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

SV51010-1.7

This chapter contains information about the Stratix® V supported configuration schemes, instructions about how to execute the required configuration schemes, and all the necessary option pin settings. This chapter also reviews the different ways you can configure your device and explains the design security and remote system upgrade features for the Stratix V devices.

This chapter includes the following sections:

Stratix V devices use SRAM cells to store configuration data. Because SRAM memory is volatile, you must download the configuration data to the Stratix V device each time the device powers up. You can configure Stratix V devices using one of four configuration schemes:

- Fast passive parallel (FPP) (×8, ×16, and ×32)
- Active serial (AS) (×1 and ×4)
- Passive serial (PS)
- JTAG

All configuration schemes use either an external controller (for example, a MAX® II device, MAX V device, or microprocessor), a configuration device, or a download cable. For more information about the configuration features, refer to "Configuration Features".

For more information about the Configuration via Protocol (CvP) scheme, refer to the *Configuration via Protocol (CvP) Implementation in Altera FPGAs User Guide*.

Configuration Features

Stratix V devices offer decompression, design security, and remote system upgrade features. Stratix V devices can receive a compressed configuration bitstream and decompress this data in real-time, reducing storage requirements and configuration time. Design security using configuration bitstream encryption is available in Stratix V devices, which protects your designs. You can make real-time system upgrades of your Stratix V designs from remote locations with the remote system upgrade feature.

Table 9–1 lists which configuration features you can use in each configuration scheme.

Table 9–1. Configuration Features for Stratix V Devices

Configuration Scheme	Decompression	Design Security	Remote System Upgrade
FPP (×8, ×16, ×32)	Y (1)	Y (1)	—
AS (×1, ×4)	Y	Y	Y
PS	Y	Y	—
JTAG	—	—	—

Note to Table 9–1:

(1) In these configuration schemes, the host system must accommodate a different DCLK-to-DATA[] ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

Power-On Reset Circuit and Configuration Pins Power Supply

The following sections describe the power-on reset (POR) circuit and the power supply for the configuration pins.

POR Delay Specification

POR delay is defined as the delay between the time when all the power supplies monitored by the POR circuitry reach the minimum recommended operating voltage to the time when the nSTATUS is released high and your device is ready to begin configuration.

For more information about the POR delay, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

Table 9–2 lists the fast and standard POR delay specification.

Table 9–2. Fast and Standard POR Delay Specification (1)

POR Delay	Minimum	Maximum
Fast	4 ms	12 ms
Standard	100 ms	300 ms

Note to Table 9–2:

(1) You can select the POR delay based on the MSEL settings as described in Table 9–4 on page 9–7.

Power-On Reset Circuit

The POR circuit keeps the entire system in reset mode until the power supply voltage levels have stabilized on power-up. After power-up, the device does not release nSTATUS until all the power supplies monitored by the POR circuitry are above the device's POR trip point. On power down, brown-out occurs if any of the power supplies monitored by the POR circuitry drops below the threshold level of the hot-socket circuitry.

For more information about which power supplies are monitored by the POR circuitry, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

V_{CCPGM} Pin

Stratix V devices have a power supply, V_{CCPGM}, for all dedicated configuration pins and dual-purpose pins. The supported configuration voltages are 1.8, 2.5, and 3.0 V.

Use the V_{CCPGM} pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bidirectional pins, and the dual-purpose pins that you use for configuration. The configuration input buffers do not have to share power lines with the regular I/O buffer in Stratix V devices.

The operating voltage for the configuration input pin is independent of the I/O banks power supply, V_{CCIO}, during configuration. Therefore, Stratix V devices do not require configuration voltage constraints on V_{CCIO}.

For more information about the configuration pins connections, refer to the *Stratix V Device Family Pin Connection Guidelines*.

V_{CCPD} Pin

Stratix V devices have a dedicated programming power supply, V_{CCPD}, which must be connected to 3.0 V or 2.5 V to power the I/O pre-drivers and JTAG I/O pins (TCK, TMS, TDI, TDO, and TRST).

V_{CCPD} must be greater than or equal to V_{CCIO}. If V_{CCIO} is set to 3.0 V, V_{CCPD} must be powered up to 3.0 V. If the V_{CCIO} of the bank is set to 2.5 V or lower, V_{CCPD} must be powered up to 2.5 V. This applies for all the banks containing the VCCPD and VCCIO pins.

For more information about the configuration pins power supply, refer to "Device Configuration Pins" on page 9–40.

Configuration Sequence

The following sections describe the general configuration process for the FPP, AS, and PS schemes.

Power Up

To begin the configuration process, you must fully power-up all the power supplies monitored by the POR circuitry to the appropriate voltage levels. The power supplies must ramp-up monotonically within the specified ramp-up time to ensure successful configuration.

All power supplies including the V_{CCPGM} and V_{CCPD} must ramp-up from 0 V to the desired voltage level within the ramp-up time specification. If these supplies are not ramped up within this specified time, your Stratix V device will not configure successfully. If your system cannot ramp-up the power supplies within the specified ramp-up time specification, you must hold nCONFIG low until all the power supplies are stable.

For more information about the ramp-up time specification, refer to the *Hot Socketing and Power-On Reset in Stratix V Devices* chapter.

Reset

After power-up, the Stratix V device goes through a POR. The POR delay depends on the MSEL settings. During POR, the device resets, holds nSTATUS low, clears the configuration RAM bits, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins remain tri-stated until the device is configured.

While nCONFIG is low, the device is in reset. When the device comes out of reset, nCONFIG must be at a logic-high level in order for the device to release the open-drain nSTATUS pin. After nSTATUS is released, it is pulled high by a pull-up resistor and the device is ready to receive configuration data. Before and during configuration, all user I/O pins are tri-stated. If nIO_pullup is driven low during power up and configuration, the user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are on after the device exits POR, before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

For more information about the POR delay specification, refer to "POR Delay Specification" on page 9–2.

Configuration

Both nCONFIG and nSTATUS must be deasserted at a logic-high level in order for the configuration stage to begin. For the FPP and PS configuration schemes, the device receives configuration data on its DATA pins and the clock source on the DCLK pin. Configuration data is latched into the Stratix V device on the rising edge of DCLK. For the AS configuration scheme, the device receives configuration data on its AS_DATA[] pins and drives the clock source on the DCLK pin. Configuration data is latched into the Stratix V device on the falling edge of DCLK.

After the Stratix V device has received all the configuration data successfully, it releases the CONF_DONE pin, which is pulled high by a pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration has completed and initialization of the device can begin. For the FPP and PS schemes, DCLK must not be left floating at the end of configuration. You must drive them either high or low, whichever is convenient on your board.

For the FPP and PS schemes, there is no maximum DCLK period, which means you can stop the configuration by holding the DCLK low for an indefinite amount of time. To resume configuration, the external host must provide data on the DATA[] pins prior to sending the first DCLK rising edge.

Configuration Error

If the **Auto-restart configuration after error** option (available in the Quartus® II software from the **General** panel of the **Device and Pin Options** dialog box) is turned on, the Stratix V device releases the nSTATUS pin high after the specified time indicated by tSTATUS and retries the configuration. If this option is turned off or if you are using a PS or FPP scheme with an external controller, the system must monitor the nSTATUS for errors and sends a low-to-high signal on nCONFIG for specified t_{CFG} time to restart the configuration.

Initialization

In Stratix V devices, initialization begins after the CONF_DONE goes high. For the FPP and PS configuration schemes, two DCLK falling edges are required after the last configuration byte is sent to the Stratix V device to begin the initialization of the device for both uncompressed and compressed configuration data.

The initialization clock source is from the internal oscillator, CLKUSR, or DCLK pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix V device provides itself with enough clock cycles for proper initialization.

Table 9–3 lists the initialization clock source option, the applicable configuration schemes, and the maximum frequency.

Table 9–3. Initialization Clock Source Option and the Maximum Frequency

Initialization Clock Source	Configuration Schemes	Maximum Frequency	Minimum Number of Clock Cycles (1)
Internal Oscillator	AS, PS, FPP	12.5 MHz	17,408 (3)
CLKUSR	AS, PS, FPP (2)	125 MHz	

Notes to Table 9–3:

(1) The minimum number of clock cycles required for device initialization.

(2) To enable CLKUSR as the initialization clock source, turn on the **Enable user-supplied start-up clock (CLKUSR)** option in the Quartus II software from the **General** panel of the **Device and Pin Options** dialog box.

(3) The number is still preliminary.

If you use the optional CLKUSR pin as the initialization clock source and nCONFIG is pulled low to restart configuration during device initialization, ensure that CLKUSR or DCLK continues toggling until nSTATUS goes low and goes high again.

CLKUSR provides you with the flexibility to synchronize initialization of multiple devices or to delay initialization. Supplying a clock on the CLKUSR pin during initialization does not affect configuration. After CONF_DONE goes high, CLKUSR or DCLK is enabled after the time specified by t_{CD2CU}. When this time period elapses, Stratix V devices require a minimum number of clock cycles to initialize properly and enter user mode as specified by the t_{CD2UMC} parameter.

User Mode

The Stratix V device enters user mode when the initialization is complete. You can monitor the end of the initialization stage by enabling the optional INIT_DONE pin. If enabled, the low-to-high transition of INIT_DONE indicates the device has completed initialization and entered user mode. In this mode, your design is executed. The user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

At any time during the configuration stage or user mode operation, you can initiate a reconfiguration by setting a low pulse on the nCONFIG pin. The pulse must meet the minimum t_{CFG} low-pulse width. When nCONFIG is pulled low, the nSTATUS and CONF_DONE pins are pulled low and all I/O pins are tri-stated. Configuration begins when the nCONFIG and nSTATUS pins return to a logic-high level.

Configuration Schemes

The following sections describe configuration schemes for Stratix V devices.

MSEL Pin Settings

Select the configuration scheme by driving the Stratix V device MSEL pins either high or low (refer to Table 9–4). The MSEL input buffers are powered by the V_{CCPGM} power supply. During POR and during reconfiguration, the MSEL pins must be at the LVTTL V_{IL} and V_{IH} levels to be considered logic low and logic high, respectively.

To avoid problems with detecting an incorrect configuration scheme, hardwire the MSEL pins to V_{CCPGM} or GND without pull-up or pull-down resistors. Use only the MSEL pin settings listed in Table 9–4. Do not drive the MSEL pins with a microprocessor or another device.

Table 9–4 lists the configuration schemes for Stratix V devices.

Table 9–4. Configuration Schemes for Stratix V Devices (1) (Part 1 of 2)

Configuration Scheme	Decompression Feature	Design Security Feature	Configuration Voltage Standard (V) (3)	POR Delay (6)	MSEL[4..0]
FPP ×8	Disabled	Disabled	1.8/2.5/3.0	Fast	10100
				Standard	11000
	Disabled	Enabled	1.8/2.5/3.0	Fast	10101
				Standard	11001
	Enabled	Optional (2)	1.8/2.5/3.0	Fast	10110
				Standard	11010
FPP ×16	Disabled	Disabled	1.8/2.5/3.0	Fast	00000
				Standard	00100
	Disabled	Enabled	1.8/2.5/3.0	Fast	00001
				Standard	00101
	Enabled	Optional (2)	1.8/2.5/3.0	Fast	00010
				Standard	00110
FPP ×32	Disabled	Disabled	1.8/2.5/3.0	Fast	01000
				Standard	01100
	Disabled	Enabled	1.8/2.5/3.0	Fast	01001
				Standard	01101
	Enabled	Optional (2)	1.8/2.5/3.0	Fast	01010
				Standard	01110
PS	Optional (2)	Optional (2)	1.8/2.5/3.0	Fast	10000
				Standard	10001
AS (×1, ×4) (4)	Optional (2)	Optional (2)	3.0	Fast	10010
				Standard	10011

Table 9–4. Configuration Schemes for Stratix V Devices (1) (Part 2 of 2)

Configuration Scheme	Decompression Feature	Design Security Feature	Configuration Voltage Standard (V) (3)	POR Delay (6)	MSEL[4..0]
JTAG-based configuration (5)	Disabled	Disabled	—	—	(7)

Notes to Table 9–4:

(1) Do not leave the MSEL pins floating. Connect them to V_{CCPGM} or GND. Use only the MSEL pin settings listed in Table 9–4.

(2) You can select to enable or disable this feature.

(3) The configuration voltage standard applied to the V_{CCPGM} power supply that powers all the configuration pins during configuration.

(4) The AS configuration scheme supports the remote system upgrade feature. For more information about the remote system upgrade feature, refer to "Remote System Upgrades" on page 9–46.

(5) JTAG-based configuration takes precedence over other configuration schemes. This means the MSEL pin settings are ignored. JTAG-based configuration does not support the design security or decompression features.

(6) For POR delay specification, refer to "POR Delay Specification" on page 9–2.

(7) Altera recommends connecting the MSEL pins to GND.

Raw Binary File Size

For the POR delay specification, refer to "POR Delay Specification" on page 9–2.

Table 9–5 lists the uncompressed raw binary file (**.rbf**) sizes for Stratix V devices.

Table 9–5. Uncompressed .rbf Sizes for Stratix V Devices (1)

Family	Device	Configuration .rbf Size (bits)	IOCSR .rbf Size (bits)
Stratix V GX	5SGXA3	213,798,720	1,689,600
	5SGXA4	213,798,720	1,689,600
	5SGXA5	269,978,848	2,239,436
	5SGXA7	269,978,848	2,239,436
	5SGXA9	387,394,048	2,141,440
	5SGXAB	387,394,048	2,141,440
	5SGXB5	270,528,480	2,236,416
	5SGXB6	270,528,480	2,236,416
	5SGXB9	387,394,048	2,141,440
	5SGXBB	387,394,048	2,141,440
Stratix V GT	5SGTC5	269,978,848	2,195,200
	5SGTC7	269,978,848	2,195,200
Stratix V GS	5SGSD3	93,080,448	933,120
	5SGSD4	209,935,224	2,004,480
	5SGSD5	209,935,224	2,004,480
	5SGSD6	266,798,896	2,236,416
	5SGSD8	266,798,896	2,236,416
Stratix V E (2)	5SEE9	387,394,048	2,141,440
	5SEEB	387,394,048	2,141,440

Notes to Table 9–5:

(1) These values are preliminary.

(2) Stratix V E devices do not have PCI Express® (PCIe®) hard IP. Stratix V E devices do not support the CvP configuration scheme.

Use the data in Table 9–5 to estimate the file size before design compilation. Different configuration file formats, such as a hexadecimal (**.hex**) or tabular text file (**.ttf**) format, have different file sizes. For the different types of configuration file and file sizes, refer to the Quartus II software. However, for a specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you are using compression, the file size can vary after each compilation because the compression ratio depends on your design.

For more information about setting device configuration options or creating configuration files, refer to the *Device Configuration Options* and *Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

Fast Passive Parallel Configuration

The FPP configuration using an external host provides the fastest method to configure Stratix V devices. FPP is supported in multiple data widths—8-bits, 16-bits, and 32-bits. You can perform a FPP configuration of Stratix V devices using an external host such as a MAX II device, MAX V device, or microprocessor. The external host controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix V device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** formats. Therefore, the design that controls the configuration stages, such as fetching the data from flash memory and sending it to the device, must be stored in the MAX II device, MAX V device, or microprocessor.

The Parallel Flash Loader (PFL) feature in MAX II and MAX V devices provides an efficient method to program CFI flash memory devices through the JTAG interface. PFL also acts as a controller to read configuration data from the flash memory device and configures the Stratix V device. PFL supports both the PS and FPP configuration schemes.

For more information about the PFL, refer to *Parallel Flash Loader Megafunction User Guide*.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed configuration data in a FPP configuration.

DCLK-to-DATA[] Ratio for FPP configuration

FPP configuration requires a different DCLK-to-DATA[]ratio when you enable the design security, decompression, or both features. Table 9–6 lists the DCLK-to-DATA[]ratio for each combination.

Table 9–6. DCLK-to-DATA[] Ratio (1) (Part 1 of 2)

Configuration Scheme	Decompression	Design Security	DCLK-to-DATA[] Ratio
FPP ×8	Disabled	Disabled	1
	Disabled	Enabled	1
	Enabled	Disabled	2
	Enabled	Enabled	2

Table 9–6. DCLK-to-DATA[] Ratio (1) (Part 2 of 2)

Configuration Scheme	Decompression	Design Security	DCLK-to-DATA[] Ratio
FPP ×16	Disabled	Disabled	1
	Disabled	Enabled	2
	Enabled	Disabled	4
	Enabled	Enabled	4
FPP ×32	Disabled	Disabled	1
	Disabled	Enabled	4
	Enabled	Disabled	8
	Enabled	Enabled	8

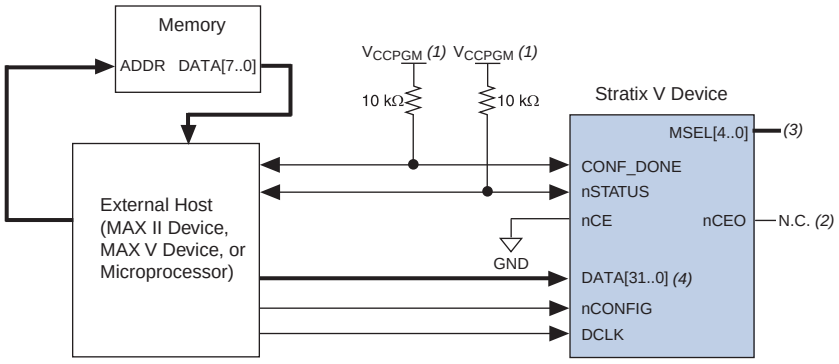
Note to Table 9–6:

(1) Depending on the DCLK-to-DATA[] ratio, the host must send a DCLK frequency that is r times the data rate in bytes per second (Bps), or words per second (Wps). For example, in FPP ×16 when the DCLK-to-DATA[] ratio is 2, the DCLK frequency must be 2 times the data rate in Wps. Stratix V devices use the additional clock cycles to decrypt and decompress the configuration data.

If the DCLK-to-DATA[] ratio is greater than 1, at the end of configuration, you can only stop the DCLK (DCLK-to-DATA[] ratio – 1) clock cycles after the last data is latched into the Stratix V device.

Figure 9–1 shows the configuration interface connections between the Stratix V device and a MAX II or MAX V device for single device configuration.

Figure 9–1. Single Device FPP Configuration Using an External Host



Notes to Figure 9–1:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different data width, configuration voltage standards, and POR delay. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) If you use FPP ×8, use DATA[7..0]. If you use FPP ×16, use DATA[15..0].

FPP Multi-Device Configuration

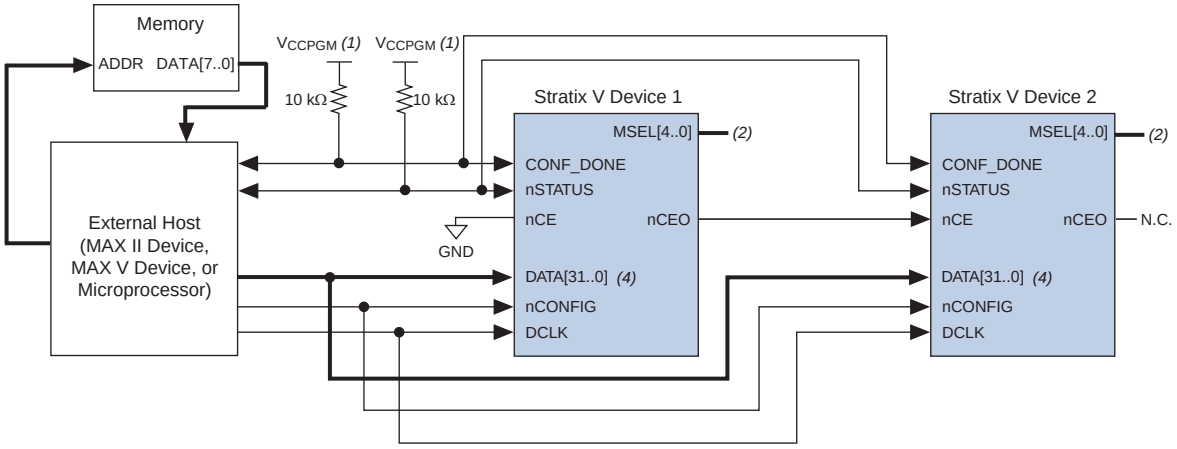
For FPP multi-device configuration, you can configure all devices with different sets of configuration data (multiple SRAM object files [**.sof**s]) or with the same configuration data (single **.sof**). In both cases, the nCONFIG, nSTATUS, DCLK, DATA[], and CONF_DONE pins are connected to every device in the chain. Ensure that the DCLK and data line are buffered for every fourth device. This ensures the signal integrity and prevents clock skew problems.

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

☞ For FPP multi-device configuration, all devices in the chain must have the same data width. If you are using FPP ×32, all devices in the chain must use FPP ×32 configuration scheme. If you are using FPP ×8, you can use the Stratix V device with other FPGA devices that support FPP ×8.

Figure 9–2 shows how to configure multiple devices using a MAX II or MAX V device when both devices receive a different set of configuration data (multiple **.sof**s). You must convert the multiple **.sof**s into a single programming file for the FPP chain programming.

Figure 9–2. Multi-Device FPP Configuration Using an External Host When Both Devices Receive a Different Set of Configuration Data



Notes to Figure 9–2:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different data width and POR delay. To connect MSEL, refer to Table 9–4 on page 9–7.

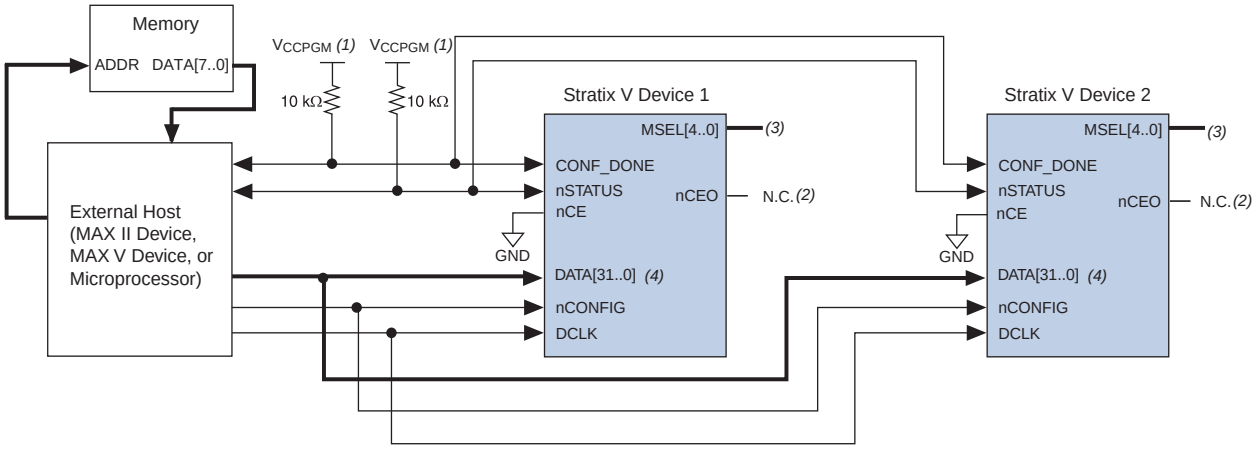
(4) If you use FPP ×8, use DATA[7..0]. If you use FPP ×16, use DATA[15..0]. All devices in the chain must have the same data width.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–2, after the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data to the second device is transparent to the MAX II device, MAX V device, or microprocessor.

Figure 9–3 shows the FPP configuration setup for multiple devices when both Stratix V devices receive the same configuration data (single **.sof**).

Figure 9–3. Multiple Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 9–3:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different data width and POR delay. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) If you use FPP ×8, use DATA[7..0]. If you use FPP ×16, use DATA[15..0]. All devices in the chain must have the same data width.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–3, because both nCE pins are tied to GND, both devices in the chain begin and complete the configuration and enter user mode at the same time.

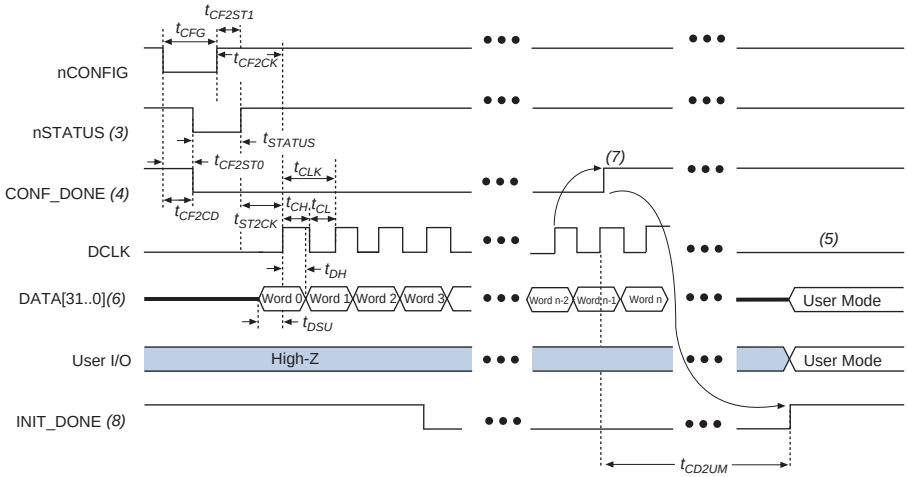
To configure FPP multi-device with a single **.sof**, all Stratix V devices in the chain must be in the same package and density.

FPP Configuration Timing

Figure 9–4 shows the timing waveform for FPP configuration when using a MAX II or MAX V device as an external host. This waveform shows timing when the DCLK-to-DATA[] ratio is 1.

When you enable the decompression or design security feature, the DCLK-to-DATA[] ratio varies for FPP ×8, FPP ×16, and FPP ×32. For the respective DCLK-to-DATA[] ratio, refer to Table 9–6 on page 9–9.

Figure 9–4. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is 1 *(1), (2)*



Notes to Figure 9–4:

(1) Use this timing waveform when the DCLK-to-DATA[] ratio is 1.

(2) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic-high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) After power-up, the Stratix V device holds nSTATUS low for the time of the POR delay.

(4) After power-up, before and during configuration, CONF_DONE is low.

(5) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(6) For FPP ×16, use DATA[15..0]. For FPP ×8, use DATA[7..0]. DATA[31..0] are available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.

(7) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high when the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(8) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–7 lists the timing parameters for Stratix V devices for FPP configuration when the DCLK-to-DATA[] ratio is 1.

Table 9–7. FPP Timing Parameters for Stratix V Devices *(1), (2)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(4)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	$0.45 \times 1/f_{MAX}$	—	s
t_{CL}	DCLK low time	$0.45 \times 1/f_{MAX}$	—	s
t_{CLK}	DCLK period	$1/f_{MAX}$	—	s
f_{MAX}	DCLK frequency (FPP ×8/×16)	—	125	MHz
	DCLK frequency (FPP ×32)	—	100	MHz
t_{CD2UM}	CONF_DONE high to user mode *(5)*	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period) *(6)*	—	—

Notes to Table 9–7:

(1) This information is preliminary.

(2) Use these timing parameters when the decompression and design security features are disabled.

(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

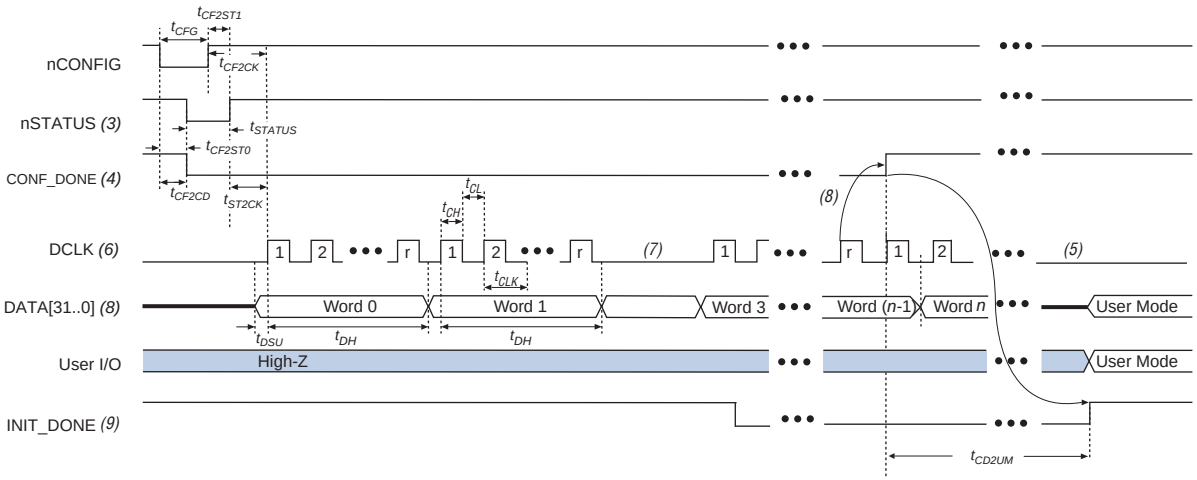
(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.

(5) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for initializing the device.

(6) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

Figure 9–5 shows the timing waveform for FPP configuration when using a MAX II device, MAX V device, or microprocessor as an external host. This waveform shows timing when the DCLK-to-DATA[] ratio is more than 1.

Figure 9–5. FPP Configuration Timing Waveform When the DCLK-to-DATA[] Ratio is >1 *(1), (2)*



Notes to Figure 9–5:

(1) Use this timing waveform and parameters when the DCLK-to-DATA[] ratio is >1. To find out the DCLK-to-DATA[] ratio for your system, refer to Table 9–6 on page 9–9.

(2) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(3) After power-up, the Stratix V device holds nSTATUS low for the time as specified by the POR delay.

(4) After power-up, before and during configuration, CONF_DONE is low.

(5) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(6) "r" denotes the DCLK-to-DATA[] ratio. For the DCLK-to-DATA[] ratio based on the decompression and the design security feature enable settings, refer to Table 9–6 on page 9–9.

(7) If needed, pause DCLK by holding it low. When DCLK restarts, the external host must provide data on the DATA[31..0] pins prior to sending the first DCLK rising edge.

(8) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high after the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(9) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

If you are using the Stratix V decompression and design security features, configuration data is latched on the rising edge of every first DCLK cycle out of the N DCLK cycles where N is the DCLK-to-DATA ratio. Altera recommends keeping the data on DATA[31..0] stable for the next N–1 clock cycles when the data is being processed.

When you are using the decompression and design security features, you can only stop DCLK during N–1 clock cycles after the last data byte was latched into the Stratix V device.

Table 9–8 lists the timing parameters for Stratix V devices for FPP configuration when the DCLK-to-DATA[] ratio is more than 1.

Table 9–8. FPP Timing Parameters for Stratix V Devices When the DCLK-to-DATA[] Ratio is >1 *(1), (2)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(3)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	N–1/f_{DCLK} *(6)*	—	s
t_{CH}	DCLK high time	0.45 × 1/f_{MAX}	—	s
t_{CL}	DCLK low time	0.45 × 1/f_{MAX}	—	s
t_{CLK}	DCLK period	1/f_{MAX}	—	s
f_{MAX}	DCLK frequency (FPP ×8/×16)	—	125	MHz
	DCLK frequency (FPP ×32)	—	100	MHz
t_R	Input rise time	—	40	ns
t_F	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period) *(5)*	—	—

Notes to Table 9–8:

(1) This information is preliminary.

(2) Use these timing parameters when you use the decompression and design security features.

(3) You can obtain this value if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.

(4) The minimum and maximum numbers apply only if you use the internal oscillator as the clock source for initializing the device.

(5) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

(6) N is the DCLK-to-DATA ratio and f_{DCLK} is the DCLK frequency the system is operating.

Active Serial Configuration (Serial Configuration Devices)

The AS configuration scheme supports AS ×1 (1-bit data width) and AS ×4 (4-bit data width) modes. Serial configuration device (EPCS) supports AS ×1 mode and quad-serial configuration device (EPCQ) supports AS ×1 and AS ×4 modes. The AS ×4 mode provides four times faster configuration time than the AS ×1 mode.

EPCS and EPCQ are low-cost devices with non-volatile memory that feature a simple four-pin interface or six-pin interface, respectively, and a small form factor. These features make the AS configuration scheme an ideal low-cost configuration solution.

If you wish to gain control of the EPCS pins, hold the nCONFIG pin low and pull the nCE pin high. This causes the device to reset and tri-state the AS configuration pins.

For more information about EPCS and EPCQ devices, refer to the volume 2 of the *Configuration Handbook*.

AS mode supports DCLK frequency up to 100 MHz. You can choose the CLKUSR or internal oscillator as the configuration clock source that drives DCLK. If you use the internal oscillator as the configuration clock source, you can choose a 12.5, 25, 50, or 100 MHz clock from the **Configuration** panel in the **Device and Pins Option** settings.

Table 9–9 lists the DCLK frequency specification in the AS configuration scheme.

Table 9–9. DCLK Frequency Specification in the AS Configuration Scheme *(1), (2), (3)*

Minimum	Typical	Maximum	Unit
5.3	7.9	12.5	MHz
10.6	15.7	25.0	MHz
21.3	31.4	50.0	MHz
42.6	62.9	100.0	MHz

Notes to Table 9–9:

(1) This information is preliminary.

(2) This applies to the DCLK frequency specification when using the internal oscillator as the configuration clock source.

(3) The AS multi-device configuration scheme does not support DCLK frequency of 100 MHz.

You can choose the internal oscillator or CLKUSR as the DCLK clock source by selecting the option under **Device and Pins Option** settings, **Configuration** panel in the Quartus II software. This sets specific option in the programming file. By default, in AS scheme, Stratix V devices power-up and begin configuration with 12.5 MHz internal oscillator as the DCLK clock source. After reading the option bits from the programming file, Stratix V devices continue using the internal oscillator at 12.5 MHz frequency, switch to a higher internal oscillator clock frequency, or switch to the CLKUSR pin.

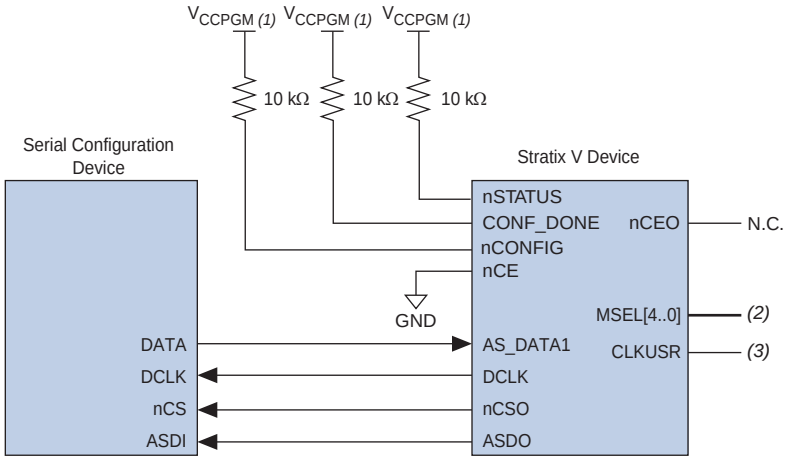
If you choose CLKUSR as configuration clock source, the maximum frequency allowed is 100 MHz.

During device configuration, Stratix V devices read the configuration data using the serial interface, decompress the data if necessary, and configure their SRAM cells. In the AS scheme, the Stratix V device controls the configuration interface. In the PS scheme, the external host (a MAX II device, MAX V device, or microprocessor) controls the interface.

You can select between the AS ×1 and AS ×4 settings by selecting the option under **Device and Pins Option** settings, **Configuration** panel in the Quartus II software. This sets a specific option bit in the programming file. By default, in the AS scheme, Stratix V devices power-up and begin configuration as an AS ×1 mode. Upon reading the option bits from the programming file, Stratix V devices either stay as an AS ×1 mode or switch to an AS ×4 mode for the rest of the configuration.

Figure 9–6 shows the single-device configuration setup for an AS ×1 mode.

Figure 9–6. Single Device AS ×1 Mode Configuration



Notes to Figure 9–6:

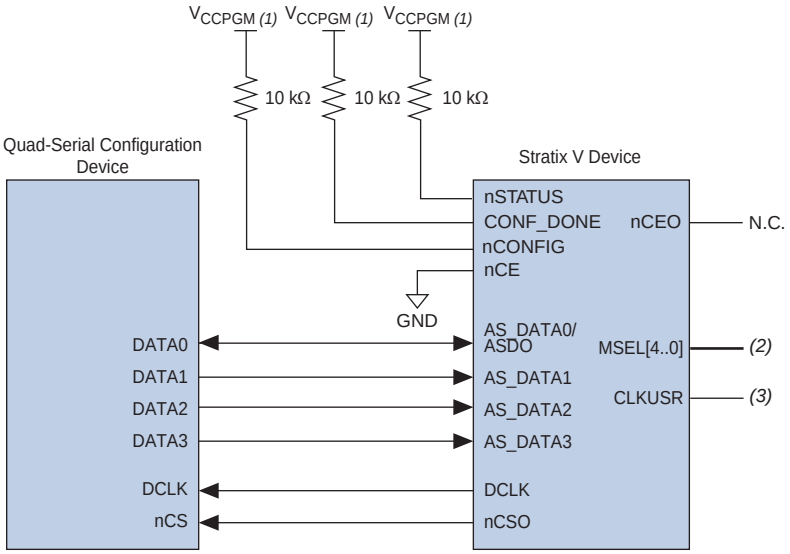
(1) Connect the pull-up resistors to V_{CCPGM} at 3.0-V supply.

(2) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(3) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–7 shows the single-device configuration setup for an AS ×4 mode.

Figure 9–7. Single Device AS ×4 Mode Configuration



Notes to Figure 9–7:

(1) Connect the pull-up resistors to V_{CCPGM} at 3.0-V supply.

(2) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(3) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

The serial clock (DCLK) generated by the Stratix V device controls the entire configuration cycle and provides timing for the serial interface. Stratix V devices use an internal oscillator or external clock (CLKUSR) as the configuration clock source (DCLK). In the AS configuration scheme, Stratix V devices drive out control signals on the falling edge of DCLK and latch in the data on the following falling edge of DCLK.

During configuration, Stratix V devices enable the EPCS or EPCQ by driving the nCSO output pin low, which connects to the chip select (nCS) pin of the EPCS or EPCQ. Stratix V devices use the serial clock (DCLK) and serial data output (ASDO) pins to send operation commands and read address signals to the EPCS or EPCQ. The EPCS or EPCQ provides data on its serial data output (DATA[]) pin, which connects to the AS_DATA[] input of the Stratix V devices.

AS Multi-Device Configuration

For the AS multi-device configuration scheme, you can configure all devices with different sets of configuration data (different **.sof**) or with the same configuration data (same **.sof**). In both cases, the nCONFIG, nSTATUS, DCLK, and data line (AS_DATA1 on the master device and DATA0 on the slave device) and CONF_DONE pins are connected to every device in the chain. Ensure that the DCLK and data line are buffered for every fourth device.

☞ The AS configuration scheme supports multi-device in AS ×1 mode. The AS ×4 mode does not support multi-device configuration setup.

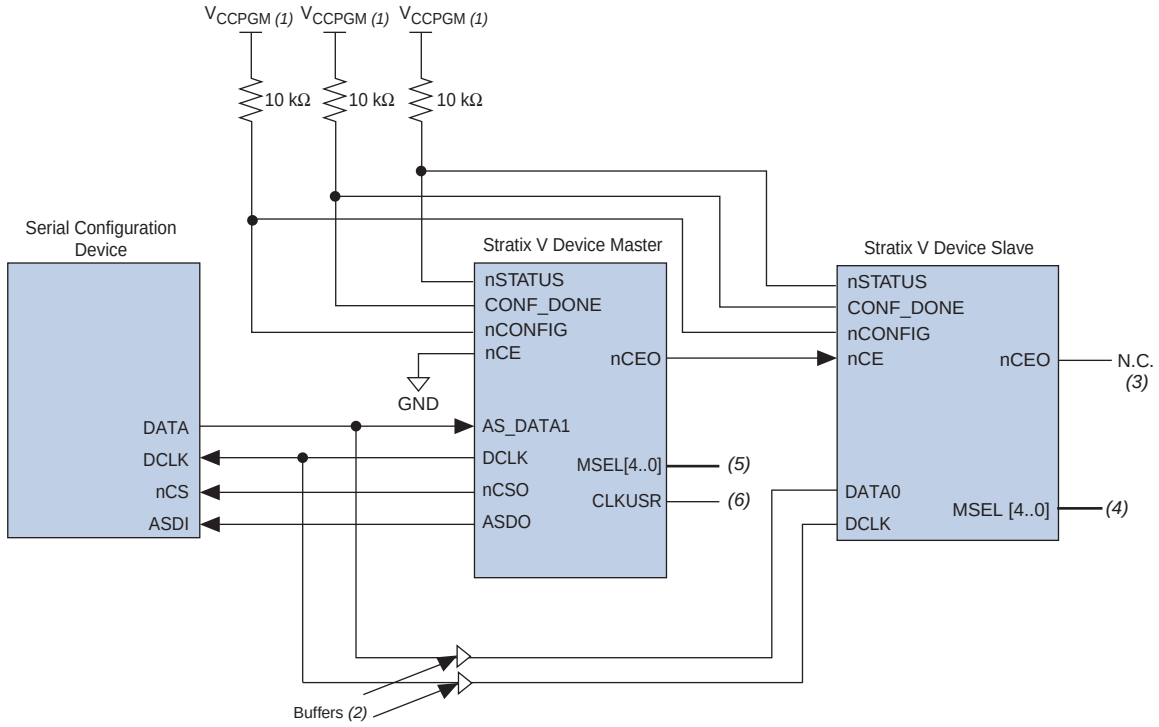
☞ The AS multi-device configuration scheme does not support DCLK frequency of 100 MHz.

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

☞ In this configuration scheme, the first Stratix V device in the chain is the configuration master and controls the configuration of the entire chain. You must connect its MSEL pins to select the AS configuration scheme. The remaining Stratix V devices are configuration slaves. You must connect their MSEL pins to select the PS configuration scheme. Any other Altera® device that supports a PS configuration can also be part of the chain as the configuration slave.

Figure 9–8 shows the multi-device configuration setup for AS ×1 mode when both devices in the chain receive different sets of configuration data (multiple **.sof**s).

Figure 9–8. AS Multi-Device Configuration When Both Devices in the Chain Receive Different Sets of Configuration Data *(1), (2)*



Notes to Figure 9–8:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.

(2) Connect the repeater buffers between the Stratix V master and slave device for AS_DATA1/DATA0 and DCLK.

(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(4) For the appropriate MSEL settings based on POR delay settings, set the slave device MSEL setting to the PS scheme. Refer to Table 9–4 on page 9–7.

(5) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

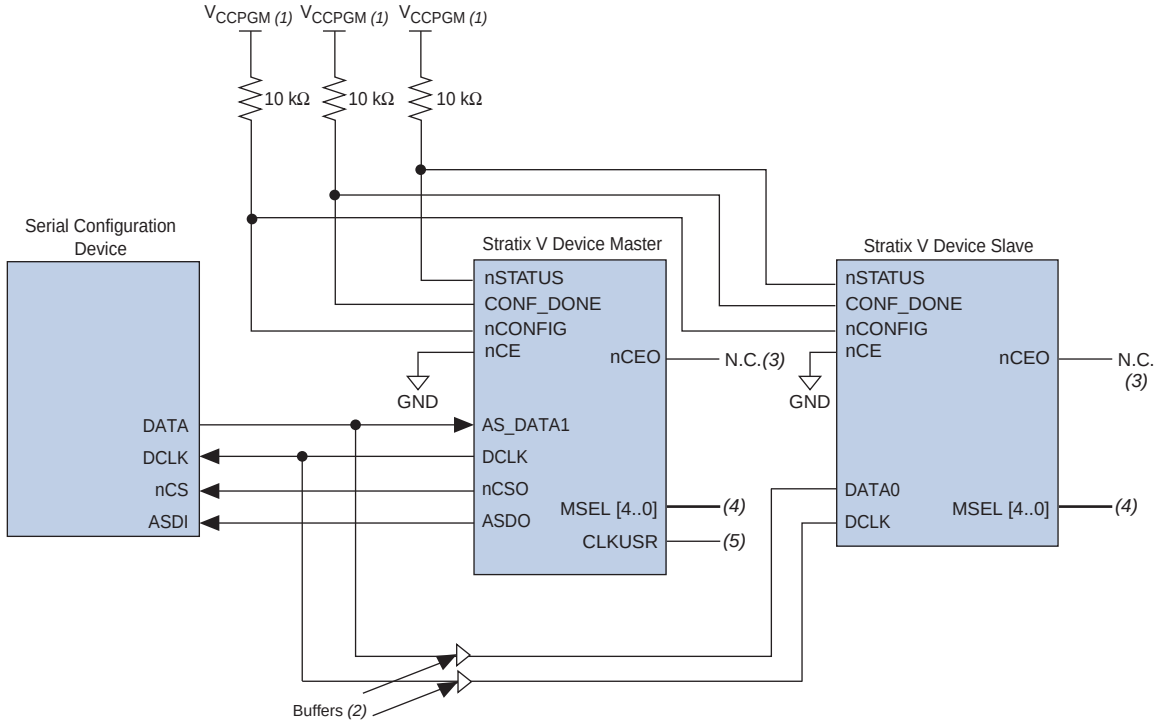
(6) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration.

(7) For 50 MHz frequency, delay DCLK in reference to DATA0 by a minimum of 5 ns and a maximum of 10 ns.

In Figure 9–8, after the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data to the second device is transparent to the first device in the chain.

Figure 9–9 shows the multi-device configuration setup for AS ×1 mode when all devices in the chain receive the same set of configuration data (single **.sof**).

Figure 9–9. AS Multi-Device Configuration When the Devices Receive the Same Data Using a Single .sof *(1)*



Notes to Figure 9–9:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V supply.
(2) Connect the repeater buffers between the Stratix V master and slave device for AS_DATA1/DATA0 and DCLK.
(3) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.
(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.
(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration.
(6) For 50 MHz frequency, delay DCLK in reference to DATA0 by a minimum of 5 ns and a maximum of 10 ns.

AS Connection Guidelines

Table 9–10 lists the board trace length and loading between the supported EPCS or EPCQ and Stratix V devices for single- and multi-device AS configurations.

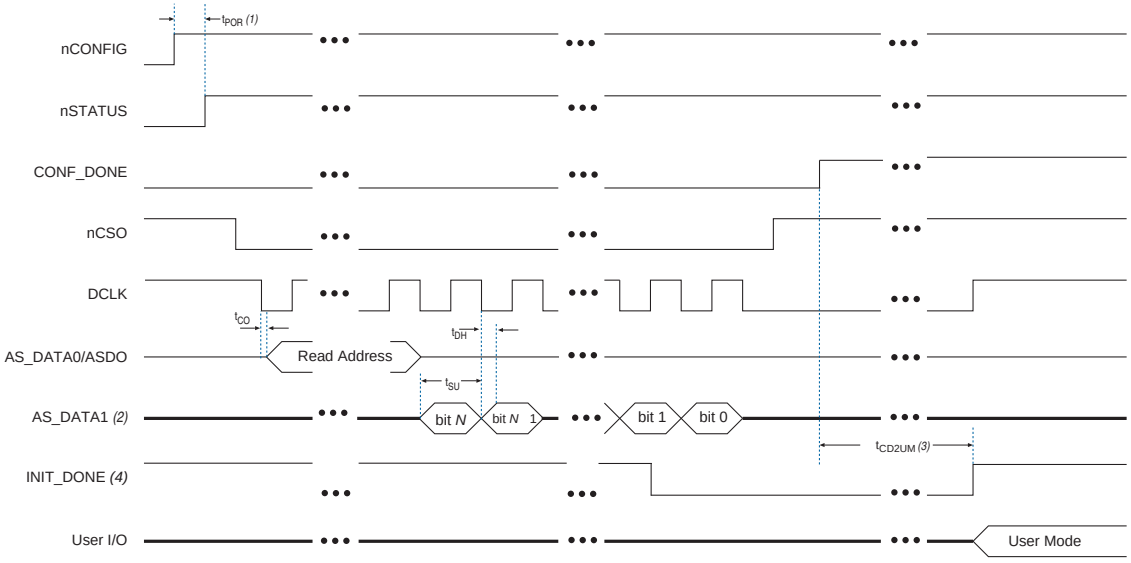
Table 9–10. Maximum Trace Length and Loading for AS ×1/×4 Configurations

Stratix V Device AS Pins	Maximum Board Trace Length from the Stratix V Device to the Serial Configuration Device for 12.5/ 25/ 50 MHz Operation (Inches)	Maximum Board Trace Length from the Stratix V Device to the Serial Configuration Device for 100 MHz Operation (Inches)	Maximum Board Load (pF)
DCLK	10	6	5
DATA[3..0]	10	6	10
nCSO	10	6	10

AS Configuration Timing

Figure 9–10 shows the timing waveform for the AS ×1 mode and AS ×4 mode configuration timing.

Figure 9–10. AS Configuration Timing



Notes to Figure 9–10:

(1) The AS scheme supports standard and fast POR delay (t_{POR}). For t_{POR} delay information, refer to "POR Delay Specification" on page 9–2.
(2) If you are using AS ×4 mode, this signal represents the AS_DATA[3..0] and EPCQ sends in 4-bits of data for each DCLK cycle.
(3) The initialization clock can be from internal oscillator or CLKUSR pin.
(4) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–11 lists the timing parameters for AS ×1 and AS ×4 configurations in Stratix V devices.

Table 9–11. AS Timing Parameters for AS ×1 and AS ×4 Configurations in Stratix V Devices *(1), (2), (3)*

Symbol	Parameter	Minimum	Maximum	Units
t_{CO}	DCLK falling edge to AS_DATA0/ASDO output	—	4	µs
t_{SU}	Data setup time before falling edge on DCLK	1.5	—	ns
t_{H}	Data hold time after falling edge on DCLK	0	—	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period)	—	—

Notes to Table 9–11:

(1) This information is preliminary.

(2) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for initializing the device.

(3) t_{CF2CD}, t_{CF2ST0}, t_{CFG}, t_{STATUS}, and t_{CF2ST1} timing parameters are identical to the timing parameters for PS mode listed in Table 9–12 on page 9–32.

(4) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on this pin, refer to "Initialization" on page 9–5.

Estimating the Active Serial Configuration Time

The AS configuration time is dominated by the time it takes to transfer data from the EPCS to the Stratix V device. This serial interface is clocked by the Stratix V DCLK.

You can estimate the minimum AS ×1 mode configuration time by using the following equation:

.rbf Size × (minimum DCLK period / 1 bit per DCLK cycle) = estimated minimum configuration time.

You can estimate the minimum AS ×4 mode configuration time by using the following equation:

.rbf Size × (minimum DCLK period / 4 bits per DCLK cycle) = estimated minimum configuration time.

Enabling compression reduces the amount of configuration data that is transmitted to the Stratix V device, which also reduces the configuration time. Your configuration time is reduced based on the compression ratio. The compression ratio varies based on the design.

Programming EPCS and EPCQ

EPCS and EPCQ are non-volatile, flash-memory-based devices. You can program these devices in-system using a USB-Blaster™, EthernetBlaster, or ByteBlaster™ II download cable. Alternatively, you can program EPCS or EPCQ device using a microprocessor with the SRunner software driver.

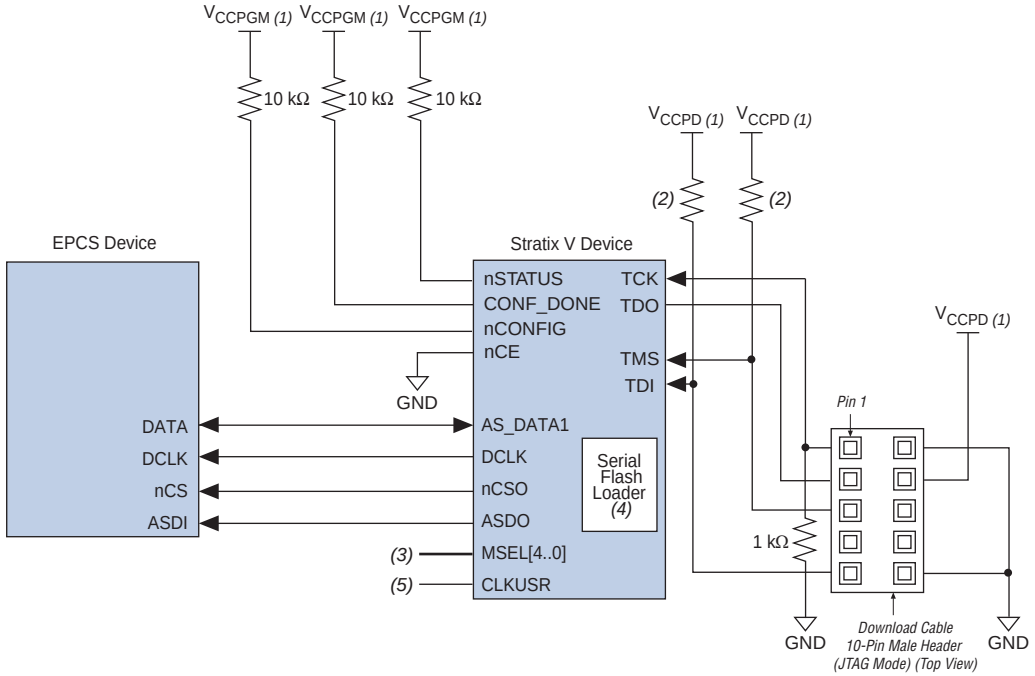
If you are not using Quartus II software or SRunner software for EPCQ 256 programming, put your EPCQ 256 device into four-byte addressing mode before you program and configure your device.

For more information about SRunner software driver, refer to *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming*.

In-system programming offers you an option to program the EPCS or EPCQ device either using an AS programming interface or a JTAG interface. Using the AS programming interface, the configuration data is programmed into the EPCS by the Quartus II software or any supported third-party software. Using the JTAG interface, an Altera IP called Serial Flash Loader (SFL) must be downloaded into the Stratix V device to form a bridge between the JTAG interface and the EPCS or EPCQ device. This allows the EPCS or the EPCQ device to be programmed directly using the JTAG interface.

Figure 9–11 shows the connection setup when programming the EPCS device using the JTAG interface.

Figure 9–11. Connection Setup for Programming the EPCS Device Using the JTAG Interface



Notes to Figure 9–11:

(1) Connect the pull-up resistors to V_{CCPGM} at a 3.0-V power supply. For more information about how to connect to V_{CCPD}, refer to "V_{CCPD} Pin" on page 9–3.

(2) Resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

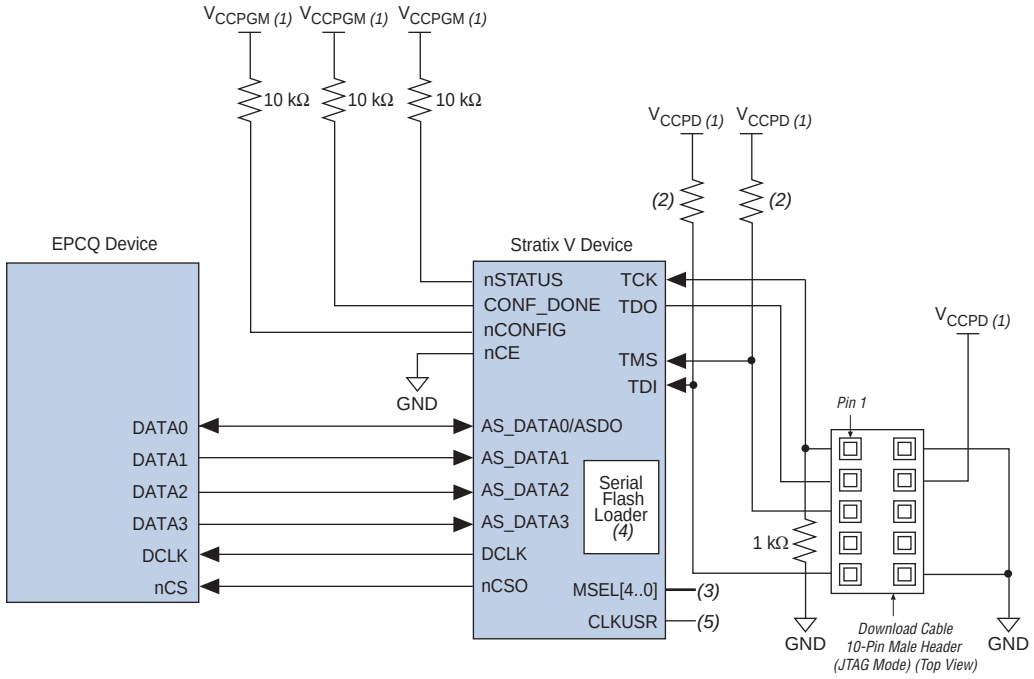
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) Instantiate SFL in your design to form a bridge between the EPCS device and the Stratix V device. For more information about SFL, refer to *AN 370: Using the Serial Flash Loader with the Quartus II Software*.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–12 shows the connection setup when programming the EPCQ device using the JTAG interface.

Figure 9–12. Connection Setup for Programming the EPCQ Device Using the JTAG Interface



Notes to Figure 9–12:

(1) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

(2) Resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

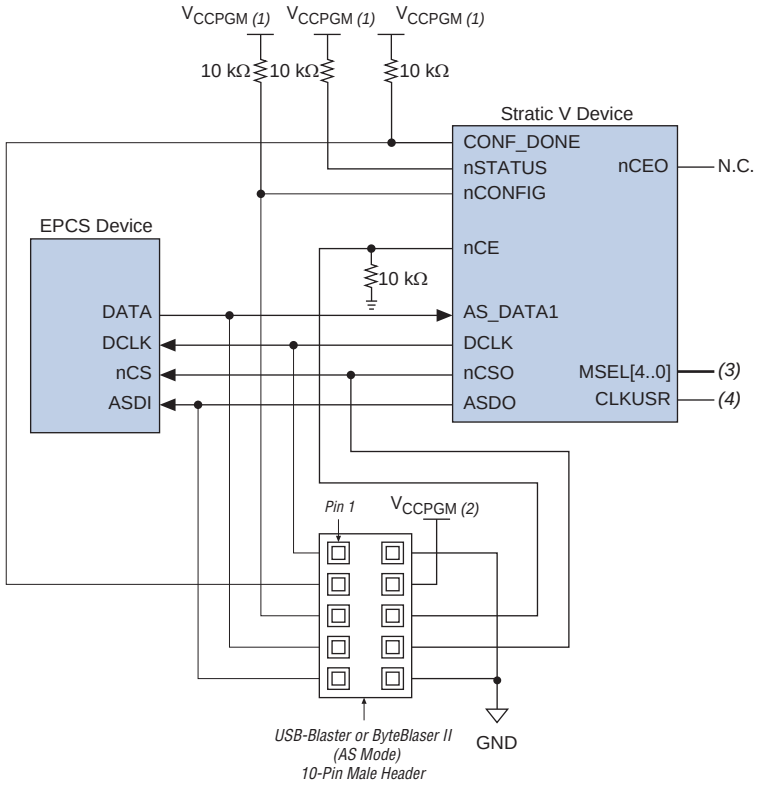
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) Instantiate SFL in your design to form a bridge between the EPCS device and the Stratix V device. For more information about SFL, refer to *AN 370: Using the Serial Flash Loader with the Quartus II Software*.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–13 shows the connection setup when programming the EPCS device using the AS interface.

Figure 9–13. Connection Setup for Programming the EPCS Device Using the AS Interface



Notes to Figure 9–13:

(1) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

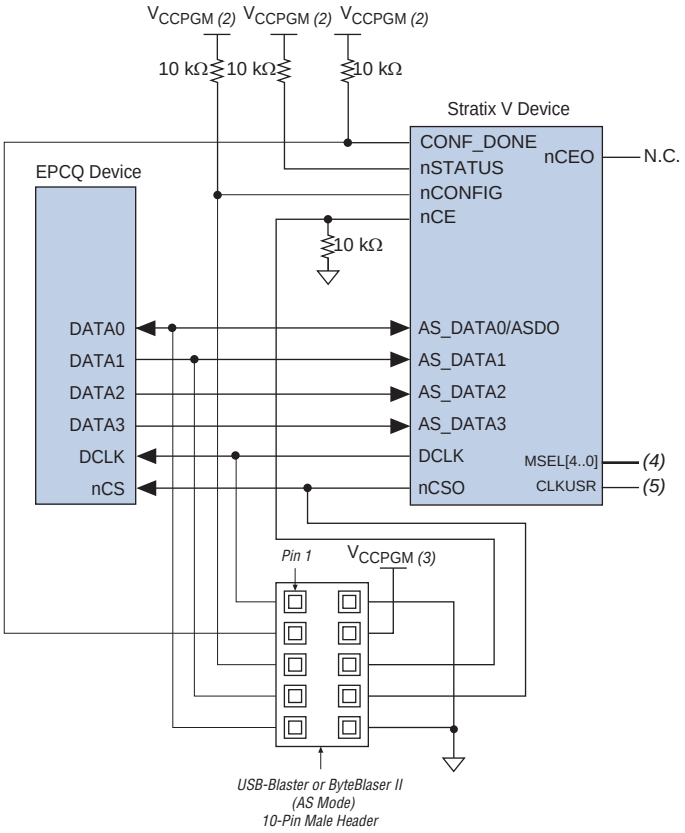
(2) Power up the USB-ByteBlaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(4) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

Figure 9–14 shows the connection setup when programming the EPCQ device using the AS interface.

Figure 9–14. Connection Setup for Programming the EPCQ Device Using the AS Interface *(1)*



Notes to Figure 9–14:

(1) Using the AS header, the programmer transmits the operation commands and the configuration bits to the EPCQ device serially on DATA0.This is equivalent to the programming operation for the EPCS device as shown in Figure 9–14.

(2) Connect the pull-up resistors to V_{CCPGM} and V_{CCPD} at a 3.0-V supply.

(3) Power up the USB-ByteBlaster, ByteBlaster II, or EthernetBlaster cable's $V_{CC(TRGT)}$ with V_{CCPGM}.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect the MSEL, refer to Table 9–4 on page 9–7.

(5) You can use the CLKUSR pin to supply the external clock source to drive the DCLK during configuration. The maximum frequency specification is 100 MHz.

During EPCS and EPCQ programming, the download cable disables the device access to the AS interface by driving the nCE pin high. The nCONFIG line is also pulled low to hold the Stratix V device in reset stage. After programming completes, the download cable releases nCE and nCONFIG, allowing the pull-down and pull-up resistors to drive the pin to GND and V_{CCPGM}, respectively.

During EPCQ programming using the download cable, DATA0 carries the programming data, operation command, and address information from the download cable into the EPCQ device. During EPCQ verification using the download cable, DATA1 carries the programming data back to the download cable.

Passive Serial Configuration

You can perform PS configuration of Stratix V devices using an external host such as a MAX II device, MAX V device, microprocessor, or a host PC. Therefore, the design that controls the configuration stages, such as fetching the data from flash memory and sending it to the device, must be stored in the external host.

The Parallel Flash Loader (PFL) feature in MAX II or MAX V devices provide an efficient method to program CFI flash memory devices through the JTAG interface. PFL also acts as a controller to read configuration data from the flash memory device and configures the Stratix V device. PFL supports both the PS and FPP configuration schemes.

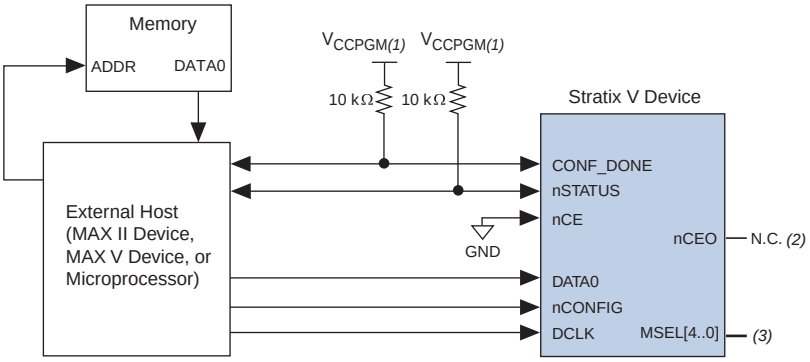
For more information about the PFL, refer to the *Parallel Flash Loader Megafunction User Guide*.

PS Configuration Using a MAX II Device, MAX V Device, or Microprocessor

The external host (a MAX II device, MAX V device, or microprocessor) reads configuration data from storage devices, such as flash memory, and transfers it to Stratix V devices. You can store the configuration data in **.pof**, **.rbf**, **.hex**, or **.ttf** format. If you are using configuration data in **.rbf**, **.hex**, or **.ttf** format, you must send the LSB of each data byte first. For example, if the **.rbf** file contains the byte sequence 02 1B EE 01 FA, the serial data transmitted to the device must be 0100-0000 1101-1000 0111-0111 1000-0000 0101-1111.

Figure 9–15 shows the configuration interface connections between a Stratix V device and a MAX II or MAX V device for single device configuration.

Figure 9–15. Single Device PS Configuration Using an External Host



Notes to Figure 9–15:

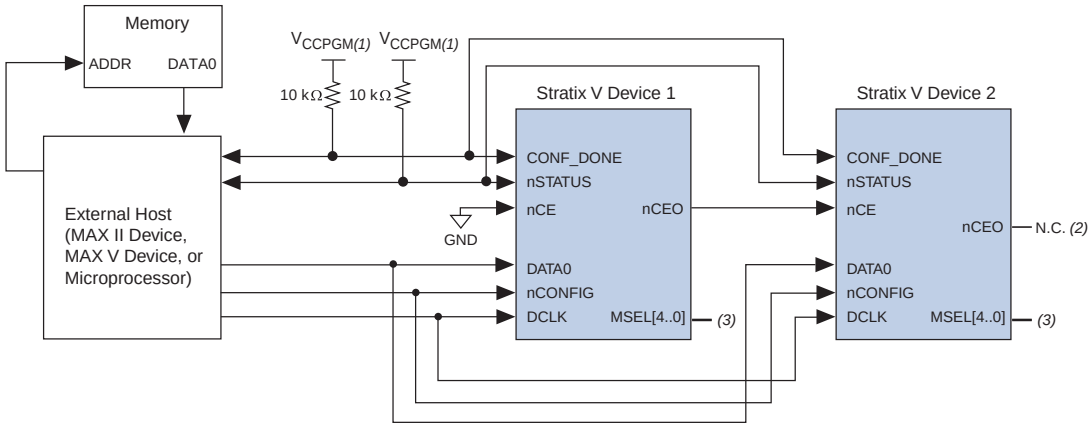
(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

Figure 9–16 shows the PS multi-device configuration using an external host when all devices in the chain receive different sets of configuration data (multiple **.sof**s).

Figure 9–16. PS Multi-device Configuration when Both Devices Receive Different Sets of Configuration Data



Notes to Figure 9–16:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin when it does not feed another device's nCE pin.

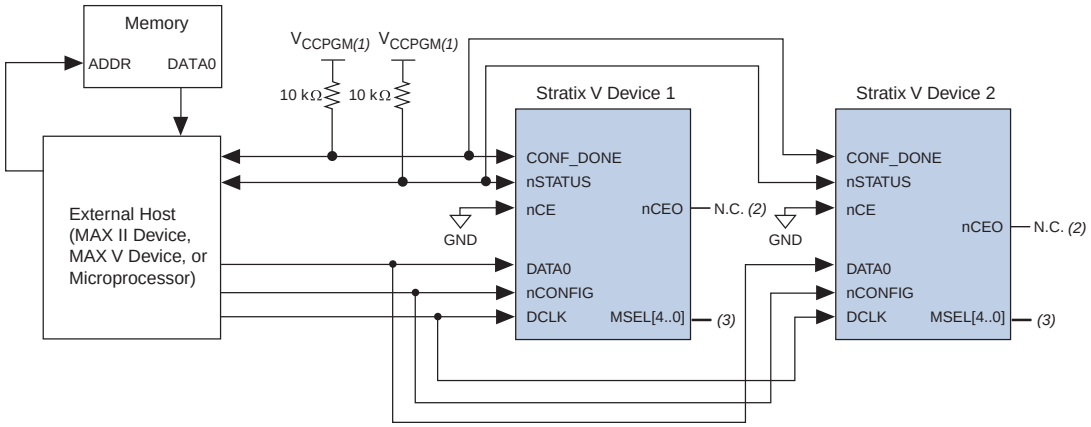
(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–16, after the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data to the second device is transparent to the external host.

Figure 9–17 shows the PS multi-device configuration when all devices receive the same set of configuration data (single **.sof**).

Figure 9–17. PS Multi-device Configuration When Both Devices Receive the Same Set of Configuration Data



Notes to Figure 9–17:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix V device. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device and the external host. Altera recommends powering up all the configuration system I/Os with V_{CCPGM}.

(2) You can leave the nCEO pin unconnected or use it as a user I/O pin.

(3) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL, refer to Table 9–4 on page 9–7.

(4) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

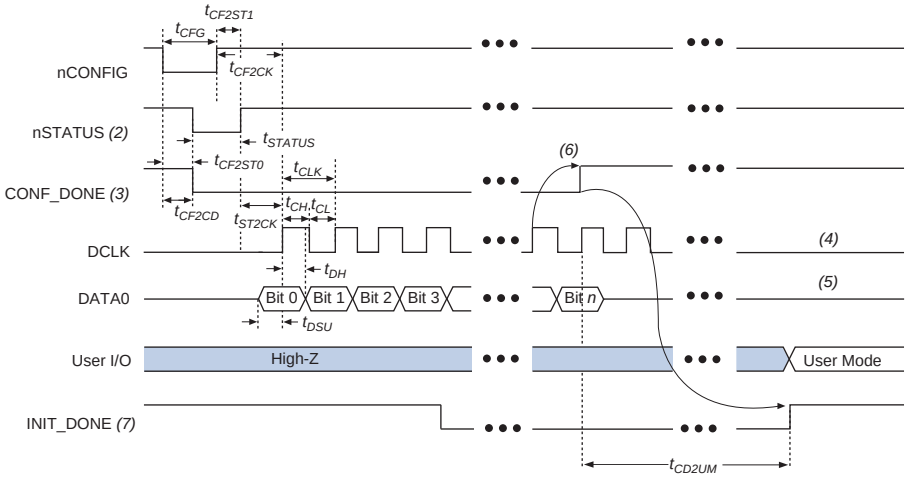
In Figure 9–17, because both nCE pins are tied to GND, both devices in the chain begin and complete the configuration and enter user mode at the same time.

To configure the PS multi-device with a single **.sof**, as shown in Figure 9–17, all Stratix V devices in the chain must be in the same package and density.

PS Configuration Timing

Figure 9–18 shows the timing waveform for a PS configuration when using a MAX II device, MAX V device, or microprocessor as an external host.

Figure 9–18. PS Configuration Timing Waveform *(1)*



Notes to Figure 9–18:

(1) The beginning of this waveform shows the device in user mode. In user mode, nCONFIG, nSTATUS, and CONF_DONE are at logic high levels. When nCONFIG is pulled low, a reconfiguration cycle begins.

(2) After power-up, the Stratix V device holds nSTATUS low for the time of the POR delay.

(3) After power-up, before and during configuration, CONF_DONE is low.

(4) Do not leave DCLK floating after configuration. You can drive it high or low, whichever is more convenient.

(5) DATA0 is available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings in the **Device and Pins Option**.

(6) To ensure a successful configuration, send the entire configuration data to the Stratix V device. CONF_DONE is released high after the Stratix V device receives all the configuration data successfully. After CONF_DONE goes high, send two additional falling edges on DCLK to begin initialization and enter user mode.

(7) After the option bit to enable the INIT_DONE pin is configured into the device, the INIT_DONE goes low.

Table 9–12 lists the PS configuration timing parameters for Stratix V devices.

Table 9–12. PS Timing Parameters for Stratix V Devices *(1)* **(Part 1 of 2)**

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	600	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	600	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	268	1,506 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	1,506 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	1,506	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	DATA[] setup time before rising edge on DCLK	5.5	—	ns
t_{DH}	DATA[] hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time	$0.45 \times 1/f_{MAX}$	—	s
t_{CL}	DCLK low time	$0.45 \times 1/f_{MAX}$	—	s

Table 9–12. PS Timing Parameters for Stratix V Devices (1) (Part 2 of 2)

Symbol	Parameter	Minimum	Maximum	Units
t_{CLK}	DCLK period	$1/f_{MAX}$	—	s
f_{MAX}	DCLK frequency	—	125	MHz
t_{CD2UM}	CONF_DONE high to user mode (4)	175	437	µs
t_{CD2CU}	CONF_DONE high to CLKUSR enabled	4 × maximum DCLK period	—	—
t_{CD2UMC}	CONF_DONE high to user mode with CLKUSR option on	t_{CD2CU} + (17,408 × CLKUSR period) (5)	—	—

Notes to Table 9–12:

(1) This information is preliminary.
(2) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.
(3) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.
(4) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for initializing the device.
(5) To enable the CLKUSR pin as the initialization clock source and to obtain the maximum frequency specification on these pins, refer to "Initialization" on page 9–5.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device for both uncompressed and compressed configuration data in the PS configuration scheme.

PS Configuration Using a Download Cable

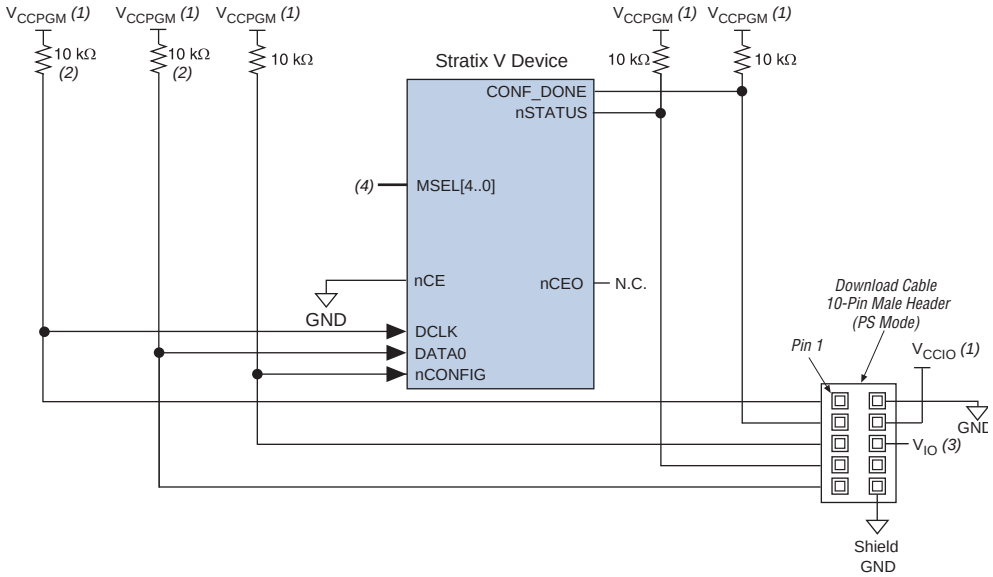
In this section, the generic term "download cable" includes the Altera USB-Blaster universal serial bus (USB) port download cable, ByteBlaster II parallel port download cable, and EthernetBlaster download cable.

In a PS configuration with a download cable, a PC acts as a host to transfer data from a storage device to the Stratix V device using the download cable. During configuration, the programming hardware or download cable places the configuration data one bit at a time on the device's DATA0 pin. The configuration data is clocked into the target device until CONF_DONE goes high.

If you turn on the CLKUSR option during PS configuration using a download cable and the Quartus II programmer, you do not have to provide a clock on the CLKUSR pin to initialize your device.

Figure 9–19 shows a PS configuration for Stratix V devices using an Altera download cable.

Figure 9–19. PS Configuration Using an Altera Download Cable



Notes to Figure 9–19:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCIO}) as the USB-Blaster, ByteBlaster II, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This ensures that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a MAX II device, MAX V device, or microprocessor, you do not need the pull-up resistors on DATA0 and DCLK.

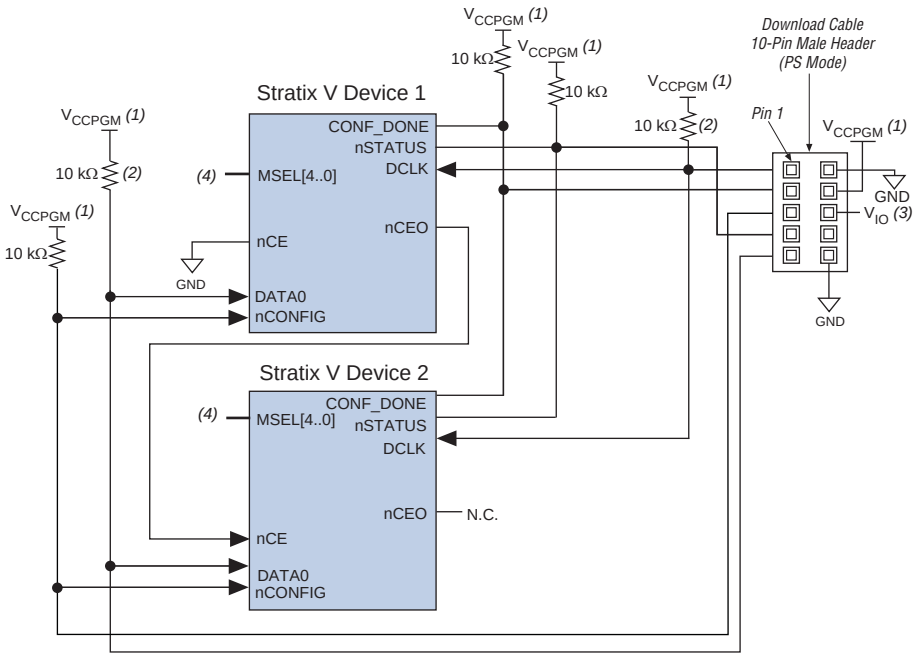
(3) In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when you use for AS programming; otherwise, it is a no connect.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL[4..0], refer to Table 9–4 on page 9–7.

Multi-Device PS Configuration Using Download Cable

You can use a download cable to configure multiple Stratix V devices as shown in Figure 9–20.

Figure 9–20. Multi-Device PS Configuration Using an Altera Download Cable



Notes to Figure 9–20:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCIO}) as the USB-Blaster, ByteBlaster II, or EthernetBlaster cable.

(2) You only need the pull-up resistors on DATA0 and DCLK if the download cable is the only configuration scheme used on your board. This ensures that DATA0 and DCLK are not left floating after configuration. For example, if you are also using a configuration device, you do not need the pull-up resistors on DATA0 and DCLK.

(3) In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when you use it for AS programming; otherwise, it is a no connect.

(4) The MSEL pin settings vary for different configuration voltage standards and POR delays. To connect MSEL[4..0], refer to Table 9–4 on page 9–7.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

In Figure 9–20, after the first device completes configuration, its nCEO pin drives low to activate the second device's nCE pin, which prompts the second device to begin configuration. The nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE pins are connected to every device in the chain. As all the CONF_DONE and nSTATUS pins are tied together, all devices initialize and enter user mode at the same time. If any device detects an error, configuration stops for the entire chain and you must reconfigure all the devices. For example, if the first device flags an error on nSTATUS, it resets the chain by pulling its nSTATUS pin low. This behavior is similar to a single device detecting an error.

JTAG Configuration

You can use the same JTAG interface specifically developed for boundary-scan test (BST) to shift the configuration data into the device. The Quartus II software automatically generates a **.sof** that you can use for JTAG configuration with a download cable in the Quartus II software programmer.

For more information about JTAG BST and the commands available using Stratix V devices, refer to the following documents:

- *JTAG Boundary-Scan Testing in Stratix V Devices* chapter
- Programming Support for Jam STAPL Language

Stratix V devices are designed such that JTAG instructions have precedence over any device configuration modes. JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration of Stratix V devices during a PS configuration, the PS configuration is terminated and the JTAG configuration begins. All user I/O pins are tri-stated during the JTAG configuration.

You cannot use the Stratix V decompression or design security features if you are configuring your Stratix V device using JTAG-based configuration.

For more information about `TDI`, `TDO`, `TMS`, and `TCK`, refer to "Device Configuration Pins" on page 9–40.

For instructions to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing in Stratix V Devices* chapter.

To configure a single device in a JTAG chain, the programming software places all the other devices in bypass mode. In bypass mode, devices pass programming data from the `TDI` pin to the `TDO` pin through a single bypass register without being affected internally. This scheme enables the programming software to program or verify the target device. Configuration data driven into the device appears on the `TDO` pin one clock cycle later. The Quartus II software verifies successful JTAG configuration after completion by checking the state of `CONF_DONE` through the JTAG port.

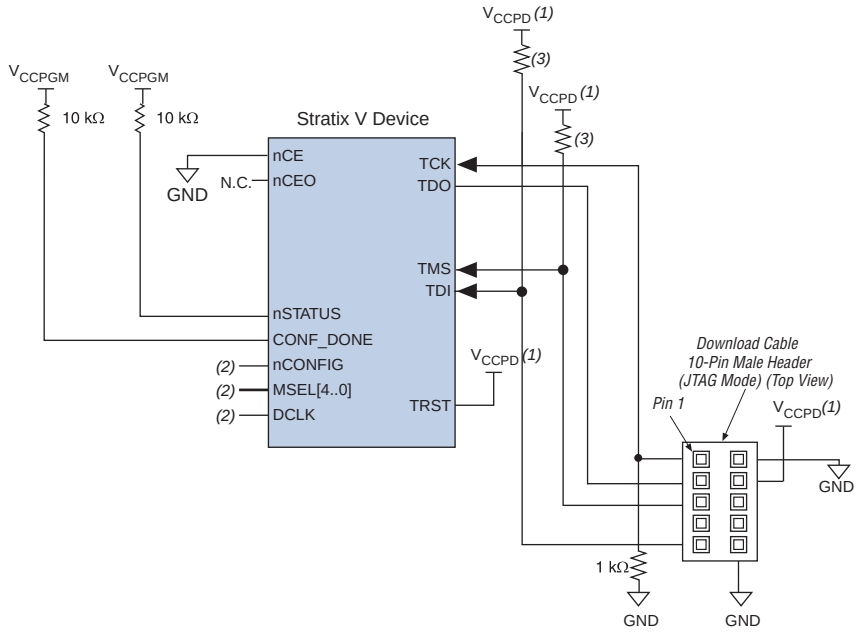
If `CONF_DONE` is not high, the Quartus II software indicates that configuration has failed. If `CONF_DONE` is high, the software indicates that configuration was successful. After the configuration data is transmitted serially using the JTAG `TDI` port, the `TCK` port is clocked an additional 1,222 cycles to perform device initialization.

The chip-wide reset (`DEV_CLRn`) and chip-wide output enable (`DEV_OE`) pins on Stratix V devices do not affect JTAG boundary-scan or programming operations.

You can generate a JAM File (**.jam**) or Jam-byte Code (**.jbc**) to be used with other third party programmer tools. Alternatively, you can use JRunner with **.rbf** to program your device.

Figure 9–21 shows the JTAG configuration of a single Stratix V device.

Figure 9–21. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 9–21:

(1) Connect the pull-up resistor V_{CCPD}. For more information, refer to V_{CCPD} requirement in "V_{CCPD} Pin" on page 9–3.

(2) If you only use JTAG configuration, connect `nCONFIG` to V_{CCPGM} and `MSEL[4..0]` to GND. Pull `DCLK` either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, connect `MSEL[4..0]`, `nCONFIG` and `DCLK` based on the selected configuration scheme.

(3) The resistor value can vary from 1 kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

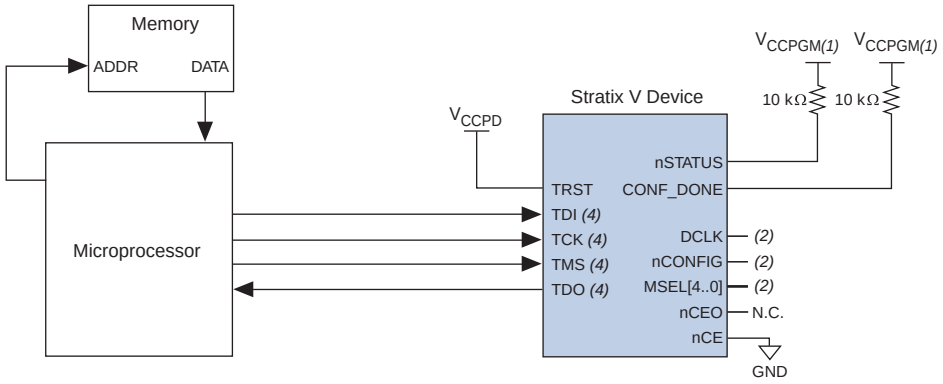
(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

Alternatively, you can use a microprocessor to program the device through the JTAG interface. You can use JRunner as your software driver.

For more information about JRunner, refer to *AN 414: The JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration*.

Figure 9–22 shows a JTAG configuration of a Stratix V device using a microprocessor.

Figure 9–22. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 9–22:

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for all Stratix V devices in the chain. V_{CCPGM} must be high enough to meet the V_{IH} specification of the I/O on the device.

(2) If you only use the JTAG configuration, connect nCONFIG to V_{CCPGM} and MSEL[4..0] to GND. Pull DCLK either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, set the MSEL[4..0] and tie nCONFIG and DCLK based on the selected configuration scheme.

(3) Connect nCE to GND or drive it low for successful JTAG configuration.

(4) The microprocessor must use the same I/O standard as V_{CCPD} to drive the JTAG pins.

(5) The nCEO pin is disabled by default in the Quartus II software. For the multi-device configuration chain, you must enable the nCEO pin in the Quartus II software. Otherwise, device configuration could fail.

CONFIG_IO Instruction

The CONFIG_IO instruction allows you to configure I/O buffers using the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix V device or waiting for a configuration device to complete configuration. After configuration is interrupted and JTAG testing is complete, you must reconfigure the part using the JTAG interface or if you support FPP, PS, or AS on your board, reconfigure the device by externally pulsing nCONFIG low. Alternatively, you can pulse nCONFIG low through the same JTAG interface using the PULSE_NCONFIG JTAG instruction.

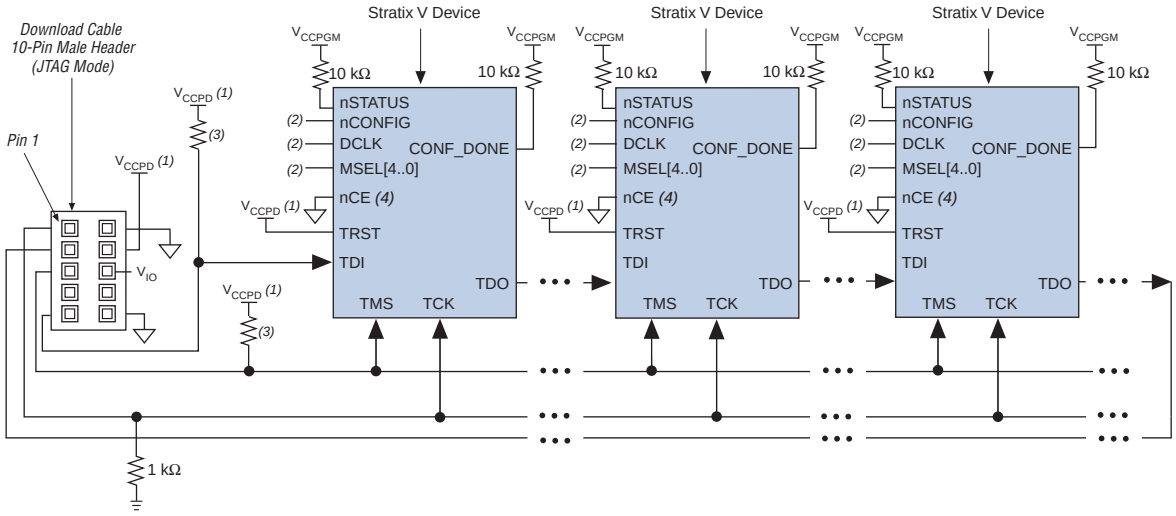
All JTAG instructions (except BYPASS, IDCODE, and SAMPLE) can be issued by first interrupting the configuration and reprogramming the I/O pins using the CONFIG_IO instruction.

Multi-Device JTAG Configuration

When programming a JTAG device chain, one JTAG-compatible header is connected to several devices. The number of devices in the JTAG chain is limited only by the drive capability of the download cable. When four or more devices are connected in a JTAG chain, Altera recommends buffering the TCK, TDI, and TMS pins with an on-board buffer. You can place other Altera devices that have JTAG support in the same JTAG chain for device programming.

JTAG-chain device programming is ideal when the system contains multiple devices or when testing your system using JTAG BST circuitry. Figure 9–23 shows a multi-device JTAG configuration.

Figure 9–23. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 9–23:

(1) Connect the pull-up resistor V_{CCPD}. For more information, refer to V_{CCPD} requirement in "V_{CCPD} Pin" on page 9–3.

(2) If you only use JTAG configuration, connect `nCONFIG` to V_{CCPGM} and `MSEL[4..0]` to GND. Pull `DCLK` either high or low, whichever is convenient on your board. If you are using JTAG in conjunction with another configuration scheme, connect the `MSEL[4..0]`, `nCONFIG`, and `DCLK` based on the selected configuration scheme.

(3) The resistor value can vary from 1kΩ to 10 kΩ. Perform signal integrity analysis to select the resistor value for your setup.

(4) You must connect `nCE` to GND or drive it low for successful JTAG configuration.

If you want to use the JTAG multi-device configuration in conjunction with other schemes, such as a FPP, PS, or AS, tie `CONF_DONE`, `nSTATUS`, and `nCONFIG` together as recommended in the FPP, PS, or AS multi-device configuration schemes. Ensure that the JTAG chain is the same order as the multi-device FPP, PS, or AS configuration chain.

If you only use JTAG configuration, Altera recommends connecting the circuitry as shown in Figure 9–22, where each of the `CONF_DONE` and `nSTATUS` signals are isolated to enable each device to enter user mode individually.

For more information about combining the JTAG configuration with other configuration schemes, refer to the *Combining Different Configuration Schemes* chapter in volume 2 of the *Configuration Handbook*.

For more information about JTAG and Jam STAPL in embedded environments, refer to *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the Jam player, visit the Altera website.

For more information about how to use the USB-Blaster, ByteBlaster II, or EthernetBlaster cables, refer to the following user guides:

- *USB-Blaster Download Cable User Guide*
- *ByteBlaster II Download Cable User Guide*
- *Ethernet Blaster Communications Cable User Guide*

Device Configuration Pins

Table 9–13 and Table 9–14 list the connections and functionality of all the configuration-related pins on the Stratix V devices. Table 9–13 lists the Stratix V configuration pins and their power supply.

Table 9–13. Configuration Pin Summary for Stratix V Devices (Part 1 of 2)

Description	Input/Output	User Mode	Powered By	Configuration Scheme
TDI	Input	—	V_{CCPD}	JTAG
TMS	Input	—	V_{CCPD}	JTAG
TCK	Input	—	V_{CCPD}	JTAG
TRST	Input	—	V_{CCPD}	JTAG
TDO	Output	—	V_{CCPD}	JTAG
CLKUSR	Input	I/O (1)	V_{CCPGM}/V_{CCIO} (4)	All schemes
CRC_ERROR	Output	I/O (1)	Pull-up	Optional, all schemes
CONF_DONE	Bidirectional	—	V_{CCPGM}/Pull-up	All schemes
DATA0	Bidirectional	I/O (2)	V_{CCPGM}/V_{CCIO} (4)	FPP, PS
DATA[31..1]	Bidirectional	I/O (2)	V_{CCPGM}/V_{CCIO} (4)	FPP
DCLK	Input	—	V_{CCPGM}	FPP, PS
	Output	—	V_{CCPGM}	AS
DEV_OE	Input	I/O (1)	V_{CCPGM}/V_{CCIO} (4)	Optional, all schemes
DEV_CLRn	Input	I/O (1)	V_{CCPGM}/V_{CCIO} (4)	Optional, all schemes
INIT_DONE	Output	I/O (1)	Pull-up	Optional, all schemes
MSEL[4..0]	Input	—	V_{CCPGM}	All schemes
nSTATUS	Bidirectional	—	V_{CCPGM}	All schemes
nCE	Input	—	V_{CCPGM}	All schemes
nCEO	Output	I/O (3)	Pull-up	All schemes
nCONFIG	Input	—	V_{CCPGM}	All schemes
nCSO	Output	—	V_{CCPGM}	AS
nIO_PULLUP	Input	—	V_{CC} (5)	All schemes
AS_DATA0/ASDO	Bidirectional	—	V_{CCPGM}	AS

Table 9–13. Configuration Pin Summary for Stratix V Devices (Part 2 of 2)

Description	Input/Output	User Mode	Powered By	Configuration Scheme
AS_DATA[3..1]	Bidirectional	—	V_{CCPGM}	AS

Notes to Table 9–13:

(1) This is a dual-purpose pin. This pin is available as an I/O if the associated option that enables this pin is turned off from the **Configuration** panel in the **Device and Pins Option** settings. For example, the DEV_OE is available as a user I/O if the **Enable device-wide output enable** option is turned off.

(2) This is a dual-purpose pin. The state of this pin in user mode depends on the **Dual-purpose Pins** settings in the **Device and Pins Option**.

(3) This pin is available as an I/O if this pin is not feeding the next device's nCE in a multi-device configuration. To use this pin to feed the next device's nCE in a multi-device chain, turn on **Enable INIT_DONE output** option under **Device and Pins Option**, **General** panel in the Quartus II Software.

(4) This pin is powered up by V_{CCPGM} during configuration. It is powered up by V_{CCIO} of the bank in which the pin resides if it is used as a regular I/O in user mode.

(5) Although nIO_PULLUP is powered up by V_{CC}, Altera recommends connecting this pin to V_{CCPGM} or GND directly without using a pull-up or pull-down resistor.

Table 9–14 lists the configuration pin descriptions.

Table 9–14. Configuration Pins Description (Part 1 of 3)

Pin Name	Description
TDI (1)	Dedicated test data input. Serial input pin for instructions as well as test and programming data. Data is shifted on the rising edge of TCK. This pin has an internal 25-kΩ pull-up that is always active.
TMS (1)	Dedicated test mode select. Input pin that provides the control signal to determine the transitions of the TAP controller state machine. TMS is evaluated on the rising edge of TCK. Therefore, you must set up TMS before the rising edge of TCK. Transitions in the state machine occur on the falling edge of TCK after the signal is applied to TMS. This pin has an internal 25-kΩ pull-up that is always active.
TCK (1)	Dedicated test clock input. Clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. It is expected that the clock input waveform have a nominal 50% duty cycle. This pin has an internal 25-kΩ pull-down that is always active.
TRST (1)	Dedicated test reset input. Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to the IEEE Std. 1149.1 standard. Connecting this pin low disables the JTAG circuitry. This pin has an internal 25-kΩ pull-up that is always active.
TDO (1)	Dedicated test data output. Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. This pin is tri-stated if the data is not being shifted out of the device.
CLKUSR	Optional user-supplied clock input. It synchronizes the initialization of one or more devices. Enable this pin by turning on the **Enable user-supplied start-up clock (CLKUSR)** option under **Device and Pins Option**, **Configuration** panel in the Quartus II software.

Table 9–14. Configuration Pins Description (Part 2 of 3)

Pin Name	Description
CRC_ERROR	Optional output pin. Signals that the device has detected a cyclical redundancy check (CRC) error during user mode operation. this pin is an open-drain output pin by default and requires a 10 kΩ pull-up resistor. To use this pin as regular output, turn-off the **Enable Open-drain on CRC_ERROR pin** in **Device and Pins Option**, **Error Detection CRC** panel in the Quartus II software. The target device drives this pin low if there is no CRC error in user mode operation. As an open-drain output, if a CRC error occurs, the device releases the pin which is then pulled high by the external pull-up resistor. Enable this pin by turning on **Enable CRC error detection on CRC_ERROR pin** option in the Quartus II software. For more information about the CRC_ERROR pin, refer to *SEU Mitigation in Stratix V Devices* chapter.
CONF_DONE	Dedicated open-drain bidirectional pin. The target device drives the CONF_DONE pin low before and during configuration. After all the configuration data is received without error and the initialization cycle starts, the target device releases the CONF_DONE pin, which is then pulled high by the external pull-up resistor. The target device then reads the CONF_DONE pin status to ensure that the CONF_DONE is at logic high. After it is sensed high, the target device initializes and enters user mode. Driving CONF_DONE low after initialization completes does not affect the configured device.
DATA0 *(3)*	Dual-purpose data input pin. The data received on DATA0 is synchronized to DCLK. After configuration completes, this pin is available as a user I/O pin.
DATA[31..1] *(3)*	Dual-purpose data input pins. If you are using FPP ×16 or FPP ×32, only a subset of these pins are required for configuration. The pins that are not required for configuration, you can use them as regular I/Os. During configuration, byte-wide, or word-wide data is received on these pins. The data received on DATA[31..1] are synchronized to the DCLK.
DCLK	Dedicated bidirectional clock pin. In the PS and FPP configurations, DCLK is the clock input used to clock data from an external source into the target device. Data is latched into the device on the rising edge of DCLK. After configuration completes, drive DCLK high or low, whichever is more convenient. In the AS mode, DCLK is an output clock to clock the EPCS or EPCQ devices. Data is latched into the device on the falling edge of the DCLK. After AS configuration completes, this pin is tri-stated with a weak pull-up resistor. Toggling this pin after configuration does not affect the configured device.
DEV_OE *(2)*	Optional input pin that allows you to override all tri-states on the device. When this pin is driven low, all the I/O pins are tri-stated. When this pin is driven high, all the I/O pins behave as programmed. Enable this pin by turning on the **Enable device-wide output enable (DEV_OE)** option in the Quartus II software.
DEV_CLRn *(2)*	Optional input pin that allows you to override all clears on all the device registers. When this pin is driven low, all the registers are cleared. When this pin is driven high, all the registers behave as programmed. This pin is enabled by turning on the **Enable device-wide reset (DEV_CLRn)** option in the Quartus II software.
INIT_DONE *(2)*	Optional output pin. Signals when the device has initialized and is in user mode. During the reset stage, after the device exits POR, and during the beginning of the configuration, the INIT_DONE pin is tri-stated and pulled high due to an external pull-up resistor. After the option bit to enable INIT_DONE is programmed into the device (during the first frame of configuration data), the INIT_DONE pin goes low. When initialization completes, the INIT_DONE pin is released and pulled high and the device enters user mode. Thus, the monitoring circuitry must be able to detect a low-to-high transition. Enable this pin by turning on the **Enable INIT_DONE output** option in the Quartus II software.

Table 9–14. Configuration Pins Description (Part 3 of 3)

Pin Name	Description
MSEL[4..0]	Dedicated input pins. Five-bit configuration input that sets the Stratix V device configuration scheme. For the appropriate connections, refer to Table 9–4 on page 9–7. The MSEL[4..0] pins have internal 25-kΩ pull-down resistors that are always active.
nSTATUS	Dedicated open-drain bidirectional pin. The device drives nSTATUS low immediately after power-up and releases it after the device exits POR. During user mode and regular configuration, this pin is pulled high by an external 10-kΩ resistor. During configuration, the device drives this pin low to indicate an error during configuration. If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. This mechanism is used during multi-device configuration setup. If one of the devices in the chain has an error and pulls its nSTATUS pin low, it resets the entire chain. Driving nSTATUS low after configuration and initialization completes does not affect the configured device.
nCE	Dedicated active-low chip enable input pin. Driving this pin low allows configuration. Drives the nCE pin low during configuration, initialization, and user mode for all single-device configurations. For a multi-device configuration, connect the nCE pin to GND or to the nCEO of the previous device in the chain based on the recommendation in the respective configuration setup diagram.
nCEO *(3)*	Dual-purpose open-drain output pin. This pin drives low when device configuration completes. To use this pin to feed the next device's nCE pin in a multi-device chain, turn on **Enable INIT_DONE output** option under **Device and Pins Option**, **General** panel in the Quartus II Software. In a single-device configuration, this pin can be used as a regular I/O. In multi-device configuration, if this pin is not feeding the nCE of the next device, you can use it as a regular I/O.
nCONFIG	Dedicated input pin. A low pulse on this pin during configuration and user mode causes the device to enter a reset state and tri-states all the I/O pins. A low-to-high logic starts a reconfiguration. During JTAG programming, the nCONFIG status is ignored.
nCSO	Dedicated output pin. Drives the control signal from the Stratix V device to the EPCS and EPCQ devices in AS mode. After AS configuration completes, these pins are tri-stated with a weak pull-up resistor.
nIO_PULLUP	Dedicated input pin. This input pin enables or disables the internal pull-up resistors on the user I/O pins and dual purpose I/O pins (DATA[31..0], CLKUSR, INIT_DONE, DEV_OE, and DEV_CLRN). A logic high turns off the weak internal pull-up resistors, while a logic low turns them on. This pin has an internal 25-kΩ pull-down resistor that is always active.
AS_DATA0/ASDO	Dedicated bidirectional data pin. In AS ×1 and AS ×4 configurations, ASDO is used to send the operation command and addresses to the EPCS or EPCQ devices. During an AS ×4 configuration, the data is received on an AS_DATA0 and is synchronized to DCLK. After AS configuration completes, this pin is tri-stated with a weak pull-up resistor.
AS_DATA[3..1]	Dedicated bidirectional data pins. During an AS configuration, the data is received on these pins and is synchronized to DCLK. After AS configuration completes, these pins are tri-stated with a weak pull-up resistor.

Notes to Table 9–14:

(1) If the JTAG interface is not required on the board, you can disable the JTAG circuitry by connecting this pin to logic high. For instructions to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing* chapter.

(2) This is a dual-purpose pin. This pin is available as an I/O if the associated option that enables this pin is turned off from the **Configuration** panel in the **Device and Pins Option** settings. For example, the DEV_OE is available as a user I/O if the **Enable device-wide output enable** option is turned off.

(3) This is a dual-purpose pin. The state of this pin in the user mode depends on **Dual-purpose Pins** settings in the **Device and Pins Option** settings.

Configuration Data Decompression

Stratix V devices support configuration data decompression, which saves configuration memory space and may shorten the configuration time. This feature allows you to store compressed configuration data in the configuration or other memory devices and transmit this compressed data to the Stratix V devices. During configuration, the Stratix V device decompresses the data in real time and programs its SRAM cells. The data decompression is done on-the-fly during configuration and does not require an additional processing time.

Preliminary data indicates that compression typically reduces the configuration data size by 30 to 55% based on the designs used. This reduces the storage requirement capacity for the flash memory. The decompression feature is supported in all configuration schemes except JTAG.

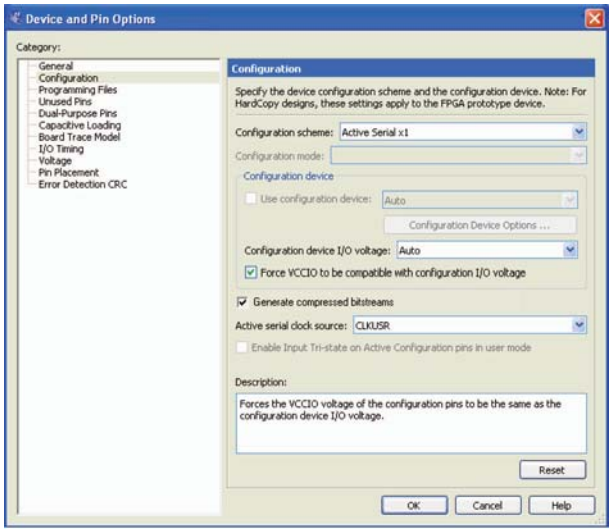
In FPP, enabling the decompression feature requires a different DCLK-to-DATA[] ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

There are two ways to enable compression for Stratix V data—before design compilation (in the Compiler Settings menu) and after design compilation (in the Convert Programming Files window).

To enable compression in the project's Compiler Settings menu, follow these steps:

1. On the Assignments menu, click **Device** to bring up the **Settings** dialog box.
2. After selecting your Stratix V device, open the **Device and Pin Options** dialog box.
3. In the **Configuration settings** panel, turn on the **Generate compressed bitstreams** option (refer to Figure 9–24).

Figure 9–24. Enabling Compression Bitstreams in Compiler Settings for Stratix V



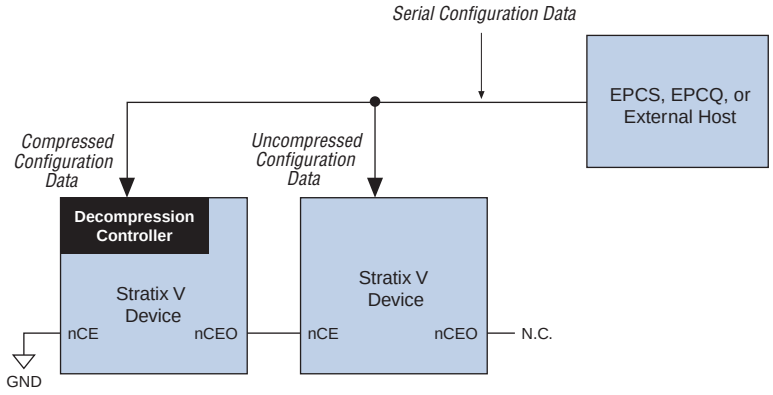
You can also enable compression when creating programming files from the **Convert Programming Files** window. To do this, follow these steps:

1. On the File menu, click **Convert Programming Files**.
2. Select the programming file type (**.pof**, **.sram**, **.hex**, **.hexout**, **.rbf**, or **.ttf**).
3. For POF output files, select a configuration device.
4. In the **Input files to convert** box, select **SOF Data**.
5. Select **Add File** and add a Stratix V device **.sof**.
6. Select the name of the file you added to the **SOF Data** area and click **Properties**.
7. Check the **Compression** check box.

If you are using a serial configuration scheme, AS ×1 or PS, for multi-device configuration, you can selectively enable the compression feature for each device in the chain. Figure 9–25 shows a chain of two Stratix V devices. The first Stratix V device has compression enabled and therefore receives compressed data from the external host. The second Stratix V device has the compression feature disabled and receives uncompressed data.

For FPP configuration schemes, a combination of compressed and uncompressed configuration in the same multi-device chain is not allowed due to the difference of the DCLK-to-DATA[] ratio.

Figure 9–25. Compressed and Uncompressed Serial Configuration Data in the Same Configuration File *(1)*



Note to Figure 9–25:

(1) The configuration for this setup can be generated from the Convert Programming Files menu in the Quartus II software.

Remote System Upgrades

This section describes the functionality and implementation of the dedicated remote system upgrade circuitry. It also defines several concepts related to remote system upgrades, including factory configuration, application configuration, remote update mode, and user watchdog timer. Additionally, this section provides design guidelines for implementing remote system upgrades with the supported configuration schemes.

System designers sometimes face challenges such as shortened design cycles, evolving standards, and system deployments in remote locations. Stratix V devices help overcome these challenges with their inherent reprogrammability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduce time-to-market, extend product life, and help to avoid system downtime.

Stratix V devices feature dedicated remote system upgrade circuitry. A soft logic (either the Nios® II embedded processor or user logic) implemented in a Stratix V device can download a new configuration image from a remote location, store it in configuration memory, and direct the dedicated remote system upgrade circuitry to start a reconfiguration cycle. The dedicated circuitry performs error detection during and after the configuration process, recovers from any error condition by reverting back to a safe configuration image, and provides error status information.

Remote system upgrades are supported in AS configuration schemes with EPCS and EPCQ devices. You can also implement remote system upgrades in conjunction with advanced Stratix V features such as real-time decompression of configuration data and design security using the advanced design security standard (AES) for secure and efficient field upgrades. The largest EPCS and EPCQ device currently supports 128 Mbits and 256 Mbits configuration data respectively.

Remote system upgrades are supported only in single-device configurations.

The remote system upgrade process in Stratix V devices involves the following steps:

1. A Nios II processor (or user logic) implemented in the Stratix V device logic array receives new configuration data from a remote location. The connection to the remote source uses a communication protocol such as TCP/IP, PCI, user datagram protocol (UDP), UART, or a proprietary interface.
2. The Nios II processor (or user logic) stores this new configuration data in non-volatile configuration memory.
3. The Nios II processor (or user logic) starts a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any errors that might occur during or after the reconfiguration cycle and provides error status information to the user design.

Figure 9–26 shows these remote system upgrade steps.

Figure 9–26. Functional Diagram of the Stratix V Remote System Upgrade Process

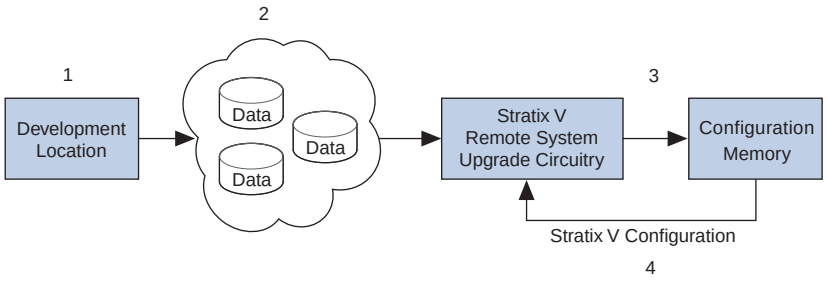
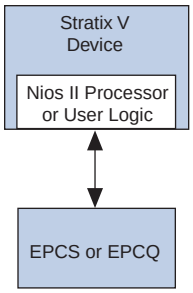


Figure 9–27 shows a block diagram for implementing a remote system upgrade with the Stratix V AS configuration scheme.

Figure 9–27. Remote System Upgrade Block Diagram for Stratix V AS Configuration Scheme *(1)*



Note to Figure 9–27:

(1) You must set the mode select pins (`MSEL[4..0]`) to **AS mode** to use remote system upgrade in your system. The `MSEL` pin settings vary for different POR delays. To connect `MSEL[4..0]`, refer to Table 9–4 on page 9–7.

Configuration Image Types

When performing a remote system upgrade, Stratix V device configuration data are classified as factory configuration images or application configuration images. An image, also referred to as a configuration image, is a design loaded into the Stratix V device that performs certain user-defined functions.

The factory image is a user-defined fall-back, or safe configuration, and is responsible for initiating the reconfiguration to a new image with the dedicated circuitry. Application images implement user-defined functionality in the target Stratix V device. You may also include the default application image functionality in the factory image. Each Stratix V device in your system requires one factory image and one or more application images.

Remote Update Mode

Stratix V remote system upgrade circuitry only supports remote update mode. In remote update mode, Stratix V devices load the factory configuration image after power up. The user-defined factory configuration determines which application configuration is to be loaded and triggers a reconfiguration cycle.

When a Stratix V device is first powered up in remote update mode, it loads the factory configuration located at the start address of `PGM[23..0]` = 24'h000000 in the EPCS and EPCQ devices. You must store your factory configuration image at this start address.

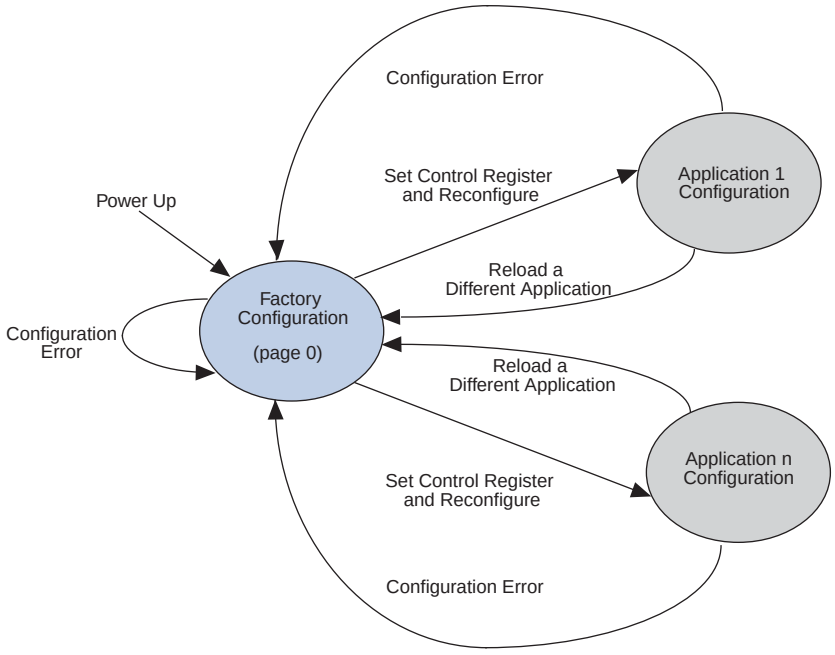
The application image start address can be at any EPCS or EPCQ sector boundary. Altera recommends using different sectors in the EPCS device for two images.

The factory image is user-designed and contains soft logic to perform the following:

- Process any errors based on status information from the dedicated remote system upgrade circuitry
- Communicate with the remote host and receive new application configurations and store this new configuration data in the local non-volatile memory device
- Determine which application configuration is to be loaded into the Stratix V device
- Enable or disable the user watchdog timer and load its time-out value
- Instruct the dedicated remote system upgrade circuitry to start a reconfiguration cycle

Figure 9–28 shows the transitions between the factory and application configurations in remote update mode.

Figure 9–28. Transitions Between Configurations in Remote Update Mode



After power up or a configuration error, the factory configuration image is loaded automatically. The system then decides to switch to the application configuration image or to stay in the factory configuration image. After the system decides to switch to an application configuration image, a reconfiguration is initiated through the remote system upgrade circuitry. In the application configuration image, the system may revert back to factory configuration image after the following reconfiguration trigger conditions are met:

- nSTATUS driven low externally
- Configuration CRC error
- User watchdog timer time-out
- Core nCONFIG signal assertion
- External nCONFIG signal assertion

After the factory configuration image is re-loaded, the user-designed factory configuration can read the remote system upgrade status register to determine the reason for the reconfiguration. The factory configuration then takes the appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

Whenever the application configuration image is successfully loaded, the soft logic (Nios II processor or state machine and the remote communication interface) determine when the remote system update is arriving. When this occurs, the soft logic receives the incoming data, writes it to the configuration memory device, and triggers the device to load the factory configuration. The factory configuration reads the remote system upgrade status register and control register, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and initiates system reconfiguration.

Remote System Upgrade Using EPCQ 256

When you are using EPCQ 256, ensure that the application image address granularity is 32'h00000100. The **.rbf** size for your application image is 76,500 bytes longer than the numbers listed in Table 9–5 on page 9–8. You need to take this extra space requirement into consideration when you try to fit multiple application images in the EPCQ 256 device.

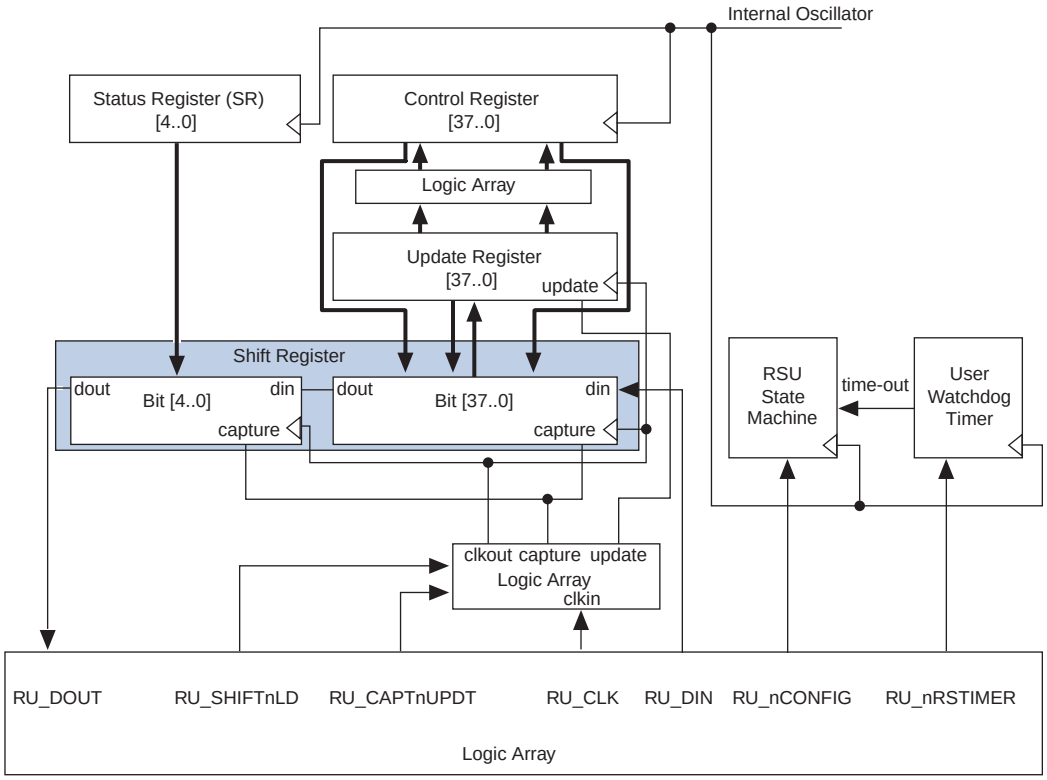
If you are not using Quartus II software or SRunner software for EPCQ 256 programming, put your EPCQ 256 device into four-byte addressing mode before you program and configure your device.

Dedicated Remote System Upgrade Circuitry

This section describes the implementation of the Stratix V dedicated remote system upgrades circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces with the user-defined factory and application configurations implemented in the Stratix V device logic array to provide the complete remote configuration solution. The remote system upgrade circuitry contains the remote system upgrade registers, a watchdog timer, and a state machine that controls those components.

Figure 9–29 shows the remote system upgrade circuitry.

Figure 9–29. Remote System Upgrade Circuitry *(1)*



Note to Figure 9–29:

(1) If you are using the ALTREMOTE_UPDATE megafunction, the RU_DOUT, RU_SHIFTnLD, RU_CAPTnUPDT, RU_CLK, RU_DIN, RU_nCONFIG, and RU_nRSTIMER signals are internally controlled by the megafunction to perform all the related remote system upgrade operations.

Table 9–15 lists the timing parameter specifications for the remote system upgrade circuitry.

Table 9–15. Remote System Upgrade Circuitry Timing Specifications

Parameter	Minimum	Maximum	Unit
$f_{MAX\_RU\_CLK}$ *(1)*	—	40	MHz
$t_{RU\_nCONFIG}$ *(2)*	250	—	ns
$t_{RU\_nRSTIMER}$ *(3)*	250	—	ns

Notes to Table 9–15:

(1) This clock is user-supplied to the remote system upgrade circuitry. If you are using the ALTREMOTE_UPDATE megafunction, the clock user-supplied to the ALTREMOTE_UPDATE megafunction must meet this specification.

(2) This is equivalent to strobing the reconfiguration input of the ALTREMOTE_UPDATE megafunction high for the minimum timing specification. For more information, refer to "Remote System Upgrade State Machine" on page 9–53.

(3) This is equivalent to strobing the reset_timer input of the ALTREMOTE_UPDATE megafunction high for the minimum timing specification. For more information, refer to "User Watchdog Timer" on page 9–53.

Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that store the page addresses, watchdog timer settings, and status information. Table 9–16 lists these registers.

Table 9–16. Remote System Upgrade Registers

Register	Description
Shift register	This register is accessible by the logic array and allows the update, status, and control registers to be written and sampled by user logic.
Control register	This register contains the current page address, user watchdog timer settings, and one bit specifying whether the current configuration is a factory configuration or an application configuration. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is initiated, the contents of the update register are written into the control register.
Update register	This register contains data similar to that in the control register. However, it can only be updated by the factory configuration by shifting data into the shift register and issuing an update operation. When a reconfiguration cycle is triggered by the factory configuration, the control register is updated with the contents of the update register. During a capture in a factory configuration, this register is read into the shift register.
Status register	This register is written to by the remote system upgrade circuitry on every reconfiguration to record the cause of the reconfiguration. This information is used by the factory configuration to determine the appropriate action following a reconfiguration. During a capture cycle, this register is read into the shift register.

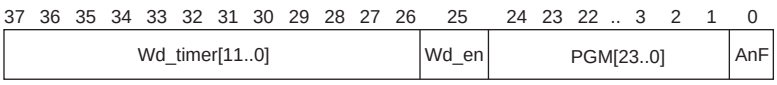
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (RU_CLK).

Control Register

The control register stores the application configuration page address and user watchdog timer settings. The control register functionality depends on the remote system upgrade mode selection. A factory configuration in remote update mode has write access to this register.

Figure 9–30 shows the control register bit positions. Table 9–17 lists the control register bits. In the figure, the numbers show the bit position of a setting in a register. For example, bit number 25 is the enable bit for the watchdog timer.

Figure 9–30. Remote System Upgrade Control Register



The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix V device is the factory configuration or an application configuration. This bit is set low by the remote system upgrade circuitry when an error condition causes a fall-back to the factory configuration. When the AnF bit is high, the control register restricts the access to only read operations and enables the watchdog timer. The factory configuration design must set this bit high (1'b1) when updating the contents of the update register with the application page address and watchdog timer settings.

Table 9–17 lists the remote system upgrade control register contents.

Table 9–17. Remote System Upgrade Control Register Contents

Control Register Bit	Value (2)	Definition
AnF (1)	1'b0	Application not factory
PGM[23..0]	24'b0×000000	AS configuration start address (StAdd[23..0])
Wd_en	1'b0	User watchdog timer enable bit
Wd_timer[11..0]	12'b000000000000	User watchdog time-out value (most significant 12 bits of 29-bit count value: {Wd_timer[11..0], 17'b0})

Notes to Table 9–17:

(1) Factory configuration designs must set the AnF bit to **1'b1** before triggering the reconfiguration to application configuration image.

(2) This is the default value of the control register bit after the device exits POR and during reconfiguration back to the factory configuration image after reconfiguration trigger conditions.

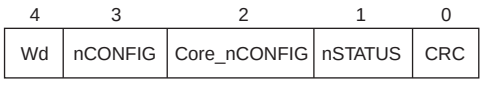
Status Register

The status register specifies the reconfiguration trigger condition. Figure 9–31 shows the status register content. The following list defines each bit:

- Bit 0—CRC error during application configuration
- Bit 1—nSTATUS assertion by an external device due to an error
- Bit 2—Stratix V device logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- Bit 3—external configuration reset (nCONFIG) assertion
- Bit 4—user watchdog timer time-out

Figure 9–31 shows the contents of the status register. The numbers in the figure show the bit positions in a 5-bit register.

Figure 9–31. Remote System Upgrade Status Register (1)



Note to Figure 9–31:

(1) After the device exits POR and powers-up, the status register content is 5'b00000.

Remote System Upgrade State Machine

After power-up, the shift register, control register, and update registers are reset to the values listed in Table 9–16, also known as POR reset values before the factory configuration image is loaded. In the factory configuration image, the user logic writes the AnF bit, page address, and watchdog timer settings for the next application configuration image to the update register. When the logic array configuration reset (RU_nCONFIG) goes low, the remote system upgrade state machine updates the control register with the contents of the update register, and triggers a reconfiguration to the new application configuration image.

If there is an error during reconfiguration to the new application configuration image, the remote system upgrade state machine directs the system to re-load a factory configuration image. The control and update registers are reset to POR reset values and the status register is updated with the error information. For example, if there is a CRC error during application configuration image configuration, the status register is updated with 5'b00001.

If there is no error during reconfiguration and the application configuration image is successfully loaded, the system stays in the application configuration image until another reconfiguration trigger condition occurs. This can be a core nCONFIG assertion, external nCONFIG assertion, or the watchdog timer time-out error. If this happens, the control register and update registers are reset to POR reset values and the status register is updated with the error information. Consequently, the system proceeds to load the factory configuration image. Based on the status register content, the user logic in the factory configuration image then decides to stay in the factory configuration image or reload a new application reconfiguration image.

Read operations during factory configuration access the contents of the update register. This feature is used by the factory configuration image user logic to verify that the page address and watchdog timer settings are written correctly. Read operations in application configurations access the contents of the control register. This information is used by the user logic in the application configuration.

User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from stalling the device indefinitely. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Stratix V device. This feature is automatically disabled in the factory configuration image and enabled in the application configuration image. Functional errors must not exist in the factory configuration because they are stored and validated during production and must never be updated remotely.

The user watchdog timer feature is automatically enabled in the application configuration image. If you do not wish to use this feature, disable it during the factory configuration image operation before triggering the reconfiguration to the application configuration image.

The user watchdog timer is a counter that counts down from the initial value loaded into the remote system upgrade control register by the factory configuration. The counter is 29 bits wide and has a maximum count value of 2^{29}. When specifying the user watchdog timer value, specify only the most significant 12 bits. The granularity of the timer setting is 2^{17} cycles. The cycle time is based on the frequency of the 12.5-MHz internal oscillator. Table 9–18 lists the operating range of the 12.5-MHz internal oscillator.

Table 9–18. 12.5-MHz Internal Oscillator Specifications *(1)*

Minimum	Typical	Maximum	Units
5.3	7.9	12.5	MHz

Note to Table 9–18:

(1) These values are preliminary.

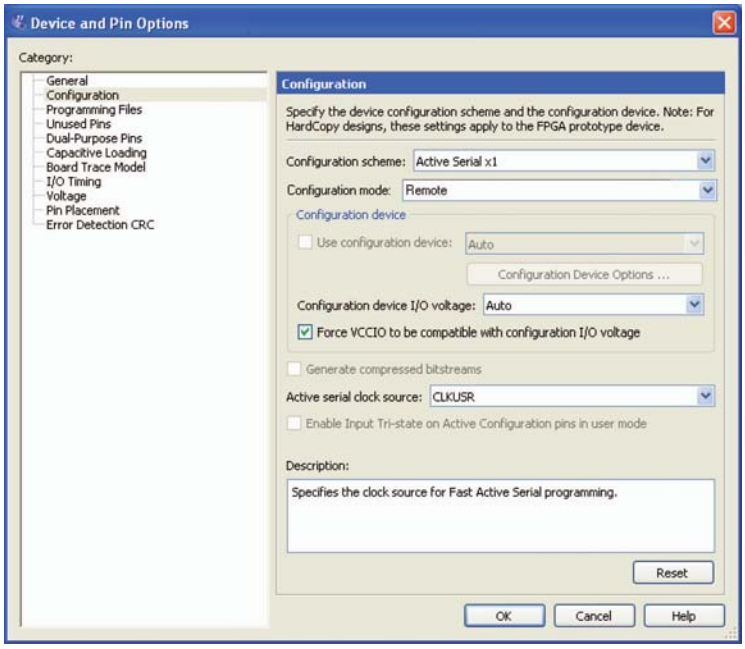
The user watchdog timer begins counting after the application configuration enters device user mode. This timer must be periodically reset by the application configuration before the timer expires by asserting `RU_nRSTIMER`. If the application configuration does not reload the user watchdog timer before the count expires, a time-out signal is generated by the remote system upgrade dedicated circuitry. This causes the device to reload the factory configuration image and update the status register to reflect the watchdog timer time-out error.

Enabling the Remote System Update Feature

You can enable remote update for Stratix V devices in the Quartus II software before design compilation (in the Compiler Settings menu). In remote update mode, the **auto-restart configuration after error** option is always enabled. To enable remote update in the project's compiler settings in the Quartus II software, follow these steps:

1. On the Assignments menu, click **Device**. The **Settings** dialog box appears.
2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
3. Click the **Configuration** panel.
4. From the Configuration scheme list, select **Active Serial x1** (you can also use **Configuration Device**) (refer to Figure 9–32).
5. From the Configuration mode list, select **Remote** (refer to Figure 9–32).
6. Click **OK**.
7. In the **Settings** dialog box, click **OK**.

Figure 9–32. Enabling Remote Update for Stratix V Devices in the Compiler Settings Menu



ALTREMOTE_UPDATE Megafunction

The ALTREMOTE_UPDATE megafunction provides a memory-like interface to the remote system upgrade circuitry and handles the shift register read and write protocol in the Stratix V device logic. This implementation is suitable for designs that implement the factory configuration functions using a Nios II processor or user logic in the device. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

For more information about the ALTREMOTE_UPDATE megafunction, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide*.

Design Security

This section provides an overview of the design security features and their implementation in Stratix V devices using the advanced encryption standard (AES). It also describes the security modes available in Stratix V devices that allow you to use these new features in your designs.

As Stratix V devices continue to play roles in larger and more critical designs in competitive commercial and military environments, it is increasingly important to protect your designs from copying, reverse engineering, and tampering. Stratix V design security supports the following features:

- Enhanced built-in AES decryption block to support 256-key industry-standard design security algorithm (FIPS-197 Certified)
- Volatile and non-volatile key programming support
- Secure operation mode for both volatile and non-volatile key through tamper protection bit setting

- JTAG secure mode is enable through tamper-protection bit
- Supports board level testing
- Supports in-socket key programming for non-volatile key
- Available in all configuration schemes except JTAG
- Supports both remote system upgrades and decompression feature


You can use the design security feature with or without the remote system upgrades or decompression features.

The Stratix V design security feature provides the following security protection for your designs:

- Security against copying—the security key is securely stored in the Stratix V device and cannot be read out through any interface. In addition, as configuration file read-back is not supported in Stratix V devices, your design information cannot be copied.
- Security against reverse engineering—reverse engineering from an encrypted configuration file is very difficult and time consuming because the Stratix V configuration file formats are proprietary and the file contains millions of bits that require specific decryption. In addition, the Stratix V devices are manufactured on the most advanced 28-nm process technology, making this process very difficult.
- Security against tampering—this disables tamper attempts through the JTAG interface. You can enhance this security feature with the tamper protection bit setting. After the tamper protection bit is set, the Stratix V device can only accept configuration files encrypted with the same key. Additionally, programming through the JTAG interface is blocked. This prevents any attempts to tamper with the device from both the JTAG interface and the configuration interface.


When you use compression with the design security feature, the configuration file is first compressed and then encrypted using the Quartus II software. During configuration, the Stratix V device first decrypts and then decompresses the configuration file.


When you use design security with Stratix V devices in a FPP configuration scheme, it requires a different DCLK-to-DATA[] ratio. For more information, refer to "Fast Passive Parallel Configuration" on page 9–9.

JTAG Secure Mode

When you enable tamper-protection bit, Stratix V devices are in JTAG secure mode after power-up. During JTAG secure mode, many JTAG instructions are disabled. Stratix V devices only allow mandatory JTAG 1149.1 and 1149.6 instructions to be exercised. These instructions are SAMPLE/PRELOAD, BYPASS, EXTEST, and optional instructions such as IDCODE and SHIFT_EDERROR_REG.

To enable the access of other JTAG instructions such as USERCODE, HIGHZ, CLAMP, PULSE_NCONFIG, and CONFIG_IO, you must issue UNLOCK instruction to deactivate the JTAG secure mode. You can issue LOCK instruction to put the device back into JTAG secure mode. Both the LOCK and UNLOCK instructions can only be issued through JTAG core access during user mode.

For more information about JTAG binary instruction code related to the LOCK and UNLOCK instructions, refer to the *JTAG Boundary-Scan Testing in Stratix V Devices* chapter.

Security Key Types

Stratix V devices offer two types of keys—volatile and non-volatile. Table 9–19 lists the differences between the volatile key and non-volatile key.

Table 9–19. Security Key Types

Key Types	Key Programmability	Power Supply for Key Storage	Programming Method
Volatile Key	■ Re-programmable ■ Erasable	Required external battery, V_{CCBAT} (1)	On-board
Non-volatile Key	One-time programming	Does not require an external battery	On-board and in-socket programming (2)

Notes to Table 9–19:

(1) V_{CCBAT} is a dedicated power supply for volatile key storage and not shared with other on-chip power supplies, such as V_{CCIO} or V_{CCPGM}. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition.

(2) In-socket programming is offered through third party vendors.

Both non-volatile and volatile key programming offers protection from reverse engineering and copying. If you set the tamper-protection bit, the design is also protected from tampering.

Perform key programming through the JTAG interface. Also, ensure that the nSTATUS pin is released high before any key-programming attempts.

To clear the volatile key, issue the KEY_CLR_VREG JTAG instruction. To verify the volatile key has been cleared, issue the KEY_VERIFY JTAG instruction.

For more information about KEY_CLR_VREG and KEY_VERIFY JTAG instructions, refer to the *JTAG Boundary-Scan Testing in Stratix V Devices* chapter.

For more information about battery specifications, refer to the *Stratix V Device Datasheet*.

For more information about the V_{CCBAT} pin connection recommendations, refer to the *Stratix V Device Family Pin Connection Guidelines*.

Security Modes

Table 9–20 lists the security modes available in Stratix V devices.

Table 9–20. Supported Security Modes

Security Mode	Tamper protection Bit Setting	Device Accepts Unencrypted File	Device Accepts Encrypted File	Security Level
No-key	—	Yes	No	—
Volatile Key	—	Yes (2)	Yes	Secure
Volatile Key with Tamper Protection Bit Set	Set (1)	No	Yes	Secure with tamper resistant
Non-volatile Key	—	Yes (2)	Yes	Secure
Non-volatile Key with Tamper Protection Bit Set	Set (1)	No	Yes	Secure with tamper resistant

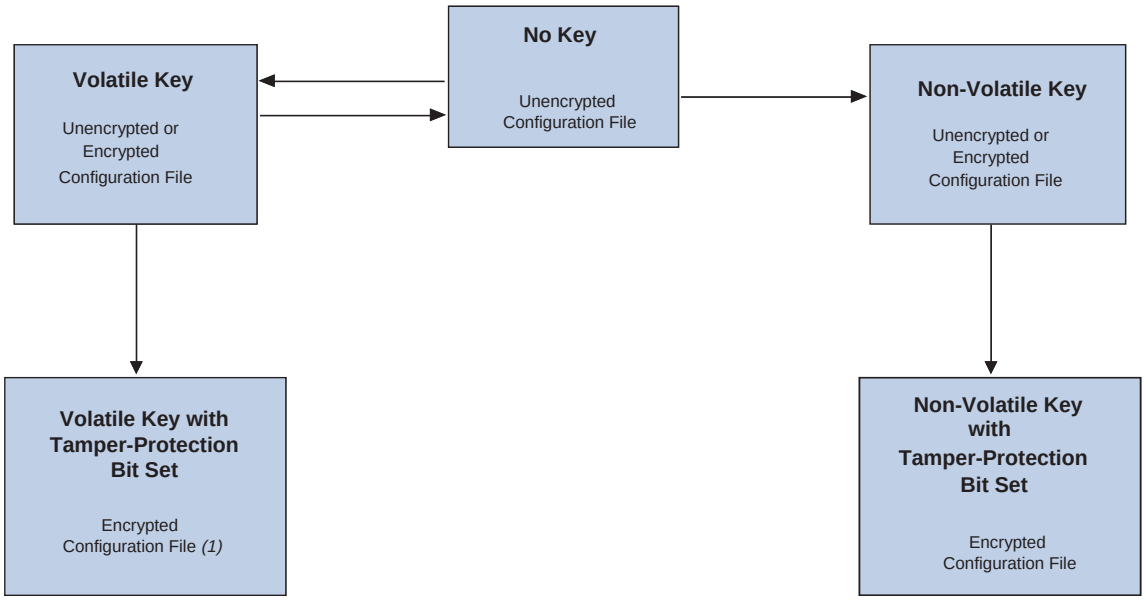
Notes to Table 9–20:

(1) Enabling the tamper protection bit disables test mode in Stratix V devices and disables programming through the JTAG interface. This process is irreversible and prevents Altera from carry-out failure analysis. Contact Altera Technical Support to enable the tamper protection bit.

(2) Use the unencrypted configuration bitstream support only for board-level testing.

Figure 9–33 shows the sequence of the security modes available in Stratix V devices.

Figure 9–33. Stratix V Security Modes—Sequence and Restrictions



Note to Figure 9–33:

(1) Stratix V devices do not accept the encrypted configuration file if the volatile key is erased. You must use the volatile key without tamper-protection bit set to reprogram the key if the volatile key in Stratix V device is erased.

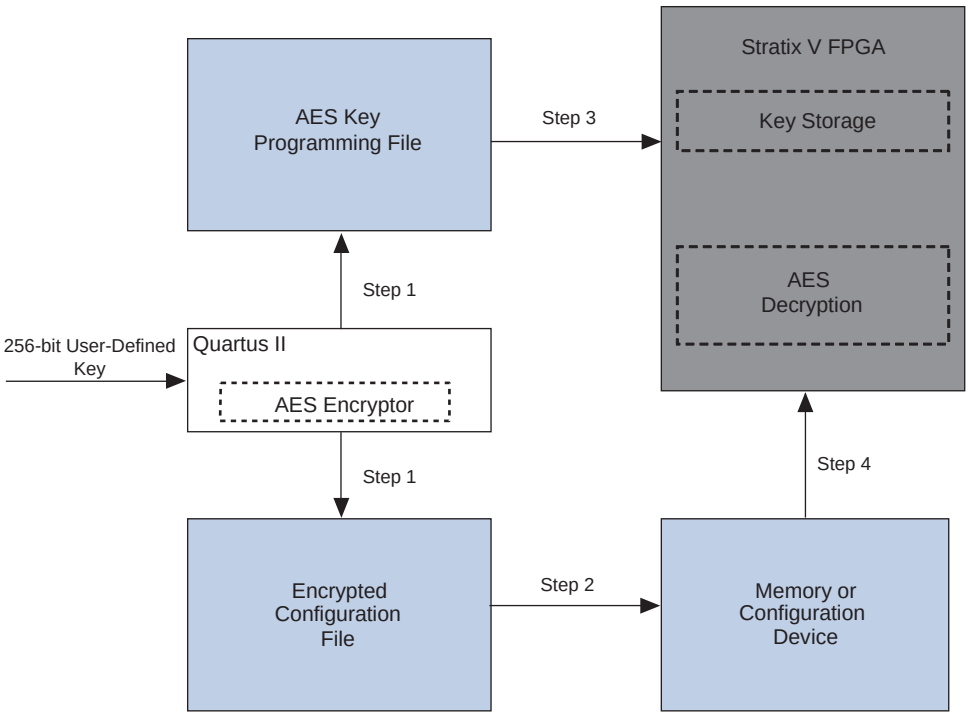
Design Security Implementation Steps

Stratix V devices are SRAM-based devices. To provide design security, Stratix V devices require a 256-bit security key for configuration bitstream design security. To carry out secure configuration, follow these steps:

1. The Quartus II software generates the design security key programming file and encrypts the configuration data using the user-defined 256-bit key.
2. Store the encrypted configuration file in the external memory.
3. Program the AES key programming file into the Stratix V device through a JTAG interface.
4. Configure the Stratix V device. At system power-up, the external memory device sends the encrypted configuration file to the Stratix V device.

Figure 9–34 shows the design security implementation steps.

Figure 9–34. Design Security Implementation Steps



Document Revision History

Table 9–21 lists the revision history for this chapter.

Table 9–21. Document Revision History

Date	Version	Changes
June 2012	1.7	■ Added MAX V devices. ■ Updated Figure 9–2, Figure 9–3, Figure 9–11, Figure 9–16, Figure 9–17, Figure 9–20, and Figure 9–23. ■ Updated Table 9–4, Table 9–5, Table 9–7, Table 9–11, and Table 9–12. ■ Updated "MSEL Pin Settings" and "FPP Multi-Device Configuration" sections.
February 2012	1.6	■ Updated "Security Key Types" section. ■ Updated Table 9–10.
December 2011	1.5	■ Updated "FPP Configuration Timing", "JTAG Secure Mode", and "Security Key Types" sections. ■ Updated Table 9–8.
November 2011	1.4	■ Updated Table 9–5, Table 9–9, and Table 9–14. ■ Updated Figure 9–8, Figure 9–9, and Figure 9–21. ■ Updated "AS Multi-Device Configuration" and "Active Serial Configuration (Serial Configuration Devices)" sections.
May 2011	1.3	■ Chapter moved to volume 2 for the 11.0 release. ■ Added "Remote System Upgrade Using EPCQ 256" and "JTAG Secure Mode" sections. ■ Updated Table 9–5. ■ Updated "Configuration", "Configuration Error", "Programming EPCS and EPCQ", "JTAG Configuration", "Remote Update Mode", and "Design Security" sections. ■ Minor text edits.
January 2011	1.2	■ Updated Table 9–7, Table 9–8, Table 9–12, and Table 9–14. ■ Updated Figure 9–15 and Figure 9–21. ■ Updated "User Watchdog Timer", "DCLK-to-DATA[] Ratio for FPP configuration", "V_{CCPD} Pin", "POR Delay Specification", and "Programming EPCS and EPCQ" sections.
December 2010	1.1	No changes to the content of this chapter for the Quartus II software 10.1.
July 2010	1.0	Initial release.