Accessing the Dual-port SRAM Block from the FPGA Side of the FPSLIC

Introduction

This application note provides designers with an understanding of how to access and achieve optimal performance from the Shared Dual-port SRAM block from the FPGA side of the FPSLIC.

Description

Atmel AT94K and AT94S FPSLIC devices contain up to 36 Kbytes of synchronous dual-port SRAM. Although a portion of the SRAM address space is dedicated to instruction code storage for the AVR microcontroller core, the remaining SRAM space is intended for data storage that can be shared equally by the FPGA and AVR cores, see Figure 1.

Code space and data space are separated by a soft partition. The user has the choice of four partition locations, allowing for size customization of program and data space.

The AT94K40AL referenced in this application note will use the “10K word Program/16K byte Data” option. Please refer to the AT94K or AT94S datasheets for details on partition sizes and settings for the entire FPSLIC family of devices.

Programmable SLI

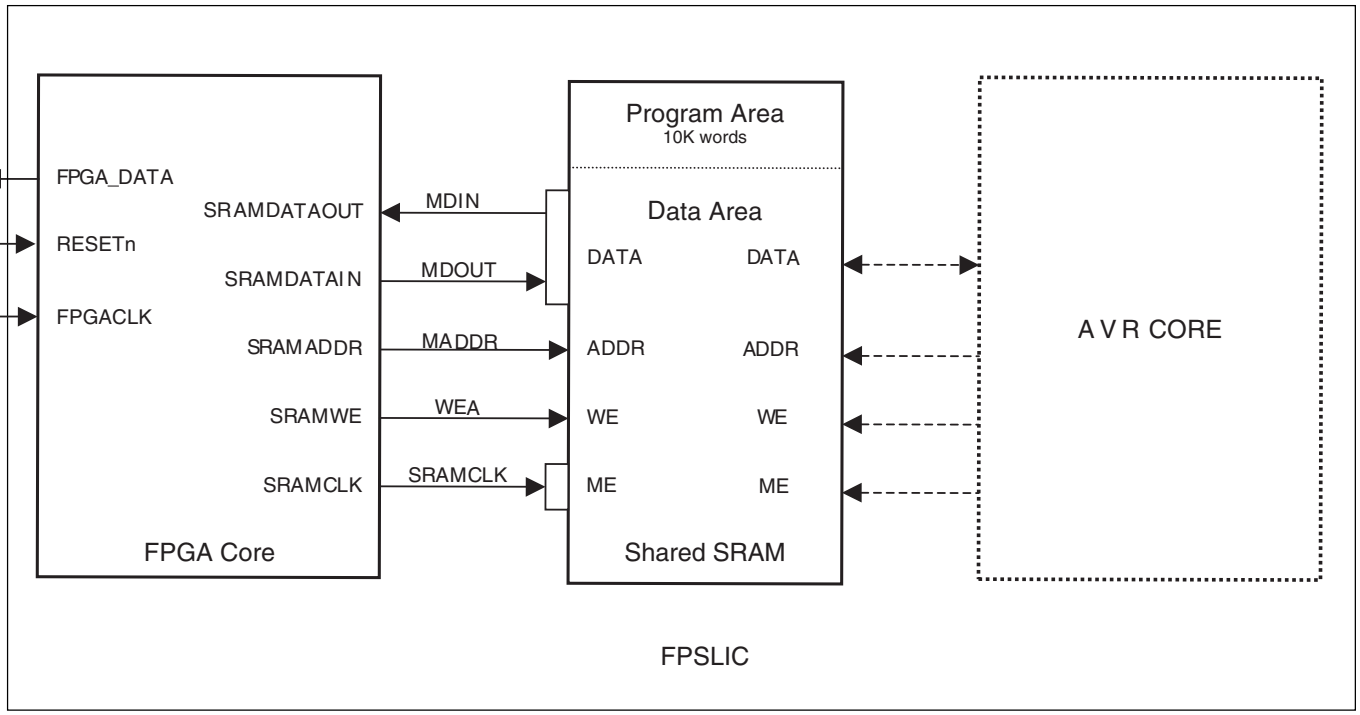
AT94K

AT94S

Application Note

2824A-FPSLI–4/03

Figure 1. Block Diagram of FPGA Logic Interfacing to Shared SRAM

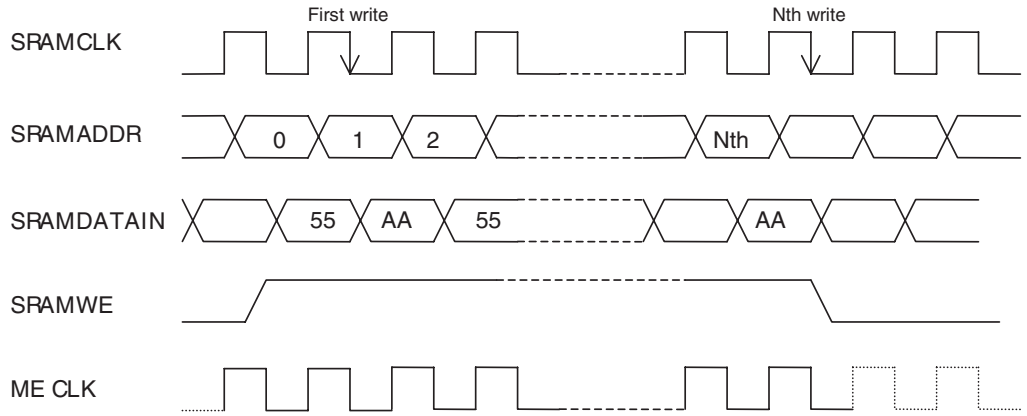


Interfacing the Data SRAM

When designing the FPGA logic that will access the shared SRAM block, it is important to note the proper clock edge and timing relationships between the address bus, data bus, write enable and clock signals being supplied to the SRAM interface ports. As indicated by the FPSLIC AT94K datasheet description of the Dual-port SRAM operation, addresses are registered on the rising edge of the internal ME clock, while data is latched on the falling edge of the ME clock. Additionally, write cycles begin by asserting a logic “1” on WE before the rising edge of the ME clock, while write cycles end on either the falling edge of the ME clock or falling edge of the WE, which ever comes first. Refer to the FPSLIC datasheet section covering the Dual-port SRAM timing characteristics. It is also important to choose the correct clock edge option in the System Designer software tool, refer to the AT94K device options.

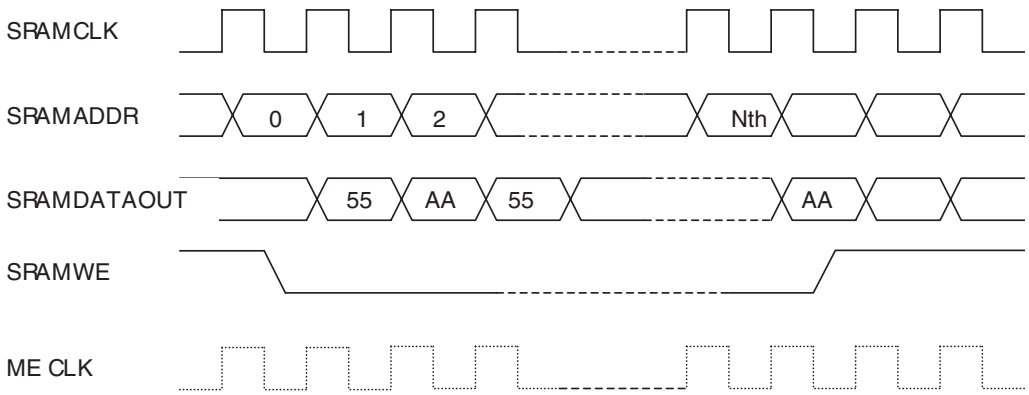
In the example provided herein, the “positive edge” clock option will be used. This will result in the SRAMCLK being passed directly to the ME clock port. Figure 2 shows the write cycle timing relationship of the FPGA logic. Notice that address updates are generated on the rising edge of the clock, while data and write strobe updates occur on the falling edge, half clock cycle later. This will result in the most optimal timing relationship at the SRAM interface.

Figure 2. SRAM Write Waveform Using “Positive Edge” Clock Option



To achieve the optimal timing relationship with the “negative edge” clock option (System Designer default setting), consider the ME clock will be inverted 180 degrees in phase from the SRAMCLK. Hence, address changes should occur with respect to the falling edge of SRAMCLK, while updates to data and the write strobe should occur on the rising edge. In Figure 3, the Read cycle timing relationship of the FPGA logic is shown. Read cycles are started by asserting the SRAMWE signal Low and by supplying the address to be read on the SRAMADDR bus prior to the rising edge of the SRAMCLK. On the rising edge of the clock, the address will be latched and valid data will be driven onto the SRAMDATAOUT bus.

Figure 3. SRAM Read Waveform Using “Positive Clock Edge” Option



Interface Connections

Connections between the I/O ports referenced in the FPGA code and the SRAM interface ports are accomplished by way of the AVR-FPGA Interface flow within the System Designer software tool.

The ports MDIN, MDOUT, MADDR, WEA and SRAMCLK referenced in figure 1 can be found in AVR-FPGA Interface connection table.

Example FPGA Code

The example code below causes the FPGA core logic to write the data pattern 55 at all even addresses and AA at all odd addresses, from address 60h to FFh. After reaching address FFh, the FPGA logic will begin to read the SRAM contents and output the data values on to FPGA I/O pins.

fpga2sram.v

```
/*********************************************************************
          Atmel Corporation

PROJECT NAME   :   FPGA core interface to shared data SRAM
TITLE          :   fpga2sram.v
REV            :   1.00
DATE           :   10/25/02
DESCRIPTION    :   This design writes a parity pattern across the entire
                   shared SRAM data space. It then reads the entire SRAM
                  contents, and sends the data to FPGA I/O.
*********************************************************************/

module fpga2sram (
      sramwe,  // SRAM write enable signal (active high writes)
      sramaddr,  // SRAM sramaddress bus
      sramdatain, // SRAM data bus to SRAM from FPGA
      sramdataout,   // SRAM data bus from SRAM to FPGA
      fpga_data, // Data bus from FPGA register to FPGA I/O
      fpgaclk,     // FPGA Clk
      sramclk,// SRAM Clk
      resetn   // FPGA Reset
         );

   input  resetn;
   input  fpgaclk;
   input [7:0] sramdataout;

   output  sramwe;
   output  sramclk;
   output [15:0] sramaddr;
   output [7:0]  fpga_data;
   output [7:0]  sramdatain;
```

```
   reg  write_strb;
   reg  sramwe;
   reg [7:0] fpga_data;
   reg [7:0]  data_reg;
   reg [15:0]  sramaddr;
   reg [7:0]  sramdatain;

   wire  sramclk;

   assign sramclk = fpgaclk;  // Use the FPGA clk to drive the SRAM clock
port.

   // Address generation process //

   // From reset on, the counter will count up from the first
   // address space visible by the AVR (60h), to highest data
   // space address,ffh (256 bytes)

   always @(posedge fpgaclk or negedge resetn)
      begin
   if(~resetn)
      begin
         write_strb <= 1'b1;
         sramaddr <= 16'h0060;
      end // if (~resetn)
   else
      if (sramaddr == 16'hff)
         begin
      write_strb  <= ~write_strb;
      sramaddr <= 16'h0060;
         end // if (sramaddr == 8'hff)
      else
         begin
      write_strb  <= write_strb;
      sramaddr <= sramaddr + 1;
         end // else: !if(sramaddr == 8'hff)
      end // always @ (posedge fpgaclk or negedge resetn)

  // Data to SRAM process//

   // Clocking the data out on the negative fpgaclk edge provides additional
   // hold time during the write cycle. Make sure the 'positive edge' clock
   // option is selected in the System Designer software tool.
   //
   // Data presented to SRAM for writes should be provided 1/2 clock cycle
   // after the address is generated. The next two processes work together.
```

```
   always @(posedge fpgaclk or negedge resetn)
      begin
   if(~resetn)
      data_reg <= 8'b0;
   else
      if (sramaddr[0]) // Store 55 at odd locations
         data_reg <= 8'h55;
      else
         data_reg <= 8'hAA; // Store AA at even locations
      end // always @ (posedge fpgaclk or negedge resetn)


   // Delay SRAM write data and write pulse by 1/2 clock cycle

   always @(negedge fpgaclk or negedge resetn)
      begin
   if(~resetn)
      begin
         sramdatain <= 8'h0;
         sramwe <= 1'b1;
      end // if (~resetn)
   else
      begin
         sramdatain <= data_reg;
         sramwe <= write_strb;
      end // else: !if(~resetn)
      end // always @ (negedge fpgaclk or negedge resetn)


   // Data from SRAM process //

   // Clock data from SRAM to FPGA I/O
       always @(posedge fpgaclk or negedge resetn)
       begin
   if(~resetn)
      fpga_data <= 8'h00;
   else
      fpga_data <= sramdataout;
       end // always @ (negedge fpgaclk or negedge resetn)

endmodule // fpga2sram
```

Atmel Corporation

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

Microcontrollers

2325 Orchard Parkway
San Jose, CA 95131
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
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
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

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

2824A–FPSLI–4/03 xM