

Stratix V Device Handbook

Volume 1: Device Interfaces and Integration



Subscribe



Send Feedback

SV-5V1
2020.09.10

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

ALTERA
now part of Intel®

Contents

Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices.....	1-1
LAB	1-1
MLAB	1-2
Local and Direct Link Interconnects	1-3
Shared Arithmetic Chain and Carry Chain Interconnects	1-4
LAB Control Signals.....	1-5
ALM Resources	1-6
ALM Output	1-7
ALM Operating Modes	1-8
Normal Mode	1-8
Extended LUT Mode	1-10
Arithmetic Mode	1-11
Shared Arithmetic Mode	1-12
LAB Power Management Techniques	1-14
Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices Revision History.....	1-14
Embedded Memory Blocks in Stratix V Devices.....	2-1
Types of Embedded Memory.....	2-1
Embedded Memory Capacity in Stratix V Devices.....	2-2
Embedded Memory Design Guidelines for Stratix V Devices.....	2-2
Guideline: Consider the Memory Block Selection.....	2-2
Guideline: Implement External Conflict Resolution.....	2-3
Guideline: Customize Read-During-Write Behavior.....	2-3
Guideline: Consider Power-Up State and Memory Initialization.....	2-7
Guideline: Control Clocking to Reduce Power Consumption.....	2-7
Embedded Memory Features.....	2-7
Embedded Memory Configurations.....	2-9
Mixed-Width Port Configurations.....	2-9
Embedded Memory Modes.....	2-11
Embedded Memory Clocking Modes.....	2-12
Clocking Modes for Each Memory Mode.....	2-12
Asynchronous Clears in Clocking Modes.....	2-13
Output Read Data in Simultaneous Read/Write.....	2-13
Independent Clock Enables in Clocking Modes.....	2-14
Parity Bit in Memory Blocks.....	2-14
Byte Enable in Embedded Memory Blocks.....	2-14
Byte Enable Controls in Memory Blocks.....	2-15
Data Byte Output.....	2-15
RAM Blocks Operations.....	2-16
Memory Blocks Packed Mode Support.....	2-16
Memory Blocks Address Clock Enable Support.....	2-16
Memory Blocks Asynchronous Clear.....	2-18

Memory Blocks Error Correction Code Support.....	2-19
Error Correction Code Truth Table.....	2-19
Embedded Memory Blocks in Stratix V Devices Revision History.....	2-20
Variable Precision DSP Blocks in Stratix V Devices.....	3-1
Features.....	3-1
Supported Operational Modes in Stratix V Devices.....	3-2
Resources.....	3-3
Design Considerations.....	3-4
Operational Modes.....	3-5
Internal Coefficient and Pre-Adder.....	3-5
Accumulator.....	3-5
Chainout Adder.....	3-6
Block Architecture.....	3-6
Input Register Bank.....	3-7
Pre-Adder.....	3-9
Internal Coefficient.....	3-9
Multipliers.....	3-10
Accumulator and Chainout Adder.....	3-10
Systolic Registers.....	3-11
Output Register Bank.....	3-11
Operational Mode Descriptions.....	3-11
Independent Multiplier Mode.....	3-11
Independent Complex Multiplier Mode.....	3-16
Multiplier Adder Sum Mode.....	3-20
Sum of Square Mode.....	3-23
18 x 18 Multiplication Summed with 36-Bit Input Mode.....	3-24
Systolic FIR Mode.....	3-25
Variable Precision DSP Block Control Signals.....	3-27
Variable Precision DSP Blocks in Stratix V Devices Revision History.....	3-28
Clock Networks and PLLs in Stratix V Devices.....	4-1
Clock Networks.....	4-1
Clock Resources in Stratix V Devices.....	4-1
Types of Clock Networks.....	4-2
Clock Sources Per Quadrant.....	4-10
Types of Clock Regions.....	4-11
Clock Network Sources.....	4-12
Clock Output Connections.....	4-14
Clock Control Block.....	4-14
Clock Power Down.....	4-17
Clock Enable Signals.....	4-17
Stratix V PLLs.....	4-19
PLL Physical Counters in Stratix V Devices.....	4-20
PLL Locations in Stratix V Devices.....	4-20
PLL Migration Guidelines	4-26
Fractional PLL Architecture.....	4-27

PLL Cascading.....	4-28
PLL External Clock I/O Pins.....	4-28
PLL Control Signals.....	4-29
Clock Feedback Modes.....	4-30
Clock Multiplication and Division.....	4-38
Programmable Phase Shift.....	4-39
Programmable Duty Cycle.....	4-39
Clock Switchover.....	4-39
PLL Reconfiguration and Dynamic Phase Shift.....	4-44
Clock Networks and PLLs in Stratix V Devices Revision History.....	4-44

I/O Features in Stratix V Devices.....5-1

I/O Standards Support in Stratix V Devices.....	5-2
I/O Standards Support in Stratix V Devices.....	5-2
I/O Standards Voltage Levels in Stratix V Devices.....	5-3
MultiVolt I/O Interface in Stratix V Devices.....	5-6
I/O Design Guidelines for Stratix V Devices.....	5-6
Mixing Voltage-Referenced and Non-Voltage-Referenced I/O Standards.....	5-6
Guideline: Use the Same V_{CCPD} for All I/O Banks in a Group.....	5-7
Guideline: Observe Device Absolute Maximum Rating for 3.3 V Interfacing.....	5-8
Guideline: Use PLL Integer Mode for LVDS Applications.....	5-8
I/O Banks in Stratix V Devices.....	5-9
I/O Banks Groups in Stratix V Devices.....	5-10
Modular I/O Banks for Stratix V E Devices.....	5-10
Modular I/O Banks for Stratix V GX Devices.....	5-11
Modular I/O Banks for Stratix V GS Devices.....	5-14
Modular I/O Banks for Stratix V GT Devices.....	5-15
I/O Element Structure in Stratix V Devices.....	5-15
I/O Buffer and Registers in Stratix V Devices.....	5-16
External Memory Interfaces.....	5-17
High-Speed Differential I/O with DPA Support.....	5-17
Programmable IOE Features in Stratix V Devices.....	5-18
Programmable Current Strength.....	5-19
Programmable Output Slew Rate Control.....	5-20
Programmable IOE Delay.....	5-21
Programmable Output Buffer Delay.....	5-21
Programmable Pre-Emphasis.....	5-22
Programmable Differential Output Voltage.....	5-23
Open-Drain Output.....	5-23
Bus-Hold Circuitry.....	5-24
Pull-up Resistor.....	5-24
On-Chip I/O Termination in Stratix V Devices.....	5-24
R_S OCT without Calibration in Stratix V Devices.....	5-25
R_S OCT with Calibration in Stratix V Devices.....	5-27
R_T OCT with Calibration in Stratix V Devices.....	5-29
Dynamic OCT in Stratix V Devices.....	5-31
LVDS Input R_D OCT in Stratix V Devices.....	5-32
OCT Calibration Block in Stratix V Devices.....	5-33

OCT Calibration in Power-Up Mode.....	5-36
OCT Calibration in User Mode.....	5-36
I/O Termination Schemes for Stratix V Devices.....	5-39
Single-ended I/O Termination.....	5-40
Differential I/O Termination.....	5-42
I/O Features in Stratix V Devices Revision History.....	5-48
High-Speed Differential I/O Interfaces and DPA in Stratix V Devices.....	6-1
Dedicated High-Speed Circuitries in Stratix V Devices.....	6-2
SERDES and DPA Bank Locations in Stratix V Devices.....	6-2
LVDS SERDES Circuitry.....	6-2
SERDES I/O Standards Support in Stratix V Devices.....	6-3
True LVDS Buffers in Stratix V Devices.....	6-5
Emulated LVDS Buffers in Stratix V Devices.....	6-7
High-Speed I/O Design Guidelines for Stratix V Devices.....	6-8
PLLs and Clocking for Stratix V Devices.....	6-8
LVDS Interface with External PLL Mode.....	6-9
Pin Placement Guidelines for DPA Differential Channels.....	6-13
Differential Transmitter in Stratix V Devices.....	6-20
Transmitter Blocks.....	6-20
Transmitter Clocking.....	6-20
Serializer Bypass for DDR and SDR Operations.....	6-21
Programmable Differential Output Voltage.....	6-22
Programmable Pre-Emphasis.....	6-23
Differential Receiver in Stratix V Devices.....	6-24
Receiver Blocks in Stratix V Devices.....	6-24
Receiver Modes in Stratix V Devices.....	6-28
Receiver Clocking for Stratix V Devices.....	6-30
Differential I/O Termination for Stratix V Devices.....	6-31
Source-Synchronous Timing Budget.....	6-32
Differential Data Orientation.....	6-32
Differential I/O Bit Position.....	6-32
Transmitter Channel-to-Channel Skew.....	6-34
Receiver Skew Margin for Non-DPA Mode.....	6-34
High-Speed Differential I/O Interfaces and DPA in Stratix V Devices Revision History.....	6-38
External Memory Interfaces in Stratix V Devices.....	7-1
External Memory Performance.....	7-2
Memory Interface Pin Support in Stratix V Devices.....	7-2
Guideline: Using DQ/DQS Pins.....	7-3
DQ/DQS Bus Mode Pins for Stratix V Devices.....	7-4
DQ/DQS Groups in Stratix V E.....	7-5
DQ/DQS Groups in Stratix V GX.....	7-6
DQ/DQS Groups in Stratix V GS.....	7-8
DQ/DQS Groups in Stratix V GT.....	7-8
External Memory Interface Features in Stratix V Devices.....	7-9
UniPHY IP.....	7-9

External Memory Interface Datapath.....	7-10
DQS Phase-Shift Circuitry.....	7-10
PHY Clock (PHYCLK) Networks.....	7-19
DQS Logic Block.....	7-20
Leveling Circuitry.....	7-22
Dynamic OCT Control.....	7-24
IOE Registers.....	7-24
Delay Chains.....	7-26
I/O and DQS Configuration Blocks.....	7-28
External Memory Interfaces in Stratix V Devices Revision History.....	7-29

Configuration, Design Security, and Remote System Upgrades in Stratix V

Devices.....	8-1
Enhanced Configuration and Configuration via Protocol.....	8-1
MSEL Pin Settings.....	8-2
Configuration Sequence.....	8-4
Power Up.....	8-5
Reset.....	8-5
Configuration.....	8-6
Configuration Error Handling.....	8-6
Initialization.....	8-6
User Mode.....	8-6
Configuration Timing Waveforms.....	8-8
FPP Configuration Timing.....	8-8
AS Configuration Timing.....	8-10
PS Configuration Timing.....	8-11
Device Configuration Pins.....	8-11
I/O Standards and Drive Strength for Configuration Pins.....	8-13
Configuration Pin Options in the Intel Quartus Prime Software.....	8-14
Fast Passive Parallel Configuration.....	8-15
Fast Passive Parallel Single-Device Configuration.....	8-15
Fast Passive Parallel Multi-Device Configuration.....	8-16
Transmitting Configuration Data.....	8-18
Active Serial Configuration.....	8-19
DATA Clock (DCLK).....	8-20
Active Serial Single-Device Configuration.....	8-20
Active Serial Multi-Device Configuration.....	8-21
Estimating the Active Serial Configuration Time.....	8-23
Using EPCS and EPCQ Devices.....	8-23
Controlling EPCS and EPCQ Devices.....	8-24
Trace Length and Loading Guideline.....	8-24
Programming EPCS and EPCQ Devices.....	8-24
Passive Serial Configuration.....	8-29
Passive Serial Single-Device Configuration Using an External Host.....	8-29
Passive Serial Single-Device Configuration Using an Altera Download Cable.....	8-30
Passive Serial Multi-Device Configuration.....	8-31
JTAG Configuration.....	8-34
JTAG Single-Device Configuration.....	8-35

JTAG Multi-Device Configuration.....	8-37
CONFIG_IO JTAG Instruction.....	8-38
Configuration Data Compression.....	8-38
Enabling Compression Before Design Compilation.....	8-38
Enabling Compression After Design Compilation.....	8-38
Using Compression in Multi-Device Configuration.....	8-39
Remote System Upgrades.....	8-39
Configuration Images.....	8-40
Configuration Sequence in the Remote Update Mode.....	8-41
Remote System Upgrade Circuitry.....	8-41
Enabling Remote System Upgrade Circuitry.....	8-42
Remote System Upgrade Registers.....	8-43
Remote System Upgrade State Machine.....	8-44
User Watchdog Timer.....	8-44
Design Security.....	8-45
Altera Unique Chip ID IP Core.....	8-46
JTAG Secure Mode.....	8-46
Security Key Types.....	8-46
Security Modes.....	8-47
Design Security Implementation Steps.....	8-48
Configuration, Design Security, and Remote System Upgrades in Stratix V Devices Revision History.....	8-48
SEU Mitigation for Stratix V Devices.....	9-1
Error Detection Features.....	9-1
Configuration Error Detection.....	9-1
User Mode Error Detection.....	9-2
Internal Scrubbing.....	9-2
Specifications.....	9-2
Minimum EMR Update Interval.....	9-3
Error Detection Frequency.....	9-3
CRC Calculation Time For Entire Device.....	9-4
Using Error Detection Features in User Mode.....	9-5
Enabling Error Detection and Internal Scrubbing.....	9-5
CRC_ERROR Pin.....	9-6
Error Detection Registers.....	9-6
Error Detection Process.....	9-9
Testing the Error Detection Block.....	9-10
SEU Mitigation for Stratix V Devices Revision History.....	9-11
JTAG Boundary-Scan Testing in Stratix V Devices.....	10-1
BST Operation Control	10-1
IDCODE	10-1
Supported JTAG Instruction	10-3
JTAG Secure Mode	10-8
JTAG Private Instruction	10-8
I/O Voltage for JTAG Operation	10-8

Performing BST	10-9
Enabling and Disabling IEEE Std. 1149.1 BST Circuitry	10-9
Guidelines for IEEE Std. 1149.1 Boundary-Scan Testing.....	10-10
IEEE Std. 1149.1 Boundary-Scan Register	10-10
Boundary-Scan Cells of a Stratix V Device I/O Pin.....	10-11
IEEE Std. 1149.6 Boundary-Scan Register.....	10-13
JTAG Boundary-Scan Testing in Stratix V Devices Revision History.....	10-15
Power Management in Stratix V Devices.....	11-1
Power Consumption.....	11-1
Dynamic Power Equation.....	11-2
Programmable Power Technology.....	11-2
Temperature Sensing Diode.....	11-3
Internal Temperature Sensing Diode.....	11-4
External Temperature Sensing Diode.....	11-4
Hot-Socketing Feature.....	11-5
Hot-Socketing Implementation.....	11-6
Power-Up Sequence.....	11-7
Power-On Reset Circuitry.....	11-8
Power Supplies Monitored and Not Monitored by the POR Circuitry.....	11-10
Power Management in Stratix V Devices Revision History.....	11-11

Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices

1

2020.09.10

SV51002



Subscribe



Send Feedback

This chapter describes the features of the logic array block (LAB) in the Stratix[®] V core fabric.

The LAB is composed of basic building blocks known as adaptive logic modules (ALMs) that you can configure to implement logic functions, arithmetic functions, and register functions.

You can use half of the available LABs in the Stratix V devices as a memory LAB (MLAB).

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

This chapter contains the following sections:

- LAB
- ALM Operating Modes

Related Information

[Stratix V Device Handbook: Known Issues](#)

Lists the planned updates to the Stratix V Device Handbook chapters.

LAB

The LABs are configurable logic blocks that consist of a group of logic resources. Each LAB contains dedicated logic for driving control signals to its ALMs.

MLAB is a superset of the LAB and includes all the LAB features.

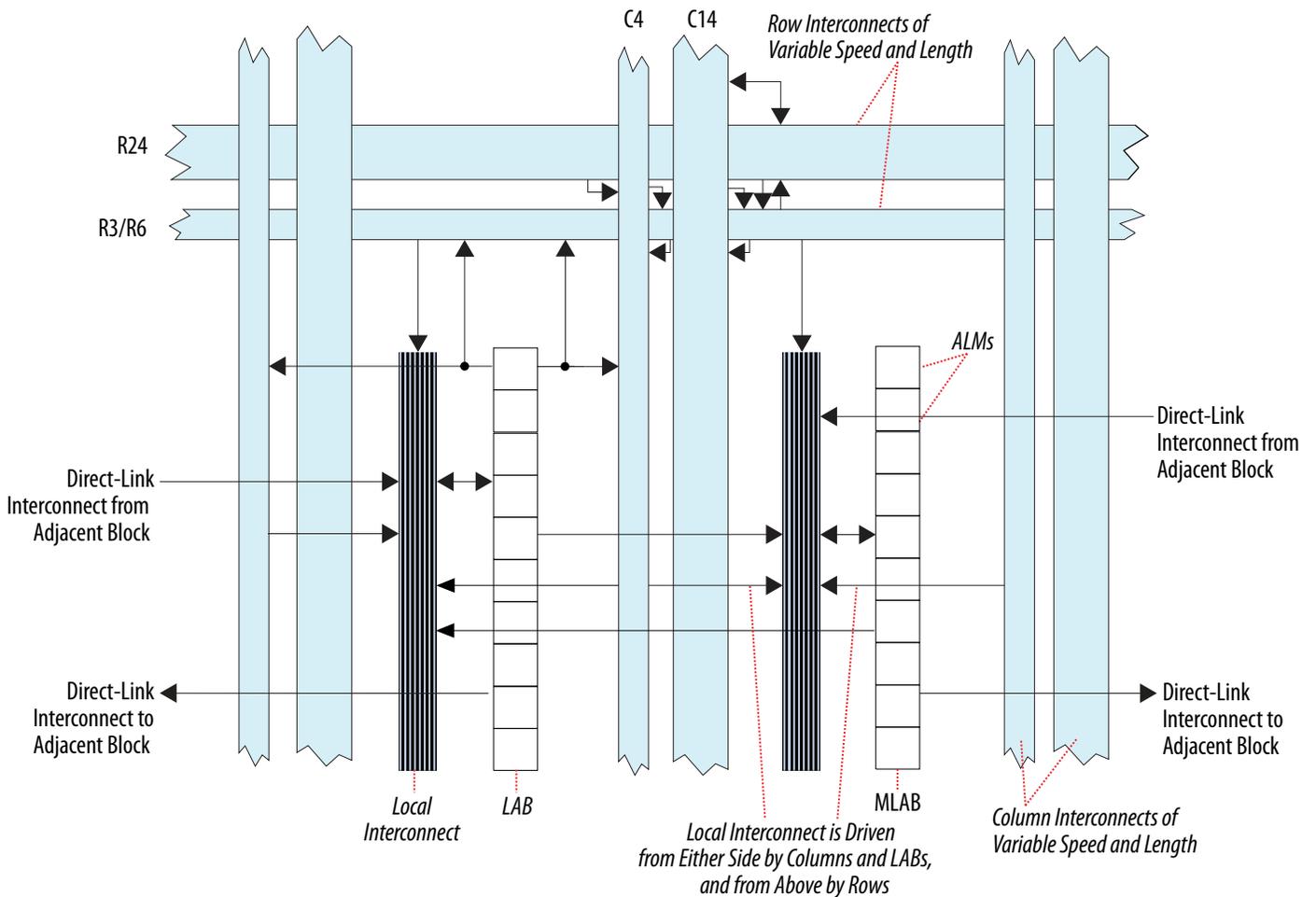
Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

Figure 1-1: LAB Structure and Interconnects Overview in Stratix V Devices

This figure shows an overview of the Stratix V LAB and MLAB structure with the LAB interconnects.

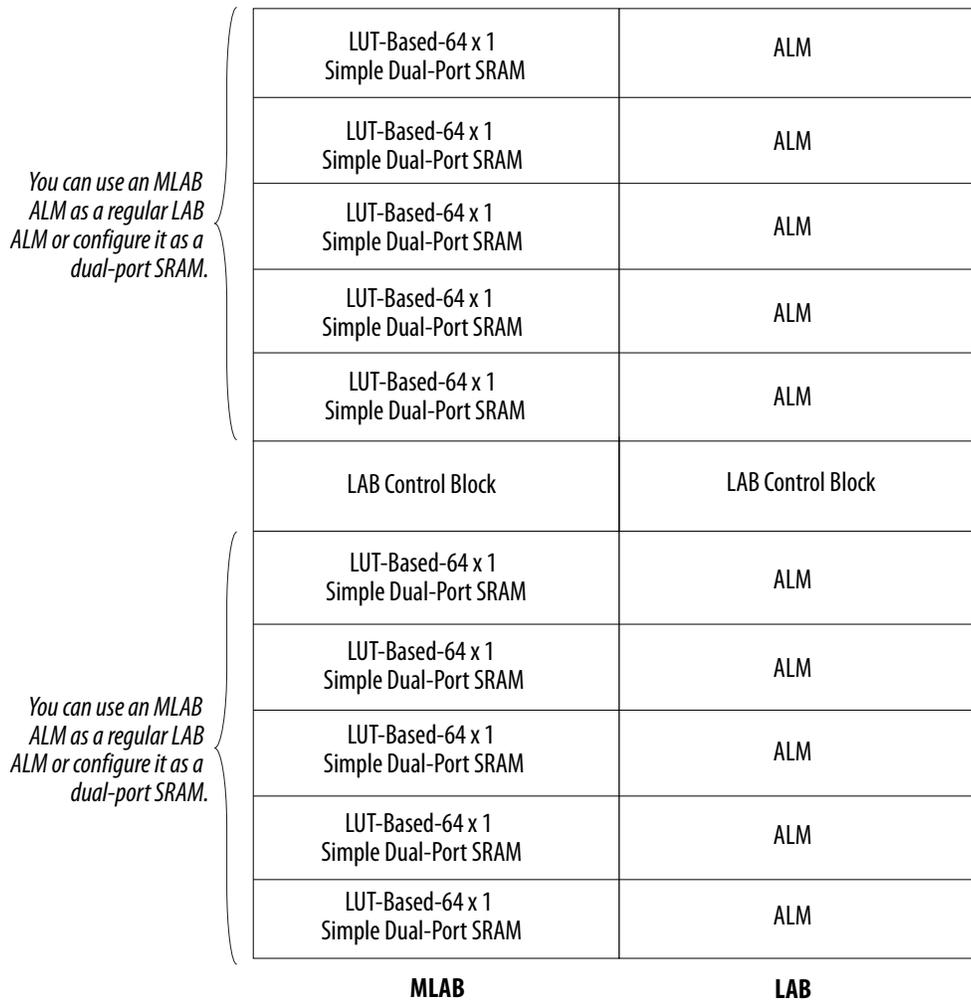


MLAB

Each MLAB supports a maximum of 640 bits of simple dual-port 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.

Figure 1-2: LAB and MLAB Structure for Stratix V Devices



Local and Direct Link Interconnects

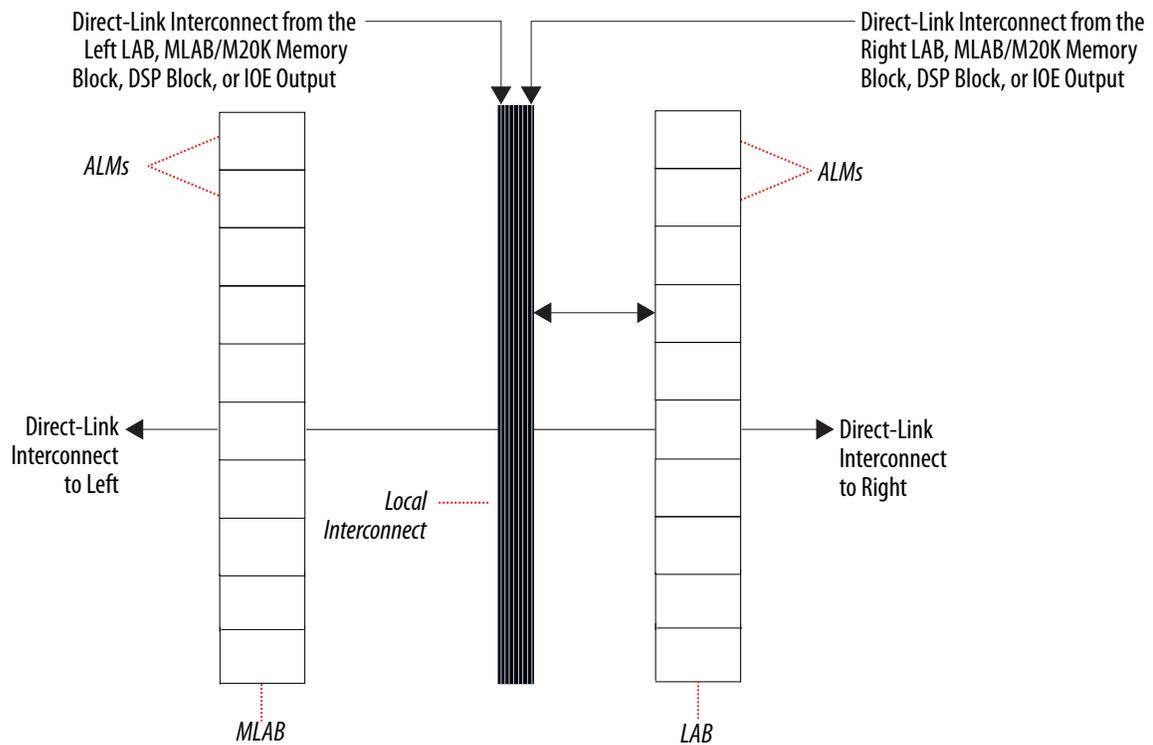
Each LAB can drive 30 ALMs through fast-local and direct-link interconnects. Ten ALMs are in any given LAB and ten ALMs are in each of the adjacent LABs.

The local interconnect can drive ALMs in the same LAB using 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 using the direct link connection.

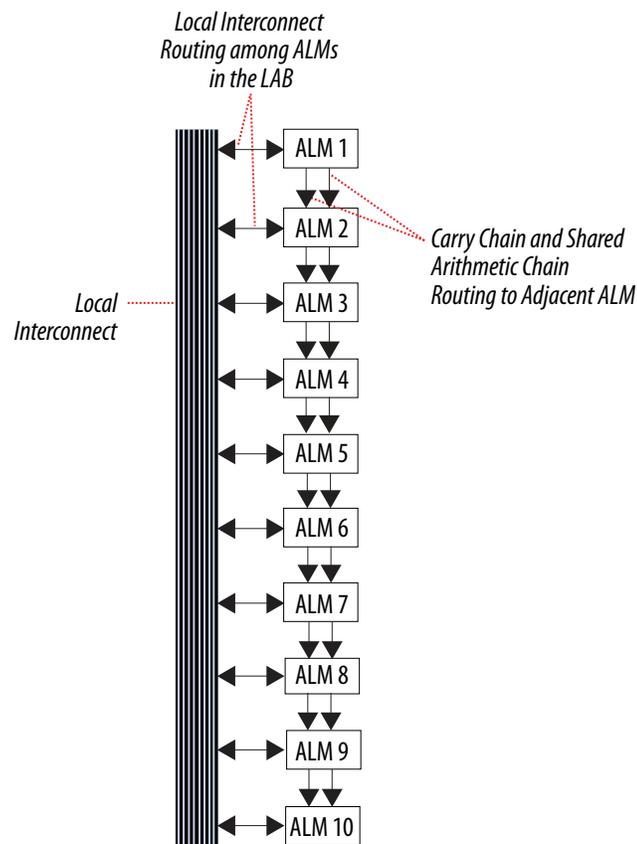
The direct link connection feature minimizes the use of row and column interconnects, providing higher performance and flexibility.

Figure 1-3: LAB Fast Local and Direct Link Interconnects for Stratix V Devices



Shared Arithmetic Chain and Carry Chain Interconnects

There are two dedicated paths between ALMs—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. These ALM-to-ALM connections bypass the local interconnect. The Intel Quartus Prime Compiler automatically takes advantage of these resources to improve utilization and performance.

Figure 1-4: Shared Arithmetic Chain and Carry Chain Interconnects

LAB Control Signals

Each LAB contains dedicated logic for driving the control signals to its ALMs, and has two unique clock sources and three clock enable signals.

The LAB control block generates up to three clocks using the two clock sources and three clock enable signals. An inverted clock source is considered as an individual clock source. Each clock and the clock enable signals are linked.

De-asserting the clock enable signal turns off the corresponding LAB-wide clock.

The LAB row clocks [5..0] and LAB local interconnects generate the LAB-wide control signals. The MultiTrack interconnect's inherent low skew allows clock and control signal distribution in addition to data. The MultiTrack interconnect consists of continuous, performance-optimized routing lines of different lengths and speeds used for inter- and intra-design block connectivity.

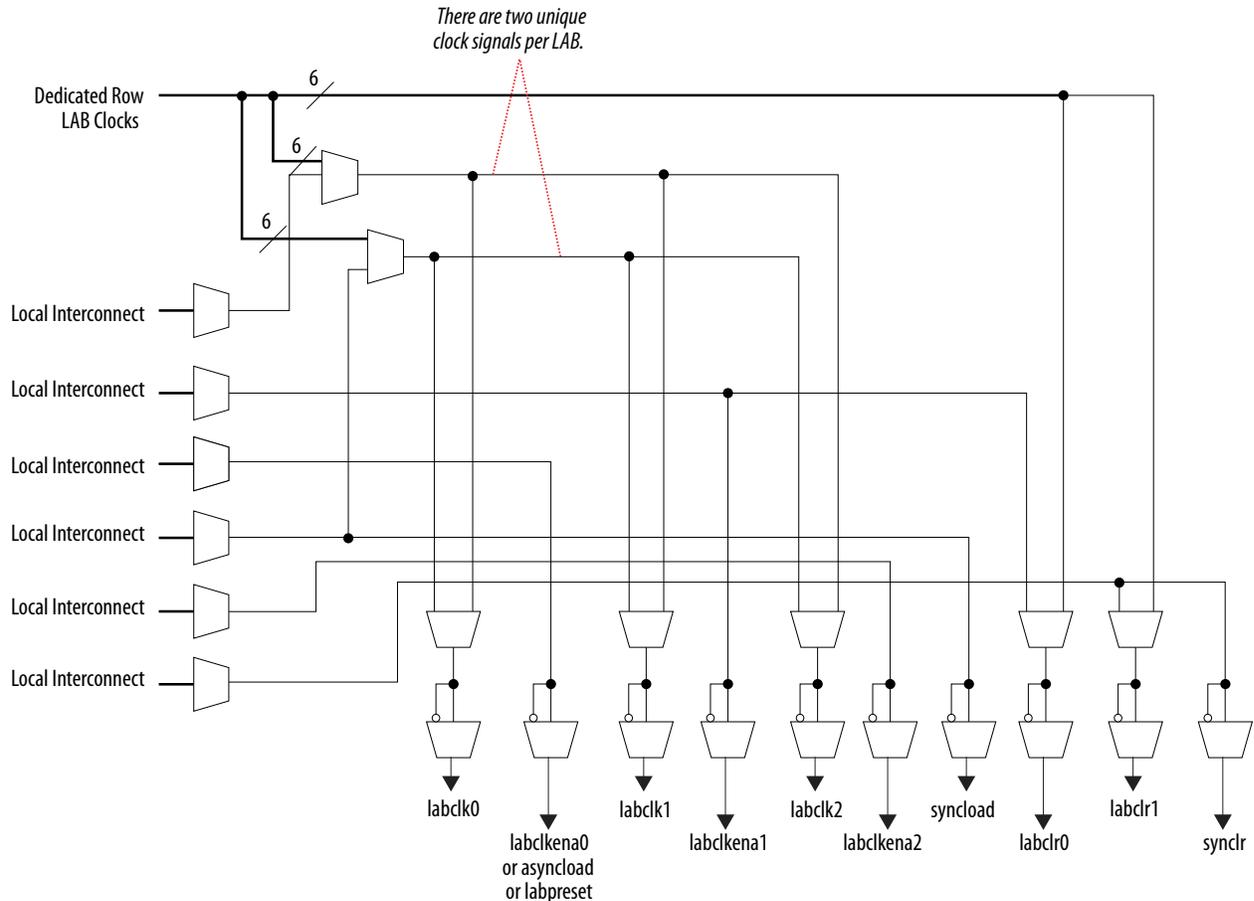
Clear and Preset Logic Control

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 **NOT-gate push-back logic** option in the Intel Quartus Prime software. 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 Intel Quartus Prime software controls this pin. This device-wide reset overrides all other control signals.

Figure 1-5: LAB-Wide Control Signals for Stratix V Devices

This figure shows the clock sources and clock enable signals in a LAB.



ALM Resources

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.

One ALM contains four programmable registers. Each register has the following ports:

- Data
- Clock
- Synchronous and asynchronous clear
- Synchronous load

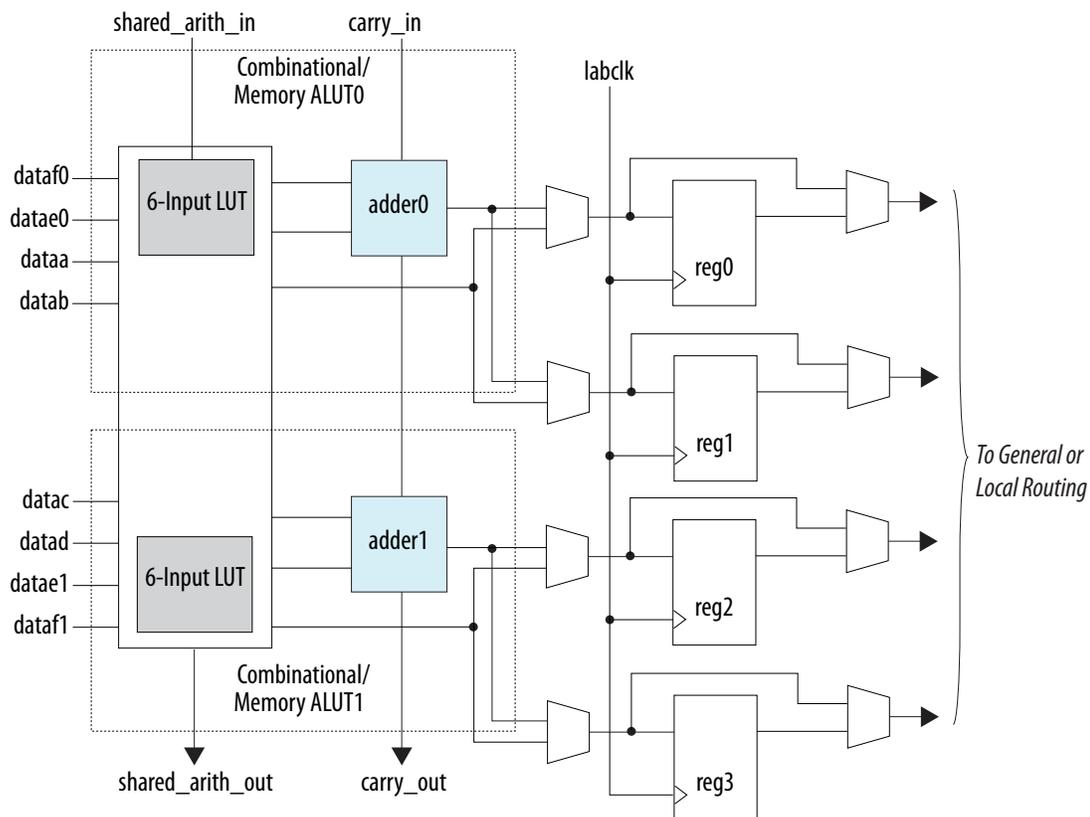
Global signals, general-purpose I/O (GPIO) pins, or any internal logic can drive the clock and clear control signals of an ALM register.

GPIO pins or internal logic drives the clock enable signal.

For combinational functions, the registers are bypassed and the output of the look-up table (LUT) drives directly to the outputs of an ALM.

Note: The Intel Quartus Prime software automatically configures the ALMs for optimized performance.

Figure 1-6: ALM High-Level Block Diagram for Stratix V Devices



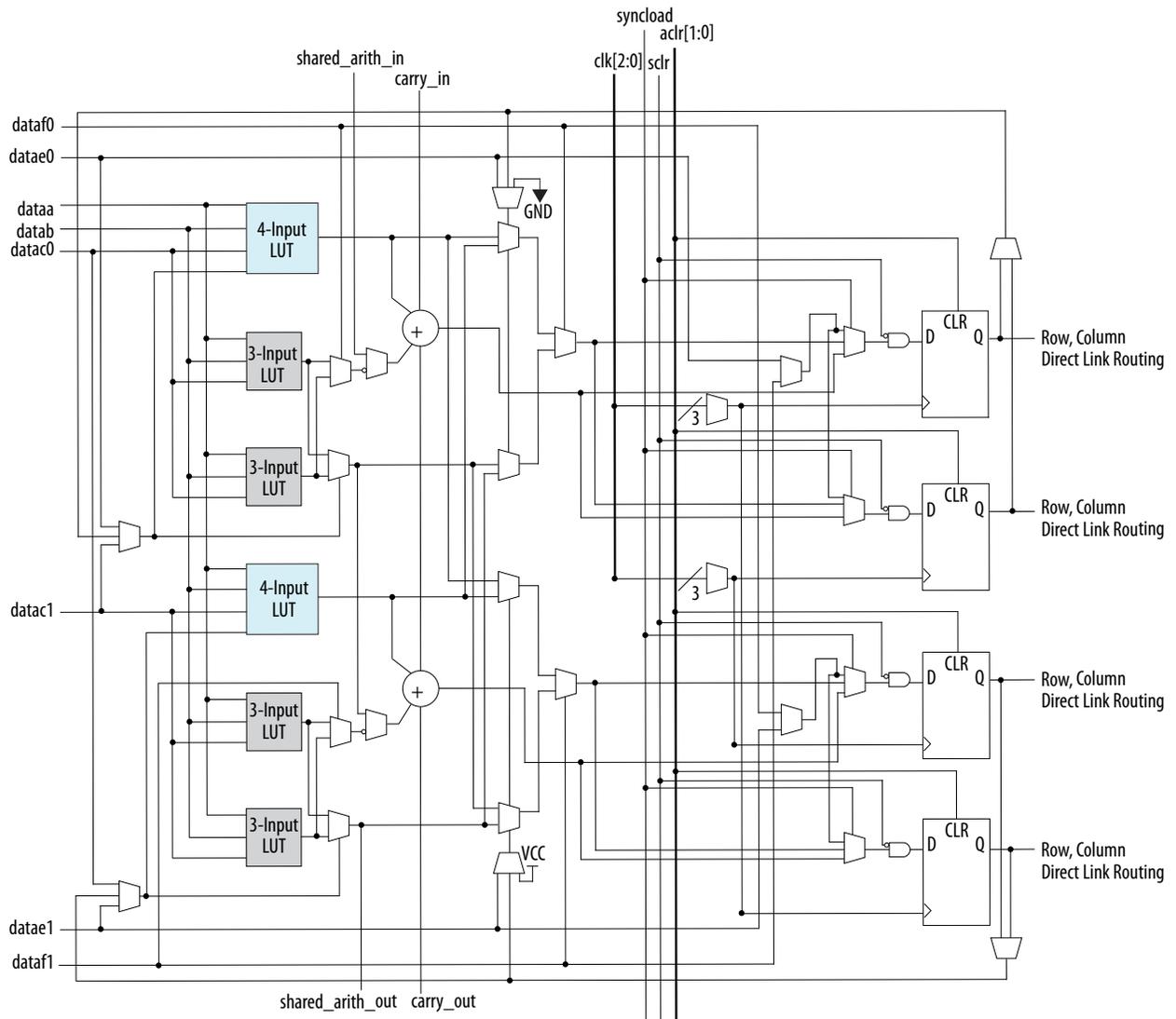
ALM Output

The general routing outputs in each ALM drive the local, row, and column routing resources. Two ALM outputs can drive column, row, or direct link routing connections, and one of these ALM outputs can also drive local interconnect resources.

The LUT, adder, or register output can drive the ALM outputs. The LUT or adder can drive one output while the register drives another output.

Register packing improves device utilization by allowing unrelated register and combinational logic to be packed into a single ALM. Another mechanism to improve fitting is to allow the register output to feed back into the look-up table (LUT) of the same ALM so that the register is packed with its own fan-out LUT. The ALM can also drive out registered and unregistered versions of the LUT or adder output.

Figure 1-7: ALM Connection Details for Stratix V Devices



ALM Operating Modes

The Stratix V ALM operates in any of the following modes:

- Normal mode
- Extended LUT mode
- Arithmetic mode
- Shared arithmetic mode

Normal Mode

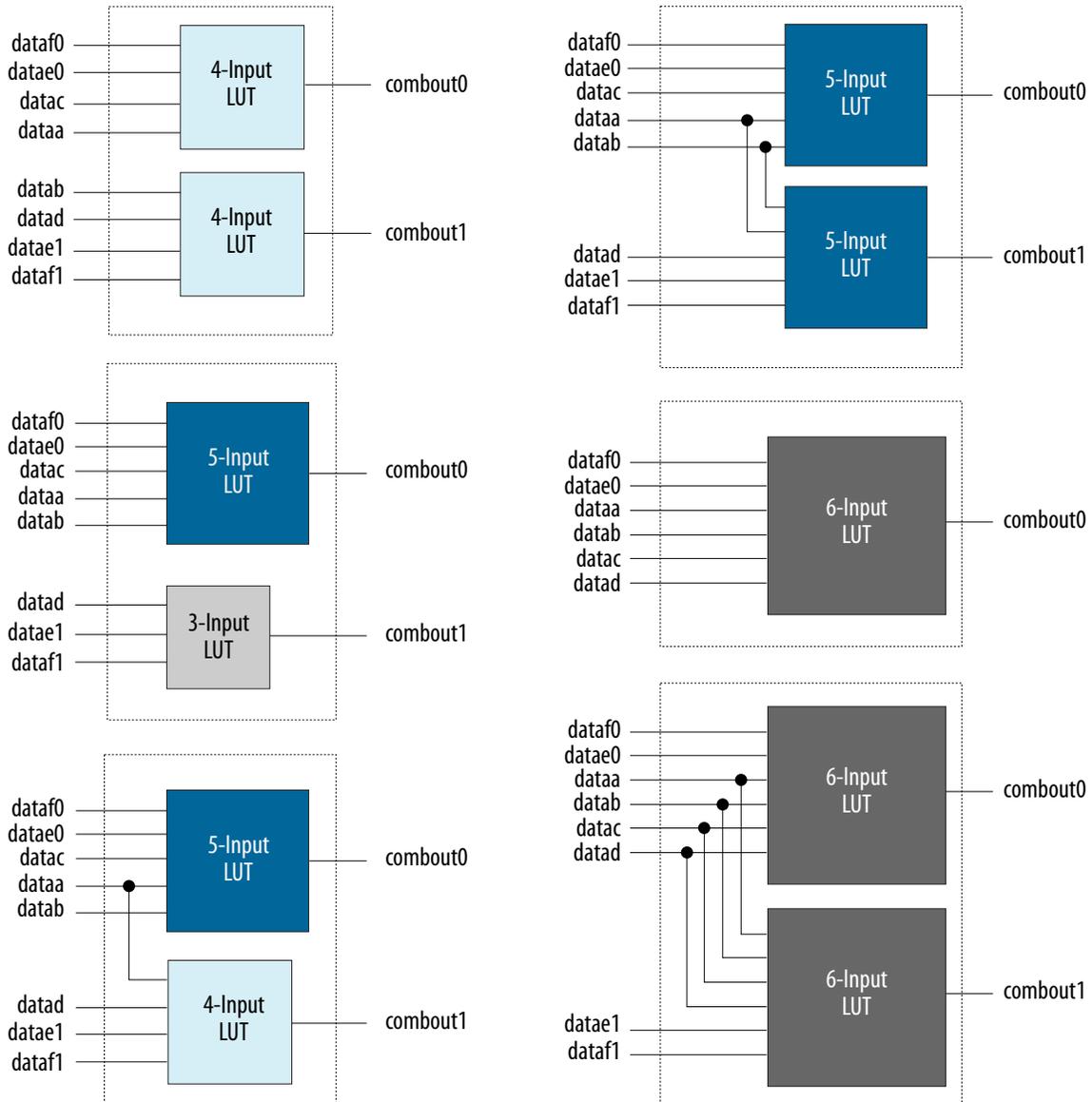
Normal mode allows two functions to be implemented in one Stratix V ALM, or a single function of up to six inputs.

Up to eight data inputs from the LAB local interconnect are inputs to the combinational logic.

The ALM can support certain combinations of completely independent functions and various combinations of functions that have common inputs.

Figure 1-8: ALM in Normal Mode

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.



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

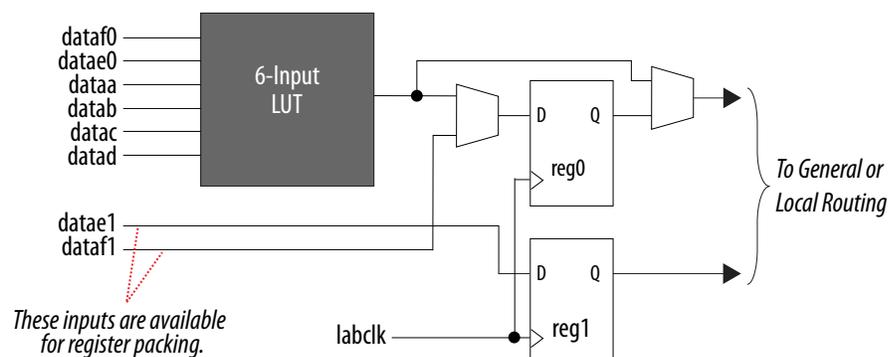
In the case of implementing 2 six-input functions in one ALM, four inputs must be shared and the combinational function must be the same. In a sparsely used device, functions that could be placed in one ALM may be implemented in separate ALMs by the Intel Quartus Prime software to achieve the best possible performance. As a device begins to fill up, the Intel Quartus Prime software automatically uses the full potential of the Stratix V ALM. The Intel Quartus Prime 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`, `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 as shown in the following figure. 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 Intel Quartus Prime Compiler automatically selects the inputs to the LUT. ALMs in normal mode support register packing.

Figure 1-9: Input Function in Normal Mode

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

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

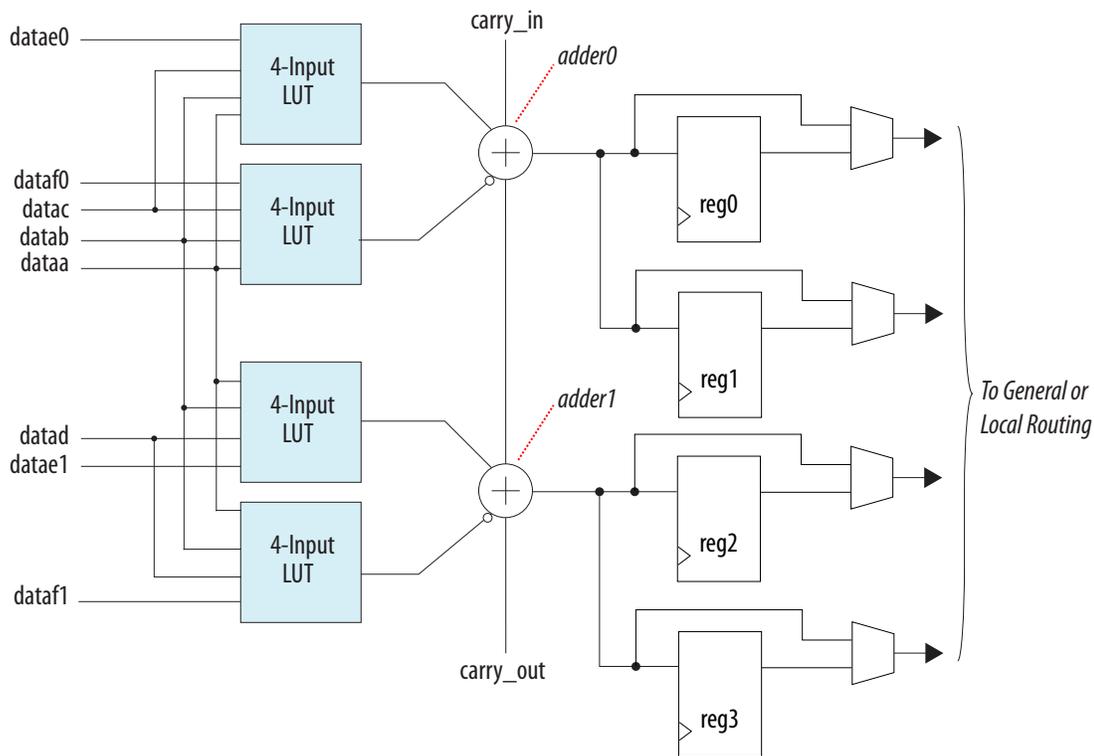


Extended LUT Mode

In this mode, if the 7-input function is unregistered, the unused eighth input is available for register packing.

Functions that fit into the template, as shown in the following figure, often appear in designs as “if-else” statements in Verilog HDL or VHDL code.

Figure 1-11: ALM in Arithmetic Mode for Stratix V Devices



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 a LAB. The final carry-out signal is routed to an ALM, where it is fed to local, row, or column interconnects.

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 in 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. You can bypass the top-half of the LAB columns and bottom-half of the MLAB columns.

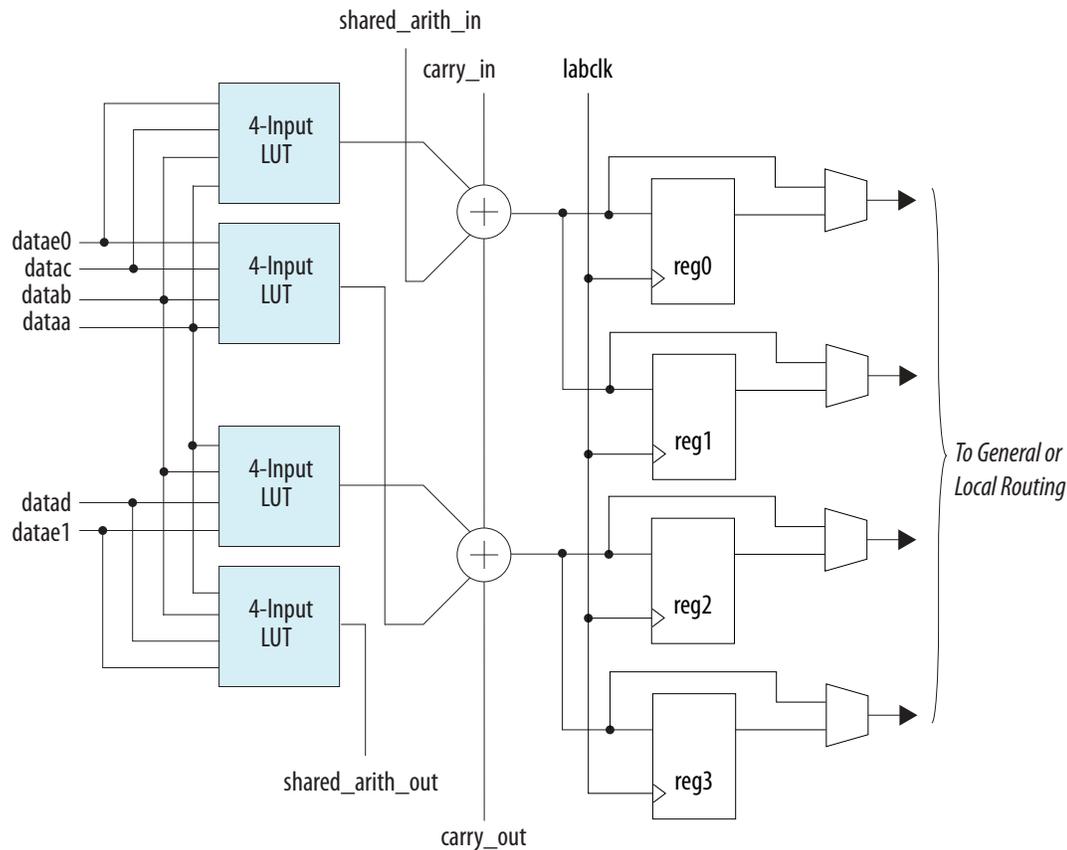
The Intel Quartus Prime Compiler creates carry chains longer than 20 ALMs (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically, allowing fast horizontal connections to the TriMatrix memory and DSP blocks. A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

Shared Arithmetic Mode

The ALM in shared arithmetic mode can implement a 3-input add in the ALM.

This mode configures the ALM with four 4-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 using a dedicated connection called the shared arithmetic chain.

Figure 1-12: ALM in Shared Arithmetic Mode for Stratix V Devices



Shared Arithmetic Chain

The shared arithmetic chain available in enhanced arithmetic mode allows the ALM to implement a 3-input adder. This significantly reduces the resources necessary to implement large adder trees or correlator functions.

The shared arithmetic chain can begin in either the first or sixth ALM in a LAB.

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

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

LAB Power Management Techniques

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

- To save AC power, the Intel Quartus Prime 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 Intel Quartus Prime software automatically chooses the appropriate mode for the LAB, based on your design and to optimize speed versus leakage trade-offs.
- Clocks represent a significant portion of dynamic power consumption because of 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 `labckena1` 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 Intel Quartus Prime 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.

Related Information

[Power Optimization chapter, Quartus Prime Handbook](#)

Provides more information about implementing static and dynamic power consumption within the LAB.

Logic Array Blocks and Adaptive Logic Modules in Stratix V Devices

Revision History

Date	Version	Changes
December 2016	2016.12.09	Added description on clock source in the LAB Control Signals section.
December 2015	2015.12.21	Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i> .
January 2014	2014.01.10	Added multiplexers for the bypass paths and register outputs in the following diagrams: <ul style="list-style-type: none"> • ALM High-Level Block Diagram for Stratix V Devices • Input Function in Normal Mode • Template for Supported 7-Input Functions in Extended LUT Mode for Stratix V Devices • ALM in Arithmetic Mode for Stratix V Devices • ALM in Shared Arithmetic Mode for Stratix V Devices

Date	Version	Changes
May 2013	2013.05.06	<ul style="list-style-type: none">• Added link to the known document issues in the Knowledge Base.• Updated the available LABs to use as a MLAB.• Removed register chain outputs information in ALM output section.• Moved all links to the Related Information section of respective topics for easy reference.
December 2012	2012.12.28	Reorganized content and updated template.
June 2012	1.4	<ul style="list-style-type: none">• Updated Figure 1–5, Figure 1–6, and Figure 1–12.• Removed register chain expression.• Minor text edits.
November 2011	1.3	<ul style="list-style-type: none">• Updated Figure 1–1, Figure 1–4, and Figure 1–6.• Removed “Register Chain” section.
May 2011	1.2	<ul style="list-style-type: none">• 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.

2020.09.10

SV51003



Subscribe



Send Feedback

The embedded memory blocks in the devices are flexible and designed to provide an optimal amount of small- and large-sized memory arrays to fit your design requirements.

Related Information

[Stratix V Device Handbook: Known Issues](#)

Lists the planned updates to the *Stratix V Device Handbook* chapters.

Types of Embedded Memory

The Stratix V devices contain two types of memory blocks:

- 20 Kb M20K blocks—blocks of dedicated memory resources. The M20K blocks are ideal for larger memory arrays while still providing a large number of independent ports.
- 640 bit memory logic array blocks (MLABs)—enhanced memory blocks that are configured from dual-purpose logic array blocks (LABs). The MLABs are ideal for wide and shallow memory arrays. The MLABs are optimized for implementation of shift registers for digital signal processing (DSP) applications, wide shallow FIFO buffers, and filter delay lines. Each MLAB is made up of ten adaptive logic modules (ALMs). In the Stratix V devices, you can configure these ALMs as ten 32 x 2 blocks, giving you one 32 x 20 simple dual-port SRAM block per MLAB. You can also configure these ALMs as ten 64 x 1 blocks, giving you one 64 x 10 simple dual-port SRAM block per MLAB.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

Embedded Memory Capacity in Stratix V Devices

Table 2-1: Embedded Memory Capacity and Distribution in Stratix V Devices

Variant	Member Code	M20K		MLAB		Total RAM Bit (Kb)
		Block	RAM Bit (Kb)	Block	RAM Bit (Kb)	
Stratix V GX	A3	957	19,140	6,415	4,009	23,149
	A4	1,900	38,000	7,925	4,953	42,953
	A5	2,304	46,080	9,250	5,781	51,861
	A7	2,560	51,200	11,736	7,335	58,535
	A9	2,640	52,800	15,850	9,906	62,706
	AB	2,640	52,800	17,960	11,225	64,025
	B5	2,100	42,000	9,250	5,781	47,781
	B6	2,660	53,200	11,270	7,043	60,243
	B9	2,640	52,800	15,850	9,906	62,706
	BB	2,640	52,800	17,960	11,225	64,025
Stratix V GT	C5	2,304	46,080	8,020	5,012	51,092
	C7	2,560	51,200	11,735	7,334	58,534
Stratix V GS	D3	688	13,760	4,450	2,781	16,541
	D4	957	19,140	6,792	4,245	23,385
	D5	2,014	40,280	8,630	5,393	45,673
	D6	2,320	46,400	11,000	6,875	53,275
	D8	2,567	51,340	13,120	8,200	59,540
Stratix V E	E9	2,640	52,800	15,850	9,906	62,706
	EB	2,640	52,800	17,960	11,225	64,025

Embedded Memory Design Guidelines for Stratix V Devices

There are several considerations that require your attention to ensure the success of your designs. Unless noted otherwise, these design guidelines apply to all variants of this device family.

Guideline: Consider the Memory Block Selection

The Intel Quartus Prime software automatically partitions the user-defined memory into the memory blocks based on your design's speed and size constraints. For example, the Intel Quartus Prime software may spread out the memory across multiple available memory blocks to increase the performance of the design.

To assign the memory to a specific block size manually, use the RAM IP core in the IP Catalog.

For the memory logic array blocks (MLAB), you can implement single-port SRAM through emulation using the Intel Quartus Prime software. Emulation results in minimal additional use of logic resources.

Because of the dual-purpose architecture of the MLAB, only data input and output registers are available in the block. The MLABs gain read address registers from the ALMs. However, the write address and read data registers are internal to the MLABs.

Guideline: Implement External Conflict Resolution

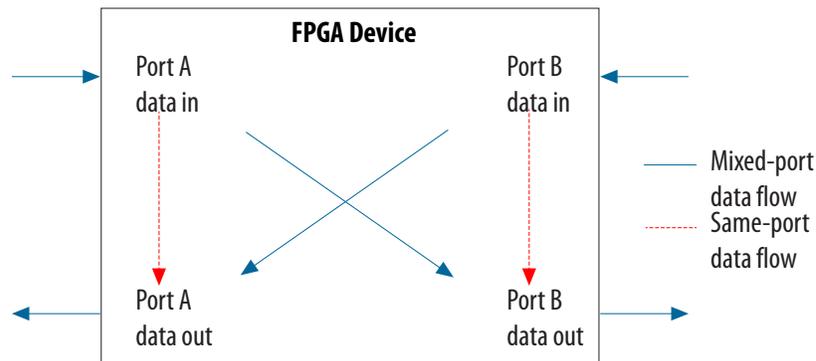
In the true dual-port RAM mode, you can perform two write operations to the same memory location. However, the memory blocks do not have internal conflict resolution circuitry. To avoid unknown data being written to the address, implement external conflict resolution logic to the memory block.

Guideline: Customize Read-During-Write Behavior

Customize the read-during-write behavior of the memory blocks to suit your design requirements.

Figure 2-1: Read-During-Write Data Flow

This figure shows the difference between the two types of read-during-write operations available—same port and mixed port.



Same-Port Read-During-Write Mode

The same-port read-during-write mode applies to a single-port RAM or the same port of a true dual-port RAM.

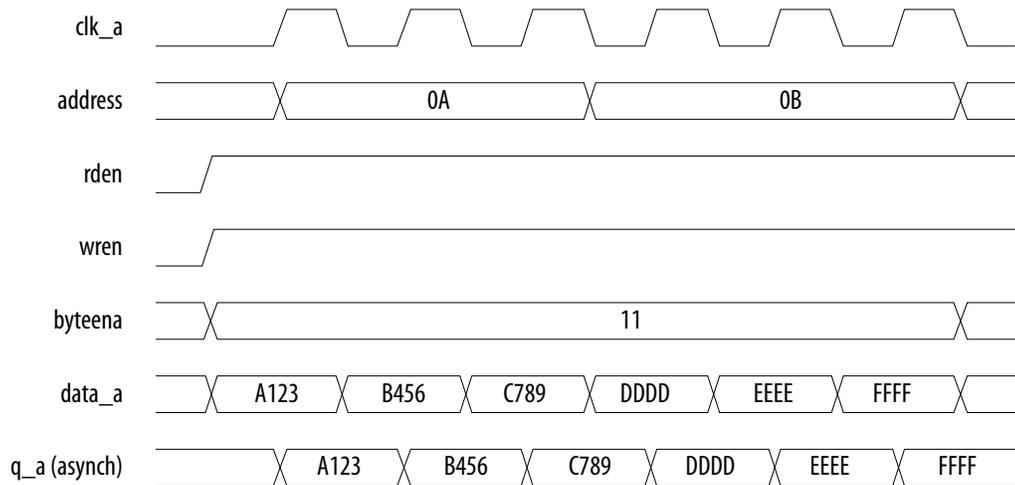
Table 2-2: Output Modes for Embedded Memory Blocks in Same-Port Read-During-Write Mode

This table lists the available output modes if you select the embedded memory blocks in the same-port read-during-write mode.

Output Mode	Memory Type	Description
"new data" (flow-through)	M20K	The new data is available on the rising edge of the same clock cycle on which the new data is written.
"don't care"	MLAB	The RAM outputs "don't care" values for a read-during-write operation.

Figure 2-2: Same-Port Read-During-Write: New Data Mode

This figure shows sample functional waveforms of same-port read-during-write behavior in the “new data” mode.



Mixed-Port Read-During-Write Mode

The mixed-port read-during-write mode applies to simple and true dual-port RAM modes where two ports perform read and write operations on the same memory address using the same clock—one port reading from the address, and the other port writing to it.

Table 2-3: Output Modes for RAM in Mixed-Port Read-During-Write Mode

Output Mode	Memory Type	Description
"new data"	MLAB	<p>A read-during-write operation to different ports causes the MLAB registered output to reflect the “new data” on the next rising edge after the data is written to the MLAB memory.</p> <p>This mode is available only if the output is registered.</p>
"old data"	M20K, MLAB	<p>A read-during-write operation to different ports causes the RAM output to reflect the “old data” value at the particular address.</p> <p>For MLAB, this mode is available only if the output is registered.</p>

Output Mode	Memory Type	Description
"don't care"	M20K, MLAB	<p>The RAM outputs “don't care” or “unknown” value.</p> <ul style="list-style-type: none"> For M20K memory, the Intel Quartus Prime software does not analyze the timing between write and read operations. For MLAB, the Intel Quartus Prime software analyzes the timing between write and read operations by default. To disable this behavior, turn on the Do not analyze the timing between write and read operation. Metastability issues are prevented by never writing and reading at the same address at the same time option.
"constrained don't care"	MLAB	<p>The RAM outputs “don't care” or “unknown” value. The Intel Quartus Prime software analyzes the timing between write and read operations in the MLAB.</p>

Figure 2-3: Mixed-Port Read-During-Write: New Data Mode

This figure shows a sample functional waveform of mixed-port read-during-write behavior for the “new data” mode.

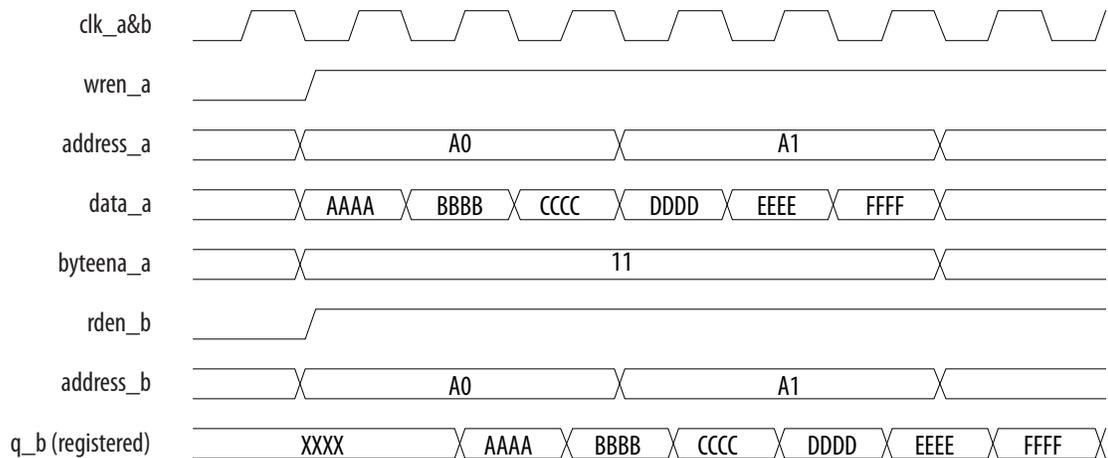
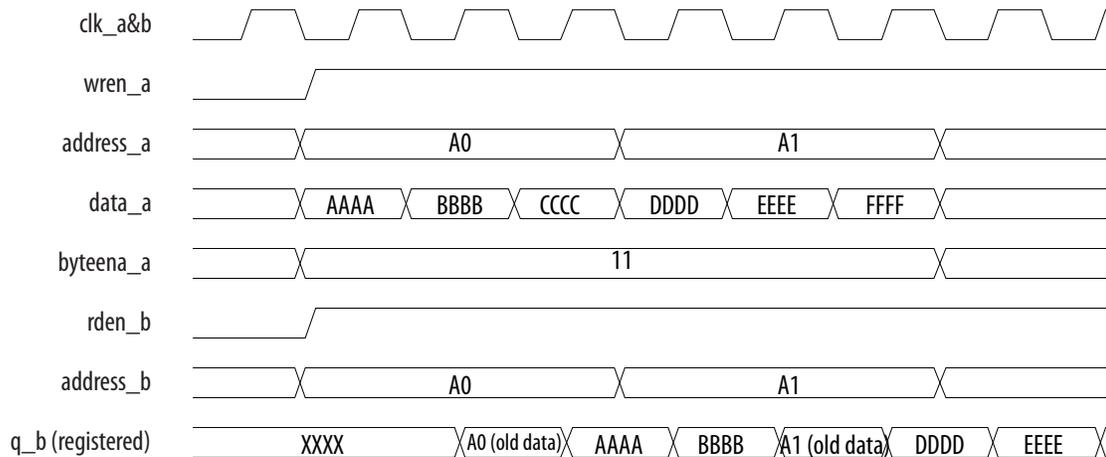
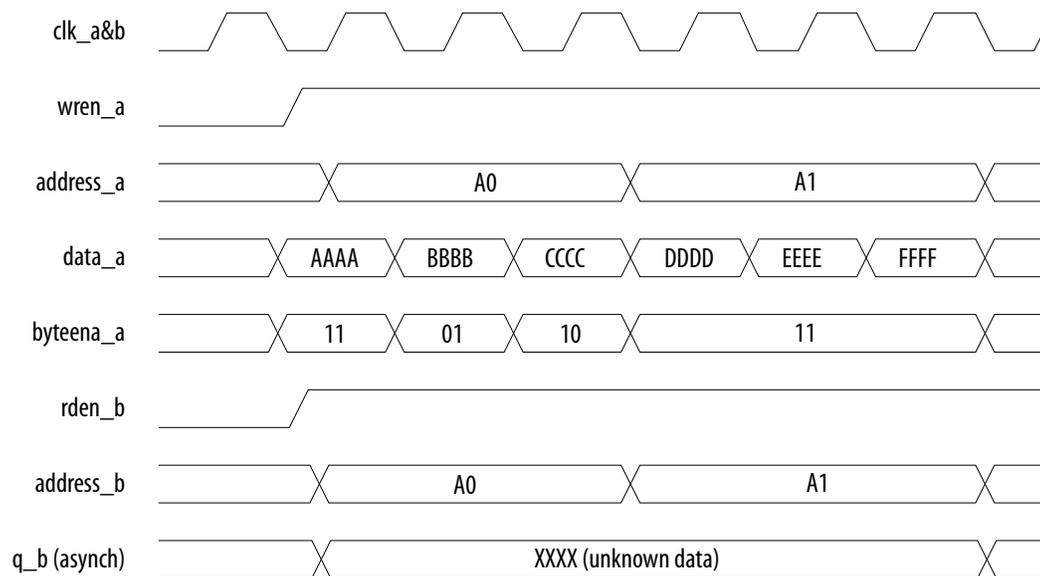


Figure 2-4: Mixed-Port Read-During-Write: Old Data Mode

This figure shows a sample functional waveform of mixed-port read-during-write behavior for the “old data” mode.

**Figure 2-5: Mixed-Port Read-During-Write: Don't Care or Constrained Don't Care Mode**

This figure shows a sample functional waveform of mixed-port read-during-write behavior for the “don't care” or “constrained don't care” mode.



In the dual-port RAM mode, the mixed-port read-during-write operation is supported if the input registers have the same clock. The output value during the operation is “unknown.”

Related Information

[Embedded Memory \(RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT\) IP Core User Guide](#)

Provides more information about the RAM IP core that controls the read-during-write behavior.

Guideline: Consider Power-Up State and Memory Initialization

Consider the power up state of the different types of memory blocks if you are designing logic that evaluates the initial power-up values, as listed in the following table.

Table 2-4: Initial Power-Up Values of Embedded Memory Blocks

Memory Type	Output Registers	Power Up Value
MLAB	Used	Zero (cleared)
	Bypassed	Read memory contents
M20K	Used	Zero (cleared)
	Bypassed	Zero (cleared)

By default, the Intel Quartus Prime software initializes the RAM cells in Stratix V devices to zero unless you specify a **.mif**.

All memory blocks support initialization with a **.mif**. You can create **.mif** files in the Intel Quartus Prime software and specify their use with the RAM IP core when you instantiate a memory in your design. Even if a memory is pre-initialized (for example, using a **.mif**), it still powers up with its output cleared.

Related Information

- [Embedded Memory \(RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT\) IP Core User Guide](#)
Provides more information about **.mif** files.
- [Quartus II Handbook](#)
Provides more information about **.mif** files.

Guideline: Control Clocking to Reduce Power Consumption

Reduce AC power consumption in your design by controlling the clocking of each memory block:

- Use the read-enable signal to ensure that read operations occur only when necessary. If your design does not require read-during-write, you can reduce your power consumption by de-asserting the read-enable signal during write operations, or during the period when no memory operations occur.
- Use the Intel Quartus Prime software to automatically place any unused memory blocks in low-power mode to reduce static power.

Embedded Memory Features

Table 2-5: Memory Features in Stratix V Devices

This table summarizes the features supported by the embedded memory blocks.

Features	M20K	MLAB
Maximum operating frequency	600 MHz	600 MHz

Features	M20K	MLAB
Capacity per block (including parity bits)	20,480	640
Parity bits	Supported	Supported
Byte enable	Supported	Supported
Packed mode	Supported	—
Address clock enable	Supported	Supported
Simple dual-port mixed width	Supported	—
True dual-port mixed width	Supported	—
FIFO buffer mixed width	Supported	—
Memory Initialization File (.mif)	Supported	Supported
Mixed-clock mode	Supported	Supported
Fully synchronous memory	Supported	Supported
Asynchronous memory	—	Only for flow-through read memory operations.
Power-up state	Output ports are cleared.	<ul style="list-style-type: none"> Registered output ports—Cleared. Unregistered output ports—Read memory contents.
Asynchronous clears	Output registers and output latches	Output registers and output latches
Write/read operation triggering	Rising clock edges	Rising clock edges
Same-port read-during-write	Output ports set to "new data".	Output ports set to "don't care".
Mixed-port read-during-write	Output ports set to "old data" or "don't care".	Output ports set to "old data", "new data", "don't care", or "constrained don't care".
ECC support	<p>Soft IP support using the Intel Quartus Prime software.</p> <p>Built-in support in x32-wide simple dual-port mode.</p>	Soft IP support using the Intel Quartus Prime software.

Related Information**[Embedded Memory \(RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT\) IP Core User Guide](#)**

Provides more information about the embedded memory features.

Embedded Memory Configurations

Table 2-6: Supported Embedded Memory Block Configurations for Stratix V Devices

This table lists the maximum configurations supported for the embedded memory blocks. The information is applicable only to the single-port RAM and ROM modes.

Memory Block	Depth (bits)	Programmable Width
MLAB	32	x16, x18, or x20
	64	x8, x9, x10
M20K	512	x40, x32
	1K	x20, x16
	2K	x10, x8
	4K	x5, x4
	8K	x2
	16K	x1

Mixed-Width Port Configurations

The mixed-width port configuration is supported in the simple dual-port RAM and true dual-port RAM memory modes.

Note: MLABs do not support mixed-width port configurations.

Related Information**[Embedded Memory \(RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT\) IP Core User Guide](#)**

Provides more information about dual-port mixed width support.

M20K Blocks Mixed-Width Configurations

The following table lists the mixed-width configurations of the M20K blocks in the simple dual-port RAM mode.

Table 2-7: M20K Block Mixed-Width Configurations (Simple Dual-Port RAM Mode)

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	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—

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
8K x 2	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
4K x 4	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
4K x 5	—	—	—	Yes	—	Yes	—	Yes	—	Yes
2K x 8	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
2K x 10	—	—	—	Yes	—	Yes	—	Yes	—	Yes
1K x 16	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
1K x 20	—	—	—	Yes	—	Yes	—	Yes	—	Yes
512 x 32	Yes	Yes	Yes	—	Yes	—	Yes	—	Yes	—
512 x 40	—	—	—	Yes	—	Yes	—	Yes	—	Yes

The following table lists the mixed-width configurations of the M20K blocks 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	Yes	Yes	Yes	—	Yes	—	Yes	—
8K x 2	Yes	Yes	Yes	—	Yes	—	Yes	—
4K x 4	Yes	Yes	Yes	—	Yes	—	Yes	—
4K x 5	—	—	—	Yes	—	Yes	—	Yes
2K x 8	Yes	Yes	Yes	—	Yes	—	Yes	—
2K x 10	—	—	—	Yes	—	Yes	—	Yes
1K x 16	Yes	Yes	Yes	—	Yes	—	Yes	—
1K x 20	—	—	—	Yes	—	Yes	—	Yes

Embedded Memory Modes

Caution: To avoid corrupting the memory contents, do not violate the setup or hold time on any of the memory block input registers during read or write operations. This is applicable if you use the memory blocks in single-port RAM, simple dual-port RAM, true dual-port RAM, or ROM mode.

Table 2-9: Memory Modes Supported in the Embedded Memory Blocks

This table lists and describes the memory modes that are supported in the Stratix V embedded memory blocks.

Memory Mode	M20K Support	MLAB Support	Description
Single-port RAM	Yes	Yes	<p>You can perform only one read or one write operation at a time.</p> <p>Use the read enable port to control the RAM output ports behavior during a write operation:</p> <ul style="list-style-type: none"> To retain the previous values that are held during the most recent active read enable—create a read-enable port and perform the write operation with the read enable port deasserted. To show the new data being written, the old data at that address, or a "Don't Care" value when read-during-write occurs at the same address location—do not create a read-enable signal, or activate the read enable during a write operation.
Simple dual-port RAM	Yes	Yes	You can simultaneously perform one read and one write operations to different locations where the write operation happens on port A and the read operation happens on port B.
True dual-port RAM	Yes	—	You can perform any combination of two port operations: two reads, two writes, or one read and one write at two different clock frequencies.
Shift-register	Yes	Yes	<p>You can use the memory blocks as a shift-register block to save logic cells and routing resources.</p> <p>This is useful in DSP applications that require local data storage such as finite impulse response (FIR) filters, pseudo-random number generators, multi-channel filtering, and auto- and cross- correlation functions. Traditionally, the local data storage is implemented with standard flip-flops that exhaust many logic cells for large shift registers.</p> <p>The input data width (w), the length of the taps (m), and the number of taps (n) determine the size of a shift register ($w \times m \times n$). You can cascade memory blocks to implement larger shift registers.</p>

Memory Mode	M20K Support	MLAB Support	Description
ROM	Yes	Yes	<p>You can use the memory blocks as ROM.</p> <ul style="list-style-type: none"> Initialize the ROM contents of the memory blocks using a .mif or .hex. The address lines of the ROM are registered on M20K blocks but can be unregistered on MLABs. The outputs can be registered or unregistered. The output registers can be asynchronously cleared. The ROM read operation is identical to the read operation in the single-port RAM configuration.
FIFO	Yes	Yes	<p>You can use the memory blocks as FIFO buffers. Use the SCFIFO and DCFIFO IP cores to implement single- and dual-clock asynchronous FIFO buffers in your design.</p> <p>For designs with many small and shallow FIFO buffers, the MLABs are ideal for the FIFO mode. However, the MLABs do not support mixed-width FIFO mode.</p>

Related Information

- [Embedded Memory \(RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT\) IP Core User Guide](#)
Provides more information memory modes.
- [RAM-Based Shift Register \(ALTSHIFT_TAPS\) IP Core User Guide](#)
Provides more information about implementing the shift register mode.
- [SCFIFO and DCFIFO IP Cores User Guide](#)
Provides more information about implementing FIFO buffers.

Embedded Memory Clocking Modes

This section describes the clocking modes for the Stratix V memory blocks.

Caution: To avoid corrupting the memory contents, do not violate the setup or hold time on any of the memory block input registers during read or write operations.

Clocking Modes for Each Memory Mode

Table 2-10: Memory Blocks Clocking Modes Supported for Each Memory Mode

Clocking Mode	Memory Mode				
	Single-Port	Simple Dual-Port	True Dual-Port	ROM	FIFO
Single clock mode	Yes	Yes	Yes	Yes	Yes
Read/write clock mode	—	Yes	—	—	Yes

Clocking Mode	Memory Mode				
	Single-Port	Simple Dual-Port	True Dual-Port	ROM	FIFO
Input/output clock mode	Yes	Yes	Yes	Yes	—
Independent clock mode	—	—	Yes	Yes	—

Note: The clock enable signals are not supported for write address, byte enable, and data input registers on MLAB blocks.

Single Clock Mode

In the single clock mode, a single clock, together with a clock enable, controls all registers of the memory block.

Read/Write Clock Mode

In the read/write clock mode, a separate clock is available for each read and write port. A read clock controls the data-output, read-address, and read-enable registers. A write clock controls the data-input, write-address, write-enable, and byte enable registers.

Input/Output Clock Mode

In input/output clock mode, a separate clock is available for each input and output port. 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.

Independent Clock Mode

In the independent clock mode, a separate clock is available for each port (A and B). Clock A controls all registers on the port A side; clock B controls all registers on the port B side.

Note: You can create independent clock enable for different input and output registers to control the shut down of a particular register for power saving purposes. From the parameter editor, click **More Options** (beside the clock enable option) to set the available independent clock enable that you prefer.

Asynchronous Clears in Clocking Modes

In all clocking modes, asynchronous clears are available only for output latches and output registers. For the independent clock mode, this is applicable on both ports.

Output Read Data in Simultaneous Read/Write

If you perform a simultaneous read/write to the same address location using the read/write clock mode, the output read data is unknown. If you require the output read data to be a known value, use single-clock or input/output clock mode and select the appropriate read-during-write behavior in the IP Catalog.

Note: MLAB memory blocks only support simultaneous read/write operations when operating in single clock mode.

Independent Clock Enables in Clocking Modes

Independent clock enables are supported in the following clocking modes:

- Read/write clock mode—supported for both the read and write clocks.
- Independent clock mode—supported for the registers of both ports.

To save power, you can control the shut down of a particular register using the clock enables.

Related Information

Guideline: Control Clocking to Reduce Power Consumption on page 2-7

Parity Bit in Memory Blocks

Table 2-11: Parity Bit Support for the Embedded Memory Blocks

This table describes the parity bit support for the memory blocks.

M20K	MLAB
<ul style="list-style-type: none"> • The parity bit is the fifth bit associated with each 4 data bits in data widths of 5, 10, 20, and 40 (bits 4, 9, 14, 19, 24, 29, 34, and 39). • In non-parity data widths, the parity bits are skipped during read or write operations. • Parity function is not performed on the parity bit. 	<ul style="list-style-type: none"> • The parity bit is the ninth bit associated with each byte. • The ninth bit can store a parity bit or serve as an additional bit. • Parity function is not performed on the parity bit.

Byte Enable in Embedded Memory Blocks

The embedded memory blocks support byte enable controls:

- The byte enable controls mask the input data so that only specific bytes of data are written. The unwritten bytes retain the values written previously.
- The write enable (`wren`) signal, together with the byte enable (`byteena`) signal, control the write operations on the RAM blocks. By default, the `byteena` signal is high (enabled) and only the `wren` signal controls the writing.
- The byte enable registers do not have a `clear` port.
- If you are using parity bits, on the M20K blocks, the byte enable function controls 8 data bits and 2 parity bits; on the MLABs, the byte enable function controls all 10 bits in the widest mode.
- The LSB of the `byteena` signal corresponds to the LSB of the data bus.
- The byte enables are active high.

Byte Enable Controls in Memory Blocks

Table 2-12: byteena Controls in x20 Data Width

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

Table 2-13: 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]

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

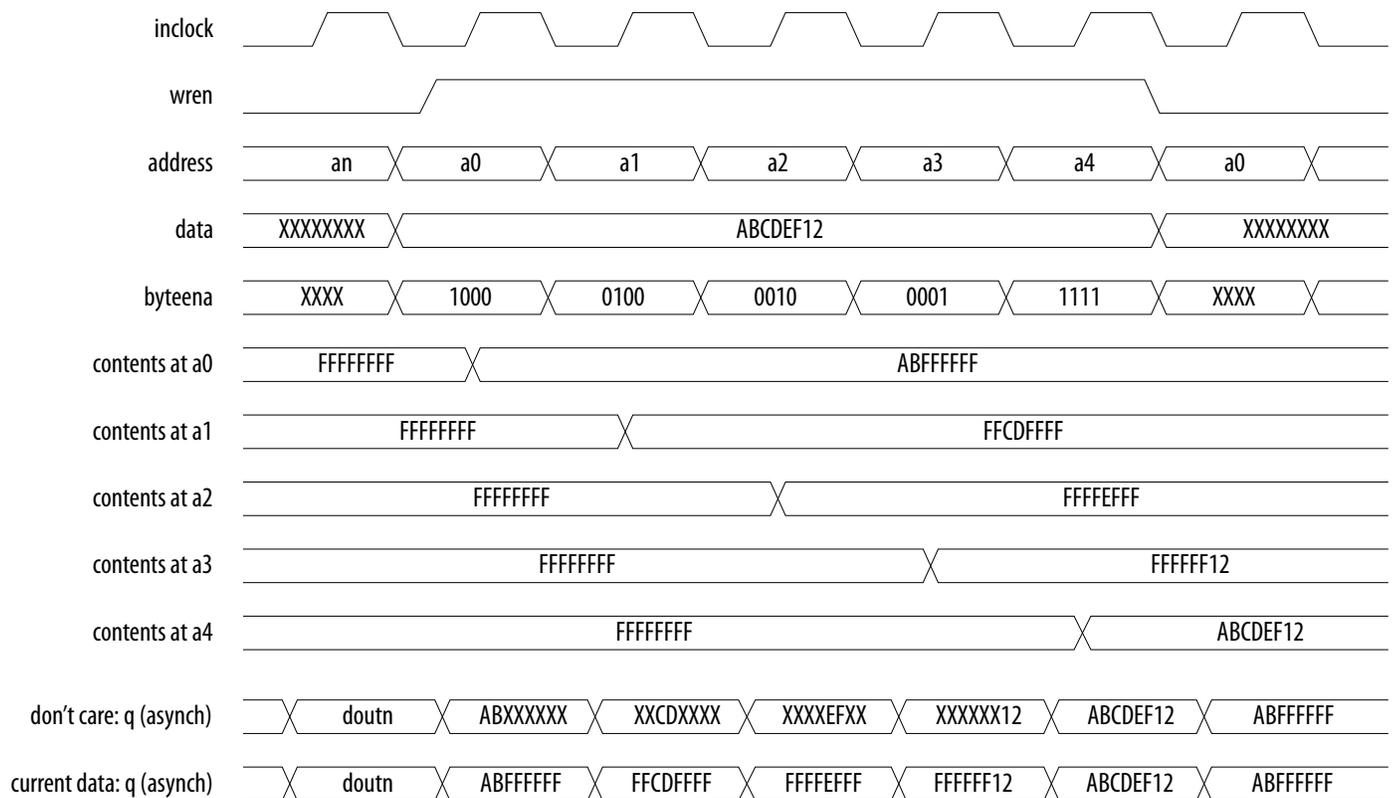
Data Byte Output

In M20K blocks or MLABs, when you de-assert a byte-enable bit during a write cycle, the corresponding data byte output appears as either a “don't care” value or the current data at that location. You can control the output value for the masked byte in the M20K blocks or MLABs by using the Intel Quartus Prime software.

RAM Blocks Operations

Figure 2-6: Byte Enable Functional Waveform

This figure shows how the `wren` and `byteena` signals control the operations of the RAM blocks.



Memory Blocks Packed Mode Support

The M20K memory blocks support packed mode.

The packed mode feature packs two independent single-port RAM blocks into one memory block. The Intel Quartus Prime 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 RAM blocks. The size of each independent single-port RAM must not exceed half of the target block size.

Memory Blocks Address Clock Enable Support

The 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 2-7: Address Clock Enable

This figure shows an address clock enable block diagram. The address clock enable is referred to by the port name `addresstall`.

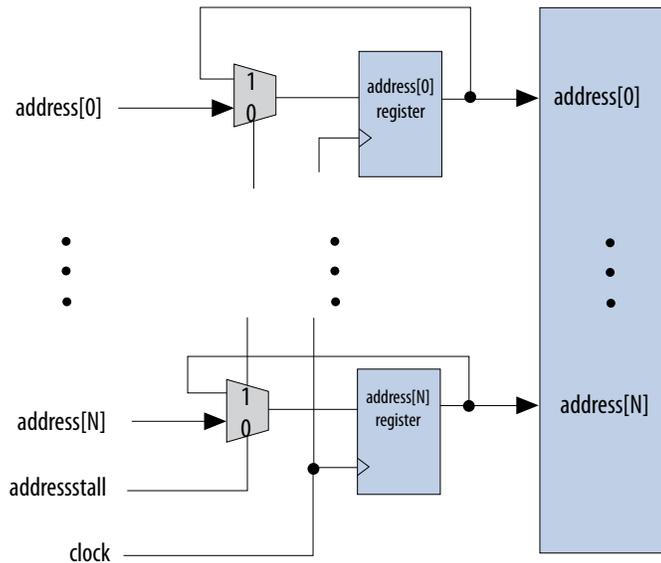


Figure 2-8: Address Clock Enable During Read Cycle Waveform

This figure shows the address clock enable waveform during the read cycle.

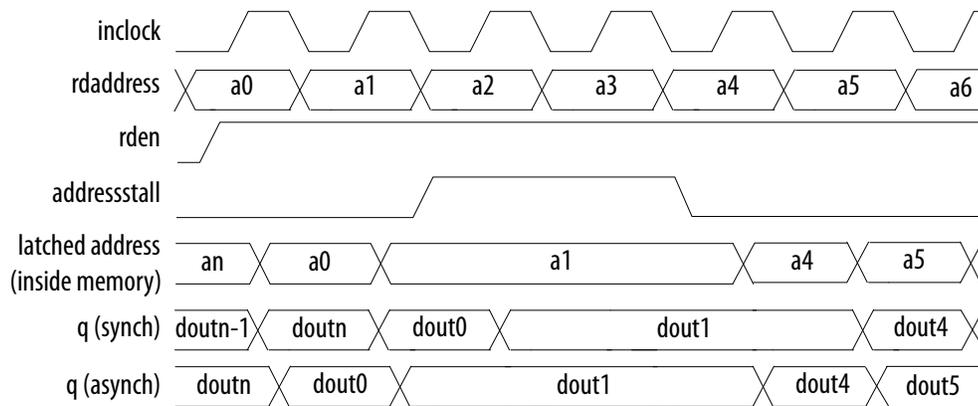
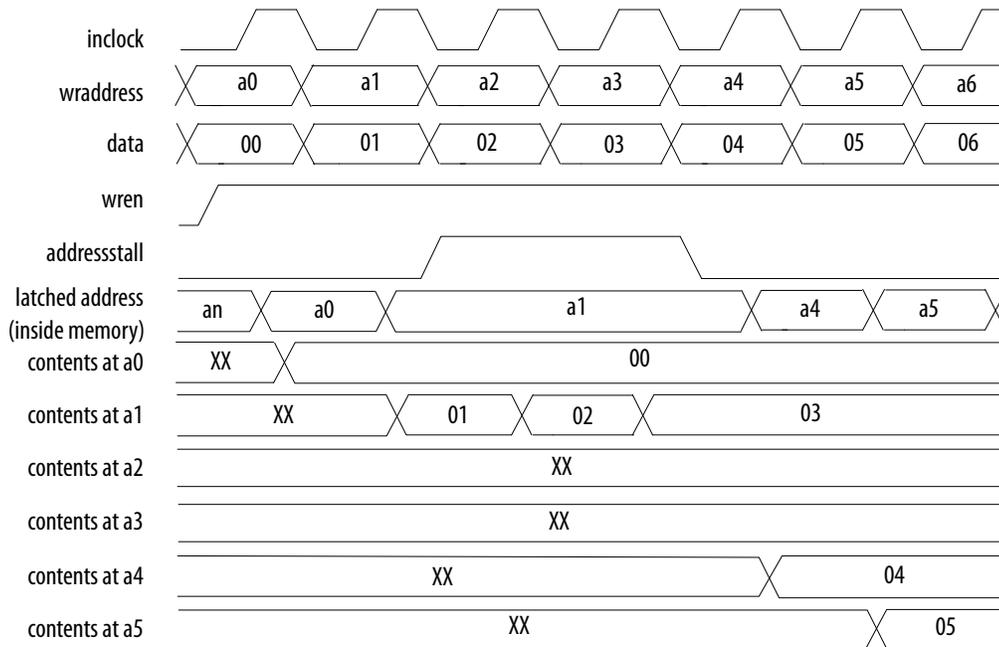


Figure 2-9: Address Clock Enable During the Write Cycle Waveform

This figure shows the address clock enable waveform during the write cycle.

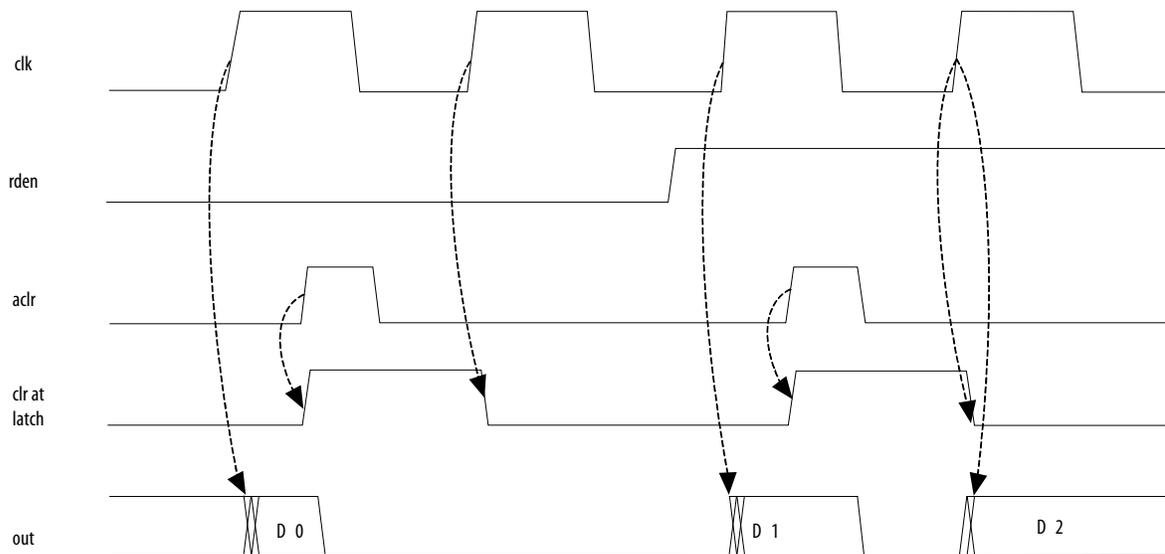


Memory Blocks Asynchronous Clear

The M20K memory blocks support asynchronous clear on output latches and output registers. If your RAM does not use output registers, clear the RAM outputs using the output latch asynchronous clear.

The clear is an asynchronous signal and 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 2-10: Output Latch Clear in Stratix V Devices



Memory Blocks Error Correction Code Support

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 blocks have built-in support for ECC when in x32-wide simple dual-port mode:

- 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 the ECC status flag replaces them.

Error Correction Code Truth Table

Table 2-14: ECC Status Flags Truth Table

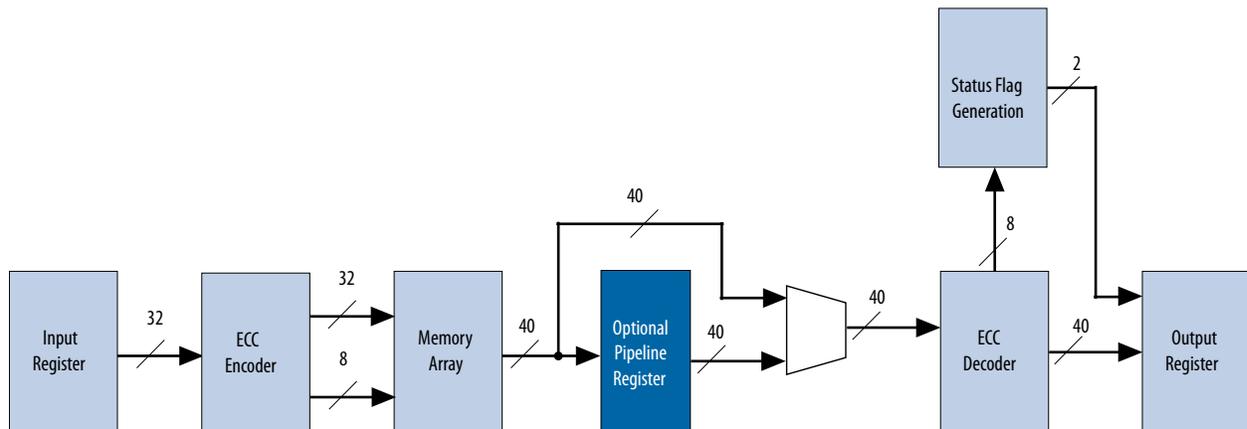
e (error) eccstatus[1]	ue (uncorrectable error) eccstatus[0]	Status
0	0	No error.
0	1	Illegal.
1	0	A correctable error occurred and the error has been corrected at the outputs; however, the memory array has not been updated.

e (error) eccstatus[1]	ue (uncorrectable error) eccstatus[0]	Status
1	1	An uncorrectable error occurred and uncorrectable data appears at the outputs.

If you engage ECC:

- You cannot use the byte enable feature.
- Read-during-write old data mode is not supported.

Figure 2-11: ECC Block Diagram for M20K Memory



Embedded Memory Blocks in Stratix V Devices Revision History

Date	Version	Changes
December 2015	2015.12.21	Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i> .
January 2015	2015.01.23	<ul style="list-style-type: none"> • Reword Total RAM bits in Memory Features in Stratix V Devices table to Capacity per Block.
June 2014	2014.06.30	Removed the term "one-hot" fashion for byte enables operation. The term one-hot indicates that only one bit can be active at a time.

Date	Version	Changes
May 2013	2013.05.06	<ul style="list-style-type: none"> • Moved all links to the Related Information section of respective topics for easy reference. • Added link to the known document issues in the Knowledge Base. • Corrected the description about the "don't care" output mode for RAM in mixed-port read-during-write. • Reorganized the structure of the supported memory configurations topics (single-port and mixed-width dual-port) to improve clarity about maximum data widths supported for each configuration. • Added a description to the table listing the maximum embedded memory configurations to clarify that the information applies only to the single port or ROM mode. • Removed the topic about mixed-width configurations for MLABs and added a note to clarify that MLABs do not support mixed-width configuration.
December 2012	2012.12.28	<ul style="list-style-type: none"> • Reorganized content and updated template. • Updated memory capacity information for accuracy (kilobits instead of megabits). • Moved information about supported memory block configurations into its own table. • Removed some information that is available in the Internal Memory (RAM and ROM) User Guide.
June 2012	1.4	Updated Table 2–1 and Table 2–2.
November 2011	1.3	<ul style="list-style-type: none"> • Updated Table 2–1 and Table 2–2. • Updated “Mixed-Port Read-During-Write Mode” section.
May 2011	1.2	<ul style="list-style-type: none"> • 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.

2020.09.10

SV51004



Subscribe



Send Feedback

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.

Related Information

[Stratix V Device Handbook: Known Issues](#)

Lists the planned updates to the Stratix V Device Handbook chapters.

Features

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

The Stratix V variable precision DSP blocks offer the following features:

- High-performance, power-optimized, and fully registered multiplication operations
- 9-bit, 18-bit, 27-bit, and 36-bit word lengths
- 18 x 25 complex multiplications for FFTs
- Floating-point arithmetic formats
- Built-in addition, subtraction, and 64-bit accumulation unit to combine multiplication results
- Cascading 18-bit and 27-bit input bus to form the 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
- Supports 18-bit and 27-bit with internal coefficient register bank for filter implementation
- 18-bit and 27-bit systolic finite impulse response (FIR) filters with distributed output adder

Related Information

[Stratix V Device Overview](#)

Provides more information about the number of multipliers in each Stratix V device.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

Supported Operational Modes in Stratix V Devices

Table 3-1: 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

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

Variable Precision DSP Block Resources	Operational Mode	Supported Instance	Pre-adder Support	Coefficient Support	Input Cascade Support	Chainout Support
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

Resources

Table 3-2: Number of Multipliers in Stratix V Devices

The table lists the variable-precision DSP resources by bit precision for each Stratix V device.

Variant	Member Code	Variable-precision DSP Block	Independent Input and Output Multiplications Operator					18 x 18 Multiplier Adder Mode	18 x 18 Multiplier Summed with 36-bit Input
			9 x 9 Multiplier	16 x 16 Multiplier	18 x 18 Multiplier with 32-bit Resolution	27 x 27 Multiplier	36 x 18 Multiplier		
Stratix V GX	A3	256	768	512	512	256	256	512	256
	A4	256	768	512	512	256	256	512	256
	A5	256	768	512	512	256	256	512	256
	A7	256	768	512	512	256	256	512	256
	A9	352	1,056	704	704	352	352	704	352
	AB	352	1,056	704	704	352	352	704	352
	B5	399	1,197	798	798	399	399	798	399
	B6	399	1,197	798	798	399	399	798	399
	B9	352	1,056	704	704	352	352	704	352
	BB	352	1,056	704	704	352	352	704	352
Stratix V GT	C5	256	768	512	512	256	256	512	256
	C7	256	768	512	512	256	256	512	256
Stratix V GS	D3	600	1,800	1,200	1,200	600	600	1,200	600
	D4	1,044	3,132	2,088	2,088	1,044	1,044	2,088	1,044
	D5	1,590	4,770	3,180	3,180	1,590	1,590	3,180	1,590
	D6	1,775	5,325	3,550	3,550	1,775	1,775	3,550	1,775
	D8	1,963	5,889	3,926	3,926	1,963	1,963	3,926	1,963
Stratix V E	E9	352	1,056	704	704	352	352	704	352
	EB	352	1,056	704	704	352	352	704	352

Design Considerations

You should consider the following elements in your design:

- Operational modes
- Internal coefficient and pre-adder
- Accumulator
- Chainout adder

Operational Modes

The Intel Quartus Prime software includes IP cores that you can use to control the operation mode of the multipliers. After entering the parameter settings with the IP Catalog, the Intel Quartus Prime software automatically configures the variable precision DSP block.

Altera provides two methods for implementing various modes of the Stratix V variable precision DSP block in a design—using the Intel Quartus Prime DSP IP cores and HDL inferring.

The following Intel Quartus Prime IP cores are supported for the Stratix V variable precision DSP blocks implementation:

- LPM_MULT
- ALTERA_MULT_ADD
- ALTMULT_COMPLEX
- ALTMEMMULT

Related Information

- [Introduction to Altera IP Cores](#)
- [Integer Arithmetic IP Cores User Guide](#)
- [Floating-Point IP Cores User Guide](#)
- [Quartus II Software Help](#)

Internal Coefficient and Pre-Adder

Mode	Stratix V
18-bit	The coefficient feature must be enabled when the pre-adder feature is enabled.
27-bit	The coefficient feature and pre-adder feature can be used independently. With pre-adder enabled: <ul style="list-style-type: none">• If the multiplicand input comes from dynamic input due to input limitations—the input data width is restricted to 22 bits.• If the multiplicand input comes from the internal coefficients—the data width of the multiplicand is 27 bits.

Note: When you enable the pre-adder feature, all input data must have the same clock setting.

Accumulator

The accumulator feature is applicable to the following modes:

- One sum of two 18 x 18 multipliers
- 27 x 27 independent multiplier
- 36 x 18 independent multiplier
- 18 x 18 multiplication summed with 36-bit input mode
- Sum of square mode

Chainout Adder

You can use the output chaining path to add results from other DSP blocks.

Block Architecture

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

- Input register bank
- Pre-adder
- Internal coefficient
- Multipliers
- Accumulator and chainout adder
- Systolic registers
- Output register bank

Figure 3-1: Variable Precision DSP Block Architecture in 18 x 18 Mode for Stratix V Devices

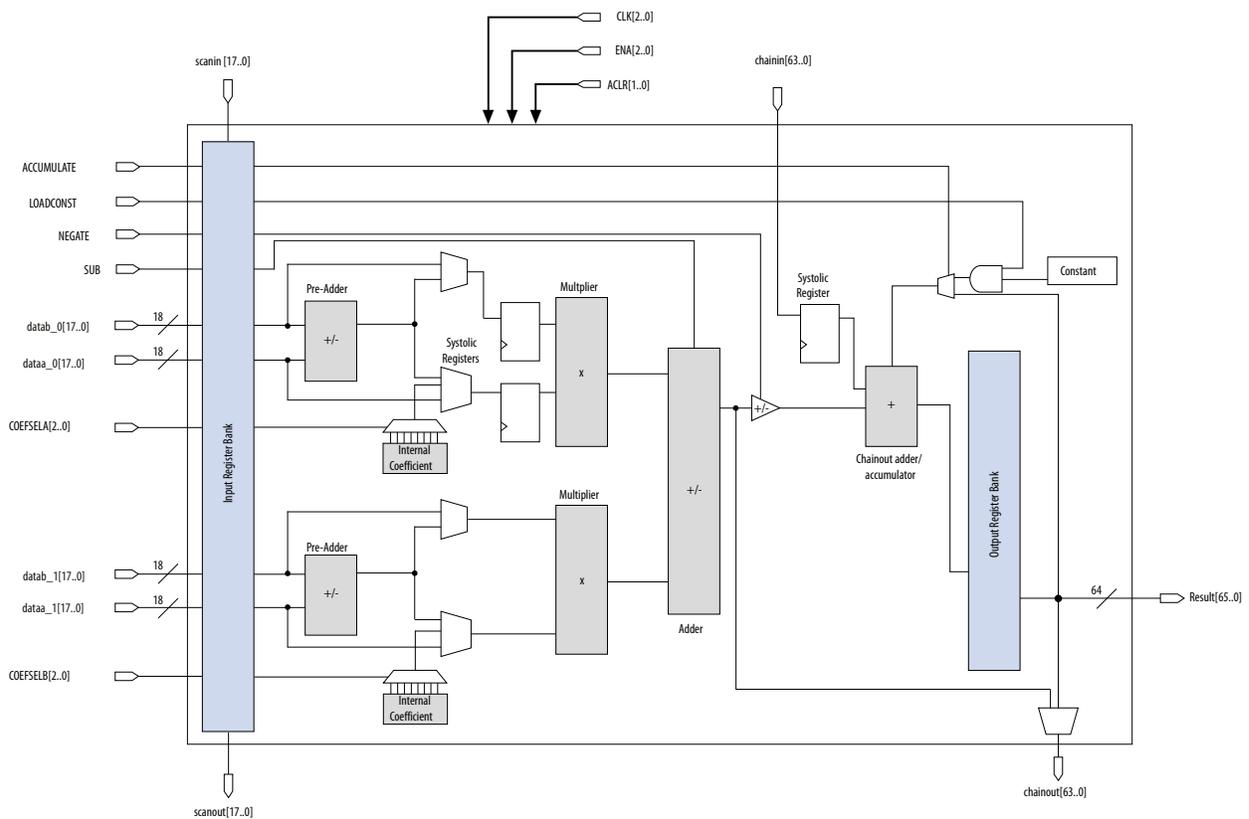
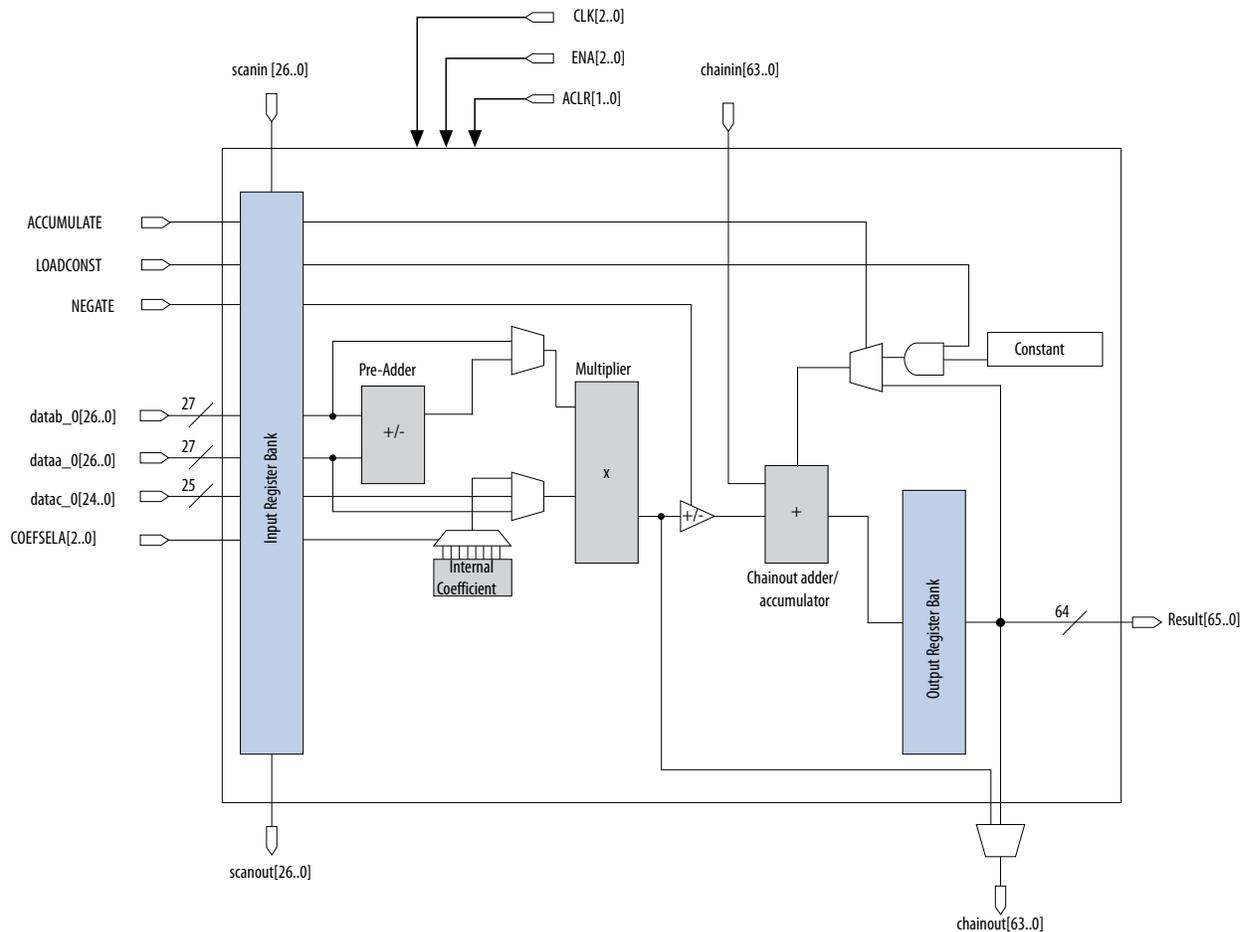


Figure 3-2: Variable Precision DSP Block Architecture in 27 x 27 Mode for Stratix V Devices



Input Register Bank

The input register bank consists of data, dynamic control signals, and two sets of delay registers.

All the registers in the DSP blocks are positive-edge triggered and cleared on 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]

In 18 x 18 mode, you can use the delay registers to balance the latency requirements when you use both the input cascade and chainout features.

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 the following figures. The Stratix V variable precision DSP block supports 18-bit and 27-bit input cascading.

Figure 3-3: Input Register of a Variable Precision DSP Block in 18 x 18 Mode for Stratix V Devices

The figures show the data registers only. Registers for the control signals are not shown.

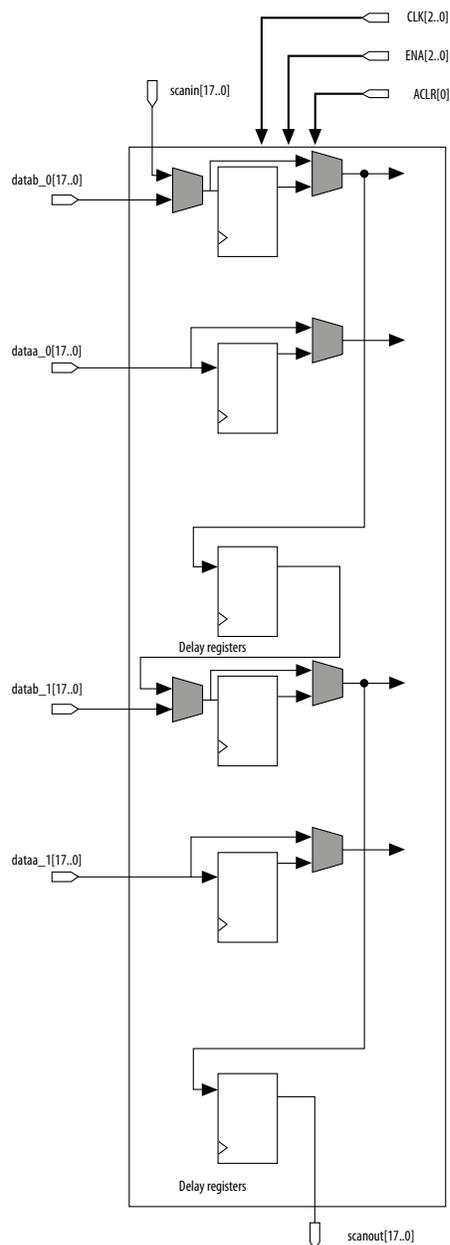
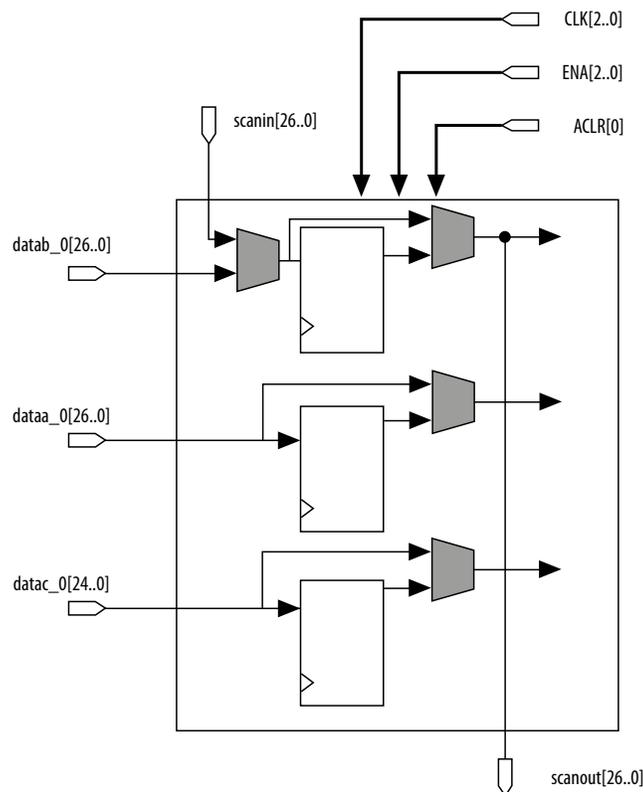


Figure 3-4: Input Register of a Variable Precision DSP Block in 27x 27 Mode for Stratix V Devices

The figures show the data registers only. Registers for the control signals are not shown.



Pre-Adder

Stratix V Devices

The pre-adder supports both addition and subtraction, which you must choose during compilation time.

Each variable precision DSP block has two 18-bit pre-adders. You can configure these pre-adders in the following configurations:

- Two independent 18-bit adders for 18-bit applications
- One 26-bit adder for 27-bit applications

Internal Coefficient

The Stratix V variable precision DSP block has the flexibility of selecting the multiplicand from either the dynamic input or the internal coefficient.

The internal coefficient can support up to eight constant coefficients for the multiplicands in 18-bit and 27-bit modes. When you enable the internal coefficient feature, `COEFSELA/COEFSELB` are used to control the selection of the coefficient multiplexer.

Multipliers

A single variable precision DSP block can perform many multiplications in parallel, depending on the data width of the multiplier.

There are two multipliers (upper multiplier and bottom multiplier) per variable precision DSP block. You can configure these two multipliers in several operational modes:

- One 27 x 27 multiplier
- Two 18 x 18 multipliers
- Three 9 x 9 multipliers

Related Information

[Operational Mode Descriptions](#) on page 3-11

Provides more information about the operational modes of the multipliers.

Accumulator and Chainout Adder

The Stratix V variable precision DSP block supports a 64-bit accumulator and a 64-bit adder.

For Stratix V devices, you can use the 64-bit adder as full adder.

The following signals can dynamically control the function of the accumulator:

- NEGATE
- LOADCONST
- ACCUMULATE

Table 3-3: Accumulator Functions and Dynamic Control Signals for 64-Bit Accumulator in Stratix V Devices

Function	Description	NEGATE	LOADCONST	ACCUMULATE
Zeroing	Disables the accumulator.	0	0	0
Preload	Loads an initial value to the accumulator. Only one 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.	0	1	0
Accumulation	Adds the current result to the previous accumulate result.	0	0	1
Decimation	This function takes the current result, converts it into two's complement, and adds it to the previous result.	1	0	1

Systolic Registers

There are two systolic registers per variable precision DSP block. If the variable precision DSP block is not configured in systolic FIR mode, both systolic registers are bypassed.

The first systolic register has two 18-bit registers that are used to register the upper multiplier's two 18-bit inputs. You must clock these registers with the same clock source as the output register bank.

The second set of systolic registers are used to delay the chainout output to the next variable precision DSP block.

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 an Stratix V variable precision DSP block to efficiently support the following operational modes:

- Independent Multiplier Mode
- Independent Complex Multiplier Mode
- Multiplier Adder Sum Mode
- Sum of Square Mode
- 18 x 18 Multiplication Summed with 36-Bit Input Mode
- Systolic FIR Mode

Independent Multiplier Mode

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

You can configure each variable precision DSP block multiplier for 9-, 16-, 18-, 27-Bit, or 36 x 18 multiplication.

For some operational modes, the unused inputs require zero padding.

Table 3-4: Variable Precision DSP Block Independent Multiplier Mode Configurations

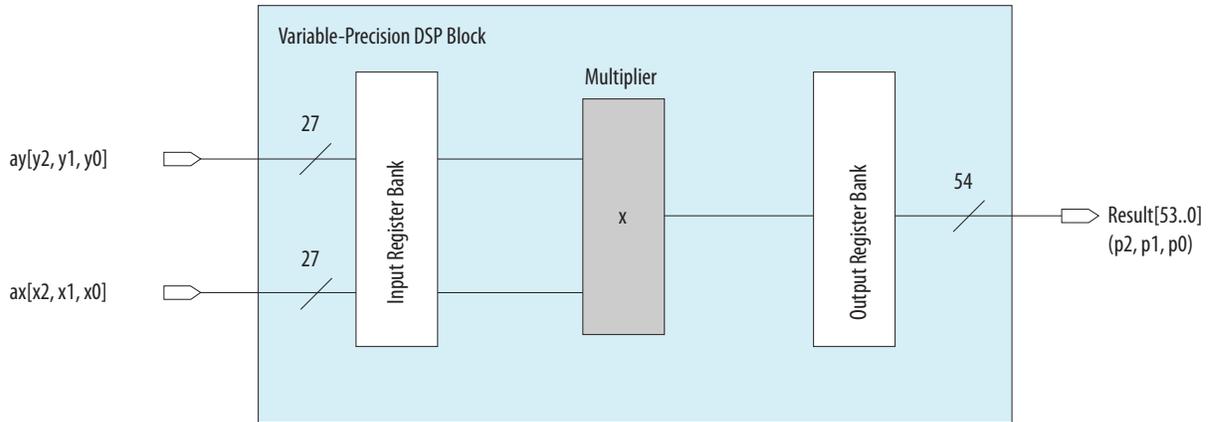
Configuration	Multipliers per block
9 x 9	3
16 x 16	2
18 x 18 (partial)	2

Configuration	Multipliers per block
18 x 18	1
27 x 27	1
36 x 18	1

9 x 9 Independent Multiplier

Figure 3-5: Three 9 x 9 Independent Multiplier Mode per Variable Precision DSP Block for Stratix V Devices

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



18 x 18 Independent Multiplier

Figure 3-6: One 18 x 18 Independent Multiplier Mode with One Variable Precision DSP Block for Stratix V Devices

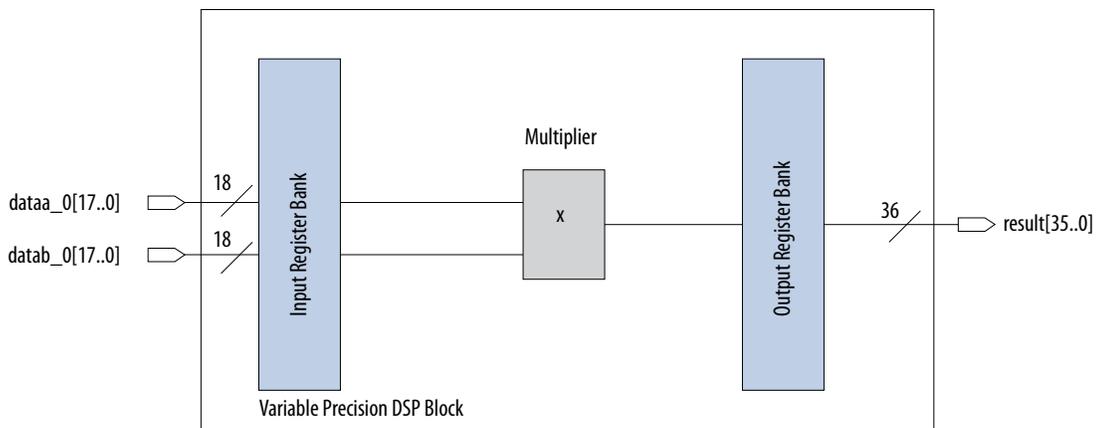
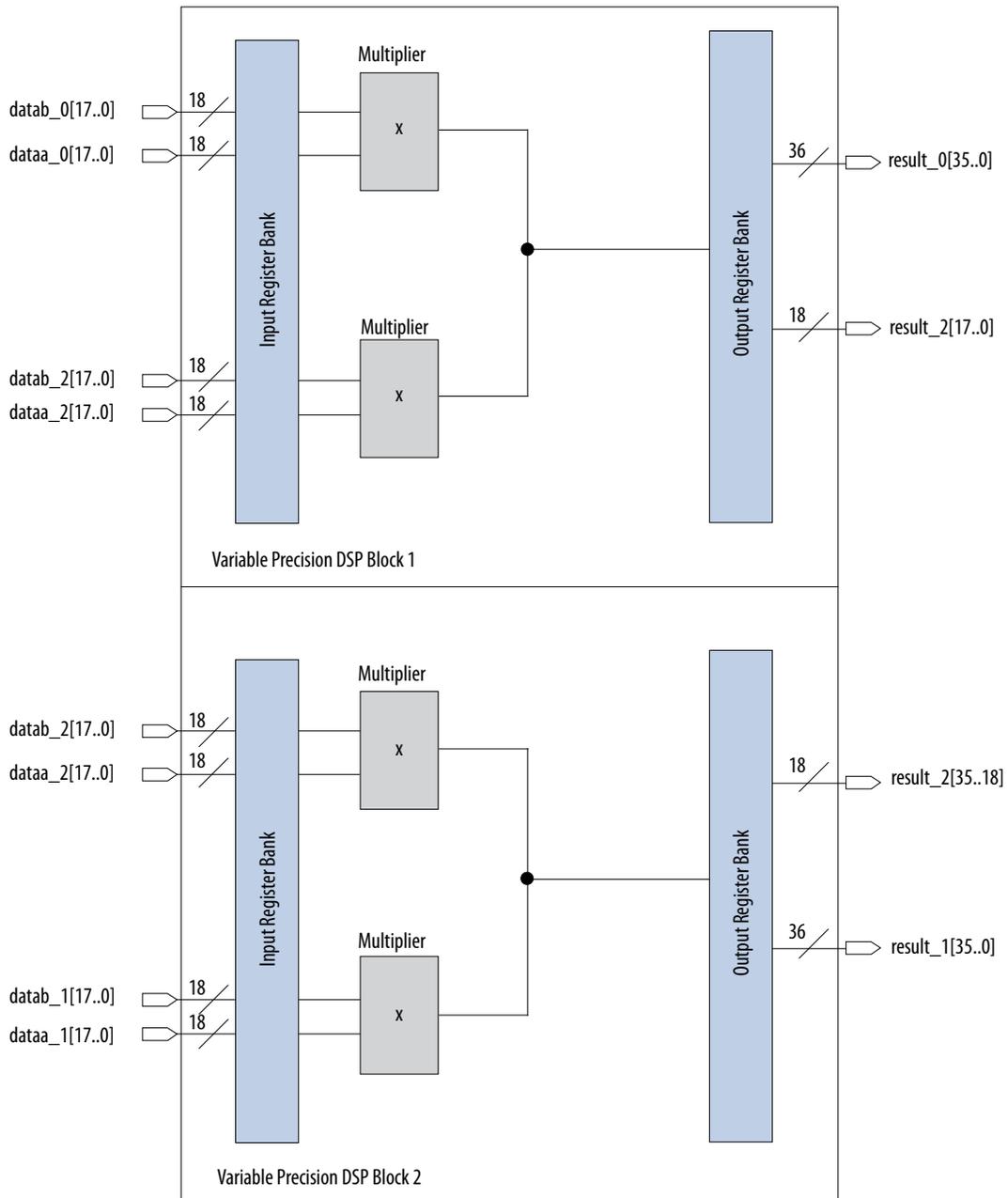


Figure 3-7: Three 18 x 18 Independent Multiplier Mode with Two Variable Precision DSP Blocks for Stratix V Devices

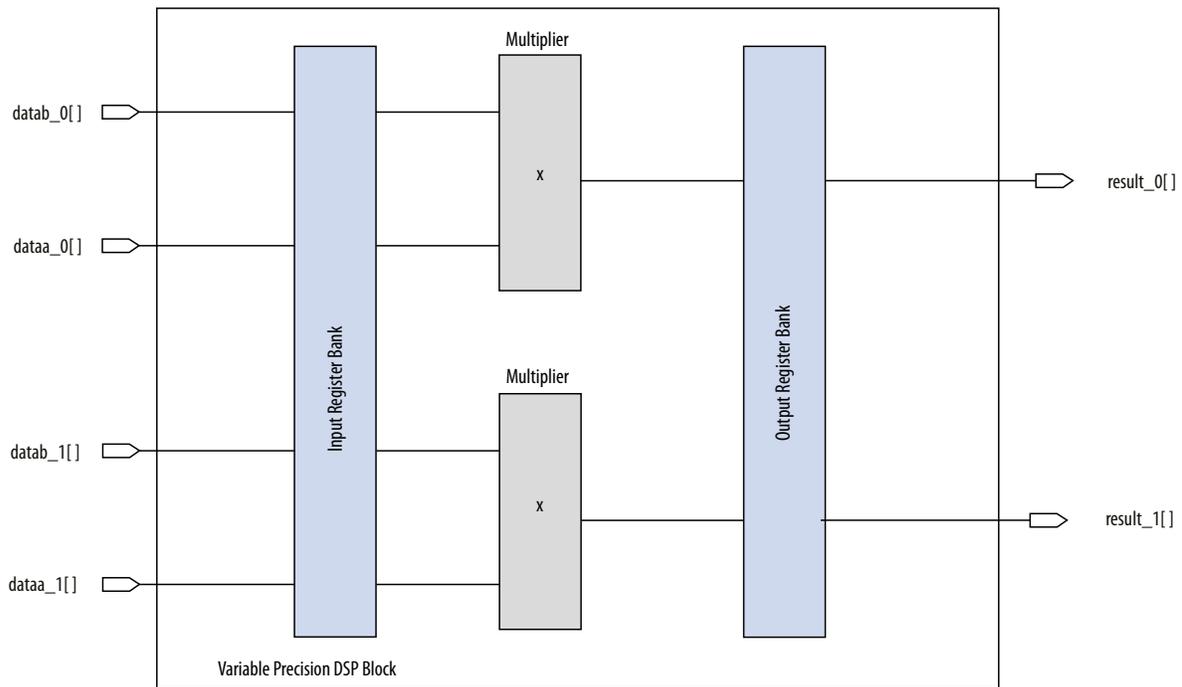


16 x 16 Independent Multiplier or 18 x 18 Independent Partial Multiplier

Figure 3-8: Two 16 x 16 Independent Multiplier Mode or Two 18 x 18 Independent Partial Multiplier Mode for Stratix V Devices

In this figure, the inputs for 16-bit independent multiplier mode are $data[15..0]$. The unused input bits require padding with zero.

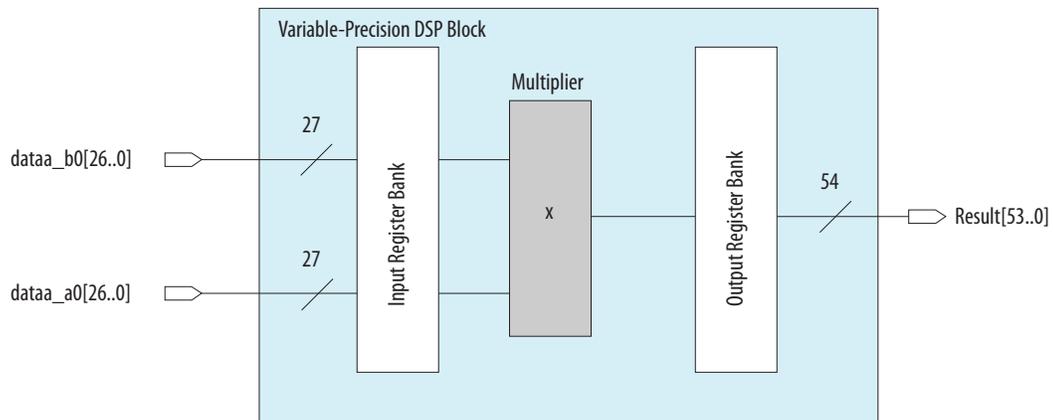
For two independent 18 x 18 partial multiplier mode, only 32-bit LSB result for each multiplication operation is routed to the output.



27 x 27 Independent Multiplier

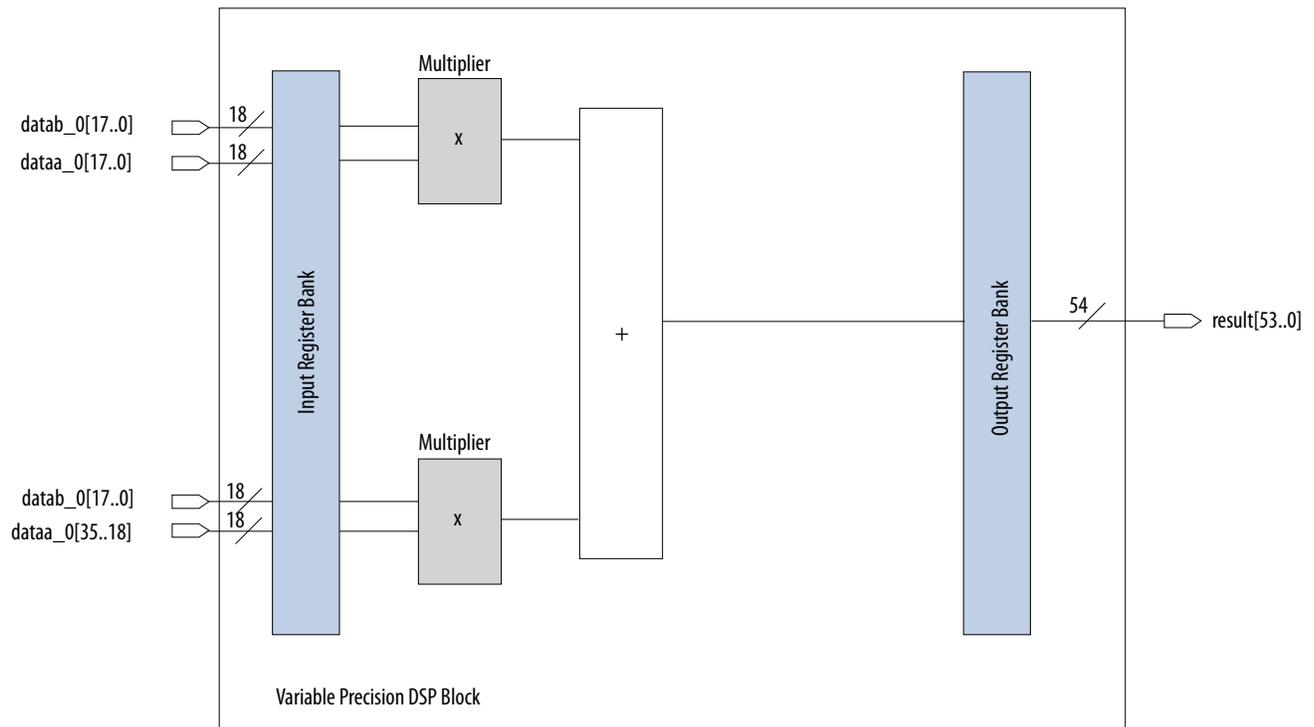
Figure 3-9: One 27 x 27 Independent Multiplier Mode per Variable Precision DSP Block for Stratix V Devices

In this mode, the `result` can be up to 64 bits when combined with a chainout adder or accumulator.



36 x 18 Independent Multiplier

Figure 3-10: One 36 x 18 Independent Multiplier Mode for Stratix V Devices

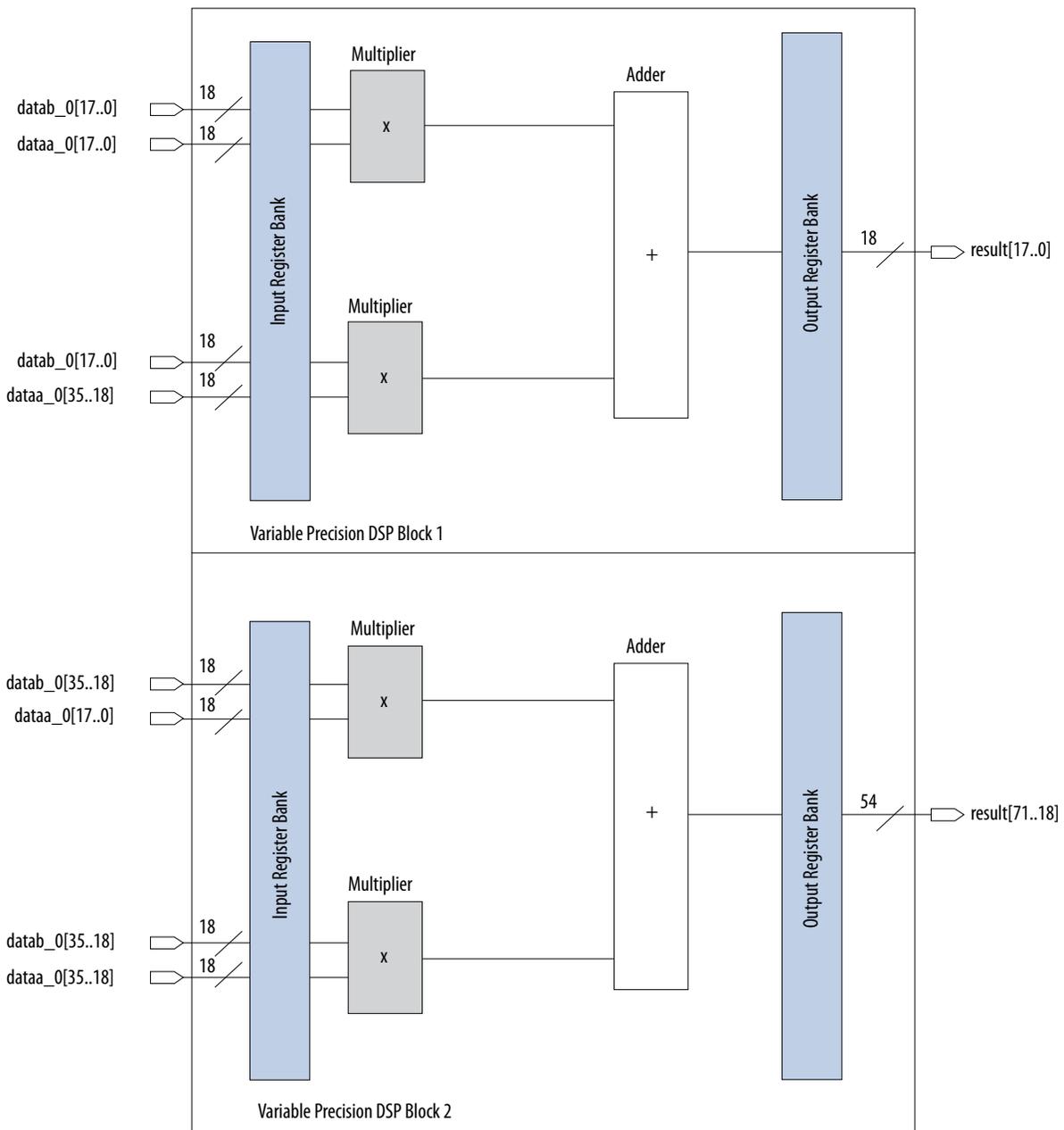


36-Bit Independent Multiplier

You can efficiently construct an individual 36-bit multiplier with two adjacent variable precision DSP blocks. The 36 x 36 multiplication consists of four 18 x 18 multipliers, as shown in [Figure 3-11](#).

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 3-11: 36-Bit Independent Multiplier Mode with Two Variable Precision DSP Blocks for Stratix V Devices



Independent Complex Multiplier Mode

The Stratix V variable precision DSP block provides the means for a complex multiplication.

Figure 3-12: Sample of 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 the following:

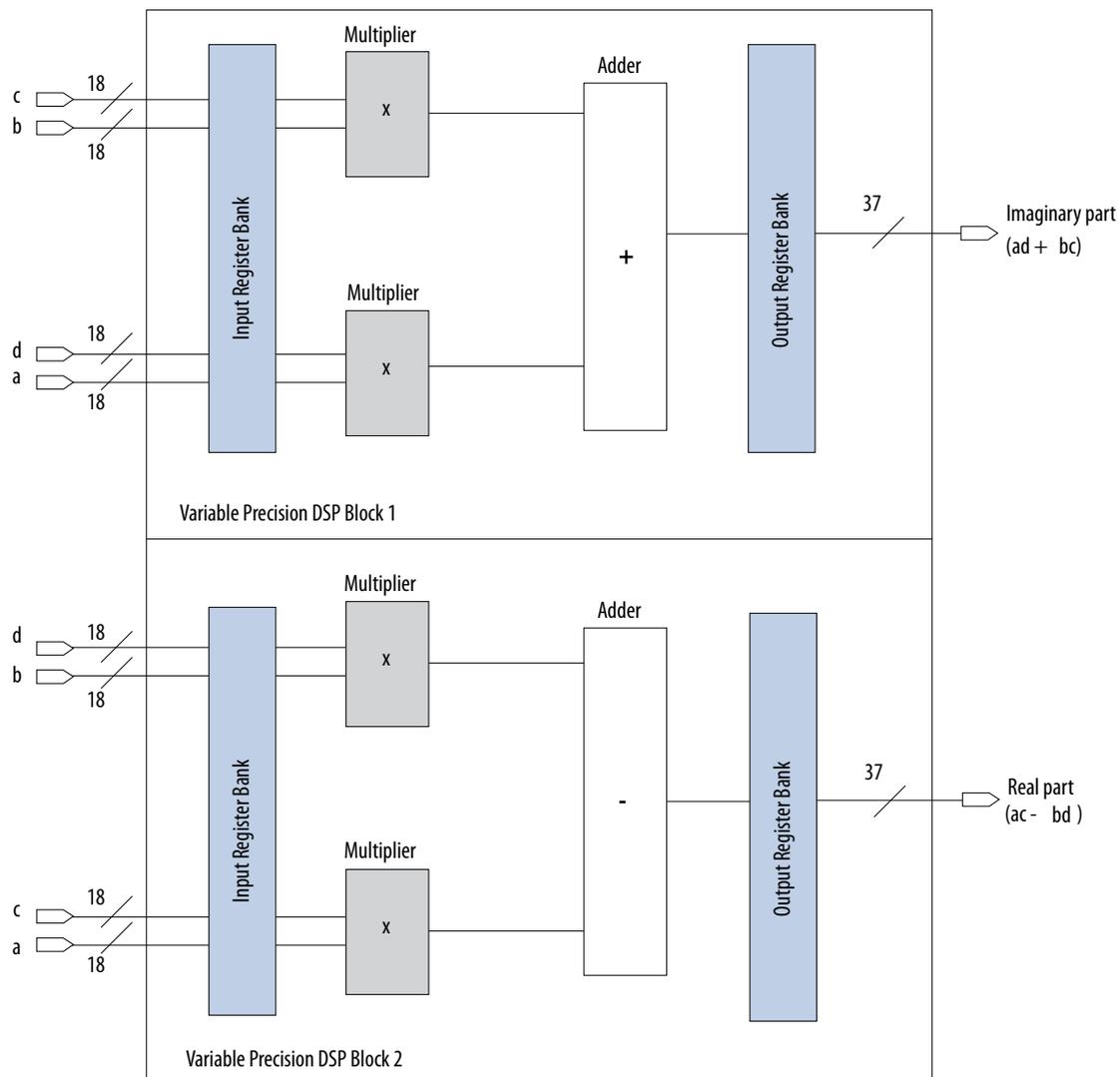
- one 18 x 18 complex multiplier
- one 18 x 25 complex multiplier
- one 27 x 27 complex multiplier

18 x 18 Complex Multiplier

For 18 x 18 complex multiplication mode, you require two variable precision DSP blocks to perform this multiplication.

You can implement the imaginary part $[(a \times d) + (b \times c)]$ in the first variable precision DSP block, and you can implement the real part $[(a \times c) - (b \times d)]$ in the second variable precision DSP block.

Figure 3-13: 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.

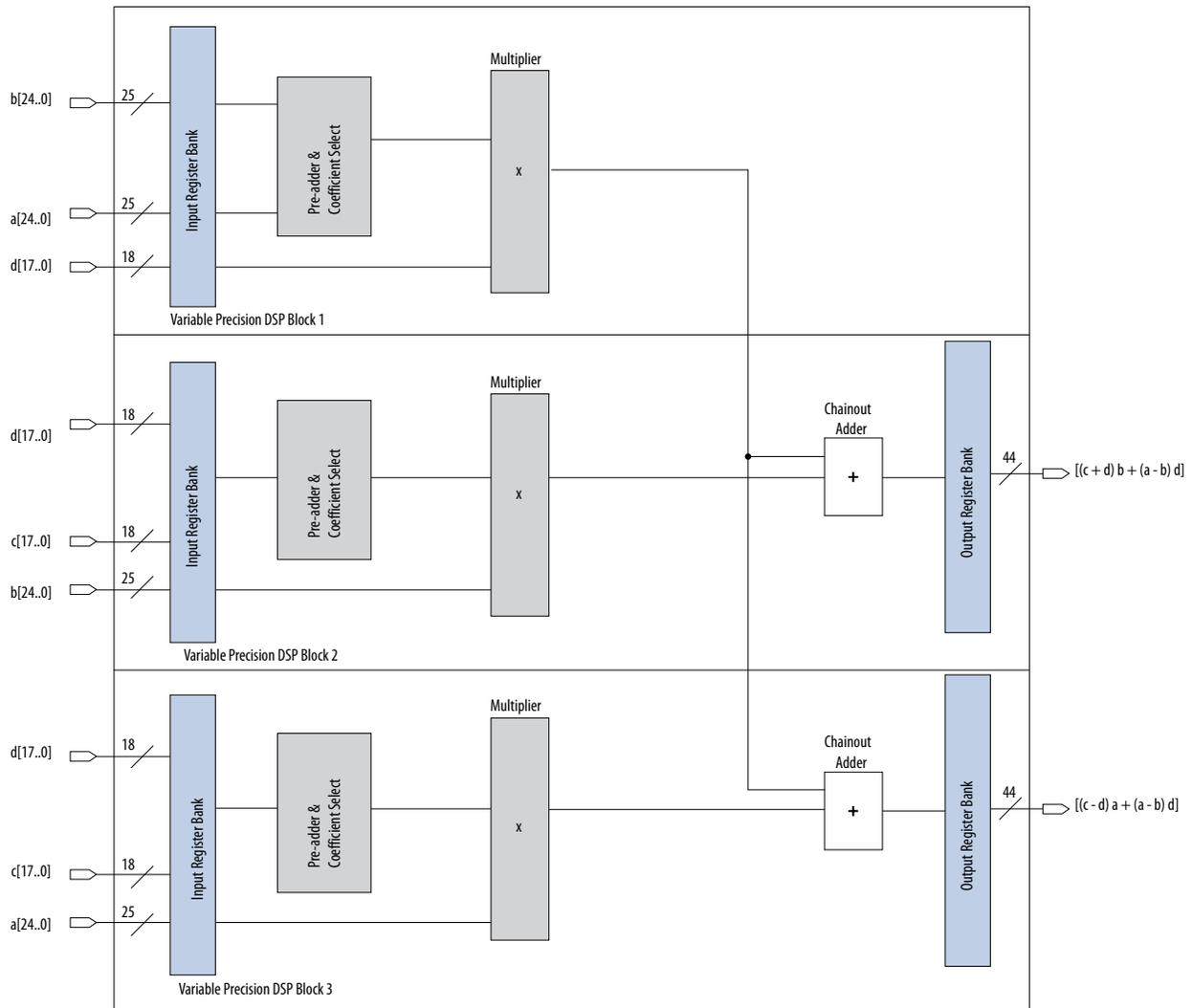
A 27 x 27 multiplier allows you to implement an individual 18 x 25 complex multiplication mode with three variable precision DSP blocks only. The pre-adder feature is automatically enabled for you to implement an individual 18 x 25 complex multiplication mode efficiently.

You can implement an 18 x 25 complex multiplication with three variable precision DSP blocks, as shown in [Figure 3-14](#)

Figure 3-14: 18 x 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 3-15: 18 x 25 Complex Multiplier with Three Variable Precision DSP Blocks for Stratix V Devices



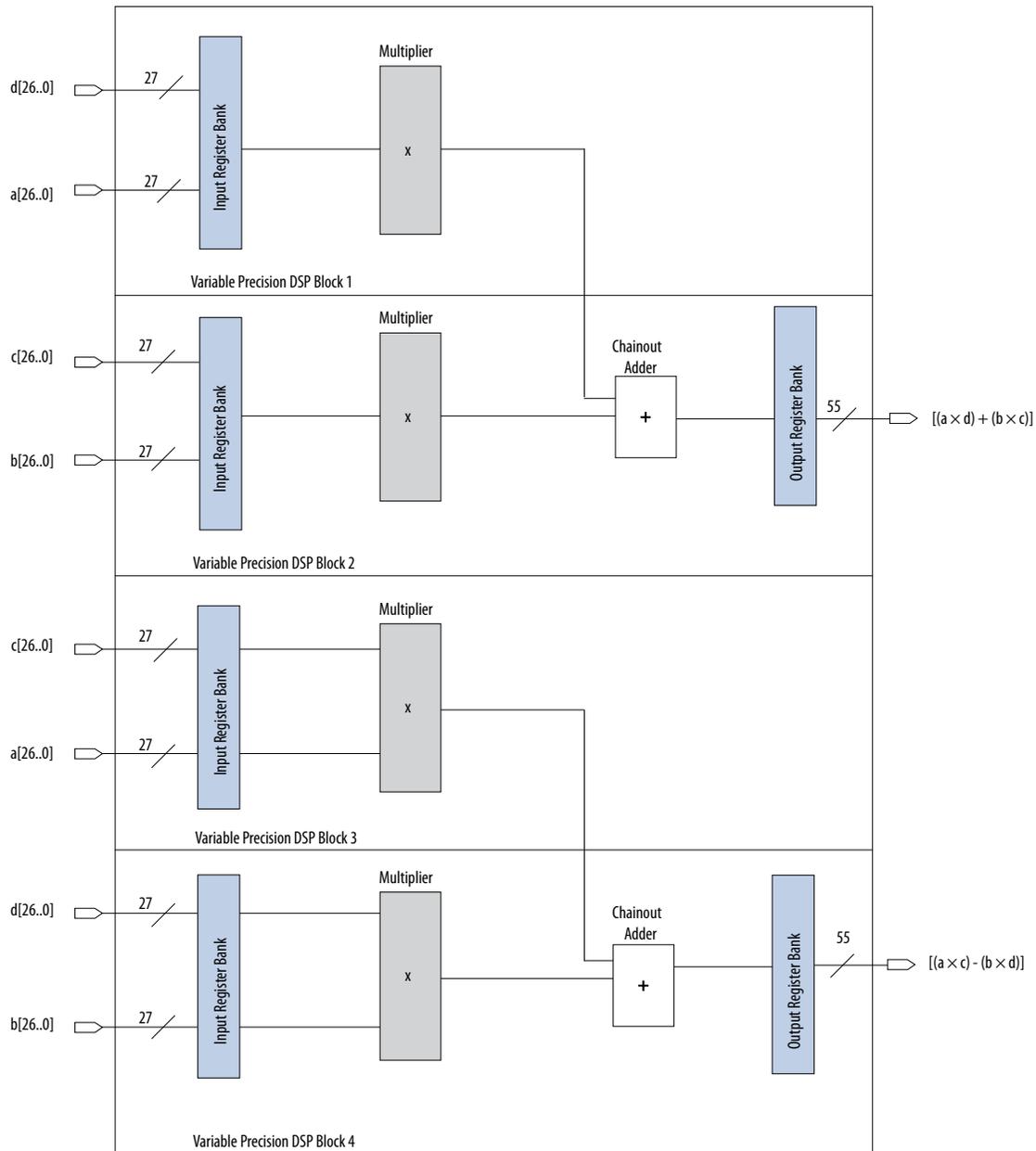
27 x 27 Complex Multiplier

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 $[(a \times d) + (b \times c)]$ in the first and second variable precision DSP blocks, and you can implement the real part $[(a \times c) - (b \times d)]$ in the third and fourth variable precision DSP blocks.

You can achieve the difference of two 27 x 27 multiplications by enabling the `NEGATE` control signal in the fourth variable precision DSP block.

Figure 3-16: 27 x 27 Complex Multiplier with Four Variable Precision Blocks for Stratix V Devices



Multiplier Adder Sum Mode

Table 3-5: Variable Precision DSP Block Multiplier Adder Sum Mode Configurations for Stratix V Devices

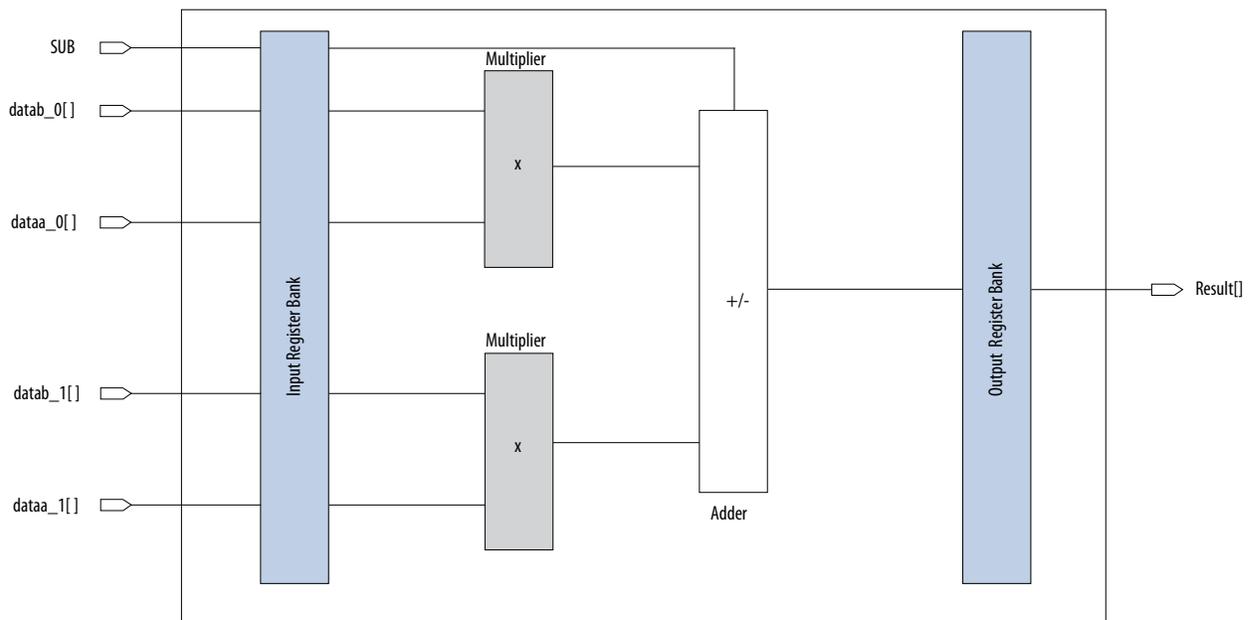
Mode	Configuration	Number of DSP Blocks Required
Two-multiplier Adder Sum	16 x 16	1
	18 x 18	1
	27 x 27	2
	18 x 36	2
Four-multiplier Adder Sum	18 x 18	2

One Sum of Two 18 x 18 Multipliers or Two 16 x 16 Multipliers

Figure 3-17: One Sum of Two 18 x 18 Multipliers or Two 16 x 16 Multipliers with One Variable Precision DSP Block for Stratix V Devices

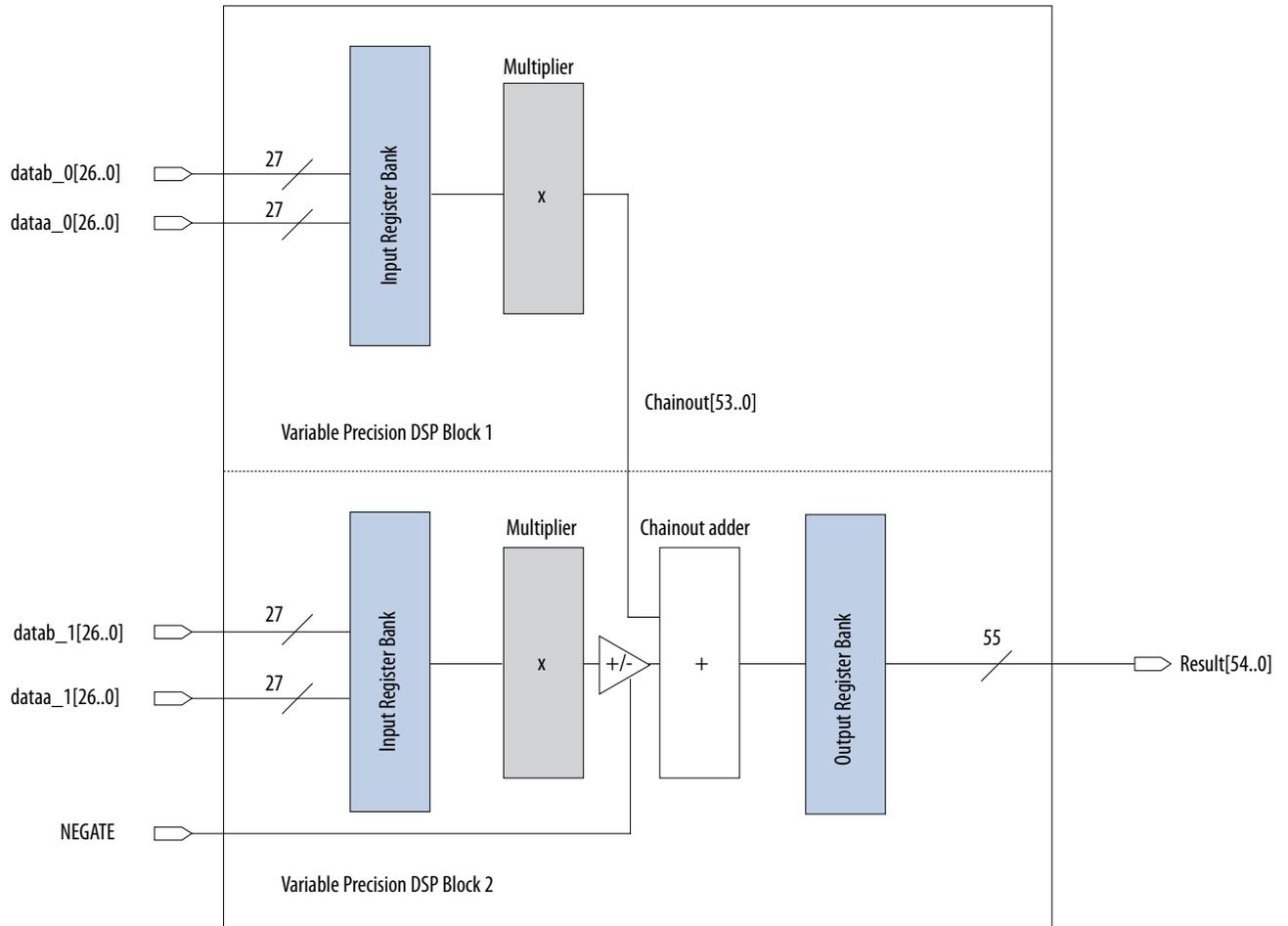
In this figure, for 18-bit multiplier adder sum mode, the input data width is 18 bits and the output data width is 37 bits.

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.



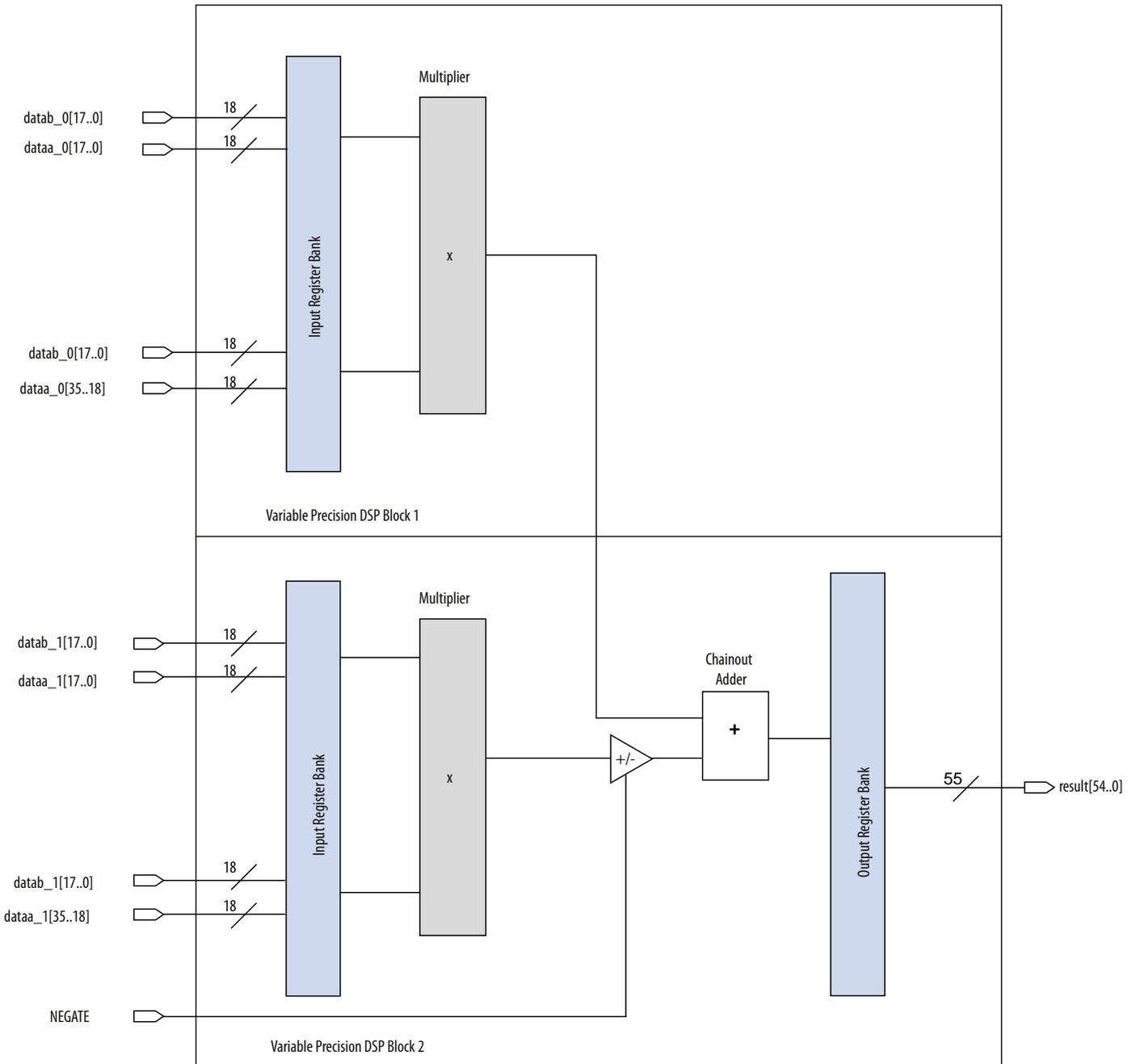
One Sum of Two 27 x 27 Multipliers

Figure 3-18: One Sum of Two 27 x 27 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices



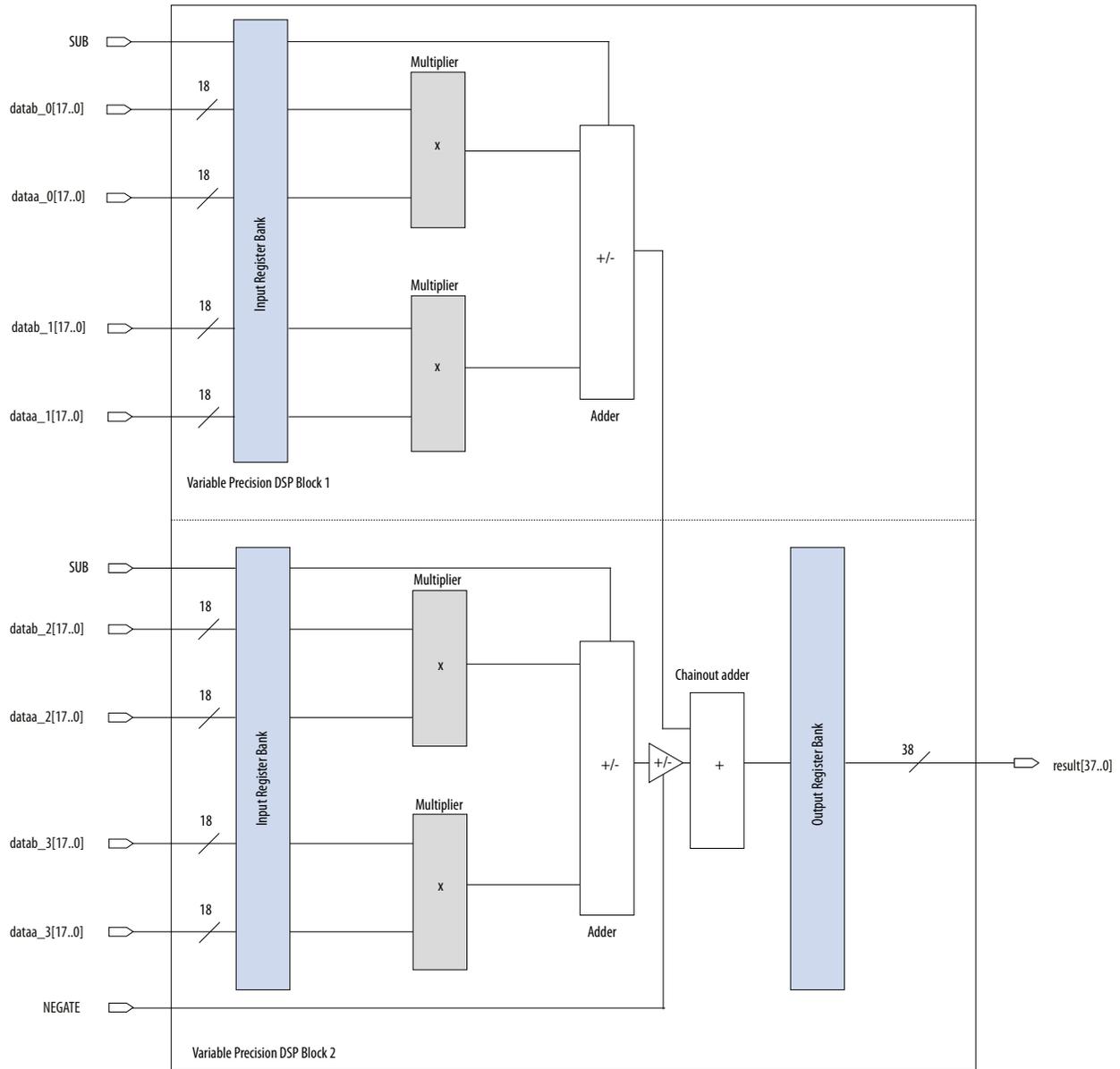
One Sum of Two 36 x 18 Multipliers

Figure 3-19: One Sum of Two 36 x 18 Multipliers with Two Variable Precision DSP Blocks for Stratix V Devices



One Sum of Four 18 x 18 Multipliers

Figure 3-20: 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.

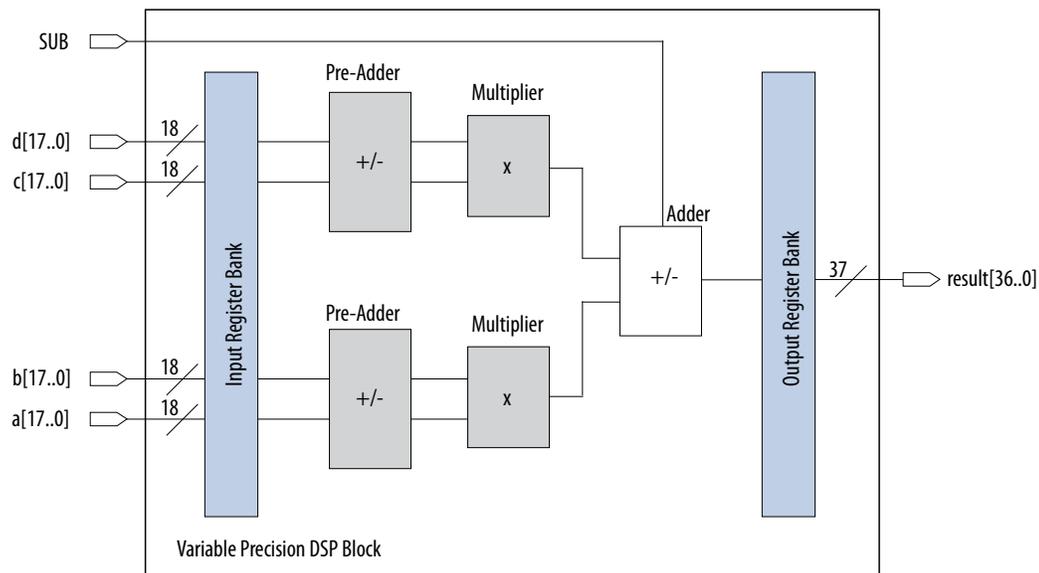
Figure 3-21: One Sum of Square Mode Equation

$$(a \pm b)^2 \times (c \pm d)^2$$

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 3-22: One Sum of Square Mode in a Variable Precision DSP Block for Stratix V Devices

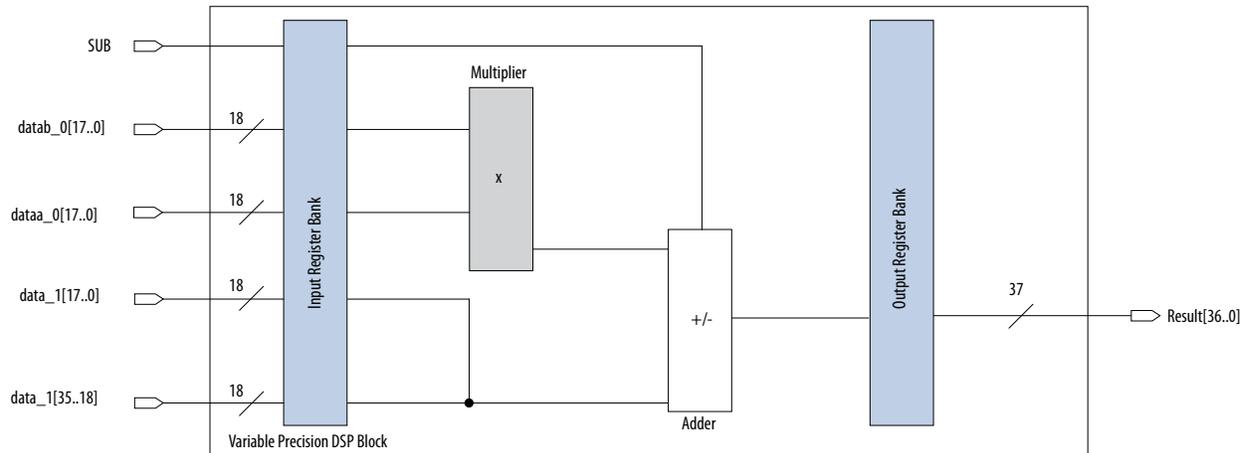


18 x 18 Multiplication Summed with 36-Bit Input Mode

Stratix V variable precision DSP blocks support one 18 x 18 multiplication summed to a 36-bit input.

Use the upper multiplier to provide the input for an 18 x 18 multiplication, while the bottom multiplier is bypassed.

The `data1[17..0]` and `data1[35..18]` signals are concatenated to produce a 36-bit input.

Figure 3-23: One 18 x 18 Multiplication Summed with 36-Bit Input Mode for Stratix V Devices

Systolic FIR Mode

Stratix V variable precision DSP blocks support the following systolic FIR structures:

- 18-bit
- 27-bit

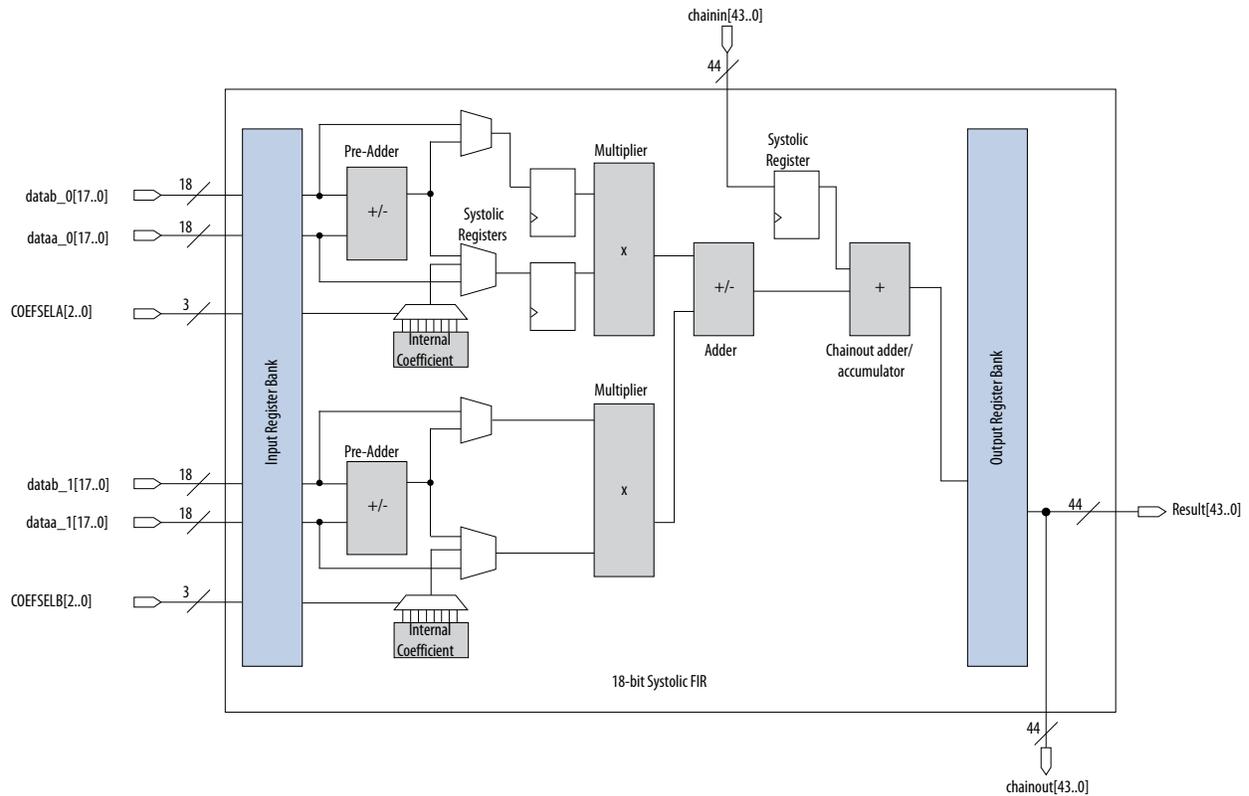
In systolic FIR mode, the input of the multiplier can come from three different sets of sources:

- Two dynamic inputs
- One dynamic input and one coefficient input
- One coefficient input and one pre-adder output

18-Bit Systolic FIR Mode

In 18-bit systolic FIR mode, the adders are configured as dual 44-bit adders, thereby giving 8 bits of overhead when using an 18-bit operation (36-bit products). This allows a total of 256 multiplier products.

Figure 3-24: 18-Bit Systolic FIR Mode with Two Dynamic Inputs for Stratix V Devices

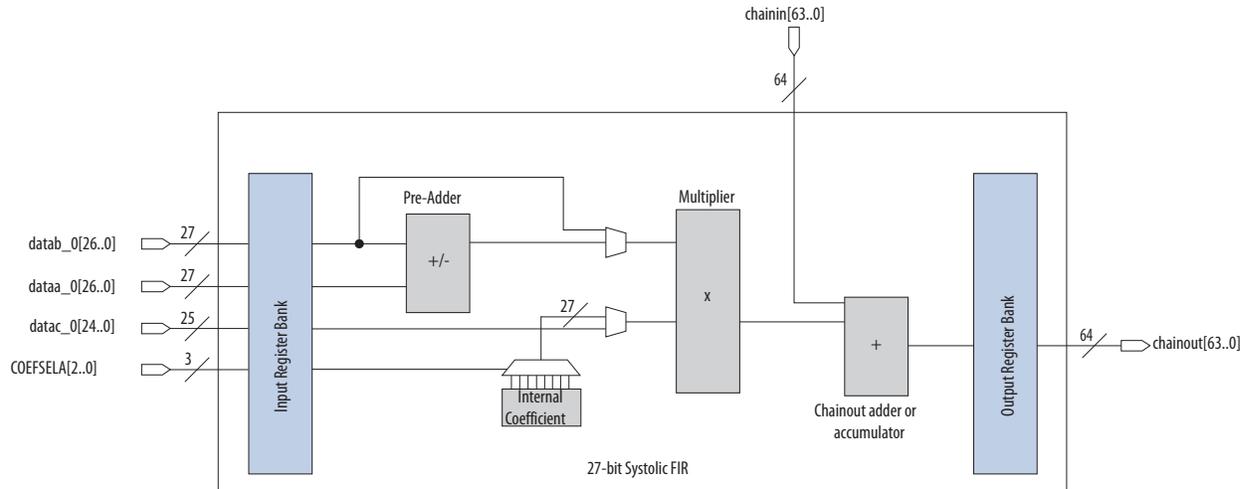


27-Bit Systolic FIR Mode

In 27-bit systolic FIR mode, the chainout adder or accumulator is configured for a 64-bit operation, providing 10 bits of overhead when using a 27-bit data (54-bit products). This allows a total of 1,024 multiplier products.

The 27-bit systolic FIR mode allows the implementation of one stage systolic filter per DSP block.

Figure 3-25: 27-Bit Systolic FIR Mode 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.

The Stratix V variable precision DSP block supports 18-bit and 27-bit input cascading.

Table 3-6: 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	This signal has two functions: <ul style="list-style-type: none"> 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

Signal Name	Function	Count
ENA0	Variable precision DSP-block-wide clock enable signals	3
ENA1		
ENA2		
ACLR0	Variable precision DSP-block-wide asynchronous clear signals	2
ACLR1		
Total Count per DSP Block		14

Variable Precision DSP Blocks in Stratix V Devices Revision History

Date	Version	Changes
December 2015	2015.12.21	Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i> .
July 2014	2014.07.22	Reinstated input register bank and systolic registers to the block architecture.
June 2014	2014.06.30	<ul style="list-style-type: none"> Updated the supported megafunctions from ALTMULT_ADD and ALTMULT_ACCUM to ALTERA_MULT_ADD. Updated modes applicable to the accumulator
May 2013	2013.05.06	<ul style="list-style-type: none"> Added link to the known document issues in the Knowledge Base. Moved all links to the Related Information section of respective topics for easy reference.

Date	Version	Changes
December 2012	2012.12.28	<ul style="list-style-type: none"> Added "Design Considerations" Updated Figure 3-1 changed Mult_L and Mult_H to Multiplier Updated Figure 3-6 changed Mult_L to Multiplier Updated Figure 3-7 changed Mult_L and Mult_H to Multiplier Updated Figure 3-8 changed Mult_L and Mult_H to Multiplier Updated Figure 3-10 changed Mult_L and Mult_H to Multiplier Updated Figure 3-11 changed Mult_L and Mult_H to Multiplier Updated Figure 3-13 changed Mult_L and Mult_H to Multiplier Updated Figure 3-17 changed Mult_L and Mult_H to Multiplier Updated Figure 3-19 changed 54 to 55 Updated Figure 3-20 changed 19 to 18 and deleted Chainout [38..0] Updated Figure 3-23 changed Mult_L to Multiplier Updated Figure 3-24 changed Mult_L and Mult_H to Multiplier and added 44 Updated Figure 3-25 added 64 Reorganized content and updated template.
June 2012	1.4	<ul style="list-style-type: none"> Added Figure 3-2. Updated Figure 3-7, Figure 3-16, and Figure 3-18. Updated Table 3-1. Updated "Chainout Adder and Accumulator" and "18 x 25 Complex Multiplier" sections.
November 2011	1.3	<ul style="list-style-type: none"> Added Figure 3-21. Updated Figure 3-1, Figure 3-2, Figure 3-11, Figure 3-12, Figure 3-14, Figure 3-16, Figure 3-17, Figure 3-18, Figure 3-19, Figure 3-20, and Figure 3-21. Updated Table 3-1 and Table 3-5. Updated "Pre-Adder and Coefficient Select", "Systolic Register", "Systolic FIR Mode", and "Software Support" sections.
May 2011	1.2	<ul style="list-style-type: none"> 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.

2020.09.10

SV51005



Subscribe



Send Feedback

This chapter describes the advanced features of hierarchical clock networks and phase-locked loops (PLLs) in Stratix V devices. The Intel Quartus Prime software enables the PLLs and their features without external devices.

Related Information

[Stratix V Device Handbook: Known Issues](#)

Lists the planned updates to the Stratix V Device Handbook chapters.

Clock Networks

The Stratix V devices contain the following clock networks that are organized into a hierarchical structure:

- Global clock (GCLK) networks
- Regional clock (RCLK) networks
- Periphery clock (PCLK) networks

Clock Resources in Stratix V Devices

Table 4-1: Clock Resources in Stratix V Devices

Clock Resource	Device	Number of Resources Available	Source of Clock Resource
Clock input pins	All	48 single-ended or 24 differential	CLK[0..23][p,n] pins
GCLK networks	All	16	CLK[0..23][p,n] pins, PLL clock outputs, and logic array
RCLK networks	All	92	CLK[0..23][p,n] pins, PLL clock outputs, and logic array

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

Clock Resource	Device	Number of Resources Available	Source of Clock Resource
PCLK networks	<ul style="list-style-type: none"> Stratix V GS D3 and D4 Stratix V GX A3 (with 24 transceivers) 	210	DPA clock outputs, PLD-transceiver interface clocks, I/O pins, and logic array
	<ul style="list-style-type: none"> Stratix V GS D5 Stratix V GX A3 (with 36 transceivers), A4, B5, and A6 	282	
	<ul style="list-style-type: none"> Stratix V GS D6 and D8 Stratix V GT C5 and C7 Stratix V GX A5 and A7 	306	
	<ul style="list-style-type: none"> Stratix V E E9 and EB Stratix V GX A9, AB, B9, and BB 	342	

For more information about the clock input pins connections, refer to the pin connection guidelines.

Related Information

- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
- [Stratix V GT Device Family Pin Connection Guidelines](#)

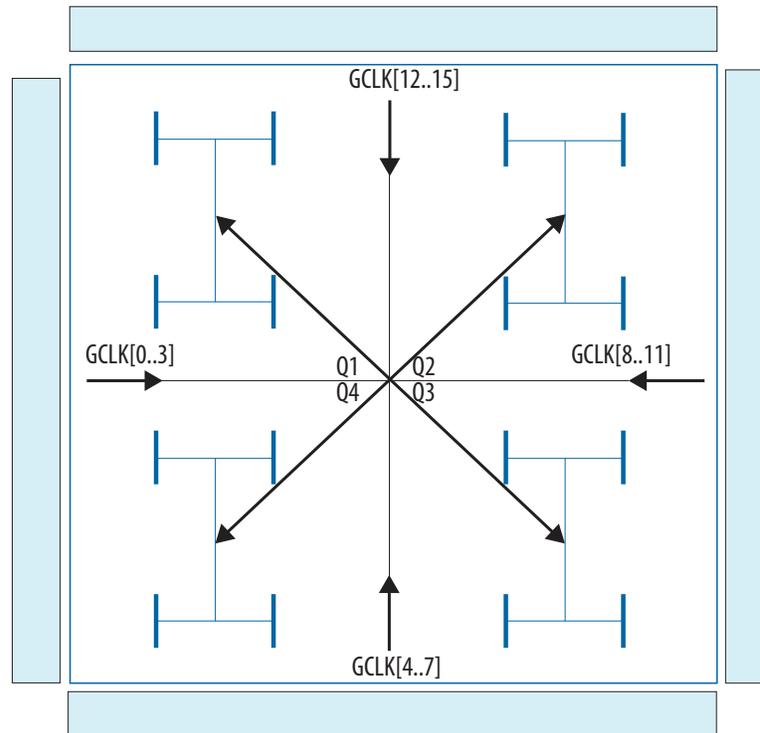
Types of Clock Networks

Global Clock Networks

Stratix V devices provide GCLKs that can drive throughout the device. The GCLKs serve as low-skew clock sources for functional blocks, such as adaptive logic modules (ALMs), digital signal processing (DSP), embedded memory, and PLLs. Stratix V I/O elements (IOEs) and internal logic can also drive GCLKs to create internally-generated global clocks and other high fan-out control signals, such as synchronous or asynchronous clear and clock enable signals.

Figure 4-1: GCLK Networks in Stratix V Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

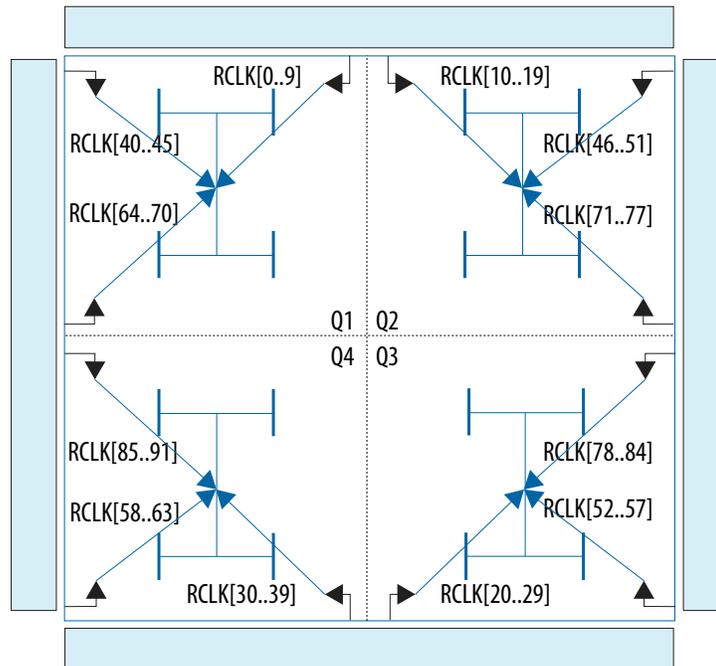


Regional Clock Networks

RCLK networks are only applicable to the quadrant they drive into. RCLK networks provide the lowest clock insertion delay and skew for logic contained within a single device quadrant. The Stratix V 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.

Figure 4-2: RCLK Networks in Stratix V Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.



Periphery Clock Networks

Depending on the routing direction, Stratix V devices provide 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.

Figure 4-3: PCLK Networks for Stratix V GS D5 Device, and Stratix V GX A3 (with 36 transceivers) and A4 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

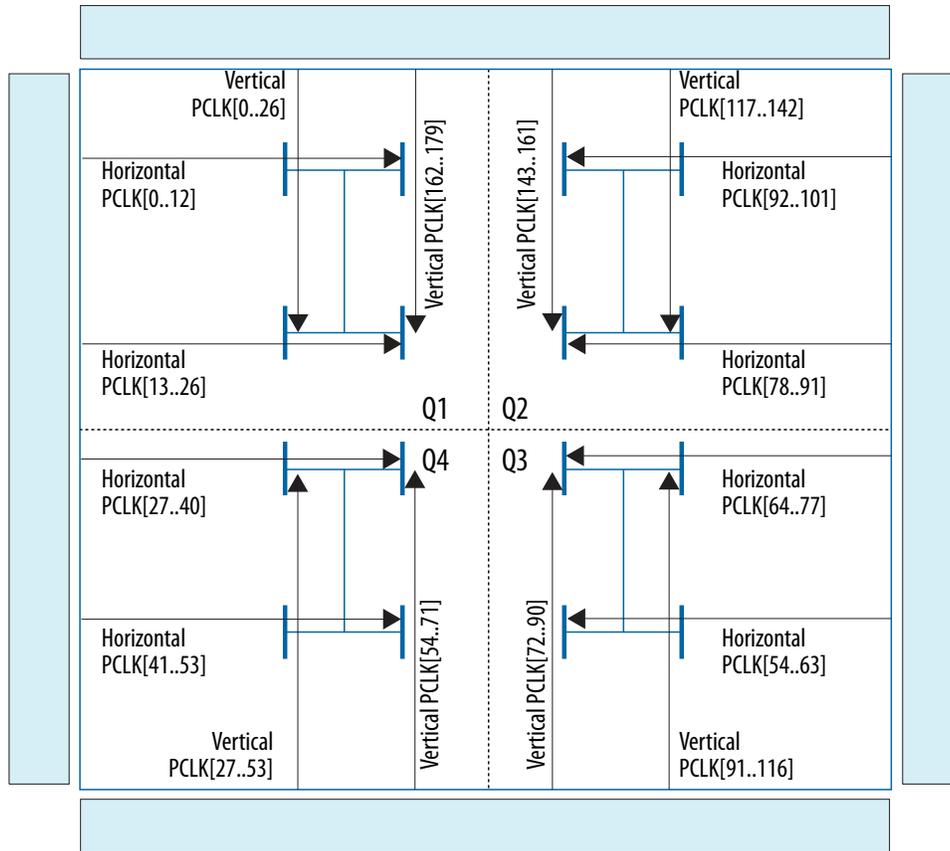


Figure 4-4: PCLK Networks for Stratix V GX B5 and B6 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

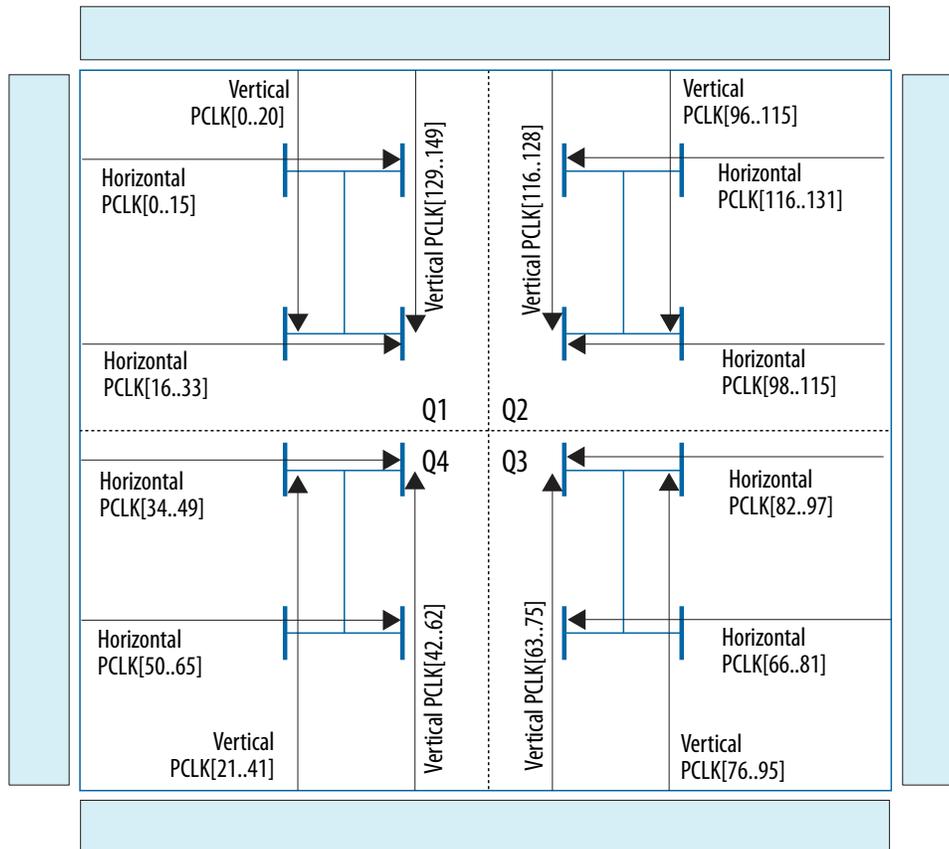


Figure 4-5: PCLK Networks for Stratix V GT C5 and C7 Devices, and Stratix V GX A5 and A7 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

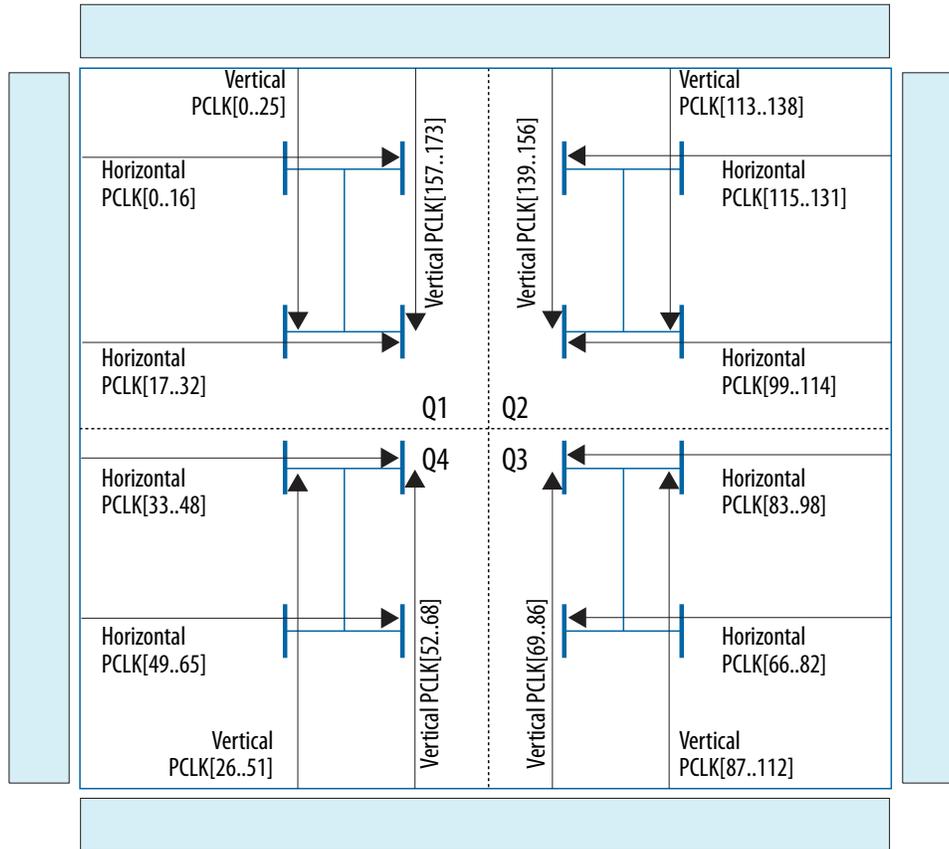


Figure 4-6: PCLK Networks for Stratix V GS D3 and D4 Devices, and Stratix V GX A3 (with 24 transceivers) Device

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

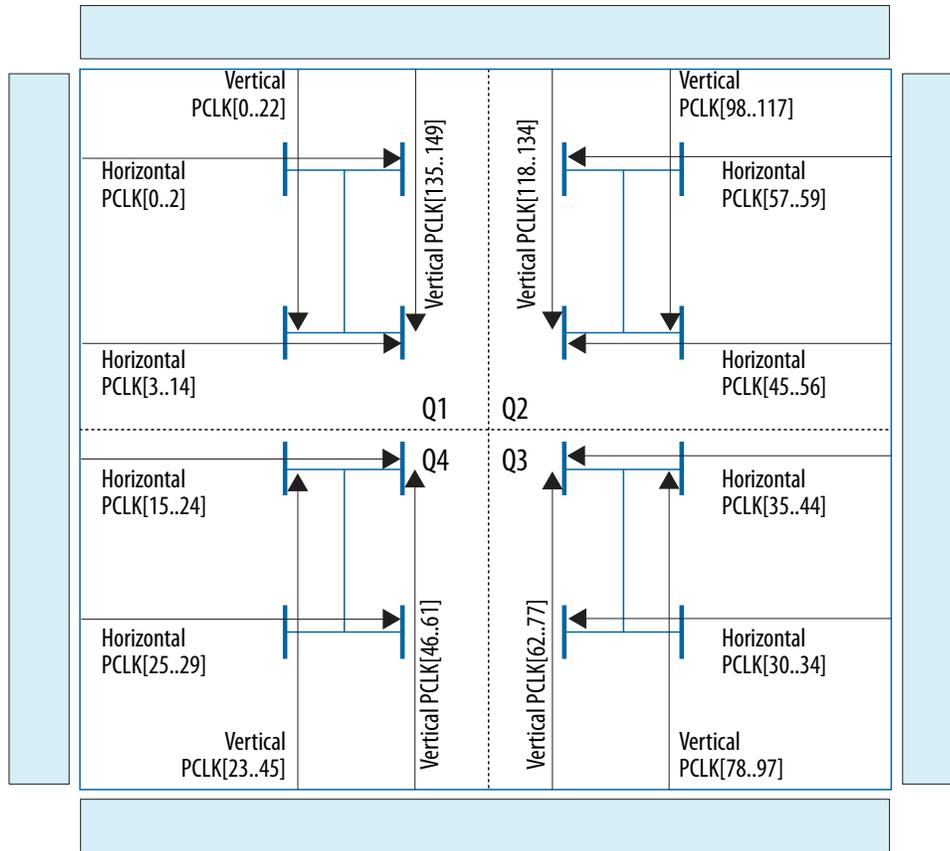


Figure 4-7: PCLK Networks for Stratix V GS D6 and D8 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

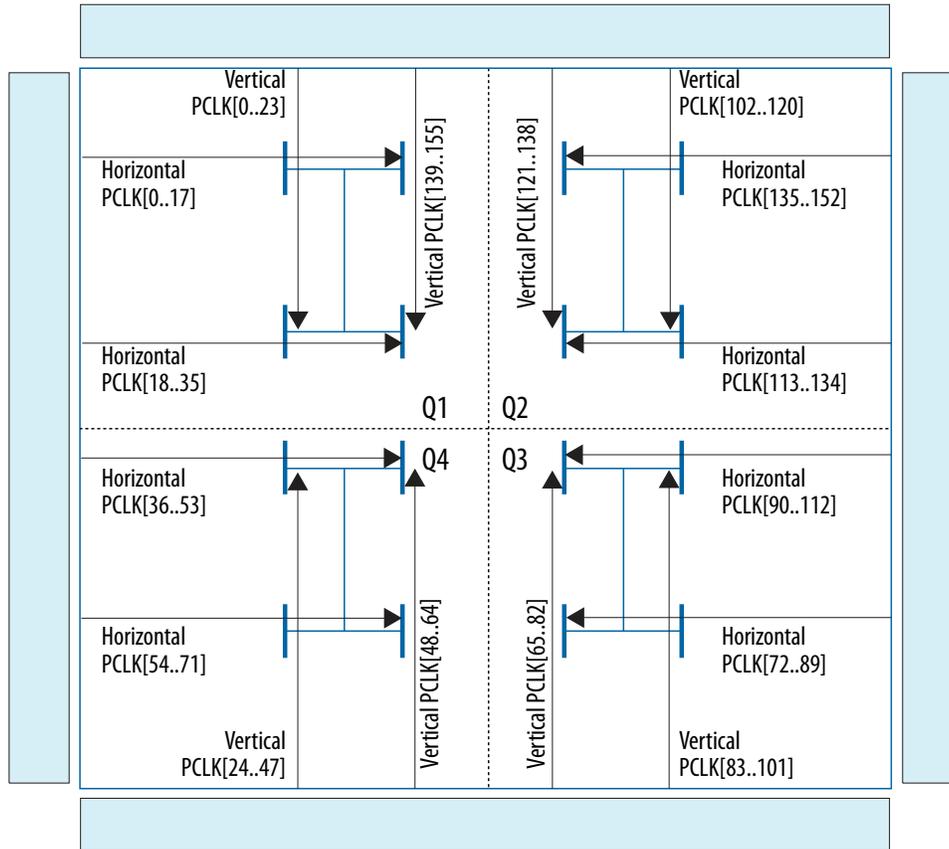
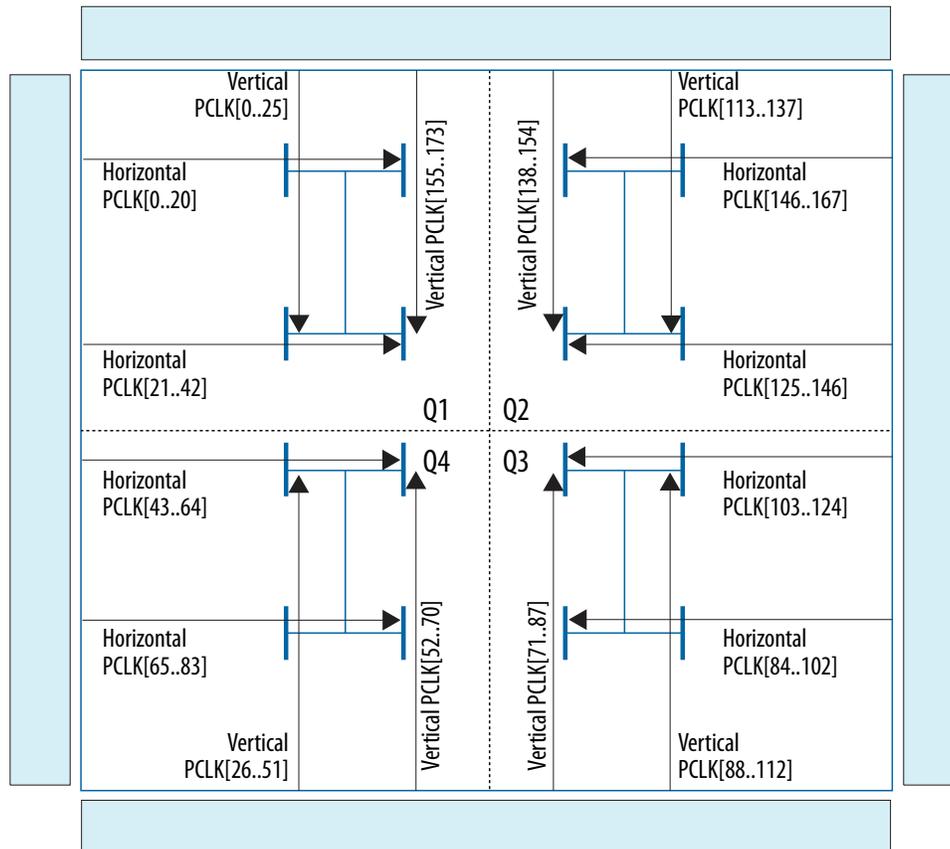


Figure 4-8: PCLK Networks for Stratix V E E9 and EB Devices, and Stratix V GX A9, AB, BB, and B9 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

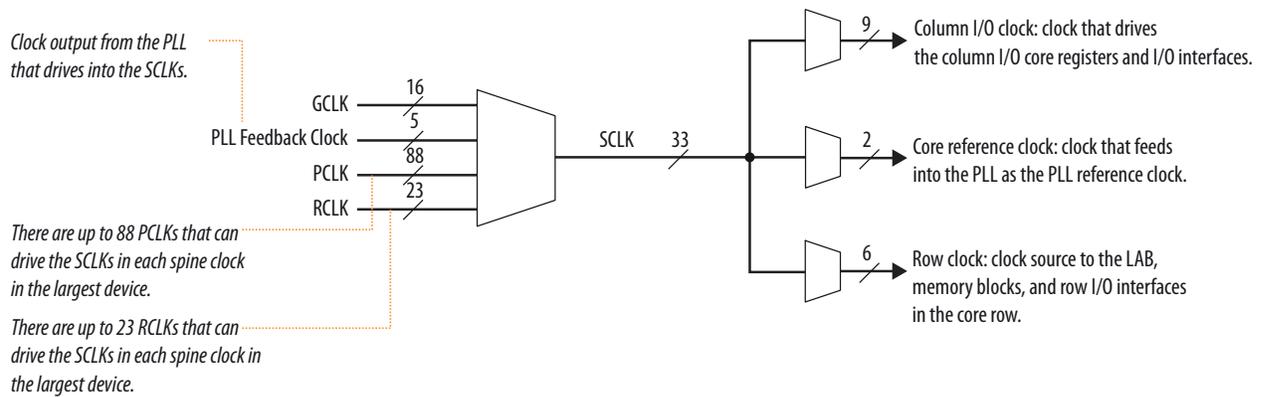


Clock Sources Per Quadrant

The Stratix V devices provide 33 section clock (SCLK) networks in each spine clock per quadrant. The SCLK networks 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.

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

The following figure shows SCLKs driven by the GCLK, RCLK, PCLK, or the PLL feedback clock networks in each spine clock per quadrant. The GCLK, RCLK, PCLK, and PLL feedback clocks share the same routing to the SCLKs. To ensure successful design fitting in the Intel Quartus Prime software, the total number of clock resources must not exceed the SCLK limits in each region.

Figure 4-9: Hierarchical Clock Networks in Each Spine Clock Per Quadrant

Types of Clock Regions

This section describes the types of clock regions in Stratix V devices.

Entire Device Clock Region

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

Regional Clock Region

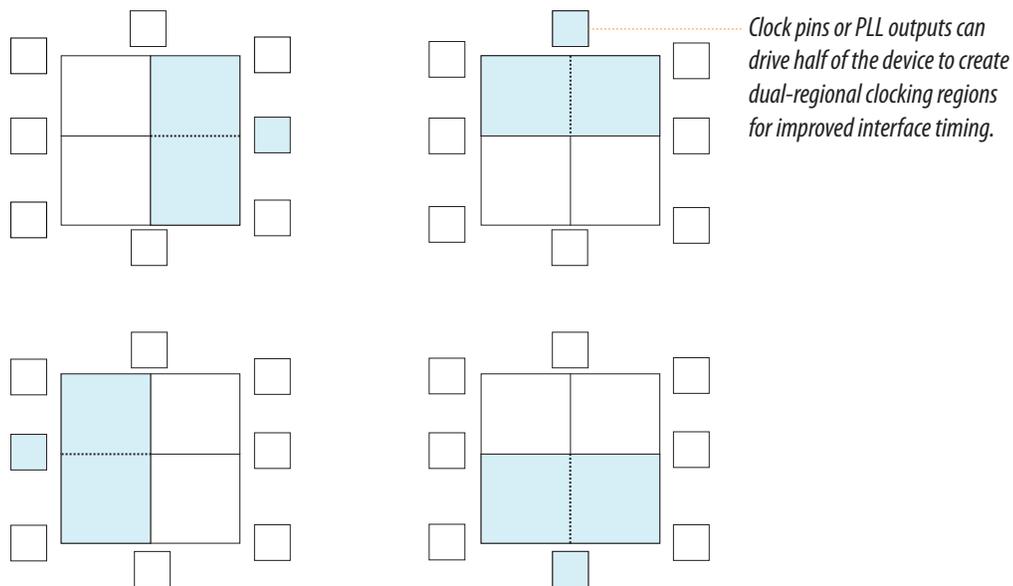
To form a regional clock region, a source drives a signal in a RCLK network that you can route throughout one quadrant of the device. This clock region provides the lowest skew in a quadrant. It is a good option if all the destinations are in a single quadrant.

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 adjacent 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.

Figure 4-10: Dual-Regional Clock Region for Stratix V Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.



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, RCLK, and PCLK networks.

Dedicated Clock Input Pins

You can use the dedicated clock input pins ($CLK[0..23]_{[p,n]}$) for high fan-out control signals, such as asynchronous clears, presets, and clock enables, for protocol signals through the GCLK or RCLK networks.

CLK pins can be either differential clocks or single-ended clocks. When you use the CLK pins as single-ended clock inputs, only the $CLK_{<\#>p}$ pins have dedicated connections to the PLL. The $CLK_{<\#>n}$ pins drive the PLLs over global or regional clock networks and do not have dedicated routing paths to the PLLs.

Driving a PLL over a global or regional clock can lead to higher jitter at the PLL input, and the PLL will not be able to fully compensate for the global or regional clock. Altera recommends using the $CLK_{<\#>p}$ pins for optimal performance when you use single-ended clock inputs to drive the PLLs.

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.

Note: Internally-generated GCLKs, RCLKs, or PCLKs cannot drive the Stratix V PLLs. The input clock to the PLL has to come from dedicated clock input pins, PLL-fed GCLKs, or PLL-fed RCLKs.

DPA Outputs

Every DPA generates one PCLK to the core.

Related Information

[High-Speed I/O Design Guidelines for Stratix V Devices](#) on page 6-8
Provides more information about DPA and HSSI outputs.

HSSI Outputs

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

Related Information

[High-Speed I/O Design Guidelines for Stratix V Devices](#) on page 6-8
Provides more information about DPA and HSSI outputs.

PLL Clock Outputs

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

Clock Input Pin Connections to GCLK and RCLK Networks

Table 4-2: Dedicated Clock Input Pin Connectivity to the GCLK Networks for Stratix V Devices

Clock Resources	CLK (p/n Pins)
GCLK[0,1,2,3]	CLK[0,1,2,3,20,21,22,23]
GCLK[4,5,6,7]	CLK[4,5,6,7]
GCLK[8,9,10,11]	CLK[8,9,10,11,12,13,14,15]
GCLK[12,13,14,15]	CLK[16,17,18,19]

Table 4-3: Dedicated Clock Input Pin Connectivity to the RCLK Networks for Stratix V Devices

A given clock input pin can drive two adjacent RCLK networks to create a dual-regional clock network.

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

Clock Resources	CLK (p/n Pins)
RCLK[52,53,54,55,56,57,73,77,80,84]	CLK[10]
RCLK[52,53,54,55,56,57,74,81]	CLK[11]
RCLK[46,47,48,49,50,51,71,75,78,82]	CLK[12]
RCLK[46,47,48,49,50,51,72,76,79,83]	CLK[13]
RCLK[46,47,48,49,50,51,73,77,80,84]	CLK[14]
RCLK[46,47,48,49,50,51,74,81]	CLK[15]
RCLK[0,4,8,10,14,18]	CLK[16]
RCLK[1,5,9,11,15,19]	CLK[17]
RCLK[2,6,12,16]	CLK[18]
RCLK[3,7,13,17]	CLK[19]
RCLK[40,41,42,43,44,45,64,68,85,89]	CLK[20]
RCLK[40,41,42,43,44,45,65,69,86,90]	CLK[21]
RCLK[40,41,42,43,44,45,66,70,87,91]	CLK[22]
RCLK[40,41,42,43,44,45,67,88]	CLK[23]

Clock Output Connections

For Stratix V PLL connectivity to GCLK and RCLK networks, refer to the PLL connectivity to GCLK and RCLK networks spreadsheet.

Related Information

[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)

Pin Mapping in Stratix V Devices

Table 4-4: Mapping Between the Input Clock Pins, PLL Counter Outputs, and Clock Control Block Inputs

Clock	Fed by
inclk[0] and inclk[1]	Any of the four dedicated clock pins on the same side of the Stratix V device.
inclk[2]	PLL counters c0 and c2 from the two center PLLs on the same side of the Stratix V devices.

Clock	Fed by
inclck[3]	PLL counters c1 and c3 from the two center PLLs on the same side of the Stratix V devices.

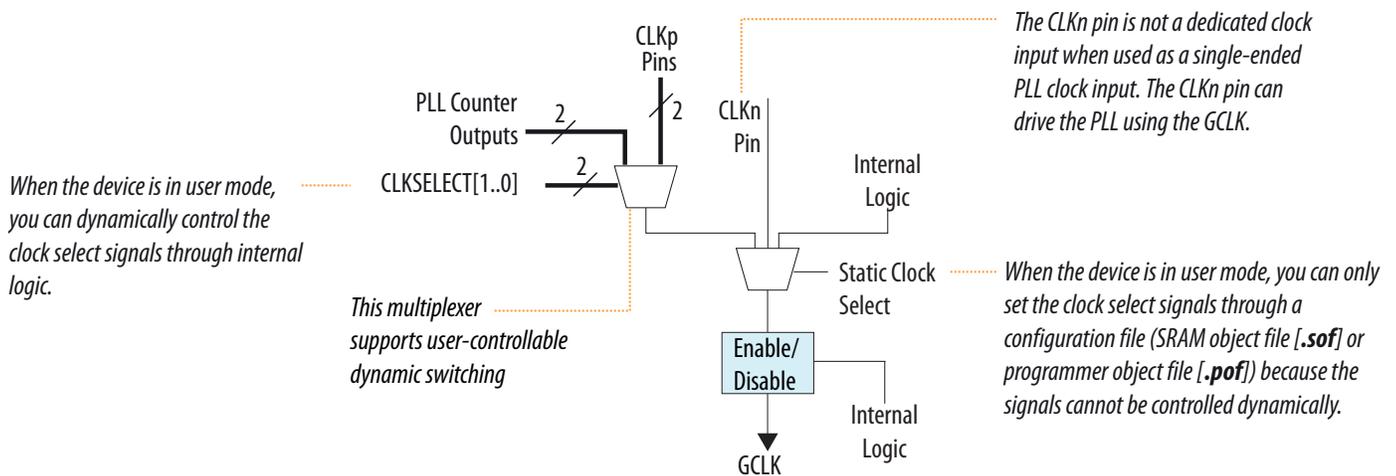
Note: You cannot use corner PLLs for dynamic clock control selection.

GCLK Control Block

You can select the clock source for the GCLK select block either statically or dynamically 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.

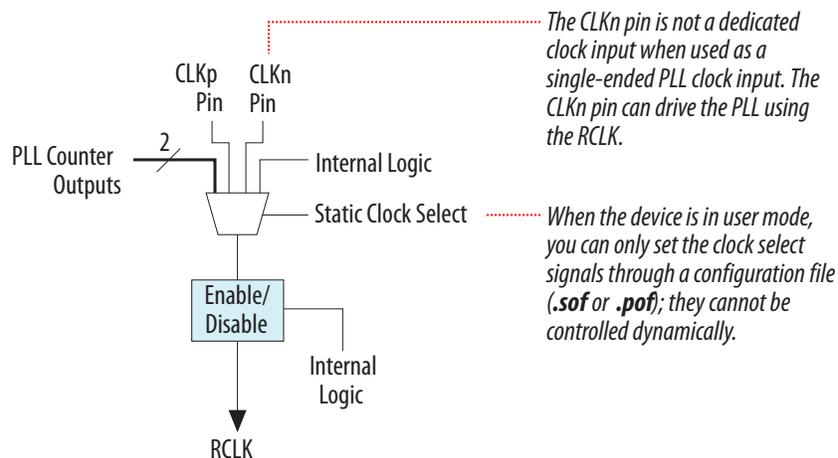
Figure 4-11: GCLK Control Block for Stratix V Devices



RCLK Control Block

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 Intel Quartus Prime software.

Figure 4-12: RCLK Control Block for Stratix V Devices



You can set the input clock sources and the `clkena` signals for the GCLK and RCLK network multiplexers through the Intel Quartus Prime software using the ALTCLKCTRL IP core.

Note: When selecting the clock source dynamically using the ALTCLKCTRL IP core, choose the inputs using the `CLKSELECT[0..1]` signal. The inputs from the clock pins feed the `inclk[0..1]` ports of the multiplexer, and the PLL outputs feed the `inclk[2..3]` ports.

Related Information

[Clock Control Block \(ALTCLKCTRL\) IP Core User Guide](#)

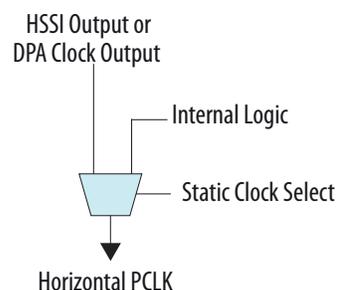
Provides more information about ALTCLKCTRL IP core.

PCLK Control Block

To drive the HSSI horizontal PCLK control block, select the HSSI output or internal logic .

To drive the DPA horizontal PCLK, select the DPA clock output or internal logic. You can only use the DPA clock output to generate the vertical PCLK to the core.

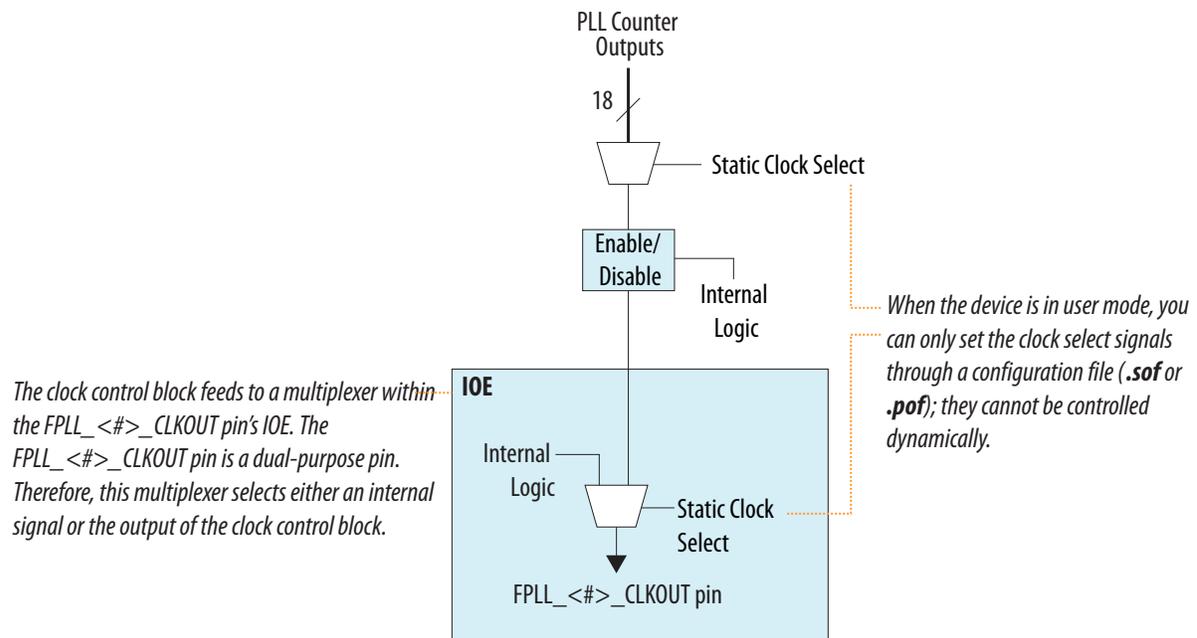
Figure 4-13: Horizontal PCLK Control Block for Stratix V Devices



External PLL Clock Output Control Block

You can enable or disable the dedicated external clock output pins using the ALTCLKCTRL IP core.

Figure 4-14: External PLL Output Clock Control Block for Stratix V Devices

**Related Information****Clock Control Block (ALTCLKCTRL) IP Core User Guide**

Provides more information about ALTCLKCTRL IP core.

Clock Power Down

You can power down the 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, 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 Intel Quartus Prime software.

The dynamic clock enable or disable feature allows the internal logic to control power-up or power-down synchronously on the GCLK and RCLK networks, including dual-regional clock regions. This feature is independent of the PLL and is applied directly on the clock network.

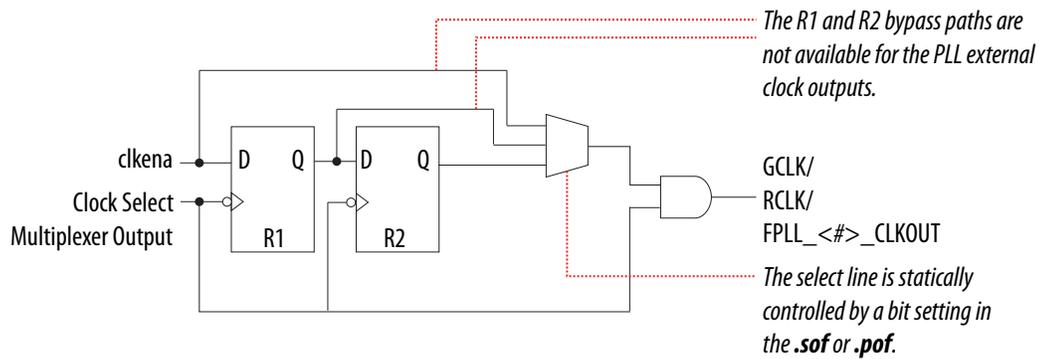
Note: You cannot dynamically enable or disable GCLK or RCLK networks that drive PLLs.

Clock Enable Signals

You cannot use the clock enable and disable circuit of the clock control block if the GCLK or RCLK output drives the input of a PLL.

Figure 4-15: `clkena` Implementation with Clock Enable and Disable Circuit

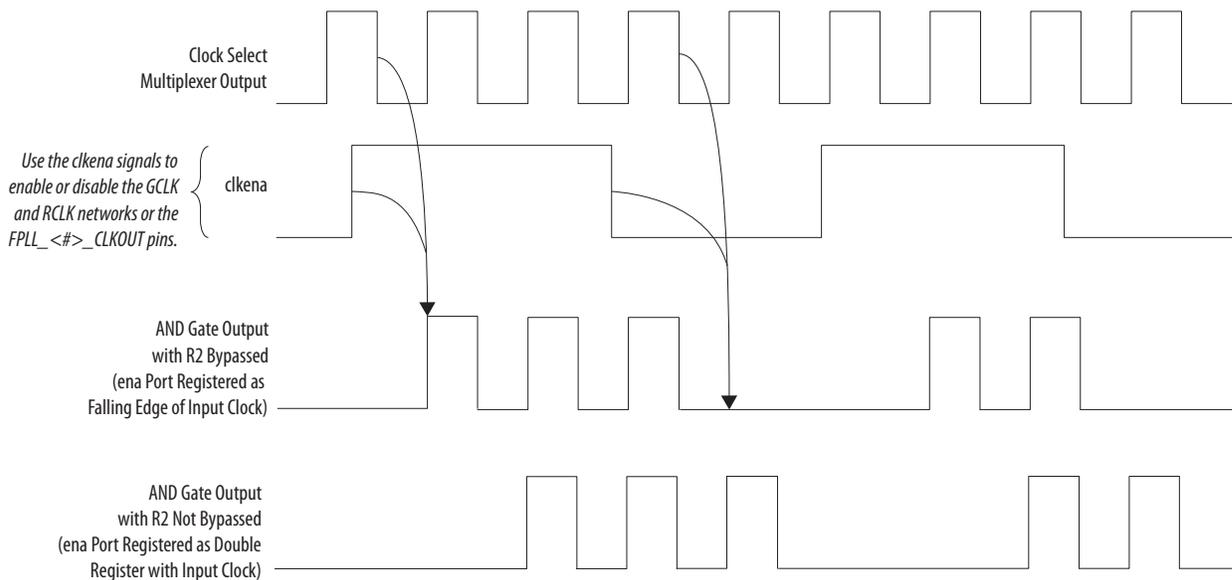
This figure shows the implementation of the clock enable and disable circuit of the clock control block.



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 4-16: Example of `clkena` Signals

This figure shows a waveform example for a clock output enable. The `clkena` signal is synchronous to the falling edge of the clock output.



Stratix V devices 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 Intel Quartus Prime software.

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

PLLs provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces.

The Stratix V device family contains fractional PLLs that can function as fractional PLLs or integer PLLs. The output counters in Stratix V devices are dedicated to each fractional PLL that support integer or fractional frequency synthesis.

Two adjacent PLLs share 18 c output counters. Any number of c counters can be assigned to each PLL, as long as the total number used by the two PLLs is 18 or less.

The Stratix V devices offer up to 32 fractional PLLs in the larger densities. All Stratix V fractional PLLs have the same core analog structure and features support.

Table 4-5: PLL Features in Stratix V Devices

Feature	Support
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
Dedicated clock input pins	4 single-ended or 4 differential
External feedback input pin	Single-ended or differential
Spread-spectrum input clock tracking	Yes ⁽²⁾
Source synchronous compensation	Yes
Direct compensation	Yes
Normal compensation	Yes
Zero-delay buffer compensation	Yes
External feedback compensation	Yes
LVDS compensation	Yes
Voltage-controlled oscillator (VCO) output drives the DPA clock	Yes
Phase shift resolution	78.125 ps ⁽³⁾

⁽²⁾ Provided input clock jitter is within input jitter tolerance specifications.

⁽³⁾ The smallest phase shift is determined by the 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.

Feature	Support
Programmable duty cycle	Yes
Power down mode	Yes

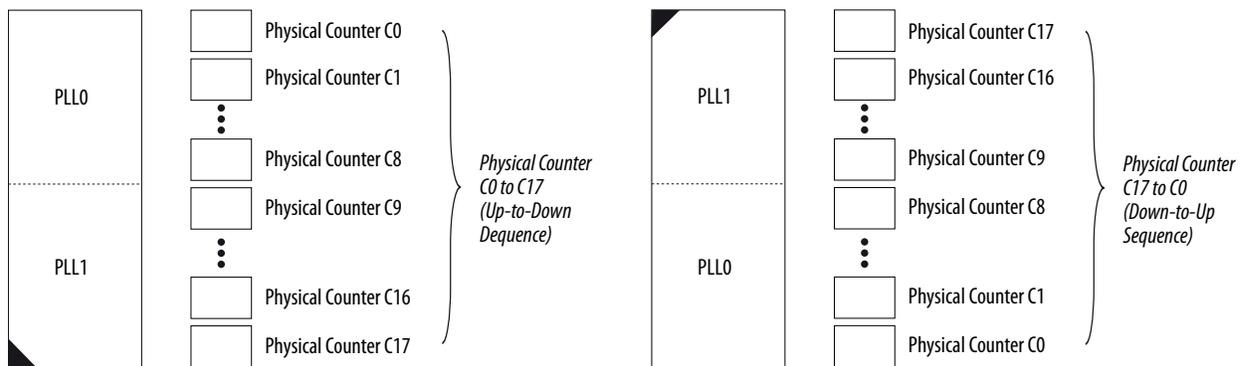
PLL Physical Counters in Stratix V Devices

The physical counters for the fractional PLLs are arranged in the following sequences:

- Up-to-down
- Down-to-up

Figure 4-17: PLL Physical Counters Orientation for Stratix V Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.



PLL Locations in Stratix V Devices

Stratix V devices provide PLLs for the transceiver channels. These PLLs are located in a strip, where the strip refers to an area in the FPGA.

The total number of PLLs in the Stratix V devices includes the PLLs in the PLL strip. However, the transceivers can only use the PLLs located in the strip.

The following figures show the physical locations of the fractional PLLs. Every index represents one fractional PLL in the device. The physical locations of the fractional PLLs correspond to the coordinates in the Quartus II Chip Planner.

Figure 4-18: PLL Locations for Stratix V GS D5 Device, and Stratix V GX A3 (with 36 transceivers) and A4 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

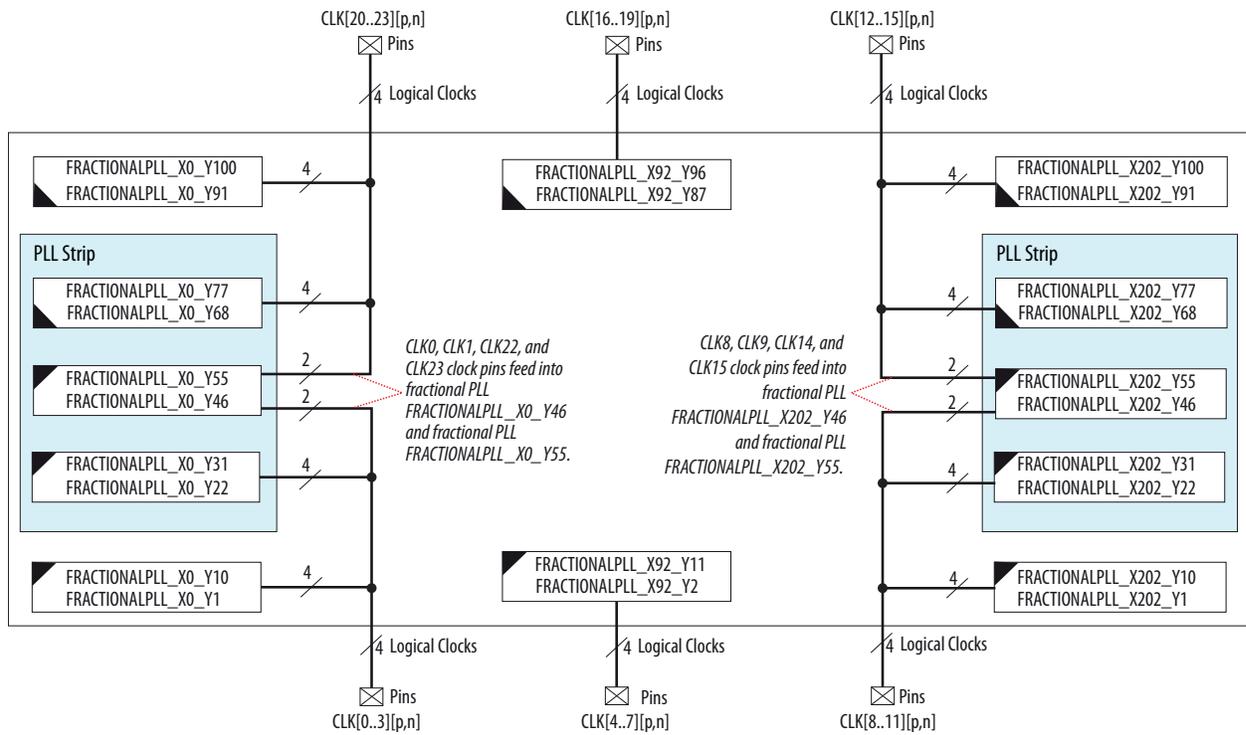


Figure 4-19: PLL Locations for Stratix V GX B5 and B6 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

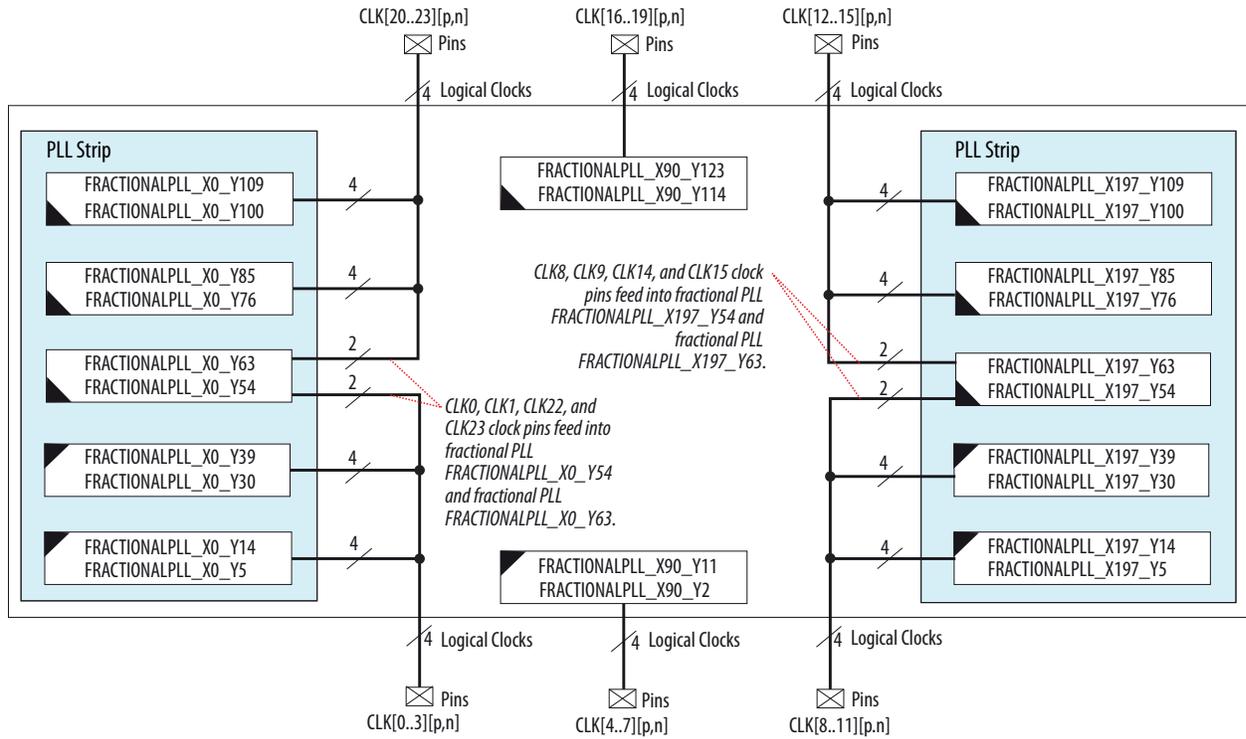


Figure 4-20: PLL Locations for Stratix V GT C5 and C7 Devices, and Stratix V GX A5 and A7 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

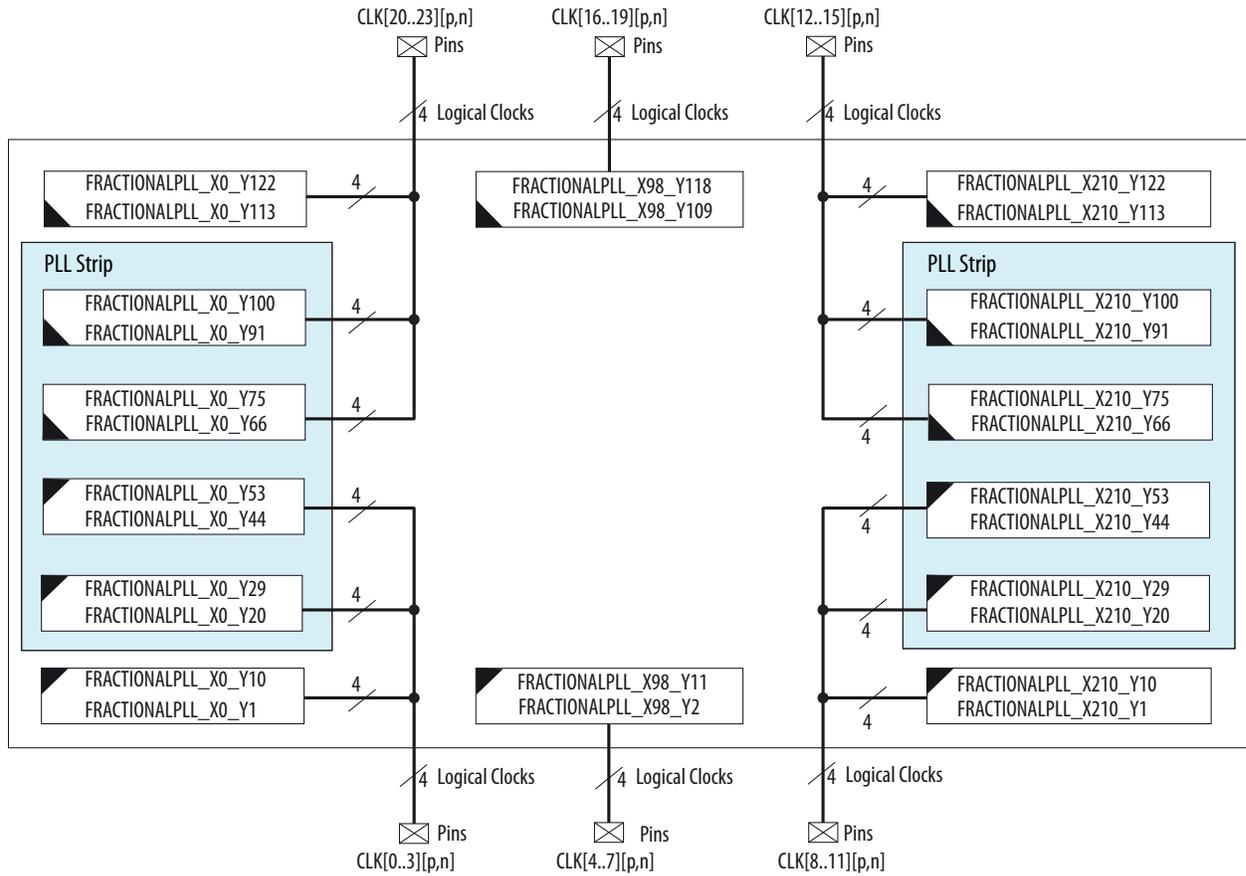


Figure 4-21: PLL Locations for Stratix V GS D3 and D4 Devices, and Stratix V GX A3 (with 24 transceivers) Device

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

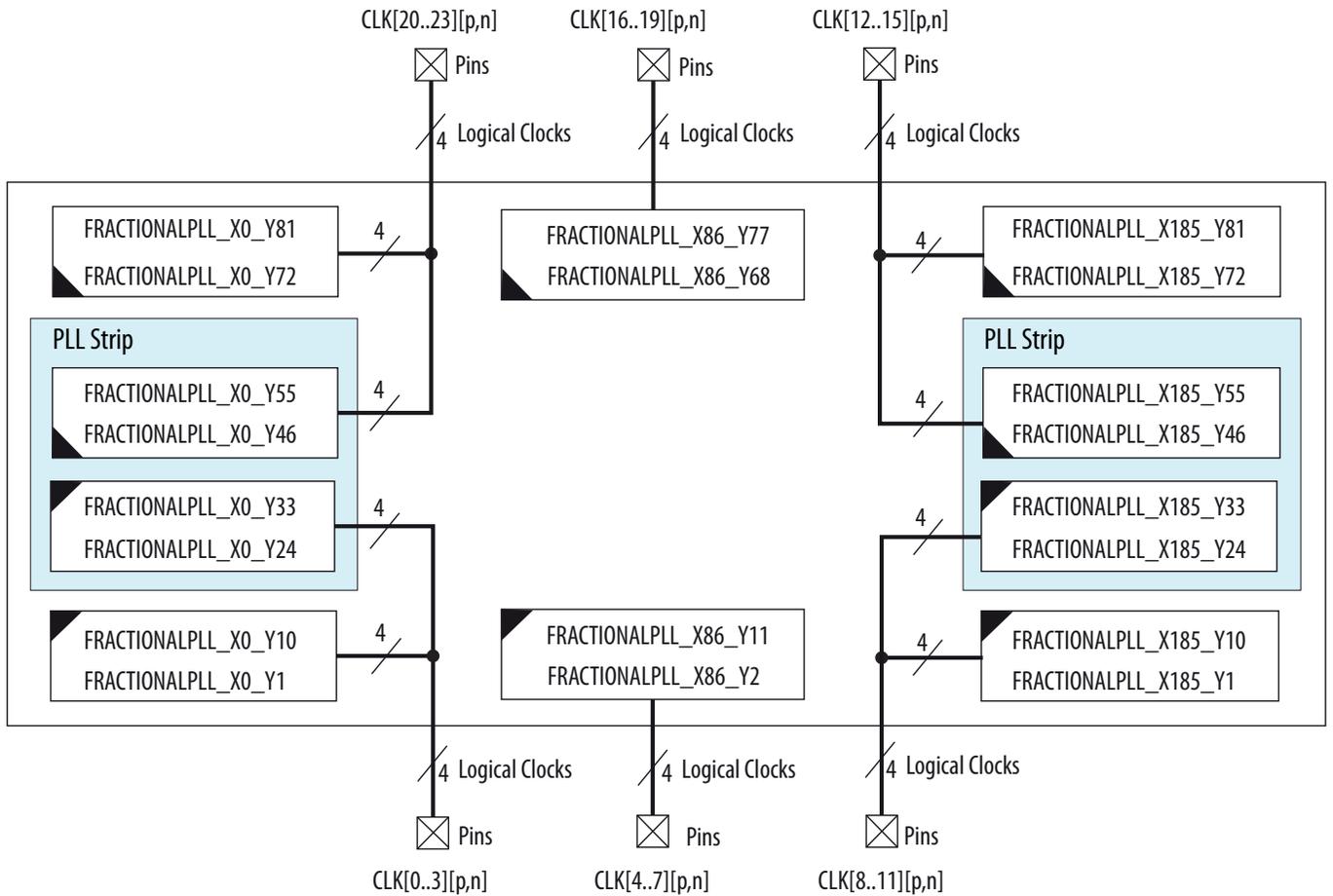


Figure 4-22: PLL Locations for Stratix V GS D6 and D8 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.

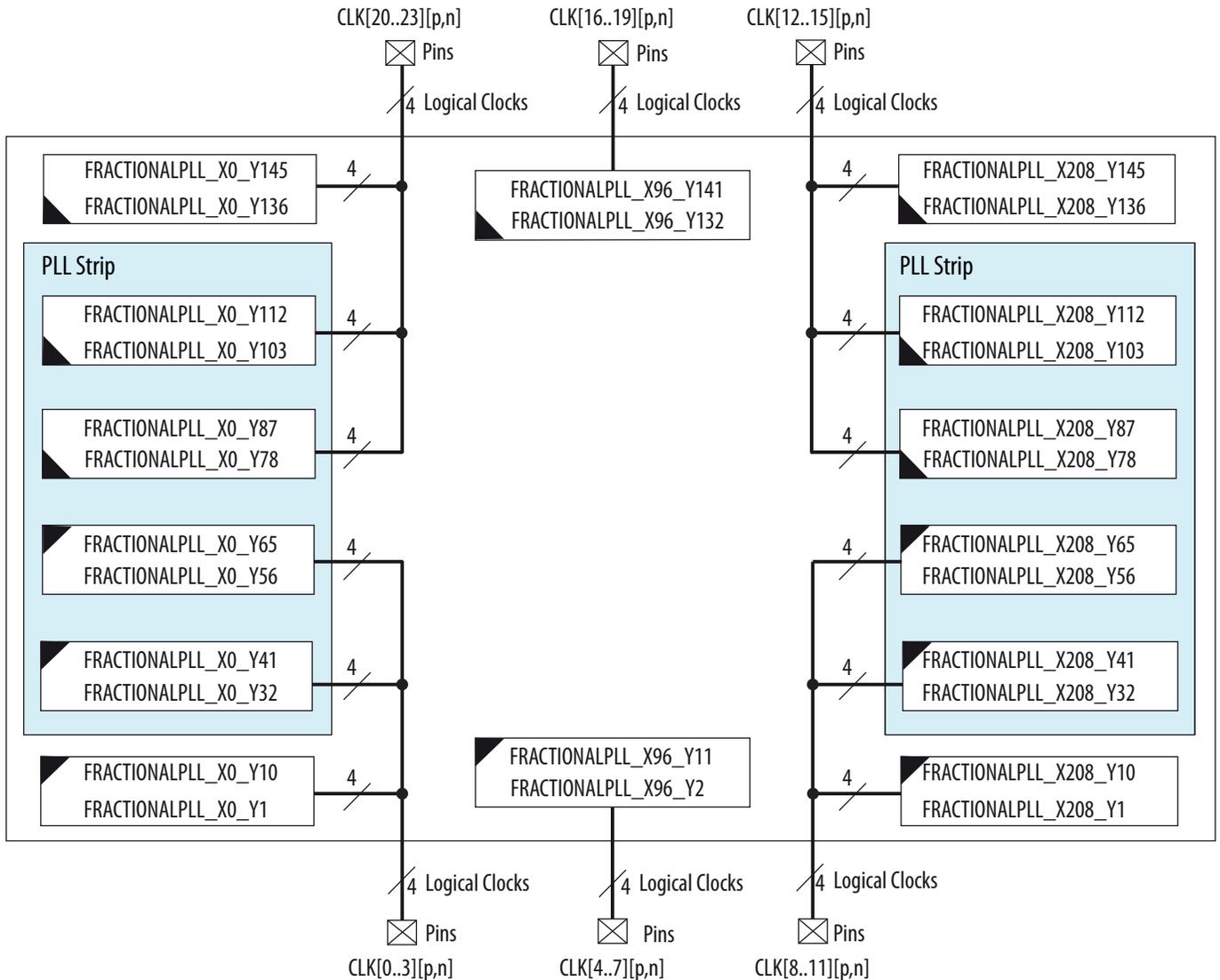
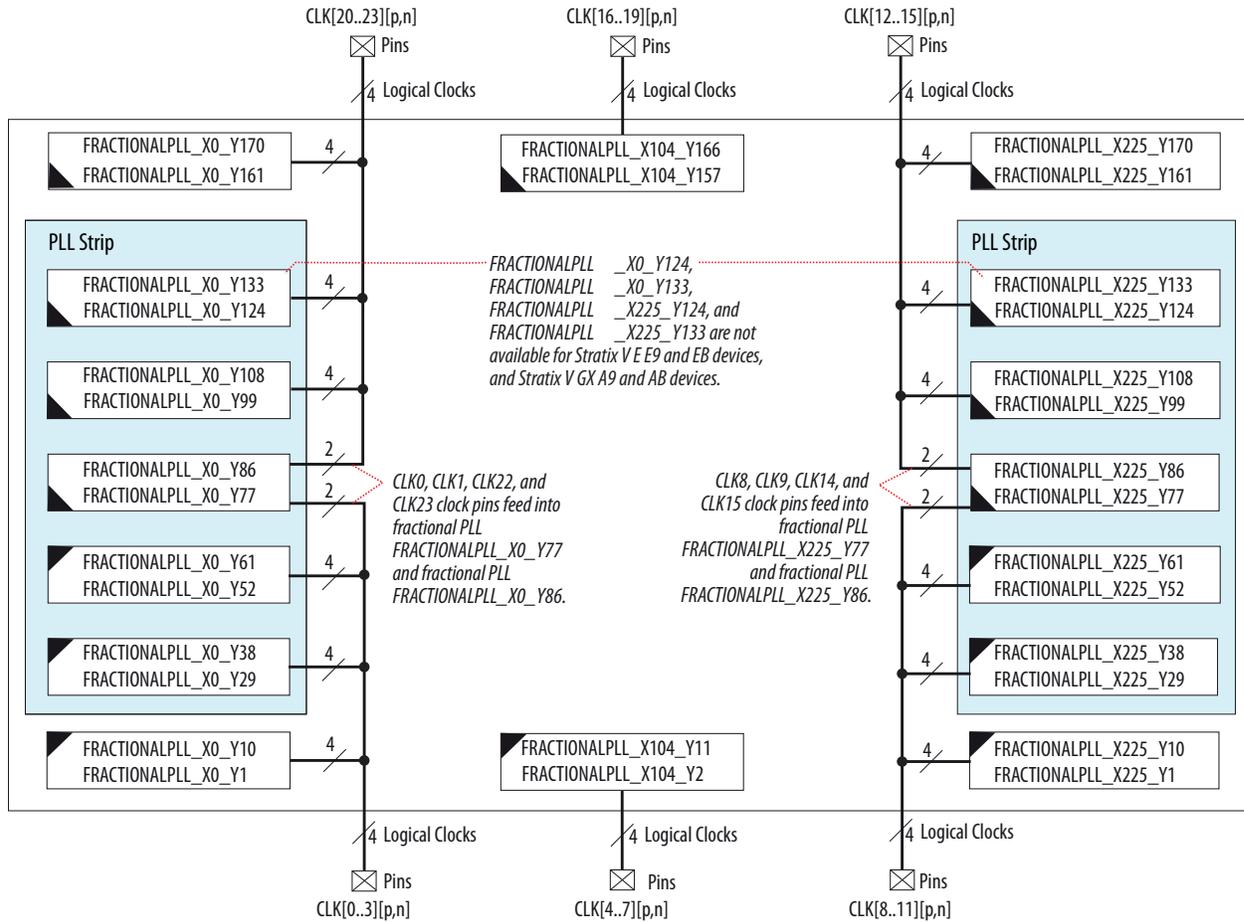


Figure 4-23: PLL Locations for Stratix V E E9 and EB Devices, and Stratix V GX A9, AB, B9, and BB Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.



Related Information

[PLL Migration Guidelines](#) on page 4-26

Provides more information about PLL migration between Stratix V GX A5, A7, A9, AB, B9, BB, D6, and D8 devices.

PLL Migration Guidelines

If you plan to migrate your design between Stratix V GX A5, A7, A9, AB, B9, BB, D6, and D8 devices with 48 transceiver channels, and your design requires a PLL to drive the HSSI and clock network (GCLK or RCLK) simultaneously, use the 2 middle PLLs on the left or right side of the device.

Table 4-6: Location of Middle PLLs for PLL Migration

Variant	Member Code	Middle PLL Location	
		Left Side	Right Side
Stratix V GX	A5	FRACTIONALPLL_X0_Y53,	FRACTIONALPLL_X210_Y53,
	A7	FRACTIONALPLL_X0_Y66	FRACTIONALPLL_X210_Y66
	A9		
	AB	FRACTIONALPLL_X0_Y77,	FRACTIONALPLL_X225_Y77,
	B9	FRACTIONALPLL_X0_Y86	FRACTIONALPLL_X225_Y86
	BB		
	D6	FRACTIONALPLL_X0_Y65,	FRACTIONALPLL_X208_Y65,
	D8	FRACTIONALPLL_X0_Y78	FRACTIONALPLL_X208_Y78

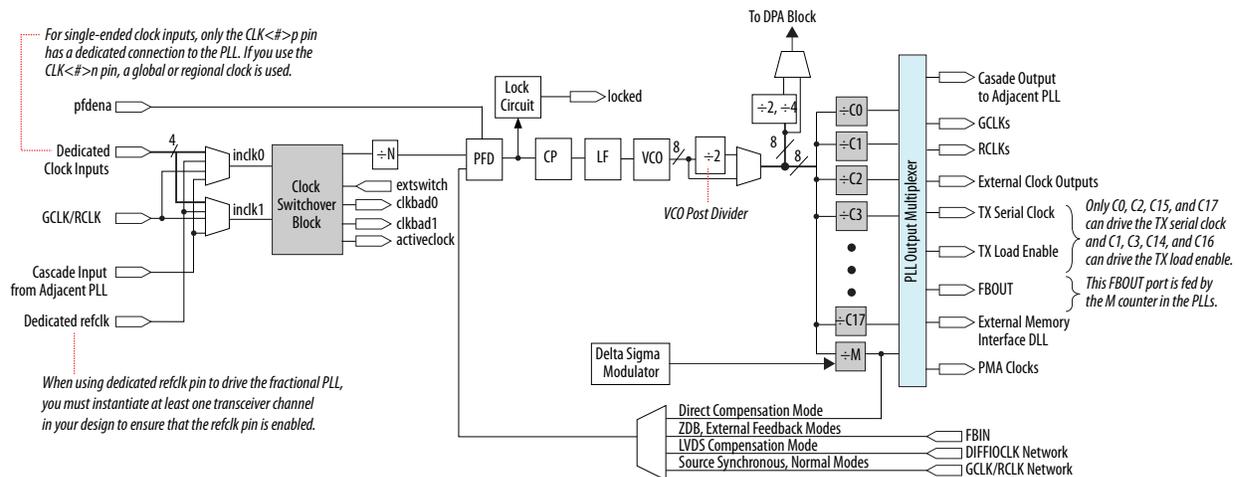
Related Information

PLL Locations in Stratix V Devices on page 4-20

Provides more information about CLKIN pin connectivity to the middle PLLs.

Fractional PLL Architecture

Figure 4-24: Fractional PLL High-Level Block Diagram for Stratix V Devices



Fractional PLL Usage

You can configure the fractional PLL to function either in the integer or in the 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.

Fractional PLLs can be used as follows:

- Reduce the number of required oscillators on the board
- Reduce the clock pins used in the FPGA by synthesizing multiple clock frequencies from a single reference clock source
- Compensate clock network delay
- Zero delay buffering
- Transmit clocking for transceivers

PLL Cascading

Stratix V devices support two types of PLL cascading.

PLL-to-PLL Cascading

This cascading mode synthesizes a more precise output frequency than a single PLL in integer mode. Cascading two PLLs in integer mode expands the effective range of the pre-scale counter, N and the multiply counter, M .

Stratix V devices use two types of input clock sources.

- The `adjpll1in` input clock source is used for inter-cascading between fracturable fractional PLLs.
- The `clock` input clock source is used for intra-cascading within fracturable fractional PLLs.

Altera recommends using a low bandwidth setting for the source (upstream) PLL and a high bandwidth setting for destination (downstream) PLL.

Counter-Output-to-Counter-Output Cascading

This cascading mode synthesizes a lower frequency output than a single post-scale counter, C . Cascading two C counters expands the effective range of C counters.

PLL External Clock I/O Pins

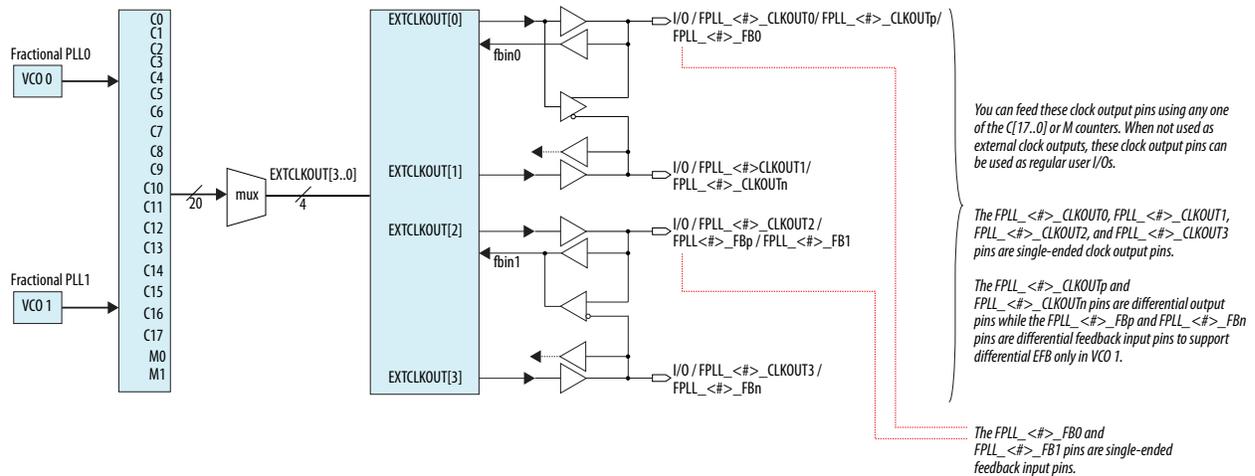
Two adjacent corner and center fractional PLLs share four dual-purpose clock I/O pins, organized as one of the following combinations:

- 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 in the I/O driver feedback for zero delay buffer (ZDB) mode support
- Two single-ended clock outputs and two single-ended feedback inputs for single-ended external feedback (EFB) mode 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)

Note: All left and right fractional PLLs in Stratix V devices do not support external clock outputs.

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

Figure 4-25: Dual-Purpose Clock I/O Pins Associated with PLL for Stratix V Devices



Each pin of a single-ended output pair can be either in-phase or 180° out-of-phase. To implement the 180° out-of-phase pin in a pin pair, the Intel Quartus Prime software places a NOT gate in the design into the IOE.

The clock output pin pairs support the following I/O standards:

- Same I/O standard for the pin pairs
- LVDS
- Differential high-speed transceiver logic (HSTL)
- Differential SSTL

Stratix V PLLs can 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.

Related Information

- [I/O Standards Support in Stratix V Devices](#) on page 5-2
Provides more information about I/O standards supported by the PLL clock input and output pins.
- [Zero-Delay Buffer Mode](#) on page 4-33
- [External Feedback Mode](#) on page 4-35

PLL Control Signals

You can use the `areset` signal to control PLL operation and resynchronization, and use the `locked` signal to observe the status of the PLL.

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 Intel Quartus Prime IP Catalog.

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

Note: 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 the following conditions:

- The PLL has locked onto the reference clock.
- The PLL clock outputs are operating at the desired phase and frequency set in the IP Catalog.

The lock detection circuit provides a signal to the core logic. The signal indicates when the feedback clock has locked onto the reference clock both in phase and frequency.

Clock Feedback Modes

This section describes the following clock feedback modes:

- Source synchronous
- LVDS compensation
- Direct
- Normal compensation
- ZDB
- EFB

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.

The input and output delays may not be fully compensated in the Intel Quartus Prime software for the following conditions:

- When a GCLK or RCLK network drives the PLL
- When the PLL is driven by a dedicated clock pin that is not associated with the PLL

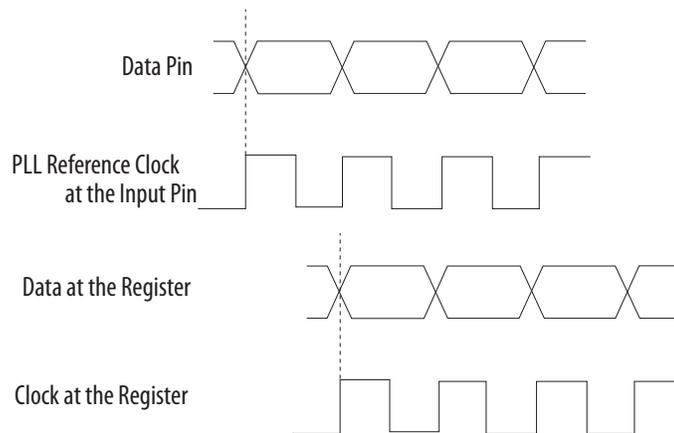
For example, when you configure a PLL in ZDB mode, the PLL input is driven by an associated dedicated clock input pin. In this configuration, a fully compensated clock path results in zero delay between the clock input and one of the clock outputs from the PLL. However, if the PLL input is fed by a non-dedicated input (using the GCLK network), the output clock may not be perfectly aligned with the input clock.

Source Synchronous Mode

If 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. Data and clock signals at the IOE experience similar buffer delays as long as you use the same I/O standard.

Altera recommends source synchronous mode for source synchronous data transfers.

Figure 4-26: Example of Phase Relationship Between Clock and Data in Source Synchronous Mode



The source synchronous mode compensates for the delay of the clock network used and any difference in the delay between the following two paths:

- Data pin to the IOE register input
- Clock input pin to the PLL phase frequency detector (PFD) input

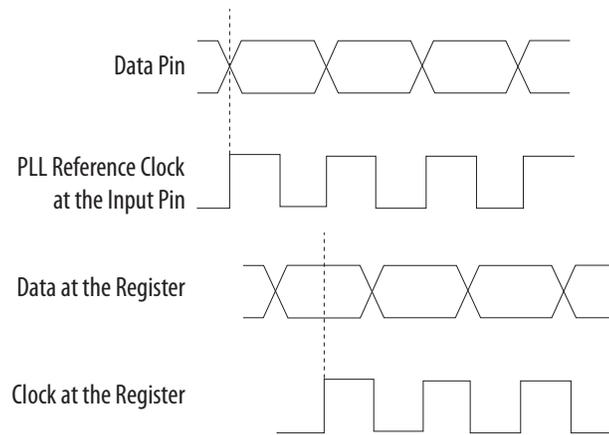
The Stratix V PLL can compensate multiple pad-to-input-register paths, such as a data bus when it is set to use source synchronous compensation mode.

LVDS Compensation Mode

The purpose of LVDS compensation 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, LVDS compensation mode ideally compensates for the delay of the LVDS clock network, including the difference in delay between the following two paths:

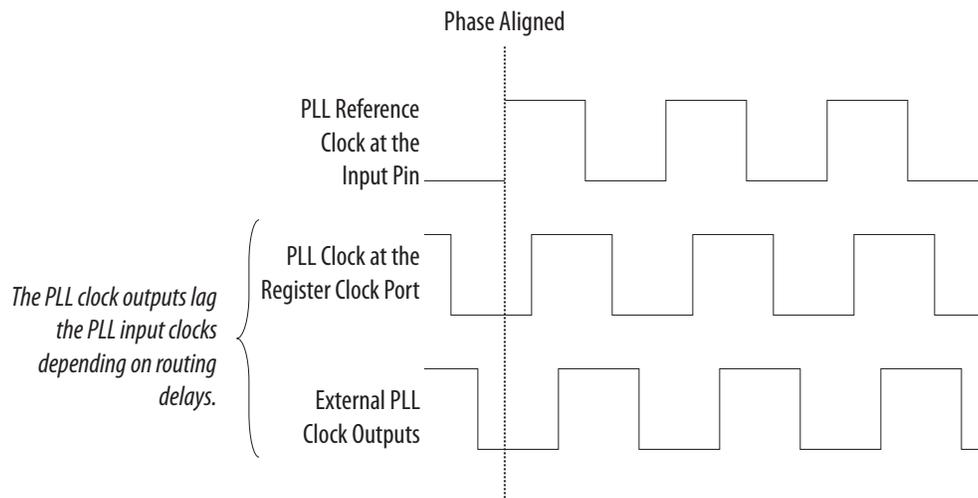
- Data pin-to-SERDES capture register
- Clock input pin-to-SERDES capture register

The output counter must provide the 180° phase shift.

Figure 4-27: Example of Phase Relationship Between the Clock and Data in LVDS Compensation Mode

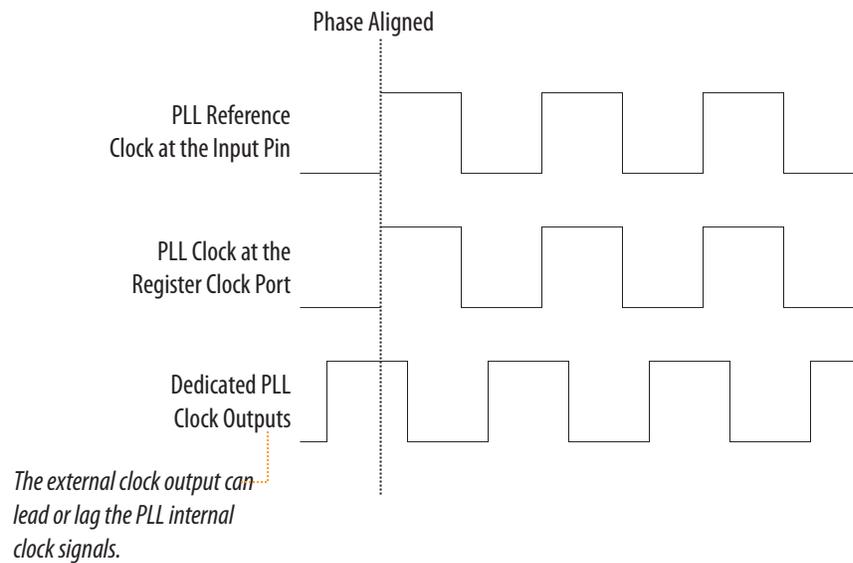
Direct Mode

In direct 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 4-28: Example of Phase Relationship Between the PLL Clocks in Direct Mode

Normal Compensation Mode

An internal clock in normal compensation 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 Intel Quartus Prime Timing Analyzer reports any phase difference between the two. In normal compensation mode, the delay introduced by the GCLK or RCLK network is fully compensated.

Figure 4-29: Example of Phase Relationship Between the PLL Clocks in Normal Compensation Mode

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. This mode is supported only on the center and corner PLLs in Stratix V devices.

When using this mode, you must use the same I/O standard on the input clocks and clock outputs to guarantee clock alignment at the input and output pins. You cannot use differential I/O standards on the PLL clock input or output pins.

To ensure phase alignment between the `clk` pin and the external clock output (`CLKOUT`) pin in ZDB mode, instantiate a bidirectional I/O pin in the design. The bidirectional I/O pin serves as the feedback path connecting the `fbout` and `fbin` ports of the PLL. The bidirectional I/O pin must always be assigned a single-ended I/O standard. 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.

Note: To avoid signal reflection when using ZDB mode, do not place board traces on the bidirectional I/O pin.

Figure 4-30: ZDB Mode in Stratix V PLLs

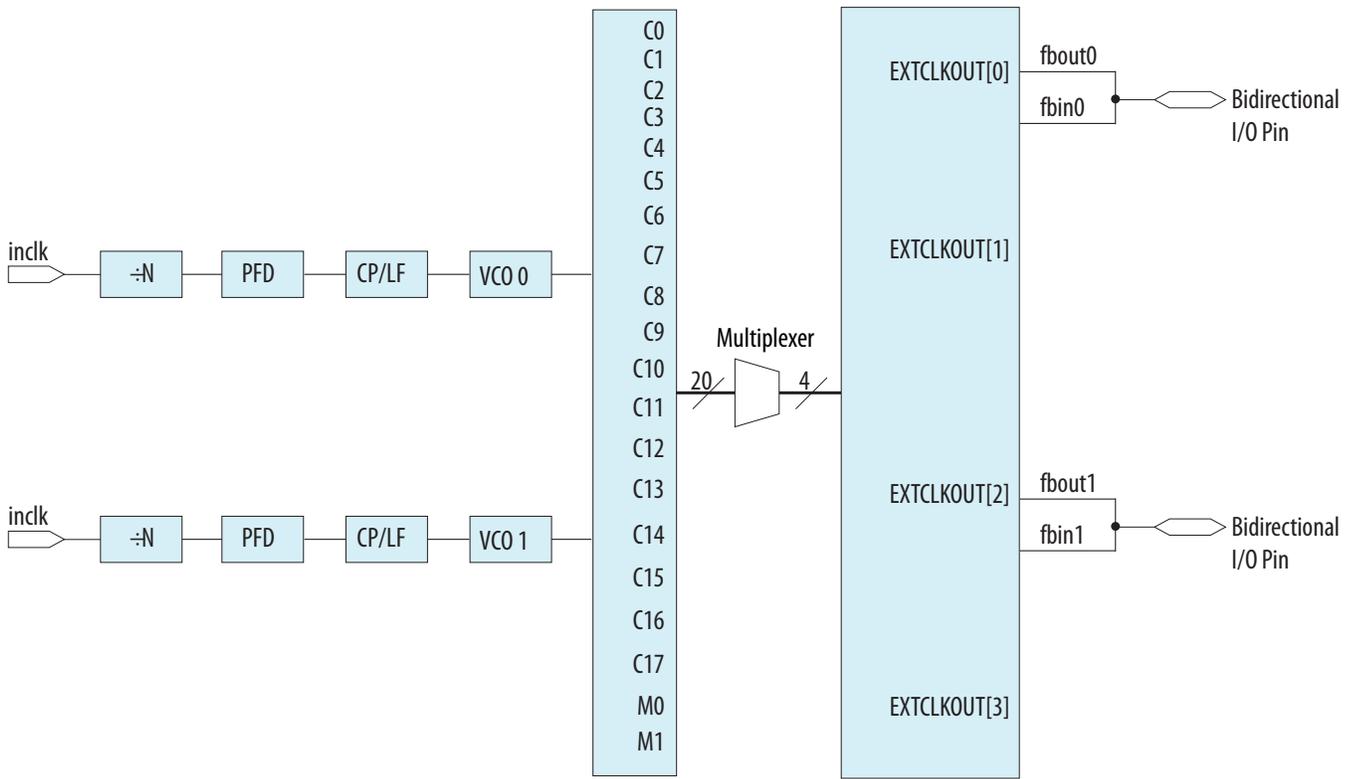
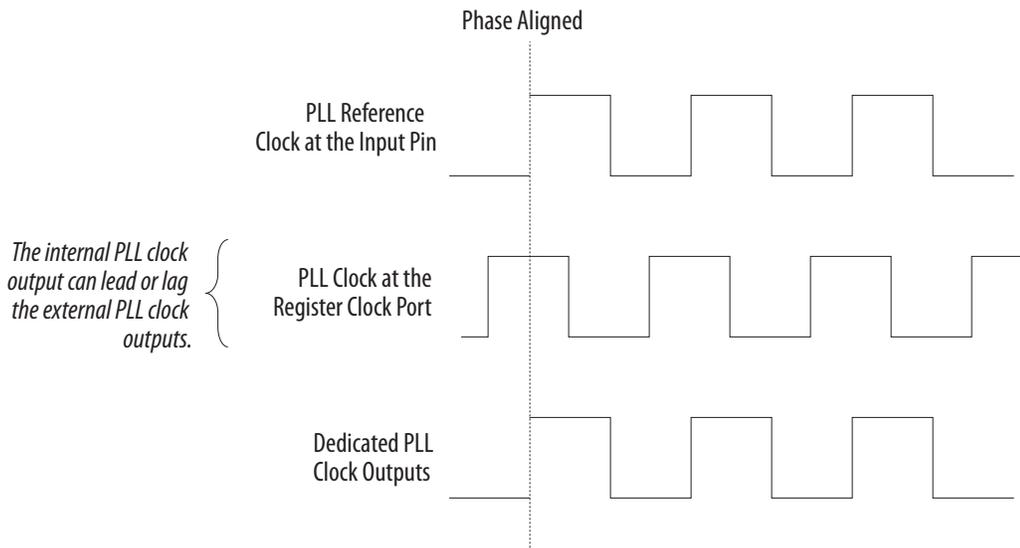


Figure 4-31: Example of Phase Relationship Between the PLL Clocks in ZDB Mode



Related Information

[PLL External Clock I/O Pins](#) on page 4-28

Provides more information about PLL clock outputs.

External Feedback Mode

In EFB mode, the output of the M counter (f_{bout}) feeds back to the PLL f_{bin} input (using a trace on the board) and becomes part of the feedback loop.

One of the dual-purpose external clock outputs becomes the f_{bin} input pin in this mode. The external feedback input pin, f_{bin} is phase-aligned with the clock input pin. Aligning these clocks allows you to remove clock delay and skew between devices.

When using EFB mode, you must use the same I/O standard on the input clock, feedback input, and clock outputs.

This mode is supported only on the center and corner fractional PLLs in Stratix V devices.

Figure 4-32: EFB Mode in Stratix V Devices

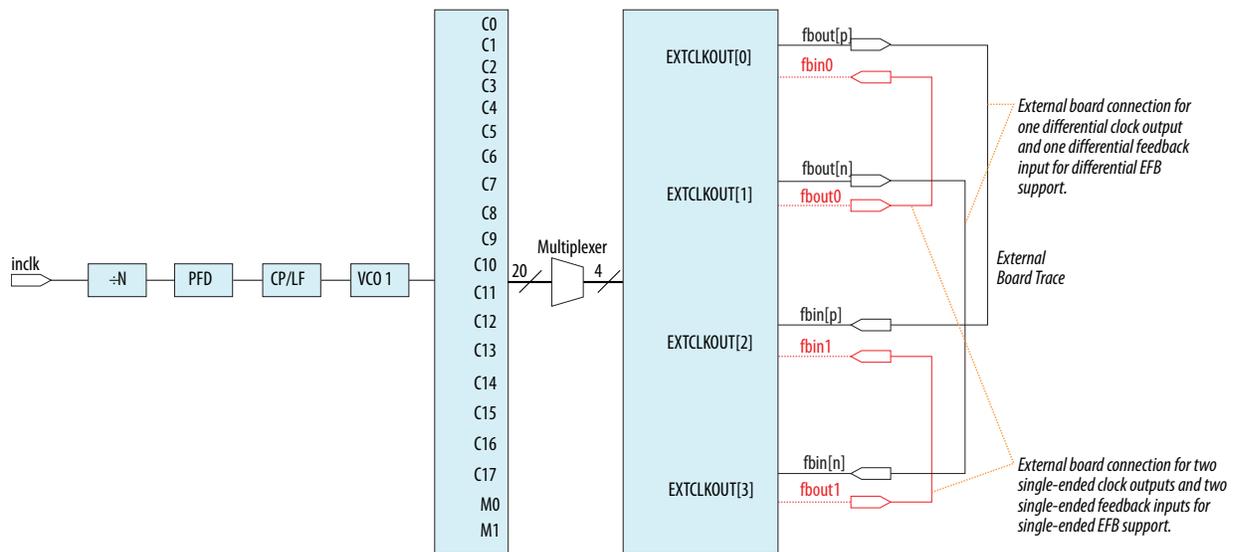
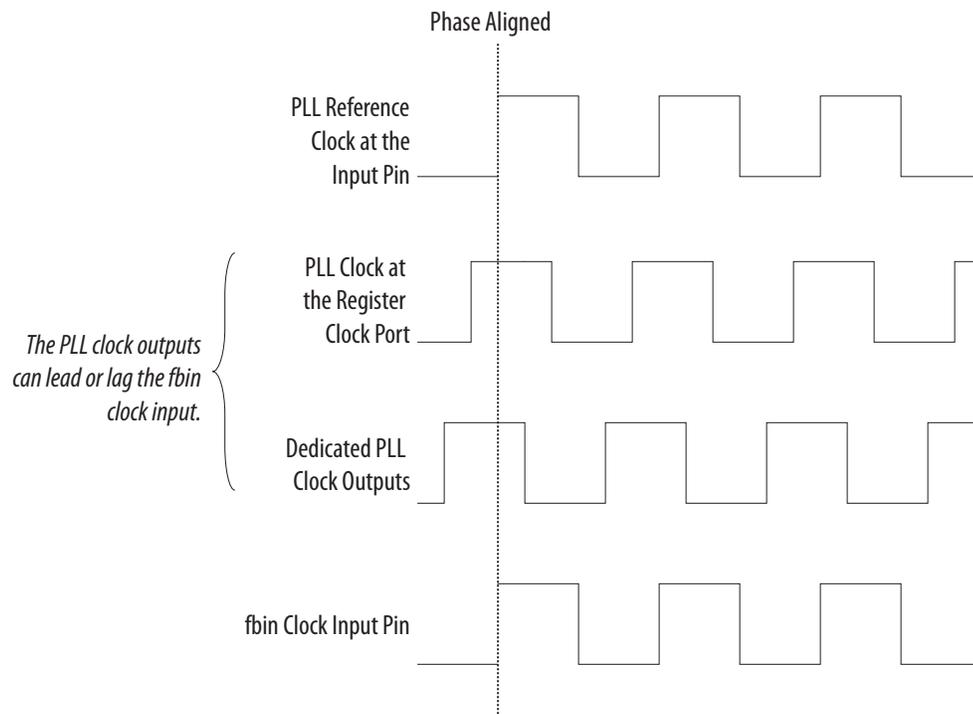


Figure 4-33: Example of Phase Relationship Between the PLL Clocks in EFB Mode**Related Information**

[PLL External Clock I/O Pins](#) on page 4-28

Provides more information about PLL clock outputs.

Multiple PLLs in Normal Mode and Source Synchronous Mode

Normal and source synchronous compensation feedback mode require GCLK or RCLK feedback path to achieve the required phase relationship. Source synchronous mode for LVDS compensation does not require the GCLK or RCLK feedback path.

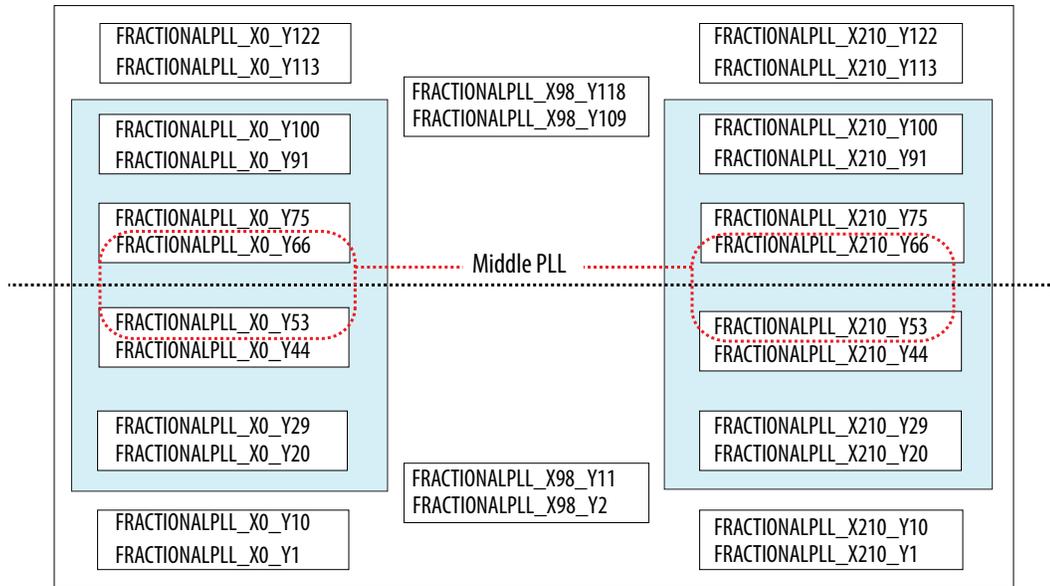
The GCLK or RCLK network feedback paths are fewer than the PLLs available on the device. You cannot implement the compensation mode that requires GCLK or RCLK feedback path on all the PLLs available on the device simultaneously.

Consider the following guidelines when implementing normal compensation or source synchronous compensation mode on multiple PLLs for the device:

- You can implement normal compensation or source synchronous compensation mode on all the center PLLs simultaneously.
- The Stratix V device has two middle PLLs on the left and right side of the device. All PLLs that reside on each side of the device can be divided equally into 2 groups as shown in the following figure.

Figure 4-34: Example of the PLL Grouping for Stratix V GX A5 and A7 Devices, and Stratix V GT C5 and C7 Devices

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.



From the PLL grouping example, the PLLs can be divided into 4 different sections (upper left, lower left, upper right, and lower right). The PLLs in each of these sections can be further divided into first and second group. The first group consists of the 2 corner PLLs and one middle PLL located in each section. The remaining PLLs in the same section are grouped into the second group. For each section, you can use up to 3 PLLs to implement source synchronous or normal compensation mode in the following combinations:

- Any of the 3 PLLs in the first group
- Any of the 2 PLLs in the first group and 1 PLL in the second group

Table 4-7: Example of the PLL Grouping for Stratix V GX A5 and A7 Devices, and Stratix V GT C5 and C7 Devices

PLL Section	PLL Location	
	First Group	Second Group
Upper left	FRACTIONALPLL_X0_Y122, FRACTIONALPLL_X0_Y113, FRACTIONALPLL_X0_Y66	FRACTIONALPLL_X0_Y100, FRACTIONALPLL_X0_Y91, FRACTIONALPLL_X0_Y75
Lower left	FRACTIONALPLL_X0_Y53, FRACTIONALPLL_X0_Y10, FRACTIONALPLL_X0_Y1	FRACTIONALPLL_X0_Y44, FRACTIONALPLL_X0_Y29, FRACTIONALPLL_X0_Y20
Upper right	FRACTIONALPLL_X210_Y122, FRACTIONALPLL_X210_Y113, FRACTIONALPLL_X210_Y66	FRACTIONALPLL_X210_Y100, FRACTIONALPLL_X210_Y91, FRACTIONALPLL_X210_Y75

PLL Section	PLL Location	
	First Group	Second Group
Lower right	FRACTIONALPLL_X210_Y53, FRACTIONALPLL_X210_Y10, FRACTIONALPLL_X210_Y1	FRACTIONALPLL_X210_Y44, FRACTIONALPLL_X210_Y29, FRACTIONALPLL_X210_Y20

Clock Multiplication and Division

Each Stratix V PLL provides clock synthesis for PLL output ports using the $M/(N \times C)$ 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 $f_{in} \times (M/N)$.

The Intel Quartus Prime software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTERA_PLL IP core.

VCO Post Divider

A VCO post divider is inserted after the VCO. When you enable the VCO post divider, the VCO post divider divides the VCO frequency by two. When the VCO post divider is bypassed, the VCO frequency goes to the output port without being divided by two.

Post-Scale Counter, c

Each output port has a unique post-scale counter, c , that divides down the output from the VCO post divider. 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 Intel Quartus Prime 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, c , scale down the VCO frequency for each output port.

Pre-Scale Counter, N and Multiply Counter, M

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 have 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.

Delta-Sigma Modulator

The delta-sigma modulator (DSM) is used together with the M multiply counter to enable the PLL to operate in fractional mode. The DSM dynamically changes the M counter divide value on a cycle to cycle basis. The different M counter values allow the "average" M counter value to be a non-integer.

Fractional Mode

In fractional mode, the M counter divide value equals to the sum of the "clock high" count, "clock low" count, and the fractional value. The fractional value is equal to $\kappa/2^X$, where κ is an integer between 0 and $(2^X - 1)$, and $X = 8, 16, 24, \text{ or } 32$.

Integer Mode

For PLL operating in integer mode, m is an integer value and DSM is disabled.

Related Information

[Altera Phase-Locked Loop \(Altera PLL\) IP Core User Guide](#)

Provides more information about PLL software support in the Quartus Prime software.

Programmable Phase Shift

The programmable phase shift feature allows the PLLs to generate output clocks with a fixed phase offset.

The VCO frequency of the PLL determines the precision of the phase shift. The minimum phase shift increment is $1/8$ of the VCO period. For example, if a PLL operates with a VCO frequency of 1000 MHz, phase shift steps of 125 ps are possible.

The Intel Quartus Prime software automatically adjusts the VCO frequency according to the user-specified phase shift values entered into the IP core.

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 Intel Quartus Prime 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.

Clock Switchover

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

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

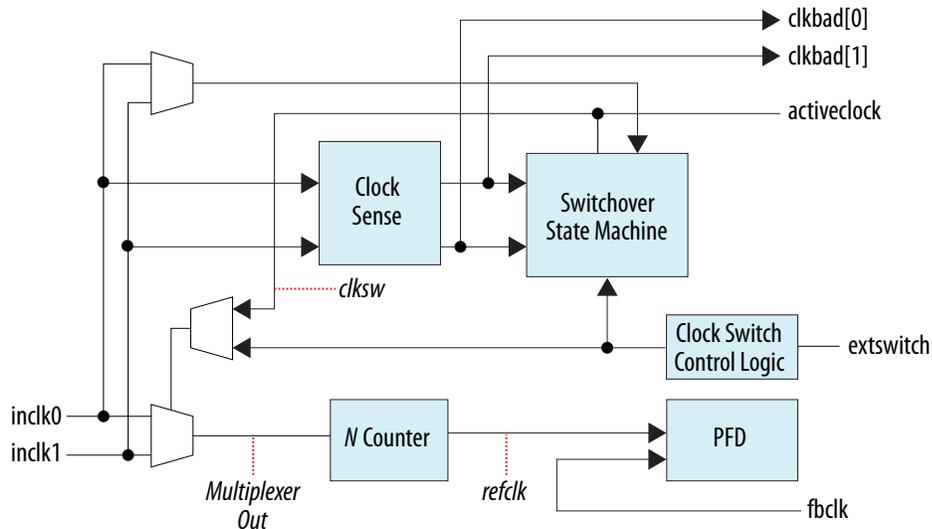
- Automatic switchover—The clock sense circuit monitors the current reference clock. If the current reference clock stops toggling, the reference clock automatically switches to `inclk0` or `inclk1` clock.
- Manual clock switchover—Clock switchover is controlled using the `extswitch` signal. When the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

Automatic Switchover

Stratix V PLLs support a fully configurable clock switchover capability.

Figure 4-35: Automatic Clock Switchover Circuit Block Diagram

This figure shows a block diagram of the automatic switchover circuit built into the PLL.



When the current reference clock is not present, the clock sense block automatically switches to the backup clock for PLL reference. You can select a clock source as the backup clock by connecting it to the `inclk1` port of the PLL in your design.

The clock switchover circuit sends out three status signals—`clkbad[0]`, `clkbad[1]`, and `activeclock`—from the PLL to implement a custom switchover circuit in the logic array.

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

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

Note: Glitches in the input clock may cause the frequency difference between the input clocks to be more than 20%.

Use the switchover circuitry to automatically switch between `inclk0` and `inclk1` when the current reference clock to the PLL stops toggling. You can switch back and forth between `inclk0` and `inclk1` any number of times when one of the two clocks fails and the other clock is available.

For example, in applications that require a redundant clock with the same frequency as the reference clock, the switchover state machine generates a signal (`clksw`) that controls the multiplexer select input. In this case, `inclk1` becomes the reference clock for the PLL.

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

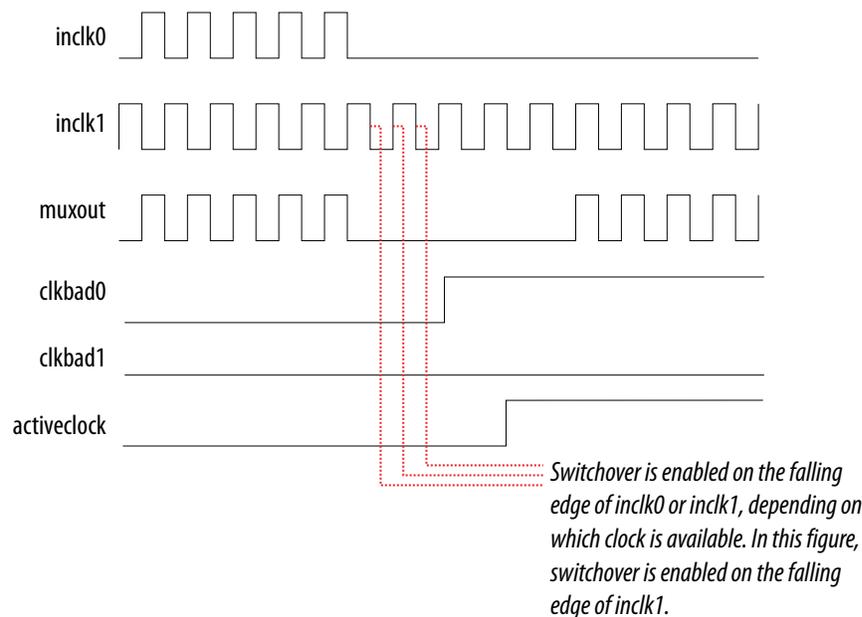
- Both clock inputs must be running when the FPGA is configured.
- The period of the two clock inputs can differ by no more than 20%.

If the current clock input stops toggling while the other clock is also not toggling, switchover is not initiated and the `clkbad[0..1]` signals are not valid. If both clock inputs are not the same frequency, but their period difference is within 20%, the clock sense block detects when a clock stops toggling. However, the PLL may lose lock after the switchover is completed and needs time to relock.

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

Figure 4-36: Automatic Switchover After Loss of Clock Detection

This figure shows an example waveform of the switchover feature in automatic switchover mode. In this example, the `inclk0` signal is stuck low. After the `inclk0` signal is stuck at low for approximately two clock cycles, the clock sense circuitry drives the `clkbad[0]` signal high. Since the reference clock signal is not toggling, the switchover state machine controls the multiplexer through the `extswitch` signal to switch to the backup clock, `inclk1`.



Automatic Switchover with Manual Override

In automatic switchover with manual override mode, you can use the `extswitch` signal for user- or system-controlled switch conditions. You can use this mode for same-frequency switchover, or to switch between inputs of different frequencies.

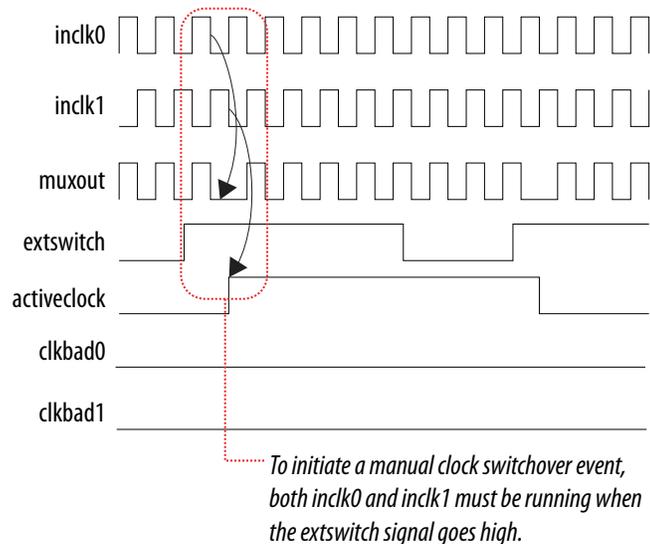
For example, if `inclk0` is 66 MHz and `inclk1` is 200 MHz, you must control switchover using the `extswitch` signal. The automatic clock-sense circuitry cannot monitor clock input (`inclk0` and `inclk1`) frequencies with a frequency difference of more than 100% (2×).

This feature is useful when the clock sources originate from multiple cards on the backplane, requiring a system-controlled switchover between the frequencies of operation.

You must choose the backup clock frequency and set the M , N , C , and K counters so that the VCO operates within the recommended operating frequency range. The ALTERA_PLL IP Catalog notifies you if a given combination of `inclk0` and `inclk1` frequencies cannot meet this requirement.

Figure 4-37: Clock Switchover Using the `extswitch` (Manual) Control

This figure shows a clock switchover waveform controlled by the `extswitch` signal. In this case, both clock sources are functional and `inclk0` is selected as the reference clock; the `extswitch` signal goes high, which starts the switchover sequence. On the falling edge of `inclk0`, the counter's reference clock, `muxout`, is gated off to prevent clock glitching. On the falling edge of `inclk1`, the reference clock multiplexer switches from `inclk0` to `inclk1` as the PLL reference. The `activeclock` signal changes to indicate the clock which is currently feeding the PLL.



In automatic override with manual switchover mode, the `activeclock` signal mirrors the `extswitch` signal. Since both clocks are still functional during the manual switch, neither `clkbad` signal goes high. Because the switchover circuit is positive-edge sensitive, the falling edge of the `extswitch` signal does not cause the circuit to switch back from `inclk1` to `inclk0`. When the `extswitch` signal goes high again, the process repeats.

The `extswitch` signal and automatic switch work only if the clock being switched to is available. If the clock is not available, the state machine waits until the clock is available.

Related Information

[Altera Phase-Locked Loop \(Altera PLL\) IP Core User Guide](#)

Provides more information about PLL software support in the Quartus Prime software.

Manual Clock Switchover

In manual clock switchover mode, the `extswitch` signal controls whether `inclk0` or `inclk1` is selected as the input clock to the PLL. By default, `inclk0` is selected.

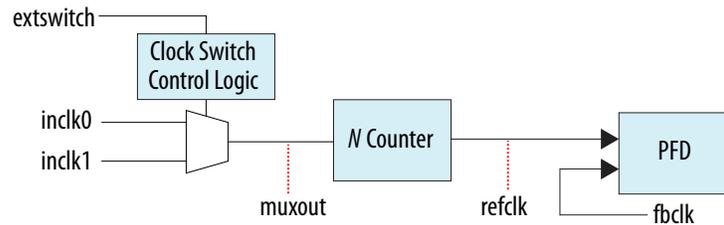
A clock switchover event is initiated when the `extswitch` signal transitions from logic low to logic high, and being held high for at least three `inclk` cycles.

You must bring the `extswitch` signal back low again for PLL to re-gain lock. If you do not require another switchover event, you can leave the `extswitch` signal in a logic low state.

Pulsing the `extswitch` signal high for at least three `inclk` cycles performs another switchover event.

If `inclk0` and `inclk1` are different frequencies and are always running, the `extswitch` signal minimum high time must be greater than or equal to three of the slower frequency `inclk0` and `inclk1` cycles.

Figure 4-38: Manual Clock Switchover Circuitry in Stratix V PLLs



You can delay the clock switchover action by specifying the switchover delay in the `ALTERA_PLL` IP core. When you specify the switchover delay, the `extswitch` signal must be held high for at least three `inclk` cycles plus the number of the delay cycles that has been specified to initiate a clock switchover.

Related Information

[Altera Phase-Locked Loop \(Altera PLL\) IP Core User Guide](#)

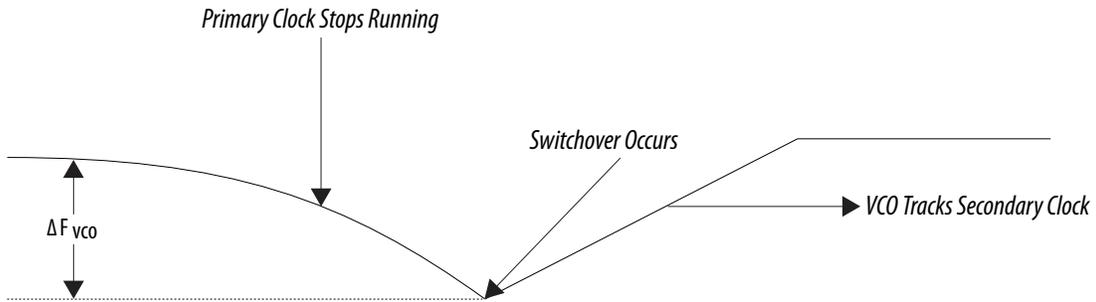
Provides more information about PLL software support in the Quartus Prime software.

Guidelines

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

- Automatic clock switchover requires that the `inclk0` and `inclk1` frequencies be within 20% of each other. Failing to meet this requirement causes the `clkbad[0]` and `clkbad[1]` signals to not function properly.
- When using manual clock switchover, the difference between `inclk0` and `inclk1` can be more than 100% (2×). However, differences in frequency, phase, or both, of the two clock sources will likely cause the PLL to lose lock. Resetting the PLL ensures that you maintain the correct phase relationships between the input and output clocks.
- Both `inclk0` and `inclk1` must be running when the `extswitch` signal goes high to initiate the manual clock switchover event. Failing to meet this requirement causes the clock switchover to not function properly.
- Applications that require a clock switchover feature and a small frequency drift must use a low-bandwidth PLL. When referencing input clock changes, the low-bandwidth PLL reacts more slowly than a high-bandwidth PLL. When switchover happens, a low-bandwidth PLL propagates the stopping of the clock to the output more slowly than a high-bandwidth PLL. However, be aware that the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The time it takes for the PLL to rellock depends on the PLL configuration.
- The phase relationship between the input clock to the PLL and the output clock from the PLL is important in your design. Assert `areset` for at least 10 ns after performing a clock switchover. Wait for the locked signal to go high and be stable before re-enabling the output clocks from the PLL.
- The VCO frequency gradually decreases when the current clock is lost and then increases as the VCO locks on to the backup clock, as shown in the following figure.

Figure 4-39: VCO Switchover Operating Frequency



PLL Reconfiguration and Dynamic Phase Shift

For more information about PLL reconfiguration and dynamic phase shifting, refer to AN661.

Related Information

[AN 661: Implementing Fractional PLL Reconfiguration with Altera PLL and Altera PLL Reconfig IP Cores](#)

Clock Networks and PLLs in Stratix V Devices Revision History

Document Version	Changes
2019.04.26	<ul style="list-style-type: none"> Corrected the signal name from <code>clkswitch</code> to <code>extswitch</code>. Updated the description for the automatic switchover with manual override mode in the <i>Clock Switchover</i> section. Updated the description about the <code>extswitch</code> signal in the <i>Manual Clock Switchover</i> section.

Date	Version	Changes
December 2016	2016.12.09	Added a note to dedicated <code>refclk</code> pin in Fractional PLL High-Level Block Diagram.
December 2015	2015.12.21	Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i> .

Date	Version	Changes
January 2014	2014.01.10	<ul style="list-style-type: none"> Removed Preliminary tags for clock resources, clock input pin connections to GCLK and RCLK networks, and PLL features tables. Updated information on dual-regional clock region. Added label for PLL strip in PLL locations diagrams. Added descriptions for PLLs located in a strip. Updated VCO post-scale counter, κ, to VCO post divider. Added information on PLL cascading. Added information on programmable phase shift. Updated automatic clock switchover mode requirement.
May 2013	2013.05.06	<ul style="list-style-type: none"> Added link to the known document issues in the Knowledge Base. Updated PCLK clock sources per device quadrant. Added PCLK networks resources and diagram for Stratix V E devices. Updated PCLK clock sources in hierarchical clock networks in each spine clock per quadrant diagram. Added PCLK networks in clock network sources section. Updated dedicated clock input pins in clock network sources section. Added information on C output counters for PLLs. Added power down mode in PLL features table. Added information on PLL physical counters. Updated the PLL locations index from CEN_X<#>_Y<#>, COR_X<#>_Y<#>, and LR_X<#>_Y<#> to FRACTIONALPLL_X<#>_Y<#>. Removed LVPECL I/O standard support for clock output pin pairs. Updated PLL support for EFB mode. Updated the scaling factors for PLL output ports. Updated the fractional value for PLL in fractional mode. Moved all links to the Related Information section of respective topics for easy reference. Reorganized content.
December 2012	2012.12.28	<ul style="list-style-type: none"> Added note to indicate that the figures shown are the top view of the silicon die. Added diagram for PLL physical counter orientation. Updated PLL locations diagrams. Removed information on pfdena PLL control signal. Removed information on PLL Compensation assignment in the Quartus II software. Updated the fractional value for PLL in fractional mode. Reorganized content and updated template.

Date	Version	Changes
June 2012	1.4	<ul style="list-style-type: none"> Added Table 4–5 and Table 4–6. Added Figure 4–6, Figure 4–8, Figure 4–20, Figure 4–22, and Figure 4–33. Updated Table 4–1, Table 4–2, and Table 4–3. Updated Figure 4–3, Figure 4–5, Figure 4–17, Figure 4–18, Figure 4–19, and Figure 4–21. Added “PLL Migration Guidelines”, “Implementing Multiple PLLs in Normal Mode and Source Synchronous Mode”, “Clock Switchover”, and “PLL Reconfiguration and Dynamic Phase Shift” sections. Updated “Clock Networks in Stratix V Devices”, “Clock Network Sources”, and “Clock Multiplication and Division” sections.
November 2011	1.3	Updated Figure 4–19 and Figure 4–28.
May 2011	1.2	<ul style="list-style-type: none"> 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.

2020.09.10

SV51006



Subscribe



Send Feedback

This chapter provides details about the features of the Stratix V I/O elements (IOEs) and how the IOEs work in compliance with current and emerging I/O standards and requirements.

The Stratix V I/Os support the following features:

- 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
- Low-voltage differential signaling (LVDS), RSDS, mini-LVDS, HSTL, HSUL, and SSTL I/O standards across all I/O banks
- Double data rate (DDR), single data rate (SDR), and half data rate input and output options
- Serializer/deserializer (SERDES)
- Deskew, read and write leveling, and clock-domain crossing functionality for high-performance memory interface
- Programmable output current strength
- Programmable slew rate
- Programmable bus-hold
- Programmable pull-up resistor
- Programmable pre-emphasis
- Programmable I/O delay
- Programmable voltage output differential (V_{OD})
- Open-drain output
- On-chip series termination (R_S OCT) with and without calibration
- On-chip parallel termination (R_T OCT)
- On-chip differential termination (R_D OCT)

Note: The information in this chapter is applicable to all Stratix V variants, unless noted otherwise.

Related Information

[Stratix V Device Handbook: Known Issues](#)

Lists the planned updates to the *Stratix V Device Handbook* chapters.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

I/O Standards Support in Stratix V Devices

This section lists the I/O standards supported in the FPGA I/Os of Stratix V devices, the typical power supply values for each I/O standard, and the MultiVolt I/O interface feature.

I/O Standards Support in Stratix V Devices

Stratix V devices support a wide range of industry I/O standards. 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: Supported I/O Standards for Stratix V Devices

This table lists the I/O standards for Stratix V devices, as well as the typical applications they support.

I/O Standard	Typical Applications	Standard Support
3.3 V LVTTTL/3.3 V LVCMOS ⁽⁴⁾	General purpose	JESD8-B
2.5 V LVCMOS	General purpose	JESD8-5
1.8 V LVCMOS	General purpose	JESD8-7
1.5 V LVCMOS	General purpose	JESD8-11
1.2 V LVCMOS	General purpose	JESD8-12
SSTL-2 Class I	DDR SDRAM	JESD8-9B
SSTL-2 Class II	DDR SDRAM	JESD8-9B
SSTL-18 Class I	DDR2 SDRAM	JESD8-15
SSTL-18 Class II	DDR2 SDRAM	JESD8-15
SSTL-15 Class I	DDR3 SDRAM	—
SSTL-15 Class II	DDR3 SDRAM	—
1.8 V HSTL Class I	QDR II/RLDRAM II	JESD8-6
1.8 V HSTL Class II	QDR II/RLDRAM II	JESD8-6
1.5 V HSTL Class I	QDR II/QDR II+/ RLDRAM II	JESD8-6
1.5 V HSTL Class II	QDR II/QDR II+/ RLDRAM II	JESD8-6
1.2 V HSTL Class I	General purpose	JESD8-16A
1.2 V HSTL Class II	General purpose	JESD8-16A
Differential SSTL-2 Class I	DDR SDRAM	JESD8-9B
Differential SSTL-2 Class II	DDR SDRAM	JESD8-9B
Differential SSTL-18 Class I	DDR2 SDRAM	JESD8-15
Differential SSTL-18 Class II	DDR2 SDRAM	JESD8-15
Differential SSTL-15 Class I	DDR3 SDRAM	—

⁽⁴⁾ Supported using V_{CCIO} at 3.0 V.

I/O Standard	Typical Applications	Standard Support
Differential SSTL-15 Class II	DDR3 SDRAM	—
Differential 1.8 V HSTL Class I	Clock interfaces	JESD8-6
Differential 1.8 V HSTL Class II	Clock interfaces	JESD8-6
Differential 1.5 V HSTL Class I	Clock interfaces	JESD8-6
Differential 1.5 V HSTL Class II	Clock interfaces	JESD8-6
Differential 1.2 V HSTL Class I	Clock interfaces	JESD8-16A
Differential 1.2 V HSTL Class II	Clock interfaces	JESD8-16A
LVDS	High-speed communications	ANSI/TIA/EIA-644
RSDS	Flat panel display	—
Mini-LVDS	Flat panel display	—
LVPECL	Video graphics and clock distribution	—
SSTL-15	DDR3 SDRAM	JESD79-3D
SSTL-135	DDR3L SDRAM	—
SSTL-125	DDR3U SDRAM	—
SSTL-12	RLDRAM 3	—
HSUL-12	LPDDR2 SDRAM	—
Differential SSTL-15	DDR3 SDRAM	JESD79-3D
Differential SSTL-135	DDR3L SDRAM	—
Differential SSTL-125	DDR3U SDRAM	—
Differential SSTL-12	RLDRAM 3	—
Differential HSUL-12	LPDDR2 SDRAM	—

I/O Standards Voltage Levels in Stratix V Devices

Table 5-2: Stratix V I/O Standards Voltage Levels

This table lists the typical power supplies for each supported I/O standards in Stratix V devices.

I/O Standard	V _{CCIO} (V)		V _{CCPD} (V) (Pre-Driver Voltage)	V _{REF} (V) (Input Ref Voltage)	V _{TT} (V) (Board Termination Voltage)
	Input ⁽⁵⁾	Output			
3.3 V LVTTL/3.3 V LVCMOS	3.0/2.5	3.0	3.0	—	—

⁽⁵⁾ Input buffers for the SSTL, HSTL, Differential SSTL, Differential HSTL, LVDS, RSDS, Mini-LVDS, LVPECL, HSUL, and Differential HSUL are powered by V_{CCPD}

I/O Standard	V_{CCIO} (V)		V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
	Input ⁽⁵⁾	Output			
2.5 V LVCMOS	3.0/2.5	2.5	2.5	—	—
1.8 V LVCMOS	1.8/1.5	1.8	2.5	—	—
1.5 V LVCMOS	1.8/1.5	1.5	2.5	—	—
1.2 V LVCMOS	1.2	1.2	2.5	—	—
SSTL-2 Class I	V_{CCPD}	2.5	2.5	1.25	1.25
SSTL-2 Class II	V_{CCPD}	2.5	2.5	1.25	1.25
SSTL-18 Class I	V_{CCPD}	1.8	2.5	0.9	0.9
SSTL-18 Class II	V_{CCPD}	1.8	2.5	0.9	0.9
SSTL-15 Class I	V_{CCPD}	1.5	2.5	0.75	0.75
SSTL-15 Class II	V_{CCPD}	1.5	2.5	0.75	0.75
1.8 V HSTL Class I	V_{CCPD}	1.8	2.5	0.9	0.9
1.8 V HSTL Class II	V_{CCPD}	1.8	2.5	0.9	0.9
1.5 V HSTL Class I	V_{CCPD}	1.5	2.5	0.75	0.75
1.5 V HSTL Class II	V_{CCPD}	1.5	2.5	0.75	0.75
1.2 V HSTL Class I	V_{CCPD}	1.2	2.5	0.6	0.6
1.2 V HSTL Class II	V_{CCPD}	1.2	2.5	0.6	0.6
Differential SSTL-2 Class I	V_{CCPD}	2.5	2.5	—	1.25
Differential SSTL-2 Class II	V_{CCPD}	2.5	2.5	—	1.25
Differential SSTL-18 Class I	V_{CCPD}	1.8	2.5	—	0.9
Differential SSTL-18 Class II	V_{CCPD}	1.8	2.5	—	0.9
Differential SSTL-15 Class I	V_{CCPD}	1.5	2.5	—	0.75
Differential SSTL-15 Class II	V_{CCPD}	1.5	2.5	—	0.75
Differential 1.8 V HSTL Class I	V_{CCPD}	1.8	2.5	—	0.9
Differential 1.8 V HSTL Class II	V_{CCPD}	1.8	2.5	—	0.9

⁽⁵⁾ Input buffers for the SSTL, HSTL, Differential SSTL, Differential HSTL, LVDS, RSDS, Mini-LVDS, LVPECL, HSUL, and Differential HSUL are powered by V_{CCPD}

I/O Standard	V_{CCIO} (V)		V_{CCPD} (V) (Pre-Driver Voltage)	V_{REF} (V) (Input Ref Voltage)	V_{TT} (V) (Board Termination Voltage)
	Input ⁽⁵⁾	Output			
Differential 1.5 V HSTL Class I	V_{CCPD}	1.5	2.5	—	0.75
Differential 1.5 V HSTL Class II	V_{CCPD}	1.5	2.5	—	0.75
Differential 1.2 V HSTL Class I	V_{CCPD}	1.2	2.5	—	0.6
Differential 1.2 V HSTL Class II	V_{CCPD}	1.2	2.5	—	0.6
LVDS	V_{CCPD}	2.5	2.5	—	—
RSDS	V_{CCPD}	2.5	2.5	—	—
Mini-LVDS	V_{CCPD}	2.5	2.5	—	—
LVPECL (Differential clock input only)	V_{CCPD}	—	2.5	—	—
SSTL-15	V_{CCPD}	1.5	2.5	0.75	Typically does not require board termination
SSTL-135	V_{CCPD}	1.35	2.5	0.675	
SSTL-125	V_{CCPD}	1.25	2.5	0.625	
SSTL-12	V_{CCPD}	1.2	2.5	0.6	
HSUL-12	V_{CCPD}	1.2	2.5	0.6	
Differential SSTL-15	V_{CCPD}	1.5	2.5	—	Typically does not require board termination
Differential SSTL-135	V_{CCPD}	1.35	2.5	—	
Differential SSTL-125	V_{CCPD}	1.25	2.5	—	
Differential SSTL-12	V_{CCPD}	1.2	2.5	—	
Differential HSUL-12	V_{CCPD}	1.2	2.5	—	

The 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.

Related Information

Guideline: Observe Device Absolute Maximum Rating for 3.3 V Interfacing on page 5-8

Provides more information about the 3.3 V LVTTTL/LVCMOS I/O standard supported in Stratix V devices.

⁽⁵⁾ Input buffers for the SSTL, HSTL, Differential SSTL, Differential HSTL, LVDS, RSDS, Mini-LVDS, LVPECL, HSUL, and Differential HSUL are powered by V_{CCPD}

MultiVolt I/O Interface in Stratix V Devices

The MultiVolt I/O interface feature allows Stratix V devices in all packages to interface with systems of different supply voltages.

You can connect the V_{CCIO} 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 V_{CCIO} pins are connected to a 1.5 V power supply, the output levels are compatible with 1.5 V systems.

For LVDS applications:

- 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.

Table 5-3: MultiVolt I/O Support in Stratix V Devices

V_{CCIO} (V)	V_{CCPD} (V) ⁽⁶⁾	Input Signal (V)	Output Signal (V)
1.2	2.5	1.2	1.2
1.25	2.5	1.25	1.25
1.35	2.5	1.35	1.35
1.5	2.5	1.5, 1.8	1.5
1.8	2.5	1.5, 1.8	1.8
2.5	2.5	2.5, 3.0, 3.3	2.5
3.0	3.0	2.5, 3.0, 3.3	3.0, 3.3

The pin current may be slightly higher than the default value. Verify that the V_{OL} maximum and V_{OH} minimum voltages of the driving device do not violate the applicable V_{IL} maximum and V_{IH} minimum voltage specifications of the Stratix V device.

The V_{CCPD} power pins must be connected to a 2.5 V 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.

Note: If the input signal is 3.0 V or 3.3 V, Altera recommends that you use an external clamping diode on the I/O pins.

I/O Design Guidelines for Stratix V Devices

There are several considerations that require your attention to ensure the success of your designs. Unless noted otherwise, these design guidelines apply to all variants of this device family.

Mixing Voltage-Referenced and Non-Voltage-Referenced I/O Standards

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 the devices.

⁽⁶⁾ Input buffers for the SSTL, HSTL, Differential SSTL, Differential HSTL, LVDS, RSDS, Mini-LVDS, LVPECL, HSUL, and Differential HSUL are powered by V_{CCPD}

Non-Voltage-Referenced I/O Standards

Each Stratix V I/O bank has its own V_{CCIO} pins and supports only one V_{CCIO} of 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 the I/O standards support the V_{CCIO} level and V_{CCPD} requirement of the I/O bank.

For output signals, a single I/O bank supports non-voltage-referenced output signals that drive at the same voltage as V_{CCIO} . Because an I/O bank can only have one V_{CCIO} value, it can only drive out the 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, 3.0 V and 3.3 V inputs but supports only 2.5 V output.

Voltage-Referenced I/O Standards

To accommodate voltage-referenced I/O standards:

- Each Stratix V I/O bank supports multiple dedicated V_{REF} pins feeding a common V_{REF} bus.
- Each bank can have only a single V_{CCIO} voltage level and a single voltage reference (V_{REF}) level.

An I/O bank featuring single-ended or differential standards can support different 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 V_{CCIO} voltage of the I/O bank. For example, you can place only SSTL-2 output pins in an I/O bank with a 2.5 V V_{CCIO} .

Mixing Voltage-Referenced and Non-Voltage Referenced I/O Standards

An I/O bank can support voltage-referenced and non-voltage-referenced pins by applying each of the rule sets individually.

Examples:

- An I/O bank can support SSTL-18 inputs and outputs, and 1.8 V inputs and outputs with a 1.8 V V_{CCIO} and a 0.9 V V_{REF} .
- An I/O bank can support 1.5 V standards, 1.8 V inputs (but not outputs), and HSTL and 1.5 V HSTL I/O standards with a 1.5 V V_{CCIO} and 0.75 V V_{REF} .

Guideline: Use the Same V_{CCPD} for All I/O Banks in a Group

One V_{CCPD} is shared in a group of I/O banks. If one I/O bank in a group uses 3.0 V V_{CCPD} , other I/O banks in the same group must also use 3.0 V V_{CCPD} .

The I/O banks with the same bank number form a group. For example, I/O banks 7A, 7B, 7C, and 7D form a group and share the same V_{CCPD} . This sharing is applicable to all I/O banks, with the following exceptions:

- I/O banks 3A and 3B form a group with one V_{CCPD} .
- I/O banks 3C, 3D, and 3E (if available) form another group with its own V_{CCPD} .

If you are using an output or bidirectional pin with the 3.3 V LVTTTL or 3.3 V LVCMOS I/O standard, you must adhere to this restriction manually with location assignments.

Related Information

- [Modular I/O Banks for Stratix V E Devices](#) on page 5-10
- [Modular I/O Banks for Stratix V GX Devices](#) on page 5-11
- [Modular I/O Banks for Stratix V GS Devices](#) on page 5-14
- [Modular I/O Banks for Stratix V GT Devices](#) on page 5-15

Guideline: Observe Device Absolute Maximum Rating for 3.3 V Interfacing

To ensure device reliability and proper operation when you use the device for 3.3 V I/O interfacing, do not violate the absolute maximum ratings of the device. For more information about absolute maximum rating and maximum allowed overshoot during transitions, refer to the device datasheet.

Tip: Perform IBIS or SPICE simulations to make sure the overshoot and undershoot voltages are within the specifications.

Transmitter Application

If you use the Stratix V device as a transmitter, use slow slew rate and series termination to limit the 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.

Receiver Application

If you use the Stratix V device as a receiver, use an off-chip clamping diode to limit the overshoot and undershoot voltage at the I/O pins.

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 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.

Related Information

- [Stratix V Device Datasheet](#)
- [Stratix V Device Datasheet](#)

Guideline: Use PLL Integer Mode for LVDS Applications

For LVDS applications, you must use the phase-locked loops (PLLs) in integer PLL mode.

Related Information

- [Guideline: Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8
- [High-Speed Differential I/O Interfaces and DPA in Stratix V Devices](#)
Provides more information about LVDS usage.

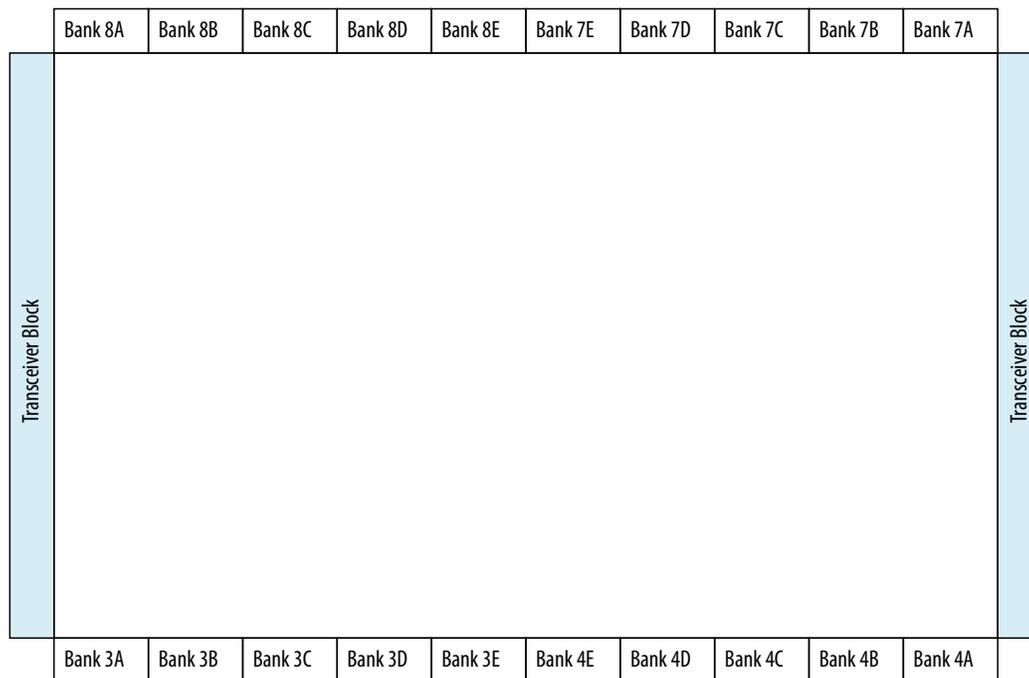
I/O Banks in Stratix V Devices

All I/O banks in Stratix V devices contain true differential input and output buffers and dedicated circuitry to support differential I/O standards:

- The number of I/O banks in a particular device depends on the device density.
- Each I/O bank 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

This figure represents the top view of the silicon die that corresponds to a reverse view of the device package.



Related Information

- [Modular I/O Banks for Stratix V E Devices](#) on page 5-10
- [Modular I/O Banks for Stratix V GX Devices](#) on page 5-11
- [Modular I/O Banks for Stratix V GS Devices](#) on page 5-14
- [Modular I/O Banks for Stratix V GT Devices](#) on page 5-15

I/O Banks Groups in Stratix V Devices

The I/O pins in Stratix V devices are arranged in groups called modular I/O banks:

- Modular I/O banks have independent power supplies that allow each bank to support different I/O standards.
- Each modular I/O bank can support multiple I/O standards that use the same V_{CCIO} and V_{CCPD} voltages.

Modular I/O Banks for Stratix V E Devices

Table 5-4: Modular I/O Banks for Stratix V E Devices

Member Code		E9		EB	
Package		H40	F45	H40	F45
Bank	3A	36	36	36	36
	3B	48	48	48	48
	3C	48	48	48	48
	3D	48	48	48	48
	3E	—	36	—	36
	4A	24	24	24	24
	4B	48	48	48	48
	4C	48	48	48	48
	4D	48	48	48	48
	4E	—	36	—	36
	7A	24	24	24	24
	7B	48	48	48	48
	7C	48	48	48	48
	7D	48	48	48	48
	7E	—	36	—	36
	8A	36	36	36	36
	8B	48	48	48	48
	8C	48	48	48	48
	8D	48	48	48	48
	8E	—	36	—	36
Total		696	840	696	840

Related Information

- **I/O Banks in Stratix V Devices** on page 5-9
- **Guideline: Use the Same VCCPD for All I/O Banks in a Group** on page 5-7
Provides guidelines about V_{CCPD} and I/O banks groups.

Modular I/O Banks for Stratix V GX Devices**Table 5-5: Modular I/O Banks for Stratix V GX A3 and A4 Devices**

Member Code		A3				A4		
Package		EH29	HF35	KF35	KF40	HF35	KF35	KF40
Bank	3A	36	36	36	36	36	36	36
	3B	48	48	48	48	48	48	48
	3C	—	—	—	48	48	—	48
	3D	24	24	24	48	24	24	48
	4A	24	24	24	24	24	24	24
	4B	—	48	48	48	48	48	48
	4C	—	—	—	48	48	—	48
	4D	24	36	36	48	24	36	48
	7A	24	24	24	24	24	24	24
	7B	—	48	48	48	48	48	48
	7C	48	48	48	48	48	48	48
	7D	36	36	36	48	36	36	48
	8A	24	24	24	36	24	24	36
	8B	—	—	—	48	—	—	48
	8C	48	—	—	48	48	—	48
	8D	24	36	36	48	24	36	48
Total		360	432	432	696	552	432	696

Table 5-6: Modular I/O Banks for Stratix V GX A5 and A7 Devices

Member Code		A5					A7				
Package		HF35	KF35	KF40	NF40	NF45	HF35	KF35	KF40	NF40	NF45
Bank	3A	36	36	36	36	36	36	36	36	36	36
	3B	48	48	48	48	48	48	48	48	48	48
	3C	48	—	48	48	48	48	—	48	48	48
	3D	24	24	48	24	48	24	24	48	24	48
	3E	—	—	—	—	36	—	—	—	—	36
	4A	24	24	24	24	24	24	24	24	24	24
	4B	48	48	48	48	48	48	48	48	48	48
	4C	48	—	48	48	48	48	—	48	48	48
	4D	24	36	48	24	48	24	36	48	24	48
	4E	—	—	—	—	36	—	—	—	—	36
	7A	24	24	24	24	24	24	24	24	24	24
	7B	48	48	48	48	48	48	48	48	48	48
	7C	48	48	48	48	48	48	48	48	48	48
	7D	36	36	48	48	48	36	36	48	48	48
	7E	—	—	—	—	36	—	—	—	—	36
	8A	24	24	36	36	36	24	24	36	36	36
	8B	—	—	48	—	48	—	—	48	—	48
	8C	48	—	48	48	48	48	—	48	48	48
	8D	24	36	48	48	48	24	36	48	48	48
	8E	—	—	—	—	36	—	—	—	—	36
Total		552	432	696	600	840	552	432	696	600	840

Table 5-7: Modular I/O Banks for Stratix V GX A9, AB, B5, B6, B9, and BB Devices

Member Code		A9		AB		B5		B6		B9	BB
Package		KH40	NF45	KH40	NF45	RF40	RF43	RF40	RF43	RH43	RH43
Bank	3A	36	36	36	36	36	36	36	36	36	36
	3B	48	48	48	48	48	48	48	48	48	48
	3C	48	48	48	48	—	48	—	48	48	48
	3D	48	48	48	48	—	36	—	36	36	36
	3E	—	36	—	36	—	—	—	—	—	—
	4A	24	24	24	24	48	48	48	48	48	48
	4B	48	48	48	48	48	48	48	48	48	48
	4C	48	48	48	48	36	36	36	36	36	36
	4D	48	48	48	48	—	—	—	—	—	—
	4E	—	36	—	36	—	—	—	—	—	—
	7A	24	24	24	24	48	48	48	48	48	48
	7B	48	48	48	48	48	48	48	48	48	48
	7C	48	48	48	48	36	36	36	36	36	36
	7D	48	48	48	48	—	—	—	—	—	—
	7E	—	36	—	36	—	—	—	—	—	—
	8A	36	36	36	36	36	36	36	36	36	36
	8B	48	48	48	48	48	48	48	48	48	48
	8C	48	48	48	48	—	48	—	48	48	48
	8D	48	48	48	48	—	36	—	36	36	36
	8E	—	36	—	36	—	—	—	—	—	—
Total		696	840	696	840	432	600	432	600	600	600

Related Information

- [I/O Banks in Stratix V Devices](#) on page 5-9
- [Guideline: Use the Same VCCPD for All I/O Banks in a Group](#) on page 5-7
Provides guidelines about V_{CCPD} and I/O banks groups.

Modular I/O Banks for Stratix V GS Devices

Table 5-8: Modular I/O Banks for Stratix V GS Devices

Member Code	D3		D4			D5		D6		D8	
Package	EF29	HF35	EF29	HF35	KF40	HF35	KF40	KF40	NF45	KF40	NF45
Bank	3A	36	36	36	36	36	36	36	36	36	36
	3B	48	48	48	48	48	48	48	48	48	48
	3C	—	—	—	—	48	48	48	48	48	48
	3D	24	24	24	24	48	24	48	48	48	48
	3E	—	—	—	—	—	—	—	36	—	36
	4A	24	24	24	24	24	24	24	24	24	24
	4B	—	48	—	48	48	48	48	48	48	48
	4C	—	—	—	—	48	48	48	48	48	48
	4D	24	24	24	24	48	24	48	48	48	48
	4E	—	—	—	—	—	—	—	36	—	36
	7A	24	24	24	24	24	24	24	24	24	24
	7B	—	24	—	24	48	48	48	48	48	48
	7C	48	48	48	48	48	48	48	48	48	48
	7D	36	36	36	36	48	36	48	48	48	48
	7E	—	—	—	—	—	—	—	36	—	36
	8A	24	24	24	24	36	24	36	36	36	36
	8B	—	—	—	—	48	—	48	48	48	48
	8C	48	48	48	48	48	48	48	48	48	48
	8D	24	24	24	24	48	24	48	48	48	48
	8E	—	—	—	—	—	—	—	36	—	36
Total	360	432	360	432	696	552	696	696	840	696	840

Related Information

- [I/O Banks in Stratix V Devices](#) on page 5-9
- [Guideline: Use the Same VCCPD for All I/O Banks in a Group](#) on page 5-7
Provides guidelines about V_{CCPD} and I/O banks groups.

Modular I/O Banks for Stratix V GT Devices

Table 5-9: Modular I/O Banks for Stratix V GT Devices

Member Code		C5	C7
Package		KF40	KF40
Bank	3A	36	36
	3B	48	48
	3C	48	48
	3D	24	24
	3E	—	—
	4A	24	24
	4B	48	48
	4C	48	48
	4D	24	24
	4E	—	—
	7A	24	24
	7B	48	48
	7C	48	48
	7D	48	48
	7E	—	—
	8A	36	36
	8B	—	—
	8C	48	48
	8D	48	48
	8E	—	—
Total		600	600

Related Information

- [I/O Banks in Stratix V Devices](#) on page 5-9
- [Guideline: Use the Same VCCPD for All I/O Banks in a Group](#) on page 5-7
Provides guidelines about V_{CCPD} and I/O banks groups.

I/O Element Structure in Stratix V Devices

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 single data rate (SDR) or double data rate (DDR) transfer.

The IOEs are located in I/O blocks around the periphery of the Stratix V device.

I/O Buffer and Registers in Stratix V Devices

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.

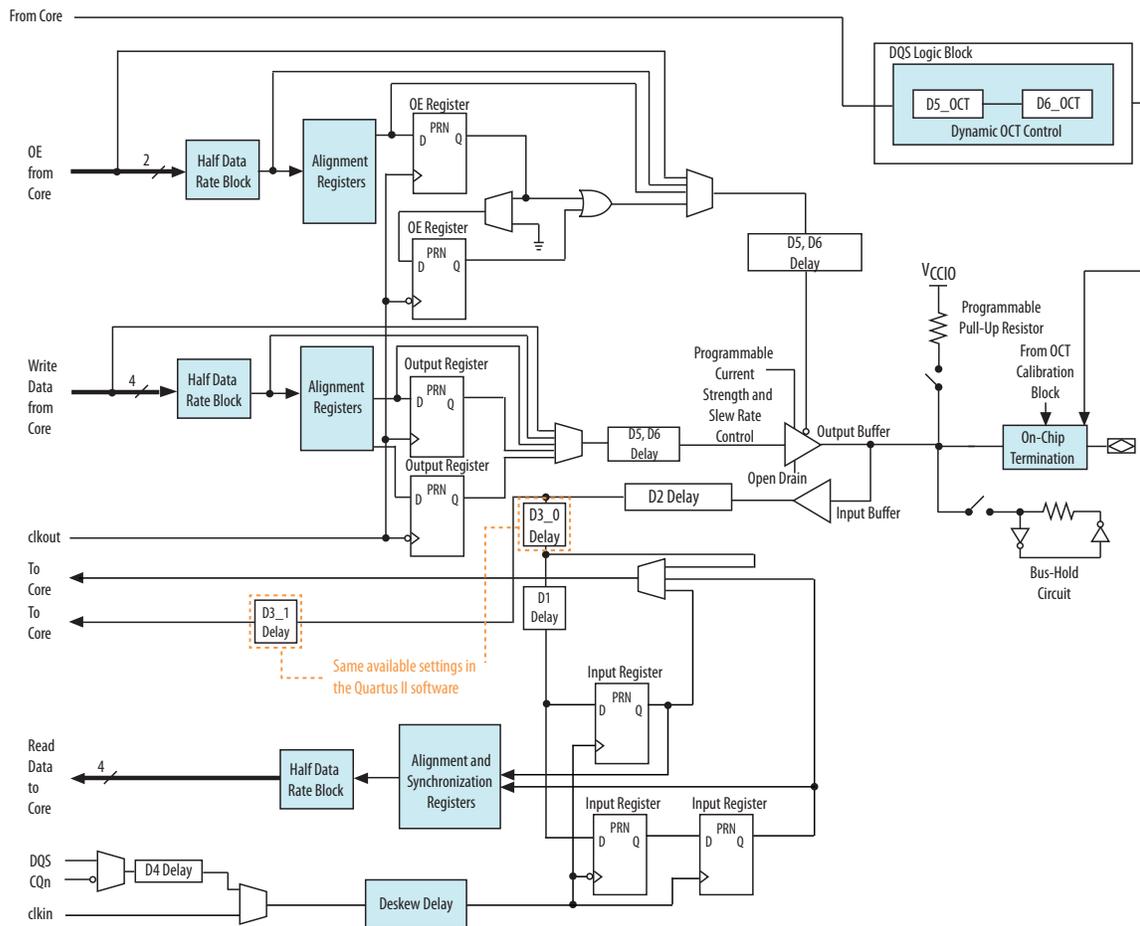
Table 5-10: Input and Output Paths in Stratix V Devices

This table summarizes the input and output path in the Stratix V devices.

Input Path	Output Path
Consists of: <ul style="list-style-type: none"> • DDR input registers • Alignment and synchronization registers • Half data rate blocks 	Consists of: <ul style="list-style-type: none"> • Output or OE registers • Alignment registers • 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.	You can bypass each block of the output and OE paths.

Figure 5-2: IOE Structure for Stratix V Devices

This figure shows the Stratix V FPGA IOE structure. In the figure, one dynamic on-chip termination (OCT) control is available for each DQ/DQS group.



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 I/O standards such as SSTL-12, SSTL-15, SSTL-125, SSTL-135, and HSUL-12.

High-Speed Differential I/O with DPA Support

To support high-speed differential I/O, Stratix V devices contain the following dedicated circuitries:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment
- DPA
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs)

Programmable IOE Features in Stratix V Devices

Table 5-11: Summary of Supported Stratix V Programmable IOE Features and Settings

Feature	Setting	Assignment Name	Supported I/O Standards
Slew Rate Control ⁽⁷⁾	<ul style="list-style-type: none"> 0 (Slow) 1 (Fast). Default is 1. 	Slew Rate	<ul style="list-style-type: none"> 3.3V LVTTTL 1.2/1.5/1.8/2.5/3.3 LVCMOS SSTL-2/SSTL-18/SSTL-15/SSTL-12 1.8/1.5/1.2V HSTL Differential SSTL-2/Differential SSTL-18/Differential SSTL-15/Differential SSTL-12 Differential 1.2/1.5/ 1.8V HSTL
Programmable Output Buffer Delay	<ul style="list-style-type: none"> 0 ps (Default) 50 ps 100 ps 150 ps 	Output Buffer Delay	<ul style="list-style-type: none"> 3.3V LVTTTL 1.2/1.5/1.8/2.5/3.3 LVCMOS SSTL-2/SSTL-18/SSTL-15/SSTL-135/SSTL-125/SSTL-12 1.8/1.5/1.2V HSTL HSUL-12 Differential SSTL-2/Differential SSTL-18/Differential SSTL-15/Differential SSTL-135/Differential SSTL-125/Differential SSTL-12 Differential 1.2/1.5/ 1.8V HSTL Differential HSUL-12
Open-Drain Output ⁽⁸⁾	<ul style="list-style-type: none"> On Off (default) 	—	<ul style="list-style-type: none"> 3.3V LVTTTL 1.2/1.5/1.8/2.5/3.3 LVCMOS SSTL-2/SSTL-12/SSTL-18/SSTL-15/SSTL-135/SSTL-125 1.8/1.5/1.2V HSTL HSUL-12

⁽⁷⁾ Disabled if you use the R_S OCT feature.

⁽⁸⁾ Open drain feature can be enabled using OPNDRN primitive.

Feature	Setting	Assignment Name	Supported I/O Standards
Bus-Hold ⁽⁹⁾	<ul style="list-style-type: none"> On Off (default) 	Enable Bus-Hold Circuitry	<ul style="list-style-type: none"> 3.3V LVTTTL 1.2/1.5/1.8/2.5/3.3 LVCMOS SSTL-2/SSTL-12/SSTL-18/SSTL-15/SSTL-135/SSTL-125 1.8/1.5/1.2V HSTL HSUL-12
Pull-up Resistor ⁽¹⁰⁾	<ul style="list-style-type: none"> On Off (default) 	Weak Pull-Up Resistor	<ul style="list-style-type: none"> 3.3V LVTTTL 1.2/1.5/1.8/2.5/3.3 LVCMOS SSTL-2/SSTL-12/SSTL-18/SSTL-15/SSTL-135/SSTL-125 1.8/1.5/1.2V HSTL HSUL-12
Pre-Emphasis	<ul style="list-style-type: none"> 0 (disabled) 1 (enabled). Default is 1. 	Programmable Pre-emphasis	<ul style="list-style-type: none"> LVDS RSDS Mini-LVDS
Differential Output Voltage	<ul style="list-style-type: none"> 0 (low) 1 (medium low) 2 (medium high) 3 (high). Default is 1 	Programmable Differential Output Voltage (VOD)	<ul style="list-style-type: none"> LVDS RSDS Mini-LVDS

Related Information

- [Stratix V Device Datasheet](#)
- [Stratix V Device Datasheet](#)
- [Programmable Current Strength](#) on page 5-19
- [Programmable Output Slew Rate Control](#) on page 5-20
- [Programmable IOE Delay](#) on page 5-21
- [Programmable Output Buffer Delay](#) on page 5-21
- [Programmable Pre-Emphasis](#) on page 5-22
- [Programmable Differential Output Voltage](#) on page 5-23
- [Stratix V Device Datasheet](#)

Programmable Current Strength

You can use the programmable current strength to mitigate the effects of high signal attenuation that is caused by a long transmission line or a legacy backplane.

⁽⁹⁾ Disabled if you use the pull-up resistor feature.

⁽¹⁰⁾ Disabled if you use the bus-hold feature.

Table 5-12: Programmable Current Strength Settings for Stratix V Devices

The output buffer for each Stratix V device I/O pin has a programmable current strength control for the I/O standards listed in this table.

I/O Standard	I_{OH} / I_{OL} Current Strength Setting (mA) (Default setting in bold)
3.3-V LVTTTL	16, 12 , 8, 4
3.3-V LVCMOS	16, 12 , 8, 4
2.5-V LVCMOS	16, 12 , 8, 4
1.8-V LVCMOS	12 , 10, 8, 6, 4, 2
1.5-V LVCMOS	12 , 10, 8, 6, 4, 2
1.2-V LVCMOS	8 , 6, 4, 2
SSTL-2 Class I	12, 10, 8
SSTL-2 Class II	16
SSTL-18 Class I	12, 10, 8 , 6, 4
SSTL-18 Class II	16 , 8
SSTL-15 Class I	12, 10, 8 , 6, 4
SSTL-15 Class II	16 , 8
1.8-V HSTL Class I	12, 10, 8 , 6, 4
1.8-V HSTL Class II	16
1.5-V HSTL Class I	12, 10, 8 , 6, 4
1.5-V HSTL Class II	16
1.2-V HSTL Class I	12, 10, 8 , 6, 4
1.2-V HSTL Class II	16

The 3.3 V LVTTTL and 3.3 V LVCMOS I/O standards are supported using V_{CCIO} and V_{CCPD} at 3.0 V.

Note: Intel recommends that you perform IBIS or SPICE simulations to determine the best current strength setting for your specific application.

Related Information

[Programmable IOE Features in Stratix V Devices](#) on page 5-18

Programmable Output Slew Rate Control

Programmable output slew rate is available for single-ended I/O standards and emulated LVDS output standards.

The programmable output slew rate control in the output buffer of each regular- and dual-function I/O pin allows you to configure the following:

- Fast slew rate—provides high-speed transitions for high-performance systems. Fast slew rates improve the available timing margin in memory-interface applications or when the output pin has high-capacitive loading.
- Slow slew rate—reduces system noise and crosstalk but adds a nominal delay to the rising and falling edges.

You can specify the slew rate on a pin-by-pin basis because each I/O pin contains a slew rate control.

Note: Altera recommends that you perform IBIS or SPICE simulations to determine the best slew rate setting for your specific application.

Related Information

[Programmable IOE Features in Stratix V Devices](#) on page 5-18

Programmable IOE Delay

You can activate the programmable IOE delays to ensure zero hold times, minimize setup times, or increase clock-to-output times. This feature helps read and write timing margins because it minimizes the uncertainties between signals in the bus.

Each single-ended and differential I/O 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 signals within a bus have the same delay going into or out of the device.

For more information about the programmable IOE delay specifications, refer to the device datasheet.

Related Information

- [Stratix V Device Datasheet](#)
- [Stratix V Device Datasheet](#)
- [Programmable IOE Features in Stratix V Devices](#) on page 5-18
- [Stratix V Device Datasheet](#)

Programmable Output Buffer Delay

The delay chains are built inside the single-ended output buffer. There are four levels of output buffer delay settings. By default, there is no delay.

The delay chains can independently control the rising and falling edge delays of the output buffer, allowing you 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
- Improve high-speed memory-interface timing margins

For more information about the programmable output buffer delay specifications, refer to the device datasheet.

Related Information

- [Stratix V Device Datasheet](#)
- [Stratix V Device Datasheet](#)
- [Programmable IOE Features in Stratix V Devices](#) on page 5-18
- [Stratix V Device Datasheet](#)

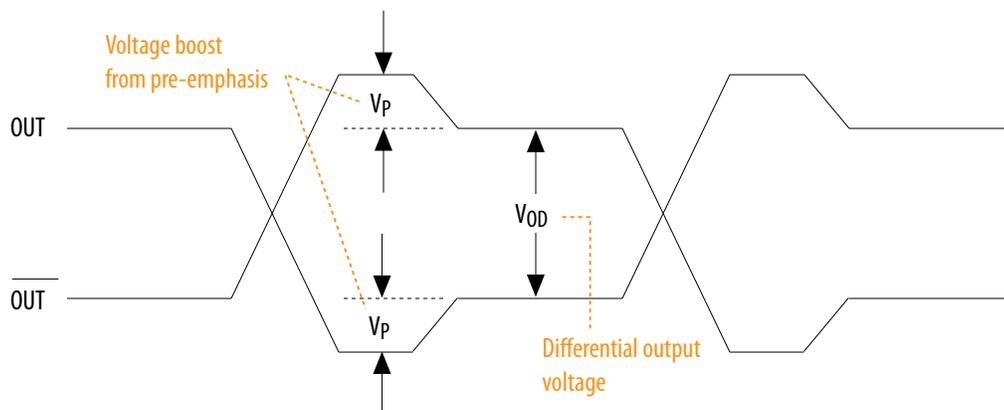
Programmable Pre-Emphasis

The V_{OD} setting and the output impedance of the driver set the output current limit of a high-speed transmission signal. At a high frequency, the slew rate may not be fast enough to reach the full V_{OD} level 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.

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. The overshoot introduced by the extra current happens only during a change of state switching to increase the output slew rate and does not ring, unlike the overshoot caused by signal reflection. The amount of pre-emphasis required depends on the attenuation of the high-frequency component along the transmission line.

Figure 5-3: Programmable Pre-Emphasis

This figure shows the LVDS output with pre-emphasis.

**Table 5-13: Intel Quartus Prime Software Assignment Editor—Programmable Pre-Emphasis**

This table lists the assignment name for programmable pre-emphasis and its possible values in the Intel Quartus Prime software Assignment Editor.

Field	Assignment
To	tx_out
Assignment name	Programmable Pre-emphasis
Allowed values	0 (disabled), 1 (enabled). Default is 1.

Related Information

[Programmable IOE Features in Stratix V Devices](#) on page 5-18

Programmable Differential Output Voltage

The programmable V_{OD} settings allow you to adjust the output eye opening to optimize the trace length and power consumption. A higher V_{OD} swing improves voltage margins at the receiver end, and a smaller V_{OD} swing reduces power consumption. You can statically adjust the V_{OD} of the differential signal by changing the V_{OD} settings in the Intel Quartus Prime software Assignment Editor.

Figure 5-4: Differential V_{OD}

This figure shows the V_{OD} of the differential LVDS output.

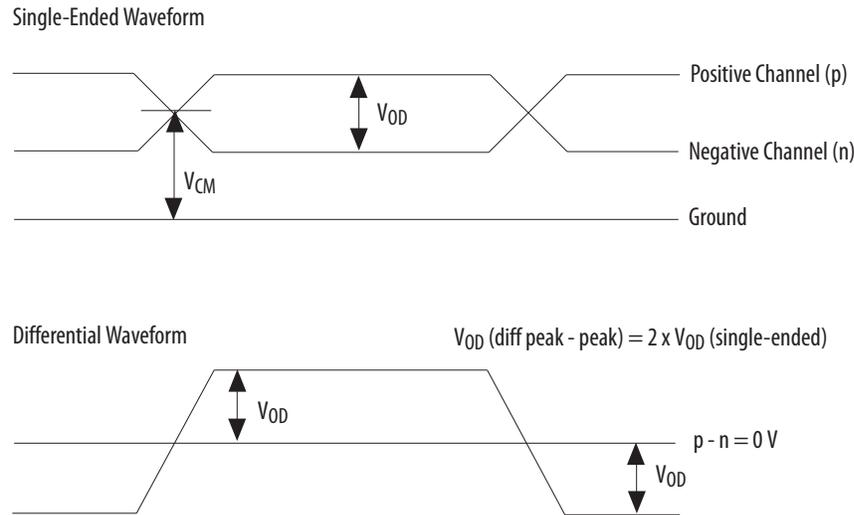


Table 5-14: Intel Quartus Prime Software Assignment Editor—Programmable V_{OD}

This table lists the assignment name for programmable V_{OD} and its possible values in the Intel Quartus Prime software Assignment Editor.

Field	Assignment
To	tx_out
Assignment name	Programmable Differential Output Voltage (V_{OD})
Allowed values	0 (low), 1 (medium low), 2 (medium high), 3 (high). Default is 1.

Related Information

[Programmable IOE Features in Stratix V Devices](#) on page 5-18

Open-Drain Output

The optional open-drain output for each I/O pin is equivalent to an open collector output. If it is configured as an open drain, the logic value of the output is either high-Z or logic low.

You can attach several open-drain output to a wire. This connection type is like a logical OR function and is commonly called an active-low wired-OR circuit. If at least one of the outputs is in logic 0 state (active), the circuit sinks the current and brings the line to low voltage.

You can use open-drain output if you are connecting multiple devices to a bus. For example, you can use the open-drain output for system-level control signals that can be asserted by any device or as an interrupt.

You can enable the open-drain output assignment using one these methods:

- Design the tristate buffer using OPNDRN primitive.
- Turn on the **Auto Open-Drain Pins** option in the Intel Quartus Prime software.

Although you can design open-drain output without enabling the option assignment, you will not be using the open-drain output feature of the I/O buffer. The open-drain output feature in the I/O buffer provides you the best propagation delay from OE to output.

Bus-Hold Circuitry

Each I/O pin provides an optional bus-hold feature that is active only after configuration. When the device enters user mode, the bus-hold circuit captures the value that is present on the pin by the end of the configuration.

The bus-hold circuitry uses a resistor with a nominal resistance (R_{BH}), approximately 7 k Ω , to weakly pull the signal level to the last-driven state of the pin. The bus-hold circuitry holds this pin state until the next input signal is present. Because of this, you do not require an external pull-up or pull-down resistor to hold a signal level when the bus is tri-stated.

For each I/O pin, you can individually specify that the bus-hold circuitry pulls non-driven pins away from the input threshold voltage—where noise can cause unintended high-frequency switching. To prevent over-driving signals, the bus-hold circuitry drives the voltage level of the I/O pin lower than the V_{CCIO} level.

If you enable the bus-hold feature, you cannot use the programmable pull-up option. To configure the I/O pin for differential signals, disable the bus-hold feature.

Pull-up Resistor

Each I/O pin provides an optional programmable pull-up resistor during user mode. The pull-up resistor, typically 25 k Ω , weakly holds the I/O to the V_{CCIO} level. If you enable this option, you cannot use the bus-hold feature.

The Stratix V device supports programmable pull-up resistors only on user I/O pins.

For dedicated configuration pins or JTAG pins with internal pull-up resistors, these resistor values are not programmable. You can find more information related to the internal pull-up values for dedicated configuration pins or JTAG pins in the Stratix V Pin Connection Guidelines.

On-Chip I/O Termination in Stratix V Devices

Dynamic R_S and R_T OCT provides I/O impedance matching and termination capabilities. OCT maintains signal quality, saves board space, and reduces external component costs.

The Stratix V devices support OCT in all I/O banks.

Table 5-15: OCT Schemes Supported in Stratix V Devices

Direction	OCT Schemes
Output	R _S OCT with calibration
	R _S OCT without calibration
Input	R _T OCT with calibration
	R _D OCT (differential LVDS I/O standard only)
Bidirectional	Dynamic R _S OCT and R _T OCT

R_S OCT without Calibration in Stratix V Devices

The Stratix V devices support R_S OCT for single-ended I/O standards. R_S OCT without calibration is supported on output only.

Table 5-16: Selectable I/O Standards for R_S OCT Without Calibration

This table lists the output termination settings for uncalibrated OCT on different I/O standards.

I/O Standard	Uncalibrated OCT (Output)
	R _S (Ω)
3.3 V LVTTTL/3.3 V LVCMOS	25/50
2.5 V LVCMOS	25/50
1.8 V LVCMOS	25/50
1.5 V LVCMOS	25/50
1.2 V LVCMOS	25/50
SSTL-2 Class I	50
SSTL-2 Class II	25
SSTL-18 Class I	50
SSTL-18 Class II	25
SSTL-15 Class I	50
SSTL-15 Class II	25
1.8 V HSTL Class I	50
1.8 V HSTL Class II	25
1.5 V HSTL Class I	50
1.5 V HSTL Class II	25
1.2 V HSTL Class I	50
1.2 V HSTL Class II	25
Differential SSTL-2 Class I	50
Differential SSTL-2 Class II	25

I/O Standard	Uncalibrated OCT (Output)
	R _S (Ω)
Differential SSTL-18 Class I	50
Differential SSTL-18 Class II	25
Differential SSTL-15 Class I	50
Differential SSTL-15 Class II	25
Differential 1.8 V HSTL Class I	50
Differential 1.8 V HSTL Class II	25
Differential 1.5 V HSTL Class I	50
Differential 1.5 V HSTL Class II	25
Differential 1.2 V HSTL Class I	50
Differential 1.2 V HSTL Class II	25
SSTL-15	25, 34, 40, 50
SSTL-135	34, 40
SSTL-125	34, 40
SSTL-12	40, 60, 240
HSUL-12	34.3, 40, 48, 60, 80

The following list specifies the default settings for R_S OCT without calibration in the Intel Quartus Prime software:

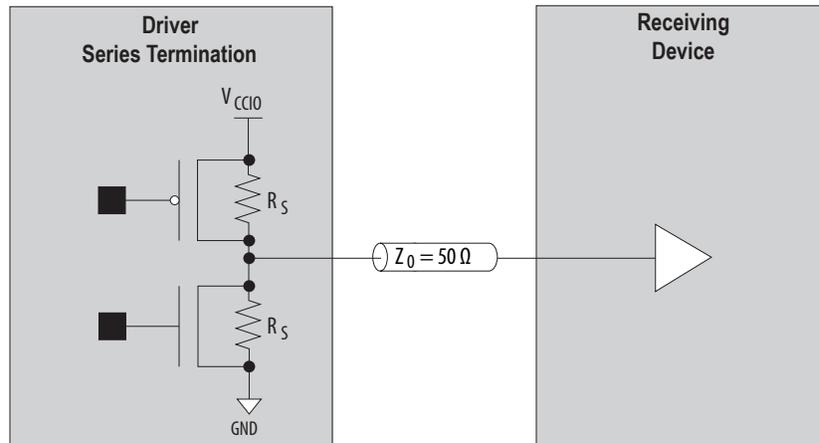
- For all non-voltage-referenced, HSTL Class I, and SSTL Class I I/O standards—50 Ω.
- For HSTL Class II and SSTL Class II I/O standards—25 Ω.

Driver-impedance matching provides the I/O driver with controlled output impedance that closely matches the impedance of the transmission line. As a result, you can significantly reduce signal reflections on PCB traces.

If you select matching impedance, current strength is no longer selectable.

Figure 5-5: R_S OCT Without Calibration

This figure shows the R_S as the intrinsic impedance of the output transistors. Typical R_S values are 25 Ω and 50 Ω.



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 in Stratix V Devices

The Stratix V devices support R_S OCT with calibration in all banks.

Table 5-17: Selectable I/O Standards for R_S OCT With Calibration

This table lists the output termination settings for calibrated OCT on different I/O standards.

I/O Standard	Calibrated OCT (Output)	
	R _S (Ω)	RZQ (Ω)
3.3 V LVTTTL/3.3 V LVCMOS	25/50	100
2.5 V LVCMOS	25/50	100
1.8 V LVCMOS	25/50	100
1.5 V LVCMOS	25/50	100
1.2 V LVCMOS	25/50	100
SSTL-2 Class I	50	100
SSTL-2 Class II	25	100
SSTL-18 Class I	50	100
SSTL-18 Class II	25	100
SSTL-15 Class I	50	100
SSTL-15 Class II	25	100

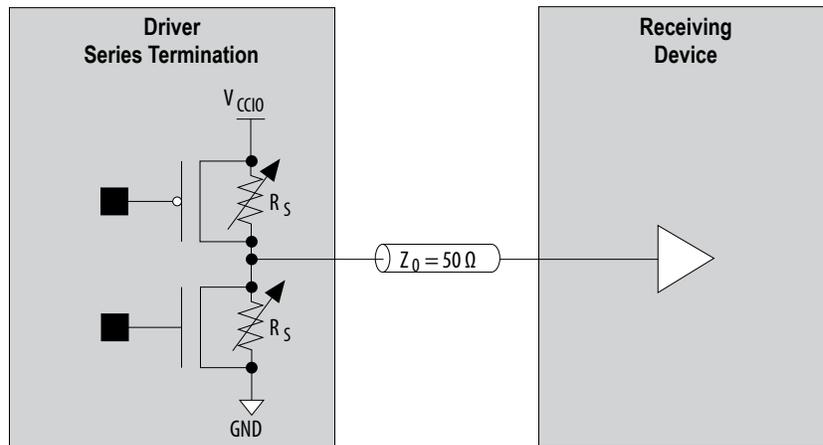
I/O Standard	Calibrated OCT (Output)	
	R _S (Ω)	RZQ (Ω)
1.8 V HSTL Class I	50	100
1.8 V HSTL Class II	25	100
1.5 V HSTL Class I	50	100
1.5 V HSTL Class II	25	100
1.2 V HSTL Class I	50	100
1.2 V HSTL Class II	25	100
Differential SSTL-2 Class I	50	100
Differential SSTL-2 Class II	25	100
Differential SSTL-18 Class I	50	100
Differential SSTL-18 Class II	25	100
Differential SSTL-15 Class I	50	100
Differential SSTL-15 Class II	25	100
Differential 1.8 V HSTL Class I	50	100
Differential 1.8 V HSTL Class II	25	100
Differential 1.5 V HSTL Class I	50	100
Differential 1.5 V HSTL Class II	25	100
Differential 1.2 V HSTL Class I	50	100
Differential 1.2 V HSTL Class II	25	100
SSTL-15	25, 50	100
	34, 40	240
SSTL-135	34, 40	240
SSTL-125	34, 40	240
SSTL-12	40, 60, 240	240
HSUL-12	34, 40, 48, 60, 80	240
Differential SSTL-15	25, 50	100
	34, 40	240
Differential SSTL-135	34, 40	240
Differential SSTL-125	34, 40	240
Differential SSTL-12	40, 60, 240	240
Differential HSUL-12	34, 40, 48, 60, 80	240

The R_S OCT calibration circuit compares the total impedance of the I/O buffer to the external reference resistor 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, the circuit powers down and stops changing the characteristics of the drivers.

Figure 5-6: R_S OCT with Calibration

This figure shows the R_S as the intrinsic impedance of the output transistors.



R_T OCT with Calibration in Stratix V Devices

The Stratix V devices support R_T OCT with calibration in all banks. R_T OCT with calibration is available only for configuration of input and bidirectional pins. Output pin configurations do not support R_T OCT with calibration. If you use R_T OCT, the V_{CCI0} of the bank must match the I/O standard of the pin where you enable the R_T OCT.

Table 5-18: Selectable I/O Standards for R_T OCT With Calibration

This table lists the input termination settings for calibrated OCT on different I/O standards.

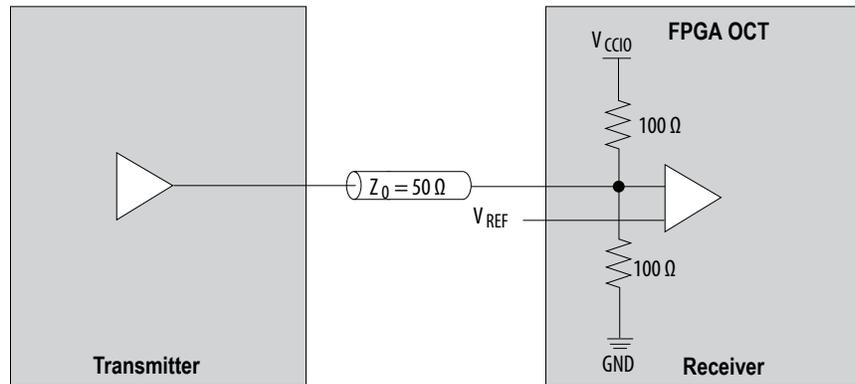
I/O Standard	Calibrated OCT (Input)	
	R _T (Ω)	RZQ (Ω)
SSTL-2 Class I	50	100
SSTL-2 Class II	50	100
SSTL-18 Class I	50	100
SSTL-18 Class II	50	100
SSTL-15 Class I	50	100
SSTL-15 Class II	50	100
1.8 V HSTL Class I	50	100
1.8 V HSTL Class II	50	100
1.5 V HSTL Class I	50	100
1.5 V HSTL Class II	50	100
1.2 V HSTL Class I	50	100
1.2 V HSTL Class II	50	100

I/O Standard	Calibrated OCT (Input)	
	R _T (Ω)	RZQ (Ω)
Differential SSTL-2 Class I	50	100
Differential SSTL-2 Class II	50	100
Differential SSTL-18 Class I	50	100
Differential SSTL-18 Class II	50	100
Differential SSTL-15 Class I	50	100
Differential SSTL-15 Class II	50	100
Differential 1.8 V HSTL Class I	50	100
Differential 1.8 V HSTL Class II	50	100
Differential 1.5 V HSTL Class I	50	100
Differential 1.5 V HSTL Class II	50	100
Differential 1.2 V HSTL Class I	50	100
Differential 1.2 V HSTL Class II	50	100
SSTL-15	20, 30, 40, 60,120	240
SSTL-135	20, 30, 40, 60, 120	240
SSTL-125	20, 30, 40, 60, 120	240
SSTL-12	60, 120	240
HSUL-12	34, 40, 48, 60, 80	240
Differential SSTL-15	20, 30, 40, 60,120	240
Differential SSTL-135	20, 30, 40, 60, 120	240
Differential SSTL-125	20, 30, 40, 60, 120	240
Differential SSTL-12	60, 120	240
Differential HSUL-12	34, 40, 48, 60, 80	240

The R_T OCT calibration circuit compares the total impedance of the I/O buffer to the external resistor connected to the RZQ pin. The circuit dynamically enables or disables the transistors until the total impedance of the I/O buffer matches the external resistor.

Calibration occurs at the end of the device configuration. When the calibration circuit finds the correct impedance, the circuit powers down and stops changing the characteristics of the drivers.

Figure 5-7: R_T OCT with Calibration



Dynamic OCT in Stratix V Devices

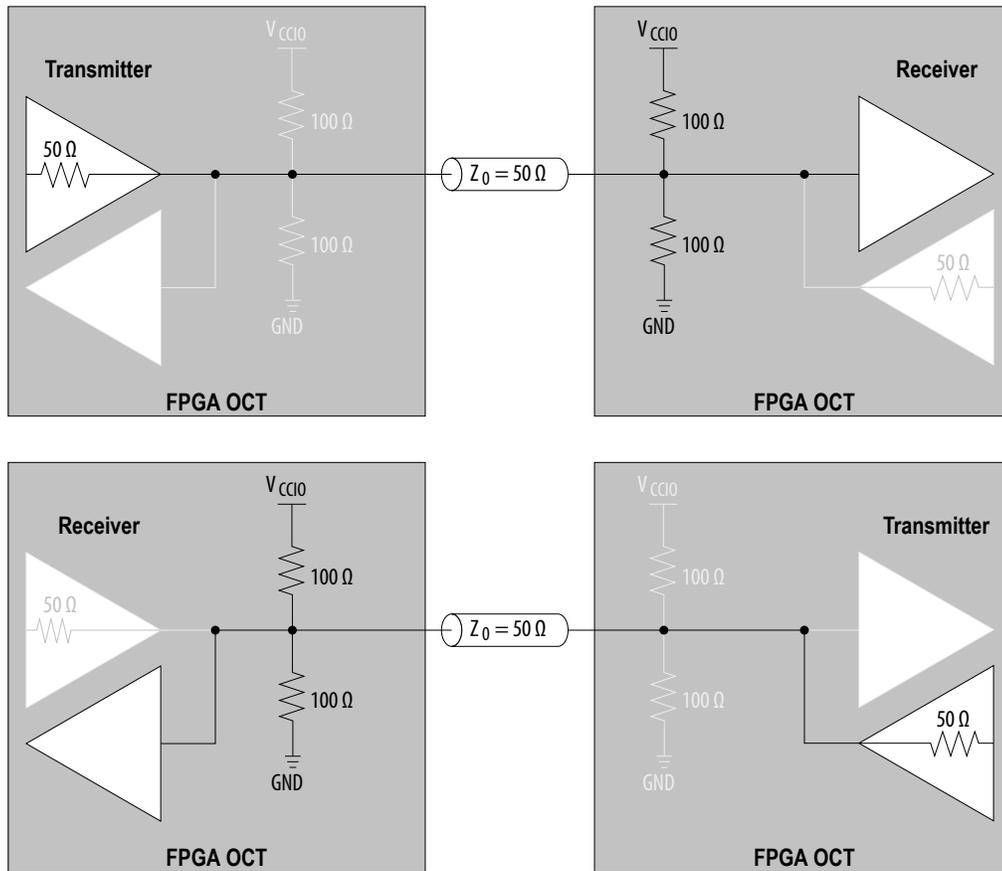
Dynamic OCT is useful for terminating a high-performance bidirectional path by optimizing the signal integrity depending on the direction of the data. Dynamic OCT also helps save power because device termination is internal—termination switches on only during input operation and thus draw less static power.

Note: If you use the HSUL-12, SSTL-12, SSTL-15, SSTL-135, and SSTL-125 I/O standards with external memory interfaces, Intel recommends that you use OCT with these I/O standards to save board space and cost. OCT reduces the number of external termination resistors used.

Table 5-19: Dynamic OCT Based on Bidirectional I/O

Dynamic R_T OCT or R_S OCT is enabled or disabled based on whether the bidirectional I/O acts as a receiver or driver.

Dynamic OCT	Bidirectional I/O	State
Dynamic R_T OCT	Acts as a receiver	Enabled
	Acts as a driver	Disabled
Dynamic R_S OCT	Acts as a receiver	Disabled
	Acts as a driver	Enabled

Figure 5-8: Dynamic R_T OCT in Stratix V Devices

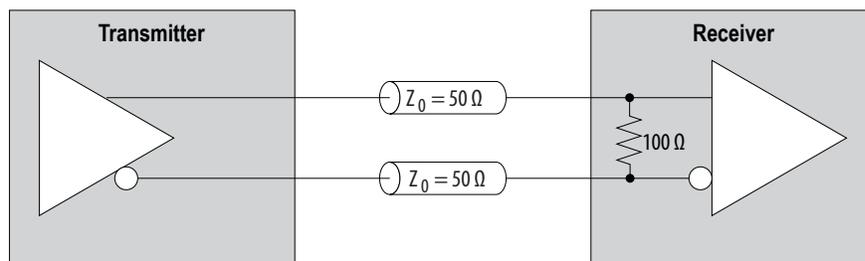
LVDS Input R_D OCT in Stratix V Devices

The Stratix V devices support R_D OCT in all I/O banks.

You can only use R_D OCT if you set the V_{CCPD} to 2.5 V.

Figure 5-9: Differential Input OCT

The Stratix V devices support OCT for differential LVDS input buffers with a nominal resistance value of 100 Ω , as shown in this figure.



OCT Calibration Block in Stratix V Devices

You can calibrate the OCT using any of the available four to eight OCT calibration blocks, depending on the device density. Each calibration block contains one RZQ pin.

You can use R_S and R_T OCT in the same I/O bank for different I/O standards if the I/O standards use the same V_{CCIO} supply voltage. You cannot configure the R_S OCT and the programmable current strength for the same I/O buffer.

The 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:

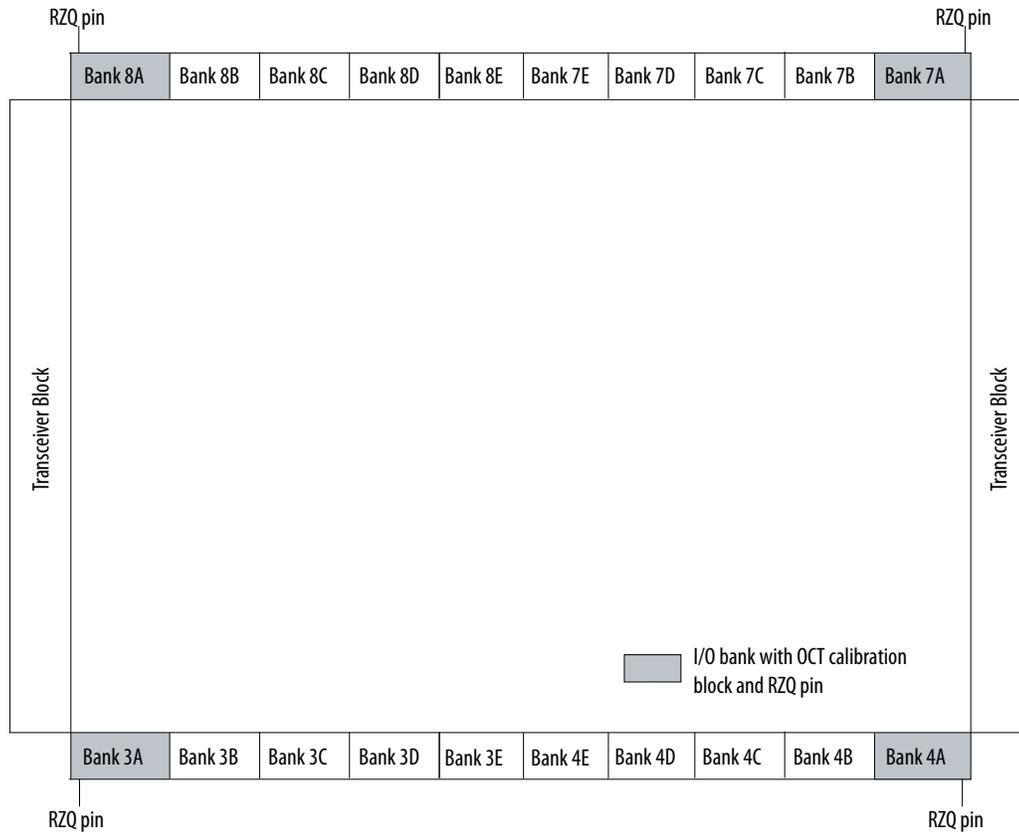
- Connect the RZQ pin to GND through an external 100 Ω or 240 Ω resistor (depending on the R_S or R_T OCT value).
- The RZQ pin shares the same V_{CCIO} supply voltage with the I/O bank where the pin is located.
- The RZQ pin is a dual-purpose I/O pin and functions as a general purpose I/O pin if you do not use the calibration circuit.

Stratix V devices support calibrated R_S and calibrated R_T OCT on all I/O pins except for dedicated configuration pins.

Calibration Block Locations in Stratix V Devices

Figure 5-10: OCT Calibration Block and RZQ Pin Location

This figure shows the location of I/O banks with OCT calibration blocks and RZQ pins in the Stratix V device. This figure represents the top view of the silicon die that corresponds to a reverse view of the device package and illustrates the highest density device in the device family.



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. 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.

I/O banks that do not have calibration blocks share the calibration blocks in the I/O banks that have calibration blocks.

All I/O banks support OCT calibration 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} . 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.

Related Information

- [OCT Calibration Block Sharing Example](#) on page 5-35

- [ALTOCT IP Core User Guide](#)
Provides more information about the OCT calibration block.

OCT Calibration Block Sharing Example

Figure 5-11: Example of Calibrating Multiple I/O Banks with One Shared OCT Calibration Block

As an example, this figure shows a group of I/O banks that has the same V_{CCIO} voltage. The figure does not show transceiver calibration blocks. This figure represents the top view of the silicon die that corresponds to a reverse view of the device package and illustrates the highest density device in the device family.



Because banks 3B, 4C, and 7B have the same V_{CCIO} as bank 7A, you can calibrate all four I/O banks (3B, 4C, 7A, and 7B) with the OCT calibration block (CB7) located in bank 7A.

To enable this calibration, serially shift out the R_S OCT calibration codes from the OCT calibration block in bank 7A to the I/O banks around the periphery.

Related Information

- [Sharing an OCT Calibration Block on Multiple I/O Banks](#) on page 5-34
- [ALTOCT IP Core User Guide](#)
Provides more information about the OCT calibration block.

OCT Calibration in 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.

OCT Calibration in 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.

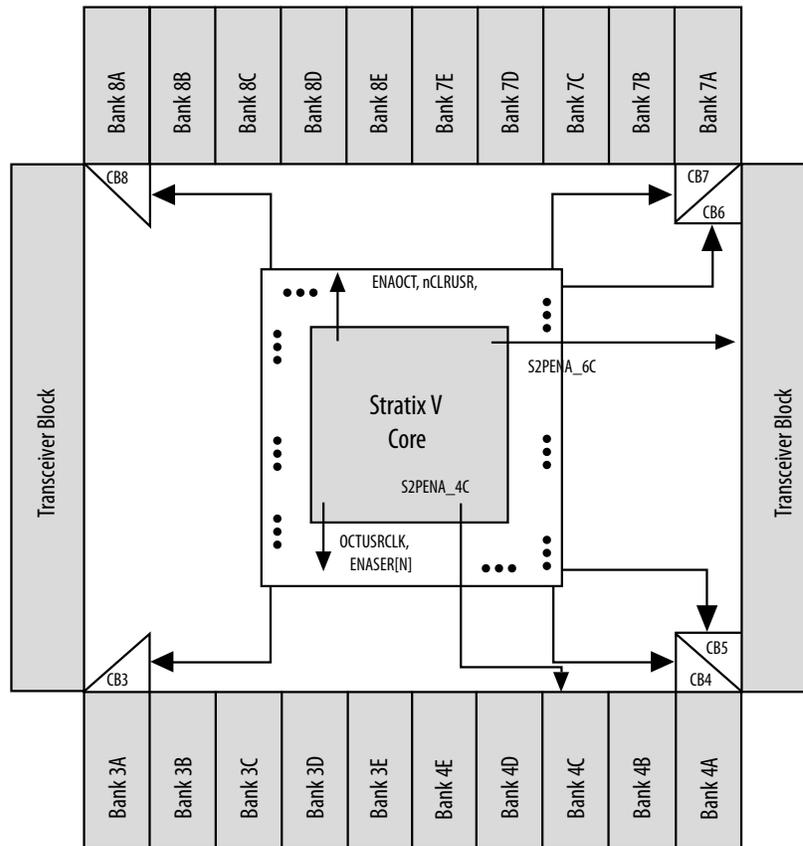
Table 5-20: OCT Calibration Block Ports for User Control

This table lists the user-controlled calibration block signal names and their descriptions

Signal Name	Description
<code>OCTUSRCLK</code>	Clock for OCT block.
<code>ENAOCT</code>	Enable OCT Calibration (generated by user IP).
<code>ENASER[7..0]</code>	<ul style="list-style-type: none"> <code>ENAOCT</code> is 0—each signal enables the OCT serializer for the corresponding OCT calibration block. <code>ENAOCT</code> is 1—each signal enables OCT calibration for the corresponding OCT calibration block.
<code>S2PENA_bank#</code>	Serial-to-parallel load enable per I/O bank.
<code>nCLRUSR</code>	Clear user.

Figure 5-12: Signals Used for User Mode Calibration

This figure shows the flow of the user signal.

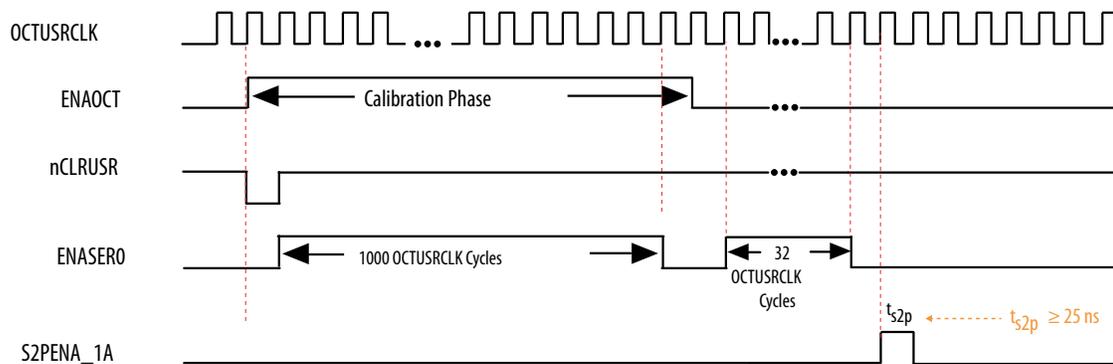


When **ENAOCR** is 1, all OCT calibration blocks are in calibration mode. When **ENAOCR** is 0, all OCT calibration blocks are in serial data transfer mode. The **OCTUSRCLK** clock frequency must be 20 MHz or less.

Note: You must generate all user signals on the rising edge of the **OCTUSRCLK** signal.

Figure 5-13: OCT User Mode Signal—Timing Waveform for One OCT Block

This figure shows the user mode signal-timing waveforms.



OCT Calibration

To calibrate OCT block N (where N is a calibration block number), you must assert $ENAOCT$ one cycle before asserting $ENASERN$. You must also set $nCLRUSR$ low for one $OCTUSRCLK$ cycle before the $ENASERN$ signal is asserted. Assert the $ENASERN$ signals for 1,000 $OCTUSRCLK$ 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

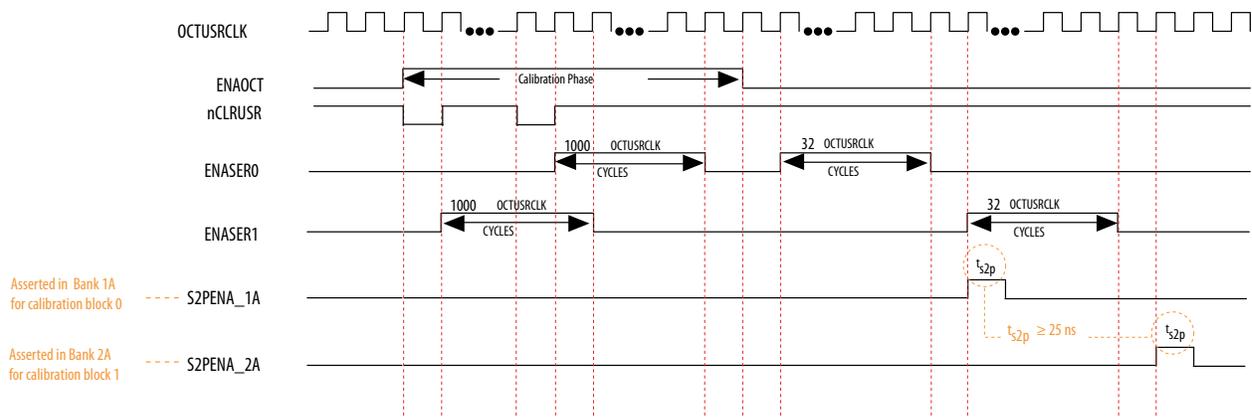
After you complete calibration, you must serially shift out the 32 bit OCT calibration codes (16 bit R_S OCT and 16 bit R_T OCT) from each OCT calibration block to the corresponding I/O buffers. Only one OCT calibration block can send out the codes at any time by asserting only one $ENASERN$ signal at a time. After you deassert $ENAOCT$, wait at least one $OCTUSRCLK$ cycle to enable any $ENASERN$ signal to begin serial transfer. To shift the 32 bit code from the OCT calibration block N , you must assert $ENASERN$ for exactly 32 $OCTUSRCLK$ cycles. Between two consecutive asserted $ENASER$ signals, there must be at least one $OCTUSRCLK$ cycle gap, as shown in the preceding figure.

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. The preceding figure shows the $S2PEN_A$ 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 $S2PEN_A$ asserted at the same time, or at a different time, even while another OCT calibration block is calibrating and serially shifting codes. The $S2PEN_A$ 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 $S2PEN_A$ is asserted for parallel codes transfer.

Example of Using Multiple OCT Calibration Blocks

Figure 5-14: OCT User-Mode Signal Timing Waveform for Two OCT Blocks

This figure 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 $ENASERN$ signal is asserted. As shown in the preceding figure, 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.

I/O Termination Schemes for Stratix V Devices

Table 5-21: Termination Schemes for Different I/O Standards

I/O Standard	External Termination Scheme
3.3-V LVTTTL/3.3-V LVCMOS	No external termination required
2.5-V LVCMOS	
1.8-V LVCMOS	
1.5-V LVCMOS	
1.2-V LVCMOS	
SSTL-2 Class I	Single-Ended SSTL I/O Standard Termination
SSTL-2 Class II	
SSTL-18 Class I	
SSTL-18 Class II	
SSTL-15 Class I	
SSTL-15 Class II	Single-Ended HSTL I/O Standard Termination
1.8-V HSTL Class I	
1.8-V HSTL Class II	
1.5-V HSTL Class I	
1.5-V HSTL Class II	
1.2-V HSTL Class I	Differential SSTL I/O Standard Termination
1.2-V HSTL Class II	
Differential SSTL-2 Class I	
Differential SSTL-2 Class II	
Differential SSTL-18 Class I	
Differential SSTL-18 Class II	Differential SSTL I/O Standard Termination
Differential SSTL-15 Class I	
Differential SSTL-15 Class II	

I/O Standard	External Termination Scheme
Differential 1.8-V HSTL Class I	Differential HSTL I/O Standard Termination
Differential 1.8-V HSTL Class II	
Differential 1.5-V HSTL Class I	
Differential 1.5-V HSTL Class II	
Differential 1.2-V HSTL Class I	
Differential 1.2-V HSTL Class II	
LVDS	LVDS I/O Standard Termination
RSDS	RSDS/mini-LVDS I/O Standard Termination
Mini-LVDS	
LVPECL	Differential LVPECL I/O Standard Termination
SSTL-15 ⁽¹¹⁾	No external termination required
SSTL-135 ⁽¹¹⁾	
SSTL-125 ⁽¹¹⁾	
SSTL-12	
HSUL-12	
Differential SSTL-15 ⁽¹¹⁾	
Differential SSTL-135 ⁽¹¹⁾	
Differential SSTL-125 ⁽¹¹⁾	
Differential SSTL-12	
Differential HSUL-12	

Single-ended I/O Termination

Voltage-referenced I/O standards require an input 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-12, SSTL-125, SSTL-135, and SSTL-15 typically do not require external board termination.

Intel recommends that you use OCT with these I/O standards to save board space and cost. OCT reduces the number of external termination resistors used.

Note: You cannot use R_S and R_T OCT simultaneously. For more information, refer to the related information.

⁽¹¹⁾ Intel recommends that you use OCT with these I/O standards to save board space and cost. OCT reduces the number of external termination resistors used.

Figure 5-15: SSTL I/O Standard Termination

This figure shows the details of SSTL I/O termination on Stratix V devices. This is not applicable for SSTL-12, SSTL-15, SSTL-125, and SSTL-135 I/O standards.

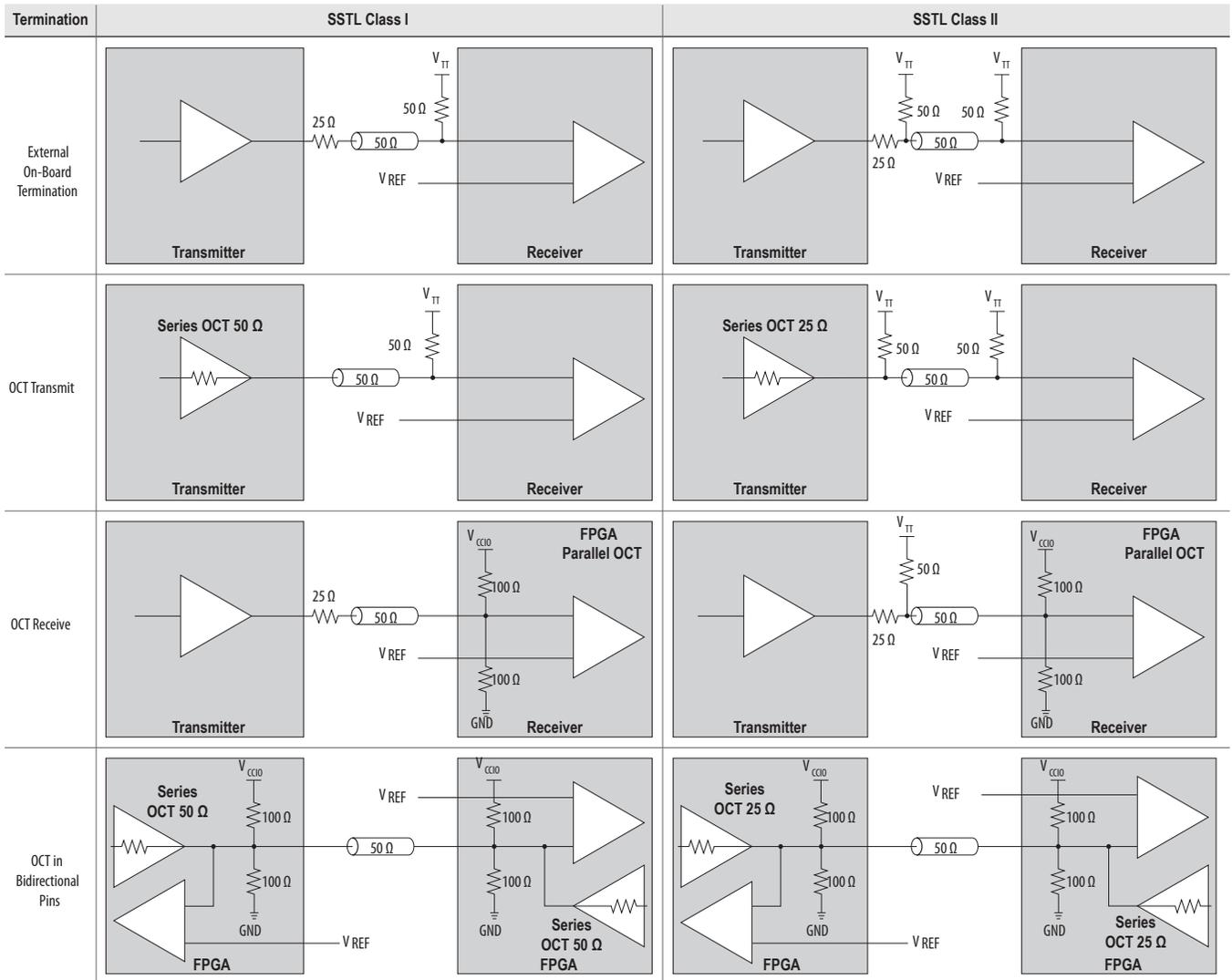
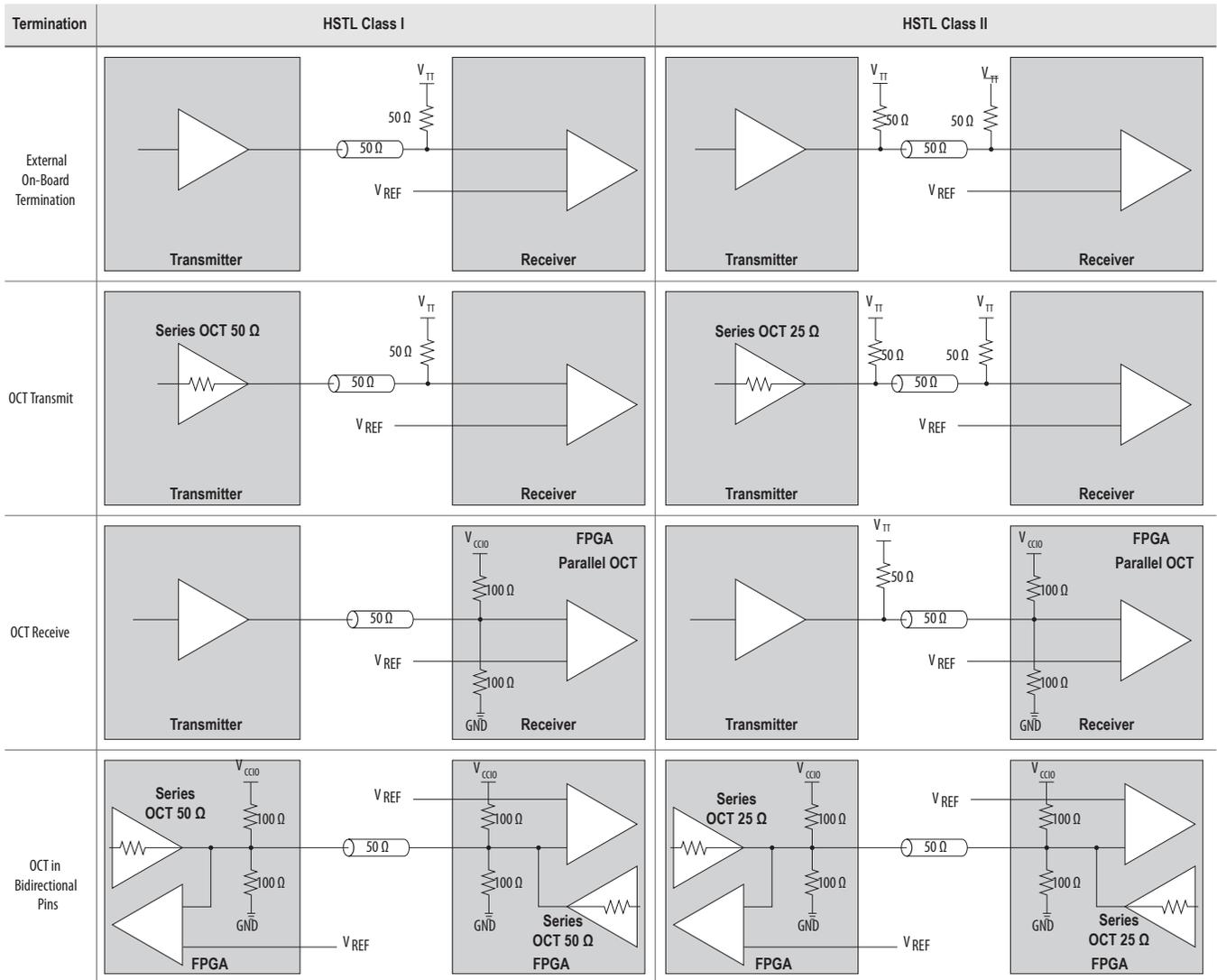


Figure 5-16: HSTL I/O Standard Termination

This figure shows the details of HSTL I/O termination on the Stratix V devices. This is not applicable for HSUL-12 I/O standard.

**Related Information**

[Dynamic OCT in Stratix V Devices](#) on page 5-31

Differential I/O Termination

The I/O pins are organized in pairs to support differential I/O standards. Each I/O pin pair can support differential input and output buffers.

The supported I/O standards such as Differential SSTL-12, Differential SSTL-15, Differential SSTL-125, and Differential SSTL-135 typically do not require external board termination.

Intel recommends that you use OCT with these I/O standards to save board space and cost. OCT reduces the number of external termination resistors used.

Differential HSTL, SSTL, and HSUL Termination

Differential HSTL, SSTL, and HSUL inputs use LVDS differential input buffers with R_D support.

Differential HSTL, SSTL, and HSUL outputs are not true differential outputs. These I/O standards use two single-ended outputs with the second output programmed as inverted.

Figure 5-17: Differential SSTL I/O Standard Termination

This figure shows the details of Differential SSTL I/O termination on Stratix V devices. This is not applicable for differential SSTL-12, differential SSTL-15, differential SSTL-125, differential SSTL-135, and differential HSUL-12 I/O standards.

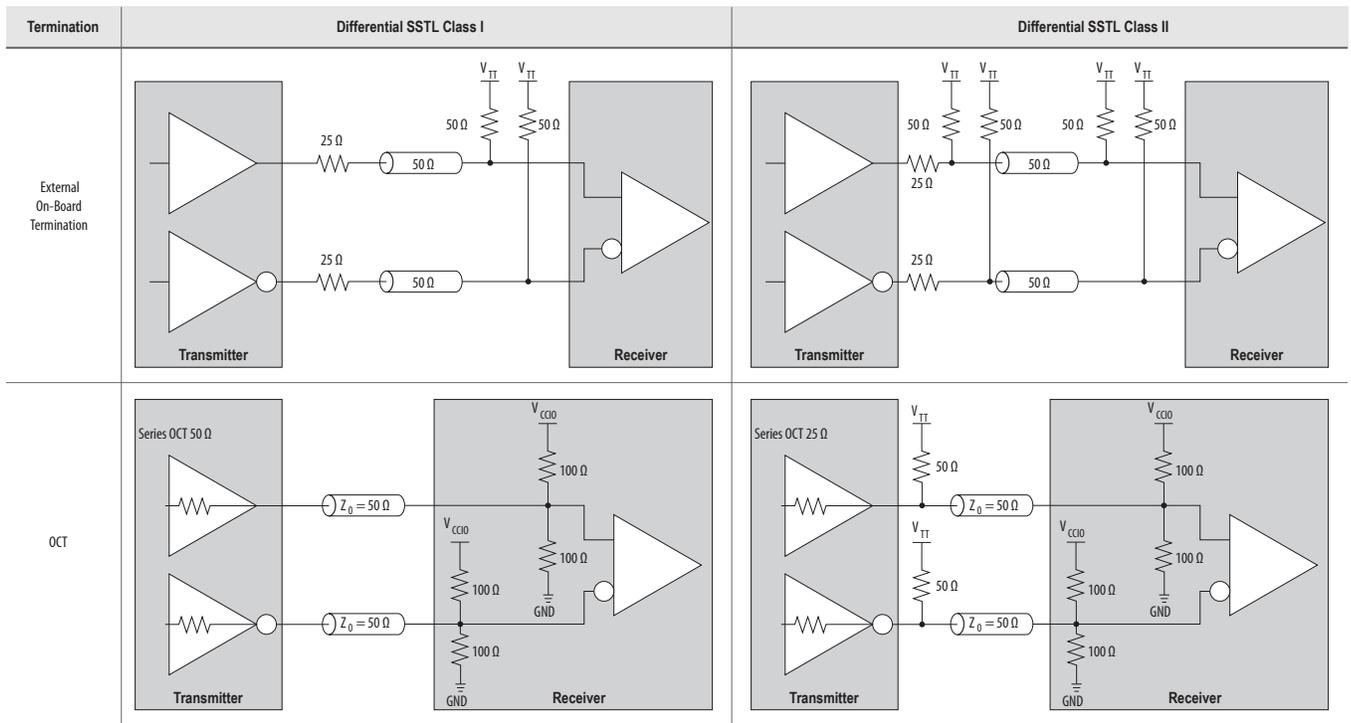
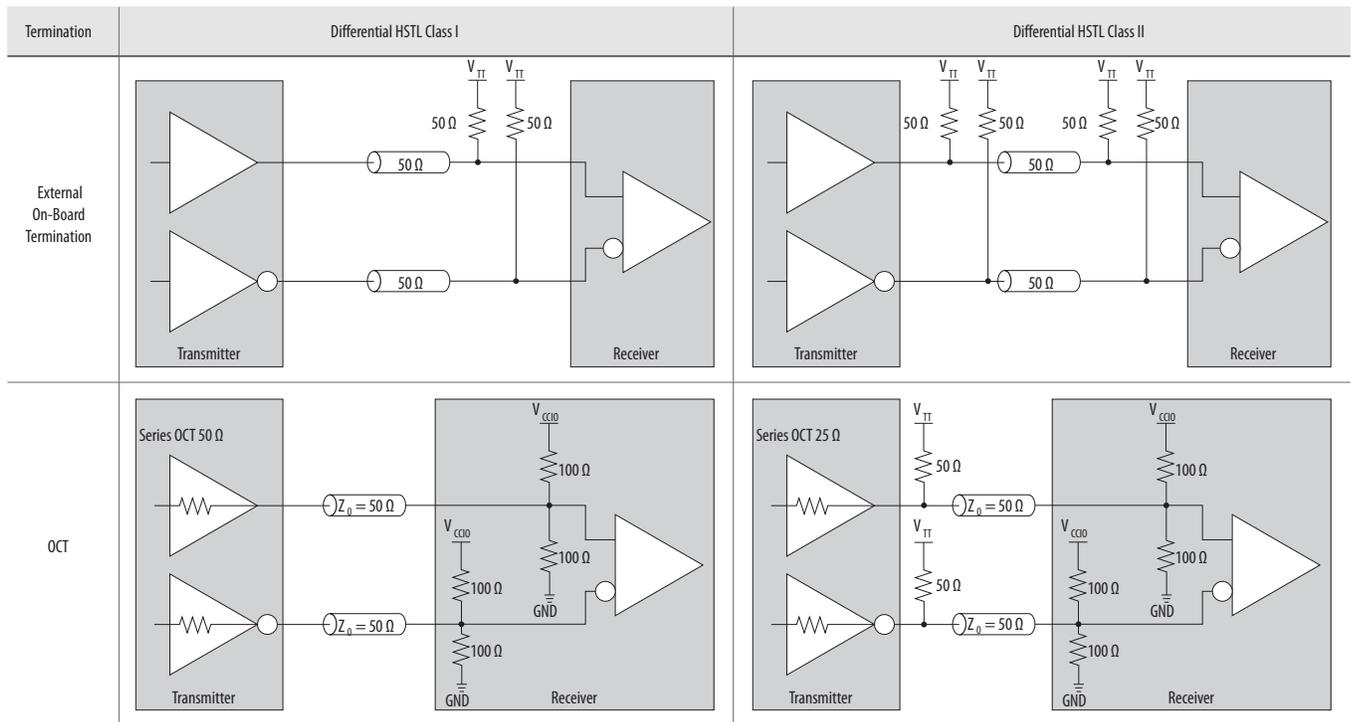


Figure 5-18: Differential HSTL I/O Standard Termination

This figure shows the details of Differential HSTL I/O standard termination on Stratix V devices. This is not applicable for differential HSUL-12 I/O standard.



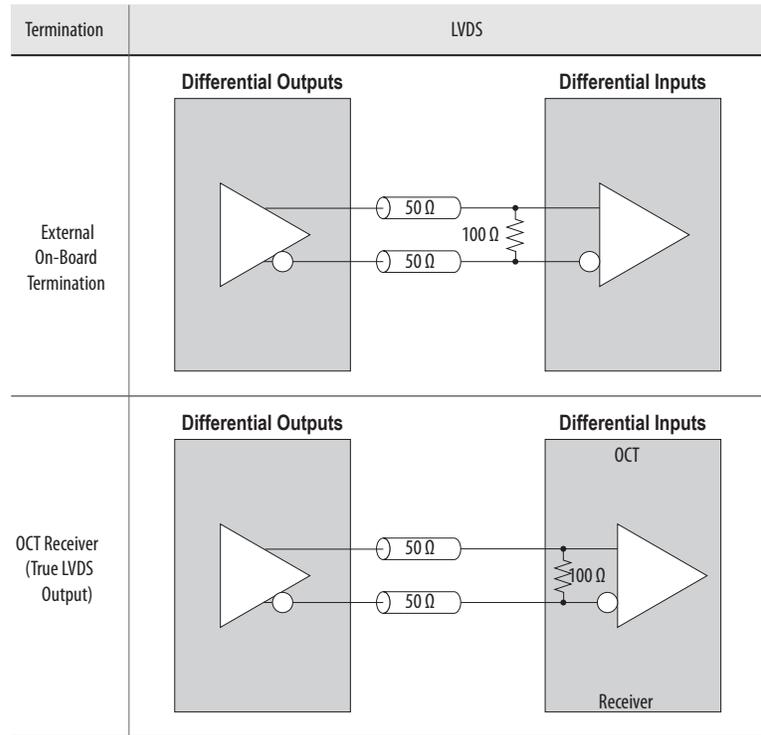
LVDS, RSDS, and Mini-LVDS Termination

All I/O banks have dedicated circuitry to support the true LVDS, RSDS, and mini-LVDS I/O standards by using true LVDS output buffers without resistor networks.

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 if V_{CCPD} is set to 2.5 V.

Figure 5-19: LVDS I/O Standard Termination

This figure shows the LVDS I/O standard termination. The on-chip differential resistor is available in all I/O banks.



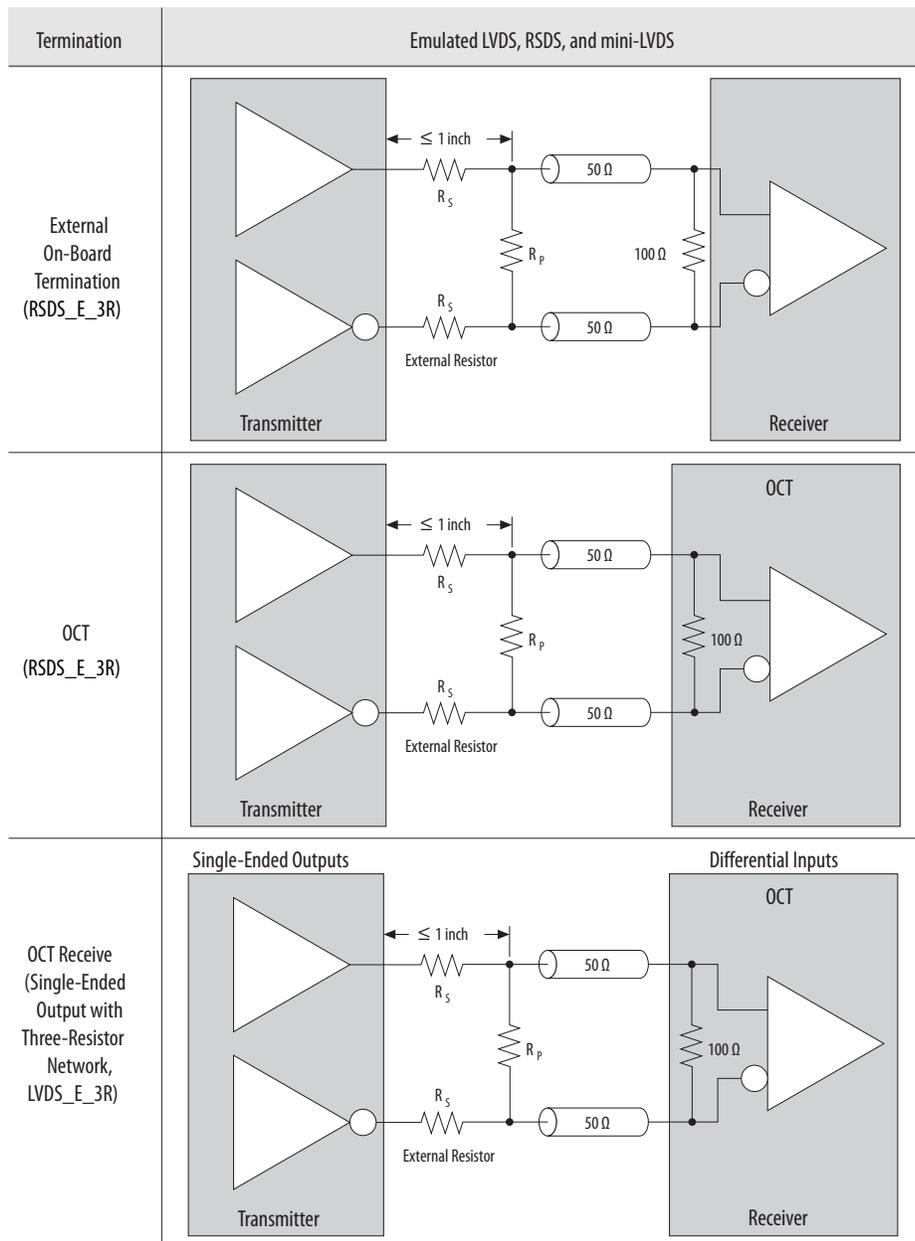
Emulated LVDS, RSDS, and Mini-LVDS Termination

The I/O banks also support emulated LVDS, RSDS, and mini-LVDS I/O standards.

Emulated LVDS, RSDS and mini-LVDS output buffers use two single-ended output buffers with an external three-resistor network, and can be tri-stated.

Figure 5-20: Emulated LVDS, RSDS, or Mini-LVDS I/O Standard Termination

The output buffers, as shown in this figure, are available in all I/O banks. For LVDS output with a three-resistor network, R_S is 120 Ω and R_P is 170 Ω . For RSDS and Mini-LVDS output, R_S and R_P values are pending characterization.



To meet the RSDS or mini-LVDS specifications, you require a resistor network to attenuate the output-voltage swing.

You can modify the three-resistor network values to reduce power or improve the noise margin. Choose resistor values that satisfy the following equation.

Figure 5-21: Resistor Network Calculation

$$\frac{R_S \times \frac{R_P}{2}}{R_S + \frac{R_P}{2}} = 50 \Omega$$

Note: Altera recommends that you perform additional simulations with IBIS or SPICE models to validate that the custom resistor values meet the RSDS or mini-LVDS I/O standard requirements.

For information about the data rates supported for external three-resistor network, refer to the device datasheet.

Related Information

- [Stratix V Device Datasheet](#)
- [Stratix V Device Datasheet](#)

LVPECL Termination

The Stratix V devices support the LVPECL I/O standard on input clock pins only:

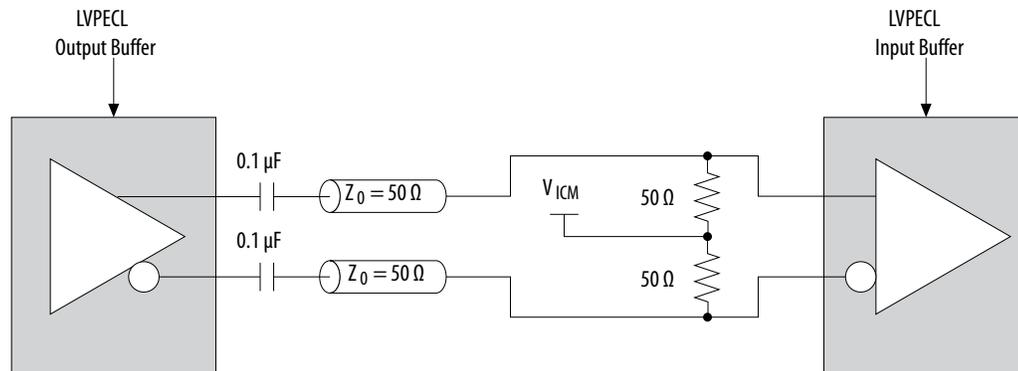
- LVPECL input operation is supported using LVDS input buffers.
- LVPECL output operation is not supported.

Use AC coupling if the LVPECL common-mode voltage of the output buffer does not match the LVPECL input common-mode voltage.

Note: Altera recommends that you use IBIS models to verify your LVPECL AC/DC-coupled termination.

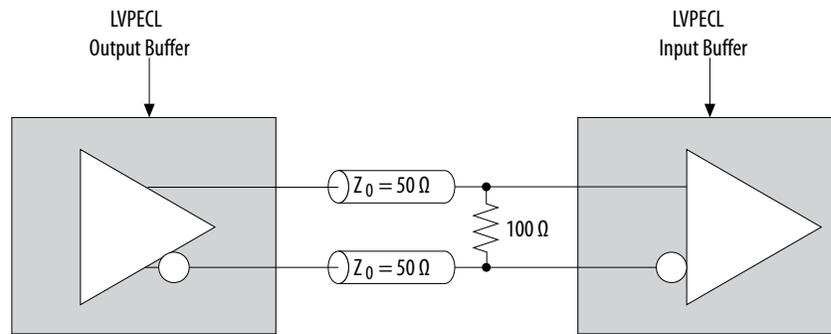
Figure 5-22: LVPECL AC-Coupled Termination

The 50 Ω resistors used at the receiver end are external to the device.



Support for DC-coupled LVPECL is available if the LVPECL output common mode voltage is within the Stratix V LVPECL input buffer specification.

Figure 5-23: LVPECL DC-Coupled Termination



I/O Features in Stratix V Devices Revision History

Document Version	Changes
2018.08.09	Updated the note in <i>Dynamic OCT in Stratix V Devices</i> topic.

Date	Version	Changes
June 2016	2016.06.10	<ul style="list-style-type: none"> Clarified the example quoted in Non-Voltage-Referenced I/O Standards can support 2.5 V, 3.0 V and 3.3 V inputs.
December 2015	2015.12.21	<ul style="list-style-type: none"> Added assignment name and supported I/O standards in Summary of Supported Programmable IOE Features and Settings Table. Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i>.
January 2015	2015.01.23	<ul style="list-style-type: none"> Corrected truncated sentence in the note about the recommendation to use dynamic OCT for several I/O standards with DDR3 external memory interface. Clarified that dedicated configuration pins, clock pins and JTAG pins do not support programmable pull-up resistor but these pins have fixed value of internal pull-up resistors. Moved the Open-Drain Output, Bus-Hold Circuitry and Pull-up Resistor sections to Programmable IOE Features in Stratix V Devices. Update Open-Drain Output section with steps to enable open-drain output in Assignment Editor.

Date	Version	Changes
June 2014	2014.06.30	<ul style="list-style-type: none"> Added footnote to clarify that some of the voltage levels listed in the MultiVolt I/O support table are for showing that multiple single-ended I/O standards are not compatible with certain V_{CCIO} voltages. Added information to clarify that programmable output slew rate is available for single-ended and emulated LVDS I/O standards. Finalized calibrated R_S and R_T OCT values and updated the R_T OCT values for HSUL-12 and Differential HSUL-12 I/O standards.
January 2014	2014.01.10	<ul style="list-style-type: none"> Updated statements in several topics to clarify that each modular I/O bank can support multiple I/O standards that use the same voltages. Updated the guideline topic about using the same V_{CCPD} for I/O banks in the same V_{CCPD} group to improve clarity. Clarified that you can only use R_D OCT if V_{CCPD} is 2.5 V. Corrected the topic about LVDS, RSDS, and Mini-LVDS termination to remove the requirement of 2.5 V V_{CCIO}. Only V_{CCPD} of 2.5 V is required for using R_D OCT. Removed all "preliminary" marks.
June 2013	2013.06.21	<ul style="list-style-type: none"> Updated the topic about LVDS input R_D OCT to remove the requirement for setting the V_{CCIO} to 2.5 V. R_D OCT now requires only that the V_{CCPD} is 2.5 V. Updated the topic about LVPECL termination to improve clarity.
May 2013	2013.05.06	<ul style="list-style-type: none"> Moved all links to the Related Information section of respective topics for easy reference. Added link to the known document issues in the Knowledge Base. Removed all references to column and row I/Os. Stratix V devices have I/O banks on the top and bottom only.
January 2013	2013.01.22	<ul style="list-style-type: none"> Corrected the guideline about using the same V_{CCPD} for all I/O banks in a group. Removed references to LVDS single-ended output with single-resistor network (LVDS_E_1R). The Stratix V devices do not support LVDS_E_1R.
December 2012	2012.12.28	<ul style="list-style-type: none"> Reorganized content and updated template. Added table about the termination schemes for different I/O standards. Updated the SSTL and HSTL I/O termination figures to add V_{REF} inputs for OCT in bidirectional pins. Added OCT diagram for LVDS single-ended output with single-resistor network (LVDS_E_1R). Removed the "Summary of OCT Assignments" table and merged the information into the restructured OCT tables.

Date	Version	Changes
June 2012	1.5	<ul style="list-style-type: none"> Added "Summary of OCT Assignments" and "LVDS Channels" sections. Updated Table 5-2, Table 5-3, Table 5-4, Table 5-5, and Table 5-8. Updated "Pull-Up Resistor", "Differential Output Voltage", and "Programmable IOE Delay" sections.
November 2011	1.4	<ul style="list-style-type: none"> Updated Figure 5-2. Updated Table 5-3, Table 5-4, and Table 5-5.
May 2011	1.3	<ul style="list-style-type: none"> 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 "VCCPD 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.

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

6

2020.09.10

SV51007



Subscribe

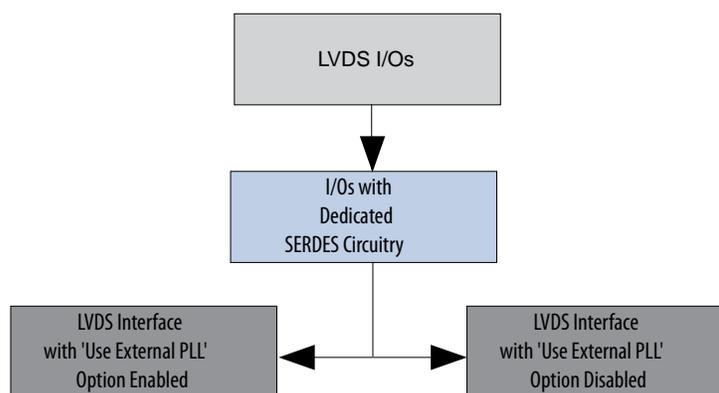


Send Feedback

The high-speed differential I/O interfaces and dynamic phase alignment (DPA) features in Stratix V devices provide advantages over single-ended I/Os and contribute to the achievable overall system bandwidth. Stratix V devices support low-voltage differential signaling (LVDS), mini-LVDS, and reduced swing differential signaling (RSDS) differential I/O standards.

The following figure shows the I/O bank support for high-speed differential I/O in the Stratix V devices.

Figure 6-1: I/O Bank Support for High-Speed Differential I/O



Related Information

- [I/O Standards Support in Stratix V Devices](#) on page 5-2
Provides information about the supported differential I/O standards.
- [Stratix V Device Handbook: Known Issues](#)
Lists the planned updates to the *Stratix V Device Handbook* chapters.
- [I/O Features in Stratix V Devices](#)
Provides information about the supported differential I/O standards.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

Dedicated High-Speed Circuitries in Stratix V Devices

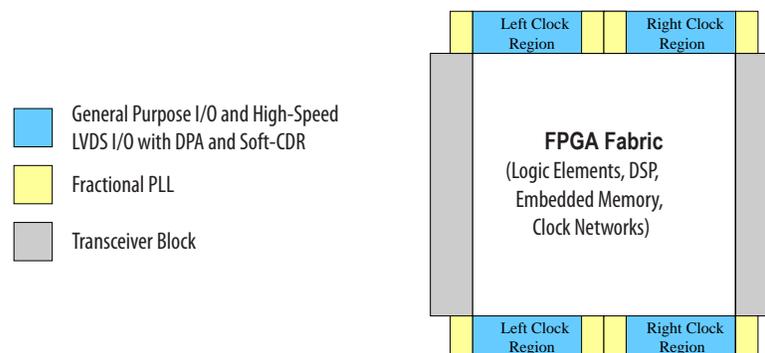
The following dedicated circuitries are available in the Stratix V device family to support high-speed differential I/O:

- Differential I/O buffer
- Transmitter serializer
- Receiver deserializer
- Data realignment (Bit-slip)
- DPA
- Synchronizer (FIFO buffer)
- Phase-locked loops (PLLs)

SERDES and DPA Bank Locations in Stratix V Devices

The dedicated serializer/deserializer (SERDES) and DPA circuitry that supports high-speed differential I/Os is located in the top and bottom banks of the Stratix V devices.

Figure 6-2: High-Speed Differential I/Os with DPA Locations in Stratix V Devices



Related Information

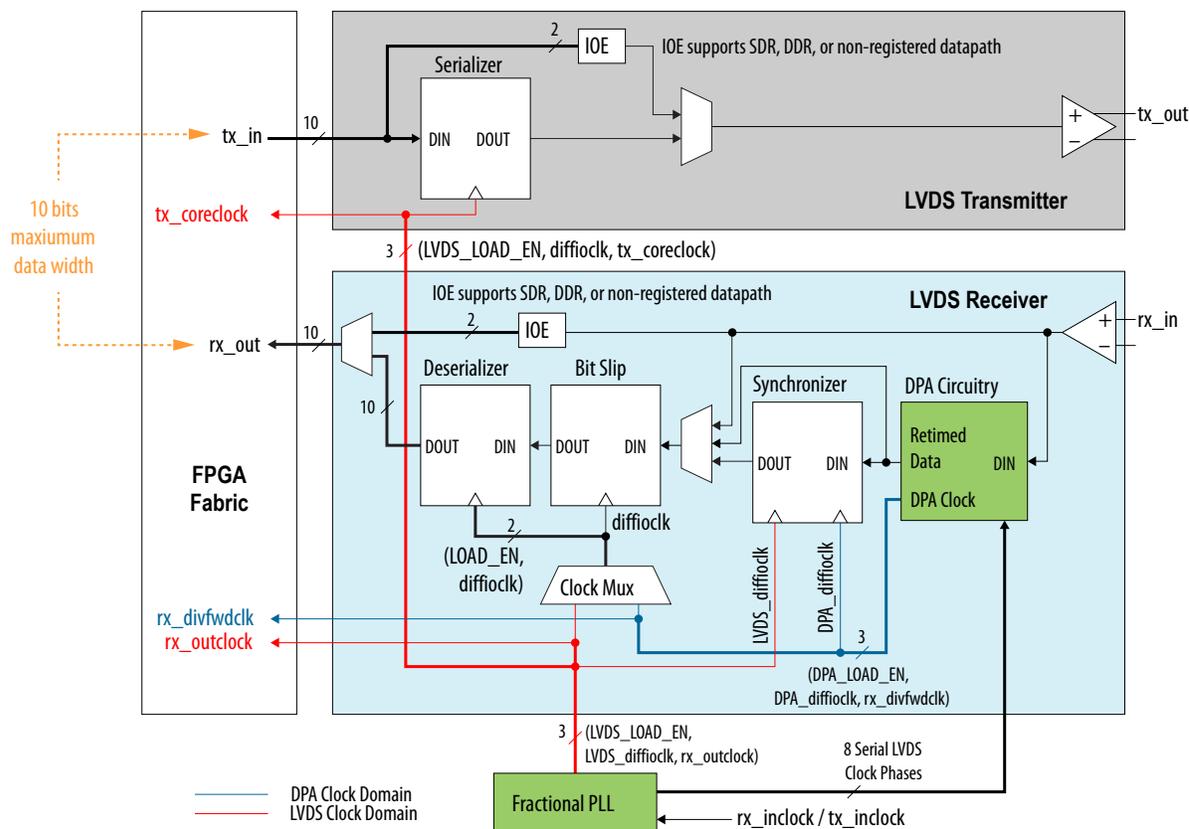
[PLLs and Clocking for Stratix V Devices](#) on page 6-8

LVDS SERDES Circuitry

The Stratix V devices have built-in serializer/deserializer (SERDES) circuitry that supports high-speed LVDS interfaces. You can configure the SERDES circuitry to support source-synchronous communication protocols such as RapidIO[®], XSBI, serial peripheral interface (SPI), and asynchronous protocols such as Gigabit Ethernet (GbE) and SGMII.

The following figure shows a transmitter and receiver block diagram for the LVDS SERDES circuitry with the interface signals of the transmitter and receiver data paths.

Figure 6-3: LVDS SERDES



The preceding figure shows a shared PLL between the transmitter and receiver. If the transmitter and receiver do not share the same PLL, you require two fractional PLLs. In single data rate (SDR) and double data rate (DDR) modes, the data width is 1 and 2 bits, respectively.

The ALTLVDS transmitter and receiver requires various clock and load enable signals from a fractional PLL. The Intel Quartus Prime software configures the PLL settings automatically. The software is also responsible for generating the various clock and load enable signals based on the input reference clock and selected data rate.

Note: For the maximum data rate supported by the Stratix V devices, refer to the device overview.

Related Information

- [Stratix V Device Overview](#)
- [LVDS SERDES Transmitter/Receiver IP Cores User Guide](#)
Provides a list of the LVDS transmitter and receiver ports and settings using ALTLVDS.
- [Guideline: Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

SERDES I/O Standards Support in Stratix V Devices

The following tables list the I/O standards supported by the SERDES receiver and transmitter, and the respective Intel Quartus Prime software assignment values.

The SERDES receiver and transmitter also support all differential HSTL, differential HSUL, and differential SSTL I/O standards.

Table 6-1: SERDES Receiver I/O Standards Support

I/O Standard	Intel Quartus Prime Software Assignment Value
True LVDS	LVDS
Differential 1.2 V HSTL Class I	Differential 1.2-V HSTL Class I
Differential 1.2 V HSTL Class II	Differential 1.2-V HSTL Class II
Differential HSUL-12	Differential 1.2-V HSUL
Differential SSTL-12	Differential 1.2-V SSTL
Differential SSTL-125	Differential 1.25-V SSTL
Differential SSTL-135	Differential 1.35-V SSTL
Differential 1.5 V HSTL Class I	Differential 1.5-V HSTL Class I
Differential 1.5 V HSTL Class II	Differential 1.5-V HSTL Class II
Differential SSTL-15	Differential 1.5-V SSTL
Differential SSTL-15 Class I	Differential 1.5-V SSTL Class I
Differential SSTL-15 Class II	Differential 1.5-V SSTL Class II
Differential 1.8 V HSTL Class I	Differential 1.8-V HSTL Class I
Differential 1.8 V HSTL Class II	Differential 1.8-V HSTL Class II
Differential SSTL-18 Class I	Differential 1.8-V SSTL Class I
Differential SSTL-18 Class II	Differential 1.8-V SSTL Class II
Differential SSTL-2 Class I	Differential 2.5-V SSTL Class I
Differential SSTL-2 Class II	Differential 2.5-V SSTL Class II

Table 6-2: SERDES Transmitter I/O Standards Support

I/O Standard	Intel Quartus Prime Software Assignment Value
True LVDS	LVDS
Differential 1.2 V HSTL Class I	Differential 1.2-V HSTL Class I
Differential 1.2 V HSTL Class II	Differential 1.2-V HSTL Class II
Differential HSUL-12	Differential 1.2-V HSUL
Differential SSTL-12	Differential 1.2-V SSTL
Differential SSTL-125	Differential 1.25-V SSTL
Differential SSTL-135	Differential 1.35-V SSTL
Differential 1.5 V HSTL Class I	Differential 1.5-V HSTL Class I
Differential 1.5 V HSTL Class II	Differential 1.5-V HSTL Class II

I/O Standard	Intel Quartus Prime Software Assignment Value
Differential SSTL-15	Differential 1.5-V SSTL
Differential SSTL-15 Class I	Differential 1.5-V SSTL Class I
Differential SSTL-15 Class II	Differential 1.5-V SSTL Class II
Differential 1.8 V HSTL Class I	Differential 1.8-V HSTL Class I
Differential 1.8 V HSTL Class II	Differential 1.8-V HSTL Class II
Differential SSTL-18 Class I	Differential 1.8-V SSTL Class I
Differential SSTL-18 Class II	Differential 1.8-V SSTL Class II
Differential SSTL-2 Class I	Differential 2.5-V SSTL Class I
Differential SSTL-2 Class II	Differential 2.5-V SSTL Class II
Emulated LVDS	LVDS_E_3R
mini-LVDS	mini-LVDS
Emulated mini-LVDS	mini-LVDS_E_3R
RSDS	RSDS
Emulated RSDS	RSDS_E_3R

True LVDS Buffers in Stratix V Devices

The Stratix V device family supports LVDS on all I/O banks:

- All I/Os support true LVDS input buffers with R_D OCT or true LVDS output buffers.
- Stratix V devices offer single-ended I/O reference clock support for the fractional PLL that drives the SERDES.

The following tables list the number of true LVDS buffers supported in Stratix V devices with these conditions:

- The LVDS channel count does not include dedicated clock pins.
- Dedicated SERDES and DPA is available for top and bottom banks only.

Table 6-3: LVDS Channels Supported in Stratix V E Devices

Member Code	Package	Side	TX	RX
E9 and EB	H40-H1517	Top	87	87
		Bottom	87	87
	F45-F1932	Top	105	105
		Bottom	105	105

Table 6-4: LVDS Channels Supported in Stratix V GX Devices

Member Code	Package	Side	TX	RX
A3	EH29-H780	Top	51	51
		Bottom	39	39
	HF35-F1152	Top	57	57
		Bottom	51	51
	KF35-F1152	Top	54	54
		Bottom	54	54
KF40-F1517	Top	87	87	
	Bottom	87	87	
A4	HF35-F1152	Top	63	63
		Bottom	75	75
	KF35-F1152	Top	54	54
		Bottom	54	54
	KF40-F1517	Top	87	87
		Bottom	87	87
A5 and A7	HF35-F1152	Top	63	63
		Bottom	75	75
	KF35-F1152	Top	54	54
		Bottom	54	54
	KF40-F1517	Top	87	87
		Bottom	87	87
	NF40-F1517	Top	75	75
		Bottom	75	75
	NF45-F1932	Top	105	105
		Bottom	105	105
A9 and AB	KH40-1517	Top	87	87
		Bottom	87	87
	NF45-F1932	Top	105	105
		Bottom	105	105
B5 and B6	RF40-F1517	Top	54	54
		Bottom	54	54
	RF43-F1760	Top	75	75
		Bottom	75	75

Member Code	Package	Side	TX	RX
B9 and BB	RH43-H1760	Top	75	75
		Bottom	75	75

Table 6-5: LVDS Channels Supported in Stratix V GS Devices

Member Code	Package	Side	TX	RX
D3	EH29-H780	Top	51	51
		Bottom	39	39
	HF35-F1152	Top	57	57
		Bottom	51	51
D4	EH29-H780	Top	51	51
		Bottom	39	39
	HF35-F1152	Top	57	57
		Bottom	51	51
	KF40-F1517	Top	87	87
		Bottom	87	87
D5	HF35-F1152	Top	63	63
		Bottom	75	75
	KF40-F1517	Top	87	87
		Bottom	87	87
D6 and D8	KF40-F1517	Top	87	87
		Bottom	87	87
	NF45-F1932	Top	105	105
		Bottom	105	105

Table 6-6: LVDS Channels Supported in Stratix V GT Devices

Member Code	Package	Side	TX	RX
C5 and C7	KF40-F1517	Top	75	75
		Bottom	75	75

Related Information

Guideline: [Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

Emulated LVDS Buffers in Stratix V Devices

The Stratix V device family supports emulated LVDS on all I/O banks:

- You can use unutilized true LVDS input channels as emulated LVDS output buffers (eTX) for serialization factors of 1 and 2.
- The emulated LVDS output buffers use two single-ended output buffers with an external resistor network to support LVDS, mini-LVDS, and RSDS I/O standards.
- The emulated differential output buffers support tri-state capability.

High-Speed I/O Design Guidelines for Stratix V Devices

There are several considerations that require your attention to ensure the success of your designs. Unless noted otherwise, these design guidelines apply to all variants of this device family.

PLLs and Clocking for Stratix V Devices

To generate the parallel clocks (`rx_outclock` and `tx_outclock`) and high-speed clocks (`diffioclck`), the Stratix V devices provide fractional PLLs in the high-speed differential I/O receiver and transmitter channels.

Related Information

- [SERDES and DPA Bank Locations in Stratix V Devices](#) on page 6-2
Provides information about the PLL locations available for each Stratix V device.
- [Guideline: Use High-Speed Clock from PLL to Clock LVDS SERDES Only](#) on page 6-8
- [Guideline: Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

Guideline: Use PLLs in Integer PLL Mode for LVDS

To drive the LVDS channels, you must use the PLLs in integer PLL mode. The center or corner PLLs can drive the LVDS receiver and transmitter channels.

However, the clock tree network cannot cross over to different I/O regions. For example, the top left corner PLL cannot cross over to drive the LVDS receiver and transmitter channels on the top right I/O bank.

Related Information

[Pin Placement Guidelines for DPA Differential Channels](#) on page 6-13
Provides more information about the fractional PLL clocking restrictions.

Guideline: Use High-Speed Clock from PLL to Clock LVDS SERDES Only

The high-speed clock generated from the PLL is intended to clock the LVDS SERDES circuitry only. Do not use the high-speed clock to drive other logic because the allowed frequency to drive the core logic is restricted by the PLL F_{OUT} specification.

For more information about the F_{OUT} specification, refer to the device datasheet.

Note: Spread-spectrum input clock is not supported in LVDS.

Related Information

[Stratix V Device Datasheet](#)

LVDS Interface with External PLL Mode

The IP Catalog 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 must also instantiate the an Altera_PLL IP core to generate the various clock and load enable signals.

If you enable the **Use External PLL** option with the ALTLVDS transmitter and receiver, the following signals are required from the Altera_PLL IP core:

- Serial clock input to the SERDES of the ALTLVDS transmitter and receiver
- Load enable to the SERDES of the ALTLVDS transmitter and receiver
- Parallel clock used to clock the transmitter FPGA fabric logic and parallel clock used for the receiver
- Asynchronous PLL reset port of the ALTLVDS receiver

Altera_PLL Signal Interface with ALTLVDS IP Core

Table 6-7: Signal Interface Between Altera_PLL and ALTLVDS IP Cores

This table lists the signal interface between the output ports of the Altera_PLL IP core and the input ports of the ALTLVDS transmitter and receiver. As an example, the table lists the serial clock output, load enable output, and parallel clock output generated on ports outclk0, outclk1, and outclk2, along with the locked signal of the Altera_PLL instance. You can choose any of the PLL output clock ports to generate the interface clocks.

From the Altera_PLL IP Core	To the ALTLVDS Transmitter	To the ALTLVDS Receiver
Serial clock output (outclk0) The serial clock output (outclk0) can only drive tx_inclock on the ALTLVDS transmitter, and rx_inclock and rx_dpaclock on the ALTLVDS receiver. This clock cannot drive the core logic.	tx_inclock (serial clock input to the transmitter)	rx_inclock (serial clock input) rx_dpaclock
Load enable output (outclk1)	tx_enable (load enable to the transmitter)	rx_enable (load enable for the deserializer)
Parallel clock output (outclk2)	Parallel clock used inside the transmitter core logic in the FPGA fabric	rx_syncclock (parallel clock input) and parallel clock used inside the receiver core logic in the FPGA fabric
~(locked)	—	pll_areset (asynchronous PLL reset port) The pll_areset signal is automatically enabled for the LVDS receiver in external PLL mode. This signal does not exist for LVDS transmitter instantiation when the external PLL option is enabled.

Note: With soft SERDES, a different clocking requirement is needed.

Related Information

[LVDS SERDES Transmitter/Receiver IP Cores User Guide](#)

More information about the different clocking requirement for soft SERDES.

Altera_PLL Parameter Values for External PLL Mode

The following examples show the clocking requirements to generate output clocks for ALTLVDS_TX and ALTLVDS_RX using the Altera_PLL IP core. The examples set the phase shift with the assumption that the clock and data are edge aligned at the pins of the device.

Note: For other clock and data phase relationships, Altera recommends that you first instantiate your ALTLVDS_RX and ALTLVDS_TX interface without using the external PLL mode option. Compile the IP cores in the Intel Quartus Prime software and take note of the frequency, phase shift, and duty cycle settings for each clock output. Enter these settings in the Altera_PLL IP core parameter editor and then connect the appropriate output to the ALTLVDS_RX and ALTLVDS_TX IP cores.

Table 6-8: Example: Generating Output Clocks Using an Altera_PLL IP Core (No DPA and Soft-CDR Mode)

This table lists the parameter values that you can set in the Altera_PLL parameter editor to generate three output clocks using an Altera_PLL IP core if you are not using DPA and soft-CDR mode.

Parameter	outclk0 (Connects to the tx_inclock port of ALTLVDS_TX and the rx_inclock port of ALTLVDS_RX)	outclk1 (Connects to the tx_enable port of ALTLVDS_TX and the rx_enable port of ALTLVDS_RX)	outclk2 (Used as the core clock for the parallel data registers for both transmitter and receiver, and connects to the rx_synclock port of ALTLVDS_RX)
Frequency	data rate	data rate/serialization factor	data rate/serialization factor
Phase shift	-180°	$[(\text{deserialization factor} - 2) / \text{deserialization factor}] \times 360^\circ$	-180/serialization factor (outclk0 phase shift divided by the serialization factor)
Duty cycle	50%	100/serialization factor	50%

The calculations for phase shift, using the RSKM equation, assume that the input clock and serial data are edge aligned. Introducing a phase shift of -180° to sampling clock (c0) ensures that the input data is center-aligned with respect to the outclk0, as shown in the following figure.

Figure 6-4: Phase Relationship for External PLL Interface Signals

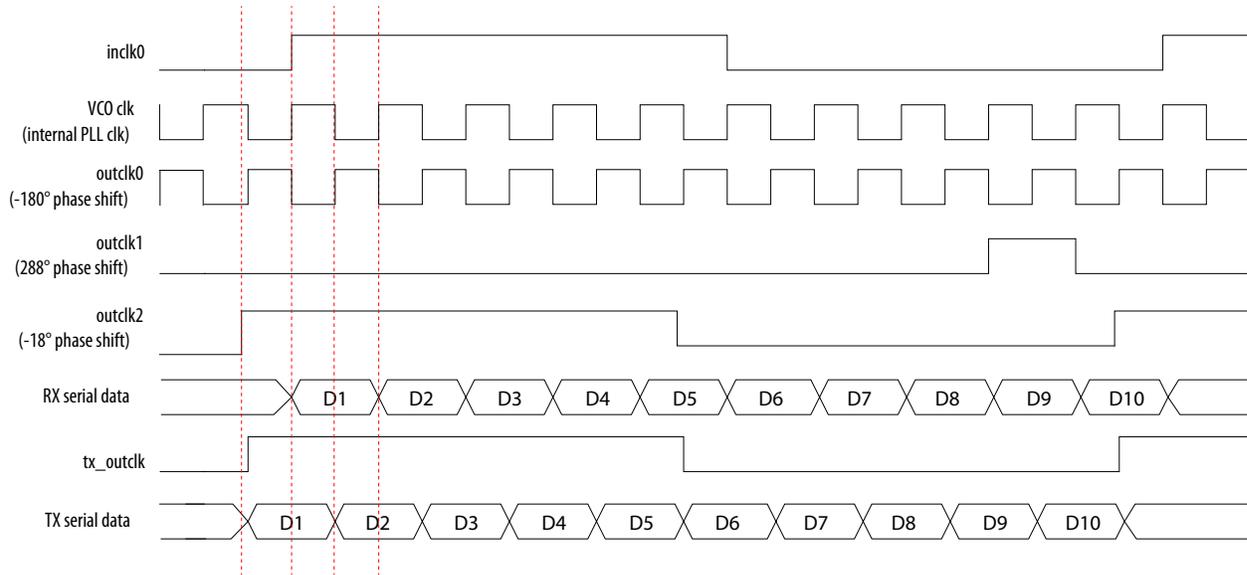


Table 6-9: Example: Generating Output Clocks Using an Altera_PLL IP Core (With DPA and Soft-CDR Mode)

This table lists the parameter values that you can set in the Altera_PLL parameter editor to generate four output clocks using an Altera_PLL IP core if you are using DPA and soft-CDR mode. The `locked` output port of Altera_PLL must be inverted and connected to the `pll_areset` port of the ALTLVDS_RX IP core if you are using DPA and soft-CDR mode.

Parameter	outclk0 (Connects to the <code>tx_inclock</code> port of ALTLVDS_TX and the <code>rx_inclock</code> port of ALTLVDS_RX)	outclk1 (Connects to the <code>tx_enable</code> port of ALTLVDS_TX and the <code>rx_enable</code> port of ALTLVDS_RX)	outclk2 (Used as the core clock for the parallel data registers for both transmitter and receiver, and connects to the <code>rx_synclock</code> port of ALTLVDS_RX)	outclk3 (Connects to the <code>rx_dpaclock</code> port of ALTLVDS_RX)
Frequency	data rate	data rate/serialization factor	data rate/serialization factor	data rate
Phase shift	-180°	$[(\text{deserialization factor} - 2) / \text{deserialization factor}] \times 360^\circ$	-180/serialization factor (outclk0 phase shift divided by the serialization factor)	-180°
Duty cycle	50%	100/serialization factor	50%	50%

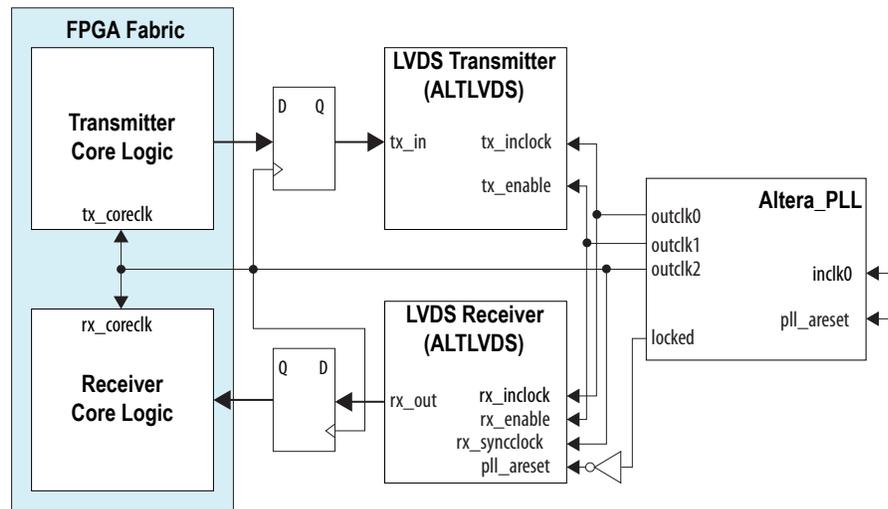
Related Information

[Receiver Skew Margin for Non-DPA Mode](#) on page 6-34

RSKM equation used for the phase shift calculations.

Connection between Altera_PLL and ALTLVDS**Figure 6-5: LVDS Interface with the Altera_PLL IP Core (Without DPA and Soft-CDR Mode)**

This figure shows the connections between the Altera_PLL and ALTLVDS IP core if you are not using DPA and soft-CDR mode.

**Figure 6-6: LVDS Interface with the Altera_PLL IP Core (With DPA)**

This figure shows the connections between the Altera_PLL and ALTLVDS IP core if you are using DPA. The `locked` output port must be inverted and connected to the `pll_areset` port.

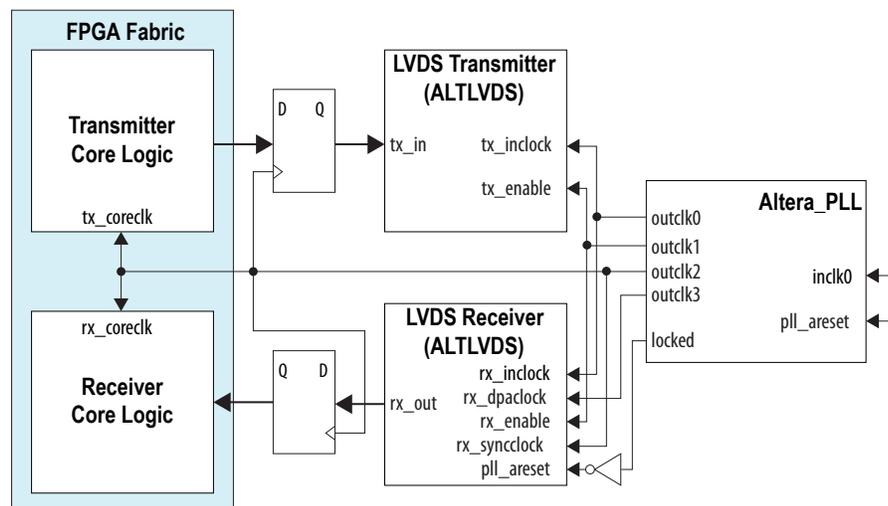
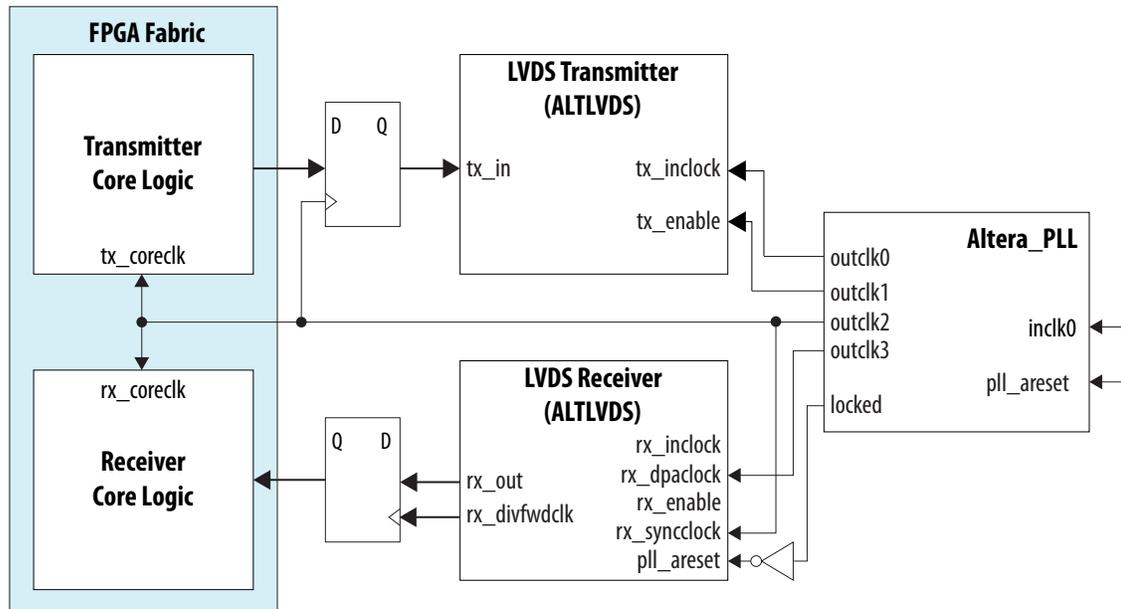


Figure 6-7: LVDS Interface with the Altera_PLL IP Core (With Soft-CDR Mode)

This figure shows the connections between the Altera_PLL and ALTLVDS IP core if you are using soft-CDR mode. The `locked` output port must be inverted and connected to the `pll_areset` port.



When generating the Altera_PLL IP core, the **Left/Right PLL** option is configured to set up the PLL in LVDS mode. Instantiation of `pll_areset` is optional.

The `rx_enable` and `rx_inclock` input ports are not used and can be left unconnected.

Pin Placement Guidelines for DPA Differential Channels

DPA usage adds some constraints on the placement of high-speed differential channels. If DPA-enabled or DPA-disabled differential channels⁽¹²⁾ in the differential banks are used, you must adhere to the differential pin placement guidelines to ensure the proper high-speed operation. The Intel Quartus Prime compiler automatically checks the design and issues an error message if the guidelines are not followed.

Note: The figures in this section show guidelines for using corner and center PLLs but do not necessarily represent the exact locations of the high-speed LVDS I/O banks.

Related Information

Guideline: Use PLLs in Integer PLL Mode for LVDS on page 6-8

Guideline: Using DPA-Enabled Differential Channels

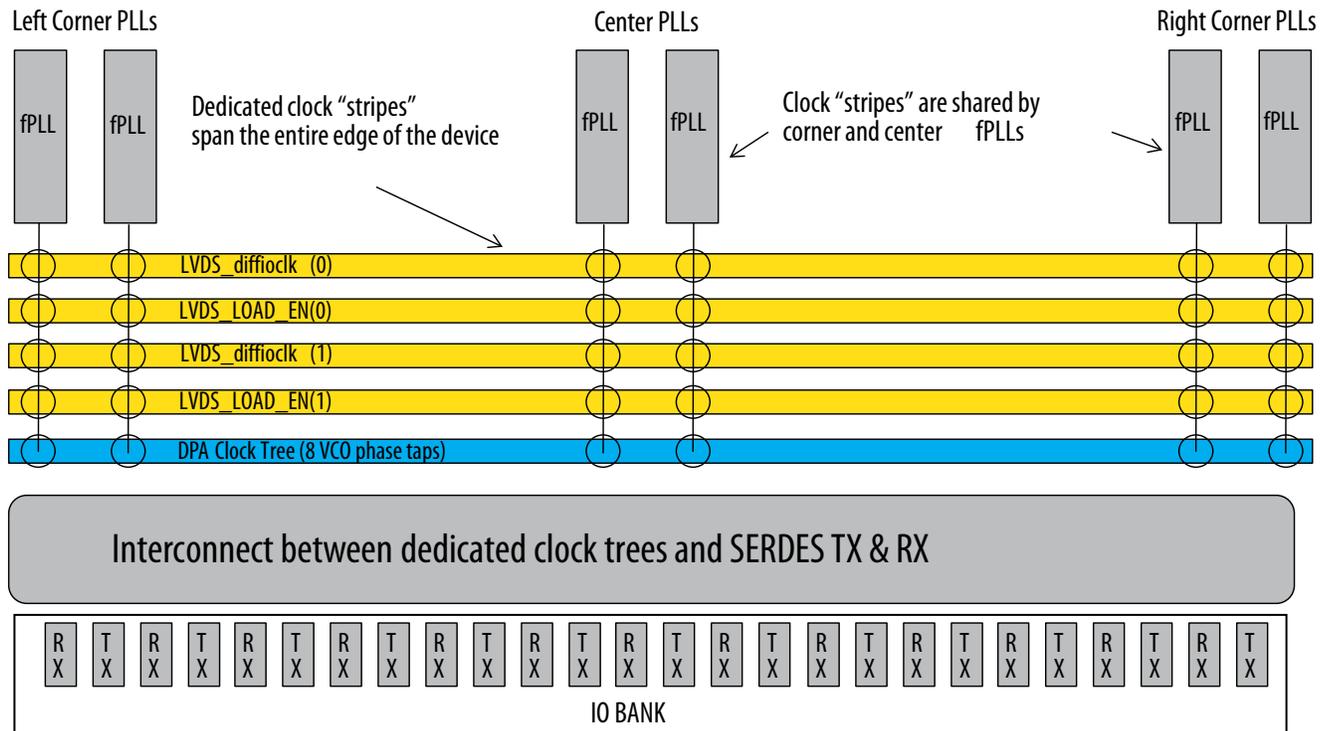
Each differential receiver in an I/O block has a dedicated DPA circuit to align the phase of the clock to the data phase of its associated channel. If you enable a DPA channel in a bank, you can use both single-ended I/Os and differential I/O standards in the bank.

⁽¹²⁾ DPA-enabled differential channels refer to DPA mode or soft-CDR mode while DPA disabled channels refer to non-DPA mode.

You can place double data rate I/O (DDIO) output pins within I/O modules that have the same pad group number as a SERDES differential channel. However, you cannot place SDR I/O output pins within I/O modules that have the same pad group number as a receiver SERDES differential channel. You must implement the input register within the FPGA fabric logic.

The following figure illustrates the clock network for DPA and SERDES resources in Stratix V devices.

Figure 6-8: LVDS and DPA Clock Network

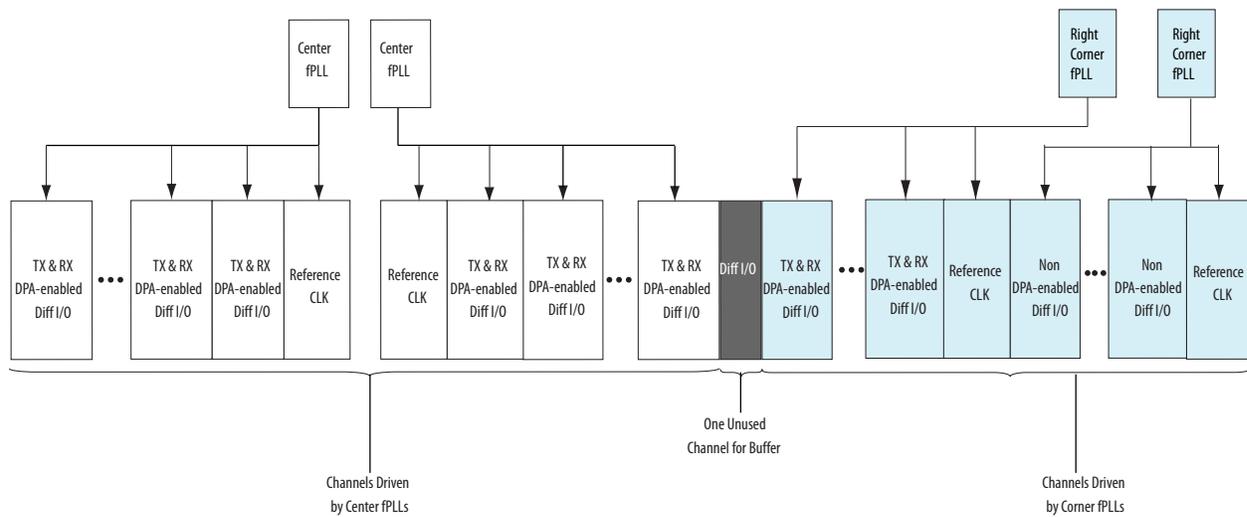


If you use DPA-enabled channels in differential banks, adhere to the following guidelines.

Using Center and Corner PLLs

If two PLLs drive the DPA-enabled channels in a bank—the corner and center PLL drive one group each—there must be at least one row (one differential channel) of separation between the two groups of DPA-enabled channels, as shown in the following figure.

Figure 6-9: Center and Corner PLLs Driving DPA-enabled Differential I/Os in the Same Bank



This separation prevents noise mixing because the two groups can operate at independent frequencies. No separation is necessary if a single PLL is driving both the DPA-enabled channels and DPA-disabled channels.

Using Both Center PLLs

You can use center PLLs to drive DPA-enabled channels simultaneously, if they drive these channels in their adjacent banks only, as shown in the previous figure. The center PLLs cannot drive cross-banks simultaneously. Refer to the following figures.

Figure 6-10: Center PLLs Driving DPA-enabled Differential I/Os

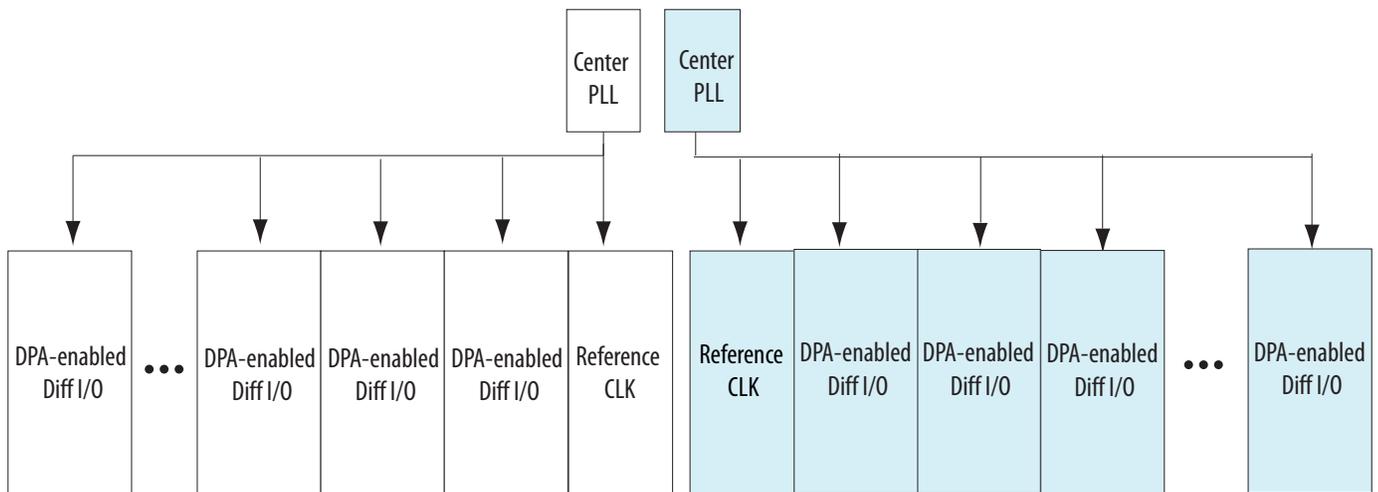
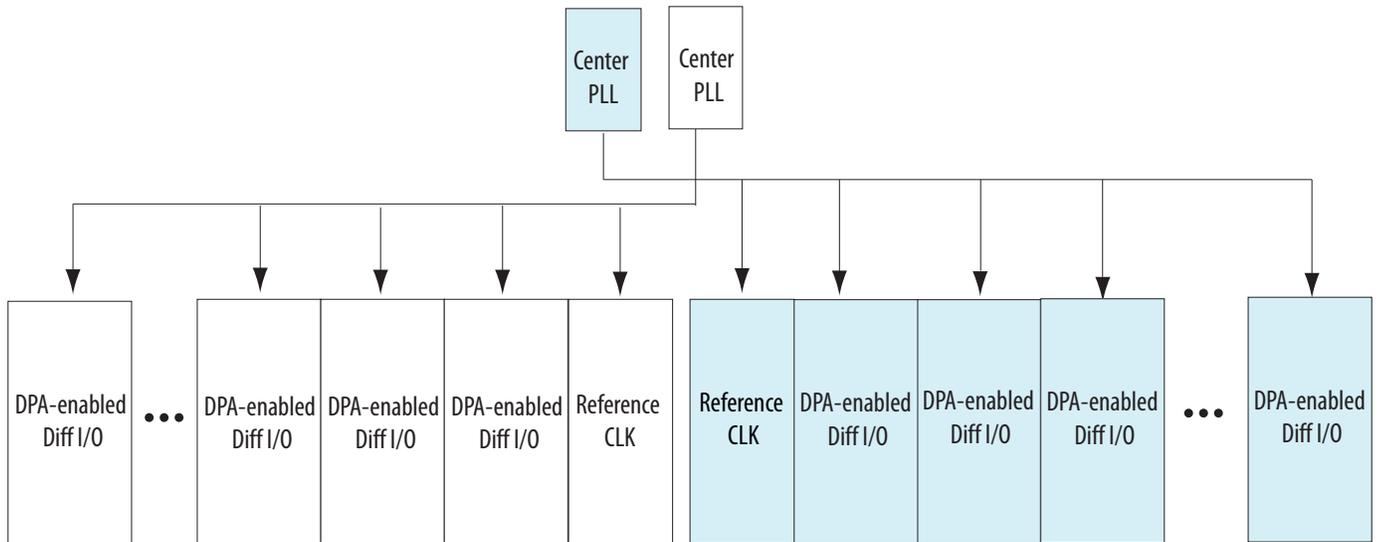


Figure 6-11: Invalid Placement of DPA-enabled Differential I/Os Driven by Both Center PLLs

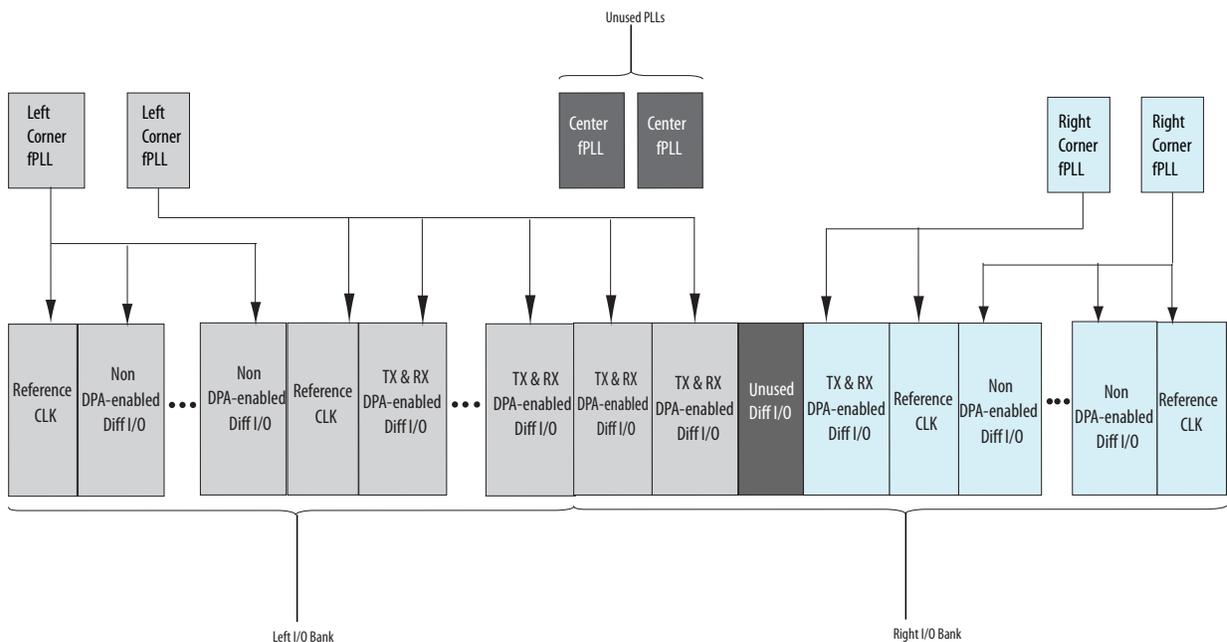


Using Both Corner PLLs

You can use the left and right corner PLLs to drive DPA-enabled channels simultaneously, if they drive the channels in their adjacent banks only. There must be at least one row of separation between the two groups of DPA-enabled channels.

There are two PLL in each corner of the device. However, only one corner PLL can be use to drive DPA-enabled channels in a quadrant.

Figure 6-12: Invalid Usage of Corner PLLs Driving DPA-enabled Differential I/Os



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.

Guideline: Using DPA-Disabled Differential Channels

If you use DPA-disabled channels, adhere to the following guidelines.

DPA-Disabled Channel Driving Distance

Each PLL can drive all the DPA-disabled channels located in the entire bank.

Using Corner and Center PLLs

You can use a corner PLL to drive all transmitter channels and a center PLL to drive all DPA-disabled receiver channels in the same I/O bank. You can drive a transmitter channel and a receiver channel in the same LAB row by two different PLLs. A corner PLL and a center PLL can drive duplex channels in the same I/O bank if the channels that are driven by each PLL are not interleaved. You do not require separation between the group of channels that are driven by the corner and center, left and right PLLs. Refer to the following figures.

Figure 6-13: Corner and Center PLLs Driving DPA-Disabled Differential I/Os in the Same Bank

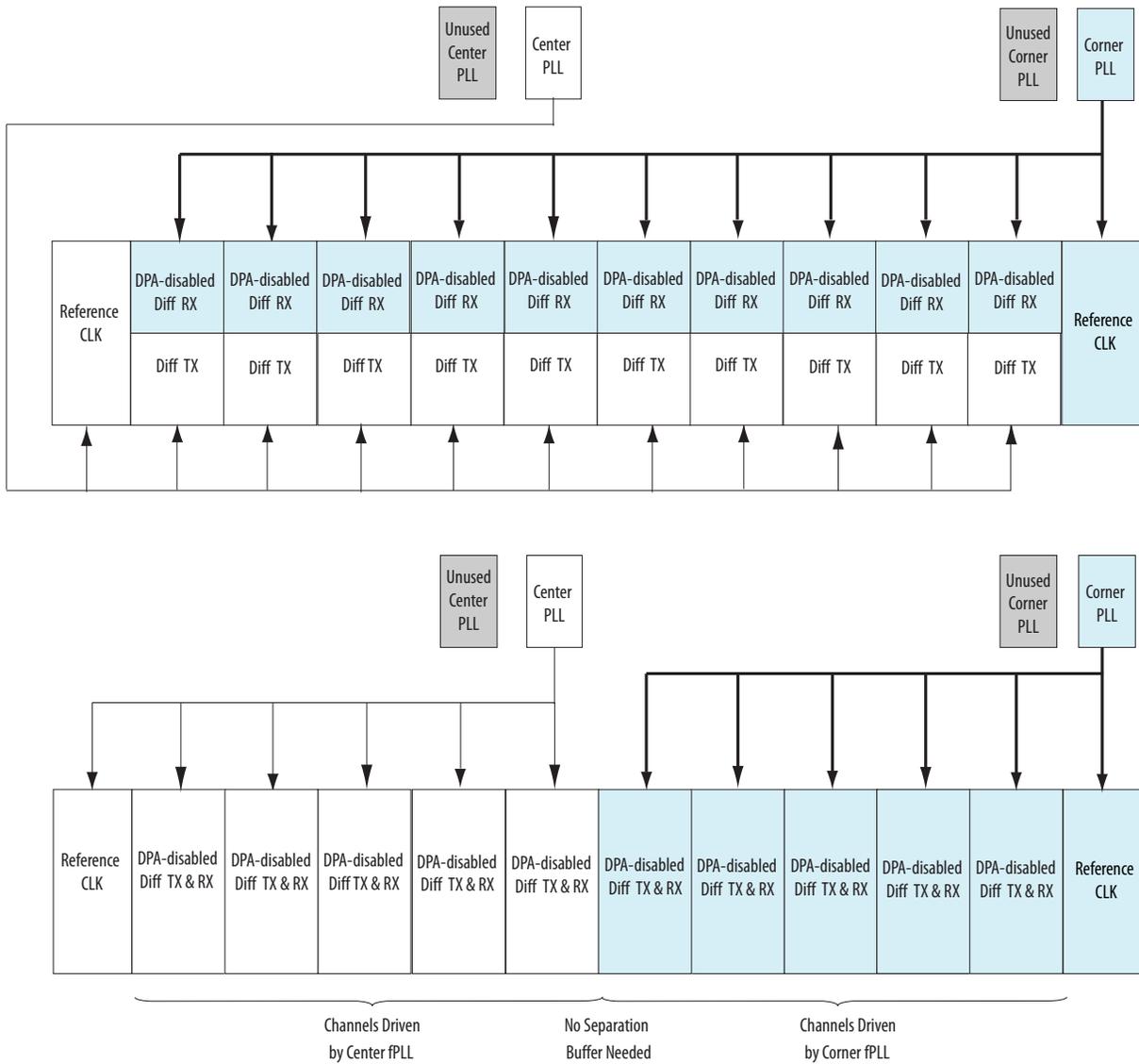
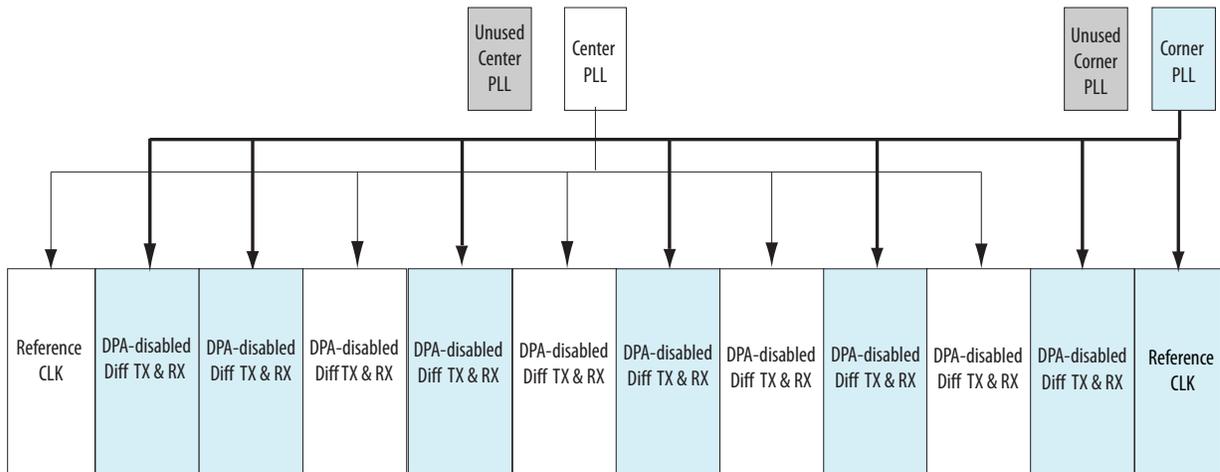


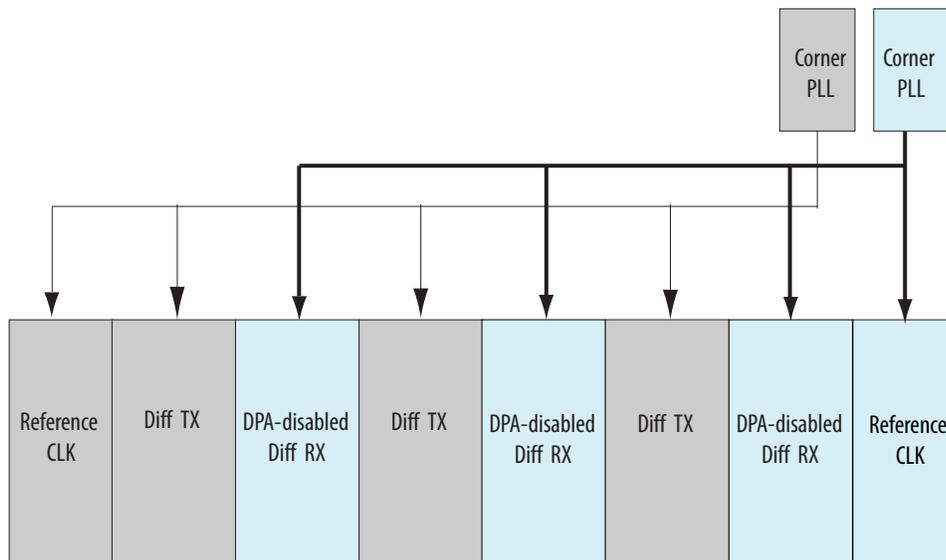
Figure 6-14: Invalid Placement of DPA-disabled Differential I/Os Due to Interleaving of Channels Driven by the Corner and Center PLLs



Using Both Corner PLLs

You can use both corner PLLs to drive DPA-disabled channels simultaneously. You can use a corner PLL to drive all the transmitter channels and the other corner PLL to drive all the DPA-disabled receiver channels in the same I/O bank. Both corner PLLs can drive duplex channels in the same I/O bank if the channels that are driven by each PLL are not interleaved. You do not require separation between the groups of channels that are driven by both corner PLLs.

Figure 6-15: Right Corner PLLs Driving LVDS Differential I/Os in the Same Bank



Differential Transmitter in Stratix V Devices

The Stratix V transmitter contains dedicated circuitry to support high-speed differential signaling. The differential transmitter buffers support the following features:

- LVDS signaling that can drive out LVDS, mini-LVDS, and RSDS signals
- Programmable V_{OD} and programmable pre-emphasis

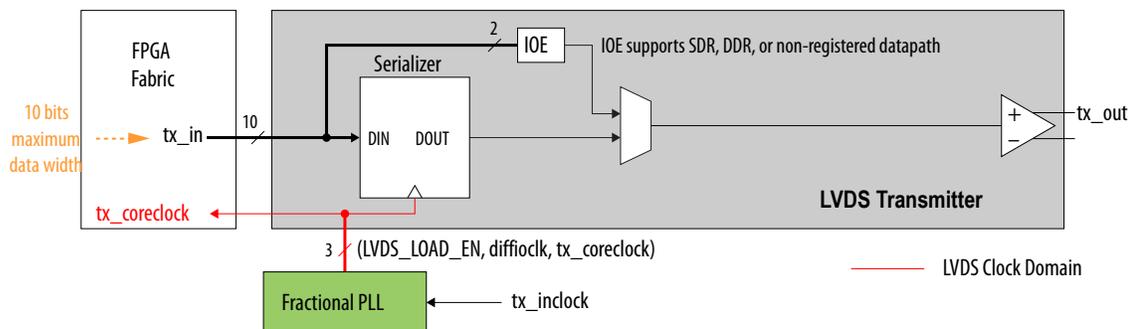
Transmitter Blocks

The dedicated circuitry consists of a true differential buffer, a serializer, and fractional PLLs that you can share between the transmitter and receiver. 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 that are clocked by the fractional PLL before sending the data to the differential buffer. The MSB of the parallel data is transmitted first.

Note: To drive the LVDS channels, you must use the PLLs in integer PLL mode.

The following figure shows a block diagram of the transmitter. In SDR and DDR modes, the data width is 1 and 2 bits, respectively.

Figure 6-16: LVDS Transmitter



Related Information

Guideline: [Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

Transmitter Clocking

The fractional PLL generates the load enable ($LVDS_LOAD_EN$) signal and the $diffioclck$ signal (the clock running at serial data rate) that 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 Intel Quartus Prime software. The load enable signal is derived from the serialization factor setting.

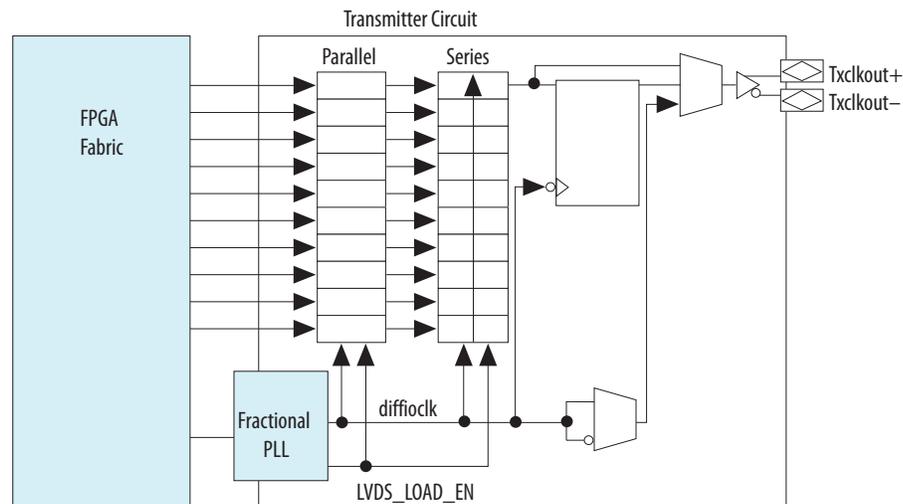
You can configure any Stratix V transmitter data channel to generate a source-synchronous transmitter clock output. This flexibility allows the placement of the output clock near the data outputs to simplify board layout and reduce clock-to-data skew.

Different applications often require specific clock-to-data alignments or specific data-rate-to-clock-rate factors. You can specify these settings statically in the Intel Quartus Prime IP Catalog:

- The transmitter can output a clock signal at the same rate as the data—with a maximum output clock frequency that each speed grade of the device supports.
- You can divide the output clock 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 using internal PLL option of the ALTLVDS IP core. The fractional PLLs provide additional support for other phase shifts in 45° increments.

The following figure shows the transmitter in clock output mode. In clock output mode, you can use an LVDS channel as a clock output channel.

Figure 6-17: Transmitter in Clock Output Mode



Related Information

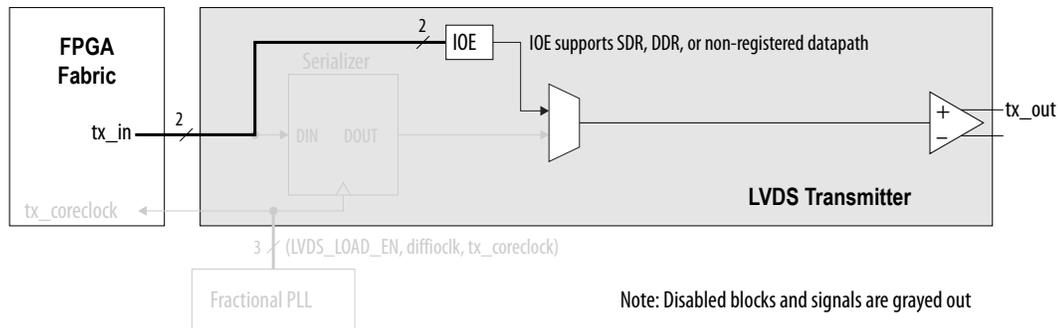
Guideline: [Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

Serializer Bypass for DDR and SDR Operations

You can bypass the 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 6-18: Serializer Bypass

This figure shows the serializer bypass path. In DDR mode, `tx_inclock` clocks the IOE register. In SDR mode, data is passed directly through the IOE. In SDR and DDR modes, the data width to the IOE is 1 and 2 bits, respectively.

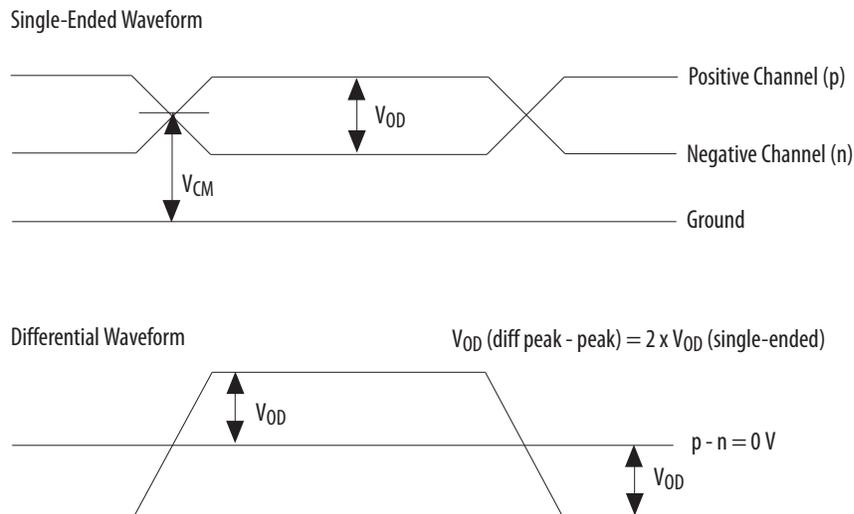


Programmable Differential Output Voltage

The programmable V_{OD} settings allow you to adjust the output eye opening to optimize the trace length and power consumption. A higher V_{OD} swing improves voltage margins at the receiver end, and a smaller V_{OD} swing reduces power consumption. You can statically adjust the V_{OD} of the differential signal by changing the V_{OD} settings in the Intel Quartus Prime software Assignment Editor.

Figure 6-19: Differential V_{OD}

This figure shows the V_{OD} of the differential LVDS output.

**Table 6-10: Intel Quartus Prime Software Assignment Editor—Programmable V_{OD}**

This table lists the assignment name for programmable V_{OD} and its possible values in the Intel Quartus Prime software Assignment Editor.

Field	Assignment
To	tx_out

Field	Assignment
Assignment name	Programmable Differential Output Voltage (V_{OD})
Allowed values	0 (low), 1 (medium low), 2 (medium high), 3 (high). Default is 1.

Related Information

[Programmable IOE Features in Stratix V Devices](#) on page 5-18

Programmable Pre-Emphasis

The V_{OD} setting and the output impedance of the driver set the output current limit of a high-speed transmission signal. At a high frequency, the slew rate may not be fast enough to reach the full V_{OD} level 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.

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. The overshoot introduced by the extra current happens only during a change of state switching to increase the output slew rate and does not ring, unlike the overshoot caused by signal reflection. The amount of pre-emphasis required depends on the attenuation of the high-frequency component along the transmission line.

Figure 6-20: Programmable Pre-Emphasis

This figure shows the LVDS output with pre-emphasis.

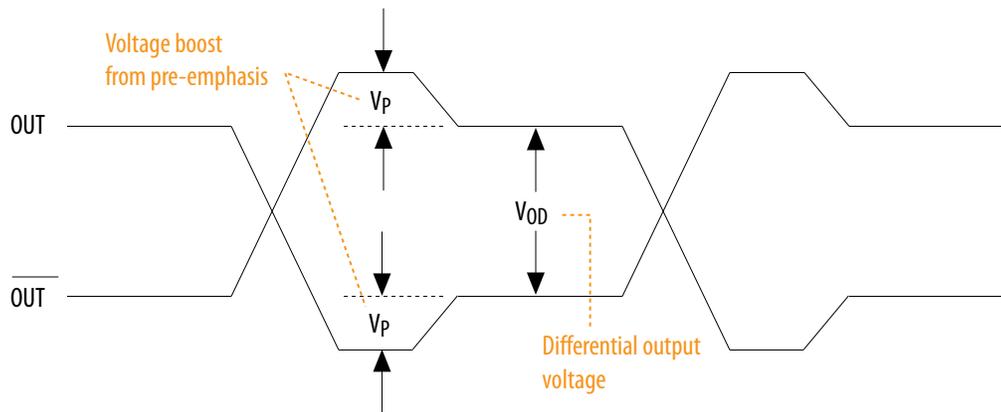


Table 6-11: Intel Quartus Prime Software Assignment Editor—Programmable Pre-Emphasis

This table lists the assignment name for programmable pre-emphasis and its possible values in the Intel Quartus Prime software Assignment Editor.

Field	Assignment
To	tx_out
Assignment name	Programmable Pre-emphasis
Allowed values	0 (disabled), 1 (enabled). Default is 1.

Related Information

[Programmable IOE Features in Stratix V Devices](#) on page 5-18

Differential Receiver in Stratix V Devices

The receiver has a differential buffer and fractional PLLs that you can share among 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. You can statically set the I/O standard of the receiver pins to LVDS, mini-LVDS, or RSDS in the Intel Quartus Prime software Assignment Editor.

Note: To drive the LVDS channels, you must use the PLLs in integer PLL mode.

Related Information

[Guideline: Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

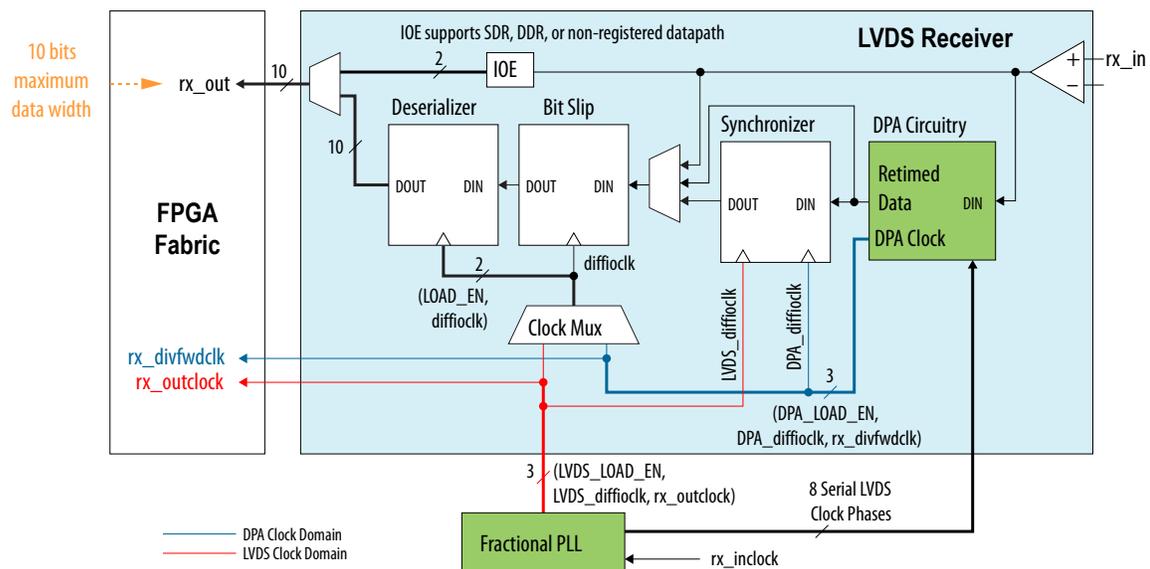
Receiver Blocks in Stratix V Devices

The Stratix V differential receiver has the following hardware blocks:

- DPA block
- Synchronizer
- Data realignment block (bit slip)
- Deserializer

The following figure shows the hardware blocks of the receiver. In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively. The deserializer includes shift registers and parallel load registers, and sends a maximum of 10 bits to the internal logic.

Figure 6-21: Receiver Block Diagram

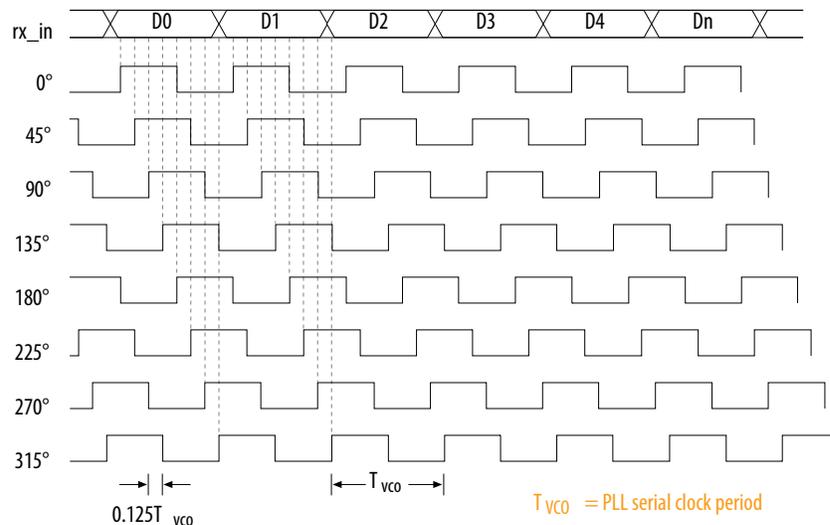


DPA Block

The DPA block takes in high-speed serial data from the differential input buffer and selects one of the eight phases that the fractional PLLs generate 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.

The following figure shows the possible phase relationships between the DPA clocks and the incoming serial data.

Figure 6-22: DPA Clock Phase to Serial Data Timing Relationship



The DPA block continuously monitors the phase of the incoming serial data and selects a new clock phase if it is required. 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, the 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. You must retrain the DPA circuitry after reset.

Note: The DPA block is bypassed in non-DPA mode.

Related Information

Guideline: [Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

Synchronizer

The synchronizer is a 1 bit wide and 6 bit deep FIFO buffer that compensates for the phase difference between `DPA_diffioclk`—the optimal clock that the DPA block selects—and the `LVDS_diffioclk` that the fractional PLLs produce. 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 that the received data is corrupted.

Note: The synchronizer circuit is bypassed in non-DPA and soft-CDR mode.

Related Information

Guideline: [Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

Data Realignment Block (Bit Slip)

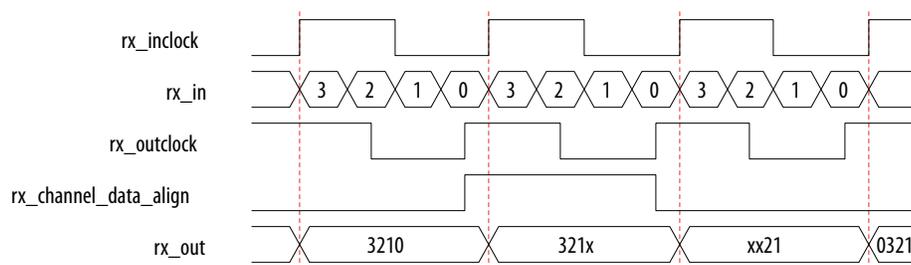
Skew in the transmitted data along with skew added by the link causes channel-to-channel skew on the received serial data streams. If you enable the DPA, the received data is captured with different clock phases on each channel. This difference may cause misalignment of the received data from channel 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 following items:

- 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.
- The signal is an edge-triggered signal.
- The valid data is available two parallel clock cycles after the rising edge of `RX_CHANNEL_DATA_ALIGN`.

Figure 6-23: Data Realignment Timing

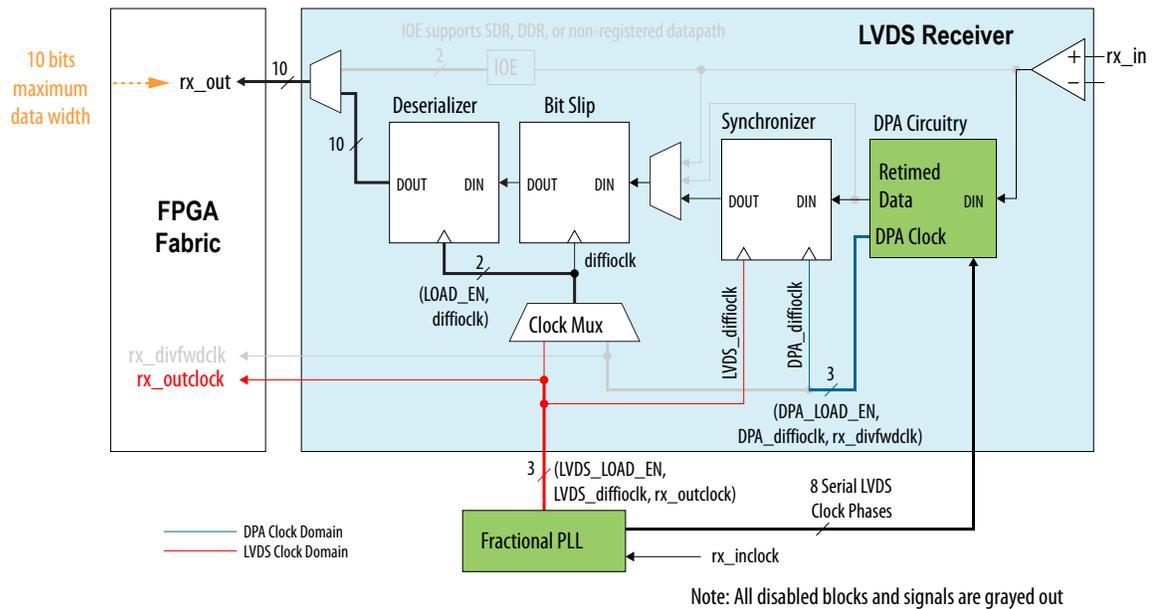
This figure shows receiver output (`RX_OUT`) after one bit slip pulse with the deserialization factor set to 4.



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. Set the programmable bit rollover point 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

Figure 6-27: Receiver Datapath in DPA Mode

In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.



Related Information

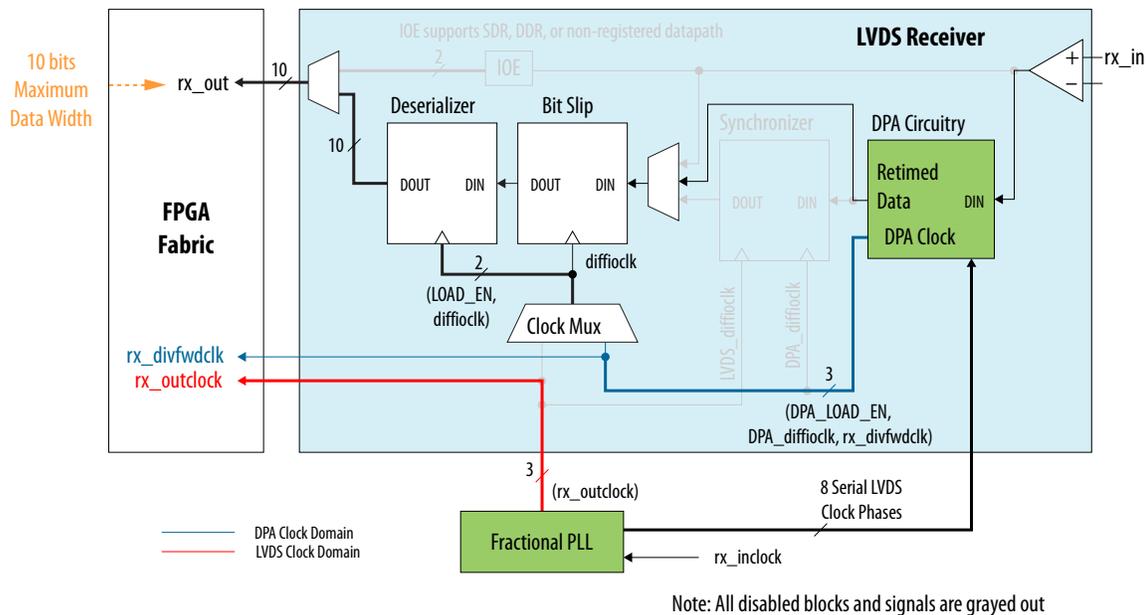
- [Receiver Blocks in Stratix V Devices](#) on page 6-24
Lists and describes the receiver hardware blocks.
- [Guideline: Use PLLs in Integer PLL Mode for LVDS](#) on page 6-8

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.

The following figure shows the soft-CDR mode datapath. In SDR and DDR modes, the data width from the IOE is 1 and 2 bits, respectively.

Figure 6-28: Receiver Datapath in Soft-CDR Mode



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.

If you use the soft-CDR mode, do not assert the `rx_reset` port after the DPA has trained. The DPA continuously chooses new phase taps from the PLL to track parts per million (PPM) differences between the reference clock and incoming data.

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.

Related Information

- [Periphery Clock Networks](#) on page 4-4
Provides more information about PCLK networks.
- [Clock Networks and PLLs in Stratix V Devices](#)
Provides more information about PCLK networks.

Receiver Clocking for Stratix V Devices

The fractional PLL receives the external clock input and generates different phases of the same clock. The DPA block automatically 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 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.

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.

Note: Only the non-DPA mode requires manual skew adjustment.

Related Information

Guideline: Use PLLs in Integer PLL Mode for LVDS on page 6-8

Differential I/O Termination for Stratix V Devices

The Stratix V devices provide 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 Intel Quartus Prime software Assignment Editor.

All I/O pins and dedicated clock input pins support on-chip differential termination, R_D OCT.

Figure 6-29: On-Chip Differential I/O Termination

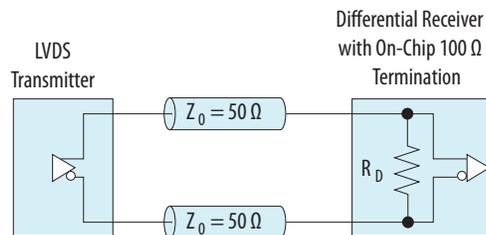


Table 6-12: Intel Quartus Prime Software Assignment Editor—On-Chip Differential Termination

This table lists the assignment name for on-chip differential termination in the Intel Quartus Prime software Assignment Editor.

Field	Assignment
To	rx_in
Assignment name	Input Termination

Field	Assignment
Value	Differential

Source-Synchronous Timing Budget

The topics in this section describe the timing budget, waveforms, and specifications for source-synchronous signaling in the Stratix V device family.

The LVDS I/O standard enables high-speed transmission of data, resulting in better overall system performance. To take advantage of fast system performance, you must analyze the timing for these high-speed signals. Timing analysis for the differential block is different from traditional synchronous timing analysis techniques.

The basis of the source synchronous timing analysis is the skew between the data and the clock signals instead of the clock-to-output setup times. 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.

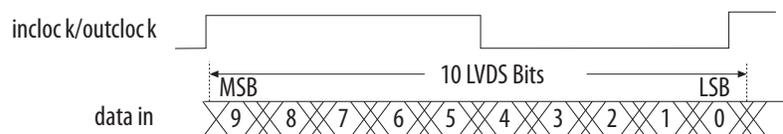
This section defines the source-synchronous differential data orientation timing parameters, the timing budget definitions for the Stratix V device family, and how to use these timing parameters to determine the maximum performance of a design.

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 6-30: Bit Orientation in the Intel Quartus Prime Software

This figure shows the data bit orientation of the x10 mode.



Differential I/O Bit Position

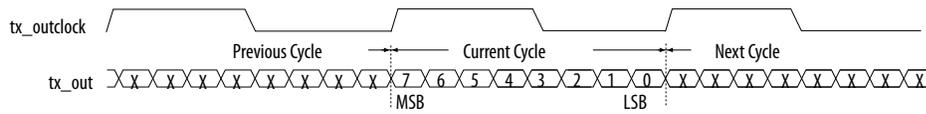
Data synchronization is necessary for successful data transmission at high frequencies.

The following figure shows the data bit orientation for a channel operation and is based on the following conditions:

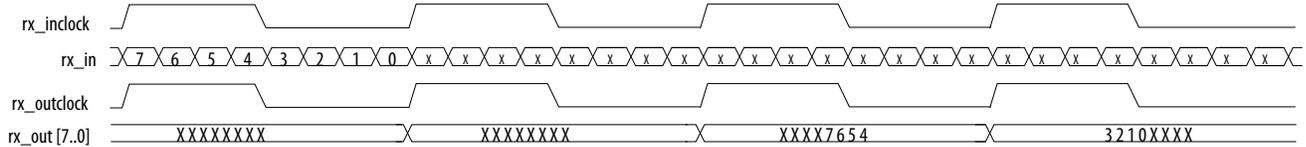
- The serialization factor is equal to the clock multiplication factor.
- The phase alignment uses edge alignment.
- The operation is implemented in hard SERDES.

Figure 6-31: Bit-Order and Word Boundary for One Differential Channel

Transmitter Channel Operation (x8 Mode)



Receiver Channel Operation (x8 Mode)



Note: These waveforms are only functional waveforms and do not convey timing information

For other serialization factors, use the Intel Quartus Prime software tools to find the bit position within the word.

Differential Bit Naming Conventions

The following table 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 6-13: Differential Bit Naming

This table 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.

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

Receiver Channel Data Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
14	111	104
15	119	112
16	127	120
17	135	128
18	143	136

Transmitter Channel-to-Channel Skew

The receiver skew margin calculation uses the transmitter channel-to-channel skew (TCCS)—an important parameter based on the Stratix V transmitter in a source-synchronous differential interface:

- TCCS is the difference between the fastest and slowest data output transitions, including the T_{CO} variation and clock skew.
- For LVDS transmitters, the Timing Analyzer provides the TCCS value in the TCCS report (`report_TCCS`) in the Intel Quartus Prime compilation report, which shows TCCS values for serial output ports.
- You can also get the TCCS value from the device datasheet.

Note: For the Stratix V devices, perform PCB trace compensation to adjust the trace length of each LVDS channel to improve channel-to-channel skew when interfacing with non-DPA receivers at data rate above 840 Mbps.

The Intel Quartus Prime software Fitter Report panel reports the amount of delay you must add to each trace for the Stratix V device. You can use the recommended trace delay numbers published under the LVDS Transmitter/Receiver Package Skew Compensation panel and manually compensate the skew on the PCB board trace to reduce channel-to-channel skew, thus meeting the timing budget between LVDS channels.

Related Information

- [Stratix V Device Datasheet](#)
- [Stratix V Device Datasheet](#)
- [LVDS SERDES Transmitter/Receiver IP Cores User Guide](#)
Provides more information about the LVDS Transmitter/Receiver Package Skew Compensation report panel.

Receiver Skew Margin for Non-DPA Mode

Different modes of LVDS receivers use different specifications, which can help in deciding the ability to sample the received serial data correctly:

- In DPA mode, use DPA jitter tolerance instead of the receiver skew margin (RSKM).
- In non-DPA mode, use RSKM, TCCS, and sampling window (SW) specifications for high-speed source-synchronous differential signals in the receiver data path.

The following equation expresses the relationship between RSKM, TCCS, and SW.

Figure 6-32: RSKM Equation

$$RSKM = \frac{TUI - SW - TCCS}{2}$$

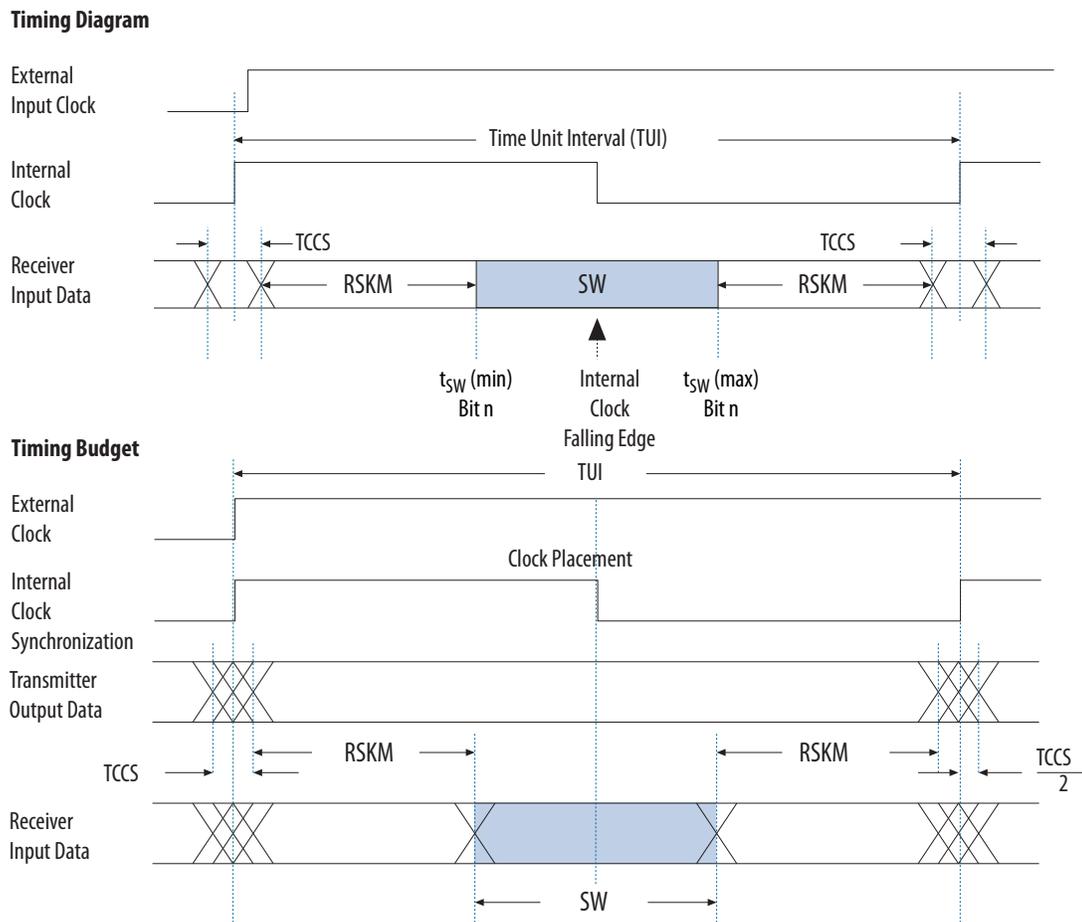
Conventions used for the equation:

- 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.

You must calculate the RSKM value to decide whether the LVDS receiver can sample the data properly or not, given the 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 sample the data properly.

The following figure shows the relationship between the RSKM, TCCS, and the SW of the receiver.

Figure 6-33: Differential High-Speed Timing Diagram and Timing Budget for Non-DPA Mode



For LVDS receivers, the Intel Quartus Prime software provides an RSKM report showing the SW, TUI, and RSKM values for non-DPA LVDS mode:

- You can generate the RSKM report by executing the `report_RSKM` command in the Timing Analyzer. You can find the RSKM report in the Intel Quartus Prime compilation report in the Timing Analyzer section.
- To obtain the RSKM value, assign the input delay to the LVDS receiver through the constraints menu of the Timing Analyzer. The input delay is determined according to the data arrival time at the LVDS receiver port, with respect to the reference clock.
- If you set the input delay in the settings parameters for the **Set Input Delay** option, set the clock name to the clock that reference the source synchronous clock that feeds the LVDS receiver.
- If you do not set any input delay in the Timing Analyzer, the receiver channel-to-channel skew defaults to zero.
- You can also directly set the input delay in a Synopsys Design Constraint file (`.sdc`) using the `set_input_delay` command.

Example 6-1: RSKM Calculation Example

This example shows the RSKM calculation for Stratix V devices at 1 Gbps data rate with a 200 ps board channel-to-channel skew.

- 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
- $RSKM = (TUI - SW - RCCS) / 2 = (1000 \text{ ps} - 300 \text{ ps} - 300 \text{ ps}) / 2 = 200 \text{ ps}$

Because the RSKM is greater than 0 ps, the receiver non-DPA mode will work correctly.

Related Information

- [LVDS SERDES Transmitter/Receiver IP Cores User Guide](#)
Provides more information about the RSKM equation and calculation.
- [The Intel Quartus Prime Timing Analyzer chapter, Intel Quartus Prime Standard Edition Handbook Volume 3 Verification](#)
Provides more information about `.sdc` commands and the Timing Analyzer.

Assigning Input Delay to LVDS Receiver Using Timing Analyzer

To obtain the RSKM value, assign an appropriate input delay to the LVDS receiver from the Timing Analyzer constraints menu.

1. On the menu in the Timing Analyzer, select **Constraints > Set Input Delay**.
2. In the **Set Input Delay** window, select the desired clock using the pull-down menu. The clock name must reference the source synchronous clock that feeds the LVDS receiver.
3. Click the **Browse** button (next to the **Targets** field).
4. In the **Name Finder** window, click **List** to view a list of all available ports. Select the LVDS receiver serial input ports according to the input delay you set, and click **OK**.
5. In the **Set Input Delay** window, set the appropriate values in the **Input delay** options and **Delay value** fields.
6. Click **Run** to incorporate these values in the Timing Analyzer.
7. Repeat from [step 1](#) to assign the appropriate delay for all the LVDS receiver input ports. If you have already assigned Input Delay and you need to add more delay to that input port, turn on the **Add Delay** option.

High-Speed Differential I/O Interfaces and DPA in Stratix V Devices Revision History

Date	Version	Changes
December 2017	2017.12.15	<ul style="list-style-type: none"> Added a note to <i>Guideline: Use High-Speed Clock from PLL to Clock LVDS SERDES Only</i> topic to clarify that spread-spectrum input clock is not supported in LVDS. Updated for latest Intel branding standards.
December 2015	2015.12.21	Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i> .
June 2015	2015.06.12	<ul style="list-style-type: none"> Changed figure title "Corner PLLs Driving DPA-enabled Differential I/Os" to "Invalid Usage of Corner PLLs Driving DPA-enabled Differential I/Os". Added LVDS and DPA Clock Network figure in <i>Guideline: Using DPA-Enabled Differential Channels</i>. Updated all figures in <i>Guideline: Using DPA-Enabled Differential Channels</i>. Updated guidelines for using both corner PLLs in Stratix V Devices. Updated figures in <i>Guideline: Using DPA-Disabled LVDS Differential Channels</i>.
January 2015	2015.01.23	<ul style="list-style-type: none"> Removed statement on explanation related to rx_synclock for figure "LVDS Interface with the Altera_PLL Megafunction (With Soft-CDR Mode)". Updated figure LVDS Interface with the Altera_PLL Megafunction (With Soft-CDR Mode) and figure Receiver Datapath in Soft-CDR Mode. Added a note to leave rx_enable and rx_inclock to be unconnected for figure LVDS Interface with the Altera_PLL Megafunction (With Soft-CDR Mode). Updated timing diagram for Phase Relationship for External PLL Interface Signals to reflect the correct phase shift and frequency for outclk2.

Date	Version	Changes
January 2014	2014.01.10	<ul style="list-style-type: none">• Updated the statement about setting the phase of the clock in relation to data in the topic about transmitter clocking.• Updated the figure that shows the phase relationship for the external PLL interface signals.• Clarified that "one row of separation" between two groups of DPA-enabled channels means a separation of one differential channel.• Clarified that "internal PLL option" refers to the option in the ATLVDS megafunction.• Updated the topic about emulated LVDS buffers to clarify that you can use unutilized true LVDS input channels (instead "buffers") as emulated LVDS output buffers.
June 2013	2013.06.21	Updated the figure about data realignment timing to correct the data pattern after a bit slip.
May 2013	2013.05.06	<ul style="list-style-type: none">• Moved all links to the Related Information section of respective topics for easy reference.• Added link to the known document issues in the Knowledge Base.• Removed all references to column and row I/Os. Stratix V devices have I/O banks on the top and bottom only.• Changed the color of the transceiver blocks in the high-speed differential I/O location diagram for clarity.• Updated the pin placement guidelines section to add figures and new topic about using DPA-disabled differential channels.• Added a topic about emulated LVDS buffers.• Edited the topic about true LVDS buffers.• Added a topic that lists the SERDES I/O standards support and the respective Quartus II assignment values.• Corrected the outclk2 waveform in Figure 6-4 to show -18° phase shift (as labeled).• Clarified that the programmable V_{OD} assignment value of "0" is also applicable for mini-LVDS.• Updated the data realignment timing figure to improve clarity.• Updated the receiver data realignment rollover figure to improve clarity.

Date	Version	Changes
December 2012	2012.12.28	<ul style="list-style-type: none"> Reorganized content and updated template. Added Altera_PLL settings for external PLL usage in DPA and non-DPA modes. Moved the PLL and clocking section into design guideline topics. Updated external PLL clocking examples without DPA and soft-CDR. Altera_PLL now supports entering negative phase shift. Added external PLL clocking example and settings for DPA and soft-CDR mode. Updated the LVDS channel tables to list the number of channels per side for each device package instead of just for the largest package. Removed the “LVDS Direct Loopback Mode” section.
June 2012	1.4	<ul style="list-style-type: none"> Added Table 6–2. Updated Table 6–1, Table 6–3, Table 6–4, and Table 6–5. Updated Figure 6–21. Updated “Non-DPA Mode”, “Soft-CDR Mode”, and “PLLs and Stratix V Clocking” sections.
November 2011	1.3	<ul style="list-style-type: none"> Updated Table 6–2. Updated Example 6–1. Updated “LVDS Direct Loopback Mode” and “LVDS Interface with the Use External PLL Option Enabled” sections.
May 2011	1.2	<ul style="list-style-type: none"> 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 6–23. 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.

External Memory Interfaces in Stratix V Devices

7

2020.09.10

SV51008



Subscribe



Send Feedback

The Stratix V devices provide an efficient architecture that allows you to fit wide external memory interfaces to support a high level of system bandwidth within the small modular I/O bank structure. The I/Os are designed to provide high-performance support for existing and emerging external memory standards.

Table 7-1: Supported External Memory Standards in Stratix V Devices

Memory Standard	Soft Memory Controller
DDR3 SDRAM	Half rate and quarter rate
DDR2 SDRAM	Full rate and half rate
RLDRAM 3	Half rate and quarter rate
RLDRAM II	Full rate and half rate
QDR II+ SRAM	Full rate and half rate
QDR II SRAM	Full rate and half rate

Related Information

- [Stratix V Device Handbook: Known Issues](#)
Lists the planned updates to the *Stratix V Device Handbook* chapters.
- [External Memory Interface Spec Estimator](#)
For the latest information and to estimate the external memory system performance specification, use Altera's External Memory Interface Spec Estimator tool.
- [External Memory Interfaces Handbook Volume 1, 2, and 3.](#)
Provides more information about the memory types supported, board design guidelines, timing analysis, simulation, and debugging information.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

External Memory Performance

Table 7-2: External Memory Interface Performance in Stratix V Devices

Interface	Voltage (V)	Soft Controller (MHz)
DDR3 SDRAM	1.5	933
	1.35	800
DDR2 SDRAM	1.8	400
RLDRAM 3	1.2	800
RLDRAM II	1.8	533
	1.5	533
QDR II+ SRAM	1.8	550
	1.5	550
QDR II SRAM	1.8	350
	1.5	350

Related Information

[External Memory Interface Spec Estimator](#)

For the latest information and to estimate the external memory system performance specification, use Altera's External Memory Interface Spec Estimator tool.

Memory Interface Pin Support in Stratix V Devices

In the Stratix V devices, the memory interface circuitry is available in every I/O bank that does not support transceivers. The devices offer differential input buffers for differential read-data strobe and clock operations.

Stratix V devices also provide an independent DQS logic block for each CQn pin for complementary read-data strobe and clock operations

The memory clock pins are generated with double data rate input/output (DDRIO) registers.

Related Information

[Planning Pin and FPGA Resources chapter, External Memory Interface Handbook](#)

Provides more information about which pins to use for memory clock pins and pin location requirements.

Guideline: Using DQ/DQS Pins

The following list provides guidelines on using the DQ/DQS pins:

- The devices support DQ and DQS signals with DQ bus modes of x4, x8/x9, x16/x18, or x32/x36.
- You can use the DQSn or CQn pins that are not used for clocking as DQ pins.
- If you do not use the DQ/DQS pins for memory interfacing, you can use these pins as user I/Os.
- Some pins have multiple functions such as RZQ or DQ. If you need extra RZQ pins, you can use the DQ/DQNs pins in some of the x4 groups as RZQ pins instead.
- You cannot use a x4 DQ/DQS group for memory interfaces if any of its 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.

Note: For the x8, x16/x18, or x32/x36 DQ/DQS groups whose members are used as RZQ pins, Altera recommends that you assign the DQ and DQS pins manually. Otherwise, the Intel Quartus Prime software might not be able to place the DQ and DQS pins, resulting in a “no-fit” error.

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.

Note: 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.

Reading the Pin Table

For the maximum number of DQ pins and the exact number per group for a particular Stratix V device, refer to the pin table in the Stratix V page of the Altera website. In the pin tables, the DQS and DQSn pins denote the differential data strobe/clock pin pairs, while the CQ and CQn pins denote the complementary echo clock signals. The pin table lists the parity, DM, BWSn, NWSn, ECC, and QVLD pins as DQ pins.

In the Stratix V pin tables, DQSn and CQn pins are marked separately. 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.

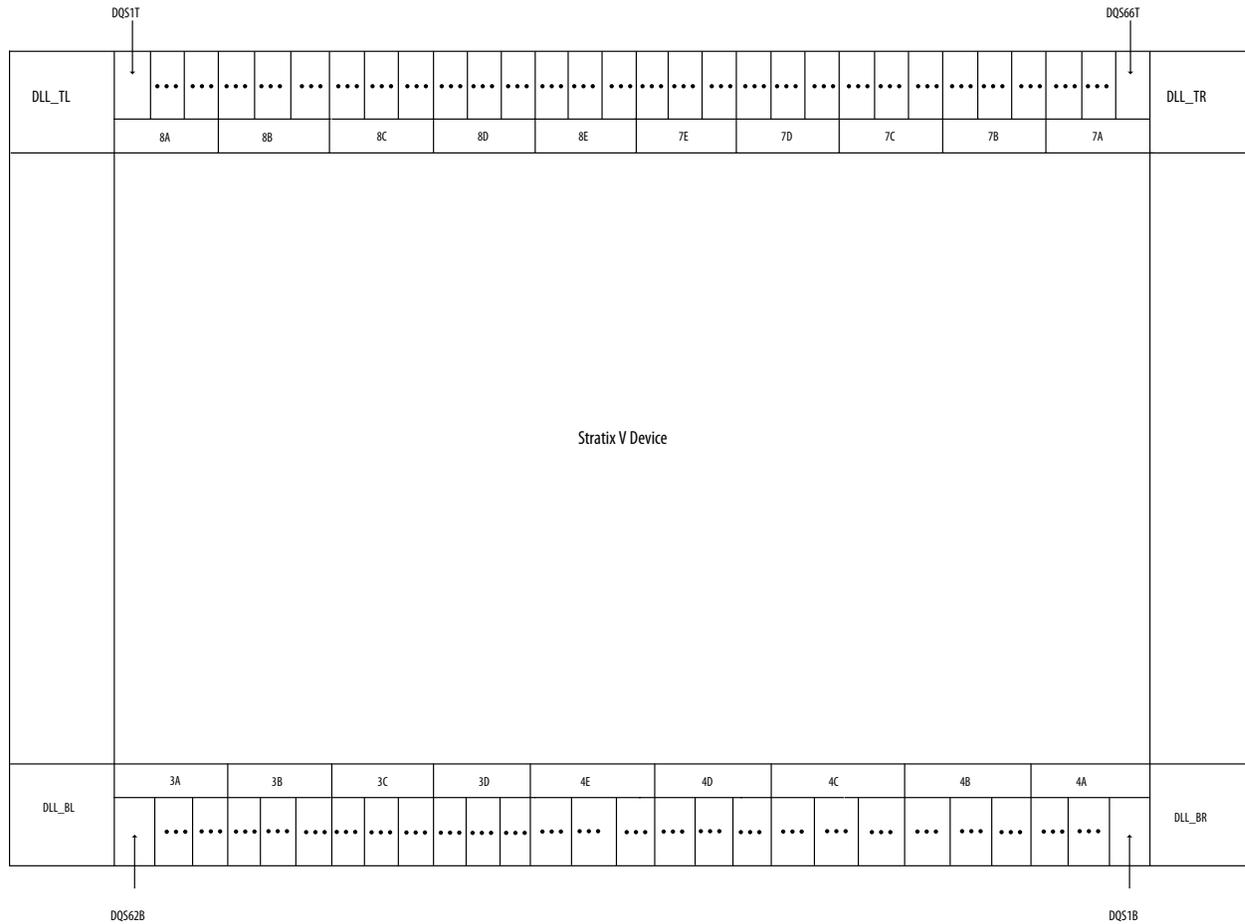
The DQS and DQSn pins are listed respectively in the Stratix V pin tables as DQS_{XY} and $DQSn_{XY}$. X indicates the DQ/DQS grouping number and Y indicates whether the group is located on the top (T), bottom (B), left (L), or right (R) side of the device. The DQ/DQS pin numbering is based on the x4 mode.

The corresponding DQ pins are marked as DQ_{XY} , where X indicates which DQS group the pins belong to and Y indicates whether the group is located on the top (T) or bottom (B) side of the device.

For example, DQS_{1T} indicates a DQS pin located on the top side of the device. The DQ pins belonging to that group are shown as DQ_{1T} in the pin table.

Figure 7-1: DQS Pins in Stratix V I/O Banks

This figure shows the DQ/DQS groups numbering in a die-top view of the device where the numbering scheme starts from the top-left corner of the device going clockwise.



DQ/DQS Bus Mode Pins for Stratix V Devices

The following table lists the pin support per DQ/DQS bus mode, including the DQS/CQ and DQSn/CQn pins. The maximum number of data pins per group listed in the table may vary according to the following conditions:

- Single-ended DQS signaling—the maximum number of DQ pins includes parity, data mask, and QVLD pins connected to the DQS bus network.
- Differential or complementary DQS signaling—the maximum number of data pins 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.
- DDR3 and DDR2 interfaces—the maximum 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.

Table 7-3: DQ/DQS Bus Mode Pins for Stratix V Devices

Mode	DQSn Support	CQn Support	Parity or Data Mask (Optional)	QVLD ⁽¹³⁾ (Optional)	Data Pins per Group		Notes
					Typical	Maximum	
x4	Yes	—	—	—	4	5	If you do not use differential DQS and the group does not have additional signals, the data mask (DM) pin is supported.
x8/x9	Yes	Yes	Yes	Yes	8 or 9	11	Two x4 DQ/DQS groups are stitched to create a x8/x9 group, so there are a total of 12 pins in this group.
x16/x18	Yes	Yes	Yes	Yes	16 or 18	23	Four x4 DQ/DQS groups are stitched to create a x16/x18 group; so there are a total of 24 pins in this group.
x32/x36	Yes	Yes	Yes	Yes	32 or 36	47	Eight x4 DQ/DQS groups are stitched to create a x32/x36 group, so there are a total of 48 pins in this group.

DQ/DQS Groups in Stratix V E

Table 7-4: Number of DQ/DQS Groups Per Side in Stratix V E Devices

Some of the x4 groups are using RZQ pins. If you use the Stratix V calibrated OCT feature, you cannot use these groups.

Member Code	Package	Side	x4	x8/x9	x16/x18	x32/x36
E9	1517-pin FineLine BGA	Top	58	29	14	6
		Bottom	58	29	14	6
	1932-pin FineLine BGA	Top	70	35	16	6
		Bottom	70	35	16	6
EB	1517-pin FineLine BGA	Top	58	29	14	6
		Bottom	58	29	14	6
	1932-pin FineLine BGA	Top	70	35	16	6
		Bottom	70	35	16	6

⁽¹³⁾ The QVLD pin is not used in the UniPHY IP core.

DQ/DQS Groups in Stratix V GX

Table 7-5: Number of DQ/DQS Groups Per Side in Stratix V GX Devices

Some of the x4 groups are using RZQ pins. If you use the Stratix V calibrated OCT feature, you cannot use these groups.

Member Code	Package	Side	x4	x8/x9	x16/x18	x32/x36
A3	780-pin FineLine BGA	Top	34	13	8	2
		Bottom	26	17	6	1
	1152-pin FineLine BGA (with 24 transceivers)	Top	42	21	10	3
		Bottom	50	25	12	4
	1152-pin FineLine BGA (with 36 transceivers)	Top	36	18	8	2
		Bottom	36	18	8	2
1517-pin FineLine BGA	Top	58	29	14	6	
	Bottom	58	29	14	6	
A4	1152-pin FineLine BGA (with 24 transceivers)	Top	42	21	10	3
		Bottom	50	25	12	4
	1152-pin FineLine BGA (with 36 transceivers)	Top	36	18	8	2
		Bottom	36	18	8	2
	1517-pin FineLine BGA	Top	58	29	14	6
		Bottom	58	29	14	6
A5	1152-pin FineLine BGA (with 24 transceivers)	Top	42	21	10	3
		Bottom	50	25	12	4
	1152-pin FineLine BGA (with 36 transceivers)	Top	36	18	8	2
		Bottom	36	18	8	2
	1517-pin FineLine BGA (with 36 transceivers)	Top	58	29	14	6
		Bottom	58	29	14	6
	1517-pin FineLine BGA (with 48 transceivers)	Top	50	25	12	5
		Bottom	50	25	12	4
	1932-pin FineLine BGA	Top	70	35	16	6
		Bottom	70	35	16	6

Member Code	Package	Side	x4	x8/x9	x16/x18	x32/x36	
A7	1152-pin FineLine BGA (with 24 transceivers)	Top	42	21	10	3	
		Bottom	50	25	12	4	
	1152-pin FineLine BGA (with 36 transceivers)	Top	36	18	8	2	
		Bottom	36	18	8	2	
	1517-pin FineLine BGA (with 36 transceivers)	Top	58	29	14	6	
		Bottom	58	29	14	6	
	1517-pin FineLine BGA (with 48 transceivers)	Top	50	25	12	5	
		Bottom	50	25	12	4	
	1932-pin FineLine BGA	Top	70	35	16	6	
		Bottom	70	35	16	6	
	A9	1517-pin FineLine BGA	Top	58	29	14	6
			Bottom	58	29	14	6
1932-pin FineLine BGA		Top	70	35	16	6	
		Bottom	70	35	16	6	
AB	1517-pin FineLine BGA	Top	58	29	14	6	
		Bottom	58	29	14	6	
	1932-pin FineLine BGA	Top	70	35	16	6	
		Bottom	70	35	16	6	
B5	1517-pin FineLine BGA	Top	36	18	8	3	
		Bottom	36	18	8	3	
	1760-pin FineLine BGA	Top	50	25	11	4	
		Bottom	50	25	11	4	
B6	1517-pin FineLine BGA	Top	36	18	8	3	
		Bottom	36	18	8	3	
	1760-pin FineLine BGA	Top	50	25	11	4	
		Bottom	50	25	11	4	
B9	1760-pin FineLine BGA	Top	50	25	11	3	
		Bottom	50	25	11	3	
BB	1760-pin FineLine BGA	Top	50	25	11	3	
		Bottom	50	25	11	3	

DQ/DQS Groups in Stratix V GS

Table 7-6: Number of DQ/DQS Groups Per Side in Stratix V GS Devices

Some of the x4 groups are using RZQ pins. If you use the Stratix V calibrated OCT feature, you cannot use these groups.

Member Code	Package	Side	x4	x8/x9	x16/x18	x32/x36
D3	780-pin FineLine BGA	Top	34	13	8	2
		Bottom	26	17	6	1
	1152-pin FineLine BGA	Top	38	19	9	2
		Bottom	34	17	8	2
D4	780-pin FineLine BGA	Top	34	13	8	2
		Bottom	26	17	6	1
	1152-pin FineLine BGA	Top	38	19	9	2
		Bottom	34	17	8	2
	1517-pin FineLine BGA	Top	58	29	14	6
		Bottom	58	29	14	6
D5	1152-pin FineLine BGA	Top	42	21	10	3
		Bottom	50	25	12	4
	1517-pin FineLine BGA	Top	58	29	14	6
		Bottom	58	29	14	6
D6	1517-pin FineLine BGA	Top	58	29	14	6
		Bottom	58	29	14	6
	1932-pin FineLine BGA	Top	70	35	16	6
		Bottom	70	35	16	6
D8	1517-pin FineLine BGA	Top	58	29	14	6
		Bottom	58	29	14	6
	1932-pin FineLine BGA	Top	70	35	16	6
		Bottom	70	35	16	6

DQ/DQS Groups in Stratix V GT

Table 7-7: Number of DQ/DQS Groups Per Side in Stratix V GT Devices

Some of the x4 groups are using RZQ pins. If you use the Stratix V calibrated OCT feature, you cannot use these groups.

Member Code	Package	Side	x4	x8/x9	x16/x18	x32/x36
C5	1517-pin FineLine BGA	Top	50	25	12	5
		Bottom	50	25	12	4
C7	1517-pin FineLine BGA	Top	50	25	12	5
		Bottom	50	25	12	4

External Memory Interface Features in Stratix V Devices

The Stratix V I/O elements (IOE) provide built-in functionality required for a rapid and robust implementation of external memory interfacing.

The following device features are available for external memory interfaces:

- DQS phase-shift circuitry
- PHY Clock (PHYCLK) networks
- DQS logic block
- Dynamic on-chip termination (OCT) control
- IOE registers
- Delay chains
- Delay-locked loops (DLLs)
- Read- and write-leveling support
- Trace mismatch compensation
- Read FIFO blocks
- Slew rate adjustment
- Programmable drive strength

UniPHY IP

The high-performance memory interface solution includes the self-calibrating UniPHY IP that is optimized to take advantage of the Stratix V I/O structure and the Intel Quartus Prime software Timing Analyzer. The UniPHY IP helps set up the physical interface (PHY) best suited for your system. This provides the total solution for the highest reliable frequency of operation across process, voltage, and temperature (PVT) variations.

The UniPHY IP instantiates a PLL to generate related clocks for the memory interface. The UniPHY IP can also dynamically choose the number of delay chains that are 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.

The UniPHY IP and the Altera memory controller IP core can run at half or quarter of the I/O interface frequency of the memory devices, allowing better timing management in high-speed memory interfaces. The Stratix V devices contain built-in circuitry in the IOE to convert data from full rate (the I/O frequency) to half rate (the controller frequency) and vice versa.

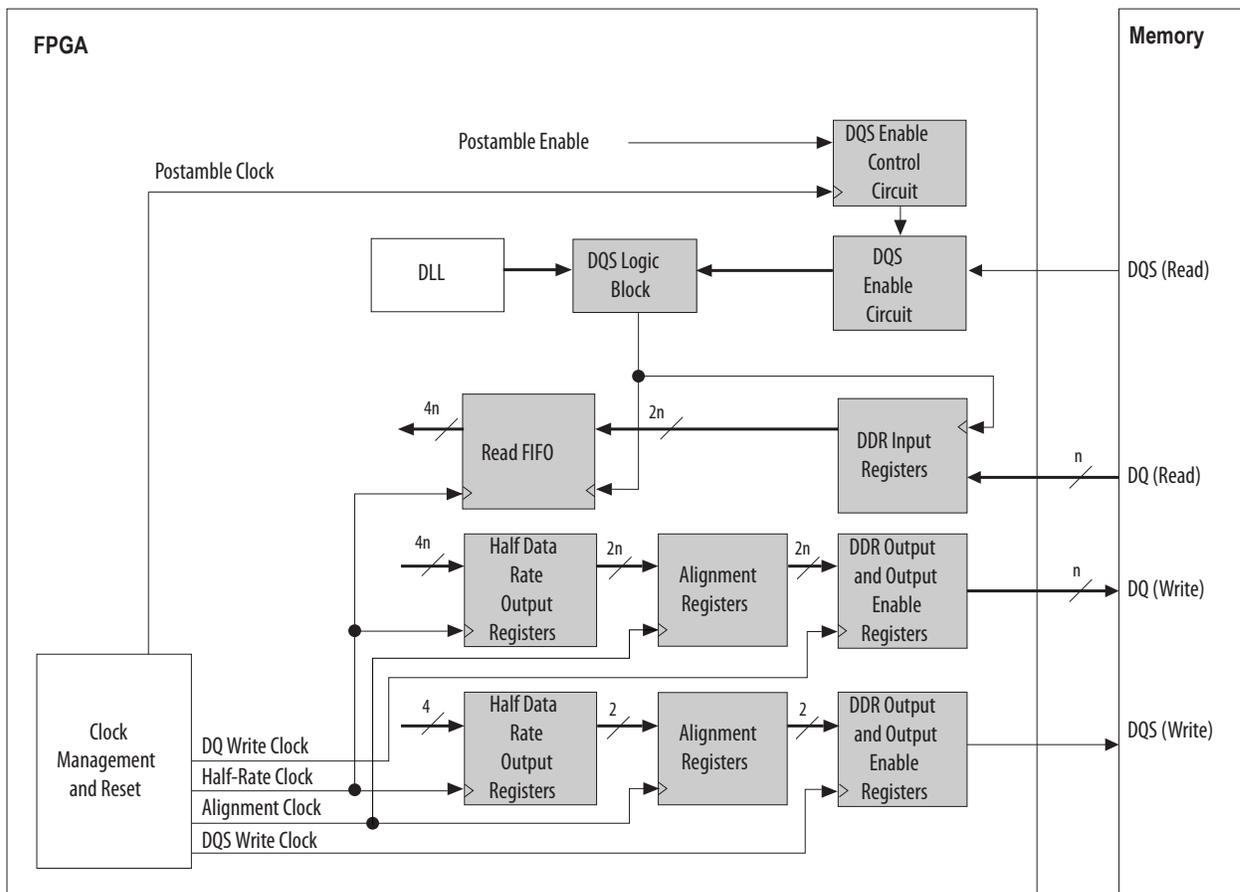
Related Information**Functional Description - UniPHY, External Memory Interface Handbook Volume 3**

Provides more information about UniPHY IP.

External Memory Interface Datapath

The following figure shows an overview of the memory interface datapath that uses the Stratix V I/O elements. In the figure, the DQ/DQS read and write signals may be bidirectional or unidirectional, depending on the memory standard. If the signal is bidirectional, it is active during read and write operations. You can bypass each register block.

Figure 7-2: External Memory Interface Datapath Overview for Stratix V Devices



Note: There are slight block differences for different memory interface standards. The shaded blocks are part of the I/O elements.

DQS Phase-Shift Circuitry

The 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. The DQS phase-shift circuitry consists of DLLs that are shared between multiple DQS pins and the phase-offset module to further fine-tune the DQS phase shift for different sides of the device.

The following figures show how the DQS phase-shift circuitry is connected to the DQS/CQ and CQn pins in the Stratix V variants.

Figure 7-3: DQS/CQ and CQn Pins and DQS Phase-Shift Circuitry in Stratix V E Devices

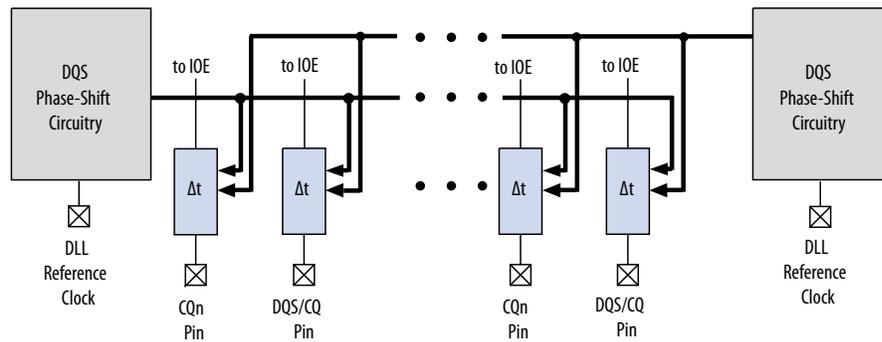
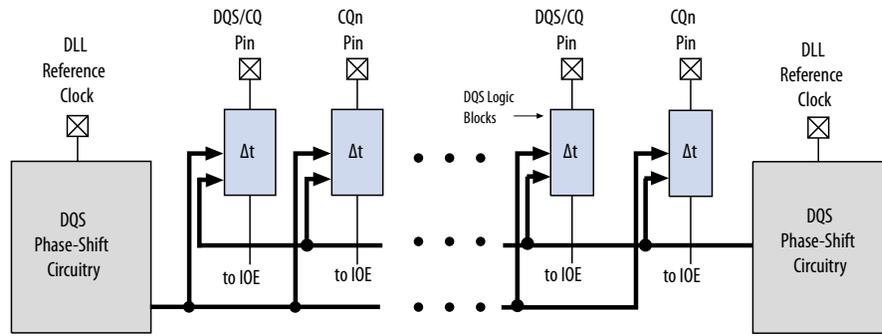
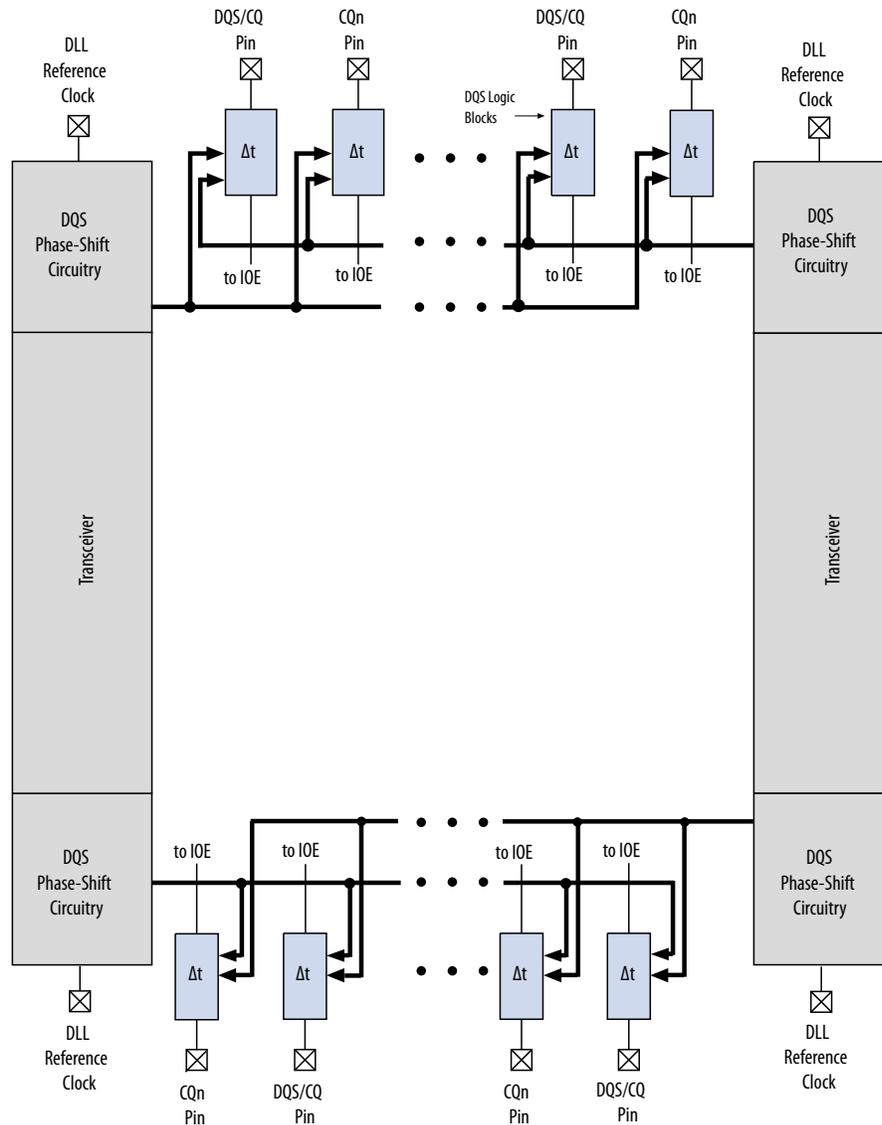


Figure 7-4: DQS/CQ and CQn Pins and DQS Phase-Shift Circuitry in Stratix V GX, GS, and GT Devices



The 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

The DQS phase-shift circuitry uses a delay-locked loop (DLL) to dynamically control the clock delay required by the DQS/CQ and CQn pin.

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 the delay to compensate for process, voltage, and temperature (PVT) variations. The DQS delay settings are gray-coded to reduce jitter if the DLL updates the settings.

There are a maximum of four DLLs, located in each corner of the Stratix V devices. You can clock each DLL using different frequencies. Each DLL can have two outputs with different phase offsets, which allows one Stratix V device to have eight different DLL phase shift settings.

You can have two different interfaces with the same frequency sharing a DLL, where the DLL controls the DQS delay settings for both interfaces.

Each I/O bank is accessible by two DLLs, giving more flexibility to create multiple frequencies and multiple-type interfaces. Each bank can use settings from one or both adjacent DLLs. For example, DQS1T can get its phase-shift settings from DLL_TR, while DQS2T can get its phase-shift settings from DLL_TL.

The reference clock for each DLL may come from the PLL output clocks or clock input pins.

Note: 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 Intel Quartus Prime software automatically changes it). Because the PLL does not use any other outputs, it does not have to compensate for any clock paths.

DLL Reference Clock Input for Stratix V Devices

Table 7-8: DLL Reference Clock Input for Stratix V E E9 and EB, and Stratix V GX A9, AB, B9, and BB Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X104_Y166	COR_X0_Y170	CLK20P	CLK16P	—
	CEN_X104_Y157	COR_X0_Y161	CLK21P	CLK17P	
			CLK22P	CLK18P	
			CLK23P	CLK19P	
DLL_TR	CEN_X104_Y166	COR_X225_Y170	—	CLK16P	CLK12P
	CEN_X104_Y157	COR_X225_Y161		CLK17P	CLK13P
				CLK18P	CLK14P
				CLK19P	CLK15P
DLL_BR	CEN_X104_Y11	COR_X225_Y10	—	CLK4P	CLK8P
	CEN_X104_Y2	COR_X225_Y1		CLK5P	CLK9P
				CLK6P	CLK10P
				CLK7P	CLK11P
DLL_BL	CEN_X104_Y11	COR_X0_Y10	CLK0P	CLK4P	—
	CEN_X104_Y2	COR_X0_Y1	CLK1P	CLK5P	
			CLK2P	CLK6P	
			CLK3P	CLK7P	

Table 7-9: DLL Reference Clock Input for Stratix V GX A3 (with 36 Transceivers) and A4, and Stratix V GS D5 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X92_Y96	COR_X0_Y100	CLK20P	CLK16P	—
	CEN_X92_Y87	COR_X0_Y91	CLK21P	CLK17P	
			CLK22P	CLK18P	
			CLK23P	CLK19P	
DLL_TR	CEN_X92_Y96	COR_X202_Y100	—	CLK16P	CLK12P
	CEN_X92_Y87	COR_X202_Y91		CLK17P	CLK13P
				CLK18P	CLK14P
				CLK19P	CLK15P
DLL_BR	CEN_X92_Y11	COR_X202_Y10	—	CLK4P	CLK8P
	CEN_X92_Y2	COR_X202_Y1		CLK5P	CLK9P
				CLK6P	CLK10P
				CLK7P	CLK11P
DLL_BL	CEN_X92_Y11	COR_X0_Y10	CLK0P	CLK4P	—
	CEN_X92_Y1	COR_X0_Y1	CLK1P	CLK5P	
			CLK2P	CLK6P	
			CLK3P	CLK7P	

Table 7-10: DLL Reference Clock Input for Stratix V GX B5 and B6 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X90_Y123	LR_X0_Y109	CLK20P	CLK16P	—
	CEN_X90_Y114	LR_X0_Y100	CLK21P	CLK17P	
			CLK22P	CLK18P	
			CLK23P	CLK19P	

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TR	CEN_X90_Y123	LR_X197_Y109	—	CLK16P	CLK12P
	CEN_X90_Y114	LR_X197_Y100		CLK17P	CLK13P
				CLK18P	CLK14P
				CLK19P	CLK15P
DLL_BR	CEN_X90_Y11	LR_X197_Y14	—	CLK4P	CLK8P
	CEN_X90_Y2	LR_X197_Y5		CLK5P	CLK9P
				CLK6P	CLK10P
				CLK7P	CLK11P
DLL_BL	CEN_X90_Y11	LR_X0_Y14	CLK0P	CLK4P	—
	CEN_X90_Y2	LR_X0_Y5	CLK1P	CLK5P	
			CLK2P	CLK6P	
			CLK3P	CLK7P	

Table 7-11: DLL Reference Clock Input for Stratix V GX A5 and A7, and Stratix V GT C5 and C7 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X98_Y118	COR_X0_Y122	CLK20P	CLK16P	—
	CEN_X98_Y109	COR_X0_Y113	CLK21P	CLK17P	
			CLK22P	CLK18P	
			CLK23P	CLK19P	
DLL_TR	CEN_X98_Y118	COR_X210_Y122	—	CLK16P	CLK12P
	CEN_X98_Y109	COR_X210_Y113		CLK17P	CLK13P
				CLK18P	CLK14P
				CLK19P	CLK15P
DLL_BR	CEN_X98_Y11	COR_X210_Y10	—	CLK4P	CLK8P
	CEN_X98_Y2	COR_X210_Y1		CLK5P	CLK9P
				CLK6P	CLK10P
				CLK7P	CLK11P

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_BL	CEN_X98_Y11	COR_X0_Y10	CLK0P	CLK4P	—
	CEN_X98_Y2	COR_X0_Y1	CLK1P	CLK5P	
			CLK2P	CLK6P	
			CLK3P	CLK7P	

Table 7-12: DLL Reference Clock Input for Stratix V GX A3 (with 24 Transceivers), and Stratix V GS D3 and D4 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X84_Y77	COR_X0_Y81	CLK20P	CLK16P	—
	CEN_X84_Y68	COR_X0_Y72	CLK21P	CLK17P	
			CLK22P	CLK18P	
			CLK23P	CLK19P	
DLL_TR	CEN_X84_Y77	COR_X185_Y81	—	CLK16P	CLK12P
	CEN_X84_Y68	COR_X185_Y72		CLK17P	CLK13P
				CLK18P	CLK14P
				CLK19P	CLK15P
DLL_BR	CEN_X84_Y11	COR_X185_Y10	—	CLK4P	CLK8P
	CEN_X84_Y2	COR_X185_Y1		CLK5P	CLK9P
				CLK6P	CLK10P
				CLK7P	CLK11P
DLL_BL	CEN_X84_Y11	COR_X0_Y10	CLK0P	CLK4P	—
	CEN_X84_Y2	COR_X0_Y1	CLK1P	CLK5P	
			CLK2P	CLK6P	
			CLK3P	CLK7P	

Table 7-13: DLL Reference Clock Input for Stratix V GS D6 and D8 Devices

DLL	PLL		CLKIN		
	Center	Corner	Left	Center	Right
DLL_TL	CEN_X96_Y141	COR_X0_Y145	CLK20P	CLK16P	—
	CEN_X96_Y132	COR_X0_Y136	CLK21P	CLK17P	
			CLK22P	CLK18P	
			CLK23P	CLK19P	
DLL_TR	CEN_X96_Y141	COR_X208_Y145	—	CLK16P	CLK12P
	CEN_X96_Y132	COR_X208_Y136		CLK17P	CLK13P
				CLK18P	CLK14P
				CLK19P	CLK15P
DLL_BR	CEN_X96_Y11	COR_X208_Y10	—	CLK4P	CLK8P
	CEN_X96_Y2	COR_X208_Y1		CLK5P	CLK9P
				CLK6P	CLK10P
				CLK7P	CLK11P
DLL_BL	CEN_X96_Y11	COR_X0_Y10	CLK0P	CLK4P	—
	CEN_X96_Y2	COR_X0_Y1	CLK1P	CLK5P	
			CLK2P	CLK6P	
			CLK3P	CLK7P	

DQS Phase-Shift

The DLL can shift the incoming DQS signals by 0°, 45°, 90°, or 135°. The shifted DQS signal is then used as the clock for the DQ IOE input registers.

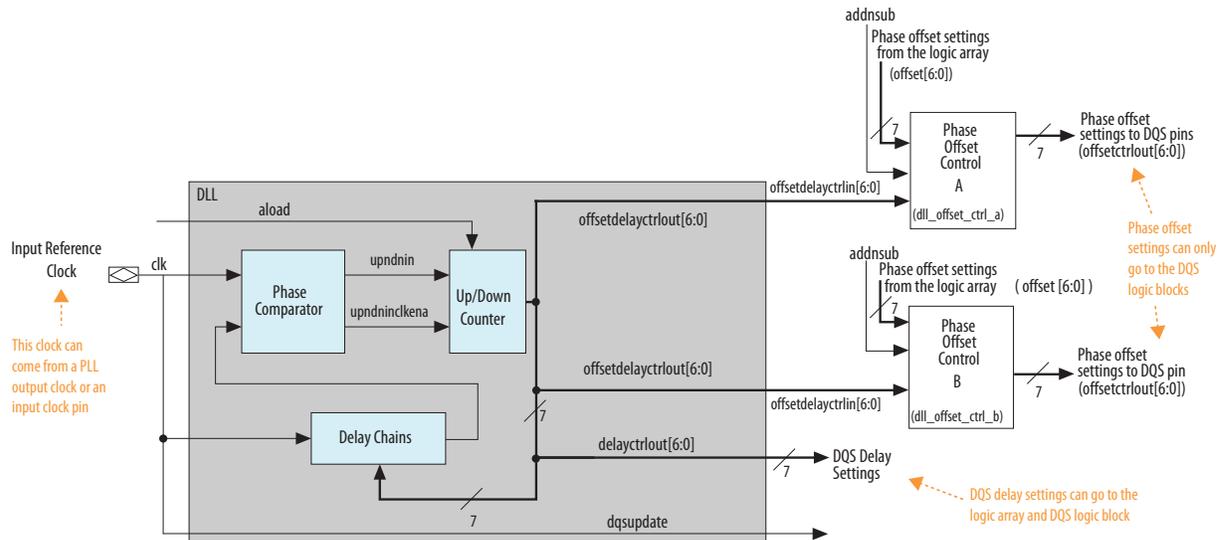
All DQS/CQ/CQn pins referenced to the same DLL, can have their input signal phase shifted by a different degree amount but all must be referenced at one particular frequency. For example, you can have a 90° phase shift on DQS1T and a 45° phase shift on DQS2T, referenced from a 300-MHz clock. However, 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 135°).

The 7-bit DQS delay settings from the DLL vary with PVT to implement the phase-shift delay. For example, with a 0° shift, the DQS/CQ signal bypasses both the DLL and DQS logic blocks. The Intel Quartus Prime software automatically sets the DQ input delay chains, so that the skew between the DQ and DQS/CQ pins 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.

Figure 7-5: Simplified Diagram of the DQS Phase-Shift Circuitry

This figure shows a simple block diagram of the DLL. All features of the DQS phase-shift circuitry are accessible from the UniPHY IP core in the Intel Quartus Prime software.



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 7-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.

Note: In the Intel Quartus Prime assignment, the phase offset control block 'A' is designated as `DLLOFFSETCTRL_CoordinateX_CoordinateY_N1` and phase offset control block 'B' is designated as `DLLOFFSETCTRL_CoordinateX_CoordinateY_N2`.

The DLL can be reset from either the logic array or a user I/O pin (if 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.

You can still use DQS phase-shift circuitry for memory interfaces running on frequencies below the minimum DLL input frequency, which is 300 MHz. The frequency of the clock feeding the DLL should be doubled when the interface frequency is between 150 MHz and 299 MHz or multiplied by four when the interface frequency is between 75 MHz and 149 MHz. Because of the changes on the DLL input clock frequency, the DQS delay chain can only shift up to 67.5° for the interface frequency between 150 MHz and 299 MHz and 33.75° for the interface frequency between 75 MHz and 149 MHz. Depending on your design, while the DQS signal might not shift exactly to the middle of the DQ valid window, the IOE is still able to capture the data accurately in low-frequency applications, where a large amount of timing margin is available.

For the frequency range of each DLL frequency mode, refer to the device datasheet.

Related Information

[Stratix V Device Datasheet](#)

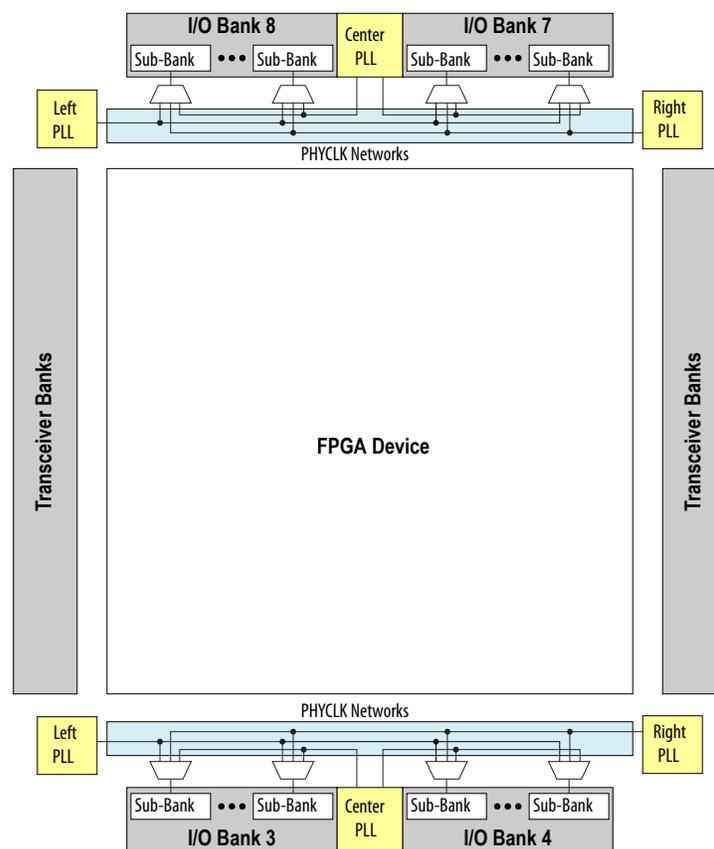
PHY Clock (PHYCLK) Networks

The PHYCLK network is a dedicated high-speed, low-skew balanced clock tree designed for a high-performance external memory interface.

The top and bottom sides of the Stratix V devices three PHYCLK networks each. Each PHYCLK network spans across one I/O bank and is driven by one of the left, right, or center PLLs located at that device side.

The following figure shows the PHYCLK networks available in the Stratix V devices.

Figure 7-6: PHYCLK Networks in Stratix V Devices



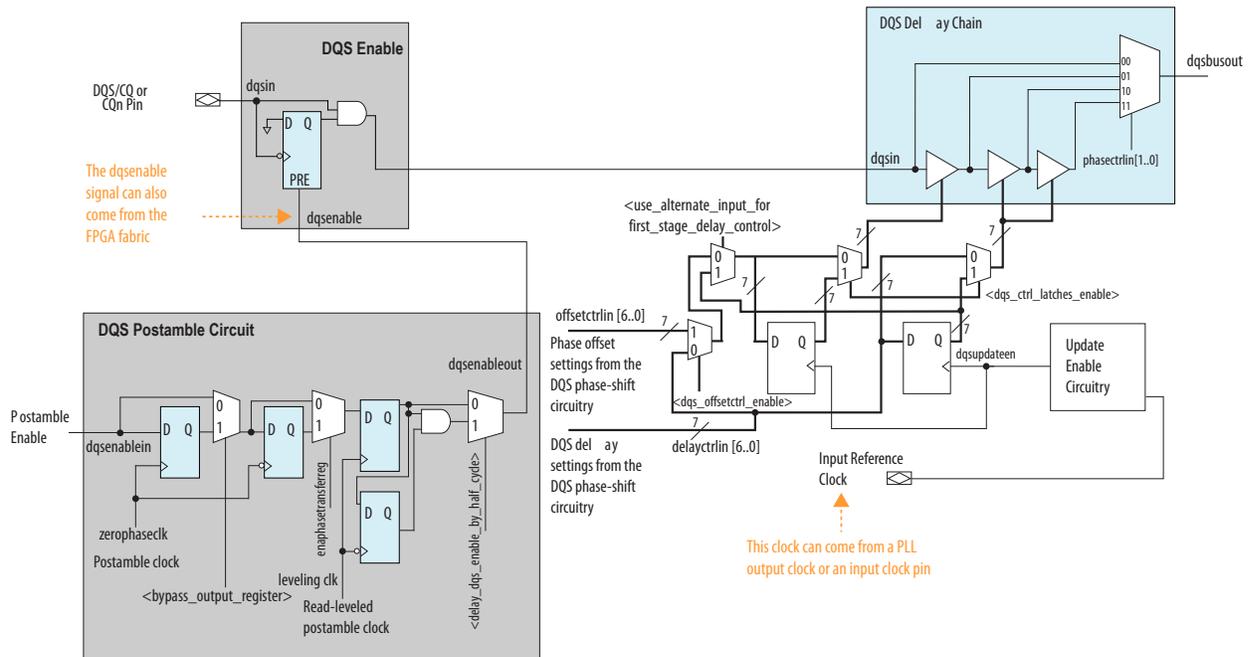
The PHYCLK network can be used to drive I/O sub-banks in each I/O bank. Each I/O sub-bank can be driven by only one PHYCLK network—all I/O pins in an I/O sub-bank are driven by the same PHYCLK network. The UniPHY IP for Stratix V devices uses the PHYCLK network to improve external memory interface performance.

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.

The following figure shows the DQS logic block.

Figure 7-7: DQS Logic Block in Stratix V Devices



Update Enable Circuitry

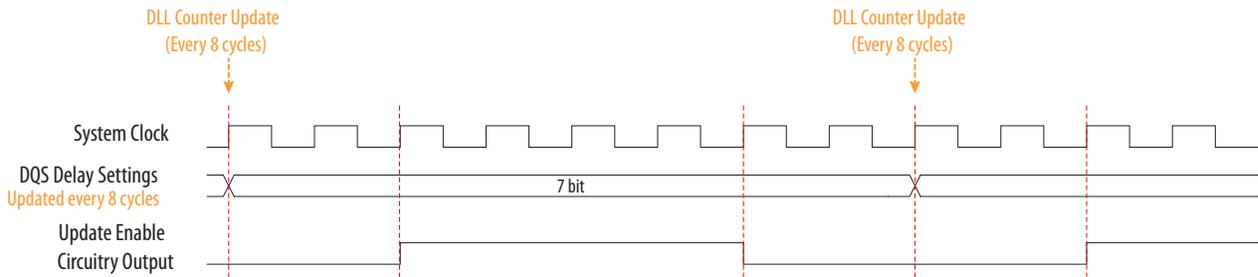
The update enable circuitry enables the registers to allow enough time for the DQS delay settings to travel from the DQS phase-shift circuitry or core logic to all the DQS logic blocks before the next change.

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, which allows them to be adjusted at the same time.

The circuitry uses the input reference clock or a user clock from the core to generate the update enable output. The UniPHY intellectual property (IP) uses this circuit by default.

Figure 7-8: DQS Update Enable Waveform

This figure shows an example waveform of the update enable circuitry output.



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 that have the same characteristics:

- Delay elements in the DQS logic block
- Delay elements in the DLL

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 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 number of delay chains required is transparent because the UniPHY IP automatically sets it when you choose the operating frequency.

In Stratix V devices, if you do not use the DLL 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 IP. These settings control 1, 2, 3, or all 4 delay elements in the DQS delay chains. The UniPHY IP core 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.

Related Information

- [ALTDQ_DQS2 IP Core User Guide](#)
Provides more information about programming the delay chains.
- [Delay Chains](#) on page 7-26

DQS Postamble Circuitry

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.

The Stratix V devices contain dedicated postamble registers that you can control to ground the shifted DQS signal that is used to clock the DQ input registers at the end of a read operation. This function ensures that any glitches on the DQS input signal during the end of a read operation and occurring while DQS is in a postamble state do not affect the DQ IOE registers.

- For preamble state, the DQS is low, just after a high-impedance state.
- For postamble state, the DQS is low, just before it returns to a high-impedance state.

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.

Half Data Rate Block

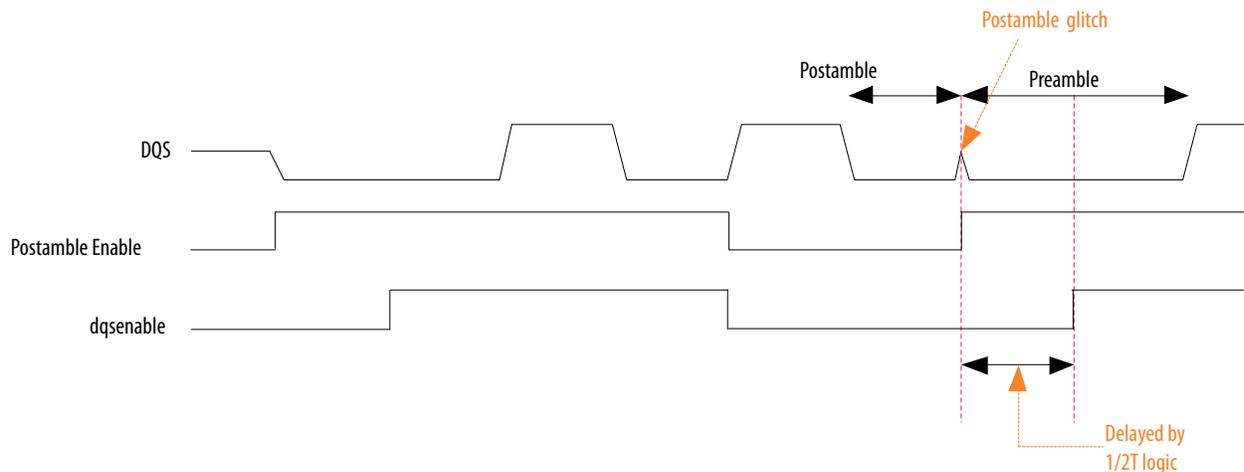
The Stratix V devices contain a half data rate (HDR) block in the postamble enable circuitry.

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.

Using the HDR block as the first stage capture register in the postamble enable circuitry block is optional. Altera recommends using these registers if the controller is running at half the frequency of the I/Os.

Figure 7-9: Avoiding Glitch on a Non-Consecutive Read Burst Waveform

This figure shows how to avoid postamble glitches using the HDR block.

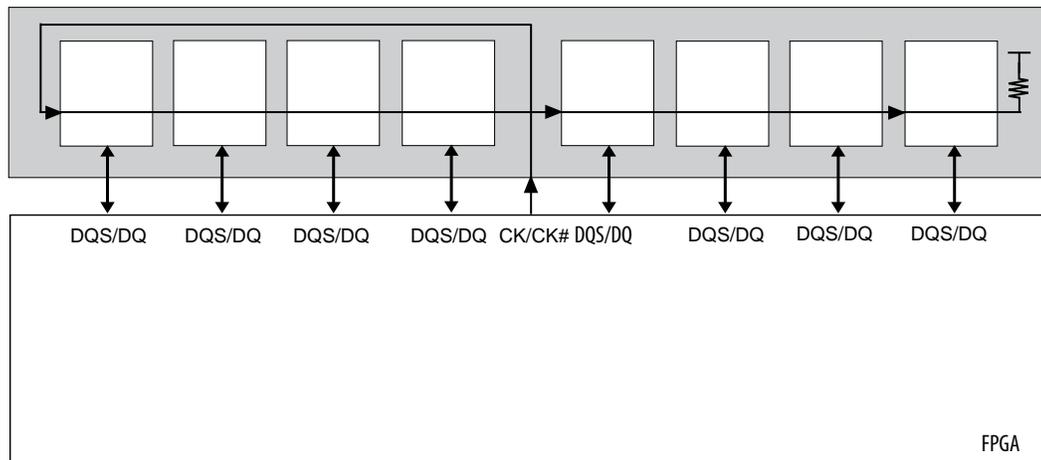


Leveling Circuitry

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.

The following figure shows the clock topology in DDR3 SDRAM unbuffered modules.

Figure 7-10: DDR3 SDRAM Unbuffered Module Clock Topology



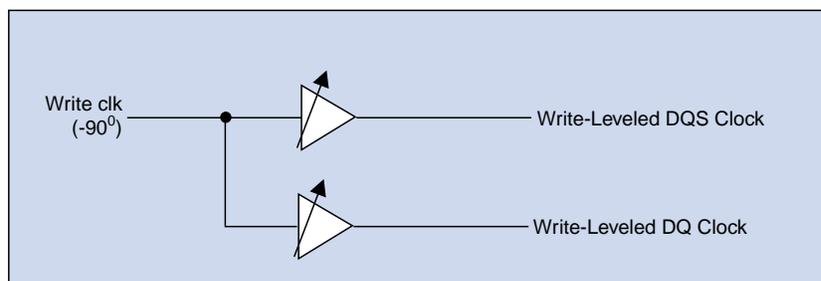
Because the data and read strobe signals are still point-to-point, take special care to ensure that the timing relationship between the CK/CK# and DQS signals (t_{DQSS} , t_{DSS} , and t_{DSH}) during a write is met at every device on the modules. In a similar way, read data coming back into the FPGA from the memory is also staggered.

The Stratix V devices 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 chain taps, such that each delay chain tap 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 IP core to select which clock phases are to be used for that x4 or x8 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 7-11: Write-Leveling Delay Chains and Multiplexers

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 IP feeds the write-leveling circuitry to produce the clock to generate the DQS and DQ signals. During initialization, the UniPHY IP 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 IP dynamically calibrates the alignment for read and write leveling during the initialization process.

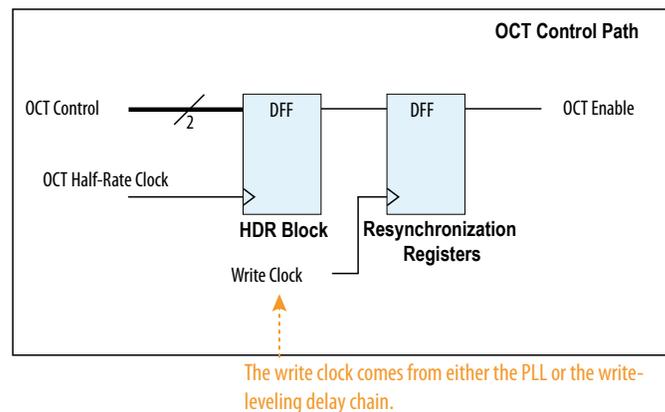
Related Information

- [Functional Description - UniPHY](#)
Provides more information about the UniPHY IP.
- [DDR2, DDR3, and DDR4 SDRAM Board Design Guidelines chapter. External Memory Interface Volume 2](#)
Provides layout guidelines for DDR3 SDRAM interface.

Dynamic OCT Control

The dynamic OCT control block includes all the registers that are required to dynamically turn the on-chip parallel termination (R_T OCT) on during a read and turn R_T OCT off during a write.

Figure 7-12: Dynamic OCT Control Block for Stratix V Devices



Related Information

- [Dynamic OCT in Stratix V Devices](#) on page 5-31
Provides more information about dynamic OCT control.
- [On-Chip I/O Termination in Stratix V Devices](#) on page 5-24
Provide more information for on-chip termination in Stratix V devices.

IOE Registers

The IOE registers are expanded to allow source-synchronous systems to have faster register-to-FIFO transfers and resynchronization. All top, bottom, and right IOEs have the same capability.

Input Registers

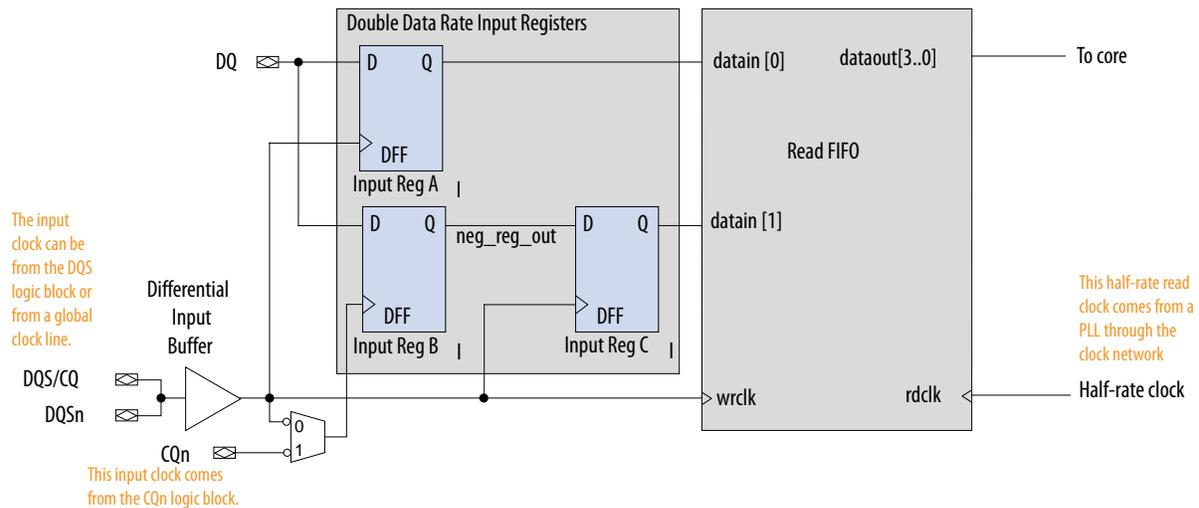
The input path consists of the DDR input registers and the read FIFO block. You can bypass each block of the input path.

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 and lowers the data rate to half rate.

The following figure shows the registers available in the Stratix V input path. For DDR3 and DDR2 SDRAM interfaces, the DQS and DQSn signals must be inverted. If you use Altera's memory interface IPs, the DQS and DQSn signals are automatically inverted.

Figure 7-13: IOE Input Registers for Stratix V Devices



Output Registers

The Stratix V output and output-enable 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.

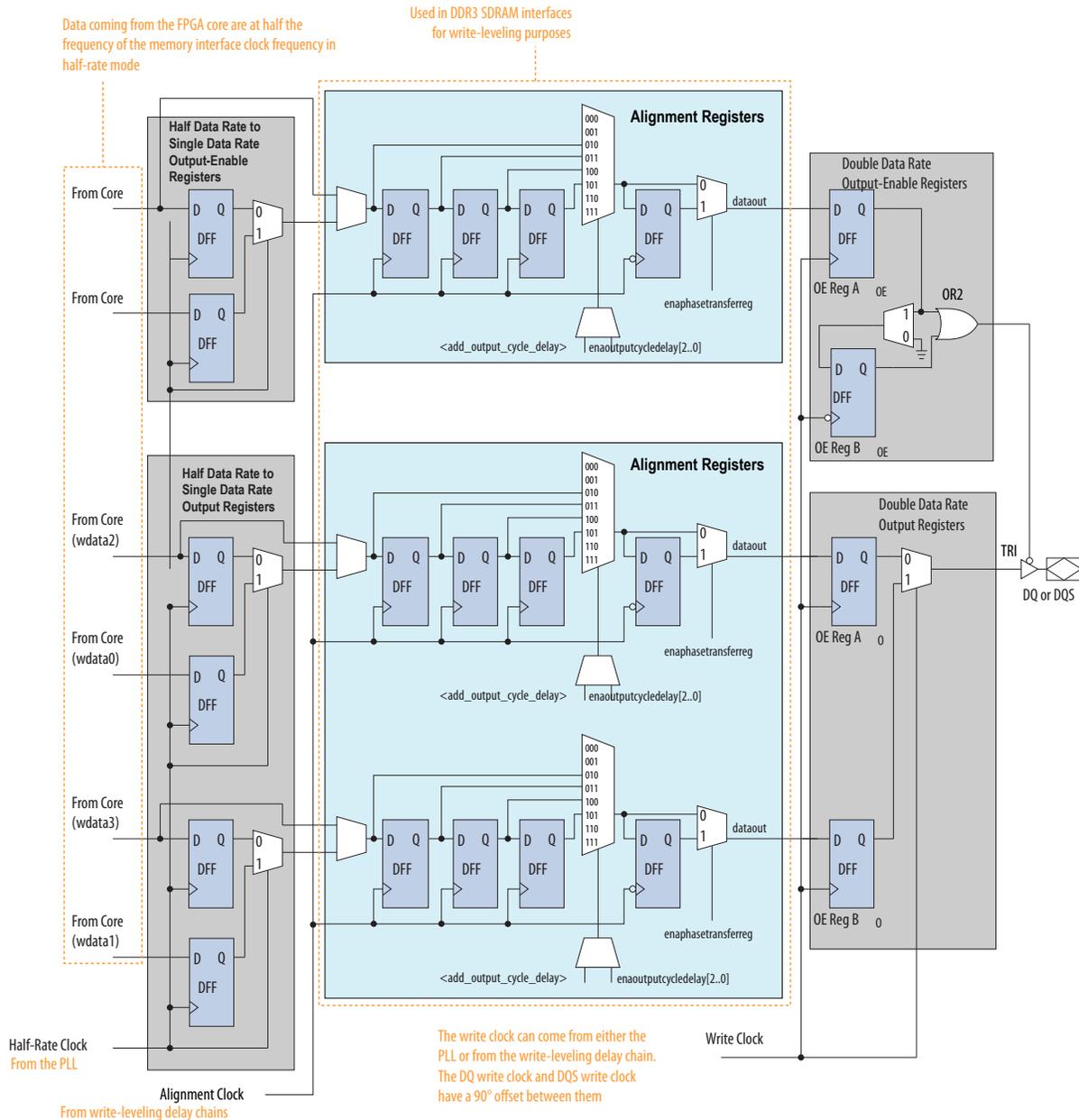
The output path is designed to route combinatorial or registered single data rate (SDR) outputs and full-rate or half-rate DDR outputs from the FPGA core. Half-rate data is converted to full-rate with the HDR block, clocked by the half-rate clock from the PLL.

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.

The output-enable path has a structure similar to the output path—ensuring that the output-enable path goes through the same delay and latency as the output path.

Figure 7-14: IOE Output and Output-Enable Path Registers

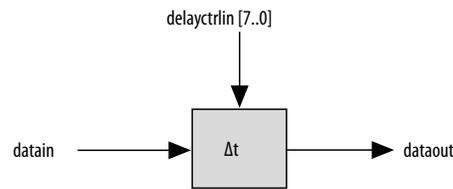
The following figure shows the registers available in the output and output-enable paths. You can bypass each register block of the output and output-enable paths.



Delay Chains

The Stratix V devices contain 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.

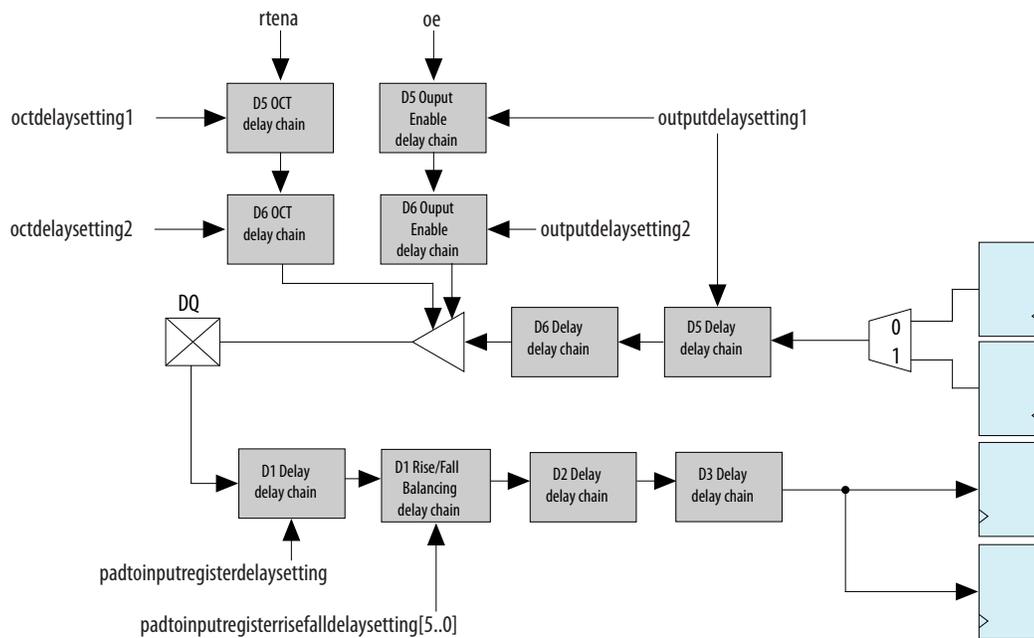
Figure 7-15: Delay Chain



Every I/O block contains two delay chains between the following elements:

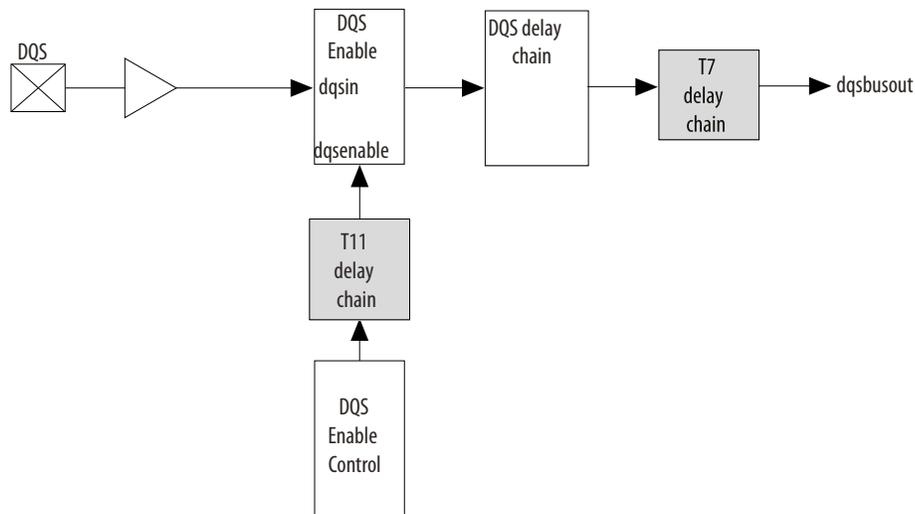
- The output registers and output buffer (in series)
- The input buffer and input register
- The output enable and output buffer
- The R_T OCT enable-control register and output buffer

Figure 7-16: 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 7-17: Delay Chains in the DQS Input Path

**Related Information**

- [ALTDQ_DQS2 IP Core User Guide](#)
Provides more information about programming the delay chains.
- [DQS Delay Chain](#) on page 7-21

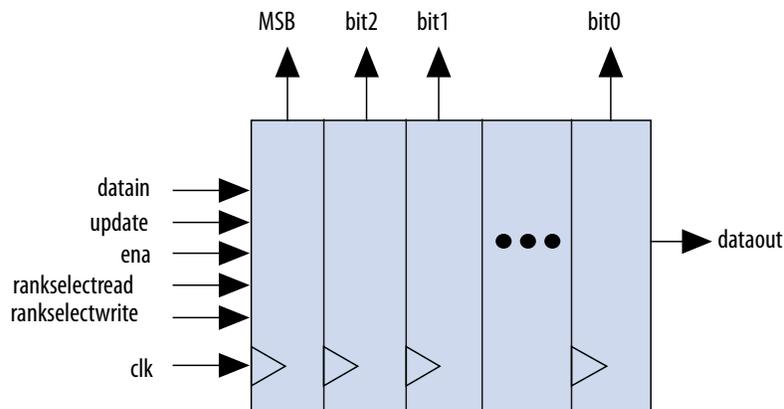
I/O and DQS Configuration Blocks

The I/O and DQS configuration blocks 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.
- Every DQS pin contains one DQS configuration block in addition to the I/O configuration register.

Figure 7-18: Configuration Block (I/O and DQS)

This figure shows the I/O configuration block and the DQS configuration block circuitry.



Related Information[ALTDQ_DQS2 IP Core User Guide](#)

Provides more information about programming the delay chains.

External Memory Interfaces in Stratix V Devices Revision History

Date	Version	Changes
December 2015	2015.12.21	Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i> .
June 2014	2014.06.30	<ul style="list-style-type: none"> Updated DDR3 1.35 V (DDR3L) performance from 933 MHz to 800 MHz.
January 2014	2014.01.10	<ul style="list-style-type: none"> Updated the figure that shows the delay chains in the Stratix V I/O block. Added related information link to ALTDQ_DQS2 Megafunction User Guide for more information about using the delay chains. Added link to Altera's External Memory Spec Estimator tool to the topic listing the external memory interface performance.
May 2013	2013.05.06	<ul style="list-style-type: none"> Moved all links to the Related Information section of respective topics for easy reference. Added link to the known document issues in the Knowledge Base. Added related information link to DDR2 and DDR3 SDRAM Board Design Guidelines. Performed some minor text edits to improve accuracy.
December 2012	2012.11.28	<ul style="list-style-type: none"> Reorganized content and updated template. Added RLDRAM 3 support. Added performance information for external memory interfaces. Separated the DQ/DQS groups tables into separate topics for each device variant for easy reference. Moved the PHYCLK networks pin placement guideline to the Planning Pin and FPGA Resources chapter of the <i>External Memory Interface Handbook</i>. Removed guidelines on DDR2 and DDR3 SDRAM DIMM interfaces. Refer to the relevant sections in the External Memory Interface Handbook for the information. Corrected “Gray-code” to “Binary-Code” in the “Phase Offset Control” section. Removed the topic about phase offset control. Removed the topics about I/O and DQS configuration block bit sequence. Refer to the relevant sections in the ALTDQ_DQS2 Megafunction User Guide.

Date	Version	Changes
June 2012	1.4	<ul style="list-style-type: none"> • Added Table 7-6, Table 7-8, and Table 7-9. • Updated Table 7-2, Table 7-3, and Table 7-7. • Updated Figure 7-18. • Updated the “PHY Clock (PHYCLK) Networks” section.
November 2011	1.3	<ul style="list-style-type: none"> • Added “PHY Clock (PHYCLK) Networks” section. • Updated “Delay-Locked Loop” section. • Updated Figure 7-3, Figure 7-5, and Figure 7-7. • Updated Table 7-2, Table 7-3, Table 7-4, Table 7-5, and Table 7-6. • Minor text edits.
May 2011	1.2	<ul style="list-style-type: none"> • 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.

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

8

2020.09.10

SV51010



Subscribe



Send Feedback

This chapter describes the configuration schemes, design security, and remote system upgrade that are supported by the Stratix V devices.

Related Information

- [Stratix V Device Handbook: Known Issues](#)
Lists the planned updates to the *Stratix V Device Handbook* chapters.
- [Stratix V Device Overview](#)
Provides more information about configuration features supported for each configuration scheme.
- [Stratix V Device Datasheet](#)
Provides more information about the estimated uncompressed **.rbf** file sizes, FPP `DCLK-to-DATA[]` ratio, and timing parameters.
- [Configuration via Protocol \(CvP\) Implementation in Altera FPGAs User Guide](#)
Provides more information about the CvP configuration scheme.
- [Design Planning for Partial Reconfiguration](#)
Provides more information about partial reconfiguration.

Enhanced Configuration and Configuration via Protocol

Table 8-1: Configuration Schemes and Features of Stratix V Devices

Stratix V devices support 1.8 V, 2.5 V, and 3.0 V programming voltages and several configuration schemes.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

Mode	Data Width	Max Clock Rate (MHz)	Max Data Rate (Mbps)	Decompression	Design Security	Partial Reconfiguration ⁽¹⁴⁾	Remote System Update
AS through the EPCS and EPCQ serial configuration device	1 bit, 4 bits	100	—	Yes	Yes	—	Yes
PS through CPLD or external microcontroller	1 bit	125	125	Yes	Yes	—	—
FPP	8 bits	125	—	Yes	Yes	—	Parallel flash loader
	16 bits	125	—	Yes	Yes	Yes ⁽¹⁵⁾	
	32 bits	100	—	Yes	Yes	—	
CvP (PCIe)	x1, x2, x4, and x8 lanes	—	—	Yes	Yes	Yes	—
JTAG	1 bit	33	33	—	—	—	—

Instead of using an external flash or ROM, you can configure the Stratix V devices through PCIe using CvP. The CvP mode offers the fastest configuration rate and flexibility with the easy-to-use PCIe hard IP block interface. The Stratix V CvP implementation conforms to the PCIe 100 ms power-up-to-active time requirement.

Related Information

[Configuration via Protocol \(CvP\) Implementation in Altera FPGAs User Guide](#)

Provides more information about the CvP configuration scheme.

MSEL Pin Settings

To select a configuration scheme, hardwire the MSEL pins to V_{CCPGM} or GND without pull-up or pull-down resistors.

Note: Altera recommends connecting the MSEL pins directly to V_{CCPGM} or GND. Driving the MSEL pins from a microprocessor or another controlling device may not guarantee the V_{IL} or V_{IH} of the MSEL pins. The V_{IL} or V_{IH} of the MSEL pins must be maintained throughout configuration stages.

⁽¹⁴⁾ Partial reconfiguration is an advanced feature of the device family. If you are interested in using partial reconfiguration, contact Altera for support.

⁽¹⁵⁾ Supported at a maximum clock rate of 62.5 MHz.

Table 8-2: MSEL Pin Settings for Each Configuration Scheme of Stratix V Devices

Configuration Scheme	Compression Feature	Design Security Feature	V _{CCPGM} (V)	Power-On Reset (POR) Delay	Valid MSEL[4..0]
FPP x8	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	Enabled/ Disabled	1.8/2.5/3.0	Fast	10110
				Standard	11010
FPP x16	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	Enabled/ Disabled	1.8/2.5/3.0	Fast	00010
				Standard	00110
FPP x32	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	Enabled/ Disabled	1.8/2.5/3.0	Fast	01010
				Standard	01110
PS	Enabled/ Disabled	Enabled/ Disabled	1.8/2.5/3.0	Fast	10000
				Standard	10001
AS (x1 and x4)	Enabled/ Disabled	Enabled/ Disabled	3.0	Fast	10010
				Standard	10011
JTAG-based configuration	Disabled	Disabled	—	—	Use any valid MSEL pin settings above

Note: You must also select the configuration scheme in the **Configuration** page of the **Device and Pin Options** dialog box in the Intel Quartus Prime software. Based on your selection, the option bit in the programming file is set accordingly.

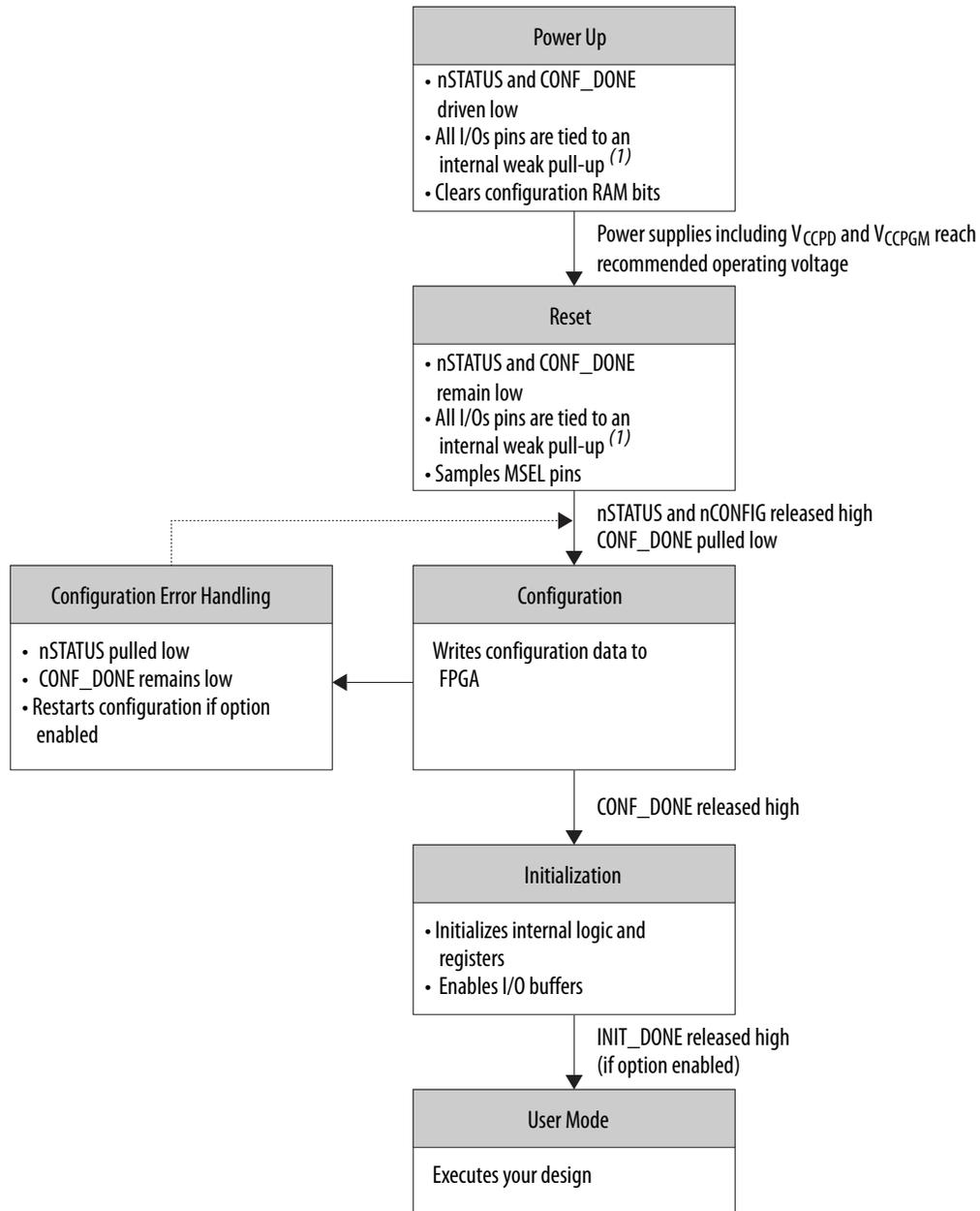
Related Information

- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
Provides more information about JTAG pins voltage-level connection.
- [Stratix V GT Device Family Pin Connection Guidelines](#)
Provides more information about JTAG pins voltage-level connection.

Configuration Sequence

Describes the configuration sequence and each configuration stage.

Figure 8-1: Configuration Sequence for Stratix V Devices



Note:

(1) The weak-pull up is enabled after the device has exited POR.

You can initiate reconfiguration by pulling the `nCONFIG` pin low to at least the minimum t_{CFG} low-pulse width except for configuration using the partial reconfiguration operation. When this pin is pulled low, the `nSTATUS` and `CONF_DONE` pins are pulled low and all I/O pins are tied to an internal weak pull-up.

Power Up

Power up all the power supplies that are monitored by the POR circuitry. All power supplies, including V_{CCPGM} and V_{CCPD} , must ramp up from 0 V to the recommended operating voltage level within the ramp-up time specification. Otherwise, hold the `nCONFIG` pin low until all the power supplies reach the recommended voltage level.

V_{CCPGM} Pin

The configuration input buffers do not have to share power lines with the regular I/O buffers 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} .

V_{CCPD} Pin

Use the V_{CCPD} pin, a dedicated programming power supply, to power the I/O pre-drivers and JTAG I/O pins (`TCK`, `TMS`, `TDI`, `TRST`, and `TDO`).

If V_{CCIO} of the bank is set to 2.5 V or lower, V_{CCPD} must be powered up at 2.5 V. If V_{CCIO} is set greater than 2.5 V, V_{CCPD} must be greater than V_{CCIO} . For example, when V_{CCIO} is set to 3.0 V, V_{CCPD} must be set at 3.0 V.

Related Information

- [Stratix V Device Datasheet](#)
Provides more information about the ramp-up time specifications.
- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
Provides more information about configuration pin connections.
- [Stratix V GT Device Family Pin Connection Guidelines](#)
Provides more information about configuration pin connections.
- [Device Configuration Pins](#) on page 8-11
Provides more information about configuration pins.
- [I/O Standards Voltage Levels in Stratix V Devices](#) on page 5-3
Provides more information about typical power supplies for each supported I/O standards in Stratix V devices.
- [I/O Features in Stratix V Devices](#)
Provides more information about typical power supplies for each supported I/O standards in Stratix V devices.

Reset

POR delay is the time frame between the time when all the power supplies monitored by the POR circuitry reach the recommended operating voltage and when `nSTATUS` is released high and the Stratix V device is ready to begin configuration.

Set the POR delay using the `MSEL` pins.

The user I/O pins are tied to an internal weak pull-up until the device is configured.

Related Information

- [MSEL Pin Settings](#) on page 8-2
- [Stratix V Device Datasheet](#)
Provides more information about the POR delay specification.

Configuration

For more information about the `DATA[]` pins for each configuration scheme, refer to the appropriate configuration scheme.

Configuration Error Handling

To restart configuration automatically, turn on the **Auto-restart configuration after error** option in the **General** page of the **Device and Pin Options** dialog box in the Intel Quartus Prime software.

If you do not turn on this option, you can monitor the `nSTATUS` pin to detect errors. To restart configuration, pull the `nCONFIG` pin low for at least the duration of t_{CFG} .

Related Information

[Stratix V Device Datasheet](#)

Provides more information about t_{STATUS} and t_{CFG} timing parameters.

Initialization

The initialization clock source is from the internal oscillator, `CLKUSR` pin, or `DCLK` pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix V device will be provided with enough clock cycles for proper initialization.

Note: If you use the optional `CLKUSR` pin as the initialization clock source and the `nCONFIG` pin is pulled low to restart configuration during device initialization, ensure that the `CLKUSR` or `DCLK` pin continues toggling until the `nSTATUS` pin goes low and then goes high again.

The `CLKUSR` pin 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 the `CONF_DONE` pin goes high, the `CLKUSR` or `DCLK` pin is enabled after the time specified by t_{CD2CU} . When this time period elapses, Stratix V devices require a minimum number of clock cycles as specified by T_{init} to initialize properly and enter user mode as specified by the t_{CD2UMC} parameter.

Related Information

[Stratix V Device Datasheet](#)

Provides more information about t_{CD2CU} , t_{init} , and t_{CD2UMC} timing parameters, and initialization clock source.

User Mode

You can enable the optional `INIT_DONE` pin to monitor the initialization stage. After the `INIT_DONE` pin is pulled high, initialization completes and your design starts executing. The user I/O pins will then function as specified by your design.

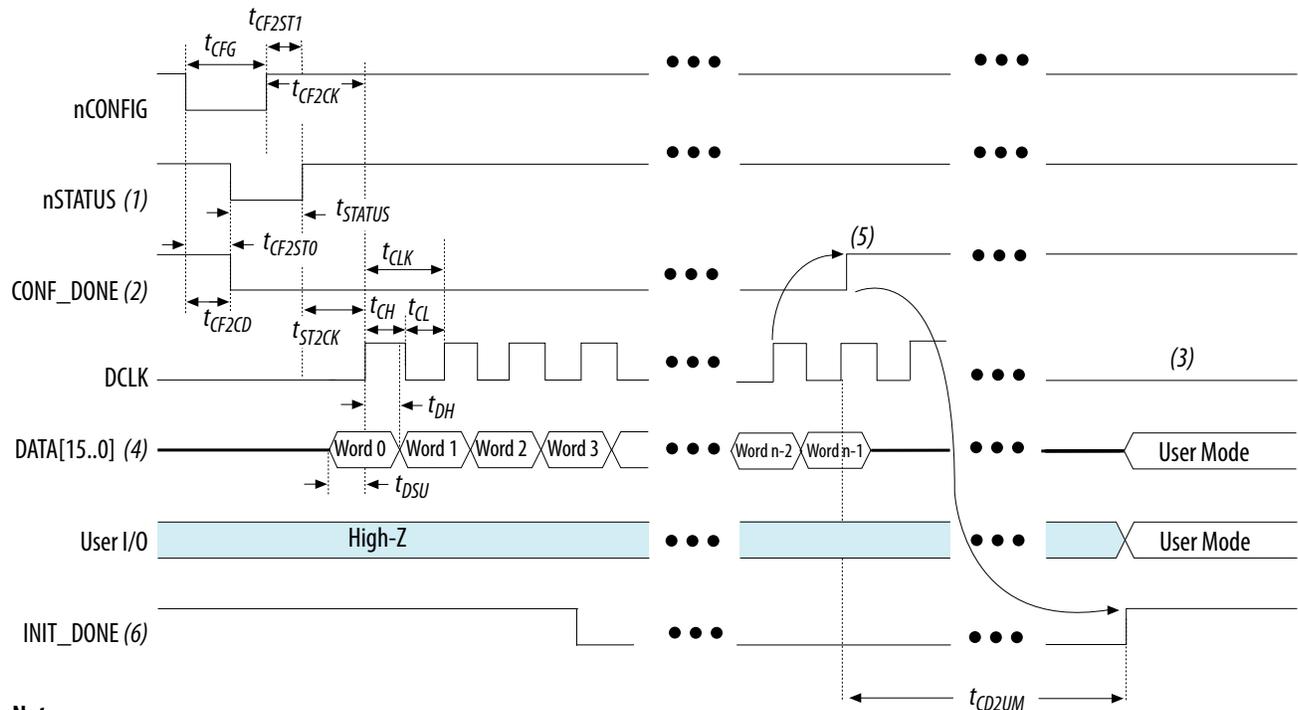
During device initialization stage, the FPGA registers, core logic, and I/O are not released from reset at the same time. The increase in clock frequency, device size, and design complexity require a reset strategy that considers the differences in the release from reset. Intel recommends that you use the following implementations to reset your design properly and until the device has fully entered user mode:

- Hold the entire design in reset for a period of time by following the `CONF_DONE` high to user mode (t_{CD2UM}) or `CONF_DONE` high to user mode with `CLKUSR` option turned on (t_{CD2UMC}) specifications as defined in the *Stratix V Device Datasheet* before starting any operation after the device enters into user mode. For example, the t_{CD2UM} range for Stratix V device is between 175 us to 437 us.
- If you have an external device that reacts based on an Intel FPGA output pin, perform the following steps to avoid false reaction:
 - Ensure that the external device ignores the state of the FPGA output pin until the external `INIT_DONE` pin goes high. Refer to the t_{CD2UM} or t_{CD2UMC} specifications in the *Stratix V Device Datasheet* for more information.
 - Keep the input state to the external device constant by using the external logic until the external `INIT_DONE` pin goes high.

Configuration Timing Waveforms

FPP Configuration Timing

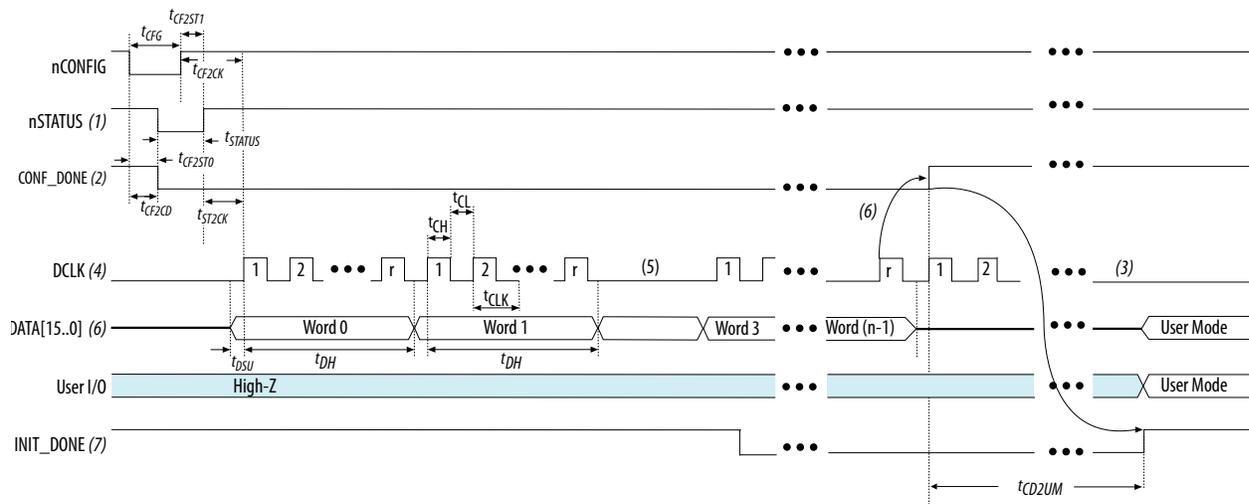
Figure 8-2: FPP Configuration Timing Waveform when DCLK-to-DATA[] Ratio is 1



Notes:

- (1) After power up, the FPGA holds nSTATUS low for the time of the POR delay.
- (2) After power up, before and during configuration, CONF_DONE is low.
- (3) Do not leave DCLK floating after configuration. DCLK is ignored after configuration is complete. It can toggle high or low if required.
- (4) For FPP x16, use DATA[15..0]. For FPP x8, use DATA[7..0]. DATA[15..5] are available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.
- (5) To ensure a successful configuration, send the entire configuration data to the FPGA. CONF_DONE is released high when the FPGA 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.
- (6) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Figure 8-3: FPP Configuration Timing Waveform when DCLK-to-DATA[] Ratio is >1



Notes:

- (1) After power up, the FPGA holds nSTATUS low for the time as specified by the POR delay.
- (2) After power up, before and during configuration, CONF_DONE is low.
- (3) Do not leave DCLK floating after configuration. DCLK is ignored after configuration is complete. It can toggle high or low if required.
- (4) "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 the DCLK-to-DATA[] Ratio table.
- (5) If needed, pause DCLK by holding it low. When DCLK restarts, the external host must provide data on the DATA[15..0] pins prior to sending the first DCLK rising edge.
- (6) To ensure a successful configuration, send the entire configuration data to the FPGA. CONF_DONE is released high after the FPGA 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.

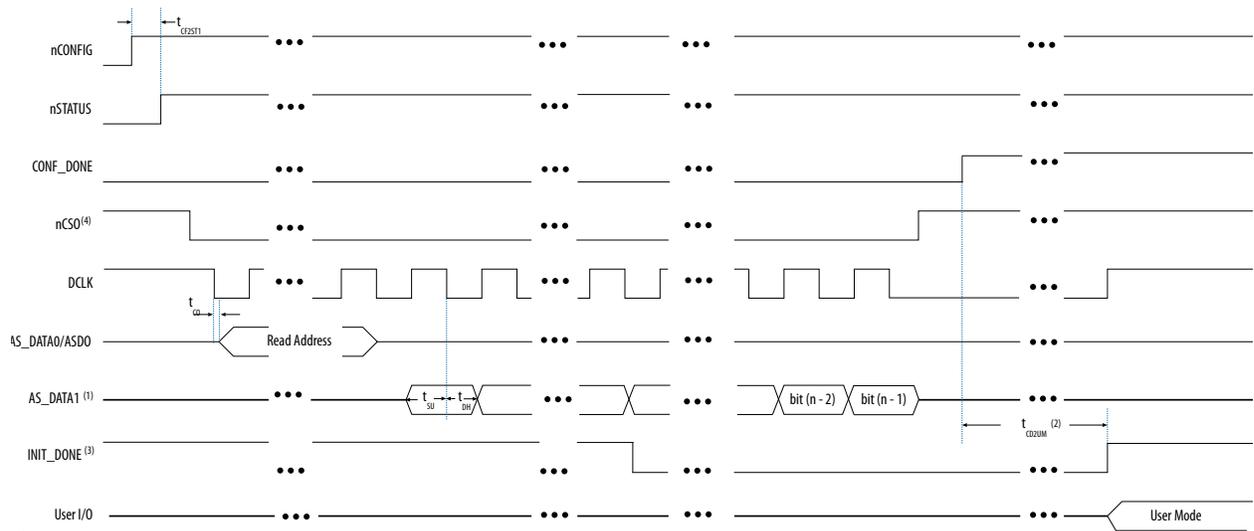
Related Information

Stratix V Device Datasheet

Provides more information about the FPP timing parameters.

AS Configuration Timing

Figure 8-4: AS Configuration Timing Waveform



Notes:

- (1) If you are using AS x4 mode, this signal represents the AS_DATA[3..0] and EPCQ sends in 4-bits of data for each DCLK cycle.
- (2) The initialization clock can be from the internal oscillator or CLKUSR pin.
- (3) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.
- (4) The time between nCSO falling edge to the first toggling of DCLK is more than 15 ns.

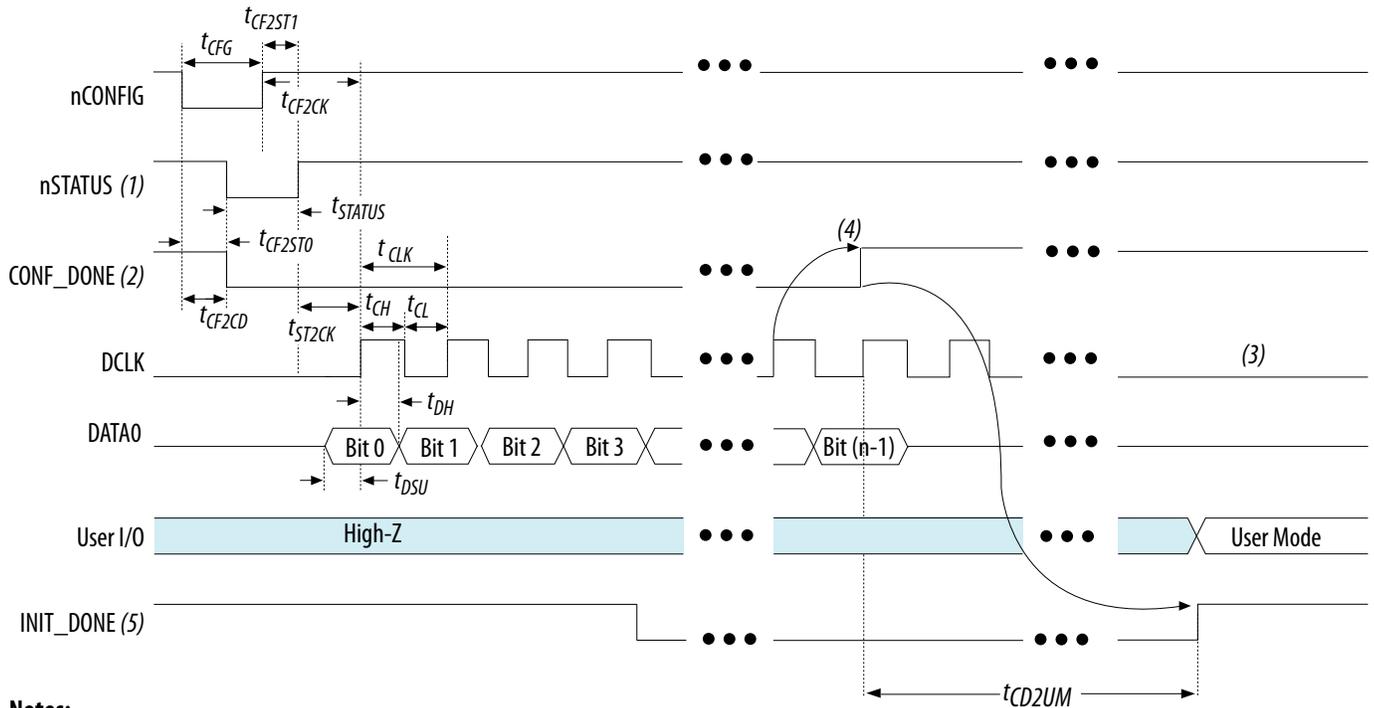
Related Information

[Stratix V Device Datasheet](#)

Provides more information about the AS timing parameters.

PS Configuration Timing

Figure 8-5: PS Configuration Timing Waveform



Notes:

- (1) After power up, the FPGA holds nSTATUS low for the time of the POR delay.
- (2) After power up, before and during configuration, CONF_DONE is low.
- (3) Do not leave DCLK floating after configuration. DCLK is ignored after configuration is complete. It can toggle high or low if required.
- (4) To ensure a successful configuration, send the entire configuration data to the FPGA. CONF_DONE is released high after the FPGA 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.
- (5) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Related Information

[Stratix V Device Datasheet](#)

Provides more information about PS timing parameters.

Device Configuration Pins

Configuration Pins Summary

The following table lists the Stratix V configuration pins and their power supply.

- Note:**
1. The TDI, TMS, TCK, TDO, and TRST pins are powered by V_{CCPD} of the bank in which the pin resides.
 2. The CLKUSR, DEV_OE, DEV_CLRn, DATA[31..1], and DATA0 pins are powered by V_{CCPGM} during configuration and by V_{CCIO} of the bank in which the pin resides if you use it as a user I/O pin.
 3. The DCLK, AS_DATA0/ASDO, AS_DATA1, AS_DATA2, AS_DATA3, and nCS0 pins have 25 kOhm pull-up resistors when the MSEL pins are set to AS configuration scheme.

Table 8-3: Configuration Pin Summary for Stratix V Devices

Configuration Pin	Configuration Scheme	Input/Output	User Mode	Powered By
TDI	JTAG	Input	—	V _{CCPD}
TMS	JTAG	Input	—	V _{CCPD}
TCK	JTAG	Input	—	V _{CCPD}
TDO	JTAG	Output	—	V _{CCPD}
TRST	JTAG	Input	—	V _{CCPD}
CLKUSR	All schemes	Input	I/O	V _{CCPGM} /V _{CCIO} ⁽¹⁶⁾
CRC_ERROR	Optional, all schemes	Output	I/O	Pull-up
CONF_DONE	All schemes	Bidirectional	—	V _{CCPGM} /Pull-up
DCLK	FPP and PS	Input	—	V _{CCPGM}
	AS	Output	—	V _{CCPGM}
DEV_OE	Optional, all schemes	Input	I/O	V _{CCPGM} /V _{CCIO} ⁽¹⁶⁾
DEV_CLRn	Optional, all schemes	Input	I/O	V _{CCPGM} /V _{CCIO} ⁽¹⁶⁾
INIT_DONE	Optional, all schemes	Output	I/O	Pull-up
MSEL[4..0]	All schemes	Input	—	V _{CCPGM}
nSTATUS	All schemes	Bidirectional	—	V _{CCPGM} /Pull-up
nCE	All schemes	Input	—	V _{CCPGM}
nCEO	All schemes	Output	I/O	Pull-up
nCONFIG	All schemes	Input	—	V _{CCPGM}
DATA[31..1]	FPP	Input	I/O	V _{CCPGM} /V _{CCIO} ⁽¹⁶⁾

⁽¹⁶⁾ This pin is powered by V_{CCPGM} during configuration and powered by V_{CCIO} of the bank in which the pin resides when you use this pin as a user I/O pin.

Configuration Pin	Configuration Scheme	Input/Output	User Mode	Powered By
DATA0	FPP and PS	Bidirectional	I/O	$V_{CCPGM}/V_{CCIO}^{(16)}$
nCS0	AS	Output	—	V_{CCPGM}
nIO_PULLUP	All schemes	Input	—	V_{CCPGM}
AS_DATA[3..1]	AS	Bidirectional	—	V_{CCPGM}
AS_DATA0/ASDO	AS	Bidirectional	—	V_{CCPGM}
PR_REQUEST	Partial Reconfiguration	Input	I/O	$V_{CCPGM}/V_{CCIO}^{(16)}$
PR_READY	Partial Reconfiguration	Output	I/O	$V_{CCPGM}/V_{CCIO}^{(16)}$
PR_ERROR	Partial Reconfiguration	Output	I/O	$V_{CCPGM}/V_{CCIO}^{(16)}$
PR_DONE	Partial Reconfiguration	Output	I/O	$V_{CCPGM}/V_{CCIO}^{(16)}$
CvP_CONFDONE	CvP (PCIe)	Output	I/O	$V_{CCPGM}/V_{CCIO}^{(16)}$

Related Information

- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
Provides more information about each configuration pin.
- [Stratix V GT Device Family Pin Connection Guidelines](#)
Provides more information about each configuration pin.

I/O Standards and Drive Strength for Configuration Pins

In configuration mode, the output drive strength is set as listed in the table below. Dual-function pin output drive strength is programmable if it is used as a regular I/O pin.

Table 8-4: I/O Standards and Drive Strength for Configuration Pins

The configuration pins listed support only fast slew rate and OCT is not enabled for these pins.

Configuration Pin	Type	I/O Standard	Drive Strength (mA)
nSTATUS	Dedicated	3.0 V LVTTL	4
CONF_DONE	Dedicated	3.0 V LVTTL	4
CvP_CONFDONE	Dual Function	3.0 V LVTTL	4
DCLK	Dedicated	3.0 V LVTTL	12

Configuration Pin	Type	I/O Standard	Drive Strength (mA)
TDO	Dedicated	3.0 V LVTTL	12
AS_DATA0 / ASDO	Dedicated	3.0 V LVTTL	8
AS_DATA1	Dedicated	3.0 V LVTTL	8
AS_DATA2	Dedicated	3.0 V LVTTL	8
AS_DATA3	Dedicated	3.0 V LVTTL	8
INIT_DONE	Dual Function	3.0 V LVTTL	8
CRC_ERROR	Dual Function	3.0 V LVTTL	8
nCSO	Dedicated	3.0 V LVTTL	8

Note: You can use the V_{CCPGM} voltage other than 3 V. Refer to the *I/O Standards and Drive Strength for Configuration Pins* table for the I/O standard for each Configuration Pin.

Configuration Pin Options in the Intel Quartus Prime Software

The following table lists the dual-purpose configuration pins available in the **Device and Pin Options** dialog box in the Intel Quartus Prime software.

Table 8-5: Configuration Pin Options

Configuration Pin	Category Page	Option
CLKUSR	General	Enable user-supplied start-up clock (CLKUSR)
DEV_CLRn	General	Enable device-wide reset (DEV_CLRn)
DEV_OE	General	Enable device-wide output enable (DEV_OE)
INIT_DONE	General	Enable INIT_DONE output
nCEO	General	Enable nCEO pin
CRC_ERROR	Error Detection CRC	Enable Error Detection CRC_ERROR pin
		Enable open drain on CRC_ERROR pin
		Enable internal scrubbing
PR_REQUEST	General	Enable PR pin
PR_READY		
PR_ERROR		
PR_DONE		

Related Information**[Reviewing Printed Circuit Board Schematics with the Quartus II Software](#)**

Provides more information about the device and pin options dialog box setting.

Fast Passive Parallel Configuration

The FPP configuration scheme uses an external host, such as a microprocessor, MAX[®] II device, or MAX V device. This scheme is the fastest method to configure Stratix V devices. The FPP configuration scheme supports 8-, 16-, and 32-bits data width.

You can use an external host to control the transfer of configuration data from an external storage such as flash memory to the FPGA. The design that controls the configuration process resides in the external host. You can store the configuration data in Raw Binary File (**.rbf**), Hexadecimal (Intel-Format) File (**.hex**), or Tabular Text File (**.ttf**) formats.

You can use the PFL IP core with a MAX II or MAX V device to read configuration data from the flash memory device and configure the Stratix V device.

Note: Two `DCLK` falling edges are required after the `CONF_DONE` pin goes high to begin the initialization of the device for both uncompressed and compressed configuration data in an FPP configuration.

Related Information

- **[Parallel Flash Loader IP Core User Guide](#)**
- **[Stratix V Device Datasheet](#)**

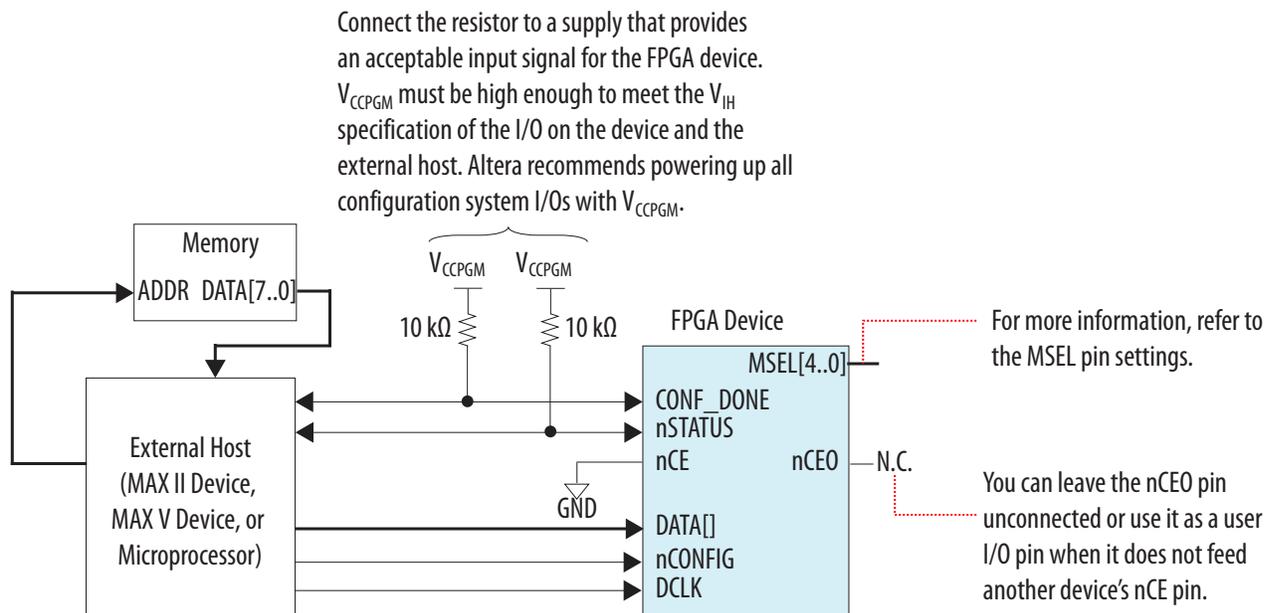
Provides more information about the FPP configuration timing.

Fast Passive Parallel Single-Device Configuration

To configure a Stratix V device, connect the device to an external host as shown in the following figure.

Note: If you are using the FPP x8 configuration mode, use `DATA[7..0]` pins. If you are using FPP x16 configuration mode, use `DATA[15..0]` pins.

Figure 8-6: Single Device FPP Configuration Using an External Host



Fast Passive Parallel Multi-Device Configuration

You can configure multiple Stratix V devices that are connected in a chain.

Pin Connections and Guidelines

Observe the following pin connections and guidelines for this configuration setup:

- Tie the following pins of all devices in the chain together:
 - nCONFIG
 - nSTATUS
 - DCLK
 - DATA[]
 - CONF_DONE

By tying the CONF_DONE and nSTATUS pins together, the devices initialize and enter user mode at the same time. If any device in the chain detects an error, configuration stops for the entire chain and you must reconfigure all the devices. For example, if the first device in the chain flags an error on the nSTATUS pin, it resets the chain by pulling its nSTATUS pin low.

- Ensure that DCLK and DATA[] are buffered for every fourth device to prevent signal integrity and clock skew problems.
- All devices in the chain must use the same data width.
- If you are configuring the devices in the chain using the same configuration data, the devices must be of the same package and density.

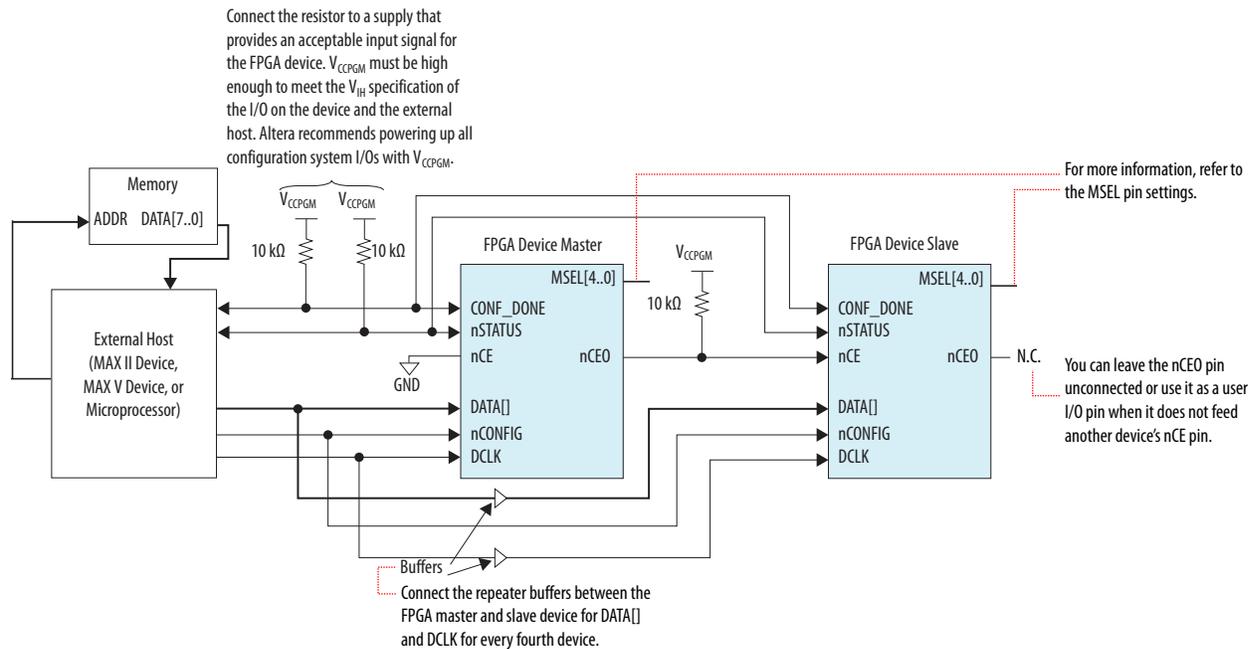
Using Multiple Configuration Data

To configure multiple Stratix V devices in a chain using multiple configuration data, connect the devices to an external host as shown in the following figure.

Note: If you are using the FPP x8 configuration mode, use $DATA[7..0]$ pins. If you are using FPP x16 configuration mode, use $DATA[15..0]$ pins.

Note: By default, the $nCEO$ pin is disabled in the Intel Quartus Prime software. For multi-device configuration chain, you must enable the $nCEO$ pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

Figure 8-7: Multiple Device FPP Configuration Using an External Host When Both Devices Receive a Different Set of Configuration Data



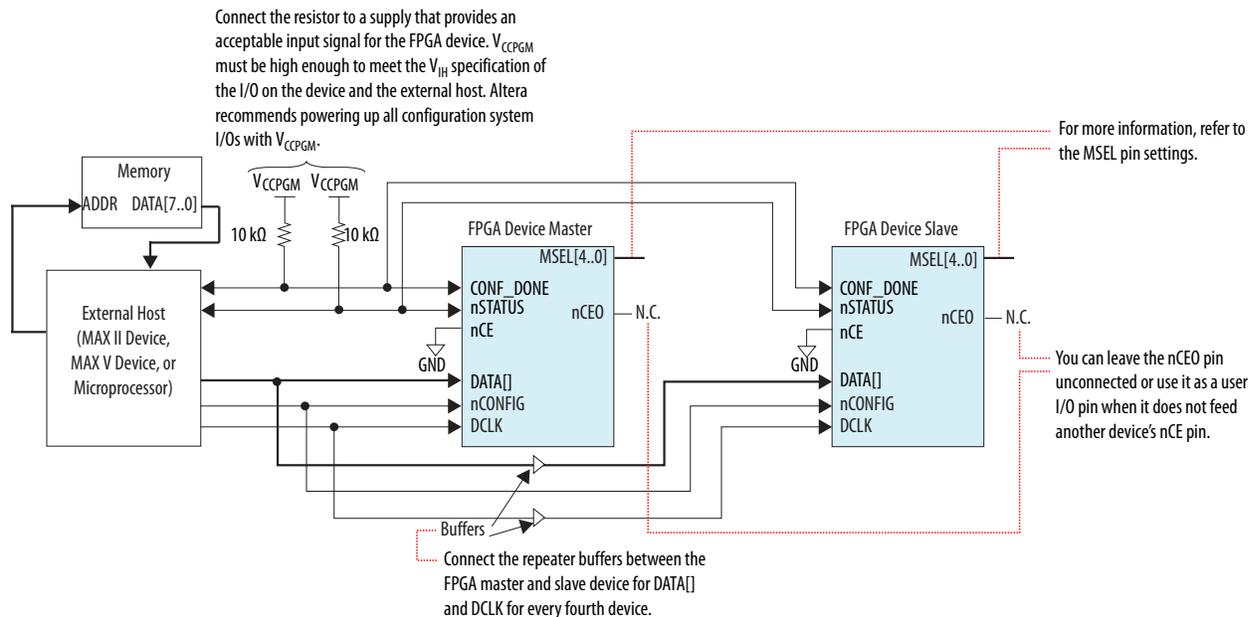
When a device completes configuration, its $nCEO$ pin is released low to activate the nCE pin of the next device in the chain. Configuration automatically begins for the second device in one clock cycle.

Using One Configuration Data

To configure multiple Stratix V devices in a chain using one configuration data, connect the devices to an external host as shown in the following figure.

Note: If you are using the FPP x8 configuration mode, use $DATA[7..0]$ pins. If you are using FPP x16 configuration mode, use $DATA[15..0]$ pins.

Note: By default, the $nCEO$ pin is disabled in the Intel Quartus Prime software. For multi-device configuration chain, you must enable the $nCEO$ pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

Figure 8-8: Multiple Device FPP Configuration Using an External Host When Both Devices Receive the Same Data

The nCE pins of the device in the chain are connected to GND, allowing configuration for these devices to begin and end at the same time.

Transmitting Configuration Data

This section describes how to transmit configuration data when you are using **.rbf** file for FPP x8, x16, and x32 configuration schemes. The configuration data in the **.rbf** file is little endian.

For example, if the **.rbf** file contains the byte sequence 02 1B EE 01, refer to the following tables for details on how this data is transmitted in the FPP x8, x16, and x32 configuration schemes.

Table 8-6: Transmitting Configuration Data for FPP x8 Configuration Scheme

In FPP x8 configuration scheme, the LSB of a byte is BIT0, and the MSB is BIT7.

BYTE0 = 02	BYTE1 = 1B	BYTE2 = EE	BYTE3 = 01
D[7..0]	D[7..0]	D[7..0]	D[7..0]
0000 0010	0001 1011	1110 1110	0000 0001

Table 8-7: Transmitting Configuration Data for FPP x16 Configuration Scheme

In FPP x16 configuration scheme, the first byte in the file is the LSB of the configuration word, and the second byte in the file is the MSB of the configuration word.

WORD0 = 1B02		WORD1 = 01EE	
LSB: BYTE0 = 02	MSB: BYTE1 = 1B	LSB: BYTE2 = EE	MSB: BYTE3 = 01
D[7..0]	D[15..8]	D[7..0]	D[15..8]
0000 0010	0001 1011	1110 1110	0000 0001

Table 8-8: Transmitting Configuration Data for FPP x32 Configuration Scheme

In FPP x32 configuration scheme, the first byte in the file is the LSB of the configuration double word, and the fourth byte is the MSB.

Double Word = 01EE1B02			
LSB: BYTE0 = 02	BYTE1 = 1B	BYTE2 = EE	MSB: BYTE3 = 01
D[7..0]	D[15..8]	D[23..16]	D[31..24]
0000 0010	0001 1011	1110 1110	0000 0001

Ensure that you do not swap the upper bits or bytes with the lower bits or bytes when performing the FPP configuration. Sending incorrect configuration data during the configuration process may cause unexpected behavior on the CONF_DONE signal.

Active Serial Configuration

The AS configuration scheme supports AS x1 (1-bit data width) and AS x4 (4-bit data width) modes. The AS x4 mode provides four times faster configuration time than the AS x1 mode. In the AS configuration scheme, the Stratix V device controls the configuration interface.

In the AS configuration process, after power-up, the Stratix V device drives the DCLK pin with the default 12.5 MHz internal oscillator to read the configuration bitstream from the serial flash. The device determines the configuration options such as clock source, DCLK frequency, ASx1, or ASx4 by reading the option bits, located from 0x80 to 0x127, at the start of the programming file stored in the serial flash.

After the option bit is decoded by the Stratix V configuration control block, the AS configuration continues with the design option by reading the rest of the programming file from the serial flash until CONF_DONE pin asserts high and eventually enters user mode. When any interruption (such as data corruption) occurs in the middle of the AS Configuration, the nSTATUS pin will assert low to indicate configuration error, and deassert high to restart the configuration process. If there is no image in the serial flash or if the image is corrupted, you will observe the nSTATUS will pulse low repeatedly as the control block tries to configure itself forever, until the configuration is successful.

Related Information

[Stratix V Device Datasheet](#)

Provides more information about the AS configuration timing.

DATA Clock (DCLK)

Stratix V devices generate the serial clock, `DCLK`, that provides timing to the serial interface. In the AS configuration scheme, Stratix V devices drive control signals on the falling edge of `DCLK` and latch the configuration data on the following falling edge of this clock pin.

The maximum `DCLK` frequency supported by the AS configuration scheme is 100 MHz except for the AS multi-device configuration scheme. You can source `DCLK` using `CLKUSR` or the internal oscillator. If you use the internal oscillator, you can choose a 12.5, 25, 50, or 100 MHz clock under the **Device and Pin Options** dialog box, in the **Configuration** page of the Intel Quartus Prime software.

After power-up, `DCLK` is driven by a 12.5 MHz internal oscillator by default. The Stratix V device determines the clock source and frequency to use by reading the option bit in the programming file.

Related Information

[Stratix V Device Datasheet](#)

Provides more information about the `DCLK` frequency specification in the AS configuration scheme.

Active Serial Single-Device Configuration

To configure a Stratix V device, connect the device to a serial configuration (EPCS) device or quad-serial configuration (EPCQ) device, as shown in the following figures.

Figure 8-9: Single Device AS x1 Mode Configuration

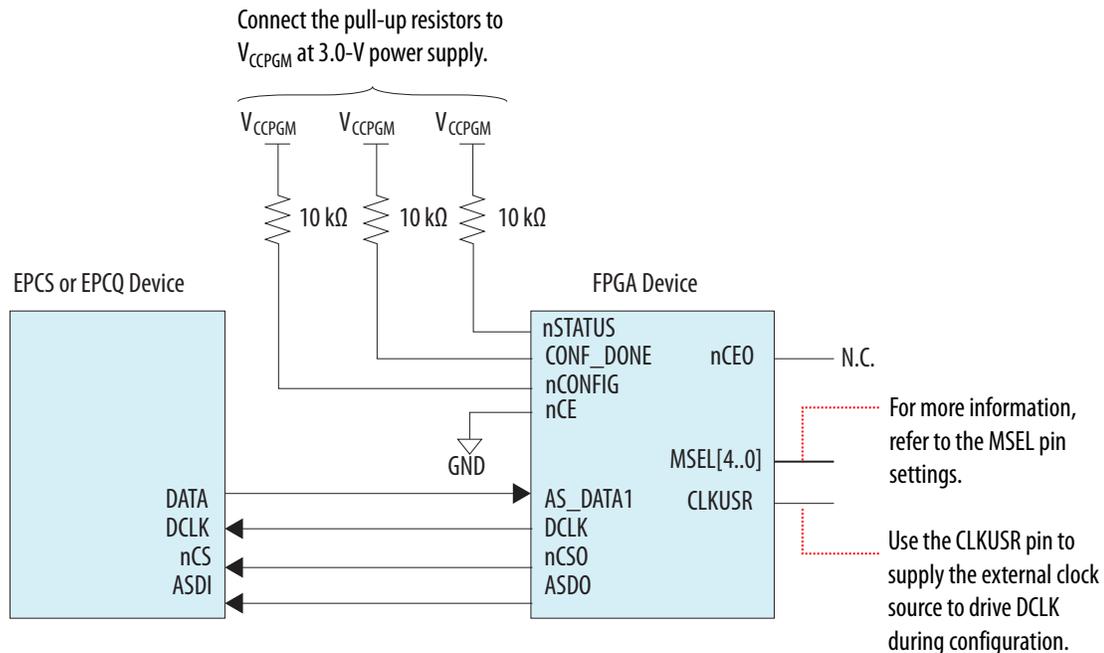
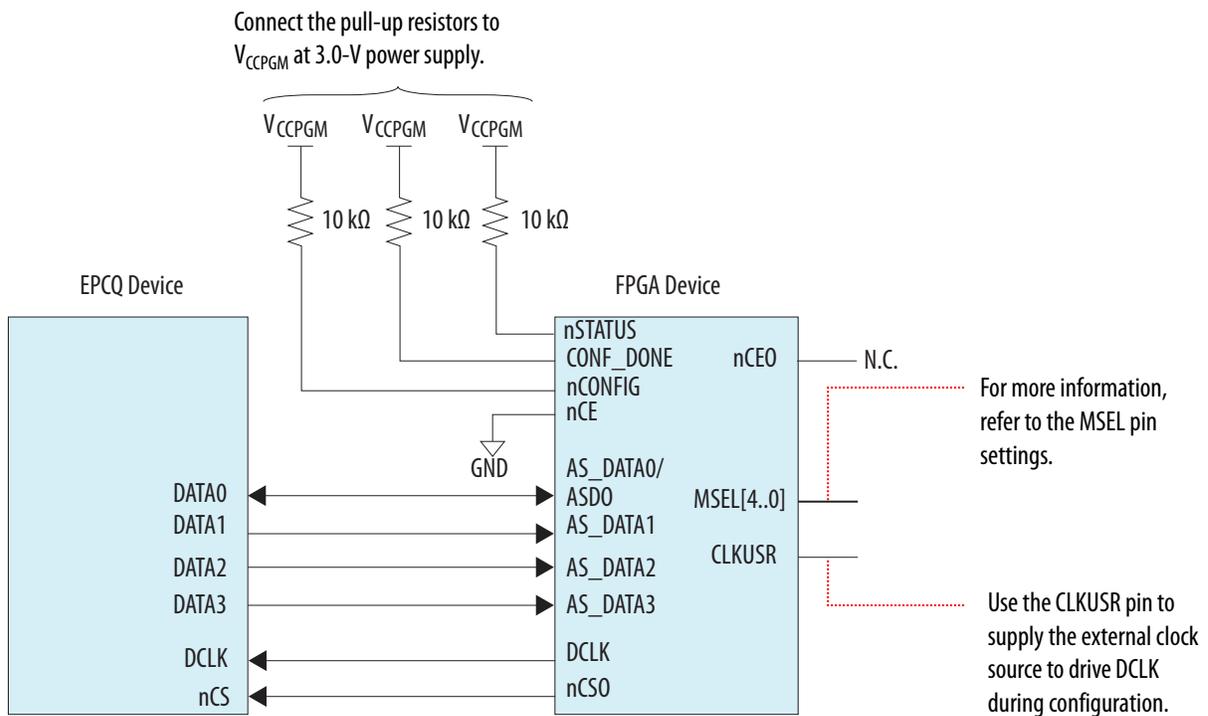


Figure 8-10: Single Device AS x4 Mode Configuration



Active Serial Multi-Device Configuration

You can configure multiple Stratix V devices that are connected to a chain. Only AS x1 mode supports multi-device configuration.

The first device in the chain is the configuration master. Subsequent devices in the chain are configuration slaves.

Note: The AS multi-device configuration scheme does not support 100 MHz DCLK frequency.

Pin Connections and Guidelines

Observe the following pin connections and guidelines for this configuration setup:

- Hardwire the `MSEL` pins of the first device in the chain to select the AS configuration scheme. For subsequent devices in the chain, hardwire their `MSEL` pins to select the PS configuration scheme. Any other Altera® devices that support the PS configuration can also be part of the chain as a configuration slave.
- Tie the following pins of all devices in the chain together:
 - `nCONFIG`
 - `nSTATUS`
 - `DCLK`
 - `DATA[]`
 - `CONF_DONE`

By tying the `CONF_DONE`, `nSTATUS`, and `nCONFIG` pins together, the devices initialize and enter user mode at the same time. If any device in the chain detects an error, configuration stops for the entire chain and you must reconfigure all the devices. For example, if the first device in the chain flags an error on the `nSTATUS` pin, it resets the chain by pulling its `nSTATUS` pin low.

- Ensure that `DCLK` and `DATA[]` are buffered every fourth device to prevent signal integrity and clock skew problems.

Using Multiple Configuration Data

To configure multiple Stratix V devices in a chain using multiple configuration data, connect the devices to an EPCS or EPCQ device, as shown in the following figure.

Related Information

- [Serial Configuration \(EPCS\) Devices Datasheet](#)
- [Quad-Serial Configuration \(EPCQ\) Devices Datasheet](#)

Controlling EPCS and EPCQ Devices

During configuration, Stratix V devices enable the EPCS or EPCQ device by driving its `nCS0` output pin low, which connects to the chip select (`nCS`) pin of the EPCS or EPCQ device. Stratix V devices use the `DCLK` and `ASDO` pins to send operation commands and read address signals to the EPCS or EPCQ device. The EPCS or EPCQ device provides data on its serial data output (`DATA[]`) pin, which connects to the `AS_DATA[]` input of the Stratix V devices.

Note: 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.

Trace Length and Loading Guideline

The maximum trace length and loading apply to both single- and multi-device AS configuration setups as listed in the following table. The trace length is the length from the Stratix V device to the EPCS or EPCQ device.

Table 8-9: Maximum Trace Length and Loading Guideline for AS x1 and x4 Configurations for Stratix V Devices

Stratix V Device AS Pins	Maximum Board Trace Length (Inches)		Maximum Board Load (pF)
	12.5/ 25/ 50 MHz	100 MHz	
DCLK	10	6	5
DATA[3 . . 0]	10	6	10
nCS0	10	6	10

Programming EPCS and EPCQ Devices

You can program EPCS and EPCQ devices in-system using a USB-Blaster™, EthernetBlaster, EthernetBlaster II, or ByteBlaster™ II download cable. Alternatively, you can program the EPCS or EPCQ using a microprocessor with the SRrunner software driver.

In-system programming (ISP) offers you the option to program the EPCS or EPCQ 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 Intel Quartus Prime software or any supported third-party software. Using the JTAG interface, an Altera IP called the 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. This allows the EPCS or EPCQ to be programmed directly using the JTAG interface.

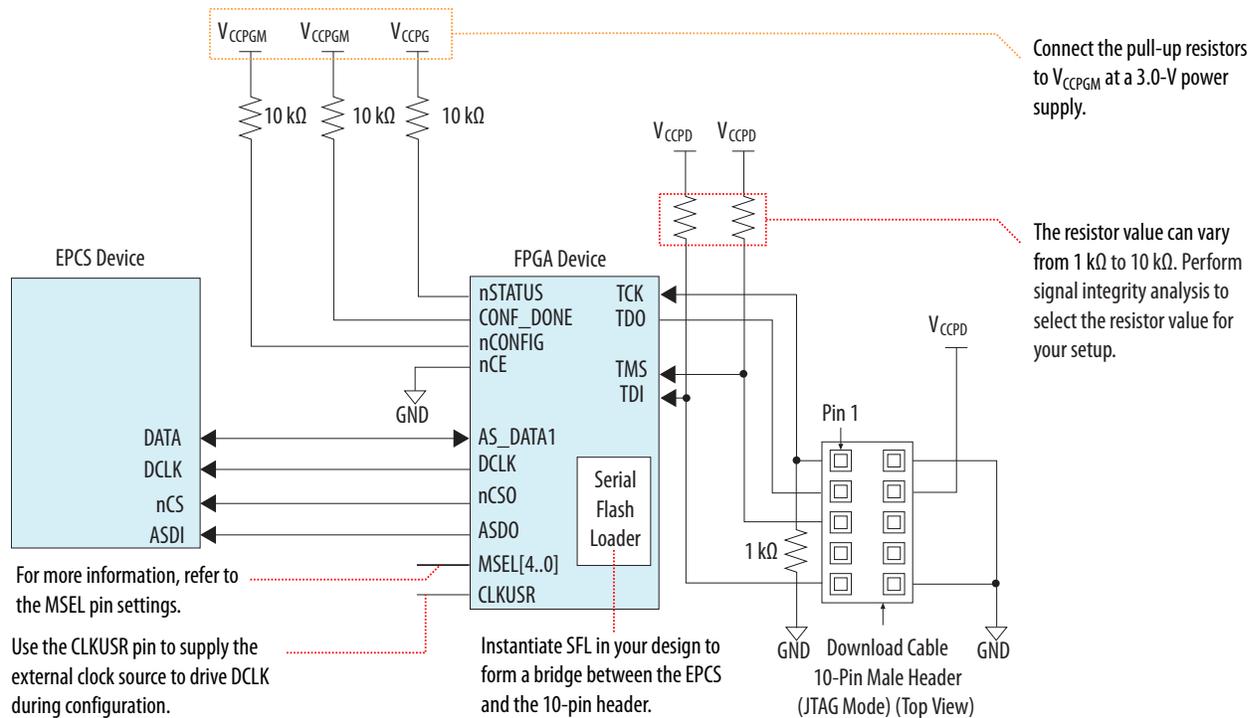
Related Information

- [AN 370: Using the Serial FlashLoader with the Quartus II Software](#)
- [AN 418: SRrunner: An Embedded Solution for Serial Configuration Device Programming](#)

Programming EPCS Using the JTAG Interface

To program an EPCS device using the JTAG interface, connect the device as shown in the following figure.

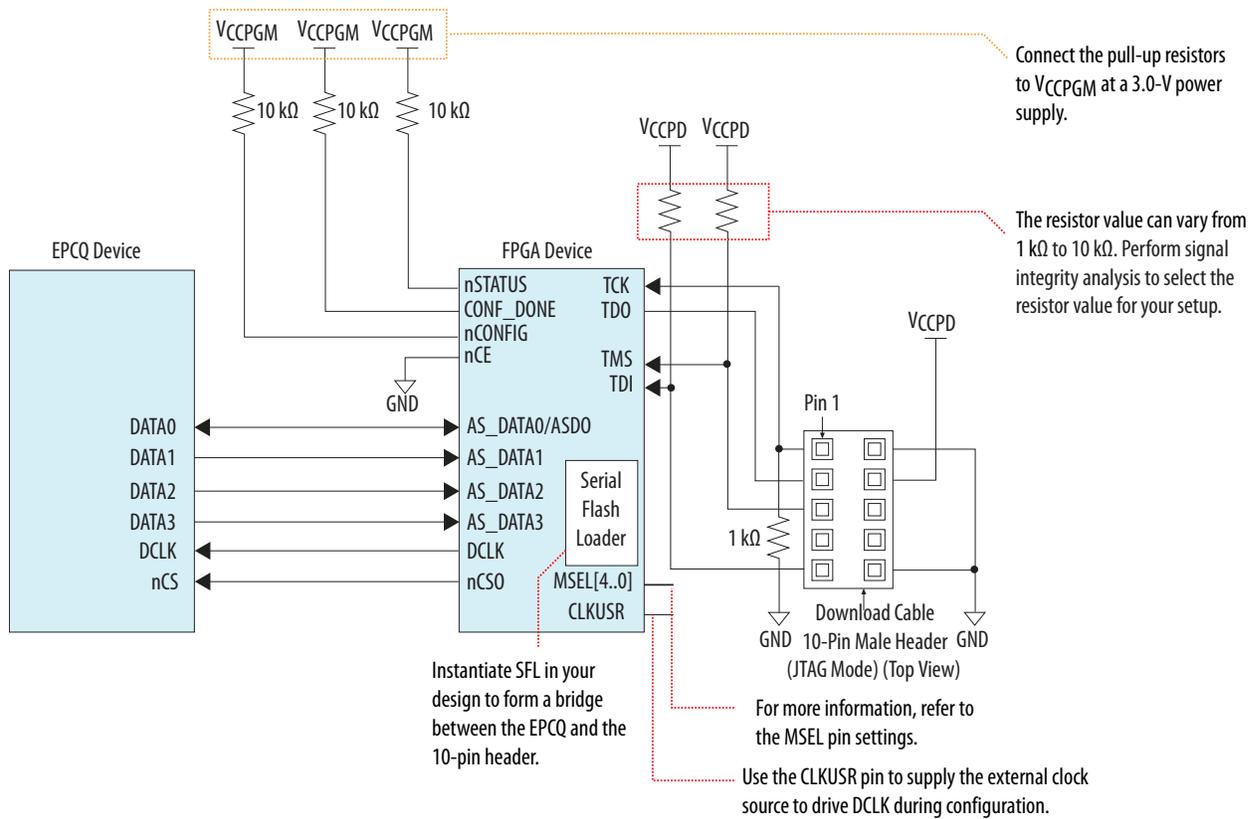
Figure 8-12: Connection Setup for Programming the EPCS Using the JTAG Interface



Programming EPCQ Using the JTAG Interface

To program an EPCQ device using the JTAG interface, connect the device as shown in the following figure.

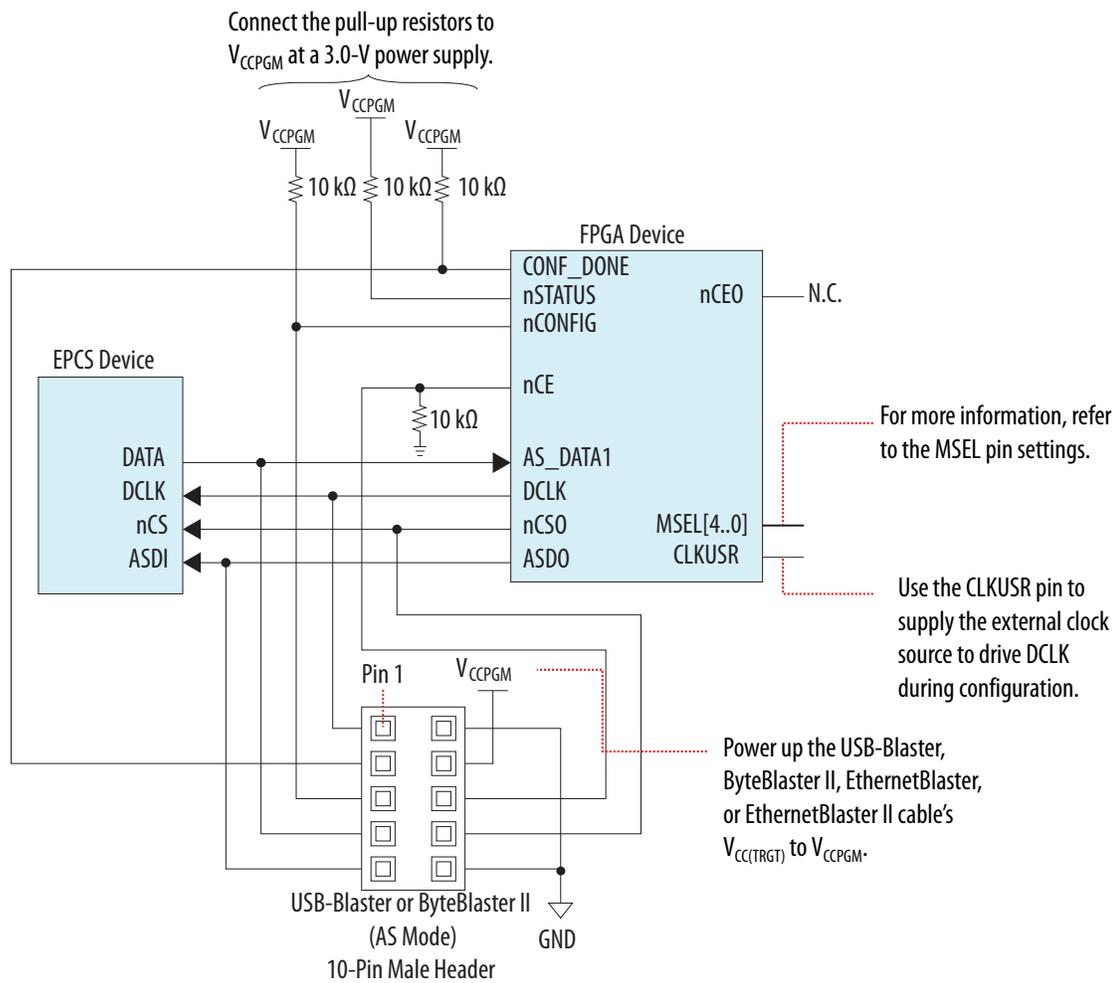
Figure 8-13: Connection Setup for Programming the EPCQ Using the JTAG Interface



Programming EPCS Using the Active Serial Interface

To program an EPCS device using the AS interface, connect the device as shown in the following figure.

Figure 8-14: Connection Setup for Programming the EPCS Using the AS Interface

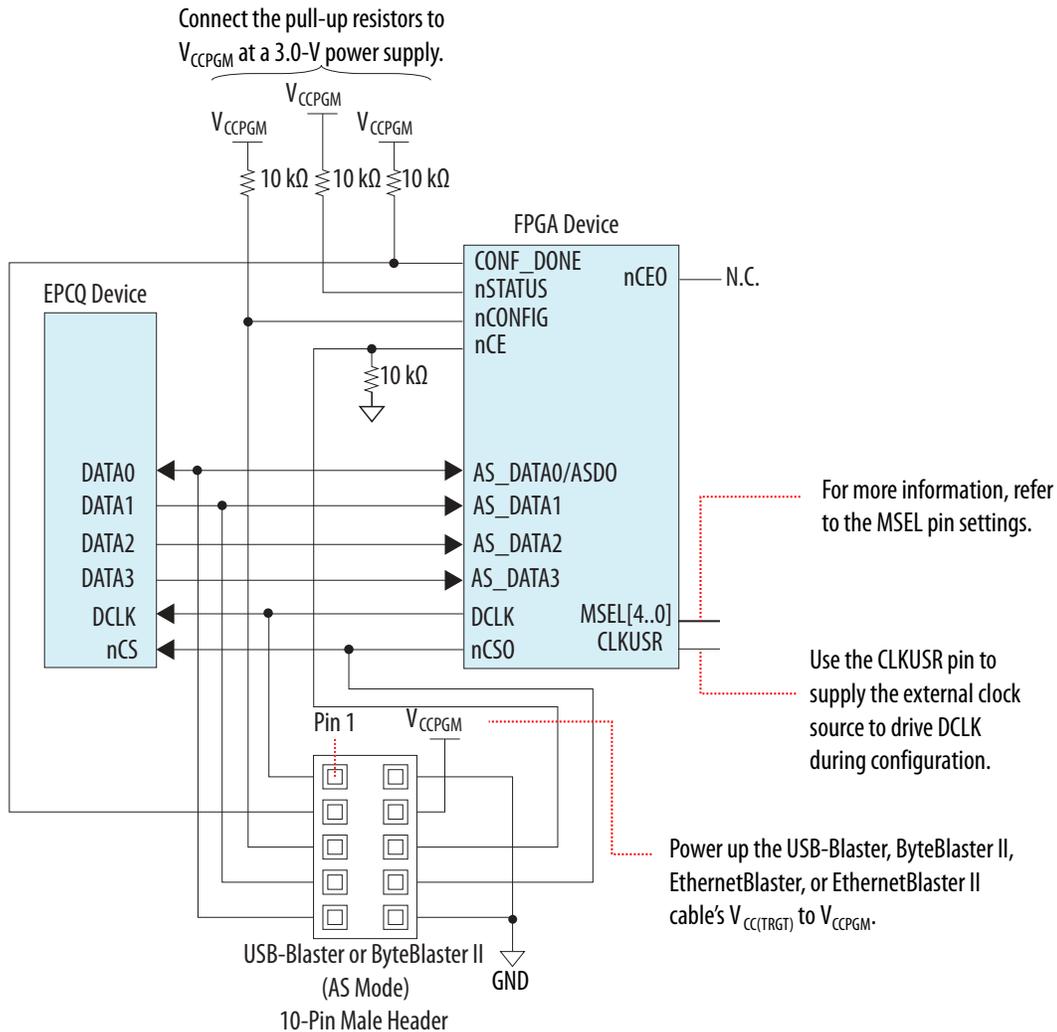


Programming EPCQ Using the Active Serial Interface

To program an EPCQ device using the AS interface, connect the device as shown in the following figure.

Figure 8-15: Connection Setup for Programming the EPCQ Using the AS Interface

Using the AS header, the programmer serially transmits the operation commands and configuration bits to the EPCQ on `DATA0`. This is equivalent to the programming operation for the EPCS.



When programming the EPCS and EPCQ devices, the download cable disables 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 the 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 the EPCQ programming using the download cable, `DATA0` transfers the programming data, operation command, and address information from the download cable into the EPCQ. During the EPCQ verification using the download cable, `DATA1` transfers the programming data back to the download cable.

Passive Serial Configuration

The PS configuration scheme uses an external host. You can use a microprocessor, MAX II device, MAX V device, or a host PC as the external host.

You can use an external host to control the transfer of configuration data from an external storage such as flash memory to the FPGA. The design that controls the configuration process resides in the external host.

You can store the configuration data in Programmer Object File (**.pof**), **.rbf**, **.hex**, or **.ttf**. If you are using configuration data in **.rbf**, **.hex**, or **.ttf**, send the LSB of each data byte first. For example, if the **.rbf** 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.

You can use the PFL IP core with a MAX II or MAX V device to read configuration data from the flash memory device and configure the Stratix V device.

For a PC host, connect the PC to the device using a download cable such as the Altera USB-Blaster USB port, ByteBlaster II parallel port, EthernetBlaster, and EthernetBlaster II download cables.

The configuration data is shifted serially into the `DATA0` pin of the device.

If you are using the Intel Quartus Prime programmer and the `CLKUSR` pin is enabled, you do not need to provide a clock source for the pin to initialize your device.

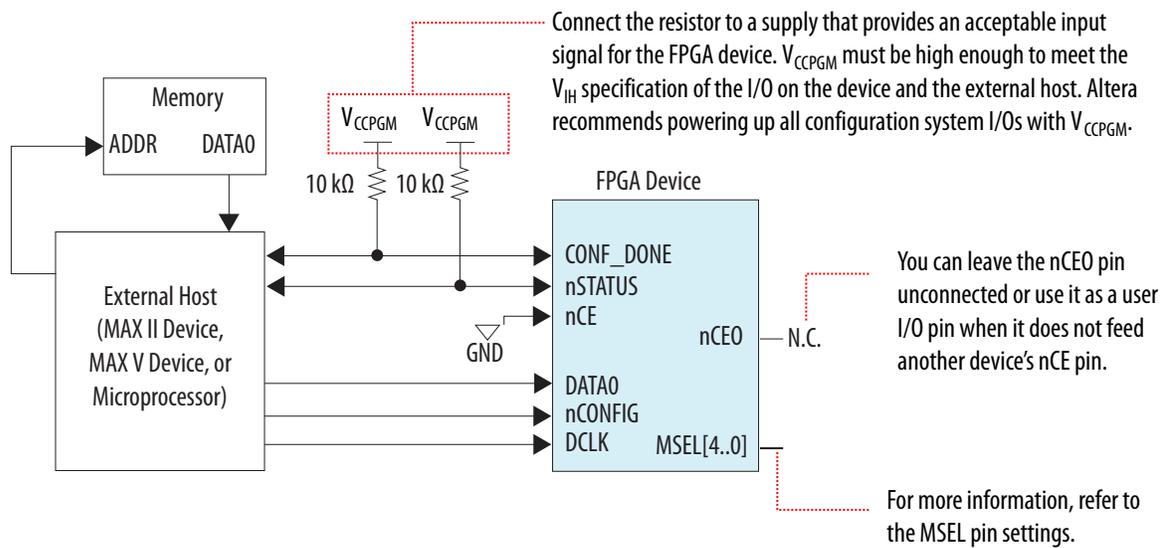
Related Information

- [Parallel Flash Loader IP Core User Guide](#)
- [Stratix V Device Datasheet](#)
Provides more information about the PS configuration timing.

Passive Serial Single-Device Configuration Using an External Host

To configure a Stratix V device, connect the device to an external host, as shown in the following figure.

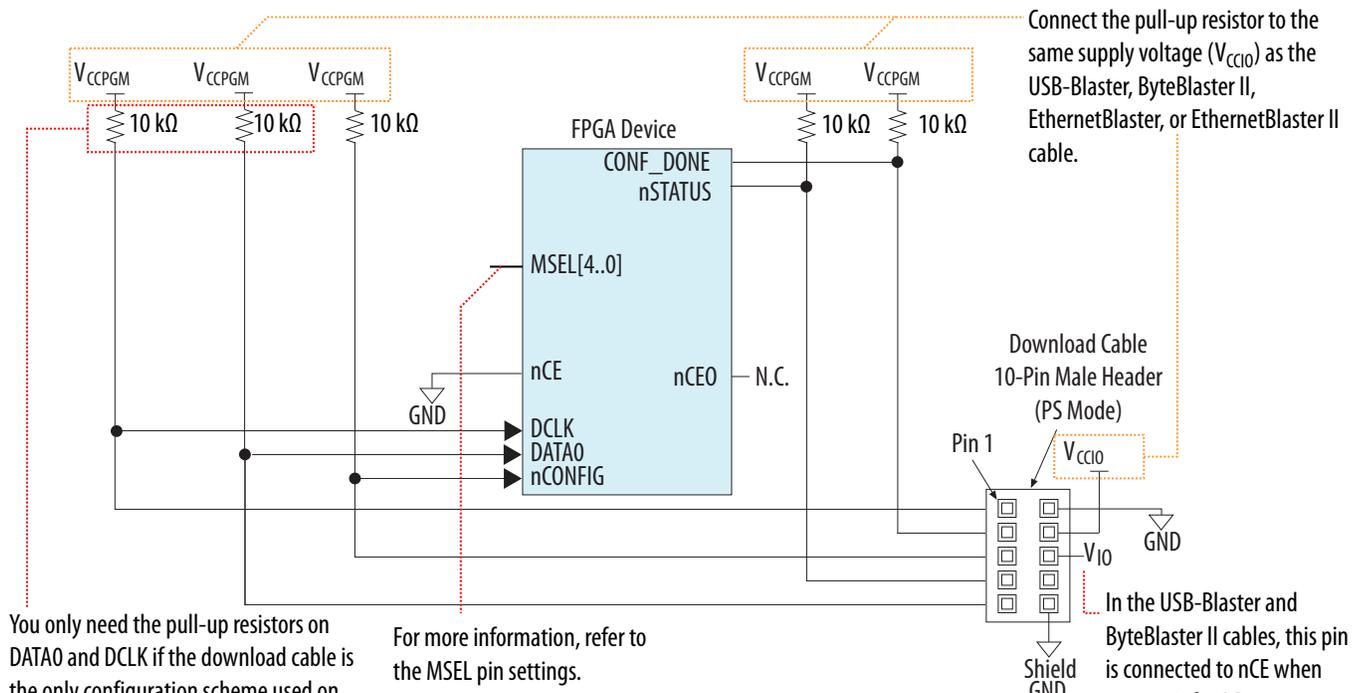
Figure 8-16: Single Device PS Configuration Using an External Host



Passive Serial Single-Device Configuration Using an Altera Download Cable

To configure a Stratix V device, connect the device to a download cable, as shown in the following figure.

Figure 8-17: Single Device PS Configuration Using an Altera Download Cable



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, MAX V device, or microprocessor, you do not need the pull-up resistors on DATA0 and DCLK.

For more information, refer to the MSEL pin settings.

Connect the pull-up resistor to the same supply voltage (V_{CCIO}) as the USB-Blaster, ByteBlaster II, EthernetBlaster, or EthernetBlaster II cable.

In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when you use it for AS programming. Otherwise, this pin is a no connect.

Passive Serial Multi-Device Configuration

You can configure multiple Stratix V devices that are connected in a chain.

Pin Connections and Guidelines

Observe the following pin connections and guidelines for this configuration setup:

- Tie the following pins of all devices in the chain together:
 - nCONFIG
 - nSTATUS
 - DCLK
 - DATA0
 - CONF_DONE

By tying the CONF_DONE and nSTATUS pins together, the devices initialize and enter user mode at the same time. If any device in the chain detects an error, configuration stops for the entire chain and you must reconfigure all the devices. For example, if the first device in the chain flags an error on the nSTATUS pin, it resets the chain by pulling its nSTATUS pin low.

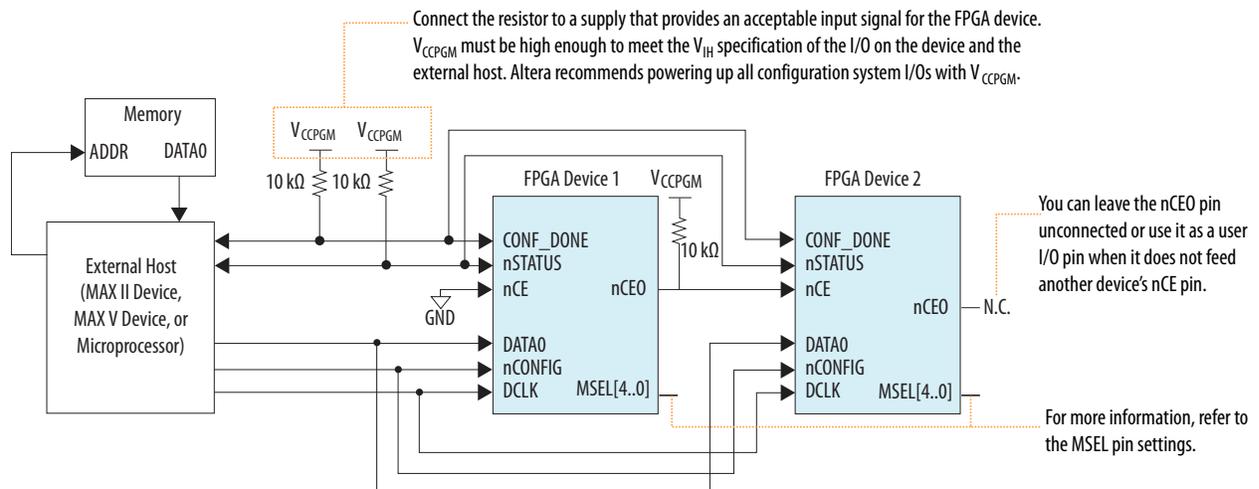
- If you are configuring the devices in the chain using the same configuration data, the devices must be of the same package and density.

Using Multiple Configuration Data

To configure multiple Stratix V devices in a chain using multiple configuration data, connect the devices to the external host as shown in the following figure.

Note: By default, the nCEO pin is disabled in the Intel Quartus Prime software. For the multi-device configuration chain, you must enable the nCEO pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

Figure 8-18: Multiple Device PS Configuration when Both Devices Receive Different Sets of Configuration Data



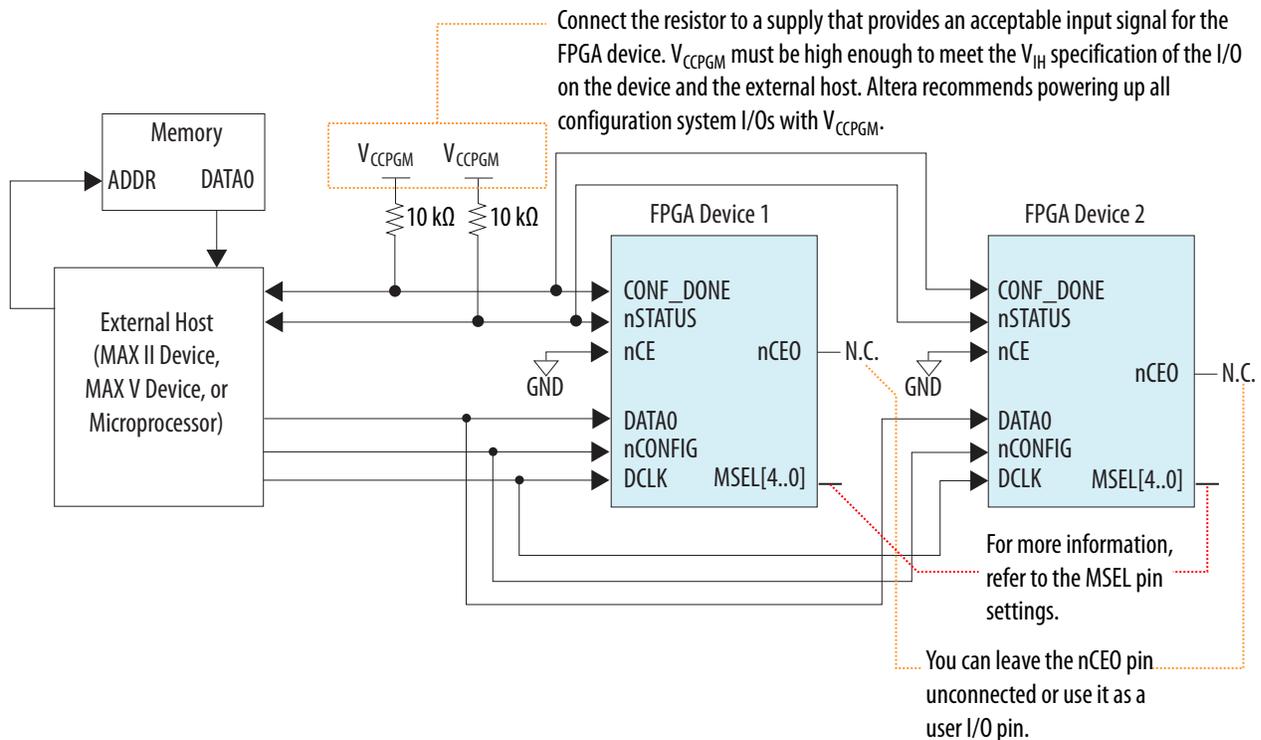
After a device completes configuration, its nCEO pin is released low to activate the nCE pin of the next device in the chain. Configuration automatically begins for the second device in one clock cycle.

Using One Configuration Data

To configure multiple Stratix V devices in a chain using one configuration data, connect the devices to an external host, as shown in the following figure.

Note: By default, the `nCEO` pin is disabled in the Intel Quartus Prime software. For the multi-device configuration chain, you must enable the `nCEO` pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

Figure 8-19: Multiple Device PS Configuration When Both Devices Receive the Same Set of Configuration Data



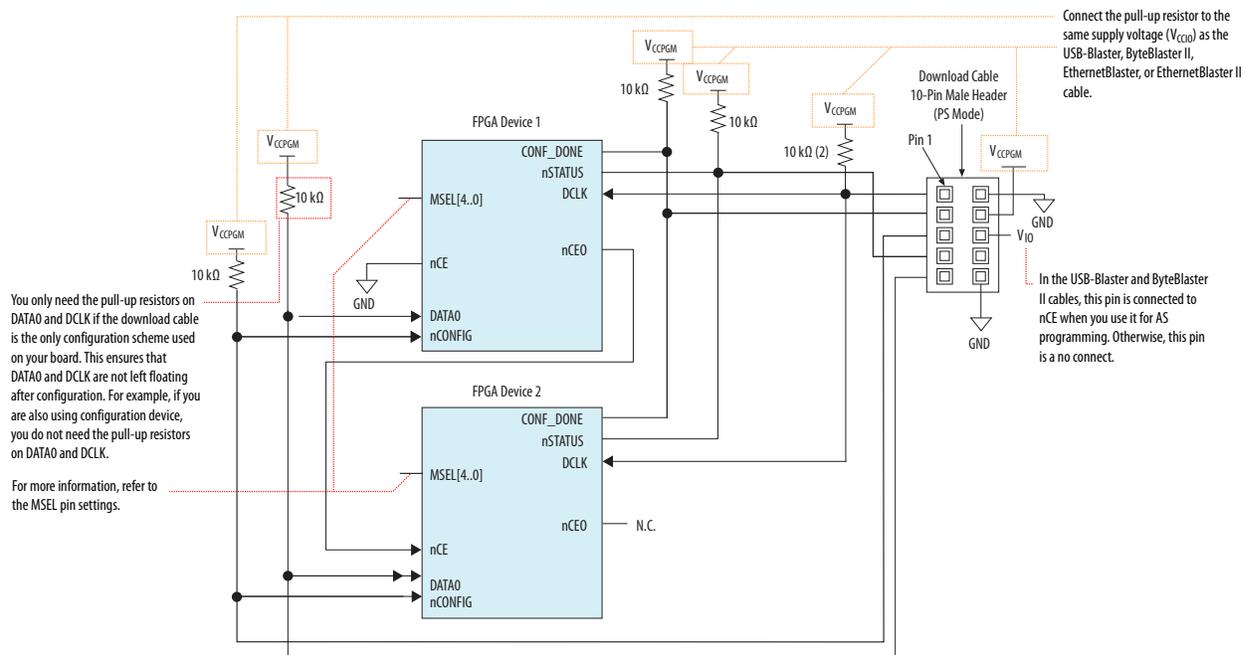
The `nCE` pins of the devices in the chain are connected to GND, allowing configuration for these devices to begin and end at the same time.

Using PC Host and Download Cable

To configure multiple Stratix V devices, connect the devices to a download cable, as shown in the following figure.

Note: By default, the `nCEO` pin is disabled in the Intel Quartus Prime software. For the multi-device configuration chain, you must enable the `nCEO` pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

Figure 8-20: Multiple Device PS Configuration Using an Altera Download Cable



When a device completes configuration, its nCEO pin is released low to activate the nCE pin of the next device. Configuration automatically begins for the second device.

JTAG Configuration

In Stratix V devices, JTAG instructions take precedence over other configuration schemes.

The Intel Quartus Prime software generates an SRAM Object File (.sof) that you can use for JTAG configuration using a download cable in the Intel Quartus Prime software programmer. Alternatively, you can use the JRunner software with .rbf or a JAM™ Standard Test and Programming Language (STAPL) Format File (.jam) or JAM Byte Code File (.jbc) with other third-party programmer tools.

Note: You cannot use the Stratix V decompression or design security features if you are configuring your Stratix V device using JTAG-based configuration.

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.

Related Information

- [JTAG Boundary-Scan Testing in Stratix V Devices](#) on page 10-1
Provides more information about JTAG boundary-scan testing.
- [Device Configuration Pins](#) on page 8-11
Provides more information about JTAG configuration pins.
- [JTAG Secure Mode](#) on page 8-46
- [AN 425: Using the Command-Line Jam STAPL Solution for Device Programming](#)

- [Stratix V Device Datasheet](#)
Provides more information about the JTAG configuration timing.
- [Programming Support for Jam STAPL Language](#)
- [USB-Blaster Download Cable User Guide](#)
- [ByteBlaster II Download Cable User Guide](#)
- [EthernetBlaster Communications Cable User Guide](#)
- [EthernetBlaster II Communications Cable User Guide](#)

JTAG Single-Device Configuration

To configure a single device in a JTAG chain, the programming software sets the other devices to the bypass mode. A device in a bypass mode transfers the programming data from the TDI pin to the TDO pin through a single bypass register. The configuration data is available on the TDO pin one clock cycle later.

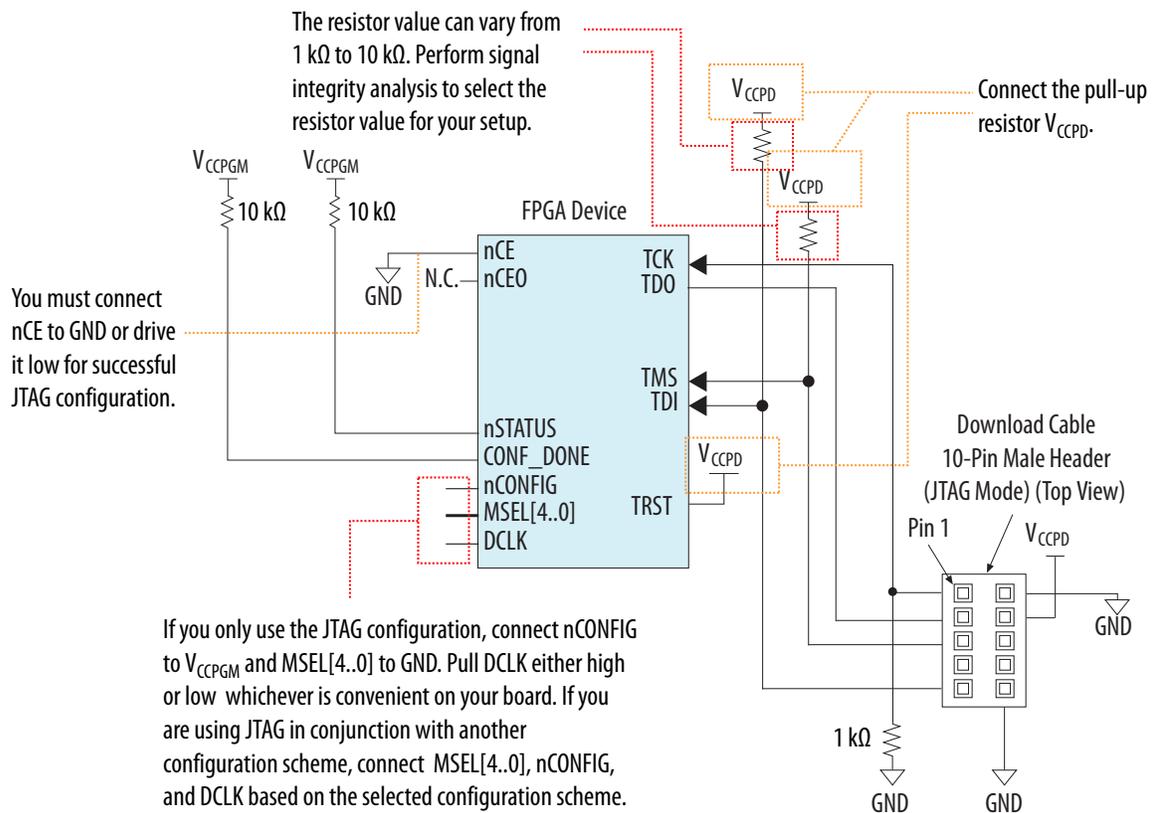
The Intel Quartus Prime software can use the CONF_DONE pin to verify the completion of the configuration process through the JTAG port:

- CONF_DONE pin is low—indicates that configuration has failed.
- CONF_DONE pin is high—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.

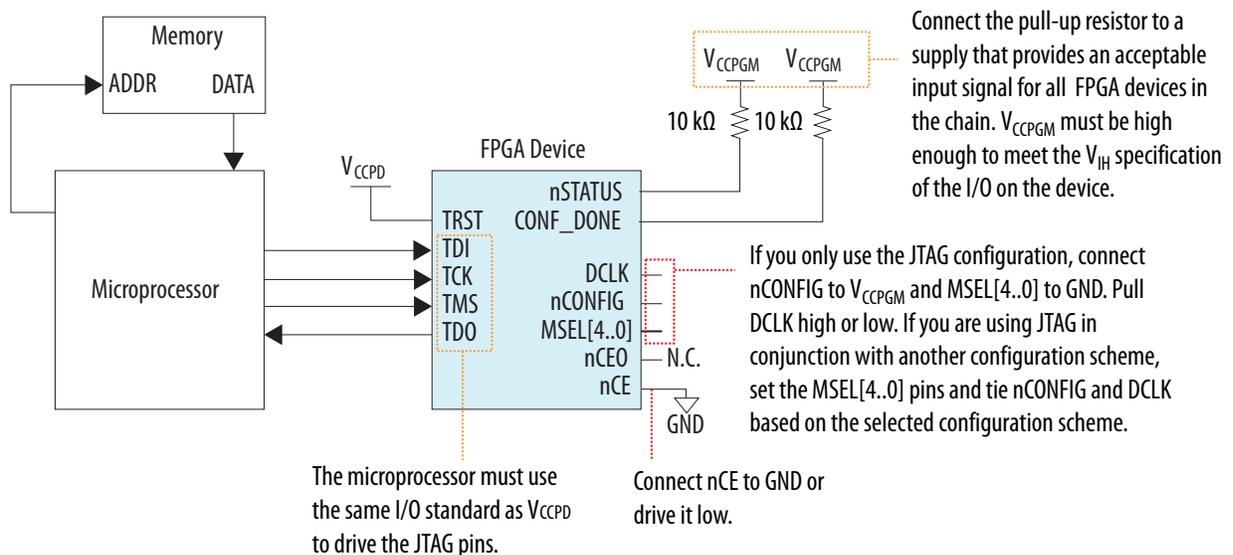
To configure a Stratix V device using a download cable, connect the device as shown in the following figure.

Figure 8-21: JTAG Configuration of a Single Device Using a Download Cable



To configure Stratix V device using a microprocessor, connect the device as shown in the following figure. You can use JRunner as your software driver.

Figure 8-22: JTAG Configuration of a Single Device Using a Microprocessor



Related Information

[AN 414: The JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration](#)

JTAG Multi-Device Configuration

You can configure multiple devices in a JTAG chain.

Pin Connections and Guidelines

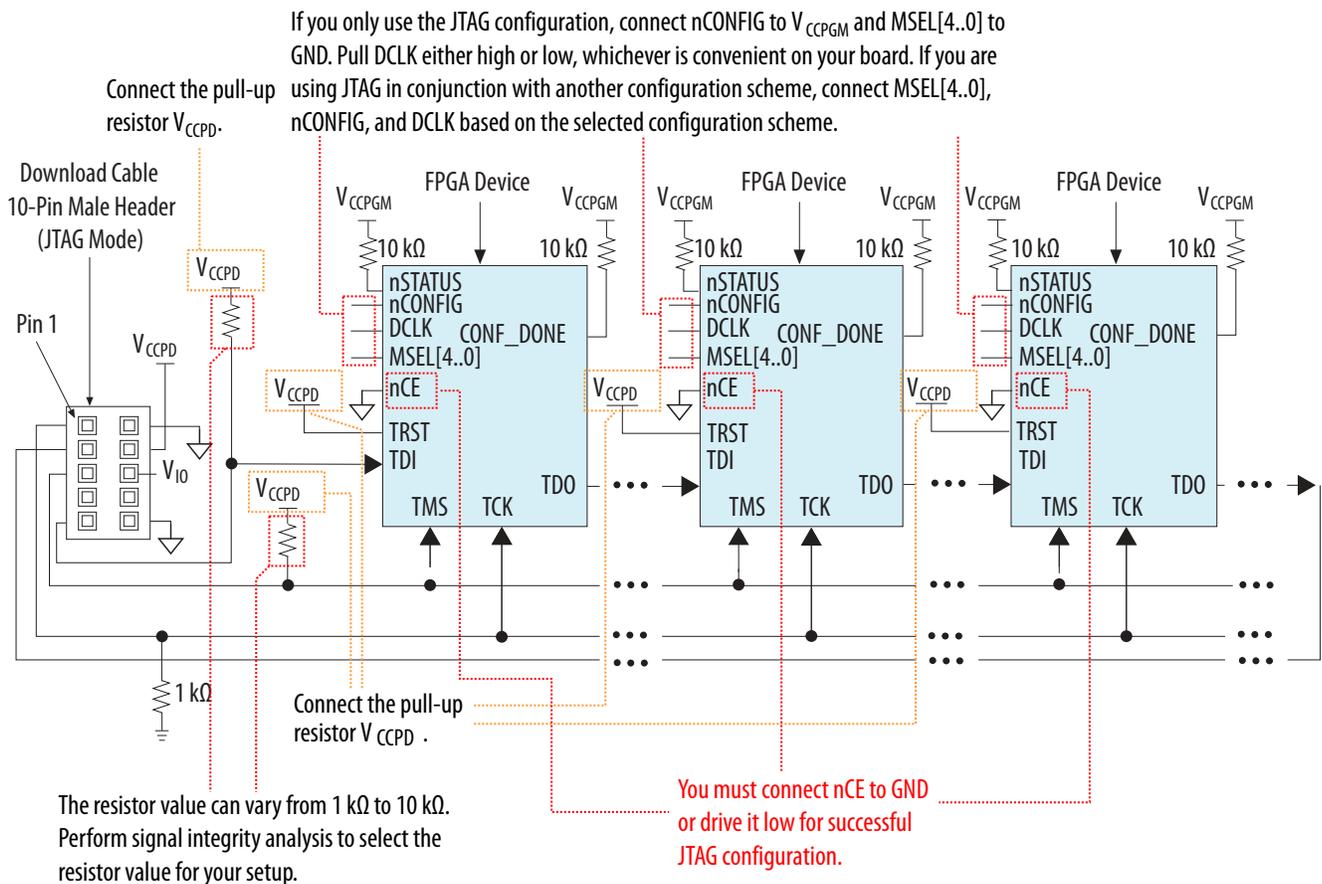
Observe the following pin connections and guidelines for this configuration setup:

- Isolate the `CONF_DONE` and `nSTATUS` pins to allow each device to enter user mode independently.
- One JTAG-compatible header is connected to several devices in a JTAG chain. The number of devices in the chain is limited only by the drive capability of the download cable.
- If you have four or more devices in a JTAG chain, buffer the `TCK`, `TDI`, and `TMS` pins with an on-board buffer. You can also connect other Altera devices with JTAG support to the chain.
- JTAG-chain device programming is ideal when the system contains multiple devices or when testing your system using the JTAG boundary-scan testing (BST) circuitry.

Using a Download Cable

The following figure shows a multi-device JTAG configuration.

Figure 8-23: JTAG Configuration of Multiple Devices Using a Download Cable



Related Information**[AN 656: Combining Multiple Configuration Schemes](#)**

Provides more information about combining JTAG configuration with other configuration schemes.

CONFIG_IO JTAG Instruction

The `CONFIO_IO` JTAG instruction allows you to configure the I/O buffers using the JTAG port before or during device configuration. When you issue this instruction, it interrupts configuration and allows you to issue all JTAG instructions. Otherwise, you can only issue the `BYPASS`, `IDCODE`, and `SAMPLE` JTAG instructions.

You can use the `CONFIO_IO` JTAG instruction to interrupt configuration and perform board-level testing. After the board-level testing is completed, you must reconfigure your device. Use the following methods to reconfigure your device:

- JTAG interface—issue the `PULSE_NCONFIG` JTAG instruction.
- FPP, PS, or AS configuration scheme—pulse the `nCONFIG` pin low.

Configuration Data Compression

Stratix V devices can receive compressed configuration bitstream and decompress the data in real-time during configuration. Preliminary data indicates that compression typically reduces the configuration file size by 30% to 55% depending on the design.

Decompression is supported in all configuration schemes except the JTAG configuration scheme.

You can enable compression before or after design compilation.

Enabling Compression Before Design Compilation

To enable compression before design compilation, follow these steps:

1. On the Assignment Menu, click **Device**.
2. Select your Stratix V device and then click **Device and Pin Options**.
3. In the **Device and Pin Options** window, select **Configuration** under the **Category** list and turn on **Generate compressed bitstreams**.

Enabling Compression After Design Compilation

To enable compression after design compilation, follow these steps:

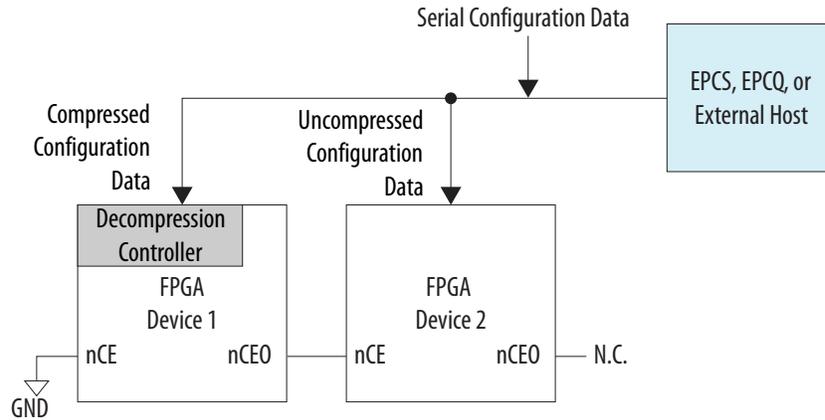
1. On the File menu, click **Convert Programming Files**.
2. Select the programming file type (**.pof**, **.sof**, **.hex**, **.hexout**, **.rbf**, or **.ttf**). For POF output files, select a configuration device.
3. Under the **Input files to convert** list, select **SOF Data**.
4. Click **Add File** and select a Stratix V device **.sof**.
5. Select the name of the file you added to the **SOF Data** area and click **Properties**.
6. Turn on the **Compression** check box.

Using Compression in Multi-Device Configuration

The following figure shows a chain of two Stratix V devices. Compression is only enabled for the first device.

This setup is supported by the AS or PS multi-device configuration only.

Figure 8-24: Compressed and Uncompressed Serial Configuration Data in the Same Configuration File

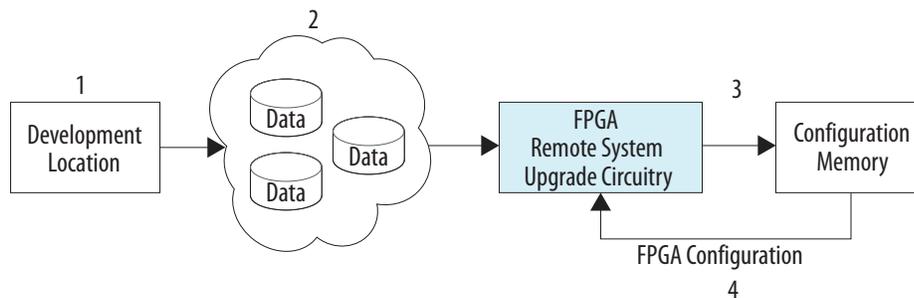


For the FPP configuration scheme, a combination of compressed and uncompressed configuration in the same multi-device configuration chain is not allowed because of the difference on the `DCLK-to-DATA[]` ratio.

Remote System Upgrades

Stratix V devices contain dedicated remote system upgrade circuitry. You can use this feature to upgrade your system from a remote location.

Figure 8-25: Stratix V Remote System Upgrade Block Diagram



You can design your system to manage remote upgrades of the application configuration images in the configuration device. The following list is the sequence of the remote system upgrade:

1. The logic (embedded processor or user logic) in the Stratix V device receives a configuration image from a remote location. You can connect the device to the remote source using communication protocols such as TCP/IP, PCI, user datagram protocol (UDP), UART, or a proprietary interface.
2. The logic stores the configuration image in non-volatile configuration memory.
3. The logic starts reconfiguration cycle using the newly received configuration image.
4. When an error occurs, the circuitry detects the error, reverts to a safe configuration image, and provides error status to your design.

Configuration Images

Each Stratix V device in your system requires one factory image. The factory image is a user-defined configuration image that contains logic to perform the following:

- Processes errors based on the status provided by the dedicated remote system upgrade circuitry.
- Communicates with the remote host, receives new application images, and stores the images in the local non-volatile memory device.
- Determines the application image to load into the Stratix V device.
- Enables or disables the user watchdog timer and loads its time-out value.
- Instructs the dedicated remote system upgrade circuitry to start a reconfiguration cycle.

You can also create one or more application images for the device. An application image contains selected functionalities to be implemented in the target device.

Store the images at the following locations in the EPCS or EPCQ devices:

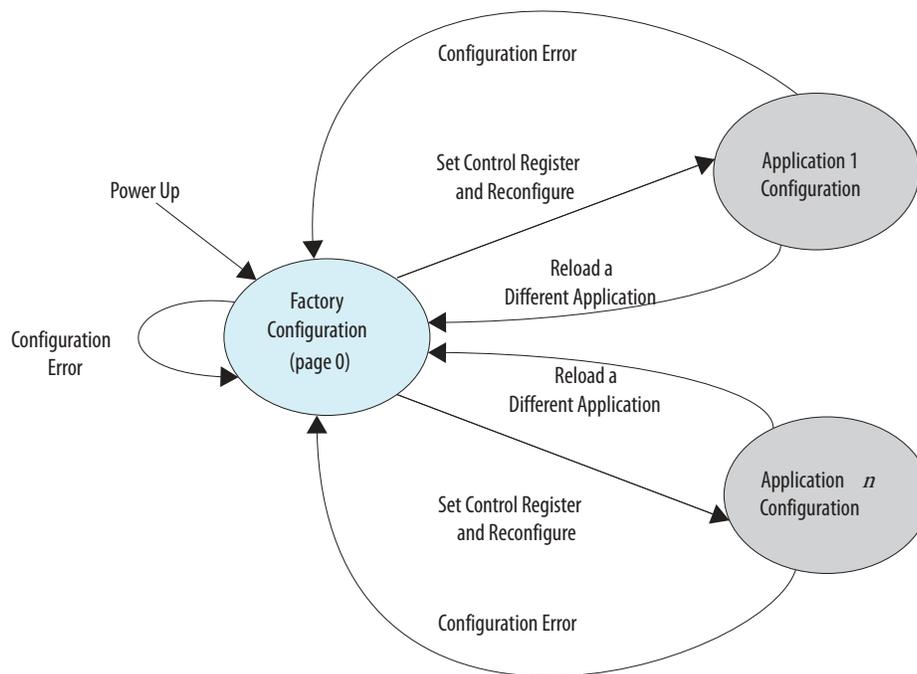
- Factory configuration image—`PGM[23..0] = 24'h000000` start address on the EPCS or EPCQ device.
- Application configuration image—any sector boundary. Altera recommends that you store only one image at one sector boundary.

When you are using EPCQ 256, ensure that the application configuration image address granularity is `32'h00000100`. The granularity requirement is having the most significant 24 bits of the 32 bits start address written to `PGM[23..0]` bits.

Note: If you are not using the Intel Quartus Prime software or SRRunner software for EPCQ 256 programming, put your EPCQ 256 device into four-byte addressing mode before you program and configure your device.

Configuration Sequence in the Remote Update Mode

Figure 8-26: Transitions Between Factory and Application Configurations in Remote Update Mode



Related Information

[Remote System Upgrade State Machine](#) on page 8-44

A detailed description of the configuration sequence in the remote update mode.

Remote System Upgrade Circuitry

The remote system upgrade circuitry contains the remote system upgrade registers, watchdog timer, and a state machine that controls these components.

Note: If you are using the Altera Remote Update IP core, the IP core controls the `RU_DOUT`, `RU_SHIFTnLD`, `RU_CAPTnUPDT`, `RU_CLK`, `RU_DIN`, `RU_nCONFIG`, and `RU_nRSTIMER` signals internally to perform all the related remote system upgrade operations.

Remote System Upgrade Registers

Table 8-10: Remote System Upgrade Registers

Register	Description
Shift	<p>Accessible by the logic array and clocked by <code>RU_CLK</code>.</p> <ul style="list-style-type: none"> Bits[4..0]—Contents of the status register are shifted into these bits. Bits[37..0]—Contents of the update and control registers are shifted into these bits.
Control	<p>This register is clocked by the 10-MHz internal oscillator. The contents of this register are shifted to the shift register for the user logic in the application configuration to read. When reconfiguration is triggered, this register is updated with the contents of the update register.</p>
Update	<p>This register is clocked by <code>RU_CLK</code>. The factory configuration updates this register by shifting data into the shift register and issuing an update. When reconfiguration is triggered, the contents of the update register are written to the control register.</p>
Status	<p>After each reconfiguration, the remote system upgrade circuitry updates this register to indicate the event that triggered the reconfiguration. This register is clocked by the 10-MHz internal oscillator.</p>

Related Information

- [Control Register](#) on page 8-43
- [Status Register](#) on page 8-44

Control Register

Table 8-11: Control Register Bits

Bit	Name	Reset Value ⁽¹⁷⁾	Description
0	AnF	1'b0	<p>Application not Factory bit. Indicates the configuration image type currently loaded in the device; 0 for factory image and 1 for application image. When this bit is 1, the access to the control register is limited to read only and the watchdog timer is enabled.</p> <p>Factory configuration design must set this bit to 1 before triggering reconfiguration using an application configuration image.</p>

⁽¹⁷⁾ This is the default value after the device exits POR and during reconfiguration back to the factory configuration image.

Bit	Name	Reset Value ⁽¹⁷⁾	Description
1..24	PGM[0..23]	24'h000000	Upper 24 bits of AS configuration start address (StAdd[31..8]), the 8 LSB are zero.
25	Wd_en	1'b0	User watchdog timer enable bit. Set this bit to 1 to enable the watchdog timer.
26..37	Wd_timer[11..0]	12'b000000000000	12-bit watchdog time-out value.

Status Register

Table 8-12: Status Register Bits

Bit	Name	Reset Value ⁽¹⁸⁾	Description
0	CRC	1'b0	When set to 1 , indicates CRC error during application configuration.
1	nSTATUS	1'b0	When set to 1 , indicates that nSTATUS is asserted by an external device due to error.
2	Core_nCONFIG	1'b0	When set to 1 , indicates that reconfiguration has been triggered by the logic array of the device.
3	nCONFIG	1'b0	When set to 1 , indicates that nCONFIG is asserted.
4	Wd	1'b0	When set to 1 , indicates that the user watchdog time-out.

Remote System Upgrade State Machine

The operation of the remote system upgrade state machine is as follows:

1. After power-up, the remote system upgrade registers are reset to **0** and the factory configuration image is loaded.
2. The user logic sets the AnF bit to **1** and the start address of the application image to be loaded. The user logic also writes the watchdog timer settings.
3. When the configuration reset (RU_CONFIG) goes low, the state machine updates the control register with the contents of the update register, and triggers reconfiguration using the application configuration image.
4. If error occurs, the state machine falls back to the factory image. The control and update registers are reset to **0**, and the status register is updated with the error information.
5. After successful reconfiguration, the system stays in the application configuration.

User Watchdog Timer

⁽¹⁷⁾ This is the default value after the device exits POR and during reconfiguration back to the factory configuration image.

⁽¹⁸⁾ After the device exits POR and power-up, the status register content is 5'b00000.

The user watchdog timer prevents a faulty application configuration from stalling the device indefinitely. You can use the timer to detect functional errors when an application configuration is successfully loaded into the device. The timer is automatically disabled in the factory configuration; enabled in the application configuration.

Note: If you do not want this feature in the application configuration, you need to turn off this feature by setting the `wd_en` bit to **1'b0** in the update register during factory configuration user mode operation. You cannot disable this feature in the application 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 user watchdog timer internal oscillator.

The timer begins counting as soon as the application configuration enters user mode. When the timer expires, the remote system upgrade circuitry generates a time-out signal, updates the status register, and triggers the loading of the factory configuration image. To reset the time, assert `RU_nRSTIMER`.

Related Information

[Stratix V Device Datasheet](#)

Provides more information about the operating range of the user watchdog internal oscillator's frequency.

Design Security

The Stratix V design security feature supports the following capabilities:

- Enhanced built-in advanced encryption standard (AES) decryption block to support 256-bit 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
- Limited accessible JTAG instruction during power-up in the JTAG secure mode
- 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 compression 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.
- Security against tampering—After you set the tamper protection bit, the Stratix V device can only accept configuration files encrypted with the same key. Additionally, programming through the JTAG interface and configuration interface is blocked.

When you use compression with the design security feature, the configuration file is first compressed and then encrypted using the Intel Quartus Prime software. During configuration, the device first decrypts and then decompresses the configuration file.

When you use design security with Stratix V devices in an FPP configuration scheme, it requires a different `DCLK-to-DATA[]` ratio.

Altera Unique Chip ID IP Core

The Altera Unique Chip ID IP core provides the following features:

- Acquiring the chip ID of an FPGA device.
- Allowing you to identify your device in your design as part of a security feature to protect your design from an unauthorized device.

Related Information

[Altera Unique Chip ID IP Core User Guide](#)

JTAG Secure Mode

When you enable the tamper-protection bit, Stratix V devices are in the JTAG secure mode after power-up. During this mode, many JTAG instructions are disabled. Stratix V devices only allow mandatory JTAG 1149.1 and 1149.6 instructions to be exercised. These JTAG 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 the `UNLOCK` instruction to deactivate the JTAG secure mode. You can issue the `LOCK` instruction to put the device back into JTAG secure mode. You can only issue both the `LOCK` and `UNLOCK` JTAG instructions during user mode.

Related Information

[Supported JTAG Instruction](#) on page 10-3

Provides more information about JTAG binary instruction code related to the `LOCK` and `UNLOCK` instructions.

Security Key Types

Stratix V devices offer two types of keys—volatile and non-volatile. The following table lists the differences between the volatile key and non-volatile keys.

Table 8-13: Security Key Types

Key Types	Key Programmability	Power Supply for Key Storage	Programming Method
Volatile	<ul style="list-style-type: none"> • Reprogrammable • Erasable 	Required external battery, V_{CCBAT} ⁽¹⁹⁾	On-board
Non-volatile	One-time programming	Does not require an external battery	On-board and in-socket programming ⁽²⁰⁾

⁽¹⁹⁾ V_{CCBAT} is a dedicated power supply for volatile key storage. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition.

⁽²⁰⁾ Third-party vendors offer in-socket programming.

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.

You can perform key programming through the JTAG pins interface. Ensure that the `nSTATUS` pin is released high before any key-programming attempts.

Note: To clear the volatile key, issue the `KEY_CLR_VREG` JTAG instruction. To verify the volatile key has been cleared, issue the `KEY_VERIFY` JTAG instruction.

Related Information

- [Supported JTAG Instruction](#) on page 10-3
Provides more information about the `KEY_CLR_VREG` and `KEY_VERIFY` instructions.
- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
Provides more information about the `VCCBAT` pin connection recommendations.
- [Stratix V GT Device Family Pin Connection Guidelines](#)
Provides more information about the `VCCBAT` pin connection recommendations.
- [Stratix V Device Datasheet](#)
Provides more information about battery specifications.

Security Modes

Table 8-14: Supported Security Modes

There is no impact to the configuration time required when compared with unencrypted configuration schemes except FPP with AES (and/or decompression), which requires a `DCLK` that is up to $\times 8$ the data rate.

Security Mode	Tamper Protection Bit Setting	Device Accepts Unencrypted File	Device Accepts Encrypted File	Security Level
No key	—	Yes	No	—
Volatile Key	—	Yes	Yes	Secure
Volatile Key with Tamper Protection Bit Set	Set	No	Yes	Secure with tamper resistant
Non-volatile Key	—	Yes	Yes	Secure
Non-volatile Key with Tamper Protection Bit Set	Set	No	Yes	Secure with tamper resistant

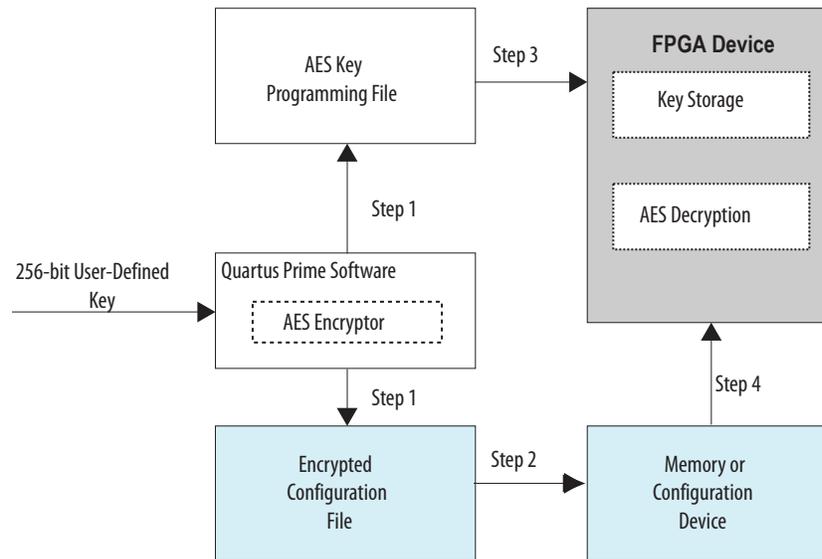
The use of unencrypted configuration bitstream in the volatile key and non-volatile key security modes is supported for board-level testing only.

Note: For the volatile key with tamper protection bit set security mode, Stratix V devices do not accept the encrypted configuration file if the volatile key is erased. If the volatile key is erased and you want to reprogram the key, you must use the volatile key security mode.

Enabling the tamper protection bit disables the test mode in Stratix V devices and disables programming through the JTAG interface. This process is irreversible and prevents Altera from carrying out failure analysis.

Design Security Implementation Steps

Figure 8-28: Design Security Implementation Steps



To carry out secure configuration, follow these steps:

1. The Intel Quartus Prime software generates the design security key programming file and encrypts the configuration data using the user-defined 256-bit security 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 the system power-up, the external memory device sends the encrypted configuration file to the Stratix V device.

Configuration, Design Security, and Remote System Upgrades in Stratix V Devices Revision History

Document Version	Changes
2020.07.24	Added topic: <i>I/O Standards and Drive Strength for Configuration Pins</i> .
2020.04.13	<ul style="list-style-type: none"> Removed topic: <i>I/O Standards and Drive Strength for Configuration Pins</i>. Updated the <i>User Mode</i> topic.

Document Version	Changes
2019.10.03	Added a note to <i>Device Configuration Pins</i> to state that the DCLK, AS_DATA0/ASDO, AS_DATA1, AS_DATA2, AS_DATA3, and nCSO pins have 25 kOhm pull-up resistors when the MSEL pins are set to AS configuration scheme.
2018.08.09	<ul style="list-style-type: none"> Updated the <i>Active Serial Configuration</i> topic. Updated Figure: <i>AS Configuration Timing Waveform</i>.

Date	Version	Changes
December 2017	2017.12.15	Added description in the <i>I/O Standards and Drive Strength for Configuration Pins</i> table.
December 2016	2016.12.09	<ul style="list-style-type: none"> Removed 1.8 V and 2.5 V V_{CCPGM} for AS mode in MSEL Pin Settings for Each Configuration Scheme of Stratix V Devices table. Added note to Power Up and Reset states in the Configuration Sequence for Stratix V Devices diagram. Changed the term "configuration mode" to "configuration scheme" when referring to a configuration scheme.
June 2016	2016.06.10	<ul style="list-style-type: none"> Added a note to specify the time between nCSO falling edge to first toggle of DCLK is more than 15ns in AS Configuration Timing figure.
December 2015	2015.12.21	<ul style="list-style-type: none"> Changed instances of <i>Quartus II</i> to <i>Quartus Prime</i>. Added the CVP_CONFDONE pin to the Configuration Pin Summary for Stratix V Devices table. Added the I/O Standards and Drive Strength for Configuration Pins table. Updated the voltage supported for the AS x1 and x4 configuration scheme in the MSEL Pin Settings for Each Configuration Scheme of Stratix V Devices table.
June 2015	2015.06.12	<ul style="list-style-type: none"> Added timing waveforms for FPP, AS, and PS configuration. Updated the Trace Length and Loading Guideline section. Updated data rate to x8 in the Supported Security Modes table.
January 2015	2015.01.23	Added the Transmitting Configuration Data section.

Date	Version	Changes
June 2014	2014.06.30	<ul style="list-style-type: none"> Updated Figure 8-17: JTAG Configuration of a Single Device Using a Download Cable. Updated Figure 8-19: JTAG Configuration of Multiple Devices Using a Download Cable. Updated the maximum clock rate for Partial Reconfiguration in Table 8-1. Updated the MSEL pin settings recommendation in the MSEL Pin Settings section.
January 2014	2014.01.10	<ul style="list-style-type: none"> Updated the Enabling Remote System Upgrade Circuitry section. Updated the Configuration Pin Summary section. Updated Figure 8-3, Figure 8-7, and Figure 8-14.
June 2013	2013.06.11	Updated the Configuration Error Handling section.
May 2013	2013.05.10	Removed support for active serial multi-device configuration using the same configuration data.
May 2013	2013.05.06	<ul style="list-style-type: none"> Added link to the known document issues in the Knowledge Base. Added the ALTCHIP_ID megafunction section. Added links for AS, PS, FPP, and JTAG configuration timing to device datasheet. Updated "Connection Setup for Programming the EPCS Using the JTAG Interface" and "Connection Setup for Programming the EPCQ Using the JTAG Interface" figures. Updated CvP support for partial reconfiguration in the Table 8-1: Configuration Modes and Features Supported by Stratix V Devices. Moved all links to the Related Information section of respective topics for easy reference.
March 2013	2013.03.04	Remove a note to the <code>nIO_PULLUP</code> pin in Table 8-3: Configuration Pin Summary for Stratix V Devices.
December 2012	2012.12.28	<ul style="list-style-type: none"> Added configuration modes and features for Stratix V devices. Reorganized content and updated template.
June 2012	1.7	<ul style="list-style-type: none"> Added MAX V devices. Updated Figure 9-2, Figure 9-3, Figure 9-11, Figure 9-16, Figure 9-17, Figure 9-20, and Figure 9-23. Updated Table 9-4, Table 9-5, Table 9-7, Table 9-11, and Table 9-12. Updated "MSEL Pin Settings" and "FPP Multi-Device Configuration" sections.
February 2012	1.6	<ul style="list-style-type: none"> Updated "Security Key Types" section. Updated Table 9-10.

Date	Version	Changes
December 2011	1.5	<ul style="list-style-type: none">• Updated "FPP Configuration Timing", "JTAG Secure Mode", and "Security Key Types" sections.• Updated Table 9-8.
November 2011	1.4	<ul style="list-style-type: none">• Updated Table 9-5, Table 9-9, and Table 9-14.• Updated Figure 9-8, Figure 9-9, and Figure 9-21.• Updated "AS Multi-Device Configuration" and "Active Serial Configuration (Serial Configuration Devices)" sections.
May 2011	1.3	<ul style="list-style-type: none">• Chapter moved to volume 2 for the 11.0 release.• Added "Remote System Upgrades 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	<ul style="list-style-type: none">• 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", "VCCPD 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.

2020.09.10

SV51011



Subscribe



Send Feedback

This chapter describes the error detection features in Stratix V devices. You can use these features to mitigate single event upset (SEU) or soft errors.

Related Information

[Stratix V Device Handbook: Known Issues](#)

Lists the planned updates to the *Stratix V Device Handbook* chapters.

Error Detection Features

The on-chip error detection CRC circuitry allows you to perform the following operations without any impact on the fitting or performance of the device:

- Auto-detection of CRC errors during configuration.
- Optional CRC error detection and identification in user mode.
- Optional internal scrubbing in user mode. When enabled, this feature corrects single-bit and double-adjacent errors automatically.
- Testing of error detection functions by deliberately injecting errors through the JTAG interface.

Configuration Error Detection

When the Intel Quartus Prime software generates the configuration bitstream, the software also computes a 16-bit CRC value for each frame. A configuration bitstream can contain more than one CRC values depending on the number of data frames in the bitstream. The length of the data frame varies for each device.

When a data frame is loaded into the FPGA during configuration, the precomputed CRC value shifts into the CRC circuitry. At the same time, the CRC engine in the FPGA computes the CRC value for the data frame and compares it against the precomputed CRC value. If both CRC values do not match, the `nSTATUS` pin is set to low to indicate a configuration error.

You can test the capability of this feature by modifying the configuration bitstream or intentionally corrupting the bitstream during configuration.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

User Mode Error Detection

In user mode, the contents of the configured CRAM bits may be affected by soft errors. These soft errors, which are caused by an ionizing particle, are not common in Altera devices. However, high-reliability applications that require the device to operate error-free may require that your designs account for these errors.

You can enable the error detection circuitry to detect soft errors. Each data frame stored in the CRAM contains a 32-bit precomputed CRC value. When this feature is enabled, the error detection circuitry continuously computes a 32-bit CRC value for each frame in the CRAM and compares the CRC value against the precomputed value.

- If the CRC values match, the 32-bit CRC signature in the `syndrome` register is set to zero to indicate that no error is detected.
- Otherwise, the resulting 32-bit CRC signature in the `syndrome` register is non-zero to indicate a CRC error. The `CRC_ERROR` pin is pulled high, and the error type and location are identified.

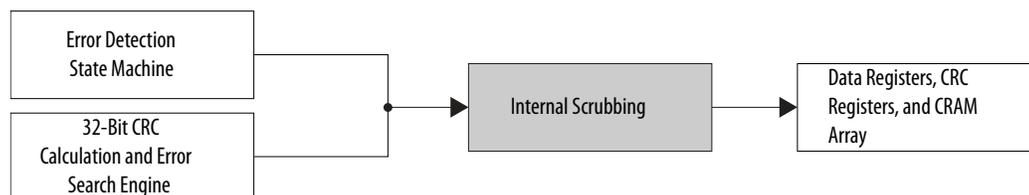
Within a frame, the error detection circuitry can detect all single-, double-, triple-, quadruple-, and quintuple-bit errors. When a single-bit or double-adjacent error is detected, the error detection circuitry reports the bit location and determines the error type for single-bit and double-adjacent errors. The probability of other error patterns is very low and the reporting of bit location is not guaranteed. The probability of more than five CRAM bits being flipped by soft errors is very low. In general, the probability of detection for all error patterns is 99.9999%. The process of error detection continues until the device is reset by setting the `nCONFIG` signal low.

Internal Scrubbing

Internal scrubbing is the ability to internally correct soft errors in user mode. This feature corrects single-bit and double-adjacent errors detected in each data frame without the need to reconfigure the device.

Note:

Figure 9-1: Block Diagram



Specifications

This section lists the EMR update interval, error detection frequencies, and CRC calculation time for error detection in user mode.

Minimum EMR Update Interval

The interval between each update of the error message register depends on the device and the frequency of the error detection clock. Using a lower clock frequency increases the interval time, hence increasing the time required to recover from a single event upset (SEU).

Table 9-1: Estimated Minimum EMR Update Interval in Stratix V Devices

Variant	Member Code	Package	Timing Interval (μ s)
Stratix V GX	A3	EH29-H780	3.13
		HF35-F1152	3.13
		KF35-F1152	3.13
		KF40-F1517/KH40-H1517	3.13
	A4	All	3.13
	A5		3.71
	A7		3.71
	A9		5.01
	AB		5.01
	B5		3.85
B6	3.85		
Stratix V GT	C5	All	3.71
	C7		3.71
Stratix V GS	D3	All	2.61
	D4	EH29-H780	2.61
		HF35-F1152	2.61
		KF40-F1517/KH40-H1517	3.13
	D5	All	3.13
	D6		4.33
D8	4.33		
Stratix V E	E9	All	5.01
	EB		5.01

Error Detection Frequency

You can control the speed of the error detection process by setting the division factor of the clock frequency in the Intel Quartus Prime software. The divisor is 2^n , where n can be any value listed in the following table.

The speed of the error detection process for each data frame is determined by the following equation:

Figure 9-2: Error Detection Frequency Equation

$$\text{Error Detection Frequency} = \frac{\text{Internal Oscillator Frequency}}{2^n}$$

Table 9-2: Error Detection Frequency Range for Stratix V Devices

The following table lists the frequencies and valid values of n .

Internal Oscillator Frequency	Error Detection Frequency		n	Divisor Range
	Maximum	Minimum		
100 MHz	100 MHz	390 kHz	0, 1, 2, 3, 4, 5, 6, 7, 8	1 – 256

CRC Calculation Time For Entire Device

While the CRC calculation is done on a per frame basis, it is important to know the time taken to complete CRC calculations for the entire device. The entire device detection time is the time taken to do CRC calculations on every frame in the device. This time depends on the device and the error detection clock frequency. The error detection clock frequency also depends on the device and on the internal oscillator frequency, which varies from 42.6 MHz to 100 MHz.

You can calculate the minimum and maximum time for any number of divisor based on the following formula:

$$\text{Maximum time } (n) = 2^{(n-8)} * t_{\text{MAX}}$$

$$\text{Minimum time } (n) = 2^n * t_{\text{MIN}}$$

where the range of n is from 0 to 8.

Table 9-3: Device EDCRC Detection Time in Stratix V Devices

The following table lists the minimum and maximum time taken to calculate the CRC value:

- The minimum time is derived using the maximum clock frequency with a divisor of 0.
- The maximum time is derived using the minimum clock frequency with a divisor of 8.

Variant	Member Code	Package	t_{MIN} (ms)	t_{MAX} (s)
Stratix V GX	A3	EH29-H780	38	19.42
		HF35-F1152	38	19.42
		KF35-F1152	38	19.42
		KF40-F1517/KH40-H1517	38	19.42
	A4	All	38	19.42
	A5		47	24.20
	A7		47	24.20
	A9		68	35.21
	AB		68	35.21
	B5		45	23.52
B6	45	23.52		
Stratix V GT	C5	All	47	24.20
	C7		47	24.20
Stratix V GS	D3	All	29	14.91
	D4	EH29-H780	29	14.91
		HF35-F1152	38	19.42
		KF40-F1517/KH40-H1517	38	19.42
	D5	All	38	19.42
	D6		54	27.81
D8	54		27.81	
Stratix V E	E9	All	68	35.21
	EB		68	35.21

Using Error Detection Features in User Mode

This section describes the pin, registers, process flow, and procedures for error detection in user mode.

Enabling Error Detection and Internal Scrubbing

To enable user mode error detection and internal scrubbing in the Intel Quartus Prime software, follow these steps:

1. On the Assignments menu, click **Device**.
2. In the Device dialog box, click **Device and Pin Options**.
3. In the **Category** list, click **Error Detection CRC**.
4. Turn on **Enable Error Detection CRC_ERROR** pin.

5. To set the `CRC_ERROR` pin as output open drain, turn on **Enable open drain on CRC_ERROR pin**. Turning off this option sets the `CRC_ERROR` pin as output.
6. To enable the on-chip error correction feature, turn on **Enable internal scrubbing**.
7. In the **Divide error check frequency by** list, select a valid divisor.
8. Click OK.

CRC_ERROR Pin

Table 9-4: Pin Description

Pin Name	Pin Type	Description
<code>CRC_ERROR</code>	I/O or output/ output open-drain	<p>An active-high signal, when driven high indicates that an error is detected in the CRAM bits. This pin is only used when you enable error detection in user mode. Otherwise, the pin is used as a user I/O pin.</p> <p>When using the WYSIWYG function, you can route the <code>crcerror</code> port from the WYSIWYG atom to the dedicated <code>CRC_ERROR</code> pin or any user I/O pin. To route the <code>crcerror</code> port to a user I/O pin, insert a D-type flipflop between them.</p>

Error Detection Registers

This section describes the registers used in user mode.

Figure 9-3: Block Diagram for Error Detection in User Mode

The block diagram shows the registers and data flow in user mode.

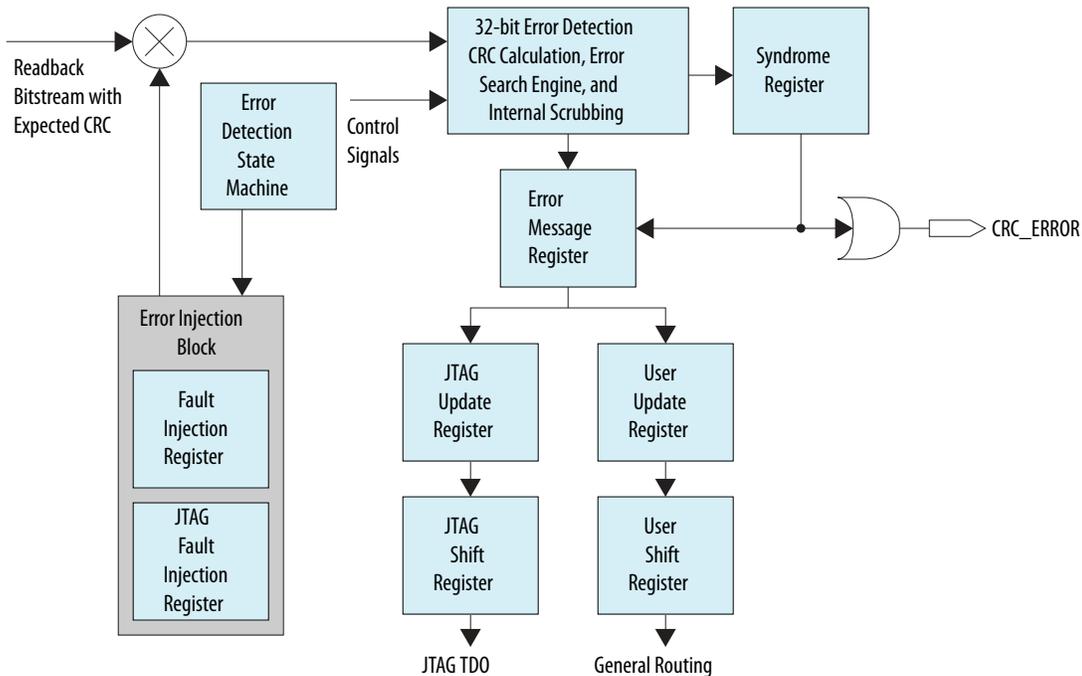


Table 9-5: Error Detection Registers

Name	Width (Bits)	Description
Syndrome register	32	Contains the 32-bit CRC signature calculated for the current frame. If the CRC value is 0, the <code>CRC_ERROR</code> pin is driven low to indicate no error. Otherwise, the pin is pulled high.
Error message register (EMR)	67	Contains error details for single-bit and double-adjacent errors. The error detection circuitry updates this register each time the circuitry detects an error. The Error Message Register Map figure shows the fields in this register and the Error Type in EMR table lists the possible error types.
JTAG update register	67	This register is automatically updated with the contents of the EMR one clock cycle after the content of this register is validated. The JTAG update register includes a clock enable, which must be asserted before its contents are written to the JTAG shift register. This requirement ensures that the JTAG update register is not overwritten when its contents are being read by the JTAG shift register.
JTAG shift register	67	This register allows you to access the contents of the JTAG update register via the JTAG interface using the <code>SHIFT_EDERROR_REG</code> JTAG instruction.

Name	Width (Bits)	Description
User update register	67	This register is automatically updated with the contents of the EMR one clock cycle after the contents of this register are validated. The user update register includes a clock enable, which must be asserted before its contents are written to the user shift register. This requirement ensures that the user update register is not overwritten when its contents are being read by the user shift register.
User shift register	67	This register allows user logic to access the contents of the user update register via the core interface.
JTAG fault injection register	46	You can use this register with the <code>EDERROR_INJECT_JTAG</code> instruction to inject errors in the bitstream. The JTAG Fault Injection Register Map table lists the fields in this register.
Fault injection register	46	This register is updated with the contents of the JTAG fault injection register.

Figure 9-4: Error Message Register Map

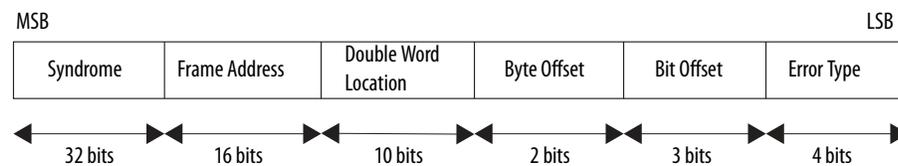


Table 9-6: Error Type in EMR

The following table lists the possible error types reported in the error type field in the EMR.

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	Error types other than single-bit and double-adjacent errors.

Table 9-7: JTAG Fault Injection Register Map

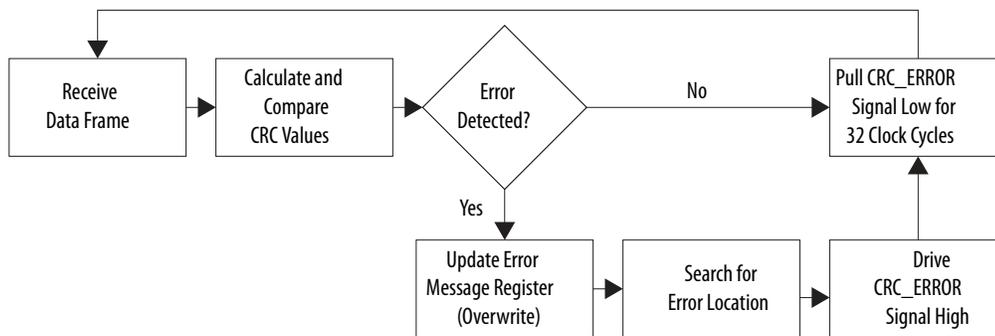
Field Name	Bit Range	Description
Error Byte Value	31:0	Contains the location of the bit error that corresponds to the error injection type to this field.
Byte Location	41:32	Contains the location of the injected error in the first data frame.

Field Name	Bit Range				Description
Error Type	45:42				Specifies the following error types.
	Bit 45	Bit 44	Bit 43	Bit 42	
	0	0	0	0	No error
	0	0	0	1	Single-bit error
	0	0	1	0	Double adjacent error

Error Detection Process

When enabled, the user mode error detection process activates automatically when the FPGA enters user mode. The process continues to run until the device is reset even when an error is detected in the current frame.

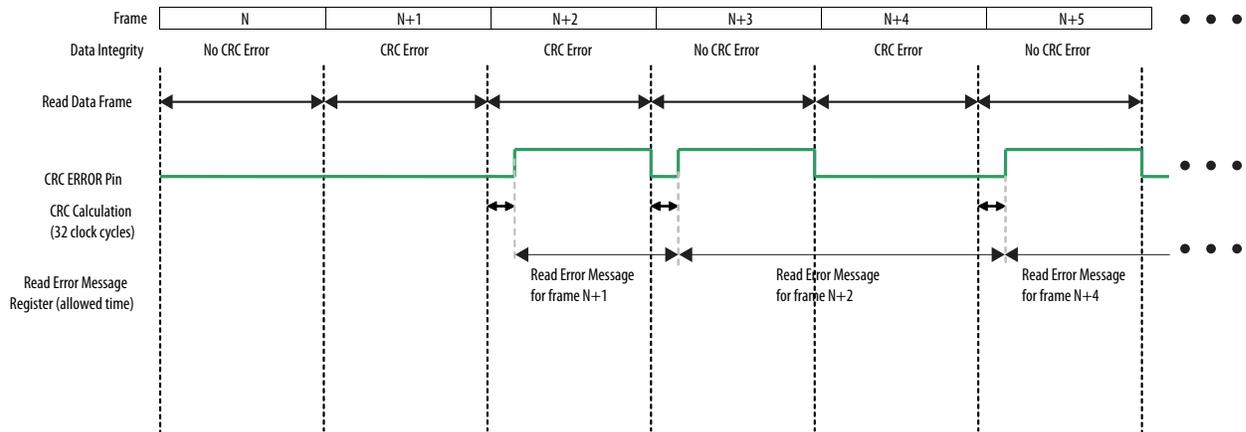
Figure 9-5: Error Detection Process Flow in User Mode



Timing

The `CRC_ERROR` pin is always driven low during CRC calculation. When an error occurs, the EDCRC hard block takes 32 clock cycles to update the EMR, the pin is driven high once the EMR is updated. Therefore, you can start retrieving the contents of the EMR at the rising edge of the `CRC_ERROR` pin. The pin stays high until the current frame is read and then driven low again for 32 clock cycles. To ensure information integrity, complete the read operation within one frame of the CRC verification. The following diagram shows the timing of these events.

Figure 9-6: Timing Requirements



Retrieving Error Information

You can retrieve the error information via the core interface or the JTAG interface using the `SHIFT_EDERROR_REG` JTAG instruction.

Recovering from CRC Errors

The system that hosts the FPGA must control device reconfiguration. To recover from a CRC error, drive the `nCONFIG` signal low. The system waits for a safe time before reconfiguring the device. When reconfiguration completes successfully, the FPGA operates as intended.

Related Information

- [Error Detection Frequency](#) on page 9-3
Provides more information about the minimum and maximum error detection frequencies.
- [Minimum EMR Update Interval](#) on page 9-3
Provides more information about the duration of each Stratix V device.
- [Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices](#)
Provides more information about how to retrieve the error information.

Testing the Error Detection Block

You can inject errors into the configuration data to test the error detection block. This error injection methodology provides design verification and system fault tolerance characterization.

Testing via the JTAG Interface

You can intentionally inject single or double-adjacent errors into the configuration data using the `EDERROR_INJECT` JTAG instruction.

Table 9-8: EDERROR_INJECT instruction

JTAG Instruction	Instruction Code	Description
EDERROR_INJECT	00 0001 0101	Use this instruction to inject errors into the configuration data. This instruction controls the JTAG fault injection register, which contains the error you want to inject into the bitstream.

You can only inject errors into the first frame of the configuration data. However, you can monitor the error information at any time. Altera recommends that you reconfigure the FPGA after the test completes.

Automating the Testing Process

You can automate the testing process by creating a Jam™ file (.jam). Using this file, you can verify the CRC functionality in-system and on-the-fly without reconfiguring the device. You can then switch to the CRC circuitry to check for real errors caused by an SEU.

Related Information

[Test Methodology of Error Detection and Recovery using CRC in Altera FPGA Devices](#)

Provides more information about how to test the error detection block.

SEU Mitigation for Stratix V Devices Revision History

Date	Version	Changes
December 2015	2015.12.21	<ul style="list-style-type: none"> Changed instances of Quartus II to Quartus Prime. Updated the clock cycles for the CRC calculation in the Error Detection Process section. Removed a note from the Internal Scrubbing section.
January 2015	2015.01.23	Updated the description in the CRC Calculation Time section.
June 2014	2014.06.30	Updated the CRC Calculation Time section.
January 2014	2014.01.10	<ul style="list-style-type: none"> Updated the CRC Calculation Time section to include a formula to calculate the minimum and maximum time. Updated the maximum error detection frequency. Removed preliminary and finalized the values for the Minimum EMR Update Interval and CRC Calculation Time.
May 2013	2013.05.06	<ul style="list-style-type: none"> Added link to the known document issues in the Knowledge Base. Moved all links to the Related Information section of respective topics for easy reference.

Date	Version	Changes
December 2012	2012.12.28	<ul style="list-style-type: none"> Updated the valid values of n in the error detection frequency equation. Updated the width of the JTAG fault injection and fault injection registers.
June 2012	2.0	Minor text edits.
February 2012	1.4	Updated Table 10–9 and Table 10–10.
November 2011	1.3	<ul style="list-style-type: none"> Chapter moved to Volume 2. Updated Table 10–9 and Table 10–10. Minor text edits.
May 2011	1.2	<ul style="list-style-type: none"> Chapter moved to Volume 2. Updated Table 10–9 and Table 10–10. Minor text edits.
December 2010	1.1	No change.
July 2010	1.0	Initial release.

JTAG Boundary-Scan Testing in Stratix V Devices 10

2020.09.10

SV51012



Subscribe



Send Feedback

This chapter describes the boundary-scan test (BST) features in Stratix V devices.

Related Information

- [JTAG Configuration](#) on page 8-34
Provides more information about JTAG configuration.
- [Stratix V Device Handbook: Known Issues](#)
Lists the planned updates to the *Stratix V Device Handbook* chapters.

BST Operation Control

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).

IDCODE

The IDCODE is unique for each Stratix V device. Use this code to identify the devices in a JTAG chain.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

Table 10-1: IDCODE Information for Stratix V Devices

Family	Member Code	IDCODE (32 Bits)			
		Version (4 Bits)	Part Number (16 Bits)	Manufacture Identity (11 Bits)	LSB (1 Bit)
Stratix V GX	A3 ⁽²¹⁾	0000	0010 1001 0100 0111	000 0110 1110	1
	A3 ⁽²²⁾	0000	0010 1001 0010 0001	000 0110 1110	1
	A4	0000	0010 1001 0010 0111	000 0110 1110	1
	A5	0000	0010 1001 0001 0011	000 0110 1110	1
	A7	0000	0010 1001 0000 0011	000 0110 1110	1
	A9	0000	0010 1001 0100 0101	000 0110 1110	1
	AB	0000	0010 1001 0010 0101	000 0110 1110	1
	B5	0000	0010 1001 0001 0010	000 0110 1110	1
	B6	0000	0010 1001 0000 0010	000 0110 1110	1
	B9	0000	0010 1001 0001 0101	000 0110 1110	1
	BB	0000	0010 1001 0000 0101	000 0110 1110	1
Stratix V GT	C5	0000	0010 1001 0010 0011	000 0110 1110	1
	C7	0000	0010 1001 0100 0011	000 0110 1110	1

⁽²¹⁾ The IDCODE is applicable for KF35 and KF40 packages only.

⁽²²⁾ The IDCODE is applicable for EH29 and HF35 packages only.

Family	Member Code	IDCODE (32 Bits)			
		Version (4 Bits)	Part Number (16 Bits)	Manufacture Identity (11 Bits)	LSB (1 Bit)
Stratix V GS	D3	0000	0010 1001 0001 0001	000 0110 1110	1
	D4 ⁽²³⁾	0000	0010 1001 0000 0001	000 0110 1110	1
	D4 ⁽²⁴⁾	0000	0010 1001 0001 0111	000 0110 1110	1
	D5	0000	0010 1001 0000 0111	000 0110 1110	1
	D6	0000	0010 1001 0001 0100	000 0110 1110	1
	D8	0000	0010 1001 0000 0100	000 0110 1110	1
Stratix V E	E9	0000	0010 1001 1001 0101	000 0110 1110	1
	EB	0000	0010 1001 1000 0101	000 0110 1110	1

Supported JTAG Instruction

Table 10-2: JTAG Instructions Supported by Stratix V Devices

JTAG Instruction	Instruction Code	Description
SAMPLE/PRELOAD	00 0000 0101	<ul style="list-style-type: none"> Allows you to capture and examine a snapshot of signals at the device pins during normal device operation and permits an initial data pattern to be an output at the device pins. Use this instruction to preload the test data into the update registers before loading the EXTTEST instruction. Used by the Signal Tap II Embedded Logic Analyzer.

⁽²³⁾ The IDCODE is applicable for EH29 and HF35 packages only.

⁽²⁴⁾ The IDCODE is applicable for KF40 package only.

JTAG Instruction	Instruction Code	Description
EXTEST	00 0000 1111	<ul style="list-style-type: none"> Allows you to test the external circuit and board-level interconnects by forcing a test pattern at the output pins, and capturing the test results at the input pins. Forcing known logic high and low levels on output pins allows you to detect opens and shorts at the pins of any device in the scan chain. The high-impedance state of EXTEST is overridden by bus hold and weak pull-up resistor features.
BYPASS	11 1111 1111	Places the 1-bit bypass register between the TDI and TDO pins. During normal device operation, the 1-bit bypass register allows the BST data to pass synchronously through the selected devices to adjacent devices.
USERCODE	00 0000 0111	<ul style="list-style-type: none"> Examines the user electronic signature (UES) within the devices along a JTAG chain. Selects the 32-bit USERCODE register and places it between the TDI and TDO pins to allow serial shifting of USERCODE out of TDO. The UES value is set to default value before configuration and is only user-defined after the device is configured.

JTAG Instruction	Instruction Code	Description
IDCODE	00 0000 0110	<ul style="list-style-type: none">Identifies the devices in a JTAG chain. If you select IDCODE, the device identification register is loaded with the 32-bit vendor-defined identification code.Selects the IDCODE register and places it between the TDI and TDO pins to allow serial shifting of IDCODE out of TDO.IDCODE is the default instruction at power up and in the TAP RESET state. Without loading any instructions, you can go to the SHIFT_DR state and shift out the JTAG device ID.
HIGHZ	00 0000 1011	<ul style="list-style-type: none">Sets all user I/O pins to an inactive drive state.Places the 1-bit bypass register between the TDI and TDO pins. During normal operation, the 1-bit bypass register allows the BST data to pass synchronously through the selected devices to adjacent devices while tri-stating all I/O pins until a new JTAG instruction is executed.If you are testing the device after configuration, the programmable weak pull-up resistor or the bus hold feature overrides the HIGHZ value at the pin.

JTAG Instruction	Instruction Code	Description
CLAMP	00 0000 1010	<ul style="list-style-type: none"> Places the 1-bit bypass register between the TDI and TDO pins. During normal operation, the 1-bit bypass register allows the BST data to pass synchronously through the selected devices to adjacent devices while holding the I/O pins to a state defined by the data in the boundary-scan register. If you are testing the device after configuration, the programmable weak pull-up resistor or the bus hold feature overrides the CLAMP value at the pin. The CLAMP value is the value stored in the update register of the boundary-scan cell (BSC).
PULSE_NCONFIG	00 0000 0001	Emulates pulsing the nCONFIG pin low to trigger reconfiguration even though the physical pin is not affected.
CONFIG_IO	00 0000 1101	Allows I/O reconfiguration (after or during reconfigurations) through the JTAG ports using I/O configuration shift register (IOCSR) for JTAG testing. You can issue the CONFIG_IO instruction only after the nSTATUS pin goes high.
LOCK	01 1111 0000	Put the device in JTAG secure mode. In this mode, only BYPASS, SAMPLE/PRELOAD, EXTEST, IDCODE, SHIFT_EDERROR_REG, and UNLOCK instructions are supported. This instruction can only be accessed through JTAG core access in user mode. It cannot be accessed through external JTAG pins in test or user mode.

JTAG Instruction	Instruction Code	Description
UNLOCK	11 0011 0001	Release the device from the JTAG secure mode to enable access to all other JTAG instructions. This instruction can only be accessed through JTAG core access in user mode. It cannot be accessed through external JTAG pins in test or user mode.
KEY_CLR_VREG	00 0010 1001	Clears the volatile key.
KEY_VERIFY	00 0001 0011	Verifies the non-volatile key has been cleared.
EXTEST_PULSE	00 1000 1111	<p>Enables board-level connectivity checking between the transmitters and receivers that are AC coupled by generating three output transitions:</p> <ul style="list-style-type: none"> • 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. <p>The EXTEST_PULSE JTAG instruction is only supported in user mode for Stratix V devices.</p>
EXTEST_TRAIN	00 0100 1111	<p>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.</p> <p>The EXTEST_TRAIN JTAG instruction is only supported in user mode for Stratix V devices.</p>

Note: If the device is in a reset state and 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.

Note: 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. AC-coupled and DC-coupled HSSI are only supported in post-configuration mode.

Related Information

[JTAG Secure Mode](#) on page 8-46

Provides more information about `PULSE_NCONFIG`, `CONFIG_IO`, `LOCK`, and `UNLOCK` JTAG instructions.

JTAG Secure Mode

If you enable the tamper-protection bit, the Stratix V device is in JTAG secure mode after power up. In the JTAG secure mode, the JTAG pins support only the `BYPASS`, `SAMPLE/PRELOAD`, `EXTEST`, `IDCODE`, `SHIFT_EDERROR_REG`, and `UNLOCK` instructions. Issue the `UNLOCK` JTAG instruction to enable support for other JTAG instructions.

JTAG Private Instruction

Caution: Never invoke the following instruction codes. These instructions can damage and render the device unusable:

- 1100010000
- 0011001001
- 1100010011
- 1100010111
- 0111100000
- 1110110011
- 0000101010

I/O Voltage for JTAG Operation

A Stratix V device operating in BST mode uses four required JTAG pins—`TDI`, `TDO`, `TMS`, `TCK`, and one optional pin, `TRST`.

The `TCK` pin has an internal weak pull-down resistor, while the `TDI` and `TMS` pins have internal weak pull-up resistors. The 3.0- or 2.5-V V_{CCPD} supply of I/O bank 3A powers the `TDO`, `TDI`, `TMS`, and `TCK` pins. All user I/O pins are tri-stated during JTAG configuration.

The JTAG chain supports several different devices. Use the supported `TDO` and `TDI` voltage combinations listed in the following table if the JTAG 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 10-3: Supported TDO and TDI Voltage Combinations

The `TDO` output buffer for V_{CCPD} of 3.0 V meets V_{OH} (MIN) of 2.4 V, and the `TDO` output buffer for V_{CCPD} of 2.5 V meets V_{OH} (MIN) of 2.0 V.

Device	TDI Input Buffer Power (V)	Stratix V TDO V _{CCPD}	
		V _{CCPD} = 3.0 V	V _{CCPD} = 2.5 V
Stratix V	V _{CCPD} = 3.0 V	Yes	Yes
	V _{CCPD} = 2.5 V	Yes	Yes
Non-Stratix V ⁽²⁵⁾	V _{CC} = 3.3 V	Yes	Yes
	V _{CC} = 2.5 V	Yes	Yes
	V _{CC} = 1.8 V	Yes	Yes
	V _{CC} = 1.5 V	Yes	Yes

Performing BST

You can issue `BYPASS`, `IDCODE`, and `SAMPLE` JTAG instructions before, after, or during configuration without having to interrupt configuration.

To issue other JTAG instructions, follow these guidelines:

- To perform testing before configuration, hold the `nCONFIG` pin low.
- To perform BST during configuration, issue `CONFIG_IO` JTAG instruction to interrupt configuration. While configuration is interrupted, you can issue other JTAG instructions to perform BST. After BST is completed, issue the `PULSE_CONFIG` JTAG instruction or pulse `nCONFIG` low to reconfigure the device.

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 configuration operations. Toggling these pins does not disrupt BST operation (other than the expected BST behavior).

If you design a board for JTAG configuration of Stratix V devices, consider the connections for the dedicated configuration pins.

Related Information

- [JTAG Configuration](#) on page 8-34
Provides more information about JTAG configuration.
- [Stratix V Device Datasheet](#)
Provides more information about JTAG configuration timing.

Enabling and Disabling IEEE Std. 1149.1 BST Circuitry

The IEEE Std. 1149.1 BST circuitry is enabled after the Stratix V device powers up.

To ensure that you do not inadvertently enable the IEEE Std. 1149.1 circuitry when it is not required, disable the circuitry permanently with pin connections as listed in the following table.

⁽²⁵⁾ The input buffer must be tolerant to the TDO V_{CCPD} voltage.

Table 10-4: Pin Connections to Permanently Disable the IEEE Std. 1149.1 Circuitry for Stratix V Devices

JTAG Pins ⁽²⁶⁾	Connection for Disabling
TMS	V _{CCPD} supply of Bank 3A
TCK	GND
TDI	V _{CCPD} supply of Bank 3A
TDO	Leave open

Guidelines for IEEE Std. 1149.1 Boundary-Scan Testing

Consider the following guidelines when you perform BST with IEEE Std. 1149.1 devices:

- If the “10...” pattern does not shift out of the instruction register through the TDO pin during the first clock cycle of the SHIFT_IR state, the TAP controller did not reach the proper state. To solve this problem, try one of the following procedures:
 - Verify that the TAP controller has reached the SHIFT_IR state correctly. To advance the TAP controller to the SHIFT_IR state, return to the RESET state and send the 01100 code to the TMS pin.
 - Check the connections to the VCC, GND, JTAG, and dedicated configuration pins on the device.
- Perform a SAMPLE/PRELOAD test cycle before the first EXTEST test cycle to ensure that known data is present at the device pins when you enter EXTEST mode. If the OEJ update register contains 0, the data in the OUTJ update register is driven out. The state must be known and correct to avoid contention with other devices in the system.
- Do not perform EXTEST testing during in-circuit reconfiguration because EXTEST is not supported during in-circuit reconfiguration. To perform testing, wait for the configuration to complete or issue the CONFIG_IO instruction to interrupt configuration.
- After configuration, you cannot test any pins in a differential pin pair. To perform BST after configuration, edit and redefine the BSC group that correspond to these differential pin pairs as an internal cell.

Related Information

[IEEE 1149.6 BSDL Files](#)

Provides more information about BSC group definitions.

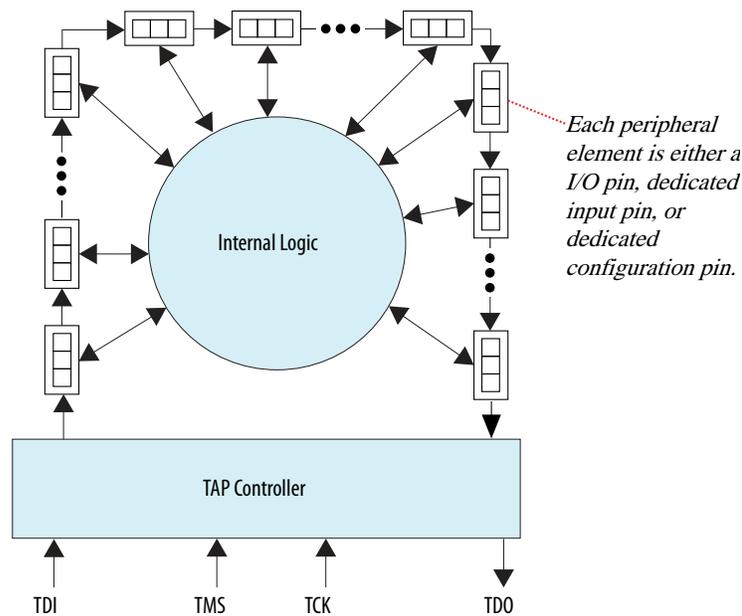
IEEE Std. 1149.1 Boundary-Scan Register

The boundary-scan register is a large serial shift register that uses the TDI pin as an input and the TDO pin as an output. The boundary-scan register consists of 3-bit peripheral elements that are associated with Stratix V I/O pins. You can use the boundary-scan register to test external pin connections or to capture internal data.

⁽²⁶⁾ The JTAG pins are dedicated. Software option is not available to disable JTAG in Stratix V devices.

Figure 10-1: Boundary-Scan Register

This figure shows how test data is serially shifted around the periphery of the IEEE Std. 1149.1 device.



Boundary-Scan Cells of a Stratix V Device I/O Pin

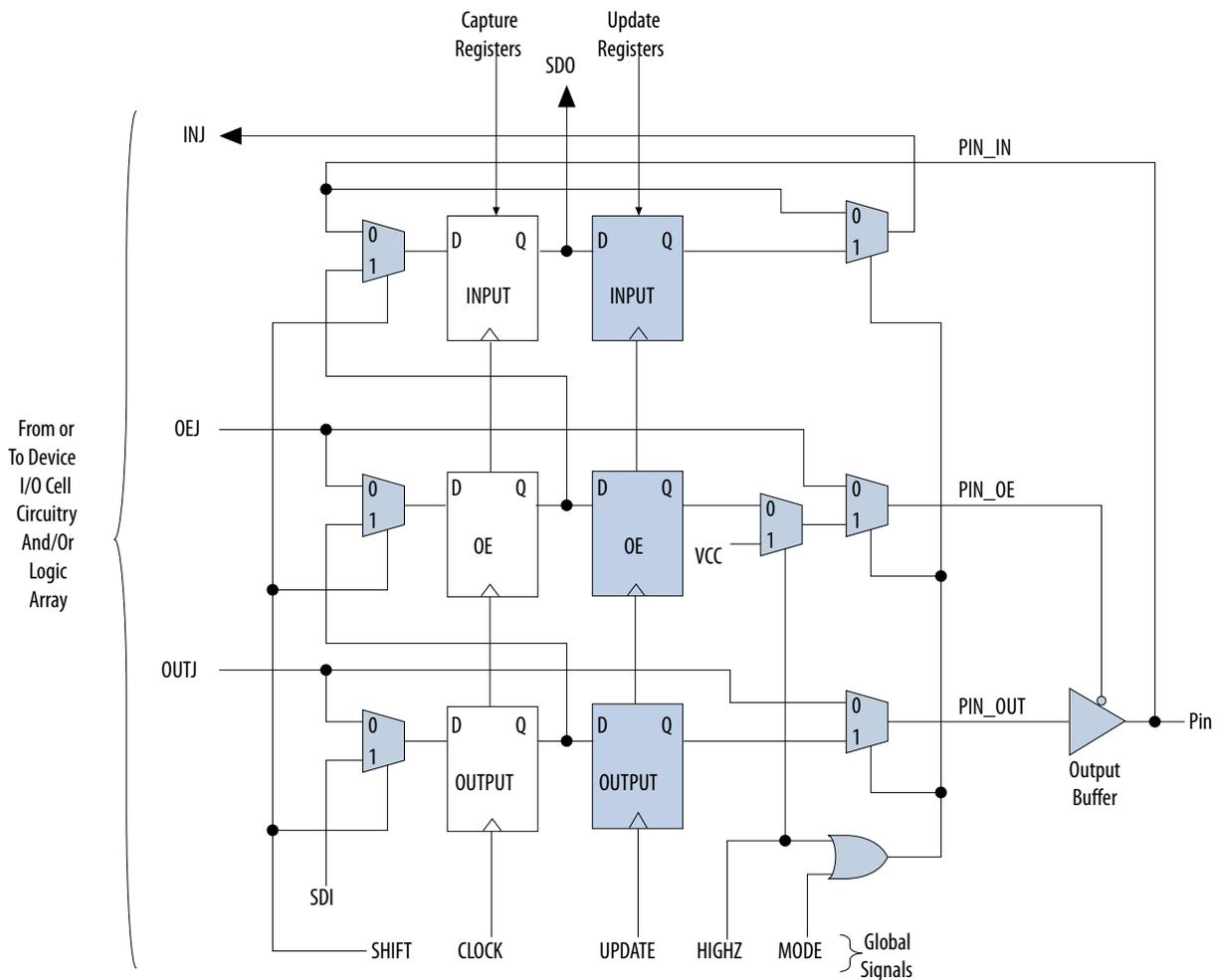
The Stratix V device 3-bit BSC consists of the following registers:

- Capture registers—Connect to internal device data through the `OUTJ`, `OEJ`, and `PIN_IN` signals.
- Update registers—Connect to external data through the `PIN_OUT` and `PIN_OE` signals.

The TAP controller generates the global control signals for the IEEE Std. 1149.1 BST registers (`shift`, `clock`, and `update`) internally. A decode of the instruction register generates the `MODE` signal.

The data signal path for the boundary-scan register runs from the serial data in (`SDI`) signal to the serial data out (`SDO`) signal. The scan register begins at the `TDI` pin and ends at the `TDO` pin of the device.

Figure 10-2: User I/O BSC with IEEE Std. 1149.1 BST Circuitry for Stratix V Devices



Note: TDI, TDO, TMS, and TCK pins, all VCC and GND pin types, and VREF pins do not have BSCs.

Table 10-5: Boundary-Scan Cell Descriptions for Stratix V Devices

This table lists the capture and update register capabilities of all BSCs within Stratix V devices.

Pin Type	Captures			Drives			Comments
	Output Capture Register	OE Capture Register	Input Capture Register	Output Update Register	OE Update Register	Input Update Register	
User I/O pins	OUTJ	OEJ	PIN_IN	PIN_OUT	PIN_OE	INJ	—
Dedicated clock input	0	1	PIN_IN	No Connect (N.C.)	N.C.	N.C.	PIN_IN drives to the clock network or logic array

Pin Type	Captures			Drives			Comments
	Output Capture Register	OE Capture Register	Input Capture Register	Output Update Register	OE Update Register	Input Update Register	
Dedicated input	0	1	PIN_IN	N.C.	N.C.	N.C.	PIN_IN drives to the control logic
Dedicated bidirectional (open drain) ⁽²⁷⁾	0	OEJ	PIN_IN	N.C.	N.C.	N.C.	PIN_IN drives to the configuration control
Dedicated bidirectional ⁽²⁸⁾	OUTJ	OEJ	PIN_IN	N.C.	N.C.	N.C.	PIN_IN drives to the configuration control and OUTJ drives to the output buffer
Dedicated output ⁽²⁹⁾	OUTJ	0	0	N.C.	N.C.	N.C.	OUTJ drives to the output buffer

IEEE Std. 1149.6 Boundary-Scan Register

The BSCs for HSSI transmitters ($GXB_TX[p, n]$) and receivers/input clock buffers ($GXB_RX[p, n]$)/(REFCLK[p, n]) in Stratix V devices are different from the BSCs for the I/O pins.

⁽²⁷⁾ This includes the CONF_DONE and nSTATUS pins.

⁽²⁸⁾ This includes the DCLK pin.

⁽²⁹⁾ This includes the nCEO pin.

Figure 10-3: HSSI Transmitter BSC with IEEE Std. 1149.6 BST Circuitry for Stratix V Devices

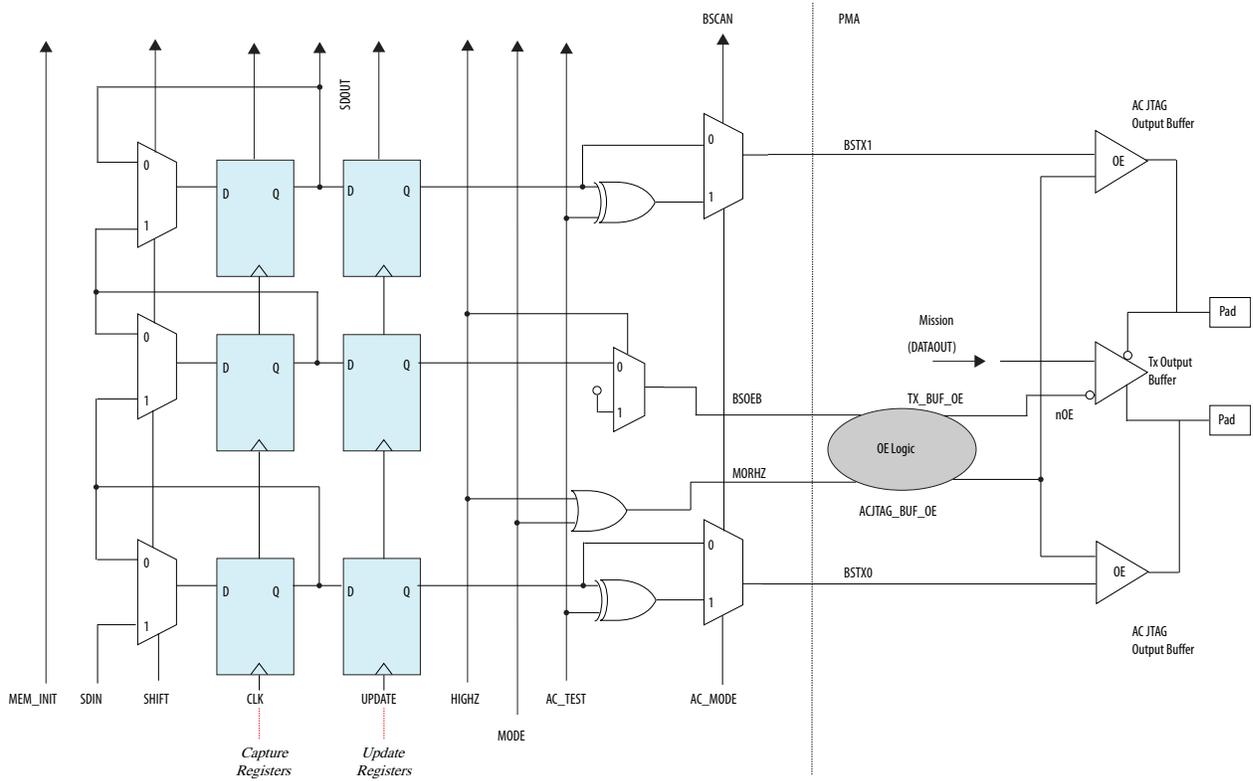
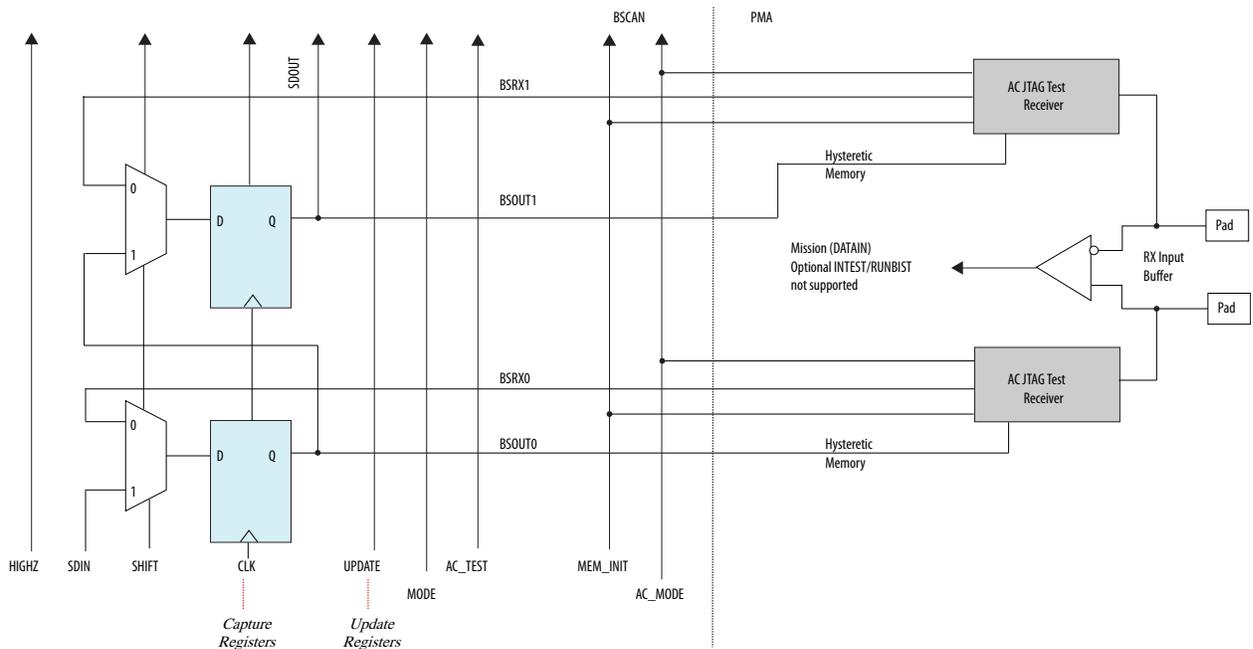


Figure 10-4: HSSI Receiver/Input Clock Buffer with IEEE Std. 1149.6 BST Circuitry for Stratix V Devices



JTAG Boundary-Scan Testing in Stratix V Devices Revision History

Date	Version	Changes
June 2016	2016.06.10	Added 0000101010 instruction code into JTAG Private Instruction section.
December 2015	2015.12.21	Changed instances of Quartus II to Quartus Prime.
January 2014	2014.01.10	<ul style="list-style-type: none">Updated the Supported JTAG Instruction section.Updated the <code>KEY_CLR_VREG</code> JTAG instruction.
May 2013	2013.05.06	<ul style="list-style-type: none">Added link to the known document issues in the Knowledge Base.Updated the description for <code>EXTEST_TRAIN</code> and <code>EXTEST_PULSE</code> JTAG instructions.Moved all links to the Related Information section of respective topics for easy reference.
December 2012	2012.12.28	Reorganized content and updated template.
June 2012	1.5	Updated Table 11-1.
December 2011	1.4	Updated Table 11-2 to include <code>KEY_CLR_VREG</code> and <code>KEY_VERIFY</code> JTAG instructions.
November 2011	1.3	Updated Table 11-1 and Table 11-2.
May 2011	1.2	<ul style="list-style-type: none">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 release.
July 2010	1.0	Initial release.

Power Management in Stratix V Devices 11

2020.09.10

SV51013



Subscribe



Send Feedback

This chapter describes the programmable power technology, hot-socketing feature, power-on reset (POR) requirements, power-up sequencing recommendation, temperature sensing diode (TSD), and their implementation in Stratix V devices.

Related Information

- [Stratix V Device Handbook: Known Issues](#)
Lists the planned updates to the Stratix V Device Handbook chapters.
- [PowerPlay Power Analysis](#)
Provides more information about the Quartus®Prime PowerPlay Power Analyzer tool in volume 3 of the Quartus Prime Handbook.
- [Stratix V Device Datasheet](#)
Provides more information about the recommended operating conditions of each power supply.
- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
Provides detailed information about power supply pin connection guidelines and power regulator sharing.
- [Stratix V GT Device Family Pin Connection Guidelines](#)
Provides detailed information about power supply pin connection guidelines and power regulator sharing.
- [Board Design Resource Center](#)
Provides detailed information about power supply design requirements.
- [PowerPlay Early Power Estimators \(EPE\) and Power Analyzer](#)
Provides more information about the two supplies which make up the V_{CC} supply. They are V_{CCL} (core V_{CC}) and V_{CCP} (periphery V_{CC}). The sum of I_{CCL} and I_{CCP} equals to I_{CC} . I_{CCL} and I_{CCP} is found on the EPE report tab.
- [Stratix V Device Design Guidelines](#)
- [Stratix V GT Device Design Guidelines](#)

Power Consumption

The total power consumption of a Stratix V device consists of the following components:

- Static power—the power that the configured device consumes when powered up but no clocks are operating.
- Dynamic power—the additional power consumption of the device due to signal activity or toggling.

Intel Corporation. All rights reserved. Intel, the Intel logo, Altera, Arria, Cyclone, Enpirion, MAX, Nios, Quartus and Stratix words and logos are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries. Intel warrants performance of its FPGA and semiconductor products to current specifications in accordance with Intel's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Intel assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Intel. Intel customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

*Other names and brands may be claimed as the property of others.

ISO
9001:2015
Registered

ALTERA
now part of Intel

Dynamic Power Equation

Figure 11-1: Dynamic Power

The following equation shows how to calculate dynamic power where P is power, C is the load capacitance, and V is the supply voltage level.

$$P = \frac{1}{2} CV^2 \times frequency$$

The equation shows that power is design-dependent and is determined by the operating frequency of your design. Stratix V devices minimize static and dynamic power using advanced process optimizations. This technology allows Stratix V designs to meet specific performance requirements with the lowest possible power.

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 Intel Quartus Prime 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 Intel Quartus Prime software determines whether a tile should be in high-speed or low-power mode based on the timing constraints of the design.

Stratix V tiles 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.

The Intel Quartus Prime 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, you may assert the `areset` pin high to keep the PLL in low-power mode.

Altera recommends that you power down unused PCIe HIPs, per side, by connecting the PCIe HIP power to GND on the PCB for additional power savings. All of the HIPs on a side of the device must be unused to be powered down. For additional information refer to the pin connection guidelines.

Table 11-1: Programmable Power Capabilities for Stratix V Devices

This table lists the available Stratix V programmable power capabilities. Speed grade considerations can add to the permutations to give you flexibility in designing your system.

Feature	Programmable Power Technology
LAB	Yes
Routing	Yes
Memory Blocks	Fixed setting ⁽³⁰⁾
DSP Blocks	Fixed setting ⁽³⁰⁾
Clock Networks	No

Related Information

- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
Provides more information about powering down PCIe HIPs.
- [Stratix V GT Device Family Pin Connection Guidelines](#)
Provides more information about powering down PCIe HIPs.

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. You can calculate junction temperature using ambient or case temperature, junction-to-ambient (j_a) or junction-to-case (j_c) thermal resistance, and device power consumption. Stratix V devices monitor its die temperature with the internal TSD with built-in analog-to-digital converter (ADC) circuitry or the external TSD with an external temperature sensor. This allows you to control the air flow to the device.

⁽³⁰⁾ 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.

Internal Temperature Sensing Diode

You can use the Stratix V internal TSD in the following operations:

- Power-up mode—to read the die's temperature during configuration, enable the Altera Temperature Sensor IP core in your design.
- User mode—to read the die's temperature during user mode, assert the `clken` signal to the internal TSD circuitry.

Note: To reduce power consumption, disable the Stratix V internal TSD when you are not using it.

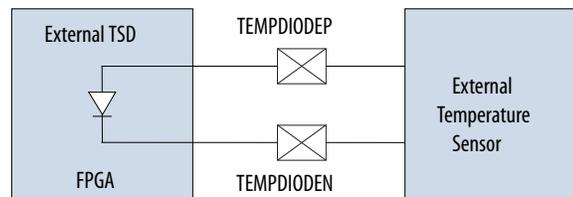
Related Information

- [Altera Temperature Sensor IP Core User Guide](#)
Provides more information about using the Altera Temperature Sensor IP core.
- [Stratix V Device Datasheet](#)
Provides more information about the Stratix V internal TSD specification.

External Temperature Sensing Diode

The Stratix V external TSD requires two pins for voltage reference. The following figure shows how to connect the external TSD with an external temperature sensor device, allowing external sensing of the Stratix V die temperature. For example, you can connect external temperature sensing devices, such as MAX1619, MAX1617A, MAX6627, and ADT7411 to the two external TSD pins for Stratix V device die temperature reading. The TSD diode is a substrate or common collector PNP diode type.

Figure 11-2: TSD External Pin Connections



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 TEMPDIODE_P/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 TEMPDIODE_P/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 the 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 (TEMPDIODE_P/TEMPDIODE_N) can be connected to GND because the external TSD pins are not used.

For details about device specification and connection guidelines, refer to the external temperature sensor device datasheet from the device manufacturer.

Related Information

- [Stratix V Device Datasheet](#)
Provides details about the external TSD specification.
- [Stratix V E, GS, and GX Device Family Pin Connection Guidelines](#)
Provides details about the TEMPDIODE_P/TEMPDIODE_N pin connection when you are not using an external TSD.
- [Stratix V GT Device Family Pin Connection Guidelines](#)
Provides details about the TEMPDIODE_P/TEMPDIODE_N pin connection when you are not using an external TSD.

Hot-Socketing Feature

Stratix V devices support hot socketing—also known as hot plug-in or hot swap.

The hot-socketing circuitry monitors the V_{CCIO}, V_{CCPD}, and V_{CC} power supplies and all V_{CCIO} and V_{CCPD} banks.

When powering up or powering down these power supplies, refer to the Power-Up Sequence section of this handbook.

During the hot-socketing operation, the I/O pin capacitance is less than 15 pF and the clock pin capacitance is less than 20 pF.

The hot-socketing capability removes some of the difficulty that designers face when using the Stratix V devices on PCBs that contain a mixture of devices with different voltage requirements.

The hot-socketing capability in Stratix V devices provides the following advantages:

- You can drive signals into the I/O, dedicated input, and dedicated clock pins before or during power up or power down without damaging the device. External input signals to the I/O pins of the unpowered device will not power the power supplies through internal paths within the device.
- The output buffers are tri-stated during system power up or power down. Because the Stratix V device does not drive signals out before or during power up, the device does not affect the other operating buses.
- You can insert or remove a Stratix V device from a powered-up system board without damaging or interfering with the system board's operation. This capability allows you to avoid sinking current through the device signal pins to the device power supply, which can create a direct connection to GND that causes power supply failures.
- During hot socketing, Stratix V devices are immune to latch up that can occur when a device is hot-socketed into an active system.

Altera uses GND as a reference for hot-socketing and I/O buffer circuitry designs. To ensure proper operation, connect GND between boards before connecting the power supplies. This prevents 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 in the Altera device.

Related Information

- [Power-Up Sequence](#) on page 11-7
- [Stratix V Device Datasheet](#)
Provides details about the Stratix V hot-socketing specifications.

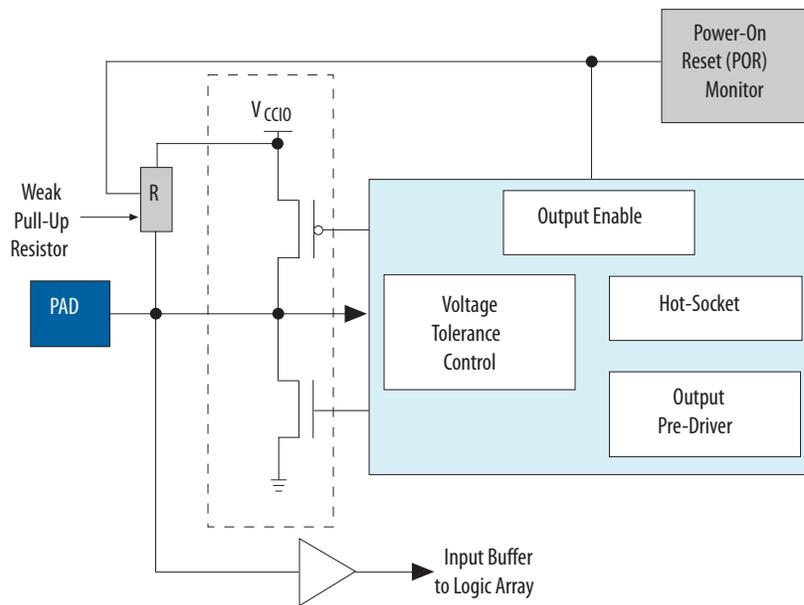
Hot-Socketing Implementation

The hot-socketing feature tri-state the output buffer during power up and power down of the power supplies. When these power supplies are below the threshold voltage, the hot-socketing circuitry generates an internal `HOTSOCKET` signal.

Hot-socketing circuitry prevents excess I/O leakage during power up. When the voltage ramps up very slowly, I/O leakage is still relatively low, even after the release of the POR signal and configuration is complete.

Note: The output buffer cannot flip from the state set by the hot-socketing circuitry at very low voltage. To allow the `CONF_DONE` and `nSTATUS` pins to operate during configuration, the hot-socketing feature is not applied to these configuration pins. Therefore, these pins will drive out during power up and power down.

Figure 11-3: Hot-Socketing Circuitry for Stratix V Devices



The POR circuitry monitors the voltage level of the power supplies 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) is enabled during configuration download to keep the I/O pins from floating.

The 3.0-V tolerance control circuit allows the I/O pins to be driven by 3.0 V before the power supplies are powered and prevents the I/O pins from driving out before the device enters user mode.

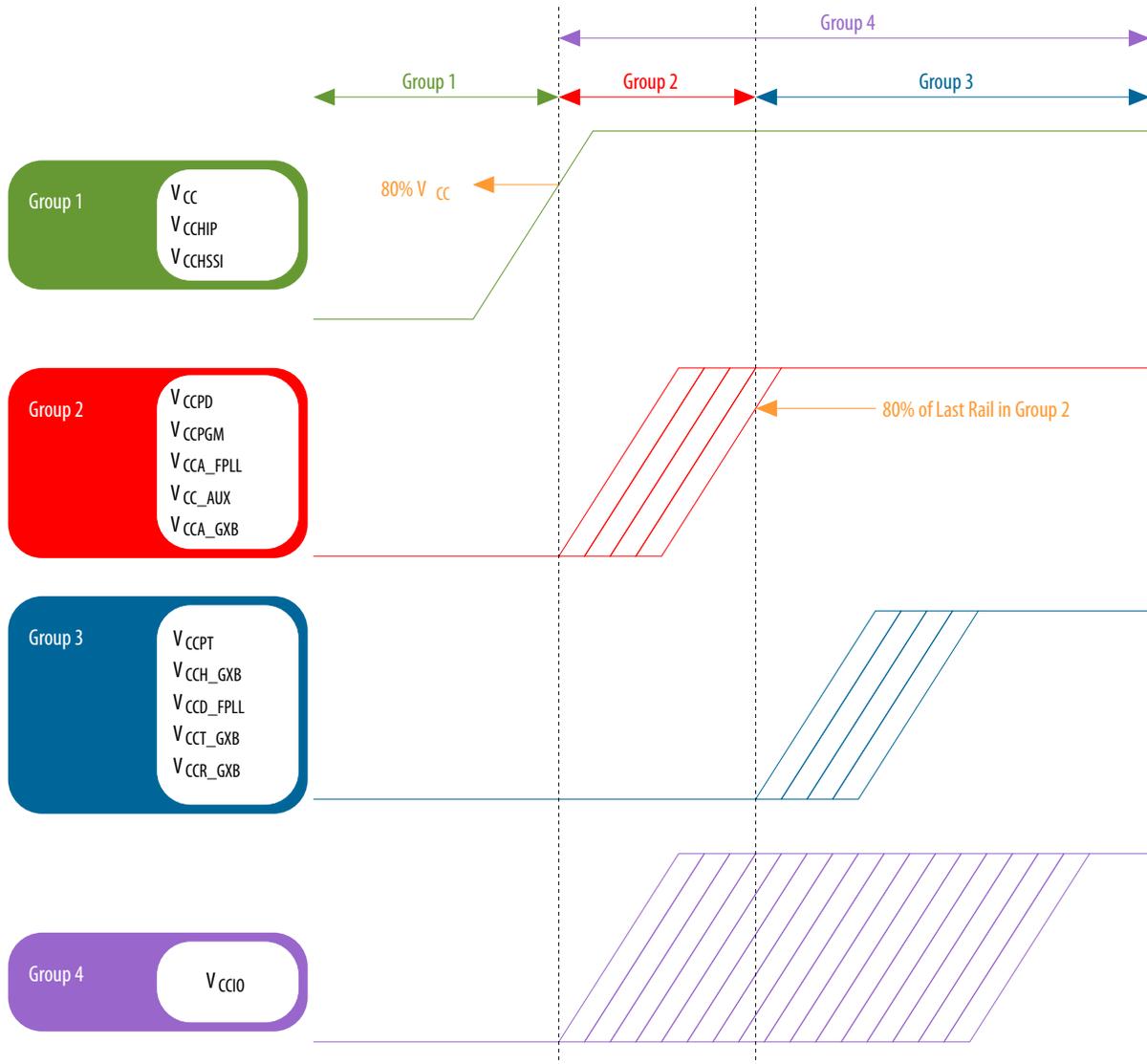
Note: For the V_{CC_AUX} power supply, POR only monitors one of the vcc_aux pins. You must connect all the vcc_aux pins.

Power-Up Sequence

The Stratix V devices require a power-up sequence as shown in the following figure to prevent excessive inrush current and ensure proper transceiver functionality. This power-up sequence is divided into four power groups. Group 1 contains the first power rails to ramp. The V_{CC} , V_{CCCHIP} , and V_{CCHSSI} power rails in this group must ramp to a minimum of 80% of their full rail before any other power rails may start. Group 1 power rails can continue to ramp to full rail. The power rails in Group 2 and Group 4 can start to ramp in any order after Group 1 has reached its minimum 80% threshold. When the last power rail in Group 2 reaches 80% of its full rail, the remaining power rails in Group 3 may start their ramp. During this time, Group 2 power rails may continue to ramp to full rail. Power rails in Group 3 may ramp in any order. All power rails must ramp monotonically. The complete power-up sequence must meet either the standard or fast POR delay time, depending on the POR delay setting that is used.

Figure 11-4: Power-Up Sequence Requirement for Stratix V Devices

Power up V_{CCBAT} at any time. If V_{CC} , V_{CCR_GXB} , and V_{CCT_GXB} have the same voltage level, they can be powered by the same regulator in Group 1 and ramp simultaneously.



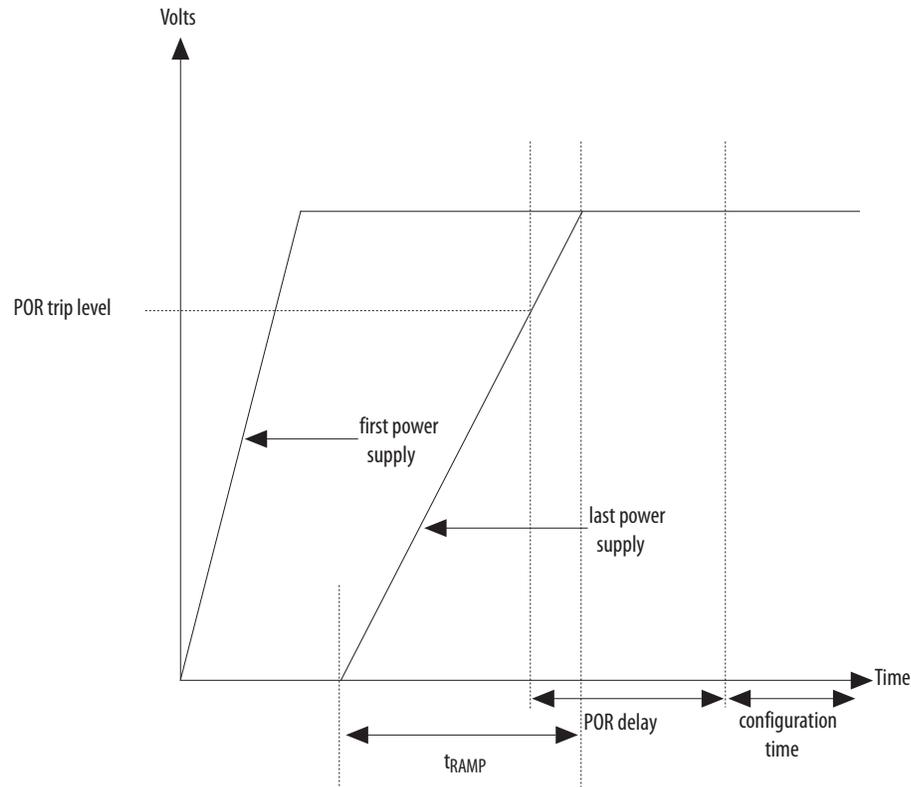
Stratix V devices may power down all power rails simultaneously. However, all rails must reach 0 V within 100 ms from the start of power-down.

Power-On Reset Circuitry

The POR circuitry keeps the Stratix V device in the reset state until the power supply outputs are within the recommended operating range.

A POR event occurs when you power up the Stratix V device until the power supplies reach the recommended operating range within the maximum power supply ramp time, t_{RAMP} . If t_{RAMP} is not met, the Stratix V device I/O pins and programming registers remain tri-stated, during which device configuration could fail.

Figure 11-5: Relationship Between t_{RAMP} and POR Delay

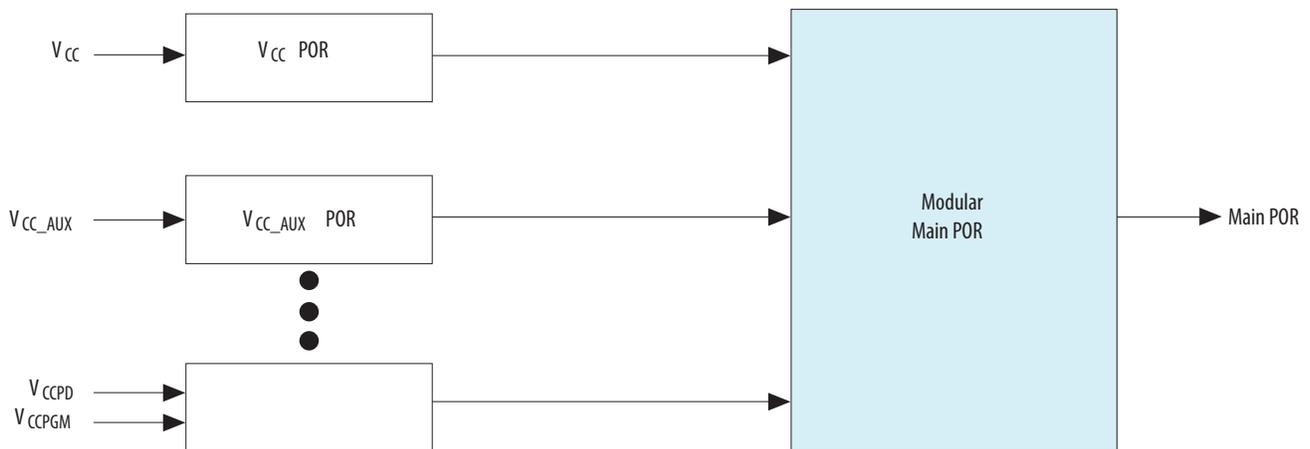


The Stratix V POR circuitry uses an individual detecting circuitry to monitor each of the configuration-related power supplies independently. The main POR circuitry is gated by the outputs of all the individual detectors. The main POR signal is asserted when the power starts to ramp up. This signal is released after the last ramp-up power reaches the POR trip level during power up.

In user mode, the main POR signal is asserted when any of the monitored power goes below its POR trip level. Asserting the POR signal forces the device into the reset state.

The POR circuitry checks the functionality of the I/O level shifters powered by the V_{CCPD} and V_{CCPGM} power supplies during power-up mode. The main POR circuitry waits for all the individual POR circuitries to release the POR signal before allowing the control block to start programming the device.

Figure 11-6: Simplified POR Diagram for Stratix V Devices

**Related Information****[Stratix V Device Datasheet](#)**

Provides more information about the POR delay specification and t_{RAMP} .

Power Supplies Monitored and Not Monitored by the POR Circuitry

Table 11-2: Power Supplies Monitored and Not Monitored by the Stratix V POR Circuitry

Power Supplies Monitored	Power Supplies Not Monitored
<ul style="list-style-type: none"> • V_{CC_AUX} • V_{CCBAT} • V_{CC} • V_{CCPT} • V_{CCPD} • V_{CCPGM} 	<ul style="list-style-type: none"> • V_{CCT_GXB} • V_{CCH_GXB} • V_{CCR_GXB} • V_{CCA_GXB} • V_{CCA_FPLL} • V_{CCD_FPLL} • V_{CCIO} • V_{CCHIP}

Note: For the device to exit POR, you must power the V_{CCBAT} power supply even if you do not use the volatile key.

Related Information**[MSEL Pin Settings](#)**

Provides more information about the $MSEL$ pin settings for each POR delay.

Power Management in Stratix V Devices Revision History

Date	Version	Changes
September 2020	2020.09.10	Updated Figure: <i>Power-Up Sequence Requirement for Stratix V Devices</i> .
December 2015	2015.12.21	<ul style="list-style-type: none"> • Changed instances of Quartus II to Quartus Prime. • Updated the External Temperature Sensing Diode section to provide the type of diode used. • Updated the Power-Up Sequence section.
January 2015	2015.01.23	Added links to the Stratix V Design Guidelines and Stratix V GT Design Guidelines.
May 2013	2013.05.06	<ul style="list-style-type: none"> • Added link to the known document issues in the Knowledge Base. • Moved all links to the Related Information section of respective topics for easy reference. • Added 'There are two supplies which make up the V_{CC} supply. They are V_{CCL} (core V_{CC}) and V_{CCP} (periphery V_{CC}). The sum of I_{CCL} and I_{CCP} equals to I_{CC}. You can refer to the Stratix V PowerPlay Early Power Estimators (EPE) and Power Analyzer for I_{CCL} and I_{CCP} on the EPE report tab.' to 'For detailed information about power supply design requirements, refer to the Board Design Resource Center page.' • Updated dynamic power in Power Consumption for improve clarity. • Added description on powering down unused PCIe HIPS in Programmable Power Technology • Updated Hot-Socketing Feature with ' When powering up these power supplies, you must follow the required power-up sequence as shown in the Power-Up Sequence section of this handbook.'
December 2012	2012.12.28	<ul style="list-style-type: none"> • Consolidated content from the Hot Socketing and Power-On Reset in Stratix V Devices chapter. • Reorganized content and updated template.
June 2012	1.3	Minor text edits.
May 2011	1.2	Chapter moved to volume 2 for the Quartus II software 11.0 release.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1 release.
July 2010	1.0	Initial release.