Amrita Vishwa Vidyapeetham
Amrita School of Computing, Bangalore
Department of Computer Science and Engineering
19CSE211 Computer Organization and Architecture
Lab Handout - 6
Arithmetic Logic Unit

Exercise Problems

1. Write a verilog code to implement positive edge triggered write operation in a register file with 16 4 bit registers.

```
module write_reg(clk,
write_enable, rst, rw, busw, q);
input clk, write_enable, rst;
input [3:0] rw;
input [3:0] busw;
output [3:0] q;
reg [3:0] register_file [15:0];
integer i;
always @ (posedge clk)
begin
     if(rst==1)
     begin
          for (i = 0; i < 16; i =
i + 1)
          begin
               register_file[i] =
4'b0000;
          end
     end
     if(write_enable==1)
     begin
          register_file[rw] =
busw;
     end
end
assign q=register_file[rw];
endmodule
```

```
module write_reg_tb;
reg [3:0] rw;
reg clk, write_enable, rst;
reg [3:0] busw;
wire [3:0] q;
write_reg t1(clk,
write_enable, rst, rw,
busw, 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 rw=%4b
busw=%4b q=%4b", $time,
clk, rst, write_enable, rw,
busw, q);
     #10 rst = 1'b0; rw =
0; write_enable=1'b1;
busw=4'b1111;
     #10 rst = 1'b0; rw =
4; write_enable=1'b1;
busw=4'b1010;
     #40 $finish;
end
endmodule
```

2. Write a verilog code to implement negative edge triggered read operation in a register file with 16 4 bit registers.

```
module read_reg(clk, rst, ra, rb,
busa, busb);
input clk, rst;
input [3:0] ra, rb;
output reg [3:0] busa, busb;
reg [3:0] register_file [15:0];
integer i;
always @ (posedge clk)
begin
     if(rst==1)
     begin
          for (i = 0; i < 16; i =
i + 1)
          begin
               register_file[i] =
4'b0000;
          end
     end
end
always @ (negedge clk)
begin
     busa = register_file[ra];
     busb = register_file[rb];
end

endmodule
```

```
module read_reg_tb;
reg [3:0] ra, rb;
reg clk, rst;
wire [3:0] busa, busb;

read_reg t1(clk, rst, ra,
rb, busa, busb);

always #5 clk = ~clk;

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

endmodule
```

Assignment

1. Write a verilog code to implement MIPS register file of 16 4bit register with transparent register operations.