Amrita Vishwa Vidyapeetham
Amrita School of Computing, Bangalore
Department of Computer Science and Engineering
19CSE211 Computer Organization and Architecture
Lab Handout - 9
MIPS Memory

Exercise Problems

1. Write a verilog code to implement positive edge triggered write operation in a byte memory with address size 6 bits and data size 4 bits.

```
module write_mem(clk,
write_enable, rst, datain, addr,
q);
input clk, write_enable, rst;
input [5:0] addr;
input [3:0] datain;
output [3:0] q;
reg [3:0] memory [63:0];
integer i;
always @ (posedge clk)
begin
     if(rst==1)
     begin
          for (i = 0; i < 64; i =
i + 1)
          begin
               memory[i] =
4'b0000;
          end
     end
     if(write_enable==1)
     begin
          memory[addr] = datain;
     end
end
assign q = memory[addr];
endmodule
```

```
module write_mem_tb;
reg [5:0] addr;
reg clk, write_enable, rst;
reg [3:0] datain;
wire [3:0] q;

write_mem t1(clk,
write_enable, rst, datain,
addr, q);
always #5 clk = ~clk;
initial
begin
     clk = 1'b0;
write_enable = 1'b0; rst =
1'b1;
     $monitor("Time=%f,
clk=%b, rst =%b,
write_enable=%b addr=%6b
datain=%4b q=%4b", $time,
clk, rst, write_enable,
addr, datain, q);
     #10 rst = 1'b0; addr =
0; write_enable=1'b1;
datain=4'b1111;
     #10 rst = 1'b0; addr =
4; write_enable=1'b1;
datain=4'b1010;
     #40 $finish;
end

endmodule
```

2. Write a verilog code to implement positive edge triggered read operation in a byte memory with address size 6 bits and data size 4 bits.

```
module read_mem(clk, rst, addr,
dataout);
input clk, rst;
input [5:0] addr;
output reg [3:0] dataout;
reg [3:0] memory [63:0];
integer i;
always @ (posedge clk)
begin
     if(rst==1)
     begin
          for (i = 0; i < 64; i =
i + 1)
          begin
               memory[i] =
4'b0000;
          end
     end
end
always @ (posedge clk)
begin
     dataout = memory[addr];
end
endmodule
```

```
module read_reg_tb;
reg [5:0] addr;
reg clk, rst;
wire [3:0] dataout;

read_mem t1(clk, rst, addr,
dataout);

always #5 clk = ~clk;

initial
begin
     clk = 1'b0; rst =
1'b1;
     $monitor("Time=%f,
clk=%b, rst =%b, addr=%6b,
dataout=%4b", $time, clk,
rst, addr, dataout);
     #10 rst = 1'b0; addr =
0;
     #20 $finish;
end

endmodule
```

Assignment

1. Write a verilog code to implement MIPS byte organized memory with address size 6 bits and data size 4 bits. The read and write operations should be performed in positive edge of the clock.