10. Configuration, Design Security, and Remote System Upgrades in Stratix IV Devices

SIV51010-3.5

This chapter describes the configuration, design security, and remote system upgrades in Stratix® IV devices. To save configuration memory space and time, Stratix IV devices provide configuration data decompression. They also provide a built-in design security feature that protects your designs against IP theft and tampering of your configuration files.

Stratix IV devices also offer remote system upgrade capability so that you can upgrade your system in real-time through any network. This helps to deliver feature enhancements and bug fixes and provides error detection, recovery, and status information to ensure reliable reconfiguration.

Overview

This chapter describes supported configuration schemes for Stratix IV devices, instructions about how to execute the required configuration schemes, and the necessary pin settings.

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

- Fast passive parallel (FPP)
- Fast active serial (AS)
- Passive serial (PS)
- Joint Test Action Group (JTAG)

All configuration schemes use either an external controller (for example, a MAX® II device or microprocessor), a configuration device, or a download cable. For more information, refer to "Configuration Features" on page 10–4.

This chapter includes the following sections:

- "Configuration Schemes" on page 10–2
- "Configuration Features" on page 10–4
- "Fast Passive Parallel Configuration" on page 10–6
- "Fast Active Serial Configuration (Serial Configuration Devices)" on page 10–16
- "Passive Serial Configuration" on page 10–24
- "JTAG Configuration" on page 10–34
- "Device Configuration Pins" on page 10–39

- "Configuration Data Decompression" on page 10–47
- "Remote System Upgrades" on page 10–49
- "Remote System Upgrade Mode" on page 10–53
- "Dedicated Remote System Upgrade Circuitry" on page 10–56
- "Quartus II Software Support" on page 10–62
- "Design Security" on page 10–63

Configuration Devices

Altera® serial configuration devices support a single-device and multi-device configuration solution for Stratix IV devices and are used in the fast AS configuration scheme. Serial configuration devices offer a low-cost, low pin-count configuration solution.

For information about serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* in volume 2 of the *Configuration Handbook*.

All minimum timing information in this chapter covers the entire Stratix IV family. Some devices may work at less than the minimum timing stated in this handbook due to process variation.

Configuration Schemes

Select the configuration scheme by driving the Stratix IV device `MSEL` pins either high or low, as shown in Table 10–1. The `MSEL` input buffers are powered by the V_{CC} power supply. Altera recommends hard wiring the `MSEL[]` pins to V_{CCPGM} and GND. The `MSEL[2..0]` pins have 5-kΩ internal pull-down resistors that are always active. During power-on reset (POR) and during reconfiguration, the `MSEL` pins must be at V_{IL} and V_{IH} levels of V_{CCPGM} voltage to be considered logic low and logic high.

To avoid problems with detecting an incorrect configuration scheme, hardwire the `MSEL[]` pins to V_{CCPGM} and GND without pull-up or pull-down resistors. Do not drive the `MSEL[]` pins by a microprocessor or another device.

Table 10–1. Configuration Schemes for Stratix IV Devices (Part 1 of 2)

Configuration Scheme	MSEL2	MSEL1	MSEL0
Fast passive parallel	0	0	0
Passive serial	0	1	0
Fast AS (40 MHz) *(1)*	0	1	1
Remote system upgrade fast AS (40 MHz) *(1)*	0	1	1
FPP with design security feature and/or decompression enabled *(2)*	0	0	1

Table 10–1. Configuration Schemes for Stratix IV Devices (Part 2 of 2)

Configuration Scheme	MSEL2	MSEL1	MSEL0
JTAG-based configuration *(4)*	*(3)*	*(3)*	*(3)*

Notes to Table 10–1:

(1) Stratix IV devices only support fast AS configuration. You must use either EPCS64 or EPCS128 devices to configure a Stratix IV device in fast AS mode.

(2) These modes are only supported when using a MAX II device or a microprocessor with flash memory for configuration. In these modes, the host system must output a DCLK that is ×4 the data rate.

(3) Do not leave the MSEL pins floating, connect them to V_{CCPGM} or GND. These pins support the non-JTAG configuration scheme used in production. If you only use the JTAG configuration, connect the MSEL pins to GND.

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

Table 10–2 lists the uncompressed raw binary file (**.rbf**) configuration file sizes for Stratix IV devices.

Table 10–2. Uncompressed Raw Binary File (.rbf) Sizes for Stratix IV Devices

Device	Data Size (Bits)
EP4SE230	94,557,472
EP4SE360	128,395,584
EP4SE530	171,722,064
EP4SE820	241,684,472
EP4SGX70	47,833,352
EP4SGX110	47,833,352
EP4SGX180	94,557,472
EP4SGX230	94,557,472
EP4SGX290	128,395,584
	171,722,064 *(1)*
EP4SGX360	128,395,584
	171,722,064 *(1)*
EP4SGX530	171,722,064
EP4S40G2	94,557,472
EP4S40G5	171,722,064
EP4S100G2	94,557,472
EP4S100G3	171,722,064
EP4S100G4	171,722,064
EP4S100G5	171,722,064

Note to Table 10–2:

(1) This only applies to the F45 package.

Use the data in Table 10–2 to estimate the file size before design compilation. Different configuration file formats; for example, a hexidecimal (**.hex**) or tabular text file (**.ttf**) format, have different file sizes. Refer to the Quartus® II software for the different types of configuration file and file sizes. However, for any 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 the 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*.

Configuration Features

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

Table 10–3 lists which configuration features you can use in each configuration scheme.

Table 10–3. Configuration Features for Stratix IV Devices

Configuration Scheme	Configuration Method	Decompression	Design Security	Remote System Upgrade
FPP	MAX II device or a microprocessor with flash memory	Y *(1)*	Y *(1)*	Y
Fast AS	Serial configuration device	Y	Y	Y
PS	MAX II device or a microprocessor with flash memory	Y	Y	—
	Download cable	Y	Y	—
JTAG	MAX II device or a microprocessor with flash memory	—	—	—
	Download cable	—	—	—

Note to Table 10–3:

(1) In these modes, the host system must send a DCLK that is ×4 the data rate.

You can also refer to the following:

- For more information about the configuration data decompression feature, refer to "Configuration Data Decompression" on page 10–47.
- For more information about the remote system upgrade feature, refer to "Remote System Upgrades" on page 10–49.
- For more information about the design security feature, refer to "Design Security" on page 10–63.

If your system already contains a common flash interface (CFI) flash memory, you can use it for Stratix IV device configuration storage as well. The MAX II parallel flash loader (PFL) feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface and provides the logic to control configuration from the flash memory device to the Stratix IV device. Both PS and FPP configuration modes are supported using this PFL feature.

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

For more information about programming Altera serial configuration devices, refer to "Programming Serial Configuration Devices" on page 10–22.

Power-On Reset Circuit

The POR circuit keeps the entire system in reset until the power supply voltage levels have stabilized on power-up. After power-up, the device does not release nSTATUS until V_{CC}, V_{CCAUX}, V_{CCPT}, V_{CCPGM}, and V_{CCPD} are above the device's POR trip point. On power down, brown-out occurs if the V_{CC}, V_{CCAUX}, V_{CCPT}, V_{CCPGM}, or V_{CCPD} drops below the threshold voltage.

In Stratix IV devices, a pin-selectable option (PORSEL) is provided that allows you to select between the standard POR time or fast POR time. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms, which has a lower power-ramp rate. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms.

V_{CCPGM} Pins

Stratix IV devices have a power supply, V_{CCPGM}, for all the dedicated configuration pins and dual function pins. The supported configuration voltage is 1.8, 2.5, and 3.0 V. Stratix IV devices do not support 1.5 V configuration.

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

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

V_{CCPD} Pins

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

V_{CCPGM} and V_{CCPD} must ramp up from 0 V to the desired voltage level within 100 ms when PORSEL is low or 4 ms when PORSEL is high. If these supplies are not ramped up within this specified time, your Stratix IV device will not configure successfully. If your system cannot ramp up the power supplies within 100 ms or 4 ms, you must hold nCONFIG low until all the power supplies are stable.

V_{CCPD} must be greater than or equal to V_{CCIO} of the same bank. If V_{CCIO} of the bank is set to 3.0 V, V_{CCPD} must be powered up to 3.0 V. If the V_{CCIO} of the bank is powered to 2.5 V or lower, V_{CCPD} must be powered up to 2.5 V.

For more information about configuration pins power supply, refer to "Device Configuration Pins" on page 10–39.

Fast Passive Parallel Configuration

Fast passive parallel configuration in Stratix IV devices is designed to meet the continuously increasing demand for faster configuration times. Stratix IV devices are designed with the capability of receiving byte-wide configuration data per clock cycle.

You can perform FPP configuration of Stratix IV devices using an intelligent host, such as a MAX II device or a microprocessor.

FPP Configuration Using a MAX II Device as an External Host

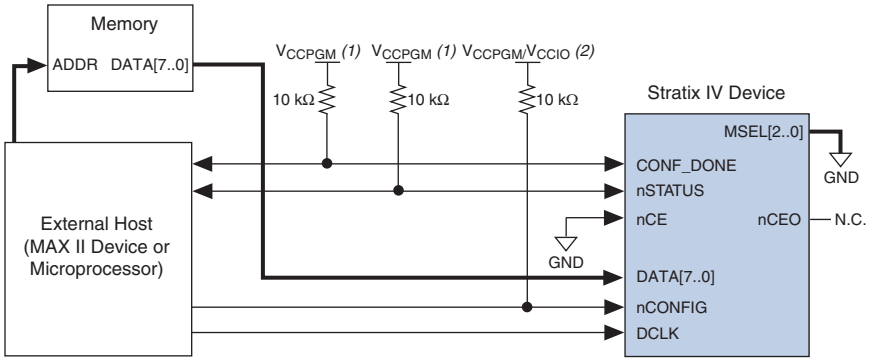
FPP configuration using an external host provides the fastest method to configure Stratix IV devices. In this configuration scheme, you can use a MAX II device as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** format. When using the MAX II device as an intelligent host, a design that controls the configuration process, such as fetching the data from flash memory and sending it to the device, must be stored in the MAX II device.

If you are using the Stratix IV decompression and/or design security features, the external host must be able to send a DCLK frequency that is ×4 the data rate.

The ×4 DCLK signal does not require an additional pin and is sent on the DCLK pin. The maximum DCLK frequency is 125 MHz, which results in a maximum data rate of 250 Mbps. If you are not using the Stratix IV decompression or design security features, the data rate is ×8 the DCLK frequency.

Figure 10–1 shows the configuration interface connections between the Stratix IV device and a MAX II device for single device configuration.

Figure 10–1. Single Device FPP Configuration Using an External Host



Note to Figure 10–1:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix IV 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) A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host is not driving the line.

After power-up, the Stratix IV device goes through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be 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 POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages: reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in the reset stage. To initiate configuration, the MAX II device must drive the nCONFIG pin from low to high.

To begin the configuration process, you must fully power V_{CCPT}, V_{CC}, V_{CCPD}, and V_{CCPGM} of the banks where the configuration pins reside to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the MAX II device places the configuration data one byte at a time on the DATA[7..0] pins.

A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host (a Max II CPLD or a microcontroller) is not driving the line. For example, during external host reprogramming or power-up where the I/O driving nCONFIG may be tri-stated. If a pull-up resistor is added to the nCONFIG line, the FPGA stays in user mode if the external host is being reprogrammed. If a pull-down resistor is added to the nCONFIG line, the FPGA goes into reset mode if the external host is being reprogrammed. Whenever the nCONFIG line is released high, ensure that the first DCLK and DATA are not driven unintentionally.

Stratix IV devices receive configuration data on the DATA[7..0] pins and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. If you are using the Stratix IV decompression and/or design security features, configuration data is latched on the rising edge of every first DCLK cycle out of the four DCLK cycles. Altera recommends that you to keep the data on DATA[7. . 0] stable for the next 3 clock cycles when the data is being processed. You can only stop DCLK after three clock cycles after the last data is latched.

Data is continuously clocked into the target device until CONF_DONE goes high. The CONF_DONE pin goes high one byte early in FPP modes. The last byte is required for FPP mode. After the device has received the next-to-last byte of the configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

In Stratix IV devices, the initialization clock source is either the internal oscillator or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix IV device provides itself with enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You can also synchronize initialization of multiple devices or delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. Supplying a clock on CLKUSR does not affect the configuration process. The CONF_DONE pin goes high one byte early in FPP modes. The last byte is required for FPP mode. After the CONF_DONE pin transitions high, CLKUSR is enabled after the time specified at t_{CD2CU}. After this time period elapses, Stratix IV devices require 8,532 clock cycles to initialize properly and enter user mode. Stratix IV devices support a CLKUSR f_{MAX} of 125 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. This **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it is high because of an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. 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 is complete, the INIT_DONE pin is released and pulled high. The MAX II device must be able to detect this low-to-high transition, which signals the device has entered user mode. When initialization is complete, the device enters user mode. In user-mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

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

To ensure DCLK and DATA[7..0] are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[7..0] pins are available as user I/O pins after configuration. When you select the FPP scheme as a default in the Quartus II software, these I/O pins are tri-stated in user mode. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause configuration by halting DCLK for an indefinite amount of time.

If you need to stop DCLK, it can only be stopped:

- three clock cycles after the last data byte was latched into the Stratix IV device when you use the decompression and/or design security features.
- two clock cycles after the last data byte was latched into the Stratix IV device when you do not use the Stratix IV decompression and/or design security features.

By stopping DCLK, the configuration circuit allows enough clock cycles to process the last byte of latched configuration data. When the clock restarts, the MAX II device must provide data on the DATA[7..0] pins prior to sending the first DCLK rising edge.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box) is turned on, the device releases nSTATUS after a reset time-out period (a maximum of 500 µs). After nSTATUS is released and pulled high by a pull-up resistor, the MAX II device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

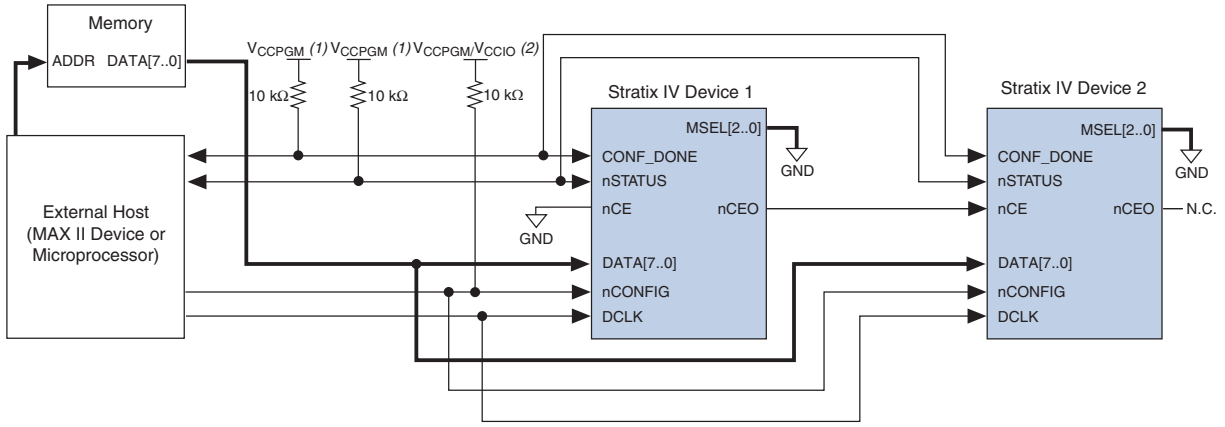
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The MAX II device must monitor the CONF_DONE pin to detect errors and determine when programming completes. If all the configuration data is sent, but the CONF_DONE or INIT_DONE signals have not gone high, the MAX II device reconfigures the target device.

If you use the optional CLKUSR pin and nCONFIG is pulled low to restart the configuration during device initialization, ensure that CLKUSR continues toggling during the time nSTATUS is low (a maximum of 500 µs).

When the device is in user mode, initiating reconfiguration is done by transitioning the nCONFIG pin low-to-high. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. After nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 10–2 shows how to configure multiple Stratix IV devices using a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the devices are cascaded for multi-device configuration.

Figure 10–2. Multi-Device FPP Configuration Using an External Host



Note to Figure 10–2:

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

(2) A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host is not driving the line.

In a multi-device FPP configuration, the first device's nCE pin is connected to GND while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. 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 within one clock cycle; therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. The configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

All nSTATUS and CONF_DONE pins are tied together; if any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain. 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.

If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (a maximum of 500 µs). After all nSTATUS pins are released and pulled high, the MAX II device tries to reconfigure the chain without pulsing nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

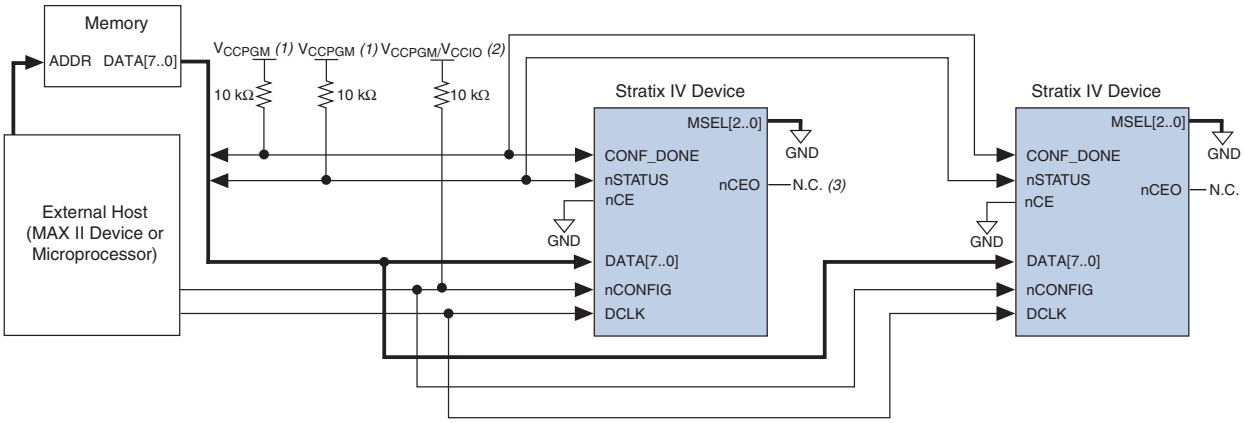
If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

In a multi-device FPP configuration chain, all Stratix IV devices in the chain must either enable or disable the decompression and/or design security features. You cannot selectively enable the decompression and/or design security features for each device in the chain because of the DATA and DCLK relationship. If the chain contains devices that do not support design security, use a serial configuration scheme.

If a system has multiple devices that contain the same configuration data, tie all device nCE inputs to GND and leave the nCEO pins floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA[7..0], and CONF_DONE) are connected to every device in the chain. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices start and complete configuration at the same time.

Figure 10–3 shows a multi-device FPP configuration when both Stratix IV devices are receiving the same configuration data.

Figure 10–3. Multiple-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data



Notes to Figure 10–3:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. 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) A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host is not driving the line.

(3) The nCEO pins of both Stratix IV devices are left unconnected when configuring the same configuration data into multiple devices.

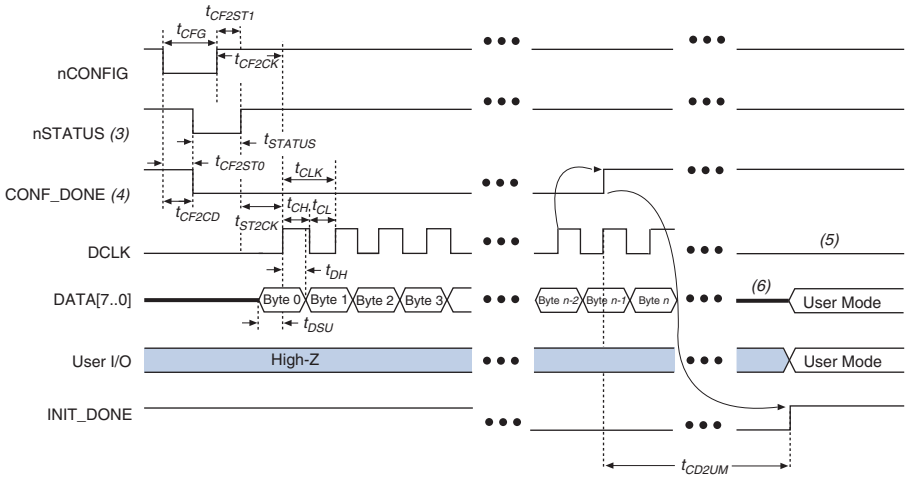
You can use a single configuration chain to configure Stratix IV devices with other Altera devices that support FPP configuration, such as other types of Stratix devices. To ensure that all devices in the chain complete configuration at the same time, or that an error flagged by one device initiates reconfiguration in all devices, tie all of the device CONF_DONE and nSTATUS pins together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* in volume 2 of the *Configuration Handbook*.

FPP Configuration Timing

Figure 10–4 shows the timing waveform for an FPP configuration when using a MAX II device as an external host. This waveform shows the timing when you have not enabled the decompression and design security features.

Figure 10–4. FPP Configuration Timing Waveform ***(Note 1), (2)***



Notes to Figure 10–4:

(1) Use this timing waveform when you have not enabled the decompression and design security features.

(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 IV 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) DATA[7..0] are available as user I/O pins after configuration except for some exceptions on Stratix IV GT devices. The state of these pins depends on the dual-purpose pin settings.

Table 10–4 lists the timing parameters for Stratix IV devices for an FPP configuration when you have not enabled the decompression and design security features.

Table 10–4. FPP Timing Parameters for Stratix IV Devices (Part 1 of 2) ***(Note 1), (2)***

Symbol	Parameter	Minimum			Maximum			Units
		Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	
t_{CF2CD}	nCONFIG low to CONF_DONE low	—			800			ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—			800			ns
t_{CFG}	nCONFIG low pulse width	2			—			µs
t_{STATUS}	nSTATUS low pulse width	10			500 *(3)*			µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—			500 *(4)*			µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	500			—			µs

Table 10–4. FPP Timing Parameters for Stratix IV Devices (Part 2 of 2) *(Note 1), (2)*

Symbol	Parameter	Minimum			Maximum			Units
		Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2			—			µs
t_{DSU}	Data setup time before rising edge on DCLK	4			—			ns
t_{DH}	Data hold time after rising edge on DCLK	1			—			ns
T_R	Input rise time	—			40			ns
t	Input fall time	—			40			ns
t_{CD2UM}	CONF_DONE high to user mode *(5)*	55			150			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} + (8532 × CLKUSR period)			—			—
t_{CH}	DCLK high time *(6)*	3.6	4.5	5.6	—			ns
t_{CL}	DCLK low time *(6)*	3.6	4.5	5.6	—			ns
t_{CLK}	DCLK period *(6)*	8	10	12.5	—			ns
f_{MAX}	DCLK frequency	—			125	100	80	MHz

Notes to Table 10–4:

(1) This information is preliminary.

(2) Use these timing parameters when you have not enabled the decompression and design security features.

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

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

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

(6) Adding up t_{CH} and t_{CL} equals to t_{CLK}. When EP4SE230 t_{CH} is 3.6 ns (min), t_{CL} must be 4.4 ns and vice versa.

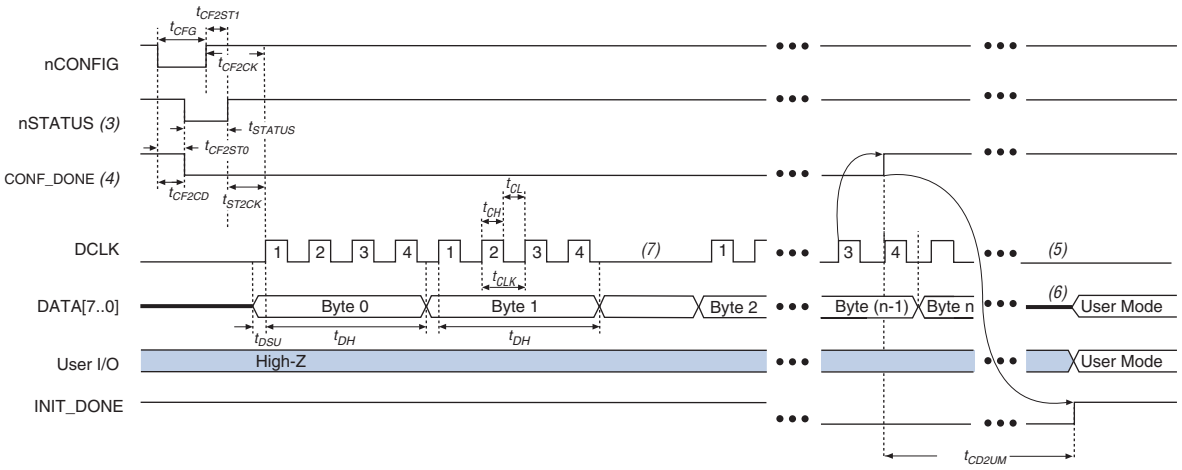
(7) Applicable to EP4SE230, EP4SE360, EP4SGX70, EP4SGX110, EP4SGX180, EP4SGX230, EP4SGX290 (except F45 package), EP4SGX360 (except F45 package), EP4S40G2, EP4S100G2 devices.

(8) Applicable to EP4SE530, EP4SGX290 (only for F45 package), EP4SGX360 (only for F45 package), EP4SGX530, EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5 devices.

(9) Applicable to EP4SE820 only.

Figure 10–5 shows the timing waveform for an FPP configuration when using a MAX II device as an external host. This waveform shows the timing when you have enabled the decompression and/or design security features.

Figure 10–5. FPP Configuration Timing Waveform with Decompression or Design Security Feature Enabled *(Note 1), (2)*



Notes to Figure 10–5:

(1) Use this timing waveform when you have enabled the decompression and/or design security features.

(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 IV 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) DATA[7..0] are available as user I/O pins after configuration except for some exceptions on Stratix IV GT devices. The state of these pins depends on the dual-purpose pin settings.

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

Table 10–5 lists the timing parameters for Stratix IV devices for an FPP configuration when you enable the decompression and/or the design security features.

Table 10–5. FPP Timing Parameters for Stratix IV Devices with the Decompression and/or Design Security Features Enabled *(Note 1), (2)* (Part 1 of 2)

Symbol	Parameter	Minimum			Maximum			Units
		Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	
t_{CF2CD}	nCONFIG low to CONF_DONE low	—			800			ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—			800			ns
t_{CFG}	nCONFIG low pulse width	2			—			µs
t_{STATUS}	nSTATUS low pulse width	10			500 *(3)*			µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—			500 *(4)*			µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	500			—			µs

Table 10–5. FPP Timing Parameters for Stratix IV Devices with the Decompression and/or Design Security Features Enabled *(Note 1)*, *(2)* (Part 2 of 2)

Symbol	Parameter	Minimum			Maximum			Units
		Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	Stratix IV *(7)*	Stratix IV *(8)*	Stratix IV *(9)*	
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2			—			µs
t_{DSU}	Data setup time before rising edge on DCLK	4			—			ns
t_{DH}	Data hold time after rising edge on DCLK	3/(DCLK frequency) + 1			—			s
t_{DATA}	Data rate	—			250			Mbps
t_R	Input rise time	—			40			ns
t	Input fall time	—			40			ns
t_{CD2UM}	CONF_DONE high to user mode *(5)*	55			150			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 *(5)*	t_{CD2CU} + (8532 × CLKUSR period)			—			—
t_{CH}	DCLK high time *(6)*	3.6	4.5	5.6	—			ns
t_{CL}	DCLK low time *(6)*	3.6	4.5	5.6	—			ns
t_{CLK}	DCLK period *(6)*	8	10	12.5	—			ns
f_{MAX}	DCLK frequency	—			125	100	80	MHz

Notes to Table 10–5:

(1) This information is preliminary.
(2) Use these timing parameters when you use the decompression and/or design security features.
(3) You can obtain this value if you do not delay the configuration by extending the nCONFIG or nSTATUS low pulse width.
(4) This value is applicable if you do not delay the configuration by externally holding nSTATUS low.
(5) The minimum and maximum numbers apply only if you chose the internal oscillator as the clock source for starting the device.
(6) Adding up t_{CH} and t_{CL} equals to t_{CLK}. When EP4SE230 t_{CH} is 3.6 ns (min), t_{CL} must be 4.4 ns and vice versa.
(7) Applicable for EP4SE230, EP4SE360, EP4SGX70, EP4SGX110, EP4SGX180, EP4SGX230, EP4SGX290 (except F45 package), EP4SGX360 (except F45 package), EP4S40G2, EP4S100G2 devices.
(8) Applicable for EP4SE530, EP4SGX290 (only for F45 package), EP4SGX360 (only for F45 package), EP4SGX530, EP4S40G5, EP4S100G3, EP4S100G4, EP4S100G5 devices.
(9) Applicable to EP4SE820 only.

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

FPP Configuration Using a Microprocessor

In this configuration scheme, a microprocessor can control the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device.

All information in "FPP Configuration Using a MAX II Device as an External Host" on page 10–6 is also applicable when using a microprocessor as an external host. Refer to this section for all configuration and timing information.

Fast Active Serial Configuration (Serial Configuration Devices)

In the fast AS configuration scheme, Stratix IV devices are configured using a serial configuration device. These configuration devices are low-cost devices with non-volatile memory that feature a simple four-pin interface and a small form factor.

The largest serial configuration device currently supports 128 MBits of configuration bitstream. Use the Stratix IV decompression features or select an FPP or PS configuration scheme for EP4SE360, EP4SGX290, EP4S40G5, EP4S100G3 and larger devices.

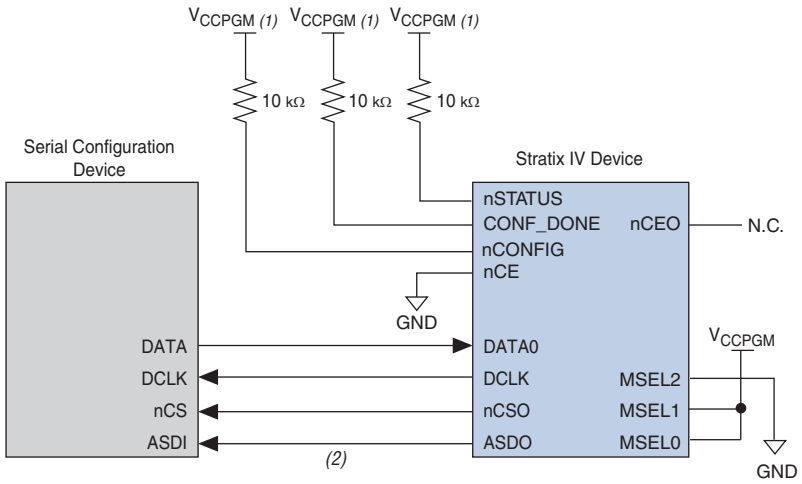
For more information about serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

Serial configuration devices provide a serial interface to access configuration data. During device configuration, Stratix IV devices read configuration data using the serial interface, decompress data if necessary, and configure their SRAM cells. This scheme is referred to as the AS configuration scheme because the Stratix IV device controls the configuration interface. This scheme contrasts with the PS configuration scheme where the configuration device controls the interface.

The Stratix IV decompression and design security features are fully available when configuring your Stratix IV device using fast AS mode.

Serial configuration devices have a four-pin interface—serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and an active-low chip select (nCS). This four-pin interface connects to Stratix IV device pins, as shown in Figure 10–6.

Figure 10–6. Single Device Fast AS Configuration



Notes to Figure 10–6:

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

(2) Stratix IV devices use the ASDO-to-ASDI path to control the configuration device.

You can power the EPCS serial configuration device with 3.0 V when you configure the Stratix IV FPGA using Active Serial (AS) configuration mode. This is feasible because the power supply to the EPCS device ranges between 2.7 V and 3.6 V. You do not need a dedicated 3.3 V power supply to power the EPCS device. The EPCS device and the VCCPGM pins on the Stratix IV device may share the same 3.0 V power supply.

After power-up, the Stratix IV devices go through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS and CONF_DONE low, and tri-states all user I/O pins. After the device successfully exits POR, all the user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors, which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages—reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. After POR, the Stratix IV device releases nSTATUS, which is pulled high by an external 10-kΩ pull-up resistor and enters configuration mode.

To begin configuration, power the V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} voltages (for the banks where the configuration pins reside) to the appropriate voltage levels.

The serial clock (DCLK) generated by the Stratix IV device controls the entire configuration cycle and provides timing for the serial interface. Stratix IV devices use an internal oscillator to generate DCLK. Using the MSEL[] pins, you can select to use a 40 MHz oscillator.

In fast AS configuration schemes, Stratix IV devices drive out control signals on the falling edge of DCLK. The serial configuration device responds to the instructions by driving out configuration data on the falling edge of DCLK. Then the data is latched into the Stratix IV device on the following falling edge of DCLK.

In configuration mode, Stratix IV devices enable the serial configuration device by driving the nCSO output pin low, which connects to the chip select (nCS) pin of the configuration device. The Stratix IV device uses the serial clock (DCLK) and serial data output (ASDO) pins to send operation commands and/or read address signals to the serial configuration device. The configuration device provides data on its serial data output (DATA) pin, which connects to the DATA0 input of the Stratix IV devices.

After all the configuration bits are received by the Stratix IV device, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ resistor. Initialization begins only after the CONF_DONE signal reaches a logic high level. All AS configuration pins (DATA0, DCLK, nCSO, and ASDO) have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and are driven high by the weak internal pull-up resistors. The CONF_DONE pin must have an external 10-kΩ pull-up resistor in order for the device to initialize.

In Stratix IV devices, the initialization clock source is either the internal oscillator or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix IV device provides itself with enough clock cycles for proper initialization. You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. When you select the **Enable user supplied start-up clock** option, the CLKUSR pin is the initialization clock source. Supplying a clock on CLKUSR does not affect the configuration process. After all configuration data is accepted and CONF_DONE goes high, CLKUSR is enabled after four clock cycles of DCLK. After this time period elapses, Stratix IV devices require 8,532 clock cycles to initialize properly and enter user mode. Stratix IV devices support a CLKUSR f_{MAX} of 125 MHz.

An optional INIT_DONE pin is available, which signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it is high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. 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 is complete, the INIT_DONE pin is released and pulled high. This low-to-high transition signals that the device has entered user mode. When initialization is complete, the device enters user mode. In user mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

If an error occurs during configuration, Stratix IV devices assert the nSTATUS signal low, indicating a data frame error, and the CONF_DONE signal stays low. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Stratix IV device resets the configuration device by pulsing nCSO, releases nSTATUS after a reset time-out period (a maximum of 500 µs), and retries configuration. If this option is turned off, the system must monitor nSTATUS for errors and then pulse nCONFIG low for at least 2 µs to restart configuration.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

When the Stratix IV device is in user mode, you can initiate reconfiguration by pulling the nCONFIG pin low. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. After nCONFIG returns to a logic high level and nSTATUS is released by the Stratix IV device, reconfiguration begins.

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.

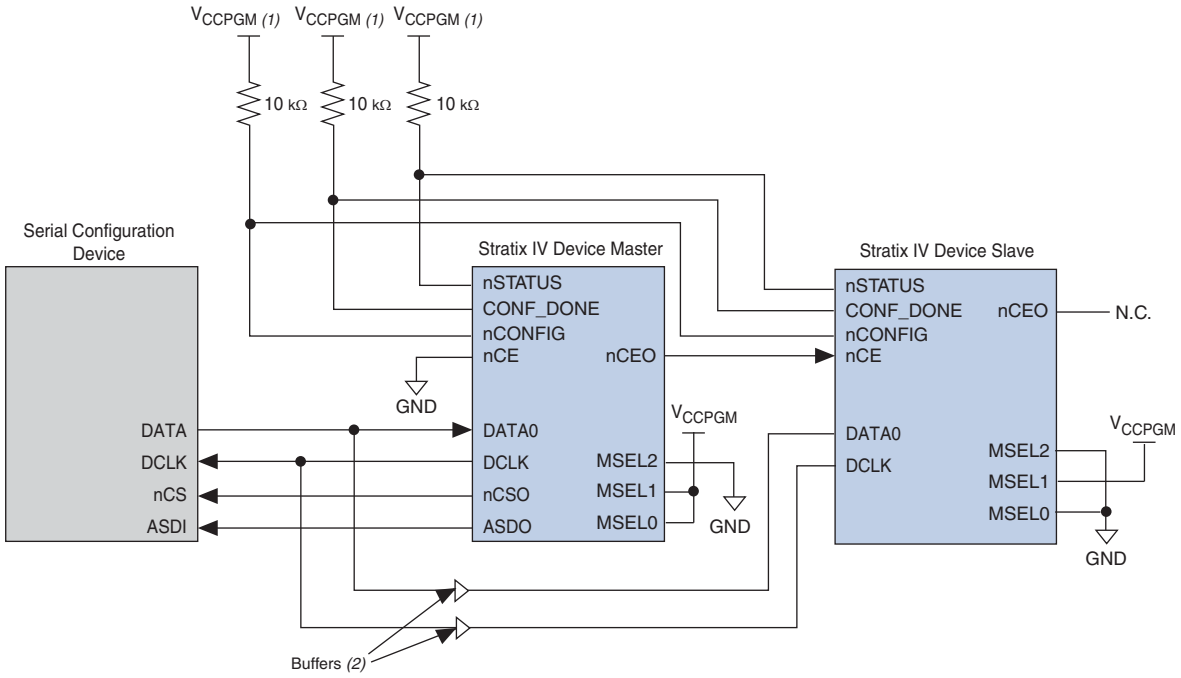
The timing parameters for AS mode are not listed here because the t_{CF2CD}, t_{CF2ST0}, t_{CFG}, t_{STATUS}, t_{CF2ST1}, and t_{CD2UM} timing parameters are identical to the timing parameters for PS mode listed in Table 10–7 on page 10–30.

You can configure multiple Stratix IV devices using a single serial configuration device. You can cascade multiple Stratix IV devices using the chip-enable (nCE) and chip-enable-out (nCEO) pins. The first device in the chain must have its nCE pin connected to GND. You must connect its nCEO pin to the nCE pin of the next device in the chain. When the first device captures all of its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. You must leave the nCEO pin of the last device unconnected. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA0 pins of each device in the chain are connected (refer to Figure 10–7).

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

Figure 10–7 shows the pin connections for the multi-device fast AS configuration.

Figure 10–7. Multi-Device Fast AS Configuration



Notes to Figure 10–7:

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

(2) Connect the repeater buffers between the Stratix IV master and slave device(s) for DATA[0] and DCLK. This is to prevent potential signal integrity and clock skew problems.

As shown in Figure 10–7, the nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors. These pins are open-drain bidirectional pins on the devices. When the first device asserts nCEO (after receiving all of its configuration data), it releases its CONF_DONE pin. But the subsequent devices in the chain keep this shared CONF_DONE line low until they have received their configuration data. When all target devices in the chain have received their configuration data and have released CONF_DONE, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode.

If an error occurs at any point during configuration, the nSTATUS line is driven low by the failing device. If you enable the **Auto-restart configuration after error** option, reconfiguration of the entire chain begins after a reset time-out period (a maximum of 500 µs). If you did not enable the **Auto-restart configuration after error** option, the external system must monitor nSTATUS for errors and then pulse nCONFIG low to restart configuration. The external system can pulse nCONFIG if it is under system control rather than tied to V_{CCGPM}.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

While you can cascade Stratix IV devices, you cannot cascade or chain together serial configuration devices.

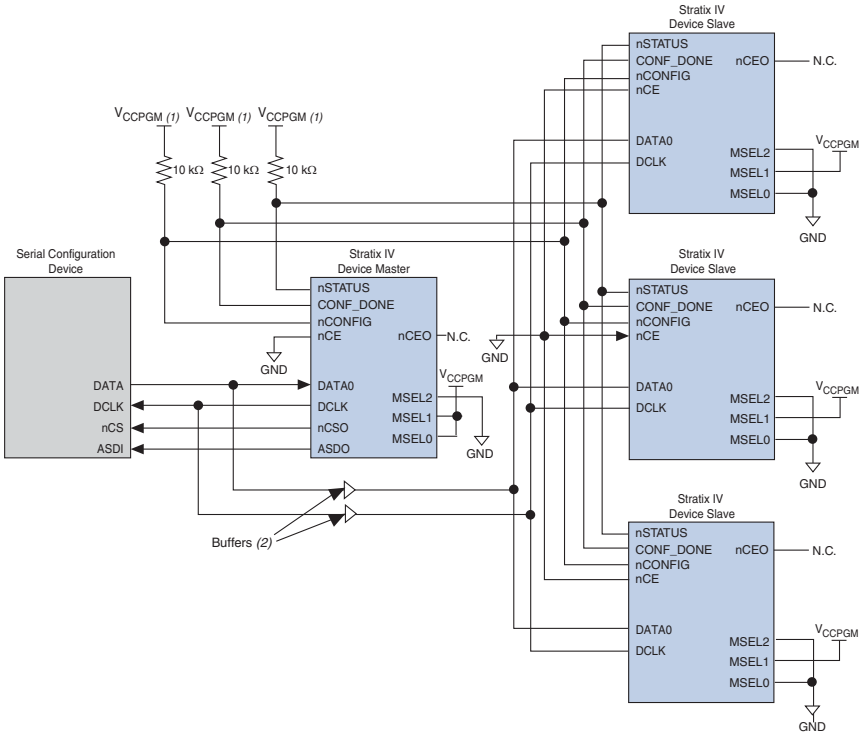
If the configuration bitstream size exceeds the capacity of a serial configuration device, you must select a larger configuration device and/or enable the compression feature. When configuring multiple devices, the size of the bitstream is the sum of the individual device configuration bitstreams.

A system may have multiple devices that contain the same configuration data. In active serial chains, you can implement this by storing one copy of the **.sof** in the serial configuration device. The same copy of the **.sof** configures the master Stratix IV device and all remaining slave devices concurrently. All Stratix IV devices must be the same density and package.

To configure four identical Stratix IV devices with the same **.sof**, set up the chain as shown in Figure 10–8. The first device is the master device and its MSEL pins must be set to select AS configuration. The other three slave devices are set up for concurrent configuration and their MSEL pins must be set to select PS configuration. The nCE input pins from the master and slave are connected to GND, and the DATA and DCLK pins connect in parallel to all four devices. During the configuration cycle, the master device reads its configuration data from the serial configuration device and transmits the configuration data to all three slave devices, configuring all of them simultaneously.

Figure 10–8 shows the multi-device fast AS configuration when the devices receive the same data using a single **.sof**.

Figure 10–8. Multi-Device Fast AS Configuration When the Devices Receive the Same Data Using a Single .sof



Notes to Figure 10–8:

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

(2) Connect the repeater buffers between the Stratix IV master and slave device(s) for `DATA[0]` and `DCLK`. This is to prevent potential signal integrity and clock skew problems.

Estimating Active Serial Configuration Time

Active serial configuration time is dominated by the time it takes to transfer data from the serial configuration device to the Stratix IV device. This serial interface is clocked by the Stratix IV `DCLK` output (generated from an internal oscillator) and must be set to **40 MHz (25 ns)**.Therefore, the minimum configuration time estimate for an EP4SE230 device (94, 600, 000 bits of uncompressed data) is:

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

94, 600, 000 bits × (25 ns / 1 bit) = 2365 ms

Enabling compression reduces the amount of configuration data that is transmitted to the Stratix IV device, which also reduces configuration time. On average, compression reduces configuration time, depending on the design.

Programming Serial Configuration Devices

Serial configuration devices are non-volatile, flash-memory-based devices. You can program these devices in-system using the USB-Blaster™, EthernetBlaster™, or ByteBlaster™ II download cable. Alternatively, you can program them using the Altera programming unit (APU), supported third-party programmers, or a microprocessor with the SRunner software driver.

You can perform in-system programming of serial configuration devices using the conventional AS programming interface or the JTAG interface solution.

Because serial configuration devices do not support the JTAG interface, the conventional method to program them is using the AS programming interface. The configuration data used to program serial configuration devices is downloaded using programming hardware.

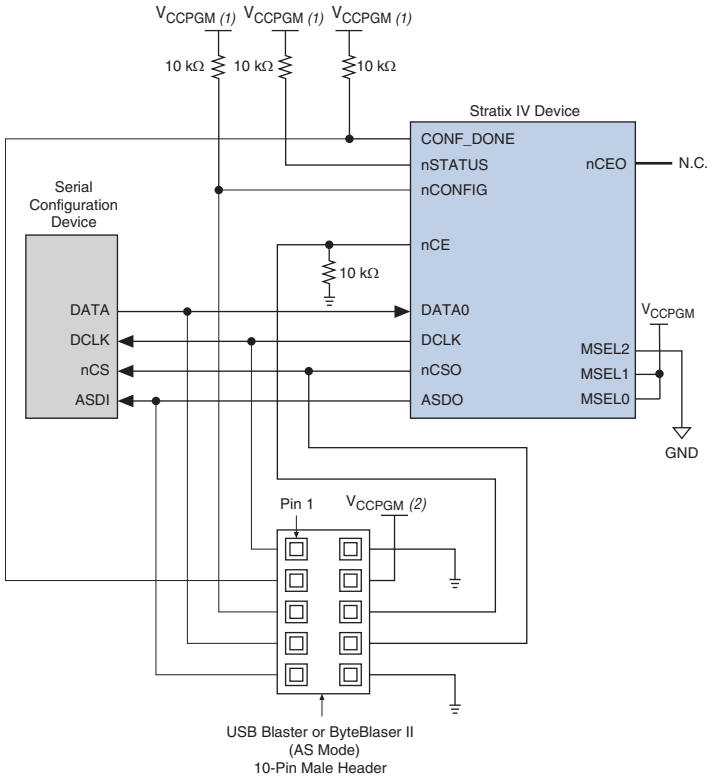
During in-system programming, the download cable disables device access to the AS interface by driving the `nCE` pin high. Stratix IV devices are also held in reset by a low level on `nCONFIG`. After programming is complete, the download cable releases `nCE` and `nCONFIG`, allowing the pull-down and pull-up resistors to drive GND and V_{CCPGM}, respectively. Figure 10–9 shows the download cable connections for the serial configuration device.

Altera has developed Serial FlashLoader (SFL), an in-system programming solution for serial configuration devices using the JTAG interface. This solution requires the Stratix IV device to be a bridge between the JTAG interface and the serial configuration device.

For more information about SFL, refer to *AN 370: Using the Serial FlashLoader with Quartus II Software*.

For more information about the USB-Blaster download cable, refer to the *USB-Blaster Download Cable User Guide*. For more information about the ByteBlaster II cable, refer to the *ByteBlaster II Download Cable User Guide*. For more information about the EthernetBlaster download cable, refer to the *EthernetBlaster Communications Cable User Guide.*

Figure 10–9. In-System Programming of Serial Configuration Devices



Notes to Figure 10–9:

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

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

You can program serial configuration devices with the Quartus II software using the Altera programming hardware and the appropriate configuration device programming adapter.

In production environments, you can program serial configuration devices using multiple methods. You can use Altera programming hardware or other third-party programming hardware to program blank serial configuration devices before they are mounted on PCBs. Alternatively, you can use an on-board microprocessor to program the serial configuration device in-system using C-based software drivers provided by Altera.

You can program a serial configuration device in-system by an external microprocessor using SRunner. SRunner is a software driver developed for embedded serial configuration device programming, which can be easily customized to fit in different embedded systems. SRunner is able to read raw programming data (.**rpd**) and write to serial configuration devices. The serial configuration device programming time using SRunner is comparable to the programming time with the Quartus II software.

For more information about SRunner, refer to *AN 418: SRunner: An Embedded Solution for Serial Configuration Device Programming* and the source code on the Altera website at www.altera.com.

For more information about programming serial configuration devices, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

Guidelines for Connecting Serial Configuration Devices on an AS Interface

For single- and multi-device AS configurations, the board trace length and loading between the supported serial configuration device and the Stratix IV device family must follow the recommendations listed in Table 10–6.

Table 10–6. Maximum Trace Length and Loading for the AS Configuration

Stratix IV Device AS Pins	Maximum Board Trace Length from the Stratix IV Device to the Serial Configuration Device (Inches)	Maximum Board Load (pF)
DCLK	10	15
DATA[0]	10	30
nCSO	10	30
ASDO	10	30

Passive Serial Configuration

You can program a PS configuration for Stratix IV devices using an intelligent host, such as a MAX II device or microprocessor with flash memory, or a download cable. In the PS scheme, an external host (a MAX II device, embedded processor, or host PC) controls configuration. Configuration data is clocked into the target Stratix IV device using the DATA0 pin at each rising edge of DCLK.

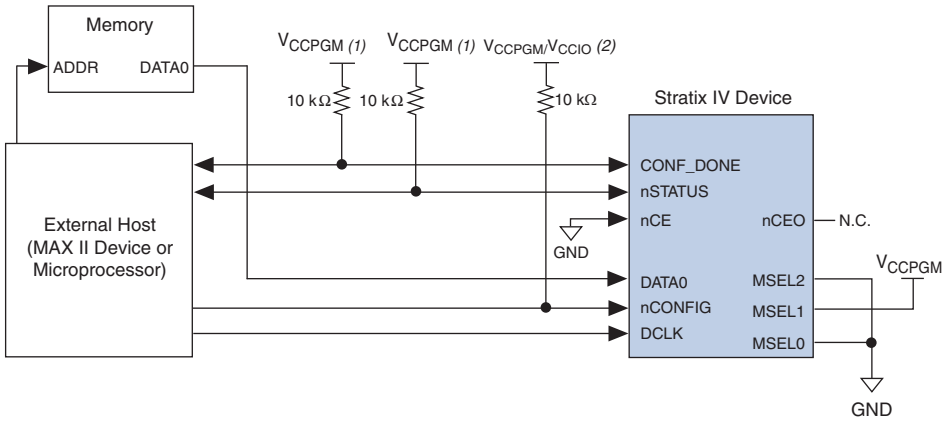
The Stratix IV decompression and design security features are fully available when configuring your Stratix IV device using PS mode.

PS Configuration Using a MAX II Device as an External Host

In this configuration scheme, you can use a MAX II device as an intelligent host that controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device. You can store configuration data in **.rbf**, **.hex**, or **.ttf** format.

Figure 10–10 shows the configuration interface connections between a Stratix IV device and a MAX II device for single device configuration.

Figure 10–10. Single Device PS Configuration Using an External Host



Note to Figure 10–10:

(1) Connect the resistor to a supply that provides an acceptable input signal for the Stratix IV 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) A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host is not driving the line.

After power-up, Stratix IV devices go through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors that are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages—reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration, the MAX II device must generate a low-to-high transition on the nCONFIG pin.

V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} of the banks where the configuration pins reside must be fully powered to the appropriate voltage levels to begin the configuration process.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the MAX II device places the configuration data one bit at a time on the DATA0 pin. 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** contains the byte sequence 02 1B EE 01 FA, the serial bitstream you must transmit to the device is 0100-0000 1101-1000 0111-0111 1000-0000 0101-1111.

A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host (a Max II CPLD or a microcontroller) is not driving the line. For example, during external host reprogramming or power-up where the I/O driving nCONFIG may be tri-stated. If a pull-up resistor is added to the nCONFIG line, the FPGA stays in user mode if the external host is being reprogrammed. If a pull-down resistor is added to the nCONFIG line, the FPGA goes into reset mode if the external host is being reprogrammed. Whenever the nCONFIG line is released high, ensure that the first DCLK and DATA are not driven unintentionally.

The Stratix IV device receives configuration data on the DATA0 pin and the clock is received on the DCLK pin. Data is latched into the device on the rising edge of DCLK. Data is continuously clocked into the target device until CONF_DONE goes high. After the device has received all configuration data successfully, it releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on CONF_DONE indicates configuration is complete and initialization of the device can begin. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

In Stratix IV devices, the initialization clock source is either the internal oscillator or the optional CLKUSR pin. By default, the internal oscillator is the clock source for initialization. If you use the internal oscillator, the Stratix IV device provides itself with enough clock cycles for proper initialization. Therefore, if the internal oscillator is the initialization clock source, sending the entire configuration file to the device is sufficient to configure and initialize the device. Driving DCLK to the device after configuration is complete does not affect device operation.

You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the CLKUSR option. You can turn on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you supply a clock on CLKUSR, it will not affect the configuration process. After all configuration data has been accepted and CONF_DONE goes high, CLKUSR is enabled after the time specified at t_{CD2CU}. After this time period elapses, Stratix IV devices require 8,532 clock cycles to initialize properly and enter user mode. Stratix IV devices support a CLKUSR f_{MAX} of 125 MHz.

An optional INIT_DONE pin is available that signals the end of initialization and the start of user-mode with a low-to-high transition. The **Enable INIT_DONE Output** option is available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box. If you use the INIT_DONE pin, it is high due to an external 10-kΩ pull-up resistor when nCONFIG is low and during the beginning of configuration. 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 is complete, the INIT_DONE pin is released and pulled high. The MAX II device must be able to detect this low-to-high transition that signals the device has entered user mode. When initialization is complete, the device enters user mode. In user-mode, the user I/O pins no longer have weak pull-up resistors and function as assigned in your design.

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

To ensure DCLK and DATA0 are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When you chose the PS scheme as a default in the Quartus II software, this I/O pin is tri-stated in user mode and must be driven by the MAX II device. To change this default option in the Quartus II software, select the **Dual-Purpose Pins** tab of the **Device and Pin Options** dialog box.

The configuration clock (DCLK) speed must be below the specified frequency to ensure correct configuration. No maximum DCLK period exists, which means you can pause the configuration by halting DCLK for an indefinite amount of time.

If an error occurs during configuration, the device drives its nSTATUS pin low, resetting itself internally. The low signal on the nSTATUS pin also alerts the MAX II device that there is an error. If the **Auto-restart configuration after error** option (available in the Quartus II software from the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Stratix IV device releases nSTATUS after a reset time-out period (a maximum of 500 µs). After nSTATUS is released and pulled high by a pull-up resistor, the MAX II device can try to reconfigure the target device without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

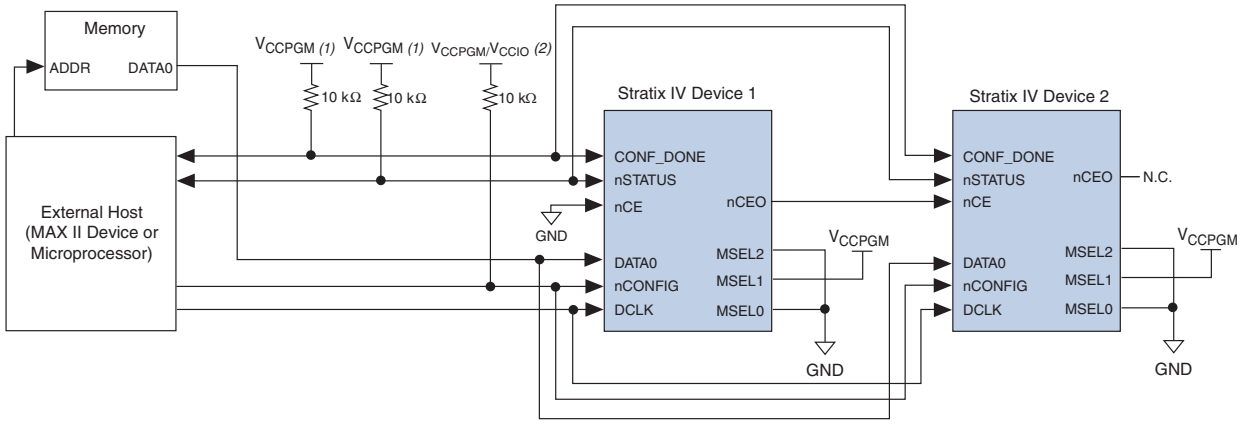
The MAX II device can also monitor the CONF_DONE and INIT_DONE pins to ensure successful configuration. The CONF_DONE pin must be monitored by the MAX II device to detect errors and determine when programming completes. If all configuration data is sent, but CONF_DONE or INIT_DONE have not gone high, the MAX II device must reconfigure the target device.

If you use the optional CLKUSR pin and nCONFIG is pulled low to restart configuration during device initialization, you must ensure that CLKUSR continues toggling during the time nSTATUS is low (a maximum of 500 µs).

When the device is in user-mode, you can initiate a reconfiguration by transitioning the nCONFIG pin low-to-high. The nCONFIG pin must be low for at least 2 µs. When nCONFIG is pulled low, the device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. After nCONFIG returns to a logic high level and nSTATUS is released by the device, reconfiguration begins.

Figure 10–11 shows how to configure multiple devices using a MAX II device. This circuit is similar to the PS configuration circuit for a single device, except the Stratix IV devices are cascaded for multi-device configuration.

Figure 10–11. Multi-Device PS Configuration Using an External Host



Note to Figure 10–11:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. 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) A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host is not driving the line.

In multi-device PS configuration, the first device's nCE pin is connected to GND, while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. 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 within one clock cycle. Therefore, the transfer of data destinations is transparent to the MAX II device. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Because all device CONF_DONE pins are tied together, all devices initialize and enter user mode at the same time.

Because all nSTATUS and CONF_DONE pins are tied together, if any device detects an error, configuration stops for the entire chain and you must reconfigure the entire chain. 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.

If the **Auto-restart configuration after error** option is turned on, the devices release their nSTATUS pins after a reset time-out period (a maximum of 500 µs). After all nSTATUS pins are released and pulled high, the MAX II device can try to reconfigure the chain without needing to pulse nCONFIG low. If this option is turned off, the MAX II device must generate a low-to-high transition (with a low pulse of at least 2 µs) on nCONFIG to restart the configuration process.

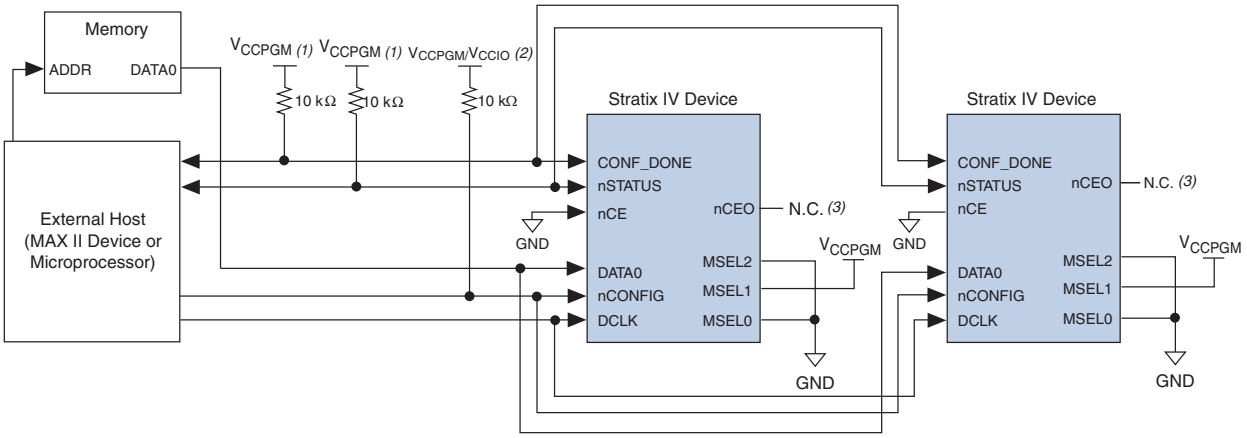
If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the nSTATUS pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

In your system, you can have multiple devices that contain the same configuration data. To support this configuration scheme, all device nCE inputs are tied to GND, while the nCEO pins are left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain.

Configuration signals can require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered for every fourth device. Devices must be the same density and package. All devices start and complete configuration at the same time.

Figure 10–12 shows multi-device PS configuration when both Stratix IV devices are receiving the same configuration data.

Figure 10–12. Multiple-Device PS Configuration When Both Devices Receive the Same Data



Notes to Figure 10–12:

(1) Connect the resistor to a supply that provides an acceptable input signal for all Stratix IV devices in the chain. 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) A pull-up or pull-down resistor helps keep the nCONFIG line in a known state when the external host is not driving the line.

(3) The nCEO pins of both devices are left unconnected when configuring the same configuration data into multiple devices.

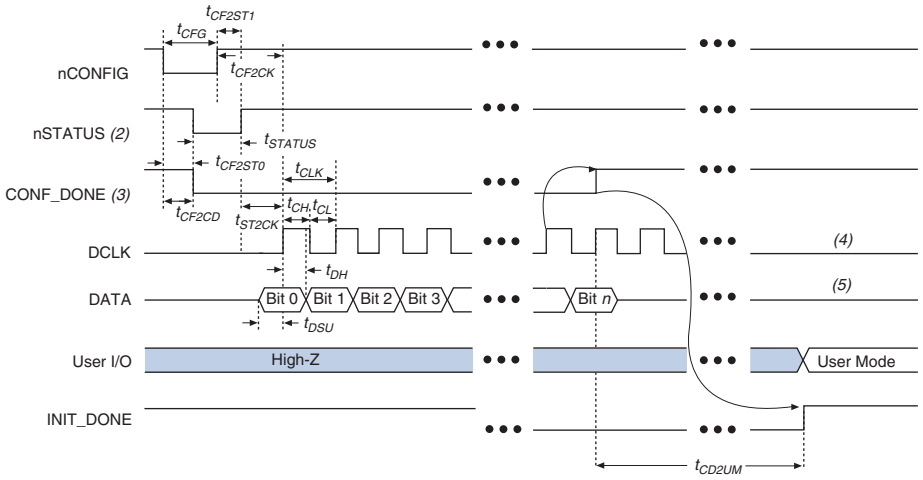
You can use a single configuration chain to configure Stratix IV devices with other Altera devices. To ensure that all devices in the chain complete configuration at the same time, or that an error flagged by one device initiates reconfiguration in all devices, all of the device CONF_DONE and nSTATUS pins must be tied together.

For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

PS Configuration Timing

Figure 10–13 shows the timing waveform for PS configuration when using a MAX II device as an external host.

Figure 10–13. PS Configuration Timing Waveform ***(Note 1)***



Notes to Figure 10–13:

(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 IV 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) DATA[0] is available as a user I/O pin after configuration. The state of this pin depends on the dual-purpose pin settings.

Table 10–7 lists the timing parameters for Stratix IV devices for PS configuration.

Table 10–7. PS Timing Parameters for Stratix IV Devices (Part 1 of 2) ***(Note 1)***

Symbol	Parameter	Minimum	Maximum	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low	—	800	ns
t_{CF2ST0}	nCONFIG low to nSTATUS low	—	800	ns
t_{CFG}	nCONFIG low pulse width	2	—	µs
t_{STATUS}	nSTATUS low pulse width	10	500 *(2)*	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high	—	500 *(3)*	µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	500	—	µs
t_{ST2CK}	nSTATUS high to first rising edge of DCLK	2	—	µs
t_{DSU}	Data setup time before rising edge on DCLK	4	—	ns
t_{DH}	Data hold time after rising edge on DCLK	0	—	ns
t_{CH}	DCLK high time *(5)*	3.2	—	ns
t_{CL}	DCLK low time *(5)*	3.2	—	ns
t_{CLK}	DCLK period *(5)*	8	—	ns
f_{MAX}	DCLK frequency	—	125	MHz
t_R	Input rise time	—	40	ns

Table 10–7. PS Timing Parameters for Stratix IV Devices (Part 2 of 2) *(Note 1)*

Symbol	Parameter	Minimum	Maximum	Units
t_F	Input fall time	—	40	ns
t_{CD2UM}	CONF_DONE high to user mode *(4)*	55	150	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} + (8532 CLKUSR period)	—	—

Notes to Table 10–7:

(1) This information is preliminary.
(2) This value is applicable if you do not delay the configuration by extending the nCONFIG or nSTATUS low pulse width.
(3) This value is applicable if you do not delay the configuration by externally holding nSTATUS low.
(4) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.
(5) Adding up t_{CH} and t_{CL} equals to t_{CLK}. When t_{CH} is 3.2 ns (min), t_{CL} must be 4.8 ns and vice versa.

Device configuration options and how to create configuration files are described in the *Device Configuration Options and Configuration File Formats* chapters in volume 2 of the *Configuration Handbook*.

PS Configuration Using a Microprocessor

In this PS configuration scheme, a microprocessor controls the transfer of configuration data from a storage device, such as flash memory, to the target Stratix IV device.

For more information about configuration and timing information, refer to "PS Configuration Using a MAX II Device as an External Host" on page 10–25. This section is also applicable when using a microprocessor as an external host.

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, MasterBlaster serial/USB communications cable, ByteBlaster II parallel port download cable, ByteBlasterMV parallel port download cable, and EthernetBlaster download cable.

In a PS configuration with a download cable, an intelligent host (such as a PC) transfers data from a storage device to the device using the USB Blaster, MasterBlaster, ByteBlaster II, EthernetBlaster, or ByteBlasterMV cable.

After power-up, Stratix IV devices go through a POR. The POR delay depends on the PORSEL pin setting. When PORSEL is driven low, the standard POR time is 100 ms < T_{POR} < 300 ms. When PORSEL is driven high, the fast POR time is 4 ms < T_{POR} < 12 ms. During POR, the device resets, holds nSTATUS low, and tri-states all user I/O pins. After the device successfully exits POR, all user I/O pins continue to be tri-stated. If nIO_pullup is driven low during power-up and configuration, the user I/O pins and dual-purpose I/O pins will have weak pull-up resistors, which are on (after POR) before and during configuration. If nIO_pullup is driven high, the weak pull-up resistors are disabled.

The configuration cycle consists of three stages—reset, configuration, and initialization. While nCONFIG or nSTATUS are low, the device is in reset. To initiate configuration in this scheme, the download cable generates a low-to-high transition on the nCONFIG pin.

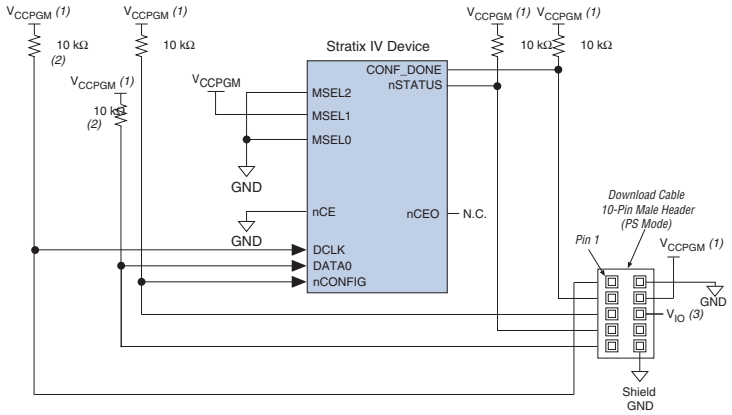
To begin configuration, power the V_{CC}, V_{CCIO}, V_{CCPGM}, and V_{CCPD} voltages (for the banks where the configuration pins reside) to the appropriate voltage levels.

When nCONFIG goes high, the device comes out of reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. The programming hardware or download cable then 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. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize.

When using a download cable, setting the **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart configuration in the Quartus II software when an error occurs. Additionally, the **Enable user-supplied start-up clock** (**CLKUSR**) option has no affect on the device initialization because this option is disabled in the **.sof** when programming the device using the Quartus II programmer and download cable. Therefore, if you turn on the CLKUSR option, you do not need to provide a clock on CLKUSR when you are configuring the device with the Quartus II programmer and a download cable.

Figure 10–14 shows PS configuration for Stratix IV devices using a USB Blaster, EthernetBlaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV cable.

Figure 10–14. PS Configuration Using a USB Blaster, EthernetBlaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV Cable



Notes to Figure 10–14:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCPGM}) as the USB Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, 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.

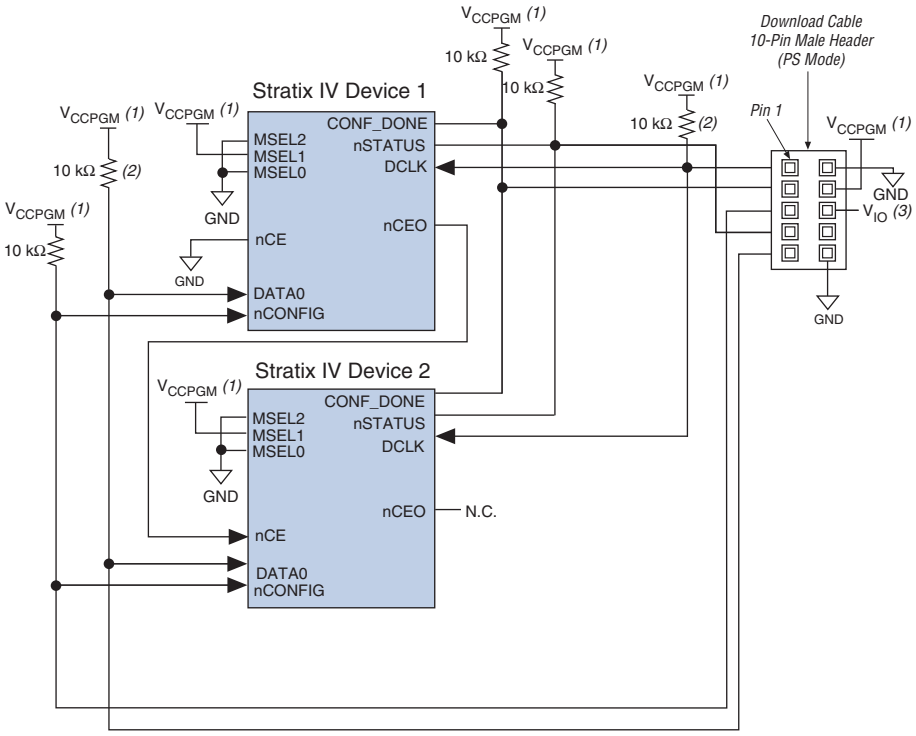
(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPGM}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cable, this pin is a no connect.

You can use a download cable to configure multiple Stratix IV devices by connecting each device's nCEO pin to the subsequent device's nCE pin. The first device's nCE pin is connected to GND, while its nCEO pin is connected to the nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. All other configuration pins (nCONFIG, nSTATUS, DCLK, DATA0, and CONF_DONE) are connected to every device in the chain. Because all CONF_DONE pins are tied together, all devices in the chain initialize and enter user mode at the same time.

In addition, because the nSTATUS pins are tied together, the entire chain halts configuration if any device detects an error. The **Auto-restart configuration after error** option does not affect the configuration cycle because you must manually restart the configuration in the Quartus II software when an error occurs.

Figure 10–15 shows how to configure multiple Stratix IV devices with a download cable.

Figure 10–15. Multi-Device PS Configuration Using a USB Blaster, EthernetBlaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV Cable



Notes to Figure 10–15:

(1) Connect the pull-up resistor to the same supply voltage (V_{CCPGM}) as the USB Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, 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 is to ensure 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) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPGM}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cables, this pin is a no connect.

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

- USB-Blaster Download Cable User Guide
- MasterBlaster Serial/USB Communications Cable User Guide
- ByteBlaster II Download Cable User Guide
- ByteBlasterMV Download Cable User Guide
- EthernetBlaster Communications Cable User Guide

JTAG Configuration

JTAG has developed a specification for boundary-scan testing. This boundary-scan test (BST) architecture offers the capability to efficiently test components on PCBs with tight lead spacing. The BST architecture can test pin connections without using physical test probes and capture functional data while a device is operating normally. You can also use JTAG circuitry to shift configuration data into the device. The Quartus II software automatically generates **.sof**s that you can use for JTAG configuration with a download cable in the Quartus II software programmer.

For more information about JTAG boundary-scan testing and commands available using Stratix IV devices, refer to the following documents:

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

Stratix IV devices are designed such that JTAG instructions have precedence over any device configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration of Stratix IV devices during PS configuration, PS configuration is terminated and JTAG configuration begins.

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

A device operating in JTAG mode uses four required pins, `TDI`, `TDO`, `TMS`, and `TCK`, and one optional pin, `TRST`. The `TCK` pin has an internal weak pull-down resistor, while the `TDI`, `TMS`, and `TRST` pins have weak internal pull-up resistors (typically 25 kΩ). The JTAG output pin `TDO` and all JTAG input pins are powered by 2.5-V/3.0-V V_{CCPD}. All the JTAG pins only support the LVTTL I/O standard.

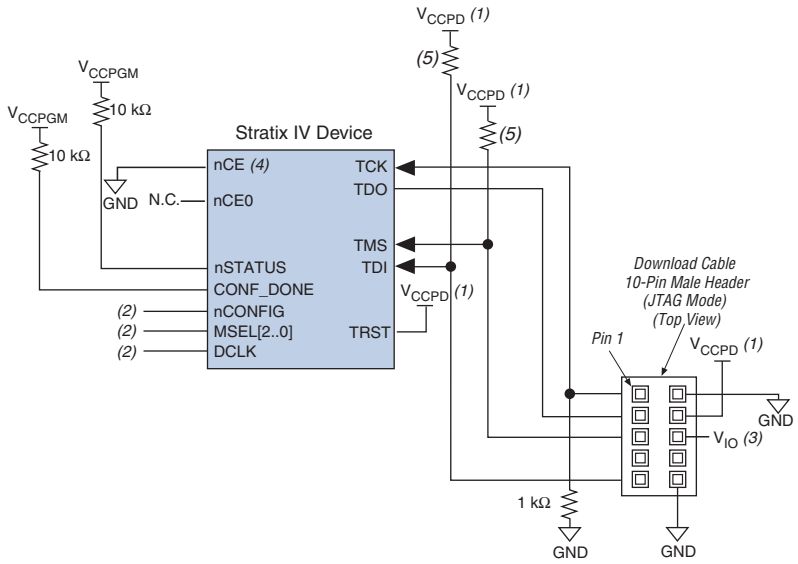
All user I/O pins are tri-stated during JTAG configuration.

All the JTAG pins are powered by the V_{CCPD} power supply of I/O bank 1A. For more information about how to connect a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing in Stratix IV Devices* chapter.

During JTAG configuration, you can download data to the device on the PCB through the USB Blaster, MasterBlaster, ByteBlaster II, EthernetBlaster, or ByteBlasterMV download cable. Configuring devices through a cable is similar to programming devices in-system, except you must connect the TRST pin to V_{CCPD}. This ensures that the TAP controller is not reset.

Figure 10–16 shows JTAG configuration of a single Stratix IV device when using a download cable.

Figure 10–16. JTAG Configuration of a Single Device Using a Download Cable



Notes to Figure 10–16:

(1) Connect the pull-up resistor to the same supply voltage as the USB Blaster, MasterBlaster (VIO pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable. The voltage supply can be connected to the V_{CCPD} of the device.

(2) Connect the nCONFIG and MSEL[2..0] pins to support a non-JTAG configuration scheme. If you only use the JTAG configuration, connect nCONFIG to V_{CCPGM} and MSEL[2..0] to GND. Pull DCLK either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPD}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cable, this pin is a no connect.

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

(5) The pull-up resistor value can vary from 1 k to 10 kΩ.

To configure a single device in a JTAG chain, the programming software places all 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 upon completion. At the end of configuration, the software checks the state of CONF_DONE through the JTAG port. When the Quartus II software generates a JAM file (**.jam)** for a multi-device chain, it contains instructions so that all the devices in the chain are initialized at the same time. 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 bitstream is transmitted serially using the JTAG TDI port, the TCK port is clocked an additional 1,094 cycles to perform device initialization.

Stratix IV devices have dedicated JTAG pins that always function as JTAG pins. Not only can you perform JTAG testing on Stratix IV devices before and after, but also during configuration. While other device families do not support JTAG testing during configuration, Stratix IV devices support the bypass, ID code, and sample instructions during configuration without interrupting configuration. All other JTAG instructions may only be issued by first interrupting configuration and reprogramming the I/O pins using the CONFIG_IO instruction.

The CONFIG_IO instruction allows I/O buffers to be configured using the JTAG port and when issued, interrupts configuration. This instruction allows you to perform board-level testing prior to configuring the Stratix IV device or waiting for a configuration device to complete configuration. After configuration has been interrupted and JTAG testing is complete, you must reconfigure the part using JTAG (PULSE_CONFIG instruction) or by pulsing nCONFIG low.

The chip-wide reset (DEV_CLRn) and chip-wide output enable (DEV_OE) pins on Stratix IV devices do not affect JTAG boundary-scan or programming operations. Toggling these pins does not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration for Stratix IV devices, consider the dedicated configuration pins. Table 10–8 lists how these pins are connected during JTAG configuration.

Table 10–8. Dedicated Configuration Pin Connections During JTAG Configuration (Part 1 of 2)

Signal	Description
nCE	On all Stratix IV devices in the chain, nCE must be driven low by connecting it to GND, pulling it low using a resistor, or driving it by some control circuitry. For devices that are also in multi-device FPP, AS, or PS configuration chains, the nCE pins must be connected to GND during JTAG configuration or JTAG must be configured in the same order as the configuration chain.
nCEO	On all Stratix IV devices in the chain, you can leave nCEO floating or connected to the nCE of the next device.
MSEL	Do not leave these pins floating. These pins support whichever non-JTAG configuration is used in production. If you only use JTAG configuration, tie these pins to GND.

Table 10–8. Dedicated Configuration Pin Connections During JTAG Configuration (Part 2 of 2)

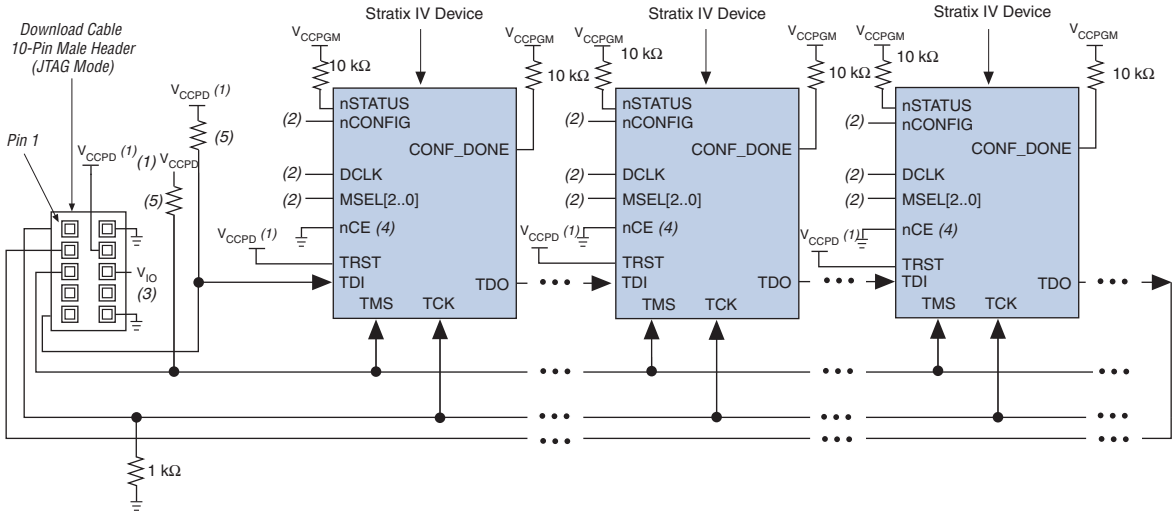
Signal	Description
nCONFIG	Driven high by connecting to V_{CCPGM}, pulling up using a resistor, or driven high by some control circuitry.
nSTATUS	Pull to V_{CCPGM} using a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin must be pulled up to V_{CCPGM} individually.
CONF_DONE	Pull to V_{CCPGM} using a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin must be pulled up to V_{CCPGM} individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration.
DCLK	Do not leave DCLK floating. Drive low or high, whichever is more convenient on your board.

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.

JTAG-chain device programming is ideal when the system contains multiple devices, or when testing your system using JTAG BST circuitry.

Figure 10–17 shows a multi-device JTAG configuration when using a download cable.

Figure 10–17. JTAG Configuration of Multiple Devices Using a Download Cable



Notes to Figure 10–17:

(1) Connect the pull-up resistor to the same supply voltage as the USB Blaster, MasterBlaster (V_{IO} pin), ByteBlaster II, ByteBlasterMV, or EthernetBlaster cable. Connect the voltage supply to V_{CCPD} of the device.

(2) Connect the nCONFIG and MSEL[2..0] pins to support a non-JTAG configuration scheme. If you only use a JTAG configuration, connect nCONFIG to V_{CCPGM} and MSEL[2..0] to GND. Pull DCLK either high or low, whichever is convenient on your board.

(3) Pin 6 of the header is a V_{IO} reference voltage for the MasterBlaster output driver. V_{IO} must match the device's V_{CCPD}. For more information about this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the USB-Blaster, ByteBlaster II, and ByteBlasterMV cables, this pin is a no connect.

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

(5) The pull-up resistor value can vary from 1 k to 10 kΩ.

You must connect the nCE pin to GND or drive it low during JTAG configuration. In multi-device FPP, AS, and PS configuration chains, the first device's nCE pin is connected to GND, while its nCEO pin is connected to nCE of the next device in the chain. The last device's nCE input comes from the previous device, while its nCEO pin is left floating. In addition, the CONF_DONE and nSTATUS signals are all shared in multi-device FPP, AS, or PS configuration chains so the devices can enter user mode at the same time after configuration is complete. When the CONF_DONE and nSTATUS signals are shared among all the devices, you must configure every device when JTAG configuration is performed.

If you only use JTAG configuration, Altera recommends connecting the circuitry as shown in Figure 10–17, where each of the CONF_DONE and nSTATUS signals are isolated, so that each device can enter user mode individually.

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. Therefore, if these devices are also in a JTAG chain, ensure the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. As long as the devices are JTAG configured in the same order as the multi-device configuration chain, the nCEO of the previous device drives the nCE of the next device low when it has successfully been JTAG configured.

You can place other Altera devices that have JTAG support in the same JTAG chain for device programming and configuration.

JTAG configuration support is enhanced and allows more than 17 Stratix IV devices to be cascaded in a JTAG chain.

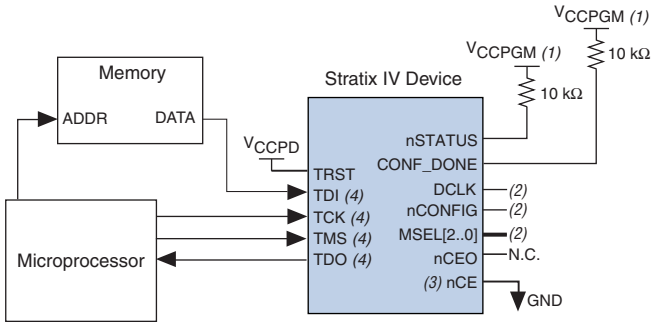
For more information about configuring multiple Altera devices in the same configuration chain, refer to the *Configuring Mixed Altera FPGA Chains* chapter in volume 2 of the *Configuration Handbook*.

You can configure Stratix IV devices using multiple configuration schemes on the same board. Combining JTAG configuration with AS configuration on your board is useful in the prototyping environment because it allows multiple methods to configure your FPGA.

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

Figure 10–18 shows JTAG configuration of a Stratix IV device using a microprocessor.

Figure 10–18. JTAG Configuration of a Single Device Using a Microprocessor



Notes to Figure 10–18:

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

(2) Connect the nCONFIG and MSEL[2..0] pins to support a non-JTAG configuration scheme. If you use only a JTAG configuration, connect nCONFIG to V_{CCGPM} and MSEL[2..0] to GND. Pull DCLK either high or low, whichever is convenient on your board.

(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.

Jam STAPL

Jam™ STAPL, JEDEC standard JESD-71, is a standard file format for in-system programmability (ISP) purposes. Jam STAPL supports programming or configuration of programmable devices and testing of electronic systems, using the IEEE 1149.1 JTAG interface. Jam STAPL is a freely licensed open standard.

The Jam Player provides an interface for manipulating the IEEE Std. 1149.1 JTAG TAP state machine.

For more information about JTAG and Jam STAPL in embedded environments, refer to *Using Jam STAPL for ISP via an Embedded Processor*. To download the Jam Player, visit the Altera website at www.altera.com.

Device Configuration Pins

The following tables list the connections and functionality of all the configuration-related pins on Stratix IV devices. Table 10–9 lists the Stratix IV configuration pins and their power supply.

Table 10–9. Stratix IV Configuration Pin Summary (Part 1 of 2) *(Note 1)*

Description	Input/Output	Dedicated	Powered By	Configuration Mode
TDI	Input	Yes	V_{CCPD}	JTAG
TMS	Input	Yes	V_{CCPD}	JTAG
TCK	Input	Yes	V_{CCPD}	JTAG
TRST	Input	Yes	V_{CCPD}	JTAG
TDO	Output	Yes	V_{CCPD}	JTAG
CRC_ERROR	Output	—	Pull-up	Optional, all modes

Table 10–9. Stratix IV Configuration Pin Summary (Part 2 of 2) ***(Note 1)***

Description	Input/Output	Dedicated	Powered By	Configuration Mode
DATA0	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	All modes except JTAG
DATA[7..1]	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	FPP
INIT_DONE	Output	—	Pull-up	Optional, all modes
CLKUSR	Input	—	V_{CCPGM}/V_{CCIO} *(3)*	Optional
nSTATUS	Bidirectional	Yes	V_{CCPGM}/Pull-up	All modes
nCE	Input	Yes	V_{CCPGM}	All modes
CONF_DONE	Bidirectional	Yes	V_{CCPGM}/Pull-up	All modes
nCONFIG	Input	Yes	V_{CCPGM}	All modes
PORSEL	Input	Yes	V_{CC} *(2)*	All modes
ASDO *(4)*	Output	Yes	V_{CCPGM}	AS
nCSO *(4)*	Output	Yes	V_{CCPGM}	AS
DCLK *(4)*	Input	Yes	V_{CCPGM}	PS, FPP
	Output	Yes	V_{CCPGM}	AS
nIO_PULLUP	Input	Yes	V_{CC} *(2)*	All modes
nCEO	Output	Yes	V_{CCPGM}	All modes
MSEL[2..0]	Input	Yes	V_{CC} *(2)*	All modes

Notes to Table 10–9:

(1) The total number of pins is 29. The total number of dedicated pins is 18.

(2) Although MSEL[2..0], PORSEL, and nIO_PULLUP are powered up by V_{CC}, Altera recommends connecting these pins to V_{CCPGM} or GND directly without using a pull-up or pull-down resistor.

(3) These pins are powered up by V_{CCPGM} during configuration. These pins are powered up by V_{CCIO} if they are used as regular I/O in user mode.

(4) To tri-state this pin, in the Quartus II software, on the Assignments menu, select **Device**. On the **Device** page, select **Device and Pin Options...** On the **Device and Pin Options** page, select **Configuration** and select the **Enable input tri-state on active configuration pins in user mode** option.

Table 10–10 lists the dedicated configuration pins. You must connect these pins properly on your board for successful configuration. Some of these pins may not be required for your configuration schemes.

Table 10–10. Dedicated Configuration Pins on the Stratix IV Device (Part 1 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
VCCPGM	N/A	All	Power	Dedicated power pin. Use this pin to power all dedicated configuration inputs, dedicated configuration outputs, dedicated configuration bidirectional pins, and some of the dual functional pins that are used for configuration. You must connect this pin to 1.8, 2.5, or 3.0 V. V_{CCPGM} must ramp-up from 0 V to V_{CCPGM} within 100 ms when PORSEL is low or 4 ms when PORSEL is high. If V_{CCPGM} is not ramped up within this specified time, your Stratix IV device will not configure successfully. If your system does not allow a V_{CCPGM} ramp-up within 100 ms or 4 ms, you must hold nCONFIG low until all power supplies are stable.
VCCPD	N/A	All	Power	Dedicated power pin. Use this pin to power the I/O pre-drivers, JTAG input and output pins, and design security circuitry. You must connect this pin to 2.5 V or 3.0 V, depending on the I/O standards selected. For the 3.0-V I/O standard, V_{CCPD} = 3.0 V. For the 2.5 V or below I/O standards, V_{CCPD} = 2.5 V. V_{CCPD} must ramp-up from 0 V to 2.5 V / 3.0 V within 100 ms when PORSEL is low or 4 ms when PORSEL is high. If V_{CCPD} is not ramped up within this specified time, your Stratix IV device will not configure successfully. If your system does not allow a V_{CCPD} to ramp-up time within 100 ms or 4 ms, you must hold nCONFIG low until all power supplies are stable.
PORSEL	N/A	All	Input	Dedicated input that selects between a standard POR time or a fast POR time. A logic low selects a standard POR time setting of 100 ms < T_{POR} < 300 ms and a logic high selects a fast POR time setting of 4 ms < T_{POR} < 12 ms. The PORSEL input buffer is powered by V_{CC} and has an internal 5-kΩ pull-down resistor that is always active. Tie the PORSEL pin directly to V_{CCPGM} or GND.
nIO_PULLUP	N/A	All	Input	Dedicated input that chooses whether the internal pull-up resistors on the user I/O pins and dual-purpose I/O pins (nCSO, nASDO, DATA[7..0], CLKUSR, and INIT_DONE) are on or off before and during configuration. A logic high turns off the weak internal pull-up resistors; a logic low turns them on. The nIO-PULLUP input buffer is powered by V_{CC} and has an internal 5-kΩ pull-down resistor that is always active. The nIO-PULLUP can be tied directly to V_{CCPGM}, using a 1-kΩ pull-up resistor or tied directly to GND, depending on your device requirements.

Table 10–10. Dedicated Configuration Pins on the Stratix IV Device (Part 2 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
MSEL[2..0]	N/A	All	Input	Three-bit configuration input that sets the Stratix IV device configuration scheme. For the appropriate connections, refer to Table 10–1 on page 10–2. You must hardwire these pins to V_{CCPGM} or GND. The MSEL[2..0] pins have internal 5-kΩ pull-down resistors that are always active.
nCONFIG	N/A	All	Input	Configuration control input. Pulling this pin low during user-mode causes the device to lose its configuration data, enter a reset state, and tri-state all I/O pins. Returning this pin to a logic high level initiates a reconfiguration. Configuration is possible only if this pin is high, except in JTAG programming mode, when nCONFIG is ignored.
nSTATUS	N/A	All	Bidirectional open-drain	The device drives nSTATUS low immediately after power-up and releases it after the POR time. During user mode and regular configuration, this pin is pulled high by an external 10-kΩ resistor. This pin, when driven low by the Stratix IV device, indicates that the device has encountered an error during configuration. ■ Status output—If an error occurs during configuration, nSTATUS is pulled low by the target device. ■ Status input—If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. Driving nSTATUS low after configuration and initialization does not affect the configured device. If you use a configuration device, driving nSTATUS low causes the configuration device to attempt to configure the device, but because the device ignores transitions on nSTATUS in user mode, the device does not reconfigure. To initiate a reconfiguration, nCONFIG must be pulled low. If you have enabled the **Auto-restart configuration after error** option, the nSTATUS pin transitions from high to low and back again to high when a configuration error is detected. This appears as a low pulse at the pin with a minimum pulse width of 10 µs to a maximum pulse width of 500 µs, as defined in the t_{STATUS} specification.

Table 10–10. Dedicated Configuration Pins on the Stratix IV Device (Part 3 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nSTATUS (continued)	—	—	—	If V_{CCPGM} is not fully powered up, the following could occur: ■ V_{CCPGM} is powered high enough for the nSTATUS buffer to function properly and nSTATUS is driven low. When V_{CCPGM} is ramped up, POR trips and nSTATUS is released after POR expires. ■ V_{CCPGM} is not powered high enough for the nSTATUS buffer to function properly. In this situation, nSTATUS might appear logic high, triggering a configuration attempt that would fail because POR did not yet trip. When V_{CCPD} is powered up, nSTATUS is pulled low because POR did not yet trip. When POR trips after V_{CCPGM} is powered up, nSTATUS is released and pulled high. At that point, reconfiguration is triggered and the device is configured.
CONF_DONE	N/A	All	Bidirectional open-drain	Status output. 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 CONF_DONE. Status input. After all the data is received and CONF_DONE goes high, the target device initializes and enters user mode. The CONF_DONE pin must have an external 10-kΩ pull-up resistor for the device to initialize. Driving CONF_DONE low after configuration and initialization does not affect the configured device.
nCE	N/A	All	Input	Active-low chip enable. The nCE pin activates the device with a low signal to allow configuration. The nCE pin must be held low during configuration, initialization, and user mode. In single device configuration, it must be tied low. In multi-device configuration, nCE of the first device is tied low, while its nCEO pin is connected to nCE of the next device in the chain. The nCE pin must also be held low for successful JTAG programming of the device.
nCEO	N/A	All	Output	Output that drives low when device configuration is complete. In single device configuration, this pin is left floating. In multi-device configuration, this pin feeds the next device's nCE pin. The nCEO of the last device in the chain is left floating. The nCEO pin is powered by V_{CCPGM}.
ASDO	N/A	AS	Output	Control signal from the Stratix IV device to the serial configuration device in AS mode used to read out configuration data. In AS mode, ASDO has an internal pull-up resistor that is always active.

Table 10–10. Dedicated Configuration Pins on the Stratix IV Device (Part 4 of 4)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nCSO	N/A	AS	Output	Output control signal from the Stratix IV device to the serial configuration device in AS mode that enables the configuration device. In AS mode, nCSO has an internal pull-up resistor that is always active.
DCLK	N/A	Synchronous configuration schemes (PS, FPP, AS)	Input (PS, FPP) Output (AS)	In 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. In AS mode, DCLK is an output from the Stratix IV device that provides timing for the configuration interface. In AS mode, DCLK has an internal pull-up resistor (typically 25 kΩ) that is always active. In AS configuration schemes, this pin is driven into an inactive state after configuration completes. You can use this pin as a user I/O during user mode. In PS or FPP schemes that use a control host, you must drive DCLK either high or low, whichever is more convenient. In passive schemes, you cannot use DCLK as a user I/O during user mode. Toggling this pin after configuration does not affect the configured device.
DATA0	N/A in AS mode. I/O in PS or FPP mode.	PS, FPP, AS	Input	Data input. In serial configuration modes, bit-wide configuration data is presented to the target device on the DATA0 pin. In AS mode, DATA0 has an internal pull-up resistor that is always active. After PS or FPP configuration, DATA0 is available as a user I/O pin. The state of this pin depends on the **Dual-Purpose Pin** settings.
DATA[7..1]	I/O	Parallel configuration schemes (FPP)	Inputs	Data inputs. Byte-wide configuration data is presented to the target device on DATA[7..0]. In serial configuration schemes, they function as user I/O pins during configuration, which means they are tri-stated. After FPP configuration, DATA[7..1] are available as user I/O pins. The state of these pins depends on the **Dual-Purpose Pin** settings.

Table 10–11 lists the optional configuration pins. If these optional configuration pins are not enabled in the Quartus II software, they are available as general-purpose user I/O pins. Therefore, during configuration, these pins function as user I/O pins and are tri-stated with weak pull-up resistors.

Table 10–11. Optional Configuration Pins

Pin Name	User Mode	Pin Type	Description
CLKUSR	N/A if option is on. I/O if option is off.	Input	Optional user-supplied clock input synchronizes the initialization of one or more devices. Enable this pin by turning on the **Enable user-supplied start-up clock** (**CLKUSR**) option in the Quartus II software.
INIT_DONE	N/A if option is on. I/O if option is off.	Output open-drain	Use as a status pin to indicate when the device has initialized and is in user mode. When nCONFIG is low and during the beginning of configuration, the INIT_DONE pin is tri-stated and pulled high due to an external 10-kΩ 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 is complete, 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.
DEV_OE	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all tri-states on the device. When this pin is driven low, all I/O pins are tri-stated. When this pin is driven high, all 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	N/A if option is on. I/O if option is off.	Input	Optional pin that allows you to override all clears on all device registers. When this pin is driven low, all registers are cleared. When this pin is driven high, all registers behave as programmed. Enable this pin by turning on the **Enable device-wide reset** (**DEV_CLRn**) option in the Quartus II software.

Table 10–12 lists the dedicated JTAG pins. JTAG pins must be kept stable before and during configuration to prevent accidental loading of JTAG instructions. The TDI, TMS, and TRST pins have weak internal pull-up resistors, while TCK has a weak internal pull-down resistor (typically 25 kΩ). If you plan to use the SignalTap® embedded logic array analyzer, you must connect the JTAG pins of the Stratix IV device to a JTAG header on your board.

Table 10–12. Dedicated JTAG Pins

Pin Name	User Mode	Pin Type	Description
TDI	N/A	Test data input	Serial input pin for instructions as well as test and programming data. Data is shifted on the rising edge of TCK. The TDI pin is powered by the 2.5-V/3.0-V V_{CCPD} supply. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting this pin to logic high using a 1-kΩ resistor.
TDO	N/A	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. The pin is tri-stated if data is not being shifted out of the device. The TDO pin is powered by V_{CCPD}. For recommendations about connecting a JTAG chain with multiple voltages across the devices in the chain, refer to the *JTAG Boundary Scan Testing in Stratix IV Devices* chapter. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by leaving this pin unconnected.
TMS	N/A	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 within the state machine occur on the falling edge of TCK after the signal is applied to TMS. The TMS pin is powered by 2.5-V/3.0-V V_{CCPD}. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting this pin to logic high using a 1-kΩ resistor.
TCK	N/A	Test clock input	Clock input to the BST circuitry. Some operations occur at the rising edge, while others occur at the falling edge. The TCK pin is powered by the 2.5-V/3.0-V V_{CCPD} supply. It is expected that the clock input waveform have a nominal 50% duty cycle. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting TCK to GND.
TRST	N/A	Test reset input (optional)	Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to IEEE Std. 1149.1. The TRST pin is powered by the 2.5-V/3.0-V V_{CCPD} supply. Hold TMS at 1 or keep TCK static while TRST is changed from 0 to 1. If the JTAG interface is not required on your board, you can disable the JTAG circuitry by connecting the TRST pin to GND.

For more information about the pin connection recommendations, refer to the *Stratix IV GX and Stratix IV E Device Family Pin Connection Guidelines*.

Configuration Data Decompression

Stratix IV devices support configuration data decompression, which saves configuration memory space and time. This feature allows you to store compressed configuration data in configuration devices or other memory and transmit this compressed bitstream to Stratix IV devices. During configuration, the Stratix IV device decompresses the bitstream in real time and programs its SRAM cells.

Preliminary data indicates that compression typically reduces the configuration bitstream size by 30% to 55% based on the designs used.

Stratix IV devices support decompression in the FPP (when using a MAX II device or microprocessor + flash), fast AS, and PS configuration schemes. The Stratix IV decompression feature is not available in the JTAG configuration scheme.

In PS mode, use the Stratix IV decompression feature because sending compressed configuration data reduces configuration time.

When you enable compression, the Quartus II software generates configuration files with compressed configuration data. This compressed file reduces the storage requirements in the configuration device or flash memory, and decreases the time needed to transmit the bitstream to the Stratix IV device. The time required by a Stratix IV device to decompress a configuration file is less than the time needed to transmit the configuration data to the device.

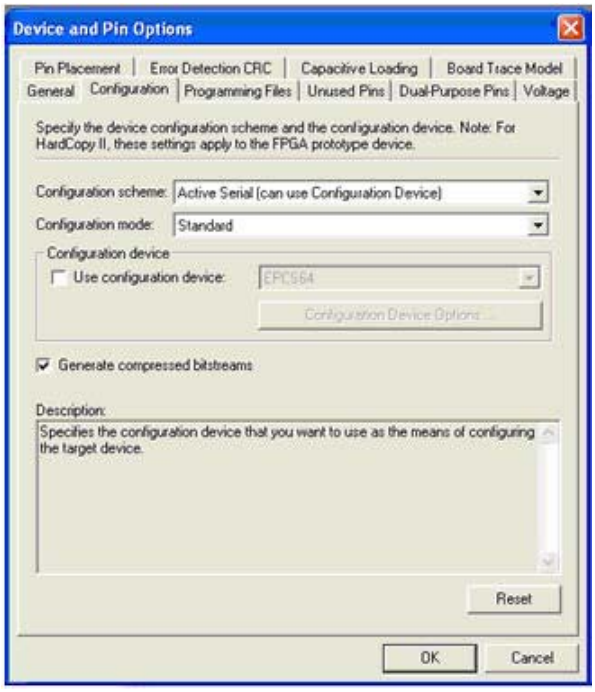
There are two ways to enable compression for Stratix IV bitstreams—before design compilation (in the Compiler Settings menu) and after design compilation (in the **Convert Programming Files** window).

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 IV device, open the **Device and Pin Options** window.

3. In the **Configuration** settings tab, turn on **Generate compressed bitstreams** (as shown in Figure 10–19).

Figure 10–19. Enabling Compression for Stratix IV Bitstreams in Compiler Settings



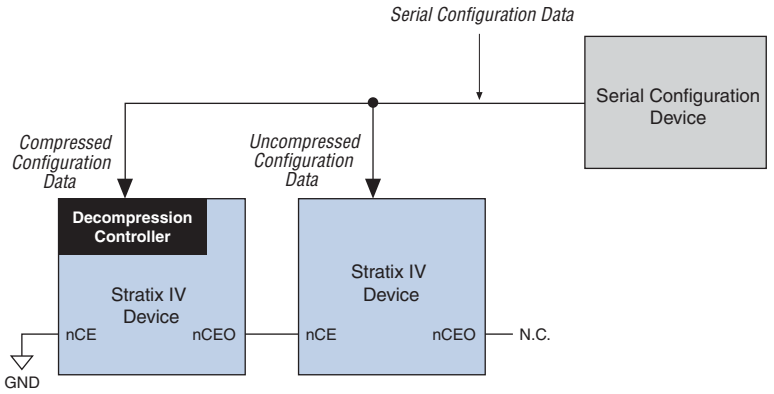
You can also enable compression when creating programming files from the **Convert Programming Files** window. To do this, follow these steps:

1. On the File menu, click **Convert Programming Files**.
2. Select the programming file type (**.pof**, **.sram**, **.hex**, **.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 IV device **.sof** file.
6. Select the name of the file you added to the **SOF Data** area and click **Properties**.
7. Check the **Compression** check box.

When multiple Stratix IV devices are cascaded, you can selectively enable the compression feature for each device in the chain if you are using a serial configuration scheme. Figure 10–20 shows a chain of two Stratix IV devices. The first Stratix IV device has compression enabled; therefore, receives a compressed bitstream from the configuration device. The second Stratix IV device has the compression feature disabled and receives uncompressed data.

In a multi-device FPP configuration chain (with a MAX II device or microprocessor + flash), all Stratix IV devices in the chain must either enable or disable the decompression feature. You cannot selectively enable the compression feature for each device in the chain because of the DATA and DCLK relationship.

Figure 10–20. Compressed and Uncompressed Configuration Data in the Same Configuration File



You can generate programming files for this setup by clicking **Convert Programming Files** on the File 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 upgrade, 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 IV 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 avoid system downtime.

Stratix IV devices feature dedicated remote system upgrade circuitry. Soft logic (either the Nios® II embedded processor or user logic) implemented in a Stratix IV 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 initiate 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 upgrade is supported in fast AS Stratix IV configuration schemes. You can also implement remote system upgrade in conjunction with advanced Stratix IV features such as real-time decompression of configuration data and design security using the advanced encryption standard (AES) for secure and efficient field upgrades. The largest serial configuration device currently supports 128 Mbits of configuration bitstream.

☞ Stratix IV devices only support remote system upgrade in the single device fast AS configuration scheme. Because the largest serial configuration device currently supports 128 Mbits of configuration bitstream, the remote system upgrade feature is not supported in EP4SGX290, EP4SE360, and larger devices.

☞ The remote system upgrade feature is not supported in a multi-device chain.

Functional Description

The dedicated remote system upgrade circuitry in Stratix IV devices manages remote configuration and provides error detection, recovery, and status information. User logic or a Nios II processor implemented in the Stratix IV device logic array provides access to the remote configuration data source and an interface to the system's configuration memory.

Stratix IV devices have remote system upgrade processes that involve the following steps:

1. A Nios II processor (or user logic) implemented in the Stratix IV device logic array receives new configuration data from a remote location. The connection to the remote source uses a communication protocol such as the transmission control protocol/Internet protocol (TCP/IP), peripheral component interconnect (PCI), user datagram protocol (UDP), universal asynchronous receiver/transmitter (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) initiates a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any error(s) that might occur during or after the reconfiguration cycle and provides error status information to the user design.

Figure 10–21 shows the steps required for performing remote configuration updates. (The numbers in Figure 10–21 coincide with the steps just mentioned.)

Figure 10–21. Functional Diagram of Stratix IV Remote System Upgrade

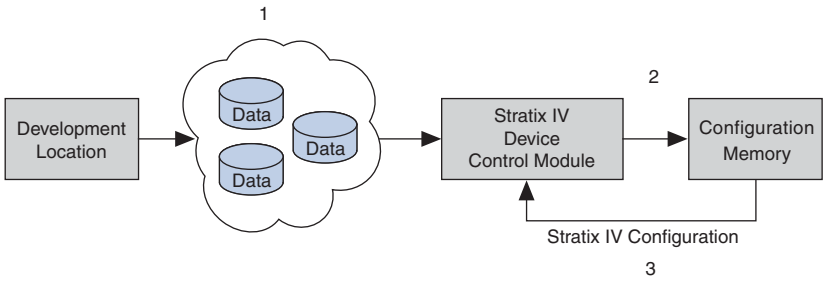
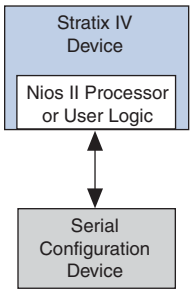


Figure 10–22 shows a block diagram for implementing a remote system upgrade with the Stratix IV fast AS configuration scheme.

Figure 10–22. Remote System Upgrade Block Diagram for Stratix IV Fast AS Configuration Scheme



You must set the mode select pins (MSEL[2..0]) to fast AS mode to use remote system upgrade in your system. Table 10–13 lists the MSEL pin settings for Stratix IV devices in standard configuration mode and remote system upgrade mode. The following sections describe remote update of the remote system upgrade mode.

For more information about standard configuration schemes supported in Stratix IV devices, refer to "Configuration Schemes" on page 10–2.

Table 10–13. Remote System Upgrade Modes in Stratix IV Devices

Configuration Scheme	MSEL[2..0]	Remote System Upgrade Mode
Fast AS (40 MHz)	011	Standard
	011	Remote update *(1)*

Note to Table 10–13:

(1) All EPCS densities are able to support DCLK up to 40 MHz, but batches of EPCS1 and EPCS4 manufactured on 0.18-µm process geometry can only support DCLK up to 20 MHz. For more information, refer to the *Serial Configuration Devices (EPCS1, EPCS4, EPCS16, EPCS64, and EPCS128) Data Sheet* chapter in volume 2 of the *Configuration Handbook*.

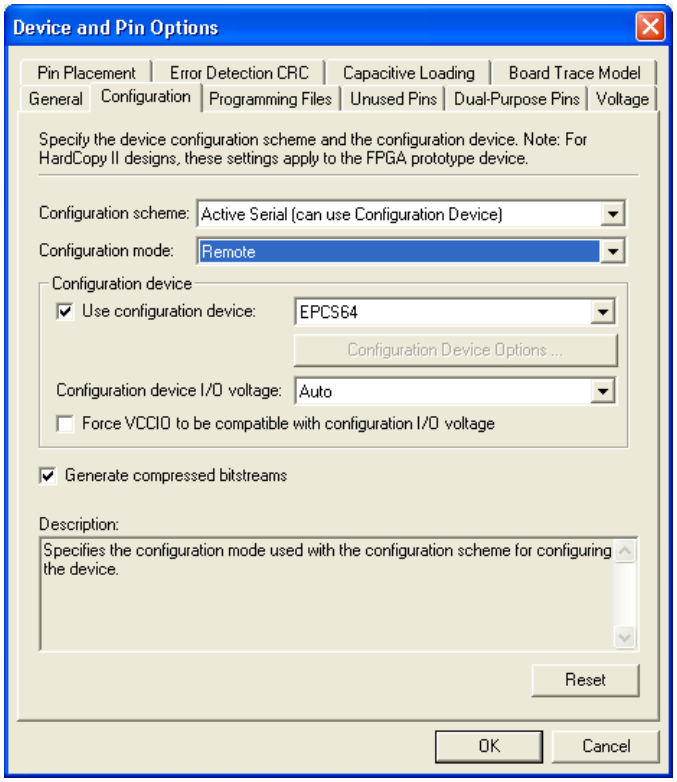
When using fast AS mode, you must select remote update mode in the Quartus II software and insert the ALTREMOTE_UPDATE megafunction to access the circuitry. For more information, refer to "ALTREMOTE_UPDATE Megafunction" on page 10–62.

Enabling Remote Update

You can enable remote update for Stratix IV 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 Assignment 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** tab.
4. From the **Configuration scheme** list, select **Active Serial** (you can also use **Configuration Device**) (Figure 10–23).
5. From the **Configuration Mode** list, select **Remote** (Figure 10–23).
6. Click **OK**.
7. In the **Settings** dialog box, click **OK**.

Figure 10–23. Enabling Remote Update for Stratix IV Devices in the Compiler Settings Menu



Configuration Image Types

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

Each Stratix IV device in your system requires one factory image or the addition of one or more application images. The factory image is a user-defined fall-back, or safe configuration, and is responsible for administering remote updates in conjunction with the dedicated circuitry. Application images implement user-defined functionality in the target Stratix IV device. You may include the default application image functionality in the factory image.

A remote system upgrade involves storing a new application configuration image or updating an existing one using the remote communication interface. After an application configuration image is stored or updated remotely, the user design in the Stratix IV device initiates a reconfiguration cycle with the new image. Any errors during or after this cycle are detected by the dedicated remote system upgrade circuitry and cause the device to automatically revert to the factory image. The factory image then performs error processing and recovery. The factory configuration is written to the serial configuration device only once by the system manufacturer and must not be remotely updated. On the other hand, application configurations may be remotely updated in the system. Both images can initiate system reconfiguration.

Remote System Upgrade Mode

Remote system upgrade has one mode of operation—remote update mode. Remote update mode allows you to determine the functionality of your system after power-up and offers several features.

Remote Update Mode

In remote update mode, Stratix IV 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. The factory configuration may also contain application logic.

When used with serial configuration devices, remote update mode allows an application configuration to start at any flash sector boundary. For example, this translates to a maximum of 128 sectors in the EPCS64 device and 32 sectors in the EPCS16 device, where the minimum size of each page is 512 KBits. Altera recommends not using the same page in the serial configuration devices for two images. Additionally, remote update mode features a user watchdog timer that determines the validity of an application configuration.

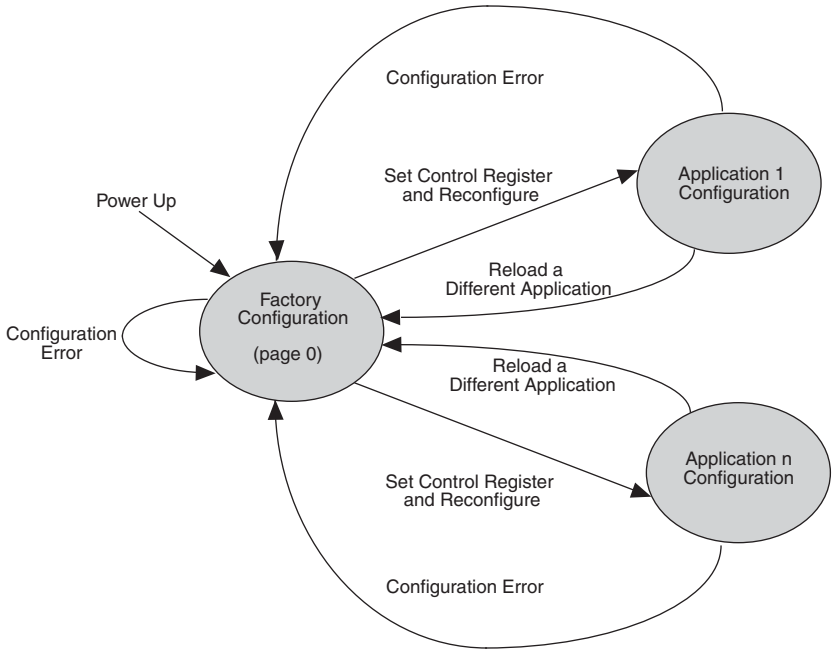
When a Stratix IV device is first powered up in remote update mode, it loads the factory configuration located at page zero (page registers `PGM[23:0] = 24'b0`). Always store the factory configuration image for your system at page address zero. This corresponds to the start address location 0×000000 in the serial configuration device.

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

- 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 IV device
- Enable or disable the user watchdog timer and load its time-out value (optional)
- Instruct the dedicated remote system upgrade circuitry to initiate a reconfiguration cycle

Figure 10–24 shows the transitions between the factory and application configurations in remote update mode.

Figure 10–24. Transitions Between Configurations in Remote Update Mode



After power up or a configuration error, the factory configuration logic is loaded automatically. The factory configuration also must specify whether to enable the user watchdog timer for the application configuration and if enabled, to include the timer setting information.

The user watchdog timer ensures that the application configuration is valid and functional. The timer must be continually reset within a specific amount of time during user mode operation of an application configuration. Only valid application configurations contain the logic to reset the timer in user mode. This timer reset logic must be part of a user-designed hardware and/or software health monitoring signal that indicates error-free system operation. If the timer is not reset in a specific amount of time; for example, the user application configuration detects a functional problem or if the system hangs, the dedicated circuitry updates the remote system upgrade status register, triggering the loading of the factory configuration.

The user watchdog timer is automatically disabled for factory configurations. For more information about the user watchdog timer, refer to "User Watchdog Timer" on page 10–61.

If there is an error while loading the application configuration, the cause of the reconfiguration is written by the dedicated circuitry to the remote system upgrade status register. Actions that cause the remote system upgrade status register to be written are:

- nSTATUS driven low externally
- Internal CRC error
- User watchdog timer time-out
- A configuration reset (logic array nCONFIG signal or external nCONFIG pin assertion to low)

Stratix IV devices automatically load the factory configuration located at page address zero. This user-designed factory configuration can read the remote system upgrade status register to determine the reason for the reconfiguration. The factory configuration then takes appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

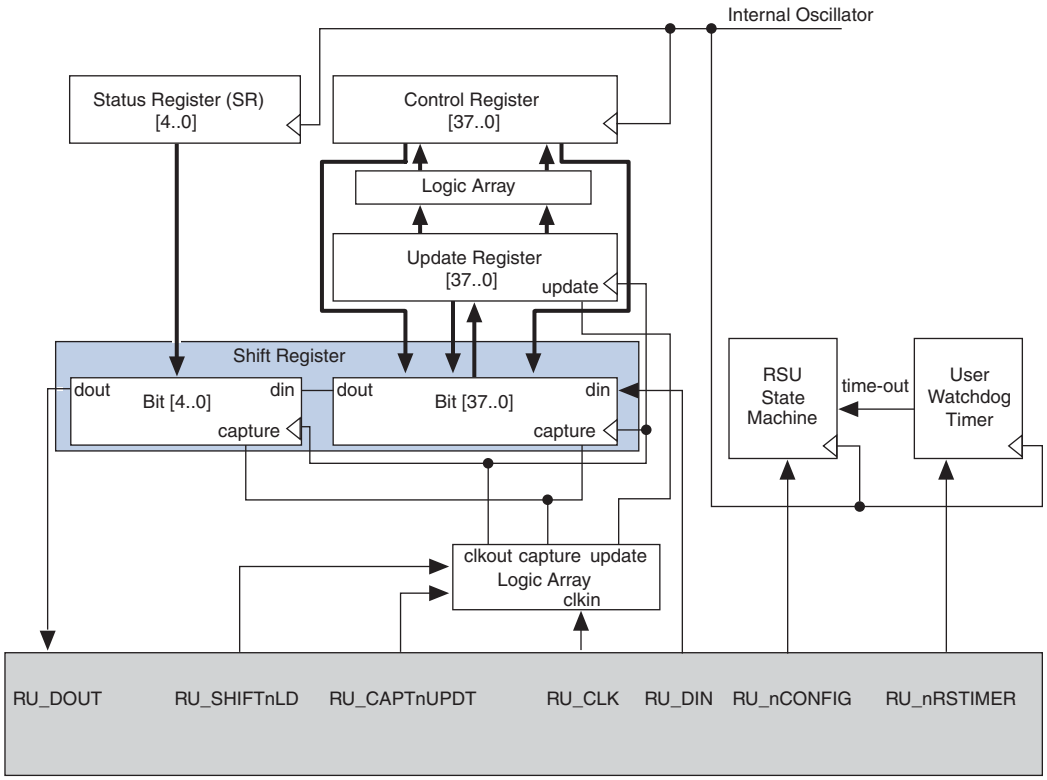
When Stratix IV devices successfully load the application configuration, they enter into user mode. In user mode, the soft logic (Nios II processor or state machine and the remote communication interface) assists the Stratix IV device in determining when a remote system update is arriving. When a remote system update arrives, 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.

Dedicated Remote System Upgrade Circuitry

This section describes the implementation of the Stratix IV remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces to the user-defined factory and application configurations implemented in the Stratix IV 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 10–25 shows the data path for the remote system upgrade block.

Figure 10–25. Remote System Upgrade Circuit Data Path ***(Note 1)***



Note to Figure 10–25:

(1) The RU_DOUT, RU_SHIFTnLD, RU_CAPTnUPDT, RU_CLK, RU_DIN, RU_nCONFIG, and RU_nRSTIMER signals are internally controlled by the ALTREMOTE_UPDATE megafunction.

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 10–14 lists these registers.

Table 10–14. 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.

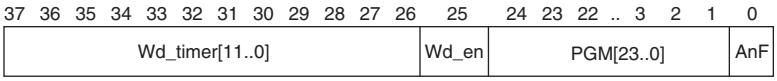
The remote system upgrade control and status registers are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer). However, the remote system upgrade shift and update registers are clocked by the user clock input (`RU_CLK`).

Remote System Upgrade Control Register

The remote system upgrade 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. In remote update mode, the control register page address bits are set to all zeros (`24'b0 = 0x000000`) at power up to load the factory configuration. A factory configuration in remote update mode has write access to this register.

Figure 10–26 and Table 10–15 specify the control register bit positions. In the figure, the numbers show the bit position of a setting within a register. For example, bit number 25 is the enable bit for the watchdog timer.

Figure 10–26. Remote System Upgrade Control Register



The application-not-factory (AnF) bit indicates whether the current configuration loaded in the Stratix IV 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 access is limited to read operations. When the AnF bit is low, the register allows write operations and disables the watchdog timer.

In remote update mode, the factory configuration design sets this bit high (1'b1) when updating the contents of the update register with the application page address and watchdog timer settings.

Table 10–15 lists the remote system upgrade control register contents.

Table 10–15. Remote System Upgrade Control Register Contents

Control Register Bit	Remote System Upgrade Mode	Value *(2)*	Definition
AnF *(1)*	Remote update	1'b0	Application not factory
PGM[23..0]	Remote update	24'b0×000000	AS configuration start address (StAdd[23..0])
Wd_en	Remote update	1'b0	User watchdog timer enable bit
Wd_timer[11..0]	Remote update	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 10–15:

(1) In remote update mode, the remote configuration block does not update the AnF bit automatically (you can update it manually).

(2) This is the default value of the control register bit.

Remote System Upgrade Status Register

The remote system upgrade status register specifies the reconfiguration trigger condition. The various trigger and error conditions include:

- Cyclic redundancy check (CRC) error during application configuration
- nSTATUS assertion by an external device due to an error
- Stratix IV device logic array triggered a reconfiguration cycle, possibly after downloading a new application configuration image
- External configuration reset (nCONFIG) assertion
- User watchdog timer time-out

Figure 10–27 and Table 10–16 specify the contents of the status register. The numbers in the figure show the bit positions within a 5-bit register.

Figure 10–27. Remote System Upgrade Status Register

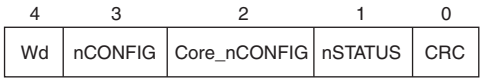


Table 10–16. Remote System Upgrade Status Register Contents

Status Register Bit	Definition	POR Reset Value
CRC (from the configuration)	CRC error caused reconfiguration	1 bit '0'
nSTATUS	nSTATUS caused reconfiguration	1 bit '0'
CORE_nCONFIG *(1)*	Device logic array caused reconfiguration	1 bit '0'
nCONFIG	nCONFIG caused reconfiguration	1 bit '0'
Wd	Watchdog timer caused reconfiguration	1 bit '0'

Note to Table 10–16:

(1) Logic array reconfiguration forces the system to load the application configuration data into the Stratix IV device. This occurs after the factory configuration specifies the appropriate application configuration page address by updating the update register.

Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (refer to Table 10–14 on page 10–57). While both registers can only be updated when the device is loaded with a factory configuration image, the update register writes are controlled by the user logic; the control register writes are controlled by the remote system upgrade state machine.

In factory configurations, the user logic sends the AnF bit (set high), the page address, and the watchdog timer settings for the next application configuration bit 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 initiates system reconfiguration from the new application page.

To ensure successful reconfiguration between the pages, assert the RU_nCONFIG signal for a minimum of 250 ns. This is equivalent to strobing the reconfiguration input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

In the event of an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (page zero or page one, based on the mode and error condition) by setting the control register accordingly. Table 10–17 lists the contents of the control register after such an event occurs for all possible error or trigger conditions.

The remote system upgrade status register is updated by the dedicated error monitoring circuitry after an error condition but before the factory configuration is loaded.

Table 10–17. Control Register Contents after an Error or Reconfiguration Trigger Condition

Reconfiguration Error/Trigger	Control Register Setting Remote Update
nCONFIG reset	All bits are 0
nSTATUS error	All bits are 0
CORE triggered reconfiguration	Update register
CRC error	All bits are 0
Wd time out	All bits are 0

Capture operations during factory configuration access the contents of the update register. This feature is used by the user logic to verify that the page address and watchdog timer settings were 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 IV device.

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 10-MHz internal oscillator. Table 10–18 lists the operating range of the 10-MHz internal oscillator.

Table 10–18. 10-MHz Internal Oscillator Specifications ***(Note 1)***

Minimum	Typical	Maximum	Units
4.3	5.3	10	MHz

Note to Table 10–18:

(1) These values are preliminary.

The user watchdog timer begins counting after the application configuration enters device user mode. This timer must be periodically reloaded or 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. The time-out signal tells the remote system upgrade circuitry to set the user watchdog timer status bit (`Wd`) in the remote system upgrade status register and reconfigures the device by loading the factory configuration.

To allow remote system upgrade dedicated circuitry to reset the watchdog timer, you must assert the `RU_nRSTIMER` signal active for a minimum of 250 ns. This is equivalent to strobing the `reset_timer` input of the ALTREMOTE_UPDATE megafunction high for a minimum of 250 ns.

The user watchdog timer is not enabled during the configuration cycle of the device. Errors during configuration are detected by the CRC engine. Also, the timer is disabled for factory configurations. Functional errors should not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

The user watchdog timer is disabled in factory configurations and during the configuration cycle of the application configuration. It is enabled after the application configuration enters user mode.

Quartus II Software Support

The Quartus II software provides the flexibility to include the remote system upgrade interface between the Stratix IV device logic array and the dedicated circuitry, generate configuration files for production, and allows remote programming of the system configuration memory.

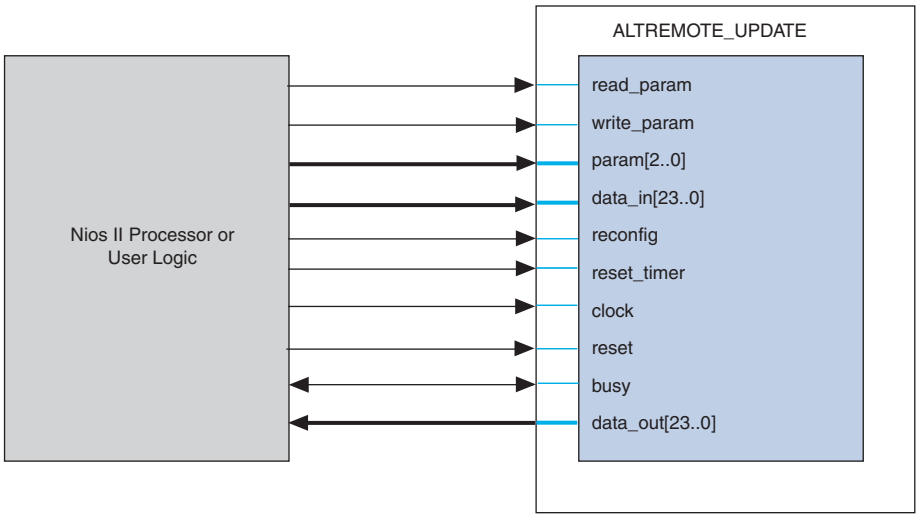
The ALTREMOTE_UPDATE megafunction is the implementation option in the Quartus II software that you use for the interface between the remote system upgrade circuitry and the device logic array interface. Using the megafunction block instead of creating your own logic saves design time and offers more efficient logic synthesis and device implementation.

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 IV 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.

Figure 10–28 shows the interface signals between the ALTREMOTE_UPDATE megafunction and Nios II processor or user logic.

Figure 10–28. Interface Signals between the ALTREMOTE_UPDATE Megafunction and the Nios II Processor



For more information about the ALTREMOTE_UPDATE megafunction and the description of ports shown in Figure 10–28, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide*.

Design Security

This section provides an overview of the design security feature and its implementation on Stratix IV devices using the advanced encryption standard (AES). It also covers the new security modes available in Stratix IV devices.

As Stratix IV devices continue play a role in larger and more critical designs in competitive commercial and military environments, it is increasingly important to protect the designs from copying, reverse engineering, and tampering.

Stratix IV devices address these concerns with both volatile and non-volatile security feature support. Stratix IV devices have the ability to decrypt configuration bitstreams using the AES algorithm, an industry-standard encryption algorithm that is FIPS-197 certified. Stratix IV devices have a design security feature that utilizes a 256-bit security key.

Stratix IV devices store configuration data in SRAM configuration cells during device operation. Because SRAM is volatile, the SRAM cells must be loaded with configuration data each time the device powers up. It is possible to intercept configuration data when it is being transmitted from the memory source (flash memory or a configuration device) to the device. The intercepted configuration data could then be used to configure another device.

When using the Stratix IV design security feature, the security key is stored in the Stratix IV device. Depending on the security mode, you can configure the Stratix IV device using a configuration file that is encrypted with the same key, or for board testing, configured with a normal configuration file.

The design security feature is available when configuring Stratix IV devices using FPP configuration mode with an external host (such as a MAX II device or microprocessor), or when using fast AS or PS configuration schemes. The design security feature is also available in remote update with fast AS configuration mode. The design security feature is not available when you are configuring your Stratix IV device using JTAG-based configuration. For more information, refer to "Supported Configuration Schemes" on page 10–67.

When using a serial configuration scheme such as PS or fast AS, configuration time is the same whether or not you enable the design security feature. If the FPP scheme is used with the design security or decompression feature, a ×4 DCLK is required. This results in a slower configuration time when compared with the configuration time of a Stratix IV device that has neither the design security nor the decompression feature enabled.

Stratix IV Security Protection

Stratix IV device designs are protected from copying, reverse engineering, and tampering using configuration bitstream encryption.

Security Against Copying

The security key is securely stored in the Stratix IV device and cannot be read out through any interfaces. In addition, as configuration file read-back is not supported in Stratix IV devices, the 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 IV configuration file formats are proprietary and the file contains millions of bits which require specific decryption. Reverse engineering the Stratix IV device is just as difficult because the device is manufactured on the most advanced 40-nm process technology.

Security Against Tampering

The non-volatile keys are one-time programmable. After the Tamper Protection bit is set in the key programming file generated by the Quartus II software, the Stratix IV device can only be configured with configuration files encrypted with the same key.

AES Decryption Block

The main purpose of the AES decryption block is to decrypt the configuration bitstream prior to entering data decompression or configuration.

Prior to receiving encrypted data, you must enter and store the 256-bit security key in the device. You can choose between a non-volatile security key and a volatile security key with battery backup.

The security key is scrambled prior to storing it in the key storage to make it more difficult for anyone to retrieve the stored key using de-capsulation of the device.

Flexible Security Key Storage

Stratix IV devices support two types of security key programming—volatile and non-volatile keys. Table 10–19 lists the differences between volatile keys and non-volatile keys.

Table 10–19. Security Key Options (Part 1 of 2)

Options	Volatile Key	Non-Volatile Key
Key programmability	Reprogrammable and erasable	One-time programmable
External battery	Required	Not required
Key programming method *(1)*	On-board	On and off board

Table 10–19. Security Key Options (Part 2 of 2)

Options	Volatile Key	Non-Volatile Key
Design protection	Secure against copying and reverse engineering	Secure against copying and reverse engineering. Tamper resistant if tamper protection bit is set.

Note to Table 10–19:

(1) Key programming is carried out using the JTAG interface.

You can program the non-volatile key to the Stratix IV device without an external battery. Also, there are no additional requirements to any of the Stratix IV power supply inputs.

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_{CC}. V_{CCBAT} continuously supplies power to the volatile register regardless of the on-chip supply condition.

After power-up, you must wait 300 ms (PORSEL = 0) or 12 ms (PORSEL = 1) before beginning key programming to ensure that V_{CCBAT} is at full rail.

For more information about how to calculate the key retention time of the battery used for volatile key storage, refer to the *Stratix III, Stratix IV, Stratix V, HardCopy III and HardCopy IV PowerPlay Early Power Estimator*.

For more information about battery specifications, refer to the *DC and Switching Characteristics for Stratix IV Devices* chapter.

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

Stratix IV Design Security Solution

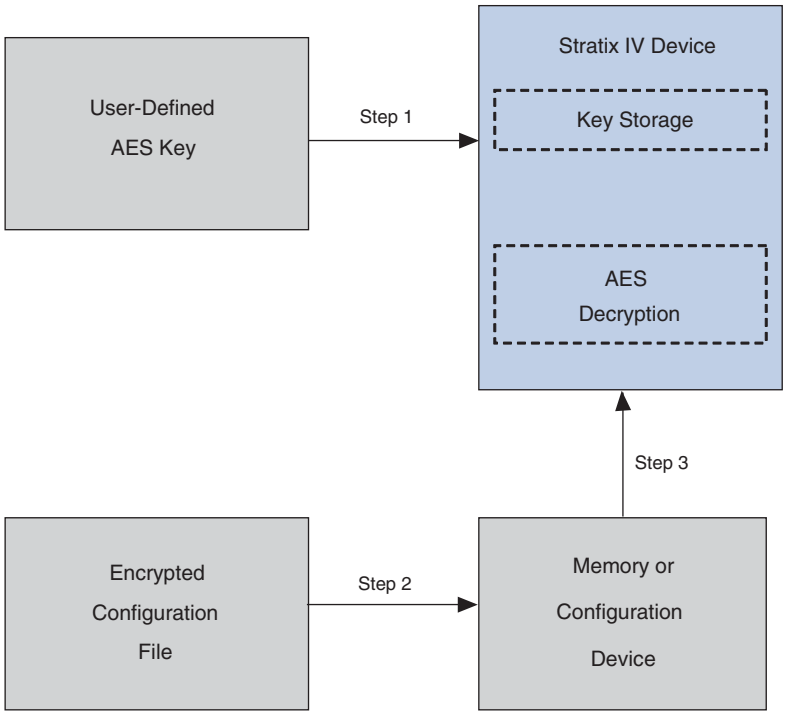
Stratix IV devices are SRAM-based devices. To provide design security, Stratix IV devices require a 256-bit security key for configuration bitstream encryption.

You can carry out secure configuration in the following steps, as shown in Figure 10–29:

1. Program the security key into the Stratix IV device.
2. Program the user-defined 256-bit AES keys to the Stratix IV device through the JTAG interface.
3. Encrypt the configuration file and store it in the external memory.
4. Encrypt the configuration file with the same 256-bit keys used to program the Stratix IV device. Encryption of the configuration file is done using the Quartus II software. The encrypted configuration file is then loaded into the external memory, such as a configuration or flash device.
5. Configure the Stratix IV device.

At system power-up, the external memory device sends the encrypted configuration file to the Stratix IV device.

Figure 10–29. Design Security ***(Note 1)***



Note to Figure 10–29:

(1) Step 1, Step 2, and Step 3 correspond to the procedure described in "Design Security" on page 10–63.

Security Modes Available

The following security modes are available on the Stratix IV device.

Volatile Key

Secure operation with volatile key programmed and required external battery: this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board-level testing only.

Non-Volatile Key

Secure operation with one time programmable (OTP) security key programmed: this mode accepts both encrypted and unencrypted configuration bitstreams. Use the unencrypted configuration bitstream support for board level testing only.

Non-Volatile Key with Tamper Protection Bit Set

Secure operation in tamper resistant mode with OTP security key programmed: only encrypted configuration bitstreams are allowed to configure the device. Tamper protection disables JTAG configuration with unencrypted configuration bitstream.

Enabling the tamper protection bit disables test mode in Stratix IV devices. This process is irreversible and prevents Altera from conducting carry-out failure analysis if test mode is disabled. Contact Altera Technical Support to enable the tamper protection bit.

No Key Operation

Only unencrypted configuration bitstreams are allowed to configure the device.

Table 10–20 lists the different security modes and configuration bitstream supported for each mode.

Table 10–20. Security Modes Supported

Mode *(1)*	Function	Configuration File
Volatile key	Secure	Encrypted
	Board-level testing	Unencrypted
Non-volatile key	Secure	Encrypted
	Board-level testing	Unencrypted
Non-volatile key with tamper protection bit set	Secure (tamper resistant) *(2)*	Encrypted

Notes to Table 10–20:

(1) In No key operation, only the unencrypted configuration file is supported.

(2) The tamper protection bit setting does not prevent the device from being reconfigured.

Supported Configuration Schemes

The Stratix IV device supports only selected configuration schemes, depending on the security mode you select when you encrypt the Stratix IV device.

Figure 10–30 shows the restrictions of each security mode when encrypting Stratix IV devices.

Figure 10–30. Security Modes in Stratix IV Devices—Sequence and Restrictions

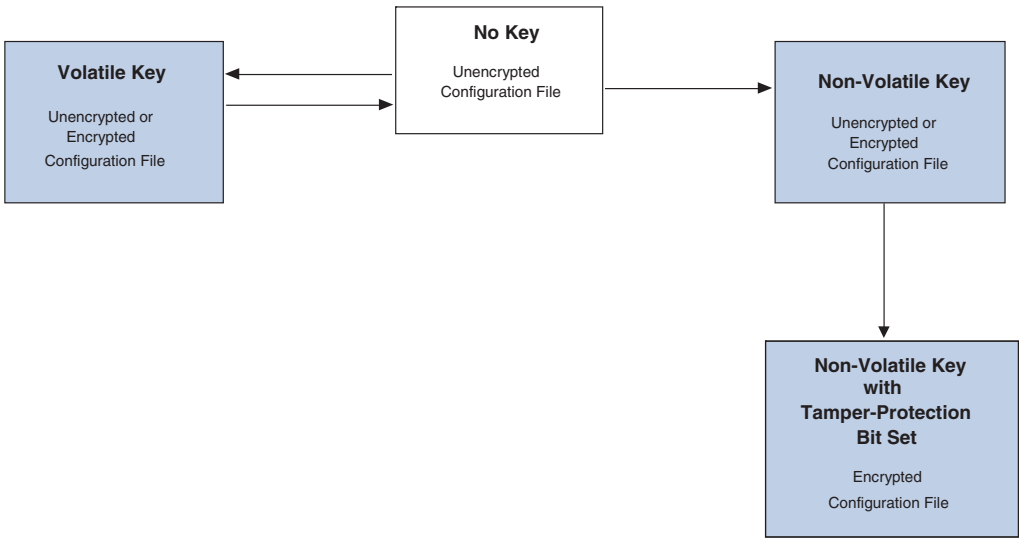


Table 10–21 lists the configuration modes allowed in each of the security modes.

Table 10–21. Allowed Configuration Modes for Various Security Modes ***(Note 1)***

Security Mode	Configuration File	Allowed Configuration Modes
No key	Unencrypted	All configuration modes that do not engage the design security feature.
Secure with volatile key	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)
Board-level testing with volatile key	Unencrypted	All configuration modes that do not engage the design security feature.
Secure with non-volatile key	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)
Board-level testing with non-volatile key	Unencrypted	All configuration modes that do not engage the design security feature.
Secure in tamper resistant mode using non-volatile key with tamper protection set	Encrypted	■ Passive serial with AES (and/or with decompression) ■ Fast passive parallel with AES (and/or with decompression) ■ Remote update fast AS with AES (and/or with decompression) ■ Fast AS (and/or with decompression)

Note to Table 10–21:

(1) There is no impact to the configuration time required when compared with unencrypted configuration modes except FPP with AES (and/or decompression), which requires a DCLK that is ×4 the data rate.

You can use the design security feature with other configuration features, such as compression and remote system upgrade features. 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 IV device first decrypts and then decompresses the configuration file.

Document Revision History

Table 10–22 lists the revision history for this chapter.

Table 10–22. Document Revision History (Part 1 of 2)

Date	Version	Changes
September 2012	3.5	■ Updated the "FPP Configuration Using a MAX II Device as an External Host" section to close FB #36583 and #63157. ■ Updated the "Estimating Active Serial Configuration Time" section to close FB #64163. ■ Updated the "PS Configuration Using a MAX II Device as an External Host" section to close FB #63157. ■ Updated Figure 10–1, Figure 10–2, Figure 10–3, Figure 10–10,Figure 10–11 and Figure 10–12 to close FB #63155.
December 2011	3.4	Updated Table 10–2 and Table 10–7.

Table 10–22. Document Revision History (Part 2 of 2)

Date	Version	Changes
April 2011	3.3	■ Updated the "FPP Configuration Using a MAX II Device as an External Host", "Fast Active Serial Configuration (Serial Configuration Devices)", and "PS Configuration Using a MAX II Device as an External Host". ■ Updated Table 10–10.
February 2011	3.2	■ Updated the "Fast Active Serial Configuration (Serial Configuration Devices)", "FPP Configuration Using a MAX II Device as an External Host" "Configuration Data Decompression", and "User Watchdog Timer" sections. ■ Updated Table 10–2, Table 10–4, Table 10–5, Table 10–7, and Table 10–9. ■ Applied new template. ■ Minor text edits.
March 2010	3.1	■ Added the "Guidelines for Connecting Serial Configuration Devices on an AS Interface" section. ■ Updated the "Power-On Reset Circuit" and "Fast Active Serial Configuration (Serial Configuration Devices)" sections. ■ Updated Table 10–2, Table 10–4, Table 10–5, Table 10–10, and Table 10–13. ■ Updated Figure 10–16 and Figure 10–17 with Note 5. ■ Updated Figure 10–4, Figure 10–5, and Figure 10–13. ■ Updated the reference in the "Configuration Schemes" section.
November 2009	3.0	■ Updated Table 10–1 and Table 10–2. ■ Updated the "FPP Configuration Using a MAX II Device as an External Host","Fast Active Serial Configuration (Serial Configuration Devices)", "Device Configuration Pins", "Remote System Upgrades", "Remote System Upgrade Mode", "Estimating Active Serial Configuration Time", "Remote System Upgrade State Machine", and "User Watchdog Timer" sections. ■ Removed Table 10-4, Table 10-7, Table 10-8, and Table 10-25. ■ Minor text edits.
June 2009	2.3	■ Updated the "VCCPD Pins", "FPP Configuration Using a MAX II Device as an External Host", "Estimating Active Serial Configuration Time", "Fast Active Serial Configuration (Serial Configuration Devices)", "Remote System Upgrades", "PS Configuration Using a MAX II Device as an External Host", and "PS Configuration Using a Download Cable" sections. ■ Updated Table 10–3, Table 10–13 and Table 10–2. ■ Added introductory sentences to improve search ability. ■ Removed the Conclusion section. ■ Minor text edits.
April 2009	2.2	■ Updated Table 10–2.
March 2009	2.1	■ Updated Table 10–1, Table 10–2, and Table 10–9. ■ Removed "Referenced Documents" section.
November 2008	2.0	■ Updated "Fast Active Serial Configuration (Serial Configuration Devices)" and "JTAG Configuration" sections. ■ Updated Figure 10–4, Figure 10–5, Figure 10–6, and Figure 10–13. ■ Updated Table 10–2 and Table 10–13.
May 2008	1.0	Initial release.