Half Adder

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

Half Subtractor

```
module  Half_Subtractor (B,D,X,Y);
            input X,Y ;
            output D,B ;
            assign  D = X^Y ;
            assign B = (~X)&Y ;
endmodule
```

Full Adder

```
module  Full_Adder (Cout,S,A,B,Cin);
            input A,B,Cin ;
            output Cout,S ;
            assign  S = (A^B)^Cin ;
            assign Cout = (A&B) | ( (A^B) & Cin );
endmodule
```

Full Subtractor

```
module  Full_Subtractor (B,D,X,Y,Z);
            input X,Y,Z ;
            output D,B ;
            assign  D = (X^Y)^Z ;
            assign B = ((~X)&Y)  | (~ (X^Y) & Z );
endmodule
```

3:8 Active High Decoder

```
module  Decoder_3to8_H (D7,D6,D5,D4,D3,D2,D1,D0,I2,I1,I0);
            input I2,I1,I0 ;
            output D7,D6,D5,D4,D3,D2,D1,D0 ;
            assign  D0 = (~I2)&(~I1)&(~I0);
            assign  D1 = (~I2)&(~I1)&(I0);
            assign  D2 = (~I2)&(I1)&(~I0);
            assign  D3 = (~I2)&(I1)&(I0);
            assign  D4 = (I2)&(~I1)&(~I0);
            assign  D5 = (I2)&(~I1)&(I0);
            assign  D6 = (I2)&(I1)&(~I0);
            assign  D7 = (I2)&(I1)&(I0);

endmodule
```

3:8 Active Low Decoder

```
module  Decoder_3to8_L (D8,D7,D6,D5,D4,D3,D2,D1,D0,I2,I1,I0);
            input I2,I1,I0 ;
            output D8,D7,D6,D5,D4,D3,D2,D1,D0 ;
            assign  D0 = ~( (~I2)&(~I1)&(~I0));
            assign  D1 = ~((~I2)&(~I1)&(I0));
            assign  D2 = ~((~I2)&(I1)&(~I0));
            assign  D3 =~( (~I2)&(I1)&(I0));
            assign  D4 = ~((I2)&(~I1)&(~I0));
            assign  D5 =~( (I2)&(~I1)&(I0));
            assign  D6 = ~((I2)&(I1)&(~I0));
            assign  D7 = ~( (I2)&(I1)&(I0));

endmodule
```