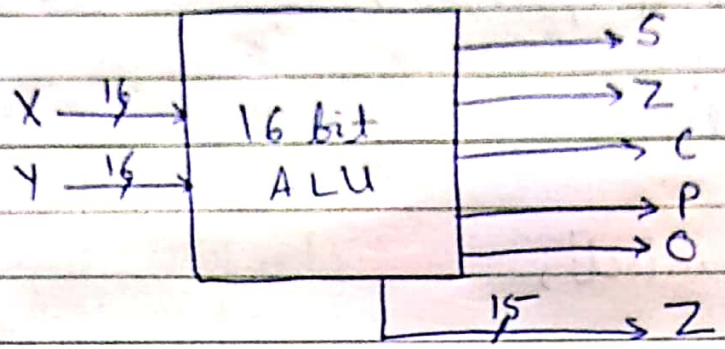
Lecture - 7

☆ Example: A 16 bit ALU

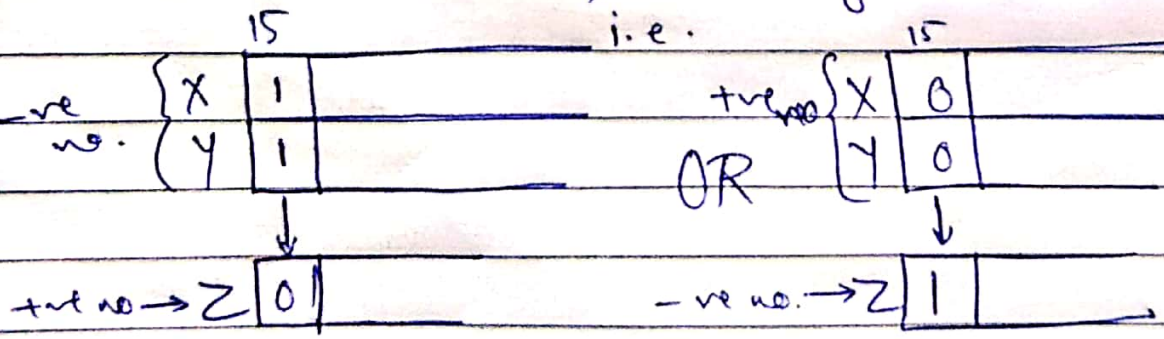



```
module ALU(X,Y,Z,sign,zero,carry,parity,overflow);
  input [15:0] X,Y;
  output [15:0] Z;
  output sign,zero,carry,parity,overflow;
  assign {carry,Z} = X+Y;
  assign sign = Z[15];
  assign zero = ~|Z;
  assign parity = ~^Z;
  assign overflow = (X[15] & Y[15] & ~Z[15]) |
                    (~X[15] & ~Y[15] & Z[15]);
endmodule
```

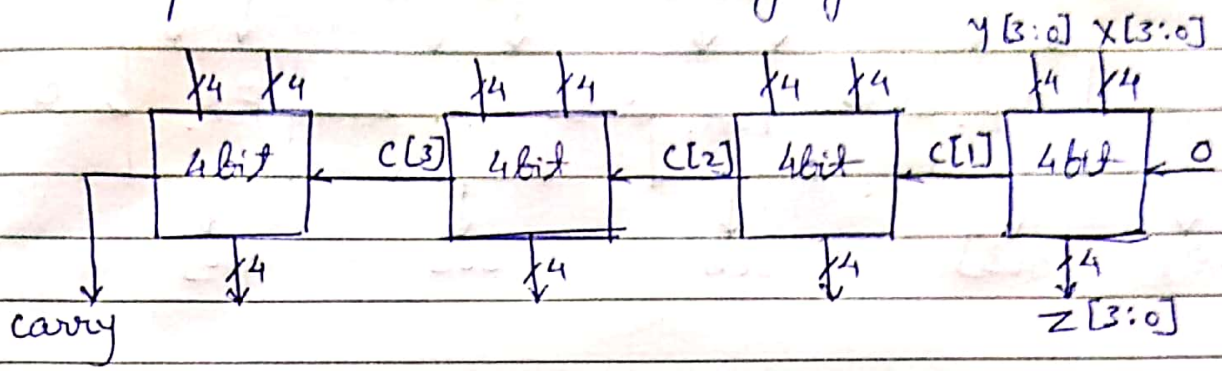
→ overflow occures in 2 conditions

1) Addition of 2 -ve nos. is +ve no.
2) Addition of 2 +ve nos. is -ve no.

i.e.



✱ Example : 16 bit adder using four 4 bit adders




```
module adder16 (X,Y,Z, carry);
    input [15:0] X,Y;
    output [15:0] Z;
    output carry;
    wire C[3:1];
    adder4  A0(Z[3:0], C[1], X[3:0], Y[3:0], 1'b0);
    adder4  A1(Z[7:4], C[2], X[7:4], Y[7:4], C[1]);
    adder4  A2(Z[11:8], C[3], X[11:8], Y[11:8], C[2]);
    adder4  A3(Z[15:12], carry, X[15:12], Y[15:12], C[3]);
endmodule
```

✱ Example: 4 bit adder using behavioral style

```
module adder4 (S, cout, A, B, Cin);
    input [3:0] A,B;
    output [3:0] S;
    input Cin;
    output Cout;
    assign {Cout, S} = A + B + Cin;
endmodule
```

✱ Write a verilog code for 4 bit binary adder using structural style: