Joshua Monson
Assignment 4.2
10/4/2014

Overview

This report details the changes made to the memory module interface and the test bench in order to test out new (to me) functionality of System Verilog. The main changes that have been made are:

1. Added a Clocking Block to the memory interface.
2. Separated my test bench into a program and a top level module.
3. Changed my protocol checker from an interface function to an assertion
4. Changed the clock speed from 100 MHz to 10 MHz

This report contains all of the System Verilog source code used in this project. Additionally, I have included waveforms and console transcripts for both the 0 delay memory model and the 75ns delay memory model.

Code: Interface

```
interface mem_if(input bit clk);
   logic write;
   logic read;
   logic [7:0]  data_in;
   logic [15:0] address;
   logic [8:0]  data_out;

   clocking cb @(posedge clk);
     output write;
     output read;
     output address;
     input data_out;
     output data_in;
   endclocking

   modport TEST(clocking cb);
   modport DUT(input clk, write, read, address, data_in, parity,
               output data_out);

   a1: assert property (@(posedge clk) !(cb.read==1 && cb.write==1));


   ////////////////////////////
   //    Even Parity Calc.   //
   ////////////////////////////
   function logic parity(logic [7:0] din);
     return (^din);
   endfunction

endinterface
```

Code: Top Module

```
//////////////////////
// Top-Level Module //
//////////////////////

module top;

  //Clock Generator
  bit clk = 0;
  always #50ns clk = ~clk;

  //Memory Interface
  mem_if mem_bus(clk);
  //Test Program
  test t1(mem_bus);
  //Memory Model
  my_mem mem(mem_bus);

endmodule
```

Code: Program Test

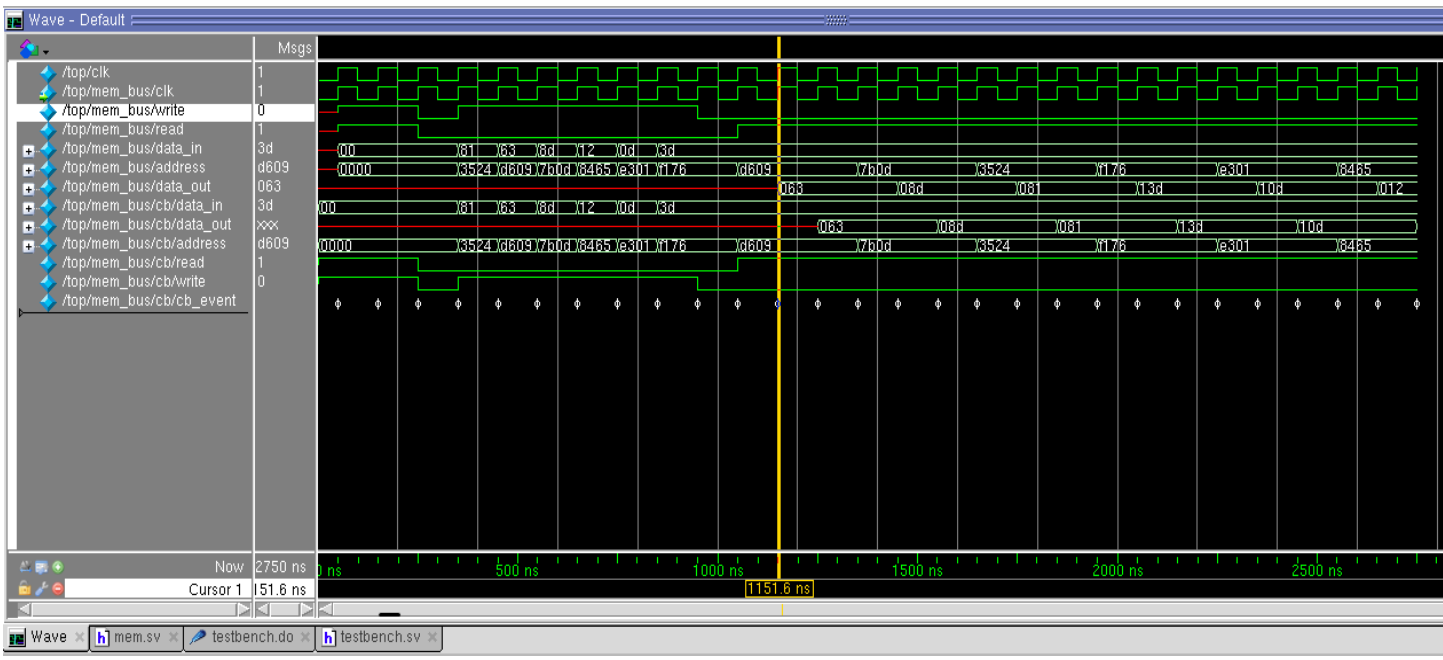
```
program automatic test(mem_if.TEST mem_bus);

  shortint address;
  byte data;
  int ErrorCount = 0;

  typedef struct {
      shortint address;
      byte data_to_write;
      bit [8:0] expected_read;
      bit [8:0] actual_read;
  } ScoreBoardEntry;

  ScoreBoardEntry ScoreBoard[];

  initial begin

    //Initialize Score Board
    ScoreBoard = new[6];

    //Generate Random Writes
    foreach(ScoreBoard[i]) begin
      address = $random;
      data = $random;
      ScoreBoard[i].address = address;
      ScoreBoard[i].data_to_write = data;
      ScoreBoard[i].expected_read = {^data, data};
    end

    mem_bus.cb.read <= 0;
```

```
  //Test Protocol Error Checker
  //This should produce three Errors
  mwrite(0, 0);
  mread(0);

  @mem_bus.cb;
  @mem_bus.cb
  @mem_bus.cb

  mem_bus.cb.write <= 0;
  mem_bus.cb.read  <= 0;

  //Perform Writes
  foreach(ScoreBoard[i]) begin
    write_memory(ScoreBoard[i]);
  end
  @(mem_bus.cb);
  mem_bus.cb.write <= 0;

  //Rearrange Array for Reads
  ScoreBoard.shuffle();

  //Perform Reads and Check Results
  foreach(ScoreBoard[i]) begin
   read_memory(ScoreBoard[i]);
   check(ScoreBoard[i]);
  end
  mem_bus.cb.read <= 0;

  //Display the Actual Values Read
  $display("Print Read Values");
  foreach(ScoreBoard[i]) begin
   $display("\t%03X", ScoreBoard[i].actual_read);
  end

  $display("Tests Performed: %d", ScoreBoard.size());
  $display("Incorrect Read Errors: %d", ErrorCount);
  //$display("3.Read/Write Signal Errors: %d", mem_bus.rdwr_error_count);

end

final begin
  $display("Test Complete.");
  $stop();
end

////////////////////////////
//     Read Function      //
////////////////////////////
//Note: does not return read value
function void mread(logic [15:0] addr);
 mem_bus.cb.read <= 1;
 mem_bus.cb.address <= addr;
endfunction

////////////////////////////
//    Write Function      //
```

```
  ////////////////////////////
  function void mwrite(logic [15:0] addr, logic [7:0] data);
   mem_bus.cb.write <= 1;
   mem_bus.cb.data_in <= data;
   mem_bus.cb.address <= addr;
  endfunction


  task automatic write_memory(ref ScoreBoardEntry Entry);
    @(mem_bus.cb);
    mwrite(Entry.address, Entry.data_to_write);
  endtask

  task automatic read_memory(ref ScoreBoardEntry Entry);
    @(mem_bus.cb);
    mread(Entry.address);
    @(mem_bus.cb);
    @(mem_bus.cb);
    Entry.actual_read = mem_bus.cb.data_out;
  endtask

  function check(ScoreBoardEntry Entry);
    if( Entry.expected_read != Entry.actual_read) begin
      $display("Found Error: %04X %04X", Entry.expected_read,
Entry.actual_read);
      ErrorCount++;
    end
  endfunction

endprogram
```

Program Output for Zero Delay Memory Model:

Waveform:



Transcript with Zero Memory Read Errors:

```
# Test Complete.

# vsim -novopt top
# Refreshing
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.top

# Loading sv_std.std
# Loading work.top
# Refreshing
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.mem_if

# Loading work.mem_if
# Refreshing
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.test
# Loading work.test
# Refreshing
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.my_mem
# Loading work.my_mem
# Print Read Values

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

# Tests Performed:        6
# Incorrect Read Errors:        0
```

Transcript with One or More Memory Read Errors:

```
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.top
# Loading sv_std.std
# Loading work.top
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.mem_if

# Loading work.mem_if
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.test
```

```
# Loading work.test
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.my_mem
# Loading work.my_mem

# Found Error: 0063 0000
# Found Error: 008d 0000
# Found Error: 0081 0000
# Found Error: 013d 0000
# Found Error: 010d 0000
# Found Error: 0012 0000

# Print Read Values

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

# Tests Performed:         6
# Incorrect Read Errors:         6
```

Transcript with One or More Assertion Errors:

```
Test Complete.

# vsim -novopt top
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.top
# Loading sv_std.std
# Loading work.top
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.mem_if
# Loading work.mem_if
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.test
# Loading work.test
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.my_mem
# Loading work.my_mem

# ** Error: Assertion error.
#    Time: 50 ns Started: 50 ns  Scope: top.mem_bus.a1 File:
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/sverilog/mem.sv Line: 23

# ** Error: Assertion error.
#    Time: 150 ns Started: 150 ns  Scope: top.mem_bus.a1 File:
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/sverilog/mem.sv Line: 23
```

 # ** Error: Assertion error.
Time: 250 ns Started: 250 ns Scope: top.mem_bus.a1 File: /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/sverilog/mem.sv Line: 23

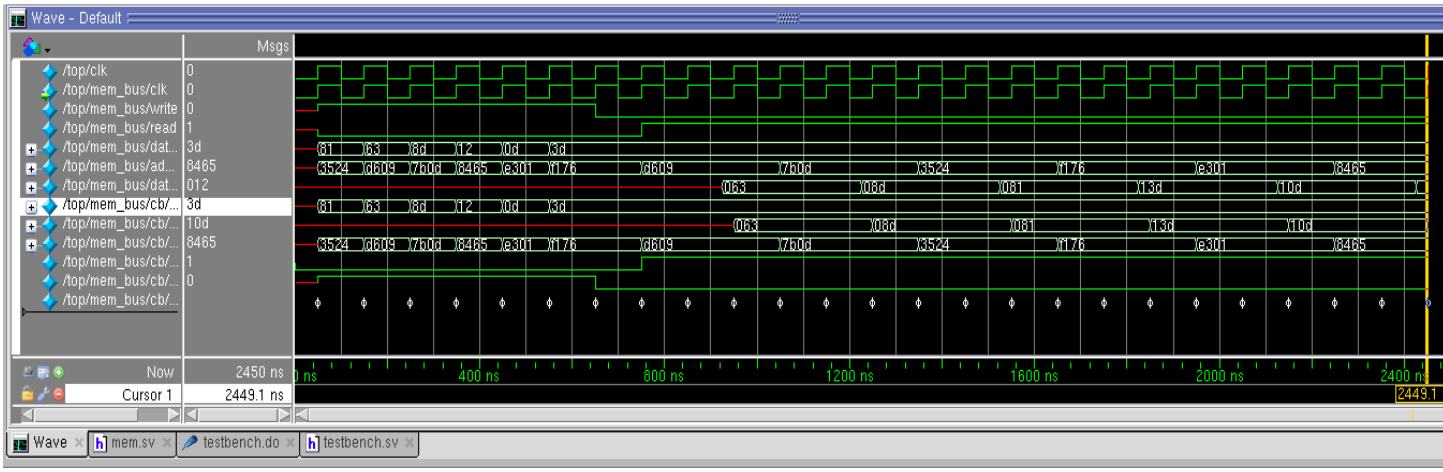
Print Read Values

063
08d
081
13d
10d
012

Tests Performed: 6
Incorrect Read Errors: 0

Program Output for 75ns Delay Memory Model:

Waveform:



Transcript with Zero Memory Read Errors:

Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.top

Loading sv_std.std
Loading work.top
Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.mem_if
Loading work.mem_if
Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.test
Loading work.test

```
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.my_mem
# Loading work.my_mem


# Print Read Values

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

# Tests Performed:        6
# Incorrect Read Errors:        0
```

Transcript with One or More Memory Read Errors:

```
# Test Complete.

# vsim -novopt top
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.top
# Loading sv_std.std
# Loading work.top
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.mem_if
# Loading work.mem_if
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.test
# Loading work.test
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.my_mem

# Loading work.my_mem

# Found Error: 0063 0000
# Found Error: 008d 0000
# Found Error: 0081 0000
# Found Error: 013d 0000
# Found Error: 010d 0000
# Found Error: 0012 0000

# Print Read Values

#       000
#       000
#       000
#       000
```

```
#         000
#         000

# Tests Performed:         6
# Incorrect Read Errors:         6
```

Transcript with One or More Assertion Errors:

```
# Test Complete.

# vsim -novopt top
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.top

# Loading sv_std.std
# Loading work.top
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.mem_if
# Loading work.mem_if
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.test
# Loading work.test
# Refreshing /net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/work.my_mem
# Loading work.my_mem

# ** Error: Assertion error.
#    Time: 50 ns Started: 50 ns  Scope: top.mem_bus.a1 File:
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/sverilog/mem.sv Line: 23
# ** Error: Assertion error.
#    Time: 150 ns Started: 150 ns  Scope: top.mem_bus.a1 File:
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/sverilog/mem.sv Line: 23
# ** Error: Assertion error.
#    Time: 250 ns Started: 250 ns  Scope: top.mem_bus.a1 File:
/net/fpga1/users/joshuas2/ECEn620/Assignments/Assignment4.2/sverilog/mem.sv Line: 23

# Print Read Values

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

# Tests Performed:         6
# Incorrect Read Errors:         0
```