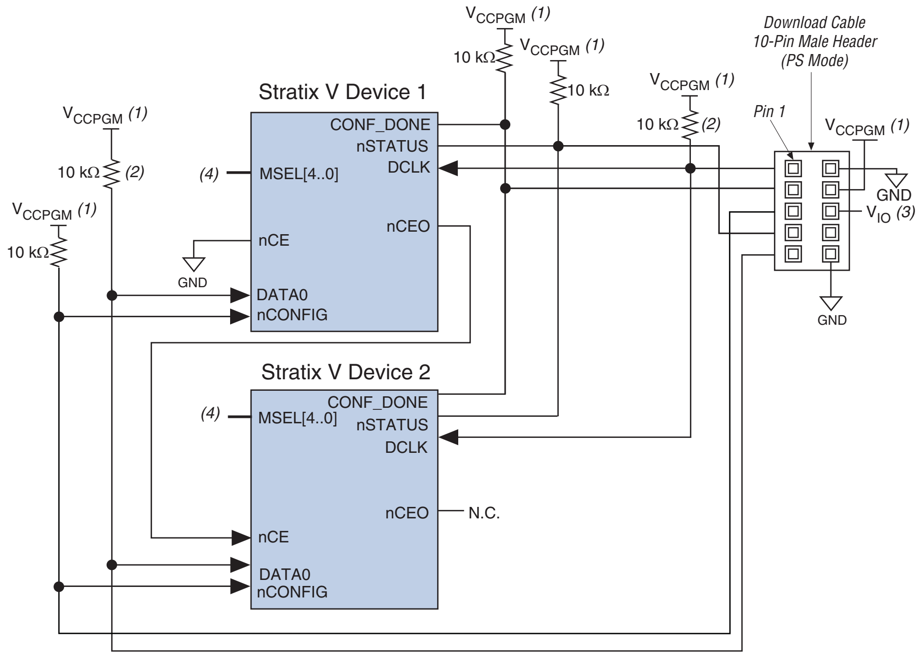
(3) In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when you use for AS programming; otherwise, it is a no connect.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL[4..0], refer to Table 9–4 on page 9–7.

Multi-Device PS Configuration Using Download Cable

You can use a download cable to configure multiple Stratix V devices as shown in Figure 9–20.

Figure 9–20. Multi-Device PS Configuration Using an Altera Download Cable



Notes to Figure 9–20:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCIO}) as the USB-Blaster, ByteBlaster II, 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) In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when you use it for AS programming; otherwise, it is a no connect.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL[4..0], refer to Table 9–4 on page 9–7.

In Figure 9–20, after the first device completes configuration, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The nCONFIG, nSTATUS, DCLK, DATA0 , and CONF_DONE pins are connected to every device in the chain. As all the CONF_DONE and nSTATUS pins are tied together, all devices initialize and enter user mode at the same time. If any device detects an error, configuration stops for the entire chain and you must reconfigure all the devices. 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.

JTAG Configuration

You can use the same JTAG interface specifically developed for boundary-scan test (BST) to shift the configuration data into the device. The Quartus II software automatically generates a **.sof** that you can use for JTAG configuration with a download cable in the Quartus II software programmer.

For more information about JTAG BST and the commands available using Stratix V devices, refer to the following documents:

- *JTAG Boundary-Scan Testing in Stratix V Devices* chapter
- Programming Support for Jam STAPL Language

Stratix V devices are designed such that JTAG instructions have precedence over any device configuration modes. JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration of Stratix V devices during a PS configuration, the PS configuration is terminated and the JTAG configuration begins. All user I/O pins are tri-stated during the JTAG configuration.

You cannot use the Stratix V decompression or design security features if you are configuring your Stratix V device using JTAG-based configuration.

For more information about TDI, TDO, TMS, and TCK, refer to "Device Configuration Pins" on page 9–38.

For instructions to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing in Stratix V Devices* chapter.

To configure a single device in a JTAG chain, the programming software places all the 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 after completion by checking the state of CONF_DONE through the JTAG port.

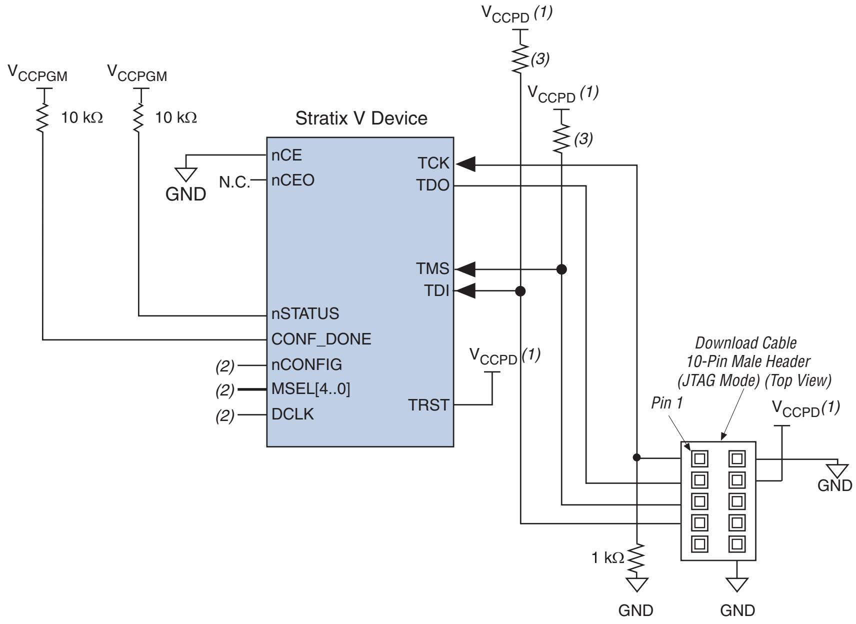
If CONF_DONE is not high, the Quartus II software indicates that configuration has failed. If CONF_DONE is high, the software indicates that configuration was successful. After the configuration data is transmitted serially using the JTAG TDI port, the TCK port is clocked an additional 1,222 cycles to perform device initialization.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Stratix V devices do not affect JTAG boundary-scan or programming operations.

You can generate a JAM File (**.jam**) or Jam-byte Code (**.jbc**) to be used with other third party programmer tools. Alternatively, you can use JRunner with **.rbf** to program your device.

Figure 9–21 shows the JTAG configuration of a single Stratix V device.

Figure 9–21. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 9–21:

(1) Connect the pull-up resistor V_{CCPD}. For more information, refer to V_{CCPD} requirement in "V_{CCPD} Pin" on page 9–3.

(2) If you only use JTAG configuration, connect `nCONFIG` to V_{CCPGM} and `MSEL[4..0]` to GND. Pull `DCLK` either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, connect `MSEL[4..0]`, `nCONFIG` and `DCLK` based on the selected configuration scheme.

(3) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

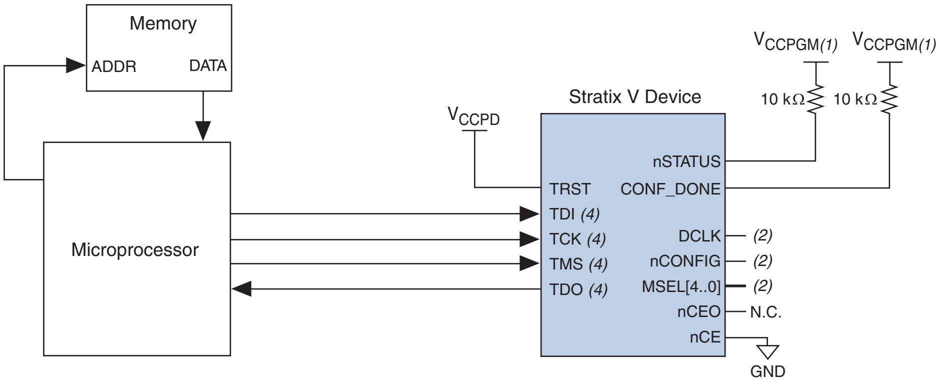
(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

Alternatively, you can use a microprocessor to program the device through the JTAG interface. You can use JRunner as your software driver.

For more information about JRunner, refer to *AN 414: The JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration*.

Figure 9–22 shows a JTAG configuration of a Stratix V device using a microprocessor.

Figure 9–22. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 9–22:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all Stratix V devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device.

(2) If you only use the JTAG configuration, connect nCONFIG to V_{CCPGM} and MSEL[4..0] to GND. Pull DCLK either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, set the MSEL[4..0] and tie nCONFIG and DCLK based on the selected configuration scheme.

(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.

CONFIG_IO Instruction

The CONFIG_IO instruction allows you to configure I/O buffers using the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix V device or waiting for a configuration device to complete configuration. After configuration is interrupted and JTAG testing is complete, you must reconfigure the part using the JTAG interface or if you support FPP, PS, or AS on your board, reconfigure the device by externally pulsing nCONFIG low. Alternatively, you can pulse nCONFIG low through the same JTAG interface using the PULSE_NCONFIG JTAG instruction.

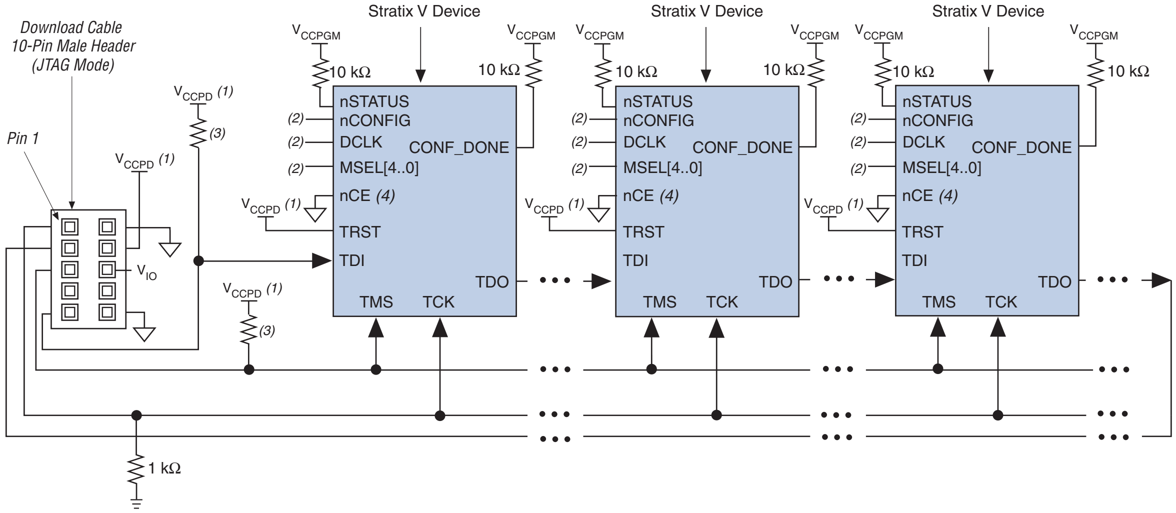
☞ All JTAG instructions (except BYPASS, IDCODE, and SAMPLE) can be issued by first interrupting the configuration and reprogramming the I/O pins using the CONFIG_IO instruction.

Multi-Device JTAG Configuration

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. You can place other Altera devices that have JTAG support in the same JTAG chain for device programming.

JTAG-chain device programming is ideal when the system contains multiple devices or when testing your system using JTAG BST circuitry. Figure 9–23 shows a multi-device JTAG configuration.

Figure 9–23. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 9–23:

(1) Connect the pull-up resistor V_{CCPD}. For more information, refer to V_{CCPD} requirement in "V_{CCPD} Pin" on page 9–3.

(2) If you only use JTAG configuration, connect `nCONFIG` to V_{CCPGM} and `MSEL[4..0]` to GND. Pull `DCLK` either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, connect the `MSEL[4..0]`, `nCONFIG`, and `DCLK` based on the selected configuration scheme.

(3) The resistor value can vary from 1kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

If you want to use the JTAG multi-device configuration in conjunction with other schemes, such as a FPP, PS, or AS, tie `CONF_DONE`, `nSTATUS`, and `nCONFIG` together as recommended in the FPP, PS, or AS multi-device configuration schemes. Ensure that the JTAG chain is the same order as the multi-device FPP, PS, or AS configuration chain.

If you only use JTAG configuration, Altera recommends connecting the circuitry as shown in Figure 9–22, where each of the `CONF_DONE` and `nSTATUS` signals are isolated to enable each device to enter user mode individually.

For more information about combining the JTAG configuration with other configuration schemes, refer to the *Combining Different Configuration Schemes* chapter in volume 2 of the *Configuration Handbook*.

For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the Jam player, visit the Altera website.

For more information about how to use the USB-Blaster, ByteBlaster II, or EthernetBlaster cables, refer to the following user guides:

- *USB-Blaster Download Cable User Guide*
- *ByteBlaster II Download Cable User Guide*
- *Ethernet Blaster Communications Cable User Guide*

Device Configuration Pins

Table 9–13 and Table 9–14 list the connections and functionality of all the configuration-related pins on the Stratix V devices. Table 9–13 lists the Stratix V configuration pins and their power supply.

Table 9–13. Configuration Pin Summary for Stratix V Devices (Part 1 of 2)

Description	Input/Output	User Mode	Powered By	Configuration Scheme
TDI	Input	—	V_{CCPD}	JTAG
TMS	Input	—	V_{CCPD}	JTAG
TCK	Input	—	V_{CCPD}	JTAG
TRST	Input	—	V_{CCPD}	JTAG
TDO	Output	—	V_{CCPD}	JTAG
CLKUSR	Input	I/O *(1)*	V_{CCPGM}/V_{CCIO} *(4)*	All schemes
CRC_ERROR	Output	I/O *(1)*	Pull-up	Optional, all schemes
CONF_DONE	Bidirectional	—	V_{CCPGM}/Pull-up	All schemes
DATA0	Bidirectional	I/O *(2)*	V_{CCPGM}/V_{CCIO} *(4)*	FPP, PS
DATA[31..1]	Bidirectional	I/O *(2)*	V_{CCPGM}/V_{CCIO} *(4)*	FPP
DCLK	Input	—	V_{CCPGM}	FPP, PS
	Output	—	V_{CCPGM}	AS
DEV_OE	Input	I/O *(1)*	V_{CCPGM}/V_{CCIO} *(4)*	Optional, all schemes
DEV_CLRn	Input	I/O *(1)*	V_{CCPGM}/V_{CCIO} *(4)*	Optional, all schemes
INIT_DONE	Output	I/O *(1)*	Pull-up	Optional, all schemes
MSEL[4..0]	Input	—	V_{CCPGM}	All schemes
nSTATUS	Bidirectional	—	V_{CCPGM}	All schemes
nCE	Input	—	V_{CCPGM}	All schemes
nCEO	Output	I/O *(3)*	Pull-up	All schemes
nCONFIG	Input	—	V_{CCPGM}	All schemes
nCSO	Output	—	V_{CCPGM}	AS
nIO_PULLUP	Input	—	V_{CC} *(5)*	All schemes
AS_DATA0/ASDO	Bidirectional	—	V_{CCPGM}	AS

Table 9–13. Configuration Pin Summary for Stratix V Devices (Part 2 of 2)

Description	Input/Output	User Mode	Powered By	Configuration Scheme
AS_DATA[3..1]	Bidirectional	—	V_{CCPGM}	AS

Notes to Table 9–13:

(1) This is a dual-purpose pin. This pin is available as an I/O if the associated option that enables this pin is turned off from the **Configuration** panel in the **Device and Pins Option** settings. For example, the DEV_OE is available as a user I/O if the **Enable device-wide output enable** option is turned off.

(2) This is a dual-purpose pin. The state of this pin in user mode depends on the **Dual-purpose Pins** settings in the **Device and Pins Option**.

(3) This pin is available as an I/O if this pin is not feeding the next device's nCE in a multi-device configuration. To use this pin to feed the next device's nCE in a multi-device chain, turn on **Enable INIT_DONE output** option under **Device and Pins Option**, **General** panel in the Quartus II Software.

(4) This pin is powered up by V_{CCPGM} during configuration. It is powered up by V_{CCIO} of the bank in which the pin resides if it is used as a regular I/O in user mode.

(5) Although nIO_PULLUP is powered up by V_{CC}, Altera recommends connecting this pin to V_{CCPGM} or GND directly without using a pull-up or pull-down resistor.

Table 9–14 lists the configuration pin descriptions.

Table 9–14. Configuration Pins Description (Part 1 of 3)

Pin Name	Description
TDI *(1)*	Dedicated test data input. Serial input pin for instructions as well as test and programming data. Data is shifted on the rising edge of TCK. This pin has an internal 25-kΩ pull-up that is always active.
TMS *(1)*	Dedicated 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 in the state machine occur on the falling edge of TCK after the signal is applied to TMS. This pin has an internal 25-kΩ pull-up that is always active.
TCK *(1)*	Dedicated test clock input. Clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. It is expected that the clock input waveform have a nominal 50% duty cycle. This pin has an internal 25-kΩ pull-down that is always active.
TRST *(1)*	Dedicated test reset input. Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to the IEEE Std. 1149.1 standard. Connecting this pin low disables the JTAG circuitry. This pin has an internal 25-kΩ pull-up that is always active.
TDO *(1)*	Dedicated 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. This pin is tri-stated if the data is not being shifted out of the device.
CLKUSR	Optional user-supplied clock input. It synchronizes the initialization of one or more devices. Enable this pin by turning on the **Enable user-supplied start-up clock (CLKUSR)** option under **Device and Pins Option**, **Configuration** panel in the Quartus II software.

Table 9–14. Configuration Pins Description (Part 2 of 3)

Pin Name	Description
CRC_ERROR	Optional output pin. Signals that the device has detected a cyclical redundancy check (CRC) error during user mode operation. this pin is an open-drain output pin by default and requires a 10 kΩ pull-up resistor. To use this pin as regular output, turn-off the **Enable Open-drain on CRC_ERROR pin** in **Device and Pins Option**, **Error Detection CRC** panel in the Quartus II software. The target device drives this pin low if there is no CRC error in user mode operation. As an open-drain output, if a CRC error occurs, the device releases the pin which is then pulled high by the external pull-up resistor. Enable this pin by turning on **Enable CRC error detection on CRC_ERROR pin** option in the Quartus II software. For more information about the CRC_ERROR pin, refer to *SEU Mitigation in Stratix V Devices* chapter.
CONF_DONE	Dedicated open-drain bidirectional pin. 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 the CONF_DONE pin, which is then pulled high by the external pull-up resistor. The target device then reads the CONF_DONE pin status to ensure that the CONF_DONE is at logic high. After it is sensed high, the target device initializes and enters user mode. Driving CONF_DONE low after initialization completes does not affect the configured device.
DATA0 *(3)*	Dual-purpose data input pin. The data received on DATA0 is synchronized to DCLK. After configuration completes, this pin is available as a user I/O pin.
DATA[31..1] *(3)*	Dual-purpose data input pins. If you are using FPP ×16 or FPP ×32, only a subset of these pins are required for configuration. The pins that are not required for configuration, you can use them as regular I/Os. During configuration, byte-wide, or word-wide data is received on these pins. The data received on DATA[31..1] are synchronized to the DCLK.
DCLK	Dedicated bidirectional clock pin. In the PS and FPP configurations, 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. After configuration completes, drive DCLK high or low, whichever is more convenient. In the AS mode, DCLK is an output clock to clock the EPCS or EPCQ devices. Data is latched into the device on the falling edge of the DCLK. After AS configuration completes, this pin is tri-stated with a weak pull-up resistor. Toggling this pin after configuration does not affect the configured device.
DEV_OE *(2)*	Optional input pin that allows you to override all tri-states on the device. When this pin is driven low, all the I/O pins are tri-stated. When this pin is driven high, all the 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 *(2)*	Optional input pin that allows you to override all clears on all the device registers. When this pin is driven low, all the registers are cleared. When this pin is driven high, all the registers behave as programmed. This pin is enabled by turning on the **Enable device-wide reset (DEV_CLRn)** option in the Quartus II software.
INIT_DONE *(2)*	Optional output pin. Signals when the device has initialized and is in user mode. During the reset stage, after the device exits POR, and during the beginning of the configuration, the INIT_DONE pin is tri-stated and pulled high due to an external 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 completes, 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.

Table 9–14. Configuration Pins Description (Part 3 of 3)

Pin Name	Description
MSEL[4..0]	Dedicated input pins. Five-bit configuration input that sets the Stratix V device configuration scheme. For the appropriate connections, refer to Table 9–4 on page 9–7. The MSEL[4..0] pins have internal 5-kΩ pull-down resistors that are always active.
nSTATUS	Dedicated open-drain bidirectional pin. The device drives nSTATUS low immediately after power-up and releases it after the device exits POR. During user mode and regular configuration, this pin is pulled high by an external 10-kΩ resistor. During configuration, the device drives this pin low to indicate an error during configuration. If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. This mechanism is used during multi-device configuration setup. If one of the devices in the chain has an error and pulls its nSTATUS pin low, it resets the entire chain. Driving nSTATUS low after configuration and initialization completes does not affect the configured device.
nCE	Dedicated active-low chip enable input pin. Driving this pin low allows configuration. Drives the nCE pin low during configuration, initialization, and user mode for all single-device configurations. For a multi-device configuration, connect the nCE pin to GND or to the nCEO of the previous device in the chain based on the recommendation in the respective configuration setup diagram.
nCEO *(3)*	Dual-purpose open-drain output pin. This pin drives low when device configuration completes. To use this pin to feed the next device's nCE pin in a multi-device chain, turn on **Enable INIT_DONE output** option under **Device and Pins Option**, **General** panel in the Quartus II Software. In a single-device configuration, this pin can be used as a regular I/O. In multi-device configuration, if this pin is not feeding the nCE of the next device, you can use it as a regular I/O.
nCONFIG	Dedicated input pin. A low pulse on this pin during configuration and user mode causes the device to enter a reset state and tri-states all the I/O pins. A low-to-high logic starts a reconfiguration. During JTAG programming, the nCONFIG status is ignored.
nCSO	Dedicated output pin. Drives the control signal from the Stratix V device to the EPCS and EPCQ devices in AS mode. After AS configuration completes, these pins are tri-stated with a weak pull-up resistor.
nIO_PULLUP	Dedicated input pin. This input pin enables or disables the internal pull-up resistors on the user I/O pins and dual purpose I/O pins (DATA[31..0], CLKUSR, INIT_DONE, DEV_OE, and DEV_CLRN). A logic high turns off the weak internal pull-up resistors, while a logic low turns them on. This pin has an internal 5-kΩ pull-down resistor that is always active.
AS_DATA0/ASDO	Dedicated bidirectional data pin. In AS ×1 and AS ×4 configurations, ASDO is used to send the operation command and addresses to the EPCS or EPCQ devices. During an AS ×4 configuration, the data is received on an AS_DATA0 and is synchronized to DCLK. After AS configuration completes, this pin is tri-stated with a weak pull-up resistor.
AS_DATA[3..1]	Dedicated bidirectional data pins. During an AS configuration, the data is received on these pins and is synchronized to DCLK. After AS configuration completes, these pins are tri-stated with a weak pull-up resistor.

Notes to Table 9–14:

(1) If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to logic high. For instructions to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing* chapter.

(2) This is a dual-purpose pin. This pin is available as an I/O if the associated option that enables this pin is turned off from the **Configuration** panel in the **Device and Pins Option** settings. For example, the DEV_OE is available as a user I/O if the **Enable device-wide output enable** option is turned off.

(3) This is a dual-purpose pin. The state of this pin in the user mode depends on **Dual-purpose Pins** settings in the **Device and Pins Option** settings.

Configuration Data Decompression

Stratix V devices support configuration data decompression, which saves configuration memory space and may shorten the configuration time. This feature allows you to store compressed configuration data in the configuration or other memory devices and transmit this compressed data to the Stratix V devices. During configuration, the Stratix V device decompresses the data in real time and programs its SRAM cells. The data decompression is done on-the-fly during configuration and does not require an additional processing time.

Preliminary data indicates that compression typically reduces the configuration data size by 30 to 55% based on the designs used. This reduces the storage requirement capacity for the flash memory. The decompression feature is supported in all configuration schemes except JTAG.

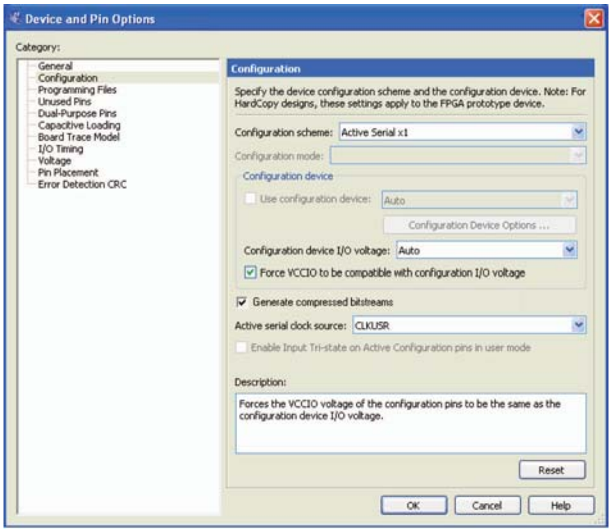
☞ In FPP, enabling the decompression feature requires a different `DCLK`-to-`DATA[]` ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

There are two ways to enable compression for Stratix V data—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, follow these steps:

1. On the Assignments menu, click **Device** to bring up the **Settings** dialog box.
2. After selecting your Stratix V device, open the **Device and Pin Options** dialog box.
3. In the **Configuration settings** panel, turn on the **Generate compressed bitstreams** option (refer to Figure 9–24).

Figure 9–24. Enabling Compression Bitstreams in Compiler Settings for Stratix V



Specify the device configuration scheme and the configuration device. Note: For HardCopy designs, these settings apply to the FPGA prototype device.

Configuration scheme: Active Serial x1

Configuration mode:

Configuration device

□ Use configuration device: Auto

Configuration device I/O voltage: Auto

☑ Force VCCIO to be compatible with configuration I/O voltage

☑ Generate compressed bitstreams

Active serial clock source: CLKUSR

□ Enable Input Tri-state on Active Configuration pins in user mode

Description:

Forces the VCCIO voltage of the configuration pins to be the same as the configuration device I/O voltage.

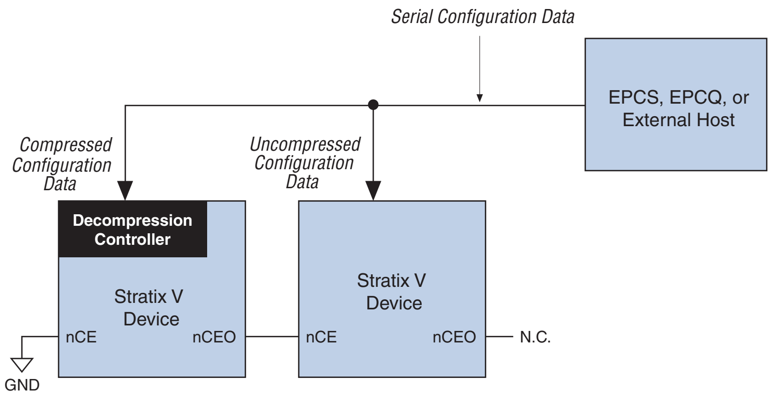
You can also enable compression when creating programming files from the **Convert Programming Files** window. To do this, follow these steps:

1. On the File menu, click **Convert Programming Files**.
2. Select the programming file type (**.pof**, **.sram**, **.hex**, **.hexout**, **.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 V device **.sof**.
6. Select the name of the file you added to the **SOF Data** area and click **Properties**.
7. Check the **Compression** check box.

If you are using a serial configuration scheme, AS ×1 or PS, for multi-device configuration, you can selectively enable the compression feature for each device in the chain. Figure 9–25 shows a chain of two Stratix V devices. The first Stratix V device has compression enabled and therefore receives compressed data from the external host. The second Stratix V device has the compression feature disabled and receives uncompressed data.

☞ For FPP configuration schemes, a combination of compressed and uncompressed configuration in the same multi-device chain is not allowed due to the difference of the `DCLK`-to-`DATA[]` ratio.

Figure 9–25. Compressed and Uncompressed Serial Configuration Data in the Same Configuration File ***(Note 1)***


Serial Configuration Data
EPCS, EPCQ, or External Host
Compressed Configuration Data
Uncompressed Configuration Data
Decompression Controller
Stratix V Device
nCE
nCEO
Stratix V Device
nCE
nCEO
N.C.
GND

Note to Figure 9–25:

(1) The configuration for this setup can be generated from the Convert Programming Files 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 upgrades, 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 V 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 help to avoid system downtime.

Stratix V devices feature dedicated remote system upgrade circuitry. A soft logic (either the Nios® II embedded processor or user logic) implemented in a Stratix V 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 start 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 upgrades are supported in AS configuration schemes with EPCS and EPCQ devices. You can also implement remote system upgrades in conjunction with advanced Stratix V features such as real-time decompression of configuration data and design security using the advanced design security standard (AES) for secure and efficient field upgrades. The largest EPCS and EPCQ device currently supports 128 Mbits and 256 Mbits configuration data respectively.

☞ Remote system upgrades are supported only in single-device configurations.

The remote system upgrade process in Stratix V devices involves the following steps:

1. A Nios II processor (or user logic) implemented in the Stratix V device logic array receives new configuration data from a remote location. The connection to the remote source uses a communication protocol such as TCP/IP, PCI, user datagram protocol (UDP), 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) starts a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any errors that might occur during or after the reconfiguration cycle and provides error status information to the user design.

Figure 9–26 shows these remote system upgrade steps.

Figure 9–26. Functional Diagram of the Stratix V Remote System Upgrade Process

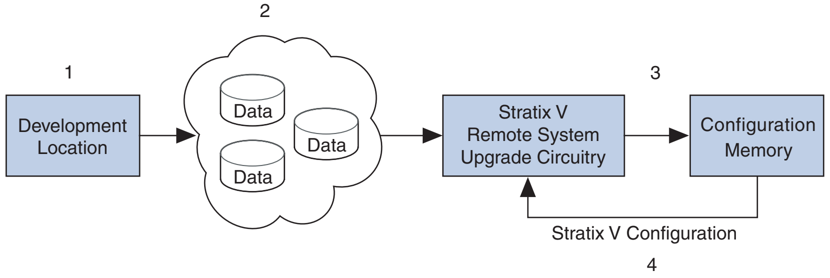
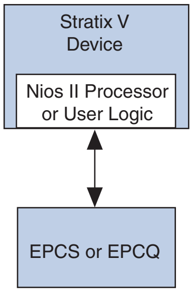


Figure 9–27 shows a block diagram for implementing a remote system upgrade with the Stratix V AS configuration scheme.

Figure 9–27. Remote System Upgrade Block Diagram for Stratix V AS Configuration Scheme *(1)*



Note to Figure 9–27:

(1) You must set the mode select pins (`MSEL[4..0]`) to **AS mode** to use remote system upgrade in your system. The `MSEL` pin settings vary for different POR delays. To connect `MSEL[4..0]`, refer to Table 9–4 on page 9–7.

Configuration Image Types

When performing a remote system upgrade, Stratix V device configuration data are classified as factory configuration images or application configuration images. An image, also referred to as a configuration image, is a design loaded into the Stratix V device that performs certain user-defined functions.

The factory image is a user-defined fall-back, or safe configuration, and is responsible for initiating the reconfiguration to a new image with the dedicated circuitry. Application images implement user-defined functionality in the target Stratix V device. You may also include the default application image functionality in the factory image. Each Stratix V device in your system requires one factory image and one or more application images.

Remote Update Mode

Stratix V remote system upgrade circuitry only supports remote update mode. In remote update mode, Stratix V 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.

When a Stratix V device is first powered up in remote update mode, it loads the factory configuration located at the start address of `PGM[23..0]` = 24'h000000 in the EPCS and EPCQ devices. You must store your factory configuration image at this start address.

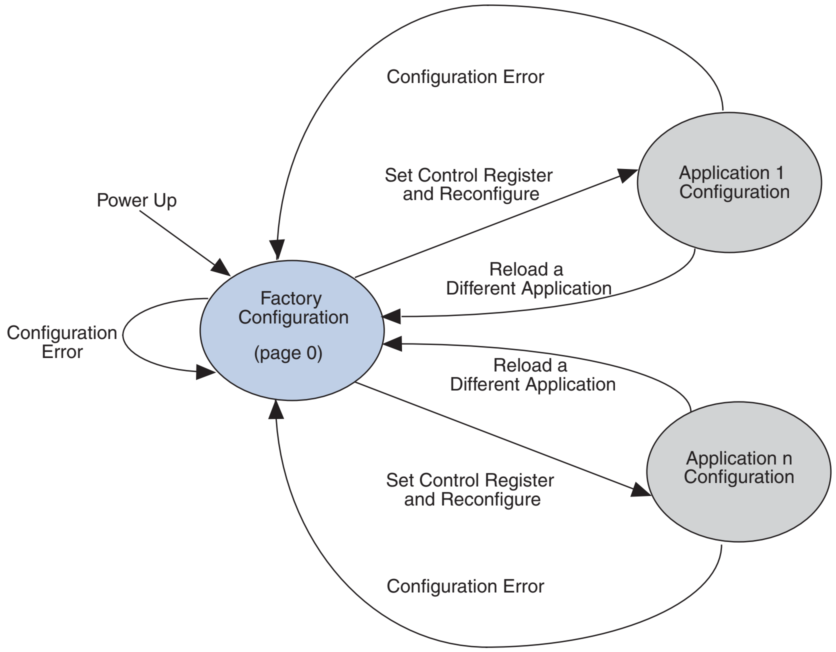
☞ The application image start address can be at any EPCS or EPCQ sector boundary. Altera recommends using different sectors in the EPCS device for two images.

The factory image is user-designed and contains soft logic to perform the following:

- 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 V device
- Enable or disable the user watchdog timer and load its time-out value
- Instruct the dedicated remote system upgrade circuitry to start a reconfiguration cycle

Figure 9–28 shows the transitions between the factory and application configurations in remote update mode.

Figure 9–28. Transitions Between Configurations in Remote Update Mode


Configuration Error
Set Control Register and Reconfigure
Application 1 Configuration
Power Up
Reload a Different Application
Configuration Error
Factory Configuration (page 0)
Reload a Different Application
Application n Configuration
Set Control Register and Reconfigure
Configuration Error

After power up or a configuration error, the factory configuration image is loaded automatically. The system then decides to switch to the application configuration image or to stay in the factory configuration image. After the system decides to switch to an application configuration image, a reconfiguration is initiated through the remote system upgrade circuitry. In the application configuration image, the system may revert back to factory configuration image after the following reconfiguration trigger conditions are met:

- `nSTATUS` driven low externally
- Configuration CRC error
- User watchdog timer time-out
- Core `nCONFIG` signal assertion
- External `nCONFIG` signal assertion

After the factory configuration image is re-loaded, the user-designed factory configuration can read the remote system upgrade status register to determine the reason for the reconfiguration. The factory configuration then takes the appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

Whenever the application configuration image is successfully loaded, the soft logic (Nios II processor or state machine and the remote communication interface) determine when the remote system update is arriving. When this occurs, 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.

Remote System Upgrade Using EPCQ 256

When you are using EPCQ 256, ensure that the application image address granularity is 32’h00000100. The **.rbf** size for your application image is 76,500 bytes longer than the numbers listed in Table 9–5 on page 9–8. You need to take this extra space requirement into consideration when you try to fit multiple application images in the EPCQ 256 device.

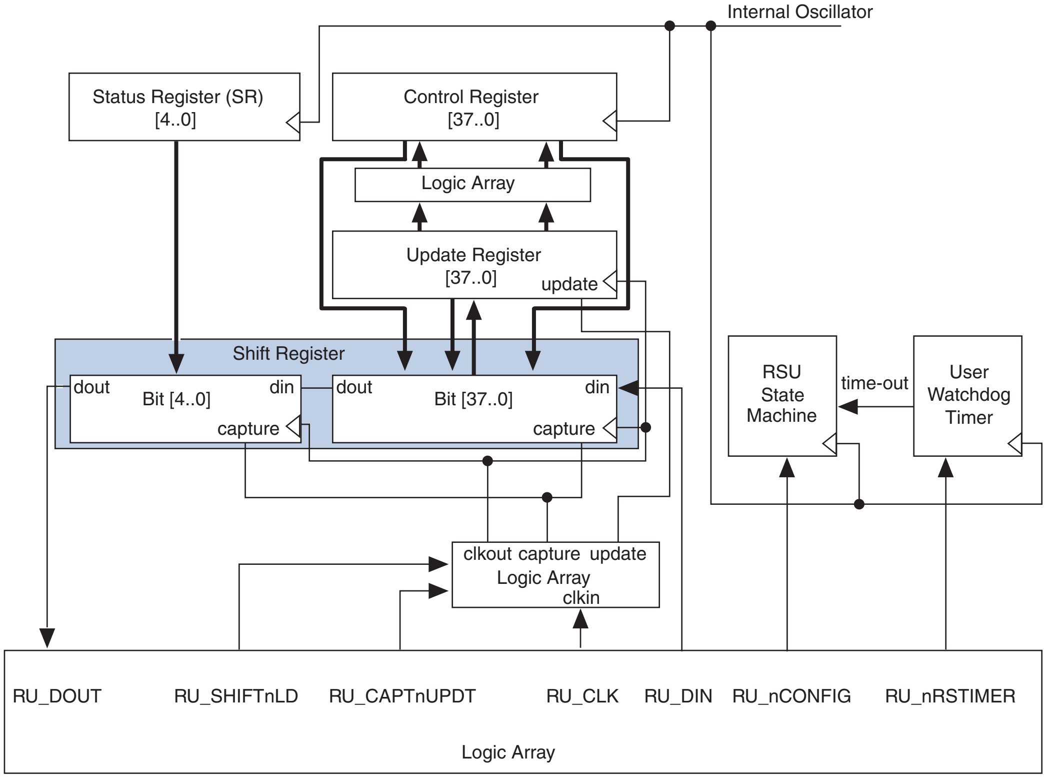
☞ If you are not using Quartus II software or SRunner software for EPCQ 256 programming, put your EPCQ 256 device into four-byte addressing mode before you program and configure your device.

Dedicated Remote System Upgrade Circuitry

This section describes the implementation of the Stratix V dedicated remote system upgrades circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces with the user-defined factory and application configurations implemented in the Stratix V 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 9–29 shows the remote system upgrade circuitry.

Figure 9–29. Remote System Upgrade Circuitry *(Note 1)*



Note to Figure 9–29:

(1) If you are using the ALTREMOTE_UPDATE megafunction, the `RU_DOUT`, `RU_SHIFTnLD`, `RU_CAPTnUPDT`, `RU_CLK`, `RU_DIN`, `RU_nCONFIG`, and `RU_nRSTIMER` signals are internally controlled by the megafunction to perform all the related remote system upgrade operations.

Table 9–15 lists the timing parameter specifications for the remote system upgrade circuitry.

Table 9–15. Remote System Upgrade Circuitry Timing Specifications

Parameter	Minimum	Maximum	Unit
$f_{MAX\_RU\_CLK}$ *(1)*	—	40	MHz
$t_{RU\_nCONFIG}$ *(2)*	250	—	ns
$t_{RU\_nRSTIMER}$ *(3)*	250	—	ns

Notes to Table 9–15:

(1) This clock is user-supplied to the remote system upgrade circuitry. If you are using the ALTREMOTE_UPDATE megafunction, the clock user-supplied to the ALTREMOTE_UPDATE megafunction must meet this specification.

(2) This is equivalent to strobing the reconfiguration input of the ALTREMOTE_UPDATE megafunction high for the minimum timing specification. For more information, refer to "Remote System Upgrade State Machine" on page 9–51.

(3) This is equivalent to strobing the reset_timer input of the ALTREMOTE_UPDATE megafunction high for the minimum timing specification. For more information, refer to "User Watchdog Timer" on page 9–51.

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 9–16 lists these registers.

Table 9–16. Remote System Upgrade Registers

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.
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.

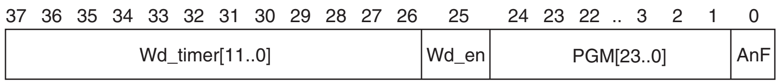
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (RU_CLK).

Control Register

The 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. A factory configuration in remote update mode has write access to this register.

Figure 9–30 shows the control register bit positions. Table 9–17 lists the control register bits. In the figure, the numbers show the bit position of a setting in a register. For example, bit number 25 is the enable bit for the watchdog timer.

Figure 9–30. Remote System Upgrade Control Register



The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix V 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 restricts the access to only read operations and enables the watchdog timer. The factory configuration design must set this bit high (1'b1) when updating the contents of the update register with the application page address and watchdog timer settings.

Table 9–17 lists the remote system upgrade control register contents.

Table 9–17. Remote System Upgrade Control Register Contents

Control Register Bit	Value *(2)*	Definition
AnF *(1)*	1'b0	Application not factory
PGM[23..0]	24'b0×000000	AS configuration start address (StAdd[23..0])
Wd_en	1'b0	User watchdog timer enable bit
Wd_timer[11..0]	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 9–17:

(1) Factory configuration designs must set the AnF bit to **1'b1** before triggering the reconfiguration to application configuration image.

(2) This is the default value of the control register bit after the device exits POR and during reconfiguration back to the factory configuration image after reconfiguration trigger conditions.

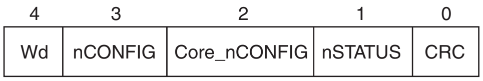
Status Register

The status register specifies the reconfiguration trigger condition. Figure 9–31 shows the status register content. The following list defines each bit:

- Bit 0—CRC error during application configuration
- Bit 1—nSTATUS assertion by an external device due to an error
- Bit 2—Stratix V device logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- Bit 3—external configuration reset (nCONFIG) assertion
- Bit 4—user watchdog timer time-out

Figure 9–31 shows the contents of the status register. The numbers in the figure show the bit positions in a 5-bit register.

Figure 9–31. Remote System Upgrade Status Register ***(Note 1)***



Note to Figure 9–31:

(1) After the device exits POR and powers-up, the status register content is 5'b00000.

Remote System Upgrade State Machine

After power-up, the shift register, control register, and update registers are reset to the values listed in Table 9–16, also known as POR reset values before the factory configuration image is loaded. In the factory configuration image, the user logic writes the AnF bit, page address, and watchdog timer settings for the next application configuration image 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 triggers a reconfiguration to the new application configuration image.

If there is an error during reconfiguration to the new application configuration image, the remote system upgrade state machine directs the system to re-load a factory configuration image. The control and update registers are reset to POR reset values and the status register is updated with the error information. For example, if there is a CRC error during application configuration image configuration, the status register is updated with 5'b00001.

If there is no error during reconfiguration and the application configuration image is successfully loaded, the system stays in the application configuration image until another reconfiguration trigger condition occurs. This can be a core nCONFIG assertion, external nCONFIG assertion, or the watchdog timer time-out error. If this happens, the control register and update registers are reset to POR reset values and the status register is updated with the error information. Consequently, the system proceeds to load the factory configuration image. Based on the status register content, the user logic in the factory configuration image then decides to stay in the factory configuration image or reload a new application reconfiguration image.

☞ Read operations during factory configuration access the contents of the update register. This feature is used by the factory configuration image user logic to verify that the page address and watchdog timer settings are 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 V device. This feature is automatically disabled in the factory configuration image and enabled in the application configuration image. Functional errors must not exist in the factory configuration because they are stored and validated during production and must never be updated remotely.

☞ The user watchdog timer feature is automatically enabled in the application configuration image. If you do not wish to use this feature, disable it during the factory configuration image operation before triggering the reconfiguration to the application configuration image.

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^{17} cycles. The cycle time is based on the frequency of the 12.5-MHz internal oscillator. Table 9–18 lists the operating range of the 12.5-MHz internal oscillator.

Table 9–18. 12.5-MHz Internal Oscillator Specifications ***(Note 1)***

Minimum	Typical	Maximum	Units
5.3	7.9	12.5	MHz

Note to Table 9–18:

(1) These values are preliminary.

The user watchdog timer begins counting after the application configuration enters device user mode. This timer must be periodically 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. This causes the device to reload the factory configuration image and update the status register to reflect the watchdog timer time-out error.

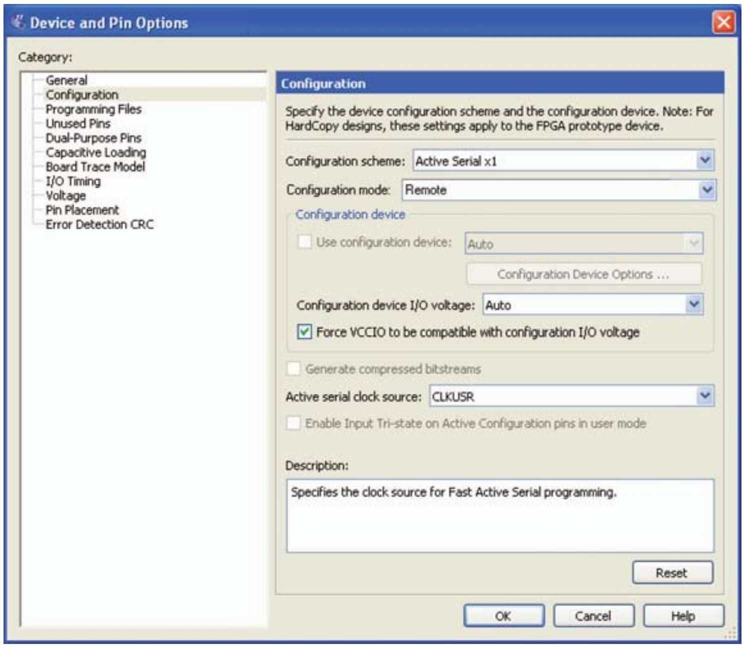
Enabling the Remote System Update Feature

You can enable remote update for Stratix V 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 in the Quartus II software, follow these steps:

1. On the Assignments 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** panel.
4. From the Configuration scheme list, select **Active Serial x1** (you can also use **Configuration Device**) (refer to Figure 9–32).
5. From the Configuration mode list, select **Remote** (refer to Figure 9–32).
6. Click **OK**.

7. In the **Settings** dialog box, click **OK**.

Figure 9–32. Enabling Remote Update for Stratix V Devices in the Compiler Settings Menu



Specify the device configuration scheme and the configuration device. Note: For HardCopy designs, these settings apply to the FPGA prototype device.

Configuration scheme: Active Serial x1

Configuration mode: Remote

☐ Use configuration device: Auto

Configuration device I/O voltage: Auto

☑ Force VCCIO to be compatible with configuration I/O voltage

☐ Generate compressed bitstreams

Active serial clock source: CLKUSR

☐ Enable Input Tri-state on Active Configuration pins in user mode

Description:

Specifies the clock source for Fast Active Serial programming.

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 V 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. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

For more information about the ALTREMOTE_UPDATE megafunction, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide*.

Design Security

This section provides an overview of the design security features and their implementation in Stratix V devices using the advanced encryption standard (AES). It also describes the security modes available in Stratix V devices that allow you to use these new features in your designs.

As Stratix V devices continue to play roles in larger and more critical designs in competitive commercial and military environments, it is increasingly important to protect your designs from copying, reverse engineering, and tampering. Stratix V design security supports the following features:

- Enhanced built-in AES decryption block to support 256-key industry-standard design security algorithm (FIPS-197 Certified)
- Volatile and non-volatile key programming support

- Secure operation mode for both volatile and non-volatile key through tamper protection bit setting
- JTAG secure mode is enable through tamper-protection bit
- Supports board level testing
- Supports in-socket key programming for non-volatile key
- Available in all configuration schemes except JTAG
- Supports both remote system upgrades and decompression feature

☞ You can use the design security feature with or without the remote system upgrades or decompression features.

The Stratix V design security feature provides the following security protection for your designs:

- Security against copying—the security key is securely stored in the Stratix V device and cannot be read out through any interface. In addition, as configuration file read-back is not supported in Stratix V devices, your 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 V configuration file formats are proprietary and the file contains millions of bits that require specific decryption. In addition, the Stratix V devices are manufactured on the most advanced 28-nm process technology, making this process very difficult.
- Security against tampering—this disables tamper attempts through the JTAG interface. You can enhance this security feature with the tamper protection bit setting. After the tamper protection bit is set, the Stratix V device can only accept configuration files encrypted with the same key. Additionally, programming through the JTAG interface is blocked. This prevents any attempts to tamper with the device from both the JTAG interface and the configuration interface.

☞ 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 V device first decrypts and then decompresses the configuration file.

☞ When you use design security with Stratix V devices in a FPP configuration scheme, it requires a different `DCLK`-to-`DATA[]` ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

JTAG Secure Mode

When you enable tamper-protection bit, Stratix V devices are in JTAG secure mode after power-up. During JTAG secure mode, many JTAG instructions are disabled. Stratix V devices only allow mandatory JTAG 1149.1 and 1149.6 instructions to be exercised. These instructions are `SAMPLE/PRELOAD`, `BYPASS`, `EXTEST`, and optional instructions such as `IDCODE` and `SHIFT_EDERROR_REG`.

To enable the access of other JTAG instructions such as `USERCODE`, `HIGHZ`, `CLAMP`, `PULSE_NCONFIG`, and `CONFIG_IO`, you must issue `UNLOCK` instruction to deactivate the JTAG secure mode. You can issue `LOCK` instruction to put the device back into JTAG secure mode. Both the `LOCK` and `UNLOCK` instructions can only be issued during user mode.

For more information about JTAG binary instruction code related to the `LOCK` and `UNLOCK` instructions, refer to the *JTAG Boundary-Scan Testing in Stratix V Devices* chapter.

Security Key Types

Stratix V devices offer two types of keys—volatile and non-volatile. Table 9–19 lists the differences between the volatile key and non-volatile key.

Table 9–19. Security Key Types

Key Types	Key Programmability	Power Supply for Key Storage	Programming Method
Volatile Key	■ Re-programmable ■ Erasable	Required external battery, V_{CCBAT} *(1)*	On-board
Non-volatile Key	One-time programming	Does not require an external battery	On-board and in-socket programming *(2)*

Notes to Table 9–19:

(1) 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_{CCPGM}. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition.

(2) In-socket programming is offered through third party vendors.

Both non-volatile and volatile key programming offers protection from reverse engineering and copying. If you set the tamper-protection bit, the design is also protected from tampering.

Perform key programming through the JTAG interface. Also, ensure that the `nSTATUS` pin is released high before any key-programming attempts.

For more information about battery specifications, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

For more information about the V_{CCBAT} pin connection recommendations, refer to the *Stratix V Device Family Pin Connection Guidelines*.

Security Modes

Table 9–20 lists the security modes available in Stratix V devices.

Table 9–20. Supported Security Modes

Security Mode	Tamper protection Bit Setting	Device Accepts Unencrypted File	Device Accepts Encrypted File	Security Level
No-key	—	Yes	No	—
Volatile Key	—	Yes *(2)*	Yes	Secure
Volatile Key with Tamper Protection Bit Set	Set *(1)*	No	Yes	Secure with tamper resistant
Non-volatile Key	—	Yes *(2)*	Yes	Secure
Non-volatile Key with Tamper Protection Bit Set	Set *(1)*	No	Yes	Secure with tamper resistant

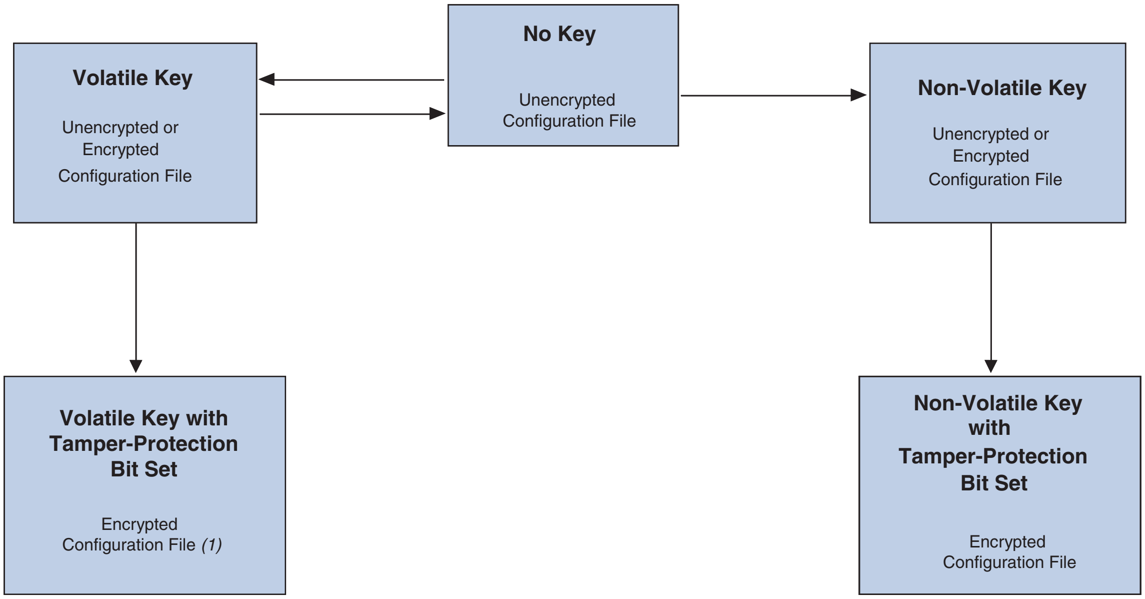
Notes to Table 9–20:

(1) Enabling the tamper protection bit disables test mode in Stratix V devices and disables programming through the JTAG interface. This process is irreversible and prevents Altera from carry-out failure analysis. Contact Altera Technical Support to enable the tamper protection bit.

(2) Use the unencrypted configuration bitstream support only for board-level testing.

Figure 9–33 shows the sequence of the security modes available in Stratix V devices.

Figure 9–33. Stratix V Security Modes—Sequence and Restrictions


No Key
Unencrypted Configuration File
Volatile Key
Unencrypted or Encrypted Configuration File
Non-Volatile Key
Unencrypted or Encrypted Configuration File
Volatile Key with Tamper-Protection Bit Set
Encrypted Configuration File (1)
Non-Volatile Key with Tamper-Protection Bit Set
Encrypted Configuration File

Note to Figure 9–33:

(1) Stratix V devices do not accept the encrypted configuration file if the volatile key is erased. You must use the volatile key without tamper-protection bit set to reprogram the key if the volatile key in Stratix V device is erased.

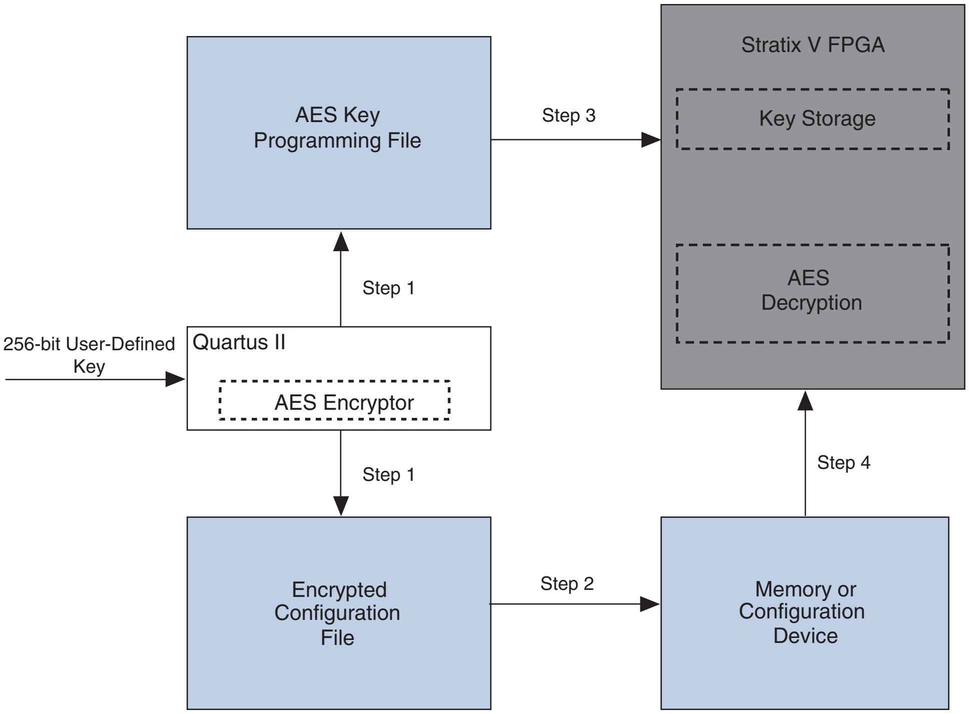
Design Security Implementation Steps

Stratix V devices are SRAM-based devices. To provide design security, Stratix V devices require a 256-bit security key for configuration bitstream design security. To carry out secure configuration, follow these steps:

1. The Quartus II software generates the design security key programming file and encrypts the configuration data using the user-defined 256-bit key.
2. Store the encrypted configuration file in the external memory.
3. Program the AES key programming file into the Stratix V device through a JTAG interface.
4. Configure the Stratix V device. At system power-up, the external memory device sends the encrypted configuration file to the Stratix V device.

Figure 9–34 shows the design security implementation steps.

Figure 9–34. Design Security Implementation Steps



Document Revision History

Table 9–21 lists the revision history for this chapter.

Table 9–21. Document Revision History

Date	Version	Changes
May 2011	1.3	■ Chapter moved to volume 2 for the 11.0 release. ■ Added "Remote System Upgrade Using EPCQ 256" and "JTAG Secure Mode" sections. ■ Updated Table 9–5. ■ Updated "Configuration", "Configuration Error", "Programming EPCS and EPCQ", "JTAG Configuration", "Remote Update Mode", and "Design Security" sections. ■ Minor text edits.
January 2011	1.2	■ Updated Table 9–7, Table 9–8, Table 9–12, and Table 9–14. ■ Updated Figure 9–15 and Figure 9–21. ■ Updated "User Watchdog Timer", "DCLK-to-DATA[] Ratio for FPP configuration", "V_{CCPD} Pin", "POR Delay Specification", and "Programming EPCS and EPCQ" sections.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

10. SEU Mitigation in Stratix V Devices

SV51011-1.2

This chapter describes how to activate and use the error detection cyclic redundancy check (CRC) feature when your Stratix® V device is in user mode and how to recover from configuration errors caused by CRC errors. The error detection feature is enhanced in the Stratix V devices.

This chapter contains the following sections:

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 V device is correct.
- Alert the system to the occurrence of a configuration error.

☞ For Stratix V devices, the error detection CRC feature is provided in the Quartus® II software starting with version 10.0.

Using the error detection CRC feature for the Stratix V device family has no impact on fitting or performance.

Error Detection Fundamentals

Error detection determines if the data received through a medium 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 function to calculate 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 V devices are successfully configured and in user mode, the error detection CRC feature ensures the integrity of the configuration data.

Configuration Error Detection

In configuration mode, a frame-based 16-bit configuration CRC is stored in the configuration data and contains the CRC value for each data frame.

During configuration, the Stratix V device calculates the 16-bit configuration CRC value based on the frame of data that is received and compares it against the pre-calculated 16-bit configuration CRC value in the data stream. If the 16-bit configuration CRC values do not match, `nSTATUS` is set low. Configuration continues until either the device detects an error or configuration is complete.

User Mode Error Detection and Correction

Stratix V devices offer on-chip circuitry for automated single event upset (SEU) detection. Some applications 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 V devices, eliminating the need for external logic. Stratix V devices have built-in error detection circuitry to detect data corruption by soft errors in the configuration random access memory (CRAM) cells. This feature allows all CRAM contents to be read and verified to match a configuration-computed 32-bit error detection CRC value. Soft errors are changes in a CRAM's bit state due to an ionizing particle.

To enable the error detection process when the device transitions into user mode, turn on the **Enable Error Detection CRC_ERROR pin** option on the **Error Detection CRC** page of the **Device and Pin Options** dialog box in the Quartus II software.

The error detection capability continuously calculates the 32-bit error detection CRC value of the configured CRAM bits and compares it with the configuration-computed 32-bit error detection CRC value. The 32-bit error detection CRC value is computed during the configuration stage. The error detection circuitry generates 32 CRC check bits per frame and then stores them in the CRAM. If the 32-bit error detection CRC values 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.

A single 32-bit error detection CRC calculation is done on a per frame basis. After the error detection circuitry has finished the CRC calculation for a frame, the resulting 32-bit signature is 0×00000000. If the error detection circuitry detects no CRAM bit errors in a frame, the output signal `CRC_ERROR` is set to low. If the circuitry detects a CRAM bit error in a frame in the device, the resulting signature is non-zero and the error detection circuitry starts searching for the error bit location.

The error detection circuitry in Stratix V 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, triple-bit, quadruple-bit, and quintuple-bit errors. The probability of more than five CRAM bits being flipped by a SEU is very low. In general, the probability of detection for all error patterns is 99.9999%.

The error detection circuitry reports the bit location and determines the type of error for single-bit errors or double-adjacent errors. The probability of other error patterns is very low and the reporting of bit location is not guaranteed.

You can also read the error bit location through the JTAG and the core interface. Before the error detection circuitry detects the next error in another frame, you must shift erroneous bits out from the error message register (EMR) with either the JTAG instruction, `SHIFT_EDERROR_REG`, or the core interface. The CRC circuitry continues to run, and if an error is detected, you must decide whether to complete the reconfiguration or to ignore the CRC error.

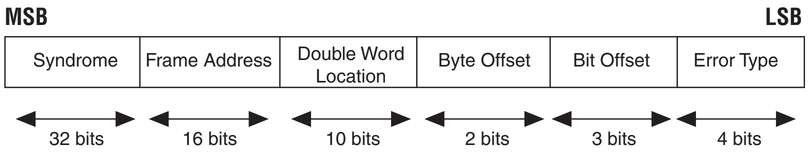
Table 10–1 lists the instruction code for the `SHIFT_EDERROR_REG` JTAG instruction.

Table 10–1. SHIFT_EDERROR_REG JTAG Instruction

JTAG Instruction	Instruction Code	Description
`SHIFT_EDERROR_REG`	00 0001 0111	The JTAG instruction connects the EMR to the JTAG pin in the error detection block between the `TDI` and `TDO` pins.

Figure 10–1 shows the content of the EMR.

Figure 10–1. Error Message Register



The type of error is identified in the first four bits of the EMR. Table 10–2 lists the error types represented in the EMR.

Table 10–2. Error Type in Error Message Register

Error Type				Description
Bit 3	Bit 2	Bit 1	Bit 0	
0	0	0	0	No CRC error.
0	0	0	1	Location of a single-bit error is identified.
0	0	1	0	Location of a double-adjacent error is identified.
1	1	1	1	There is more than one error.
Others				Reserved.

For more information about the timing requirement to shift out error information from the EMR, refer to "Error Detection Timing" on page 10–8.

The error detection circuitry continues to calculate the 32-bit error detection CRC value and 32-bit signatures for the next frame of data regardless of whether an error has occurred in the current frame or not. You must monitor the `CRC_ERROR` signal and take the appropriate actions if a CRC error occurs.

The error detection circuitry in Stratix V devices uses a 32-bit CRC-ANSI standard (32-bit polynomial) as the CRC generator. The computed 32-bit CRC signature for each frame is stored in the CRAM. The total storage size is 32 (number of bits per frame) × the number of frames.

The Stratix V device error detection CRC feature does not check memory blocks and I/O buffers. Thus, the CRC_ERROR signal may 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. MLAB and M20K memory blocks support parity bits that are used to check the contents of memory blocks for any error.

For more information about error detection in Stratix V memory blocks, refer to the *TriMatrix Embedded Memory Blocks in Stratix V Devices* chapter.

In Stratix V, in addition to the error detection capability, the error detection circuitry also supports error correction or internal scrubbing, which is an ability to internally correct soft errors that have been detected. This is done on a per-frame basis. If internal scrubbing is enabled, the device corrects single-bit error or double-adjacent error in the CRAM bits while the device is still running.

To provide testing capability of the error detection block, a JTAG instruction, EDERROR_INJECT, is provided. This instruction is able to change the content of the 47-bit JTAG fault injection register used for error injection in Stratix V devices, thereby enabling testing of the error detection block.

You can only execute the EDERROR_INJECT JTAG instruction when the device is in user mode.

Table 10–3 lists the EDERROR_INJECT JTAG instruction.

Table 10–3. EDERROR_INJECT JTAG Instruction

JTAG Instruction	Instruction Code	Description
EDERROR_INJECT	00 0001 0101	This instruction controls the 47-bit JTAG fault injection register 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 and on-the-fly, without having to reconfigure the device. You can then switch to the CRC circuit to check for real errors induced by a 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 JTAG instruction to flip the readback bits. The Stratix V device is then forced into error test mode. Altera recommends reconfiguring the device after the test completes.

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 10–7.

Table 10–4 lists how the fault injection register is implemented and describes error injection.

Table 10–4. Fault Injection Register

Description	Bit[46..43]					Bit[42..32]	Bit[31..0]
	Error Type				Error Injection Type	Byte Location of the Injected Error	Error Byte Value
	Bit[46]	Bit[45]	Bit[44]	Bit[43]			
Content	0	0	0	0	No error injection	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.
	0	0	0	1	Single error injection		
	0	0	1	0	Double-adjacent error injection		
	Others				Reserved		

Error Detection Pin Description

Table 10–5 lists the `CRC_ERROR` pin.

Table 10–5. CRC_ERROR Pin Description

Pin Name	Pin Type	Description
`CRC_ERROR`	I/O, output, or output open-drain	Active high signal indicating that the error detection circuit has detected errors in the configuration CRAM bits. This is an optional pin and is used when you enable the error detection CRC circuit. When you disable the error detection CRC circuit, it is an user I/O pin. When using the WYSIWYG function, you can route the `crcerror` port from the WYSIWYG atom to the dedicated `CRC_ERROR` pin or any user I/O. To route the `crcerror` port to user I/O, you must insert a D-type flipflop in between the `crcerror` port and the I/O. By default, the Quartus II software sets the `CRC_ERROR` pin as output open-drain when you enable the error detection CRC circuitry. By option, you can set this pin to be an output by turning off the **Enable open-drain on CRC_ERROR pin** option on the **Error Detection CRC** page of the **Device and Pin Option** dialog box in the Quartus II software. If the `CRC_ERROR` pin is used as an output, you must ensure that the V_{CCIO} of the bank in which the pin resides meet the input voltage specification of the system receiving the signal. Using the pin as an open-drain provides an advantage on the voltage leveling. To use the `CRC_ERROR` pin as an open-drain, you can tie this pin to V_{CCPGM} through a 10-kΩ resistor. Alternatively, depending on the input voltage specification of the system receiving the signal, tie this pin to a different pull-up voltage.

Error Detection Block

The error detection block contains the logic necessary to calculate the 32-bit error detection CRC signature for the configuration CRAM bits in the Stratix V device.

The CRC circuit continues running even if an error occurs. When a CRC error occurs, the device sets the CRC_ERROR pin high. Table 10–6 lists the two types of CRC detection that check the configuration bits.

Table 10–6. Two Types of CRC Detection

User Mode CRC Error Detection	Configuration CRC Error Detection
■ This is the CRAM error checking ability (32-bit error detection CRC) during user mode for use by the CRC_ERROR pin. ■ For each frame of data, the pre-calculated 32-bit error detection 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 finds the location of the error. ■ The error messages can be shifted out through the JTAG instruction or core interface logics while the error detection block continues running. ■ The JTAG interface reads out the 32-bit error detection CRC result for the first frame and also shifts the 32-bit error detection CRC bits to the 32-bit error detection CRC storage registers for test purposes. ■ You can deliberately introduce single error or double-adjacent error to the configuration memory for testing and design verification.	■ This is the 16-bit configuration CRC that is embedded in every configuration data frame. ■ During configuration, after a frame of data is loaded into the Stratix V device, the pre-computed configuration CRC is shifted into the CRC circuitry. ■ At the same time, the 16-bit configuration CRC value for the data frame shifted-in is calculated. If the pre-computed configuration CRC and calculated configuration CRC values do not match, nSTATUS is set low. Every data frame has a 16-bit configuration CRC; therefore, there are many 16-bit configuration CRC values for the whole configuration bitstream as there are many data frames. Every device has different lengths of the configuration data frame.

☞ The "Error Detection Registers" section focuses on the user mode CRC error detection.

Error Detection Registers

There is one set of 32-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 10–2 shows the error detection circuitry, syndrome registers, and error injection block.

Figure 10–2. Error Detection Circuitry, Syndrome Registers, and Error Injection Block

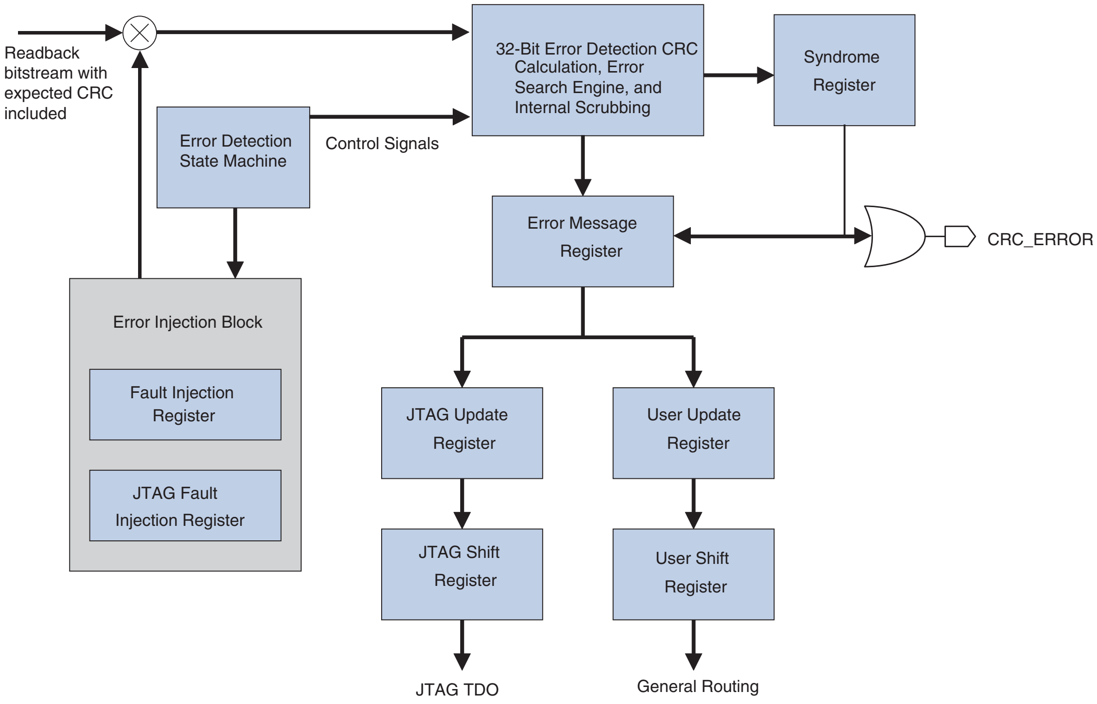


Table 10–7 lists the registers shown in Figure 10–2.

Table 10–7. Error Detection Registers (Part 1 of 2)

Register	Description
Syndrome Register	This 32-bit 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 67-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 EMR. The content of the register is shifted out through the SHIFT_EDERROR_REG JTAG instruction or to the core through the core interface.
JTAG Update Register	This 67-bit register is automatically updated with the contents of the EMR one cycle after this register content is validated. It includes a clock enable, which 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 EMR at the same time that the JTAG shift register is reading its contents.

Table 10–7. Error Detection Registers (Part 2 of 2)

Register	Description
User Update Register	This 67-bit register is automatically updated with the contents of the EMR one cycle after this register content is validated. It includes a clock enable, which 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 EMR at exactly the same time that the user shift register is reading its contents.
JTAG Shift Register	This 67-bit register is accessible by the JTAG interface and allows the contents of the JTAG update register to be sampled and read out by `SHIFT_EDERROR_REG` JTAG instruction.
User Shift Register	This 67-bit 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 47-bit register is fully controlled by the `EDERROR_INJECT` JTAG instruction. 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 47-bit register when it is updated.

Error Detection Timing

When you enable the error detection CRC feature through the Quartus II software, the device automatically activates the CRC error detection process after entering user mode.

If an error is detected within a frame, `CRC_ERROR` is driven high at the end of the error location search, after the EMR 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, `CRC_ERROR` is driven high again after the EMR is overwritten by the new value. You can start to unload the error message on each rising edge of the `CRC_ERROR` pin. 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 10–8 lists the minimum and maximum error detection frequencies.

Table 10–8. Minimum and Maximum Error Detection Frequencies

Device Type	Error Detection Frequency	Maximum Error Detection Frequency	Minimum Error Detection Frequency	Valid Divisors (n)
Stratix V	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 10–11). The divisor is a power of two (2), where *n* is between 1 and 8. The divisor ranges from 2 through 256 (Equation 10–1).

Equation 10–1.

$$\text{Error Detection Frequency} = \frac{100\text{MHz}}{2^n}$$

☞ The error detection frequency reflects the frequency of the error detection process for a frame because the CRC calculation in Stratix V devices is done on a per-frame basis.

The EMR is updated whenever an error occurs. If the error location and message are not shifted out before the next error location is found, the previous error location and message are overwritten by the new information. To avoid this, you must shift these bits out within one frame of the CRC verification. The minimum interval time between each update for the EMR depends on the device and the error detection clock frequency. However, slowing down the error detection clock frequency slows down the error recovery time for the SEU event.

Table 10–9 lists the estimated minimum interval time between each update for the EMR for Stratix V devices.

Table 10–9. Minimum Update Interval for Error Message Register—Preliminary

Family	Device	Timing Interval (μs)
Stratix V GX	5SGXA3	2.73
	5SGXA4	2.73
	5SGXA5	3.59
	5SGXA7	3.59
	5SGXA9	4.87
	5SGXAB	4.87
	5SGXB5	3.73
	5SGXB6	3.73
Stratix V GT	5SGTC5	3.59
	5SGTC7	3.59
Stratix V GS	5SGSD2	TBD
	5SGSD3	TBD
	5SGSD4	2.99
	5SGSD5	2.99
	5SGSD6	4.55
	5SGSD8	4.55
Stratix V E	5SEE9	4.87
	5SEEB	4.87

The 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.

The minimum CRC calculation time is calculated using the maximum error detection frequency with a divisor factor of 1. The maximum CRC calculation time is calculated using the minimum error detection frequency with a divisor factor of 8. You can estimate the CRC calculation time for other valid divisors (*n*) for a specific device density using the Equation 10–2:

Equation 10–2.

$$\text{CRC Calculation Time} = \frac{\text{Minimum calculation time}}{2} \times 2^{n}$$

Table 10–10 lists the minimum and maximum estimated clock frequency time for each CRC calculation for Stratix V devices.

Table 10–10. CRC Calculation Time—Preliminary

Family	Device	Minimum Time (µs)	Maximum Time (µs)
Stratix V GX	5SGXA3	2.71	5.42
	5SGXA4	2.71	5.42
	5SGXA5	3.57	7.14
	5SGXA7	3.57	7.14
	5SGXA9	4.85	9.70
	5SGXAB	4.85	9.70
	5SGXB5	3.71	7.42
	5SGXB6	3.71	7.42
Stratix V GT	5SGTC5	3.57	7.14
	5SGTC7	3.57	7.14
Stratix V GS	5SGSD2	TBD	TBD
	5SGSD3	TBD	TBD
	5SGSD4	2.97	5.94
	5SGSD5	2.97	5.94
	5SGSD6	4.53	9.06
	5SGSD8	4.53	9.06
Stratix V E	5SEE9	4.85	9.70
	5SEEB	4.85	9.70

Software Support

The Quartus II software, starting with version 10.0, supports the error detection CRC feature for Stratix V devices. Enabling this feature in the **Device and Pin Options** dialog box generates the `CRC_ERROR` output to the optional dual-purpose `CRC_ERROR` pin.

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 V device.
2. On the Assignments menu, click **Device**. The **Device** dialog box appears.
3. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
4. In the **Category** list, click **Error Detection CRC**.
5. Turn on **Enable Error Detection CRC_ERROR pin**.
6. By default, the Quartus II software sets the `CRC_ERROR` pin as output open-drain when you enable the error detection CRC circuitry. You can set this pin to be an output by turning off the **Enable open-drain on CRC_ERROR pin**.
7. In the **Divide error check frequency by** box, enter a valid divisor as shown in Table 10–8 on page 10–8.
8. To enable or disable the on-chip error correction feature, turn on or turn off **Enable internal scrubbing**.
9. Click **OK**.

Recovering From CRC Errors

The system that the Stratix V device resides in must control 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.

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 may require a design to account for these errors.

Document Revision History

Table 10–11 lists the revision history for this chapter.

Table 10–11. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Chapter moved to volume 2 for the 11.0 release. ■ Updated Table 10–9 and Table 10–10. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

11. JTAG Boundary-Scan Testing in Stratix V Devices

SV51012-1.2

This chapter describes the boundary-scan test (BST) features that are supported in Stratix® V devices.

Stratix V devices support IEEE Std. 1149.1 and IEEE Std. 1149.6. The IEEE Std. 1149.6 is only supported on the high-speed serial interface (HSSI) transceivers in Stratix V devices. IEEE Std. 1149.6 enables board-level connectivity checking between transmitters and receivers that are AC coupled (connected with a capacitor in series between the source and destination).

This chapter includes the following sections:

For more information about the following IEEE Std. 1149.1 BST features, refer to the *IEEE 1149.1 (JTAG) Boundary-Scan Testing in Stratix III Devices* chapter in volume 1 of the *Stratix III Device Handbook*:

- IEEE Std. 1149.1 BST architecture and circuitry
- IEEE Std. 1149.1 boundary-scan register
- IEEE Std. 1149.1 BST guidelines
- Test access port (TAP) controller state-machine

IEEE Std. 1149.6 Boundary-Scan Register

The boundary-scan cells (BSCs) for HSSI transmitters (`GXB_TX[p,n]`) and receivers/input clock buffer (`GXB_RX[p,n]`)/(`REFCLK[p,n]`) in Stratix V devices are different from the BSCs for the I/O pins.

Figure 11–1 shows the Stratix V HSSI transmitter BSC.

Figure 11–1. HSSI Transmitter BSC with IEEE Std. 1149.6 BST Circuitry for Stratix V Devices

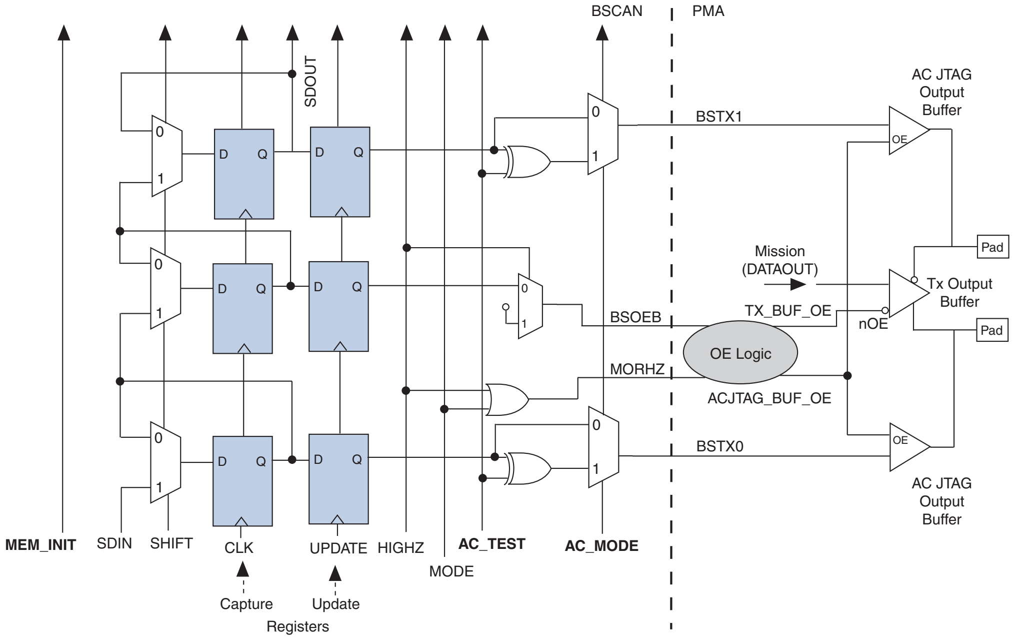
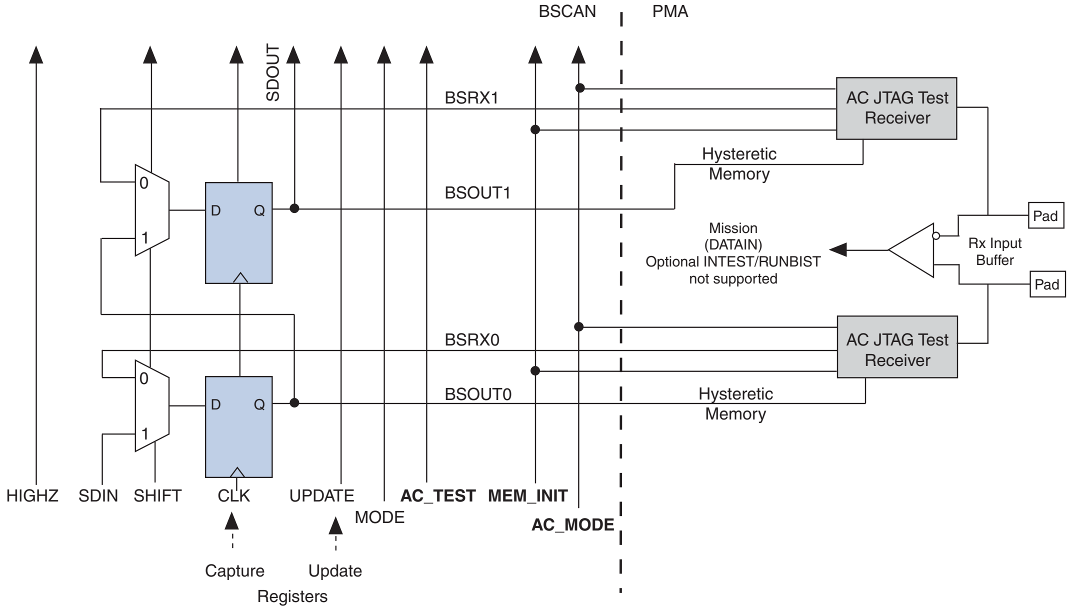


Figure 11–2 shows the Stratix V HSSI receiver/input clock buffer BSC.

Figure 11–2. HSSI Receiver/Input Clock Buffer BSC with IEEE Std. 1149.6 BST Circuitry for Stratix V Devices



BST Operation Control

Table 11–1 lists the IDCODE information for Stratix V devices.

Table 11–1. 32-Bit IDCODE Information for Stratix V Devices—Preliminary

Family	Device	IDCODE (32 Bits) *(1)*			
		Version (4 Bits)	Part Number (16 Bits)	Manufacturer Identity (11 Bits)	LSB (1 Bit) *(2)*
Stratix V GX	5SGXA3	0000	0010 1001 0001 0001	000 0110 1110	1
	5SGXA4	0000	0010 1001 0000 0001	000 0110 1110	1
	5SGXA5	0000	0010 1001 0001 0011	000 0110 1110	1
	5SGXA7	0000	0010 1001 0000 0011	000 0110 1110	1
	5SGXA9	0000	0010 1001 0001 0101	000 0110 1110	1
	5SGXAB	0000	0010 1001 0000 0101	000 0110 1110	1
	5SGXB5	0000	0010 1001 0001 0010	000 0110 1110	1
	5SGXB6	0000	0010 1001 0000 0010	000 0110 1110	1
Stratix V GT	5SGTC5	0000	0010 1001 0010 0011	000 0110 1110	1
	5SGTC7	0000	0010 1001 0100 0011	000 0110 1110	1
Stratix V GS	5SGSD2	TBD	TBD	TBD	TBD
	5SGSD3	TBD	TBD	TBD	TBD
	5SGSD4	TBD	TBD	TBD	TBD
	5SGSD5	TBD	TBD	TBD	TBD
	5SGSD6	TBD	TBD	TBD	TBD
	5SGSD8	TBD	TBD	TBD	TBD
Stratix V E	5SEE9	TBD	TBD	TBD	TBD
	5SEEB	TBD	TBD	TBD	TBD

Notes to Table 11–1:

(1) The MSB is on the left.

(2) The LSB of the IDCODE is always 1.

Table 11–2 lists the JTAG instructions that are supported by Stratix V devices.

Table 11–2. JTAG Instruction Supported by Stratix V Devices (Part 1 of 2)

JTAG Instruction	Instruction Code	Description
SAMPLE/PRELOAD	00 0000 0101	Allows a snapshot of signals at the device pins to be captured and examined during normal device operation and permits an initial data pattern to be an output at the device pins. Also used by the SignalTap™ II Embedded Logic Analyzer.
EXTEST *(1)*	00 0000 1111	Allows the external circuit and board-level interconnects to be tested by forcing a test pattern at the output pins and capturing test results at the input pins.
BYPASS	11 1111 1111	Places the 1-bit bypass register between the TDI and TDO pins, allowing the BST data to pass synchronously through selected devices to adjacent devices during normal device operation.

Table 11–2. JTAG Instruction Supported by Stratix V Devices (Part 2 of 2)

JTAG Instruction	Instruction Code	Description
USERCODE	00 0000 0111	Selects the 32-bit USERCODE register and places it between the TDI and TDO pins, allowing USERCODE to be serially shifted out of TDO.
IDCODE	00 0000 0110	Selects the IDCODE register and places it between the TDI and TDO pins, allowing IDCODE to be serially shifted out of TDO. IDCODE is the default instruction at power up and in the TAP RESET state.
HIGHZ *(1)*	00 0000 1011	Places the 1-bit bypass register between the TDI and TDO pins, allowing the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while tri-stating all of the I/O pins.
CLAMP *(1)*	00 0000 1010	Places the 1-bit bypass register between the TDI and TDO pins, allowing the BST data to pass synchronously through selected devices to adjacent devices during normal device operation while holding I/O pins to a state defined by the data in the boundary-scan register.
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is unaffected.
CONFIG_IO	00 0000 1101	Allows I/O reconfiguration through JTAG ports using IOCSR for JTAG testing. This is executed after or during configurations. The nSTATUS pin must go high before you can issue the CONFIG_IO instruction.
FACTORY	10 1000 0001	Enables access to all other JTAG instructions (other than BYPASS, SAMPLE/PRELOAD, EXTEST, IDCODE, EXTEST_PULSE, and EXTEST_TRAIN instructions, which are supported after power up). This instruction also clears the device configuration data and advanced encryption standard (AES) volatile key.
EXTEST_PULSE	00 1000 1111	Enables board-level connectivity checking between the transmitters and receivers that are AC coupled by generating three output transitions: ■ Driver drives data on the falling edge of TCK in the UPDATE_IR/DR state. ■ Driver drives inverted data on the falling edge of TCK after entering the RUN_TEST/IDLE state. ■ Driver drives data on the falling edge of TCK after leaving the RUN_TEST/IDLE state.
EXTEST_TRAIN	00 0100 1111	Behaves the same as the EXTEST_PULSE instruction except that the output continues to toggle on the TCK falling edge as long as the TAP controller is in the RUN_TEST/IDLE state.

Note to Table 11–2:

(1) Bus hold and weak pull-up resistor features override the high-impedance state of HIGHZ, CLAMP, and EXTEST.

Only the five mandatory JTAG instructions (BYPASS, SAMPLE/PRELOAD, EXTEST, EXTEST_PULSE, and EXTEST_TRAIN), the IDCODE optional instruction, and the FACTORY private instruction are supported by the JTAG pins after power up and before configuration. The FACTORY instruction must be issued before the device starts loading the core configuration data to enable access to all other JTAG instructions. This instruction also clears the device configuration data and AES volatile key.

The IDCODE instruction is the default instruction when the TAP controller is in the reset state. Without loading any instructions, you can go to the SHIFT_DR state and shift out the JTAG device ID.

☞ If the device is in reset state, when the nCONFIG or nSTATUS signal is low, the device IDCODE might not be read correctly. To read the device IDCODE correctly, you must issue the IDCODE JTAG instruction only when the nCONFIG and nSTATUS signals are high.

IEEE Std. 1149.6 mandates the addition of two new instructions—EXTEST_PULSE and EXTEST_TRAIN. These two instructions enable edge-detecting behavior on the signal path containing the HSSI pins. These instructions implement new test behaviors for the HSSI pins and simultaneously behave identically to the IEEE Std. 1149.1 EXTEST instruction for non-HSSI pins.

☞ If you use DC coupling on the HSSI signals, execute the EXTEST instruction. If you use AC coupling on the HSSI signals, execute the EXTEST_PULSE instruction.

I/O Voltage Support in a JTAG Chain

A device operating in 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 3A. All user I/O pins are tri-stated during JTAG configuration.

The JTAG chain can support several different devices. However, use caution if the chain contains devices that have different V_{CCIO} levels. The output voltage level of the TDO pin must meet the specification of the TDI pin it drives.

Table 11–3 lists board design recommendations to ensure proper JTAG chain operation.

Table 11–3. Supported TDO and TDI Voltage Combinations

Device	TDI Input Buffer Power	Stratix V TDO V_{CCPD}	
		V_{CCPD} = 3.0 V *(1)*	V_{CCPD} = 2.5 V *(2)*
Stratix V	V_{CCPD} = 3.0 V	✓	✓
	V_{CCPD} = 2.5 V	✓	✓
Non-Stratix V	V_{CC} = 3.3 V	✓ *(3)*	✓ *(4)*
	V_{CC} = 2.5 V	✓ *(3)*	✓ *(4)*
	V_{CC} = 1.8 V	✓ *(3)*	✓ *(4)*
	V_{CC} = 1.5 V	✓ *(3)*	✓ *(4)*

Notes to Table 11–3:

(1) The TDO output buffer meets V_{OH} (MIN) = 2.4 V.

(2) The TDO output buffer meets V_{OH} (MIN) = 2.0 V.

(3) Input buffer must be 3.0-V tolerant.

(4) Input buffer must be 2.5-V tolerant.

Boundary-Scan Description Language Support

The boundary-scan description language (BSDL), a subset of VHDL, provides a syntax that allows you to describe the features of an IEEE Std. 1149.6 BST-capable device that can be tested. You can test software development systems, then use the BSDL files for test generation, analysis, and failure diagnostics.

For more information about BSDL files for IEEE Std. 1149.6-compliant Stratix V devices, refer to the IEEE 1149.6 BSDL Files page on the Altera website.

You can also generate BSDL files (pre-configuration and post-configuration) for IEEE Std. 1149.6-compliant Stratix V devices with the Quartus® II software version 10.0 SP1 and later. For more information about the procedure to generate BSDL files using the Quartus II software, refer to the BSDL Files Generation in QII page on the Altera website.

Document Revision History

Table 11–4 lists the revision history for this chapter.

Table 11–4. Document Revision History

Date	Version	Changes
May 2011	1.2	Chapter moved to volume 2 for the 11.0 release. Updated Table 11–1.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

12. Power Management in Stratix V Devices

SV51013-1.2

This chapter describes power management in Stratix® V devices. Stratix V devices offer programmable power technology options for low-power operation. You can use these options, along with speed grade choices, in different permutations to give the best power and performance combination.

For thermal management, use the Stratix V internal temperature sensing diode (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.

Stratix V FPGAs deliver a breakthrough level of system bandwidth and power efficiency for high-end applications, allowing you to innovate without compromise. Stratix V devices use advanced power management techniques to enable both density and performance increases while simultaneously reducing power dissipation.

The total power of an FPGA includes static 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. You can calculate dynamic power with the equation shown in Equation 12–1.

Equation 12–1. Dynamic Power Equation ***(Note 1)***

$$P = \frac{1}{2}CV^2 \times \text{frequency}$$

Note to Equation 12–1:

(1) P = power; C = load capacitance; and V = supply voltage level.

Equation 12–1 shows that power is design-dependent and is determined by the operating frequency of the design. However, you can vary the voltage to lower dynamic power consumption by the square value of the voltage difference. Stratix V devices minimize static and dynamic power with advanced process optimizations and programmable power technology. These technologies enable Stratix V designs to optimally meet design-specific performance requirements with the lowest possible power.

The Quartus® II software optimizes all designs with Stratix V 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 V devices offer a TSD feature that self-monitors the device junction temperature and can be used with external circuitry for other activities, such as controlling air flow to the Stratix V FPGA.

This chapter contains the following sections:

- "Stratix V Programmable Power Technology"
- "Stratix V External Power Supply Requirements"
- "Temperature Sensing Diode"

Stratix V Programmable Power Technology

Stratix V 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 V 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.

A Stratix V tile 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
- PCI Express® (PCIe) hard IP
- Physical Coding Sublayer (PCS)

All blocks and routing associated with the tile share the same setting of either high-speed or low-power mode. 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 have 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 sets unused device resources in the design to low-power mode to reduce the static power. It also sets the following resources to low-power mode when they are not used in the design:

- LABs and MLABs
- TriMatrix memory blocks

- DSP blocks

If a phase-locked loop (PLL) is instantiated in the design, user may assert the `areset` pin high to keep the PLL in low-power mode.

Table 12–1 lists the available Stratix V programmable power capabilities. Speed grade considerations can add to the permutations to give you flexibility in designing your system.

Table 12–1. Programmable Power Capabilities for Stratix V Devices

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 12–1:

(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 V External Power Supply Requirements

This section describes the different external power supplies required to power Stratix V devices. You can supply some of the power supply pins with the same external power supply, provided they have the same voltage level requirements.

For more information about power supply pin connection guidelines and power regulator sharing, refer to the *Stratix V 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 V 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 V 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.

Internal Temperature Sensing Diode

You can use the Stratix V 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 you enable the ALTTEMP_SENSE megafunction 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 power consumption, disable the internal TSD with built-in ADC circuitry when not in use.

Table 12–2 lists the internal TSD specification.

Table 12–2. Internal Temperature Sensing Diode Specification

Temperature Range	Accuracy	Offset Calibrated Option	Sampling Rate	Conversion Time	Resolution	Minimum Resolution with no Missing Codes
–40°C to 100°C	±8°C *(1)*	No	1 MHz, 500 KHz	< 100 ms	8 bits	8 bits

Note to Table 12–2:

(1) Pending on silicon characterization.

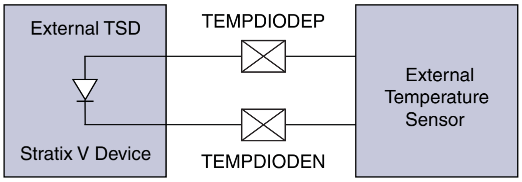
For more information about using the ALTTEMP_SENSE megafunction, refer to the *Thermal Sensor (ALTTEMP_SENSE) Megafunction User Guide*.

The external temperature sensor steers bias current through the Stratix V external TSD, which measures forward voltage and converts 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 V device and can be used for intelligent power management.

External Temperature Sensing Diode

The Stratix V external TSD requires two pins for voltage reference. Figure 12–1 shows how to connect the external TSD with an external temperature sensor device, allowing external sensing of the Stratix V die temperature. As an example, you can connect external temperature sensing devices, such as MAX1619, MAX1617A, MAX6627, and ADT 7411 to the two external TSD pins for Stratix V device die temperature reading.

Figure 12–1. TSD External Pin Connections for Stratix V Devices



For more information about the external TSD specification, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

The TSD is a very sensitive circuit that can be influenced by noise coupled from other traces on the board or within the device package itself, depending on your device usage. The interfacing signal from the Stratix V device to the external temperature sensor 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 $\text{TEMPDIODE}_P/\text{TEMPDIODE}_N$ 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 $\text{TEMPDIODE}_P/\text{TEMPDIODE}_N$ 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.
- You can use internal TSD with built-in ADC circuitry and external TSD at the same time.
- If you only use internal ADC circuitry, the external TSD pins ($\text{TEMPDIODE}_P/\text{TEMPDIODE}_N$) can be connected to GND because the external TSD pins are not used.

  For more information about the $\text{TEMPDIODE}_P/\text{TEMPDIODE}_N$ pin connection when you are not using an external TSD, refer to the *Stratix V Device Family Pin Connection Guidelines*.

For more information about device specification and connection guidelines, refer to the external temperature sensor device data sheet from the device manufacturer.

Document Revision History

Table 12–3 shows the revision history for this document.

Table 12–3. Document Revision History

Date	Version	Changes
May 2011	1.2	Chapter moved to volume 2 for the 11.0 release.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

This chapter provides additional information about the document and Altera.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact (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 to Table:

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

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate 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, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *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** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the 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`. The suffix `n` denotes an active-low signal. 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`).
↵	An angled arrow instructs you to press the Enter key.
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 question mark directs you to a software help system with related information.
👣	The feet direct you to another document or website with related information.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚡ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
✉	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.

Stratix V Device Handbook

Volume 3: Transceivers

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

SV5V2-1.3
11.0

Contents

Chapter 5. Transceiver Custom Configurations in Stratix V Devices

Chapter 6. Transceiver Loopback Support in Stratix V Devices

Chapter 7. Dynamic Reconfiguration in Stratix V Devices

Additional Information

Chapter Revision Dates

The chapters in this document, *Stratix V Device Handbook Volume 3: Transceivers*, were revised on the following dates. Where chapters or groups of chapters are available separately, part numbers are listed.

Chapter 1. Transceiver Architecture in Stratix V Devices
Revised: *May 2011*
Part Number: *SV52002-1.4*

Chapter 2. Transceiver Clocking in Stratix V Devices
Revised: *May 2011*
Part Number: *SV52003-1.2*

Chapter 3. Transceiver Reset Control in Stratix V Devices
Revised: *May 2011*
Part Number: *SV52004-2.0*

Chapter 4. Transceiver Protocol Configurations in Stratix V Devices
Revised: *May 2011*
Part Number: *SV52005-1.2*

Chapter 5. Transceiver Custom Configurations in Stratix V Devices
Revised: *May 2011*
Part Number: *SV52006-1.2*

Chapter 6. Transceiver Loopback Support in Stratix V Devices
Revised: *May 2011*
Part Number: *SV52007-2.0*

Chapter 7. Dynamic Reconfiguration in Stratix V Devices
Revised: *May 2011*
Part Number: *SV52008-1.2*

1. Transceiver Architecture in Stratix V Devices

SV52002-1.4

This chapter provides details about the Stratix® V GX and GS transceiver architecture, transceiver channels, and a description of the transmitter and receiver channel datapaths. Stratix V GX and GS devices provide up to 66 backplane capable full-duplex clock data recovery (CDR)-based transceivers with physical coding sublayer (PCS) and physical medium attachment (PMA) at serial data rates between 600 Mbps and 12.5 Gbps.

For information about features that will be supported in a future release of the Quartus® II software, refer to the *Upcoming Stratix V Device Features* document.

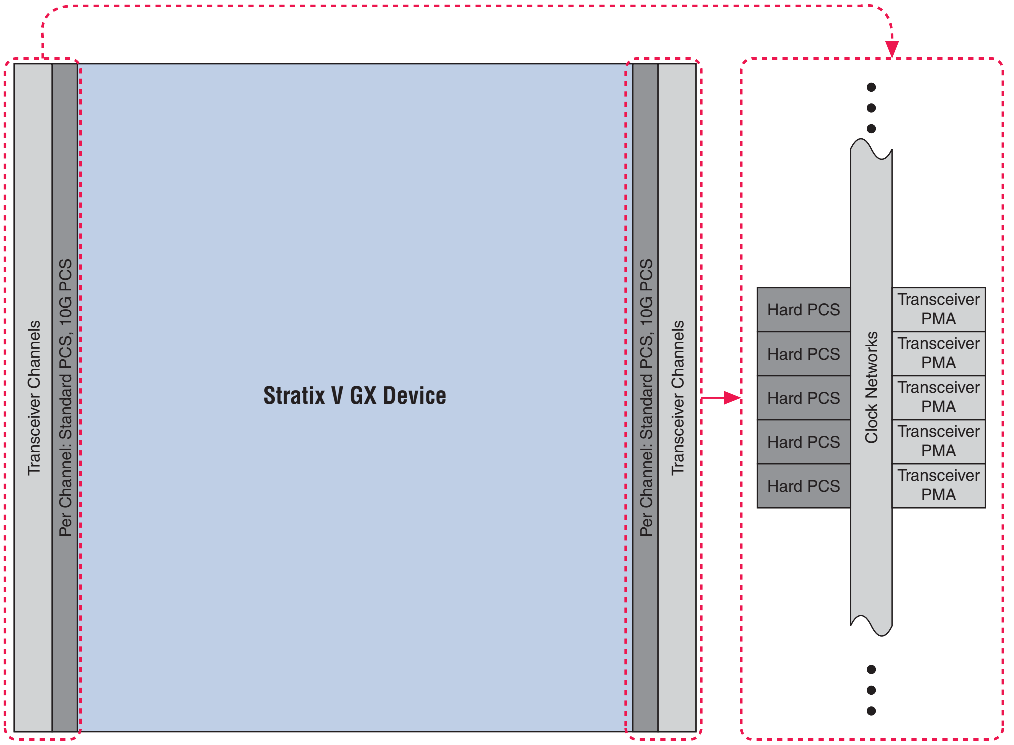
The following sections included in this chapter:

- "PMA Architecture" on page 1–5
- "Standard PCS Architecture" on page 1–18
- "10G PCS Architecture" on page 1–32
- "Bonded Configuration" on page 1–42
- "PLL Sharing" on page 1–43

Altera® 28-nm Stratix V FPGAs deliver the highest bandwidth, levels of system integration, and flexibility. This FPGA family allows you to meet the demands for increasingly high bandwidth while meeting cost and power budgets.

Stratix V GX devices have columns of transceivers on the left and right sides of the devices, as shown in Figure 1–1. Stratix V GS devices have columns of transceivers on the left side only.

Figure 1–1. Basic Layout of Transceivers in Stratix V GX Devices



The location of the transceiver bank boundaries are important for clocking resources, bonding channels, and fitting. The transceivers are grouped in transceiver blocks of three and six channels. Figure 1–2 to Figure 1–5 show the location of the blocks. For example, a 5SGSB7 device has 27 transceiver channels on one side of the device, while a 5SGXA3 device has 18 transceiver channels on each side of the device, for a total of 36 channels. Some package variations reduce the transceiver count in Figure 1–3 and Figure 1–4.

Figure 1–2. Number of Channels in Each Transceiver Bank for 5SGXB5 and 5SGXB6 Devices

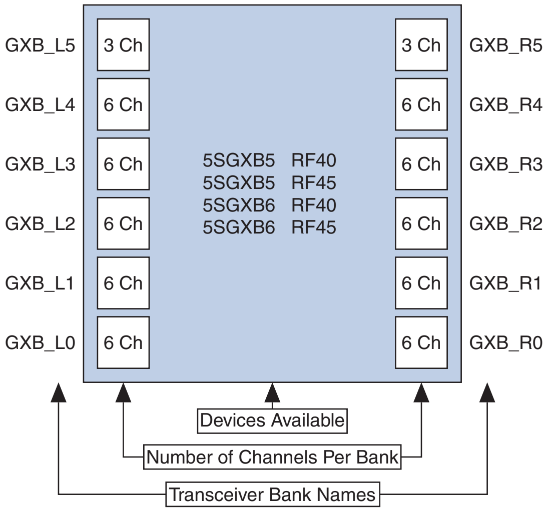


Figure 1–3. Number of Channels in Each Transceiver Bank for 5SGXA5, 5SGXA7, 5SGXAB, and 5SGXA9 Devices

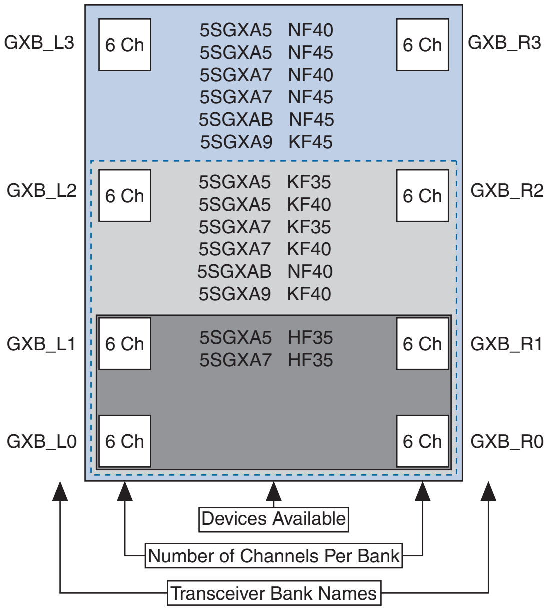


Figure 1–4. Number of Channels in Each Transceiver Bank for 5SGXA3 and 5SGXA4 Devices

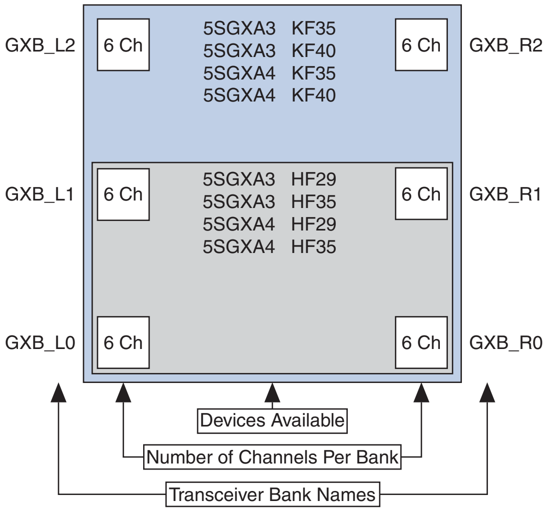
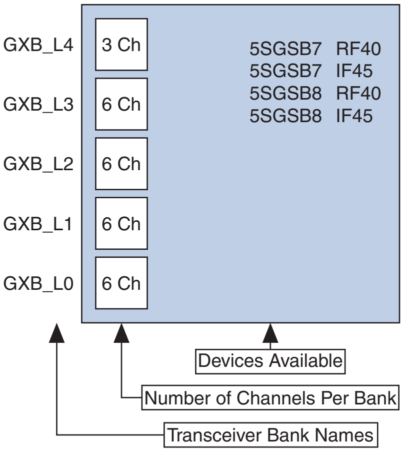


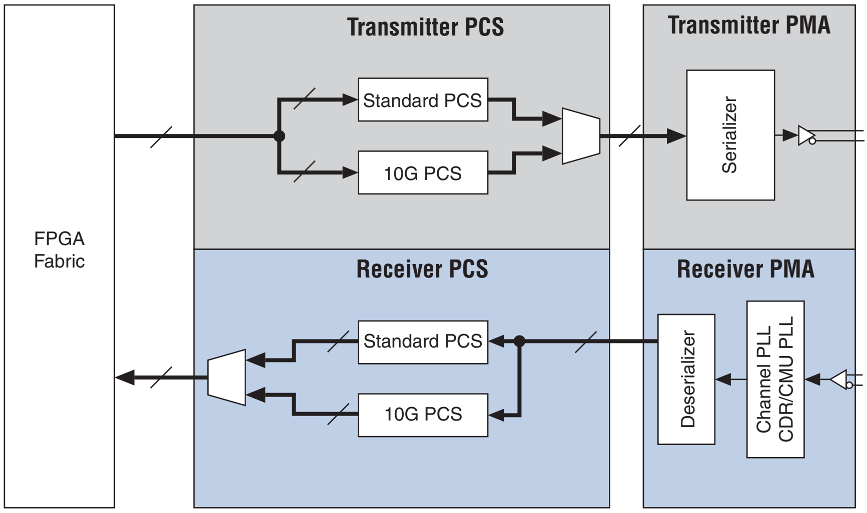
Figure 1–5. Number of Channels in Each Transceiver Bank for 5SGSB7 and 5SGSB8 Devices



For more information about device options, refer to the *Stratix V Device Family Overview* chapter.

Stratix V GX and GS transceivers are structured into full-duplex (transmitter and receiver) six-channel groups called transceiver blocks. The transmitter and receiver can operate separately. Each channel’s transmitter and receiver is made up of a PMA and a PCS section (Figure 1–6). The PMA contains the transceiver buffer, channel phase-locked loop (PLL), serializer, and deserializer. The PCS section has a choice of either standard PCS or 10G PCS.

Figure 1–6. Full Duplex Channel Showing PMA and PCS Interfaces



PMA Architecture

The PMA receives and transmits off-chip high-speed serial data streams. You can configure a PMA channel as a full-duplex channel with a transmitter and receiver or as a clock multiplier unit (CMU) PLL. Specific CMU PLLs have better performance with direct clock lines and access to the ×6 clock lines.

For more information, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

The following sections describe the Stratix V PMA architecture:

- “Clock Data Recovery Unit” on page 1–6
- “Transmitter PLLs” on page 1–11

The receiver PMA includes the following features:

- “Receiver Input Buffer” on page 1–8
- “Programmable Differential On-Chip Termination” on page 1–9
- “Programmable V_{CM}” on page 1–9
- “Programmable Equalization” on page 1–9
- “Signal Threshold Detection Circuitry” on page 1–9
- “Offset Cancellation in the Receiver Buffer and Receiver CDR” on page 1–9
- “DC Gain” on page 1–10
- “Deserializer” on page 1–10

The transmitter PMA includes the following features:

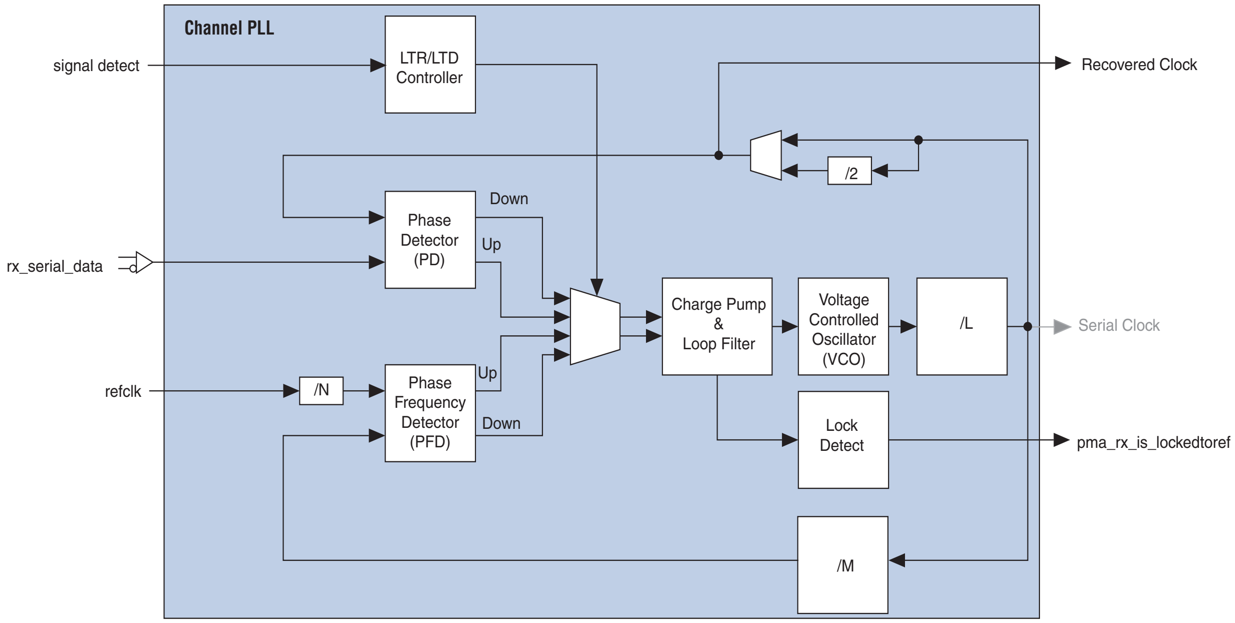
- "Transmitter Output Buffer" on page 1–15
- "Programmable Transmitter Termination" on page 1–15
- "Programmable Output Differential Voltage" on page 1–16
- "Programmable Pre-Emphasis" on page 1–16
- "Serializer" on page 1–16
- "PCIe Receiver Detect" on page 1–17
- "PCIe Electrical Idle" on page 1–17

Clock Data Recovery Unit

The PMA of each receiver channel includes a channel PLL that you can configure as a CMU PLL or receiver CDR that generates the serial transceiver clocks.

Figure 1–7 shows a CDR PLL.

Figure 1–7. CDR PLL in Stratix V Devices



Each receiver channel has a channel PLL that you can configure as an independent CDR unit to recover the clock from the incoming serial data stream. The serial and parallel recovered clocks are used to clock the receiver PMA and PCS blocks.

The CDR supports the full range of data rates. The voltage-controlled oscillator (VCO) operates at half rate. The L-counter dividers after the VCO extend the CDR's data rate range. The settings are automatically chosen by the Quartus II software.

The CDR operates in either lock-to-reference (LTR) or lock-to-data (LTD) mode. In LTR mode, the CDR tracks the input reference clock. In LTD mode, the CDR tracks the incoming serial data.

After the receiver power up and reset cycle, you must keep the CDR in LTR mode until it locks to the input reference clock. When locked to the input reference clock, the CDR output clock is trained to the configured data rate. The CDR now switches to LTD mode to recover the clock from the incoming data. The LTR/LTD controller controls the switch between the LTR and LTD modes.

Lock-to-Reference (LTR) Mode

In LTR mode, the phase frequency detector (PFD) in the CDR tracks the receiver input reference clock. The PFD controls the charge pump that tunes the VCO in the CDR. Depending on the data rate and the selected input reference clock frequency, the Quartus II software automatically selects the appropriate /M and /L divider values so the CDR output clock frequency is half the data rate. The `pma_rx_is_lockedtoref` status signal is asserted active high to indicate that the CDR has locked to the phase and frequency of the receiver input reference clock.

☞ The phase detector is inactive in LTR mode and `pma_rx_is_lockedtodata` is ignored.

Lock-to-Data (LTD) Mode

During normal operation, the CDR must be in LTD mode to recover the clock from the incoming serial data. In LTD mode, the phase detector in the CDR tracks the incoming serial data at the receiver buffer. Depending on the phase difference between the incoming data and the CDR output clock, the phase detector controls the CDR charge pump that tunes the VCO.

☞ The PFD is inactive in LTD mode. The `pma_rx_is_lockedtoref` signal toggles randomly and has no significance in LTD mode.

After switching to LTD mode, the `pma_rx_is_lockedtodata` status signal is asserted. It can take a maximum of 1 ms for the CDR to be locked to the incoming data and produce a stable recovered clock. The actual lock time depends on the transition density of the incoming data and the parts per million (PPM) difference between the receiver input reference clock and the upstream transmitter reference clock. The receiver PCS logic must be held in reset until the CDR produces a stable recovered clock.

Automatic Lock Mode

In automatic lock mode, the LTR/LTD controller initially sets the CDR to lock to the input reference clock (LTR mode). After the CDR locks to the input reference clock, the LTR/LTD controller automatically sets it to lock to the incoming serial data (LTD mode) when the following conditions are met:

- Signal threshold detection circuitry indicates the presence of valid signal levels at the receiver input buffer (PCI Express® [PCIe®] configuration only. This condition defaults to **true** for all other configurations.)
- The CDR output clock is within the configured PPM frequency threshold setting with respect to the input reference clock (frequency locked)
- The CDR output clock and the input reference clock are phase matched within approximately 0.08 UI (phase locked)

The switch from LTR to LTD mode is indicated by the assertion of the `pma_rx_is_lockedtodata` signal.

In LTD mode, the CDR uses a phase detector to keep the recovered clock phase-matched to the data. If the CDR does not stay locked to data because of frequency drift or severe amplitude attenuation, the LTR/LTD controller switches the CDR back to LTR mode to lock to the input reference clock. In automatic lock mode, the LTR/LTD controller switches the CDR from LTD to LTR mode when the following conditions are met:

- Signal threshold detection circuitry indicates the absence of valid signal levels at the receiver input buffer (PCIe configuration only. This condition defaults to **true** for all other configurations.)
- The CDR output clock is not within the configured PPM frequency threshold setting with respect to the input reference clock

The switch from LTD to LTR mode is indicated by the de-assertion of the `pma_rx_is_lockedtodata` signal.

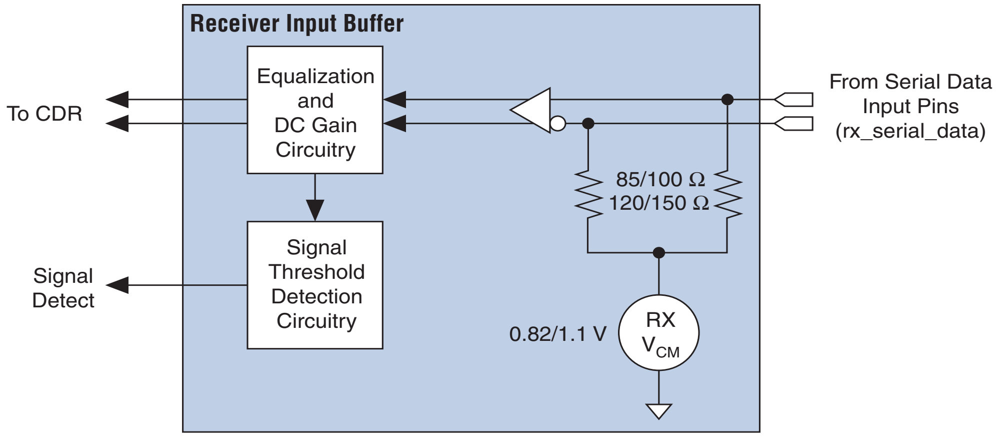
Receiver Buffer

The receiver input buffers support programmable common mode voltage (RX V_{CM}), equalization, DC gain, on-chip termination (OCT) settings, signal detect, and offset cancellation. Equalization and DC gain are described in "Receiver Analog Settings" on page 1–9.

Receiver Input Buffer

The receiver input buffer receives serial data from the `rx_serial_data` port and feeds it to the channel PLL configured as the CDR unit, as shown in Figure 1–8.

Figure 1–8. Receiver Input Buffer


Receiver Input Buffer
To CDR
Equalization and DC Gain Circuitry
Signal Threshold Detection Circuitry
Signal Detect
From Serial Data Input Pins (rx_serial_data)
85/100 Ω
120/150 Ω
0.82/1.1 V
RX V_CM

For information about the electrical features of the receiver buffer, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Programmable Differential On-Chip Termination

The receiver buffers support optional differential OCT resistances of 85, 100, 120, and 150 Ω . To select the desired receiver OCT resistor, make the assignments shown in Table 1–1 in the Quartus II Assignment Editor.

Table 1–1. Receiver On-Chip Termination Assignment Settings for Stratix V Devices

Assign To	rx_serial_data (Receiver Input Data Pins)
Assignment Name	Input Termination
Available Values	OCT 85 Ω, OCT 100 Ω, OCT 120 Ω, OCT 150 Ω, OFF

☞ The receiver OCT resistors have calibration support to compensate for process, voltage, and temperature (PVT) variations.

Programmable V_{CM}

The receiver buffers have on-chip biasing circuitry to establish the required V_{CM} at the receiver input. It supports V_{CM} settings of 0.82 V and 1.1 V.

☞ On-chip biasing circuitry is effective only if you select **on-chip receiver termination**. If you select **external termination**, you must implement off-chip biasing circuitry to establish the V_{CM} at the receiver input buffer.

Signal Threshold Detection Circuitry

In a PCIe configuration, you can enable the optional signal threshold detection circuitry. If enabled, this option senses whether the signal level present at the receiver input buffer is above the signal detect threshold voltage that you specified.

For more information, refer to the *Altera Transceiver PHY IP Core User Guide*.

Offset Cancellation in the Receiver Buffer and Receiver CDR

As silicon progresses towards smaller process nodes, the performance of circuits at these smaller nodes depends more on process variations. These process variations result in analog voltages that can be offset from the required ranges. Offset cancellation logic corrects these offsets. The receiver buffer and receiver CDR require offset cancellation.

Receiver Analog Settings

This section describes programmable equalization and DC gain, which you can change to improve the signal integrity (SI).

Programmable Equalization

Each receiver buffer has five independently programmable tap equalization circuits that boost the high-frequency gain of the incoming signal, thereby compensating for the low-pass filter effects of the physical medium. The amount of high-frequency gain required depends on the loss characteristics of the physical medium. The equalization circuitry provides up to 20 dB of high-frequency boost.

DC Gain

The receiver buffers also support programmable DC gain circuitry. Unlike the equalization circuits, the DC gain circuitry provides an equal boost to the incoming signal across the frequency spectrum. The receiver buffer supports DC gain settings of 0, 3, 6, 9, and 12 dB.

Deserializer

The deserializer block clocks in serial input data from the receiver buffer using the high-speed serial recovered clock and deserializes it using the low-speed parallel recovered clock. It forwards the deserialized data to the receiver PCS.

- In single-width mode, the deserializer supports 8- and 10-bit deserialization factors.
- In double-width mode, the deserializer supports 16- and 20-bit deserialization factors.
- In quadruple-width mode, the deserializer supports 32- and 40-bit deserialization factors.

Figure 1–9 shows the deserializer operation in single-width mode with a 10-bit deserialization factor.

Figure 1–9. Deserializer Operation in Single-Width Mode

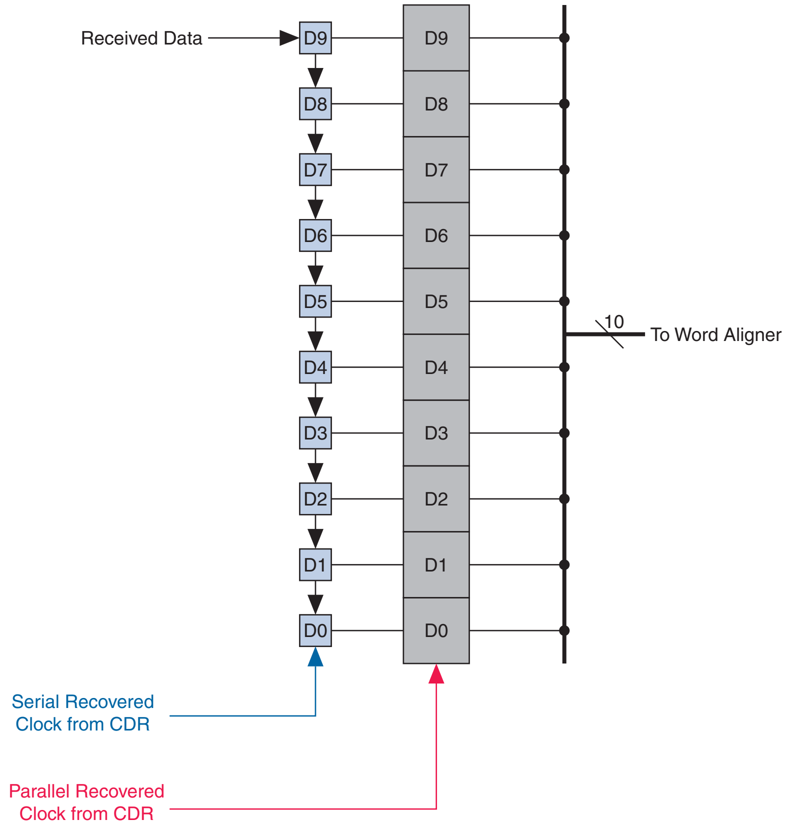
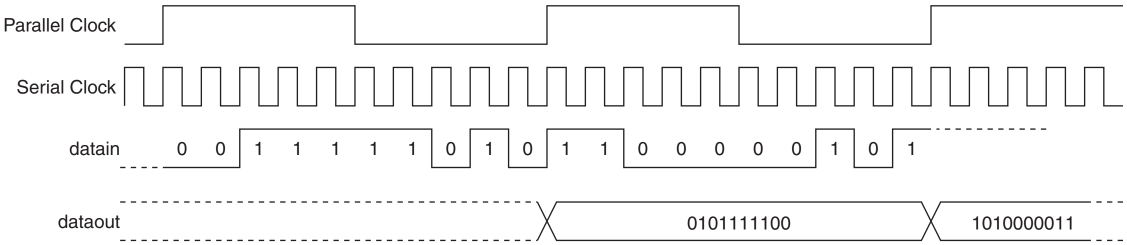

Received Data
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
10
To Word Aligner
Serial Recovered Clock from CDR
Parallel Recovered Clock from CDR

Figure 1–10 shows the serial bit order of the deserializer block input and the parallel data output of the deserializer block in single-width mode with a 10-bit deserialization factor. The serial stream (0101111100) is deserialized to a value 10'h17C. The serial data is assumed to be received LSB to MSB.

Figure 1–10. Deserializer Bit Order in Single-Width Mode

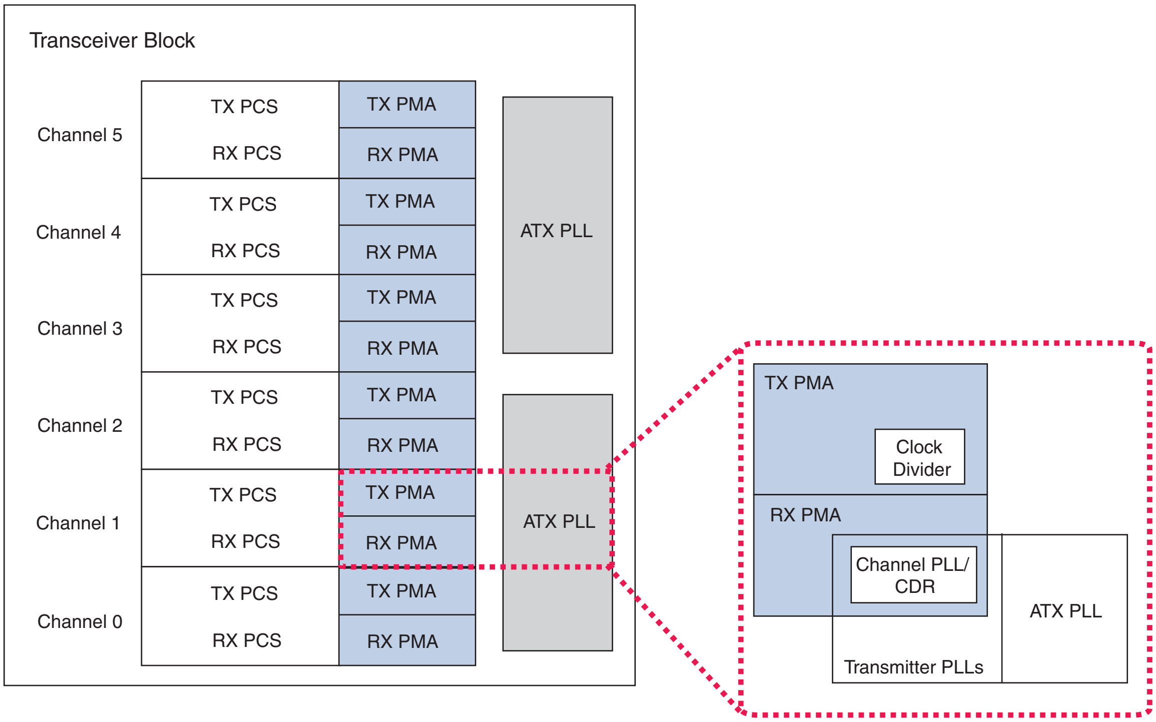


Transmitter PLLs

The following sections describe the Stratix V transmitter PLLs.

Figure 1–11 shows the transmitter PLL locations.

Figure 1–11. Transmitter PLL Locations in Stratix V Devices



Channel PLL Used as a CMU PLL (Transmitter PLL)

As stated previously, each channel PLL can function as a CMU PLL or a CDR PLL. This section describes the CMU PLL.

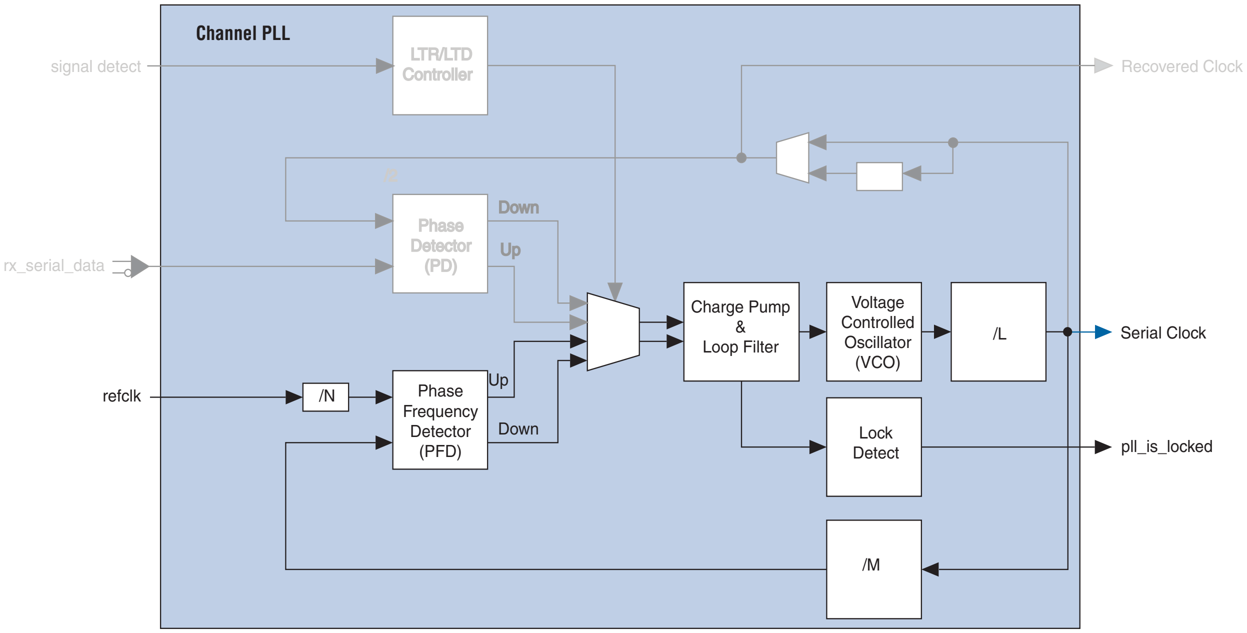
When you use the channel PLL as the CMU PLL, the receiver channel is not available as a receiver, but the transmitter in that channel is still available. The channel PLL is in LTR mode only.

The PLL's VCO operates at half rate and the L-counter dividers, after the VCO, extend the PLL's data rate range. The serial clock from the PLL is routed to the transmitter clock dividers and can be further divided down to half the data rate of the individual channels. All settings for the PLL and clock dividers are automatically chosen by the Quartus II software for the best performance based on the data rate and input clock frequency.

For CMU PLL specifications such as input or output frequency ranges, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 1–12 shows the portion of a channel PLL that functions as a CMU PLL.

Figure 1–12. CMU PLL in Stratix V Devices

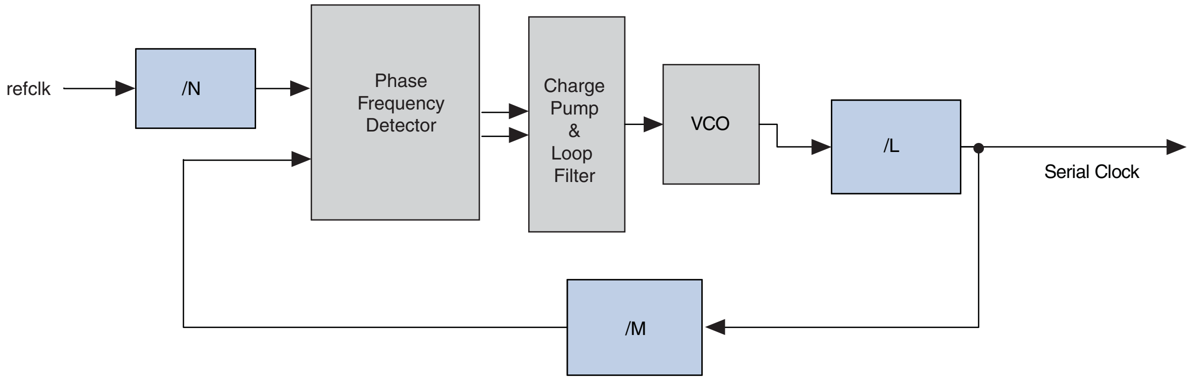


For more information, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

Auxiliary Transmit (ATX) PLL Architecture

Stratix V GX and GS devices contain two ATX PLLs per transceiver block that can generate the high-speed clocks for the transmitter channels. With these PLLs, you save the receiver resource that would be used in the channel PLL and used as a CMU PLL. The ATX PLL has the same building blocks as the other PLLs, as shown in Figure 1–13, but are tuned for better performance. ATX PLLs may not operate over the full data rate range but they generate less jitter.

Figure 1–13. ATX PLL Architecture in Stratix V Devices



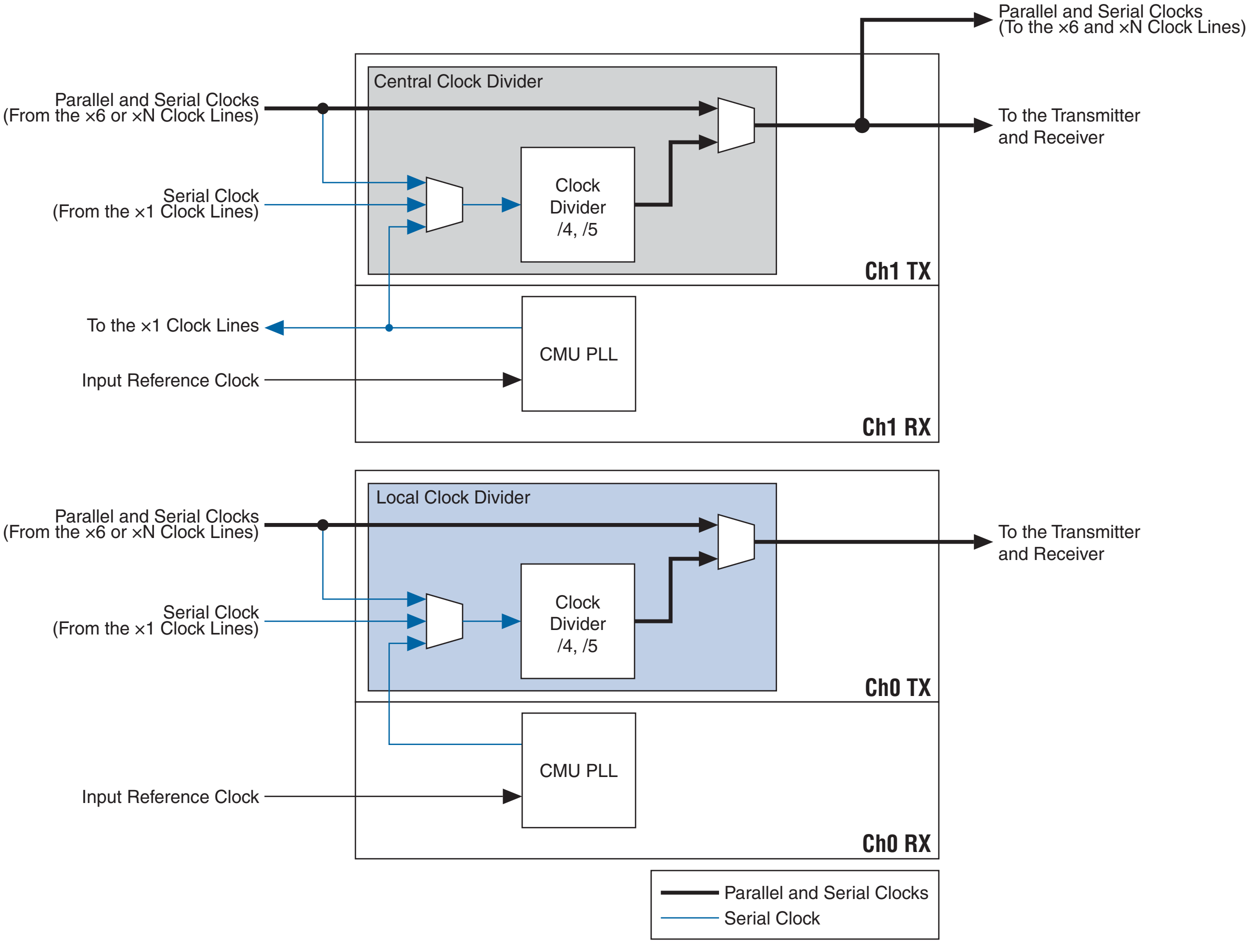
As in all PLLs, the VCO operates at half rate and the L-counter dividers, after the VCO, extends the PLL's data rate range. The serial clock from the PLL is routed to the transmitter clock dividers and can be further divided down to half the data rate of the individual channels. All settings for the PLL and clock dividers are automatically chosen by the Quartus II software for the best performance based on the data rate and input clock frequency. Any dedicated reference clocks along the same side of the device as the ATX PLL can be used to apply the reference input frequency.

For ATX PLL specifications such as input or output frequency ranges, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Clock Divider

Each transmitter channel has a clock divider called a local clock divider. Some clock dividers have special access and are called a central clock divider. Figure 1–14 shows the two types of clock dividers. The central clock dividers are located in Channels 1 and 4 of the transceiver block channels numbered 0-5. The combination of the CMU through the clock divider generates the parallel and serial clock sources for the transmitter and optionally in the receiver PCS. The central clock divider can feed the clock lines used to bond channels.

Figure 1–14. Clock Dividers



Transmitter Buffer

Transmitter buffers support programmable differential output voltage (V_{OD}), pre-emphasis, and OCT settings.

Transmitter Output Buffer

The transmitter buffer power supply only provides voltage to the transmitter output buffers in the transceiver channels. The transmitter output buffer has additional circuitry to improve signal integrity, such as the V_{OD}, programmable three-tap pre-emphasis circuitry, internal termination circuitry, and receiver detect capability to support a PCIe configuration.

Programmable Transmitter Termination

Transmitter buffers include programmable on-chip differential termination of 85, 100, 120, or 150 Ω . The resistance is adjusted by the on-chip calibration circuit during calibration, which compensates for PVT changes. The transmitter buffers are current mode drivers. Therefore, the resultant V_{OD} is a function of the transmitter termination value.

Table 1–2 lists the available termination settings.

Table 1–2. OCT Assignment Settings for Stratix V Devices

Assign To	Transmitter Serial Output Data Pin
Assignment Name	Output termination
Available Values	OCT 85 Ω, OCT 100 Ω, OCT 120 Ω, OCT 150 Ω, and OFF

You can disable OCT and use external termination. If you select external termination, the transmitter common mode is tri-stated. Common mode is based on the external termination connection.

Transmitter Analog Settings

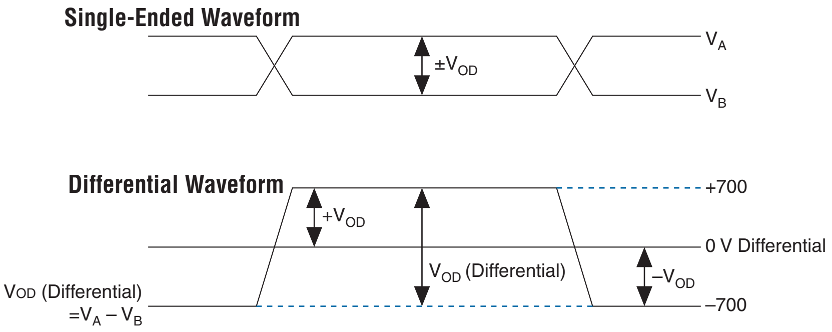
This section describes the transmitter analog setting capability to improve signal integrity.

For more information about the available settings, refer to the *Altera Transceiver PHY IP Core User Guide*.

Programmable Output Differential Voltage

You can customize the differential output voltage to handle different trace lengths, various backplanes, and receiver requirements. Figure 1–15 shows the single-ended and differential waveforms.

Figure 1–15. V_{OD} (Differential) Signal Level



Programmable Pre-Emphasis

The programmable pre-emphasis module in each transmit buffer boosts high frequencies in the transmit data signal, which might be attenuated in the transmission media. Using pre-emphasis can maximize the data eye opening at the far-end receiver.

The transceivers provide three pre-emphasis taps—pre-tap, first post-tap, and second post-tap. The pre-tap sets the pre-emphasis on the data bit before the transition. The first post-tap and second post-tap set the pre-emphasis on the transition bit and the successive bit, respectively. The pre-tap and second post-tap also provide inversion control, shown by negative values.

Serializer

The serializer converts the incoming low-speed parallel data from the transceiver PCS to high-speed serial data and sends it to the transmitter buffer. The serializer supports an 8- and 10-bit, 16- and 20-bit, and 32- and 40-bit serialization factor. The serializer block sends out the LSB of the input data first. The transmitter serializer also has polarity inversion and bit reversal capabilities.

Transmitter Polarity Inversion

The positive and negative signals of a serial differential link might accidentally be swapped during board layout. Solutions such as a board re-spin or major updates to the logic in the FPGA fabric can be expensive. The transmitter polarity inversion feature is provided to correct this situation.

A high value on the `tx_invpolarity` port inverts the polarity of every bit of the input data word to the serializer in the transmitter datapath. Because inverting the polarity of each bit has the same effect as swapping the positive and negative signals of the differential link, correct data is sent to the receiver. The dynamic signal `tx_invpolarity` might cause initial disparity errors at the receiver of an 8B/10B encoded link. The downstream system must be able to tolerate these disparity errors.

CAUTION If the polarity inversion is asserted midway through a serializer word, the word will be corrupted.

Transmitter Bit Reversal

Table 1–3 lists the transmission bit order with and without the transmitter bit reversal enabled.

Table 1–3. Transmission Bit Order for the Bit Reversal Feature for Stratix V Devices

Transmitter Bit Reversal Feature	Single-Width Mode (8 or 10 Bit)	Double-Width Mode (16 or 20 Bit)
Not enabled (default)	LSB to MSB	LSB to MSB
Enabled	MSB to LSB For example: ■ 8-bit—`D[7:0]` rewired to `D[0:7]` ■ 10-bit—`D[9:0]` rewired to `D[0:9]`	MSB to LSB For example: ■ 16-bit—`D[15:0]` rewired to `D[0:15]` ■ 20-bit—`D[19:0]` rewired to `D[0:19]`

CAUTION If reversal is asserted midway through a serializer word, the word will be corrupted.

Transmitter Protocol Specific

There are two PCIe features in the transmitter PMA section—receiver detect and electrical idle.

PCIe Receiver Detect

The transmitter buffers have a built-in receiver detection circuit for use in the PCIe configuration for Gen1 and Gen2 data rates. This circuit detects if there is a receiver downstream by sending out a pulse on the common mode of the transmitter and monitoring the reflection.

PCIe Electrical Idle

The transmitter output buffers support transmission of PCIe electrical idle (or individual transmitter tri-state).

For more information about receiver detect and electrical idle, refer to the *PCI Express PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Calibration Blocks

Stratix V GX and GS devices contain calibration circuits that calibrate the OCT resistors and the analog portions of the transceiver blocks to ensure that the functionality is independent of PVT variations.

The calibration block internally generates a constant internal reference voltage, independent of PVT variations. It uses the internal reference voltage and external reference resistor (you must connect the resistor to the `RREF` pin) to generate constant reference currents. These reference currents are used by the analog block calibration circuit to calibrate the transceiver blocks.

You must connect a separate 2 kΩ (tolerance max ±1%) external resistor on each `RREF` pin to ground. To ensure the calibration block operates properly, the `RREF` resistor connection in the board must be free from external noise.

AC JTAG

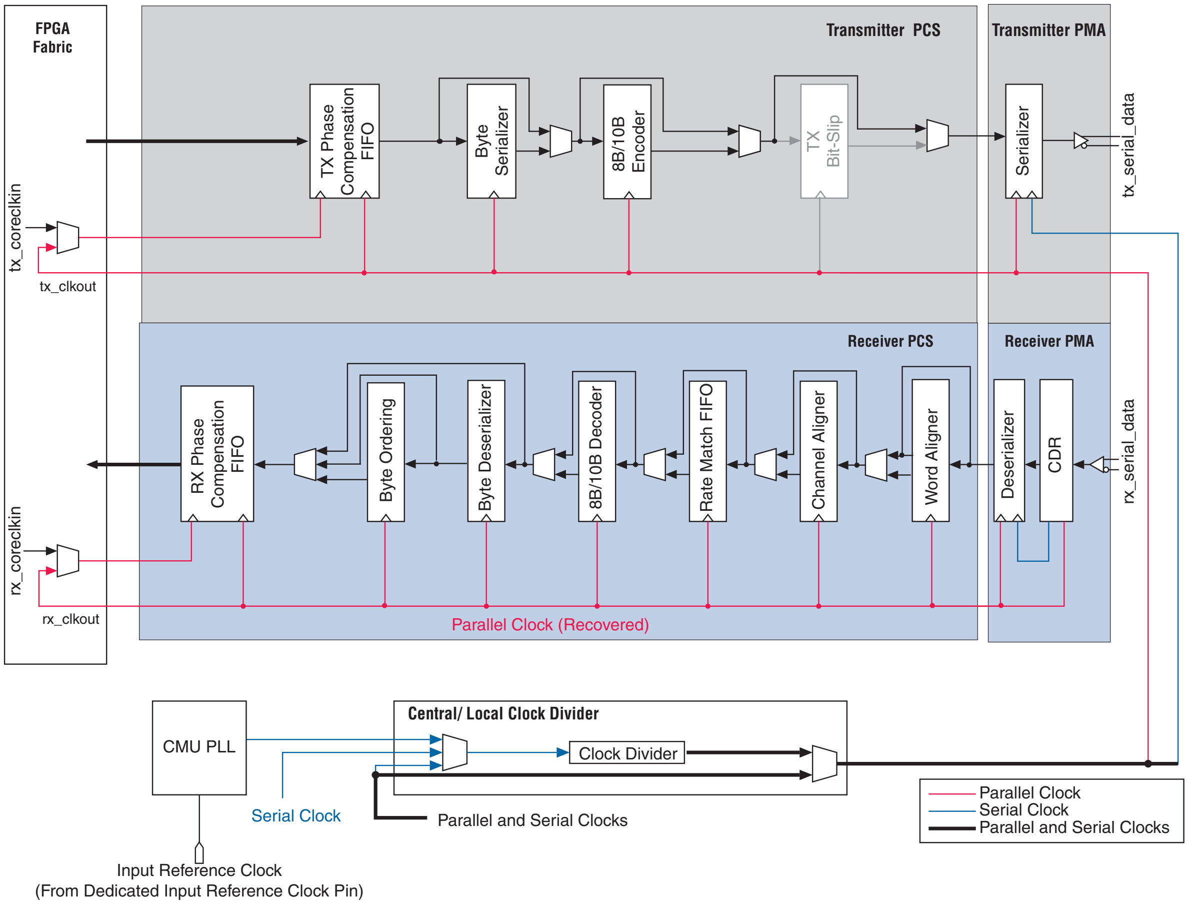
Stratix V GX and GS devices have AC JTAG to enable board-level testing of high-speed transceivers. AC JTAG is designed to conform to the *IEEE Standard for Boundary Scan Testing of Advanced Digital Networks* (1149.6-2003). The implementation features both AC and DC testing of the transceiver and receiver clock serial pins.

Standard PCS Architecture

This section describes the transceiver circuit blocks available in single- and double-width mode and within the data rate range of 600 Mbps to 8.5 Gbps.

Figure 1–16 shows the standard PCS transceiver channel datapath including the PMA-PCS and PCS-FPGA fabric interfaces.

Figure 1–16. Transceiver Datapath in Stratix V Devices



Each transceiver standard PCS consists of a transmitter channel and a receiver channel.

The receiver channel consists of the following components:

- "Word Aligner" on page 1–20
- "Rate Match (Clock Rate Compensation) FIFO" on page 1–24
- "8B/10B Decoder" on page 1–25
- "Byte Deserializer" on page 1–26
- "Byte Ordering Block" on page 1–27
- "Receiver Phase Compensation FIFO" on page 1–28

The transmitter channel consists of the following components:

- "Transmitter Phase Compensation FIFO" on page 1–29
- "Byte Serializer" on page 1–29
- "8B/10B Encoder" on page 1–30

Each transceiver channel interfaces to the PCIe hard IP block, the PIPE interface for soft IP implementations of PCIe, or directly to the FPGA fabric (FPGA fabric-transceiver interface). The transceiver channel interfaces to the PCIe hard IP block if you use the hard IP block to implement the PCIe PHY MAC, data link layer, and transaction layer. Otherwise, the transceiver channel interfaces directly to the FPGA fabric.

The PCIe hard IP-transceiver interface is outside the scope of this chapter. This chapter describes the FPGA fabric-transceiver interface only. For more information, refer to the "PCI Express PHY IP Core" chapter in the *Altera Transceiver IP Core User Guide* and to the *PCI Express Compiler User Guide*.

The standard transceiver channel datapath can be divided into two configurations based on the FPGA fabric-transceiver interface width (channel width) and the transceiver channel PMA-PCS width (serialization factor):

- Single-width configuration
- Double-width configuration

☞ The standard transceiver channel datapath does not support the quadruple-width configuration.

Table 1–4 lists the FPGA fabric-transceiver interface widths (channel width) and transceiver PMA-PCS widths (serialization factor) allowed in single-width and double-width configurations.

Table 1–4. FPGA Fabric-Transceiver Interface Width and Transceiver PMA-PCS Widths for Stratix V Devices

Name	Single-Width	Double-Width
PMA-PCS interface widths	8 and 10 bit	16 and 20 bit
FPGA fabric-transceiver interface width	8 and 10 bit 16 and 20 bit	16 and 20 bit 32 and 40 bit
Supported configurations	■ PCIe Gen1 and Gen2 ■ XAUI ■ Custom single-width	Custom double-width
Data rate range in a custom configuration	0.6 to 3.75 Gbps	1.0 to 8.5 Gbps

Receiver Standard PCS Datapath

This section describes the receiver channel datapath architecture. The sub-blocks in the receiver datapath are described in order from the word aligner to the receiver phase compensation FIFO buffer at the FPGA fabric-transceiver interface. Figure 1–16 on page 1–18 shows the receiver channel datapath.

The receiver channel standard PCS datapath consists of the following blocks:

- "Word Aligner" on page 1–20
- "Rate Match (Clock Rate Compensation) FIFO" on page 1–24
- "8B/10B Decoder" on page 1–25 (available in PCIe configuration only)
- "Byte Deserializer" on page 1–26
- "Byte Ordering Block" on page 1–27
- "Receiver Phase Compensation FIFO" on page 1–28

The receiver datapath is flexible and allows multiple modes, depending on the selected configuration.

Word Aligner

Because the data is serialized before transmission and then deserialized at the receiver, the data loses the word boundary of the upstream transmitter after deserialization. The word aligner receives parallel data from the deserializer and restores the word boundary based on a pre-defined alignment pattern that must be received during link synchronization.

Serial protocols such as PCIe specify a standard word alignment pattern. For proprietary protocols, the transceiver architecture allows you to select a custom word alignment pattern specific to your implementation.

In addition to restoring the word boundary, the word aligner also implements the following features:

- Synchronization state machine in configurations such as PCIe
- Programmable run length violation detection in all configurations
- Receiver polarity inversion in all configurations except PCIe
- Receiver bit reversal in custom single-width and custom double-width configurations
- Receiver byte reversal in custom double-width configurations

Depending on the configuration, the word aligner operates in one of the following three modes:

- Manual alignment
- Automatic synchronization state machine
- Bit-slip

Table 1–5 lists the available word aligner options.

Table 1–5. Word Aligner Options Available for Stratix V Devices

Configuration	PMA-PCS Interface Width (Bits)	Word Alignment Mode	Word Alignment Pattern Length	Word Alignment Behavior
Custom Single-Width	8	Manual Alignment	16 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is re-asserted.
		Bit-Slip	16 bits	User-controlled signal shifts data one bit at a time.
	10	Manual Alignment	7 and 10 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is re-asserted.
		Bit-Slip	7 and 10 bits	User-controlled signal shifts data one bit at a time.
		Automatic Synchronized State Machine	7 and 10 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.
Custom Double-Width	16	Manual Alignment	8, 16, and 32 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is re-asserted.
		Bit-Slip	8, 16, and 32 bits	User-controlled signal shifts data one bit at a time.
	20	Manual Alignment	7, 10, and 20 bits	User-controlled signal starts the alignment process. Alignment happens once unless the signal is re-asserted.
		Bit-Slip	7, 10, and 20 bits	User-controlled signal shifts data one bit at a time.
		Automatic Synchronized State Machine	7 and 10 bits	Data is required to be 8B/10B encoded. Aligns to the selected word aligner pattern.
PCIe	10	Automatic Synchronized State Machine	10 bits	Automatically selected word aligner pattern length and pattern.

Bit-Slip Mode Word Aligner with an 8-Bit PMA-PCS Interface Configuration

Custom single-width configuration with an 8-bit PMA-PCS interface width allows the word aligner to be configured in bit-slip mode. In bit-slip mode, the word aligner operation is controlled by the rx_bitslip bit of the pcs8g_rx_wa_control register. At every 0-1 transition of the rx_bitslip bit of the pcs8g_rx_control register, the bit-slip circuitry slips one bit into the received data stream, effectively shifting the word boundary by one bit. Also in bit-slip mode, the word aligner pcs8g_rx_wa_status register bit for rx_patterndetect is driven high for one parallel clock cycle when the received data after bit-slipping matches the 16-bit word alignment pattern programmed.

You can implement a bit-slip controller in the FPGA fabric that monitors the rx_parallel_data signal, the rx_patterndetect signal, or both, and controls the rx_bitslip signal to achieve word alignment.

Automatic Synchronization State Machine Mode Word Aligner with a 10-Bit PMA-PCS Interface Configuration

Protocols such as PCIe require the receiver PCS logic to implement a synchronization state machine to provide hysteresis during link synchronization. Each of these protocols defines a specific number of synchronization code groups that the link must receive to acquire synchronization and a specific number of erroneous code groups that it must receive to fall out of synchronization.

In PCIe configurations, the word aligner is in automatic synchronization state machine mode. It automatically selects the word alignment pattern length and pattern as specified by each protocol.

Table 1–6 lists the synchronization state machine modes. The synchronization state machine parameters are fixed for PCIe configurations as specified by the respective protocol.

Table 1–6. Synchronization State Machine Modes for Stratix V Devices

Mode	PCIe
Number of valid synchronization code groups or ordered sets received to achieve synchronization	4
Number of erroneous code groups received to lose synchronization	17
Number of continuous good code groups received to reduce the error count by one	16

After de-assertion of the reset_rx_digital signal in automatic synchronization state machine mode, the word aligner starts looking for the word alignment pattern or synchronization code groups in the received data stream. When the programmed number of valid synchronization code groups or ordered sets is received, the rx_syncstatus status bit is driven high to indicate that synchronization is acquired. The rx_syncstatus status bit is constantly driven high until the programmed number of erroneous code groups is received without receiving intermediate good groups; after which rx_syncstatus is driven low. The word aligner indicates loss of synchronization (rx_syncstatus remains low) until the programmed number of valid synchronization code groups are received again.

Programmable Run Length Violation Detection

The programmable run length violation circuit resides in the word aligner block and detects consecutive 1s or 0s in the data. If the data stream exceeds the preset maximum number of consecutive 1s or 0s, the violation is signified by the assertion of the `rx_rlv` status bit.

Table 1–7 lists the detection capabilities of the run length violation circuit.

Table 1–7. Detection Capabilities of the Run Length Violation Circuit for Stratix V Devices

Mode	PMA-PCS Interface Width (Bits)	Run Length Violation Detector Range	
		Minimum	Maximum
Single-width mode	8	4	128
	10	5	160
Double-width mode	16	8	512
	20	10	640

Receiver Polarity Inversion

The positive and negative signals of a serial differential link are often erroneously swapped during board layout. Solutions such as board re-spin or major updates to the PLD logic can be expensive. The receiver polarity inversion feature is provided to correct this situation.

Receiver Bit Reversal

By default, the receiver assumes a LSB-to-MSB transmission. If the transmission order is MSB-to-LSB, the receiver forwards the bit-flipped version of the parallel data to the FPGA fabric on `rx_parallel_data`. The receiver bit reversal feature is available to correct this situation.

Flipping the parallel data using this feature allows the receiver to forward the correct bit-ordered data to the FPGA fabric on `rx_parallel_data` in the case of an MSB-to-LSB transmission.

Table 1–8 lists the transmission bit order with and without the receiver bit reversal enabled.

Table 1–8. Received Bit Order for the Bit Reversal Feature for Stratix V Devices

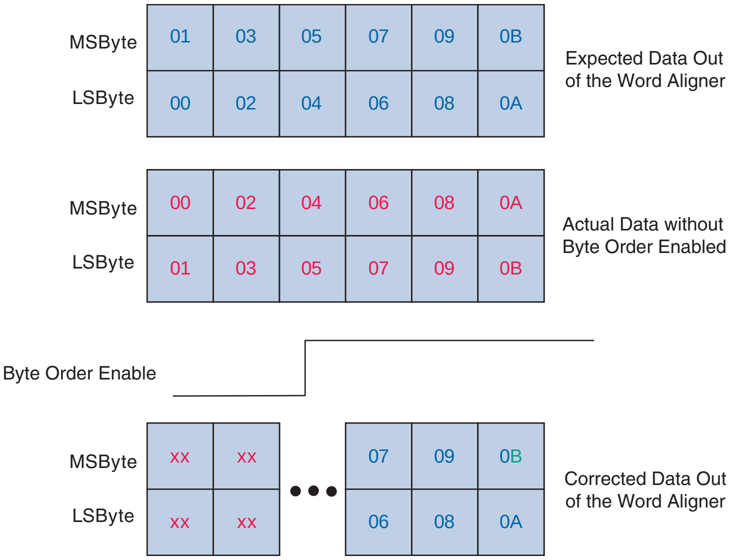
Receiver Bit Reversal Feature	Single-Width Mode (8 or 10 Bit)	Double-Width Mode (16 or 20 Bit)
Not enabled (default)	LSB to MSB	LSB to MSB
Enabled	MSB to LSB For example: ■ 8-bit—`D[7:0]` rewired to `D[0:7]` ■ 10-bit—`D[9:0]` rewired to `D[0:9]`	MSB to LSB For example: ■ 16-bit—`D[15:0]` rewired to `D[0:15]` ■ 20-bit—`D[19:0]` rewired to `D[0:19]`

Receiver Byte Reversal in Custom Double-Width Configurations

The MSByte and LSByte of the input data to the transmitter may be erroneously swapped. The receiver byte reversal feature is available to correct this situation.

Figure 1–17 shows the receiver byte reversal feature.

Figure 1–17. Receiver Byte Reversal Feature



Rate Match (Clock Rate Compensation) FIFO

In asynchronous systems, you can use independent reference clocks to clock the upstream transmitter and local receiver. Frequency differences in the order of a few hundred PPM can corrupt the data when latching from the recovered clock domain (the same clock domain as the upstream transmitter reference clock) to the local receiver reference clock domain.

The rate match (clock rate compensation) FIFO compensates for small clock frequency differences between the upstream transmitter and the local receiver clocks by inserting or removing SKP symbols or ordered sets from the inter-packet gap (IPG) or idle streams. It deletes SKP symbols or ordered sets when the upstream transmitter reference clock frequency is higher than the local receiver reference clock frequency. It inserts SKP symbols or ordered-sets when the local receiver reference clock frequency is higher than the upstream transmitter reference clock frequency.

For more information about how to use the rate match FIFO with PCIe, XAUI, and Custom protocols, refer to the *Transceiver Protocol Configurations in Stratix V Devices* chapter.

8B/10B Decoder

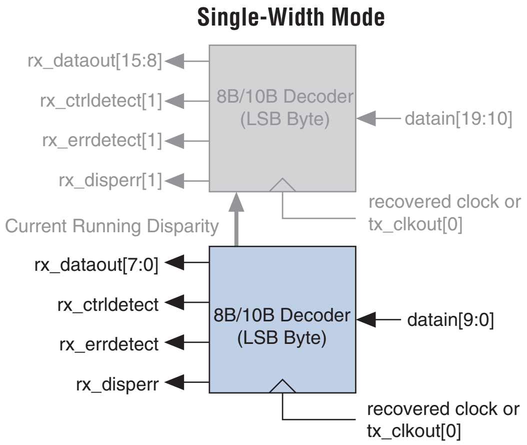
PCIe mode requires the serial data sent over the link to be 8B/10B encoded to maintain the DC balance in the transmitted serial data. This protocol requires the receiver PCS logic to implement an 8B/10B decoder to decode the data before forwarding it to the upper layers for packet processing.

The receiver channel PCS datapaths implement the 8B/10B decoder after the rate match FIFO. In configurations with the rate match FIFO enabled, the 8B/10B decoder receives data from the rate match FIFO. In configurations with the rate match FIFO disabled, the 8B/10B decoder receives data from the word aligner.

8B/10B Decoder in Single-Width Mode

Figure 1–18 shows the 8B/10B decoder in single-width mode. In this mode, the 8B/10B decoder receives 10-bit data from the rate match FIFO or word aligner (when the rate match FIFO is disabled) and decodes it into an 8-bit data +1-bit control identifier. The decoded data is fed to the byte deserializer or the receiver phase compensation FIFO (if byte deserializer is disabled).

Figure 1–18. 8B/10B Decoder in Single-Width Mode



The 8B/10B decoder is designed toward Clause 36 in the IEEE802.3 specification.

The 8B/10B decoder operates in single-width mode in the PCIe configuration only. PCIe forces selection of the 8B/10B decoder in the receiver datapath.

Control Code Group Detection

The 8B/10B decoder indicates whether the decoded 8-bit code group is a data or control code group on the `rx_datak` signal. If the received 10-bit code group is one of the 12 control code groups (/Kx.y/) specified in the IEEE802.3 specification, the `rx_datak` signal is driven high. If the received 10-bit code group is a data code group (/Dx.y/), the `rx_datak` signal is driven low.

Byte Deserializer

The FPGA fabric-transceiver interface frequency has an upper limit. In configurations that have a receiver PCS frequency greater than the upper limit stated, the parallel received data and status signals cannot be forwarded directly to the FPGA fabric because it violates this upper limit for the FPGA fabric-transceiver interface frequency. In such configurations, the byte deserializer is required to reduce the FPGA fabric-transceiver interface frequency to half while doubling the parallel data width.

The byte deserializer is required in configurations that exceed the FPGA fabric-transceiver interface clock upper frequency limit. It is optional in configurations that do not exceed the FPGA fabric-transceiver interface clock upper frequency limit.

The byte deserializer operates in two modes:

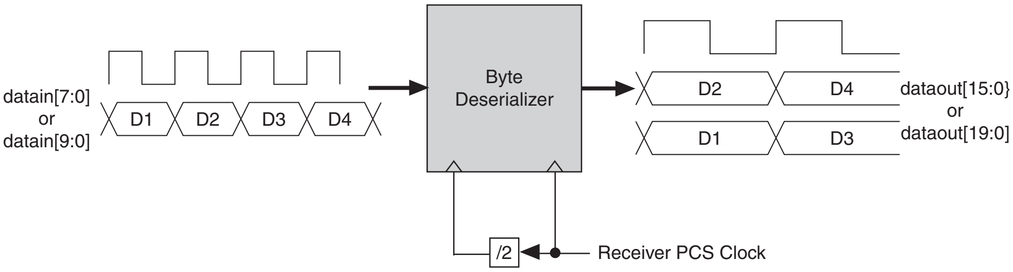
- Single-width mode
- Double-width mode

Byte Deserializer in Single-Width Mode

In single-width mode, the byte deserializer receives 8-bit wide data from the 8B/10B decoder or 10-bit wide data from the word aligner (if the 8B/10B decoder is disabled) and deserializes it into 16- or 20-bit wide data at half the speed.

Figure 1–19 shows the byte deserializer in single-width mode.

Figure 1–19. Byte Deserializer in Single-Width Mode

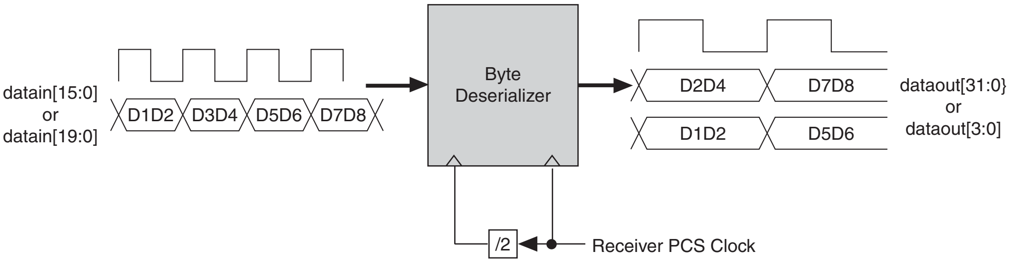


Byte Deserializer in Double-Width Mode

In double-width mode, the byte deserializer receives 16-bit wide data from the 8B/10B decoder or 20-bit wide data from the word aligner (if the 8B/10B decoder is disabled) and deserializes it into 32- or 40-bit wide data at half the speed.

Figure 1–20 shows the byte deserializer in double-width mode.

Figure 1–20. Byte Deserializer in Double-Width Mode

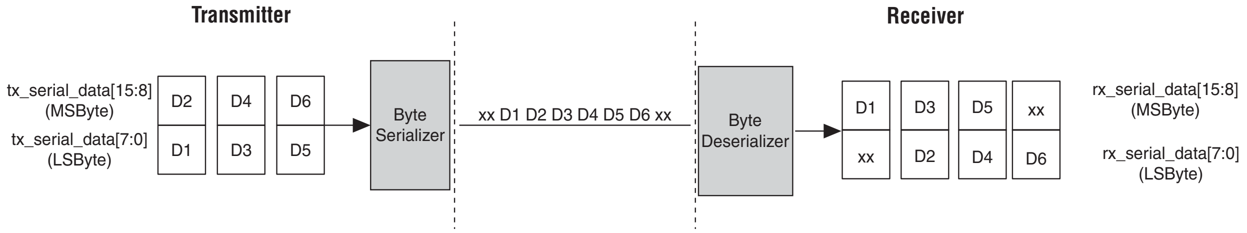


Byte Ordering Block

In single-width mode with the 16- or 20-bit FPGA fabric-transceiver interface, the byte deserializer receives one data byte (8 or 10 bits) and deserializes it into two data bytes (16 or 20 bits). Depending on when the receiver PCS logic comes out of reset, the byte ordering at the output of the byte deserializer may or may not match the original byte ordering of the transmitted data. The byte misalignment resulting from byte deserialization is unpredictable because it depends on which byte is being received by the byte deserializer when it comes out of reset.

Figure 1–21 shows a scenario in which the MSByte and LSByte of the two-byte transmitter data appears straddled across two word boundaries after being byte deserialized at the receiver.

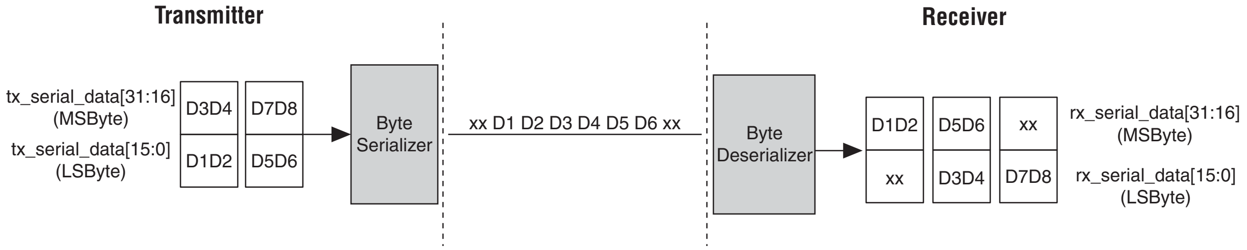
Figure 1–21. MSByte and LSByte of the Two-Bit Transmitter Data Straddled Across Two Word Boundaries



In double-width modes with a 32-bit FPGA fabric-transceiver interface, the byte deserializer receives two data bytes (16 bits) and deserializes it into four data bytes (32 bits).

Figure 1–22 shows a scenario in which the two MSBytes and LSBytes of the four-byte transmitter data appears straddled across two word boundaries after being byte deserialized at the receiver.

Figure 1–22. MSByte and LSByte of the Four-Bit Transmitter Data Straddled Across Two Word Boundaries



The transceivers have an optional byte ordering block in the receiver datapath that you can use to restore proper byte ordering before forwarding the data to the FPGA fabric. The byte ordering block looks for the user-programmed byte ordering pattern in the byte-deserialized data. You must select a byte ordering pattern that you know appears at the LSByte(s) position of the parallel transmitter data. If the byte ordering block finds the programmed byte ordering pattern in the MSByte(s) position of the byte-deserialized data, it inserts the appropriate number of user-programmed PAD bytes to push the byte ordering pattern to the LSByte(s) position, thereby restoring proper byte ordering.

Byte Ordering Block in Single-Width Modes

In custom single-width configuration, you can program a custom byte ordering pattern and byte ordering PAD pattern. Table 1–9 lists the byte ordering pattern length allowed in custom single-width configuration.

Table 1–9. Byte Ordering Pattern Length in Custom Single-Width Configuration for Stratix V Devices

Configuration	Byte Ordering Pattern Length (Bits)	Byte Ordering PAD Pattern Length (Bits)
Custom single-width configuration with: ■ 16-bit FPGA fabric-transceiver interface ■ No 8B/10B decoder ■ Word aligner in manual alignment mode	8	8

Byte Ordering Block in Double-Width Modes

In custom double-width configurations, you can program a custom byte ordering pattern and byte ordering PAD pattern in the ALT PHY IP megafunction. Table 1–10 lists the byte ordering pattern length allowed in custom double-width configuration.

Table 1–10. Byte Ordering Pattern Length in Custom Double-Width Configuration for Stratix V Devices

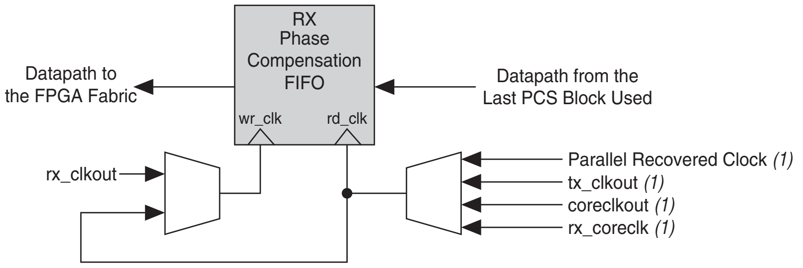
Configuration	Byte Ordering Pattern Length (Bits)	Byte Ordering PAD Pattern Length (Bits)
Custom double-width configuration with: ■ 32-bit FPGA fabric-transceiver interface ■ No 8B/10B decoder (16-bit PMA-PCS interface) ■ Word aligner in manual alignment mode	16, 8	8

Receiver Phase Compensation FIFO

The receiver phase compensation FIFO in each channel ensures reliable transfer of data and status signals between the receiver channel and the FPGA fabric. The receiver phase compensation FIFO compensates for the phase difference between the parallel receiver PCS clock (FIFO write clock) and the FPGA fabric clock (FIFO read clock).

Figure 1–23 shows the receiver phase compensation FIFO.

Figure 1–23. Receiver Phase Compensation FIFO


RX Phase Compensation FIFO
wr_clk
rd_clk
Datapath to the FPGA Fabric
Datapath from the Last PCS Block Used
rx_clkout
Parallel Recovered Clock (1)
tx_clkout (1)
coreclkout (1)
rx_coreclk (1)

Note to Figure 1–23:

(1) These clocks may have been divided by two if a byte deserializer was used.

Transmitter Standard PCS Datapath

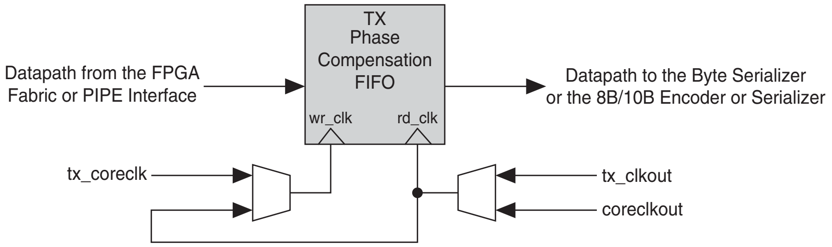
The transmitter PCS datapath, shown in Figure 1–16 on page 1–18, consists of the following blocks:

- "Transmitter Phase Compensation FIFO" on page 1–29
- "Byte Serializer" on page 1–29
- "8B/10B Encoder" on page 1–30 (only available in PCIe configurations)

Transmitter Phase Compensation FIFO

The transmitter phase compensation FIFO interfaces with the transmitter channel PCS and the FPGA fabric or PCIe interface. It compensates for the phase difference between the low-speed parallel clock and the FPGA fabric interface clock. Figure 1–24 shows the datapath and clocking of the transmitter phase compensation FIFO.

Figure 1–24. Transmitter Phase Compensation FIFO



Byte Serializer

The byte serializer divides the input datapath by two. This allows you to run the transceiver channel at higher data rates while keeping the FPGA fabric interface frequency within the maximum limit. In single-width mode, it converts the two-byte wide datapath to a one-byte wide datapath. In double-width mode, it converts the four-byte wide datapath to a two-byte wide datapath. It is optional in configurations that do not exceed the FPGA fabric-transceiver interface maximum frequency limit.

☞ The byte deserializer is required in configurations that exceed the FPGA fabric-transceiver interface maximum frequency limit.

Single-Width Mode

The byte serializer forwards the LSByte first, followed by the MSByte. The input data width to the byte serializer depends on the channel width option. For example, in single-width mode, assuming a channel width of 20, the byte serializer sends out the least significant word `tx_parallel_data[9:0]` of the parallel data from the FPGA fabric, followed by `tx_parallel_data[19:10]`. Table 1–11 lists the input and output data widths of the byte serializer in single-width mode.

Table 1–11. Input and Output Data Width of the Byte Serializer in Single-Width Mode for Stratix V Devices

Deserialization Width	Input Data Width to the Byte Serializer	Output Data Width from the Byte Serializer
Single-width mode	16	8
	20	10

Double-Width Mode

The operation in double-width mode is similar to that of single-width mode. For example, assuming a channel width of 32, the byte serializer forwards `tx_parallel_data[15:0]` first, followed by `tx_parallel_data[31:16]`. Table 1–12 lists the input and output data widths of the byte serializer in double-width mode.

Table 1–12. Input and Output Data Width of the Byte Serializer in Double-Width Mode for Stratix V Devices

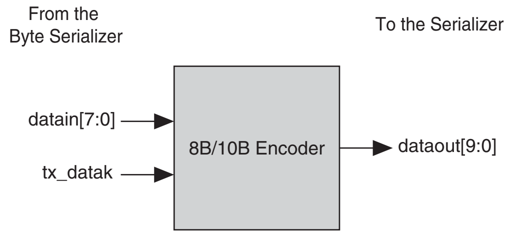
Deserialization Width	Input Data Width to the Byte Serializer	Output Data Width from the Byte Serializer
Double-width mode	32	16
	40	20

If you select the **8B/10B Encoder** option, the 8B/10B encoder uses the output from the byte serializer. Otherwise, the byte serializer output is forwarded to the serializer.

8B/10B Encoder

This is only available in PCIe configurations. The 8B/10B encoder generates 10-bit code groups from the 8-bit data and 1-bit control identifier. Figure 1–25 shows the 8B/10B encoder in single-width mode.

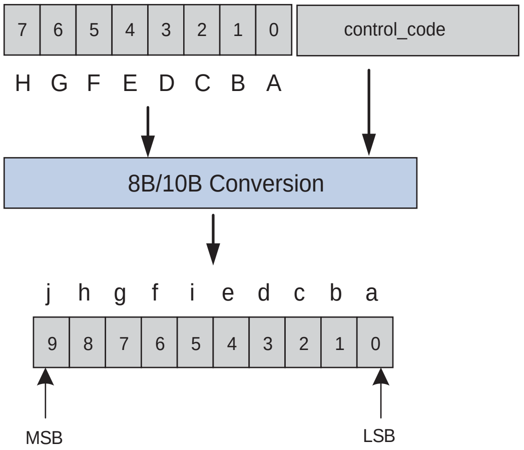
Figure 1–25. 8B/10B Encoder in Single-Width Mode



Single-Width Mode

Figure 1–25 shows the 8B/10B encoder in single-width mode. In this mode, the 8B/10B encoder translates the 8-bit data to a 10-bit code group (control word or data word) with proper disparity. If the tx_datak input is high, the 8B/10B encoder translates the input data[7:0] to a 10-bit control word. If the tx_datak input is low, the 8B/10B encoder translates the input data[7:0] to a 10-bit data word. Figure 1–26 shows the conversion format. The LSB is transmitted first.

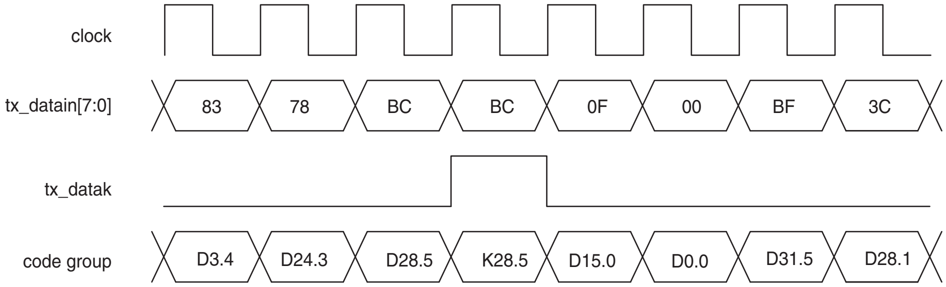
Figure 1–26. 8B/10B Conversion Format



Control Code Encoding

The 8B/10B block provides the tx_datak signal to indicate whether the 8-bit data at the tx_parallel_data signal should be encoded as a control word (Kx.y). When tx_datak is low, the 8B/10B encoder block encodes the byte at the tx_parallel_data signal as data (Dx.y). When tx_datak is high, the 8B/10B encoder encodes the input data as a Kx.y code group. Figure 1–27 shows the second 0xBC encoded as a control word (K28.5). The rest of the tx_parallel_data bytes are encoded as a data word (Dx.y).

Figure 1–27. Control Word and Data Word Transmission



The IEEE 802.3 8B/10B encoder specification identifies only a set of 8-bit characters for which tx_datak must be asserted. If you assert tx_datak for any other set of bytes, the 8B/10B encoder might encode the output 10-bit code as an invalid code (it does not map to a valid Dx.y or Kx.y code), or unintended valid Dx.y code, depending on the value entered. It is possible for a downstream 8B/10B decoder to decode an invalid control word into a valid Dx.y code without asserting code error flags.

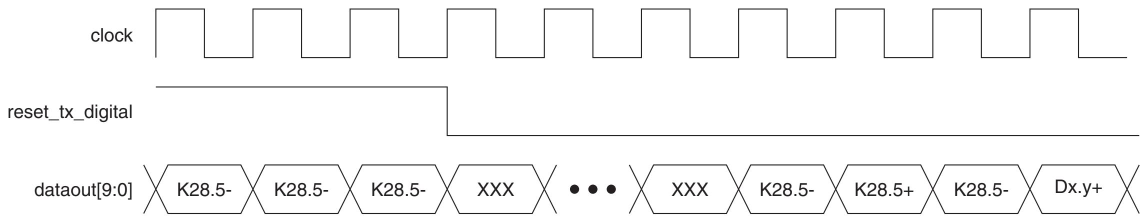
Reset Condition

The `reset_tx_digital` signal resets the 8B/10B encoder. During reset, running disparity and data registers are cleared. Also, the 8B/10B encoder outputs a K28.5 pattern from the RD- column continuously until `reset_tx_digital` is de-asserted. The input data and control code from the FPGA fabric is ignored during the reset state. After reset, the 8B/10B encoder starts with a negative disparity (RD-) and transmits three K28.5 code groups for synchronization before it starts encoding and transmitting the data on its output.

☞ While `reset_tx_digital` is asserted, the downstream 8B/10B decoder that receives the data might observe synchronization or disparity errors.

Figure 1–28 shows the reset behavior of the 8B/10B encoder. When in reset (`reset_tx_digital` is high), a K28.5- (K28.5 10-bit code group from the RD-column) is sent continuously until `reset_tx_digital` is low. Because of some pipelining of the transmitter channel PCS, some “don’t cares” (10’hxxx) are sent before the three synchronizing K28.5 code groups. User data follows the third K28.5 code group.

Figure 1–28. 8B/10B Encoder Output During reset_tx_digital Assertion



10G PCS Architecture

The 10G PCS offers a full duplex (transmitter and receiver) transceiver channel that supports serial data rates from 2.0 to 12.5 Gbps.

Several of the functional blocks are customized for protocols such as Interlaken or 10GBASE-R. The different datapath configurations for these protocols are available through the different ALT PHY MegaWizard™ Plug-In Managers, as shown in Table 1–13.

Table 1–13. Configurations Supported in 10G PCS for Stratix V Devices

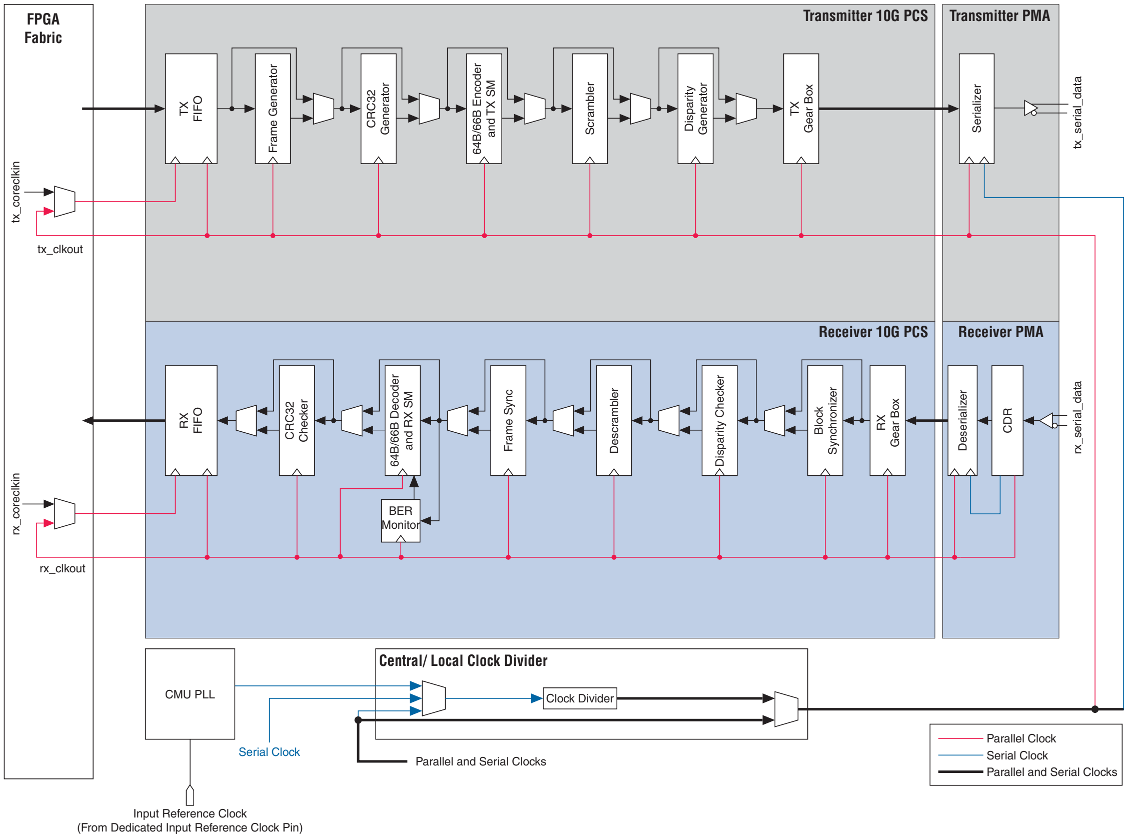
Transceiver Configuration	Transceiver ALT PHY IP	Data Rate (Gbps)	Refer to
10GBASE-R	10GBASE-R PHY IP	10.3125	10GBASE-R in the *Transceiver Protocol Configurations in Stratix V Devices* chapter
Interlaken	Interlaken PHY IP	3.125 to 10.3125	Interlaken in the *Transceiver Protocol Configurations in Stratix V Devices* chapter
10G Custom	Low Latency PHY IP	2.0 to 12.5	10G Low Latency Configuration in the *Transceiver Custom Configurations in Stratix V Devices* chapter

The functional blocks in the 10G PCS hard macro provide status and control signals to the FPGA fabric. For the signal names, refer to the *Altera Transceiver PHY IP Core User Guide*.

The clocking schemes and placement restrictions for the different datapath configurations are described in the *Transceiver Protocol Configurations in Stratix V Devices* and *Transceiver Custom Configurations in Stratix V Devices* chapters.

Figure 1–29 shows the 10G PCS datapath.

Figure 1–29. 10G PCS Datapath for Stratix V Devices *(Note 1)*



Note to Figure 1–29:

(1) Not all the blocks shown in the 10G PCS datapath are available in every configuration.

Receiver 10G PCS Datapath

The receiver channel datapath shown in Figure 1–29 consists of the following blocks:

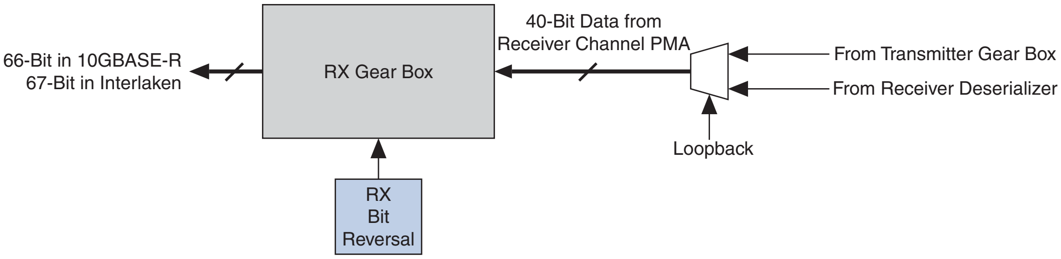
- "Receiver Gear Box" on page 1–34
- "Block Synchronizer" on page 1–35
- "Disparity Checker" on page 1–35
- "Descrambler" on page 1–35
- "Frame Synchronizer" on page 1–36
- "Bit-Error Rate (BER) Monitor" on page 1–37
- "64B/66B Decoder" on page 1–37
- "CRC-32 Checker" on page 1–37
- "Receiver FIFO" on page 1–37

Receiver Gear Box

The receiver gear box adapts the PMA data width to a larger bus width so for interfacing with the PCS.

The PMA bus width is smaller than the PCS bus width, so the receiver gear box expands the data bus width from the PMA to the PCS. Because this process is transparent, you can continuously feed data to the receiver gear box. Figure 1–30 shows the receiver gear box for the 10GBASE-R configuration.

Figure 1–30. Receiver Gear Box


66-Bit in 10GBASE-R
67-Bit in Interlaken
RX Gear Box
40-Bit Data from Receiver Channel PMA
From Transmitter Gear Box
From Receiver Deserializer
Loopback
RX Bit Reversal

In addition to providing the bus width adaptation, the receiver gear box provides the receiver bit reversal feature.

Receiver Bit Reversal

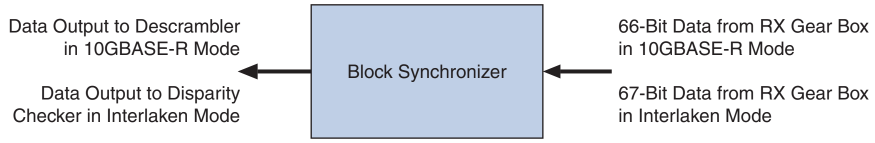
The receiver gear box allows bit reversal of received data. This feature is similar to the transmitter bit reversal feature. Some protocols, such as Interlaken, require the bit reversal feature. For more information, refer to "Transmitter Bit Reversal" on page 1–17.

Block Synchronizer

The block synchronizer (Figure 1–31) is designed towards both the Interlaken protocol specification and the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

The block synchronizer determines the block boundary of a 66-bit word in the case of 10GBASE-R or a 67-bit word in the case of Interlaken. The incoming data stream is slipped one bit at a time until a valid synchronization header (bits 65 and 66) is detected in the received data stream. After the predefined number of synchronization headers (as required by the protocol specification) is detected, it asserts the status signal to other receiver PCS blocks down the receiver datapath and to the FPGA fabric.

Figure 1–31. Block Synchronizer


Data Output to Descrambler in 10GBASE-R Mode
Data Output to Disparity Checker in Interlaken Mode
Block Synchronizer
66-Bit Data from RX Gear Box in 10GBASE-R Mode
67-Bit Data from RX Gear Box in Interlaken Mode

Disparity Checker

The design of the disparity checker is based on the Interlaken protocol specifications. After word synchronization is achieved, the disparity checker monitors the status of the 67th bit of the incoming word and determines whether or not to invert bits `[63:0]` of the received word.

☞ The disparity checker is only used in the Interlaken configuration.

Table 1–14 interprets the MSB in the 67-bit word.

Table 1–14. Interpretation of the MSB in the 67-Bit Payload for Stratix V Devices

MSB	Interpretation
0	Bits `[63:0]` are not inverted; the receiver may process this word without modification
1	Bits `[63:0]` are inverted; the receiver must invert the word to achieve the original word before processing it

Descrambler

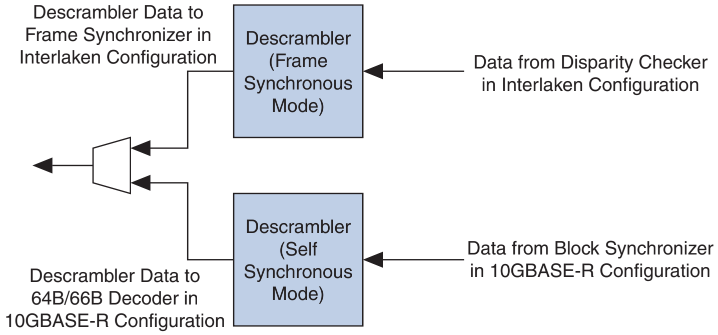
The descrambler descrambles received data per the protocol specifications supported by the 10G PCS.

The descrambler operates in two modes:

- Frame synchronous
- Self synchronous

Figure 1–32 shows the descrambler.

Figure 1–32. Descrambler


Descrambler Data to Frame Synchronizer in Interlaken Configuration
Descrambler (Frame Synchronous Mode)
Data from Disparity Checker in Interlaken Configuration
Descrambler (Self Synchronous Mode)
Data from Block Synchronizer in 10GBASE-R Configuration
Descrambler Data to 64B/66B Decoder in 10GBASE-R Configuration

Frame Synchronous Mode

Use frame synchronous mode in the Interlaken configuration only. When block synchronization is achieved, the descrambler uses the scrambler seed from the received scrambler state word. This block also forwards the current descrambler state to the frame synchronizer.

Self Synchronous Mode

Use self synchronous mode in the 10GBASE-R configuration.

Frame Synchronizer

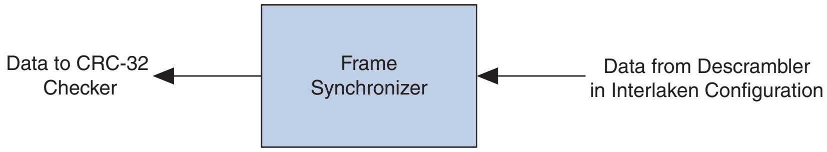
The frame synchronizer block is supported in the Interlaken configuration only.

The frame synchronizer block obtains lock by looking for four synchronization words in consecutive metaframes. After synchronization, it monitors the scrambler word in the metaframe and de-asserts the lock signal after three consecutive mismatches and starts the synchronization process again. The lock status is available to the FPGA fabric.

☞ The frame synchronizer is only used in the Interlaken configuration.

Figure 1–33 shows the frame synchronizer.

Figure 1–33. Frame Synchronizer



Bit-Error Rate (BER) Monitor

The BER monitor block is designed towards the 10GBASE-R protocol specification as described in 802.3-2008 clause-49. After the block lock is achieved, the BER monitor starts to count the number of invalid synchronization headers within a 125-µs period. If more than 16 invalid synchronization headers are observed in a 125-µs period, the BER monitor provides the status signal to the FPGA fabric, indicating a high bit error rate condition.

64B/66B Decoder

The 64B/66B decoder block is designed towards the 10GBASE-R protocol specification as described in IEEE 802.3-2008 clause-49.

This block contains two sub-blocks:

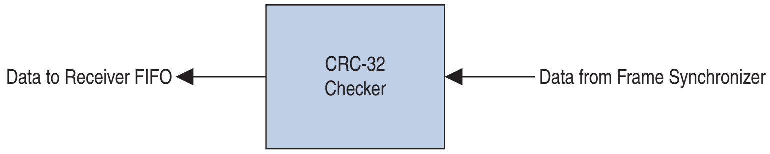
- 64B/66B decoder
- Receiver state machine

The 64B/66B decoder converts the received data from the descrambler into 64-bit data and 8-bit control characters. The receiver state machine monitors the status signal from the BER monitor. If it is asserted, the receiver state machine sends local fault ordered sets to the FPGA interface.

CRC-32 Checker

The CRC-32 checker block is designed to support the Interlaken protocol. The CRC-32 checker calculates the CRC from the incoming data and compares it to the CRC value sent in the diagnostic word. The CRC error signal is provided to the FPGA fabric. Figure 1–34 shows the CRC-32 checker.

Figure 1–34. CRC-32 Checker



Receiver FIFO

The receiver FIFO block operates in different modes based on the transceiver datapath configuration, as listed in Table 1–15.

Table 1–15. Receiver FIFO Operating Modes for Stratix V Devices

Configuration	Receiver FIFO Mode
10GBASE-R	"Clock Compensation Mode"
Interlaken	"Generic Mode"
Custom	"Phase Compensation Mode"

The Quartus II software automatically selects the receiver FIFO mode for the configuration used.

Clock Compensation Mode

The receiver FIFO is configured in clock compensation mode for the 10GBASE-R configuration. In this mode, the FIFO deletes idles OR ordered sets and inserts only idles to compensate up to a ±100 PPM clock difference between the remote transmitter and the local receiver.

Generic Mode

The receiver FIFO is configured in generic mode for the Interlaken configuration. In this mode, the receiver FIFO provides the FIFO partially empty and FIFO full status signals to the FPGA fabric to control the read side of the FIFO.

Phase Compensation Mode

The receiver FIFO is configured in phase compensation mode for the 10G custom configuration. In this mode, the FIFO compensates for the phase difference between the FIFO write clock and the read clock.

Transmitter 10G PCS Datapath

The transmitter channel datapath shown in Figure 1–29 on page 1–33 consists of the following blocks:

- "Transmitter FIFO"
- "Frame Generator" on page 1–38
- "CRC-32 Generator" on page 1–39
- "64B/66B Encoder" on page 1–40
- "Scrambler" on page 1–40
- "Disparity Generator" on page 1–41
- "Transmitter Gear Box" on page 1–41

Transmitter FIFO

The transmitter FIFO provides an interface between the transmitter channel PCS and the FPGA fabric.

In a 10GBASE-R configuration, the transmitter FIFO receives data from the FPGA fabric. The data output from this block goes to the 64B/66B encoder.

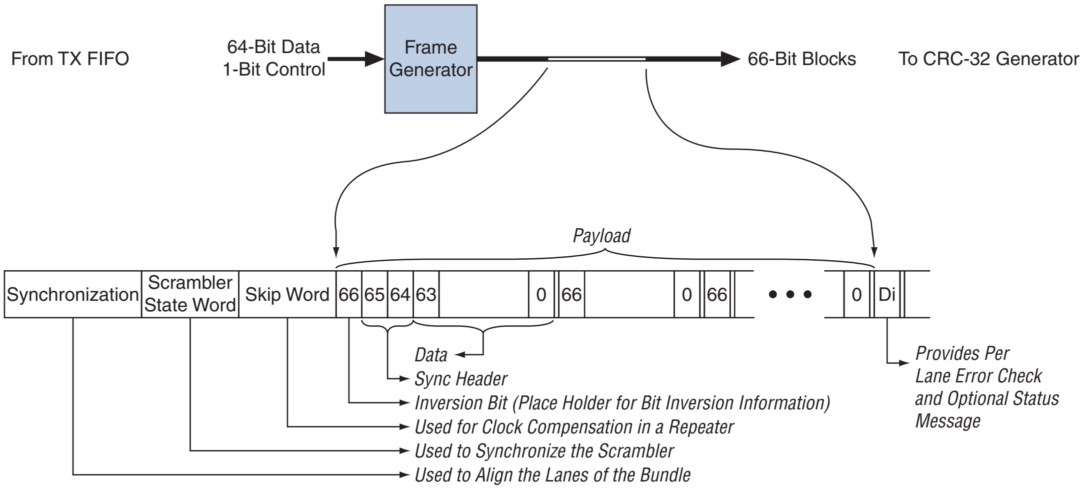
In an Interlaken configuration, the transmitter FIFO sends a control signal to indicate whether it is ready to receive data from the FPGA fabric. The user logic sends the data to the FIFO only if this signal is asserted. Data output from the transmitter FIFO block goes to the frame generator in this configuration.

Frame Generator

The frame generator block (Figure 1–35) is designed to support the Interlaken protocol. This block takes the data from the transmitter FIFO and encapsulates the payload and the burst/idle control words from the FPGA fabric with the framing layer's control words, such as the synchronization word, scrambler state word, skip word, and diagnostic word, to form a metaframe. The Interlaken PHY IP MegaWizard Plug-In Manager interface provides an option to set the metaframe length.

The frame generator is only used in the Interlaken configuration.

Figure 1–35. Frame Generator


From TX FIFO
64-Bit Data
1-Bit Control
Frame Generator
66-Bit Blocks
To CRC-32 Generator
Payload
Synchronization
Scrambler State Word
Skip Word
66
65
64
63
0
66
0
66
• • •
0
Di
Data
Sync Header
Inversion Bit (Place Holder for Bit Inversion Information)
Used for Clock Compensation in a Repeater
Used to Synchronize the Scrambler
Used to Align the Lanes of the Bundle
Provides Per Lane Error Check and Optional Status Message

CRC-32 Generator

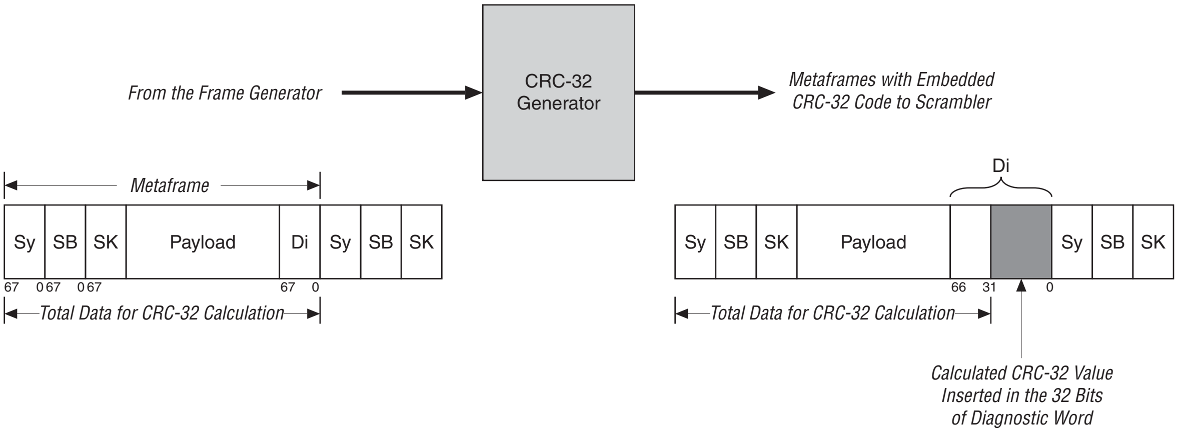
The CRC-32 generator (Figure 1–36) is designed to support the Interlaken protocol. The CRC-32 generator block receives data from the frame generator and calculates the cyclic redundancy check (CRC) code for each block of data. This value is stored in the CRC32 field of the diagnostic word.

The CRC-32 calculation covers the complete metaframe including the diagnostic word, except the following:

- bits `[66:64]` of each word
- 58-bit scrambler state within the scrambler state word
- 32-bit CRC-32 field within the diagnostic word

The CRC-32 generator is only used in the Interlaken configuration.

Figure 1–36. CRC-32 Generator


From the Frame Generator
CRC-32 Generator
Metaframes with Embedded CRC-32 Code to Scrambler
Metaframe
Sy
SB
SK
Payload
Di
Sy
SB
SK
67
0 67
0 67
67
0
Total Data for CRC-32 Calculation
Di
Sy
SB
SK
Payload
Sy
SB
SK
66
31
0
Total Data for CRC-32 Calculation
Calculated CRC-32 Value Inserted in the 32 Bits of Diagnostic Word

64B/66B Encoder

The 64B/66B encoder (Figure 1–37) is designed for the 10GBASE-R protocol specification as described in the IEEE 802.3-2008 clause-49.

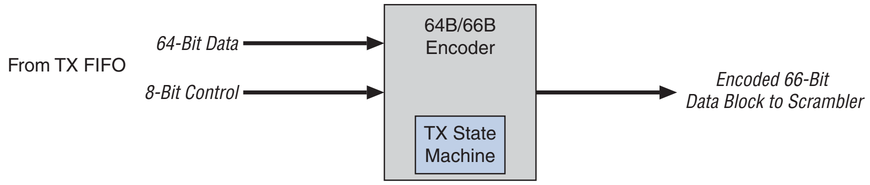
The block contains two sub-blocks:

- 64B/66B Encoder
- Transmitter State Machine

The 64B/66B encoder block (Figure 1–37) receives data from the transmitter FIFO. It encodes the 64-bit data and 8-bit control characters to the 66-bit data block required by the 10GBASE-R configuration. The transmit state machine in this block checks the validity of the 64-bit data from the MAC layer and ensures proper block sequencing.

☞ The 64B/66B encoder is only used in the 10GBASE-R configuration.

Figure 1–37. 64B/66B Encoder



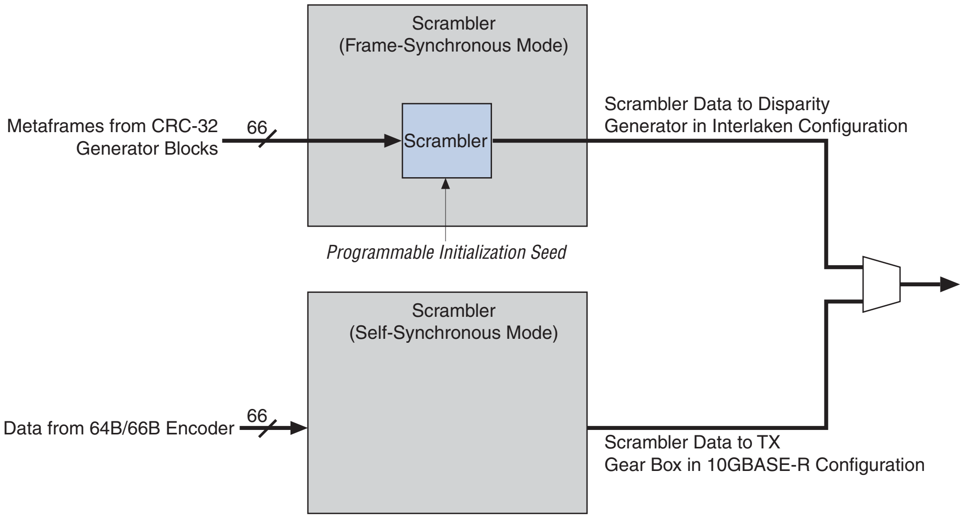
Scrambler

Long sequences of zeros or ones and repetition of data patterns in the data stream can cause interference with adjacent channels and electromagnetic interference (EMI). Data scrambling reduces these effects.

The scrambler (Figure 1–38) operates in two modes:

- Frame synchronous mode—used in the Interlaken configuration
- Self synchronous mode—used in the 10GBASE-R configuration as specified in the IEEE 802.3-2008 clause-49

Figure 1–38. Scrambler


Scrambler
(Frame-Synchronous Mode)
Metaframes from CRC-32 Generator Blocks
66
Scrambler
Scrambler Data to Disparity Generator in Interlaken Configuration
Programmable Initialization Seed
Scrambler
(Self-Synchronous Mode)
Data from 64B/66B Encoder
66
Scrambler Data to TX Gear Box in 10GBASE-R Configuration

Disparity Generator

The disparity generator block is designed for the Interlaken protocol specification and provides a DC-balanced data output. It receives data from the scrambler and inverts the running disparity to stay within the ±96-bit boundary. To ensure this running disparity requirement, it inverts bits `[63:0]` and sets bit 66 to indicate inversion.

☞ The disparity generator is only used in the Interlaken configuration.

Table 1–16 interprets the MSB of the 67-bit payload.

Table 1–16. Interpretation of the MSB in the 67-Bit Payload for Stratix V Devices

MSB	Interpretation
0	Bits `[63:0]` are not inverted; the receiver may process this word without modification
1	Bits `[63:0]` are inverted; the receiver must invert the word to achieve the original word before processing it

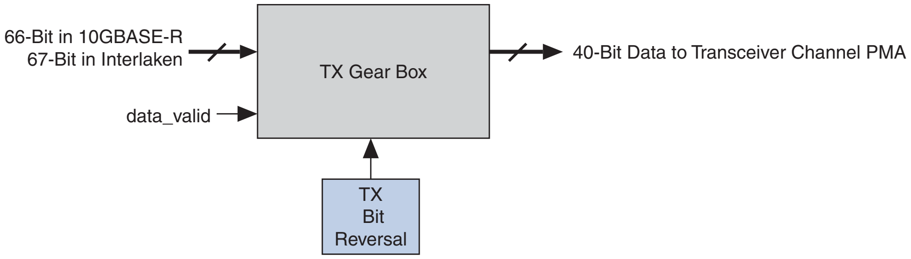
Transmitter Gear Box

The transmitter gear box adapts the PCS data width to a smaller bus width for interfacing with the PMA. Because of the transmitter gear box, the difference in the bus widths between the PCS and PMA is made transparent to the logic in the FPGA fabric.

For the supported PMA-PCS widths, refer to the *Transceiver Protocol Configurations in Stratix V Devices* chapter. For custom configurations, refer to the *Transceiver Custom Configurations in Stratix V Devices* chapter.

Figure 1–39 shows the transmitter gear box with data widths for the Interlaken and 10GBASE-R configurations.

Figure 1–39. Transmitter Gear Box



In addition to providing bus width adaptation, the transmitter gear box provides the transmitter bit reversal and bit-slip features.

Transmitter Bit Reversal

The transmitter gear box also provides the ability to reverse the order of transmitted bits. By default, the transmitter first sends out the LSB of a word. Some protocols, such as Interlaken, require that the MSB of a word (bit 66 in a word `[66:0]`) is transmitted first. When you enable the transmitter bit reversal feature, the parallel input to the gear box is swapped and the MSB is sent out first. The Quartus II software automatically sets the bit reversal feature for the Interlaken configuration.

Transmitter Bit-Slip

The transmitter bit-slip feature allows you to compensate for the channel-to-channel skew between multiple transmitter channels by slipping the data sent to the PMA. The maximum number of bits slipped is controlled from the FPGA fabric and is equal to the width of the PMA-PCS minus 1. This feature is supported only in 10G custom configurations.

For more information, refer to the *10G Low Latency Configuration* section in the *Transceiver Custom Configurations in Stratix V Devices* chapter.

Bonded Configuration

The high-speed serial clock and low-speed parallel clock skew between channels and unequal latency in the transmitter phase compensation FIFO contribute to transmitter channel-to-channel skew. Bonded transmitter datapath clocking provides low channel-to-channel skew when compared with non-bonded channel configurations.

- Bonded channel configurations—the serial clock and parallel clock for all bonded channels are generated by the transmit PLL and central clock divider, resulting in lower channel-to-channel clock skew.

  The transmitter phase compensation FIFO in all bonded channels share common pointers and control logic generated in the central clock divider, resulting in equal latency in the transmitter phase compensation FIFO of all bonded channels. The lower transceiver clock skew and equal latency in the transmitter phase compensation FIFOs in all channels provide lower channel-to-channel skew in bonded channel configurations.

- Non-bonded channel configurations—the parallel clock in each channel are generated independently by its local clock divider, resulting in higher channel-to-channel clock skew.

  The transmitter phase compensation FIFO in each non-bonded channel has its own pointers and control logic that can result in unequal latency in the transmitter phase compensation FIFO of each channel. The higher transceiver clock skew and unequal latency in the transmitter phase compensation FIFO in each channel can result in higher channel-to-channel skew.

The Stratix V transceivers support various bonded and non-bonded transceiver clocking configurations. For more information, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

PLL Sharing

Two different protocol configurations in a Quartus II design can be merged to share the same CMU PLL resources. The two configurations must fit in the same transceiver bank and the input `refclk` and PLL output frequencies must be the same.

Document Revision History

Table 1–17 lists the revision history for this chapter.

Table 1–17. Document Revision History

Date	Version	Changes
May 2011	1.4	■ Chapter moved to Volume 3. ■ Minor text edits.
December 2010	1.3	■ Updated Figure 1–16. ■ Minor graphic edits
December 2010	1.2	■ Updated to include Quartus II version 10.1 information. ■ Reorganized chapter information. ■ Added the "Channel PLL Used as a CMU PLL (Transmitter PLL)" and "Auxiliary Transmit (ATX) PLL Architecture" sections. ■ Added Figure 1–11. ■ Updated the "Bonded Configuration" section. ■ Updated Figure 1–2, Figure 1–3, Figure 1–4, Figure 1–5, Figure 1–7, Figure 1–8, Figure 1–14, Figure 1–16, and Figure 1–29. ■ Reversed Rx directional flow in Figure 1–8, Figure 1–18, Figure 1–19, Figure 1–20, Figure 1–21, and Figure 1–22. ■ Minor text edits.
July 2010	1.1	Initial release in the *Stratix V Device Handbook*.
April 2010	1.0	Initial release for EAP.

2. Transceiver Clocking in Stratix V Devices

SV52003-1.2

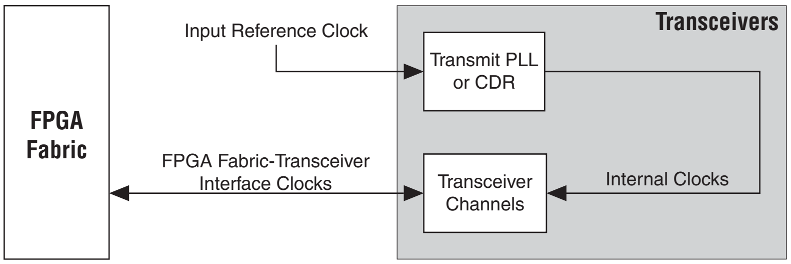
This chapter provides detailed information about the Stratix® V transceiver clocking architecture.

The clocking architecture chapter is divided into three sections:

- "Input Reference Clocking"—Describes how the reference clock to the transmit phase-locked loop (PLL), ATX PLL, and clock data recovery (CDR) is provided to generate the clocks required for transceiver operation.
- "Internal Clocking"—Describes the clocking architecture internal to the transceiver.
- "FPGA Fabric-Transceiver Interface Clocking"—Describes the clocking options available when the transceiver interfaces with the FPGA fabric.

Figure 2–1 shows an overview of the clocking architecture.

Figure 2–1. Transceiver Clocking Architecture Overview



For upcoming clocking related features, refer to the *Upcoming Stratix V Device Features* document.

Input Reference Clocking

Each transceiver channel has a channel PLL that can be configured as a transmitter clock multiplier unit (CMU) PLL or receiver CDR. In a CMU PLL configuration, the PLL uses the input reference clock to generate a serial clock. In receiver CDR configuration, the PLL locks to the input reference clock in lock-to-reference (LTR) mode. The ATX PLL also uses the input reference clock to synthesize a serial clock.

☞ The input reference clock is different from the clock forwarded from the transceiver to the FPGA fabric that clocks the transceiver logic and FPGA fabric-transceiver interface.

Input Reference Clock Sources

The transceiver channel PLL and ATX PLL derive the input clock from a dedicated `refclk` pin, a fractional PLL, or through the reference clock network. Figure 2–2 shows an overview of the input reference clock input to the transceiver channel.

Figure 2–2. Input Reference Clock Sources to Transceiver Channel

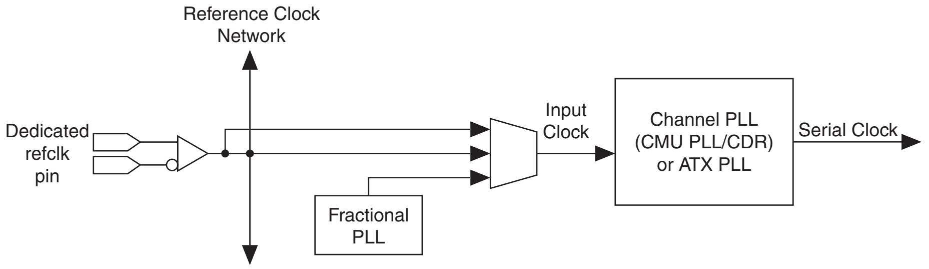


Table 2–1 lists the jitter performance of the input reference clock resources for transmit PLLs.

Table 2–1. Input Reference Clock Source Performance *(Note 1)*, *(2)*

Reference Clock Source	Jitter Performance
Dedicated `refclk` pin	1
Reference clock network	2
Fractional PLL	3

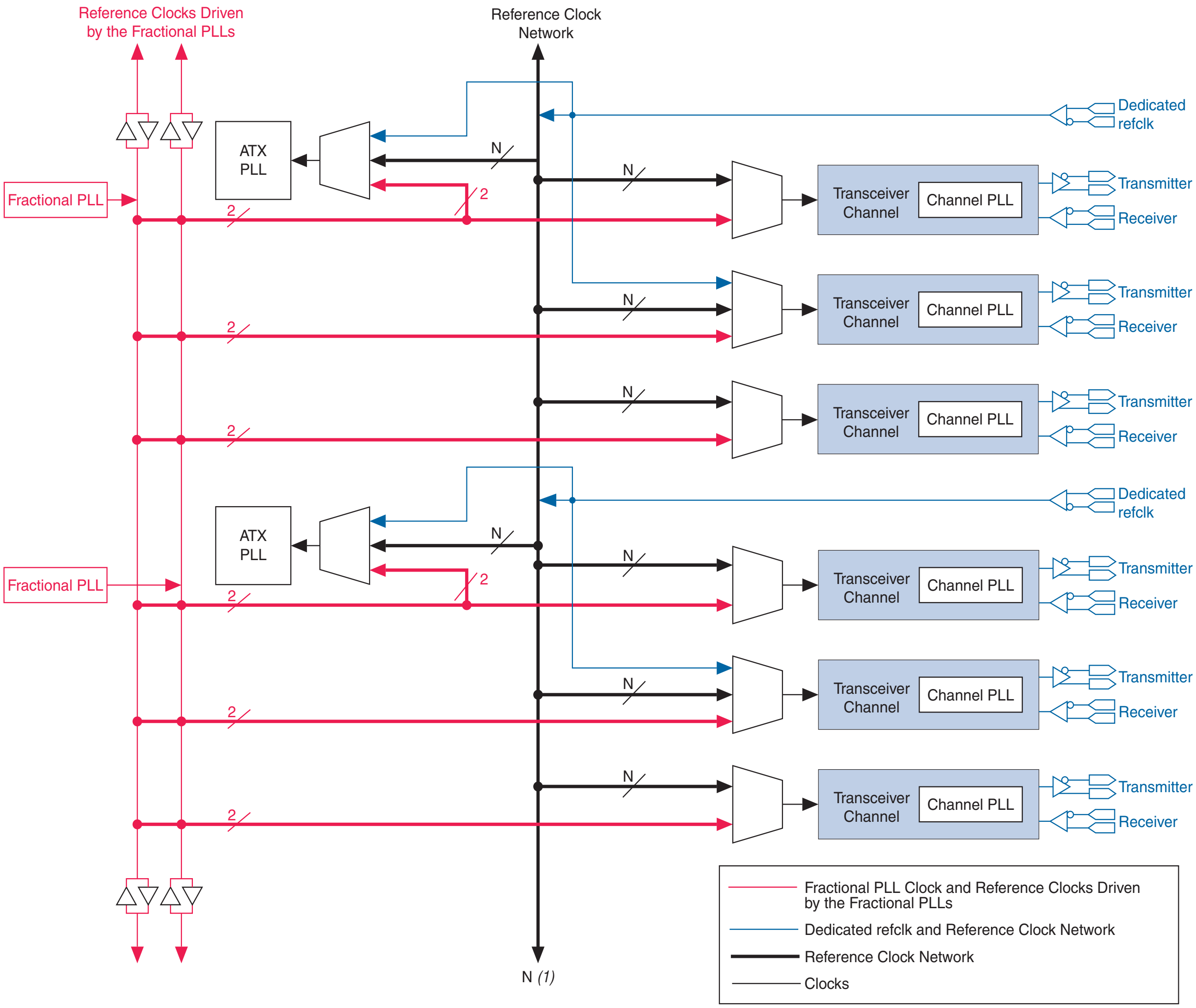
Notes to Table 2–1:

(1) 1 indicates the best jitter performance.

(2) The performance is pending characterization.

Figure 2–3 shows the input reference clock sources for a transceiver bank.

Figure 2–3. Input Reference Clock Sources for Transceiver Channels in a Transceiver Bank


Reference Clocks Driven by the Fractional PLLs
Reference Clock Network
Dedicated refclk
ATX PLL
Fractional PLL
N
2
Transceiver Channel
Channel PLL
Transmitter
Receiver
N (1)
Fractional PLL Clock and Reference Clocks Driven by the Fractional PLLs
Dedicated refclk and Reference Clock Network
Reference Clock Network
Clocks

Note to Figure 2–3:

(1) N = number of transceiver channels on a side divided by 3, which is equal to the number of dedicated `refclk` pins.

Dedicated refclk Pins

Stratix V devices have one dedicated `refclk` pin for each group of three transceiver channels. Every dedicated reference clock pin drives a clock network spanning the side of the device.

Dedicated refclk Pins Using the Reference Clock Network

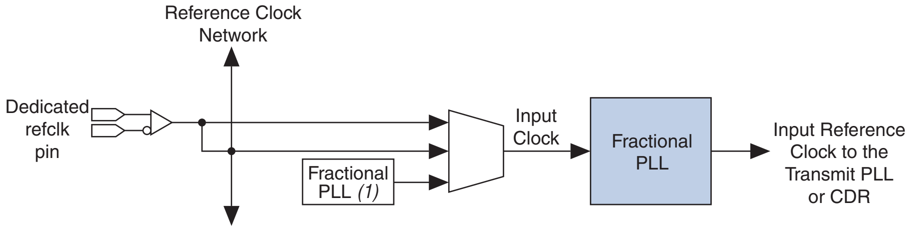
Each dedicated `refclk` pin can drive any transmit PLL on the same side of the device through the reference clock network. Designs using multiple transmit PLLs that require the same reference clock frequency and are located along the same side of the device can share the same dedicated `refclk` pin.

Fractional PLLs

Stratix V devices provide a fractional PLL for each group of three transceiver channels. Each fractional PLL drives one of two clock lines spanning the side of the device that can provide a clock to any transmit PLL or CDR on the same side of the device. A fractional PLL enables you to use an input reference clock in your system that is not supported by the transmit PLL or CDR to synthesize a supported input reference clock required by the transmit PLL or CDR. For the Quartus II software 11.0, only the integer mode is supported. Integer mode allows you to synthesize clocks that are integer multiples or factors of itself. For example, if you have a 10 MHz clock available, you can synthesize a clock of 5, 20, and 30 MHz.

Figure 2–4 shows the input clock sources for the fractional PLL.

Figure 2–4. Fractional PLL Input Clock Sources



Note to Figure 2–4:

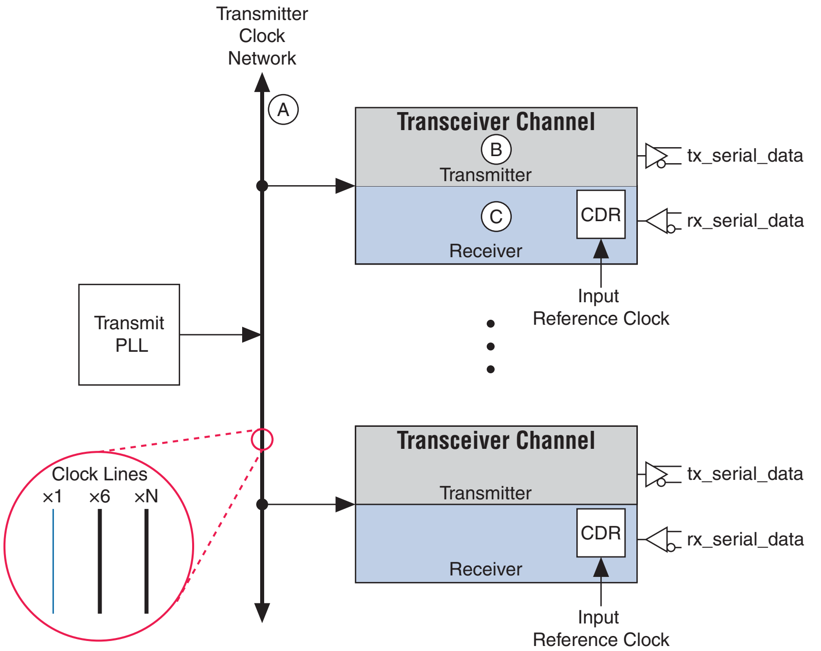
(1) A fractional PLL can provide a clock source to another fractional PLL through the reference clock lines driven by the fractional PLLs.

Internal Clocking

This section describes the clocking architecture internal to Stratix V transceivers. Different physical coding sublayer (PCS) configurations and channel bonding options result in various transceiver clock paths. Figure 2–5 shows the following sections of the transceiver internal clocking:

- "Transmitter Clock Network" (A in Figure 2–5)
- "Transmitter Clocking" (B in Figure 2–5)
- "Receiver Clocking" (C in Figure 2–5)

Figure 2–5. Internal Clocking



The reference clock from one of the sources shown in Figure 2–2 on page 2–2 is fed to a transmit PLL. The transmit PLL could be either a CMU PLL or an ATX PLL. The transmit PLL generates a serial clock that is distributed using a transmitter clock network to the transceiver channels.

☞ Clocking described in this section is internal to the transceiver, and clock routing is primarily performed by the Quartus II software based on the transceiver configuration selected.

Transmitter Clock Network

The transmitter clock network routes the clock from the transmit PLL to the transmitter channel (as shown in Figure 2–5) and provides two clocks to the transmitter channel:

- Serial clock—high-speed clock for the serializer
- Parallel clock—low-speed clock for the serializer and the PCS

Stratix V transceivers support various non-bonded and bonded transceiver clocking configurations. If you use a bonded configuration, both the serial clock and parallel clock are routed from the transmit PLL to the transmitter channel. If you use a non-bonded configuration, then only the serial clock is routed from the transmit PLL to the transmitter channel and the parallel clock is generated by the local clock divider or central clock divider of each channel.

Transmitter Clock Network Architecture

The transmitter clock network consists of the two types of dedicated clocking resources:

- Non-bonded configurations
  - ×1 clock lines
- Bonded configurations
  - ×6 clock lines
  - ×N clock lines

☞ The Quartus II software performs the clock routing related to the transmitter clock network based on the transceiver configuration selected.

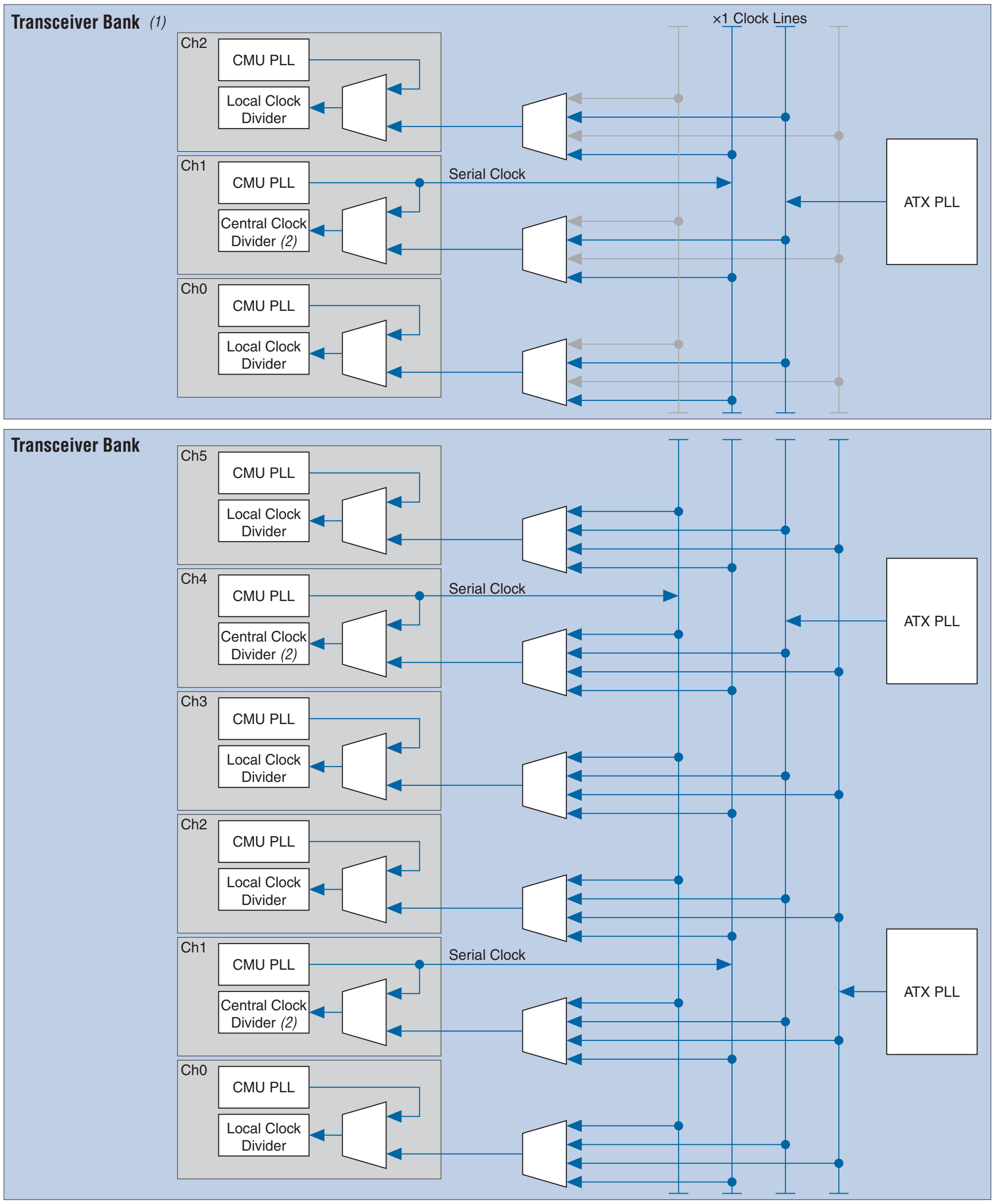
×1 clock lines are used for non-bonded configurations and only route the serial clock lines from the transmit PLL to the clock divider of the transceiver channels. Figure 2–6 shows the ×1 clock lines. The following resources can drive the ×1 clock lines:

- Channel PLLs (configured as a CMU PLL) of channel 1 and 4 in a transceiver bank
- ATX PLLs in the transceiver bank

The ×1 clock lines can drive the local clock divider and central clock divider of any channel within a transceiver bank.

☞ The channel PLL can be used to drive the local clock divider or the central clock divider of its own channel if it is configured as a CMU PLL. However, you will not be able to use the channel PLL as a CDR. Without a CDR, you can only use the channel as a transmitter channel.

Figure 2–6. ×1 Clock Lines Used for Non-Bonded Configuration



Notes to Figure 2–6:

(1) Stratix V devices 5SGXB5, 5SGXB6, 5SGSB7, and 5SGSB8 have one transceiver bank on each side with only three transceiver channels. For more information, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

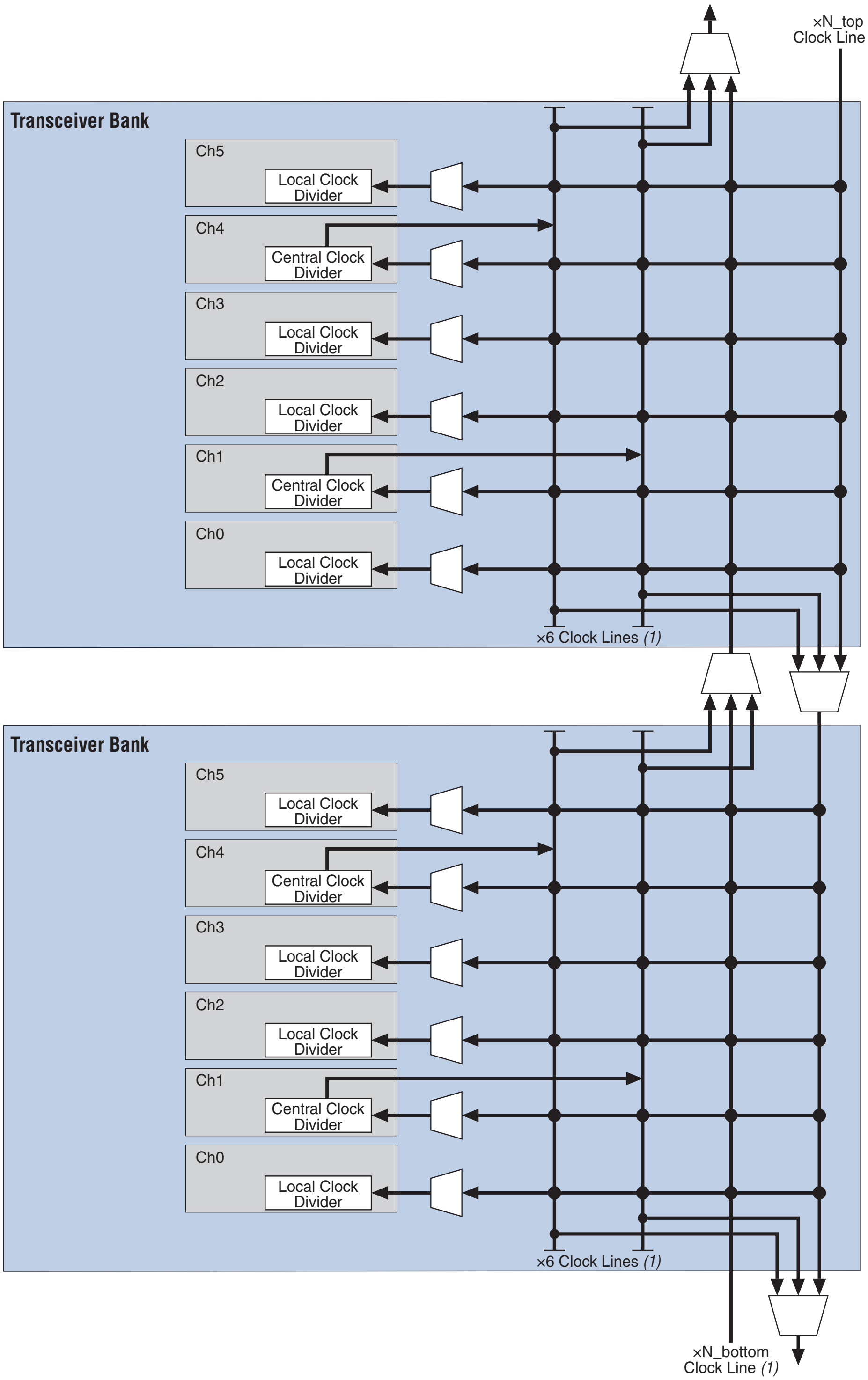
(2) You can use the central clock divider as a local clock divider.

The ×6 and ×N clock lines are used for bonded configurations and route both the serial clock and parallel clock from the central clock dividers to the transceiver channels.

The ×6 and ×N clock lines can be used to route the serial clock from the central clock dividers to the transceiver channels for non-bonded configurations to conserve the number of transmit PLLs used in the design.

Figure 2–7 shows both ×6 and ×N clock lines. You can only drive the ×6 clock lines with a central clock divider of channel 1 and 4 in a transceiver bank. You can drive the ×N clock lines with the ×6 clock lines. ×6 clock lines can drive any channel within a transceiver bank. The ×N clock lines span the entire side of the device and can directly drive any channel within or outside a transceiver bank.

Figure 2–7. ×6 and ×N Clock Lines Used for Bonded Configurations



Note to Figure 2–7:

(1) The clock lines carry both serial and parallel clocks.

The ×N clock lines are currently only supported for PCIe ×8 Gen1 and Gen2 configurations.

Table 2–2 lists the span and data rates supported by clock sources and networks in Stratix V devices.

Table 2–2. Data Rates and Spans Supported Using Stratix V Clock Sources and Clock Networks

Clock Network	Clock Source	Max Data Rate *(1)*	Bonding	Span
×1	■ Ch1 or Ch4 CMU PLL in a transceiver bank ■ ATX PLLs in a transceiver bank	14.1 Gbps	No	Transceiver bank
×6	Central clock dividers in a transceiver bank (in Ch1 or Ch4 only)	14.1 Gbps	Yes	Transceiver bank
×N	Central clock dividers in a transceiver bank through ×6 clock lines (in Ch1 or Ch4 only)	5 Gbps	Yes	Side wide *(2)*

Notes to Table 2–2:

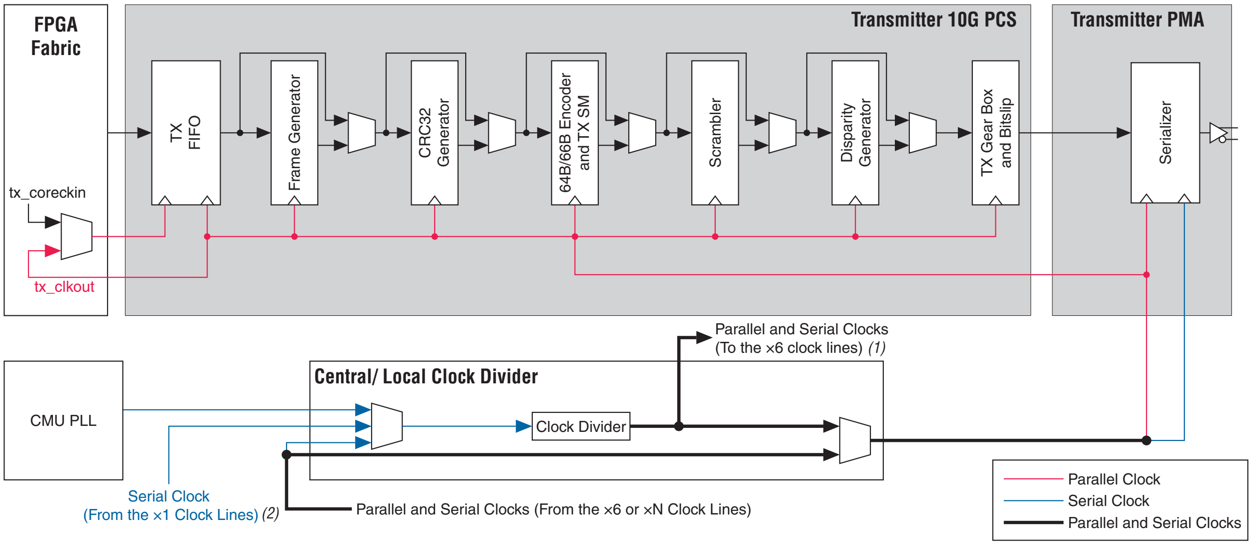
(1) For the fastest speed grade only. For the remaining speed grades, refer to *DC and Switching Characteristics for Stratix V Devices*.

(2) Supported in the PCIe Gen1 and Gen2 ×8 configurations only.

Transmitter Clocking

Transmitter clocking refers to the clocking architecture internal to the transmitter channel of a transceiver. Figure 2–8 shows clocking for the transmitter 10G PCS and transmitter physical medium attachment (PMA).

Figure 2–8. Transmitter 10G PCS Clocking


FPGA Fabric
tx_coreckin
tx_clkout
Transmitter 10G PCS
TX FIFO
Frame Generator
CRC32 Generator
64B/66B Encoder and TX SM
Scrambler
Disparity Generator
TX Gear Box and Bitslip
Transmitter PMA
Serializer
CMU PLL
Central/ Local Clock Divider
Clock Divider
Parallel and Serial Clocks (To the ×6 clock lines) (1)
Serial Clock (From the ×1 Clock Lines) (2)
Parallel and Serial Clocks (From the ×6 or ×N Clock Lines)
Parallel Clock
Serial Clock
Parallel and Serial Clocks

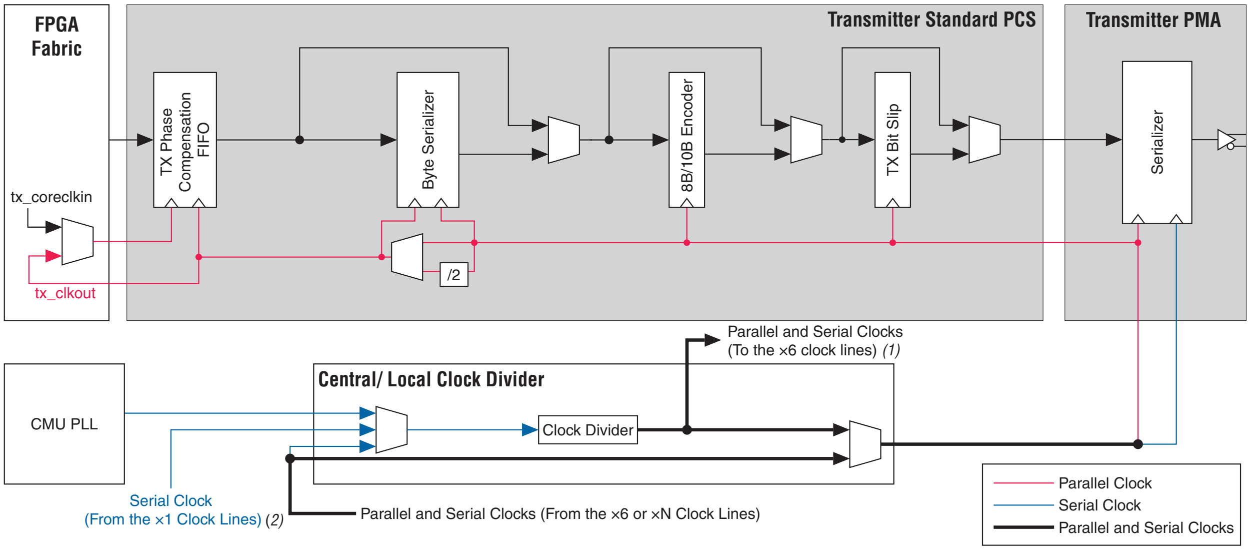
Notes to Figure 2–8:

(1) Only available in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

(2) ×1 clock lines can be driven either by a CMU PLL or ATX PLL.

Figure 2–9 shows clocking for the transmitter standard PCS and transmitter PMA.

Figure 2–9. Transmitter Standard PCS Clocking


FPGA Fabric
Transmitter Standard PCS
Transmitter PMA
TX Phase Compensation FIFO
Byte Serializer
8B/10B Encoder
TX Bit Slip
Serializer
tx_coreclkin
tx_clkout
/2
Parallel and Serial Clocks (To the ×6 clock lines) (1)
Central/ Local Clock Divider
CMU PLL
Clock Divider
Serial Clock (From the ×1 Clock Lines) (2)
Parallel and Serial Clocks (From the ×6 or ×N Clock Lines)
Parallel Clock
Serial Clock
Parallel and Serial Clocks

Notes to Figure 2–9:

(1) Only available in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

(2) ×1 clock lines can be driven either by a CMU PLL or ATX PLL.

As shown in Figure 2–8 and Figure 2–9, the clock divider block provides the serial clock to the serializer of the transmitter PMA and the parallel clock to the transmitter PCS and the serializer of the transmitter PMA.

In the 10G PCS channel, the parallel clock is used by all the blocks up to the read side of the transmitter (TX) FIFO. This clock is also forwarded to the FPGA fabric to interface the FPGA fabric with the transceiver.

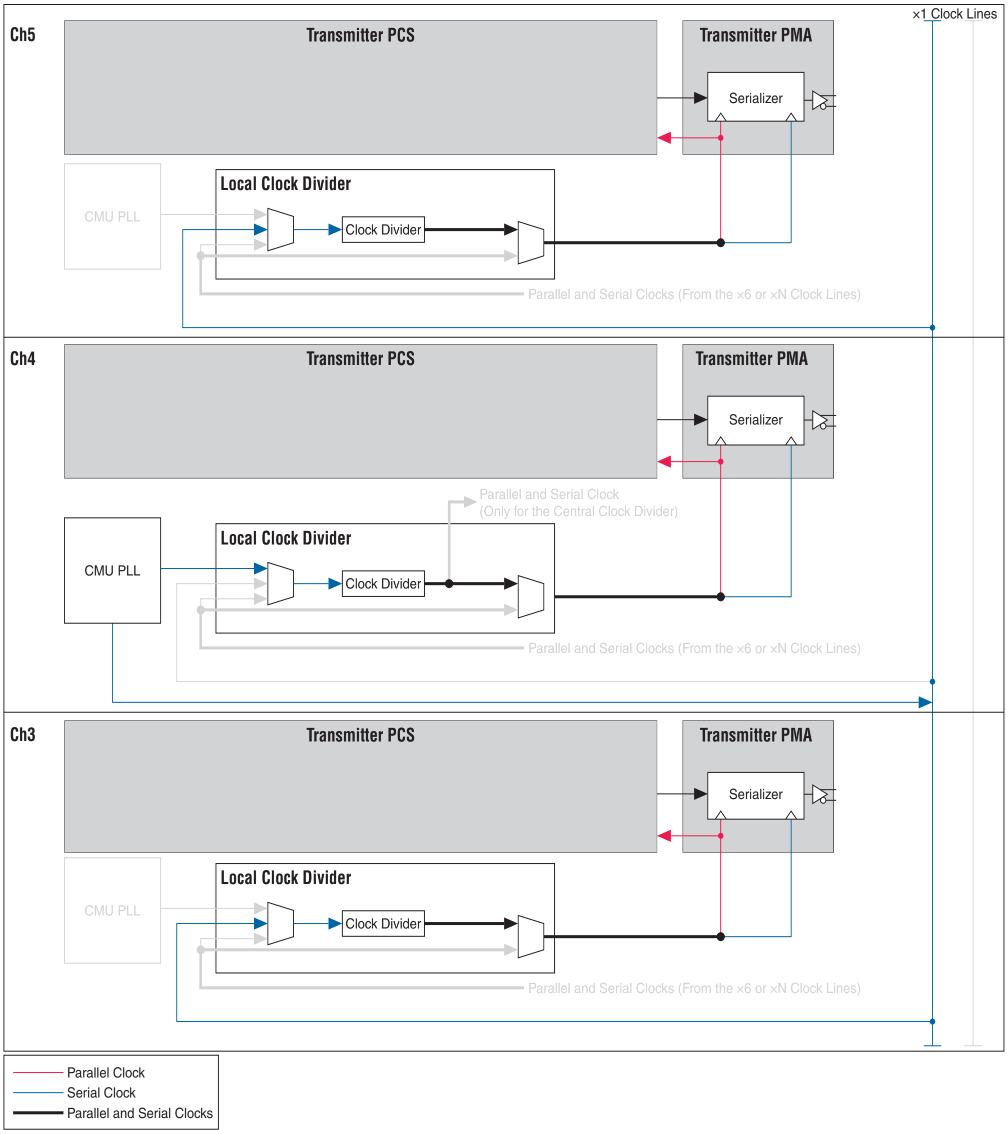
In the standard PCS channel, the parallel clock is used by all the blocks up to the read side of the TX phase compensation FIFO in all configurations that do not use the byte serializer block. For configurations that use the byte serializer block, the clock is divided by a factor of 2 for the byte serializer and the read side of the TX phase compensation FIFO. The clock used to clock the read side of the TX phase compensation FIFO is also forwarded to the FPGA fabric to interface the FPGA fabric with the transceiver.

For more information about clocking schemes used in different configurations, refer to the *Transceiver Protocol Configurations in Stratix V Devices* and *Transceiver Custom Configurations in Stratix V Devices* chapters.

Non-Bonded Channel Configurations

In non-bonded configurations, the parallel clock is generated by the clock divider of individual channels. Figure 2–10 shows three transmit-only channels in non-bonded configuration driven by the channel PLL of channel 4 configured as a CMU PLL driving the ×1 clock line. The clock divider block of each channel generates its own parallel clock by dividing down the serial clock from the ×1 clock line.

Figure 2–10. Three Transmit-Only Channels Configured in Non-Bonded Configuration



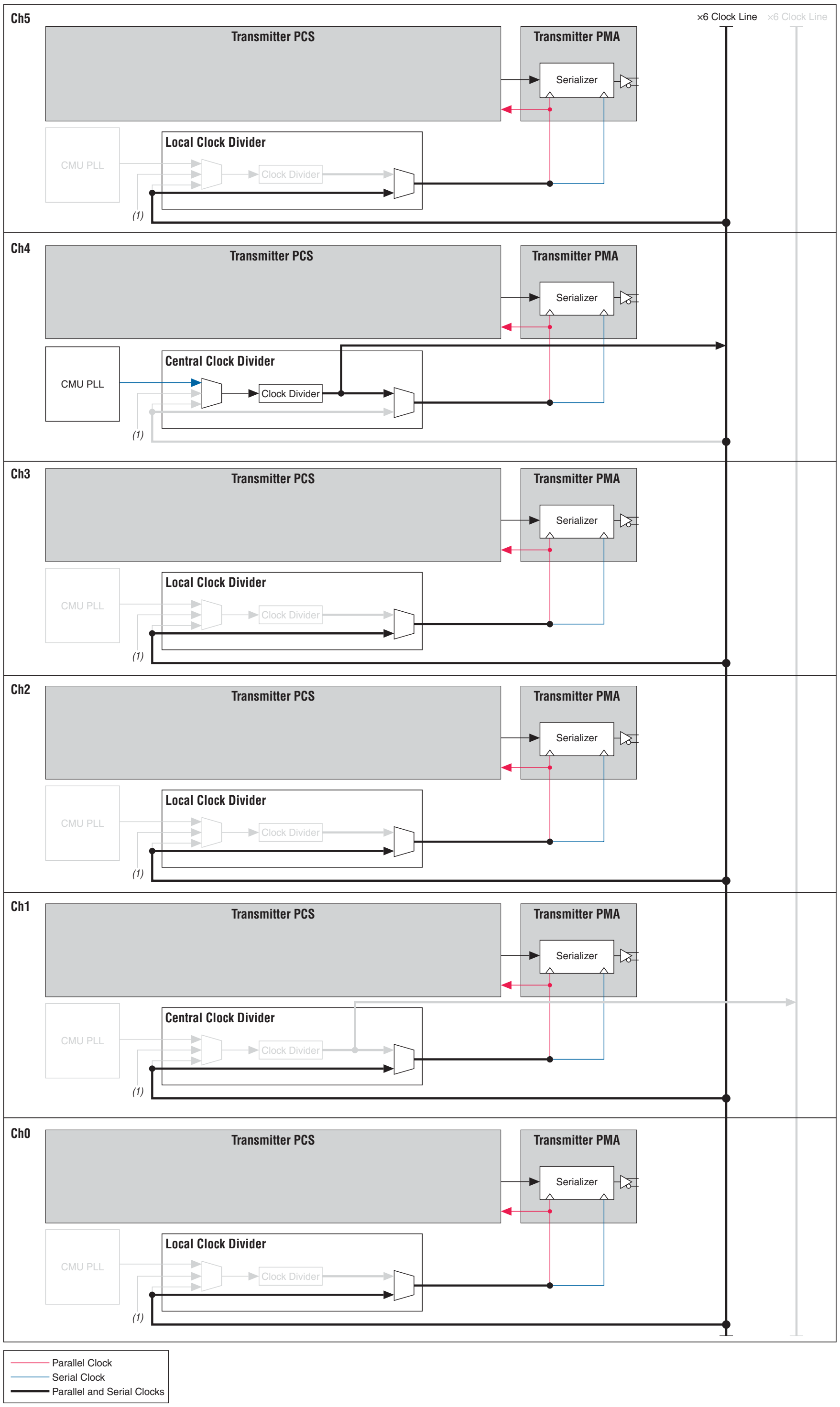
Bonded Channel Configurations

In bonded configurations, both the parallel clock and serial clock are sourced from either the ×6 or ×N clock line. The central clock dividers source the serial clock from a transmit PLL from the same transceiver bank using the ×1 clock line. The central clock divider generates the parallel clock and drives both the serial clock and parallel clock on the ×6 clock line, which can drive the ×N clock line.

☞ Currently, the ×N clock lines are only supported for the PCIe Gen1 and Gen2 ×8 protocols.

Figure 2–11 shows six transmit-only channels configured in bonded configuration and driven by the channel PLL of channel 4 configured as a CMU PLL. The central clock divider of channel 4 generates a parallel clock and drives both the serial clock and parallel clock on the ×6 clock line. All bonded channels source both serial and parallel clocks from the ×6 clock line.

Figure 2–11. Six Transmit-Only Channels Configured in Bonded Configuration



Note to Figure 2–11:

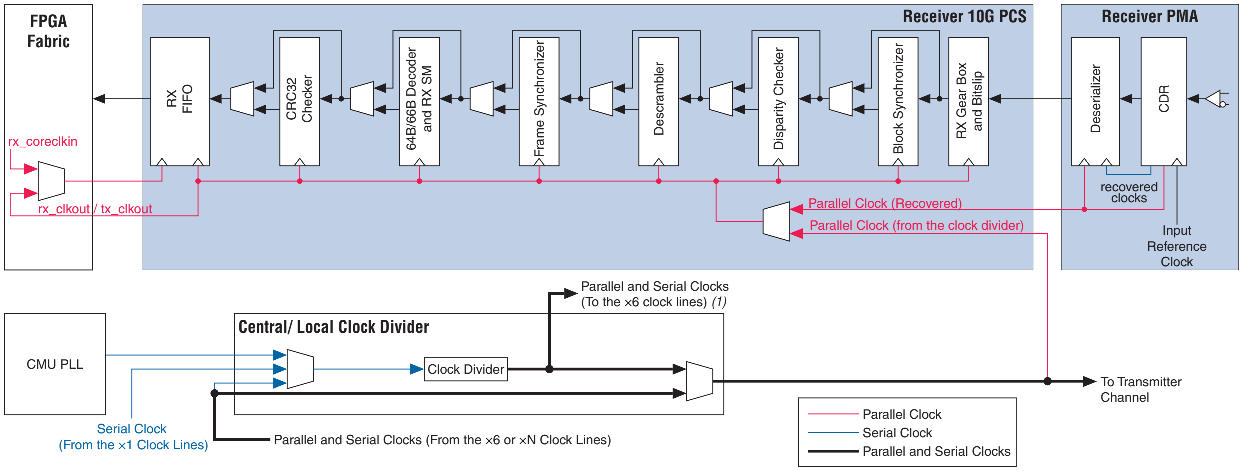
(1) Serial clock from the ×1 clock lines.

For an example of using the ×N clock lines, refer to the PCIe ×8 configuration in the *Transceiver Protocol Configurations in Stratix V Devices* and *Transceiver Custom Configurations in Stratix V Devices* chapters.

Receiver Clocking

Receiver clocking refers to the clocking architecture internal to the receiver channel of a transceiver. Figure 2–12 shows clocking for the receiver 10G PCS receiver and the receiver PMA.

Figure 2–12. Receiver 10G PCS Clocking

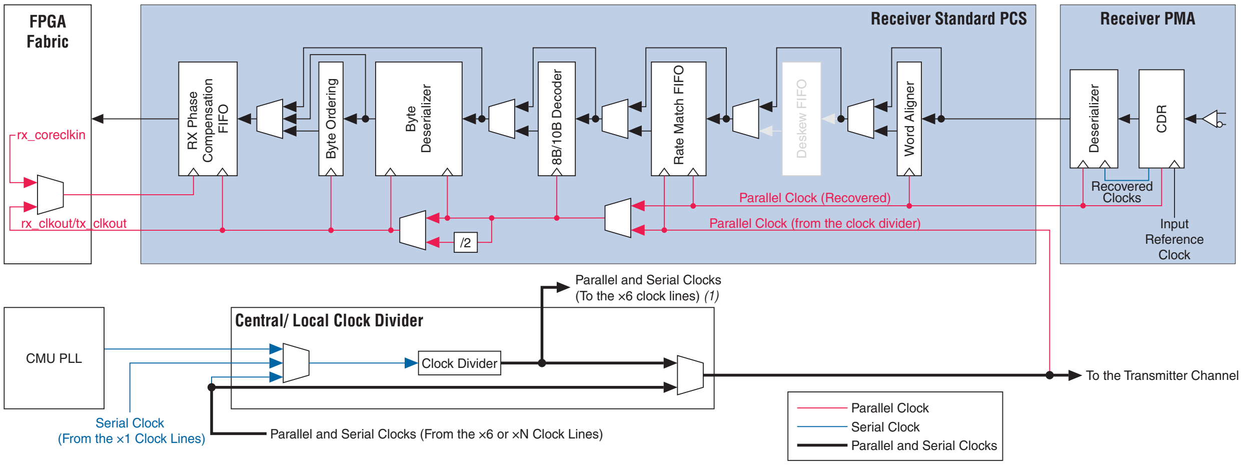


Note to Figure 2–12:

(1) Only available in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

Figure 2–13 shows clocking for the receiver standard PCS and receiver PMA.

Figure 2–13. Receiver Standard PCS Clocking


FPGA Fabric
rx_coreclkin
rx_clkout/tx_clkout
Receiver Standard PCS
RX Phase Compensation FIFO
Byte Ordering
Byte Deserializer
8B/10B Decoder
Rate Match FIFO
Deskew FIFO
Word Aligner
/2
Parallel Clock (Recovered)
Parallel Clock (from the clock divider)
Receiver PMA
Deserializer
CDR
Recovered Clocks
Input Reference Clock
Parallel and Serial Clocks (To the x6 clock lines) (1)
CMU PLL
Central/ Local Clock Divider
Clock Divider
To the Transmitter Channel
Serial Clock (From the x1 Clock Lines)
Parallel and Serial Clocks (From the x6 or xN Clock Lines)
Parallel Clock
Serial Clock
Parallel and Serial Clocks

Note to Figure 2–13:

(1) Only available in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

The CDR in the PMA of each channel recovers the serial clock from the incoming data. The CDR also divides the serial clock (recovered) to generate the parallel clock (recovered). Both clocks are used by the deserializer. The receiver PCS can use the following clocks depending on the configuration of the receiver channel:

- Parallel clock (recovered) from the CDR in the PMA
- Parallel clock from the clock divider used by the transmitter PCS for that channel

Table 2–3 lists the different clock sources that are available for each block in the receiver PCS.

Table 2–3. Clock Sources for All Receiver PCS Blocks

PCS	Block	Clock Source
Standard	Word aligner	Parallel clock (recovered)
	Rate match FIFO	Write side: parallel clock (recovered) Read side: parallel clock from the clock divider
	8B/10B decoder	If rate matcher is not used: parallel clock (recovered) If rate matcher is used: parallel clock from the clock divider
	Byte deserializer	Write side: ■ If rate matcher is not used: parallel clock (recovered) ■ If rate matcher is used: parallel clock from the clock divider Read side: Divided down version of the write side clock depending on the deserialization factor of 1 or 2, also called the parallel clock (divided)
	Byte ordering	Parallel clock (divided)
	Receiver (RX) phase compensation FIFO	Write Side: Parallel clock (divided). This clock is also forwarded to the FPGA fabric Read Side: Clock sourced from the FPGA fabric
10G	All PCS blocks	Regular mode: parallel clock (recovered) Loopback mode: parallel clock from the clock divider *(1)*

Note to Table 2–3:

(1) For more information about loopback mode, refer to the *Transceiver Loopback Support in Stratix V Devices* chapter.

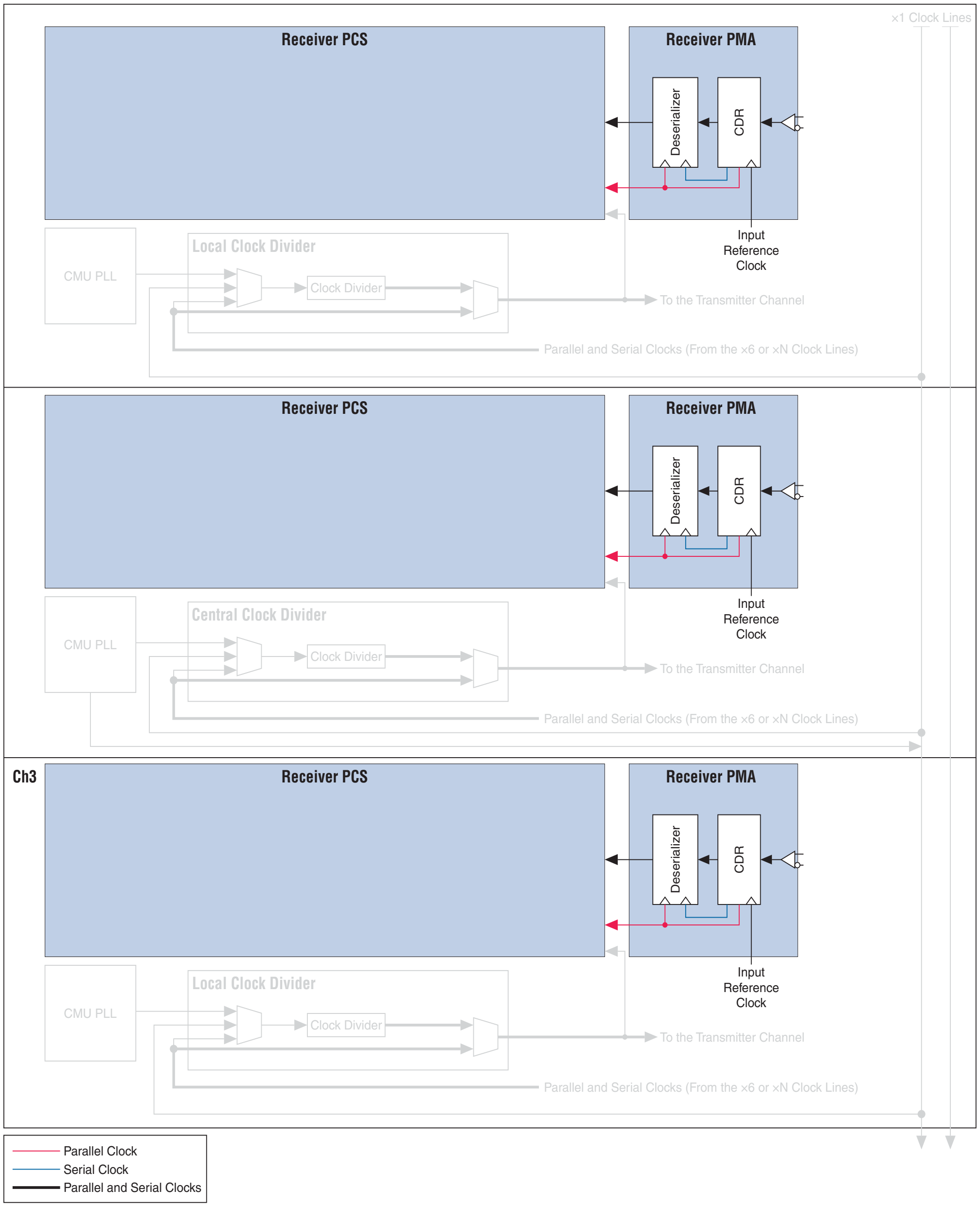
Non-Bonded Channel Configurations

In non-bonded configurations, the receiver standard PCS requires both the parallel clock (recovered) and parallel clock from the clock divider. Depending on the configuration, it may require the parallel clock from the clock divider that is used for the transmitter PCS.

In non-bonded configurations, the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

Figure 2–14 shows three channels, configured in non-bonded configuration, using the receiver standard PCS that do not use a rate match FIFO. The CDR of each channel recovers the serial clock (recovered) from the incoming data and generates the parallel clock (recovered) by dividing the serial clock (recovered). Depending on the configuration, the receiver PCS may also use the parallel clock from the clock divider that is generated by the local clock divider for the transmitter.

Figure 2–14. Three Channels Configured in Non-Bonded Configuration



For more information about the clocking scheme used in different configurations, refer to the *Transceiver Protocol Configurations in Stratix V Devices* and *Transceiver Custom Configurations in Stratix V Devices* chapters.

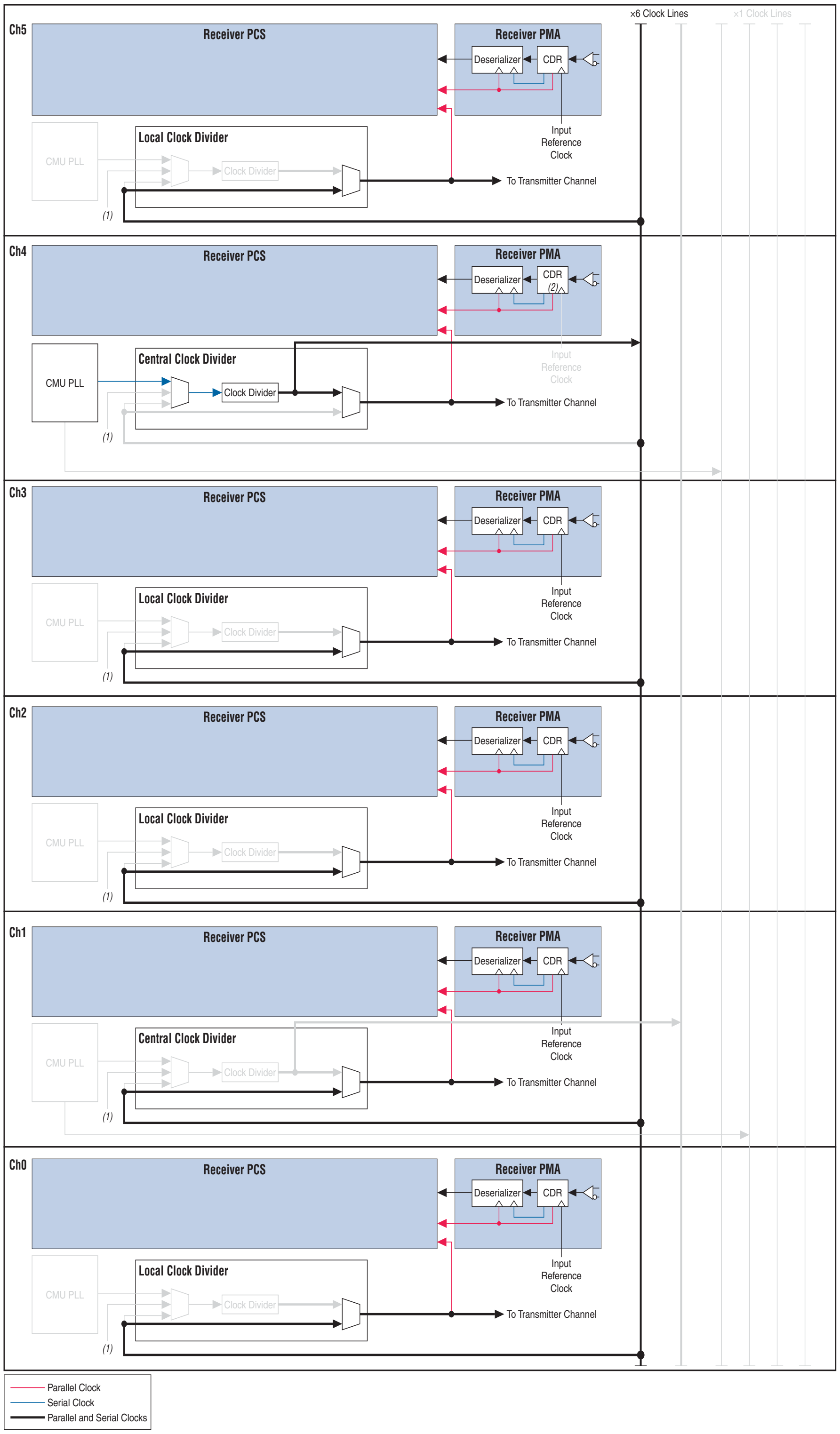
Bonded Channel Configurations

In bonded configurations, the receiver standard PCS requires both the parallel clock (recovered) and parallel clock from the clock divider.

In bonded configurations, the receiver 10G PCS uses only the parallel clock (recovered) for all its blocks.

Figure 2–15 shows five channels in a transceiver bank configured in bonded configuration using the receiver standard PCS. The receiver PCS uses both the parallel clock (recovered) and parallel clock from the clock divider. The parallel clock from the clock divider is generated by the central clock divider for the transmitter PCS. It also drives some blocks in the receiver PCS depending on the configuration you use.

Figure 2–15. Five Channels Configured in Bonded Configuration



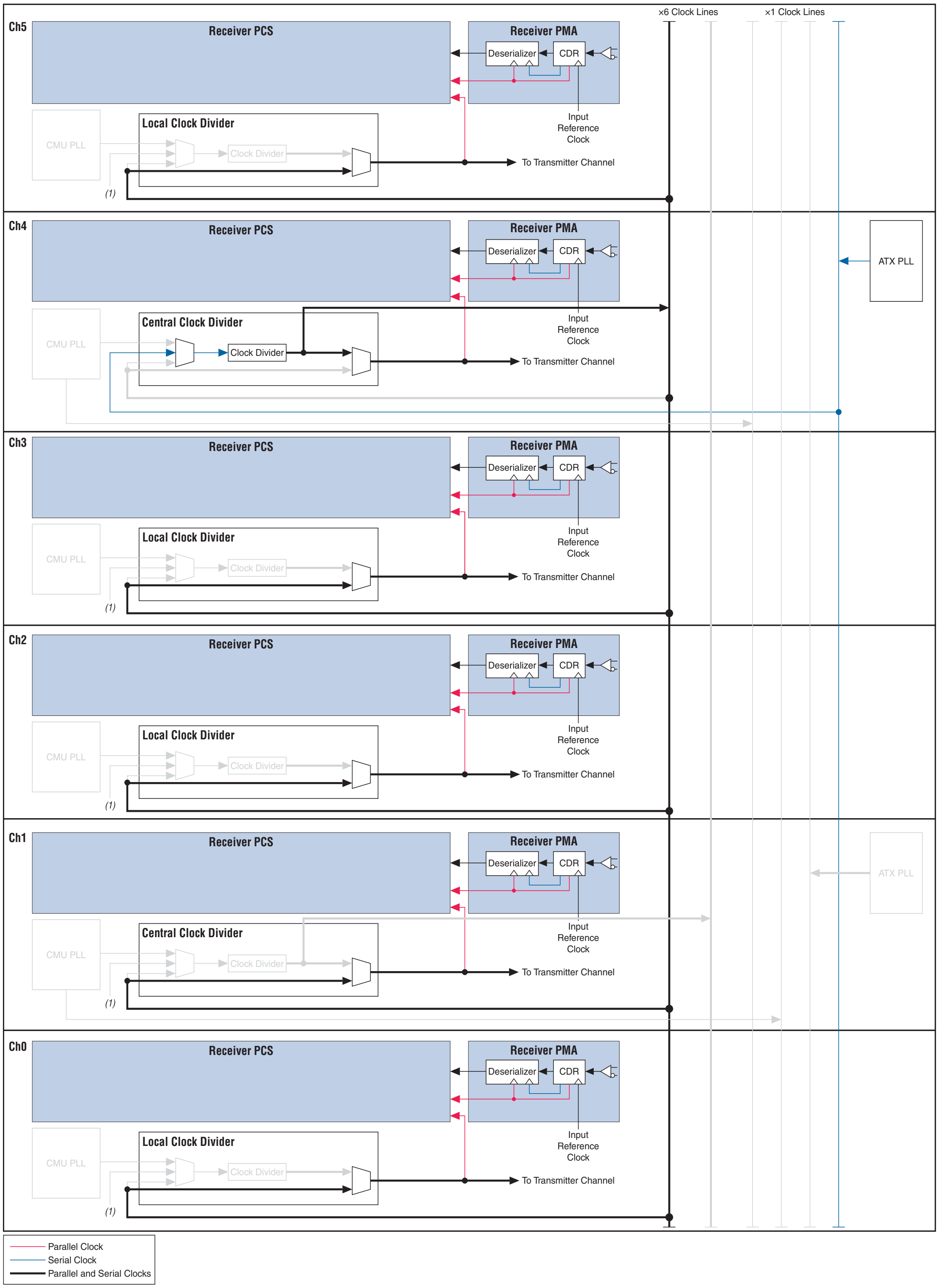
Notes to Figure 2–15:

(1) Serial clock from the ×1 clock lines.

(2) For channel 4, you cannot use the CDR for the receiver because the channel PLL is being used as a CMU PLL.

Figure 2–16 shows all six channels in the transceiver bank in bonded configuration, as opposed to a maximum of five as shown in Figure 2–15. This is possible because the ATX PLL is used as a transmit PLL instead of a channel PLL in the transceiver bank. Using the ATX PLL frees the channel PLLs of both channels 1 and 4 to be configured as CDRs to perform receiver operations.

Figure 2–16. Six Channels Configured in Bonded Configuration Using ATX PLL



Note to Figure 2–16:

(1) Serial clock from the ×1 clock lines.

For more information about the clocking scheme used in different configurations, refer to the *Transceiver Protocol Configurations in Stratix V Devices* and *Custom Transceiver Configuration Datapath in Stratix V Devices* chapters.

FPGA Fabric-Transceiver Interface Clocking

The FPGA fabric-transceiver interface clocks consist of clock signals from the FPGA fabric to the transceiver blocks and clock signals from the transceiver blocks to the FPGA fabric. These clock resources use the clock networks in the FPGA core, including the global (GCLK), regional (RCLK), and periphery (PCLK) clock networks.

The FPGA fabric-transceiver interface clocks can be subdivided into the following three categories:

- Input reference clocks—Refer to "Input Reference Clock Sources" on page 2–2. The input reference clock can be an FPGA fabric-transceiver interface clock when it is also forwarded to the FPGA fabric to clock logic in the FPGA fabric.
- Transceiver datapath interface clocks—Used to transfer data, control, and status signals between the FPGA fabric and the transceiver channels. The transceiver channel forwards the `tx_clkout` signal to the FPGA fabric to clock the data and control signals into the transmitter. The transceiver channel also forwards the recovered `rx_clkout` clock (in configurations without the rate matcher) or the `tx_clkout` clock (in configurations with the rate matcher) to the FPGA fabric to clock the data and status signals from the receiver into the FPGA fabric.
- Other transceiver clocks—The following transceiver clocks form a part of the FPGA fabric-transceiver interface clocks:
  - `mgmt_clk`—Avalon-MM interface clock used for controlling the transceivers, dynamic reconfiguration, and calibration
  - `fixed_clk`—125 MHz fixed-rate clock used in the PCIe (PIPE) receiver detect circuitry

Table 2–4 lists the FPGA fabric-transceiver interface clocks.

Table 2–4. FPGA Fabric-Transceiver Interface Clocks *(Note 1)* (Part 1 of 2)

Clock Name	Clock Description	Interface Direction	FPGA Fabric Clock Resource Utilization
`pll_ref_clk`	Input reference clock used for clocking logic in the FPGA fabric	Transceiver-to-FPGA fabric	GCLK, RCLK, PCLK
`tx_clkout`	Clock forwarded by the transceiver for clocking the transceiver datapath interface	Transceiver-to-FPGA fabric	GCLK, RCLK, PCLK
`rx_clkout`	Clock forwarded by the receiver for clocking the receiver datapath interface	Transceiver-to-FPGA fabric	GCLK, RCLK, PCLK
`tx_coreclkin`	User-selected clock for clocking the transmitter datapath interface	FPGA fabric-to-transceiver	GCLK, RCLK, PCLK
`rx_coreclkin`	User-selected clock for clocking the receiver datapath interface	FPGA fabric-to-transceiver	GCLK, RCLK, PCLK
`fixed_clk`	PCIe receiver detect clock	FPGA fabric-to-transceiver	GCLK, RCLK, PCLK

Table 2–4. FPGA Fabric-Transceiver Interface Clocks *(Note 1)* **(Part 2 of 2)**

Clock Name	Clock Description	Interface Direction	FPGA Fabric Clock Resource Utilization
mgmt_clk *(2)*	Avalon-MM interface management clock	FPGA fabric-to-transceiver	GCLK, RCLK, PCLK

Notes to Table 2–4:

(1) For more information about the GCLK, RCLK, and PCLK resources available in each device, refer to the *Clock Networks and PLLs in Stratix V Devices* chapter.

(2) mgmt_clk is a free-running clock that is not derived from the transceiver blocks.

Table 2–5 lists the port names for tx_clkout and rx_clkout.

Table 2–5. Configuration Specific Port Names for tx_clkout and rx_clkout

Configuration	Port Name for tx_clkout	Port Name for rx_clkout
Custom	tx_clkout	rx_clkout
Interlaken	tx_clkout	rx_clkout
Low Latency	tx_clkout	rx_clkout
PCIe	pipe_pclk	pipe_pclk
XAUI	xgmii_tx_clk	xgmii_rx_clk

Transmitter Datapath Interface Clocking

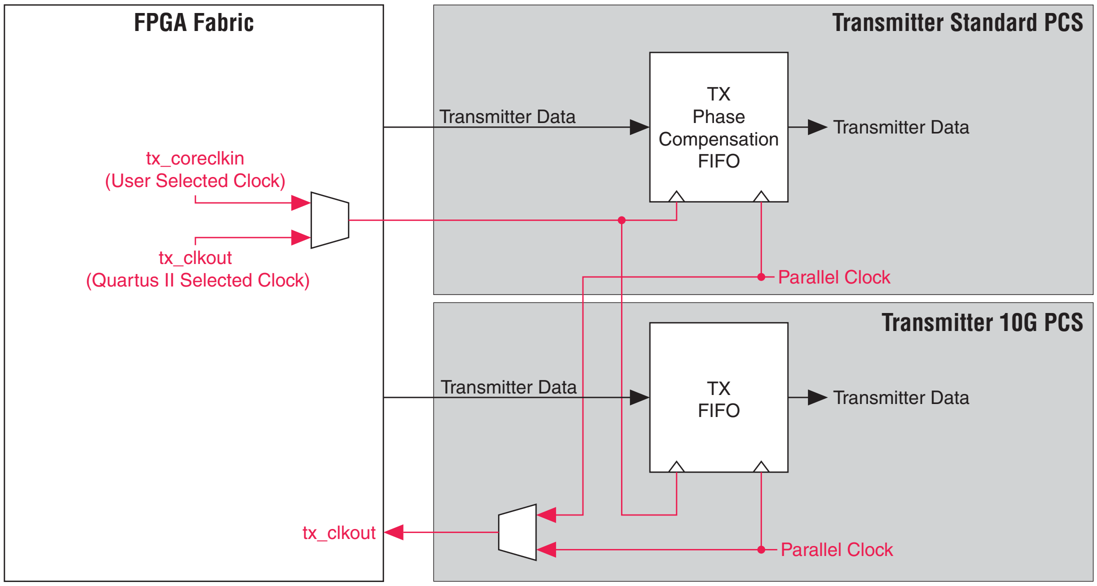
The transmitter datapath interface is comprised of the following:

- Write side of the TX phase compensation FIFO—for configurations that use the standard PCS channel
- Write side of the TX FIFO—for configurations that use the 10G PCS channel

This interface is clocked by the transmitter datapath interface clock. Figure 2–17 shows transmitter datapath interface clocking. The transmitter PCS forwards the following clocks to the FPGA fabric:

- tx_clkout for each transmitter channel in non-bonded configuration
- tx_clkout[0] for all transmitter channels in bonded configuration

Figure 2–17. Transmitter Datapath Interface Clocking



All configurations using the standard PCS channel must have a 0 parts per million (PPM) difference between the transmitter datapath interface clock and the read side clock of the TX phase compensation FIFO.

For more information about interface clocking for each configuration, refer to the *Transceiver Custom Configurations in Stratix V Devices* chapter and the clocking sections for each configuration in the *Transceiver Protocol Configurations in Stratix V Devices* chapter.

You can clock the transmitter datapath interface by using one of the following:

- Quartus II-selected transmitter datapath interface clock
- User-selected transmitter datapath interface clock

User-selection is provided to share the transceiver datapath interface clocks to reduce GCLK, RCLK, and PCLK resource utilization in your design.

Quartus II-Selected Transmitter Datapath Interface Clock

The Quartus II software automatically picks the appropriate clock from the FPGA fabric to clock the transmitter datapath interface. Figure 2–18 shows the transmitter datapath interface of two non-bonded channels clocked by their respective transmitter PCS clocks that are forwarded to the FPGA fabric.

Figure 2–18. Transmitter Datapath Interface Clocking for Non-Bonded Channels

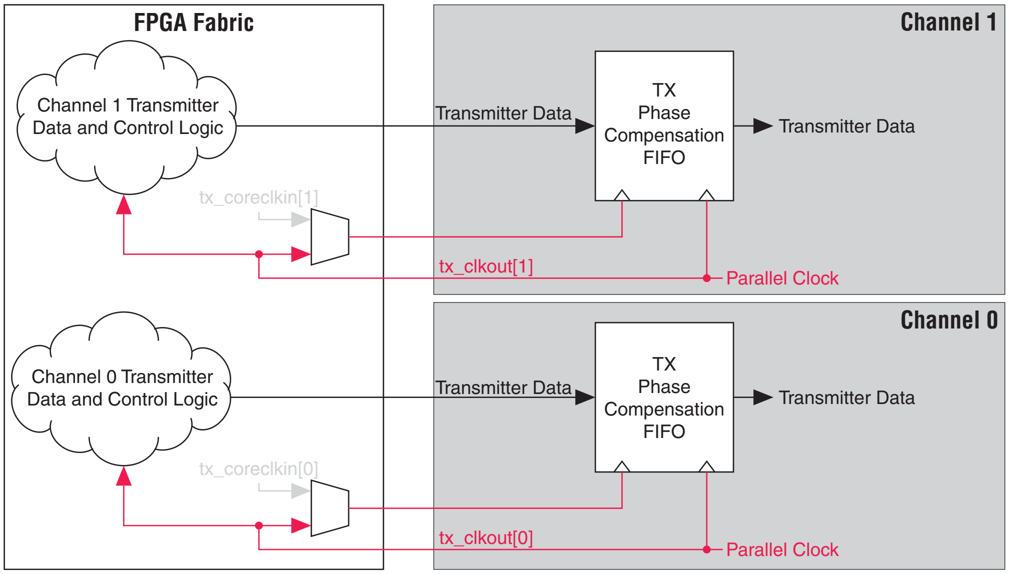
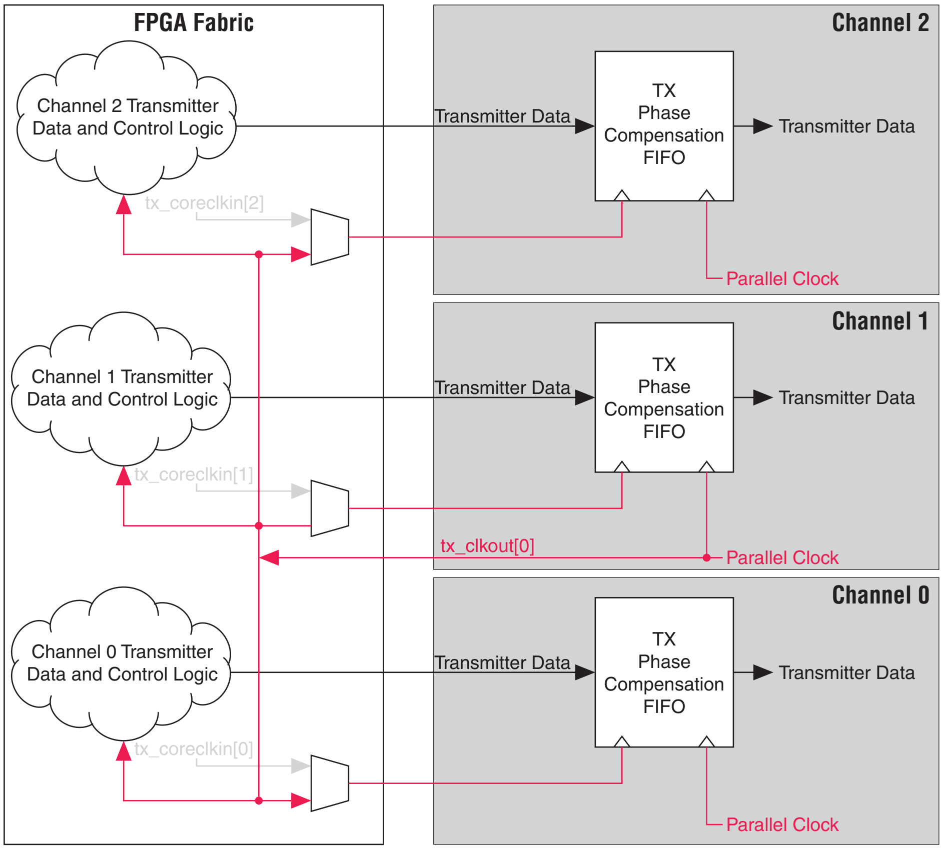


Figure 2–19 shows the transmitter datapath interface of three bonded channels clocked by the `tx_clkout[0]` clock. The `tx_clkout[0]` clock is derived from the central clock divider of channel 1 or 4 in a transceiver bank.

Figure 2–19. Transmitter Datapath Interface Clocking for Three Bonded Channels



User-Selected Transmitter Datapath Interface Clock

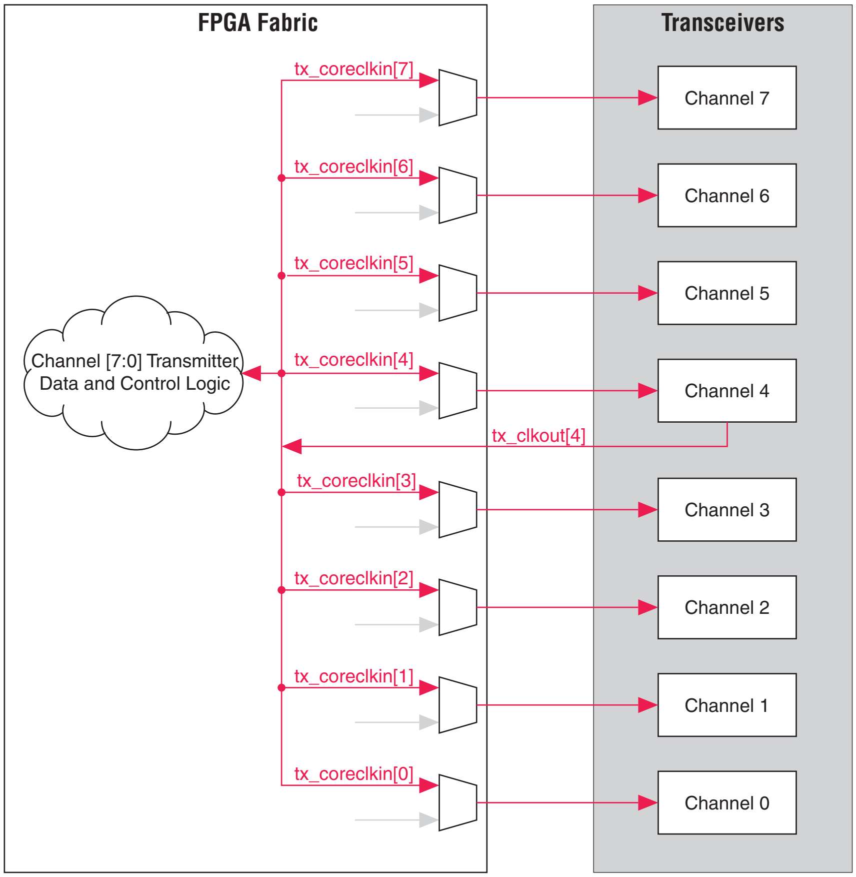
Multiple transmitter channels that are non-bonded lead to high utilization of GCLK, RCLK, and PCLK resources (one clock resource per channel as shown in Figure 2–18). You can significantly reduce GCLK, RCLK, and PCLK resource use for transmitter datapath clocks if the transmitter channels are identical.

Identical transmitter channels are defined as channels that have the same input reference clock source, the same transmit PLL configuration, and the same transmitter PMA and PCS configuration. Identical transmitter channels may have different analog settings, such as transmitter voltage output differential (V_{OD}), transmitter common mode voltage (V_{CM}), or pre-emphasis setting.

To achieve the clock resource savings, select a common clock driver for the transmitter datapath interface of all identical transmitter channels. Figure 2–20 shows eight identical channels clocked by a single clock (`tx_clkout` of channel 4). To clock eight identical channels with a single clock, instantiate the `tx_coreclkin` port for all the identical transmitter channels (`tx_coreclkin[7:0]`). Connect `tx_clkout[4]` to the `tx_coreclkin[7:0]` ports. Also, connect `tx_clkout[4]` to the transmitter data and control logic for all eight channels.

☞ Resetting or powering down channel 4 will lead to a loss of the clock for all eight channels.

Figure 2–20. Eight Identical Channels with a Single User-Selected Transmitter Interface Clock



The common clock must have a 0 PPM difference with respect to the read side of the TX FIFO (in the 10G PCS channel) or TX phase compensation FIFO (in the standard PCS channel) of all the identical channels. A frequency difference causes the FIFO to under-run or overflow, depending on whether the common clock is slower or faster, respectively. You can drive the 0 PPM common clock by one of the following sources:

- `tx_clkout` of any channel in non-bonded channel configurations
- `tx_clkout[0]` in bonded channel configurations
- Dedicated `refclk` pins

☞ The Quartus II software does not allow gated clocks or clocks generated in the FPGA logic to drive the `tx_coreclkin` ports.

Because the Quartus II software allows you to use external pins, such as dedicated `refclk` pins, it has no way of ensuring a 0 PPM difference. You must ensure a 0 PPM difference.

Receiver Datapath Interface Clock

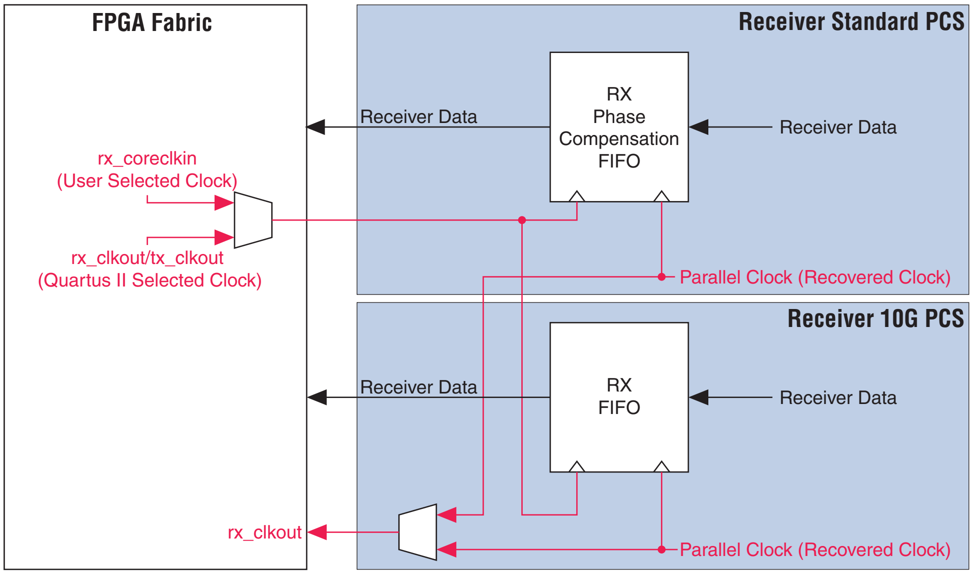
The receiver datapath interface is comprised of the following:

- Read side of the RX phase compensation FIFO—for configurations that use the standard PCS channel
- Read side of the RX FIFO—for configurations that use the 10G PCS channel

This interface is clocked by the receiver datapath interface clock. Figure 2–21 shows receiver datapath interface clocking. The receiver PCS forwards the following clocks to the FPGA fabric:

- `rx_clkout`—for each receiver channel in non-bonded configuration when a rate matcher is not used
- `tx_clkout`—for each receiver channel in non-bonded configuration when a rate matcher is used
- single `tx_clkout[0]`—for all receiver channels in bonded configuration

Figure 2–21. Receiver Datapath Interface Clocking


FPGA Fabric
Receiver Standard PCS
RX Phase Compensation FIFO
Receiver Data
Receiver Data
rx_coreclkin (User Selected Clock)
rx_clkout/tx_clkout (Quartus II Selected Clock)
Parallel Clock (Recovered Clock)
Receiver 10G PCS
RX FIFO
Receiver Data
Receiver Data
rx_clkout
Parallel Clock (Recovered Clock)

All configurations that use the standard PCS channel must have a 0 PPM difference between the receiver datapath interface clock and the read side clock of the RX phase compensation FIFO.

For more information about interface clocking for each configuration, refer to the clocking sections for each configuration in the *Transceiver Protocol Configurations in Stratix V Devices* chapter.

You can clock the receiver datapath interface by using one of the following:

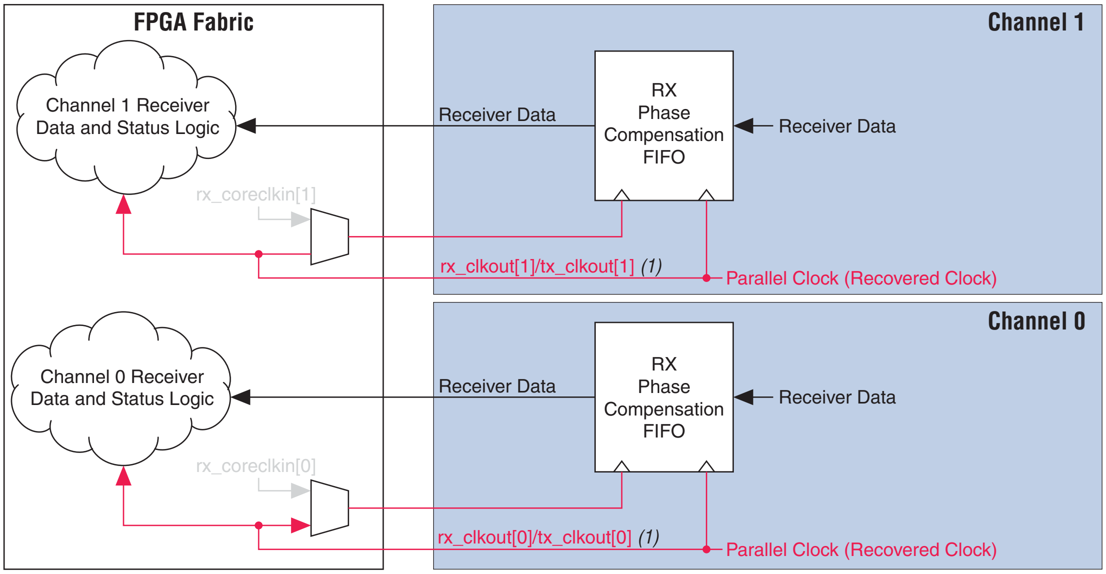
- Quartus II-selected receiver datapath interface clock
- User-selected receiver datapath interface clock

User-selection is provided to share the transceiver datapath interface clocks to reduce GCLK, RCLK, and PCLK resource utilization in your design.

Quartus II Software-Selected Receiver Datapath Interface Clock

The Quartus II software automatically picks the appropriate clock from the FPGA fabric to clock the receiver datapath interface. Figure 2–22 shows the receiver datapath interface of two non-bonded channels clocked by their respective receiver PCS clocks forwarded to the FPGA fabric.

Figure 2–22. Receiver Datapath Interface Clocking for Non-Bonded Channels

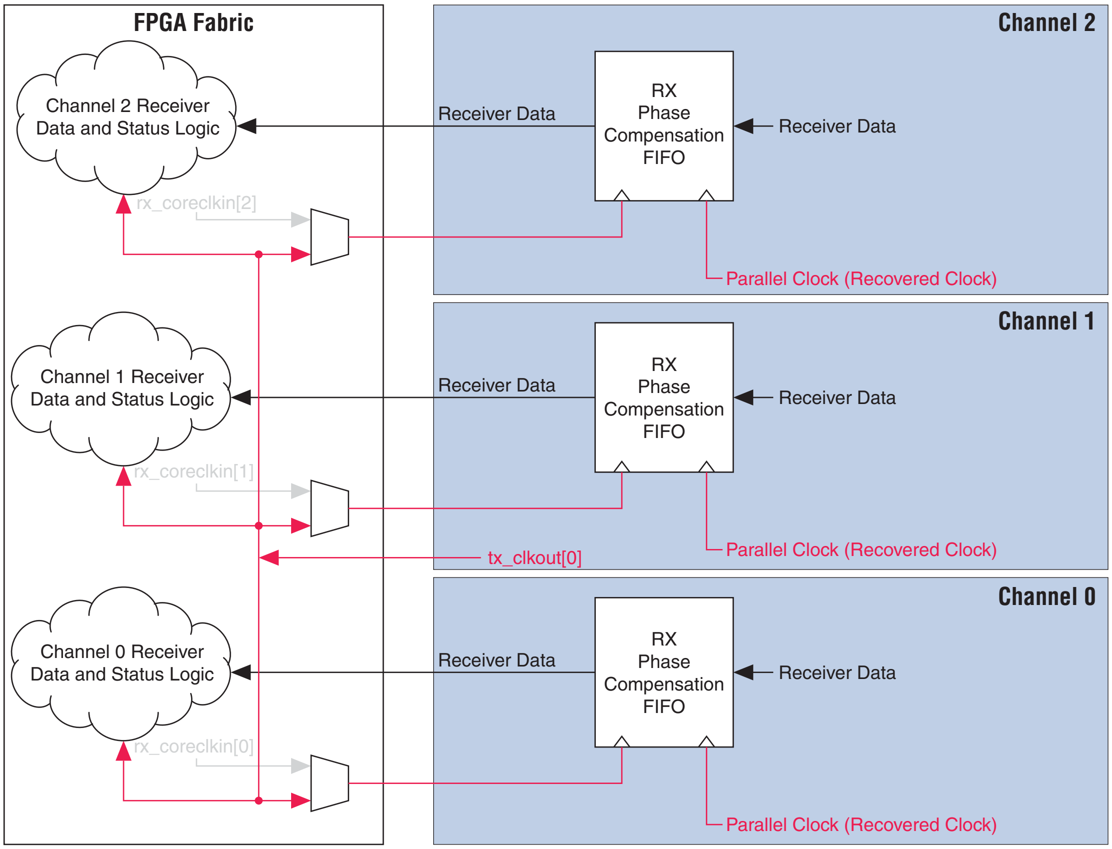

FPGA Fabric
Channel 1
Channel 1 Receiver Data and Status Logic
Receiver Data
RX Phase Compensation FIFO
Receiver Data
rx_coreclkin[1]
rx_clkout[1]/tx_clkout[1] (1)
Parallel Clock (Recovered Clock)
Channel 0
Channel 0 Receiver Data and Status Logic
Receiver Data
RX Phase Compensation FIFO
Receiver Data
rx_coreclkin[0]
rx_clkout[0]/tx_clkout[0] (1)
Parallel Clock (Recovered Clock)

Note to Figure 2–22:

(1) If you use a rate matcher, the `tx_clkout` clock is used.

Figure 2–23 shows the receiver datapath interface of three bonded channels clocked by the `tx_clkout[0]` clock. The `tx_clkout[0]` clock is derived from the central clock divider of channel 1 or 4 in a transceiver bank.

Figure 2–23. Receiver Datapath Interface Clocking for Three Bonded Channels


FPGA Fabric
Channel 2 Receiver Data and Status Logic
rx_coreclkin[2]
Channel 1 Receiver Data and Status Logic
rx_coreclkin[1]
Channel 0 Receiver Data and Status Logic
rx_coreclkin[0]
Channel 2
Receiver Data
RX Phase Compensation FIFO
Receiver Data
Parallel Clock (Recovered Clock)
Channel 1
Receiver Data
RX Phase Compensation FIFO
Receiver Data
Parallel Clock (Recovered Clock)
tx_clkout[0]
Channel 0
Receiver Data
RX Phase Compensation FIFO
Receiver Data
Parallel Clock (Recovered Clock)

User-Selected Receiver Datapath Interface Clock

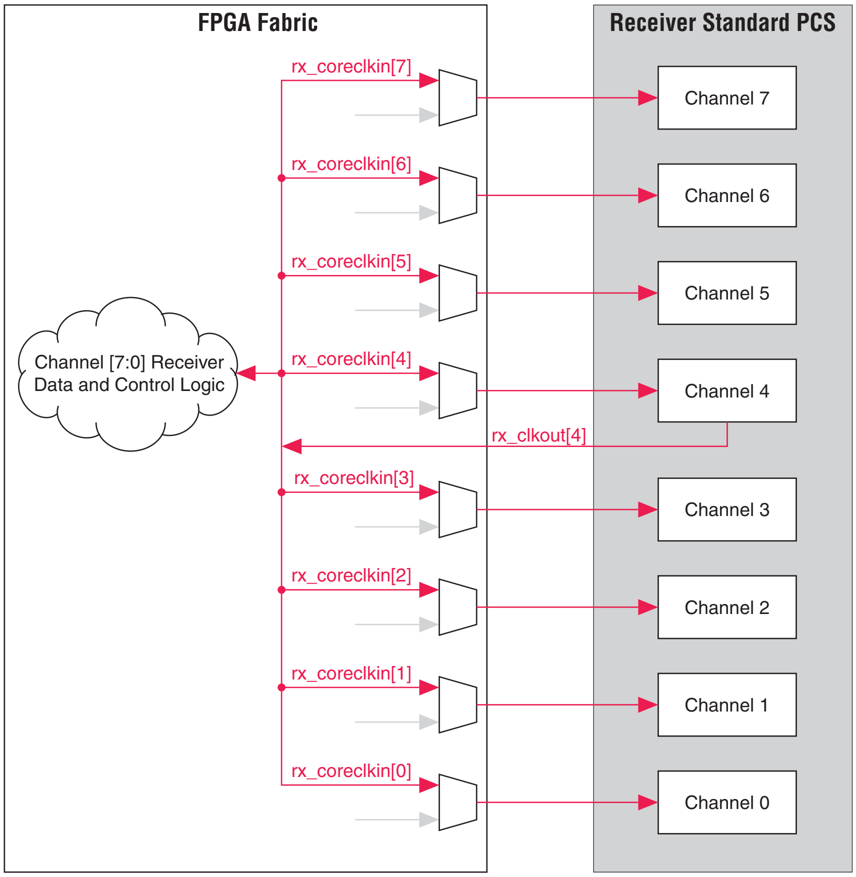
Multiple receiver channels that are non-bonded lead to high utilization of GCLK, RCLK, and PCLK resources (one clock resource per channel as shown in Figure 2–22 on page 2–29). You can significantly reduce GCLK, RCLK, and PCLK resource use for receiver datapath clocks if the receiver channels are identical.

☞ Identical receiver channels are defined as channels that have the same input reference clock source for the CDR and the same receiver PMA and PCS configuration. Identical receiver channels may have different analog settings, such as receiver common mode voltage (V_{ICM}), equalization, or DC gain setting.

To achieve clock resource savings, select a common clock driver for the receiver datapath interface of all identical receiver channels. This is done by instantiating the `rx_coreclkin` port for all the identical receiver channels and connecting the common clock driver to their receiver datapath interface and receiver data and control logic. Figure 2–24 shows eight identical channels that are clocked by a single clock (`rx_clkout` of channel 4). To clock eight identical channels with a single clock, instantiate the `rx_coreclkin` port for all the identical receiver channels (`rx_coreclkin[7:0]`). Connect `rx_clkout[4]` to the `rx_coreclkin[7:0]` ports. Also, connect `rx_clkout[4]` to the receiver data and control logic for all eight channels.

☞ Resetting or powering down channel 4 will lead to a loss of the clock for all eight channels.

Figure 2–24. Eight Identical Channels with a Single User-Selected Receiver Interface Clock



The common clock must have a 0 PPM difference with respect to the write side of the RX FIFO (in the 10G PCS channel) or RX phase compensation FIFO (in the standard PCS channel) of all the identical channels. A frequency difference causes the FIFO to under-run or overflow, depending on whether the common clock is faster or slower, respectively. You can drive the 0 PPM common clock driver by one of the following sources:

- `tx_clkout` of any channel in non-bonded receiver channel configurations with the rate matcher
- `rx_clkout` of any channel in non-bonded receiver channel configurations without the rate matcher
- `tx_clkout[0]` in bonded receiver channel configurations
- Dedicated `refclk` pins

☞ The Quartus II software does not allow gated clocks or clocks generated in the FPGA logic to drive the `rx_coreclkin` ports.

Because the Quartus II software allows you to use external pins, such as dedicated `refclk` pins, it has no way of ensuring a 0 PPM difference. You must ensure a 0 PPM difference.

Document Revision History

Table 2–6 lists the revision history for this chapter.

Table 2–6. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Added information about fractional PLLs as they provide an input reference clock in "Input Reference Clocking". ■ Chapter moved to Volume 3.
December 2010	1.1	■ Updated clock names ■ Updated figures for more accurate depiction of transceiver clocking ■ Added information about ATX PLLs
July 2010	1.0	Initial release

3. Transceiver Reset Control in Stratix V Devices

SV52004-2.0

This chapter provides the recommended transceiver initialization and reset sequence for Stratix® V devices. The recommended reset sequence ensures that the physical coding sublayer (PCS) and physical medium attachment (PMA) in each transceiver channel is initialized correctly. This is critical for reliable transceiver operations after the initial power-up and for re-establishing the transceiver link.

Transceiver Reset Signals

Table 3–1 lists the transceiver reset signals and the transceiver circuitry impacted by each signal.

Table 3–1. Transceiver Reset Signals ***(Note 1)***

Internal Signal Name	Impacted Transceiver Circuitry
pll_powerdown	Resets the transmitter PLL when asserted high
tx_digitalreset	Resets all blocks in the transmitter PCS when asserted high
rx_analogreset	Resets the receiver CDR when asserted high
rx_digitalreset	Resets all blocks in the receiver PCS when asserted high

Note to Table 3–1:

(1) These are internal signal names that could be different from the PHY IP or Custom PHY register names.

Transceiver Reset Controller Implementation

The transceiver reset sequence controller is automatically implemented as part of the PHY IP core in each transceiver configuration. This approach simplifies transceiver-based design development because the embedded reset sequence controller ensures reliable transceiver link initialization.

☞ There is only one reset controller for all the channels in a PHY IP instance.

Figure 3–1 shows a block diagram of the PHY IP core instance with the embedded reset sequence controller.

Figure 3–1. PHY IP Instance with Embedded Reset Sequence Controller

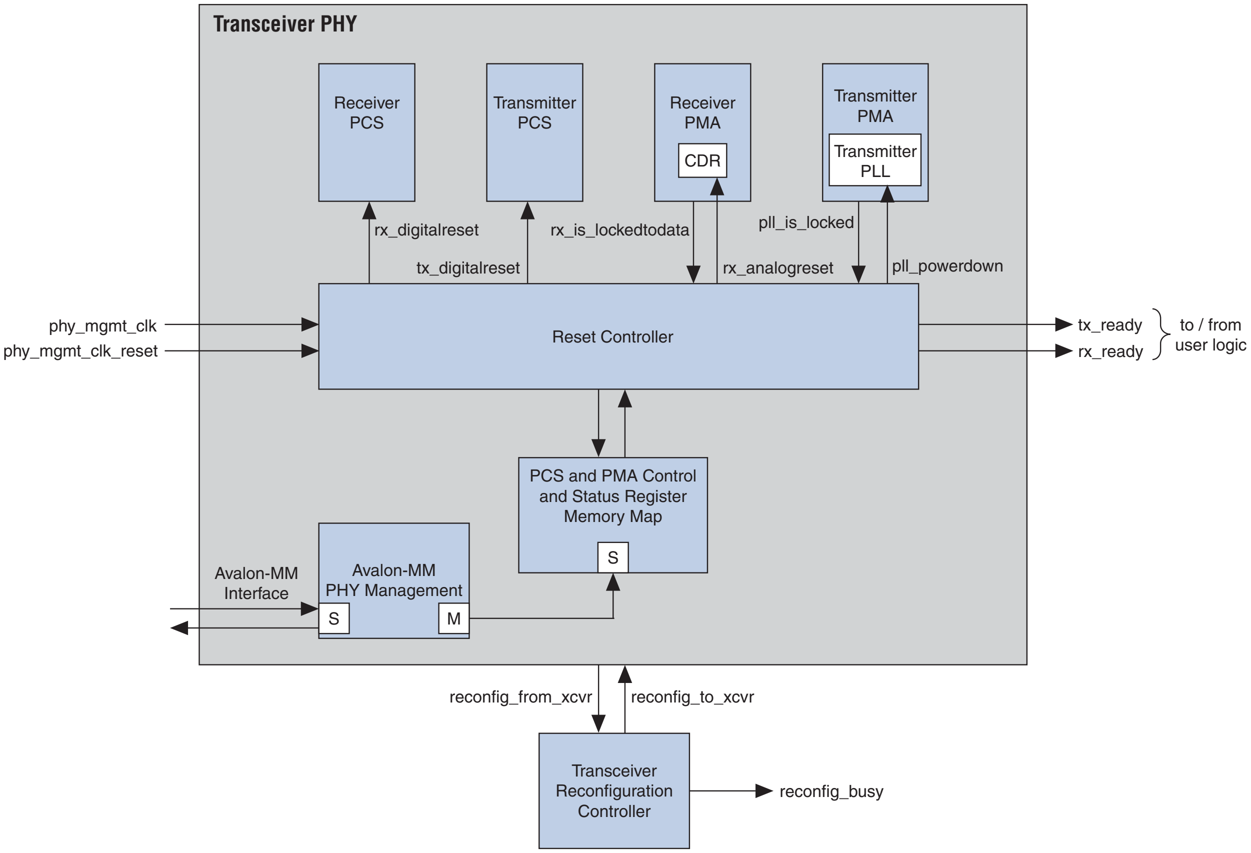


Table 3–2 lists the reset control and status signals provided to the user logic.

Table 3–2. Reset Control and Status Signals

Signal Name	Signal Type	Description
phy_mgmt_clk	Control Input	Clock for the reset controller
phy_mgmt_clk_reset	Control Input	Low-to-high transition initiates the transceiver reset sequence
tx_ready	Status Output	Low indicates that the TX datapath is in reset. High indicates that the TX datapath is out of reset and ready for data transmission.
rx_ready	Status Output	Low indicates that the RX datapath is in reset. High indicates that the RX datapath is out of reset and ready for data reception.

Transceiver Reset Sequence

The recommended transceiver reset sequence is divided into two categories:

- Reset sequence for CDR in automatic lock mode
- Reset sequence for CDR in manual lock mode

Reset Sequence for CDR in Automatic Lock Mode

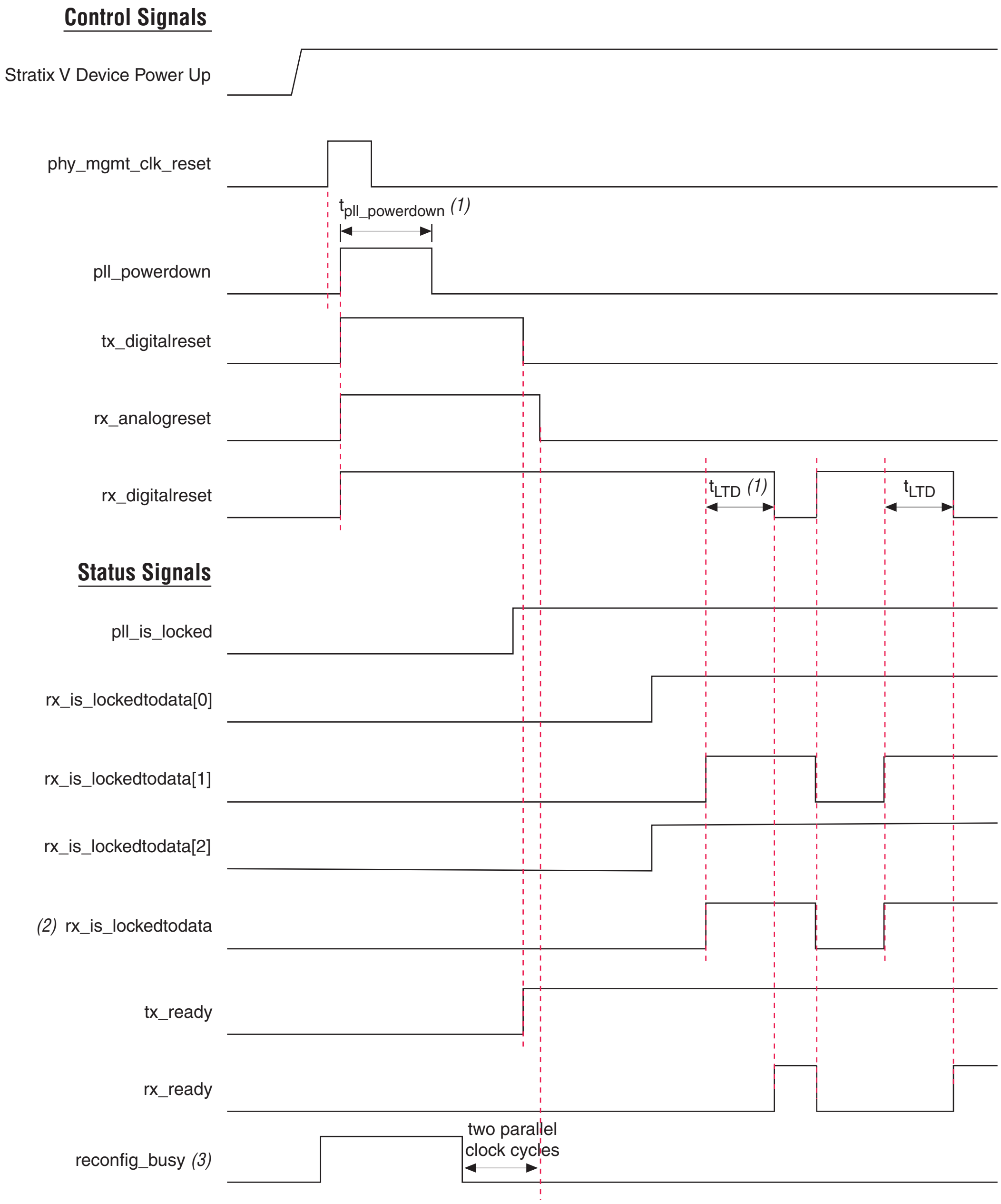
Figure 3–2 shows the timing diagram of the transceiver reset sequence for CDR in automatic lock mode. The reset sequence is implemented automatically by the embedded reset controller. After device power-up, the reset controller initiates the reset sequence when it receives a positive edge on the `phy_mgmt_clk_reset` input signal. The reset controller then indicates that the transmitter and receiver channels are ready for data transmission and reception by asserting the `tx_ready` and `rx_ready` signals, respectively.

After the initial reset sequence, the reset controller continuously monitors all the status signals and asserts the appropriate reset signals is case of loss of link or loss of reference clock.

☞ The reset sequence for PIPE follows the same reset sequence as the CDR in automatic lock mode.

If you are implementing your own reset controller, follow the same sequence as shown in the timing diagram (Figure 3–2).

Figure 3–2. Transceiver Reset Sequence Timing Diagram for CDR in Automatic Lock Mode



Notes to Figure 3–2:

(1) $t_{pll\_powerdown}$ and t_{LTD} are pending characterization.

(2) The `rx_is_lockedtodata` signal shown in this figure is the logical AND of the `rx_is_lockedtodata` signals from all channels in a PHY IP instance. If one RX channel loses lock, all channels in the PHY IP instance are reset.

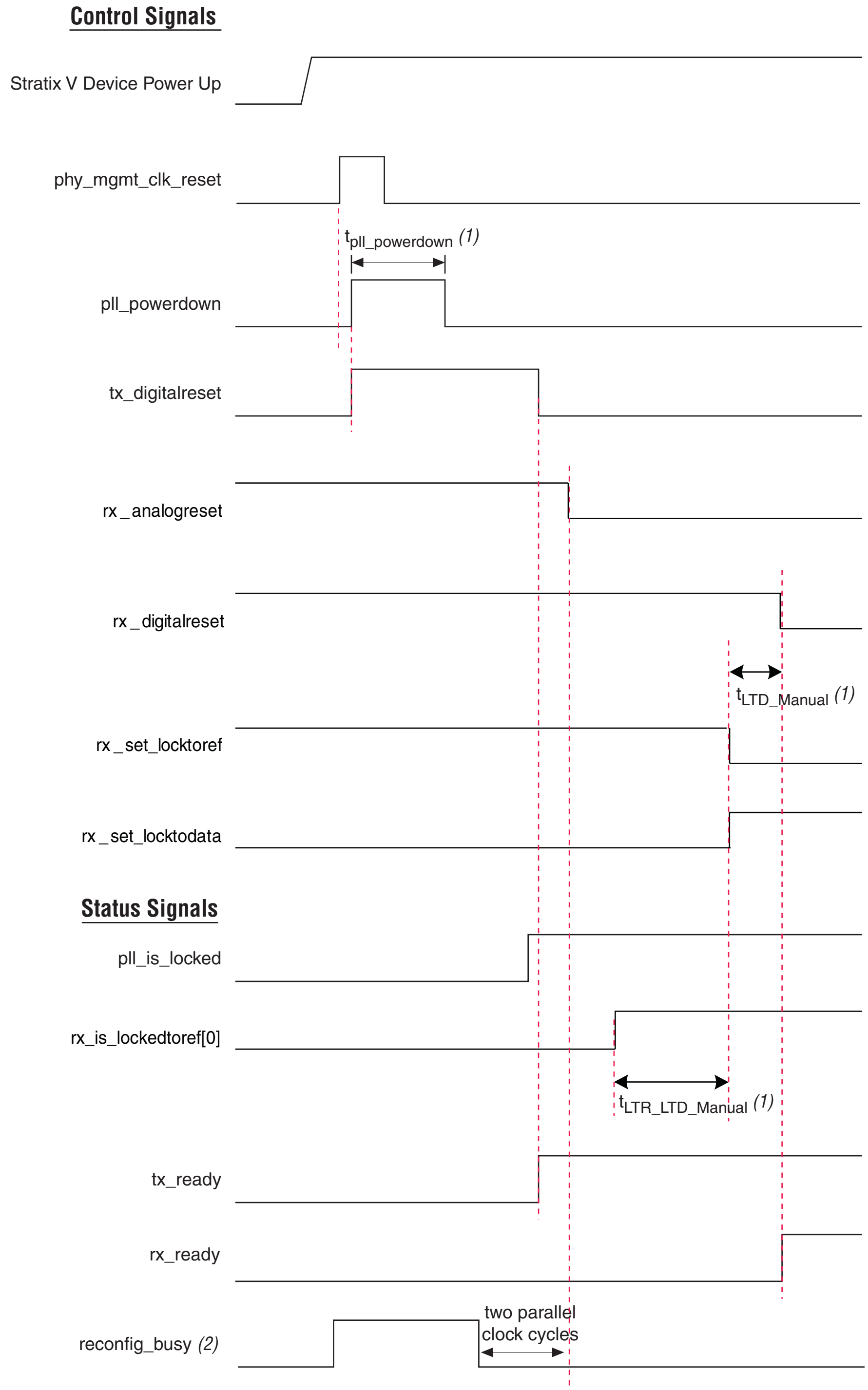
(3) `reconfig_busy` is driven from the transceiver reconfiguration controller.

Reset Sequence for CDR in Manual Lock Mode

In manual lock mode, you are responsible for monitoring all the status signals and asserting the appropriate reset signals. In case of loss of signal or loss of the reference clock, you must assert the necessary reset signals.

Figure 3–3 shows the timing diagram of the transceiver reset sequence for CDR in manual lock mode. You must manually perform the reset sequence for the receiver datapath by asserting `rx_set_locktoref` high, waiting for the status `rx_is_lockedtoref` to go high and then asserting `rx_set_locktodata` high.

Figure 3–3. Transceiver Reset Sequence Timing Diagram for CDR in Manual Lock Mode



Notes to Figure 3–3:

(1) $t_{pll\_powerdown}$, t_{LTD_Manual}, and $t_{LTR\_LTD\_Manual}$ are pending characterization.

(2) `reconfig_busy` is driven from the transceiver reconfiguration controller.

Document Revision History

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

Table 3–3. Document Revision History

Date	Version	Changes
May 2011	2.0	■ Added sections for CDR reset sequence in automatic and manual lock mode. ■ Removed sections for PCIe and non-PCIe reset sequences. ■ Chapter moved to Volume 3.
December 2010	1.1	No changes to the content of this chapter for the Quartus® II software 10.1.
July 2010	1.0	Initial release.

4. Transceiver Protocol Configurations in Stratix V Devices

SV52005-1.2

This chapter provides the transceiver channel datapath, clocking guidelines, channel placement guidelines, and a brief description of protocol features supported in each transceiver configuration for Stratix® V devices.

Stratix V devices have dedicated transceiver physical coding sublayer (PCS) and physical medium attachment (PMA) circuitry to support the following communication protocols:

- "10GBASE-R"
- "Interlaken" on page 4–8
- "PCI Express (PCIe)—Gen1 and Gen2" on page 4–15
- "GIGE" on page 4–30
- "XAUI" on page 4–37

For a complete list of serial protocols supported by Stratix V devices, refer to the *Upcoming Stratix V Device Features* document.

Use this chapter along with the *Altera Transceiver PHY IP Core User Guide* to implement your intended protocol links in Stratix V devices.

Table 4–1 lists the Quartus® II PHY IP Core instance names that you must instantiate for each supported transceiver configuration.

Table 4–1. Quartus II PHY IP Core Names

Transceiver Configuration	Quartus II PHY IP Core
10GBASE-R	10GBASE-R PHY
Interlaken	Interlaken PCS
PCI Express® (PCIe®)	PCI Express PHY (PIPE)
XAUI	XAUI PHY

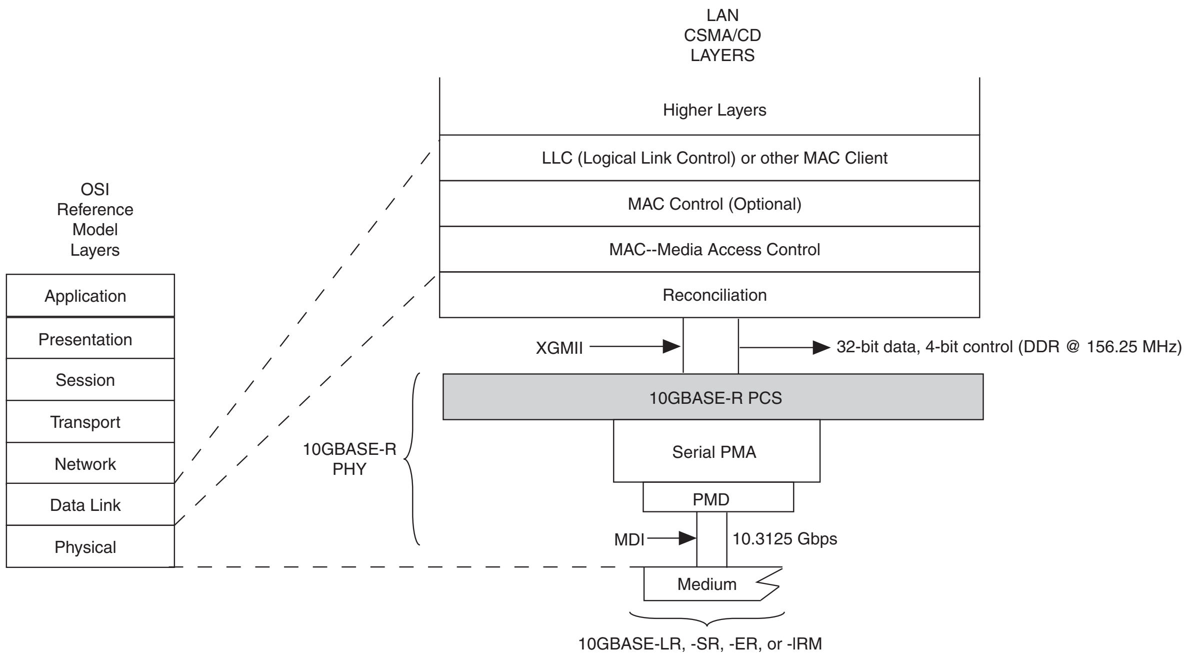
10GBASE-R

This section describes 10GBASE-R link implementation using Stratix V transceivers. It provides the transceiver channel datapath, clocking, and channel placement guidelines when configured in a 10GBASE-R configuration.

10GBASE-R is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in clause 49 of the IEEE 802.3-2008 specification. As shown in Figure 4–1, the 10GBASE-R PHY uses the XGMII interface to connect to the IEEE802.3 media access control (MAC) and reconciliation sublayer (RS). The IEEE 802.3-2008 specification requires each 10GBASE-R link to support a 10 Gbps data rate at the XGMII interface and a 10.3125 Gbps serial line rate with 64B/66B encoding.

Figure 4–1 shows the relationship between the 10GBASE-R PHY and other sublayers in the OSI reference model.

Figure 4–1. 10GBASE-R PHY Connection to IEEE802.3 MAC and RS



Transceiver Datapath Configuration

Figure 4–2 shows the transceiver blocks and settings enabled in a 10GBASE-R configuration. The blocks shown as “Disabled” are not used, but incur latency. The blocks shown as “Bypassed” are not used and do not incur any latency.

Figure 4–2. Transceiver Blocks Enabled in 10GBASE-R

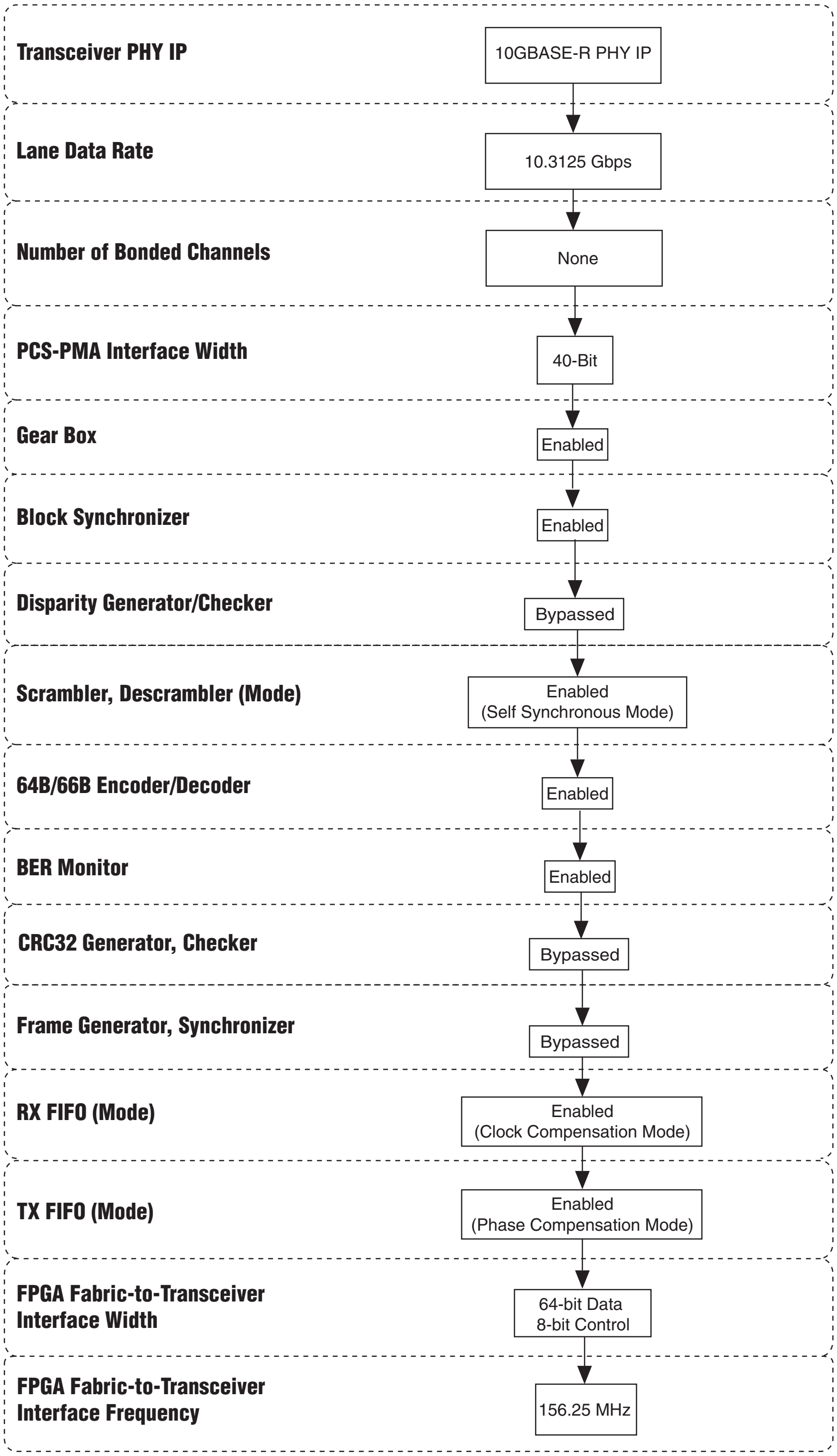
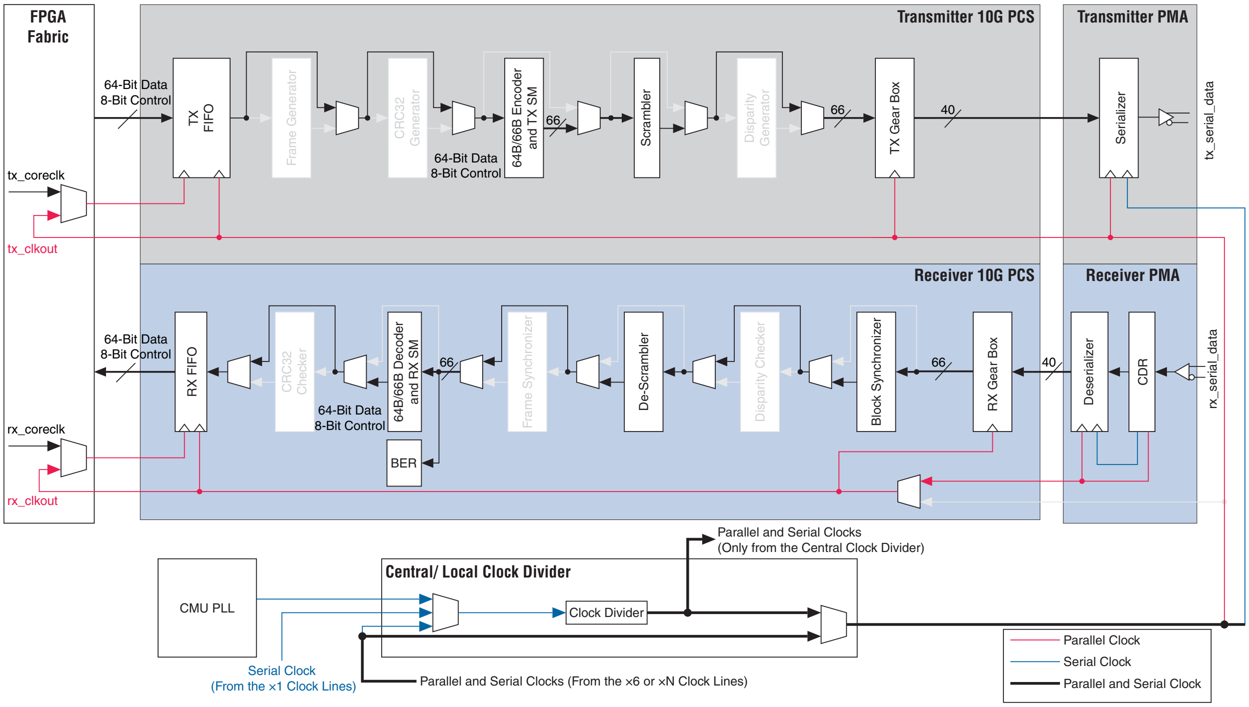

Transceiver PHY IP
10GBASE-R PHY IP
Lane Data Rate
10.3125 Gbps
Number of Bonded Channels
None
PCS-PMA Interface Width
40-Bit
Gear Box
Enabled
Block Synchronizer
Enabled
Disparity Generator/Checker
Bypassed
Scrambler, Descrambler (Mode)
Enabled (Self Synchronous Mode)
64B/66B Encoder/Decoder
Enabled
BER Monitor
Enabled
CRC32 Generator, Checker
Bypassed
Frame Generator, Synchronizer
Bypassed
RX FIFO (Mode)
Enabled (Clock Compensation Mode)
TX FIFO (Mode)
Enabled (Phase Compensation Mode)
FPGA Fabric-to-Transceiver Interface Width
64-bit Data 8-bit Control
FPGA Fabric-to-Transceiver Interface Frequency
156.25 MHz

Figure 4–3 shows the transceiver datapath in a 10GBASE-R configurations.

Figure 4–3. Channel Datapath in a 10GBASE-R Configuration



Supported Features

The following sections describe the features supported by Stratix V transceivers in a 10GBASE-R configuration.

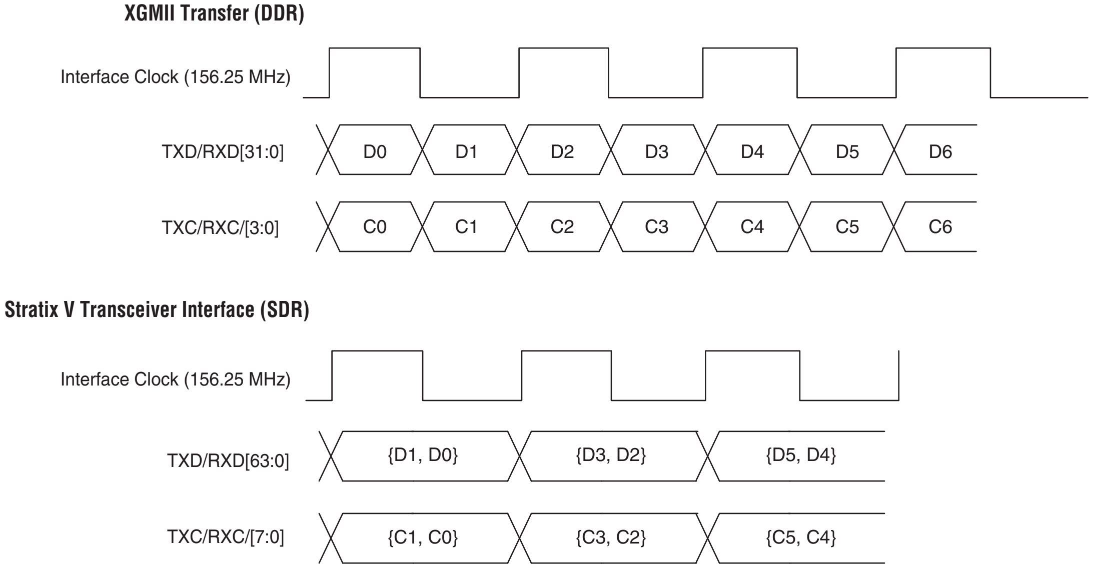
For details about 10GBASE-R PHY IP control and status signals associated with each feature, refer to the *10GBASE-R PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

64-Bit Single Data Rate (SDR) Interface to the MAC/RS

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the 10GBASE-R PCS and the Ethernet MAC/RS. The XGMII interface defines 32-bit data and 4-bit wide control character clocked between the MAC/RS and the PCS at both the positive and negative edge (DDR) of the 156.25 MHz interface clock.

Stratix V transceivers do not support the XGMII interface to the MAC/RS as defined in the IEEE 802.3-2008 specification. Instead, they support a 64-bit data and 8-bit control SDR interface between the MAC/RS and the PCS, as shown in Figure 4–4.

Figure 4–4. XGMII interface (DDR) versus Stratix V Transceiver Interface (SDR) for 10GBASE-R



64B/66B Encoding/Decoding

Stratix V transceivers in a 10GBASE-R configuration support 64B/66B encoding and decoding as specified in Clause 49 of the IEEE802.3-2008 specification. The 64B/66B encoder receives 64-bit data and 8-bit control code from the transmitter FIFO and converts it into 66-bit encoded data. The 66-bit encoded data contains two overhead sync header bits that are used by the receiver PCS for block synchronization and bit-error rate (BER) monitoring.

The 64B/66B encoding also ensures enough transitions on the serial data stream for the receiver clock data recovery (CDR) to maintain its lock to the incoming data.

Transmitter and Receiver State Machines

Stratix V transceivers in a 10GBASE-R configuration implement the transmit and receive state diagrams shown in Figure 49-14 and Figure 49-15 of the IEEE802.3-2008 specification.

Besides encoding the raw data as per the rules of the 10GBASE-R PCS, the transmit state diagram performs functions such as transmitting local faults (LBLOCK_T) under reset as well as transmitting error codes (EBLOCK_T) when the 10GBASE-R PCS rules are violated.

Besides decoding the incoming data as per the rules of the 10GBASE-R PCS, the receive state diagram performs functions such as sending local faults (LBLOCK_R) to the MAC/RS under reset and substituting error codes (EBLOCK_R) when the 10GBASE-R PCS rules are violated.

Block Synchronization

The block synchronizer in the receiver PCS determines when the receiver has obtained lock to the received data stream. It implements the lock state diagram shown in Figure 49-12 of the IEEE 802.3-2008 specification.

The block synchronizer provides a status signal to indicate whether it has achieved block synchronization or not.

Self-Synchronous Scrambling/Descrambling

The scrambler/descrambler blocks in the transmitter/receiver PCS implements the self-synchronizing scrambler/descrambler polynomial 1 + x39 + x58 as described in clause 49 of the IEEE 802.3-2008 specification. The scrambler/descrambler blocks are self-synchronizing and do not require an initialization seed. Barring the two sync header bits in each 66-bit data block, the entire payload is scrambled or descrambled.

BER Monitor

The BER monitor block in the receiver PCS implements the BER monitor state diagram shown in Figure 49-13 of the IEEE 802.3-2009 specification. The BER monitor provides a status signal to the MAC whenever the link BER threshold is violated.

The 10GBASE-R PHY IP core provides a status flag to indicate a high BER whenever 16 synchronization header errors are received within a 125 μs window.

Clock Compensation

The receiver FIFO in the receiver PCS datapath is designed to compensate up to ±100 PPM difference between the remote transmitter and the local receiver. It does so by inserting Idles (/I/), and deleting Idles (/I/) or Ordered Sets (/O/) depending on the PPM difference.

Idle Insertion

The receiver FIFO inserts eight /I/ codes following an /I/ or /O/ to compensate for clock rate disparity.

Idle (/I/) or Sequence Ordered Set (/O/) Deletion

The receiver FIFO deletes either four /I/ codes or ordered sets (/O/) to compensate for the clock rate disparity. It implements the following IEEE802.3-2008 deletion rules:

- Deletes four /I/ codes if the most significant 32-bits of the preceding word do not contain a Terminate /T/ control character.
- Deletes one /O/ ordered set only when it receives two consecutive /O/ ordered sets.

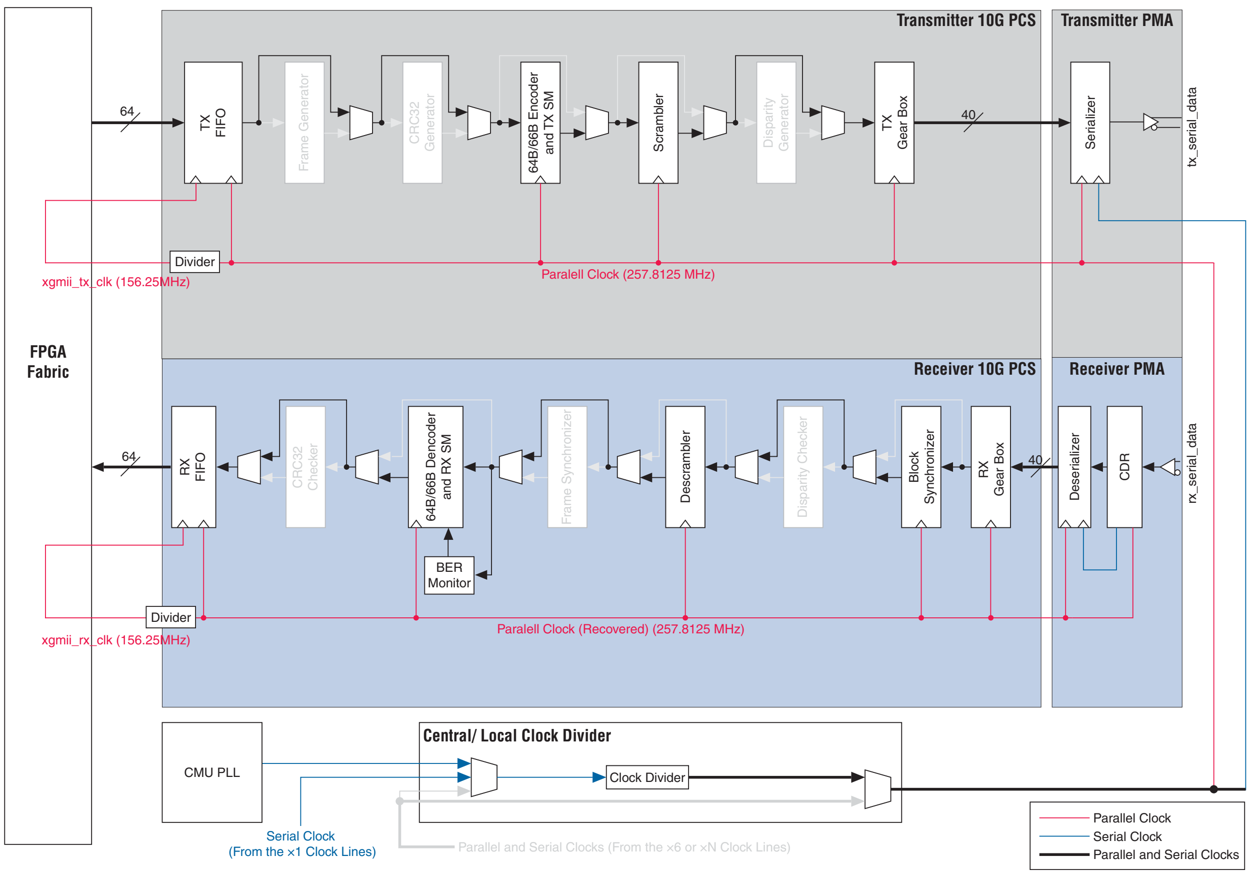
Transceiver Clocking and Channel Placement Guidelines

This section describes the transceiver clocking and channel placement guidelines for the 10GBASE-R protocol supported in Stratix V devices.

Transceiver Clocking

Figure 4–5 shows transceiver clocking in a 10GBASE-R configuration.

Figure 4–5. 10GBASE-R Single-Lane Configuration



One of the two channel phase-locked loops (PLLs) or one of the two auxiliary transmit (ATX) PLLs in a transceiver bank generates the transmitter serial and parallel clocks for the 10GBASE-R channel(s). Table 4–2 lists the input reference clock frequency, FPGA fabric-transceiver interface width, and the interface frequency supported in a 10GBASE-R configuration.

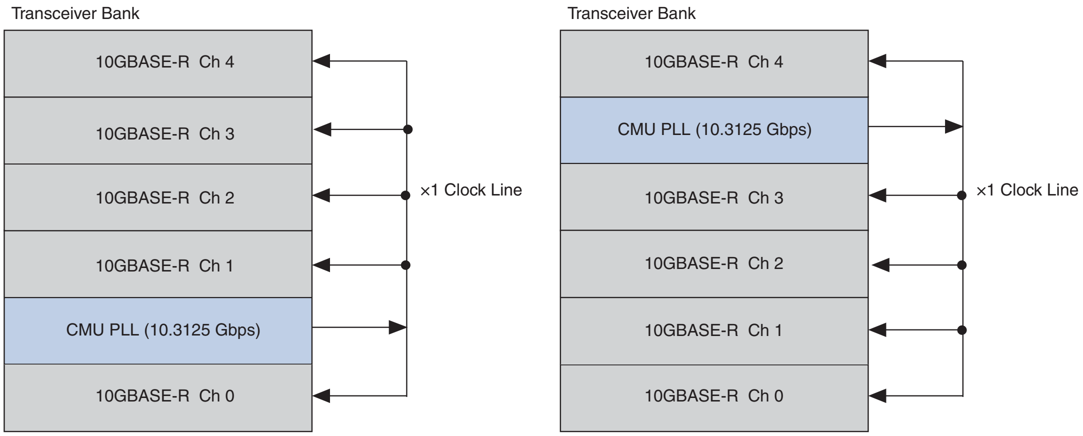
Table 4–2. Input Reference Clock Frequency and Interface Speed Specifications for 10GBASE-R Configurations

Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Width (MHz)
644.53125, 322.265625	64-bit data, 8-bit control	156.25

Transceiver Channel Placement Guidelines

Stratix V devices allow placing up to five 10GBASE-R channels in a transceiver bank using a clock multiplier unit (CMU) channel PLL. However, all six channels in the same transceiver bank can be placed in 10GBASE-R mode using one of the two ATX PLLs in the same transceiver bank. Figure 4–6 shows legal 10GBASE-R channel locations in a transceiver bank using one of the two CMU channel PLLs.

Figure 4–6. Channel Placement Guidelines in 10GBASE-R Configurations



Interlaken

This section describes Interlaken link implementation using Stratix V transceivers. It provides the transceiver channel datapath, clocking, and channel placement guidelines when configured in an Interlaken configuration.

Interlaken is a scalable, chip-to-chip interconnect protocol designed to enable transmission speeds from 10 to 100 Gbps and beyond. Stratix V devices support a transmission speed of up to 10.3125 Gbps per lane in an Interlaken configuration. All the PCS blocks in the Interlaken configuration are designed towards the Interlaken Protocol Definition, Rev 1.2.

In the MegaWizard™ Plug-In Manager, you can implement an Interlaken link by instantiating the **Interlaken PHY IP** core under **Interlaken** in the **Interfaces** menu.

Transceiver Datapath Configuration

Figure 4–7 shows the transceiver blocks and settings enabled in an Interlaken configuration. Blocks shown as “Disabled” are not used, but incur latency. Blocks shown as “Bypassed” are not used and do not incur any latency.

Figure 4–7. Interlaken Configuration

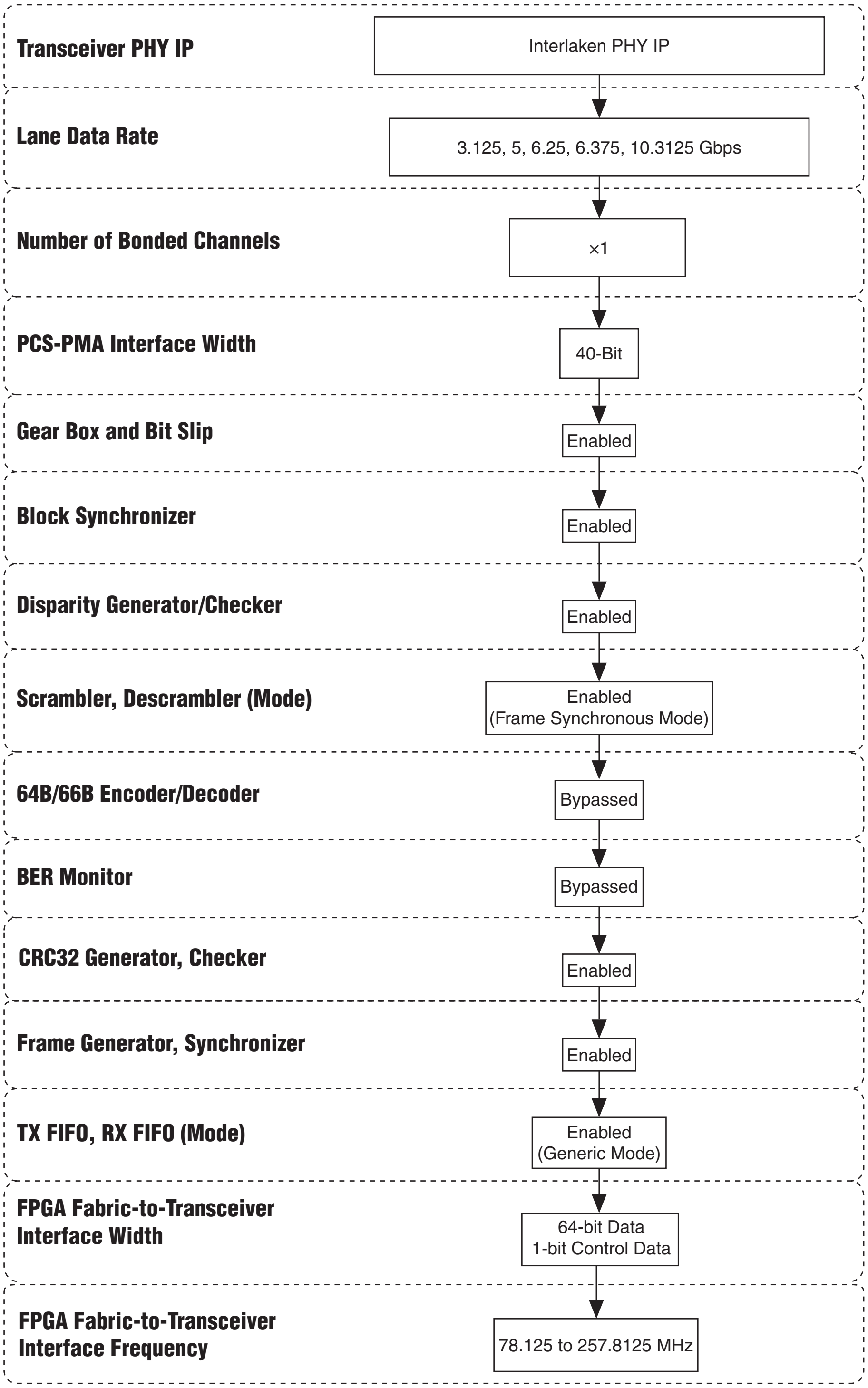
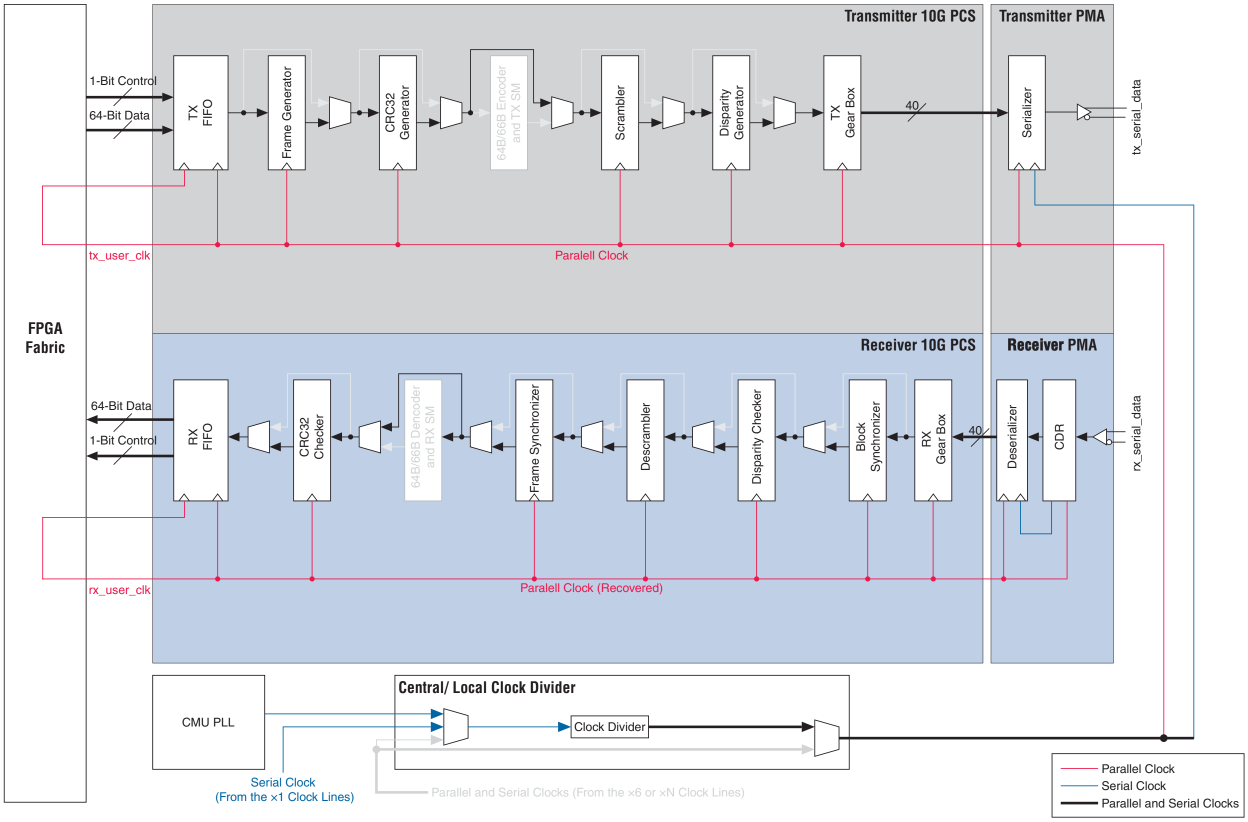


Figure 4–8 shows the PCS and PMA blocks used in the transceiver datapath for Interlaken configurations.

Figure 4–8. Interlaken Channel Datapath



Supported Features

Table 4–3 lists the framing layer functions that are supported by Stratix V devices. These functions are defined in the Interlaken Protocol Definition, Rev 1.2.

Table 4–3. Supported Features in Interlaken Configuration

Feature	Supported
Block synchronization	✓
64B/67B framing	✓
±96 bits disparity maintenance	✓
Frame synchronous scrambling and descrambling	✓
Skip word clock compensation	✓
Diagnostic word generation and CRC-32 checking of lane data integrity	✓

For more information about Interlaken PHY IP control and status signals associated with each feature, refer to the *Interlaken PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Block Synchronization

The block synchronizer in the receiver PCS achieves and maintains a 64B/67B word boundary lock. This block searches for valid synchronization header bits within the data stream and achieves lock after 64 consecutive legal synchronization patterns are found. After a 64B/67B word boundary lock is achieved, it continuously monitors and flags for invalid synchronization header bits. If 16 or more invalid synchronization header bits are found within 64 consecutive word boundaries, the block synchronizer de-asserts the lock state and searches again for valid synchronization header bits.

The block synchronizer implements the flow diagram shown in Figure 13 of Interlaken Protocol Definition v1.2 and provides the word lock status to the FPGA fabric.

64B/67B Framing

The frame generator implements 64B/67B encoding as explained in Interlaken Protocol Definition v1.2, and maps the transmit data into metaframes. The metaframe length is programmable from 5 to a maximum value of 8191, 8-byte words.

☞ Ensure that the metaframe length is programmed to the same value for both the transmitter and receiver.

The frame synchronizer delineates the metaframe boundaries and searches for each of the framing layer control words: Synchronization, Scrambler State, Skip, and Diagnostic. When four consecutive synchronization words have been identified, the frame synchronizer achieves the frame locked state. Subsequent metaframes are then checked for valid synchronization and scrambler state words. If four consecutive invalid synchronization words or three consecutive mismatched scrambler state words are received, the frame synchronizer loses frame lock. In addition, the frame synchronizer provides a receiver metaframe lock status to the FPGA fabric.

Running Disparity

The disparity generator inverts the sense of bits in each transmitted word to maintain a running disparity of ± 96 bit boundary. It supplies a framing bit in bit position 66 as explained in Table 4 of Interlaken Protocol Definition Revision 1.2. The framing bit enables the disparity checker to identify whether the bits for that word are inverted.

Frame Synchronous Scrambling/Descrambling

The scrambler/descrambler block in the transmitter/receiver PCS implements the scrambler/descrambler polynomial $x^{58} + x^{39} + 1$ per Interlaken Protocol Definition Revision 1.2. synchronization and scrambler state words, as well as the 64B/67B framing bits are not scrambled/descrambled. The Interlaken PHY IP core automatically programs random linear feedback shift register (LFSR) initialization seed values per lane.

The receiver PCS synchronizes the scrambler with the metaframe as described in the state flow shown in Figure 1 of Interlaken Protocol Definition Revision 1.2.

The frame synchronizer features a whole set of error and performance monitoring ports to the FPGA fabric interface and register status bits when using the Avalon® Memory-Mapped Management Interface. A receiver ready port, frame lock status, and cyclic redundancy check (CRC)-32 error detection port is available to the FPGA fabric. The Avalon Memory-Mapped Management Interface provides additional functionality with word boundary lock, frame lock status, synchronization word error detection, scrambler mismatch error, and CRC-32 error detection status register bits.

Clock Compensation for Repeater Applications

The receiver FIFO in the receiver Interlaken PCS datapath is capable of compensating a ±100 PPM difference between the remote transmitter and the local receiver using metaframe lengths between 5 to 8191 words. Interlaken employs clock compensation for repeater applications by inserting skip words on the egress traffic and silently deleting skip words on the ingress traffic, depending on the PPM difference.

Skip Word Insertion

The frame generator generates the mandatory skip words with every metaframe following the scrambler state word for clock rate compensation and generates additional skip words based on the transmit FIFO capacity state.

Skip Word Deletion

The frame synchronizer silently discards the skip words it receives.

Diagnostic Word Generation and Checking of Lane Data Integrity (CRC-32)

The CRC-32 generator calculates the CRC for each metaframe and appends it to the diagnostic word of the metaframe. The CRC-32 checker, in addition to checking for lane CRC-32 errors, also retrieves the lane status message in the bit-33 location and link status message in the bit-32 location of the diagnostic word. A CRC-32 error flag is also provided to the FPGA fabric.

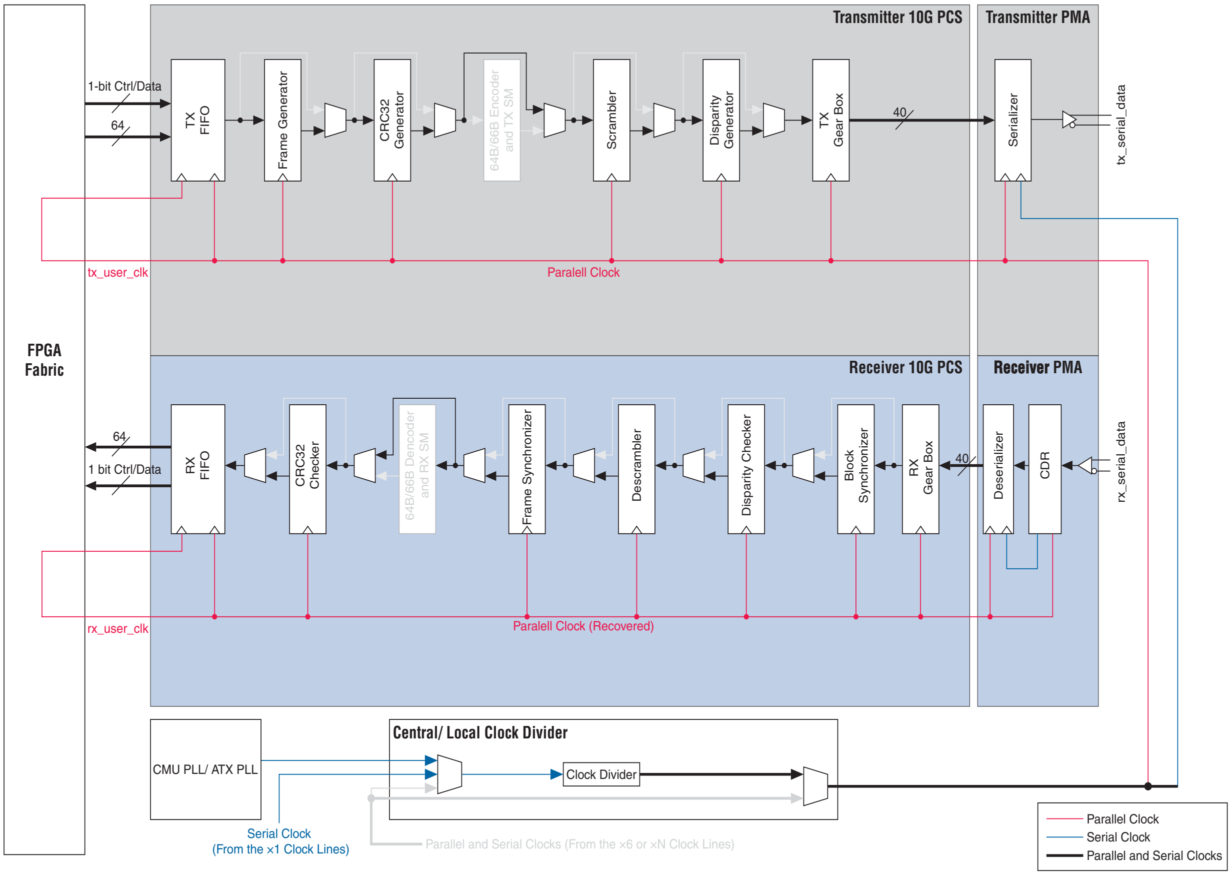
Transceiver Clocking and Channel Placement Guidelines

This section describes the transceiver clocking and channel placement guidelines for the Interlaken protocol supported in Stratix V devices.

Transceiver Clocking

Current Interlaken protocol supports single-lane clocking (non-bonded configurations) only. Figure 4–9 shows the clocking resources available in a single-lane Interlaken configuration.

Figure 4–9. Interlaken Single-Lane Configuration



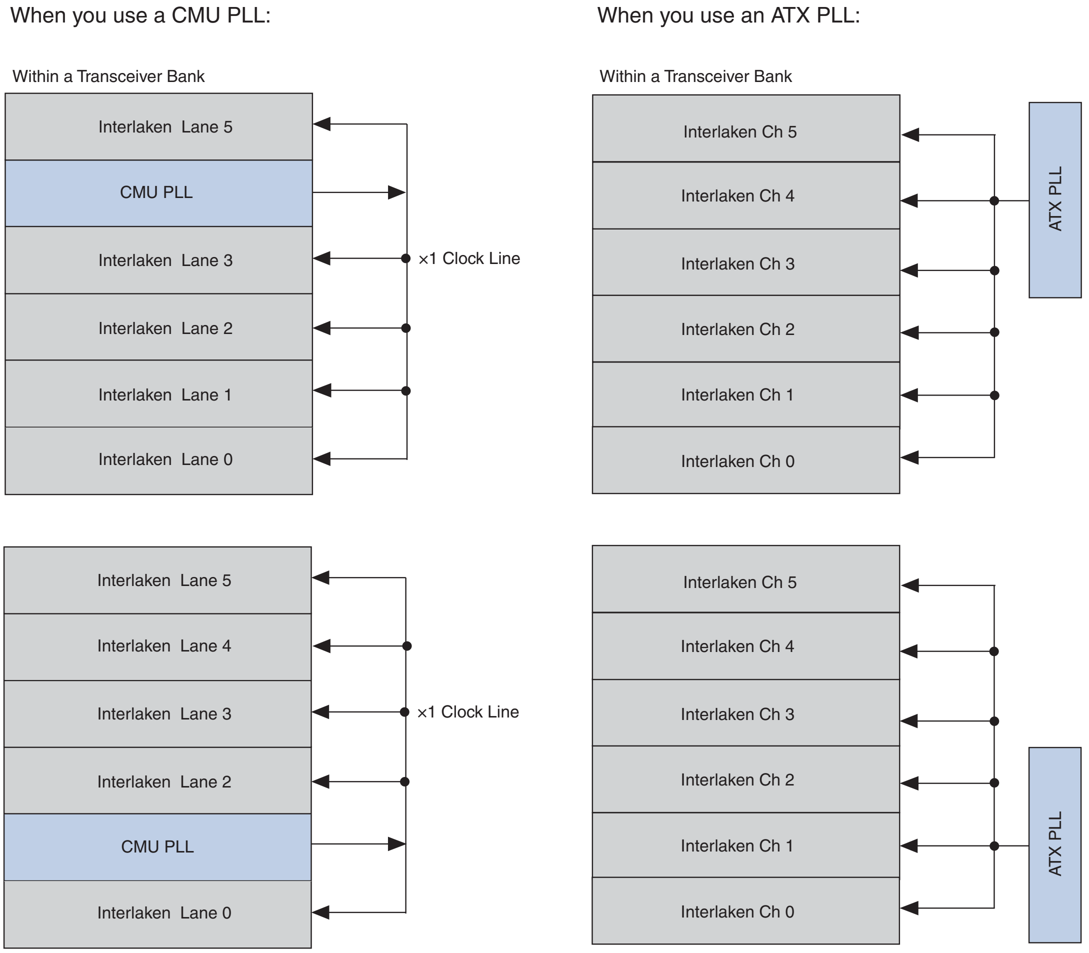
A CMU PLL or ATX PLL may provide a clock to up to five Interlaken channels within a transceiver bank of six channels.

Transceiver Channel Placement Guidelines

Stratix V devices allow placing up to five Interlaken channels in a transceiver bank. Figure 4–10 shows the legal Interlaken channel locations in a transceiver bank when using the CMU PLL or when using the ATX PLL.

To enable the ATX PLL, you must select a minimum bond size of **six** in the **Bonded Group Size** parameter in the **Interlaken PHY IP**. You must also select the **ATX PLL** from the Quartus II Assignment Editor.

Figure 4–10. Channel Placement Guidelines in Interlaken Configurations



For more information about channel placement guidelines, refer to the *Internal Clocking* section in the *Transceiver Clocking in Stratix V Devices* chapter.

PCI Express (PCIe)—Gen1 and Gen2

The PCIe specification (version 2.0) provides implementation details for a PCIe-compliant physical layer device at both Gen1 (2.5 Gbps) and Gen2 (5 Gbps) signaling rates.

Stratix V devices have built-in PCIe hard IP blocks to implement the PHY MAC layer, data link layer, and transaction layer of the PCIe protocol stack. The hard IP block resides in the Embedded Hardcopy Block within the Stratix V device. To implement a PCI Express-compliant PHY, configure the Stratix V transceiver in PCIe configuration. If you enable the PCIe hard IP block, the transceiver interfaces with the hard IP block. Otherwise, the transceiver interfaces directly with the FPGA fabric.

You can configure the Stratix V transceivers in a PCIe functional configuration using one of the following methods:

- PHY interface for PCI Express (PIPE)—PCIe hard IP block disabled
- PCIe compiler—PCIe hard IP block enabled

For a description of the PCIe hard IP architecture and the allowed PCIe configurations when you enable PCIe hard IP, refer to the *PCI Express Compiler User Guide*.

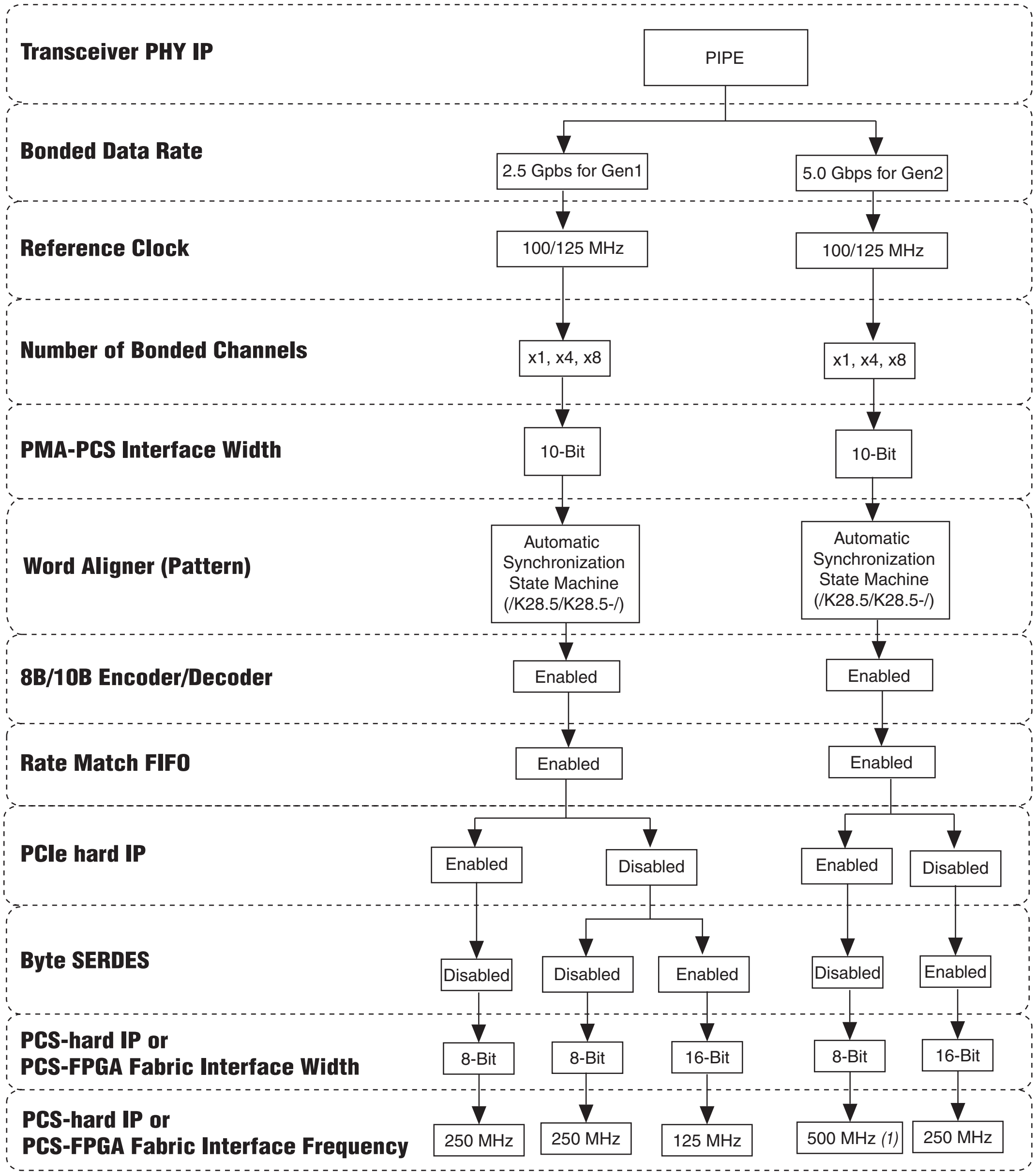
Stratix V devices support both Gen1 and Gen2 data rates in a PIPE configuration. When configured for the Gen2 data rate, the Stratix V transceivers allow dynamic switching between Gen2 and Gen1 line rates. Dynamic switching between the two line rates is critical for speed negotiation during link training.

Stratix V transceivers support ×1, ×4, and ×8 lane configurations in both 2.5 Gbps and 5 Gbps data rates. In a PCIe ×1 configuration, the PCS and PMA blocks of each channel are clocked and reset independently. PCIe ×4 and ×8 configurations support channel bonding for four-lane and eight-lane PCIe links. In these bonded channel configurations, the PCS and PMA blocks of all bonded channels share common clock and reset signals.

Transceiver Datapath Configuration

Figure 4–11 shows the transceiver configurations allowed in a PIPE configuration.

Figure 4–11. Stratix V Transceivers in a PIPE Configuration



Note to Figure 4–11:

(1) Applies to a PCS-hard IP interface. A PCS-FPGA fabric interface frequency is limited to 250 MHz.

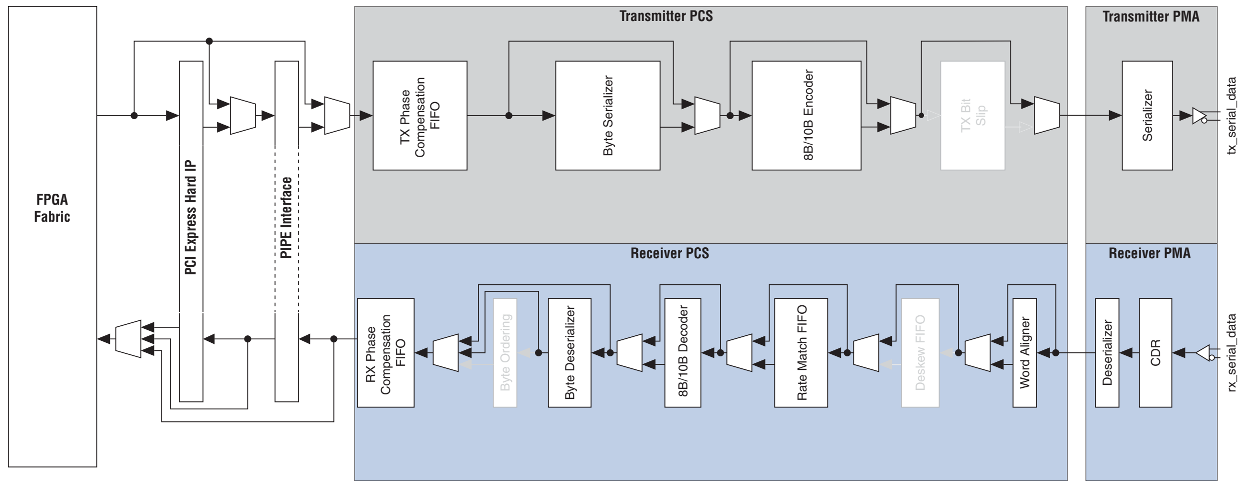
Transceiver datapath clocking varies between non-bonded (×1) and bonded (×4 and ×8) configurations.

For more information about transceiver datapath clocking in different PIPE configurations, refer to "Transceiver Clocking" on page 4–24.

Transceiver Channel Datapath

Figure 4–12 shows the Stratix V transmitter and receiver channel datapath in a PCIe configuration.

Figure 4–12. Stratix V Transmitter Channel Datapath in a PCIe Configurations



For more information about the blocks in the transmitter datapath, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

Supported Features

Table 4–4 lists the features supported in a PCIe configuration for the 2.5 Gbps and 5 Gbps data rate configurations.

Table 4–4. Supported Features in PCIe Configurations

Feature	Gen1 (12.5 Gbps)	Gen2 (5 Gbps)
×1, ×4, ×8 link configurations	✓	✓
PCIe-compliant synchronization state machine	✓	✓
±300 PPM (total 600 PPM) clock rate compensation	✓	✓
8-bit FPGA fabric-transceiver interface	✓	—
16-bit FPGA fabric-transceiver interface	✓	✓
Transmitter buffer electrical idle	✓	✓
Receiver Detection	✓	✓
8B/10B encoder disparity control when transmitting compliance pattern	✓	✓
Power state management	✓	✓
Receiver status encoding	✓	✓
Dynamic switching between 2.5 Gbps and 5 Gbps signaling rate	—	✓
Dynamically selectable transmitter margining for differential output voltage control	—	✓
Dynamically selectable transmitter buffer de-emphasis of -3.5 dB and -6 dB	—	✓

PIPE 2.0 Interface

In a PCIe configuration, each channel has a PIPE interface block that transfers data, control, and status signals between the PHY-MAC layer and the transceiver channel PCS and PMA blocks. The PIPE interface block complies with the PIPE 2.0 specification. If you use the PIPE hard IP block, the PHY-MAC layer is implemented in the hard IP block. If you use a PIPE configuration, you must implement the PHY-MAC layer using soft IP in the FPGA fabric.

☞ The PIPE interface block is only used in a PIPE configuration and cannot be bypassed.

Besides transferring data, control, and status signals between the PHY-MAC layer and the transceiver, the PIPE interface block implements the following functions required in a PCIe-compliant physical layer device:

- Forcing the transmitter buffer in the electrical idle state
- Initiating the receiver detect sequence
- Controlling the 8B/10B encoder disparity when transmitting compliance pattern
- Managing the PCIe power states
- Indicating the completion of various PHY functions; for example, receiver detection and power state transitions on the `pipe_phystatus` signal
- Encoding the receiver status and error conditions on the `pipe_rxstatus[2:0]` signal, as specified in the PCIe specification

PCI Express Gen2 (5 Gbps) Support

The PIPE configuration supports the following additional features when configured for the 5 Gbps data rate:

- Dynamic switching between 2.5 Gbps and 5 Gbps signaling rate
- Dynamic selection of transmitter margining for differential output voltage control
- Dynamic selection of transmitter buffer de-emphasis of –3.5 dB and –6 dB

Dynamic Switching Between Gen1 (2.5 Gbps) and Gen2 (5 Gbps) Signal Rates

In a PIPE configuration, the PIPE MegaWizard™ Plug-In Manager provides an input signal (`pipe_rate`) that is functionally equivalent to the `RATE` signal specified in the PCIe specification. A low-to-high transition on this input signal (`pipe_rate`) initiates a data rate switch from Gen1 to Gen2. A high-to-low transition on the input signal initiates a data rate switch from Gen2 to Gen1. The signaling rate switch between Gen1 and Gen2 is achieved by changing the transceiver datapath clock frequency between 250 MHz and 500 MHz, while maintaining a constant, 16-bit width transceiver interface.

For more information about using this input signal, and a timing diagram showing the sequence of rate switch events and status signals, refer to the *PCI Express PIPE PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*. For the power state requirements when switching between Gen1 and Gen2 data rates, refer to the PCIe Base Specification 2.0.

Transmitter Electrical Idle Generation

The PIPE interface block in Stratix V devices puts the transmitter buffer in the channel in an electrical idle state when the electrical idle input signal is asserted. During electrical idle, the transmitter buffer differential and common configuration output voltage levels are compliant to the PCIe Base Specification 2.0 for both PCIe Gen1 and Gen2 data rates.

The PCIe specification requires the transmitter buffer to be in electrical idle in certain power states. For more information about input signal levels required in different power states, refer to "Power State Management".

For more information about the electrical idle input signal and transmitter buffer state, refer to the *PCI Express PIPE PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Power State Management

The PCIe specification defines four power states—P0, P0s, P1, and P2—that the physical layer device must support to minimize power consumption:

- P0 is the normal operating state during which packet data is transferred on the PCIe link.
- P0s, P1, and P2 are low-power states into which the physical layer must transition as directed by the PHY-MAC layer to minimize power consumption.

The PIPE interface in Stratix V transceivers provides an input port for each transceiver channel configured in a PIPE configuration.

For more information about input signals and status signals to manipulate power states, refer to the *PCI Express PIPE PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

When transitioning from the P0 power state to lower power states (P0s, P1, and P2), the PCIe specification requires the physical layer device to implement power saving measures. Stratix V and transceivers do not implement these power saving measures except putting the transmitter buffer in electrical idle in the lower power states.

8B/10B Encoder Usage for Compliance Pattern Transmission Support

The PCIe transmitter transmits a compliance pattern when the Link Training and Status State Machine (LTSSM) state machine enters a polling compliance substate. The polling compliance substate assesses if the transmitter is electrically compliant with the PCIe voltage and timing specifications.

For more information about the 8B/10B signals required for compliance pattern transmission support, refer to the *PCI Express PIPE PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Receiver Electrical Idle Inference

The PCIe protocol allows inferring the electrical idle condition at the receiver instead of detecting the electrical idle condition using analog circuitry.

In all PIPE configurations, (×1, ×4, and ×8), each receiver channel PCS has an optional Electrical Idle Inference module designed to implement the electrical idle inference conditions specified in the PCIe Base Specification 2.0.

Receiver Status

The PCIe specification requires the PHY to encode the receiver status on a 3-bit status signal (`pipe_rxstatus[2:0]`). This status signal is used by the PHY-MAC layer for its operation. The PIPE interface block receives status signals from the transceiver channel PCS and PMA blocks, and encodes the status on the `pipe_rxstatus[2:0]` signal to the FPGA fabric. The encoding of the status signals on the `pipe_rxstatus[2:0]` signal is compliant with the PCIe specification.

For information about the encoding of status signals on the `pipe_rxstatus[2:0]` signal, refer to the *PCI Express PIPE PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Receiver Detection

The PIPE interface block in Stratix V transceivers provides an input signal (`pipe_txdetectrx_loopback`) for the receiver detect operation required by the PCIe protocol during the detect substate of the LTSSM. When the `pipe_txdetectrx_loopback` signal is asserted in the P1 power state, the PCIe interface block sends a command signal to the transmitter buffer in that channel to initiate a receiver detect sequence. In the P1 power state, the transmitter buffer must always be in the electrical idle state. After receiving this command signal, the receiver detect circuitry creates a step voltage at the output of the transmitter buffer. If an active receiver (that complies with the PCIe input impedance requirements) is present at the far end, the time constant of the step voltage on the trace is higher when compared with the time constant of the step voltage when the receiver is not present. The receiver detect circuitry monitors the time constant of the step signal seen on the trace to determine if a receiver was detected. The receiver detect circuitry monitor requires a 125-MHz clock for operation that you must drive on the `fixedclk` port.

For the receiver detect circuitry to function reliably, the AC-coupling capacitor on the serial link and the receiver termination values used in your system must be compliant with the PCIe Base Specification 2.0.

The PIPE core provides a 1-bit PHY status (`pipe_phystatus`) and a 3-bit receiver status signal (`pipe_rxstatus[2:0]`) to indicate whether a receiver was detected or not, as per the PIPE 2.0 specifications.

For more information about input signals and status signals related to receiver detection, refer to the *PCI Express PIPE PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

Clock Rate Compensation Up to ±300 PPM

In compliance with the PCIe protocol, Stratix V receiver channels are equipped with a rate match FIFO to compensate for small clock frequency differences up to ±300 PPM between the upstream transmitter and the local receiver clocks.

For more information about the rate match FIFO operation in a PCIe configuration, refer to the *Transceiver Architecture in Stratix V Devices* chapter.

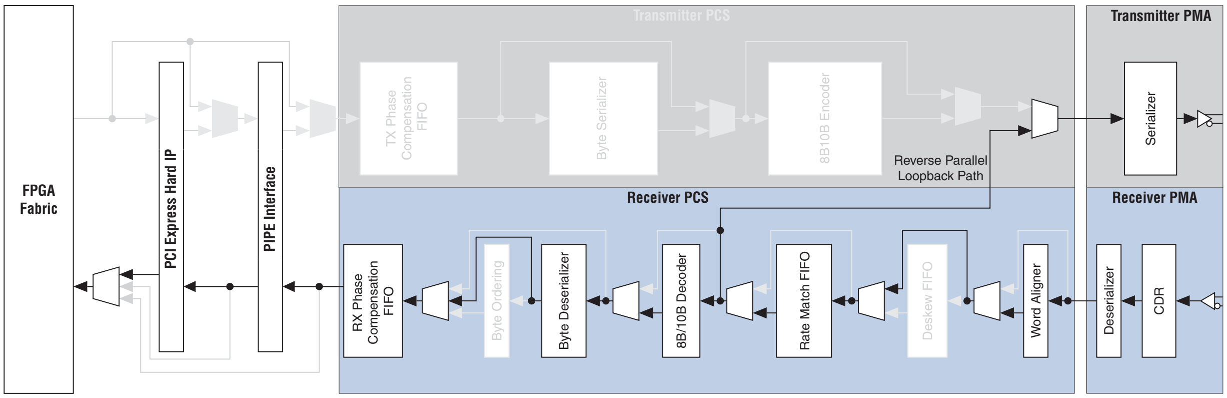
For more information about status signals and registers for the rate match FIFO, refer to the *PCI Express PIPE PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

PCIe Reverse Parallel Loopback

PCIe reverse parallel loopback is only available in a PCIe functional configuration for Gen1 and Gen2 data rates. As shown in Figure 4–13, the received serial data passes through the receiver CDR, deserializer, word aligner, and rate matching FIFO buffer. It is then looped back to the transmitter serializer and transmitted out through the transmitter buffer. The received data is also available to the FPGA fabric through the port. This loopback mode is compliant with the PCIe specification 2.0. Stratix V devices provide an input signal to enable this loopback mode.

This is the only loopback option supported in PIPE configurations.

Figure 4–13. PCIe Reverse Parallel Loopback Mode Datapath (Grayed-out Blocks are Inactive)



Transceiver Clocking and Channel Placement Guidelines

This section describes the transceiver clocking and channel placement guidelines for PIPE configurations.

☞ For PIPE Gen1 configurations, ATX PLL is currently not supported for 100 MHz reference clock.

Transceiver Channel Placement Guidelines

Table 4–5 lists the physical placement of PIPE channels in ×1, ×4, and ×8 bonding configurations. The Quartus II software automatically places the CMU PLL in a channel different from that of the data channels.

Table 4–5. PIPE Configuration Channel Placement

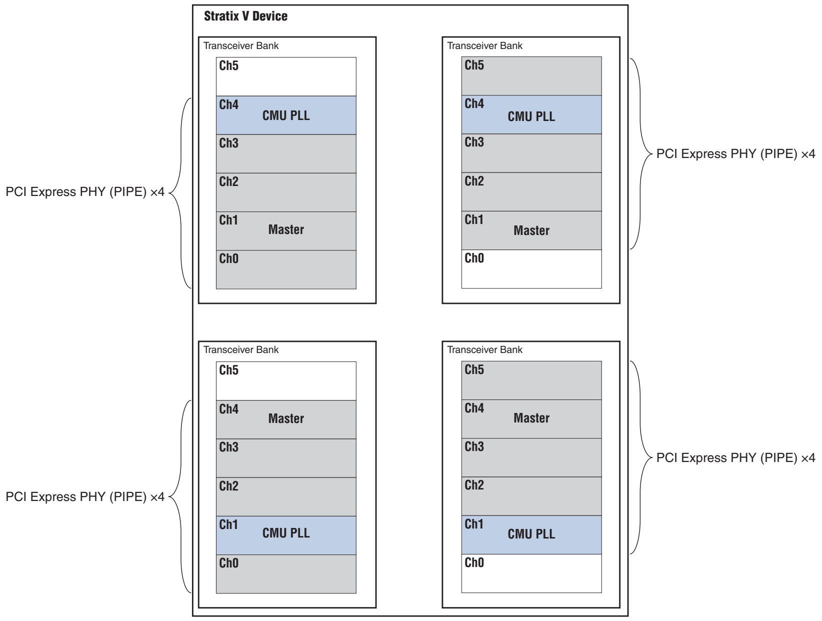
Configuration	Data Channel Placement	Channel Utilization Using CMU PLL *(1)*	Channel Utilization Using ATX PLL *(1)*
×1	Any channel	2	1
×4	Contiguous channels	5	4
×8	Contiguous channels	9	8

Notes to Table 4–5:

(1) Placement by the Quartus II software may vary with design, thus resulting in higher channel usage.

For PIPE ×1 configurations, the channel can be placed anywhere within a transceiver bank that contains the transmit PLL. Figure 4–14 and Figure 4–15 show examples of channel placement for PIPE ×4 and ×8 configurations.

Figure 4–14. Example of PIPE ×4 Channel Placement Using the CMU PLL *(Note 1)*, *(2)*, *(3)*



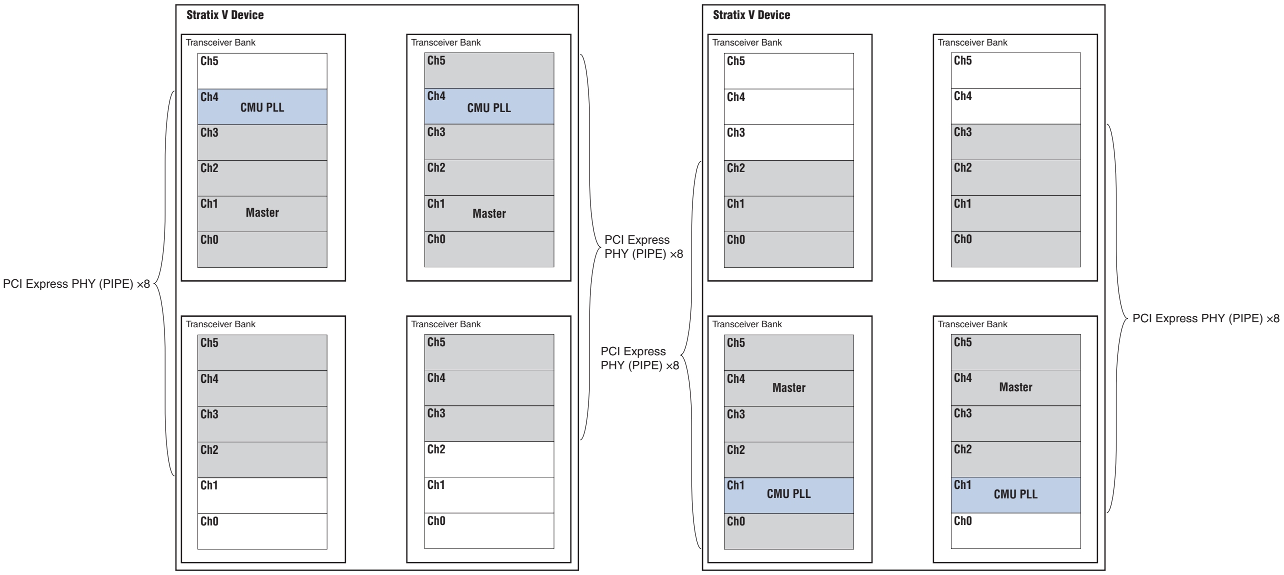
Notes to Figure 4–14:

(1) Channels shaded in blue provide the high-speed serial clock.

(2) Channels shaded in gray are data channels.

(3) The Quartus II software automatically places the clock generator and master channel in either channel 1 or channel 4 within a transceiver bank.

Figure 4–15. Example of PIPE ×8 Channel Placement Using a CMU PLL *(Note 1)*, *(2)*



Notes to Figure 4–15:

(1) Channels shaded in blue provide the serial clock.

(2) Channels shaded in gray are data channels.

Transceiver Clocking

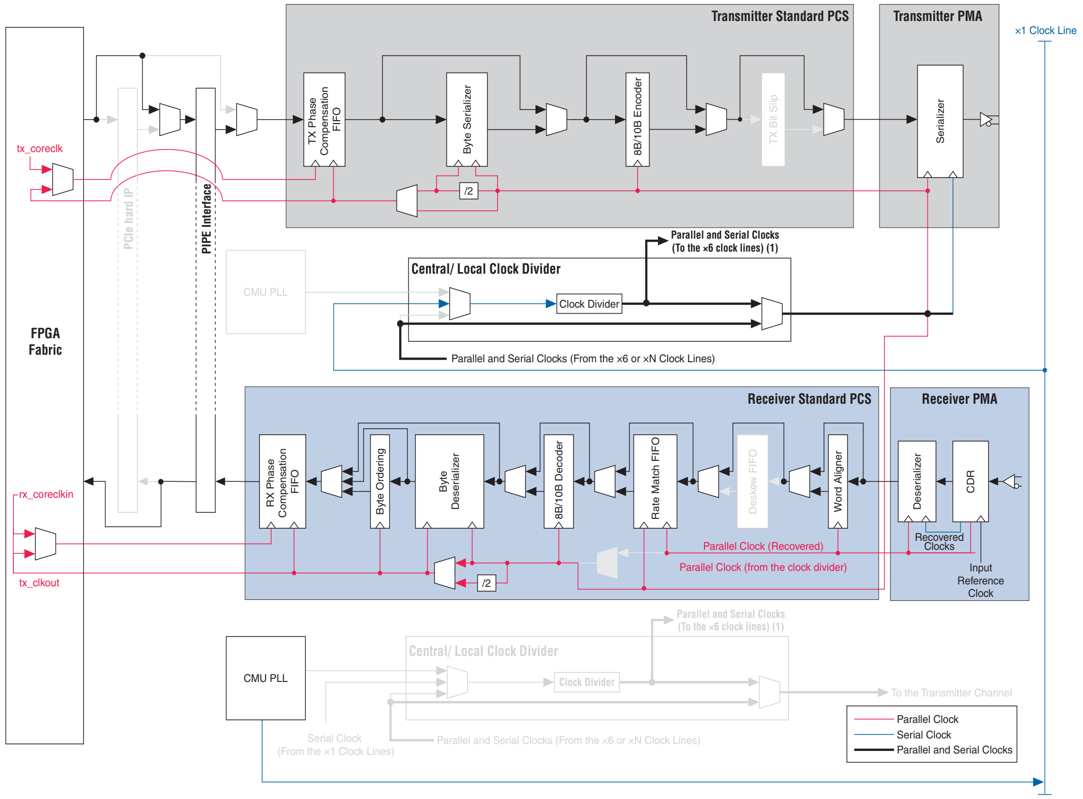
This section describes the transceiver clocking configurations for PIPE.

PIPE ×1 Configuration

Figure 4–16 shows the transceiver clocking configuration in a PIPE ×1 configuration.

The serial clock is provided by the CMU PLL in a channel different from that of the data channel. The local clock divider block in the data channel generates a parallel clock from this high-speed clock and distributes both clocks to the PMA and PCS of the data channel.

Figure 4–16. Transceiver Clocking Configuration in a PIPE ×1 Configuration


Transmitter Standard PCS
Transmitter PMA
×1 Clock Line
TX Phase Compensation FIFO
Byte Serializer
8B/10B Encoder
TX Bit Slip
Serializer
/2
tx_coreclk
PCIe hard IP
PIPE Interface
Parallel and Serial Clocks (To the ×6 clock lines) (1)
Central/ Local Clock Divider
CMU PLL
Clock Divider
Parallel and Serial Clocks (From the ×6 or ×N Clock Lines)
FPGA Fabric
Receiver Standard PCS
Receiver PMA
RX Phase Compensation FIFO
Byte Ordering
Byte Deserializer
8B/10B Decoder
Rate Match FIFO
Deskew FIFO
Word Aligner
Deserializer
CDR
rx_coreclkin
tx_clkout
Parallel Clock (Recovered)
Parallel Clock (from the clock divider)
/2
Recovered Clocks
Input Reference Clock
Parallel and Serial Clocks (To the ×6 clock lines) (1)
Central/ Local Clock Divider
CMU PLL
Clock Divider
To the Transmitter Channel
Serial Clock (From the ×1 Clock Lines)
Parallel and Serial Clocks (From the ×6 or ×N Clock Lines)
Parallel Clock
Serial Clock
Parallel and Serial Clocks

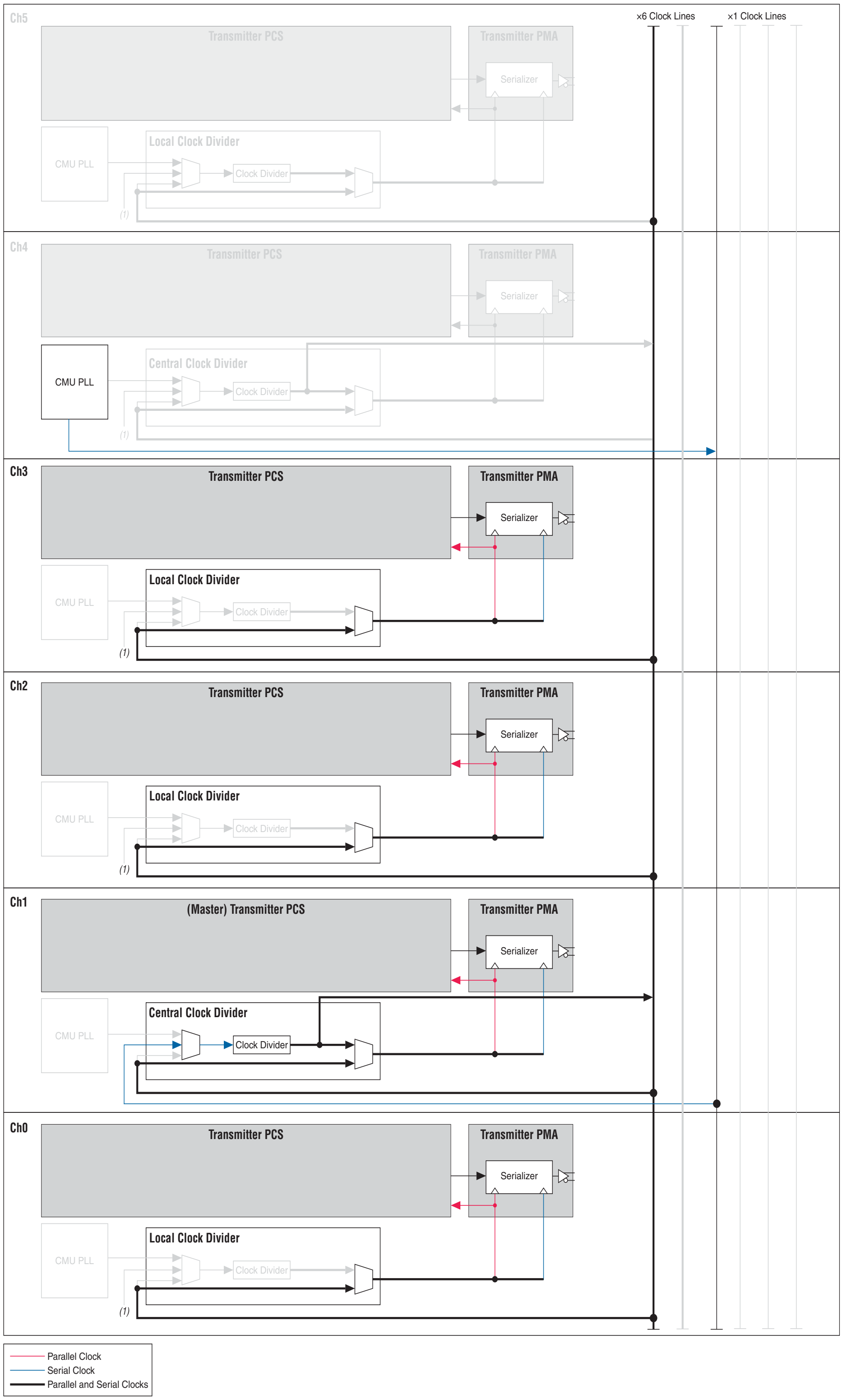
Note to Figure 4–16:

(1) Only available in the central clock dividers of channel 1 and channel 4 in a transceiver bank.

PIPE ×4 Configuration

Figure 4–17 shows transmitter clocking for a PIPE ×4 bonded configuration. Clocking within the PCS is independent for each receiver channel. Clocking is bonded only for transmit channels, whereas the control signals are bonded for both transmit and receive channels. The Quartus II software automatically places the clock generator and master channel in either channel 1 or channel 4 within a transceiver bank, as shown in Figure 4–18.

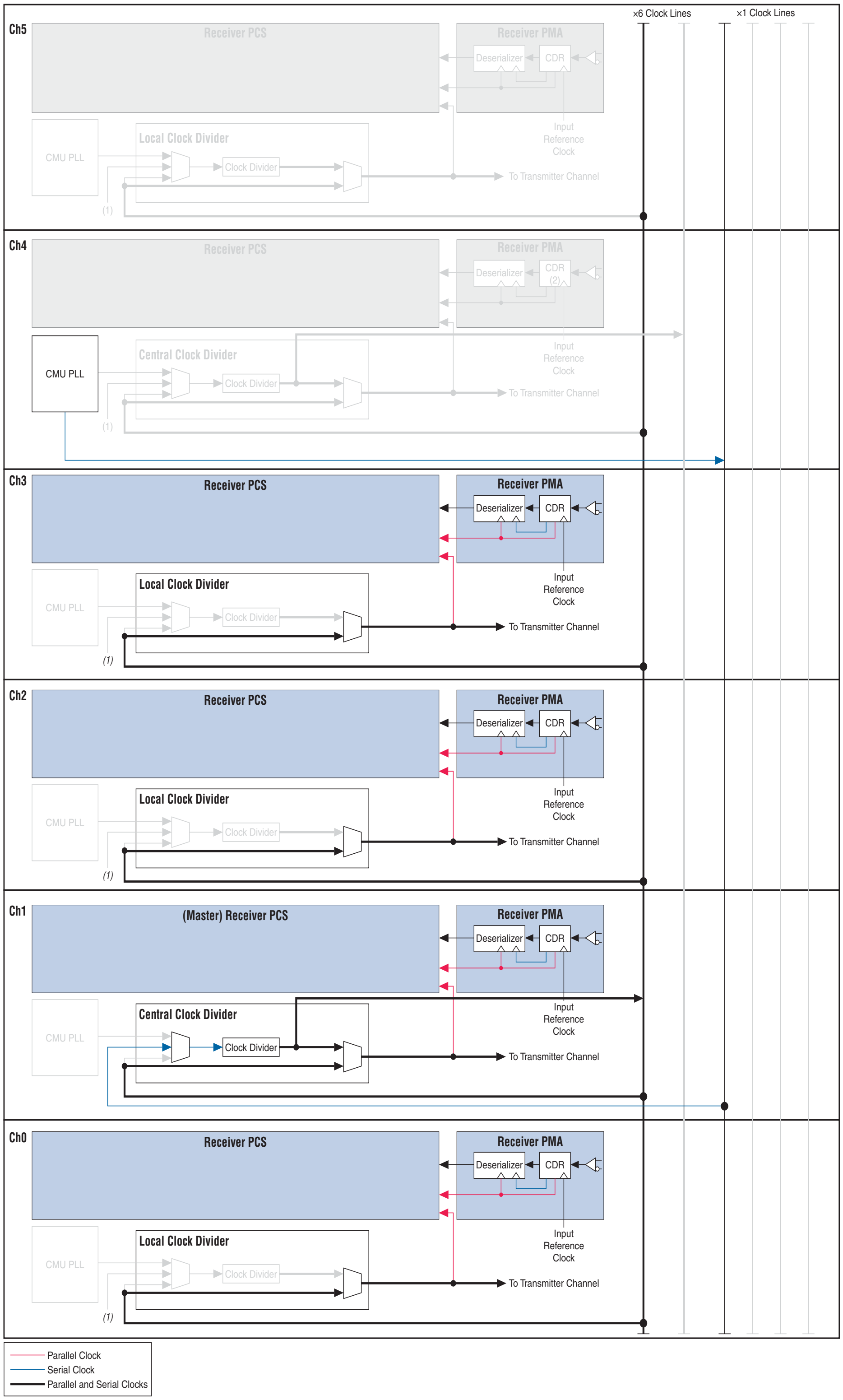
Figure 4–17. Transmitter Clocking Configuration in a PIPE ×4 Configuration



Note to Figure 4–17:

(1) Serial clock from the ×1 clock lines.

Figure 4–18. Receiver Clocking Configuration in a PIPE ×4 Configuration



Note to Figure 4–18:

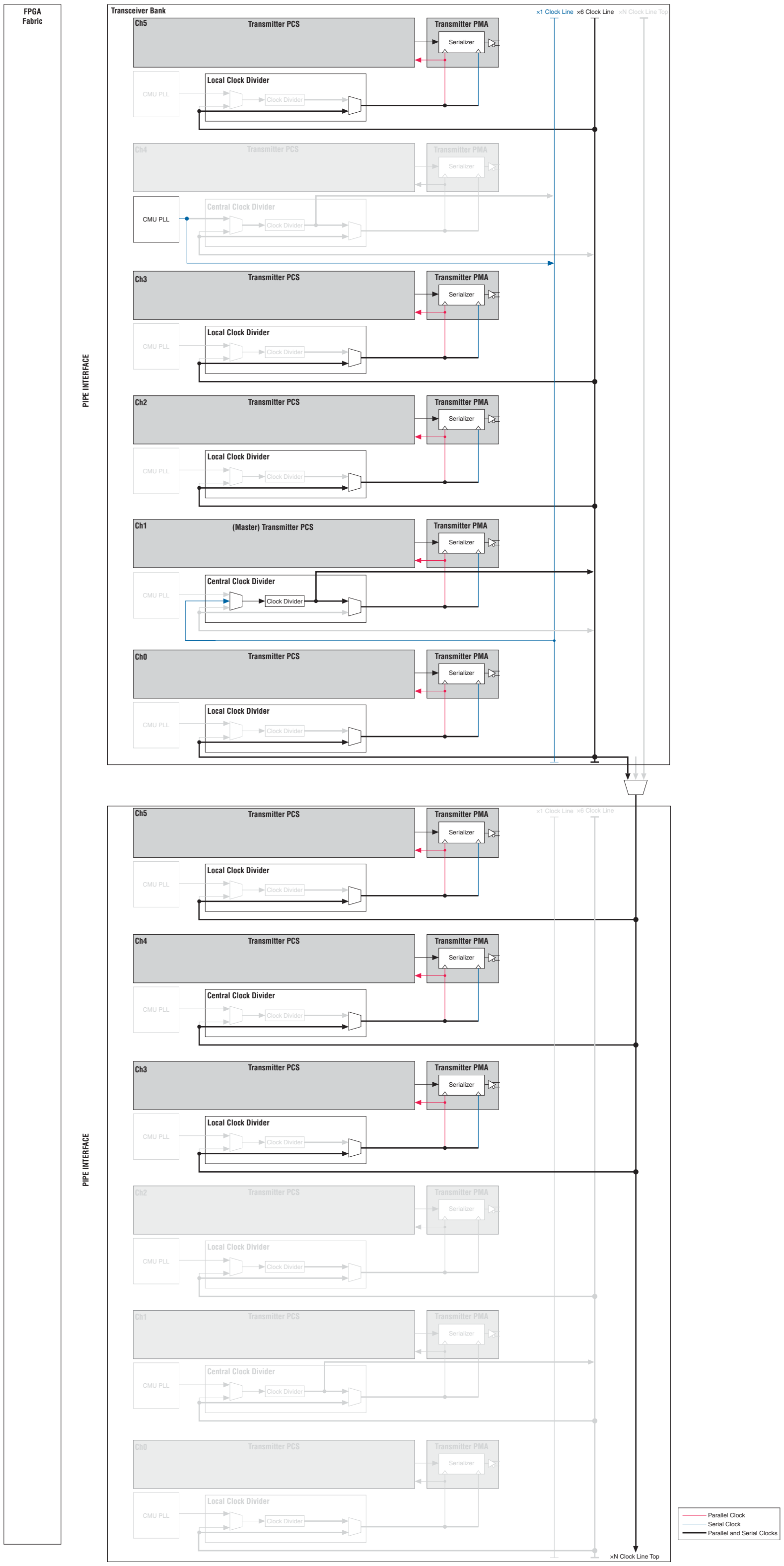
(1) Serial clock from the ×1 clock lines.

PIPE ×8 Configuration

Figure 4–19 shows clocking for PMA and PCS blocks in the ×8 PCIe bonded configuration. Clocking is independent for receiver channels. Clocking and control signals are bonded only for transmitter channels.

For more information about clocking in Stratix V devices, refer to the *Transceiver Clocking in Stratix V Devices* chapter.

Figure 4–19. Transceiver Clocking Configuration in a PIPE ×8 Configuration



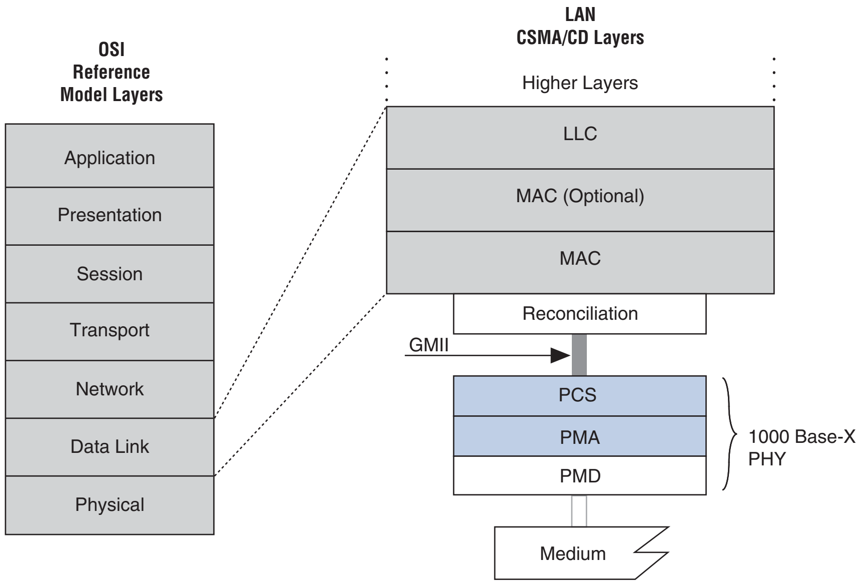
GIGE

This section describes Gigabit Ethernet (GIGE) link implementation using Stratix V transceivers. It provides the transceiver channel datapath, clocking, and channel placement guidelines when configured in a GIGE configuration.

IEEE 802.3 defines the 1000 Base-X PHY as an intermediate, or transition, layer that interfaces various physical media with the MAC in a GIGE system. It shields the MAC layer from the specific nature of the underlying medium. The 1000 Base-X PHY, which has a physical interface data rate of 1.25 Gbps, is subdivided into three sublayers—the physical coding sublayer (PCS), physical media attachment (PMA), and physical medium dependent (PMD). These sublayers interface with the MAC through the gigabit medium independent interface (GMII).

Figure 4–20 shows the 1000 Base-X PHY position in a Gigabit Ethernet OSI reference model.

Figure 4–20. 1000 Base-X PHY in a Gigabit Ethernet OSI Reference Model



When configured in GIGE functional mode, Stratix V transceivers have built-in PCS and PMA circuitry to support 8B/10B encoding and decoding, synchronization, rate matching, CDR, and serialization and deserialization.

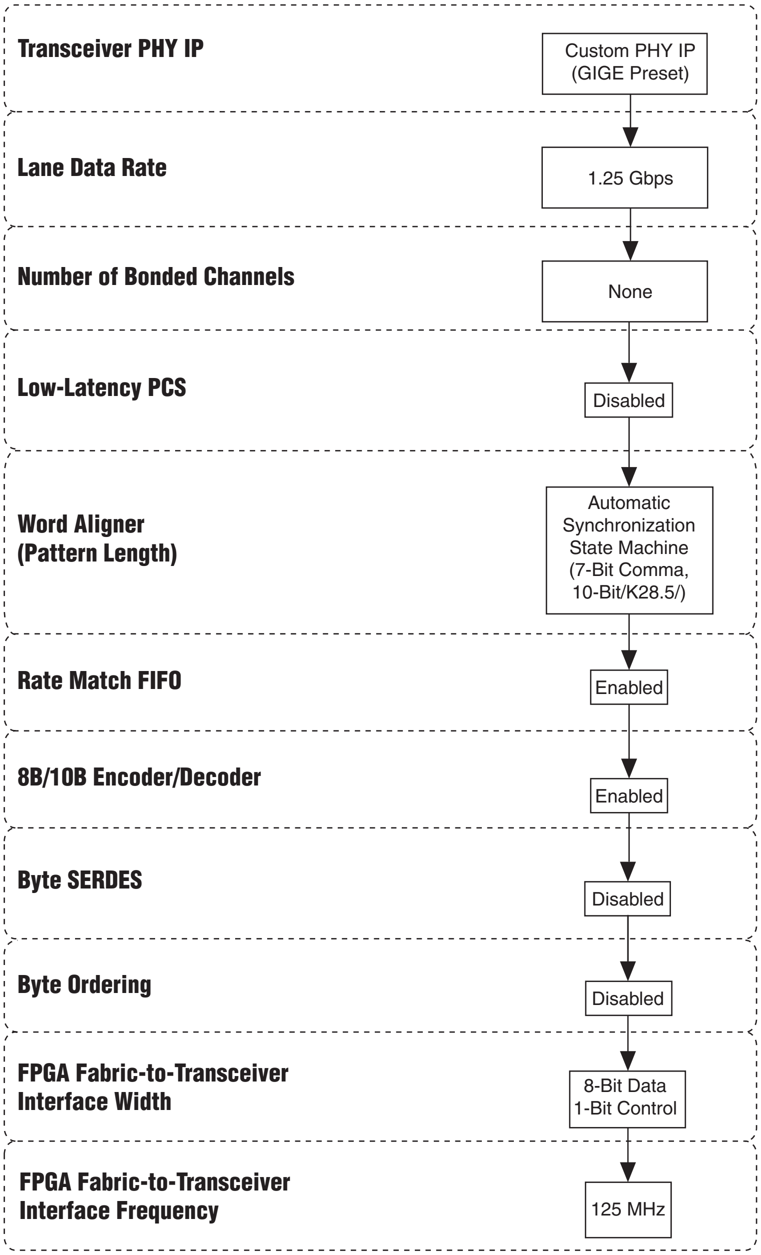
In the MegaWizard Plug-In Manager, you can implement a GIGE link by instantiating the **Custom PHY IP** core and selecting the **GIGE Preset** under **Transceiver PHY** in the **Interfaces** menu.

☞ Stratix V transceivers do not have built-in hard-IP support for other PCS functions, such as idle-ordered sets substitution, auto-negotiation state machine, collision-detect, and carrier-sense. If required, these functions must be implemented in a programmable logic device (PLD) logic array or external circuits.

Transceiver Datapath Configuration

Figure 4–21 shows the transceiver blocks and settings enabled in a GIGE configuration. Blocks shown as "Disabled" are not used, but incur latency. Blocks shown as "Bypassed" are not used and do not incur any latency.

Figure 4–21. GIGE Mode for Stratix V Devices


Transceiver PHY IP
Custom PHY IP (GIGE Preset)
Lane Data Rate
1.25 Gbps
Number of Bonded Channels
None
Low-Latency PCS
Disabled
Word Aligner (Pattern Length)
Automatic Synchronization State Machine (7-Bit Comma, 10-Bit/K28.5/)
Rate Match FIFO
Enabled
8B/10B Encoder/Decoder
Enabled
Byte SERDES
Disabled
Byte Ordering
Disabled
FPGA Fabric-to-Transceiver Interface Width
8-Bit Data 1-Bit Control
FPGA Fabric-to-Transceiver Interface Frequency
125 MHz

Supported Features

Table 4–6 lists the GIGE-PCS functions that are supported by Stratix V devices. These functions are defined in the IEEE 802.3-2008 GIGE protocol definition, Section 3 Clause 36.

Table 4–6. Supported Features in a GIGE Configuration

Feature	Supported
8B/10B Encoding/Decoding	✓
Synchronization	✓
Clock Compensation using Rate Matching	✓

8B/10B Encoding/Decoding

In GIGE mode, the 8B/10B encoder clocks in 8-bit data and 1-bit control identifiers from the transmitter phase compensation FIFO and generates 10-bit encoded data. 8B/10B encoding limits the maximum number of consecutive 1s and 0s in the serial data stream to five, thereby ensuring DC balance as well as enough transitions for the receiver CDR to maintain a lock to the incoming data. The 10-bit encoded data is then fed to the serializer for transmission.

The 8B/10B decoder parses the serial stream of code groups from the rate match FIFO block and recovers the 8-bit data and control characters. The GIGE Custom PHY IP provides a running disparity error, run length violation error, and invalid 8B/10B code group error status ports.

Idle Ordered-Set Generation/Substitution

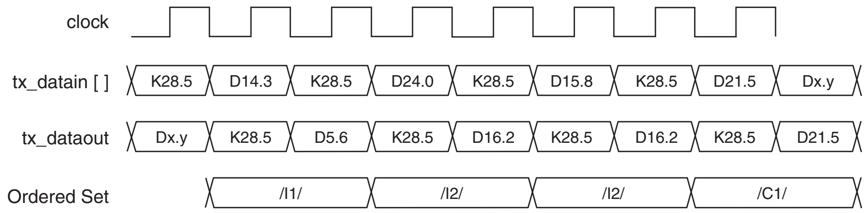
The IEEE 802.3 specification requires the GIGE PHY to transmit idle ordered sets (/I/) continuously and repetitively whenever the GMII is idle. This ensures that the receiver maintains bit and word synchronization whenever there is no active data to be transmitted. Idle ordered-set substitution must be implemented in a PLD logic array.

In GIGE functional mode, any /Dx.y/ following a /K28.5/ comma is replaced by the transmitter with either a /D5.6/ (/I1/ ordered set) or a /D16.2/ (/I2/ ordered set), depending on the current running disparity. The exception is when the data following the /K28.5/ is /D21.5/ (/C1/ ordered set) or /D2.2/ (/C2/) ordered set. If the running disparity before the /K28.5/ is positive, an /I1/ ordered set is generated. If the running disparity is negative, a /I2/ ordered set is generated. The disparity at the end of a /I1/ is the opposite of that at the beginning of the /I1/. The disparity at the end of a /I2/ is the same as the beginning running disparity (right before the idle code). This ensures a negative running disparity at the end of an idle ordered set. A /Kx.y/ following a /K28.5/ is not replaced.

☞ Note that /D14.3/, /D24.0/, and /D15.8/ are replaced by /D5.6/ or /D16.2/ (for /I1/, /I2/ ordered sets). /D21.5/ (part of the /C1/ order set) is not replaced.

Figure 4–22 shows the automatic idle ordered set generation.

Figure 4–22. Automatic Ordered Set Generation



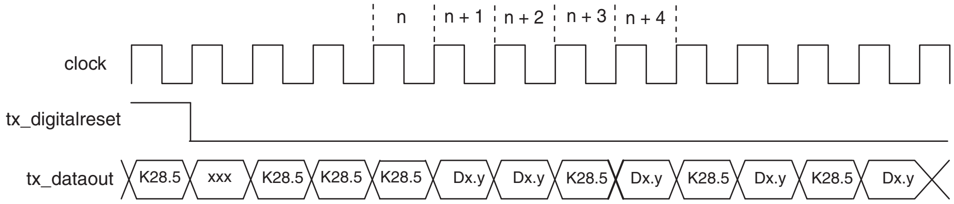
Reset Condition

After assertion of `tx_ready after reset`, the GIGE transmitter automatically transmits three /K28.5/ comma code groups before transmitting user data on the `tx_parallel_data[7:0]` and `tx_datak` port. This could affect the synchronization state machine behavior at the receiver.

Depending on when you start transmitting the synchronization sequence, there could be an even or odd number of /Dx.y/ code groups transmitted between the last of the three automatically sent /K28.5/ code groups and the first /K28.5/ code group of the synchronization sequence. If there is an even number of /Dx.y/ code groups received between these two /K28.5/ code groups, the first /K28.5/ code group of the synchronization sequence begins at an odd code group boundary (`rx_even` = FALSE). An IEEE802.3-compliant GIGE synchronization state machine treats this as an error condition and goes into the loss of sync state.

Figure 4–23 shows an example of even numbers of /Dx.y/ between the last automatically sent /K28.5/ and the first user-sent /K28.5/. The first user-sent /K28.5/ code group received at an odd code group boundary in cycle n + 3 takes the receiver synchronization state machine in the loss of sync state. The first synchronization ordered set /K28.5/Dx.y/ in cycles n + 3 and n + 4 is discounted and three additional ordered sets are required for successful synchronization.

Figure 4–23. Reset Condition in GIGE Mode



Synchronization

The word aligner in GIGE functional mode is configured in automatic synchronization state machine mode. The Quartus II software automatically configures the synchronization state machine to indicate synchronization when the receiver receives three consecutive synchronization ordered sets. A synchronization ordered set is a /K28.5/ code group followed by an odd number of valid /Dx.y/ code groups. The fastest way for the receiver to achieve synchronization is to receive three continuous {/K28.5/, /Dx.y/} ordered sets.

Receiver synchronization is indicated on the `rx_syncstatus` port when the word aligner status port is enabled or on the register bit using the Avalon Memory Mapped Management Interface for each channel. A high on the `rx_syncstatus` port indicates that the lane is synchronized; a low on the `rx_syncstatus` port indicates that the lane has fallen out of synchronization. The receiver loses synchronization when it detects four invalid code groups separated by less than three valid code groups or when it is reset. It would take four consecutive valid code groups in order to reduce the error count by one.

Clock Compensation Using the Rate Match FIFO

In GIGE mode, the rate match FIFO is capable of compensating for up to ±100 PPM (200 PPM total) difference between the upstream transmitter and the local receiver reference clock. The GIGE protocol requires the transmitter to send idle ordered sets /I1/ (/K28.5/D5.6/) and /I2/ (/K28.5/D16.2/) during inter-packet gaps adhering to the rules listed in the IEEE 802.3 specification.

The rate match operation begins after the synchronization state machine in the word aligner indicates synchronization is acquired by driving the `rx_syncstatus` signal high. The rate matcher silently deletes or inserts both symbols (/K28.5/ and /D16.2/) of the /I2/ ordered sets even if it requires deleting only one symbol to prevent the rate match FIFO from overflowing or under-running. It can insert or delete as many /I2/ ordered sets as necessary to perform the rate match operation.

Two flags, `rx_rmfifodatadeleted` and `rx_rmfifodatainserted`, indicating rate match FIFO deletion and insertion events, respectively, are forwarded to the FPGA fabric. Both the `rx_rmfifodatadeleted` and `rx_rmfifodatainserted` flags are asserted for two clock cycles for each deleted and inserted /I2/ ordered set, respectively.

Figure 4–24 shows an example of rate match FIFO deletion where three symbols are required to be deleted. Because the rate match FIFO can only delete /I2/ ordered set, it deletes two /I2/ ordered sets (four symbols deleted).

Figure 4–24. Rate Match Deletion in GIGE Mode

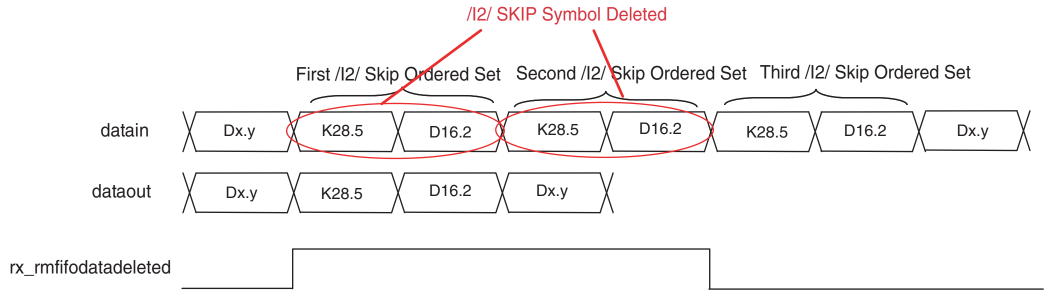
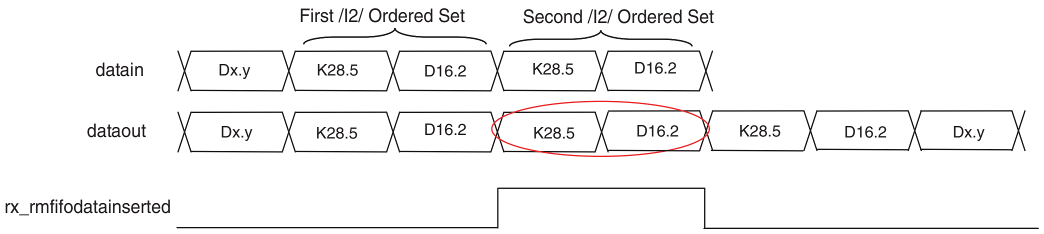


Figure 4–25 shows an example of rate match FIFO insertion in the case where one symbol is required to be inserted. Because the rate match FIFO can only delete /I2/ ordered set, it inserts one /I2/ ordered set (two symbols inserted).

Figure 4–25. Rate Match Insertion in GIGE Mode



Transceiver Clocking and Channel Placement Guidelines

This section describes the transceiver clocking and placement guidelines for the GIGE protocol supported in Stratix V devices.

Transceiver Clocking

Figure 4–26 shows the transceiver clocking with rate matching enabled when configured for GIGE.

Figure 4–26. GIGE Mode Datapath with Rate Matching Enabled

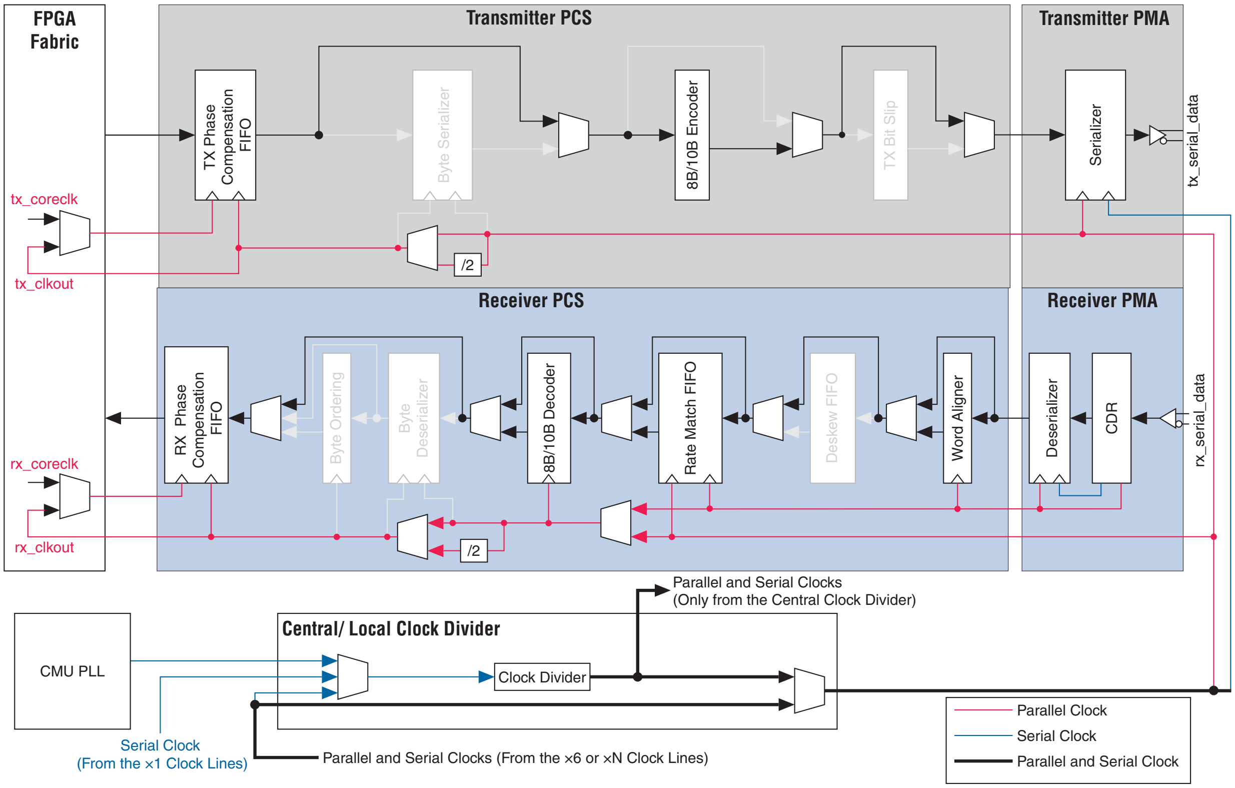


Table 4–7 lists the transceiver datapath clock frequencies in GIGE functional mode.

Table 4–7. Transceiver Datapath Clock Frequencies in GIGE Mode

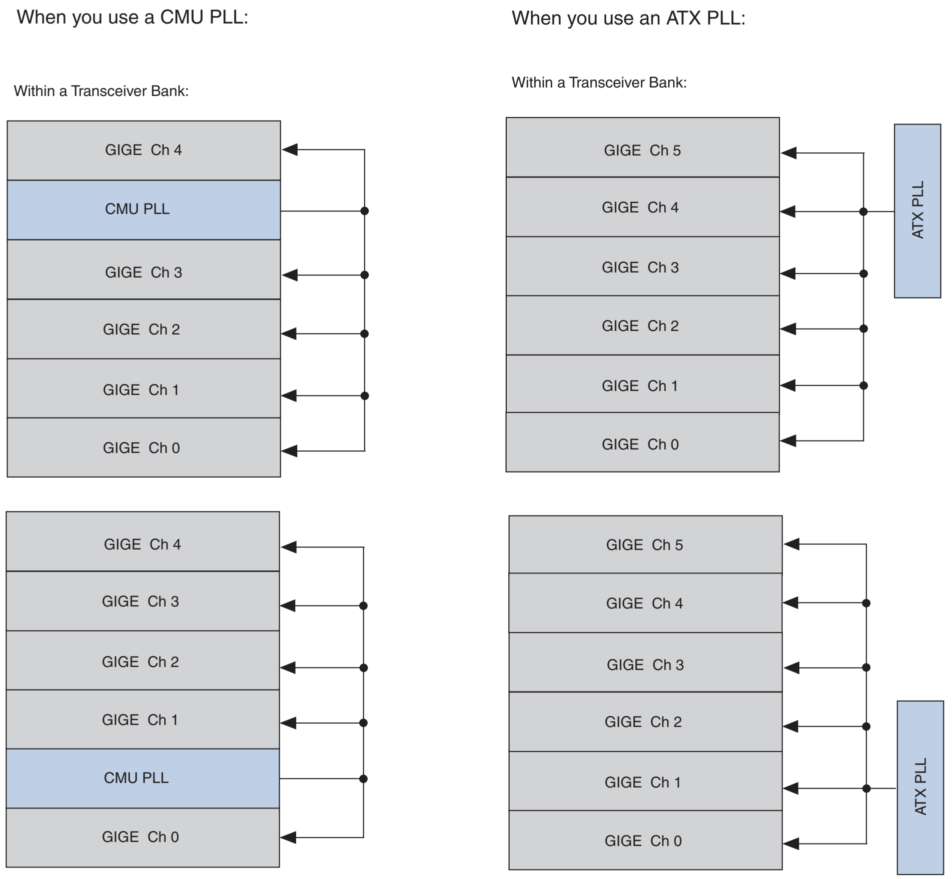
Functional Mode	Line Data Rate	Half-Rate High-Speed Serial Clock Frequency	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Clock Frequency
GIGE	1.25 Gbps	625 MHz	8-bit data, 1-bit control	125 MHz

Transceiver Channel Placement Guidelines

There are no specific placement constraints when placing a GIGE transceiver channels in a transceiver bank. Up to five GIGE channels may be placed on channels 0, 2, 3, 5, and either channel 1 or channel 4 when using the CMU PLL; Up to six GIGE channels may be placed anywhere on one of the six transceiver channels per bank when using the ATX PLL. Figure 4–27 shows the allowed channel placement when using either the CMU PLL or the ATX PLL to drive the GIGE link. If a CMU PLL is implemented, channel 1 or channel 4 in a transceiver bank generates the transmitter serial clock to drive the ×1 clock lines for up to five GIGE channels per transceiver bank. If an ATX PLL is implemented, the ATX PLL generates the transmitter serial clock to drive the ×1 clock lines for up to six GIGE channels per transceiver bank.

To enable the use of the ATX PLL, a minimum base data rate of 2.5 Gbps must be selected in the **Custom PHY IP GIGE Preset** to generate the 1.25 GHz clock for each of the GIGE transceiver channel local clock dividers. In addition, the ATX PLL must also be selected in the **Quartus II Assignment Editor**.

Figure 4–27. Channel Placement Guidelines in a GIGE Configuration



XAUI

This section describes XAUI link implementation using Stratix V transceivers. It provides the transceiver channel datapath description, clocking, and channel placement guidelines when configured in a XAUI configuration.

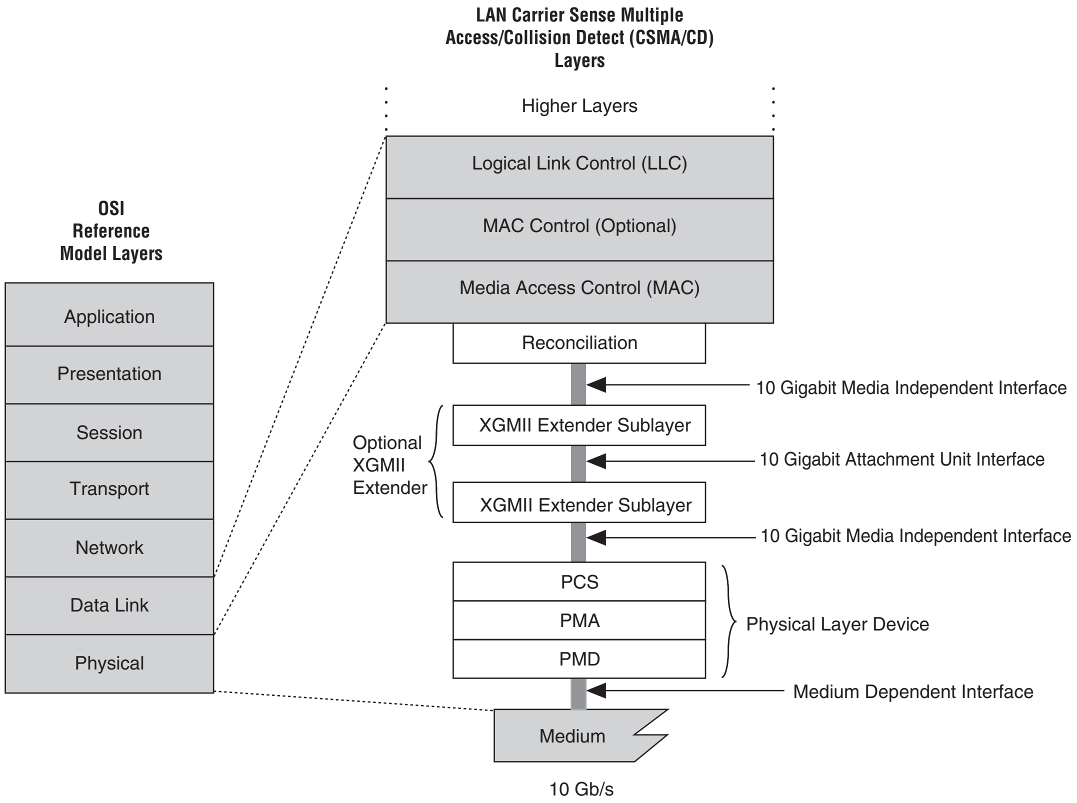
In the MegaWizard Plug-In Manager, you can implement a XAUI link. Under **Ethernet** in the **Interfaces** menu, select the **XAUI PHY** IP core. Currently, the XAUI PHY IP core implements the XAUI PCS in soft logic.

For more information about the XAUI PHY IP core, see the *Altera Transceiver PHY IP Core User Guide*.

XAUI is a specific physical layer implementation of the 10 Gigabit Ethernet link defined in the IEEE 802.3ae-2002 specification. As shown in Figure 4–28, the XAUI PHY uses the XGMII interface to connect to the IEEE802.3 MAC and Reconciliation Sublayer (RS). The IEEE 802.3ae-2002 specification requires the XAUI PHY link to support a 10 Gbps data rate at the XGMII interface and four lanes each at 3.125 Gbps at the PMD interface.

Figure 4–28 shows the relationships between the XAUI PHY and other sublayers in the OSI reference model.

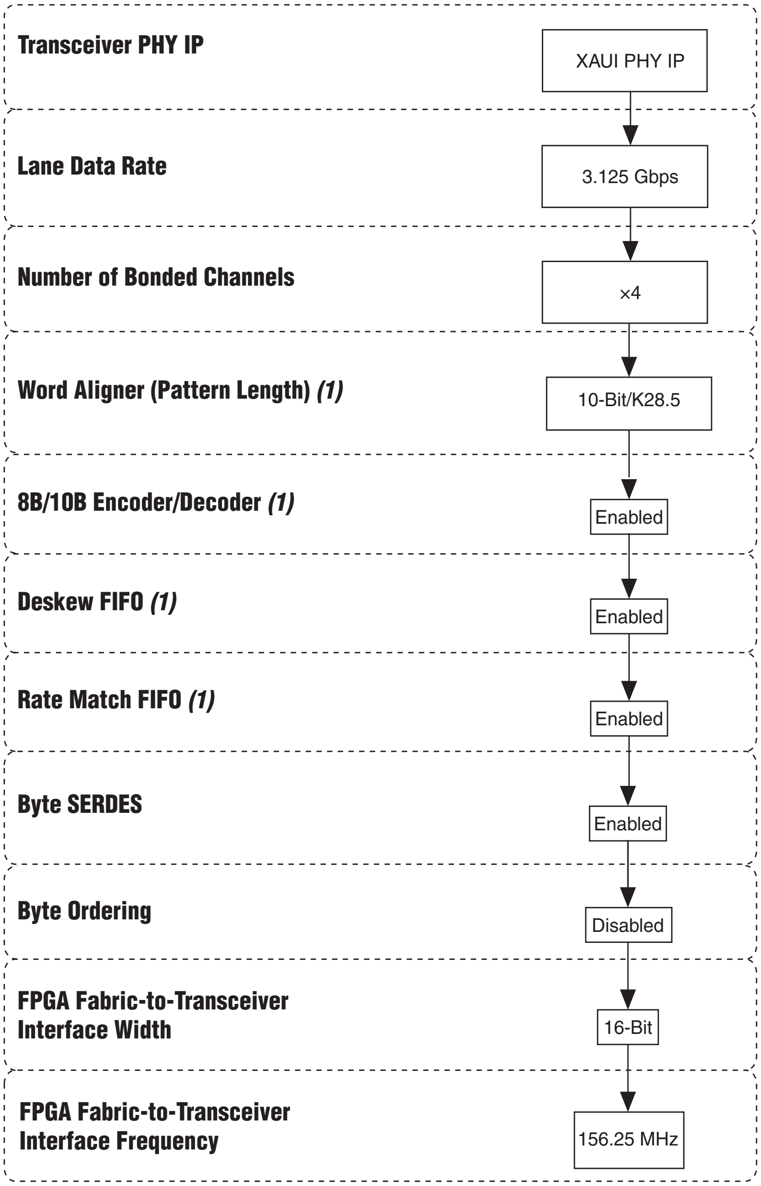
Figure 4–28. XAUI and XGMII Layers


LAN Carrier Sense Multiple Access/Collision Detect (CSMA/CD) Layers
Higher Layers
Logical Link Control (LLC)
MAC Control (Optional)
Media Access Control (MAC)
Reconciliation
10 Gigabit Media Independent Interface
Optional XGMII Extender
XGMII Extender Sublayer
10 Gigabit Attachment Unit Interface
XGMII Extender Sublayer
10 Gigabit Media Independent Interface
PCS
PMA
PMD
Physical Layer Device
Medium Dependent Interface
Medium
10 Gb/s
OSI Reference Model Layers
Application
Presentation
Session
Transport
Network
Data Link
Physical

Transceiver Datapath in a XAUI Configuration

Figure 4–29 shows the transceiver blocks enabled in a XAUI configuration. Currently, the XAUI PCS is implemented in soft logic inside the FPGA core when using the XAUI PHY IP core. In future versions of the Quartus II software, a hard XAUI PCS will be supported. If you plan on migrating to the hard XAUI PCS in the future, you must ensure that your channel placement is compatible between the soft and hard PCS implementations. For placement guidelines, refer to "Transceiver Channel Placement Guidelines" on page 4–44.

Figure 4–29. Stratix V XAUI Configuration

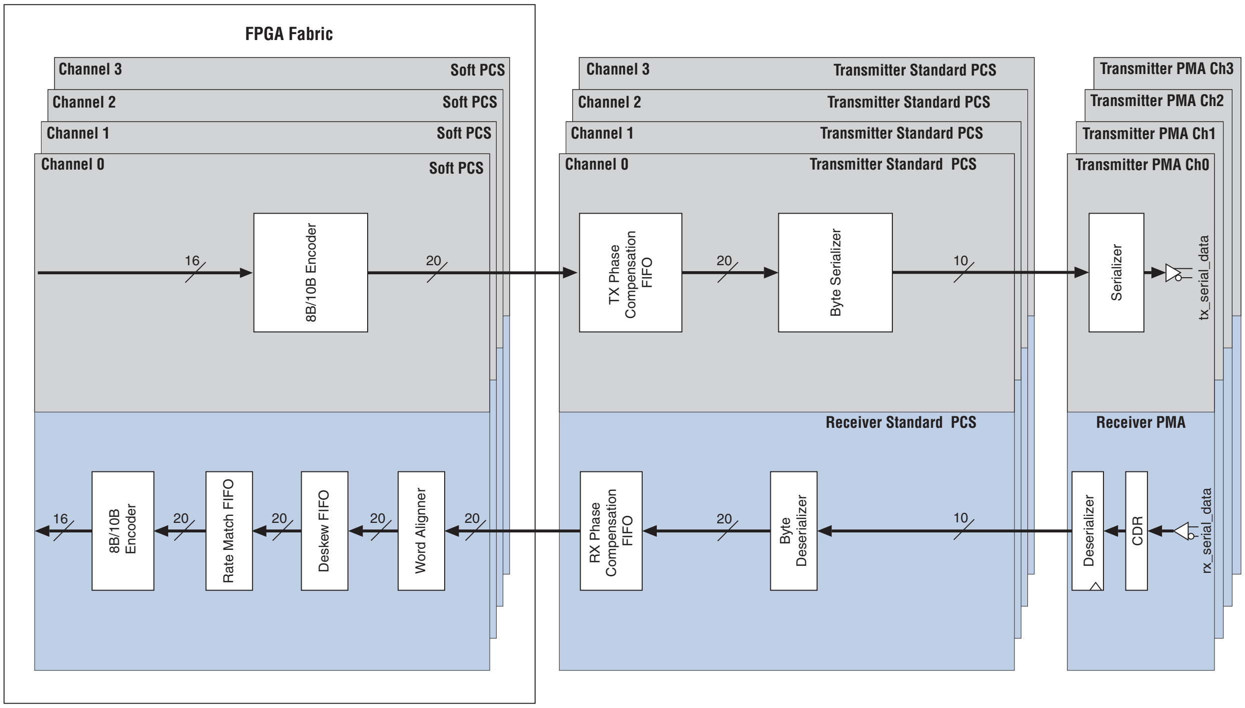


Note to Figure 4–29:

(1) Implemented in soft logic.

Figure 4–30 shows the transceiver datapath in a XAUI configurations.

Figure 4–30. XAUI Configuration Datapath *(Note 1)*



Note to Figure 4–30:

(1) Standard PCS in a low latency configuration is used in this configuration. Additionally, a portion of the PCS is implemented in soft logic.

Supported Features

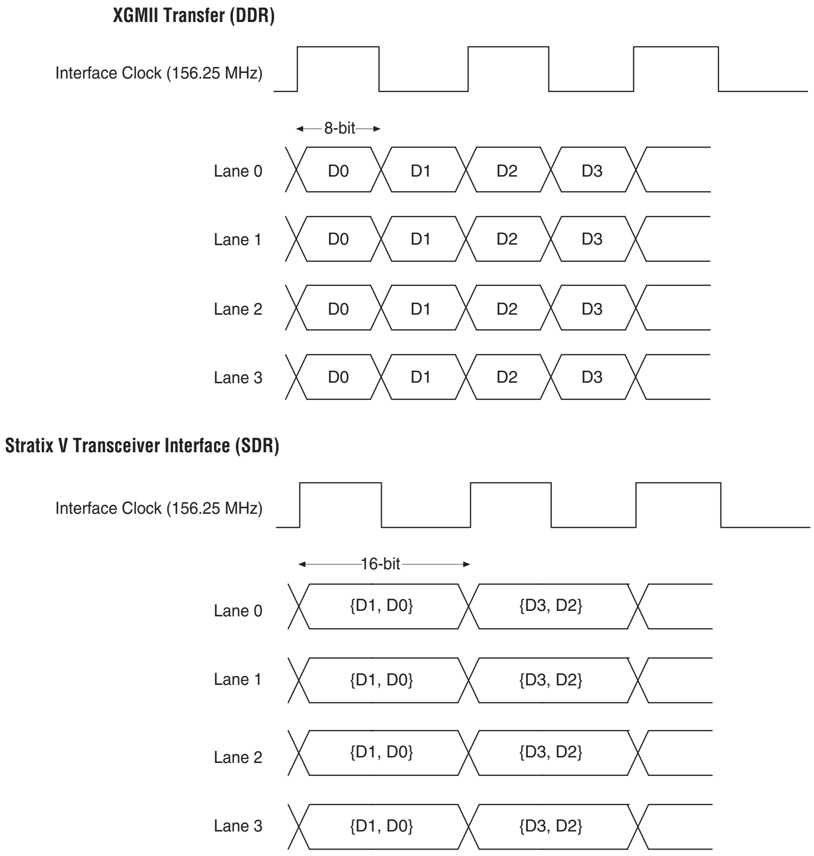
Stratix V transceivers support the following features in a XAUI configuration.

64-Bit SDR Interface to the MAC/RS

Clause 46 of the IEEE 802.3-2008 specification defines the XGMII interface between the XAUI PCS and the Ethernet MAC/RS. It requires each of the four XAUI lanes to transfer 8-bit data and 1-bit wide control code at both the positive and negative edge (DDR) of the 156.25 MHz interface clock.

Stratix V transceivers in a XAUI configuration do not support the XGMII interface to the MAC/RS as defined in IEEE 802.3-2008 specification. Instead, they allow the transferring of 16-bit data and 2-bit control code on each of the four XAUI lanes, only at the positive edge (SDR) of the 156.25 MHz interface clock, as shown in Figure 4–31.

Figure 4–31. Implementation of the XGMII Specification in Stratix V Devices



8B/10B Encoding/Decoding

Each of the four lanes in a XAUI configuration support an independent 8B/10B encoder/decoder as specified in Clause 48 of the IEEE802.3-2008 specification. 8B/10B encoding limits the maximum number of consecutive 1s and 0s in the serial data stream to five, thereby ensuring DC balance as well as enough transitions for the receiver CDR to maintain a lock to the incoming data.

The XAUI PHY IP core provides status signals to indicate running disparity as well as the 8B/10B code group error.

Transmitter and Receiver State Machines

In a XAUI configuration, the Stratix V transceivers implement the transmitter and receiver state diagrams shown in Figure 48-6 and Figure 48-9 of the IEEE802.3-2008 specification.

In additions to encoding the XGMII data to PCS code groups, per the rules of the 10GBASE-X PCS, the transmitter state diagram performs functions such as converting Idle ||I|| ordered sets into Sync ||K||, Align ||A||, and Skip ||R|| ordered sets.

In addition to decoding the PCS code groups to XGMII data, per the rules of the 10GBASE-X PCS, the receive state diagram performs functions such as converting Sync ||K||, Align ||A||, and Skip ||R|| ordered sets to Idle ||I|| ordered sets.

Synchronization

The word aligner block in the receiver PCS of each of the four XAUI lanes implements the receiver synchronization state diagram shown in Figure 48-7 of the IEEE802.3-2008 specification.

The XAUI PHY IP core provides a status signal per lane to indicate if the word aligner is synchronized to a valid word boundary.

Deskew

The channel aligner block in the receiver PCS implements the receiver deskew state diagram shown in Figure 48-8 of the IEEE 802.3-2008 specification.

The channel aligner starts the deskew process only after the word aligner block in each of the four XAUI lanes indicates successful synchronization to a valid word boundary.

The XAUI PHY IP core provides a status signal to indicate successful lane deskew in the receiver PCS.

Clock Compensation

The rate match FIFO in the receiver PCS datapath is designed to compensate up to ±100 PPM difference between the remote transmitter and the local receiver. It does so by inserting and deleting Skip ||R|| columns depending on the PPM difference.

The clock compensation operation begins after:

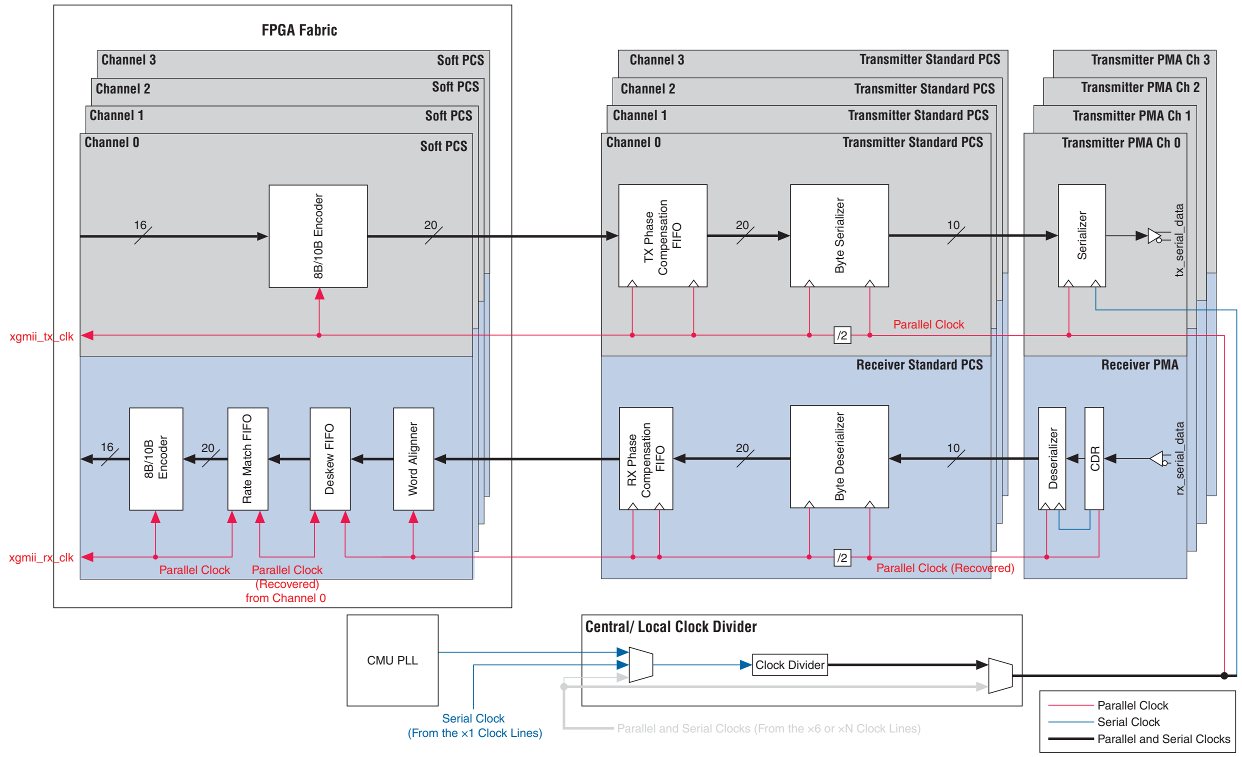
- The word aligner in all four XAUI lanes indicates successful synchronization to a valid word boundary.
- The channel aligner indicates a successful lane deskew.

The rate match FIFO provides status signals to indicate the insertion and deletion of the Skip ||R|| column for clock rate compensation.

Transceiver Clocking

Figure 4–32 shows transceiver clocking in a XAUI configuration.

Figure 4–32. XAUI Clocking Diagram



One of the two channel PLLs in a transceiver bank generates the transmitter serial and parallel clocks for the four XAUI channels. The ×6 clock line carries the transmitter clocks to the PMA and PCS of each of the four channels.

Table 4–8 lists the input reference clock frequency, FPGA fabric-transceiver interface width, and interface frequency supported in a XAUI configuration.

Table 4–8. Input Reference Clock Frequency and Interface Speed Specifications for XAUI Configurations

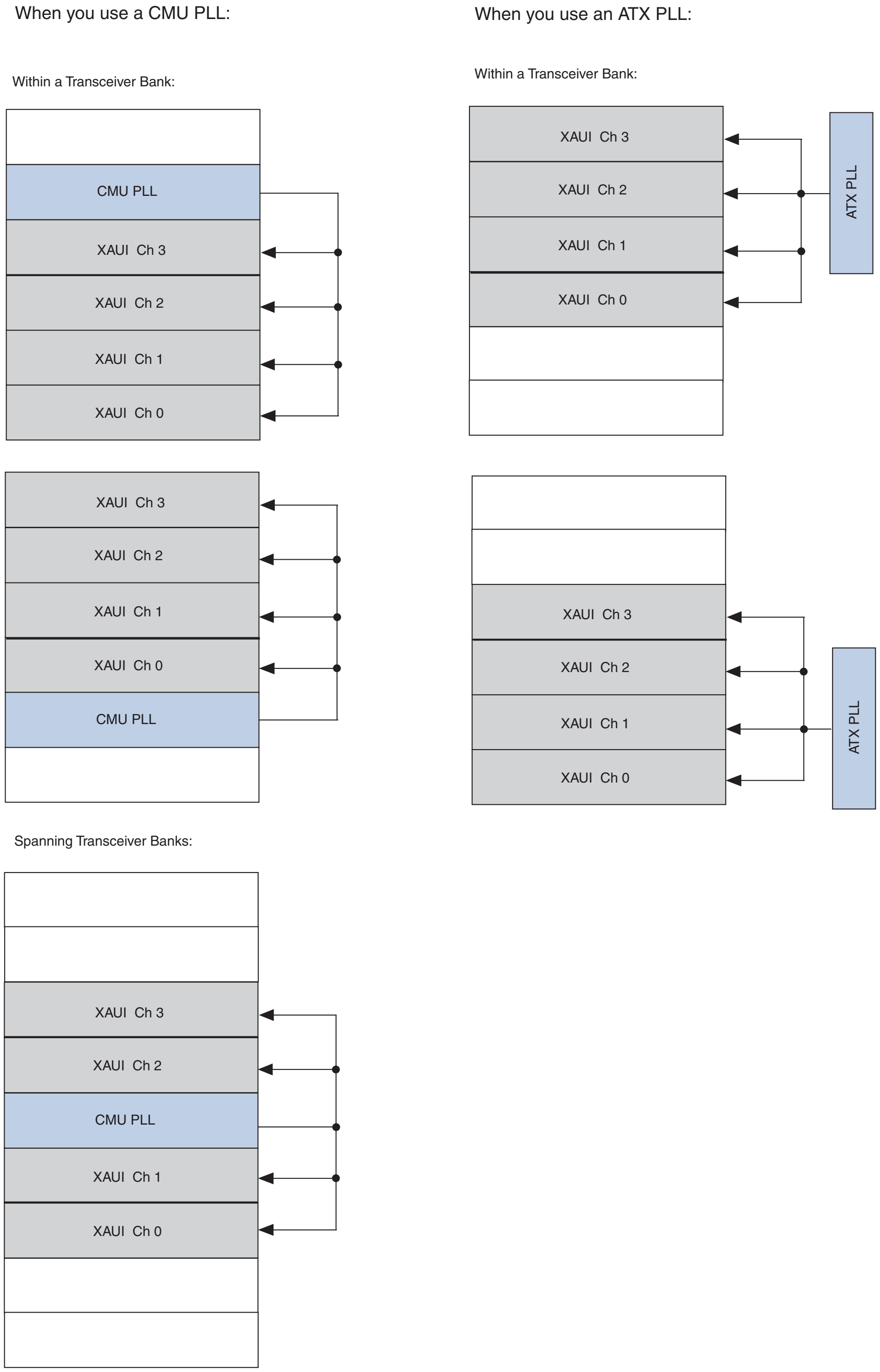
Input Reference Clock Frequency (MHz)	FPGA Fabric-Transceiver Interface Width	FPGA Fabric-Transceiver Interface Width (MHz)
156.25	16-bit data, 2-bit control	156.25

Transceiver Channel Placement Guidelines

Although the soft PCS implementation of the XAUI configuration has no placement restrictions, if you plan on migrating to the hard PCS version in the future, you must place the channels according to the guidelines that follow.

Figure 4–33 shows the allowed channel placement when using either the CMU PLL or the ATX PLL to drive the XAUI link. This placement only applies when using the hard PCS block in the transceiver. The current version of the Quartus II software (11.0) implements the XAUI PCS in soft logic and therefore does not have any placement restrictions. If you plan to use the XAUI hard PCS in future versions of the Quartus II software, enure that your channels are placed in one of the configurations shown in Figure 4–33.

Figure 4–33. Channel Placement Guidelines in a XAUI Configuration



Document Revision History

Table 4–9 lists the revision history for this chapter.

Table 4–9. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Updated Figure 4–8 and Figure 4–9. ■ Updated "Supported Features" on page 4–10. ■ Updated Table 4–5. ■ Updated Figure 4–16, Figure 4–18, and Figure 4–19. ■ Added "GIGE" section. ■ Updated "XAUI" on page 4–37. ■ Updated "Transceiver Datapath in a XAUI Configuration" on page 4–39. ■ Updated "Transceiver Channel Placement Guidelines" on page 4–44. ■ Updated Figure 4–33. ■ Chapter moved to volume 3 for the 11.0 release.
December 2010	1.1	■ Updated "PCI Express (PIPE) 2.0 Interface", "Dynamic Switching Between Gen1 (2.5 Gbps) and Gen2 (5 Gbps) Signal Rates", "Receiver Status", and "Receiver Detection" sections. ■ Updated Figure 4–32.
April 2010	1.0	Initial release.

5. Transceiver Custom Configurations in Stratix V Devices

SV52006-1.2

This chapter describes the custom transceiver configuration datapath in Stratix® V devices for the 10G and standard physical coding sublayer (PCS) blocks.

This chapter contains the following sections:

- "10G Low Latency Configuration"
- "Standard PCS Custom and Low Latency Configurations" on page 5–8

10G Low Latency Configuration

A Low Latency PHY IP core using the 10G PCS is available for 32-bit, 40-bit, 50-bit, 64-bit, or 66-bit PCS data width configurations. Figure 5–1 shows the 10G Low Latency configuration datapath.

Figure 5–1. 10G Low Latency Datapath

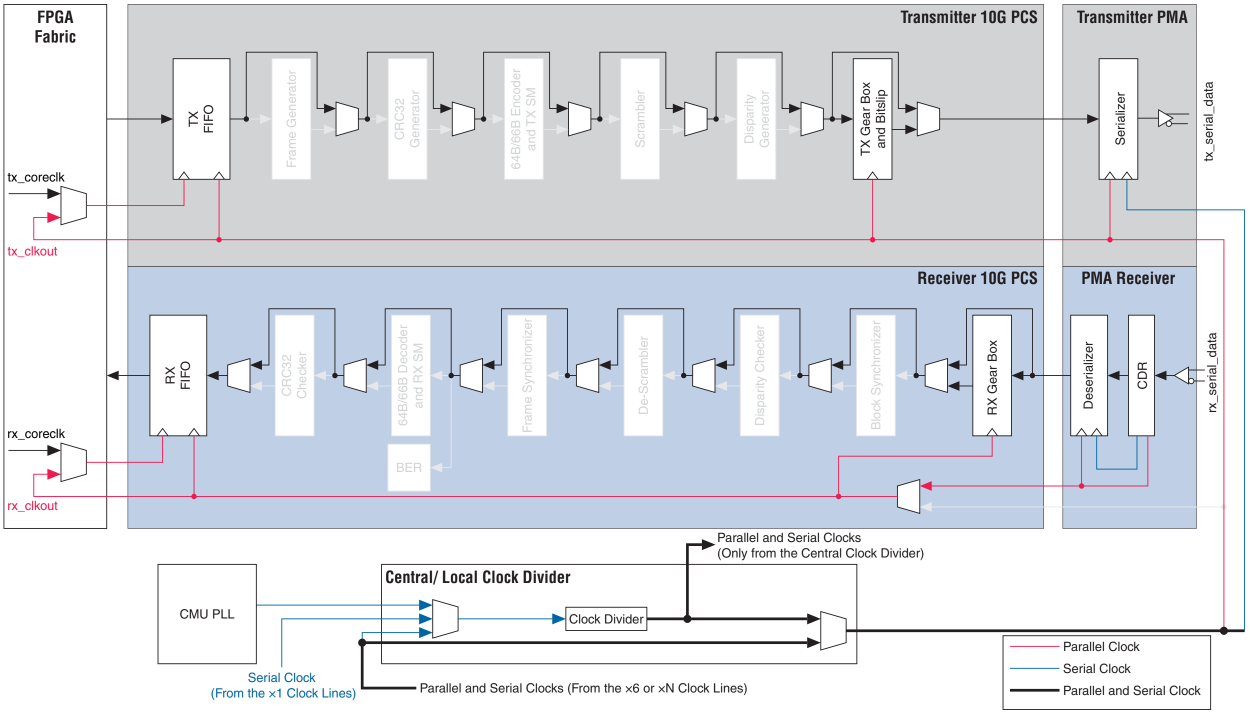


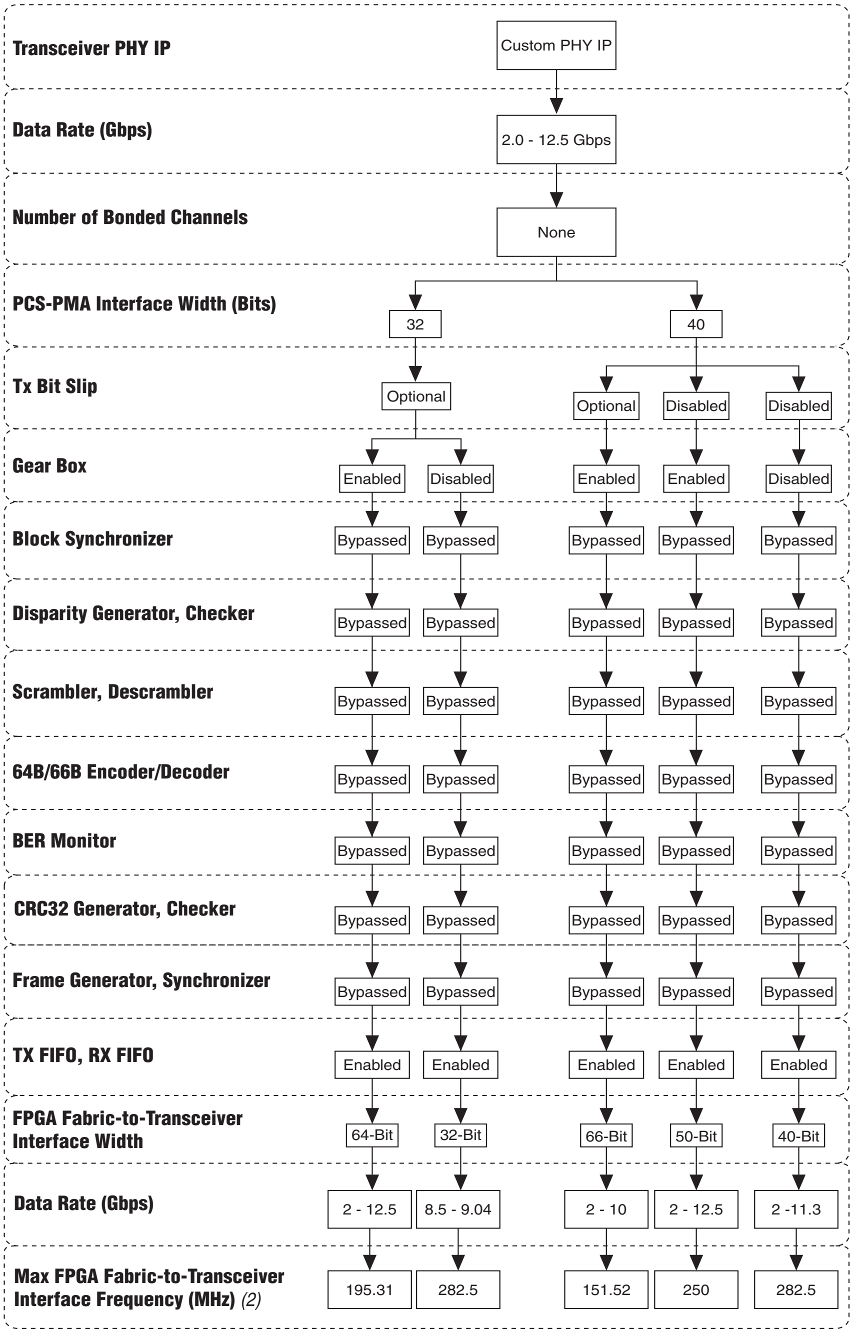
Figure 5–2 shows the different options available in the 10G Low Latency configuration. To select the custom transceiver configuration, use the Low Latency PHY IP core. This IP core supports custom transceiver configurations that use the standard PCS. You can enable this option in the IP core by enabling the **select 10G PCS** option in the MegaWizard™ Plug-In Manager.

For more information about the Low Latency PHY IP core, refer to the *Low Latency PHY IP Core* chapter in the *Altera Transceiver PHY IP Core User Guide*.

The blocks shown as “Disabled” are not used, but incur latency. The blocks shown as “Bypassed” are not used and do not incur any latency. The FPGA fabric-to-transceiver interface frequency specified in Figure 5–2 is for maximum speed grade devices.

For the limits for all speed grades, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–2. Options for 10G Low Latency Configuration ***(Note 1)***



Notes to Figure 5–2:

(1) For the limits of all speed grades, refer to the *"Transceiver Specifications for Stratix V GX and GS Devices"* table in the *DC and Switching Characteristics for Stratix V Devices* chapter.

(2) You must generate an `rx_coreclk` with the specified frequency whenever the gear box is enabled.

The Quartus® II software only supports the non-bonded configuration (×1) when the 10G PCS is enabled. This implies that if you create multiple channels with the 10G PCS, a common low-speed parallel clock (used in the bonded channel configuration, such as XAUI) is not generated by the central clock divider block. Each transmitter channel takes the high-speed clock, generated by the channel PLL, and locally divides it to generate the parallel clock.

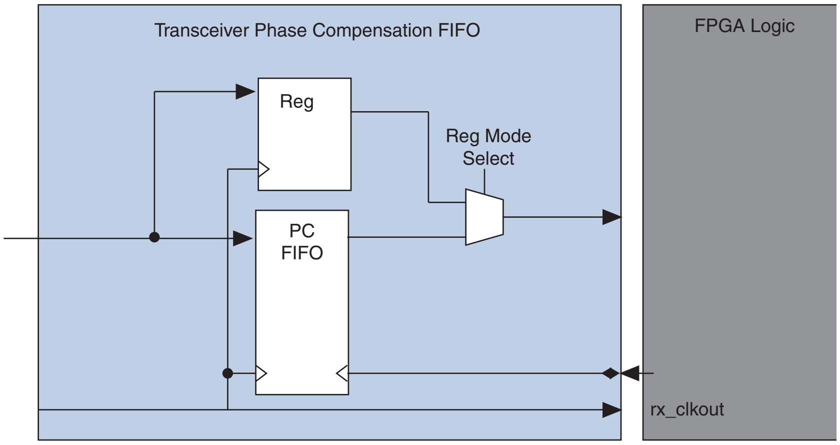
Datapath Functionality

This section discusses the different PCS blocks that you can use in the 10G Low Latency transceiver configuration.

Transmitter and Receiver FIFO

The FIFOs can be configured in phase compensation or registered mode, as shown in Figure 5–3. In phase compensation mode, the FIFO compensates the phase differences in the clock between the read and write side of the FIFO. The clocking scheme for the write side of the transmitter (TX) and receiver (RX) FIFOs depends on whether the gear box is enabled, and its ratio (32:64, 40:50, or 40:66). The clocking scheme is described in the "Clocking" section.

Figure 5–3. Phase Compensation FIFO in RX Path



Gear Box

The gear box translates the datapath width differences between the PCS and the physical medium attachment (PMA) interfaces. The gear box contains handshake control logic and FIFOs to implement the data-width translation. For the supported gear box ratio, refer to Figure 5–2.

TX Bit Slip Feature

The bit slip feature supported in custom configurations enables you to slip the transmit side bits before they are sent to the gear box. The number of bits slipped is equal to the FPGA fabric-to-transceiver interface width of 1. For example, if the FPGA fabric-to-transceiver interface width is 64 bits, a maximum of 63 bits can be slipped. That is, `bits[63]` from the first word and `bits[62:0]` are concatenated to form a 64 bit word (`bit[62:0]` from the second word, `bit[63]` from the first word LSB). The 7-bit input control signal is available to the FPGA fabric. For a 63-bit shift mentioned above, set the value of the input control to 7'b0011111.

Clocking

This section describes the transceiver datapath clocking. Figure 5–4 shows the clocking scheme when the gear box is not enabled. There is no frequency difference between the read and write side of the TX and RX FIFO clocks because the gear box is not enabled. The Quartus II software automatically connects the clocks to the read and write side of the TX FIFO and RX FIFO. In this configuration, the data from the TX FIFO is directly fed to the serializer.

For more information about the channel PLL, refer to *"Channel PLL Architecture"* in the *Transceiver Architecture of Stratix V Devices* chapter.

Figure 5–4. 10G PCS Low Latency Datapath with Gear Box Not Enabled

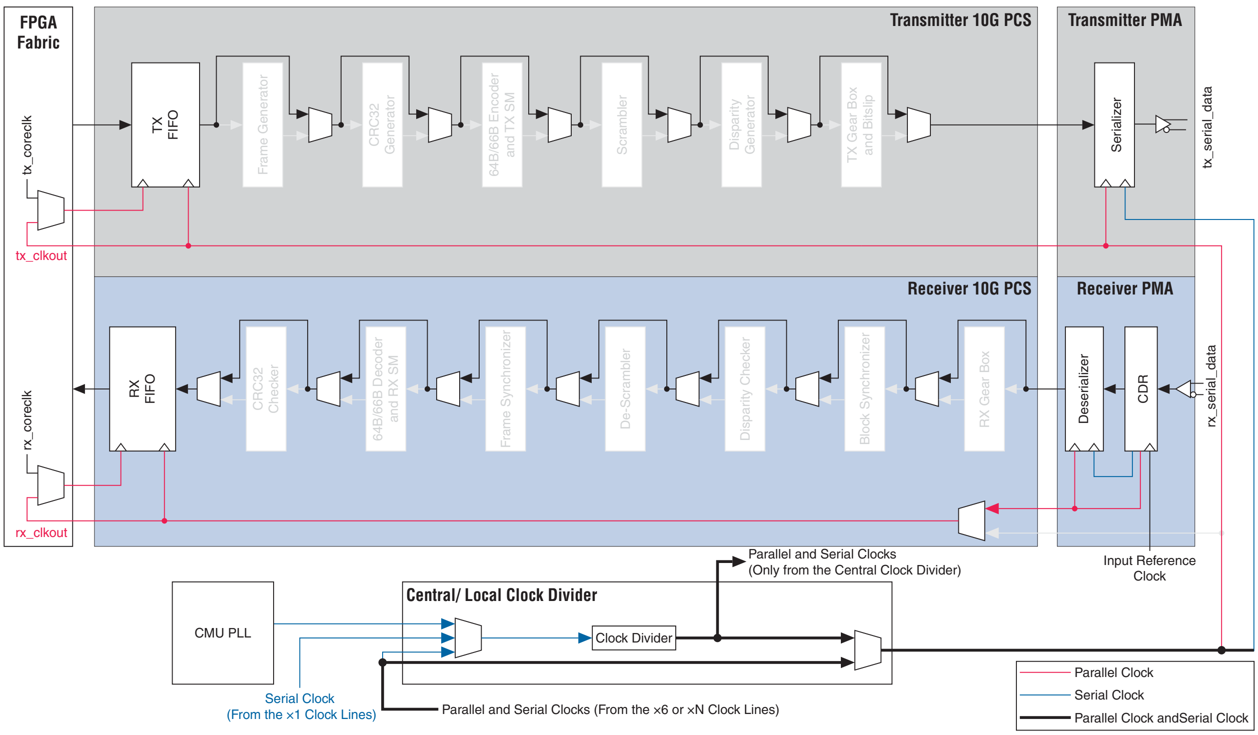


Figure 5–5 shows the configuration where the gear box ratio is 32:64. The FPGA fabric interface width (64 bits) is exactly twice the internal transceiver datapath width. You can divide the `tx_clkout` and `rx_clkout` in the FPGA fabric by two, and use them to clock the write side of TX FIFO and the read side of RX FIFO, respectively. Select the `tx_coreclk` and the `rx_coreclk` ports in the Low Latency PHY IP core and connect the divided clock to these ports, as shown in Figure 5–5.

Figure 5–5. 10G PCS Low Latency Datapath with the Gear Box Ratio of 32:64

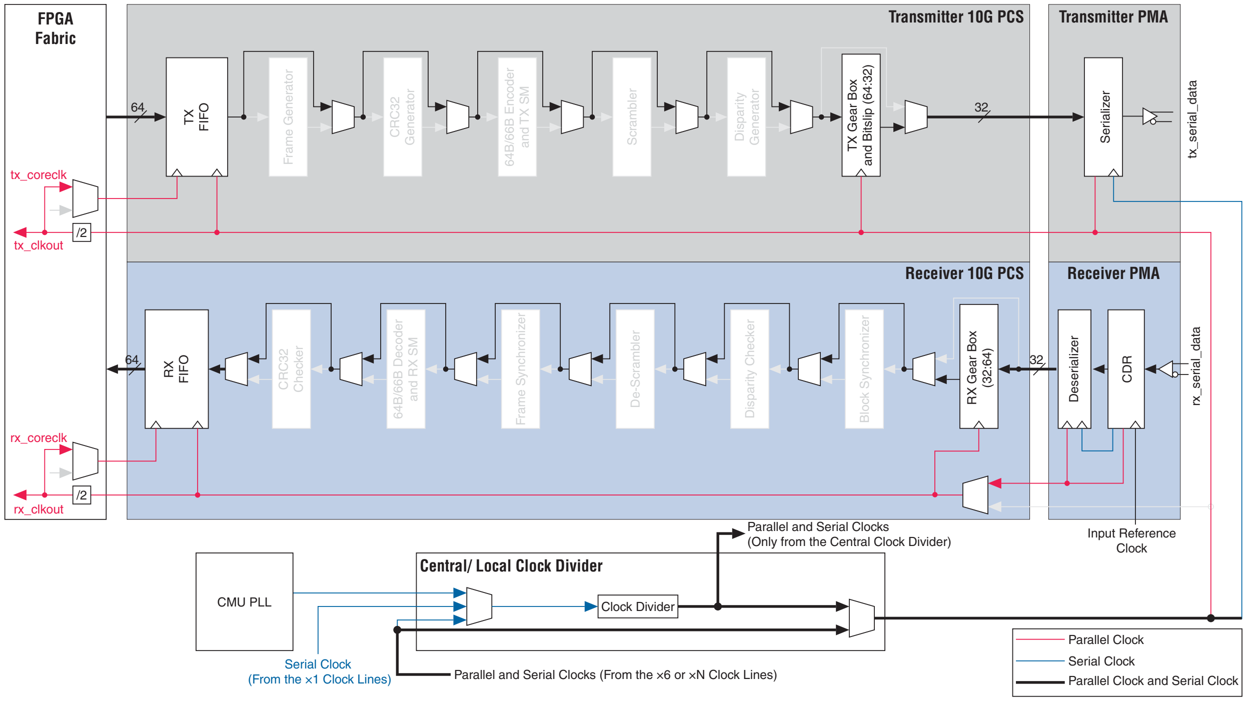
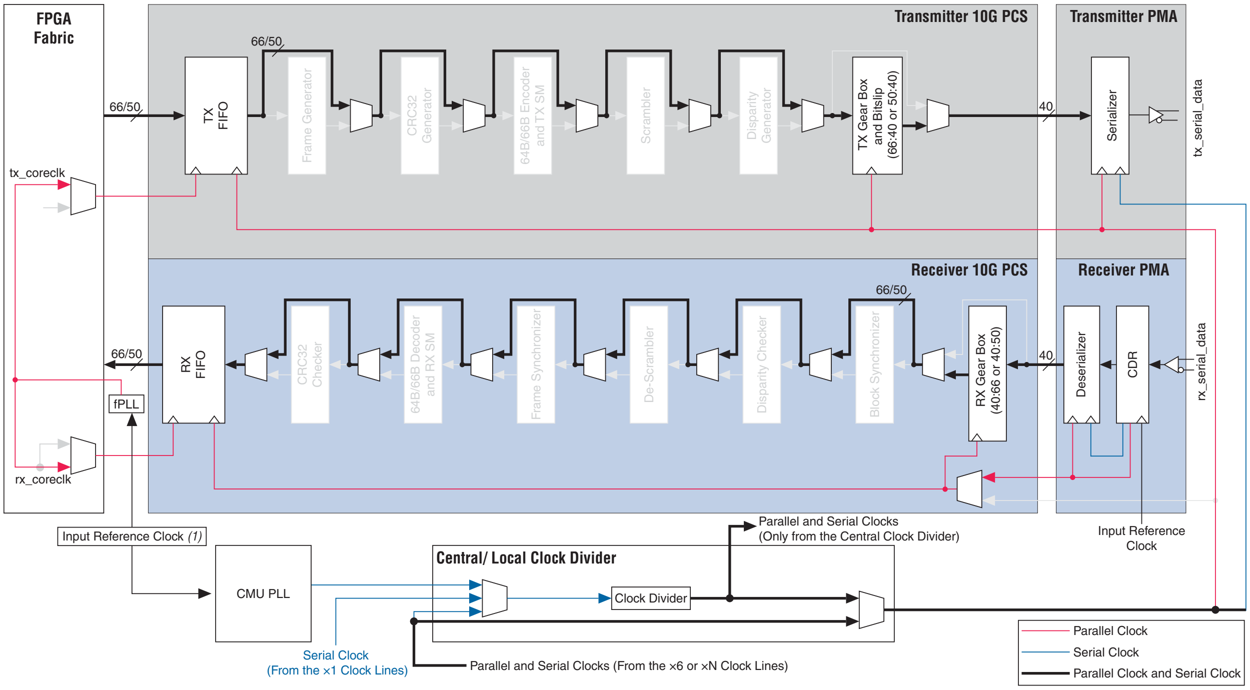


Figure 5–6 shows the clocking scheme when the gear box ratio (40:66 or 40:50) is not an integral multiple of the FPGA fabric interface width. You must use a fractional PLL to provide the appropriate clock frequency to the write side of the TX FIFO. Set the division factor in the fractional PLL so that its output frequency is equal to the transmitter data rate divided by 66 or 50. The clock source that provides the input reference clock to the fractional PLL and the CMU PLL must be the same because the TX FIFO operates as a phase compensation FIFO; therefore, the clock requires a 0 PPM between the read and write sides.

For the receive side, enable the `rx_coreclk` port and connect the fractional PLL output to the `rx_coreclk` port. The RX FIFO operates as a phase compensation FIFO. Therefore, the read and write side of the RX FIFO must have a 0 PPM difference. This requires that the receive side and the upstream transmitter are clocked by the same clock source (synchronous systems).

Figure 5–6. 10G PCS Low Latency Datapath with the Gear Box Ratio of 40:66 or 40:50



Note to Figure 5–6:

(1) The clock source that provides the input reference clock to the fractional PLL (fPLL in Figure 5–6) and the CMU PLL (the CMU PLL generates the high-speed clock for the serializer) must be the same. The transmitter and the receiver FIFOs can only compensate for phase differences. Therefore, the same clock source ensures 0 PPM between the read and write clocks of the FIFOs.

Using coreclks

The `tx_coreclk` and `rx_coreclk` ports offer the flexibility to use the `tx_clkout` and `rx_clkout` from one channel to clock the TX and RX FIFOs multiple channels.

For more information about the core clocking scheme, refer to the following sections in the *Transceiver Clocking for Stratix V Devices* chapter:

- "User-Selected Transmitter Datapath Interface Clock"
- "User-Selected Receiver Datapath Interface Clock"

Merging Instances

You can merge transmitter and receiver instances with the different 10G PCS datapath configurations in the same 10 Gbps physical channel. For example, the Quartus II software allows you to create the two following instances and place them in the same physical transceiver channel:

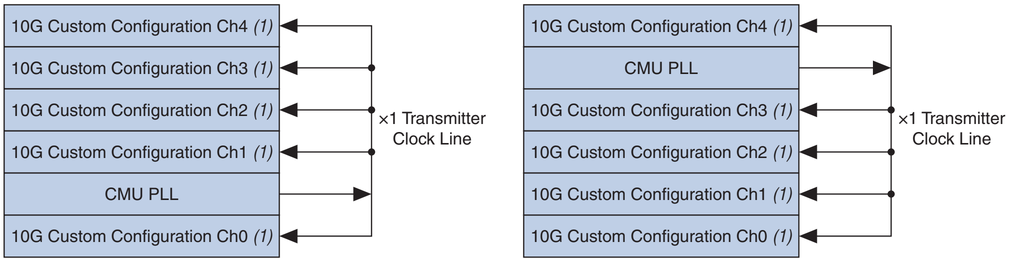
- Transmitter only instance with a 40-bit FPGA fabric interface
- Receiver only instance with a 64-bit FPGA fabric interface

You cannot merge a transmitter instance and receiver instance (1 channel instance) using different PCS blocks (10G PCS and standard PCS) within the same physical transceiver channel.

Transceiver Channel Placement Guidelines

Stratix V devices allow the placement of up to five channels in 10G custom configurations (same data rate) within the same transceiver bank. Figure 5–7 shows a supported channel placement scenario.

Figure 5–7. Channel Placement Guidelines in 10G Custom Configurations


10G Custom Configuration Ch4 (1)
10G Custom Configuration Ch3 (1)
10G Custom Configuration Ch2 (1)
10G Custom Configuration Ch1 (1)
CMU PLL
10G Custom Configuration Ch0 (1)
×1 Transmitter Clock Line
10G Custom Configuration Ch4 (1)
CMU PLL
10G Custom Configuration Ch3 (1)
10G Custom Configuration Ch2 (1)
10G Custom Configuration Ch1 (1)
10G Custom Configuration Ch0 (1)
×1 Transmitter Clock Line

Note to Figure 5–7:

(1) All channels shown in Figure 5–7 are assumed to contain a transmitter and receiver.

Standard PCS Custom and Low Latency Configurations

With custom configurations using the standard PCS, you can select which blocks to use and the data width by creating user-defined configurations. You can also implement protocols such as SONET, Fibre Channel, or SerialLite II by customizing the transceiver PCS configuration. Low latency configurations bypass much of the standard PCS, allowing for more design control in the FPGA fabric. This section discusses how to use the custom and Low Latency PHY IP core with the standard PCS.

Table 5–1 shows the datapath latency for the standard PCS transmitter channel.

Table 5–1. Transmitter Standard PCS Datapath Latency *(Note 1)*

Block	Normal Latency	Low Latency
TX Phase Compensation FIFO *(3)*	4–5	3–5
Byte Serializer	1–2	0–2 *(3)*

Notes to Table 5–1:

(1) These numbers are preliminary.

(2) The TX Phase Compensation FIFO can be configured as register mode. The implementation is the same as TX FIFO in 10G PCS.

(3) This value is dependent on whether the block is enabled or disabled.

Table 5–2 shows the datapath latency for the standard PCS receiver channel.

Table 5–2. Receiver Standard PCS Datapath Latency *(Note 1)*

Block	Normal Latency	Low Latency
Word Aligner	3–7 *(2)*	1
Byte Serializer	1–2	1
Byte Ordering	1–3	0
RX Phase Compensation FIFO	3–4	2–3

Notes to Table 5–2:

(1) These numbers are preliminary.

(2) This value is dependent on the configuration mode.

To use this chapter successfully, the *Custom PHY IP Core* and *Low Latency PHY IP Core* chapters in the *Altera Transceiver PHY IP Core User Guide* should be understood and used as references.

Custom Configurations with the Standard PCS

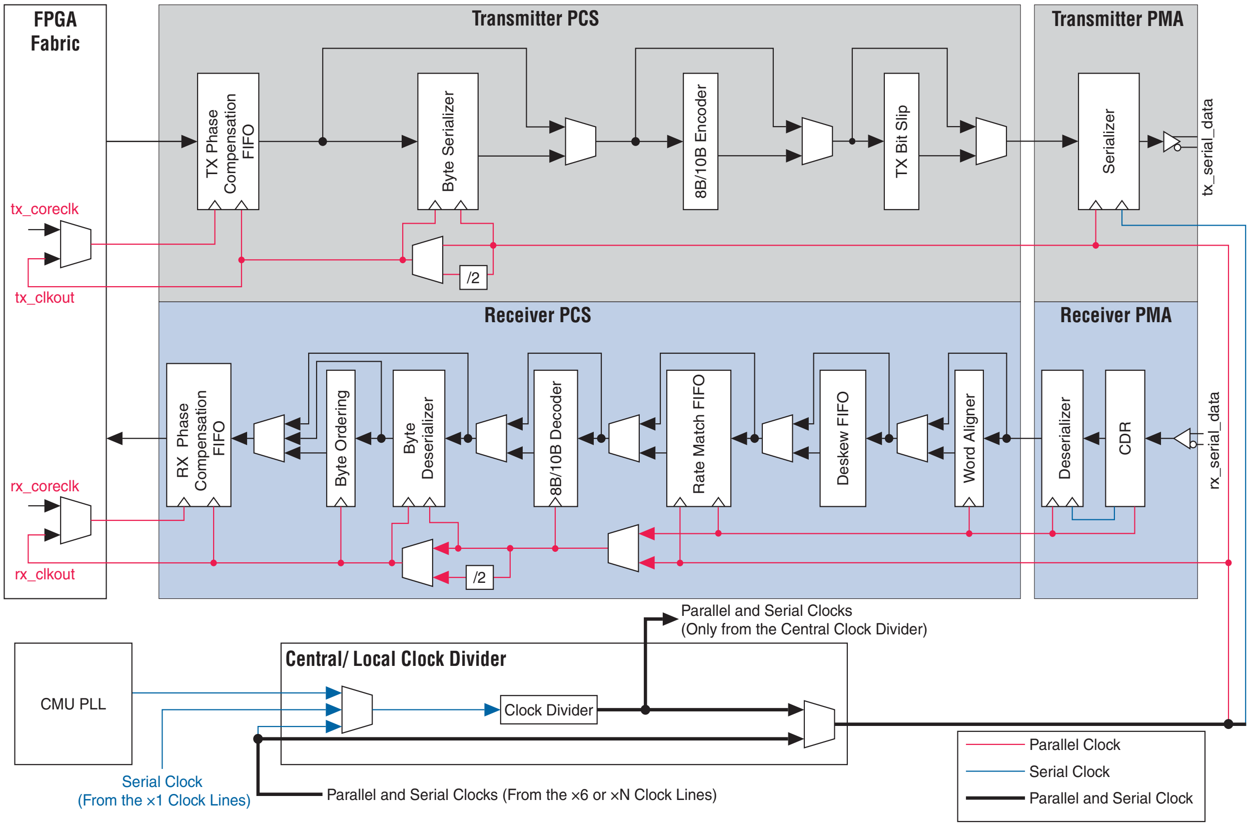
You can configure the custom PHY IP core in 8-bit or 10-bit width mode for low speed, or in 16-bit or 20-bit width mode for higher data rates. Table 5–3 lists the supported data rates.

Table 5–3. PCS-PMA Interface Widths and Data Rates in Custom Single-Width and Double-Width Modes for Stratix V Devices

PCS-PMA Interface Width	Supported Data Rate Range PMA
Custom 8- or 10-bit width	600 Mbps to 3.75 Gbps
Custom 16-bit width	1 Gbps to 8 Gbps
Custom 20-bit width	1 Gbps to 8.5 Gbps

Figure 5–8 shows the complete datapath for custom transceiver configurations with the standard PCS. The options available at different speeds are shown in Figure 5–9 through Figure 5–12.

Figure 5–8. Datapath for Custom Configurations with the Standard PCS



The standard PCS datapath offers flexibility by allowing you to modify, enable, or disable blocks based on your requirements.

Figure 5–9 shows the available options for the standard PCS custom 8-bit PMA-PCS interface width. The maximum frequencies shown in Figure 5–9 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–9. Standard PCS Custom 8-Bit PMA-PCS Interface Width

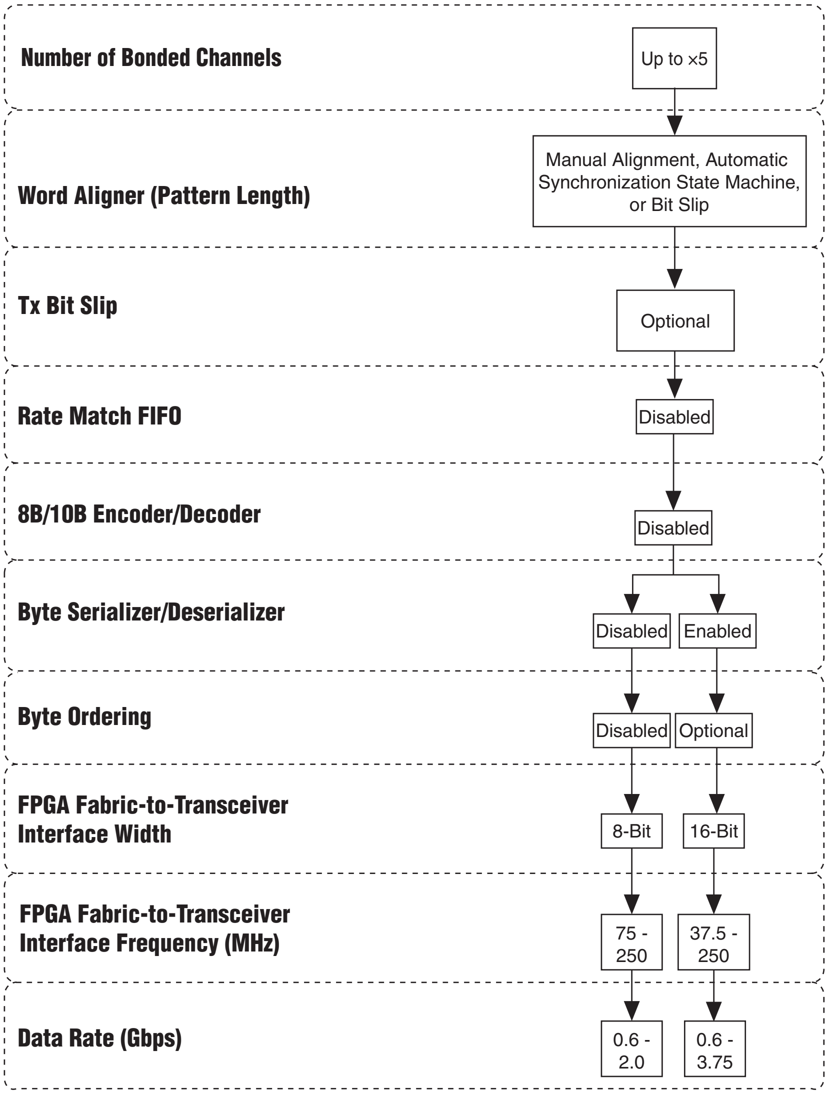

Number of Bonded Channels
Up to ×5
Word Aligner (Pattern Length)
Manual Alignment, Automatic Synchronization State Machine, or Bit Slip
Tx Bit Slip
Optional
Rate Match FIFO
Disabled
8B/10B Encoder/Decoder
Disabled
Byte Serializer/Deserializer
Disabled
Enabled
Byte Ordering
Disabled
Optional
FPGA Fabric-to-Transceiver Interface Width
8-Bit
16-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
75 - 250
37.5 - 250
Data Rate (Gbps)
0.6 - 2.0
0.6 - 3.75

Figure 5–10 shows the available options for the standard PCS custom 10-bit PMA-PCS interface width. The maximum frequencies shown in Figure 5–10 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–10. Standard PCS Custom 10-Bit PMA-PCS Interface Width

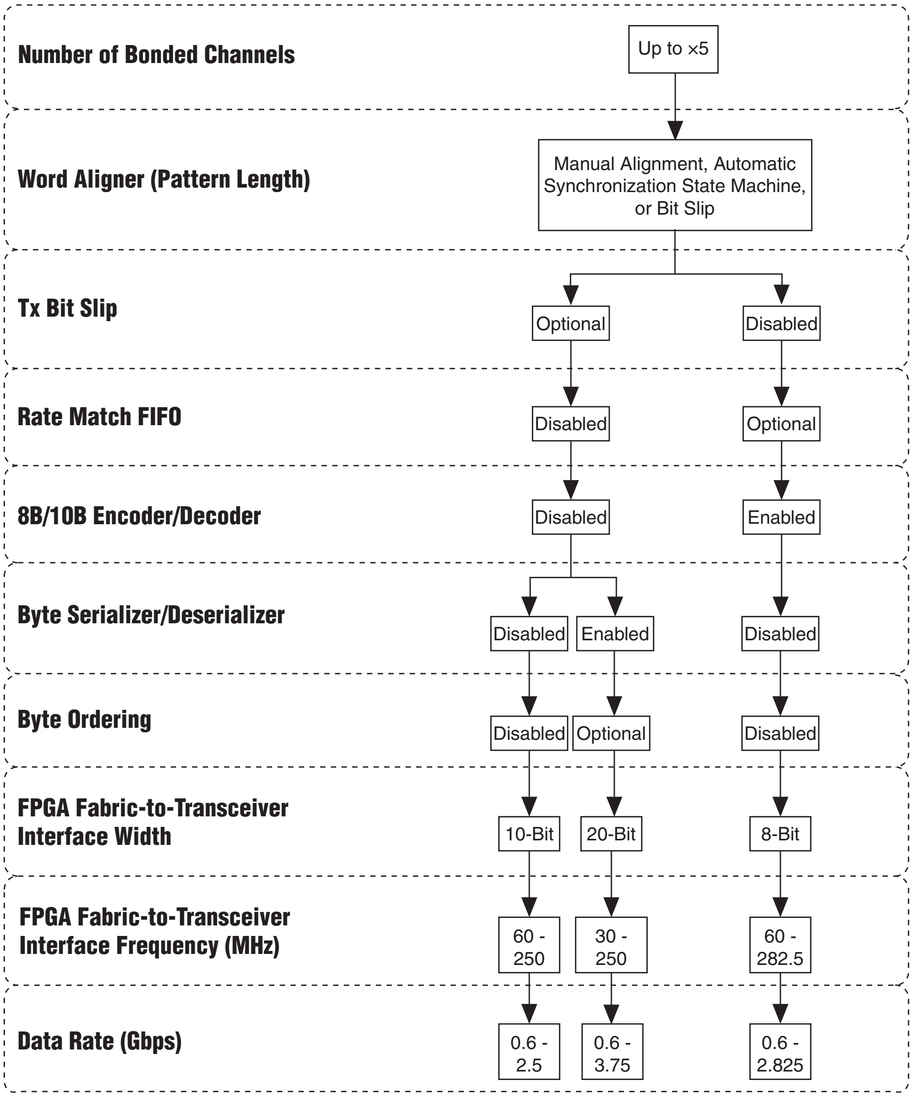


Figure 5–11 shows the available options for the standard PCS custom 16-bit PMA-PCS interface width. The maximum frequencies shown in Figure 5–11 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–11. Standard PCS Custom 16-Bit PMA-PCS Interface Width

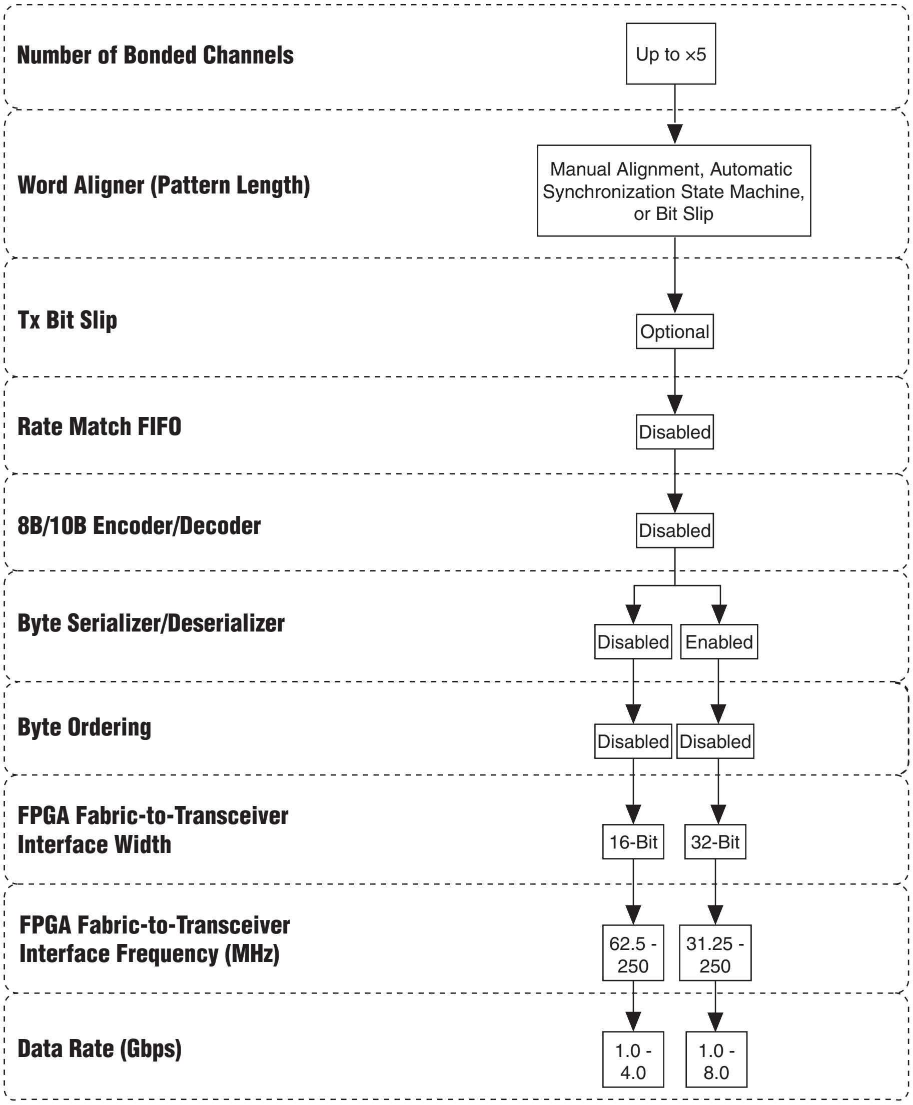
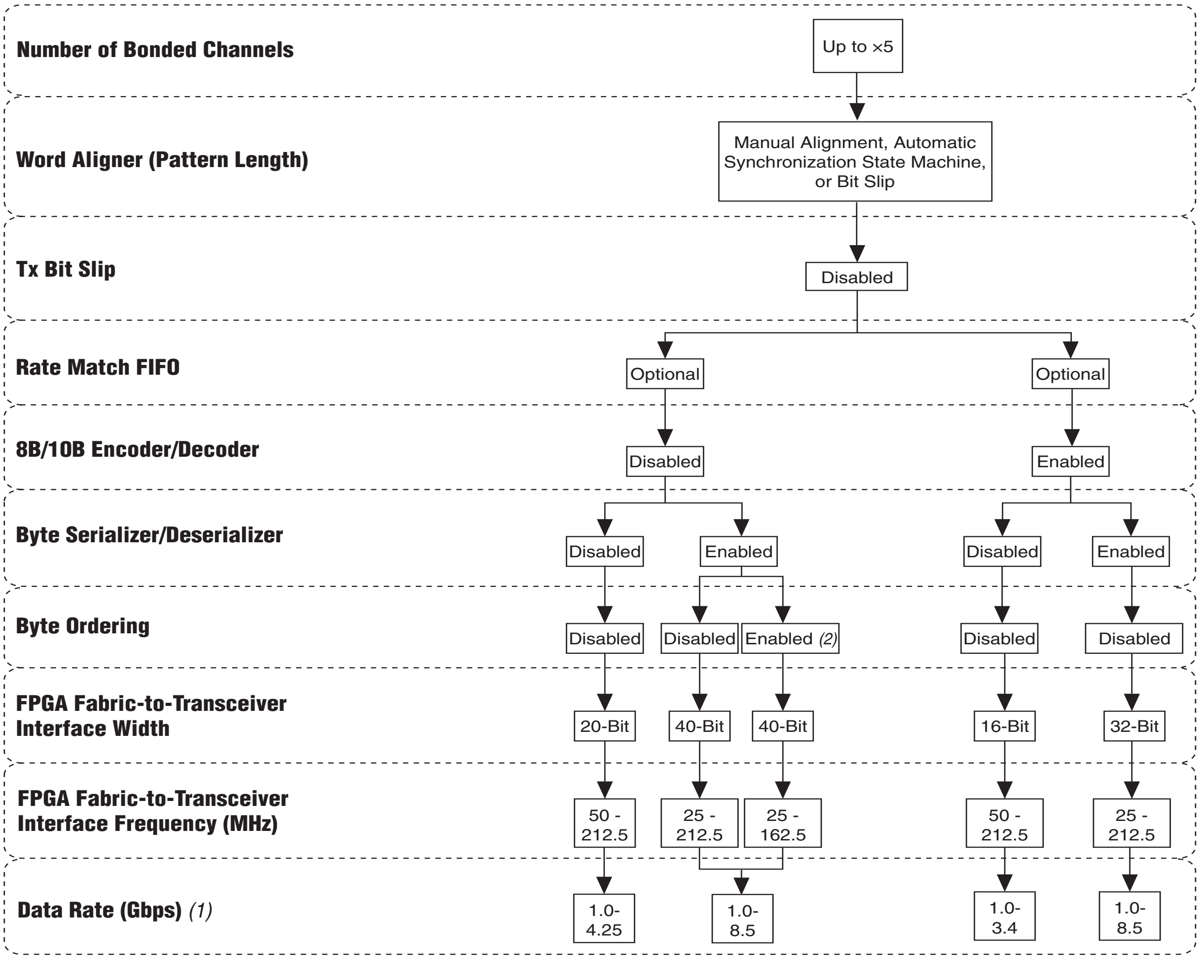

Number of Bonded Channels
Up to ×5
Word Aligner (Pattern Length)
Manual Alignment, Automatic Synchronization State Machine, or Bit Slip
Tx Bit Slip
Optional
Rate Match FIFO
Disabled
8B/10B Encoder/Decoder
Disabled
Byte Serializer/Deserializer
Disabled
Enabled
Byte Ordering
Disabled
Disabled
FPGA Fabric-to-Transceiver Interface Width
16-Bit
32-Bit
FPGA Fabric-to-Transceiver Interface Frequency (MHz)
62.5 - 250
31.25 - 250
Data Rate (Gbps)
1.0 - 4.0
1.0 - 8.0

Figure 5–12 shows the available options for the standard PCS custom 20-bit PMA-PCS interface width. The maximum frequencies shown in Figure 5–12 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–12. Standard PCS Custom 20-Bit PMA-PCS Interface Width



Notes to Figure 5–12:

(1) The maximum data rate specification shown in Figure 5–12 is valid only for the -2 (fastest) speed grade devices. For data rate specifications for other speed grades, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

(2) The byte ordering block is available only if you select the word alignment pattern length of 20 bits.

Low Latency Using the Standard PCS Datapath

A Low Latency PHY IP core using the standard PCS is available for 8-bit, 10-bit, 16-bit, or 20-bit PCS data width configurations. Figure 5–13 shows the available transmitter and receiver channel PCS blocks if you use the Low Latency PHY IP core.

The low latency PCS datapath consists of the following blocks:

- Transmitter channel PCS
  - TX phase compensation FIFO
  - Byte serializer
- Receiver channel PCS
  - RX phase compensation FIFO
  - Byte deserializer

Figure 5–13. Standard PCS Low Latency Datapath

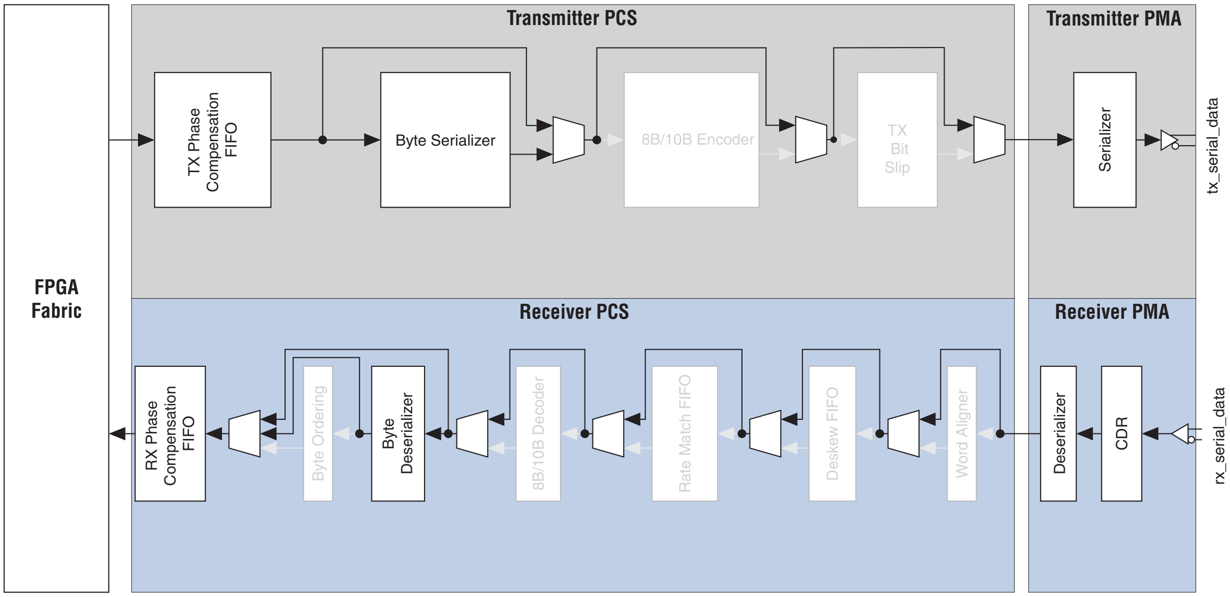


Table 5–4 lists the PCS and PMA interface widths and data rates in custom single-width and double-width modes.

Table 5–4. PCS-PMA Interface Widths and Data Rates

Low Latency PHY IP Core	Supported Data Rate Range PMA
Low latency 8-bit width	600 Mbps to 4.0 Gbps
Low latency 10-bit width	600 Mbps to 5.0 Gbps
Low latency 16-bit or 20-bit width	1 Gbps to 8.5 Gbps

In low latency PCS configurations, the TX and RX phase compensation FIFOs are always enabled. Depending on the targeted data rate, you can optionally bypass the byte serializer and deserializer blocks.

Figure 5–14 shows the available options for the standard PCS low latency 8-bit PMA-PCS interface width. The blocks shown as “Disabled” are not used, but incur latency. The blocks shown as “Bypassed” are not used and do not incur any latency. The maximum frequencies shown in Figure 5–14 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–14. Standard PCS Low Latency 8-Bit PMA-PCS Interface Width

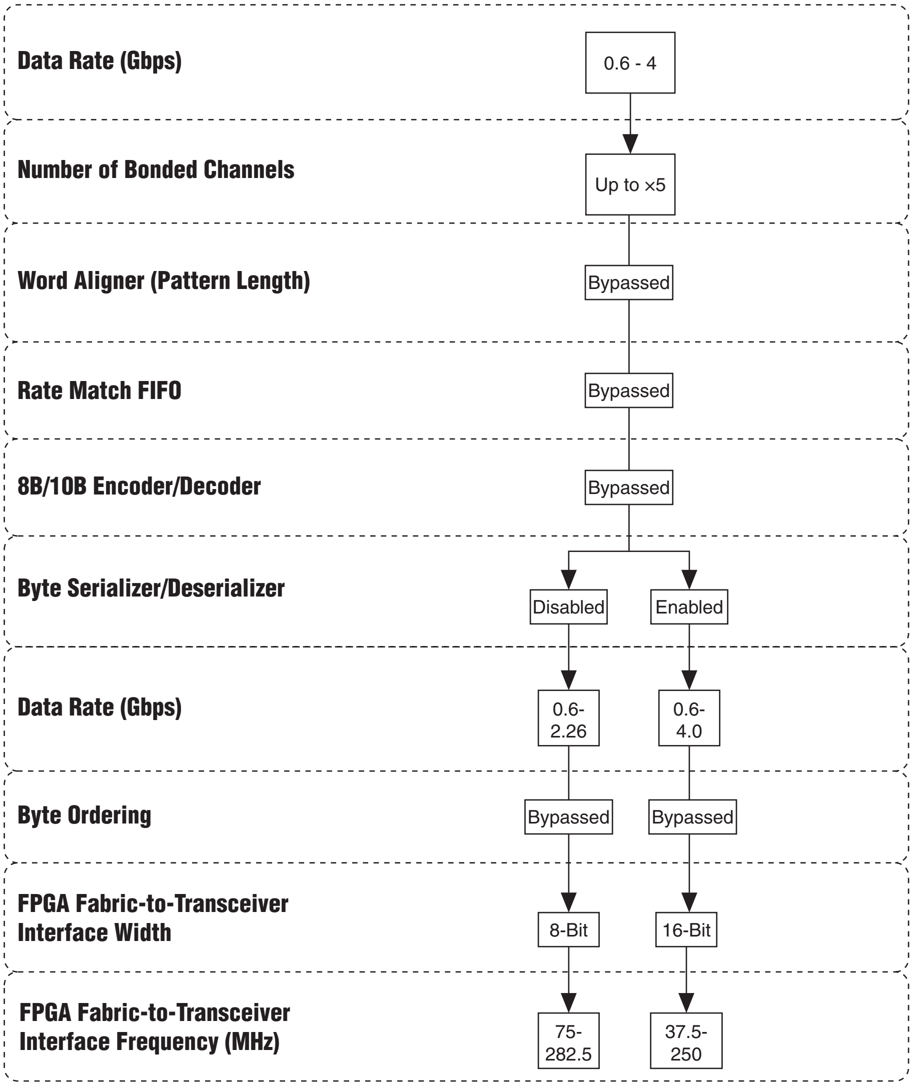


Figure 5–15 shows the available options for the standard PCS low latency 10-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used, but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies shown in Figure 5–15 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–15. Standard PCS Low Latency 10-Bit PMA-PCS Interface Width

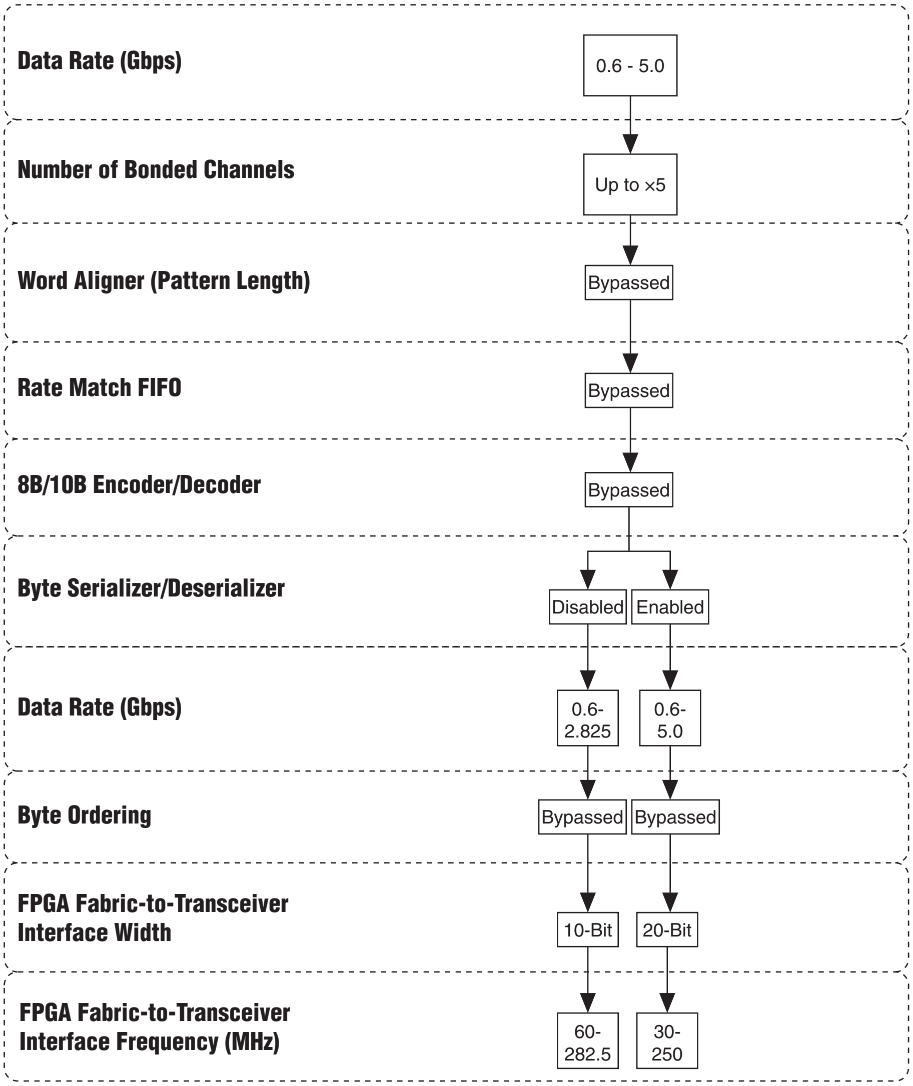


Figure 5–16 shows the available options for the standard PCS low latency 16-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used, but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies shown in Figure 5–16 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–16. Standard PCS Low Latency 16-Bit PMA-PCS Interface Width

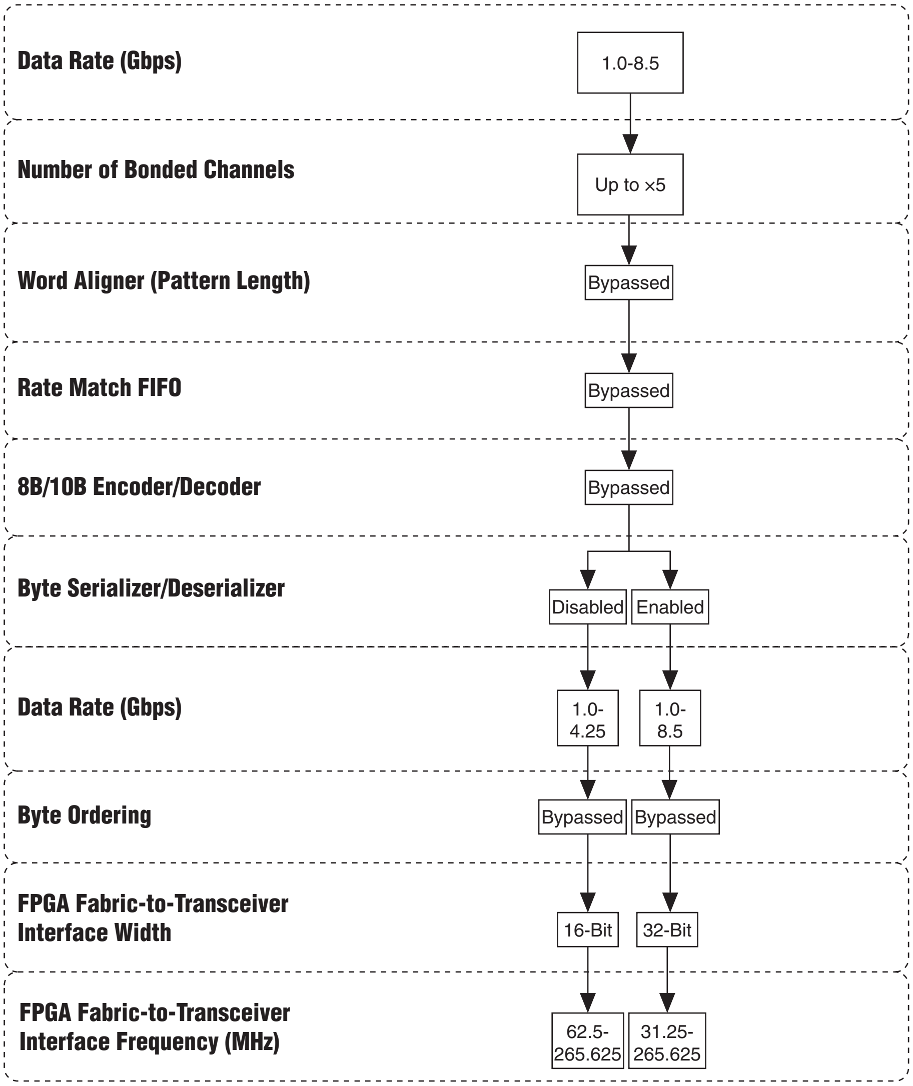
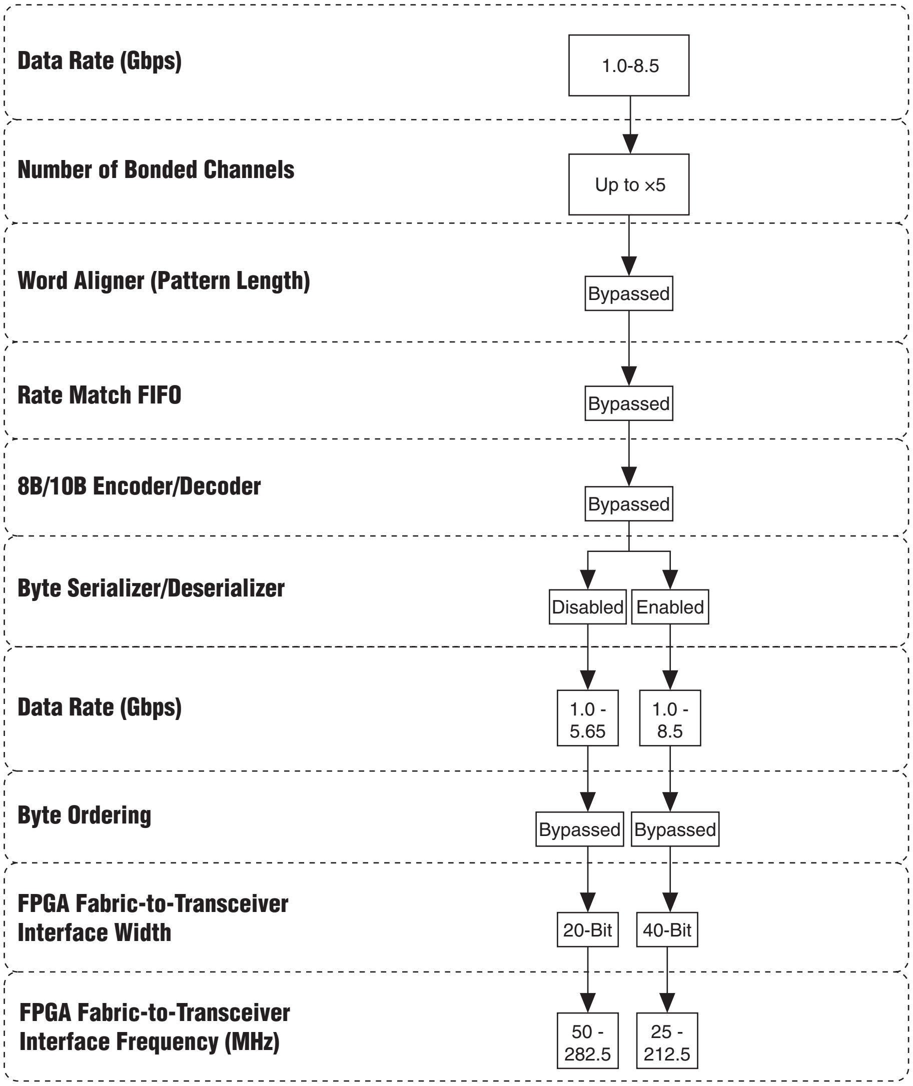


Figure 5–17 shows the available options for the standard PCS low latency 20-bit PMA-PCS interface width. The blocks shown as "Disabled" are not used, but incur latency. The blocks shown as "Bypassed" are not used and do not incur any latency. The maximum frequencies shown in Figure 5–17 are for the fastest devices.

For more information about the maximum data rate for a certain speed grade, refer to the *DC and Switching Characteristics for Stratix V Devices* chapter.

Figure 5–17. Standard PCS Low Latency 20-Bit PMA-PCS Interface Width



Document Revision History

Table 5–5 lists the revision history for this chapter.

Table 5–5. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Updated Table 5–1. ■ Added Figure 5–3. ■ Updated Figure 5–9, Figure 5–11, Figure 5–12 and Figure 5–17. ■ Moved chapter to volume 3 for the Quartus II software release 11.0. ■ Minor text edits.
December 2010	1.1	■ Updated “Clocking” on page 5–3. ■ Updated “Standard PCS Custom and Low Latency Configurations” on page 5–7. ■ Updated Figure 5–1, Figure 5–2, Figure 5–3, Figure 5–4, Figure 5–5, Figure 5–7, Figure 5–8, Figure 5–9, Figure 5–10, and Figure 5–11. ■ Minor text edits.
July 2010	1.0	Initial release.

6. Transceiver Loopback Support in Stratix V Devices

SV52007-2.0

This chapter describes the loopback options used on Stratix® V GX and GS devices, which allow you to verify how different functional blocks work in the transceiver standard physical coding sublayer (PCS).

The four available standard PCS loopback options are:

- "Serial Loopback"—available in all configurations except the PCI Express® (PCIe®) configuration
- "PCIe Reverse Parallel Loopback" on page 6–2—supported in the PCIe configuration only
- "Reverse Serial Loopback" on page 6–3—supported in custom configuration
- "Reverse Serial Pre-CDR Loopback" on page 6–4—supported in custom configuration

Serial Loopback

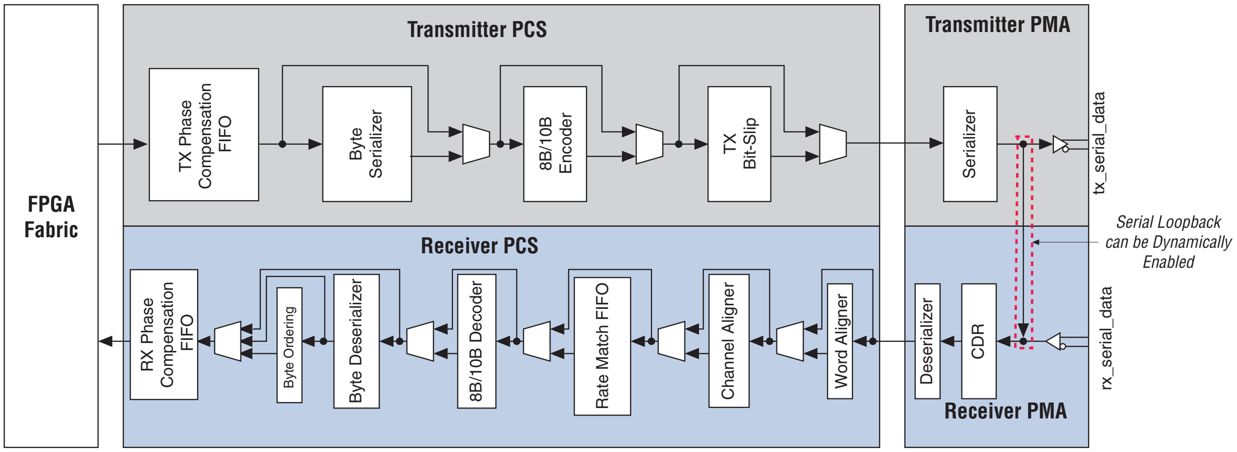
Serial loopback is available for all configurations except the PCIe configuration. Figure 6–1 shows the datapath for serial loopback. The data from the FPGA fabric passes through the transmitter channel and is looped back to the receiver channel, bypassing the receiver buffer. The received data is available to the FPGA logic for verification. With this option, you can review how the enabled PCS and physical media attachment (PMA) functional blocks in the transmitter and receiver channel work. Furthermore, you can dynamically enable serial loopback on a channel-by-channel basis.

When you enable serial loopback, the transmitter channel sends the data to both the `tx_serial_data` output port and to the receiver channel. The differential output voltage on the `tx_serial_data` ports is based on the selected differential output voltage (V_{OD}) settings. The looped back data is received by the receiver clock data recovery (CDR) and is retimed through different clock domains. You must provide an alignment pattern for the word aligner to enable the receiver channel to retrieve the byte boundary.

If the device is not in the serial loopback configuration and is receiving data from a remote device, the receiver CDR's recovered clock is locked to the data from that source. If the device is placed in the serial loopback configuration, the data source to the receiver changes from the remote device to the local transmitter channel. This prompts the receiver CDR to start tracking the phase of the new data source. During this time, the receiver CDR's recovered clock may be unstable. As the receiver PCS is running off of this recovered clock, you must place the receiver PCS under reset by asserting the `rx_digitalreset` signal during this time period.

When moving into or out of serial loopback, you must assert `rx_digitalreset` for a minimum of two parallel clock cycles.

Figure 6–1. Serial Loopback Datapath



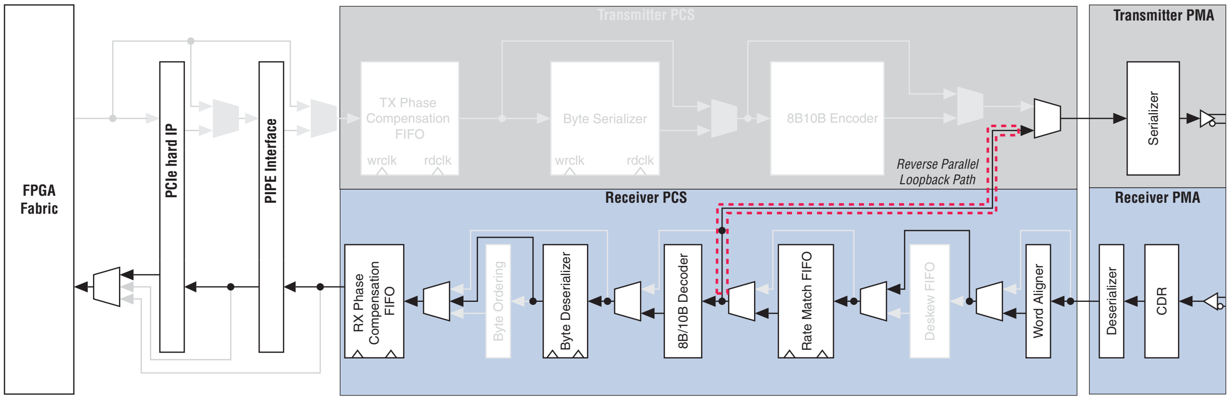
PCIe Reverse Parallel Loopback

PCIe reverse parallel loopback is only available in the PCIe configuration for Gen1 and Gen2 data rates. As shown in Figure 6–2, the received serial data passes through the receiver CDR, deserializer, word aligner, and rate matching FIFO buffer. It is then looped back to the transmitter serializer and transmitted out through the `tx_serial_data` port. The received data is also available to the FPGA fabric through the `rx_parallel_data` signal. This loopback configuration is designed for the PCIe specification 2.0. To enable this loopback configuration, assert the `tx_detectrxloopback` signal.

This is the only loopback option supported in the PCIe configuration.

For more information, refer to the “PCI Express Reverse Parallel Loopback” section in the *Transceiver Protocol Configuration in Stratix V Devices* chapter.

Figure 6–2. PCIe Reverse Parallel Loopback Configuration Datapath *(Note 1)*



Note to Figure 6–2:

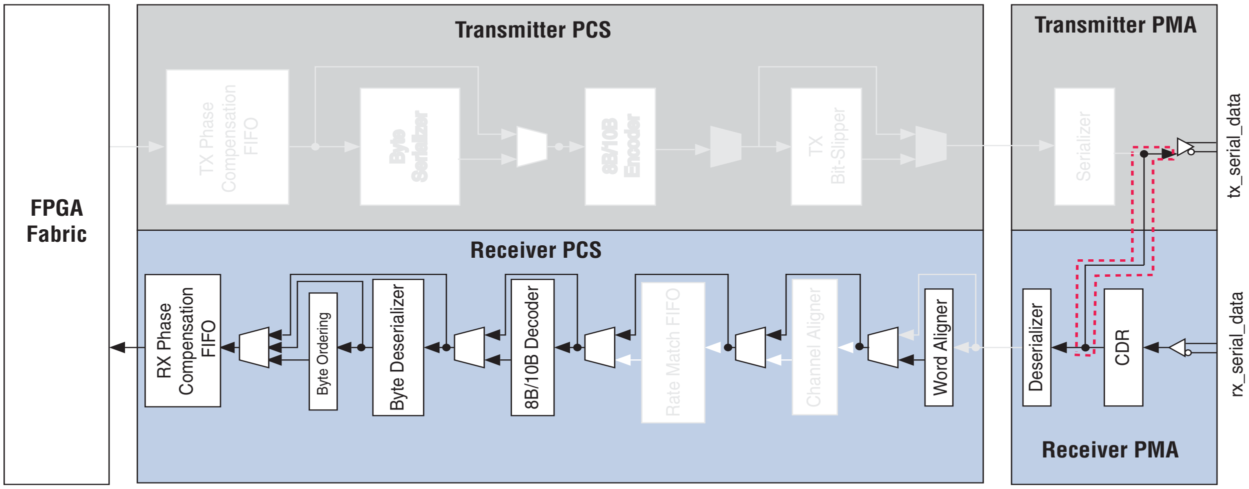
(1) Grayed-out blocks are not active in this configuration.

Reverse Serial Loopback

Reverse serial loopback is available as a subprotocol under custom configuration. In reverse serial loopback, the data is received through the `rx_serial_data` port, retimed through the receiver CDR, and sent out to the `tx_serial_data` port. The received data is also available to the FPGA logic. No dynamic pin control is available to select or deselect reverse serial loopback. Figure 6–3 shows the transceiver channel datapath for reverse serial loopback mode.

The active block of the transmitter channel is only the transmitter buffer. You can change the V_{OD} and the pre-emphasis first post tap values on the transmitter buffer through the ALTGX MegaWizard™ Plug-In Manager or through the dynamic reconfiguration controller. Reverse serial loopback is often implemented when using a bit error rate tester (BERT) on the upstream transmitter.

Figure 6–3. Reverse Serial Loopback Datapath *(Note 1)*



Note to Figure 6–3:

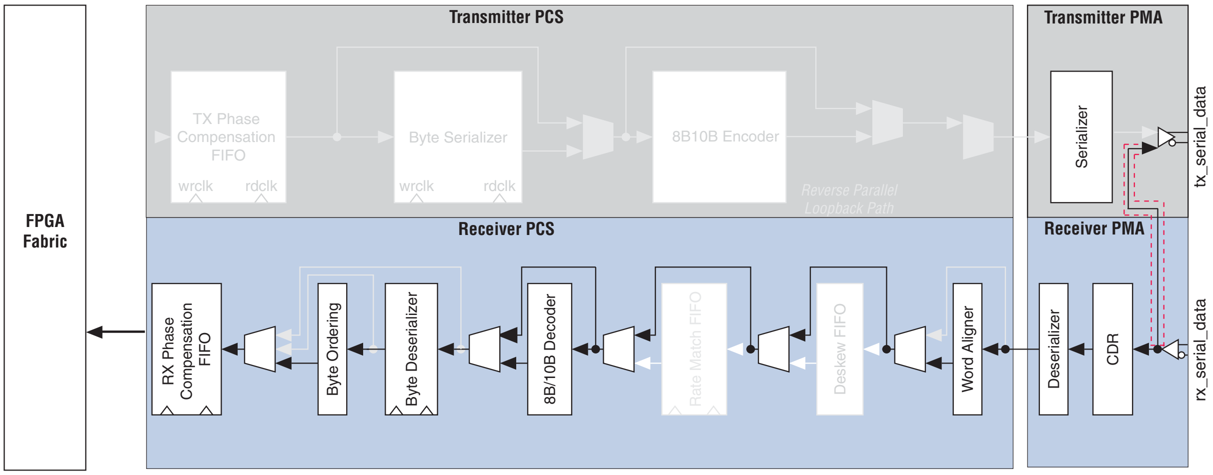
(1) Grayed-out blocks are not active in this configuration.

Reverse Serial Pre-CDR Loopback

Reverse serial pre-CDR loopback is available as a subprotocol under custom configuration. In reverse serial pre-CDR loopback, the data received through the `rx_serial_data` port is looped back to the `tx_serial_data` port *before* the receiver CDR. The received data is also available to the FPGA logic. Figure 6–4 shows the transceiver channel datapath for reverse serial pre-CDR loopback mode.

The active block of the transmitter channel is only the transmitter buffer. You can change the V_{OD} on the transmitter buffer through the ALTGX MegaWizard Plug-In Manager. The pre-emphasis settings for the transmitter buffer cannot be changed in this configuration.

Figure 6–4. Reverse Serial Pre-CDR Loopback Datapath ***(Note 1)***



Note to Figure 6–4:

(1) Grayed-out blocks are not active in this configuration.

Document Revision History

Table 6–1 lists the revision history for this chapter.

Table 6–1. Document Revision History

Date	Version	Changes
May 2011	2.0	■ Added the "Reverse Serial Loopback" and "Reverse Serial Pre-CDR Loopback" sections. ■ Updated Figure 6–2. ■ Updated the chapter title. ■ Chapter moved to Volume 3. ■ Minor text edits.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

7. Dynamic Reconfiguration in Stratix V Devices

SV52008-1.2

This chapter describes the dynamic reconfiguration features available in Stratix® V transceivers.

For information about features that will be supported in a future release of the Quartus® II software, refer to the *Upcoming Stratix V Device Features* document.

Stratix V transceivers have a dedicated reconfiguration space to support the following features:

- Offset cancellation
- Physical media attachment (PMA) controls reconfiguration

You can use the transceiver reconfiguration controller to reconfigure the following transceiver PMA controls:

- Differential output voltage (V_{OD})
- Pre-emphasis taps
- Receiver equalization control
- Receiver equalization DC gain

For more information about the controller, its settings, and implementing it in your design, refer to the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

PMA Controls Reconfiguration

Every transceiver channel in Stratix V devices has offset cancellation circuitry to compensate for the offset variations due to process. After the device powers up for the first time, you can dynamically reconfigure the PMA controls only after the offset cancellation and reset sequence are done, as indicated by the assertion of the `reconfig_busy` output signal of the transceiver reconfiguration controller.

You can continue with subsequent reconfigurations of PMA controls when the `reconfig_busy` status output is low.

For more information about the reset and powerdown of Stratix V transceivers, refer to the *Transceiver Reset Control in Stratix V Devices* chapter.

For more information about reconfiguring a specific PMA control, refer to “How to Reconfigure a PMA Control” in the *Transceiver Reconfiguration Controller* chapter of the *Altera Transceiver PHY IP Core User Guide*.

For a specific design example, refer to *AN 645: Dynamic Reconfiguration of PMA Controls in Stratix V Designs*.

Document Revision History

Table 7–1 lists the revision history for this chapter.

Table 7–1. Document Revision History

Date	Version	Changes
May 2011	1.2	■ Deleted "Indirect Addressing" section and part of introduction. ■ Added link to AN 645. ■ Chapter moved to Volume 3.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.

Additional Information

This chapter provides additional information about the document and Altera.

How to Contact Altera

To locate the most up-to-date information about Altera products, refer to the following table.

Contact *(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 to Table:

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

Typographic Conventions

The following table shows the typographic conventions this document uses.

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Indicate 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, software utility names, and GUI labels. For example, **\qdesigns** directory, **D:** drive, and **chiptrip.gdf** file.
Italic Type with Initial Capital Letters	Indicate document titles. For example, *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** file.
Initial Capital Letters	Indicate keyboard keys and menu names. For example, the 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`. The suffix `n` denotes an active-low signal. 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`).
↵	An angled arrow instructs you to press the Enter key.
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 question mark directs you to a software help system with related information.
👣	The feet direct you to another document or website with related information.
⚠ CAUTION	A caution calls attention to a condition or possible situation that can damage or destroy the product or your work.
⚡ WARNING	A warning calls attention to a condition or possible situation that can cause you injury.
✉	The envelope links to the Email Subscription Management Center page of the Altera website, where you can sign up to receive update notifications for Altera documents.