Digital System Design - Assignment 4

Name: Megh Manoj Bhalerao

Roll No: 16EE234

1) a) 8x8 array multiplier:

For nxn array multiplier:

n^2 AND gate, $n(n-2)$ Full adders, n - half adders.

∴ 64 AND gates

48 Full adders

8 Half adders

b) 8 x 8 array multiplier

t_g (AND gate delay) = 1ns

t_{ad} (HA / FA delay) = 2ns

We know that, the longest path from input to output goes through. n-adders in the top row, n-1 adders in the bottom row and n-3 adders in the middle row.

Hence, worst case multiply time is:

$(3n-4)\,t_{ad} + t_g = (3\times 8 - 4)\,2ns + 1ns$

$= \underline{\underline{41ns}}$

Worst case can be improved to $2n t_{ad} + t_g$ by forwarding carry from each adder to diagonally lower adder.

⊗ Hence, ~~the time becomes~~

$2 \times 8 \times 2\,ns + 1ns = 33ns$

c) Assuming worst case time (time taken by array multiplier is 41 ns).

In an add and shift multiplier : the worst case number of changes in state is $2n+2$, which takes $2n+1$ clock cycles, assuming 'n' is the # of bits multiplier.

∴ For 8 bit add - shift multiplier :-

∴ $2 \times 8 + 1 = 17$ clock cycles

17 clock cycles → 41 ns

1 clock → x → $\frac{41ns}{17}$

T = (time period of 1 clock cycl) : $\frac{41}{17}$ ns = 2.411ns.

f (of clock cycle) = 0.4147×10^9 Hz (f of clock)

2) a) 16x16 array multiplier

AND gates = n^2 = 256 AND gates

# of Adders (HA & FA) = $n(n-2) + n = 16 \times 14 + 16$

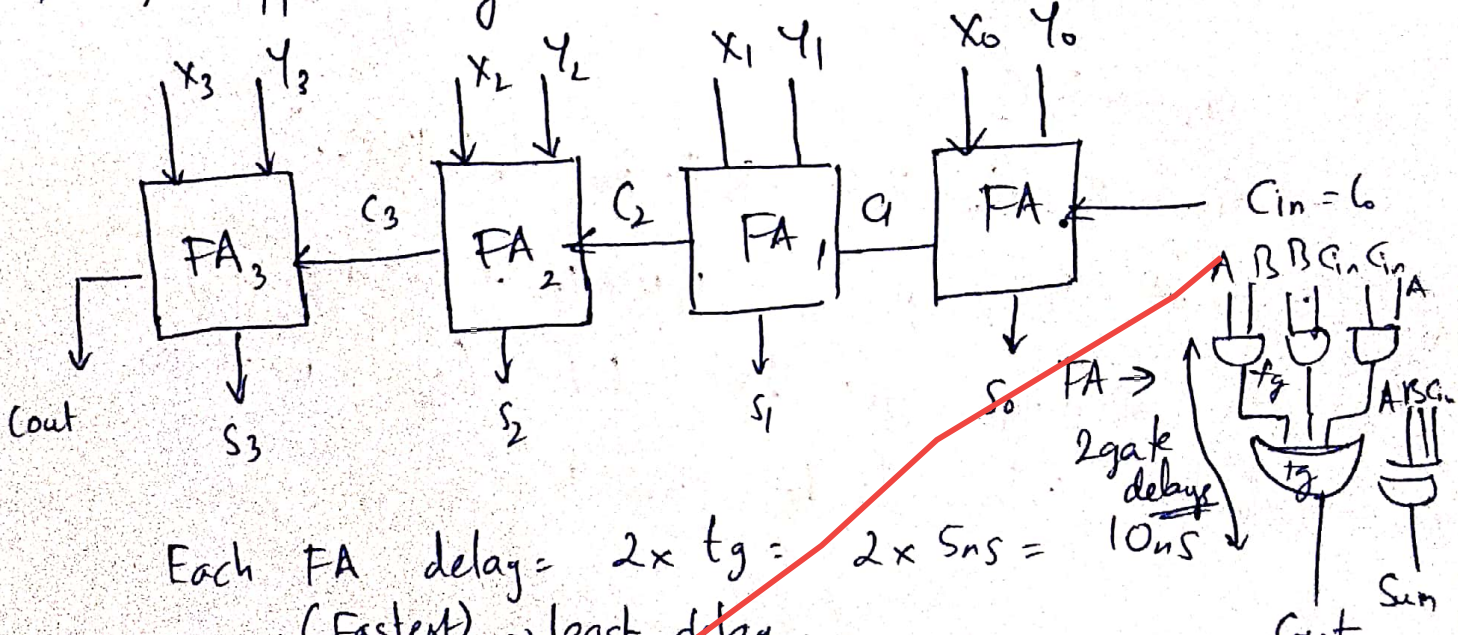
FA HA = 240 adders

b) Worst case/Longest delay in standard array multiplier representation :

$(3n-4)\,t_{ad} + t_g$ → Without improvement

$2n\,t_{ad} + t_g$ → With improvement → forward carry from each adder to diagonally lower adder

3) a) Ripple carry adder



Each FA delay = $2 \times t_g = 2 \times 5ns$ = 10nS (Fastest) → least delay.

∴ 4 adder delay = 4×10 = 40nS

3) b) The Carry look-ahead adder is the fastest bit adder. The CLA predicts the carry before it is calculated.

$$C_i = G_i + P_i \cdot C_{i-1}$$

$$G_i = A_i B_i, \quad P_i = A_i \oplus B_i$$

* From this recursive equation, all the carrys can be predicted in advance

2nd last carry in a 4-bit adder

~~$C_3 = (G_3 + P_3 (G_2 + P$~~

$C_2 = G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_{-1}) =$

$G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1}$

Directly ~~last~~ second-last bit

$C_3 = G_3 + P_3 C_2$

$= G_3 + P_3 (G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_{-1})$

$C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_{-1}$

↳ Can directly get last bit. (No need to calculate intermediate circuit.

- The general form for the delay of CLA with L- levels is $2(2L-1)+4 = 4L+2$ gate delays

Here # of levels for 4 bit CLA (fastest) = 1

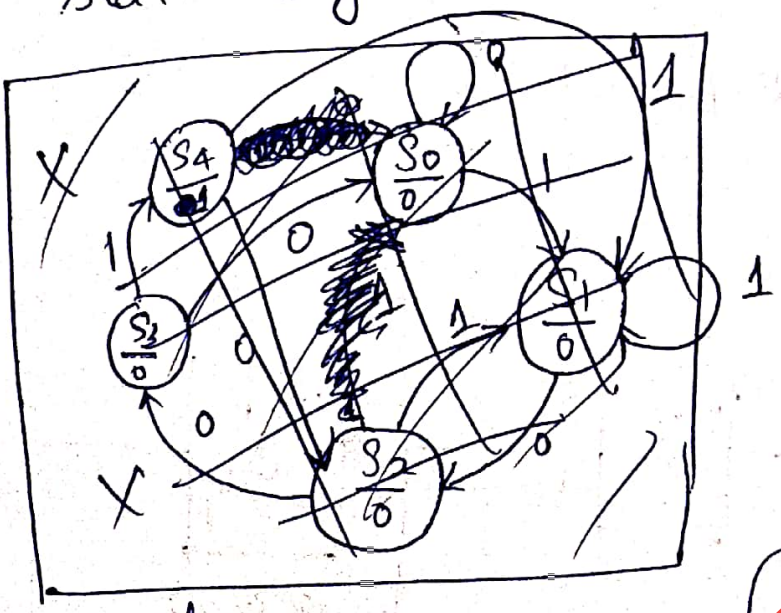
$\therefore$ $4\times1+2$ gate delay

$\Rightarrow$ $6\ tg = 6\times5ns =$ (30 ns) delay in CLA

4) 1001 detector :-

# of sq. boxes in ASM = # of states of the system.

State diagram for 1001 detector

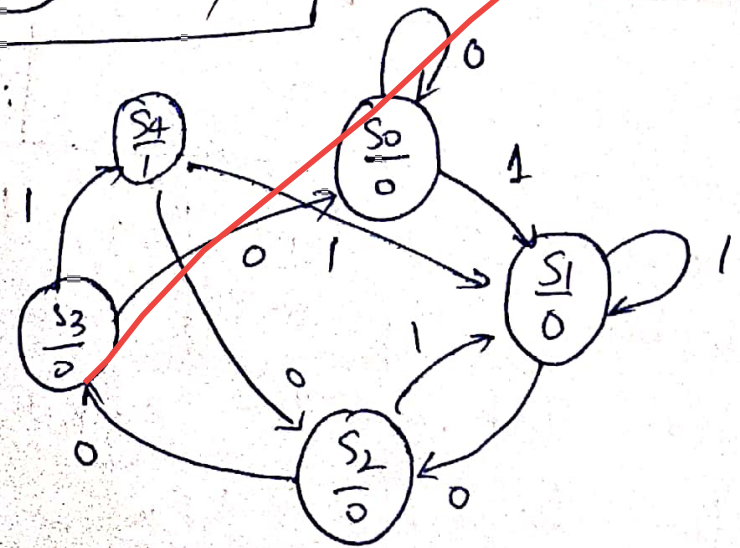


1001 - detector

1000

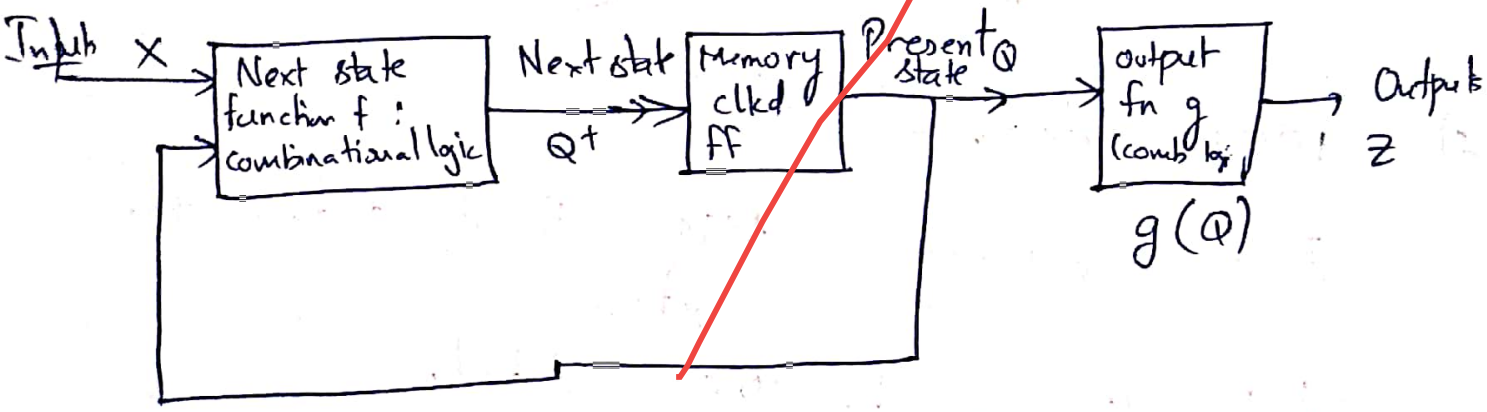
$S_0 \rightarrow$ Reset state

State diagram:

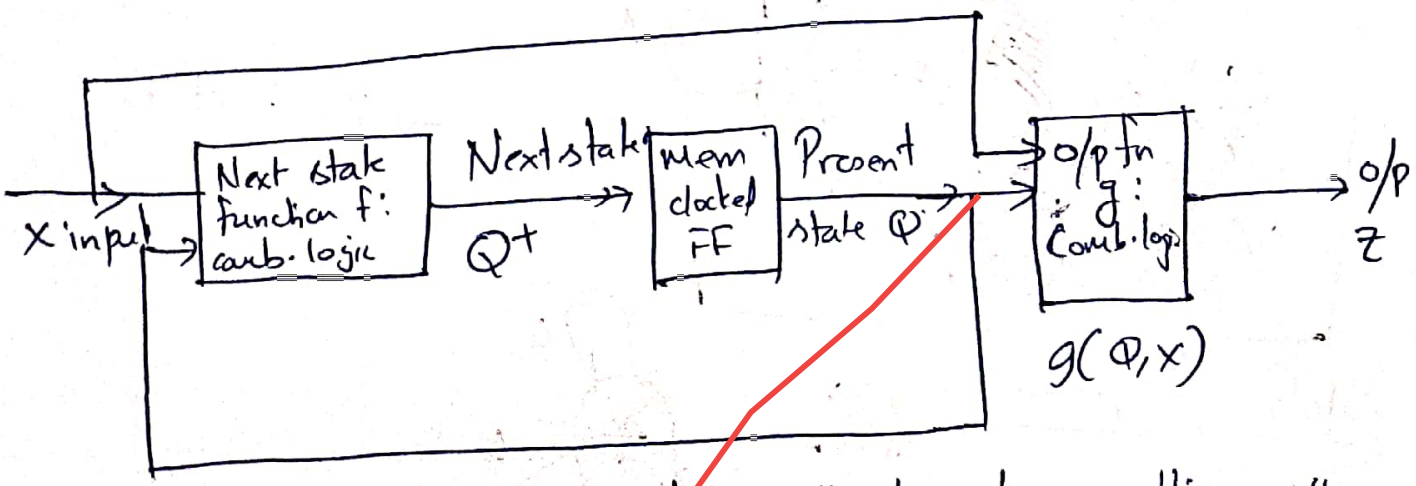


→ Moore stak diagram has 5 -states
→ Hence ASM representation will have 5 stak boxes.

5) Moore Machine Representation:



Mealy Machine Representation:



Hence, it can be seen here that by setting the g(Q,X) function in the mealy machine such that 'X'-has no effect on the output, the mealy representation becomes a moore representation i.e g(Q)

⊗ However, the conversion $g(Q) \rightarrow g(Q, X)$ is not possible since we need to add another dependency on X, but there is no such provision made in the machine representation for a moore machine.

Hence:

① Mealy can be represented as Moore.

② But Moore can't be represented as mealy.

6) The ASM can be converted into the state table for easier analysis.

(2 states)

X = don't care

→ State table from the ASM diagram:

Present state	Next state (AB form) I/P 00	01	10	11	Output (z)
S_0	S_0	S_0	S_1	S_1	0
S_1	S_1	S_0	S_1	S_0	1

→ Encoding the states as numbers:

PS (Q)	NS (I/P = AB form) 00	01	10	11	Output z
0	0	0	1	1	0
1	1	0	1	0	1

→ Truth table for the corresponding state table

Q	A	B	Q^+	Z
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	1	1	1	0
1	0	0	1	1
1	0	1	0	1
1	1	0	1	1
1	1	1	0	1

$Z = Q$

From TT

$$Q^+ = \bar{Q}A + Q\bar{B}$$

This is the characteristic eqn (excitation eq) of A JK Flip flop where

$J = A$

$K = \bar{B}$

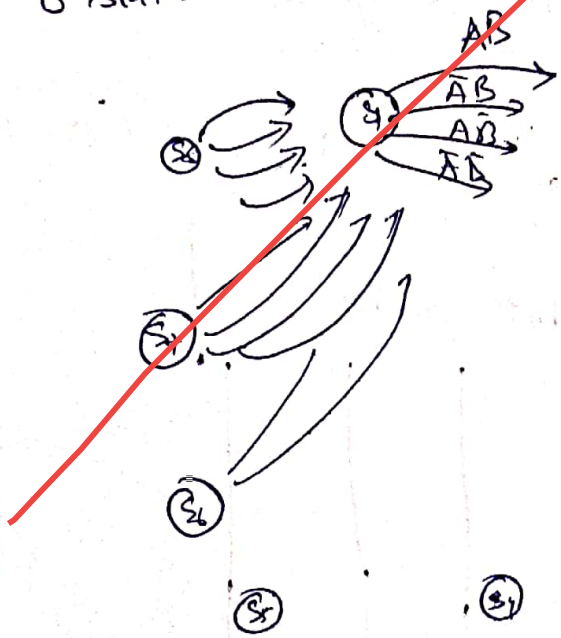
7) To determine the MAX number of transition arrows leaving and entering the state.

- ⓧ 3FF
- ⓧ 2 INPUTS → AB
- ⓧ 5 outputs → Doesn't matter since state dependent

} Moore machine

3FF → 2^3 = 8 states max.

I/P = AB



Max # of arrows leaving a state = 4 (2^2 → 2 I/P's)

Max # of arrows entering a state ⇒ If all states point to one of the state (say S_1) ⇒

# of incoming arrows on S_1 are :

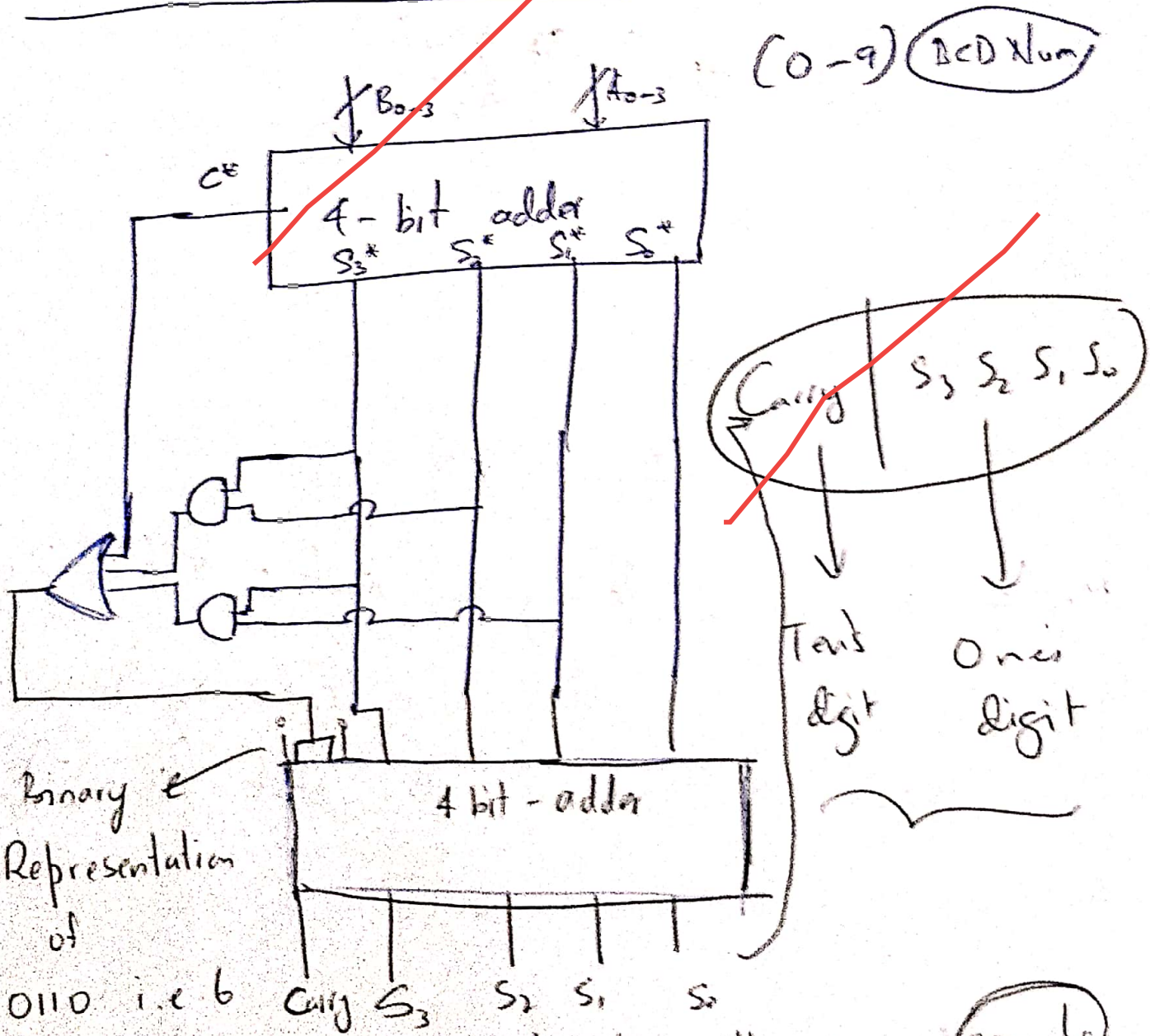
$(2^3 - 1) \times 2^2 = 28$ inbound arrow at most for any given state

5) N states → # of flip-flops or

$$\left[\log_2(N)\right] + 1 = \text{No. of FF}$$

$[\cdot]$ denotes the step-function or Greatest Integer function

9) BCD Adder circuit diagram:



Hence, the statement is FALSE.

Since if result is 1001, 0110 must be added

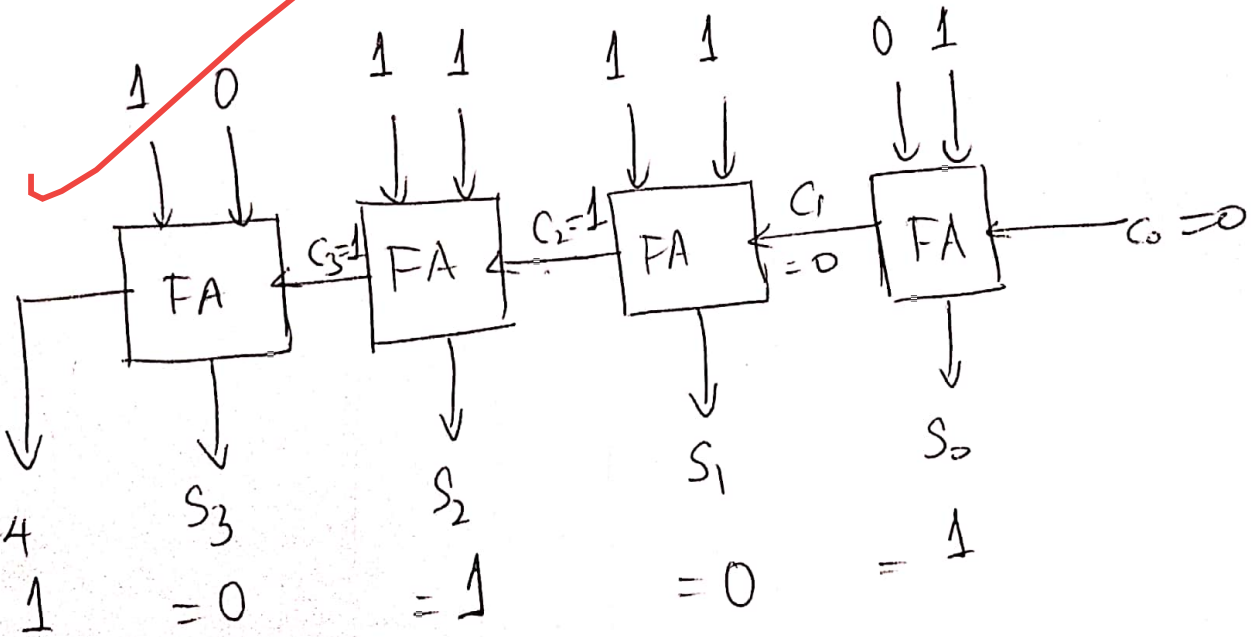
Greater than (Not less-than)

10) 4-bit adder:

14 → in binary form - 1110 → A

7 → in binary 4bit form - 0111 → B

∴ The 4 bit adder will look like:



∴ $C_3C_2C_1 = 110$

$C_4S_3S_2S_1S_0 : 10101 \rightarrow (21)$ (Correct Sum)