

## COA LAB TEST -1

(Itish Agarwal, 18CS30021)  
21 Oct 2020

Q1)

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

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

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

$$\text{For } S = 0011,$$

$$\begin{aligned} Z &= \bar{X} Y + \bar{X} \bar{Y} \\ &= \bar{X} \end{aligned}$$

(c) (a) To realize XOR function,

$$S = \begin{matrix} 0 & 1 & 1 & 0 \\ S_3 & S_2 & S_1 & S_0 \end{matrix}$$

(b) To realize OR function,

$$S = \begin{matrix} 1 & 1 & 1 & 0 \\ S_3 & S_2 & S_1 & S_0 \end{matrix}$$

Q2.

PTO

Q2)

We have,

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

We see,

$$\text{ITE}(a, b, 0) = ab$$

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

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

Now,

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

$$\star a \wedge b = \text{ITE}(a, b, 0)$$

$$\star c \wedge d = \text{ITE}(c, d, 0)$$

Then,

~~$$\text{AOI}(a, b, c, d)$$~~

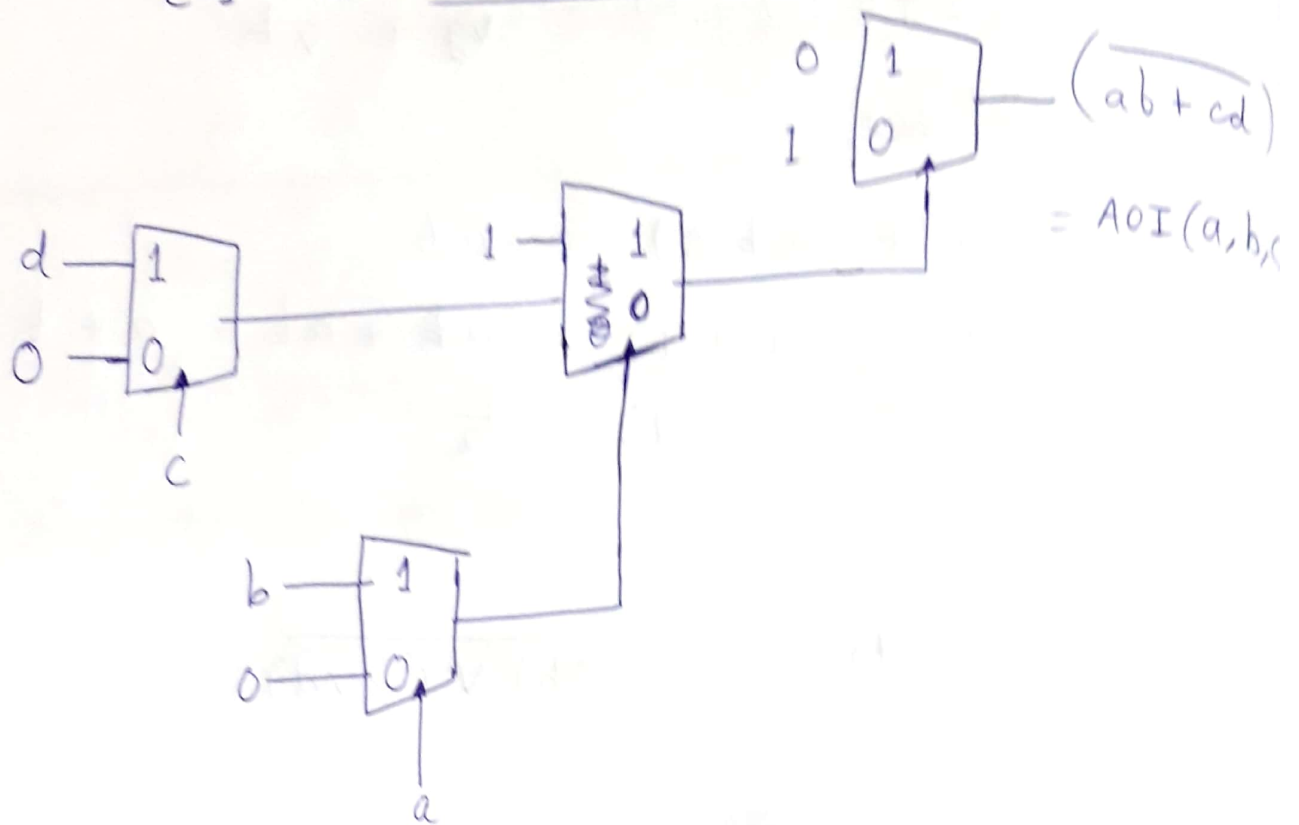
$$(a \wedge b) \vee (c \wedge d) = \text{ITE}(\text{ITE}(a, b, 0), 1, \text{ITE}(c, d, 0))$$

Then,

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

Logic diagram on next page

## 2:1 MUX based implementation:



NOTE: Q3 after Q4

Q4) Booth's multiplication offers a speed advantage 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 <sup>unsigned</sup> number be  $x$ . After step 1 (ie, complementing all bits of  $x$ ), let the resultant be  $y$ . Now in process  $x \rightarrow y$ , bits are flipped ( $1 \rightarrow 0$  and  $0 \rightarrow 1$ ), which can be done by subtracting  $x$  from  $n$ -bit number containing all 1's, ie,  $2^n - 1$ ;

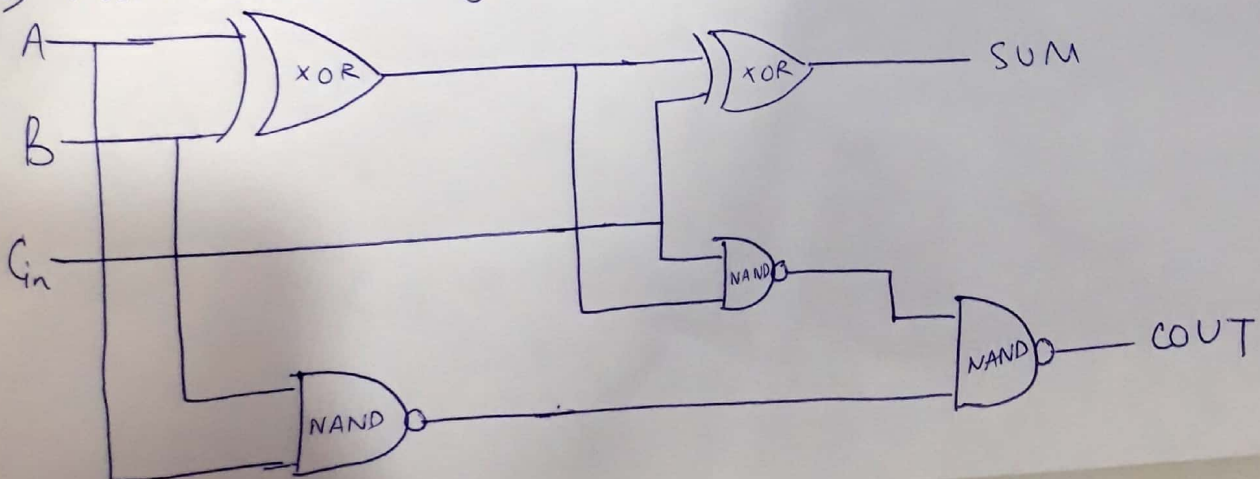
$$\text{So, } y = (2^n - 1) - x$$

$$\text{Now our answer} = y + 1$$

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

Q5) After Q6

Q6) Full adder using 2 half adders





$$\text{Delay of XOR} = D_{\text{XOR}}$$

$$\text{Delay of NAND} = D_{\text{NAND}}$$

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

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

Thus, for n-bit RCA:

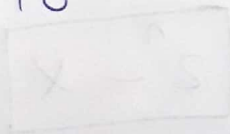
Max Delay is for final sum bit, value is:

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

$\therefore$  Worst case delay for RCA:

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

Q5) PTO



The diagram illustrates a digital circuit for a flag register. It features two multiplexers (MUX) and one latch.

- Top MUX:** Selects between three inputs based on the `flag_i-0` signal. The inputs are:
  - `S = 2'b00; I0`
  - `S = 2'b11; I1`
  - `S = default; I2`
 Its output is `S[1:0]`.
- Bottom MUX:** Selects between two inputs based on the `flag_i` signal. The inputs are:
  - `V = B'10''; S = 2'b00; I0[1:0]`
  - `S = 2'b11; I1[1:0]`
 Its output is `O[1:0]`.
- Latch:** An RTL\_LATCH with input `D[1:0]` and output `Q[1:0]`. The output `Q[1:0]` is connected to the `I2` input of the top MUX and also serves as the `flag_o:1` output of the circuit.
- Control Signals:**
  - `cur_state [0:1]`: Connected to the clock inputs of both MUXes and the latch.
  - `flag_i-0`: Select signal for the top MUX.
  - `flag_i`: Select signal for the bottom MUX.

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

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

(1) P.T.O.

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

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

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