Replacement of a RAM with Atmel FreeRAM[™] in Verilog[™]-based Designs

Features

- Verilog Source Code for FreeRAM Implementation
- Examples for Converting Xilinx[®] RAM to Atmel FreeRAM

FreeRAM Features

Atmel's FreeRAM is a versatile component. It can be configured to four different types:

Figure 1. Single-port RAM: Asynchronous or Synchronous







The routing resources connecting the RAM are optimized for each mode. For example, the routing nets needed to route the DOUT signals on the dual-port RAM are not used on the single-port RAMs. These unused resources can be used for other routing schemes for another area in the design.



Programmable SLI AT40K AT40KAL AT94KAL AT94SAL

Application Note



Architectural Differences	Atmel has dedicated RAM blocks inside the FPGA devices. Within every 4 x 4 core cell sector, there is a FreeRAM cell that can be used as 32 x 4 dual-port RAM. This frees up the core cells to be used for logic. Xilinx RAM does not have any separate RAM and must use its CLB to generate them.
	Area consumption becomes a major drawback for Xilinx's RAM. For example, a FIFO design is created that requires the use of a 128 x 16 dual-port RAM. This design would require 128 Xilinx CLBs alone just for the RAM; logic would increase that number. The smallest device required would be a 4005E. If this same design was developed using Atmel's FreeRAM, the RAM would fit into an AT40K05, but the core cells are not affected by the FreeRAM, allowing additional logic to fit into a small device.
Dual Port Capabilities	Dual-port RAM capability is truly achieved in the Atmel architecture. Xilinx uses only one signal (WE) for this write and output enable. Simultaneous read and write is not possible. By asserting WE high, the RAM can only perform a write operation; disabling WE performs a read operation. Atmel RAM implements separate signals for the write and output enables as well as the data input and output. This allows simultaneous read and write operations.
Modifying Verilog Codes to Implement Atmel RAM	Figure 3 shows how a Xilinx RAM is implemented in Verilog. Figure 4 shows how that same design is converted to implement Atmel RAM. It is very important to remember that the polarity for the write enable in the Atmel RAM is active low. For designs where the write signals were active high, an inverter is necessary to correct the polarity.
	During synthesis, the front-end tool will generate a black box for the RAM component. Creation of the RAM block will be generated by the Macro Generator in the IDS soft- ware. The RAM will be stored inside a user-defined library. When the design netlist is imported to IDS, the black box definition calls the RAM definition within the user library. The design can be placed and routed after the successful import.
Replacing Xilinx Components in Verilog	It is possible to convert a design using Xilinx RAM into the Atmel architecture without modifying the original Verilog code. A new file needs to be created. Inside this file is the entity and architecture declaration of the Xilinx RAM. Under the architecture block section, an Atmel component is called and port mapped to simulate the Xilinx RAM. Figure 5, Figure 6 and Figure 7 show how to implement an Atmel RAM underneath the Xilinx component. Synthesis of the design follows the same procedure previously mentioned.
Importing design into Figaro	Prior to importing the *.edf netlist, the RAM must be generated and stored within a user- defined library. This is achieved by using the Macro Generator tool integrated with the software. After generating the RAM, the macro will be called during the *.edf import.

Replacement of a RAM with Atmel FreeRAM in Verilog

Figure 3. Sample Verilog File Implementing Xilinx RAM

```
module example (clk, wr, data_a, data_b);
input
             clk;
input
             wr;
input [31:0] data_a;
                                                                             SPO[15:0]<sup>(1)</sup>
input [15:0] data_b;
                                                              D[15:0]
wire [3:0] ptra;
                                                                                           16/
                                                                              DPO[15:0]
                                                             A[3:0]
wire [4:0] ptrb;
wire [15:0] nullnode;
                                                              DPRA[3:0]
            wrblo;
wire
            wrbhi;
wire
                                                              WR EN
assign wrblo = wr & (~ ptrb[0]);
                                                              WR_CLK
assign wrbhi = wr & ptrb[0];
                                                                               dpr16x16
dpr16x16 RAMLO(data_a[15:0], nullnode[15:0], data_b[15:0], ptrb[4:1], ptra[3:0], clk, wrblo);
dpr16x16 RAMHI(data_a[31:16], nullnode[15:0], data_b[15:0], ptrb[4:1], ptra[3:0], clk, wrbhi);
endmodule
module dpr16x16 (DPO, SPO, DI, A, DPRA, WR_CLK, WR_EN);
output [15:0] DPO;
output [15:0] SPO;
input [15:0] DI;
input [3:0] A;
input [3:0] DPRA;
input WR_CLK;
input WR_EN;
endmodule
```

Note: 1. SPO is never used in the design. The user must create the dummy signal nullhode and connect it to the component so synthesis will not complain.





Figure 4. Sample Verilog File Implementing Atmel RAM with HDLPlanner⁽¹⁾⁽²⁾



Notes: 1. This component will be created using the Macro Generator in IDS within the HDLPlanner. For this example, the following information will be entered:

- Section: Memory, RAM Dual-port Address Width: 8
- Width: 32Ram Type: Synchronous
- 2. Xilinx RAM uses one signal to control the write and output enables of their RAM. Atmel has separate ports for these controllers. The simplest solution is to always assert the output enable. Xilinx RAM can only perform a write or read but not both. It's assumed that the data coming out of DOUT will only be sampled when the write enable is disabled. The original design assumes that the write enable is active high. Atmel RAM defines its write enable to be active LOW. Therefore, inverters are needed to correct the signal interface.

4 Replacement of a RAM with Atmel FreeRAM in Verilog =

Replacement of a RAM with Atmel FreeRAM in Verilog

Figure 5. Xilinx Component with Buried Atmel Component - Example 1







Figure 6. Xilinx Component with Buried Atmel Component – Example 2



Replacement of a RAM with Atmel FreeRAM in Verilog

Figure 7. Xilinx Component with Buried Atmel Component - Example 3

```
module ram16x4d (D, A, DPRA, WE, WCLK, SPO, DPO);
input [3:0] D;
input [4:0] A;
input [3:0] DPRA;
input
            WE;
input
            WCLK;
output [3:0] SPO;
output [3:0] DPO;
             OE;
wire
assign OE = 'b0;
sdpr16x4 U1(.WEN(~WE), .OEN(OE), .CLK(WCLK), .AIN(A), .AOUT(A), .DIN(D), .DOUT(SPO));
sdpr16x4 U2(.WEN(~WE), .OEN(WE), .CLK(WCLK), .AIN(A), .AOUT(DPRA), .DIN(D), .DOUT(DPO));
endmodule
module sdpr16x4 (WEN, OEN, CLK, AIN, AOUT, DIN, DOUT);
input WEN, OEN, CLK;
input [4:0] AIN;
input [4:0} AOUT;
input [3:0] DIN;
output [3:0] DOUT;
endmodule
```











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 fpga@atmel.com – fpslic@atmel.com

FAQ Available on web site

© 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[®] is the registered trademark of Atmel. HDLPlanner[™] and FreeRAM[™] are the trademarks of Atmel.

Verilog[™] is the trademark of Gateway Design Automation Corporation; Xilinx[®] is the registered trademark of Xilinx, Inc. Other terms and product names may be the trademarks of others.



Web Site http://www.atmel.com

BBS 1-(408) 436-4309

