
Implementing the “Scrolling Design” on the ATSTK94 FPSLIC™ Starter Kit

Features

- Implemented on the ATSTK94 FPSLIC Starter Kit
- AVR® Driving PortD to LED
- FPGA I/O Driving Alphanumeric Display
- Alphanumeric Values Determined by the AVR Code
- Implementation of AVR-FPGA Interface
- Use of Co-verification to Verify the Design in Software

Introduction

This application note explains how to use the alphanumeric display connected to the FPGA and the LED driven by PortD from the AVR.

The AVR Timer Counter 1 is used to slow down the operation of the LED and the alphanumeric output to make the output visible on the starter kit. The AVR Timer Counter 0 is used to view from Co-verification, see the code under “Co-verification” on page 4.

This design directory (C:\SystemDesigner\examples\AT94K\atstk94 designs\scrolling) contains the following files:

- scrolling.vhd is the top-level vhd file, which contains the ROM table for all the characters to be used for the alphanumeric display.
- scroll.c outputs the alphanumeric displays on the starter kit showing increment on the LED and the characters that will be shifted on the alphanumeric display.
- rom.hex is the file used to generate the ROM macro which it is implemented inside scrolling.vhd.
- scrolling.ict is the AVR-FPGA interface connection.
- scrolling.pin is the pin lock file, which will be used within Figaro to lock the pins of the FPGA.

Scrolling Design Description

The string of characters passing through from an 8-bit data bus, which is enabled when the AVR I/O Write Enable (FIOWEA) signal and the corresponding FPGA I/O (Load) Select are both active. From the FPGA side, it will toggle each cathode so only one character will be displayed at a time. A 7-bit address will be decoded from the AVR for selecting the corresponding character from the ROM macro. A complete ROM table is already implemented in the ROM macro earlier, so any character can be displayed from the alphanumeric display. Whenever the FPGA is toggling the cathode signal, the scrolling behavior will be seen from the display. See Figure 1 for further architectural information of the scrolling design.



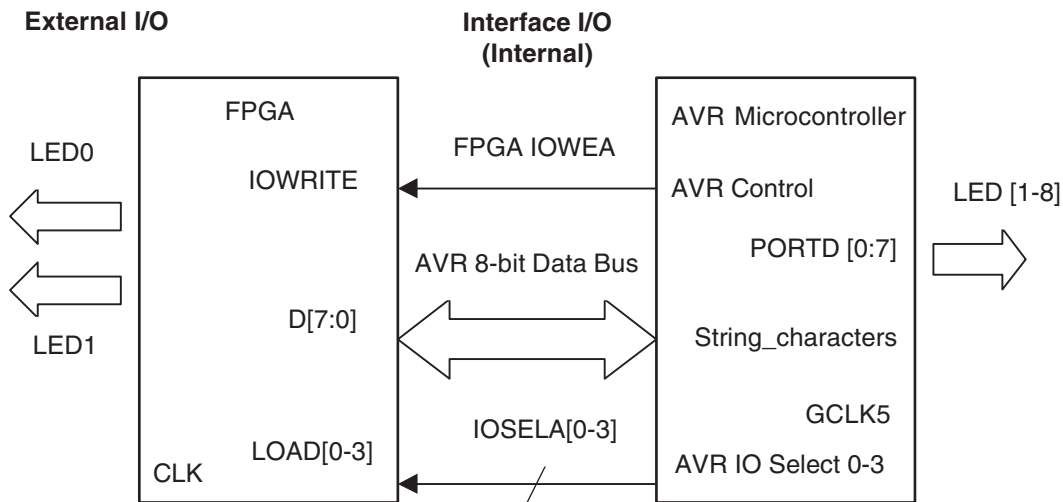
ATSTK94 Series Field Programmable System Level Integrated Circuit

Application Note

Rev. 2316A-06/01



Figure 1. Functional Block Diagram of Scrolling Design



Design Implementation

1. Synthesize Scrolling.vhd
2. Compile Scroll.c
3. Implement the FPGA-AVR Interface as shown in Table 1.
4. Before performing place and route create a ROM macro by launching FIGARO IDS in Standalone mode.
 - Launch Macro Generator from: Tools > Entry > Generators
 - Select the Memory tab and ROM
 - Provide the address width with a maximum address of 7 bits and the data width with a maximum of 15 bits.
 - Type rom7x15 for the macro name and press generate to create the ROM macro, see scrolling.vhd for reference.
 - Relaunch Figaro IDS by selecting Open EDIF netlist.
5. Use scrolling.pin to perform place and route of the PPGA portion. Go to: File> Import (constraints) after selecting Figaro and running open EDF.
6. Use the bitstream utilities to generate a combined bitstream file containing scrolling.bst, generated by the place and route step and the scroll.hex file generated by the ICC Compiler.
7. Use the Configurator Programming System (CPS) software provided with System Designer™ to program the starter kit. See the “Starter Kit User Guide” supplied with the kit for more details.
8. When the starter kit has been programmed, make sure that all the jumpers for the LED are set to the “A” position, meaning connected to the AVR PortD.

Table 1. FPGA-AVR Interface Connections

FPGA Design Port Name	Action	AVR Port Name	Tab Label (Left)
Load(0)	<<Connect>>	IOSELA0	AVR IO Selects
Load(1)	<<Connect>>	IOSELA1	AVR IO Selects
Load(2)	<<Connect>>	IOSELA2	AVR IO Selects
Load(3)	<<Connect>>	IOSELA3	AVR IO Selects
D(0)	<<Connect>>	ADINA0	Data from AVR
D(1)	<<Connect>>	ADINA1	Data from AVR
D(2)	<<Connect>>	ADINA2	Data from AVR
D(3)	<<Connect>>	ADINA3	Data from AVR
D(4)	<<Connect>>	ADINA4	Data from AVR
D(5)	<<Connect>>	ADINA5	Data from AVR
D(6)	<<Connect>>	ADINA6	Data from AVR
Io_write	<<Connect>>	FIOWEA	AVR Control
Clk	<<Connect>>	GCLK5	FPGA Clocks

9. When switching from PROG to RUN, the scrolling design runs. FPSLIC STARTER KIT IS HERE appears on the alphanumeric display from right to left. The LEDs increments as the alphanumeric display changes shifting each character from right to left. The string value is coming from the AVR.
10. Set JP19 to AVRRESET and hold down SW12 RESET, the string of characters freezes and the LEDs do not light up.
11. Release SW12, both LEDs and the alphanumeric display restart.
12. 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.
13. The user can change the sentence within the quotation marks to display what will be shown on the alphanumeric display:


```
unsigned char alphaString[] = " FPSLIC STARTER KIT IS HERE ";
```
14. The AVR will write the data on the databus 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];
}
```
15. Recompile your C code.
16. Re-run the bitstream generation.
17. Reprogram the EEPROM.

Co-verification

This design is run purely in hardware. Running this design in simulation takes a long time, because it is visual at one- or two-change per second. In order to do simulation or co-verification, the code needs to be modified to run faster, so simulation and co-verification can run efficiently. See the “Starter Kit User Guide” supplied with the kit for more details on running co-verification.

1. The user only needs to uncomment or comment the code.

Note: Use Timer1 to view the result on the Starter Kit. Use Timer0 to view Co-verification. Make sure all the registers are consistent within the code.

```
...  
...  
TCCR1B = 0x00; //stop  
TCCR1B = 0x03; //start Timer  
//TCCR0 = 0x00; //stop  
//TCCR0 = 0x01; //start Timer  
...  
...
```

2. If the starter kit is run with timer 0 instead of timer 1, the LED light will be on but the design will not be visible because it is running too fast.
3. Verify the result from co-simulation and change the Timer back to Timer1.
4. Combine the bitstream again to view the proper string value from the alphanumeric display.

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 Hotline or your local FAE as listed on the contacts page.



Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
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

Atmel Colorado Springs

1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs

Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

Atmel Grenoble

Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex
France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel FPSLIC Hotline

(408) 436-4118

Atmel FPSLIC e-mail

fpslic@atmel.com

FAQ

Available on web site

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

© Atmel Corporation 2001.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

FPSLIC and System Designer are trademarks of Atmel. AVR is the registered trademark of Atmel.

Terms and product names in this document may be trademarks of others.



Printed on recycled paper.