Configuring FLASHlogic Devices

April 1995, ver. 1 Application Note 45

Introduction

The Altera FLASHlogic family of programmable logic devices (PLDs) is based on CMOS technology with SRAM configuration elements. This technology supports in-circuit reconfigurability (ICR) via the Joint Test Action Group (JTAG) interface without requiring any special programming voltages. FLASHlogic devices can be reconfigured with the FLASHlogic Download Cable or an intelligent host.

Each FLASHlogic device has on-board, non-volatile FLASH or EPROM memory that stores a single configuration. An external controller, such as a microcontroller or state machine, is required for reconfiguration. When the device is powered up, the configuration information is written into the on-board SRAM and the device is ready to perform its intended function.

The FLASH or EPROM memory is programmed with Altera's PENGN software, which is not discussed in this application note. For more information on programming FLASHlogic devices, refer to the ***PLDshell Plus/PLDasm User's Guide V5.0***, available from Altera Literature at (408) 894-7144.

JED2JTAG Operation

To reconfigure a FLASHlogic device, you must first use Altera's PLDshell Plus software or third-party software to create a JEDEC File (**.jed**) that contains the FLASHlogic device configuration information. You can then process the file with Altera's JED2JTAG software and use it to reconfigure the device. The JED2JTAG software is included with PLDshell Plus.

The Altera MAX+PLUS II development system provides programming-only support for FLASHlogic devices. Full device support is planned for the second half of 1995.

JED2JTAG converts a JEDEC File into one of three different output formats: a Bit File (**.bit**), a Binary File (**.bin**), or a Hexadecimal (Intel-Format) File (**.hex**). The output format you choose depends on the type of controller used to reconfigure the device. The controller then uses the selected output format to send the configuration data to the FLASHlogic device via the JTAG interface. Table 1 describes the JED2JTAG input files.

Go to the ***PLDshell Plus/PLDasm User's Guide V5.0*** for additional information on the FLASHlogic Download Cable, the JED2JTAG software, and FLASHlogic device configuration.

Table 1. JED2JTAG Input Files	
File	**Description**
String Description File (**.sdl)**	Describes the JTAG chain, which is the series of connected devices on a circuit board.
<*file name*>**.jed**	JEDEC File(s) generated with PLDshell Plus software can be used to configure the FLASHlogic devices.
jtag.dvc	Library file, included with the PLDshell Plus software, that contains a description of several common JTAG devices. You can easily add information for other JTAG devices from device data sheets or from the appropriate Boundary Scan Description Language File (BSDL). Instructions on adding devices are included in the **jtag.dvc** file.

The JED2JTAG software produces one of the output formats shown in Table 2.

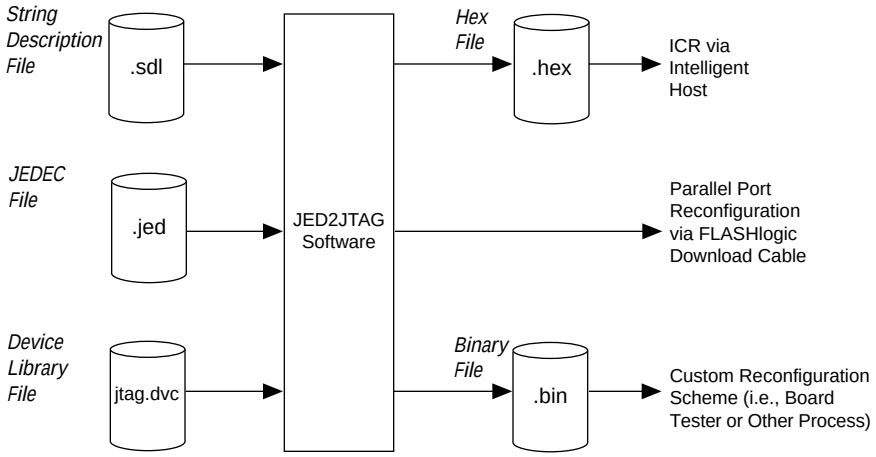
Table 2. JED2JTAG Output Files	
File	**Description**
<*file name*>**.bit**	JED2JTAG intermediate file. JED2JTAG creates one Bit File for each JEDEC File, then compares the time stamps. If the JEDEC File is more recent, JED2JTAG recreates the Bit File.
<*project name*>**.bin**	Bin File containing the JED2JTAG stream of JTAG signals. This file is used to create a Hex File or is sent to the computer's parallel port. The Bin File is created from Bit Files.
<*project name*>**.hex**	Hex File used for programming standard memory devices.

To start the JED2JTAG software, type the following command at the DOS prompt:

`jed2jtag` *<project name>* ↵

Figure 1 shows a block diagram of JED2JTAG operation.

Figure 1. JED2JTAG Block Diagram



The String Description File lists the JTAG device chain and any associated JEDEC Files, as shown in the **example.sdl** file in Figure 2.

Figure 2. String Description File (example.sdl)

In this example, a pipe character (|) indicates a comment line.

```
|FILE: example.sdl -Simple Prototype Board
{
|          Port_Num  Port_Type
STRING     3         BIN_FILE
| Port_Num 1 - for LPT1:
| Port_Type must be PARALLEL_PORT or BIN_FILE
|       Loc.  Ref   Device       JEDEC File
DEVICE  0     U3    EPX780QC132  RIGHT.JED
DEVICE  1     U4    EPX780LC84   LEFT.JED
DEVICE  2     U1    TI_74BCT8373
}
```

The `STRING` instruction in the String Description File controls which operations JED2JTAG uses for the JTAG chain. To create a Hex File or a Bin File, specify the following:

`STRING 3` *<file type>*

where `3` specifies the configuration option that enables you to generate a file, and *<file type>* is either `HEX_FILE` or `BIN_FILE`.

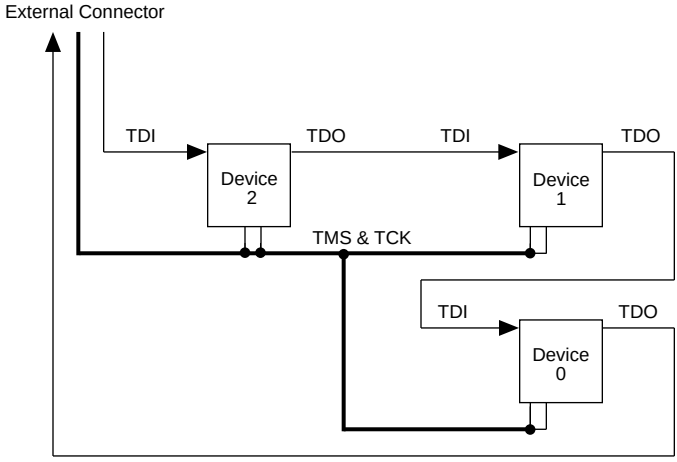
To send the configuration via the PC's parallel port, you must specify parallel port instructions in the String Description File as follows:

`STRING` <*parallel port*> `PARALLEL_PORT`

where <*parallel port*> is `1` to specify LPT1 and `2` to specify LPT2.

The **example.sdl** file shown in Figure 2 describes the JTAG device chain shown in Figure 3. In this example, Device 2 does not have a JEDEC File, and will not be reconfigured by JED2JTAG.

Figure 3. JTAG Chain (example.sdl)



Configuring FLASHlogic Devices Using ICR

In-circuit reconfigurability (ICR) enables you to reconfigure devices that are already soldered onto a printed circuit board. For example, you can use the FLASHlogic Download Cable or an intelligent host to change the logic in add-on card applications that interface with different buses.

In-Circuit Reconfiguration with the FLASHlogic Download Cable

If a printed circuit board has a JTAG chain with a header, you can use the FLASHlogic Download Cable to reconfigure devices in-circuit.

To use the FLASHlogic Download Cable:

1. Connect one end of the cable to your PC's parallel port, and connect the other end of the cable to the JTAG header on your printed circuit board.

2. Create the JEDEC File(s) for the design.

3. Create a String Description File that describes the JTAG chain. Be sure to include the appropriate parallel port `STRING` instruction.

4. Run JED2JTAG. This software generates a Bin File and downloads it to your board to reconfigure the FLASHlogic device(s).

In-Circuit Reconfiguration with an Intelligent Host

When you reconfigure a device using an intelligent host, your system must have the appropriate storage capability, such as ROM, external peripherals, or disk drives. To reconfigure a device with an intelligent host:

1. Create the JEDEC File(s) for the design.

2. Create a String Description File that describes the JTAG chain. Be sure to include the appropriate Hex File `STRING` instruction.

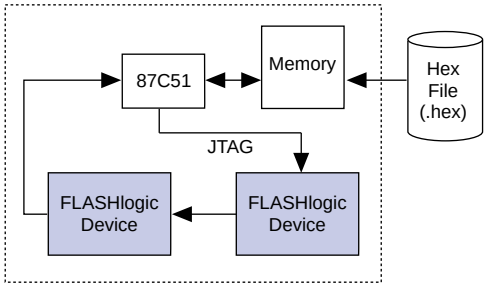
3. Run JED2JTAG to create the Hex File.

4. Program the storage memory with the Hex File.

5. Develop software using the guidelines specified by your microcontroller manufacturer to download the configuration data from the storage device into the FLASHlogic device(s).

You can use almost any combination of CPU and memory to reconfigure FLASHlogic devices in-circuit. For example, you can use an Intel 87C51 microprocessor as an intelligent host. See Figure 4.

Figure 4. Implementing ICR with an Intelligent Host



The interconnect between the CPU and the JTAG chain requires four I/O signals (TMS, TCK, TDI, and TDO), plus a RESET signal, if necessary. See Figure 5.

Figure 5. Sample Schematic for Implementing ICR with an Intelligent Host

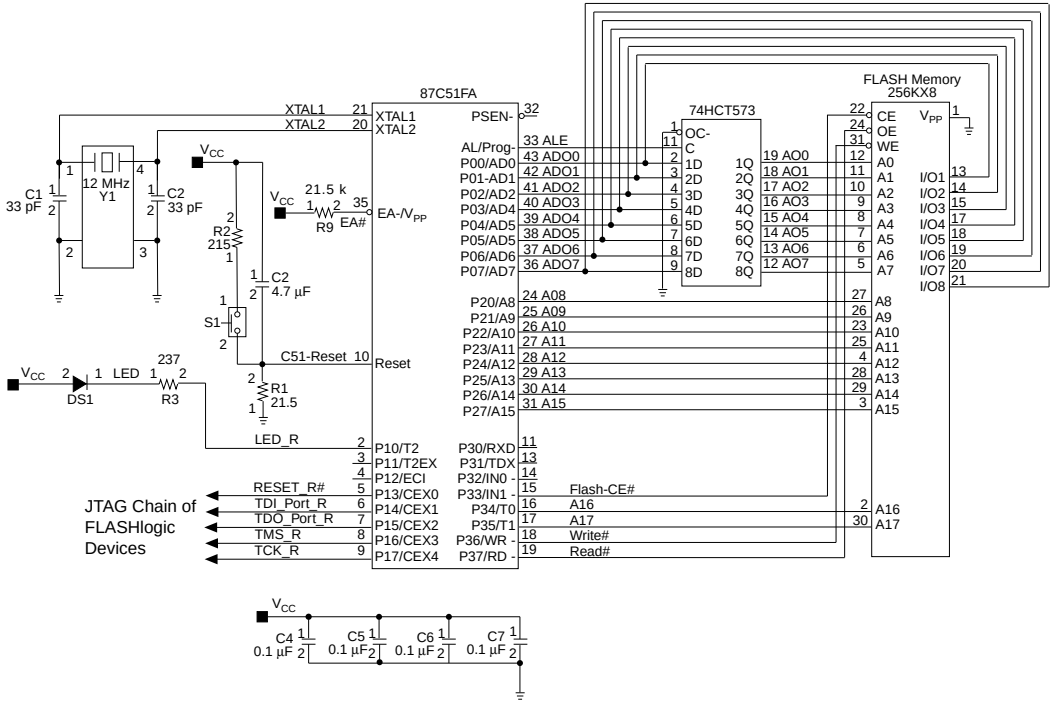


Figure 6 shows the code used by the 87C51 microprocessor to control the JTAG chain.

Figure 6. Hex File Download Algorithm for the 87C51 (Part 1 of 2)

```
;*********************************************************************
;* JTAG STAND-ALONE LOADER CODE
;**
;* This program usually executes in an embedded controller from an
;* executable memory bank.  It transmits data from a local nonvolatile
;* data memory bank to a JTAG IEEE 1149.1 chain, typically to load the
;* SRAM control memory in the FLASHlogic devices in the chain.
;**
;* Data is stored in the non-volatile memory off-line, e.g., with a PROM
;* programmer.  It is formatted as a JTAG data stream.  Each byte
;* contains TDO/TMS data for 4 TCK cycles. Bit 0 of each byte is the
;* first TDO, bit 1 is the first TMS, bit 2 is the second TDO, etc.
;* No inversion occurs between memory and TMS/TDO.
;**
;* The first four bytes in the non-volatile memory specify the Ending
;* Address + 1. The least significant byte is at data address 0. The
;* JTAG data stream starts at data address 4.
;**
;* Data is automatically downloaded each time the embedded controller
;* is reset, including power-on.
;*********************************************************************
; Initialization
SET LED# port bit ; turn off LED
DELAY (500) ; delay 500 ms.
; Give time for host JTAG string to completely power-up before
beginning.
CLR TICK port bit ; init TCK to LOW
CLR RESET# port bit ; init RESET# to LOW (activate host RESET)
CLR CE# port bit ; enable FLASH memory
CLR LED# port bit ; turn on LED
; Transmit data
adrs = 0; initialize data address and
last-adrs = (adrs) ; last address + 1 (these are 32 bit values)
adrs = adrs + 4;
WHILE (adrs < last_adrs)
( ; loop until all data is moved from memory to JTAG chain
; output lst TMS/TDO
MOVE bit 0 of (adrs) to TDO port bit
MOVE bit 1 of (adrs) to TMS port bit
SET TCK port bit ; pulse TCK
CLR TCK port bit
```

Figure 6. Hex File Download Algorithm for the 87C51 (Part 2 of 2)

```
; output 2nd TMS/TDO
MOVE bit 2 of (adrs ) to TDO port bit
MOVE bit 3 of (adrs) to TMS port bit
SET TCK port bit ; pulse TCK
CLR TCK port bit
; output 3rd TMS/TDO
MOVE bit 4 of (adrs) to TDO port bit
MOVE bit 5 of (adrs) to TMS Port bit
SET TCK port bit ; pulse TCK
CLR TCK port bit
; output 4th TMS/TDO
MOVE bit 6 of (adrs) to TDO port bit
MOVE bit 7 of (adrs) to TMS port bit
SET TCK port bit ; pulse TCK
CLR TCK port bit
adrs = adrs + 1
)
; Completion
SET RESET# port bit ; release host
SET LED# port bit ; turn off LED
SET CE# port bit ; disable memory to save power
SLEEP ; disable embedded controller to save power
```

Conclusion

Altera's FLASHlogic family of PLDs provides the features and flexibility designers need for many of today's applications. These devices can be reconfigured in-circuit, allowing quick and easy design iterations and field upgrades.

2610 Orchard Parkway
San Jose, CA 95134-2020
(408) 894-7000
Applications Hotline:
(800) 800-EPLD
Customer Marketing:
(408) 894-7104
Literature Services:
(408) 894-7144