COA LAB TEST -1

(Itish Agarwal, 18CS30021)
21 Oct' 2020

Q1)

(a) $Z = S_3 XY + S_2 X\bar{Y} + S_1 \bar{X} Y + S_0 \bar{X}\bar{Y}$

(b) (a) $S = 0110$,

$Z = X\bar{Y} + \bar{X}Y = X \oplus Y$

for $S = 0011$,

$Z = \bar{X}Y + \bar{X}\bar{Y}$

$= \bar{X}$

(c) (a) To realize XOR function,

$S = \underset{S_3}{0}\,\underset{S_2}{1}\,\underset{S_1}{1}\,\underset{S_0}{0}$

(b) To realize OR function,

$S = \underset{S_3}{1}\,\underset{S_2}{1}\,\underset{S_1}{1}\,\underset{S_0}{0}$

Q2. PTO

Q2) We have,

$$ITE(v, g, h) = vg + \bar{v}h$$

We see,

$$ITE(a, b, 0) = ab$$

$$ITE(a, 1, b) = a\cancel{b} + \bar{a}b = a + b$$

$$ITE(a, 0, 1) = \bar{a}$$

Now,

$$AOI(a, b, c, d) = \overline{(a \wedge b) \vee (c \wedge d)}$$

* $a \wedge b = ITE(a, b, 0)$
* $c \wedge d = ITE(c, d, 0)$

Then,

~~AOI(a,b~~

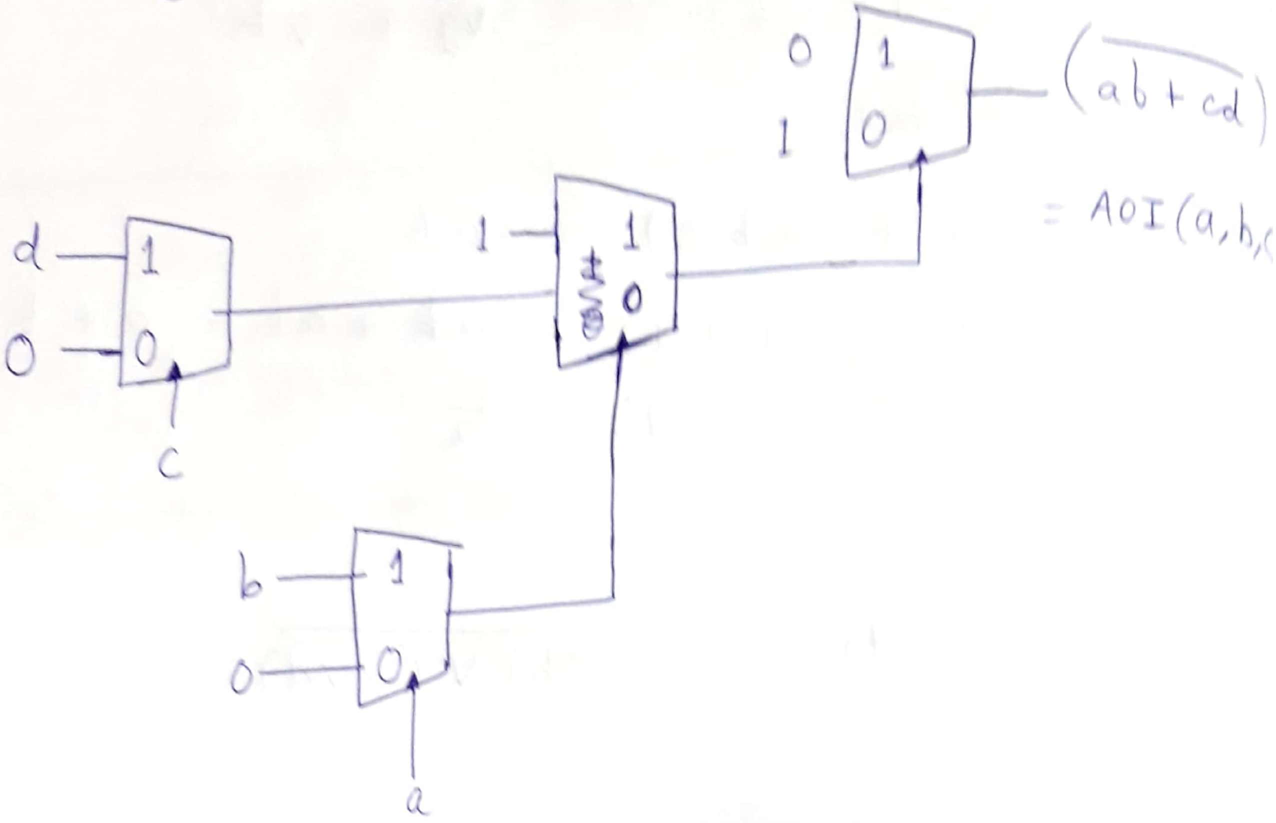
$$(a \wedge b) \vee (c \wedge d) = ITE(ITE(a, b, 0), 1, ITE(c, d, 0))$$

Then,

$$\overline{(a \wedge b) \vee (c \wedge d)} = ITE(ITE(ITE(a, b, 0), 1, ITE(c, d, 0)), 0, 1)$$

Logic diagram on next page

2:1 MUX based implementation:



NOTE: Q3 after Q4

Q4) Booth's multiplication offers a speed advantag in situations where:

☆ It handles both positive and negative multiplier uniformly.

☆ It achieves efficiency in the number of additions required when multiplier has few large blocks of 1's

☆ Booth's algorithm makes multiplication with small (in absolute terms) negative values significantly faster than shift-and-add multiplication, because the representation would have a large sequence of 1's. The number of operations performed would be less.

Q3) Let initial unsigned number be x. After step 1 (ie, complementing all bits of x), let the resultant be y. Now in process x → y, bits are flipped (1 → 0 and 0 → 1), which can be done by subtracting x from n-bit number containing all 1's, ie $2^n - 1$;

So,

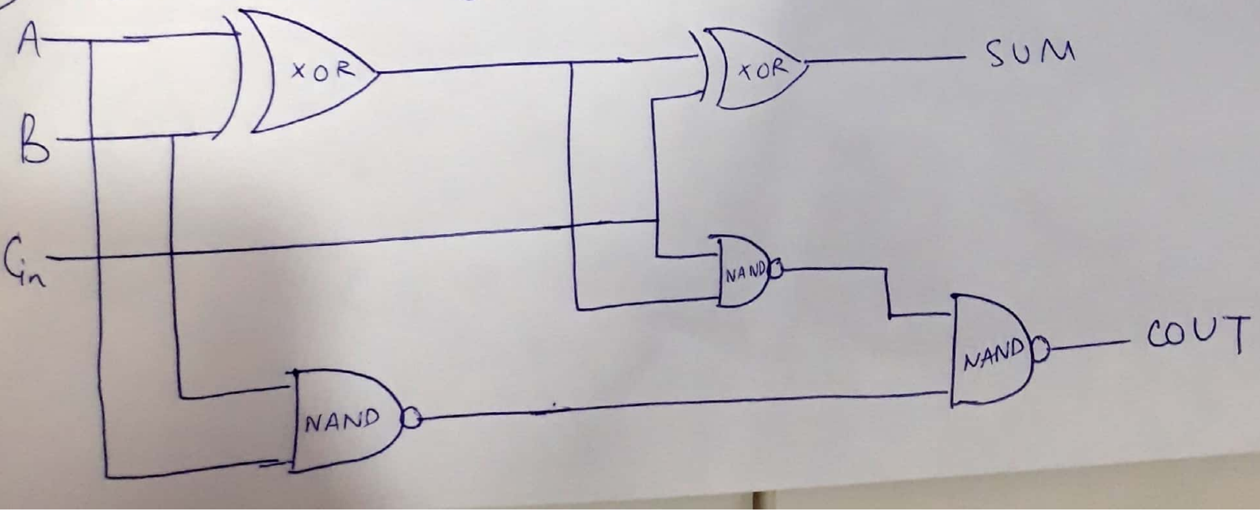
$$y = (2^n - 1) - x$$

Now our answer $= y + 1$

$$= \boxed{2^n - X}$$

Q5) After Q 6

Q6) Full adder using 2 half adders



Delay of XOR = $\cancel{\xi_{XOR}}$ D_{XOR}

Delay of NAND = D_{NAND}

for SUM, delay = $2D_{XOR}$

for COUT, delay = $D_{XOR} + 2D_{NAND}$

Thus, for n-bit RCA :

Max Delay is for final sum bit, value is :

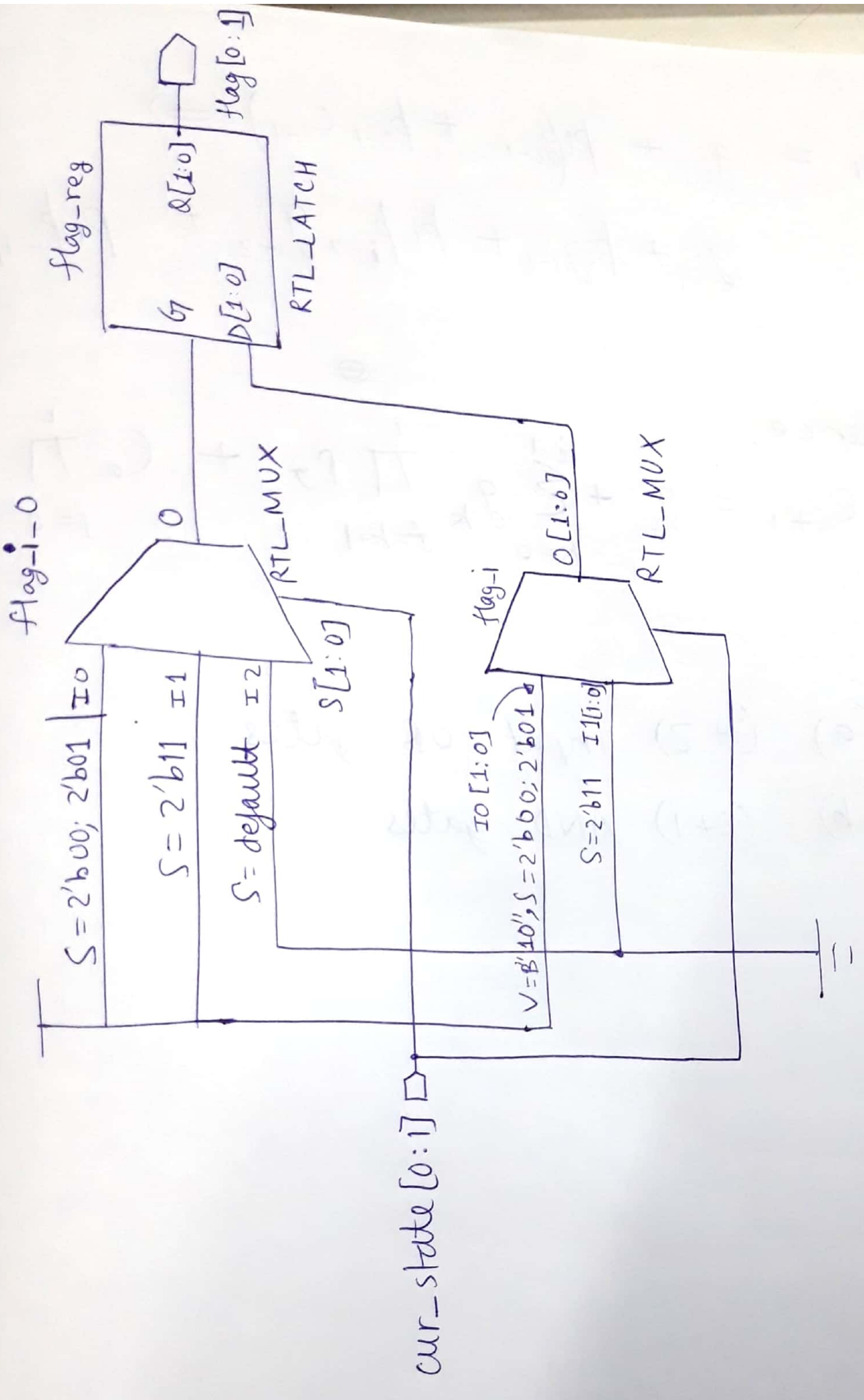
$$2D_{XOR} + (n-1)D_{XOR} + 2(n-1)D_{NAND}$$

∴ Worst case delay for RCA :

$$(n+1)D_{XOR} + 2(n-1)D_{NAND}$$

Q5) PTO

Q5)



Q7)

(a) $C_{i+1} = g_i + p_i C_i$

(b) $\overline{C_{i+1}} = p_i \overline{C_i} + k_i$

(c) P T O

(c) $c_{i+1} = g_i + p_i(g_{i-1} + p_{i-1} c_{i-1}) \Rightarrow$

$$g_i + p_i g_{i-1} + p_i p_{i-2} g_{i-2} + p_i p_{i-1} p_{i-2} g_{i-3} + \cdots$$

Hence,

$$c_{i+1} = g_i + \sum_{k=0}^{i-1} g_k \prod_{j=k+1}^{i} p_j + c_0 \prod_{j=0}^{i} p_j$$

(d) (a) $(i+2)$ input OR gates

(b) $(i+1)$ AND gates