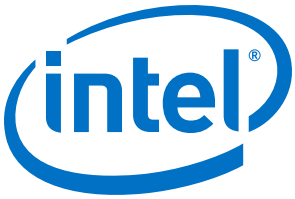


Intel® Arria® 10 Core Fabric and General Purpose I/Os Handbook

A10-HANDBOOK | 2020.06.30
Latest document on the web: PDF | HTML

Contents

1. Logic Array Blocks and Adaptive Logic Modules in Intel® Arria® 10 Devices

The logic array block (LAB) is composed of basic building blocks known as adaptive logic modules (ALMs). You can configure the LABs to implement logic functions, arithmetic functions, and register functions.

You can use a quarter of the available LABs in the Intel Arria 10 devices as a memory LAB (MLAB). Certain devices may have higher MLAB ratio.

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

Intel Arria 10 Device Handbook: Known Issues
Lists the planned updates to the Intel Arria 10 Device Handbook chapters.

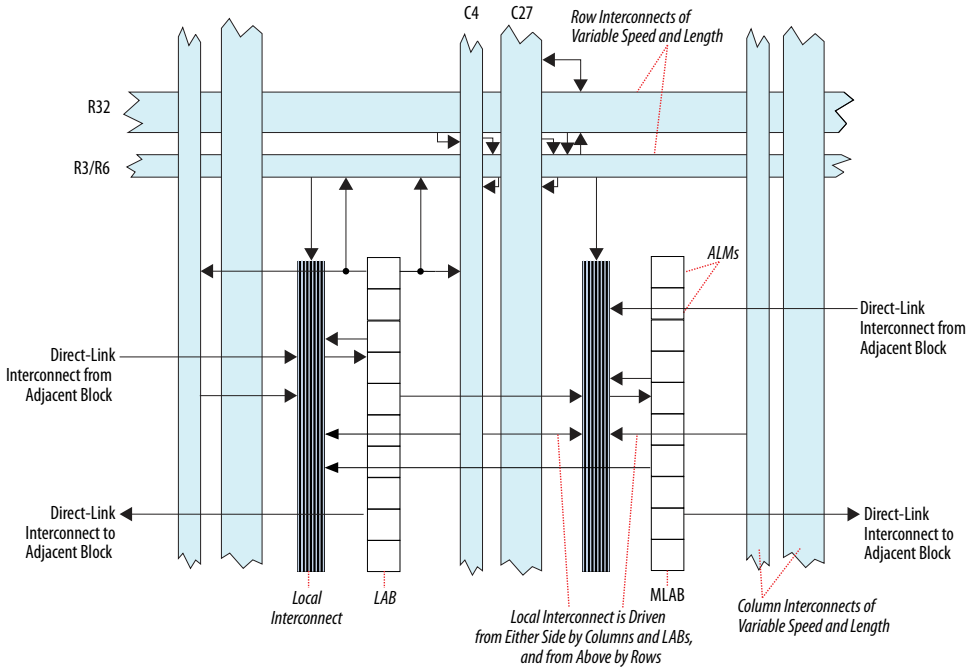
1.1. LAB

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.

Figure 1. LAB Structure and Interconnects Overview in Intel Arria 10 Devices

This figure shows an overview of the Intel Arria 10 LAB and MLAB structure with the LAB interconnects.



1.1.1. MLAB

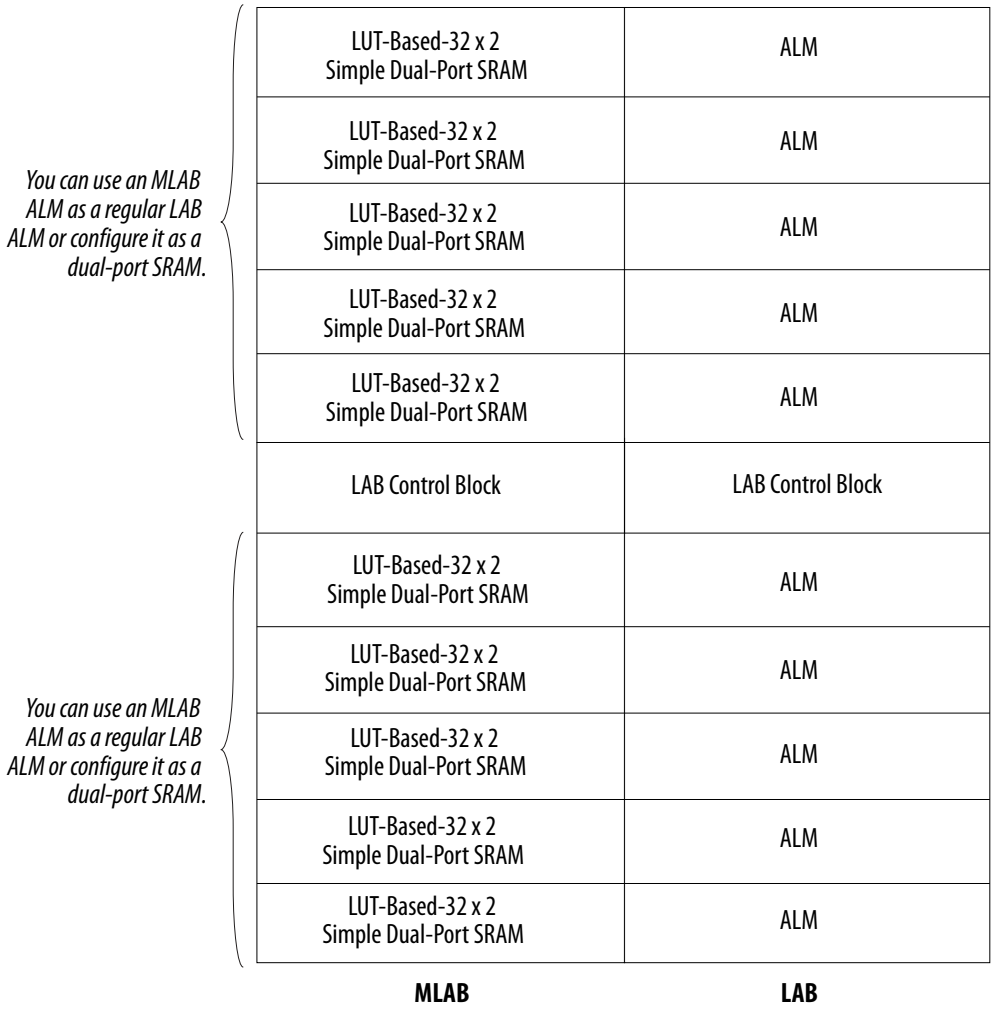
Each MLAB supports a maximum of 640 bits of simple dual-port SRAM.

You can configure each ALM in an MLAB as a 32 (depth) × 2 (width) memory block, resulting in a configuration of 32 (depth) × 20 (width) simple dual-port SRAM block.

MLAB supports the following 64-deep modes in soft implementation using the Intel Quartus Prime software:

- 64 (depth) × 8 (width)
- 64 (depth) × 9 (width)
- 64 (depth) × 10 (width)

Figure 2. LAB and MLAB Structure for Intel Arria 10 Devices



1.1.2. Local and Direct Link Interconnects

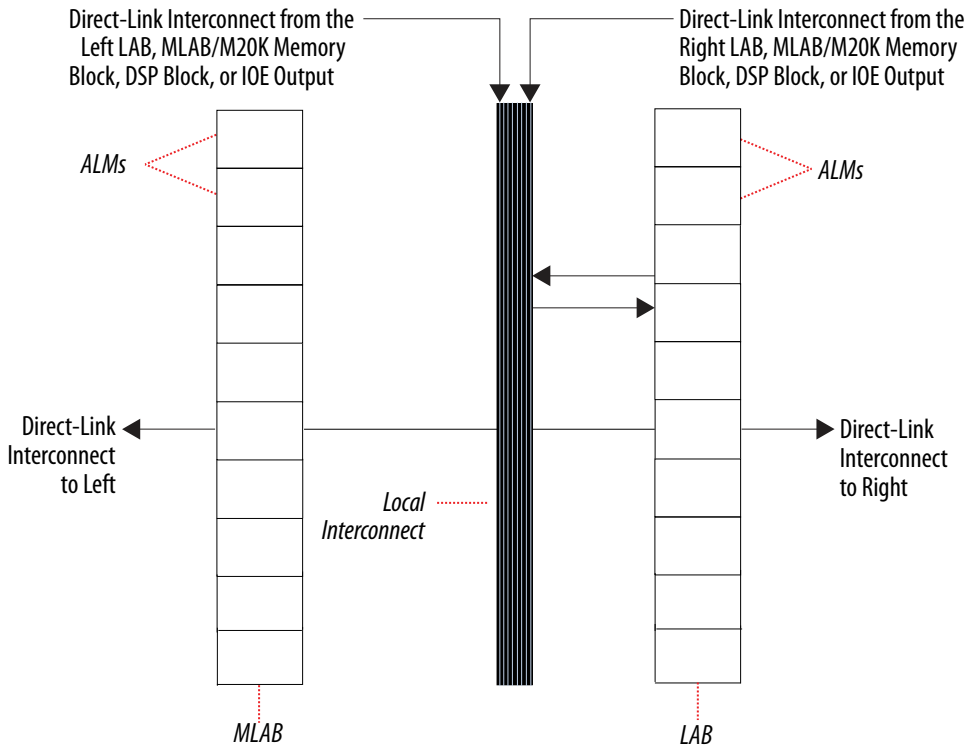
Each LAB can drive out 40 ALM outputs. Two groups of 20 ALM outputs can drive the adjacent LABs directly through direct-link interconnects.

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

The local interconnect drives 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.

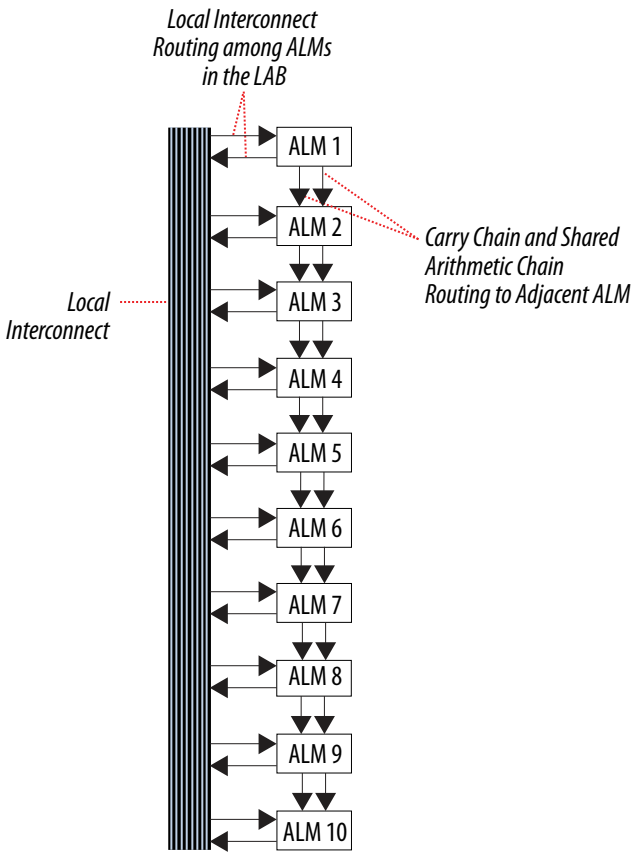
Figure 3. LAB Local and Direct Link Interconnects for Intel Arria 10 Devices



1.1.3. Shared Arithmetic Chain and Carry Chain Interconnects

There are two dedicated paths between ALMs—carry chain and shared arithmetic chain. Intel Arria 10 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 4. Shared Arithmetic Chain and Carry Chain Interconnects



1.1.4. 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.

Deasserting the clock enable signal turns off the corresponding LAB-wide clock.

The LAB row clocks [5..0] and LAB local interconnects generate the LAB-wide control signals. The inherent low skew of the MultiTrack interconnect 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.

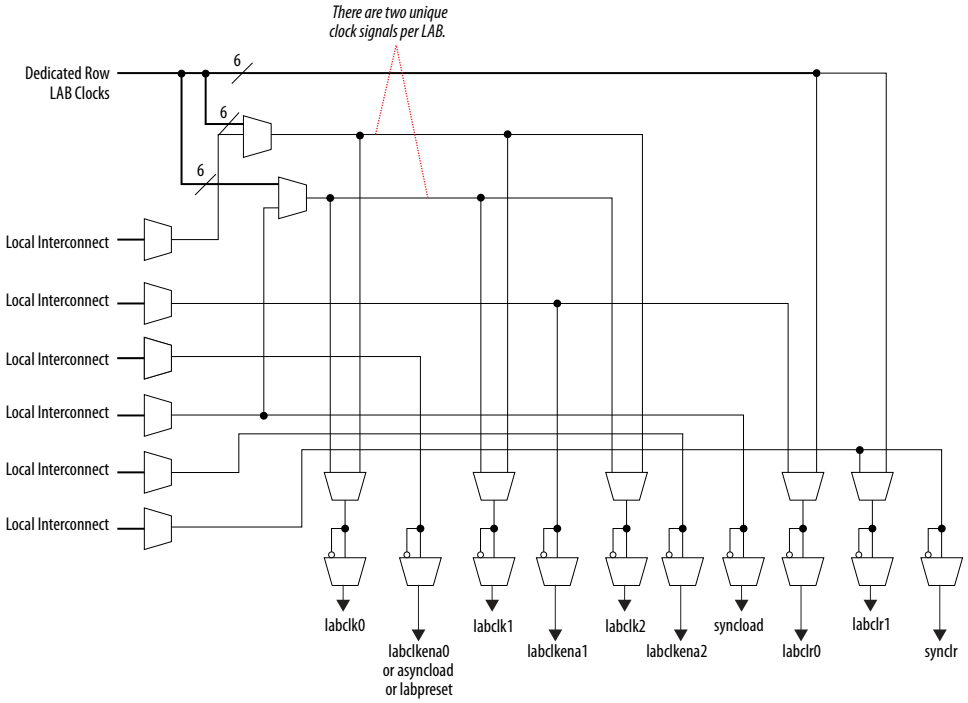
Clear and Preset Logic Control

LAB-wide signals control the logic for the register's clear signal. The ALM directly supports an asynchronous clear function. The register preset is implemented as the **NOT-gate push-back** logic in the Intel Quartus Prime software. Each LAB supports up to two clears.

Intel Arria 10 devices provide a device-wide reset pin (DEV_CLRn) that resets all the registers in the device. You can enable the DEV_CLRn pin in the Intel Quartus Prime software before compilation. The device-wide reset signal overrides all other control signals.

Figure 5. LAB-Wide Control Signals for Intel Arria 10 Devices

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



1.1.5. 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:

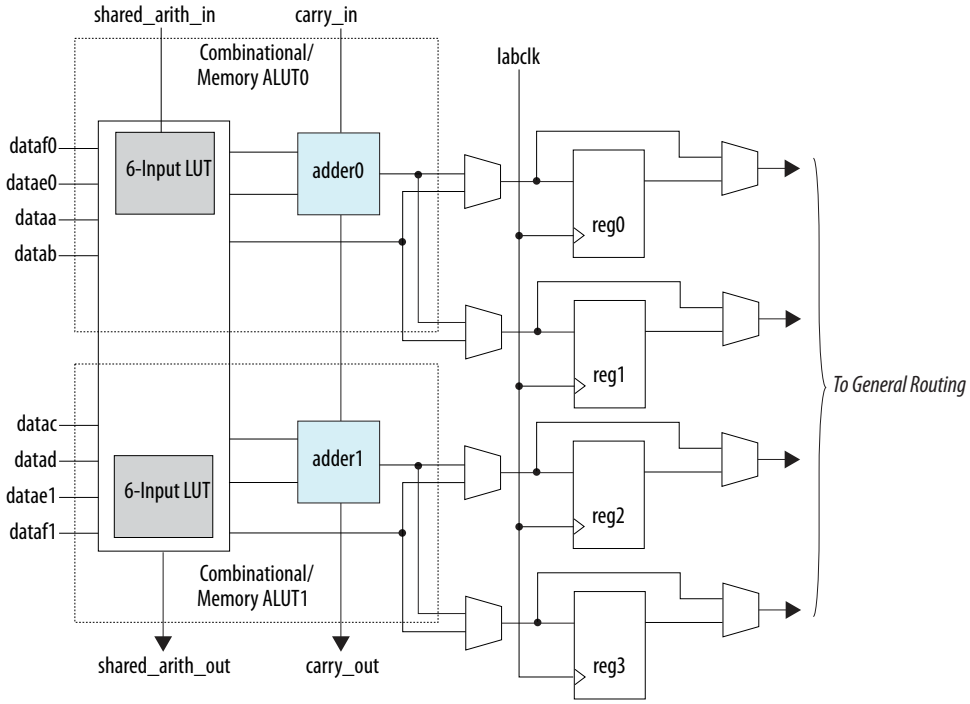
- Data
- Clock
- Synchronous and asynchronous clear
- Synchronous load

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

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 6. ALM High-Level Block Diagram for Intel Arria 10 Devices



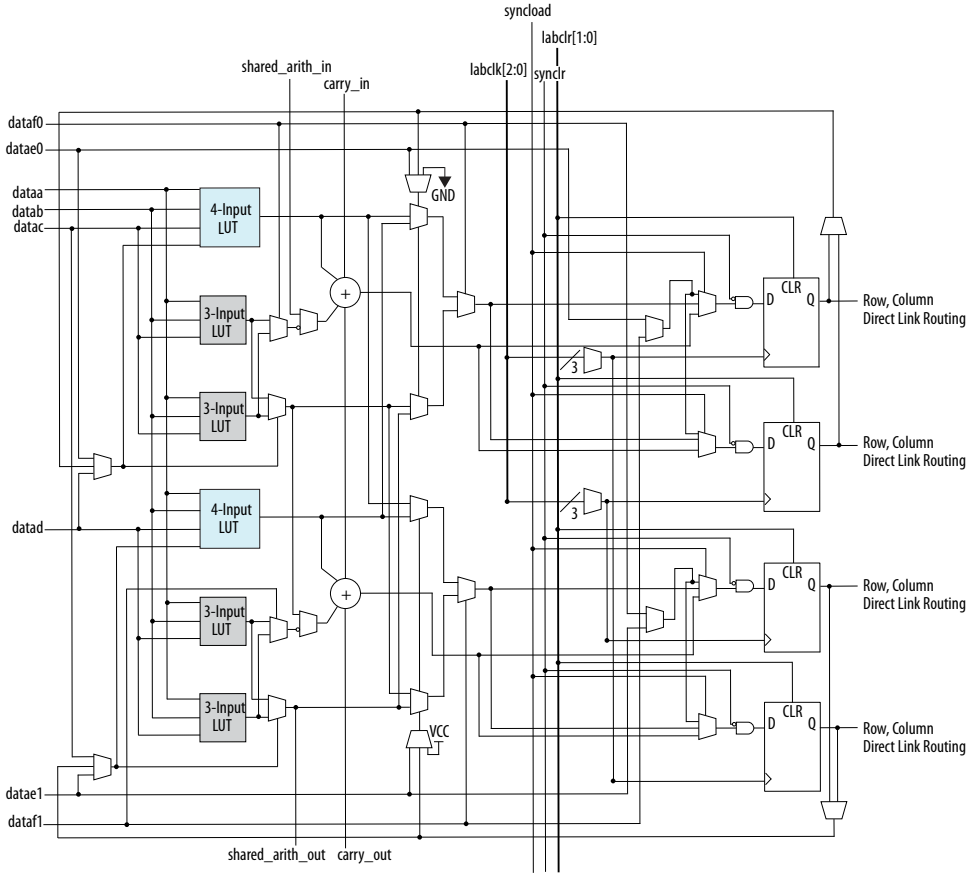
1.1.6. ALM Output

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.

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 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 7. ALM Connection Details for Intel Arria 10 Devices



1.2. ALM Operating Modes

The Intel Arria 10 ALM operates in any of the following modes:

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

1.2.1. Normal Mode

Normal mode allows two functions to be implemented in one Intel Arria 10 ALM, or a single function of up to six inputs.

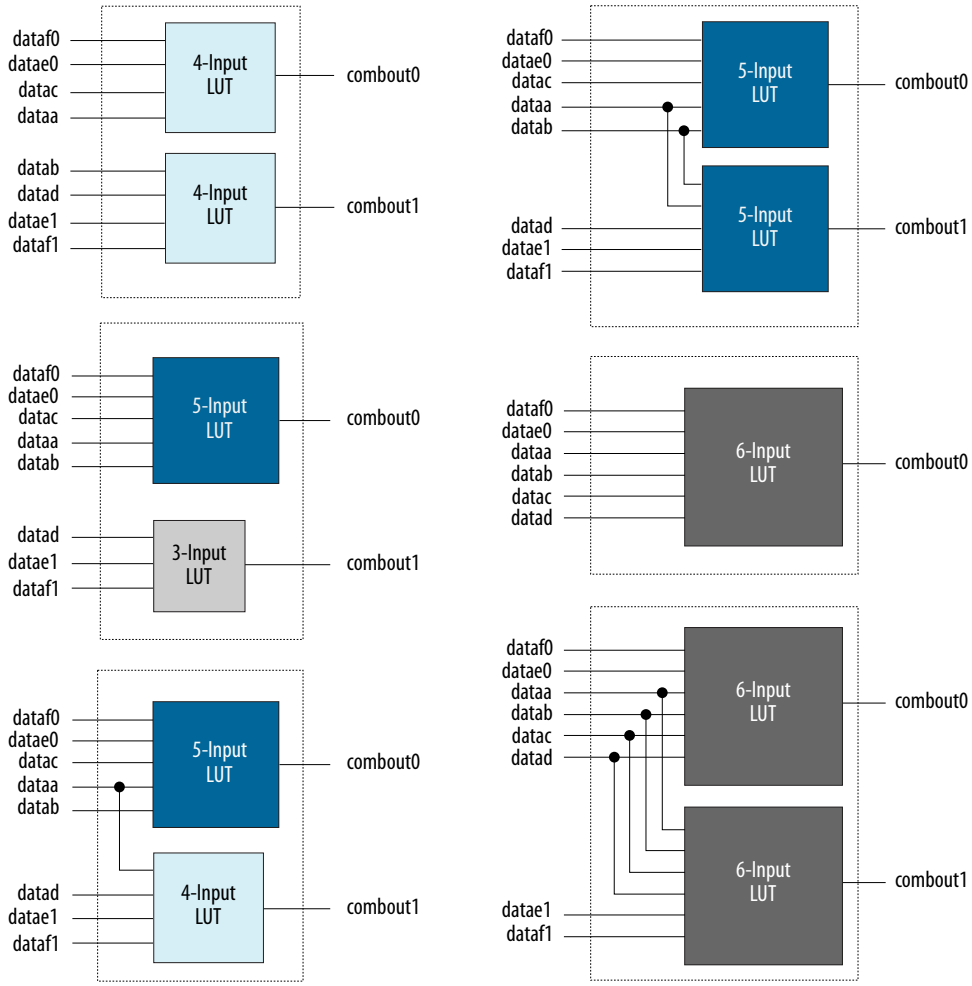
Up to eight data inputs from the LAB local interconnect are inputs to the combinational logic.

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

The Intel Quartus Prime Compiler automatically selects the inputs to the LUT. ALMs in normal mode support register packing.

Figure 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: four and three, three and three, three and two, and five and two.

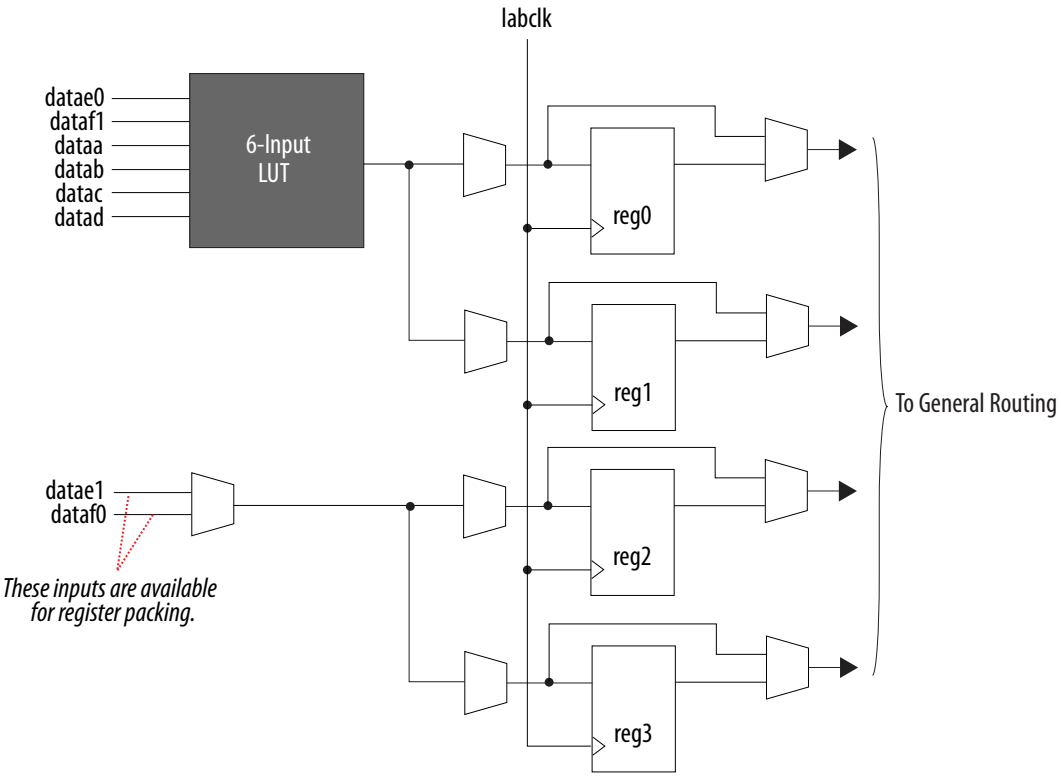


For the packing of two five-input functions into one ALM, the functions must have at least two common inputs. The common inputs are `dataa` and `datab`. The combination of a four-input function with a five-input function requires one common input (either `dataa` or `datab`).

In the case of implementing two 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 Intel Arria 10 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.

Figure 9. Input Function in Normal Mode



You can implement any six-input function using the following inputs:

- `dataa`
- `datab`
- `datac`
- `datad`
- `datae0` and `dataf1`, or `datae1` and `dataf0`

If you use `datae0` and `dataf1` inputs, you can obtain the following outputs:

- Output driven to `register0` or `register0` is bypassed
- Output driven to `register1` or `register1` is bypassed

You can use the `datae1` or `dataf0` input, whichever is available, as the packed register input to `register2` or `register3`.

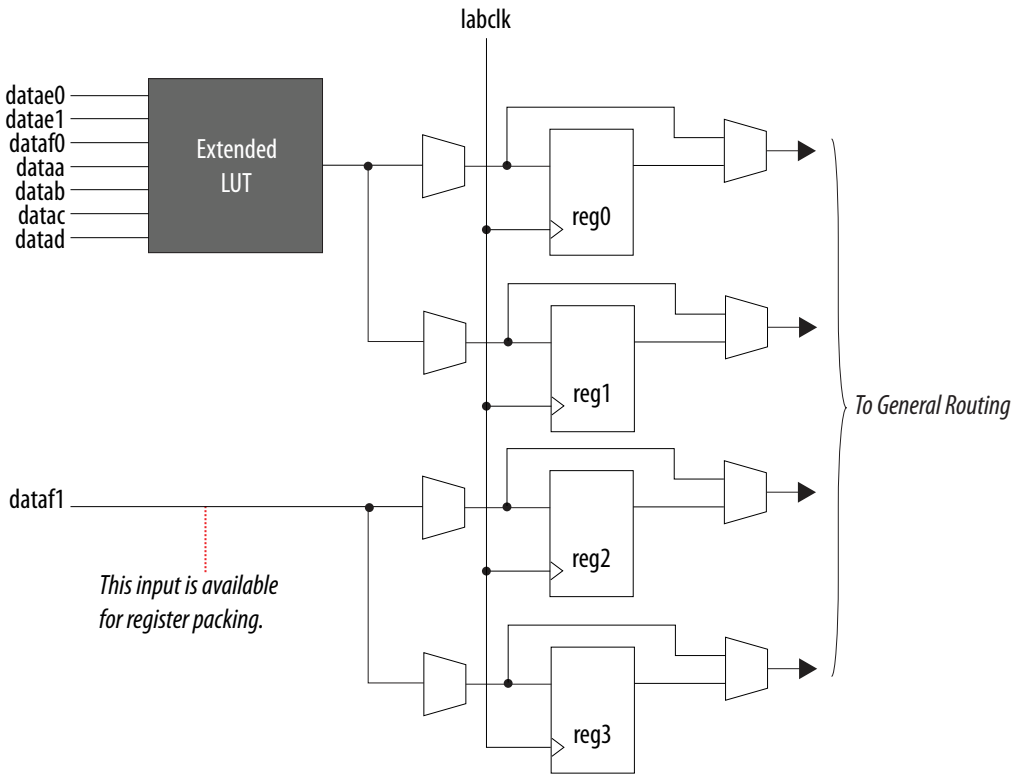
If you use `datae1` and `dataf0` inputs, you can obtain the following outputs:

- Output driven to `register2` or `register2` is bypassed
- Output driven to `register3` or `register3` is bypassed

You can use the `datae0` or `dataf1` input, whichever is available, as the packed register input to `register0` or `register1`.

1.2.2. Extended LUT Mode

Figure 10. Template for Supported Seven-Input Functions in Extended LUT Mode for Intel Arria 10 Devices



A seven-input function can be implemented in a single ALM using the following inputs:

- `dataa`
- `datab`
- `datac`
- `datad`
- `datae0`
- `datae1`
- `dataf0` or `dataf1`

If you use `dataf0` input, you can obtain the following outputs:

- Output driven to `register0` or `register0` is bypassed
- Output driven to `register1` or `register1` is bypassed

You can use the `dataf1` input as the packed register input to `register2` or `register3`.

If you use `dataf1` input, you can obtain the following outputs:

- Output driven to `register2` or `register2` is bypassed
- Output driven to `register3` or `register3` is bypassed

You can use the `dataf0` input as the packed register input to `register0` or `register1`.

1.2.3. Arithmetic Mode

The ALM in arithmetic mode uses two sets of two four-input LUTs along with two dedicated full adders.

The dedicated adders allow the LUTs to perform pre-adder logic; therefore, each adder can add the output of two four-input functions.

The ALM supports simultaneous use of the adder's carry output along with combinational logic outputs. The adder output is ignored in this operation.

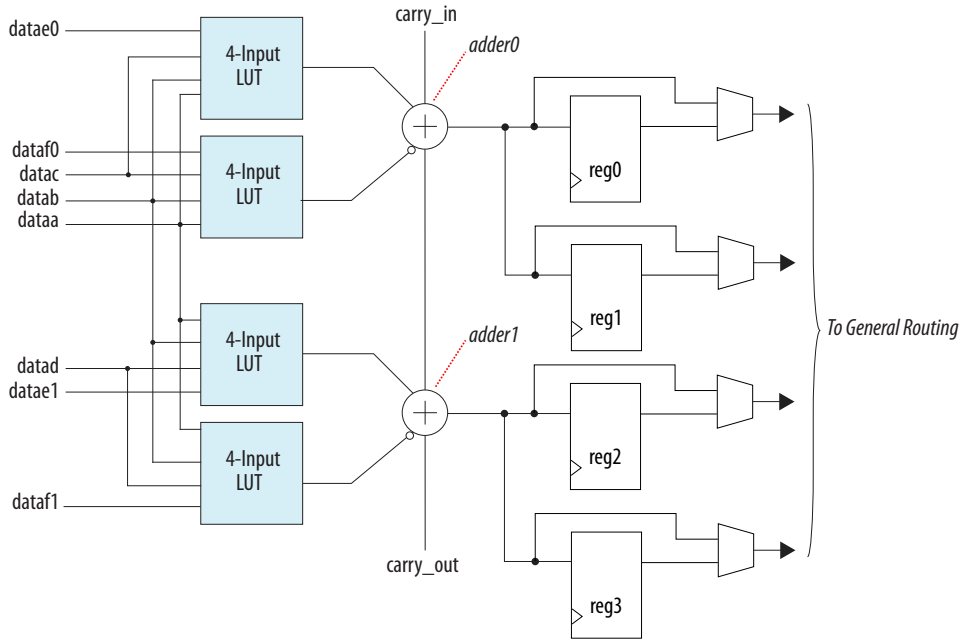
Using the adder with the combinational logic output provides resource savings of up to 50% for functions that can use this mode.

Arithmetic mode also offers clock enable, counter enable, synchronous up and down control, add and subtract control, synchronous clear, and synchronous load.

The LAB local interconnect data inputs generate the clock enable, counter enable, synchronous up/down, and add/subtract control signals. These control signals are good candidates for the inputs that are shared between the four LUTs in the ALM.

The synchronous clear and synchronous load options are LAB-wide signals that affect all registers in the LAB. You can individually disable or enable these signals for each register. The Intel Quartus Prime software automatically places any registers that are not used by the counter into other LABs.

Figure 11. ALM in Arithmetic Mode for Intel Arria 10 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 Intel Arria 10 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 if 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.

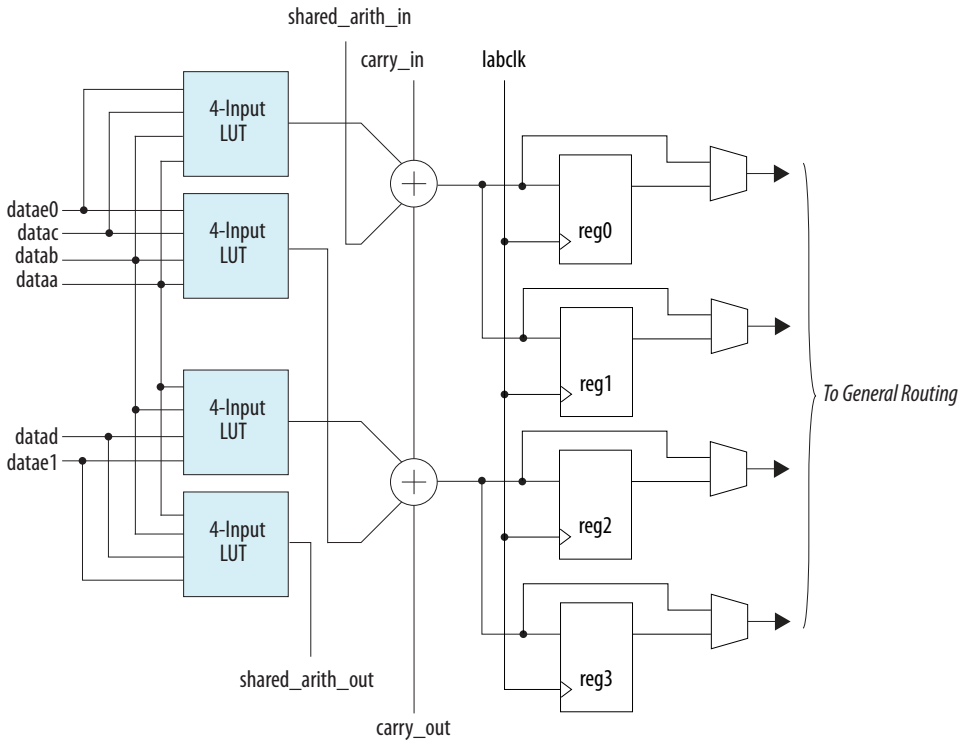
The Intel Quartus Prime Compiler creates carry chains longer than 20 ALMs (10 ALMs in arithmetic or shared arithmetic mode) by linking LABs together automatically. For enhanced fitting, a long carry chain runs vertically, allowing fast horizontal connections to the TriMatrix memory and DSP blocks. A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

1.2.4. Shared Arithmetic Mode

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

This mode configures the ALM with four four-input LUTs. Each LUT either computes the sum of three inputs or the carry of three inputs. The output of the carry computation is fed to the next adder using a dedicated connection called the shared arithmetic chain.

Figure 12. ALM in Shared Arithmetic Mode for Intel Arria 10 Devices



Shared Arithmetic Chain

The shared arithmetic chain available in enhanced arithmetic mode allows the ALM to implement a four-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.

1.3. LAB Power Management Techniques

Use the following techniques to manage static and dynamic power consumption within the LAB:

- Intel Arria 10 LABs operate in high-performance mode or low-power mode. The Intel Quartus Prime software automatically optimizes the LAB power consumption mode based on your design.
- Clocks, especially LAB clocks, consumes a significant portion of dynamic power. Each LAB's clock and clock enable signals are linked and can be controlled by a shared, gated clock. Use the LAB-wide clock enable signal to gate the LAB-wide clock without disabling the entire clock tree. In your HDL code for registered logic, use a clock-enable construct.

Related Information

Power Optimization chapter, Intel Quartus Prime Handbook
Provides more information about implementing static and dynamic power consumption within the LAB.

1.4. Logic Array Blocks and Adaptive Logic Modules in Intel Arria 10 Devices Revision History

Document Version	Changes
2019.12.30	Updated the following signal names in *ALM Connection Details for Intel Arria 10 Devices* figure: • `aclr[1:0]` to `labclr[1:0]` • `sclr` to `synclr` • `clk[2:0]` to `labclk[2:0]`

Date	Version	Changes
March 2017	2017.03.15	Rebranded as Intel.
October 2016	2016.10.31	Added description on clock source in the LAB Control Signals section.
November 2015	2015.11.02	Changed instances of *Quartus II* to *Quartus Prime*.
December 2013	2013.12.02	Initial release.

2. Embedded Memory Blocks in Intel Arria 10 Devices

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

Arria 10 Device Handbook: Known Issues
Lists the planned updates to the *Intel Arria 10 Device Handbook* chapters.

2.1. Types of Embedded Memory

The Intel Arria 10 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 and shallow FIFO buffers, and filter delay lines. Each MLAB is made up of ten adaptive logic modules (ALMs). In the Intel Arria 10 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.

Related Information

embedded cell (EC)
Provides information about embedded cell

2.1.1. Embedded Memory Capacity in Intel Arria 10 Devices

Table 1. Embedded Memory Capacity and Distribution in Intel Arria 10 Devices

Variant	Product Line	M20K		MLAB		Total RAM Bit (Kb)
		Block	RAM Bit (Kb)	Block	RAM Bit (Kb)	
Intel Arria 10 GX	GX 160	440	8,800	1,680	1,050	9,850
	GX 220	587	11,740	2,703	1,690	13,430
	GX 270	750	15,000	3,922	2,452	17,452
	GX 320	891	17,820	4,363	2,727	20,547
	GX 480	1,431	28,620	6,662	4,164	32,784
	GX 570	1,800	36,000	8,153	5,096	41,096
	GX 660	2,131	42,620	9,260	5,788	48,408
	GX 900	2,423	48,460	15,017	9,386	57,846
	GX 1150	2,713	54,260	20,774	12,984	67,244
Intel Arria 10 GT	GT 900	2,423	48,460	15,017	9,386	57,846
	GT 1150	2,713	54,260	20,774	12,984	67,244
Intel Arria 10 SX	SX 160	440	8,800	1,680	1,050	9,850
	SX 220	587	11,740	2,703	1,690	13,430
	SX 270	750	15,000	3,922	2,452	17,452
	SX 320	891	17,820	4,363	2,727	20,547
	SX 480	1,431	28,620	6,662	4,164	32,784
	SX 570	1,800	36,000	8,153	5,096	41,096
	SX 660	2,131	42,620	9,260	5,788	48,408

2.2. Embedded Memory Design Guidelines for Intel Arria 10 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.

2.2.1. 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 parameter editor.

For the MLABs, 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 registers, output registers, and write address registers are available in the block. The MLABs gain read address registers from the ALMs.

Note: For Intel Arria 10 devices, the Resource Property Editor and the Timing Analyzer report the location of the M20K block as EC_X<number>_Y<number>_N<number>, although the allowed assigned location is M20K_X<number>_Y<number>_N<number>. Embedded Cell (EC) is the sublocation of the M20K block.

2.2.2. Guideline: Implement External Conflict Resolution

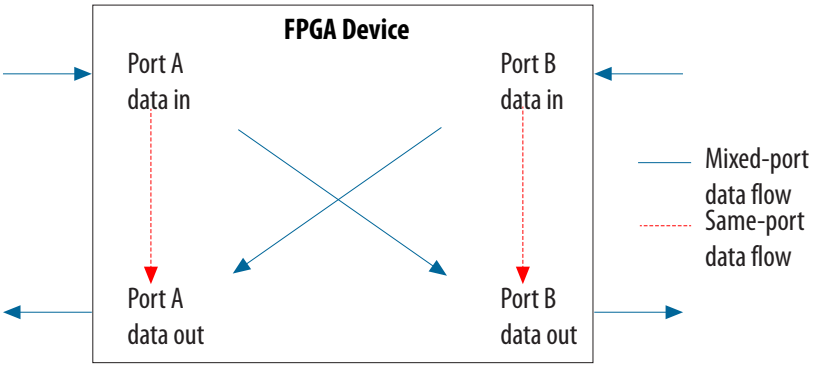
In the true dual-port RAM mode, you can perform two write operations to the same memory location. However, the memory blocks do not have internal conflict resolution circuitry. To avoid unknown data being written to the address, implement external conflict resolution logic to the memory block.

2.2.3. Guideline: Customize Read-During-Write Behavior

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

Figure 13. 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.



2.2.3.1. 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.

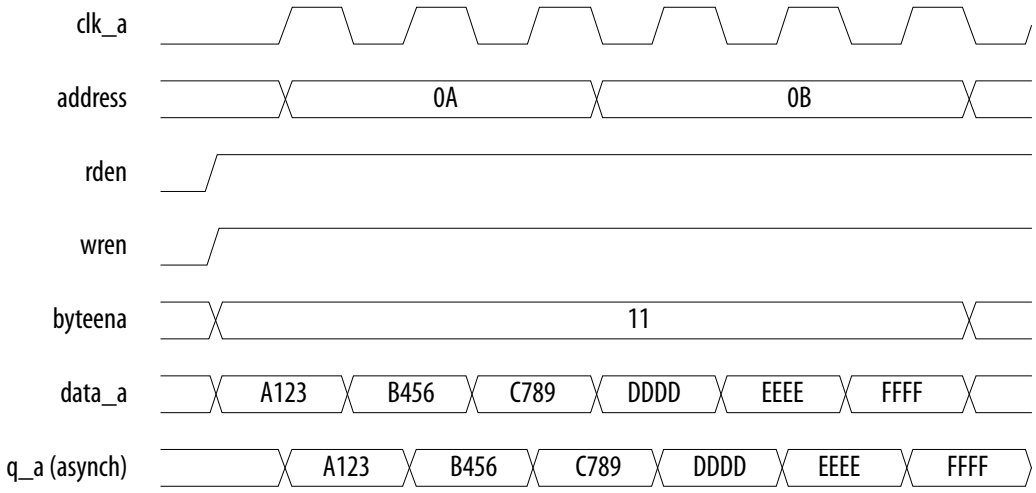
Table 2. Output Modes for Embedded Memory Blocks in Same-Port Read-During-Write Mode

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"	M20K, MLAB	The RAM outputs "don't care" values for a read-during-write operation.

Figure 14. 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.



2.2.3.2. 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 3. Output Modes for RAM in Mixed-Port Read-During-Write Mode

Output Mode	Memory Type	Description
"new data"	MLAB	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. This mode is available only if the output is registered.
"old data"	M20K, MLAB	A read-during-write operation to different ports causes the RAM output to reflect the "old data" value at the particular address. For MLAB, this mode is available only if the output is registered.
"don't care"	M20K, MLAB	The RAM outputs "don't care" or "unknown" value. • 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	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.

Figure 15. 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.

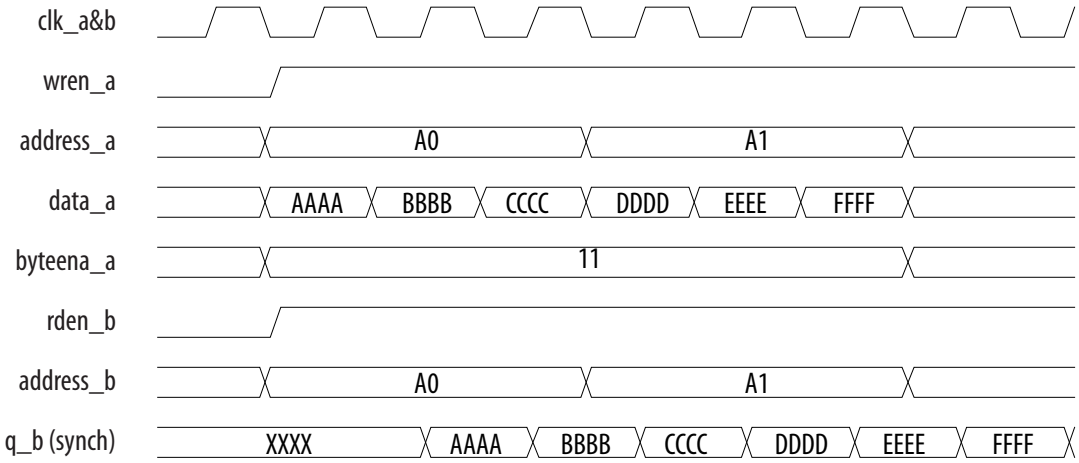


Figure 16. Mixed-Port Read-During-Write: Old Data Mode

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

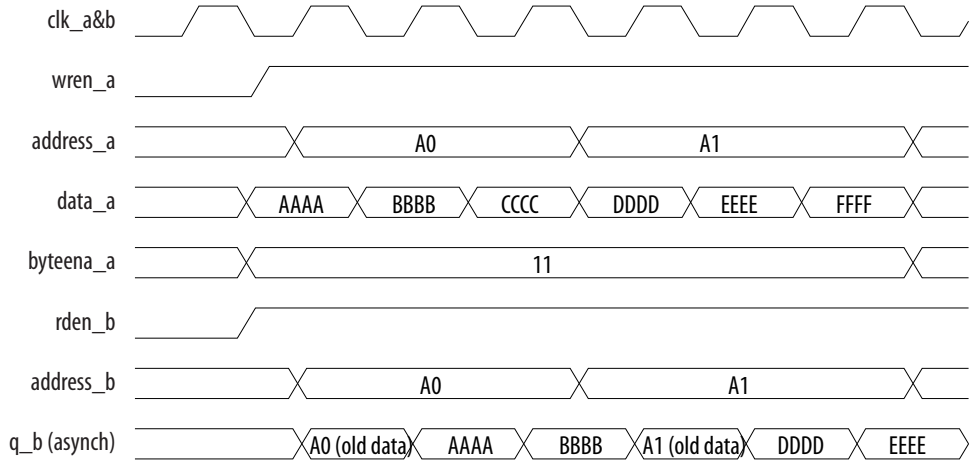
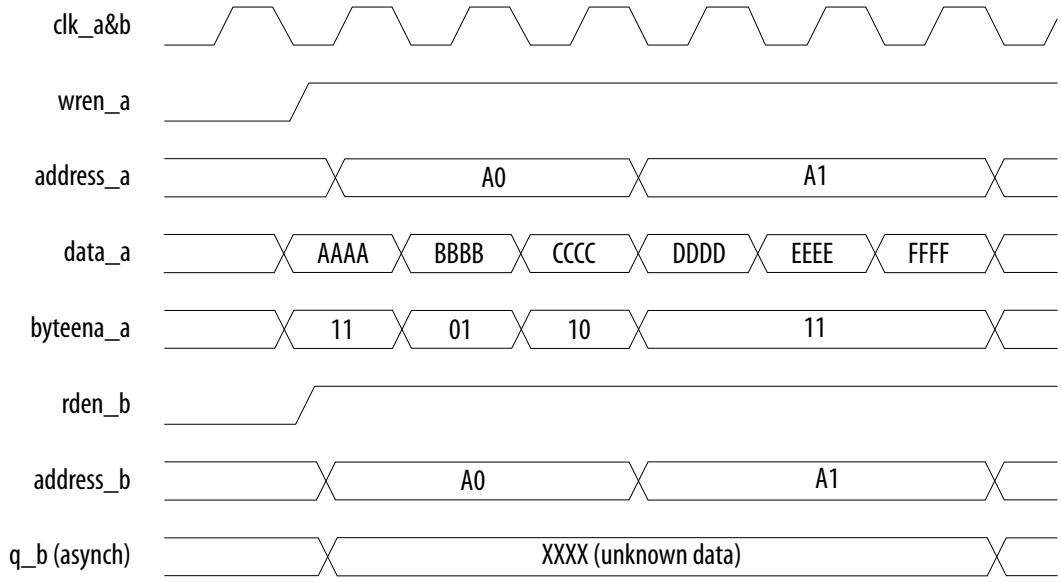


Figure 17. Mixed-Port Read-During-Write: Don't Care or Constrained Don't Care Mode

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.

Related Information

Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide

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

2.2.4. 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 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 Intel Arria 10 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) User Guide
  Provides more information about `.mif` files.
- Quartus Prime Handbook Volume 1: Design and Synthesis
  Provides more information about **.mif** files.

2.2.5. Guideline: Control Clocking to Reduce Power Consumption

Reduce AC power consumption of each memory block in your design:

- Use Intel Arria 10 memory block clock-enables to allow you to control 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 deasserting 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.

2.3. Embedded Memory Features

Table 5. Memory Features in Intel Arria 10 Devices

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

Features	M20K	MLAB
Maximum operating frequency	730 MHz	700 MHz
Total RAM bits (including parity bits)	20,480	640
Parity bits	Supported	—
Byte enable	Supported	Supported
Packed mode	Supported	—
Address clock enable	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.

continued...

Features	M20K	MLAB
Power-up state	Output ports are cleared.	• 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" or "don't care".	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	Soft IP support using the Intel Quartus Prime software. Built-in support in x32-wide simple dual-port mode.	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) User Guide
Provides more information about the embedded memory features.

2.4. Embedded Memory Modes

Table 6. Memory Modes Supported in the Embedded Memory Blocks

This table lists and describes the memory modes that are supported in the Intel Arria 10 embedded memory blocks.

Memory Mode	M20K Support	MLAB Support	Description
Single-port RAM	Yes	Yes	You can perform only one read or one write operation at a time. Use the read enable port to control the RAM output ports behavior during a write operation: • 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	You can use the memory blocks as a shift-register block to save logic cells and routing resources. 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.

continued...

Memory Mode	M20K Support	MLAB Support	Description
			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 × m × n). You can cascade memory blocks to implement larger shift registers.
ROM	Yes	Yes	You can use the memory blocks as ROM. • 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	You can use the memory blocks as FIFO buffers. Use the SCFIFO and DCFIFO megafunctions to implement single- and dual-clock asynchronous FIFO buffers in your design. 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.

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.

Related Information

- Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Provides more information about 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.

2.4.1. Embedded Memory Configurations for Single-port Mode

Table 7. Single-port Embedded Memory Configurations for Intel Arria 10 Devices

This table lists the maximum configurations supported for single-port RAM and ROM modes.

Memory Block	Depth (bits)	Programmable Width
MLAB	32	x16, x18, or x20
	64 (1)	x8, x9, x10
M20K	512	x40, x32
	1K	x20, x16
	2K	x10, x8

continued...

(1) Supported through software emulation and consumes additional MLAB blocks.

Memory Block	Depth (bits)	Programmable Width
	4K	x5, x4
	8K	x2
	16K	x1

2.4.2. Embedded Memory Configurations for Dual-port Modes

Table 8. Memory Configurations for Simple Dual-Port RAM Mode

This table lists the memory configurations for the simple dual-port RAM mode. Mixed-width configurations are only supported in M20K blocks.

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

Table 9. Memory Configurations for True Dual-Port Mode

This table lists the memory configurations for the true dual-port RAM mode. Mixed-width configurations are only supported in M20K blocks.

Port A	Port B							
	16K×1	8K×2	4K×4	4K×5	2K×8	2K×10	1K×16	1K×20
16K×1	Yes	Yes	Yes	—	Yes	—	Yes	—
8K×2	Yes	Yes	Yes	—	Yes	—	Yes	—
4K×4	Yes	Yes	Yes	—	Yes	—	Yes	—
4K×5	—	—	—	Yes	—	Yes	—	Yes
2K×8	Yes	Yes	Yes	—	Yes	—	Yes	—
2K×10	—	—	—	Yes	—	Yes	—	Yes
1K×16	Yes	Yes	Yes	—	Yes	—	Yes	—
1K×20	—	—	—	Yes	—	Yes	—	Yes

2.5. Embedded Memory Clocking Modes

This section describes the clocking modes for the Intel Arria 10 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.

2.5.1. Clocking Modes for Each Memory Mode

Table 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
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.

2.5.1.1. Single Clock Mode

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

2.5.1.2. 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.

2.5.1.3. 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.

2.5.1.4. 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.

2.5.2. 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.

2.5.3. 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 core parameter editor.

2.5.4. 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 28

2.6. Parity Bit in Embedded Memory Blocks

The following describes the parity bit support for M20K blocks:

- 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.

2.7. 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 enable signals are active high.

2.7.1. Byte Enable Controls in Memory Blocks

Table 11. 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 12. 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]

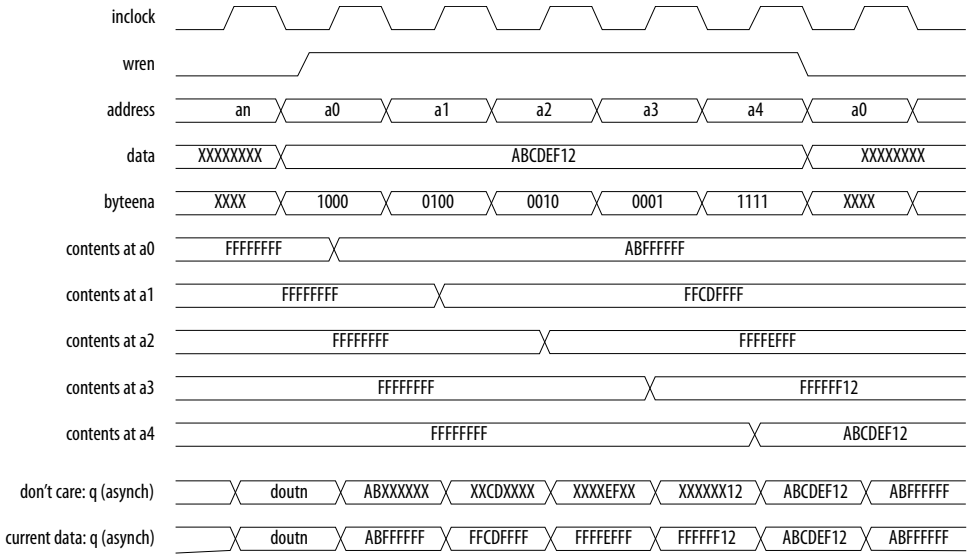
2.7.2. Data Byte Output

In M20K blocks or MLABs, when you set a byte-enable bit to 0, the embedded memory IP sets the corresponding data byte output to a "don't care" value. You must ensure that the option **Get X's for write masked bytes instead of old data when byte enable** is always selected.

2.7.3. RAM Blocks Operations

Figure 18. Byte Enable Functional Waveform

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



2.8. 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.

2.9. 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 19. Address Clock Enable

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

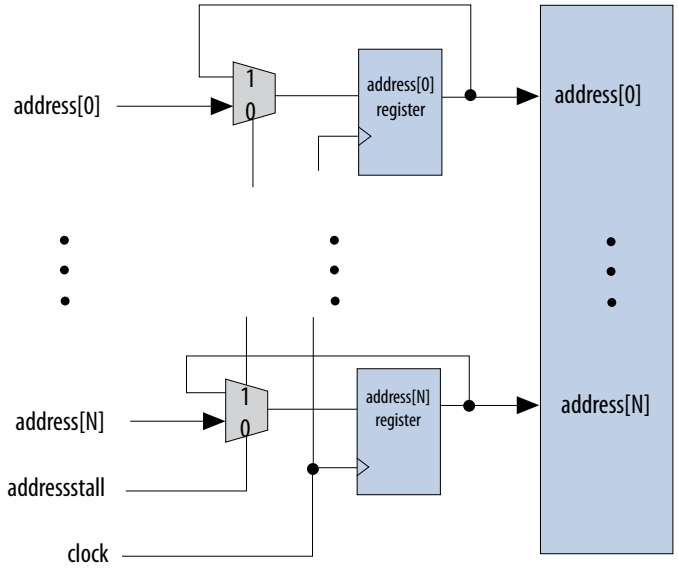


Figure 20. Address Clock Enable During Read Cycle Waveform

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

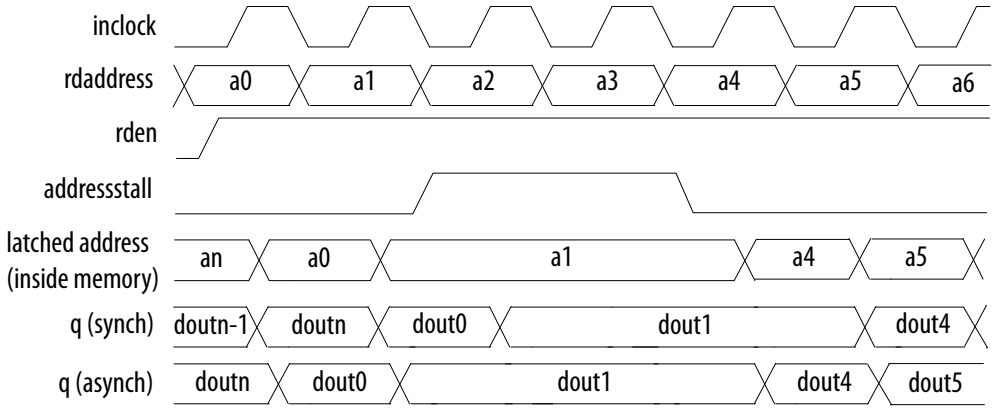
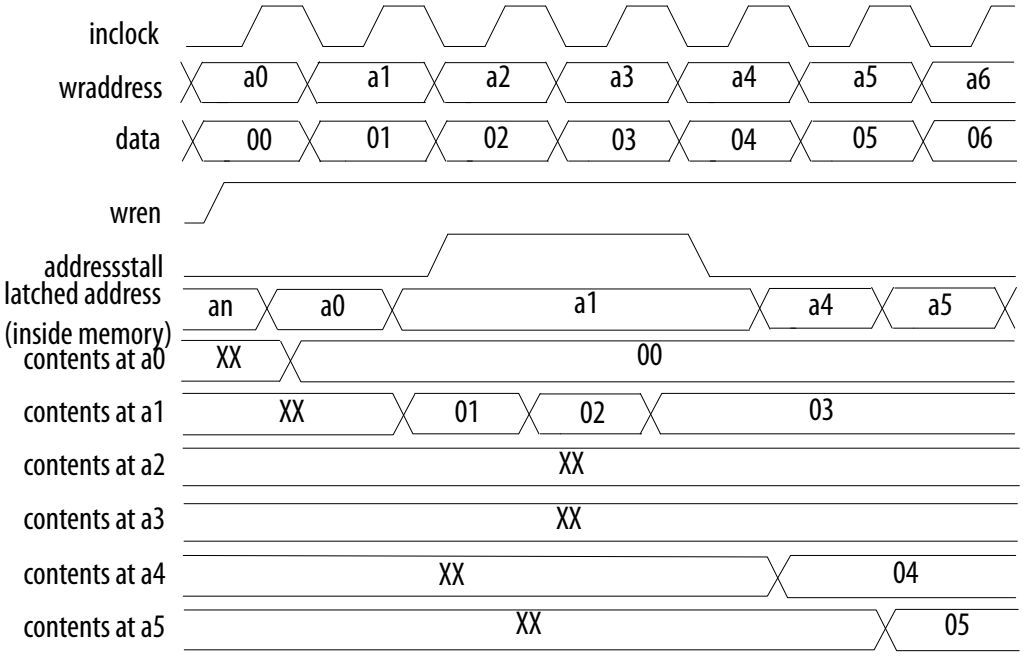


Figure 21. Address Clock Enable During the Write Cycle Waveform

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



2.10. 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 22. Output Latch Clear in Intel Arria 10 Devices (Non-ECC Mode)

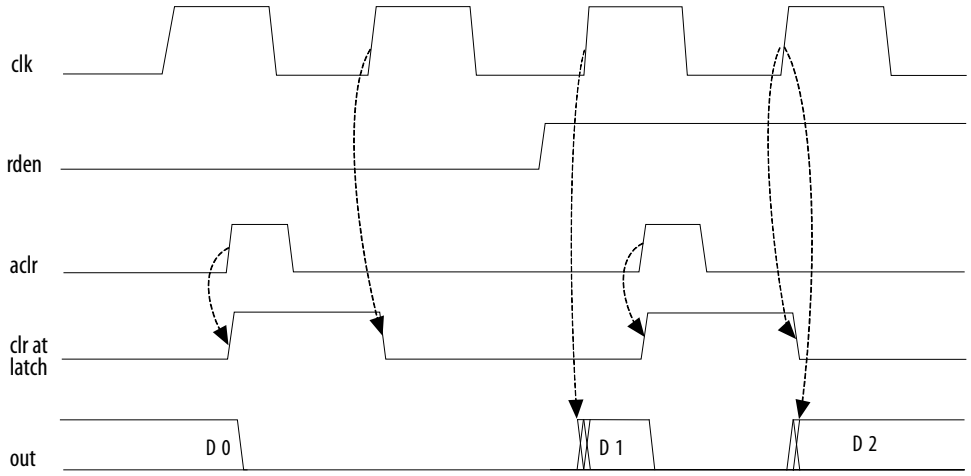
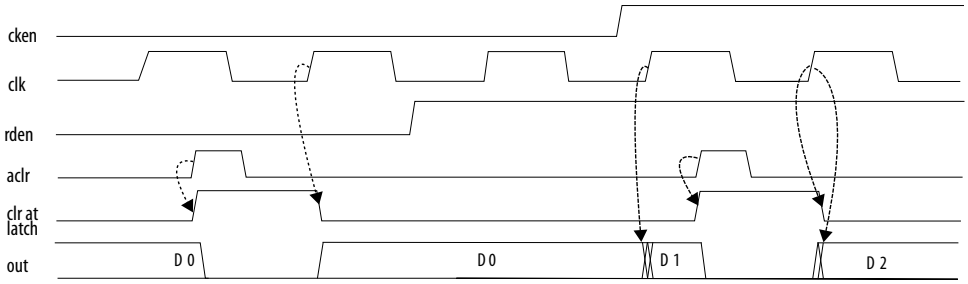


Figure 23. Output Latch Clear in Intel Arria 10 Devices (ECC Mode)



2.11. 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 higher performance compared to non-pipeline ECC 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.

Related Information

Memory Blocks Error Correction Code Support

2.11.1. Error Correction Code Truth Table

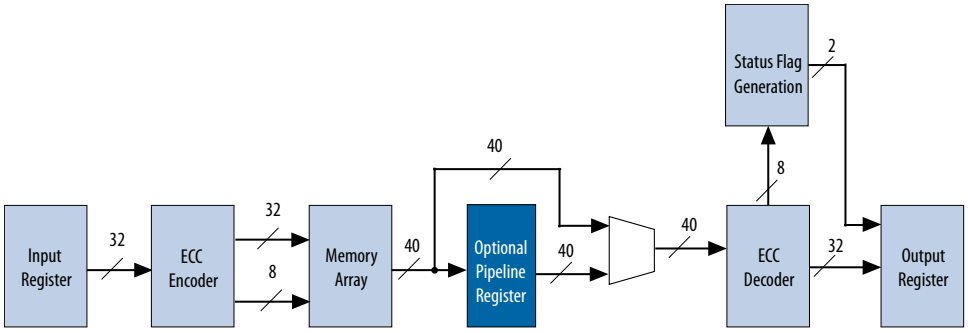
Table 13. ECC Status Flags Truth Table

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.
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 24. ECC Block Diagram for M20K Memory



2.12. Embedded Memory Blocks in Intel Arria 10 Devices Revision History

Date	Version	Changes
December 2017	2017.12.15	• Updated the ECC Block Diagram for M20K Memory figure. • 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.
March 2017	2017.03.15	• Rebranded as Intel. • Removed parity bit support for MLAB under the Error Correction Code (ECC) support feature in the Memory Features in Arria 10 Devices table. • Removed parity bit support for MLAB blocks in the Parity Bit topic.
October 2016	2016.10.31	• Removed Address clock enable support for MLAB block.
December 2015	2015.12.14	• Updated the number of M20K memory blocks for Arria 10 GX 660 from 2133 to 2131 and corrected the total RAM bit from 48,448 Kb to 48,408 Kb.

continued...

Date	Version	Changes
November 2015	2015.11.02	• Updated the following topics: Embedded Memory Configurations for Single-port mode and Embedded Memory Configurations for Dual-port mode. • Updated the description in the Data Byte Output topic. • Updated the Embedded Memory Capacity and Distribution table. • Changed instances of *Quartus II* to *Quartus Prime*.
June 2015	2015.06.15	Updated links.
May 2015	2015.05.04	• Updated Mega Wizard Plug-In manager to IP Core parameter editor. • Updated Megafunction to IP core.
August 2014	2014.08.18	• Added a new timing diagram for output latch clear in ECC mode. • Added a note to clarify that for Intel Arria 10 devices, the Resource Property Editor and the TimeQuest Timing Analyzer report the location of the M20K block as EC_X<number>_Y<number>_N<number> • Updated the RAM bit value in M20K block for Arria 10 GX 660 and Arria 10 SX 660.
December 2013	2013.12.02	Initial release.

3. Variable Precision DSP Blocks in Intel Arria 10 Devices

This chapter describes how the variable-precision digital signal processing (DSP) blocks in Intel Arria 10 devices are optimized to support higher bit precision in high-performance DSP applications.

3.1. Supported Operational Modes in Intel Arria 10 Devices

Table 14. Supported Combinations of Operational Modes and Features for Variable Precision DSP Block in Intel Arria 10 Devices

Variable-Precision DSP Block Resource	Operation Mode	Supported Operation Instance	Pre-Adder Support	Coefficient Support	Input Cascade Support	Chainin Support	Chainout Support
1 variable precision DSP block	Fixed-point independent 18 x 19 multiplication	2	Yes	Yes	Yes (2)	No	No
	Fixed-point independent 27 x 27 multiplication	1	Yes	Yes	Yes (3)	Yes	Yes
	Fixed-point two 18 x 19 multiplier adder mode	1	Yes	Yes	Yes(2)	Yes	Yes
	Fixed-point 18 x 18 multiplier adder summed with 36-bit input	1	No	No	No	Yes	Yes
	Fixed-point 18 x 19 systolic mode	1	Yes	Yes	Yes(2)	Yes	Yes
1 variable precision DSP block	Floating-point multiplication mode	1	No	No	No	No	Yes
	Floating-point adder or subtract mode	1	No	No	No	No	Yes

continued...

(2) Each of the two inputs to a pre-adder has a maximum width of 18-bit. When the input cascade is used to feed one of the pre-adder inputs, the maximum width for the input cascade is 18-bit.

(3) When you enable the pre-adder feature, the input cascade support is not available.

Variable-Precision DSP Block Resource	Operation Mode	Supported Operation Instance	Pre-Adder Support	Coefficient Support	Input Cascade Support	Chainin Support	Chainout Support
	Floating-point multiplier adder or subtract mode	1	No	No	No	Yes	Yes
	Floating-point multiplier accumulate mode	1	No	No	No	No	Yes
	Floating-point vector one mode	1	No	No	No	Yes	Yes
	Floating-point vector two mode	1	No	No	No	Yes	Yes
2 Variable precision DSP blocks	Complex 18x19 multiplication	1	No	No	Yes	No	No

Table 15. Supported Combinations of Operational Modes and Dynamic Control Features for Variable Precision DSP Blocks in Intel Arria 10 Devices

Variable-Precision DSP Block Resource	Operation Mode	Dynamic ACCUMULATE	Dynamic LOADCONST	Dynamic SUB	Dynamic NEGATE
1 variable precision DSP block	Fixed-point independent 18 x 19 multiplication	No	No	No	No
	Fixed-point independent 27 x 27 multiplication	Yes	Yes	No	Yes
	Fixed-point two 18 x 19 multiplier adder mode	Yes	Yes	Yes	Yes
	Fixed-point 18 x 18 multiplier adder summed with 36-bit input	Yes	Yes	Yes	Yes
	Fixed-point 18 x 19 systolic mode	Yes	Yes	Yes	Yes
	Floating-point multiplication mode	No	No	No	No
	Floating-point adder or subtract mode	No	No	No	No
	Floating-point multiplier adder or subtract mode	No	No	No	No
	Floating-point multiplier accumulate mode	Yes	No	No	No

continued...

Variable-Precision DSP Block Resource	Operation Mode	Dynamic ACCUMULATE	Dynamic LOADCONST	Dynamic SUB	Dynamic NEGATE
	Floating-point vector one mode	No	No	No	No
	Floating-point vector two mode	No	No	No	No
2 variable precision DSP blocks	Complex 18 x 19 multiplication	No	No	No	No

3.1.1. Features

The Intel Arria 10 variable precision DSP blocks support fixed-point arithmetic and floating-point arithmetic.

Features for fixed-point arithmetic:

- High-performance, power-optimized, and fully registered multiplication operations
- 18-bit and 27-bit word lengths
- Two 18 x 19 multipliers or one 27 x 27 multiplier per DSP block
- Built-in addition, subtraction, and 64-bit double accumulation register to combine multiplication results
- Cascading 19-bit or 27-bit when pre-adder is disabled and cascading 18-bit when pre-adder is used 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 19-bit and 27-bit modes for symmetric filters
- Internal coefficient register bank in both 18-bit and 27-bit modes for filter implementation
- 18-bit and 27-bit systolic finite impulse response (FIR) filters with distributed output adder
- Biased rounding support

Features for floating-point arithmetic:

- A completely hardened architecture that supports multiplication, addition, subtraction, multiply-add, and multiply-subtract
- Multiplication with accumulation capability and a dynamic accumulator reset control
- Multiplication with cascade summation capability
- Multiplication with cascade subtraction capability
- Complex multiplication
- Direct vector dot product
- Systolic FIR filter

Related Information

- Intel Arria 10 Device Handbook: Known Issues
  Lists the planned updates to the *Arria 10 Device Handbook* chapters.

- Intel Arria 10 Device Overview - Variable-Precision DSP Block
  Provides more information about the number of multipliers in each Intel Arria 10 device.

3.2. Resources

Table 16. Resources for Fixed-Point Arithmetic in Intel Arria 10 Devices

The table lists the variable-precision DSP resources by bit precision for each Intel Arria 10 device.

Variant	Product Line	Variable-precision DSP Block	Independent Input and Output Multiplications Operator		18 x 19 Multiplier Adder Sum Mode	18 x 18 Multiplier Adder Summed with 36 bit Input
			18 x 19 Multiplier	27 x 27 Multiplier		
AIntel Arria 10 GX	GX 160	156	312	156	156	156
	GX 220	192	384	192	192	192
	GX 270	830	1,660	830	830	830
	GX 320	984	1,968	984	984	984
	GX 480	1,368	2,736	1,368	1,368	1,368
	GX 570	1,523	3,046	1,523	1,523	1,523
	GX 660	1,687	3,374	1,687	1,687	1,687
	GX 900	1,518	3,036	1,518	1,518	1,518
	GX 1150	1,518	3,036	1,518	1,518	1,518
Intel Arria 10 GT	GT 900	1,518	3,036	1,518	1,518	1,518
	GT 1150	1,518	3,036	1,518	1,518	1,518
Intel Arria 10 SX	SX 160	156	312	156	156	156
	SX 220	192	384	192	192	192
	SX 270	830	1,660	830	830	830
	SX 320	984	1,968	984	984	984
	SX 480	1,368	2,736	1,368	1,368	1,368
	SX 570	1,523	3,046	1,523	1,523	1,523
	SX 660	1,687	3,374	1,687	1,687	1,687

Table 17. Resources for Floating-Point Arithmetic in Intel Arria 10 Devices

The table lists the variable-precision DSP resources by bit precision for each Intel Arria 10 device.

Variant	Product Line	Variable-precision DSP Block	Single Precision Floating-Point Multiplication Mode	Single-Precision Floating-Point Adder Mode	Single-Precision Floating-Point Multiply Accumulate Mode	Peak Giga Floating-Point Operations per Second (GFLOPs)
Intel Arria 10 GX	GX 160	156	156	156	156	140
	GX 220	192	192	192	192	173
	GX 270	830	830	830	830	747

continued...

Variant	Product Line	Variable-precision DSP Block	Single Precision Floating-Point Multiplication Mode	Single-Precision Floating-Point Adder Mode	Single-Precision Floating-Point Multiply Accumulate Mode	Peak Giga Floating-Point Operations per Second (GFLOPs)
	GX 320	984	984	984	984	886
	GX 480	1,369	1,368	1,368	1,368	1,231
	GX 570	1,523	1,523	1,523	1,523	1,371
	GX 660	1,687	1,687	1,687	1,687	1,518
	GX 900	1,518	1,518	1,518	1,518	1,366
	GX 1150	1,518	1,518	1,518	1,518	1,366
Intel Arria 10 GT	GT 900	1,518	1,518	1,518	1,518	1,366
	GT 1150	1,518	1,518	1,518	1,518	1,366
Intel Arria 10 SX	SX 160	156	156	156	156	140
	SX 220	192	192	192	192	173
	SX 270	830	830	830	830	747
	SX 320	984	984	984	984	886
	SX 480	1,369	1,368	1,368	1,368	1,231
	SX 570	1,523	1,523	1,523	1,523	1,371
	SX 660	1,687	1,687	1,687	1,687	1,518

3.3. Design Considerations

You should consider the following elements in your design:

Table 18. Design Considerations

DSP Implementation	Fixed-Point Arithmetic	Floating-Point Arithmetic
Design elements	• Operational modes • Internal coefficient and pre-adder • Accumulator • Chainout adder	• Operational modes • Chainout adder

The Intel Quartus Prime software provides the following design templates for you to implement DSP blocks in Intel Arria 10 devices.

Table 19. DSP Design Templates Available in Intel Arria 10 Devices

Operational Mode	Available Design Templates
18 x 18 Independent Multiplier Mode	Single Multiplier with Preadder and Coefficient
27 x 27 Independent Multiplier Mode	• M27x27 with Dynamic Negate • M27x27 with Preadder and Coefficient • M27x27 with Input Cascade, Output Chaining, Accumulator, Double Accumulator, and Preload Constant

continued...

Operational Mode	Available Design Templates
Multiplier Adder Sum Mode	• M18x19_sumof2 with Dynamic Sub and Dynamic Negate • M18x19_sumof2 with Preadder and Coefficient • M18x19_sumof2 with Input Cascade, Output Chaining, Accumulator, Double Accumulator, and Preload Constant
18 x 19 Multiplication Summed with 36-Bit Input Mode	• M18x19_plus36 with Dynamic Sub and Dynamic Negate • M18x19_plus36 with Input Cascade, Output Chaining, Accumulator, Double Accumulator, and Preload Constant
18-bit Systolic FIR Mode	• M18x19_systolic with Preadder and Coefficient • M18x19_systolic with Input Cascade, Output Chaining, Accumulator, Double Accumulator, and Preload Constant

You can get the design templates using the following steps:

1. In Intel Quartus Prime software, open a new Verilog HDL or VHDL file.
2. From Edit tab, click **Insert Template**.
3. From the Insert Template window prompt, you may select **Verilog HDL** or **VHDL** depending on your preferred design language.
4. Click **Full Designs** to expand the options.
5. From the options, click **Arithmetic ➤ DSP Features ➤ ➤ DSP Features for 20-nm Device**.
6. Choose the design template that match your system requirement and click **Insert to append the design template to a new .v or .vhd file**.

3.3.1. 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.

Variable-precision DSP block can also be implemented using DSP Builder for Intel FPGAs and OpenCL™.

Table 20. Operational Modes

Fixed-Point Arithmetic	Floating-Point Arithmetic
Intel provides two methods for implementing various modes of the Intel Arria 10 variable precision DSP block in a design—using the Intel Quartus Prime DSP IP core and HDL inferring. The following Intel Quartus Prime IP cores are supported for the Intel Arria 10 variable precision DSP blocks in the fixed-point arithmetic implementation: • Intel FPGA Multiply Adder • ALTMULT_COMPLEX • Intel Arria 10 Native Fixed Point DSP IP core	Intel provides one method for implementing various modes of the Intel Arria 10 variable precision DSP block in a design—using the Intel Quartus Prime DSP IP core. The following Intel Quartus Prime IP cores are supported for the Intel Arria 10 variable precision DSP blocks in the floating-point arithmetic implementation: • ALTERA_FP_FUNCTIONS • Intel Arria 10 Native Floating Point DSP IP core

Related Information

- Introduction to Intel FPGA IP Cores
- Intel FPGA Integer Arithmetic Megafunctions User Guide
- Floating-Point Megafunctions User Guide - ALTERA_FP_FUNCTIONS IP Core

- Quartus Prime Software Help
- Intel Arria 10 Native Fixed Point DSP IP User Guide

3.3.2. Internal Coefficient and Pre-Adder for Fixed-Point Arithmetic

When you enable input register for the pre-adder feature, these input registers must have the same clock setting.

The input cascade support is only available for 18-bit mode when you enable the pre-adder feature.

In both 18-bit and 27-bit modes, you can use the coefficient feature and pre-adder feature independently.

When internal coefficient feature is enabled in 18-bit modes, you must enable both top and bottom coefficient.

When pre-adder feature is enabled in 18-bit modes, you must enable both top and bottom pre-adder.

3.3.3. Accumulator for Fixed-Point Arithmetic

The accumulator in the Intel Arria 10 devices supports double accumulation by enabling the 64-bit double accumulation registers located between the output register bank and the accumulator.

3.3.4. Chainout Adder

Table 21. Chainout Adder

Fixed-Point Arithmetic	Floating-Point Arithmetic
You can use the output chaining path to add results from another DSP block.	You can use the output chaining path to add results from another DSP block. Support for certain operation modes: • Multiply-add or multiply-subtract mode • Vector one mode • Vector two mode

3.4. Block Architecture

The Intel Arria 10 variable precision DSP block consists of the following elements:

Table 22. Block Architecture

DSP Implementation	Fixed-Point Arithmetic	Floating-Point Arithmetic
Block architecture	• Input register bank • Pipeline register • Pre-adder • Internal coefficient • Multipliers • Adder • Accumulator and chainout adder	• Input register bank • Pipeline register • Multipliers • Adder • Accumulator and chainout adder • Output register bank

DSP Implementation	Fixed-Point Arithmetic	Floating-Point Arithmetic
	• Systolic registers • Double accumulation register • Output register bank	

If the variable precision DSP block is not configured in fixed-point arithmetic systolic FIR mode, both systolic registers are bypassed.

Figure 25. Variable Precision DSP Block Architecture in 18 x 19 Mode for Fixed-Point Arithmetic in Intel Arria 10 Devices

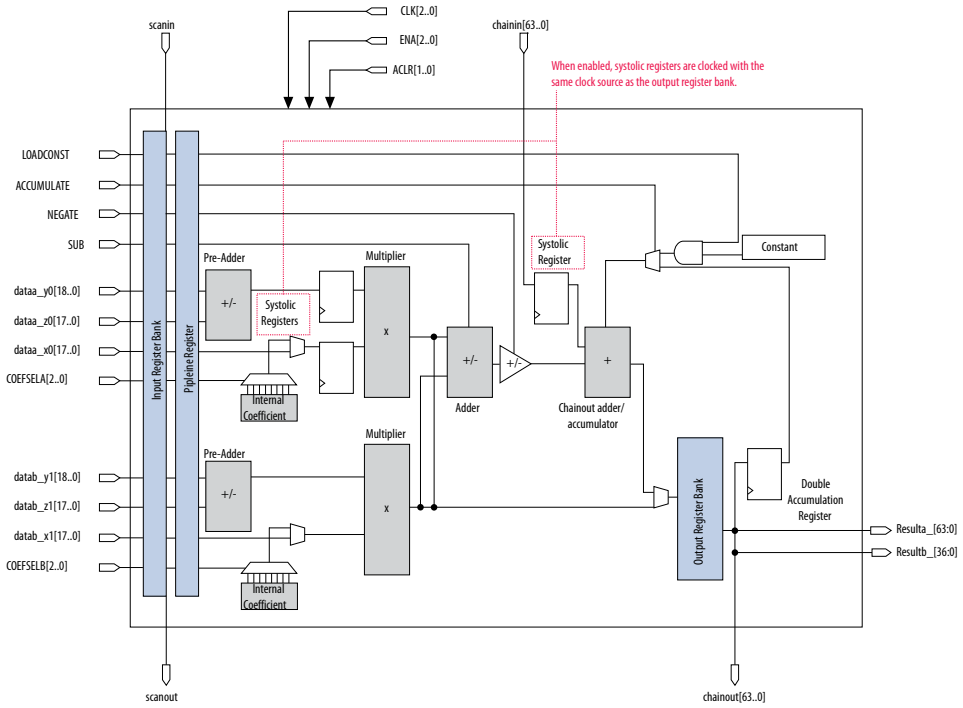


Figure 26. Variable Precision DSP Block Architecture in 27 x 27 Mode for Fixed-Point Arithmetic in Intel Arria 10 Devices

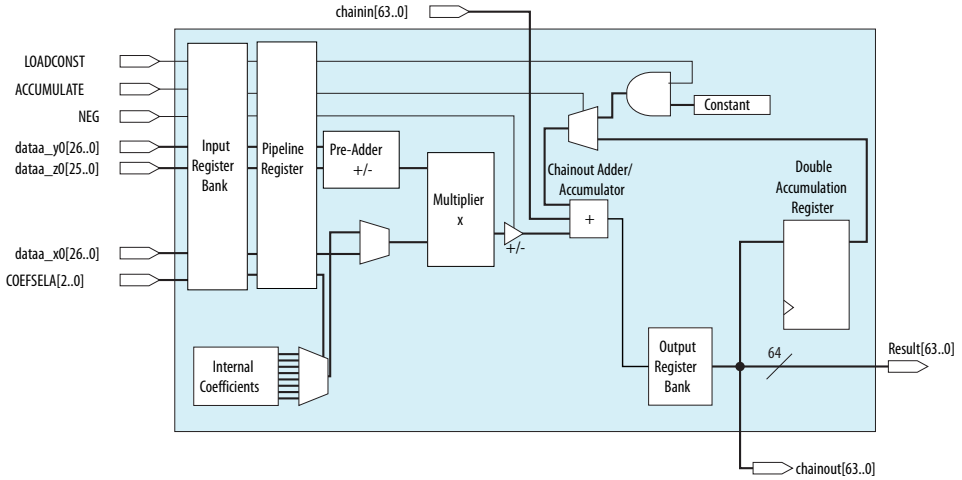
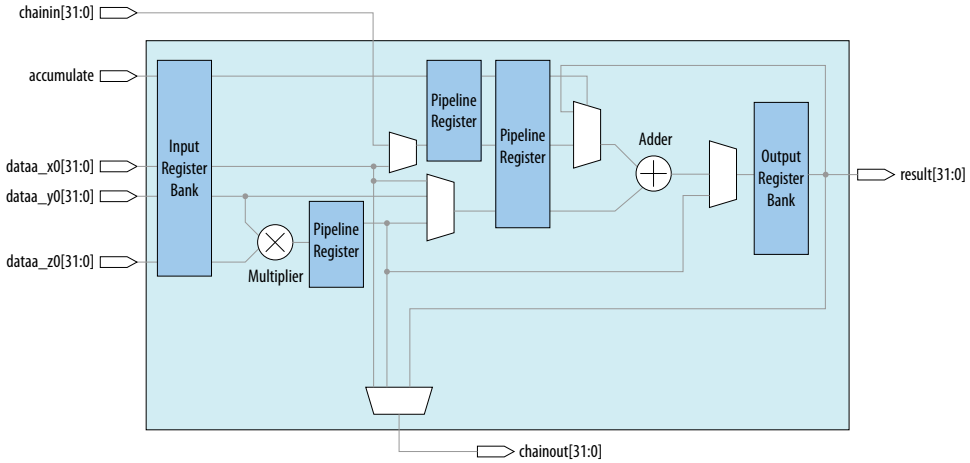


Figure 27. Variable Precision DSP Block Architecture for Floating-Point Arithmetic in Intel Arria 10 Devices



3.4.1. Input Register Bank

Table 23. Input Register Bank

Fixed-Point Arithmetic	Floating-Point Arithmetic
• Data • Dynamic control signals • Two sets of delay registers	• Data • Dynamic ACCUMULATE control signal

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 fixed-point arithmetic 18 x 19 mode, you can use the delay registers to balance the latency requirements when you use both the input cascade and chainout features.

The tap-delay line feature allows you to drive the top leg of the multiplier input, dataa_y0 and datab_y1 in fixed-point arithmetic 18 x 19 mode and dataa_y0 only in fixed-point arithmetic 27 x 27 mode, from the general routing or cascade chain.

3.4.1.1. Two Sets of Delay Registers for Fixed-Point Arithmetic

The two delay registers along with the input cascade chain that can be used in fixed-point arithmetic 18 x 19 mode are the top delay registers and bottom delay registers. Delay registers are not supported in 18 × 19 multiplication summed with 36-bit input mode and 27 × 27 mode.

Figure 28. Input Register of a Variable Precision DSP Block in Fixed-Point Arithmetic 18 x 19 Mode for Intel Arria 10 Devices

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

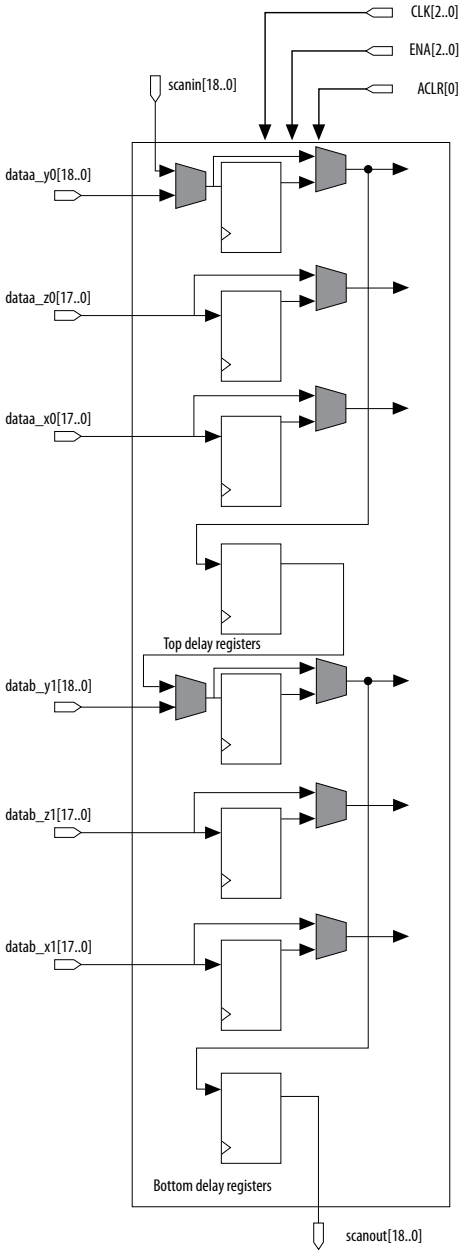
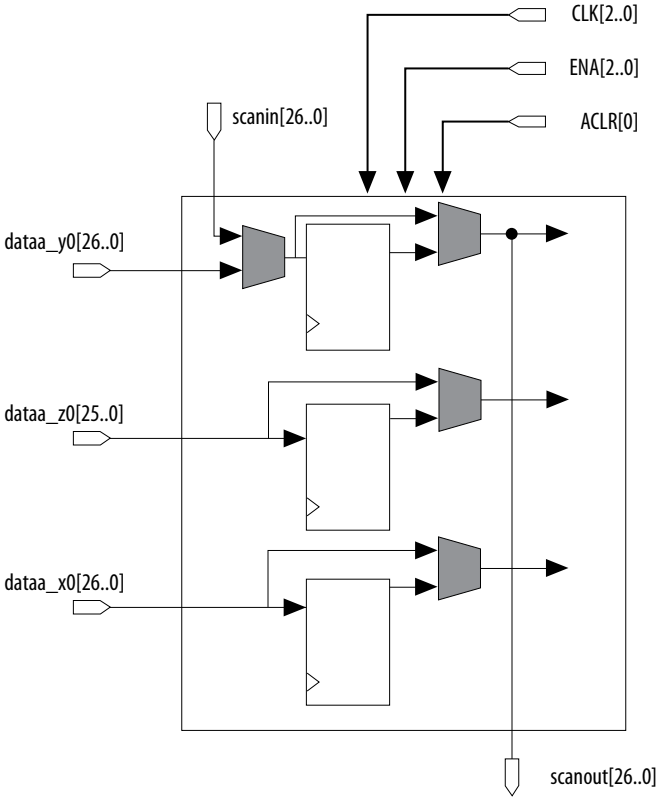


Figure 29. Input Register of a Variable Precision DSP Block in Fixed-Point Arithmetic 27 x 27 Mode for Intel Arria 10 Devices

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



3.4.2. Pipeline Register

Pipeline register is used to get the maximum Fmax performance. Pipeline register can be bypassed if high Fmax is not needed.

The following variable precision DSP block signals control the pipeline registers within the variable precision DSP block:

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

Floating-point arithmetic has 2 latency layers of pipeline registers where you can perform one of the following:

- Bypass all latency layers of pipeline registers
- Use either one latency layers of pipeline registers
- Use both latency layers of pipeline registers

3.4.3. Pre-Adder for Fixed-Point Arithmetic

Each variable precision DSP block has two 19-bit pre-adders. You can configure these pre-adders in the following configurations:

- Two independent 19-bit pre-adders
- One 27-bit pre-adder

The pre-adder supports both addition and subtraction in the following input configurations:

- 18-bit (signed or unsigned) addition or subtraction for 18 x 19 mode
- 26-bit addition or subtraction for 27 x 27 mode

When both pre-adders within the same DSP block are used, they must share the same operation type (either addition or subtraction).

3.4.4. Internal Coefficient for Fixed-Point Arithmetic

The Intel Arria 10 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, C0EFSELA/C0EFSELB are used to control the selection of the coefficient multiplexer.

3.4.5. Multipliers

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

There are two multipliers per variable precision DSP block. You can configure these two multipliers in several operational modes:

Table 24. Operational Modes

Fixed-Point Arithmetic	Floating-Point Arithmetic
• One 27 x 27 multiplier • Two 18 (signed or unsigned) x 19 (signed) multipliers	One floating-point arithmetic single precision multiplier

Related Information

Operational Mode Descriptions on page 54
Provides more information about the operational modes of the multipliers.

3.4.6. Adder

Depending on the operational mode, you can use the adder as follows:

- One 55-bit or 38-bit adder
- One floating-point arithmetic single precision adder

DSP Implementation	Addition Using Dynamic SUB Port	Subtraction Using Dynamic SUB Port
Fixed-Point Arithmetic	Yes	Yes
Floating-Point Arithmetic	No	No

3.4.7. Accumulator and Chainout Adder for Fixed-Point Arithmetic

The Intel Arria 10 variable precision DSP block supports a 64-bit accumulator and a 64-bit adder for fixed-point arithmetic.

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

- NEGATE
- LOADCONST
- ACCUMULATE

The accumulator supports double accumulation by enabling the 64-bit double accumulation registers located between the output register bank and the accumulator.

The accumulator and chainout adder features are not supported in two fixed-point arithmetic independent 18 x 19 modes.

Table 25. Accumulator Functions and Dynamic Control Signals

This table lists the dynamic signals settings and description for each function. In this table, X denotes a "don't care" value.

Function	Description	NEGATE	LOADCONST	ACCUMULATE
Zeroing	Disables the accumulator.	0	0	0
Preload	The result is always added to the preload value. 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	X	1
Decimation + Accumulate	This function takes the current result, converts it into two's complement, and adds it to the previous result.	1	X	1
Decimation + Chainout Adder	This function takes the current result, converts it into two's complement, and adds it to the output of previous DSP block.	1	0	0

3.4.8. Systolic Registers for Fixed-Point Arithmetic

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

The first set of systolic registers consists of 18-bit and 19-bit registers that are used to register the 18-bit and 19-bit inputs of the upper multiplier, respectively.

The second set of systolic registers are used to delay the chainin input from the previous variable precision DSP block.

You must clock all the systolic registers with the same clock source as the output register. Output registers must be turned on.

3.4.9. Double Accumulation Register for Fixed-Point Arithmetic

The double accumulation register is an extra register in the feedback path of the accumulator. Enabling the double accumulation register causes an extra clock cycle delay in the feedback path of the accumulator.

This register has the same CLK, ENA, and ACLR settings as the output register bank.

By enabling this register, you can have two accumulator channels using the same number of variable precision DSP block. This is useful when processing interleaved complex data (I, Q).

3.4.10. Output Register Bank

The positive edge of the clock signal triggers the 74-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]`

3.5. Operational Mode Descriptions

This section describes how you can configure an Intel Arria 10 variable precision DSP block to efficiently support the fixed-point arithmetic and floating-point arithmetic operational modes.

Table 26. Operational Modes

Fixed-Point Arithmetic	Floating-Point Arithmetic
• Independent multiplier mode • Multiplier adder sum mode • Independent complex multiplier • 18 x 18 multiplication summed with 36-Bit input mode • Systolic FIR mode	• Multiplication mode • Adder or subtract mode • Multiply-add or multiply-subtract mode • Multiply accumulate mode • Vector one mode • Vector two mode • Direct vector dot product • Complex multiplication

3.5.1. Operational Modes for Fixed-Point Arithmetic

3.5.1.1. Independent Multiplier Mode

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

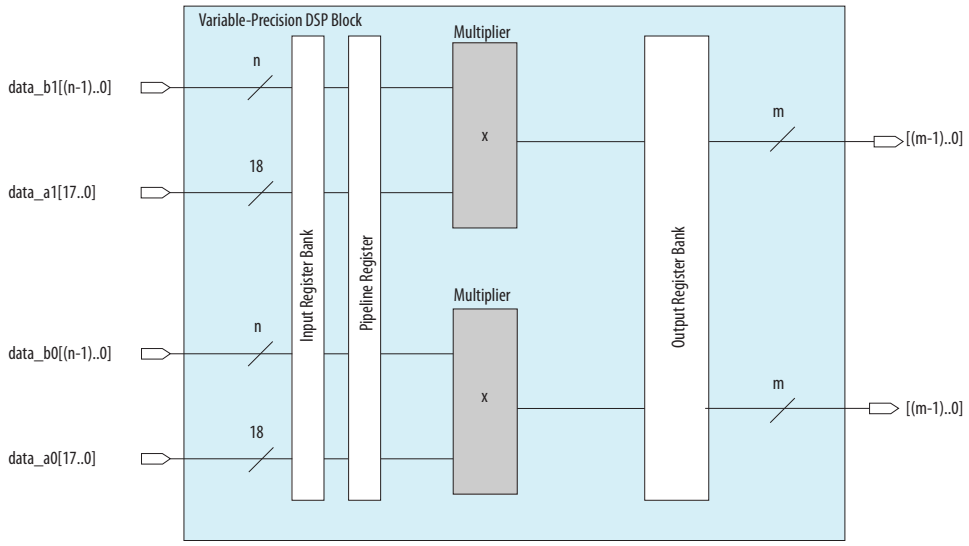
Configuration	Multipliers per Block
18 (signed) x 19 (signed)	2
18 (unsigned) x 18 (unsigned)	2
27 (signed or unsigned) x 27 (signed or unsigned)	1

3.5.1.1.1. 18 x 18 or 18 x 19 Independent Multiplier

Figure 30. Two 18 x 18 or 18 x 19 Independent Multiplier per Variable Precision DSP Block for Intel Arria 10 Devices

In this figure, the variables are defined as follows:

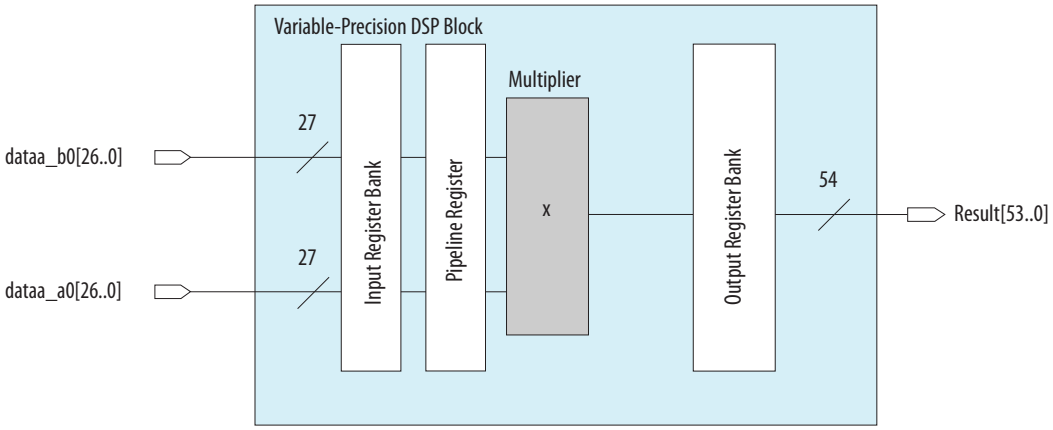
- n = 19 and m = 37 for 18 x 19 operand
- n = 18 and m = 36 for 18 x 18 operand



3.5.1.1.2. 27 x 27 Independent Multiplier

Figure 31. One 27 x 27 Independent Multiplier Mode per Variable Precision DSP Block for Intel Arria 10 Devices

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



3.5.1.2. Independent Complex Multiplier

The Intel Arria 10 devices support the 18 x 19 complex multiplier mode using two fixed-point arithmetic multiplier adder sum mode.

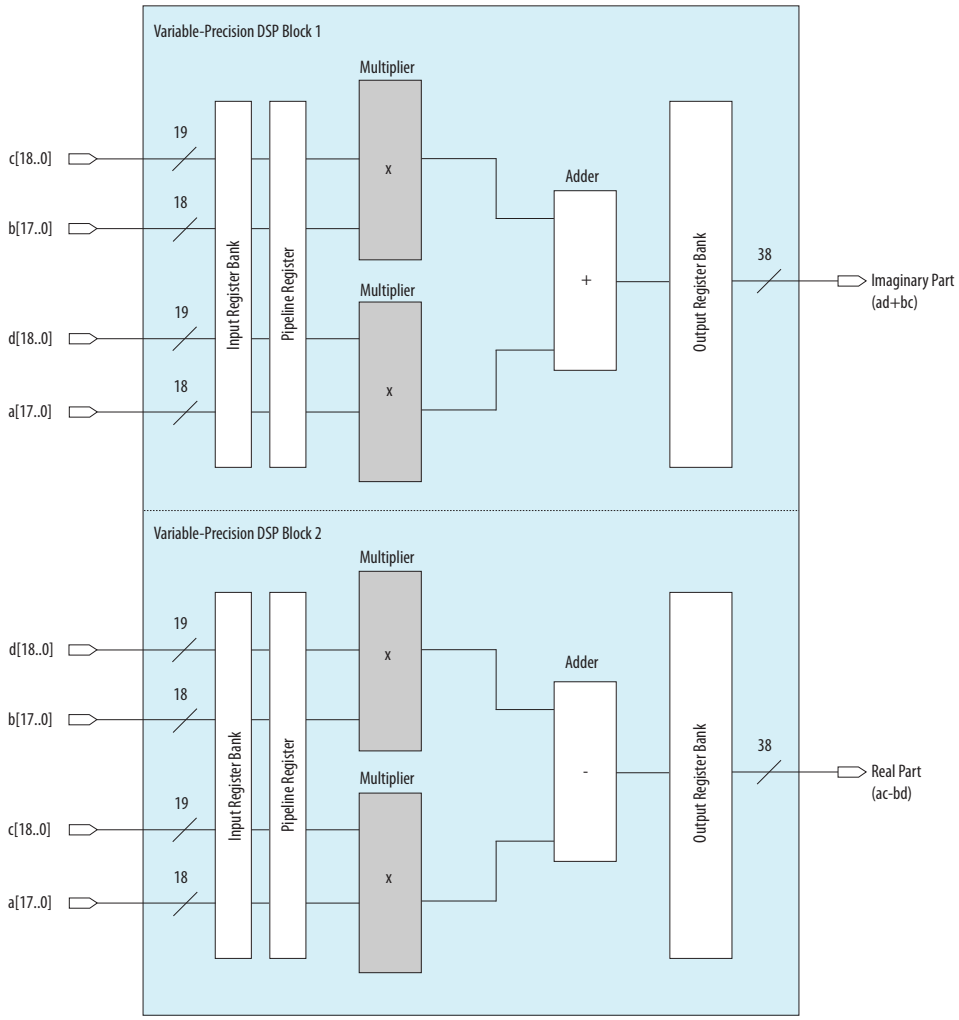
Figure 32. Sample of Complex Multiplication Equation

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

The imaginary part [(a × d) + (b × c)] is implemented in the first variable-precision DSP block, while the real part [(a × c) - (b × d)] is implemented in the second variable-precision DSP block.

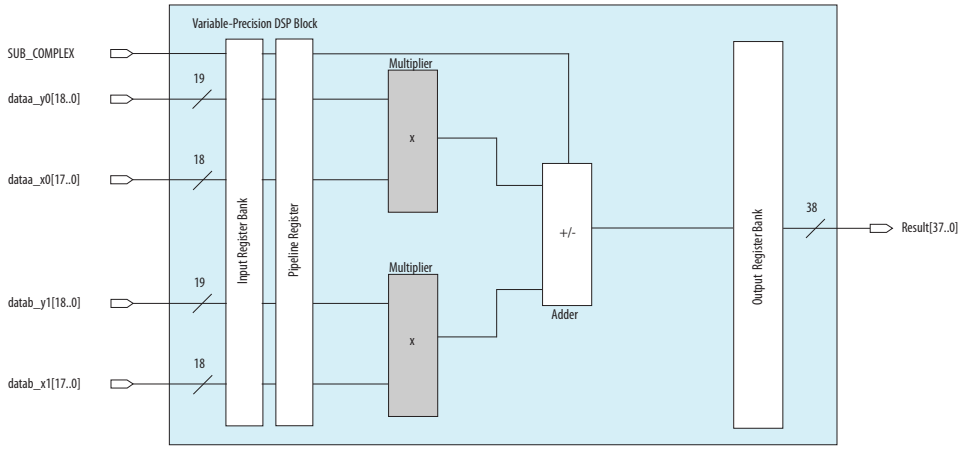
3.5.1.2.1. 18 x 19 Complex Multiplier

Figure 33. One 18 x 19 Complex Multiplier with Two Variable Precision DSP Blocks for Intel Arria 10 Devices



3.5.1.3. Multiplier Adder Sum Mode

Figure 34. One Sum of Two 18 x 19 Multipliers with One Variable Precision DSP Block for Intel Arria 10 Devices

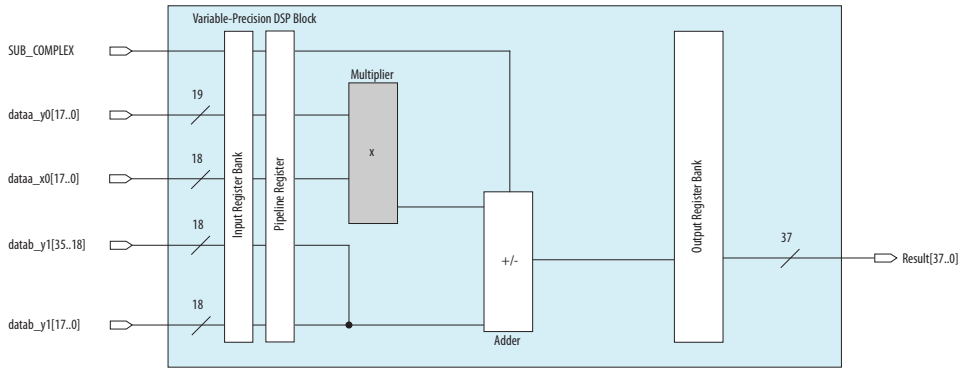


3.5.1.4. 18 x 19 Multiplication Summed with 36-Bit Input Mode

Intel Arria 10 variable precision DSP blocks support one 18 x 19 multiplication summed to a 36-bit input.

Use the upper multiplier to provide the input for an 18 x 19 multiplication, while the bottom multiplier is bypassed. The `datab_y1[17..0]` and `datab_y1[35..18]` signals are concatenated to produce a 36-bit input.

Figure 35. One 18 x 19 Multiplication Summed with 36-Bit Input Mode for Intel Arria 10 Devices



3.5.1.5. Systolic FIR Mode

The basic structure of a FIR filter consists of a series of multiplications followed by an addition.

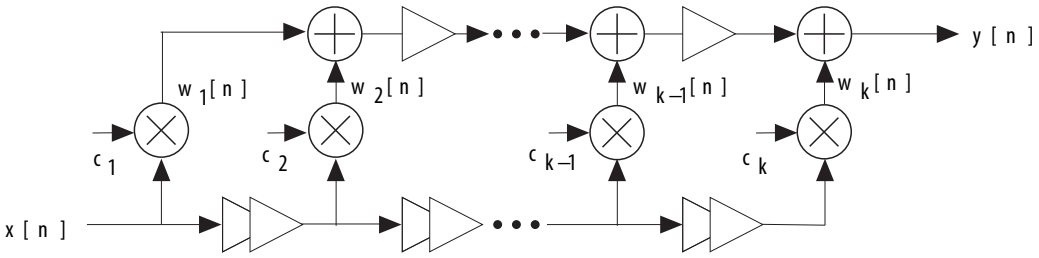
Figure 36. Basic FIR Filter Equation

$$y[n] = (\sum_{i=1}^{k} w_i[n-k+i] + w_1[n-k+2])$$

$$\text{Where } i \text{ start from 1,} \qquad w_i[n] = c_i x[n-2i+2]$$

Depending on the number of taps and the input sizes, the delay through chaining a high number of adders can become quite large. To overcome the delay performance issue, the systolic form is used with additional delay elements placed per tap to increase the performance at the cost of increased latency.

Figure 37. Systolic FIR Filter Equivalent Circuit



Intel Arria 10 variable precision DSP blocks support the following systolic FIR structures:

- 18-bit
- 27-bit

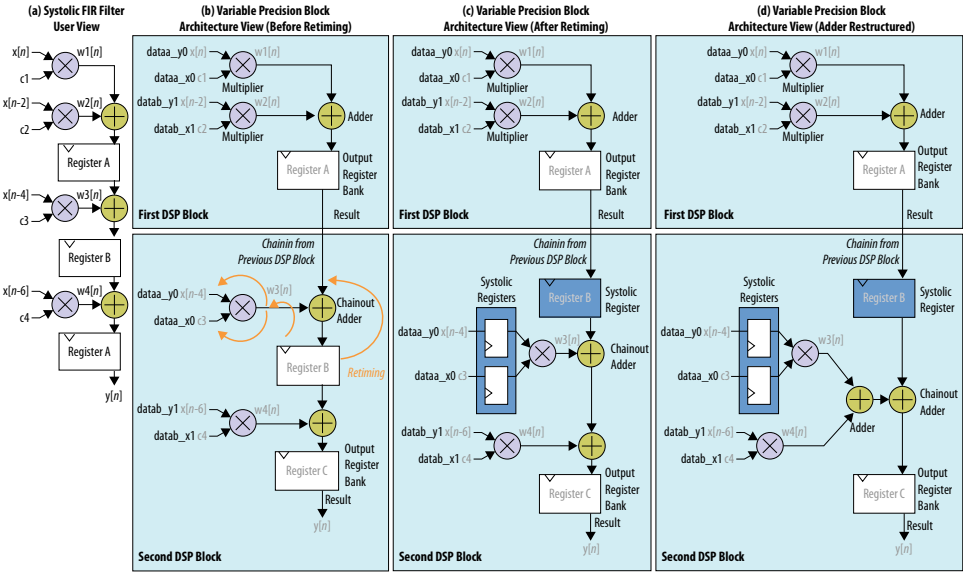
In systolic FIR mode, the input of the multiplier can come from four different sets of sources:

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

3.5.1.5.1. Mapping Systolic Mode User View to Variable Precision Block Architecture View

The following figure shows that the user view of the systolic FIR filter (a) can be implemented using the Intel Arria 10 variable precision DSP blocks (d) by retiming the register and restructuring the adder. Register B can be retimed into systolic registers at the chainin, dataa_y0 and dataa_x0 input paths as shown in (b). The end result of the register retiming is shown in (c). The summation of two multiplier results by restructuring the inputs and location of the adder are added to the chainin input by the chainout adder as shown in (d).

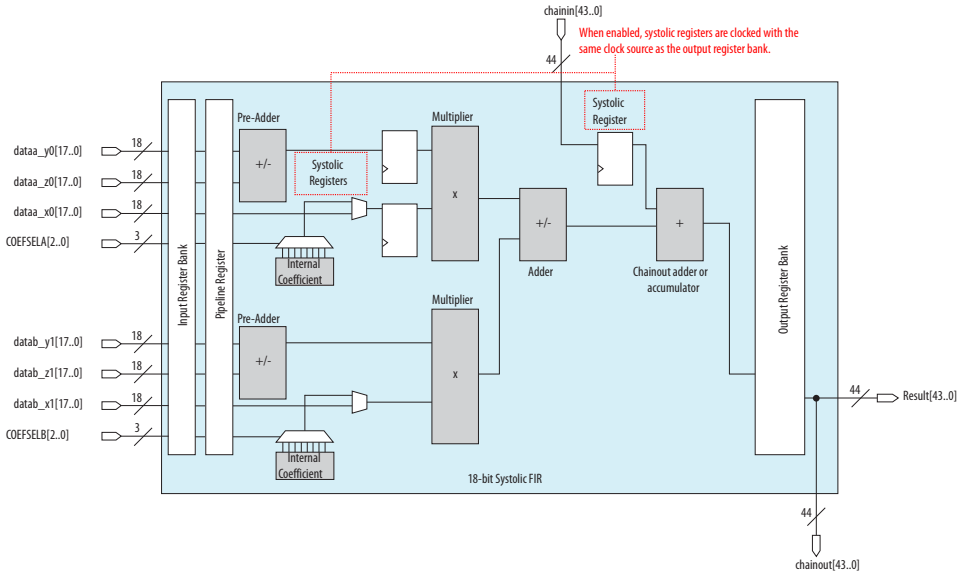
Figure 38. Mapping Systolic Mode User View to Variable Precision Block Architecture View



3.5.1.5.2. 18-Bit Systolic FIR Mode

In 18-bit systolic FIR mode, the adders are configured as dual 44-bit adders, thereby giving 7 bits of overhead when using an 18 x 19 operation mode, resulting 37-bit result. This allows a total sixteen 18 x 19 multipliers or eight Intel Arria 10 variable precision DSP blocks to be cascaded as systolic FIR structure.

Figure 39. 18-Bit Systolic FIR Mode for Intel Arria 10 Devices

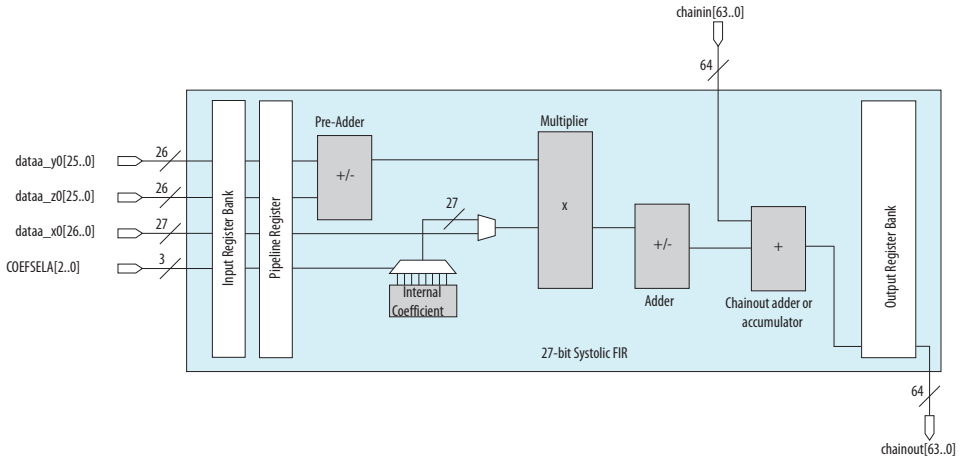


3.5.1.5.3. 27-Bit Systolic FIR Mode

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 eleven 27 x 27 multipliers or eleven Intel Arria 10 variable precision DSP blocks to be cascaded as systolic FIR structure.

The 27-bit systolic FIR mode allows the implementation of one stage systolic filter per DSP block. Systolic registers are not required in this mode.

Figure 40. 27-Bit Systolic FIR Mode for Intel Arria 10 Devices



3.5.2. Operational Modes for Floating-Point Arithmetic

3.5.2.1. Single Floating-Point Arithmetic Functions

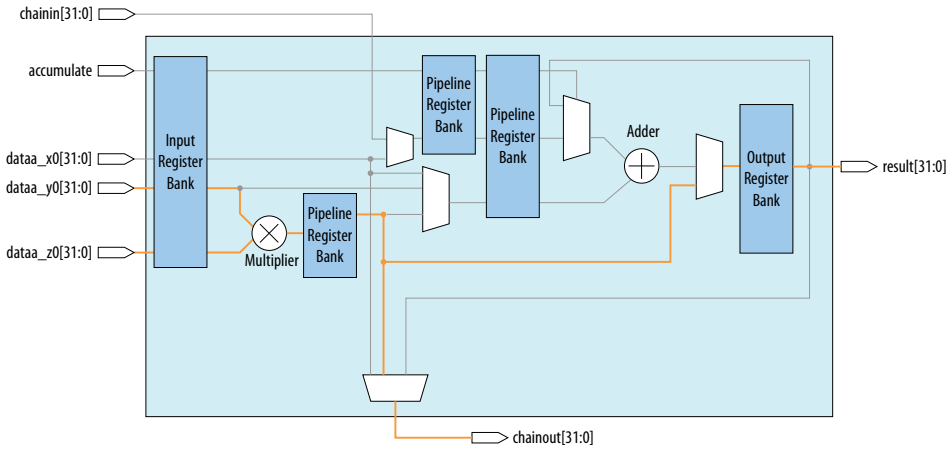
One floating-point arithmetic DSP can perform the following:

- Multiplication mode
- Adder or subtract mode
- Multiply accumulate mode

3.5.2.1.1. Multiplication Mode

This mode allows you to apply basic floating-point multiplication (y*z).

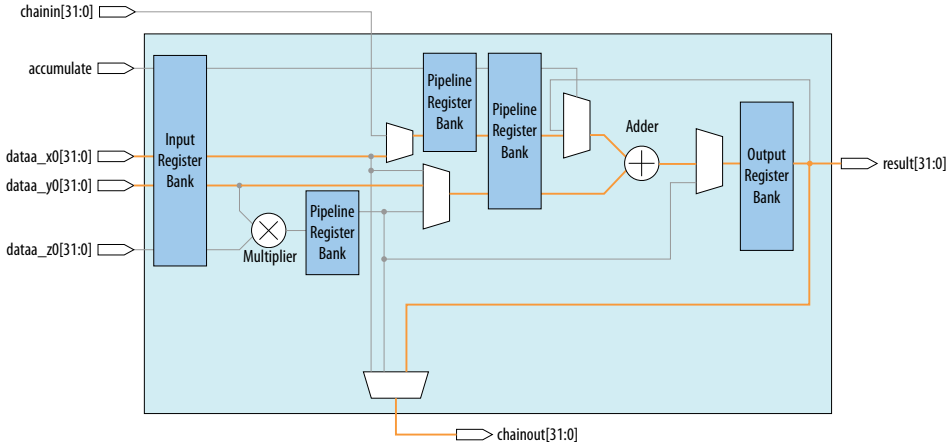
Figure 41. Multiplication Mode for Intel Arria 10 Devices



3.5.2.1.2. Adder or Subtract Mode

This mode allows you to apply basic floating-point addition (x+y) or basic floating-point subtraction (y-x).

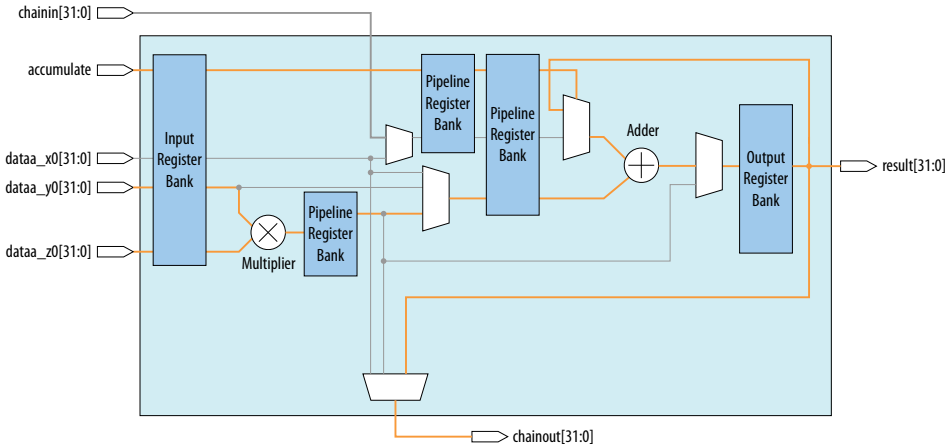
Figure 42. Adder or Subtract Mode for Intel Arria 10 Devices



3.5.2.1.3. Multiply Accumulate Mode

This mode performs floating-point multiplication followed by floating-point addition with the previous multiplication result { ((y*z) + acc) or ((y*z) - acc) }

Figure 43. Multiply Accumulate Mode for Intel Arria 10 Devices



3.5.2.2. Multiple Floating-Point Arithmetic Functions

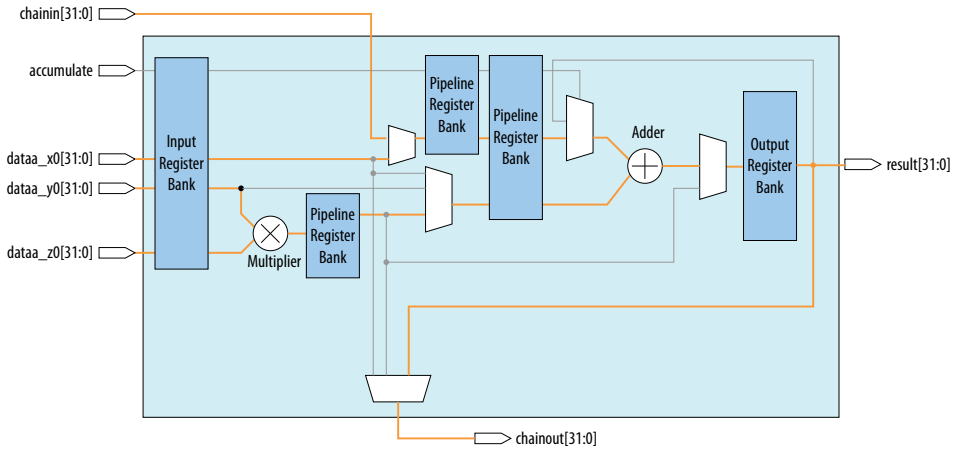
Two or more floating-point arithmetic DSP can perform the following:

- Multiply-add or multiply-subtract mode which uses single floating-point arithmetic DSP if the chainin parameter is turn off
- Vector one mode
- Vector two mode
- Direct vector dot product
- Complex multiplication

3.5.2.2.1. Multiply-Add or Multiply-Subtract Mode

This mode performs floating-point multiplication followed by floating-point addition or floating-point subtraction { ((y*z) + x) or ((y*z) - x) }. The chainin parameter allows you to enable a multiple-chain mode.

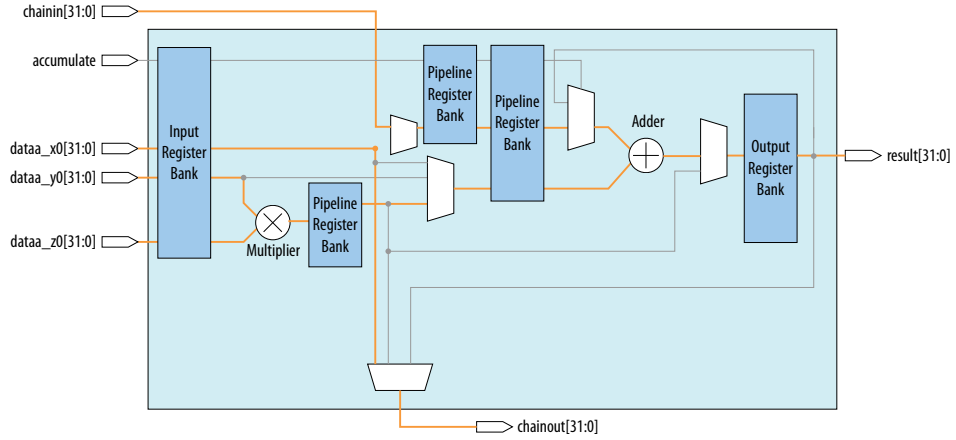
Figure 44. Multiply-Add or Multiply-Subtract Mode for Intel Arria 10 Devices



3.5.2.2.2. Vector One Mode

This mode performs floating-point multiplication followed by floating-point addition with the chainin input from the previous variable DSP Block. Input x is directly fed into chainout. (result = y*z + chainin , where chainout = x)

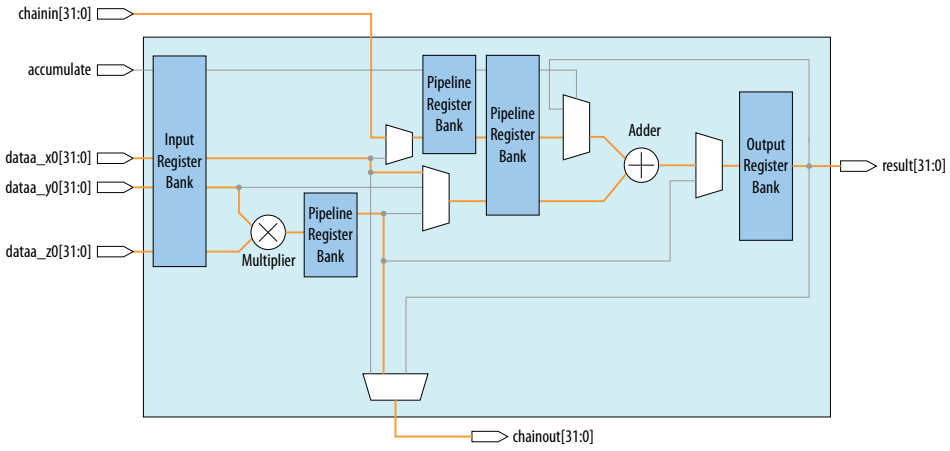
Figure 45. Vector One Mode for Intel Arria 10 Devices



3.5.2.2.3. Vector Two Mode

This mode performs floating-point multiplication where the multiplication result is directly fed to chainout. The chainin input from the previous variable DSP Block is then added to input x as the output result. (result = x + chainin, where chainout = y*z)

Figure 46. Vector Two Mode for Intel Arria 10 Devices

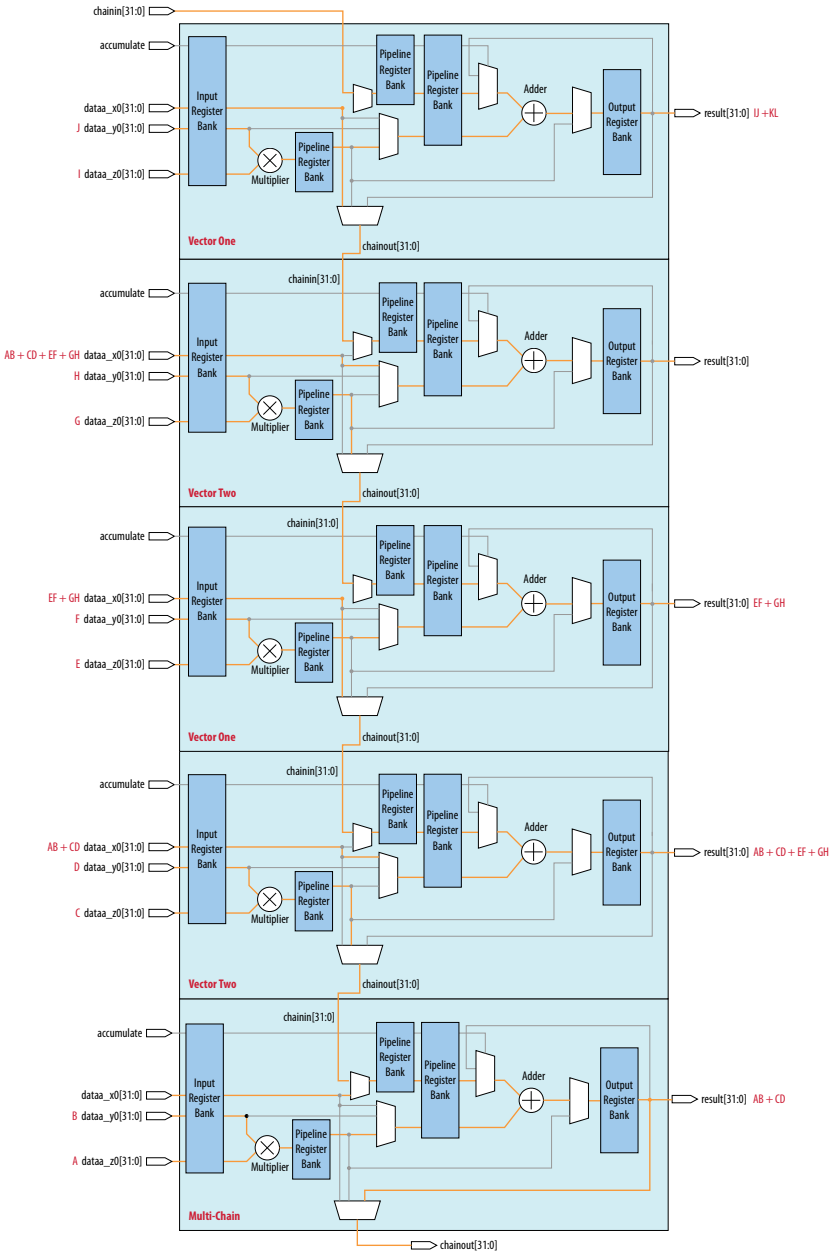


3.5.2.2.4. Direct Vector Dot Product

In the following figure, the direct vector dot product is implemented by several DSP blocks by setting the following DSP modes:

- Multiply-add and subtract mode with chainin parameter turned on
- Vector one
- Vector two

Figure 47. Direct Vector Dot Product



3.5.2.2.5. Complex Multiplication

The Intel Arria 10 devices support the floating-point arithmetic single precision complex multiplier using four Intel Arria 10 variable-precision DSP blocks.

Figure 48. Sample of Complex Multiplication Equation

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

The imaginary part [(a × d) + (b × c)] is implemented in the first two variable-precision DSP blocks, while the real part [(a × c) - (b × d)] is implemented in the second variable-precision DSP block.

Figure 49. Complex Multiplication with Result Real

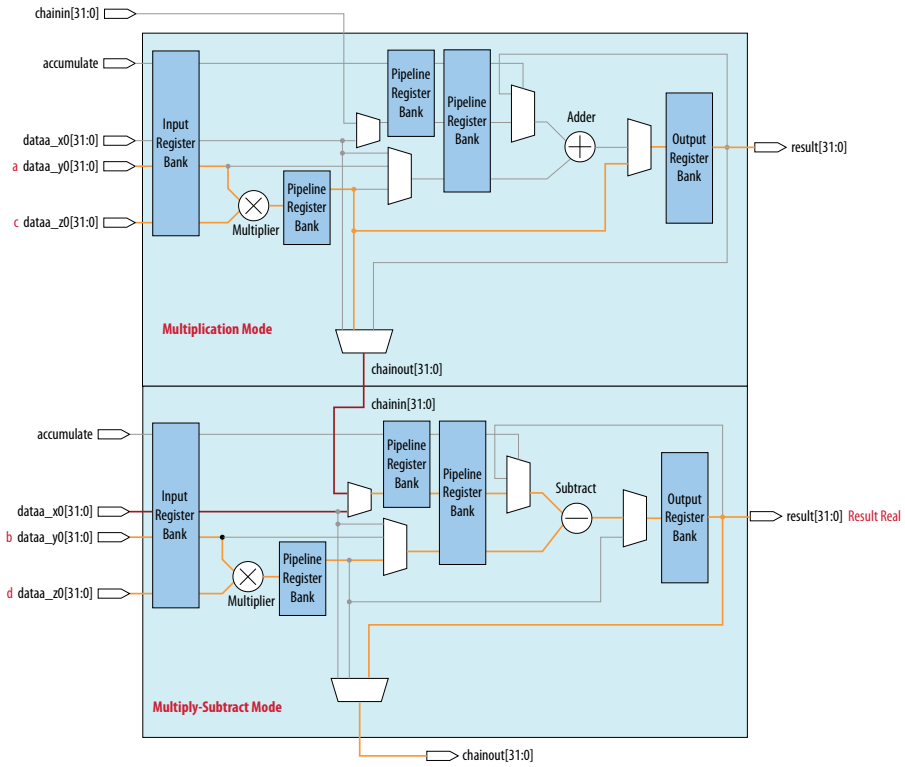
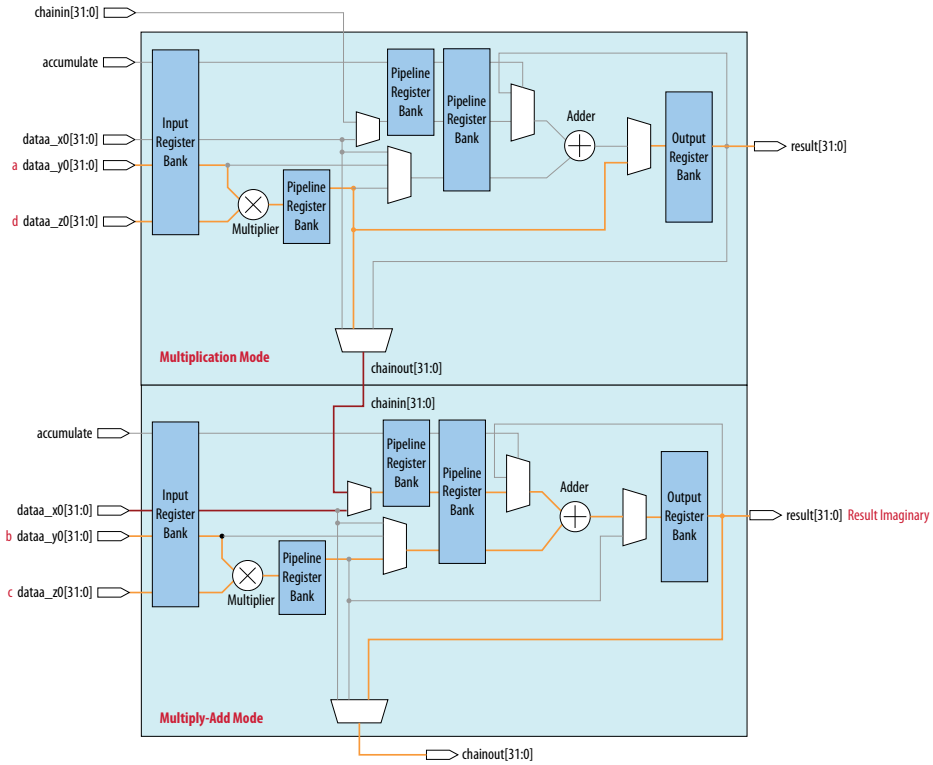


Figure 50. Complex Multiplication with Imaginary Result



3.6. Variable Precision DSP Blocks in Intel Arria 10 Devices Revision History

Date	Version	Changes
March 2017	2017.03.15	• Rebranded as Intel. • Changed subtraction from x-y to y-x.
December 2015	2015.11.14	• Corrected the number of DSP blocks for Arria 10 GX 660 from 1688 to 1687 in the table listing floating-point arithmetic resources.
November 2015	2015.11.02	• Update resource count for Arria 10 GX 320, GX 480, GX 660, SX 320, SX 480, a SX 660 devices in **Number of Multipliers in Intel Arria 10 Devices** table. • Updated the Input Register Bank table to specify input register bank for dynamic control signal in floating-point arithmetic is only applicable for Dynamic ACCUMULATE control signal. • Clarified that 18 x19 systolic FIR mode, there are 7-bits overhead and 37-bits result. • Updated the number of supported cascaded DSP blocks for 18-bit and 27-bit systolic FIR modes. • Changed instances of *Quartus II* to *Quartus Prime*

continued...

Date	Version	Changes
May 2015	2015.05.04	• Update Chainin and Chainout support for all Floating Point modes in Supported Combinations of Operational Modes and Features for Variable Precision DSP Block in Intel Arria 10 Devices table. • Added steps to retrieve design templates for Independent Multiplier Mode, Multiplier Adder Sum Mode, and Systolic FIR Mode. • Added Arria 10 Native Floating Point DSP IP core in Operational Modes table.
January 2015	2015.01.23	• Added information for primitive DSP. • Update **Supported Combinations of Operational Modes and Features for Variable Precision DSP Block in Intel Arria 10 Devices** table with the column title **Supported Operation Instance**. • Update resource for Single Precision Floating Point Adders in **Number of Multipliers in Intel Arria 10 Devices** table. • Removed double accumulation registers are set statically in the programming file statement in Accumulator for Fixed-Point Arithmetic section. • Added ALTERA_FP_FUNCTIONS in the list of Quartus II DSP IP for floating-point arithmetic. • Added clarification on operational modes supported for delay registers in fixed-point arithmetic. • Added clarification that both top and bottom internal coefficient and pre-adder must be enabled if these features are being used.
August 2014	2014.08.18	• Added floating-point arithmetic. • Added Dynamic ACCUMULATE, Dynamic LOADCONST, Dynamic SUB, Dynamic NEGATE to variable precision DSP blocks operational modes. • Added top delay registers and bottom delay registers along the input cascade chain. • Added the variable precision DSP block signals that control the piepeline registers within the variable precision DSP block. • Added condition that when both pre-adders within the same DSP block are used, they must share the same operation type (either addition or subtraction). • Updated 55-bit adder. • Added 38-bit adder. • Updated two 18 x 19 modes—where the adder is bypassed. • Updated Decimation to Decimation + Accumulate. • Added Decimation + Chainout Adder for accumulator functions and dynamic control signals. • Added 27 (signed or unsigned) x 27 (signed or unsigned) configuration with 1 multiplier per block. • Removed the chainout adder or accumulator from one sum of two 18 x 19 multipliers with one variable precision DSP block and one 18 x 18 multiplication summed with 36-Bit input mode block diagram. • Updated the basic FIR filter equation. • Added mapping systolic mode user view to variable precision block architecture view. • Added systolic registers are not required in 27-bit systolic FIR mode.
December 2013	2013.12.02	Initial release.

4. Clock Networks and PLLs in Intel Arria 10 Devices

This chapter describes the advanced features of hierarchical clock networks and phase-locked loops (PLLs) in Intel Arria 10 devices. The Intel Quartus Prime software enables the PLLs and their features without external devices.

Related Information

Intel Arria 10 Device Handbook: Known Issues
Lists the planned updates to the Intel Arria 10 Device Handbook chapters.

4.1. Clock Networks

The Intel Arria 10 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
  - Small periphery clock (SPCLK) networks
  - Large periphery clock (LPCLK) networks

4.1.1. Clock Resources in Intel Arria 10 Devices

Table 27. Clock Resources in Intel Arria 10 Devices

Clock Input Pins		
Device	**Number of Resources Available**	**Source of Clock Resource**
• 10AS016 • 10AS022 • 10AX016 • 10AX022	• HSSI: 4 differential • I/O: 32 single-ended or 16 differential	For high-speed serial interface (HSSI): `REFCLK_GXB[L,R][1:4][C,D,E,F,G,H,I,J]_CH[B,T][p,n]` pins For I/O: `CLK_[2,3][A..L]_[0,1][p,n]` pins
• 10AS027 • 10AS032 • 10AX027 • 10AX032	• HSSI: 8 differential • I/O: 32 single-ended or 16 differential	
• 10AS048 • 10AX048	• HSSI: 12 differential • I/O: 48 single-ended or 24 differential	
• 10AS057 • 10AS066 • 10AX057 • 10AX066	• HSSI: 16 differential • I/O: 64 single-ended or 32 differential	
• 10AT090 • 10AT115 • 10AX090 • 10AX115	• HSSI: 32 differential • I/O: 64 single-ended or 32 differential	

GCLK Networks		
Device	**Number of Resources Available**	**Source of Clock Resource**
All	32	• Physical medium attachment (PMA) and physical coding sublayer (PCS) TX and RX clocks per channel • PMA and PCS TX and RX divide clocks per channel • Hard IP core clock output signals • DLL clock outputs • Fractional PLL (fPLL) and I/O PLL C counter outputs • I/O PLL M counter outputs for feedback • REFCLK and clock input pins • Core signals • Phase aligner counter outputs

RCLK Networks		
Device	**Number of Resources Available**	**Source of Clock Resource**
• 10AS016 • 10AS022 • 10AS027 • 10AS032 • 10AX016	8	• Physical medium attachment (PMA) and physical coding sublayer (PCS) TX and RX clocks per channel • PMA and PCS TX and RX divide clocks per channel • Hard IP core clock output signals • DLL clock outputs • fPLL and I/O PLL C counter outputs

continued...

RCLK Networks		
Device	**Number of Resources Available**	**Source of Clock Resource**
• 10AX022 • 10AX027 • 10AX032		• I/O PLL M counter outputs for feedback • REFCLK and clock input pins • Core signals • Phase aligner counter outputs
• 10AS048 • 10AX048	12	
• 10AS057 • 10AS066 • 10AX057 • 10AX066 • 10AT090 • 10AT115 • 10AX090 • 10AX115	16	

SPCLK Networks		
Device	**Number of Resources Available**	**Source of Clock Resource**
• 10AS016 • 10AS022 • 10AX016 • 10AX022 • 10AS027 • 10AS032 • 10AX027 • 10AX032	144	For HSSI: • Physical medium attachment (PMA) and physical coding sublayer (PCS) TX and RX clocks per channel • PMA and PCS TX and RX divide clocks per channel • Hard IP core clock output signals • DLL clock outputs • fPLL C counter outputs • REFCLK and clock input pins • Core signals For I/O: • DPA outputs (LVDS I/O only) • I/O PLL C and M counter outputs • Clock input pins • Core signals • Phase aligner counter outputs
• 10AS048 • 10AX048	216	
• 10AS057 • 10AS066 • 10AX057 • 10AX066	288	
• 10AT090 • 10AT115 • 10AX090 • 10AX115	384	

LPCLK Networks		
Device	**Number of Resources Available**	**Source of Clock Resource**
• 10AS016 • 10AS022 • 10AX016 • 10AX022 • 10AS027 • 10AS032 • 10AX027 • 10AX032	24	For HSSI: • Physical medium attachment (PMA) and physical coding sublayer (PCS) TX and RX clocks per channel • PMA and PCS TX and RX divide clocks per channel • Hard IP core clock output signals • DLL clock outputs

continued...

LPCLK Networks		
Device	**Number of Resources Available**	**Source of Clock Resource**
• 10AS048 • 10AX048	36	• fPLL C and M counter outputs • REFCLK and clock input pins • Core signals For I/O: • DPA outputs (LVDS I/O only) • I/O PLL C and M counter outputs • Clock input pins • Core signals • Phase aligner counter outputs
• 10AS057 • 10AS066 • 10AX057 • 10AX066	48	
• 10AT090 • 10AT115 • 10AX090 • 10AX115	64	

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

Related Information

- Guideline: I/O Standards Supported for I/O PLL Reference Clock Input Pin on page 179
- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
- Guideline: I/O Standards Supported for I/O PLL Reference Clock Input Pin on page 179

4.1.2. Hierarchical Clock Networks

Intel Arria 10 devices cover 3 levels of clock networks hierarchy. The sequence of the hierarchy is as follows:

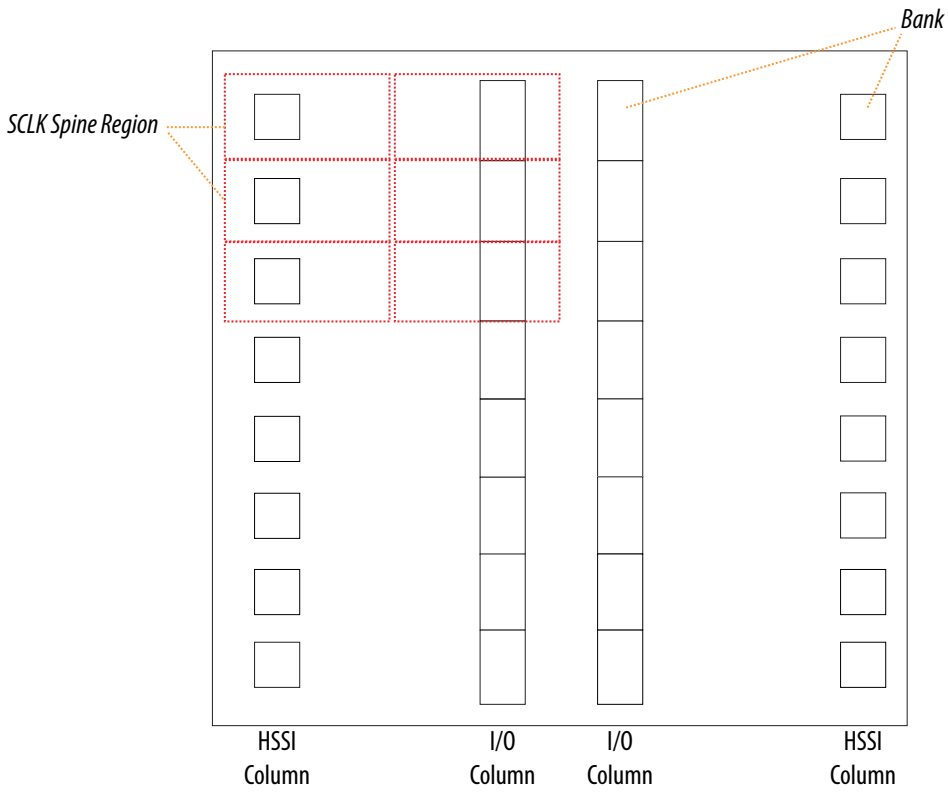
1. GCLK, RCLK, PCLK, and GCLK and RCLK feedback clocks
2. Section clock (SCLK)
3. Row clocks

Each HSSI and I/O column contains clock drivers to drive down shared buses to the respective GCLK, RCLK, and PCLK clock networks.

Intel Arria 10 clock networks (GCLK, RCLK, and PCLK) are routed through SCLK before each clock is connected to the clock routing for each HSSI or I/O bank. The settings for SCLK are transparent. The Intel Quartus Prime software automatically routes the SCLK based on the GCLK, RCLK, and PCLK networks.

Each SCLK spine has a consistent height, matching that of HSSI and I/O banks. The number of SCLK spine in a device depends on the number of HSSI and I/O banks.

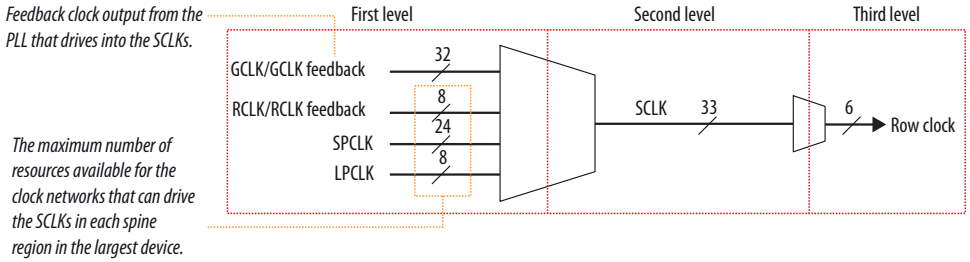
Figure 51. SCLK Spine Regions for Intel Arria 10 Devices



Intel Arria 10 devices provide a maximum of 33 SCLK networks in the SCLK spine region. The SCLK networks can drive six row clocks in each row clock region. The row clocks are the clock resources to the core functional blocks, PLLs, and I/O interfaces, and HSSI interfaces of the device. Six unique signals can be routed into each row clock region. The connectivity pattern of the multiplexers that drive each SCLK limits the clock sources to the SCLK spine region. Each SCLK can select the clock resources from GCLK, RCLK, LPCLK, or SPCLK lines.

The following figure shows SCLKs driven by the GCLK, RCLK, PCLK, or GCLK and RCLK feedback clock networks in each SCLK spine region. The GCLK, RCLK, PCLK, and GCLK and RCLK feedback clocks share the same SCLK routing resources. 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 SCLK spine region.

Figure 52. Hierarchical Clock Networks in SCLK Spine



4.1.3. Types of Clock Networks

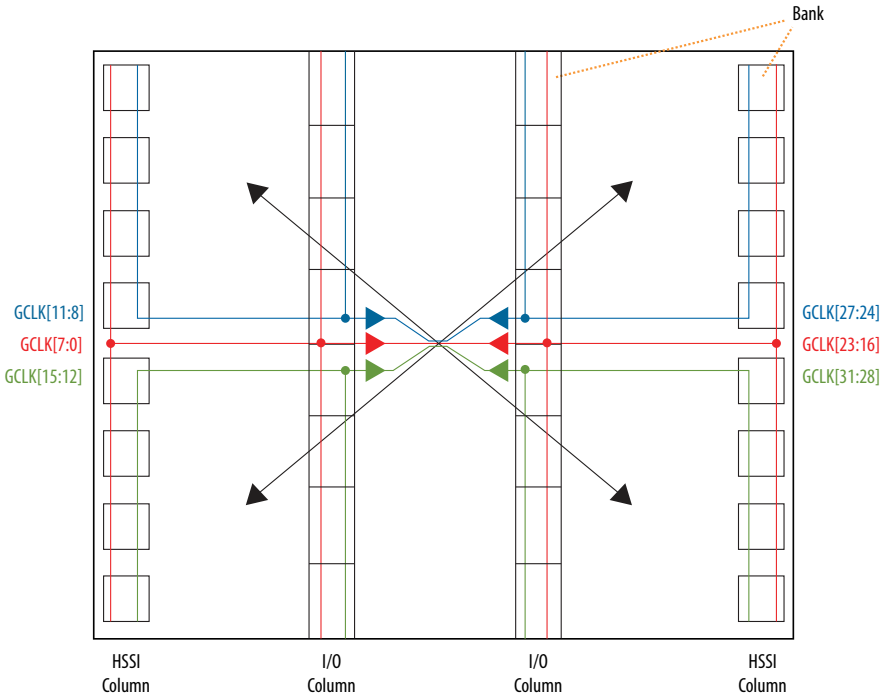
4.1.3.1. Global Clock Networks

GCLK networks serve as low-skew clock sources for functional blocks, such as adaptive logic modules (ALMs), digital signal processing (DSP), embedded memory, and PLLs. Intel Arria 10 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.

Intel Arria 10 devices provide GCLKs that can drive throughout the device. GCLKs cover every SCLK spine region in the device. Each GCLK is accessible through the direction as indicated in the Symbolic GCLK Networks diagram.

Figure 53. Symbolic GCLK Networks in Intel Arria 10 Devices

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



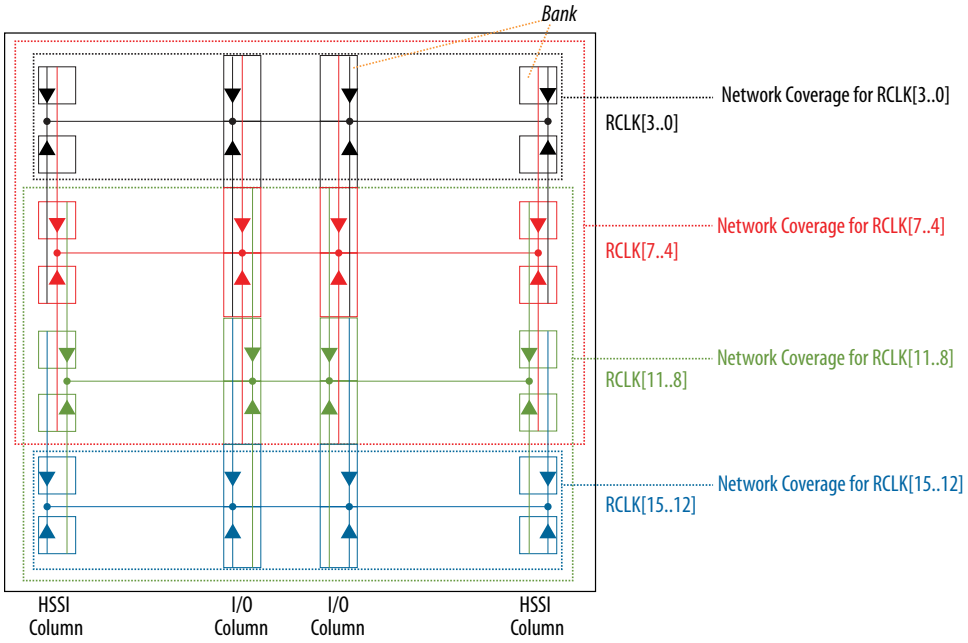
4.1.3.2. Regional Clock Networks

RCLK networks provide low clock insertion delay and skew for logic contained within a single RCLK region. The Intel Arria 10 IOEs and internal logic within a given region can also drive RCLKs to create internally-generated regional clocks and other high fan-out signals.

Intel Arria 10 devices provide RCLKs that can drive through the chip horizontally. RCLKs cover all the SCLK spine regions in the same row of the device. The top and bottom HSSI and I/O banks have RCLKs that cover 2 rows vertically. The other intermediate HSSI and I/O banks have RCLKs that cover 6 rows vertically. The following figure shows the RCLK network coverage.

Figure 54. RCLK Networks in Intel Arria 10 Devices

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



4.1.3.3. Periphery Clock Networks

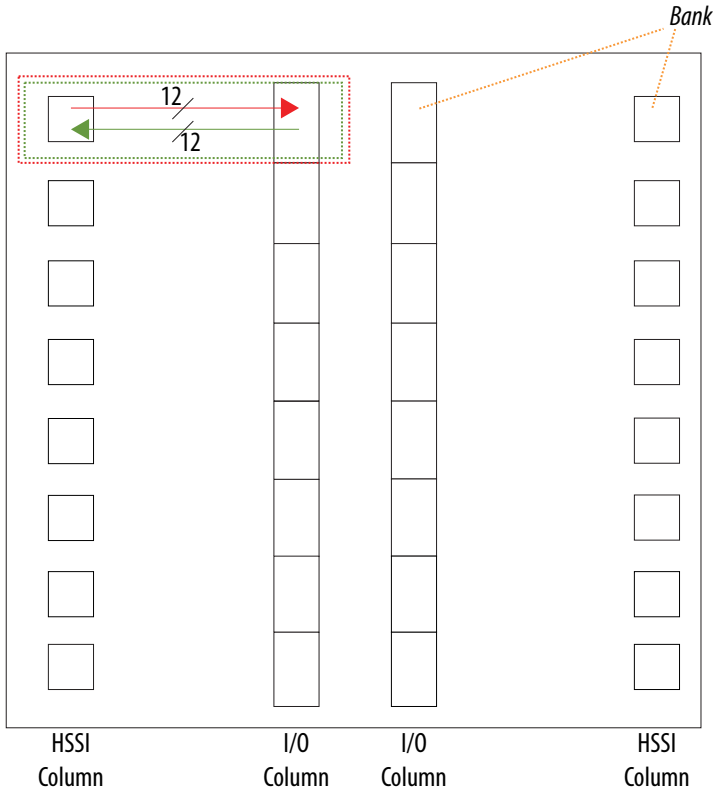
PCLK networks provide the lowest insertion delay and the same skew as RCLK networks.

Small Periphery Clock Networks

Each HSSI or I/O bank has 12 SPCLKs. SPCLKs cover one SCLK spine region in HSSI bank and one SCLK spine region in I/O bank adjacent to each other in the same row.

Figure 55. SPCLK Networks for Intel Arria 10 Devices

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

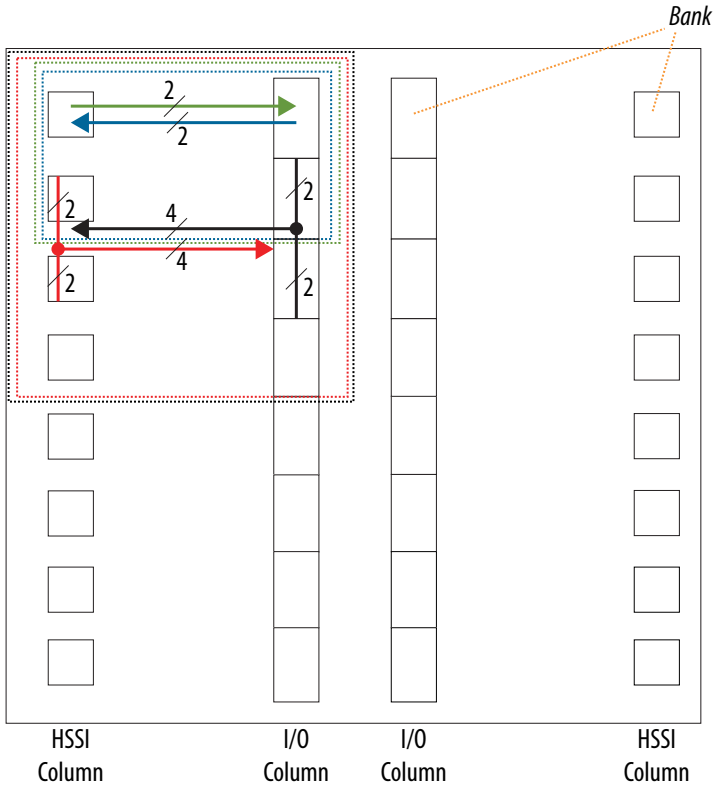


Large Periphery Clock Networks

Each HSSI or I/O bank has 2 LPCLKs. LPCLKs have larger network coverage compared to SPCLKs. LPCLKs cover one SCLK spine region in HSSI bank and one SCLK spine region in I/O bank adjacent to each other in the same row. Top and bottom HSSI and I/O banks have LPCLKs that cover 2 rows vertically. The other intermediate HSSI and I/O banks have LPCLKs that cover 4 rows vertically.

Figure 56. LPCLK Networks for Intel Arria 10 Devices

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



4.1.4. Clock Network Sources

This section describes the clock network sources that can drive the GCLK, RCLK, and PCLK networks.

4.1.4.1. Dedicated Clock Input Pins

The sources of dedicated clock input pins are as follows:

- fPLL—`REFCLK_GXB[L,R][1:4][C,D,E,F,G,H,I,J]_CH[B,T][p,n]` from HSSI column
- I/O PLL—`CLK_[2,3][A..L]_[0,1][p,n]` from I/O column

You can use the dedicated clock input pins for high fan-out control signals, such as asynchronous clears, presets, and clock enables, for protocol signals through the GCLK or RCLK networks.

The dedicated clock input pins can be either differential clocks or single-ended clocks for I/O PLL. For single-ended clock inputs, both the CLKp and CLKn pins have dedicated connections to the I/O PLL. fPLLs only support differential clock inputs.

Driving a PLL over a global or regional clock can lead to higher jitter at the PLL input, and the PLL is not be able to fully compensate for the global or regional clock. Intel recommends using the dedicated clock input pins for optimal performance to drive the PLLs.

Related Information

Guideline: I/O Standards Supported for I/O PLL Reference Clock Input Pin on page 179

4.1.4.2. Internal Logic

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

4.1.4.3. DPA Outputs

Each DPA can drive the PCLK networks.

4.1.4.4. HSSI Clock Outputs

HSSI clock outputs can drive the GCLK, RCLK, and PCLK networks.

4.1.4.5. PLL Clock Outputs

The fPLL and I/O PLL clock outputs can drive all clock networks.

4.1.5. 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)
- Clock power down (static or dynamic clock enable or disable available only for GCLKs and RCLKs)

Related Information

Clock Control Block (ALTCLKCTRL) IP Core User Guide
Provides more information about ALTCLKCTRL IP core and clock multiplexing schemes.

4.1.5.1. Pin Mapping in Intel Arria 10 Devices

Table 28. Mapping Between the Clock Input Pins, PLL Counter Outputs, and Clock Control Block Inputs for HSSI Column

Clock	Fed by
`inclk[0]`	PLL counters `C0` and `C2` from adjacent fPLLs.
`inclk[1]`	PLL counters `C1` and `C3` from adjacent fPLLs.
`inclk[2]` and `inclk[3]`	Any of the two dedicated clock pins on the same HSSI bank.

Table 29. Mapping Between the Clock Input Pins, PLL Counter Outputs, and Clock Control Block Inputs for I/O Column

One counter can only be assigned to one `inclk`.

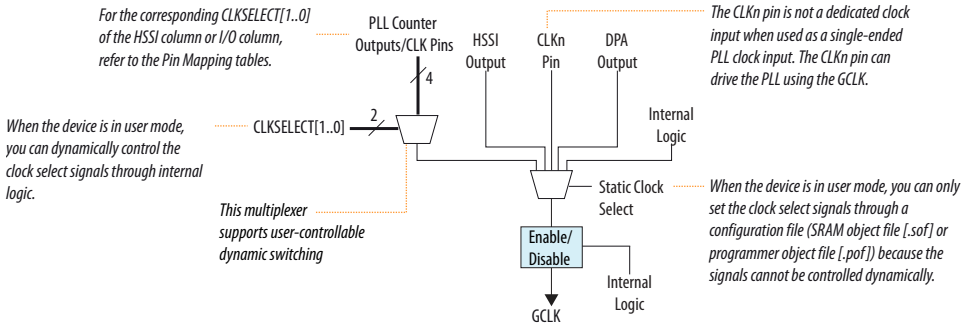
Clock	Fed by
`inclk[0]`	`CLK_[2,3][A..L]_0p` or any counters from adjacent I/O PLLs.
`inclk[1]`	`CLK_[2,3][A..L]_0n` or any counters from adjacent I/O PLLs.
`inclk[2]`	`CLK_[2,3][A..L]_1p` or any counters from adjacent I/O PLLs.
`inclk[3]`	`CLK_[2,3][A..L]_1n` or any counters from adjacent I/O PLLs.

4.1.5.2. 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 57. GCLK Control Block for Intel Arria 10 Devices



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

When selecting the clock source dynamically using the ALTCLKCTRL IP core, choose the inputs using the `CLKSELECT[0..1]` signal.

Note: You can only switch dedicated clock inputs from the same I/O or HSSI bank.

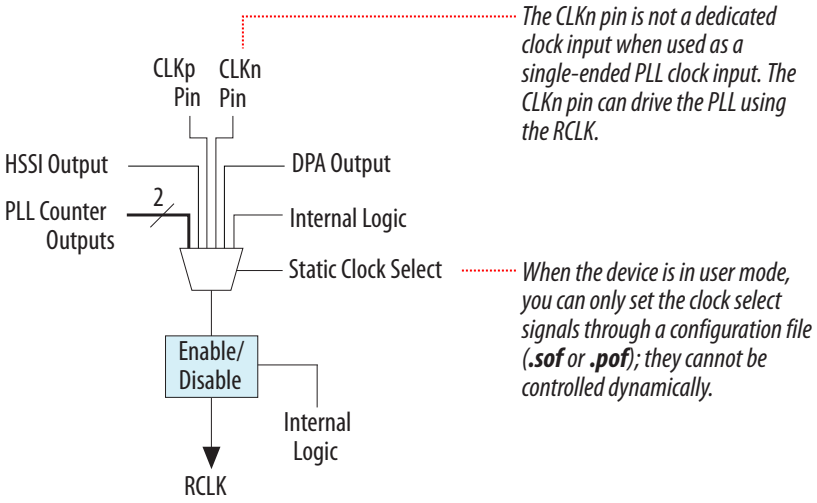
Related Information

Pin Mapping in Intel Arria 10 Devices on page 79
Provides the mapping between the clock input pins, PLL counter outputs, and clock control block inputs for HSSI column and I/O column.

4.1.5.3. 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 58. RCLK Control Block for Intel Arria 10 Devices



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

4.1.5.4. PCLK Control Block

PCLK control block drives both SPCLK and LPCLK networks.

To drive the HSSI PCLK, select the HSSI output, fPLL output, or clock input pin.

To drive the I/O PCLK, select the DPA clock output, I/O PLL output, or clock input pin.

Figure 59. PCLK Control Block for HSSI Column for Intel Arria 10 Devices

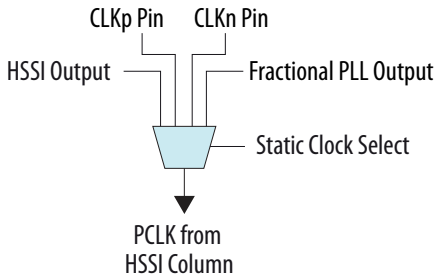
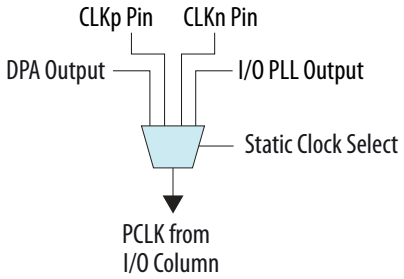


Figure 60. PCLK Control Block for I/O Column for Intel Arria 10 Devices



You can set the input clock sources and the clkena signals for the PCLK networks through the Intel Quartus Prime software using the ALTCLKCTRL IP core.

4.1.6. 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. This feature is independent of the PLL and is applied directly on the clock network.

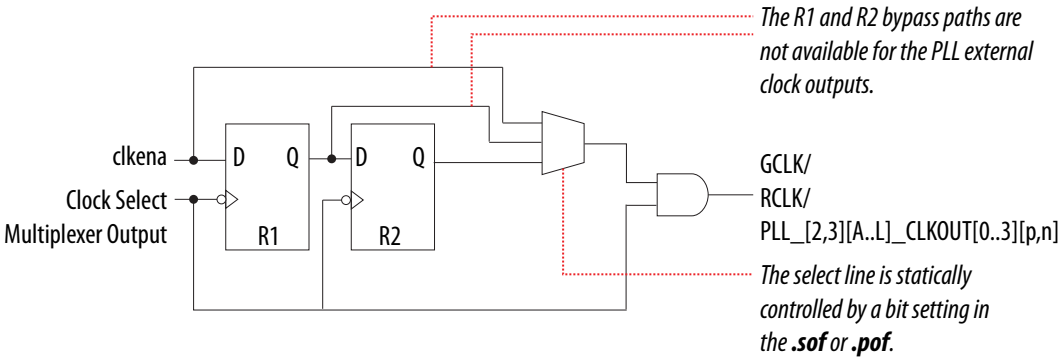
Note: You cannot dynamically enable or disable GCLK or RCLK networks that drive PLLs. Dynamically gating a large clock may affect the chip performance when the core frequency is high.

4.1.7. 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 61. clkena Implementation with Clock Enable and Disable Circuit

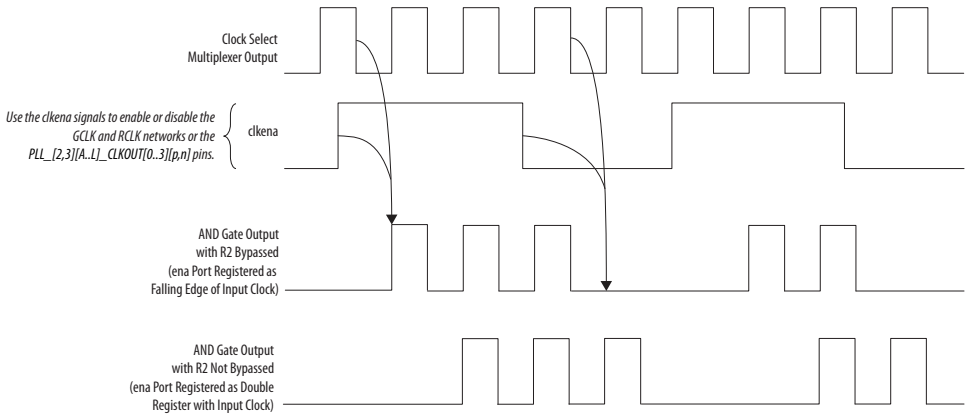
This figure shows the implementation of the clock enable and disable circuit of the clock control block.



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 62. 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.



Intel Arria 10 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.

4.2. Intel Arria 10 PLLs

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

The Intel Arria 10 device family contains the following PLLs:

- fPLLs—can function as fractional PLLs or integer PLLs
- I/O PLLs—can only function as integer PLLs

The fPLLs are located adjacent to the transceiver blocks in the HSSI banks. Each HSSI bank contains two fPLLs. You can configure each fPLL independently in conventional integer mode or fractional mode. In fractional mode, the fPLL can operate with third-order delta-sigma modulation. Each fPLL has four C counter outputs and one L counter output.

The I/O PLLs are located adjacent to the hard memory controllers and LVDS serializer/deserializer (SERDES) blocks in the I/O banks. Each I/O bank contains one I/O PLL. The I/O PLLs can operate in conventional integer mode. Each I/O PLL has nine C counter outputs. In some specific device package, you can use the I/O PLLs in the I/O banks that are not bonded out in your design. These I/O PLLs must take their reference clock source from the FPGA core or through a dedicated cascade connection from another I/O PLL in the same I/O column.

Intel Arria 10 devices have up to 32 fPLLs and 16 I/O PLLs in the largest densities. Intel Arria 10 PLLs have different core analog structure and features support.

Table 30. PLL Features in Intel Arria 10 Devices

Feature	Fractional PLL	I/O PLL
Integer mode	Yes	Yes
Fractional mode	Yes	—
C output counters	4	9
M counter divide factors	8 to 127	4 to 160
N counter divide factors	1 to 32	1 to 80
C counter divide factors	1 to 512	1 to 512
L counter divide factors	1, 2, 4, 8	—
Dedicated external clock outputs	—	Yes
Dedicated clock input pins	Yes	Yes
External feedback input pin	—	Yes
Spread-spectrum input clock tracking (4)	Yes	Yes
Source synchronous compensation	—	Yes
Direct compensation	Yes	Yes
Normal compensation	—	Yes
Zero-delay buffer compensation	—	Yes
External feedback compensation	—	Yes
LVDS compensation	—	Yes
Feedback compensation bonding	Yes	—
Voltage-controlled oscillator (VCO) output drives the DPA clock	—	Yes
Phase shift resolution (5)	72 ps	78.125 ps
Programmable duty cycle	Fixed 50% duty cycle	Yes
Power down mode	Yes	Yes

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

(5) The smallest phase shift is determined by the VCO period divided by four (for fPLL) or eight (for I/O PLL). For degree increments, the Intel Arria 10 device can shift all output frequencies in increments of at least 45° (for I/O PLL) or 90° (for fPLL). Smaller degree increments are possible depending on the frequency and divide parameters.

4.2.1. PLL Usage

fPLLs are optimized for use as transceiver transmit PLLs and for synthesizing reference clock frequencies. You can use the fPLLs 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
- Transmit clocking for transceivers

I/O PLLs are optimized for use with memory interfaces and LVDS SERDES. You can use the I/O PLLs 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
- Simplify the design of external memory interfaces and high-speed LVDS interfaces
- Ease timing closure because the I/O PLLs are tightly coupled with the I/Os
- Compensate clock network delay
- Zero delay buffering

4.2.2. PLL Architecture

Figure 63. Fractional PLL High-Level Block Diagram for Intel Arria 10 Devices

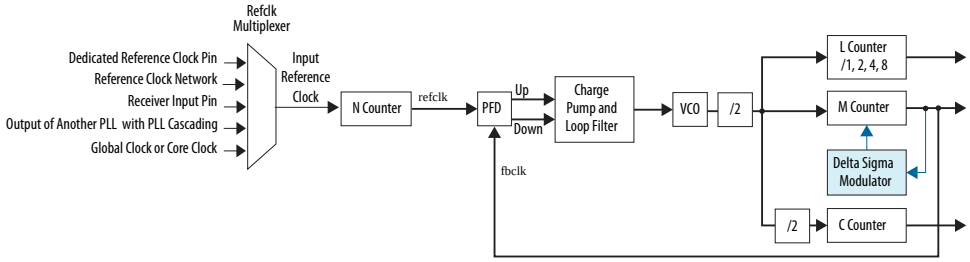
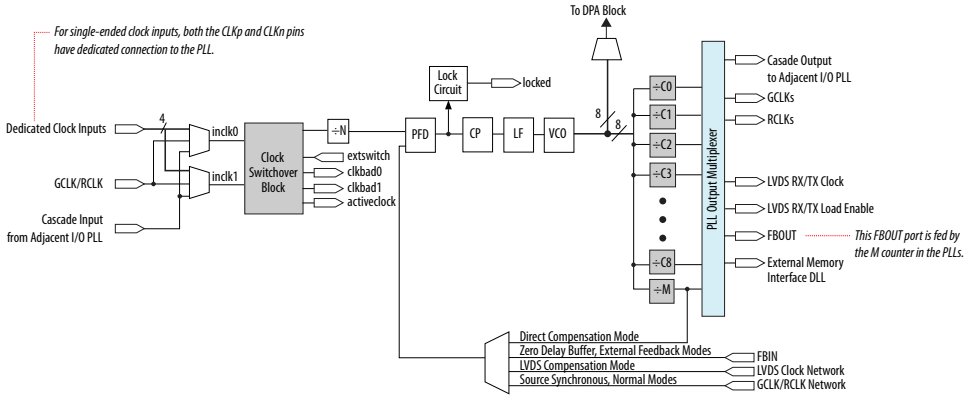


Figure 64. I/O PLL High-Level Block Diagram for Intel Arria 10 Devices



4.2.3. PLL Control Signals

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

4.2.3.1. Reset

The reset signal port of the IP core for each PLL is as follows:

- fPLL—`pll_powerdown`
- I/O PLL—`reset`

The reset signal is the reset or resynchronization input for each PLL. The device input pins or internal logic can drive these input signals.

When the reset signal 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 the reset signal is driven low again, the PLL resynchronizes to its input clock source as it re-locks.

You must assert the reset 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 parameter editor.

You must include the reset 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 when the FPGA transitions into user mode, reset the PLL after the input clock is stable and within specifications, even when the self-reset feature is enabled.
- If the PLL is not able to lock to the reference clock after reconfiguring the PLL or the external clock source, reset the PLL after the input clock is stable and within specifications, even when the self-reset feature is enabled.
- For fPLL, after device power-up, you must reset the fPLL when the fPLL power-up calibration process has completed (`pll_cal_busy` signal deasserts).

4.2.3.2. Locked

The locked signal port of the IP core for each PLL is as follows:

- fPLL—`pll_locked`
- I/O PLL—`locked`

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.

4.2.4. Clock Feedback Modes

Clock feedback modes compensate for clock network delays to align the PLL clock input rising edge with the rising edge of the clock output. Select the appropriate type of compensation for the timing critical clock path in your design.

PLL compensation is not always needed. A PLL should be configured in direct (no compensation) mode unless a need for compensation is identified. Direct mode provides the best PLL jitter performance and avoids expending compensation clocking resources unnecessarily.

The default clock feedback mode is direct compensation mode.

fPLLs support the following clock feedback modes:

- Direct compensation
- Feedback compensation bonding

I/O PLLs support the following clock feedback modes:

- Direct compensation
- Normal compensation
- Source synchronous compensation
- LVDS compensation
- Zero delay buffer (ZDB) compensation
- External feedback (EFB) compensation

Related Information

- Intel FPGA I/O Phase-Locked Loop (Intel FPGA IOPLL) IP Core User Guide
  Provides more information about I/O PLL operation modes.
- PLL Feedback and Cascading Clock Network, Intel Arria 10 Transceiver PHY User Guide
  Provides more information about fPLL operation modes.

4.2.5. Clock Multiplication and Division

An Intel Arria 10 PLL output frequency is related to its input reference clock source by a scale factor of M/(N × C) in integer mode. The input clock is divided by a pre-scale factor, N, and is then multiplied by the M feedback factor. The control loop drives the VCO to match f_{in} × (M/N).

The Intel Quartus Prime software automatically chooses the appropriate scale factors according to the input frequency, multiplication, and division values entered into the Altera IOPLL IP core for I/O PLL and Arria 10 FPLL IP core for fPLL.

Pre-Scale Counter, N and Multiply Counter, M

Each PLL has one pre-scale counter, N, and one multiply counter, M. The M and N counters do not use duty-cycle control because the only purpose of these counters is to calculate frequency division.

Post-Scale Counter, C

Each output port has a unique post-scale counter, C. For multiple C counter 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 I/O PLL are 55 MHz and 100 MHz, the Intel Quartus

Prime software sets the VCO frequency to 1.1 GHz (the least common multiple of 55 MHz and 100 MHz within the VCO operating frequency range). Then the post-scale counters, C, scale down the VCO frequency for each output port.

Post-Scale Counter, L

The fPLL has an additional post-scale counter, L. The L counter synthesizes the frequency from its clock source using the M/(N × L) scale factor. The L counter generates a differential clock pair (0 degree and 180 degree) and drives the HSSI clock network.

Delta-Sigma Modulator

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

Fractional Mode

In fractional mode, the M counter value equals to the sum of the M feedback factor and the fractional value. The fractional value is equal to $K/2^{32}$, where K is an integer between 0 and $(2^{32} - 1)$.

Integer Mode

For a fPLL operating in integer mode, M is an integer value and DSM is disabled.

The I/O PLL can only operate in integer mode.

Related Information

- Intel FPGA I/O Phase-Locked Loop (Intel FPGA IOPLL) IP Core User Guide
  Provides more information about I/O PLL software support in the Intel Quartus Prime software.
- PLLs and Clock Networks chapter, Intel Arria 10 Transceiver PHY User Guide
  Provides more information about fPLL software support in the Intel Quartus Prime software.

4.2.6. Programmable Phase Shift

The programmable phase shift feature allows both fPLLs and I/O 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 (for I/O PLL) or 1/4 (for fPLL) of the VCO period. For example, if an I/O 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.

4.2.7. Programmable Duty Cycle

The programmable duty cycle feature allows I/O PLLs to generate clock outputs with a variable duty cycle. This feature is only supported by the I/O PLL post-scale counters, C. fPLLs do not support the programmable duty cycle feature and only have fixed 50% duty cycle.

The I/O PLL C counter value determines the precision of the duty cycle. The precision is 50% divided by the post-scale counter value. For example, if the C0 counter is 10, steps of 5% are possible for duty-cycle options from 5% to 90%. If the I/O PLL is in external feedback mode, set the duty cycle for the counter driving the `fbin` pin to 50%.

The Intel Quartus Prime software automatically adjusts the VCO frequency according to the required duty cycle that you enter in the Intel FPGA IOPLL IP core parameter editor.

Combining the programmable duty cycle with programmable phase shift allows the generation of precise non-overlapping clocks.

4.2.8. PLL Cascading

Intel Arria 10 devices support PLL-to-PLL cascading. You can only cascade a maximum of two PLLs. The cascaded PLLs must be adjacent PLLs. PLL cascading synthesizes more output clock frequencies than a single PLL.

If you cascade PLLs in your design, the source (upstream) PLL must have a low-bandwidth setting and the destination (downstream) PLL must have a high-bandwidth setting. During cascading, the output of the source PLL serves as the reference clock (input) of the destination PLL. The bandwidth settings of cascaded PLLs must be different. If the bandwidth settings of the cascaded PLLs are the same, the cascaded PLLs may amplify phase noise at certain frequencies.

Intel Arria 10 devices only support I/O-PLL-to-I/O-PLL cascading via dedicated cascade path for core applications. In this mode, upstream I/O PLL and downstream I/O PLL must be located within the same I/O column.

Intel Arria 10 fPLL does not support PLL cascading mode for core applications.

Related Information

- Intel FPGA I/O Phase-Locked Loop (Intel FPGA IOPLL) IP Core User Guide
  Provides more information about I/O PLL cascading in the Intel Quartus Prime software.
- Implementing PLL cascading, Intel Arria 10 Transceiver PHY User Guide
  Provides more information about fPLL cascading in the Intel Quartus Prime software.
- KDB link: How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Intel Arria 10 PLL reference clock?
  For Intel Quartus Prime version prior to 17.1, the I/O PLL and fPLL output may experience additional jitter. The additional jitter occurs if you source the reference clock from a cascaded PLL output, global clock, or core clock. To compensate for the jitter, the designs require additional constraints. This issue has been fixed in the Intel Quartus Prime version 17.1.

4.2.9. Reference Clock Sources

There are three possible reference clock sources to the I/O PLL. The clock can come from a dedicated pin, a core clock network, or the dedicated cascade network.

Intel recommends providing the I/O PLL reference clock using a dedicated pin when possible. If you want to use a non-dedicated pin for the PLL reference clock, you have to explicitly promote the clock to a global signal in the Intel Quartus Prime software.

You can provide up to two reference clocks to the I/O PLL.

- Both reference clocks can come from dedicated pins.
- Only one reference clock can come from a core clock.
- Only one reference clock can come from a dedicated cascade network.

You need to ensure that the PLL input reference clock is in the lock range as stated in the Intel Quartus Prime PLL Usage Summary under Fitter report. PLL loses lock if the input reference clock exceeds the stated range value. You need to reconfigure the PLL if the input reference clock that you are sourcing exceeds this frequency lock range.

Related Information

KDB link: How do I compensate for the jitter of PLL cascading or non-dedicated clock path for Intel Arria 10 PLL reference clock?

For Intel Quartus Prime version prior to 17.1, the I/O PLL and fPLL output may experience additional jitter. The additional jitter occurs if you source the reference clock from a cascaded PLL output, global clock, or core clock. To compensate for the jitter, the designs require additional constraints. This issue has been fixed in the Intel Quartus Prime version 17.1.

4.2.10. 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 to 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`.

Intel Arria 10 PLLs support the following clock switchover modes:

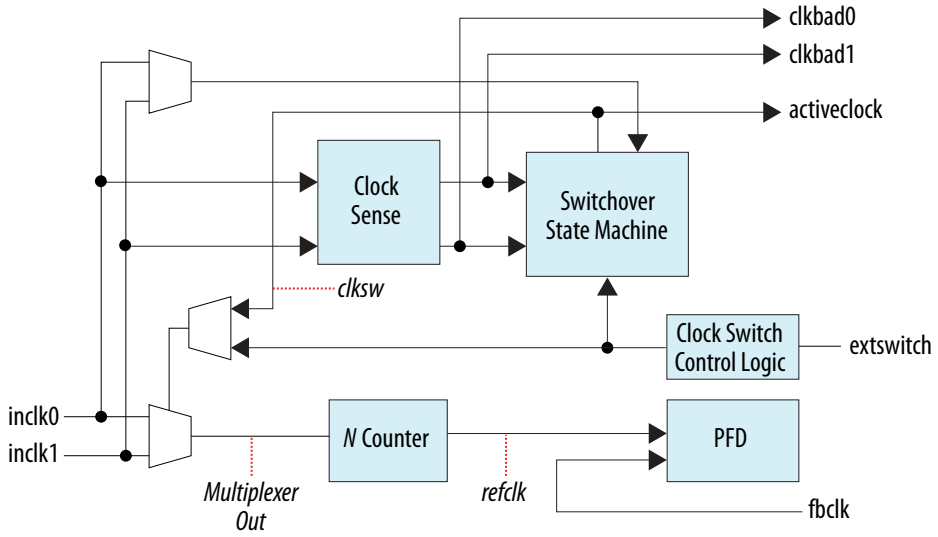
- Automatic switchover—The clock sense circuit monitors the current reference clock. If the current reference clock stops toggling, the reference clock automatically switches to `inclk0` or `inclk1` clock.
- Manual clock switchover—Clock switchover is controlled using the `extswitch` signal. When the `extswitch` signal pulse stays low for at least three clock cycles for the `inclk` being switched to, 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 low, it overrides the automatic clock switchover function. As long as the `extswitch` signal is low, further switchover action is blocked.

4.2.10.1. Automatic Switchover

Intel Arria 10 PLLs support a fully configurable clock switchover capability.

Figure 65. 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—`clkbad0`, `clkbad1`, and `activeclock`—from the PLL to implement a custom switchover circuit in the logic array.

In automatic switchover mode, the `clkbad0` and `clkbad1` 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.

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%.

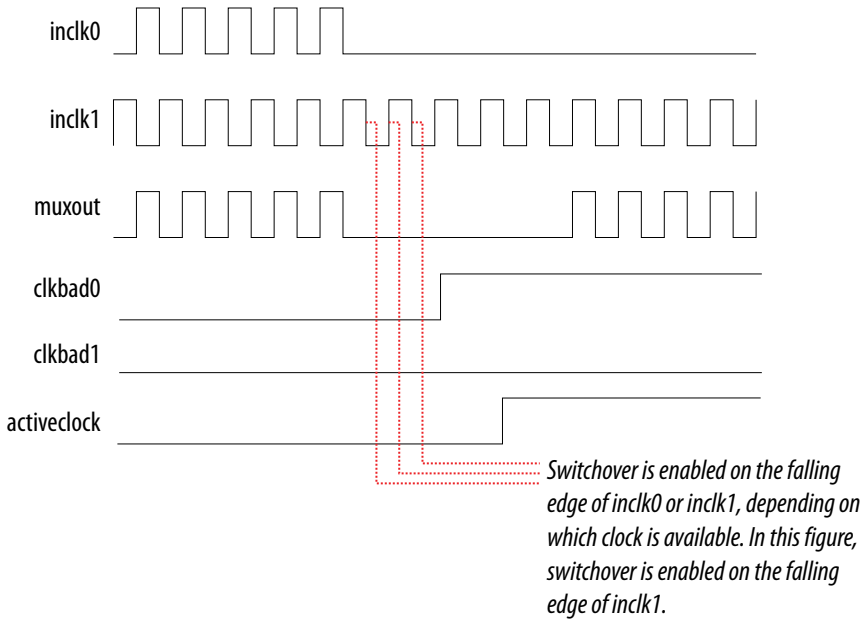
The input clocks must meet the input jitter specifications to ensure proper operation of the status signals. Glitches in the input clock may be seen as a greater than 20% difference in frequency between the input clocks.

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: You must reset the PLL using the reset signal to maintain the phase relationships between the PLL input and output clocks when using clock switchover.

Figure 66. 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 held low. After the `inclk0` signal is held low for approximately two clock cycles, the clock sense circuitry drives the `clkbad0` signal high. As the reference clock signal (`inclk0`) is not toggling, the switchover state machine controls the multiplexer through the `extswitch` signal to switch to the backup clock, `inclk1`.



4.2.10.2. 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.

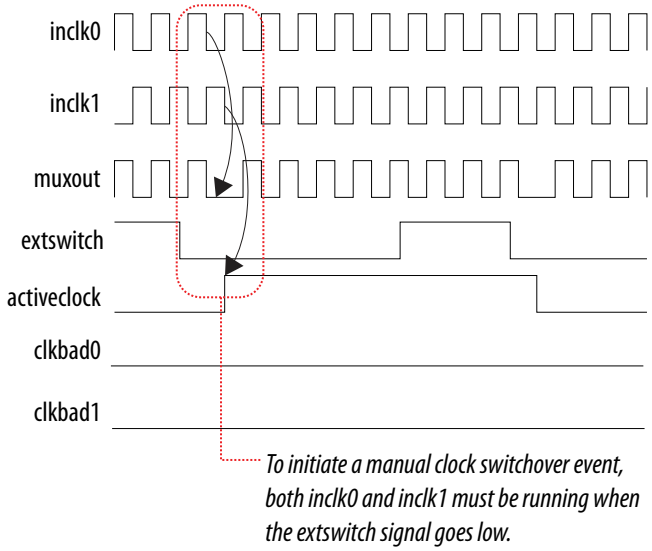
For example, if `inclk0` is 66 MHz and `inclk1` is 200 MHz, you must control switchover using the `extswitch` signal. The automatic clock-sense circuitry cannot monitor clock input (`inclk0` and `inclk1`) frequencies with a frequency difference of more than 100% (2×).

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, L, and K counters so that the VCO operates within the recommended operating frequency range. The Altera IOPLL (for I/O PLL) and Arria 10 FPLL (for fPLL) parameter editors notifies you if a given combination of `inclk0` and `inclk1` frequencies cannot meet this requirement.

Figure 67. 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 switchover sequence starts when the `extswitch` signal goes low. 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 inverts after the `extswitch` signal transitions from logic high to logic low. Since both clocks are still functional during the manual switch, neither `clkbad` signal goes high. Because the switchover circuit is negative-edge sensitive, the rising edge of the `extswitch` signal does not cause the circuit to switch back from `inclk1` to `inclk0`. When the `extswitch` signal goes low 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

- Intel FPGA I/O Phase-Locked Loop (Intel FPGA IOPLL) IP Core User Guide
  Provides more information about I/O PLL software support in the Intel Quartus Prime software.
- PLLs and Clock Networks chapter, Intel Arria 10 Transceiver PHY User Guide
  Provides more information about fPLL software support in the Intel Quartus Prime software.

4.2.10.3. 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.

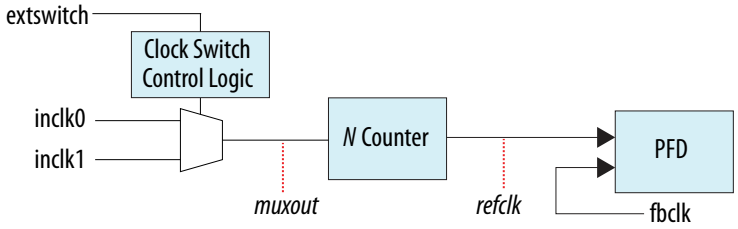
A clock switchover event is initiated when the `extswitch` signal transitions from logic high to logic low, and being held low for at least three `inclk` cycles for the `inclk` being switched to.

You must bring the `extswitch` signal back high again to perform another switchover event. If you do not require another switchover event, you can leave the `extswitch` signal in a logic low state after the initial switch.

Pulsing the `extswitch` signal low for at least three `inclk` cycles for the `inclk` being switched to performs another switchover event.

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

Figure 68. Manual Clock Switchover Circuitry in Intel Arria 10 PLLs



You can delay the clock switchover action by specifying the switchover delay in the Altera IOPLL (for I/O PLL) and Arria 10 FPLL (for fPLL) IP cores. When you specify the switchover delay, the `extswitch` signal must be held low for at least three `inclk` cycles for the `inclk` being switched to plus the number of the delay cycles that has been specified to initiate a clock switchover.

Related Information

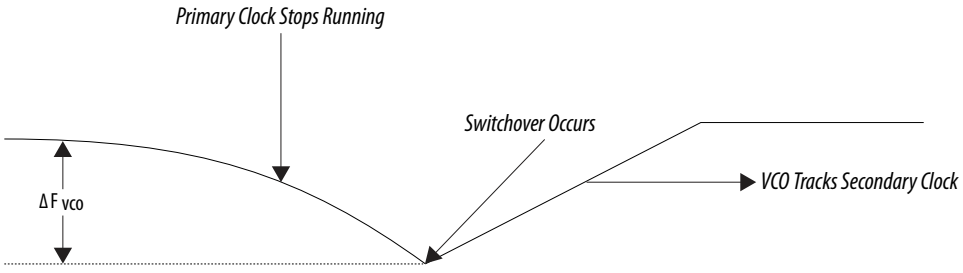
- Intel FPGA I/O Phase-Locked Loop (Intel FPGA IOPLL) IP Core User Guide
  Provides more information about I/O PLL software support in the Intel Quartus Prime software.
- PLLs and Clock Networks chapter, Intel Arria 10 Transceiver PHY User Guide
  Provides more information about fPLL software support in the Intel Quartus Prime software.

4.2.10.4. Guidelines

When implementing clock switchover in Intel Arria 10 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 `clkbad0` and `clkbad1` 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 is likely to 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 low 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 relock 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 the reset signal 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 69. VCO Switchover Operating Frequency



4.2.11. PLL Reconfiguration and Dynamic Phase Shift

fPLLs and I/O PLLs support PLL reconfiguration and dynamic phase shift with the following features:

- PLL reconfiguration—Reconfigure the `M`, `N`, and `C` counters. Able to reconfigure the fractional settings (for fPLL).
- Dynamic phase shift—Perform positive or negative phase shift. fPLLs support only single phase step in one dynamic phase shift operation, where each phase step is equal to 1/4 of the VCO period. I/O PLLs support multiple phase steps in one dynamic phase shift operation, where each phase step is equal to 1/8 of the VCO period.

Related Information

- Intel Arria 10 Transceiver Register Map
  Provides the register address for `M`, `N`, and `C` counters reconfiguration in the Extended Register Map tab.
- AN 728: I/O PLL Reconfiguration and Dynamic Phase Shift for Intel Arria 10 Devices
  Provides more information about implementing I/O PLL reconfiguration in the PLL Reconfig IP core and implementing I/O PLL dynamic phase shift in the IOPLL IP core.
- Reconfiguration Interface and Dynamic Reconfiguration chapter, Intel Arria 10 Transceiver PHY User Guide
  Provides more information about implementing fPLL reconfiguration in the Intel Quartus Prime software.

4.3. Clock Networks and PLLs in Intel Arria 10 Devices Revision History

Document Version	Changes
2019.06.24	• Updated the description for single-ended clock inputs in the *Dedicated Clock Input Pins* section. • Added description about PLL lock range in the *Reference Clock Sources* section.

Date	Version	Changes
December 2017	2017.12.15	• Updated the notes on PLL reset in the *Reset* section. • Updated the description on I/O-PLL-to-I/O-PLL cascading in the *PLL Cascading* section. • Added KDB link on PLL jitter compensation in the following sections: — *PLL Cascading* — *Reference Clock Sources* • Updated the links in the *PLL Reconfiguration and Dynamic Phase Shift* section.
May 2017	2017.05.08	• Updated information on PLL cascading. • Removed all "Preliminary" marks.
March 2017	2017.03.15	Rebranded as Intel.
October 2016	2016.10.31	• Changed clock switchover control signal from `clkswitch` to `extswitch`. • Updated the clock switchover control signal to active low in the Manual Clock Switchover section.

continued...

Date	Version	Changes
May 2016	2016.05.02	• Updated the Clock Resources in Intel Arria 10 Devices table. — Updated the number of resources available for HSSI. — Removed fPLL `M` counter output as the source of clock resource for HSSI. • Updated descriptions on dedicated clock input pins. • Updated the note in Clock Power Down section. • Updated the description on fPLL mode in Intel Arria 10 PLLs section. • Updated Fractional PLL High-Level Block Diagram for Intel Arria 10 diagram. • Removed dedicated refclk input in I/O PLL High-Level Block Diagram for Intel Arria 10 Devices diagram. • Updated supported PLL cascading mode for Intel Arria 10 devices. • Added Reference Clock Sources section.
November 2015	2015.11.02	• Updated the description in *Hierarchical Clock Networks* section: Intel Arria 10 devices provide a maximum of 33 SCLK networks in the SCLK spine region. • Updated GCLK Control Block for Intel Arria 10 Devices diagram. • Removed the following description in the GCLK Control Block section: 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. • Added descriptions about I/O PLL in the Intel Arria 10 PLLs section. • Updated PLL Features in Intel Arria 10 Devices table. — Updated the feature from integer and fractional PLLs to integer and fractional modes. — Updated `M` counter divide factors for fPLL from "1 to 320" to "8 to 127". — Updated `M` counter divide factors for I/O PLL from "1 to 512" to "4 to 160". — Updated `N` counter divide factors for fPLL from "1 to 512" to "1 to 80". — Updated `C` counter divide factors for fPLL from "1 to 320" to "1 to 512". — Removed normal compensation support in fPLL. — Changed "Fractional PLL bonding compensation" to "Feedback compensation bonding". — Updated phase shift resolution for fPLL from 41.667 ps to 72 ps. • Updated compensation mode in Fractional PLL High-Level Block Diagram for Intel Arria 10 Devices. • Updated clock feedback modes for fPLL. — Removed normal compensation. — Changed fPLL bonding compensation to feedback compensation bonding. • Updated description for dynamic phase shift in the PLL Reconfiguration and Dynamic Phase Shift section. • Changed instances of *Quartus II* to *Quartus Prime*.
May 2015	2015.05.04	• Updated the number of RCLK/RCLK feedback from 12 to 8 in the Hierarchical Clock Networks in SCLK Spine diagram. • Added description to the Global Clock Networks section: Each GCLK is accessible through the direction as indicated in the Symbolic GCLK Networks diagram. • Updated HSSI outputs to HSSI clock outputs in the Clock Network Sources section. • Specified that the fPLL and I/O PLL clock outputs can drive all clock networks in the PLL Clock Outputs section. • Added descriptions on PLL cascading bandwidth requirements and PLL cascading modes. • Added a note on fPLL reset requirement in the PLL Control Signals (Reset) section.

continued...

Date	Version	Changes
January 2015	2015.01.23	• Updated the dedicated clock input pins that have dedicated connections to the I/O PLL (CLK_[2,3][A..L]_[0,1][p,n]) when used as single-ended clock inputs. • Removed the dedicated clock input pins, CLK_[2,3][A..L]_[0,1]n, that drive the I/O PLLs over global or regional clock networks and do not have dedicated routing paths to the I/O PLLs. • Removed a note to Internal Logic in the Clock Network Sources section. Note removed: Internally-generated GCLKs or RCLKs cannot drive the Intel Arria 10 PLLs. The input clock to the PLL has to come from dedicated clock input pins, PLL-fed GCLKs, or PLL-fed RCLKs. • Added clock control block pin mapping tables for HSSI and I/O columns. • Updated Fractional PLL High-Level Block Diagram for Intel Arria 10 Devices. Changed CLKp to REFCLK_GXBp and CLKn to REFCLK_GXBn in the note for dedicated clock inputs. • Updated the note to dedicated clock inputs in I/O PLL High-Level Block Diagram for Intel Arria 10 Devices because all four clock inputs can be used as dedicated clock inputs for I/O PLL. The note was changed from "For single-ended clock inputs, only the CLKp pin has a dedicated connection to the PLL. If you use the CLKn pin, a global or regional clock is used." to "For single-ended clock inputs, both the CLKp and CLKn pins have dedicated connection to the PLL." • Added PLL cascading information. • Clarified that when the reset signal is driven low again, the PLL resynchronizes to its input clock source as it re-locks. • Added description for clock feedback mode: Clock feedback modes compensate for clock network delays to align the PLL clock input rising edge with the rising edge of the clock output. Select the appropriate type of compensation for the timing critical clock path in your design. PLL compensation is not always needed. A PLL should be configured in direct (no compensation) mode unless a need for compensation is identified. Direct mode provides the best PLL jitter performance and avoids expending compensation clocking resources unnecessarily. • Updated clock switchover clkswitch signal from positive trigger to negative trigger. • Added the links to the following documents: — Altera I/O Phase-Locked Loop (Altera IOPLL) IP Core User Guide—Provides more information about I/O PLL software support in the Intel Quartus Prime software. — PLLs and Clock Networks chapter, Intel Arria 10 Transceiver PHY User Guide—Provides more information about fPLL software support in the Intel Quartus Prime software. — I/O PLL Reconfiguration and Dynamic Phase Shift for Intel Arria 10 Devices—Provides more information about implementing I/O PLL reconfiguration in Altera PLL Reconfig IP core and implementing I/O PLL dynamic phase shift in Altera IOPLL IP core.
August 2014	2014.08.18	• Updated the dedicated clock input pins name from HSSI banks. • Updated the description in Hierarchical Clock Networks section. • Updated the description in Dedicated Clock Input Pins section. • Removed PCLK network from the Internal Logic section. • Updated the description in PCLK Control Block section. • Updated the following diagrams: — PCLK Control Block for HSSI Column for Arria 10 Devices — PCLK Control Block for I/O Column for Arria 10 Devices • Removed IQTXRXCLK compensation mode. • Updated fractional PLL and I/O PLL high-level block diagrams. • Updated the description for manual clock switchover. • Updated the description for PLL reconfiguration.
December 2013	2013.12.02	Initial release.

5. I/O and High Speed I/O in Intel Arria 10 Devices

The Intel Arria 10 I/Os support the following features:

- Single-ended, non-voltage-referenced, and voltage-referenced I/O standards
- Low-voltage differential signaling (LVDS), RSDS, mini-LVDS, HSTL, HSUL, SSTL, and POD I/O standards
- Serializer/deserializer (SERDES)
- Programmable output current strength
- Programmable slew rate
- Programmable bus-hold
- Programmable weak pull-up resistor
- Programmable pre-emphasis for DDR4 and LVDS standards
- Programmable I/O delay
- Programmable differential output voltage (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)
- HSTL and SSTL input buffer with dynamic power down
- Dynamic on-chip parallel termination for all I/O banks
- Internally generated V_{REF} with DDR4 calibration

Note: The information in this chapter is applicable to all Intel Arria 10 variants, unless noted otherwise.

Related Information

Arria 10 Device Handbook: Known Issues
Lists the planned updates to the *Intel Arria 10 Device Handbook* chapters.

5.1. I/O and Differential I/O Buffers in Intel Arria 10 Devices

The general purpose I/Os (GPIOs) consist of LVDS I/O and 3 V I/O banks:

- LVDS I/O bank—supports differential and single-ended I/O standards up to 1.8 V. The LVDS I/O pins form pairs of true differential LVDS channels. Each pair supports a parallel input/output termination between the two pins. You can use each LVDS channel as transmitter only or receiver only. Each LVDS channel supports transmit SERDES and receive SERDES with DPA circuitry. For example, if you use 30 channels of the available 72 channels as transmitters, you can use the remaining 42 channels as receivers.
- 3 V I/O bank—supports single-ended and differential SSTL, HSTL, and HSUL I/O standards up to 3 V. Single-ended I/O within this I/O bank support all programmable I/O element (IOE) features except:
  - Programmable pre-emphasis
  - R_D on-chip termination (OCT)
  - Calibrated R_S and R_T OCT
  - Internal V_{REF} generation

Intel Arria 10 devices support LVDS on all LVDS I/O banks:

- All LVDS I/O banks support true LVDS input with R_D OCT and true LVDS output buffer.
- The devices do not support emulated LVDS channels.
- The devices support both single-ended and differential I/O reference clock for the I/O PLL that drives the SERDES.

Related Information

- FPGA I/O Resources in Intel Arria 10 GX Packages on page 110
  Lists the number of 3 V and LVDS I/O buffers available inIntel Arria 10 GX packages.
- FPGA I/O Resources in Intel Arria 10 GT Packages on page 111
  Lists the number of 3 V and LVDS I/O buffers available in Intel Arria 10 GT packages.
- FPGA I/O Resources in Intel Arria 10 SX Packages on page 112
  Lists the number of 3 V and LVDS I/O buffers available in Intel Arria 10 SX packages.

5.2. I/O Standards and Voltage Levels in Intel Arria 10 Devices

The Intel Arria 10 device family consists of FPGA and SoC devices. The Intel Arria 10 FPGA devices have only FPGA I/O buffers. The Intel Arria 10 SoC devices have FPGA I/O and HPS I/O buffers. The HPS I/O buffers in Intel Arria 10 SoC devices support different I/O standards than the FPGA I/O buffers.

5.2.1. I/O Standards Support for FPGA I/O in Intel Arria 10 Devices

Table 31. Supported I/O Standards in FPGA I/O for Intel Arria 10 Devices

I/O Standard	Device Variant Support	I/O Buffer Type Support		Application	Standard Support
		LVDS I/O	3V I/O		
3.0 V LVTTL/3.0 V LVCMOS	Devices with 3 V I/O banks only. Refer to related information.	No	Yes	General purpose	JESD8-B
2.5 V LVCMOS	Devices with 3 V I/O banks only. Refer to related information.	No	Yes	General purpose	JESD8-5
1.8 V LVCMOS	All	Yes	Yes	General purpose	JESD8-7
1.5 V LVCMOS	All	Yes	Yes	General purpose	JESD8-11
1.2 V LVCMOS	All	Yes	Yes	General purpose	JESD8-12
SSTL-18 Class I and Class II	All	Yes	Yes	DDR2	JESD8-15
SSTL-15 Class I and Class II	All	Yes	Yes	DDR3	—
SSTL-15	All	Yes	Yes	DDR3	JESD79-3D
SSTL-135, SSTL-135 Class I and Class II	All	Yes	Yes	DDR3L	—
SSTL-125, SSTL-125 Class I and Class II	All	Yes	Yes	DDR3U	—
SSTL-12, SSTL-12 Class I and Class II	All	Yes	No	RLDRAM 3	—
POD12	All	Yes	No	DDR4	JESD8-24
1.8 V HSTL Class I and Class II	All	Yes	Yes	DDR II+, QDR II+, and RLDRAM 2	JESD8-6
1.5 V HSTL Class I and Class II	All	Yes	Yes	DDR II+, QDR II+, QDR II, and RLDRAM 2	JESD8-6
1.2 V HSTL Class I and Class II	All	Yes	Yes	General purpose	JESD8-16A
HSUL-12	All	Yes	Yes	LPDDR2	—
Differential SSTL-18 Class I and Class II	All	Yes	Yes	DDR2	JESD8-15
Differential SSTL-15 Class I and Class II	All	Yes	Yes	DDR3	—
Differential SSTL-15	All	Yes	Yes	DDR3	JESD79-3D

continued...

I/O Standard	Device Variant Support	I/O Buffer Type Support		Application	Standard Support
		LVDS I/O	3V I/O		
Differential SSTL-135, SSTL-135 Class I and Class II	All	Yes	Yes	DDR3L	—
Differential SSTL-125, SSTL-125 Class I and Class II	All	Yes	Yes	DDR3U	—
Differential SSTL-12, SSTL-12 Class I and Class II	All	Yes	No	RLDRAM 3	—
Differential POD12	All	Yes	No	DDR4	JESD8-24
Differential 1.8 V HSTL Class I and Class II	All	Yes	Yes	DDR II+, QDR II+, and RLDRAM 2	JESD8-6
Differential 1.5 V HSTL Class I and Class II	All	Yes	Yes	DDR II+, QDR II+, QDR II, and RLDRAM 2	JESD8-6
Differential 1.2 V HSTL Class I and Class II	All	Yes	Yes	General purpose	JESD8-16A
Differential HSUL-12	All	Yes	Yes	LPDDR2	—
LVDS	All	Yes	No	SGMII, SFI, and SPI	ANSI/TIA/EIA-644
Mini-LVDS	All	Yes	No	SGMII, SFI, and SPI	—
RSDS	All	Yes	No	SGMII, SFI, and SPI	—
LVPECL	All	Yes	No	SGMII, SFI, and SPI	—

Related Information

- FPGA I/O Resources in Intel Arria 10 GX Packages on page 110
  Lists the number of 3 V and LVDS I/O buffers available inIntel Arria 10 GX packages.
- FPGA I/O Resources in Intel Arria 10 GT Packages on page 111
  Lists the number of 3 V and LVDS I/O buffers available in Intel Arria 10 GT packages.
- FPGA I/O Resources in Intel Arria 10 SX Packages on page 112
  Lists the number of 3 V and LVDS I/O buffers available in Intel Arria 10 SX packages.

5.2.2. I/O Standards Support for HPS I/O in Intel Arria 10 Devices

Table 32. Supported I/O Standards in HPS I/O for Intel Arria 10 SX Devices

I/O Standard	Application	Standard Support
3.0 V LVTTL/3.0 V LVCMOS	General purpose	JESD8-B
2.5 V LVCMOS	General purpose	JESD8-5
1.8 V LVCMOS	General purpose	JESD8-7

5.2.3. I/O Standards Voltage Levels in Intel Arria 10 Devices

Intel Arria 10 devices in all packages can interface with systems of different supply voltages.

- The I/O buffers are powered by V_{CC}, V_{CCPT} and V_{CCIO}.
- Each I/O bank has its own V_{CCIO} supply and supports only one V_{CCIO} voltage.
- In all I/O banks, you can use any of the listed V_{CCIO} voltages except 2.5 V and 3.0 V.
- The 2.5 V and 3.0 V V_{CCIO} voltages are supported only on the 3 V I/O banks.
- For the maximum and minimum input voltages allowed, refer to the device datasheet.

Table 33. Intel Arria 10 I/O Standards Voltage Levels

This table lists the typical power supplies for each supported I/O standards in Intel Arria 10 devices.

I/O Standard	V_{CCIO}(V)		V_{CCPT}(V) (Pre-Driver Voltage)	V_{REF}(V) (Input Ref Voltage)	V_{TT}(V) (Board Termination Voltage)
	Input[6]	Output			
3.0 V LVTTL/3.0 V LVCMOS	3.0/2.5[7]	3.0	1.8	—	—
2.5 V LVCMOS	3.0/2.5	2.5	1.8	—	—
1.8 V LVCMOS	1.8	1.8	1.8	—	—
1.5 V LVCMOS	1.5	1.5	1.8	—	—
1.2 V LVCMOS	1.2	1.2	1.8	—	—
SSTL-18 Class I and Class II	V_{CCPT}	1.8	1.8	0.9	0.9
SSTL-15 Class I and Class II	V_{CCPT}	1.5	1.8	0.75	0.75
SSTL-15	V_{CCPT}	1.5	1.8	0.75	0.75
SSTL-135, SSTL-135 Class I and Class II	V_{CCPT}	1.35	1.8	0.675	—
SSTL-125, SSTL-125 Class I and Class II	V_{CCPT}	1.25	1.8	0.625	—
SSTL-12, SSTL-12 Class I and Class II	V_{CCPT}	1.2	1.8	0.6	—
POD12	V_{CCPT}	1.2	1.8	0.84	1.2
1.8 V HSTL Class I and Class II	V_{CCPT}	1.8	1.8	0.9	0.9
1.5 V HSTL Class I and Class II	V_{CCPT}	1.5	1.8	0.75	0.75
1.2 V HSTL Class I and Class II	V_{CCPT}	1.2	1.8	0.6	0.6
HSUL-12	V_{CCPT}	1.2	1.8	0.6	—
Differential SSTL-18 Class I and Class II	V_{CCPT}	1.8	1.8	—	0.9
Differential SSTL-15 Class I and Class II	V_{CCPT}	1.5	1.8	—	0.75
Differential SSTL-15	V_{CCPT}	1.5	1.8	—	0.75

continued...

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

(7) When using a 3.0 V LVTTL /3.0 V LVCMOS I/O standard with a 2.5 V V_{CCIO} voltage, input signals that exceed 2.5 V will be clamped as the clamping diode is turned on.

I/O Standard	V_{CCIO}(V)		V_{CCPT}(V) (Pre-Driver Voltage)	V_{REF}(V) (Input Ref Voltage)	V_{TT}(V) (Board Termination Voltage)
	Input(6)	Output			
Differential SSTL-135, SSTL-135 Class I and Class II	V_{CCPT}	1.35	1.8	—	0.675
Differential SSTL-125, SSTL-125 Class I and Class II	V_{CCPT}	1.25	1.8	—	0.625
Differential SSTL-12, SSTL-12 Class I and Class II	V_{CCPT}	1.2	1.8	—	0.6
Differential POD12	V_{CCPT}	1.2	1.8	—	1.2
Differential 1.8 V HSTL Class I and Class II	V_{CCPT}	1.8	1.8	—	0.9
Differential 1.5 V HSTL Class I and Class II	V_{CCPT}	1.5	1.8	—	0.75
Differential 1.2 V HSTL Class I and Class II	V_{CCPT}	1.2	1.8	—	0.6
Differential HSUL-12	V_{CCPT}	1.2	1.8	—	—
LVDS	V_{CCPT}	1.8	1.8	—	—
Mini-LVDS	V_{CCPT}	1.8	1.8	—	—
RSDS	V_{CCPT}	1.8	1.8	—	—
LVPECL (Differential clock input only)	V_{CCPT}	—	1.8	—	—

Related Information

- Guideline: Observe Device Absolute Maximum Rating for 3.0 V Interfacing on page 178
- Guideline: VREF Sources and VREF Pins on page 177
- I/O Standard Specifications, *Intel Arria 10 Device Datasheet*
  Lists the maximum and minimum input voltages (V_{IH} and V_{IL}), output voltages (V_{OH} and V_{OL}), and current drive characteristics (I_{OH} and I_{OL}) for various I/O standards supported by Intel Arria 10 devices.

5.3. Intel FPGA I/O IP Cores for Intel Arria 10 Devices

The I/O system is supported by several Intel FPGA I/O IP cores.

- Intel FPGA GPIO—supports operations of the GPIO components.
- Intel FPGA LVDS SERDES—supports operations of the high-speed source-synchronous SERDES.
- Intel FPGA OCT—supports the OCT calibration block.
- Intel FPGA PHYlite for Parallel Interfaces —supports dynamic OCT and I/O delays for strobe-based capture I/O elements. This IP core can also be used for generic source synchronous interfaces using single ended I/O.

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

Related Information

- *IntelFPGA PHYLite for Parallel Interfaces IP Core User Guide*
- *Intel FPGA GPIO IP Core User Guide*
- *Intel FPGA OCT IP Core User Guide*
- *Intel FPGA LVDS SERDES IP Core User Guide*

5.4. I/O Resources in Intel Arria 10 Devices

5.4.1. GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices

The I/O banks are located in I/O columns. Each I/O bank contains its own PLL, DPA, and SERDES circuitries.

For more details about the I/O banks available in each device package, refer to the related information.

Figure 70. I/O Banks for Intel Arria 10 GX 160 and GX 220 Devices

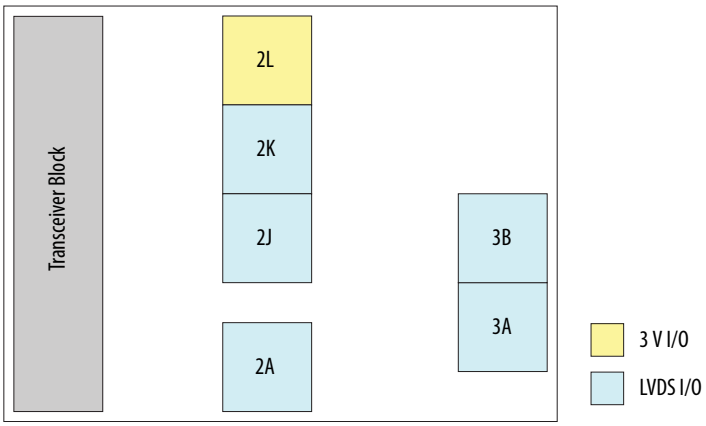


Figure 71. I/O Banks for Intel Arria 10 SX 160 and SX 220 Devices

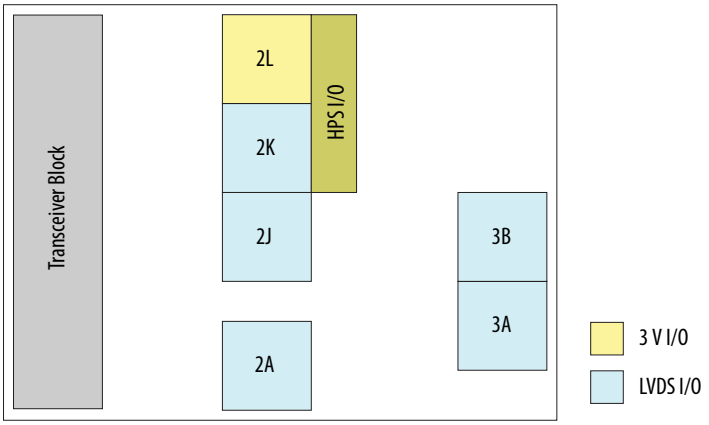


Figure 72. I/O Banks for Intel Arria 10 GX 270 and GX 320 Devices

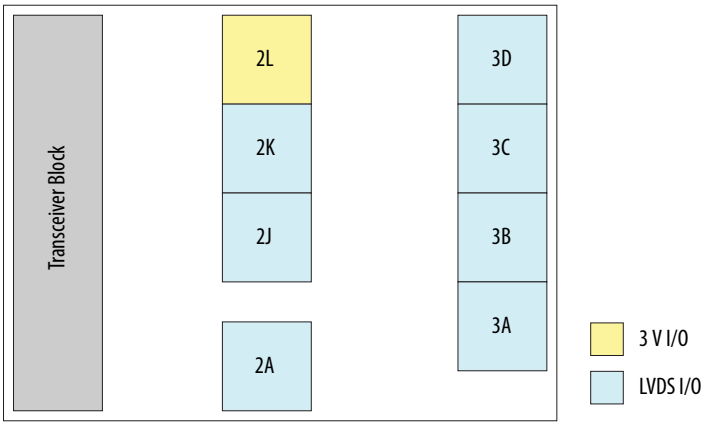


Figure 73. I/O Banks for Intel Arria 10 SX 270 and SX 320 Devices

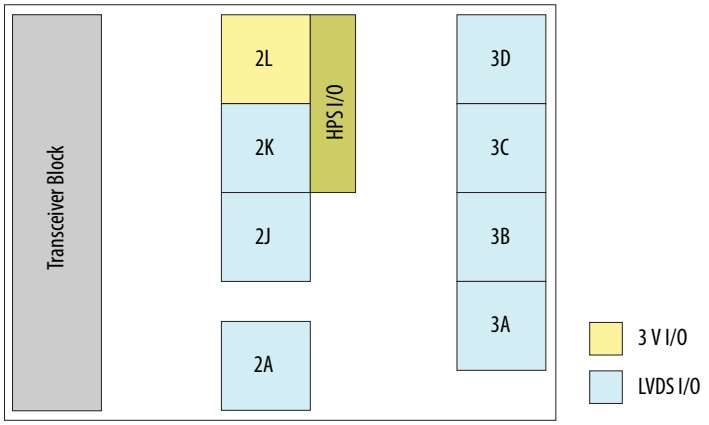


Figure 74. I/O Banks for Intel Arria 10 GX 480 Devices

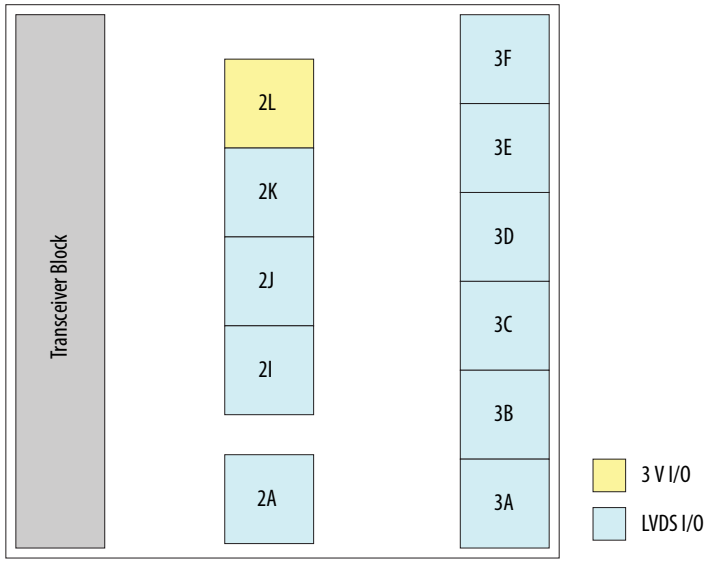


Figure 75. I/O Banks for Intel Arria 10 SX 480 Devices

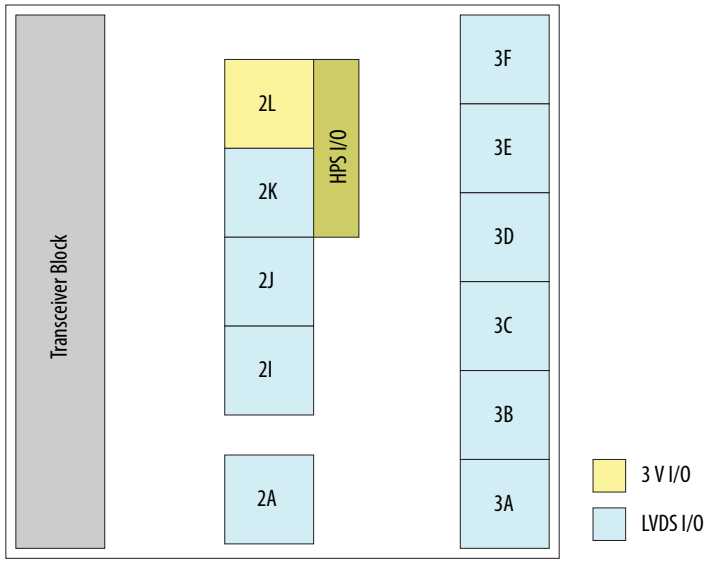


Figure 76. I/O Banks for Intel Arria 10 GX 570 and GX 660 Devices

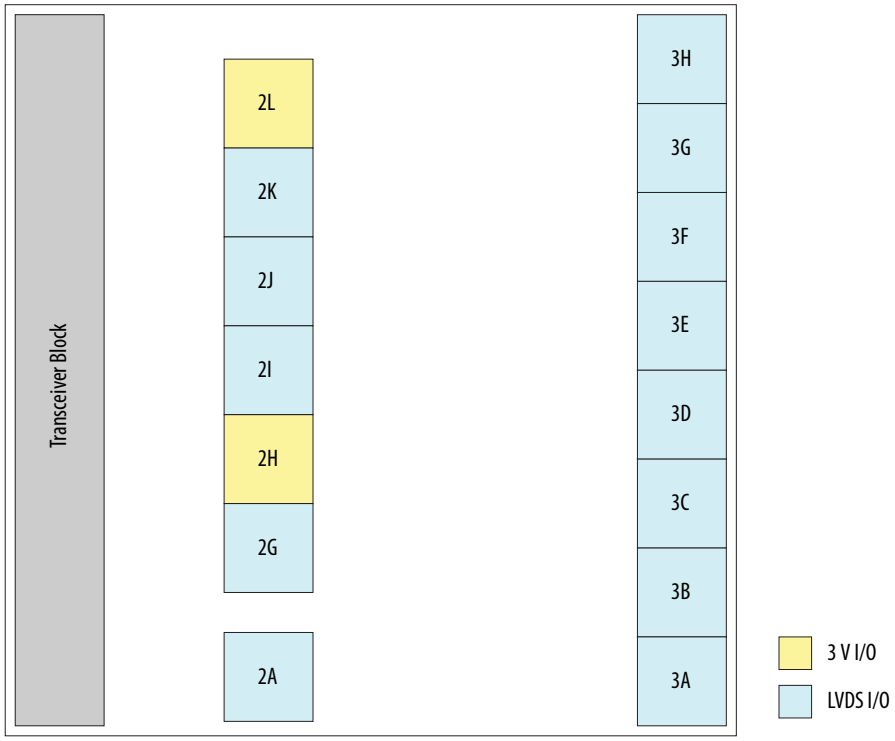


Figure 77. I/O Banks for Intel Arria 10 SX 570 and SX 660 Devices

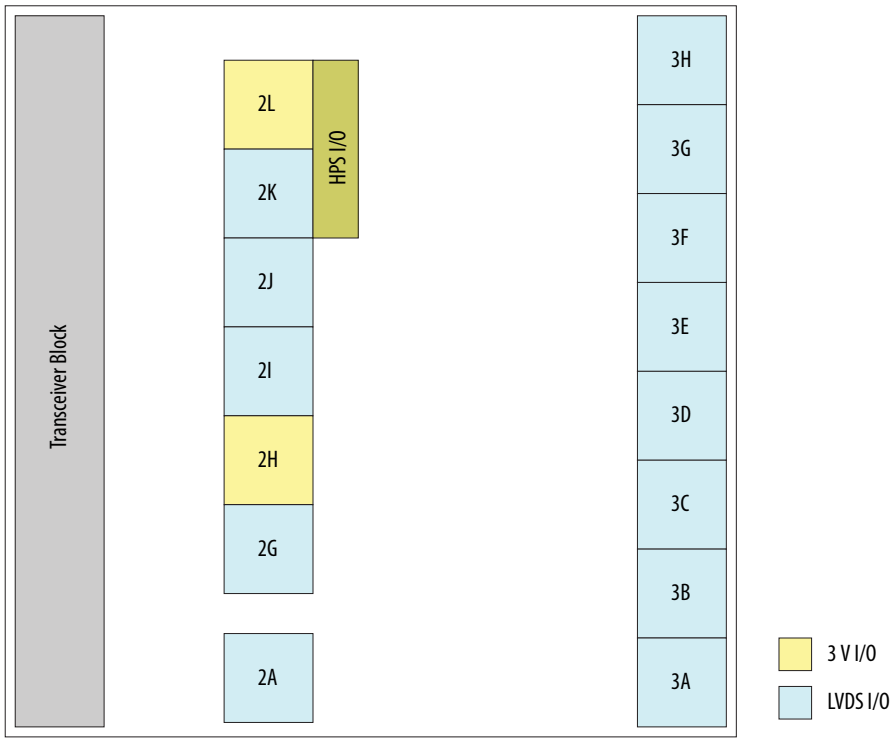
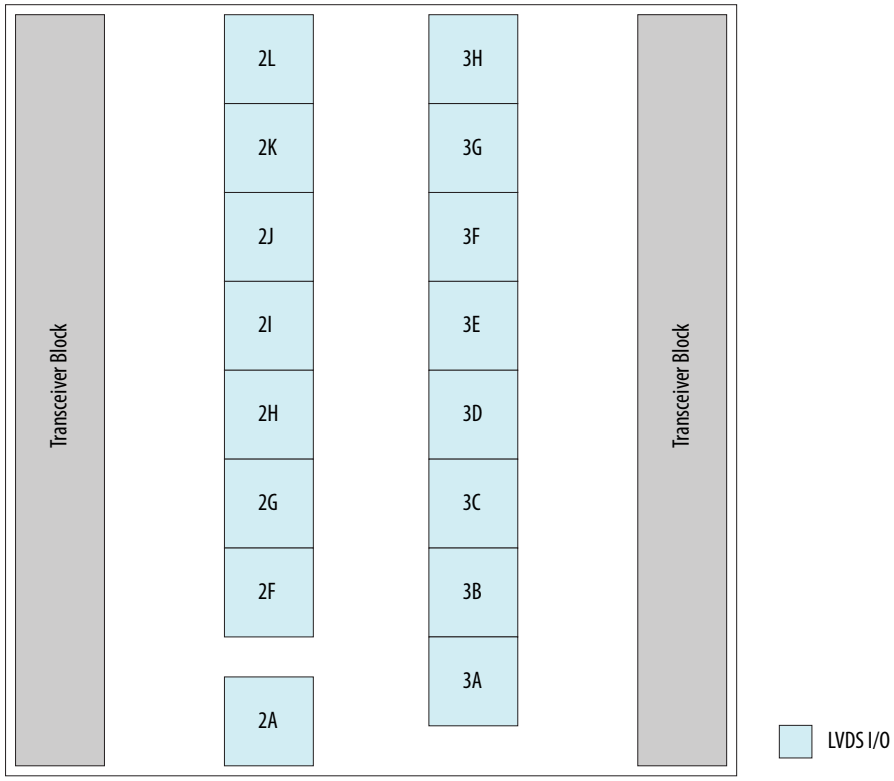


Figure 78. I/O Banks for Intel Arria 10 GX 900, GX 1150, GT 900, and GT 1150 Devices



Related Information

- Device Transceiver Layout
  Provides more information about the transceiver banks in Intel Arria 10 devices.
- I/O Banks for Intel Arria 10 GX Devices on page 113
  Lists the number of I/O pins in the available I/O banks for each Intel Arria 10 GX package.
- I/O Banks for Intel Arria 10 GT Devices on page 116
  Lists the number of I/O pins in the available I/O banks for each Intel Arria 10 GT package.
- I/O Banks for Intel Arria 10 SX Devices on page 117
  Lists the number of I/O pins in the available I/O banks for each Intel Arria 10 SX package.
- FPGA I/O Resources in Intel Arria 10 GX Packages on page 110
  Lists the number of 3 V and LVDS I/O buffers available inIntel Arria 10 GX packages.
- FPGA I/O Resources in Intel Arria 10 GT Packages on page 111
  Lists the number of 3 V and LVDS I/O buffers available in Intel Arria 10 GT packages.
- FPGA I/O Resources in Intel Arria 10 SX Packages on page 112
  Lists the number of 3 V and LVDS I/O buffers available in Intel Arria 10 SX packages.

- Intel Arria 10 Device Pin-Out Files
  Provides the pin-out file for each Intel Arria 10 device. For the SoC devices, the pin-out files also list the I/O banks that are shared by the FPGA fabric and the HPS.
- *Intel FPGA GPIO IP Core User Guide*
- PLLs and Clocking for Intel Arria 10 Devices on page 161

5.4.2. GPIO Buffers and LVDS Channels in Intel Arria 10 Devices

5.4.2.1. FPGA I/O Resources in Intel Arria 10 GX Packages

Table 34. GPIO Buffers and LVDS Channels in Intel Arria 10 GX Devices

- The U19 package is a ball grid array with 0.8 mm pitch. All other packages are ball grid arrays with 1.0 mm pitch.
- The number of LVDS channels does not include dedicated clock pins.

Product Line	Package		GPIO			LVDS Channels
	Code	Type	3 V I/O	LVDS I/O	Total	
GX 160	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
GX 220	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
GX 270	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
GX 320	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
GX 480	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
GX 570	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300
GX 660	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174

continued...

Product Line	Package		GPIO			LVDS Channels
	Code	Type	3 V I/O	LVDS I/O	Total	
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300
GX 900	F34	1,152-pin FBGA	0	504	504	252
	NF40	1,517-pin FBGA	0	600	600	300
	RF40	1,517-pin FBGA	0	342	342	154
	NF45	1,932-pin FBGA	0	768	768	384
	SF45	1,932-pin FBGA	0	624	624	312
	UF45	1,932-pin FBGA	0	480	480	240
GX 1150	F34	1,152-pin FBGA	0	504	504	252
	NF40	1,517-pin FBGA	0	600	600	300
	RF40	1,517-pin FBGA	0	342	342	154
	NF45	1,932-pin FBGA	0	768	768	384
	SF45	1,932-pin FBGA	0	624	624	312
	UF45	1,932-pin FBGA	0	480	480	240

Related Information

- I/O Banks for Intel Arria 10 GX Devices on page 113
  Lists the number of I/O pins in the available I/O banks for each Intel Arria 10 GX package.
- I/O Standards Support for FPGA I/O in Intel Arria 10 Devices on page 101
- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- I/O and Differential I/O Buffers in Intel Arria 10 Devices on page 100

5.4.2.2. FPGA I/O Resources in Intel Arria 10 GT Packages

Table 35. GPIO Buffers and LVDS Channels in Intel Arria 10 GT Devices

- The SF45 package is a ball grid array with 1.0 mm pitch.
- The number of LVDS channels does not include dedicated clock pins.

Product Line	Package		GPIO Buffers			LVDS Channels
	Code	Type	3 V I/O	LVDS I/O	Total	
GT 900	SF45	1,932-pin FBGA	0	624	624	312
GT 1150	SF45	1,932-pin FBGA	0	624	624	312

Related Information

- I/O Banks for Intel Arria 10 GT Devices on page 116
  Lists the number of I/O pins in the available I/O banks for each Intel Arria 10 GT package.
- I/O Standards Support for FPGA I/O in Intel Arria 10 Devices on page 101
- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105

- I/O and Differential I/O Buffers in Intel Arria 10 Devices on page 100

5.4.2.3. FPGA I/O Resources in Intel Arria 10 SX Packages

Table 36. GPIO Buffers and LVDS Channels in Intel Arria 10 SX Devices

- The U19 package is a ball grid array with 0.8 mm pitch. All other packages are ball grid arrays with 1.0 mm pitch.
- The number of LVDS channels does not include dedicated clock pins.

Product Line	Package		GPIO Buffers			LVDS Channels
	Code	Type	3 V I/O	LVDS I/O	Total	
SX 160	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
SX 220	U19	484-pin UBGA	48	148	196	74
	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	240	288	120
SX 270	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
SX 320	F27	672-pin FBGA	48	192	240	96
	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	336	384	168
	F35	1,152-pin FBGA	48	336	384	168
SX 480	F29	780-pin FBGA	48	312	360	156
	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
SX 570	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300
SX 660	F34	1,152-pin FBGA	48	444	492	222
	F35	1,152-pin FBGA	48	348	396	174
	NF40	1,517-pin FBGA	48	540	588	270
	KF40	1,517-pin FBGA	96	600	696	300

Related Information

- I/O Banks for Intel Arria 10 SX Devices on page 117
  Lists the number of I/O pins in the available I/O banks for each Intel Arria 10 SX package.
- I/O Standards Support for FPGA I/O in Intel Arria 10 Devices on page 101

- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- I/O and Differential I/O Buffers in Intel Arria 10 Devices on page 100

5.4.3. I/O Banks Groups in Intel Arria 10 Devices

The I/O pins in Intel Arria 10 devices are arranged in groups called I/O banks:

- The I/O banks have independent supplies that allow each bank to support different I/O standards.
- Each I/O bank can support multiple I/O standards that use the same voltage.

Related Information

- I/O Banks for Intel Arria 10 GX Devices on page 113
- I/O Banks for Intel Arria 10 GT Devices on page 116
- I/O Banks for Intel Arria 10 SX Devices on page 117

5.4.3.1. I/O Banks for Intel Arria 10 GX Devices

The following tables list the I/O banks available, the total number of I/O pins in each bank, and the total number of I/O pins for each product line and device package of the Intel Arria 10 GX device family variant.

Table 37. I/O Banks for Intel Arria 10 GX 160 and GX 220 Devices

Product Line		GX 160			GX 220		
Package		U19	F27	F29	U19	F27	F29
I/O Bank	2A	48	48	48	48	48	48
	2J	48	48	48	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48
	3A	—	48	48	—	48	48
	3B	4	—	48	4	—	48
Total		196	240	288	196	240	288

Table 38. I/O Banks for Intel Arria 10 GX 270 and GX 320 Devices

Product Line		GX 270				GX 320			
Package		F27	F29	F34	F35	F27	F29	F34	F35
I/O Bank	2A	48	48	48	48	48	48	48	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	—	48	48	48	—	48	48	48

continued...

Product Line		GX 270				GX 320			
Package		F27	F29	F34	F35	F27	F29	F34	F35
	3C	—	48	48	48	—	48	48	48
	3D	—	24	48	48	—	24	48	48
Total		240	360	384	384	240	360	384	384

Table 39. I/O Banks for Intel Arria 10 GX 480 Devices

Product Line		GX 480		
Package		F29	F34	F35
I/O Bank	2A	48	48	48
	2I	—	12	12
	2J	48	48	48
	2K	48	48	48
	2L	48	48	48
	3A	48	48	48
	3B	48	48	48
	3C	48	48	48
	3D	24	48	48
	3E	—	48	—
	3F	—	48	—
Total		360	492	396

Table 40. I/O Banks for Intel Arria 10 GX 570 and GX 660 Devices

Product Line		GX 570				GX 660			
Package		F34	F35	NF40	KF40	F34	F35	NF40	KF40
I/O Bank	2A	48	48	48	48	48	48	48	48
	2G	—	—	—	24	—	—	—	24
	2H	—	—	—	48	—	—	—	48
	2I	12	12	12	48	12	12	12	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	48	48	48	48	48	48	48	48
	3C	48	48	48	48	48	48	48	48
	3D	48	48	48	48	48	48	48	48
	3E	48	—	48	48	48	—	48	48
	3F	48	—	48	48	48	—	48	48

continued...

Product Line		GX 570				GX 660			
Package		F34	F35	NF40	KF40	F34	F35	NF40	KF40
	3G	—	—	48	48	—	—	48	48
	3H	—	—	48	48	—	—	48	48
Total		492	396	588	696	492	396	588	696

Table 41. I/O Banks for Intel Arria 10 GX 900 Devices

Product Line		GX 900					
Package		F34	NF40	RF40	NF45	SF45	UF45
I/O Bank	2A	48	48	48	48	48	48
	2F	—	—	48	48	—	—
	2G	—	—	—	48	—	—
	2H	—	—	—	48	—	—
	2I	24	24	—	48	48	48
	2J	48	48	—	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48
	3A	48	48	28	48	48	48
	3B	48	48	27	48	48	48
	3C	48	48	—	48	48	48
	3D	48	48	—	48	48	48
	3E	48	48	—	48	48	48
	3F	48	48	—	48	48	—
	3G	—	48	47	48	48	—
	3H	—	48	48	48	48	—
Total		504	600	342	768	624	480

Table 42. I/O Banks for Intel Arria 10 GX 1150 Devices

Product Line		GX 1150					
Package		F34	NF40	RF40	NF45	SF45	UF45
I/O Bank	2A	48	48	48	48	48	48
	2F	—	—	48	48	—	—
	2G	—	—	—	48	—	—
	2H	—	—	—	48	—	—
	2I	24	24	—	48	48	48
	2J	48	48	—	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48

continued...

Product Line		GX 1150					
Package		F34	NF40	RF40	NF45	SF45	UF45
	3A	48	48	28	48	48	48
	3B	48	48	27	48	48	48
	3C	48	48	—	48	48	48
	3D	48	48	—	48	48	48
	3E	48	48	—	48	48	48
	3F	48	48	—	48	48	—
	3G	—	48	47	48	48	—
	3H	—	48	48	48	48	—
Total		504	600	342	768	624	480

Related Information

- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- FPGA I/O Resources in Intel Arria 10 GX Packages on page 110
- I/O Banks Groups in Intel Arria 10 Devices on page 113
- Guideline: Intel FPGA LVDS SERDES IP Core Instantiation on page 181

5.4.3.2. I/O Banks for Intel Arria 10 GT Devices

The following table lists the I/O banks available, the total number of I/O pins in each bank, and the total number of I/O pins for each product line and device package of the Intel Arria 10 GT device family variant.

Table 43. I/O Banks for Intel Arria 10 GT 900 and GT 1150 Devices

Product Line		GT 900	GT 1150
Package		SF45	SF45
I/O Bank	2A	48	48
	2I	48	48
	2J	48	48
	2K	48	48
	2L	48	48
	3A	48	48
	3B	48	48
	3C	48	48
	3D	48	48
	3E	48	48
	3F	48	48
	3G	48	48
	3H	48	48
Total		624	624

Related Information

- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- FPGA I/O Resources in Intel Arria 10 GT Packages on page 111
- I/O Banks Groups in Intel Arria 10 Devices on page 113
- Guideline: Intel FPGA LVDS SERDES IP Core Instantiation on page 181

5.4.3.3. I/O Banks for Intel Arria 10 SX Devices

The following tables list the I/O banks available, the total number of I/O pins in each bank, and the total number of I/O pins for each product line and device package of the Intel Arria 10 SX device family variant.

Table 44. I/O Banks for Intel Arria 10 SX 160 and SX 220 Devices

Product Line		SX 160			SX 220		
Package		U19	F27	F29	U19	F27	F29
I/O Bank	2A	48	48	48	48	48	48
	2J	48	48	48	48	48	48
	2K	48	48	48	48	48	48
	2L	48	48	48	48	48	48
	3A	—	48	48	—	48	48
	3B	4	—	48	4	—	48
Total		196	240	288	196	240	288

Table 45. I/O Banks for Intel Arria 10 SX 270 and SX 320 Devices

Product Line		SX 270				SX 320			
Package		F27	F29	F34	F35	F27	F29	F34	F35
I/O Bank	2A	48	48	48	48	48	48	48	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	—	48	48	48	—	48	48	48
	3C	—	48	48	48	—	48	48	48
	3D	—	24	48	48	—	24	48	48
Total		240	360	384	384	240	360	384	384

Table 46. I/O Banks for Intel Arria 10 SX 480 Devices

Product Line		SX 480		
Package		F29	F34	F35
I/O Bank	2A	48	48	48
	2I	—	12	12

continued...

Product Line		SX 480		
Package		F29	F34	F35
	2J	48	48	48
	2K	48	48	48
	2L	48	48	48
	3A	48	48	48
	3B	48	48	48
	3C	48	48	48
	3D	24	48	48
	3E	—	48	—
	3F	—	48	—
Total		360	492	396

Table 47. I/O Banks for Intel Arria 10 SX 570 and SX 660 Devices

Product Line		SX 570				SX 660			
Package		F34	F35	NF40	KF40	F34	F35	NF40	KF40
I/O Bank	2A	48	48	48	48	48	48	48	48
	2G	—	—	—	24	—	—	—	24
	2H	—	—	—	48	—	—	—	48
	2I	12	12	12	48	12	12	12	48
	2J	48	48	48	48	48	48	48	48
	2K	48	48	48	48	48	48	48	48
	2L	48	48	48	48	48	48	48	48
	3A	48	48	48	48	48	48	48	48
	3B	48	48	48	48	48	48	48	48
	3C	48	48	48	48	48	48	48	48
	3D	48	48	48	48	48	48	48	48
	3E	48	—	48	48	48	—	48	48
	3F	48	—	48	48	48	—	48	48
	3G	—	—	48	48	—	—	48	48
	3H	—	—	48	48	—	—	48	48
Total		492	396	588	696	492	396	588	696

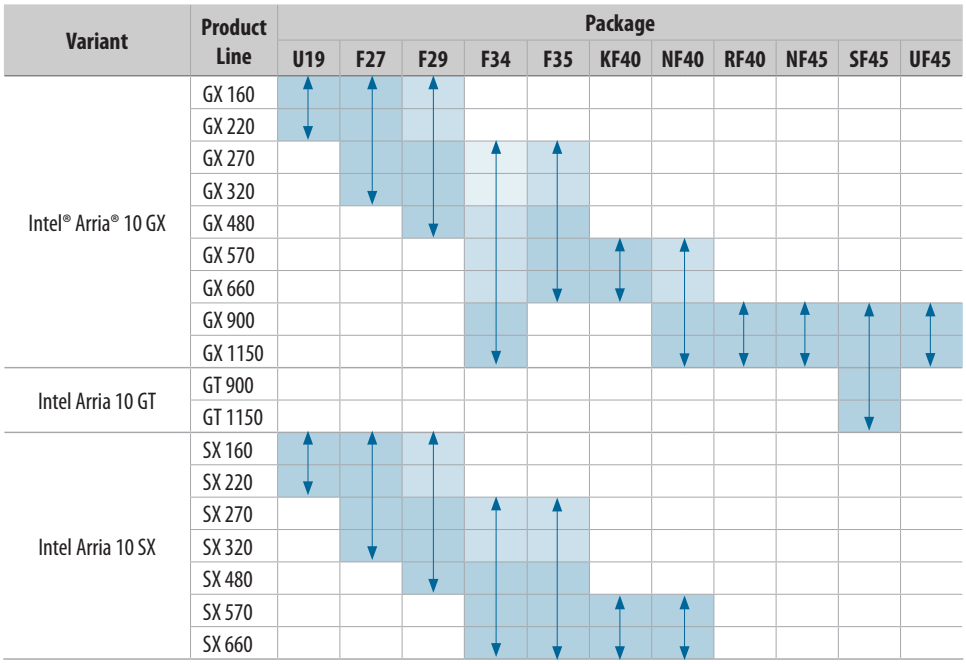
Related Information

- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- FPGA I/O Resources in Intel Arria 10 SX Packages on page 112
- I/O Banks Groups in Intel Arria 10 Devices on page 113
- Guideline: Intel FPGA LVDS SERDES IP Core Instantiation on page 181

5.4.4. I/O Vertical Migration for Intel Arria 10 Devices

Figure 79. Migration Capability Across Intel Arria 10 Product Lines

- The arrows indicate the migration paths. The devices included in each vertical migration path are shaded. Devices with fewer resources in the same path have lighter shades.
- To achieve the full I/O migration across product lines in the same migration path, restrict I/Os and transceivers usage to match the product line with the lowest I/O and transceiver counts.
- An LVDS I/O bank in the source device may be mapped to a 3 V I/O bank in the target device. To use memory interface clock frequency higher than 533 MHz, assign external memory interface pins only to banks that are LVDS I/O in both devices.
- There may be nominal 0.15 mm package height difference between some product lines in the same package type.
- Some migration paths are not shown in the Intel Quartus Prime software **Pin Migration View**.



Note: To verify the pin migration compatibility, use the **Pin Migration View** window in the Intel Quartus Prime software Pin Planner.

Related Information

- Verifying Pin Migration Compatibility on page 119
- Migrating Assignments to Another Target Device, *Intel Quartus Prime Standard Edition Handbook Volume 2: Design Implementation and Optimization*
  Provides more information about vertical I/O migrations.

5.4.4.1. Verifying Pin Migration Compatibility

You can use the **Pin Migration View** window in the Intel Quartus Prime software Pin Planner to assist you in verifying whether your pin assignments migrate to a different device successfully. You can vertically migrate to a device with a different density while using the same device package, or migrate between packages with different densities and ball counts.

1. Open **Assignments ➤ Pin Planner** and create pin assignments.
2. If necessary, perform one of the following options to populate the Pin Planner with the node names in the design:
   — Analysis & Elaboration
   — Analysis & Synthesis
   — Fully compile the design
3. Then, on the menu, click **View ➤ Pin Migration View**.
4. To select or change migration devices:
   a. Click **Device** to open the **Device** dialog box.
   b. Under **Migration compatibility** click **Migration Devices**.
5. To show more information about the pins:
   a. Right-click anywhere in the **Pin Migration View** window and select **Show Columns**.
   b. Then, click the pin feature you want to display.
6. If you want to view only the pins, in at least one migration device, that have a different feature than the corresponding pin in the migration result, turn on **Show migration differences**.
7. Click **Pin Finder** to open the **Pin Finder** dialog box to find and highlight pins with specific functionality.

   If you want to view only the pins highlighted by the most recent query in the **Pin Finder** dialog box, turn on **Show only highlighted pins**.
8. To export the pin migration information to a Comma-Separated Value file (**.csv**), click **Export**.

Related Information

- I/O Vertical Migration for Intel Arria 10 Devices on page 119
- Migrating Assignments to Another Target Device, *Intel Quartus Prime Standard Edition Handbook Volume 2: Design Implementation and Optimization*
  Provides more information about vertical I/O migrations.

5.5. Architecture and General Features of I/Os in Intel Arria 10 Devices

5.5.1. I/O Element Structure in Intel Arria 10 Devices

The I/O elements (IOEs) in Intel Arria 10 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 columns within the core fabric of the Intel Arria 10 device.

Intel Arria 10 SX devices also have IOEs for the HPS.

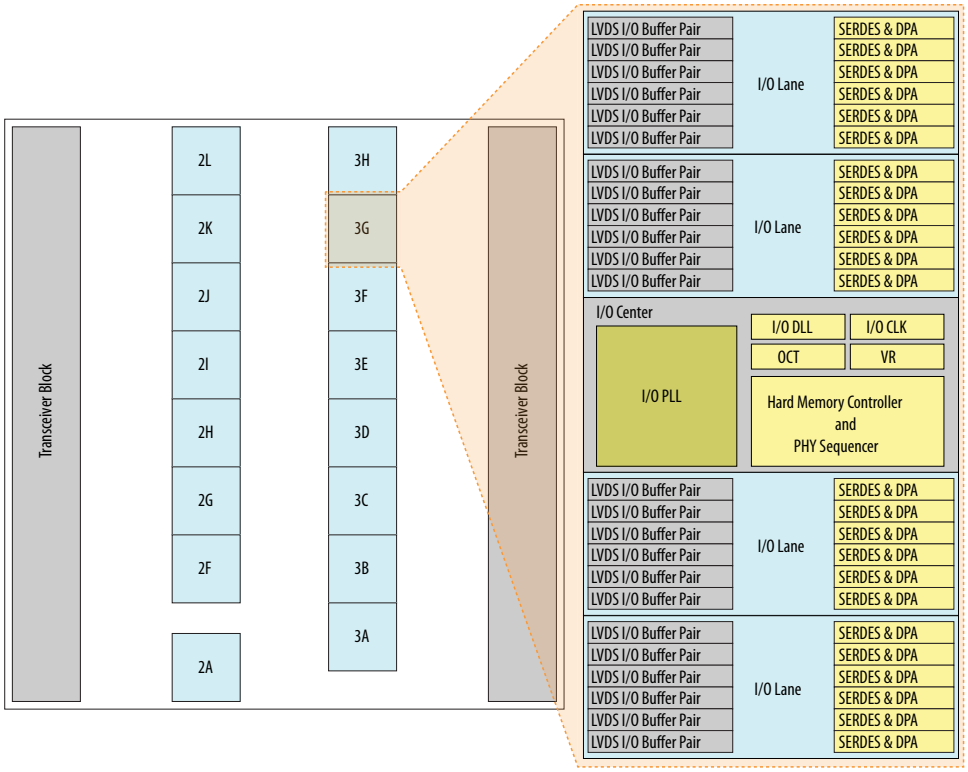
The GPIO IOE register consists of the DDR register, the half rate register, and the transmitter delay chains for input, output, and output enable (OE) paths:

- You can take data from the combinatorial path or the registered path.
- Only the core clock clocks the data.
- The half rate clock routed from the core clocks the half rate register.
- The full rate clock from the core clocks the full rate register.

5.5.1.1. I/O Bank Architecture in Intel Arria 10 Devices

In each I/O bank, there are four I/O lanes with 12 I/O pins in each lane. Other than the I/O lanes, each I/O bank also contains dedicated circuitries including the I/O PLL, DPA block, SERDES, hard memory controller, and I/O sequencer.

Figure 80. I/O Bank Structure



Related Information

Guideline: VREF Sources and VREF Pins on page 177
Describes VREF restrictions related to the I/O lanes.

5.5.1.2. I/O Buffer and Registers in Intel Arria 10 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. Use the Intel FPGA GPIO Intel FPGA IP to utilize these registers to implement DDR circuitry.

The input and output paths contain the following blocks:

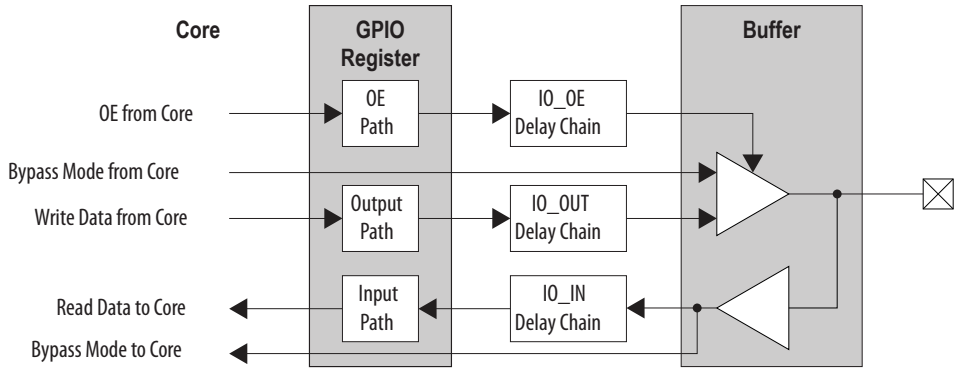
- Input registers—support half or full rate data transfer from peripheral to core, and support double or single data rate data capture from I/O buffer.
- Output registers—support half or full rate data transfer from core to peripheral, and support double or single data rate data transfer to I/O buffer.
- OE registers—support half or full rate data transfer from core to peripheral, and support single data rate data transfer to I/O buffer.

The input and output paths also support the following features:

- Clock enable.
- Asynchronous or synchronous reset.
- Bypass mode for input and output paths.
- Delays chains on input and output paths.

Figure 81. IOE Structure for Intel Arria 10 Devices

This figure shows the Intel Arria 10 FPGA IOE structure.



5.5.2. Features of I/O Pins in Intel Arria 10 Devices

Open-Drain Output on page 122
Bus-Hold Circuitry on page 123
Weak Pull-up Resistor on page 123

5.5.2.1. 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.

Use an external resistor to pull the signal to a logic high.

Note: Do not pull the output voltage higher than the Vi (DC) level. Intel recommends that you perform HSPICE simulation to verify the output voltage in your selected topology. You must ensure the output voltage meets the V_{IH} and V_{IL} requirements of the receiving device.

5.5.2.2. 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 kΩ, 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.

5.5.2.3. Weak Pull-up Resistor

Each I/O pin provides an optional programmable pull-up resistor during user mode. The pull-up resistor, typically 25 kΩ, weakly holds the I/O to the V_{CCIO} level.

The Intel Arria 10 device supports programmable weak pull-up resistors only on user I/O pins but not on dedicated configuration pins, dedicated clock pins, or JTAG pins .

If you enable this option, you cannot use the bus-hold feature.

5.5.3. Programmable IOE Features in Intel Arria 10 Devices

Table 48. Intel Arria 10 Programmable IOE Features Settings and Assignment Name

Feature	Setting	Condition	Intel Quartus Prime Assignment Name
Slew Rate Control	0 (Slow), 1 (Fast). Default is 1.	Disabled if you use the R_S OCT feature.	SLEW_RATE
I/O Delay	Refer to the device datasheet	—	INPUT_DELAY_CHAIN OUTPUT_DELAY_CHAIN
Open-Drain Output	On, Off. Default is Off	—	AUTO_OPEN_DRAIN_PINS
Bus-Hold	On, Off. Default is Off.	Disabled if you use the weak pull-up resistor feature.	ENABLE_BUS_HOLD_CIRCUITRY

continued...

Feature	Setting	Condition	Intel Quartus Prime Assignment Name
Weak Pull-up Resistor	On, Off. Default is Off.	Disabled if you use the bus-hold feature.	WEAK_PULL_UP_RESISTOR
Pre-Emphasis	0 (disabled), 1 (enabled). Default is 1.	—	PROGRAMMABLE_PREEMPHASIS
Differential Output Voltage	0 (low), 1 (medium low), 2 (medium high), 3 (high). Default is 2.	—	PROGRAMMABLE_VOD

Table 49. Intel Arria 10 Programmable IOE Features I/O Buffer Types and I/O Standards Support

This table lists the I/O buffer types and I/O standards that support the programmable IOE features. For information about which I/O standards are available for each I/O buffer type, refer to the related information.

Feature	I/O Buffer Type Support			I/O Standards Support
	LVDS I/O	3 V I/O	HPS I/O (SoC Devices Only)	
Slew Rate Control	Yes	Yes	Yes	• 3.0 V LVTTL • 1.2 V, 1.5 V, 1.8 V, and 3.0 V LVCMOS • SSTL-18, SSTL-15, SSTL-135, SSTL-125, and SSTL-12 • 1.2 V, 1.5 V, and 1.8 V HSTL • HSUL-12 • POD12 • Differential SSTL-18, Differential SSTL-15, Differential SSTL-135, Differential SSTL-125, and Differential SSTL-12 • Differential 1.2 V, 1.5 V, and 1.8 V HSTL • Differential HSUL-12
I/O Delay	Yes	Yes	—	
Open-Drain Output	Yes	Yes	Yes	• 3.0 V LVTTL • 1.2 V, 1.5 V, 1.8 V, and 3.0 V LVCMOS
Bus-Hold	Yes	Yes	Yes	
Weak Pull-up Resistor	Yes	Yes	Yes	
Pre-Emphasis	Yes	—	—	• LVDS • RSDS • Mini-LVDS • LVPECL • Differential POD12
Differential Output Voltage	Yes	—	—	• LVDS • RSDS • Mini-LVDS • LVPECL

Related Information

- Programmable IOE Delay, *Intel Arria 10 Device Datasheet*
- Programmable Current Strength on page 125
- Programmable Output Slew Rate Control on page 126
- Programmable IOE Delay on page 127

- Programmable Open-Drain Output on page 127
- Programmable Pre-Emphasis on page 127
- Programmable Differential Output Voltage on page 128
- I/O Standards Support for FPGA I/O in Intel Arria 10 Devices on page 101
  Lists the I/O standards supported by the LVDS I/O and 3 V I/O buffers.
- I/O Standards Support for HPS I/O in Intel Arria 10 Devices on page 102
  Lists the I/O standards supported by the HPS I/O buffers.

5.5.3.1. 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.

Note: To use programmable current strength, you must specify the current strength assignment in the Intel Quartus Prime software. Without explicit assignments, the Intel Quartus Prime software uses these predefined default values:

- All HSTL and SSTL Class I, and all non-voltage-referenced I/O standards—50 Ω R_S OCT without calibration
- All HSTL and SSTL Class II I/O standards—25 Ω R_S OCT without calibration
- POD12 I/O standard—34 Ω R_S OCT without calibration

Table 50. Programmable Current Strength Settings for Intel Arria 10 Devices

The output buffer for each Intel Arria 10 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) (8)	Supported in HPS (SoC Devices Only) (9)
3.0 V LVTTL	24, 20, **16**, 12, 8, 4	24, 20, **16**, 12, 8, 4
3.0 V CMOS	16, **12**, 8, 4	16, **12**, 8, 4
2.5 V LVCMOS	16, **12**, 8, 4	16, **12**, 8, 4
1.8 V LVCMOS	**12**, 10, 8, 6, 4, 2	**12**, 10, 8, 6, 4, 2
1.5 V LVCMOS	**12**, 10, 8, 6, 4, 2	**12**, 10, 8, 6, 4, 2
1.2 V LVCMOS	**8**, 6, 4, 2	—
SSTL-18 Class I	12, 10, **8**, 6, 4	12, 10, **8**, 6, 4
SSTL-18 Class II	**16**	8, **16**
SSTL-15 Class I	12, 10, **8**, 6, 4	12, 10, **8**, 6, 4
SSTL-15 Class II	**16**	8, **16**
SSTL-135 Class I	12, 10, **8**, 6, 4	—
SSTL-135 Class II	**16**	—

continued...

(8) For I/O standards with DDR3 OCT settings, refer to On-Chip I/O Termination in Intel Arria 10 Devices on page 129.

(9) The programmable current strength information for the HPS is preliminary.

I/O Standard	I_{OH} / I_{OL} Current Strength Setting (mA) (Default setting in bold) (8)	Supported in HPS (SoC Devices Only) (9)
SSTL-125 Class I	12, 10, **8**, 6, 4	—
SSTL-125 Class II	**16**	—
SSTL-12 Class I	12, 10, **8**, 6, 4	—
SSTL-12 Class II	**16**	—
POD12	16, 12, 10, **8**, 6, 4	—
1.8 V HSTL Class I	12, 10, **8**, 6, 4	12, 10, **8**, 6, 4
1.8 V HSTL Class II	**16**	**16**
1.5 V HSTL Class I	12, 10, **8**, 6, 4	12, 10, **8**, 6, 4
1.5 V HSTL Class II	**16**	**16**
1.2 V HSTL Class I	12, 10, **8**, 6, 4	—
1.2 V HSTL Class II	**16**	—
Differential SSTL-135 Class I	12, 10, **8**, 6, 4	—
Differential SSTL-135 Class II	**16**	—
Differential SSTL-125 Class I	12, 10, **8**, 6, 4	—
Differential SSTL-125 Class II	**16**	—
Differential SSTL-12 Class I	12, 10, **8**, 6, 4	—
Differential SSTL-12 Class II	**16**	—
Differential POD12	16, 12, 10, **8**, 6, 4	—

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

5.5.3.2. Programmable Output Slew Rate Control

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

(8) For I/O standards with DDR3 OCT settings, refer to On-Chip I/O Termination in Intel Arria 10 Devices on page 129.

(9) The programmable current strength information for the HPS is preliminary.

5.5.3.3. Programmable IOE Delay

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

To ensure that the signals within a bus have the same delay going into or out of the device, each pin can have different delay values:

- Delay from input pin to input register
- Delay from output pin to output register
- In the output and 0E paths, there are the output and 0E delays which have 50 ps incremental delays and a maximum delay of 800 ps.
- In the input paths, there are two input delay chains with incremental delays of 50 ps and a maximum delay of 3.2 ns.

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

Related Information

Programmable IOE Delay, *Intel Arria 10 Device Datasheet*

5.5.3.4. Programmable Open-Drain Output

The programmable open-drain output provides a high-impedance state on output when logic to the output buffer is high. If logic to the output buffer is low, output is low.

You can attach several open-drain outputs 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 of these methods:

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

You can design open-drain output without enabling the option assignment. However, your design will not use the I/O buffer's open-drain output feature. The open-drain output feature of the I/O buffer provides you the best propagation delay from OE to output.

5.5.3.5. 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 82. Programmable Pre-Emphasis

This figure shows the LVDS output with pre-emphasis.

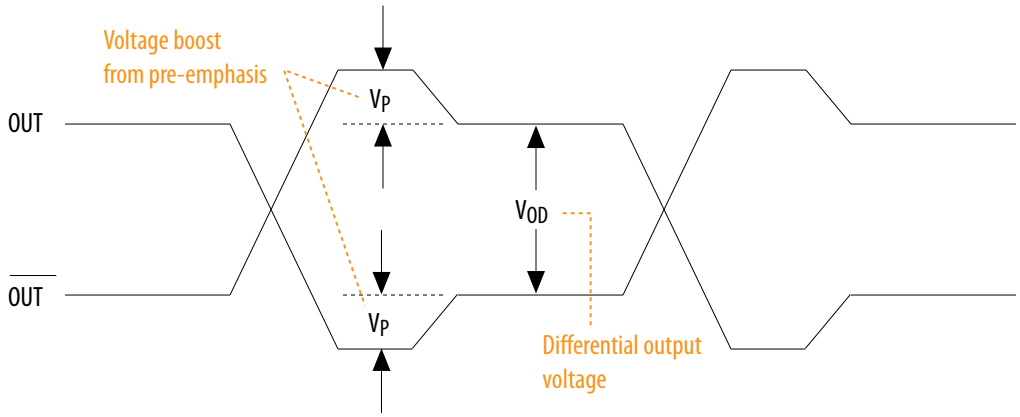


Table 51. Intel Quartus Prime Software Assignment Editor—Programmable Pre-Emphasis

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.

5.5.3.6. 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 83. Differential V_{OD}

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

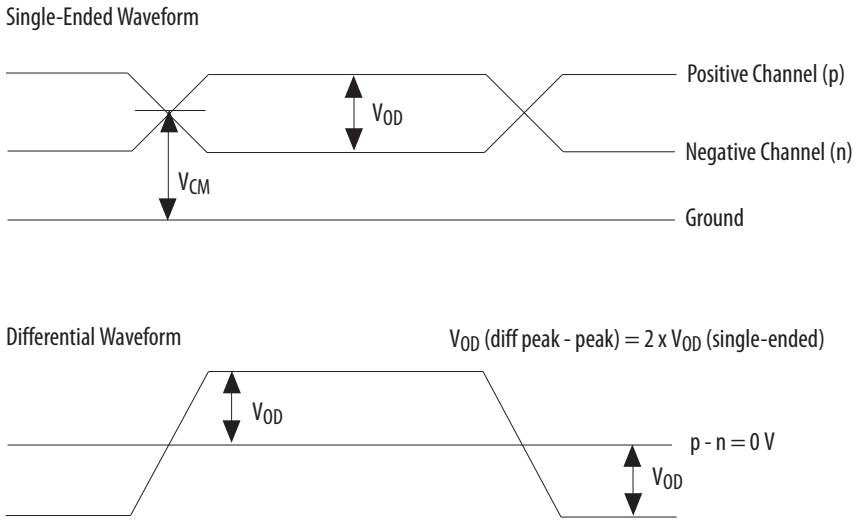


Table 52. Intel Quartus Prime Software Assignment Editor—Programmable V_{OD}

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 2.

5.5.4. On-Chip I/O Termination in Intel Arria 10 Devices

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

The Intel Arria 10 devices support OCT in all FPGA and HPS I/O banks. For the 3 V and HPS I/Os, the I/Os support only OCT without calibration.

Figure 84. Single-ended Termination (R_S and R_T)

This figure shows the single-ended termination schemes supported in Intel Arria 10 devices. R_{T1} and R_{T2} are dynamic parallel terminations and are enabled only if the device is receiving. In bidirectional applications, R_{T1} and R_{T2} are automatically switched on when the device is receiving and switched off when the device is driving.

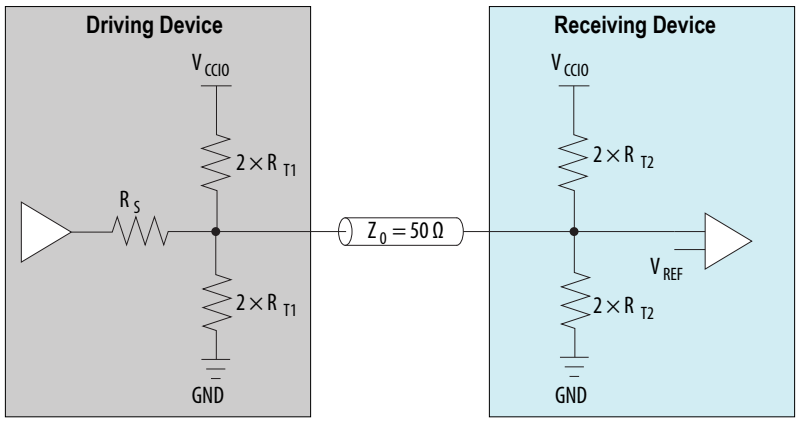


Table 53. OCT Schemes Supported in Intel Arria 10 Devices

Direction	OCT Schemes	I/O Type Support		
		LVDS I/O	3 V I/O	HPS I/O
Output	R_S OCT with calibration	Yes	—	—
	R_S OCT without calibration	Yes	Yes	Yes
Input	R_T OCT with calibration	Yes	—	—
	R_D OCT (LVDS I/O standard only)	Yes	—	—
Bidirectional	Dynamic R_S and R_T OCT	Yes	—	—

Related Information

- *Intel FPGA OCT IP Core User Guide*
- RS OCT without Calibration in Intel Arria 10 Devices on page 130
- RS OCT with Calibration in Intel Arria 10 Devices on page 132
- RT OCT with Calibration in Intel Arria 10 Devices on page 134
- Dynamic OCT on page 136
- Differential Input RD OCT on page 137
- OCT Calibration Block in Intel Arria 10 Devices on page 138

5.5.4.1. R_S OCT without Calibration in Intel Arria 10 Devices

The Intel Arria 10 devices support R_S OCT for single-ended and voltage-referenced I/O standards. R_S OCT without calibration is supported on output only.

Table 54. 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	Device Variant Support	Uncalibrated OCT (Output)
		R_S (Ω)
3.0 V LVTTL/3.0 V LVCMOS	GX, SX	25/50
2.5 V LVCMOS	GX, SX	25/50
1.8 V LVCMOS	All	25/50
1.5 V LVCMOS	All	25/50
1.2 V LVCMOS	All	25/50
SSTL-18 Class I	All	50
SSTL-18 Class II	All	25
SSTL-15 Class I	All	50
SSTL-15 Class II	All	25
SSTL-15	All	34, 40
SSTL-135	All	34, 40
SSTL-125	All	34, 40
SSTL-12	All	40, 60, 120, 240
POD12	All	34, 40, 48, 60
1.8 V HSTL Class I	All	50
1.8 V HSTL Class II	All	25
1.5 V HSTL Class I	All	50
1.5 V HSTL Class II	All	25
1.2 V HSTL Class I	All	50
1.2 V HSTL Class II	All	25
HSUL-12	All	34.3, 40, 48, 60, 80
Differential SSTL-18 Class I	All	50
Differential SSTL-18 Class II	All	25
Differential SSTL-15 Class I	All	50
Differential SSTL-15 Class II	All	25
Differential SSTL-15	All	34, 40
Differential SSTL-135	All	34, 40
Differential SSTL-125	All	34, 40
Differential SSTL-12	All	40, 60, 120, 240
Differential POD12	All	34, 40, 48, 60
Differential 1.8 V HSTL Class I	All	50
Differential 1.8 V HSTL Class II	All	25
Differential 1.5 V HSTL Class I	All	50

continued...

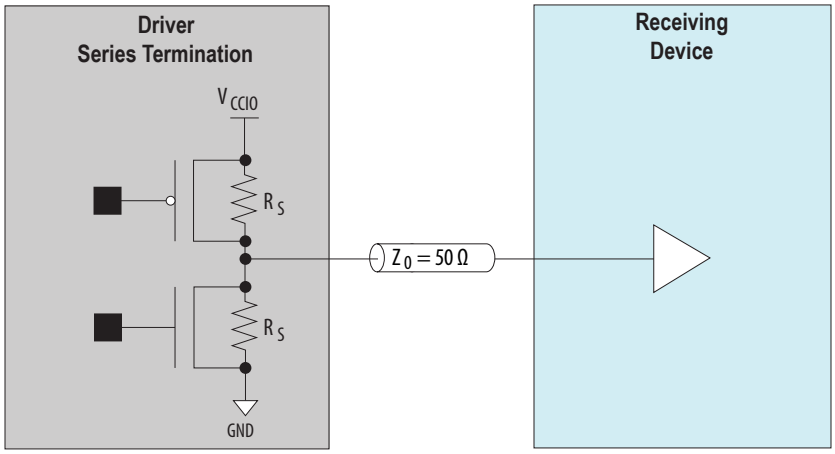
I/O Standard	Device Variant Support	Uncalibrated OCT (Output)
		R_S (Ω)
Differential 1.5 V HSTL Class II	All	25
Differential 1.2 V HSTL Class I	All	50
Differential 1.2 V HSTL Class II	All	25
Differential HSUL-12	All	34.3, 40, 48, 60, 80

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 85. R_S OCT Without Calibration

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



Related Information

On-Chip I/O Termination in Intel Arria 10 Devices on page 129

5.5.4.2. R_S OCT with Calibration in Intel Arria 10 Devices

The Intel Arria 10 devices support R_S OCT with calibration in all LVDS I/O banks.

Table 55. 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	Device Variant Support	Calibrated OCT (Output)	
		R_S (Ω)	RZQ (Ω)
1.8 V LVCMOS	All	25, 50	100
1.5 V LVCMOS	All	25, 50	100
1.2 V LVCMOS	All	25, 50	100
SSTL-18 Class I	All	50	100

continued...

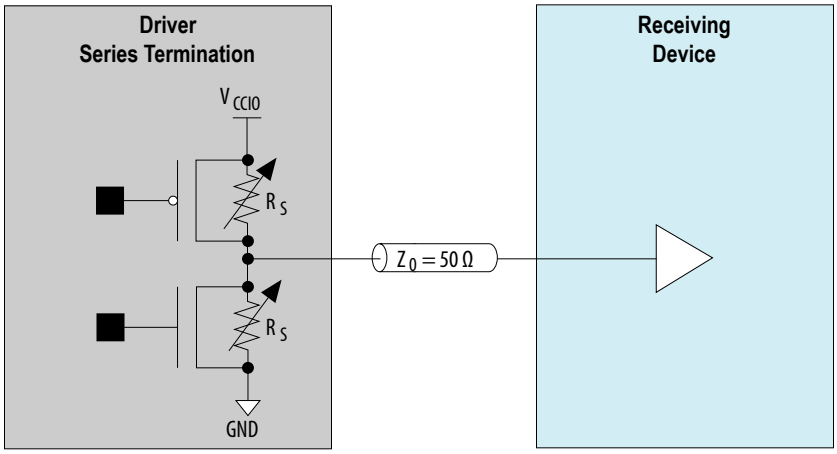
I/O Standard	Device Variant Support	Calibrated OCT (Output)	
		R_S (Ω)	RZQ (Ω)
SSTL-18 Class II	All	25	100
SSTL-15 Class I	All	50	100
SSTL-15 Class II	All	25	100
SSTL-15	All	25, 50	100
		34, 40	240
SSTL-135	All	34, 40	240
SSTL-125	All	34, 40	240
SSTL-12	All	40, 60, 120, 240	240
POD12	All	34, 40, 48, 60	240
1.8 V HSTL Class I	All	50	100
1.8 V HSTL Class II	All	25	100
1.5 V HSTL Class I	All	50	100
1.5 V HSTL Class II	All	25	100
1.2 V HSTL Class I	All	50	100
1.2 V HSTL Class II	All	25	100
HSUL-12	All	34, 40, 48, 60, 80	240
Differential SSTL-18 Class I	All	50	100
Differential SSTL-18 Class II	All	25	100
Differential SSTL-15 Class I	All	50	100
Differential SSTL-15 Class II	All	25	100
Differential SSTL-15	All	25, 50	100
		34, 40	240
Differential SSTL-135	All	34, 40	240
Differential SSTL-125	All	34, 40	240
Differential SSTL-12	All	40, 60, 120, 240	240
Differential POD12	All	34, 40, 48, 60	240
Differential 1.8 V HSTL Class I	All	50	100
Differential 1.8 V HSTL Class II	All	25	100
Differential 1.5 V HSTL Class I	All	50	100
Differential 1.5 V HSTL Class II	All	25	100
Differential 1.2 V HSTL Class I	All	50	100
Differential 1.2 V HSTL Class II	All	25	100
Differential HSUL-12	All	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 86. R_S OCT with Calibration

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



Related Information

On-Chip I/O Termination in Intel Arria 10 Devices on page 129

5.5.4.3. R_T OCT with Calibration in Intel Arria 10 Devices

The Intel Arria 10 devices support R_T OCT with calibration in all LVDS I/O banks but not in the 3 V I/O 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_{CCIO} of the bank must match the I/O standard of the pin where you enable the R_T OCT.

Table 56. 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	Device Variant Support	Calibrated OCT (Input)	
		R_T (Ω)	RZQ (Ω)
SSTL-18 Class I	All	50	100
SSTL-18 Class II	All	50	100
SSTL-15 Class I	All	50	100
SSTL-15 Class II	All	50	100
SSTL-15	All	30, 40, 60,120	240
SSTL-135	All	30, 40, 60, 120	240
SSTL-125	All	30, 40, 60, 120	240
SSTL-12	All	60, 120	240
POD12	All	34, 40, 48, 60, 80, 120, 240	240
1.8 V HSTL Class I	All	50	100

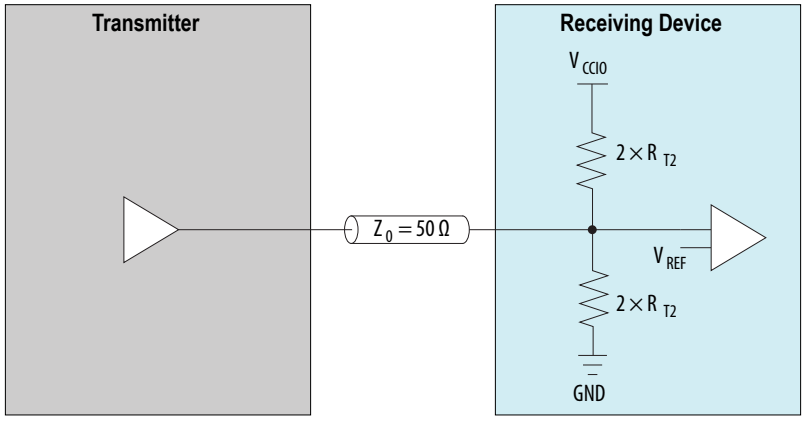
continued...

I/O Standard	Device Variant Support	Calibrated OCT (Input)	
		R_T (Ω)	RZQ (Ω)
1.8 V HSTL Class II	All	50	100
1.5 V HSTL Class I	All	50	100
1.5 V HSTL Class II	All	50	100
1.2 V HSTL Class I	All	50	100
1.2 V HSTL Class II	All	50	100
Differential SSTL-18 Class I	All	50	100
Differential SSTL-18 Class II	All	50	100
Differential SSTL-15 Class I	All	50	100
Differential SSTL-15 Class II	All	50	100
Differential SSTL-15	All	30, 40, 60,120	240
Differential SSTL-135	All	30, 40, 60, 120	240
Differential SSTL-125	All	30, 40, 60, 120	240
Differential SSTL-12	All	60, 120	240
Differential POD12	All	34, 40, 48, 60, 80, 120, 240	240
Differential 1.8 V HSTL Class I	All	50	100
Differential 1.8 V HSTL Class II	All	50	100
Differential 1.5 V HSTL Class I	All	50	100
Differential 1.5 V HSTL Class II	All	50	100
Differential 1.2 V HSTL Class I	All	50	100
Differential 1.2 V HSTL Class II	All	50	100

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 87. R_T OCT with Calibration



Related Information

On-Chip I/O Termination in Intel Arria 10 Devices on page 129

5.5.4.4. Dynamic OCT

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. Internal termination switches on only during input operation and thus draws less static power.

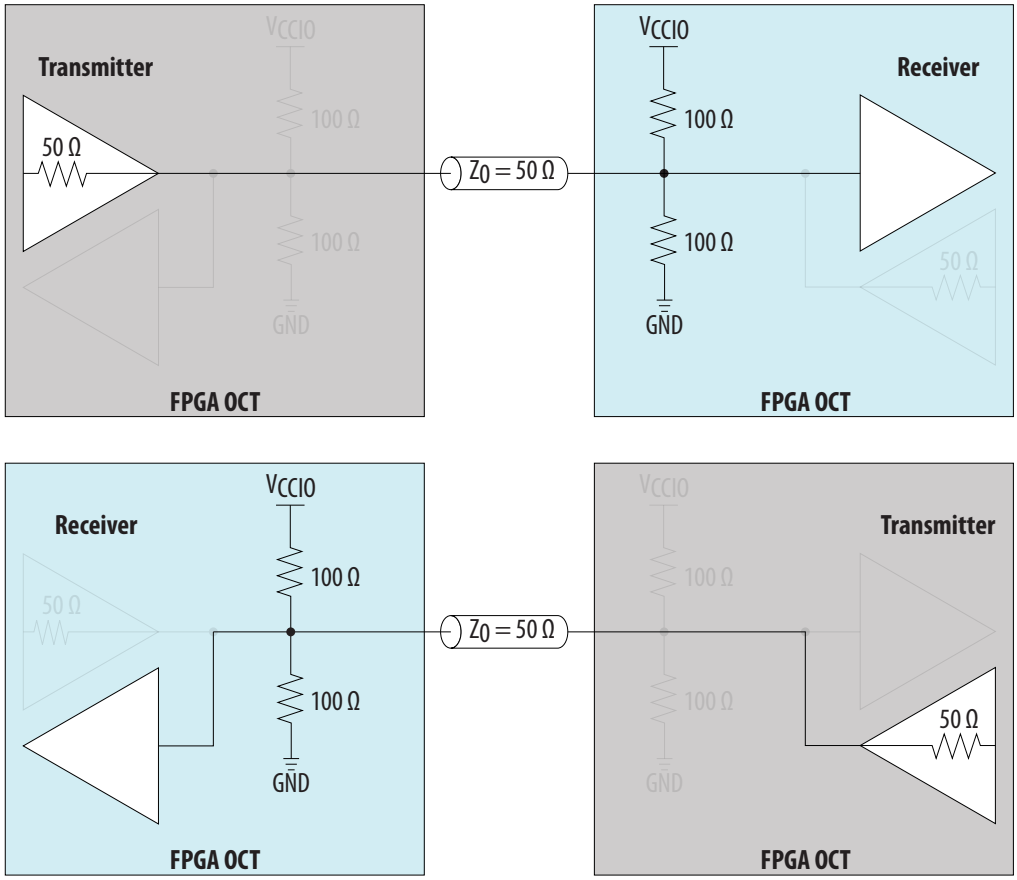
Note: If you use the SSTL-15, SSTL-135, and SSTL-125 I/O standards with the DDR3 memory interface, Intel recommends that you use OCT with these I/O standards to save board space and cost. OCT reduces the number of external termination resistors used.

Table 57. 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 88. Dynamic R_T OCT in Intel Arria 10 Devices



Related Information

On-Chip I/O Termination in Intel Arria 10 Devices on page 129

5.5.4.5. Differential Input R_D OCT

All I/O pins and dedicated clock input pins in Intel Arria 10 devices support on-chip differential termination, R_D OCT. The Intel Arria 10 devices provide a 100 Ω, on-chip differential termination option on each differential receiver channel for LVDS standards.

You can enable on-chip termination in the Intel Quartus Prime software Assignment Editor.

Figure 89. On-Chip Differential I/O Termination

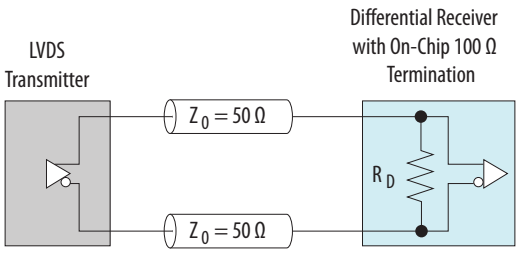


Table 58. Intel Quartus Prime Software Assignment Editor—On-Chip Differential Termination

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
Value	Differential

Related Information

On-Chip I/O Termination in Intel Arria 10 Devices on page 129

5.5.4.6. OCT Calibration Block in Intel Arria 10 Devices

You can calibrate the OCT using the OCT calibration block in any I/O bank in the same I/O column. The I/O bank that contains the OCT calibration block must have the same V_{CCIO} as the I/O bank of the OCT.

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:

- Each OCT calibration block has an external 240 Ω reference resistor associated with it through the RZQ pin.
- 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.

Intel Arria 10 devices support calibrated R_S and calibrated R_T OCT on all LVDS I/O pins except for dedicated configuration pins.

Related Information

- *Intel FPGA OCT IP Core User Guide*
- On-Chip I/O Termination in Intel Arria 10 Devices on page 129

5.5.5. External I/O Termination for Intel Arria 10 Devices

Table 59. External Termination Schemes for Different I/O Standards

I/O Standard	External Termination Scheme
2.5 V LVCMOS	No external termination required
1.8 V LVCMOS	
1.5 V LVCMOS	
1.2 V LVCMOS	
SSTL-18 Class I	Single-Ended SSTL I/O Standard Termination
SSTL-18 Class II	
SSTL-15 Class I	
SSTL-15 Class II	
SSTL-15 (10)	No external termination required
SSTL-135 (10)	
SSTL-125 (10)	
SSTL-12(10)	
POD12	Single-Ended POD I/O Standard Termination
Differential SSTL-18 Class I	Differential SSTL I/O Standard Termination
Differential SSTL-18 Class II	
Differential SSTL-15 Class I	
Differential SSTL-15 Class II	
Differential SSTL-15 (10)	No external termination required
Differential SSTL-135 (10)	
Differential SSTL-125 (10)	
Differential SSTL-12(10)	
Differential POD12	Differential POD I/O Standard Termination
1.8 V HSTL Class I	Single-Ended HSTL I/O Standard Termination
1.8 V HSTL Class II	
1.5 V HSTL Class I	
1.5 V HSTL Class II	
1.2 V HSTL Class I	
1.2 V HSTL Class II	
HSUL-12	No external termination required
Differential 1.8 V HSTL Class I	Differential HSTL I/O Standard Termination
Differential 1.8 V HSTL Class II	

continued...

(10) 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.

I/O Standard	External Termination Scheme
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	
Differential HSUL-12	No external termination required
LVDS	LVDS I/O Standard Termination
RSDS	RSDS/mini-LVDS I/O Standard Termination
Mini-LVDS	
LVPECL	Differential LVPECL I/O Standard Termination

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

5.5.5.1. 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.

Figure 90. SSTL I/O Standard Termination

This figure shows the details of SSTL I/O termination on Intel Arria 10 devices.

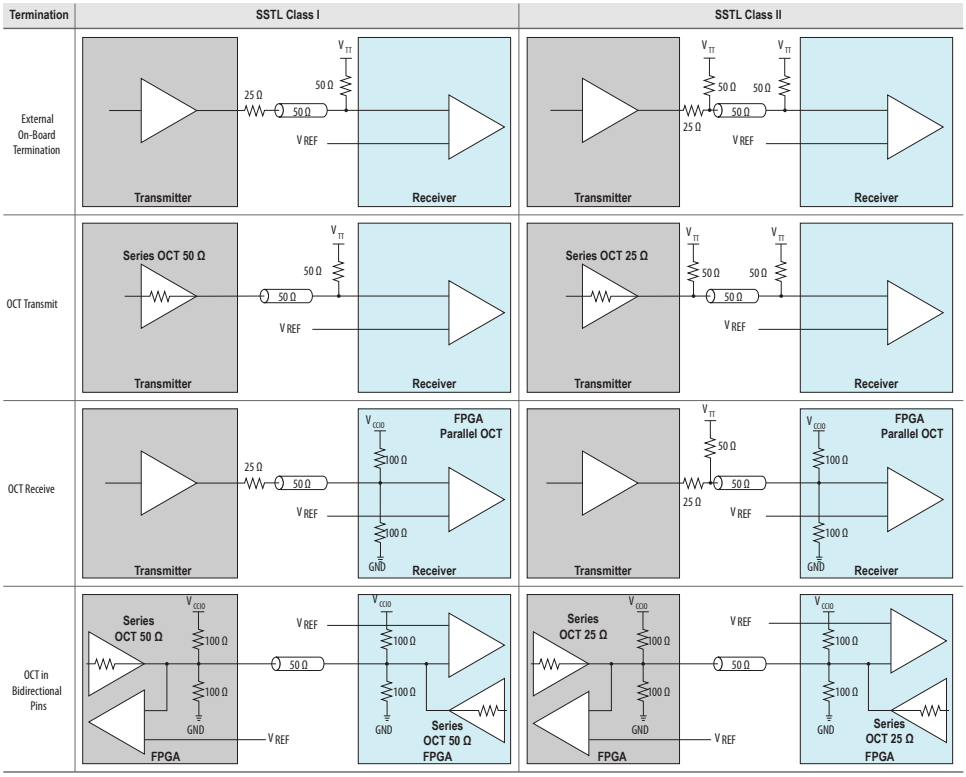


Figure 91. HSTL I/O Standard Termination

This figure shows the details of HSTL I/O termination on the Intel Arria 10 devices.

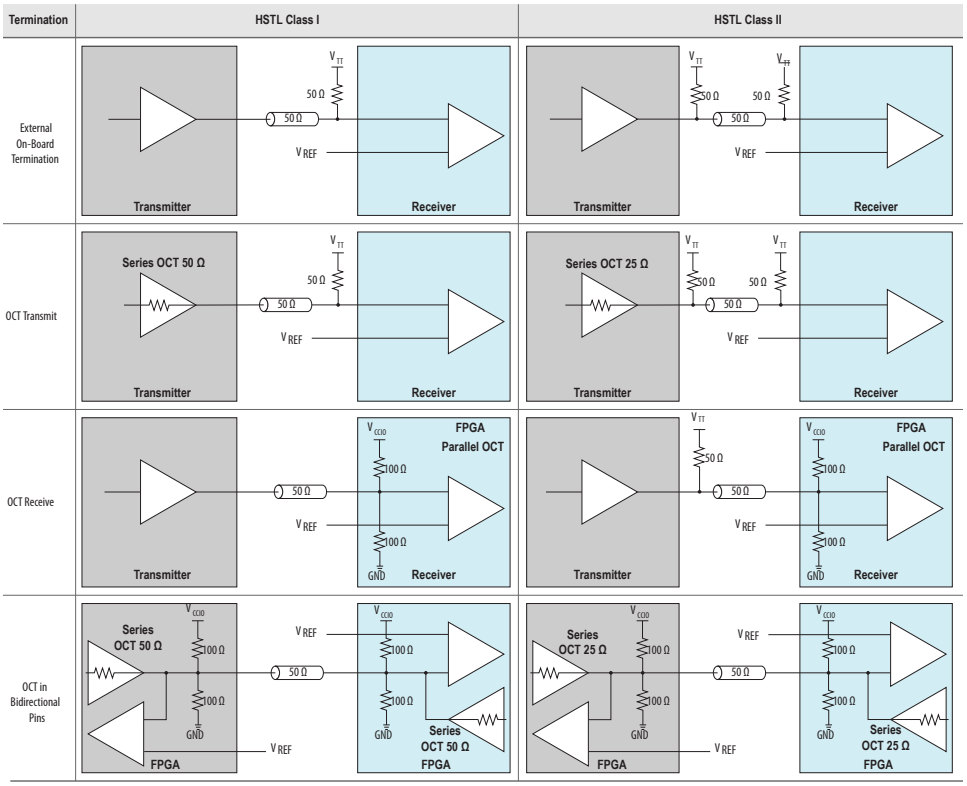
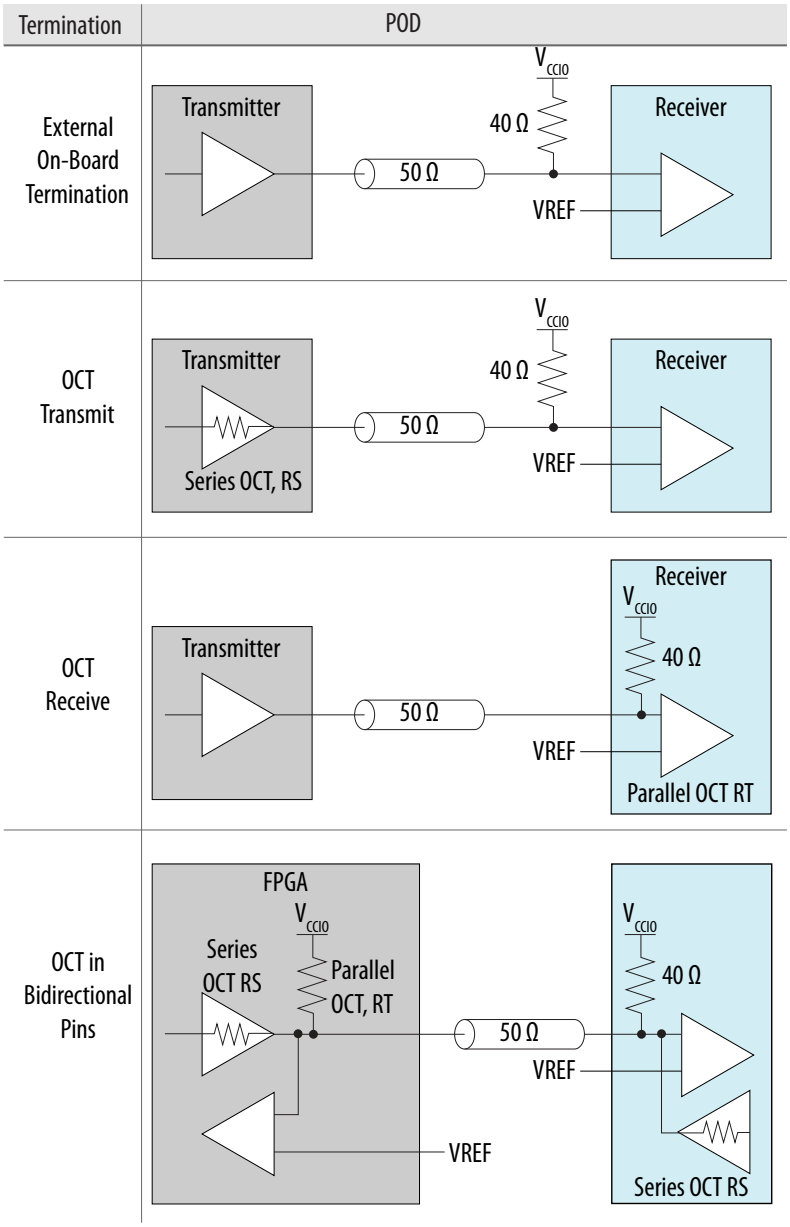


Figure 92. POD I/O Standard Termination

This figure shows the details of POD I/O termination on the Intel Arria 10 devices.



Related Information

Dynamic OCT on page 136

5.5.5.2. Differential I/O Termination for Intel Arria 10 Devices

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.

Related Information

- Differential HSTL, SSTL, HSUL, and POD Termination on page 144
- LVDS, RSDS, and Mini-LVDS Termination on page 146
- LVPECL Termination on page 146

5.5.5.2.1. Differential HSTL, SSTL, HSUL, and POD Termination

Differential HSTL, SSTL, HSUL, and POD inputs use LVDS differential input buffers. However, R_D support is only available if the I/O standard is LVDS.

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

Figure 93. Differential SSTL I/O Standard Termination

This figure shows the details of Differential SSTL I/O termination on Intel Arria 10 devices.

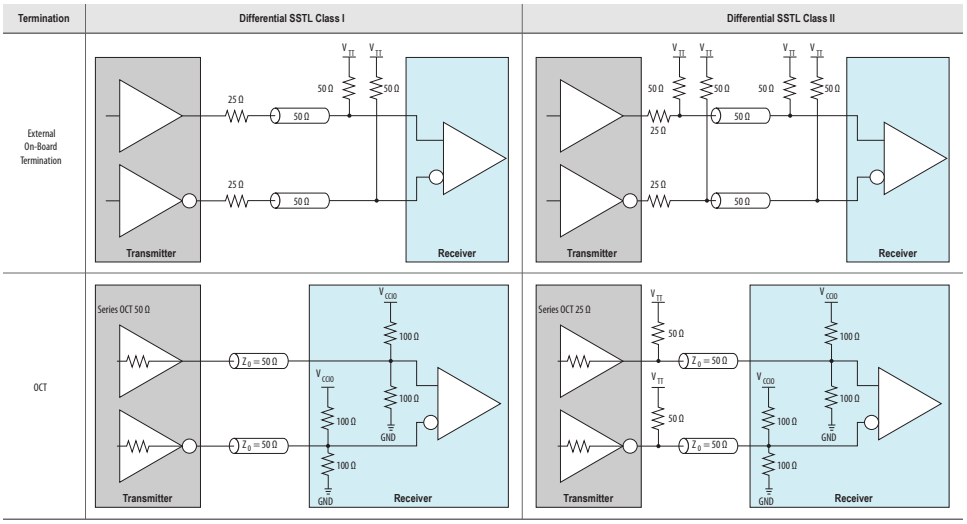


Figure 94. Differential HSTL I/O Standard Termination

This figure shows the details of Differential HSTL I/O standard termination on Intel Arria 10 devices.

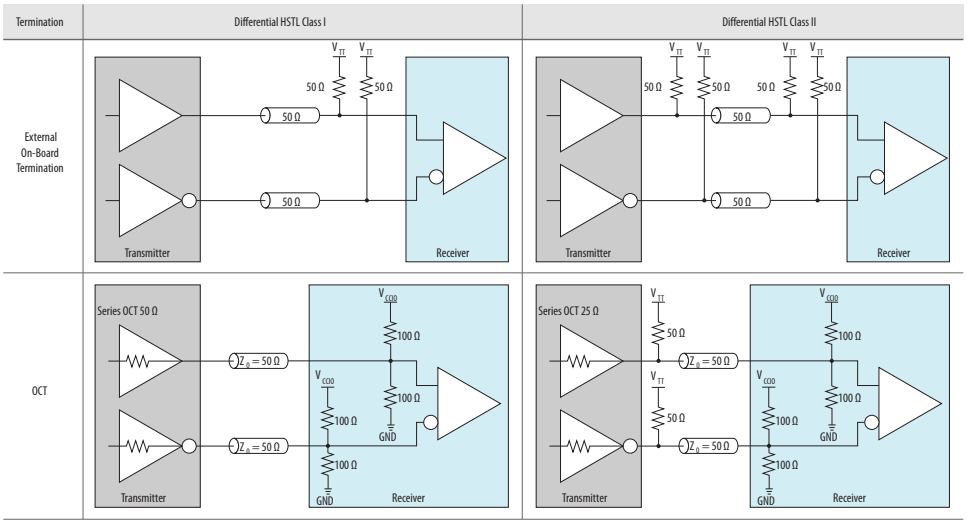
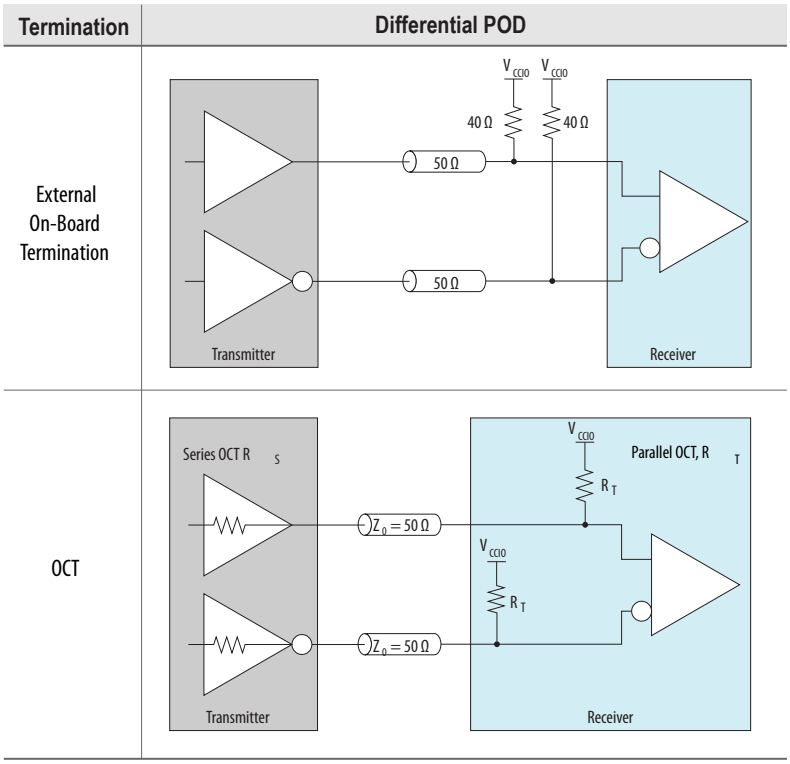


Figure 95. Differential POD I/O Standard Termination

This figure shows the details of Differential POD I/O termination on the Intel Arria 10 devices.



Related Information

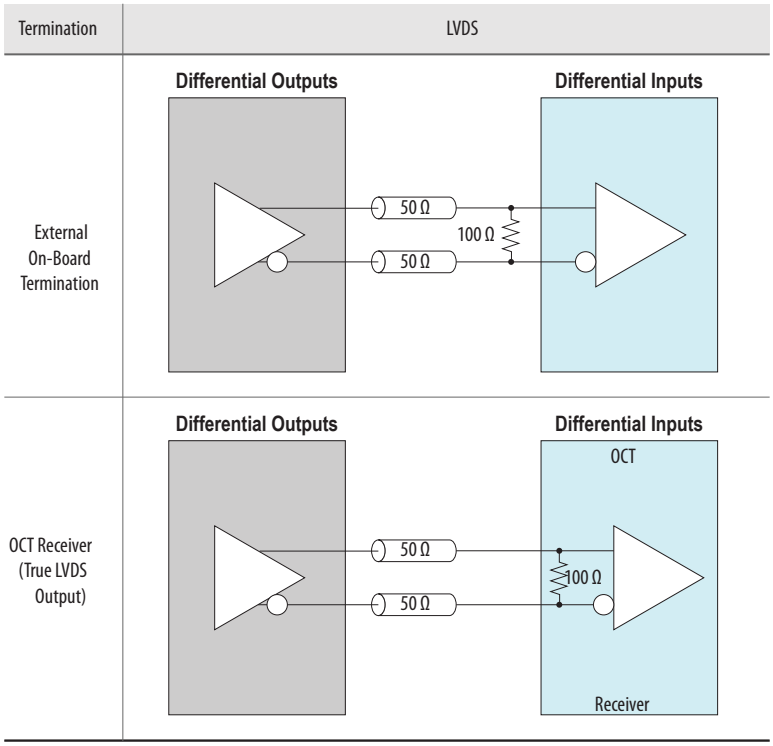
Differential I/O Termination for Intel Arria 10 Devices on page 143

5.5.5.2.2. 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.

Figure 96. 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.



Related Information

- Differential I/O Standards Specifications, *Intel Arria 10 Device Datasheet*
- Differential I/O Termination for Intel Arria 10 Devices on page 143

5.5.5.2.3. LVPECL Termination

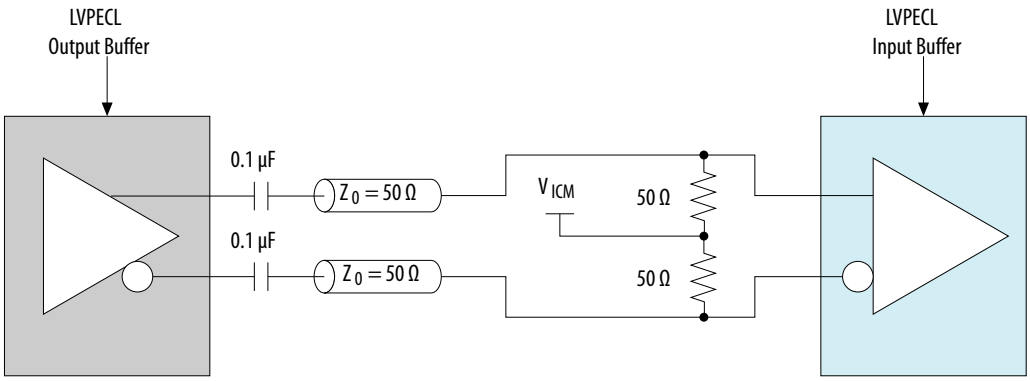
The Intel Arria 10 devices support the LVPECL I/O standard on input clock pins only:

- 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.

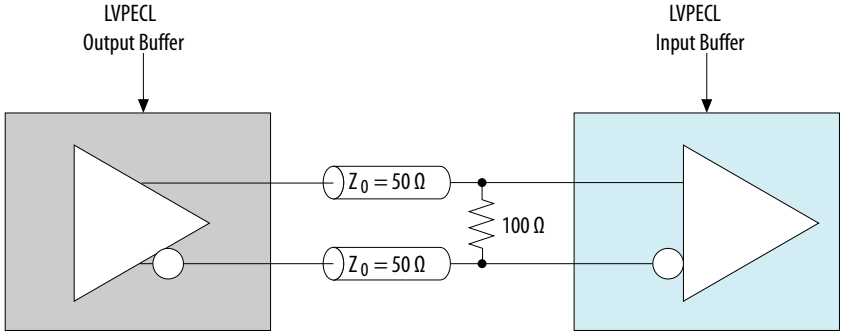
Note: Intel recommends that you use IBIS models to verify your LVPECL AC/DC-coupled termination.

Figure 97. LVPECL AC-Coupled External Termination



Support for DC-coupled LVPECL is available if the LVPECL output common mode voltage is within the Intel Arria 10 LVPECL input buffer specification.

Figure 98. LVPECL DC-Coupled External Termination



For information about the V_{ICM} specification, refer to the device datasheet.

Related Information

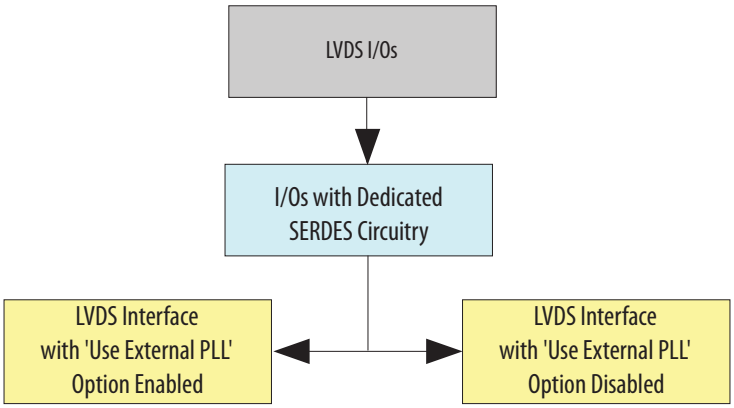
- Differential I/O Standards Specifications, *Intel Arria 10 Device Datasheet*
- Differential I/O Termination for Intel Arria 10 Devices on page 143

5.6. High Speed Source-Synchronous SERDES and DPA in Intel Arria 10 Devices

The high-speed differential I/O interfaces and DPA features in Intel Arria 10 devices provide advantages over single-ended I/Os and contribute to the achievable overall system bandwidth. Intel Arria 10 devices support the LVDS, mini-LVDS, and reduced swing differential signaling (RSDS) differential I/O standards.

Figure 99. I/O Bank Support for High-Speed Differential I/O

This figure shows the I/O bank support for high-speed differential I/O in the Intel Arria 10 devices.



Related Information

- I/O Standards Support for FPGA I/O in Intel Arria 10 Devices on page 101
  Provides information about the supported differential I/O standards.
- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- FPGA I/O Resources in Intel Arria 10 GX Packages on page 110
  Provides the number of LVDS channels.
- FPGA I/O Resources in Intel Arria 10 GT Packages on page 111
  Provides the number of LVDS channels.
- FPGA I/O Resources in Intel Arria 10 SX Packages on page 112
  Provides the number of LVDS channels.
- *Intel FPGA LVDS SERDES IP Core User Guide*

5.6.1. Intel Arria 10 LVDS SERDES Usage Modes

Table 60. Usage Modes Summary of the Intel Arria 10 LVDS SERDES

All SERDES usage modes in this table support SERDES factors of 3 to 10.

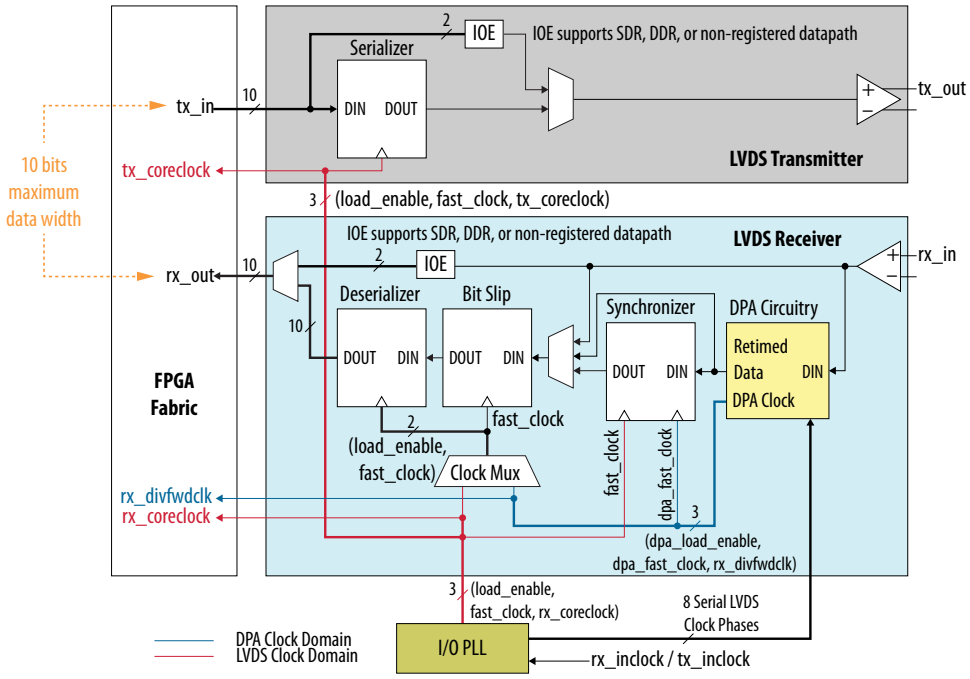
Usage Mode	Quick Guideline
Transmitter	In this mode, the SERDES block acts as a serializer.
DPA Receiver	• This mode is useful for source-synchronous clocking applications. • The dynamic phase alignment block (DPA) automatically adjusts the clock phase to achieve optimal data-to-clock skew.
Non-DPA Receiver	• This mode is useful for source-synchronous clocking applications. • You must manage the data-to-clock skew.
Soft-CDR Receiver	• The soft clock data recovery (soft-CDR) mode is useful for asynchronous clocking applications. • An asynchronous clock drives the Intel FPGA LVDS SERDES IP core. The IP core outputs a recovered clock from the received data.
Bypass the SERDES	You can bypass the serializer to use SERDES factor of 2 by using the Intel FPGA GPIO IP core: • Single data rate (SDR) mode—you do not require clocks. • Double data rate (DDR) mode—useful for slow source-synchronous clocking applications.

5.6.2. SERDES Circuitry

Each LVDS I/O channel in Intel Arria 10 devices has 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.

Figure 100. SERDES

This figure shows a transmitter and receiver block diagram for the LVDS SERDES circuitry with the interface signals of the transmitter and receiver data paths. The figure shows a shared PLL between the transmitter and receiver. If the transmitter and receiver do not share the same PLL, you require two I/O PLLs. In single data rate (SDR) and double data rate (DDR) modes, the data widths are 1 and 2 bits, respectively.



The Intel FPGA LVDS SERDES Intel FPGA IP transmitter and receiver require various clock and load enable signals from an I/O 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 Intel Arria 10 devices, refer to the device overview.

Related Information

- Summary of Features, *Intel Arria 10 Device Overview*
- Guideline: Use PLLs in Integer PLL Mode for LVDS on page 163

5.6.3. SERDES I/O Standards Support in Intel Arria 10 Devices

These 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 61. 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 POD12	Differential 1.2-V POD

Table 62. 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

continued...

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 POD12	`Differential 1.2-V POD`
mini-LVDS	`mini-LVDS`
RSDS	`RSDS`

5.6.4. Differential Transmitter in Intel Arria 10 Devices

The Intel Arria 10 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

Table 63. Dedicated Circuitries and Features of the Differential Transmitter

Dedicated Circuitry / Feature	Description
Differential I/O buffer	Supports LVDS, mini-LVDS, and RSDS
SERDES	Up to 10-bit wide serializer
Phase-locked loops (PLLs)	Clocks the load and shift registers
Programmable V_{OD}	Static
Programmable pre-emphasis	Boosts output current

Related Information

Guideline: Use PLLs in Integer PLL Mode for LVDS on page 163

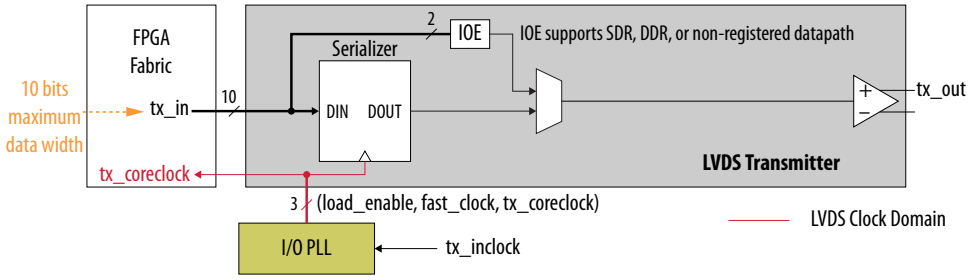
5.6.4.1. Transmitter Blocks in Intel Arria 10 Devices

The dedicated circuitry consists of a true differential buffer, a serializer, and I/O PLLs that you can share between the transmitter and receiver. The serializer takes up to 10-bit wide parallel data from the FPGA fabric and clocks the data into the load registers. Then, the serializer serializes the data using shift registers that are clocked by the I/O PLL. After serializing the data, the serializer sends the data to the differential buffer. The MSB of the parallel data is transmitted first.

Note: The PLL that drives the LVDS SERDES channel must operate in integer PLL mode. You do not need a PLL if you bypass the serializer.

Figure 101. LVDS Transmitter

This figure shows a block diagram of the transmitter. In SDR and DDR modes, the data width is 1 and 2 bits, respectively.



Related Information

Guideline: Use PLLs in Integer PLL Mode for LVDS on page 163

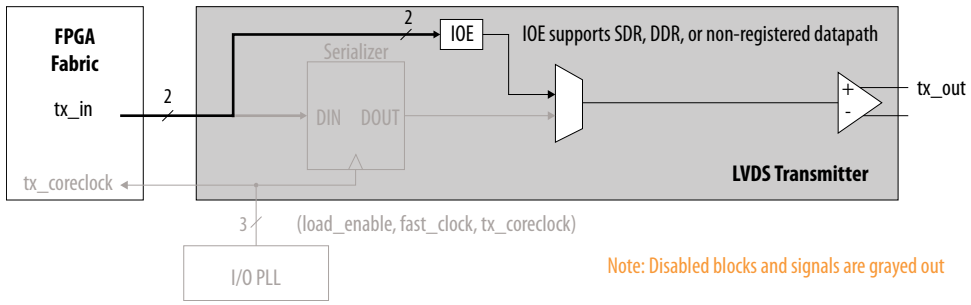
5.6.4.2. Serializer Bypass for DDR and SDR Operations

The I/O element (IOE) contains two data output registers that can each operate in either DDR or SDR mode.

You can bypass the serializer to support DDR (x2) and SDR (x1) operations to achieve a serialization factor of 2 and 1, respectively. The deserializer bypass is supported through the Intel FPGA GPIO Intel FPGA IP.

Figure 102. Serializer Bypass

This figure shows the serializer bypass path.



- In SDR mode:
  - The IOE data width is 1 bit.
  - Registered output path requires a clock.
  - Data is passed directly through the IOE.
- In DDR mode:
  - The IOE data width is 2 bits.
  - The Intel FPGA GPIO IP core requires a clock.
  - `tx_inclock` clocks the IOE register.

5.6.5. Differential Receiver in Intel Arria 10 Devices

The receiver has a differential buffer and I/O 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: The PLL that drives the LVDS SERDES channel must operate in integer PLL mode. You do not need a PLL if you bypass the deserializer

Table 64. Dedicated Circuitries and Features of the Differential Receiver

Dedicated Circuitry / Feature	Description
Differential I/O buffer	Supports LVDS, mini-LVDS, and RSDS
SERDES	Up to 10-bit wide deserializer
Phase-locked loops (PLLs)	Generates different phases of a clock for data synchronizer
Data realignment (Bit slip)	Inserts bit latencies into serial data
DPA	Chooses a phase closest to the phase of the serial data
Synchronizer (FIFO buffer)	Compensate for phase differences between the data and the receiver's input reference clock
Skew adjustment	Manual
On-chip termination (OCT)	100 Ω in LVDS I/O standards

Related Information

Guideline: Use PLLs in Integer PLL Mode for LVDS on page 163

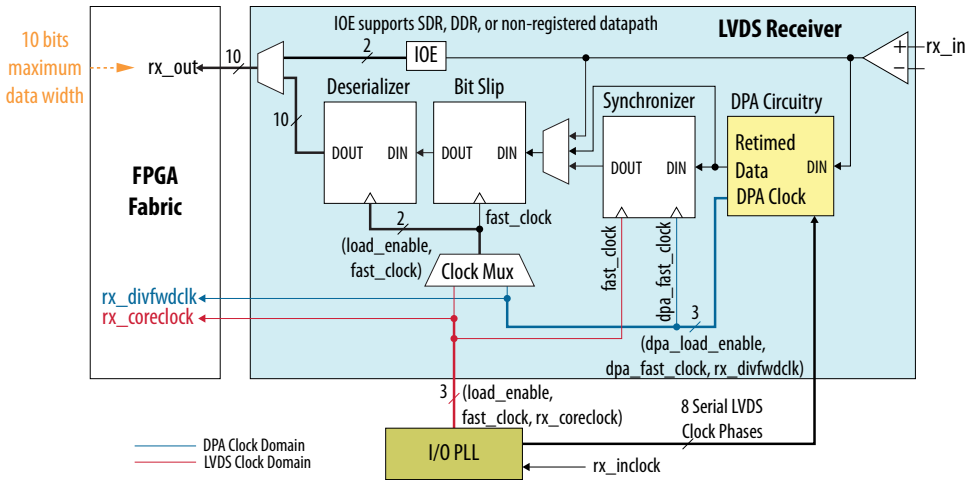
5.6.5.1. Receiver Blocks in Intel Arria 10 Devices

The Intel Arria 10 differential receiver has the following hardware blocks:

- DPA block
- Synchronizer
- Data realignment block (bit slip)
- Deserializer

Figure 103. Receiver Block Diagram

This 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.

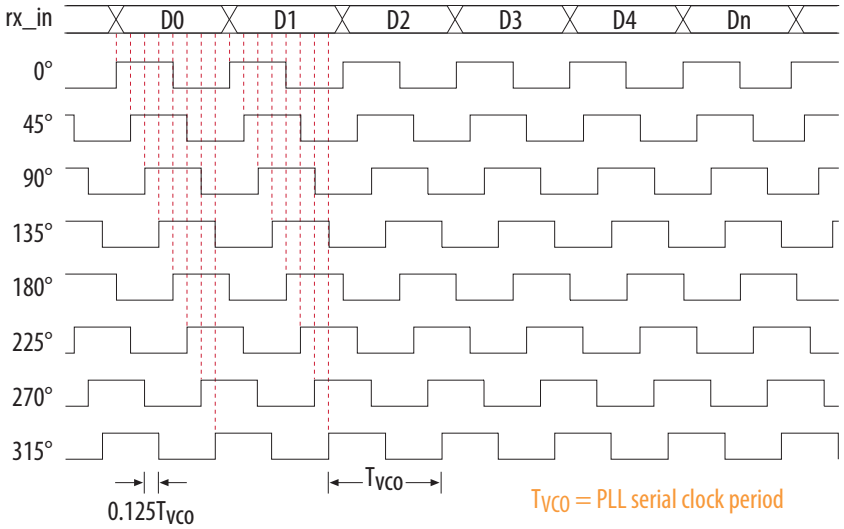


5.6.5.1.1. DPA Block

The DPA block takes in high-speed serial data from the differential input buffer and selects one of the eight phases that the I/O 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 unit interval (UI)[11], which is the maximum quantization error of the DPA. The eight phases of the clock are equally divided, offering a 45° resolution.

Figure 104. DPA Clock Phase to Serial Data Timing Relationship

This figure shows the possible phase relationships between the DPA clocks and the incoming serial data.



(11) The unit interval is the period of the clock running at the serial data rate (fast clock).

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_dpa_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. 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_dpa_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 163

5.6.5.1.2. Synchronizer

The synchronizer is a one-bit wide and six-bit deep FIFO buffer that compensates for the phase difference between `dpa_fast_clock`—the optimal clock that the DPA block selects—and the `fast_clock` that the I/O 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. Intel recommends that you use `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 163

5.6.5.1.3. Data Realignment Block (Bit Slip)

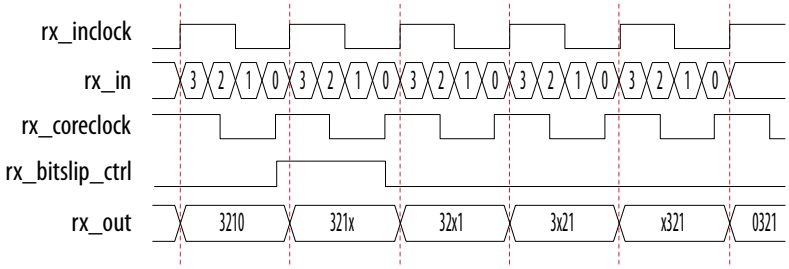
Skew in transmitted data and skew added by the link cause channel-to-channel skew on the received serial data streams. If you enable the DPA, the received data is captured with different clock phases on each channel. This difference may cause misalignment of the received data from channel to channel. 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_bitslip_ctrl` 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_bitslip_ctrl`. The requirements for the `rx_bitslip_ctrl` 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 four parallel clock cycles after the rising edge of `rx_bitslip_ctrl`.

Figure 105. Data Realignment Timing

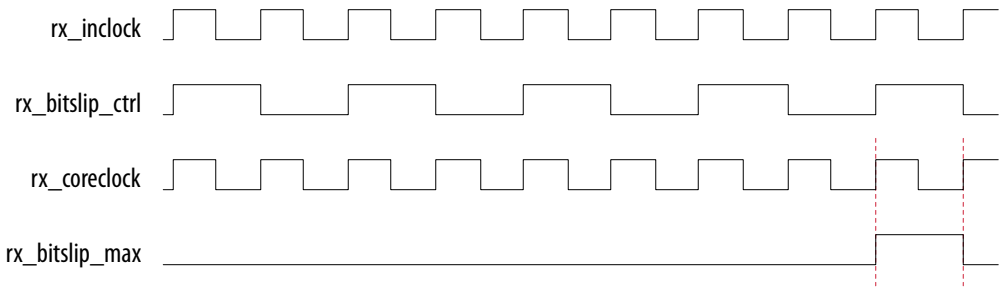
This figure shows receiver output (`rx_out`) after one bit slip pulse with the deserialization factor set to **4**.



The data realignment circuit has a bit slip rollover value set to the deserialization factor. An optional status port, `rx_bitslip_max`, is available to the FPGA fabric from each channel to indicate the reaching of the preset rollover point.

Figure 106. Receiver Data Realignment Rollover

This figure shows a preset value of four bit cycles before rollover occurs. The `rx_bitslip_max` signal pulses for one `rx_coreclock` cycle to indicate that rollover has occurred.



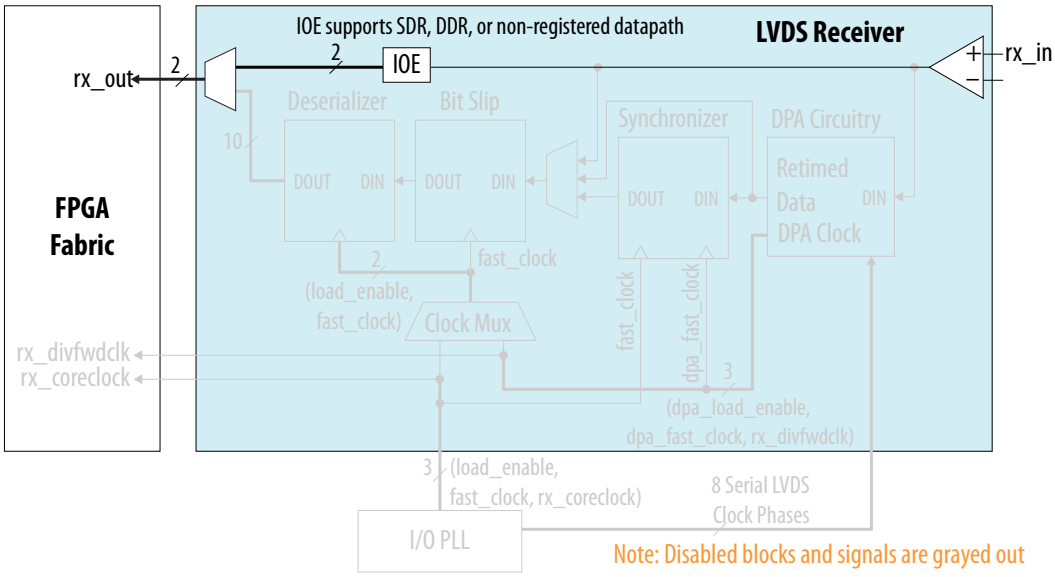
5.6.5.1.4. Deserializer

You can statically set the deserialization factor to x3, x4, x5, x6, x7, x8, x9, or x10 by using the Intel Quartus Prime software.

The IOE contains two data input registers that can operate in DDR or SDR mode. You can bypass the deserializer to support DDR (x2) and SDR (x1) operations. The deserializer bypass is supported through the Intel FPGA GPIO IP core.

Figure 107. Deserializer Bypass

This figure shows the deserializer bypass path.



- If you bypass the deserializer in SDR mode:
  - The IOE data width is 1 bit.
  - Registered input path requires a clock.
  - Data is passed directly through the IOE.
- If you bypass the deserializer in DDR mode:
  - The IOE data width is 2 bits.
  - The Intel FPGA GPIO IP core requires a clock.
  - `rx_inclock` clocks the IOE register. The clock must be synchronous to `rx_in`.
  - You must control the data-to-clock skew.

You cannot use the DPA and data realignment circuit when you bypass the deserializer.

5.6.5.2. Receiver Modes in Intel Arria 10 Devices

The Intel Arria 10 devices support the following receiver modes:

- Non-DPA mode
- DPA mode
- Soft-CDR mode

Note: If you use DPA mode, follow the recommended initialization and reset flow. The recommended flow ensures that the DPA circuit can detect the optimum phase tap from the PLL to capture data on the receiver.

Related Information

Recommended Initialization and Reset Flow, *Intel FPGA LVDS SERDES IP Core User Guide*
Provides the recommended procedure to initialize and reset the LVDS SERDES IP core.

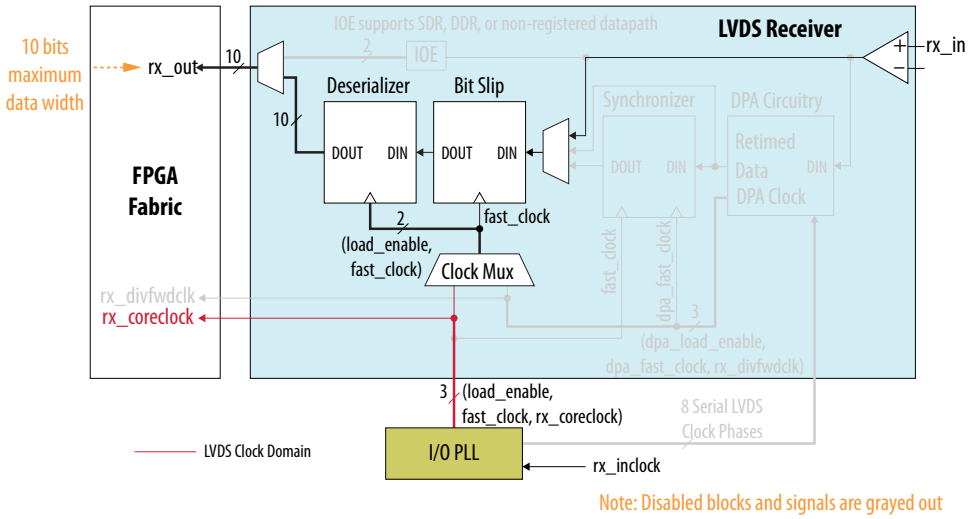
5.6.5.2.1. Non-DPA Mode

The non-DPA mode disables the DPA and synchronizer blocks. Input serial data is registered at the rising edge of the serial `fast_clock` clock that is produced by the I/O PLLs.

The `fast_clock` clock that is generated by the I/O PLLs clocks the data realignment and deserializer blocks.

Figure 108. Receiver Datapath in Non-DPA Mode

This figure shows the non-DPA datapath block diagram.

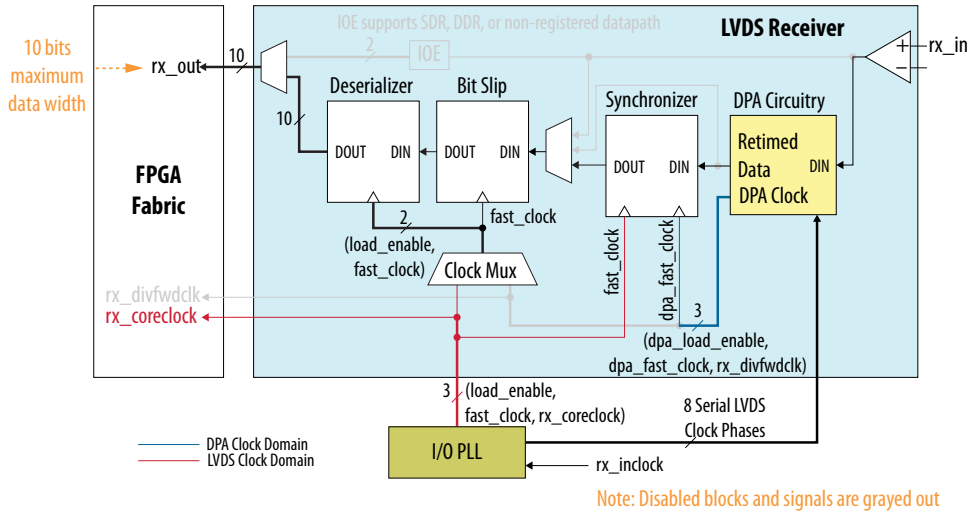


5.6.5.2.2. DPA Mode

The DPA block chooses the best possible clock (`dpa_fast_clock`) from the eight fast clocks that the I/O PLL sent. This serial `dpa_fast_clock` clock is used for writing the serial data into the synchronizer. A serial `fast_clock` clock is used for reading the serial data from the synchronizer. The same `fast_clock` clock is used in data realignment and deserializer blocks.

Figure 109. Receiver Datapath in DPA Mode

This figure shows the DPA mode datapath. In the figure, all the receiver hardware blocks are active.



Note: In DPA mode, you must place all receiver channels of an LVDS instance in one I/O bank. Because each I/O bank has a maximum of 24 LVDS I/O buffer pairs, each LVDS instance can support a maximum of 24 DPA channels.

Related Information

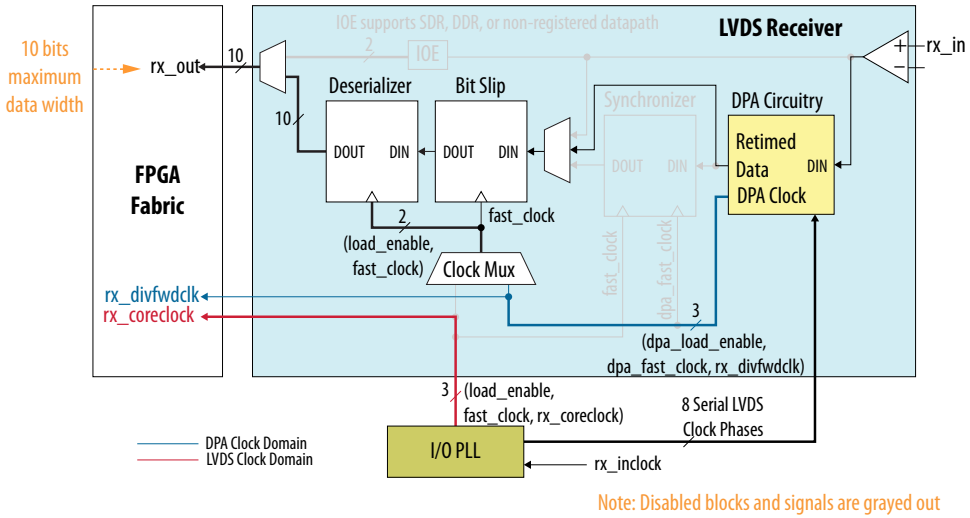
- Guideline: Use PLLs in Integer PLL Mode for LVDS on page 163
- Receiver Blocks in Intel Arria 10 Devices on page 153
  Lists and describes the receiver hardware blocks.

5.6.5.2.3. Soft-CDR Mode

The Intel Arria 10 LVDS channel offers the soft-CDR mode to support the GbE and SGMII protocols. A receiver PLL uses the local clock source for reference.

Figure 110. Receiver Datapath in Soft-CDR Mode

This figure shows the soft-CDR mode datapath.



In soft-CDR mode, the synchronizer block is inactive. The DPA circuitry selects an optimal DPA clock phase to sample the data. This clock is used 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_dpa_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 Intel Arria 10 device family. In soft-CDR mode, the `rx_dpa_locked` signal is not valid because the DPA continuously changes its phase to track PPM differences between the upstream transmitter and the local receiver input reference clocks. However, you can use the `rx_dpa_locked` signal to determine the initial DPA locking conditions that indicate the DPA has selected the optimal phase tap to capture the data. The `rx_dpa_locked` signal is expected to deassert when operating in soft-CDR mode. The parallel clock, `rx_coreclock`, generated by the I/O PLLs, is also forwarded to the FPGA fabric.

Note: In soft-CDR mode, you must place all receiver channels of an LVDS instance in one I/O bank. Because each I/O bank has a maximum of 12 PCLK resources, each LVDS instance can support a maximum of 12 soft-CDR channels.

Related Information

- Guideline: LVDS SERDES Pin Pairs for Soft-CDR Mode on page 182
- Periphery Clock Networks on page 76
  Provides more information about PCLK networks.

5.6.6. PLLs and Clocking for Intel Arria 10 Devices

To generate the parallel clocks (`rx_coreclock` and `tx_coreclock`) and high-speed clocks (`fast_clock`), the Intel Arria 10 devices provide I/O PLLs in the high-speed differential I/O receiver and transmitter channels.

Related Information

- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- Clocking Differential Transmitters on page 161
- Clocking Differential Receivers on page 162
- Guideline: Use PLLs in Integer PLL Mode for LVDS on page 163
- Guideline: Use High-Speed Clock from PLL to Clock LVDS SERDES Only on page 163
- Guideline: Pin Placement for Differential Channels on page 163
- LVDS Interface with External PLL Mode on page 166
- Guideline: I/O Standards Supported for I/O PLL Reference Clock Input Pin on page 179

5.6.6.1. Clocking Differential Transmitters

The I/O PLL generates the load enable (`load_enable`) signal and the `fast_clock` 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.

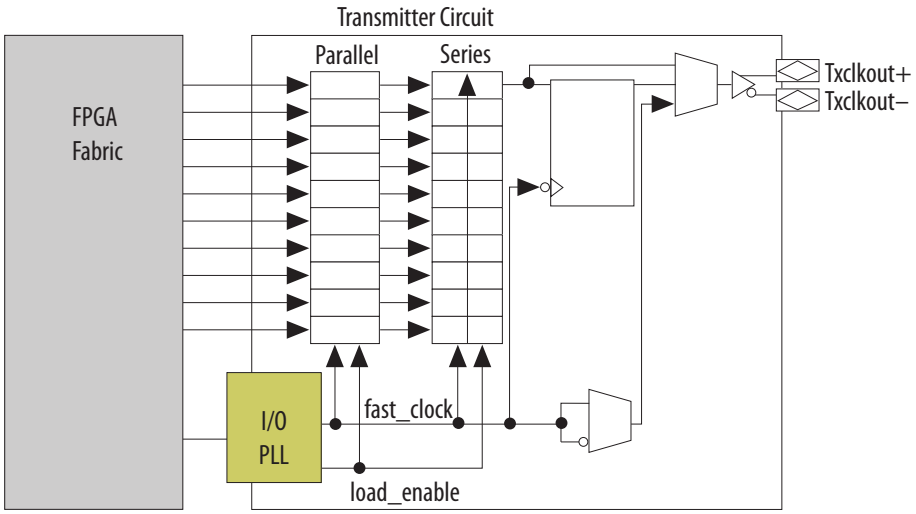
You can configure any Intel Arria 10 transmitter data channel to generate a source-synchronous transmitter clock output. This flexibility allows the placement of the output clock near the data outputs to simplify board layout and reduce clock-to-data skew.

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 parameter editor:

- 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 at 0° or 180° (edge- or center-aligned). The I/O PLLs provide additional support for other phase shifts in 45° increments.

Figure 111. Transmitter in Clock Output Mode

This figure shows the transmitter in clock output mode. In clock output mode, you can use an LVDS channel as a clock output channel.



Related Information

- Guideline: Use PLLs in Integer PLL Mode for LVDS on page 163
- PLLs and Clocking for Intel Arria 10 Devices on page 161

5.6.6.2. Clocking Differential Receivers

The I/O 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 I/O 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 163
- PLLs and Clocking for Intel Arria 10 Devices on page 161

5.6.6.2.1. Guideline: Clocking DPA Interfaces Spanning Multiple I/O Banks

DPA interfaces that use more than 24 channels span multiple I/O banks. Intel recommends that you feed the I/O PLL in each I/O bank of the DPA interface with its own dedicated `refclk` pin. Follow this recommendation to achieve the maximum DPA LVDS specifications listed in the device datasheet.

Related Information

High-Speed I/O Specifications, *Intel Arria 10 Device Datasheet*

5.6.6.2.2. Guideline: I/O PLL Reference Clock Source for DPA or Non-DPA Receiver

The reference clock to the I/O PLL for the DPA or non-DPA LVDS receiver must come from the dedicated reference clock pin within the I/O bank.

Note: This requirement is not applicable to LVDS transmitters.

5.6.6.3. Guideline: Use PLLs in Integer PLL Mode for LVDS

Each I/O bank has its own PLL (I/O PLL) to drive the LVDS channels. These I/O PLLs operate in integer mode only.

Related Information

PLLs and Clocking for Intel Arria 10 Devices on page 161

5.6.6.4. 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.

Related Information

- PLL Specifications, *Intel Arria 10 Device Datasheet*
- PLLs and Clocking for Intel Arria 10 Devices on page 161

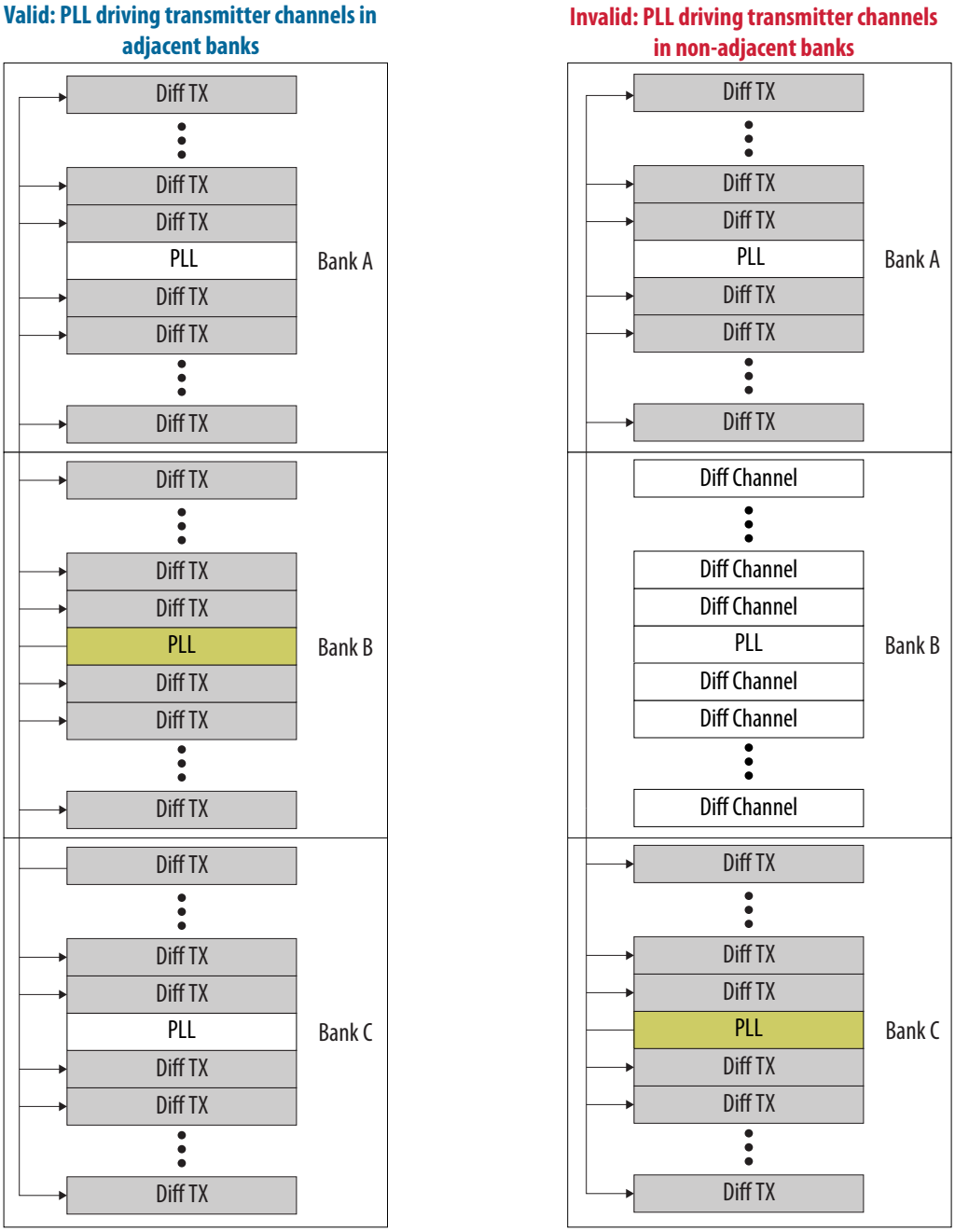
5.6.6.5. Guideline: Pin Placement for Differential Channels

Each I/O bank contains its own PLL. The I/O bank PLL can drive all receiver and transmitter channels in the same bank, and transmitter channels in adjacent I/O banks. However, the I/O bank PLL cannot drive receiver channels in another I/O bank or transmitter channels in non-adjacent I/O banks.

PLLs Driving Differential Transmitter Channels

For differential transmitters, the PLL can drive the differential transmitter channels in its own I/O bank and adjacent I/O banks. However, the PLL cannot drive the channels in a non-adjacent I/O bank.

Figure 112. PLLs Driving Differential Transmitter Channels



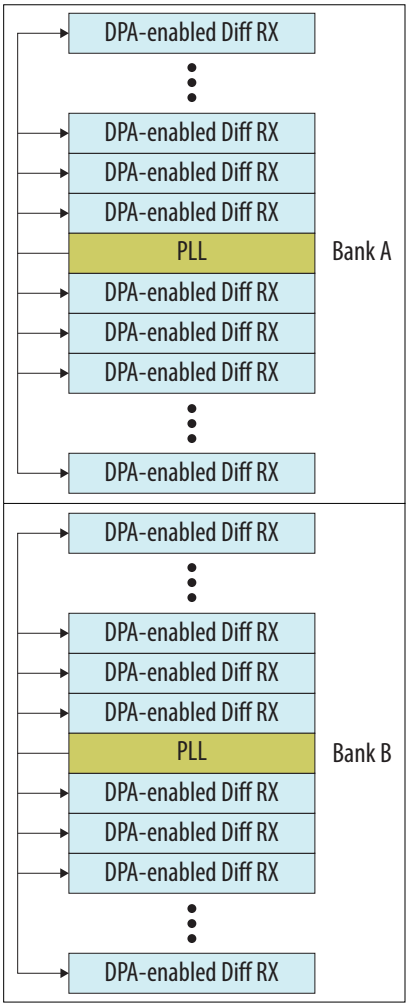
PLLs Driving DPA-Enabled Differential Receiver Channels

For differential receivers, the PLL can drive only the channels within the same I/O bank.

Each differential receiver in an I/O bank 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 assign the unused I/O pins in the bank to single-ended or differential I/O standards that has the same V_{CCIO} voltage level used by the bank.

DPA usage adds some constraints to the placement of high-speed differential receiver channels. The Intel Quartus Prime compiler automatically checks the design and issues error messages if there are placement guidelines violations. Adhere to the guidelines to ensure proper high-speed I/O operation.

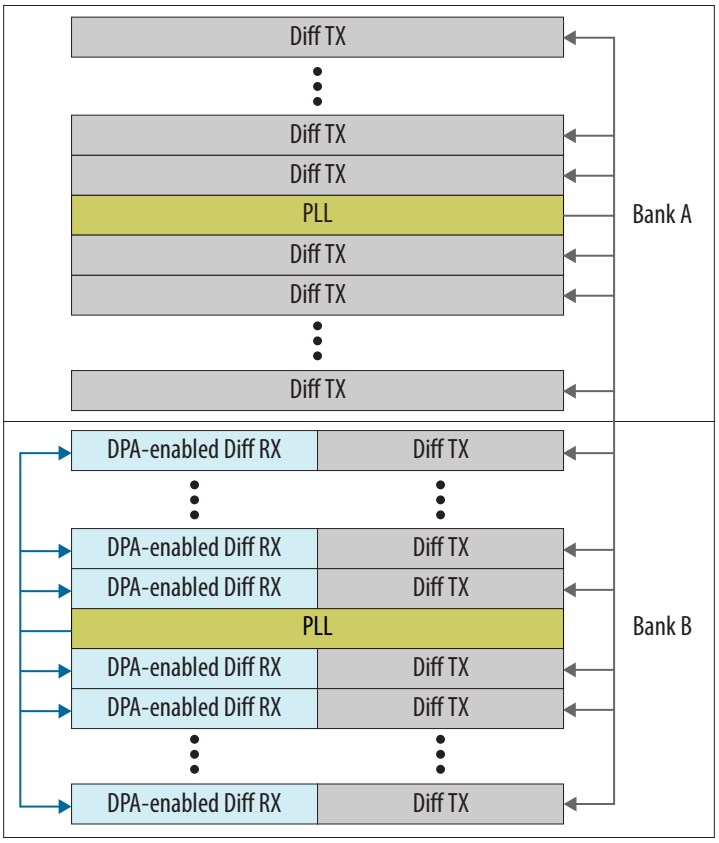
Figure 113. PLLs Driving DPA-Enabled Differential Receiver Channels



PLLs Driving DPA-Enabled Differential Receiver and Transmitter Channels in LVDS Interface Spanning Multiple I/O Banks

If you use both differential transmitter and DPA-enabled receiver channels in a bank, the PLL can drive the transmitters spanning multiple adjacent I/O banks, but only the receivers in its own I/O bank.

Figure 114. PLLs Driving DPA-Enabled Differential Receiver and Transmitter Channels Across I/O Banks



Related Information

PLLs and Clocking for Intel Arria 10 Devices on page 161

5.6.6.6. LVDS Interface with External PLL Mode

The Intel FPGA LVDS SERDES IP core parameter editor 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.

If you enable the **Use External PLL** option with the Intel FPGA LVDS SERDES IP core transmitter and receiver, the following signals are required from the Intel FPGA IOPLL Intel FPGA IP:

- Serial clock (fast clock) input to the SERDES of the Intel FPGA LVDS SERDES IP core transmitter and receiver
- Load enable to the SERDES of the Intel FPGA LVDS SERDES IP core transmitter and receiver
- Parallel clock (core clock) used to clock the transmitter FPGA fabric logic and parallel clock used for the receiver
- Asynchronous PLL reset port of the Intel FPGA LVDS SERDES IP core receiver
- PLL VCO signal for the DPA and soft-CDR modes of the Intel FPGA LVDS SERDES IP core receiver

The **Clock Resource Summary** tab in the LVDS SERDES IP core parameter editor provides the details for the signals in the preceding list.

You must instantiate an Intel FPGA IOPLL IP core to generate the various clocks and load enable signals. You must configure these settings in Intel FPGA IOPLL IP core parameter editor:

- **LVDS External PLL** options in the **Settings** tab
- **Output Clocks** options in the **PLL** tab
- **Compensation Mode** option in the **PLL** tab

Table 65. Compensation Mode Setting to Generate Intel FPGA IOPLL IP Core

When you generate the Intel FPGA IOPLL IP core, use the PLL setting in this table for the corresponding LVDS functional mode.

LVDS Functional Mode	Intel FPGA IOPLL IP Core Setting
TX, RX DPA, RX Soft-CDR	Direct mode
RX non-DPA	LVDS compensation mode

Related Information

- *Intel FPGA LVDS SERDES IP Core User Guide*
- PLLs and Clocking for Intel Arria 10 Devices on page 161
- Intel FPGA IOPLL IP Core Signal Interface with Intel FPGA LVDS SERDES IP Core on page 168
- Intel FPGA IOPLL Parameter Values for External PLL Mode on page 169
- Connection between Intel FPGA IOPLL and Intel FPGA LVDS SERDES in External PLL Mode on page 171

5.6.6.6.1. Intel FPGA IOPLL IP Core Signal Interface with Intel FPGA LVDS SERDES IP Core

Table 66. Signal Interface between Intel FPGA IOPLL and Intel FPGA LVDS SERDES IP cores

This table lists the signal interface between the output ports of the Intel FPGA IOPLL IP core and the input ports of the Intel FPGA LVDS SERDES IP core transmitter and receiver.

From the Intel FPGA IOPLL IP core	To the Intel FPGA LVDS SERDES IP core transmitter	To the Intel FPGA LVDS SERDES IP core receiver
`lvds_clk[0]` (serial clock output signal) • Configure this signal using `outclk0` in the PLL. • Select **Enable LVDS_CLK/ LOADEN 0** or **Enable LVDS_CLK/ LOADEN 0 & 1** option for the **Access to PLL LVDS_CLK/ LOADEN output port** setting. In most cases, select **Enable LVDS_CLK/LOADEN 0**. The serial clock output can only drive `ext_fclk` on the Intel FPGA LVDS SERDES IP core transmitter and receiver. This clock cannot drive the core logic.	`ext_fclk` (serial clock input to the transmitter)	`ext_fclk` (serial clock input to the receiver)
`loaden[0]` (load enable output) • Configure this signal using `outclk1` in the PLL. • Select **Enable LVDS_CLK/ LOADEN 0** or **Enable LVDS_CLK/ LOADEN 0 & 1** option for the **Access to PLL LVDS_CLK/ LOADEN output port** setting. In most cases, select **Enable LVDS_CLK/LOADEN 0**.	`ext_loaden` (load enable to the transmitter)	`ext_loaden` (load enable for the deserializer) This signal is not required for LVDS receiver in soft-CDR mode.
`outclk2` (parallel clock output)	`ext_coreclock` (parallel core clock)	`ext_coreclock` (parallel core clock)
`locked`	—	`pll_areset` (asynchronous PLL reset port)
`phout[7:0]` • This signal is required only for LVDS receiver in DPA or soft-CDR mode. • Configure this signal by turning on **Specify VCO frequency** in the PLL and specifying the **VCO frequency** value. • Turn on **Enable access to PLL DPA output port**.	—	`ext_vcoph` This signal is required only for LVDS receiver in DPA or soft-CDR mode.

Related Information

- *Intel FPGA LVDS SERDES IP Core User Guide*
  Provides more information about the different clocking requirement for soft SERDES.
- LVDS Interface with External PLL Mode on page 166

5.6.6.6.2. Intel FPGA IOPLL Parameter Values for External PLL Mode

The following examples show the clocking requirements to generate output clocks for Intel FPGA LVDS SERDES IP core using the Intel FPGA IOPLL 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, Intel recommends that you first instantiate your Intel FPGA LVDS SERDES IP core 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 Intel FPGA IOPLL IP core parameter editor and then connect the appropriate output to the Intel FPGA LVDS SERDES IP cores.

Table 67. Example: Generating Output Clocks Using an Intel FPGA IOPLL IP core (Receiver in Non-DPA Mode)

This table lists the parameter values that you can set in the Intel FPGA IOPLL IP core parameter editor to generate three output clocks using an Intel FPGA IOPLL IP core if you are using the non-DPA receiver.

Parameter	outclk0 (Connects as `lvds_clk[0]` to the `ext_fclk` port of Intel FPGA LVDS SERDES IP core transmitter or receiver)	outclk1 (Connects as `loaden[0]` to the `ext_loaden` port of Intel FPGA LVDS SERDES IP core transmitter or receiver)	outclk2 (Used as the core clock for the parallel data registers for both transmitter and receiver, and connects to the `ext_coreclock` port of Intel FPGA LVDS SERDES IP core)
Frequency	data rate	data rate/serialization factor	data rate/serialization factor
Phase shift	180°	[(deserialization factor – 1)/ deserialization factor] x 360°	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 (outclk0) ensures that the input data is center-aligned with respect to the outclk0, as shown in the following figure.

Figure 115. Phase Relationship for External PLL Interface Signals

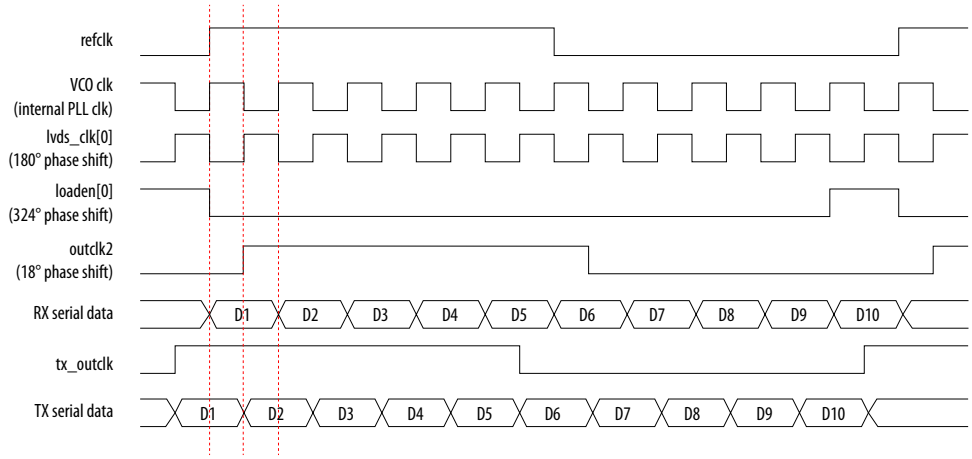


Table 68. Example: Generating Output Clocks Using an Intel FPGA IOPLL IP core (Receiver in DPA or Soft-CDR Mode)

This table lists the parameter values that you can set in the Intel FPGA IOPLL IP core parameter editor to generate four output clocks using an Intel FPGA IOPLL IP core if you are using the DPA or soft-CDR receiver. The `locked` output port of Intel FPGA IOPLL IP core must be inverted and connected to the `pll_areset` port of the Intel FPGA LVDS SERDES IP core if you are using the DPA or soft-CDR receiver.

Parameter	outclk0 (Connects as `lvds_clk[0]` to the `ext_fclk` port of Intel FPGA LVDS SERDES IP core transmitter or receiver)	outclk1 (Connects as `loaden[0]` to the `ext_loaden` port of Intel FPGA LVDS SERDES IP core transmitter or receiver) Not required for the soft-CDR receiver.	outclk2 (Used as the core clock for the parallel data registers for both transmitter and receiver, and connects to the `ext_coreclock` port of Intel FPGA LVDS SERDES IP core)	VCO Frequency (Connects as `phout[7:0]` to the `ext_vcoph[7:0]` port of Intel FPGA LVDS SERDES IP core)
Frequency	data rate	data rate/serialization factor	data rate/serialization factor	data rate
Phase shift	180°	[(deserialization factor - 1)/deserialization factor] x 360°	180/serialization factor (outclk0 phase shift divided by the serialization factor)	—
Duty cycle	50%	100/serialization factor	50%	—

Table 69. Example: Generating Output Clocks Using a Shared Intel FPGA IOPLL IP core for Transmitter Spanning Multiple Banks Shared with Receiver Channels (Receiver in DPA or Soft-CDR Mode)

This table lists the parameter values that you can set in the Intel FPGA IOPLL IP core parameter editor to generate six output clocks using an Intel FPGA IOPLL IP core. Use these settings if you use transmitter channels that span multiple banks shared with receiver channels in DPA or soft-CDR mode. The `locked` output port of Intel FPGA IOPLL IP core must be inverted and connected to the `pll_areset` port of the Intel FPGA LVDS SERDES IP core if you are using the DPA or soft-CDR mode.

Parameter	outclk0 (Connects as `lvds_clk[0]` to the `ext_fclk` port of Intel FPGA LVDS SERDES IP core receiver)	outclk1 (Connects as `loaden[0]` to the `ext_loaden` port of Intel FPGA LVDS SERDES IP core receiver) Not required for the soft-CDR receiver.	outclk4 (Used as the core clock for the parallel data registers for both transmitter and receiver, and connects to the `ext_coreclock` port of Intel FPGA LVDS SERDES IP core)	VCO Frequency (Connects as `phout[7:0]` to the `ext_vcoph[7:0]` port of Intel FPGA LVDS SERDES IP core)
	outclk2 (Connects as `lvds_clk[1]` to the `ext_fclk` port of Intel FPGA LVDS SERDES IP core transmitter)	outclk3 (Connects as `loaden[1]` to the `ext_loaden` port of Intel FPGA LVDS SERDES IP core transmitter)		
Frequency	data rate	data rate/serialization factor	data rate/serialization factor	data rate
Phase shift	180°	[(deserialization factor - 1)/deserialization factor] x 360°	180/serialization factor (outclk0 phase shift divided by the serialization factor)	—
Duty cycle	50%	100/serialization factor	50%	—

Related Information

- Receiver Skew Margin for Non-DPA Mode on page 174
  RSKM equation used for the phase shift calculations.
- LVDS Interface with External PLL Mode on page 166

5.6.6.6.3. Connection between Intel FPGA IOPLL and Intel FPGA LVDS SERDES in External PLL Mode

Figure 116. Non-DPA LVDS Receiver Interface with Intel FPGA IOPLL IP Core in External PLL Mode

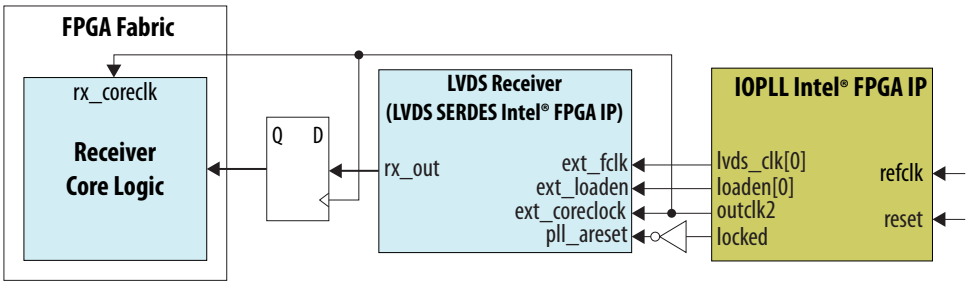


Figure 117. DPA LVDS Receiver Interface with the Intel FPGA IOPLL IP Core in External PLL Mode

Invert the `locked` output port and connect it to the `pll_areset` port.

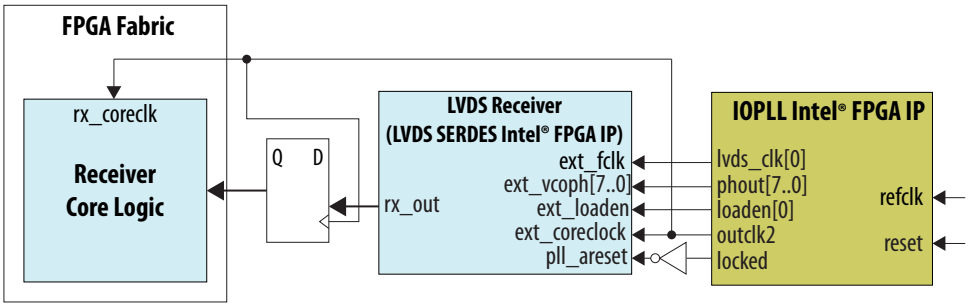


Figure 118. Soft-CDR LVDS Receiver Interface with the Intel FPGA IOPLL IP Core in External PLL Mode

Invert the `locked` output port and connect it to the `pll_areset` port.

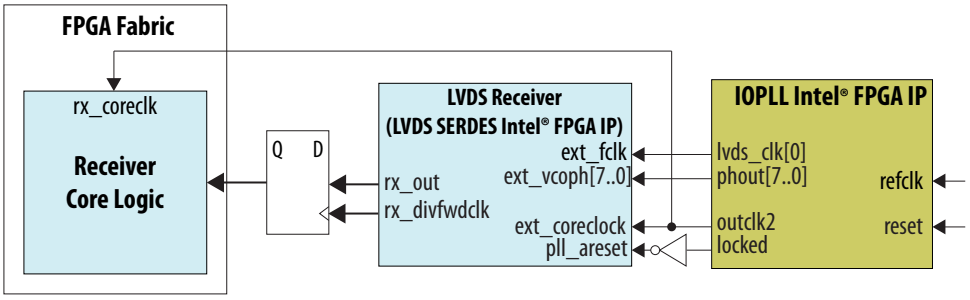
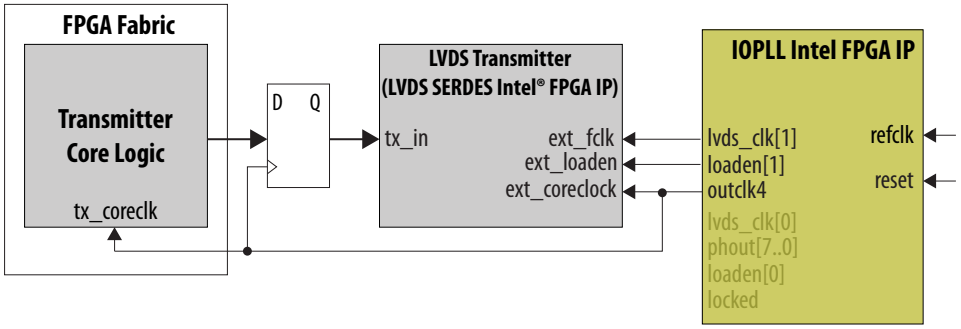


Figure 119. LVDS Transmitter Interface with the Intel FPGA IOPLL IP Core in External PLL Mode

Connect the I/O PLL `lvds_clk[1]` and `loaden[1]` ports to the `ext_fclk` and `ext_loaden` ports of the LVDS transmitter.



The `ext_coreclock` port is automatically enabled in the LVDS IP core in external PLL mode. The Intel Quartus Prime compiler outputs error messages if this port is not connected as shown in the preceding figures.

Related Information

LVDS Interface with External PLL Mode on page 166

5.6.7. Timing and Optimization for Intel Arria 10 Devices

5.6.7.1. Source-Synchronous Timing Budget

The topics in this section describe the timing budget, waveforms, and specifications for source-synchronous signaling in the Intel Arria 10 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.

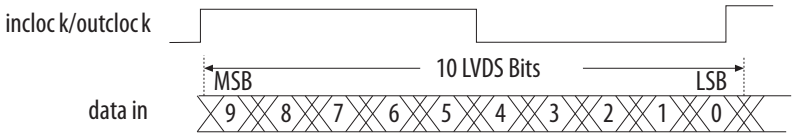
This section defines the source-synchronous differential data orientation timing parameters, the timing budget definitions for the Intel Arria 10 device family, and how to use these timing parameters to determine the maximum performance of a design.

5.6.7.1.1. 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 120. Bit Orientation in the Intel Quartus Prime Software

This figure shows the data bit orientation of the x10 mode.



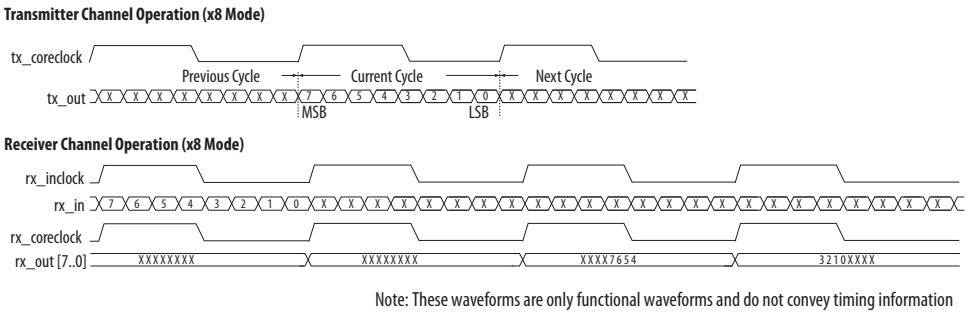
5.6.7.1.2. Differential I/O Bit Position

Data synchronization is necessary for successful data transmission at high frequencies.

Figure 121. Bit-Order and Word Boundary for One Differential Channel

This 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.



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

Table 70. 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

continued...

Receiver Channel Data Number	Internal 8-Bit Parallel Data	
	MSB Position	LSB Position
9	71	64
10	79	72
11	87	80
12	95	88
13	103	96
14	111	104
15	119	112
16	127	120
17	135	128
18	143	136

5.6.7.1.3. Transmitter Channel-to-Channel Skew

The receiver skew margin calculation uses the transmitter channel-to-channel skew (TCCS)—an important parameter based on the Intel Arria 10 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.

For Intel Arria 10 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 Intel Arria 10 device. You can use the recommended trace delay numbers shown 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

- High-Speed I/O Specifications, *Intel Arria 10 Device Datasheet*
- *Intel FPGA LVDS SERDES IP Core User Guide*
  Provides more information about the LVDS Transmitter/Receiver Package Skew Compensation report panel.

5.6.7.1.4. 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.

Related Information

- *Intel FPGA LVDS SERDES IP Core User Guide*
  Provides more information about the LVDS Transmitter/Receiver Package Skew Compensation report panel.
- The Intel Quartus Prime Timing Analyzer, *Intel Quartus Prime Standard Edition Handbook Volume 3: Verification*
  Provides more information about .sdc commands and the Timing Analyzer.
- I/O Timing Analysis
- Obtaining RSKM Report
- Obtaining TCCS Report

RSKM Equation

The RSKM equation expresses the relationship between RSKM, TCCS, and SW.

Figure 122. RSKM Equation

$$RSKM = \frac{TUI - SW - TCCS}{2}$$

Conventions used for the equation:

- RSKM—the timing margin between the clock input of the receiver and the data input sampling window, and the jitter induced from core noise and I/O switching noise.
- 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 the LVDS receiver samples the data successfully. The SW is a device property and varies according to device speed grade.
- TCCS—the timing difference between the fastest and the slowest output edges across channels driven by the same PLL. The TCCS measurement includes the t_{CO} variation, clock, and clock skew.

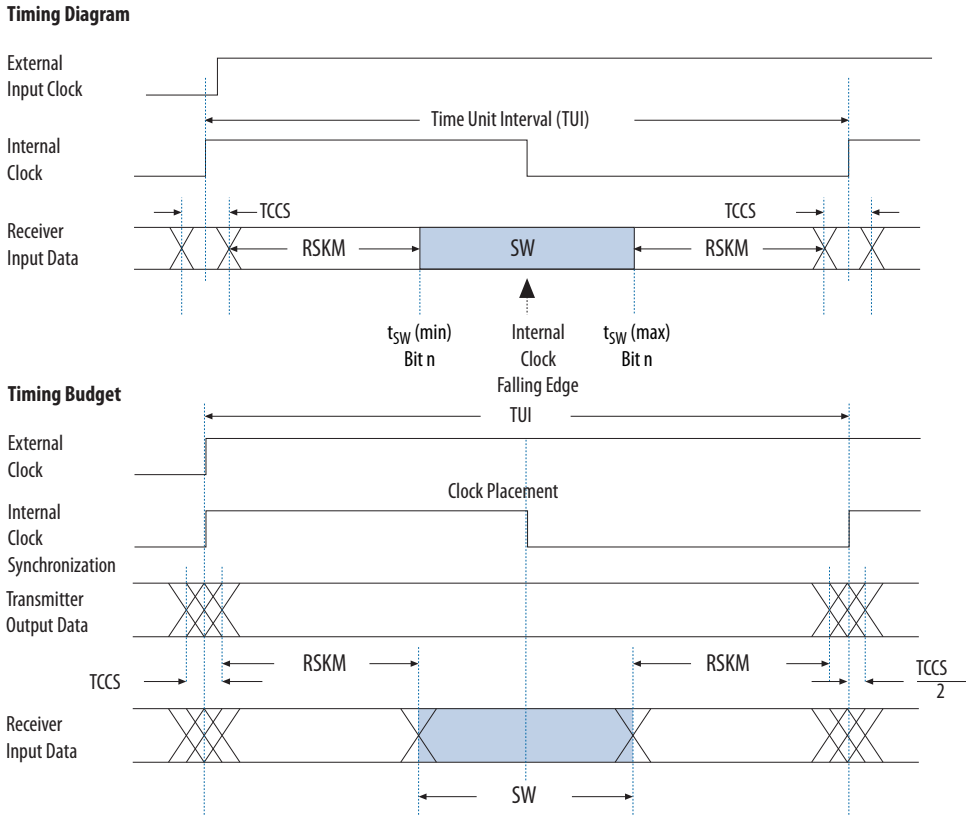
Note: If there is additional board channel-to-channel skew, consider the total receiver channel-to-channel skew (RCCS) instead of TCCS.
Total RCCS = TCCS+board channel-to-channel skew.

You must calculate the RSKM value, based on the data rate and device, to determine if the LVDS receiver can sample the data:

- A positive RSKM value, after deducting transmitter jitter, indicates that the LVDS receiver can sample the data properly.
- A negative RSKM value, after deducting transmitter jitter, indicates that the LVDS receiver cannot sample the data properly.

Figure 123. Differential High-Speed Timing Diagram and Timing Budget for Non-DPA Mode

This figure shows the relationship between the RSKM, TCCS, and the SW of the receiver.



RSKM Report for LVDS Receiver

For LVDS receivers, the Intel Quartus Prime software provides an RSKM report showing the SW, TUI, and RSKM values for the non-DPA LVDS mode.

- To generate the RSKM report, run the `report_RSKM` command in the TimeQuest Timing Analyzer. The RSKM report is available in the TimeQuest Timing Analyzer section of the Intel Quartus Prime compilation report.
- To obtain a more realistic RSKM value, assign the input delay to the LVDS receiver through the constraints menu of the TimeQuest 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 references the source-synchronous clock that feeds the LVDS receiver.
- If you do not set any input delay in the TimeQuest 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**) by using the `set_input_delay` command.

Example: RSKM Calculation

This example shows the RSKM calculation for FPGA devices at 1 Gbps data rate with a 200 ps board channel-to-channel skew.

- TCCS = 150 ps
- SW = 300 ps
- TUI = 1000 ps
- Total RCCS = TCCS + Board channel-to-channel skew = 150 ps + 200 ps = 350 ps
- RSKM = (TUI – SW – RCCS) / 2 = (1000 ps – 300 ps – 350 ps) / 2 = 175 ps

If the RSKM is greater than 0 ps after deducting transmitter jitter, the non-DPA receiver will work correctly.

5.7. Using the I/Os and High Speed I/Os in Intel Arria 10 Devices

5.7.1. I/O and High-Speed I/O General Guidelines for Intel Arria 10 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: VREF Sources and VREF Pins on page 177

Guideline: Observe Device Absolute Maximum Rating for 3.0 V Interfacing on page 178

Guideline: I/O Standards Supported for I/O PLL Reference Clock Input Pin on page 179

5.7.1.1. Guideline: V_{REF} Sources and VREF Pins

For Intel Arria 10 devices, consider the following VREF pins guidelines:

- Intel Arria 10 devices support internal and external V_{REF} sources.
  - There is an external VREF pin for every I/O bank, providing one external V_{REF} source for all I/Os in the same bank.
  - Each I/O lane in the bank also has its own internal V_{REF} generator. You can configure each I/O lane independently to use its internal V_{REF} or the I/O bank's external V_{REF} source. All I/O pins in the same I/O lane use the same V_{REF} source.
- You can use the internal V_{REF} with calibration to support DDR4 using the POD12 I/O standard. The internal VREF is supported only for the POD12 I/O standard.
- You can place any combination of input, output, or bidirectional pins near VREF pins. There is no VREF pin placement restriction.
- The VREF pins are dedicated for voltage-referenced single-ended I/O standards. You cannot use the VREF pins as user I/Os.

For more information about pin capacitance of the VREF pins, refer to the device datasheet.

Related Information

- I/O Standards Voltage Levels in Intel Arria 10 Devices on page 102
- I/O Standard Specifications, *Intel Arria 10 Device Datasheet*
  Lists the maximum and minimum input voltages (V_{IH} and V_{IL}), output voltages (V_{OH} and V_{OL}), and current drive characteristics (I_{OH} and I_{OL}) for various I/O standards supported by Intel Arria 10 devices.
- Pin Capacitance, *Intel Arria 10 Device Datasheet*
- Single-Ended I/O Standards Specifications, *Intel Arria 10 Device Datasheet*
- Single-Ended SSTL, HSTL, and HSUL I/O Reference Voltage Specifications, *Intel Arria 10 Device Datasheet*
- Single-Ended SSTL, HSTL, and HSUL I/O Standards Signal Specifications, *Intel Arria 10 Device Datasheet*
- Input Buffer Reference Voltage (VREF), *PHY Lite for Parallel Interfaces Intel FPGA IP Core User Guide: Intel Stratix® 10, Intel Arria 10, and Intel Cyclone® 10 GX Devices*
  Provides more information about VREF settings.
- I/O Bank Architecture in Intel Arria 10 Devices on page 121

5.7.1.2. Guideline: Observe Device Absolute Maximum Rating for 3.0 V Interfacing

To ensure device reliability and proper operation when you use the device for 3.0 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.

Single-Ended Transmitter Application

If you use the Intel Arria 10 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.

Single-Ended Receiver Application

If you use the Intel Arria 10 device as a receiver, use an external clamping diode to limit the overshoot and undershoot voltage at the I/O pins.

The 3.0 V I/O standard is supported using the bank supply voltage (V_{CCIO}) at 3.0 V and a V_{CCPT} voltage of 1.8 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

- I/O Standards Voltage Levels in Intel Arria 10 Devices on page 102

- I/O Standard Specifications, *Intel Arria 10 Device Datasheet*
  Lists the maximum and minimum input voltages (V_{IH} and V_{IL}), output voltages (V_{OH} and V_{OL}), and current drive characteristics (I_{OH} and I_{OL}) for various I/O standards supported by Intel Arria 10 devices.
- Absolute Maximum Ratings, *Intel Arria 10 Device Datasheet*
- Maximum Allowed Overshoot and Undershoot Voltage, *Intel Arria 10 Device Datasheet*

5.7.1.3. Guideline: I/O Standards Supported for I/O PLL Reference Clock Input Pin

The I/O PLL reference clock (REFCLK) input pin supports the following I/O standards only:

- Single-ended I/O standards
- LVDS

Intel Arria 10 devices support Differential HSTL and Differential SSTL input operation using LVDS input buffers. To support the electrical specifications of Differential HSTL or Differential SSTL signaling, assign the LVDS I/O standard to the REFCLK pin in the Intel Quartus Prime software.

5.7.2. 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.

5.7.2.1. Non-Voltage-Referenced I/O Standards

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 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 standard inputs and outputs, and 3.0 V LVCMOS inputs only.

5.7.2.2. Voltage-Referenced I/O Standards

To accommodate voltage-referenced I/O standards:

- Each Intel Arria 10 FPGA I/O bank contains a dedicated VREF pin.
- Each bank can have only a single V_{CCIO} voltage level and a single voltage reference (V_{REF}) level.

The voltage-referenced input buffer is powered by V_{CCPT}. Therefore, an I/O bank featuring single-ended or differential standards can support different voltage-referenced standards under the following conditions:

- The V_{REF} are the same levels.
- On-chip parallel termination (R_T OCT) is disabled.

If you enable R_T OCT, the voltage for the input standard and the V_{CCIO} of the bank must match.

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.

5.7.2.3. 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 1.5 V HSTL I/O standards with a 1.5 V V_{CCIO} and 0.75 V V_{REF}.

5.7.3. Guideline: Maximum Current Driving I/O Pins While Turned Off and During Power Sequencing

The following guideline applies to all I/O standards supported by the LVDS I/O banks, including single-ended and differential I/Os. This guideline is not applicable to the 3 V I/O and transceiver pins.

While the device is turned off, or during power up and power down conditions:

- Keep the maximum current driving through any I/O pin at 10 mA or less.
- Keep the total current driving through an I/O bank at no more than 100 mA.
- Ensure that the voltage level does not exceed 1.89 V.

Related Information

LVDS I/O Pin Guidance for Unpowered FPGA, *Power Sequencing Considerations for Intel Cyclone® 10 GX, Intel Arria 10, and Intel Stratix® 10 Devices*

5.7.4. Guideline: Using the I/O Pins in HPS Shared I/O Banks

In Intel Arria 10 SX devices, I/O banks 2K, 2J, and 2I connect the HPS to an SDRAM device through a dedicated HPS external memory interface.

Each of the I/O bank has four lanes:

- Lane 3—IO[47..36]
- Lane 2—IO[35..24]
- Lane 1—IO[23..12]
- Lane 0—IO[11..0]

If you do not include any HPS external memory interface in your system, you can use banks 2K, 2J, and 2I in the Intel Arria 10 SX device as FPGA GPIOs.

If you include an HPS external memory interface in your system, adhere to these guidelines if you want to use the unused pins in banks 2K, 2J, and 2I for FPGA GPIOs:

- Bank 2K is used for SDRAM ECC, and address and command signals:
  - Lane 3 is used for SDRAM ECC signals. You can use the remaining pins in this lane for FPGA inputs only.
  - Lanes 2, 1, and 0 are used for SDRAM address and command signals. You can use the remaining pins in these lanes for FPGA inputs and outputs.
- Bank 2J is used for SDRAM data signals [31..0] and bank 2I is used for SDRAM data signals [63..32].
  - 16 bits data width—two lanes of bank 2J is used for data. You can use the remaining pins in these two data lanes as FPGA inputs only. You can use the pins in the other two lanes of bank 2J, and all lanes of bank 2I as FPGA inputs or outputs.
  - 32 bits data width—you can use the remaining pins in all lanes of bank 2J as FPGA inputs only. You can use the pins in all lanes of bank 2I as FPGA inputs and outputs.
  - 64 bits data width—you can use the remaining pins in all lanes of banks 2J and 2I as FPGA inputs only.

5.7.5. Guideline: Maximum DC Current Restrictions

There are no restrictions on the maximum DC current for any number of consecutive I/O pins for Intel Arria 10 devices.

Intel Arria 10 devices conform to the V_{CCIO} Electro-Migration (EM) rule and IR drop targets for all I/O standard drive strength settings—ensuring reliability over the lifetime of the devices.

5.7.6. Guideline: Intel FPGA LVDS SERDES IP Core Instantiation

In DPA or soft-CDR mode, you can instantiate only one Intel FPGA LVDS SERDES IP core instance for each I/O bank.

Related Information

- I/O Banks for Intel Arria 10 GX Devices on page 113
- I/O Banks for Intel Arria 10 GT Devices on page 116
- I/O Banks for Intel Arria 10 SX Devices on page 117

5.7.7. Guideline: LVDS SERDES Pin Pairs for Soft-CDR Mode

You can use only specific LVDS pin pairs in soft-CDR mode. Refer to the pinout file of each device to determine the LVDS pin pairs that support the soft-CDR mode.

Related Information

- Intel Arria 10 Device Pin-Out Files
  Provides the pin-out file for each Intel Arria 10 device. For the SoC devices, the pin-out files also list the I/O banks that are shared by the FPGA fabric and the HPS.
- Soft-CDR Mode on page 159
- Periphery Clock Networks on page 76
  Provides more information about PCLK networks.

5.7.8. Guideline: Minimizing High Jitter Impact on Intel Arria 10 GPIO Performance

In your Intel Arria 10 design flow, follow these guidelines to minimize undesired jitter impact on the GPIO performance.

- Perform power delivery network analysis using Intel PDN tool 2.0. This analysis helps you to design a robust and efficient power delivery networks with the necessary decoupling capacitors. Use the Intel Arria 10 Early Power Estimator (EPE) to determine the current requirements for V_{CC} and other power supplies. Perform the PDN analysis based on the current requirements of all the power supply rails especially the V_{CC} power rail.
- Use voltage regulator with remote sensor pins to compensate for the DC IR drop associated with the PCB and device package from the V_{CC} power supply while maintaining the core performance. For more details about the connection guideline for the differential remote sensor pins for V_{CC} power, refer to the pin connection guidelines.
- The input clock jitter must comply with the Intel Arria 10 PLL input clock cycle-to-cycle jitter specification to produce low PLL output clock jitter. You must supply a clean clock source with jitter of less than 120 ps. For details about the recommended operating conditions, refer to the PLL specifications in the device datasheet.
- Use dedicated PLL clock output pin to transmit clock signals for better jitter performance. The I/O PLL in each I/O bank supports two dedicated clock output pins. You can use the PLL dedicated clock output pin as a reference clock source for the FPGA. For optimum jitter performance, supply an external clean clock source. For details about the jitter specifications for the PLL dedicated clock output pin, refer to the device datasheet.

- If the GPIO is operating at a frequency higher than 250 MHz, use terminated I/O standards. SSTL, HSTL, POD and HSUL I/O standards are terminated I/O standards. Intel recommends that you use the HSUL I/O standard for shorter trace or interconnect with a reference length of less than two inches.
- Implement the GPIO or source synchronous I/O interface using the Altera PHYLite for Parallel Interfaces IP core. Intel recommends that you use the Altera PHYLite for Parallel Interfaces IP core if you cannot close the timing for the GPIO or source-synchronous I/O interface for data rates of more than 200 Mbps. For guidelines to migrate your design from the Altera GPIO IP core to the Altera PHYLite for Parallel Interfaces IP core, refer to the related information.
- Use the small periphery clock (SPCLK) network. The SPCLK network is designed for high speed I/O interfaces and provides the smallest insertion delay. The following list ranks the clock insertion delays for the clock networks, from the largest to the smallest:
  - Global clock network (GCLK)
  - Regional clock network (RCLK)
  - Large periphery clock network (LPCLK)
  - SPCLK

Related Information

- *Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines*
- *Intel Arria 10 Device Datasheet*
- *GPIO to PHYLite Design Migration Guidelines*

5.7.9. Guideline: Usage of I/O Bank 2A for External Memory Interfaces

Other than for general purpose I/O usages, Intel Arria 10 devices also use I/O bank 2A for operations related to device configuration. Because of the configuration-related usage, there are several guidelines that you must follow to use I/O bank 2A for external memory interfaces.

- Do not use I/O bank 2A's pins that are required for configuration-related operations as external memory interface pins, even after configuration is complete. For example:
  - Pins that are used for the Fast Passive Parallel (FPP) configuration bus
  - Pins that are used for Partial Reconfiguration control signals
- Ensure that the external memory interface I/O voltage is compatible with the configuration I/O voltage.
- Run the Intel Quartus Prime Fitter to determine if the placement of pins for external memory interfaces in your device is valid.

For more information about the configuration pins, refer to the "Configuration Function" column in the pin-out file for your device.

Related Information

- Intel Arria 10 Device Pin-Out Files
  Provides the pin-out file for each Intel Arria 10 device. For the SoC devices, the pin-out files also list the I/O banks that are shared by the FPGA fabric and the HPS.

- Configuration Schemes on page 225
- Device Configuration Pins on page 257
- I/O Standards and Drive Strength for Configuration Pins on page 258
- Memory Interfaces Support in Intel Arria 10 Device Packages on page 194

5.8. I/O and High Speed I/O in Intel Arria 10 Devices Revision History

Document Version	Changes
2020.06.30	Added a footnote to table *Intel Arria 10 I/O Standards Voltage Levels* in section *I/O Standards Voltage Levels in Intel Arria 10 Devices*.
2019.12.30	Updated the programmable pre-emphasis diagram to remove the word "peak-peak".
2019.05.06	Added a topic that provides a usage modes summary of the Intel Arria 10 LVDS SERDES.
2019.01.11	• Removed statement that says that the programmable V_{OD} value of "0" is not available for the LVDS I/O standard. • Removed `ext_loaden` signal in figures showing the LVDS receiver in soft-CDR mode. • Specified that connecting the Intel FPGA IOPLL loaden signal to the LVDS receiver `ext_loaden` signal is not required for LVDS receivers in soft-CDR mode. • Updated the V_{REF} sources and pins guideline to specify that the internal V_{REF} is supported only for the POD12 I/O standard.
2018.08.28	• Removed the *MultiVolt I/O Interface in Intel Arria 10 Devices* topic. • Updated the *I/O Standards Voltage Levels in Intel Arria 10 Devices* topic to add information about interfacing with systems of different voltages.
2018.04.17	Updated the table listing the OCT schemes supported in Intel Arria 10 devices to specify that 3 V I/O and HPS I/O do not support bidirectional OCT.
2018.03.09	• Changed "logic-to-pin" to "logic to the output buffer" in the topic about programmable open-drain output. • In the guideline topic about pin placement for differential channels, clarified that in an I/O bank where a DPA channel is enabled, you can assign unused pins an I/O standard that has the same V_{CCIO} as the I/O bank. • Added link to the *Intel FPGA PHYLite for Parallel Interfaces IP Core User Guide* for more information about using internal V_{REF} for the POD 12 I/O standards. • Removed the guideline topic about not driving I/O pins during power sequencing. • Added guideline topic about maximum currents driving through I/O pins on LVDS I/O banks while turned off and during power sequencing. • Changed the term "modular I/O bank" to "I/O bank". In Intel Arria 10 devices, all I/O banks are modular.

Date	Version	Changes
December 2017	2017.12.15	• Added SSTL-12, SSTL-125, SSTL135, Differential SSTL-12, Differential SSTL-125, and Differential SSTL-135 I/O standards into *Supported I/O Standards in FPGA I/O for Intel Arria 10 Devices* and *Intel Arria 10 I/O Standards Voltage Levels* tables. • Removed DDR3 OCT Setting from *Programmable Current Strength Settings for Intel Arria 10 Devices* table and added a note to refer to *On-Chip I/O Termination in Intel Cyclone® 10 GX Devices* section for I/O standards with DDR3 OCT Setting. • Added a note to the topic about the open-drain output to specify that you must not pull the output voltage higher than the Vi (DC) level. • Updated the table listing the programmable current strength to update the 3.0 V LVTTL current strength settings—added 24 mA and 20 mA, and specified 16 mA as the default setting.

continued...

Date	Version	Changes
		• Updated the note about driving LVDS channels with the PLL in integer PLL mode to clarify that you do not need a PLL if you bypass the SERDES. • Updated the topic about the serializer bypass for DDR and SDR operation to add more information about clocks to the IOE. • Updated the topic about the deserializer to add more information about bypassing the deserializer. • Removed the statement about SDR and DDR data width from the figures that show the receiver datapath in non-DPA, DPA, and soft-CDR modes. • Corrected typographical error in the example showing the parameter values to generate output clock in external PLL mode by updating "c0" to "outclk0". • Updated the figure titles in the topic about LVPECL termination to clarify that the figures refer to external termination. There is no OCT support for LVPECL I/O standard. • Updated the RSKM calculation example. • Updated several links and link titles. • Clarified that I/O banks used for differential receiver, the PLL can drive only the channels within the same I/O bank in *Guideline: Pin Placement for Differential Channels*. • Added Intel FPGA PHYLite for Parallel Interfaces IP core description in *FPGA I/O IP Cores for Intel Arria 10 Devices*. • Clarified that to utilize the I/O registers when implementing DDR circuitry, use the Intel FPGA GPIO IP core in *I/O Buffer and Registers in Intel Arria 10 Devices*. • Clarified that 3 V I/O bank supports single-ended and differential SSTL, HSTL, and HSUL I/O standards. • Clarified that all singled-ended I/O configured to 3 V I/O bank supports all programmable I/O elements except programmable pre-emphasis, R_D on-chip termination (OCT), calibrated R_S and R_T OCT, and internal V_{REF} generation. • Updated *I/O and Differential I/O Buffers in Intel Arria 10 Devices* topic that differential reference clock is supported for the I/O PLL that drives the SERDES. • Specified that VREF pins are dedicated for voltage-reference signal-ended I/O standards in *Guideline: V_{REF} Sources and VREF Pins*. • Clarified the type of I/O buffers available in Intel Arria 10 FPGA devices and Intel Arria 10 SoC devices in *I/O Standards and Voltage Levels in Intel Arria 10 Devices*..
May 2017	2017.05.08	• Updated the vertical migration table to remove vertical migration between Intel Arria 10 GX and Intel Arria 10 SX device variants. • Updated the topic about the LVDS interface with external PLL mode to clarify that the **Clock Resource Summary** tab in the LVDS SERDES IP core parameter editor provides the details for the signals required from the GPIO IP core. • Updated the table that lists the programmable IOE features supported by the I/O buffer types and I/O standards. • Removed all "Preliminary" marks.
March 2017	2017.03.15	Rebranded as Intel.

continued...

Date	Version	Changes
October 2016	2016.10.31	• Added information about the default predefined current strength if you do not specifically assign a current strength in the Intel Quartus Prime software. • Updated the topic about OCT calibration block to verify that you can calibrate the OCT using OCT calibration block in any I/O bank of the same I/O column. • Removed the F36 package from the Intel Arria 10 GX device family variant. • Updated the topic about receiver skew margin for non-DPA mode to clarify TCCS and RCCS usage in calculating the RSKM value. • Updated the guideline about not driving the I/O pins during power sequencing to stress that excess I/O pin current can affect device reliability and damage the device.
June 13	2016.06.13	• Updated the I/O vertical migration figure to add the KF40 package for the SX 570 and SX 660 devices. • Updated the table listing the I/O standards voltage levels to add 2.5 V input to 3.0 V LVTTL/3.0 V LVCMOS, and 3.0 V input to 2.5 V LVCMOS.
May 2016	2016.05.02	• Removed the NF40 and UF45 packages from the Intel Arria 10 GT device family variant. • Corrected the modular I/O banks information for the Intel Arria 10 GT 1150 device by updating the package from NF45 to SF45. • Updated the tables listing the I/O standards to clarify Class I and Class II support for SSTL-12, SSTL-125, SSTL-135, Differential SSTL-12, Differential SSTL-125, and Differential SSTL-135 I/O standards. • Corrected the table listing programmable IOE features to remove differential output voltage support for 3 V I/O banks. • Updated the list of programmable current strengths to add support for SSTL-135, SSTL-125, SSTL-12, POD-12, Differential SSTL-135, Differential SSTL-125, Differential SSTL-12, and Differential POD12 I/O standards. • Added 120 Ω OCT option for SSTL-12 and Differential SSTL-12 I/O standards. • Added guideline about clocking DPA interfaces that use more than 24 channels. • Added guideline about the I/O PLL reference clock source. • Added guideline about the I/O standards supported for the I/O PLL reference clock input pin. • Added guideline about using I/O pins in the HPS shared I/O banks. • Updated the maximum DC current restrictions guideline topic to specify that there are no restrictions for any number of consecutive I/O pins. • Updated the topics about using the LVDS interface with external PLL mode. The update adds examples and connection diagrams for using transmitter channels that span multiple banks and shared with receiver channels in DPA and soft-CDR modes. • Removed the restriction of using I/O bank 2A for external memory interfaces and added guidelines for using I/O bank 2A for external memory interfaces.
December 2015	2015.12.14	• Updated the table listing the I/O standards voltage support to remove 3.0 V V_{CCIO} input from the 2.5 V I/O standard. • Updated the topic about MultiVolt I/O interface to update V_{CCP} to V_{CC}. • Corrected the I/O standards supported for the open-drain output, bus-hold, and weak pull-up resistor features in the table summarizing the programmable IOE features. • Updated the topic about the data realignment block (bit slip) to specify that valid data is available four parallel clock cycles after the rising edge of `rx_bitslip_ctrl`. Previously, valid data is available after two parallel clock cycles. • Updated the topic about external I/O termination for devices to add footnotes about using OCT for SSTL-12 and Differential SSTL-12 I/O standards, and note about recommendation to perform IBIS or SPICE simulations.

continued...

Date	Version	Changes
		• Updated the topic about uncalibrated R_S OCT: — Updated the R_S values of SSTL-15 to remove 25 Ω and 50 Ω. — Added the Differential SSTL-15, Differential SSTL-135, Differential SSTL-125, Differential SSTL-12, Differential POD12, and Differential HSUL-12 I/O standards. • Updated the topic about calibrated R_S OCT to add the Differential POD12 I/O standard. • Updated the topic about calibrated R_T OCT to remove 20 Ω R_T OCT support and to add the Differential POD12 I/O standard. • Removed the Differential SSTL-2 Class I and Class II I/O standards from the tables listing the SERDES receiver and transmitter I/O standards support. • Updated the topic about the voltage-referenced I/O standard under the guideline for mixing voltage-referenced and non-voltage-referenced I/O standards. • Added design guideline for minimizing high jitter impact on the GPIO performance. • Updated the following signal names: — `dpa_diffioclk` to `dpa_fast_clock` — `dpa_load_en` to `dpa_load_enable`
November 2015	2015.11.02	• Updated the topic about serializer bypass for SDR and DDR operations to specify that the serializer bypass is supported through the Intel FPGA GPIO IP core. • Added a footnote with the definition of unit interval (UI) in the topic about the DPA block. • Updated the topic about the data realignment block (bit slip). The bit slip rollover value is now automatically set to the deserialization factor. • Updated the topic about the deserializer to specify that the deserializer bypass is supported through the Intel FPGA GPIO IP core. • Updated the topic about PLLs and clocking to correct the parallel clock names from `rx_outclock` and `tx_outclock` to `rx_coreclock` and `tx_coreclock`. • Updated the topic about using the PLLs in integer mode for LVDS to clarify that the I/O PLLs operate in integer mode only. • Updated the following port/signal names: — `rx_dpll_hold` to `rx_dpa_hold` — `rx_reset` to `rx_dpa_reset` — `rx_channel_data_align` to `rx_bitslip_ctrl` — `rx_cda_max` to `rx_bitslip_max` — `rx_outclock` to `rx_coreclock` — `lvds_diffioclk` and `diffioclk` to `fast_clock` — `lvds_load_en` and `load_en` to `load_enable` • Updated the topic about pin placement for differential channels: — Improved clarity about PLLs driving interleaved differential transmitter and DPA-enabled receiver channels — Removed the note about bank placement DDIO and SDR I/Os • Updated the topic about the signal interface between Intel FPGA IOPLL and the Intel FPGA LVDS SERDES IP core in external PLL mode. • Updated the topic about Intel FPGA IOPLL IP core parameter values for external PLL mode: — Phase shift of outclk0 from -180° to 180° — Phase shift of outclk2 from -180/serialization factor to 180/serialization factor (-18° to 18°) • Updated the definition of RSKM for the RSKM equation in the topic about the receiver skew margin in non-DPA mode. • Changed instances of *Quartus II* to *Quartus Prime*.

continued...

Date	Version	Changes
June 2015	2015.06.15	Corrected label for Intel Arria 10 GT product lines in the vertical migration figure.
May 2015	2015.05.04	• Updated the statements in the topic about the I/O and differential I/O buffers to improve clarity. • Updated the I/O resources information for the U19 package of the Intel Arria 10 GX 160, GX 220, SX 160, and SX 220 devices: — Updated LVDS I/O count from 144 to 148 — Updated total GPIO from 192 to 196 — Updated number of LVDS channels from 72 to 74 — Added bank 3A and removed bank 3C in the figures and related modular I/O banks tables • Updated the figure showing the IOE structure to clarify that the delay chains are separate. • Updated the modular I/Os for banks 3A (from null to 48) and 3B (from 48 to null) for the F27 package of the Intel Arria 10 GX 270, GX 320, SX 270, and SX 320 devices.
January 2015	2014.01.23	• Added topic about programmable open-drain output. • Restructured the topic about pin placement for differential channels to enhance clarity. • Corrected contents that specified DPA-enabled transmitter channels. There is no DPA for transmitter channels. • Added guideline about instantiating only one Altera LVDS SERDES IP core instance for each I/O bank. • Added guideline about using only specific LVDS pin pairs in soft-CDR mode. • Updated the section that describes usage of the LVDS interface with external PLL: — Updated information about the required signals in Altera IOPLL and Altera LVDS SERDES IP cores. — Updated the examples of parameter values to generate output clocks using Altera IOPLL IP core. — Updated the LVDS clock phase relationship diagram for external PLL interface signals. — Updated the diagrams that show the connections between Altera IOPLL and Altera LVDS SERDES IP cores. • Added footnote to clarify that you can use pre-emphasis for LVDS and POD12 I/O standards. The POD12 I/O standard supports DDR4.
August 2014	2014.08.18	• Updated description of the 3 V I/O bank regarding support for programmable IOE features. • Added statement to clarify that apart from FPGA I/O buffers, the Intel Arria 10 SoC devices also contains HPS I/O buffers with different I/O standards support. • Separated I/O bank 2A in each I/O banks location figures to signify that it is not consecutive with other I/O banks. • Updated LVDS I/O and SERDES circuitry descriptions to clarify that each LVDS channel have built-in transmit SERDES and receive SERDES. • Removed reference to on-chip clamping diode. Arria 10 devices do not have on-chip clamping diode. Use an external clamping diode where applicable. • Added a related information link to the Arria 10 Transceiver PHY User Guide that describes the transceiver I/O banks locations. • Updated the I/O vertical migration figure to show vertical migration between Arria 10 GX and Arria 10 SX devices. • Updated all references to "megafunction" to "IP core".

continued...

Date	Version	Changes
		• Updated all references to "MegaWizard Plug-in Manager" to "parameter editor". • Updated all references to Altera PLL IP core to Altera IOPLL IP core. • Updated the signal names for using the LVDS interface with the External PLL mode: — `tx_inclock` and `rx_inclock` to `ext_fclk` — `tx_enable` `rx_enable` to `ext_loaden` — `rx_dpaclock` to `ext_vcoph[7..0]` — `rx_synclock` to `ext_coreclock`
December 2013	2013.12.02	Initial release.

6. External Memory Interfaces in Intel Arria 10 Devices

The efficient architecture of the Intel Arria 10 external memory interface allows you to fit wide external memory interfaces within the small modular I/O banks structure. This capability enables you to support a high level of system bandwidth.

Compared to previous generation Arria devices, the new architecture and solution provide the following advantages:

- Pre-closed timing in the controller and from the controller to the PHY.
- Easier pin placement.

For maximum performance and flexibility, the architecture offers hard memory controller and hard PHY for key interfaces.

Related Information

- Arria 10 Device Handbook: Known Issues
  Lists the planned updates to the *Intel Arria 10 Device Handbook* chapters.
- Arria 10 FPGA and SoC External Memory Resources
  Provides more resources about the Intel Arria 10 external memory solution.
- External Memory Interface Spec Estimator
  Provides a parametric tool that allows you to find and compare the performance of the supported external memory interfaces in Intel FPGAs.

6.1. Key Features of the Intel Arria 10 External Memory Interface Solution

- The solution offers completely hardened external memory interfaces for several protocols.
- The devices feature columns of I/Os that are mixed within the core logic fabric instead of I/O banks on the device periphery.
- A single hard Nios® II block calibrates all the memory interfaces in an I/O column.
- The I/O columns are composed of groups of I/O modules called I/O banks.
- Each I/O bank contains a dedicated integer PLL (IO_PLL), hard memory controller, and delay-locked loop.
- The PHY clock tree is shorter compared to previous generation Arria devices and only spans one I/O bank.
- Interfaces spanning multiple I/O banks require multiple PLLs using a balanced reference clock network.

Related Information

External Memory Interface Architecture of Intel Arria 10 Devices on page 210
Provides more information about the I/O columns and I/O banks architecture.

6.2. Memory Standards Supported by Intel Arria 10 Devices

The I/Os are designed to provide high performance support for existing and emerging external memory standards.

Table 71. Memory Standards Supported by the Hard Memory Controller

This table lists the overall capability of the hard memory controller. For specific details, refer to the External Memory Interface Spec Estimator and Intel Arria 10 Device Datasheet.

Memory Standard	Rate Support	Ping Pong PHY Support	Maximum Frequency (MHz)
DDR4 SDRAM	Quarter rate	Yes	1,067
		—	1,200
DDR3 SDRAM	Half rate	Yes	533
		—	667
	Quarter rate	Yes	1,067
		—	1,067
DDR3L SDRAM	Half rate	Yes	533
		—	667
	Quarter rate	Yes	933
		—	933
LPDDR3 SDRAM	Half rate	—	533
	Quarter rate	—	800

Table 72. Memory Standards Supported by the Soft Memory Controller

Memory Standard	Rate Support	Maximum Frequency (MHz)
RLDRAM 3 (12)	Quarter rate	1,200
QDR IV SRAM(12)	Quarter rate	1,067
QDR II SRAM	Full rate	333
	Half rate	633
QDR II+ SRAM	Full rate	333
	Half rate	633
QDR II+ Xtreme SRAM	Full rate	333
	Half rate	633

(12) Intel Arria 10 devices support this external memory interface using hard PHY with soft memory controller.

Table 73. Memory Standards Supported by the HPS Hard Memory Controller

The hard processor system (HPS) is available in Intel Arria 10 SoC devices only.

Memory Standard	Rate Support	Maximum Frequency (MHz)
DDR4 SDRAM	Half rate	1,200
DDR3 SDRAM	Half rate	1,067
DDR3L SDRAM	Half rate	933

Related Information

- External Memory Interface Spec Estimator
  Provides a parametric tool that allows you to find and compare the performance of the supported external memory interfaces in Intel FPGAs.
- Ping Pong PHY IP on page 209
  Provides a brief description of the Ping Pong PHY.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.3. External Memory Interface Widths in Intel Arria 10 Devices

The Intel Arria 10 devices can support the following external memory interface widths:

- Up to x144 interfaces for DDR4 and DDR3
- Up to x72 for RLDRAM 3 and QDR II+ Xtreme

Table 74. Required I/O Banks for Interface Widths

This table lists the number of I/O banks required to support different external memory interface widths. You must implement each single memory interface using the I/O banks in the same I/O column.

This table is a guideline and represents the worst-case scenario for these interface widths. Certain interfaces can be implemented using fewer I/Os and does not take up the full I/O bank.

Except for DDR4 interfaces, if the total number of address/command pins exceeds 36, you require one more I/O bank than the number listed in this table. For DDR4 interfaces, the additional I/O bank is required if the number of address/command pins exceeds 37.

Interface Width	Required Number of I/O Banks
x8	1
x16, x24, x32, x40	2
x48, x56, x64, x72	3
x80, x88, x96, x104	4
x112, x120, x128, x136	5
x144	6

6.4. External Memory Interface I/O Pins in Intel Arria 10 Devices

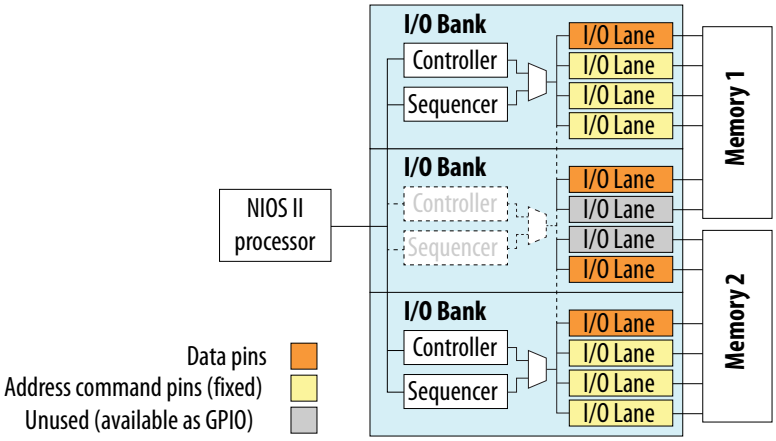
The memory interface circuitry is available in every I/O bank. The Intel Arria 10 devices feature differential input buffers for differential read-data strobe and clock operations.

The controller and sequencer in an I/O bank can drive address command (A/C) pins only to fixed I/O lanes location in the same I/O bank. The minimum requirement for the A/C pins are three lanes. However, the controller and sequencer of an I/O bank can drive data groups to I/O lanes in adjacent I/O banks (above and below).

Pins that are not used for memory interfacing functions are available as general purpose I/O (GPIO) pins.

Figure 124. I/O Banks Interface Sharing

This figure shows an example of two x16 interfaces shared by three I/O banks.



Related Information

External Memory Interface Architecture of Intel Arria 10 Devices on page 210
Provides more information about the I/O columns and I/O banks architecture.

6.4.1. Guideline: Usage of I/O Bank 2A for External Memory Interfaces

Other than for general purpose I/O usages, Intel Arria 10 devices also use I/O bank 2A for operations related to device configuration. Because of the configuration-related usage, there are several guidelines that you must follow to use I/O bank 2A for external memory interfaces.

- Do not use I/O bank 2A's pins that are required for configuration-related operations as external memory interface pins, even after configuration is complete. For example:
  - Pins that are used for the Fast Passive Parallel (FPP) configuration bus
  - Pins that are used for Partial Reconfiguration control signals
- Ensure that the external memory interface I/O voltage is compatible with the configuration I/O voltage.
- Run the Intel Quartus Prime Fitter to determine if the placement of pins for external memory interfaces in your device is valid.

For more information about the configuration pins, refer to the "Configuration Function" column in the pin-out file for your device.

Related Information

- Intel Arria 10 Device Pin-Out Files
  Provides the pin-out file for each Intel Arria 10 device. For the SoC devices, the pin-out files also list the I/O banks that are shared by the FPGA fabric and the HPS.
- Configuration Schemes on page 225
- Device Configuration Pins on page 257
- I/O Standards and Drive Strength for Configuration Pins on page 258
- Memory Interfaces Support in Intel Arria 10 Device Packages on page 194

6.5. Memory Interfaces Support in Intel Arria 10 Device Packages

Note: The number of I/O pins in an I/O bank and the availability of I/O banks vary across device packages. Each memory interface requires at least one I/O bank with 48 I/O pins for the A/C pins. I/O banks with less than 48 I/O pins can support data pins only. For details about the I/O banks available for each device package and the locations of consecutive I/O banks, refer to the related information.

Intel Arria 10 Package Support for DDR3 x40 with ECC on page 195

Intel Arria 10 Package Support for DDR3 x72 with ECC Single and Dual-Rank on page 197

Intel Arria 10 Package Support for DDR4 x40 with ECC on page 199

Intel Arria 10 Package Support for DDR4 x72 with ECC Single-Rank on page 201

Intel Arria 10 Package Support for DDR4 x72 with ECC Dual-Rank on page 203

HPS External Memory Interface Connections in Intel Arria 10 on page 204

Related Information

- GPIO Banks, SERDES, and DPA Locations in Intel Arria 10 Devices on page 105
- I/O Banks for Intel Arria 10 GX Devices on page 113
- I/O Banks for Intel Arria 10 GT Devices on page 116
- I/O Banks for Intel Arria 10 SX Devices on page 117
- Guideline: Usage of I/O Bank 2A for External Memory Interfaces on page 183

6.5.1. Intel Arria 10 Package Support for DDR3 x40 with ECC

To support one DDR3 x40 interface with ECC (32 bits data + 8 bits ECC), you require two I/O banks.

Table 75. Number of DDR3 x40 Interfaces (with ECC) Supported Per Device Package (without HPS Instance)

Note: For some device packages, you can also use the 3 V I/O banks for external memory interfaces. However, the maximum memory interface clock frequency is capped at 533 MHz. To use higher memory clock frequencies, exclude the 3 V I/O bank from external memory interfaces.

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	1	1	2	—	—	—	—	—	—	—	—
GX 220	1	1	2	—	—	—	—	—	—	—	—
GX 270	—	1	2	3	3	—	—	—	—	—	—
GX 320	—	1	2	3	3	—	—	—	—	—	—
GX 480	—	—	2	4	3	—	—	—	—	—	—
GX 570	—	—	—	4	3	5	6(13)	—	—	—	—
GX 660	—	—	—	4	3	5	6(13)	—	—	—	—
GX 900	—	—	—	4	—	5	—	1	7	6	4
GX 1150	—	—	—	4	—	5	—	1	7	6	4
GT 900	—	—	—	—	—	—	—	—	—	6	—
GT 1150	—	—	—	—	—	—	—	—	—	6	—
SX 160	1 (14)	1(14)	2(14)	—	—	—	—	—	—	—	—
SX 220	1(14)	1(14)	2 (14)	—	—	—	—	—	—	—	—
SX 270	—	1 (14)	2 (14)	3 (14)	3 (14)	—	—	—	—	—	—
SX 320	—	1 (14)	2 (14)	3 (14)	3 (14)	—	—	—	—	—	—
SX 480	—	—	2 (14)	4 (14)	3 (14)	—	—	—	—	—	—
SX 570	—	—	—	4 (14)	3 (14)	5 (14)	6(13) (14)	—	—	—	—
SX 660	—	—	—	4 (14)	3 (14)	5 (14)	6(13) (14)	—	—	—	—

(13) This number includes using the 3 V I/O bank for external memory interfaces. Otherwise, the number of external memory interfaces possible is reduced by one.

(14) This number includes HPS shared I/O banks to implement core EMIF configurations.

Table 76. Number of DDR3 x40 Interfaces (with ECC) Supported Per Device Package (with HPS Instance)

The number of supported interfaces shown in this table excludes the interface used to connect the HPS to external SDRAM. Masters in the FPGA core can access the HPS-connected external memory interface via FPGA-to-SDRAM bridge ports configurable in the HPS.

Note: For some device packages, you can also use the 3 V I/O banks for external memory interfaces. However, the maximum memory interface clock frequency is capped at 533 MHz. To use higher memory clock frequencies, exclude the 3 V I/O bank from external memory interfaces.

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	1	—	—	—	—	—	—	—	—
SX 220	0	0	1	—	—	—	—	—	—	—	—
SX 270	—	0	1	2	2	—	—	—	—	—	—
SX 320	—	0	1	2	2	—	—	—	—	—	—
SX 480	—	—	1	3	2	—	—	—	—	—	—
SX 570	—	—	—	3	2	4	4(15)	—	—	—	—
SX 660	—	—	—	3	2	4	4 (15)	—	—	—	—

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

(15) This number includes using the 3 V I/O bank for external memory interfaces. Otherwise, the number of external memory interfaces possible is reduced by one.

6.5.2. Intel Arria 10 Package Support for DDR3 x72 with ECC Single and Dual-Rank

To support one DDR3 x72 interface with ECC (64 bits data + 8 bits ECC) single and dual-rank, you require three I/O banks.

Table 77. Number of DDR3 x72 Interfaces (with ECC) Single and Dual-rank Supported Per Device Package (without HPS Instance)

Note: For some device packages, you can also use the 3 V I/O banks for external memory interfaces. However, the maximum memory interface clock frequency is capped at 533 MHz. To use higher memory clock frequencies, exclude the 3 V I/O bank from external memory interfaces.

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	1(16)	1(16)	1(16)	—	—	—	—	—	—	—	—
GX 220	1(16)	1(16)	1(16)	—	—	—	—	—	—	—	—
GX 270	—	1(16)	2(16)	2(16)	2(16)	—	—	—	—	—	—
GX 320	—	1(16)	2(16)	2(16)	2(16)	—	—	—	—	—	—
GX 480	—	—	2(16)	3(16)	2(16)	—	—	—	—	—	—
GX 570	—	—	—	3(16)	2(16)	3(16)	3	—	—	—	—
GX 660	—	—	—	3(16)	2(16)	3(16)	3	—	—	—	—
GX 900	—	—	—	3	—	3	—	0	4	3	2
GX 1150	—	—	—	3	—	3	—	0	4	3	2
GT 900	—	—	—	—	—	—	—	—	—	3	—
GT 1150	—	—	—	—	—	—	—	—	—	3	—
SX 160	1(16)(17)	1(16) (17)	1(16) (17)	—	—	—	—	—	—	—	—
SX 220	1(16)(17)	1(16) (17)	1(16) (17)	—	—	—	—	—	—	—	—
SX 270	—	1(16) (17)	2(16) (17)	2(16) (17)	2(16) (17)	—	—	—	—	—	—
SX 320	—	1(16) (17)	2(16) (17)	2(16) (17)	2(16) (17)	—	—	—	—	—	—
SX 480	—	—	2(16) (17)	3(16) (17)	2(16) (17)	—	—	—	—	—	—
SX 570	—	—	—	3(16) (17)	2(16) (17)	3(16) (17)	3 (17)	—	—	—	—
SX 660	—	—	—	3(16) (17)	2(16) (17)	3(16) (17)	3 (17)	—	—	—	—

(16) This number includes using the 3 V I/O bank for external memory interfaces. Otherwise, the number of external memory interfaces possible is reduced by one.

(17) This number includes HPS shared I/O banks to implement core EMIF configurations.

Table 78. Number of DDR3 x72 Interfaces (with ECC) Single and Dual-rank Supported Per Device Package (with HPS Instance)

The number of supported interfaces shown in this table excludes the interface used to connect the HPS to external SDRAM. Masters in the FPGA core can access the HPS-connected external memory interface via FPGA-to-SDRAM bridge ports configurable in the HPS.

Note: For some device packages, you can also use the 3 V I/O banks for external memory interfaces. However, the maximum memory interface clock frequency is capped at 533 MHz. To use higher memory clock frequencies, exclude the 3 V I/O bank from external memory interfaces.

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1 (18)	1 (18)	1 (18)	—	—	—	—	—	—
SX 320	—	0	1 (18)	1 (18)	1 (18)	—	—	—	—	—	—
SX 480	—	—	1 (18)	2 (18)	1 (18)	—	—	—	—	—	—
SX 570	—	—	—	2 (18)	1 (18)	2 (18)	2	—	—	—	—
SX 660	—	—	—	2 (18)	1 (18)	2 (18)	2	—	—	—	—

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

(18) This number includes using the 3 V I/O bank for external memory interfaces. Otherwise, the number of external memory interfaces possible is reduced by one.

6.5.3. Intel Arria 10 Package Support for DDR4 x40 with ECC

To support one DDR4 x40 interface with ECC (32 bits data + 8 bits ECC), you require two I/O banks.

Table 79. Number of DDR4 x40 Interfaces (with ECC) Supported Per Device Package (without HPS Instance)

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	1	1	2	—	—	—	—	—	—	—	—
GX 220	1	1	2	—	—	—	—	—	—	—	—
GX 270	—	1	2	3	3	—	—	—	—	—	—
GX 320	—	1	2	3	3	—	—	—	—	—	—
GX 480	—	—	2	4	3	—	—	—	—	—	—
GX 570	—	—	—	4	3	5	5	—	—	—	—
GX 660	—	—	—	4	3	5	5	—	—	—	—
GX 900	—	—	—	4	—	5	—	1	7	6	4
GX 1150	—	—	—	4	—	5	—	1	7	6	4
GT 900	—	—	—	—	—	—	—	—	—	6	—
GT 1150	—	—	—	—	—	—	—	—	7	6	—
SX 160	1 (19)	1 (19)	2 (19)	—	—	—	—	—	—	—	—
SX 220	1 (19)	1 (19)	2 (19)	—	—	—	—	—	—	—	—
SX 270	—	1 (19)	2 (19)	3 (19)	3	—	—	—	—	—	—
SX 320	—	1 (19)	2 (19)	3 (19)	3 (19)	—	—	—	—	—	—
SX 480	—	—	2	4(19)	3 (19)	—	—	—	—	—	—
SX 570	—	—	—	4 (19)	3 (19)	5 (19)	6(20)(19)	—	—	—	—
SX 660	—	—	—	4 (19)	3 (19)	5 (19)	6(20)(19)	—	—	—	—

(19) This number includes HPS shared I/O banks to implement core EMIF configurations.

(20) This number includes using the 3 V I/O bank for external memory interfaces. Otherwise, the number of external memory interfaces possible is reduced by one.

Table 80. Number of DDR4 x40 Interfaces (with ECC) Supported Per Device Package (with HPS Instance)

The number of supported interfaces shown in this table excludes the interface used to connect the HPS to external SDRAM. Masters in the FPGA core can access the HPS-connected external memory interface via FPGA-to-SDRAM bridge ports configurable in the HPS.

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	1	—	—	—	—	—	—	—	—
SX 220	0	0	1	—	—	—	—	—	—	—	—
SX 270	—	0	1	2	2	—	—	—	—	—	—
SX 320	—	0	1	2	2	—	—	—	—	—	—
SX 480	—	—	1	3	2	—	—	—	—	—	—
SX 570	—	—	—	3	2	4	4	—	—	—	—
SX 660	—	—	—	3	2	4	4	—	—	—	—

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Examples of External Memory Interface Implementations for DDR4
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.5.4. Intel Arria 10 Package Support for DDR4 x72 with ECC Single-Rank

To support one DDR4 x72 interface (64 bits data + 8 bits ECC) single-rank, you require three I/O banks.

Table 81. Number of DDR4 x72 Interfaces (with ECC) Single-Rank Supported Per Device Package (without HPS Instance)

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	0	0	0	—	—	—	—	—	—	—	—
GX 220	0	0	0	—	—	—	—	—	—	—	—
GX 270	—	0	1	1	1	—	—	—	—	—	—
GX 320	—	0	1	1	1	—	—	—	—	—	—
GX 480	—	—	1	2	1	—	—	—	—	—	—
GX 570	—	—	—	2	1	2	3	—	—	—	—
GX 660	—	—	—	2	1	2	3	—	—	—	—
GX 900	—	—	—	3	—	3	—	0	4	3	2
GX 1150	—	—	—	3	—	3	—	0	4	3	2
GT 900	—	—	—	—	—	—	—	—	—	3	—
GT 1150	—	—	—	—	—	—	—	—	—	3	—
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1 (21)	1 (21)	1 (21)	—	—	—	—	—	—
SX 320	—	0	1 (21)	1 (21)	1 (21)	—	—	—	—	—	—
SX 480	—	—	1 (21)	2 (21)	1 (21)	—	—	—	—	—	—
SX 570	—	—	—	2 (21)	1 (21)	2 (21)	3 (21)	—	—	—	—
SX 660	—	—	—	2 (21)	1 (21)	2 (21)	3 (21)	—	—	—	—

(21) This number includes HPS shared I/O banks to implement core EMIF configurations.

Table 82. Number of DDR4 x72 Interfaces (with ECC) Single-Rank Supported Per Device Package (with HPS Instance)

The number of supported interfaces shown in this table excludes the interface used to connect the HPS to external SDRAM. Masters in the FPGA core can access the HPS-connected external memory interface via FPGA-to-SDRAM bridge ports configurable in the HPS.

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1	1	1	—	—	—	—	—	—
SX 320	—	0	1	1	1	—	—	—	—	—	—
SX 480	—	—	1	2	1	—	—	—	—	—	—
SX 570	—	—	—	2	1	2	2	—	—	—	—
SX 660	—	—	—	2	1	2	2	—	—	—	—

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Examples of External Memory Interface Implementations for DDR4
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.5.5. Intel Arria 10 Package Support for DDR4 x72 with ECC Dual-Rank

To support one DDR4 x72 interface with ECC (64 bits data + 8 bits ECC) dual-rank, you require 3.25 I/O banks (three I/O banks and one I/O lane in an adjacent I/O bank).

Table 83. Number of DDR4 x72 Interfaces (with ECC) Dual-Rank Supported Per Device Package (without HPS Instance)

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
GX 160	0	0	0	—	—	—	—	—	—	—	—
GX 220	0	0	0	—	—	—	—	—	—	—	—
GX 270	—	0	1	1	1	—	—	—	—	—	—
GX 320	—	0	1	1	1	—	—	—	—	—	—
GX 480	—	—	1	1	1	—	—	—	—	—	—
GX 570	—	—	—	1	1	2	2	—	—	—	—
GX 660	—	—	—	1	1	2	2	—	—	—	—
GX 900	—	—	—	2	—	3	—	0	4	3	2
GX 1150	—	—	—	2	—	3	—	0	4	3	2
GT 900	—	—	—	—	—	—	—	—	—	3	—
GT 1150	—	—	—	—	—	—	—	—	—	3	—
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1 (22)	1 (22)	1 (22)	—	—	—	—	—	—
SX 320	—	0	1 (22)	1 (22)	1 (22)	—	—	—	—	—	—
SX 480	—	—	1 (22)	1 (22)	1 (22)	—	—	—	—	—	—
SX 570	—	—	—	1 (22)	1 (22)	2 (22)	2 (22)	—	—	—	—
SX 660	—	—	—	1 (22)	1 (22)	2 (22)	2 (22)	—	—	—	—

(22) This number includes HPS shared I/O banks to implement core EMIF configurations.

Table 84. Number of DDR4 x72 Interfaces (with ECC) Dual-Rank Supported Per Device Package (with HPS Instance)

The number of supported interfaces shown in this table excludes the interface used to connect the HPS to external SDRAM. Masters in the FPGA core can access the HPS-connected external memory interface via FPGA-to-SDRAM bridge ports configurable in the HPS.

Product Line	Package										
	U19	F27	F29	F34	F35	NF40	KF40	RF40	NF45	SF45	UF45
SX 160	0	0	0	—	—	—	—	—	—	—	—
SX 220	0	0	0	—	—	—	—	—	—	—	—
SX 270	—	0	1	1	1	—	—	—	—	—	—
SX 320	—	0	1	1	1	—	—	—	—	—	—
SX 480	—	—	1	1	1	—	—	—	—	—	—
SX 570	—	—	—	1	1	2	2	—	—	—	—
SX 660	—	—	—	1	1	2	2	—	—	—	—

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Examples of External Memory Interface Implementations for DDR4
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.5.6. HPS External Memory Interface Connections in Intel Arria 10

You must use the Intel Arria 10 External Memory Interfaces for HPS Platform Designer (Standard) IP component to connect external SDRAM to the HPS. You can instantiate the Intel Arria 10 External Memory Interfaces for HPS component in your Platform Designer (Standard) subsystem in addition to the HPS Platform Designer (Standard) component. You must connect the HPS component's EMIF conduit to the Intel Arria 10 External Memory Interfaces for HPS's EMIF conduit to connect the HPS to external SDRAM memory.

The HPS memory interface is fixed to I/O Banks 2Kand 2J for x40 widths and 2K, 2J, and 2I for x64/x72 widths. When an external SDRAM memory is connected to the HPS, there are restrictions on the availability of unused I/O to the FPGA core in the I/O banks (2K, 2J, 2I) utilized for the HPS memory interface.
When the HPS is connected to external SDRAM memory, no other Intel Arria 10 External Memory Interface IP instances can be placed in the same I/O column.

Related Information

External Memory Interface Handbook Volume 3: Reference Material - Functional Description - HPS Memory Controller
More detail regarding Intel Arria 10 EMIF Hard Processor Subsystem restrictions and placement information.

6.5.6.1. Intel Arria 10 Package Support for DDR3 x40 with ECC for HPS

To support one DDR3 x40 interface with ECC (32 bits data + 8 bits ECC) for HPS, you are required to use two I/O banks below the top 3 V I/O bank in the DDR column.

Table 85. Number of DDR3 x40 Interfaces (with ECC) for HPS Supported Per Device Package

This table lists the number of external memory interfaces supported for HPS only.

Product Line	Package						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	1	1	1	—	—	—	—
SX 220	1	1	1	—	—	—	—
SX 270	—	1	1	1	1	—	—
SX 320	—	1	1	1	1	—	—
SX 480	—	—	1	1	1	—	—
SX 570	—	—	—	1	1	1	1
SX 660	—	—	—	1	1	1	1

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.5.6.2. Intel Arria 10 Package Support for DDR3 x72 with ECC Single and Dual-Rank for HPS

To support one DDR3 x72 interface with ECC (64 bits data + 8 bits ECC) single and dual-rank for HPS, you are required to use three I/O banks below the top 3 V I/O bank in the DDR column.

Table 86. Number of DDR3 x72 Interfaces (with ECC) Single and Dual-Rank for HPS Supported Per Device Package

This table lists the number of external memory interfaces supported for HPS only.

Product Line	Package						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	0	0	0	—	—	—	—
SX 220	0	0	0	—	—	—	—
SX 270	—	0	0	0	0	—	—
SX 320	—	0	0	0	0	—	—
SX 480	—	—	0	0	0	—	—
SX 570	—	—	—	0	0	0	1
SX 660	—	—	—	0	0	0	1

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.5.6.3. Intel Arria 10 Package Support for DDR4 x40 with ECC for HPS

To support one DDR4 x40 interface with ECC (32 bits data + 8 bits ECC) for HPS, you are required to use two I/O banks below the top 3 V I/O bank in the DDR column.

Table 87. Number of DDR4 x40 Interfaces (with ECC) Supported Per Device Package for HPS

This table lists the number of external memory interfaces supported for HPS only.

Product Line	Package						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	1	1	1	—	—	—	—
SX 220	1	1	1	—	—	—	—
SX 270	—	1	1	1	1	—	—
SX 320	—	1	1	1	1	—	—
SX 480	—	—	1	1	1	—	—
SX 570	—	—	—	1	1	1	1
SX 660	—	—	—	1	1	1	1

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.5.6.4. Intel Arria 10 Package Support for DDR4 x72 with ECC Single-Rank for HPS

To support one DDR4 x72 interface with ECC (64 bits data + 8 bits ECC) single-rank for HPS, you are required to use three I/O banks below the top 3 V I/O bank in the DDR column.

Table 88. Number of DDR4 x72 Interfaces (with ECC) Single-Rank for HPS Supported Per Device Package

This table lists the number of external memory interfaces supported for HPS only.

Product Line	Package						
	U19	F27	F29	F34	F35	NF40	KF40
SX 160	0	0	0	—	—	—	—
SX 220	0	0	0	—	—	—	—
SX 270	—	0	0	0	0	—	—
SX 320	—	0	0	0	0	—	—
SX 480	—	—	0	0	0	—	—
SX 570	—	—	—	0	0	0	1
SX 660	—	—	—	0	0	0	1

Related Information

- Device Variants and Packages
  Provides more information about the device packages such as the types, sizes, and number of pins.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.
- Intel Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller
  Provides information on supported memory interface clock frequency per device speed grade.

6.6. External Memory Interface IP Support in Intel Arria 10 Devices

Table 89. Types of Intel FPGA IP Support for Each Memory Standard

This table lists the memory controller IP provided by Intel. You can also use your own soft memory controller for all memory standards supported by Intel Arria 10 devices.

Memory Standard	Controller		Hard Sequencer
	Hard	Soft	
DDR4 SDRAM[23]	Yes	—	Yes
DDR3 SDRAM [24]	Yes	—	Yes

continued...

(23) x4/x8 DQ group, POD12 I/O standard, and burst lengths BL8.

Memory Standard	Controller		Hard Sequencer
	Hard	Soft	
DDR3L SDRAM [24]	Yes	—	Yes
LPDDR3 SDRAM[25]	Yes	—	Yes
RLDRAM 3[26]	—	Yes	Yes
QDR IV SRAM[26]	—	Yes	Yes
QDR II/II+/II+ Xtreme SRAM	—	Yes	Yes

Related Information

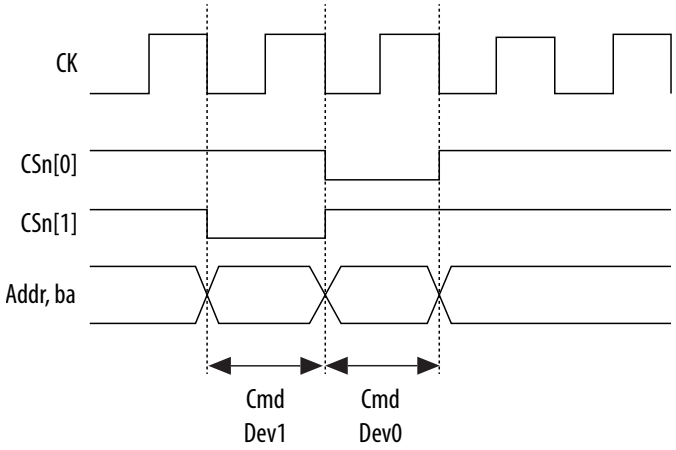
Memory Standards Supported by Intel Arria 10 Devices on page 191
Lists all memory standards that the Intel Arria 10 devices support.

6.6.1. Ping Pong PHY IP

The Ping Pong PHY IP allows two memory interfaces to share address/command buses using time multiplexing. The Ping Pong PHY IP gives you the advantage of using less pins compared to two independent interfaces, without any impact on throughput.

Figure 125. Ping Pong PHY 1T Timing

With the Ping Pong PHY, address and command signals from two independent controllers are multiplexed onto shared buses by delaying one of the controller outputs by one full-rate clock cycle. The result is 1T timing, with a new command being issued on each full-rate clock cycle.



Related Information

- Memory Standards Supported by Intel Arria 10 Devices on page 191
- Hard Memory Controller Features on page 212

(24) x4/x8 DQ group and burst lengths BL8.

(25) Intel Arria 10 devices support single component x32 data using x8 DQ group.

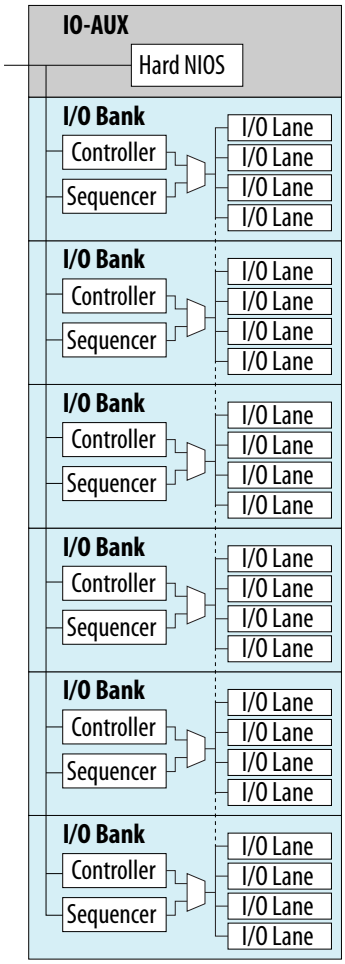
(26) Intel Arria 10 devices support this external memory interface using hard PHY with soft memory controller.

6.7. External Memory Interface Architecture of Intel Arria 10 Devices

The Intel Arria 10 external memory interface solution is designed to provide a high performance, rapid, and robust implementation of external memory interfacing. Instead of periphery I/Os like in the previous generation Arria devices, Intel Arria 10 devices feature columns of I/Os.

Figure 126. I/O Column Architecture

The I/O column consists of the I/O banks and an I/O-AUX block.



Related Information

- Key Features of the Intel Arria 10 External Memory Interface Solution on page 190
- External Memory Interface I/O Pins in Intel Arria 10 Devices on page 193

6.7.1. I/O Bank

The hard IP is organized into vertical I/O banks. These modular I/O banks can be stitched together to form large interfaces.

Each I/O bank consists of the following blocks:

- Embedded hard controller
- Hard sequencer
- Dedicated DLL
- Integer PLL
- OCT calibration block
- PHY clock network
- Four I/O lanes

6.7.1.1. Hard Memory Controller

The Intel Arria 10 hard memory controller is designed for high speed, high performance, high flexibility, and area efficiency. The hard memory controller supports all the popular and emerging memory standards including DDR4, DDR3, and LPDDR3.

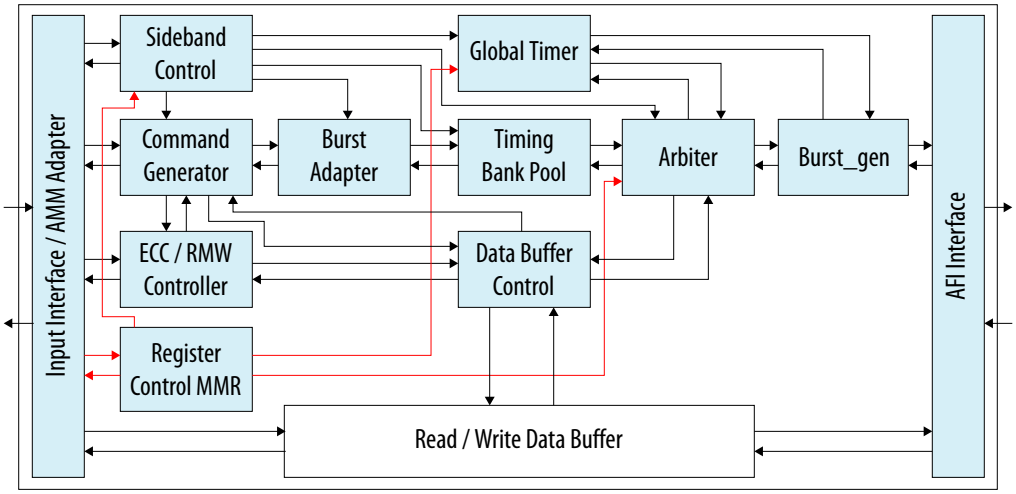
The high performance is achieved by implementing advanced dynamic command and data reordering algorithms. In addition, efficient pipelining techniques are also applied to the design to improve the memory bandwidth usage and reduce the latency while keeping the speed high. The hard solution offers the best availability and shorter time-to-market. The timing inside the controller and from the controller to the PHY have been pre-closed by Intel—simplifying timing closure.

The controller architecture is a modular design and fits in a single I/O bank. This structure offers you the best flexibility from the hard solution:

- You can configure each I/O bank as either one of the following paths:
  - A control path that drives all the address/command pins for the memory interface.
  - A data path that drives up to 32 data pins for DDR-type interfaces.
- You can place your memory controller in any location.
- You can pack up multiple banks together to form memory interfaces of different widths up to 144 bits.

For more flexibility, you can bypass the hard memory controller and use your custom IP if required.

Figure 127. Hard Memory Controller Architecture



The hard memory controller consists of the following logic blocks:

- Core and PHY interfaces
- Main control path
- Data buffer controller
- Read and write data buffers

The core interface supports the Avalon® Memory-Mapped (Avalon-MM) interface protocol. The interface communicating to the PHY follows the Altera PHY Interface (AFI) protocol. The whole control path is split into the main control path and the data buffer controller.

6.7.1.1.1. Hard Memory Controller Features

Table 90. Features of the Intel Arria 10 Hard Memory Controller

Feature	Description
Memory devices support	Supports the following memory devices: • DDR4 SDRAM • DDR3 SDRAM • LPDDR3 for low power
Memory controller support	• Custom controller support—configurable bypass mode that allows you to bypass the hard memory controller and use custom controller. • Ping Pong controller—allows two instances of the hard memory controller to time-share the same set of address/command pins.
Interface protocols support	• Supports Avalon-MM and Avalon-ST interfaces. • The PHY interface adheres to the AFI protocol.
Rate support	You can configure the controller to run at half rate or quarter rate.
Configurable memory interface width	Supports widths from 8 to 144 bits, in 8 bits increments.
Multiple ranks support	Supports up to 4 ranks.
	continued...

Feature	Description
Burst adaptor	Able to accept bursts of any size up to a maximum burst length of 127 on the local interface of the controller and map the bursts to efficient memory commands. *Note:* For applications that must strictly adhere to the Avalon-MM specification, the maximum burst length is 64.
Efficiency optimization features	• Open-page policy—by default, data traffic is closed-page on every access. However, the controller intelligently keep a row open based on incoming traffic, which can improve the efficiency of the controller especially for random traffic. • Pre-emptive bank management—the controller is able to issue bank management commands early, which ensure that the required row is open when the read or write occurs. • Data reordering—the controller reorders read/write commands. • Additive latency—the controller can issue a `READ`/`WRITE` command after the `ACTIVATE` command to the memory bank prior to t_{RCD}, which increases the command efficiency.
User requested priority	You can assign priority to commands. This feature allows you to specify that higher priority commands get issued earlier to reduce latency.
Starvation counter	Ensures all requests are served after a predefined time out period, which ensures that low priority access are not left behind while reordering data for efficiency.
Timing for address/command bus	To maximize command bandwidth, you can double the number of memory commands in one controller clock cycle: • Quasi-1T addressing for half-rate address/command bus. • Quasi-2T addressing for quarter-rate address/command bus.
Bank interleaving	Able to issue read or write commands continuously to "random" addresses. You must correctly cycle the bank addresses.
On-die termination	The controller controls the on-die termination signal for the memory. This feature improves signal integrity and simplifies your board design.
Refresh features	• User-controlled refresh timing—optionally, you can control when refreshes occur and this allows you to prevent important read or write operations from clashing with the refresh lock-out time. • Per-rank refresh—allows refresh for each individual rank. • Controller-controlled refresh.
ECC support	• 8 bit ECC code; single error correction, double error detection (SECDED). • User ECC supporting pass through user ECC bits as part of data bits.
Power saving features	• Low power modes (power down and self-refresh)—optionally, you can request the controller to put the memory into one of the two low power states. • Automatic power down—puts the memory device in power down mode when the controller is idle. You can configure the idle waiting time. • Memory clock gating.
Mode register set	Access the memory mode register.
DDR4 features	• Bank group support—supports different timing parameters for between bank groups. • Data Bus CRC—data bus encoding and decoding. • Command/Address parity—command and address bus parity check. • Alert reporting—responds to the error alert flag. • Multipurpose register access— supports multipurpose register access in serial readout mode. • Fine granularity refresh—supports 1x, 2x, and 4x fixed refresh rates.

continued...

Feature	Description
	• Temperature controlled refresh—adjust refresh rate according to temperature range. • Low power auto self refresh— operating temperature triggered auto adjustment to self refresh rate. • Maximum power saving.
LPDDR3 feature	• Deep power down mode—achieves maximum power reduction by eliminating power to memory array. Data is not retained when the device enters the deep power down mode. • Partial array self refresh. • Per bank refresh.
ZQ calibration command	Support long or short ZQ calibration command for DDR3 or DDR4.

Related Information

Ping Pong PHY IP on page 209
Provides a brief description of the Ping Pong PHY.

6.7.1.1.2. Main Control Path

The main control path performs the following functions:

- Contains the command processing pipeline.
- Monitors all the timing parameters.
- Keeps track of memory access commands dependencies.
- Guards against memory access hazards.

Table 91. Main Control Path Components

Component	Description
Input interface	• Accepts memory access commands from the core logic at half or quarter rate. • Uses the Avalon-MM or Avalon-ST protocol. The default protocol is Avalon-ST. You can enable a hard adapter through a configuration register to make the input interface Avalon-MM compatible. • The hard memory controller has a native Avalon-ST interface. You can instantiate a standard soft adaptor to bridge the Avalon-ST interface to AMBA AXI. • To support all bypass modes and keep the port count minimum, the super set of all port lists is used as the physical width. Ports are shared among the bypass modes.
Command generator and burst adapter	• Drains your commands from the input interface and feeds them to the timing bank pool. • If read-modify-write is required, inserts the necessary read-modify-write read and write commands into the stream. • The burst adapter chops your arbitrary burst length to the number specified by the memory types.
Timing Bank Pool	• Key component in the memory controller. • Sets parallel queues to track command dependencies. • Signals the ready status of each command being tracked to the arbiter for the final dispatch. • Big scoreboard structure. The number of entries is currently sized to 8 where it monitors up to 8 commands at the same time. • Handles the memory access hazards (RAW, WAR and WAW) while part of the timing constraints are being tracked. • High responsibility to assist the arbiter in implementing reordering: — Row command reordering (activate and pre-charge). — Column command reordering (read and write). • When the pool is full, a flow control signal is sent back upstream to stall the traffic.

continued...

Component	Description
Arbiter	• Enforces the arbitration rules. • Performs the final arbitration to select a command from all ready commands, and issues the selected command to the memory. • Supports quasi-1T mode for half rate and quasi-2T mode for quarter rate. • For the quasi modes, a row command must be paired with a column command.
Global Timer	Tracks the global timing constraints including: • t_{FAW}—the Four Activates Window parameter that specifies the time period in which only four activate commands are allowed. • t_{RRD}—the delay between back-to-back activate commands to different banks. • Some of the bus turnaround time parameters.
MMR/IOCSR	• The host of all the configuration registers. • Uses Avalon-MM bus to talk to the core. • Core logic can read and write all the configuration bits. • The debug bus is routed to the core through this block.
Sideband	Executes the refresh and power down features.
ECC controller	Although ECC encoding and decoding is performed in soft logic(27), the ECC controller maintains the read-modify-write state machine in the hard solution.
AFI interface	The memory controller communicates to the PHY using this interface.

6.7.1.1.3. Data Buffer Controller

The data buffer controller has the following main responsibilities:

- Manages the read and write access to the data buffers:
  - Provides the data storing pointers to the buffers when the write data is accepted or the read return data arrives.
  - Provides the draining pointer when the write data is dispatched to memory or the read data is read out of the buffer and sent back to users.
- Satisfies the required write latency.
- If ECC support is enabled, assists the main control path to perform read-modify-write.

Data reordering is performed with the data buffer controller and the data buffers.

Each I/O bank contains two data buffer controller blocks for the data buffer lanes that are split within each bank. To improve your timing, place the data buffer controller physically close to the I/O lanes.

6.7.1.2. Delay-Locked Loop

The delay-locked loop (DLL) finds the delay setting for 9 bits delay chain so that the delay of the chain is equivalent to one clock cycle.

Each I/O bank has one delay-locked loop (DLL) located in the center that supports a frequency range of 600 MHz to 1.3 GHz.

(27) ECC encoding and decoding is performed in soft logic to exempt the hard connection from routing data bits to a central ECC calculation location. Routing data to a central location removes the modular design benefits and reduces flexibility.

The reference clock for the DLL comes from the output of the PLL in the same I/O bank. The DLL divides the reference clock by eight and creates two clock pulses—`launch` and `measure`. The phase difference between `launch` and `measure` is one reference clock cycle. The clock pulse `launch` is routed through the delay setting controlled by the delay chain. The delayed `launch` is then compared to `measure`.

The setting for the DLL delay chains is from a 9 bit counter, which moves up or down to alter the delay time until the delayed `launch` and `measure` are aligned in the same phase. Once the DLL is locked, the delay through the delay chain is equivalent to one reference clock cycle, and the delay setting is sent out to the DQS delay block.

6.7.1.3. Sequencer

The sequencer enables high-frequency memory interface operation by calibrating the interface to compensate for variations in setup and hold requirements caused by transmission delays.

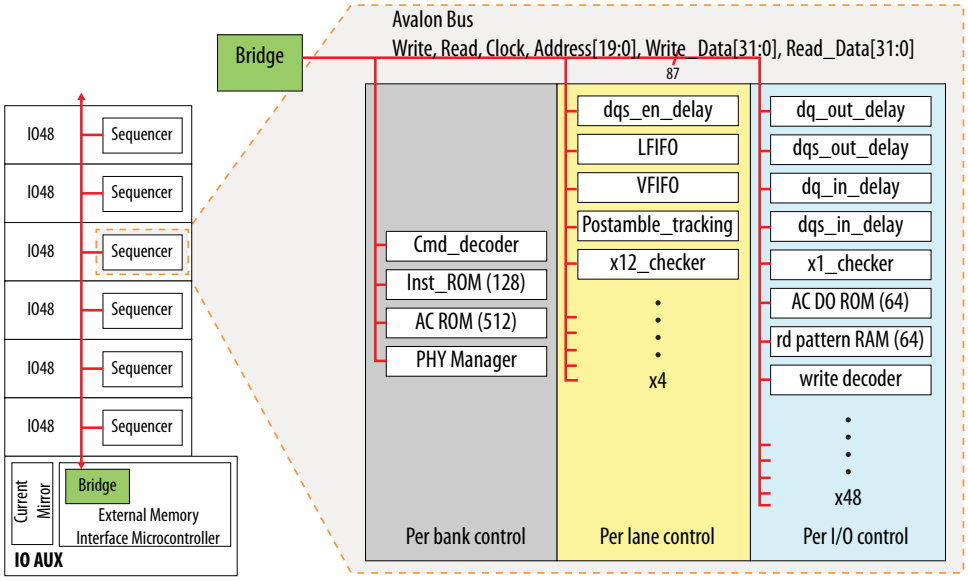
The sequencer implements a calibration algorithm to determine the combination of delay and phase settings that are necessary to maintain center-alignment of data and clock signals, even in the presence of significant delay variations. Programmable delay chains in the FPGA I/Os then implement the calculated delays to ensure that data remains centered.

A sequencer is embedded in every I/O bank. The sequencer is comprised of the following components:

- A read-write manager.
- An address/command set or instruction ROM.
- Helper modules such as PHY manager, data manager, and tracking manager.
- Data pattern and data out buffers on a per-pin basis that are managed by the read-write manager.

All major components of the sequencer are connected on the Avalon bus, providing controllability, visibility, and flexibility to the Nios II subsystem.

Figure 128. Sequencer



6.7.1.4. Clock Tree

The Intel Arria 10 external memory interface PHY clock network is designed to support the 1.2 GHz DDR4 memory standard.

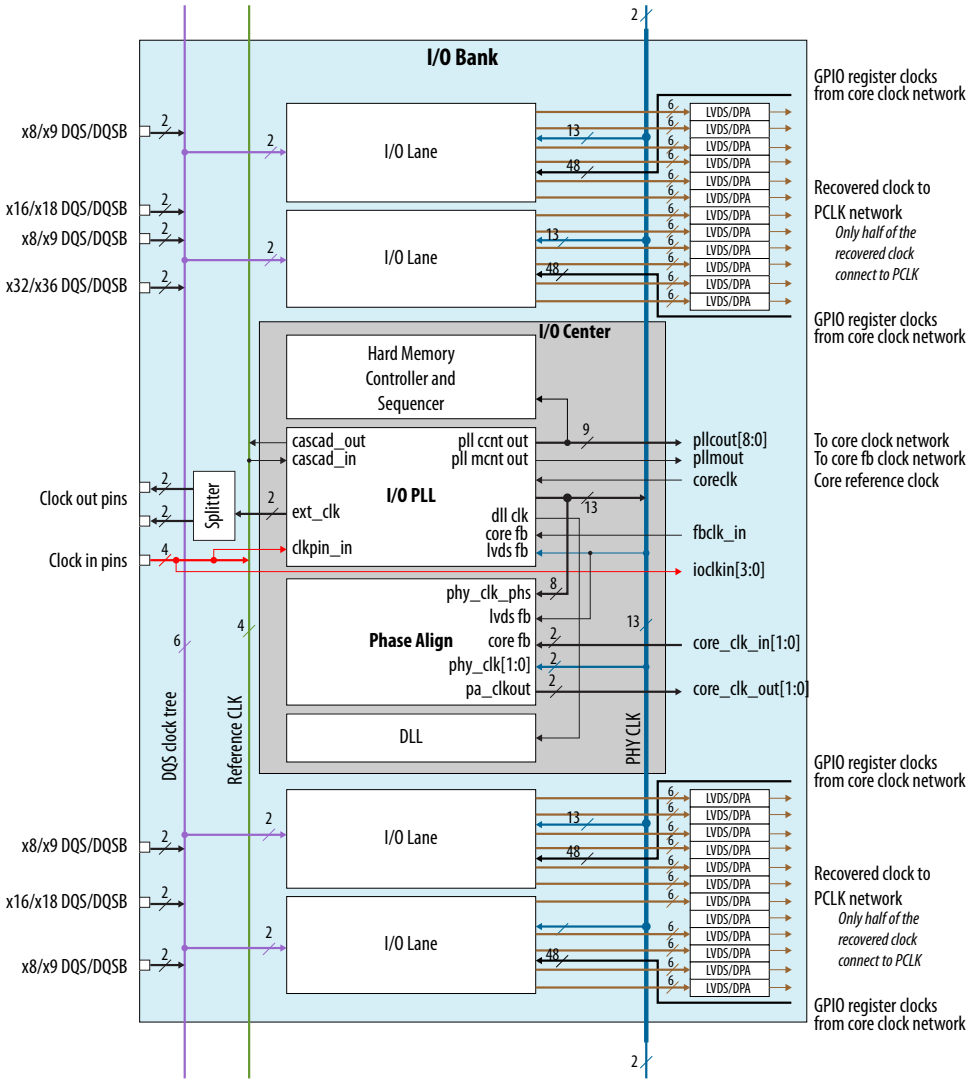
Compared to previous generation devices, the PHY clock network has a shorter clock tree that generates less jitter and less duty cycle distortion.

The PHY clock network consists of these clock trees:

- Reference clock tree
- PHY clock tree
- DQS clock tree

Figure 129. Clock Network Diagram

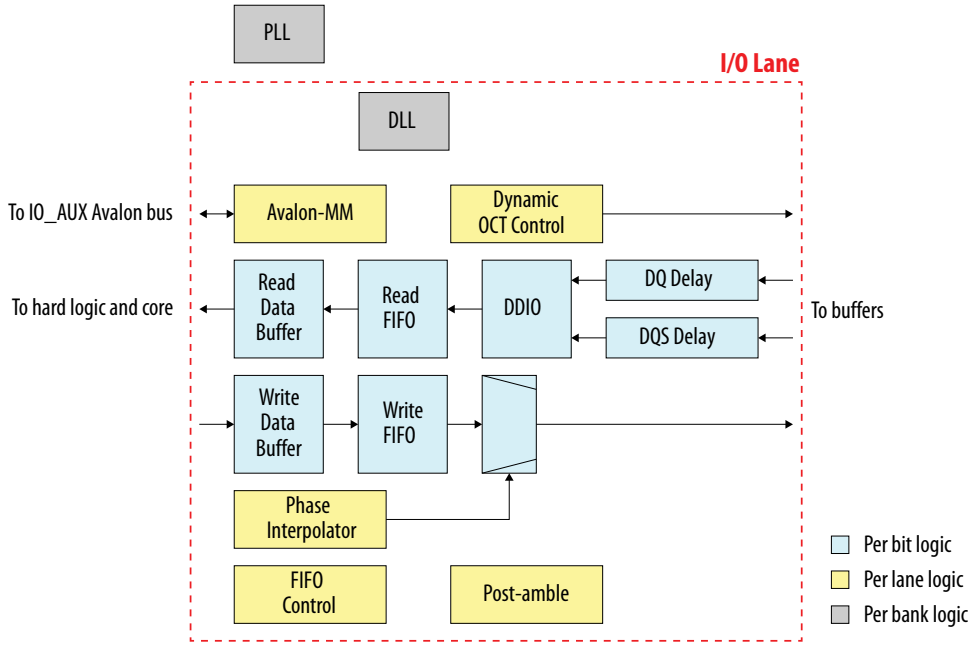
The reference clock tree adopts a modular design to facilitate easy integration.



6.7.1.5. I/O Lane

There are four I/O lanes in each I/O bank. Each I/O lane contains 12 I/O pins with identical read and write data paths and buffers.

Figure 130. I/O Lane Architecture



Data Path Component	Description
Input path	Contains capture registers and read FIFO.
Output or output enable (oe) path	Consists of: • Write FIFO • Clock mux • Phase interpolater— supports around 5 to 10 ps resolution based on frequency • Double data rate control
Input delay chain	Supports around 5 ps resolution with a delay range of 0 to 625 ps.
Read/write buffer	The write data buffer has built in options to take data from the core or from the hard memory controller.

Related Information

General Pin-Out Guidelines for Arria 10 EMIF IP

6.7.1.5.1. DQS Logic Block

The DQS logic block contains:

- Post-amble register
- DQS delay chain
- FIFO control
- Multi-rank switch control block

DQS Delay Chain

The DQS delay chain provides variable delay to the DQS signal, allowing you to adjust the DQS signal timing during calibration to maximize the t_{setup} and t_{hold} for DQ capture.

To keep the delay value constant, the DQS delay chain also contains:

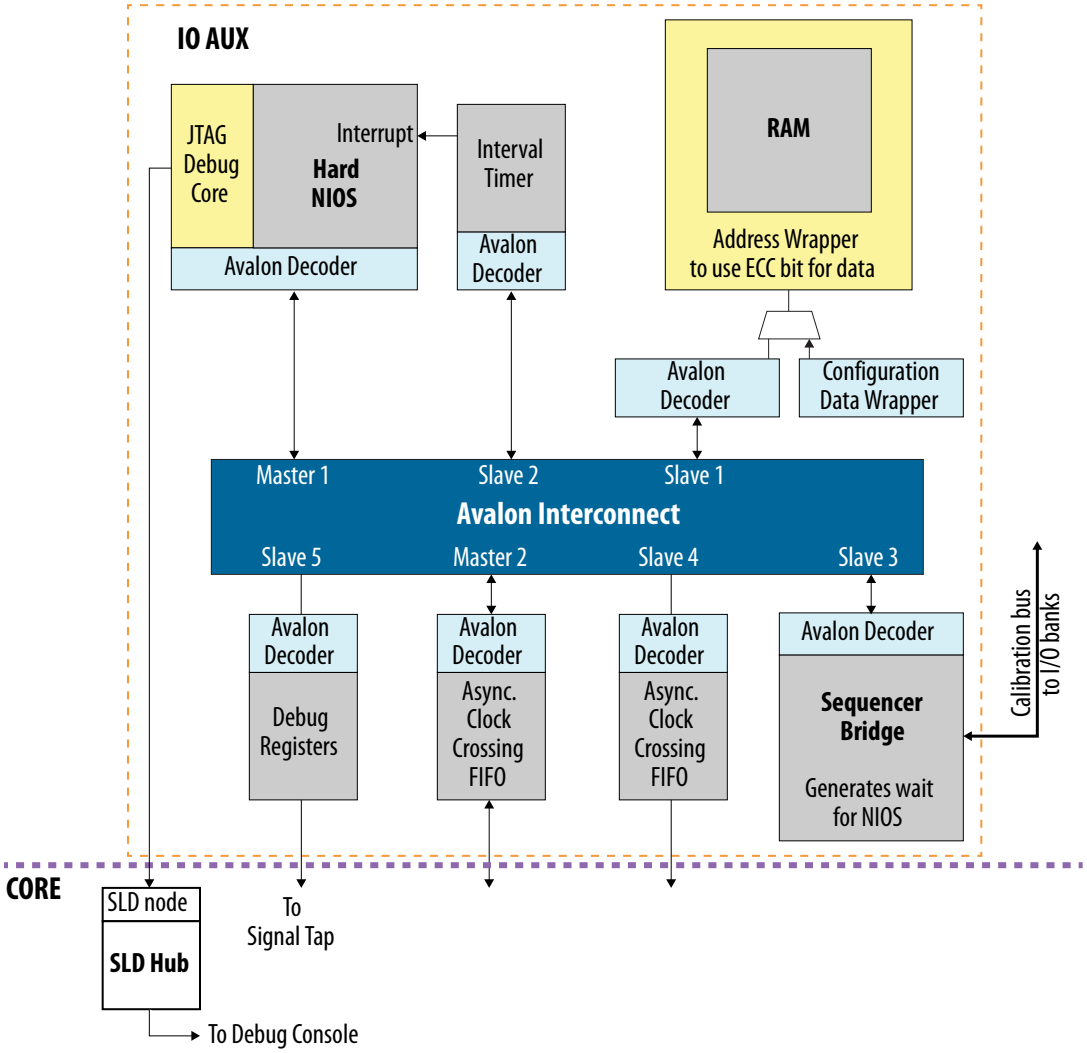
- Logic to track temperature and low frequency voltage variation
- Shadow registers to hold calibrated delay settings for multi-rank interfaces, and switch the DQS delay chain setting to one of up to four different settings.

6.7.2. I/O AUX

There is one I/O AUX block in each I/O column:

- Contains a hard Nios II processor and supporting embedded memory block
- Handles the calibration algorithm for the entire I/O column
- Communicates to the sequencer in each I/O bank through a dedicated Avalon interface

Figure 131. IO AUX Block Diagram



The hard Nios II processor performs the following operations:

- Configures and starts calibration tasks on the sequencers
- Collects and processes data
- Uses the final results to configure the I/Os

A combination of both Nios II code and the sequencers, the algorithm implementation supports calibration for the following memory interface standards:

- DDR2, DDR3, and DDR4 SDRAM
- QDR II and QDR IV SRAM
- RLDRAM 3
- LPDDR2 and LPDDR3

Note: Intel recommends that you use the Nios subsystem for memory interface calibration.

6.8. External Memory Interface in Intel Arria 10 Devices Revision History

Date	Version	Changes
December 2017	2017.12.15	• Updated Delay-Locked Loop frequency range to 600MHz - 1.3GHz. • Updated maximum burst length for burst adaptor in Table 90 on page 212 table.
June 2017	2017.06.21	Updated the note about the memory interfaces support to clarify that I/O banks with less than 48 pins can be used for data pins only. Therefore, all external memory interfaces require at least one 48-pins I/O bank to place the A/C pins.
March 2017	2017.03.15	• Removed Avalon Streaming (Avalon ST) interface protocol support for hard memory controller. • Rebranded as Intel.
October 2016	2016.10.31	Removed the F36 package from the Intel Arria 10 GX device family variant.
May 2016	2016.05.02	• Updated maximum frequency for QDR II, QDR II+ and QDR II+ Xtreme SRAM. • Updated maximum supported frequency for DDR4 SDRAM. • Removed NF40 and UF45 packages support for Arria 10 GT devices. • Added Guideline: Usage of I/O Bank 2A for External Memory Interfaces section in External Memory Interface I/O Pins in Intel Arria 10 Devices chapter. • Removed LPDDR3 support in HPS Hard Memory Controller. • Added HPS External Memory Interface Connections in Intel Arria 10 chapter to explain the restriction for using HPS EMIF with non-HPS EMIF within the same the device. • Updated number of interfaces supported for DDR4 x40 with ECC in F36 and KF40 packages (GX 570 and GX 660 devices). • Removed note and footnote about using 3 V I/O bank to support DDR4 x40 with ECC interfaces. • Added tables to show numbers of supported memory interfaces for Intel Arria 10 SX device packages when HPS EMIF instances are used within the same device. • Removed burst chop feature for DDR3 and DDR4 in Table Main Control Path Components. • Removed DDR4 gear down mode feature in Table Hard Memory Controller Features. • Removed DQS tracking feature in Hard Memory Controller in Table Hard Memory Controller Features.
November 2015	2015.11.02	• Removed BC4 and On-the-fly supports for DDR4, DDR3 and DDR3L SDRAM in Table Types of Altera IP Support for Each Memory Standard. • Change supported DQ Group for DDR4, DDR3, and DDR3L SDRAM to x4/x8 in Table Types of Altera IP Support for Each Memory Standard. • Added LPDDR3 SDRAM in hard memory controller and IP support. • Added link to Arria 10 Device Datasheet - Memory Standards Supported by the Hard Memory Controller and Arria 10 Device Datasheet - Memory Standards Supported by the Soft Memory Controller. • Added Intel Arria 10 package support for DDR3 x32 with ECC for HPS, DDR3 x 72 Single and Dual-Rank for HPS, DDR4 x32 with ECC for HPS, and DDR3 x72 Single-Rank tables. • Changed instances of *Quartus II* to *Quartus Prime*.
June 2015	2015.06.15	Removed the DFI label on the figure showing the hard memory controller architecture. Intel Arria 10 devices do not support DFI.
May 2015	2015.05.15	Corrected the DDR3 half rate and quarter rate maximum frequencies in the table that lists the memory standards supported by the Intel Arria 10 hard memory controller.

continued...

Date	Version	Changes
May 2015	2015.05.04	Updated the table that lists the memory standards supported by the hard memory controller in Intel Arria 10 devices.
January 2015	2015.01.23	• Updated the table that lists the memory standards supported by Intel Arria 10 devices. • Removed hard memory controller and IP support for LPDDR3 SDRAM. • Removed support for RLDRAM 2. • Updated support for QDR II+/II+ Xtreme SRAM to also include QDR II SRAM. • Added soft memory controller support for QDR IV. • Added footnote to clarify that the number of DDR4 x32 interfaces support for the F34 package of the Intel Arria 10 SX 480 device includes using I/O bank 2K. If you use I/O bank 2K in a DDR4 x32 interface for the FPGA, the HPS will not have access to a DDR4 x32 interface. • Added information to clarify that the DDR3 and DDR4 x32 interface with ECC includes 32 bits data and 8 bits ECC. • Removed information about hard and soft portions of the Nios subsytem. The hard memory controller IP for Intel Arria 10 calibrates the external memory interface using the hard Nios II processor only.
August 2014	2014.08.18	• Removed hard memory controller half rate support for DDR4 SDRAM. • Removed hard memory controller and IP support for DDR3U SDRAM. • Added soft memory controller full rate support for QDR II+ SRAM and QDR II+ Xtreme SRAM. • Updated the list of external memory standards supported by the HPS. • Updated the number of DDR3 x72 (single-rank) memory interfaces supported for the U19 package. • Removed the note about using 3 V I/O banks for the HPS. For the HPS, the 3 V I/O bank is not used for external memory interfaces. • Updated the number of DDR3 x72 (dual-rank) memory interfaces supported for the Arria 10 SX devices. • Updated the number of DDR4 x32 (with ECC) memory interfaces supported for the NF45 package of the Arria 10 GT 1150 device. • Added soft memory controller IP support for QDR II+ SRAM. • Added information to clarify that RLDRAM3 support uses hard PHY with soft memory controller. • Updated the table that lists the features of the hard memory controller to improve accuracy and add missing information. • Added a note before the topics listing external memory interface package support to clarify that not all I/O banks are available for external memory interfaces. • Moved the external memory interface pins guidelines and the examples of external memory interface implementations for DDR4 to the External Memory Interface Handbook.
December 2013	2013.12.10	Updated the HPS memory standards support from LPDDR2 to LPDDR3.
December 2013	2013.12.02	Initial release.

7. Configuration, Design Security, and Remote System Upgrades in Intel Arria 10 Devices

This chapter describes the configuration schemes, design security, and remote system upgrade that are supported by the Intel Arria 10 devices.

Related Information

- Intel Arria 10 Device Handbook: Known Issues
  Lists the planned updates to the Intel Arria 10 Device Handbook chapters.
- Intel Arria 10 Device Datasheet
  Provides more information about the estimated uncompressed .rbf file sizes, FPP DCLK-to-DATA[] ratio, and timing parameters for all supported configuration schemes.
- PLLs and Clock Networks Chapter of the Intel Arria 10 Transceiver PHY User Guide
  For more details about the need to configure unused transceiver channels when Intel Arria 10 devices are powered up to normal operating conditions.

7.1. Enhanced Configuration and Configuration via Protocol

Table 92. Configuration Schemes and Features of Intel Arria 10 Devices

Intel Arria 10 devices support 1.8 V programming voltage and several configuration schemes.

Scheme	Data Width	Max Clock Rate (MHz)	Max Data Rate (Mbps) (28)	Decompression	Design Security (29)	Partial Reconfiguration (30)	Remote System Update
JTAG	1 bit	33	33	—	—	Yes (31)	—
Active Serial (AS) through the EPCQ-L configuration device	1 bit, 4 bits	100	400	Yes	Yes	Yes (31)	Yes

continued...

(28) Enabling either compression or design security features affects the maximum data rate. Refer to the Intel Arria 10 Device Datasheet for more information.

(29) Encryption and compression cannot be used simultaneously.

(30) Partial reconfiguration is an advanced feature of the device family. If you are interested in using partial reconfiguration, contact Intel for support.

(31) Partial configuration can be performed only when it is configured as internal host.

Scheme	Data Width	Max Clock Rate (MHz)	Max Data Rate (Mbps) (28)	Decompression	Design Security (29)	Partial Reconfiguration (30)	Remote System Update
Passive serial (PS) through CPLD or external microcontroller	1 bit	100	100	Yes	Yes	Yes (31)	Parallel Flash Loader (PFL) Intel FPGA IP core
Fast passive parallel (FPP) through CPLD or external microcontroller	8 bits	100	3200	Yes	Yes	Yes (32)	PFL Intel FPGA IP core
	16 bits			Yes	Yes		
	32 bits			Yes	Yes		
Configuration via HPS	16 bits	100	3200	Yes	Yes	Yes (32)	—
	32 bits			Yes	Yes		
Configuration via Protocol [CvP (PCIe*)]	x1, x2, x4, x8 lanes	—	8000	Yes	Yes	Yes (31)	—

You can configure Intel Arria 10 devices through PCIe using Configuration via Protocol (CvP). The Intel Arria 10 CvP implementation conforms to the PCIe 100 ms power-up-to-active time requirement.

Related Information

Configuration via Protocol (CvP) Implementation in Intel FPGAs User Guide
Provides more information about the CvP configuration scheme.

7.2. Configuration Schemes

This section describes the AS, PS, FPP, and JTAG configuration schemes.

Related Information

- Design Planning for Partial Reconfiguration
  Provides more information about partial reconfiguration.
- Configuration via Protocol (CvP) Implementation in Intel FPGAs User Guide
  Provides more information about the CvP configuration scheme.

(28) Enabling either compression or design security features affects the maximum data rate. Refer to the Intel Arria 10 Device Datasheet for more information.

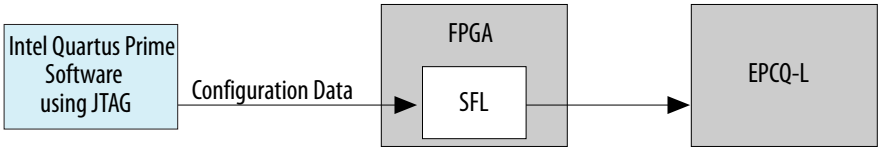
(29) Encryption and compression cannot be used simultaneously.

(30) Partial reconfiguration is an advanced feature of the device family. If you are interested in using partial reconfiguration, contact Intel for support.

(32) Supported at a maximum clock rate of 100 MHz.

7.2.1. Active Serial Configuration

Figure 132. High-Level Overview of EPCQ-L Programming for the AS Configuration Scheme



In the AS configuration scheme, configuration data is stored in the EPCQ-L configuration device. You can program the EPCQ-L device in-system using the JTAG interface with the Serial Flash Loader (SFL) IP core. The SFL acts as a bridge in the FPGA between the JTAG interface and the EPCQ-L device. The AS memory interface block in the Intel Arria 10 device controls the configuration process.

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 Intel Arria 10 device controls the configuration interface.

Note: For Active Serial programming using SFL, the MSEL pins must be set to Active Serial setting to allow the programmer to read the EPCQ-L ID.

Related Information

- Intel Arria 10 Device Datasheet
  Provides more information about the AS configuration timing.
- AN 370: Using the Serial Flash Loader with the Intel Quartus Prime Software
- Nios II Flash Programmer User Guide
- Intel Supported Configuration Devices, *Device Configuration - Support Center*
  Provides a list of Intel supported third party configuration devices, and a link to a list of Intel FPGA configuration devices and supported third party flash devices.
- EPCQ-L Serial Configuration Devices Datasheet
- EPCQ-L Device Package Information
  Provides more information about EPCQ-L packaging specifications, thermal resistance and dimensions.

7.2.1.1. DATA Clock (DCLK)

Intel Arria 10 devices generate the serial clock, DCLK, that provides timing to the serial interface. In the AS configuration scheme, Intel Arria 10 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. 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 Intel Arria 10 device determines the clock source and frequency to use by reading the option bit in the programming file.

Related Information

Intel Arria 10 Device Datasheet
Provides more information about the DCLK frequency specification in the AS configuration scheme.

7.2.1.2. Active Serial Single-Device Configuration

To configure Intel Arria 10 device, connect the device to a quad-serial configuration (EPCQ-L) device, as shown in the following figures.

Figure 133. Single Device AS x1 Mode Configuration

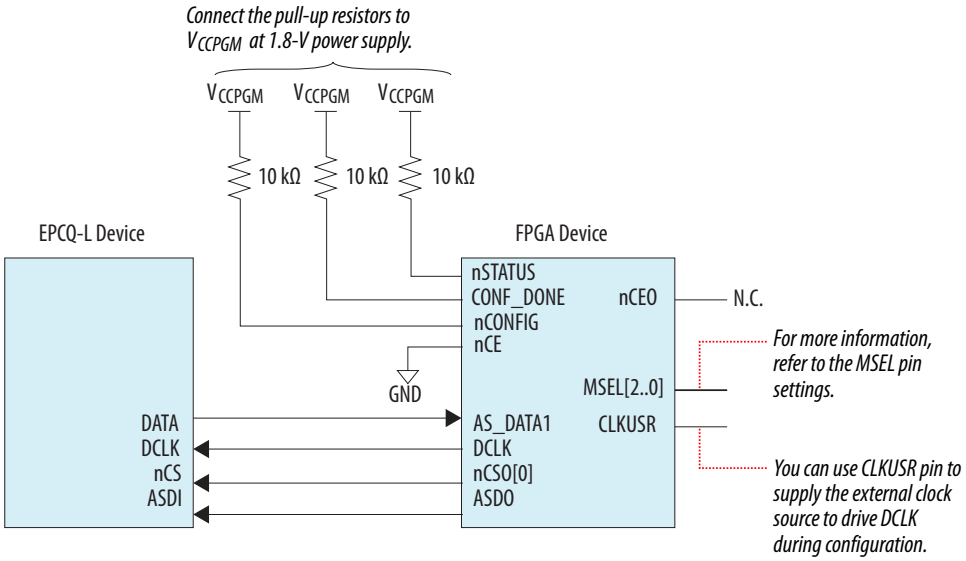
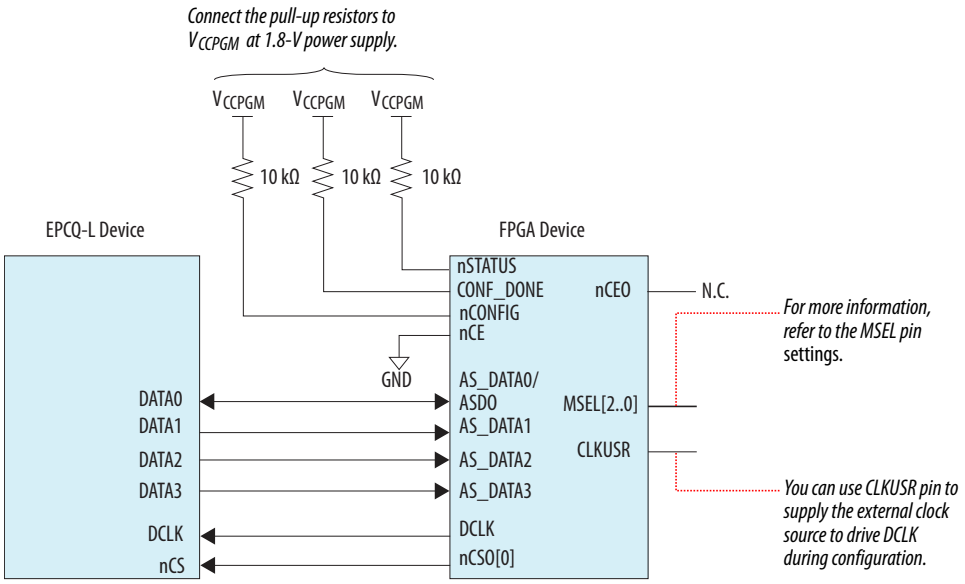


Figure 134. Single Device AS x4 Mode Configuration



7.2.1.3. Active Serial Multi-Device Configuration

You can configure multiple devices that are connected in 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.

When the active serial multi-device configuration scheme is used in Intel Arria 10 devices, the DCLK frequency for slave devices are determined as follows:

- When the configuration clock source is set to Internal Oscillator, the DCLK frequency for slave device is at the range for 12.5 MHz setting regardless of the setting of active serial clock source.
- When the configuration clock source is set to CLKUSR, the DCLK frequency for slave device follows CLKUSR clock frequency.

The **Configuration clock source** setting menu is located at **Assignments** > **Device** > **Device and Pin Options** > **General**.

The **Active serial clock source** setting menu is located at **Assignments** > **Device** > **Device and Pin Options** > **Configuration**.

For 12.5 MHz frequency range setting, refer to the *DCLK Frequency Specification in the AS Configuration Scheme* section in the *Intel Arria 10 Device Datasheet*.

Related Information

Intel Arria 10 Device Datasheet

7.2.1.3.1. Pin Connections and Guidelines

Observe the following pin connections and guidelines for this configuration setup:

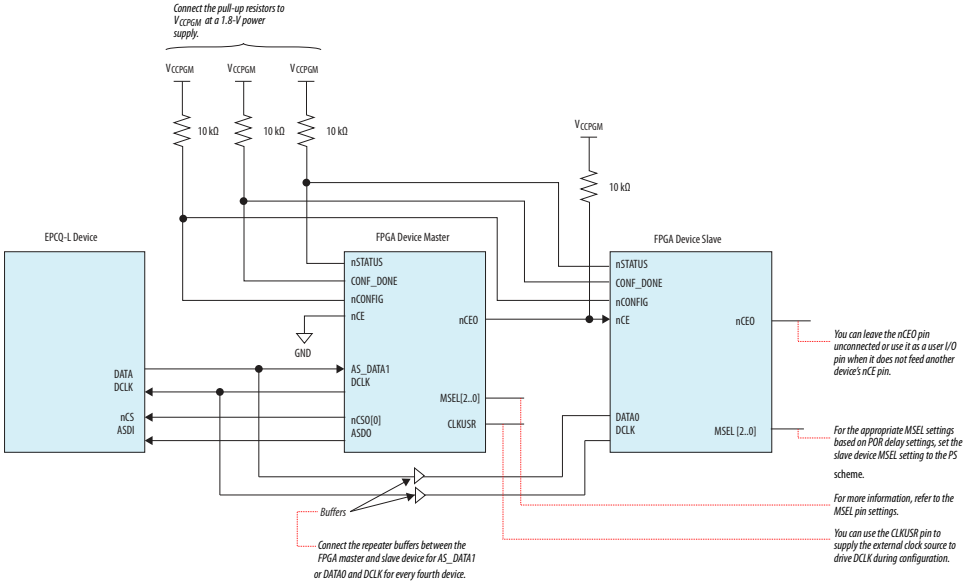
- Hardwire the MSEL pins of the first device in the chain to select the AS configuration scheme. For subsequent devices in the chain, hardwire their MSEL pins to select the PS configuration scheme. Any other Intel FPGAs that support the PS configuration can also be part of the chain as a configuration 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.

7.2.1.3.2. Using Multiple Configuration Data

To configure multiple Intel Arria 10 devices in a chain using multiple configuration data, connect the devices to an EPCQ-L device, as shown in the following figure.

Figure 135. Multiple Device AS Configuration When Both Devices in the Chain Receive Different Sets of Configuration Data



When a master device completes configuration, its nCE0 pin is released low to activate the nCE pin of the next device (for example, a slave device) in the chain. Configuration automatically begins for the slave device in one clock cycle.

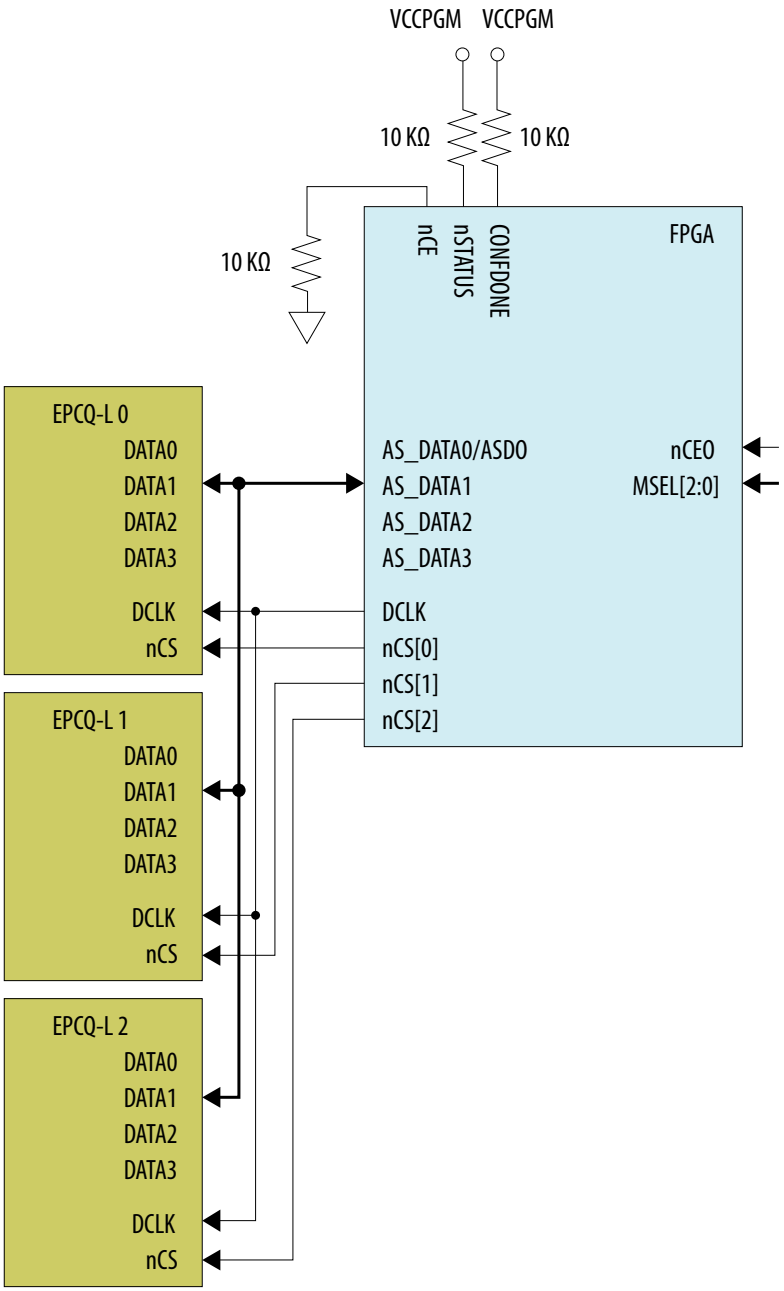
Set the master device MSEL pin setting to AS scheme and the slave device MSEL pin setting to PS scheme. If your slave device (for example, 28 nm devices such as Stratix® V, Arria V, and Cyclone V) supports configuration voltage (V_{CCPGM}) up to 3.0 V, Intel recommends that you send the V_{CCPGM} at 1.8 V so that a voltage translator is not needed.

7.2.1.4. Active Serial Configuration with Multiple EPCQ-L Devices

Intel Arria 10 devices support up to three EPCQ-L devices for configuration and remote system upgrade.

You can use up to three EPCQ-L devices per Intel Arria 10 device. Each EPCQ-L device gets a dedicated nCS0 pin, but shares other pins, as shown in the following figure.

Figure 136. AS Configuration with Multiple EPCQ-L Devices



You can choose the number of EPCQ-L devices using the Intel Quartus Prime software.

7.2.1.5. Using EPCQ-L Devices

EPCQ-L devices support AS x1 and AS x4 modes.

Note: Intel Arria 10 devices support EPCQ-L devices only.

Each Intel Arria 10 device has three nCSO pins—nCSO[2..0]. This allows Intel Arria 10 device to connect up to three EPCQ-L devices.

The advantages of connecting up to three EPCQ-L devices:

- Ability to store multiple design files for remote system upgrade.
- Increase storage beyond the largest single EPCQ-L device available.

Related Information

- EPCQ-L Serial Configuration Devices Datasheet
- EPCQ-L Device Package Information
  Provides more information about EPCQ-L packaging specifications, thermal resistance and dimensions.

7.2.1.5.1. Controlling EPCQ-L Devices

During configuration, Intel Arria 10 devices enable the EPCQ-L device by driving its nCSO output pin low, which connects to the chip select (nCS) pin of the EPCQ-L device. Intel Arria 10 devices use the DCLK and ASDO pins to send operation commands and read address signals to the EPCQ-L device. The EPCQ-L device provides data on its serial data output (DATA[]) pin, which connects to the AS_DATA[] input of the Intel Arria 10 devices.

Note: If you wish to gain control of the EPCQ-L 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.

7.2.1.5.2. Trace Length Guideline

The maximum trace length apply to both single- and multi-device AS configuration setups as listed in the following table. The trace length is the length from the Intel Arria 10 device to the EPCQ-L device.

Note: To evaluate the data setup (tSU) and data hold time (tDH) slack on your board in order to ensure that you are meeting the tSU and tDH requirements, Intel recommends that you follow the guideline in the *Evaluating Data Setup and Hold Timing Slack* section of the *AN822: Intel FPGA Configuration Device Migration Guideline*.

Table 93. Maximum Trace Length for AS x1 and x4 Configurations for Intel Arria 10 Devices

Intel Arria 10 Device AS Pins	Maximum Board Trace Length (Inches)	
	12.5/ 25/ 50 MHz	100 MHz
DCLK	10	6
AS_DATA[3..0]	10	6
nCSO[2..0]	10	6

Related Information

- AS Timing Parameters in Intel Arria 10 Device Datasheet
  Provides more information about data setup time and hold time requirement.
- AN822: Intel FPGA Configuration Device Migration Guideline

7.2.1.5.3. Programming EPCQ-L Devices

You can program EPCQ-L devices in-system using an Intel FPGA download cable. Alternatively, you can program the EPCQ-L using a microprocessor with the SRunner software driver.

In-system programming (ISP) offers you the option to program the EPCQ-L either using an AS programming interface or a JTAG interface. Using the AS programming interface, the configuration data is programmed into the EPCQ-L by the Intel Quartus Prime software or any supported third-party software. Using the JTAG interface, an Intel FPGA IP called the SFL IP core must be downloaded into the Intel Arria 10 device to form a bridge between the JTAG interface and the EPCQ-L. This allows the EPCQ-L to be programmed directly using the JTAG interface.

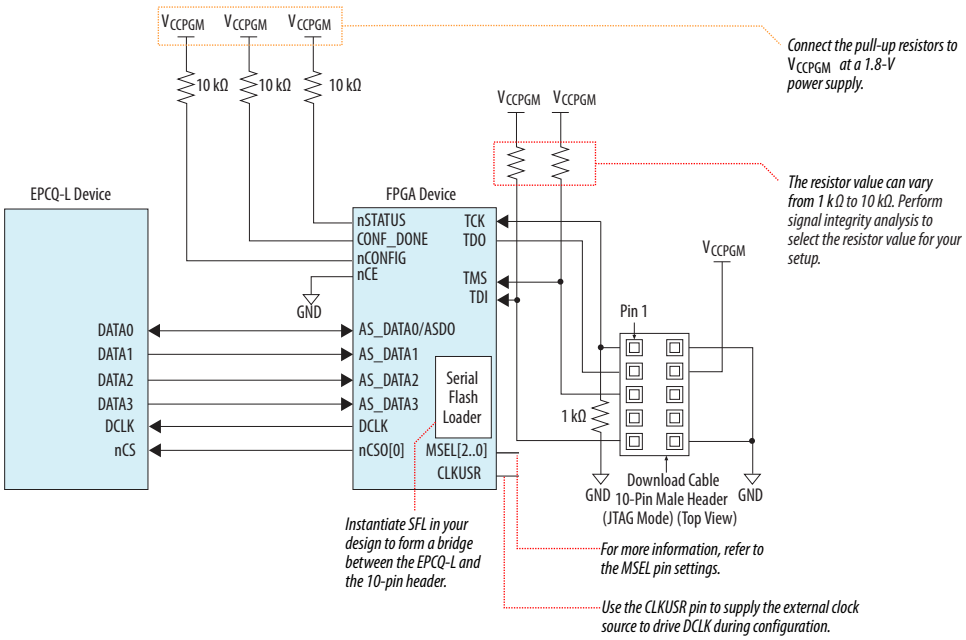
Related Information

- AN 370: Using the Serial Flash Loader with the Intel Quartus Prime Software
- AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming
- Nios II Flash Programmer User Guide

Programming EPCQ-L Using the JTAG Interface

To program an EPCQ-L device using the JTAG interface, connect the device as shown in the following figure.

Figure 137. Connection Setup for Programming the EPCQ-L Using the JTAG Interface

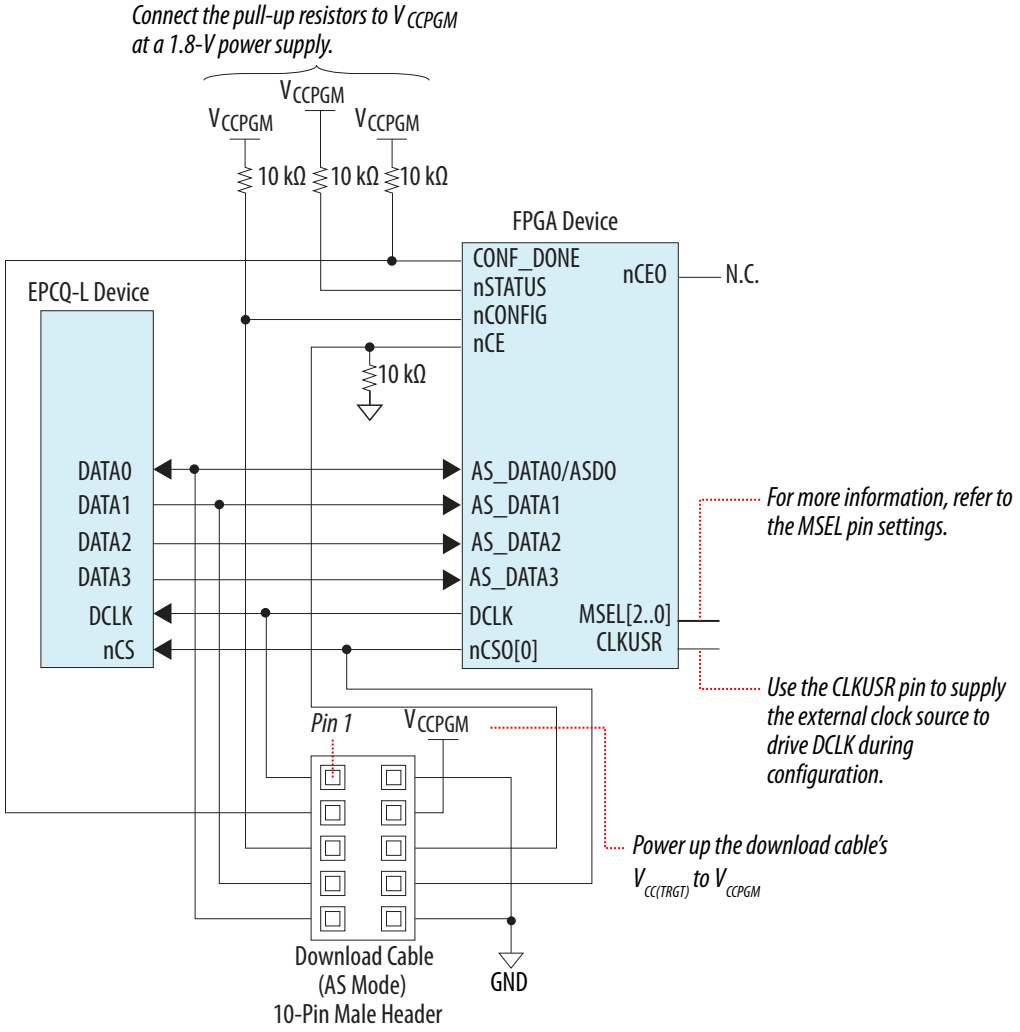


Programming EPCQ-L Using the Active Serial Interface

To program an EPCQ-L device using the AS interface, connect the device as shown in the following figure.

Figure 138. Connection Setup for Programming the EPCQ-L Using the AS Interface

Using the AS header, the programmer serially transmits the operation commands and configuration bits to the EPCQ-L on DATA0.



When programming the EPCQ-L 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 Intel Arria 10 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-L programming using the download cable, DATA0 transfers the programming data, operation command, and address information from the download cable into the EPCQ-L. During the EPCQ-L verification using the download cable, DATA1 transfers the programming data back to the download cable.

Multiple Configuration Devices Support

You are unable to select the number of configuration devices. Based on the configuration device you selected in Convert Programming File tool, the Intel Quartus Prime software indicates the number of configuration devices required to fit the configuration file generated. The following examples explain the multiple flash support:

Table 94. Example of Multiple Configuration Devices Requirements

- Only identical configuration devices are supported for the multiple configuration devices feature.
- Multiple configuration device feature supports up to 3 devices only.

Configuration Device Selected	Configuration Data Size	Configuration Device Required
EPCQ-L256	Larger than 256 Mbit and smaller than 512 MBit	2
EPCQ-L256	Larger than 512 Mbit and smaller than 768 MBit	3

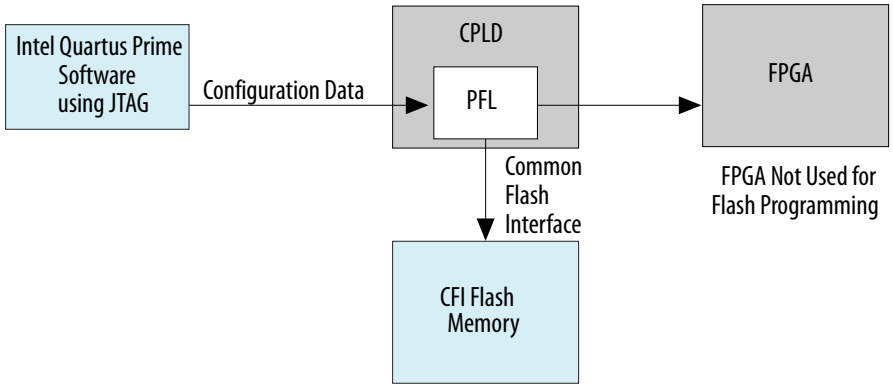
Note: The Intel Quartus Prime Convert Programming File tool creates a `.jic` file based on the setting you set. Configuration will fail if the wrong configuration device type selected in the Convert Programming File tool. However, configuration will work if the configuration devices on your board are more than to the configuration devices required by the generated configuration file.

The Intel Quartus Prime programmer sees multiple configuration devices as a big storage unit. It spans across the flash boundary automatically when the content to be stored exceeds a particular flash capacity.

For example, in Table 94 on page 235, only a single JIC file will be generated. For RPD file generation, multiple RPD files will be generated because the RPD files is programmed directly to the flash with other tools, such as third-party programmer. You must manage the RPD files and determine the right RPD to be programmed into each flash.

7.2.2. Passive Serial Configuration

Figure 139. High-Level Overview of Flash Programming for PS Configuration Scheme



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 Intel Arria 10 device.

For a PC host, connect the PC to the device using an Intel FPGA download cable.

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.

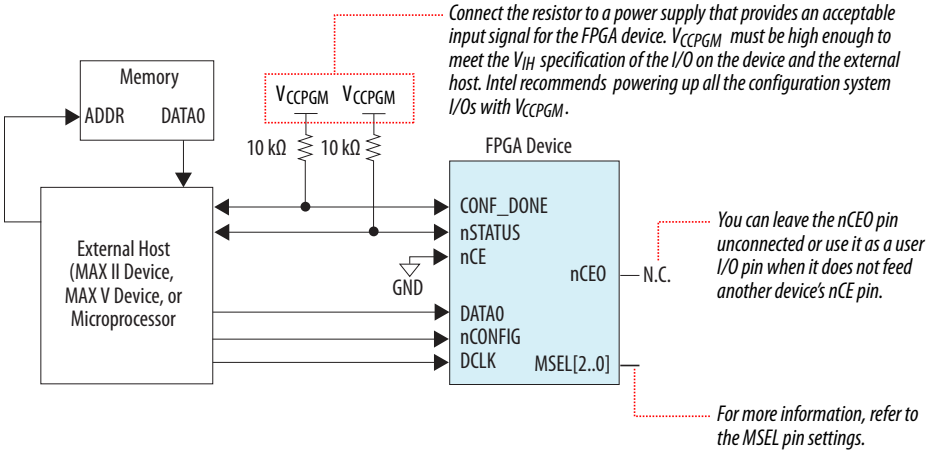
Related Information

- Intel Arria 10 Hard Processor System Technical Reference Manual
  Provides more information about the configuration via HPS.
- Parallel Flash Loader Intel FPGA IP Core User Guide

7.2.2.1. Passive Serial Single-Device Configuration Using an External Host

To configure Intel Arria 10 device, connect the device to an external host, as shown in the following figure.

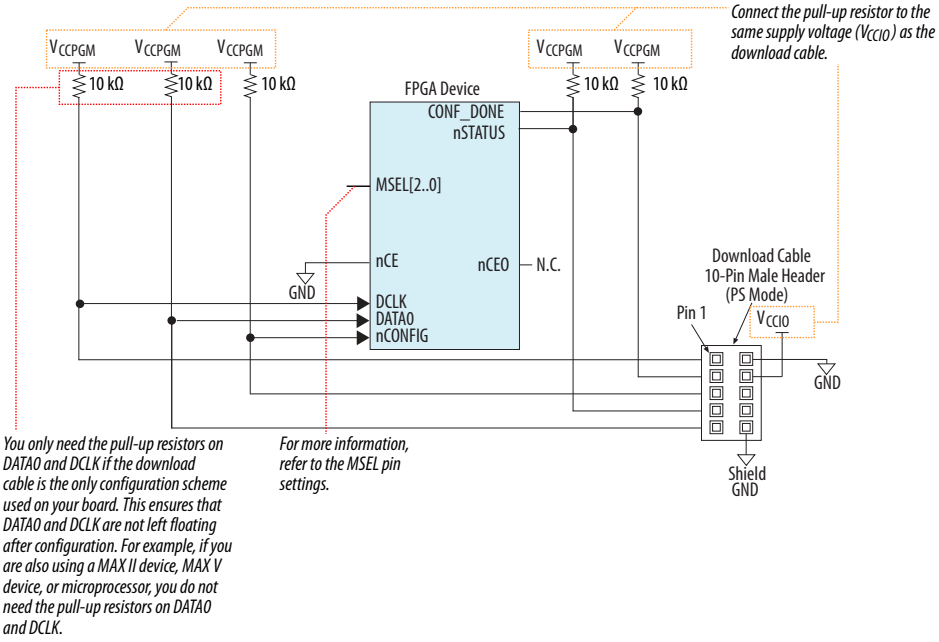
Figure 140. Single Device PS Configuration Using an External Host



7.2.2.2. Passive Serial Single-Device Configuration Using an Intel FPGA Download Cable

To configure Intel Arria 10 device, connect the device to a download cable, as shown in the following figure.

Figure 141. Single Device PS Configuration Using an Intel FPGA Download Cable



7.2.2.3. Passive Serial Multi-Device Configuration

You can configure multiple Intel Arria 10 devices that are connected in a chain.

7.2.2.3.1. 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

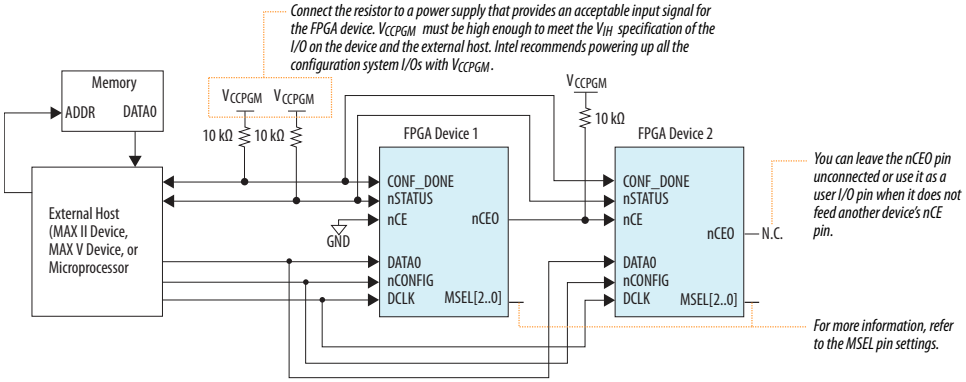
  By tying the CONF_DONE and nSTATUS pins together, the devices initialize and enter user mode at the same time. If any device in the chain detects an error, configuration stops for the entire chain and you must reconfigure all the devices. For example, if the first device in the chain flags an error on 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.

7.2.2.3.2. Using Multiple Configuration Data

To configure multiple Intel Arria 10 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 nCE0 pin is disabled in the Intel Quartus Prime software. For the multi-device configuration chain, you must enable the nCE0 pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

Figure 142. Multiple Device PS Configuration when Both Devices Receive Different Sets of Configuration Data



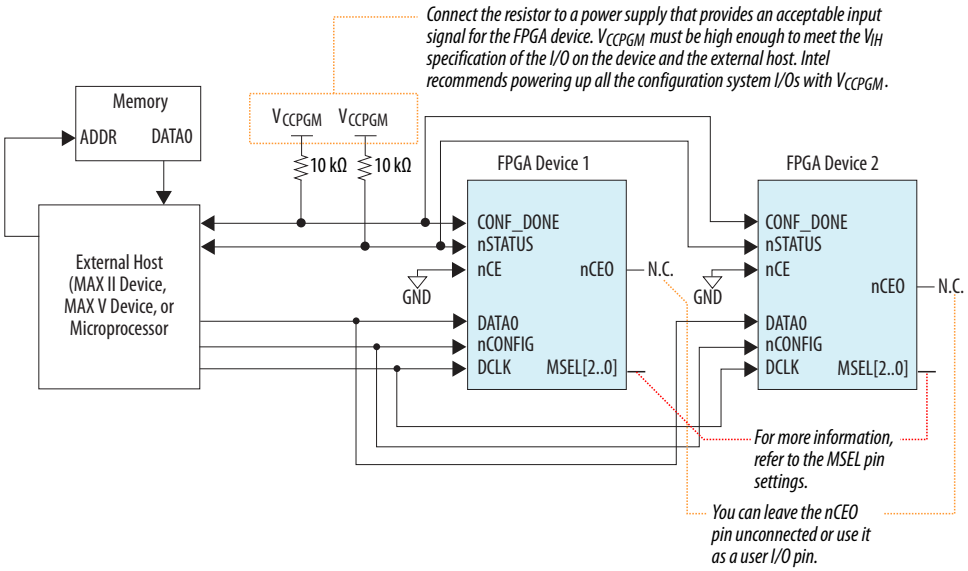
After a device completes configuration, its nCE0 pin is released low to activate the nCE pin of the next device in the chain. Configuration automatically begins for the second device in one clock cycle.

7.2.2.3.3. Using One Configuration Data

To configure multiple Intel Arria 10 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 nCE0 pin is disabled in the Intel Quartus Prime software. For the multi-device configuration chain, you must enable the nCE0 pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

Figure 143. Multiple Device PS Configuration When Both Devices Receive the Same Set of Configuration Data



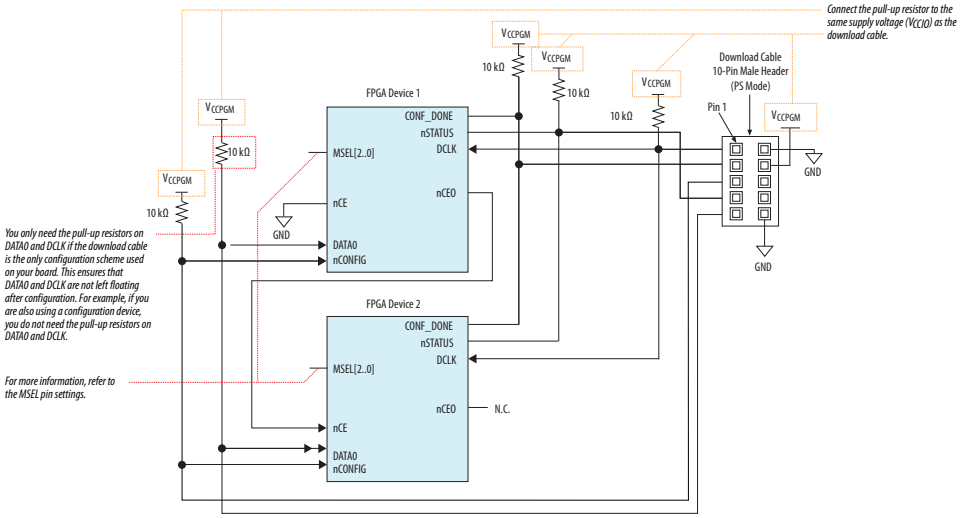
The nCE pins of the devices in the chain are connected to GND, allowing configuration for these devices to begin and end at the same time.

7.2.2.3.4. Using PC Host and Download Cable

To configure multiple Intel Arria 10 devices, connect the devices to a download cable, as shown in the following figure.

Note: By default, the nCE0 pin is disabled in the Intel Quartus Prime software. For the multi-device configuration chain, you must enable the nCE0 pin in the Intel Quartus Prime software. Otherwise, device configuration could fail.

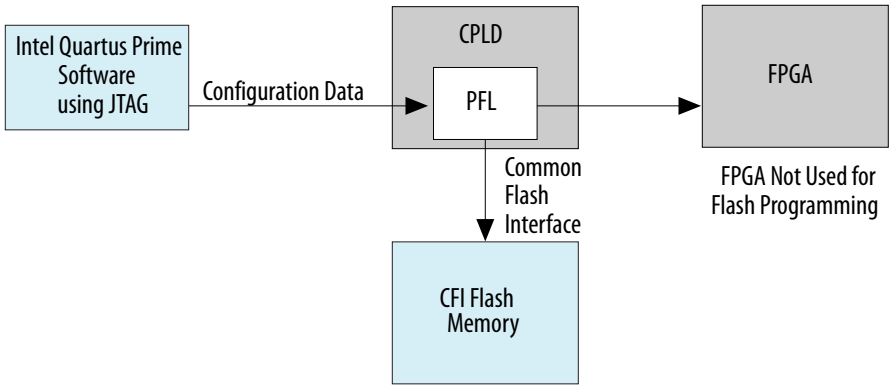
Figure 144. Multiple Device PS Configuration Using an Intel FPGA Download Cable



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.

7.2.3. Fast Passive Parallel Configuration

Figure 145. High-Level Overview of Flash Programming for FPP Configuration Scheme



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 Intel Arria 10 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 Intel Arria 10 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

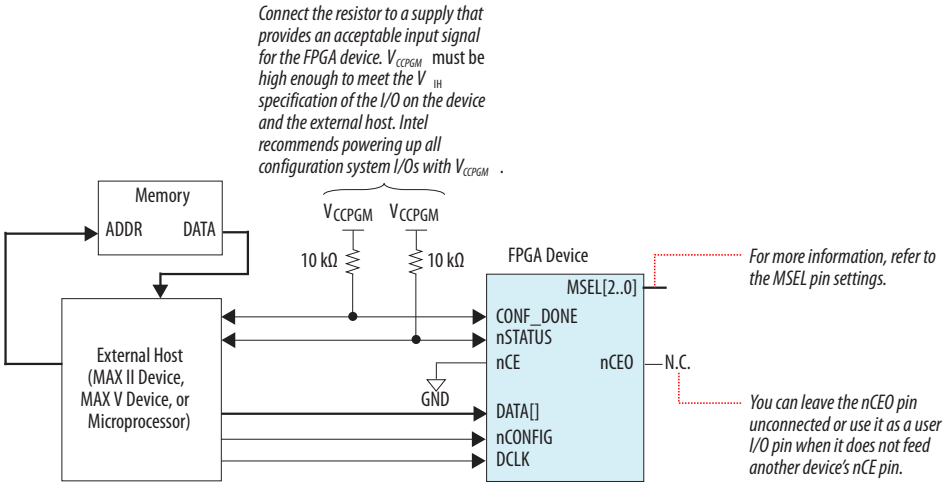
- Parallel Flash Loader Intel FPGA IP Core User Guide
- Intel Arria 10 Device Datasheet
  Provides more information about the FPP configuration timing.

7.2.3.1. Fast Passive Parallel Single-Device Configuration

To configure an Intel Arria 10 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. If you are using FPP x32 configuration mode, use DATA[31..0] pins.

Figure 146. Single Device FPP Configuration Using an External Host



7.2.3.2. Fast Passive Parallel Multi-Device Configuration

You can configure multiple Intel Arria 10 devices that are connected in a chain.

7.2.3.2.1. 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.

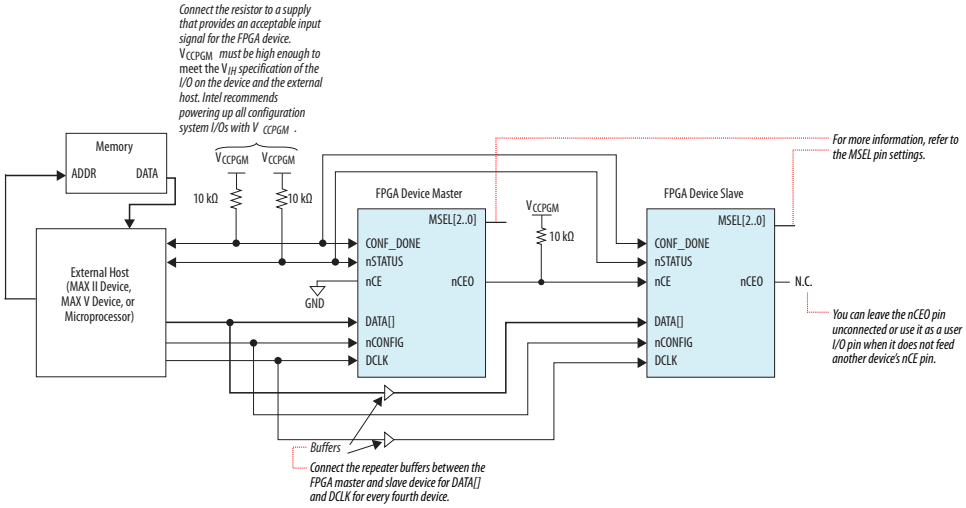
7.2.3.2.2. Using Multiple Configuration Data

To configure multiple Intel Arria 10 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. If you are using FPP x32 configuration mode, use DATA[31..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 147. Multiple Device FPP Configuration Using an External Host When Both Devices Receive a Different Set of Configuration Data



When a master device completes configuration, its nCEO pin is released low to activate the nCE pin of the next device (for example, a slave device) in the chain. Configuration automatically begins for the slave device in one clock cycle.

Note that after the master device receives the configuration bitstream and asserts its nCEO pin low, it will no longer read data even when DCLK is toggling. Once the slave device has received the complete configuration data from the host controller, the master device will enter user mode.

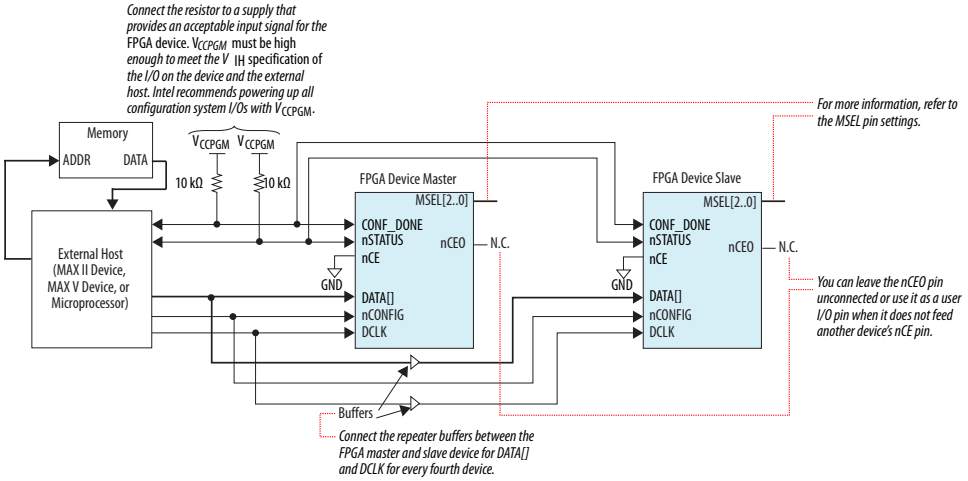
7.2.3.2.3. Using One Configuration Data

To configure multiple Intel Arria 10 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. If you are using FPP x32 configuration mode, use `DATA[31..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 148. 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.

7.2.4. JTAG Configuration

In Intel Arria 10 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 Intel Arria 10 decompression or design security features if you are configuring your Intel Arria 10 device using JTAG-based configuration.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Intel Arria 10 devices do not affect JTAG boundary-scan or programming operations.

The Intel FPGA download cable can support V_{CCPGM} supply at 1.5 V or 1.8 V; it does not support a target supply voltage of 1.2 V.

Related Information

- Device Configuration Pins on page 257
  Provides more information about JTAG configuration pins.
- JTAG Secure Mode on page 271
- Intel Arria 10 Device Datasheet
  Provides more information about the JTAG configuration timing.
- Programming Support for Jam STAPL Language
- Intel FPGA USB Download Cable User Guide
- ByteBlaster II Download Cable User Guide

- EthernetBlaster Communications Cable User Guide
- EthernetBlaster II Communications Cable User Guide

7.2.4.1. JTAG Single-Device Configuration

To configure a single device in a JTAG chain, the programming software sets the other devices to 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.

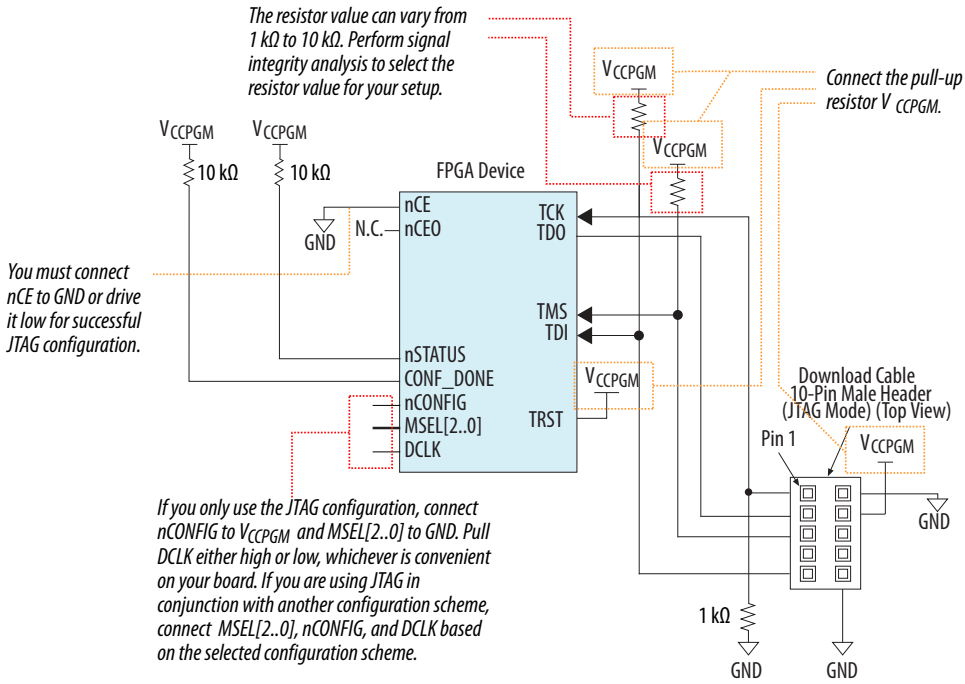
The Intel Quartus Prime software can use the CONF_DONE pin to verify the completion of the configuration process through the JTAG port:

- 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.

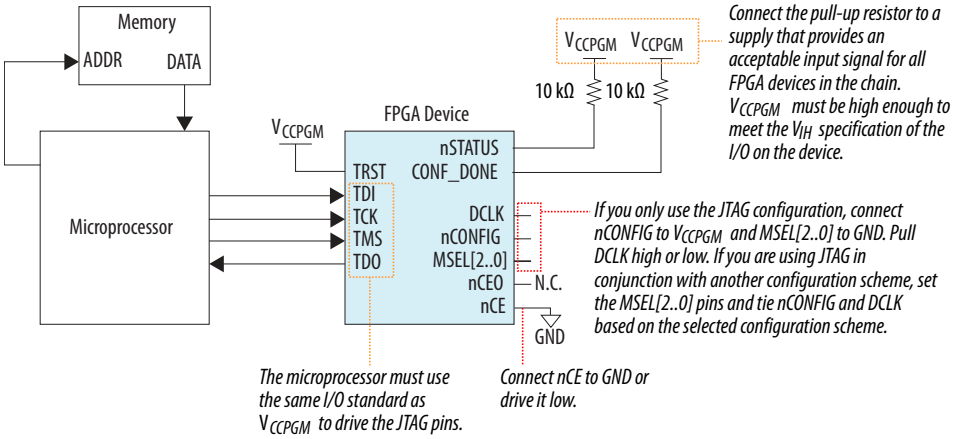
To configure Intel Arria 10 device using a download cable, connect the device as shown in the following figure.

Figure 149. JTAG Configuration of a Single Device Using a Download Cable



To configure Intel Arria 10 device using a microprocessor, connect the device as shown in the following figure. You can use JRunner as your software driver.

Figure 150. JTAG Configuration of a Single Device Using a Microprocessor



Related Information

AN 414: The JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration

7.2.4.2. JTAG Multi-Device Configuration

You can configure multiple devices in a JTAG chain.

7.2.4.2.1. Pin Connections and Guidelines

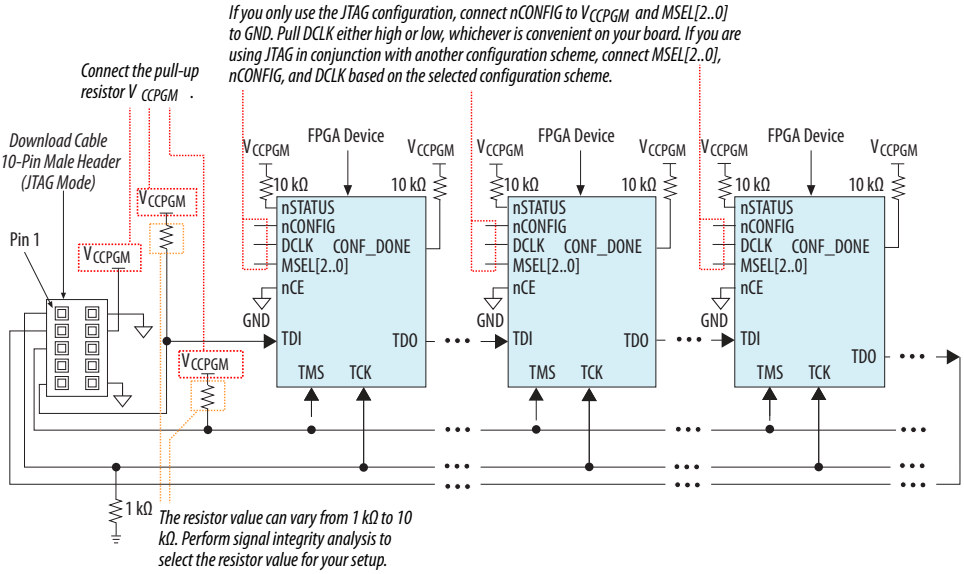
Observe the following pin connections and guidelines for this configuration setup:

- 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 Intel FPGAs 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.

7.2.4.2.2. Using a Download Cable

The following figure shows a multi-device JTAG configuration.

Figure 151. 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.

7.3. Configuration Details

This section describes the MSEL pin settings, configuration sequence, device configuration pins, configuration pin options, and configuration data compression.

7.3.1. 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:

Table 95. MSEL Pin Settings for Each Configuration Scheme of Intel Arria 10 Devices

- Do not drive the MSEL pins with a microprocessor or another device.
- Use PS or FPP MSEL pin setting for configuration via HPS.

Configuration Scheme	V_{CCPGM} (V)	Power-On Reset (POR) Delay	Valid MSEL[2..0]
JTAG-based configuration	—	—	Use any valid MSEL pin settings below
AS (x1 and x4)	1.8	Fast	010
		Standard	011
PS and FPP (x8, x16, and x32)	1.2/1.5/1.8	Fast	000
		Standard	001

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

- Intel Arria 10 Hard Processor System Technical Reference Manual
  Provides more information about the configuration via HPS.
- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
  Provides more information about JTAG pins voltage-level connection.

7.3.2. CLKUSR

You can use CLKUSR pin as the clock source for Intel Arria 10 device configuration and initialization. CLKUSR pin can also be used for configuration and transceiver calibration simultaneously.

For transceiver calibration, CLKUSR must be a free-running clock running between 100 MHz to 125 MHz at power-up depending on the device's configuration scheme as shown in the following table. Transceiver calibration starts utilizing the CLKUSR during device configuration and may continue to use it even when the device enters user mode.

Note: CLKUSR cannot be used as the reference clock for PLL.

Table 96. Available Configuration Clock Source and Transceiver Calibration CLKUSR Frequency for Intel Arria 10 Devices

Configuration Scheme	Supported Clock Source for Device Configuration	Supported Clock Source for Device Initialization	Supported CLKUSR Frequency for Transceiver Calibration
AS	Internal Oscillator, CLKUSR	Internal Oscillator, CLKUSR	100 MHz
PS	DCLK only	Internal Oscillator, CLKUSR, DCLK	100 to 125 MHz
FPP (x8, x16, x32)			

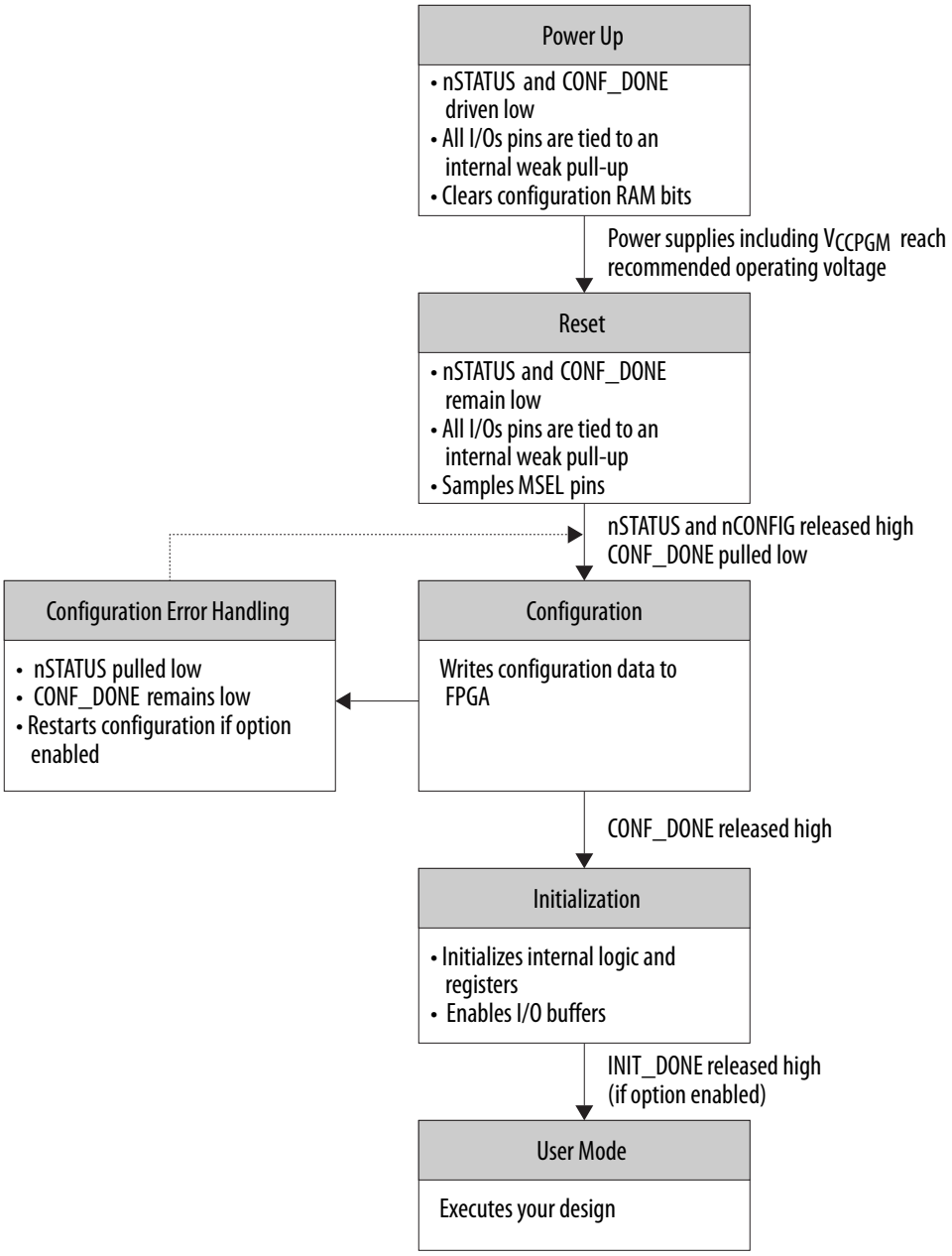
Related Information

Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
Provides more information about CLKUSR pin.

7.3.3. Configuration Sequence

Describes the configuration sequence and each configuration stage.

Figure 152. Configuration Sequence for Intel Arria 10 Devices



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.

7.3.3.1. Power Up

Power up all the power supplies that are monitored by the POR circuitry. All power supplies, including V_{CCPGM}, 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 Intel Arria 10 devices. Connect V_{CCPGM} to 1.8 V.

The operating voltage for the configuration input pin is independent of the I/O banks power supply, V_{CCIO}, during configuration. Therefore, Intel Arria 10 devices do not require configuration voltage constraints on V_{CCIO}.

Intel recommends connecting the I/O banks power supply, V_{CCIO}, of the dual-purpose configuration pins for FPP x8, x16, and x32 to V_{CCPGM}.

Related Information

- Intel Arria 10 Device Datasheet
  Provides more information about the ramp-up time specifications.
- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
  Provides more information about configuration pin connections.
- Device Configuration Pins on page 257
  Provides more information about configuration pins.

7.3.3.2. 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 Intel Arria 10 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 247
- Intel Arria 10 Device Datasheet
  Provides more information about the POR delay specification.

7.3.3.3. Configuration

For more information about the DATA[] pins for each configuration scheme, refer to the appropriate configuration scheme.

7.3.3.3.1. Configuration Error Detection

When the Intel Quartus Prime software generates the configuration bitstream, the software also computes a 32-bit CRC value for each CRAM frame. A configuration bitstream contains one CRC value for each data frames. The length of the data frame can vary for each device.

As each 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.

7.3.3.4. 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

Intel Arria 10 Device Datasheet
Provides more information about t_{STATUS} and t_{CFG} timing parameters.

7.3.3.5. 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 Intel Arria 10 device provides 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, Intel Arria 10 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

Intel Arria 10 Device Datasheet
Provides more information about t_{CD2CU}, t_{init}, and t_{CD2UMC} timing parameters, and initialization clock source.

7.3.3.6. 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 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 *Intel Arria 10 Device Datasheet* before starting any operation after the device enters into user mode. For example, the t_{CD2UM} range for Intel Arria 10 device is between 175 us to 830 us.
- Use an internal `init_done` signal to hold the reset of your core registers, core logic, and I/O registers until the device has fully entered user mode. The internal `init_done` signal is high (enabled) until the entire device enters user mode.

```
twentynm_controller u1 (
.initdonecore(init_done)
);
```

- 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 *Intel Arria 10 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.

Related Information

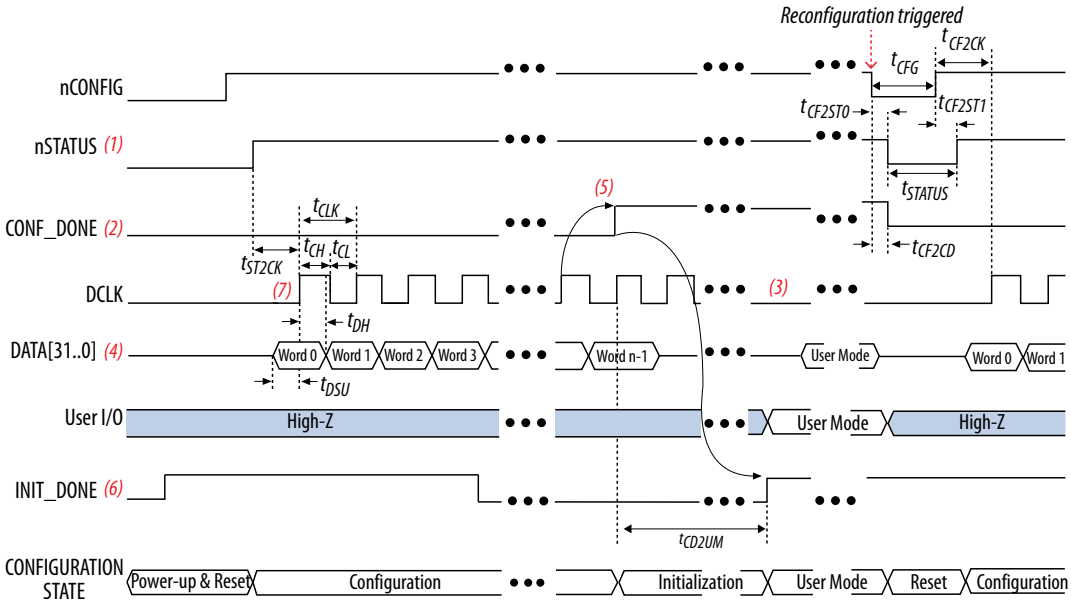
Intel Arria 10 Device Datasheet
Provides more information about t CD2UM and t CD2UMC specifications.

7.3.4. Configuration Timing Waveforms

7.3.4.1. FPP Configuration Timing

Figure 153. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is 1

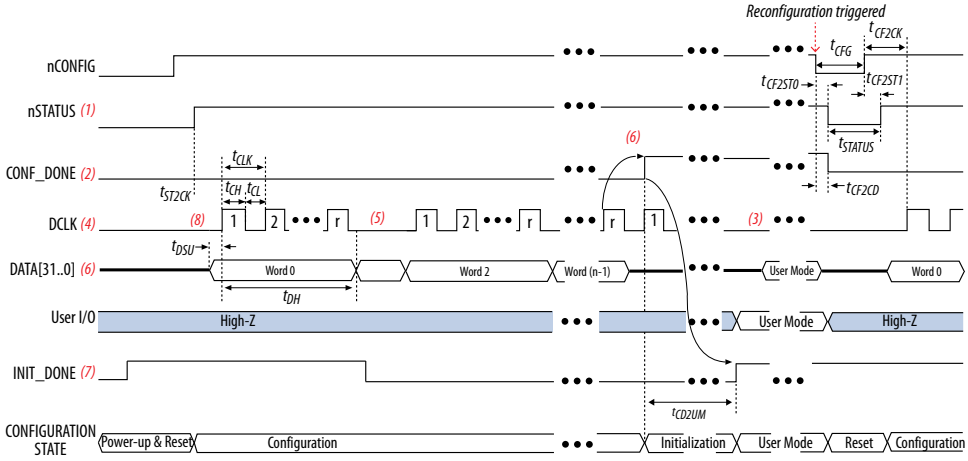
The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.



(1) After power-up, the device 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 ×16, use DATA[15..0]. For FPP ×8, use DATA[7..0]. DATA[31..0] are available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.
(5) To ensure a successful configuration, send the entire configuration data to the device. CONF_DONE is released high when the 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.
(6) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.
(7) Do not toggle the DCLK high before nSTATUS is pulled high.

Figure 154. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is >1

The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.



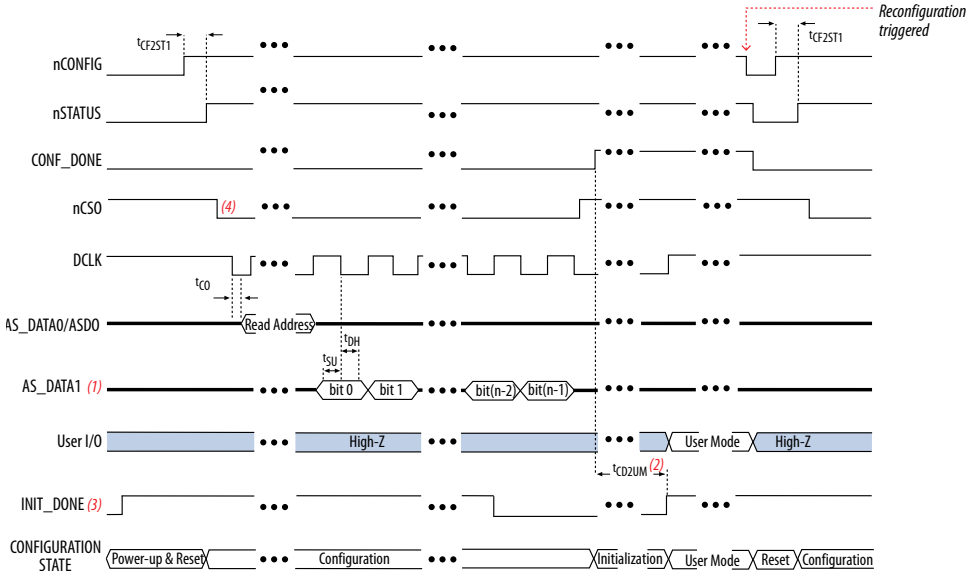
(1) After power-up, the device holds nSTATUS low for the time as specified by the POR delay.
(2) After power-up, before and during configuration, CONF_DONE is low.
(3) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.
(4) "r" denotes the DCLK-to-DATA[] ratio. For the DCLK-to-DATA[] ratio based on the decompression and the design security feature enable settings.
(5) If needed, pause DCLK by holding it low. When DCLK restarts, the external host must provide data on the DATA[31..0] pins prior to sending the first DCLK rising edge.
(6) To ensure a successful configuration, send the entire configuration data to the device. CONF_DONE is released high after the 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.
(8) Do not toggle the DCLK high before nSTATUS is pulled high.

Related Information

DCLK-to-DATA[] Ratio (r) for FPP Configuration

7.3.4.2. AS Configuration Timing

Figure 155. AS Configuration Timing Waveform

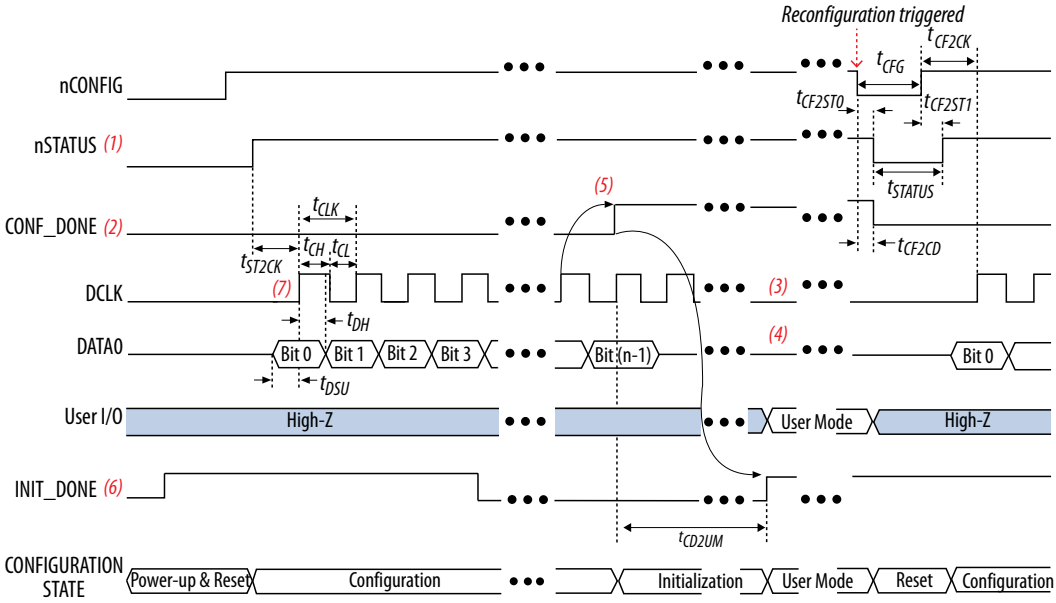


(1) If you are using AS ×4 mode, this signal represents the AS_DATA[3..0] and EPCQ-L sends in 4-bits of data for each DCLK cycle.
(2) The initialization clock can be from internal oscillator or CLKUSR pin.
(3) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.
(4) The time between the falling edge of nCSO to the first toggling of DCLK is more than 15ns.

7.3.4.3. PS Configuration Timing

Figure 156. PS Configuration Timing Waveform

The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.



(1) After power-up, the device 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. You can drive it high or low, whichever is more convenient.
(4) DATA0 is available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings in the Device and Pins Option.
(5) To ensure a successful configuration, send the entire configuration data to the device. CONF_DONE is released high after the 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.
(6) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.
(7) Do not toggle the DCLK high before nSTATUS is pulled high.

7.3.5. Estimating Configuration Time

The configuration time is mostly the time it takes to transfer the configuration data from a CFI flash memory or an EPCQ-L device to the Intel Arria 10 device.

Use the following equations to estimate the configuration time:

AS Configuration

By default, the AS x1 mode is used. The Intel Arria 10 device determines the AS mode by reading the option bit in the programming file.

- AS x1 mode

  Estimated minimum configuration time=
  .rbf size x (minimum DCLK period / 1 bit per DCLK cycle)

- AS x4 mode

  Estimated minimum configuration time
  = **.rbf** size x (minimum DCLK period / 4 bits per DCLK cycle)

PS Configuration

Estimated minimum configuration time = **.rbf** size x (minimum DCLK period / 1 bit per DCLK cycle)

FPP Configuration

Estimated minimum configuration time = **.rbf** size/FPP data width x *r* x minimum DCLK period

where *r* is the DCLK-to-DATA[] ratio.

Note: Compressing the configuration data decreases the configuration time. The amount of time increased varies depending on the configuration method and corresponding DCLK ratio.

Related Information

DCLK-to-DATA[] Ratio (r) for FPP Configuration

7.3.6. Device Configuration Pins

Configuration Pins Summary

The following table lists the Intel Arria 10 configuration pins and their power supply.

Note:
1. The TDI, TMS, TCK, TDO, and TRST pins are powered by V_{CCPGM}.
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.

Table 97. Configuration Pin Summary for Intel Arria 10 Devices

Configuration Pin	Configuration Scheme	Input/Output	User Mode	Powered By
TDI	JTAG	Input	—	V_{CCPGM}
TMS	JTAG	Input	—	V_{CCPGM}
TCK	JTAG	Input	—	V_{CCPGM}
TDO	JTAG	Output	—	V_{CCPGM}
TRST	JTAG	Input	—	V_{CCPGM}
CLKUSR	Optional, All schemes	Input	I/O	V_{CCPGM}/V_{CCIO} (34)
CRC_ERROR	Optional, all schemes	Output	I/O	V_{CCPGM} /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} (34)
DEV_CLRn	Optional, all schemes	Input	I/O	V_{CCPGM}/V_{CCIO} (34)
INIT_DONE	Optional, all schemes	Output	I/O	Pull-up

continued...

Configuration Pin	Configuration Scheme	Input/Output	User Mode	Powered By
MSEL[2..0]	All schemes	Input	—	V_{CCPGM}
nSTATUS	All schemes	Bidirectional	—	V_{CCPGM}/Pull-up
nCE	All schemes	Input	—	V_{CCPGM}
nCEO	Optional, All schemes	Output	I/O	Pull-up
nCONFIG	All schemes	Input	—	V_{CCPGM}
DATA[31..1]	FPP	Input	I/O	V_{CCPGM}/V_{CCIO} (34)
DATA0	FPP and PS	Input	I/O	V_{CCPGM}/V_{CCIO} (34)
nCSO[2..0]	AS	Output	—	V_{CCPGM}
nIO_PULLUP (33)	All schemes	Input	—	V_{CC}
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} (34)
PR_READY	Partial Reconfiguration	Output	I/O	V_{CCPGM}/V_{CCIO} (34)
PR_ERROR	Partial Reconfiguration	Output	I/O	V_{CCPGM}/V_{CCIO} (34)
PR_DONE	Partial Reconfiguration	Output	I/O	V_{CCPGM}/V_{CCIO} (34)

Related Information

Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
Provides more information about each configuration pin.

7.3.6.1. I/O Standards and Drive Strength for Configuration Pins

The standard I/O voltage for Intel Arria 10 devices is 1.8 V. The drive strength setting for dedicated configuration I/O are hardwired. The default drive strength for dual function configuration I/O pins during configuration is 1.8V at 50 Ω. When you enable the configuration pins, the Intel Quartus Prime software sets the CVP_CONF_DONE pin to a drive strength of 1.8 V CMOS 4 mA, and the INIT_DONE and CRC_ERROR pins to a drive strength of 1.8 V CMOS 8 mA.

Table 98. I/O Standards and Drive Strength for Configuration Pins

Configuration Pin	Input/Output/ Bidir	Drive Strength (mA)	Slew Rate (Fast/Slow)	I/O Standard
TDI	Input	—	—	1.8V schmitt trigger
TCK	Input	—	—	1.8V schmitt trigger

continued...

(33) If you tie nIO_PULLUP pin to VCC, ensure that all user I/O pins and dual-purpose I/O pins are at valid logic (0 or 1) after all the power supplies have reached full nominal voltage, before and during configuration.

(34) This pin is powered by V_{CCPGM} before and during configuration and is powered by V_{CCIO} if used as a user I/O pin during user mode.

Configuration Pin	Input/Output/ Bidir	Drive Strength (mA)	Slew Rate (Fast/Slow)	I/O Standard
TMS	Input	—	—	1.8V schmitt trigger
TRST	Input	—	—	1.8V schmitt trigger
TDO	Output	12		1.8V LVCMOS
AS_DATA[0]/ASDO	Bidir	24[35]	1 (Fast)	1.8V LVCMOS
AS_DATA[3:1]	Bidir	24[35]	1 (Fast)	1.8V LVCMOS
DCLK	Bidir	24[35]	1 (Fast)	1.8V LVCMOS
MSEL[2:0]	Input	—	—	1.8V schmitt trigger
nCE	Input	—	—	1.8V schmitt trigger
nCONFIG	Input	—	—	1.8V schmitt trigger
nIO_PULLUP	Input	—	—	1.8V schmitt trigger
nSTATUS	Bidir	4	1 (Fast)	1.8V LVCMOS
CONF_DONE	Bidir	4	1 (Fast)	1.8V LVCMOS
INIT_DONE	Bidir	8	1 (Fast)	1.8V LVCMOS
nCSO[2:0]	Output	12	1 (Fast)	1.8V LVCMOS
DATA[31:0]	Input	—	—	1.8V LVCMOS
CRC_ERROR	Bidir	8	1 (Fast)	1.8V LVCMOS
CvP_CONFDONE	Bidir	4	1 (Fast)	1.8V LVCMOS

7.3.6.2. 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 99. 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

continued...

(35) 24 mA drive strength IBIS model is not available for I/O simulation. Intel recommends to use 12 mA drive strength IBIS model as a replacement.

Configuration Pin	Category Page	Option
PR_REQUEST	General	Enable PR pin
PR_READY		
PR_ERROR		
PR_DONE		

Related Information

Reviewing Printed Circuit Board Schematics with the Intel Quartus Prime Software
Provides more information about the device and pin options dialog box setting.

7.3.7. Configuration Data Compression

Intel Arria 10 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.

Note: You cannot enable encryption and compression at the same time for all configuration scheme.

7.3.7.1. Enabling Compression Before Design Compilation

To enable compression before design compilation, follow these steps:

1. On the Assignment Menu, click **Device**.
2. Select your Intel Arria 10 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**.

7.3.7.2. Enabling Compression After Design Compilation

To enable compression after design compilation, follow these steps:

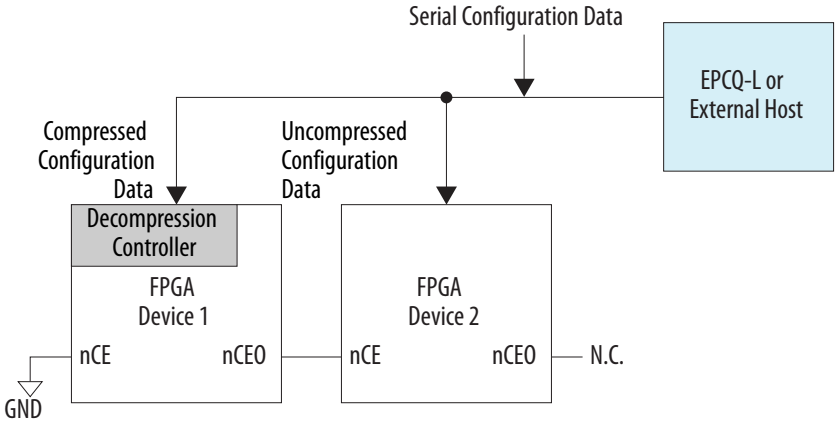
1. On the File menu, click **Convert Programming Files**.
2. Select the programming file type (**.pof**, **.sof**, **.hex**, **.hexout**, **.rbf**, or **.ttf**). For POF output files, select a configuration device.
3. Under the **Input files to convert** list, select **SOF Data**.
4. Click **Add File** and select an Intel Arria 10 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.

7.3.7.3. Using Compression in Multi-Device Configuration

The following figure shows a chain of two Intel Arria 10 devices. Compression is only enabled for the first device.

This setup is supported by the AS or PS multi-device configuration only.

Figure 157. Compressed and Uncompressed Serial Configuration Data in the Same Configuration File

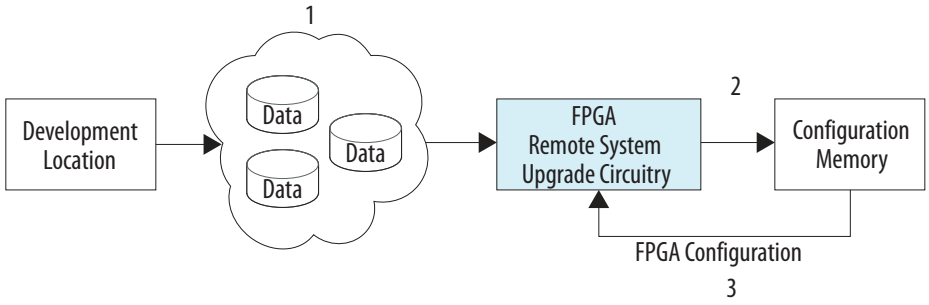


For the FPP configuration scheme, a combination of compressed and uncompressed configuration in the same multi-device configuration chain is not allowed because of the difference on the DCLK-to-DATA[] ratio.

7.4. Remote System Upgrades Using Active Serial Scheme

Intel Arria 10 devices contain dedicated remote system upgrade circuitry. You can use this feature to upgrade your system from a remote location.

Figure 158. Intel Arria 10 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 Intel Arria 10 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.

When an error occurs, the circuitry detects the error, reverts to a safe configuration image, and provides error status to your design.

7.4.1. Configuration Images

Intel Arria 10 devices offer a new remote system upgrade feature which provides direct-to-application and application-to-application updates. When the Intel Arria 10 device is powered up in the remote update programming mode, the Intel Arria 10 device loads the factory or application configuration image as indicated by the start address pointer at `32'd0` address of the EPCQ-L device.

Each Intel Arria 10 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 Intel Arria 10 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.

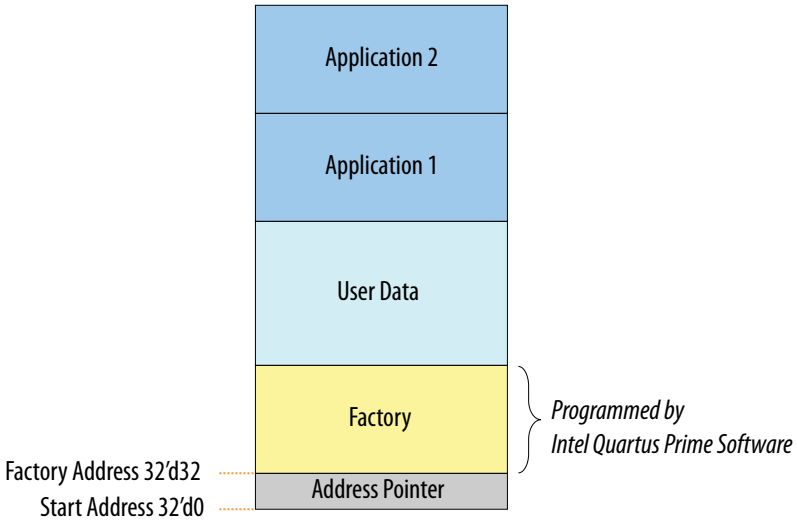
You can also create one or more application images for the device. An application image contains selected functionalities to be implemented in the target device.

Store the images at the following locations in the EPCQ-L devices:

- Factory configuration image—`PGM[31..0] = 32'h00000020` start address on the EPCQ-L device.
- Application configuration image—any sector boundary. Intel recommends that you store only one image at one sector boundary.
- Start address (0x00 to 0x1F)—storing the 32-bit address pointer to load the application configuration image upon power up.

Figure 159. Start Address and Factory Address Location

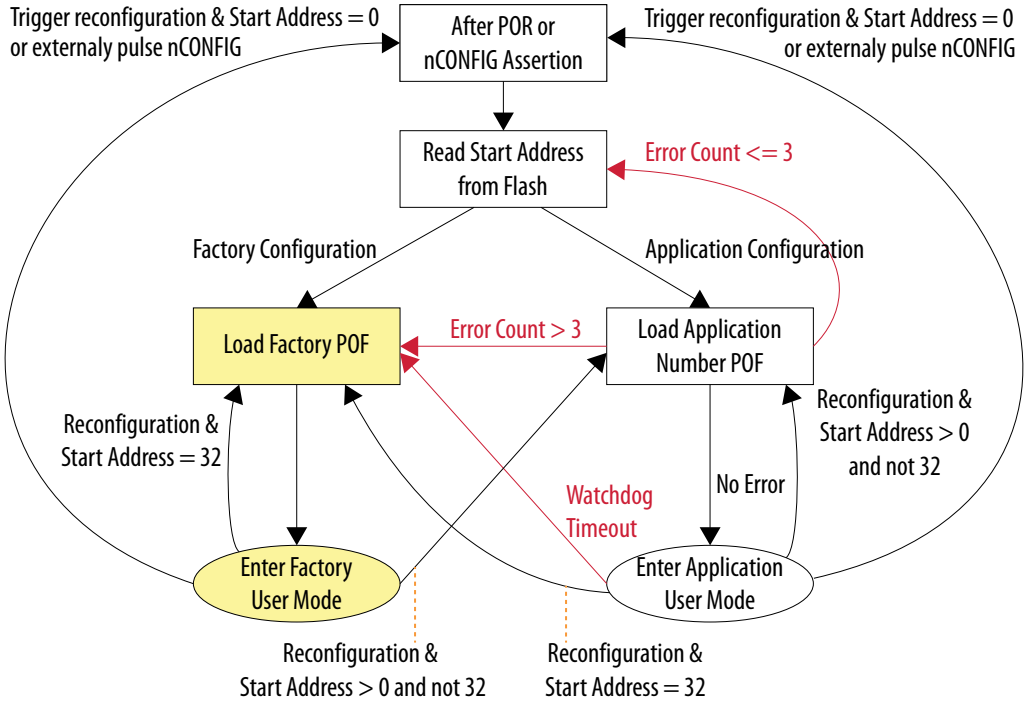
The following diagram illustrates factory, user data, application 1, and application 2 sections. Each section starts at a new sector boundary.



Note: Intel recommends that you set a fixed start address and never update the start address during user mode. You should only overwrite an existing application configuration image when you have a new application image. This is to avoid the factory configuration image to be erased unintentionally every time you update the start address.

7.4.2. Configuration Sequence in the Remote Update Mode

Figure 160. Transitions Between Factory and Application Configurations in Remote Update Mode



Upon power up or reconfiguration triggered using nCONFIG, the AS controller reads the start address from the EPCQ-L device and loads the initial configuration image, either the factory or application configuration image. If the initial image is an application configuration image and an error occurs, the controller tries to load the same initial application configuration image for three times before loading the factory configuration image. If the initial application configuration image encounters a user watchdog timeout error, the controller loads the factory configuration image. You can load a new application configuration image during factory user mode or application user mode. If an error is encountered, the controller loads the factory configuration image.

Note: When error occurs, the AS controller loads the same application configuration image for three times before reverting to factory configuration image. By that time, the total time taken exceeds 100ms and violates the PCIe boot-up time when using CvP. If your design is sensitive to the PCIe boot-up requirement, Intel recommends that you do not use the direct-to-application feature.

Related Information

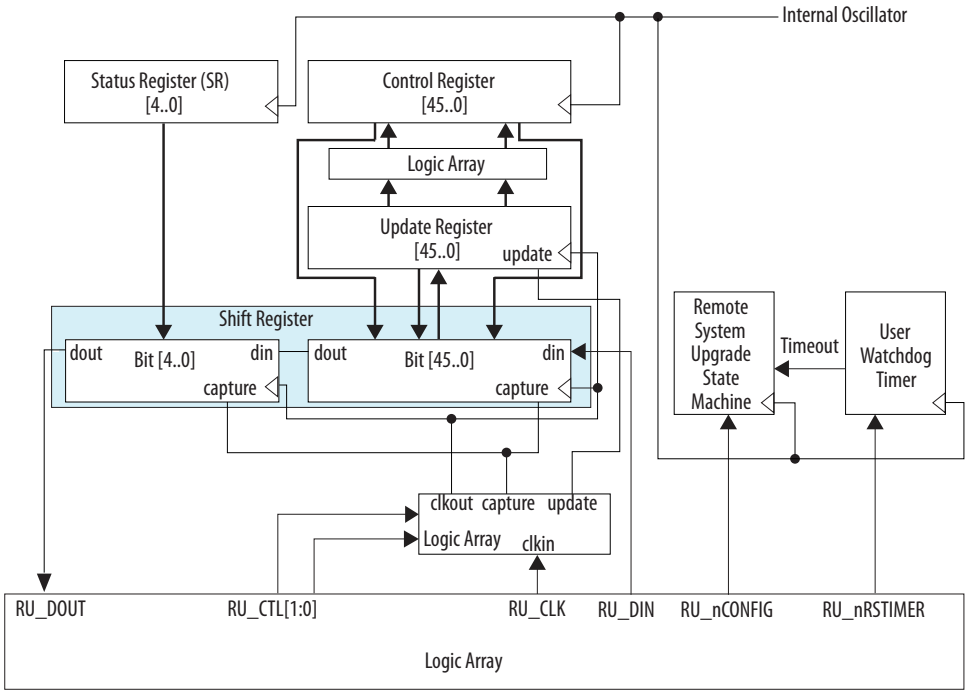
Remote System Upgrade State Machine on page 267
A detailed description of the configuration sequence in the remote update mode.

7.4.3. 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_CTL[1:0], RU_CLK, RU_DIN, RU_nCONFIG, and RU_nRSTIMER signals internally to perform all the related remote system upgrade operations.

Figure 161. Remote System Upgrade Circuitry



Related Information

Intel Arria 10 Device Datasheet
Provides more information about remote system upgrade circuitry timing specifications.

7.4.4. Enabling Remote System Upgrade Circuitry

To enable the remote system upgrade feature, select **Active Serial** or **Configuration Device** from the Configuration scheme list in the **Configuration** page of the **Device and Pin Options** dialog box in the Intel Quartus Prime software.

Intel-provided Remote Update Intel FPGA IP core provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read and write protocol in the Intel Arria 10 device logic.

Related Information

Remote Update Intel FPGA IP User Guide

7.4.5. Remote System Upgrade Registers

Table 100. Remote System Upgrade Registers

Register	Description
Shift	This register is accessible by the core logic and allows the update, status, and control registers to be written and sampled by user logic.
Control	This register contains the current page address, watchdog timer settings, and one bit specifying the current configuration image—factory configuration or application configuration image. This register is used by the AS controller to load the configuration image from the EPCQ-L device during remote system upgrade.
Update	This register contains similar data as the control register, but this register is updated by the factory configuration or application configuration image by shifting data into the shift register, followed by an update. The soft IP core of the remote system upgrade updates this register with the values to be used in the control register during the next reconfiguration cycle.
Status	This register is written by the remote update block during every reconfiguration cycle to record the trigger of a reconfiguration. This information is used by the soft IP core of the remote system upgrade to determine the appropriate action following a reconfiguration cycle.

Related Information

- Control Register on page 266
- Status Register on page 267

7.4.5.1. Control Register

Table 101. Control Register Bits

Bit	Name	Reset Value[36]	Description
0	AnF	1'b0	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. Factory configuration design must set this bit to **1** before triggering reconfiguration using an application configuration image.
1..32	PGM[0..31]	32'h00000000	AS configuration start address.
33	Wd_en	1'b0	User watchdog timer enable bit. Set this bit to **1** to enable the watchdog timer.
34..45	Wd_timer[11..0]	12'h000	User watchdog time-out value.

(36) This is the default value after the device exits POR and during reconfiguration back to the factory configuration image.

7.4.5.2. Status Register

Table 102. Status Register Bits

Bit	Name	Reset Value[37]	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.

7.4.6. 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 or application configuration image is loaded based on the start address stored at 0x00 to 0x1F in the EPCQ-L device.
2. In factory configuration image, 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.

7.4.7. User Watchdog Timer

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.

(37) After the device exits POR and power-up, the status register content is 5'b00000.

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

Intel Arria 10 Device Datasheet
Provides more information about the operating range of the user watchdog internal oscillator's frequency.

7.5. Design Security

The Intel Arria 10 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 mode
- Limited accessible JTAG instruction during power-up in the JTAG Secure mode
- Supports POF authentication and protection against Side-Channel Attack
- Provides JTAG access control and security key control through fuse bit or option bits
- Disables all JTAG instructions from power-up until the device is initialized
- Supports board-level testing
- Supports off-board key programming for non-volatile key
- Stand-alone Qcrypt tool to encrypt and decrypt with other security settings to configuration bit stream.
- Available in all configuration schemes except JTAG
- Supports remote system upgrades feature

Table 103. Design Security Approach for Intel Arria 10 FPGAs

Design Security Element	Description
Non-Volatile key	The non-volatile key is securely stored in fuses within the device. Proprietary security features make it difficult to determine this key.
Volatile Key	The volatile key is securely stored in battery-backed RAM within the device. Proprietary security features make it difficult to determine this key.
Key Generation	A user provided 256-bit key is processed by a one-way function before being programmed into the device.
Key Choice	Both volatile and non-volatile key can exist in a device. User can choose which key to use by setting the option bits in encrypted configuration file through the Convert Programming File tool or the Qcrypt tool.

continued...

Design Security Element	Description
Tamper Protection Mode	Tamper protection mode prevents the FPGA from being loaded with an unencrypted configuration file. When you enable this mode, the FPGA can only be loaded with a configuration that has been encrypted with your key. Unencrypted configurations and configurations encrypted with the wrong key results in a configuration failure. You can enable this mode by setting a fuse within the device.
Configuration Readback	These devices do not support a configuration readback feature. From a security perspective, this makes readback of your unencrypted configuration data infeasible.
Security Key Control	By using different JTAG instructions and the security option in the Qcrypt tool, you have the flexibility to permanently or temporarily disable the use of the non-volatile or volatile key. You can also choose to lock the volatile key to prevent it from being overwritten or reprogrammed.
JTAG Access Control	You can enable various levels of JTAG access control by setting the OTP fuses or option bits in the configuration file using the Qcrypt tool: 1. Force full configuration or partial configuration to be done through HPS only. 2. Bypass external JTAG pin or HPS JTAG. This feature disables external JTAG or HPS JTAG access, but can be unlocked through internal core access. 3. Disable all AES key related JTAG instructions from external JTAG pins. 4. Allows only a limited set of mandatory JTAG instruction to be accessed through external JTAG, similar to JTAG Secure mode.

Note:
- You cannot enable encryption and compression at the same time for all configuration scheme.
- When you use design security with Intel Arria 10 devices in an FPP configuration scheme, it requires a different DCLK-to-DATA[] ratio.

Related Information

AN 556: Using the Design Security Features in Intel FPGAs
Provides more information about applying design security features in Intel Arria 10 devices.

7.5.1. Security Key Types

Intel Arria 10 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 104. Security Key Types

Key Types	Key Programmability	Power Supply for Key Storage	Programming Method
Volatile	• Reprogrammable • Erasable	Required external battery, V_{CCBAT} (38)	On-board
Non-volatile	One-time programming	Does not require an external battery	On-board and in-socket programming (39)

Both non-volatile and volatile key programming offers protection from reverse engineering and copying. If you set the tamper-protection mode, the design is also protected from tampering.

(38) 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.

(39) Third-party vendors offer in-socket programming.

Related Information

- AN 556: Using the Design Security Features in Intel FPGAs
  Provides more information about programming volatile and non-volatile key into the FPGA.
- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
  Provides more information about the V_{CCBAT} pin connection recommendations.
- Intel Arria 10 Device Datasheet
  Provides more information about battery specifications.
- Supported JTAG Instruction on page 300

7.5.2. Security Modes

Table 105. Security Modes Available in Intel Arria 10 Devices

Note: For additional details on these instructions or how to burn the fuse for each mode, contact your Intel technical support. Alternatively, you can use the Qcrypt tool to enable all of these design security modes. The Qcrypt tool provides an impermanent solution compared to the burning the fuse which has the one-time programming limitation.

Security Mode	JTAG Instruction	Security Feature
JTAG Secure[40]	EXT_JTAG_SECURE	Allows only mandatory IEEE Std. 1149.1 BST JTAG instructions. See Table 106 on page 271.
Tamper Protection	OTP_VOLKEY_SECURE	Allows only configuration file encrypted with the correct key to be loaded into the Intel Arria 10 device. Unencrypted or wrong encryption key results in configuration failure.
JTAG Bypass	EXTERNAL_JTAG_BYPASS	Disables all the direct control from external JTAG pins or HPS JTAG. Compared to the JTAG Secure mode, devices in JTAG Bypass mode allow access to external JTAG pins or HPS JTAG interface through internal JTAG core.
Key Related Instruction Disable	KEY_EXT_JTAG_DISABLE	Disables all JTAG instructions related to AES key issued from the external JTAG pins.
HPS Configuration Only	FORCE_HPS_CONFIG	Disables the external JTAG pins from configuring or partially reconfiguring the device. Only HPS controls the configuration pins and the MSEL pins will be in passive mode.
HPS JTAG Bypass	EXTERNAL_JTAG_BYPASS	Bypasses the HPS JTAG controller and disables the HPS internal master control.
PR and Scrubbing Disable	PR_SCRUBBING_DISABLE	Disables partial reconfiguration and external scrubbing from external pins and HPS. Only the FPGA core can perform partial reconfiguration.
Volatile Key Lock	VOLKEY_LOCK	Locks the volatile key being zeroed-out or reprogrammed. However, you can erase the volatile key using KEY_CLR_VREG instruction. You can issue the VOLKEY_LOCK instruction only after volatile key is programmed into the device.

continued...

(40) Enabling the JTAG Secure or Test Disable mode disables the test mode in Intel Arria 10 devices and disables programming through the JTAG interface. This process is irreversible and prevents Intel from carrying out failure analysis.

Security Mode	JTAG Instruction	Security Feature
Volatile Key Disable	VOLKEY_DISABLE	Disables any future volatile key programming. If there is an existing volatile key programmed into the device, it will not be used to decrypt the configuration file.
Non-Volatile Key Disable	OTP_DISABLE	Disables any future non-volatile key programming. If there is an existing non-volatile key programmed into the device, it will not be used to decrypt the configuration file.
Test Disable Mode	TEST_DISABLE	Disables all test modes and all test-related JTAG instructions. This process is irreversible and prevents Intel from carrying out failure analysis.

Related Information

SoC Security of the Intel Arria 10 Hard Processor System Technical Reference Manual
Provides more information about HPS Configuration Only and HPS JTAG Bypass security modes.

7.5.2.1. JTAG Secure Mode

When the Intel Arria 10 device is in the JTAG Secure mode, all JTAG instructions except for the mandatory IEEE Standard JTAG 1149.1 BST JTAG instructions are disabled.

Table 106. Mandatory and Non-Mandatory IEEE Standard 1149.1 BST JTAG Instructions

Mandatory IEEE Standard 1149.1 BST JTAG Instructions	Non-Mandatory IEEE Standard 1149.1 BST JTAG Instructions
• BYPASS • EXTEST • IDCODE • SAMPLE/PRELOAD • SHIFT_EDERROR_REG	• CONFIG_IO • CLAMP • EXTEST_PULSE(41) • EXTEST_TRAIN(41) • HIGHZ • KEY_CLR_VREG • KEY_VERIFY(41) • PULSE_NCONFIG • USERCODE

Note: After you issue the EXT_JTAG_SECURE instruction, the Intel Arria 10 device cannot be unlocked.

Related Information

Supported JTAG Instruction on page 300

7.5.3. Intel Arria 10 Qcrypt Security Tool

The Qcrypt tool is a stand-alone encryption tool for encrypting and decrypting Intel Arria 10 FPGA configuration bit-stream files. The Qcrypt tool can also be used to encrypt HPS boot images through a script. Different kinds of security settings that are currently not accessible from the Intel Quartus Prime graphical user interface can be set through the Qcrypt tool.

(41) You can execute these JTAG instructions during JTAG Secure mode.

The Qcrypt tool encrypts and decrypts raw binary files (`.rbf`) only and not other configuration files, such as `.sof` and `.pof` files. Throughout the encryption flow, the Qcrypt tool generates an authentication tag while encrypting the `.rbf` file. The authentication tag prevents any modification or tampering of the configuration bit-stream. Besides encryption and decryption, the Qcrypt tool allows you to enable and set various security features and settings. By incorporating security features and settings into the `.rbf` file, you have the flexibility to use different kinds of security features on Intel Arria 10 devices without permanently burning the security fuses. To generate the `.ekp` file or encrypted configuration file other than `.rbf`, you have to use the Intel Quartus Prime Convert Programming File tool.

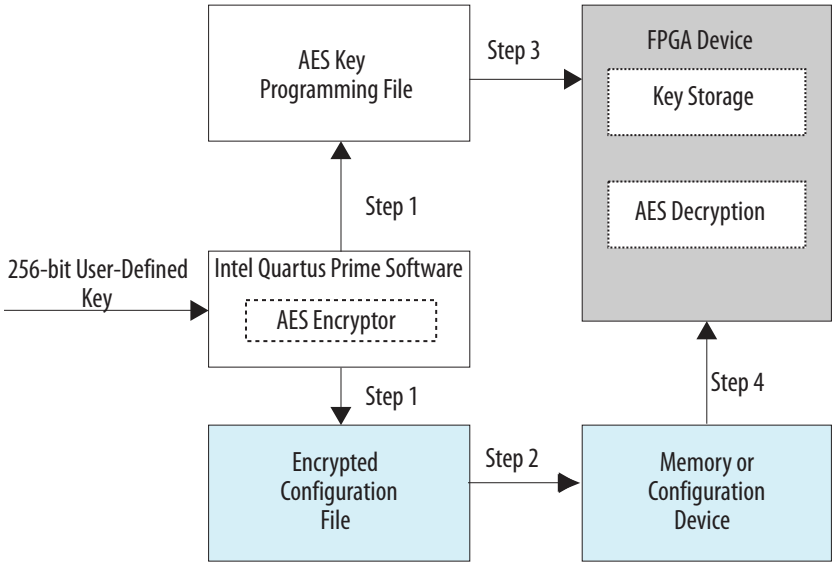
Note: The Qcrypt tool is not license-protected and can be used by all Intel Quartus Prime software user.

Related Information

- Qcypt Tool Options of the AN 556: Using the Design Security Features in Intel FPGAs
  Provides more information about Qcrypt tool features.
- AN 759: Arria 10 SoC Secure Boot User Guide
  Provides more information about encrypting HPS boot images.
- AN 556: Using the Design Security Features in Intel FPGAs
  Provides more information about applying design security features in Intel Arria 10 devices.

7.5.4. Design Security Implementation Steps

Figure 162. 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 Intel Arria 10 device through a JTAG interface.
4. Configure the Intel Arria 10 device. At the system power-up, the external memory device sends the encrypted configuration file to the Intel Arria 10 device.

Related Information

AN 556: Using the Design Security Features in Intel FPGAs
Provides more information about applying design security features in Intel Arria 10 devices.

7.6. Configuration, Design Security, and Remote System Upgrades in Intel Arria 10 Devices Revision History

Document Version	Changes
2020.06.30	Updated the *Active Serial Multi-Device Configuration* section on determining the DCLK frequency for slave devices.
2020.03.31	Updated the *User Mode* topic.
2019.12.30	Added note to clarify users should use 12 mA drive strength IBIS model when simulating the AS_DATA[0]/ASDO, AS_DATA[3:1], and DCLK configuration pins for 1.8V LVCMOS I/O standard in the *I/O Standards and Drive Strength for Configuration Pins* table.
2019.03.28	Updated the note in the *Trace Length Guideline* topic.
2019.01.23	Added a link to the Intel Supported Configuration Devices section of the *Device Configuration - Support Center* page on the Intel website.
2019.01.11	• Added a note in *CLKUSR* to state that CLKUSR cannot be used as the reference clock for PLL. • Updated Table: *Configuration Pin Summary for Intel Arria 10 Devices* to update the note for nIO_PULLUP.
2018.08.28	• Added *Multiple Configuration Devices Support* section. • Updated the description in the *Using Multiple Configuration Data* section for active serial multi-device configuration. • Updated the description in the *Using Multiple Configuration Data* section for fast passive parallel multi-device configuration. • Updated Table: *I/O Standards and Drive Strength for Configuration Pins*.
2018.03.09	Updated *Configuration Pin Summary for Intel Arria 10 Devices* table to indicate CLKUSR and nCEO pins are optional.

Date	Version	Changes
December 2017	2017.12.15	• Removed LOCK and UNLOCK instructions from *Mandatory IEEE Standard 1149.1 BST JTAG Instructions*. • Updated *Single Device FPP Configuration Using an External Host*, *Multiple Device FPP Configuration Using an External Host When Both Devices Receive a Different Set of Configuration Data*, and *Multiple Device FPP Configuration Using an External Host When Both Devices Receive the Same Data* figures.
March 2017	2017.03.15	• Rebranded as Intel. • Added note to nIO_PULLUP pin in *Configuration Pin Summary for Intel Arria 10 Devices* table. • Added pull-up resistor between nCEO and nCE in *Multiple Device FPP Configuration Using an External Host When Both Devices Receive a Different Set of Configuration Data* figure. • Added pull-up resistor between nCEO and nCE in *Multiple Device AS Configuration When Both Devices in the Chain Receive Different Sets of Configuration Data* figure. • Added pull-up resistor between nCEO and nCE in *Multiple Device PS Configuration when Both Devices Receive Different Sets of Configuration Data* figure.
October 2016	2016.10.31	• Updated the drive strength for configuration pins: — DCLK—from 1.8 V CMOS 12 mA to 1.8 V CMOS 24 mA. — NCSO[2..0]—from 1.8 V CMOS 8 mA to 1.8 V CMOS 12 mA. — AS_DATA0/ASD0, AS_DATA1, AS_DATA2, and AS_DATA3—from 1.8 V CMOS 8 mA to 1.8 V CMOS 24 mA.
June 2016	2016.05.13	• Updated design security features and approaches. • Updated instances of EX_JTAG_SECURE to EXT_JTAG_SECURE • Added list of mandatory and non-mandatory IEEE Standard 1149.1 JTAG BST instructions. • Updated security modes available in Arria 10 devices and instructions to enable them. • Added the Qrypt secuity tool information.
May 2016	2016.05.02	• Added FPP and PS configuration time estimation to *Estimating Configuration Time* and moved subsection under *Configuration Details* section. • Added note on possible PCIe timing violation when using direct-to-application. • Added note on recommending user to set a fixed configuration image start address. • Added *I/O Standards and Drive Strength for Configuration Pins* section. • Updated AS configuration timing waveform to include nCSO. • Updated T_{SU} and T_{DH} in AS configuration timing waveform.
December 2015	2015.12.14	• Updated CLKUSR information. • Moved *CLKUSR* subsection from *Active Serial Configuration* to *Configuration Details*.
November 2015	2015.11.02	• Updated the term configuration mode to configuration scheme for consistency. • Added link at MSEL pin setting to Arria 10 Hard Processor System Technical Reference Manual. • Combined PS and FPP row in MSEL pin settings table. Both schemes have the same MSEL pin setting. • Added description to MSEL pin setting table for configuration via HPS to use PS or FPP MSEL pin setting. • Updated configuration modes and features table to include Yes for partial reconfiguration in JTAG, AS and PS configuration mode together with a footnote mentioning only if partial reconfiguration is configured as internal host.

continued...

Date	Version	Changes
		• Updated note about compression and encryption cannot be used a the same time for all configuration scheme. • Updated timing waveforms FPP, AS and PS to include pre power-up state. • Removed step to select **Remote** from the Configuration mode list in the **Configuration** page of the **Device and Pin Options** dialog box in the Quartus II software. • Added note about setting the MSEL pin to active serial to prevent EPCQ-L ID read failure during SFL programming. • Changed instances of *Quartus II* to *Quartus Prime*.
May 2015	2015.05.04	• Added Timing waveforms for FPP, AS and PS configuration. • Updated 'Trace Length and Loading' to 'Trace Length Guideline' and remove loading contents. • Added link to Arria 10 Device Datasheet for loading information. • Update FPP to support 8 and 32 bits in 'Configuration Modes and Features of Arria 10 Devices'. • Added note in 'Design Security' and 'Configuration Data Compression' about compression and encryption cannot be used a the same time.
January 2015	2015.01.23	• Updated CLKUSR pin usage during AS configuration at 100 MHz. • Updated Max clock rate of PS, FPP x8, FPP x16 and Configuration via HPS from 125 MHz to 100 MHz. • Updated Remote System Upgrade Circuitry diagram by replacing RU_SHIFTnLD and RU_CAPTnUPDT to RU_CTL[1:0]. • Updated ALTREMOTE_UPDATE megafunction to Altera remote Update IP Core. • Updated user watchdog time-out value from 34..46 to 34..45. • Updated nIO_PULLUP to be powered by V_{CC}. • Added note to Max Data Rate in Configuration Modes and Features of Intel Arria 10 Devices table.
August 2014	2014.08.18	• Added the Active Serial Configuration with Multiple EPCQ-L Devices section. • Removed the Unique Chip ID section. • Updated the JTAG Configuration section to include details on the USB-Blaster download cable support. • Updated the Power Up section. • Updated Configuration Images section to include start address. • Updated the Configuration Sequence in the Remote Update Mode section. • Updated the Remote System Upgrade State Machine section. • Updated Figure 7-18: JTAG Configuration of a Single Device Using a Microprocessor to update the power reference of the JTAG pins. • Updated Figure 7-20: Configuration Sequence for Arria 10 Devices. • Updated Figure 7-22: Arria 10 Remote System Upgrade Block Diagram. • Updated Table 7-1: Configuration Modes and Features of Arria 10 Devices to update the supported clock rate for Partial Reconfiguration. • Updated Table 7-3: MSEL Pin Settings for Each Configuration Scheme of Arria 10 Devices to include the supported V_{CCPGM} voltages for the FPP and PS configuration schemes. • Updated Table 7-6: Remote System Upgrade Registers to update the description for the shift, control, update, and status registers. • Updated Table 7-7: Control Register Bits. • Removed the Unique Chip ID section.
December 2013	2013.12.02	Initial release.

8. SEU Mitigation for Intel Arria 10 Devices

8.1. Intel Arria 10 SEU Mitigation Overview

Single event upsets (SEUs) are rare, unintended changes in the state of an FPGA's internal memory elements caused by cosmic radiation effects. The change in state is a soft error and the FPGA incurs no permanent damage. Because of the unintended memory state, the FPGA may operate erroneously until background scrubbing fixes the upset.

The Intel Quartus Prime software offers several features to detect and correct the effects of SEU, or soft errors, as well as to characterize the effects of SEU on your designs. Additionally, some Intel FPGAs contain dedicated circuitry to help detect and correct errors.

Intel FPGAs have memory in user logic (block memory and registers) and in Configuration Random Access Memory (CRAM). The Intel Quartus Prime Programmer loads the CRAM with a `.sof` file. Then, the CRAM configures all FPGA logic and routing. If an SEU strikes a CRAM bit, the effect can be harmless if the device does not use the CRAM bit. However, the effect can be severe if the SEU affects critical logic or internal signal routing.

Often, a design does not require SEU mitigation because of the low chance of occurrence. However, for highly complex systems, such as systems with multiple high-density components, the error rate may be a significant system design factor. If your system includes multiple FPGAs and requires very high reliability and availability, you should consider the implications of soft errors. Use the techniques in this chapter to detect and recover from these types of errors.

Related Information

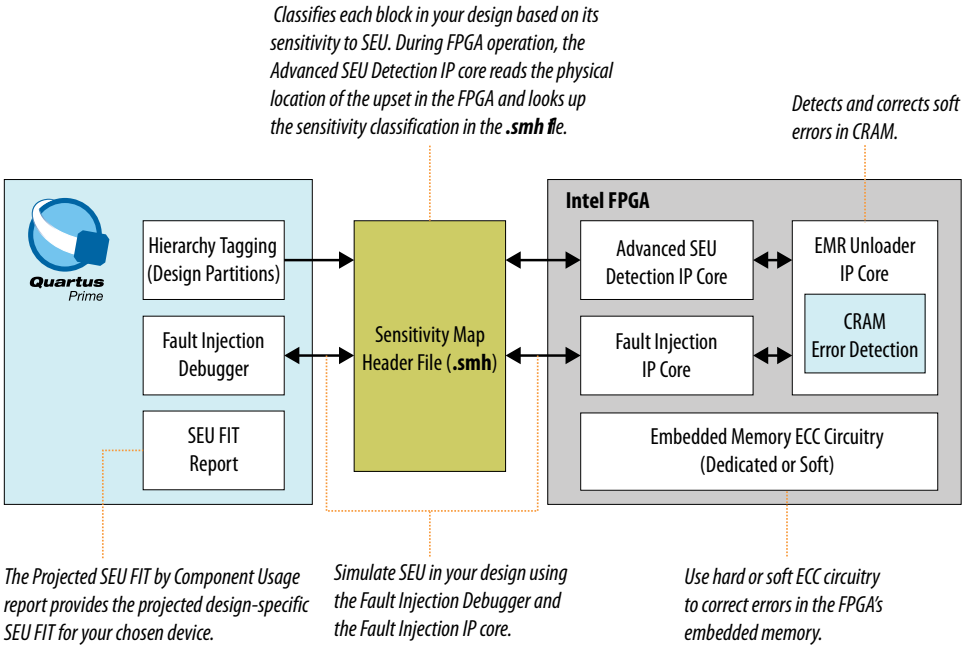
- Introduction to Single-Event Upsets
- Understanding Single Event Functional Interrupts in FPGA Designs
- Intel Arria 10 Device Handbook: Known Issues
  Lists the planned updates to the *Intel Arria 10 Device Handbook* chapters.
- AN 737: SEU Detection and Recovery in Intel Arria 10 Devices
  Describes the implementation of Intel Arria 10 SEU detection and recovery with a reference design.

8.1.1. Mitigating Single Event Upset

Single event upsets (SEUs) are rare, unintended changes in the state of an FPGA's internal memory elements caused by cosmic radiation effects. The change in state is a soft error and the FPGA incurs no permanent damage. Because of the unintended memory state, the FPGA may operate erroneously until background scrubbing fixes the upset.

The Intel Quartus Prime software offers several features to detect and correct the effects of SEU, or soft errors, as well as to characterize the effects of SEU on your designs. Additionally, some Intel FPGAs contain dedicated circuitry to help detect and correct errors.

Figure 163. Tools, IP, and Circuitry for Detecting and Correcting SEU



Intel FPGAs have memory in user logic (block memory and registers) and in Configuration Random Access Memory (CRAM). The Intel Quartus Prime Programmer loads the CRAM with a `.sof` file. Then, the CRAM configures all FPGA logic and routing. If an SEU strikes a CRAM bit, the effect can be harmless if the device does not use the CRAM bit. However, the effect can be severe if the SEU affects critical logic or internal signal routing.

Often, a design does not require SEU mitigation because of the low chance of occurrence. However, for highly complex systems, such as systems with multiple high-density components, the error rate may be a significant system design factor. If your system includes multiple FPGAs and requires very high reliability and availability, you should consider the implications of soft errors. Use the techniques in this chapter to detect and recover from these types of errors.

8.1.1.1. Configuration RAM

FPGAs use memory both in user logic (bulk memory and registers) and in Configuration RAM (CRAM). CRAM is the memory loaded with the user's design. The CRAM configures all logic and routing in the device. If an SEU strikes a CRAM bit, the effect can be harmless if the CRAM bit is not in use. However, a functional error is possible if it affects critical internal signal routing or critical lookup table logic bits as part of the user's design.

8.1.1.2. Embedded Memory

The Intel Arria 10 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 and shallow FIFO buffers, and filter delay lines. Each MLAB is made up of ten adaptive logic modules (ALMs). In the Intel Arria 10 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.

Embedded memory is susceptible to SEU, Intel implement interleaving and special layout techniques to minimize the FIT rate, and added Error Correction Code (ECC) feature to reduce SEU FIT rate to close to zero.

Related Information

Embedded Memory Blocks in Intel Arria 10 Devices

8.1.1.3. Failure Rates

The Soft Error Rate (SER) or SEU reliability is expressed in Failure in Time (FIT) units. One FIT unit is one soft error occurrence per billion hours of operation.

- For example, a design with 5,000 FIT experiences a mean of 5,000 SEU events in one billion hours (or 114,155.25 years). Because SEU events are statistically independent, FIT is additive. If a single FPGA has 5,000 FIT, then ten FPGAs have 50,000 FIT (or 50K failures in 114,155.25 years).

Another reliability measurement is the mean time to failure (MTTF), which is the reciprocal of the FIT or 1/FIT.

- For a FIT of 5,000 in standard units of failures per billion hours, MTTF is:

  1 ÷ (5,000 ÷ 1 Bh)=1 billion ÷ 5,000 = 200,000 hours = 22.83 years

SEU events follow a Poisson distribution and the cumulative distribution function (CDF) for mean time between failures (MTBF) is an exponential distribution. For more information about failure rate calculation, refer to the *Intel FPGA Reliability Report.*

Neutron SEU incidence varies by altitude, latitude, and other environmental factors. The Intel Quartus Prime software provides SEU FIT reports based on compiles for sea level in Manhattan, New York. The JESD89A specification defines the test parameters.

Tip: You can convert the data to other locations and altitudes using calculators, such as those at www.seutest.com. Additionally, you can adjust the SEU rates in your project by including the relative neutron flux (calculated at www.seutest.com) in your project's `.qsf` file.

8.2. Intel Arria 10 SEU Mitigation Techniques

Intel Arria 10 devices feature various single-event upset (SEU) mitigation approaches for different application areas.

Table 107. SEU Mitigation Areas and Approaches for Intel Arria 10 Devices

Area	SEU Mitigation Approach
Silicon design: CRAM/SRAMs/flip flops	Intel uses various design techniques to reduce upsets or limit to correctable double-bit errors.
Error Detection Cyclic redundancy check (EDCRC) / Scrubbing	You can enable the EDCRC feature for detecting CRAM SEU events and automatic correction of CRAM contents.
M20K SRAM block	Intel FPGA implements interleaving, special layout techniques, and Error Correction Code (ECC) to reduce SEU FIT rate to almost zero.
Sensitivity processing	You can use sensitivity processing to identify if the SEU in CRAM bit is a used or unused bit.
Fault injection	You can use fault injection feature to validate the system response to the SEU event by changing the CRAM state to trigger an error.
Hierarchical tagging	A complementary capability to sensitivity processing and fault injection for reporting SEU and constraining injection to specific portions of design logic.
Triple Modular Redundancy (TMR)	You can implement TMR technique on critical logic such as state machines.

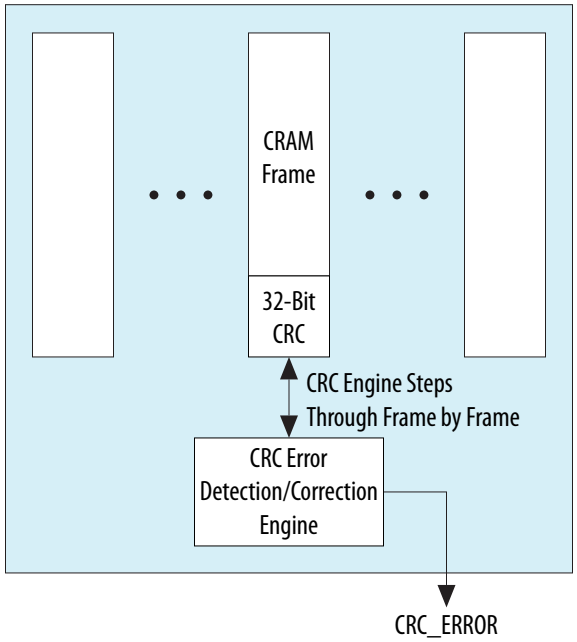
8.2.1. Mitigating SEU Effects in Configuration RAM

Intel Arria 10 devices contain error detect CRC (EDCRC) hard blocks. These blocks detect and correct soft errors in CRAM, and are similar to those that protect internal user memory.

Intel FPGAs contain frames of CRAM. The size and number of frames is device specific. The device continually checks the CRAM frames for errors by loading each frame into a data register. The EDCRC block checks the frame for errors.

When the FPGA finds a soft error, the FPGA asserts its CRC_ERR0R pin. You can monitor this pin in your system. When your system detects that the FPGA asserted this pin during operation, indicating the FPGA detected a soft error in the configuration RAM, the system can take action to recover from the error. For example, the system can perform a soft reset (after waiting for background scrubbing), reprogram the FPGA, or classify the error as benign and ignore it.

Figure 164. CRAM Frame



To enable error detection, point to **Assignments ➤ Device ➤ Device and Pin Options ➤ Error Detection CRC**, and turn on error detection settings.

8.2.1.1. Error Detection Cyclic Redundancy Check

In user mode, the contents of the configured configuration RAM (CRAM) bits can be affected by soft errors. These soft errors, which are caused by an ionizing particle, are not common in Intel FPGA devices. However, high-reliability applications that require error-free device operation may require your design to consider these errors.

The hardened on-chip EDCRC circuitry allows you to perform the following operations without any impact on the fitting or performance of the device:

- Auto-detection of cyclic redundancy check (CRC) errors during configuration.
- Optional soft errors (SEU and multiple bit upset) detection and identification in user mode.
- Fast soft error detection. The error detection speed is improved.
- Two types of check-bits:
  - Frame-based check-bits—stored in CRAM and used to verify the integrity of the frame.
  - Column-based check-bits—stored in registers and used to protect integrity of all frames.

During error detection in user mode, a number of EDCRC engines run in parallel for Intel Arria 10 devices. The number of error detection CRC engines depends on the frame length—total bits in a frame.

Each column-based error detection CRC engine reads 128 bits from each frame and processes within four cycles. To detect errors, the error detection CRC engine needs to read back all frames.

Figure 165. Block Diagram for Error Detection in User Mode

The block diagram shows the registers and data flow in user mode.

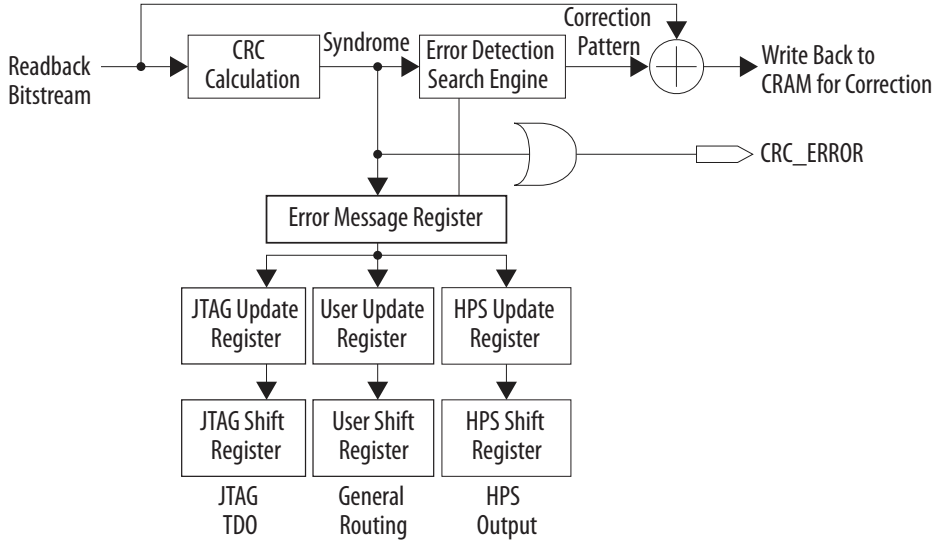


Table 108. Error Detection Registers

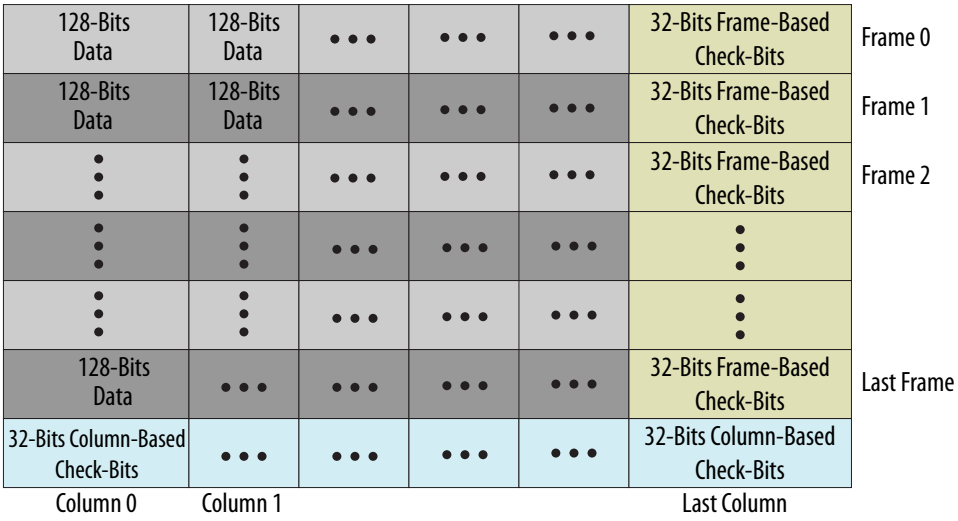
Name	Description
Error message registers (EMR)	Contains error details for single-bit and double-adjacent errors. The error detection circuitry updates this register each time the circuitry detects an error.
User update register	This register is automatically updated with the contents of the EMR one clock 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	This register allows user logic to access the contents of the user update register via the core interface. You can use the Error Message Register Unloader Intel FPGA IP core to shift-out the EMR information through user shift register. For more information, please refer to related information.
JTAG update register	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	This register allows you to access the contents of the JTAG update register via the JTAG interface using the `SHIFT_EDERROR_REG` JTAG instruction.
Hard Processor System (HPS) update register	This register is automatically updated with the contents of the EMR one clock cycle after the content of this register is validated. The (HPS) update register includes a clock enable, which must be asserted before its contents are written to the HPS shift register. This requirement ensures that the HPS update register is not overwritten when its contents are being read by the HPS shift register.
HPS shift register	This register allows you to access the contents of the HPS update register via the HPS interface.

Related Information

- Error Message Register Unloader Intel FPGA IP Core User Guide
  Provides more information about using the user shift register to shift-out the EMR.
- FPGA Manager Address Map and Register Definitions in Intel Arria 10 Hard Processor System Technical Reference Manual
  Provides more information about using hard processor system to read the error detection registers.

8.2.1.1.1. Column-Based and Frame-Based Check-Bits

Figure 166. Column-Based and Frame-Based Check-Bits



EDCRC Check-Bits Updates

Frame-based check-bits are calculated on-chip during configuration. Column-based check-bits are updated after configuration.

When you enable the EDCRC feature, after the device enters user mode, the EDCRC function starts reading CRAM frames. The data collected from the read-back frame is validated against the frame-based check-bits.

After the initial frame-based verification is completed, the column-based check-bits is calculated based on the respective column CRAM. The EDCRC hard block recalculates the column-based check-bits in one of the following scenarios:

- FPGA re-configuration
- After successful partial reconfiguration (PR) session
- After configuration via protocol (CvP) session

8.2.1.1.2. Error Message Register

The EMR contains information on the error type, the location of the error, and the actual syndrome. This register is 78 bits wide in Intel Arria 10 devices. The EMR does not identify the location bits for uncorrectable errors. The location of the errors consists of the frame number, double word location and bit location within the frame and column.

You can shift out the contents of the register through the following:

- EMR Unloader IP core—core interface
- SHIFT_EDERR0R_REG JTAG instruction—JTAG interface
- HPS Shift register—HPS interface

Figure 167. Error Message Register Map

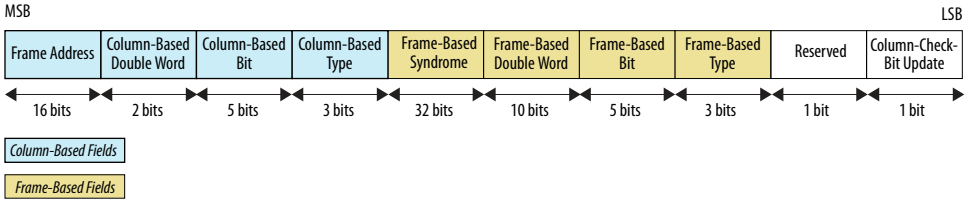


Table 109. Error Message Register Width and Description

Name	Width (Bits)	Description
Frame Address	16	Frame Number of the error location
Column-Based Double Word	2	There are 4 double words per frame in a column. It indicates the double word location of the error
Column-Based Bits	5	Error location within 32-bit double word
Column-Based Type	3	Types of error shown in Table 110 on page 284
Frame-Based syndrome register	32	Contains the 32-bit CRC signature calculated for the current frame. If the CRC value is 0, the CRC_ERROR pin is driven low to indicate no error. Otherwise, the pin is pulled high.
Frame-Based Double Word	10	Double word location within the CRAM frame.
Frame-Based Bit	5	Error location within 32-bit double word
Frame-Based Type	3	Types of error shown in Table 110 on page 284
Reserved	1	Reserved bit
Column-Based Check-Bits Update	1	Logic high if there is error encountered during the column check-bits update stage. The CRC_ERR0R pin will be asserted and stay high until the FPGA is reconfigured.

Retrieving Error Information

You can retrieve the EMR contents via the core interface or the JTAG interface using the SHIFT_EDERR0R_REG JTAG instruction. Intel provides the Error Message Register Unloader IP Core that unload EMR content via core interface and allows it to be shared between several design component.

Related Information

Error Message Register Unloader Intel FPGA IP Core User Guide
Provides more information about using the user shift register to shift-out the EMR.

Error Type in EMR

Table 110. Error Type in EMR

The following table lists the possible error types reported in the error type field in the EMR.

Error Types	Bit 2	Bit 1	Bit 0	Description
Frame-based	0	0	0	No error
	0	0	1	Single-bit error
	0	1	X	Double-adjacent error
	1	1	1	Uncorrectable error
Column-Based	0	0	0	No error
	0	0	1	Single bit error
	0	1	X	Double-adjacent error in a same frame
	1	0	X	Double-adjacent error in a different frame
	1	1	0	Double-adjacent error in a different frame
	1	1	1	Uncorrectable error

8.2.1.1.3. CRC_ERROR Pin Behavior

The Intel Arria 10 fast EDCRC feature runs all the column-based check-bits engine in parallel. When an SEU is detected, the column-based check-bits asserts the CRC_ERROR, the detected frame location is then passed to the frame-based check-bits to further localize the affected bit. This process causes the CRC_ERROR pin to assert twice. Column-based check-bits assert the first CRC_ERROR pulse and followed by the frame-based check-bits asserting the second pulse.

In Intel Arria 10, as soon as an SEU is detected, the CRC_ERROR is asserted high and remains high until the EMR is ready to be read. You can unload the EMR data as soon as the CRC_ERROR pin goes low. Once EMR data is unloaded, can determine the error type and the affected location. With these information you can decide how your system should respond to the specific SEU event.

Figure 168. Fast EDCRC Process Flow Chart

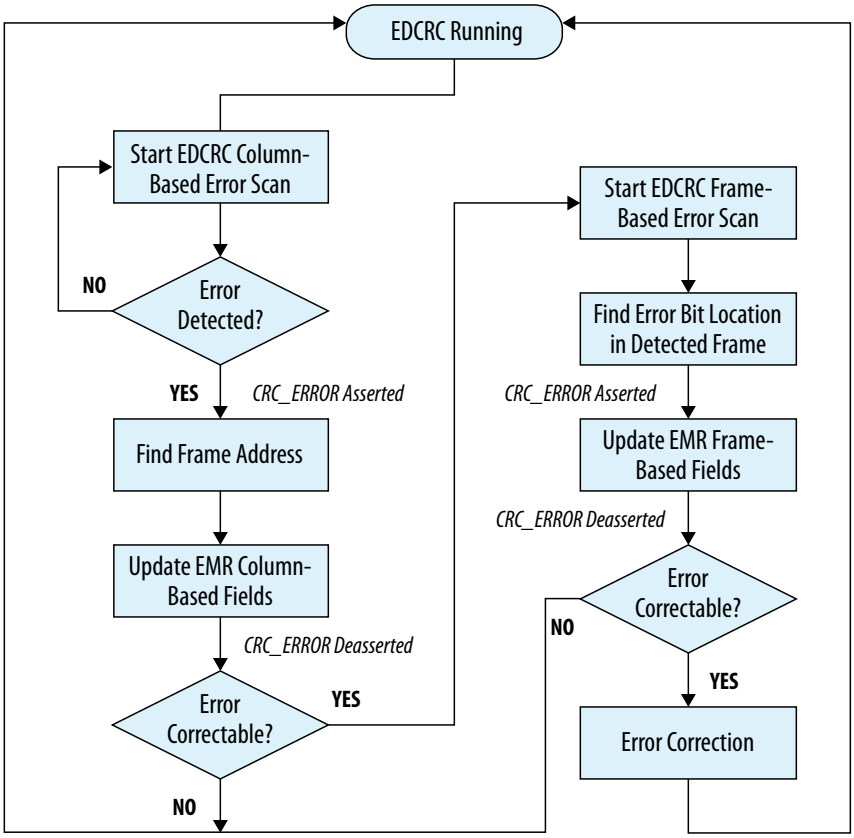
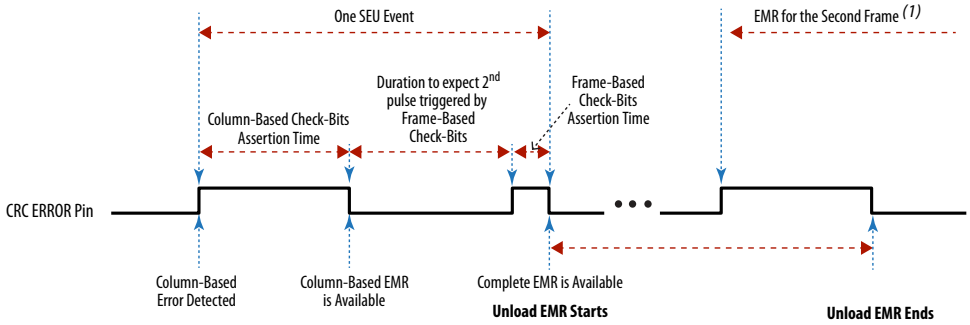


Figure 169. Timing Diagram for Column-Based Check-Bits

If the error is correctable, in most cases, there is a second pulse in a single SEU event .There are cases where the error is uncorrectable when the CRC_ERROR pin asserts 2 pulses, refer to Correctable and Uncorrectable Error for complete correctable and uncorrectable error cases. The complete EMR is only available at the falling edge of the second pulse.

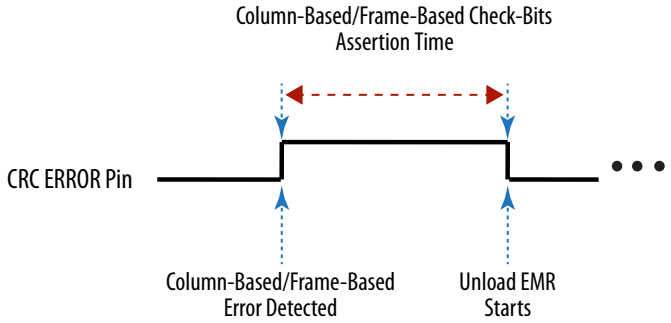


(1) In a rare event of correctable double-adjacent error located in different frames.

In the rare event of an uncorrectable and un-locatable error, the CRC_ERROR signal is asserted only once. There is no second pulse assertion by frame-based check-bits due to the uncorrectable error location cannot be located. The statistical likelihood of uncorrectable multi-bit SEU is less than one in 10,000 years for a device in typical environmental conditions.

Figure 170. Timing Diagram for Column-Based or Frame-Based Check-Bits

Example of CRC_ERROR pin behavior for column-based/frame-based check-bits with a single pulse observed in one SEU event.



Related Information

Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
Provides more information about CRC_ERROR connection guidelines.

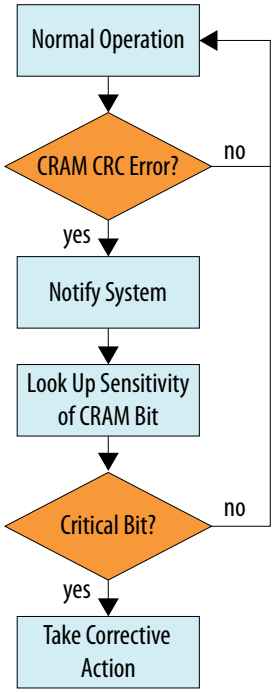
8.2.1.2. SEU Sensitivity Processing

Reconfiguring a running FPGA has a significant impact on the system using the FPGA. When planning for SEU recovery, account for the time required to bring the FPGA to a state consistent with the current state of the system. For example, if an internal state machine is in an illegal state, it may require reset. In addition, the surrounding logic may need to account for this unexpected operation.

Often an SEU impacts CRAM bits not used by the implemented design. Many configuration bits are not used because they control logic and routing wires that are not used in a design. Depending on the implementation, 40% of all CRAM bits can be used even in the most heavily utilized devices. This means that only 40% of SEU events require intervention, and you can ignore 60% of SEU events. The utilized bits are considered as critical bits while the non-utilized bits are considered as non-critical bits.

You can determine that portions of the implemented design are not utilized in the FPGA's function. Examples may include test circuitry implemented but not important to the operation of the device, or other non-critical functions that may be logged but do not need to be reprogrammed or reset.

Figure 171. Sensitivity Processing Flow



8.2.1.3. Hierarchy Tagging

Hierarchy tagging is the process of classifying the sensitivity of the portions of your design.

You can perform hierarchy tagging using the Intel Quartus Prime software by creating a design partition, and then assigning the parameter Advanced SEU Detection (ASD) Region to that partition. The parameter can assume a value from 0 to 15, so there are 16 different classifications of system responses to the portions of your design.

The design hierarchy sensitivity processing depends on the contents of the Sensitivity Map Header file (`.smh`). This file determines which portion of the FPGA's logic design is sensitive to a CRAM bit flip. You can use sensitivity information from the `.smh` file to determine the correct (least disruptive) recovery sequence.

To generate the functionally valid `.smh`, you must designate the sensitivity of the design from a functional logic view, using the hierarchy tagging procedure.

Related Information

Advanced SEU Detection Intel FPGA IP Core User Guide
Provides more information about hierarchy tagging using Advanced SEU Detection Intel FPGA IP core.

8.2.1.4. Evaluating Your System's Response to Functional Upsets

The ratio of SEU strikes versus functional interrupts is the Single Event Functional Interrupt (SEFI) ratio. Minimizing this ratio improves SEU mitigation. SEUs can randomly strike any memory element, system testing is important to ensure a comprehensive recovery response.

You can use fault injection to aid in SEU recovery response. The fault injection feature allows you to operate the FPGA in your system and inject random CRAM bit flips to test the ability of the FPGA and the system to detect and recover fully from an SEU. You should be able to observe your FPGA and your system recover from these simulated SEU strikes. You can then refine your FPGA and system recovery sequence by observing these strikes. You can determine the SEFI rate of your design by using the fault injection feature.

Related Information

Fault Injection Intel FPGA IP Core User Guide
Provides more information about injecting soft error to simulate SEU using Fault Injection Intel FPGA IP Core.

8.2.1.5. Recovering from CRC Errors

Intel Arria 10 devices support the internal scrubbing capability. The internal scrubbing feature corrects correctable CRAM upsets automatically when an upset is detected. However, internal scrubbing can not fix the FPGA to a known good state. The time between the error and completion of scrubbing can be tens of millisecond. This duration represents thousands of clock cycles in which the corrupted data was written to memory or status registers. It is a good practice to always follow any SEU event with a soft-reset to bring the FPGA operation to a known good state.

If a soft-reset is unable to bring the FPGA to a known good state, you can reconfigure the device to rewrite the CRAM and reinitialize the design registers. The system that hosts the Intel Arria 10 device must control the device reconfiguration. When reconfiguration completes successfully, the Intel Arria 10 device operates as intended.

Related Information

Configuration, Design Security, and Remote System Upgrades for Intel Arria 10 Devices
Provides more information about configuration sequence.

8.2.1.5.1. Enabling Error Correction (Internal Scrubbing)

Intel Arria 10 supports the internal scrubbing feature to automatically scrub away the flipped bit induced by the SEU. To enable the internal scrubbing feature, follow these steps:

1. On the **Assignments** menu, click **Device**.
2. Click **Device and Pin Options** and select the **Error Detection CRC** tab.
3. Turn on **Enable internal scrubbing**.
4. Click **OK**.

8.2.2. Mitigating SEU Effects in Embedded User RAM

You can reduce the FIT rate for these memories to near zero by enabling the ECC encode/decode blocks. On ingress, the ECC encoder adds 8 bits of redundancy to a 32 bit word. On egress, the decoder converts the 40 bit word back to 32 bits. You use the redundant bits to detect and correct errors in the data resulting from SEU.

The existence of hard ECC and the strength of the ECC code (number of corrected and detected bits) varies by device family. Refer to the device handbook for details. If a device does not have a hard ECC block you can add ECC parity or use an ECC IP core.

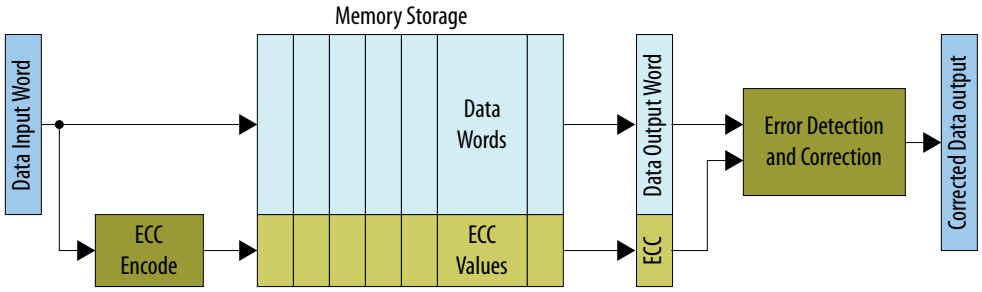
The SRAM memories associated with processor subsystems, such as for SoC devices, contain dedicated hard ECC. You do not need to take action to protect these memories.

8.2.2.1. Configuring RAM to Enable ECC

To enable ECC, configure the RAM as a 2-port RAM with independent read and write addresses. Using this feature does not reduce the available logic.

Although the ECC checking function results in some additional output delay, the hard ECC has a much higher $\mathbf{f_{MAX}}$ compared with an equivalent soft ECC block implemented in general logic. Additionally, you can pipeline the hard IP in the M20K block by configuring the ECC-enabled RAM to use an output register at the corrected data output port. This implementation increases performance and adds latency.
For devices without dedicated circuitry, you can implement ECC by instantiating the ALTECC IP core, which performs ECC generation and checking functions.

Figure 172. Memory Storage and ECC



8.2.3. Triple-Module Redundancy

Use Triple-Module Redundancy (TMR) if your system cannot suffer downtime due to SEU. TMR is an established SEU mitigation technique for improving hardware fault tolerance. A TMR design has three identical instances of hardware with voting hardware at the output. If an SEU affects one of the hardware instances, the voting logic notes the majority output. This operation masks malfunctioning hardware.

With TMR, your design does not suffer downtime in the case of a single SEU; if the system detects a faulty module, the system can scrub the error by reprogramming the module. The error detection and correction time is many orders of magnitude less than the MTBF of SEU events. Therefore, the system can repair a soft interrupt before another SEU affects another instance in the TMR application.

The disadvantage of TMR is its hardware resource cost: it requires three times as much hardware in addition to voting logic. You can minimize this hardware cost by implementing TMR for only the most critical parts of your design.

There are several automated ways to generate TMR designs by automatically replicating designated functions and synthesizing the required voting logic. Synopsys offers automated TMR synthesis.

8.2.4. Intel Quartus Prime Software SEU FIT Reports

The Intel Quartus Prime software generates reports that contain the parameters involved in SEU FIT calculations and the result of these calculations for each component. These reports are available only for licensed users.

8.2.4.1. SEU FIT Parameters Report

The SEU FIT Parameters report shows the environmental assumptions that influence the FIT/Mb values.

Figure 173. SEU FIT Parameters

SEU FIT Parameters

	Parameter	value
1	Device	5SGXEA7N2F45I2
2	Altitude	0.00
3	Neutron Flux	JESD - 89A assuming sea - level(> 10 MeV) 13.00 n / hr / cm2
4	Neutron Flux Multiplier	1.000
5	Alpha Flux	0.001 CPH/cm^2

Change the Neutron Flux Multiplier using the assignment:
set_global_assignment RELATIVE_NEUTRON_FLUX <relative_flux>

- **Altitude** represents the default altitude (above sea-level).
- **Neutron Flux Multiplier** is the relative flux for the default location, which is New York City per JESD specification. The default is 1. Change the setting by adding the following assignment to your `.qsf` file:

```
set_global_assignment RELATIVE_NEUTRON_FLUX <relative_flux>
```

  Note: You can compute scaled values using the JESD published equations for altitude, latitude, and longitude. Websites, such as www.seutest.com, can make this computation for you.
- **Alpha Flux** is the default for standard Intel packages; you cannot override the default.

  Note: When you change the relative **Neutron Flux Multiplier**, the Intel Quartus Prime software only scales the neutron component of FIT. Location does not affect the Alpha flux.

8.2.4.2. Projected SEU FIT by Component Usage Report

The Projected SEU FIT by Component Usage report shows the different components (or cell types) that comprise the total FIT rate, and SEU FIT calculation results.

An Intel FPGA's sensitivity to soft errors varies by process technology, component type, and your design choices when implementing the component (such as tradeoffs between area/delay and SEU rates). The report shows all bits (the raw FIT), utilized bits (only resources the design actually uses), and the ECC-mitigated bits.

Figure 174. Projected SEU FIT by Component Usage Report

Projected SEU FIT by Component Usage

	Component	Raw	Utilized	w/ECC	AVF 0.5	AVF 0.25
1	Configuration (CRAM)	8486	3817	3817	1909	955
1	--Logic	2125	1071	1071	536	268
2	--Routing	6314	2716	2716	1358	679
3	--I/O config	47	30	30	15	8
2	RAM	41696	8593	1218	609	304
1	--HARD-IP (E.G. PCIe)	1446	692	0	0	0
2	--Embedded RAM	40250	7901	1218	609	304
3	--MLAB (LUTRAM)	0	0	0	0	0
3	Registers	2563	794	794	396	198
1	--Hard-IP FF	474	177	177	88	44
2	--DSP/M20K FF	298	61	61	30	15
3	--Design FF	1791	556	556	278	139
4						
5	TOTAL	52745	13204	5829	2914	1457

8.2.4.2.1. Component FIT Rates

The Projected SEU FIT by Component report shows FIT for the following components:

- SRAM embedded memory in embedded processors hard IP and M20K or M10K blocks
- CRAM used for LUT masks and routing configuration bits
- LABs in MLAB mode
- I/O configuration registers, which the FPGA implements differently than CRAM and design flipflops
- Standard flipflops the design uses in the address and data registers of M20K blocks, in DSP blocks, and in hard IP
- User flipflops the design implements in logic cells (ALMs or LEs)

8.2.4.2.2. Raw FIT

The Intel Quartus Prime Projected SEU FIT by Component Usage report provides raw FIT data. Raw FIT is the FIT rate of the FPGA if the design uses every component. Raw FIT data is not design specific.

Note: The Intel Reliability Report, available on the Intel FPGA web site, also provides reliability data and testing procedures for Intel FPGA devices.

The Intel Quartus Prime software computes the FIT for each component using (component Mb × intrinsic FIT/Mb × Neutron Flux Multiplier) for the device family and process node. (For flip flops, "Mb" represents a million flip flops.)

To give the worst-case raw FIT, the report assumes the maximum amount of CRAM that implements MLABs in the device. Thus, the CRAM raw FIT is the sum of CRAM and MLAB entries.

Note: The Intel Quartus Prime software counts device bits for target devices using different parameter information than the Reliability Report. Therefore, expect a ±5% variation in the Projected SEU FIT by Component Usage report **Raw** column compared to the Reliability Report data.

8.2.4.2.3. Utilized FIT

The **Utilized** column shows FIT calculations considering only resources that the design actually uses. Since SEU events in unused resources do not affect the FPGA, you can safely ignore these bits for resiliency statistics.

Additionally, the **Utilized** column discounts unused memory bits. For example, implementing a 16 × 16 memory in an M20K block uses only 256 bits of the 20 Kb.

Note: The Error Detection flag and the Projected SEU FIT by Component report do not distinguish between critical bit upsets, such as fundamental control logic, or non critical bit upsets, such as initialization logic that executes only once in the design. Apply hierarchy tags at the system level to filter out less important logic errors.

The Projected SEU FIT by Component report's **Utilized** CRAM FIT represents provable deflation of the FIT rate to account for CRAM upsets that do not matter to the design. Thus, the SEU incidence is always higher than the utilized FIT rate.

Comparing .smh Critical Bits Report to Utilized Bit Count

The number of design critical bits that the Compiler reports during `.smh` generation correlates to the utilized bits in the report, but it is not the same value. The difference occurs because the `.smh` file includes all bits in a resource, even when the resource usage is partial.

Considerations for Small Designs

The raw FIT for the entire device is always correct. In contrast, the utilized FIT is very conservative, and only becomes accurate for designs that reasonably fill up the chosen device. FPGAs contain overhead, such as the configuration state machine, the clock network control logic, and the I/O calibration block. These infrastructure blocks contain flip flops, memories, and sometimes I/O configuration blocks.

The Projected SEU FIT by Component report includes the constant overhead for GPIO and HSSI calibration circuitry for first I/O block or transceiver the design uses. Because of this overhead, the FIT of a 1-transceiver design is much higher than 1/10 the FIT of a 10-transceiver design. However, a trivial design such as "a single AND gate plus flipflop" could use so few bits that its CRAM FIT rate is 0.01, which the report rounds to zero.

8.2.4.2.4. Mitigated FIT

You can lower FIT by reducing the observed FIT rate, such as by enabling ECC. You can also use the optional M20K ECC to mitigate FIT, as well as the (not optional) hard processor ECC and other hard IP such as memory controllers, PCIe, and I/O calibration blocks.

The Projected SEU FIT by Component Usage report's **w/ECC** column represents the FPGA's lowest guaranteed, provable FIT rate that the Intel Quartus Prime software can calculate. ECC does not affect CRAM and flipflop rates; therefore, the data in the **w/ECC** column for these components is the same as the in **Utilized** column.

The ECC code strength varies with the device family. In Intel Arria 10 devices, the M20K block can correct up to two errors, and the FIT rate beyond two (not corrected) is small enough to be negligible in the total.

An MLAB is simply a LAB configured with writable CRAM. However, when the Intel Quartus Prime software configures the RAM as write enabled (MLAB), the MLAB has a slightly different FIT/Mb. The Projected SEU FIT by Component Usage report displays a FIT rate in the MLAB row when the design uses MLABs, otherwise the report accounts for the block's FIT in the CRAM row. During compilation, if the Intel Quartus Prime software changes a LAB to an MLAB, the FIT accounting moves from the LAB row to the MLAB row.

The **w/ECC** column does not account for other forms of FIT protection in the design, such as designer-inserted parity, soft ECC blocks, bounds checking, system monitors, triple-module redundancy, or the impact of higher-level protocols on general fault tolerance. Additionally, it does not account for single event effects that occur in the logic but the design never reads or notices. For example, if you implement a non-ECC FIFO function 512 bits deep and an SEU event occurs outside of the front and back pointers, the application does not observe the SEU event. However, the report accounts for the full 512 bit deep memory and includes it in the **w/ECC** FIT rate. Designers often combine these factors into general deflation factors (called architectural vulnerability factors or AVF) based on knowledge of their design. Designers use AVF factors as low (aggressive) as 5% and as high (conservative) as 50% based on experience, fault-injection or neutron beam testing, or high-level system monitors.

8.2.4.2.5. Architectural Vulnerability Factor

The Single Event Functional Interrupt (SEFI) ratio measures bit errors due to SEU strikes versus functional interrupts. Minimizing this ratio improves SEU mitigation. 10% SEFI factors are A typical specification to deflate the raw FIT to that observed in practice. For reference, the last two columns in the Projected SEU FIT by Component Usage report show AVF deflations for a conservative SEFI of 50% and a moderate SEFI of 25%.

SEFI represents a combination of factors. A utilization + ECC factor of 40% and AVF of 25% thus represents a global SEFI factor of 10%, because 0.4 × 0.25 = 0.1. An end-to-end SEFI factor of 10% is typical for a full design.

8.2.4.3. Enabling the Projected SEU FIT by Component Usage Report

The Intel Quartus Prime Fitter generates the Projected SEU FIT by Component Usage report. The Intel Quartus Prime software only generates reports for designs that successfully pass place and route.

To enable the report:

1. Obtain and install the SEU license.
2. Add the following assignments to your project's `.qsf` file:

```
set_global_assignment –name ENABLE_ADV_SEU_DETECTION ON
set_global_assignment –name SEU_FIT_REPORT ON
```

8.3. CRAM Error Detection Settings Reference

To define these settings in the Intel Quartus Prime software, point to **Assignments ➤ Device ➤ Device and Pin Options ➤ Error Detection CRC**.

Figure 175. Device and Pin Options Error Detection CRC Tab

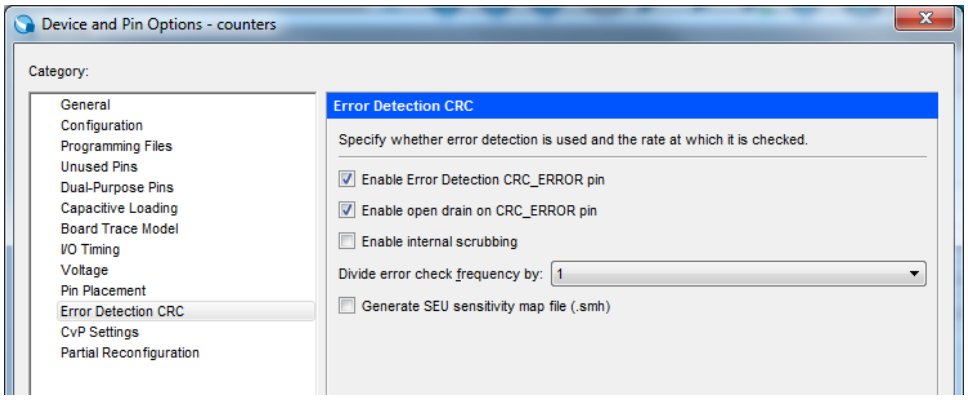


Table 111. CRC Errors Settings

Setting	Description
Enable Error Detection CRC_ERROR pin	Enables CRAM frame scanning
Enable open drain on CRC_ERROR pin	Enables the CRC_ERROR pin as an open-drain output
Divide error check frequency by	To guarantee the availability of a clock, the EDCRC function operates on an independent clock generated internally on the FPGA itself. To enable EDCRC operation on a divided version of the clock, select a value from the list.

8.4. Specifications

This section lists the error detection frequencies and CRC calculation time for error detection in user mode.

8.4.1. Error Detection Frequency

When you are unable to unload the EMR within the EMR update interval specification, you can reduce the 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.

Note: There is no significant power benefited from reducing the error detection frequency.

The speed of the error detection process for each data frame is determined by the following equation:

Figure 176. Error Detection Frequency Equation

$$\text{Error Detection Frequency} = \frac{\text{Internal Oscillator Frequency}}{N}$$

Table 112. Error Detection Frequency Range for Intel Arria 10 Devices

The following table lists the F_{MIN} and F_{MAX} for each speed grade.

Note: Frequencies shown are when N = 1. For N = 2 or 4, divide the frequency shown accordingly.

Speed Grade	Error Detection Frequency	
	f_{MIN}	f_{MAX}
1	49	77
2	45	77
3	42	77

8.4.2. Error Detection Time

The time taken to detect the SEU error relative to the actual SEU event. This is determined by the device in use and the frequency of the error detection clock.

Table 113. CRC Calculation Time in Intel Arria 10 Devices

$$\text{Error detection time}_{Maximum} = \text{Error detection time} \times \left(\frac{\text{Error Detection Frequency } f_{MAX}}{\text{Error Detection Frequency } f_{MIN}} \right) \times N$$

$$\text{Error detection time}_{Minimum} = \text{Error detection time} \times N$$

- Speed grade 1: N=1, 2 or 4.
- Speed grade 2 and 3: N=2 or 4 only.

Variant	Density	Error Detection Time (ms)
GX/SX	160 / 220	14.29
	270 / 320	14.29
	480	21.13
	570/ 660	27.84
GX/ GT	900 / 1150	27.84

8.4.3. EMR Update Interval

You must unload the EMR data within the minimum EMR Update Interval to avoid the current EMR data from being overwritten by the information of the next error. However, EMR Unloader IP Core can handle this by ensuring no EMR data loss during unloading process. The IP core detects the loss of EMR information by flagging the emr_error signal.

The interval between each update of the error message register depends on the device and the frequency of the error detection clock.

Table 114. Estimated EMR Update Interval in Intel Arria 10 Devices

$$\text{EMR update interval}_{\text{Maximum}} = \text{EMR update interval} \times \left(\frac{\text{Error Detection Frequency } f_{MAX}}{\text{Error Detection Frequency } f_{MIN}} \right) \times N$$

$$\text{EMR update interval}_{\text{Minimum}} = \text{EMR update interval} \times N$$

- Speed grade 1: N=1, 2 or 4.
- Speed grade 2 and 3: N=2 or 4 only.

Variant	Density	EMR Update Interval (ms)
GX/SX	160 / 220	0.28
	270 / 320	0.28
	480	0.41
	570/ 660	0.54
GX/ GT	900 / 1150	0.55

8.4.4. Error Correction Time

Intel Arria 10 offers fast error correction capability, the correction time for each device variant are shown in the following table.

Table 115. Error Correction Time

$$\text{Correction time}_{\text{Maximum}} = \text{Correction time} \times \left(\frac{\text{Error Detection Frequency } f_{MAX}}{\text{Error Detection Frequency } f_{MIN}} \right) \times N$$

$$\text{Correction time}_{\text{Minimum}} = \text{Correction time} \times N$$

Note:
- Speed grade 1: N=1, 2 or 4.
- Speed grade 2 and 3: N=2 or 4 only.

Variant	Density	Correction Time (µs)
GX/SX	160/220	19.73
	270/320	27.62
	480	27.21
	570/660	27.21
GX/GT	900/1150	39.68

8.5. SEU Mitigation for Intel Arria 10 Devices Revision History

Document Version	Changes
2019.09.06	Updated the topic about failure rates to correct the number of years of one billion hours.
2018.06.08	Added *Failure Rates*, *Configurating RAM to Enable ECC*, *Triple Module Redundancy*, *Software SEU FIT Reports* and sub-sections, and *CRAM Error Detection Settings Reference* sections.
2018.03.09	Added missing figure in *SEU Sensitivity Processing*.

Date	Version	Changes
March 2017	2017.03.15	Rebranded as Intel.
May 2016	2016.05.02	• Edited *Error Detection Cyclic Redundancy Check*. • Updated CRC check bit instances to column-based check-bits and frame-based check-bits. • Updated *Error Message Register Map* figure. • Added *Fast EDCRC Process Flow Chart*figure. • Added note stating that there is no significant power benefit from reducing error detection frequency.
December 2015	2015.12.14	• Updated chapter structure. • Added *Error Correction Time* specification. • Added brief description and external related information link for *Embedded Memory*, *Memory Blocks Error Correction Code Support*, *SEU Sensitivity*, *Hierarchy Tagging*, and *Evaluating your System Response to SEU*. • Updated divisor value in Error Detection Frequency. • Updated Error Detection Frequency Range table showing fMAX and fMIN. • Updated Estimated EMR Update Interval and CRC Calculation Time table. • Added equation for EMR update interval and CRC calculation time.

continued...

Date	Version	Changes
November 2015	2015.11.02	Changed instances of Quartus II to Quartus Prime.
June 2015	2015.06.15	Updated links to Altera EMR Unloader IP Core User Guide, Altera Fault Injection IP Core User Guide and Altera Advance SEU Detection IP Core User Guide.
May 2015	2015.05.04	• Added links to Altera EMR Unloader IP Core User Guide, Altera Fault Injection IP Core User Guide and Altera Advance SEU Detection IP Core User Guide. • Updated CRC_ERROR pin behavior to included column-based and frame-based CRC error detection and frame-based only CRC error detection. • Updated column-based type in 'Error Type in EMR' at Bit 0. • Editorial changes. • Updated the divisor value and range for error detection frequency. • Updated CRC calculation time by including speed grade and rearranged accordingly. • Updated EMR update interval. • Updated Error Message Register Map and registers in Error Detection in User Mode block diagram.
January 2015	2015.01.23	• Added EMR timing interval. • Added CRC calculation time. • Added timing diagram
August 2014	2014.08.18	• Updated the Error Detection Features section. • Updated the Configuration Error Detection section to revise the CRC value. • Updated the User Mode Error Detection section to add in the check bits calculation for error detection CRC. • Updated the CRC_ERROR Pin Behavior section. • Updated the Retrieving Error Information section. • Updated the CRC_ERROR Pin section to update the pin description. • Updated Table 8-4 to updated the description of the frame-based syndrome register, user update register, and user shift register. • Updated Table 8-5 to update the error types naming to frame-based and column-based types.
December 2013	2013.12.02	Initial release.

9. JTAG Boundary-Scan Testing in Intel Arria 10 Devices

This chapter describes the boundary-scan test (BST) features in Intel Arria 10 devices.

Related Information

Intel Arria 10 Device Handbook: Known Issues
Lists the planned updates to the *Arria 10 Device Handbook* chapters.

9.1. BST Operation Control

Intel Arria 10 GX, Intel Arria 10 GT, and Intel Arria 10 SX devices support IEEE Std. 1149.1 BST and IEEE Std. 1149.6 BST. You can perform BST on Intel Arria 10 devices before, after, and during configuration.

9.1.1. IDCODE

The IDCODE is unique for each Intel Arria 10 device. Use this code to identify the devices in a JTAG chain.

Table 116. IDCODE Information for Intel Arria 10 Devices

Variant	Product Line	IDCODE (32 Bits)			
		Version (4 Bits)	Part Number (16 Bits)	Manufacture Identity (11 Bits)	LSB (1 Bit)
Intel Arria 10 GX	GX 160	0000	0010 1110 1110 0010	000 0110 1110	1
	GX 220	0000	0010 1110 0010 0010	000 0110 1110	1
	GX 270	0000	0010 1110 1110 0011	000 0110 1110	1
	GX 320	0000	0010 1110 0010 0011	000 0110 1110	1
	GX 480	0000	0010 1110 0010 0100	000 0110 1110	1
	GX 570	0000	0010 1110 1110 0101	000 0110 1110	1
	GX 660	0000	0010 1110 0010 0101	000 0110 1110	1
	GX 900	0000	0010 1110 1110 0110	000 0110 1110	1
	GX 1150	0000	0010 1110 0110 0110	000 0110 1110	1
Intel Arria 10 GT	GT 900	0000	0010 1110 0010 0110	000 0110 1110	1
	GT 1150	0000	0010 1110 0000 0110	000 0110 1110	1
Intel Arria 10 SX	SX 160	0000	0010 1110 0110 0010	000 0110 1110	1
	SX 220	0000	0010 1110 0000 0010	000 0110 1110	1
	SX 270	0000	0010 1110 0110 0011	000 0110 1110	1
	SX 320	0000	0010 1110 0000 0011	000 0110 1110	1

continued...

Variant	Product Line	IDCODE (32 Bits)			
		Version (4 Bits)	Part Number (16 Bits)	Manufacture Identity (11 Bits)	LSB (1 Bit)
	SX 480	0000	0010 1110 0000 0100	000 0110 1110	1
	SX 570	0000	0010 1110 0110 0101	000 0110 1110	1
	SX 660	0000	0010 1110 0000 0101	000 0110 1110	1

9.1.2. Supported JTAG Instruction

Table 117. JTAG Instructions Supported by Intel Arria 10 Devices

JTAG Instruction	Instruction Code	Description
SAMPLE[42]/PRELOAD	00 0000 0101	• 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 pattern into the update registers before loading the EXTEST instruction.
EXTEST	00 0000 1111	• 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. • You will get a '0' reading in the bypass register out.
USERCODE	00 0000 0111	Selects the 32-bit USERCODE register and places it between the TDI and TDO pins to allow serial shifting of USERCODE out of TDO.

continued...

(42) The SAMPLE JTAG instruction is not supported for high-speed serial interface (HSSI) pins.

JTAG Instruction	Instruction Code	Description
IDCODE	00 0000 0110	• 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	• 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.
CLAMP	00 0000 1010	• 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.
EXTEST_PULSE	00 1000 1111	Enables board-level connectivity checking between the transmitters and receivers that are AC coupled by generating three output transitions:

continued...

JTAG Instruction	Instruction Code	Description
		• Driver drives data on the falling edge of TCK in the UPDATE_IR/DR state. • Driver drives inverted data on the falling edge of TCK after entering the RUN_TEST/IDLE state. • Driver drives data on the falling edge of TCK after leaving the RUN_TEST/IDLE state.
EXTEST_TRAIN	00 0100 1111	Behaves the same as the EXTEST_PULSE instruction except that the output continues to toggle on the TCK falling edge as long as the TAP controller is in the RUN_TEST/IDLE state.
SHIFT_EDERROR_REG	00 0001 0111	The JTAG instruction connects the EMR to the JTAG pin in the error detection block between the TDI and TDO pins.

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.

9.1.3. JTAG Secure Mode

In the JTAG secure mode, the JTAG pins support only the BYPASS, SAMPLE/PRELOAD, EXTEST, and IDCODE JTAG instructions.

Related Information

JTAG Secure Mode in AN 556
Provides more information about JTAG Secure Mode

9.1.4. JTAG Private Instruction

Caution: Never invoke the following instruction codes. These instructions can damage and render the device unusable:

- 1100010000
- 1100010011
- 0111100000
- 0101011110
- 0000101010
- 0011100000
- 0000101010
- 0101000001
- 1110000001
- 0001010101
- 1010100001

9.2. I/O Voltage for JTAG Operation

The Intel Arria 10 device operating in IEEE Std. 1149.1 and IEEE Std. 1149.6 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, TMS, and TRST pins have internal weak pull-up resistors. The 1.8-, 1.5-, or 1.2-V V_{CCPGM} supply powers the TDI, TDO, TMS, TCK, and TRST pins. All user I/O pins are tri-stated during JTAG configuration.

The JTAG pins support 1.8 V, 1.5V, and 1.2V TTL/CMOS I/O standard. For any voltages higher than 1.8 V, you have to use level shifter. The output voltage of the level shifter for the JTAG pins must be the same as set for the V_{CCPGM} supply.

Note: Do not drive a signal with a voltage higher than 1.8-, 1.5-, and 1.2-V V_{CCPGM} supply for the TDI, TMS, TCK, and TRST pins. The voltage supplies for TDI, TMS, TCK, and TRST input pins must be the same as set for the V_{CCPGM} supply.

Table 118. TDO Output Buffer

TDO Output Buffer	Voltage (V)		
V_{CCPGM}	1.8	1.5	1.2
V_{OH} (MIN)	1.7	1.4	1.1

9.3. 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_NCONFIG 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 Intel Arria 10 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 Intel Arria 10 devices, consider the connections for the dedicated configuration pins.

Note: For SoC devices, JTAG connections in the FPGA block and JTAG connections in the HPS block are chained to the Intel Arria 10 device. JTAG connections in the FPGA have higher priority over the JTAG connections in the HPS block.

Note: If you perform the HIGHZ JTAG instruction before or during configuration, you need to pull the nIO_PULLUP pin to high to disable the internal weak pull-up resistors in the I/O elements. If you perform this JTAG instruction during user mode, you can pull high or pull low the nIO_PULLUP pin.

Note: If you perform BST during user mode, you are not able to capture the correct values for the PR_ENABLE, CRC_ERROR, and CVP_CONFDONE pins when these pins are not used as user I/O pins.

Note: You can perform JTAG BST only when both nCONFIG and nSTATUS goes high after power-up.

Related Information

- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
  Provides more information about pin connections.
- JTAG Configuration
  Provides more information about JTAG configuration timing.
- JTAG Configuration
  Provides more information about JTAG configuration timing.
- JTAG Configuration on page 244

9.4. Enabling and Disabling IEEE Std. 1149.1 BST Circuitry

The IEEE Std. 1149.1 BST circuitry is enabled after the Intel Arria 10 device powers up. However for Intel Arria 10 SoC FPGAs, you must power up both HPS and FPGA to perform BST.

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.

Table 120. Pin Connections to Permanently Disable the IEEE Std. 1149.1 Circuitry for Intel Arria 10 Devices

JTAG Pins[44]	Connection for Disabling
TMS	V_{CCPGM}
TCK	GND
TDI	V_{CCPGM}
TDO	Leave open
TRST	GND

(44) The JTAG pins are dedicated. Software option is not available to disable JTAG in Intel Arria 10 devices.

9.5. 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

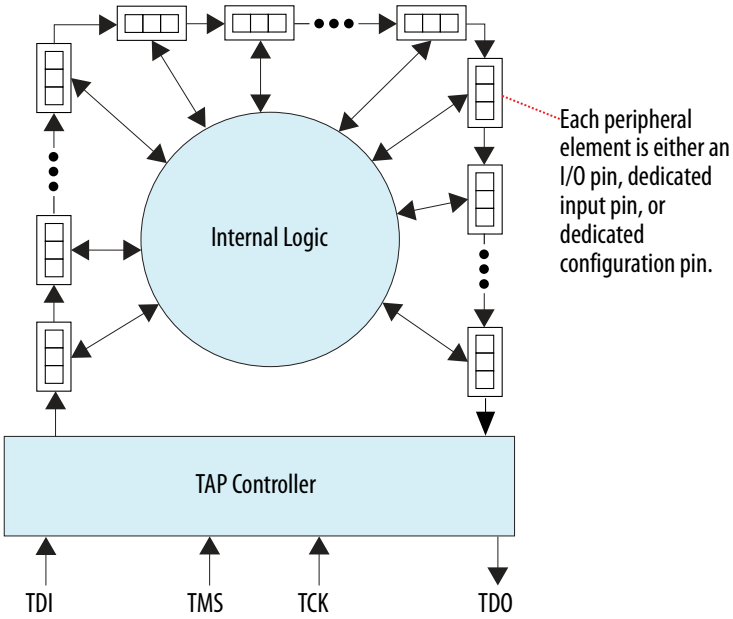
IEEE 1149.1 BSDL Files
Provides more information about the BSC group definitions.

9.6. 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 Intel Arria 10 I/O pins. You can use the boundary-scan register to test external pin connections or to capture internal data.

Figure 177. Boundary-Scan Register

This figure shows how test data is serially shifted around the periphery of the IEEE Std. 1149.1 device.



9.6.1. Boundary-Scan Cells of an Intel Arria 10 Device I/O Pin

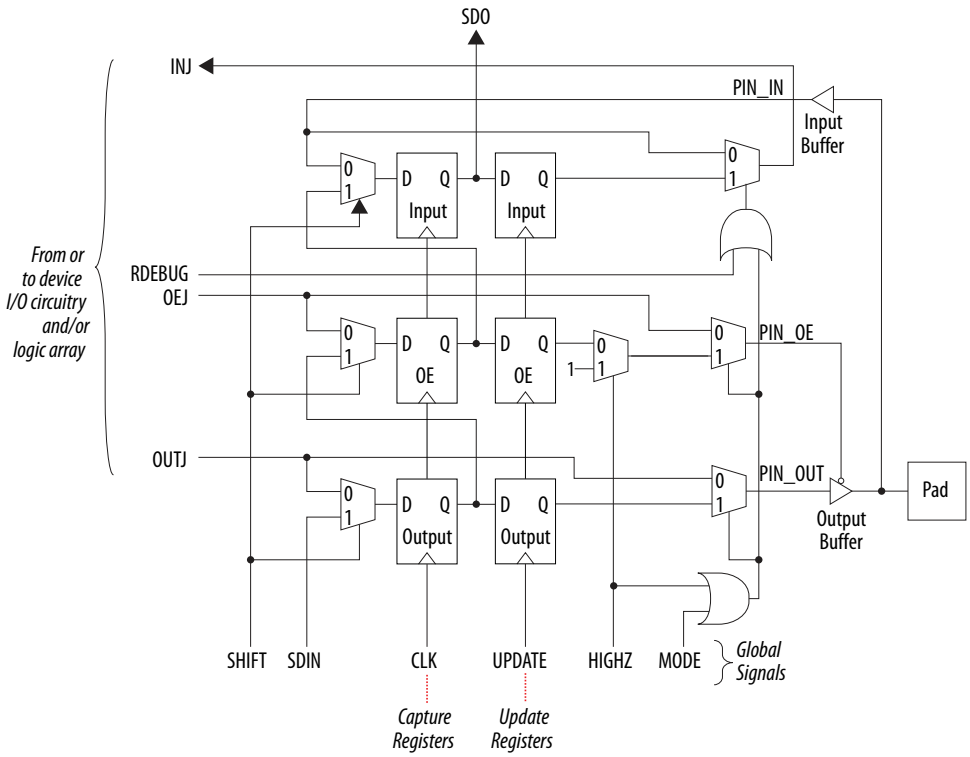
The Intel Arria 10 device 3-bit BSC consists of the following registers:

- 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 178. User I/O BSC with IEEE Std. 1149.1 BST Circuitry for Intel Arria 10 Devices



Note: TDI, TDO, TMS, TCK, TRST, VCC, GND, VREF, VSIGP, VSIGN, TEMPDIODE, and RREF pins do not have BSCs.

Table 121. Boundary-Scan Cell Descriptions for Intel Arria 10 Devices

This table lists the capture and update register capabilities of all BSCs within Intel Arria 10 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	No Connect (N.C.)	N.C.	PIN_IN	N.C.	N.C.	N.C.	PIN_IN drives to the clock network or logic array
Dedicated input	N.C.	N.C.	PIN_IN	N.C.	N.C.	N.C.	PIN_IN drives to the control logic

continued...

Pin Type	Captures			Drives			Comments
	Output Capture Register	OE Capture Register	Input Capture Register	Output Update Register	OE Update Register	Input Update Register	
Dedicated bidirectional (open drain) (45)	0	OEJ	PIN_IN	N.C.	N.C.	N.C.	PIN_IN drives to the configuration control
Dedicated bidirectional (46)	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 (47)	OUTJ	0	0	N.C.	N.C.	N.C.	OUTJ drives to the output buffer

9.6.2. 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 Intel Arria 10 devices are different from the BSCs for the I/O pins.

Note: You have to use the EXTEST_PULSE JTAG instruction for AC-coupling on HSSI transceiver. Do not use the EXTEST JTAG instruction for AC-coupling on HSSI transceiver. You can perform AC JTAG on the Intel Arria 10 device before, after, and during configuration.

(45) This includes the CONF_DONE and nSTATUS pins.

(46) This includes the DCLK pin.

(47) This includes the nCEO pin.

Figure 179. HSSI Transmitter BSC with IEEE Std. 1149.6 BST Circuitry for Intel Arria 10 Devices

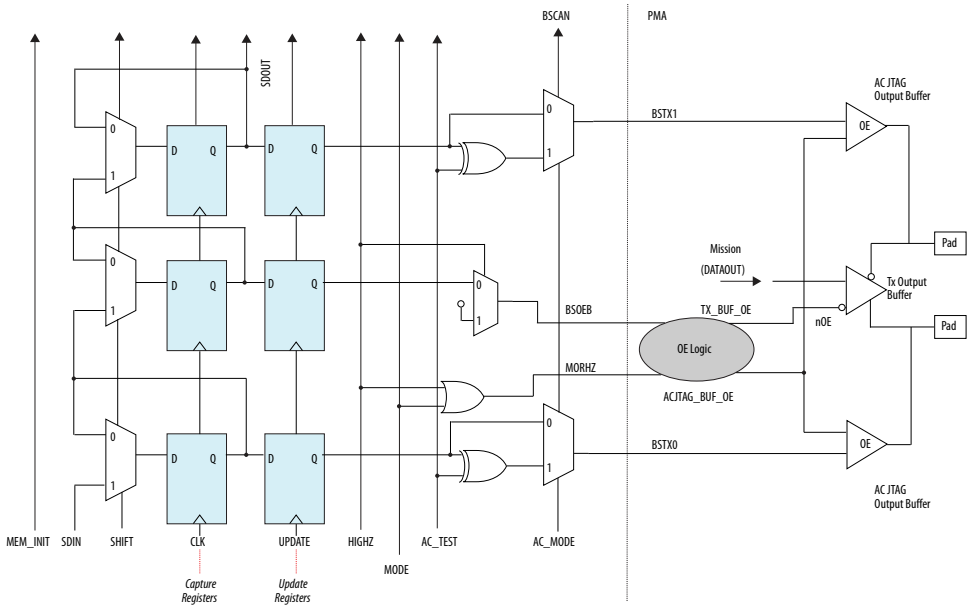
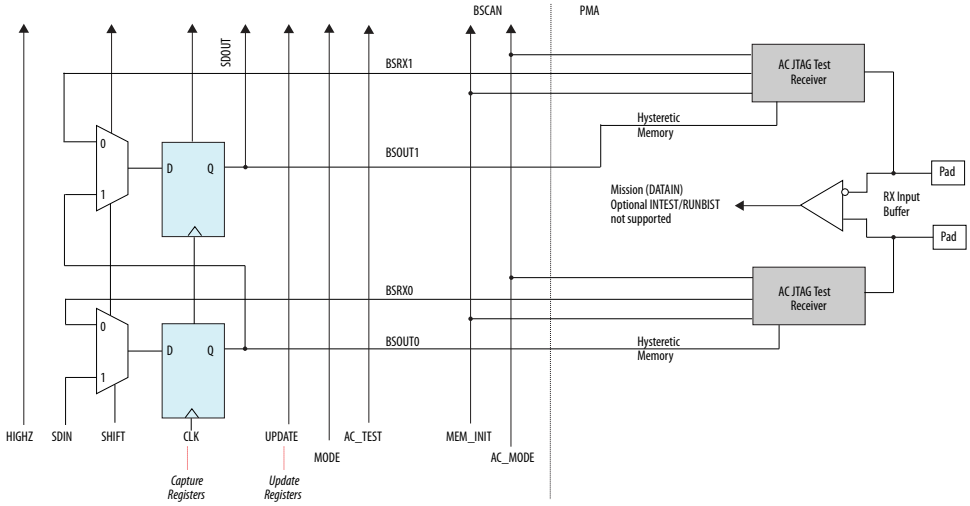


Figure 180. HSSI Receiver/Input Clock Buffer with IEEE Std. 1149.6 BST Circuitry for Intel Arria 10 Devices



9.7. JTAG Boundary-Scan Testing in Intel Arria 10 Devices Revision History

Date	Version	Changes
March 2017	2017.03.15	Rebranded as Intel.
May 2016	2016.05.02	• Updated IDCODE. • Added note about SAMPLE instruction is not available for HSSI pins.

continued...

Date	Version	Changes
December	2015.12.14	• Updated *User I/O BSC with IEEE Std. 1149.1 BST Circuitry for Arria 10 Devices* figure. • Added SHIFT_EDERROR_REG in Supported JTAG instruction table.
November 2015	2015.11.02	Added note to state that JTAG BST can be performed after nSTATUS and nCONFIG are high.
August 2014	2014.08.18	• Updated the JTAG Private Instruction section to add a new instruction code. • Updated the I/O Voltage for JTAG Operation section to update the TDO output buffer details. • Updated the Performing BST section to add a note on performing BST in user mode. • Updated the Boundary-Scan Cells of an Arria 10 Device I/O Pin section.
December 2013	2013.12.02	Initial release.

10. Power Management in Intel Arria 10 Devices

This chapter describes the power consumption, power reduction techniques, power sense line feature, on-chip voltage sensor, internal and external temperature sensing diode (TSD), power-on reset (POR) requirements, power-up and power-down sequencing requirements, and power supply design.

Related Information

- Intel Arria 10 Device Handbook: Known Issues
  Lists the planned updates to the *Arria 10 Device Handbook* chapters.
- Power Analysis chapter in volume 3 of the Quartus Prime Handbook
  Provides more information about the Intel Quartus Prime Power Analyzer tool.
- Recommended Operating Conditions
  Provides more information about the recommended operating conditions of each power supply.
- Intel Arria 10 GX, GT, and SX 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.
- Early Power Estimators (EPE) and Power Analyzer
  Provides more information about the power supplies and the current requirements for each power rail.
- Intel Power Management PowerSoC Solutions
  Provides more information about Intel's Power Management IC and PowerSoC solutions designed for powering FPGAs.

10.1. Power Consumption

The total power consumption of an Intel Arria 10 device consists of the following components:

- Static power—the power that the configured device consumes when powered up but no user clocks are operating.
- Dynamic power— the additional power consumption of the device due to signal activity or toggling.

10.1.1. Dynamic Power Equation

Figure 181. 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. The frequency refers to the clock frequency and data toggles once every clock cycle.

$$P = \frac{1}{2} CV^2 \times frequency$$

The equation shows that power is design-dependent and is determined by the operating frequency of your design. Intel Arria 10 devices minimize static and dynamic power using advanced process optimizations. These optimizations allow Intel Arria 10 designs to meet specific performance requirements with the lowest possible power.

10.2. Power Reduction Techniques

Intel Arria 10 devices leverage advanced 20nm process technology, an enhanced core architecture, and process optimization to reduce total power consumption. The optional power reduction techniques listed below are offered and supported in the Intel Quartus Prime Power Analyzer, which can be used to estimate the power reduction impact by enabling each of them in an Intel Arria 10 design.

- SmartVID
- Programmable Power Technology
- Low Static Power Device Grades

10.2.1. SmartVID

The SmartVID feature allows a power regulator to provide the Intel Arria 10 device with a lower V_{CC} and V_{CCP} voltage level while maintaining the performance of the specific device speed grade. Operating the Intel Arria 10 device at lower than nominal V_{CC} and V_{CCP} voltage levels reduces total power consumption. The minimum voltage level required by Intel Arria 10 devices is programmed into a fuse block during device manufacturing. Intel provides an IP core to read these values and communicate them to an external power regulator or system power controller. This feature is supported in –2 and –3 speed grades devices with –V power option only.

When the SmartVID feature is used, Intel Arria 10 devices need to be powered up at the nominal voltage level. During configuration and partial reconfiguration modes, Intel Arria 10 devices continue to operate at the nominal voltage level. Upon entering user mode, the Intel Arria 10 device can operate at a lower voltage as indicated in the fuse block. The error detection cyclic redundancy check (EDCRC) feature is supported for –2 speed grade devices even when the SmartVID feature is used. However, for other speed grades, Intel Arria 10 devices need to operate at nominal voltage when performing the EDCRC feature. The scrubbing and partial reconfiguration features are supported only when the device is operated at nominal voltage.

Related Information

- Power Reduction Features in Intel Arria 10 Devices
- SmartVID Controller IP Core User Guide

10.2.2. Programmable Power Technology

Intel Arria 10 devices offer the ability to configure portions of the core, called tiles, for high-speed or low-power mode of operation. This configuration is performed by the Intel Quartus Prime software automatically and without the need for 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. 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.

Intel Arria 10 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

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. Unused M20K blocks are set to sleep mode by disabling V_{CCERAM} to reduce static power. Clock networks do not support programmable power technology.

With programmable power technology, faster speed grade FPGAs may require less static power compared with FPGA devices without programmable power technology. For device with programmable power technology, critical path is a small portion of the design. Therefore, there are fewer high-speed MLAB and LAB pairs in high-speed mode. For device without programmable power technology, the whole FPGA has to be over designed to meet the timing of the critical path.

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.

Table 122. Programmable Power Capabilities for Intel Arria 10 Devices

This table lists the available Intel Arria 10 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

continued...

Feature	Programmable Power Technology
Memory Blocks	Fixed setting (48)
DSP Blocks	Fixed setting (48)
Clock Networks	No

Related Information

Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
Provides more information about the required voltage levels for each power rail.

10.2.3. Low Static Power Device Grades

Intel offers some Intel Arria 10 device grades that consume lower static power than standard power devices while maintaining performance. The low static power device grades feature is only offered for selected devices with the power option 'L'.

Related Information

Intel Arria 10 Device Variants and Packages
Provides more information about the ordering code.

10.2.4. SmartVID Feature Implementation

The implementation of the SmartVID feature consists of 7-bit voltage identification (VID) that is programmed into a fuse block during device manufacturing.

The 7-bit VID represents a voltage level in the range of 0.85 V to 0.9 V. Each device has its own specific 7-bit VID. You can read the 7-bit VID using the SmartVID Controller IP core. You have the option to enable or disable the VID bit reading.

The 7-bit VID is read from the fuse block and sent to the external regulator or system power controller through the Intel FPGA-supported interface. Upon receiving the 7-bit VID value, an adjustable regulator tunes down the V_{CC} and V_{CCP} voltage levels to a lower voltage as specified by the 7-bit VID. Multiple interface methods are supported for the Intel Arria 10 device to communicate the VID value to an external regulator or system power controller. The first method to be available is the 7-bit parallel interface.

Intel offers external regulators and system power controllers that support the SmartVID feature and are compatible with the multiple interfaces methods utilized by the Intel Arria 10 device.

The 7-Bit Parallel Interface Solution

The 7-bit parallel solution is a parallel VID bit interface that is supported by Intel. This interface requires seven I/O pins for seven parallel VID bits and one pin for the VID_EN to communicate with the external regulator.

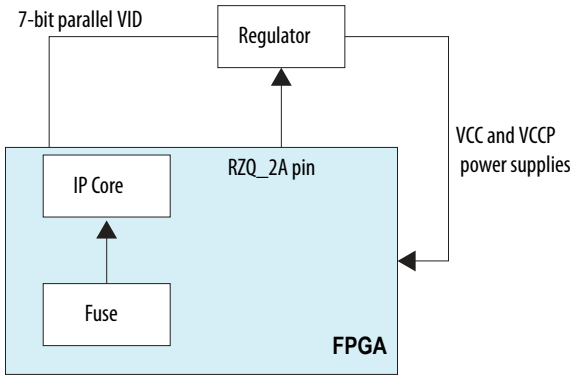
Intel recommends you to use the RZQ_2A pin for the VID_EN function. If bank 2A is used for DDR interface, and the RZQ_2A pin must be used for calibration purpose, you can use other available general-purpose I/O pins for the VID_EN pin function. Before

(48) 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.

the VID_EN pin is asserted, you need to ensure the I/O bank that hosts the VID_EN pin and VID pins are powered up. Connect the VID_EN pin to a 1-kΩ pull-down resistor.

The VID pins need to be tri-stated during power-up and before the VID_EN pin is asserted. Intel recommends using a level shifter to isolate the VID signals and voltage regulator controller. This is because some of the VID bit settings may exceed the maximum V_{CC} and V_{CCP} values.

Figure 182. External Interface Connection for the 7-Bit Parallel Solution



The following table lists the regulator requirement to meet the Intel SmartVID solution.

Table 123. Regulator Requirement for Intel SmartVID Solution

Specification	Value
Voltage range	0.82 V – 0.93 V (49)
Voltage step	10 mV step
V_{CC} power supply	10 W – 100 W
VID input	7-bit VID
Nominal voltage	0.85 V – 0.9 V (50)
Ramp time	0.5 mV/us
VID_EN pin	1 pin

Related Information

SmartVID Controller IP Core User Guide

10.3. Power Sense Line

Intel Arria 10 devices support the power sense line feature. VCCLSENSE and GNDSENSE pins are differential remote sense pins to monitor the V_{CC} power supply.

(49) This voltage range is the output of the regulator to the Intel Arria 10 device, inclusive of tolerance.

(50) The nominal device power-up voltage is 0.9 V.

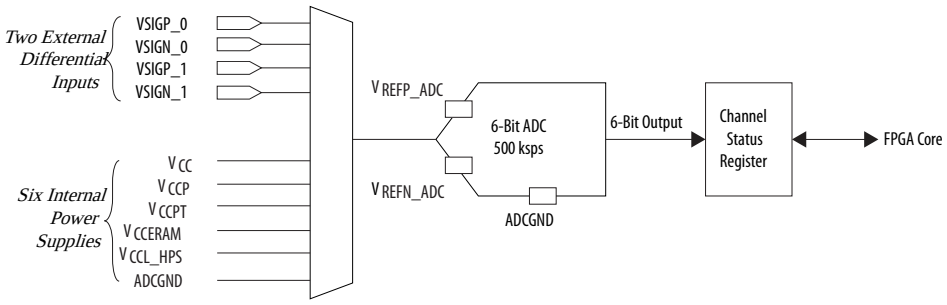
Intel recommends connecting the VCCLSENSE and GNDSENSE pins for regulators that support the power sense line feature. The required conditions to connect VCCLSENSE and GNDSENSE lines to the regulator's remote sense inputs are shown below:

- The V_{CC} or V_{CCP} current is > 30A.
- The SmartVID feature is used.

10.4. Voltage Sensor

Intel Arria 10 supports an on-chip voltage sensor. The voltage sensor provides a 6-bit digital representation of the analog signal being observed. The voltage sensor monitors two external differential inputs and six internal power supplies as shown in the following figure. To get the ADC input, the V_{CCPT} voltage value is divided by two. To get the actual V_{CCPT} voltage value, multiply the ADC output by two.

Figure 183. Voltage Sensor



The conversion speed of the ADC is 500 ksps cumulative. When multiple channels are used, the speed per channel is reduced accordingly.

Note: VREFP_ADC pins consume very little current, most of the current drawn is attributed to the leakage current, which is less than 10 µA. For VREFN_ADC pins, the current is less than 0.1 mA.

For better ADC performance, tie VREFP_ADC and VREFN_ADC pins to an external 1.25 V accurate reference source (±0.2%). An on-chip reference source (±10%) is activated by connecting the VREFP_ADC pin to GND. Treat VREFN_ADC together with VREFP_ADC as an analog 1.25 V differential signal.

Connect both VREFP_ADC and VREFN_ADC pins to GND if no external reference is supplied.

Related Information

Altera Voltage Sensor IP Core User Guide

10.4.1. Input Signal Range for External Analog Signal

You can configure the ADC to measure unipolar analog external input signal.

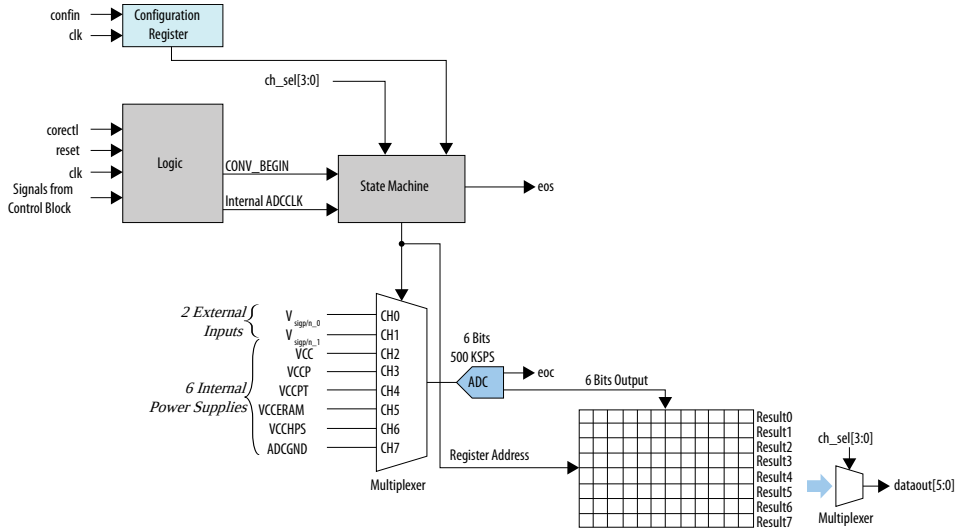
10.4.1.1. Unipolar Input Mode

In unipolar input mode, the voltage on the VSIGP pin which is measured with respect to the VSIGN pin must always be positive. The VSIGP input must always be driven by an external analog signal. The VSIGN pin is connected to a local ground or common mode signal.

10.4.2. Using Voltage Sensor in Intel Arria 10 Devices

You can use the voltage sensor feature to monitor critical on-chip power supplies and external analog voltages. The voltage sensor block for Intel Arria 10 devices supports access from the FPGA core. The following sections describe the flow in using the voltage sensor for Intel Arria 10 devices.

Figure 184. Voltage Sensor Components



10.4.2.1. Accessing the Voltage Sensor Using FPGA Core Access

In user mode, you can implement a soft IP to access the voltage sensor block. To access the voltage sensor block from the core fabric, include the following WYSIWYG atom in your Intel Quartus Prime project:

Example 1. WYSIWYG Atom to Access the Voltage Sensor Block

```
twentynm_vsblock<name>
(
    .clk (<input>, clock signal from core),
    .reset(<input>, reset signal from core),
    .corectl(<input>, core enable signal from core),
    .coreconfig(<input>, config signal from core),
    .confin(<input>, config data signal from core),
    .chsel(<input>, 4 bits channel selection signal from core),
    .eoc(<output>, end of conversion signal from vsblock),
```

```
    .eos(<output>, end of sequence signal from vsblock),
    .dataout(<output>, 12 bits data out of vsblock)
);
```

Table 124. Description for the Voltage Sensor Block WYSIWYG

Port Name	Type	Description
clk	Input	Clock signal from the core. The voltage sensor supports up to an 11-MHz clock.
reset	Input	Active high reset signal. An asynchronous high-to-low transition on the reset signal starts voltage sensor conversion. All registers are cleared and the internal voltage sensor clock is gated off when the reset signal is high.
corectl	Input	Active high signal. "1" indicates the voltage sensor is enabled for core access. "0" indicates the voltage sensor is disabled for core access.
coreconfig	Input	Serial configuration signal. Active high.
confin	Input	Serial input data from the core to configure the configuration register. The configuration register for the core access mode is 8 bits wide. The LSB is the first bit shifted in.
chsel[3:0]	Input	4-bit channel address. Specifies the channel to be converted.
eoc	Output	Indicates the end of the conversion. This signal is asserted after the conversion of each channel data packet.
eos	Output	Indicates the end of sequence. This signal is asserted for one cycle after the completion of the conversion of the selected sequence.
dataout[11:0]	Output	• dataout[11:6]—6-bit output data. • dataout[5:0]—Reserved.

10.4.2.1.1. Configuration Registers for the Core Access Mode

The core access configuration register is an 8-bit register.

Figure 185. Core Access Configuration Register

D7	D6	D5	D4	D3	D2	D1	D0
NA	CAL	NA	NA	BU1	BU0	MD1	MD0

Table 125. Description for the Core Access Configuration Register

Bit Number	Bit Name	Description
D0	MD0	Mode select for channel sequencer: • MD[1:0]=2'b00—channel sequencer cycles from channel 2 to channel 7 • MD[1:0]=2'b01—channel sequencer cycles from channel 0 to channel 7 • MD[1:0]=2'b10—channel sequencer cycles from channel 0 to channel 1 • MD[1:0]=2'b11—controlled by IP core. Specify the channel to be converted on chsel[3:0].
D1	MD1	
D2	BU0	Channel 0—Register bit that indicates channel 0. Set to "0".
D3	BU1	Channel 1—Register bit that indicates channel 1. Set to "0".
D4	NA	Reserved. Set to "0".

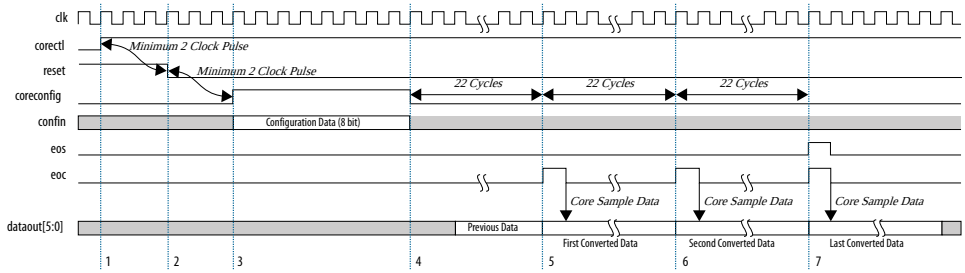
continued...

Bit Number	Bit Name	Description
D5	NA	Reserved. Set to "0".
D6	CAL	Calibration enable bit. "0" indicates calibration is off. "1" indicates calibration is on. The calibration result does not include the final 12-bit converted data when calibration is off.
D7	NA	Reserved. Set to "0".

10.4.2.1.2. Accessing the Voltage Sensor in the Core Access Mode when MD[1:0] is not Equal to 2'b11

The following timing diagram shows the IP core timing to access the voltage sensor when MD[1:0] is not equal to 2'b11.

Figure 186. Timing Diagram when MD[1:0] is not Equal to 2'b11



1. A low-to-high transition on the `corectl` signal enables the core access mode.

   Wait for a minimum of two clock cycles before proceeding to step 2.
2. De-asserting the `reset` signal releases the voltage sensor from the reset state.

   Wait for a minimum two clock cycles before proceeding to step 3.
3. Configure the voltage sensor by writing the configuration registers and asserting the `coreconfig` signal for eight clock cycles. The configuration register access mode is 8 bits and configuration data is shifted in serially.
4. The `coreconfig` signal going low indicates the start of the conversion based on the configuration defined in the configuration register.
5. Poll the `eoc` and `eos` status signals to check if conversion for the first channel defined by MD[1:0] is complete. Latch the output data on the `dataout[5:0]` signal at the falling edge of the `eoc` signal.
6. Poll the `eoc` and `eos` status signals to check if conversion for the subsequent channels defined by MD[1:0] are complete. Latch the output data on the `dataout[5:0]` signal at the falling edge of the `eoc` signal.
7. Repeat step 6 until the `eos` signal is asserted, indicating the completion of the conversion of one cycle on the channels specified by MD[1:0].
   a. Both the `eoc` and `eos` signals are asserted on the same clock cycle when the voltage sensor completes the conversion for the last channel.

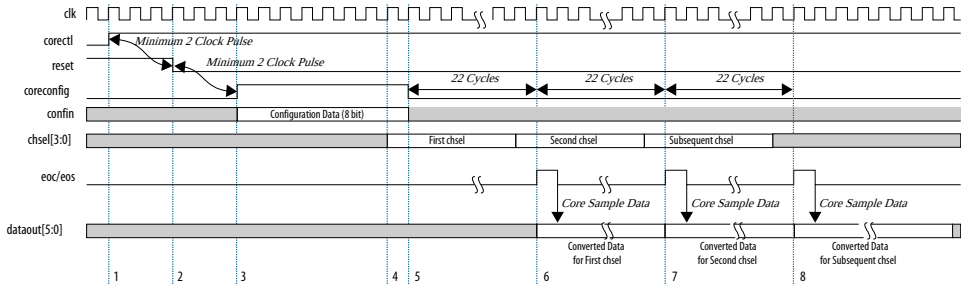
b. You can only interrupt the operation of the voltage sensor by writing into the configuration register after one cycle of the eos signal completes.

8. When the sequence completes, and if the corectl and reset signals remain unchanged, the conversion repeats the same sequence again until corectl is 0 and reset is 1. If you want to measure other sequences, repeat step 2 to step 7.

10.4.2.1.3. Accessing the Voltage Sensor in the Core Access Mode when MD[1:0] is Equal to 2'b11

The following timing diagram shows the IP core timing to access the voltage sensor when MD[1:0] is equal to 2'b11.

Figure 187. Timing Diagram when MD[1:0] is Equal to 2'b11



1. A low-to-high transition on the corectl signal enables the core access mode.

   Wait for a minimum of two clock cycles before proceeding to step 2.

2. De-asserting the reset signal releases the voltage sensor from the reset state.

   Wait for a minimum two clock cycles before proceeding to step 3.

3. Configure the voltage sensor by writing the configuration registers and asserting the coreconfig signal for eight clock cycles. The configuration register access mode is 8 bits and configuration data is shifted in serially.

4. Specify the channel for conversion on the chsel[3:0] signal. Data on the chsel[3:0] signal must be stable before the coreconfig signal is de-asserted.

5. The coreconfig signal going low indicates the start of the conversion based on the configuration defined in the configuration register and the chsel[3:0] signal.

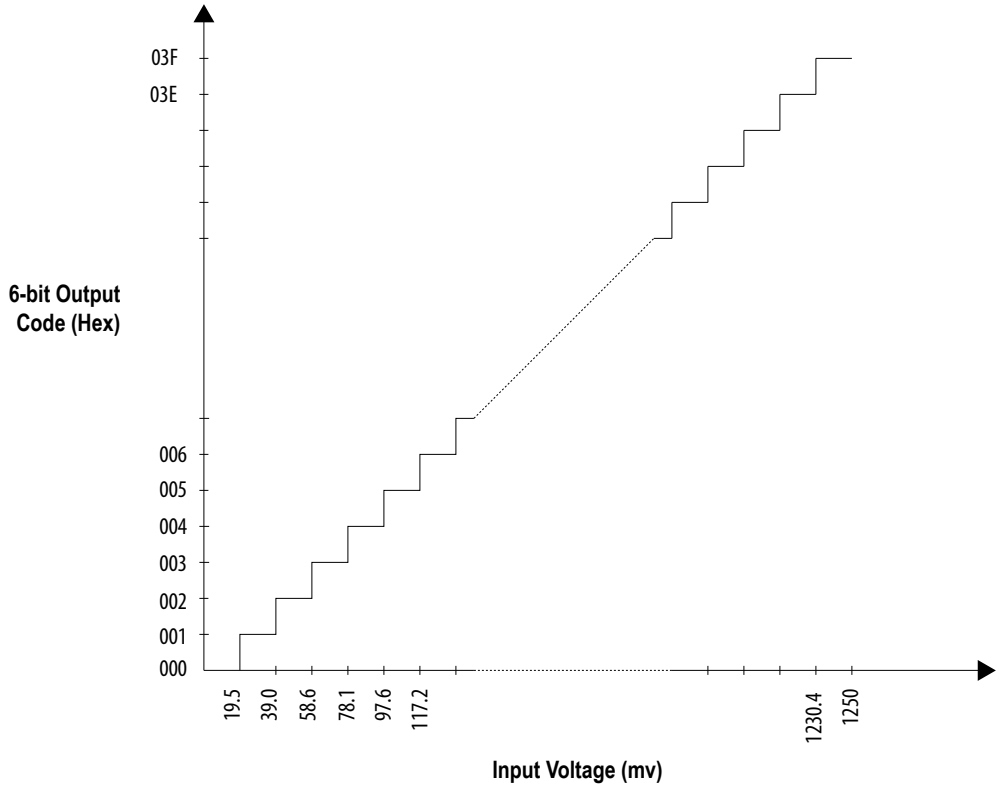
6. Specify the next channel for conversion on the chsel[3:0] signal. Data on the chsel[3:0] signal must be stable one cycle before the eoc signal asserts. Poll the eoc and eos status signals to check if conversion for the first channel defined by the chsel[3:0] signal in step 4 is complete. Latch the output data on the dataout[5:0] signal at the falling edge of the eoc signal.

7. Repeat step 6 for all the subsequent channels.

10.4.2.2. Voltage Sensor Transfer Function

The following figure shows the voltage sensor transfer function for the unipolar mode.

Figure 188. Voltage Sensor Transfer Function for the Unipolar Mode



10.5. Temperature Sensing Diode

The Intel Arria 10 temperature sensing diode (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 (ja) or junction-to-case (jc) thermal resistance, and device power consumption.

An Intel Arria 10 device monitors 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.

Table 126. Overview of the Internal and External Temperature Sensors

Feature	Internal Sensor	External TSD
Temperature sensing	Uses the built-in ADC to sample the on-chip temperature	Interfaces the TSD with an external temperature sensing chip
Readout access	Through the Temperature Sensor Intel FPGA IP	From the external temperature sensing chip
Operation availability	When the device is in user mode	When the device is in user mode or off

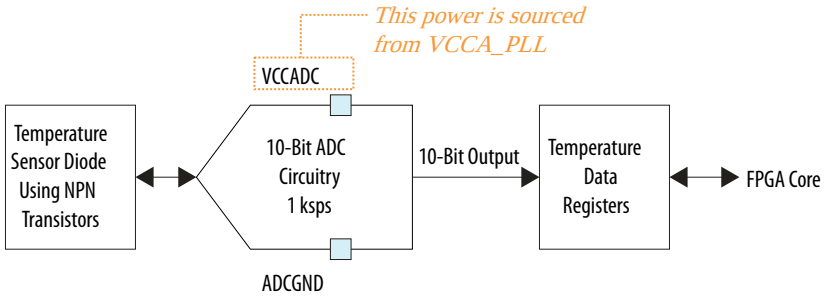
Related Information

Intel FPGA Temperature Sensor IP Core User Guide

10.5.1. Internal Temperature Sensing Diode

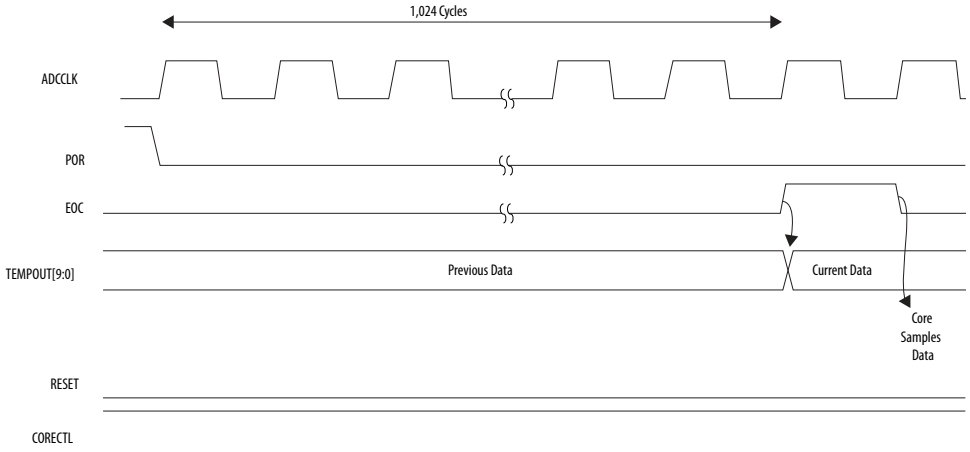
The Intel Arria 10 device supports an internal TSD with a built-in 10-bit ADC circuitry to monitor die temperature. The Intel Arria 10 device uses a set of NPN transistors to sense the temperature and generate its own reference voltage for conversion. The conversion speed of the internal TSD is around 1 ksps.

Figure 189. Internal TSD Block Diagram



To read the temperature of the die during user mode, assert the CORECTL signal from low to high. The active high RESET signal can be used to reset the registers at any time. The ADC circuitry takes 1,024 clock cycles to complete one conversion. The EOC signal goes high for one clock cycle indicating completion of the conversion. The FPGA core reads out the data on the TEMPOUT[9:0] signal at the falling edge of the EOC signal.

Figure 190. Internal TSD Timing Diagram



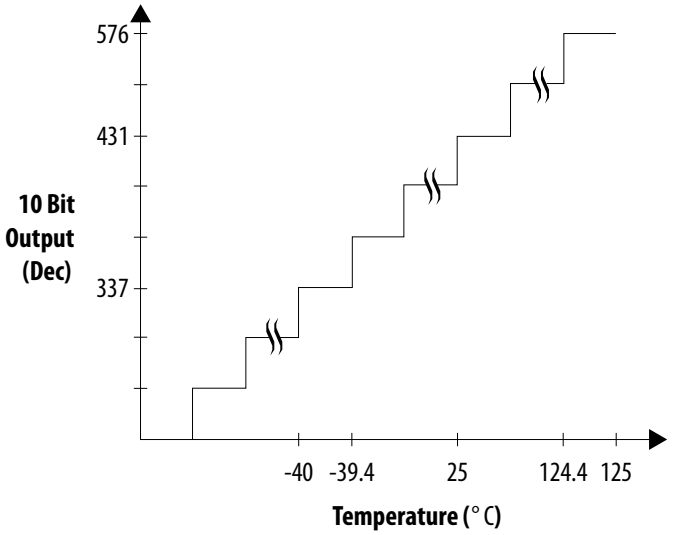
Related Information

- Internal Temperature Sensing Diode Specifications
  Provides more information about the Intel Arria 10 internal TSD specification.
- Intel FPGA Temperature Sensor IP Core User Guide

10.5.1.1. Transfer Function for Internal TSD

The following figure shows the transfer function for internal TSD.

Figure 191. ADC Transfer Function



You can calculate the temperature from `tempout[9:0]` value using this formula:

```
Temperature = {(AxC)÷1024} - B
```

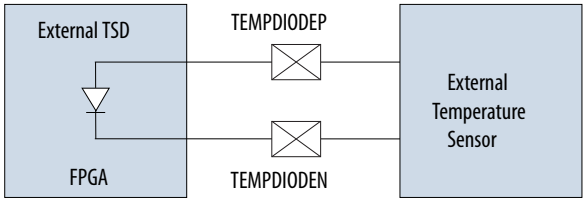
Where:

- A = 693
- B = 265
- C = decimal value of `tempout[9..0]`

10.5.2. External Temperature Sensing Diode

The Intel Arria 10 external TSD requires two pins for voltage reference. The following figure shows how to connect the external TSD with an external temperature sensor device to allow external sensing of the Intel Arria 10 die temperature.

Figure 192. 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 Intel Arria 10 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. Intel 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.
- Intel 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 and 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.

Note: You must ensure that the ideality factor setting of the temperature sensing device matches the ideality factor setting of the Intel Arria 10 external temperature sensing diode for optimal accuracy. For more information on the diode ideality factor, refer to the *Intel Arria 10 Device Datasheet*.

For details about device specification and connection guidelines, refer to the external temperature sensor device datasheet from the device manufacturer.

Related Information

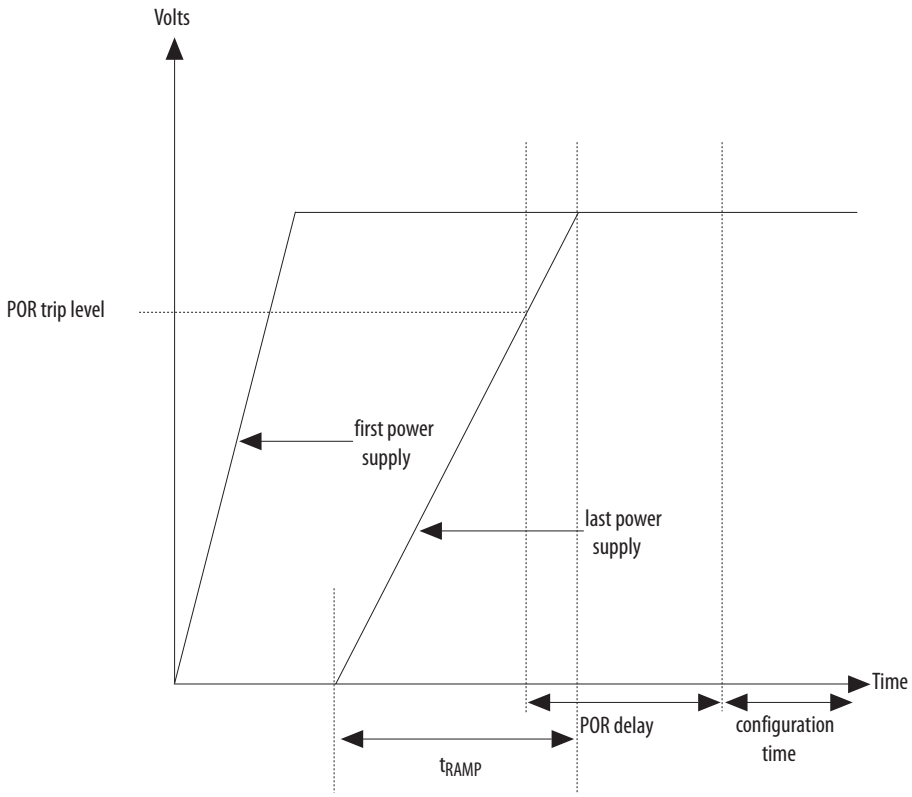
- External Temperature Sensing Diode Specifications
  Provides details about the external TSD specification.
- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
  Provides details about the $TEMPDIODE_P$/$TEMPDIODE_N$ pin connection when you are not using an external TSD.

10.6. Power-On Reset Circuitry

The POR circuitry keeps the Intel Arria 10 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 Intel Arria 10 device until all power supplies reach the recommended operating range within the maximum power supply ramp time, t_{RAMP}. If t_{RAMP} is not met, the Intel Arria 10 device I/O pins and programming registers remain tri-stated, during which device configuration could fail.

Figure 193. Relationship Between t_{RAMP} and POR Delay



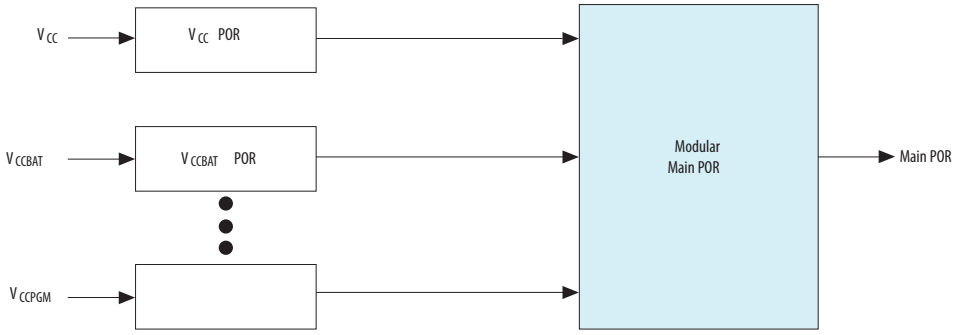
The Intel Arria 10 POR circuitry uses an individual detecting circuitry to monitor each of the configuration-related power supplies independently. The main POR circuitry is gated by the outputs of all the individual detectors. The main POR signal is asserted when the power starts to ramp up. This signal is released after the last ramp-up power reaches the POR trip level followed by a POR delay. You can select the fast or standard POR delay time by setting the MSEL pins.

For configuration via protocol (CvP), the total TRAMP must be less than 10 ms from the first power supply ramp-up to the last power supply ramp-up. Select a fast POR delay setting to allow sufficient time for the PCI Express* (PCIe*) link initialization and configuration.

In user mode, the main POR signal is asserted when any of the monitored power supplies go 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_{CCPT} 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 194. Simplified POR Diagram for Intel Arria 10 Devices



Related Information

- POR Specifications
  Provides more information about the POR delay specification.
- MSEL Pin Settings
  Provides more information about the MSEL pin settings for each POR delay.
- Recommended Operating Conditions
  Provides more information about the power supply ramp time.

10.6.1. Power Supplies Monitored and Not Monitored by the POR Circuitry

Table 127. Power Supplies Monitored and Not Monitored by the Intel Arria 10 POR Circuitry

Power Supplies Monitored	Power Supplies Not Monitored
• V_{CCBAT} • V_{CC} • V_{CCIO} (51) • V_{CCERAM} • V_{CCP} • V_{CCPT} • V_{CCPGM} • V_{CCL_HPS}(52), (53)	• V_{CCH_GXB} • V_{CCR_GXB} • V_{CCT_GXB} • V_{CCA_PLL} • V_{CCIO_HPS}(53) • V_{CCPLL_HPS} (53)

Note: For the device to exit POR, you must power the V_{CCBAT} power supply even if you do not use the volatile key.

10.7. Power Sequencing Considerations for Intel Arria 10 Devices

The Intel Arria 10 devices require a specific power-up and power-down sequence. This document describes several power management options and discusses proper I/O management during device power-up and power-down. Design your power supply solution to properly control the complete power sequence.

(51) Only for V_{CCIO} of bank 2A.

(52) V_{CCL_HPS} is a power supply monitored for the HPS block only and does not gate the main POR. If you do not use the HPS block, connect V_{CCL_HPS} to GND.

(53) These are only supported by system-on-a-chip (SoC) FPGA.

The requirements in this document must be followed to prevent unnecessary current draw to the FPGA device. Intel Arria 10 devices do not support 'Hot-Socketing' except under the conditions stated in the table below. The tables below also show what the unpowered pins can tolerate during power-up and power-down sequences.

Table 128. Pin Tolerance – Power-Up/Power-Down

'√' is Acceptable; '-' is Not Applicable.

	Power-Up				Power-Down			
Pin Type	**Tristate**	**Drive to GND**	**Drive to VCCIO**	**Driven with < 1.1 Vp-p**	**Tristate**	**Drive to GND**	**Drive to VCCIO**	**Driven with < 1.1 Vp-p**
3VIO banks	√	-	-	-	√	√	-	-
LVDS I/O banks	√	√	√(54)	-	√	√	√(54)	-
Differential Transceiver pins	√	√	-	√	√	√	-	√

Related Information

- LVDS I/O Pin Guidance for Unpowered FPGA
- Transceiver Pin Guidance for Unpowered FPGA

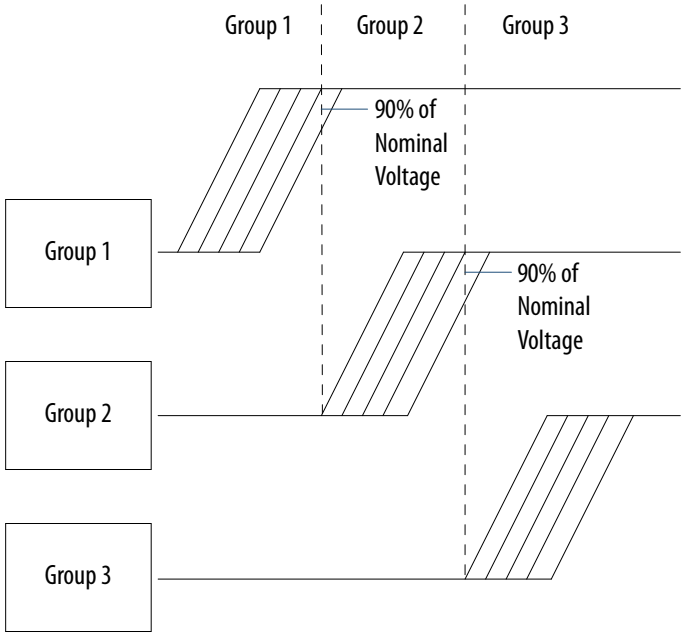
10.7.1. Power-Up Sequence Requirements for Intel Arria 10 Devices

The power rails in Intel Arria 10 devices are each divided into three groups. Refer to the *Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines* and the *AN692: Power Sequencing Considerations for Intel Cyclone 10 GX, Intel Arria 10, and Intel Stratix 10 Devices* for additional details.

The diagram below illustrates the voltage groups of the Intel Arria 10 devices and their required power-up sequence.

(54) The maximum current allowed through any LVDS I/O bank pin when the device is unpowered or during power up/down conditions = 10 mA (refer to "LVDS I/O Pin Guidance for Unpowered FPGA Pins").

Figure 195. Power-Up Sequence Requirement for Intel Arria 10 Devices



Note: VCCBAT is not in any of the groups below. VCCBAT does not have any sequence requirements. VCCBAT holds the contents of the security keys.

Table 129. Voltage Rails

Power Group	Intel Arria 10 Power Supplies
Group 1	V_{CC} V_{CCP} V_{CCERAM} V_{CCR_GXB} V_{CCT_GXB} V_{CCL_HPS}
Group 2	V_{CCPT} V_{CCH_GXB} V_{CCA_PLL} V_{CCPLL_HPS} $V_{CCIOREF\_HPS}$
Group 3	V_{CCPGM} V_{CCIO} V_{CCIO_HPS}

All power rails in Group 1 must ramp up (in any order) to a minimum of 90% of their respective nominal voltage before the power rails from Group 2 can start ramping up.

The power rails within Group 2 can ramp up in any order after the last power rail in Group 1 ramps to the minimum threshold of 90% of its nominal voltage. All power rails in Group 2 must ramp to a minimum threshold of 90% of their nominal value before the Group 3 power rails can start ramping up.

The power rails within Group 3 can ramp up in any order after the last power rail in Group 2 ramps up to a minimum threshold of 90% of their full value.

For Intel Arria 10 devices, you can combine and ramp up Group 3 power rails with Group 2 power rails if the two groups share the same voltage level and the same voltage regulator as Group 2 power rail V_{CCPT}.

Note: Ensure that the newly combined power rails do not cause any driving of unpowered GPIO or transceiver pins.

All power rails must ramp up monotonically. The power-up sequence should meet either the standard or the fast Power On Reset (POR) delay time. The POR delay time depends on the POR delay setting you use. For the POR specifications of the Intel Arria 10 devices, refer to the POR Specifications section in the *Intel Arria 10 Device Datasheet*.

The power-up sequence must meet either the standard or fast POR delay time depending on the POR delay setting you use.

Related Information

- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
- AN692: Power Sequencing Considerations for Intel Cyclone 10 GX, Intel Arria 10, and Intel Stratix 10 Devices
- POR Specifications

10.7.2. Power-Down Sequence Recommendations and Requirements for Intel Arria 10 Devices

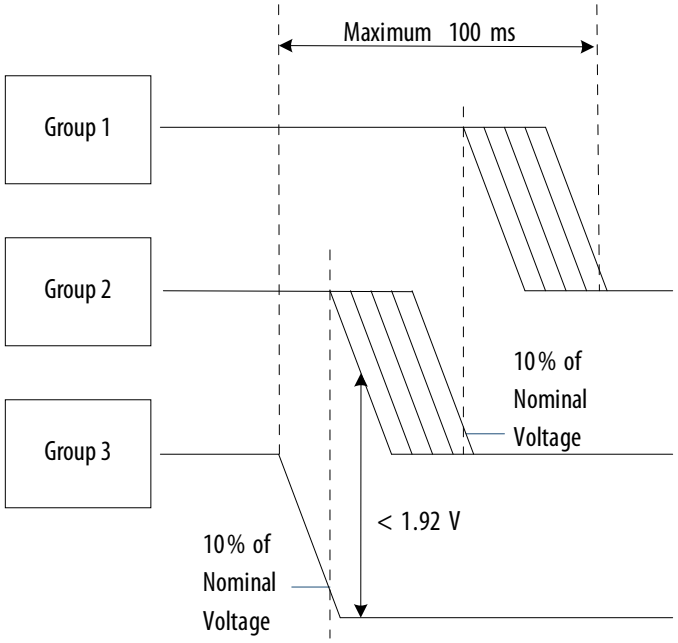
Intel's FPGAs need to follow certain requirements during a power-down sequence. The power-down sequence can be a controlled power-down event via an on/off switch or an uncontrolled event as with a power supply collapse. In either case, you must follow a specific power-down sequence. Below are four power-down sequence specifications. They are either Recommended (one), Required (two), or Relaxed (one). To comply with Intel’s FPGA Power-Down requirements, the Recommended option is best.

Note: If you cannot follow the Recommended specification, you must follow the Required specification.

Recommended Power-Down Ramp Specification

This is the best option to minimize power supply currents.

Figure 196. Recommended Power-Down Ramp Specification



- Power down all power rails fully within 100 ms.
- Power down power supplies within the same Group in any order.
- Before Group 2 supplies power down, power down all Group 3 supplies within 10% of GND.
- Before Group 1 supplies power down, power down all Group 2 supplies within 10% of GND.
- The maximum voltage differential between any Group 3 supply and any Group 2 supply is 1.92 V.

For Intel Arria 10 devices, you can combine and ramp down Group 3 power rails with Group 2 power rails if the two groups share the same voltage level and the same voltage regulator as the Group 2 power rails.

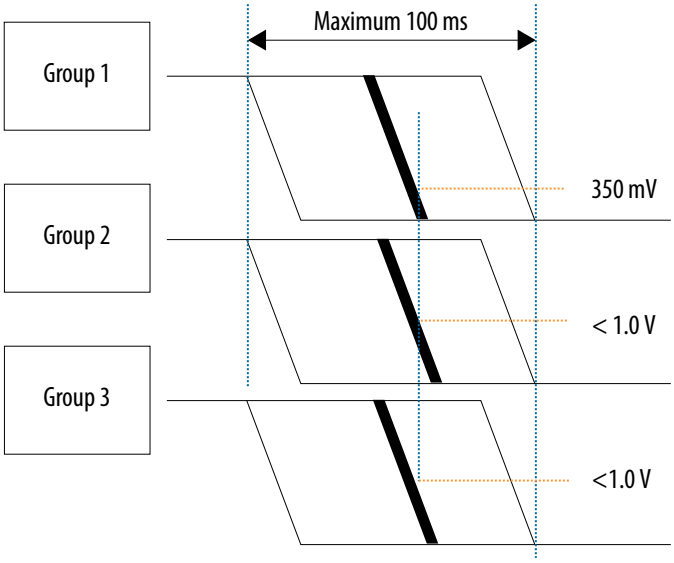
- Ensure that the newly combined power rails do not cause any driving of unpowered GPIO or transceiver pins.
- Ensure that the newly combined power rails do not violate any power-down sequencing specification due to device (third party) leakage; maintain the Required Voltage Differential Specification.

During the power-up/down sequence, the device output pins are tri-stated. To ensure long term reliability of the device, Intel recommends that you do not drive the input pins during this time.

Required Power-Down Ramp Specification

In cases where power supply is collapsing or if the recommended specification cannot be met, the following PDS sequence is required.

Figure 197. Required Power-Down Ramp Specification



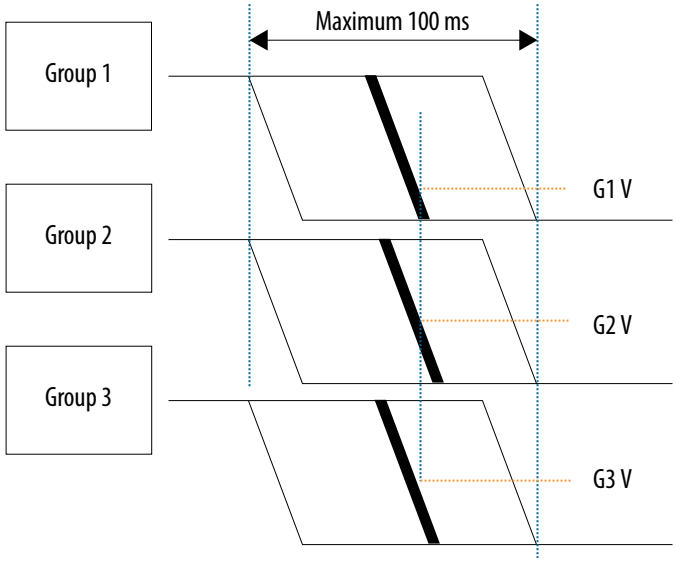
- Power down all power rails fully within 100 ms.
- As soon as possible, disable all power supplies.
  - Tri-state Group 1 supplies, and do not drive them actively to GND.
  - If possible, drive or terminate Group 2 and Group 3 supplies to GND.
- Ensure no alternative sourcing of any power supply exists during the power-down sequence; reduce all supplies monotonically and with a consistent RC typical decay.
- By the time any Group 1 supply goes under 0.35 V, all Group 2 and Group 3 supplies must be under 1.0 V.

Required Voltage Differential Specification

To not excessively overstress device transistors during power-down, there is an additional voltage requirement between any two power supplies between different power groups during power-down:

$\Delta V < \Delta V_{nom} + 500$ mV

Figure 198. Required Voltage Differential Specification



- Power down all power rails fully within 100 ms.
- For example, if Group 1 Voltage = 0.9 V, Group 2 Voltage = 1.8 V, and Group 3 Voltage = 3.0 V, then:

$G3V_{nom}$ = 3.0 V $G2V_{nom}$ = 1.8 V	$G2V_{nom}$ = 1.8 V $G1V_{nom}$ = 0.9 V	$G3V_{nom}$ = 3.0 V $G1V_{nom}$ = 0.9 V
$(G3V - G2V)_{nom}$ = 1.2 V	$(G2V - G1V)_{nom}$ = 0.9 V	$(G3V - G1V)_{nom}$ = 2.1 V
(G3V – G2V) <= 1.2 V + .5 V	(G2V – G1V) <= 0.9 V + .5 V	(G3V – G1V) <= 2.1 V + .5 V
(G3V – G2V) <= 1.7 V	(G2V – G1V) <= 1.4 V	(G3V – G1V) <= 2.6 V

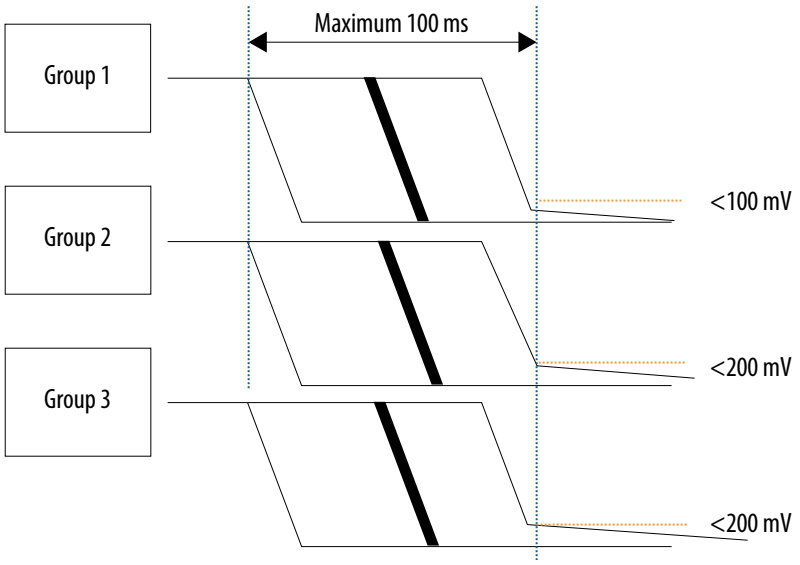
- To meet this voltage differential requirement, ramp down all power supplies as soon as possible according to the Required Power-Down Ramp Specification.

Note: Not following the required power sequence can result in unpredictable device operation and internal high current paths.

Relaxed Power-Down Duration Specification

For supplies being powered down with no active termination, voltage reduction to GND slows down as supply approaches 0 V. In this case, the 100 ms power requirement is relaxed - measure it when supply reaches near GND.

Figure 199. Relaxed Power-Down Duration Specification



- Ensure all Group 1 supplies reach < 100 mV within 100 ms.
- Ensure all Group 2 and Group 3 supplies reach < 200 mV within 100 ms.

Related Information

AN692: Power Sequencing Considerations for Intel Cyclone 10 GX, Intel Arria 10, and Intel Stratix 10 Devices

10.8. Power Supply Design

The power supply requirements for Intel Arria 10 devices vary depending on the static and dynamic power for each specific use case. To reduce dynamic power of the Intel Arria 10 device to a negligible amount before power down, hold the nCONFIG pin low to force the Intel Arria 10 device into a reset state. Intel's Enpirion® portfolio of power management solutions, combined with comprehensive design tools, enable optimized Intel Arria 10 device power supply design. The Enpirion portfolio includes power management solutions that are compatible with the multiple interface methods utilized by the Intel Arria 10 device and designed to support Intel Arria 10 power reduction features such as the SmartVID feature.

Intel Arria 10 devices have multiple input voltage rails that require a regulated power supply in order to operate. Multiple input rail requirements may be grouped according to system considerations such as voltage requirements, noise sensitivity, and sequencing. The *Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines* provides a more detailed recommendation about which input rails may be grouped. The Power Analyzer for Intel Arria 10 devices also seamlessly and automatically provides input rail power requirements and specific device recommendations based on each specific Intel Arria 10 use case. Individual input rail voltage and current requirements are summarized on the "Report" tab while input rail groupings and specific power supply recommendations can be found on the "Main" and "Enpirion" tabs, respectively.

Related Information

- Intel Arria 10 GX, GT, and SX Device Family Pin Connection Guidelines
  Provides detailed information about power supply pin connection guidelines and power regulator sharing.
- Early Power Estimators (EPE) and Power Analyzer
  Provides more information about the power supplies and the current requirements for each power rail.
- Intel FPGA Power Management PowerSoC Solutions
  Provides more information about Intel's Power Management IC and PowerSoC solutions designed for powering FPGAs.
- Power Delivery Network (PDN) Tool for Intel Stratix 10 and Intel Arria 10 Devices

10.9. Power Management in Intel Arria 10 Devices Revision History

Document Version	Changes
2019.12.30	Added *Overview of the Internal and External Temperature Sensors* table to clarify the differences between internal and external temperature sensors in *Temperature Sensing Diode* chapter.
2019.03.28	Updated the *Power-Up Sequence Requirements for Intel Arria 10 Devices* section to include information about Group 2 and Group 3 power rails.
2018.05.07	• Updated the *External Temperature Sensing Diode* section to include information about the diode ideality factor. • Updated the power-up and power-down sequences in the the *Power Sequencing Considerations for Intel Arria 10 Devices* section. • Editorial updates.

Date	Version	Changes
December 2017	2017.12.15	• Added a link to AN692 in the Power-Up and Power-Down Sequences section. • Updated RESET signal from HIGH to LOW in Internal TSD Timing Diagram.
March 2017	2017.03.15	• Rebranded as Intel. • Updated the Power-Up and Power-Down Sequences section.
October 2016	2016.10.31	• Removed the link to AN692. • Updated the Description for the Voltage Sensor Block WYSIWYG table. • Updated the topic about power-up and power-down sequences to stress that excess I/O pin current can affect device reliability and damage the device.
June 2016	2016.06.13	• Updated the value of the VID_EN pin in the Regulator Requirement for Altera SmartVID Solution table. • Updated the Power-Up and Power-Down Sequences section to include more information for the power-down sequence. • Added the Voltage Sensor Transfer Function for the Unipolar Mode figure.
May 2016	2016.05.02	• Updated the WYSIWYG Atom to Access the Voltage Sensor Block example. • Updated the voltage range and nominal voltage range in the Regulator Requirement for Altera SmartVID Solution table. • Updated the Description for the Voltage Sensor Block WYSIWYG table. • Updated the Description for the Core Access Configuration Register table.

continued...

Date	Version	Changes
		• Updated the condition for the Group 1 power-up sequence in the Power Groups Ramping Sequence table. • Updated the requirement for the CvP configuration scheme in the Power-On Reset Circuitry section. • Removed support for the V_{CC} PowerManager feature.
December 2015	2015.12.14	• Added a note to V_{CCIO} and V_{CCL_HPS} power rails in the Power Supplies Monitored and Not Monitored by the Arria 10 POR Circuitry table. • Updated the RESET and CORECTL signals of the Internal TSD Timing Diagram figure. • Updated the formula for the ADC Transfer Function. • Updated the supported speed grades devices for the SmartVID feature. • Updated the conditions for Group 1 in the Power Groups Ramping Sequence table. • Updated the Power-Up and Power-Down Sequences section. • Updated the Voltage Sensor section. • Updated the External Temperature Sensing Diode section. • Removed the Bipolar Input Mode support from the Voltage Sensor feature. • Removed the JTAG Access Mode support from the Voltage Sensor feature. • Removed the Voltage Sensor Transfer Function section.
November 2015	2015.11.02	• Updated the ADC Transfer Function figure. • Changed instances of Quartus II to Intel Quartus Prime.
June 2015	2015.06.15	• Added a note to describe the current for the VREFP_ADC and VREFN_ADC pins in the Voltage Sensor section. • Updated the ADC Transfer Function figure.
May 2015	2015.05.04	• Updated the Power-Up and Power-Down Sequences with requirements for the power-down sequence for each group of power rails. • Updated the description of the `config` port in Table 10-4. • Updated the Transfer Function for Internal TSD section with the formula to calculate temperature from the `tempout[9:0]` value. • Updated the supported parallel VID bit interface to 7 bit in the SmartVID and V_{CC} PowerManager Features Implementation section. • Updated the note to the voltage range of the SmartVID and V_{CC} PowerManager, in which the range includes tolerance. • Updated the on-chip reference source to ±10%.
January 2015	2015.01.23	• Updated the Unipolar Input Mode section. • Updated the on-chip reference source for the VREFP_ADC pin in the Voltage Sensor section. • Updated the steps in the Accessing the Voltage Sensor Using JTAG Access section. • Updated the description for `reset` and `corectl` ports in the Description for the Voltage Sensor Block WYSIWYG table. • Updated the Internal Temperature Sensing Diode section on how to read the temperature of the die during user mode. • Updated the Timing Diagram when MD[1:0] is not Equal to 2'b11 figure. • Updated the Timing Diagram when MD[1:0] is Equal to 2'b11 figure. • Updated the Internal TSD Timing Diagram figure.
August 2014	2014.08.18	• Added the SmartVID and V_{CC} PowerManager Features Implementation section. • Added the Using Voltage Sensor in Arria 10 Devices section. • Added the Transfer Function for Internal TSD section. • Added the Power Supply Design section.

continued...

Date	Version	Changes
		• Updated the Dynamic Power Equation section. • Updated the Power Reduction Techniques section. • Updated the SmartVID section. • Updated the Programmable Power Technology section. • Updated the Voltage Sensor section. • Updated the Power-Up and Power-Down Sequence section.
December 2013	2013.12.02	Initial release.

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Intel:
10AX057K1F40I1SG 10AS066K1F40I1SG 10AX066K1F40I1SG 10AX057K2F40I1SG 10AS057K2F40E1SG 10AS066K2F40E1SG 10AX057K2F40E1SG 10AX057K1F40E1SG 10AS057K1F40I1SG 10AS057K1F40E1SG 10AS066K2F40I1SG 10AX066K2F40E1SG 10AS057K2F40I1SG 10AX066K1F40E1SG 10AX066K2F40I1SG 10AS066K1F40E1SG