Josh Monson  
ECEN 620 Assignment 2.1  
9/18/14

**Memory Model Code**

`default\_nettype none

**module** my\_mem**(**clk**,**

write**,**

read**,**

data\_in**,**

address**,**

data\_out**);**

**input** logic clk**;**

**input** logic write**;**

**input** logic read**;**

**input** logic **[**7**:**0**]** data\_in**;**

**input** logic **[**15**:**0**]** address**;**

**output** logic **[**8**:**0**]** data\_out**;**

// Declare a 9-bit associative array using the logic data type

logic **[**8**:**0**]** mem\_array**[**shortint**];**

**always** **@(posedge** clk**)** **begin**

**if** **(**write**)**

mem\_array**[**address**]** **=** **{^**data\_in**,** data\_in**};**

**else** **if** **(**read**)**

data\_out **=** mem\_array**[**address**];**

**end**

endmodule

**Test Bench Code**`default\_nettype none  
`timescale 1ns**/**100ps

**module** testbench **();**

logic clk**;**

logic write**;**

logic read**;**

logic **[**7**:**0**]** data\_in**;**

logic **[**15**:**0**]** address**;**

logic **[**8**:**0**]** data\_out**;**

shortint address\_array**[];**

byte data\_to\_write\_array**[];**

bit **[**8**:**0**]** data\_read\_expect\_assoc**[**shortint**];**

int error\_counter**;**

bit **[**8**:**0**]** data\_read\_queue**[**$**];**

**initial** **begin**

clk **=** 0**;**

error\_counter **=** 0**;**

read **=** 0**;**

address\_array **=** new**[**6**];**

data\_to\_write\_array **=** new**[**6**];**

foreach**(**address\_array**[**i**])** **begin**

address\_array**[**i**]** **=** $random**;**

data\_to\_write\_array**[**i**]** **=** $random**;**

data\_read\_expect\_assoc**[**address\_array**[**i**]]** **=** **{^**data\_to\_write\_array**[**i**],** data\_to\_write\_array**[**i**]};**

**end**

foreach**(**address\_array**[**i**])** **begin**

write\_to\_memory**(**address\_array**[**i**],** data\_to\_write\_array**[**i**]);**

**end**

address\_array**.**reverse**();**

foreach**(**address\_array**[**i**])** **begin**

**@(negedge** clk**)**

read **=** 1**;**

address **=** address\_array**[**i**];**

**@(posedge** clk**)**

**#**1

data\_read\_queue**.**push\_back**(**data\_out**);**

**if(**data\_read\_expect\_assoc**[**address**]** **!=** data\_out**)** **begin**

$display**(**"Found Error: %03X %03X"**,** data\_read\_expect\_assoc**[**address**],** data\_out**);**

error\_counter **=** error\_counter **+** 1**;**

**end**

**end**

$display**(**"Print Read Values"**);**

foreach**(**data\_read\_queue**[**i**])** **begin**

$display**(**"\t%03X"**,** data\_read\_queue**[**i**]);**

**end**

$display**(**"Error Count: %d"**,** error\_counter**);**

**end**

**task** write\_to\_memory**(input** shortint addr**,** byte data**);**

write **=** 1**;**

data\_in **=** data**;**

address **=** addr**;**

**@(posedge** clk**);**

write **=** 0**;**

**@(negedge** clk**);**   
 **endtask;**

**always** **begin**

**#**5 clk **=** **~**clk**;**

**end**

my\_mem mem**(**clk**,**

write**,**

read**,**

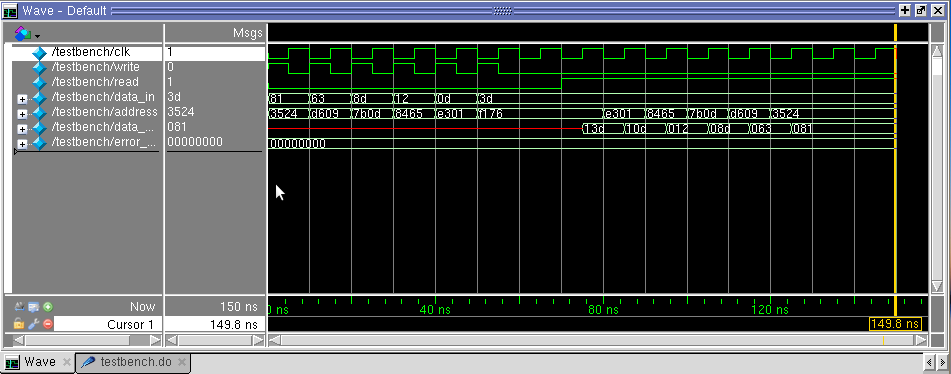
data\_in**,**

address**,**

data\_out**);**

endmodule

**Screen Shot of Waveform**

****

**Working Transcript Window**

vsim -novopt testbench

# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment2.1/work.testbench  
# Loading sv\_std.std  
# Loading work.testbench  
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment2.1/work.my\_mem  
# Loading work.my\_mem

# Print Read Values  
# 13d  
# 10d  
# 012  
# 08d  
# 063  
# 081

# Error Count: 0

**Non-Working Transcript Window**

# vsim -novopt testbench   
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment2.1/work.testbench  
# Loading sv\_std.std  
# Loading work.testbench  
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment2.1/work.my\_mem  
# Loading work.my\_mem  
# Found Error: 13d 000  
# Found Error: 10d 000  
# Found Error: 012 000  
# Found Error: 08d 000  
# Found Error: 063 000  
# Found Error: 081 000

# Print Read Values

# 000  
# 000  
# 000  
# 000  
# 000  
# 000

# Error Count: 6