Configuring FLEX 6000 Devices

November 1998, ver. 1.03 Application Note 87

Introduction

FLEX® 6000 devices can be configured using a variety of configuration schemes, which are ideal for a variety of systems. You can configure FLEX 6000 devices with either an EPC1 Configuration EPROM or a microprocessor. See Table 1.

This application note discusses how to configure one or more FLEX 6000 devices. FLEX 6000 device configuration is similar to FLEX 8000 and FLEX 10K device configuration. This application note should be used together with the *FLEX 6000 Programmable Logic Device Family Data Sheet* and the *Configuration EPROMs for FLEX Devices Data Sheet*.

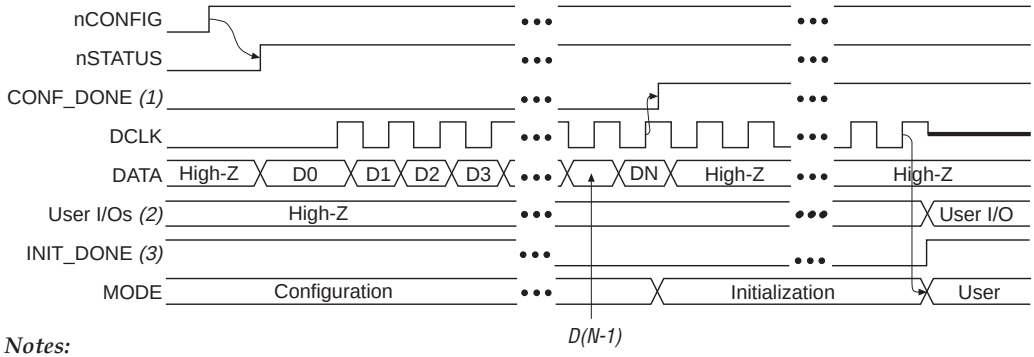
The following topics are discussed:

- Device configuration overview
- FLEX 6000 device configuration schemes
  - Configuration EPROM
  - Passive serial with the BitBlaster™ or the ByteBlaster™
  - Passive serial with a microprocessor
  - Passive serial asynchronous
- Device options
- Device configuration pins
- Device configuration files
- Device configuration
- Configuration reliability

Device Configuration Overview

During device operation, 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 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 configuration, initialization, and user modes. The arrows show which signal transitions are dependent on other transitions.

Figure 1. 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. 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.

The configuration data for a FLEX 6000 device can be loaded using an EPC1 Configuration EPROM configuration scheme or a passive configuration scheme:

- When configuring a FLEX 6000 device with an EPC1 Configuration EPROM, control and synchronization signals are generated by the FLEX 6000 device and the EPC1. The EPC1 provides the configuration data to the FLEX 6000 device.
- When configuring a FLEX 6000 device with a passive configuration scheme (i.e., configuration with a microprocessor), the FLEX 6000 device is incorporated into a system with an intelligent host that controls the configuration process. The host supplies configuration data from a storage device, e.g., a hard disk, RAM, or other system memory. With passive configuration, the functionality of the FLEX 6000 device can be changed while the system is in operation. Also, in-field upgrades are possible by distributing a new programming file to system users. Table 1 summarizes the configuration schemes.

Table 1. FLEX 6000 Configuration Schemes

Configuration Scheme	Typical Use
Configuration EPROM	Configuration with the EPC1 Configuration EPROM.
Passive serial	Configuration with a serial synchronous microprocessor interface, the BitBlaster, or the ByteBlaster.
Passive serial asynchronous	Configuration with a serial asynchronous microprocessor interface.

The configuration scheme is chosen by driving the FLEX 6000 device's MSEL pin either high or low, as shown in Table 2. The value of the MSEL pin can be changed between configurations to change the configuration mode. However, this pin is most commonly tied to VCC or GND.

Table 2. Configuration Schemes

Configuration Scheme	MSEL
Configuration EPROM or passive serial, including the BitBlaster or the ByteBlaster	0
Passive serial asynchronous	1

Device option bits and device configuration pins are discussed on pages 20 and 23, respectively.

Table 3 summarizes the approximate configuration file size required for each FLEX 6000 device. To calculate the amount of data storage space required for multi-device configurations, simply add the file size for each FLEX 6000 device in the design. Because different versions of MAX+PLUS® II may add a slightly different number of padding bits during programming, the exact file size may vary. However, for any specific version of MAX+PLUS II, any design in the same device will use the same configuration file size.

Table 3. FLEX 6000 Configuration File Sizes

Device	Data Size (Bits)	Data Size (Kbytes)
EPF6010, *Note (1)*	160,000	19.5
EPF6016, EPF6016A	260,000	31.7
EPF6024A, *Note (1)*	420,000	51.3

Note:
(1) Data size information for this device is preliminary.

One EPC1 Configuration EPROM is large enough to configure any FLEX 6000 device, and one or more EPC1 Configuration EPROMs can configure multiple FLEX 6000 devices.

FLEX 6000 Device Configuration Schemes

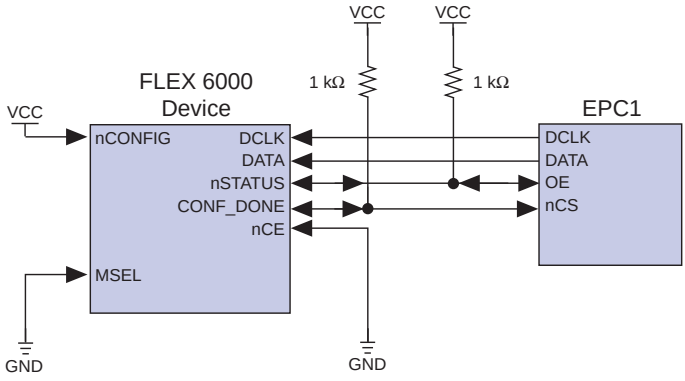
FLEX 6000 devices can be configured using the following configuration schemes. The passive serial scheme is divided into two different implementations.

- Configuration EPROM
- Passive serial with the BitBlaster or the ByteBlaster
- Passive serial with a microprocessor
- Passive serial asynchronous

Configuration EPROM Configuration

The Configuration EPROM scheme uses an Altera®-supplied serial EPC1 Configuration EPROM to supply data to the FLEX 6000 device in a serial bitstream. See Figure 2.

Figure 2. Configuration EPROM Scheme Circuit



In the Configuration EPROM scheme, nCONFIG is usually tied to VCC so that the system will automatically configure after power-up. Upon device power-up, the FLEX 6000 device recognizes the low-to-high transition on nCONFIG, which initiates configuration. The FLEX 6000 device drives the open-drain CONF_DONE pin low, which asserts the nCS pin on the EPC1 device. The open-drain nSTATUS pin is released by the FLEX 6000 device and the EPC1 as they exit power-on reset. The EPC1 has a power-on reset delay of 100 ms to allow the power supply to stabilize before beginning configuration; during this time the EPC1 drives its OE pin low. Because the OE pin is connected to nSTATUS on the FLEX 6000 device, driving the OE pin low delays configuration. When both devices have completed power-on reset, they release nSTATUS, which is pulled high by the pull-up resistor. When multiple EPC1 or FLEX 6000 devices are used, configuration will not begin until all devices have released their OE or nSTATUS pins.

The EPC1 then uses its internal oscillator to serially clock data from its EPROM cells to the FLEX 6000 device. At the end of a successful configuration, the FLEX 6000 device will release CONF_DONE to be pulled high by the pull-up resistor. The EPC1 will clock the FLEX 6000 device to initialize it and then the EPC1 will enter its power-down mode.

If an error occurs during configuration, the FLEX 6000 device drives the nSTATUS pin low, resetting the EPC1 and internally resetting the FLEX 6000 device. If the *Auto-Restart Configuration on Frame Error* option—available from the **Global Project Device Options** dialog box (Assign menu)—is turned on in MAX+PLUS II, the FLEX 6000 device in conjunction with the EPC1 device automatically reconfigures when an error occurs. After the FLEX 6000 device drives nSTATUS low to signify the error, it releases nSTATUS if the *Auto-Restart Configuration on Frame Error* option is set. Once nSTATUS goes high, the EPC1 resends the data to the FLEX 6000 device.

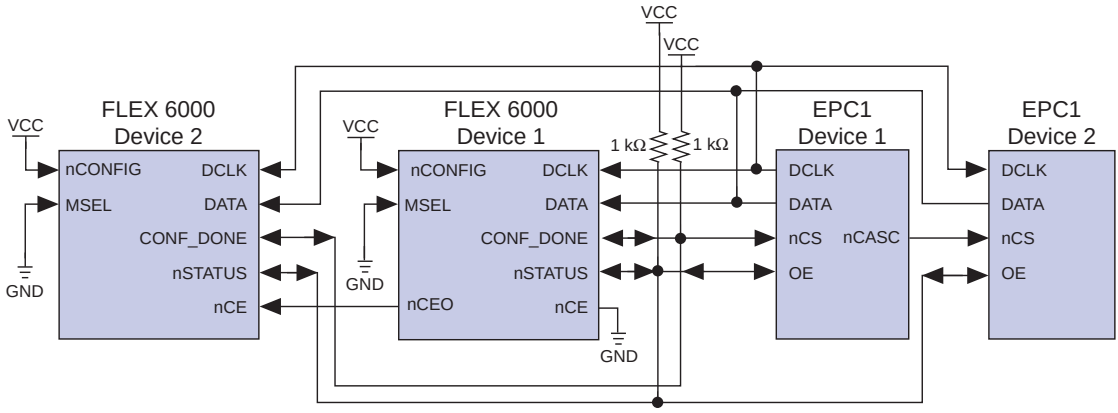
Additionally, if the EPC1 device sends all of its data and then detects that CONF_DONE has not gone high, the EPC1 will recognize that the FLEX 6000 device has not successfully configured. In this case, the EPC1 will pulse its OE pin low for a few microseconds, driving nSTATUS on the FLEX 6000 device low. If the *Auto-Restart Configuration on Frame Error* option is set, the FLEX 6000 device will then reset itself and pulse the nSTATUS pin low. When nSTATUS goes high again, the EPC1 will reconfigure the FLEX 6000 device. At the end of device configuration, the EPC1 device will drive DCLK low.

Driving CONF_DONE low after device configuration will cause the EPC1 to recognize that the FLEX 6000 device was not successfully configured; therefore, it is important to not use this scheme to delay initialization.

If the *Auto-Restart Configuration on Frame Error* option is turned off, the outside system must monitor nSTATUS and CONF_DONE to detect an error and then pulse nCONFIG low to restart configuration. The outside system can pulse nCONFIG if nCONFIG is under system control rather than tied to VCC. When configuration is complete, the FLEX 6000 device releases CONF_DONE, which disables the EPC1 by driving nCS high. The EPC1 device will drive DCLK low before and after configuration.

For multi-device configuration with Configuration EPROM devices, FLEX 6000 devices are cascaded. See Figure 3.

Figure 3. Configuration EPROM Scheme Multi-Device Configuration Circuit



After the first FLEX 6000 device has been configured, the nCEO pin on the first device activates the nCE pin on the second device, prompting the second device to begin configuration. The CONF_DONE pins on all the FLEX 6000 devices are tied together; therefore, the FLEX 6000 devices initialize and enter user mode at the same time.

Additionally, the nSTATUS lines are tied together; thus, if any device detects an error, the entire chain (including the EPC1 devices) will halt configuration. Also, if the first EPC1 device does not detect CONF_DONE going high at the end of configuration, it will reset the chain by pulsing its OE pin low for a few microseconds. This low pulse will drive OE low on other EPC1s and drive nSTATUS low on all FLEX 6000 devices, causing them to enter the error state, as if the FLEX 6000 devices were detecting an error. If the *Auto-Restart Configuration on Frame Error* option is set, the FLEX 6000 devices will release their nSTATUS pins after a reset time-out period. When the nSTATUS pins are released and pulled high, the EPC1 devices will reconfigure the chain. If the *Auto-Restart Configuration on Frame Error* option is not set, the FLEX 6000 devices will drive nSTATUS low until they are reset with a low pulse on nCONFIG.

If more than five FLEX 6000 devices are used, Altera recommends using buffers to split the fan-out on the DCLK signal.

EPC1 Configuration EPROMs can also be cascaded for configuration of several FLEX 6000 devices. When all the data from the first EPC1 device has been sent, the EPC1 drives nCASC low, which drives nCS on the next EPC1. Less than one clock cycle is required for one EPC1 device to activate the next EPC1 for configuration, so the stream of data is uninterrupted.

Figure 4 shows the timing waveform for the Configuration EPROM scheme where nCONFIG is tied to VCC.

Figure 4. Configuration EPROM Scheme Timing Waveform

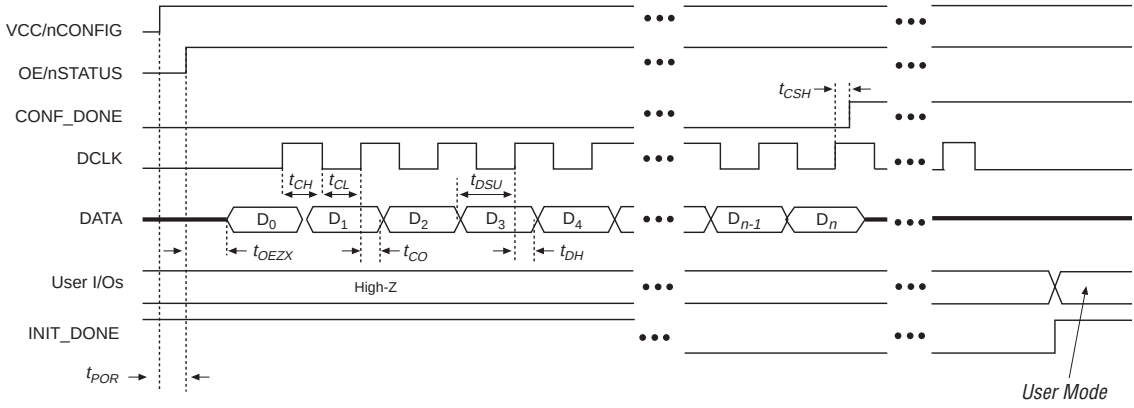


Table 4 defines the timing parameters for the Configuration EPROM scheme.

Table 4. Configuration EPROM Scheme Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{POR}	Power-on-reset delay, *Note (1)*		100	ms
t_{OEZX}	OE high to DATA output enabled		160	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
t_{CSH}	nCS/CONF_DONE low hold time after DCLK rising edge	0		ns
f_{MAX}	DCLK frequency	2	10	MHz

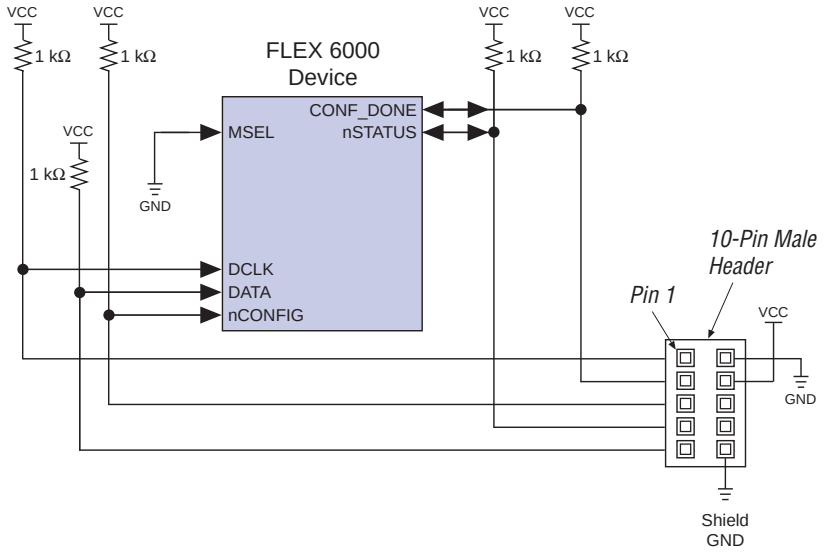
Note:

(1) Power-on-reset delay is imposed by the EPC1 device upon initial power-up to allow the voltage supply to stabilize. Subsequent reconfigurations will not incur this delay.

Passive Serial Configuration with the BitBlaster or ByteBlaster

In passive serial (PS) configuration, the BitBlaster or ByteBlaster generates a low-to-high transition on the nCONFIG pin to initiate configuration. The programming hardware then places the configuration data on the DATA pin of the FLEX 6000 device one bit at a time. The data is clocked into the FLEX 6000 device until CONF_DONE goes high. The programming hardware is used in its FLEX configuration mode, not in its multi-device JTAG configuration or programming mode. When using programming hardware, setting the *Auto-Restart Configuration on Frame Error* option does not affect the configuration cycle because MAX+PLUS II must restart configuration when an error occurs. See Figure 5.

Figure 5. PS Configuration Circuit with BitBlaster or ByteBlaster



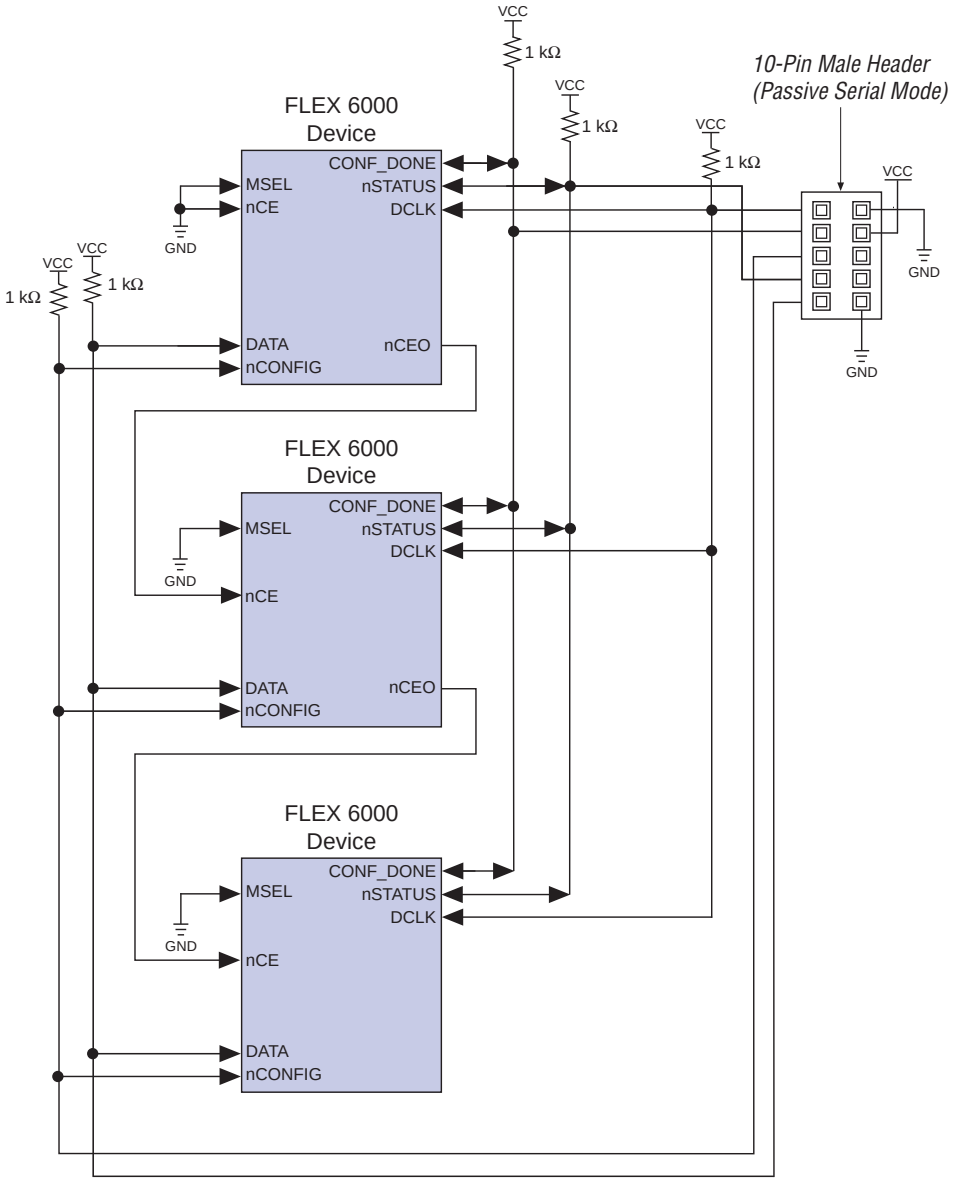
Multiple FLEX 6000 devices can be configured with programming hardware by connecting the nCEO pin of each device to the nCE pin of the subsequent device. The other configuration pins are connected to all FLEX 6000 devices in the chain. All FLEX 6000 device CONF_DONE pins are tied together, so that all FLEX 6000 devices can initialize and enter user mode at the same time.

Additionally, because the nSTATUS lines are tied together, if any device detects an error, the entire chain will stop configuration. In this situation, MAX+PLUS II must restart configuration; therefore, setting the *Auto-Restart Configuration on Frame Error* option has no affect on the configuration cycle.

If more than five FLEX 6000 devices are used, Altera recommends using buffers to split the fan-out on the DCLK signal.

Figure 6 shows the schematic for configuring multiple FLEX 6000 devices with programming hardware.

Figure 6. FLEX 6000 Multi-Device Configuration with BitBlaster or ByteBlaster



Configuration with a BitBlaster or ByteBlaster should not be attempted while a Configuration EPROM is connected to the FLEX 6000 device. If this operation must be performed, the EPC1 should be electrically isolated from the FLEX 6000 device and the BitBlaster or ByteBlaster via added logic.

For more information on how to use the BitBlaster, go to the *BitBlaster Serial Download Cable Data Sheet*. For more information on how to use the ByteBlaster, go to the *ByteBlaster Parallel Port Download Cable Data Sheet*.

Passive Serial with Microprocessor

In PS configuration, the microprocessor generates a low-to-high transition on the nCONFIG pin. The microprocessor or programming hardware then places the configuration data on the DATA pin of the FLEX 6000 device one bit at a time. The data is clocked into the FLEX 6000 device until CONF_DONE goes high. The microprocessor or programming hardware must first present the least significant bit (LSB) of each byte of data to the FLEX 6000 device. Data transfer should not begin until nSTATUS is released by the FLEX 6000 device.

After CONF_DONE goes high, DCLK must be clocked an additional 10 times to initialize the device. The configuration files created by MAX+PLUS II incorporate extra bits to accommodate the initialization. Driving DCLK to the device after configuration will not affect device operation. Therefore, sending the entire configuration file to the device is sufficient to configure and initialize it.

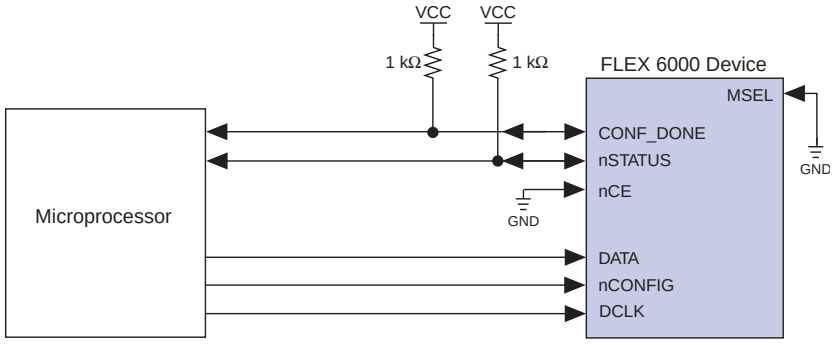
Handshaking signals are not used in PS configuration modes. Therefore, the configuration clock speed must be below 10 MHz to ensure correct configuration. There is no maximum DCLK period; configuration may be paused by halting DCLK.

If the FLEX 6000 device detects an error during configuration, it will drive its nSTATUS line 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, the FLEX 6000 device will release 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.

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

Figure 7 shows the configuration circuit for PS configuration with a microprocessor.

Figure 7. PS Configuration Circuit with a Microprocessor



For multi-device PS configuration, the nCEO pin on the first FLEX 6000 device is cascaded to the nCE pin of the next device. The second FLEX 6000 device begins configuration within one clock cycle; 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. To combine programming files, use the **Combine Programming Files** command (File menu) in the MAX+PLUS II Programmer.

Additionally, the nSTATUS lines are tied together; thus, if any device detects an error, the entire chain will halt configuration and drive 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, the FLEX 6000 devices will release nSTATUS after a reset time-out period. After nSTATUS is released, the microprocessor can reconfigure the FLEX 6000 devices. If more than five FLEX 6000 devices are used, Altera recommends using buffers to split the fan-out on the DCLK signal. See Figure 8.

Figure 8. PS Multi-Device Configuration Circuit with a Microprocessor

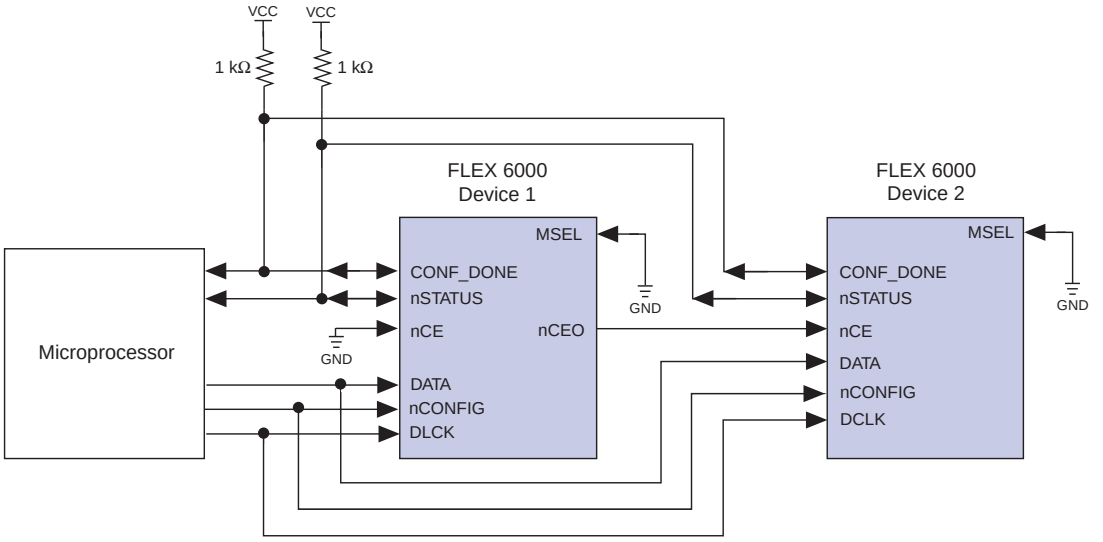
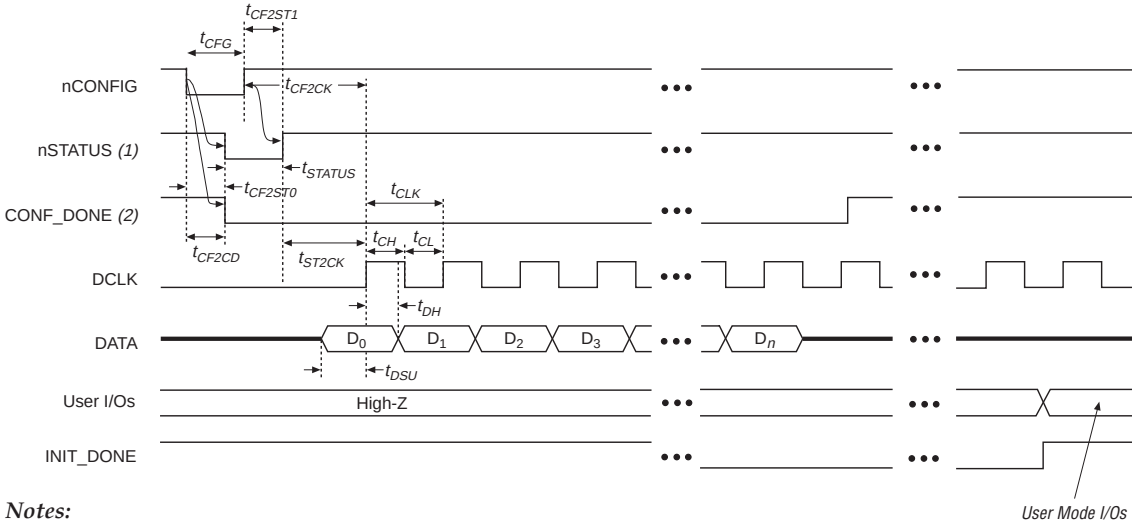


Figure 9 shows the timing waveform for PS configuration.

Figure 9. PS Timing Waveform



Notes:

(1) Upon power-up, nSTATUS is held low for five microseconds.

(2) Upon power-up and before configuration, CONF_DONE is low.

Table 5 defines the timing parameters for PS configuration.

Table 5. PS Timing Parameters

Symbol	Parameter	Min	Max	Units
t_{CF2CD}	nCONFIG low to CONF_DONE low		1	µs
t_{CF2ST0}	nCONFIG low to nSTATUS low		1	µs
t_{CF2ST1}	nCONFIG high to nSTATUS high		4	µs
t_{CFG}	nCONFIG low pulse width	2		µs
t_{STATUS}	nSTATUS low pulse width	2.5		µs
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	5		µs
t_{ST2CK}	nSTATUS high to first rising edge on DCLK	1		µs
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_{CH}	DCLK high time	50		ns
t_{CL}	DCLK low time	50		ns
t_{CLK}	DCLK period	100		ns
f_{MAX}	DCLK maximum frequency		10	MHz

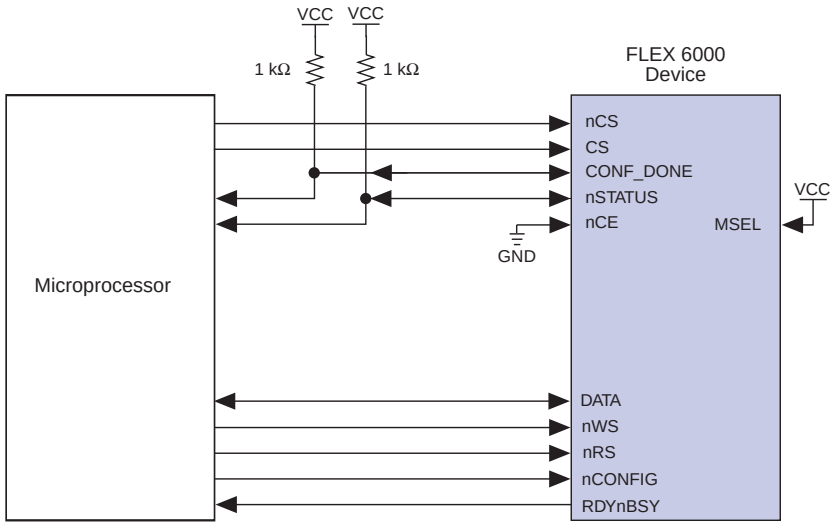
Passive Serial Asynchronous Configuration

In passive serial asynchronous (PSA) configuration, configuration is controlled by a microprocessor. To begin configuration, the microprocessor drives nCONFIG high. The microprocessor then asserts the nCS and CS inputs to the FLEX 6000 device. The microprocessor places a configuration bit on the DATA input of the FLEX 6000 device and pulses nWS low on the FLEX 6000 device. On the rising edge of the low pulse on nWS, the FLEX 6000 device latches a bit of configuration data. The FLEX 6000 device then drives the RDYnBSY signal low, indicating that it is processing the bit of configuration data. The microprocessor can then perform other system functions while the FLEX 6000 device is processing the bit of data.

Next, the microprocessor checks nSTATUS and CONF_DONE. If nSTATUS is asserted, the FLEX 6000 device is signaling an error. At this point the microprocessor should restart the configuration. However, if CONF_DONE is asserted the FLEX 6000 device is ready for initialization because all the configuration data has been received. If both nSTATUS and CONF_DONE are not asserted, the microprocessor sends the next data bit.

The nCS or CS pins on the FLEX 6000 device must remain asserted until configuration and initialization are complete. Figure 10 shows the PSA configuration circuit.

Figure 10. PSA Configuration Circuit



Optionally, an address decoder controls nCS and CS on the FLEX 6000 device. This decoder allows the microprocessor to select the FLEX 6000 device by accessing a particular address, simplifying the configuration process. The decoder is optional; the microprocessor may directly control the nCS and CS signals. Either of the nCS or CS signals may be tied to its active state (i.e., nCS may be tied low) and then the other signal can be toggled to control configuration.

The FLEX 6000 device can internally process data without the microprocessor. When the FLEX 6000 device is ready for the next bit of configuration data, it asserts RDYnBSY. When the microprocessor detects that the RDYnBSY signal is asserted, it strobes the next bit of configuration data into the FLEX 6000 device. Alternatively, the nRS signal can be strobed, 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, reading the state of the configuration data by strobing nRS permits the system to save an I/O port. Data should not be driven onto DATA while nRS is low because this will cause contention. If the nRS pin is not used to monitor configuration, it should be tied high.

After configuration, nCS, CS, nRS, nWS, and RDYnBSY are user I/Os. However, when the PSA mode is chosen in MAX+PLUS II, 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 will drive 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, the FLEX 6000 device will release 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.

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 CONF_DONE is not asserted, 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 PS mode. However, the FLEX 6000 device can initialize itself in PSA mode. Therefore, the FLEX 6000 device will assert CONF_DONE and the device will initialize before all data has been sent. The microprocessor can stop sending configuration data when CONF_DONE is asserted.

To combine programming files, use the **Combine Programming Files** command (File menu) in the MAX+PLUS II Programmer.

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, which asserts the nCE pin on the second FLEX 6000 device, 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 11.

Figure 11. PSA Multi-Device Configuration Circuit

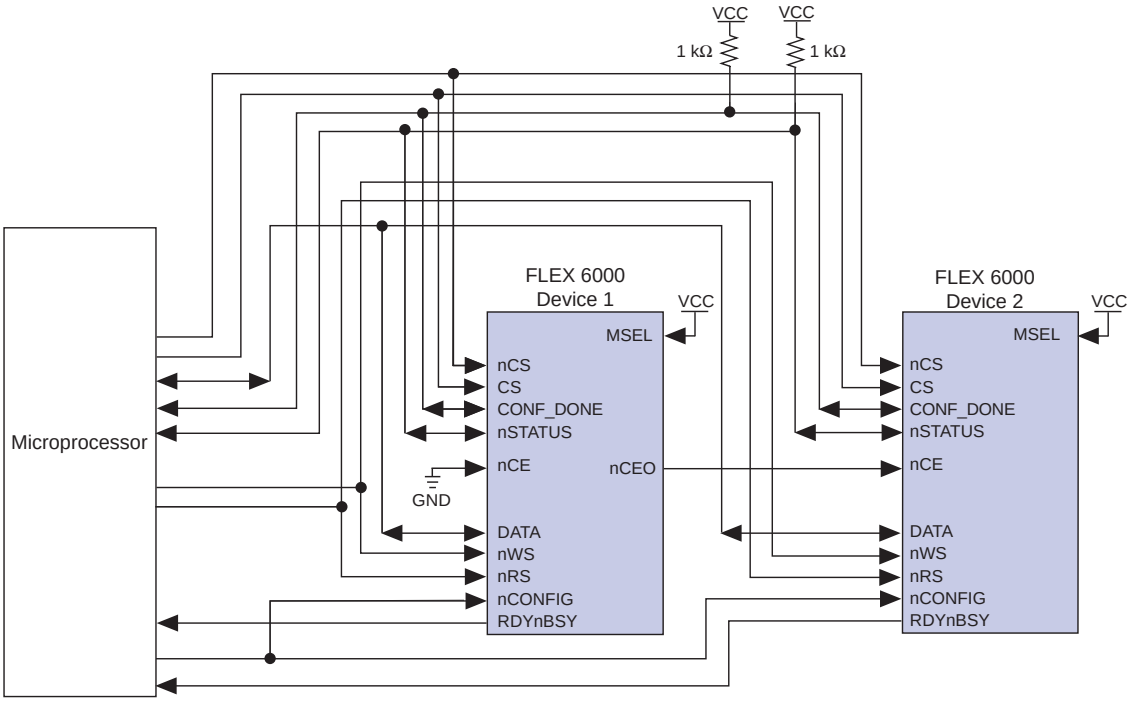
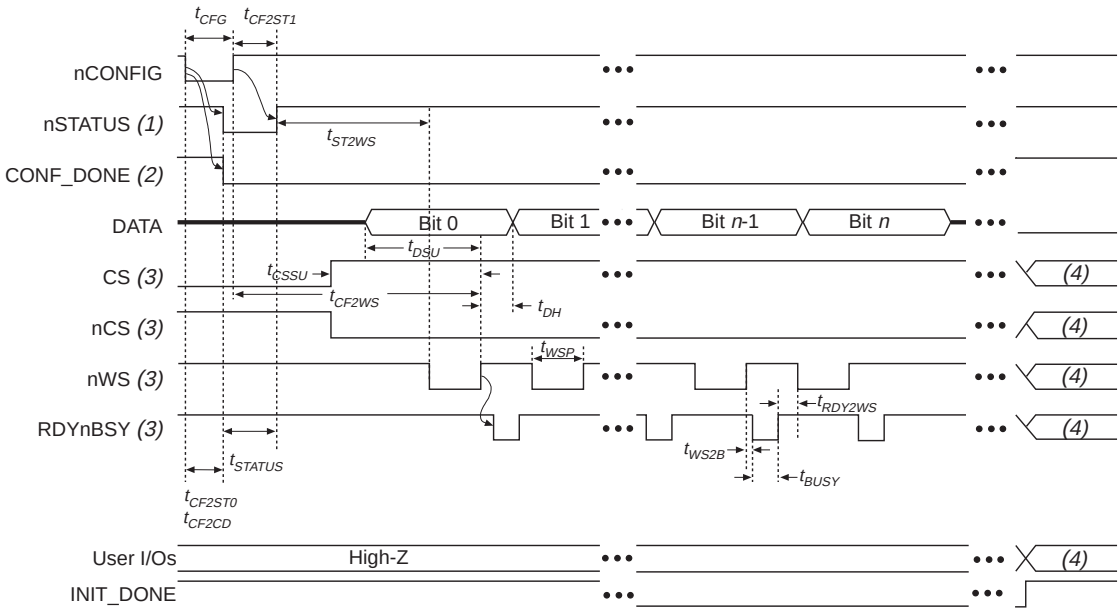


Figure 12 shows the timing waveform for PSA configuration.

Figure 12. PSA Timing Waveform

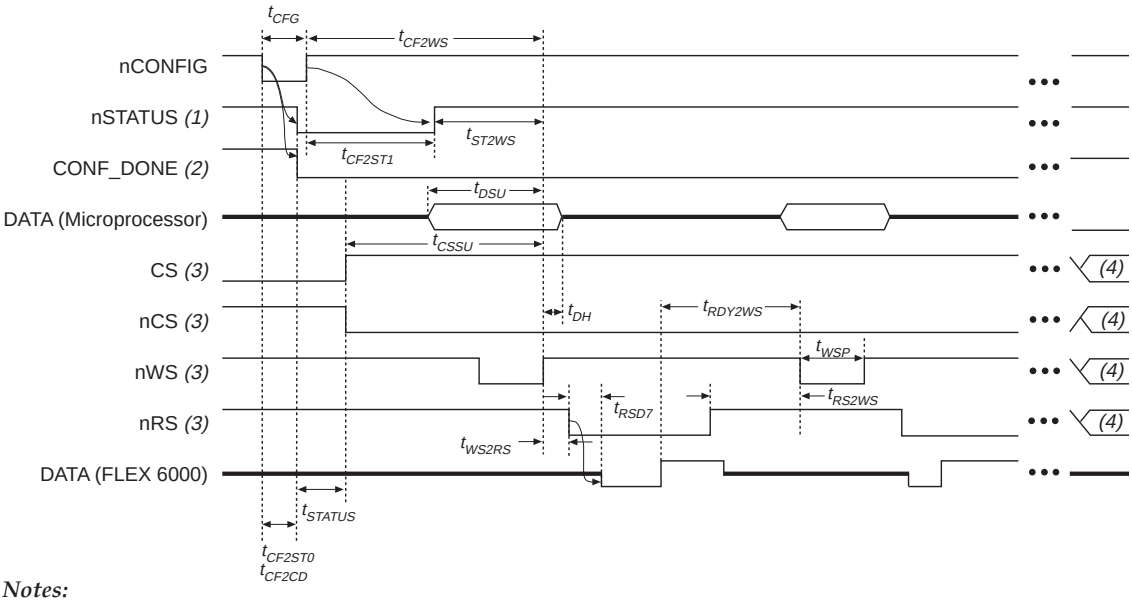


Notes:

(1) Upon power-up, nSTATUS is held low for five microseconds.
(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 FLEX 6000 device.
(4) Device I/O pins are in user mode.

Figure 13 shows the timing waveform when using a strobed nRS signal.

Figure 13. PSA Timing Waveform Using nRS and nWS



Notes:

(1) Upon power-up, nSTATUS is held low for five microseconds.
(2) Upon power-up, CONF_DONE is low.
(3) After configuration, the state of CS, nCS, nWS, and nRS depends on the design programmed into the FLEX 6000 device.
(4) Device I/O pins are in user mode.

Table 6 summarizes the timing parameters for PSA configuration.

Table 6. PSA Timing Parameters (Part 1 of 2)

Symbol	Parameter	Min	Max	Units
t_{CFG}	nCONFIG low pulse width	2		µs
t_{STATUS}	nSTATUS low pulse width	2.5		µs
t_{CF2ST1}	nCONFIG high to nSTATUS high		4	µs
t_{ST2WS}	nSTATUS high to first rising edge on nWS	1		µs
t_{CF2WS}	nCONFIG high to first rising edge on nWS	5		µs
t_{DSU}	Data setup time before rising edge on nWS	50		ns
t_{DH}	Data hold time after rising edge on nWS	0		ns
t_{CSSU}	Chip select setup time before rising edge on nWS	50		ns
t_{WSP}	nWS low pulse width	50		ns

Table 6. PSA Timing Parameters (Part 2 of 2)

Symbol	Parameter	Min	Max	Units
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	µs
t_{CF2ST0}	nCONFIG low to nSTATUS low		1	µs

Device Options

You can set FLEX 6000 device operation options in Altera's MAX+PLUS II development software with the **Global Project Device Options** command (Assign menu). Table 7 summarizes each of these options.

Table 7. FLEX 6000 Configuration Option Bits (Part 1 of 3)

Device Option	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
User-Supplied Start-Up Clock	The FLEX 6000 device must be clocked 10 times after it is configured to initialize the device. The user can choose the clock source.	In the PSA configuration scheme, the internal FLEX 6000 oscillator supplies the initialization clock. In Configuration EPROM and PS configuration, the internal oscillator is disabled. Therefore, external circuitry must provide the initialization clock on the DCLK pin. In the Configuration EPROM scheme, the EPC1 supplies the clock; in PS configuration, the microprocessor supplies the clock or programming hardware.	The user provides the clock on the CLKUSR pin. This clock can be used to synchronize the initialization of multiple FLEX 6000 devices. The clock should be supplied after CONF_DONE goes high. Supplying the clock during configuration will not affect the cycle. In user mode, the operation of CLKUSR is controlled by MAX+PLUS II.

Table 7. FLEX 6000 Configuration Option Bits (Part 2 of 3)

Device Option	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
Release Clears before Tri-States	During configuration, the I/O pins on the device are tri-stated. The user can choose the order in which to release the tri-states and clear the registers during initialization.	This option directs the device to release the tri-states on the I/O pins of the device before releasing the Clear signal on the device's registers.	This option directs the device to release the Clear signals on its registers before releasing the tri-states. This option may be used 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 every register on the FLEX 6000 device.	Chip-Wide Reset is not enabled. The DEV_CLRn pin is available as a user I/O pin.	Chip-Wide Reset is enabled for all registers within the device. All registers are cleared when the DEV_CLRn pin is driven low.
Enable Chip-Wide Output Enable	Enables a single pin to control all of the tri-states on a FLEX 6000 device.	Chip-Wide Output Enable is not enabled. The DEV_OE pin is available as a user I/O pin.	Chip-Wide Output Enable is enabled for all I/O pins on the FLEX 6000 device. After configuration, all user I/O pins are tri-stated when DEV_OE is low.
Enable INIT_DONE Output	Enables a pin to drive out a signal when the initialization process is complete and the device has entered user mode.	The INIT_DONE signal is not available. The INIT_DONE pin is available as a user I/O pin.	The INIT_DONE signal is available on the open-drain INIT_DONE pin. This pin drives low during configuration. After initialization, it is released and pulled high externally. The INIT_DONE pin must be connected to a 1-kΩ pull-up resistor.

Table 7. FLEX 6000 Configuration Option Bits (Part 3 of 3)

Device Option	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
Auto-Restart Configuration on Frame Error	If a data error occurs during FLEX 6000 device configuration, the user can choose how to restart the configuration.	The configuration process stops until the user directs the device to restart configuration. nSTATUS is driven low when there is an error. When nCONFIG is pulled low and then high, the device begins to reconfigure.	The configuration process will automatically restart. The nSTATUS pin drives low and then releases. The nSTATUS pin is then pulled to V_{CC} by the pull-up resistor, indicating that the configuration process can begin. In the Configuration EPROM scheme, the nSTATUS reset pulse automatically resets the EPC1 Configuration EPROM if the nSTATUS pin on the FLEX 6000 device is tied to OE on the Configuration EPROM. The EPC1 will begin reconfiguration and its OE pin goes high. If an error occurs during passive configuration, the device can be reconfigured without the system having to pulse nCONFIG. After nSTATUS goes high, reconfiguration can begin.
Enable JTAG Support	Enables post-configuration JTAG boundary-scan testing support in FLEX 6000 devices.	JTAG boundary-scan testing can be performed before configuration; however, it can not be performed during or after configuration. During JTAG boundary-scan testing, nCONFIG must be held low.	JTAG boundary-scan testing can be performed before or after device configuration via the four JTAG pins (TDI, TDO, TMS, TCLK); however, it cannot be performed during configuration. When JTAG boundary-scan testing is performed before device configuration, nCONFIG must be held low.

Device Configuration Pins

Table 8 summarizes the FLEX 6000 configuration pins.

Table 8. Pin Functions (Part 1 of 3)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
MSEL	–	All	Input	1-bit configuration input. Signals the configuration scheme to the FLEX 6000 device. These bits are set as shown in Table 2 on page 3.
nSTATUS	–	All	Bidirectional open drain	The FLEX 6000 device drives nSTATUS low immediately after power-up and releases it within 5 µs. (However, when configuring with a Configuration EPROM, the pin will be held low by the EPROM device for up to 100 ms.) The nSTATUS pin must be pulled up to V_{CC} with a 1-kΩ resistor. If an error occurs during configuration, nSTATUS is pulled low by the FLEX 6000 device. If an external source drives the nSTATUS pin low during configuration or initialization (as in multi-device configuration), the FLEX 6000 device enters an error state. Driving nSTATUS low after configuration and initialization does not affect the cycle.
nCONFIG	–	All	Input	Configuration control input. A low resets the FLEX 6000 device. A low-to-high transition begins configuration.
CONF_DONE	–	All	Bidirectional open drain	Status output. The CONF_DONE pin is driven low by the FLEX 6000 device before and during configuration. After all configuration data has been received without errors, the FLEX 6000 device releases CONF_DONE. Status input. A high on this input after all data has been received directs the FLEX 6000 device to initialize and enter user mode. The CONF_DONE net must be pulled to V_{CC} with a 1-kΩ resistor and may be driven low by an external source to delay the initialization process, except when configuring with an EPC1. Driving CONF_DONE low after configuration and initialization does not affect the cycle.
DCLK	–	Configuration EPROM, PS	Input	Clock input used to clock data from an external source into the FLEX 6000 device. When PSA is used, DCLK should be held low.

Table 8. Pin Functions (Part 2 of 3)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nCE	–	All	Input	Active-low chip enable. 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 and initialization.
nCEO	I/O	Multi-device	Output	Output that drives low when FLEX 6000 configuration is complete. This pin feeds the nCE of another FLEX 6000 device in a multi-device, cascaded configuration scheme.
nWS	I/O	PSA	Input	Write strobe input. A low-to-high transition causes the FLEX 6000 device to latch a byte of data on the DATA pin.
nRS	I/O	PSA	Input	Read strobe input. A low input directs the FLEX 6000 device to drive the RDYnBSY signal on the DATA pin. If the nRS pin is not used, it should be tied high.
RDYnBSY	I/O	PSA	Output	Ready output. A high output indicates that the FLEX 6000 device is ready to accept another bit of data. A low output indicates that the FLEX 6000 device is not ready to receive another bit of data.
nCS CS	I/O	PSA	Inputs	Chip-select inputs. A low on nCS and a high on CS selects the FLEX 6000 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	I/O	All	Input	Optional user-supplied clock input. Synchronizes initialization of one or more FLEX 6000 devices.
DATA	–	Configuration EPROM, 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.
INIT_DONE	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 will drive low during configuration. Before and after configuration, the INIT_DONE pin is released and is pulled to V_{CC} by an external pull-up resistor. Because INIT_DONE is tri-stated before configuration, it will be pulled high by the external pull-up resistor. Therefore, it is important that the monitoring circuitry be able to detect a low-to-high transition. This option is set in MAX+PLUS II. The INIT_DONE pin must be pulled up with a 1-kΩ resistor.

Table 8. Pin Functions (Part 3 of 3)

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
DEV_OE	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 defined in the user design. This option is set in MAX+PLUS II.
TDI	I/O or JTAG pins	All	Input	JTAG pins. When used as user I/Os, JTAG pins must be kept stable before and during configuration. JTAG pin stability prevents accidental loading of JTAG instructions.
TDO		All	Output	
TMS		All	Input	
TCK		All	Input	
DEV_CLRn	I/O	All	Input	Optional pin that allows the user to override all clears on all registers on the device. When this pin is driven low, all registers are cleared; when this pin is driven high, all registers behave as defined in the user design. This option is set in MAX+PLUS II.

Device Configuration Files

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

SRAM Object File (.sof)

An SOF is used in PS configuration when the data is downloaded with the BitBlaster or ByteBlaster. The MAX+PLUS II Compiler's Assembler module automatically creates the SOF for each FLEX 6000 device in your design. All other configuration files are created from the SOF.

Programming Object File (.pof)

A POF is used by the Altera or third-party programming hardware to program an EPC1 Configuration EPROM, which is used to configure FLEX 6000 devices in Configuration EPROM mode. A POF is automatically generated when each FLEX 6000 project is compiled. Each EPC1 requires a POF for programming. A POF can be generated from multiple SOFs for multi-device configuration.

Serial Bitstream File (.sbf)

An SBF is used to configure FLEX 6000 devices in-system in PS mode with the BitBlaster serial download cable. For information on how to use the BitBlaster, refer to the *BitBlaster Serial Download Cable Data Sheet*. For information on creating SBFs, search for "SBF" in MAX+PLUS II Help.

Hexadecimal (Intel-Format) File (.hex)

A Hex File is an ASCII file in the Intel Hex format. Hex Files are used to program parallel EPROMs with third-party programming hardware. You can use parallel EPROMs in the PS and PSA configuration schemes, in which a microprocessor uses the parallel EPROM as the data source. For information on creating Hex Files, search for "Hex File" in MAX+PLUS II Help.

Tabular Text File (.ttf)

The TTF is a tabular ASCII file that provides a comma-separated version of the configuration data for the PS and PSA configuration schemes. In some applications, the storage device that contains the FLEX 6000 configuration data is neither dedicated to nor connected directly to the FLEX 6000 device. For example, an EPROM can also contain executable code for a system (e.g., BIOS routines) and other data. The TTF allows you to include the FLEX 6000 configuration data as part of the source code for the microprocessor using "include" or "source" commands. The microprocessor can access this data from an EPROM or a mass-storage device and load it into the FLEX 6000 device.

A TTF can be imported into nearly any assembly language or high-level language compiler. Consult the documentation for your compiler or assembler for information on including other source files. For information on creating TTFs, search for "TTF" in MAX+PLUS II Help.

Raw Binary File (.rbf)

The RBF is a binary file containing the FLEX 6000 configuration data. Data must be stored so that the least significant bit (LSB) of each byte of data is loaded first. The converted image can be stored on a mass storage device. The microprocessor can then read data from the RBF and load it into the FLEX 6000 device. You can also use the microprocessor to perform real-time conversion during configuration. In the PS and PSA configuration schemes, the data is shifted in serially, LSB first. For information on creating RBFs, search for "RBF" in MAX+PLUS II Help.

Device Configuration

You can configure a FLEX 6000 device with data from an EPC1 Configuration EPROM, or you can download data into the FLEX 6000 device with the MAX+PLUS II software.

Configuration with a Configuration EPROM

You can program the EPC1 Configuration EPROM with data for FLEX 6000 device configuration using MAX+PLUS II, the Master Programming Unit (MPU), and the appropriate Configuration EPROM programming adapter. The PLMJ1213 adapter programs EPC1 Configuration EPROMs in 8-pin plastic dual in-line packages (PDIP) and 20-pin plastic J-lead chip carrier (PLCC) packages.

To program an Altera EPC1 Configuration EPROM:

1. Choose **Programmer** (MAX+PLUS II menu) to open the Programmer window.
2. By default, the Programmer loads the POF for the current project. If necessary, load a different POF with the **Select Programming File** command (File menu). The appropriate device for the current programming file is displayed in the *Device* field.
3. Insert a blank Configuration EPROM into the 8-pin DIP or 20-pin J-lead socket on the programming adapter.
4. Choose the **Program** button.

After successful programming, you can place the EPC1 Configuration EPROM on the target board to configure a FLEX 6000 device in the Configuration EPROM scheme.

For more information on the EPC1, refer to the *Configuration EPROMs for FLEX Devices Data Sheet*.

Configuration with MAX+PLUS II & the BitBlaster

For instructions on how to configure FLEX devices with the BitBlaster serial download cable, refer to the *BitBlaster Serial Download Cable Data Sheet*.

Configuration with MAX+PLUS II & the ByteBlaster

For instructions on how to configure FLEX devices with the ByteBlaster parallel port download cable, refer to the *ByteBlaster Parallel Port Download Cable Data Sheet*.

Configuration Reliability

The 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 process.

Cyclic redundancy code (CRC) circuitry is used to validate every data frame (i.e., sequence of data bits) as it is loaded into the FLEX 6000 device. If the CRC generated by the FLEX 6000 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. If the *Auto Reconfigure on Error* option bit is set, nSTATUS will be released and the device will be ready for reconfiguration; an EPC1 Configuration EPROM will then automatically reconfigure the device. CRC circuitry ensures that noisy systems will not cause errors that yield an incorrect or incomplete configuration.

The FLEX 6000 architecture also provides a very high level of reliability in low-voltage brown-out conditions. The FLEX 6000 device's 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 power-on reset (POR) circuitry in the FLEX 6000 device. Therefore, the FLEX 6000 device stops operating if the V_{CC} starts to fail, and indicates an operation error by pulling and holding the nSTATUS pin low. As stated above, if the *Auto Reconfigure on Error* option bit is set, the device will prepare itself for reconfiguration. The device must then be reconfigured before it can resume operation as a logic device. In EPC1 Configuration EPROM configuration schemes, reconfiguration begins as soon as V_{CC} returns to an acceptable level provided the nCONFIG pin is tied to V_{CC}. The low pulse on nSTATUS resets the EPROM by driving OE low. In passive configuration schemes, the host system starts the reconfiguration process.

These device features ensure that 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.

Revision History

The information contained in *Application Note 87 (Configuring FLEX 6000 Devices)* version 1.03 supersedes information published in previous versions.

Version 1.03 Changes

Application Note 87 (Configuring FLEX 6000 Devices) version 1.03 contains the following changes:

- Corrected the *Auto-Restart Configuration on Frame Error* option's default configuration and modified configuration definitions in Table 7.
- Made minor textual and style changes throughout the document.

Version 1.02 Changes

Application Note 87 (Configuring FLEX 6000 Devices) version 1.02 contained the following changes:

- Changed the t_{ST2CK} timing parameter from maximum to minimum in Table 5.
- Changed the nCEO pin's user mode to I/O in Table 8.

Version 1.01 Changes

Application Note 87 (Configuring FLEX 6000 Devices) version 1.01 contained the following changes:

- Corrected the t_{ST2WS} timing parameter in Table 6.
- Changed the TMS and TCK JTAG pins in Table 8.

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