# 8. Configuration and Remote System Upgrades in Cyclone IV Devices

CYIV-51008-1.7

This chapter describes the configuration and remote system upgrades in Cyclone® IV devices. Cyclone IV (Cyclone IV GX and Cyclone IV E) devices use SRAM cells to store configuration data. You must download the configuration data to Cyclone IV devices each time the device powers up because SRAM memory is volatile.

Cyclone IV devices are configured using one of the following configuration schemes:

- Active serial (AS)
- Active parallel (AP) (supported in Cyclone IV E devices only)
- Passive serial (PS)
- Fast passive parallel (FPP) (not supported in EP4CGX15, EP4CGX22, and EP4CGX30 [except for the F484 package] devices)
- JTAG

Cyclone IV devices offer the following configuration features:

- Configuration data decompression ("Configuration Data Decompression" on page 8–2)
- Remote system upgrade ("Remote System Upgrade" on page 8–69)

System designers face difficult challenges, such as shortened design cycles, evolving standards, and system deployments in remote locations. Cyclone IV devices help overcome these challenges with inherent re-programmability and dedicated circuitry to perform remote system upgrades. Remote system upgrades help deliver feature enhancements and bug fixes without costly recalls, reduced time-to-market, and extended product life.

## Configuration

This section describes Cyclone IV device configuration and includes the following topics:

- "FPP Configuration" on page 8–40
- "JTAG Configuration" on page 8–45
- "Device Configuration Pins" on page 8–62

## Configuration Features

Table 8–1 lists the configuration methods you can use in each configuration scheme.

**Table 8–1. Configuration Features in Cyclone IV Devices**

| Configuration Scheme | Configuration Method | Decompression | Remote System Upgrade (1) |
|---|---|---|---|
| AS | Serial Configuration Device | ✓ | ✓ |
| AP | Supported Flash Memory (2) | — | ✓ |
| PS | External Host with Flash Memory | ✓ | ✓ (3) |
| | Download Cable | ✓ | — |
| FPP | External Host with Flash Memory | — | ✓ (3) |
| JTAG based configuration | External Host with Flash Memory | — | — |
| | Download Cable | — | — |

**Notes to Table 8–1:**

(1) Remote update mode is supported when you use the Remote System Upgrade feature. You can enable or disable remote update mode with an option setting in the Quartus® II software.

(2) For more information about the supported device families for the Micron commodity parallel flash, refer to Table 8–10 on page 8–22.

(3) Remote update mode is supported externally using the Parallel Flash Loader (PFL) with the Quartus II software.

### Configuration Data Decompression

Cyclone 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 send the compressed bitstream to Cyclone IV devices. During configuration, Cyclone IV devices decompress the bitstream in real time and program the SRAM cells.

Compression may reduce the configuration bitstream size by 35 to 55%.

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 required to send the bitstream to the Cyclone IV device. The time required by a Cyclone IV device to decompress a configuration file is less than the time required to send the configuration data to the device. There are two methods for enabling compression for the Cyclone IV device bitstreams in the Quartus II software:

- Before design compilation (through the Compiler Settings menu)
- After design compilation (through the **Convert Programming Files** dialog box)

To enable compression in the compiler settings of the project in the Quartus II software, perform the following steps:

1. On the Assignments menu, click **Device**. The **Settings** dialog box appears.
2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.

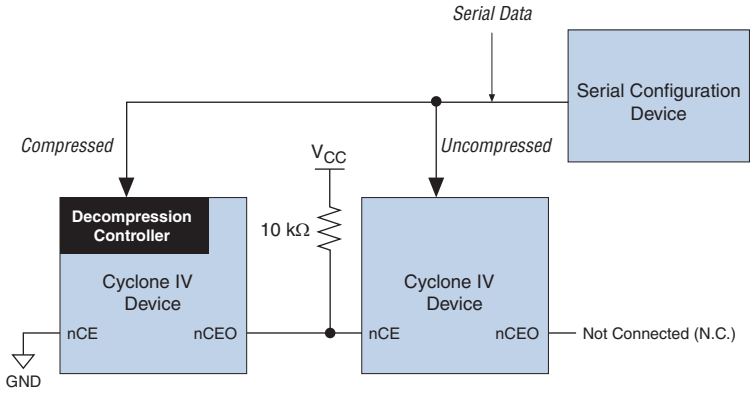
3. Click the **Configuration** tab.
4. Turn on **Generate compressed bitstreams**.
5. Click **OK**.
6. In the **Settings** dialog box, click **OK**.

You can enable compression when creating programming files from the **Convert Programming Files** dialog box. To enable compression, perform the following steps:

1. On the File menu, click **Convert Programming Files**.
2. Under **Output programming file**, select your desired file type from the **Programming file type** list.
3. If you select **Programmer Object File (.pof)**, you must specify the configuration device in the **Configuration device** list.
4. Under **Input files to convert**, select **SOF Data**.
5. Click **Add File** to browse to the Cyclone IV device SRAM object files (**.sof)**.
6. In the **Convert Programming Files** dialog box, select the **.pof** you added to **SOF Data** and click **Properties**.
7. In the **SOF File Properties** dialog box, turn on the **Compression** option.

When multiple Cyclone IV devices are cascaded, you can selectively enable the compression feature for each device in the chain. Figure 8–1 shows a chain of two Cyclone IV devices. The first device has compression enabled and receives compressed bitstream from the configuration device. The second device has the compression feature disabled and receives uncompressed data. You can generate programming files for this setup in the **Convert Programming Files** dialog box.

**Figure 8–1. Compressed and Uncompressed Configuration Data in the Same Configuration File**




## Configuration Requirement

This section describes Cyclone IV device configuration requirement and includes the following topics:

- "Power-On Reset (POR) Circuit" on page 8–4
- "Configuration File Size" on page 8–4
- "Power Up" on page 8–6

## Power-On Reset (POR) Circuit

The POR circuit keeps the device in reset state until the power supply voltage levels have stabilized during device power up. After device power up, the device does not release nSTATUS until $V_{CCINT}$, $V_{CCA}$, and $V_{CCIO}$ (for I/O banks in which the configuration and JTAG pins reside) are above the POR trip point of the device. $V_{CCINT}$ and $V_{CCA}$ are monitored for brown-out conditions after device power up.

$V_{CCA}$ is the analog power to the phase-locked loop (PLL).

In some applications, it is necessary for a device to wake up very quickly to begin operation. Cyclone IV devices offer the fast POR time option to support fast wake-up time applications. The fast POR time option has stricter power-up requirements when compared with the standard POR time option. You can select either the fast option or the standard POR option with the MSEL pin settings.

If your system exceeds the fast or standard POR time, you must hold nCONFIG low until all the power supplies are stable.

For more information about the POR specifications, refer to the *Cyclone IV Device Datasheet*.

For more information about the wake-up time and POR circuit, refer to the *Power Requirements for Cyclone IV Devices* chapter.

## Configuration File Size

Table 8–2 lists the approximate uncompressed configuration file sizes for Cyclone IV devices. To calculate the amount of storage space required for multiple device configurations, add the file size of each device together.

**Table 8–2. Uncompressed Raw Binary File (.rbf) Sizes for Cyclone IV Devices (Part 1 of 2)**

| Device | | Data Size (bits) |
|---|---|---|
| Cyclone IV E | EP4CE6 | 2,944,088 |
| | EP4CE10 | 2,944,088 |
| | EP4CE15 | 4,086,848 |
| | EP4CE22 | 5,748,552 |
| | EP4CE30 | 9,534,304 |
| | EP4CE40 | 9,534,304 |
| | EP4CE55 | 14,889,560 |
| | EP4CE75 | 19,965,752 |
| | EP4CE115 | 28,571,696 |

**Table 8–2. Uncompressed Raw Binary File (.rbf) Sizes for Cyclone IV Devices (Part 2 of 2)**

| Device | | Data Size (bits) |
|---|---|---|
| Cyclone IV GX | EP4CGX15 | 3,805,568 |
| | EP4CGX22 | 7,600,040 |
| | EP4CGX30 | 7,600,040 |
| | | 22,010,888 (1) |
| | EP4CGX50 | 22,010,888 |
| | EP4CGX75 | 22,010,888 |
| | EP4CGX110 | 39,425,016 |
| | EP4CGX150 | 39,425,016 |

**Note to Table 8–2:**

(1) Only for the F484 package.

Use the data in Table 8–2 to estimate the file size before design compilation. Different configuration file formats, such as Hexadecimal (**.hex**) or Tabular Text File (**.ttf**) formats, have different file sizes. However, for any specific version of the Quartus II software, any design targeted for the same device has the same uncompressed configuration file size. If you use compression, the file size varies after each compilation, because the compression ratio depends on the design.

For more information about setting device configuration options or creating configuration files, refer to the *Software Settings* section in volume 2 of the *Configuration Handbook*.

## Configuration and JTAG Pin I/O Requirements

Cyclone IV devices are manufactured using the TSMC 60-nm low-k dielectric process. Although Cyclone IV devices use TSMC 2.5-V transistor technology in the I/O buffers, the devices are compatible and able to interface with 2.5, 3.0, and 3.3-V configuration voltage standards by following specific requirements.

All I/O inputs must maintain a maximum AC voltage of 4.1 V. When using a serial configuration device in an AS configuration scheme, you must connect a 25-Ω series resistor for the DATA[0] pin. When cascading the Cyclone IV device family in a multi-device configuration for AS, AP, FPP, and PS configuration schemes, you must connect the repeater buffers between the master and slave devices for the DATA and DCLK pins. When using the JTAG configuration scheme in a multi-device configuration, connect 25-Ω resistors on both ends of the TDO-TDI path if the TDO output driver is a non-Cyclone IV device.

The output resistance of the repeater buffers and the TDO path for all cases must fit the maximum overshoot equation shown in Equation 8–1.

**Equation 8–1. (1)**

$$0.8Z_O \leq R_E \leq 1.8Z_O$$

**Note to Equation 8–1:**

(1) $Z_O$ is the transmission line impedance and $R_E$ is the equivalent resistance of the output buffer.

## Configuration Process

This section describes Cyclone IV device configuration requirements and includes the following topics:



For more information about the Altera® FPGA configuration cycle state machine, refer to the *Configuring Altera FPGAs* chapter in volume 1 of the *Configuration Handbook*.

### Power Up

If the device is powered up from the power-down state, $V_{CCINT}$, $V_{CCA}$, and $V_{CCIO}$ (for the I/O banks in which the configuration and JTAG pins reside) must be powered up to the appropriate level for the device to exit from POR.

### Reset

After power up, Cyclone IV devices go through POR. POR delay depends on the MSEL pin settings, which correspond to your configuration scheme. During POR, the device resets, holds nSTATUS and CONF_DONE low, and tri-states all user I/O pins (for PS and FPP configuration schemes only).

To tri-state the configuration bus for AS and AP configuration schemes, you must tie nCE high and nCONFIG low.

The user I/O pins and dual-purpose I/O pins have weak pull-up resistors, which are always enabled (after POR) before and during configuration. When the device exits POR, all user I/O pins continue to tri-state. While nCONFIG is low, the device is in reset. When nCONFIG goes high, the device exits reset and releases the open-drain nSTATUS pin, which is then pulled high by an external 10-kΩ pull-up resistor. After nSTATUS is released, the device is ready to receive configuration data and the configuration stage starts.

For more information about the value of the weak pull-up resistors on the I/O pins that are on before and during configuration, refer to the *Cyclone IV Device Datasheet* chapter.

### Configuration

Configuration data is latched into the Cyclone IV device at each DCLK cycle. However, the width of the data bus and the configuration time taken for each scheme are different. After the device receives all the configuration data, the device releases the open-drain CONF_DONE pin, which is pulled high by an external 10-kΩ pull-up resistor. A low-to-high transition on the CONF_DONE pin indicates that the configuration is complete and initialization of the device can begin.

You can begin reconfiguration by pulling the nCONFIG pin low. The nCONFIG pin must be low for at least 500 ns. When nCONFIG is pulled low, the Cyclone IV device is reset. The Cyclone IV device also pulls nSTATUS and CONF_DONE low and all I/O pins are tri-stated. When nCONFIG returns to a logic-high level and nSTATUS is released by the Cyclone IV device, reconfiguration begins.

## Configuration Error

If an error occurs during configuration, Cyclone 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 in the **General** tab of the **Device and Pin Options** dialog box) is turned on, the Cyclone IV device releases nSTATUS after a reset time-out period (a maximum of 230 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 500 ns to restart configuration.

## Initialization

In Cyclone 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 device provides itself with enough clock cycles for proper initialization. When using the internal oscillator, you do not have to send additional clock cycles from an external source to the CLKUSR pin during the initialization stage. Additionally, you can use the CLKUSR pin as a user I/O pin.

You also have the flexibility to synchronize initialization of multiple devices or to delay initialization with the **CLKUSR** option. The CLKUSR pin allows you to control when your device enters user mode for an indefinite amount of time. You can turn on the **Enable user-supplied start-up clock (CLKUSR)** option in the Quartus II software in the **General** tab of the **Device and Pin Options** dialog box. When you turn on the **Enable user supplied start-up clock option (CLKUSR)** option, the CLKUSR pin is the initialization clock source. Supplying a clock on the CLKUSR pin does not affect the configuration process. After the configuration data is accepted and CONF_DONE goes high, Cyclone IV devices require 3,192 clock cycles to initialize properly and enter user mode.

If you use the optional CLKUSR pin and the nCONFIG pin is pulled low to restart configuration during device initialization, ensure that the CLKUSR pin continues to toggle when nSTATUS is low (a maximum of 230 µs).

## User Mode

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

## Configuration Scheme

A configuration scheme with different configuration voltage standards is selected by driving the MSEL pins either high or low, as shown in Table 8–3, Table 8–4, and Table 8–5.

Hardwire the MSEL pins to $V_{CCA}$ or GND without pull-up or pull-down resistors to avoid problems detecting an incorrect configuration scheme. Do not drive the MSEL pins with a microprocessor or another device.

**Table 8–3. Configuration Schemes for Cyclone IV GX Devices (EP4CGX15, EP4CGX22, and EP4CGX30 [except for F484 Package])**

| Configuration Scheme | MSEL2 | MSEL1 | MSEL0 | POR Delay | Configuration Voltage Standard (V) (1) |
|---|---|---|---|---|---|
| AS | 1 | 0 | 1 | Fast | 3.3 |
| | 0 | 1 | 1 | Fast | 3.0, 2.5 |
| | 0 | 0 | 1 | Standard | 3.3 |
| | 0 | 1 | 0 | Standard | 3.0, 2.5 |
| PS | 1 | 0 | 0 | Fast | 3.3, 3.0, 2.5 |
| | 1 | 1 | 0 | Fast | 1.8, 1.5 |
| | 0 | 0 | 0 | Standard | 3.3, 3.0, 2.5 |
| JTAG-based configuration (2) | (3) | (3) | (3) | — | — |

**Notes to Table 8–3:**

(1) Configuration voltage standard applied to the $V_{CCIO}$ supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to $V_{CCA}$ or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

**Table 8–4. Configuration Schemes for Cyclone IV GX Devices (EP4CGX30 [only for F484 package], EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150) (Part 1 of 2)**

| Configuration Scheme | MSEL3 | MSEL2 | MSEL1 | MSEL0 | POR Delay | Configuration Voltage Standard (V) (1) |
|---|---|---|---|---|---|---|
| AS | 1 | 1 | 0 | 1 | Fast | 3.3 |
| | 1 | 0 | 1 | 1 | Fast | 3.0, 2.5 |
| | 1 | 0 | 0 | 1 | Standard | 3.3 |
| | 1 | 0 | 1 | 0 | Standard | 3.0, 2.5 |
| PS | 1 | 1 | 0 | 0 | Fast | 3.3, 3.0, 2.5 |
| | 1 | 1 | 1 | 0 | Fast | 1.8, 1.5 |
| | 1 | 0 | 0 | 0 | Standard | 3.3, 3.0, 2.5 |
| | 0 | 0 | 0 | 0 | Standard | 1.8, 1.5 |
| FPP | 0 | 0 | 1 | 1 | Fast | 3.3, 3.0, 2.5 |
| | 0 | 1 | 0 | 0 | Fast | 1.8, 1.5 |
| | 0 | 0 | 0 | 1 | Standard | 3.3, 3.0, 2.5 |
| | 0 | 0 | 1 | 0 | Standard | 1.8, 1.5 |

**Table 8–4. Configuration Schemes for Cyclone IV GX Devices (EP4CGX30 [only for F484 package], EP4CGX50, EP4CGX75, EP4CGX110, and EP4CGX150) (Part 2 of 2)**

| Configuration Scheme | MSEL3 | MSEL2 | MSEL1 | MSEL0 | POR Delay | Configuration Voltage Standard (V) (1) |
|---|---|---|---|---|---|---|
| JTAG-based configuration (2) | (3) | (3) | (3) | (3) | — | — |

**Notes to Table 8–4:**

(1) Configuration voltage standard applied to the $V_{CCIO}$ supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to $V_{CCA}$ or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

Smaller Cyclone IV E devices or package options (E144 and F256 packages) do not have the MSEL[3] pin. The AS Fast POR configuration scheme at 3.0- or 2.5-V configuration voltage standard and the AP configuration scheme are not supported in Cyclone IV E devices without the MSEL[3] pin. To configure these devices with other supported configuration schemes, select MSEL[2..0] pins according to the MSEL settings in Table 8–5.

**Table 8–5. Configuration Schemes for Cyclone IV E Devices**

| Configuration Scheme | MSEL3 | MSEL2 | MSEL1 | MSEL0 | POR Delay | Configuration Voltage Standard (V) (1) |
|---|---|---|---|---|---|---|
| AS | 1 | 1 | 0 | 1 | Fast | 3.3 |
| | 0 | 1 | 0 | 0 | Fast | 3.0, 2.5 |
| | 0 | 0 | 1 | 0 | Standard | 3.3 |
| | 0 | 0 | 1 | 1 | Standard | 3.0, 2.5 |
| AP | 0 | 1 | 0 | 1 | Fast | 3.3 |
| | 0 | 1 | 1 | 0 | Fast | 1.8 |
| | 0 | 1 | 1 | 1 | Standard | 3.3 |
| | 1 | 0 | 1 | 1 | Standard | 3.0, 2.5 |
| | 1 | 0 | 0 | 0 | Standard | 1.8 |
| PS | 1 | 1 | 0 | 0 | Fast | 3.3, 3.0, 2.5 |
| | 0 | 0 | 0 | 0 | Standard | 3.3, 3.0, 2.5 |
| FPP | 1 | 1 | 1 | 0 | Fast | 3.3, 3.0, 2.5 |
| | 1 | 1 | 1 | 1 | Fast | 1.8, 1.5 |
| JTAG-based configuration (2) | (3) | (3) | (3) | (3) | — | — |

**Notes to Table 8–5:**

(1) Configuration voltage standard applied to the $V_{CCIO}$ supply of the bank in which the configuration pins reside.

(2) JTAG-based configuration takes precedence over other configuration schemes, which means the MSEL pin settings are ignored.

(3) Do not leave the MSEL pins floating. Connect them to $V_{CCA}$ or GND. These pins support the non-JTAG configuration scheme used in production. Altera recommends connecting the MSEL pins to GND if your device is only using JTAG configuration.

For Cyclone IV E devices, the Quartus II software prohibits you from using the LVDS I/O standard in I/O Bank 1 when the configuration device I/O voltage is not 2.5 V. If you need to assign LVDS I/O standard in I/O Bank 1, navigate to **Assignments**>**Device**>**Settings**>**Device and Pin Option**>**Configuration** to change the Configuration Device I/O voltage to **2.5 V** or **Auto**.

## AS Configuration (Serial Configuration Devices)

In the AS configuration scheme, Cyclone IV devices are configured with a serial configuration device. These configuration devices are low-cost devices with non-volatile memories that feature a simple four-pin interface and a small form factor. These features make serial configuration devices the ideal low-cost configuration solution.

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

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

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

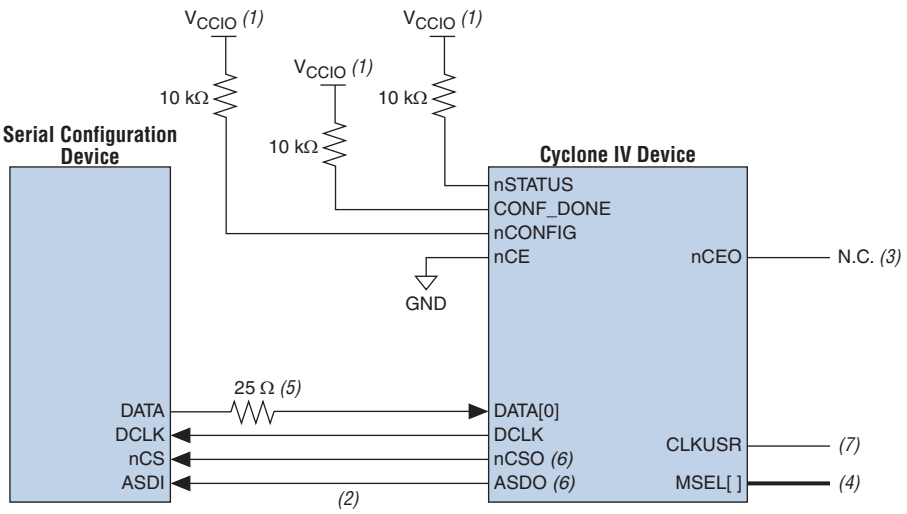
### Single-Device AS Configuration

The four-pin interface of serial configuration devices consists of the following pins:

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

This four-pin interface connects to Cyclone IV device pins, as shown in Figure 8–2.

**Figure 8–2. Single-Device AS Configuration**




**Notes to Figure 8–2:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

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

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(5) Connect the series resistor at the near end of the serial configuration device.

(6) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as the DATA[1] pin in AP and FPP modes.

(7) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

To tri-state the configuration bus for AS configuration schemes, you must tie nCE high and nCONFIG low.

The 25-Ω resistor at the near end of the serial configuration device for DATA[0] works to minimize the driver impedance mismatch with the board trace and reduce the overshoot seen at the Cyclone IV device DATA[0] input pin.

In the single-device AS configuration, the maximum board loading and board trace length between the supported serial configuration device and the Cyclone IV device must follow the recommendations in Table 8–7 on page 8–18.

The DCLK generated by the Cyclone IV device controls the entire configuration cycle and provides timing for the serial interface. Cyclone IV devices use an internal oscillator or an external clock source to generate the DCLK. For Cyclone IV E devices, you can use a 40-MHz internal oscillator to generate the DCLK and for Cyclone IV GX devices you can use a slow clock (20 MHz maximum) or a fast clock (40 MHz maximum) from the internal oscillator or an external clock from CLKUSR to generate the DCLK. There are some variations in the internal oscillator frequency because of the process, voltage, and temperature (PVT) conditions in Cyclone IV

devices. The internal oscillator is designed to ensure that its maximum frequency is guaranteed to meet EPCS device specifications. Cyclone IV devices offer the option to select CLKUSR as the external clock source for DCLK. You can change the clock source option in the Quartus II software in the **Configuration** tab of the **Device and Pin Options** dialog box.

EPCS1 does not support Cyclone IV devices because of its insufficient memory capacity.

**Table 8–6. AS DCLK Output Frequency**

| Oscillator | Minimum | Typical | Maximum | Unit |
|---|---|---|---|---|
| 40 MHz | 20 | 30 | 40 | MHz |

In configuration mode, the Cyclone IV device enables the serial configuration device by driving the nCSO output pin low, which connects to the nCS pin of the configuration device. The Cyclone IV device uses the DCLK and DATA[1] pins to send operation commands and read address signals to the serial configuration device. The configuration device provides data on its DATA pin, which connects to the DATA[0] input of the Cyclone IV device.

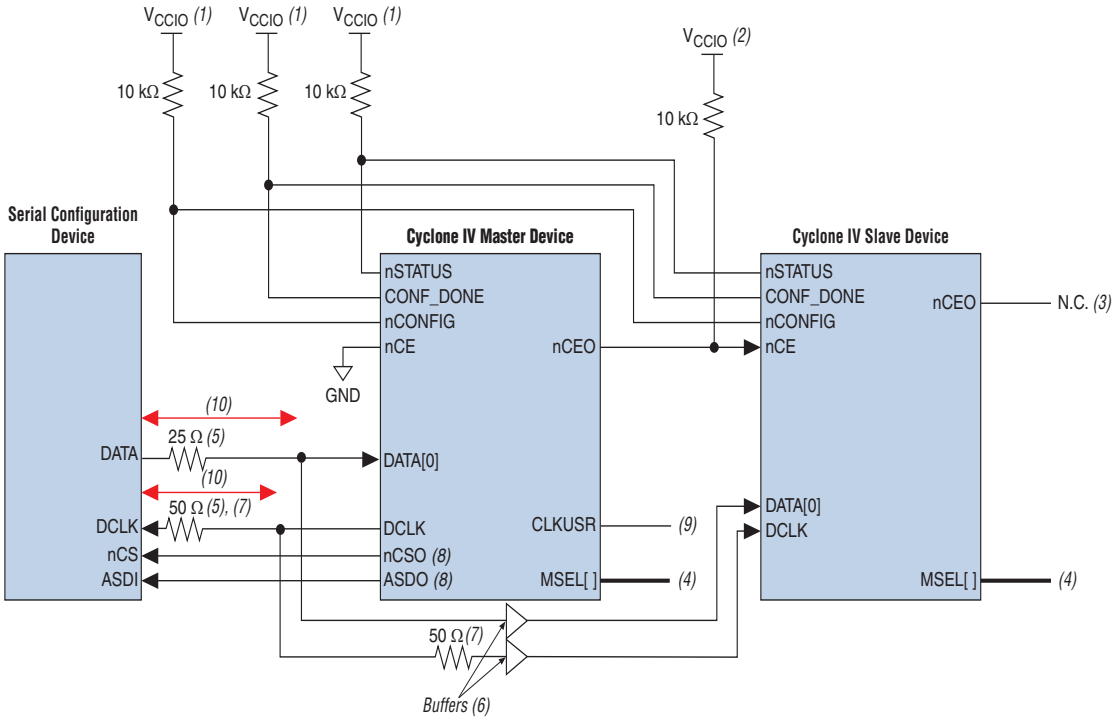
All AS configuration pins (DATA[0], DCLK, nCSO, and DATA[1]) have weak internal pull-up resistors that are always active. After configuration, these pins are set as input tri-stated and are driven high by the weak internal pull-up resistors.

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 shown in Table 8–12 on page 8–36.

## Multi-Device AS Configuration

You can configure multiple Cyclone IV devices with a single serial configuration device. When the first device captures all its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. If the last device in the chain is a Cyclone IV device, you can leave the nCEO pin of the last device unconnected or use it as a user I/O pin after configuration. The nCONFIG, nSTATUS, CONF_DONE, DCLK, and DATA[0] pins of each device in the chain are connected together (Figure 8–3).

**Figure 8–3. Multi-Device AS Configuration**




**Notes to Figure 8–3:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) Connect the pull-up resistor to the $V_{CCIO}$ supply voltage of I/O bank in which the nCE pin resides.

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

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device of the Cyclone IV device in AS mode and the slave devices in PS mode. To connect the MSEL pins for the master device in AS mode and slave devices in PS mode, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(5) Connect the series resistor at the near end of the serial configuration device.

(6) Connect the repeater buffers between the master and slave devices of the Cyclone IV device for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) The 50-Ω series resistors are optional if the 3.3-V configuration voltage standard is applied. For optimal signal integrity, connect these 50-Ω series resistors if the 2.5- or 3.0-V configuration voltage standard is applied.

(8) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(9) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

(10) For multi-devices AS configuration using Cyclone IV E with 1.0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

The first Cyclone IV device in the chain is the configuration master and it controls the configuration of the entire chain. Other Altera devices that support PS configuration can also be part of the chain as configuration slaves.

In the multi-device AS configuration, the board trace length between the serial configuration device and the master device of the Cyclone IV device must follow the recommendations in Table 8–7 on page 8–18.

The nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors, as shown in Figure 8–3 on page 8–13. These pins are open-drain bidirectional pins on the devices. When the first device asserts nCEO (after receiving all its configuration data), it releases its CONF_DONE pin. However, the subsequent devices in the chain keep this shared CONF_DONE line low until they receive their configuration data. When all target devices in the chain receive their configuration data and release CONF_DONE, the pull-up resistor drives a high level on CONF_DONE line and all devices simultaneously enter initialization mode.

Although you can cascade Cyclone IV devices, serial configuration devices cannot be cascaded or chained together.

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

## Configuring Multiple Cyclone IV Devices with the Same Design

Certain designs require that you configure multiple Cyclone IV devices with the same design through a configuration bitstream, or a **.sof**. You can do this through the following methods:

- Multiple **.sof**
- Single **.sof**

For both methods, the serial configuration devices cannot be cascaded or chained together.

### Multiple SRAM Object Files

Two copies of the **.sof** are stored in the serial configuration device. Use the first copy to configure the master device of the Cyclone IV device and the second copy to configure all remaining slave devices concurrently. All slave devices must have the same density and package. The setup is similar to Figure 8–3 on page 8–13.

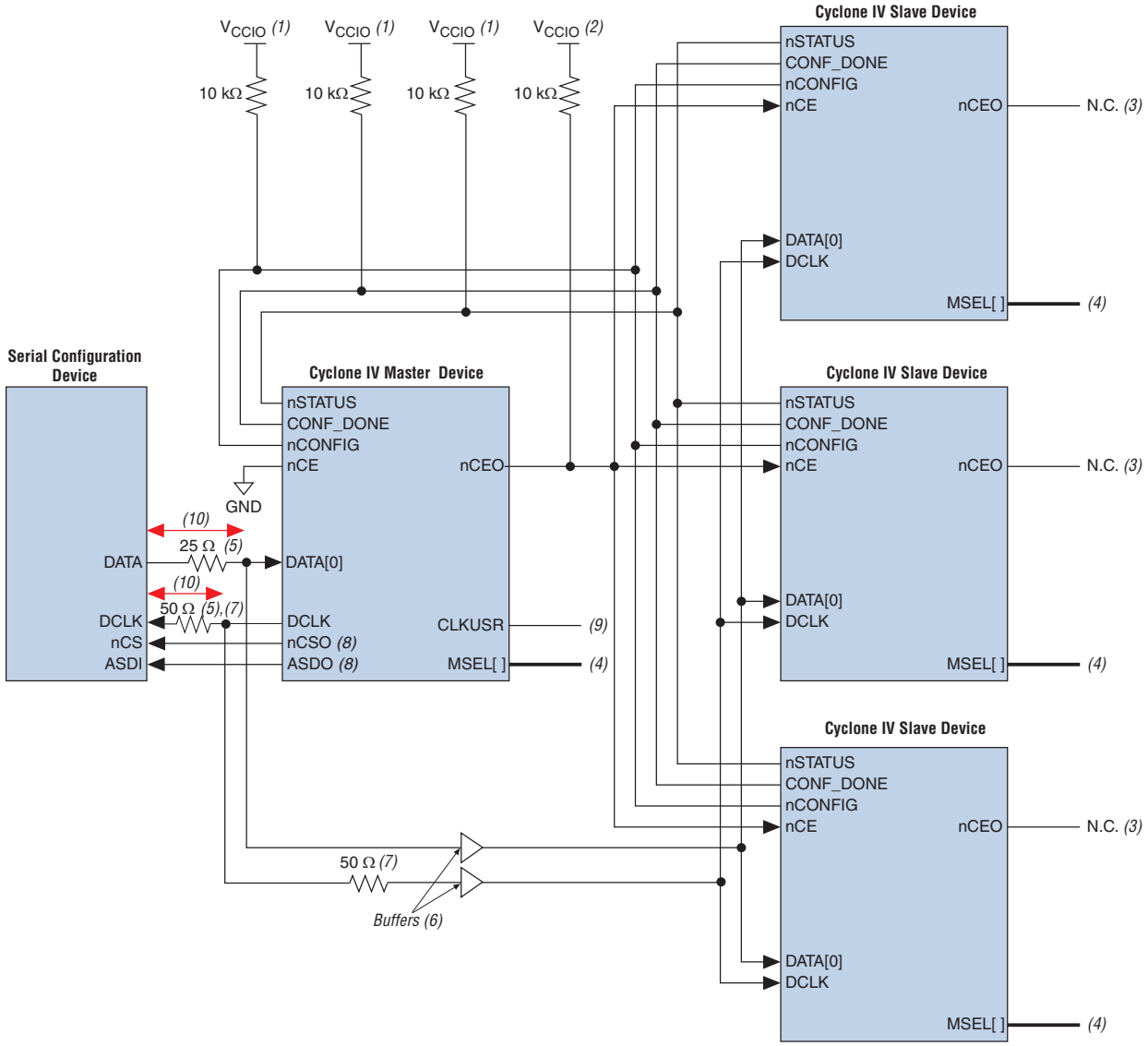
To configure four identical Cyclone IV devices with the same **.sof**, you must set up the chain similar to the example shown in Figure 8–4. The first device is the master device and its MSEL pins must be set to select AS configuration. The other three slave devices are set up for concurrent configuration and their MSEL pins must be set to select PS configuration. The nCEO pin from the master device drives the nCE input pins on all three slave devices, as well as the DATA and DCLK pins that connect in parallel to all

four devices. During the first configuration cycle, the master device reads its configuration data from the serial configuration device while holding nCEO high. After completing its configuration cycle, the master device drives nCE low and sends the second copy of the configuration data to all three slave devices, configuring them simultaneously.

The advantage of the setup in Figure 8–4 is that you can have a different **.sof** for the master device. However, all the slave devices must be configured with the same **.sof**. You can either compress or uncompress the **.sof** in this configuration method.

You can still use this method if the master and slave devices use the same **.sof**.

**Figure 8–4. Multi-Device AS Configuration in Which Devices Receive the Same Data with Multiple .sof**




**Notes to Figure 8–4:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) Connect the pull-up resistor to the $V_{CCIO}$ supply voltage of the I/O bank in which the `nCE` pin resides.

(3) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(4) The `MSEL` pin settings vary for different configuration voltage standards and POR time. You must set the master device in AS mode and the slave devices in PS mode. To connect the `MSEL` pins for the master device in AS mode and the slave devices in PS mode, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the `MSEL` pins directly to $V_{CCA}$ or GND.

(5) Connect the series resistor at the near end of the serial configuration device.

(6) Connect the repeater buffers between the master and slave devices for `DATA[0]` and `DCLK`. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) The 50-Ω series resistors are optional if the 3.3-V configuration voltage standard is applied. For optimal signal integrity, connect these 50-Ω series resistors if the 2.5- or 3.0-V configuration voltage standard is applied.

(8) These pins are dual-purpose I/O pins. The `nCSO` pin functions as `FLASH_nCE` pin in AP mode. The `ASDO` pin functions as `DATA[1]` pin in AP and FPP modes.

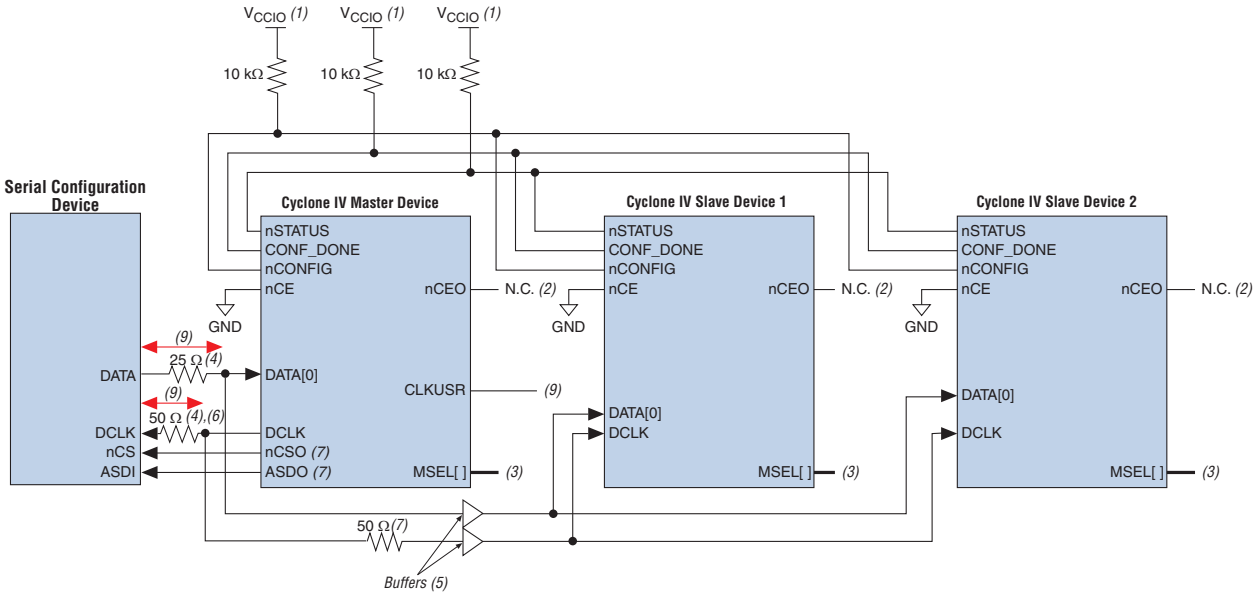
(9) Only Cyclone IV GX devices have an option to select `CLKUSR` (40 MHz maximum) as the external clock source for `DCLK`.

(10) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both `DCLK` and `Data0` line is 3.5 inches.

### Single SRAM Object File

The second method configures both the master device and slave devices with the same **.sof**. The serial configuration device stores one copy of the **.sof**. You must set up one or more slave devices in the chain. All the slave devices must be set up in the same way (Figure 8–5).

**Figure 8–5. Multi-Device AS Configuration in Which Devices Receive the Same Data with a Single .sof**




**Notes to Figure 8–5:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device of the Cyclone IV device in AS mode and the slave devices in PS mode. To connect the MSEL pins for the master device in AS mode and slave devices in PS mode, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(4) Connect the series resistor at the near end of the serial configuration device.

(5) Connect the repeater buffers between the master and slave devices for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(6) The 50-Ω series resistors are optional if the 3.3-V configuration voltage standard is applied. For optimal signal integrity, connect these 50-Ω series resistors if the 2.5- or 3.0-V configuration voltage standard is applied.

(7) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(8) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

(9) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

In this setup, all the Cyclone IV devices in the chain are connected for concurrent configuration. This reduces the AS configuration time because all the Cyclone IV devices are configured in one configuration cycle. Connect the nCE input pins of all the Cyclone IV devices to GND. You can either leave the nCEO output pins on all the Cyclone IV devices unconnected or use the nCEO output pins as normal user I/O pins. The DATA and DCLK pins are connected in parallel to all the Cyclone IV devices.

Altera recommends putting a buffer before the DATA and DCLK output from the master device to avoid signal strength and signal integrity issues. The buffer must not significantly change the DATA-to-DCLK relationships or delay them with respect to other AS signals (ASDI and nCS). Also, the buffer must only drive the slave devices to ensure that the timing between the master device and the serial configuration device is unaffected.

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

## Guidelines for Connecting a Serial Configuration Device to Cyclone IV Devices for an AS Interface

For single- and multi-device AS configurations, the board trace length and loading between the supported serial configuration device and Cyclone IV device must follow the recommendations listed in Table 8–7.

**Table 8–7. Maximum Trace Length and Loading for AS Configuration**

| Cyclone IV Device AS Pins | Maximum Board Trace Length from a Cyclone IV Device to a Serial Configuration Device (Inches) | | Maximum Board Load (pF) |
|---|---|---|---|
| | Cyclone IV E | Cyclone IV GX | |
| DCLK | 10 | 6 | 15 |
| DATA[0] | 10 | 6 | 30 |
| nCSO | 10 | 6 | 30 |
| ASDO | 10 | 6 | 30 |

**Note to Table 8–7:**

(1) For multi-devices AS configuration using Cyclone IV E with 1,0 V core voltage, the maximum board trace-length from the serial configuration device to the junction-split on both DCLK and Data0 line is 3.5 inches.

### Estimating AS Configuration Time

AS configuration time is dominated by the time it takes to transfer data from the serial configuration device to the Cyclone IV device. This serial interface is clocked by the Cyclone IV device DCLK output (generated from a 40-MHz internal oscillator for Cyclone IV E devices, a 20- or 40-MHz internal oscillator, or an external CLKUSR of up to 40 MHz for Cyclone IV GX devices).

Equation 8–2 and Equation 8–3 show the configuration time calculations.

**Equation 8–2.**

$$\text{Size} \times \left( \frac{\text{maximum DCLK period}}{\text{1 bit}} \right) = \text{estimated maximum configuration ti}$$

**Equation 8–3.**

$$9{,}600{,}000 \text{ bits} \times \left( \frac{50 \text{ ns}}{\text{1 bit}} \right) = 480 \text{ ms}$$

Table 8–8 provides the configuration time for AS configuration.

**Table 8–8. AS Configuration Time for Cyclone IV Devices** *(1)*

| Symbol | Parameter | Cyclone IV E | Cyclone IV GX | Unit |
|---|---|---|---|---|
| $t_{SU}$ | Setup time | 10 | 8 | ns |
| $t_H$ | Hold time | 0 | 0 | ns |
| $t_{CO}$ | Clock-to-output time | 4 | 4 | ns |

**Note to Table 8–8:**

(1) For the AS configuration timing diagram, refer to the *Serial Configuration (EPCS) Devices Datasheet*.

Enabling compression reduces the amount of configuration data that is sent to the Cyclone IV device, which also reduces configuration time. On average, compression reduces configuration time by 50%.

## Programming Serial Configuration Devices

Serial configuration devices are non-volatile, flash memory-based devices. You can program these devices in-system with the USB-Blaster™ or ByteBlaster™ II download cables. Alternatively, you can program them with 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 through the AS programming interface. During in-system programming, the download cable disables device access to the AS interface by driving the nCE pin high. Cyclone 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 $V_{CC}$ and GND, respectively.

To perform in-system programming of a serial configuration device through the AS programming interface, you must place the diodes and capacitors as close as possible to the Cyclone IV device. You must ensure that the diodes and capacitors maintain a maximum AC voltage of 4.1 V (Figure 8–6).

If you want to use the setup shown in Figure 8–6 to perform in-system programming of a serial configuration device and single- or multi-device AS configuration, you do not require a series resistor on the DATA line at the near end of the serial configuration device. The existing diodes and capacitors are sufficient.

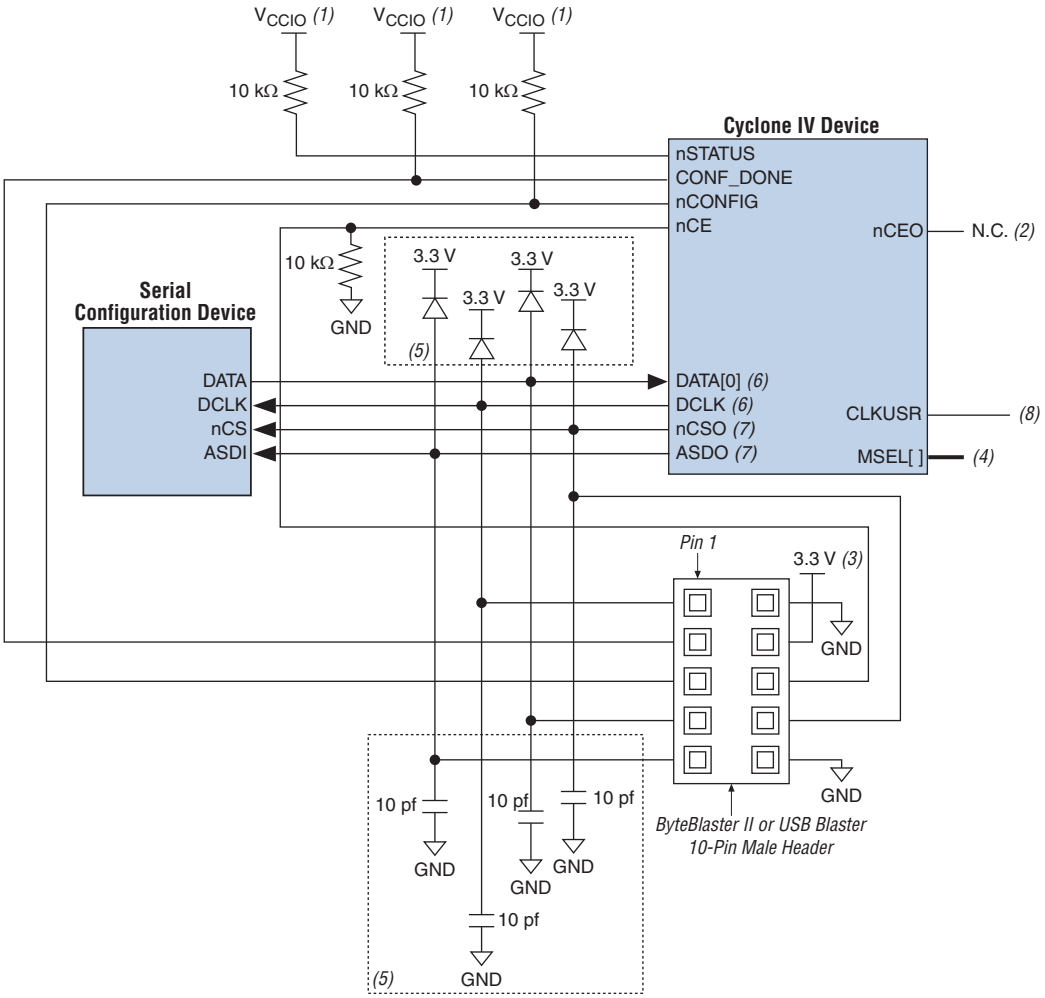
Altera has developed the Serial FlashLoader (SFL), a JTAG-based in-system programming solution for Altera serial configuration devices. The SFL is a bridge design for the Cyclone IV device that uses its JTAG interface to access the EPCS JIC (JTAG Indirect Configuration Device Programming) file and then uses the AS interface to program the EPCS device. Both the JTAG interface and AS interface are bridged together inside the SFL design.

For more information about implementing the SFL with Cyclone IV devices, refer to *AN 370: Using the Serial FlashLoader with the 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 download cable, refer to the *ByteBlaster II Download Cable User Guide*.

Figure 8–6 shows the download cable connections to the serial configuration device.

**Figure 8–6. In-System Programming of Serial Configuration Devices**




**Notes to Figure 8–6:**

(1) Connect these pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) Power up the $V_{CC}$ of the ByteBlaster II or USB-Blaster download cable with the 3.3-V supply.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(5) The diodes and capacitors must be placed as close as possible to the Cyclone IV device. You must ensure that the diodes and capacitors maintain a maximum AC voltage of 4.1 V. The external diodes and capacitors are required to prevent damage to the Cyclone IV device AS configuration input pins due to possible overshoot when programming the serial configuration device with a download cable. Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes, for effective voltage clamping.

(6) When cascading Cyclone IV devices in a multi-device AS configuration, connect the repeater buffers between the master and slave devices for DATA[0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(8) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

You can use the Quartus II software with the APU and the appropriate configuration device programming adapter to program serial configuration devices. All serial configuration devices are offered in an 8- or 16-pin small outline integrated circuit (SOIC) package.

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

A serial configuration device is programmed in-system by an external microprocessor with the SRunner software driver. The SRunner software driver is a software driver developed for embedded serial configuration device programming, which is easily customized to fit in different embedded systems. The SRunner software driver is able to read a Raw Programming Data (**.rpd**) file and write to serial configuration devices. The serial configuration device programming time, using the SRunner software driver, is comparable to the programming time with the Quartus II software.

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

## AP Configuration (Supported Flash Memories)

The AP configuration scheme is only supported in Cyclone IV E devices. In the AP configuration scheme, Cyclone IV E devices are configured using commodity 16-bit parallel flash memory. These external non-volatile configuration devices are industry standard microprocessor flash memories. The flash memories provide a fast interface to access configuration data. The speed up in configuration time is mainly due to the 16-bit wide parallel data bus, which is used to retrieve data from the flash memory.

Some of the smaller Cyclone IV E devices or package options do not support the AP configuration scheme. Table 8–9 lists the supported AP configuration scheme for each Cyclone IV E devices.

**Table 8–9. Supported AP Configuration Scheme for Cyclone IV E Devices**

| Device | Package Options | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | E144 | M164 | M256 | U256 | F256 | F324 | U484 | F484 | F780 |
| EP4CE6 | — | — | — | — | — | — | — | — | — |
| EP4CE10 | — | — | — | — | — | — | — | — | — |
| EP4CE15 | — | — | — | — | — | — | — | ✓ | — |
| EP4CE22 | — | — | — | — | — | — | — | — | — |
| EP4CE30 | — | — | — | — | — | ✓ | — | ✓ | ✓ |
| EP4CE40 | — | — | — | — | — | ✓ | ✓ | ✓ | ✓ |
| EP4CE55 | — | — | — | — | — | — | ✓ | ✓ | ✓ |
| EP4CE75 | — | — | — | — | — | — | ✓ | ✓ | ✓ |
| EP4CE115 | — | — | — | — | — | — | — | ✓ | ✓ |

During device configuration, Cyclone IV E devices read configuration data using the parallel interface and configure their SRAM cells. This scheme is referred to as the AP configuration scheme because the device controls the configuration interface. This scheme contrasts with the FPP configuration scheme, where an external host controls the interface.

## AP Configuration Supported Flash Memories

The AP configuration controller in Cyclone IV E devices is designed to interface with two industry-standard flash families—the Micron P30 Parallel NOR flash family and the Micron P33 Parallel NOR flash family. Unlike serial configuration devices, both of the flash families supported in AP configuration scheme are designed to interface with microprocessors. By configuring from an industry standard microprocessor flash which allows access to the flash after entering user mode, the AP configuration scheme allows you to combine configuration data and user data (microprocessor boot code) on the same flash memory.

The Micron P30 flash family and the P33 flash family support a continuous synchronous burst read mode at 40 MHz DCLK frequency for reading data from the flash. Additionally, the Micron P30 and P33 flash families have identical pin-out and adopt similar protocols for data access.

Cyclone IV E devices use a 40-MHz oscillator for the AP configuration scheme. The oscillator is the same oscillator used in the Cyclone IV E AS configuration scheme.

Table 8–10 lists the supported families of the commodity parallel flash for the AP configuration scheme.

**Table 8–10. Supported Commodity Flash for AP Configuration Scheme for Cyclone IV E Devices** *(1)*

| Flash Memory Density | Micron P30 Flash Family *(2)* | Micron P33 Flash Family *(3)* |
|---|---|---|
| 64 Mbit | ✓ | ✓ |
| 128 Mbit | ✓ | ✓ |
| 256 Mbit | ✓ | ✓ |

**Notes to Table 8–10:**

(1) The AP configuration scheme only supports flash memory speed grades of 40 MHz and above.
(2) 3.3- , 3.0-, 2.5-, and 1.8-V I/O options are supported for the Micron P30 flash family.
(3) 3.3-, 3.0- and 2.5-V I/O options are supported for the Micron P33 flash family.

Configuring Cyclone IV E devices from the Micron P30 and P33 family 512-Mbit flash memory is possible, but you must properly drive the extra address and FLASH_nCE pins as required by these flash memories.

To check for supported speed grades and package options, refer to the respective flash datasheets.

The AP configuration scheme in Cyclone IV E devices supports flash speed grades of 40 MHz and above. However, AP configuration for all these speed grades must be capped at 40 MHz. The advantage of faster speed grades is realized when your design in the Cyclone IV E devices accesses flash memory in user mode.

For more information about the operation of the Micron P30 Parallel NOR and P33 Parallel NOR flash memories, search for the keyword "P30" or "P33" on the Micron website (www.micron.com) to obtain the P30 or P33 family datasheet.

## Single-Device AP Configuration

The following groups of interface pins are supported in Micron P30 and P33 flash memories:

- Control pins
- Address pins
- Data pins

The following control signals are from the supported parallel flash memories:

- CLK
- active-low reset (RST#)
- active-low chip enable (CE#)
- active-low output enable (OE#)
- active-low address valid (ADV#)
- active-low write enable (WE#)

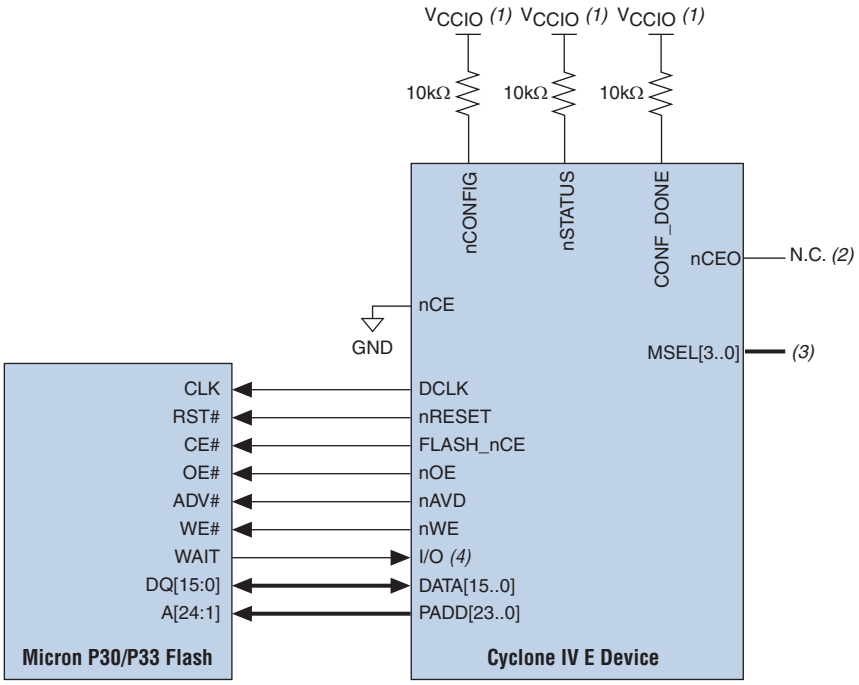
The supported parallel flash memories output a control signal (WAIT) to Cyclone IV E devices to indicate when synchronous data is ready on the data bus. Cyclone IV E devices have a 24-bit address bus connecting to the address bus (A[24:1]) of the flash memory. A 16-bit bidirectional data bus (DATA[15..0]) provides data transfer between the Cyclone IV E device and the flash memory.

The following control signals are from the Cyclone IV E device to flash memory:

- DCLK
- active-low hard rest (nRESET)
- active-low chip enable (FLASH_nCE)
- active-low output enable for the DATA[15..0] bus and WAIT pin (nOE)
- active-low address valid signal and is used to write data into the flash (nAVD)
- active-low write enable and is used to write data into the flash (nWE)

Figure 8–7 shows the interface for the Micron P30 flash memory and P33 flash memory to the Cyclone IV E device pins.

**Figure 8–7. Single-Device AP Configuration Using Micron P30 and P33 Flash Memory**




**Notes to Figure 8–7:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL[3..0], refer to Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(4) AP configuration ignores the WAIT signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

To tri-state the configuration bus for AP configuration schemes, you must tie nCE high and nCONFIG low.

In a single-device AP configuration, the maximum board loading and board trace length between supported parallel flash and Cyclone IV E devices must follow the recommendations listed in Table 8–11 on page 8–28.

If you use the AP configuration scheme for Cyclone IV E devices, the $V_{CCIO}$ of I/O banks 1, 6, 7, and 8 must be 3.3, 3.0, 2.5, or 1.8 V. Altera does not recommend using the level shifter between the Micron P30 or P33 flash and the Cyclone IV E device in the AP configuration scheme.

There are no series resistors required in AP configuration mode for Cyclone IV E devices when using the Micron flash at 2.5-, 3.0-, and 3.3-V I/O standard. The output buffer of the Micron P30 IBIS model does not overshoot above 4.1 V. Thus, series resistors are not required for the 2.5-, 3.0-, and 3.3-V AP configuration option. However, if there are any other devices sharing the same flash I/Os with Cyclone IV E devices, all shared pins are still subject to the 4.1-V limit and may require series resistors.

Default read mode of the supported parallel flash memory and all writes to the parallel flash memory are asynchronous. Both the parallel flash families support a synchronous read mode, with data supplied on the positive edge of DCLK.

The serial clock (DCLK) generated by Cyclone IV E devices controls the entire configuration cycle and provides timing for the parallel interface.

## Multi-Device AP Configuration

You can configure multiple Cyclone IV E devices using a single parallel flash. You can cascade multiple Cyclone IV E 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. Use an external 10-kΩ pull-up resistor to pull the nCEO signal high to its $V_{CCIO}$ level to help the internal weak pull-up resistor. When the first device captures all its configuration data from the bitstream, it drives the nCEO pin low, enabling the next device in the chain. You can leave the nCEO pin of the last device unconnected or use it as a user I/O pin after configuration if the last device in the chain is a Cyclone IV E device. The nCONFIG, nSTATUS, CONF_DONE, DCLK, DATA[15..8], and DATA[7..0] pins of each device in the chain are connected (Figure 8–8 on page 8–26 and Figure 8–9 on page 8–27).

The first Cyclone IV E device in the chain, as shown in Figure 8–8 on page 8–26 and Figure 8–9 on page 8–27, is the configuration master device and controls the configuration of the entire chain. You must connect its MSEL pins to select the AP configuration scheme. The remaining Cyclone IV E devices are used as configuration slaves. You must connect their MSEL pins to select the FPP configuration scheme. Any other Altera device that supports FPP configuration can also be part of the chain as a configuration slave.

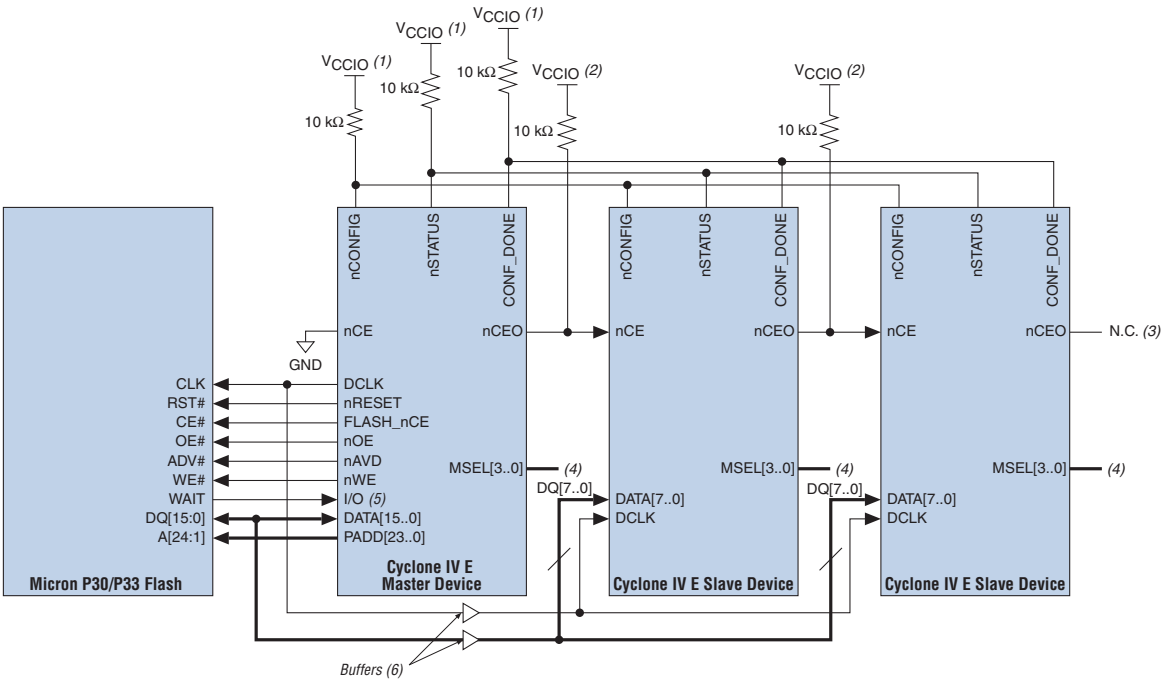
The following are the configurations for the DATA[15..0] bus in a multi-device AP configuration:

- Byte-wide multi-device AP configuration
- Word-wide multi-device AP configuration

## Byte-Wide Multi-Device AP Configuration

The simpler method for multi-device AP configuration is the byte-wide multi-device AP configuration. In the byte-wide multi-device AP configuration, the LSB of the DATA[7..0] pin from the flash and master device (set to the AP configuration scheme) is connected to the slave devices set to the FPP configuration scheme, as shown in Figure 8–8.

**Figure 8–8. Byte-Wide Multi-Device AP Configuration**




**Notes to Figure 8–8:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) Connect the pull-up resistor to the $V_{CCIO}$ supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. You must set the master device in AP mode and the slave devices in FPP mode. To connect MSEL[3..0] for the master device in AP mode and the slave devices in FPP mode, refer to Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

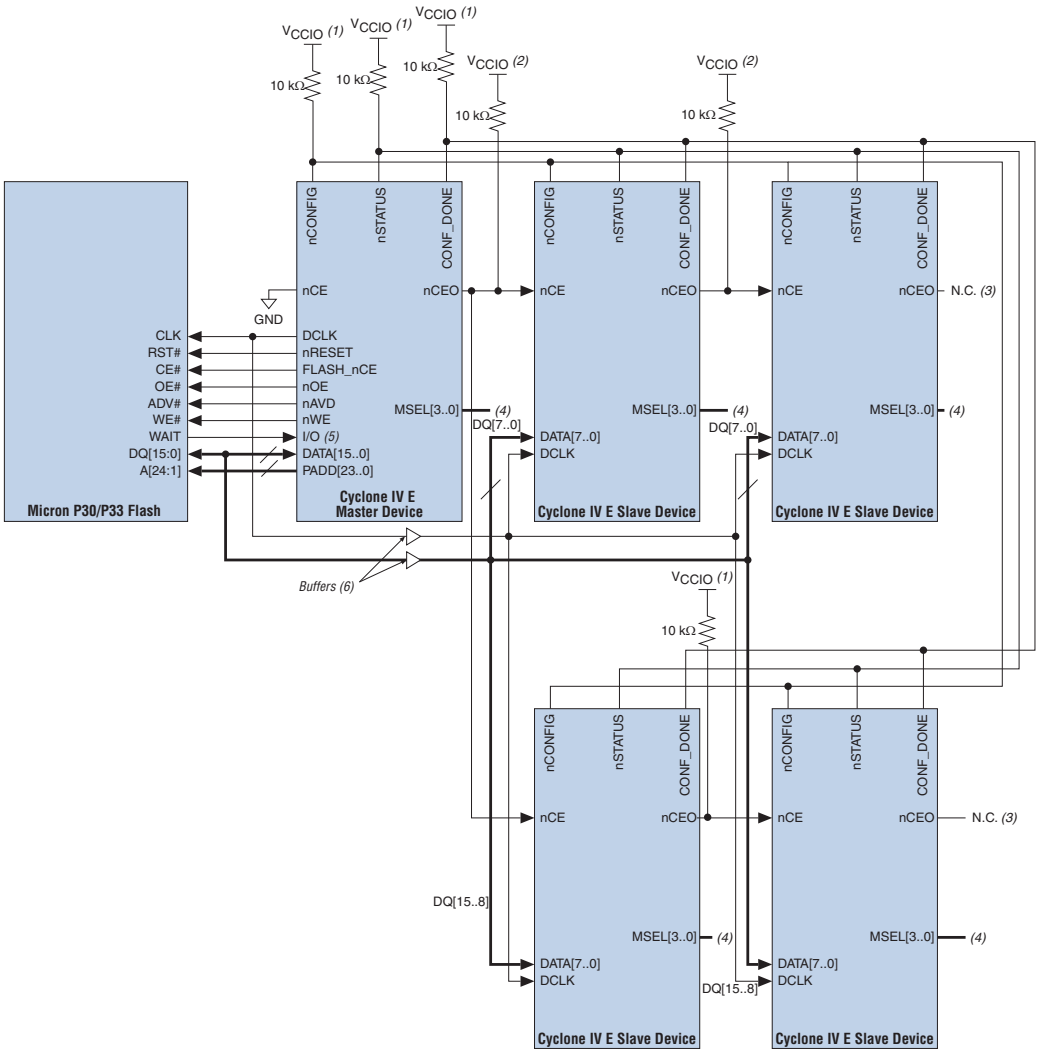
(5) The AP configuration ignores the WAIT signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

(6) Connect the repeater buffers between the Cyclone IV E master device and slave devices for DATA[15..0] and DCLK. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

## Word-Wide Multi-Device AP Configuration

The more efficient setup is one in which some of the slave devices are connected to the LSB of the DATA[7..0] and the remaining slave devices are connected to the MSB of the DATA[15..8]. In the word-wide multi-device AP configuration, the nCEO pin of the master device enables two separate daisy chains of slave devices, allowing both chains to be programmed concurrently, as shown in Figure 8–9.

**Figure 8–9. Word-Wide Multi-Device AP Configuration**




**Notes to Figure 8–9:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) Connect the pull-up resistor to the $V_{CCIO}$ supply voltage of the I/O bank in which the `nCE` pin resides.

(3) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(4) The `MSEL` pin settings vary for different configuration voltage standards and POR time. You must set the master device in AP mode and the slave devices in FPP mode. To connect `MSEL[3..0]` for the master device in AP mode and the slave devices in FPP mode, refer to Table 8–5 on page 8–9. Connect the `MSEL` pins directly to $V_{CCA}$ or GND.

(5) The AP configuration ignores the `WAIT` signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O pin to monitor the `WAIT` signal from the Micron P30 or P33 flash.

(6) Connect the repeater buffers between the Cyclone IV E master device and slave devices for `DATA[15..0]` and `DCLK`. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

☞ In a multi-device AP configuration, the board trace length between the parallel flash and the master device must follow the recommendations listed in Table 8–11.

The nSTATUS and CONF_DONE pins on all target devices are connected together with external pull-up resistors, as shown in Figure 8–8 on page 8–26 and Figure 8–9 on page 8–27. These pins are open-drain bidirectional pins on the devices. When the first device asserts nCEO (after receiving all its configuration data), it releases its CONF_DONE pin. However, the subsequent devices in the chain keep this shared CONF_DONE line low until they receive their configuration data. When all target devices in the chain receive their configuration data and release CONF_DONE, the pull-up resistor drives a high level on this line and all devices simultaneously enter initialization mode.

## Guidelines for Connecting Parallel Flash to Cyclone IV E Devices for an AP Interface

For single- and multi-device AP configuration, the board trace length and loading between the supported parallel flash and Cyclone IV E devices must follow the recommendations listed in Table 8–11. These recommendations also apply to an AP configuration with multiple bus masters.

**Table 8–11. Maximum Trace Length and Loading for AP Configuration**

| Cyclone IV E AP Pins | Maximum Board Trace Length from Cyclone IV E Device to Flash Device (inches) | Maximum Board Load (pF) |
|---|---|---|
| DCLK | 6 | 15 |
| DATA[15..0] | 6 | 30 |
| PADD[23..0] | 6 | 30 |
| nRESET | 6 | 30 |
| Flash_nCE | 6 | 30 |
| nOE | 6 | 30 |
| nAVD | 6 | 30 |
| nWE | 6 | 30 |
| I/O (1) | 6 | 30 |

**Note to Table 8–11:**

(1) The AP configuration ignores the WAIT signal from the flash during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the WAIT signal from the Micron P30 or P33 flash.

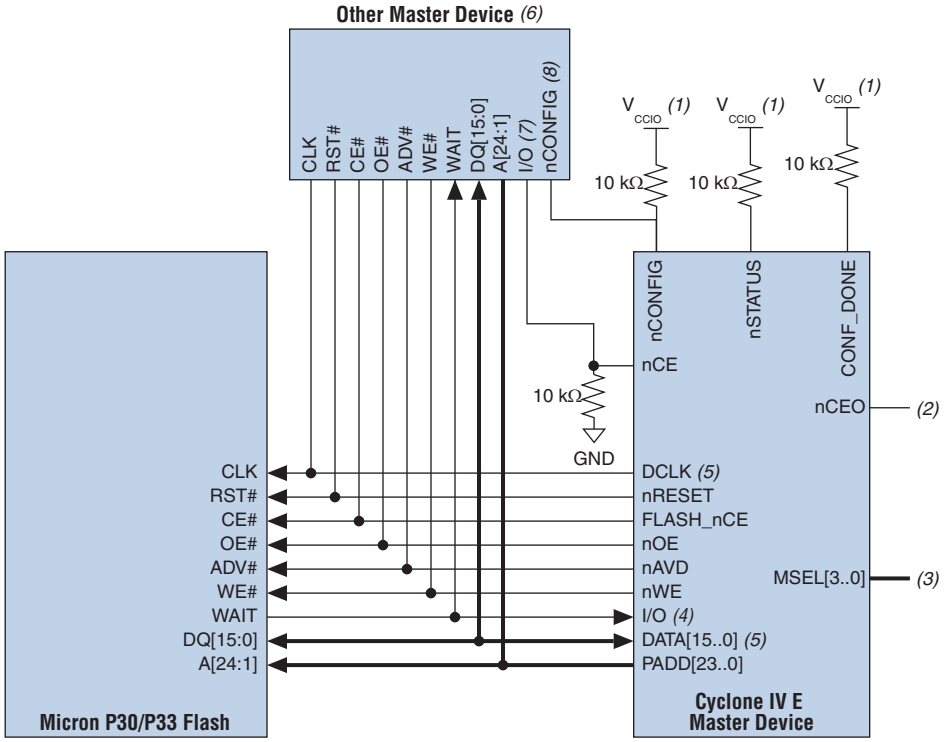
## Configuring With Multiple Bus Masters

Similar to the AS configuration scheme, the AP configuration scheme supports multiple bus masters for the parallel flash. For another master to take control of the AP configuration bus, the master must assert nCONFIG low for at least 500 ns to reset the master Cyclone IV E device and override the weak 10-kΩ pull-down resistor on the nCE pin. This resets the master Cyclone IV E device and causes it to tri-state its AP configuration bus. The other master device then takes control of the AP configuration bus. After the other master device is done, it releases the AP configuration bus, then releases the nCE pin, and finally pulses nCONFIG low to restart the configuration.

In the AP configuration scheme, multiple masters share the parallel flash. Similar to the AS configuration scheme, the bus control is negotiated by the nCE pin.

Figure 8–10 shows the AP configuration with multiple bus masters.

**Figure 8–10. AP Configuration with Multiple Bus Masters**




**Notes to Figure 8–10:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(3) The `MSEL` pin settings vary for different configuration voltage standards and POR time. To connect `MSEL[3..0]`, refer to Table 8–5 on page 8–9. Connect the `MSEL` pins directly to $V_{CCA}$ or GND.

(4) The AP configuration ignores the `WAIT` signal during configuration mode. However, if you are accessing flash during user mode with user logic, you can optionally use the normal I/O to monitor the `WAIT` signal from the Micron P30 or P33 flash.

(5) When cascading Cyclone IV E devices in a multi-device AP configuration, connect the repeater buffers between the master device and slave devices for `DATA[15..0]` and `DCLK`. All I/O inputs must maintain a maximum AC voltage of 4.1 V. The output resistance of the repeater buffers must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

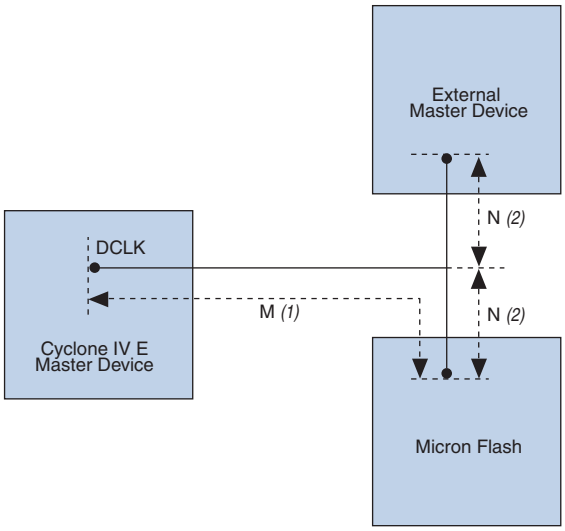
(6) The other master device must fit the maximum overshoot equation outlined in "Configuration and JTAG Pin I/O Requirements" on page 8–5.

(7) The other master device can control the AP configuration bus by driving the nCE to high with an output high on the I/O pin.

(8) The other master device can pulse `nCONFIG` if it is under system control and not tied to $V_{CCIO}$.

Figure 8–11 shows the recommended balanced star routing for multiple bus master interfaces to minimize signal integrity issues.

**Figure 8–11. Balanced Star Routing**




**Notes to Figure 8–11:**

(1) Altera recommends that *M* does not exceed 6 inches, as listed in Table 8–11 on page 8–28.

(2) Altera recommends using a balanced star routing. Keep the *N* length equal and as short as possible to minimize reflection noise from the transmission line. The *M* length is applicable for this setup.

## Estimating AP Configuration Time

AP configuration time is dominated by the time it takes to transfer data from the parallel flash to Cyclone IV E devices. This parallel interface is clocked by the Cyclone IV E `DCLK` output (generated from an internal oscillator). The `DCLK` minimum frequency when using the 40-MHz oscillator is 20 MHz (50 ns). In word-wide cascade programming, the `DATA[15..0]` bus transfers a 16-bit word and essentially cuts configuration time to approximately 1/16 of the AS configuration time. Equation 8–4 and Equation 8–5 show the configuration time calculations.

**Equation 8–4.**

$$\text{Size} \times \left( \frac{\text{maximum DCLK period}}{\text{16 bits per DCLK cycle}} \right) = \text{estimated maximum configuration time}$$

**Equation 8–5.**

$$9{,}600{,}000 \text{ bits} \times \left( \frac{50 \text{ ns}}{16 \text{ bit}} \right) = 30 \text{ ms}$$

## Programming Parallel Flash Memories

Supported parallel flash memories are external non-volatile configuration devices. They are industry standard microprocessor flash memories. For more information about the supported families for the commodity parallel flash, refer to Table 8–10 on page 8–22.

Cyclone IV E devices in a single- or multiple-device chain support in-system programming of a parallel flash using the JTAG interface with the flash loader megafunction. The board intelligent host or download cable uses the four JTAG pins on Cyclone IV E devices to program the parallel flash in system, even if the host or download cable cannot access the configuration pins of the parallel flash.

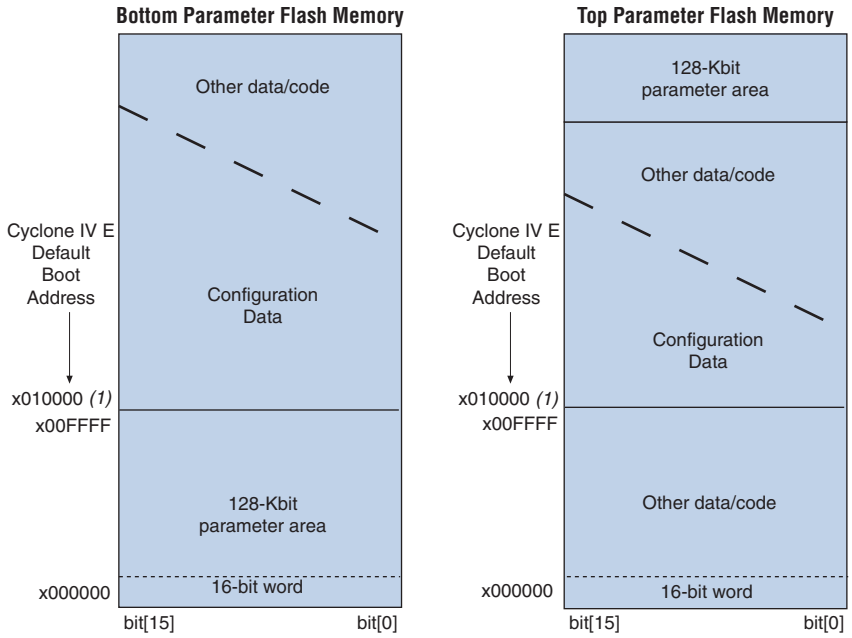
For more information about using the JTAG pins on Cyclone IV E devices to program the parallel flash in-system, refer to *AN 478: Using FPGA-Based Parallel Flash Loader (PFL) with the Quartus II Software*.

In the AP configuration scheme, the default configuration boot address is 0×010000 when represented in 16-bit word addressing in the supported parallel flash memory (Figure 8–12). In the Quartus II software, the default configuration boot address is 0×020000 because it is represented in 8-bit byte addressing. Cyclone IV E devices configure from word address 0×010000, which is equivalent to byte address 0×020000.

The Quartus II software uses byte addressing for the default configuration boot address. You must set the start address field to **0×020000**.

The default configuration boot address allows the system to use special parameter blocks in the flash memory map. Parameter blocks are at the top or bottom of the memory map. Figure 8–12 shows the configuration boot address in the AP configuration scheme. You can change the default configuration default boot address 0×010000 to any desired address using the APFC_BOOT_ADDR JTAG instruction. For more information about the APFC_BOOT_ADDR JTAG instruction, refer to "JTAG Instructions" on page 8–57.

**Figure 8–12. Configuration Boot Address in AP Flash Memory Map**




**Note to Figure 8–12:**

(1) The default configuration boot address is x010000 when represented in 16-bit word addressing.

## PS Configuration

You can perform PS configuration on Cyclone IV devices with an external intelligent host, such as a MAX® II device, microprocessor with flash memory, or a download cable. In the PS scheme, an external host controls the configuration. Configuration data is clocked into the target Cyclone IV device through DATA[0] at each rising edge of DCLK.

If your system already contains a common flash interface (CFI) flash memory, you can use it for Cyclone IV device configuration storage as well. The MAX II PFL feature provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control the configuration from the flash memory device to the Cyclone IV device.

For more information about the PFL, refer to *AN 386: Using the Parallel Flash Loader with the Quartus II Software*.

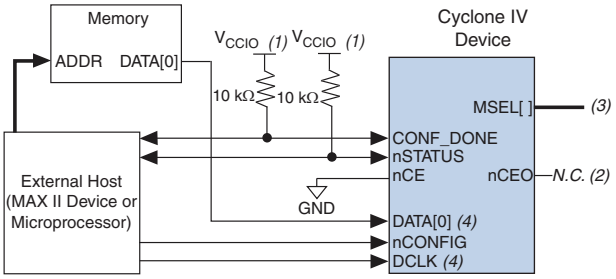
Cyclone IV devices do not support enhanced configuration devices for PS configuration.

## PS Configuration Using an External Host

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

Figure 8–13 shows the configuration interface connections between a Cyclone IV device and an external host device for single-device configuration.

**Figure 8–13. Single-Device PS Configuration Using an External Host**




**Notes to Figure 8–13:**

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device. $V_{CC}$ must be high enough to meet the $V_{IH}$ specification of the I/O on the device and the external host.

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

To begin the configuration, the external host device must generate a low-to-high transition on the nCONFIG pin. When nSTATUS is pulled high, the external host device must place the configuration data one bit at a time on DATA[0]. If you use configuration data in **.rbf**, **.ttf**, or **.hex**, you must first send the LSB of each data byte. For example, if the **.rbf** contains the byte sequence 02 1B EE 01 FA, the serial bitstream you must send to the device is:

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

Cyclone IV devices receive configuration data on DATA[0] and the clock is received on DCLK. 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 and the device enters initialization state.

Two DCLK falling edges are required after CONF_DONE goes high to begin the initialization of the device.

INIT_DONE is released and pulled high when initialization is complete. The external host 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.

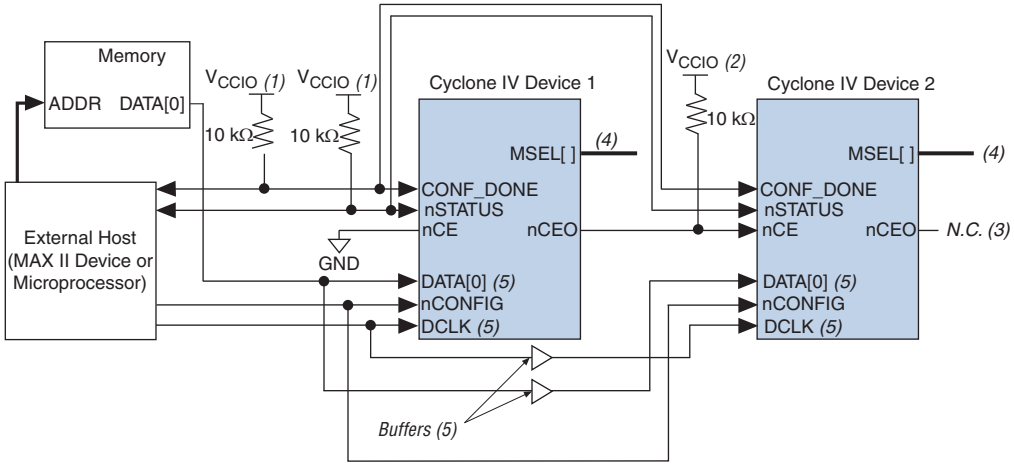
To ensure DCLK and DATA[0] are not left floating at the end of configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. In the PS scheme, the DATA[0] pin is tri-stated by default in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

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

Figure 8–14 shows how to configure multiple devices using an external host device. This circuit is similar to the PS configuration circuit for a single device, except that Cyclone IV devices are cascaded for multi-device configuration.

**Figure 8–14. Multi-Device PS Configuration Using an External Host**




**Notes to Figure 8–14:**

(1) The pull-up resistor must be connected to a supply that provides an acceptable input signal for all devices in the chain. $V_{CC}$ must be high enough to meet the $V_{IH}$ specification of the I/O on the device and the external host.

(2) Connect the pull-up resistor to the $V_{CCIO}$ supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(5) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

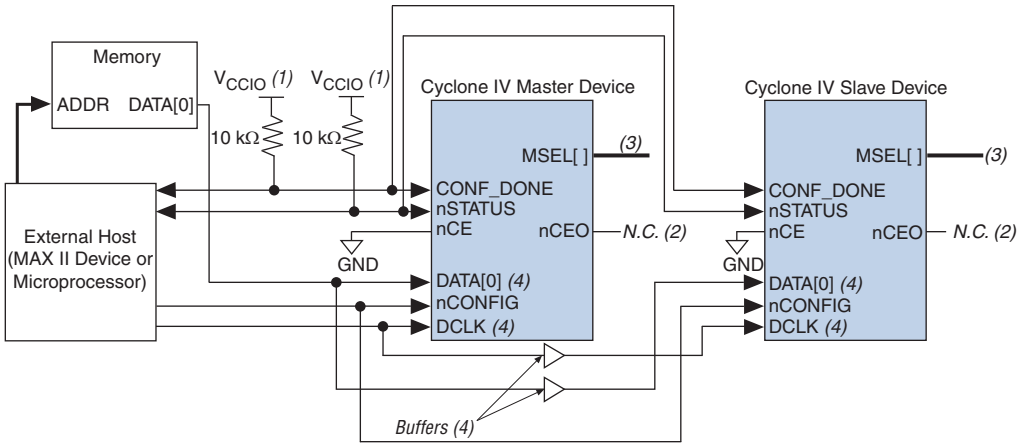
After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle. Therefore, the transfer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

You can have multiple devices that contain the same configuration data in your system. To support this configuration scheme, all device nCE inputs are tied to GND, while the nCEO pins are left floating. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE configuration pins are connected to every device in the chain. To ensure signal integrity and prevent clock skew problems, configuration signals may require buffering. Ensure that the DCLK and DATA lines are buffered. Devices must be of the same density and package. All devices start and complete configuration at the same time.

Figure 8–15 shows a multi-device PS configuration when both Cyclone IV devices are receiving the same configuration data.

**Figure 8–15. Multi-Device PS Configuration When Both Devices Receive the Same Data**




**Notes to Figure 8–15:**

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. $V_{CC}$ must be high enough to meet the $V_{IH}$ specification of the I/O on the device and the external host.

(2) The nCEO pins of both devices are left unconnected or used as user I/O pins when configuring the same configuration data into multiple devices.

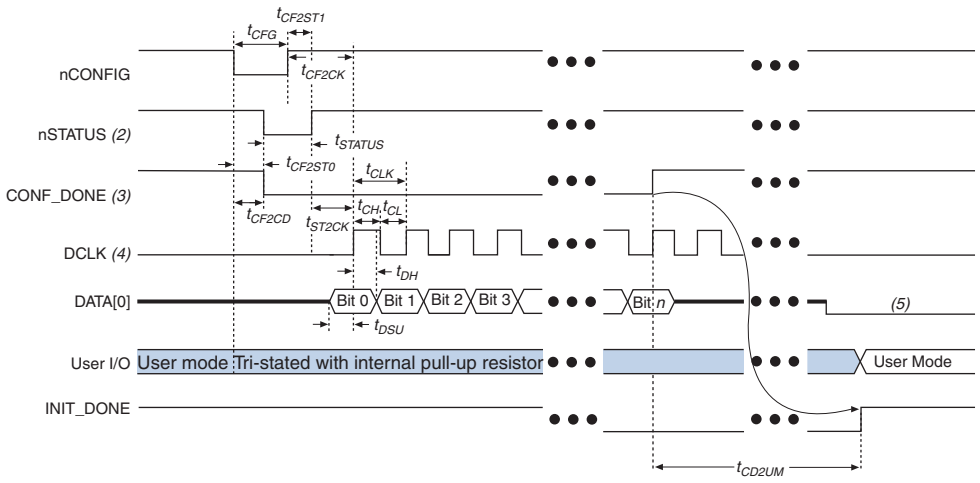
(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

## PS Configuration Timing

A PS configuration must meet the setup and hold timing parameters and the maximum clock frequency. When using a microprocessor or another intelligent host to control the PS interface, ensure that you meet these timing requirements. Figure 8–16 shows the timing waveform for PS configuration when using an external host device.

**Figure 8–16. PS Configuration Timing Waveform** *(1)*




**Notes to Figure 8–16:**

(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 Cyclone IV device holds nSTATUS low during POR delay.

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

(4) In user mode, drive DCLK either high or low when using the PS configuration scheme, whichever is more convenient. When using the AS configuration scheme, DCLK is a Cyclone IV device output pin and must not be driven externally.

(5) Do not leave the DATA[0] pin floating after configuration. Drive the DATA[0] pin high or low, whichever is more convenient.

Table 8–12 lists the PS configuration timing parameters for Cyclone IV devices.

**Table 8–12. PS Configuration Timing Parameters For Cyclone IV Devices (Part 1 of 2)**

| Symbol | Parameter | Minimum | | Maximum | | Unit |
|---|---|---|---|---|---|---|
| | | Cyclone IV *(1)* | Cyclone IV E *(2)* | Cyclone IV *(1)* | Cyclone IV E *(2)* | |
| $t_{CF2CD}$ | nCONFIG low to CONF_DONE low | — | | 500 | | ns |
| $t_{CF2ST0}$ | nCONFIG low to nSTATUS low | — | | 500 | | ns |
| $t_{CFG}$ | nCONFIG low pulse width | 500 | | — | | ns |
| $t_{STATUS}$ | nSTATUS low pulse width | 45 | | 230 *(3)* | | µs |

**Table 8–12. PS Configuration Timing Parameters For Cyclone IV Devices (Part 2 of 2)**

| Symbol | Parameter | Minimum | | Maximum | | Unit |
|---|---|---|---|---|---|---|
| | | Cyclone IV (1) | Cyclone IV E (2) | Cyclone IV (1) | Cyclone IV E (2) | |
| $t_{CF2ST1}$ | nCONFIG high to nSTATUS high | — | | 230 (4) | | µs |
| $t_{CF2CK}$ | nCONFIG high to first rising edge on DCLK | 230 (3) | | — | | µs |
| $t_{ST2CK}$ | nSTATUS high to first rising edge of DCLK | 2 | | — | | µs |
| $t_{DH}$ | Data hold time after rising edge on DCLK | 0 | | — | | ns |
| $t_{CD2UM}$ | CONF_DONE high to user mode (5) | 300 | | 650 | | 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 | $t_{CD2CU}$ + (3,192 × CLKUSR period) | | — | | — |
| $t_{DSU}$ | Data setup time before rising edge on DCLK | 5 | 8 | — | — | ns |
| $t_{CH}$ | DCLK high time | 3.2 | 6.4 | — | — | ns |
| $t_{CL}$ | DCLK low time | 3.2 | 6.4 | — | — | ns |
| $t_{CLK}$ | DCLK period | 7.5 | 15 | — | — | ns |
| $f_{MAX}$ | DCLK frequency (6) | — | — | 133 | 66 | MHz |

**Notes to Table 8–12:**

(1) Applicable for Cyclone IV GX and Cyclone IV E devices with 1.2-V core voltage.
(2) Applicable for Cyclone IV E devices with 1.0-V core voltage.
(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.
(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.
(5) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.
(6) Cyclone IV E devices with 1.0-V core voltage have slower $F_{MAX}$ when compared with Cyclone IV GX devices with 1.2-V core voltage.

## PS Configuration Using a Download Cable

In this section, the generic term "download cable" includes the Altera USB-Blaster USB port download cable, MasterBlaster™ serial and USB communications cable, ByteBlaster II parallel port download cable, the ByteBlasterMV™ parallel port download cable, and the EthernetBlaster communications cable.

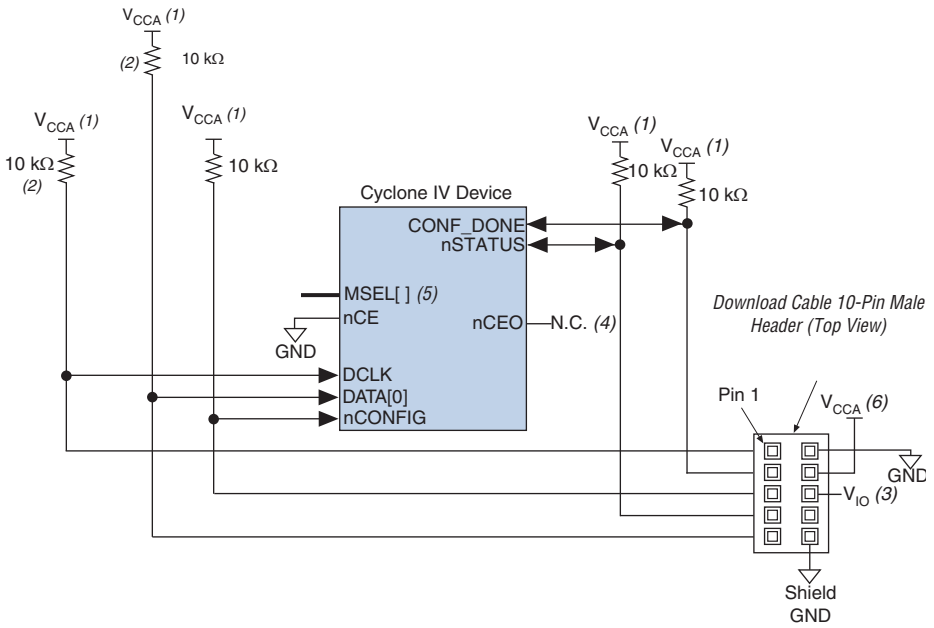
In the PS configuration with a download cable, an intelligent host (such as a PC) transfers data from a storage device to the Cyclone IV device through the download cable.

The programming hardware or download cable then places the configuration data one bit at a time on the DATA[0] pin of the device. 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 you use 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 if an error occurs. Additionally, the **Enable user-supplied start-up clock (CLKUSR)** option has no effect on device initialization, because this option is disabled in the **.sof** when programming the device with the Quartus II Programmer and download cable. Therefore, if you turn on the **CLKUSR** option, you do not have to provide a clock on CLKUSR when you configure the device with the Quartus II Programmer and a download cable.

Figure 8–17 shows PS configuration for Cyclone IV devices with a download cable.

**Figure 8–17. PS Configuration Using a Download Cable**




**Notes to Figure 8–17:**

(1) You must connect the pull-up resistor to the same supply voltage as the $V_{CCA}$ supply.

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

(3) Pin 6 of the header is a $V_{IO}$ reference voltage for the MasterBlaster output driver. $V_{IO}$ must match the $V_{CCA}$ of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. With the USB-Blaster, ByteBlaster II, ByteBlaster MV, and EthernetBlaster, this pin is a no connect.

(4) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

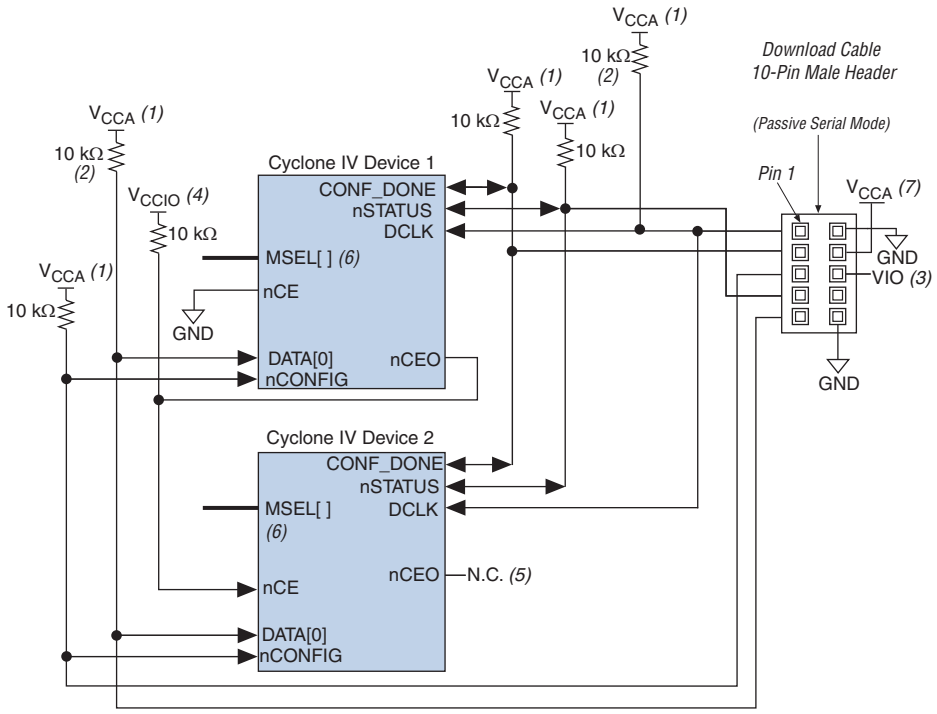
(5) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9 for PS configuration schemes. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(6) Power up the $V_{CC}$ of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from $V_{CCA}$. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a $V_{CC}$ power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

You can use a download cable to configure multiple Cyclone IV device configuration pins. nCONFIG, nSTATUS, DCLK, DATA[0], and CONF_DONE are connected to every device in the chain. All devices in the chain utilize and enter user mode at the same time because all CONF_DONE pins are tied together.

In addition, the entire chain halts configuration if any device detects an error because the nSTATUS pins are tied together. Figure 8–18 shows the PS configuration for multiple Cyclone IV devices using a MasterBlaster, USB-Blaster, ByteBlaster II, or ByteBlasterMV cable.

**Figure 8–18. Multi-Device PS Configuration Using a Download Cable**




**Notes to Figure 8–18:**

(1) You must connect the pull-up resistor to the same supply voltage as the $V_{CCA}$ supply.

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

(3) Pin 6 of the header is a $V_{IO}$ reference voltage for the MasterBlaster output driver. $V_{IO}$ must match the $V_{CCA}$ of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using USB-Blaster, ByteBlaster II, and EthernetBlaster cables, this pin is connected to nCE when it is used for AS programming. Otherwise, it is a no connect.

(4) Connect the pull-up resistor to the $V_{CCIO}$ supply voltage of the I/O bank in which the nCE pin resides.

(5) The nCEO pin of the last device in the chain is left unconnected or used as a user I/O pin.

(6) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for PS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(7) Power up the $V_{CC}$ of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5 V supply from $V_{CCA}$. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a $V_{CC}$ power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

## FPP Configuration

The FPP configuration in Cyclone IV devices is designed to meet the increasing demand for faster configuration time. Cyclone IV devices are designed with the capability of receiving byte-wide configuration data per clock cycle.

You can perform FPP configuration of Cyclone IV devices with an intelligent host, such as a MAX II device or microprocessor with flash memory. If your system already contains a CFI flash memory, you can use it for the Cyclone IV device configuration storage as well. The MAX II PFL feature in MAX II devices provides an efficient method to program CFI flash memory devices through the JTAG interface and the logic to control configuration from the flash memory device to the Cyclone IV device.

For more information about the PFL, refer to *AN 386: Using the Parallel Flash Loader with the Quartus II Software*.

FPP configuration is supported in EP4CGX30 (only for F484 package), EP4CGX50, EP4CGX75, EP4CGX110, EP4CGX150, and all Cyclone IV E devices.

The FPP configuration is not supported in E144 package of Cyclone IV E devices.

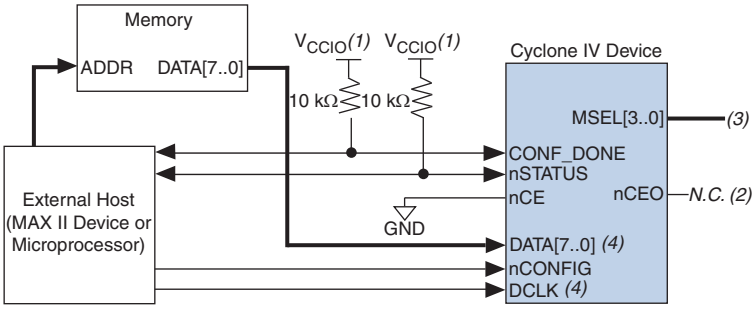
Cyclone IV devices do not support enhanced configuration devices for FPP configuration.

### FPP Configuration Using an External Host

FPP configuration using an external host provides a fast method to configure Cyclone IV devices. In the FPP configuration scheme, you can use an external host device to control the transfer of configuration data from a storage device, such as flash memory, to the target Cyclone IV device. You can store configuration data in an **.rbf**, **.hex**, or **.ttf** format. When using the external 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 external host device. Figure 8–19 shows the configuration interface connections between the Cyclone IV devices and an external device for single-device configuration.

**Figure 8–19. Single-Device FPP Configuration Using an External Host**




**Notes to Figure 8–19:**

(1) Connect the pull-up resistor to a supply that provides an acceptable input signal for the device. $V_{CC}$ must be high enough to meet the $V_{IH}$ specification of the I/O on the device and the external host.

(2) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–4 on page 8–8 and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[7..0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

After nSTATUS is released, the device is ready to receive configuration data and the configuration stage begins. When nSTATUS is pulled high, the external host device places the configuration data one byte at a time on the DATA[7..0] pins.

Cyclone 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. Data is continuously clocked into the target device until CONF_DONE goes high. The CONF_DONE pin goes high one byte early in FPP configuration mode. The last byte is required for serial configuration (AS and PS) modes.

Two DCLK falling edges are required after CONF_DONE goes high to begin initialization of the device.

Supplying a clock on CLKUSR does not affect the configuration process. After the CONF_DONE pin goes high, CLKUSR is enabled after the time specified as $t_{CD2CU}$. After this time period elapses, Cyclone IV devices require 3,192 clock cycles to initialize properly and enter user mode. For more information about the supported CLKUSR $f_{MAX}$ value for Cyclone IV devices, refer to Table 8–13 on page 8–44.

The INIT_DONE pin is released and pulled high when initialization is complete. The external host 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.

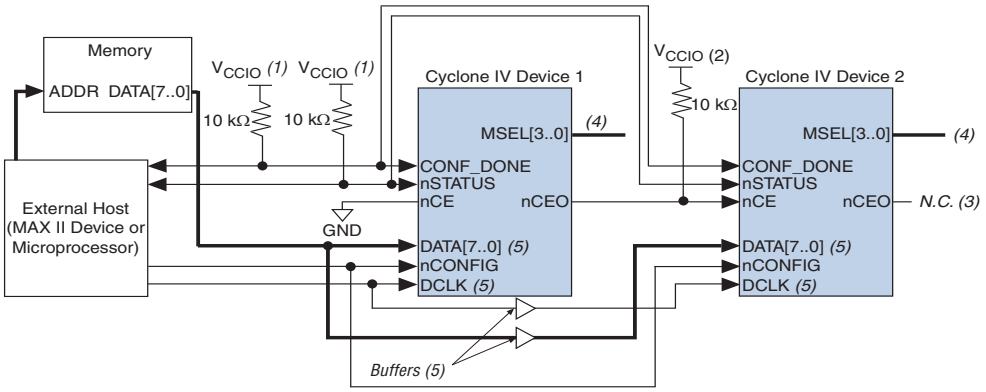
To ensure that DCLK and DATA[0] are not left floating at the end of the configuration, the MAX II device must drive them either high or low, whichever is convenient on your board. The DATA[0] pin is available as a user I/O pin after configuration. When you choose the FPP scheme in the Quartus II software, the DATA[0] pin is tri-stated by default in user mode and must be driven by the external host 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 DCLK speed must be below the specified system 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.

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

Figure 8–20 shows how to configure multiple devices with a MAX II device. This circuit is similar to the FPP configuration circuit for a single device, except the Cyclone IV devices are cascaded for multi-device configuration.

**Figure 8–20. Multi-Device FPP Configuration Using an External Host**




**Notes to Figure 8–20:**

(1) The pull-up resistor must be connected to a supply that provides an acceptable input signal for all devices in the chain. $V_{CC}$ must be high enough to meet the $V_{IH}$ specification of the I/O on the device and the external host.

(2) Connect the pull-up resistor to the $V_{CCIO}$ supply voltage of the I/O bank in which the nCE pin resides.

(3) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(4) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–4 on page 8–8 and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(5) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[7..0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

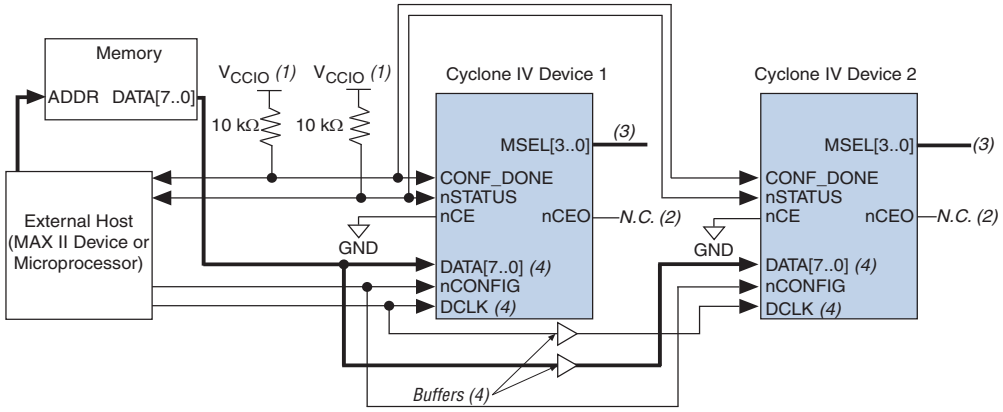
After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. The second device in the chain begins configuration in one clock cycle; therefore, the transfer of data 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. Configuration signals may require buffering to ensure signal integrity and prevent clock skew problems. Ensure that the DCLK and DATA lines are buffered. All devices initialize and enter user mode at the same time, because all device CONF_DONE pins are tied together.

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

Figure 8–21 shows multi-device FPP configuration when both Cyclone IV devices are receiving the same configuration data. 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. Devices must be of the same density and package. All devices start and complete configuration at the same time.

**Figure 8–21. Multi-Device FPP Configuration Using an External Host When Both Devices Receive the Same Data**




**Notes to Figure 8–21:**

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain. $V_{CC}$ must be high enough to meet the $V_{IH}$ specification of the I/O on the device and the external host.

(2) The nCEO pins of both devices are left unconnected or used as user I/O pins when configuring the same configuration data into multiple devices.

(3) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect the MSEL pins, refer to Table 8–4 on page 8–8 and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. DATA[7..0] and DCLK must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

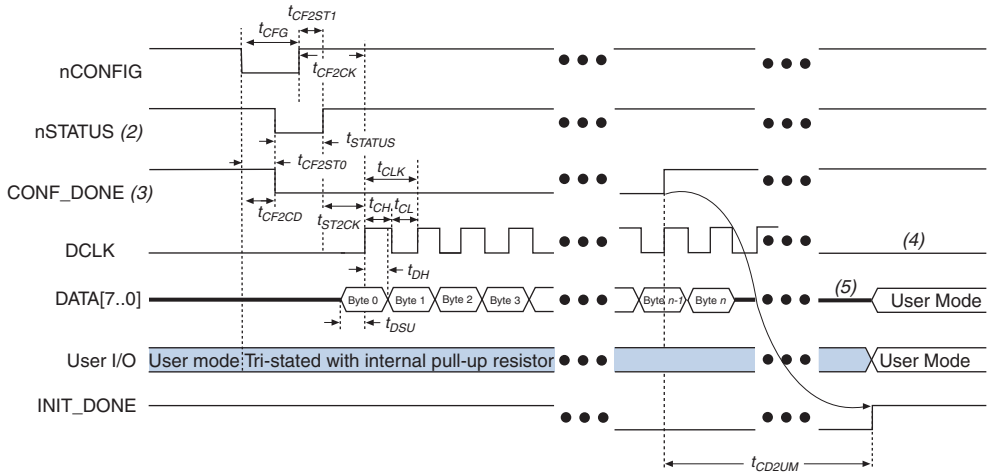
You can use a single configuration chain to configure Cyclone IV devices with other Altera devices that support FPP configuration. To ensure that all devices in the chain complete configuration at the same time or that an error flagged by one device starts reconfiguration in all devices, tie all the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

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

## FPP Configuration Timing

Figure 8–22 shows the timing waveform for the FPP configuration when using an external host.

**Figure 8–22. FPP Configuration Timing Waveform** *(1)*




**Notes to Figure 8–22:**

(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 Cyclone IV device holds nSTATUS low during POR delay.

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

(4) Do not leave DCLK floating after configuration. It must be driven high or low, whichever is more convenient.

(5) DATA[7..0] is available as a user I/O pin after configuration; the state of the pin depends on the dual-purpose pin settings.

Table 8–13 lists the FPP configuration timing parameters for Cyclone IV devices.

**Table 8–13. FPP Timing Parameters for Cyclone IV Devices (Part 1 of 2)**

| Symbol | Parameter | Minimum | | Maximum | | Unit |
|---|---|---|---|---|---|---|
| | | Cyclone IV *(1)* | Cyclone IV E *(2)* | Cyclone IV *(1)* | Cyclone IV E *(2)* | |
| $t_{CF2CD}$ | nCONFIG low to CONF_DONE low | — | | 500 | | ns |
| $t_{CF2ST0}$ | nCONFIG low to nSTATUS low | — | | 500 | | ns |
| $t_{CFG}$ | nCONFIG low pulse width | 500 | | — | | ns |
| $t_{STATUS}$ | nSTATUS low pulse width | 45 | | 230 *(3)* | | µs |
| $t_{CF2ST1}$ | nCONFIG high to nSTATUS high | — | | 230 *(4)* | | µs |
| $t_{CF2CK}$ | nCONFIG high to first rising edge on DCLK | 230 *(3)* | | — | | µs |

**Table 8–13. FPP Timing Parameters for Cyclone IV Devices (Part 2 of 2)**

| Symbol | Parameter | Minimum | | Maximum | | Unit |
|---|---|---|---|---|---|---|
| | | Cyclone IV (1) | Cyclone IV E (2) | Cyclone IV (1) | Cyclone IV E (2) | |
| $t_{ST2CK}$ | nSTATUS high to first rising edge of DCLK | 2 | | — | | µs |
| $t_{DH}$ | Data hold time after rising edge on DCLK | 0 | | — | | ns |
| $t_{CD2UM}$ | CONF_DONE high to user mode (5) | 300 | | 650 | | 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 | $t_{CD2CU}$ + (3,192 × CLKUSR period) | | — | | — |
| $t_{DSU}$ | Data setup time before rising edge on DCLK | 5 | 8 | — | — | ns |
| $t_{CH}$ | DCLK high time | 3.2 | 6.4 | — | — | ns |
| $t_{CL}$ | DCLK low time | 3.2 | 6.4 | — | — | ns |
| $t_{CLK}$ | DCLK period | 7.5 | 15 | — | — | ns |
| $f_{MAX}$ | DCLK frequency (6) | — | — | 133 | 66 | MHz |

**Notes to Table 8–13:**

(1) Applicable for Cyclone IV GX and Cyclone IV E with 1.2-V core voltage.
(2) Applicable for Cyclone IV E with 1.0-V core voltage.
(3) This value is applicable if you do not delay configuration by extending the nCONFIG or nSTATUS low pulse width.
(4) This value is applicable if you do not delay configuration by externally holding the nSTATUS low.
(5) The minimum and maximum numbers apply only if you choose the internal oscillator as the clock source for starting the device.
(6) Cyclone IV E devices with 1.0-V core voltage have slower $F_{MAX}$ when compared with Cyclone IV GX devices with 1.2-V core voltage.

## JTAG Configuration

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

For more information about the JTAG boundary-scan testing, refer to the *JTAG Boundary-Scan Testing for Cyclone IV Devices* chapter.

JTAG instructions have precedence over any other configuration modes. Therefore, JTAG configuration can take place without waiting for other configuration modes to complete. For example, if you attempt JTAG configuration in Cyclone IV devices during PS configuration, PS configuration terminates and JTAG configuration begins. If the MSEL pins are set to AS mode, the Cyclone IV device does not output a DCLK signal when JTAG configuration takes place.

The four required pins for a device operating in JTAG mode are TDI, TDO, TMS, and TCK. All the JTAG input pins are powered by the $V_{CCIO}$ pin and support the LVTTL I/O standard only. All user I/O pins are tri-stated during JTAG configuration. Table 8–14 explains the function of each JTAG pin.

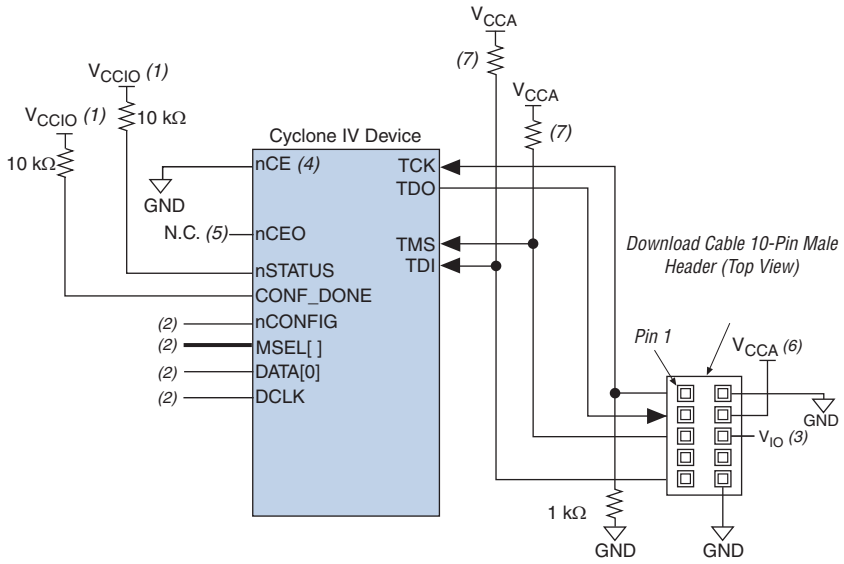
**Table 8–14. Dedicated JTAG Pins**

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

You can download data to the device through the USB-Blaster, MasterBlaster, ByteBlaster II, or ByteBlasterMV download cable, or the EthernetBlaster communications cable during JTAG configuration. Configuring devices with a cable is similar to programming devices in-system. Figure 8–23 and Figure 8–24 show the JTAG configuration of a single Cyclone IV device.

For device using $V_{CCIO}$ of 2.5, 3.0, and 3.3 V, refer to Figure 8–23. All I/O inputs must maintain a maximum AC voltage of 4.1 V because JTAG pins do not have the internal PCI clamping diodes to prevent voltage overshoot when using $V_{CCIO}$ of 2.5, 3.0, and 3.3 V. You must power up the $V_{CC}$ of the download cable with a 2.5-V supply from $V_{CCA}$. For device using $V_{CCIO}$ of 1.2, 1.5, and 1.8 V, refer to Figure 8–24. You can power up the $V_{CC}$ of the download cable with the supply from $V_{CCIO}$.

**Figure 8–23. JTAG Configuration of a Single Device Using a Download Cable (2.5, 3.0, and 3.3-V $V_{CCIO}$ Powering the JTAG Pins)**




**Notes to Figure 8–23:**

(1) Connect these pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) Connect the `nCONFIG` and `MSEL` pins to support a non-JTAG configuration scheme. If you only use JTAG configuration, connect the `nCONFIG` pin to logic-high and the `MSEL` pins to GND. In addition, pull `DCLK` and `DATA[0]` to 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 $V_{CCA}$. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the USB-Blaster, ByteBlaster II, ByteBlasterMV, and EthernetBlaster cables, this pin is a no connect.

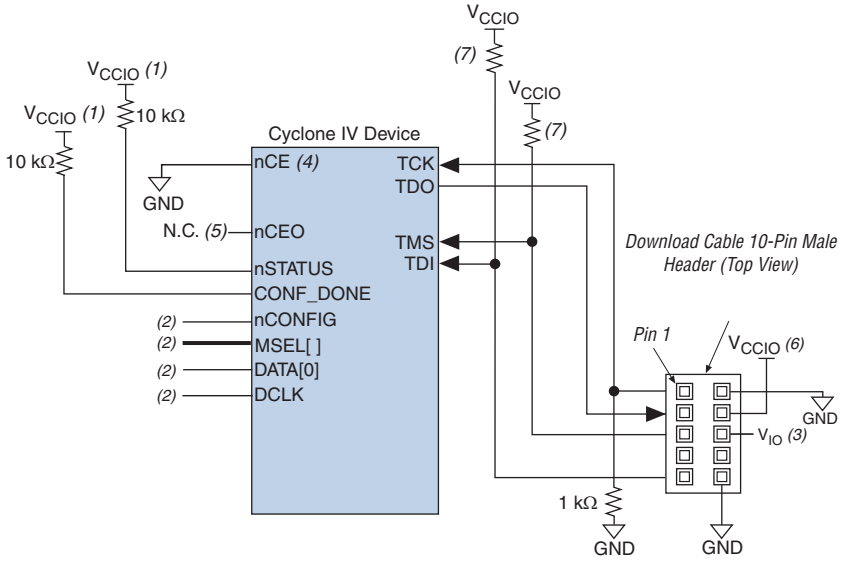
(4) The `nCE` pin must be connected to GND or driven low for successful JTAG configuration.

(5) The `nCEO` pin is left unconnected or used as a user I/O pin when it does not feed the `nCE` pin of another device.

(6) Power up the $V_{CC}$ of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from $V_{CCA}$. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a $V_{CC}$ power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(7) Resistor value can vary from 1 kΩ to 10 kΩ.

**Figure 8–24. JTAG Configuration of a Single Device Using a Download Cable (1.5-V or 1.8-V $V_{CCIO}$ Powering the JTAG Pins)**




**Notes to Figure 8–24:**

(1) Connect these pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

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

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

(4) The nCE must be connected to GND or driven low for successful JTAG configuration.

(5) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(6) Power up the $V_{CC}$ of the EthernetBlaster, ByteBlaster II or USB-Blaster cable with supply from $V_{CCIO}$. The Ethernet-Blaster, ByteBlaster II, and USB-Blaster cables do not support a target supply voltage of 1.2 V. For the target supply voltage value, refer to the *ByteBlaster II Download Cable User Guide*, the *USB-Blaster Download Cable User Guide*, and the *EthernetBlaster Communications Cable User Guide*.

(7) Resistor value can vary from 1 kΩ 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 after completion. At the end of configuration, the software checks the state of CONF_DONE through the JTAG port. When Quartus II generates a **.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 serially sent using the JTAG TDI port, the TCK port clocks an additional clock cycles to perform device initialization.

You can perform JTAG testing on Cyclone IV devices before, during, and after configuration. Cyclone IV devices support the BYPASS, IDCODE, and SAMPLE instructions during configuration without interrupting configuration. All other JTAG instructions can only be issued by first interrupting configuration and reprogramming I/O pins with the ACTIVE_DISENGAGE and CONFIG_IO instructions.

The CONFIG_IO instruction allows you to configure the I/O buffers through the JTAG port and interrupts configuration when issued after the ACTIVE_DISENGAGE instruction. This instruction allows you to perform board-level testing prior to configuring the Cyclone IV device or waiting for a configuration device to complete configuration. Prior to issuing the CONFIG_IO instruction, you must issue the ACTIVE_DISENGAGE instruction. This is because in Cyclone IV devices, the CONFIG_IO instruction does not hold nSTATUS low until reconfiguration, so you must disengage the active configuration mode controller when active configuration is interrupted. The ACTIVE_DISENGAGE instruction places the active configuration mode controllers in an idle state prior to JTAG programming. Additionally, the ACTIVE_ENGAGE instruction allows you to re-engage a disengaged active configuration mode controller.

You must follow a specific flow when executing the ACTIVE_DISENGAGE, CONFIG_IO, and ACTIVE_ENGAGE JTAG instructions in Cyclone IV devices.

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

When designing a board for JTAG configuration of Cyclone IV devices, consider the dedicated configuration pins. Table 8–15 describes how you must connect these pins during JTAG configuration.

**Table 8–15. Dedicated Configuration Pin Connections During JTAG Configuration**

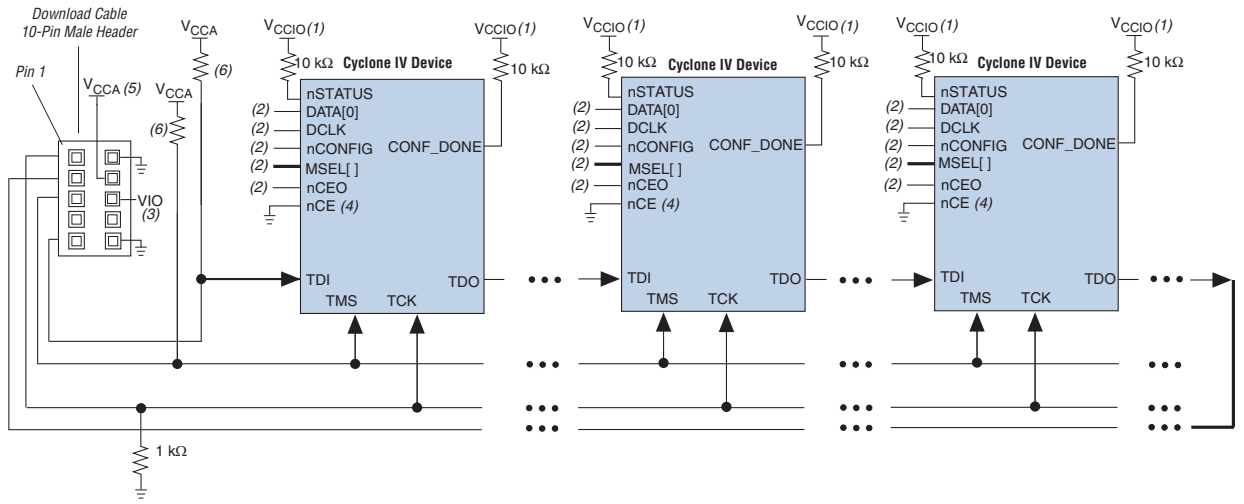
| Signal | Description |
|---|---|
| nCE | On all Cyclone IV devices in the chain, nCE must be driven low by connecting it to GND, pulling it low through a resistor, or driving it by some control circuitry. For devices that are also in multi-device AS, AP, PS, or FPP configuration chains, you must connect the nCE pins to GND during JTAG configuration or JTAG configured in the same order as the configuration chain. |
| nCEO | On all Cyclone IV devices in the chain, nCEO is left floating or connected to the nCE of the next device. |
| MSEL | These pins must not be left floating. These pins support whichever non-JTAG configuration that you used in production. If you only use JTAG configuration, tie these pins to GND. |
| nCONFIG | Driven high by connecting to the $V_{CCIO}$ supply of the bank in which the pin resides and pulling up through a resistor or driven high by some control circuitry. |
| nSTATUS | Pull to the $V_{CCIO}$ supply of the bank in which the pin resides through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin must be pulled up to the $V_{CCIO}$ individually. |
| CONF_DONE | Pull to the $V_{CCIO}$ supply of the bank in which the pin resides through a 10-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin must be pulled up to $V_{CCIO}$ supply of the bank in which the pin resides individually. CONF_DONE going high at the end of JTAG configuration indicates successful configuration. |
| DCLK | Must not be left 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 with JTAG BST circuitry. Figure 8–25 and Figure 8–26 show multi-device JTAG configuration.

For devices using 2.5-, 3.0-, and 3.3-V $V_{CCIO}$ supply, you must refer to Figure 8–25. All I/O inputs must maintain a maximum AC voltage of 4.1 V because JTAG pins do not have the internal PCI clamping diodes to prevent voltage overshoot when using 2.5-, 3.0-, and 3.3- V $V_{CCIO}$ supply. You must power up the $V_{CC}$ of the download cable with a 2.5-V $V_{CCA}$ supply. For device using $V_{CCIO}$ of 1.2, 1.5 V, and 1.8 V, refer to Figure 8–26. You can power up the $V_{CC}$ of the download cable with the supply from $V_{CCIO}$.

**Figure 8–25. JTAG Configuration of Multiple Devices Using a Download Cable (2.5, 3.0, and 3.3-V $V_{CCIO}$ Powering the JTAG Pins)**




**Notes to Figure 8–25:**

(1) Connect these pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

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

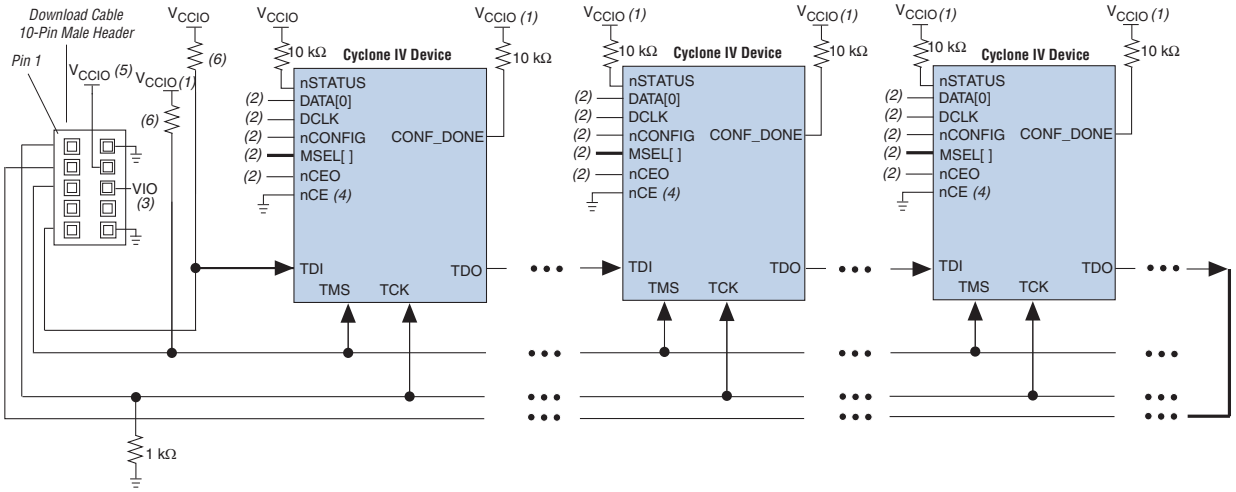
(3) Pin 6 of the header is a $V_{IO}$ reference voltage for the MasterBlaster output driver. $V_{IO}$ must match the $V_{CCA}$ of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. In the ByteBlasterMV cable, this pin is a no connect. In the USB-Blaster and ByteBlaster II cables, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

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

(5) Power up the $V_{CC}$ of the ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V supply from $V_{CCA}$. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a $V_{CC}$ power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(6) Resistor value can vary from 1 kΩ to 10 kΩ.

**Figure 8–26. JTAG Configuration of Multiple Devices Using a Download Cable (1.2, 1.5, and 1.8-V $V_{CCIO}$ Powering the JTAG Pins)**




**Notes to Figure 8–26:**

(1) Connect these pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

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

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

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

(5) Power up the $V_{CC}$ of the ByteBlaster II or USB-Blaster cable with supply from $V_{CCIO}$. The ByteBlaster II and USB-Blaster cables do not support a target supply voltage of 1.2 V. For the target supply voltage value, refer to the *ByteBlaster II Download Cable User Guide* and the *USB-Blaster Download Cable User Guide*.

(6) Resistor value can vary from 1 kΩ to 10 kΩ.

If a non-Cyclone IV device is cascaded in the JTAG-chain, TDO of the non-Cyclone IV device driving into TDI of the Cyclone IV device must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

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

If you only use JTAG configuration, Altera recommends that you connect the circuitry as shown in Figure 8–25 or Figure 8–26, in which each of the CONF_DONE and nSTATUS signals are isolated so that each device can enter user mode individually.

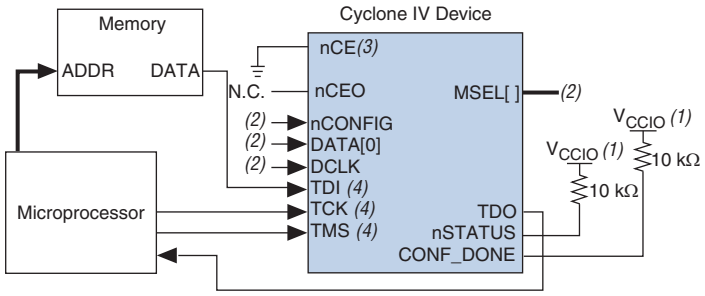
After the first device completes configuration in a multi-device configuration chain, its nCEO pin drives low to activate the nCE pin of the second device, which prompts the second device to begin configuration. Therefore, if these devices are also in a JTAG chain, ensure that the nCE pins are connected to GND during JTAG configuration or that the devices are JTAG configured in the same order as the configuration chain. 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 pin 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 allows an unlimited number of Cyclone 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*.

Figure 8–27 shows JTAG configuration with a Cyclone IV device and a microprocessor.

**Figure 8–27. JTAG Configuration of a Single Device Using a Microprocessor**




**Notes to Figure 8–27:**

(1) You must connect the pull-up resistor to a supply that provides an acceptable input signal for all devices in the chain.

(2) Connect the `nCONFIG` and `MSEL` pins to support a non-JTAG configuration scheme. If you only use a JTAG configuration, connect the `nCONFIG` pin to logic-high and the `MSEL` pins to GND. In addition, pull `DCLK` and `DATA[0]` to either high or low, whichever is convenient on your board.

(3) You must connect the `nCE` pin to GND or driven low for successful JTAG configuration.

(4) All I/O inputs must maintain a maximum AC voltage of 4.1 V. Signals driving into `TDI`, `TMS`, and `TCK` must fit the maximum overshoot outlined in Equation 8–1 on page 8–5.

## Configuring Cyclone IV Devices with 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 *AN 425: Using Command-Line Jam STAPL Solution for Device Programming*. To download the Jam Player, visit the Altera website (www.altera.com).

## Configuring Cyclone IV Devices with the JRunner Software Driver

The JRunner software driver allows you to configure Cyclone IV devices through the ByteBlaster II or ByteBlasterMV cables in JTAG mode. The supported programming input file is in **.rbf** format. The JRunner software driver also requires a Chain Description File (**.cdf**) generated by the Quartus II software. The JRunner software driver is targeted for embedded JTAG configuration. The source code is developed for the Windows NT operating system (OS). You can customize the code to make it run on your embedded platform.

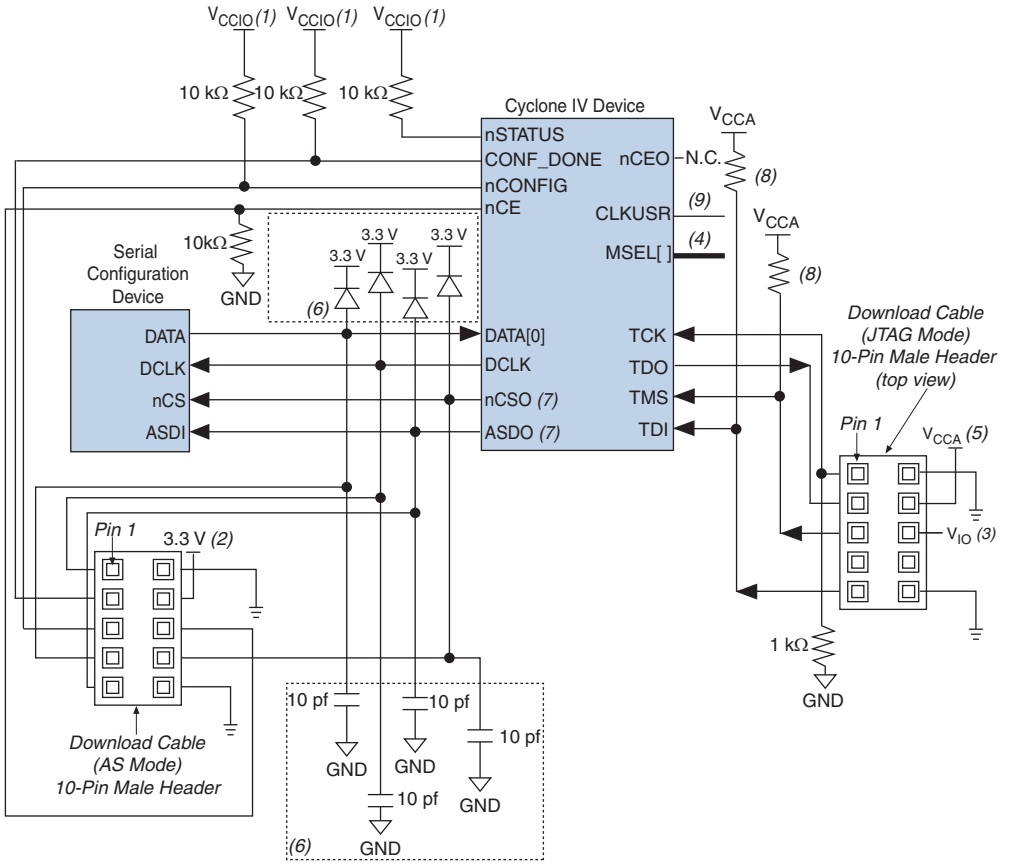
The **.rbf** used by the JRunner software driver cannot be a compressed **.rbf** because the JRunner software driver uses JTAG-based configuration. During JTAG-based configuration, the real-time decompression feature is not available.

For more information about the JRunner software driver, refer to *AN 414: JRunner Software Driver: An Embedded Solution for PLD JTAG Configuration* and the source files on the Altera website at (www.altera.com).

## Combining JTAG and AS Configuration Schemes

You can combine the AS configuration scheme with the JTAG-based configuration (Figure 8–28). This setup uses two 10-pin download cable headers on the board. One download cable is used in JTAG mode to configure the Cyclone IV device directly through the JTAG interface. The other download cable is used in AS mode to program the serial configuration device in-system through the AS programming interface. If you try configuring the device using both schemes simultaneously, JTAG configuration takes precedence and AS configuration terminates.

**Figure 8–28. Combining JTAG and AS Configuration Schemes**




**Notes to Figure 8–28:**

(1) Connect these pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) Power up the $V_{CC}$ of the EthernetBlaster, ByteBlaster II, or USB-Blaster cable with the 3.3-V supply.

(3) Pin 6 of the header is a $V_{IO}$ reference voltage for the MasterBlaster output driver.The $V_{IO}$ must match the $V_{CCA}$ of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using the USB-Blaster and ByteBlaster II cables, this pin is connected to `nCE` when it is used for AS programming, otherwise it is a no connect.

(4) The `MSEL` pin settings vary for different configuration voltage standards and POR time. To connect `MSEL` for AS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the `MSEL` pins directly to $V_{CCA}$ or GND.

(5) Power up the $V_{CC}$ of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5-V $V_{CCA}$ supply. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a $V_{CC}$ power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(6) You must place the diodes and capacitors as close as possible to the Cyclone IV device. Altera recommends using the Schottky diode, which has a relatively lower forward diode voltage (VF) than the switching and Zener diodes, for effective voltage clamping.

(7) These pins are dual-purpose I/O pins. The `nCSO` pin functions as `FLASH_nCE` pin in AP mode. The `ASDO` pin functions as `DATA[1]` pin in AP and FPP modes.

(8) Resistor value can vary from 1 kΩ to 10 kΩ.

(9) Only Cyclone IV GX devices have an option to select `CLKUSR` (40 MHz maximum) as the external clock source for `DCLK`.

## Programming Serial Configuration Devices In-System with the JTAG Interface

Cyclone IV devices in a single- or multiple-device chain support in-system programming of a serial configuration device with the JTAG interface through the SFL design. The intelligent host or download cable of the board can use the four JTAG pins on the Cyclone IV device to program the serial configuration device in system, even if the host or download cable cannot access the configuration pins (DCLK, DATA, ASDI, and nCS pins).

The SFL design is a JTAG-based in-system programming solution for Altera serial configuration devices. The SFL is a bridge design for the Cyclone IV device that uses their JTAG interface to access the EPCS JTAG Indirect Configuration Device Programming (**.jic**) file and then uses the AS interface to program the EPCS device. Both the JTAG interface and AS interface are bridged together inside the SFL design.

In a multiple device chain, you must only configure the master device that controls the serial configuration device. Slave devices in the multiple device chain that are configured by the serial configuration device do not have to be configured when using this feature. To successfully use this feature, set the MSEL pins of the master device to select the AS configuration scheme (Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9). The serial configuration device in-system programming through the Cyclone IV device JTAG interface has three stages, which are described in the following sections:

- "Loading the SFL Design"
- "ISP of the Configuration Device" on page 8–56
- "Reconfiguration" on page 8–57

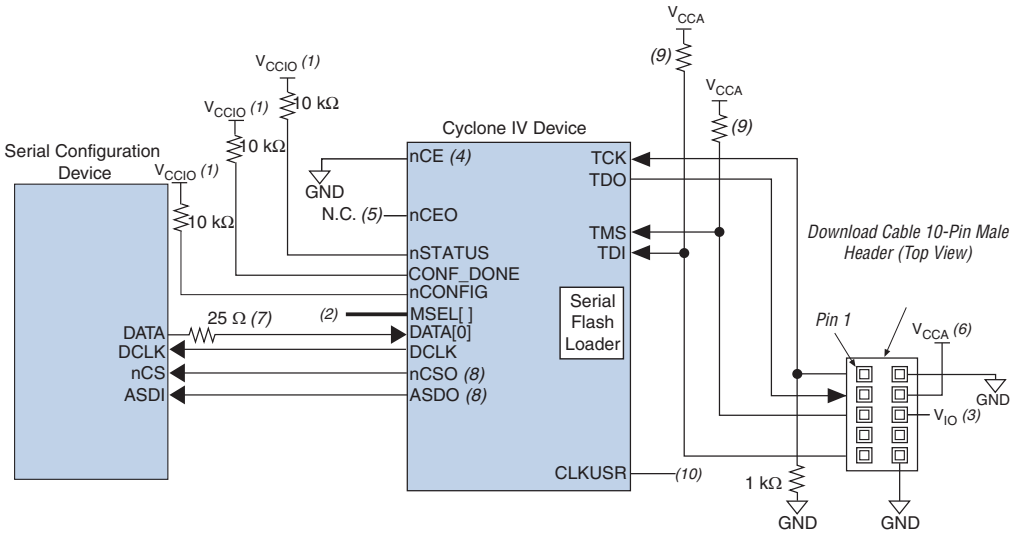
### Loading the SFL Design

The SFL design is a design inside the Cyclone IV device that bridges the JTAG interface and AS interface with glue logic.

The intelligent host uses the JTAG interface to configure the master device with a SFL design. The SFL design allows the master device to control the access of four serial configuration device pins, also known as the Active Serial Memory Interface (ASMI) pins, through the JTAG interface. The ASMI pins are serial clock input (DCLK), serial data output (DATA), AS data input (ASDI), and active-low chip select (nCS) pins.

If you configure a master device with an SFL design, the master device enters user mode even though the slave devices in the multiple device chain are not being configured. The master device enters user mode with a SFL design even though the CONF_DONE signal is externally held low by the other slave devices in chain. Figure 8–29 shows the JTAG configuration of a single Cyclone IV device with a SFL design.

**Figure 8–29. Programming Serial Configuration Devices In-System Using the JTAG Interface**




**Notes to Figure 8–29:**

(1) Connect the pull-up resistors to the $V_{CCIO}$ supply of the bank in which the pin resides.

(2) The MSEL pin settings vary for different configuration voltage standards and POR time. To connect MSEL for AS configuration schemes, refer to Table 8–3 on page 8–8, Table 8–4 on page 8–8, and Table 8–5 on page 8–9. Connect the MSEL pins directly to $V_{CCA}$ or GND.

(3) Pin 6 of the header is a $V_{IO}$ reference voltage for the MasterBlaster output driver. The $V_{IO}$ must match the $V_{CCA}$ of the device. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*. When using the ByteBlasterMV download cable, this pin is a no connect. When using USB-Blaster, ByteBlaster II, and EthernetBlaster cables, this pin is connected to nCE when it is used for AS programming, otherwise it is a no connect.

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

(5) The nCEO pin is left unconnected or used as a user I/O pin when it does not feed the nCE pin of another device.

(6) Power up the $V_{CC}$ of the EthernetBlaster, ByteBlaster II, USB-Blaster, or ByteBlasterMV cable with a 2.5- V $V_{CCA}$ supply. Third-party programmers must switch to 2.5 V. Pin 4 of the header is a $V_{CC}$ power supply for the MasterBlaster cable. The MasterBlaster cable can receive power from either 5.0- or 3.3-V circuit boards, DC power supply, or 5.0 V from the USB cable. For this value, refer to the *MasterBlaster Serial/USB Communications Cable User Guide*.

(7) Connect the series resistor at the near end of the serial configuration device.

(8) These pins are dual-purpose I/O pins. The nCSO pin functions as FLASH_nCE pin in AP mode. The ASDO pin functions as DATA[1] pin in AP and FPP modes.

(9) Resistor value can vary from 1 kΩ to 10 kΩ.

(10) Only Cyclone IV GX devices have an option to select CLKUSR (40 MHz maximum) as the external clock source for DCLK.

## ISP of the Configuration Device

In the second stage, the SFL design in the master device allows you to write the configuration data for the device chain into the serial configuration device with the Cyclone IV device JTAG interface. The JTAG interface sends the programming data for the serial configuration device to the Cyclone IV device first. The Cyclone IV device then uses the ASMI pins to send the data to the serial configuration device.

### Reconfiguration

After the configuration data is successfully written into the serial configuration device, the Cyclone IV device does not automatically start reconfiguration. The intelligent host issues the PULSE_NCONFIG JTAG instruction to initialize the reconfiguration process. During reconfiguration, the master device is reset and the SFL design no longer exists in the Cyclone IV device and the serial configuration device configures all the devices in the chain with the user design.

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

## JTAG Instructions

For more information about the JTAG binary instruction code, refer to the *JTAG Boundary-Scan Testing for Cyclone IV Devices* chapter.

### I/O Reconfiguration

Use the CONFIG_IO instruction to reconfigure the I/O configuration shift register (IOCSR) chain. This instruction allows you to perform board-level testing prior to configuring the Cyclone IV device or waiting for a configuration device to complete configuration. After the configuration is interrupted and JTAG testing is complete, you must reconfigure the part through the PULSE_NCONFIG JTAG instruction or by pulsing the nCONFIG pin low.

You can issue the CONFIG_IO instruction any time during user mode.

You must meet the following timing restrictions when using the CONFIG_IO instruction:

- The CONFIG_IO instruction cannot be issued when the nCONFIG pin is low
- You must observe a 230 µs minimum wait time after any of the following conditions:
  - nCONFIG pin goes high
  - Issuing the PULSE_NCONFIG instruction
  - Issuing the ACTIVE_ENGAGE instruction, before issuing the CONFIG_IO instruction
- You must wait 230 µs after power up, with the nCONFIG pin high before issuing the CONFIG_IO instruction (or wait for the nSTATUS pin to go high)

Use the ACTIVE_DISENGAGE instruction with the CONFIG_IO instruction to interrupt configuration. Table 8–16 lists the sequence of instructions to use for various CONFIG_IO usage scenarios.

**Table 8–16. JTAG CONFIG_IO (without JTAG_PROGRAM) Instruction Flows** (1)

| JTAG Instruction | Configuration Scheme and Current State of the Cyclone IV Device | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Prior to User Mode (Interrupting Configuration) | | | | User Mode | | | | Power Up | | | |
| | PS | FPP | AS | AP | PS | FPP | AS | AP | PS | FPP | AS | AP |
| ACTIVE_DISENGAGE | O | O | O | O | O | O | O | O | — | — | — | — |
| CONFIG_IO | R | R | R | R | R | R | R | R | NA | NA | NA | NA |
| JTAG Boundary Scan Instructions (no JTAG_PROGRAM) | O | O | O | O | O | O | O | O | — | — | — | — |
| ACTIVE_ENGAGE | A | A | R (2) | R (2) | A | A | R (2) | R (2) | — | — | — | — |
| PULSE_NCONFIG | | | A (3) | A (3) | | | O | O | — | — | — | — |
| Pulse nCONFIG pin | | | A (3) | A (3) | | | O | O | — | — | — | — |
| JTAG TAP Reset | R | R | R | R | R | R | R | R | — | — | — | — |

**Notes to Table 8–16:**

(1) You must execute "R" indicates that the instruction before the next instruction, "O" indicates the optional instruction, "A" indicates that the instruction must be executed, and "NA" indicates that the instruction is not allowed in this mode.

(2) Required if you use ACTIVE_DISENGAGE.

(3) Neither of the instruction is required if you use ACTIVE_ENGAGE.

The CONFIG_IO instruction does not hold nSTATUS low until reconfiguration. You must disengage the AS or AP configuration controller by issuing the ACTIVE_DISENGAGE and ACTIVE_ENGAGE instructions when active configuration is interrupted. You must issue the ACTIVE_DISENGAGE instruction alone or prior to the CONFIG_IO instruction if the JTAG_PROGRAM instruction is to be issued later (Table 8–17). This puts the active configuration controllers into the idle state. The active configuration controller is re-engaged after user mode is reached through JTAG programming (Table 8–17).

While executing the CONFIG_IO instruction, all user I/Os are tri-stated.

If reconfiguration after interruption is performed using configuration modes (rather than using JTAG_PROGRAM), it is not necessary to issue the ACTIVE_DISENGAGE instruction prior to CONFIG_IO. You can start reconfiguration by either pulling nCONFIG low for at least 500 ns or issuing the PULSE_NCONFIG instruction. If the ACTIVE_DISENGAGE instruction was issued and the JTAG_PROGRAM instruction fails to enter user mode, you must issue the ACTIVE_ENGAGE instruction to reactivate the active configuration controller. Issuing the ACTIVE_ENGAGE instruction also triggers reconfiguration in configuration modes; therefore, it is not necessary to pull nCONFIG low or issue the PULSE_NCONFIG instruction.

### ACTIVE_DISENGAGE

The ACTIVE_DISENGAGE instruction places the active configuration controller (AS and AP) into an idle state prior to JTAG programming. The two purposes of placing the active controller in an idle state are:

- To ensure that it is not trying to configure the device during JTAG programming
- To allow the controllers to properly recognize a successful JTAG programming that results in the device reaching user mode

The ACTIVE_DISENGAGE instruction is required before JTAG programming regardless of the current state of the Cyclone IV device if the MSEL pins are set to an AS or AP configuration scheme. If the ACTIVE_DISENGAGE instruction is issued during a passive configuration scheme (PS or FPP), it has no effect on the Cyclone IV device. Similarly, the CONFIG_IO instruction is issued after an ACTIVE_DISENGAGE instruction, but is no longer required to properly halt configuration. Table 8–17 lists the required, recommended, and optional instructions for each configuration mode. The ordering of the required instructions is a hard requirement and must be met to ensure functionality.

**Table 8–17. JTAG Programming Instruction Flows** *(1)*

| JTAG Instruction | Configuration Scheme and Current State of the Cyclone IV Device | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Prior to User Mode (Interrupting Configuration) | | | | User Mode | | | | Power Up | | | |
| | PS | FPP | AS | AP | PS | FPP | AS | AP | PS | FPP | AS | AP |
| ACTIVE_DISENGAGE | O | O | R | R | O | O | O | R | O | O | R | R |
| CONFIG_IO | Rc | Rc | O | O | O | O | O | O | NA | NA | NA | NA |
| Other JTAG instructions | O | O | O | O | O | O | O | O | O | O | O | O |
| JTAG_PROGRAM | R | R | R | R | R | R | R | R | R | R | R | R |
| CHECK_STATUS | Rc | Rc | Rc | Rc | Rc | Rc | Rc | Rc | Rc | Rc | Rc | Rc |
| JTAG_STARTUP | R | R | R | R | R | R | R | R | R | R | R | R |
| JTAG TAP Reset/other instruction | R | R | R | R | R | R | R | R | R | R | R | R |

**Note to Table 8–17:**

(1) "R" indicates that the instruction must be executed before the next instruction, "O" indicates the optional instruction, "Rc" indicates the recommended instruction, and "NA" indicates that the instruction is not allowed in this mode.

In the AS or AP configuration scheme, the ACTIVE_DISENGAGE instruction puts the active configuration controller into idle state. If a successful JTAG programming is executed, the active controller is automatically re-engaged after user mode is reached through JTAG programming. This causes the active controller to transition to their respective user mode states.

If JTAG programming fails to get the Cyclone IV device to enter user mode and re-engage active programming, there are available methods to achieve this:

- In AS configuration scheme, you can re-engage the AS controller by moving the JTAG TAP controller to the reset state or by issuing the ACTIVE_ENGAGE instruction.

- In AP configuration scheme, the only way to re-engage the AP controller is to issue the ACTIVE_ENGAGE instruction. In this case, asserting the nCONFIG pin does not re-engage either active controller.

#### ACTIVE_ENGAGE

The ACTIVE_ENGAGE instruction allows you to re-engage a disengaged active controller. You can issue this instruction any time during configuration or user mode to re-engage an already disengaged active controller, as well as trigger reconfiguration of the Cyclone IV device in the active configuration scheme.

The ACTIVE_ENGAGE instruction functions as the PULSE_NCONFIG instruction when the device is in the PS or FPP configuration schemes. The nCONFIG pin is disabled when the ACTIVE_ENGAGE instruction is issued.

Altera does not recommend using the ACTIVE_ENGAGE instruction, but it is provided as a fail-safe instruction for re-engaging the active configuration controller (AS and AP).

#### Overriding the Internal Oscillator

This feature allows you to override the internal oscillator during the active configuration scheme. The AS and AP configuration controllers use the internal oscillator as the clock source. You can change the clock source to CLKUSR through the JTAG instruction.

The EN_ACTIVE_CLK and DIS_ACTIVE_CLK JTAG instructions toggle on or off whether or not the active clock is sourced from the CLKUSR pin or the internal configuration oscillator. To source the active clock from the CLKUSR pin, issue the EN_ACTIVE_CLK instruction. This causes the CLKUSR pin to become the active clock source. When using the EN_ACTIVE_CLK instruction, you must enable the internal oscillator for the clock change to occur. By default, the configuration oscillator is disabled after configuration and initialization is complete as well as the device has entered user mode.

However, the internal oscillator is enabled in user mode by any of the following conditions:

- A reconfiguration event (for example, driving the nCONFIG pin to go low)
- Remote update is enabled
- Error detection is enabled

When using the EN_ACTIVE_CLK and DIS_ACTIVE_CLK JTAG instructions to override the internal oscillator, you must clock the CLKUSR pin at two times the expected DCLK frequency. The CLKUSR pin allows a maximum frequency of 40 MHz (40 MHz DCLK).

Normally, a test instrument uses the CLKUSR pin when it wants to drive its own clock to control the AS state machine.

To revert the clock source back to the configuration oscillator, issue the DIS_ACTIVE_CLK instruction. After you issue the DIS_ACTIVE_CLK instruction, you must continue to clock the CLKUSR pin for 10 clock cycles. Otherwise, even toggling the nCONFIG pin does not revert the clock source and reconfiguration does not occur. A POR reverts the clock source back to the configuration oscillator. Toggling the nCONFIG pin or driving the JTAG state machine to reset state does not revert the clock source.

### EN_ACTIVE_CLK

The EN_ACTIVE_CLK instruction causes the CLKUSR pin signal to replace the internal oscillator as the clock source. When using the EN_ACTIVE_CLK instruction, you must enable the internal oscillator for the clock change to occur. After this instruction is issued, other JTAG instructions can be issued while the CLKUSR pin signal remains as the clock source. The clock source is only reverted back to the internal oscillator by issuing the DIS_ACTIVE_CLK instruction or a POR.

### DIS_ACTIVE_CLK

The DIS_ACTIVE_CLK instruction breaks the CLKUSR enable latch set by the EN_ACTIVE_CLK instruction and causes the clock source to revert back to the internal oscillator. After the DIS_ACTIVE_CLK instruction is issued, you must continue to clock the CLKUSR pin for 10 clock cycles.

## Changing the Start Boot Address of the AP Flash

In the AP configuration scheme (for Cyclone IV E devices only), you can change the default configuration boot address of the parallel flash memory to any desired address using the APFC_BOOT_ADDR JTAG instruction.

### APFC_BOOT_ADDR

The APFC_BOOT_ADDR instruction is for Cyclone IV E devices only and allows you to define a start boot address for the parallel flash memory in the AP configuration scheme.

This instruction shifts in a start boot address for the AP flash. When this instruction becomes the active instruction, the TDI and TDO pins are connected through a 22-bit active boot address shift register. The shifted-in boot address bits get loaded into the 22-bit AP boot address update register, which feeds into the AP controller. The content of the AP boot address update register can be captured and shifted-out of the active boot address shift register from TDO.

The boot address in the boot address shift register and update register are shifted to the right (in the LSB direction) by two bits versus the intended boot address. The reason for this is that the two LSB of the address are not accessible. When this boot address is fed into the AP controller, two 0s are attached in the end as LSB, thereby pushing the shifted-in boot address to the left by two bits, which become the actual AP boot address the AP controller gets.

If you have enabled the remote update feature, the APFC_BOOT_ADDR instruction sets the boot address for the factory configuration only.

The APFC_BOOT_ADDR instruction is retained after reconfiguration while the system board is still powered on. However, you must reprogram the instruction whenever you restart the system board.

## Device Configuration Pins

Table 8–18 through Table 8–21 describe the connections and functionality of all the configuration related pins on Cyclone IV devices. Table 8–18 and Table 8–19 list the device pin configuration for the Cyclone IV GX and Cyclone IV E, respectively.

**Table 8–18. Configuration Pin Summary for Cyclone IV GX Devices**

| Bank | Description | Input/Output | Dedicated | Powered By | Configuration Mode |
|---|---|---|---|---|---|
| 8 | Data[4:2] | Input | — | $V_{CCIO}$ | FPP |
| 3 | Data[7:5] | Input | — | $V_{CCIO}$ | FPP |
| 9 | nCSO (2) | Output | — | $V_{CCIO}$ | AS |
| 3 | CRC_ERROR | Output | — | $V_{CCIO}$/Pull-up (1) | Optional, all modes |
| 9 | DATA[0] (2) | Input | Yes | $V_{CCIO}$ | PS, FPP, AS |
| 9 | DATA[1]/ASDO (2) | Input | — | $V_{CCIO}$ | FPP |
| | | Output | | $V_{CCIO}$ | AS |
| 3 | INIT_DONE | Output | — | Pull-up | Optional, all modes |
| 3 | nSTATUS | Bidirectional | Yes | Pull-up | All modes |
| 9 | nCE | Input | Yes | $V_{CCIO}$ | All modes |
| 9 | DCLK (2) | Input | Yes | $V_{CCIO}$ | PS, FPP |
| | | Output | | $V_{CCIO}$ | AS |
| 3 | CONF_DONE | Bidirectional | Yes | Pull-up | All modes |
| 9 | TDI | Input | Yes | $V_{CCIO}$ | JTAG |
| 9 | TMS | Input | Yes | $V_{CCIO}$ | JTAG |
| 9 | TCK | Input | Yes | $V_{CCIO}$ | JTAG |
| 9 | nCONFIG | Input | Yes | $V_{CCIO}$ | All modes |
| 8 | CLKUSR | Input | — | $V_{CCIO}$ | Optional |
| 3 | nCEO | Output | — | $V_{CCIO}$ | Optional, all modes |
| 3 | MSEL | Input | Yes | $V_{CCINT}$ | All modes |
| 9 | TDO | Output | Yes | $V_{CCIO}$ | JTAG |
| 6 | DEV_OE | Input | — | $V_{CCIO}$ | Optional |
| 6 | DEV_CLRn | Input | — | $V_{CCIO}$ | Optional |

**Notes to Table 8–18:**

(1) The CRC_ERROR pin is a dedicated open-drain output or an optional user I/O pin. Active high signal indicates that the error detection circuit has detected errors in the configuration SRAM bits. This pin is optional and is used when the CRC error detection circuit is enabled in the Quartus II software from the **Error Detection CRC** tab of the **Device and Pin Options** dialog box. When using this pin, connect it to an external 10-kΩ pull-up resistor to an acceptable voltage that satisfies the input voltage of the receiving device.

(2) To tri-state AS configuration pins in the AS configuration scheme, turn on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, nCSO, Data[0], and Data[1]/ASDO pins. Dual-purpose pins settings for these pins are ignored. To set these pins to different settings, turn off the **Enable input tri-state on active configuration pins in user mode** option and set the desired setting from the Dual-purpose Pins Setting menu.

**Table 8–19. Configuration Pin Summary for Cyclone IV E Devices (Part 1 of 3)**

| Bank | Description | Input/Output | Dedicated | Powered By | Configuration Mode |
|---|---|---|---|---|---|
| 1 | nCSO (1)<br>FLASH_nCE (2) | Output | — | $V_{CCIO}$ | AS, AP |
| 6 | CRC_ERROR (3) | Output | — | $V_{CCIO}$/Pull-up (4) | Optional, all modes |

**Table 8–19. Configuration Pin Summary for Cyclone IV E Devices (Part 2 of 3)**

| Bank | Description | Input/Output | Dedicated | Powered By | Configuration Mode |
|---|---|---|---|---|---|
| 1 | DATA[0] (1), (2) | Input | — | $V_{CCIO}$ | PS, FPP, AS |
| | | Bidirectional | | $V_{CCIO}$ | AP |
| 1 | DATA[1] (2)/ASDO (1) | Input | — | $V_{CCIO}$ | FPP |
| | | Output | | $V_{CCIO}$ | AS |
| | | Bidirectional | | $V_{CCIO}$ | AP |
| 8 | DATA[7..2] (2) | Input | — | $V_{CCIO}$ | FPP |
| | | Bidirectional | | $V_{CCIO}$ | AP |
| 8 | DATA[15..8] (2) | Bidirectional | — | $V_{CCIO}$ | AP |
| 6 | INIT_DONE | Output | — | Pull-up | Optional, all modes |
| 1 | nSTATUS | Bidirectional | Yes | Pull-up | All modes |
| 1 | nCE | Input | Yes | $V_{CCIO}$ | All modes |
| 1 | DCLK (1), (2) | Input | Yes | $V_{CCIO}$ | PS, FPP |
| | | Output | — | $V_{CCIO}$ | AS, AP |
| 6 | CONF_DONE | Bidirectional | Yes | Pull-up | All modes |
| 1 | TDI | Input | Yes | $V_{CCIO}$ | JTAG |
| 1 | TMS | Input | Yes | $V_{CCIO}$ | JTAG |
| 1 | TCK | Input | Yes | $V_{CCIO}$ | JTAG |
| 1 | nCONFIG | Input | Yes | $V_{CCIO}$ | All modes |
| 6 | CLKUSR | Input | — | $V_{CCIO}$ | Optional |
| 6 | nCEO | Output | — | $V_{CCIO}$ | Optional, all modes |
| 6 | MSEL[] | Input | Yes | $V_{CCINT}$ | All modes |
| 1 | TDO | Output | Yes | $V_{CCIO}$ | JTAG |
| 7 | PADD[14..0] | Output | — | $V_{CCIO}$ | AP |
| 8 | PADD[19..15] | Output | — | $V_{CCIO}$ | AP |
| 6 | PADD[23..20] | Output | — | $V_{CCIO}$ | AP |
| 1 | nRESET | Output | — | $V_{CCIO}$ | AP |
| 6 | nAVD | Output | — | $V_{CCIO}$ | AP |
| 6 | nOE | Output | — | $V_{CCIO}$ | AP |
| 6 | nWE | Output | — | $V_{CCIO}$ | AP |
| 5 | DEV_OE | Input | — | $V_{CCIO}$ | Optional, AP |

**Table 8–19. Configuration Pin Summary for Cyclone IV E Devices (Part 3 of 3)**

| Bank | Description | Input/Output | Dedicated | Powered By | Configuration Mode |
|---|---|---|---|---|---|
| 5 | DEV_CLRn | Input | — | $V_{CCIO}$ | Optional, AP |

**Notes to Table 8–19:**

(1) To tri-state AS configuration pins in the AS configuration scheme, turn-on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, nCSO, Data[0], and Data[1]/ASDO pins. Dual-purpose pins settings for these pins are ignored. To set these pins to different settings, turn off the **Enable input tri-state on active configuration pins in user mode** option and set the desired setting from the Dual-purpose Pins Setting menu.

(2) To tri-state AP configuration pins in the AP configuration scheme, turn-on the **Enable input tri-state on active configuration pins in user mode** option from the **Device and Pin Options** dialog box. This tri-states DCLK, Data[0..15], FLASH_nCE, and other AP pins. Dual-purpose pins settings for these pins are ignored. To set these pins to different settings, turn off the **Enable input tri-state on active configuration pins in user mode** option and set the desired setting from the Dual-purpose Pins Setting menu.

(3) The CRC_ERROR pin is not available in Cyclone IV E devices with 1.0-V core voltage.

(4) The CRC_ERROR pin is a dedicated open-drain output or an optional user I/O pin. Active high signal indicates that the error detection circuit has detected errors in the configuration SRAM bits. This pin is optional and is used when the CRC error detection circuit is enabled in the Quartus II software from the **Error Detection CRC** tab of the **Device and Pin Options** dialog box. When using this pin, connect it to an external 10-kΩ pull-up resistor to an acceptable voltage that satisfies the input voltage of the receiving device.

Table 8–20 describes the dedicated configuration pins. You must properly connect these pins on your board for successful configuration. You may not need some of these pins for your configuration schemes.

**Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 1 of 4)**

| Pin Name | User Mode | Configuration Scheme | Pin Type | Description |
|---|---|---|---|---|
| MSEL | N/A | All | Input | Configuration input that sets the Cyclone IV device configuration scheme. You must hardwire these pins to $V_{CCA}$ or GND. The MSEL pins have internal 9-kΩ pull-down resistors that are always active. |
| nCONFIG | N/A | All | Input | Configuration control input. Pulling this pin low with external circuitry during user mode causes the Cyclone IV 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 starts a reconfiguration. |
| nSTATUS | N/A | All | Bidirectional open-drain | The Cyclone IV device drives nSTATUS low immediately after power-up and releases it after the POR time.<br>■ Status output—if an error occurs during configuration, nSTATUS is pulled low by the target device.<br>■ Status input—if an external source (for example, another Cyclone IV device) drives the nSTATUS pin low during configuration or initialization, the target device enters an error state.<br>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 start a reconfiguration, you must pull nCONFIG low. |

**Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 2 of 4)**

| Pin Name | User Mode | Configuration Scheme | Pin Type | Description |
|---|---|---|---|---|
| CONF_DONE | N/A | All | Bidirectional open-drain | ■ Status output—the target Cyclone IV 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.<br>■ 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 in order for the device to initialize.<br>Driving CONF_DONE low after configuration and initialization does not affect the configured device. Do not connect bus holds or ADC to CONF_DONE pin. |
| nCE | N/A | All | Input | Active-low chip enable. The nCE pin activates the Cyclone IV device with a low signal to allow configuration. You must hold nCE pin low during configuration, initialization, and user-mode. In a single-device configuration, you must tie the nCE pin low. In a 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. You must hold the nCE pin low for successful JTAG programming of the device. |
| nCEO | N/A if option is on. I/O if option is off. | All | Output open-drain | Output that drives low when configuration is complete. In a single-device configuration, you can leave this pin floating or use it as a user I/O pin after configuration. In a multi-device configuration, this pin feeds the nCE pin of the next device. The nCEO of the last device in the chain is left floating or used as a user I/O pin after configuration.<br>If you use the nCEO pin to feed the nCE pin of the next device, use an external 10-kΩ pull-up resistor to pull the nCEO pin high to the $V_{CCIO}$ voltage of its I/O bank to help the internal weak pull-up resistor.<br>If you use the nCEO pin as a user I/O pin after configuration, set the state of the pin on the **Dual-Purpose Pin** settings. |
| nCSO, FLASH_nCE *(1)* | I/O | AS, AP *(2)* | Output | Output control signal from the Cyclone IV device to the serial configuration device in AS mode that enables the configuration device. This pin functions as nCSO in AS mode and FLASH_nCE in AP mode.<br>Output control signal from the Cyclone IV device to the parallel flash in AP mode that enables the flash. Connects to the CE# pin on the Micron P30 or P33 flash. *(2)*<br>This pin has an internal pull-up resistor that is always active. |

**Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 3 of 4)**

| Pin Name | User Mode | Configuration Scheme | Pin Type | Description |
|---|---|---|---|---|
| DCLK (1) | N/A | PS, FPP, AS, AP (2) | Input (PS, FPP) (2) | In PS and FPP configuration, DCLK is the clock input used to clock data from an external source into the target Cyclone IV device. Data is latched into the device on the rising edge of DCLK.<br>In AS mode, DCLK is an output from the Cyclone IV device that provides timing for the configuration interface. It has an internal pull-up resistor (typically 25 kΩ) that is always active.<br>In AP mode, DCLK is an output from the Cyclone IV E device that provides timing for the configuration interface. (2)<br>In AS or AP configuration schemes, this pin is driven into an inactive state after configuration completes. Alternatively, in active schemes, 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 in user mode. Toggling this pin after configuration does not affect the configured device. |
| | I/O | | Output (AS, AP) | |
| DATA[0] (1) | I/O | PS, FPP, AS, AP (2) | Input (PS, FPP, AS). Bidirectional (AP) (2) | Data input. In serial configuration modes, bit-wide configuration data is presented to the target Cyclone IV device on the DATA[0] pin.<br>In AS mode, DATA[0] has an internal pull-up resistor that is always active. After AS configuration, DATA[0] is a dedicated input pin with optional user control.<br>After PS or FPP configuration, DATA[0] is available as a user I/O pin. The state of this pin depends on the **Dual-Purpose Pin** settings.<br>After AP configuration, DATA[0] is a dedicated bidirectional pin with optional user control. (2) |
| DATA[1]/ ASDO (1) | I/O | FPP, AS, AP (2) | Input (FPP). Output (AS). Bidirectional (AP) (2) | The DATA[1] pin functions as the ASDO pin in AS mode. Data input in non-AS mode. Control signal from the Cyclone IV device to the serial configuration device in AS mode used to read out configuration data.<br>In AS mode, DATA[1] has an internal pull-up resistor that is always active. After AS configuration, DATA[1] is a dedicated output pin with optional user control.<br>In a PS configuration scheme, DATA[1] functions as a user I/O pin during configuration, which means it is tri-stated.<br>After FPP configuration, DATA[1] is available as a user I/O pin and the state of this pin depends on the **Dual-Purpose Pin** settings.<br>In an AP configuration scheme, for Cyclone IV E devices only, the byte-wide or word-wide configuration data is presented to the target Cyclone IV E device on DATA[7..0] or DATA[15..0], respectively. After AP configuration, DATA[1] is a dedicated bidirectional pin with optional user control. (2) |

**Table 8–20. Dedicated Configuration Pins on the Cyclone IV Device (Part 4 of 4)**

| Pin Name | User Mode | Configuration Scheme | Pin Type | Description |
|---|---|---|---|---|
| DATA[7..2] | I/O | FPP, AP (2) | Inputs (FPP). Bidirectional (AP) (2) | In an AS or PS configuration scheme, DATA[7..2] function as user I/O pins during configuration, which means they are tri-stated.<br>After FPP configuration, DATA[7..2] are available as user I/O pins and the state of these pin depends on the **Dual-Purpose Pin** settings.<br>In an AP configuration scheme, for Cyclone IV E devices only, the byte-wide or word-wide configuration data is presented to the target Cyclone IV E device on DATA[7..0] or DATA[15..0], respectively. After AP configuration, DATA[7..2] are dedicated bidirectional pins with optional user control. (2) |
| DATA[15..8] | I/O | AP (2) | Bidirectional | Data inputs. Word-wide configuration data is presented to the target Cyclone IV E device on DATA[15..0].<br>In a PS, FPP, or AS configuration scheme, DATA[15:8] function as user I/O pins during configuration, which means they are tri stated.<br>After AP configuration, DATA[15:8] are dedicated bidirectional pins with optional user control. |
| PADD[23..0] | I/O | AP (2) | Output | In AP mode, it is a 24-bit address bus from the Cyclone IV E device to the parallel flash. Connects to the A[24:1] bus on the Micron P30 or P33 flash. |
| nRESET | I/O | AP (2) | Output | Active-low reset output. Driving the nRESET pin low resets the parallel flash. Connects to the RST# pin on the Micron P30 or P33 flash. |
| nAVD | I/O | AP (2) | Output | Active-low address valid output. Driving the nAVD pin low during read or write operation indicates to the parallel flash that a valid address is present on the PADD[23..0] address bus. Connects to the ADV# pin on the Micron P30 or P33 flash. |
| nOE | I/O | AP (2) | Output | Active-low output enable to the parallel flash. During the read operation, driving the nOE pin low enables the parallel flash outputs (DATA[15..0]). Connects to the OE# pin on the Micron P30 or P33 flash. |
| nWE | I/O | AP (2) | Output | Active-low write enable to the parallel flash. During the write operation, driving the nWE pin low indicates to the parallel flash that data on the DATA[15..0] bus is valid. Connects to the WE# pin on the Micron P30 or P33 flash. |

**Note to Table 8–20:**

(1) If you are accessing the EPCS device with the ALTASMI_PARALLEL megafunction or your own user logic in user mode, in the **Device and Pin Options** window of the Quartus II software, in the **Dual-Purpose Pins** category, select **Use as regular I/O** for this pin.

(2) The AP configuration scheme is for Cyclone IV E devices only.

Table 8–21 lists the optional configuration pins. If you do not enable these optional configuration pins 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 8–21. 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. This pin is enabled by turning on the **Enable user-supplied start-up clock (CLKUSR)** option in the Quartus II software.<br>In AS configuration for Cyclone IV GX devices, you can use this pin as an external clock source to generate the DCLK by changing the clock source option in the Quartus II software in the **Configuration** tab of the **Device and Pin Options** dialog box. |
| INIT_DONE | N/A if option is on. I/O if option is off. | Output open-drain | Status pin is used to indicate when the device has initialized and is in user-mode. When nCONFIG is low, the INIT_DONE pin is tri-stated and pulled high due to an external 10-kΩ pull-up resistor 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 and the device enters user mode. Thus, the monitoring circuitry must be able to detect a low-to-high transition. This pin is enabled by turning on the **Enable INIT_DONE output** option in the Quartus II software.<br>The functionality of this pin changes if the **Enable OCT_DONE** option is enabled in the Quartus II software. This option controls whether the INIT_DONE signal is gated by the OCT_DONE signal, which indicates the power-up on-chip termination (OCT) calibration is complete. If this option is turned off, the INIT_DONE signal is not gated by the OCT_DONE signal. |
| 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. This pin is enabled 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. You can enable this pin by turning on the **Enable device-wide reset (DEV_CLRn)** option in the Quartus II software. |

# Remote System Upgrade

Cyclone IV devices support remote system upgrade in AS and AP configuration schemes. You can also implement remote system upgrade with advanced Cyclone IV features such as real-time decompression of configuration data in the AS configuration scheme.

Remote system upgrade is not supported in a multi-device configuration chain for any configuration scheme.

## Functional Description

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

Configuration memory refers to serial configuration devices (EPCS) or supported parallel flash memory, depending on the configuration scheme that is used.

The remote system upgrade process of the Cyclone IV device consists of the following steps:

1. A Nios II processor (or user logic) implemented in the Cyclone IV device logic array receives new configuration data from a remote location. The connection to the remote source is 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) writes this new configuration data into a configuration memory.
3. The Nios II processor (or user logic) starts a reconfiguration cycle with the new or updated configuration data.
4. The dedicated remote system upgrade circuitry detects and recovers from any error that might occur during or after the reconfiguration cycle and provides error status information to the user design.

Figure 8–30 shows the steps required for performing remote configuration updates (the numbers in Figure 8–30 coincide with steps 1–3).

**Figure 8–30. Functional Diagram of Cyclone IV Device Remote System Upgrade**

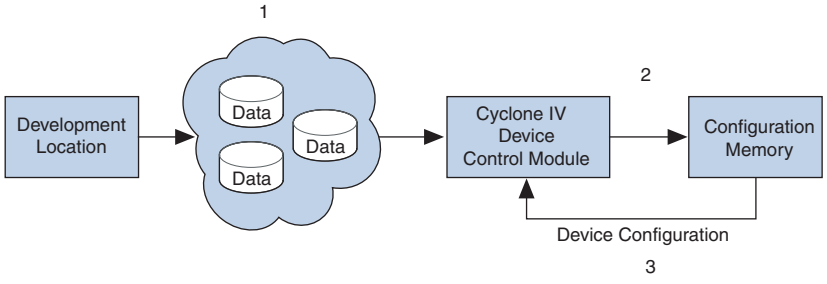
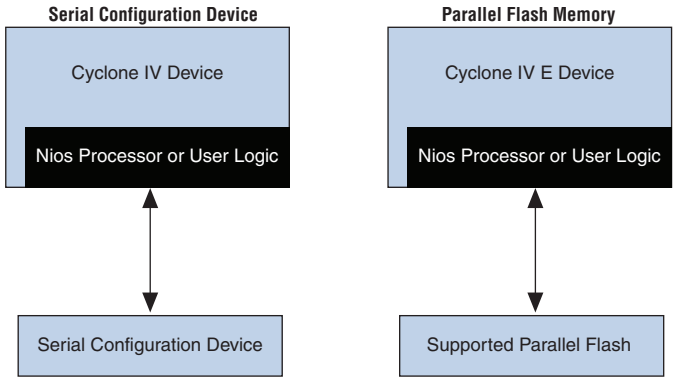


Figure 8–31 shows the block diagrams to implement remote system upgrade in Cyclone IV devices.

**Figure 8–31. Remote System Upgrade Block Diagrams for AS and AP Configuration Schemes**




The MSEL pin setting in the remote system upgrade mode is the same as the standard configuration mode. Standard configuration mode refers to normal Cyclone IV device configuration mode with no support for remote system upgrades (the remote system upgrade circuitry is disabled). When using remote system upgrade in Cyclone IV devices, you must enable the remote update mode option setting in the Quartus II software.

## Enabling Remote Update

You can enable or disable remote update for Cyclone IV devices in the Quartus II software before design compilation (in the Compiler Settings menu). To enable remote update in the compiler settings of the project, perform the following steps:

1. On the Assignments menu, click **Device**. The **Settings** dialog box appears.
2. Click **Device and Pin Options**. The **Device and Pin Options** dialog box appears.
3. Click the **Configuration** tab.
4. From the **Configuration Mode** list, select **Remote**.
5. Click **OK**.
6. In the **Settings** dialog box, click **OK**.

### Configuration Image Types

When using remote system upgrade, Cyclone 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 device that performs certain user-defined functions. Each device in your system requires one factory image or with 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 with the dedicated circuitry. Application images implement user-defined functionality in the target Cyclone IV device. You can include the default application image functionality in the factory image.

# Remote System Upgrade Mode

In remote update mode, Cyclone IV devices load the factory configuration image after power up. The user-defined factory configuration determines the application configuration to be loaded and triggers a reconfiguration cycle. The factory configuration can also contain application logic.

When used with configuration memory, the remote update mode allows an application configuration to start at any flash sector boundary. Additionally, the remote update mode features a user watchdog timer that can detect functional errors in an application configuration.

## Remote Update Mode

In AS configuration scheme, when a Cyclone IV device is first powered up in remote update, it loads the factory configuration located at address `boot_address[23:0] = 24b'0`. Altera recommends storing the factory configuration image for your system at boot address `24b'0`, which corresponds to the start address location `0x000000` in the serial configuration device. A factory configuration image is a bitstream for the Cyclone IV device in your system that is programmed during production and is the fall-back image when an error occurs. This image is stored in non-volatile memory and is never updated or modified using remote access.

When you use the AP configuration in Cyclone IV E devices, the Cyclone IV E device loads the default factory configuration located at the following address after device power-up in remote update mode:
`boot_address[23:0] = 24'h010000 = 24'b1 0000 0000 0000 0000`.

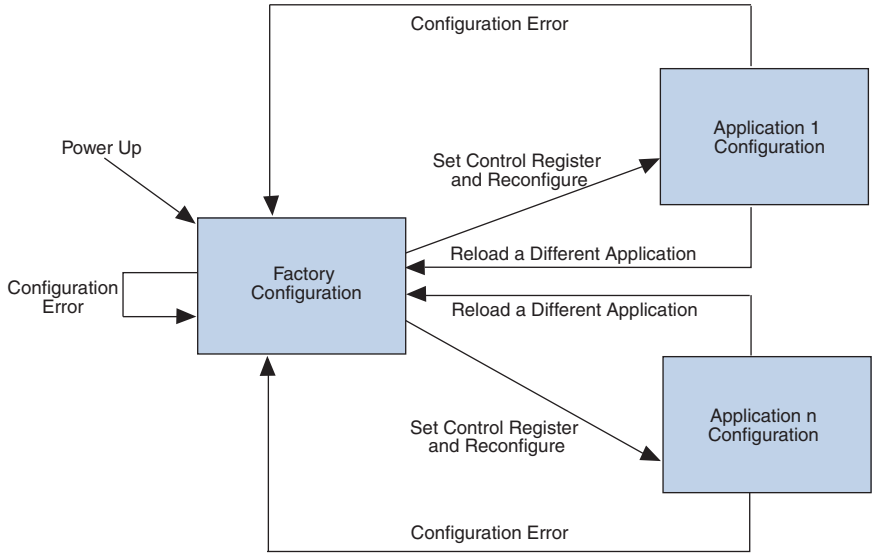
You can change the default factory configuration address to any desired address using the `APFC_BOOT_ADDR` JTAG instruction. The factory configuration image is stored in non-volatile memory and is never updated or modified using remote access. This corresponds to the default start address location `0x010000` represented in 16-bit word addressing (or the updated address if the default address is changed) in the supported parallel flash memory. For more information about the application of the `APFC_BOOT_ADDR` JTAG instruction in AP configuration scheme, refer to the "JTAG Instructions" on page 8–57.

The factory configuration image is user-designed and contains soft logic (Nios II processor or state machine and the remote communication interface) 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 the new configuration data in the local non-volatile memory device
- Determine the application configuration to be loaded into the Cyclone IV device
- Enable or disable the user watchdog timer and load its time-out value (optional)
- Instruct the dedicated remote system upgrade circuitry to start a reconfiguration cycle

Figure 8–32 shows the transitions between the factory configuration and application configuration in remote update mode.

**Figure 8–32. Transitions Between Configurations in Remote Update Mode**




After power up or a configuration error, the factory configuration logic writes the remote system upgrade control register to specify the address of the application configuration to be loaded. The factory configuration also specifies whether or not to enable the user watchdog timer for the application configuration and, if enabled, specifies the timer setting.

Only valid application configurations designed for remote update mode include the logic to reset the timer in user mode. For more information about the user watchdog timer, refer to the "User Watchdog Timer" on page 8–79.

If there is an error while loading the application configuration, the remote system upgrade status register is written by the dedicated remote system upgrade circuitry of the Cyclone IV device to specify the cause of the reconfiguration.

The following actions cause the remote system upgrade status register to be written:

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

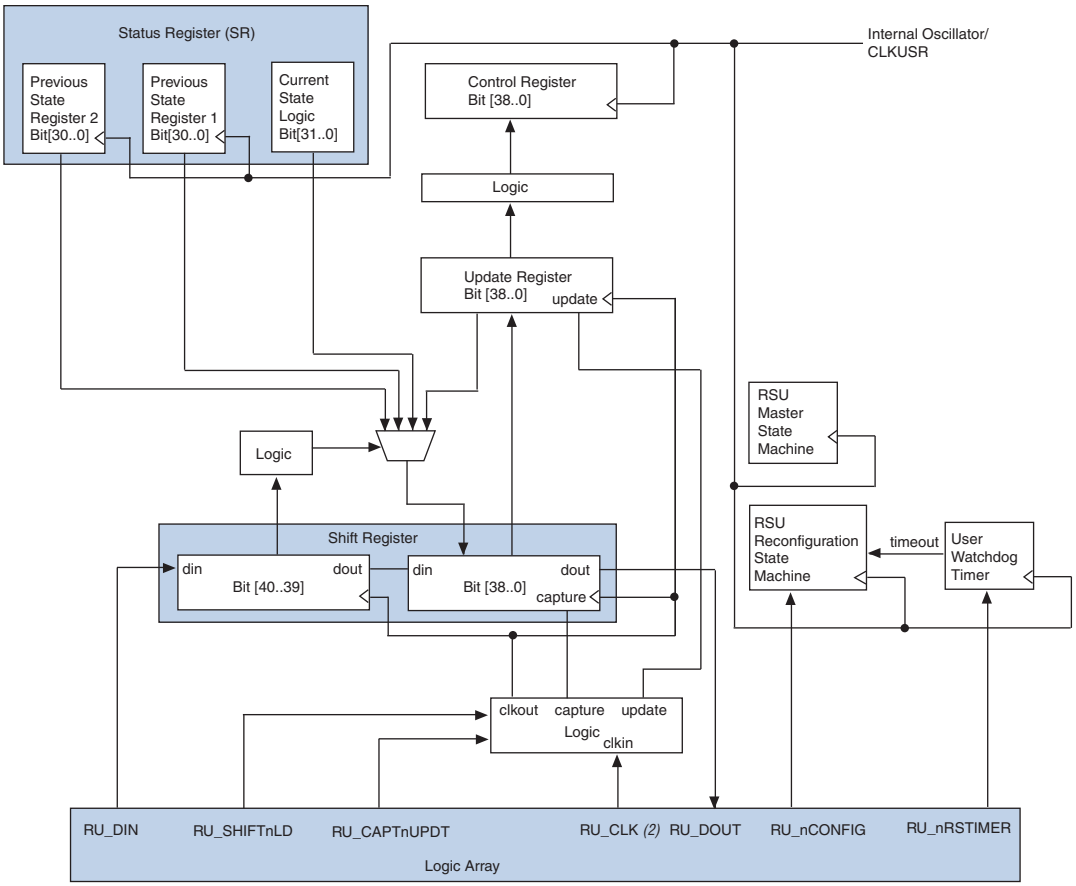
The Cyclone IV device automatically loads the factory configuration when an error occurs. This user-designed factory configuration reads the remote system upgrade status register to determine the reason for reconfiguration. Then the factory configuration takes the appropriate error recovery steps and writes to the remote system upgrade control register to determine the next application configuration to be loaded.

When Cyclone IV devices successfully load the application configuration, they enter user mode. In user mode, the soft logic (the Nios II processor or state machine and the remote communication interface) assists the Cyclone 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, determines the valid application configuration to load, writes the remote system upgrade control register accordingly, and starts system reconfiguration.

## Dedicated Remote System Upgrade Circuitry

This section describes the implementation of the Cyclone IV device remote system upgrade dedicated circuitry. The remote system upgrade circuitry is implemented in hard logic. This dedicated circuitry interfaces with the user-defined factory application configurations implemented in the Cyclone 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 state machines that control those components. Figure 8–33 shows the data path of the remote system upgrade block.

**Figure 8–33. Remote System Upgrade Circuit Data Path** *(1)*




**Notes to Figure 8–33:**

(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.

(2) The RU_CLK refers to the ALTREMOTE_UPDATE megafunction block "clock" input. For more information, refer to the *Remote Update Circuitry (ALTREMOTE_UPDATE) Megafunction User Guide.*

## Remote System Upgrade Registers

The remote system upgrade block contains a series of registers that stores the configuration addresses, watchdog timer settings, and status information. Table 8–22 lists these registers.

**Table 8–22. 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. Write access is enabled in remote update mode for factory configurations to allow writing to the update register. Write access is disabled for all application configurations in remote update mode. |
| Control register | This register contains the current configuration address, the user watchdog timer settings, one option bit for checking early CONF_DONE, and one option bit for selecting the internal oscillator as the startup state machine clock. During a read operation in an application configuration, this register is read into the shift register. When a reconfiguration cycle is started, 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 read in a factory configuration, this register is read into the shift register. |
| Status register | This register is written 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 control and status registers of the remote system upgrade are clocked by the 10-MHz internal oscillator (the same oscillator that controls the user watchdog timer) or the CLKUSR. However, the shift and update registers of the remote system upgrade are clocked by the maximum frequency of 40-MHz user clock input (RU_CLK). There is no minimum frequency for RU_CLK.

### Remote System Upgrade Control Register

The remote system upgrade control register stores the application configuration address, the user watchdog timer settings, and option bits for a application configuration. In remote update mode for the AS configuration scheme, the control register address bits are set to all zeros (24'b0) at power up to load the AS factory configuration. In remote update mode for the AP configuration scheme, the control register address bits are set to 24'h010000 (24'b1 0000 0000 0000 0000) at power up to load the AP default factory configuration. However, for the AP configuration scheme, you can change the default factory configuration address to any desired address using the APFC_BOOT_ADDR JTAG instruction. Additionally, a factory configuration in remote update mode has write access to this register.

Figure 8–34 shows the control register bit positions. Table 8–23 defines the control register bit contents. The numbers in Figure 8–34 show the bit position of a setting in a register. For example, bit number 35 is the enable bit for the watchdog timer.

**Figure 8–34. Remote System Upgrade Control Register**

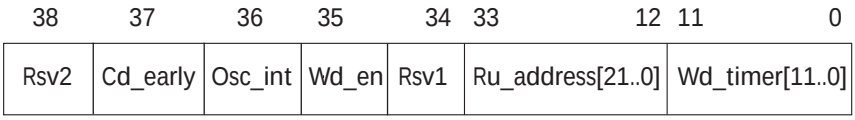



**Table 8–23. Remote System Upgrade Control Register Contents**

| Control Register Bit | Value | Definition |
|---|---|---|
| Wd_timer[11..0] | 12'b000000000000 | User watchdog time-out value (most significant 12 bits of 29-bit count value: {Wd_timer[11..0],17'b1000}) |
| Ru_address[21..0] | 22'b0000000000000000000000 | Configuration address (most significant 22 bits of 24-bit boot address value: boot_address[23:0] = {Ru_address[21..0],2'b0}) |
| Rsv1 | 1'b0 | Reserved bit |
| Wd_en | 1'b1 | User watchdog timer enable bit |
| Osc_int (1) | 1'b1 | Internal oscillator as startup state machine clock enable bit |
| Cd_early (1) | 1'b1 | Early CONF_DONE check |
| Rsv2 | 1'b1 | Reserved bit |

**Note to Table 8–23:**

(1) Option bit for the application configuration.

When enabled, the early CONF_DONE check (Cd_early) option bit ensures that there is a valid configuration at the boot address specified by the factory configuration and that it is of the proper size. If an invalid configuration is detected or the CONF_DONE pin is asserted too early, the device resets and then reconfigures the factory configuration image. The internal oscillator (as the startup state machine clock [Osc_int] option bit) ensures a functional startup clock to eliminate the hanging of startup. When all option bits are turned on, they provide complete coverage for the programming and startup portions of the application configuration. Altera recommends turning on both the Cd_early and Osc_int option bits.

The Cd_early and Osc_int option bits for the application configuration must be turned on by the factory configuration.

### Remote System Upgrade Status Register

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

- Cyclical redundancy check (CRC) error during application configuration
- nSTATUS assertion by an external device due to an error
- Cyclone IV device logic array triggers a reconfiguration cycle, possibly after downloading a new application configuration image

- External configuration reset (nCONFIG) assertion
- User watchdog timer time out

Table 8–24 lists the contents of the current state logic in the status register, when the remote system upgrade master state machine is in factory configuration or application configuration accessing the factory information or application information, respectively. The status register bit in Table 8–24 lists the bit positions in a 32-bit logic.

**Table 8–24. Remote System Upgrade Current State Logic Contents In Status Register**

| Remote System Upgrade Master State Machine | Status Register Bit | Definition | Description |
|---|---|---|---|
| Factory information (1) | 31:30 | Master state machine current state | The current state of the remote system upgrade master state machine |
| | 29:24 | Reserved bits | Padding bits that are set to all 0's |
| | 23:0 | Boot address | The current 24-bit boot address that was used by the configuration scheme as the start address to load the current configuration. |
| Application information 1 (2) | 31:30 | Master state machine current state | The current state of the remote system upgrade master state machine |
| | 29 | User watchdog timer enable bit | The current state of the user watchdog enable, which is active high |
| | 28:0 | User watchdog timer time-out value | The current entire 29-bit watchdog time-out value. |
| Application information 2 (2) | 31:30 | Master state machine current state | The current state of the remote system upgrade master state machine |
| | 29:24 | Reserved bits | Padding bits that are set to all 0's |
| | 23:0 | Boot address | The current 24-bit boot address that was used as the start address to load the current configuration |

**Notes to Table 8–24:**

(1) The remote system upgrade master state machine is in factory configuration.

(2) The remote system upgrade master state machine is in application configuration.

The previous two application configurations are available in the previous state registers (previous state register 1 and previous state register 2), but only for debugging purposes.

Table 8–25 lists the contents of previous state register 1 and previous state register 2 in the status register. The status register bit in Table 8–25 shows the bit positions in a 3-bit register. The previous state register 1 and previous state register 2 have the same bit definitions. The previous state register 1 reflects the current application configuration and the previous state register 2 reflects the previous application configuration.

**Table 8–25. Remote System Upgrade Previous State Register 1 and Previous State Register 2 Contents in Status Register**

| Status Register Bit | Definition | Description |
|---|---|---|
| 30 | nCONFIG source | One-hot, active-high field that describes the reconfiguration source that caused the Cyclone IV device to leave the previous application configuration. If there is a tie, the higher bit order indicates precedence. For example, if nCONFIG and remote system upgrade nCONFIG reach the reconfiguration state machine at the same time, the nCONFIG precedes the remote system upgrade nCONFIG. |
| 29 | CRC error source | |
| 28 | nSTATUS source | |
| 27 | User watchdog timer source | |
| 26 | Remote system upgrade nCONFIG source | |
| 25:24 | Master state machine current state | The state of the master state machine during reconfiguration causes the Cyclone IV device to leave the previous application configuration. |
| 23:0 | Boot address | The address used by the configuration scheme to load the previous application configuration. |

If a capture is inappropriately done while capturing a previous state before the system has entered remote update application configuration for the first time, a value outputs from the shift register to indicate that the capture is incorrectly called.

## Remote System Upgrade State Machine

The remote system upgrade control and update registers have identical bit definitions, but serve different roles (Table 8–22 on page 8–75). 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, and the control register writes are controlled by the remote system upgrade state machine.

In factory configurations, the user logic should send the option bits (Cd_early and Osc_int), the configuration address, and watchdog timer settings for the next application configuration bit to the update register. When the logic array configuration reset (RU_nCONFIG) goes high, the remote system upgrade state machine updates the control register with the contents of the update register and starts system reconfiguration from the new application page.

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

If there is an error or reconfiguration trigger condition, the remote system upgrade state machine directs the system to load a factory or application configuration (based on mode and error condition) by setting the control register accordingly.

Table 8–26 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 8–26. Control Register Contents After an Error or Reconfiguration Trigger Condition**

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

## User Watchdog Timer

The user watchdog timer prevents a faulty application configuration from indefinitely stalling the device. The system uses the timer to detect functional errors after an application configuration is successfully loaded into the Cyclone 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 remote system upgrade circuitry appends 17'b1000 to form the 29-bits value for the watchdog timer. The granularity of the timer setting is $2^{17}$ cycles. The cycle time is based on the frequency of the 10-MHz internal oscillator or CLKUSR (maximum frequency of 40 MHz).

Table 8–27 lists the operating range of the 10-MHz internal oscillator.

**Table 8–27. 10-MHz Internal Oscillator Specifications**

| Minimum | Typical | Maximum | Unit |
|---|---|---|---|
| 5 | 6.5 | 10 | MHz |

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 the 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.

Errors during configuration are detected by the CRC engine. Functional errors must not exist in the factory configuration because it is stored and validated during production and is never updated remotely.

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

Implementation in your design requires a remote system upgrade interface between the Cyclone IV device logic array and remote system upgrade circuitry. You must also generate configuration files for production and remote programming of the system configuration memory. The Quartus II software provides these features.

The two implementation options, ALTREMOTE_UPDATE megafunction and remote system upgrade atom, are 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.

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

# Document Revision History

Table 8–28 lists the revision history for this chapter.

**Table 8–28. Document Revision History (Part 1 of 2)**

| Date | Version | Changes |
|---|---|---|
| May 2013 | 1.7 | ■ Added Table 8–6.<br>■ Updated Table 8–9 to add new device options and packages.<br>■ Updated Figure 8–16 and Figure 8–22 to include user mode.<br>■ Updated the “Dedicated” column for DATA[0] and DCLK in Table 8–19.<br>■ Updated the “User Mode” and “Pin Type” columns for DCLK in Table 8–20. |
| February 2013 | 1.6 | Updated Table 8–9 to add new device options and packages. |
| October 2012 | 1.5 | ■ Updated “AP Configuration Supported Flash Memories”, “Configuration Data Decompression”, and “Overriding the Internal Oscillator” sections.<br>■ Updated Figure 8–3, Figure 8–4, Figure 8–5, Figure 8–7, Figure 8–8, Figure 8–9, Figure 8–10, and Figure 8–11.<br>■ Updated Table 8–2, Table 8–8, Table 8–12, Table 8–13, Table 8–18, and Table 8–19. |
| November 2011 | 1.4 | ■ Added information about how to gain control of EPCS pins.<br>■ Updated “Reset”, “Single-Device AS Configuration”, “Single-Device AP Configuration”, and “Overriding the Internal Oscillator” sections.<br>■ Added Table 8–7.<br>■ Updated Table 8–6 and Table 8–19.<br>■ Updated Figure 8–3, Figure 8–4, and Figure 8–5. |
| December 2010 | 1.3 | ■ Updated for the Quartus II software version 10.1 release.<br>■ Added Cyclone IV E new device package information.<br>■ Updated Table 8–7, Table 8–10, and Table 8–11.<br>■ Minor text edits. |

**Table 8–28. Document Revision History (Part 2 of 2)**

| Date | Version | Changes |
|---|---|---|
| July 2010 | 1.2 | Updated for the Quartus II software 10.0 release:<br>■ Updated "Power-On Reset (POR) Circuit", "Configuration and JTAG Pin I/O Requirements", and "Reset" sections.<br>■ Updated Figure 8–10.<br>■ Updated Table 8–16 and Table 8–17. |
| February 2010 | 1.1 | Updated for the Quartus II software 9.1 SP1 release:<br>■ Added "Overriding the Internal Oscillator" and "AP Configuration (Supported Flash Memories)" sections.<br>■ Updated "JTAG Instructions" section.<br>■ Added Table 8–6.<br>■ Updated Table 8–2, Table 8–3, Table 8–4, Table 8–6, Table 8–11, Table 8–13, Table 8–14, Table 8–15, and Table 8–18.<br>■ Updated Figure 8–4, Figure 8–5, Figure 8–6, Figure 8–13, Figure 8–14, Figure 8–15, Figure 8–17, Figure 8–18, Figure 8–23, Figure 8–24, Figure 8–25, Figure 8–26, Figure 8–27, Figure 8–28, and Figure 8–29. |
| November 2009 | 1.0 | Initial release. |