

## LAB Final 3108

Problem 1. Design a 4-bit ALU circuit that supports the following operations with the required flags.

AND(00), SUB(01), RIGHT SHIFT(10), NOT(11)

Problem 2. Write an HDL program to perform the right rotate operation on a 4 bit operand.

HDL Code:

// HDL program to perform the right rotate operation on a 4 bit operand

```
module right_rotate (  
    input [3:0] data_in,  
    input rotate_en,  
    output reg [3:0] data_out  
);  
    always @(*) begin  
        if (rotate_en)  
            data_out = {data_in[0], data_in[3:1]};  
        else  
            data_out = data_in;  
        end  
    endmodule
```

Test Bench:

// HDL module to test the right\_rotate module

module Tests\_right\_rotate;

```
    reg A;  
    reg B;  
    reg C;  
    reg D;  
    wire [3:0] S;
```

```
    right_rotate UUT (  
        .data_in({A, B, C, D}),  
        .rotate_en(1'b1),  
        .data_out(S)  
    );
```

```
    initial begin
```

A = 0;  
B = 0;  
C = 0;  
D = 0;  
#20  
A = 0;  
B = 0;  
C = 0;  
D = 1;  
#20  
A = 0;  
B = 0;  
C = 1;  
D = 0;  
#20  
A = 0;  
B = 0;  
C = 1;  
D = 1;  
#20  
A = 0;  
B = 1;  
C = 0;  
D = 0;  
#20  
A = 0;  
B = 1;  
C = 0;  
D = 1;  
#20  
A = 0;  
B = 1;  
C = 1;  
D = 0;  
#20  
A = 0;  
B = 1;  
C = 1;  
D = 1;  
#20  
A = 1;  
B = 0;  
C = 0;  
D = 0;  
#20  
A = 1;  
B = 0;  
C = 0;  
D = 1;  
#20  
A = 1;  
B = 0;

```

    C = 1;
    D = 0;
    #20
    A = 1;
    B = 0;
    C = 1;
    D = 1;
    #20
    A = 1;
    B = 1;
    C = 0;
    D = 0;
    #20
    A = 1;
    B = 1;
    C = 0;
    D = 1;
    #20
    A = 1;
    B = 1;
    C = 1;
    D = 0;
    #20
    A = 1;
    B = 1;
    C = 1;
    D = 1;
end

initial begin
    $monitor("A=%b,B=%b,C=%b,D=%b,S=%b\n", A, B, C, D, S);
end

endmodule

```

Output: