Prasann Viswanathan
190070047
Digital Circuits Lab Midsem
EE 214

channel 5

4 4 6

$A_3 A_2 A_1 A_0$ $B_3 B_2 B_1 B_0$ $Y_5 \ldots Y_0$

unused MSB of output = 0

→ BCD - Binary to Decimal Conversion. (0-15) (A) (I_0)

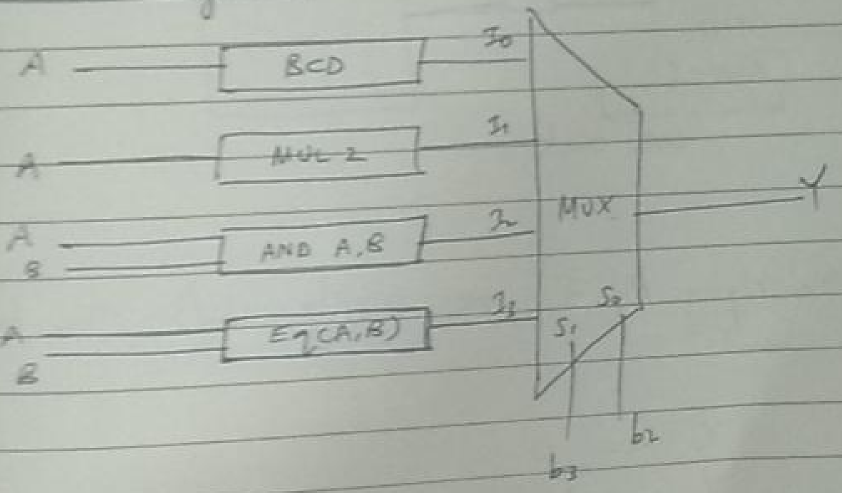
→ MUL 2 - Multiplies A by 2 (I_1)

→ AND A B - Bitwise AND A and B. (I_2)

→ Eq(A,B) - Op A when A=B otherwise 0000 (I_3)

→ MUX (I_0, I_1, I_2, I_3) $S_0 = b_2$ $S_1 = b_3$

Basic block diagram:-



Design blocks:-

→ BCD converter. 0-9 remain unchanged.
(0000 - 1001)
10-15 map as:-

1010 → 01 0000
1011 → 01 0001
1100 → 01 0010
1101 → 01 0011
1110 → 01 0100
1111 → 01 0101

Equivalent to adding 6 to each value.

190070047

0 b010
ΦDΦΦ1
1100 1

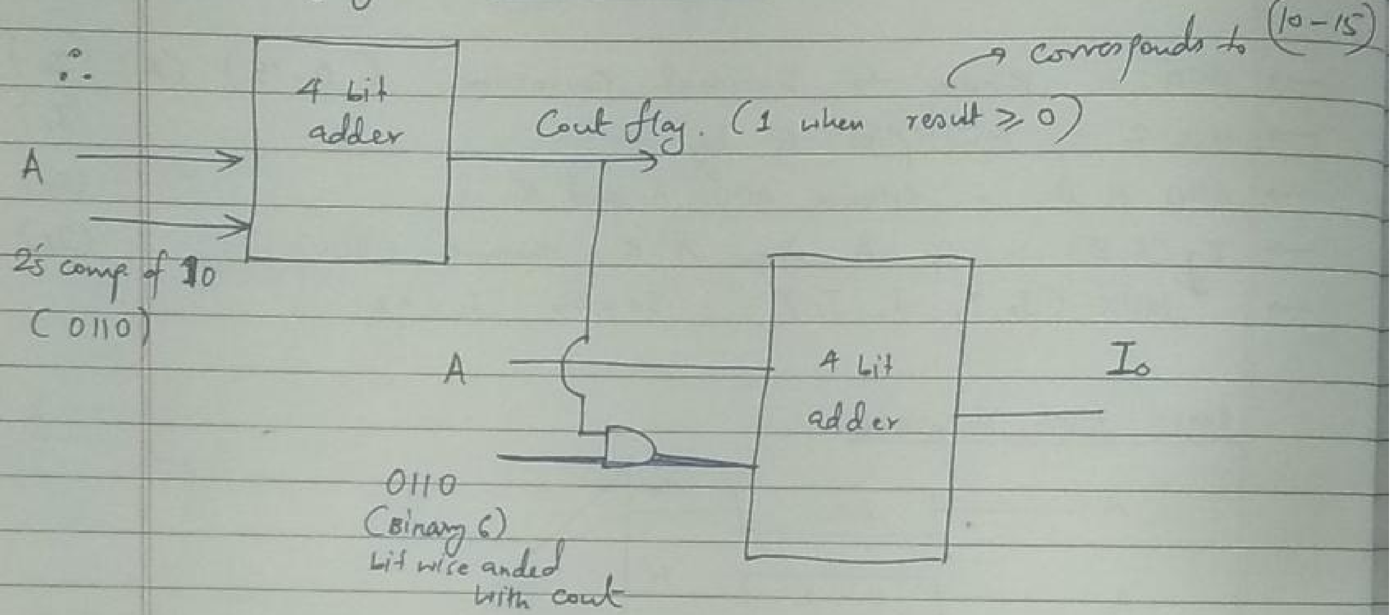
0010
0ΦΦΦ
100Φ

Logic :-

Subtract 10 from input A.

If result is negative (MSB set) ⟹ Use the carry set as a flag to add / not add 6 to A.

∴



⟶ When multiplying by 2, we simply shift the bits left by one unit.

⟹ $Y_0 <=$ '0'

$Y_1 <=$ 'A0'

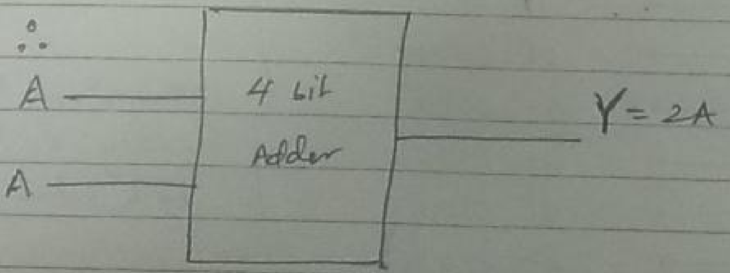
⋮

$Y_4 <= A_3$

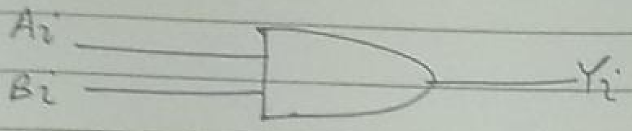
$Y_5 <=$ '0'

However, this is sequential logic. Instead we can add A to itself

∴



→ For each digit 'i' of A and B



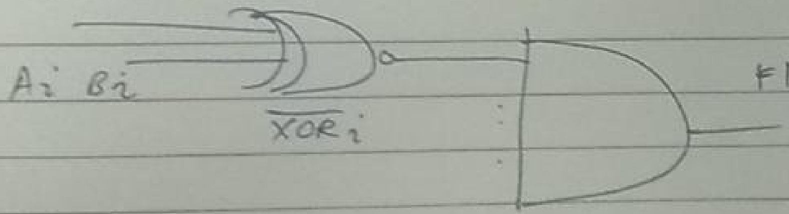
AND gate.

→ We wish to check if (A = B) bitwise.

If $A_i \neq B_i \implies 0$ we need $\overline{XOR}$

A_i	B_i	output
0	0	1
0	1	0
1	0	0
1	1	1

Collect all $\overline{XOR}$ results (4) and AND them to make flag.



if 1, Y = A = B
if 0, Y = 0.

Entity :-

```
entity midsem is
    port ( ~~Input~~ A3, A2, A1, A0, B3, B2, B1, B0 : in std_logic,
           Y5, Y4, Y3, Y2, Y1, Y0 : out std-logic) ;
end entity midsem;
```