Performing Dynamic Reconfiguration in FPSLIC™ Devices – A Scrolling Message Display

Features

- **Can Be Implemented on the ATSTK94 FPSLIC Starter Kit**
- **Dynamic Reconfiguration Used to Generate Scrolling Message**
- **FPGA I/O Drives the Alphanumeric Display, AVR® Drives the LEDs**
- **Alphanumeric Values Determined by the AVR Code**

Introduction

This application note demonstrates the principles of dynamic reconfiguration and how to implement them on Atmel's AT94K,S series FPSLIC devices. In this design the contents of a ROM located in the FPGA core are dynamically reconfigured by the AVR. The varied ROM contents changes the appearance of characters scrolled across an alphanumeric display. [1]

Description

The AVR block sends a message string of characters over the AVR data bus to the FPGA block of the FPSLIC. The FPGA block contains control logic and a character ROM lookup table that receives then decodes the ASCII characters into their corresponding alphanumeric representation. The FPGA I/O toggle the cathode signals to the alphanumeric display, allowing a scrolling message to be viewed on the ATSTK94 board.(1)

Every other execution of the main loop, the AVR reconfigures the ROM decoder in the FPGA (see Figure 1). Reconfiguration is accomplished by re-writing the FPGA's SRAM configuration information. Four special purpose registers FPGAX, FPGAY, FPGAZ, and FPGAD are used in this process. Changing the content of the FPGA's configuration SRAM is also known as Cache Logic Reconfiguration.

Initially, the character decode ROM is loaded with lower case letter decoding. However, after reconfiguration the ROM table is remapped with upper-case letter decoding. Hence, the exact same message string from the AVR which was first shown in lower-case on the alphanumeric display, is thereafter shown in upper-case. This process is repeated continuously.

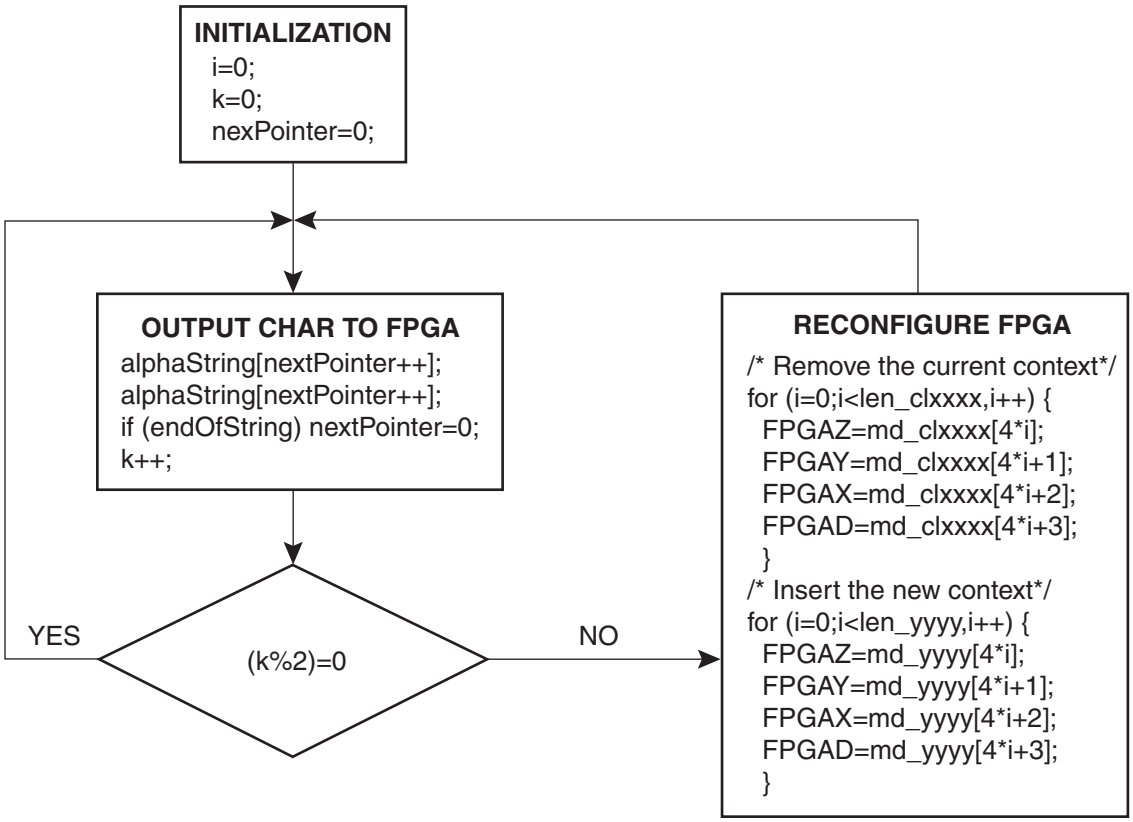
Note: 1. References: The design was provided by *The Institute of Information Theory* and *Automation, and the Kayser Italy Company*: http://www.utia.cas.cz/, http://www.reconf.org/, www.kayser.it.

Performing Dynamic Reconfiguration in FPSLIC™ Devices

Application Note

3515A–FPSLI–9/04

Figure 1. AVR Program Flow



Design Files

The accompanying ZIP file (Reconf-Scrolling.zip) contains the following files:

- **scrolling_smacro.vhd** is the top-level *vhd* file that contains the top entity of the design and the instance of the reconfigurable ROM decoder.
- **DisplayRom_lower.vhd** and **DisplayRom_upper.vhd** are the two contents of the ROM decoder to be reconfigured on the fly.
- **scrolling_smacro.dcf** is the design constraints file required by the Figaro Temporal System Planner Tool to generate proper configuration contexts.
- **scrolling_smacro.ict** is the AVR-FPGA interface connection.
- **scrolling_smacro.pin** is the pin lock file to be used in Figaro to lock the pins of the FPGA.
- **static.bst**, **displayrom_lower_one.md4**, **clear_displayrom_lower_one.md4**, **displayrom_upper_one.md4**, and **clear_displayrom_upper_one.md4** are bitstream files that define the configuration of the FPGA static part and the two contents of the reconfigurable ROM decoder.
- **Scroll-smacro.c** outputs a user-defined string to the alphanumeric displays on the starter kit; at the same time it shows the position within the string on the LEDs connected to the AVR side, and reconfigures the ROM decoder every other loop.
- The files **displayrom_lower_one.inc**, **clear_displayrom_lower_one.inc**, **displayrom_upper_one.inc**, and **clear_displayrom_upper_one.inc** contain the bitstream data necessary to reconfigure the ROM decoder. These files are based on the MD4 bitstream data generated by Figaro IDS 7.6.7 patch level 7.
- **Scroll-smacro.hex** is the compiled AVR binary file.

- **fpslic_scrolling_reconf.bst** is the composite FPSLIC bitstream that implements the design.

Design Implementation

There are three ways to see the working demo:

1. You can download the file **fpslic_scrolling_reconf.bst** to ATSTK94 and observe its operation.
2. You can use Atmel System Designer 3.0 to link these files to produce an FPSLIC bitstream:

   static.bst
   scroll_smacro.hex
3. You can modify either the VHDL code or the C code, re-compile it, and link the rest. If you would like to re-implement the VHDL code, make sure you have the proper version of Atmel Figaro IDS with built-in support for dynamic reconfiguration (for more details on implementing designs in Figaro IDS see the Atmel Application Note ref. #2316, the Figaro IDS User's Guide, and the Temporal System Planner User's Guide).

IMPORTANT: *When you modify and re-implement the VHDL code, remember to update the ".inc" files included in the C source code to contain valid configuration information. Also, be aware of the limited size of the internal SRAM memory.*

How to Run the Demo

1. When switching from **PROG** to **RUN**, the scrolling design runs. **RECONF WELCOMES YOU** appears on the alphanumeric display from right to left. The LEDs increment as the alphanumeric display changes, shifting each character from right to left. The string values are coming from the AVR.
2. Set JP19 to AVRRESET and hold down SW12 RESET. The string of characters freezes and the LEDs do not light up.
3. Release SW12. Both LEDs and the alphanumeric display restart.
4. If the design does not run immediately, set JP19 to RESET and use the RESET button SW12 to force a download from the Configurator to the FPSLIC device; or power cycle the board.

Should you wish to change the scrolling banner, perform the following steps:

1. Change the sentence within the quotation marks to display what will be shown on the alphanumeric display:

   unsigned char alphaString[] = "reconf welcomes you";
2. The AVR will write the data on the data bus using the following command line:

```
/*Output string_characters on FPGA I/O address line*/
for (counter = 0; counter < width; counter ++)
{FISCR = counter;
if ((nextPointer < width) && (counter > nextPointer)) FISUA = ' ';
else FISUA = alphaString[nextPointer - counter];
}
```

3. Recompile your C code.
4. Re-run the bitstream generation.
5. Reprogram the EEPROM.

Troubleshooting

If the board does not perform as described:

1. Check that the mode has been switched from **PROG** to **RUN**.
2. Power-cycle the board.
3. Check the jumper positions on the board.
4. Retry the programming with CPS.
5. Contact the FPSLIC Hot line or your local FAE as listed on the contact page.

Package Contents

```
avr/
     bin/                          - AVR bitstreams
     src/                          - AVR sources
            Scroll-smacro.c        - the main C program
            clear_displayrom_lower_one.inc
            clear_displayrom_upper_one.inc
            displayrom_lower_one.inc
            displayrom_upper_one.inc
                                   - the include files with configuration information
fpga/
     src/
             DisplayRom_lower.vhd- the lowercase ROM decoder
             DisplayRom_lower.vhd- the uppercase ROM decoder
             scrolling_smacro.vhd- the top-level file with AVR-FPGA interface
     syn/
            scrolling_smacro.rct- the pin constraint file
            scrolling_smacro.ict- the AVR-FPGA interface connections
            scrolling_smacro.dcf- the design constraints file for reconfiguration
     bst/                              - FPGA bitstreams
            static/
                   static.bst          - the static part of the FPGA
            differential/
                   clear_displayrom_lower_one.md4
                   clear_displayrom_upper_one.md4
                   displayrom_lower_one.md4
                   displayrom_upper_one.md4
                                       - bitstreams implementing dynamic reconfiguration of
                                         the FPGA
bst/                                   - FPSLIC bitstreams
      fpslic_scrolling_reconf.bst
                                       - an FPSLIC bitstream with the reconfigurable scroll-
                                         ing demo
```

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

3515A–FPSLI–9/04