
AVR-FPGA Interface Design 6

Features

- Initialization and Use of AVR-FPGA Interface and Interrupts
- Initialization and Use of the Shared Dual-port SRAM
- Initialization and Use of the AVR® Hardware Multiplier
- Initialization and Use of the AVR UARTs
- Initialization and Use of the AVR General-purpose I/O Ports
- Full Source Code for AVR Microcontroller and FPGA Included

Description

Atmel's AT94K sample designs are provided to familiarize the user with the AT94K FPSLIC™ device. This design is an enhancement of Design 5 (doc2329.pdf). The only difference between the two designs is that this design also displays the interrupt counts on the AVR General-purpose I/Os (PORTD and PORTE).

This design is composed of a simple AVR program and two loadable ripple-carry counters implemented in the FPGA. The counters begin counting at zero upon power-up and will generate an interrupt to the AVR using the ripple-carry out signal. The active low interrupt must be held for three clock cycles prior to being acknowledged by the AVR.



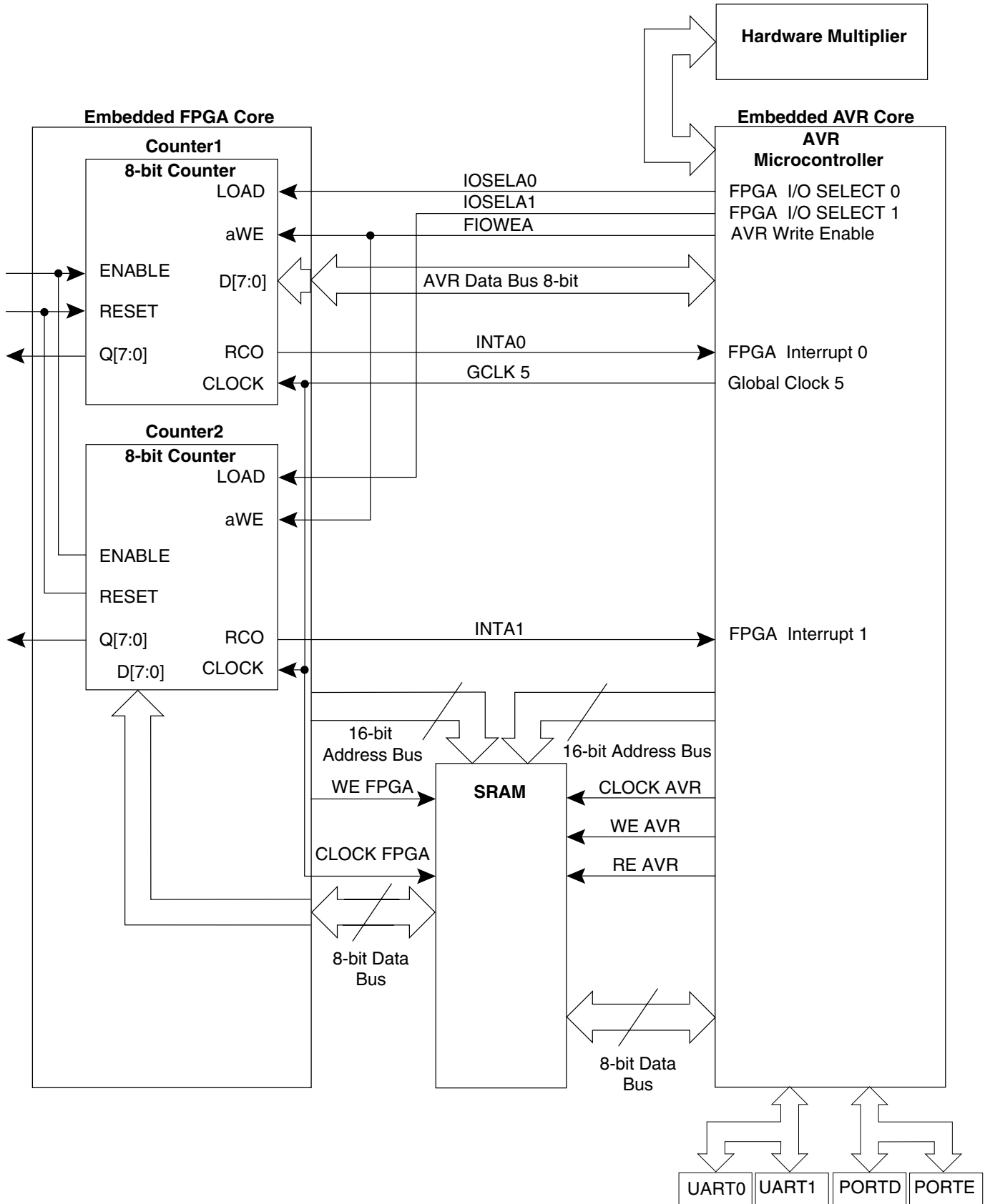
**Programmable
SLI
AT94K**

**Application
Note**

Rev. 2330A-11/01



Functional Block Diagram



The block diagram shows the connections used in this example, all interface connections are implemented using dedicated resources. This example makes use of the AVR-FPGA Data Bus, two FPGA Interrupts, two I/O Select signals, two SRAM locations, the AVR Hardware Multiplier, the two AVR UARTs and the two AVR General-purpose I/O Ports.

Implementation

The AVR source of this design has been implemented in C for the CodeVision AVR, IAR Systems and ImageCraft compilers, and in assembly for the Atmel Assembler. The FPGA design has been implemented in Verilog and VHDL. Each file is commented.

Assuming a default System Designer™ installation, the source files for this application note can be found within the `\SystemDesigner\Examples\AT94K\Designs\doc2330` directory. Alternatively, the source files may be found in the FPSLIC Software section of the Atmel web site (<http://www.atmel.com>), under the **doc2330.zip** archive. The source files for the Design 6 are shown in Table 1.

Table 1. Source Files for Design 6

File	Description
AT94KDEF.INC	Atmel AVR Assembler FPSLIC Include File
D6-ATML.ASM	Atmel AVR Assembler Design 6 Source File
D6-CVAVR.C	CodeVision AVR Design 6 Source File
D6-IAR.C	IAR Systems Design 6 Source File
D6-ICC.C	ImageCraft Design 6 Source File
V150-AT94K-32.XCL	IAR Systems v1.50 Design XLINK File
V200-AT94K-32.XCL	IAR Systems v2.00 Design XLINK File
COUNTER.V	Counter One Implementation Verilog Source File
COUNTER2.V	Counter Two Implementation Verilog Source File
COUNTER3.V	Top-level Verilog Source File
COUNTER.VHD	Counter One Implementation VHDL Source File
COUNTER2.VHD	Counter Two Implementation VHDL Source File
COUNTER3.VHD	Top-level VHDL Source File

The top-level HDL source code is **COUNTER3.V** or **COUNTER3.VHD**. This code instantiates two 8-bit loadable ripple-carry counters, which count from \$00 to \$3F and \$00 to \$1F. The counters' ENABLE and RESET lines are active high. The ripple-carry out signals are connected to AVR Interrupt Signals INTA0 and INTA1, respectively.

The file containing the AVR microcontroller source code depends on the programming language and the compiler being targeted, consult the archive contents above to determine the corresponding file to your tool flow. When the AVR senses an interrupt resulting from the first counter's ripple-carry out signal, the Interrupt Service Routine (ISR) for INTA0 is executed. During the ISR for INTA0, the AVR increments the count of INTA0 occurrences and stores the count in the shared dual-port SRAM at location \$0F9B. The ISR then uses the Hardware Multiplier and multiplies the two interrupt counts together, storing the results in SRAM at locations \$0F9D and \$0F9E. The multiplication result is then transmitted from UART0 to UART1 and the received bytes are stored in SRAM at \$0F9F and \$0FA0. The multiplication result is compared with the received bytes, if an error occurred the error count at SRAM location \$0FA1 is incremented. Finally, the ISR places the incremented count on PORTD and the 8-bit AVR-FPGA Data Bus, which triggers the AVR Write Enable (FPGA signal aWE) and FPGA I/O Select 0

(FPGA signal LOAD), loading the counter with the value from the AVR-FPGA Data Bus. When the AVR senses an interrupt resulting from the second counter's ripple-carry out signal, the Interrupt Service Routine (ISR) for INTA1 is executed. During the ISR for INTA1, the AVR increments the count of INTA1 occurrences and stores the count in the shared dual-port SRAM at location \$0F9C. Finally, the ISR places the incremented count on PORTE and triggers the AVR Write Enable (FPGA signal aWE) and FPGA I/O Select 1 (FPGA signal LOAD), loading the counter with the value from the shared dual-port SRAM at location \$0F9C

Once the aWE and LOAD signals from either counter are released, the corresponding counter will start counting from the loaded value. When either counter has reached its terminal count, its RCO signal will be driven low and will generate an interrupt to the AVR, causing the corresponding ISR to be executed. This operation will then be repeated.

Simulation

When performing co-verification simulation using Atmel's System Designer software, it is necessary to provide stimulus for the counter's RESET and ENABLE signals. Listed below is the suggested HDL stimulus that should be added to the testbench:

- For VHDL Design Flows:

```
stimulus: process
begin
sig_reset <= '0';
sig_enable <= '0';
wait for 100 ns;

sig_reset <= '1';
wait for 100 ns;

sig_reset <= '0';
sig_enable <= '1';
wait for 200 us;

end process stimulus;
```

- For Verilog Design Flows:

```
initial
begin
sig_reset = 1'b0;
sig_enable = 1'b0;

#100 sig_reset = 1'b1;

#100 sig_reset = 1'b0;
sig_enable = 1'b1;

#200000;
end
```



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 Product Operations

Atmel Colorado Springs

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

Atmel Grenoble

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

Atmel Heilbronn

Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

Atmel Nantes

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

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 Programmable SLI Hotline
(408) 436-4119

Atmel Programmable SLI e-mail
fpslic@atmel.com

FAQ
Available on web site

e-mail
literature@atmel.com

Web Site
<http://www.atmel.com>

© 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.

ATMEL® and AVR® are the registered trademarks of Atmel; FPSLIC™ and System Designer™ are the trademarks of Atmel.

Other terms and product names may be the trademarks of others.



Printed on recycled paper.