Dynamic Reconfiguration of Excalibur Devices with Linux

June 2003, ver. 2.0 Application Note 278

Introduction

This application note describes a remote dynamic reconfiguration of Excalibur™ devices with Linux demonstration. The demonstration runs on an EPXA1 development board and has the following features:

- An Apache Web Server that runs as an application in the operating system's file system
- A web page hosted on the board that allows you to upload a **.tar** file to the board containing an FPGA hardware image, a driver for this hardware, and an application to interface to the hardware
- **.tar** files that contain a simple peripheral I/O to enable the LEDs to be controlled from software, and an LCD driver to enable characters to be printed on the LCD panel

The demonstration shows that the Excalibur processor subsystem can run an operating system without involving the FPGA. This allows the FPGA to be reprogrammed with new hardware at any time, as required by an application. The full processing power of the ARM® embedded processor is not used by this demonstration, emphasizing that more complicated tasks can be run at the same time. The demonstration also shows the non-invasive remote FPGA reconfiguration capability that is unique to Excalibur devices. This demonstration does not use the full processing power of the ARM® embedded processor—it is a simple platform on which far more complicated tasks can be run.

The demonstration shows you can embed information about the configuration of the FPGA hardware into the FPGA configuration file. Embedding configuration information enables the device drivers associated with the hardware to be loaded when the FPGA is programmed, and consequently the drivers for the old hardware can be unloaded. In addition, the device drivers can be configured with the parameters embedded in the file, such as the base address of each peripheral.

For more information on rebuilding the dynamic reconfiguration of Excalibur devices with Linux demonstration, refer to the *AN276: Rebuilding the Excalibur Dynamic Reconfiguration with Linux Demonstration*.

Getting Started

The following sections explain how to set up the EPXA1 development board and run the demonstration software. The *Excalibur Device Applications* CD-ROM contains the software required to run the demonstration.

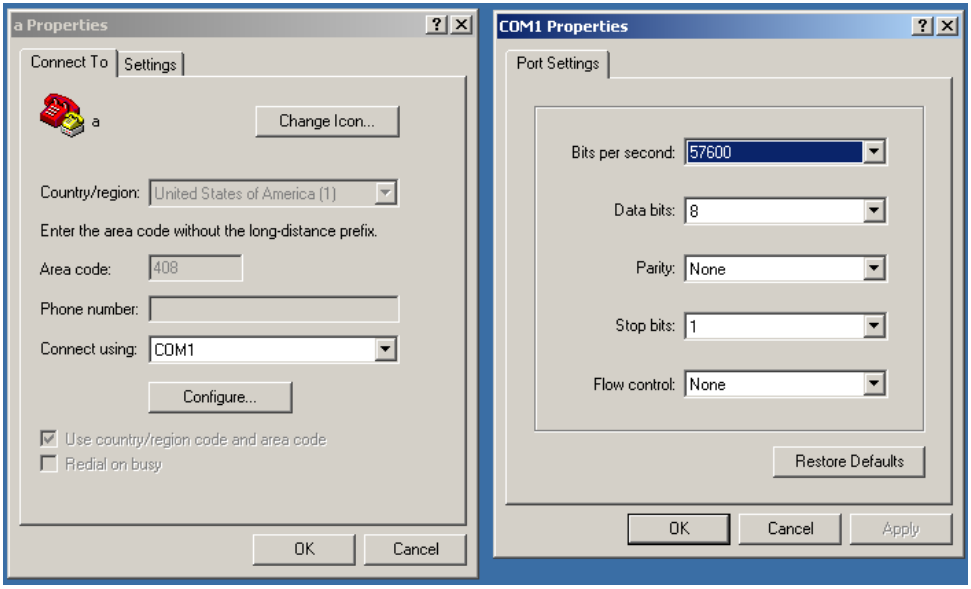
Board Connections

To start the Linux/remote reconfiguration demonstration, make the following board connections:

1. Connect one end of the null-modem serial cable to the DB9 connector (P2) on the EPXA1 development board. Connect the other end into an available COM port on your PC.

2. Start the HyperTerminal program on the PC and configure it to use the appropriate COM port with the settings shown in Figure 1.

Figure 1. HyperTerminal COM1 Properties



3. Connect the ByteBlasterMV™ download cable into the 10-pin header on the EPXA1 development board (labeled JTAG). Connect the other end of the ByteBlasterMV cable into a parallel port extension cable that is connected to an available parallel port on your PC.

4. Connect the board to a network by plugging a standard RJ11 Ethernet cable between connector RJ1 on the EPXA1 development board and an activated RJ11 network port. A crossover Ethernet cable can also be used to connect the board directly to a PC.

5. Connect the LCD module's ribbon cable to header J4 of the EPXA1 development board. Ensure that the orientation of the ribbon cable is such that the cable lies toward the power connector (J1).

6. Connect power to the board by plugging the barrel connector of the power-supply module into connector J1 on the EPXA1 development board. Plug the outlet cable appropriate for your region into an available wall outlet to power up the board.

Downloading the Image to Flash Memory

To download the Linux/remote reconfiguration demonstration into the EPXA1 development board flash memory, follow the steps below:

1. Open a Command Prompt window on your PC.

2. Change to the <*Excalibur Applications CD dir*>**reference_designs****linux_reconfig** directory, where the demo's .**hex** files are installed.

3. Type

   ```
   exc_flash_programmer –f –p –v –e 0 flashdisk.hex0↵
   ```

   to load the flash device on EBI0 with its portion of the image.

   Flash memory programming takes several minutes; its progress is displayed in the command window.

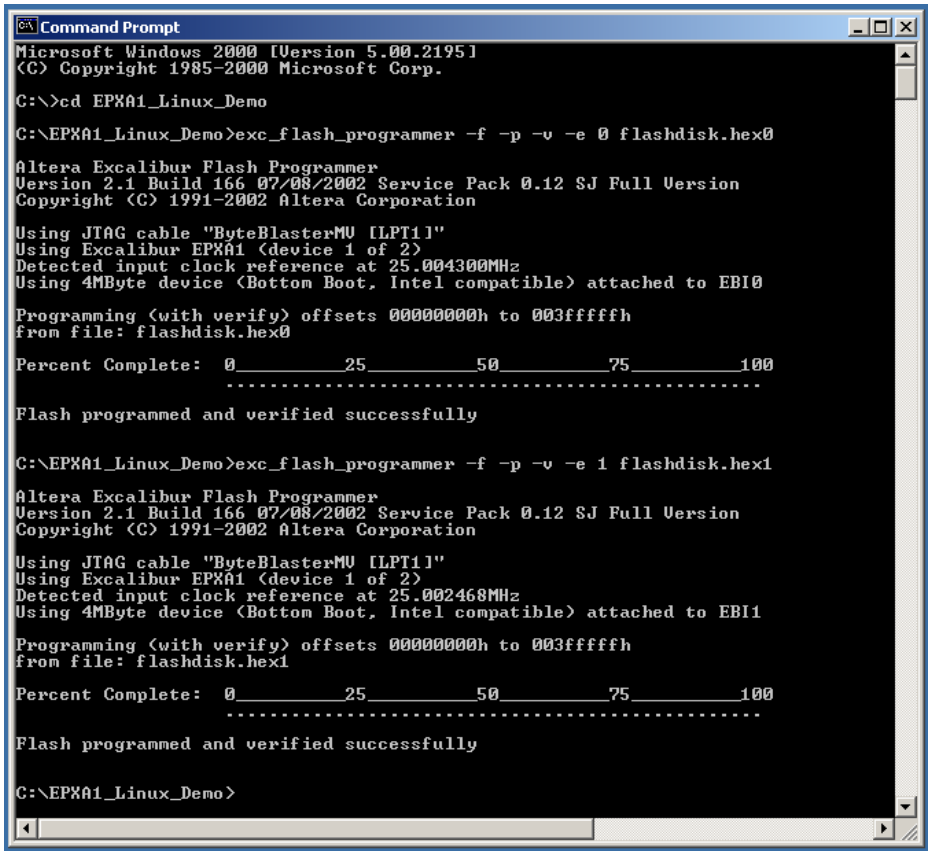
4. Type

   ```
   exc_flash_programmer –f –p –v –e 1 flashdisk.hex1↵
   ```

   to load the flash memory device on EBI1 with its portion of the image.

Figure 2 on page 4 shows the output from the flash programming commands.

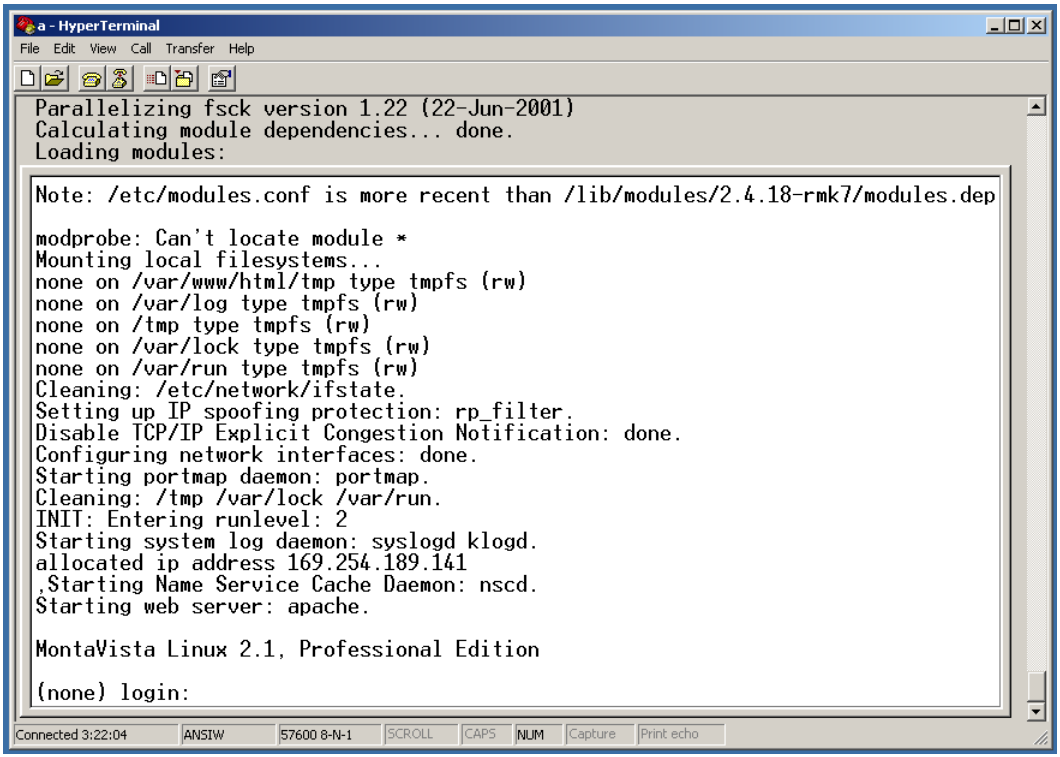
Figure 2. Output from the Flash Programming Commands



Booting the Board

When the flash memory has been loaded with the images, you can boot the system by pressing the reset button, Reset, on the EPXA1 development board. You can view the text output of the boot process in the HyperTerminal window. Booting takes approximately four minutes and is finished when the HyperTerminal windows displays the login prompt shown in Figure 3 on page 5.

Figure 3. Text Output from the Boot Process



Logging into the System

At this point, you can log in to the Linux system, and run Linux commands such as `ls` and `cd`. To log in, enter the user name `root`; no password is required.

IP Address

When the system boots, it attempts to obtain an IP address from the network using DHCP. If no DHCP server can be found within 60 seconds, the system assigns itself a random IP address from the subnet 169.254.0.0, having probed the network to ensure that the address selected is not already in use. The IP address obtained, either by DHCP, or by random, is displayed on the connected LCD module.

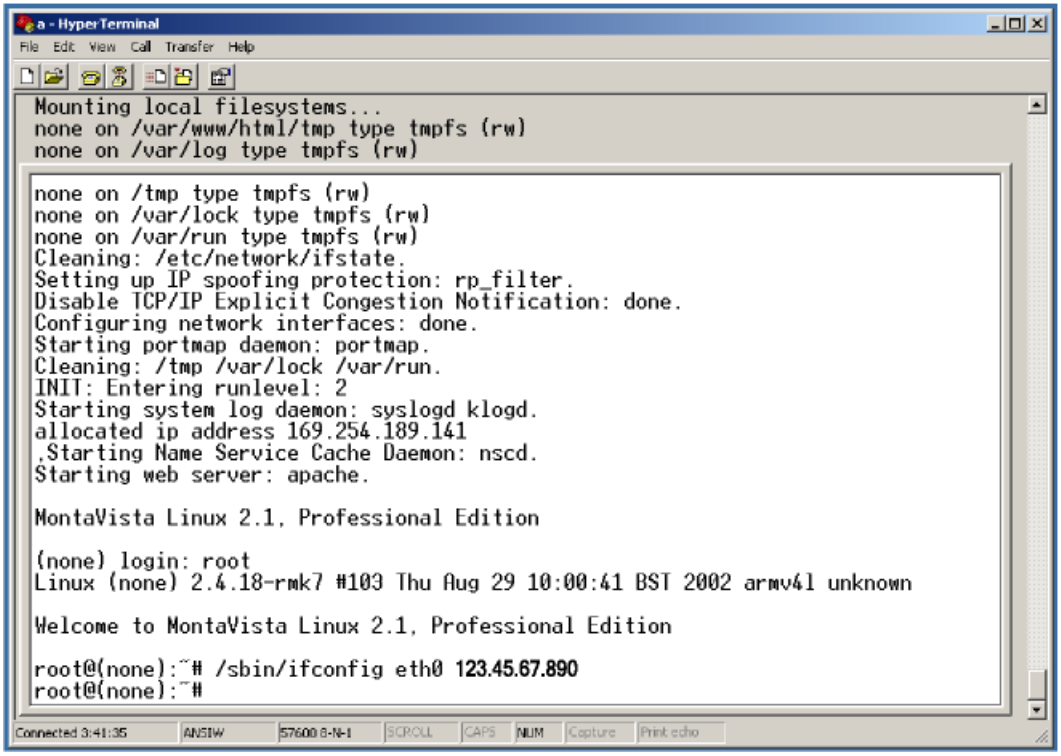
The default subnet used by the board is the same default subnet that a windows PC uses if it is set to use DHCP but fails to find a DHCP server. This ensures that a PC and a board connected through a crossover Ethernet cable are on the same subnet.

To assign the board a specific IP address manually, log in to the system and then type:

```
/sbin/ifconfig eth0 <ip-address>
```

Figure 4 shows an example of how to assign an IP address to the board manually.

Figure 4. Assigning an IP Address Manually



Connecting to the Web Server

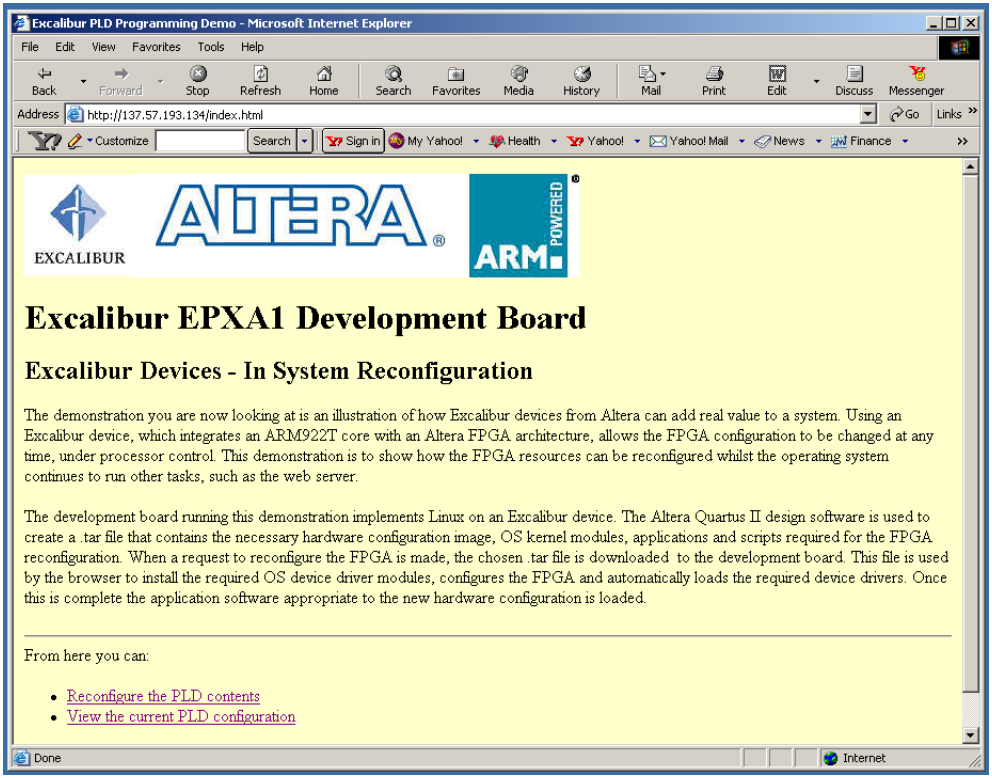
When the board boots, it starts to run a web server. Connect to the web server as follows:

1. Open a web browser on a PC that is either connected to the same network as the board, or is directly connected to it with a crossover Ethernet cable

2. Enter the IP address of the board into the **Address** field of the web browser, http://<ip-address-of-board>

   The web page shown in Figure 5 should appear in the web browser.

Figure 5. Web Page



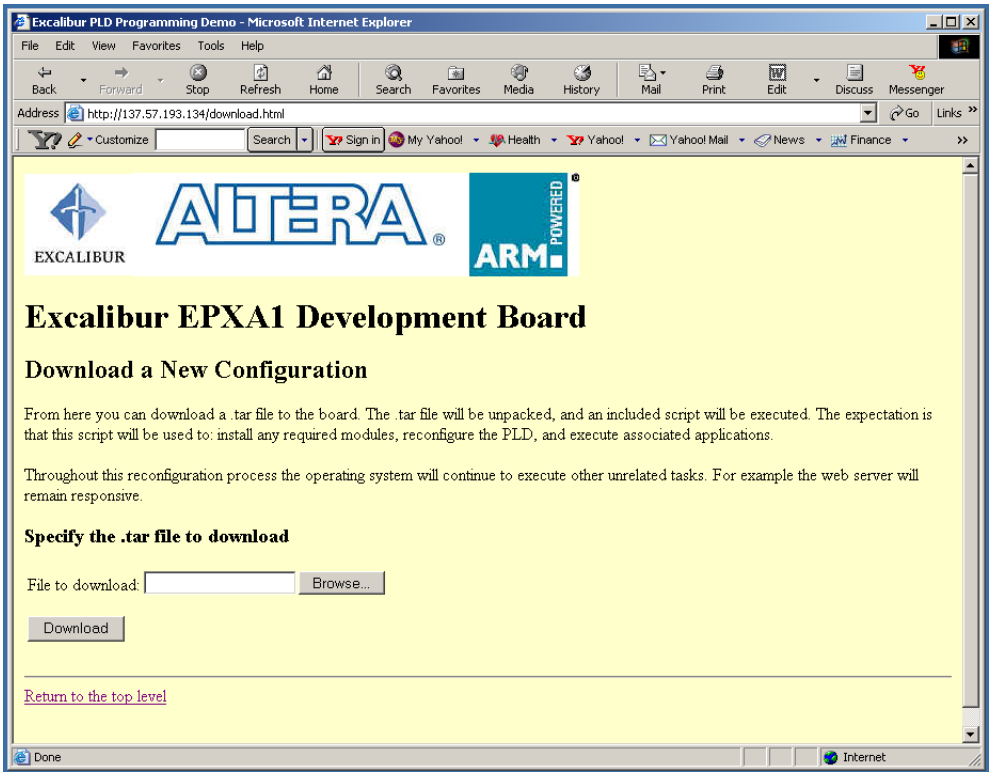
Downloading a New Image Remotely

When the board is connected to the website, a new configuration can be downloaded to the board over the network. Two pre-made configuration images have been included with this demo, **lcd.tar**, and **led.tar**.

To reconfigure the board with a new image, follow the steps below:

1. Click on **Reconfigure the PLD contents** in the web page displayed. The web page shown in Figure 6 should appear.

Figure 6. Download a New Configuration Web Page



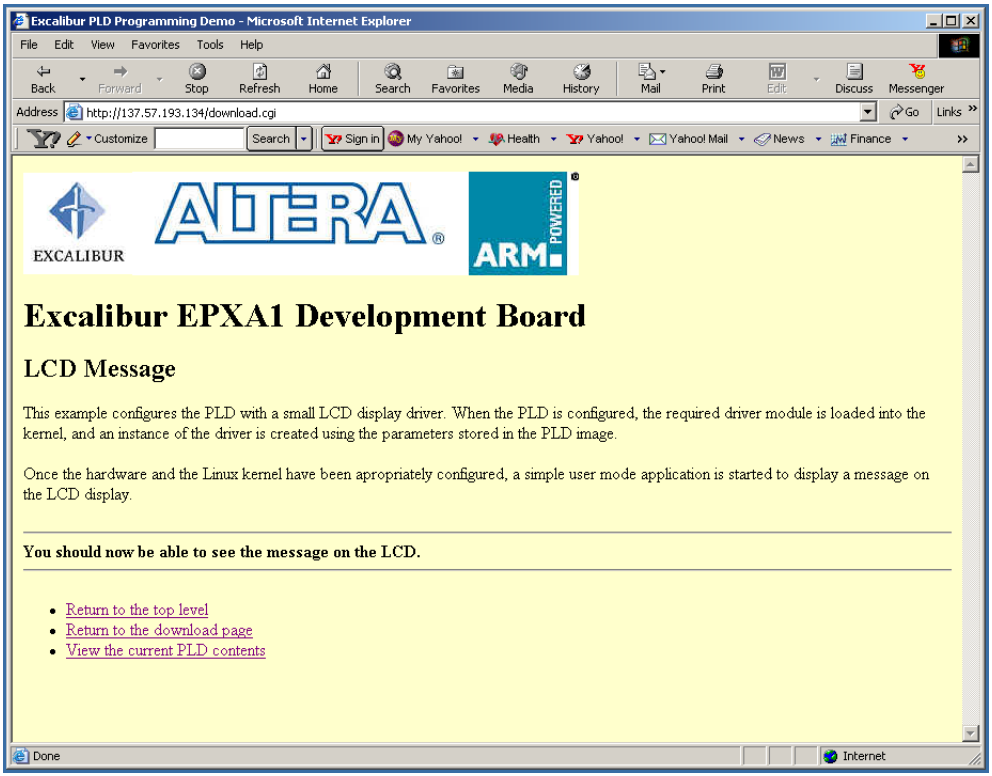
2. Click the **Browse** button, browse to either the **\lcd** or **\led** directory to find the appropriate **.tar** reconfiguration image, and select it.

3. Click the **Download** button to download the image to the system and reconfigure it.

The design corresponding to the image you selected begins to run on the board. The image **lcd.tar** presents a flashing message on the connected LCD module. The image **led.tar** scrolls the LEDs on the board.

After the system is reconfigured, a new web page is displayed, with a short description of the design with which the PLD was just configured. Figure 7 shows the page displayed after the system is reconfigured with **lcd.tar**.

Figure 7. Page Displayed Following Reconfiguration with lcd.tar



Implementation

The demonstration is based around MontaVista Linux running on the EPXA1 development board—Monta Vista Linux is due for release in Q4 2002. For this demonstration the board also runs the Apache Web Server. The two .**tar** files supplied each contain the following items:

1. A .**sbi** FPGA configuration file, containing the FPGA hardware as well as a header describing the software contents.
2. The kernel driver(s) associated with this hardware.
3. A user application.
4. A stop script to be run when the current .**tar** file is to be replaced.
5. A start script to be run when the .**tar** file is first downloaded.

The system relies upon the concept of kernel mode modules within Linux, which allows device drivers to be loaded at run time. This feature is not unique to Linux, so the concept could be expanded to other software systems.

The .**tar** file is downloaded to a RAM disk, which is located in the **/var/www/html/tmp** directory.

To start a demonstration, perform the following steps:

1. Run the stop script for the previously-loaded hardware, to kill the old application.
2. Decompress and unpack the relevant .**tar** file.
3. Run the start script.

The start script performs the following operations:

1. Installs the drivers.
2. Programs the .**sbi** file into the FPGA. This is done using a user-mode application called **pld_config**. **pld_config** parses the .**sbi** file and then passes it over to a kernel mode device driver that actually programs the FPGA, as well as loading and unloading the drivers.
3. Launches the user application.

For more information about the cgi script, see **/var/www/html/download.cgi**

Not all **.sbi** files contain a header describing the hardware contained within the file. This information has to be added using the Quartus® II software utility **sbi_config**.

Additional Information

This section identifies other sources of information about the EPXA1 development board.

For details of the EPXA1 development board, refer to the *EPXA1 Development Board Hardware Reference Manual*.

For details on the Excalibur toolflow, refer to the *EPXA1 Development Board Getting Started User Guide*.

For information on the source files for this demonstration, contact your Altera representative.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
lit_req@altera.com