Configuring FLEX 8000 Devices

June 2000, ver. 3.03 Application Note 33

Introduction

The architecture of Altera's Flexible Logic Element MatriX (FLEX) devices supports several different configuration schemes for loading a design into a single FLEX 8000 device on the circuit board. This application note provides complete details on all aspects of configuring individual FLEX 8000 devices, including sample schematics and timing information.

This application note should be used together with the current *FLEX 8000 Programmable Logic Device Family* and *Configuration EPROMs for FLEX 8000 Devices* data sheets. For information on configuring multiple FLEX 8000 devices in a system, refer to *Application Note 38 (Configuring Multiple FLEX 8000 Devices)* in this handbook. If appropriate, illustrations in this application note show devices with generic "FLEX 8000" and "Configuration EPROM" labels to indicate that they are valid for all FLEX 8000 devices and Altera Configuration EPROMs. All timing parameters shown in figures and tables apply to all FLEX 8000 device speed grades.

The following topics are discussed:

FLEX 8000 Device Operating Modes

The FLEX 8000 architecture uses SRAM cells to store the configuration data for the device. These SRAM cells must be loaded each time the circuit powers up and begins operation. The process of physically loading the SRAM programming data into the FLEX 8000 device is called *configuration*. After configuration, the FLEX 8000 device resets its registers, enables its I/O pins, and begins operation as a logic device. This reset operation is called *initialization*. Together, the configuration and initialization processes

are called *command mode*; normal in-circuit device operation is called *user mode*.

SRAM technology allows FLEX 8000 devices to be reconfigured in-circuit by loading new configuration data. Real-time reconfiguration can be performed by forcing the device into command mode with a dedicated device pin, loading different configuration data, reinitializing the device, and resuming user-mode operation. The entire process requires less than 100 ms, and can be used to dynamically reconfigure FLEX 8000 devices during system operation.

You can update existing systems that incorporate FLEX 8000 devices by installing new data in the system. Such in-field upgrades can be as simple as copying a new configuration file to a hard disk or inserting an EPROM programmed with new configuration data into the circuit.

Device configuration can occur either automatically at system power-up or under the control of external logic. Initialization can be controlled by the internal oscillator in the FLEX 8000 device or by an external Clock signal. Dedicated device configuration pins can be used to control when configuration and initialization begin. This range of command-mode control features provides excellent flexibility for designs implemented in FLEX 8000 devices.

Overview of Configuration Schemes

The configuration data for a FLEX 8000 device can be loaded with one of six configuration schemes, which you choose on the basis of the target application. Both active and passive schemes are available. In an active configuration scheme, the FLEX 8000 device guides the configuration operation, controlling external memory devices and the initialization process. The Clock source for all active configuration schemes is an internal oscillator in the FLEX 8000 device that typically operates in the range of 2 to 6 MHz. In a passive configuration scheme, an external controller guides the configuration of the FLEX 8000 device, which operates as a slave. Table 1 shows the source of data for each of the six configuration schemes.

Table 1. Configuration Schemes

Configuration Scheme	Acronym	Data Source
Active serial	AS	Altera Configuration EPROM
Active parallel up	APU	Parallel EPROM
Active parallel down	APD	Parallel EPROM
Passive serial	PS	Serial data path
Passive parallel synchronous	PPS	Intelligent host
Passive parallel asynchronous	PPA	Intelligent host

Each FLEX 8000 device has a different size requirement for its configuration data, based on the number of SRAM cells in the device. Table 2 shows the approximate size of data, expressed in both bits and Kbytes, necessary to configure each FLEX 8000 device. You can use this table to calculate the data space (i.e., data storage resources) required in a parallel or serial data source for a system that incorporates FLEX 8000 devices.

Table 2. FLEX 8000 Device Data-Size Spaces

Device	Data Size (bits)	Data Size (Kbytes)
EPF8282, EPF8282V	40,000	5
EPF8452	64,000	8
EPF8636	96,000	12
EPF8820	128,000	16
EPF81188	192,000	24
EPF81500	250,000	31

Active Configuration

In an active configuration scheme, the FLEX 8000 device controls the entire configuration process and generates the synchronization and control signals necessary to configure and initialize itself from an external memory. The active serial (AS) configuration scheme uses an Altera Configuration EPROM to store the configuration data. The active parallel up (APU) and active parallel down (APD) configuration schemes use a parallel-format memory such as a 32K × 8-bit EPROM as the data source.

Passive Configuration

In a passive configuration scheme, the FLEX 8000 device is incorporated into a system with an intelligent host that controls the configuration process. The intelligent host transparently selects a serial or parallel data source, and the data is presented to the FLEX 8000 device on a common data bus. In this type of system, the configuration data can be stored in a mass-storage medium, such as a hard disk. With passive configuration schemes, new configuration data is easily installed by supplying a new configuration file on a diskette or tape.

Choosing a Configuration Scheme

The best configuration scheme for a particular application depends on many factors, such as the presence of an intelligent host in the system, the need to reconfigure in real-time, and the need to periodically install new configuration data. Available board space is also a consideration for configuration schemes that use parallel or serial EPROMs to store configuration data.

The following guidelines can help you decide which configuration scheme is most appropriate for your application:

- For fast time-to-market, the easiest and quickest configuration schemes to implement are the three active configuration schemes: active serial (AS), active parallel up (APU), and active parallel down (APD). These configuration schemes require no external intelligence. The FLEX 8000 device is typically configured automatically at system power-up. If the FLEX 8000 device senses a power failure, it automatically triggers a reconfiguration cycle.

- For fast prototyping and development work, the passive serial (PS) configuration scheme, together with the FLEX Download Cable, provides the quickest means of iterative design analysis. The MAX+PLUS II Programmer can directly download configuration data to a FLEX 8000 device on the prototype circuit board.

- If a FLEX 8000 device is incorporated into a system with an intelligent host, you can use this host to control the configuration process in one of the passive configuration schemes: passive parallel asynchronous (PPA), passive parallel synchronous (PPS), or passive serial (PS). The configuration data can be stored in a mass-storage medium, such as a hard disk, thereby reducing the number of ICs required for the system. The FLEX 8000 device configuration can also be synchronized with any other system resources that must be initialized.

- In applications that require real-time device reconfiguration—such as data transformation filters, video formatters, and encryption/decryption circuits—the best choice is one of the passive configuration schemes. Reconfigurability allows you to reuse the logic resources within the FLEX 8000 device, instead of designing redundant or duplicate circuitry into your systems. Passive configuration schemes easily support the multiple sources of configuration data that may be required for real-time configuration. However, these schemes require more external circuitry. The FLEX 8000 device must rely on an intelligent host to retrieve and load new configuration data, and cannot perform any of the tasks required for reconfiguration.

- If field upgrades are anticipated, passive configuration schemes offer the ability to easily install new configuration data. New configuration files can be supplied to end users on diskette or tape. (In active schemes, a new EPROM must be inserted into the system.)

You can also use multiple configuration schemes during system operation. If you choose a single configuration scheme, you can simply hard-wire the three configuration scheme selection pins (`nSP`, `MSEL1`, and `MSEL0`) to their necessary levels (V_{CC} or GND). If you use multiple configuration

schemes, you can drive these selection pins with some controlling logic or connect them to a port on an intelligent host. For example, you can configure a FLEX 8000 device with an AS configuration scheme to load its "start-up" configuration data, then dynamically change the configuration scheme selection bits to select a different configuration scheme, and provide a different configuration data source.

FLEX 8000 Device Configuration Schemes

The following sections describe each configuration scheme in detail:

- ❏ Active serial (AS) configuration
- ❏ Active parallel up (APU) and active parallel down (APD) configuration
- ❏ Passive parallel synchronous (PPS) configuration
- ❏ Passive parallel asynchronous (PPA) configuration
- ❏ Passive serial (PS) configuration

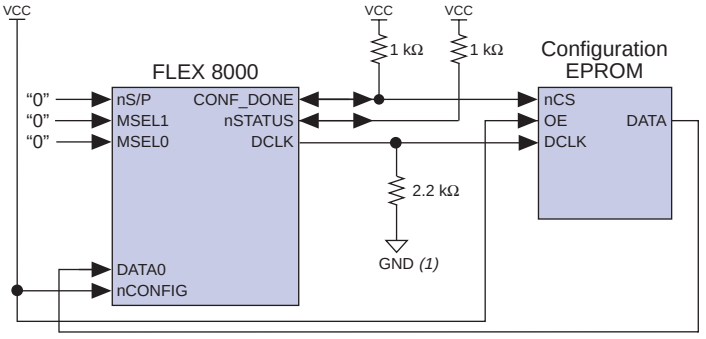
In-circuit reconfiguration, device configuration option bits, device configuration pins, and the source of data for each configuration scheme are described later in this application note.

Active Serial Configuration

The active serial (AS) configuration scheme uses an Altera-supplied serial Configuration EPROM (e.g., EPC1213) as a data source for FLEX 8000 devices. The Configuration EPROM presents its data to the FLEX 8000 device in a serial bit-stream. Figure 1 shows a typical circuit in which the FLEX 8000 device controls the configuration process and uses a Configuration EPROM as the data source.

The nCONFIG pin on the FLEX 8000 device in Figure 1 is connected to V_{CC}, so the device automatically configures itself at system power-up. The system can monitor the nSTATUS pin to ensure that configuration occurs

Figure 1. Active Serial Device Configuration



Note:
(1) Optional. For active-serial configuration modes only where system-noise on DCLK may be present.

correctly. Immediately after power-up, the FLEX 8000 device pulls the nSTATUS pin low and releases it within 100 ms. Once released, the open-drain nSTATUS pin is pulled up to V_{CC} by an external 1.0-kΩ pull-up resistor. If an error occurs during configuration, the FLEX 8000 device pulls the nSTATUS pin low, indicating that configuration was unsuccessful.

The DCLK signal, which is driven by the FLEX 8000 device, clocks sequential data bits from the Configuration EPROM. While the SRAM data is being loaded, the FLEX 8000 device holds the open-drain CONF_DONE pin at GND, indicating that data is loading. A 24-bit program-length counter within the FLEX 8000 device stores the program length, i.e., the total number of configuration bits. Once the terminal count value for the configuration data (i.e., the last configuration data bit) has been reached, the FLEX 8000 device releases the CONF_DONE pin, which is subsequently pulled up to V_{CC} by an external 1.0-kΩ pull-up resistor. The resulting high input on the nCS pin causes the Configuration EPROM to tri-state its DATA output, electrically removing the Configuration EPROM from the circuit.

After it releases the CONF_DONE pin, the FLEX 8000 device uses it as an input for monitoring the configuration process. When the FLEX 8000 device senses a high logic level on CONF_DONE, it completes the initialization process and enters user mode. Figure 2 shows the timing associated with the AS configuration process and the order of transitions on the control signals.

Figure 2. Active Serial Configuration Timing Waveforms

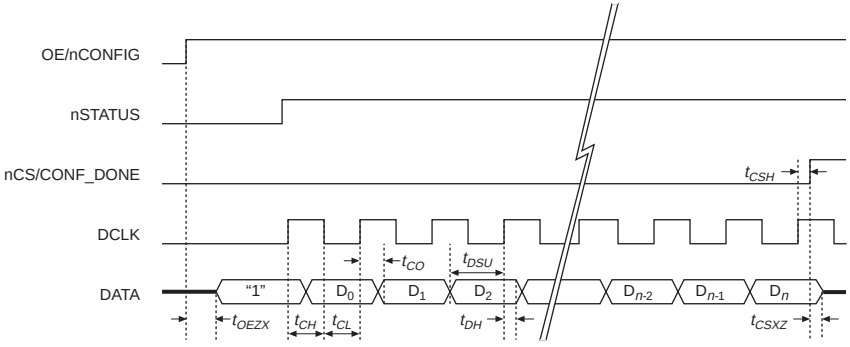


Table 3 provides values for the AS timing parameters.

Table 3. Active Serial Configuration Timing Parameters

Symbol	Parameter	Min	Max	Unit
t_{OEZX}	OE high to DATA output enabled		50	ns
t_{CSZX}	nCS low to DATA output enabled		50	ns
t_{CSXZ}	nCS high to DATA output disabled		50	ns
t_{CH}	DCLK high time	80	250	ns
t_{CL}	DCLK low time	80	250	ns
t_{DSU}	Data setup time before rising edge on DCLK	50		ns
t_{DH}	Data hold time after rising edge on DCLK	0		ns
t_{CO}	DCLK to DATA out		75	ns
t_{OEW}	OE low pulse width to guarantee counter reset	100		ns
t_{CSH}	nCS low hold time after DCLK rising edge	0		ns
f_{MAX}	DCLK frequency	2	6	MHz

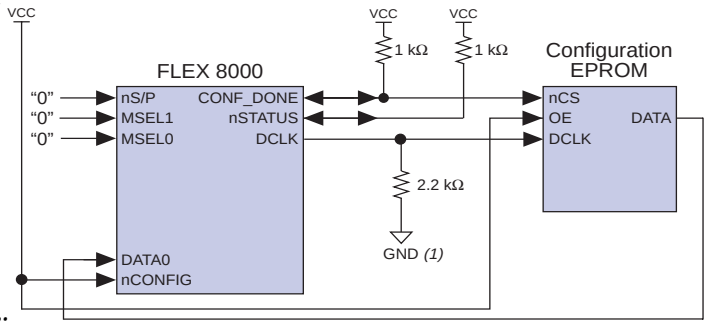
In the circuit shown in Figure 1, the nCONFIG pin on the FLEX 8000 device is tied to the Output Enable (OE) input of the Configuration EPROM; both are tied to V_{CC}. A high logic level on the nCONFIG input automatically starts the configuration. The output of the Configuration EPROM is enabled by a high input on its OE pin. If an error occurs during circuit configuration, the FLEX 8000 device pulls and holds the nSTATUS pin low, indicating a configuration error. External circuitry is used to monitor the nSTATUS pin and take appropriate action if configuration fails. This circuitry must assert a high-low-high pulse on the nCONFIG pin to reconfigure the device after the error. The same circuitry can also be used to begin reconfiguring the FLEX 8000 device at any time after system power-up.

The FLEX 8000 device's built-in *Auto-Restart Configuration on Frame Error* option bit, which can be set with MAX+PLUS II software, allows the device to automatically reconfigure itself if it encounters an error during configuration. (For descriptions of all FLEX 8000 device option bits, refer to "Device Configuration Option Bits" later in this application note.) If this option bit is turned on, a configuration error causes the FLEX 8000 device to pull the nSTATUS pin low for 10 internal Clock cycles and then release it. This 1- to 3-µs pulse on the nSTATUS pin provides an external indication that reconfiguration is about to begin. It also can be used to reset the Configuration EPROM.

Figure 3 shows a circuit that uses the *Auto-Restart Configuration on Frame Error* option. The nSTATUS pin is connected to the OE input on the Configuration EPROM so that the error-reset pulse on nSTATUS resets the internal address counter on the Configuration EPROM and prepares it to reconfigure the FLEX 8000 device. The nCONFIG input is also available to initiate a reconfiguration cycle externally. Since the nSTATUS pin is pulled

low and then released whenever configuration begins, it resets the Configuration EPROM before reconfiguration. If V_{CC} drops below the power-on reset (POR) threshold for the FLEX 8000 device during device operation, nSTATUS is pulsed and the Configuration EPROM is reset in the same way to provide automatic reconfiguration. Timing for the circuit in Figure 3 is identical to the timing shown in Figure 2 for the AS configuration scheme (the error-reset pulse on nSTATUS is not shown).

Figure 3. Active Serial Device Configuration with Automatic Reconfiguration on Error



Note:

(1) Optional. For active-serial configuration modes only where system-noise on DCLK may be present.

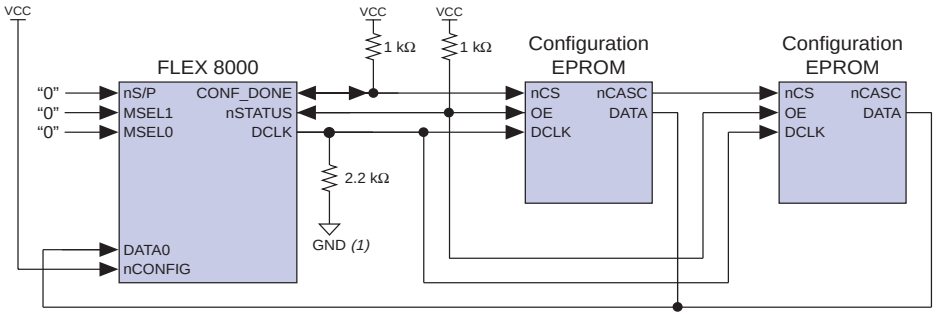
Altera Configuration EPROMs are designed for performance that is compatible with the setup and hold time requirements of FLEX 8000 devices. Refer to the current *Configuration EPROMs for FLEX 8000 Devices Data Sheet* for complete details on timing and circuitry. Details on device programming are given in "Programming a Configuration EPROM" later in this application note.

Active Serial Configuration for Multiple Configuration EPROMs

Multiple Configuration EPROMs can be serially connected to configure a FLEX 8000 device that requires more configuration data than a single Configuration EPROM can store. For example, the EPF81500 requires approximately 250 Kbits of configuration data, but an EPC1213 Configuration EPROM stores a maximum of 213 Kbits. Therefore, two Configuration EPROMs are needed to configure an EPF81500 device.

Figure 4 shows a typical circuit in which a FLEX 8000 device is configured by two Altera Configuration EPROMs. The FLEX 8000 device drives the DCLK signal out to both Configuration EPROMs during configuration, and receives configuration data on its DATA0 input. The first Configuration EPROM drives its nCASC output low and tri-states its DATA pin after clocking out all of its configuration data. The high-to-low transition on nCASC enables the nCS input on the second Configuration EPROM and

Figure 4. Active Serial Configuration of an EPF81500 Device with Automatic Reconfiguration on Error



Note:
(1) Optional. For active-serial configuration modes only where system-noise on DCLK may be present.

activates the EPROM within one DCLK cycle. This handshaking is transparent to the FLEX 8000 device.

Once all configuration data has been clocked into the FLEX 8000 device, the device releases the CONF_DONE pin, which is subsequently pulled up to V_{CC} by an external 1.0-kΩ pull-up resistor. The resulting high input on the nCS input to the first Configuration EPROM drives its nCASC output high, which in turn drives the nCS input to the second Configuration EPROM high, electrically removing both Configuration EPROMs from the circuit.

In the circuit shown in Figure 4, the FLEX 8000 device's built-in *Auto-Restart Configuration on Frame Error* option bit allows the device to automatically reconfigure itself if it encounters an error during configuration. The nSTATUS pin is connected to the OE pins on the Configuration EPROMs. If the FLEX 8000 device detects a configuration error, it pulls the nSTATUS pin low for 10 internal Clock cycles and then releases it. This 1- to 3-µs pulse on the nSTATUS pin resets the Configuration EPROMs with a low pulse on the OE pins.

Active Parallel Up & Active Parallel Down Configuration

In the active parallel up (APU) and active parallel down (APD) configuration schemes, the FLEX 8000 device generates sequential addresses that drive the address inputs to an external PROM. The PROM then returns the appropriate byte of data on the data pins DATA[7..0]. Sequential addresses are generated until the FLEX 8000 device has been completely loaded. The CONF_DONE pin is then released and pulled high externally, indicating that configuration has been completed. The counting sequence can be ascending (00000H to 3FFFFH) for APU configuration or descending (3FFFFH to 00000H) for APD configuration.

Figure 5 shows a typical circuit with a FLEX 8000 device and a parallel EPROM for APU or APD configuration. In this circuit, the nCONFIG input to the FLEX 8000 device is connected to a system-wide, active-low Reset signal. The nCONFIG pin can be tied to V_{CC} (as shown in Figure 1) to start configuration automatically at system power-up; however, the system-wide Reset allows you to explicitly control the time at which configuration begins. The nCONFIG pin must be held low to meet the minimum low pulse width requirement for t_{CFG} (see Table 4 later in this application note).

Figure 5. Active Parallel Device Configuration with a 256-Kbyte EPROM

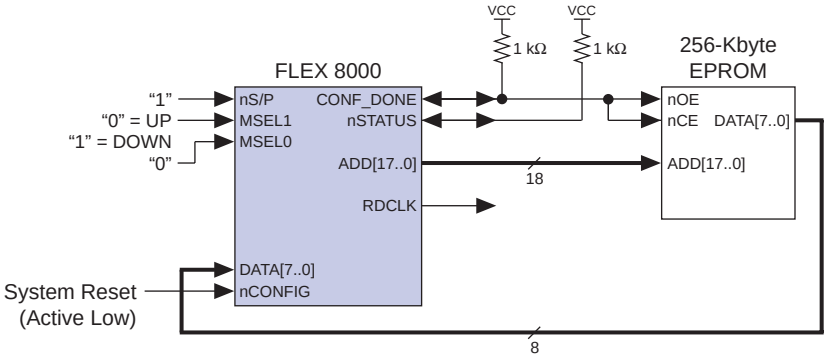
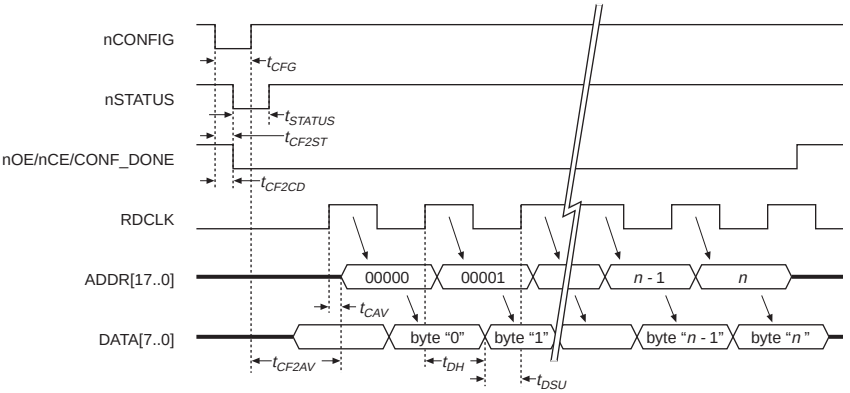


Figure 6 shows the timing associated with the circuit in Figure 5. The high-low-high pulse on the nCONFIG pin starts the configuration process. The nSTATUS pin is pulled low for up to 100 ms, and the CONF_DONE pin is pulled down to GND. Once the CONF_DONE pin is low, address generation begins. The low logic level on the CONF_DONE pin also enables the output of the EPROM. In an APU configuration scheme, the first address generated is 00000H; in an APD configuration scheme, it is 3FFFFH.

The configuration events in Figure 6 are based on the RDCLK signal rather than the DCLK signal. The RDCLK signal, a Clock signal that is generated by dividing the DCLK signal by eight, is used to frame the data bytes supplied by the parallel EPROM. In the APU and APD configuration schemes, the FLEX 8000 device generates the DCLK signal internally and uses it to serialize the incoming data words. On each pulse of the RDCLK signal, the FLEX 8000 device latches an 8-bit byte, and the following eight pulses on DCLK convert that 8-bit value into a serial data stream. The RDCLK signal is available as an output pin during configuration. (In user mode, the RDCLK pin is available as an I/O pin.) You can monitor this signal to ensure that the parallel EPROM observes the data setup and hold time requirements for the FLEX 8000 device.

Figure 6. Active Parallel Up Configuration Timing Waveforms

A rising edge on RDCLK increments the address counter ADDR[17..0], which is driven out to the parallel EPROM. The parallel EPROM then sends the addressed byte of configuration data to the FLEX 8000 device.



A new address is presented on the ADD[17..0] pins a short time (t_{CAV}) after a rising edge on RDCLK. Table 4 shows the timing parameters for the APU and APD configuration in Figure 5. Before the subsequent rising edge on RDCLK, the external parallel EPROM must present valid data soon enough to meet the t_{DSU} setup time for the data. This subsequent rising edge on RDCLK latches data, based on the address generated by the previous Clock cycle. EPROMs with access times faster than 500 ns should be used to guarantee the data setup time.

Table 4. Active Parallel Up & Down Configuration Timing Parameters

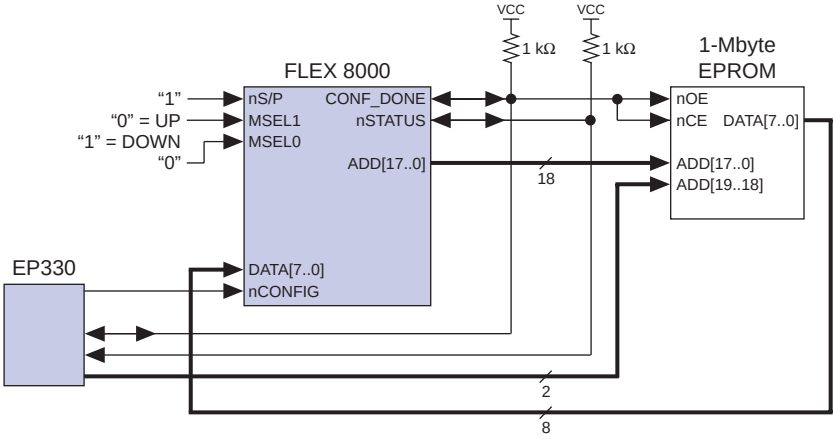
Symbol	Parameter	Min	Max	Unit
t_{CF2ST}	nCONFIG low to nSTATUS low		1	µs
t_{CFG}	nCONFIG low pulse width	2		µs
t_{STATUS}	nSTATUS low pulse width	2.5		µs
t_{CF2CD}	nCONFIG low to CONF_DONE low		1	µs
t_{CF2AV}	nCONFIG high to first valid address		3.5	µs
t_{CAV}	RDCLK rising edge to address valid		1	µs
t_{DH}	Data hold time after rising clock edge (RDCLK)	0		ns
t_{DSU}	Data setup time before rising clock edge (RDCLK)	50		ns

Once the terminal count value for the FLEX 8000 device configuration data is reached, the FLEX 8000 device releases the CONF_DONE pin. The CONF_DONE pin is pulled up to V_{CC} via the pull-up resistor, and the FLEX 8000 device disables the output on the EPROM.

All FLEX 8000 devices provide 18 address lines, which are sufficient to uniquely decode up to 256 Kbytes of data, much more than the largest FLEX 8000 device requires (see Table 2). Although the 18 address lines limit FLEX 8000 devices to addressing 256 Kbytes of data, you can use a larger EPROM device (e.g., 512 Kbytes, 1 Mbyte, 2 Mbytes, etc.) by masking in the necessary offset addresses. In larger EPROMs, the FLEX 8000 device configuration information is treated as a separate "page" in the EPROM, and can be placed on any convenient boundary. However, some additional logic is required to provide the offset address.

Figure 7 shows how you can use an Altera EP330 device as a decoder that asserts the necessary page-offset address onto the address bus during configuration. The EP330 allows the 18-bit address generated by the FLEX 8000 device to select one of four 256-Kbyte "pages" in the EPROM. The EP330 should monitor the nSTATUS and CONF_DONE signals to ensure that errors are handled correctly. The inputs to the EP330 must be system-level control signals that select the appropriate page in the EPROM to be loaded into the FLEX 8000 device, and control when the configuration actually occurs. Timing for the circuit in Figure 7 is identical to the timing shown in Figure 6.

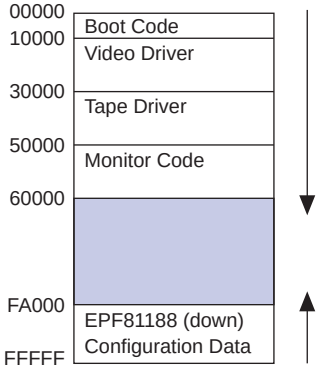
Figure 7. Active Parallel Device Configuration with Offset Address Generation Circuitry



The active parallel configuration schemes can generate addresses in either an ascending or descending order, depending on your system requirements. Counting up (APU configuration) is appropriate if the FLEX 8000 configuration data is stored at the beginning of an EPROM, or if the configuration data has been placed at some known offset in an EPROM larger than 256 Kbytes. Counting down (APD configuration) is appropriate if the low addresses are not available, e.g., if the CPU code must use the beginning of the EPROM or if the EPROM is also used to store other information that is expected to increase as an application evolves. The changing nature of the data size is characteristic of basic I/O system (BIOS) and boot PROMs.

Figure 8 shows an example of a BIOS EPROM memory map, in which the FLEX 8000 configuration data is placed at the bottom of the memory space in an APD configuration.

Figure 8. Typical BIOS EPROM Memory Map



Passive Parallel Synchronous Configuration

In a passive parallel synchronous (PPS) configuration scheme, the FLEX 8000 device is tied to an intelligent host. With PPS configuration, data can be driven directly onto a common data bus between the host and the FLEX 8000 device. The `DCLK`, `CONF_DONE`, `nCONFIG`, and `nSTATUS` signals are connected to a port on the host. Although you can drive the `DCLK` signal from the system Clock, you must have precise control of any interrupts that can influence the internal counting of the FLEX 8000 devices. This precise control is required because the FLEX 8000 device latches data on the rising edge of the `DCLK` signal, and the next eight falling edges of the `DCLK` signal serialize the latched data. New data is latched on every eighth rising edge of the `DCLK` signal until the FLEX 8000 device is completely configured.

Figure 9 illustrates PPS configuration of a FLEX 8000 device. In this circuit, the CPU generates a byte of configuration data and directs the FLEX 8000 device to latch and serialize the data by strobing a high pulse on the DCLK input. In Figure 9, no specific source is shown for the data bus DATA[7..0], which is typically driven by a dedicated data latch. A microcontroller host usually has byte-wide ports that can be used for this data bus. If the host is a CPU or intelligent logic, a dedicated data register can be implemented with an octal latch. Depending on the capability of the host and the memory space implementation in the system, you can use an external memory instead to drive the data onto the system data bus. This type of external memory usage requires the memory to hold the data on the bus while the host executes the commands to direct the FLEX 8000 device to latch and serialize the data. However, not all processors can accommodate this type of operation.

Figure 9. Passive Parallel Synchronous Device Configuration

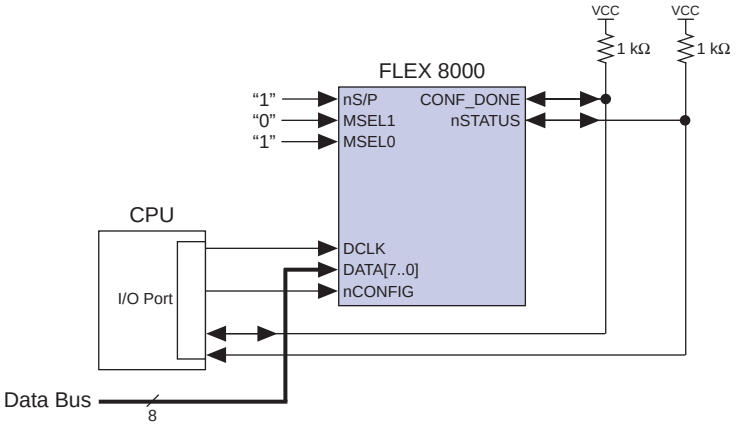


Figure 10 shows the timing for the PPS configuration scheme. The CPU generates Clock cycles and data; eight DCLK cycles are required to latch and serialize each 8-bit data word. A new data word must be present at the DATA[7..0] inputs upon every eighth DCLK cycle.

Figure 10. Passive Parallel Synchronous Configuration Timing Waveforms

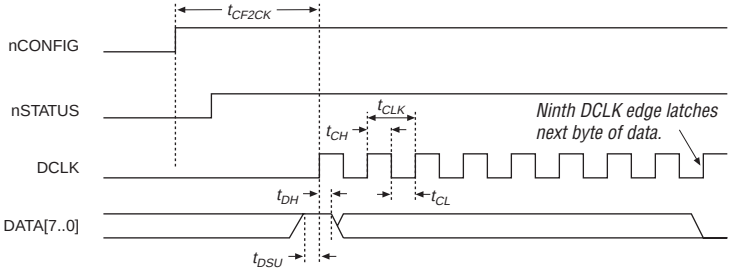


Table 5 shows the timing parameters associated with PPS configuration.

Table 5. Passive Parallel Synchronous Configuration Timing Parameters

Symbol	Parameter	Min	Max	Unit
t_{CF2CK}	nCONFIG high to first rising edge on DCLK	5		µs
t_{DSU}	Data setup time before rising edge on DCLK	50		ns
t_{DH}	Data hold time after rising edge on DCLK	0		ns
t_{CH}	DCLK clock high time	80		ns
t_{CL}	DCLK clock low time	80		ns
t_{CLK}	DCLK period	160		ns
f_{MAX}	DCLK maximum frequency		6	MHz

Passive Parallel Asynchronous Configuration

With the passive parallel asynchronous (PPA) configuration scheme, a FLEX 8000 device in a system can be configured in parallel with the rest of a board. The FLEX 8000 device accepts a parallel byte of input data, then serializes the data with its internal synchronization Clock. The device is selected with the nCS and CS chip select pins, so multiple devices can reside on the same data bus. The ability to select individual FLEX 8000 devices allows multiple devices to be configured in parallel by a single intelligent host.

This efficient handshaking allows an intelligent host to simultaneously configure multiple FLEX 8000 devices or other configurable portions of the system. Figure 11 illustrates PPA configuration of a FLEX 8000 device. A microcontroller is used as the intelligent host to ensure that sufficient dedicated I/O ports are available to drive all control signals and the data bus to the FLEX 8000 device. The chip select signals CS and nCS are both used to select the device. However, you can also tie nCS to GND and control chip selection with the CS pin only (or vice-versa), thus saving one bit in the I/O port.

Figure 11. Passive Parallel Asynchronous Device Configuration with Dedicated Ports

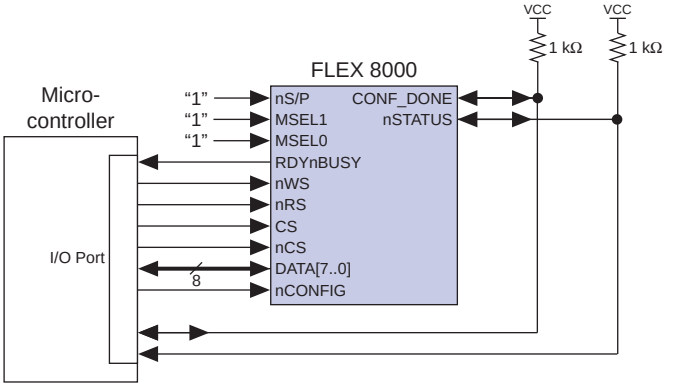
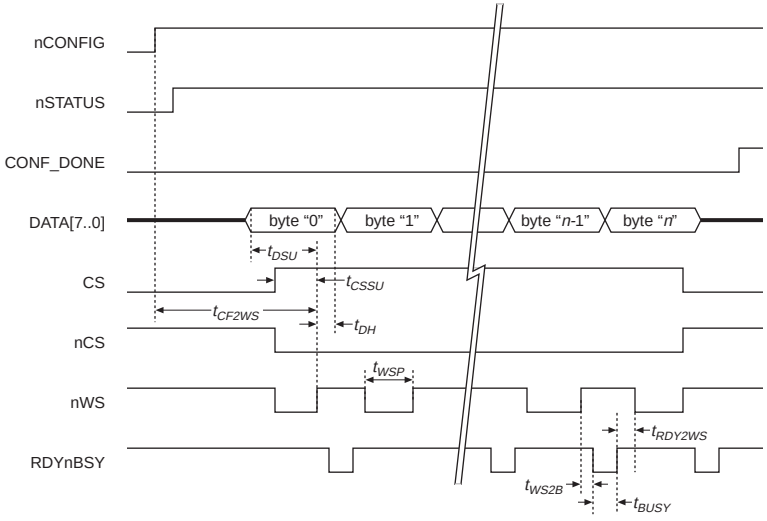


Figure 12 shows the timing for the PPA configuration scheme. The CPU presents an 8-bit data word to the FLEX 8000 device, and indicates that the word is valid by strobing a low pulse on the nWS input. The FLEX 8000 device senses the rising edge of the nWS signal, latches the data on the DATA[7..0] inputs, and uses its internal oscillator to serialize the 8-bit data word.

Figure 12. Passive Parallel Asynchronous Timing Waveforms



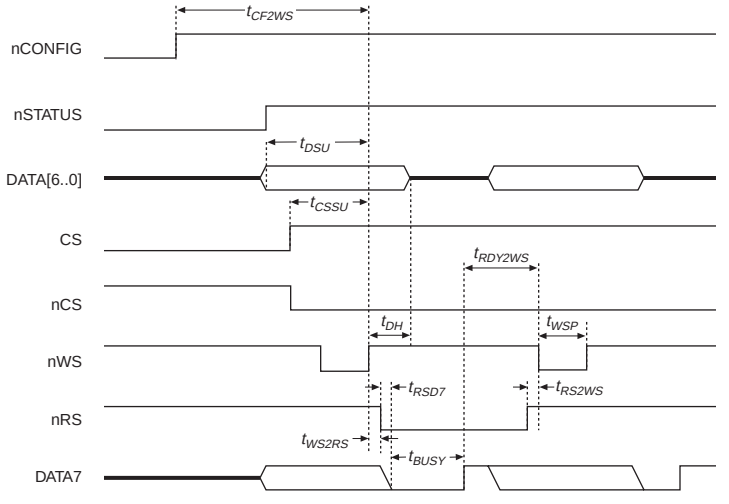
The CPU must poll the RDYnBUSY signal to establish when the FLEX 8000 device is ready to receive more data. RDYnBUSY falls immediately after the rising edge of the nWS signal that latches data, indicating that the device is busy. While the FLEX 8000 device processes the data byte, RDYnBUSY remains low. On the eighth falling edge of DCLK, RDYnBUSY returns to V_{CC}, indicating that another byte of data can be latched. Table 6 shows the timing parameters associated with PPA configuration.

Table 6. Passive Parallel Asynchronous Configuration Timing Parameters

Symbol	Parameter	Min	Max	Unit
t_{CF2WS}	nCONFIG high to first nWS rising edge	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 selected delay before rising edge on nWS	50		ns
t_{WSP}	nWS low pulse width	500		ns
t_{WS2B}	nWS rising edge to RDYnBSY low		50	ns
t_{BUSY}	RDYnBSY low pulse width		4	µs
t_{RDY2WS}	RDYnBSY rising edge to nWS falling edge	50		ns
t_{WS2RS}	nWS rising edge to nRS falling edge	500		ns
t_{RS2WS}	nRS rising edge to nWS falling edge	500		ns
t_{RSD7}	nRS falling edge to DATA7 valid with RDYnBSY signal		50	ns

As an alternative to polling the RDYnBUSY signal, the CPU can determine the status of the FLEX 8000 device by strobing a low pulse on the nRS input to the FLEX 8000 device. This strobe causes the FLEX 8000 device to present the RDYnBUSY status on the bidirectional pin DATA7 so that the CPU can determine device status from the data bus, instead of using an additional port on the CPU for the RDYnBUSY signal. This low pulse on nRS must occur only during the corresponding high pulse (inactive) on the nWS signal. The timing waveforms in Figure 13 show how the nRS pin can be used to poll the status of the device with the bidirectional pin DATA7 of the circuit shown in Figure 11. The timing parameters given in Table 6 also apply to Figure 13.

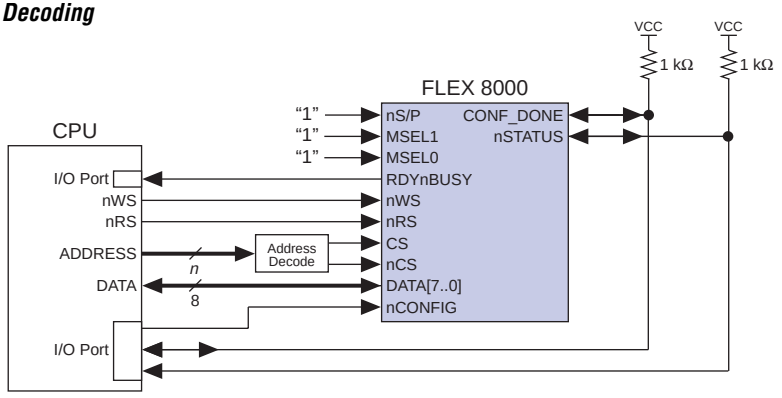
Figure 13. Passive Parallel Asynchronous Timing Waveforms Using nRS & DATA7



The circuit in Figure 11 takes advantage of the architecture of a microcontroller host. Figure 14 shows an alternative to this circuit, in which a CPU serves as the intelligent host and the FLEX 8000 device is treated more as a memory than as a port. The nWS and nRS inputs to the FLEX 8000 device are driven by the CPU's memory read/write control pins; the DATA[7..0] inputs to the FLEX 8000 device are driven directly by the system data bus. As in Figure 11, the nSTATUS and nCONFIG control signals must be driven by an intelligent I/O port, but the CS and nCS chip select signals are decoded from the address bus and not driven from an I/O port on the CPU. This address decoding scheme allows the CPU to write to the FLEX 8000 device as a memory. A small programmable logic device, such as the Altera EPM7032, is ideal for quickly decoding a wide address and selecting the FLEX 8000 device.

PPA configuration is useful when multiple FLEX 8000 devices are configured simultaneously. The CPU reads a byte of configuration data from the disk or from memory, and then writes it to the FLEX 8000 device. The CPU then polls the RDYnBUSY signal (or the DATA7 pin via the nRS input) to determine when another data byte can be written. Timing for this circuit is identical to the timing shown in Figure 13, although t_{CSSU}, the minimum chip select delay before the rising edge of nWS, must increase to account for the time required to decode the address.

Figure 14. Passive Parallel Asynchronous Device Configuration with Address Decoding



The configuration process is generally controlled with a precise order of steps, so the timing constraints are minimal. The following steps show the typical control sequence executed by the CPU:

1. Pull the nCONFIG pin to GND, hold it for 10 µs, then pull it up to V_{CC}.
2. Read the next byte of configuration data from an EPROM or a mass storage device such as a hard disk.
3. Generate the address of the FLEX 8000 device.
4. Perform a memory write cycle to the FLEX 8000 device address using the stored configuration data byte.
5. Poll the RDYnBUSY signal. When it goes high, transfer the next byte of configuration data by repeating steps 2 through 4.
6. Repeat steps 2 through 5 until the FLEX 8000 device pulls the CONF_DONE net high, which indicates that configuration is complete.

Passive Serial Configuration

The passive serial (PS) configuration scheme uses an external controller to configure the FLEX 8000 device with a serial bit-stream. The FLEX 8000 device is treated as a slave device with a 5-wire interface to the external controller. The external controller can be one of the following:

- ❏ The MAX+PLUS II Programmer, used together with the PL-MPU Master Programming Unit, an appropriate device adapter, and the FLEX Download Cable.

- ❏ An intelligent host such as a microcontroller or a CPU. This type of PS configuration is similar to the PPA and PPS configuration schemes, but uses a bit-wide serial data path instead of a byte-wide parallel data path.

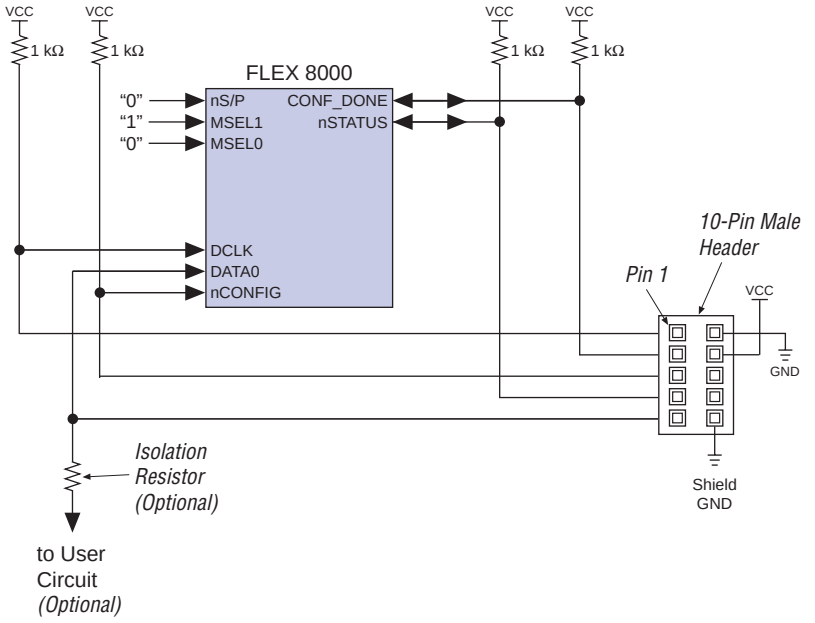
- The Altera BitBlaster, an RS-232-compatible serial download cable. For information on using the BitBlaster for configuring FLEX 8000 devices, refer to the current *BitBlaster Serial Download Cable Data Sheet*.

Passive Serial Configuration with the FLEX Download Cable

Passive serial configuration with the FLEX Download Cable uses the MAX+PLUS II Programmer and Altera programming hardware as the external controller. The Altera FLEX Download Cable can connect any Configuration EPROM programming adapter, which is installed on the PL-MPU Master Programming Unit, to a single target FLEX 8000 device in the prototype system. The FLEX Download Cable provides a 5-wire connection between the FLEX 8000 device and the programming adapter. Configuration data is taken from the SRAM Object File (**.sof**) generated automatically during project compilation and downloaded by the MAX+PLUS II Programmer. Once the device is configured, the programming hardware is tri-stated and electrically removed from the circuit. This type of PS configuration allows you to perform multiple design iterations rapidly.

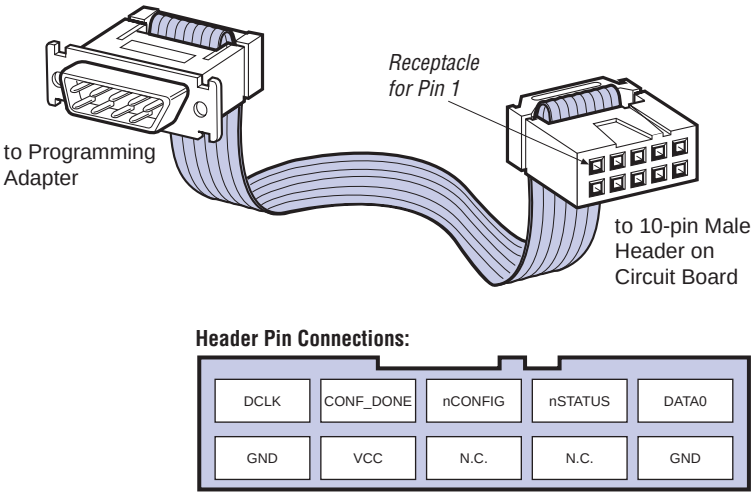
Figure 15 shows how the FLEX Download Cable interfaces to the target FLEX 8000 device. The 10-pin male header on the circuit board has two rows of five pins, spaced on 0.1-inch centers, that connect to the

Figure 15. Passive Serial Device Configuration with the FLEX Download Cable



configuration pins on the FLEX 8000 device. Standard 10-pin IDS-type male headers are readily available to provide the target board connections. A 10-pin female plug on one end of the FLEX Download Cable is connected to the 10-pin male header on the circuit board; the other end of the FLEX Download Cable is connected to a Configuration EPROM programming adapter. See Figure 16. Timing for PS configuration is identical to the timing for bit-wide PS configuration shown later in this application note.

Figure 16. FLEX Download Cable Signals & Positions



When a FLEX 8000 device is configured via the FLEX Download Cable, the DCLK, CONF_DONE, nCONFIG, DATA0, and nSTATUS pins on the cable are connected directly to the pins of the same names on the FLEX 8000 device. The VCC and GND pins must be tied to the system power planes. These VCC and GND pins supply power to the optical isolation circuitry in the programming adapter; they do not supply power to the target FLEX 8000 device. Refer to the current *FLEX 8000 Programmable Logic Device Family Data Sheet* for device pin numbers.

The DCLK, CONF_DONE, nCONFIG, and nSTATUS pins on the FLEX 8000 device are dedicated configuration pins. Since they are not available as user I/O pins, they do not require isolation from the rest of the circuit. However, a system must include pull-up resistors that pull these pins up to V_{CC}, as shown in Figure 15. These resistors allow you to remove the FLEX Download Cable after configuration is complete without introducing any noise from floating inputs.

The DATA0 pin is available as an I/O pin during user-mode operation, and may require isolation, depending on how it is used. During configuration,

the DATA0 pin on the FLEX 8000 device acts as an input, and is driven by the programming hardware. If the DATA0 pin is an output pin during user mode, the signal that it drives does not need to be buffered. However, if the DATA0 pin is an input or bidirectional pin during user mode, contention may occur between the user-mode signal and the FLEX Download Cable during configuration.

If the signal that drives the DATA0 pin during user mode is tri-stated during configuration and initialization, no conflict occurs. However, if this signal is active during configuration, the DATA0 input pin must be isolated from the active source. You can isolate the DATA0 pin by inserting a tri-state buffer between the DATA0 pin and the rest of the network that it drives. This tri-state buffer must be controlled by external logic.

If you cannot use active isolation, placing a 550-Ω resistor between the user-mode signal and the DATA0 pin should provide adequate isolation. The FLEX Download Cable is driven by 12-mA drivers, which supply sufficient current to mask any signals that may be present at the other end of the resistor. Resistive isolation may not be suitable for very-high-speed circuits. Actual in-circuit performance should be evaluated in the laboratory to ensure that this isolation scheme does not affect other portions of the circuit.

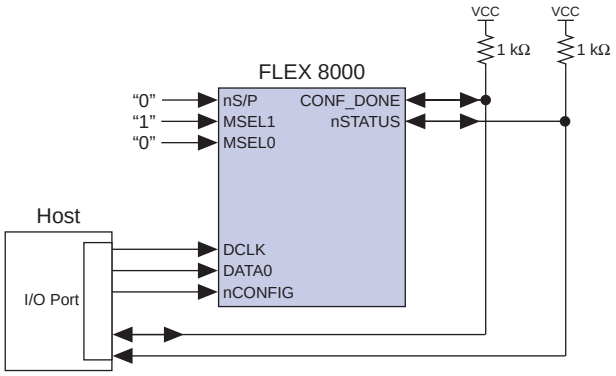
The no connect (N.C.) pins shown in Figure 16 are reserved, and should not be tied to any data or power signals. The header should be placed as close as possible to the FLEX 8000 device.

For additional information on passive serial configuration with the FLEX Download Cable, refer to "Configuring a FLEX 8000 Device In-System with MAX+PLUS II & the FLEX Download Cable" later in this application note.

Bit-Wide Passive Serial Configuration

The passive serial configuration scheme provides a bit-wide passive interface for device configuration. No handshaking is provided in any PS configuration. Therefore, the FLEX 8000 device must be configured at 6 MHz or less. Figure 17 shows how a bit-wide PS configuration is implemented. Data bits are presented on the DATA0 input, with the least significant bit of each byte of data presented first. The DCLK is strobed with a high pulse to latch the data. This serial data loading continues until the CONF_DONE pin goes high, indicating that the device is fully configured. After the last data byte, the DCLK pin must be clocked 10 times for the FLEX 8000 device to release CONF_DONE and initialize the device. After CONF_DONE is released, the DCLK pin must be clocked 10 times for the FLEX 8000 device to release CONF_DONE The data source can be any source

Figure 17. Bit-Wide Passive Serial Device Configuration



that the host can address.

Figure 18. Bit -Wide Passive Serial Timing Waveforms

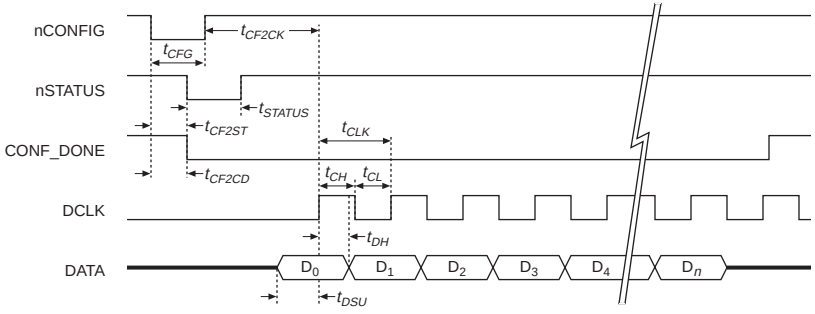


Figure 18 shows the timing for bit-wide PS configuration.

Table 7. Passive Serial Configuration Timing Parameters

Symbol	Parameter	Min	Max	Unit
t_{CF2CD}	nCONFIG low to CONF_DONE low		1	µs
t_{CF2ST}	nCONFIG low to nSTATUS low		1	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_{DSU}	Data setup time before rising edge on DCLK	50		ns
t_{DH}	Data hold time after rising edge on DCLK	0		ns
t_{CH}	DCLK high time	80		ns
t_{CL}	DCLK low time	80		ns
t_{CLK}	DCLK period	160		ns
f_{MAX}	DCLK maximum frequency		6	MHz

Table 7 gives the timing parameters for bit-wide PS configuration.

In-Circuit Reconfiguration

After a FLEX 8000 device has entered the user mode, you can choose to replace the configuration data pattern inside the device at any time. In this process, called *in-circuit reconfiguration*, new configuration data is selected using one of three methods, depending on the configuration scheme:

- ❏ In a passive configuration scheme, a different file can be downloaded from a mass-storage system.
- ❏ In the AS configuration scheme, multiple sets of configuration data can be stored in one or more serial Configuration EPROMs. Each set of data is used in succession.
- ❏ In the APU and APD configuration schemes, new configuration data is selected by externally multiplexing a different EPROM source onto the data path or by providing offset address generation circuitry to select a different page within the same EPROM.

Because the SRAM cells used to configure the functionality of the FLEX 8000 architecture are volatile, they can be reprogrammed without removing the FLEX 8000 device from the circuit board.

The nCONFIG input controls device reconfiguration. In the active configuration schemes shown in Figures 1, 3, and 4, the nCONFIG pin is tied to V_{CC} to force the FLEX 8000 device to automatically configure itself at system power-up. In the PPA and PPS configuration schemes, controlling logic is used on the nCONFIG input to determine when the configuration starts, as shown in Figures 9, 11, and 14. However, all configuration

schemes allow you to connect the nCONFIG pin to a port on an intelligent host, which can be used to control the configuration process. If nCONFIG is held low, the configuration process can be delayed as necessary. For example, the nCONFIG pin can be held low during system initialization and then pulled high when it is appropriate to configure the FLEX 8000 device.

At any time during system operation, regardless of the current state of the FLEX 8000 device, the nCONFIG pin can be used to restart the configuration process. When nCONFIG is driven low and then high again, the device resets itself and prepares for configuration. In an active configuration scheme, the FLEX 8000 device immediately starts retrieving data from the external EPROM; in a passive configuration scheme, it prepares to receive the data from the intelligent host. An example of a reset pulse on nCONFIG in an APU configuration scheme is shown in Figure 6 earlier in this application note. This nCONFIG timing applies to all configuration schemes whenever the device is reconfigured.

All latched and registered data in the device is lost during reconfiguration, so any counter values or the current state of the device should be stored either in the intelligent host's storage system or in some external circuitry, such as an Altera EPLD. The entire reconfiguration process requires about 100 ms. The system resumes normal operation after the FLEX 8000 device releases the CONF_DONE pin, indicating that initialization is complete.

Configuration Control Features

Within a FLEX 8000 device, the configuration and initialization processes can be controlled with two types of built-in resources:

- ❏ Device configuration option bits
- ❏ Device configuration pins

This section provides detailed information on configuration option bits and pins. The usage of various options and pins is discussed in the descriptions of individual configuration schemes earlier in this application note. Some configuration pins and options can also be used together to provide additional configuration and initialization control.

Device Configuration Option Bits

FLEX 8000 devices have device configuration option bits that allow you to control device behavior during configuration. Table 8 describes all FLEX 8000 device option bits and their availability in different configuration schemes. You can set these options on a device-by-device basis in the MAX+PLUS II software with the **FLEX 8000 Individual Device Options** dialog box. You can also enter global default device option settings for an

Table 8. FLEX 8000 Device Configuration Option Bits (Part 1 of 2)

Device Option	Configuration Scheme	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
User-Supplied Start-Up Clock	All	After a FLEX 8000 device is configured, it must be initialized over the course of 10 Clock cycles. The user can choose the source of the Clock.	In the AS, APU, APD, and PPA configuration schemes, the internal FLEX 8000 device oscillator supplies the initialization Clock. In the PS and PPS configuration schemes, the internal oscillator is disabled, so external circuitry must provide the initialization Clock on the DCLK pin.	The user provides the Clock on the CLKUSR pin. This type of Clock can be used to fully synchronize initialization for multiple FLEX 8000 devices. The maximum user-supplied Clock frequency is 6 MHz, and the Clock should have a 50% duty cycle.
Auto-Restart Configuration on Frame Error	AS, APU, APD AS	If a data error occurs when a FLEX 8000 device is configured with an active configuration scheme, the user can choose how to restart the configuration.	The configuration process halts and the user must externally direct the device to restart the configuration process. If a configuration error occurs, the nSTATUS pin is driven and held low until the nCONFIG pin is externally pulled low and then high again. In an AS configuration scheme, the external nCONFIG reset pulse resets the Configuration EPROM if the nCONFIG pin on the FLEX 8000 device is tied to the Output Enable pin on the Configuration EPROM.	Directs the device to automatically restart the configuration process. The nSTATUS pin is driven and held low for 10 Clock cycles and is then released. The nSTATUS pin subsequently pulls up to V_{CC}, indicating to any external circuitry that the reconfiguration process has started. In an AS configuration scheme, the nSTATUS reset pulse automatically resets the Configuration EPROM if the nSTATUS pin on the FLEX 8000 device is tied to the Output Enable pin on the Configuration EPROM.
Release Clears Before Tri-States	All	During configuration, the I/O pins on the device are tri-stated by an Output Enable override. The user can choose the order in which the tri-states are released and the registered logic cells and peripheral registers are cleared during initialization.	Directs the device to release the Output Enable override on the tri-state buffer before releasing the Clear signal on registered logic cells and peripheral registers during initialization.	Directs the device to release the Clear signal on registered logic cells and peripheral registers before releasing the Output Enable override on the tri-state buffer during initialization.
Enable DCLK Output In User Mode	AS, APU, APD, PPA	FLEX 8000 devices drive the DCLK signal during configuration in all active configuration schemes and the PPA configuration scheme. The DCLK signal can range from 2 to 6 MHz in frequency. The user can choose whether to enable the DCLK signal during user mode. The duty cycle and frequency of the DCLK signal are not guaranteed.	Disables the DCLK pin when the device operates in user mode after device configuration and initialization have been completed.	Enables the DCLK pin when the device operates in user mode after device configuration and initialization have been completed.

Table 8. FLEX 8000 Device Configuration Option Bits (Part 2 of 2)

Device Option	Configuration Scheme	Option Usage	Default Configuration (Option Off)	Modified Configuration (Option On)
Disable Start-Up Time-Out	All	The CONF_DONE pin, a bidirectional open-drain pin, is held at GND by the FLEX 8000 device during configuration. Once configuration is complete, the CONF_DONE pin is released and the FLEX 8000 device treats the pin as an input pin. In most applications, the CONF_DONE pin is pulled up to V_{CC} via a 1.0-kΩ resistor. This low-to-high transition directs the FLEX 8000 device to begin initialization. The user can enable or disable the time-out error checking that determines whether CONF_DONE goes high within 10 Clock cycles.	If the CONF_DONE pin does not go high within 10 Clock cycles after being released by the device, the device drives the nSTATUS pin low at the end of the configuration cycle, indicating an error condition.	If the CONF_DONE pin does not go high within 10 Clock cycles after being released by the device, the device continues to wait for CONF_DONE to go high. To delay initialization, the CONF_DONE node can be held low externally after the FLEX 8000 device has released the CONF_DONE pin, if, for example, the user wishes to control the time required for the FLEX 8000 device to enter user mode.
Enable JTAG Support	All	Enables post-configuration JTAG boundary-scan testing support in FLEX 8000 devices that provide JTAG circuitry.	JTAG boundary-scan testing is not available. In the EPF8282, EPF8282V, EPF8636, and EPF8820 devices, the four JTAG pins (TDI, TDO, TMS, and TCLK) are available as user I/O pins. In EPF81500 devices, these four pins are disabled.	JTAG boundary-scan testing is available on the four JTAG pins (TDI, TDO, TMS, and TCLK) after device configuration has been completed.

entire project with the **FLEX 8000 Device Options** dialog box.

Device Configuration Pins

FLEX 8000 devices include control pins that modify the sequence and timing of the configuration and initialization processes, and provide a variety of configuration options. Some configuration pins have the same

Table 9. Pin Functions (Part 1 of 2) *Note (1)*

Pin Name	User Mode	Configuration Scheme	Pin Type	Description
nSP	n/a	All	Input	Serial/Parallel selection input. A low input selects a serial configuration scheme; a high input selects a parallel configuration scheme.
MSEL1 MSEL0	n/a	All	Input	2-bit configuration scheme selection inputs that are used in conjunction with nSP to select the configuration scheme. The bit patterns of nSP:MSEL1:MSEL0 are associated with the following configuration schemes: 000 = AS 100 = APU 001 = Reserved 101 = PPS 010 = PS 110 = APD 011 = Reserved 111 = PPA
nSTATUS	n/a	All	Bidirectional Open Drain	Command mode status output. The FLEX 8000 device drives the nSTATUS pin low immediately after power-up, then releases it within 100 ms. The nSTATUS pin must be pulled up to V_{CC} with a 1.0-kΩ resistor. If an error occurs during configuration, nSTATUS is pulled low again by the FLEX 8000 device.
nCONFIG	n/a	All	Input	Configuration control input. A low input resets the FLEX 8000 device. A low-to-high transition starts a configuration cycle.
CONF_DONE	n/a	All	Bidirectional Open Drain Input	Status output. Driven low by the FLEX 8000 device during the configuration process. Status input. A high input directs the device to execute the initialization process and enter user mode. The CONF_DONE net must be pulled up to V_{CC} with a 1.0-kΩ resistor. The CONF_DONE pin may be actively driven low by an external source to delay the FLEX 8000 device initialization process. This feature is useful when the configuration process will be completed some time before actual operation is necessary.
DCLK	*(2)* *(3)*	AS PPS, PS	Output Input	Clock source for external PROM devices. Clock input from external host.
nWS	I/O	PPA	Input	Write Strobe input. A low-to-high transition causes the FLEX 8000 device to latch a byte of data on the DATA[7..0] pins.
nRS	I/O	PPA	Input	Read Strobe input. A low input directs the FLEX 8000 device to place the RDYnBUSY signal on the DATA7 pin.
RDCLK	I/O	APD, APU	Output	Divide-by-8 of DCLK output. Used internally to serialize an 8-bit data stream in the byte-wide APU or APD configuration scheme.
nCS CS	I/O	PPA	Input	Chip Select inputs. A low input on nCS and a high input on CS selects a specific FLEX 8000 device for configuration. If only one of the chip selects is used, the other must be tied to its active level (e.g., nCS would be tied to GND).
RDYnBUSY	I/O	PPA	Output	Ready output. A high output indicates that the FLEX 8000 device is ready to accept another byte of data. A low output indicates that the device is not ready to receive data.
CLKUSR	I/O	All	Input	Optional user-supplied Clock input. Synchronizes the initialization process.
ADD17 to ADD0	I/O	APD, APU	Outputs	Address outputs. Driven by the FLEX 8000 device to uniquely address up to 256 Kbytes of external configuration memory devices.

Table 9. Pin Functions (Part 2 of 2)

DATA7 to DATA0	I/O	APD, APU, PPA, PPS	Inputs	Data inputs. Byte-wide configuration data is presented to the FLEX 8000 device on all 8 data pins.
DATA0		AS, PS	Input	Data input. Bit-wide configuration data is presented to the FLEX 8000 device on the DATA0 pin.
DATA7		PPA	Output	In the PPA configuration scheme, the DATA7 pin presents the RDYnBUSY signal after the device receives an nRS strobe. Using the DATA7 pin may be more convenient than using the RDYnBUSY output pin.
SDOUT	I/O	All	Output	Reserved configuration output. Drives out during command mode.

effect regardless of the selected configuration scheme; others are specific to a particular configuration scheme. Table 9 summarizes the functionality of each configuration pin.

Notes:

(1) The maximum number of dual-purpose configuration pins that can be used as I/O pins in user mode varies in different configuration schemes:

AS:	3 pins	APU:	29 pins	APD:	29 pins
PS:	3 pins	PPS:	10 pins	PPA:	15 pins

(2) The internally generated DCLK signal that is used to configure FLEX 8000 devices with the AS, APU, APD, and PPA configuration schemes is available during user-mode operation if the *Enable DCLK Output in User Mode* configuration option bit is turned on. The DCLK signal can range from 2 to 6 MHz in frequency; the duty cycle and frequency are not guaranteed.

(3) An externally generated DCLK signal is used to configure FLEX 8000 devices with the PS and PPS configuration schemes. After configuration has finished, the external host can continue to drive the DCLK signal during user-mode operation.

Seven of the device pins are dedicated to the configuration process and cannot be used as I/O pins in user mode. Other configuration pins are dual-purpose pins that also can be used as I/O pins when the device operates in user mode. You can choose whether to use each dual-purpose pin as an I/O pin in user mode, as well as whether to force a dual-purpose pin to tri-state (i.e., drive a high-impedance logic level).

You can specify these settings for each pin on a device-by-device basis in MAX+PLUS II with the **FLEX 8000 Individual Device Options** dialog box. You can also enter global default pin settings for an entire project with the **FLEX 8000 Device Options** dialog box. Turning on the *Reserve* option for a specified pin in either dialog box prevents the pin from being used as an I/O pin during user mode; turning on the *Tri-State* option forces the pin to tri-state. A reserved pin should not be connected to any circuitry on the target board unless it is also tri-stated. Otherwise, the reserved pin will drive an unknown logic level that may cause logic contention with other signals on the board.

The nSTATUS, nCONFIG, CONF_DONE, and CLKUSR device configuration pins are available to monitor the configuration process and control how the device loads data, initializes, and enters user-mode operation. These pins can be used together with configuration option bits to provide additional configuration and initialization control.

nSTATUS Pin

The nSTATUS pin is an open-drain, bidirectional pin. When the FLEX 8000 device powers up, it pulls this pin low and then releases it within 100 ms. During configuration, the nSTATUS pin can be polled externally to verify that the FLEX 8000 device is being configured. If an error occurs during configuration, the nSTATUS pin is pulled and held low. In addition, if an external circuit pulls the nSTATUS pin low during either command-mode or user-mode operation, the FLEX 8000 device senses an error condition. After the pin is pulled low, configuration must be restarted.

Configuration is restarted with a high-low-high pulse on the nCONFIG pin. As an alternative, if the *Auto-Restart Configuration on Frame Error* option bit is turned on, the FLEX 8000 device can restart the configuration automatically when an error is detected. If this option bit is turned on, the nSTATUS pin is pulled low for a few microseconds and then released, indicating that the reconfiguration cycle has started. See Figure 3 earlier in this application note for an example of an AS configuration scheme that supports auto-reconfiguration.

If V_{CC} falls below an acceptable level during user-mode operation, the nSTATUS pin is pulled and held low, indicating an error condition. See "Configuration Reliability" later in this application note for more details.

nCONFIG Pin

The nCONFIG pin is a dedicated input that is used to start a configuration cycle. In most applications, the nCONFIG pin is tied to V_{CC}, directing the FLEX 8000 device either to immediately start configuration in an active configuration scheme, or to prepare immediately for configuration in a passive configuration scheme.

When the nCONFIG pin is held at GND, the FLEX 8000 device is reset and ready to start configuration. Configuration begins only after the pin is pulled up to V_{CC}. The nCONFIG pin can thus be held low to delay the configuration process and thus prevent data from loading until the desired time.

If an application requires a FLEX 8000 device to be reconfigured after system power-up, the nCONFIG pin must be tied to some external intelligent circuitry that monitors and controls that configuration process, as described in "In-Circuit Reconfiguration" earlier in this application note.

CONF_DONE Pin

The CONF_DONE pin is an open-drain, bidirectional pin that reflects the configuration status. When a FLEX 8000 device is ready to begin loading data, the CONF_DONE pin is pulled to GND and remains at GND while the data is loading, indicating that the FLEX 8000 device is being configured. After the last configuration data byte has been read, the CONF_DONE pin is released and pulled to V_{CC} by an external pull-up resistor, indicating that configuration is finished. The FLEX 8000 device interprets this low-to-high transition on the CONF_DONE signal as the command to initialize and enter the user mode.

If the CONF_DONE pin does not pull up to V_{CC} within ten Clock cycles of the final configuration data byte, the FLEX 8000 device detects an error condition, aborts the initialization process, and drives and holds the nSTATUS pin low. If the nSTATUS pin is low, it indicates either that an error has occurred in the application circuit, or that the configuration data-stream is corrupt.

The CONF_DONE pin can also be used to control the initialization process. You can disable error checking on the CONF_DONE net by turning on the device's *Disable Start-Up Time-Out* configuration option bit, so that the failure of CONF_DONE to pull to V_{CC} does not cause an error condition. The CONF_DONE network can then be driven by some external logic and held low until initialization is desired.

CLKUSR Pin

The CLKUSR pin can coordinate the initialization of multiple FLEX 8000 devices or synchronize the configuration of a FLEX 8000 device with other application logic in the system. In most applications, the FLEX 8000 device uses its internal oscillator (available externally as DCLK) to complete the initialization. After ten Clock cycles, the device enters user mode. You can turn on the *User-Supplied Start-Up Clock* configuration option bit and supply these ten Clock cycles on the CLKUSR pin to ensure that the device enters the user mode precisely when desired. Since the internal oscillators on all FLEX 8000 devices are not guaranteed to have the same frequency, you can use the CLKUSR pin to synchronize multiple FLEX 8000 devices in the same system.

MAX+PLUS II Configuration & Programming Support

The MAX+PLUS II software can generate four different types of configuration files for FLEX 8000 devices, as shown in Table 10. During project compilation, MAX+PLUS II automatically generates a POF and an SOF for each FLEX 8000 device. If necessary, you can generate a TTF or Hex File, as well as different POF(s), after compilation with the **Combine Programming Files** command (File menu) in the MAX+PLUS II Programmer or Compiler.

Table 10. FLEX 8000 Device Programming Files

File Type	Filename Extension	File Format	File Utilization
SRAM Object File	**.sof**	Binary	Downloaded directly into the FLEX 8000 device with the MAX+PLUS II Programmer using the FLEX 8000 Download Cable and Altera programming hardware.
Programmer Object File	**.pof**	Binary	Programmed into an Altera Configuration EPROM. The POF contains the configuration data, as well as the header, CRC, and pad bytes for configuring the FLEX 8000 device in an AS configuration scheme.
Hexadecimal (Intel-Format) File	**.hex**	ASCII text	Programmed into an industry-standard parallel EPROM. The Hex File contains the configuration data, as well as the header, CRC, and pad bytes for programming a parallel EPROM that configures a FLEX 8000 device in an APU or APD configuration scheme.
Tabular Text File	**.ttf**	ASCII text	A comma-separated version of the Hex File, used as source code in high-level programming languages. The TTF can be included in the source code for an intelligent host that configures the FLEX 8000 device in a PPA, PPS, or bit-wide PS configuration scheme. It can also be converted into an equivalent binary format that is directly loaded (LSB first) into the FLEX 8000 device.

Together, the MAX+PLUS II Programmer and Altera programming hardware provide the following capabilities:

- ❏ A POF can be programmed into an Altera Configuration EPROM for an AS configuration scheme.
- ❏ An SOF can be downloaded via the FLEX Download Cable for in-circuit PS configuration of a FLEX 8000 device.

For information on configuring FLEX 8000 devices with the BitBlaster, refer to the current *BitBlaster Serial Download Cable Data Sheet*.

Programming & Configuration Files

This section provides information on the characteristics of each type of configuration file. The process of creating different configuration files is described in "Combining & Converting Programming Files" later in this application note.

SRAM Object File

The SRAM Object File (**.sof**) is used during passive serial configuration when the data is downloaded directly into the FLEX 8000 device in-system with the MAX+PLUS II Programmer, the FLEX Download Cable, and Altera programming hardware. MAX+PLUS II automatically inserts the necessary header, formatting, and synchronization bits into the data stream when it downloads an SOF into a FLEX 8000 device. See "Configuring a FLEX 8000 Device In-System with MAX+PLUS II & the FLEX Download Cable" later in this application note for more information.

If configuration files are needed for other configuration schemes, MAX+PLUS II uses the data in SOF(s) to generate the appropriate POF(s), a TTF, or a Hex File.

Programmer Object File

The Programmer Object File (**.pof**) is used to program Altera Configuration EPROMs for an AS configuration scheme. MAX+PLUS II automatically generates a POF for every FLEX 8000 device in a project. In a multi-device project, each FLEX 8000 device has a dedicated Configuration EPROM. MAX+PLUS II selects the appropriate Configuration EPROM to most efficiently store the data for each FLEX 8000 device.

Hexadecimal (Intel-Format) File

The Hexadecimal File (**.hex**) is an ASCII file in the Intel Hex format. This file contains the configuration and formatting data for an industry-standard byte-wide parallel EPROM that is used to configure a FLEX 8000 device in an APU or APD configuration scheme. The data in the Hex File is interpreted by the programming software when it is programmed into a parallel EPROM.

The usual base address for FLEX 8000 configuration data is the origin of the EPROM. In some applications, the origin of the EPROM is required by other system resources, so some offset is necessary. In an APU configuration scheme, the FLEX 8000 device generates ascending addresses starting at 00000H; in an APD configuration scheme, it generates descending addresses starting at 3FFFFH. The FLEX 8000 device provides these base addresses for the configuration data during configuration, but any needed offset address must be generated externally, as shown earlier in Figure 7. The APU scheme is appropriate if the FLEX 8000 configuration data can be stored at the beginning of an EPROM or at some known offset in an EPROM larger than 256 Kbytes. The APD scheme is appropriate if the FLEX 8000 configuration data is placed in an EPROM in which the low addresses are not available (as shown in Figure 8), or in an EPROM that

also stores other information that is expected to increase as an application evolves.

Tabular Text File

The Tabular Text File (**.ttf**) is a tabular ASCII file that provides a comma-separated version of the configuration data for the PPA, PPS, and bit-wide PS configuration schemes. In some applications, the storage device that contains the FLEX 8000 configuration data is neither dedicated to nor connected directly to the FLEX 8000 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 8000 configuration data as part of the source code for the intelligent host (using "include" or "source" commands). The host can access this data from an EPROM or a mass-storage device and load it into the FLEX 8000 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.

If you do not include the TTF in the source code for an intelligent host, the file's comma-separated ASCII representation of the binary data must be converted into its equivalent 8-bit binary format (e.g., 85 would become 01010101) before it is loaded into the FLEX 8000 device. Data must be stored so that the least significant bit (LSB) of each byte of data is loaded first. The Altera Applications bulletin board service (BBS) provides the **ttf2rbf** conversion utility for this purpose. The converted binary image can be stored on a mass storage device. The intelligent host can then read data from the binary file and load it into the FLEX 8000 device. You can also use the intelligent host to perform real-time conversion during configuration. In the PPA and PPS configuration schemes, the FLEX 8000 device receives its information in parallel from the data bus, a data port on the CPU, or some other byte-wide channel. In the bit-wide PS configuration scheme, the data is shifted in serially.

Programming a Configuration EPROM

You can program Altera Configuration EPROMs with MAX+PLUS II, the PL-MPU Master Programming Unit, and the appropriate Configuration EPROM programming adapter. The PLMJ1213 adapter programs Configuration EPROMs in 8-pin plastic dual in-line packages (PDIP) and 20-pin plastic J-lead chip carrier (PLCC) packages; the PLMT1064 adapter programs Configuration EPROMs in 32-pin thin quad flat pack (TQFP) adapters.

To program an Altera Configuration EPROM:

1. Choose the **Programmer** command (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, 20-pin J-lead, or 32-pin QFP socket on the programming adapter. The socket for the FLEX 8000 device (if any) must be empty.
4. Choose the **Program** button.

After successful programming, you can place the Configuration EPROM on the target board to configure a FLEX 8000 device in the AS configuration scheme.

Configuring a FLEX 8000 Device In-System with MAX+PLUS II & the FLEX Download Cable

To configure a FLEX 8000 device with the FLEX Download Cable:

1. Connect the FLEX Download Cable to the 9-pin D-type connector on a Configuration EPROM programming adapter.
2. Connect the other end of the FLEX Download Cable to the 10-pin male header on the target board.
3. Start MAX+PLUS II and choose the **Programmer** command (MAX+PLUS II menu) to open the Programmer window.
4. Choose the **Select Programming File** command (File menu).
5. Select the desired SOF filename in the *Files* box or type a name in the *File Name* box. If you choose a programming file from another project, you are asked if you wish to change the current project name.
6. Choose **OK**.
7. Choose the **Program** button to configure the device.

After the device is configured and initialized, it enters user mode and operates as a logic device. The FLEX Download Cable is electrically removed from the circuit and does not influence circuit operation. You can also physically disconnect the FLEX Download Cable without disturbing the FLEX 8000 configuration data or device operation.

Combining & Converting Programming Files

MAX+PLUS II automatically generates a POF and an SOF for every FLEX 8000 device in a project, as described earlier in this application note. The POF can be programmed into an Altera Configuration EPROM used in an AS configuration scheme; by default, each FLEX 8000 device has one

dedicated Configuration EPROM (two Configuration EPROMs are required for an EPF81500 device).

You may wish to combine and/or convert the automatically generated SOFs into a different format for the following purposes:

- ❏ To use a configuration scheme other than AS. You must convert an SOF into a Hex File or a TTF for programming a parallel EPROM, BIOS EPROM, or another data source.
- ❏ To combine multiple sets of configuration data to be used for in-circuit reconfiguration in any configuration scheme.

To convert an SOF into a Hex File or TTF:

1. Refer to Table 2 to calculate the required data space in a parallel or serial data source.
2. Choose the **Combine Programming Files** command (File menu) in the MAX+PLUS II Programmer or Compiler.
3. Select the desired SOF name in the *Files* box or type a name in the *File Name* box under *Input Files*. Choose the **Add** button to add it to the *Selected Files* box.
4. Specify information for the desired configuration scheme:

   – If the FLEX 8000 device will be configured with a parallel EPROM in the APU or APD configuration scheme, select *.hex (Single-Device)* in the *File Format* drop-down list box under *Output File*.

     In addition, if the FLEX 8000 configuration data will not start at the origin of the EPROM, specify the base address for the configuration data in the *Address* box under *Input Files.* Choose *Up* or *Down* under *Count* to specify whether the FLEX 8000 device should count up or down. The counting sequence can be either ascending (00000H to 3FFFFH) for APU configuration or descending (3FFFFH to 00000H) for APD configuration, as described in "Hexadecimal (Intel-Format) File" earlier in this application note.

   or:

   – If the FLEX 8000 device will be configured with a PPA, PPS, or bit-wide PS scheme, select *.ttf (Sequential)* in the *File Format* drop-down list box under *Output File*. The TTF can be incorporated as source code for a data structure in a high-level programming language. Otherwise, the TTF data must be converted into its equivalent 8-bit binary format before it is loaded into the FLEX 8000 device, as described in "Tabular Text File" earlier in this application note.

5. The default name for the output file is the current project name plus the extension **.hex** or **.ttf**. To give a different name to the file, type a name in the *File Name* box under *Output File*.
6. Choose **OK** to generate the Hex File or TTF. The file is placed in the current project directory.

You can use in-circuit reconfiguration to load multiple sets of configuration data into the FLEX 8000 device in a system. The following procedure describes how to combine SOFs for in-circuit reconfiguration of a FLEX 8000 device.

To combine SOFs for in-circuit reconfiguration with multiple sets of configuration data:

1. Refer to Table 2 to calculate the required data space in a parallel or serial data source.
2. Choose the **Combine Programming Files** command (File menu) in the MAX+PLUS II Programmer or Compiler.
3. Select the SOF with the first set of configuration data and choose the **Add** button to add it to the *Selected Files* box.
4. Repeat step 3 until all SOFs have been added to the *Selected Files* box.
5. Arrange the selected files in the order in which the different sets of configuration data will be used by selecting each SOF filename and choosing the **Up** or **Down** button under *Order*.
6. Specify information for the desired configuration scheme, select the output filename, and choose **OK**. The default name for the output file is the current project name plus the extension **.hex**, **.ttf**, or **.pof**. To give a different name to the file, type a name in the *File Name* box under *Output File*. If multiple POFs are generated, they are uniquely identified by a sequence number appended to the filename (e.g., the first is **device.pof**, the second is **device1.pof**, etc.). You can specify an output filename that has less than the maximum of eight characters to leave room for the numerical index; otherwise, the last character(s) are truncated to include it.

Configuration Reliability

The FLEX 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 check (CRC) circuitry is used to validate every data frame (i.e., sequence of data bits) as it is loaded into the FLEX 8000 device. If the CRC generated by the FLEX 8000 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. This

CRC circuitry ensures that noisy systems will not cause errors that yield an incorrect or incomplete configuration.

The FLEX architecture also provides a very high level of reliability in low-voltage brown-out conditions. The SRAM cells require a certain V_{CC} level to maintain accurate data. Since this voltage threshold is significantly lower than that required to activate the power-on reset (POR) circuitry in the FLEX 8000 device, the FLEX 8000 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, reconfiguration begins as soon as V_{CC} returns to an acceptable level if the `nCONFIG` pin is tied to V_{CC}. Otherwise, the host system must start the reconfiguration process.

These device features ensure that FLEX 8000 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 families of Altera programmable logic devices.

Revision History

The information contained in the *Application Note 33 (Configuring FLEX 8000 Devices)* version 3.03 supersedes information published in previous versions.

Version 3.03 Changes

Application Note 33 (Configuring FLEX 8000 Devices) version 3.03 contained the following changes:

- Updated text on page 54.
- Updated text on page 65.

Version 3.02 Changes

Application Note 33 (Configuring FLEX 8000 Devices) version 3.02 contains the following changes:

- Updated information under the "Bit-wide Passive Serial Configuration" section on page 54.
- Updated information under the "Programmer Object File" section on page 65.

Version 3.01 Changes

Application Note 33 (Configuring FLEX 8000 Devices) version 3.01 contained the following changes:

- Updated Figure 1 on page 37.
- Updated Figure 3 on page 40.
- Updated Figure 4 on page 41.