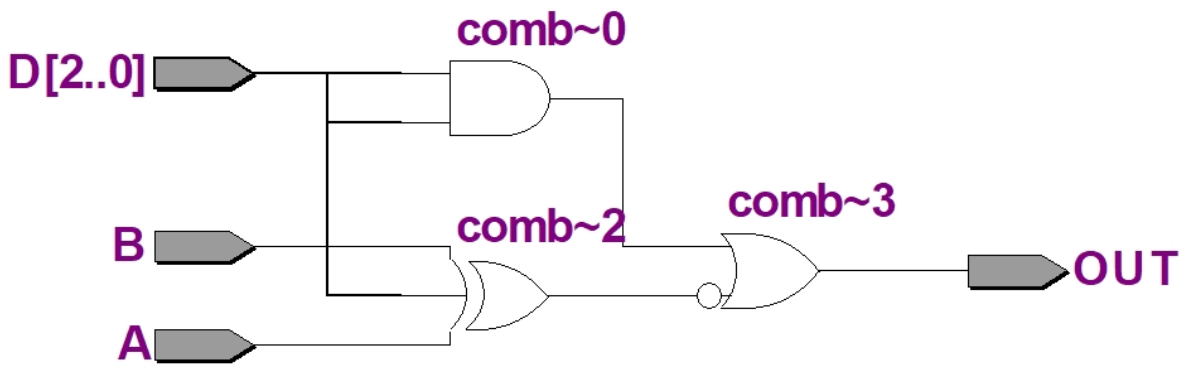
Q1

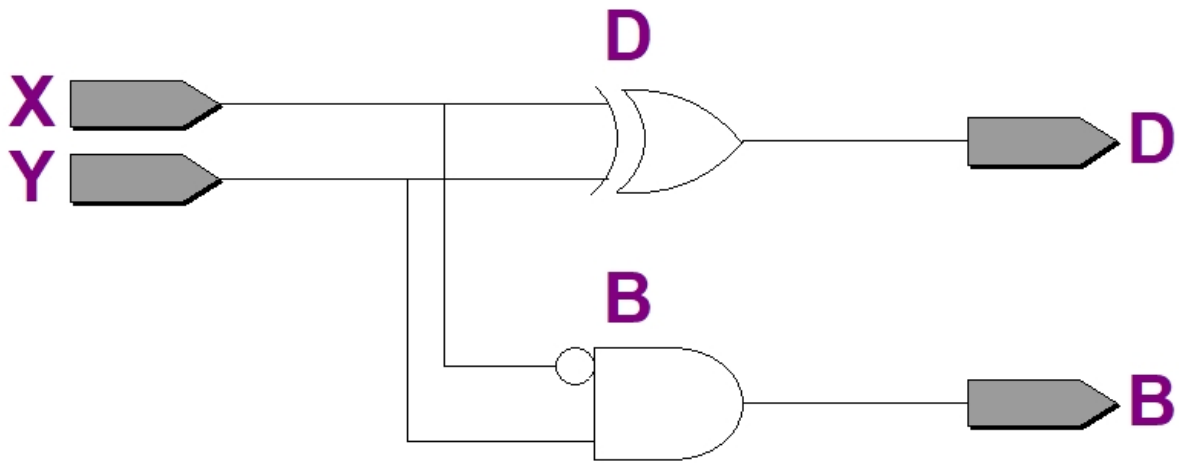
```
module block1(input[2:0] D,input A,B, output OUT);
or(OUT,D[0]&D[1],!(D[2]^A^B));
endmodule
```



Q2

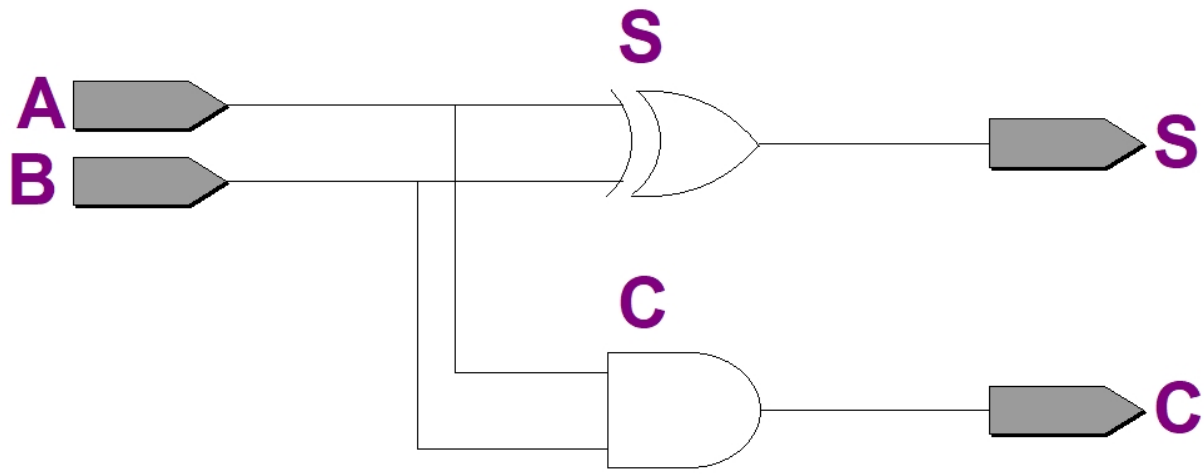
1

```
module half_sub(input X,Y,output D,B);
wire S1;
assign D = X^Y;
assign S1 = !X;
assign B = S1 & Y;
endmodule
```



2

```
module half_add(input A,B, output S,C);
assign S = A^B;
assign C = A&B;
endmodule
```



3

Top module

```
module add_or_sub(input [1:0] IN, input SEL, output SUM, CARRY);
wire ca,cs,sa,ss,c;
half_add add(IN[1],IN[0],sa,ca);
```

```
half_sub sub(IN[1],IN[0],ss,cs);
mux_2x1 mux_sum(ss,sa,!SEL,SUM);
mux_2x1 mux_carry(!cs,!ca,!SEL,c);
assign CARRY = !c;
endmodule
```

2x1 mux:

```
module mux_2x1(input a,b,sel,  output out);
assign out = (sel)?b:a;
endmodule
```

half adder and half subtractor are above