12. Read-Only Zip File System

NII52012-9.1.0

Introduction

Altera provides a read-only zip file system for use with the hardware abstraction layer (HAL). The read-only zip file system provides access to a simple file system stored in flash memory. The drivers take advantage of the HAL generic device driver framework for file subsystems. Therefore, you can access the zip file subsystem using the ANSI C standard library I/O functions, such as `fopen()` and `fread()`.

The Altera® read-only zip file system is provided as a software package. All source and header files for the HAL drivers are located in the directory *<Nios II EDS install path>***/components/altera_ro_zipfs/HAL/**.

Using the Zip File System in a Project

The read-only zip file system is supported by both Nios® II software development flows. You need not edit any source code to include and configure the file system. To use the zip file system, you use the Nios II development tools to include it as a software package for the board support package (BSP) project.

You must specify the following four parameters to configure the file system:

- The name of the flash device where you wish to program the file system.
- The offset in the address space of this flash device.
- The name of the mount point for this file subsystem in the HAL file system. For example, if you name the mount point **/mnt/zipfs**, the following code opens a file in the zip file:

  ```
  fopen("/mnt/zipfs/hello", "r");
  ```

  This code, called from within a HAL-based program, opens the file **hello** for reading.
- The name of the zip file you wish to use.

The next time you build your project after you make these settings, the Nios II development tools include and link the file subsystem in the project. After you rebuild the project, the **system.h** file reflects the presence of this software package in the system.

Preparing the Zip File

The zip file must be uncompressed. The Altera read-only zip file system uses the zip format only for bundling files together; it does not provide the file decompression features for which zip utilities are known.

Creating a zip file with no compression is straightforward using the WinZip GUI. Alternately, use the `-e0` option to disable compression when using either `winzip` or `pkzip` from a command line.

Programming the Zip File to Flash

For your program to access files in the zip file subsystem, you must first program the zip data to flash. As part of the project build process, the Nios II development tools create a Motorola S-record file (**.flash**) that includes the data for the zip file system.

You then use the Nios II Flash Programmer to program the zip file system data to flash memory on the board.

For details about programming flash, refer to the *Nios II Flash Programmer User Guide*.

Referenced Documents

This chapter references the following document:

Nios II Flash Programmer User Guide

Document Revision History

Table 12–1 shows the revision history for this document.

Table 12–1. Document Revision History (Part 1 of 2)

Date & Document Version	Changes Made	Summary of Changes
November 2009 v9.1.0	No change from previous release.	
March 2009 v9.0.0	■ Reorganized and updated information and terminology to clarify role of Nios II Software Build Tools. ■ Corrected minor typographical errors.	
May 2008 v8.0.0	No change from previous release.	
October 2007 v7.2.0	No change from previous release.	
May 2007 v7.1.0	■ Added table of contents to “Introduction” section. ■ Added Referenced Documents section.	
March 2007 v7.0.0	No change from previous release.	
November 2006 v6.1.0	No change from previous release.	
May 2006 v6.0.0	No change from previous release.	
October 2005 v5.1.0	No change from previous release.	

Table 12–1. Document Revision History (Part 2 of 2)

Date & Document Version	Changes Made	Summary of Changes
May 2005 v5.0.0	No change from previous release.	
May 2004 v1.0	Initial Release.	