Application Note

Programming External Flash Memory on CoreMP7-based Development Boards

Introduction

Programming external memory for an embedded processor in an FPGA can be a challenge for System designers. An embedded processor requires at least two memory spaces for operation. A Program Memory (typically flash) and a data memory (typically SRAM). The program memory stores the application being executed on the embedded processor. The data memory is used at runtime for variable storage, buffering, etc. Unless there is an expensive JTAG interface on the device, there is no simple mechanism for prototyping an application to program the external memory. Designers often choose between implementing a hardware bootloader (using FPGA gates) or a software bootloader (an application running in debug mode on the embedded processor) to program the external memory receiving the memory contents from an external communications source (i.e. a serial link). This application note focuses on a third mechanism that uses the debugging interface's communications console with built-in memory programming commands to program the memories found on the CoreMP7 development kits.

Actel's software development tool, SoftConsole, based on the GNU Toolchain, includes a communications client for interfacing between the GNU Project Debugger (GDB) and the FlashPro3 programmer used for On-Chip Debugging (OCD). The communications client, FS2 Console, is responsible for communicating and bridging the GDB commands to JTAG commands. The JTAG commands can then be communicated and executed on the embedded processor. The FS2 Console communications client can be operated in a standalone mode and used to program external flash memory via the tool's memory programming feature.

Implementation

The FS2 Console flash memory programming interface supports a number of flash memories and configurations. Additional memories and configurations are added quickly and easily by editing a text file (flash.cfg) with the proper configuration settings for the target memory. For more details on configuring an alternative flash memory device, consult the Memory Commands > Flash section of the cliarm.hlp file. The cliarm.hlp and the ISA ARM debugger User Guide are installed as part of the SoftConsole tool and reside on the users hard drive.

The flash.cfg configuration file needs to be edited if the programming target is either the COREMP7-1000-DEV-KIT or SYSMGMT-DEV-KIT development kit as the STMicro flash memory configuration is not contained within the configuration file. Copy and paste the configuration details below into the flash.cfg file found in the \ARM\bin directory of the SoftConsole installation file. A pre-edited version of the flash.cfg file is also included with design files folder (CoreMP7_DF.zip).

```
[M29W800-quad]
comment=Two 16-bit STMicro Flash devices
size=4
program=0x1554,0x00aa00aa 0xaa8,0x00550055 0x1554,0x00a000a0 %A,%D
chipErase=0x1554,0x00aa00aa 0xaa8,0x00550055 0x1554,0x00800080 0x1554,0x00aa00aa
0xaa8,0x00550055 0x1554,0x00100010
sectorErase=0x1554,0x00aa00aa 0xaa8,0x00550055 0x1554,0x00800080 0x1554,0x00aa00aa
0xaa8,0x00550055 %A,0x00300030
reset=0x1554,0x00f000f0
protect=0,%N
unprotect=0,%N
```

If using either the M7AFS-DEV-KIT-SCS or the M7A3P-DEV-KIT-SCS development kit, the Intel flash memory configuration is included by default and no edits are necessary.

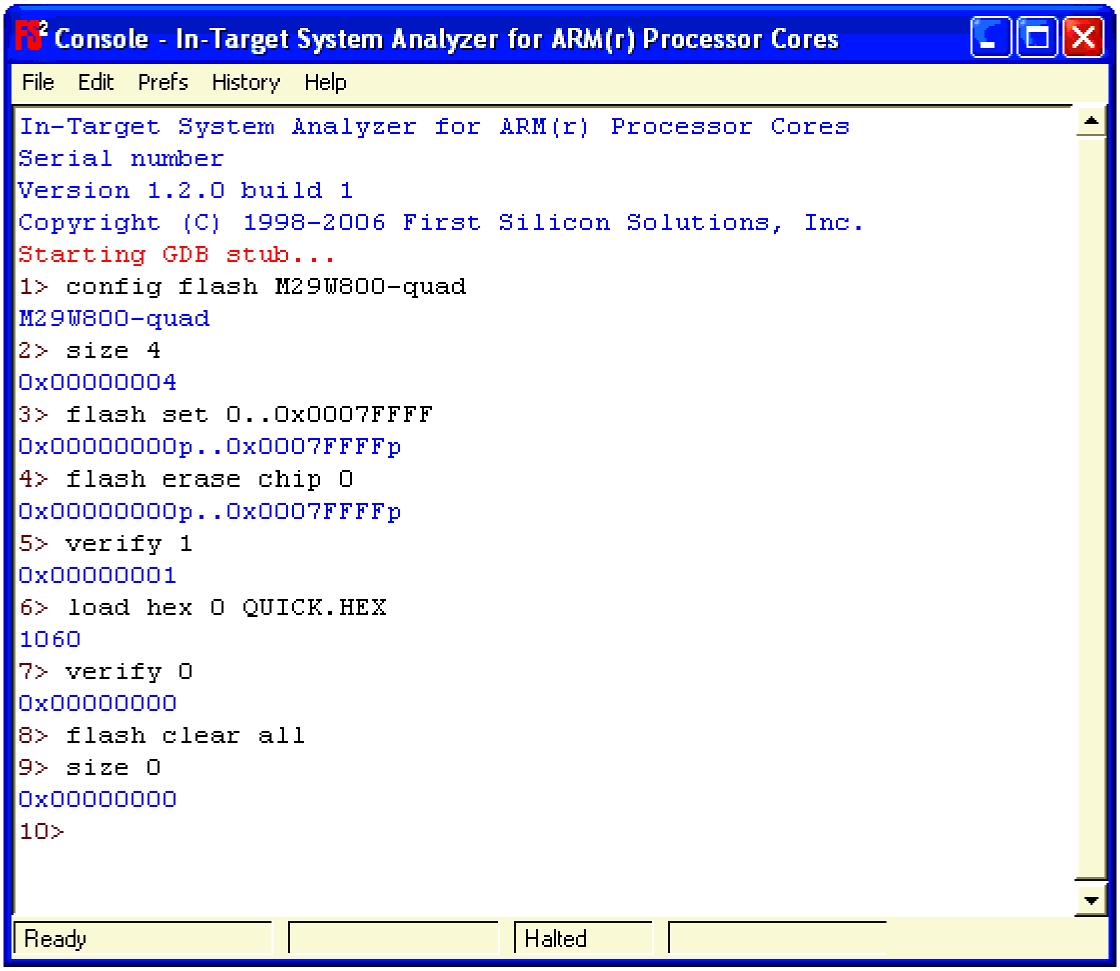
Examples

The following examples demonstrate how to program the flash memories populated on the development kits. The FS2 Console communications client is launched via the command line by navigating to the \ARM\bin directory of the SoftConsole installation file and typing

```
cliarm initarm.tcl.
```

The examples assume that the target programming file (i.e. quick.hex) is also located in the \ARM\bin directory. The tool supports the Intel Hex, Motorola S-Record and binary file formats.

Figure 1 details the commands required for programming the STMicro flash memory populated on the CoreMP7 and System Management development kits. The sequence of commands first specifies the type of memory and access size being targeted, followed by the mapping of its address range. The flash memory is erased, verification enabled, and then programmed. Once programming has completed, verification is disabled, flash memory is unmapped, and the access size is returned to normal. Finally, the FS2 Console utility needs to be terminated and the embedded processor reset. The processor will execute the programmed application.


FS2 Console - In-Target System Analyzer for ARM(r) Processor Cores
File
Edit
Prefs
History
Help
Ready
Halted

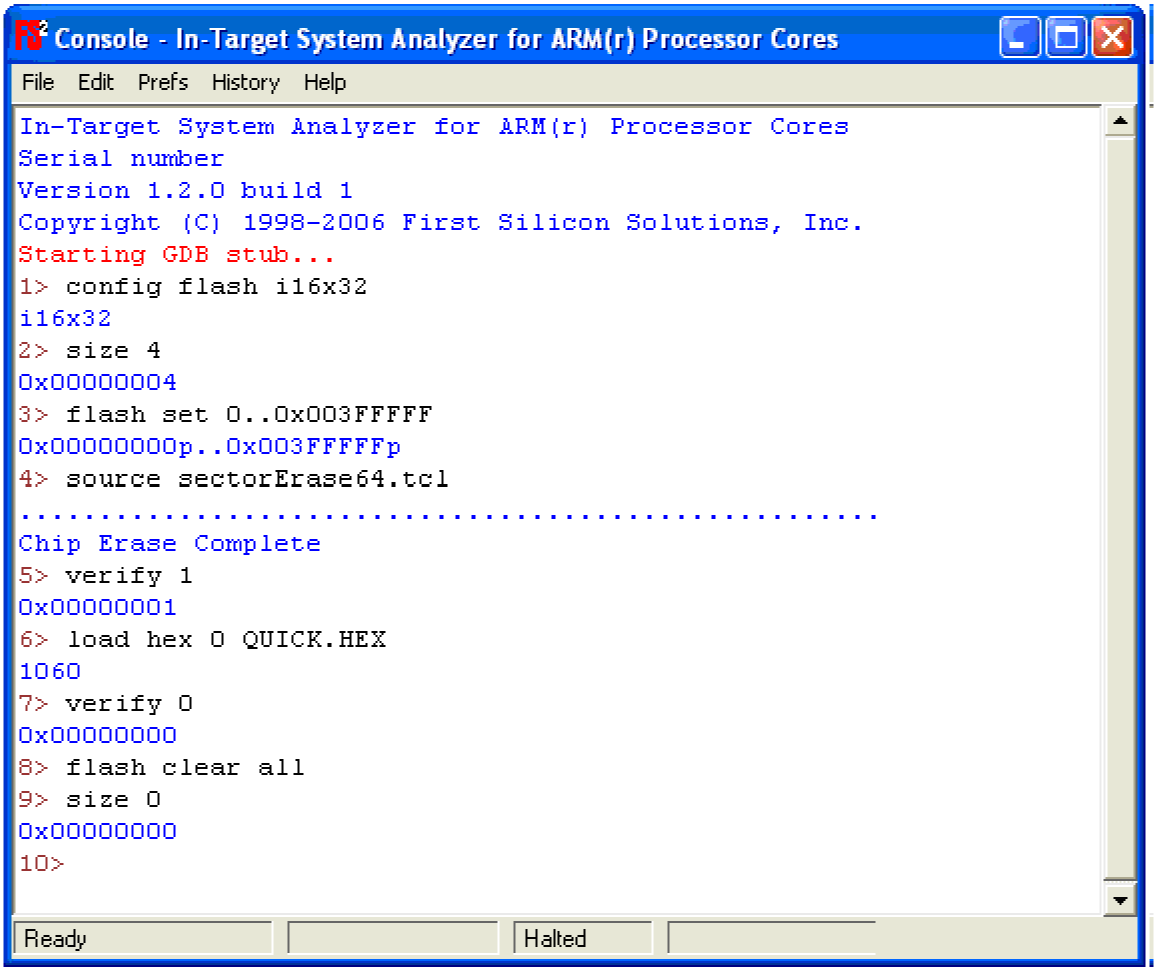
```
In-Target System Analyzer for ARM(r) Processor Cores
Serial number
Version 1.2.0 build 1
Copyright (C) 1998-2006 First Silicon Solutions, Inc.
Starting GDB stub...
1> config flash M29W800-quad
M29W800-quad
2> size 4
0x00000004
3> flash set 0..0x0007FFFF
0x00000000p..0x0007FFFFp
4> flash erase chip 0
0x00000000p..0x0007FFFFp
5> verify 1
0x00000001
6> load hex 0 QUICK.HEX
1060
7> verify 0
0x00000000
8> flash clear all
9> size 0
0x00000000
10>
```

Figure 1 • **FS2 Console Programming Commands for STMicro Flash M**

Figure 2 on page 3 demonstrates the commands required for programming the Intel® flash memories populated on the M7A3P-DEV-KIT-SCS and M7AFS-DEV-KIT-SCS development kits. The fundamental procedure is the same as for programming the STMicro flash memory, with two differences:

- The first command is altered to target the Intel flash memory instead of the STMicro memory.
- Intel® flash devices do not have a dedicated Chip Erase command. A series of Sector Erase commands must be used in series to erase the device. To simplify the chip erase process, a TCL script is supplied. The script calls the sector erase command for each of the 64 sectors found on the populated flash device.

Prior to its execution, copy the supplied sector Erase64.tcl file to the \ARM\bin directory of the SoftConsole installation file. During the script's execution a period will be displayed after each sector has been erased successfully.


Console - In-Target System Analyzer for ARM(r) Processor Cores
File
Edit
Prefs
History
Help
Ready
Halted

```
In-Target System Analyzer for ARM(r) Processor Cores
Serial number
Version 1.2.0 build 1
Copyright (C) 1998-2006 First Silicon Solutions, Inc.
Starting GDB stub...
1> config flash i16x32
i16x32
2> size 4
0x00000004
3> flash set 0..0x003FFFFF
0x00000000p..0x003FFFFFp
4> source sectorErase64.tcl
......................................................
Chip Erase Complete
5> verify 1
0x00000001
6> load hex 0 QUICK.HEX
1060
7> verify 0
0x00000000
8> flash clear all
9> size 0
0x00000000
10>
```

Figure 2 • **FS2 Console Programming Commands for Intel Flash Memory**

The design files (CoreMP7_DF.zip) for this application note contain two STAPL files. One file targets the COREMP7-1000-DEV-KIT and the other targets the M7A3P-DEV-KIT-SCS development kit. Both STAPL files have the external flash memory mapped to 0x00000000, the external SRAM mapped to 0x10000000, and CoreGPIO for input/output mapped to the 0xC2000000 memory locations. For successful programming, the CoreMemCtrl must be configured with only one wait-state for both read and write accesses to the flash memory. The quick.hex programming file is the compiled output of the SoftConsole Quick Start Guide application and simply displays a different pattern on the LEDs based on which button or dip switch is pressed.

Conclusion

Programming external flash memory is no longer a concern for designers utilizing embedded processors with regards to Actel's CoreMP7. The FS2 Console utility provides designers with the flexibility needed to program both current and future flash memory devices. The built-in flexibility removes the requirement of using a specific device from a specific manufacturer. This allows both software and hardware engineers to focus on the task at hand and not have to worry about general systems issues.

References

Actel SoftConsole Quick Start Guide

http://www.actel.com/documents/SoftConsole_QS_UG.pdf

COREMP7 flash programming files_

http://www.actel.com/documents/CoreMP7_DF.zip

Intel 28F640J3D Flash Memory Datasheet

http://download.intel.com/design/flcomp/datashts/31657704.pdf

STMicro M29W800DT Flash Memory Datasheet

http://www.st.com/stonline/products/literature/ds/8156/m29w800dt.pdf

www.actel.com

Actel Corporation
2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.
River Court, Meadows Business Park
Station Approach, Blackwater
Camberley Surrey GU17 9AB
United Kingdom
Phone +44 (0) 1276 609 300
Fax +44 (0) 1276 607 540

Actel Japan
EXOS Ebisu Building 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668
www.jp.actel.com

Actel Hong Kong
Room 2107, China Resources Building
26 Harbour Road
Wanchai, Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488
www.actel.com.cn