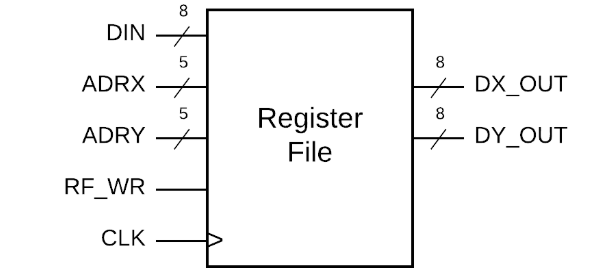
CPE 233: RAT assignment 3

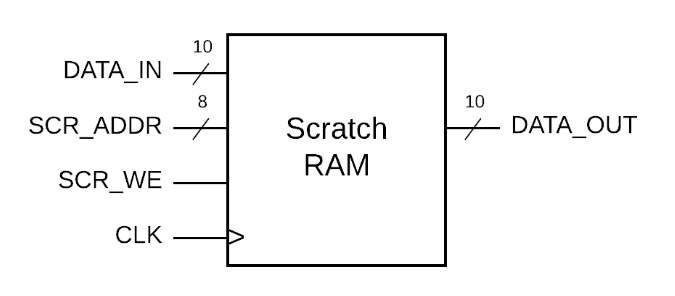
Prof. Bridget Benson

Luis Gomez, Jared Rocha

BBD



*Figure 1: Black Box Diagram Register File*



*Figure 2: Black Box Diagram Scratch Ram*

Behavior

In this assignment, we built two RAM memory modules in system verilog a register file and a scratch ram. The following will explain each.

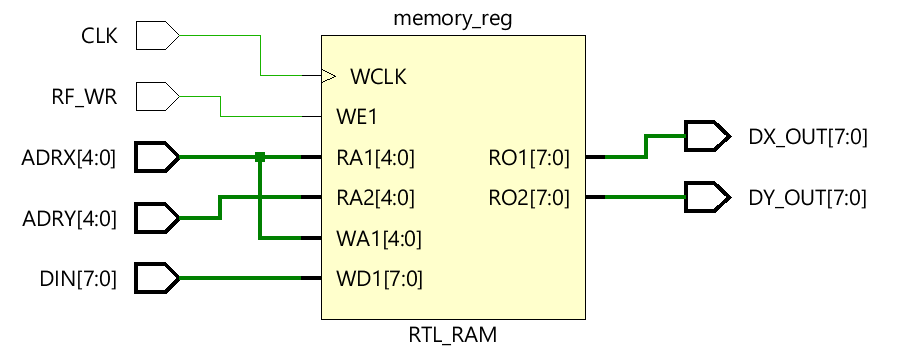
**Register File:**

A simple RAM memory module that holds 32 registers that can each store 8 - bits. Dual port RAM allowing reading from two locations or addresses simultaneously. Address inputs ADRX and ADRY with a value between 0-31, and output values saved in registers to DX\_OUT and DY\_OUT. Only a signal register can be saved per clock cycle with ADRX being the only save address. DATA\_IN is a 8-bit value saved to the location of ADRX. Data is read asynchronously, while data is saved synchronously with the rising clock edge and only if RF\_WR input is 1.

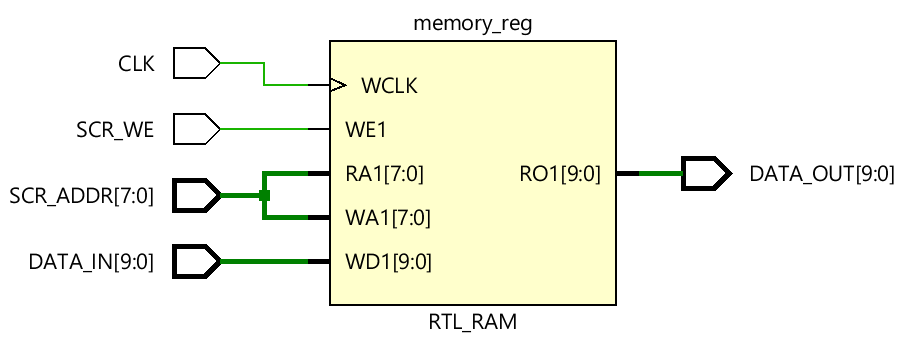
**Scratch Ram:**

Is a RAM memory module that has 256 locations that can store 10 bits. Used for stack memory and temporary saving data. Data can be transferred from Scratch Ram to Register file. Single port RAM so only one address SCR\_ADDR can be read from and written into. DATA\_IN is read asynchronously and written synchronously on the rising clock edge and input SCR\_WE is 1.

Structural Design



*Figure 3: Register File Schematic Drawing*



*Figure 4: Scratch Ram Schematic Drawing*

Verification

**Register File:** We did two test. One with RF\_WR = 1 and one with RF\_WR = 0.

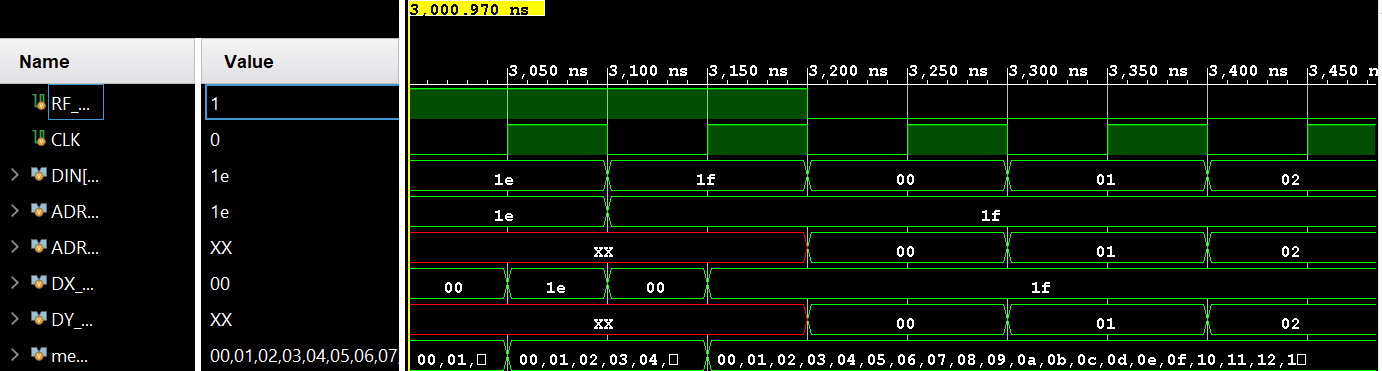
***Test 1:*** With RF\_WR = 1 to write to memory. DIN = i, ADRX = i. With clock cycle of 100 nanoseconds. We ran a for loop to write to all locations of memory and read all locations. Testing ADRX to write to memory and have DX\_OUT read ADRX value.

***Test 2:*** With RF\_WR = 0 to just read from memory but, not write. DIN = i, ADRY = i. With a clock cycle of 100 nanoseconds. We ran a for loop to read all locations of memory. Testing to ensure memory was not written too.

***Test Table:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **RF\_WR** | **DIN** | **ADRX** | **ADRY** | **Explanation** |
| **1** | **1** | **i** | **i** | **N/A** | **Testing writing and reading to memory** |
| **2** | **0** | **i** | **N/A** | **i** | **Testing reading but not writing to memory** |

*Figure 5: Register File Test Table*



*Figure 6: Register File Simulation*

***Register File Test Bench Code:***

module register\_file\_tb();  
 logic RF\_WR, CLK;  
 logic [7 : 0] DIN;  
 logic [4 : 0] ADRX;  
 logic [4 : 0] ADRY;  
 logic [7 : 0] DX\_OUT;  
 logic [7 : 0] DY\_OUT;  
 register\_file DUT(.\*);  
   
 always   
 begin   
 CLK = 0; #50;  
 CLK = 1; #50;  
 end  
 initial begin   
 int i;   
 RF\_WR = 1;  
 for (i =0; i<32; i++)   
 begin  
 DIN = i;  
 ADRX = i;  
 #100;  
 end  
 RF\_WR = 0;  
 for (i =0; i<32; i++)   
 begin  
 DIN = i;  
 ADRY = i;  
 #100;  
endmodule

*Figure 7: Register File Test Bench Code*

**Scratch Ram:** We did two test. One for writing and one for reading.

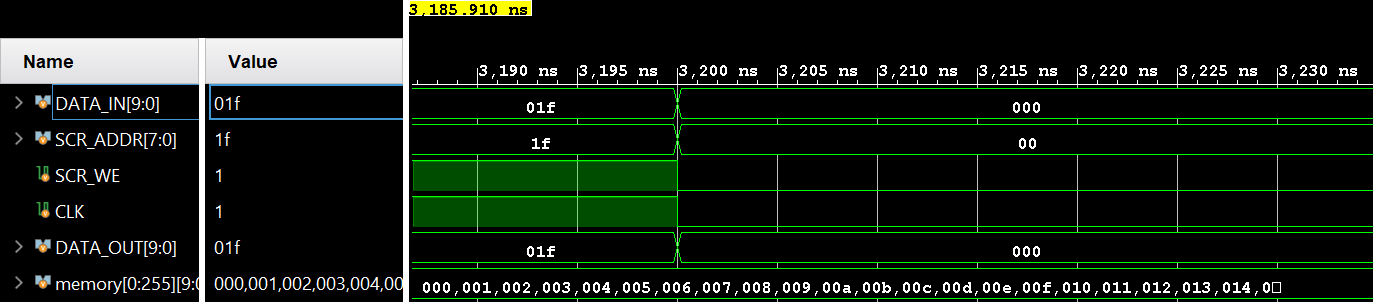
***Test 1:*** We tested for writing to all memory locations and, reading of output.

***Test 2:*** We test for reading of output without writing to memory locations.

***Test Table:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test #** | **SCR\_WE** | **DATA\_IN** | **SCR\_ADDR** | **Explanation** |
| **1** | **1** | **i** | **i** | **Testing writing and reading to memory** |
| **2** | **0** | **i** | **i** | **Testing reading but not writing to memory** |

*Figure 8: Scratch Ram Test Table*



*Figure 9: Scratch Ram Simulation*

***Scratch Ram Test Bench Code:***

module scratch\_ram\_tb();  
 logic [9:0] DATA\_IN;  
 logic [7:0] SCR\_ADDR;  
 logic SCR\_WE, CLK;  
 logic [9:0] DATA\_OUT;  
 scratch\_ram DUT(.\*);  
   
 always   
 begin   
 CLK = 0; #50;  
 CLK = 1; #50;  
 end  
 initial begin   
 int i;   
 SCR\_WE = 1;  
 for (i =0; i<32; i++)   
 begin  
 DATA\_IN = i;  
 SCR\_ADDR = i;  
 #100;  
 end  
 SCR\_WE = 0;  
 for (i =0; i<32; i++)   
 begin  
 DATA\_IN = i;  
 SCR\_ADDR = i;  
 #100;  
 end  
 end   
endmodule

*Figure 10: Scratch File Test Bench Code*

Source Code

**Register File**

module register\_file(  
input RF\_WR, CLK,  
input [7 : 0] DIN,  
input [4 : 0] ADRX,  
input [4 : 0] ADRY,   
output [7 : 0] DX\_OUT,output [7 : 0] DY\_OUT  
);  
  
logic [7 : 0] memory [0 : 31];  
initial begin   
 int i;  
 for (i =0; i<32; i++) begin  
 memory[i] = 0;  
 end  
end  
  
always\_ff @ (posedge CLK)  
begin  
 if (RF\_WR==1)  
 memory[ADRX] = DIN;  
end  
assign DX\_OUT = memory[ADRX];  
assign DY\_OUT = memory[ADRY];  
Endmodule

*Figure 11: Register File Source Code*

**Scratch Ram**

module scratch\_ram(  
input [9:0] DATA\_IN,  
input [7:0] SCR\_ADDR,  
input SCR\_WE, CLK,  
output [9:0] DATA\_OUT  
);  
   
logic [9:0] memory [0:255];  
initial begin   
 int i;  
 for (i =0; i<256; i++) begin  
 memory[i] = 0;  
 end  
end  
  
always\_ff @ (posedge CLK)  
begin  
 if (SCR\_WE==1)  
 memory[SCR\_ADDR] = DATA\_IN;  
end  
assign DATA\_OUT = memory[SCR\_ADDR];  
Endmodule

*Figure 12: Scratch Ram Source Code*