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 gate level modeling of a 4-bit added/subtractor.

```
module full_adder(a, b, cin, sum,
cout);
input a, b, cin ;
output sum, cout;
wire w1, w2, w3;
xor g1(w1, a, b);
xor g2(sum, w1, cin);
and g3(w2, a, b);
and g4(w3, w1, cin);
or g5(cout, w2, w3);
endmodule

module add_sub_4bit(a, b, sum, cout,
ctrl);
input [3:0] a, b;
input ctrl;
output [3:0] sum;
output cout;
wire [3:0] wb;
wire [2:0] carryw;
xor g1(wb[0], ctrl, b[0]);
xor g2(wb[1], ctrl, b[1]);
xor g3(wb[2], ctrl, b[2]);
xor g4(wb[3], ctrl, b[3]);
full_adder a0(a[0], wb[0], ctrl,
sum[0], carryw[0]);
full_adder a1(a[1], wb[1], carryw[0],
sum[1], carryw[1]);
full_adder a2(a[2], wb[2], carryw[1],
sum[2], carryw[2]);
full_adder a3(a[3], wb[3], carryw[2],
sum[3], cout);

endmodule
```

```
module add_sub_4bit_tb;
reg [3:0] a, b;
reg ctrl;
wire [3:0] sum;
wire cout;

add_sub_4bit i1(a, b, sum,
cout, ctrl);

initial
begin
      a = 4'b0000;
      b = 4'b0000;
      ctrl = 1'b0;
      $monitor("Time:%0t
ctrl=%b a=%4b b=%4b sum=%4b
cout=%b", $time, ctrl, a, b,
sum, cout);
      #5 a = 4'b0001;
b=4'b0010; ctrl = 1'b1;
      #5 a = 4'b0111;
b=4'b0001; ctrl = 1'b0;
      #5 a = 4'b1111;
b=4'b0001; ctrl = 1'b1;
      #5 a = 4'b1001;
b=4'b0001; ctrl = 1'b0;
end
endmodule
```

2. Write a verilog code to implement data flow modeling of a 4-bit logic unit. Which can perform AND, OR, NOT and XOR operation based on the control input 00-AND, 01-OR, 10-NOT and 11-XOR.

```
module logical_unit_4bit(input1,
input2, out, ctrl);
input [3:0] input1, input2;
input [1:0] ctrl;
output [3:0] out;

assign out = ctrl[0]?(ctrl[1]?
(input1 ^ input2): (input1 |
input2)) : (ctrl[1]?(~input1):
(input1 & input2));

endmodule
```

```
module
logical_unit_4bit_tb;

reg [3:0] input1, input2;
reg [1:0] ctrl;
wire [3:0] out;

logical_unit_4bit
i1(input1, input2, out,
ctrl);

initial
begin
     input1 = 4'b0000;
     input2 = 4'b0000;
     ctrl = 2'b00;
     $monitor("Time:%0t
ctrl=%b input1=%4b
input2=%4b output=%4b",
$time, ctrl, input1,
input2, out);
     #5 input1 = 4'b1010;
input2 = 4'b1010; ctrl =
2'b00;
     #5 input1 = 4'b0011;
input2 = 4'b0111; ctrl =
2'b01;
     #5 input1 = 4'b0000;
ctrl = 2'b10;
     #5 input1 = 4'b1111;
input2 = 4'b1111; ctrl =
2'b11;
end

endmodule
```

Assignment

1. Write a verilog code to implement a 4 bit ALU unit. Which performs Add, Sub, AND, OR, NOR, XOR operations. Refer the table below for the ALU_Ctrl for each operation.

Hint: Use the modules from Exp1 and Exp2

ALU Control

ALU Control Value	Operation
0XX	Add
1XX	Sub
X00	AND
X01	OR
X10	NOT
X11	XOR