

## Introduction

APEX™ 20K, FLEX® 10K, and FLEX 6000 devices can be configured using one of six configuration schemes, which are ideal for a variety of systems. All configuration schemes use either a microprocessor or configuration device. See [Table 1](#).

<b>Table 1. Configuration Schemes</b>		
<b>Configuration Scheme</b>	<b>Device Family</b>	<b>Typical Use</b>
Configuration Device	APEX 20K FLEX 10K FLEX 6000	Configuration with the EPC2, EPC1, or EPC1441 configuration devices.
Passive Serial (PS)	APEX 20K FLEX 10K FLEX 6000	Configuration with a serial synchronous microprocessor interface and the MasterBlaster™ communications cable, ByteBlasterMV™ parallel port download cable, or BitBlaster™ serial download cable. <a href="#">(1)</a> , <a href="#">(2)</a> , <a href="#">(4)</a>
Passive Parallel Synchronous (PPS)	APEX 20K FLEX 10K	Configuration with a parallel synchronous microprocessor interface.
Passive Parallel Asynchronous (PPA)	APEX 20K FLEX 10K	Configuration with a parallel asynchronous microprocessor interface. In this scheme, the microprocessor treats the target device as memory.
Passive Serial Asynchronous (PSA)	FLEX 6000	Configuration with a serial asynchronous microprocessor interface.
Joint Test Action Group (JTAG)	APEX 20K FLEX 10K	Configuration through the IEEE Std. 1149.1 (JTAG) pins. <a href="#">(3)</a>

### Notes:

- (1) The MasterBlaster communications cable uses a standard PC serial or universal serial bus (USB) hardware interface that downloads configuration data to APEX 20K, FLEX 10K, and FLEX 6000 devices. It supports operation with  $V_{CC}$  at 5.0 V, 3.3 V, or 2.5 V and is supported by the Quartus™ software and the MAX+PLUS® II software versions 9.3 and higher. For more information on the MasterBlaster cable, see the [MasterBlaster Serial/USB Communications Cable Data Sheet](#).
- (2) The ByteBlaster™ download cable is obsolete and is replaced by the ByteBlasterMV parallel port download cable.
- (3) Although you cannot configure FLEX 6000 devices through the JTAG pins, you can perform JTAG boundary-scan testing.
- (4) The BitBlaster serial download cable is not supported by the Quartus software and can not be used to configure APEX devices.

This application note discusses how to configure one or more APEX 20K (including APEX 20KE), FLEX 10K (including FLEX 10KE and FLEX 10KA), and FLEX 6000 devices. This application note should be used together with the following documents:

- [APEX 20K Programmable Logic Device Family Data Sheet](#)
- [FLEX 10K Embedded Programmable Logic Family Data Sheet](#)
- [FLEX 10KE Embedded Programmable Logic Family Data Sheet](#)
- [FLEX 6000 Programmable Logic Device Family Data Sheet](#)
- [Configuration Devices for APEX & FLEX Devices Data Sheet](#)



If appropriate, illustrations in this application note show devices with generic “APEX 20K”, “FLEX 10K”, and “FLEX 6000” labels to indicate they are valid for all APEX 20K, FLEX 10K, and FLEX 6000 devices.

## Contents

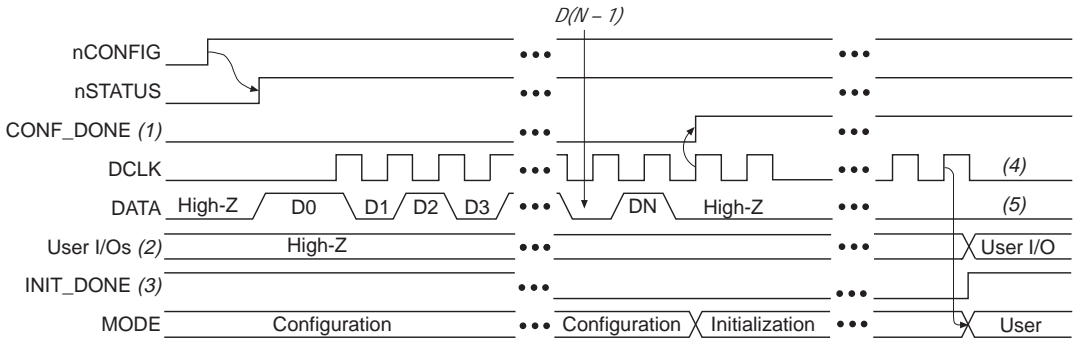
This application note provides information on the following topics:

Device Configuration Overview .....	2
Configuration Schemes .....	6
Configuration Device .....	6
PS Configuration with a Download Cable .....	16
PS Configuration with a Microprocessor .....	23
PPS Configuration (APEX 20K & FLEX 10K Devices Only).....	28
PSA Configuration (FLEX 6000 Devices Only).....	32
PPA Configuration (APEX 20K & FLEX 10K Devices Only).....	38
JTAG Programming & Configuration (APEX 20K & FLEX 10K Devices Only).....	45
JTAG Programming & Configuration of Multiple Devices (APEX 20K & FLEX 10K Devices Only).....	48
Jam Programming & Test Language .....	50
Combining Different Configuration Schemes .....	51
Device Options .....	55
Device Configuration Pins.....	58
Device Configuration Files .....	64
Device Configuration .....	67
Configuration Reliability .....	68
Board Layout Tips.....	69

## Device Configuration Overview

During device operation, APEX 20K, FLEX 10K, and FLEX 6000 devices store configuration data in SRAM cells. Because SRAM memory is volatile, the SRAM cells must be loaded with configuration data each time the device powers up. After the APEX 20K, FLEX 10K, or FLEX 6000 device is configured, its registers and I/O pins must be initialized. After initialization, the device enters user mode for in-system operation.

**Figure 1** shows the state of the device during the configuration, initialization, and user modes.

**Figure 1. APEX 20K, FLEX 10K & FLEX 6000 Configuration Cycle****Notes:**

- (1) During initial power-up and configuration, CONF\_DONE is low. After configuration, CONF\_DONE goes high. If the device is reconfigured, CONF\_DONE goes low after nCONFIG is driven low.
- (2) User I/O pins are tri-stated during configuration. APEX 20K and FLEX 10KE devices also have a weak pull-up resistor on I/O pins during configuration. After initialization, the user I/O pins perform the function assigned in the user's design.
- (3) When used, the optional INIT\_DONE signal is high when nCONFIG is low before configuration and during approximately the first 40 clock cycles of configuration.
- (4) DCLK should not be left floating. It should be driven high or low, whichever is more convenient.
- (5) DATA (FLEX 6000 devices) and DATA0 (APEX 20K and FLEX 10K devices) should not be left floating. It should be driven high or low, whichever is more convenient.

The configuration data for APEX 20K, FLEX 10K, and FLEX 6000 devices can be loaded using an active or passive configuration scheme. When using an active configuration scheme with a configuration device, both the target device and configuration device generate the control and synchronization signals. When both devices are ready to begin configuration, the configuration device sends data to the APEX 20K, FLEX 10K, or FLEX 6000 device.

When using any passive configuration scheme, the APEX 20K, FLEX 10K, or FLEX 6000 device is incorporated into a system with an intelligent host, such as a microprocessor, that controls the configuration process. The host supplies configuration data from a storage device (e.g., a hard disk, RAM, or other system memory). When using passive configuration, you can change the target device's functionality while the system is in operation by reconfiguring it. You can also perform in-field upgrades by distributing a new programming file to system users.

You select an APEX 20K or FLEX 10K device's configuration scheme by driving its MSEL0 and MSEL1 pins either high or low as shown in [Table 2](#).

MSEL1	MSEL0	Configuration Scheme
0	0	Configuration device or passive serial.
1	0	Passive parallel synchronous.
1	1	Passive parallel asynchronous.

**Note:**

- (1) The MSEL1 and MSEL0 pins can be used to change configuration modes between configurations. However, they are generally connected to  $V_{CC}$  or ground.

For FLEX 6000 devices, the MSEL pin controls configuration, as shown in [Table 3](#).

MSEL	Configuration Scheme
0	Configuration device or passive serial scheme, using the MasterBlaster, ByteBlasterMV, or BitBlaster cables.
1	Passive serial asynchronous.

**Note:**

- (1) The MSEL pin can be used to change configuration modes between configurations. However, it is generally connected to  $V_{CC}$  or ground.



Device option bits and device configuration pins are discussed further in [“Device Options” on page 55](#) and [“Device Configuration Pins” on page 58](#), respectively.

[Table 4](#) summarizes the approximate configuration file size required for each APEX 20K, FLEX 10K, and FLEX 6000 device. To calculate the amount of storage space required for multi-device configurations, simply add each device’s file size together.

**Table 4. APEX 20K, FLEX 10K & FLEX 6000 Configuration File Sizes** *Note (1)*

Device	Data Size (Bits)	Data Size (Kbytes)
EP20K1000E	8,938,000	1,092
EP20K600E	5,564,000	680
EP20K400, EP20K400E	3,878,000	474
EP20K200	1,950,000	238
EP20K100	985,000	121
EPF10K250A	3,300,000	403
EPF10K200E	2,757,000	337
EPF10K130E	1,840,000	225
EPF10K130V	1,600,000	194
EPF10K100E	1,336,000	164
EPF10K100, EPF10K100A, EPF10K100B	1,200,000	146
EPF10K70	892,000	109
EPF10K50E	785,000	96
EPF10K50, EPF10K50V	621,000	76
EPF10K40	498,000	61
EPF10K30E	470,000	58
EPF10K30A	406,000	50
EPF10K30	376,000	46
EPF10K20	231,000	29
EPF10K10A	120,000	15
EPF10K10	118,000	15
EPF6024A	398,000	49
EPF6016, EPF6016A	260,000	32
EPF6010A	260,000	32

**Note:**

(1) Raw Binary Files (.rbf) were used to determine these file sizes.

The numbers in [Table 4](#) should only be used to estimate the file size before design compilation. The exact file size may vary because different MAX+PLUS II or Quartus software versions may add a slightly different number of padding bits during programming. However, for any specific version of the MAX+PLUS II or Quartus software, any design targeted for the same device has the same configuration file size.

Table 5 lists Altera configuration devices that can be used to configure APEX 20K, FLEX 10K, and FLEX 6000 devices.

<b>Device</b>	<b>Description</b>
EPC2	1,695,680 × 1-bit device with 5.0-V or 3.3-V operation
EPC1	1,046,496 × 1-bit device with 5.0-V or 3.3-V operation
EPC1441	440,800 × 1-bit device with 5.0-V or 3.3-V operation

You can use the data from Tables 4 and 5 to determine the number of configuration devices required to configure your device. For example, to configure one EPF10K100 device, you need two EPC1 devices but only one EPC2 device. Similarly, one EP20K400 device requires three EPC2 devices.

## Configuration Schemes

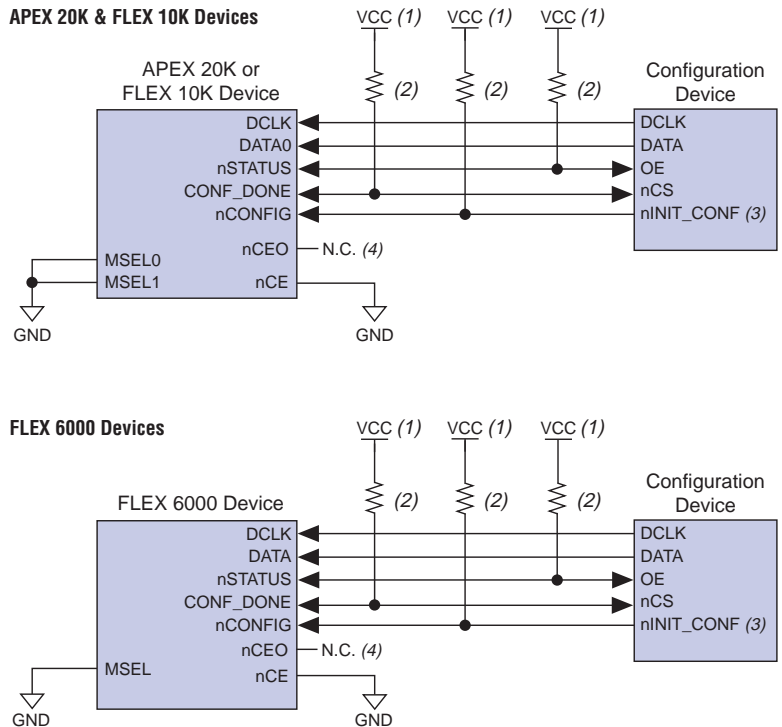
This section describes how to configure APEX 20K, FLEX 10K, and FLEX 6000 devices with the following configuration schemes:

- Configuration Device
- PS Configuration with a Download Cable
- PS Configuration with a Microprocessor
- PPS Configuration (APEX 20K and FLEX 10K Devices Only)
- PSA Configuration (FLEX 6000 Devices Only)
- PPA Configuration (APEX 20K and FLEX 10K Devices Only)
- JTAG Programming and Configuration (APEX 20K and FLEX 10K Devices Only)
- JTAG Programming and Configuration of Multiple Devices (APEX 20K and FLEX 10K Devices Only)

### Configuration Device

The configuration device scheme uses an Altera-supplied serial configuration device to supply data to the APEX 20K, FLEX 10K, or FLEX 6000 device in a serial bitstream. See Figure 2.

Figure 2. Configuration Device Scheme Circuit

**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the configuration device.
- (2) All pull-up resistors are 1 kΩ. The EPC2 device's OE, nCS, and nINIT\_CONF pins have internal, user-configurable 1-kΩ pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins.
- (3) The nINIT\_CONF pin is available on EPC2 devices only. If nINIT\_CONF is not available (i.e., on EPC1 devices) or not used, nCONFIG must be pulled to V<sub>CC</sub> either directly or through a 1-kΩ resistor.
- (4) The nCEO pin is left unconnected.

In the configuration device scheme, nCONFIG is usually tied to V<sub>CC</sub> (in EPC2 devices, nCONFIG is connected to nINIT\_CONF). Upon device power-up, the target APEX 20K, FLEX 10K, or FLEX 6000 device senses the low-to-high transition on nCONFIG and initiates configuration. The target device then drives the open-drain CONF\_DONE pin low, which in-turn drives the configuration device's nCS pin low. When exiting power-on reset (POR), both the target and configuration device release the open-drain nSTATUS pin.

Before configuration begins, the configuration device issues a POR delay of 200 ms (maximum) to allow the power supply to stabilize; during this time, the configuration device drives its OE pin low. This low signal delays configuration because the OE pin is connected to the target device's nSTATUS pin. When both devices complete POR, they release nSTATUS, which is then pulled high by a pull-up resistor. When configuring multiple devices, configuration does not begin until all devices release their OE or nSTATUS pins. When all devices are ready, the configuration device clocks data out serially to the target devices using an internal oscillator.

After successful configuration, the configuration device starts clocking the target device for initialization. The CONF\_DONE pin is released by the target device and then pulled high by a pull-up resistor. When initialization is complete, the configuration device enters user mode.

If an error occurs during configuration, the target device drives its nSTATUS pin low, resetting itself internally and resetting the configuration device. If the *Auto-Restart Configuration on Frame Error* option—available in the MAX+PLUS II **Global Project Device Options** dialog box (Assign menu)—is turned on, the device reconfigures automatically if an error occurs. The Quartus software provides a similar option for APEX 20K devices.

If this option is turned off, the external system must monitor nSTATUS for errors and then pulse nCONFIG low to restart configuration. The external system can pulse nCONFIG if nCONFIG is under system control rather than tied to V<sub>CC</sub>. When configuration is complete, the target device releases CONF\_DONE, which disables the configuration device by driving nCS high. The configuration device drives DCLK low before and after configuration.

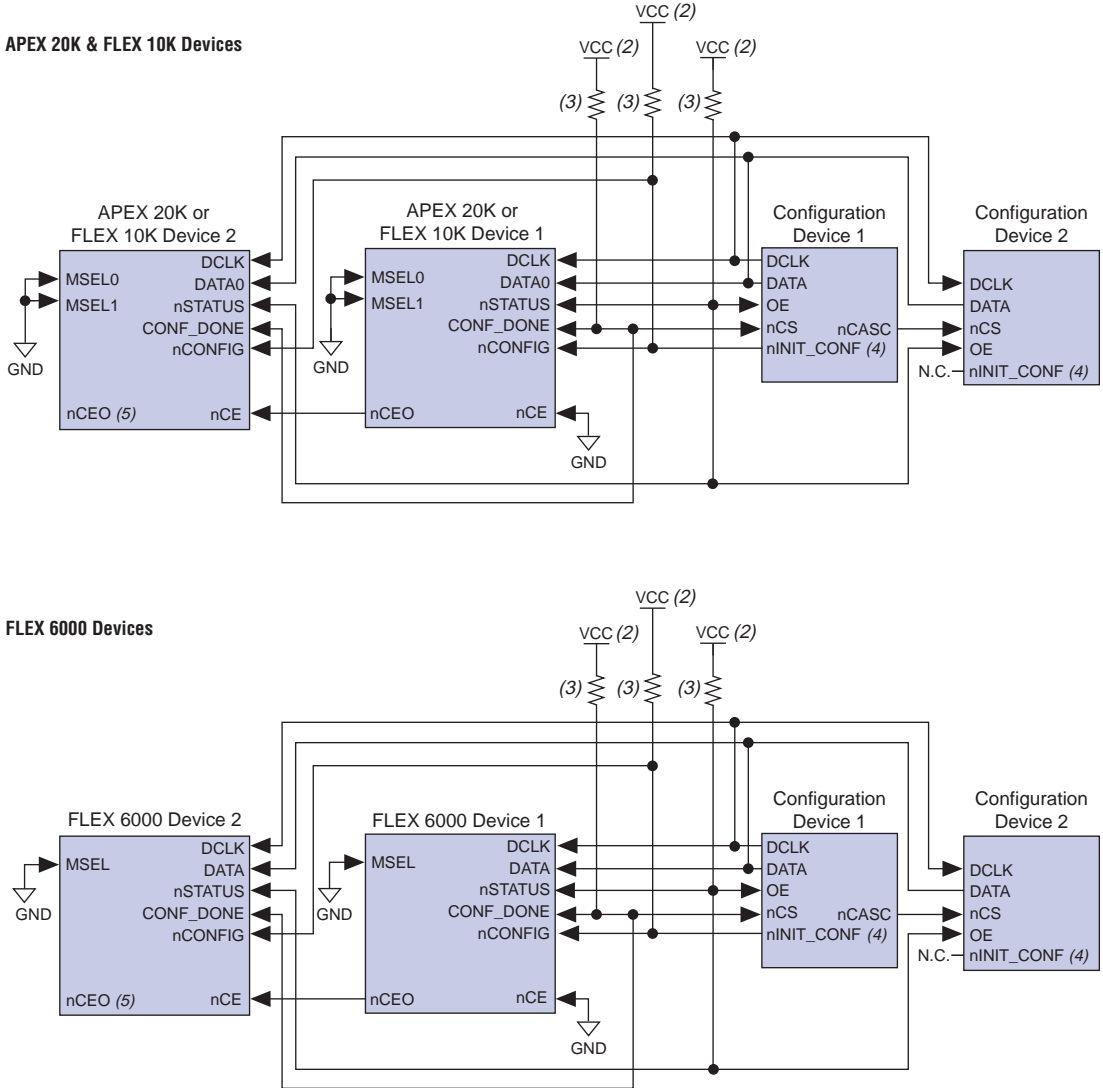
In addition, if the configuration device sends all of its data and then detects that CONF\_DONE has not gone high, it recognizes that the target device has not configured successfully. In this case, the configuration device pulses its OE pin low for a few microseconds, driving the target device's nSTATUS pin low. If the *Auto-Restart Configuration on Frame Error* option is set in the software, the target device resets and then pulses its nSTATUS pin low. When nSTATUS returns high, the configuration device reconfigures the target device. When configuration is complete, the configuration device drives DCLK low.



When CONF\_DONE is driven low after device configuration, the configuration device recognizes that the target device has not configured successfully; therefore, you should not use this scheme to delay initialization. Instead, you should use the MAX+PLUS II or Quartus software's *User-Supplied Start-Up Clock* option to synchronize the initialization of multiple devices that are not in the same configuration chain. Devices in the same configuration chain will initialize together. For more information on this option, see ["Device Options" on page 55](#).

[Figure 3](#) shows how to configure multiple devices with a configuration device. This circuit is similar to the configuration device circuit for a single device, except the APEX 20K, FLEX 10K, or FLEX 6000 devices are cascaded for multi-device configuration.

**Figure 3. Multi-Device Configuration Circuit** *Note (1)*



**Notes:**

- (1) When performing multi-device active serial configuration, you must generate the EPC1 or EPC2 device's Programmer Object File (.pof) from each project's SRAM Object File (.sof). You can combine multiple SOFs using the MAX+PLUS II software's **Combine Programming Files** dialog box (File menu). For APEX 20K devices, the Quartus software provides a similar option. For more information on how to create configuration and programming files, see ["Device Configuration Files"](#) on page 64.
- (2) The pull-up resistor should be connected to the same supply voltage as the configuration device.
- (3) All pull-up resistors are 1 kΩ. The EPC2 device's OE, nCS, and nINIT\_CONF pins have internal, user-configurable 1-kΩ pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins.
- (4) The nINIT\_CONF pin is available on EPC2 devices only. If nINIT\_CONF is not available (i.e., on EPC1 devices) or not used, nCONFIG must be pulled to V<sub>CC</sub> either directly or through a 1-kΩ resistor.
- (5) The nCEO pin is left unconnected for the last device in the chain.

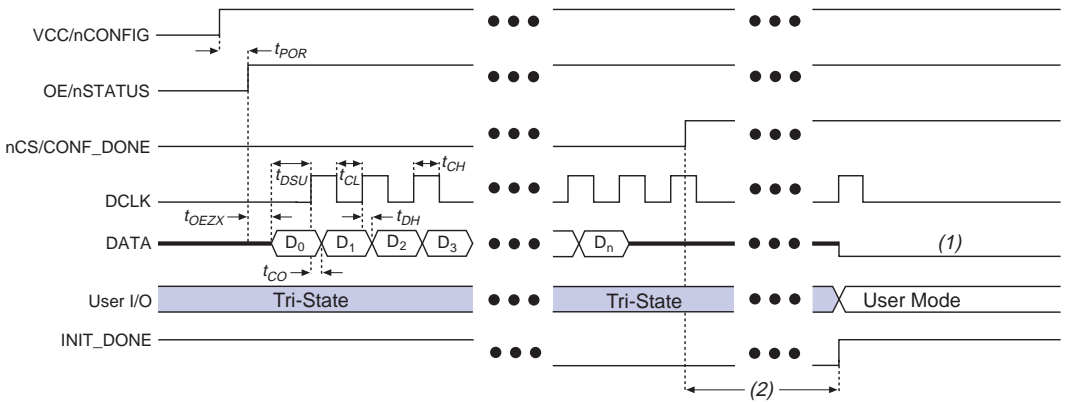
After the first device completes configuration during multi-device configuration, its `nCEO` pin activates the second device's `nCE` pin, prompting the second device to begin configuration. Because all device `CONF_DONE` pins are tied together, all devices initialize and enter user mode at the same time.

In addition, all `nSTATUS` pins are tied together; thus, if any device (including the configuration devices) detects an error, configuration stops for the entire chain. Also, if the first configuration device does not detect `CONF_DONE` going high at the end of configuration, it resets the chain by pulsing its `OE` pin low for a few microseconds. This low pulse drives the `OE` pin low on the second configuration device and drives `nSTATUS` low on all APEX 20K, FLEX 10K, or FLEX 6000 devices, causing them to enter an error state; this state is similar to an APEX 20K, FLEX 10K, or FLEX 6000 device detecting an error.

If the *Auto-Restart Configuration on Frame Error* option is set in the software, the APEX 20K, FLEX 10K, or FLEX 6000 devices release their `nSTATUS` pins after a reset time-out period. When the `nSTATUS` pins are released and pulled high, the configuration devices reconfigure the chain. If the *Auto-Restart Configuration on Frame Error* option is not set, the APEX 20K, FLEX 10K, or FLEX 6000 devices drive `nSTATUS` low until they are reset with a low pulse on `nCONFIG`.

You can also cascade several configuration devices to configure multiple APEX 20K, FLEX 10K, or FLEX 6000 devices. When all data from the first configuration device is sent, it drives `nCASC` low, which in turn drives `nCS` on the subsequent configuration device. Because a configuration device requires less than one clock cycle to activate a subsequent configuration device, the data stream is uninterrupted. [Figure 4](#) shows the timing waveform for the configuration device scheme.

Figure 4. Configuration Device Scheme Timing Waveform



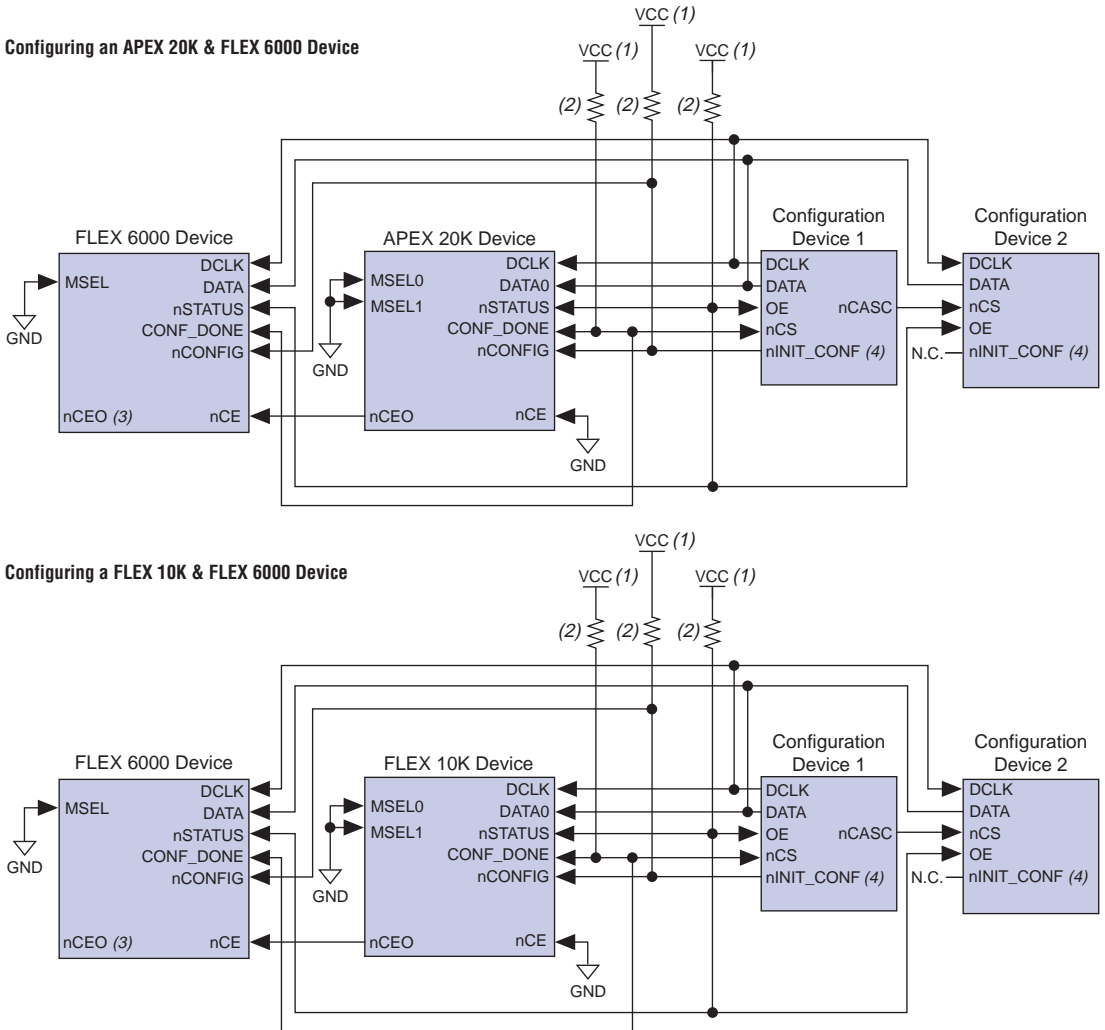
**Notes:**

- (1) DATA should not be left floating. It should be driven high or low, whichever is more convenient.
- (2) For APEX 20K devices, the device enters user mode 40 clock cycles after CONF\_DONE goes high. For FLEX 10K and FLEX 6000 devices, the device enters user mode 10 clock cycles after CONF\_DONE goes high.

You can use a single configuration chain to configure multiple APEX 20K, FLEX 10K, and FLEX 6000 devices. In this scheme, the nCEO pin of the first device is connected to the nCE pin of the second device in the chain. To configure properly, all of the device CONF\_DONE and nSTATUS pins must be tied together.

Figure 5 shows examples of configuring multiple APEX 20K, FLEX 10K, and FLEX 6000 devices using a configuration device.

Figure 5. Multi-Device Configuration with APEX 20K, FLEX 10K & FLEX 6000 Devices



**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the configuration device.
- (2) All pull-up resistors are 1 kΩ. The EPC2 device's OE, nCS, and nINIT\_CONF pins have internal, user-configurable 1-kΩ pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins.
- (3) The nCEO pin is left unconnected for the last device in the chain.
- (4) The nINIT\_CONF pin is available on EPC2 devices only. If nINIT\_CONF is not available (i.e., on EPC1 devices) or not used, nCONFIG must be pulled to V<sub>CC</sub> either directly or through a 1-kΩ resistor.

Table 6 defines the APEX 20K, FLEX 10K, and FLEX 6000 timing parameters when using EPC2 devices at 3.3 V.

Symbol	Parameter	Min	Max	Units
$t_{POR}$	POR delay (2)		200	ms
$t_{OEZX}$	OE high to DATA output enabled		80	ns
$t_{CH}$	DCLK high time	40	100	ns
$t_{CL}$	DCLK low time	40	100	ns
$t_{DSU}$	Data setup time before rising edge on DCLK	30		ns
$t_{DH}$	Data hold time after rising edge on DCLK	0		ns
$t_{CO}$	DCLK to DATA out		30	ns
$t_{OEW}$	OE low pulse width to guarantee counter reset	100		ns
$f_{CLK}$	DCLK frequency	5	12.5	MHz

**Notes:**

- (1) For more information regarding configuration device timing parameters, see the [Configuration Devices for APEX & FLEX Devices Data Sheet](#).
- (2) The configuration device imposes a POR delay upon initial power-up to allow the voltage supply to stabilize. Subsequent reconfigurations do not incur this delay.

Table 7 defines the APEX 20K, FLEX 10K, and FLEX 6000 timing parameters when using EPC1 and EPC1441 devices at 3.3 V.

**Table 7. APEX 20K, FLEX 10K & FLEX 6000 Timing Parameters using EPC1 & EPC1441 Devices at 3.3 V** *Note (1)*

Symbol	Parameter	Min	Max	Units
$t_{POR}$	POR delay (2)		200	ms
$t_{OEZX}$	OE high to DATA output enabled		80	ns
$t_{CH}$	DCLK high time	50	250	ns
$t_{CL}$	DCLK low time	50	250	ns
$t_{DSU}$	Data setup time before rising edge on DCLK	30		ns
$t_{DH}$	Data hold time after rising edge on DCLK	0		ns
$t_{CO}$	DCLK to DATA out		30	ns
$t_{OEW}$	OE low pulse width to guarantee counter reset	100		ns
$f_{CLK}$	DCLK frequency	2	10	MHz

**Notes:**

- (1) For more information regarding configuration device timing parameters, see the [Configuration Devices for APEX & FLEX Devices Data Sheet](#).
- (2) The configuration device imposes a POR delay upon initial power-up to allow the voltage supply to stabilize. Subsequent reconfigurations do not incur this delay.

Table 8 defines the APEX 20K, FLEX 10K, and FLEX 6000 timing parameters when using EPC2, EPC1, and EPC1441 devices at 5.0 V.

**Table 8. APEX 20K, FLEX 10K & FLEX 6000 Timing Parameters using EPC2, EPC1 & EPC1441 Devices at 5.0 V** *Note (1)*

Symbol	Parameter	Min	Max	Units
$t_{POR}$	POR delay (2)		200	ms
$t_{OEZX}$	OE high to DATA output enabled		50	ns
$t_{CH}$	DCLK high time	30	75	ns
$t_{CL}$	DCLK low time	30	75	ns
$t_{DSU}$	Data setup time before rising edge on DCLK	30		ns
$t_{DH}$	Data hold time after rising edge on DCLK	0		ns
$t_{CO}$	DCLK to DATA out		30	ns
$t_{OEW}$	OE low pulse width to guarantee counter reset	100		ns
$f_{CLK}$	DCLK frequency	6.7	16.7	MHz

**Notes:**

- (1) For more information regarding configuration device timing parameters, see the [Configuration Devices for APEX & FLEX Devices Data Sheet](#).
- (2) The configuration device imposes a POR delay upon initial power-up to allow the voltage supply to stabilize. Subsequent reconfigurations do not incur this delay.



The I/O pins are tri-stated after power up and during configuration. In user mode, these pins perform their programmed function.

Table 9 shows the status of the device DATA pins before and after configuration. (APEX 20K and FLEX 10K devices have a DATA[7..0] bus, while FLEX 6000 devices have a DATA pin only.)

Pins	APEX 20K or FLEX 10K Device		FLEX 6000 Device	
	Before	After	Before	After
DATA (1)	–	–	Tri-state	Tri-state
DATA0 (1)	Tri-state	Tri-state	–	–
DATA[7..1] (2)	User defined	User defined	–	–

**Notes:**

- (1) The status shown is for configuration with a configuration device.
- (2) The function of these pins depends upon the settings specified in the MAX+PLUS II software's **Global Project Device Options** dialog box. For APEX 20K devices, the Quartus software provides a similar option. For more information, refer to MAX+PLUS II or Quartus Help.



For information on how to create configuration and programming files for this configuration scheme, see “[Device Configuration Files](#)” on page 64.

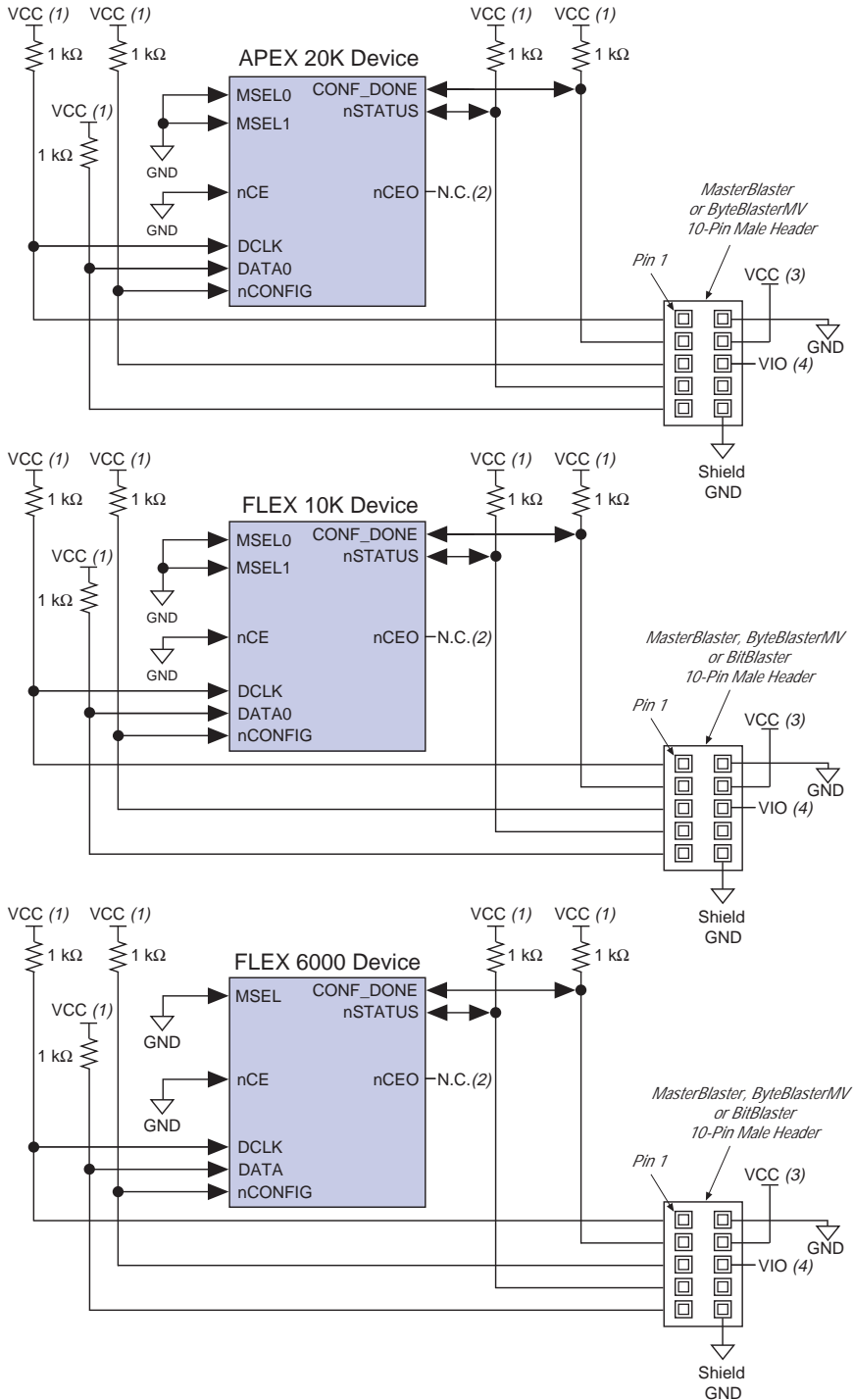
## PS Configuration with a Download Cable

In PS configuration with a download cable, an intelligent host transfers data from a storage device to the APEX 20K, FLEX 10K, or FLEX 6000 device via the MasterBlaster, ByteBlasterMV, or BitBlaster cable. To initiate configuration in this scheme, the download cable generates a low-to-high transition on the nCONFIG pin. The programming hardware then places the configuration data one bit at a time on the device's DATA pin (the DATA0 pin in APEX 20K and FLEX 10K devices, and the DATA pin in FLEX 6000 devices). The data is clocked into the target device until CONF\_DONE goes high.

When using programming hardware for APEX 20K, FLEX 10K, or FLEX 6000 devices, setting the *Auto-Restart Configuration on Frame Error* option does not affect the configuration cycle because the MAX+PLUS II or Quartus software must restart configuration when an error occurs. Figure 6 shows PS configuration for APEX 20K, FLEX 10K, and FLEX 6000 devices using a MasterBlaster, ByteBlasterMV, or BitBlaster cable.



Figure 6. PS Configuration Circuit with MasterBlaster, ByteBlasterMV, or BitBlaster Cable



### Notes to figure:

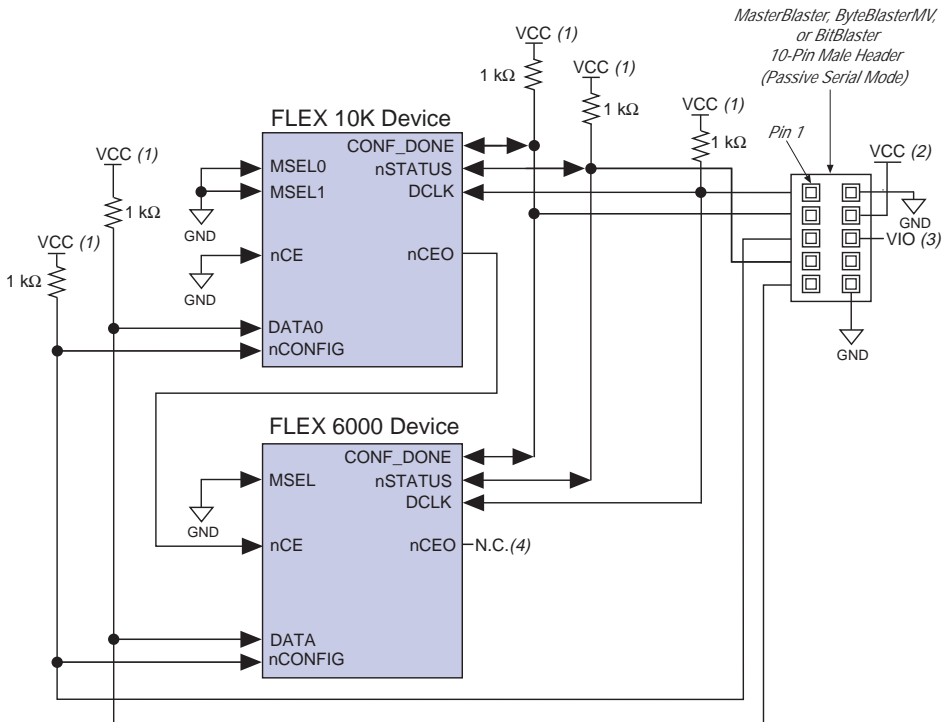
- (1) The pull-up resistor should be connected to the same supply voltage as the MasterBlaster, ByteBlasterMV, or BitBlaster cable.
- (2) The nCEO pin is left unconnected for the last device in the chain.
- (3) Power supply voltage:  $V_{CC} = 2.5\text{ V}$ ,  $3.3\text{ V}$ , or  $5.0\text{ V}$  for the MasterBlaster cable.  
 $V_{CC} = 3.3\text{ V}$  or  $5.0\text{ V}$  for the ByteBlasterMV cable.  
 $V_{CC} = 5.0\text{ V}$  for the BitBlaster cable.
- (4) Pin 6 of the header is a VIO reference voltage for the MasterBlaster output driver. VIO should match the device's  $V_{CCIO}$ . This pin is a no connect pin for the ByteBlasterMV and BitBlaster header.

You can use programming hardware to configure multiple APEX 20K, FLEX 10K, or FLEX 6000 devices by connecting each device's nCEO pin to the subsequent device's nCE pin. All other configuration pins are connected to each device in the chain. Because all CONF\_DONE pins are tied together, all devices in the chain initialize and enter user mode at the same time.

In addition, because the nSTATUS pins are tied together, the entire chain halts configuration if any device detects an error. In this situation, the MAX+PLUS II or Quartus software must restart configuration; the *Auto-Restart Configuration on Frame Error* option does not affect the configuration cycle.

Figure 7 shows the schematic for configuring multiple FLEX 10K and FLEX 6000 devices with the MasterBlaster, ByteBlasterMV, or BitBlaster download cable.

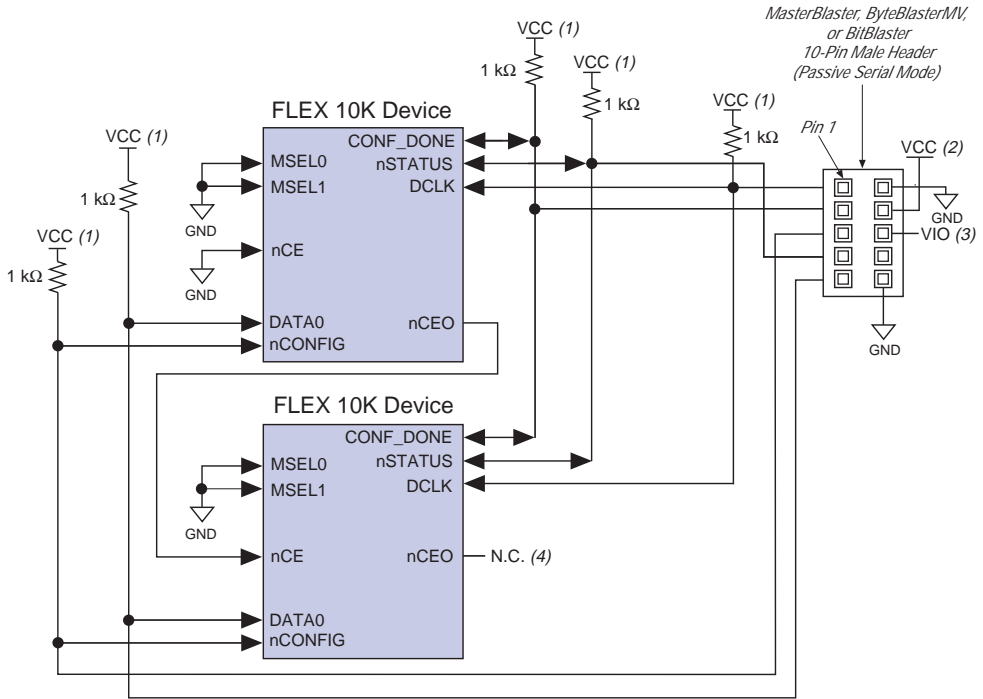
Figure 7. PS Multi-Device Configuration for FLEX 10K &amp; FLEX 6000 Devices with a Cable

**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the MasterBlaster, ByteBlasterMV, or BitBlaster cable.
- (2) Power supply voltage:  $V_{CC} = 2.5\text{ V}, 3.3\text{ V}, \text{ or } 5.0\text{ V}$  for the MasterBlaster cable.  
 $V_{CC} = 3.3\text{ V}$  or  $5.0\text{ V}$  for the ByteBlasterMV cable.  
 $V_{CC} = 5.0\text{ V}$  for the BitBlaster cable.
- (3) VIO is a reference voltage for the MasterBlaster output driver. VIO should match the device's  $V_{CCI0}$ . Refer to the [MasterBlaster Serial/USB Communications Cable Data Sheet](#) for this value.
- (4) The nCEO pin is left unconnected for the last device in the chain.

Figure 8 shows the schematic for configuring multiple FLEX 10K devices with the MasterBlaster, ByteBlasterMV, or BitBlaster download cable.

Figure 8. PS Multi-Device Configuration for Multiple FLEX 10K Devices with a Cable

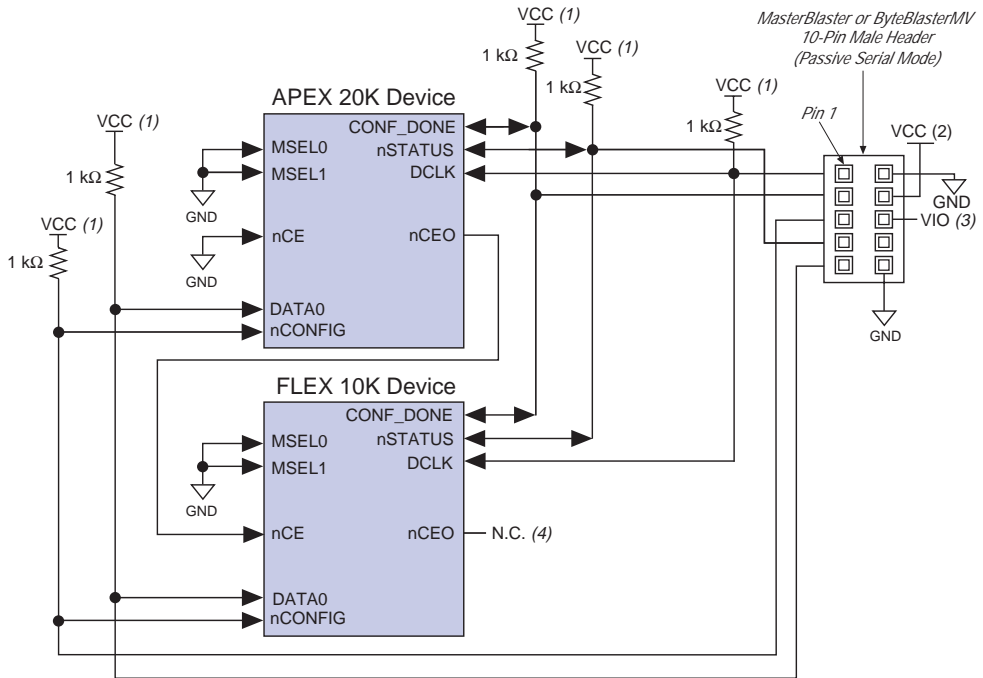


**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the MasterBlaster, ByteBlasterMV, or BitBlaster cable.
- (2) Power supply voltage:  $V_{CC} = 2.5\text{ V}, 3.3\text{ V}, \text{ or } 5.0\text{ V}$  for the MasterBlaster cable.  
 $V_{CC} = 3.3\text{ V}$  or  $5.0\text{ V}$  for the ByteBlasterMV cable.  
 $V_{CC} = 5.0\text{ V}$  for the BitBlaster cable.
- (3) VIO is a reference voltage for the MasterBlaster output driver. VIO should match the device's  $V_{CCIO}$ . Refer to the *MasterBlaster Serial/USB Communications Cable Data Sheet* for this value.
- (4) The nCEO pin is left unconnected for the last device in the chain.

Figure 9 shows the schematic for configuring multiple APEX 20K and FLEX 10K devices with the MasterBlaster, ByteBlasterMV, or BitBlaster cable.

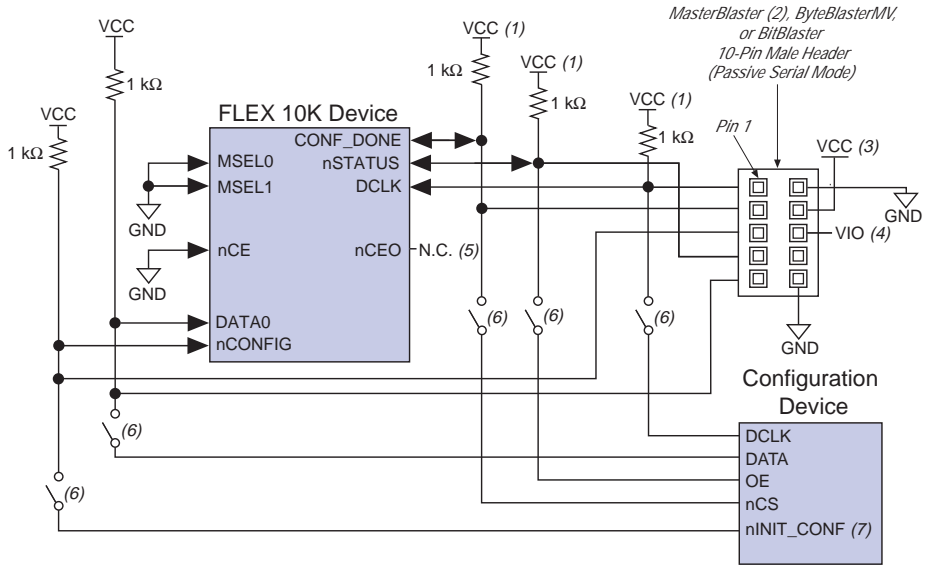
Figure 9. PS Multi-Device Configuration for APEX 20K &amp; FLEX 10K Devices with a Cable

**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the MasterBlaster, ByteBlasterMV, or BitBlaster cable.
- (2) Power supply voltage:  $V_{CC} = 2.5\text{ V}$ ,  $3.3\text{ V}$ , or  $5.0\text{ V}$  for the MasterBlaster cable.  
 $V_{CC} = 3.3\text{ V}$  or  $5.0\text{ V}$  for the ByteBlasterMV cable.  
 $V_{CC} = 5.0\text{ V}$  for the BitBlaster cable.
- (3) VIO is a reference voltage for the MasterBlaster output driver. VIO should match the device's  $V_{CCIO}$ . Refer to the [MasterBlaster Serial/USB Communications Cable Data Sheet](#) for this value.
- (4) The nCEO pin is left unconnected for the last device in the chain.

If you are using a MasterBlaster, ByteBlasterMV, or BitBlaster cable to configure device(s) on a board that also has configuration device(s) and cable, you should electrically isolate the configuration device from the target device(s) and cable. One way to isolate the configuration device is to add logic, such as a multiplexer, that can select between the configuration device and the cable. Another option is to add switches to the five common signals (i.e., CONF\_DONE, nSTATUS, DCLK, nCONFIG, and DATA0) between the cable and the configuration device. The last option is to remove the configuration device from the board when configuring with the cable. Figure 10 shows a combination of a configuration device and a MasterBlaster, ByteBlasterMV, or BitBlaster cable to configure a FLEX 10K device.

Figure 10. Configuring with a Combined PS & Configuration Device Scheme



**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the configuration device.
- (2) Refer to the [MasterBlaster Serial/USB Communications Cable Data Sheet](#) for more information on the MasterBlaster cable.
- (3) Power supply voltage:  $V_{CC} = 2.5\text{ V}, 3.3\text{ V},$  or  $5.0\text{ V}$  for the MasterBlaster cable.  
 $V_{CC} = 3.3\text{ V}$  or  $5.0\text{ V}$  for the ByteBlasterMV cable.  
 $V_{CC} = 5.0\text{ V}$  for the BitBlaster cable.
- (4) Pin 6 of the header is a VIO reference voltage for the MasterBlaster output driver. VIO should match the target device's  $V_{CCIO}$ . This pin is a no connect pin for the ByteBlasterMV and BitBlaster header.
- (5) The nCEO pin is left unconnected.
- (6) You should not attempt configuration with a MasterBlaster, ByteBlasterMV, or BitBlaster cable while a configuration device is connected to an APEX 20K, FLEX 10K, or FLEX 6000 device. Instead, you should either remove the configuration device from its socket when using the download cable or place a switch on the five common signals between the download cable and the configuration device.
- (7) The nINIT\_CONF pin is available on EPC2 devices only. If nINIT\_CONF is not available (i.e., on EPC1 devices) or not used, nCONFIG must be pulled to  $V_{CC}$  either directly or through a 1-k $\Omega$  resistor.



For more information on how to use the MasterBlaster, ByteBlasterMV, or BitBlaster cables, see the following documents:

- [MasterBlaster Serial/USB Communications Cable Data Sheet](#)
- [ByteBlasterMV Parallel Port Download Cable Data Sheet](#)
- [BitBlaster Serial Download Cable Data Sheet](#)



For information on how to create configuration and programming files for this configuration scheme, see “[Device Configuration Files](#)” on page 64.

## PS Configuration with a Microprocessor

In PS configuration with a microprocessor, a microprocessor transfers data from a storage device to the target APEX 20K, FLEX 10K, or FLEX 6000 device via programming hardware. To initiate configuration in this scheme, the microprocessor must generate a low-to-high transition on the `nCONFIG` pin and the target device must release `nSTATUS`. The microprocessor or programming hardware then places the configuration data one bit at a time on the `DATA` pin of the target device (the `DATA0` pin for APEX 20K and FLEX 10K devices, and the `DATA` pin for FLEX 6000 devices). The least significant bit (LSB) of each data byte must be presented first. Data is clocked continuously into the target device until `CONF_DONE` goes high.

After all data is transferred, `DCLK` must be clocked an additional 10 times for FLEX 10K and FLEX 6000 devices or an additional 40 times for APEX 20K devices to initialize the device. The device's `CONF_DONE` pin goes high to show successful configuration and to start initialization. The configuration files created by the MAX+PLUS II or Quartus software incorporate extra bits for initialization. Driving `DCLK` to the device after configuration does not affect device operation. Therefore, sending the entire configuration file to the device is sufficient to configure and initialize it.

Handshaking signals are not used in PS configuration modes. Therefore, the configuration clock speed must be below the specified frequency to ensure correct configuration. No maximum `DCLK` period exists. You can pause configuration by halting `DCLK`.

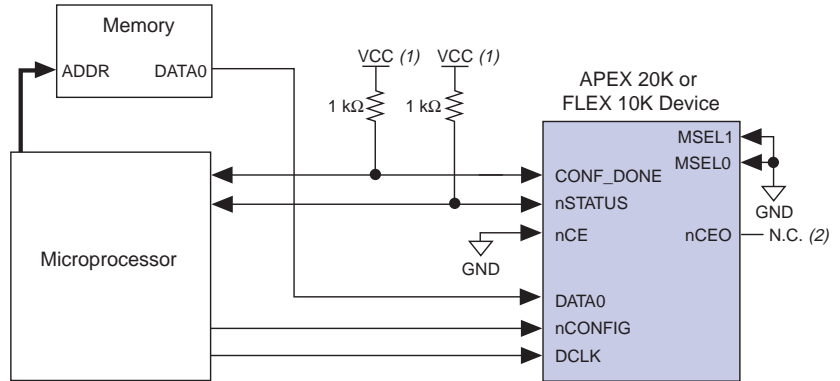
If the target device detects an error during configuration, it drives its `nSTATUS` pin low to alert the microprocessor. The microprocessor can then pulse `nCONFIG` low to restart the configuration process. Alternatively, if the *Auto-Restart Configuration on Frame Error* option is set in the MAX+PLUS II or Quartus software, the target device releases `nSTATUS` after a reset time-out period. After `nSTATUS` is released, the microprocessor can reconfigure the target device without needing to pulse `nCONFIG` low.

The microprocessor can also monitor the `CONF_DONE` and `INIT_DONE` pins to ensure successful configuration. If the microprocessor sends all data and the initialization clock starts but `CONF_DONE` has not gone high, it must reconfigure the target device.

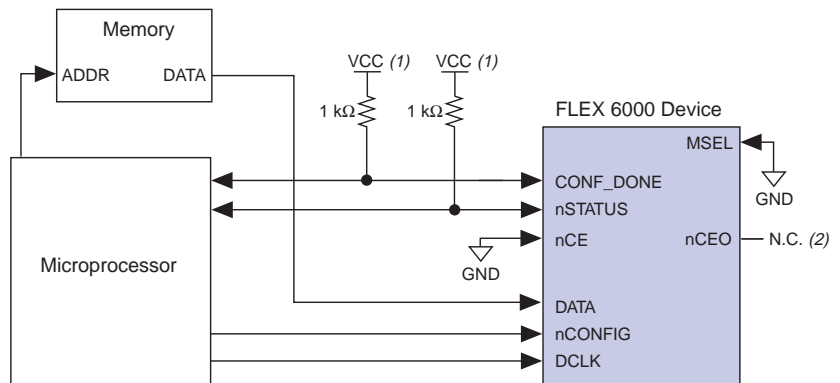
Figure 11 shows the circuit for PS configuration with a microprocessor.

**Figure 11. PS Configuration Circuit with Microprocessor**

**APEX 20K or FLEX 10K Devices**



**FLEX 6000 Devices**



**Notes:**

- (1) The pull-up resistor should be connected to any  $V_{CC}$  that meets the device high-level input voltage ( $V_{IH}$ ) specification.
- (2) The nCEO pin is left unconnected.

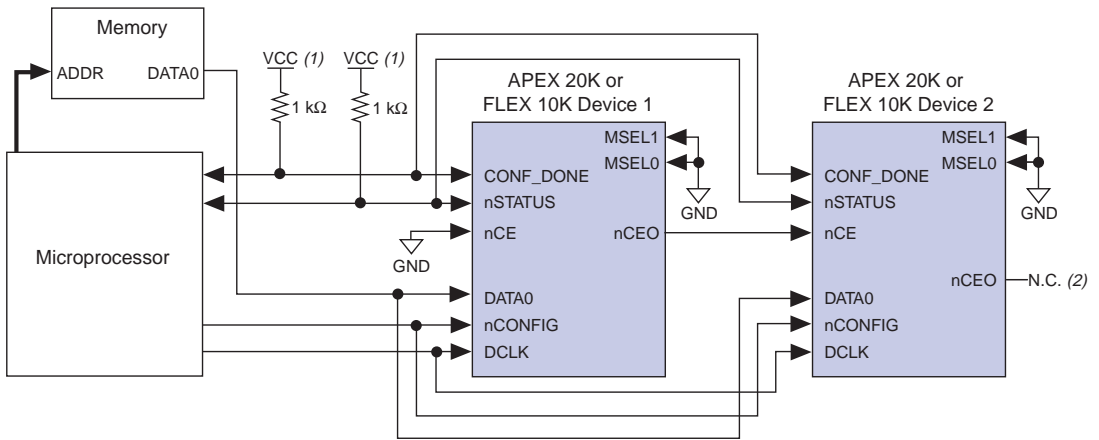
For multi-device PS configuration with a microprocessor, the first device's nCEO pin is cascaded to the second device's nCE pin. The second device in the chain begins configuration within one clock cycle; therefore, the transfer of data destinations is transparent to the microprocessor. Because all device CONF\_DONE pins are tied together, all devices initialize and enter user mode at the same time.



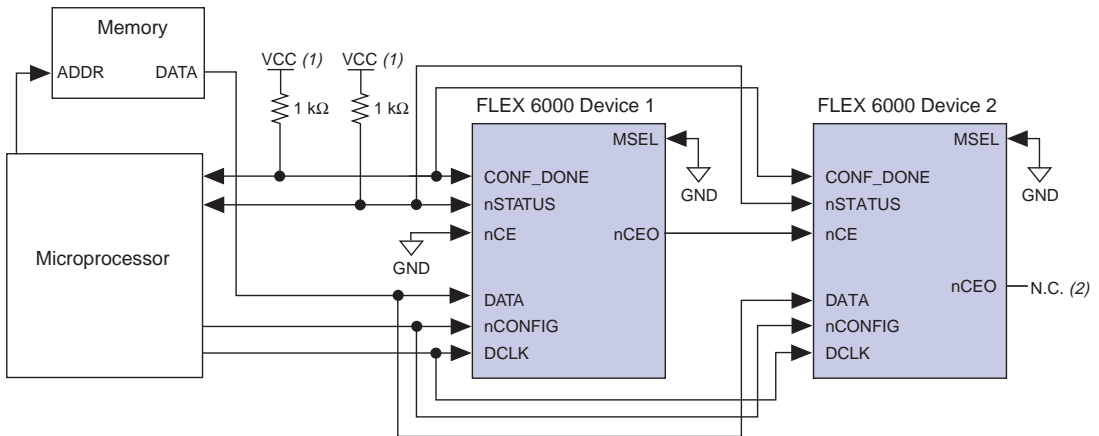
In addition, the nSTATUS pins are tied together; thus, if any device detects an error, the entire chain halts configuration and drives nSTATUS low. The microprocessor can then pulse nCONFIG low to restart the configuration process. Alternatively, if the *Auto-Restart Configuration on Frame Error* option is set in the MAX+PLUS II or Quartus software, the target devices release nSTATUS after a reset time-out period. After nSTATUS is released, the microprocessor can reconfigure the target devices. Figure 12 shows multi-device configuration with a microprocessor.

Figure 12. PS Multi-Device Configuration with a Microprocessor

APEX 20K or FLEX 10K Devices



FLEX 6000 Devices

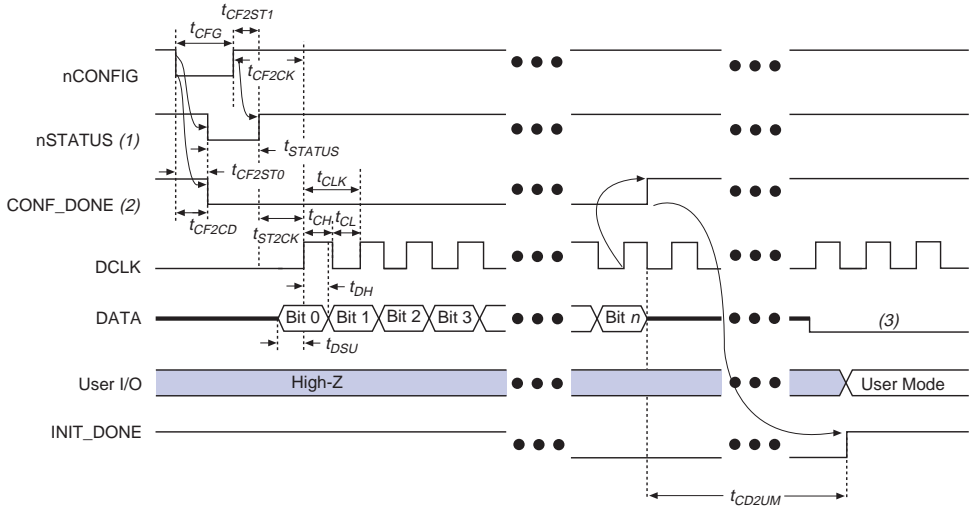


Notes:

- (1) The pull-up resistor should be connected to a supply that provides an acceptable power level for all devices in the chain. For example, when a device chain contains a mixture of 5.0-V FLEX 10K devices and 2.5-V FLEX 10KE devices, the pull-up resistor should be connected to 5.0 V. You should use 5.0 V in this scenario because FLEX 10KE I/O pins are 5.0-V tolerant.
- (2) The nCEO pin is left unconnected for the last device in the chain.

Figure 13 shows the PS configuration timing waveform for APEX 20K, FLEX 10K, and FLEX 6000 devices.

Figure 13. PS Timing Waveform for APEX 20K, FLEX 10K & FLEX 6000 Devices



**Notes:**

- (1) Upon power-up, nSTATUS is held low not more than 5  $\mu$ s when  $V_{CC}$  reaches its minimum requirement.
- (2) Upon power-up and before configuration, CONF\_DONE is low.
- (3) DATA should not be left floating. It should be driven high or low, whichever is more convenient.

Tables 10 and 11 contain preliminary timing information for APEX 20K, FLEX 10K, and FLEX 6000 devices.

**Table 10. PS Timing Parameters for APEX 20K Devices**

Symbol	Parameter	Min	Max	Units
t <sub>CF2CD</sub>	nCONFIG low to CONF_DONE low		200	ns
t <sub>CF2ST0</sub>	nCONFIG low to nSTATUS low		200	ns
t <sub>CF2ST1</sub>	nCONFIG to nSTATUS high		4	µs
t <sub>CFG</sub>	nCONFIG low pulse width (1)	8		µs
t <sub>STATUS</sub>	nSTATUS low pulse width	10	40	µs
t <sub>CF2CK</sub>	nCONFIG high to first rising edge on DCLK	40		µs
t <sub>ST2CK</sub>	nSTATUS high to first rising edge on DCLK	1		µs
t <sub>DSU</sub>	Data setup time before rising edge on DCLK	10		ns
t <sub>DH</sub>	Data hold time after rising edge on DCLK	0		ns
t <sub>CH</sub>	DCLK high time	30		ns
t <sub>CL</sub>	DCLK low time	30		ns
t <sub>CLK</sub>	DCLK period	60		ns
f <sub>MAX</sub>	DCLK maximum frequency		16.7	MHz
t <sub>CD2UM</sub>	CONF_DONE high to user mode (2)	2	8	µs

**Table 11. PS Timing Parameters for FLEX 10K & FLEX 6000 Devices**

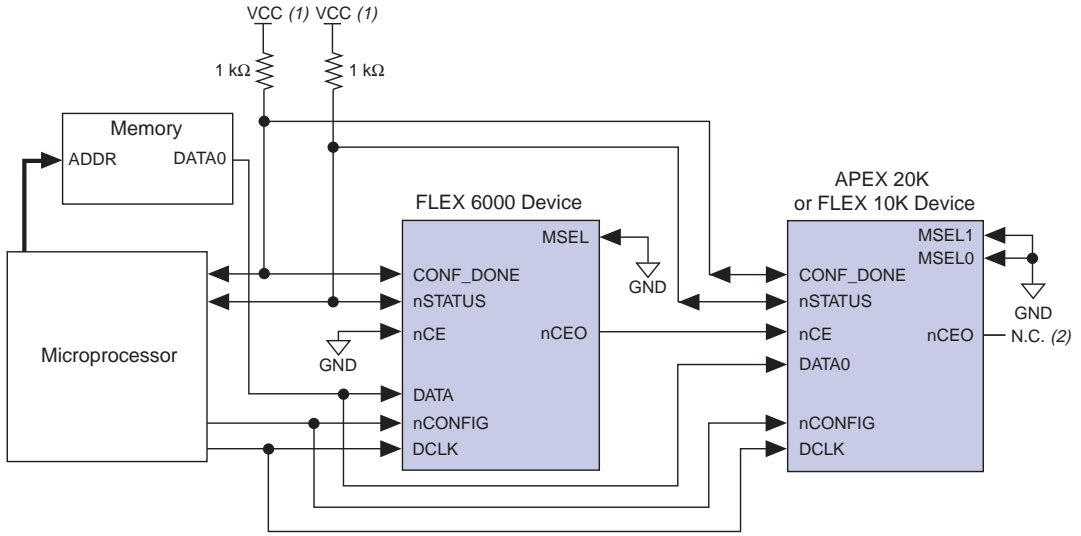
Symbol	Parameter	Min	Max	Units
t <sub>CF2CD</sub>	nCONFIG low to CONF_DONE low		200	ns
t <sub>CF2ST0</sub>	nCONFIG low to nSTATUS low		200	ns
t <sub>CF2ST1</sub>	nCONFIG high to nSTATUS high		4	µs
t <sub>CFG</sub>	nCONFIG low pulse width (1)	2		µs
t <sub>STATUS</sub>	nSTATUS low pulse width	1		µs
t <sub>CF2CK</sub>	nCONFIG high to first rising edge on DCLK	5		µs
t <sub>ST2CK</sub>	nSTATUS high to first rising edge on DCLK	1		µs
t <sub>DSU</sub>	Data setup time before rising edge on DCLK	10		ns
t <sub>DH</sub>	Data hold time after rising edge on DCLK	0		ns
t <sub>CH</sub>	DCLK high time	30		ns
t <sub>CL</sub>	DCLK low time	30		ns
t <sub>CLK</sub>	DCLK period	60		ns
f <sub>MAX</sub>	DCLK maximum frequency		16.7	MHz
t <sub>CD2UM</sub>	CONF_DONE high to user mode (2)	0.6	2	µs

**Notes to tables:**

- (1) If configuration is stopped and reinitiated before CONF\_DONE goes high, this value applies when the internal oscillator is selected as the clock source. If configuration is stopped and reinitiated before CONF\_DONE goes high and the clock source is CLKUSR or DCLK, multiply the clock period by 40 for APEX devices or 10 for FLEX 10K and FLEX 6000 devices.
- (2) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 40 for APEX 20K devices or 10 for FLEX 10K and FLEX 6000 devices to obtain this value.

Figure 14 shows the PS multi-device configuration circuit with APEX 20K, FLEX 10K, and FLEX 6000 devices using a microprocessor.

**Figure 14. PS Multi-Device Configuration of APEX 20K, FLEX 10K & FLEX 6000 Devices with a Microprocessor**



**Notes:**

- (1) The pull-up resistor should be connected to a supply that provides an acceptable power level for all devices in the chain. For example, when a device chain contains a mixture of 5.0-V FLEX 10K devices and 2.5-V FLEX 10KE devices, the pull-up resistor should be connected to 5.0 V. You should use 5.0 V in this scenario because FLEX 10KE I/O pins are 5.0-V tolerant.
- (2) The nCEO pin is left unconnected for the last device in the chain.



For information on how to create configuration and programming files for this configuration scheme, see “[Device Configuration Files](#)” on page 64.

## PPS Configuration (APEX 20K & FLEX 10K Devices Only)

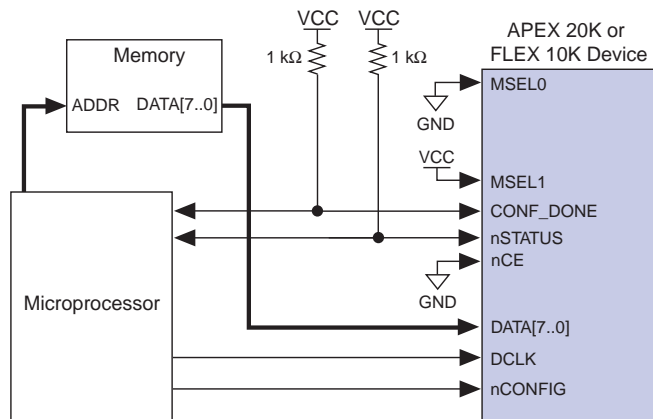
In a passive parallel synchronous (PPS) configuration scheme, an intelligent host drives the target APEX 20K or FLEX 10K device. The host system outputs parallel data and the serializing clock to the device. The target device latches the byte-wide data on the DATA[7..0] pins, and serializes it internally.

The DCLK, CONF\_DONE, nCONFIG, nSTATUS, and DATA[7..0] pins are connected to a port on the intelligent host, such as a microprocessor. To begin configuration, nCONFIG is given a low-to-high transition and the host places an 8-bit configuration word on the target device's data inputs. The host clocks the target device; new data should be presented by the host and latched by the target device every eight clock cycles.

On the first rising clock edge, a byte of configuration data is latched into the target device; the subsequent eight falling clock edges serialize the data in the device. On the ninth rising clock edge, the next byte of configuration data is latched and serialized into the target device. A status pin (RDYnBSY) on the target device indicates when it is serializing data and when it is ready to accept the next data byte. If an error occurs during configuration, the nSTATUS pin drives low. The host senses this low signal and begins reconfiguration or issues an error.

Once the target device configures successfully, it releases the CONF\_DONE pin. When CONF\_DONE goes high, it indicates that configuration is complete. After the last data byte, the DCLK pin must be clocked 40 times for APEX 20K devices and 10 times for FLEX 10K devices to release CONF\_DONE and initialize the device. See Figure 15.

**Figure 15. PPS Configuration Circuit** *Note (1)*

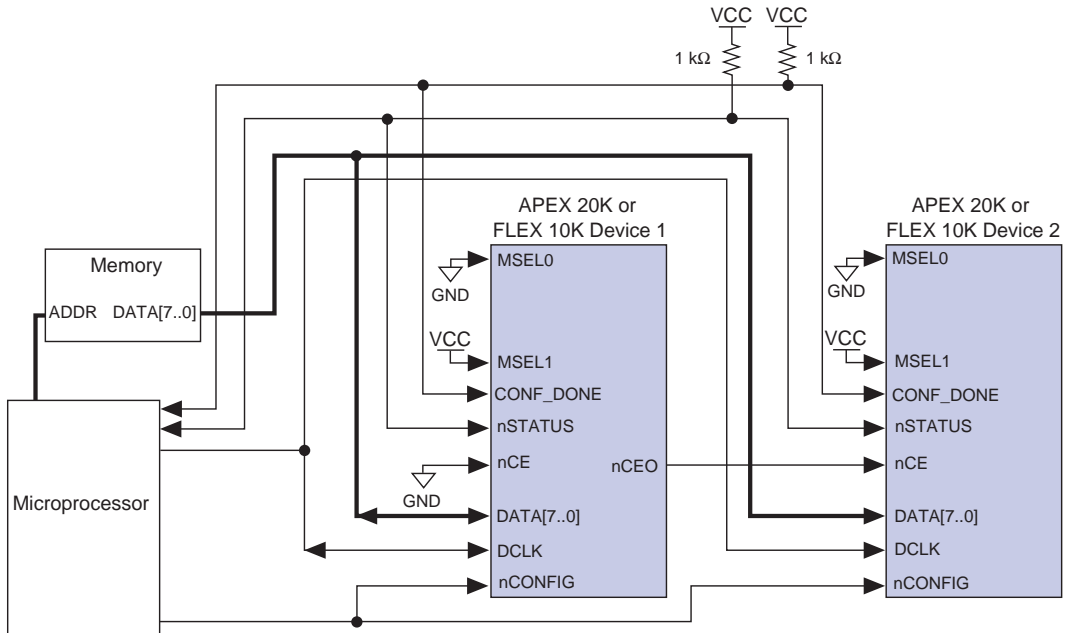


**Note:**

- (1) For FLEX 10K devices, the configuration word can be in Tabular Text File (.tff), Raw Binary File (.rbf), or Hexadecimal (.hex) format; for APEX 20K devices, the configuration word can be in RBF or Hex format. For information on how to create configuration and programming files, see “Device Configuration Files” on page 64.

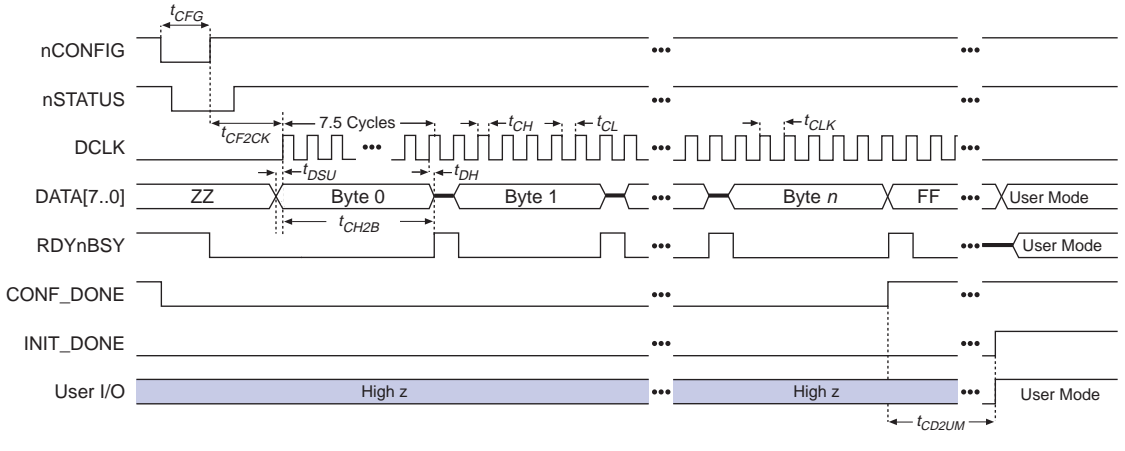
You can configure multiple APEX 20K or FLEX 10K devices in PPS mode by cascading the devices. Once the first device is configured, it drives its nCEO pin low, driving the second device's nCE pin low. The second device begins configuration within one clock cycle. Because all device CONF\_DONE pins are tied together, all devices initialize and enter user mode at the same time. In addition, all nSTATUS pins are tied together; thus, if any device detects an error, the entire chain is reset for automatic reconfiguration. See [Figure 16](#).

**Figure 16. PPS Multi-Device Configuration Circuit**



[Figure 17](#) shows the timing waveforms for PPS configuration for APEX 20K and FLEX 10K devices.

Figure 17. PPS Timing Waveform for APEX 20K &amp; FLEX 10K Devices



Tables 12 and 13 define the timing parameters for PPS configuration in APEX 20K and FLEX 10K devices.

Symbol	Parameter	Min	Max	Units
$t_{CF2CK}$	nCONFIG high to first rising edge on DCLK	40		$\mu\text{s}$
$t_{DSU}$	Data setup time before rising edge on DCLK	10		ns
$t_{DH}$	Data hold time after rising edge on DCLK	0		ns
$t_{CH2B}$	First rising DCLK to first rising RDYnBSY (1)	0.75		$\mu\text{s}$
$t_{CFG}$	nCONFIG low pulse width (2)	8		$\mu\text{s}$
$t_{CH}$	DCLK high time	30		ns
$t_{CL}$	DCLK low time	30		ns
$t_{CLK}$	DCLK period	60		ns
$f_{MAX}$	DCLK frequency		16.7	MHz
$t_{CD2UM}$	CONF_DONE high to user mode (3)	2	8	$\mu\text{s}$

**Notes:**

- (1) This parameter depends on the DCLK frequency. The RDYnBSY signal goes high 7.5 clock cycles after the rising edge of DCLK. This value was calculated with a DCLK frequency of 10 MHz.
- (2) This value applies only if the internal oscillator is selected as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 40 to obtain this value.
- (3) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 40 to obtain this value.

Symbol	Parameter	Min	Max	Units
$t_{CF2CK}$	nCONFIG high to first rising edge on DCLK	5		$\mu$ s
$t_{DSU}$	Data setup time before rising edge on DCLK	10		ns
$t_{DH}$	Data hold time after rising edge on DCLK	0		ns
$t_{CH2B}$	First rising DCLK to first rising RDYnBSY (1)	0.75		$\mu$ s
$t_{CFG}$	nCONFIG low pulse width (2)	2		$\mu$ s
$t_{CH}$	DCLK high time	30		ns
$t_{CL}$	DCLK low time	30		ns
$t_{CLK}$	DCLK period	60		ns
$f_{MAX}$	DCLK frequency		16	MHz
$t_{CD2UM}$	CONF_DONE high to user mode (3)	0.6	2	$\mu$ s

**Notes:**

- (1) This parameter depends on the DCLK frequency. The RDYnBSY signal goes high 7.5 clock cycles after the rising edge of DCLK. This value was calculated with a DCLK frequency of 10 MHz.
- (2) This value applies only if the internal oscillator is selected as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 10 to obtain this value.
- (3) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 10 to obtain this number.



For information on how to create configuration and programming files for this configuration scheme, see “[Device Configuration Files](#)” on page 64.

## PSA Configuration (FLEX 6000 Devices Only)

In passive serial asynchronous (PSA) configuration, a microprocessor drives data to a FLEX 6000 device via a download cable. When in PSA mode, you should pull the DCLK pin high using a 1-k $\Omega$  pull-up resistor to prevent unused configuration pins from floating.

To begin configuration, the microprocessor drives nCONFIG high and then pulls the FLEX 6000 device’s nCS pin low and CS pin high. The microprocessor places a configuration bit on the FLEX 6000 device’s DATA input and pulses nWS low to write data to the FLEX 6000 device. On the next rising edge of nWS, the FLEX 6000 device latches a bit of configuration data. Next, the FLEX 6000 device drives the RDYnBSY signal low, indicating that it is processing the configuration data. The microprocessor can then perform other system functions while the FLEX 6000 device is processing the data bit.



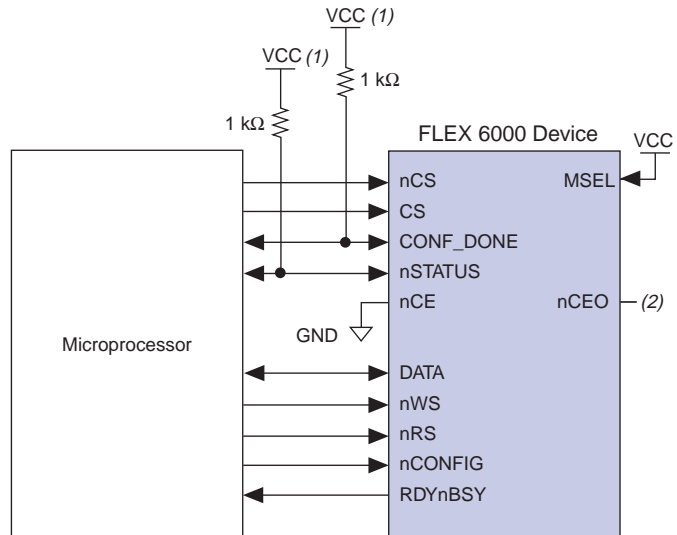
Afterward, the microprocessor checks `nSTATUS` and `CONF_DONE`. If the device asserts `nSTATUS` low, it has encountered an error and the microprocessor should restart configuration. If `nSTATUS` is not low and all configuration data has been received, the FLEX 6000 device is ready for initialization. At the beginning of initialization, `CONF_DONE` goes high to indicate that configuration is complete. If both `nSTATUS` and `CONF_DONE` are not low, the microprocessor sends the next data bit.

The microprocessor can also monitor the `CONF_DONE` and `INIT_DONE` pins to ensure successful configuration. If the microprocessor has sent all configuration data and has started initialization but `CONF_DONE` is not high, the microprocessor must reconfigure the FLEX 6000 device.

The MAX+PLUS II-generated programming files include the extra bits required to initialize the device in PSA configuration. However, in PSA configuration, the FLEX 6000 device can initialize itself. Therefore, the FLEX 6000 device asserts `CONF_DONE` high and initializes itself before all data is sent. The microprocessor can stop sending configuration data when `CONF_DONE` is asserted high.

The FLEX 6000 device's `nCS` or `CS` pins can be toggled during PSA configuration if the design meets the specifications specified for `tCSSU`, `tWSP`, and `tCSH` in Table 14 on page 38. Figure 18 shows PSA configuration for FLEX 6000 devices.

**Figure 18. PSA Configuration Circuit for FLEX 6000 Devices**



**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the FLEX 6000 device.
- (2) The `nCEO` pin is left unconnected.

An optional address decoder can control the device's  $nCS$  and  $CS$  pins. This decoder allows the microprocessor to select the FLEX 6000 device by accessing a particular address, simplifying the configuration process. The microprocessor can also control the  $nCS$  and  $CS$  signals directly. You can tie one of the  $nCS$  or  $CS$  signals to its active state (i.e.,  $nCS$  can be tied low) and the other signal can be toggled to control configuration.

The FLEX 6000 device can process data internally without the microprocessor. When the device is ready for the next bit of configuration data, it pulls  $RDYnBSY$  high, causing the microprocessor to strobe the next bit of configuration data into the FLEX 6000 device. Alternatively, the  $nRS$  signal can be strobed low, causing the  $RDYnBSY$  signal to appear on  $DATA$ . To simplify configuration, the microprocessor can wait for the  $t_{BUSY}$  time (the maximum  $RDYnBSY$  low pulse time) before sending the next data bit.

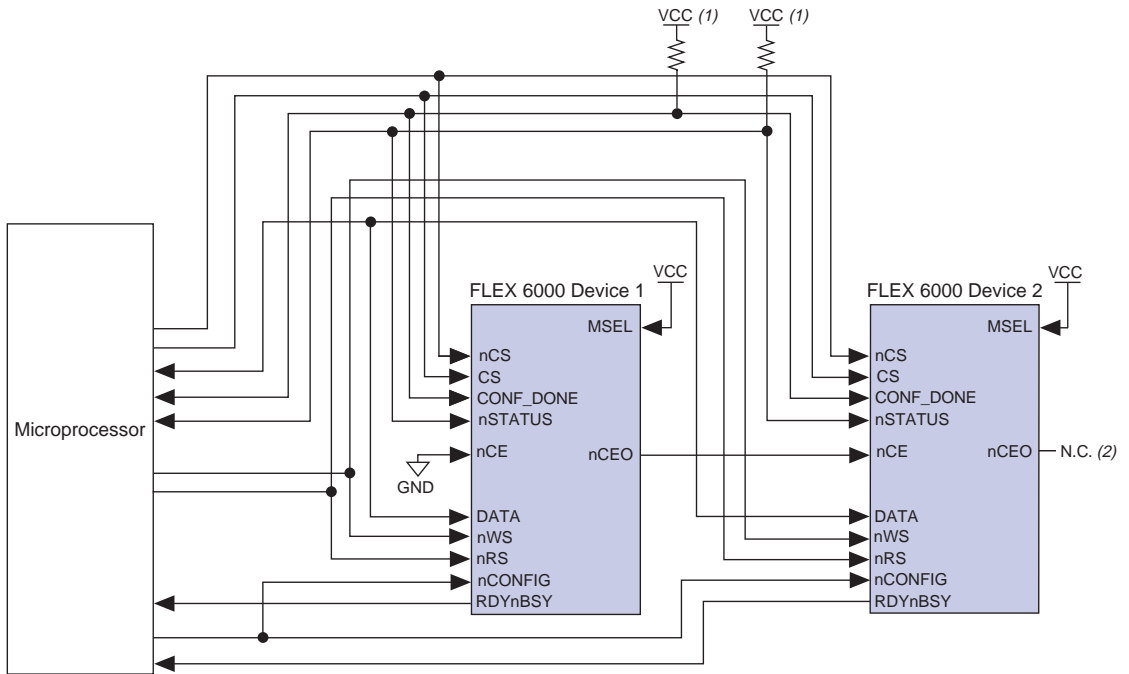
Because  $RDYnBSY$  does not need to be monitored, strobing  $nRS$  to read the state of the configuration data saves one system I/O port. You should not drive data onto the  $DATA$  pin while  $nRS$  is low because it causes contention. If the  $nRS$  pin is not used to monitor configuration, it should be tied high.

After configuration, the  $nCS$ ,  $CS$ ,  $nRS$ ,  $nWS$ , and  $RDYnBSY$  pins act as user I/O pins. However, when using a PSA scheme, these pins are tri-stated in user mode and should be driven by the microprocessor.

If the FLEX 6000 device detects an error during configuration, it drives  $nSTATUS$  low to alert the microprocessor. The microprocessor can then pulse  $nCONFIG$  low to restart the configuration process. Alternatively, if the *Auto-Restart Configuration on Frame Error* option is set in the MAX+PLUS II or Quartus software, the FLEX 6000 device releases  $nSTATUS$  after a reset time-out period. After  $nSTATUS$  is released, the microprocessor can reconfigure the FLEX 6000 device. At this point, the microprocessor does not need to pulse  $nCONFIG$  low.

PSA mode can also be used to configure multiple FLEX 6000 devices. Multi-device PSA configuration is similar to single-device PSA configuration, except that the FLEX 6000 devices are cascaded. After the first FLEX 6000 device is configured,  $nCEO$  is asserted low, which asserts the second device's  $nCE$  pin low, causing it to begin configuration. The second FLEX 6000 device begins configuration within one write cycle of the first device; therefore, the transfer of data destinations is transparent to the microprocessor. All FLEX 6000 device  $CONF\_DONE$  pins are tied together, so all FLEX 6000 devices initialize and enter user mode at the same time. If more than five FLEX 6000 devices are used, Altera recommends using buffers to split the fan-out on the  $DCLK$  signal. See [Figure 19](#).

Figure 19. PSA Multi-Device Configuration Circuit for FLEX 6000 Devices

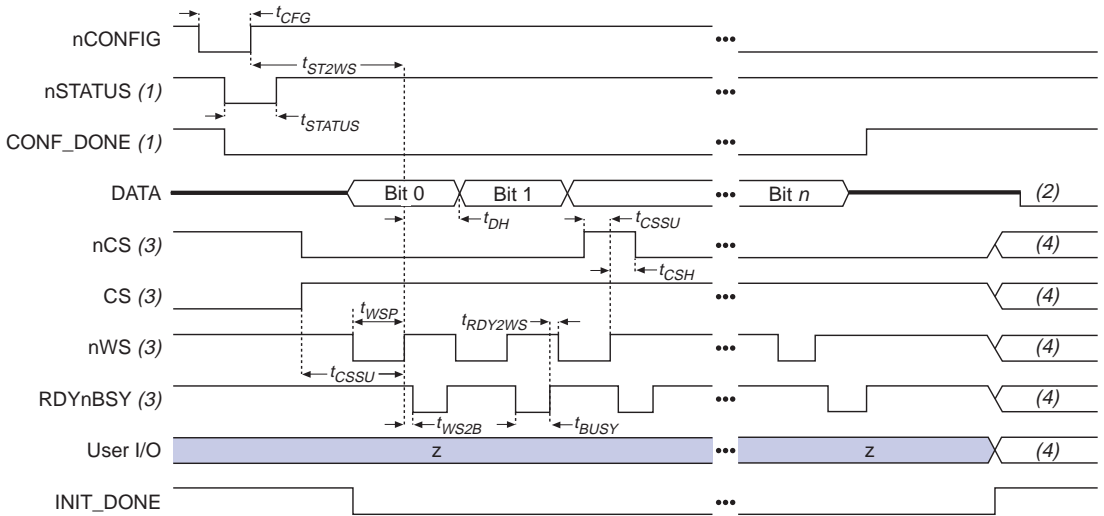


**Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the FLEX 6000 device.
- (2) The nCEO pin is left unconnected for the last device in the chain.

Figure 20 shows the FLEX 6000 timing waveforms for PSA configuration.

Figure 20. PSA Timing Waveforms in FLEX 6000 Devices

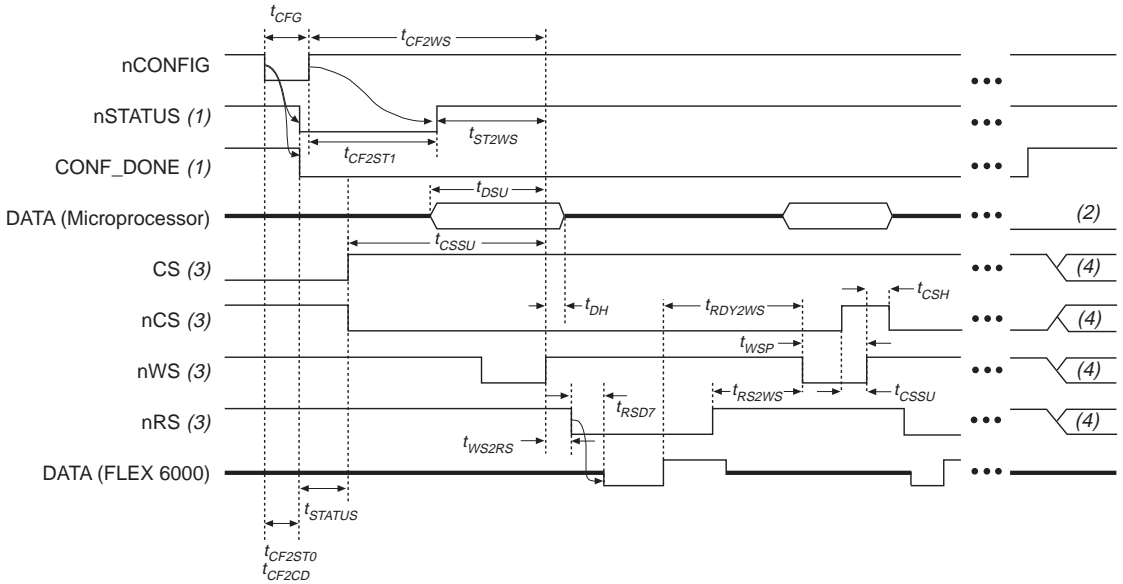


**Notes:**

- (1) Upon power-up, nSTATUS is held low not more than 5  $\mu$ s when  $V_{CC}$  reaches its minimum requirement.
- (2) DATA should not be left floating. It should be driven high or low, whichever is more convenient.
- (3) After configuration, the state of CS, nCS, nWS, and RDYnBSY depends on design programming in the FLEX 6000 device.
- (4) Device I/O pins are in user mode.

Figure 21 shows the FLEX 6000 timing waveforms when using a strobed nRS and nWS signal.

**Figure 21. PSA Timing Waveforms Using nRS & nWS in FLEX 6000 Devices**



**Notes:**

- (1) Upon power-up, nSTATUS is held low not more than 5  $\mu$ s when V<sub>CC</sub> reaches its minimum requirement.
- (2) DATA should not be left floating. It should be driven high or low, whichever is more convenient.
- (3) After configuration, the state of CS, nCS, nWS, nRS, and RDYnBSY depends on design programming in the FLEX 6000 device.
- (4) Device I/O pins are in user mode.

Table 14 summarizes the timing parameters for PSA configuration.

Symbol	Parameter	Min	Max	Units
$t_{CFG}$	nCONFIG low pulse width (1)	2		$\mu$ s
$t_{STATUS}$	nSTATUS low pulse width	2.5		$\mu$ s
$t_{CF2ST1}$	nCONFIG high to nSTATUS high		4	$\mu$ s
$t_{ST2WS}$	nSTATUS high to first rising edge on nWS	1		$\mu$ s
$t_{CF2WS}$	nCONFIG high to first rising edge on nWS	5		$\mu$ s
$t_{DSU}$	Data setup time before rising edge on nWS	20		ns
$t_{DH}$	Data hold time after rising edge on nWS	0		ns
$t_{CSSU}$	Chip select setup time before rising edge on nWS	20		ns
$t_{CSH}$	Chip select hold time after rising edge on nWS	5		ns
$t_{WSP}$	nWS low pulse width	50		ns
$t_{WS2B}$	nWS rising edge to RDYnBSY low		50	ns
$t_{BUSY}$	RDYnBSY low pulse width		200	ns
$t_{RDY2WS}$	RDYnBSY rising edge to nWS falling edge	50		ns
$t_{WS2RS}$	nWS rising edge to nRS falling edge	200		ns
$t_{RS2WS}$	nRS rising edge to nWS falling edge	200		ns
$t_{RSD7}$	nRS falling edge to DATA7 valid with RDYnBSY signal		50	ns
$t_{CF2CD}$	nCONFIG low to CONF_DONE low		1	$\mu$ s
$t_{CF2ST0}$	nCONFIG low to nSTATUS low		1	$\mu$ s

**Note:**

- (1) This value applies only if the internal oscillator is selected as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 10 to obtain this value.



For information on how to create configuration and programming files for this configuration scheme, see “[Device Configuration Files](#)” on page 64.

## PPA Configuration (APEX 20K & FLEX 10K Devices Only)

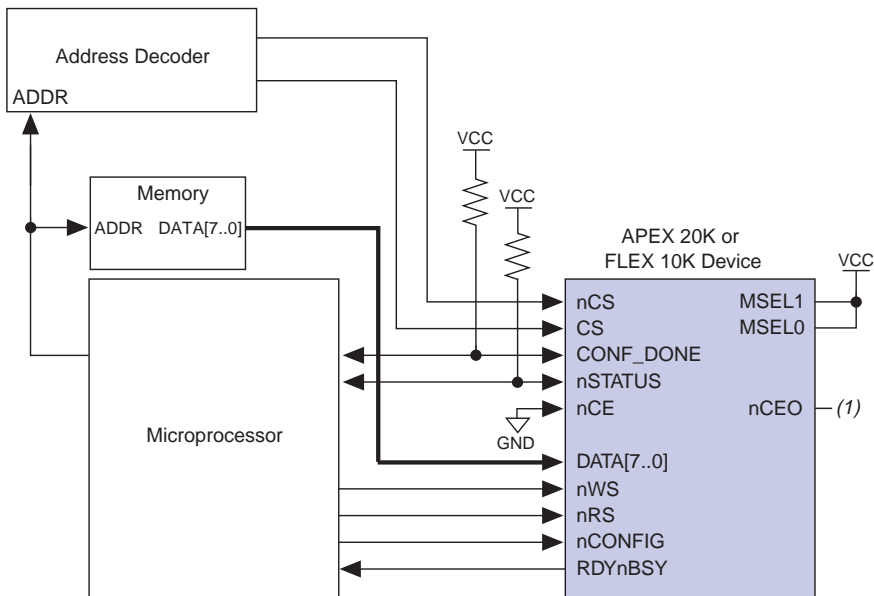
In passive parallel asynchronous (PPA) schemes, a microprocessor drives data to the APEX 20K or FLEX 10K device via a download cable. When using a PPA scheme, you should pull the DCLK pin high through a 1-k $\Omega$  pull-up resistor to prevent unused configuration pins from floating.

To begin configuration, the microprocessor drives nCONFIG high and then asserts the target device's nCS pin low and CS pin high. Next, the microprocessor places an 8-bit configuration word on the target device's data inputs and pulses nWS low. On the rising edge of nWS, the target device latches a byte of configuration data and then drives its RDYnBSY signal low, indicating that it is processing the byte of configuration data. The microprocessor can then perform other system functions while the APEX 20K or FLEX 10K device is processing the byte of configuration data.

Next, the microprocessor checks nSTATUS and CONF\_DONE. If both nSTATUS and CONF\_DONE are not low, the microprocessor sends the next data byte. If nSTATUS is low, the device is signaling an error and the microprocessor should restart configuration. However, if nSTATUS is not low and all the configuration data has been received, the device is ready for initialization. At the beginning of initialization, CONF\_DONE goes high to indicate that configuration is complete.

Figure 22 shows the PPA configuration circuit. An optional address decoder controls the device nCS and CS pins. This decoder allows the microprocessor to select the APEX 20K or FLEX 10K device by accessing a particular address, simplifying the configuration process.

Figure 22. PPA Configuration Circuit in APEX 20K & FLEX 10K Devices



**Note:**

(1) The nCEO pin is left unconnected.

The device's  $nCS$  or  $CS$  pins can be toggled during PPA configuration if the design meets the specifications specified for  $t_{CSSU}$ ,  $t_{WSP}$ , and  $t_{CSH}$  in [Tables 15 and 16 on page 44](#). The  $nCS$  and  $CS$  signals can also be controlled directly by the microprocessor. You can tie one of the  $nCS$  or  $CS$  signals to its active state (i.e.,  $nCS$  may be tied low) and the other signal can be toggled to control configuration.

APEX 20K or FLEX 10K devices can serialize data internally without the microprocessor. When the APEX 20K or FLEX 10K device is ready for the next byte of configuration data, it drives  $RDYnBSY$  high. If the microprocessor senses a high signal when it polls  $RDYnBSY$ , the microprocessor strobes the next byte of configuration data into the device. Alternatively, the  $nRS$  signal can be strobed, causing the  $RDYnBSY$  signal to appear on  $DATA7$ . Because  $RDYnBSY$  does not need to be monitored, reading the state of the configuration data by strobing  $nRS$  low saves a system I/O port. Data should not be driven onto the data bus while  $nRS$  is low because it causes contention on  $DATA7$ . If the  $nRS$  pin is not used to monitor configuration, it should be tied high. To simplify configuration, the microprocessor can wait for the  $t_{BUSY}$  time (the maximum  $RDYnBSY$  low pulse time) before sending the next data bit.

After configuration, the  $nCS$ ,  $CS$ ,  $nRS$ ,  $nWS$ , and  $RDYnBSY$  pins act as user I/O pins. However, when the PPA scheme is chosen in the MAX+PLUS II or Quartus software, these pins are tri-stated in user mode and should be driven by the microprocessor.

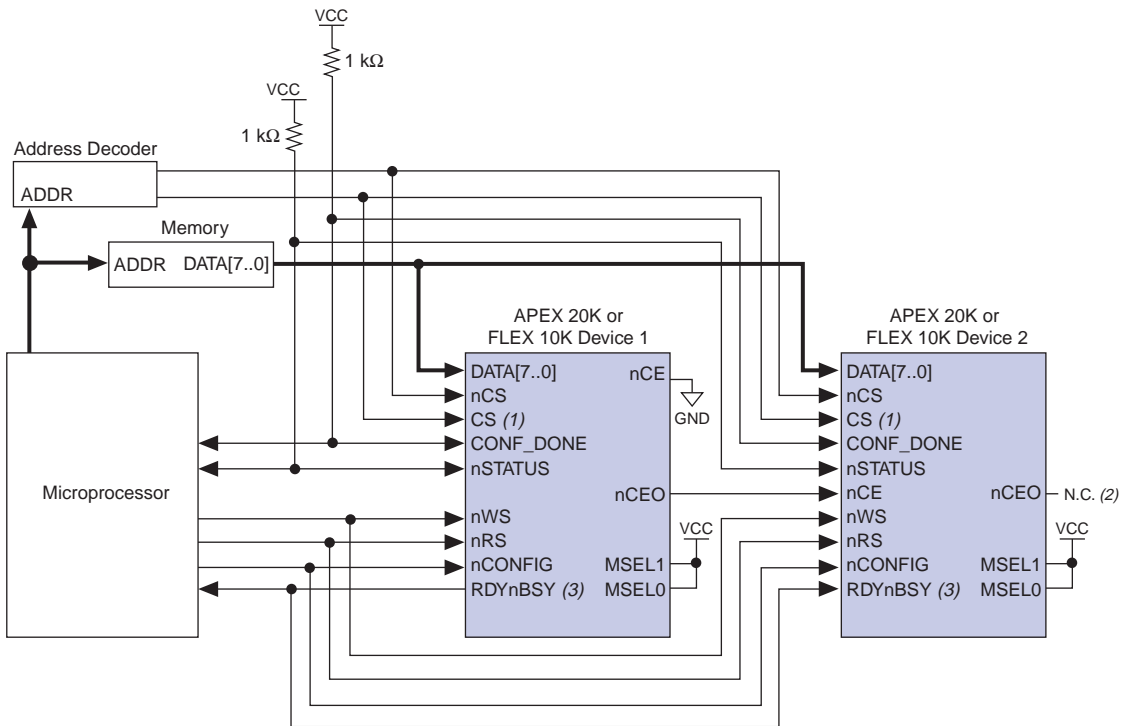
If the APEX 20K or FLEX 10K device detects an error during configuration, it drives  $nSTATUS$  low to alert the microprocessor. The microprocessor can then pulse  $nCONFIG$  low to restart the configuration process. Alternatively, if the *Auto-Restart Configuration on Frame Error* option has been set in the MAX+PLUS II or Quartus software, the APEX 20K or FLEX 10K device releases  $nSTATUS$  after a reset time-out period. After  $nSTATUS$  is released, the microprocessor can reconfigure the APEX 20K or FLEX 10K device. At this point, the microprocessor does not need to pulse  $nCONFIG$  low.

The microprocessor can also monitor the  $CONF\_DONE$  and  $INIT\_DONE$  pins to ensure successful configuration. The  $CONF\_DONE$  pin must be monitored by the microprocessor to detect errors and determine when programming completes. If the microprocessor sends all configuration data and starts initialization but  $CONF\_DONE$  is not asserted, the microprocessor must reconfigure the APEX 20K or FLEX 10K device.



PPA mode can also be used to configure multiple APEX 20K or FLEX 10K devices. Multi-device PPA configuration is similar to single-device PPA configuration, except that the APEX 20K or FLEX 10K devices are cascaded. After the first APEX 20K or FLEX 10K device is configured, nCEO is asserted, which asserts the nCE pin on the second device, causing it to begin configuration. The second APEX 20K or FLEX 10K device begins configuration within one write cycle of the first device; therefore, the transfer of data destinations is transparent to the microprocessor. All APEX 20K or FLEX 10K device CONF\_DONE pins are tied together, so all devices initialize and enter user mode at the same time. See Figure 23.

Figure 23. PPA Multi-Device Configuration Circuit for APEX 20K & FLEX 10K Devices

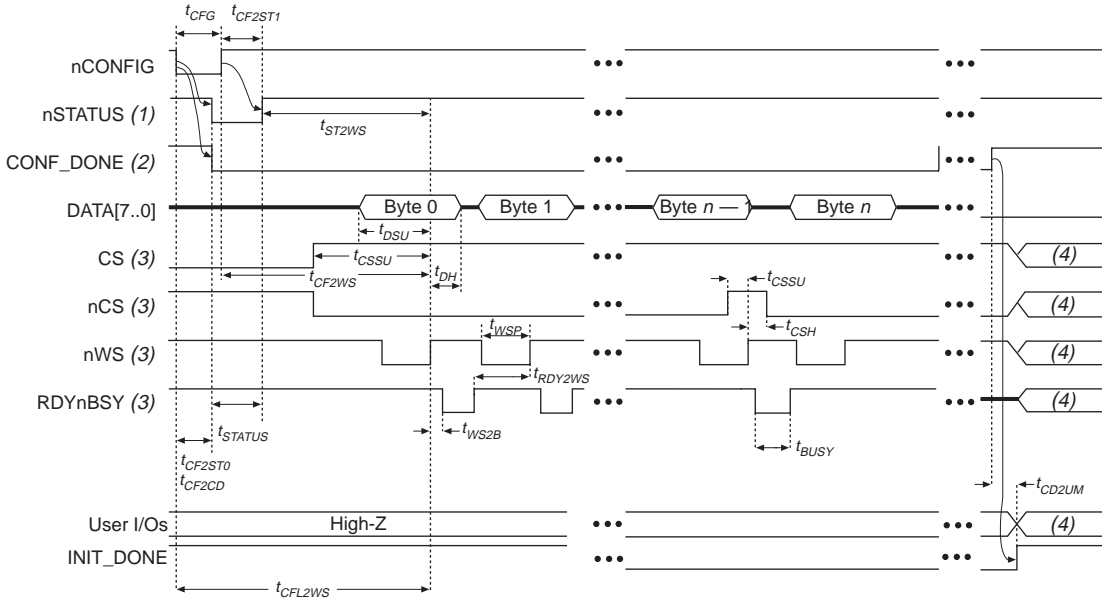


**Notes:**

- (1) If not used, the CS pin can be connected to V<sub>CC</sub> directly.
- (2) The nCEO pin is left unconnected for the last device in the chain.
- (3) When CONF\_DONE goes high, RDYnBSY is triggered low. Because CONF\_DONE does not go high until all devices in the chain are configured, all device RDYnBSY pins go high at the same time after all devices have been configured.

Figure 24 shows the APEX 20K and FLEX 10K timing waveforms for PPA configuration.

Figure 24. PPA Timing Waveforms for APEX 20K & FLEX 10K Devices

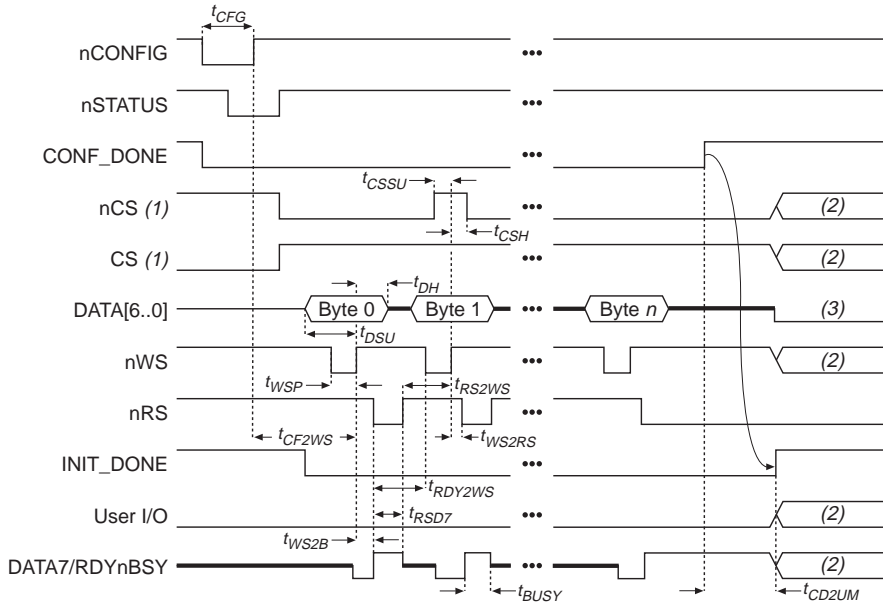


**Notes:**

- (1) Upon power-up, nSTATUS is held low not more than 5  $\mu$ s when  $V_{CC}$  reaches its minimum requirement.
- (2) Upon power-up, CONF\_DONE is low.
- (3) After configuration, the state of CS, nCS, nWS, and RDYnBSY depends on the design programmed into the APEX 20K or FLEX 10K device.
- (4) Device I/O pins are in user mode.

Figure 25 shows the APEX 20K or FLEX 10K timing waveforms when using a strobed nRS and nWS signal.

Figure 25. PPA Timing Waveforms Using nRS & nWS



**Notes:**

- (1) The user can toggle nCS or CS during configuration if the design meets the specification for  $t_{CSSU}$ ,  $t_{WSP}$ , and  $t_{CSH}$ .
- (2) Device I/O pins are in user mode.
- (3) DATA0 should not be left floating. It should be driven high or low, whichever is more convenient.

Tables 15 and 16 define the APEX 20K and FLEX 10K timing parameters for PPA configuration.

Symbol	Parameter	Min	Max	Units
<b>t<sub>CF2WS</sub></b>	nCONFIG high to first rising edge on nWS	40		μs
<b>t<sub>DSU</sub></b>	Data setup time before rising edge on nWS	10		ns
<b>t<sub>DH</sub></b>	Data hold time after rising edge on nWS	0		ns
<b>t<sub>CSSU</sub></b>	Chip select setup time before rising edge on nWS	10		ns
<b>t<sub>CSH</sub></b>	Chip select hold time after rising edge on nWS	0		ns
<b>t<sub>WSP</sub></b>	nWS low pulse width	200		ns
<b>t<sub>CFG</sub></b>	nCONFIG low pulse width (2)	8		μs
<b>t<sub>WS2B</sub></b>	nWS rising edge to RDYnBSY low		50	ns
<b>t<sub>BUSY</sub></b>	RDYnBSY low pulse width	0.4	1.6	μs
<b>t<sub>RDY2WS</sub></b>	RDYnBSY rising edge to nWS rising edge	50		ns
<b>t<sub>WS2RS</sub></b>	nWS rising edge to nRS falling edge	200		ns
<b>t<sub>RS2WS</sub></b>	nRS rising edge to nWS rising edge	200		ns
<b>t<sub>RS2D7</sub></b>	nRS falling edge to DATA7 valid with RDYnBSY signal		50	μs
<b>t<sub>CD2UM</sub></b>	CONF_DONE high to user mode (3)	2	8	μs

Symbol	Parameter	Min	Max	Units
<b>t<sub>CF2WS</sub></b>	nCONFIG high to first rising edge on nWS	5		μs
<b>t<sub>DSU</sub></b>	Data setup time before rising edge on nWS	20		ns
<b>t<sub>DH</sub></b>	Data hold time after rising edge on nWS	0		ns
<b>t<sub>CSSU</sub></b>	Chip select setup time before rising edge on nWS	20		ns
<b>t<sub>CSH</sub></b>	Chip select hold time after rising edge on nWS	10 (4) 15 (5)		ns
<b>t<sub>WSP</sub></b>	nWS low pulse width	200		ns
<b>t<sub>CFG</sub></b>	nCONFIG low pulse width	2		μs
<b>t<sub>WS2B</sub></b>	nWS rising edge to RDYnBSY low		50	ns
<b>t<sub>BUSY</sub></b>	RDYnBSY low pulse width	0.4	1.6	μs
<b>t<sub>RDY2WS</sub></b>	RDYnBSY rising edge to nWS rising edge	50		ns
<b>t<sub>WS2RS</sub></b>	nWS rising edge to nRS falling edge	200		ns
<b>t<sub>RS2WS</sub></b>	nRS rising edge to nWS rising edge	200		ns
<b>t<sub>RS2D7</sub></b>	nRS falling edge to DATA7 valid with RDYnBSY signal		50	ns
<b>t<sub>CD2UM</sub></b>	CONF_DONE high to user mode (6)	0.6	2	μs

**Notes to tables:**

- (1) This information is preliminary.
- (2) This value applies only if the internal oscillator is selected as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 40 to obtain this value.
- (3) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 40 to obtain this value.
- (4) This parameter value applies to EPF10K10, EPF10K20, EPF10K40, EPF10K50, all FLEX 10KA, and FLEX 10KE devices.
- (5) This parameter value applies to EPF10K70 and EPF10K100 devices only.
- (6) The minimum and maximum numbers apply only if the internal oscillator is chosen as the clock source for starting up the device. If the clock source is CLKUSR or DCLK, multiply the clock period by 10 to obtain this value.



For information on how to create configuration and programming files for this configuration scheme, see [“Device Configuration Files” on page 64](#).

## JTAG Programming & Configuration (APEX 20K & FLEX 10K Devices Only)

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



For more information on JTAG boundary-scan testing, see [Application Note 39 \(IEEE 1149.1 \(JTAG\) Boundary-Scan Testing in Altera Devices\)](#).

A device operating in JTAG mode uses four required pins, TDI, TDO, TMS, and TCK, and one optional pin, TRST. All other pins are tri-stated during JTAG configuration. You should not begin JTAG configuration until all other configuration is complete. [Table 17](#) shows each JTAG pin’s function.

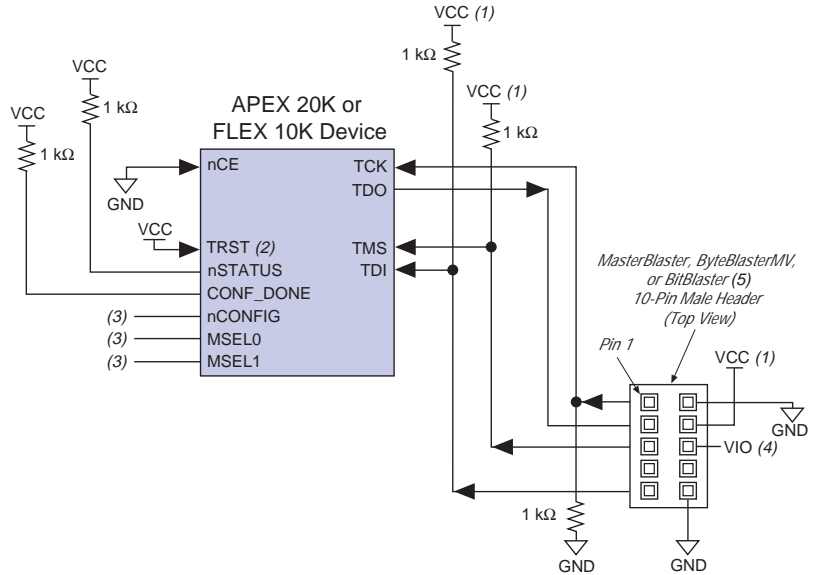
**Table 17. JTAG Pin Descriptions**

Pin	Description	Function
TDI	Test data input	Serial input pin for instructions as well as test and programming data. Data is shifted in on the rising edge of TCK.
TDO	Test data output	Serial data output pin for instructions as well as test and programming data. Data is shifted out on the falling edge of TCK. The pin is tri-stated if data is not being shifted out of the device.
TMS	Test mode select	Input pin that provides the control signal to determine the transitions of the TAP controller state machine. Transitions within 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.
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.
TRST (1)	Test reset input (optional)	Active-low input to asynchronously reset the boundary-scan circuit. The TRST pin is optional according to IEEE Std. 1149.1.

**Note:**

- (1) FLEX 10K devices in 144-pin thin quad flat pack (TQFP) packages do not have a TRST pin. Therefore, the TRST pin can be ignored when using these devices.

During JTAG configuration, data is downloaded to the device on the PCB through the MasterBlaster, ByteBlasterMV, or BitBlaster header. Configuring devices through a cable is similar to programming devices in-system, except the TRST pin should be connected to  $V_{CC}$ ; this connection ensures that the TAP controller is not reset. See [Figure 26](#).

**Figure 26. JTAG Configuration of a Single APEX 20K or FLEX 10K Device****Notes:**

- (1) The pull-up resistor should be connected to the same supply voltage as the MasterBlaster, ByteBlasterMV, or BitBlaster cable.
- (2) FLEX 10K devices in 144-pin TQFP packages do not have a TRST pin. Therefore, the TRST pin can be ignored when configuring FLEX 10K devices in 144-pin TQFP packages.
- (3) The nCONFIG, MSEL0, and MSEL1 pins should be connected to support a FLEX configuration scheme. If only JTAG configuration is used, connect nCONFIG to VCC, and MSEL0 and MSEL1 to ground.
- (4) VIO is a reference voltage for the MasterBlaster output driver. VIO should match the device's V<sub>CCIO</sub>. Refer to the [MasterBlaster Serial/USB Communications Cable Data Sheet](#) for this value.
- (5) The BitBlaster serial download cable is not supported by the Quietus software and can not be used to configure APEX devices.

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.

APEX 20K and FLEX 10K devices have dedicated JTAG pins that always function as JTAG pins. JTAG testing can be performed on APEX 20K, FLEX 10K, and FLEX 6000 devices both before and after configuration, but not during configuration. The chip-wide reset and output enable pins on APEX 20K, FLEX 10K, and FLEX 6000 devices do not affect JTAG boundary-scan or programming operations. Toggling these pins does not affect JTAG operations (other than the usual boundary-scan operation).

When designing a board for JTAG configuration of APEX 20K or FLEX 10K devices, the regular configuration pins should be considered. [Table 18](#) shows how these pins should be connected during JTAG configuration.

Signal	Description
nCE	All APEX 20K or FLEX 10K devices in the chain should be driven low by connecting nCE to ground, pulling it down via a resistor, or driving it by some control circuitry.
nSTATUS	Pulled to V <sub>CC</sub> via a 1-kΩ resistor. When configuring multiple devices in the same JTAG chain, each nSTATUS pin should be pulled up to V <sub>CC</sub> individually. (1)
CONF_DONE	Pulled to V <sub>CC</sub> via a 1-kΩ resistor. When configuring multiple devices in the same JTAG chain, each CONF_DONE pin should be pulled up to V <sub>CC</sub> individually. (1)
nCONFIG	Driven high by connecting to V <sub>CC</sub> , pulling up via a resistor, or driven by some control circuitry.
MSELO, MSEL1	These pins must not be left floating. These pins support whichever non-JTAG configuration is used in production. If only JTAG configuration is used, you should tie both pins to ground.
DCLK	Should not be left floating. Drive low or high, whichever is more convenient.
DATA0	Should not be left floating. Drive low or high, whichever is more convenient.
TRST	This JTAG pin is not connected to the download cable. It should be driven to logic high.

**Note:**

- (1) nSTATUS pulling low in the middle of JTAG configuration indicates that an error has occurred; CONF\_DONE pulling high at the end of JTAG configuration indicates successful configuration.



For information on how to create configuration and programming files for this configuration scheme, see [“Device Configuration Files” on page 64](#).

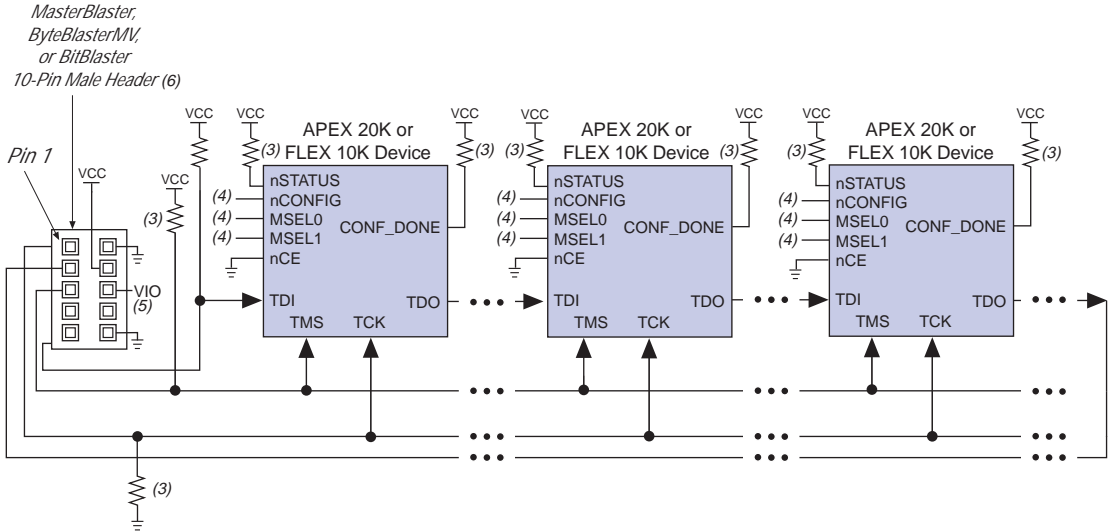
## JTAG Programming & Configuration of Multiple Devices (APEX 20K & FLEX 10K Devices Only)

When programming a JTAG device chain, one JTAG-compatible header, such as the ByteBlasterMV or BitBlaster 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. However, when more than five 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 PCB contains multiple devices, or when testing the PCB using JTAG BST circuitry. [Figure 27](#) shows multi-device JTAG configuration.

**Figure 27. Multi-Device JTAG Configuration** *Notes (1), (2)*



**Notes:**

- (1) APEX 20K, FLEX 10K, and MAX devices can be placed within the same JTAG chain for device programming and configuration.
- (2) For more information on all configuration pins connected in this mode, refer to [Table 18 on page 48](#).
- (3) All pull-up/pull-down resistors are 1 k $\Omega$ .
- (4) The nCONFIG, MSEL0, and MSEL1 pins should be connected to support a FLEX configuration scheme. If only JTAG configuration is used, connect nCONFIG to V<sub>CC</sub>, and MSEL0 and MSEL1 to ground.
- (5) VIO is a reference voltage for the MasterBlaster output driver. VIO should match the device's V<sub>CCIO</sub>. Refer to the [MasterBlaster Serial/USB Communications Cable Data Sheet](#) for this value.
- (6) The BitBlaster serial download cable is not supported by the Quietus software and can not be used to configure APEX devices.

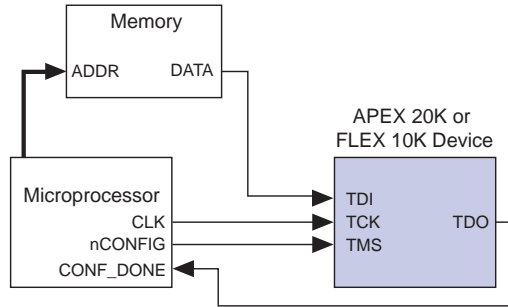
Successful JTAG configuration is verified automatically by the MAX+PLUS II or Quartus software at the end of JTAG configuration. The software checks the state of CONF\_DONE through the JTAG port at the end of configuration. If CONF\_DONE is not in the correct state, the MAX+PLUS II or Quartus software indicates that configuration has failed. If CONF\_DONE is in the correct state, the software indicates that configuration was successful.



When using the JTAG pins for configuration, if V<sub>CCIO</sub> is tied to 3.3 V, both the I/O pins and JTAG TDO port will drive at 3.3-V levels. The TDO pin meets JTAG requirements at any specified voltage level.

JTAG and non-JTAG configuration should not be attempted simultaneously. When configuring via JTAG, allow any non-JTAG configuration to complete first. Figure 28 shows the JTAG configuration of an APEX 20K or FLEX 10K device with a microprocessor.

**Figure 28. JTAG Configuration of APEX 20K or FLEX 10K Device with a Microprocessor**



## Jam Programming & Test Language

In-circuit configuration via an embedded processor enables easy design prototyping, streamlines production, and allows quick and efficient in-field upgrades. The Jam™ programming and test language, a new standard file format using the IEEE Std. 1149.1 (JTAG) interface, further simplifies in-circuit configuration by providing small file sizes and increased flexibility. Jam Files and Jam Byte-Code Files (.jbc) contain both the programming algorithm and data required to upgrade one or more devices. The Jam language is supported by MAX+PLUS II software versions 8.0 and higher.

You can estimate the JBC size using the following equation:

$$\text{JBC Size} = \text{Alg} + \sum_{k=1}^N \text{Data}$$

- Where:
- $\text{Alg}$  = Space used by the algorithm (see Table 19)
  - $\text{Data}$  = Space used by compressed programming data (see Table 20)
  - $k$  = Index representing family type(s) being targeted
  - $N$  = Number of target devices in the chain

Device Family	Typical Algorithm Size (Kbytes)	
	ASCII Jam File	Jam Byte-Code File
FLEX 10K FLEX 10KE	6	4

Device	Typical Data Size (Kbytes) Compressed	
	ASCII Jam	JBC
EPF10K10, EPF10K10A	11	8
EPF10K20	23	17
EPF10K30, EPF10K30A, EPF10K30E	39	28
EPF10K40	51	37
EPF10K50, EPF10K50E, EPF10K50V	60	44
EPF10K70	95	69
EPF10K100, EPF10K100A, EPF10K100B, EPF10K100E	130	95
EPF10K130E, EPF10K130V	197	128
EPF10K200E	196	143
EPF10K250A	245	179

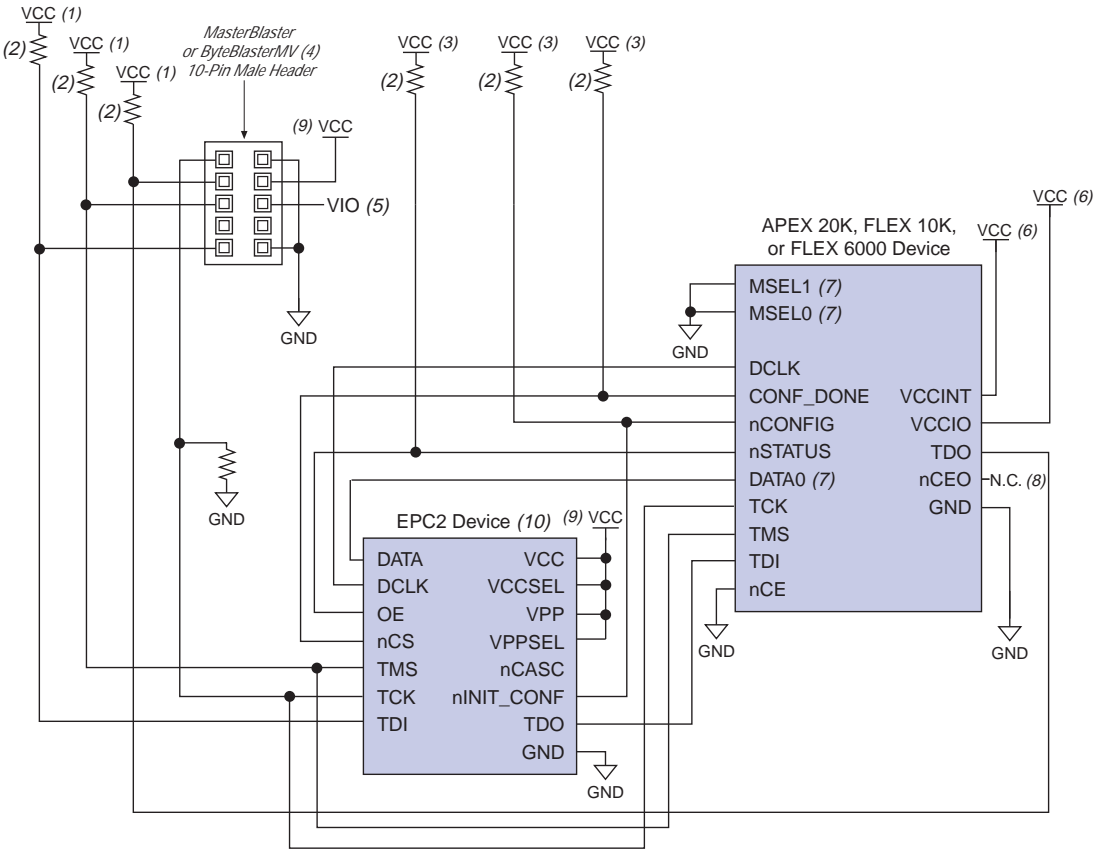


For more information on how to configure devices using the Jam programming and test language, see [Application Note 88 \(Using the Jam Language for ISP & ICR via an Embedded Processor\)](#). A future version of the application note will contain the Jam File and Jam Byte-Code File sizes for APEX 20K devices.

## Combining Different Configuration Schemes

This section shows how to configure APEX 20K, FLEX 10K, and FLEX 6000 devices using multiple configuration schemes on the same board. [Figure 29](#) shows a schematic for configuring an APEX 20K, FLEX 10K, or FLEX 6000 device using a download cable and a configuration device.

Figure 29. Device Configuration with a Download Cable & Configuration Device

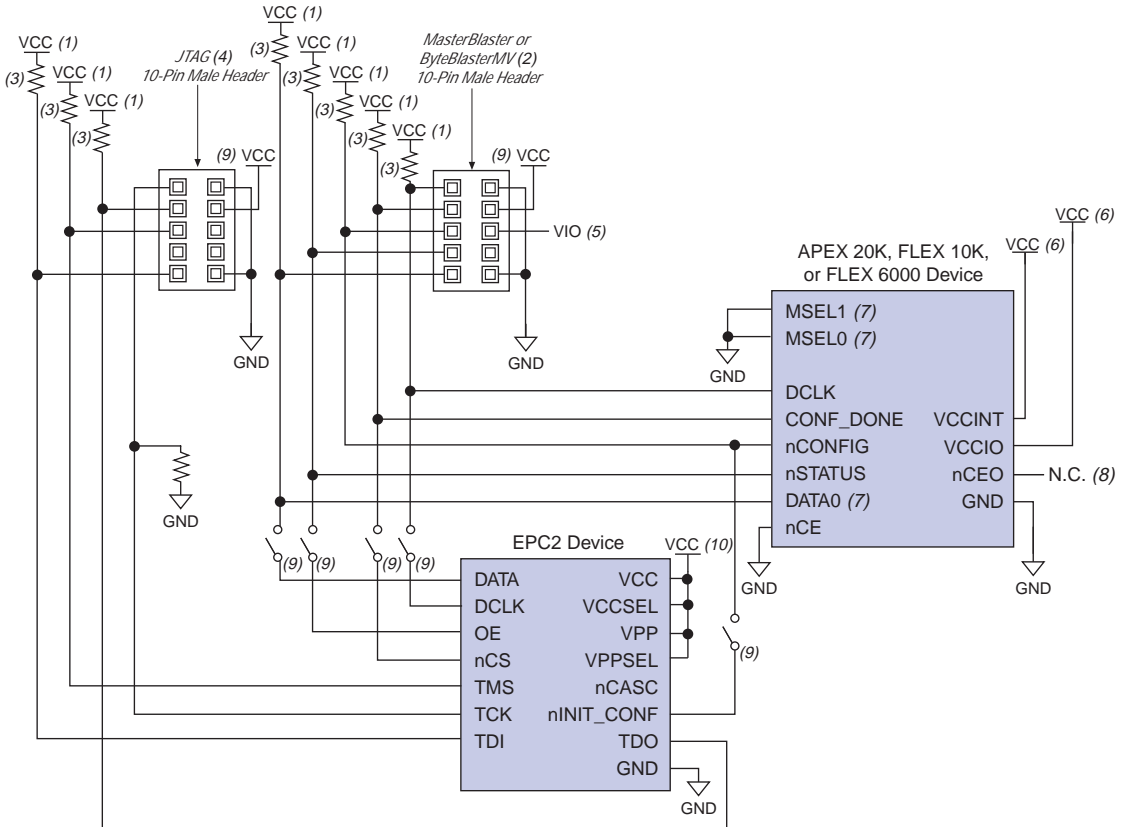


**Notes:**

- (1) V<sub>CC</sub> should be connected to the same supply voltage as the configuration device.
- (2) All pull-up resistors are 1 kΩ.
- (3) The pull-up resistor should be connected to the same supply voltage as the configuration device.
- (4) The download cable programs the configuration device.
- (5) V<sub>IO</sub> is a reference voltage for the MasterBlaster output driver. V<sub>IO</sub> should match the device's V<sub>CCIO</sub>. Refer to the [MasterBlaster Serial/USB Communications Cable Data Sheet](#) for this value.
- (6) V<sub>CCINT</sub> and V<sub>CCIO</sub> should be applied according to the target device's V<sub>CCINT</sub> and V<sub>CCIO</sub>.
- (7) FLEX 6000 devices have a single MSEL pin, which is tied to ground, and its DATA0 pin is named DATA.
- (8) The nCEO pin is left unconnected.
- (9) If a 3.3-V supply voltage is used, the V<sub>CC</sub>, V<sub>CCSEL</sub>, V<sub>PP</sub>, and V<sub>PPSEL</sub> pins should be connected to a 3.3-V supply. If a 5.0-V supply voltage is used, the V<sub>CC</sub> and V<sub>PP</sub> pins should be connected to a 5.0-V supply, and the V<sub>CCSEL</sub> and V<sub>PPSEL</sub> pins should be connected to ground. To improve in-system programming times, you can connect V<sub>PP</sub> to 5.0 V, V<sub>CC</sub> to 3.3 V, and V<sub>PPSEL</sub> to ground. For more information on these pins, see [Table 23 on page 62](#).
- (10) The configuration device configures the APEX 20K, FLEX 10K, or FLEX 6000 device. This figure shows the pin connections for an EPC2 configuration device. For any other configuration device, connect the pins appropriately.

Figure 30 shows a schematic for configuring APEX 20K, FLEX 10K, or FLEX 6000 devices using two download cables and an EPC2 device.

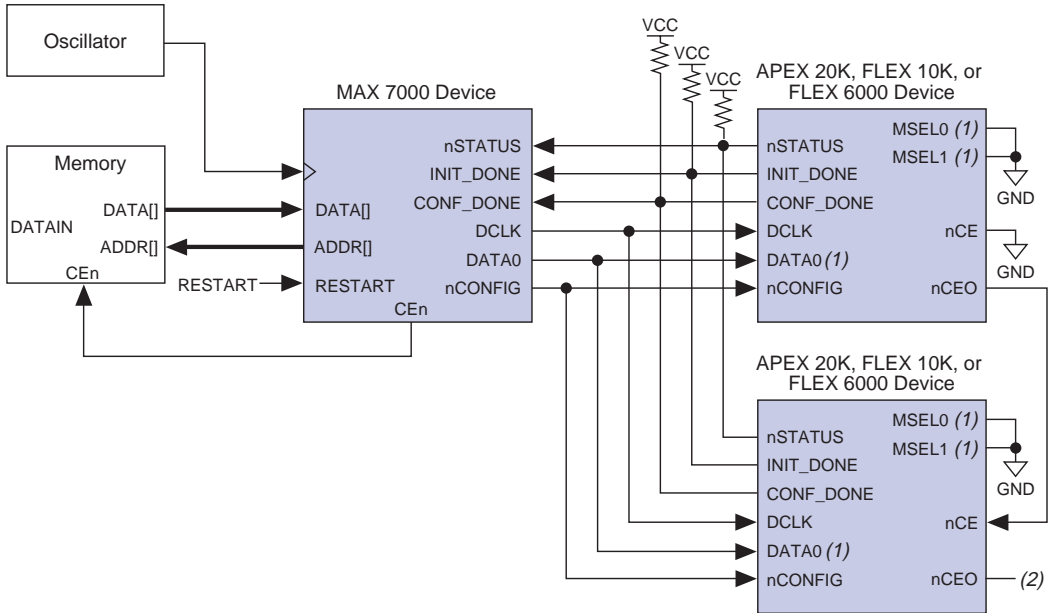
Figure 30. Device Configuration with Two Download Cables &amp; EPC2 Device

**Notes:**

- (1)  $V_{CC}$  should be connected to the same supply voltage as the configuration device.
- (2) The target APEX 20K or FLEX 10K device can be configured by either the configuration device or the download cable.
- (3) All pull-up resistors are 1 k $\Omega$ .
- (4) The download cable programs the configuration device through the JTAG circuitry.
- (5) VIO is a reference voltage for the MasterBlaster output driver. VIO should match the device's  $V_{CCIO}$ . Refer to the [MasterBlaster Serial/USB Communications Cable Data Sheet](#) for this value.
- (6)  $V_{CCINT}$  and  $V_{CCIO}$  should be applied according to the target device's  $V_{CCINT}$  and  $V_{CCIO}$ .
- (7) FLEX 6000 devices have a single MSEL pin, which is tied to ground, and its DATA0 pin is named DATA.
- (8) The nCEO pin is left unconnected.
- (9) You should not attempt configuration with a download cable while a configuration device is connected to an APEX 20K or FLEX 10K device. To perform this operation, you should either remove the configuration device from its socket when using the download cable, or place a switch on the five common signals between the download cable and the configuration device.
- (10) If a 3.3-V supply voltage is used, the VCC, VCCSEL, VPP, and VPPSEL pins should be connected to a 3.3-V supply voltage. If a 5.0-V supply voltage is used, the VCC and VPP pins should be connected to a 5.0-V supply voltage, and the VCCSEL and VPPSEL pins should be connected to ground. To improve in-system programming times, you can connect VPP to 5.0 V, VCC to 3.3 V, and VPPSEL to ground. For more information on these pins, see [Table 23 on page 62](#).

You can use external memory to configure an APEX 20K, FLEX 10K, or FLEX 6000 device by using a MAX 7000 device. Figure 31 shows the schematic for this configuration scheme. A sample design for the MAX 7000 device, called `promldr.txt`, is available on the Altera web site at <http://www.altera.com>.

Figure 31. Device Configuration using External Memory & a MAX 7000 Device

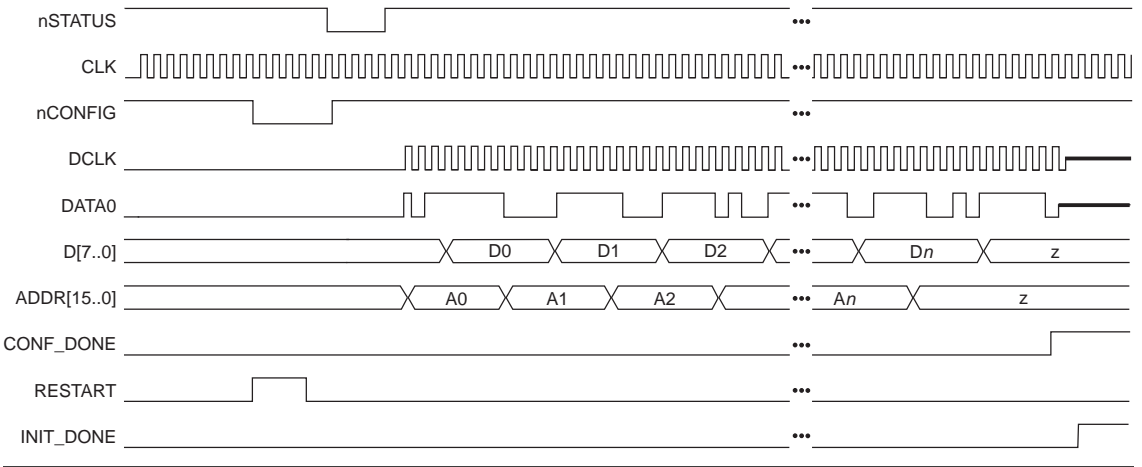


**Notes:**

- (1) FLEX 6000 devices have a single MSEL pin, which is tied to ground, and its DATA0 pin is named DATA.
- (2) The nCEO pin is left unconnected for the last device in the chain.

Figure 32 shows the timing waveform for configuring an APEX 20K, FLEX 10K, or FLEX 6000 device using external memory and a MAX 7000 device.

**Figure 32. Timing Waveform for Configuration using External Memory & a MAX 7000 Device**



## Device Options

You can set FLEX 10K and FLEX 6000 device options in Altera’s MAX+PLUS II development software by choosing **Global Project Device Options** (Assign menu). You can set APEX 20K device options using a similar dialog box in the Quartus software. [Table 21](#) summarizes each of these options.

Device Option	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
User-supplied start-up clock (FLEX 10K and FLEX 6000 devices only.)	To complete initialization, the device must be clocked 10 times after all data is transferred. <code>CONF_DONE</code> goes high after the initialization process has begun. You can select which clock source to use for initialization by choosing <code>CLKUSR</code> rather than <code>DCLK</code> .	In the PPA and PSA configuration schemes, the device’s internal oscillator supplies the initialization clock.  In the configuration device, PS, and PPS schemes, the internal oscillator is disabled. Thus, external circuitry, such as a configuration device or download cable, must provide the initialization clock on the <code>DCLK</code> pin. In the configuration device scheme, the configuration device supplies the clock; in PS and PPS schemes, the microprocessor supplies the clock.	The user provides the clock on the <code>CLKUSR</code> pin. This clock can synchronize the initialization of multiple devices. The clock should be supplied when the last data byte is transferred. Supplying the clock during configuration will not affect the configuration process. The operation of the <code>CLKUSR</code> pin during user mode is selected in the MAX+PLUS II software.

**Table 21. APEX 20K, FLEX 10K & FLEX 6000 Configuration Option Bits (Part 2 of 3)**

Device Option	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
User-supplied start-up clock (APEX 20K devices only.)	To begin initialization, the device must be clocked 40 times after all data is transferred. <code>CONF_DONE</code> goes high after the initialization process begins. The device can be initialized by the internal oscillator or by external clocks provided on <code>DCLK</code> or <code>CLKUSR</code> .	The device's internal oscillator supplies the start-up clock.	The user provides the clock on the <code>CLKUSR</code> or <code>DCLK</code> pin. The Quartus software specifies which pin is used. This clock synchronizes the initialization of multiple devices. The clock should be supplied when the last data byte is transferred. Supplying a clock on <code>CLKUSR</code> will not affect the configuration process. The Quartus software specifies the <code>CLKUSR</code> pin's operation mode. When using a configuration device, you can use <code>CLKUSR</code> to synchronize the initialization; <code>DCLK</code> can be used in passive configuration only.
Auto-restart configuration on frame error	If a data error occurs during configuration, you can choose how to restart configuration.	The configuration process stops until you direct the device to restart configuration. The <code>nSTATUS</code> pin is driven low when an error occurs. When <code>nCONFIG</code> is pulled low and then high, the device begins to reconfigure.	<p>The configuration process restarts automatically. The <code>nSTATUS</code> pin drives low and releases. The <code>nSTATUS</code> pin is then pulled to <math>V_{CC}</math> by the pull-up resistor, indicating that configuration can restart.</p> <p>In the configuration device scheme, if the target device's <code>nSTATUS</code> pin is tied to the configuration device's <code>OE</code> pin, the <code>nSTATUS</code> reset pulse resets the configuration device automatically. The configuration device releases its <code>OE</code> pin (which is pulled high) and reconfiguration begins.</p> <p>If an error occurs during passive configuration, the device can be reconfigured without the system having to pulse <code>nCONFIG</code>. After <code>nSTATUS</code> goes high, reconfiguration can begin.</p>



**Table 21. APEX 20K, FLEX 10K & FLEX 6000 Configuration Option Bits (Part 3 of 3)**

Device Option	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
Release clears before tri-states	During configuration, the device I/O pins are tri-stated. During initialization, you choose the order for releasing the tri-states and clearing the registers.	The device releases the tri-states on its I/O pins before releasing the clear signal on its registers.	The device releases the clear signals on its registers before releasing the tri-states. You can use this option to allow the design to operate before it drives out, so all outputs do not start up low.
Enable chip-wide reset	Enables a single pin to reset all device registers.	Chip-wide reset is not enabled. The <code>DEV_CLRn</code> pin is available as a user I/O pin.	Chip-wide reset is enabled for all registers in the device. All registers are cleared when the <code>DEV_CLRn</code> pin is driven low.
Enable chip-wide output enable	Enables a single pin to control all device tri-states.	Chip-wide output enable is not enabled. The <code>DEV_OE</code> pin is available as a user I/O pin.	Chip-wide output enable is enabled for all device tri-states. After configuration, all user I/O pins are tri-stated when <code>DEV_OE</code> is low.
Enable <code>INIT_DONE</code> output	Enables a pin to drive out a signal when the initialization process is complete and the device has entered user mode.	The <code>INIT_DONE</code> signal is not available. The <code>INIT_DONE</code> pin is available as a user I/O pin.	The <code>INIT_DONE</code> signal is available on the open-drain <code>INIT_DONE</code> pin. This pin drives low during configuration. After initialization, it is released and pulled high externally. The <code>INIT_DONE</code> pin must be connected to a 1-k $\Omega$ pull-up resistor. If the <code>INIT_DONE</code> output is used, the <code>INIT_DONE</code> pin cannot be used as a user I/O pin.
Enable JTAG support (FLEX 6000 devices only. In APEX 20K and FLEX 10K devices, JTAG support is always enabled.)	Enables post-configuration JTAG boundary-scan testing support in FLEX 6000 devices.	JTAG boundary-scan testing can be performed before configuration; however, it cannot be performed during or after configuration. During JTAG boundary-scan testing, <code>nCONFIG</code> must be held low.	JTAG boundary-scan testing can be performed before or after device configuration via the four JTAG pins ( <code>TDI</code> , <code>TDO</code> , <code>TMS</code> , and <code>TCK</code> ); however, it cannot be performed during configuration. When JTAG boundary-scan testing is performed before device configuration, <code>nCONFIG</code> must be held low.

## Device Configuration Pins

Table 22 summarizes the APEX 20K, FLEX 10K, and FLEX 6000 device configuration pins.

<b>Table 22. Pin Functions (Part 1 of 4)</b>					
<b>Pin Name</b>	<b>Device Family</b>	<b>User Mode</b>	<b>Configuration Scheme</b>	<b>Pin Type</b>	<b>Description</b>
MSEL0 MSEL1	APEX 20K FLEX 10K	–	All	Input	2-bit configuration input. Sets the APEX 20K and FLEX 10K device configuration scheme. See <a href="#">Table 2 on page 4</a> .
MSEL	FLEX 6000	–	All	Input	1-bit configuration input. Sets the FLEX 6000 device configuration scheme. See <a href="#">Table 3 on page 4</a> .
nSTATUS	APEX 20K FLEX 10K FLEX 6000	–	All	Bidirectional open-drain	The device drives nSTATUS low immediately after power-up and releases it within 5 $\mu$ s. (When using a configuration device, the configuration device holds nSTATUS low for up to 200 ms.) The nSTATUS pin must be pulled up to V <sub>CC</sub> with a 1-k $\Omega$ resistor. If an error occurs during configuration, nSTATUS is pulled low by the target device. If an external source drives the nSTATUS pin low during configuration or initialization, the target device enters an error state. Driving nSTATUS low after configuration and initialization does not affect the configured device. However, if a configuration device is used, driving nSTATUS low will cause that device to attempt to configure the APEX or FLEX device.
nCONFIG	APEX 20K FLEX 10K FLEX 6000	–	All	Input	Configuration control input. A low resets the target device; a low-to-high transition begins configuration.

**Table 22. Pin Functions (Part 2 of 4)**

Pin Name	Device Family	User Mode	Configuration Scheme	Pin Type	Description
CONF_DONE	APEX 20K FLEX 10K FLEX 6000	–	All	Bidirectional open-drain	<p>Status output. The target device drives the CONF_DONE pin low before and during configuration. Once all configuration data is received without error and the initialization clock cycle starts, the target device releases CONF_DONE.</p> <p>Status input. After all data is received and CONF_DONE goes high, the target device initializes and enters user mode.</p> <p>The CONF_DONE pin must be pulled to <math>V_{CC}</math> with a 1-k<math>\Omega</math> resistor. An external source can drive this pin low to delay the initialization process, except when configuring with a configuration device. Driving CONF_DONE low after configuration and initialization does not affect the configured device.</p>
DCLK	APEX 20K FLEX 10K FLEX 6000	–	Configuration device PS PPS	Input	Clock input used to clock data from an external source into the target device. In PSA or PPA schemes, DCLK should be held low to prevent this pin from floating.
nCE	APEX 20K FLEX 10K FLEX 6000	–	All	Input	Active-low chip enables. The nCE pin activates the device with a low signal to allow configuration and should be tied low for single device configuration. The nCE pin must be held low during configuration, initialization, and user mode.
nCEO	APEX 20K FLEX 10K FLEX 6000	– I/O	Multi-device	Output	Output that drives low when device configuration is complete. During multi-device configuration, this pin feeds a subsequent device's nCE pin.
nWS	APEX 20K FLEX 10K FLEX 6000	I/O I/O	PPA PSA	Input	Write strobe input. For APEX 20K and FLEX 10K devices, a low-to-high transition causes the device to latch a byte of data on the DATA[7..0] pins. For FLEX 6000 devices, a low-to-high transition causes the device to latch a bit of data on the DATA pin.

**Table 22. Pin Functions (Part 3 of 4)**

Pin Name	Device Family	User Mode	Configuration Scheme	Pin Type	Description
nRS	APEX 20K FLEX 10K	I/O	PPA	Input	Read strobe input. For APEX 20K and FLEX 10K devices, a low input directs the device to drive the RDYnBSY signal on the DATA7 pin. For FLEX 6000 devices, a low input directs the device to drive the RDYnBSY signal on the DATA pin. If the nRS pin is not used, it should be tied high.
	FLEX 6000	I/O	PSA		
RDYnBSY	APEX 20K FLEX 10K FLEX 6000	I/O	PPA PSA	Output	Ready output. A high output indicates that the target device is ready to accept another data byte. A low output indicates that the target device is not ready to receive another data byte.
nCS CS	APEX 20K FLEX 10K FLEX 6000	I/O	PPA PSA	Input	Chip-select inputs. A low on nCS and a high on CS select the target device for configuration. If only one chip-select input is used, the other must be tied to the active value (e.g., nCS can be tied to ground if CS is used). The nCS and CS pins must be held active during configuration and initialization.
CLKUSR	APEX 20K FLEX 10K FLEX 6000	I/O	All	Input	Optional user-supplied clock input. Synchronizes the initialization of one or more devices.
DATA	FLEX 6000	–	Configuration device PS PSA	Input	Data inputs. Bit-wide configuration data is presented to the FLEX 6000 device on the DATA pin. In the PSA configuration scheme, the DATA pin presents the RDYnBSY signal after the nRS signal has been strobed, which may be more convenient for microprocessors than using the RDYnBSY pin.
DATA[7..1]	APEX 20K FLEX 10K	I/O	PPS PPA	Inputs	Data inputs. Byte-wide configuration data is presented to the target device on DATA[7..0].
DATA0	APEX 20K FLEX 10K	–	Configuration device PS PPA PPS	Input	Data input. In serial configuration modes, bit-wide configuration data is presented to the target device on the DATA0 pin.

**Table 22. Pin Functions (Part 4 of 4)**

Pin Name	Device Family	User Mode	Configuration Scheme	Pin Type	Description
DATA7	APEX 20K FLEX 10K	I/O	PPA	Output	In the PPA configuration scheme, the DATA7 pin presents the RDYnBSY signal after the nRS signal has been strobed, which may be more convenient for microprocessors than using the RDYnBSY pin.
INIT_DONE	APEX 20K FLEX 10K FLEX 6000	I/O	All	Output open-drain	Status pin. Can be used to indicate when the device has initialized and is in user mode. The INIT_DONE pin drives low during configuration. Before and after configuration, the INIT_DONE pin is released and is pulled to V <sub>CC</sub> by an external pull-up resistor. Because INIT_DONE is tri-stated before configuration, it is pulled high by the external pull-up resistor. Thus, the monitoring circuitry must be able to detect a low-to-high transition. This option is set in the MAX+PLUS II or Quartus software.
DEV_OE	APEX 20K FLEX 10K FLEX 6000	I/O	All	Input	Optional pin that allows the user to override all tri-states on the device. When this pin is driven low, all I/Os are tri-stated; when this pin is driven high, all I/Os behave as programmed. This option is set in the MAX+PLUS II or Quartus software.
DEV_CLRn	APEX 20K FLEX 10K FLEX 6000	I/O	All	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. This option is set in the MAX+PLUS II or Quartus software.
TDI	APEX 20K FLEX 10K FLEX 6000	I/O or JTAG pins	All	Input	JTAG pins. When used as user I/O pins, JTAG pins must be kept stable before and during configuration. JTAG pin stability prevents accidental loading of JTAG instructions.
TDO				Output	
TMS				Input	
TCK				Input	

Table 23 describes EPC2, EPC1, and EPC1441 pin functions during APEX 20K, FLEX 10K, and FLEX 6000 device configuration.

**Table 23. EPC2, EPC1 & EPC1441 Pin Functions During APEX 20K, FLEX 10K & FLEX 6000 Configuration (Part 1 of 2)**

Pin Name	Pin Number			Pin Type	Description
	8-Pin PDIP (1)	20-Pin PLCC	32-Pin TQFP (2)		
DATA	1	2	31	Output	Serial data output. The DATA pin is tri-stated before configuration when the nCS pin is high, and after the configuration device finishes sending its configuration data. This operation is independent of the device's position in the cascade chain.
DCLK	2	4	2	I/O	DCLK is a clock output when configuring with a single configuration device or when the configuration device is the first device in a configuration device chain. DCLK is a clock input for subsequent configuration devices in a configuration device chain. Rising edges on DCLK increment the internal address counter and present the next bit of data to the DATA pin. The counter is incremented only if the OE input is held high, the nCS input is held low, and all configuration data has not been transferred to the target device. When configuring with the first EPC2 or EPC1 device in a configuration device chain or with a single EPC1441 device, the DCLK pin drives low after configuration is complete or when OE is low.
OE (3)	3	8	7	I/O Open-drain	Output enable (active high) and reset (active low). A low logic level resets the address counter. A high logic level enables DATA and permits the address counter to count. If this pin is low (reset) during configuration, the internal oscillator becomes inactive and DCLK drives low.
nCS (3)	4	9	10	Input	Chip select input (active low). A low input allows DCLK to increment the address counter and enables DATA to drive out. If EPC1 or EPC2 is reset with nCS low, the device initializes as the first device in a configuration chain. If the EPC1 or EPC2 device is reset with nCS high, the device initializes as the subsequent device in the chain.
nCASC (4)	6	12	15	Output	Cascade select output (active low). This output goes low when the address counter has reached its maximum value. In a chain of EPC1 or EPC2 devices, the nCASC pin of one device is connected to the nCS pin of the next device, which permits DCLK to clock data from the next EPC1 or EPC2 device in the chain.

**Table 23. EPC2, EPC1 & EPC1441 Pin Functions During APEX 20K, FLEX 10K & FLEX 6000 Configuration (Part 2 of 2)**

Pin Name	Pin Number			Pin Type	Description
	8-Pin PDIP (1)	20-Pin PLCC	32-Pin TQFP (2)		
nINIT_CONF (3), (5), (6)	–	13	16	Output Open-drain	Allows the INIT_CONF JTAG instruction to initiate configuration. This pin is connected to the nCONFIG pin of the FLEX or APEX device to initiate configuration from the EPC2 via a JTAG instruction. If a chain of EPC2 devices is used, only the first EPC2 has its nINIT_CONF pin tied to the FLEX device's nCONFIG pin. (6)
TDI (5)	–	11	13	Input	JTAG data input pin. Connect this pin to V <sub>CC</sub> if the JTAG circuitry is not used.
TDO (5)	–	1	28	Output	JTAG data output pin. Do not connect this pin if the JTAG circuitry is not used.
TMS (5)	–	19	25	Input	JTAG mode select pin. Connect this pin to V <sub>CC</sub> if the JTAG circuitry is not used.
TCK (5)	–	3	32	Input	JTAG clock pin. Connect this pin to ground if the JTAG circuitry is not used.
VCCSEL	–	5	3	Input	Mode select for V <sub>CC</sub> supply. VCCSEL must be connected to ground if the device uses a 5.0-V power supply (i.e., V <sub>CC</sub> = 5.0 V). VCCSEL must be connected to V <sub>CC</sub> if the device uses a 3.3-V power supply (i.e., V <sub>CC</sub> = 3.3 V).
VPPSEL	–	14	17	Input	Mode select for V <sub>PP</sub> . VPPSEL must be connected to ground if V <sub>PP</sub> uses a 5.0-V power supply (i.e., V <sub>PP</sub> = 5.0 V). VPPSEL must be connected to V <sub>CC</sub> if V <sub>PP</sub> uses a 3.3-V power supply (i.e., V <sub>PP</sub> = 3.3 V).
VPP	–	18	23	Power	Programming power pin. For the EPC2 device, this pin is normally tied to V <sub>CC</sub> . If the EPC2 V <sub>CC</sub> is 3.3 V, V <sub>PP</sub> can be tied to 5.0 V to improve in-system programming times. For EPC1 and EPC1441 devices, V <sub>PP</sub> must be tied to V <sub>CC</sub> .
VCC	7, 8	20	27	Power	Power pin.
GND	5	10	12	Ground	Ground pin. A 0.2-μF decoupling capacitor must be placed between the VCC and GND pins.

**Notes to table:**

- (1) This package is available for EPC1 and EPC1441 devices only.
- (2) This package is available for EPC2 and EPC1441 devices only.
- (3) The OE, nCS, and nINIT\_CONF pins on EPC2 devices have internal, user-configurable 1-kΩ pull-up resistors. If internal pull-up resistors are used, external pull-up resistors should not be used on these pins. This EPC2 feature is available in the MAX+PLUS II software version 9.2 and higher. In earlier versions of the MAX+PLUS II software, the internal pull-up resistors on these pins are disabled.
- (4) The EPC1441 device does not support data cascading. EPC2 and EPC1 devices support data cascading.
- (5) This pin applies to EPC2 devices only.
- (6) This instruction is supported by MAX+PLUS II software versions 9.2 and higher.

## Device Configuration Files

Altera's MAX+PLUS II and Quartus development tools can create one or more configuration and programming files to support the configuration schemes discussed in this application note. This section describes these files.

### SRAM Object File (.sof)

You should use an SOF during PS configuration when the data is downloaded directly from the Altera programming hardware with a MasterBlaster, ByteBlasterMV, or BitBlaster cable. For APEX 20K, FLEX 10K, and FLEX 6000 devices, the MAX+PLUS II or Quartus Compiler's Assembler module automatically creates the SOF for each device in your design. The MAX+PLUS II or Quartus software controls the configuration sequence and automatically inserts the appropriate headers into the configuration data stream. All other configuration files are created from the SOF.

### Programmer Object File (.pof)

A Programmer Object File (.pof) is used by the Altera programming hardware to program a configuration device. A POF is generated automatically when an APEX 20K, FLEX 10K, or FLEX 6000 project is compiled. For smaller FLEX devices (e.g., EPF10K20 devices), multiple POFs can fit into one configuration device; for larger devices (e.g., APEX 20K devices), multiple configuration devices are required to hold the configuration data.

### Raw Binary File (.rbf)

The Raw Binary File (.rbf) is a binary file, e.g., one byte of RBF data is 8 configured bits 10000101 (85 Hex), containing the APEX 20K, FLEX 10K, or FLEX 6000 configuration data. Data must be stored so that the least significant bit (LSB) of each data byte is loaded first. The converted image can be stored on a mass storage device. The microprocessor can then read data from the binary file and load it into the APEX 20K, FLEX 10K, or FLEX 6000 device. You can also use the microprocessor to perform real-time conversion during configuration. In the PPA and PPS configuration schemes, the target device receives its information in parallel from the data bus, a data port on the microprocessor, or some other byte-wide channel. In PS and PSA configuration schemes, the data is shifted in serially, LSB first.

The following steps explain how to create an RBF file for FLEX 10K or FLEX 6000 devices using the MAX+PLUS II software. You can follow a similar procedure in the Quartus software to generate RBFs for APEX 20K devices.



1. In the MAX+PLUS II Compiler or Programmer, choose the **Convert SRAM Object Files** command (File menu.)
2. In the **Convert SRAM Object Files** dialog box, specify which SOF files to combine and then select *.rbf (Sequential)* in the *File Format* box. Click **OK**.



For more information on creating RBFs, search for “RBF” in MAX+PLUS II or Quartus Help.

## Hexadecimal (Intel-Format) File (.hex)

A Hex File is an ASCII file in the Intel Hex format. This file is used by third-party programmers to program Altera’s serial configuration devices. Hex Files are also used to program parallel configuration devices with third-party programming hardware. You can use parallel configuration devices in the PPS, PPA, or PSA configuration schemes, in which a microprocessor uses the parallel configuration device as the data source.

The following steps explain how to create a Hex file for FLEX 10K or FLEX 6000 devices using the MAX+PLUS II software. You can follow a similar procedure in the Quartus software to generate Hex files for APEX 20K devices.


1. In the MAX+PLUS II Programmer or Compiler, choose the **Convert SRAM Object Files** command (File menu).
2. In the **Convert SRAM Object Files** dialog box, specify which SOF files to combine and then select *.hex* in the *File Format* box. Click **OK**.



For more information on creating Hex Files, search for “Hex File” in MAX+PLUS II or Quartus Help.

## Tabular Text File (.tff)

The Tabular Text File (.tff) is a tabular ASCII file that provides a comma-separated version of the configuration data for the PPA, PPS, PSA, and bit-wide PS configuration schemes. In some applications, the storage device containing the FLEX 10K or FLEX 6000 configuration data is neither dedicated to nor connected directly to the target device. For example, a configuration device can also contain executable code for a system (e.g., BIOS routines) and other data. The TTF allows you to include the configuration data as part of the microprocessor's source code using the `include` or `source` commands. The microprocessor can access this data from a configuration device or mass-storage device and load it into the target device. A TTF can be imported into nearly any assembly language or high-level language compiler.

 TTFs are supported by the MAX+PLUS II software only.

The following steps explain how to create a TTF file for FLEX 10K or FLEX 6000 devices using the MAX+PLUS II software.


1. In the MAX+PLUS II Programmer or Compiler, choose the **Convert SRAM Object Files** command (File menu).
2. In the **Convert SRAM Object Files** dialog box, specify which SOF files to combine and then select *.tff (Sequential)* in the *File Format* box. Click **OK**.



For more information on creating TTFs, search for "TTF" in MAX+PLUS II or Quartus Help.

## Serial Bitstream File (.sbf)

A Serial Bitstream File (.sbf) is used in PS schemes to configure FLEX 10K and FLEX 6000 devices in-system with the BitBlaster cable.

 SBFs are supported by the MAX+PLUS II software only.

The following steps explain how to create an SBF file for FLEX 10K or FLEX 6000 devices using the MAX+PLUS II software.

1. In the MAX+PLUS II Programmer or Compiler, choose the **Convert SRAM Object Files** command (File menu).
2. In the **Convert SRAM Object Files** dialog box, specify which SOF files to combine and then select *.sbf (Sequential)* in the *File Format* box. Click **OK**.



For more information on creating SBFs, search for “SBF” in MAX+PLUS II Help.

### Jam File (.jam)

A Jam File is an ASCII text file in the Jam device programming language that stores device programming information. These files are used to program, verify, and blank-check one or more devices in the MAX+PLUS II or Quartus Programmer or in an embedded processor-type environment.

### Jam Byte-Code File (.jbc)

A Jam Byte-Code File (.jbc) is a binary file of a Jam File in a byte-code representation. JBC files store device programming information used to program, verify, and blank-check one or more devices in the MAX+PLUS II Programmer or in an embedded processor-type environment.

You can configure APEX 20K, FLEX 10K, and FLEX 6000 devices using data stored in either a configuration device or the MAX+PLUS II or Quartus software.

## Device Configuration

### Configuration with a Configuration Device

You program configuration devices using the MAX+PLUS II or Quartus software, the Master Programming Unit (MPU), and the appropriate configuration device programming adapter. [Table 24](#) shows which programming adapter to use with each configuration device.

<i>Table 24. Programming Adapters</i>		
<b>Device</b>	<b>Package</b>	<b>Adapter</b>
EPC2	20-pin J-Lead	PLMJ1213
	32-pin TQFP	PLMT1213
EPC1	8-pin DIP	PLMJ1213
	20-pin J-Lead	PLMJ1213
EPC1441	8-pin DIP	PLMJ1213
	20-pin J-Lead	PLMJ1213
	32-pin TQFP	PLMT1064

The following steps explain how to program Altera configuration devices using the MAX+PLUS II software:

1. Open the MAX+PLUS II Programmer.
2. Load the appropriate POF using the **Select Programming File** dialog box (File menu). By default, the Programmer loads the current project's POF. The *Device* field displays the appropriate device for the current programming file.
3. Insert a blank configuration device into the programming adapter's socket.
4. Click the **Program** button.

After successful programming, you can place the configuration device on the PCB to configure an APEX 20K, FLEX 10K, or FLEX 6000 device.



For more information on configuration devices, see the [Configuration Devices for APEX & FLEX Devices Data Sheet](#).

### Configuration with the MasterBlaster, ByteBlasterMV, or BitBlaster Cable

For more information on the MasterBlaster, ByteBlasterMV, or BitBlaster cable, see the following documents:

- [MasterBlaster Serial/USB Communications Cable Data Sheet](#)
- [ByteBlasterMV Parallel Port Download Cable Data Sheet](#)
- [BitBlaster Serial Download Cable Data Sheet](#)

## Configuration Reliability

The APEX 20K, FLEX 10K, and FLEX 6000 architecture has been designed to minimize the effects of power supply and data noise in a system, and to ensure that the configuration data is not corrupted during configuration or normal user-mode operation. A number of circuit design features are provided to ensure the highest possible level of reliability from this SRAM technology.

Cyclic redundancy code (CRC) circuitry is used to validate each data frame (i.e., sequence of data bits) as it is loaded into the target device. If the CRC generated by the APEX 20K or FLEX device does not match the data stored in the data stream, the configuration process is halted, and the nSTATUS pin is pulled and held low to indicate an error condition. CRC circuitry ensures that noisy systems will not cause errors that yield an incorrect or incomplete configuration.

The APEX 20K, FLEX 10K, and FLEX 6000 architecture also provides a very high level of reliability in low-voltage brown-out conditions. These devices' SRAM cells require a certain  $V_{CC}$  level to maintain accurate data. This voltage threshold is significantly lower than the voltage required to activate the device's POR circuitry. Therefore, the target device stops operating if the  $V_{CC}$  starts to fail, and indicates an operation error by pulling and holding the  $nSTATUS$  pin low. The device must then be reconfigured before it can resume operation as a logic device. In active configuration schemes in which  $nCONFIG$  is tied to  $V_{CC}$ , reconfiguration begins as soon as  $V_{CC}$  returns to an acceptable level. The low pulse on  $nSTATUS$  resets the configuration device by driving OE low. In passive configuration schemes, the host system starts the reconfiguration process.

These device features ensure that APEX 20K, FLEX 10K, and FLEX 6000 devices have the highest possible reliability in a wide variety of environments, and provide the same high level of system reliability that exists in other Altera programmable logic devices (PLDs).

## Board Layout Tips

Even though the DCLK signal (used in configuration device, PS, and PPS configuration schemes) is fairly low-frequency, it drives edge-triggered pins on the APEX 20K, FLEX 10K, or FLEX 6000 device. Therefore, any overshoot, undershoot, ringing, or other noise can affect configuration. When designing the board, lay out the DCLK trace using the same techniques as laying out a clock line, including appropriate buffering. If more than five devices are used, Altera recommends using buffers to split the fan-out on the DCLK signal.

## Revision History

The information contained in *Application Note 116 (Configuring APEX 20K, FLEX 10K & FLEX 6000 Devices)* version 1.02 supersedes information published in previous versions.

### Version 1.02 Changes

Version 1.02 contains the following changes:

- Corrected the notes to [Figures 13, 20, and 24](#).
- Corrected [Figures 10, 24, and 31](#).
- Corrected a note to [Table 11](#).
- Added a note to [Table 18](#).

## Version 1.01 Changes

Version 1.01 contained the following changes:

- Added a note to [Table 1](#).
- Updated information in [Tables 4, 10, 11, 13, 14, 15, and 16](#).
- Corrected the minimum  $t_{CFG}$  value in [Tables 12 and 13](#).
- Corrected [Figures 1, 3, 4, 5, 6, 9, 10, 13, 16, 17, 20, 21, 24, 25, 29, and 30](#).
- Added a note to [Figures 26 and 27](#).



*Notes:*



101 Innovation Drive  
San Jose, CA 95134  
(408) 544-7000  
<http://www.altera.com>  
**Applications Hotline:**  
(800) 800-EPLD  
**Customer Marketing:**  
(408) 544-7104  
**Literature Services:**  
(888) 3-ALTERA  
[lit\\_req@altera.com](mailto:lit_req@altera.com)

Altera, APEX, APEX 20K, APEX 20KE, FLEX, FLEX 10K, FLEX 10KA, FLEX 10KE, FLEX 6000, BitBlaster, ByteBlaster, ByteBlasterMV, MasterBlaster, Jam, MAX, MAX+PLUS, MAX+PLUS II, and Quartus are trademarks and/or service marks of Altera Corporation in the United States and other countries. Altera acknowledges the trademarks of other organizations for their respective products or services mentioned in this document. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

Copyright © 1999 Altera Corporation. All rights reserved.



I.S. EN ISO 9001