Joshua Monson
Assignment 3
9/30/14

my_mem 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;

//This Function Calculates Even Parity
function automatic logic ev_parity(logic [7:0] din);
  return (^din);
endfunction

   // 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] = {ev_parity(data_in), data_in};
      else if (read)
        data_out =  mem_array[address];
   end

endmodule
```

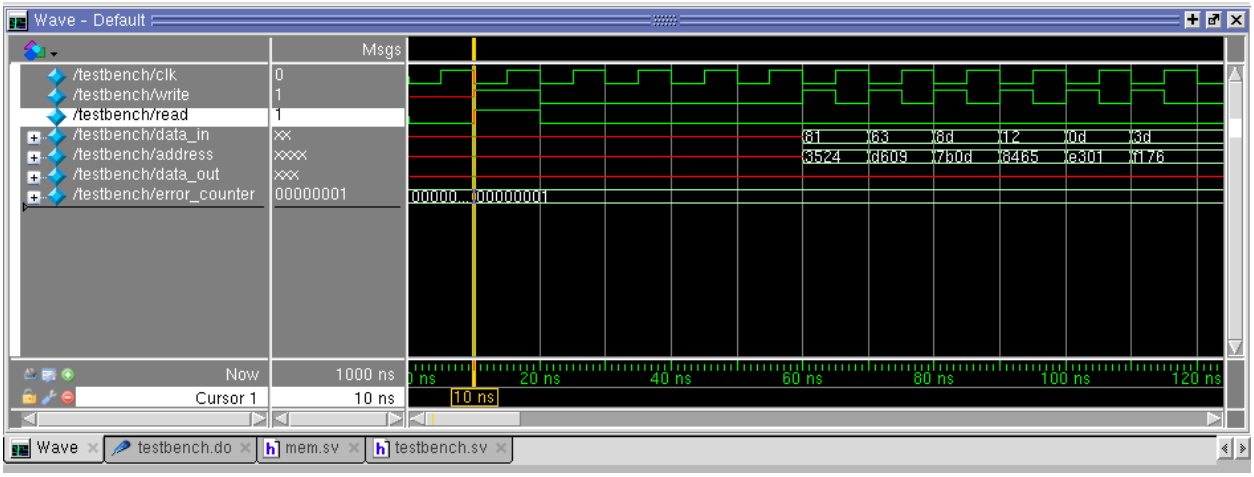
testbench code:

```
`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[$];

  ////////////////////////////
  //   Read Write Checker   //
  ////////////////////////////
  function automatic logic ReadWriteChecker(logic rd, logic wr);
    if(rd == 1 && wr == 1) begin
      $display("Error@%d: Read and Write asserted at the same time\n",
$time);
      error_counter++;
    end
  endfunction

  ////////////////////////////
  //Test Read Write Checker//
  ////////////////////////////
  always_comb begin
    ReadWriteChecker(read, write);
  end

  ////////////////////////////
  //    Test Bench Code     //
  ////////////////////////////
  initial begin
    clk = 0;
    error_counter = 0;
    read = 0;
    address_array = new[6];
    data_to_write_array = new[6];

    //Test Checker
    @(negedge clk);
    read = 1;
    write = 1;
    @(negedge clk);
    read = 0;
    write = 0;

    /////////////////
    //Do While Loop//
    /////////////////
    do
    begin
      $display("Waiting...");
      #10ns;
     end
    while ($time < 60 );

    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
```

Waveform:



Transcripts:

Error Free:

vsim -novopt testbench

Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment3/work.testbench
Loading sv_std.std
Loading work.testbench
Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment3/work.my_mem
Loading work.my_mem

Waiting...
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...

Print Read Values

```
#       13d
#       10d
#       012
#       08d
#       063
#       081

# Error Count:          0
```

Read = 1 Write =1 Error

```
# Refreshing
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment3/work.t
estbench

# Loading sv_std.std
# Loading work.testbench
# Refreshing
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment3/work.
my_mem
# Loading work.my_mem
# Error@                10: Read and Write asserted at the same time
#
# Waiting...
# Waiting...
# Waiting...
# Waiting...

# Print Read Values

#       13d
#       10d
#       012
#       08d
```

063
081

Error Count: 1

Data Read Error

vsim -novopt testbench

Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment3/work.testbench

Loading sv_std.std
Loading work.testbench
Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment3/work.my_mem

Loading work.my_mem
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...
Waiting...

 # 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
```