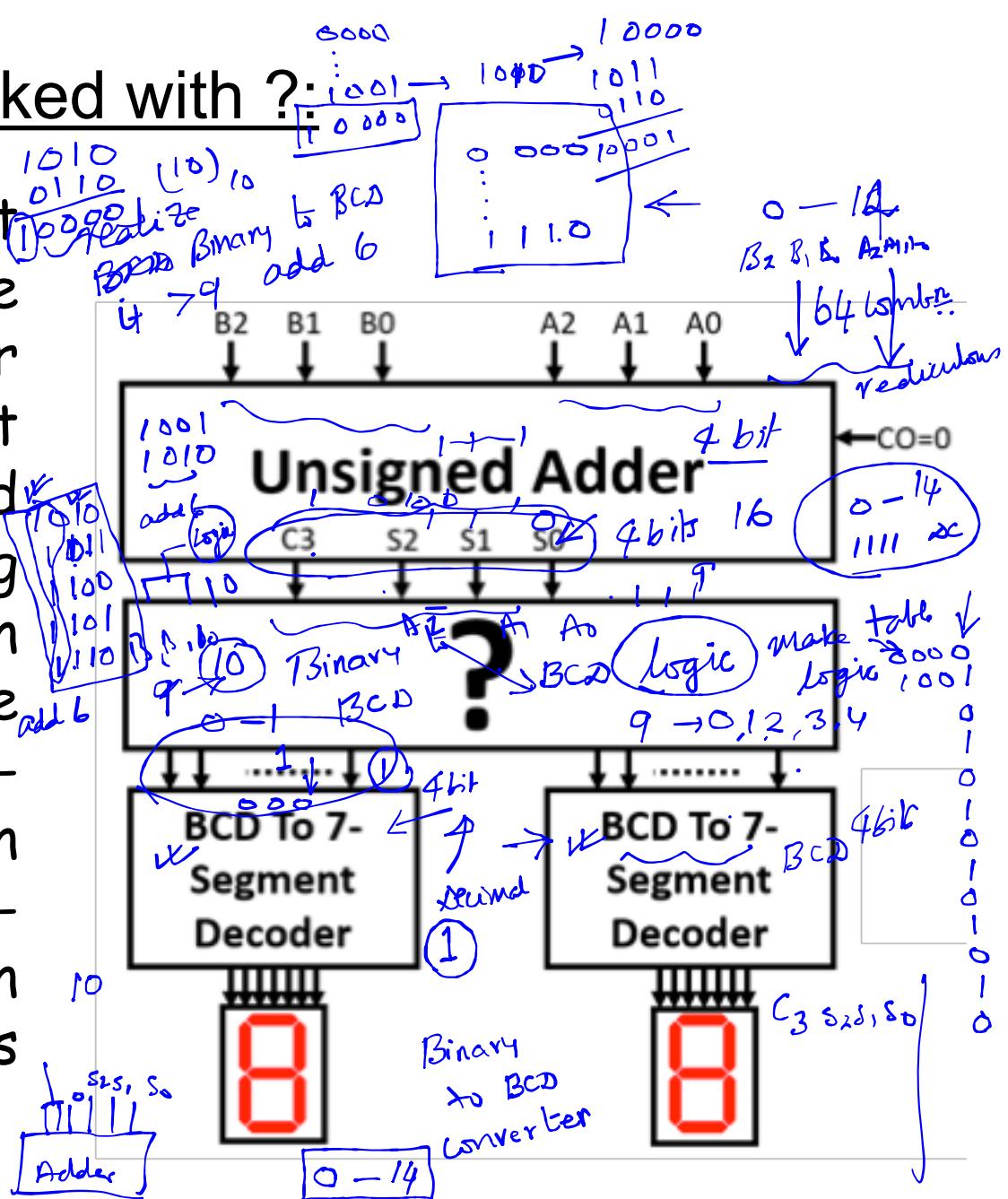


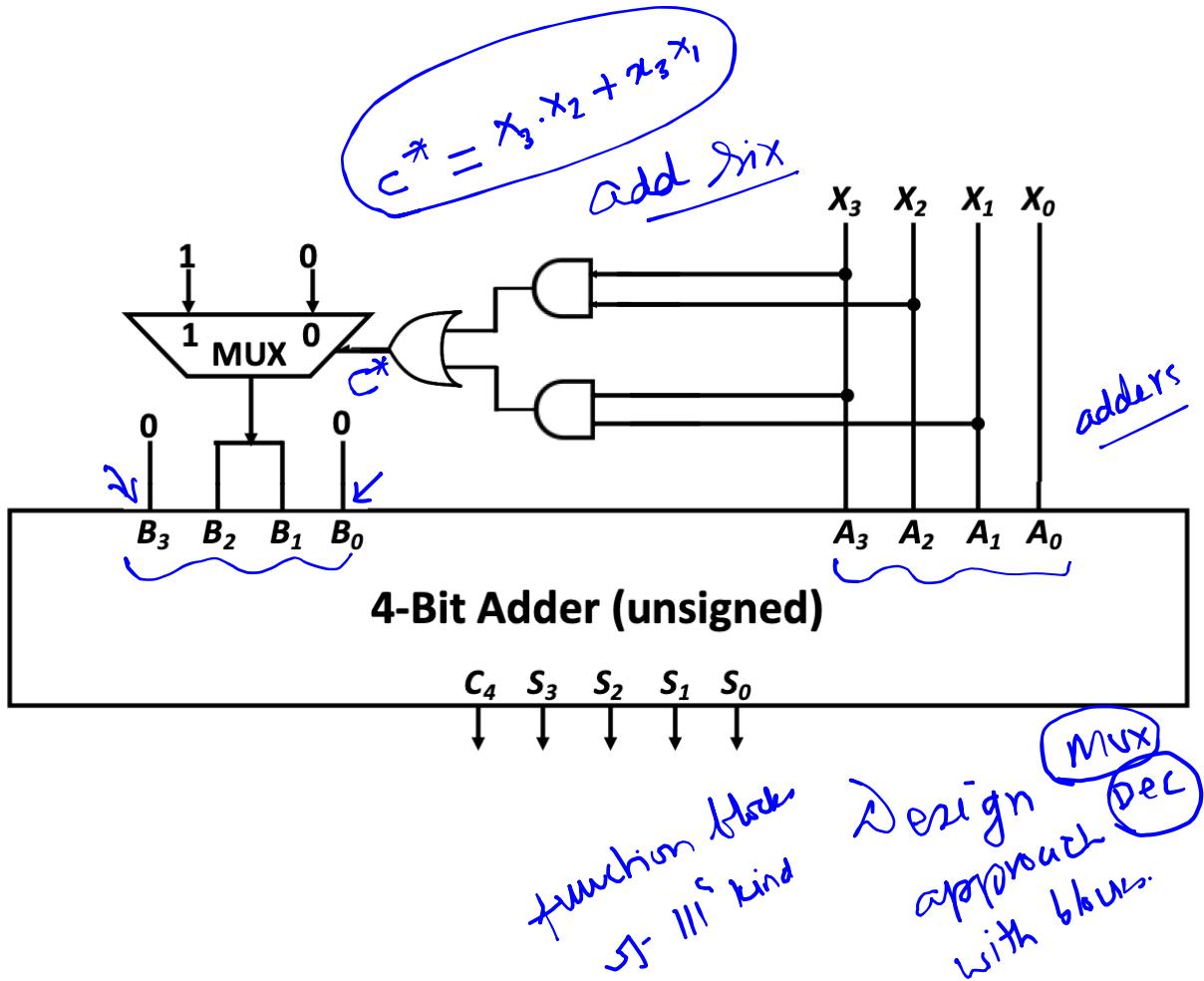


# HW --- Identify the Functional Block marked with ?:

Consider the circuit shown where the output of a 3-bit adder needs to be displayed on the two digits of the 7-segment display. For example, if the adder output is 9, then circuit should display 09 where 0 should be displayed on left display and 9 on right display. Existing BCD-to-7 segment decoder works only when input number is between 0 and 9 and we have to use it since it has been hardwired to 7-segment display board. You need to design combinational circuit between Adder and 7-segment decoder (shown as question mark in the Figure) so that the adder output is correctly displayed on the 7-segment display.



## H.W: Identify the functionality:



$A_3$	$A_2$	$A_1$	$A_0$
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

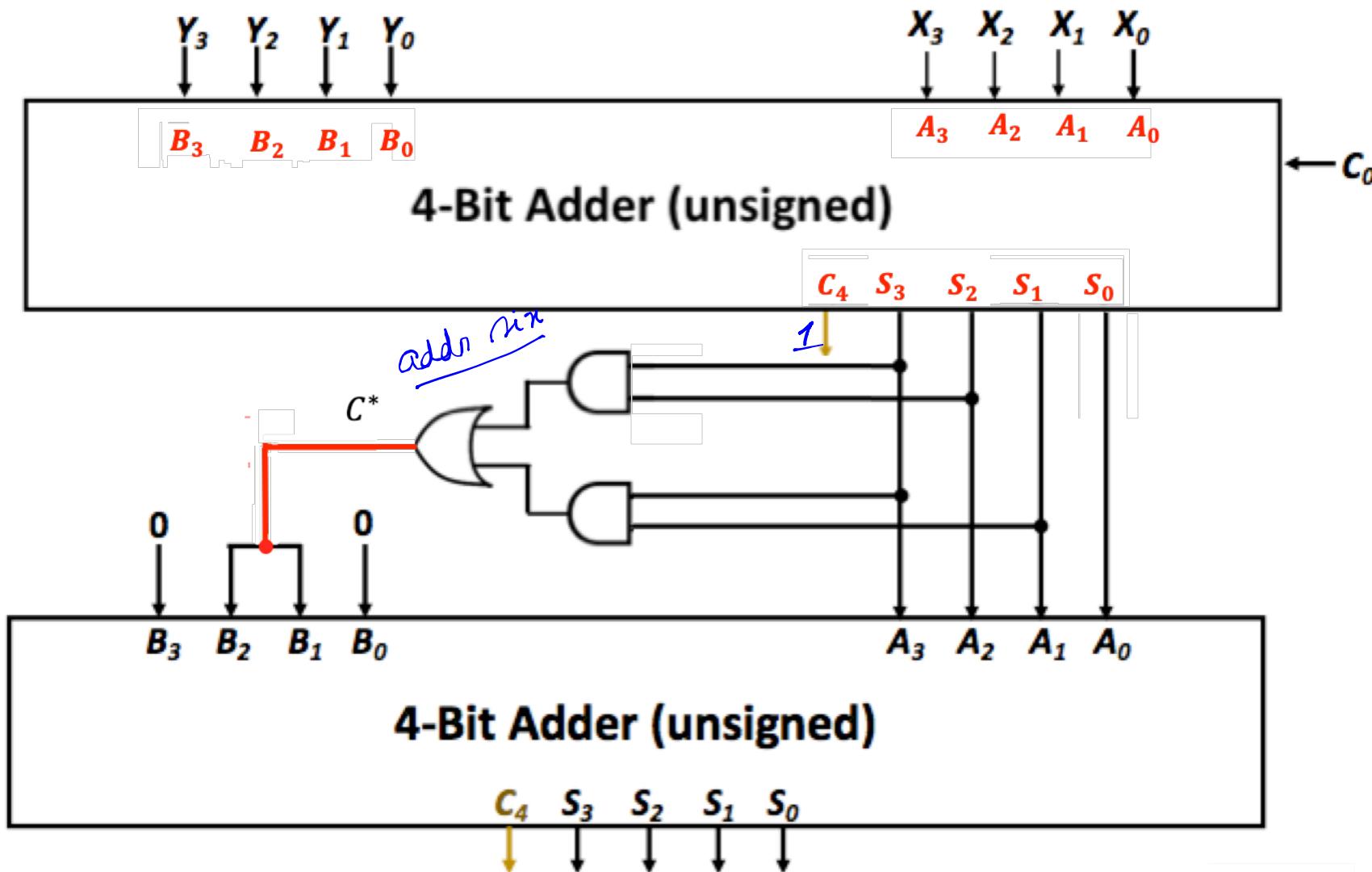
$B_3$	$B_2$	$B_1$	$B_0$
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0
0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

$C_4$	$S_3$	$S_2$	$S_1$	$S_0$
0	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	0	1	1
0	0	1	0	0
0	0	1	0	1
0	0	1	1	0
0	0	1	1	1
0	1	0	0	0
0	1	0	0	1
1	0	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	0	1	1
1	0	1	0	0
1	0	1	0	1

## BCD Adder:

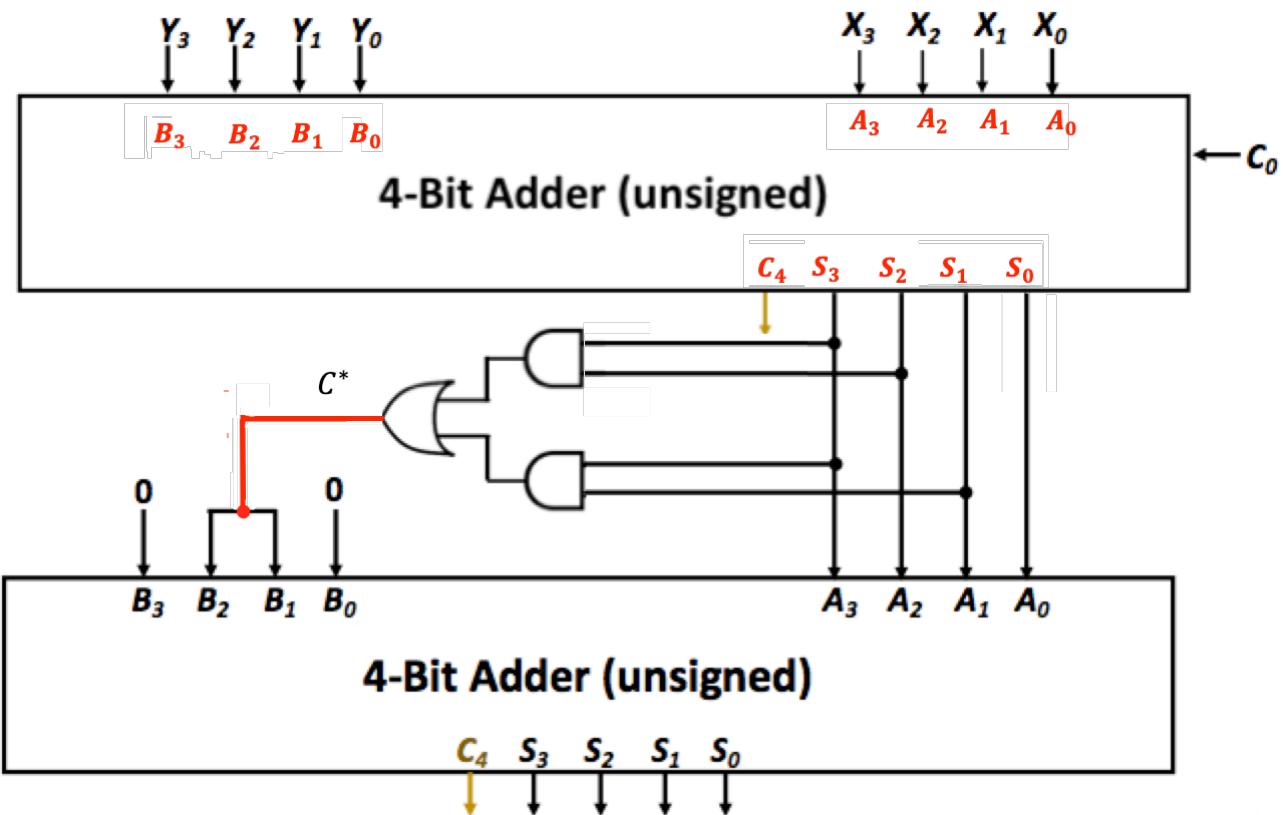
- Let us design a **binary circuit** which performs the addition of two single digit BCD numbers plus carry and gives the output in correct BCD form

## 1-Digit BCD Adder:

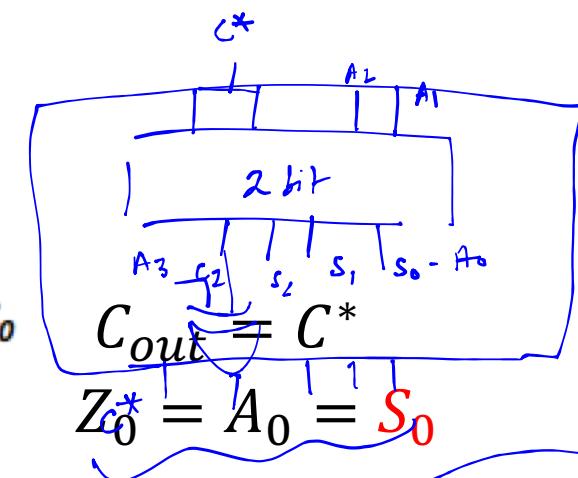
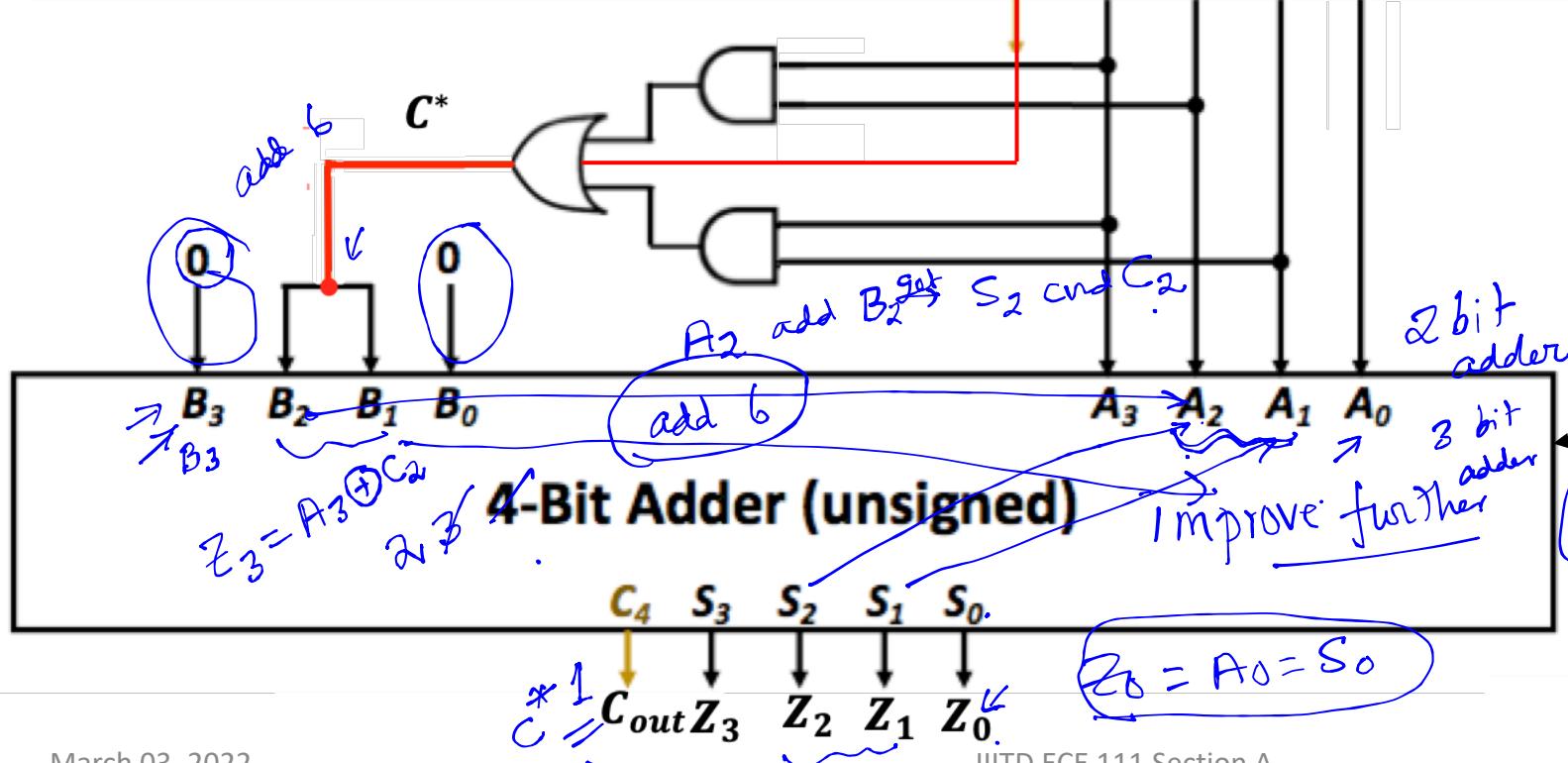
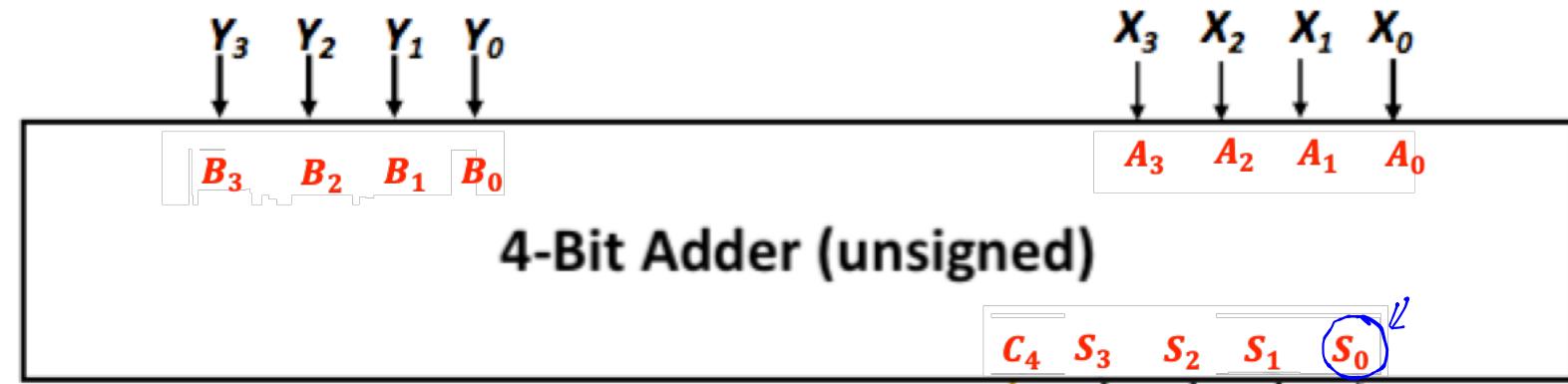


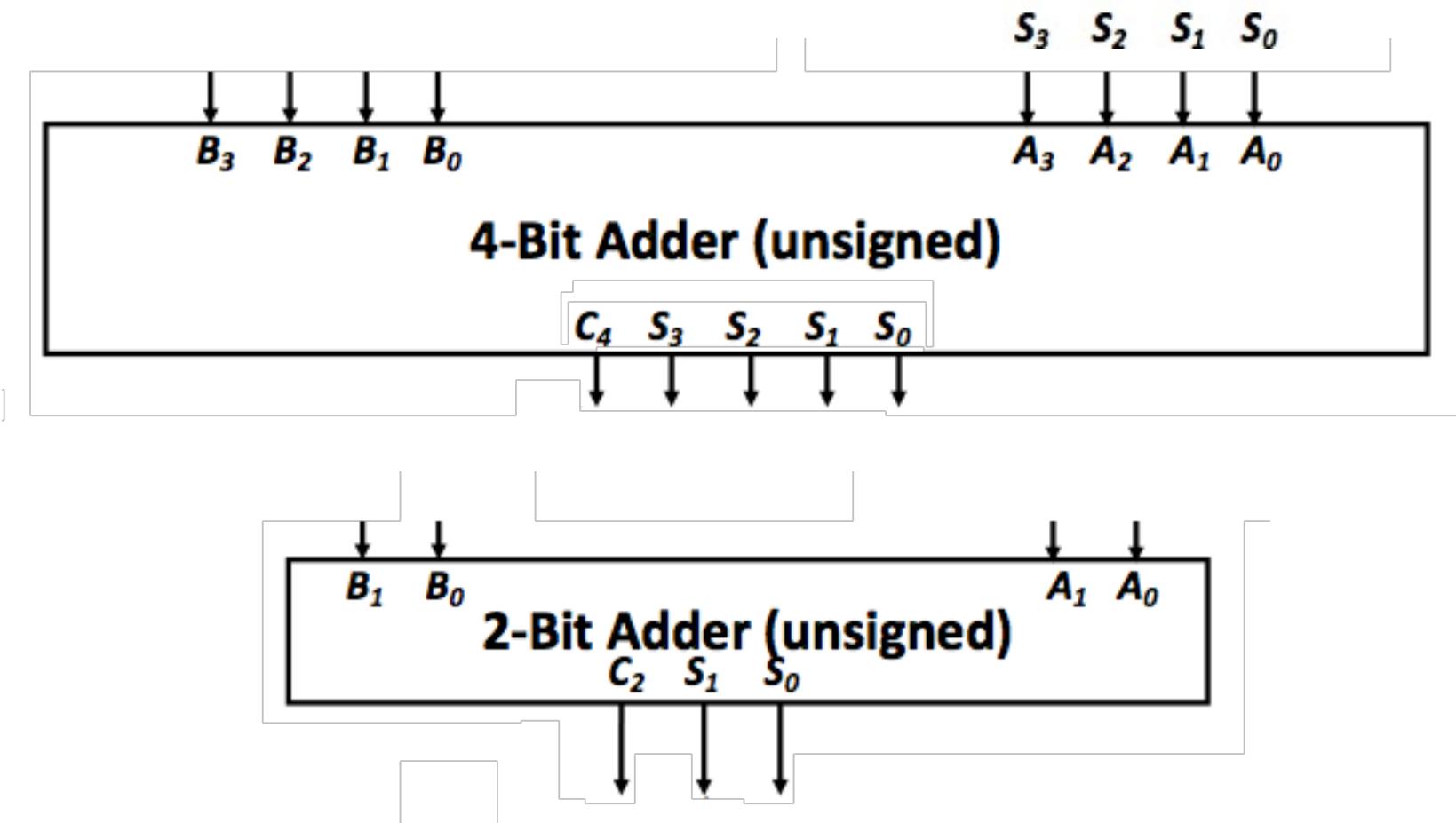
DE	$C_4$	$S_3$	$S_2$	$S_1$	$S_0$	$C^*$	$C_4$	$S_3$	$S_2$	$S_1$	$S_0$
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	1
2	0	0	0	1	0	0	0	0	0	1	0
3	0	0	0	1	1	0	0	0	0	1	1
4	0	0	1	0	0	0	0	0	1	0	0
5	0	0	1	0	1	0	0	0	1	0	1
6	0	0	1	1	0	0	0	0	1	1	0
7	0	0	1	1	1	0	0	0	1	1	1
8	0	1	0	0	0	0	0	1	0	0	0
9	0	1	0	0	1	0	0	1	0	0	1
10	0	1	0	1	0	1	1	0	0	0	0
11	0	1	0	1	1	1	1	0	0	0	1
12	0	1	1	0	0	1	1	0	0	1	0
13	0	1	1	0	1	1	1	0	0	1	1
14	0	1	1	1	0	1	1	0	1	0	0
15	0	1	1	1	1	1	1	0	1	0	1

DE	$C_4$	$S_3$	$S_2$	$S_1$	$S_0$	$C^*$	$C_4$	$S_3$	$S_2$	$S_1$	$S_0$
16	1	0	0	0	0	0	1	1	0	1	1
17	1	0	0	0	1	1	1	1	0	1	1
18	1	0	0	1	0	1	1	1	1	0	0



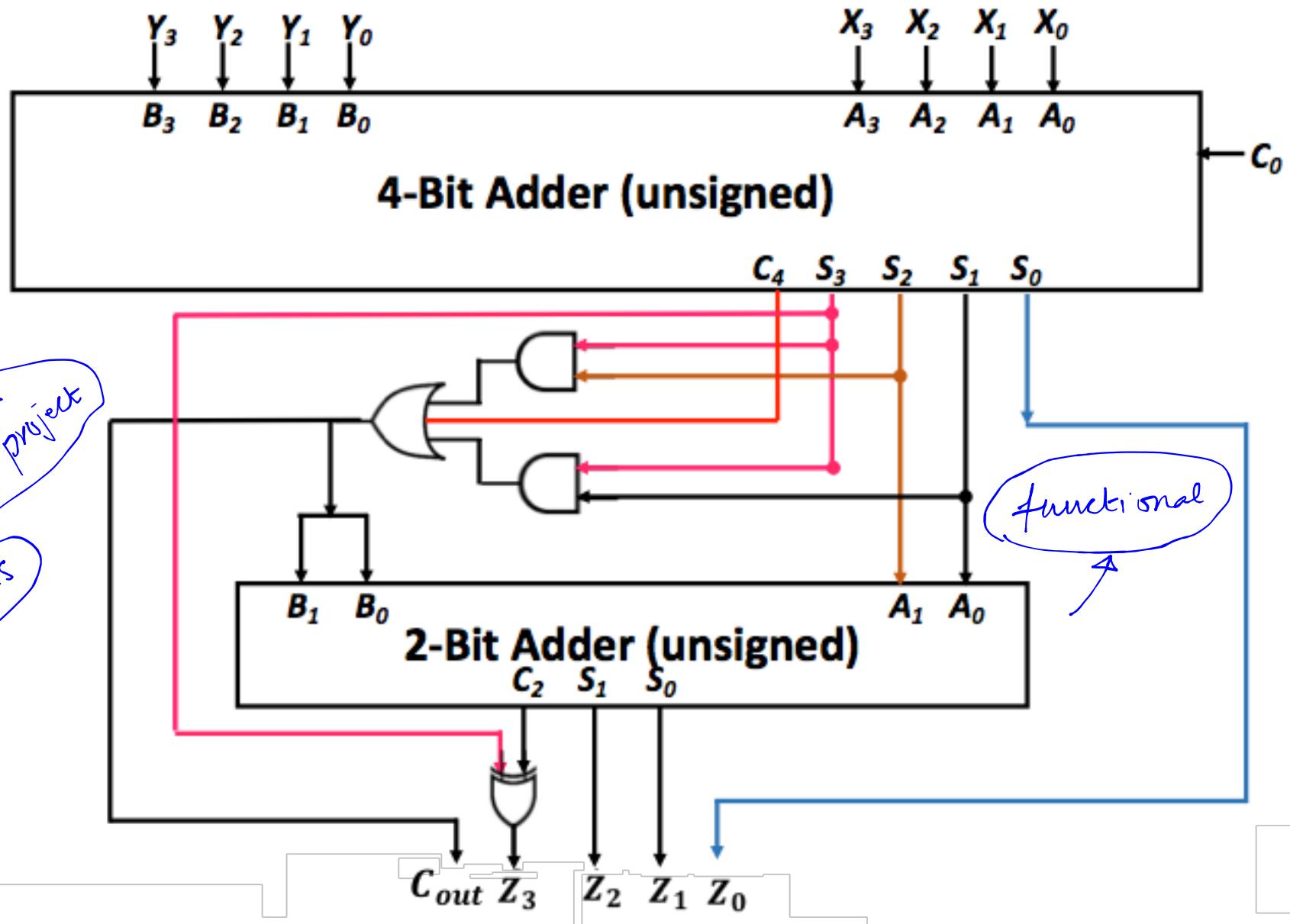
# 1-Digit BCD Adder:





## 1 Digit BCD Adder:

as difficult as you can handle.  
See whether you would like to do a small project  
if you want to do a project  
2 hours  
acceptable



**HW:**

Using a 1-Digit BCD adder, design a 3-Digit BCD adder.

# Multiplication:

## Unsigned numbers

$$\begin{array}{r} 14 \\ \times 17 \\ \hline 98 \\ 14 \\ \hline 238 \end{array}$$

Partial Product

Final Product

$$\begin{array}{r} 01110 \\ \times 10001 \\ \hline 0000001110 \\ 000000000 \\ 000000000 \\ 000000000 \\ 001110 \\ \hline 0011101110 \end{array}$$

$$\begin{array}{r} 01110 \\ \times 10001 \\ \hline 0000001110 \\ 000000000 \\ \hline 0000001110 \\ 000000000 \\ \hline 0000001110 \\ 000000000 \\ \hline 0000001110 \\ 001110 \\ \hline 0011101110 \end{array}$$

## Multiplication of signed 2's compliment numbers: (one positive and one negative number)

$$\begin{array}{r}
 +14 \\
 \times -15 \\
 \hline
 70 \\
 14 \\
 \hline
 -210
 \end{array}$$

Partial product to be subtracted since multiplier is negative.

$$\begin{array}{r}
 01110 \\
 \times 10001 \\
 \hline
 000001110 \\
 000000000 \\
 00000000 \\
 0000000 \\
 \hline
 001110
 \end{array}$$

2's kompl.  
 representation

~~1's complement~~  
~~of the partial product to be subtracted.~~

$$\begin{array}{r} 01110 \\ \times 10001 \\ \hline 0000001110 \\ 000000000 \\ \hline 0000001110 \\ 000000000 \\ \hline 0000001110 \\ 000000000 \\ \hline 0000001110 \end{array}$$

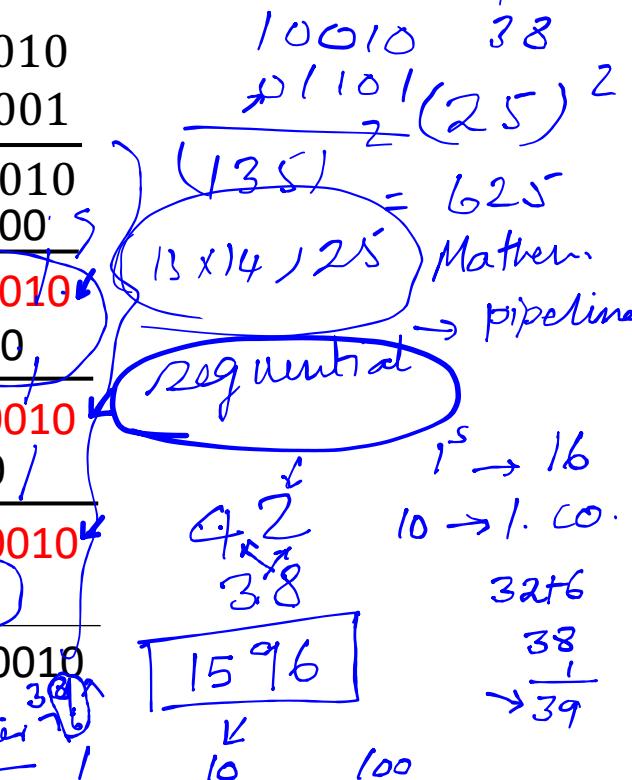
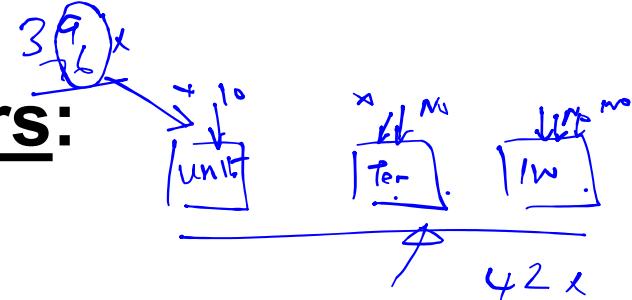
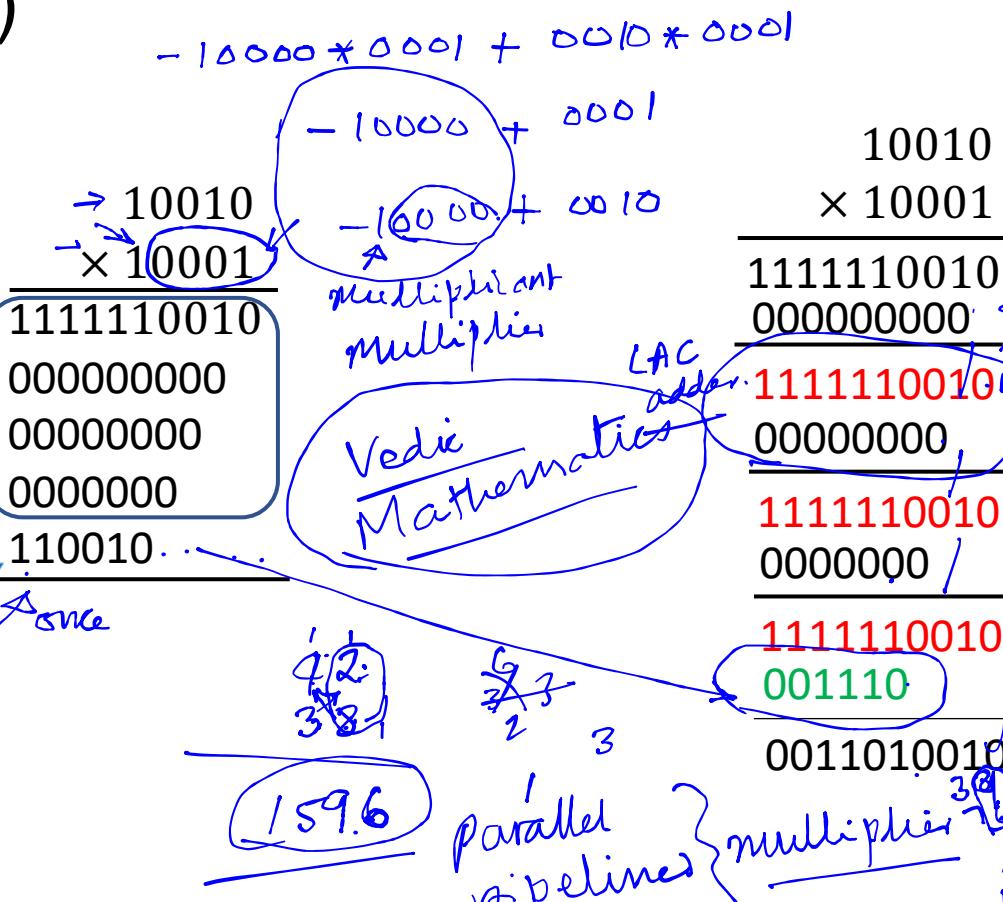
# Multiplication of signed 2's compliment numbers:

(both negative number)

$$\begin{array}{r} -14 \\ \times -15 \\ \hline 70 \\ 14 \\ \hline +210 \end{array}$$

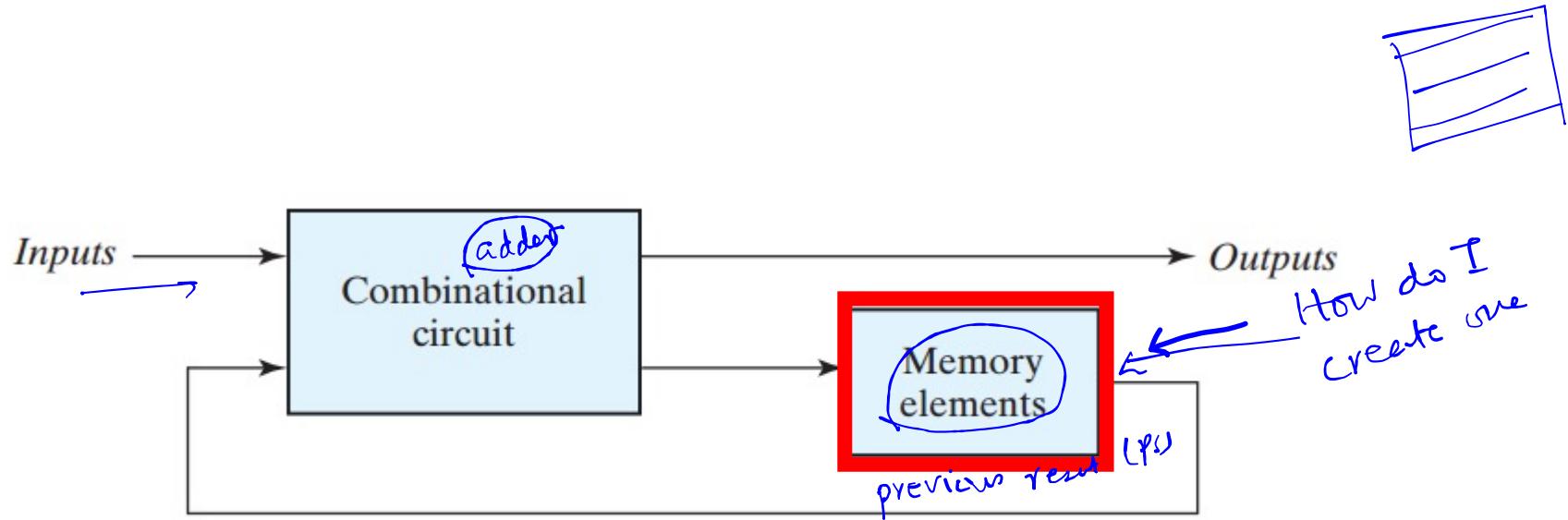
Negative partial product

Partial product to be subtracted since multiplier is negative.

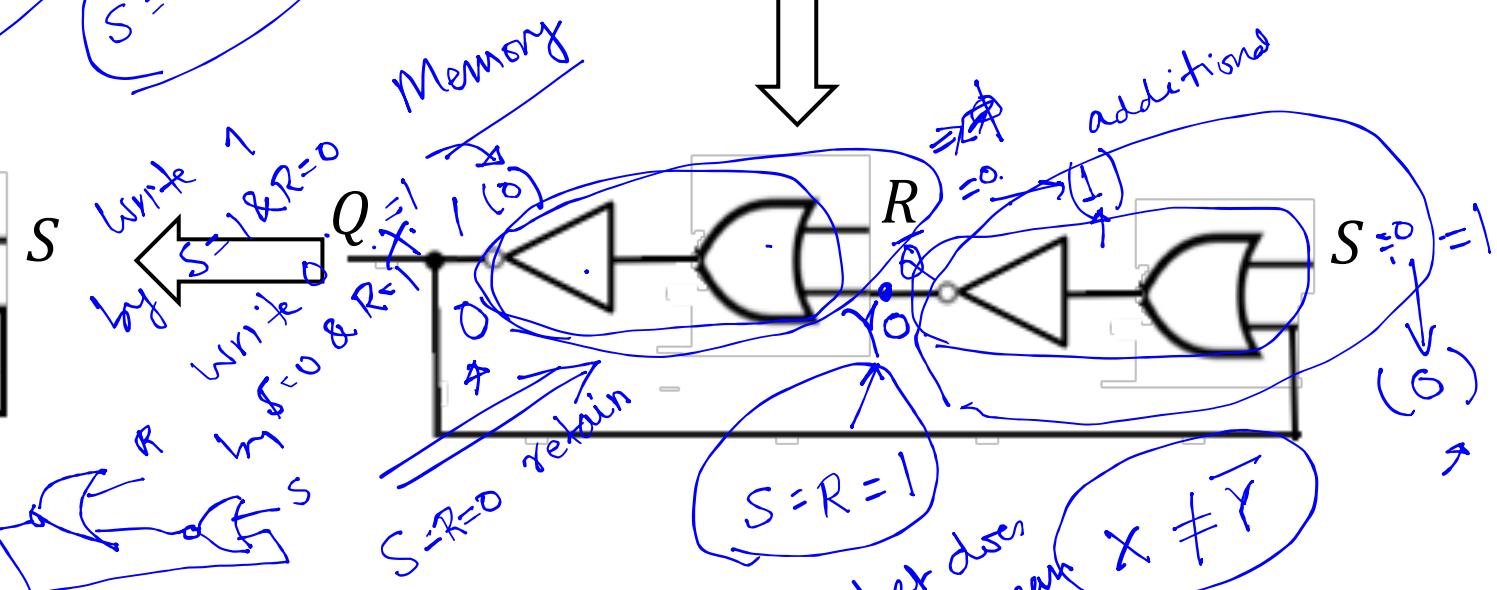
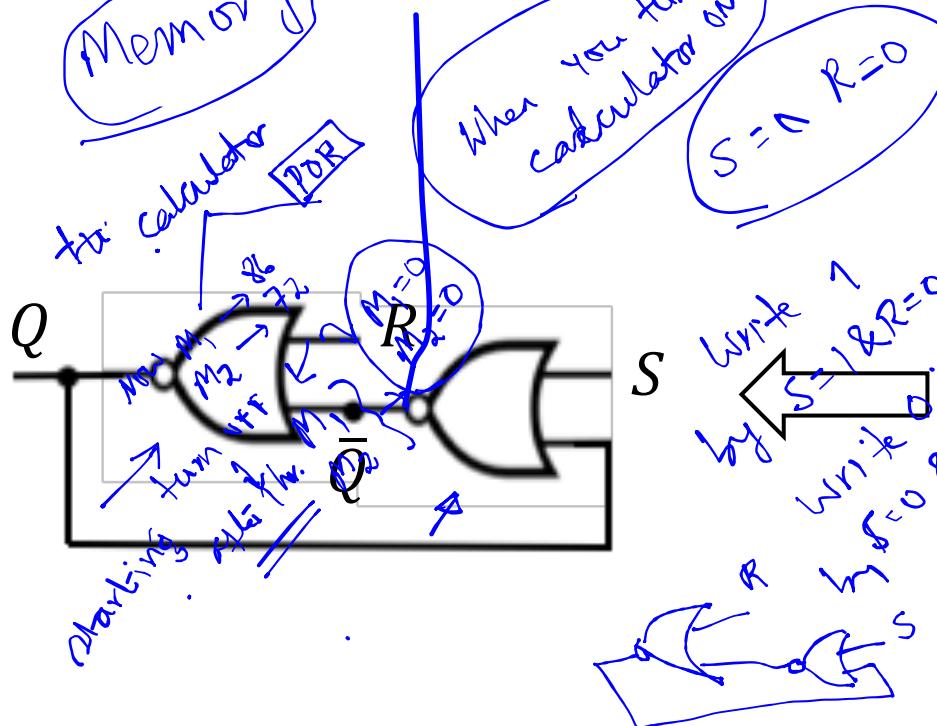
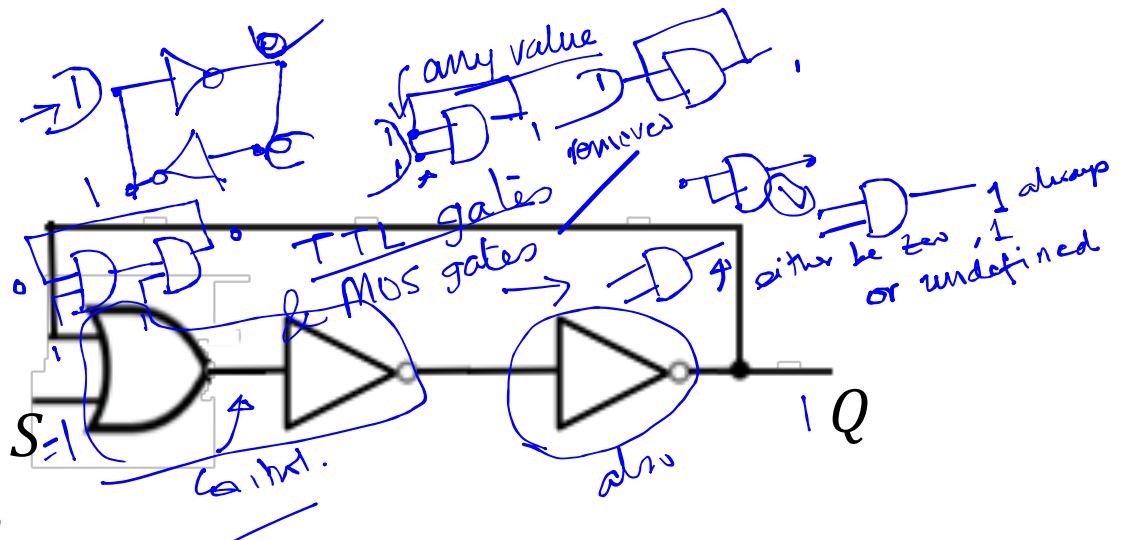
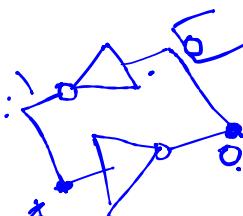
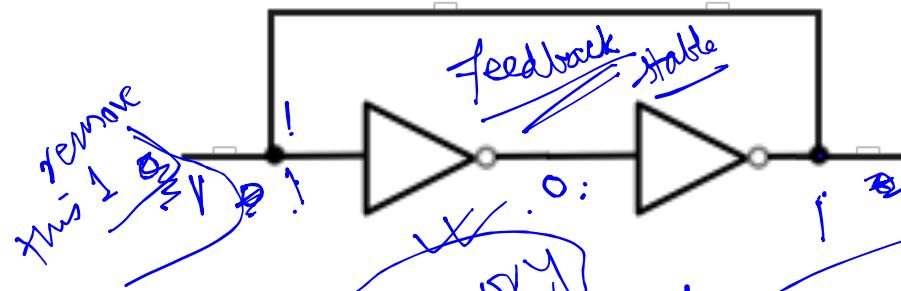


# SEQUENTIAL CIRCUITS:

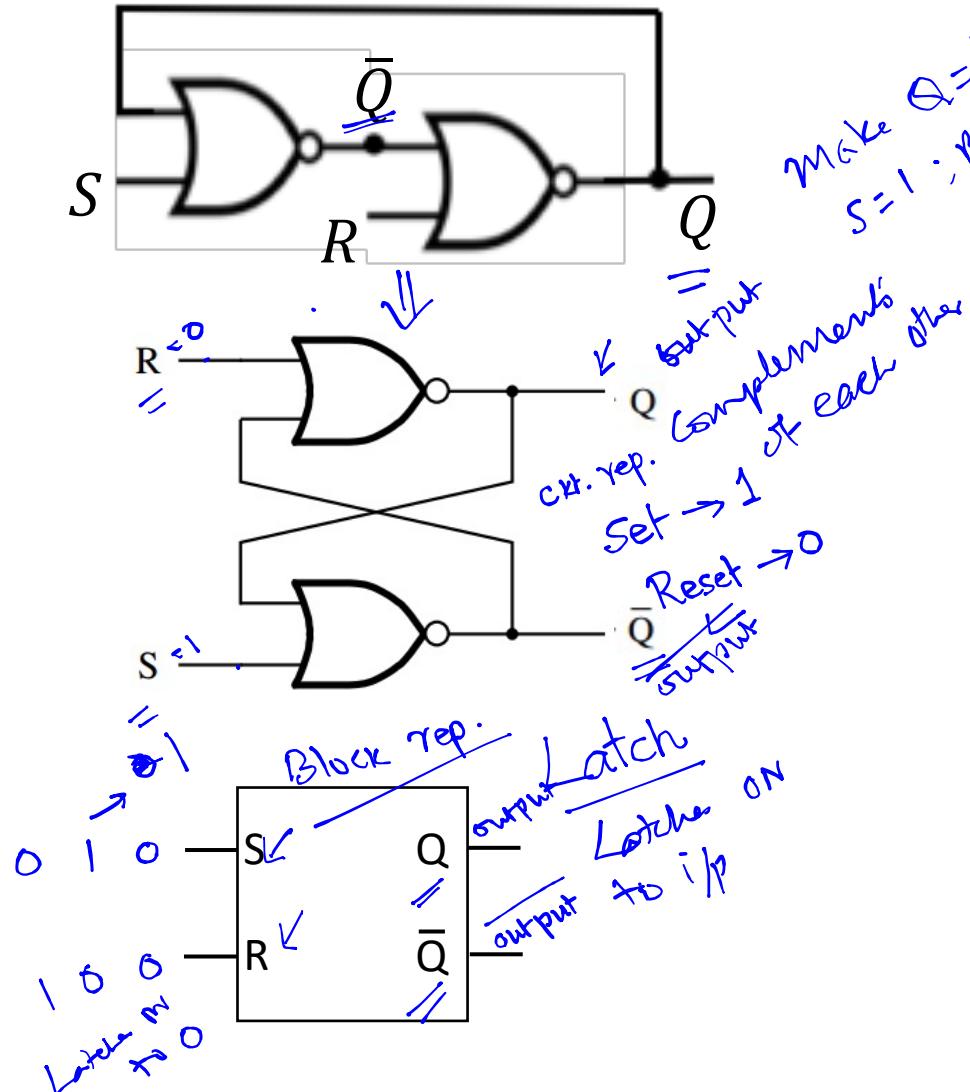
# Sequential Circuits:



## Memory Element:



# Latch: S-R Latch Using NOR Gate



$Q=0 ; R=1$  Reset

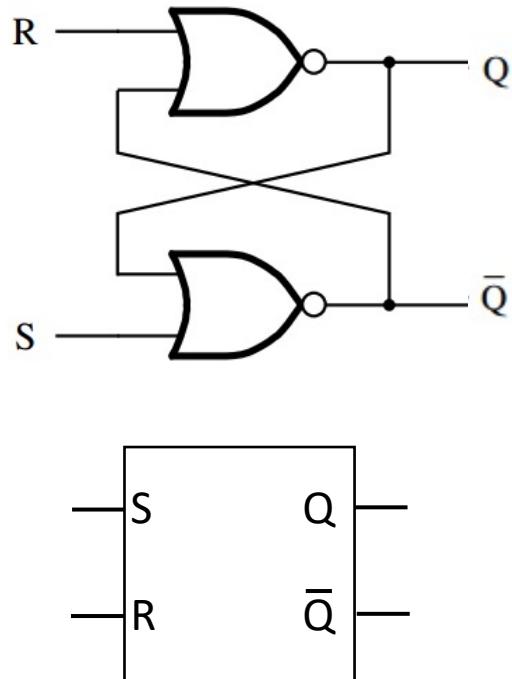
Truth Table of SR Latch

Set

$T$	$S$	$R$	$Q \leftrightarrow \bar{Q}$	
1	0	1	0 $\leftrightarrow$ 1	→ Reset state
2	0	0	0 $\leftrightarrow$ 1	Memory state
3	1	0	1 $\leftrightarrow$ 0	Set state
4	0	0	1 $\leftrightarrow$ 0	Memory state
5	1	1	0 $\leftrightarrow$ 0	forbidden state

? problematic  
don't want to use  
naive  
We will answer it later

## Latch: S-R Latch (Characteristic Table)

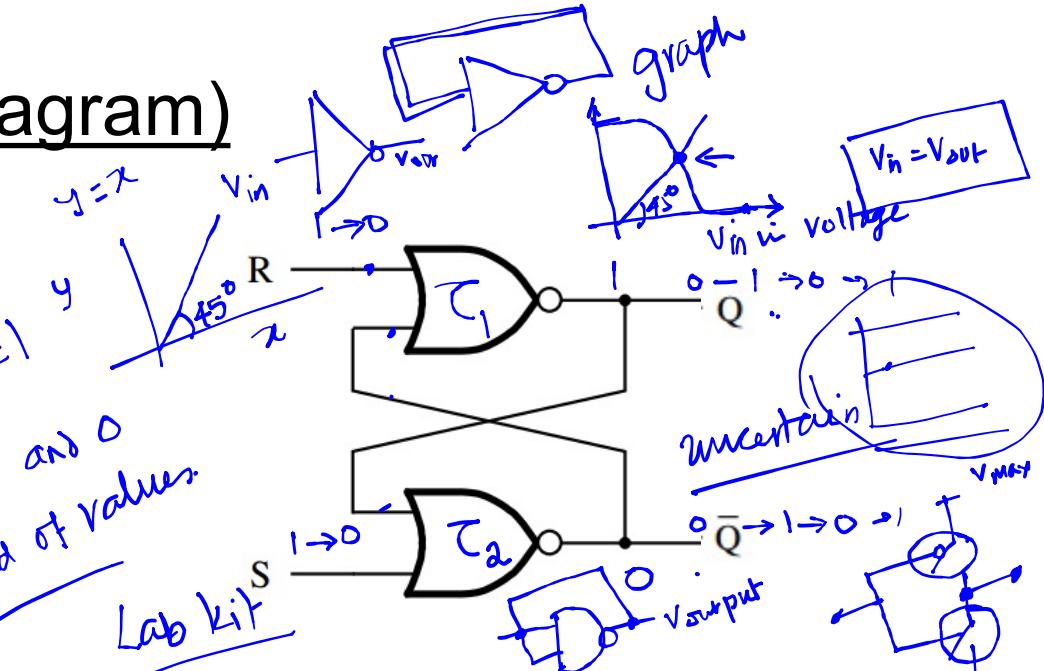
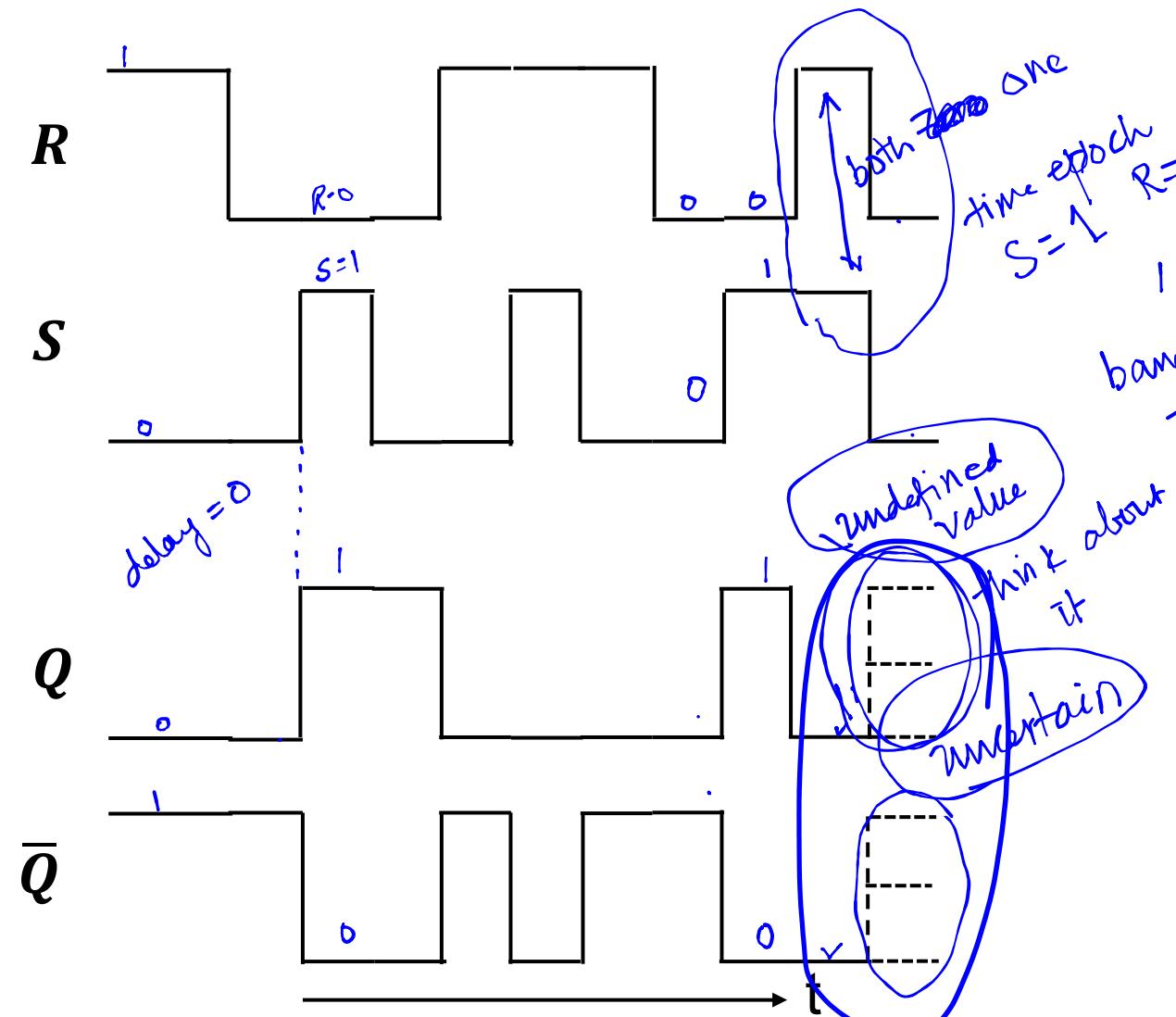


*Characteristic Table*

$S$	$R$	$Q_n$	$\bar{Q}_n$	State
0	0	$Q_{n-1}$	$\bar{Q}_{n-1}$	Memory
1	0	1	0	Set
0	1	0	1	Reset
1	1	?	?	Forbidden

*Why? address this next class*

# S-R Latch Using NOR Gate (Timing Diagram)



$S$	$R$	$Q_n$	$\overline{Q_n}$	State
0	0	$Q_{n-1}$	$\overline{Q_{n-1}}$	Memory
1	0	1	0	Set
0	1	0	1	Reset
1	1	?	?	Forbidden

