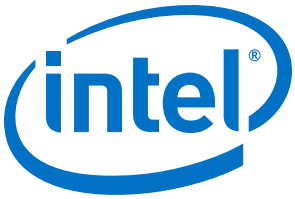


Intel® Cyclone® 10 LP Core Fabric and General Purpose I/Os Handbook

Subscribe
Send Feedback

C10LP51003 | 2020.12.03
Latest document on the web: PDF | HTML

Contents

1. Logic Elements and Logic Array Blocks in Intel® Cyclone® 10 LP Devices..... 6
1.1. Logic Elements.....6
1.1.1. LE Features.....7
1.1.2. LE Operating Modes.....8
1.2. Logic Array Block.....10
1.2.1. LAB Interconnects.....11
1.2.2. LAB Control Signals.....12
1.3. Logic Elements and Logic Array Blocks in Intel Cyclone 10 LP Devices Revision History..14

2. Embedded Memory Blocks in Intel Cyclone 10 LP Devices.....15
2.1. Embedded Memory Capacity.....15
2.2. Intel Cyclone 10 LP Embedded Memory General Features.....16
2.2.1. Control Signals.....16
2.2.2. Parity Bit.....17
2.2.3. Read Enable.....17
2.2.4. Byte Enable.....17
2.2.5. Read-During-Write.....19
2.2.6. Packed Mode Support.....19
2.2.7. Address Clock Enable Support.....19
2.2.8. Asynchronous Clear.....20
2.3. Intel Cyclone 10 LP Embedded Memory Operation Modes.....21
2.3.1. Supported Memory Operation Modes.....21
2.4. Intel Cyclone 10 LP Embedded Memory Clock Modes.....32
2.4.1. Asynchronous Clear in Clock Modes.....32
2.4.2. Output Read Data in Simultaneous Read and Write.....32
2.4.3. Independent Clock Enables in Clock Modes.....32
2.5. Intel Cyclone 10 LP Embedded Memory Configurations.....32
2.5.1. Port Width Configurations.....33
2.5.2. Memory Configurations for Dual-Port Modes.....33
2.5.3. Maximum Block Depth Configuration.....34
2.6. Intel Cyclone 10 LP Embedded Memory Design Consideration.....34
2.6.1. Implement External Conflict Resolution.....34
2.6.2. Customize Read-During-Write Behavior.....35
2.6.3. Consider Power-Up State and Memory Initialization.....37
2.6.4. Control Clocking to Reduce Power Consumption.....38
2.6.5. Selecting Read-During-Write Output Choices.....38
2.7. Embedded Memory Blocks in Intel Cyclone 10 LP Devices Revision History.....38

3. Embedded Multipliers in Intel Cyclone 10 LP Devices.....39
3.1. Embedded Multiplier Block Overview.....39
3.2. Multipliers Resources in Intel Cyclone 10 LP Devices.....41
3.3. Embedded Multipliers Architecture.....41
3.3.1. Input Register.....42
3.3.2. Multiplier Stage.....42
3.3.3. Output Register.....43
3.4. Embedded Multipliers Operational Modes.....43
3.4.1. 18-Bit Multipliers.....44
3.4.2. 9-Bit Multipliers.....44

3.5. Embedded Multipliers in Intel Cyclone 10 LP Devices Revision History 46

4. Clock Networks and PLLs in Intel Cyclone 10 LP Devices 47

4.1. Clock Networks 47
4.1.1. GCLK Network 47
4.1.2. GCLK Network Sources 47
4.1.3. Clock Control Block 49
4.1.4. GCLK Network Clock Source Generation 51
4.1.5. GCLK Network Power Down 52
4.1.6. Clock Enable Signals 53
4.2. PLLs in Intel Cyclone 10 LP Devices 54
4.2.1. PLL Features 54
4.2.2. PLL Architecture 55
4.2.3. External Clock Outputs 56
4.2.4. Clock Feedback Modes 57
4.2.5. Clock Multiplication and Division 59
4.2.6. Post-Scale Counter Cascading 60
4.2.7. Programmable Duty Cycle 61
4.2.8. PLL Control Signals 61
4.2.9. Clock Switchover 61
4.2.10. Programmable Bandwidth 66
4.2.11. Programmable Phase Shift 66
4.2.12. PLL Cascading 67
4.2.13. PLL Reconfiguration 67
4.2.14. Spread-Spectrum Clocking 74
4.3. Clock Networks and PLLs in Intel Cyclone 10 LP Devices Revision History 75

5. I/O and High Speed I/O in Intel Cyclone 10 LP Devices 76

5.1. Intel Cyclone 10 LP I/O Standards Support 76
5.1.1. Intel Cyclone 10 LP I/O Standards Voltage and Pin Support 78
5.2. I/O Resources in Intel Cyclone 10 LP Devices 80
5.2.1. Intel Cyclone 10 LP Devices I/O Resources Per Package 80
5.2.2. Intel Cyclone 10 LP I/O Vertical Migration 81
5.2.3. Intel Cyclone 10 LP VREF Pins Per I/O Bank 81
5.2.4. Intel Cyclone 10 LP LVDS Channels Support 82
5.3. Intel FPGA I/O IP Cores for Intel Cyclone 10 LP Devices 84
5.4. Intel Cyclone 10 LP I/O Elements 85
5.4.1. Intel Cyclone 10 LP I/O Banks Architecture 86
5.4.2. Intel Cyclone 10 LP I/O Banks Locations 86
5.5. Intel Cyclone 10 LP Clock Pins Input Support 87
5.6. Programmable IOE Features in Intel Cyclone 10 LP Devices 88
5.6.1. Programmable Open Drain 88
5.6.2. Programmable Bus Hold 88
5.6.3. Programmable Pull-Up Resistor 89
5.6.4. Programmable Current Strength 89
5.6.5. Programmable Output Slew Rate Control 90
5.6.6. Programmable IOE Delay 91
5.6.7. PCI Clamp Diode 92
5.6.8. Programmable Pre-Emphasis 93
5.7. I/O Standards Termination 93
5.7.1. Voltage-Referenced I/O Standards Termination 94

5.7.2. Differential I/O Standards Termination....95
5.7.3. Intel Cyclone 10 LP On-Chip I/O Termination....96
5.8. Intel Cyclone 10 LP High-Speed Differential I/Os and SERDES....99
5.8.1. High-Speed Differential I/O Interface....100
5.8.2. Differential I/O Standards Support....100
5.8.3. High-Speed I/O Timing Budget....107
5.9. Using the I/Os and High Speed I/Os in Intel Cyclone 10 LP Devices....109
5.9.1. Guideline: Validate Your Pin Placement....109
5.9.2. Guideline: Check for Illegal Pad Placements....109
5.9.3. Guideline: Voltage-Referenced I/O Standards Restriction....110
5.9.4. Guideline: Simultaneous Usage of Multiple I/O Standards....110
5.9.5. Guideline: LVTTL or LVCMOS Inputs in Intel Cyclone 10 LP Devices....111
5.9.6. Guideline: Differential Pad Placement....111
5.9.7. Guideline: Board Design for Signal Quality....111
5.10. I/O and High Speed I/O in Intel Cyclone 10 LP Devices Revision History....112

6. Configuration and Remote System Upgrades....113
6.1. Configuration Schemes....113
6.1.1. Active Serial (AS) Configuration....113
6.1.2. Passive Serial Configuration....119
6.1.3. Fast Passive Parallel Configuration....126
6.1.4. JTAG Configuration....129
6.2. Configuration Requirement....141
6.2.1. Power-On Reset (POR) Circuit....141
6.2.2. Configuration File Size....142
6.2.3. Configuration and JTAG Pin I/O Requirements....142
6.3. Configuration Details....143
6.3.1. MSEL Pin Settings....143
6.3.2. Configuration Sequence....145
6.3.3. Configuration Timing Waveforms....148
6.3.4. Device Configuration Pins....150
6.4. Configuration Data Compression....151
6.4.1. Enabling Compression Before Design Compilation....152
6.4.2. Enabling Compression After Design Compilation....152
6.4.3. Using Compression in Multi-Device Configuration....153
6.5. Remote System Upgrades....153
6.5.1. Enabling Remote Update....154
6.5.2. Configuration Sequence in the Remote Update Mode....155
6.5.3. Remote System Upgrade Circuitry....156
6.5.4. Remote System Upgrade Registers....156
6.5.5. Remote System Upgrade State Machine....159
6.5.6. User Watchdog Timer....160
6.6. Configuration and Remote System Upgrades in Intel Cyclone 10 LP Devices Revision History....160

7. SEU Mitigation in Intel Cyclone 10 LP Devices....162
7.1. Configuration Error Detection....162
7.2. User Mode Error Detection....162
7.3. Error Detection Features....163
7.3.1. Error Detection Block....163
7.4. Using Error Detection Features in User Mode....166
7.4.1. Enabling Error Detection....166

7.4.2. Accessing Error Detection Block Through User Logic.....166
7.5. SEU Mitigation for Intel Cyclone 10 LP Devices Revision History.....168

8. JTAG Boundary-Scan Testing for Intel Cyclone 10 LP Devices.....170
8.1. BST Operation Control.....170
8.2. I/O Voltage Support in the JTAG Chain.....170
8.3. Boundary-Scan Description Language Support.....171
8.4. JTAG Boundary-Scan Testing for Intel Cyclone 10 LP Devices Revision History.....171

9. Power Management in Intel Cyclone 10 LP Devices.....172
9.1. External Power Supply Requirements.....172
9.2. Hot-Socketing Specifications.....172
9.2.1. Drive Intel Cyclone 10 LP Devices Before Power Up.....173
9.2.2. I/O Pins Remain Tri-stated During Power Up.....173
9.3. Hot-Socketing Feature Implementation.....173
9.4. Power-On Reset Circuitry.....173
9.5. Power Management in Intel Cyclone 10 LP Devices Revision History.....174

1. Logic Elements and Logic Array Blocks in Intel® Cyclone® 10 LP Devices

The logic array block (LAB) is composed of logic elements (LEs). You can configure the LABs to implement logic functions, arithmetic functions, and register functions.

1.1. Logic Elements

LE is the smallest unit of logic in the Intel® Cyclone® 10 LP device family architecture. LEs are compact and provide advanced features with efficient logic usage.

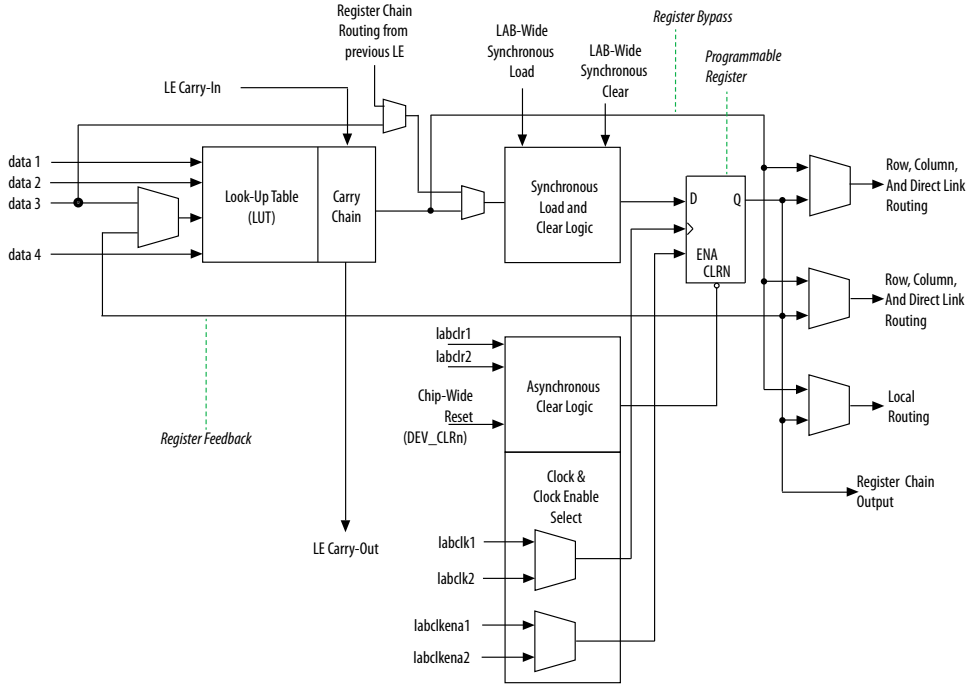
Each LE has the following features:

- A four-input look-up table (LUT) that can implement any function of four variables
- A programmable register
- A carry chain connection
- A register chain connection
- The ability to drive the following interconnects:
  - Local
  - Row
  - Column
  - Register chain
  - Direct link
- Register packing support
- Register feedback support

1.1.1. LE Features

LEs contain inputs, outputs, and registers to enable several features.

Figure 1. LE High-Level Block Diagram for Intel Cyclone 10 LP Devices.



LE Inputs

Each LE input is directed to different destinations to implement the desired logic function. In both the normal or arithmetic operating modes of the LE, there are six available inputs:

- Four data inputs from the LAB local interconnect
- One LE carry-in from the previous LE carry-chain
- One register chain connection

LE Outputs

Each LE has three general routing outputs:

- Two LE outputs drive the column or row and direct link routing connections
- One LE output drives the local interconnect resources

Intel Cyclone 10 LP devices support register packing. With register packing, the LUT or register output drives the three outputs independently. This feature improves device utilization by using the register and the LUT for unrelated functions.

The LAB-wide synchronous load control signal is not available if you use register packing.

Register Chain Output

Each LE has a register chain output that allows registers in the same LAB to cascade together. This feature speeds up connections between LABs and optimizes local interconnect resources:

- LUTs are used for combinational functions
- Registers are used for an unrelated shift register implementation

Programmable Register

You can configure the programmable register of each LE for D, T, JK, or SR flipflop operation. Each register has the following inputs:

- Clock—driven by signals that use the global clock network, general-purpose I/O pins, or internal logic
- Clear—driven by signals that use the global clock network, general-purpose I/O pins, or internal logic
- Clock enable—driven by the general-purpose I/O pins or internal logic

For combinational functions, the LUT output bypasses the register and drives directly to the LE outputs.

Register Feedback

The register feedback mode allows the register output to feed back into the LUT of the same LE. Register feedback ensures that the register is packed with its own fan-out LUT, providing another mechanism for improving fitting. The LE can also drive out registered and unregistered versions of the LUT output.

1.1.2. LE Operating Modes

The LEs in Intel Cyclone 10 LP devices operate in two modes.

- Normal mode
- Arithmetic mode

These operating modes use LE resources differently. Both LE modes have six available inputs and LAB-wide signals.

The Intel Quartus® Prime software automatically chooses the appropriate mode for common functions, such as counters, adders, subtractors, and arithmetic functions, in conjunction with parameterized functions such as the library of parameterized modules (LPM) functions.

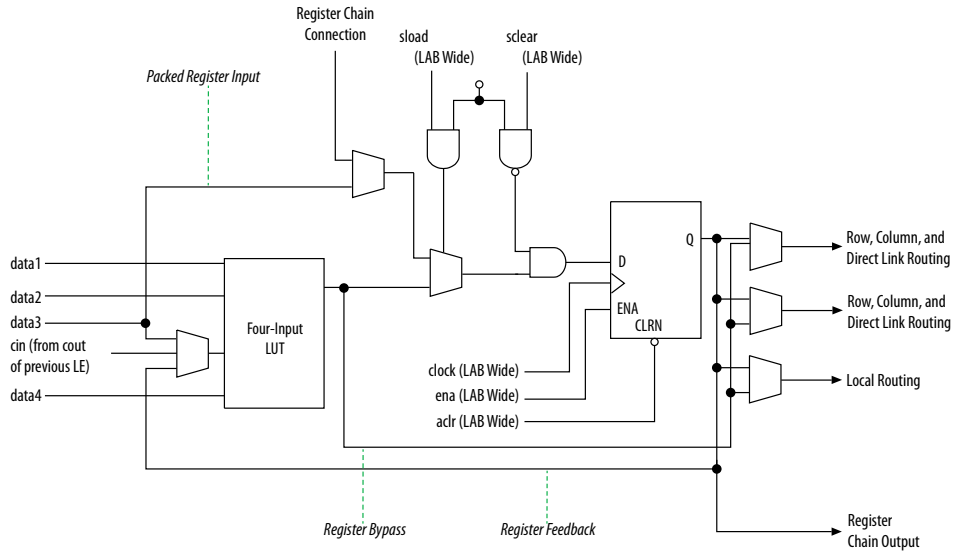
You can also create special-purpose functions that specify which LE operating mode to use for optimal performance.

1.1.2.1. Normal Mode

Normal mode is suitable for general logic applications and combinational functions.

In normal mode, four data inputs from the LAB local interconnect are inputs to a four-input LUT. The Intel Quartus Prime Compiler automatically selects the carry-in (cin) or the data3 signal as one of the inputs to the LUT. LEs in normal mode support packed registers and register feedback.

Figure 2. LE Operating in Normal Mode for Intel Cyclone 10 LP devices

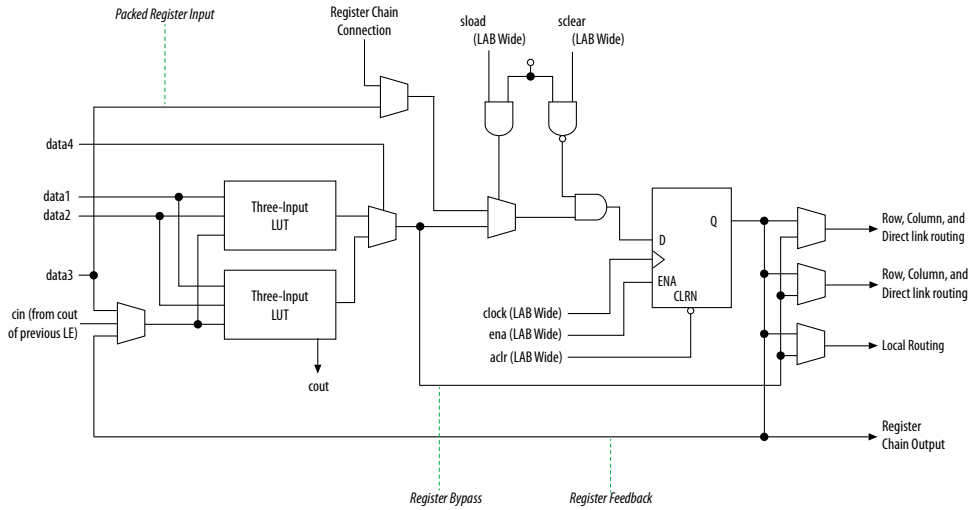


1.1.2.2. Arithmetic Mode

Arithmetic mode is ideal for implementing adders, counters, accumulators, and comparators.

The LE in arithmetic mode implements a two-bit full adder and basic carry chain. LEs in arithmetic mode can drive out registered and unregistered versions of the LUT output. Register feedback and register packing are supported when LEs are used in arithmetic mode.

Figure 3. LE Operating in Arithmetic Mode for Intel Cyclone 10 LP devices



Carry Chain

The Intel Quartus Prime Compiler automatically creates carry chain logic during design processing. You can also manually create the carry chain logic during design entry. Parameterized functions, such as LPM functions, automatically take advantage of carry chains for the appropriate functions. The Intel Quartus Prime Compiler creates carry chains longer than 16 LEs by automatically linking LABs in the same column.

To enhanced fitting, a long carry chain runs vertically, which allows fast horizontal connections to M9K memory blocks or embedded multipliers through direct link interconnects. For example, if a design has a long carry chain in an LAB column next to a column of M9K memory blocks, any LE output can feed an adjacent M9K memory block through the direct link interconnect.

If the carry chains run horizontally, any LAB which is not next to the column of M9K memory blocks uses other row or column interconnects to drive a M9K memory block.

A carry chain continues as far as a full column.

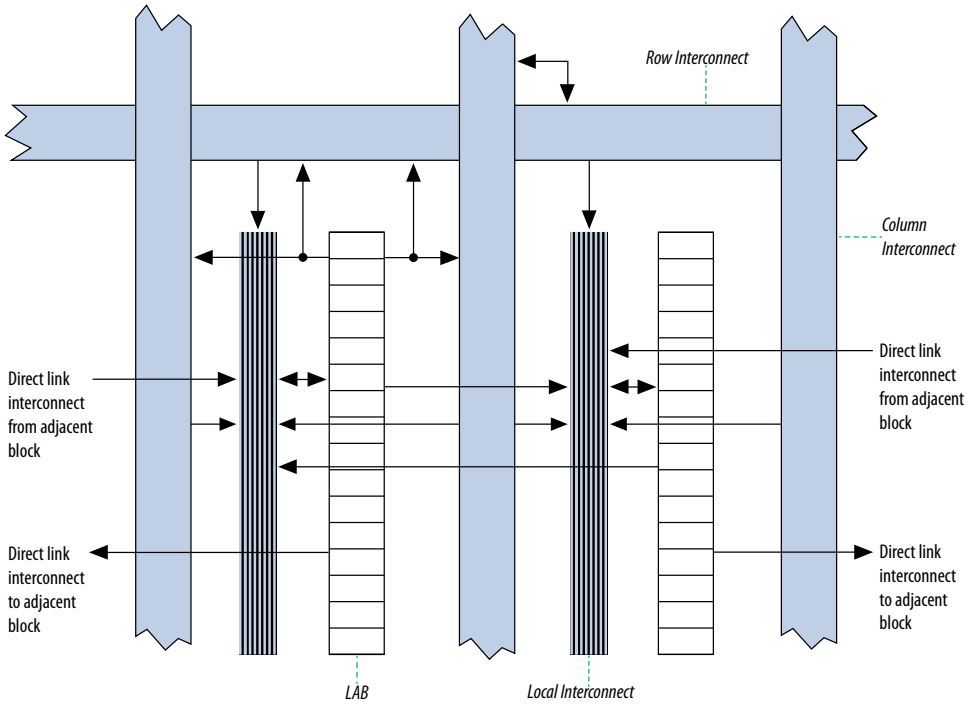
1.2. Logic Array Block

The LABs are configurable logic blocks that consist of a group of logic resources.

Each LAB consists of the following:

- 16 logic elements (LEs)—smallest logic unit in Intel Cyclone 10 LP devices
- LE carry chains—carry chains propagated serially through each LE within an LAB
- LAB control signals—dedicated logic for driving control signals to LEs within an LAB
- Local interconnect—transfers signals between LEs in the same LAB
- Register chains—transfers the output of one LE register to the adjacent LE register in an LAB

Figure 4. LAB Structure of Intel Cyclone 10 LP Devices



The Intel Quartus Prime Compiler places associated logic in an LAB or adjacent LABs, allowing the use of local and register chain connections for performance and area efficiency.

1.2.1. LAB Interconnects

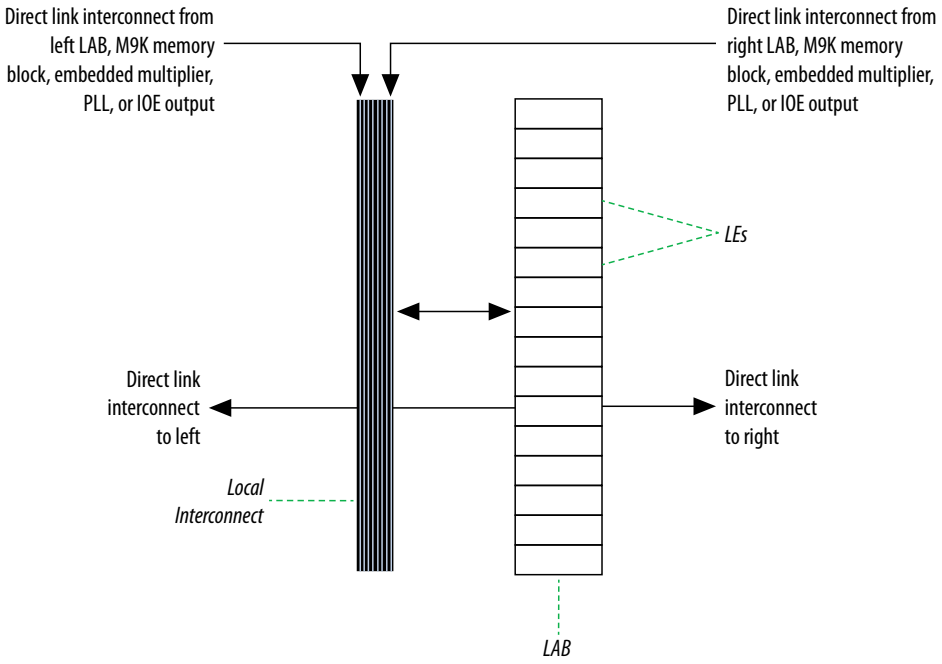
The LAB local interconnect is driven by column and row interconnects and LE outputs in the same LAB.

The direct link connection minimizes the use of row and column interconnects to provide higher performance and flexibility. The direct link connection enables the neighboring elements from left and right to drive the local interconnect of an LAB. The elements are:

- LABs
- PLLs
- M9K embedded memory blocks
- Embedded multipliers

Each LE can drive up to 48 LEs through local and direct link interconnects.

Figure 5. LAB Local and Direct Link Interconnects for Intel Cyclone 10 LP Devices



1.2.2. LAB Control Signals

Each LAB contains dedicated logic for driving control signals to its LEs.

The control signals include:

- Two clock signals
- Two clock enable signals
- Two asynchronous clear signals
- One synchronous clear signal
- One synchronous load signal

Figure 6. LAB-Wide Control Signals for Intel Cyclone 10 LP Devices

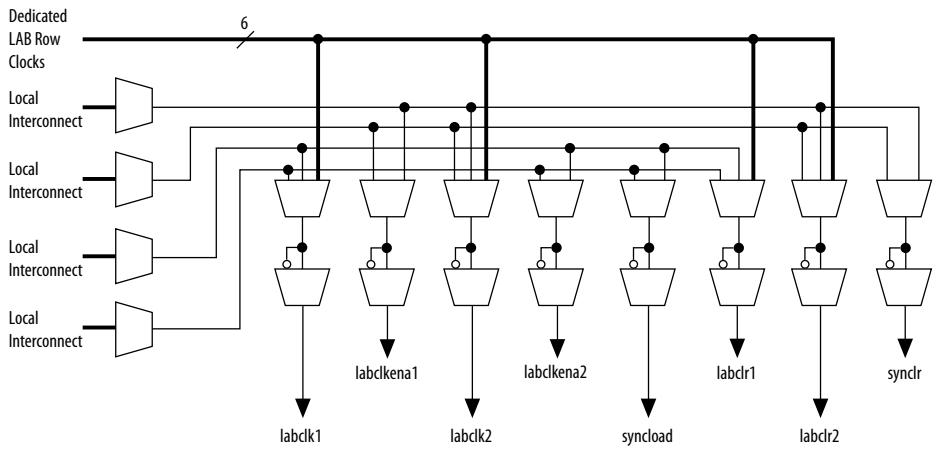


Table 1. Control Signal Descriptions for Intel Cyclone 10 LP Devices

Control Signal	Description
`labclk1` `labclk2`	• Each LAB can use two clocks signals. The clock and clock enable signals of each LAB are linked. For example, any LE in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. • If the LAB uses both the rising and falling edges of a clock, it also uses both LAB-wide clock signals. • The LAB row clocks `[5..0]` and LAB local interconnect generate the LAB-wide clock signals. The MultiTrack interconnect inherent low skew allows clock and control signal distribution in addition to data distribution.
`labclkena1` `labclkena2`	• Each LAB can use two clock enable signals. The clock and clock enable signals of each LAB are linked. For example, any LE in a particular LAB using the `labclk1` signal also uses the `labclkena1` signal. • Deasserting the clock enable signal turns off the LAB-wide clock signal.
`labclr1` `labclr2`	Asynchronous clear signals: • LAB-wide control signals that control the logic for the clear signal of the register. • The LE directly supports an asynchronous clear function.
`syncload` `synclr`	Synchronous load and synchronous clear signals: • Can be used for implementing counters and other functions • LAB-wide control signals that affect all registers in the LAB

You can use up to eight control signals at a time. Register packing and synchronous load cannot be used simultaneously.

Each LAB can have up to four non-global control signals. You can use additional LAB control signals as long as they are global signals.

An LAB-wide asynchronous load signal to control the logic for the preset signal of the register is not available. The register preset is achieved with a NOT gate push-back technique. Intel Cyclone 10 LP devices only support either a preset or asynchronous clear signal.

In addition to the clear port, Intel Cyclone 10 LP devices provide a chip-wide reset pin (DEV_CLRn) to reset all registers in the device. An option set before compilation in the Intel Quartus Prime software controls this pin. This chip-wide reset overrides all other control signals.

1.3. Logic Elements and Logic Array Blocks in Intel Cyclone 10 LP Devices Revision History

Date	Version	Changes
May 2017	2017.05.08	Initial release.

2. Embedded Memory Blocks in Intel Cyclone 10 LP Devices

The Intel Cyclone 10 LP embedded memory structure consists of columns of M9K memory block. You can configure each M9K block in different widths and configurations to provide various memory functions such as RAM, ROM, shift registers, and FIFO buffers.

Related Information

- Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Provides more information about the supported modes, signals, and parameters of the embedded memory IP cores that support the M9K blocks in the Intel Cyclone 10 LP devices.
- Introduction to Intel FPGA IP Cores
  Provides general information about all Intel FPGA IP cores, including parameterizing, generating, upgrading, and simulating IP cores.
- Creating Version-Independent IP and Qsys Simulation Scripts
  Create simulation scripts that do not require manual updates for software or IP version upgrades.
- Project Management Best Practices
  Guidelines for efficient management and portability of your project and IP files.

2.1. Embedded Memory Capacity

Table 2. Embedded Memory Capacity in Intel Cyclone 10 LP Devices

Device	M9K Blocks	RAM Capacity (Kb)
10CL006	30	270
10CL010	46	414
10CL016	56	504
10CL025	66	594
10CL040	126	1,134
10CL055	260	2,340
10CL080	305	2,745
10CL120	432	3,888

2.2. Intel Cyclone 10 LP Embedded Memory General Features

Intel Cyclone 10 LP embedded memory supports the following general features:

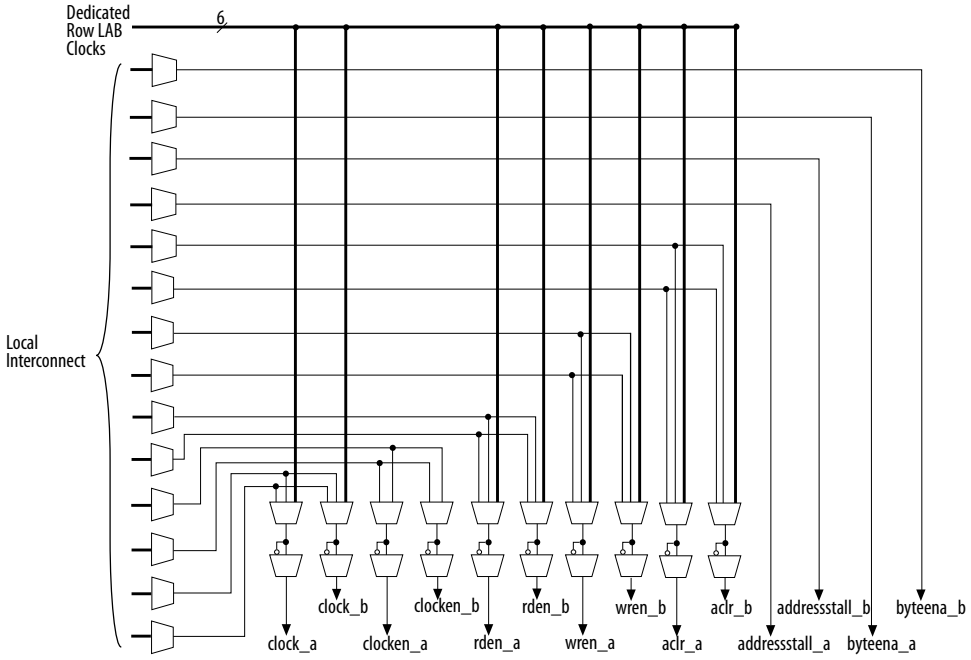
- 8,192 memory bits per block (9,216 bits per block including parity).
- Independent read-enable (`rden`) and write-enable (`wren`) signals for each port.
- Packed mode in which the M9K memory block is split into two 4.5 K single-port RAMs.
- Variable port configurations.
- Single-port and simple dual-port modes support for all port widths.
- True dual-port (one read and one write, two reads, or two writes) operation.
- Byte enables for data input masking during writes.
- Two clock-enable control signals for each port (port A and port B).
- Initialization file to preload memory content in RAM and ROM modes.

2.2.1. Control Signals

The clock-enable control signal controls the clock entering the input and output registers and the entire M9K memory block. This signal disables the clock so that the M9K memory block does not see any clock edges and does not perform any operations.

The `rden` and `wren` control signals control the read and write operations for each port of the M9K memory blocks. You can disable the `rden` or `wren` signals independently to save power whenever the operation is not required.

Figure 7. Register Clock, Clear, and Control Signals Implementation in M9K Embedded Memory Block



2.2.2. Parity Bit

You can perform parity checking for error detection with the parity bit along with internal logic resources. The M9K memory blocks support a parity bit for each storage byte. You can use this bit as either a parity bit or as an additional data bit. No parity function is actually performed on this bit. If error detection is not desired, you can use the parity bit as an additional data bit.

2.2.3. Read Enable

M9K memory blocks support the read enable feature for all memory modes.

Table 3. Effects of Read Enable on Data Output Port

If you...	...Then
Create the read-enable port and perform a write operation with the read enable port deasserted.	The data output port retains the previous values from the most recent active read enable.
Activate the read enable during a write operation or do not create a read-enable signal.	The output port shows either the new data being written and the old data at that address, or a "Don't Care" value when read-during-write occurs at the same address location.

2.2.4. Byte Enable

If you implement the M9K memory as RAM blocks, the embedded memory supports the byte enable features.

The byte enable features mask the input data to enable the writing of only specific bytes. The unwritten bytes retain the previous values. The write enable signal, `wren`, together with the byte enable signal, `byteena`, control the write operations on the RAM blocks. By default, the `byteena` signal is enabled (high) and only the `wren` signal controls the writing.

The M9K blocks support byte enables when the write port has a data width of ×16, ×18, ×32, or ×36 bits. In True Dual-Port memory configuration, byte enables are available only if both PortA and PortB data widths of each M9K memory blocks are multiples of 8 or 9 bits.

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

Byte enables are active high. The byte enable registers do not have a `clear` port.

2.2.4.1. Byte Enable Controls

Table 4. M9K Blocks Byte Enable Selections

byteena[3:0]	Affected Bytes. Any Combination of Byte Enables is Possible.			
	datain x 16	datain x 18	datain x 32	datain x 36
[0] = 1	[7:0]	[8:0]	[7:0]	[8:0]
[1] = 1	[15:8]	[17:9]	[15:8]	[17:9]
[2] = 1	—	—	[23:16]	[26:18]
[3] = 1	—	—	[31:24]	[35:27]

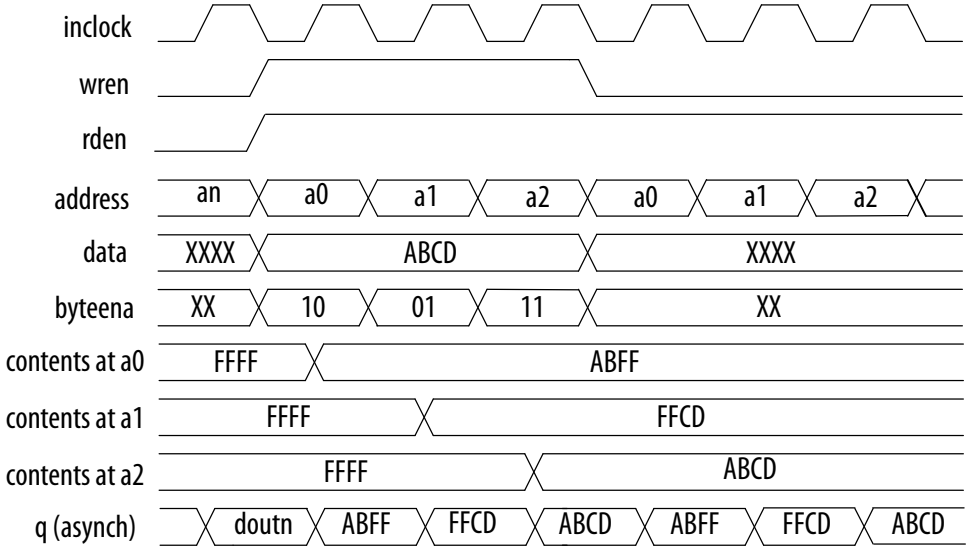
2.2.4.2. Data Byte Output

If you...	...Then
Deassert a byte-enable bit during a write cycle	The old data in the memory appears in the corresponding data-byte output.
Assert a byte-enable bit during a write cycle	The corresponding data-byte output depends on the Intel Quartus Prime software setting. The setting can be either the newly written data or the old data at that location.

2.2.4.3. RAM Blocks Operations

Figure 8. Byte Enable Functional Waveform

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



For this functional waveform, New Data Mode is selected.

2.2.5. Read-During-Write

The read-during-write operation occurs when a read operation and a write operation target the same memory location at the same time.

The read-during-write operation operates in the following ways:

- Same-port
- Mixed-port

2.2.6. Packed Mode Support

You can implement two single-port memory blocks in a single block under the following conditions:

- Each of the two independent block sizes is less than or equal to half of the M9K block size. The maximum data width for each independent block is 18 bits wide.
- Each of the single-port memory blocks is configured in single-clock mode.

Related Information

- Supported Memory Operation Modes on page 21
  Provides more information about the single-port mode.
- Intel Cyclone 10 LP Embedded Memory Clock Modes on page 32
  Provides more information about the single-clock mode.

2.2.7. Address Clock Enable Support

You can use the address clock enable feature to improve the effectiveness of cache memory applications during a cache-miss. If you configure M9K memory blocks in dual-port mode, each port has its own independent address clock enable.

By default, the address clock enable signal, `addressstall`, is disabled and the signal is active low. While the `addressstall` signal is high (`addressstall = 1`), the address register holds the previous address value.

Figure 9. Address Clock Enable Block Diagram

In this diagram, the address register output feeds back to its input using a multiplexer. The `addressstall` signal selects the multiplexer output.

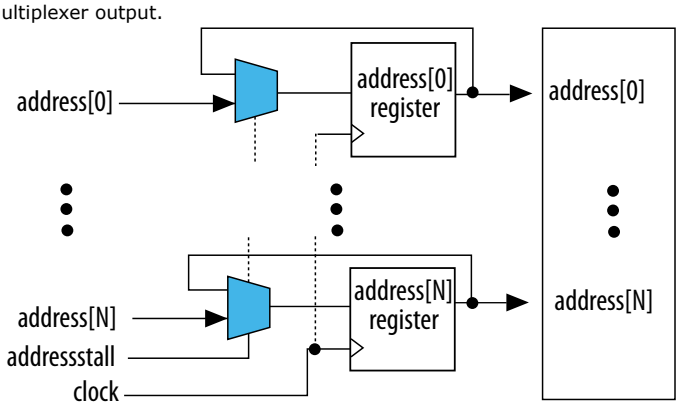


Figure 10. Address Clock Enable Waveform During Read Cycle

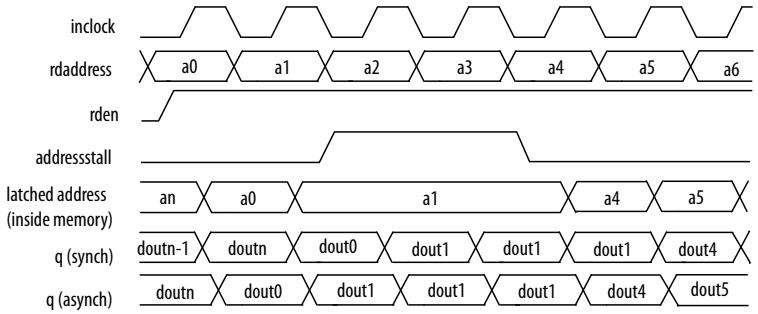
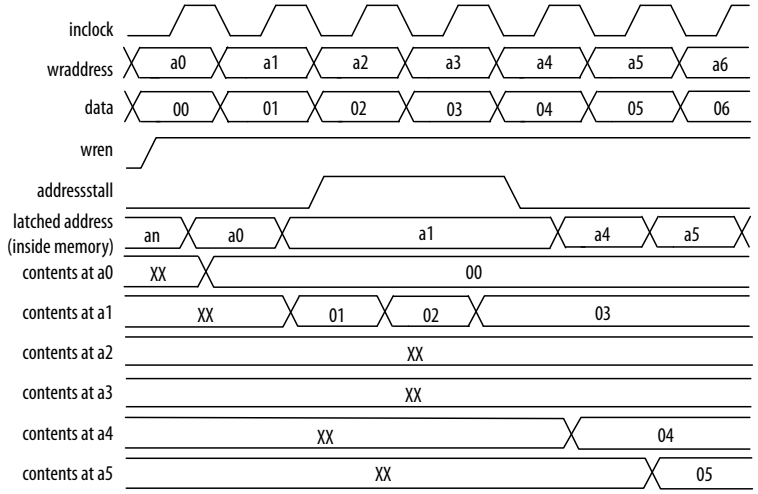


Figure 11. Address Clock Enable Waveform During Write Cycle



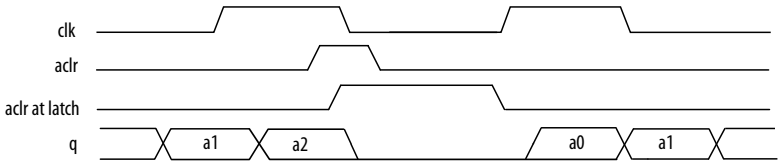
2.2.8. Asynchronous Clear

You can selectively enable asynchronous clear per logical memory using the RAM: 1-PORT and RAM: 2-PORT IP cores.

Support of asynchronous clear in the M9k memory block:

- Read address registers—input registers other than read address registers are not supported. Asserting asynchronous clear to the read address register during a read operation might corrupt the memory content.
- Output registers—if applied to output registers, the asynchronous clear signal clears the output registers and the effects are immediate. If your RAM does not use output registers, you can still clear the RAM outputs using the output latch asynchronous clear feature.
- Output latches

Figure 12. Output Latch Asynchronous Clear Waveform



Related Information

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

Provides more information about asynchronous clears in the embedded memory IP core.

2.2.8.1. Resetting Registers in M9K Blocks

There are three ways to reset registers in the M9K blocks:

- Power up the device
- Use the `aclr` signal for output register only
- Assert the device-wide reset signal using the **DEV_CLRn** option

2.3. Intel Cyclone 10 LP Embedded Memory Operation Modes

The M9K memory blocks allow you to implement fully-synchronous SRAM memory in multiple operation modes. The M9K memory blocks do not support asynchronous (unregistered) memory inputs.

Note: Violating the setup or hold time on the M9K memory block input registers may corrupt memory contents. This applies to both read and write operations.

2.3.1. Supported Memory Operation Modes

Table 5. Supported Memory Operation Modes in the M9K Embedded Memory Blocks

Memory Operation Mode	Related IP Core	Description
Single-port RAM	RAM: 1-PORT IP Core	Single-port mode supports non-simultaneous read and write operations from a single address. Use the read enable port to control the RAM output ports behavior during a write operation: • To show either the new data being written or the old data at that address, activate the read enable (`rden`) during a write operation. • To retain the previous values that are held during the most recent active read enable, perform the write operation with the read enable port deasserted.
Simple dual-port RAM	RAM: 2-PORT IP Core	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. In this memory mode, the M9K memory blocks support separate `wren` and `rden` signals. To save power, keep `rden` signal low (inactive) when not reading.

continued...

Memory Operation Mode	Related IP Core	Description
True dual-port RAM	RAM: 2-PORT IP Core	You can perform any combination of two port operations: • Two reads, two writes, or; • One read and one write at two different clock frequencies. In this memory mode, the M9K memory blocks support separate `wren` and `rden` signals. To save power, keep `rden` signal low (inactive) when not reading.
Single-port ROM	ROM: 1-PORT IP Core	Only one address port is available for read operation. You can use the memory blocks as a ROM. • Initialize the ROM contents of the memory blocks using a **.mif** or **.hex** file. • The address lines of the ROM are registered. • The outputs can be registered or unregistered. • The ROM read operation is identical to the read operation in the single-port RAM configuration.
Dual-port ROM	ROM: 2-PORT IP Core	The dual-port ROM has almost similar functional ports as single-port ROM. The difference is dual-port ROM has an additional address port for read operation. You can use the memory blocks as a ROM. • Initialize the ROM contents of the memory blocks using a **.mif** or **.hex** file. • The address lines of the ROM are registered. • The outputs can be registered or unregistered. • The ROM read operation is identical to the read operation in the single-port RAM configuration.
Shift-register	Shift Register (RAM-based) IP Core	You can use the memory blocks as a shift-register block to save logic cells and routing resources. The input data width (w), the length of the taps (m), and the number of taps (n) determine the size of a shift register ($w \times m \times n$). The size of the shift register must be less than or equal to the maximum number of memory bits (9,216 bits). The size of ($w \times n$) must be less than or equal to the maximum of width of the blocks (36 bits). You can cascade memory blocks to implement larger shift registers.
FIFO	FIFO IP Core	You can use the memory blocks as FIFO buffers. • Use the FIFO IP core in single clock FIFO (SCFIFO) mode and dual clock FIFO (DCFIFO) mode to implement single- and dual-clock FIFO buffers in your design. • Use dual clock FIFO buffers when transferring data from one clock domain to another clock domain. • The M9K memory blocks do not support simultaneous read and write from an empty FIFO buffer.

Related Information

- Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Provides more information about the supported modes, signals, and parameters of the embedded memory IP cores that support the M9K blocks in the Intel Cyclone 10 LP devices.
- Customize Read-During-Write Behavior on page 35
- Same-Port Read-During-Write Mode on page 35
- Mixed-Port Read-During-Write Mode on page 36
- Mixed-Port Read-During-Write Operation with Dual Clocks on page 37

- Memory Configurations for Dual-Port Modes on page 33
- Maximum Block Depth Configuration on page 34
- Control Clocking to Reduce Power Consumption on page 38

2.3.1.1. RAM: 1-Port IP Core References

The RAM: 1-Port IP core implements the single-port RAM memory mode.

Figure 13. RAM: 1-Port IP Core Signals with the Single Clock Option Enabled

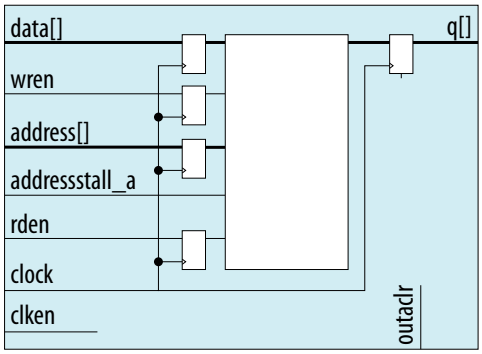
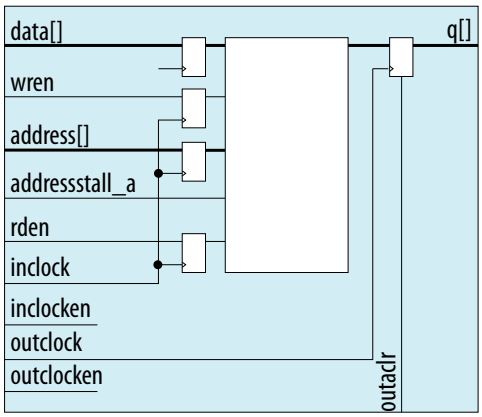


Figure 14. RAM: 1-Port IP Core Signals with the Dual Clock Option Enabled



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 IP core.
- Signals, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the signals of the embedded memory IP core.
- RAM:1-Port IP Core Parameters, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the parameters of the RAM: 1-PORT IP core.

2.3.1.2. RAM: 2-PORT IP Core References

The RAM: 2-PORT IP core implements the simple dual-port RAM and true dual-port RAM memory modes.

Figure 15. RAM: 2-Port IP Core Signals With the One Read Port and One Write Port, and Single Clock Options Enabled

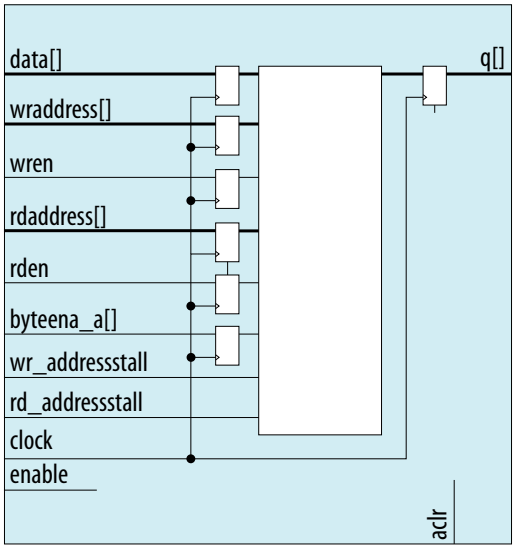


Figure 16. RAM: 2-Port IP Core Signals with the One Read Port and One Write Port, and Dual Clock: Use Separate 'Read' and 'Write' Clocks Options Enabled

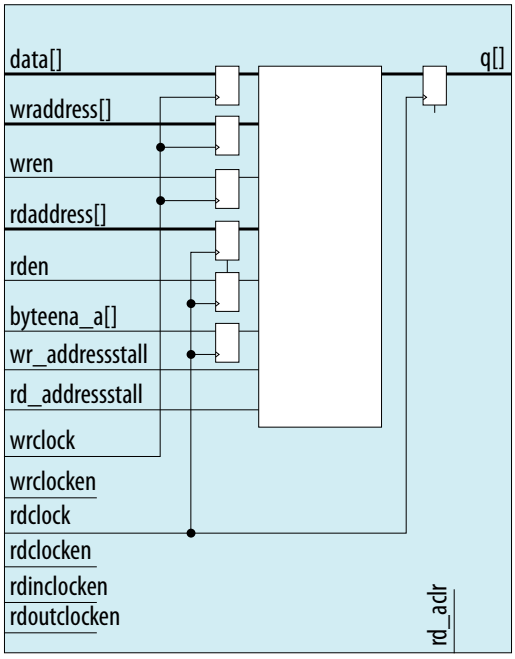


Figure 17. RAM: 2-Port IP Core Signals with the One Read Port and One Write Port, and Dual Clock: Use Separate 'Input' and 'Output' Clocks Options Enabled

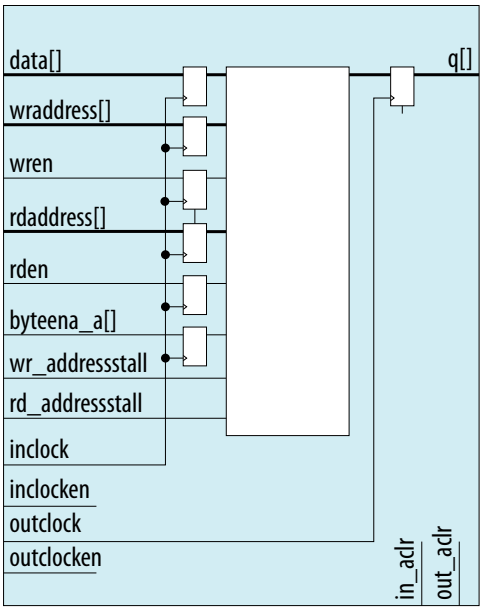


Figure 18. RAM: 2-Port IP Core Signals with the Two Read/Write Ports and Single Clock Options Enabled

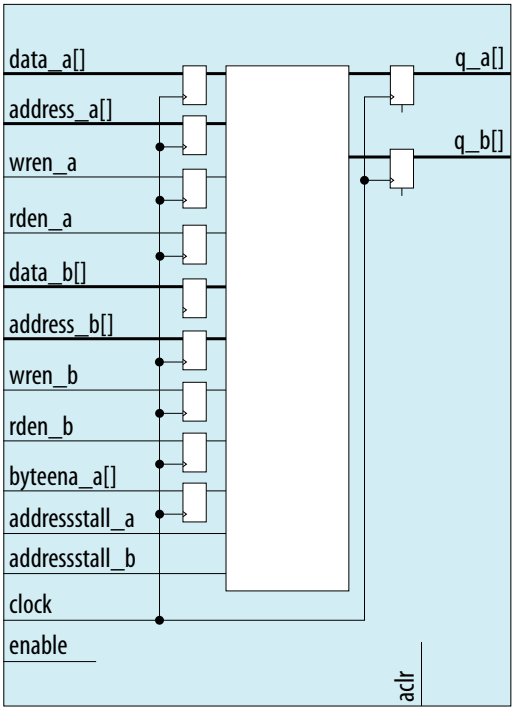


Figure 19. RAM: 2-Port IP Core Signals with the Two Read/Write Ports and Dual Clock: Use Separate 'Input' and 'Output' Clocks Options Enabled

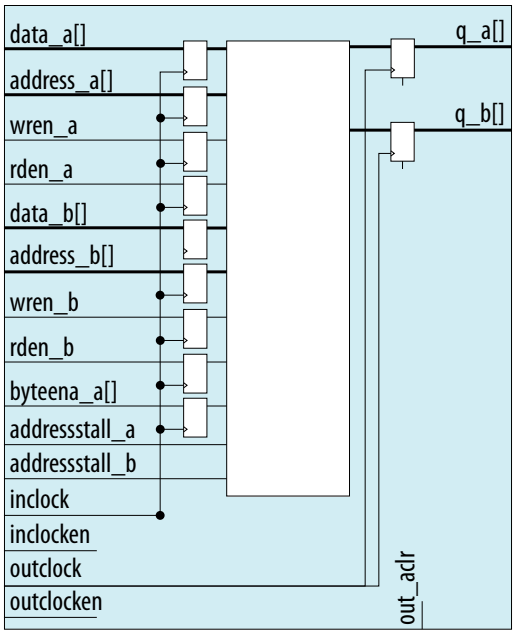
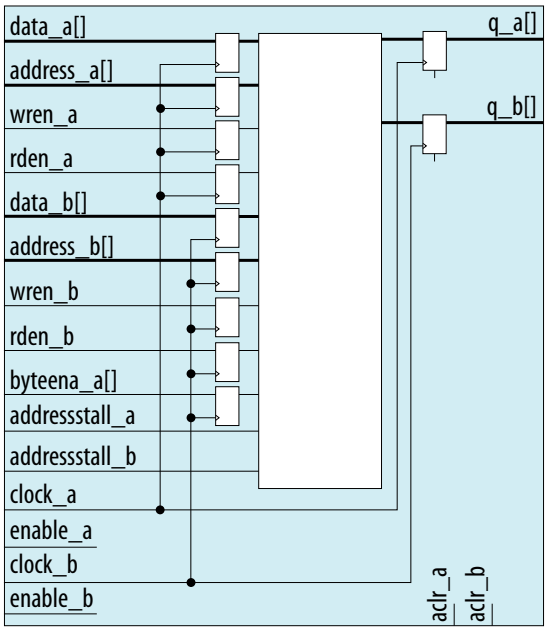


Figure 20. RAM: 2-Port IP Core Signals with the Two Read/Write Ports and Dual Clock: Use Separate for A and B Ports Options Enabled



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 IP core.

- Signals, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the signals of the embedded memory IP core.
- RAM: 2-Port IP Core Parameters, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the parameters of the RAM: 2-PORT IP core.

2.3.1.3. ROM: 1-PORT IP Core References

The ROM: 1-PORT IP core implements the single-port ROM memory mode.

Figure 21. ROM: 1-PORT IP Core Signals with the Single Clock Option Enabled

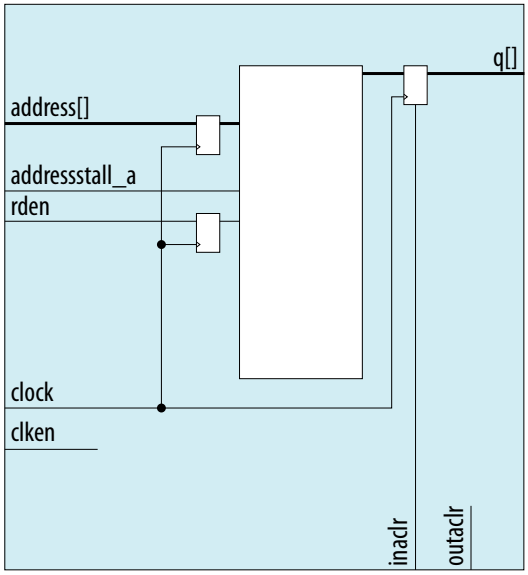
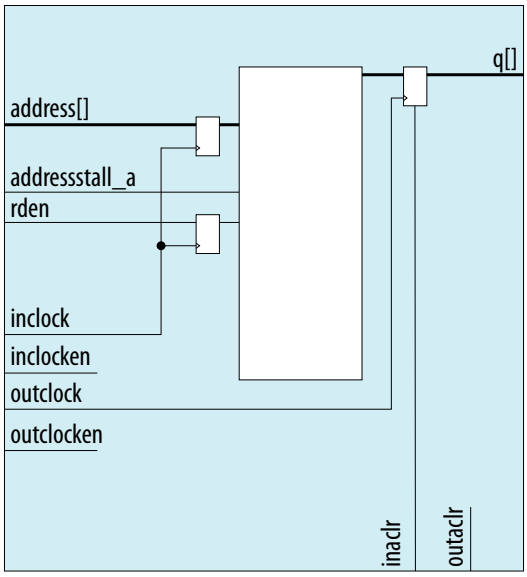


Figure 22. ROM: 1-PORT IP Core Signals with the Dual Clock: Use Separate 'Input' and 'Output' Clocks Option Enabled



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 IP core.
- Signals, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the signals of the embedded memory IP core.
- ROM: 1-PORT IP Core Parameters, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the parameters of the ROM: 1-PORT IP core.

2.3.1.4. ROM: 2-PORT IP Core References

This IP core implements the dual-port ROM memory mode. The dual-port ROM has almost similar functional ports as single-port ROM. The difference is dual-port ROM has an additional address port for read operation.

Figure 23. ROM: 2-PORT IP Core Signals with the Single Clock Option Enabled

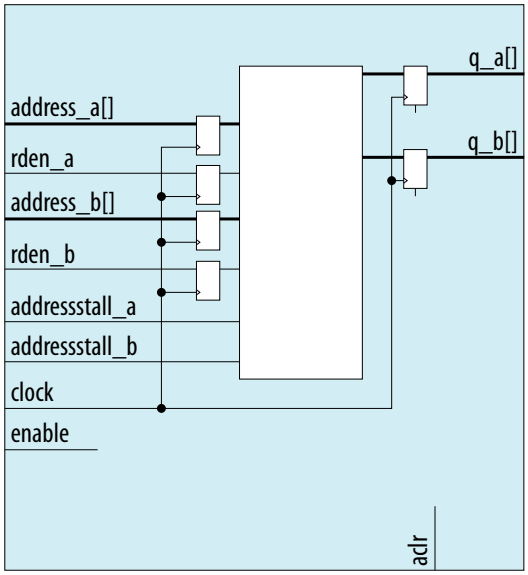


Figure 24. ROM: 2-PORT IP Core Signals with the Dual Clock: Use Separate 'Input' and 'Output' Clocks Option Enabled

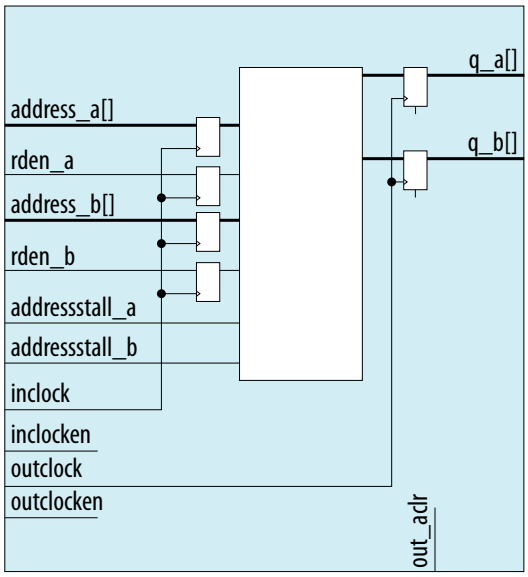
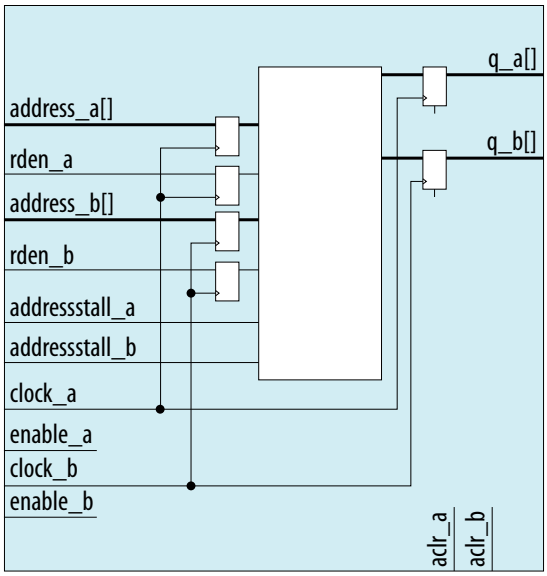


Figure 25. ROM: 2-PORT IP Core Signals with the Dual Clock: Use Separate Clocks for A and B Ports Option Enabled



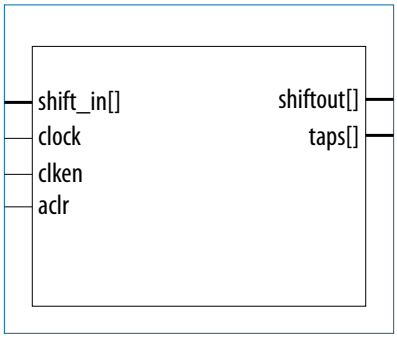
Related Information

- Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Provides more information about the embedded memory IP core.
- Signals, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the signals of the embedded memory IP core.
- ROM: 2-PORT IP Core Parameters, Embedded Memory (RAM: 1-PORT, RAM: 2-PORT, ROM: 1-PORT, and ROM: 2-PORT) User Guide
  Lists the parameters of the ROM: 2-PORT IP core.

2.3.1.5. Shift Register (RAM-based) IP Core References

The Shift Register (RAM-based) IP core contains additional features not found in a conventional shift register. You can use the memory blocks as a shift-register block to save logic cells and routing resources. You can cascade memory blocks to implement larger shift registers.

Figure 26. Shift Register (RAM-based) IP Core Signals



Related Information

RAM-Based Shift Register (ALTSHIFT_TAPS) IP Core User Guide
Provides more information about the RAM-based shift register IP core, its parameters, and signals.

2.3.1.6. FIFO IP Core References

The FIFO IP core implements the FIFO mode, enabling you to use the memory blocks as FIFO buffers.

- Use the FIFO IP core in single clock FIFO (SCFIFO) and dual clock FIFO (DCFIFO) modes to implement single- and dual-clock FIFO buffers in your design.
- Dual clock FIFO buffers are useful when transferring data from one clock domain to another clock domain.
- The M9K memory blocks do not support simultaneous read and write from an empty FIFO buffer.

Figure 27. FIFO IP Core: SCFIFO Mode Signals

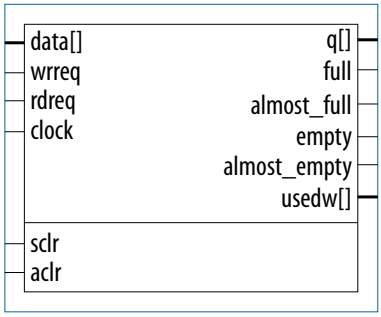
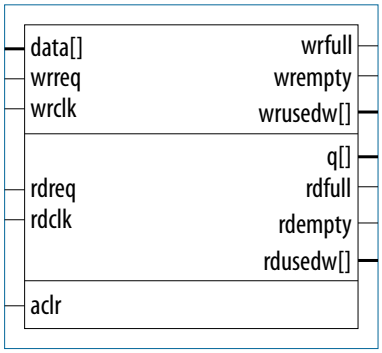


Figure 28. FIFO IP Core: DCFIFO Mode Signals



Related Information

- SCFIFO and DCFIFO IP Cores User Guide
  Provides more information about the FIFO IP cores.
- SCFIFO and DCFIFO Parameters, SCFIFO and DCFIFO IP Cores User Guide
  Lists the parameters of the FIFO IP cores.
- SCFIFO and DCFIFO Signals, SCFIFO and DCFIFO IP Cores User Guide
  Lists signals of the FIFO IP cores.

2.4. Intel Cyclone 10 LP Embedded Memory Clock Modes

Clock Mode	Description	Modes				
		True Dual-Port	Simple Dual-Port	Single-Port	ROM	FIFO
Independent Clock Mode	A separate clock is available for the following ports: • Port A—Clock A controls all registers on the port A side. • Port B—Clock B controls all registers on the port B side.	Yes	—	—	Yes	—
Input/Output Clock Mode	• M9K memory blocks can implement input or output clock mode for single-port, true dual-port, and simple dual-port memory modes. • An input clock controls all input registers to the memory block, including `data`, `address`, `byteena`, `wren`, and `rden` registers. • An output clock controls the data-output registers.	Yes	Yes	Yes	Yes	—
Read or Write Clock Mode	• M9K memory blocks support independent clock enables for both the read and write clocks. • A read clock controls the data outputs, read address, and read enable registers. • A write clock controls the data inputs, write address, and write enable registers.	—	Yes	—	—	Yes
Single-Clock Mode	A single clock, together with a clock enable, controls all registers of the memory block.	Yes	Yes	Yes	Yes	Yes

2.4.1. Asynchronous Clear in Clock Modes

In all clock modes, asynchronous clear is available only for output latches and output registers. For independent clock mode, this is applicable on port A and port B.

2.4.2. Output Read Data in Simultaneous Read and Write

If you perform a simultaneous read/write to the same address location using the read or write clock mode, the output read data is unknown. If you want the output read data to be a known value, use single-clock or input/output clock mode and then select the appropriate read-during-write behavior in the RAM: 1-PORT and RAM: 2-PORT IP cores.

2.4.3. Independent Clock Enables in Clock Modes

Table 6. Supported Clock Modes for Independent Clock Enables

Clock Mode	Description
Read/write	Supported for both the read and write clocks.
Independent	Supported for the registers of both ports.

2.5. Intel Cyclone 10 LP Embedded Memory Configurations

2.5.1. Port Width Configurations

The following equation defines the port width configuration: Memory depth (number of words) × Width of the data input bus.

- If your port width configuration (either the depth or the width) is more than the amount an internal memory block can support, additional memory blocks (of the same type) are used. For example, if you configure your M9K as 512 × 36, which exceeds the supported port width, two 512 × 18 M9Ks are used to implement your RAM.
- In addition to the supported configuration provided, you can set the memory depth to a non-power of two, but the actual memory depth allocated can vary. The variation depends on the type of resource implemented.
- If the memory is implemented in dedicated memory blocks, setting a non-power of two for the memory depth reflects the actual memory depth.
- When you implement your memory using dedicated memory blocks, refer to the Fitter report to check the actual memory depth.

2.5.2. Memory Configurations for Dual-Port Modes

Table 7. Simple Dual-Port Memory Configurations for M9K Blocks

This table lists the configuration supported simple dual-port memory configuration.

Read Port	Write Port								
	8192 × 1	4096 × 2	2048 × 4	1024 × 8	512 × 16	256 × 32	1024 × 9	512 × 18	256 × 36
8192 × 1	Yes	Yes	Yes	Yes	Yes	Yes	—	—	—
4096 × 2	Yes	Yes	Yes	Yes	Yes	Yes	—	—	—
2048 × 4	Yes	Yes	Yes	Yes	Yes	Yes	—	—	—
1024 × 8	Yes	Yes	Yes	Yes	Yes	Yes	—	—	—
512 × 16	Yes	Yes	Yes	Yes	Yes	Yes	—	—	—
256 × 32	Yes	Yes	Yes	Yes	Yes	Yes	—	—	—
1024 × 9	—	—	—	—	—	—	Yes	Yes	Yes
512 × 18	—	—	—	—	—	—	Yes	Yes	Yes
256 × 36	—	—	—	—	—	—	Yes	Yes	Yes

Table 8. True Dual-Port Memory Configurations for M9K Blocks

This table lists the configuration supported true dual-port memory configuration.

Read Port	Write Port						
	8192 × 1	4096 × 2	2048 × 4	1024 × 8	512 × 16	1024 × 9	512 × 18
8192 × 1	Yes	Yes	Yes	Yes	Yes	—	—
4096 × 2	Yes	Yes	Yes	Yes	Yes	—	—
2048 × 4	Yes	Yes	Yes	Yes	Yes	—	—
1024 × 8	Yes	Yes	Yes	Yes	Yes	—	—

continued...

Read Port	Write Port						
	8192 × 1	4096 × 2	2048 × 4	1024 × 8	512 × 16	1024 × 9	512 × 18
512 × 16	Yes	Yes	Yes	Yes	Yes	—	—
1024 × 9	—	—	—	—	—	Yes	Yes
512 × 18	—	—	—	—	—	Yes	Yes

2.5.3. Maximum Block Depth Configuration

The **Set the maximum block depth** parameter allows you to set the maximum block depth of the dedicated memory block you use. You can slice the memory block to your desired maximum block depth. For example, the capacity of an M9K block is 9,216 bits, and the default memory depth is 8K, in which each address is capable of storing 1 bit (8K × 1). If you set the maximum block depth to 512, the M9K block is sliced to a depth of 512 and each address is capable of storing up to 18 bits (512 × 18).

Use this parameter to save power usage in your devices and to reduce the total number of memory blocks used. However, this parameter might increase the number of LEs and affects the design performance.

When the RAM is sliced shallower, the dynamic power usage decreases. However, for a RAM block with a depth of 256, the power used by the extra LEs starts to outweigh the power gain achieved by shallower slices.

The maximum block depth must be in a power of two, and the valid values vary among different dedicated memory blocks.

This table lists the valid range of maximum block depth for M9K memory blocks.

Table 9. Valid Range of Maximum Block Depth for M9K Memory Blocks

Memory Block	Valid Range
M9K	256 - 8K. The maximum block depth must be in a power of two.

The IP parameter editor prompts an error message if you enter an invalid value for the maximum block depth. Intel recommends that you set the value of the **Set the maximum block depth** parameter to **Auto** if you are unsure of the appropriate maximum block depth to set or the setting is not important for your design. The **Auto** setting enables the Compiler to select the maximum block depth with the appropriate port width configuration for the type of internal memory block of your memory.

2.6. Intel Cyclone 10 LP Embedded Memory Design Consideration

There are several considerations that require your attention to ensure the success of your designs.

2.6.1. Implement External Conflict Resolution

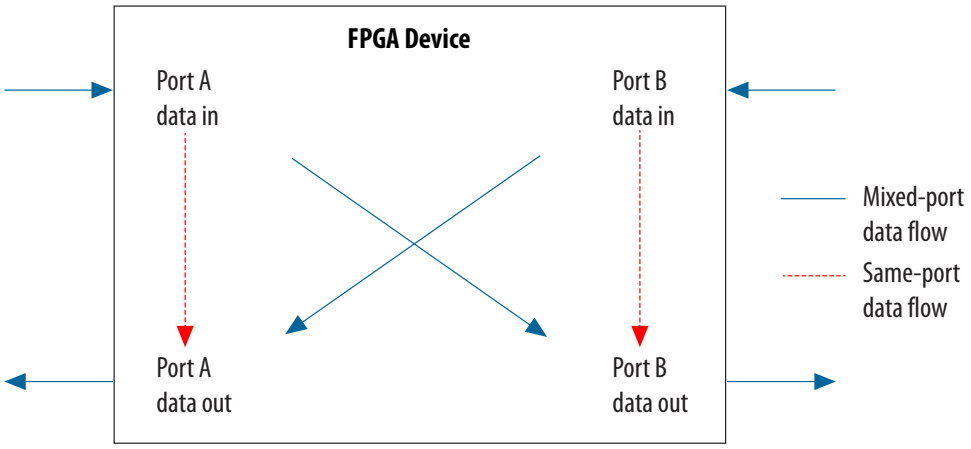
In the true dual-port RAM mode, you can perform two write operations to the same memory location. However, the memory blocks do not have internal conflict resolution circuitry.

To avoid unknown data being written to the address, implement external conflict resolution logic to the memory block.

2.6.2. Customize Read-During-Write Behavior

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

Figure 29. Difference Between the Two Types of Read-during-Write Operations —Same Port and Mixed Port.



Related Information

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

Provides more information about the supported modes, signals, and parameters of the embedded memory IP cores that support the M9K blocks in the Intel Cyclone 10 LP devices.

2.6.2.1. 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 10. 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	Description
"new data" (flow-through)	The new data is available on the rising edge of the same clock cycle on which the new data is written. When using New Data mode together with byte enable, you can control the output of the RAM: • When byte enable is high, the data written into the memory passes to the output (flow-through). • When byte enable is low, the masked-off data is not written into the memory and the old data in the memory appears on the outputs. Therefore, the output can be a combination of new and old data determined by `byteena`.
"don't care"	The RAM outputs reflect the old data at that address before the write operation proceeds.

Figure 30. Same-Port Read-During-Write: New Data Mode

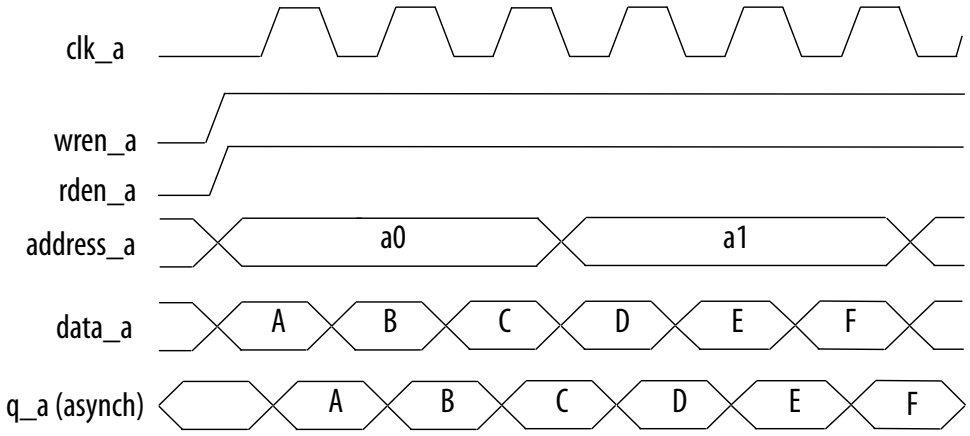
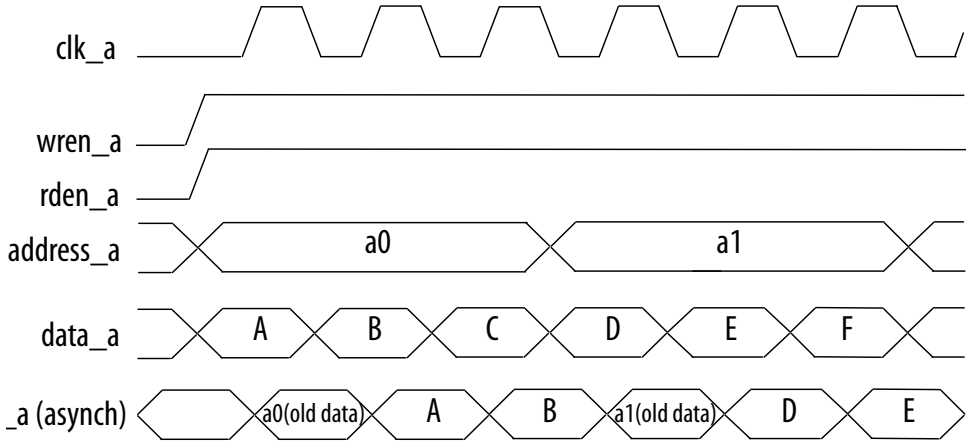


Figure 31. Same Port Read-During-Write: Old Data Mode



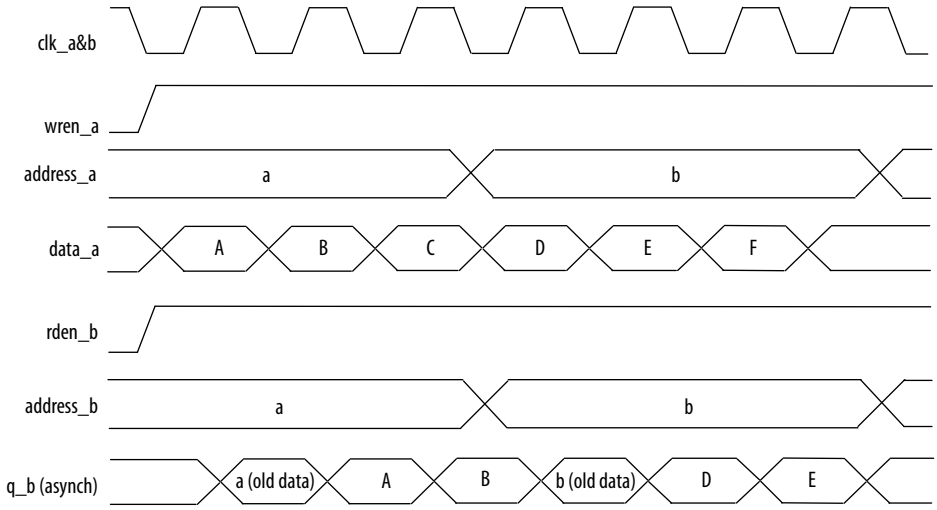
2.6.2.2. Mixed-Port Read-During-Write Mode

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

Output Mode	Description
"old data"	A read-during-write operation to different ports causes the RAM output to reflect the "old data" value at the particular address.
"don't care"	The RAM outputs "don't care" or "unknown" value.

Figure 32. Mixed-Port Read-During-Write: Old Data Mode



In Don't Care mode, the old data is replaced with "Don't Care".

2.6.2.2.1. Mixed-Port Read-During-Write Operation with Dual Clocks

For mixed-port read-during-write operation with dual clocks, the relationship between the clocks determines the output behavior of the memory.

If You...	...Then
Use the same clock for the two clocks	The output is the old data from the address location.
Use different clocks	The output is unknown during the mixed-port read-during-write operation. This unknown value may be the old or new data at the address location, depending on whether the read happens before or after the write.

2.6.3. 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.

Table 12. Initial Power-Up Values of Embedded Memory Blocks

Memory Type	Output Registers	Power Up Value
M9K	Used	Zero (cleared)
	Bypassed	Zero (cleared)

By default, the Intel Quartus Prime software initializes the RAM cells 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 when you instantiate a memory in your design. Even if a memory is preinitialized (for example, using a **.mif**), it still powers up with its output cleared. Only the subsequent read after power up outputs the preinitialized values.

2.6.4. Control Clocking to Reduce Power Consumption

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

- Use the read-enable signal to ensure that read operations occur only when necessary. If your design does not require read-during-write, you can reduce your power consumption by 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.
- 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.

2.6.5. Selecting Read-During-Write Output Choices

- Single-port RAM supports only same-port read-during-write. The clock mode must be either single clock mode or input/output clock mode.
- Simple dual-port RAM supports only mixed-port read-during-write. The clock mode must be either single clock mode, or input/output clock mode.
- True dual-port RAM supports same port read-during-write and mixed-port read-during-write:
  - For same port read-during-write, the clock mode must be either single clock mode, input/output clock mode, or independent clock mode.
  - For mixed port read-during-write, the clock mode must be either single clock mode, or input/output clock mode.

Note: If you are not concerned about the output when read-during-write occurs and want to improve performance, select **Don't Care**. Selecting **Don't Care** increases the flexibility in the type of memory block being used if you do not assign block type when you instantiate the memory block.

Table 13. Output Choices for the Same-Port and Mixed-Port Read-During-Write

Memory Block	Single-Port RAM	Simple Dual-Port RAM	True Dual-Port RAM	
	Same-Port Read-During-Write	Mixed-Port Read-During-Write	Same-Port Read-During-Write	Mixed-Port Read-During-Write
M9K	• Don’t Care • New Data • Old Data	• Old Data • Don’t Care	• New Data • Old Data	• Old Data • Don’t Care

2.7. Embedded Memory Blocks in Intel Cyclone 10 LP Devices Revision History

Date	Version	Changes
May 2017	2017.05.08	Initial release.

3. Embedded Multipliers in Intel Cyclone 10 LP Devices

The Intel Cyclone 10 LP devices include a combination of on-chip resources and external interfaces that help increase performance, reduce system cost, and lower the power consumption of digital signal processing (DSP) systems.

The Intel Cyclone 10 LP devices, either alone or as DSP device coprocessors, improves the price-to-performance ratios of DSP systems. The Intel Cyclone 10 LP devices are optimized for applications that benefit from an abundance of parallel processing resources, which include video and image processing, intermediate frequency (IF) modems used in wireless communications systems, and multi-channel communications and video systems.

3.1. Embedded Multiplier Block Overview

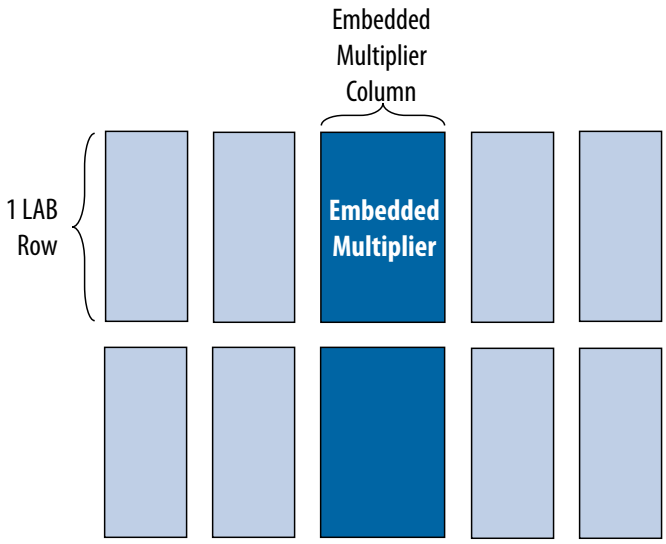
Each embedded multiplier block in Intel Cyclone 10 LP devices supports one individual 18 × 18-bit multiplier or two individual 9 × 9-bit multipliers.

For multiplications greater than 18 x 18, the Intel Quartus Prime software cascades the multiplier blocks to form wider or deeper logic structures. There are no restrictions on the data width of the multiplier but the greater the data width, the slower the multiplication process.

You can control the operation of the embedded multiplier blocks using the following options:

- Parameterize the relevant IP cores with the Quartus Prime parameter editor
- Infer the multipliers directly with VHDL or Verilog HDL

Figure 33. Embedded Multipliers Arranged in Columns with Adjacent LABS



Additionally, you can implement soft multipliers by using the M9K memory blocks as look-up tables (LUTs). The LUTs contain partial results from the multiplication of input data with coefficients that implements variable depth and width high-performance soft multipliers. Using soft multipliers increases the number of available multipliers in the device.

Related Information

- LPM_MULT (Multiplier) IP Core
  Provides more information about the LPM_MULT IP core that implements a multiplier to multiply two input data values to produce a product as an output.
- ALTMULT_ACCUM (Multiply-Accumulate) IP Core
  Provides more information about the ALTMULT_ACCUM IP core that allows you to implement a multiplier-accumulator.
- ALTMULT_ADD (Multiply-Adder) IP Core
  Provides more information about the ALTMULT_ADD IP core that allows you to implement a multiplier-adder.
- ALTMULT_COMPLEX (Complex Multiplier) IP Core
  Provides more information about the ALTMULT_COMPLEX IP core that implements the multiplication of two complex numbers in canonical and conventional modes.
- ALTMEMMULT (Memory-based Constant Coefficient Multiplier) IP Core
  Provides more information about the ALTMEMMULT IP core that you can use to create soft multipliers using the M9K memory blocks.

3.2. Multipliers Resources in Intel Cyclone 10 LP Devices

Table 14. Number of Multipliers in Intel Cyclone 10 LP Devices

Device	Embedded Multipliers		Multipliers by Type		
	9 × 9(1)	18 × 18(1)	Embedded	Soft (16 × 16)(2)	Total (Embedded + Soft)(3)
10CL006	30	15	15	30	45
10CL010	46	23	23	46	69
10CL016	112	56	56	56	112
10CL025	132	66	66	66	132
10CL040	252	126	126	126	252
10CL055	308	156	156	260	416
10CL080	488	244	244	305	549
10CL120	576	288	288	432	720

3.3. Embedded Multipliers Architecture

Each embedded multiplier consists of the following elements:

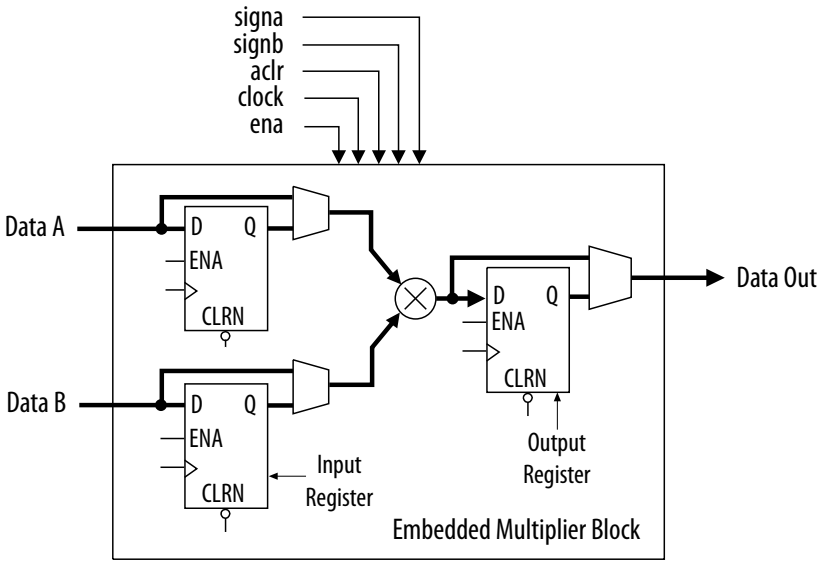
- Multiplier stage
- Input and output registers
- Input and output interfaces

(1) These columns show the number of 9 x 9 or 18 x 18 multipliers for each device. The total number of multipliers for each device is not the sum of all the multipliers.

(2) Soft multipliers are implemented in sum of multiplication mode. M9K memory blocks are configured with 18-bit data widths to support 16-bit coefficients. The sum of the coefficients requires 18-bits of resolution to account for overflow.

(3) The total number of multipliers may vary, depending on the multiplier mode you use.

Figure 34. Multiplier Block Architecture



3.3.1. Input Register

Depending on the operational mode of the multiplier, you can send each multiplier input signal into either one of the following:

- An input register
- The multiplier in 9- or 18-bit sections

Each multiplier input signal can be sent through a register independently of other input signals. For example, you can send the multiplier `Data A` signal through a register and send the `Data B` signal directly to the multiplier.

The following control signals are available to each input register in the embedded multiplier:

- Clock
- Clock enable
- Asynchronous clear

All input and output registers in a single embedded multiplier are fed by the same clock, clock enable, and asynchronous clear signals.

3.3.2. Multiplier Stage

The multiplier stage of an embedded multiplier block supports 9 × 9 or 18 × 18 multipliers and other multipliers in between these configurations. Depending on the data width or operational mode of the multiplier, a single embedded multiplier can perform one or two multiplications in parallel.

Each multiplier operand is a unique signed or unsigned number. Two signals, `signa` and `signb`, control an input of a multiplier and determine if the value is signed or unsigned. If the `signa` signal is high, the `Data A` operand is a signed number. If the `signa` signal is low, the `Data A` operand is an unsigned number.

The following table lists the sign of the multiplication results for the various operand sign representations. The results of the multiplication are signed if any one of the operands is a signed value.

Data A		Data B		Result
signa Value	Logic Level	signb Value	Logic Level	
Unsigned	Low	Unsigned	Low	Unsigned
Unsigned	Low	Signed	High	Signed
Signed	High	Unsigned	Low	Signed
Signed	High	Signed	High	Signed

Each embedded multiplier block has only one `signa` and one `signb` signal to control the sign representation of the input data to the block. If the embedded multiplier block has two 9 × 9 multipliers, the Data A input of both multipliers share the same `signa` signal, and the Data B input of both multipliers share the same `signb` signal.

You can dynamically change the `signa` and `signb` signals to modify the sign representation of the input operands at run time. You can send the `signa` and `signb` signals through a dedicated input register. The multiplier offers full precision, regardless of the sign representation.

When the `signa` and `signb` signals are unused, the Intel Quartus Prime software sets the multiplier to perform unsigned multiplication by default.

3.3.3. Output Register

You can register the embedded multiplier output using output registers in either 18- or 36-bit sections. This depends on the operational mode of the multiplier. The following control signals are available for each output register in the embedded multiplier:

- Clock
- Clock enable
- Asynchronous clear

All input and output registers in a single embedded multiplier are fed by the same clock, clock enable, and asynchronous clear signals.

3.4. Embedded Multipliers Operational Modes

You can use an embedded multiplier block in one of two operational modes, depending on the application needs:

- One 18-bit x 18-bit multiplier
- Up to two 9-bit x 9-bit independent multipliers

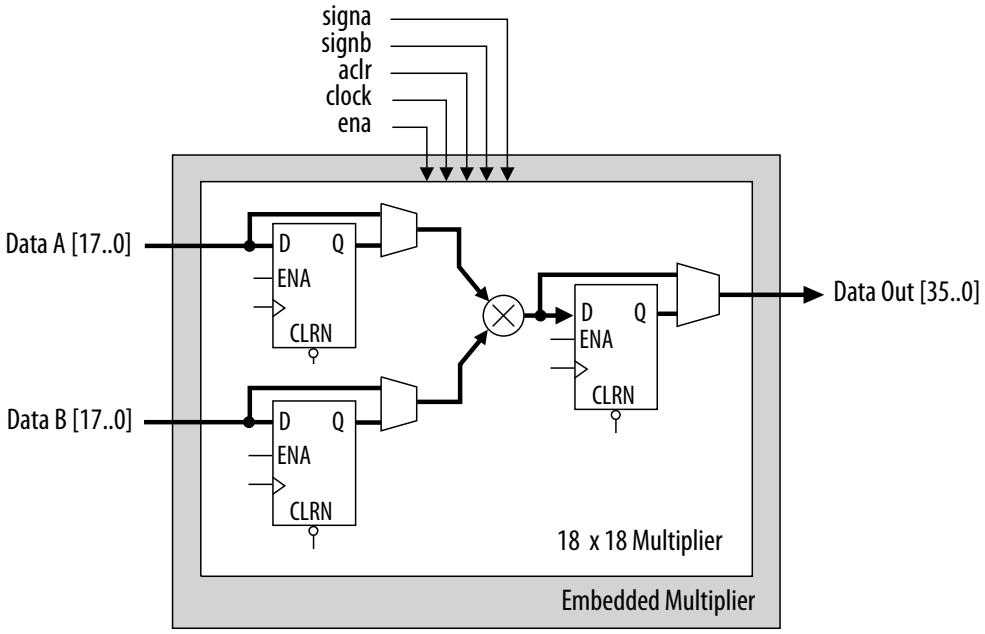
You can also use embedded multipliers of the Intel Cyclone 10 LP devices to implement multiplier adder and multiplier accumulator functions. The multiplier portion of the function is implemented using embedded multipliers. The adder or accumulator function is implemented in logic elements (LEs).

3.4.1. 18-Bit Multipliers

You can configure each embedded multiplier to support a single 18 x 18 multiplier for input widths of 10 to 18 bits.

The following figure shows the embedded multiplier configured to support an 18-bit multiplier.

Figure 35. 18-Bit Multiplier Mode



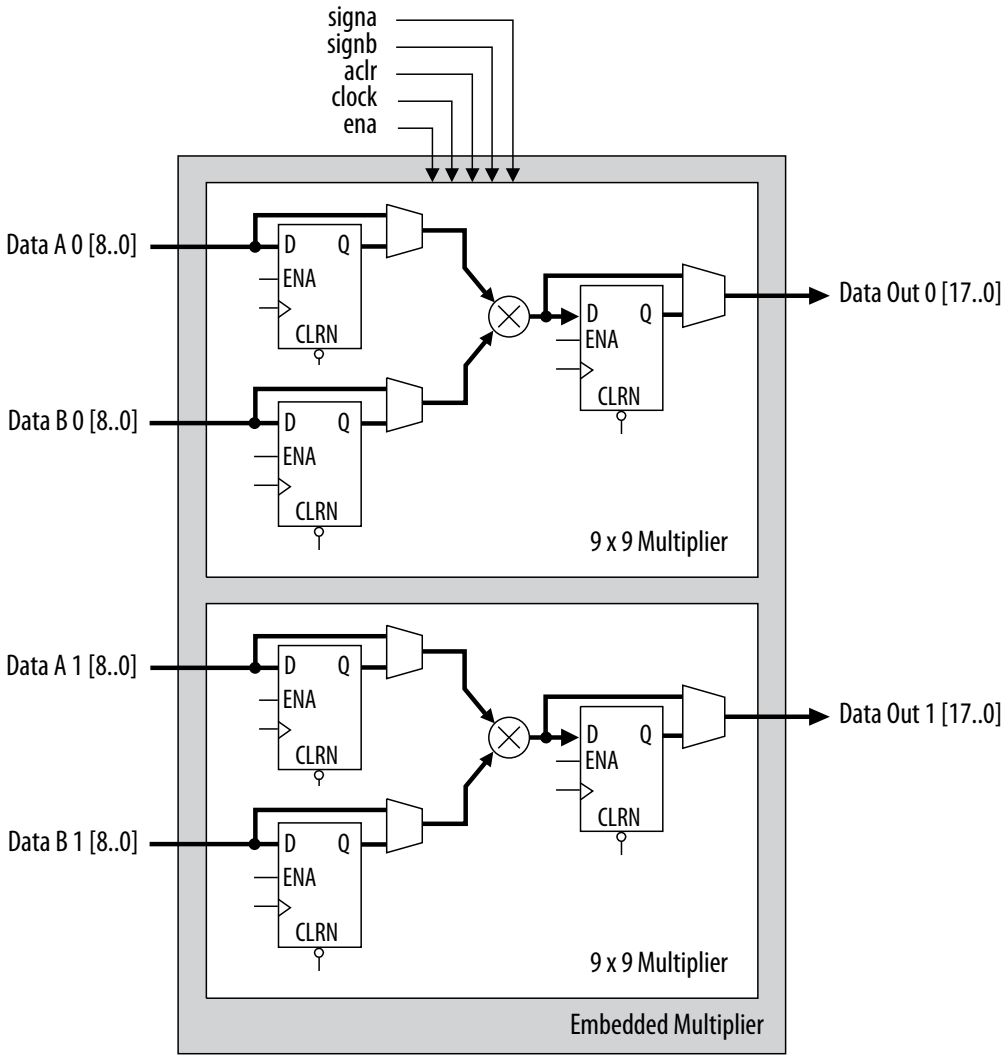
All 18-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both. Also, you can dynamically change the `signa` and `signb` signals and send these signals through dedicated input registers.

3.4.2. 9-Bit Multipliers

You can configure each embedded multiplier to support two 9 × 9 independent multipliers for input widths of up to 9 bits.

The following figure shows the embedded multiplier configured to support two 9-bit multipliers.

Figure 36. 9-Bit Multiplier Mode



All 9-bit multiplier inputs and results are independently sent through registers. The multiplier inputs can accept signed integers, unsigned integers, or a combination of both.

Each embedded multiplier block has only one `signa` and one `signb` signal to control the sign representation of the input data to the block. If the embedded multiplier block has two 9 × 9 multipliers the following applies:

- The `Data A` input of both multipliers share the same `signa` signal
- The `Data B` input of both multipliers share the same `signb` signal

3.5. Embedded Multipliers in Intel Cyclone 10 LP Devices Revision History

Date	Version	Changes
May 2017	2017.05.08	Initial release.

4. Clock Networks and PLLs in Intel Cyclone 10 LP Devices

This chapter describes the hierarchical clock networks and phase-locked loops (PLLs) with advanced features in the Intel Cyclone 10 LP devices. It includes details about the ability to reconfigure the PLL counter clock frequency and phase shift in real time, allowing you to sweep PLL output frequencies and dynamically adjust the output clock phase shift.

The Intel Quartus Prime software enables the PLLs and their features without external devices.

4.1. Clock Networks

The Intel Cyclone 10 LP device provides up to 16 dedicated clock pins (CLK[15..0]) that can drive up to 20 global clocks (GCLKs). Intel Cyclone 10 LP devices support three dedicated clock pins on the left side and four dedicated clock pins on the top, right, and bottom sides of the device except 10CL006 and 10CL010 devices. 10CL006 and 10CL010 devices only support three dedicated clock pins on the left side and four dedicated clock pins on the right side of the device.

Related Information

Intel Cyclone 10 LP Maximum Resources, Intel Cyclone 10 LP Device Overview
Provides more information about the number of GCLK networks in each device density.

4.1.1. GCLK Network

GCLKs drive throughout the entire device, feeding all device quadrants. All resources in the device (I/O elements, logic array blocks (LABs), dedicated multiplier blocks, and M9K memory blocks) can use GCLKs as clock sources. Use these clock network resources for control signals, such as clock enables and clears fed by an external pin. Internal logic can also drive GCLKs for internally generated GCLKs and asynchronous clears, clock enables, or other control signals with high fan-out.

4.1.2. GCLK Network Sources

Table 15. Intel Cyclone 10 LP Clock Sources Connectivity to the GCLK Networks

10CL006 and 10CL010 devices only have GCLK networks 0 to 9.

GCLK Network Clock Sources	GCLK Networks
CLK0/DIFFCLK_0p	GCLK[0,2,4]
CLK1/DIFFCLK_0n	GCLK[1,2]
CLK2/DIFFCLK_1p	GCLK[1,3,4]

continued...

GCLK Network Clock Sources	GCLK Networks
CLK3/DIFFCLK_1n	GCLK[0,3]
CLK4/DIFFCLK_2p	GCLK[5,7,9]
CLK5/DIFFCLK_2n	GCLK[6,7]
CLK6/DIFFCLK_3p	GCLK[6,8,9]
CLK7/DIFFCLK_3n	GCLK[5,8]
CLK8/DIFFCLK_5n (4)	GCLK[10,12,14]
CLK9/DIFFCLK_5p (4)	GCLK[11,12]
CLK10/DIFFCLK_4n (4)	GCLK[11,13,14]
CLK11/DIFFCLK_4p (4)	GCLK[10,13]
CLK12/DIFFCLK_7n (4)	GCLK[15,17,19]
CLK13/DIFFCLK_7p (4)	GCLK[16,17]
CLK14/DIFFCLK_6n (4)	GCLK[16,18,19]
CLK15/DIFFCLK_6p (4)	GCLK[15,18]
PLL_1_C0 (5)	GCLK[0,3]
PLL_1_C1 (5)	GCLK[1,4]
PLL_1_C2 (5)	GCLK[0,2]
PLL_1_C3 (5)	GCLK[1,3]
PLL_1_C4 (5)	GCLK[2,4]
PLL_2_C0 (5)	GCLK[5,8]
PLL_2_C1 (5)	GCLK[6,9]
PLL_2_C2 (5)	GCLK[5,7]
PLL_2_C3 (5)	GCLK[6,8]
PLL_2_C4 (5)	GCLK[7,9]
PLL_3_C0	GCLK[10,13]
PLL_3_C1	GCLK[11,14]
PLL_3_C2	GCLK[10,12]
PLL_3_C3	GCLK[11,13]
PLL_3_C4	GCLK[12,14]
PLL_4_C0	GCLK[15,18]
PLL_4_C1	GCLK[16,19]
PLL_4_C2	GCLK[15,17]

continued...

(4) These pins apply to all Intel Cyclone 10 LP devices except10CL006 and 10CL010 devices.

(5) 10CL006 and 10CL010 devices only have PLL_1 and PLL_2.

GCLK Network Clock Sources	GCLK Networks
PLL_4_C3	GCLK[16,18]
PLL_4_C4	GCLK[17,19]
DPCLK0	GCLK[0]
DPCLK1	GCLK[1]
DPCLK7 (6), CDPCLK0, or CDPCLK7 (4) (7)	GCLK[2]
DPCLK2 (6), CDPCLK1, or CDPCLK2 (4) (7)	GCLK[3,4]
DPCLK5 (6), DPCLK7 (4)	GCLK[5]
DPCLK4 (6), DPCLK6 (4)	GCLK[6]
DPCLK6 (6), CDPCLK5, or CDPCLK6 (4) (7)	GCLK[7]
DPCLK3 (6), CDPCLK4, or CDPCLK3 (4) (7)	GCLK[8,9]
DPCLK8	GCLK[10]
DPCLK11	GCLK[11]
DPCLK9	GCLK[12]
DPCLK10	GCLK[13,14]
DPCLK5	GCLK[15]
DPCLK2	GCLK[16]
DPCLK4	GCLK[17]
DPCLK3	GCLK[18,19]

Related Information

Clock and PLL Pins, Intel Cyclone 10 LP Device Family Pin Connection Guidelines
Provides more information about the clock and PLL pins connections.

4.1.3. Clock Control Block

The clock control block drives the GCLKs. Clock control blocks are located on each side of the device, close to the dedicated clock input pins. GCLKs are optimized for minimum clock skew and delay.

(6) This pin applies only to 10CL006 and 10CL010 devices.

(7) Only one of the two CDPCLK pins can feed the clock control block. You can use the other pin as a regular I/O pin.

Table 16. Clock Control Block Inputs

Input	Description
Dedicated clock input pins	Dedicated clock input pins can drive clocks or global signals, such as synchronous and asynchronous clears, presets, or clock enables onto given GCLKs.
Dual-purpose clock (DPCLK and CDPCLK) I/O input	DPCLK and CDPCLK pins are bidirectional dual function pins that are used for high fan-out control signals, such as protocol signals, TRDY and IRDY signals for PCI via the GCLK. Clock control blocks that have inputs driven by dual-purpose clock I/O pins cannot drive PLL inputs.
PLL outputs	PLL counter outputs can drive the GCLK.
Internal logic	You can drive the GCLK through logic array routing to enable the internal logic elements (LEs) to drive a high fan-out, low-skew signal path. Clock control blocks that have inputs driven by internal logic cannot drive PLL inputs.

In Intel Cyclone 10 LP devices, dedicated clock input pins, PLL counter outputs, dual-purpose clock I/O inputs, and internal logic can all feed the clock control block for each GCLK. The output from the clock control block in turn feeds the corresponding GCLK. The GCLK can drive the PLL input if the clock control block inputs are outputs of another PLL or dedicated clock input pins.

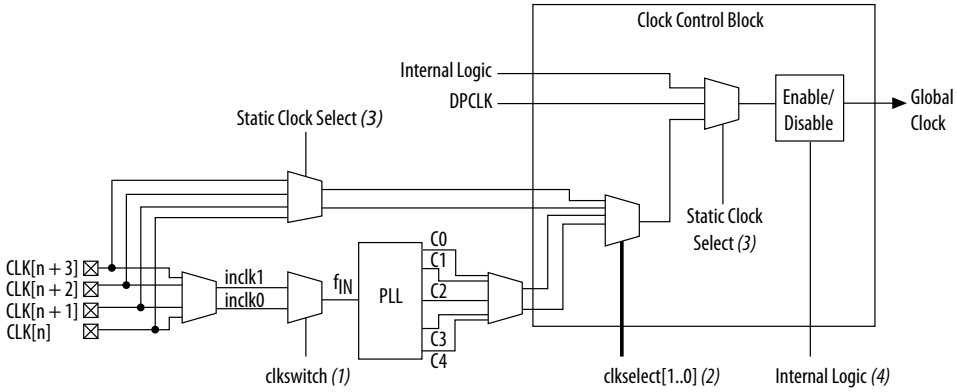
The maximum number of clock control blocks per Intel Cyclone 10 LP device is 20.

The control block has two functions:

- Dynamic GCLK clock source selection (not applicable for DPCLK, CDPCLK, and internal logic input)
- GCLK network power down (dynamic enable and disable)

Figure 37. Clock Control Block

Each PLL generates five clock outputs through the c[4..0] counters. Two of these clocks can drive the GCLK through a clock control block.



Notes:

(1) The clkswitch signal can either be set through the configuration file or dynamically set when using the manual PLL switchover feature. The output of the multiplexer is the input clock (fIN) for the PLL.

(2) The clkselect[1..0] signals are fed by internal logic and are used to dynamically select the clock source for the GCLK when the device is in user mode.

(3) The static clock select signals are set in the configuration file. Therefore, dynamic control when the device is in user mode is not feasible.

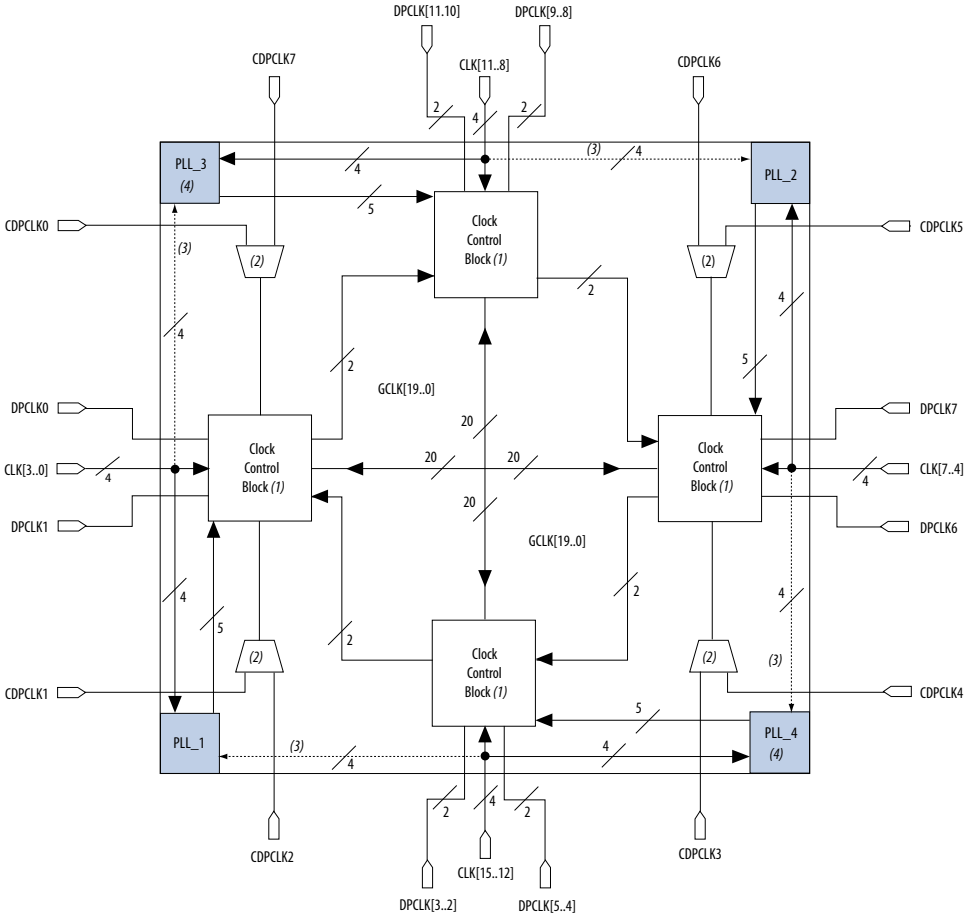
(4) You can use internal logic to enable or disable the GCLK in user mode.

Related Information

- GCLK Network Clock Source Generation on page 51
  Shows the clock control block locations.
- Clock Control Block (ALTCLKCTRL) IP Core User Guide
  Provides more information about the clock control block in the Intel Quartus Prime software.

4.1.4. GCLK Network Clock Source Generation

Figure 38. Clock Networks and Clock Control Block Locations in Intel Cyclone 10 LP Devices



Notes:
(1) There are five clock control blocks on each side.
(2) Only one of the corner CDPCLK pins in each corner can feed the clock control block at a time. You can use the other CDPCLK pins as general-purpose I/O (GPIO) pins.
(3) Dedicated clock pins can feed into this PLL. However, these paths are not fully compensated.
(4) PLL_3 and PLL_4 are not available in 10CL006 and 10CL010 devices.

The inputs to the five clock control blocks on each side of the Intel Cyclone 10 LP device must be chosen from among the following clock sources:

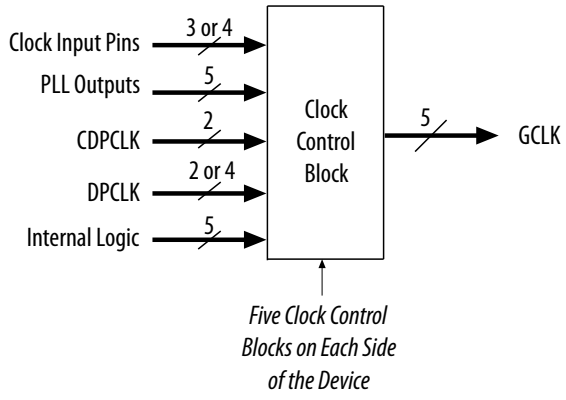
- Three or four clock input pins, depending on the specific device
- Five PLL counter outputs
- Two DPCLK pins and two CDPCLK pins from both the left and right sides and four DPCLK pins from both the top and bottom
- Five signals from internal logic

From the clock sources listed above, only two clock input pins, two PLL clock outputs, one DPCLK or CDPCLK pin, and one source from internal logic can drive into any given clock control block.

Out of these six inputs to any clock control block, the two clock input pins and two PLL outputs are dynamically selected to feed a GCLK. The clock control block supports static selection of the signal from internal logic.

Figure 39. Clock Control Blocks on Each Side of Intel Cyclone 10 LP Device

The left and right sides of the device have two DPCLK pins; the top and bottom of the device have four DPCLK pins.



Related Information

Clock Control Block on page 49

4.1.5. GCLK Network Power Down

You can disable an Intel Cyclone 10 LP device's GCLK (power down) using both static and dynamic approaches. In the static approach, configuration bits are set in the configuration file generated by the Intel Quartus Prime software, which automatically disables unused GCLKs. The dynamic clock enable or disable feature allows internal logic to control clock enable or disable the GCLKs in Intel Cyclone 10 LP devices.

When a clock network is disabled, all the logic fed by the clock network is in an off-state, thereby reducing the overall power consumption of the device. This function is independent of the PLL and is applied directly on the clock network.

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

Related Information

- Clock Control Block on page 49
- Clock Control Block (ALTCLKCTRL) IP Core User Guide
  Provides more information about the clock control block in the Intel Quartus Prime software.

4.1.6. Clock Enable Signals

Intel Cyclone 10 LP devices support `clkena` signals at the GCLK network level. This allows you to gate-off the clock even when a PLL is used. Upon re-enabling the output clock, the PLL does not need a resynchronization or re-lock period because the circuit gates off the clock at the clock network level. In addition, the PLL can remain locked independent of the `clkena` signals because the loop-related counters are not affected.

Figure 40. clkena Implementation

This figure shows the implementation of the `clkena` signal with a single register. The `clkena` circuitry controlling the output `C0` of the PLL to an output pin is implemented with two registers instead of a single register.

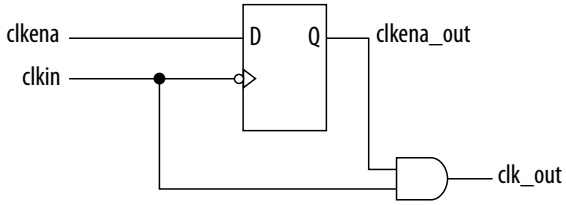
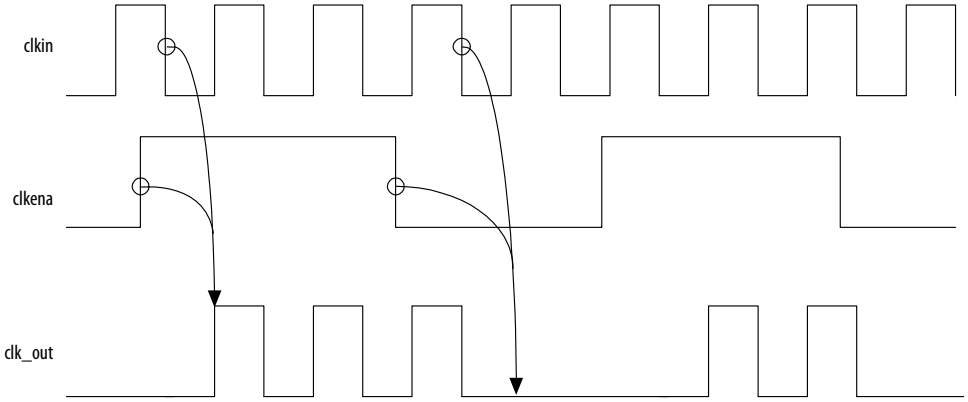


Figure 41. Example Waveform of clkena Implementation with Output Enable

This figure shows the waveform example for a clock output enable. The `clkena` signal is sampled on the falling edge of the clock (`clkin`). This feature is useful for applications that require low power or sleep mode.



The `clkena` signal can also disable clock outputs if the system is not tolerant to frequency overshoot during PLL resynchronization.

Intel recommends using the clkena signals when switching the clock source to the PLLs or the GCLK. The recommended sequence is:

1. Disable the primary output clock by de-asserting the clkena signal.
2. Switch to the secondary clock using the dynamic select signals of the clock control block.
3. Allow some clock cycles of the secondary clock to pass before reasserting the clkena signal. The exact number of clock cycles you must wait before enabling the secondary clock is design-dependent. You can build custom logic to ensure glitch-free transition when switching between different clock sources.

4.2. PLLs in Intel Cyclone 10 LP Devices

Intel Cyclone 10 LP devices only have the general purpose PLLs. The general purpose PLLs are used for general-purpose applications in the FPGA fabric and periphery such as external memory interfaces.

Intel Cyclone 10 LP devices have up to four general purpose PLLs that provide robust clock management and synthesis for device clock management, external system clock management, and high-speed I/O interfaces.

The general I/O pins cannot drive the PLL clock input pins.

Related Information

Intel Cyclone 10 LP Package Plan, Intel Cyclone 10 LP Device Overview
Provides more information about the number of general purpose PLLs in each device density.

4.2.1. PLL Features

Table 17. Intel Cyclone 10 LP PLL Features

Feature	Support
C output counters	5
M, N, C counter sizes	1 to 512 (8)
Dedicated clock outputs	1 single-ended or 1 differential
Dedicated clock input pins	4 single-ended or 2 differential
Spread-spectrum input clock tracking	Yes (9)
PLL cascading	Through GCLK
Source synchronous compensation	Yes
No compensation mode	Yes
Normal compensation	Yes
Zero-delay buffer compensation	Yes

continued...

(8) C counters range from 1 through 512 if the output clock uses a 50% duty cycle. For any output clocks using a non-50% duty cycle, the post-scale counters range from 1 through 256.

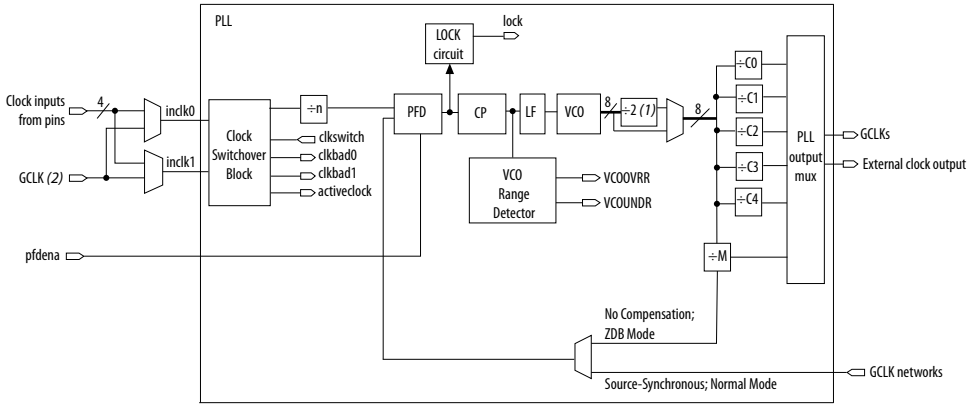
(9) Only applicable if the input clock jitter is in the input jitter tolerance specifications.

Feature	Support
Phase shift resolution	Down to 96 ps increments (10)
Programmable duty cycle	Yes
Output counter cascading	Yes
Input clock switchover	Yes
User mode reconfiguration	Yes
Loss of lock detection	Yes

4.2.2. PLL Architecture

Figure 42. Intel Cyclone 10 LP PLL Block Diagram

Each clock source can come from any of the four clock pins located on the same side of the device as the PLL.



Notes:
(1) This is the VCO post-scale counter K.
(2) This input port is fed by a pin-driven dedicated GCLK, or through a clock control block if the clock control block is fed by an output from another PLL or a pin-driven dedicated GCLK. An internally-generated global signal cannot drive the PLL.

The VCO post-scale counter, K, is used to divide the supported VCO range by two. The VCO frequency reported by the Intel Quartus Prime software in the PLL summary section of the compilation report takes into consideration the VCO post-scale counter value. Therefore, if the VCO post-scale counter has a value of 2, the frequency reported is lower than the fVCO specification specified in the Intel Cyclone 10 LP Device Datasheet.

Related Information

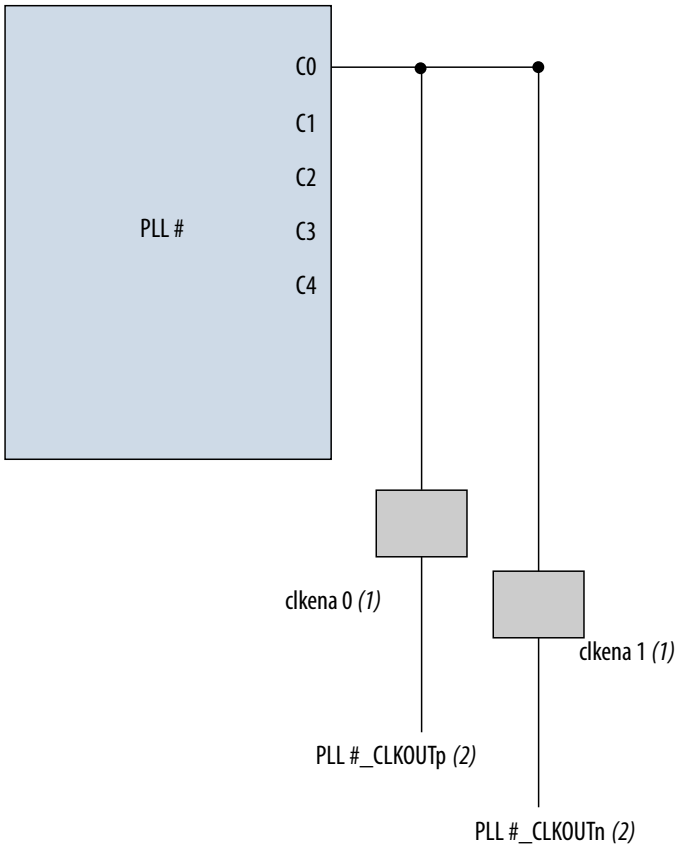
PLL Specifications, Intel Cyclone 10 LP Device Datasheet
Provides the f_{VCO} specification.

(10) The smallest phase shift is determined by the VCO period divided by eight. For degree increments, the Intel Cyclone 10 LP devices can shift all output frequencies in increments of at least 45°. Smaller degree increments are possible depending on the frequency and divide parameters.

4.2.3. External Clock Outputs

Each PLL of Intel Cyclone 10 LP devices supports one single-ended clock output or one differential clock output. Only the C0 output counter can feed the dedicated external clock outputs without going through the GCLK. Other output counters can feed other I/O pins through the GCLK.

Figure 43. External Clock Outputs for PLLs



Notes:

(1) These external clock enable signals are available only when using the ALTCLKCTRL IP core.

(2) PLL#_CLKOUTp and PLL#_CLKOUTn pins are dual-purpose I/O pins that you can use as one single-ended clock output or one differential clock output. When using both pins as single-ended I/Os, one of them can be the clock output while the other pin is configured as a regular user I/O.

Each pin of a differential output pair is 180° out of phase. The Intel Quartus Prime software places the NOT gate in your design into the I/O element to implement 180° phase with respect to the other pin in the pair. The clock output pin pairs support the same I/O standards as standard output pins.

Intel Cyclone 10 LP PLLs can drive out to any regular I/O pin through the GCLK. You can also use the external clock output pins as GPIO pins if external PLL clocking is not required.

Related Information

I/O Standards Voltage and Pin Support, Intel Cyclone 10 LP Core Fabric and General Purpose I/Os Handbook
Lists the I/O standards supported by the PLL clock input and output.

4.2.4. Clock Feedback Modes

Intel Cyclone 10 LP PLLs support up to five different clock feedback modes. Each mode allows clock multiplication and division, phase shifting, and programmable duty cycle.

Input and output delays are fully compensated by the PLL only if you are using the dedicated clock input pins associated with a given PLL as the clock sources.

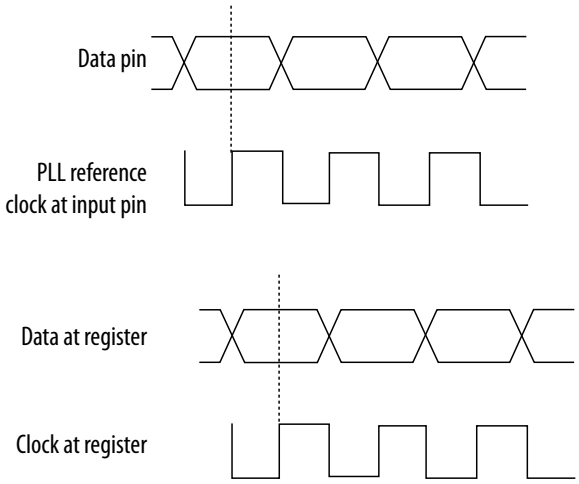
When driving the PLL using the GCLK network, the input and output delays may not be fully compensated in the Intel Quartus Prime software.

4.2.4.1. Source Synchronous Mode

If the data and clock arrive at the same time at the input pins, the phase relationship between the data and clock remains the same at the data and clock ports of any I/O element input register.

You can use this mode for source synchronous data transfers. Data and clock signals at the I/O element experience similar buffer delays as long as both signals use the same I/O standard.

Figure 44. Example of Phase Relationship Between Clock and Data in Source Synchronous Mode



Source synchronous mode compensates for clock network delay, including any difference in delay between the following two paths:

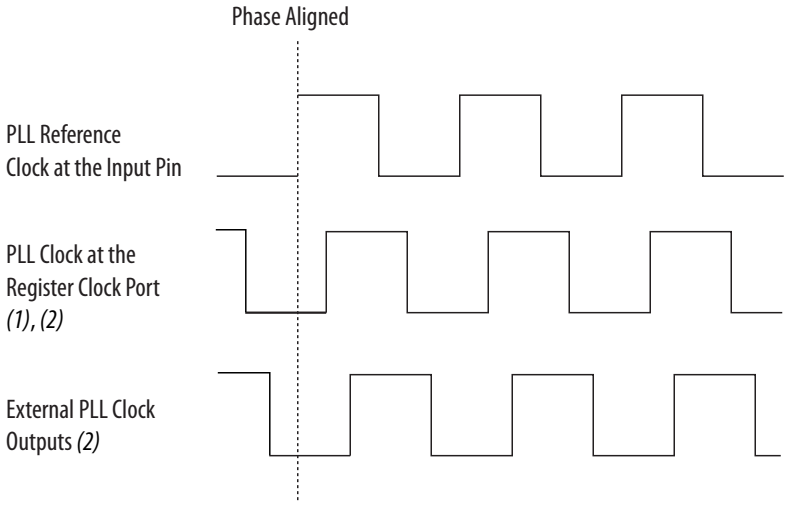
- Data pin to I/O element register input
- Clock input pin to the PLL phase frequency detector (PFD) input

For all data pins clocked by a source synchronous mode PLL, set the input pin to the register delay chain in the I/O element to zero in the Intel Quartus Prime software. All data pins must use the **PLL COMPENSATED logic** option in the Intel Quartus Prime software.

4.2.4.2. No Compensation Mode

In no compensation mode, the PLL does not compensate for any clock networks. This mode provides better jitter performance because clock feedback into the PFD does not pass through as much circuitry. Both the PLL internal and external clock outputs are phase-shifted with respect to the PLL clock input.

Figure 45. Example of Phase Relationship Between the PLL Clocks in No Compensation Mode

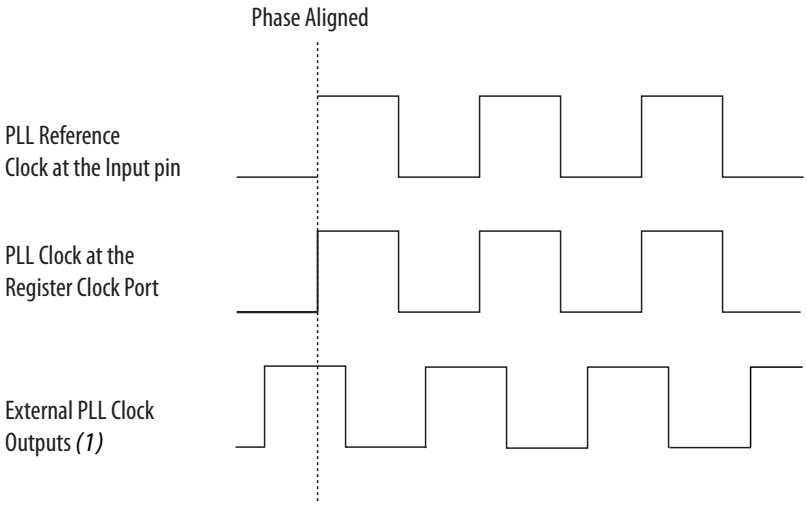


Notes:
(1) Internal clocks fed by the PLL are phase-aligned to each other.
(2) The PLL clock outputs can lead or lag the PLL input clocks.

4.2.4.3. Normal Mode

An internal clock in normal mode is phase-aligned to the input clock pin. The external clock output pin has a phase delay relative to the clock input pin if connected in this mode. The Intel Quartus Prime software timing analyzer reports any phase difference between the two. In normal mode, the PLL fully compensates the delay introduced by the GCLK network.

Figure 46. Example of Phase Relationship Between the PLL Clocks in Normal Compensation Mode

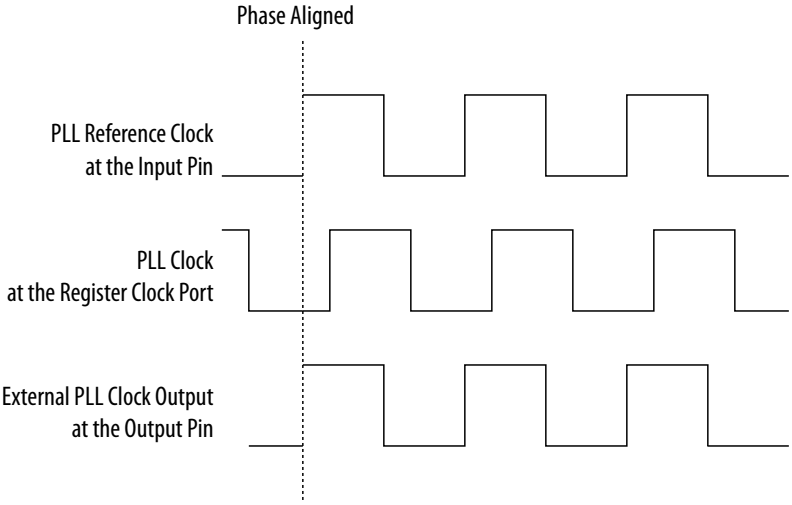


Note:
(1) The external clock output can lead or lag the PLL internal clock signals.

4.2.4.4. Zero-Delay Buffer Mode

In zero-delay buffer (ZDB) mode, the external clock output pin is phase-aligned with the clock input pin for zero delay through the device. When using this mode, use the same I/O standard for the input clock and output clocks to ensure clock alignment at the input and output pins.

Figure 47. Example of Phase Relationship Between the PLL Clocks in ZDB Mode



4.2.5. Clock Multiplication and Division

Each Intel Cyclone 10 LP PLL provides clock synthesis for PLL output ports using M/(N × post-scale counter) scaling factors. The input clock is divided by a pre-scale factor, N, and is then multiplied by the M feedback factor. The control loop drives the VCO to

match f_{IN} (M/N). Each output port has a unique post-scale counter that divides down the high-frequency VCO. For multiple PLL outputs with different frequencies, the VCO value is the least common multiple of the output frequencies that meets its frequency specifications. For example, if output frequencies required from one PLL are 33 and 66 MHz, the Intel Quartus Prime software sets the VCO to 660 MHz (the least common multiple of 33 and 66 MHz in the VCO range). Then, the post-scale counters scale down the VCO frequency for each output port.

There is one pre-scale counter, N, and one multiply counter, M, per PLL, with a range of 1 to 512 for both M and N. The N counter does not use duty cycle control because the purpose of this counter is only to calculate frequency division. There are five generic post-scale counters per PLL that can feed GCLKs or external clock outputs. These post-scale counters range from 1 to 512 with a 50% duty cycle setting. The post-scale counters range from 1 to 256 with any non-50% duty cycle setting. The sum of the high/low count values chosen for a design selects the divide value for a given counter.

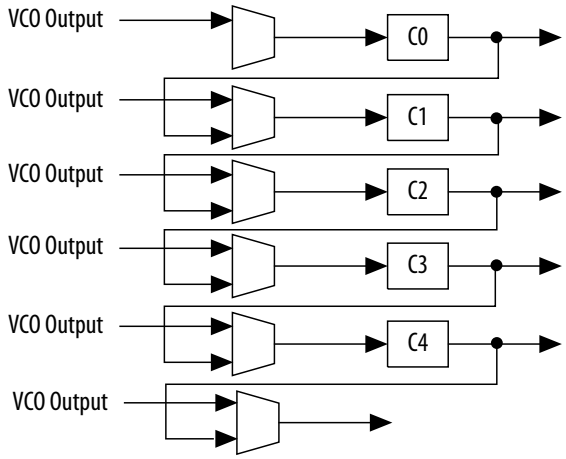
The Intel Quartus Prime software automatically chooses the appropriate scaling factors according to the input frequency, multiplication, and division values entered into the ALTPLL IP core.

Phase alignment between output counters is determined using the t_{PLL_PSERR} specification.

4.2.6. Post-Scale Counter Cascading

The Intel Cyclone 10 LP PLLs support post-scale counter cascading to create counters larger than 512. This is implemented by feeding the output of one C counter into the input of the next C counter.

Figure 48. Counter-to-Counter Cascading



When cascading counters to implement a larger division of the high-frequency VCO clock, the cascaded counters behave as one counter with the product of the individual counter settings.

For example, if C0 = 4 and C1 = 2, the cascaded value is C0 × C1 = 8.

Post-scale counter cascading is automatically set by the Intel Quartus Prime software in the configuration file. Post-scale counter cascading cannot be performed using PLL reconfiguration.

4.2.7. Programmable Duty Cycle

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

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

4.2.8. PLL Control Signals

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

Related Information

Advanced Control Signals (`pllena`, `areset`, `pfdena`), ALTPLL (Phase-Locked Loop) IP Core User Guide
Provides more information about the PLL control signals.

4.2.9. Clock Switchover

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

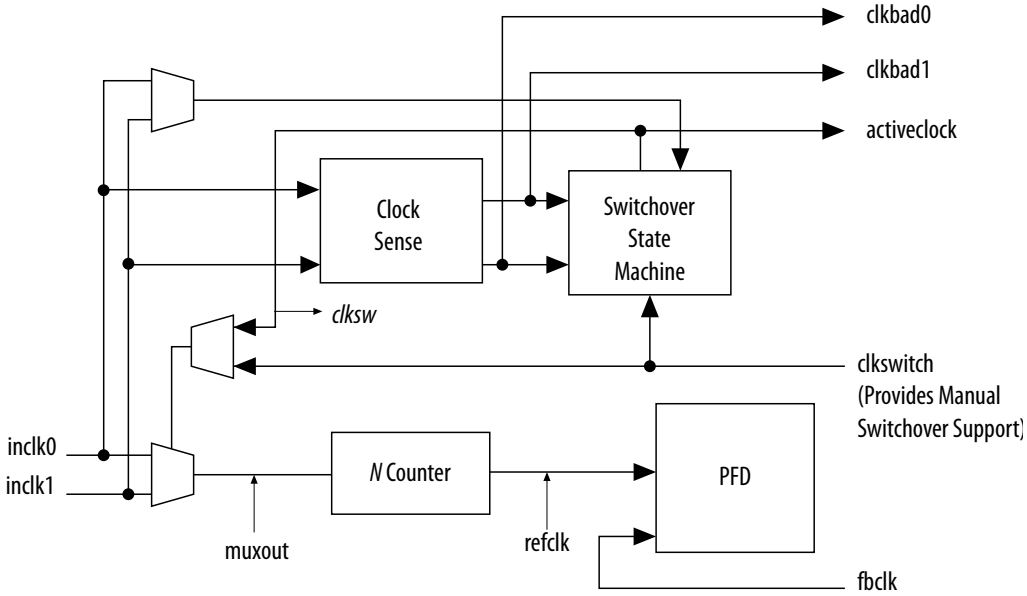
4.2.9.1. Automatic Clock Switchover

The Intel Cyclone 10 LP PLLs support a fully configurable clock switchover capability.

When the current reference clock is not present, the clock-sense block automatically switches to the backup clock for PLL reference. The clock switchover circuit also sends out three status signals—`clkbad0`, `clkbad1`, and `activeclock`—from the PLL to implement a custom switchover circuit. You can select a clock source at the backup clock by connecting it to the inclk1 port of the PLL in your design.

Figure 49. Automatic Clock Switchover Circuit Block Diagram

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

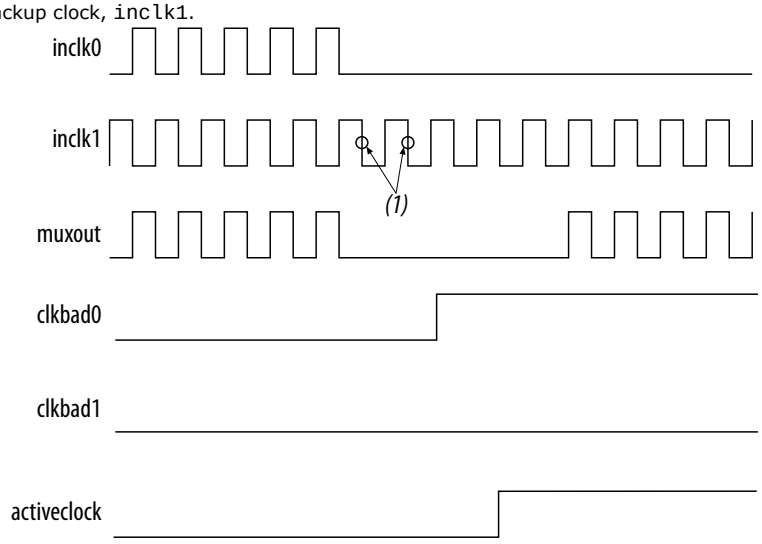


There are two ways to use the clock switchover feature:

- Use the switchover circuitry for switching from `inclk0` to `inclk1` running at the same frequency. For example, in applications that require a redundant clock with the same frequency as the reference clock, the switchover state machine generates a signal that controls the multiplexer select input. In this case, `inclk1` becomes the reference clock for the PLL. This automatic switchover can switch back and forth between the `inclk0` and `inclk1` clocks any number of times, when one of the two clocks fails and the other clock is available.
- Use the clkswitch input for user- or system-controlled switch conditions. This is possible 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 the switchover because the automatic clock-sense circuitry cannot monitor primary and secondary clock frequencies with a frequency difference of more than 20%. This feature is useful when clock sources can originate from multiple cards on the backplane, requiring a system-controlled switchover between frequencies of operation. Choose the secondary clock frequency so the VCO operates in the recommended frequency range. Also, set the `M`, `N`, and `C` counters accordingly to keep the VCO operating frequency in the recommended range.

Figure 50. Example of Automatic Switchover After Loss of Clock Detection

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



Note:

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

4.2.9.2. Manual Override

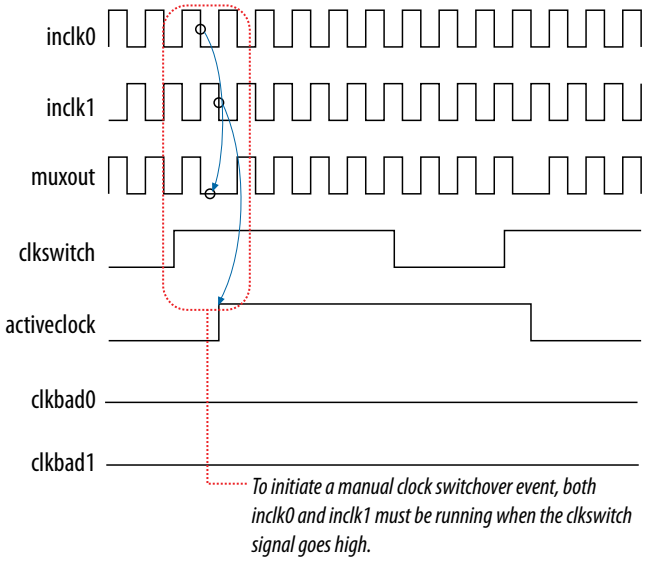
If you are using the automatic switchover, you must switch input clocks with the manual override feature with the `clkswitch` input.

The following figure shows an example of a waveform illustrating the switchover feature when controlled by `clkswitch`. In this case, both clock sources are functional and `inclk0` is selected as the reference clock. A low-to-high transition of the `clkswitch` signal starts the switchover sequence. The `clkswitch` signal must be high for at least three clock cycles (at least three of the longer clock period if `inclk0` and `inclk1` have different frequencies). On the falling edge of `inclk0`, the reference clock of the counter, `muxout`, is gated off to prevent any clock glitching. On the falling edge of `inclk1`, the reference clock multiplexer switches from `inclk0` to `inclk1` as the PLL reference, and the `activeclock` signal changes to indicate which clock is currently feeding the PLL.

In this mode, the `activeclock` signal mirrors the `clkswitch` signal. As both blocks are still functional during the manual switch, neither `clkbad` signals go high. Because the switchover circuit is positive edge-sensitive, the falling edge of the `clkswitch` signal does not cause the circuit to switch back from `inclk1` to `inclk0`. When the `clkswitch` signal goes high again, the process repeats. The `clkswitch` signal and the automatic switch only works depending on the availability of the clock that is switched to. If the clock is unavailable, the state machine waits until the clock is available.

When CLKSWITCH = 1, it overrides the automatic switchover function. As long as clkswitch signal is high, further switchover action is blocked.

Figure 51. Example of Clock Switchover Using the clkswitch (Manual) Control



4.2.9.3. Manual Clock Switchover

The Intel Cyclone 10 LP PLLs support manual switchover, in which the clkswitch signal controls whether inclk0 or inclk1 is the input clock to the PLL. The characteristics of a manual switchover are similar to the manual override feature in an automatic clock switchover, in which the switchover circuit is edge-sensitive. When the clkswitch signal goes high, the switchover sequence starts. The falling edge of the clkswitch signal does not cause the circuit to switch back to the previous input clock.

Related Information

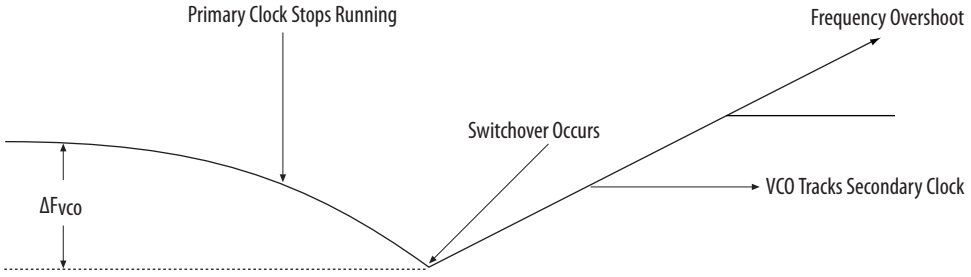
Customizing and Generating IP Cores, ALTPLL (Phase-Locked Loop) IP Core User Guide
Provides more information about PLL software support in the Intel Quartus Prime software.

4.2.9.4. Guidelines

Use the following guidelines to design with clock switchover in PLLs:

- Clock loss detection and automatic clock switchover require the `inclk0` and `inclk1` frequencies be within 20% of each other. Failing to meet this requirement causes the `clkbad0` and `clkbad1` signals to function improperly. When using manual clock switchover, the difference between `inclk0` and `inclk1` can be more than 20%. However, differences between the two clock sources (frequency, phase, or both) can cause the PLL to lose lock. Resetting the PLL ensures that the correct phase relationships are maintained between the input and output clocks.
- Both `inclk0` and `inclk1` must be running when the `clkswitch` signal goes high to start the manual clock switchover event. Failing to meet this requirement causes the clock switchover to malfunction.
- 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 slower than a high-bandwidth PLL. When the switchover happens, the low-bandwidth PLL propagates the stopping of the clock to the output slower than the high-bandwidth PLL. The low-bandwidth PLL filters out jitter on the reference clock. However, the low-bandwidth PLL also increases lock time.
- After a switchover occurs, there may be a finite resynchronization period for the PLL to lock onto a new clock. The exact amount of time it takes for the PLL to re-lock is dependent on the PLL configuration.
- If the phase relationship between the input clock to the PLL and output clock from the PLL is important in your design, assert `areset` for 10 ns after performing a clock switchover. Wait for the `locked` signal (or gated lock) to go high before re-enabling the output clocks from the PLL.
- Disable the system during switchover if the system is not tolerant to frequency variations during the PLL resynchronization period. You can use the `clkbad0` and `clkbad1` status signals to turn off the PFD (`pfdena` = 0) so the VCO maintains its last frequency. You can also use the switchover state machine to switch over to then secondary clock. Upon enabling the PFD, output clock enable signals (`clkena`) can disable clock outputs during the switchover and resynchronization period. After the lock indication is stable, the system can re-enable the output clock or clocks.
- The VCO frequency gradually decreases when the primary clock is lost and then increases as the VCO locks on to the secondary clock, as shown in the following figure. After the VCO locks on to the secondary clock, some overshoot can occur (an over-frequency condition) in the VCO frequency.

Figure 52. VCO Switchover Operating Frequency



4.2.10. Programmable Bandwidth

The PLL bandwidth is the measure of the PLL's ability to track the input clock and its associated jitter. The Intel Cyclone 10 LP PLLs provide advanced control of the PLL bandwidth using the programmable characteristics of the PLL loop, including loop filter and charge pump. The closed-loop gain 3-dB frequency in the PLL determines the PLL bandwidth. The bandwidth is approximately the unity gain point for open loop PLL response.

4.2.11. Programmable Phase Shift

Phase shift is used to implement a robust solution for clock delays in Intel Cyclone 10 LP devices. Phase shift is implemented with a combination of the VCO phase output and the counter starting time. The VCO phase output and counter starting time are the most accurate methods of inserting delays, because they are based only on counter settings that are independent of process, voltage, and temperature.

You can phase shift the output clocks from the Intel Cyclone 10 LP PLLs in one of two ways:

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

Fine resolution phase shifts are implemented by allowing any of the output counters (`C[4..0]`) or the `M` counter to use any of the eight phases of the VCO as the reference clock. This allows you to adjust the delay time with a fine resolution.

The following equation shows the minimum delay time that you can insert using this method.

Figure 53. Fine Resolution Phase Shift Equation

f_{REF} in this equation is the input reference clock frequency

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

For example, if f_{REF} is 100 MHz, `N` = 1, and `M` = 8, then f_{VCO} = 800 MHz, and Φ_{fine} = 156.25 ps. The PLL operating frequency defines this phase shift, a value that depends on reference clock frequency and counter settings.

Coarse resolution phase shifts are implemented by delaying the start of the counters for a predetermined number of counter clocks.

Figure 54. Coarse Resolution Phase Shift Equation

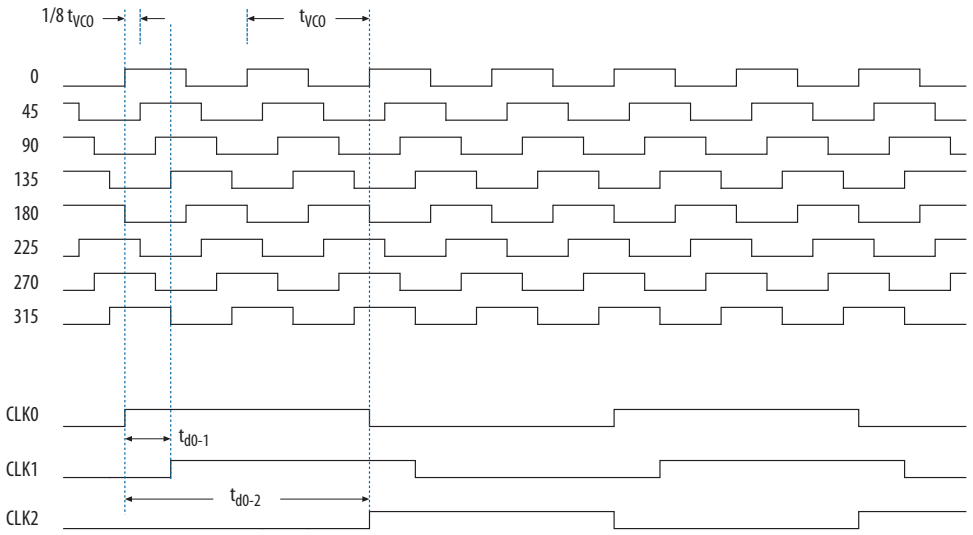
C in this equation is the count value set for the counter delay time—the initial setting in the PLL usage section of the compilation report in the Intel Quartus Prime software. If the initial value is 1, *C* – 1 = 0° phase shift.

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

Figure 55. Example of Delay Insertion Using VCO Phase Output and Counter Delay Time

The observations in this example are as follows:

- CLK0 is based on 0° phase from the VCO and has the *C* value for the counter set to one.
- CLK1 signal is divided by four, two VCO clocks for high time and two VCO clocks for low time. CLK1 is based on the 135° phase tap from the VCO and has the *C* value for the counter set to one.
- CLK2 signal is also divided by four. In this case, the two clocks are offset by 3 Φ_{fine}. CLK2 is based on the 0° phase from the VCO but has the *C* value for the counter set to three. This creates a delay of two Φ_{coarse} (two complete VCO periods).



You can use the coarse and fine phase shifts to implement clock delays in Intel Cyclone 10 LP devices.

Intel Cyclone 10 LP devices support dynamic phase shifting of VCO phase taps only. The phase shift is configurable for any number of times. Each phase shift takes about one scanclk cycle, allowing you to implement large phase shifts quickly.

4.2.12. PLL Cascading

Two PLLs are cascaded to each other through the clock network. If your design cascades PLLs, the source (upstream) PLL must have a low-bandwidth setting, while the destination (downstream) PLL must have a high-bandwidth setting.

4.2.13. PLL Reconfiguration

PLLs use several divide counters and different VCO phase taps to perform frequency synthesis and phase shifts. In Intel Cyclone 10 LP PLLs, you can reconfigure both counter settings and phase shift the PLL output clock in real time. You can also change the charge pump and loop filter components, which dynamically affects PLL bandwidth. You can use these PLL components to update the output clock frequency, PLL bandwidth, and phase shift in real time, without reconfiguring the entire FPGA.

The ability to reconfigure the PLL in real time is useful in applications that might operate at multiple frequencies. It is also useful in prototyping environments, allowing you to sweep PLL output frequencies and adjust the output clock phase dynamically. For instance, a system generating test patterns is required to generate and send

patterns at 75 or 150 MHz, depending on the requirements of the device under test. Reconfiguring PLL components in real time allows you to switch between two such output frequencies in a few microseconds.

You can also use this feature to adjust clock-to-out (t_{CO}) delays in real time by changing the PLL output clock phase shift. This approach eliminates the need to regenerate a configuration file with the new PLL settings.

4.2.13.1. PLL Reconfiguration Hardware

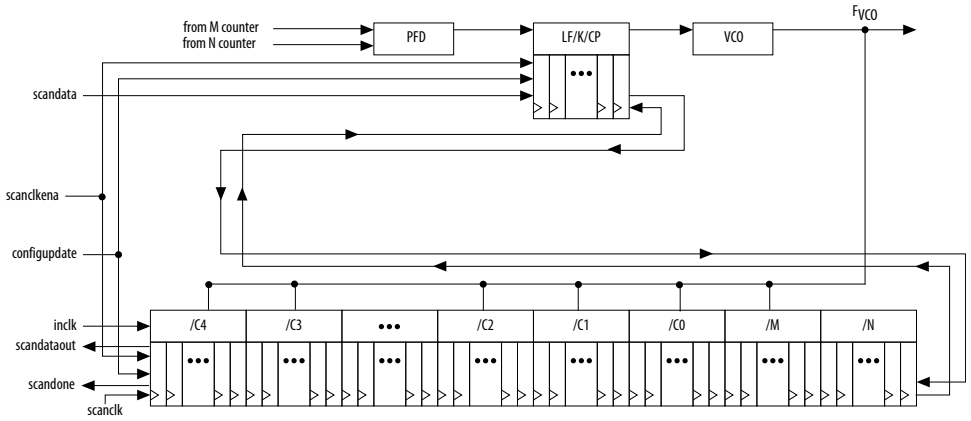
The following PLL components are configurable in real time:

- Pre-scale counter (N)
- Feedback counter (M)
- Post-scale output counters (C0-C4)
- Charge pump current (I_{CP})
- Loop filter components (R, C)

You can dynamically adjust the charge pump current (I_{CP}) and loop filter components (R, C) to facilitate on-the-fly reconfiguration of the PLL bandwidth.

Figure 56. PLL Reconfiguration Scan Chain

This figure shows the dynamic adjustment of the PLL counter settings by shifting their new settings into a serial shift register chain or scan chain. Serial data shifts to the scan chain via the `scandata` port, and shift registers are clocked by `scanclk`. The maximum `scanclk` frequency is 100 MHz. After shifting the last bit of data, asserting the `configupdate` signal for at least one `scanclk` clock cycle synchronously updates the PLL configuration bits with the data in the scan registers.



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

4.2.13.1.1. Post-Scale Counters (C0 to C4)

You can configure the multiply or divide values and duty cycle of the post-scale counters in real time. Each counter has an 8-bit high time setting and an 8-bit low time setting. The duty cycle is the ratio of output high or low time to the total cycle time, which is the sum of the two.

The post-scale counters have two control bits:

- `rbypass`—For bypassing the counter
- `rselodd`—For selecting the output clock duty cycle

When the `rbypass` bit is set to 1, it bypasses the counter, resulting in a division by one. When this bit is set to 0, the PLL computes the effective division of the VCO output frequency based on the high and low time counters. The PLL implements this duty cycle by transitioning the output clock from high-to-low on the rising edge of the VCO output clock.

For example, if the post-scale divide factor is 10, the high and low count values are set to 5 and 5 respectively, to achieve a 50–50% duty cycle. However, a 4 and 6 setting for the high and low count values, respectively, would produce an output clock with 40–60% duty cycle.

The `rselodd` bit indicates an odd divide factor for the VCO output frequency with a 50% duty cycle. The PLL implements this duty cycle by transitioning the output clock from high-to-low on a falling edge of the VCO output clock.

For example, if the post-scale divide factor is 3, the high and low time count values are 2 and 1 respectively, to achieve this division. This implies a 67%–33% duty cycle. If you need a 50%–50% duty cycle, you must set the `rselodd` control bit to 1 to achieve this duty cycle despite an odd division factor. When you set `rselodd` = 1, subtract 0.5 cycles from the high time and add 0.5 cycles to the low time.

The calculation for the example is shown as follows:

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

4.2.13.1.2. Scan Chain

The Intel Cyclone 10 LP PLLs have a 144-bit scan chain.

Table 18. PLL Component Reprogramming Bits

Block Name	Number of Bits		
	Counter	Other	Total
C4[11]	16	2 [12]	18
C3	16	2 [12]	18
C2	16	2 [12]	18

continued...

(11) LSB bit for C4 low-count value is the first bit shifted into the scan chain.

(12) These two control bits include `rbypass`, for bypassing the counter, and `rselodd`, for selecting the output clock duty cycle.

Block Name	Number of Bits		
	Counter	Other	Total
C1	16	2 (12)	18
C0	16	2 (12)	18
M	16	2 (12)	18
N	16	2 (12)	18
Charge Pump	9	0	9
Loop Filter(13)	9	0	9
Total number of bits			144

Figure 57. PLL Component Scan Chain Order

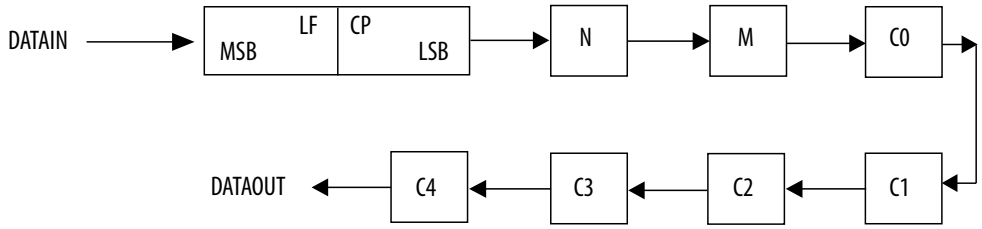
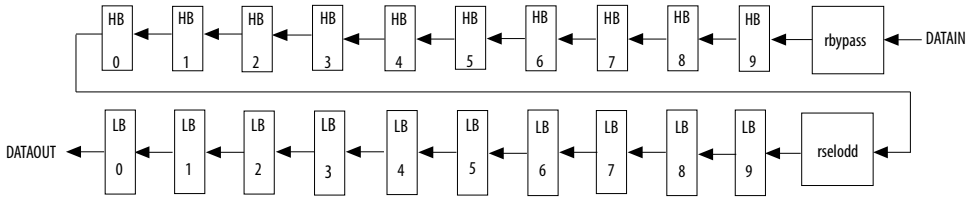


Figure 58. PLL Post-Scale Counter Scan Chain Bit Order



4.2.13.1.3. Charge Pump and Loop Filter

You can reconfigure the following settings to update the PLL bandwidth in real time:

- Charge pump (I_{CP})
- Loop filter resistor (R)
- Loop filter capacitor (C)

Table 19. Charge Pump Bit Control

CP[2]	CP[1]	CP[0]	Setting (Decimal)
0	0	0	0
1	0	0	1
1	1	0	3
1	1	1	7

(13) MSB bit for loop filter is the last bit shifted into the scan chain.

Table 20. Loop Filter Resistor Value Control

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

Table 21. Loop Filter High Frequency Capacitor Control

LFC[1]	LFC[0]	Setting (Decimal)
0	0	0
0	1	1
1	1	3

4.2.13.1.4. Bypassing PLL Counter

Bypassing a PLL counter results in a divide (N, C0 to C4 counters) factor of one.

Table 22. PLL Counter Settings

Description	PLL Scan Chain Bits [0..8] Settings								
	LSB								MSB
PLL counter bypassed	X	X	X	X	X	X	X	X	1 (14)
PLL counter not bypassed	X	X	X	X	X	X	X	X	0 (14)

To bypass any of the PLL counters, set the bypass bit to 1. The values on the other bits are ignored.

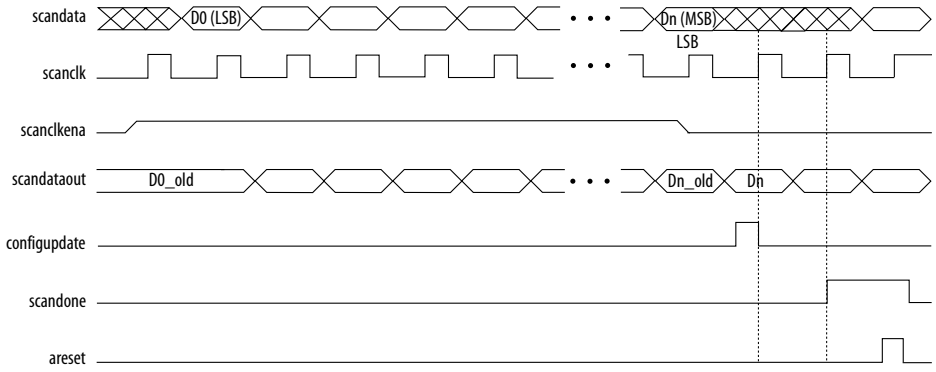
(14) Bypass bit

4.2.13.2. PLL Reconfiguration Implementation

To reconfigure the PLL counters, perform the following steps:

1. Assert the scanclkena signal at least one scanclk cycle prior to shifting in the first bit of scandata (D0).
2. Shift the serial data (scandata) into the scan chain on the second rising edge of scanclk.
3. After all 144 bits have been scanned into the scan chain, deassert the scanclkena signal to prevent inadvertent shifting of bits in the scan chain.
4. Assert the configupdate signal for one scanclk cycle to update the PLL counters with the contents of the scan chain.
   The scandone signal goes high indicating that the PLL is being reconfigured. A falling edge indicates that the PLL counters have been updated with new settings.
5. Reset the PLL using the areset signal if you make any changes to the M, N, post-scale output C counters, or the I_{CP} , R, and C settings.
6. You can repeat steps 1 through 5 to reconfigure the PLL any number of times.

Figure 59. PLL Reconfiguration Scan Chain Functional Simulation



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

4.2.13.3. Dynamic Phase Shift

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

Table 23. Dynamic Phase Shifting Control Signals

Signal Name	Description	Source	Destination
phasecounterselect[2..0]	Counter select. Three bits decoded to select either the M or one of the C counters for phase adjustment. One address map to select all C counters. This signal is registered in the PLL on the rising edge of scanclk.	Logic array or I/O pins	PLL reconfiguration circuit
phaseupdown	Selects dynamic phase shift direction; 1= UP, 0 = DOWN. Signal is registered in the PLL on the rising edge of scanclk.	Logic array or I/O pins	PLL reconfiguration circuit
phasestep	Logic high enables dynamic phase shifting.	Logic array or I/O pins	PLL reconfiguration circuit
scanclk	Free running clock from core used in combination with phasestep to enable or disable dynamic phase shifting. Shared with scanclk for dynamic reconfiguration.	GCLK or I/O pins	PLL reconfiguration circuit
phasedone	When asserted, it indicates to core logic that the phase adjustment is complete and PLL is ready to act on a possible second adjustment pulse. Asserts based on internal PLL timing. De-asserts on the rising edge of scanclk.	PLL reconfiguration circuit	Logic array or I/O pins

Table 24. Phase Counter Select Mapping

PLL Counter Selection	PHASECOUNTERSELECT		
	[2]	[1]	[0]
All output counters	0	0	0
M counter	0	0	1
C0 counter	0	1	0
C1 counter	0	1	1
C2 counter	1	0	0
C3 counter	1	0	1
C4 counter	1	1	0

4.2.13.4. Dynamic Phase Shift Implementation

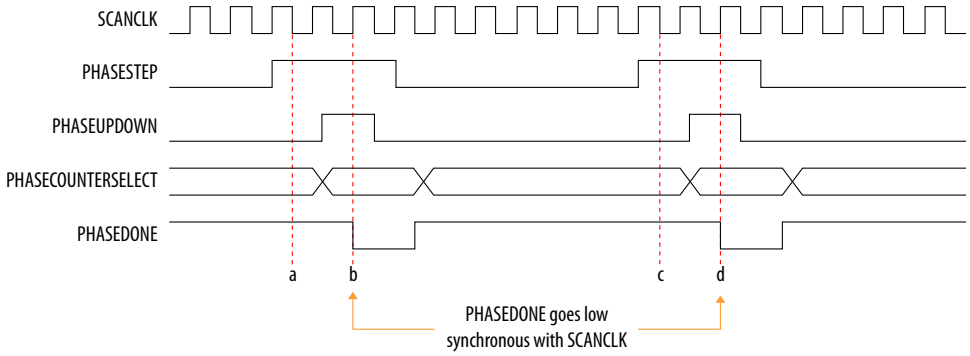
To perform one dynamic phase shift step, perform the following steps:

1. Set PHASEUPDOWN and PHASECOUNTERSELECT as required.
2. Assert PHASESTEP for at least two SCANCLK cycles. Each PHASESTEP pulse allows one phase shift.
3. Deassert PHASESTEP after PHASEDONE goes low.
4. Wait for PHASEDONE to go high.
5. Repeat steps 1 through 4 as many times as required to perform multiple phase shifts.

PHASEUPDOWN and PHASECOUNTERSELECT signals are synchronous to SCANCLK and must meet the t_{su} and t_h requirements with respect to the SCANCLK edges.

You can repeat dynamic phase-shifting indefinitely. For example, in a design where the VCO frequency is set to 1,000 MHz and the output clock frequency is set to 100 MHz, performing 40 dynamic phase shifts (each one yields 125 ps phase shift) results in shifting the output clock by 180º, which is a phase shift of 5 ns.

Figure 60. Dynamic Phase Shift Timing Diagram



The PHASESTEP signal is latched on the negative edge of SCANCLK (a,c) and must remain asserted for at least two SCANCLK cycles. Deassert PHASESTEP after PHASEDONE goes low.

On the second SCANCLK rising edge (b,d) after PHASESTEP is latched, the values of PHASEUPDOWN and PHASECOUNTERSELECT are latched. The PLL starts dynamic phase-shifting for the specified counters and in the indicated direction.

The PHASEDONE signal is deasserted synchronous to SCANCLK at the second rising edge (b,d) and remains low until the PLL finishes dynamic phase-shifting. Depending on the VCO and SCANCLK frequencies, PHASEDONE low time may be greater than or less than one SCANCLK cycle.

You can perform another dynamic phase-shift after the PHASEDONE signal goes from low to high. Each PHASESTEP pulse enables one phase shift. The PHASESTEP pulses must be at least one SCANCLK cycle apart.

Related Information

Phase-Locked Loop Reconfiguration (ALTPLL_RECONFIG) IP Core User Guide
Provides more information about the ALTPLL_RECONFIG parameter editor.

4.2.14. Spread-Spectrum Clocking

Intel Cyclone 10 LP devices can accept a spread-spectrum input with typical modulation frequencies. However, the device cannot automatically detect that the input is a spread-spectrum signal. Instead, the input signal looks like deterministic jitter at the input of the PLL. The Intel Cyclone 10 LP PLLs can track a spread-spectrum input clock as long as it is in the input jitter tolerance specifications and the modulation frequency of the input clock is below the PLL bandwidth, that is specified in the fitter report. Intel Cyclone 10 LP devices cannot generate spread-spectrum signals internally.

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

Document Version	Changes
2018.10.22	• Updated the *Example of Automatic Switchover After Loss of Clock Detection* diagram.
2018.05.07	• Removed the note to the `CLK[n]` pin in the *Clock Control Block* diagram. The `CLK0` pin is available in the Intel Cyclone 10 LP devices. Note removed: `CLK[n]` is not available on the left side of the Intel Cyclone 10 LP devices. • Updated the clock pin to `CLK[3..0]` on the left side of the Intel Cyclone 10 LP devices in the *Clock Networks and Clock Control Block Locations in Intel Cyclone 10 LP Devices* diagram.

Date	Version	Changes
December 2017	2017.12.22	• Updated the dedicated clock pins in the *Clock Networks* section. • Added GCLK resources for `CLK0/DIFFCLK_0p` and `CLK1/DIFFCLK_0n` in the *Intel Cyclone 10 LP Clock Sources Connectivity to the GCLK Networks* table.
May 2017	2017.05.08	Initial release.

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

Intel Cyclone 10 LP devices offer highly configurable GPIOs with these features:

- Support for various single-ended and differential I/O standards.
- Programmable current strength, bus hold, pull-up resistors, and delays.
- Programmable slew rate control to optimize signal integrity.
- Optional open-drain output for each I/O pin.
- True and emulated LVDS buffers with LVDS SERDES implemented using logic elements in the device core.
- Programmable pre-emphasis for the true LVDS output buffers.
- Calibrated on-chip series termination (R_S OCT) or driver impedance matching (R_S) for single-endd I/O standards.
- Support for hot socketing implementation.

Related Information

- Intel Cyclone 10 LP Device Pin-Outs
- Intel Cyclone 10 LP Pin Connection Guidelines

5.1. Intel Cyclone 10 LP I/O Standards Support

Intel Cyclone 10 LP devices support a wide range of I/O standards, including single-ended, voltage-referenced single-ended, and differential I/O standards.

Table 25. Supported I/O Standards in Intel Cyclone 10 LP Devices

I/O Standard	Type	I/O Bank	Direction		Application	Standard Support
			Input	Output		
3.3 V LVTTL/3.3 V LVCMOS(15)	Single-ended	All	Yes	Yes	General purpose	JESD8-B
3.0 V LVTTL/3.0 V LVCMOS(15)	Single-ended	All	Yes	Yes	General purpose	JESD8-B
2.5 V LVTTL/2.5 V LVCMOS	Single-ended	All	Yes	Yes	General purpose	JESD8-5
1.8 V LVTTL/1.8 V LVCMOS	Single-ended	All	Yes	Yes	General purpose	JESD8-7
1.5 V LVCMOS	Single-ended	All	Yes	Yes	General purpose	JESD8-11
1.2 V LVCMOS	Single-ended	All	Yes	Yes	General purpose	JESD8-12

continued...

(15) You must enable the PCI clamp diode.

I/O Standard	Type	I/O Bank	Direction		Application	Standard Support
			Input	Output		
3.0 V PCI/3.0 V PCI-X	Single-ended	All	Yes	Yes	General purpose	PCI Rev. 2.2
SSTL-2 Class I and II (16)	Voltage-referenced	All	Yes	Yes	General purpose	JESD8-9B
SSTL-18 Class I and II (16)	Voltage-referenced	All	Yes	Yes	General purpose	JESD8-15
1.8 V HSTL Class I and II (16)	Voltage-referenced	All	Yes	Yes	General purpose	JESD8-6
1.5 V HSTL Class I and II (16)	Voltage-referenced	All	Yes	Yes	General purpose	JESD8-6
1.2 V HSTL Class I	Voltage-referenced	All	Yes	Yes	General purpose	JESD8-16A
1.2 V HSTL Class II	Voltage-referenced	1, 2, 5, 6	Yes	—		
		3, 4, 7, 8	Yes	Yes		
Differential SSTL-2 Class I and II (16)	Differential	All	Yes(17)	Yes(18)	General purpose	JESD8-9B
Differential SSTL-18 Class I (16)	Differential	All	Yes(17)	Yes(18)	General purpose	JESD8-15
Differential SSTL-18 Class II (16)	Differential	All	Yes(17)	—		
Differential 1.8 V HSTL Class I (16)	Differential	All	Yes(17)	Yes(18)	General purpose	JESD8-6
Differential 1.8 V HSTL Class II (16)	Differential	All	Yes(17)	—		
Differential 1.5 V HSTL Class I (16)	Differential	All	Yes(17)	Yes(18)	General purpose	JESD8-6
Differential 1.5 V HSTL Class II (16)	Differential	All	Yes(17)	—		
Differential 1.2 V HSTL Class I	Differential	All	Yes(17)	Yes(18)	General purpose	JESD8-16A
Differential 1.2 V HSTL Class II	Differential	All	Yes(17)	—		
LVDS (dedicated)	Differential	1, 2, 5, 6	Yes	Yes	—	ANSI/TIA/EIA-644
LVDS (emulated)(19)	Differential	3, 4, 5, 6, 7, 8	Yes	Yes	—	ANSI/TIA/EIA-644

continued...

(16) Even though the Intel Cyclone 10 LP I/O buffers support various I/O standards for memory application, Intel does not validate nor support any IP that is intended for memory applications such as DDR or DDR2.

(17) The inputs treat differential inputs as two single-ended inputs but decode only one of them.

(18) The outputs uses two single-ended outputs with the second output programmed as inverted.

(19) Emulated LVDS, mini-LVDS, RSDS, and PPDS I/O standards requires external resistors.

I/O Standard	Type	I/O Bank	Direction		Application	Standard Support
			Input	Output		
Mini-LVDS (dedicated)	Differential	1, 2, 5, 6	—	Yes	—	—
Mini-LVDS (emulated)(19)	Differential	3, 4, 5, 6, 7, 8	—	Yes	—	—
RSDS (dedicated)	Differential	1, 2, 5, 6	—	Yes	—	—
RSDS (emulated)(19)	Differential	3, 4, 5, 6, 7, 8	—	Yes	—	—
PPDS (dedicated)	Differential	1, 2, 5, 6	—	Yes	—	—
PPDS (emulated)(19)	Differential	All	—	Yes	—	—
Differential LVPECL	Differential	All	Yes	—	—	—
Bus LVDS	Differential	All	Yes(20)	Yes(18)	—	—

Related Information

- Intel Cyclone 10 LP I/O Banks Locations on page 86
  Shows the location of the row and column I/O banks.
- Differential I/O Standards Support on page 100
- Intel Cyclone 10 LP Device Pin-Outs

5.1.1. Intel Cyclone 10 LP I/O Standards Voltage and Pin Support

Table 26. Intel Cyclone 10 LP I/O Standards Voltage Levels and Pin Support

Note: The I/O standards that each pin type supports depends on the I/O standards that the pin's I/O bank supports. For example, only the I/O banks 1, 2, 4, and 6 support the LVDS (dedicated) I/O standard. To determine the pin's I/O bank locations for your device, check your device's pin out file.

I/O Standard	V_{CCIO} (V)		Pin Type Support		
	Input	Output	PLL_CLKOUT	CLK	User I/O
3.3 V LVTTL/3.3 V LVCMOS(21)	3.3/3.0/2.5	3.3	Yes	Yes	Yes
3.0 V LVTTL/3.0 V LVCMOS(21)	3.3/3.0/2.5	3.0	Yes	Yes	Yes
2.5 V LVTTL/2.5 V LVCMOS	3.3/3.0/2.5	2.5	Yes	Yes	Yes
1.8 V LVTTL/1.8 V LVCMOS	1.8/1.5	1.8	Yes	Yes	Yes
1.5 V LVCMOS	1.8/1.5	1.5	Yes	Yes	Yes
1.2 V LVCMOS	1.2	1.2	Yes	Yes	Yes
3.0 V PCI/3.0 V PCI-X(22)	3.0	3.0	Yes	Yes	Yes

continued...

(20) The inputs use LVDS input buffers.

(21) You must enable the PCI clamp diode.

(22) The 3.0 V PCI and PCI-X support is fully compatible for direct interface with 3.3 V PCI systems without additional components requirement. The outputs conforms to the V_{IH} and V_{IL} requirements of 3.3 V PCI and PCI-X inputs with sufficient noise margin.

I/O Standard	V_{CCIO} (V)		Pin Type Support		
	Input	Output	PLL_CLKOUT	CLK	User I/O
SSTL-2 Class I and II	2.5	2.5	Yes	Yes	Yes
SSTL-18 Class I and II	1.8	1.8	Yes	Yes	Yes
1.8 V HSTL Class I and II	1.8	1.8	Yes	Yes	Yes
1.5 V HSTL Class I and II	1.5	1.5	Yes	Yes	Yes
1.2 V HSTL Class I	1.2	1.2	Yes	Yes	Yes
1.2 V HSTL Class II	1.2	—	—	Yes	Yes
	—	1.2	Yes	Yes(23)	Yes(23)
Differential SSTL-2 Class I and II	2.5	—	—	Yes	—
	—	2.5	Yes	—	—
Differential SSTL-18 Class I	1.8	—	—	Yes	—
	—	1.8	Yes	—	—
Differential SSTL-18 Class II	1.8	—	—	Yes	—
Differential 1.8 V HSTL Class I	1.8	—	—	Yes	—
	—	1.8	Yes	—	—
Differential 1.8 V HSTL Class II	1.8	—	—	Yes	—
Differential 1.5 V HSTL Class I	1.5	—	—	Yes	—
	—	1.5	Yes	—	—
Differential 1.5 V HSTL Class II	1.5	—	—	Yes	—
Differential 1.2 V HSTL Class I	1.2	—	—	Yes	—
	—	1.2	Yes	—	—
Differential 1.2 V HSTL Class II	1.2	—	—	Yes	—
LVDS (dedicated)	2.5	2.5	—	Yes(24)	Yes(24)
LVDS (emulated)(25)	2.5	2.5	Yes	Yes(26)	Yes(26)
Mini-LVDS (dedicated)	—	2.5	—	—	Yes(24)
Mini-LVDS (emulated)(25)	—	2.5	Yes	—	Yes(26)
RSDS (dedicated)	—	2.5	—	—	Yes(24)
RSDS (emulated)(25)	—	2.5	Yes	—	Yes(26)
PPDS (dedicated)	—	2.5	Yes	—	Yes(24)

continued...

(23) Only for I/O banks 3, 4, 7, and 8.

(24) Only for I/O banks 1, 2, 5, and 6.

(25) Emulated LVDS, mini-LVDS, RSDS, and PPDS I/O standards requires external resistors.

(26) Only for I/O banks 3, 4, 5, 6, 7, and 8.

I/O Standard	V_{CCIO} (V)		Pin Type Support		
	Input	Output	PLL_CLKOUT	CLK	User I/O
PPDS (emulated)[25]	—	2.5	Yes	—	Yes
Differential LVPECL	2.5	—	—	Yes	—
Bus LVDS	2.5	2.5	—	—	Yes

Related Information

- Intel Cyclone 10 LP I/O Banks Locations on page 86
  Shows the location of the row and column I/O banks.
- Differential I/O Standards Support on page 100

5.2. I/O Resources in Intel Cyclone 10 LP Devices

5.2.1. Intel Cyclone 10 LP Devices I/O Resources Per Package

Table 27. Package Plan for Intel Cyclone 10 LP Devices

The GPIO counts do not include the DCLK pins. The LVDS counts include DIFFIO and DIFFCLK pairs only—LVDS I/Os with both p and n pins. Refer to the related information.

Device	Package											
	Type	M164 164-pin MBGA		U256 256-pin UBGA		U484 484-pin UBGA		E144 144-pin EQFP		F484 484-pin FBGA		F780 780-pin FBGA	
	I/O Type	GPIO	LVDS	GPIO	LVDS	GPIO	LVDS	GPIO	LVDS	GPIO	LVDS	GPIO	LVDS
10CL006		—	—	176	65	—	—	88	22	—	—	—	—
10CL010		101	26	176	65	—	—	88	22	—	—	—	—
10CL016		87	22	162	53	340	137	78	19	340	137	—	—
10CL025		—	—	150	52	—	—	76	18	—	—	—	—
10CL040		—	—	—	—	325	124	—	—	325	124	—	—
10CL055		—	—	—	—	321	132	—	—	321	132	—	—
10CL080		—	—	—	—	289	110	—	—	289	110	423	178
10CL120		—	—	—	—	—	—	—	—	277	103	525	230

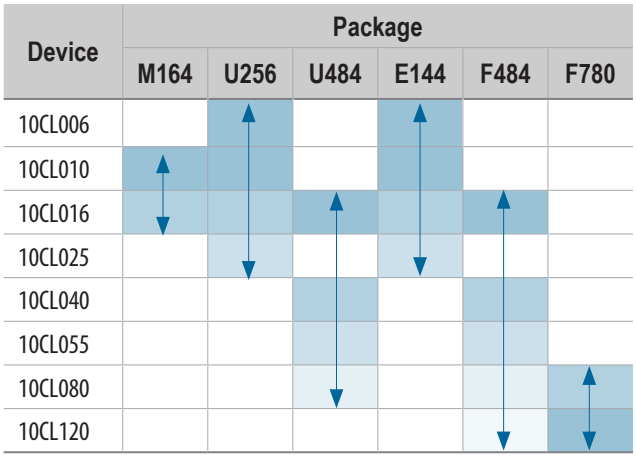
Related Information

- Why does the Intel Quartus Prime software device pin-out show a different number of pins compared to the Intel Cyclone 10 LP Device Overview?
- How is the LVDS pair count that is published in the Intel Cyclone 10 LP Device Overview calculated?

5.2.2. Intel Cyclone 10 LP I/O Vertical Migration

Figure 61. Migration Capability Across Intel Cyclone 10 LP Devices

- The arrows indicate the migration paths. The devices included in each vertical migration path are shaded. Devices with lesser I/O resources in the same path have lighter shades.
- To achieve full I/O migration across devices in the same migration path, restrict I/O usage to match the device with the lowest I/O count.



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

5.2.3. Intel Cyclone 10 LP VREF Pins Per I/O Bank

Table 28. Number of VREF Pins Per I/O Bank in Intel Cyclone 10 LP Device Packages

Device	Package	I/O Bank							
		1	2	3	4	5	6	7	8
10CL006	U256	1	1	1	1	1	1	1	1
	E144	1	1	1	1	1	1	1	1
10CL010	M164	1	1	1	1	1	1	1	1
	U256	1	1	1	1	1	1	1	1
	E144	1	1	1	1	1	1	1	1
10CL016	M164	2	2	2	2	2	2	2	2
	U256	2	2	2	2	2	2	2	2
	U484	2	2	2	2	2	2	2	2
	E144	2	2	2	2	2	2	2	2
	F484	2	2	2	2	2	2	2	2
10CL025	U256	1	1	1	1	1	1	1	1
	E144	1	1	1	1	1	1	1	1
10CL040	U484	4	4	4	4	4	4	4	4
	F484	4	4	4	4	4	4	4	4

continued...

Device	Package	I/O Bank							
		1	2	3	4	5	6	7	8
10CL055	U484	2	2	2	2	2	2	2	2
	F484	2	2	2	2	2	2	2	2
10CL080	U484	3	3	3	3	3	3	3	3
	F484	3	3	3	3	3	3	3	3
	F780	3	3	3	3	3	3	3	3
10CL120	F484	3	3	3	3	3	3	3	3
	F780	3	3	3	3	3	3	3	3

5.2.4. Intel Cyclone 10 LP LVDS Channels Support

The LVDS channels available vary among Intel Cyclone 10 LP device packages.

Table 29. LVDS Buffers in Intel Cyclone 10 LP Devices

- True LVDS input is supported on top, right, and bottom I/O banks while output is supported on the right and left I/O banks.
- Emulated LVDS input and output supported on all I/O banks.

Product Line	Package	LVDS Pairs	Device Side	LVDS Pairs Per Side
10CL006	U256	65	Top	20
			Right	12
			Left	11
			Bottom	22
	E144	22	Top	6
			Right	6
			Left	3
			Bottom	7
10CL010	M164	26	Top	9
			Right	7
			Left	3
			Bottom	7
	U256	65	Top	20
			Right	12
			Left	11
			Bottom	22
	E144	22	Top	6
			Right	6
			Left	3
			Bottom	7

continued...

Product Line	Package	LVDS Pairs	Device Side	LVDS Pairs Per Side
10CL016	M164	22	Top	7
			Right	6
			Left	3
			Bottom	6
	U256	53	Top	14
			Right	12
			Left	9
			Bottom	18
	U484	137	Top	36
			Right	37
			Left	31
			Bottom	33
	E144	19	Top	6
			Right	5
			Left	2
			Bottom	6
	F484	137	Top	36
			Right	37
			Left	31
			Bottom	33
10CL025	U256	52	Top	17
			Right	10
			Left	10
			Bottom	15
	E144	18	Top	5
			Right	5
			Left	3
			Bottom	5
10CL040	U484	124	Top	33
			Right	32
			Left	29
			Bottom	30
	F484	124	Top	33
			Right	32
			Left	29
			Bottom	30

continued...

Product Line	Package	LVDS Pairs	Device Side	LVDS Pairs Per Side
10CL055	U484	132	Top	36
			Right	34
			Left	29
			Bottom	33
	F484	132	Top	36
			Right	34
			Left	29
			Bottom	33
10CL080	U484	110	Top	29
			Right	29
			Left	26
			Bottom	26
	F484	110	Top	29
			Right	29
			Left	26
			Bottom	26
	F780	178	Top	51
			Right	43
			Left	35
			Bottom	49
10CL120	F484	103	Top	28
			Right	26
			Left	25
			Bottom	24
	F780	230	Top	65
			Right	53
			Left	49
			Bottom	63

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

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

- ALTIOBUF—supports operations of the GPIO components.
- ALTLVDS—supports operations of the high-speed source-synchronous SERDES.
- ALTDDIO—supports configuration and implementation of double data rate I/O registers.

Related Information

- LVDS SERDES Transmitter/Receiver IP Cores User Guide
- I/O Buffer (ALTIOBUF) IP Core User Guide
- Double Data Rate I/O (ALTDDIO_IN, ALTDDIO_OUT, and ALTDDIO_BIDIR) IP Cores User Guide

5.4. Intel Cyclone 10 LP I/O Elements

The Intel Cyclone 10 LP I/O elements (IOEs) contain a bidirectional I/O buffer and five registers for registering input, output, output-enable signals, and complete embedded bidirectional single data rate (SDR) and double data rate (DDR) transfer.

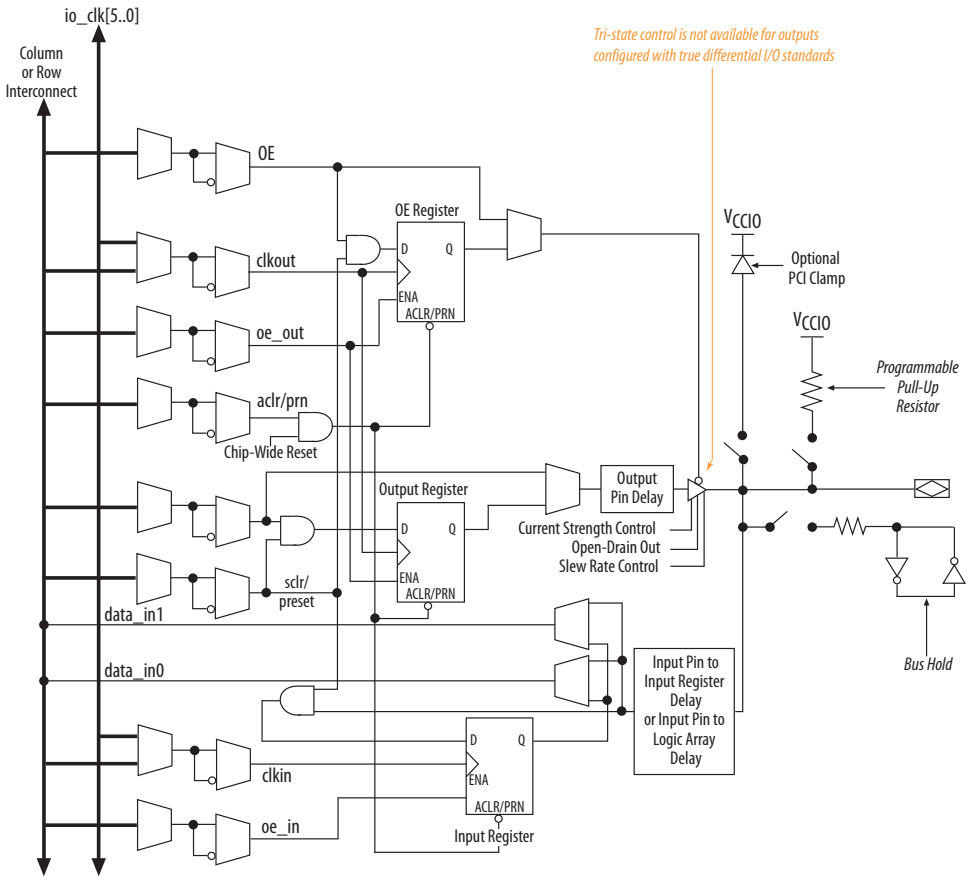
Note: Intel does not validate or support any IP that is intended for memory application, such as DDR, in Intel Cyclone 10 LP devices.

Each IOE contains one input register, two output registers, and two output-enable (OE) registers:

- The two output registers and two OE registers are used for DDR applications.
- You can use the input registers for fast setup times and output registers for fast clock-to-output times.
- You can use the OE registers for fast clock-to-output enable times.

You can use the IOEs for input, output, or bidirectional data paths. The I/O pins support various single-ended and differential I/O standards.

Figure 62. IOE Structure in Bidirectional Configuration for SDR Mode



5.4.1. Intel Cyclone 10 LP I/O Banks Architecture

The device I/O pins are grouped into eight groups and each group is associated with an I/O bank. Every I/O bank in the device has a separate power bus.

The only exception is HSTL-12 Class II, which is only supported in column I/O banks.

There are two types of I/O banks in Intel Cyclone 10 LP devices:

- Row I/O banks—I/O banks 1, 2, 5, and 6, located on the left and right side of the device
- Column I/O banks—I/O banks 3, 4, 7, and 8, located on the top and bottom side of the device

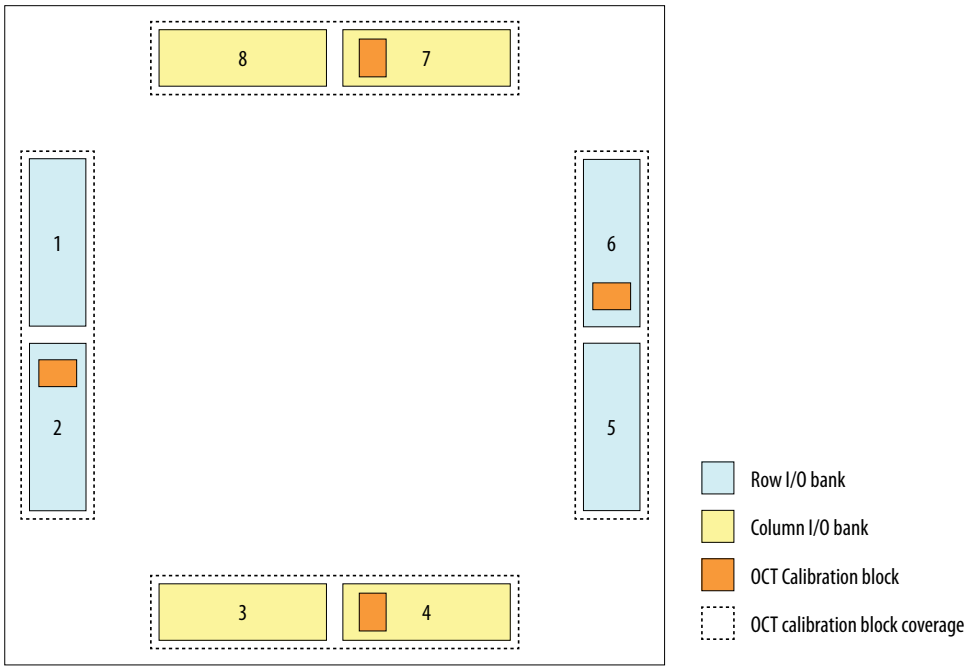
Only the column I/O banks support the 1.2 V HSTL Class II I/O standard. Both row and column I/O banks support all the other single-ended and differential I/O standards that the Intel Cyclone 10 LP device supports.

5.4.2. Intel Cyclone 10 LP I/O Banks Locations

The I/O banks are located at the periphery of the device.

Figure 63. I/O Banks for Intel Cyclone 10 LP Devices

This figure is only a graphical representation showing the top view of the silicon die.



For details about the I/O banks and the exact pin locations in each device package, refer to the Intel Quartus Prime software and the relevant device pin-out file.

Related Information

- Intel Cyclone 10 LP I/O Standards Voltage and Pin Support on page 78
  Lists the supported I/O direction, voltages, and pin type for each I/O standard.
- Intel Cyclone 10 LP I/O Standards Support on page 76
  Lists the supported I/O standards and availability of each I/O standard in the device I/O banks.
- Intel Cyclone 10 LP Device Pin-Outs
- OCT Calibration on page 97

5.5. Intel Cyclone 10 LP Clock Pins Input Support

Intel Cyclone 10 LP clock pins have multiple purposes.

Table 30. Clock Pins Input Support in Intel Cyclone 10 LP Devices

Clock Pin	Input Support
CLK	Input support for single-ended and voltage-referenced standards.
DIFFCLK	Input support for differential I/O standards. If used as DIFFCLK pins: • You can use DC or AC coupling depending on the interface requirements. • The input requires external termination.

Related Information

- Intel Cyclone 10 LP I/O Standards Support on page 76
- Intel Cyclone 10 LP I/O Standards Voltage and Pin Support on page 78
- Differential I/O Standards Support on page 100

5.6. Programmable IOE Features in Intel Cyclone 10 LP Devices

The Intel Cyclone 10 LP I/O buffers support a range of programmable features. These features increase the flexibility of I/O utilization and provide an alternative to reduce the usage of external discrete components such as a pull-up resistor and a diode.

Table 31. Summary of Supported Intel Cyclone 10 LP Programmable I/O Buffer Features and Settings

Feature	Setting	Condition	Assignment Name
Open Drain	• On • Off—Default	To enable this feature, use the OPNDRN primitive.	—
Bus-Hold	• On • Off—Default	Disabled if you use the weak pull-up resistor feature.	Enable Bus-Hold Circuitry
Pull-up Resistor	• On • Off—Default	Disabled if you use the bus-hold feature.	Weak Pull-Up Resistor
Slew Rate Control	• 0 (Slow) • 1 (Medium) • 2 (Fast)—Default	Disabled if you use OCT with calibration.	Slew Rate
PCI Clamp Diode	• On—Default for input pins using supported I/O standards • Off—Default for output pins except 3.0 V PCI/PCI-X.	—	PCI I/O
Pre-Emphasis	0 (disabled), 1 (enabled). Default is 1.	—	Programmable Pre-emphasis

5.6.1. Programmable Open Drain

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.

5.6.2. Programmable Bus Hold

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 holds the signal on an I/O pin at its last-driven 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.

The bus-hold circuitry is not available on dedicated clock pins.

For the specific sustaining current for each V_{CCIO} voltage level driven through the resistor and for the overdrive current used to identify the next driven input level, refer to the device datasheet.

Related Information

Intel Cyclone 10 LP Device Datasheet

5.6.3. Programmable Pull-Up Resistor

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

The Intel Cyclone 10 LP 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 the weak pull-up resistor, you cannot use the bus-hold feature.

Note: When the optional DEV_OE signal drives low, all I/O pins remain tri-stated even if the programmable pull-up option is enabled.

5.6.4. 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.

Table 32. Programmable Current Strength Settings for Intel Cyclone 10 LP Devices

The output buffer for each Intel Cyclone 10 LP 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)	
	Column I/O	Row I/O
3.3 V LVCMOS	**2**	
3.3 V LVTTL	**8**, 4	
3.0 V LVTTL/3.0 V LVCMOS	16, 12, **8**, 4	
2.5 V LVTTL/2.5 V LVCMOS	16, 12, **8**, 4	
1.8 V LVTTL/1.8 V LVCMOS	16, 12, 10, **8**, 6, 4, 2	
1.5 V LVCMOS	16, 12, 10, **8**, 6, 4, 2	
1.2 V LVCMOS	12, 10, **8**, 6, 4, 2	10, **8**, 6, 4, 2
SSTL-2 Class I	12, **8**	

continued...

I/O Standard	I_{OH} / I_{OL} Current Strength Setting (mA) (Default setting in bold)	
	Column I/O	Row I/O
SSTL-2 Class II	**16**	
SSTL-18 Class I	12, 10, **8**	
SSTL-18 Class II	**16**, 12	
1.8 V HSTL Class I	12, 10, **8**	
1.8 V HSTL Class II	**16**	
1.5 V HSTL Class I	12, 10, **8**	
1.5 V HSTL Class II	**16**	
1.2 V HSTL Class I	12, 10, **8**	10, **8**
1.2 V HSTL Class II	**14**	—
Differential SSTL-2 Class I	12, **8**	
Differential SSTL-2 Class II	**16**	
Differential SSTL-18 Class I	12, 10, **8**—	
Differential SSTL-18 Class II	12. **16**	
Differential 1.8 V HSTL Class I	12, 10, **8**	
Differential 1.8 V HSTL Class II	**16**	
Differential 1.5 V HSTL Class I	12, 10, **8**	
Differential 1.5 V HSTL Class II	**16**	
Differential 1.2 V HSTL Class I	12, 10, **8**	
Differential 1.2 V HSTL Class II	**14**	
BLVDS	16, **12**, 8	

Default current strength setting in the Intel Quartus Prime software:

- All non-voltage referenced I/O standards—50–Ω OCT without calibration
- SSTL Class I and HSTL Class I I/O standards—50–Ω OCT without calibration
- SSTL Class II and HSTL Class II I/O standards—25–Ω OCT without calibration

Programmable current strength is not available if you use R_S OCT.

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

5.6.5. Programmable Output Slew Rate Control

You have the option of three settings for programmable slew rate control—0, 1, and 2 with 2 as the default setting. Setting 0 is the slow slew rate and 2 is the fast slew rate.

- 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.

Table 33. Programmable Output Slew Rate Control for Intel Cyclone 10 LP Devices

This table lists the I/O standards and current strength settings that support programmable output slew rate control. For I/O standards and current strength settings that do not support programmable slew rate control, the default slew rate setting is 2 (fast slew rate).

I/O Standard	I_{OH} / I_{OL} Current Strength Supporting Slew Rate Control
3.0 V LVTTL/3.0 V LVCMOS	16, 12, 8
2.5 V LVTTL/2.5 V LVCMOS	16, 12, 8
1.8 V LVTTL/1.8 V LVCMOS	16, 12, 10, 8
1.5 V LVCMOS	16, 12, 10, 8
1.2 V LVCMOS	12, 10, 8
SSTL-2 Class I	12, 8
SSTL-2 Class II	16
SSTL-18 Class I	12, 10, 8
SSTL-18 Class II	16, 12
1.8 V HSTL Class I	12, 10, 8
1.8 V HSTL Class II	16
1.5 V HSTL Class I	12, 10, 8
1.5 V HSTL Class II	16
1.2 V HSTL Class I	12, 10, 8
1.2 V HSTL Class II	14
Differential SSTL-2 Class I	12, 8
Differential SSTL-2 Class II	16
Differential SSTL-18	12, 10, 8
Differential 1.8 V HSTL	12, 10, 8
Differential 1.5 V HSTL	12, 10, 8
Differential 1.2 V HSTL	12, 10, 8
BLVDS	16, 12, 8

You can specify the slew rate on a pin-by-pin basis because each I/O pin contains a slew rate control. The slew rate control affects both the rising and falling edges.

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

5.6.6. Programmable IOE Delay

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

Each pin can have a different delay value to ensure signals within a bus have the same delay going into or out of the device.

Each dual-purpose clock input pin provides a programmable delay to the global clock networks.

Table 34. Programmable Delay Chain

Programmable Delays	Intel Quartus Prime Logic Option
Input pin-to-logic array delay	Input delay from pin to internal cells
Input pin-to-input register delay	Input delay from pin to input register
Output pin delay	Delay from output register to output pin
Dual-purpose clock input pin delay	Input delay from dual-purpose clock pin to fan-out destinations

There are two paths in the IOE for an input to reach the logic array. Each of the two paths can have a different delay. This allows you to adjust delays from the pin to the internal logic element (LE) registers that reside in two different areas of the device. You must set the two combinational input delays with the input delay from pin to internal cells logic option in the Intel Quartus Prime software for each path. If the pin uses the input register, one of the delays is disregarded and the delay is set with the input delay from pin to input register logic option in the Intel Quartus Prime software.

The IOE registers in each I/O block share the same source for the preset or clear features. You can program preset or clear for each individual IOE, but you cannot use both features simultaneously. You can also program the registers to power-up high or low after configuration is complete. If programmed to power-up low, an asynchronous clear can control the registers. If programmed to power-up high, an asynchronous preset can control the registers. This feature prevents the inadvertent activation of the active-low input of another device upon power up. If one register in an IOE uses a preset or clear signal, all registers in the IOE must use that same signal if they require preset or clear. Additionally, a synchronous reset signal is available for the IOE registers.

Related Information

- Timing Closure and Optimization chapter, Volume 2: Design Implementation and Optimization, Intel Quartus Prime Handbook
  Provides more information about the input and output pin delay settings.
- Intel Cyclone 10 LP Device Datasheet

5.6.7. PCI Clamp Diode

The Intel Cyclone 10 LP devices are equipped with optional PCI clamp diode that you can enable for the input and output of each I/O pin. You can use this diode to protect I/O pins during voltage overshoot.

The PCI clamp diode is available and enabled by default in the Intel Quartus Prime software for the following I/O standards:

- 3.3 V LVTTL/3.3 V LVCMOS
- 3.0 V LVTTL/3.0 V LVCMOS
- 2.5 V LVTTL/2.5 V LVCMOS
- 3.0 V PCI/PCI-X

Dual-purpose configuration pins support the diode in user mode if you do not use the pins as configuration pins for the selected configuration scheme. For example, if you use the active serial (AS) configuration scheme, you cannot use the clamp diode on the ASD0 and nCS0 pins in user mode. The dedicated configuration pins do not support the on-chip diode.

Note: For the 3.3 V LVTTL, 3.3 V LVCMOS, 3.0 V LVTTL, and 3.0 V LVCMOS I/O standards, you must enable the PCI clamp diode.

5.6.8. Programmable Pre-Emphasis

The differential output voltage (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. Pre-emphasis momentarily boosts the output current during switching to increase the output slew rate.

Pre-emphasis increases the amplitude of the high-frequency component of the output signal. This increase compensates for the frequency-dependent attenuation along the transmission line.

The overshoot introduced by the extra current occurs only during change of state switching. This overshoot increases the output slew rate but 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 64. LVDS Output with Programmable Pre-Emphasis

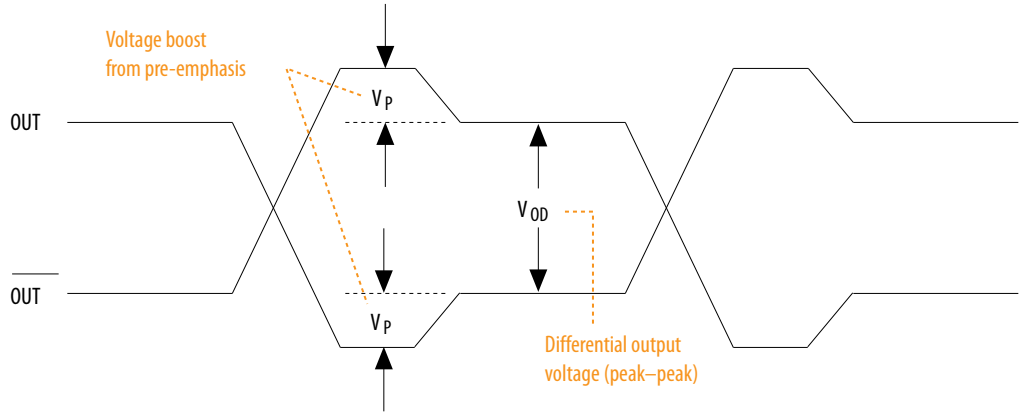


Table 35. Intel Quartus Prime Software Assignment for Programmable Pre-Emphasis

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

5.7. I/O Standards Termination

Voltage-referenced and differential I/O standards requires different termination schemes.

According to JEDEC standards, the following I/O standards do not specify a recommended termination scheme:

- 3.3-V LVTTL
- 3.0 V LVTTL/3.0 V LVCMOS
- 2.5 V LVTTL/2.5 V LVCMOS
- 1.8 V LVTTL/1.8 V LVCMOS
- 1.5 V LVCMOS
- 1.2 V LVCMOS
- 3.0 V PCI/PCI-X

5.7.1. Voltage-Referenced I/O Standards Termination

Voltage-referenced I/O standards require an input reference voltage (V_{REF}) and a termination voltage (V_{TT}). The reference voltage of the receiving device tracks the termination voltage of the transmitting device.

Figure 65. HSTL I/O Standard Termination

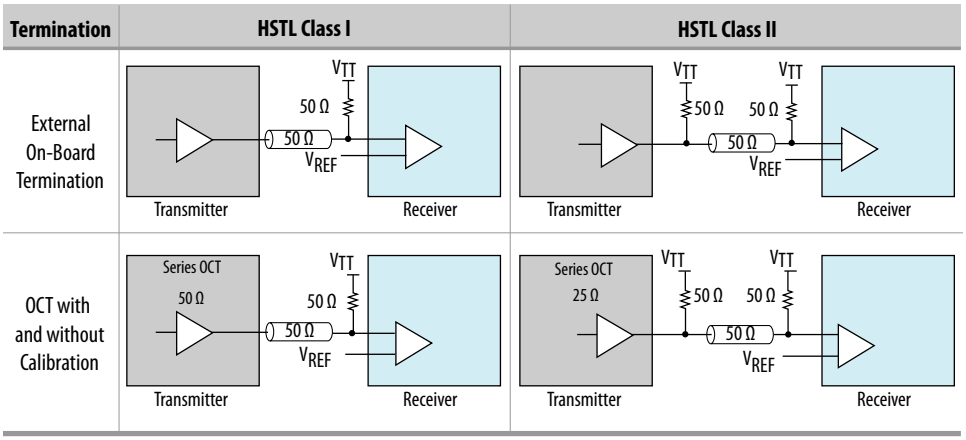
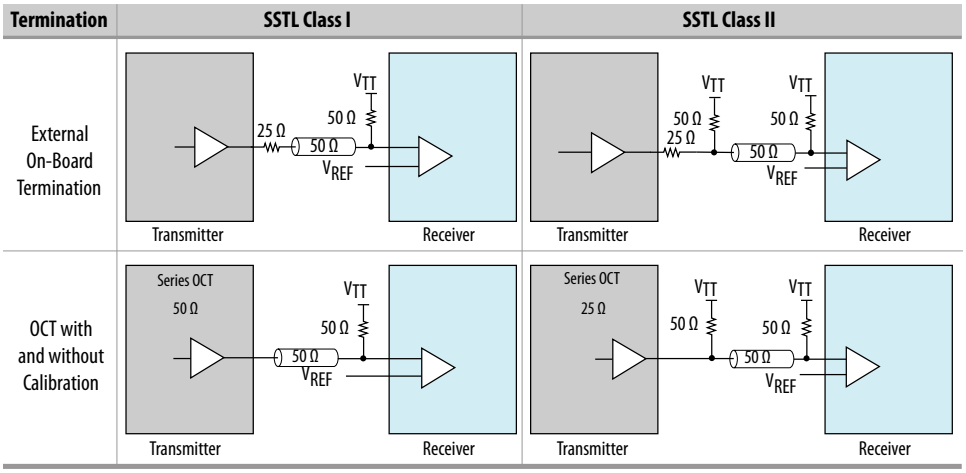


Figure 66. SSTL I/O Standard Termination



5.7.2. Differential I/O Standards Termination

Differential I/O standards typically require a termination resistor between the two signals at the receiver. The termination resistor must match the differential load impedance of the bus.

Figure 67. Differential HSTL I/O Standard Termination

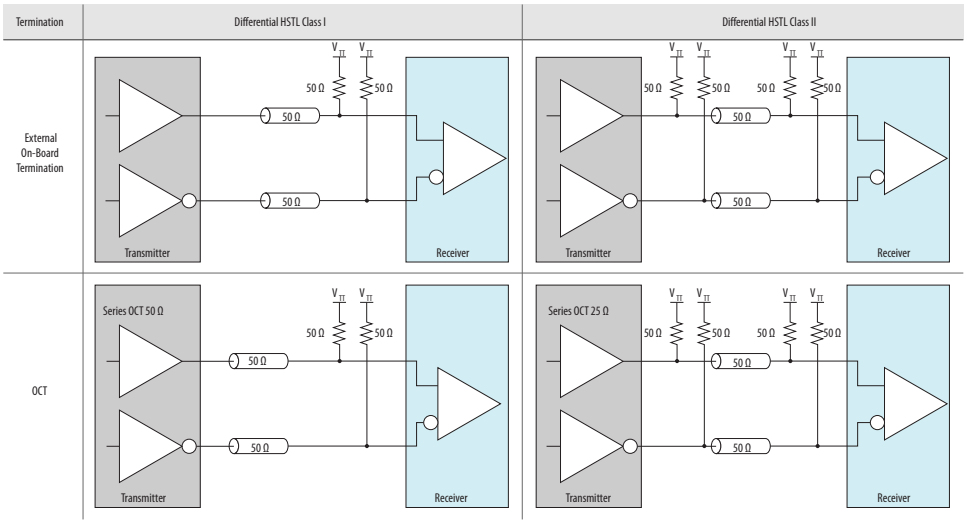
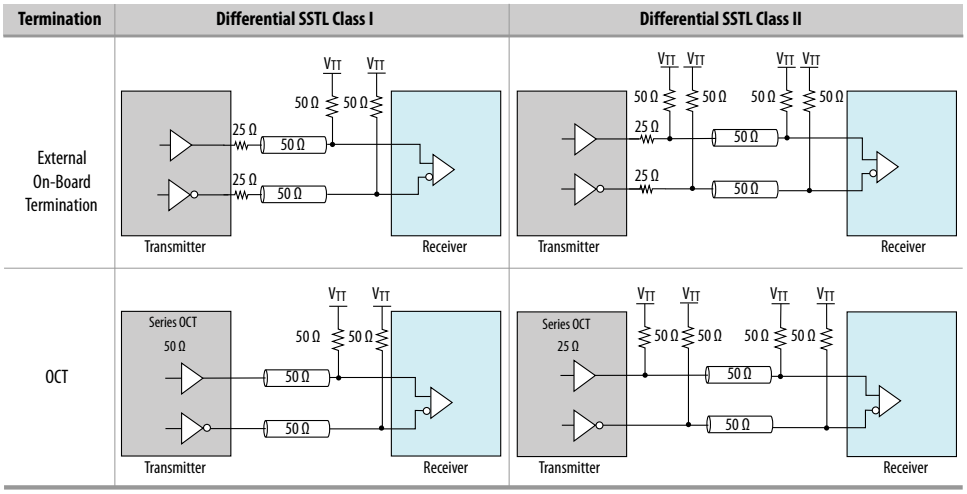


Figure 68. Differential SSTL I/O Standard Termination

Only differential SSTL-2 I/O standard supports Class II output.



Related Information

- Differential SSTL I/O Standard in Intel Cyclone 10 LP Devices on page 106
- Differential HSTL I/O Standard in Intel Cyclone 10 LP Devices on page 107

5.7.3. Intel Cyclone 10 LP On-Chip I/O Termination

The on-chip termination (OCT) block in Intel Cyclone 10 LP devices provides I/O impedance matching and termination capabilities. OCT maintains signal quality, saves board space, and reduces external component costs.

The devices support driver impedance matching to match the impedance of the transmission line, which is typically 25 Ω or 50 Ω. Impedance matching uses the capabilities of the output driver and is subject to a certain degree of variation, depending on the process, voltage, and temperature.

Table 36. OCT Schemes Supported in Intel Cyclone 10 LP Devices

Direction	OCT Schemes	I/O Bank Support
Output	R_S OCT with calibration	Top, bottom, and right I/O banks
	R_S OCT without calibration	All I/O banks

The Intel Cyclone 10 LP devices support serial (R_S) OCT for single-ended output pins and bidirectional pins.

- For bidirectional pins, OCT is active for output only.
- V_{CCIO} and V_{REF} must be compatible for all I/O pins to enable R_S OCT in an I/O bank.
- I/O standards that support different R_S values can reside in the same I/O bank if their V_{CCIO} and V_{REF} do not conflict.
- Dedicated configuration pins and JTAG pins do not support impedance matching or series termination.

Figure 69. R_S OCT with Calibration in Intel Cyclone 10 LP Devices

In this figure, the R_S is the intrinsic impedance of the transistors that make up the I/O buffer.

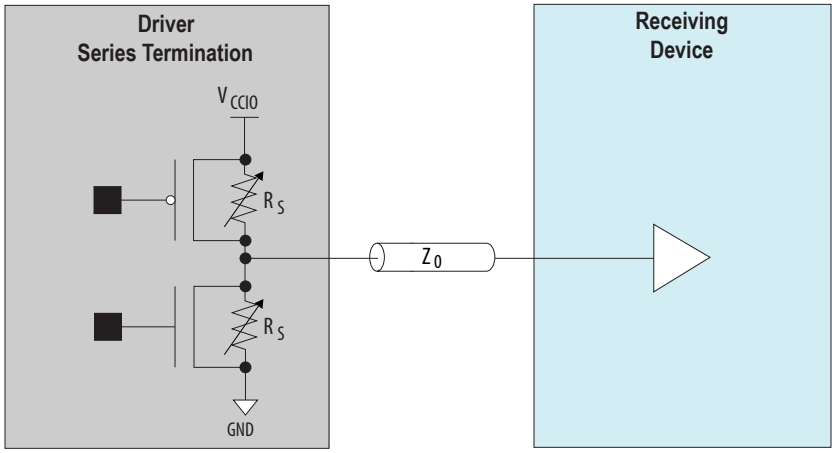
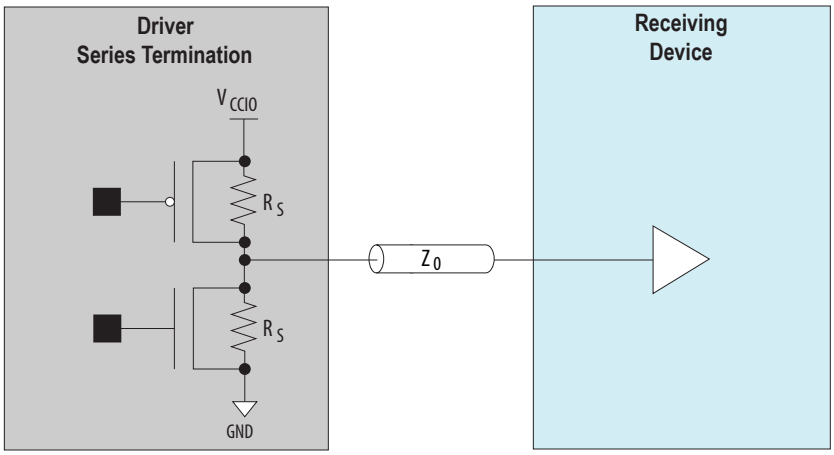


Figure 70. R_S OCT without Calibration in Intel Cyclone 10 LP Devices

In this figure, the R_S is the intrinsic transistor impedance.



Related Information

Intel Cyclone 10 LP Device Datasheet

5.7.3.1. OCT Calibration

The OCT calibration circuit compares the total impedance of the output buffer to the external resistors connected to the RUP and RDN pins. The circuit dynamically adjusts the output buffer impedance until it matches the external resisters.

In Intel Cyclone 10 LP devices, there is one OCT calibration block on each side of the device. The calibration block supports both I/O banks on the side on which it is located:

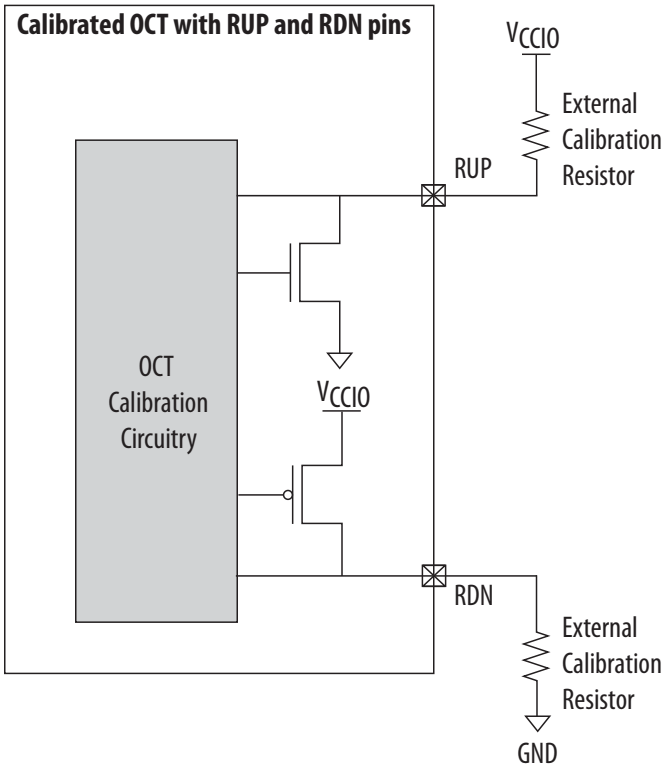
- If you enable OCT calibration for both I/O banks on the same device side, use the same V_{CCIO} on both banks.
- If the V_{CCIO} values of the I/O banks on the same side are not the same, you can use OCT calibration only on the I/O bank that contains the calibration block.

Each OCT calibration block comes with a pair of RUP and RDN pins. During calibration, the RUP and RDN pins are each connected through an external 25 Ω ±1% or 50 Ω ±1% resistor for respective on-chip series termination value of 25 Ω or 50 Ω:

- RUP—connected to VCCIO.
- RDN—connected to GND.

Figure 71. R_S OCT with Calibration Setup

This figure shows the external calibration resistors setup on the RUP and RDN pins and the associated OCT calibration circuitry.



The OCT calibration circuit compares the external resistors to the internal resistance using comparators. The OCT calibration block uses the comparators' output to dynamically adjust buffer impedance.

During calibration, the resistance of the RUP and RDN pins varies. To estimate of the maximum possible current through the external calibration resistors, assume a minimum resistance of 0 Ω on the RUP and RDN pins.

The RUP and RDN pins go to a tri-state condition when calibration is completed or not running. These two pins are dual-purpose I/Os and function as regular I/Os if you do not use the calibration circuit.

Related Information

Intel Cyclone 10 LP I/O Banks Locations on page 86

5.7.3.2. R_S OCT in Intel Cyclone 10 LP Devices

Table 37. Selectable I/O Standards for R_S OCT

This table lists the output termination settings for R_S OCT with and without calibration on different I/O standards.

I/O Standard	Calibrated R_SOCT (Ω)		Uncalibrated R_S OCT (Ω)	
	Column I/O	Row I/O	Column I/O	Row I/O
3.0 V LVTTL/3.0V LVCMOS	25, 50		25, 50	
2.5 V LVTTL/2.5 V LVCMOS	25, 50		25, 50	
1.8 V LVTTL/1.8 V LVCMOS	25, 50		25, 50	
1.5 V LVCMOS	25, 50		25, 50	
1.2 V LVCMOS	25, 50	50	25, 50	50
SSTL-2 Class I	50		50	
SSTL-2 Class II	25		25	
SSTL-18 Class I	50		50	
SSTL-18 Class II	25		25	
1.8 V HSTL Class I	50		50	
1.8 V HSTL Class II	25		25	
1.5 V HSTL Class I	50		50	
1.5 V HSTL Class II	25		25	
1.2 V HSTL Class I	50		50	
1.2 V HSTL Class II	25	—	25	—
Differential SSTL-2 Class I	50		50	
Differential SSTL-2 Class II	25		25	
Differential SSTL-18	50	—	50	—
Differential 1.8 V HSTL	50	—	50	—
Differential 1.5 V HSTL	50	—	50	—
Differential 1.2 V HSTL	50	—	50	—

Related Information

Intel Cyclone 10 LP Device Datasheet

5.8. Intel Cyclone 10 LP High-Speed Differential I/Os and SERDES

The Intel Cyclone 10 LP devices use registers and logic in the core fabric to implement LVDS input and output interfaces.

- For LVDS transmitters and receivers, Intel Cyclone 10 LP devices use the the double data rate I/O (DDIO) registers that reside in the I/O elements (IOE). This architecture improves performance with regards to the receiver input skew margin (RSKM) or transmitter channel-to-channel skew (TCCS).
- For the LVDS serializer/deserializer (SERDES), Intel Cyclone 10 LP devices use logic elements (LE) registers.
- The device uses shift registers, internal phase-locked loops (PLLs), and I/O cells to perform serial-to-parallel conversions on incoming data and parallel-to-serial conversion on outgoing data.

Note: In Intel Cyclone 10 LP devices, the interface configured using the ALTLVDS IP core always sends the MSB of your parallel data first.

Related Information

LVDS SERDES Transmitter/Receiver IP Cores User Guide

5.8.1. High-Speed Differential I/O Interface

The Intel Cyclone 10 LP devices support LVDS, RSDS, mini-LVDS, and PPDS high-speed differential I/O standards.

Differential Input

The Intel Cyclone 10 LP devices features true input buffers for these I/O standards on the top, bottom, and right I/O banks.

Differential Output

The Intel Cyclone 10 LP devices features dedicated differential output buffers:

- The true output drivers are available on the row I/O banks.
- Some of the differential pin pairs (p and n pins) are not located on adjacent pins. In these cases, a power pin is located between the p and n pins.

The Intel Cyclone 10 LP devices provide emulated support for these I/O standards on all columns and row I/O banks:

- Emulated differential output uses a pair of single-ended output pins.
- In the pin pair, the second output pin is programmed as inverted.
- The emulated differential output requires an external resistor network.

5.8.2. Differential I/O Standards Support

Table 38. Differential I/O Standards Support in Intel Cyclone 10 LP I/O Banks

I/O Standard	I/O Bank	External Resistor at Transmitter	TX	RX
LVDS	1, 2, 5, 6	Not required	Yes	Yes
	All	Three resistors		
RSDS	1, 2, 5, 6	Not required	Yes	—
	3, 4, 7, 8	Three resistors		

continued...

I/O Standard	I/O Bank	External Resistor at Transmitter	TX	RX
	All	Single resistor		
Mini-LVDS	1, 2, 5, 6	Not required	Yes	—
	All	Three resistors		
PPDS	1, 2, 5, 6	Not required	Yes	—
	All	Three resistors		
Bus LVDS	All	Single resistor	Yes	Yes
Differential LVPECL	All	—	—	Yes
Differential SSTL-2	All	—	Yes	Yes
Differential SSTL-18	All	—	Yes	Yes
Differential 1.8 V HSTL	All	—	Yes	Yes
Differential 1.5 V HSTL	All	—	Yes	Yes
Differential 1.2 V HSTL	All	—	Yes	Yes

The following conditions apply to the differential I/O standard support:

- The Bus LVDS transmitter and receiver f_{MAX} depend on system topology and performance requirement.
- The Differential LVPECL I/O standard is supported only on dedicated clock input pins.
- The following differential I/O standards are supported only on clock input pins and PLL clock output pins:
  - Differential SSTL-2 and Differential SSTL-18
  - Differential 1.8 V HSTL, Differential 1.5 V HSTL, and Differential 1.2 V HSTL
- PLL clock output pins do not support Class II interface for these differential I/O standards:
  - Differential SSTL-18
  - Differential 1.8 V HSTL, Differential 1.5 V HSTL, and Differential 1.2 V HSTL
- Differential 1.2 V HSTL Class II I/O standard is supported only in column I/O banks.

Related Information

- Intel Cyclone 10 LP I/O Standards Voltage and Pin Support on page 78
  Lists the supported I/O direction, voltages, and pin type for each I/O standard.
- Intel Cyclone 10 LP I/O Standards Support on page 76
  Lists the supported I/O standards and availability of each I/O standard in the device I/O banks.

5.8.2.1. LVDS I/O Standard in Intel Cyclone 10 LP Devices

The LVDS I/O standard is a high-speed, low-voltage swing, low power, and GPIO interface standard.

Intel Cyclone 10 LP devices meet the ANSI/TIA/EIA-644 standard with the following exceptions:

- The maximum differential output voltage (V_{OD}) is increased to 600 mV. The maximum V_{OD} for ANSI specification is 450 mV.
- The input voltage range is reduced to the range of 1.0 V to 1.6 V, 0.5 V to 1.85 V, or 0 V to 1.8 V based on different frequency ranges. The ANSI/TIA/EIA-644 specification supports an input voltage range of 0 V to 2.4 V.

The Intel Cyclone 10 LP left and right I/O banks (row I/Os) support true LVDS transmitters. The top and bottom I/O banks support emulated LVDS transmitters with external resistors.

For the LVDS receiver, you require an external 100 Ω termination resistor between the two signals at the input buffer.

Figure 72. LVDS Interface with True Output Buffer on the Right Side I/O Banks

This figure shows a point-to-point LVDS interface using the true LVDS output and input buffers.

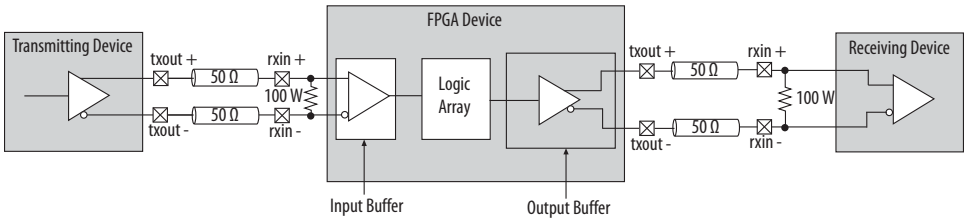
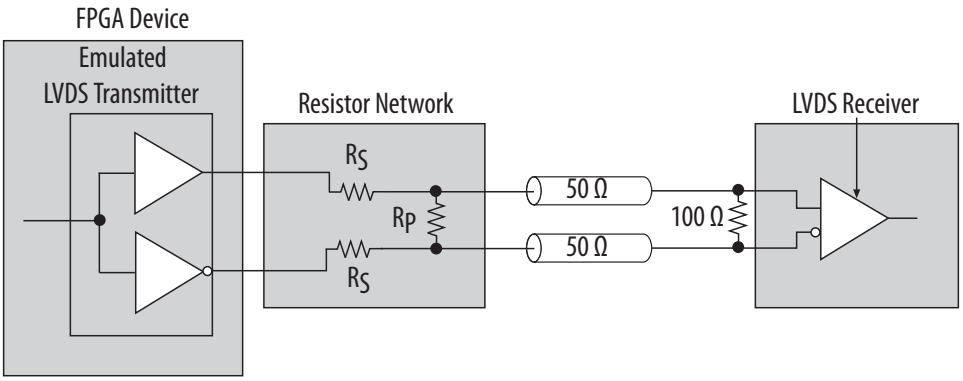


Figure 73. LVDS Interface with External Resistor Network on the Top and Bottom I/O Banks

This figure shows a point-to-point LVDS interface using two single-ended output buffers and external resistors. In the figure, R_S is 120 Ω while R_P is 170 Ω.



Related Information

Intel Cyclone 10 LP Device Datasheet

5.8.2.2. Bus LVDS I/O Standard in Intel Cyclone 10 LP Devices

The Bus LVDS I/O standard is a multipoint I/O standard that supports bidirectional half-duplex communication.

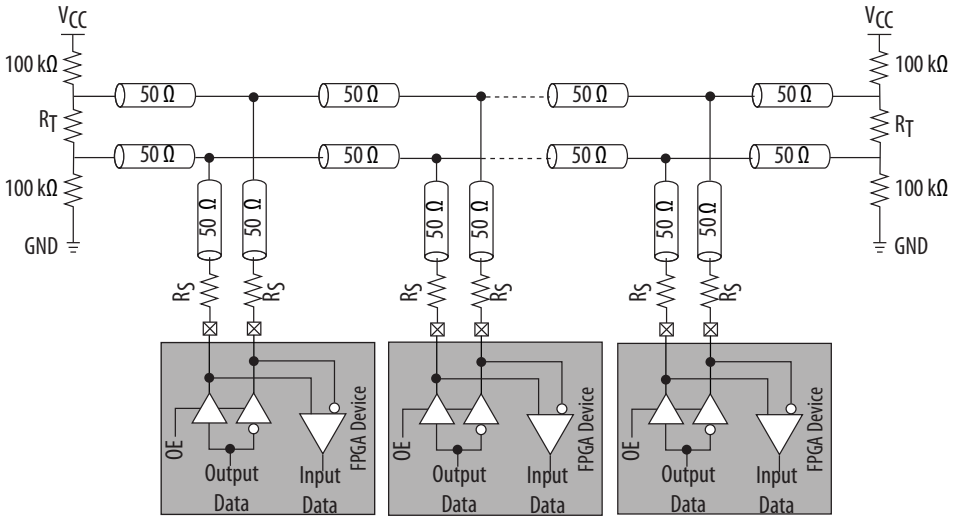
The Intel Cyclone 10 LP top, bottom, and right side I/O banks support the Bus LVDS I/O standard. For the Bus LVDS transmitter, Intel Cyclone 10 LP devices use emulated differential output. For the Bus LVDS receiver, Intel Cyclone 10 LP devices use the true

LVDS input buffer. The transmitter and receiver share the same pins. Intel Cyclone 10 LP devices require an output enable (OE) signal to tristate the output buffers when the LVDS input buffer receives a signal.

The Bus LVDS bidirectional communication requires termination at both ends of the bus.

- The termination resistor (R_T) must match the bus differential impedance, which in turn depends on the loading on the bus. Increasing the load decreases the bus differential impedance.
- With termination at both ends of the bus, termination is not required between the two signals at the input buffer.
- Bus LVDS requires a single series resistor (R_S) at the output buffer to match the impedance to the transmission line. The series resistor affects the voltage swing at the input buffer.
- The maximum data rate achievable depends on many factors.

Figure 74. Typical Bus LVDS Topology with Multiple Intel Cyclone 10 LP Transmitters and Receivers



Note: Intel recommends that you perform simulation using the IBIS model while considering factors such as bus loading, termination values, and output and input buffer location on the bus to ensure that the required performance is achieved.

Related Information

Intel Cyclone 10 LP Device Datasheet

5.8.2.3. RSDS, Mini-LVDS, and PPDS I/O Standard in Intel Cyclone 10 LP Devices

The RSDS, mini-LVDS, and PPDS I/O standards are used in chip-to-chip applications between the timing controller and the column drivers on the display panels such as LCD monitor panels and LCD televisions.

To support RSDS, mini-LVDS, and PPDS output standards, Intel Cyclone 10 LP devices conform to the following specifications:

- National Semiconductor Corporation RSDS Interface Specification
- Texas Instruments mini-LVDS Interface Specification
- National Semiconductor Corporation PPDS Interface Specification

Intel Cyclone 10 LP I/O banks support RSDS, mini-LVDS, and PPDS output standards:

- The right I/O banks support true RSDS, mini-LVDS, and PPDS transmitters.
- The top and bottom I/O banks support emulated RSDS, mini-LVDS, and PPDS transmitters with external resistors.

Figure 75. RSDS, Mini-LVDS, or PPDS Interface with True Output Buffer on the Right Side I/O Banks

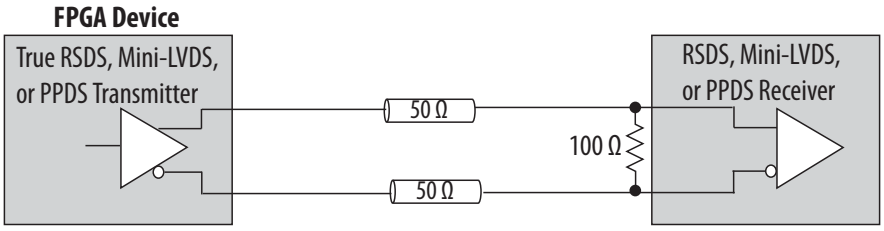
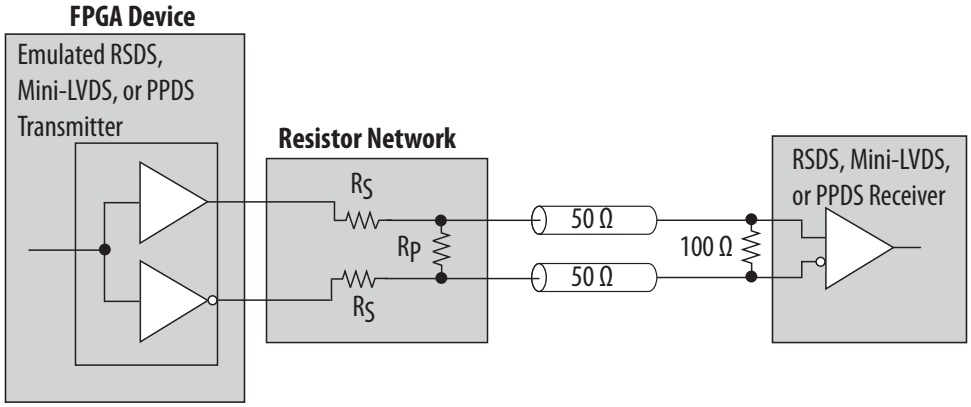


Figure 76. Emulated RSDS, Mini-LVDS, or PPDS Interface with External Resistor Network on the Top and Bottom I/O Banks



A resistor network is required to attenuate the output voltage swing to meet RSDS, mini-LVDS, and PPDS specifications when using emulated transmitters. You can modify the resistor network values to reduce power or improve the noise margin.

Equation 1. Resistor Network Values

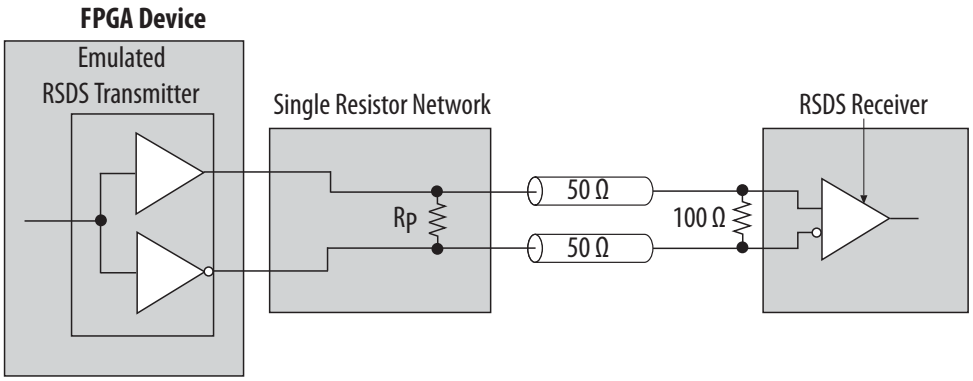
The resistor values must satisfy this equation.

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

Note: Intel recommends that you perform simulations using Intel Cyclone 10 LP devices IBIS models to validate the custom resistor values meet the RSDS, mini-LVDS, or PPDS requirements.

For an RSDS interface, you can use a single external resistor instead of three. The single-resistor solution reduces the external resistor count while still achieving the required RSDS signaling level. However, the performance is lower than with a three-resistor network.

Figure 77. RSDS Interface with Single Resistor Network on the Top and Bottom I/O Banks



Related Information

Intel Cyclone 10 LP Device Datasheet

5.8.2.4. LVPECL I/O Standard in Intel Cyclone 10 LP Devices

The LVPECL I/O standard is a differential interface standard that requires a 2.5 V V_{CCIO}. This standard is used in video graphics, telecommunications, data communications, and clock distribution applications.

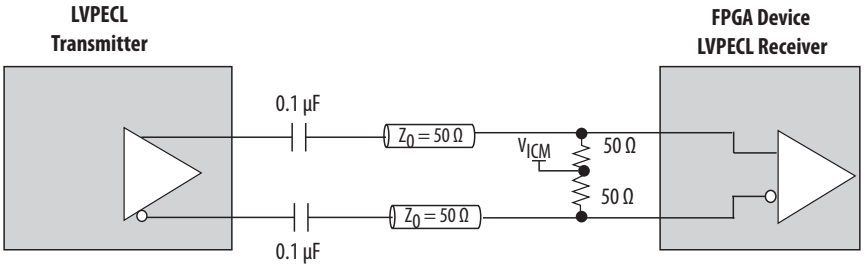
Intel Cyclone 10 LP devices support the LVPECL input standard at the dedicated clock input pins only. The LVPECL receiver requires an external 100 Ω termination resistor between the two signals at the input buffer.

LVPECL requires AC coupling when the common mode voltage of the output buffer is higher than the LVPECL input common mode voltage on the Intel Cyclone 10 LP device.

In the following figures, the 50 Ω resistors at the receiver are external to the device.

Figure 78. LVPECL AC-Coupled Termination

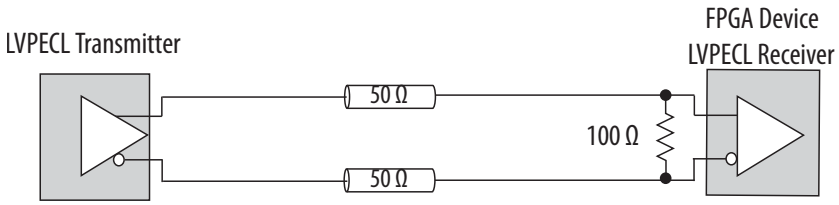
The LVPECL AC-coupled termination is applicable only when you use an Intel FPGA transmitter.



DC-coupled LVPECL is supported if the LVPECL output common mode voltage is in the LVPECL input buffer specification of the Intel Cyclone 10 LP devices.

Figure 79. LVPECL DC-Coupled Termination

The LVPECL DC-coupled termination is applicable only when you use an Intel FPGA transmitter.



Related Information

Intel Cyclone 10 LP Device Datasheet

5.8.2.5. Differential SSTL I/O Standard in Intel Cyclone 10 LP Devices

The Differential SSTL I/O standard is a memory-bus standard used for applications such as high-speed DDR SDRAM interfaces.

Note: Intel does not validate or support any IP that is intended for memory application, such as DDR, in Intel Cyclone 10 LP devices.

Intel Cyclone 10 LP devices support Differential SSTL-2 and Differential SSTL-18 I/O standards. The Differential SSTL output standard is only supported at the PLL#_CLKOUT pins using two single-ended SSTL output buffers (PLL#_CLKOUTp and PLL#_CLKOUTn), with the second output programmed to have opposite polarity.

The Differential SSTL input standard is supported on the GCLK pins only, treating differential inputs as two single-ended SSTL and only decoding one of them.

The Differential SSTL I/O standard requires two differential inputs with an external reference voltage (V_{REF}) and an external termination voltage (V_{TT}) of 0.5 × V_{CCIO} to which termination resistors are connected.

Related Information

- Differential I/O Standards Termination on page 95
  Provides a figure showing the Differential SSTL and Differential HTSL I/O standards termination.

- Intel Cyclone 10 LP Device Datasheet

5.8.2.6. Differential HSTL I/O Standard in Intel Cyclone 10 LP Devices

The Differential HSTL I/O standard is used for the applications designed to operate in 0 V to 1.2 V, 0 V to 1.5 V, or 0 V to 1.8 V HSTL logic switching range.

Intel Cyclone 10 LP devices support Differential 1.8 V HSTL, Differential 1.5 V HSTL, and Differential 1.2 V HSTL I/O standards.

The differential HSTL input standard is available on GCLK pins only, treating the differential inputs as two single-ended HSTL and only decoding one of them.

The differential HSTL output standard is only supported at the PLL#_CLKOUT pins using two single-ended HSTL output buffers (PLL#_CLKOUTp and PLL#_CLKOUTn), with the second output programmed to have opposite polarity.

The Differential HSTL I/O standard requires two differential inputs with an external reference voltage (V_{REF}), as well as an external termination voltage (V_{TT}) of 0.5 × V_{CCIO} to which termination resistors are connected.

Related Information

- Differential I/O Standards Termination on page 95
  Provides a figure showing the Differential SSTL and Differential HTSL I/O standards termination.
- Intel Cyclone 10 LP Device Datasheet

5.8.3. High-Speed I/O Timing Budget

The LVDS I/O standard enables high-speed transmission of data. To take advantage of the fast performance, analyze the timing of high-speed signals. 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. Use the timing parameters provided by IC vendors. High-speed differential data transmission is strongly influenced by board skew, cable skew, and clock jitter.

Intel Cyclone 10 LP devices implement the SERDES in LEs. You must set proper timing constraints to indicate whether the SERDES captures the data as expected or otherwise. You can set the timing constraints using the Timing Analyzer tool in the Intel Quartus Prime software or manually in the Synopsys* Design Constraints (.sdc) file.

Related Information

- Intel Cyclone 10 LP Device Datasheet
- Source-Synchronous Timing Analysis and Timing Constraints, LVDS SERDES Transmitter/Receiver IP Cores User Guide
  Provides more information about high-speed I/O timing budget.

5.8.3.1. Receiver Input Skew Margin

Use RSKM, TCCS, and sampling window (SW) specifications for high-speed source-synchronous differential signals in the receiver data path.

5.8.3.2. RSKM Equation

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

Figure 80. 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.

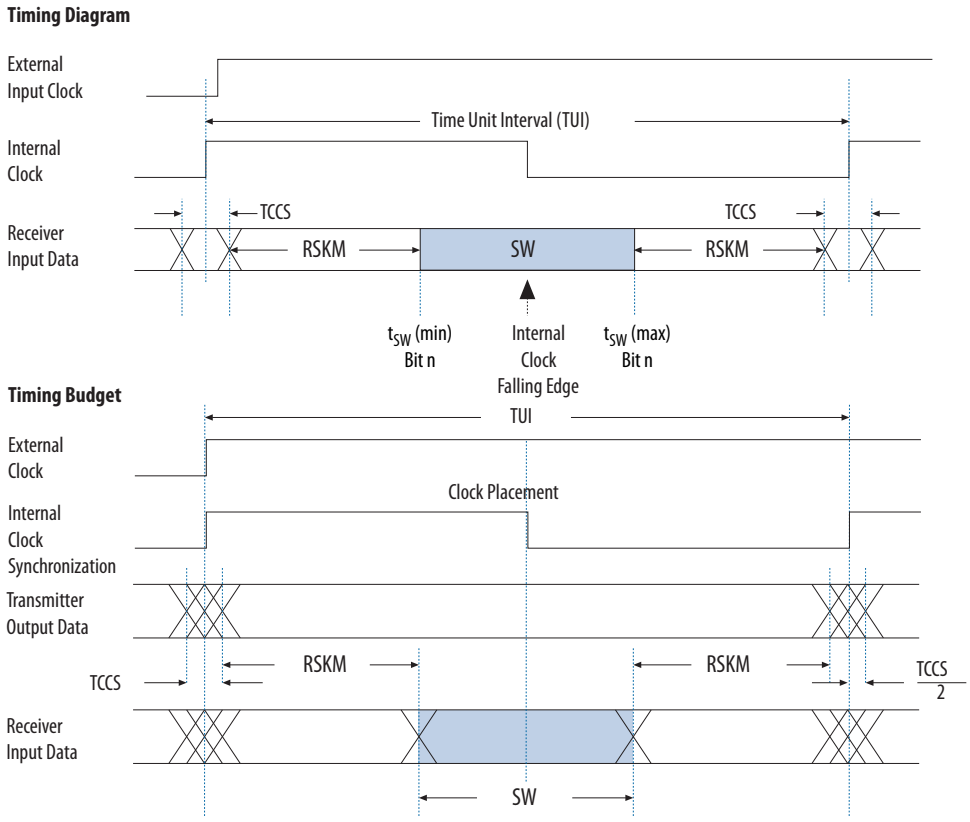
Note: If there is additional board channel-to-channel skew, consider the total receiver channel-to-channel skew (RCCS) instead of TCCS.
Total RCCS = TCCS+board channel-to-channel skew.

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 81. Differential High-Speed Timing Diagram and Timing Budget

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



5.9. Using the I/Os and High Speed I/Os in Intel Cyclone 10 LP Devices

5.9.1. Guideline: Validate Your Pin Placement

Intel recommends that you create an Intel Quartus Prime design, enter your device I/O assignments and compile your design to validate your pin placement.

The Intel Quartus Prime software checks your pin connections with respect to the I/O assignment and placement rules to ensure proper device operation.

These rules depend on device density, package, I/O assignments, voltage assignments and other factors that are not fully described in this chapter.

5.9.2. Guideline: Check for Illegal Pad Placements

Set the DC current sink or source value to **Electromigration Current** assignment on each of the output pins that are connected to the external resistive load.

This setting allows the Intel Quartus Prime software to automatically check for illegal pad placements according to the DC guidelines.

The programmable current strength setting affects the amount of DC current that an output pin can source or sink. Determine if the current strength setting is sufficient for the external resistive load condition on the output pin.

5.9.3. Guideline: Voltage-Referenced I/O Standards Restriction

Each Intel Cyclone 10 LP I/O bank has a V_{REF} bus to accommodate voltage-referenced I/O standards. Each VREF pin is the reference source for its V_{REF} group.

- If you use a V_{REF} group for voltage-referenced I/O standards, connect the VREF pin for that group to the appropriate voltage level.
- If you do not use all the V_{REF} groups in the I/O bank for voltage-referenced I/O standards, you can use the VREF pin in the unused voltage-referenced groups as regular I/O pins.
- The VREF pins are shorted together within the same I/O bank. If you use multiple V_{REF} groups in the same I/O bank, you must power all the VREF pins with the same voltage level.

For example, if you have SSTL-2 Class I input pins in I/O bank 1 and you place them all in the VREFB1N[0] group, you must power VREFB1N[0] with 1.25 V. You can use the remaining VREFB1N[1..3] pins, if available, as I/O pins.

Note: If you use VREF pins as regular I/Os, the VREF pins have higher pin capacitance than regular user I/O pins. This affects the timing if the pins are used as inputs and outputs.

Related Information

- Intel Cyclone 10 LP Device Pin-Outs
- Intel Cyclone 10 LP Device Datasheet

5.9.4. Guideline: Simultaneous Usage of Multiple I/O Standards

Each Intel Cyclone 10 LP I/O bank has its own V_{CCIO} pins and supports only one V_{CCIO} of 1.2 V, 1.5 V, 1.8 V, 2.5 V, 3.0 V, or 3.3 V. An I/O bank can simultaneously support several I/O standards if the I/O standards use the same V_{CCIO} levels for input and output pins.

An I/O bank can simultaneously support several voltage-referenced standards if all the I/O standards use the same V_{REF} and V_{CCIO} values. For example, For examples:

- SSTL-2 and SSTL-18 I/O standards require different V_{REF} values. If you implement both of these I/O standards in the device, you must assign them to I/O pins in different I/O banks.
- If you implement SSTL-2 and 2.5 V LVCMOS I/O standards, both with V_{CCIO} of 2.5 V and V_{REF} of 1.25 V, you can assign them to I/O pins in the same I/O bank.

5.9.5. Guideline: LVTTL or LVCMOS Inputs in Intel Cyclone 10 LP Devices

If you are designing LVTTL/LVCMOS inputs with Intel Cyclone 10 LP devices, follow these guidelines:

- All pins accept input voltage (V_I) up to a maximum limit (3.6 V) stated in the recommended operating conditions in the device datasheet.
- Whenever the input level is higher than the bank's V_{CCIO}, expect higher leakage current.
- The LVTTL or LVCMOS I/O standard input pins can only conform to the V_{IH} and V_{IL} levels according to the bank's voltage level.

If you use an Intel Cyclone 10 LP device as a receiver in a 3.3 V, 3.0 V, or 2.5 V LVTTL or LVCMOS systems, you must manage the overshoot or undershoot to stay within the absolute maximum ratings and the recommended operating conditions. Refer to the device datasheet.

5.9.6. Guideline: Differential Pad Placement

To maintain an acceptable noise level on the V_{CCIO} supply, you must observe single-ended I/O pins placement restrictions in relation to differential pads.

The Intel Quartus Prime software can validate your pin placement and check for illegal pad placements. Refer to the related information.

Related Information

- Guideline: Validate Your Pin Placement on page 109
- Guideline: Check for Illegal Pad Placements on page 109

5.9.7. Guideline: Board Design for Signal Quality

Consider the critical issues of controlled impedance of traces and connectors, differential routing, and termination techniques to get the best performance from Intel Cyclone 10 LP devices.

Use the following general guidelines to improve signal quality:

- Base board designs on controlled differential impedance. Calculate and compare all parameters, such as trace width, trace thickness, and the distance between two differential traces.
- Maintain equal distance between traces in differential I/O standard pairs as much as possible. Routing the pair of traces close to each other maximizes the common-mode rejection ratio (CMRR).
- Longer traces have more inductance and capacitance. These traces must be as short as possible to limit signal integrity issues.
- Place termination resistors as close to receiver input pins as possible.
- Use surface mount components.
- Avoid 90° corners on board traces.
- Use high-performance connectors.
- Design backplane and card traces so that trace impedance matches the impedance of the connector and termination.

- Keep an equal number of vias for both signal traces.
- Create equal trace lengths to avoid skew between signals. Unequal trace lengths result in misplaced crossing points and decrease system margins as the TCCS value increases.
- Limit vias because they cause discontinuities.
- Keep switching transistor-to-transistor logic (TTL) signals away from differential signals to avoid possible noise coupling.
- Do not route TTL clock signals to areas under or above the differential signals.
- Analyze system-level signals.

Related Information

Support Resources: Board Design
Provides board design-related resources for Intel FPGA devices.

5.10. I/O and High Speed I/O in Intel Cyclone 10 LP Devices Revision History

Document Version	Changes
2020.05.21	At the package plan table, added description and related information links that explain how the GPIO and LVDS pins are counted.
2019.01.15	Updated the table listing the programmable current strength to add missing information and to mark the default settings.

Date	Version	Changes
December 2017	2017.12.22	• Updated the section about high-speed I/O timing budget: — Updated the high-speed I/O timing budget topic to clarify that Intel Cyclone 10 LP devices implements SERDES in LEs. — Removed information about obtaining RSKM report in the Intel Quartus Prime software. The software does not support generating RSKM report for Intel Cyclone 10 LP devices. — Removed the topic about assigning input delay to the LVDS receiver. — Added links to section about source-synchronous timing analysis and constraints in the *LVDS SERDES Transmitter / Receiver IP Cores User Guide*.
May 2017	2017.05.08	Initial release.

6. Configuration and Remote System Upgrades

Intel Cyclone 10 LP devices use volatile SRAM cells to store configuration data. You must download the configuration data to the Intel Cyclone 10 LP device each time the device powers up.

Intel Cyclone 10 LP device configuration offers the following features:

- Support for several configuration schemes to suit your requirements
- Ability to decompress configuration data from external configuration device or storage
- Ability to receive configuration data from remote location through several communication protocols

Table 39. Configuration Schemes and Features Supported by Intel Cyclone 10 LP Devices

Configuration Scheme	Configuration Method	Decompression	Remote System Upgrade
Active serial (AS)	Serial configuration device	Yes	Yes
Passive serial (PS)	External host with flash memory	Yes	Yes
	Download cable	Yes	—
Fast passive parallel (FPP)	External host with flash memory	—	Yes
JTAG	External host with flash memory	—	—
	Download cable	—	—

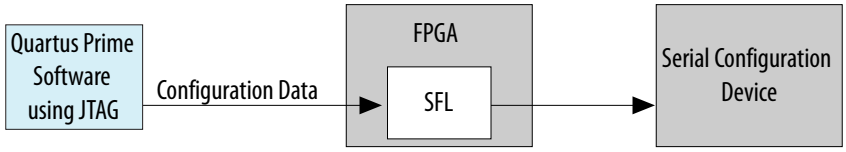
Note: Remote update mode is supported when you use the Remote System Upgrade feature. You can enable or disable remote update mode in the Intel Quartus Prime software.

6.1. Configuration Schemes

Intel offers a wide range of configuration solutions to configure the Intel Cyclone 10 LP devices.

6.1.1. Active Serial (AS) Configuration

Figure 82. High-Level Overview of Serial Configuration Device Programming for the Active Serial Configuration Scheme



In the AS configuration scheme, configuration data is stored in a serial configuration device. These devices are low-cost devices with non-volatile memories that feature a simple four-pin interface and a small form factor.

The four-pin interface consists of these pins:

- Serial clock input (DCLK)
- Serial data output (DATA)
- Active-low chip select (nCS)
- AS data input (ASDI)

Serial configuration devices provide a serial interface to access the configuration data. During device configuration, Intel Cyclone 10 LP devices read the configuration data through the serial interface, decompress the data if necessary, and configure their SRAM cells.

In this scheme, the Intel Cyclone 10 LP device controls the configuration interface. To gain control of the serial configuration device pins, hold the nCONFIG pin low and pull the nCE pin high to cause the device to reset and tri-state the AS configuration pins.

Related Information

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.

AS Configuration Guidelines

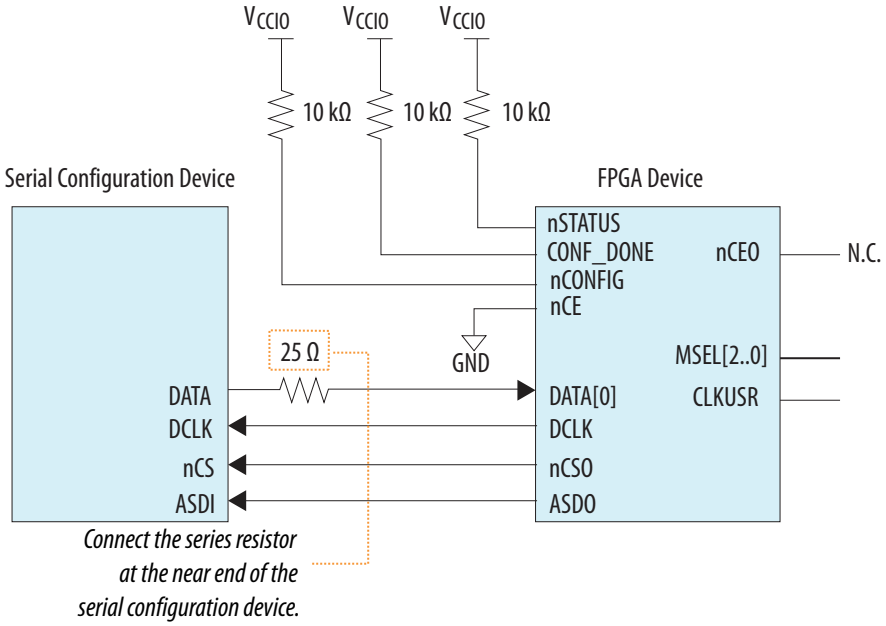
Consider the following guidelines when you configure the Intel Cyclone 10 LP devices:

- Connect the pull-up resistors of the FPGA device to the V_{CC} supply of the bank in which the pin resides.
- You can leave the nCE0 pin unconnected or use it as a user I/O pin when it does not feed the nCE pin of another device.
- The MSEL pin settings vary for different configuration voltage standards and POR time.
- The nCS0 and ASD0 pins are dual-purpose I/O pins. The ASD0 pin also functions as the DATA[1] pin in FPP mode.
- For multi-device configurations, connect the pull-up resistor of the slave FPGA device(s) to the V_{CCIO} supply voltage of I/O bank in which the nCE pin resides.
- Connect the repeater buffers between the master and slave devices of the FPGA device for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation.
- The 50 Ω series resistors are optional if the 3.3 V configuration voltage standard is applied. For optimal signal integrity, connect these 50 Ω series resistors if the 3.0 V configuration voltage standard is applied.

6.1.1.1. Active Serial Single-Device Configuration

For single-device AS configurations, connect the Intel Cyclone 10 LP device to a serial configuration device.

Figure 83. Active Serial Single Device Configuration



The Intel Cyclone 10 LP device uses the DCLK and DATA[1] pins to send operation commands and read address signals to the serial configuration device. The configuration device provides data on its DATA pin, which connects to the DATA[0] input of the Intel Cyclone 10 LP device. All AS configuration pins (DATA[0], DCLK, nCSO, and DATA[1]) have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and driven high by the weak internal pull-up resistors.

Intel Cyclone 10 LP generate the serial clock, DCLK, that provides timing to the serial interface. In the AS configuration scheme, Intel Cyclone 10 LP 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 recommended DCLK frequency supported by the AS configuration scheme is 40 MHz. You can source DCLK using the internal oscillator. The internal oscillator ensures that its maximum frequency is guaranteed to meet serial configuration device specifications. Intel Cyclone 10 LP devices offer the option to select CLKUSR as the external clock source for DCLK. You can change the clock source option in the Intel Quartus Prime software in the **Configuration** tab of the **Device and Pin Options** dialog box.

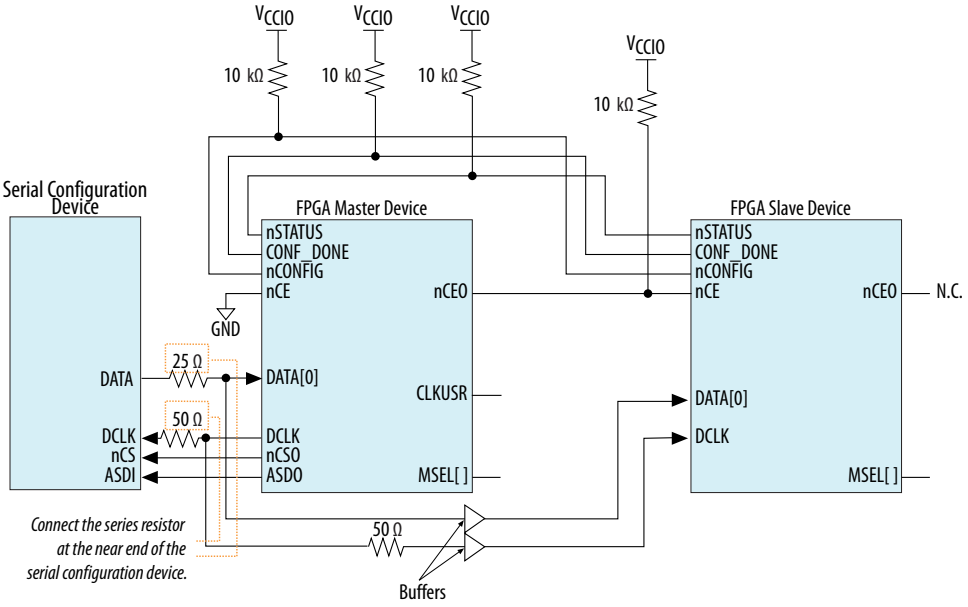
6.1.1.2. Active Serial Multi-Device Configuration

For multi-device AS configurations, connect multiple Intel Cyclone 10 LP devices to a single serial configuration device.

The first device in the chain is the configuration master. Subsequent devices in the chain are configuration slaves.

- When the first device captures all its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain.
- If the last device in the chain is an Intel Cyclone 10 LP device, you can leave the nCEO pin of the last device unconnected or use it as a user I/O pin after configuration.
- The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA[0] pins of each device in the chain are connected.

Figure 84. Active Serial Multi Device Configuration



Note: For multi-devices AS configuration using Intel Cyclone 10 LP devices with 1.0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and DATA[0] line is 3.5 inches.

The nSTATUS and CONF_DONE pins on all target devices are connected to the external pull-up resistors. These pins are open-drain bidirectional pins on the Intel Cyclone 10 LP devices.

- When the first device asserts nCEO (after receiving all its configuration data), it releases its CONF_DONE pin.
- The subsequent devices in the chain hold this shared CONF_DONE line low until they receive their configuration data.
- When all target devices in the chain receive their configuration data and release CONF_DONE, the pull-up resistor drives CONF_DONE line high and all devices simultaneously enter initialization mode.

Note: Although you can cascade Intel Cyclone 10 LP devices, you cannot cascade or chain serial configuration devices.

If the configuration bitstream size exceeds the capacity of a serial configuration device, you must select a larger configuration device, enable the compression feature, or both. When configuring multiple devices, the size of the bitstream is the sum of the individual device's configuration bitstream.

6.1.1.3. Configuring Multiple Intel Cyclone 10 LP Devices with the Same Design

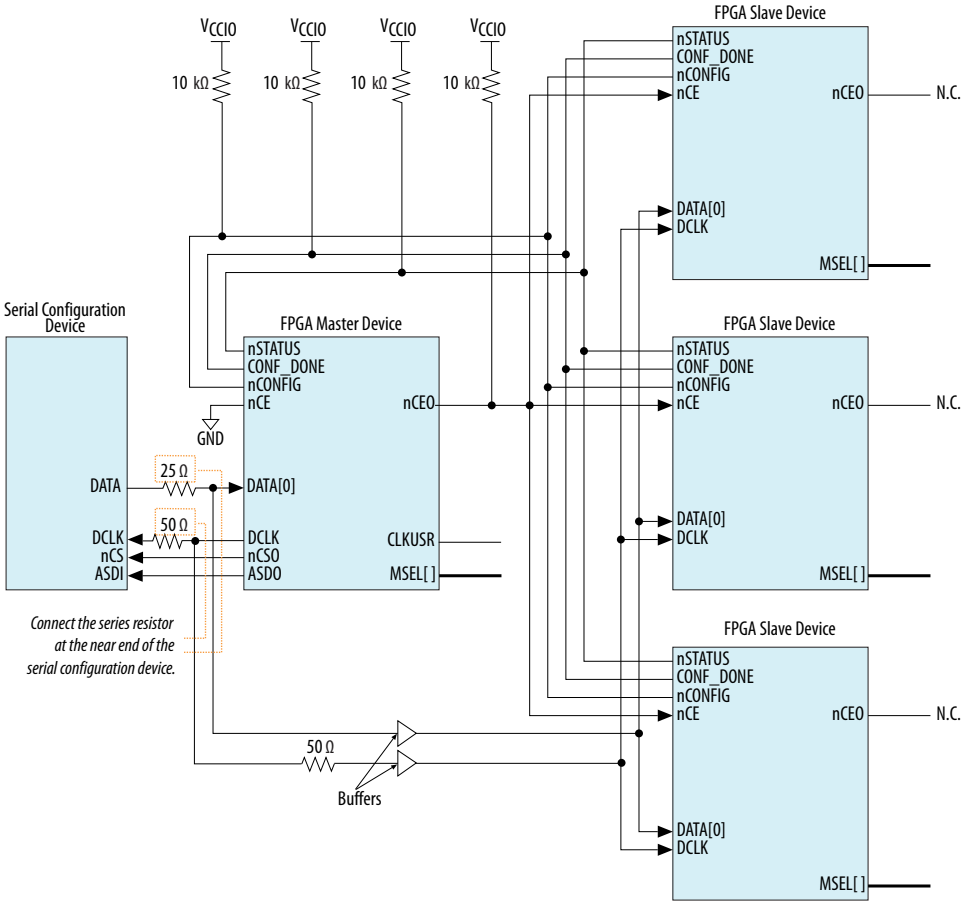
Certain designs require that you configure multiple Intel Cyclone 10 LP devices with the same design through a configuration bitstream, or `.sof`.

You can either use a single `.sof` or multiple `.sof` files.

6.1.1.3.1. Multiple .sof Files

Use both copies of `.sof` stored in the serial configuration device.

Figure 85. Devices Receiving the Same Data with Multiple .sof Files in Multi-Device AS Configuration



Use the first copy to configure the master device of the Intel Cyclone 10 LP device and the second copy to configure all remaining slave devices concurrently. All slave devices must have the same density and package.

- Set the MSEL pins of the master device to select AS configuration.
- Set up the other three slave devices for concurrent configuration and set their MSEL pins to select PS configuration.
- The nCEO pin from the master device drives the nCE input pins on all three slave devices, and the DATA and DCLK pins that connect in parallel to all four devices.
- During the first configuration cycle, the master device reads its configuration data from the serial configuration device while holding nCEO high.
- After completing its configuration cycle, the master device drives nCE low and sends the second copy of the configuration data to all three slave devices, configuring them simultaneously.

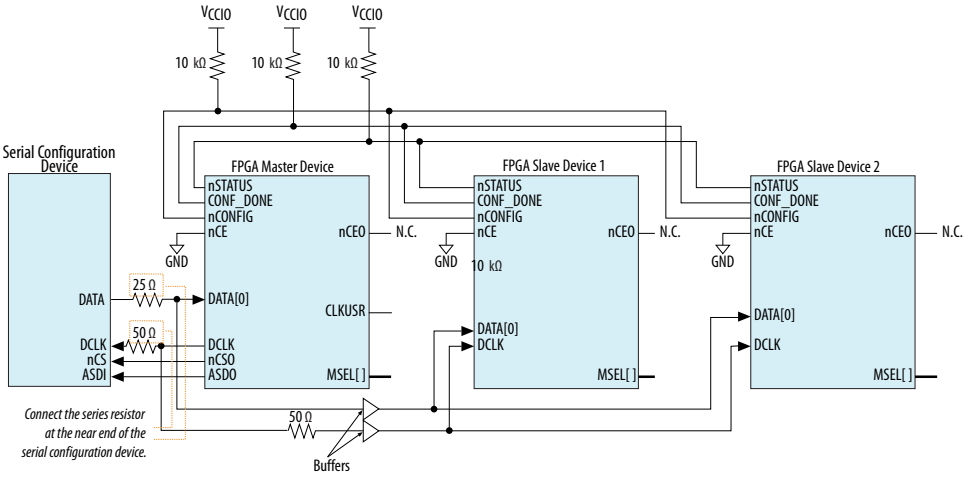
The advantage of the setup is that you can have a different `.sof` for the master device. However, you must configure all the slave devices with the same `.sof`. You can either compress or decompress the `.sof` in this configuration method.

Note: You can still use this method if the master and slave devices use the same `.sof`.

6.1.1.3.2. Single .sof File

This method configures both the master device and slave devices with the same `.sof`.

Figure 86. Devices Receiving the Same Data with a Single .sof File in Multi-Device AS Configuration



The serial configuration device stores one copy of the `.sof`. You must set up one or more slave devices in the chain. In this setup, all the Intel Cyclone 10 LP devices in the chain are connected for concurrent configuration. This reduces the AS configuration time because all the Intel Cyclone 10 LP devices are configured in one configuration cycle.

1. Connect the nCE input pins of all the Intel Cyclone 10 LP devices to GND.
2. Leave the nCEO output pins on all the Intel Cyclone 10 LP devices unconnected or use the nCEO output pins as normal user I/O pins.
3. The DATA and DCLK pins are connected in parallel to all the Intel Cyclone 10 LP devices.

Intel recommends putting a buffer before the DATA and DCLK outputs from the master device to avoid signal strength and signal integrity issues.

- The buffer should not significantly change the DATA-to-DCLK relationships or delay them with respect to other AS signals (ASDI and nCS).
- The buffer must only drive the slave devices to ensure that the timing between the master device and the serial configuration device is unaffected.

This configuration method supports both compressed and uncompressed `.sof`. If the configuration bitstream size exceeds the capacity of a serial configuration device, you can enable the compression feature in the `.sof` or you can select a larger serial configuration device.

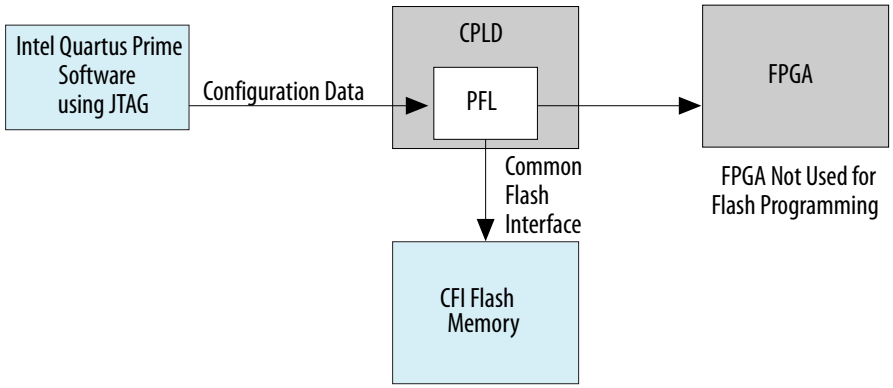
6.1.2. Passive Serial Configuration

The PS configuration scheme uses an external host.

You can use external hosts such as a MAX® V device, microprocessor with flash memory, or a download cable.

In the PS scheme, an external host controls the configuration. Configuration data is clocked into the target Intel Cyclone 10 LP device through DATA[0] at each rising edge of DCLK.

Figure 87. High-Level Overview of Flash Programming for PS Configuration Scheme



Related Information

Parallel Flash Loader IP Core User Guide
Provides more information about how to instantiate the Parallel Flash Loader (PFL) IP core in your design, programming flash memory, and configuring your FPGA from the flash memory.

PS Configuration Connection Guidelines

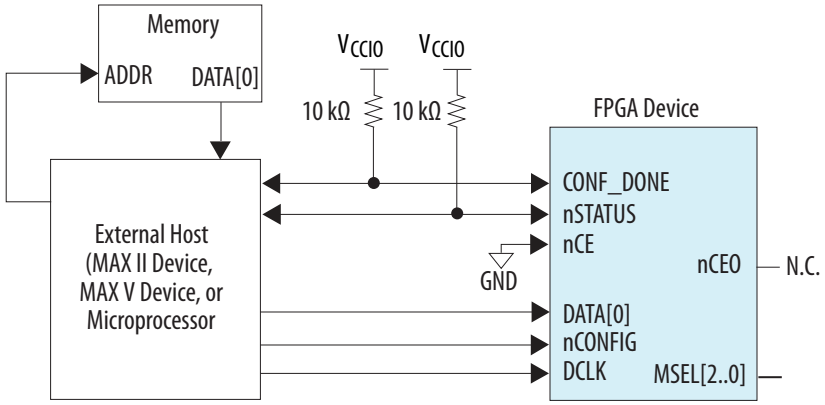
Consider the following guidelines when you configure the Intel Cyclone 10 LP devices:

- Connect the pull-up resistors of the FPGA device to the V_{CC} supply of the bank in which the pin resides.
- You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed the nCE pin of another device.
- The MSEL pin settings vary for different configuration voltage standards and POR time.
- The nCSO and ASDO pins are dual-purpose I/O pins. The ASDO pin also functions as the DATA[1] pin in FPP mode.
- For multi-device configurations, connect the pull-up resistor of the slave FPGA device(s) to the V_{CCIO} supply voltage of I/O bank in which the nCE pin resides.
- Connect the repeater buffers between the master and slave devices of the FPGA device for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation.
- The 50 Ω series resistors are optional if the 3.3 V configuration voltage standard is applied. For optimal signal integrity, connect these 50 Ω series resistors if the 2.5 V or 3.0 V configuration voltage standard is applied.

6.1.2.1. Passive Serial Single-Device Configuration Using an External Host

To configure Intel Cyclone 10 LP device, connect the device to an external host.

Figure 88. Single Device PS Configuration Using an External Host



You can use the 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 `.rbf`, `.hex`, or `.ttf`.

1. The configuration begins when the external host device generates a low-to-high transition on the nCONFIG pin.
2. When nSTATUS is high, the external host device places the configuration data one bit at a time on DATA[0].
3. 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 you must send to the device is:

   ```
   0100-0000 1101-1000 0111-0111 1000-0000 0101-1111
   ```

4. The Intel Cyclone 10 LP device receives configuration data on DATA[0] and clock on DCLK.
5. The configuration data latches onto the device on the rising edge of DCLK.
6. The data continuously clocks into the target device until CONF_DONE goes high and the device enters initialization state.

   Note: Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device.
7. When initialization completes, INIT_DONE releases and goes high. The external host device must be able to detect this low-to-high transition signal, which indicates the device has entered user mode.

   Note: In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

To ensure DCLK and DATA[0] are not left floating at the end of configuration, the external processor must drive them high or low, depending on your board. The DATA[0] pin is available as a user I/O pin after configuration. In the PS scheme, the DATA[0] pin is tri-stated by default in user mode and must be driven by the external host device.

- The configuration clock (DCLK) speed must be below the specified system frequency to ensure correct configuration.
- You can pause configuration by halting DCLK for an indefinite amount of time because there is no maximum DCLK period.

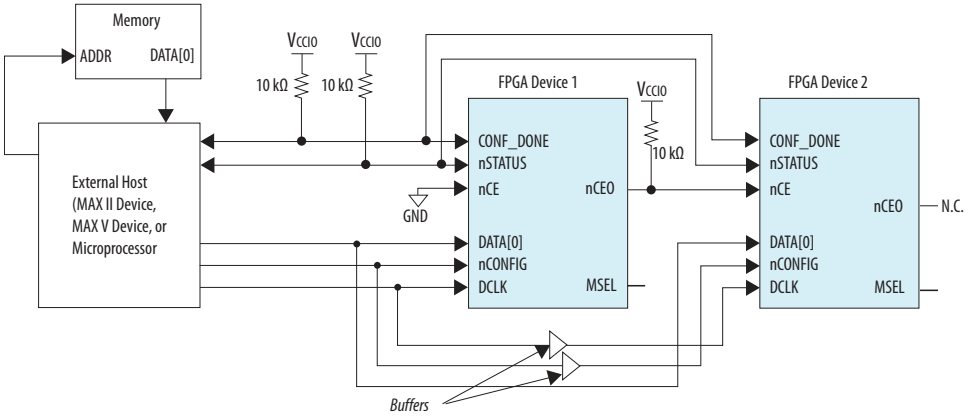
The external host device also monitors CONF_DONE and INIT_DONE to ensure successful configuration.

- The CONF_DONE pin must be monitored by the external device to detect errors and to determine when programming is complete.
- If all configuration data is sent, but CONF_DONE or INIT_DONE has not gone high, the external device must reconfigure the target device.

6.1.2.2. Passive Serial Multi-Device Configuration Using an External Host

To configure multiple devices using an external host, cascade the Intel Cyclone 10 LP devices.

Figure 89. Multi-Device PS Configuration Using an External Host



1. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device prompting it to begin configuration.
2. The second device in the chain begins configuration in one clock cycle. The destinations of data transfer is transparent to the external host device. The nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain. To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that DCLK and DATA lines are buffered.
3. All devices initialize and enter user mode at the same time because all the CONF_DONE pins are tied together.
4. If any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain because all nSTATUS and CONF_DONE pins are tied together. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

You can have multiple devices that contain the same configuration data in your system. To support this configuration scheme, ensure these conditions:

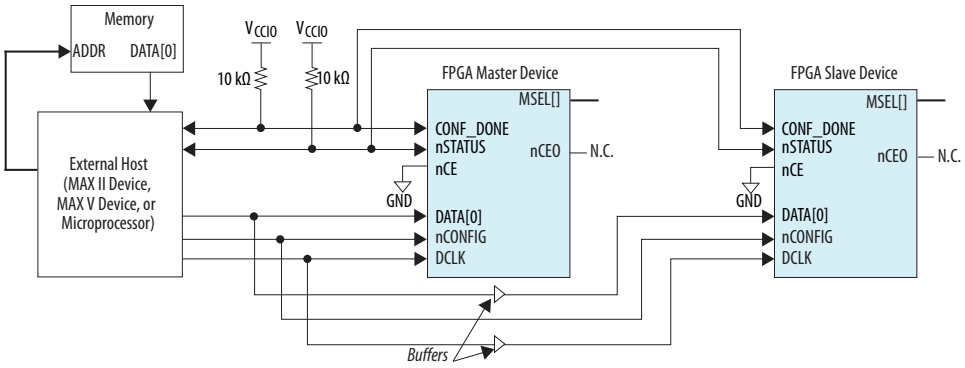
- All device nCE inputs are tied to GND, while the nCEO pins are left floating.
- nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain.

  Note: To ensure signal integrity and avoid clock skew problems, configuration signals may require buffering. Ensure that the DCLK and DATA lines are buffered.
- All devices must be of the same density and package.
- All devices start and complete configuration at the same time.

6.1.2.2.1. Passive Serial Multi Devices Using Same Configuration Data

You can have multiple devices that contain the same configuration data in your system.

Figure 90. Multiple Device PS Configuration When Both Devices Receive the Same Configuration 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. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain.

To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that the DCLK and DATA lines are buffered.

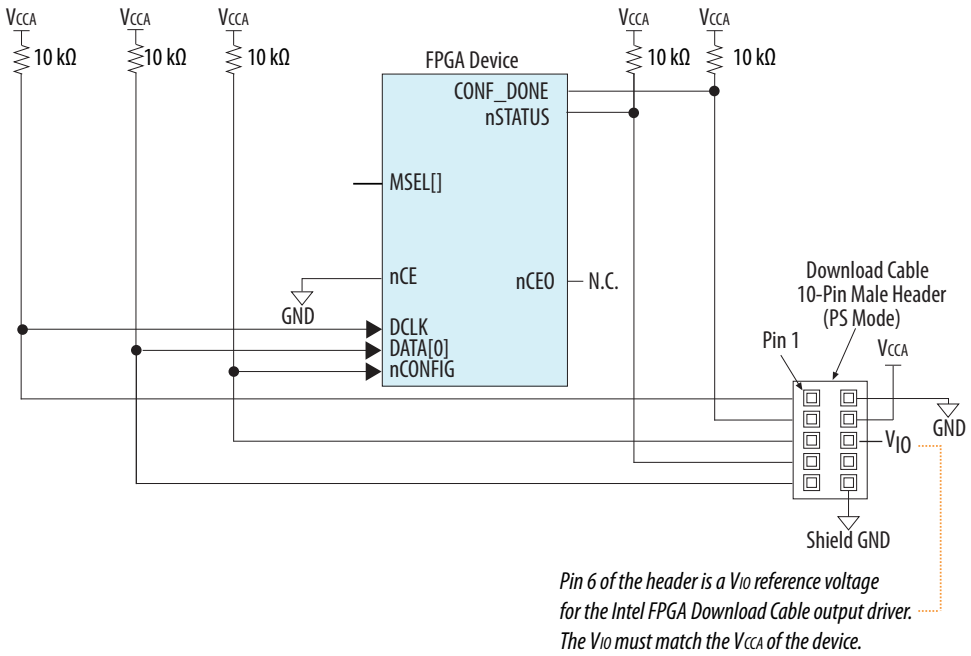
6.1.2.3. Passive Serial Single-Device Configuration Using a Download Cable

To configure Intel Cyclone 10 LP device, connect the device to a download cable.

The download cable could be an Intel FPGA download cable or Intel FPGA Ethernet cable.

Note: Power up the V_{CC} of the Intel FPGA download cable with a 2.5 V supply from V_{CCA}. Third-party programmers must switch to 2.5 V.

Figure 91. Single Device PS Configuration Using an Intel Download Cable



Note: The pull-up resistors on DATA[0] and DCLK are only required if the download cable is the only configuration scheme used on your board. This is to ensure that DATA[0] and DCLK are not left floating after configuration. For example, if you also use a configuration device, the pull-up resistors on DATA[0] and DCLK are not required.

1. The configuration begins when the external host transfers data from a storage device to the Intel Cyclone 10 LP through the download cable.
2. The programming hardware or download cable then places the configuration data one bit at a time on the DATA[0] pin of the device.
3. The configuration data clocks into the target device until CONF_DONE goes high.

   Note: The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

When you use the Intel FPGA download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Intel Quartus Prime software if an error occurs. The **Enable user-supplied start-up clock (CLKUSR)** option has no effect on device initialization, because this option is disabled in the .sof when programming the device with the Intel Quartus Prime Programmer and download cable. If you turn on the **Enable user-supplied start-up clock (CLKUSR)** option, you do not have to provide a clock on CLKUSR when you configure the device with the Intel Quartus Prime Programmer and a download cable.

You can also use the Intel FPGA download cable to configure multiple Intel Cyclone 10 LP device configuration pins. These pins are connected to every device in the chain:

- nCONFIG
- nSTATUS
- DCLK
- DATA[0]
- CONF_DONE

All devices in the chain utilize and enter user mode at the same time because all the CONF_DONE pins are tied together. The entire chain halts configuration if any device detects an error because the nSTATUS pins are tied together.

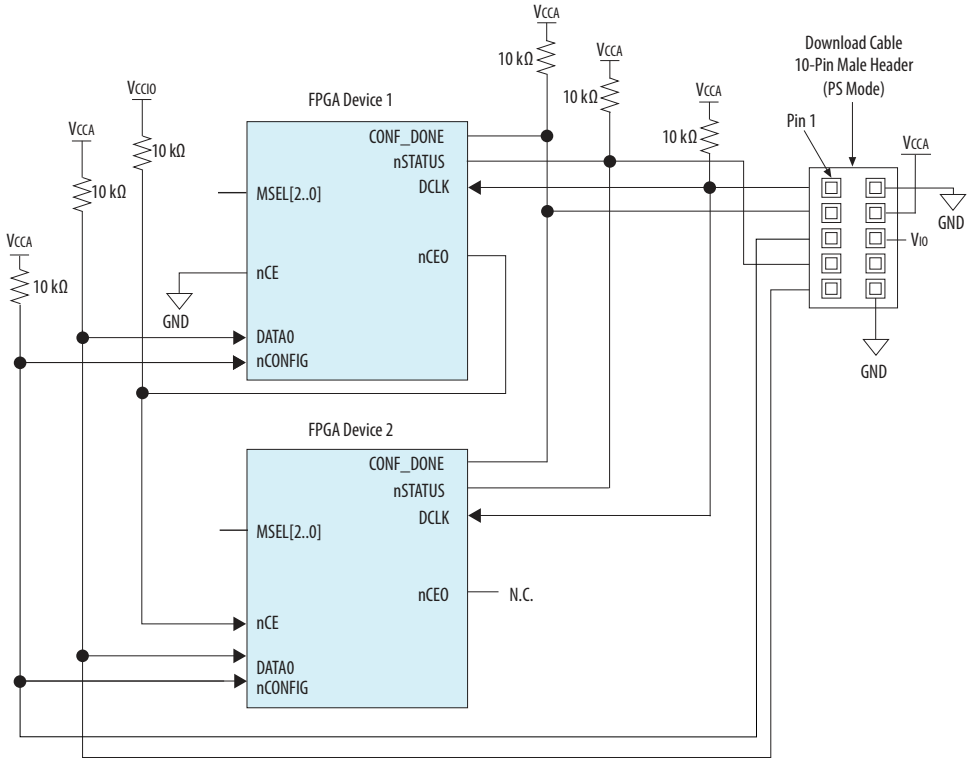
6.1.2.4. Passive Serial Multi-Device Configuration Using a Download Cable

You can also use a download cable to configure multiple Intel Cyclone 10 LP device configuration pins.

The download cable could be an Intel FPGA download cable or Intel FPGA Ethernet cable.

Note: Power up the V_{CC} of the Intel FPGA download cable with a 2.5-V supply from V_{CCA}. Third-party programmers must switch to 2.5 V.

Figure 92. Multiple Device PS Configuration Using an Intel FPGA Download Cable



Note: The pull-up resistors on DATA[0] and DCLK are only required if the download cable is the only configuration scheme used on your board. This is to ensure that DATA[0] and DCLK are not left floating after configuration. For example, if you also use a configuration device, the pull-up resistors on DATA[0] and DCLK are not required.

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.

6.1.3. Fast Passive Parallel Configuration

The FPP configuration scheme uses an external host, such as a microprocessor, MAX II device, or MAX V device. This scheme allows for a faster configuration time.

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 Cyclone 10 LP device.

Note:

1. Two DCLK falling edges are required after the CONF_DONE pin goes high to begin the initialization of the device.
2. The FPP configuration is not supported in E144 package of Intel Cyclone 10 LP devices.
3. Intel Cyclone 10 LP devices do not support enhanced configuration devices for FPP configuration.

Related Information

Parallel Flash Loader IP Core User Guide
Provides more information about how to instantiate the Parallel Flash Loader (PFL) IP core in your design, programming flash memory, and configuring your FPGA from the flash memory.

FPP Configuration Guidelines

Consider the following guidelines when you configure the Intel Cyclone 10 LP devices:

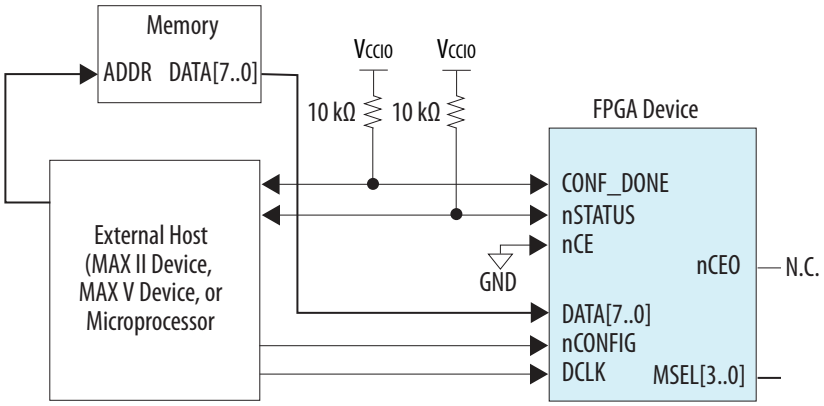
- Connect the pull-up resistors of the FPGA device to the V_{CC} supply of the bank in which the pin resides.
- You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed the nCE pin of another device.
- The MSEL pin settings vary for different configuration voltage standards and POR time.
- The nCSO and ASDO pins are dual-purpose I/O pins. The ASDO pin also functions as the DATA[1] pin in FPP mode.

- For multi-device configurations, connect the pull-up resistor of the slave FPGA device(s) to the V_{CCIO} supply voltage of I/O bank in which the nCE pin resides.
- Connect the repeater buffers between the master and slave devices of the FPGA device for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation.
- The 50 Ω series resistors are optional if the 3.3 V configuration voltage standard is applied. For optimal signal integrity, connect these 50 Ω series resistors if the 2.5 V or 3.0 V configuration voltage standard is applied.

6.1.3.1. Fast Passive Parallel Single-Device Configuration

To configure an Intel Cyclone 10 LP device, connect the device to an external host.

Figure 93. Single Device FPP Configuration Using an External Host



1. The configuration begins when nSTATUS releases and the Intel Cyclone 10 LP device is ready to receive configuration data.
2. When nSTATUS is high, the external host device places the configuration data one byte at a time on DATA[7..0].
3. The Intel Cyclone 10 LP device receives configuration data on DATA[7..0] and clock on DCLK.
4. The configuration data latches onto the device on the rising edge of DCLK.
5. The data continuously clocks into the target device until CONF_DONE goes high.

   Note: CONF_DONE goes high one byte early in FPP configuration mode. The last byte is required for serial configuration (AS and PS) modes.
6. When initialization completes, INIT_DONE releases and goes high. The external host device must be able to detect this low-to-high transition signal, which indicates the device has entered user mode.

To ensure DCLK and DATA[0] are not left floating at the end of configuration, the MAX V device must drive them high or low, depending on your board. The DATA[0] pin is available as a user I/O pin after configuration.

In the FPP scheme, the DATA[0] pin is tri-stated by default in user mode and must be driven by the external host device. You change this default option by selecting the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box in the Intel Quartus Prime software.

- The configuration clock (DCLK) speed must be below the specified system frequency to ensure correct configuration.
- You can pause configuration by halting DCLK for an indefinite amount of time because there is no maximum DCLK period.

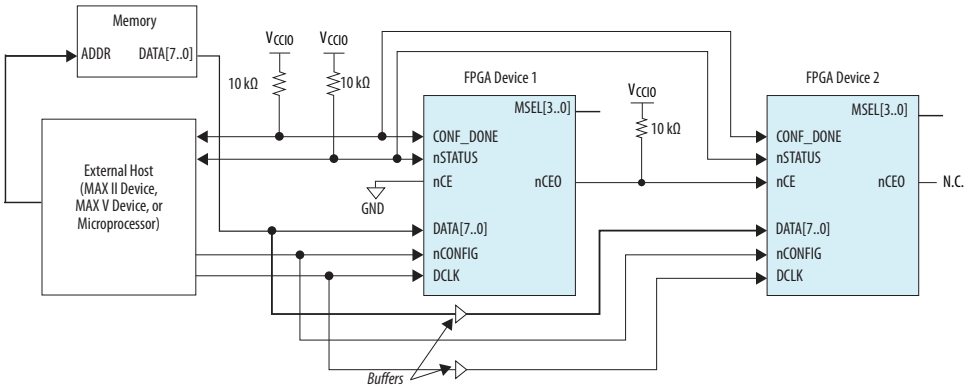
The external host device also monitors CONF_DONE and INIT_DONE to ensure successful configuration.

- The CONF_DONE pin must be monitored by the external device to detect errors and to determine when programming is complete.
- If all configuration data is sent, but CONF_DONE or INIT_DONE has not gone high, the external device must reconfigure the target device.

6.1.3.2. Fast Passive Parallel Multi-Device Configuration

To configure multiple devices using an external host, cascade the Intel Cyclone 10 LP devices.

Figure 94. Multi-Device FPP Configuration Using an External Host



1. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device prompting it to begin configuration.
2. The second device in the chain begins configuration in one clock cycle. The destinations of data transfer is transparent to the external host device. The nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE configuration pins are

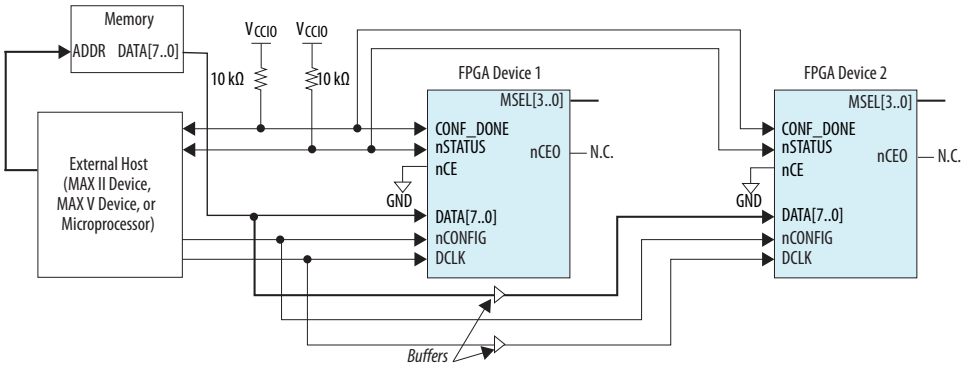
connected to every device in the chain. To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that DCLK and DATA lines are buffered.

3. All devices initialize and enter user mode at the same time because all the CONF_DONE pins are tied together.
4. If any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain because all nSTATUS and CONF_DONE pins are tied together. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

6.1.3.2.1. Fast Passive Parallel Multi Devices Using Same Configuration Data

You can have multiple devices that contain the same configuration data in your system.

Figure 95. Multiple Device FPP Configuration When Both Devices Receive the Same Configuration 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. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain.

To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that the DCLK and DATA lines are buffered.

6.1.4. JTAG Configuration

In Intel Cyclone 10 LP devices, JTAG instructions take precedence over other configuration schemes.

JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration in Intel Cyclone 10 LP devices during PS configuration, PS configuration terminates and JTAG configuration begins. If the MSEL pins are set to AS mode, the Intel Cyclone 10 LP device does not transmit a DCLK signal when JTAG configuration takes place.

JTAG has developed a specification for boundary-scan testing (BST). The BST architecture offers the capability to efficiently test components on PCBs with tight lead spacing. The BST architecture can test pin connections without using physical test probes and capture functional data while a device is normally operating. You can also use the JTAG circuitry to shift configuration data into the device.

The Intel Quartus Prime software generates an `.sof` that you can use for JTAG configuration using a download cable in the Intel Quartus Prime software programmer.

Table 40. Dedicated JTAG Pins

Pin Name	Pin Type	Description
TDI	Test data input	• Serial input pin for instructions, and test and programming data. • Data shifts in on the rising edge of TCK. • If the JTAG interface is not required on the board, JTAG circuitry is disabled by connecting this pin to V_{CC}. • TDI pin has weak internal pull-up resistors (typically 25kΩ).
TDO	Test data output	• Serial data output pin for instructions, and test and programming data. • Data shifts out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device. • If the JTAG interface is not required on the board, the JTAG circuitry is disabled by leaving this pin unconnected.
TMS	Test mode select	• Input pin that provides the control signal to determine the transitions of the TAP controller state machine. • Transitions in the state machine occur on the rising edge of TCK. Therefore, TMS must be set up before the rising edge of TCK. TMS is evaluated on the rising edge of TCK. • If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to V_{CC}. • TMS pin has weak internal pull-up resistors (typically 25 kΩ).
TCK	Test clock input	• The clock input to the BST circuitry. • Some operations occur at the rising edge, while others occur at the falling edge. • If the JTAG interface is not required on the board, the JTAG circuitry is disabled by connecting this pin to GND. • The TCK pin has an internal weak pull-down resistor.

You can download data to the device through the Intel FPGA download cable or the Intel FPGA Ethernet cable during JTAG configuration. Configuring devices with a cable is similar to programming devices in-system.

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 Cyclone 10 LP decompression if you are configuring your Intel Cyclone 10 LP device using JTAG-based configuration.

JTAG Configuration Connection Guidelines

Consider the following guidelines when you configure the Intel Cyclone 10 LP devices:

- Connect the pull-up resistors of the FPGA device to the V_{CC} supply of the bank in which the pin resides.
- You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed the nCE pin of another device.
- The MSEL pin settings vary for different configuration voltage standards and POR time.
- The nCSO and ASDO pins are dual-purpose I/O pins. The ASDO pin also functions as the DATA[1] pin in FPP mode.
- For multi-device configurations, connect the pull-up resistor of the slave FPGA device(s) to the V_{CCIO} supply voltage of I/O bank in which the nCE pin resides.
- Connect the repeater buffers between the master and slave devices of the FPGA device for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation.
- The 50 Ω series resistors are optional if the 3.3 V configuration voltage standard is applied. For optimal signal integrity, connect these 50 Ω series resistors if the 2.5 V or 3.0 V configuration voltage standard is applied.

6.1.4.1. Configuring Intel Cyclone 10 LP Devices with the JRunner Software Driver

The JRunner software driver allows you to configure Intel Cyclone 10 LP devices through the Intel FPGA download cable in JTAG mode.

The supported programming input file is in `.rbf` format. The JRunner software driver also requires a Chain Description File (`.cdf`) generated by the Intel Quartus Prime software. The JRunner software driver is specifically for embedded JTAG configuration. You can customize the code to make it run on your embedded platform.

Note: The `.rbf` used by the JRunner software driver cannot be a compressed `.rbf` because the JRunner software driver uses JTAG-based configuration. During JTAG-based configuration, the real-time decompression feature is not available.

6.1.4.2. Configuring Intel Cyclone 10 LP Devices with Jam STAPL

Jam* STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface

Jam STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL is a freely licensed open standard. The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

6.1.4.3. JTAG Single-Device Configuration

To configure a single device in a JTAG chain, the programing 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 uses the CONF_DONE pin to verify the completion of the configuration process through the JTAG port:

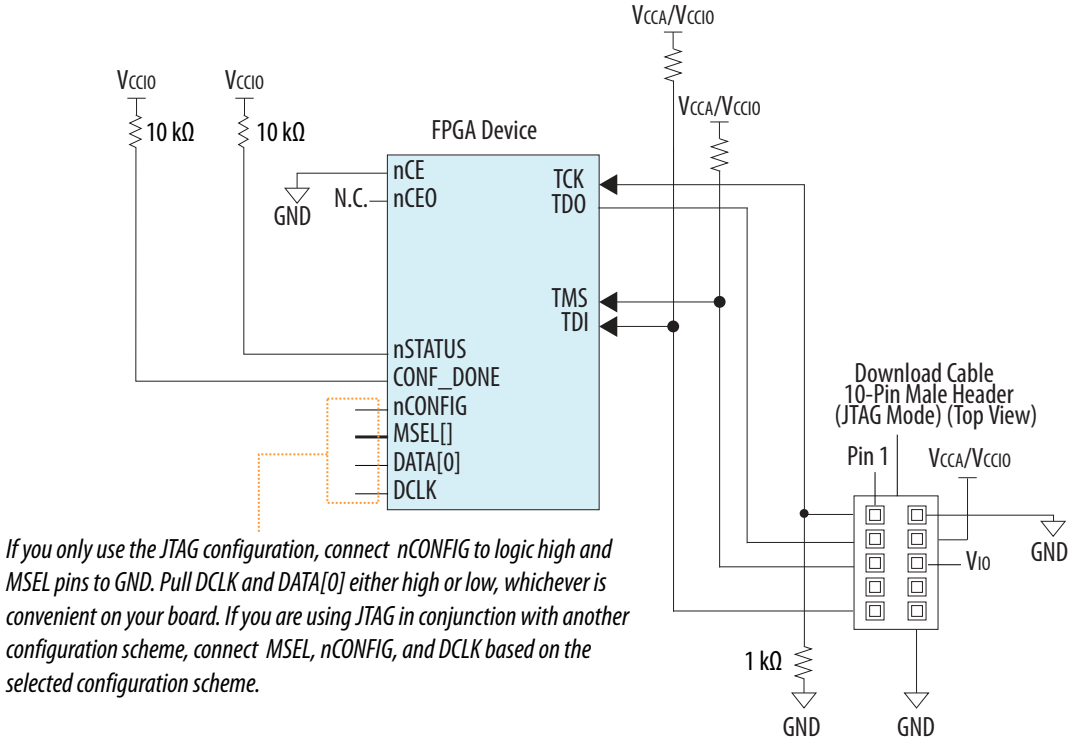
- CONF_DONE pin is low—indicates that configuration has failed.
- CONF_DONE pin is high—indicates that configuration was successful.

After the configuration data is transmitted serially using the JTAG TDI port, the TCK port clocks additional cycles to perform device initialization.

To configure Intel Cyclone 10 LP device using an Intel FPGA download cable, connect the device as shown in the following figure.

- For device using V_{CCIO} of 2.5, 3.0, and 3.3 V, all I/O inputs must maintain a maximum AC voltage of 4.1 V because JTAG pins do not have the internal PCI clamping diodes to prevent voltage overshoot. You must power up the V_{CC} of the download cable with a 2.5-V supply from V_{CCA}
- For device using V_{CCIO} of 1.2 V, 1.5 V, and 1.8 V, you can power up the V_{CC} of the download cable with the supply from V_{CCIO}.

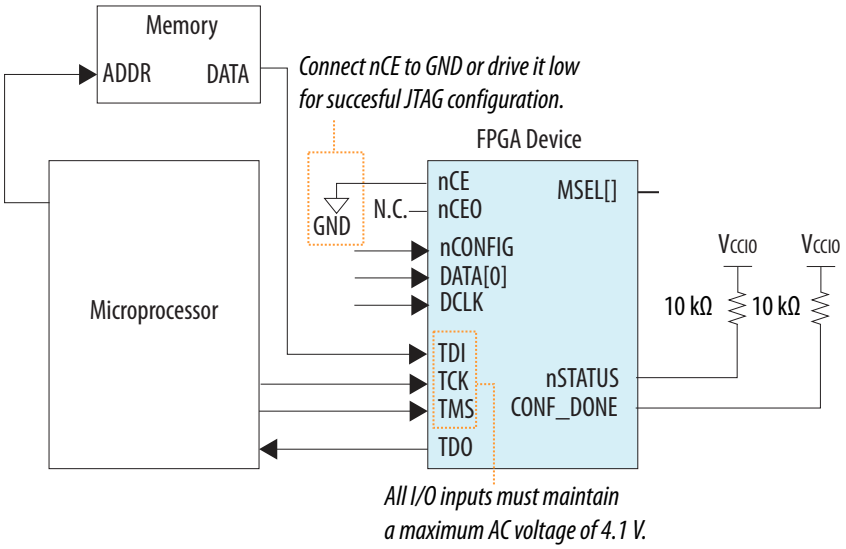
Figure 96. JTAG Configuration of a Single Device Using a Download Cable



If you only use the JTAG configuration, connect nCONFIG to logic high and MSEL pins to GND. Pull DCLK and DATA[0] either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, connect MSEL, nCONFIG, and DCLK based on the selected configuration scheme.

To configure Intel Cyclone 10 LP device using a microprocessor, connect the device as shown in the following figure.

Figure 97. JTAG Configuration of a Single Device Using a Microprocessor



Note: Connect the nCONFIG and MSEL pins to support a non-JTAG configuration scheme. If you only use a JTAG configuration, connect the nCONFIG pin to logic-high and the MSEL pins to GND. Pull DCLK and DATA[0] to either high or low, whichever is convenient on your board.

6.1.4.4. JTAG Multi-Device Configuration

You can configure multiple devices in a JTAG chain.

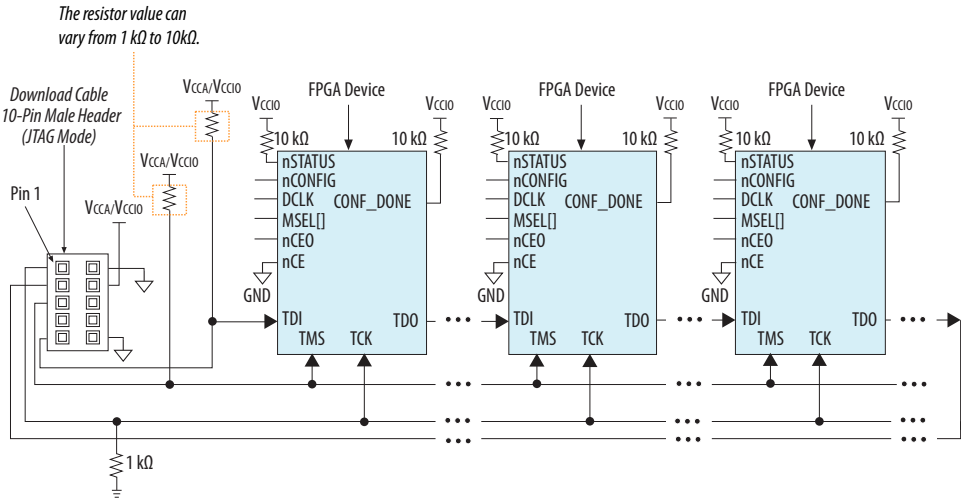
The CONF_DONE and nSTATUS signals are shared in multi-device AS, PS, and FPP configuration chains to ensure that the devices enter user mode at the same time after configuration is complete. When the CONF_DONE and nSTATUS signals are shared among all the devices, you must configure every device when JTAG configuration is performed.

1. After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device prompting it to begin configuration.
2. The second device in the chain begins configuration. If both devices are also in a JTAG chain, ensure that the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG-configured in the same order as the configuration chain.
3. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the nCEO pin of the previous device drives the nCE pin of the next device low when it has successfully been JTAG-configured.
4. Place other Intel devices that have JTAG support in the same JTAG chain for device programming and configuration.

To configure an Intel Cyclone 10 LP device using a download cable, connect the device as shown in the following figure.

- For device using V_{CCIO} of 2.5, 3.0, and 3.3 V:
  - All I/O inputs must maintain a maximum AC voltage of 4.1 V because JTAG pins do not have the internal PCI clamping diodes to prevent voltage overshoot.
  - You must power up the V_{CC} of the download cable with a 2.5-V supply from V_{CCA}.
- For device using V_{CCIO} of 1.2, 1.5, and 1.8 V:
  - You can power up the V_{CC} of the download cable with the supply from V_{CCIO}.
  - In the Intel FPGA download cable and Intel FPGA parallel port cable, this pin is connected to nCE when it is used for AS programming, otherwise it is not connected.

Figure 98. JTAG Configuration of Multiple Devices Using a Download Cable



Note: Connect the nCONFIG and MSEL pins to support a non-JTAG configuration scheme. If you only use a JTAG configuration, connect the nCONFIG pin to logic-high and the MSEL pins to GND. Pull DCLK and DATA[0] to either high or low, whichever is convenient on your board.

6.1.4.5. Combining JTAG and AS Configuration Schemes

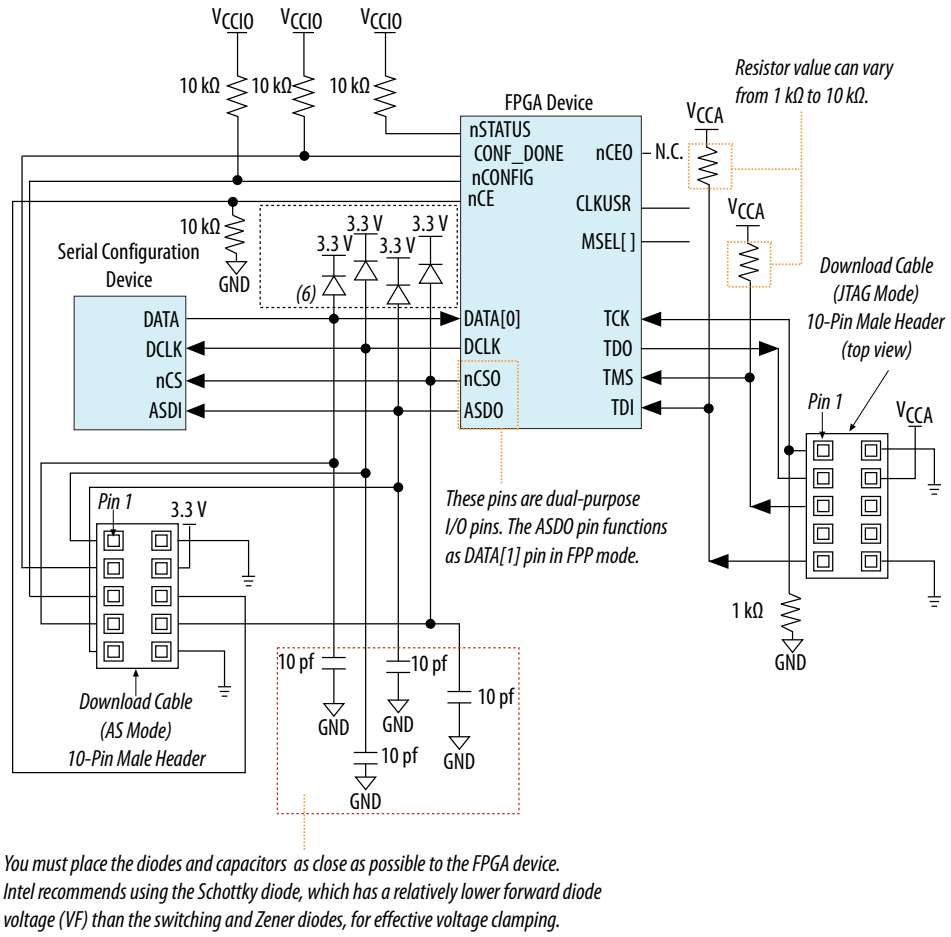
You can combine the AS configuration scheme with the JTAG-based configuration.

This setup uses two 10-pin download cable headers on the board.

- One download cable is used in JTAG mode to configure the Intel Cyclone 10 LP device directly through the JTAG interface.
- The other download cable is used in AS mode to program the serial configuration device in-system through the AS programming interface.

If you try configuring the device using both schemes simultaneously, JTAG configuration takes precedence and AS configuration terminates.

Figure 99. Combining JTAG and AS Configuration Schemes



6.1.4.6. Programming Serial Configuration Devices In-System with the JTAG Interface

Intel Cyclone 10 LP devices in a single- or multiple-device chain support in-system programming of a serial configuration device with the JTAG interface through the SFL design.

The intelligent host or download cable of the board can use the four JTAG pins on the Intel Cyclone 10 LP device to program the serial configuration device in system, even if the host or download cable cannot access the configuration pins (DCLK, DATA, ASDI, and nCS).

The SFL design is a JTAG-based in-system programming solution for Intel serial configuration devices. The SFL is a bridge design for the Intel Cyclone 10 LP device that uses its JTAG interface to access the JTAG Indirect Configuration Device Programming (`.jic`) file and then uses the AS interface to program the serial configuration device. In a multiple device chain, you must only configure the master device that controls the serial configuration device. Slave devices in the multiple

device chain that are configured by the serial configuration device do not have to be configured when using this feature. To successfully use this feature, set the MSEL pins of the master device to select the AS configuration scheme.

The serial configuration device in-system programming through the Intel Cyclone 10 LP device JTAG interface has three stages:

- Loading the SFL design
- Configuring the device
- Reconfiguring the device

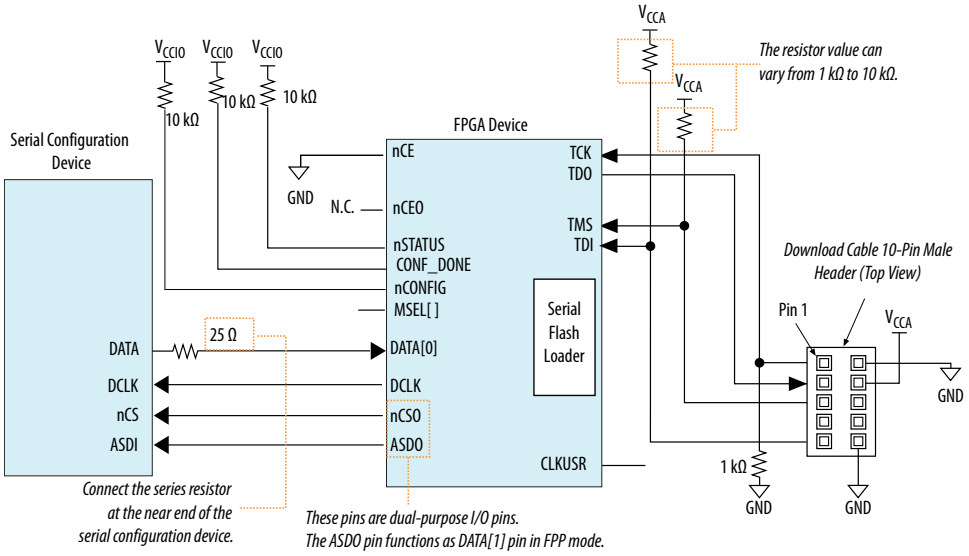
6.1.4.6.1. Loading the SFL Design

The intelligent host uses the JTAG interface to configure the master device with an SFL design.

The SFL design bridges the JTAG interface and AS interface with glue logic. The SFL design allows the master device to control the access of four serial configuration device pins or Active Serial Memory Interface (ASMI) pins, through the JTAG interface. The ASMI pins are serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and active-low chip select (nCS) pins.

If you configure a master device with an SFL design, the master device enters user mode even though the slave devices in the multiple device chain are not being configured. The master device enters user mode with a SFL design even though the CONF_DONE signal is externally held low by the other slave devices in chain.

Figure 100. Programming Serial Configuration Devices In-System Using the JTAG Interface



6.1.4.6.2. Configuring the Device

In the second stage, the SFL design in the master device enables you to write the configuration data for the device chain into the serial configuration device with the Intel Cyclone 10 LP device JTAG interface.

The JTAG interface sends the programming data for the serial configuration device to the Intel Cyclone 10 LP device first. The Intel Cyclone 10 LP device then uses the ASMI pins to send the data to the serial configuration device.

6.1.4.6.3. Reconfiguring the Device

After the configuration data is successfully written into the serial configuration device, the Intel Cyclone 10 LP device does not automatically start reconfiguration.

The intelligent host issues the PULSE_NCONFIG JTAG instruction to initialize the reconfiguration process. During reconfiguration, the master device resets and the SFL design no longer exists in the Intel Cyclone 10 LP device and the serial configuration device configures all the devices in the chain with the user design.

6.1.4.7. JTAG Instructions

You can perform JTAG testing on Intel Cyclone 10 LP devices before, during, and after configuration using the JTAG instructions.

Intel Cyclone 10 LP devices support the BYPASS, IDCODE, and SAMPLE instructions during configuration without interrupting configuration.

All other JTAG instructions can only be issued by first interrupting configuration and reprogramming I/O pins with the ACTIVE_DISENGAGE and CONFIG_IO instructions.

- The CONFIG_IO instruction configures the I/O buffers through the JTAG port and interrupts configuration when issued after the ACTIVE_DISENGAGE instruction. This instruction allows you to perform board-level testing prior to configuring the Intel Cyclone 10 LP device or waiting for a configuration device to complete configuration.
- Before issuing the CONFIG_IO instruction, you must issue the ACTIVE_DISENGAGE instruction. In Intel Cyclone 10 LP devices, the CONFIG_IO instruction does not hold nSTATUS low until reconfiguration, so you must disengage the active configuration mode controller when active configuration is interrupted. The ACTIVE_DISENGAGE instruction places the active configuration mode controllers in an idle state before JTAG programming.

The ACTIVE_ENGAGE instruction allows you to re-engage a disengaged active configuration mode controller.

6.1.4.7.1. CONFIG_IO

Use the CONFIG_IO instruction to reconfigure the I/O configuration shift register (IOCSR) chain.

This instruction allows you to perform board-level testing before configuring the Intel Cyclone 10 LP device or waiting for a configuration device to complete configuration. After the configuration is interrupted and JTAG testing is complete, you must reconfigure the part through the PULSE_NCONFIG JTAG instruction or by pulsing the nCONFIG pin low.

You can issue the CONFIG_IO instruction any time during user mode.

You must meet the following timing restrictions when using the CONFIG_IO instruction:

- The CONFIG_IO instruction cannot be issued when the nCONFIG pin is low.
- You must observe a 230µs minimum wait time after any of the following conditions:
  - nCONFIG pin goes high
  - Issue the PULSE_NCONFIG instruction
  - Issue the ACTIVE_ENGAGE instruction, before issuing the CONFIG_IO instruction
- You must wait 230 µs after power up, with the nCONFIG pin high before issuing the CONFIG_IO instruction (or wait for the nSTATUS pin to go high).

Use the ACTIVE_DISENGAGE instruction with the CONFIG_IO instruction to interrupt configuration.

Table 41. JTAG CONFIG_IO (without JTAG_PROGRAM) Instruction Flows

The table below lists the sequence of instructions to use for various CONFIG_IO usage scenarios. The list shows the configuration scheme and current state of the device.

Note: *R* indicates that the instruction must be executed before the next instruction, *O* indicates the optional instruction, *A* indicates that the instruction must be executed, and *NA* indicates that the instruction is not allowed in this mode.

JTAG Instruction	Prior to User Mode(Interrupting Configuration)			User Mode			Power Up		
	PS	FPP	AS	PS	FPP	AS	PS	FPP	AS
ACTIVE_DISENGAGE	O	O	O	O	O	O	–	–	–
CONFIG_IO	R	R	R	R	R	R	NA	NA	NA
JTAG Boundary Scan Instructions (no JTAG_PROGRAM)	O	O	O	O	O	O	–	–	–
ACTIVE_ENGAGE	A	A	R (27)	A	A	R (27)	–	–	–
PULSE_nCONFIG			A (28)			O	–	–	–
Pulse nCONFIG pin			A (28)			O	–	–	–
JTAG TAP Reset	R	R	R	R	R	R	–	–	–

The CONFIG_IO instruction does not hold nSTATUS low until reconfiguration. You must disengage the AS configuration controller by issuing the ACTIVE_DISENGAGE and ACTIVE_ENGAGE instructions when active configuration is interrupted. You must issue the ACTIVE_DISENGAGE instruction alone or prior to the CONFIG_IO instruction if the JTAG_PROGRAM instruction is to be issued later. This puts the active configuration controllers into the idle state. The active configuration controller is reengaged after user mode is reached through JTAG programming.

Note: While executing the CONFIG_IO instruction, all user I/Os are tri-stated.

(27) Required if you use ACTIVE_ENGAGE.

(28) Not required if you use ACTIVE_ENGAGE.

If reconfiguration after interruption is performed using configuration modes (rather than using JTAG_PROGRAM), it is not necessary to issue the ACTIVE_DISENGAGE instruction prior to CONFIG_IO. You can start reconfiguration by either pulling nCONFIG low for at least 500 ns or issuing the PULSE_NCONFIG instruction. If the ACTIVE_DISENGAGE instruction was issued and the JTAG_PROGRAM instruction fails to enter user mode, you must issue the ACTIVE_ENGAGE instruction to reactivate the active configuration controller. Issuing the ACTIVE_ENGAGE instruction also triggers reconfiguration in configuration modes; therefore, it is not necessary to pull nCONFIG low or issue the PULSE_NCONFIG instruction.

6.1.4.7.2. ACTIVE_DISENGAGE

The ACTIVE_DISENGAGE instruction places the active configuration controller (AS) into an idle state prior to JTAG programming.

You place the active controller in an idle state for two reasons:

- To ensure that it is not trying to configure the device during JTAG programming.
- To allow the controllers to properly recognize a successful JTAG programming that results in the device reaching user mode.

The ACTIVE_DISENGAGE instruction is required before JTAG programming regardless of the current state of the Intel Cyclone 10 LP device if the MSEL pins are set to an AS configuration scheme. If the ACTIVE_DISENGAGE instruction is issued during a passive configuration scheme (PS or FPP), it has no effect on the Intel Cyclone 10 LP device. Similarly, the CONFIG_IO instruction is issued after an ACTIVE_DISENGAGE instruction, but is no longer required to properly halt configuration.

Table 42. JTAG Programming Instruction Flows

The table below lists the required, recommended, and optional instructions for each configuration mode. The ordering of the required instructions is a hard requirement and must be met to ensure functionality. The list shows the configuration scheme and current state of the device.

Note: *R* indicates that the instruction must be executed before the next instruction, *O* indicates the optional instruction, *Rc* indicates the recommended instruction, and *NA* indicates that the instruction is not allowed in this mode.

JTAG Instruction	Prior to User Mode(Interrupting Configuration)			User Mode			Power Up		
	PS	FPP	AS	PS	FPP	AS	PS	FPP	AS
ACTIVE_DISENGAGE	O	O	R	O	O	O	O	O	R
CONFIG_IO	Rc	Rc	O	O	O	O	NA	NA	NA
Other JTAG Instructions	O	O	O	O	O	O	O	O	O
JTAG_PROGRAM	R	R	R	R	R	R	R	R	R
CHECK_STATUS	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc	Rc
JTAG_STARTUP	R	R	R	R	R	R	R	R	R
JTAG TAP Reset/ Other Instructions	R	R	R	R	R	R	R	R	R

In the AS configuration scheme, the ACTIVE_DISENGAGE instruction puts the active configuration controller into idle state.

- If a successful JTAG programming is executed, the active controller is automatically re-engaged after user mode is reached through JTAG programming. This causes the active controller to transition to their respective user mode states.
- If JTAG programming fails to get the Intel Cyclone 10 LP device to enter user mode and re-engage active programming, you can re-engage the AS controller by moving the JTAG TAP controller to the reset state or by issuing the ACTIVE_ENGAGE instruction.

6.1.4.7.3. ACTIVE_ENGAGE

The ACTIVE_ENGAGE instruction allows you to re-engage a disengaged active controller.

You can issue this instruction any time during configuration or user mode to reengage an already disengaged active controller, as well as trigger reconfiguration of the Intel Cyclone 10 LP device in the active configuration scheme. The ACTIVE_ENGAGE instruction functions as the PULSE_NCONFIG instruction when the device is in the PS or FPP configuration schemes. The nCONFIG pin is disabled when the ACTIVE_ENGAGE instruction is issued.

Note: Intel does not recommend using the ACTIVE_ENGAGE instruction, but it is provided as a fail-safe instruction for re-engaging the active configuration controller (AS).

6.1.4.7.4. Overriding the Internal Oscillator

EN_ACTIVE_CLK and DIS_ACTIVE_CLK enable you to override the internal oscillator during the active configuration scheme.

The AS configuration controllers use the internal oscillator as the clock source. You can change the clock source to CLKUSR through the JTAG instruction.

The EN_ACTIVE_CLK and DIS_ACTIVE_CLK JTAG instructions toggle on or off whether or not the active clock is sourced from the CLKUSR pin or the internal configuration oscillator. To source the active clock from the CLKUSR pin, issue the EN_ACTIVE_CLK instruction. The CLKUSR pin becomes the active clock source.

When using the EN_ACTIVE_CLK instruction, you must enable the internal oscillator for the clock change to occur. By default, the configuration oscillator is disabled after configuration and initialization is complete as well as the device has entered user mode. However, the internal oscillator is enabled in user mode by any of the following conditions:

- A reconfiguration event (e.g. driving the nCONFIG pin to go low)
- Remote update is enabled
- Error detection is enabled

Note: When using the EN_ACTIVE_CLK and DIS_ACTIVE_CLK JTAG instructions to override the internal oscillator, you must clock the CLKUSR pin at two times the expected DCLK frequency. The CLKUSR pin allows a maximum frequency of 40 MHz (40 MHz DCLK).

Normally, a test instrument uses the CLKUSR pin when it wants to drive its own clock to control the AS state machine. To revert the clock source back to the configuration oscillator, issue the DIS_ACTIVE_CLK instruction. After you issue the DIS_ACTIVE_CLK instruction, you must continue to clock the CLKUSR pin for 10 clock

cycles. Otherwise, even toggling the nCONFIG pin does not revert the clock source and reconfiguration does not occur. A POR reverts the clock source back to the configuration oscillator. Toggling the nCONFIG pin or driving the JTAG state machine to reset state does not revert the clock source.

EN_ACTIVE_CLK

The DIS_ACTIVE_CLK instruction breaks the CLKUSR enable latch set by the EN_ACTIVE_CLK instruction and causes the clock source to revert back to the internal oscillator.

After the DIS_ACTIVE_CLK instruction is issued, you must continue to clock the CLKUSR pin for 10 clock cycles.

DIS_ACTIVE_CLK

The DIS_ACTIVE_CLK instruction breaks the CLKUSR enable latch set by the EN_ACTIVE_CLK instruction and causes the clock source to revert back to the internal oscillator.

After the DIS_ACTIVE_CLK instruction is issued, you must continue to clock the CLKUSR pin for 10 clock cycles.

6.2. Configuration Requirement

The Intel Cyclone 10 LP device have specific configuration pin requirements.

6.2.1. Power-On Reset (POR) Circuit

The POR circuit keeps the device in reset state until the power supply voltage levels have stabilized during device power up.

After device power up, the device does not release nSTATUS until V_{CCINT}, V_{CCA}, and V_{CCIO} (for I/O banks in which the configuration and JTAG pins reside) are above the POR trip point of the device.

- V_{CCINT} and V_{CCA} are monitored for brown-out conditions after device power up.
- V_{CCA} is the analog power to the phase-locked loop (PLL).

In some applications, it is necessary for a device to wake up very quickly to begin operation. Intel Cyclone 10 LP devices offer the fast POR time option to support fast wake-up time applications. The fast POR time option has stricter power-up requirements compared to the standard POR time option. You can select either the fast or standard POR option with the MSEL pin settings.

Note: If your system exceeds the fast or standard POR time, you must hold nCONFIG low until all the power supplies are stable.

Related Information

Intel Cyclone 10 LP Device Datasheet
Provides more information about the POR specifications.

6.2.2. Configuration File Size

To calculate the amount of storage space required for multiple device configurations, add the file size of each device together.

Use the following data to estimate the file size before design compilation.

Note: Different configuration file formats, such as Hexadecimal (`.hex`) or Tabular Text File (`.ttf`) formats, have different file sizes. However, for any specific version of the Intel Quartus Prime software, any design targeted for the same device has the same uncompressed configuration file size.

If you use compression, the file size varies after each compilation, because the compression ratio depends on the design.

Table 43. Approximate Uncompressed Configuration Raw Binary File (.rbf) Sizes for Intel Cyclone 10 LP Devices

Device	Data Size (bits)
10CL006	2,944,088
10CL010	2,944,088
10CL016	4,086,848
10CL025	5,748,552
10CL040	9,534,304
10CL055	14,889,560
10CL080	19,965,752
10CL120	28,571,696

6.2.3. Configuration and JTAG Pin I/O Requirements

Intel Cyclone 10 LP devices are manufactured using the TSMC 60-nm low-k dielectric process. Although the Intel Cyclone 10 LP devices use TSMC 2.5 V transistor technology in the I/O buffers, the devices are compatible and able to interface with 2.5 V, 3.0 V, and 3.3 V configuration voltage standards by following specific requirements.

All I/O inputs must maintain a maximum AC voltage of 4.1 V.

- When using a serial configuration device in an AS configuration scheme, you must connect a 25 Ω series resistor for the `DATA[0]` pin.
- When cascading the Intel Cyclone 10 LP device family in a multi-device configuration for AS, FPP, and PS configuration schemes, you must connect the repeater buffers between the master and slave devices for the DATA and DCLK pins.
- When using the JTAG configuration scheme in a multi-device configuration, connect 25 Ω resistors on both ends of the `TDO-TDI` path if the `TDO` output driver is not an Intel Cyclone 10 LP device.

The output resistance of the repeater buffers and the TDO path for all cases must fit the maximum overshoot equation below:

$$0.8Z_O \leq R_E \leq 1.8Z_O$$

Note: Z_O is the transmission line impedance and R_E is the equivalent resistance of the output buffer.

6.3. Configuration Details

This section describes the MSEL pin settings, configuration sequence, device configuration pins, configuration pin options, and configuration data compression.

6.3.1. MSEL Pin Settings

To select a configuration scheme, hardwire the MSEL pins to V_{CCA} or GND without pull-up or pull-down resistors.

Table 44. MSEL Pin Settings for Each Configuration Scheme of Intel Cyclone 10 LP Devices with MSEL[3:0] Pins

Configuration Scheme	Valid MSEL[3..0]	POR Delay	Configuration Voltage Standard (V)
AS	1101	Fast	3.3
	0100	Fast	3.0
	0010	Standard	3.3
	0011	Standard	3.0
PS	1100	Fast	3.3/3.0/2.5
	0000	Standard	3.3/3.0/2.5
FPP	1110	Fast	3.3/3.0/2.5
	1111	Fast	1.8/1.5

Smaller Intel Cyclone 10 LP devices or package options (E144, M164, and U256) do not have the MSEL[3] pin. To configure these devices, select the MSEL[2:0] pins according to the table below.

Table 45. MSEL Pin Settings for AS Configuration Scheme with MSEL[2:0] Pins

Configuration Scheme	Valid MSEL[2..0]	POR Delay	Configuration Voltage Standard (V)
AS	101	Fast	3.3
	010	Standard	3.3
	011	Standard	3.0
PS	100	Fast	3.3/3.0/2.5
	000	Standard	3.3/3.0/2.5
FPP	110	Fast	3.3/3.0/2.5
	111	Fast	1.8/1.5

Note: The configuration voltage standard applies to the V_{CCIO} supply of the bank in which the configuration pins reside.

For JTAG-based configuration schemes, the MSEL pin settings are ignored. Do not leave the MSEL pins floating. Connect them to V_{cca} or GND. These pins support the non-JTAG configuration scheme used in production. Intel recommends you connect the MSEL pins to GND if your device is only using JTAG configuration.

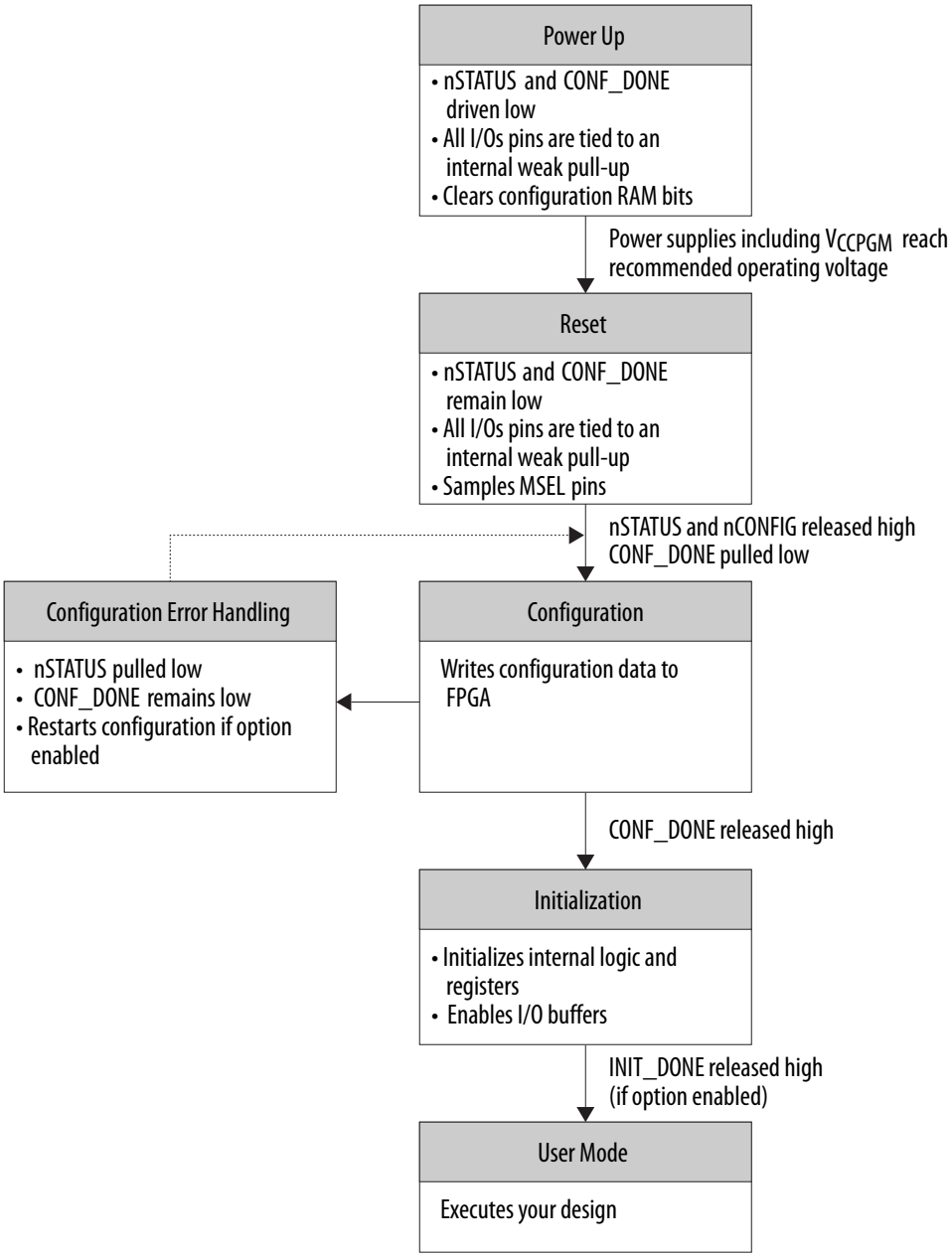
Note: You must also select the configuration scheme in the **Configuration** page of the **Device and Pin Options** dialog box in the Intel Quartus Prime software. Based on your selection, the option bit in the programming file is set accordingly.

6.3.2. Configuration Sequence

The configuration process flow includes power up, reset, configuration, and initialization.

Figure 101. Configuration Sequence for Intel Cyclone 10 LP Devices

The figure describes the configuration sequence and each configuration stage.



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.

6.3.2.1. Power Up

Power up all the power supplies that are monitored by the POR circuitry.

If the device is powered up from the power-down state, VCCINT, VCCA, and VCCIO (for the I/O banks in which the configuration and JTAG pins reside) must be powered up to the appropriate level for the device to exit from POR.

6.3.2.2. Reset

After power up, Intel Cyclone 10 LP devices go through POR.

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 Cyclone 10 LP device is ready to begin configuration.

Set the POR delay using the MSEL pins. During POR, the device resets, holds nSTATUS and CONF_DONE low, and tri-states all user I/O pins (for PS and FPP configuration schemes only). To tri-state the configuration bus for AS configuration schemes, you must tie nCE high and nCONFIG low.

The user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are always enabled (after POR) before and during configuration.

- When the device exits POR, all user I/O pins continue to tri-state. While nCONFIG is low, the device is in reset.
- When nCONFIG goes high, the device exits reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10 kΩ pull-up resistor.
- After nSTATUS is released, the device is ready to receive configuration data and the configuration stage starts.

6.3.2.3. Configuration

Configuration data latches into the Intel Cyclone 10 LP at each DCLK cycle.

However, the width of the data bus and the configuration time taken differ for each scheme. After the device receives all the configuration data, the device releases the open-drain CONF_DONE pin, which is pulled high by an external 10 kΩ pull-up resistor.

A low-to-high transition on the CONF_DONE pin indicates that the configuration is complete and initialization of the device can begin.

Begin reconfiguration by pulling the nCONFIG pin low. The nCONFIG pin must be low for at least 500 ns. When nCONFIG is pulled low, the Intel Cyclone 10 LP device is reset.

The Intel Cyclone 10 LP device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. Reconfiguration begins when nCONFIG returns to a logic-high level and the Intel Cyclone 10 LP device releases nSTATUS.

6.3.2.4. Configuration Error Handling

If an error occurs during configuration, Intel Cyclone 10 LP devices assert the nSTATUS signal low to indicate a data frame error and the CONF_DONE signal stays low.

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. The device releases nSTATUS after a reset time-out period (a maximum of 230 µs), and retries configuration.

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 500 ns.

6.3.2.5. 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 Cyclone 10 LP device will be provided with enough clock cycles for proper initialization.

Note: If you use the optional CLKUSR pin as the initialization clock source and the nCONFIG pin is pulled low to restart configuration during device initialization, ensure that the CLKUSR or DCLK pin continues toggling until the nSTATUS pin goes low and then goes high again.

The CLKUSR pin provides you with the flexibility to synchronize initialization of multiple devices or to delay initialization. Supplying a clock on the CLKUSR pin during initialization does not affect configuration.

The CLKUSR pin allows you to control when your device enters user mode for an indefinite amount of time. You can turn on the **Enable user-supplied start-up clock (CLKUSR**) option in the **General** page of the **Device and Pin Options** dialog box in the Intel Quartus Prime software. When you turn on this option, the CLKUSR pin is the

initialization clock source.

After the configuration data is accepted and CONF_DONE goes high, Intel Cyclone 10 LP devices require 3,192 clock cycles to initialize properly and enter user mode.

Note: If you use the optional CLKUSR pin and the nCONFIG pin is pulled low to restart configuration during device initialization, ensure that the CLKUSR pin continues to toggle when nSTATUS is low (a maximum of 230 µs).

6.3.2.6. User Mode

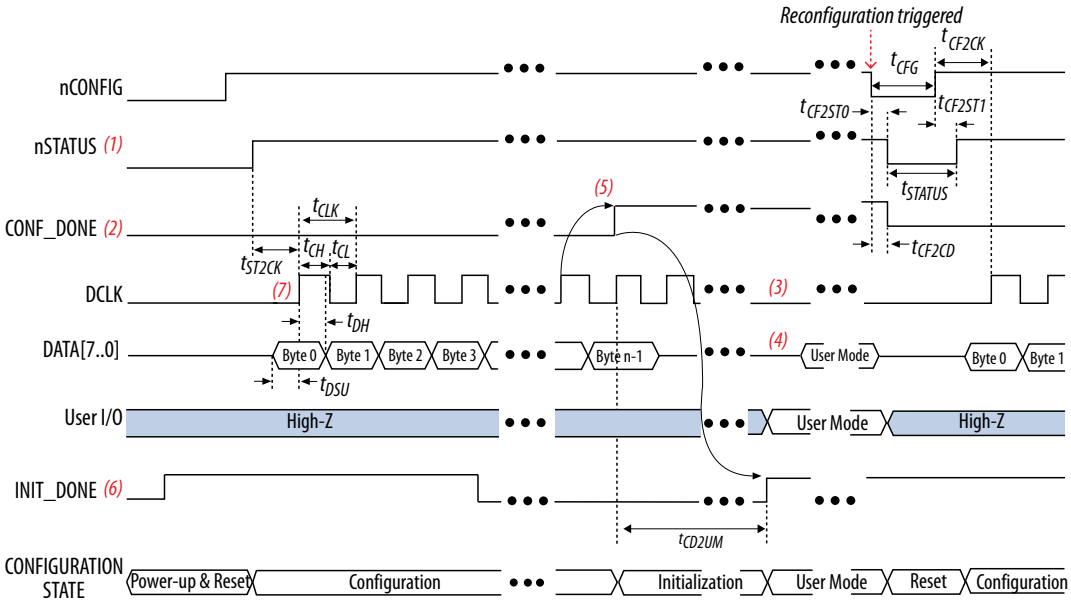
You can enable the optional INIT_DONE pin to monitor the end of initialization and the start of user mode with a low-to-high transition.

After the INIT_DONE pin is pulled high, initialization completes and your design starts executing. The user I/O pins will then function as specified by your design.

6.3.3. Configuration Timing Waveforms

6.3.3.1. FPP Configuration Timing

Figure 102. FPP Configuration Timing Waveform



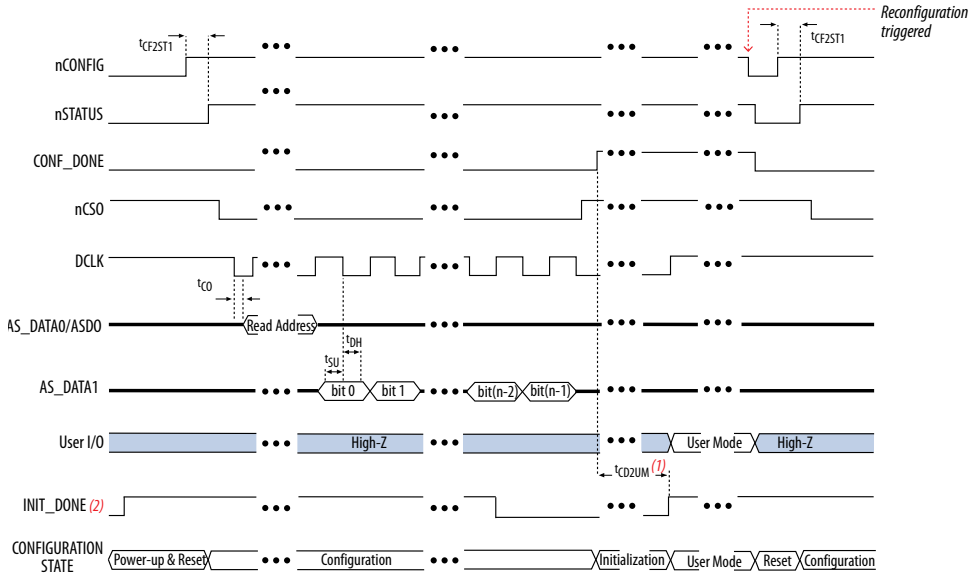
(1) After power-up, the device holds nSTATUS low for the time of the POR delay.
(2) After power-up, before and during configuration, CONF_DONE is low.
(3) Do not leave DCLK floating after configuration. DCLK is ignored after configuration is complete. It can toggle high or low if required.
(4) DATA[7..0] 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.

Related Information

FPP Configuration Timing
Provides the FPP configuration timing parameters.

6.3.3.2. AS Configuration Timing

Figure 103. AS Configuration Timing Waveform



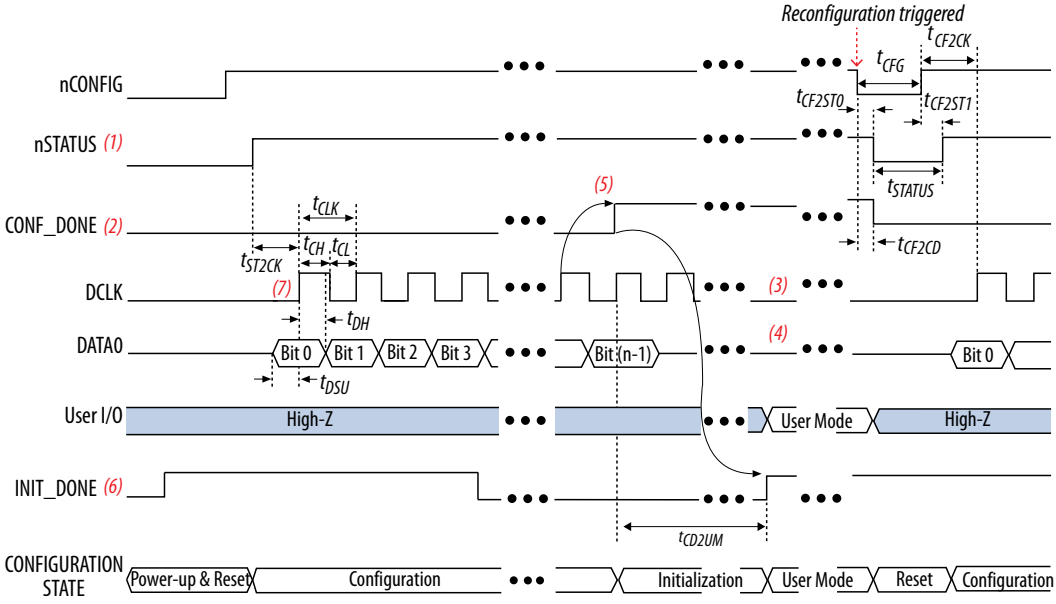
(1) The initialization clock can be from internal oscillator or CLKUSR pin.
(2) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Related Information

AS Configuration Timing
Provides the AS configuration timing parameters.

6.3.3.3. PS Configuration Timing

Figure 104. PS Configuration Timing Waveform



(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 the DCLK pin floating after configuration. Drive the DCLK pin high or low, whichever is more convenient.
(4) Do not leave the DATA[0] pin floating after configuration. Drive the DATA[0] pin high or low, whichever is more convenient.
(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.

Related Information

PS Configuration Timing
Provides the PS configuration timing parameters.

6.3.4. Device Configuration Pins

Table 46. Configuration Pins Summary for Intel Cyclone 10 LP Devices

Bank	Configuration Pin	Dedicated	Input/Output	Powered By	Configuration Mode
1	nCSO	–	Output	V_{CCIO}	AS
6	CRC_ERR0R	–	Output	V_{CCIO}/Pull-up	Optional, all modes
1	DATA[0]	–	Input	V_{CCIO}	PS, FPP, AS
1	DATA[1]/ASDO	–	Input	V_{CCIO}	FPP
			Output	V_{CCIO}	AS
8	DATA[7..2]	–	Input	V_{CCIO}	FPP

continued...

Bank	Configuration Pin	Dedicated	Input/Output	Powered By	Configuration Mode
6	INIT_DONE	–	Output	Pull-up	Optional, all modes
1	nSTATUS	Yes	Bidirectional	Pull-up	All modes
1	nCE	Yes	Input	V_{CCIO}	All modes
1	DCLK	Yes	Input	V_{CCIO}	PS, FPP
		–	Output	V_{CCIO}	AS
6	CONF_DONE	Yes	Bidirectional	—	All modes
1	TDI	Yes	Input	V_{CCIO}	JTAG
1	TMS	Yes	Input	V_{CCIO}	JTAG
1	TCK	Yes	Input	V_{CCIO}	JTAG
1	nCONFIG	Yes	Input	V_{CCIO}	All modes
6	CLKUSR	–	Input	V_{CCIO}	Optional
6	nCEO	–	Output	V_{CCIO}	Optional, all modes
6	MSEL[]	Yes	Input	V_{CCINT}	All modes
1	TDO	Yes	Output	V_{CCIO}	JTAG
5	DEV_CLRn	–	Input	V_{CCIO}	Optional
5	DEV_OE	—	Input	V_{CCIO}	Optional

To tri-state AS configuration pins in the AS configuration scheme, turn on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, nCSO, Data[0], and Data[1]/ASDO pins.

Dual-purpose pins settings for these pins are ignored. To set these pins to different settings, turn off the **Enable input tri-state on active configuration pins in user mode** option and set the desired setting from the **Dual-purpose Pins Setting** menu.

Related Information

Intel Cyclone 10 LP Device Family Pin Connection Guidelines
Provides detailed information about the Intel Cyclone 10 LP device configurations pins.

6.4. Configuration Data Compression

Intel Cyclone 10 LP devices receive compressed configuration bitstream and decompress the data in real-time during configuration.

Compression typically reduces the configuration file size by 35% to 55% depending on the design. You can enable compression before or after design compilation.

6.4.1. Enabling Compression Before Design Compilation

You can enable compression before design compilation through **Compiler Settings** in the Intel Quartus Prime.

To enable compression before design compilation:

1. Click **Assignment Menu ➤ Device**.
2. Select the appropriate Intel Cyclone 10 LP 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**.
4. Click **OK**.

6.4.2. Enabling Compression After Design Compilation

You can enable compression after design compilation when you create programming files.

To enable compression after design compilation:

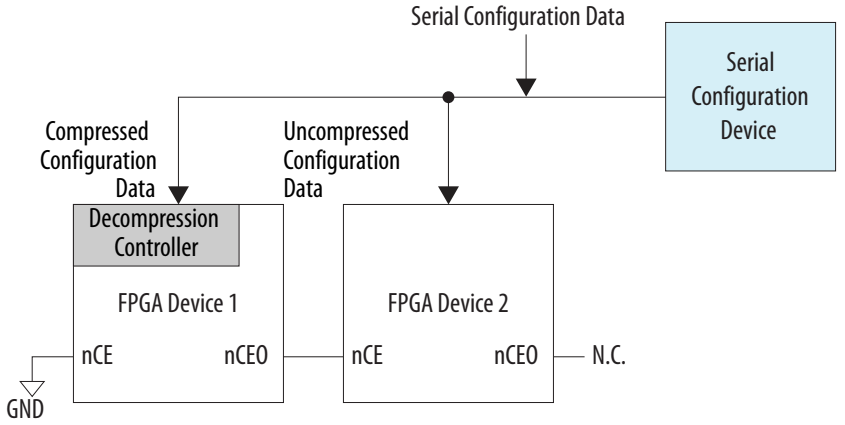
1. Click **File ➤ Convert Programming Files**.
2. Select the programming file type. If you select **Programmer Object File (.pof)**, select the appropriate configuration device.
3. Under the **Input files to convert**, select **SOF Data**.
4. Click **Add File** and navigate to select Intel Cyclone 10 LP device SRAM object files (`.sof`).
5. Select the `.sof` file you added to the **SOF Data** area and click **Properties**.
6. Turn on **Compression**.
7. Click **OK**.

6.4.3. Using Compression in Multi-Device Configuration

When multiple Intel Cyclone 10 LP devices are cascaded, you can selectively enable the compression feature for each device in the chain.

Figure 105. Compressed and Uncompressed Configuration Data in the Same Configuration File

The following figure shows a chain of two Intel Cyclone 10 LP devices. Compression is only enabled for the first device.

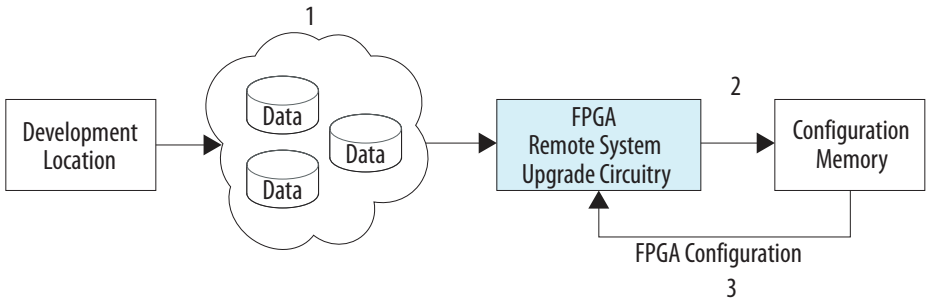


6.5. Remote System Upgrades

Intel Cyclone 10 LP devices contain dedicated remote system upgrade circuitry.

You can use this feature to upgrade your system from a remote location in AS configuration schemes.

Figure 106. Intel Cyclone 10 LP 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 Cyclone 10 LP 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 or updated configuration image.

When an error occurs, the circuitry detects the error, reverts to a safe configuration image, and provides error status to your design.

6.5.1. Enabling Remote Update

You can enable or disable remote update for Intel Cyclone 10 LP devices in the Intel Quartus Prime software before design compilation (in the Compiler Settings menu).

To enable remote update in the compiler settings of the project, perform the following steps:

1. On the **Assignments** menu, click **Device**. The **Settings** dialog box appears.
2. Click **Device and Pin Options**.
3. Click the **Configuration** tab.
4. From the **Configuration Mode** list, select **Remote**.
5. Click **OK**.
6. In the **Settings** dialog box, click **OK**.

6.5.1.1. Configuration Images

Each Intel Cyclone 10 LP device in your system requires one factory image.

The factory image is a user-defined configuration image that contains logic to perform the following:

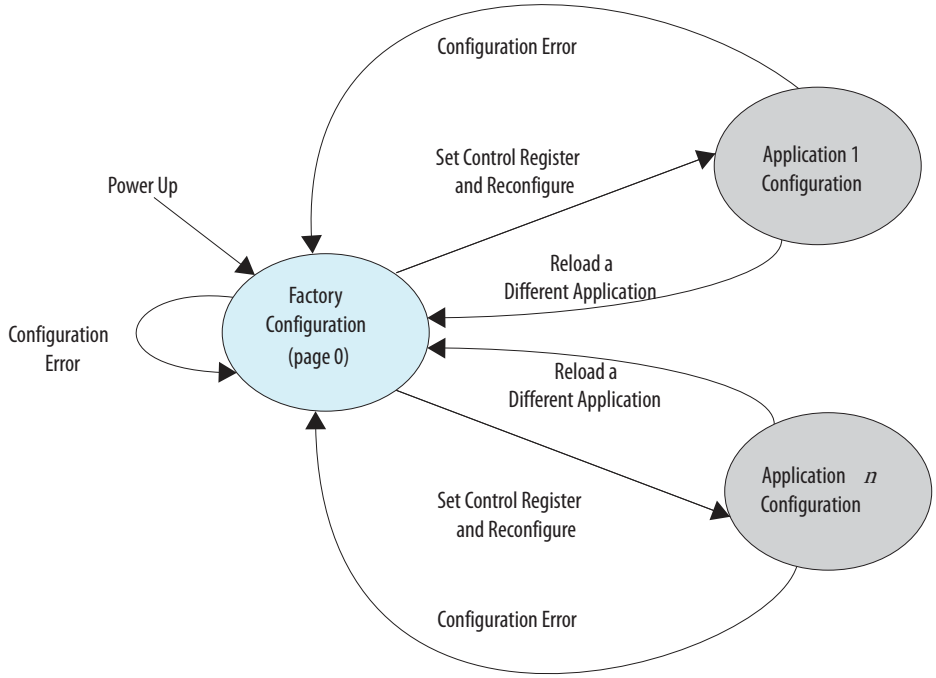
- Process errors based on the status provided by the dedicated remote system upgrade circuitry.
- Communicate with the remote host, receives new application images, and stores the images in the local non-volatile memory device.
- Determine the application image to load into the Intel Cyclone 10 LP device.
- Enable or disable the user watchdog timer and loads its time-out value.
- Instruct the dedicated remote system upgrade circuitry to start a reconfiguration cycle.

In AS configuration scheme, when an Intel Cyclone 10 LP is first powered, it loads the factory configuration located at address `boot_address[23:0] = 24b'0`. Intel recommends that you store the factory configuration image for your system at boot address `24b'0`, which corresponds to the start address location `0×000000` in the serial configuration device. A factory configuration image is a bitstream for the Intel

Cyclone 10 LP device in your system that is programmed during production and is the fall-back image when an error occurs. This image is stored in non-volatile memory and is never updated or modified using remote access.

6.5.2. Configuration Sequence in the Remote Update Mode

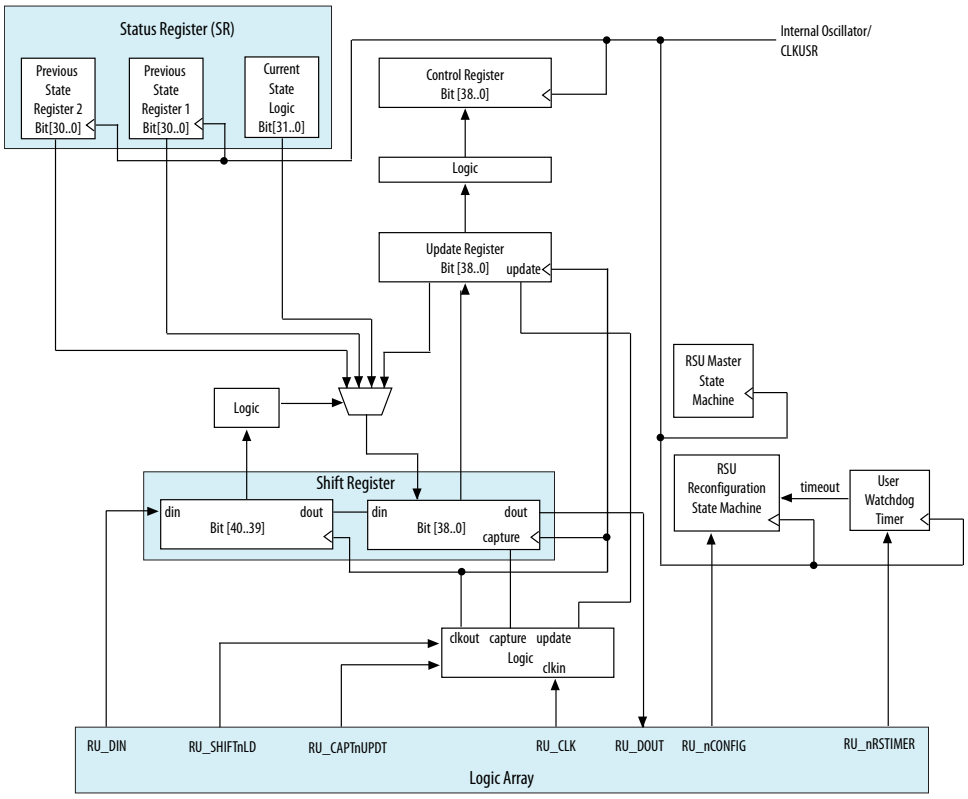
Figure 107. Transitions Between Factory and Application Configurations in Remote Update Mode



6.5.3. 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.

Figure 108. Remote System Upgrade Circuitry



Note: If you are using the Altera Remote Update IP core, the IP core controls the RU_DOUT, RU_SHIFTnLD, RU_CAPTnUPDT, RU_CLK, RU_DIN, RU_nCONFIG, and RU_nRSTIMER signals internally to perform all the related remote system upgrade operations.

6.5.4. Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that store the configuration addresses, watchdog timer settings, and status information.

Table 47. Remote System Upgrade Registers

Register	Description
Shift	Accessible by the logic array and clocked by RU_CLK. Allows the update, status, and control registers to be written and sampled by user logic.

continued...

Register	Description
	• Write access is enabled in remote update mode for factory configurations to allow writing to the update register. • Write access is disabled for all application configurations in remote update mode.
Control	This register is clocked by the 10-MHz internal oscillator. It contains the current configuration address, the user watchdog timer settings, one option bit for checking early CONF_DONE, and one option bit for selecting the internal oscillator as the startup state machine clock. The contents of this register are shifted to the shift register for the user logic in the application configuration to read. When reconfiguration is triggered, this register is updated with the contents of the update register.
Update	This register is clocked by RU_CLK. It contains data similar to that in the control register. The factory configuration updates this register by shifting data into the shift register and issuing an update. When reconfiguration is triggered, the contents of the update register are written to the control register.
Status	After each reconfiguration, the remote system upgrade circuitry updates this register to indicate the event that triggered the reconfiguration. This register is clocked by the 10-MHz internal oscillator.

6.5.4.1. Control Register

The remote system upgrade control register stores the application configuration address, the user watchdog timer settings, and option bits for an application configuration.

Table 48. Control Register Bits

Bit	Name	Value	Description
11–0	Wd_timer[11..0]	12'b000000000000	User watchdog time-out value; most significant 12 bits of 29-bit count value: {Wd_timer[11..0],17'b1000}.
33–12	Ru_address[21..0]	22'b0000000000000000000000	Configuration address time-out value; most significant 22 bits of 24-bit boot address value: boot_address[23:0] = {Ru_address[21..0],2'b0}.
34	Rsv1	1'b0	Reserved bit
35	Wd_en	1'b1	User watchdog timer enable bit. Set this bit to **1** to enable the watchdog timer.
36	Osc_int	1'b1	Internal oscillator as startup state machine clock enable bit. Option bit for the application configuration. This bit ensures a functional startup clock to eliminate the hanging of start up. *Note:* When all option bits are turned on, they provide complete coverage for the programming and startup portions of the application configuration. Intel recommends turning on both the Osc_int and Cd_early option bits.
37	Cd_early	1'b1	Early CONF_DONE check. Option bit for the application configuration. When enabled, this option bit ensures that there is a valid configuration at the boot address specified by the factory configuration and that it is of the proper size. If an invalid configuration is

continued...

Bit	Name	Value	Description
			detected or the CONF_DONE pin asserts too early, the device resets and then reconfigures the factory configuration image. *Note:* When all option bits are turned on, they provide complete coverage for the programming and startup portions of the application configuration. Intel recommends turning on both the Osc_int and Cd_early option bits.
38	Rsv2	1'b1	Reserved bit

6.5.4.2. Status Register

The remote system upgrade status register specifies the reconfiguration trigger condition.

The various trigger and error conditions include:

- Cyclical redundancy check (CRC) error during application configuration
- nSTATUS assertion by an external device due to an error
- Intel Cyclone 10 LP device logic array triggers a reconfiguration cycle, possibly after downloading a new application configuration image
- External configuration reset (nCONFIG) assertion
- User watchdog timer time out

Table 49. Remote System Upgrade Current State Logic Contents In Status Register

The table lists the contents of the current state logic in the status register, when the remote system upgrade master state machine is in factory configuration or application configuration accessing the factory information or application information, respectively.

Remote System Upgrade Master State Machine	Status Register Bit	Name	Description
Factory Information *Note:* The remote system upgrade master state machine is in factory configuration.	31:30	Master state machine current state	The current state of the remote system upgrade master state machine.
	29:24	Reserved bits	Padding bits that are set to all 0's.
	23:0	Boot address	The current 24-bit boot address that was used by the configuration scheme as the start address to load the current configuration.
Application Information 1 *Note:* The remote system upgrade master state machine is in application configuration.	31:30	Master state machine current state	The current state of the remote system upgrade master state machine.
	29	User watchdog timer enable bit	The current state of the user watchdog enable, which is active high.
	28:0	User watchdog timer time-out value	The current entire 29-bit watchdog time-out value.
Application Information 2 *Note:* The remote system upgrade master state machine is in application configuration.	31:30	Master state machine current state	The current state of the remote system upgrade master state machine.
	29:24	Reserved bits	Padding bits that are set to all 0's.
	23:0	Boot address	The current 24-bit boot address used as the start address to load the current configuration.

The previous two application configurations are available in the previous state registers (previous state register 1 and previous state register 2), but only for debugging purposes.

Table 50. Remote System Upgrade Previous State Register 1 and Previous State Register 2 Contents in Status Register

The previous state register 1 and previous state register 2 have the same bit definitions. The previous state register 1 reflects the current application configuration and the previous state register 2 reflects the previous application configuration.

Bit	Name	Description
30	nCONFIG source	One-hot, active-high field that describes the reconfiguration source that caused the Intel Cyclone 10 LP device to leave the previous application configuration. If there is a tie, the higher bit order indicates precedence. For example, if nCONFIG and remote system upgrade nCONFIG reach the reconfiguration state machine at the same time, nCONFIG precedes remote system upgrade nCONFIG.
29	CRC error source	
28	nSTATUS source	
27	User watchdog timer source	
26	Remote system upgrade nCONFIG source	
25:24	Master state machine current state	The state of the master state machine during reconfiguration causes the Intel Cyclone 10 LP device to leave the previous application configuration.
23:0	Boot address	The address used by the configuration scheme to load the previous application configuration.

If a capture is inappropriately done while capturing a previous state before the system has entered remote update application configuration for the first time, a value outputs from the shift register to indicate that the capture is incorrectly called.

6.5.5. Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions but serve different functions.

Both registers can only be updated when the device is loaded with a factory configuration image. However, the user logic controls the update register writes and the remote system upgrade state machine controls the control register writes.

In factory configurations, the user logic should send the option bits (`Cd_early` and `Osc_int`), the configuration address, and watchdog timer settings for the next application configuration bit to the update register. When the logic array configuration reset (`RU_nCONFIG`) goes high, the remote system upgrade state machine updates the control register with the contents of the update register and starts system reconfiguration from the new application page.

Note: To ensure the successful reconfiguration between the pages, assert the `RU_nCONFIG` signal for a minimum of 250 ns. This is equivalent to strobing the `reconfig` input of the Altera Remote Update IP core high for a minimum of 250 ns.

If there is an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (based on mode and error condition) by setting the control register accordingly.

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition, but before the factory configuration is loaded.

Table 51. Control Register Contents After an Error or Reconfiguration Trigger Condition

The table below lists the contents of the control register after such an event occurs for all possible error or trigger conditions.

Reconfiguration Error/Trigger	Control Register Setting In Remote Update
nCONFIG reset	All bits are 0
nSTATUS error	All bits are 0
CORE triggered reconfiguration	Update register
CRC error	All bits are 0
Wd time out	All bits are 0

6.5.6. 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, and enabled in the application configuration.

The counter is 29 bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The granularity of the timer setting is 2^{17} cycles. The cycle time is based on the frequency of 10 MHz internal oscillator or CLKUSR (maximum frequency of 40 MHz).

The timer begins counting as soon as the application configuration enters user mode. The application configuration periodically reload or reset this timer before the count expires by asserting RU_nRSTIMER. If the application configuration does not reload the user watchdog timer before the count expires, the remote system upgrade circuitry generates a time-out signal. The time-out signal triggers the circuitry to set the user watchdog timer status bit (Wd) in the remote system upgrade status register and reconfigures the device by loading the factory configuration image. To reset the time, assert RU_nRSTIMERactive for a minimum of 250 ns. This is equivalent to strobing the `reset_timer` input of the Altera Remote Update IP core high for a minimum of 250 ns.

Errors during configuration are detected by the CRC engine. Functional errors must not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

6.6. Configuration and Remote System Upgrades in Intel Cyclone 10 LP Devices Revision History

Document Version	Changes
2020.12.03	Updated the following figures:

continued...

Document Version	Changes
	• Figure: *Single Device PS Configuration Using an External Host* • Figure: *Multi-Device PS Configuration Using an External Host* • Figure: *Multiple Device PS Configuration When Both Devices Receive the Same Configuration Data*
2020.04.29	Corrected TMS pull up connection in the *JTAG Configuration of Multiple Devices Using a Download Cable* diagram.
2019.12.23	Updated Figure: *AS Configuration Timing Waveform*.
2019.01.24	Added a link to the Intel Supported Configuration Devices section of the *Device Configuration - Support Center* page on the Intel website.
2018.10.22	• Updated the notes in the *Fast Passive Parallel Configuration* topic. • Updated the diagram that shows the combination of JTAG and AS configuration schemes to correct the resistor value connected to the TCK pin from 10 kΩ to 1 kΩ.
2018.05.07	• Added the *FPP Configuration Timing Waveform*, *AS Configuration Timing Waveform*, and *PS Configuration Timing Waveform*. • Added the DEV_OE pin to the *Configuration Pins Summary for Intel Cyclone 10 LP Devices* table. • Updated the CLKUSR and nCEO pin information in the *Configuration Pins Summary for Intel Cyclone 10 LP Devices* table. • Removed the FLASH_nCE pin from the *Configuration Pins Summary for Intel Cyclone 10 LP Devices* table. • Removed the 2.5-V configuration voltage standard support for the AS configuration scheme in the *MSEL Pin Settings for Each Configuration Scheme of Intel Cyclone 10 LP Devices with MSEL[3:0] Pins* table, *MSEL Pin Settings for AS Configuration Scheme with MSEL[2:0] Pins* table, and *AS Configuration Guidelines* section.

Date	Version	Changes
December 2017	2017.12.22	• Edited the *MSEL Pin Settings for Each Configuration Scheme of Intel Cyclone 10 LP Devices* table. The voltage listed should refer to the configuration voltage standard. • Edited the typo in the *Programming Serial Configuration Devices In-System Using the JTAG Interface* figure. The resistor connected to the DATA pin should be 25 ohm.
May 2017	2017.05.08	Initial release.

7. SEU Mitigation in Intel Cyclone 10 LP Devices

The dedicated circuitry built in Intel Cyclone 10 LP devices consists of an error detection cyclic redundancy check (CRC) feature. You can use this feature to mitigate single-event upset (SEU) or soft errors.

7.1. Configuration Error Detection

In configuration mode, a frame-based CRC is stored in the configuration data and contains the CRC value for each data frame.

During configuration, the Intel Cyclone 10 LP device calculates the CRC value based on the frame of data that is received and compares it against the frame CRC value in the data stream. Configuration continues until the device detects an error or when all the values are calculated.

For Intel Cyclone 10 LP devices, the CRC is computed by the Quartus Prime software and downloaded into the device as part of the configuration bit stream. These devices store the CRC in the 32-bit storage register at the end of the configuration mode.

7.2. User Mode Error Detection

In user mode, the contents of the configured configuration random-access memory (CRAM) bits may be affected by soft errors.

These soft errors, which are caused by an ionizing particle, are not common in Intel devices. However, high-reliability applications that require the device to operate error-free may require that your designs account for these errors.

You can enable the error detection circuitry to detect soft errors. This error detection circuitry continuously computes the CRC of the configured CRAM bits. The computed 32-bit CRC value is then compared with the 32-bit pre-calculated CRC value obtained at the end of the configuration. The process of error detection continues until the device is reset—by setting nCONFIG to low.

- If the CRC values match, the resulting signature value is zero (CRC_ERROR= low) indicating the configured CRAM bits has no error.
- Otherwise, the resulting signature value is non-zero (CRC_ERROR= high) to indicate a CRC error.

The Intel Cyclone 10 LP device error detection feature does not check memory blocks and I/O buffers. These device memory blocks support parity bits that are used to check the contents of memory blocks for any error. The I/O buffers are not verified during error detection because the configuration data uses flip-flops as storage elements that are more resistant to soft errors. Similar flip-flops are used to store the pre-calculated CRC and other error detection circuitry option bits.

Note: Intel Cyclone 10 LP devices also offer on-chip circuitry for automated SEU detection. For automated checking, you can implement the error detection CRC feature in the Intel Quartus Prime software with the existing circuitry in Intel Cyclone 10 LP devices.

7.3. Error Detection Features

The on-chip error detection CRC circuitry allows you to perform the following operations without any impact on the fitting or performance of the device:

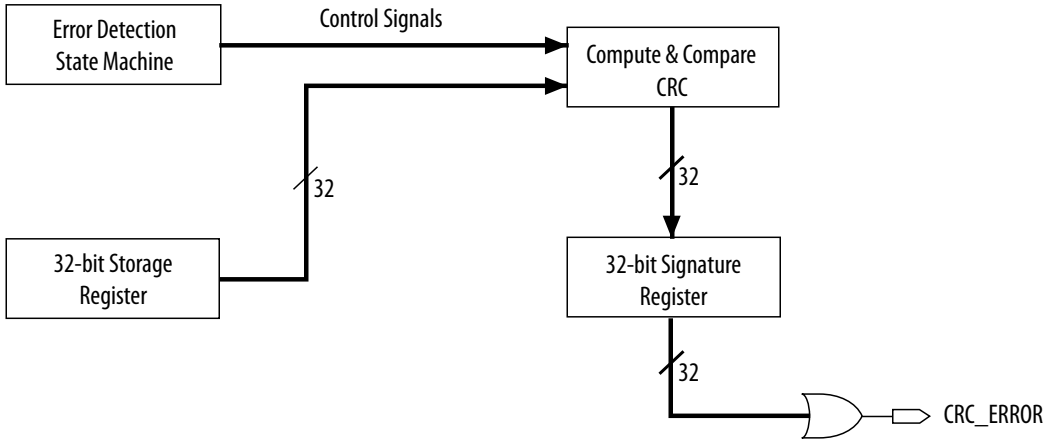
- Auto-detection of CRC errors during configuration.
- Optional CRC error detection and identification in user mode.
- Testing of error detection functions by deliberately injecting errors through the JTAG interface.

Note: Only the Intel Cyclone 10 LP devices with 1.2-V core voltage support the user mode error detection feature.

7.3.1. Error Detection Block

Figure 109. Error Detection Block Diagram

The figure shows the error detection block diagram with the two related 32-bit registers—the signature register and the storage register.



The error detection circuitry has two sets of 32-bit registers that store the computed CRC signature and the pre-calculated CRC value. A non-zero value on the signature register causes CRC_ERROR to go high.

Table 52. Error Detection Registers for Intel Cyclone 10 LP Devices

Register	Description
32-bit signature register	This register contains the CRC signature. The signature register contains the result of the user mode calculated CRC value compared against the pre-calculated CRC value. If no errors are detected, the signature register is all zeroes. A non-zero signature register indicates an error in the configuration CRAM contents. The CRC_ERROR signal is derived from the contents of this register.
32-bit storage register	This register is loaded with the 32-bit pre-computed CRC signature at the end of the configuration stage. The signature is then loaded into the 32-bit Compute and Compare CRC block during user mode to calculate the CRC error. This register forms a 32-bit scan chain during execution of the CHANGE_EDREG JTAG instruction. The CHANGE_EDREG JTAG instruction can change the content of

continued...

Register	Description
	the storage register. Therefore, the functionality of the error detection CRC circuitry is checked in-system by executing the instruction to inject an error during the operation. The operation of the device is not halted when issuing the CHANGE_EDREG JTAG instruction.

7.3.1.1. CRC_ERROR Pin

Table 53. Pin Description

Pin Name	Pin Type	Description
CRC_ERROR	I/O or output/output open-drain	An active-high signal, when driven high indicates that an error is detected in the CRAM bits. This pin is only used when you enable CRC error detection in user mode. Otherwise, the pin is used as a user I/O pin. When using this pin, connect it to an external 10-kΩ pull-up resistor to an acceptable voltage that satisfies the input voltage of the receiving device.

7.3.1.2. CHANGE_EDREG JTAG Instruction

Table 54. CHANGE_EDREG JTAG Instruction Description

JTAG Instruction	Instruction Code	Description
CHANGE_EDREG	00 0001 0101	This instruction connects the 32-bit CRC storage register between TDI and TDO. Any precomputed CRC is loaded into the CRC storage register to test the operation of the error detection CRC circuitry at the CRC_ERROR pin.

7.3.1.2.1. Example JAM File

```
NOTE MAX_FREQ "10000000";
ACTION CHANGE_EDREG = EXECUTE;
PROCEDURE EXECUTE;
BOOLEAN X = 0;
DRSTOP IDLE;
IRSTOP IDLE;
STATE IDLE;
IRSCAN 10, $015;
DRSCAN 32, $FFFFFFFF;
STATE IDLE;
EXIT 0;
ENDPROC;
```

7.3.1.2.2. Procedure

1. To configure Intel Cyclone 10 LP device, connect the device to a download cable.
2. Save a copy of the Example JAM File on page 164 as `crc.jam` with any text editor, and place the file into your working directory.
3. Access the command prompt of your operating system, and change the current directory to your working directory.
4. Type the following command at the command prompt of your operating system:

   `quartus_jli -a change_edreg -c <cable number> crc.jam`

   You can identify the cable number using the `jtagconfig` command.

After the CHANGE_EDREG is executed successfully, the 32-bit storage register of the error detection block is modified and a CRC error is detected.

7.3.1.3. Error Detection Timing

When the error detection CRC feature is enabled through the Intel Quartus Prime software, the device automatically activates the CRC process upon entering user mode, after configuration and initialization is complete.

The CRC_ERROR pin will remain low until the error detection circuitry has detected a corrupted bit in the previous CRC calculation. After the pin goes high, it remains high during the next CRC calculation. This pin does not log the previous CRC calculation. If the new CRC calculation does not contain any corrupted bits, the CRC_ERROR pin is driven low. The error detection runs until the device is reset.

The error detection circuitry is clocked by an internal configuration oscillator with a divisor that sets the maximum frequency. The CRC calculation time depends on the device and the error detection clock frequency.

7.3.1.4. Error Detection Frequency

You can control the speed of the error detection process by setting the division factor of the clock frequency in the Intel Quartus Prime software.

The divisor is 2^n, where n can be any value between 0 to 8.

The speed of the error detection process for each data frame is determined by this equation:

$$\text{Error Detection Frequency} = \frac{\text{Internal Oscillator Frequency}}{2^n}$$

Table 55. Error Detection Frequency Range for Intel Cyclone 10 LP Devices

Internal Oscillator Frequency	Error Detection Frequency		n	Divisor Range
	Maximum	Minimum		
80 MHz	80 MHz	312.5 kHz	0, 1, 2, 3, 4, 5, 6, 7, 8	1 – 256

7.3.1.5. CRC Calculation Time

CRC calculation time depends on the device and the error detection clock frequency.

Table 56. CRC Calculation Time in Intel Cyclone 10 LP Devices

The following table lists the minimum and maximum time taken to calculate the CRC value:

- The minimum time corresponds to the maximum error detection clock frequency and may vary with different processes, voltages, and temperatures (PVT).
- The maximum time corresponds to the minimum error detection clock frequency and may vary with different PVT.

Device	t_{MIN} (ms)	t_{MAX} (s)
10CL006	5	2.29
10CL010	5	2.29
10CL016	7	3.17

continued...

Device	t_{MIN} (ms)	t_{MAX} (s)
10CL025	9	4.51
10CL040	15	7.48
10CL055	23	11.77
10CL080	31	15.81
10CL120	45	22.67

7.4. Using Error Detection Features in User Mode

This section describes the pin, registers, process flow, and procedures for error detection in user mode.

7.4.1. Enabling Error Detection

To enable user mode error detection in the Intel Quartus Prime software, follow these steps:

1. On the Assignments menu, click **Device**.
2. In the Device dialog box, click **Device and Pin Options**.
3. In the **Category** list, click **Error Detection CRC**.
4. Turn on **Enable Error Detection CRC_ERROR pin**.
5. To set the CRC_ERROR pin as output open drain, turn on **Enable open drain on CRC_ERROR pin**. Turning off this option sets the CRC_ERROR pin as output.
6. In the **Divide error check frequency by** list, select a valid divisor.
7. Click **OK**.

7.4.2. Accessing Error Detection Block Through User Logic

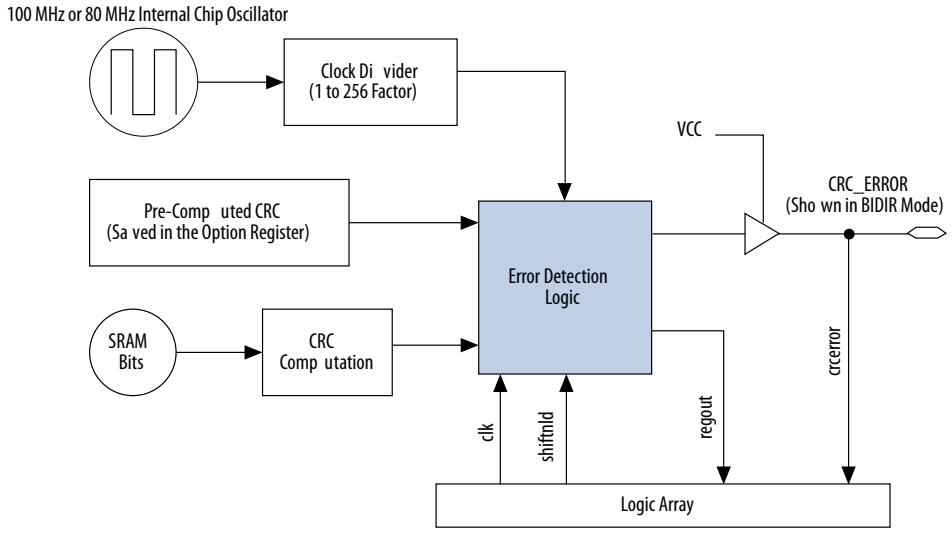
You can also enable error detection through the user logic in the Intel Cyclone 10 LP devices.

The error detection circuitry stores the computed 32-bit CRC signature in a 32-bit register, which is read out by user logic from the core.

The `cyclone10lp_crcblock` primitive is a WYSIWYG component used to establish the interface from the user logic to the error detection circuit. The `cyclone10lp_crcblock` primitive atom contains the input and output ports that must be included in the atom.

To access the logic array, insert the `cyclone10lp_crcblock` WYSIWYG atom into your design.

Figure 110. The Error Detection Block Interfacing with the WYSIWYG Atom



Note: Any soft error failure affects the user logic. Therefore, do not rely on the `regout` signal in the 32-bit CRC signature to detect a soft error. The CRC_ERROR output signal provides a more accurate reading because is it not affected by a soft error.

Table 57. CRC Block Input and Output Ports

Port	Direction	Description
<*crcblock_name*>	Input	Unique identifier for the CRC block, and represents any identifier name that is legal for the given description language (e.g. Verilog HDL, VHDL, and AHDL). This field is required.
`.clk(`<*clock source*>`)`	Input	This signal designates the clock input of this cell. All operations of this cell relate to the rising edge of the clock. Whether it is the loading of the data into the cell or data out of the cell, it always occurs on the rising edge. This port is required.
`.shiftnld` (<*shiftnld source*>)	Input	This signal is an input into the error detection block. • If `shiftnld=1`, the data shifts from the internal shift register to the `regout` at each rising edge of `clk`. • If `shiftnld=0`, the shift register parallel loads either the pre-calculated CRC value or the update register contents, depending on the `ldsrc` port input. To do this, `shiftnld` must be driven low for at least two clock cycles. This port is required.

continued...

Port	Direction	Description
.ldsrc(<*ldsrc source*>)	Input	This signal is an input into the error detection block. • While shiftnld=1, this port is ignored. • While shiftnld=0: — If ldsrc=0, the pre-computed CRC register gets selected for loading into the 32-bit shift register at the rising edge of clk when shiftnld=0. — If ldsrc=1, the signature register (result of the CRC calculation) gets selected for loading into the shift register at the rising edge of clk when shiftnld=0. This port is required.
.crcerror (<*crcerror indicator output*>)	Output	This signal is the output of the cell that synchronizes to the internal oscillator of the device (80 MHz internal oscillator) and not to the clk port. It asserts high if the error block detects that a SRAM bit has flipped and the internal CRC computation shows a difference value compared to the pre-computed value. You must connect this signal either to an output pin or a bidirectional pin. • If it is connected to an output pin, you can only monitor the CRC_ERROR pin (the core cannot access this output). • If the CRC_ERROR signal is used by the core logic to read error detection logic, you must connect this signal to a BIDIR pin. The signal is fed to the core indirectly by feeding a BIDIR pin that has its output enable port connected to V_{CC}.
.regout (<*registered output*>)	Output	This signal is the output of the error detection shift register synchronized to the clk port to be read by the core logic. It shifts one bit at each cycle, so you should clock the clk signal 31 cycles to read out the 32 bits of the shift register.

To enable the cyclone10lp_crcblock WYSIWYG atom, name the atom for each Intel Cyclone 10 LP device accordingly.

Example 1. Defining Input and Output Ports in a WYSIWYG Atom

```
cyclone10lp_crcblock<crcblock_name>
(
.clk(<clock source>),
.shiftnld(<shiftnld source>),
.ldsrc(<ldsrc source>),
.crcerror(<crcerror out destination>),
.regout(<output destination>),
);
```

7.5. SEU Mitigation for Intel Cyclone 10 LP Devices Revision History

Document Version	Changes
2020.01.06	Added new topics to the *CHANGE_EDREG JTAG Instruction* section: • *Example JAM File* • *Procedure*
2019.01.15	Updated the topic about accessing the error detection block through user logic to improve clarity of the .ldsrc description.

Date	Version	Changes
May 2017	2017.05.08	Initial release.

8. JTAG Boundary-Scan Testing for Intel Cyclone 10 LP Devices

Intel Cyclone 10 LP devices support the IEEE Std. 1149.1 (JTAG) boundary-scan testing (BST).

Related Information

AN39: IEEE 1149.1 JTAG Boundary-Scan Testing in Intel Devices
Provides more information about the IEEE Std. 1149.1 BST architecture and circuitry, TAP controller state-machine, and instruction modes.

8.1. BST Operation Control

Table 58. Boundary-Scan Register Length for Intel Cyclone 10 LP Devices

Device	Boundary-Scan Register Length
10CL010	603
10CL016	1080
10CL025	732
10CL040	1632
10CL055	1164
10CL080	1314
10CL120	1620

8.2. I/O Voltage Support in the JTAG Chain

An Intel Cyclone 10 LP device operating in BST mode uses four required pins—TDI, TDO, TMS, and TCK. The TDO output pin and all JTAG input pins are powered by the V_{CCIO} power supply of I/O Bank 1 .

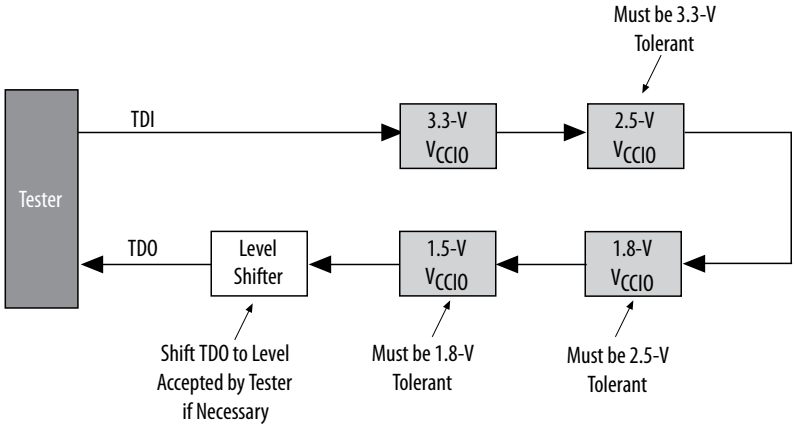
The TDO pin of a device drives out at the voltage level according to the V_{CCIO} of the device. The devices can interface with each other although the devices may have different V_{CCIO} levels.

For example, a device with 3.3-V V_{CCIO} can drive a device with 5.0-V V_{CCIO} because 3.3 V meets the minimum V_{IH} on transistor-to-transistor logic (TTL)-level input for the 5.0-V V_{CCIO} device.

For multiple devices in a JTAG chain with the 3.0-V/3.3-V I/O standard, you must connect a 25-Ω series resistor on a TDO pin driving a TDI pin.

To interface the TDI and TDO lines of the JTAG pins of devices that have different V_{CCIO} levels, insert a level shifter between the devices. If possible, construct the JTAG chain where device with a higher V_{CCIO} level drives to a device with an equal or lower V_{CCIO} level. In this setup, you only require a level shifter for shifting the TDO level to a level JTAG tester accept.

Figure 111. JTAG Chain of Mixed Voltages and Level Shifters



8.3. Boundary-Scan Description Language Support

The BSDL—a subset of VHDL—provides a syntax that allows you to describe the features of an IEEE Std. 1149.1 BST-capable device that can be tested. Test software development systems then use the BSDL files for test generation, analysis, failure diagnostics, and in-system programming.

Related Information

IEEE 1149.1 BSDL Files

8.4. JTAG Boundary-Scan Testing for Intel Cyclone 10 LP Devices Revision History

Date	Version	Changes
May 2017	2017.05.08	Initial release.

9. Power Management in Intel Cyclone 10 LP Devices

The Intel Cyclone 10 LP devices are offered with the following features:

- Available in 1.2 V and 1.0 V core voltage options
- Built-in hot-socketing compliance
- Power-on reset (POR) circuitry

9.1. External Power Supply Requirements

Table 59. Power Supply Descriptions for the Intel Cyclone 10 LP Devices

- You must power up VCCA even if the PLL is not used.
- I/O banks 1, 6, 7, and 8 contain configuration pins.

Power Supply Pin	Nominal Voltage Level (V)	Description
VCCINT	1.0, 1.2	Core voltage power supply
VCCA	2.5	PLL analog power supply
VCCD_PLL	1.0, 1.2	PLL digital power supply
VCCIO	1.2, 1.5, 1.8, 2.5, 3.0, 3.3	I/O banks power supply

Related Information

- Recommended Operating Conditions
  Provides more information about the recommended power supply's operating conditions.
- Intel Cyclone 10 LP Device Family Pin Connection Guidelines
  Provides more information about the pin connection guidelines and power regulator sharing.

9.2. Hot-Socketing Specifications

The Intel Cyclone 10 LP device is a hot-socketing compliant device that does not need any external components or special design requirements. Hot-socketing support in the Intel Cyclone 10 LP device has the following advantages:

- You can drive the devices before power up without damaging the device.
- I/O pins remain tri-stated during power up. The device does not drive out before or during power up, therefore not affecting other buses in operation.

Related Information

Hot-Socketing
Provides more information about hot-socketing specification.

9.2.1. Drive Intel Cyclone 10 LP Devices Before Power Up

Before or during power up or power down, you can drive signals into regular I/O pins without damaging the Intel Cyclone 10 LP devices.

The Intel Cyclone 10 LP device supports any power-up or power-down sequence to simplify system-level design.

9.2.2. I/O Pins Remain Tri-stated During Power Up

The output buffers of the Intel Cyclone 10 LP device are turned off during system power up or power down. The Intel Cyclone 10 LP device family does not drive out until the device is configured and working in recommended operating conditions. The I/O pins are tristated during power up or power down.

Note: The user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are always enabled (after POR) before and during configuration. The weak pull-up resistors are not enabled prior to POR.

A possible concern for semiconductor devices in general regarding hot-socketing is the potential for latch up. Latch up can occur when electrical subsystems are hot-socketed into an active system. During hot-socketing, the signal pins may be connected and driven by the active system. This occurs before the power supply can provide current to the V_{CC} of the device and ground planes. This condition can lead to latch up and cause a low-impedance path from V_{CC} to ground in the device. As a result, the device extends a large amount of current, possibly causing electrical damage.

The design of the I/O buffers and hot-socketing circuitry ensures that the Intel Cyclone 10 LP device family is immune to latch up during hot-socketing.

9.3. Hot-Socketing Feature Implementation

The hot-socketing circuit does not include the CONF_DONE, nCEO, and nSTATUS pins to ensure that these pins are able to operate during configuration. Thus, its is an expected behavior for these pins to drive out during power-up and power-down sequences.

Intel uses GND as reference for hot-socketing operation and I/O buffer designs. To ensure proper operation, Intel recommends connecting the GND between boards before connecting the power supplies. This prevents the GND on your board from being pulled up inadvertently by a path to power through other components on your board. A pulled up GND can cause an out-of-specification I/O voltage or current condition with the Intel FPGA.

9.4. Power-On Reset Circuitry

Intel Cyclone 10 LP devices contain POR circuitry to keep the device in a reset state until the power supply voltage levels have stabilized during power up. During POR, all user I/O pins are tri-stated until the power supplies reach the recommended operating levels. In addition, the POR circuitry also ensures the V_{CCIO} level of I/O banks that contain configuration pins reach an acceptable level before configuration is triggered.

The POR circuit of the Intel Cyclone 10 LP device monitors the V_{CCINT}, V_{CCA}, and V_{CCIO} (of banks 1, 5, 6, and 8) that contain configuration pins during power-on. You can power up or power down the VCCINT, VCCA, and VCCIO pins in any sequence. The V_{CCINT}, V_{CCA}, and V_{CCIO} must have a monotonic rise to their steady state levels. All VCCA pins must be powered to 2.5V (even when phase-locked loops [PLLs] are not used), and must be powered up and powered down at the same time.

After the Intel Cyclone 10 LP device enters the user mode, the POR circuit continues to monitor the VCCINT and VCCA pins so that a brown-out condition during user mode is detected. If the V_{CCINT} or V_{CCA} voltage sags below the POR trip point during user mode, the POR circuit resets the device. If the V_{CCIO} voltage sags during user mode, the POR circuit does not reset the device.

In some applications, it is necessary for a device to wake up very quickly to begin operation. Intel Cyclone 10 LP devices offer the Fast-On feature to support fast wake-up time applications. The MSEL pin settings determine the POR time (t_{POR}) of the device.

Related Information

- MSEL Pin Settings on page 143
  Provides more information about the MSEL pin settings.
- Recommended Operating Conditions
  Provides more information about the POR specifications.

9.5. Power Management in Intel Cyclone 10 LP Devices Revision History

Date	Version	Changes
May 2017	2017.05.08	Initial release.