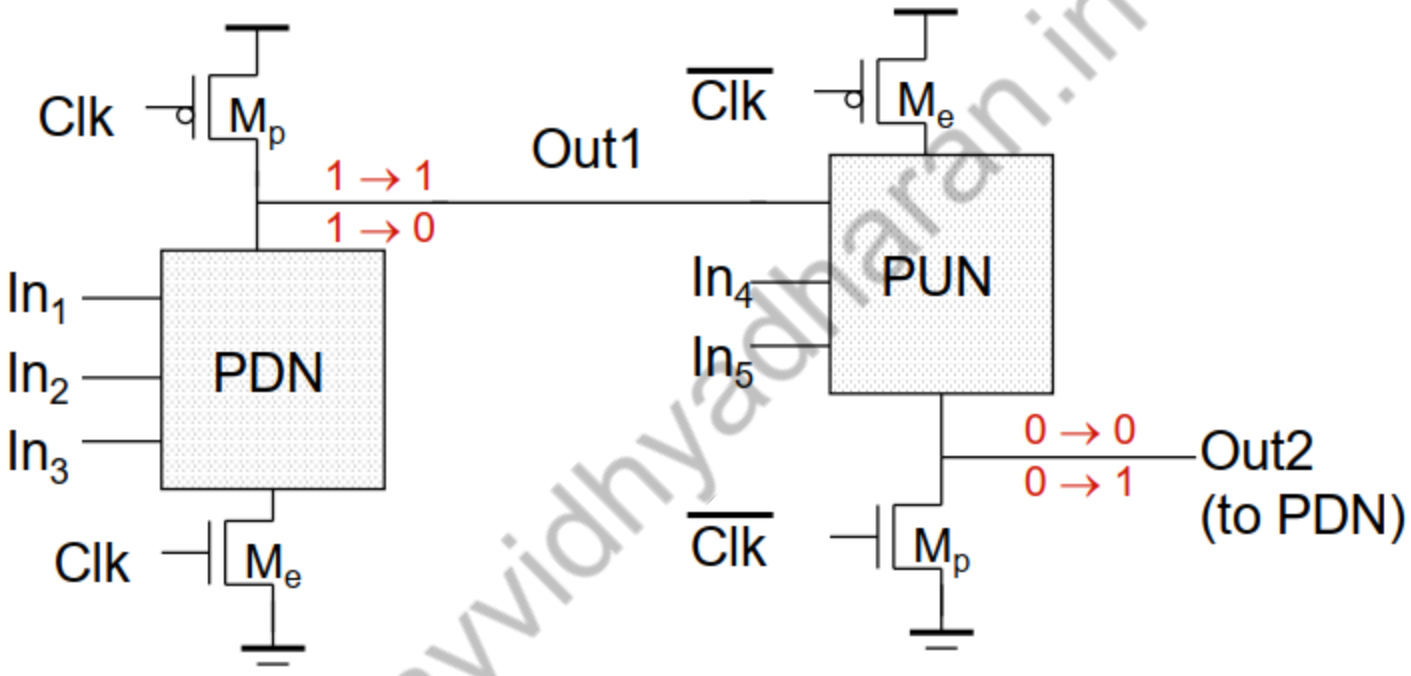
VLSI Design: 2022-23
Lecture 11
Arithmetic Circuits: Part-1

By Dr. Sanjay Vidhyadharan

np-CMOS or NORA

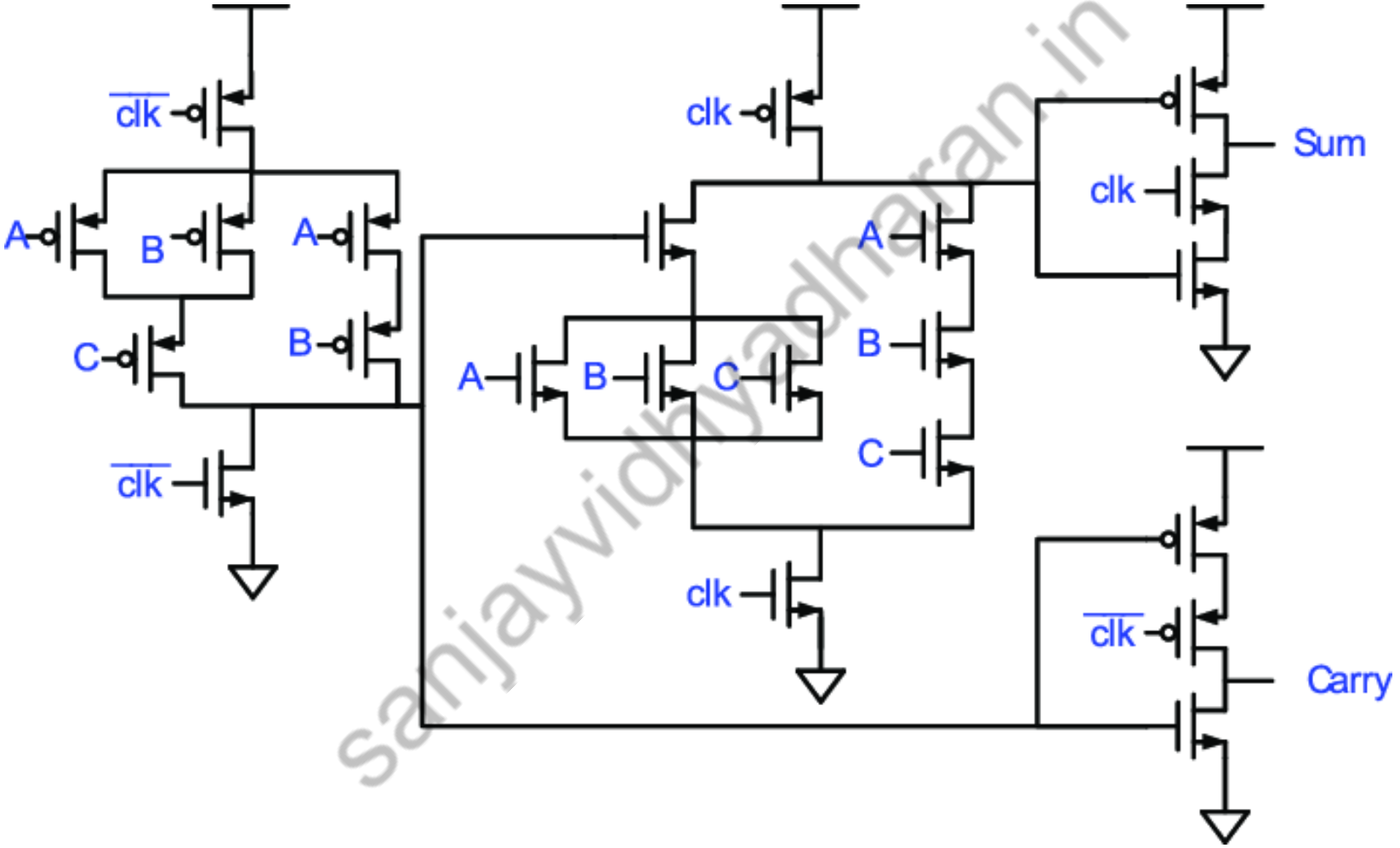


Only 0 → 1 transitions allowed at inputs of PDN

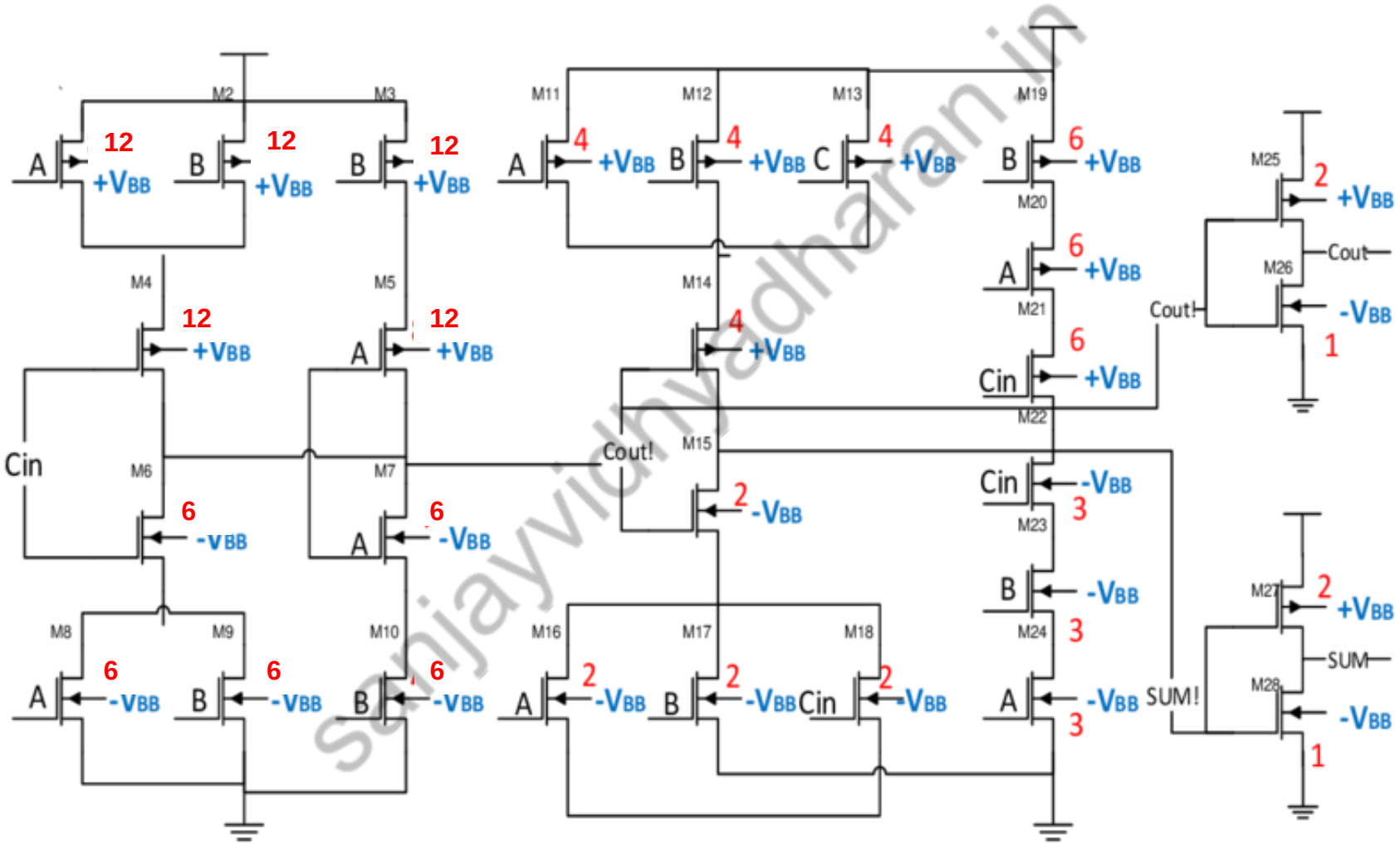
Only 1 → 0 transitions allowed at inputs of PUN

A disadvantage of the *np*-CMOS logic style is that the *p*-tree blocks are slower than the *n*-tree modules, due to the lower current drive of the PMOS transistors in the logic network. Equalizing the propagation delays requires extra area.

np-CMOS



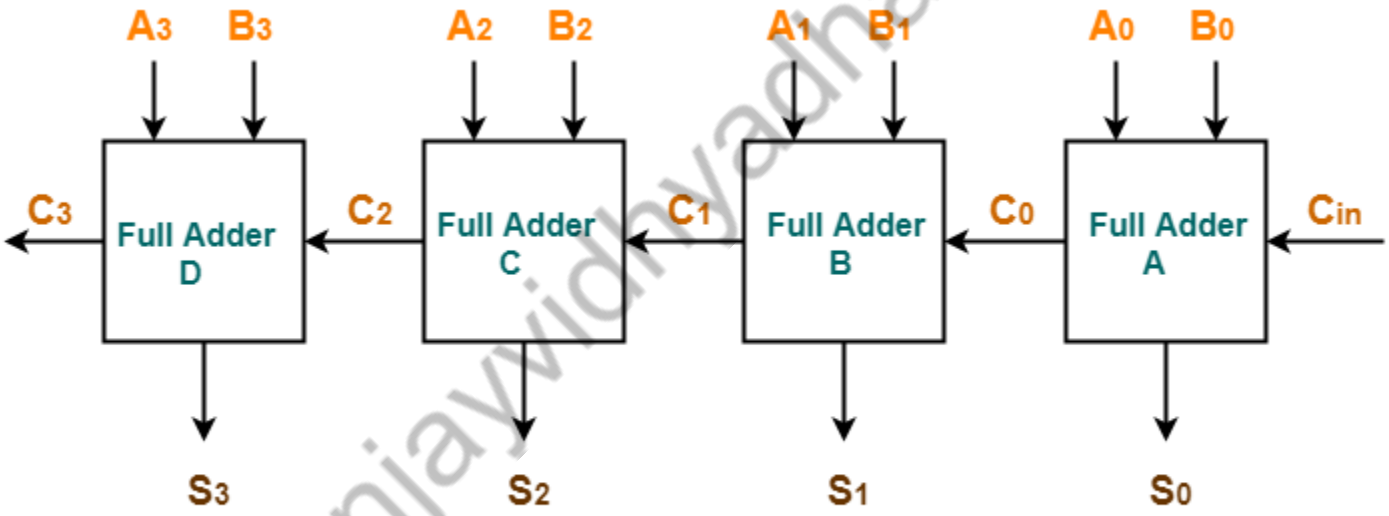
CMOS 28T Mirror Adder



Carry Delay = 2 Gate Delay and Sum = 3 Gate Delays

Ripple Carry Adder

This is called Ripple Carry Adder, because of the construction with full adders are connected in cascade.



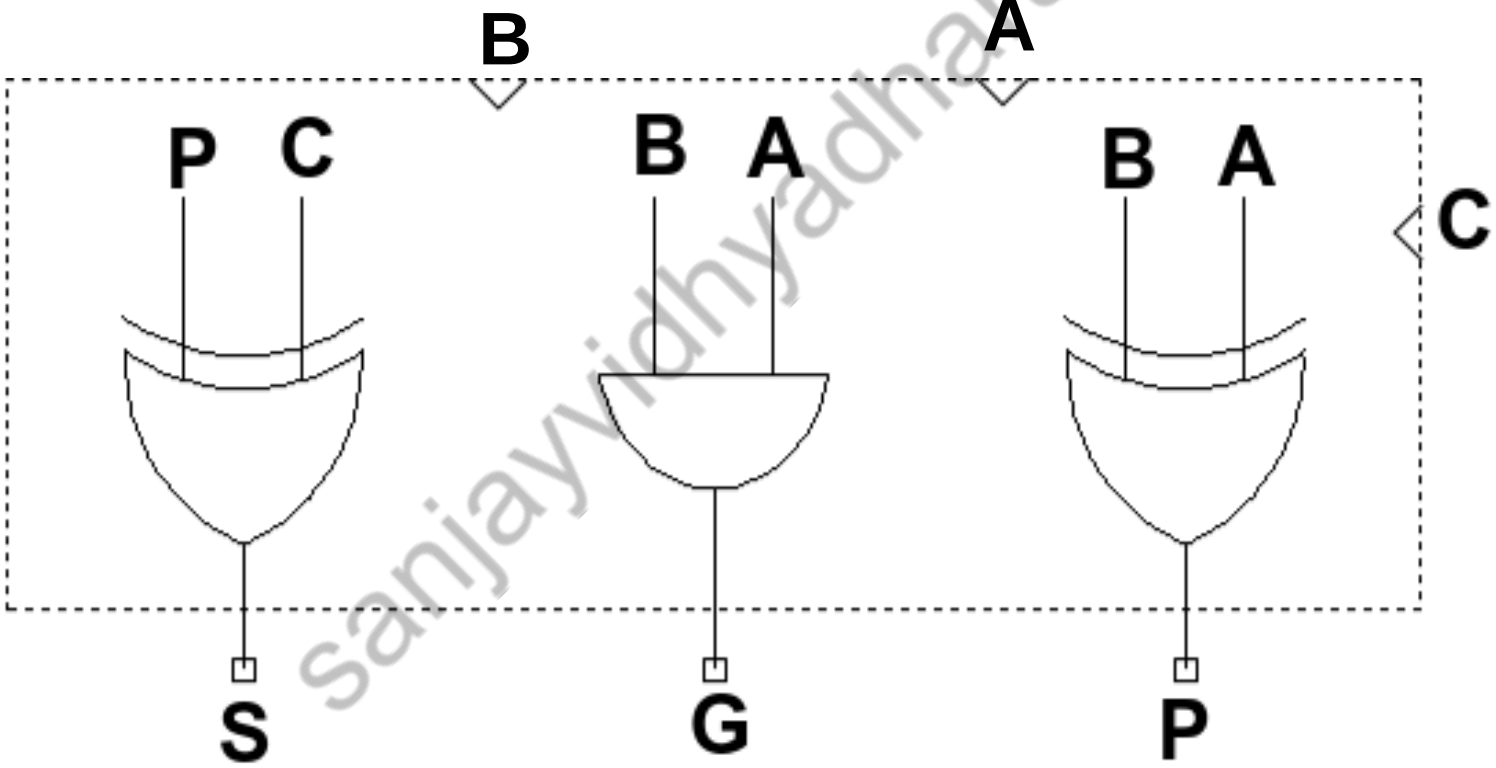
4-bit Ripple Carry Adder

Delay= 3 Carry + 1 Sum = 9 Gate Delays

Delay= (N-1) t_{carry} + t_{sum}

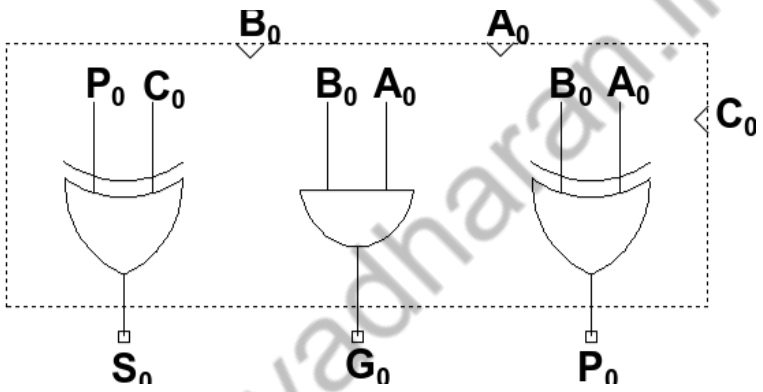
Carry Look-Ahead Adder

1-bit CLA

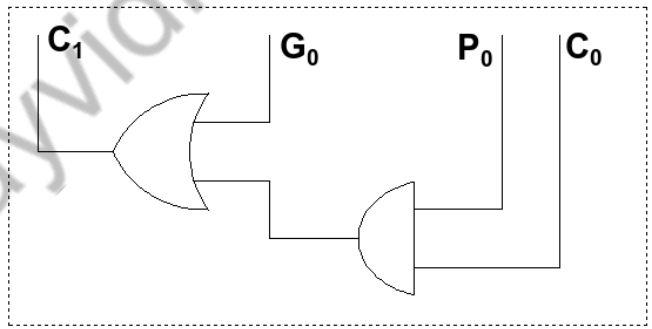


Carry Look-Ahead Adder

CLA

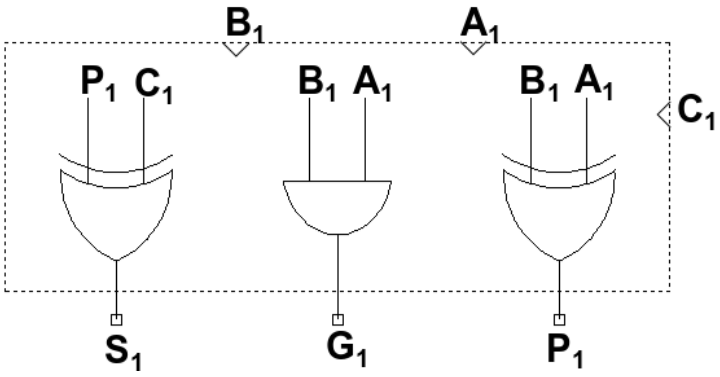


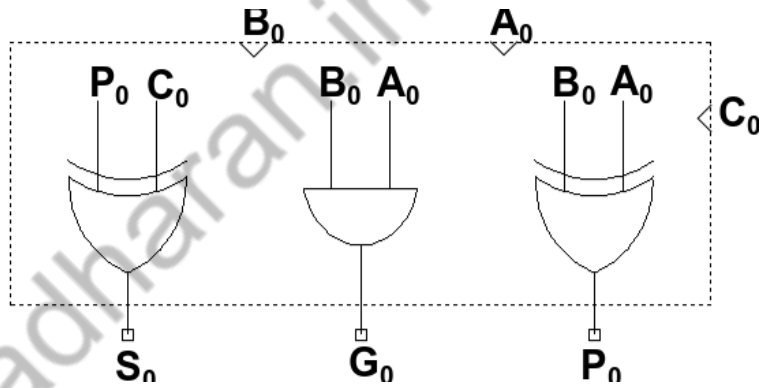
CLLB



$C_1=G_0+P_0C_0$

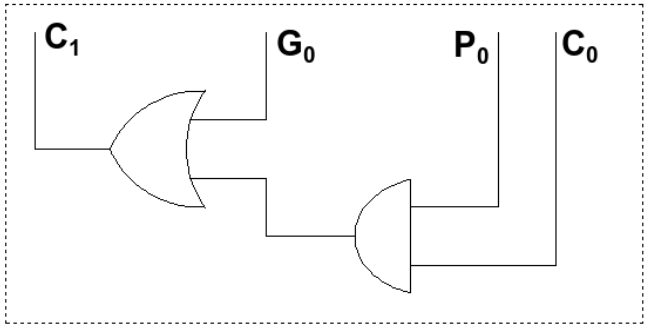
Carry Look-Ahead Adder



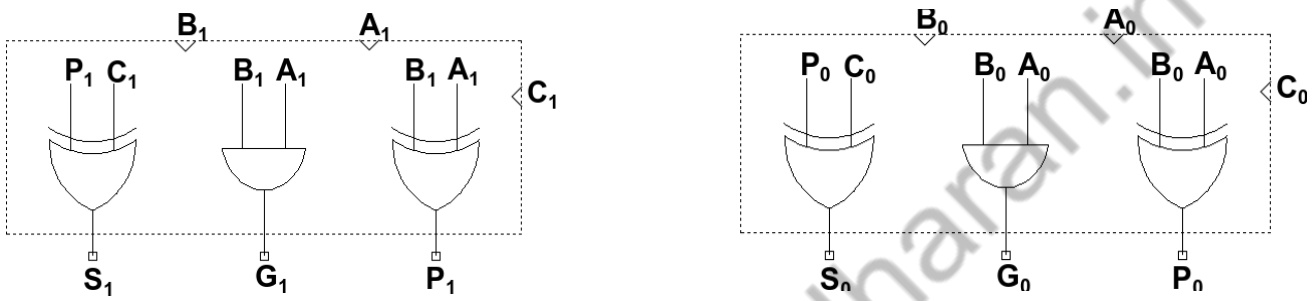


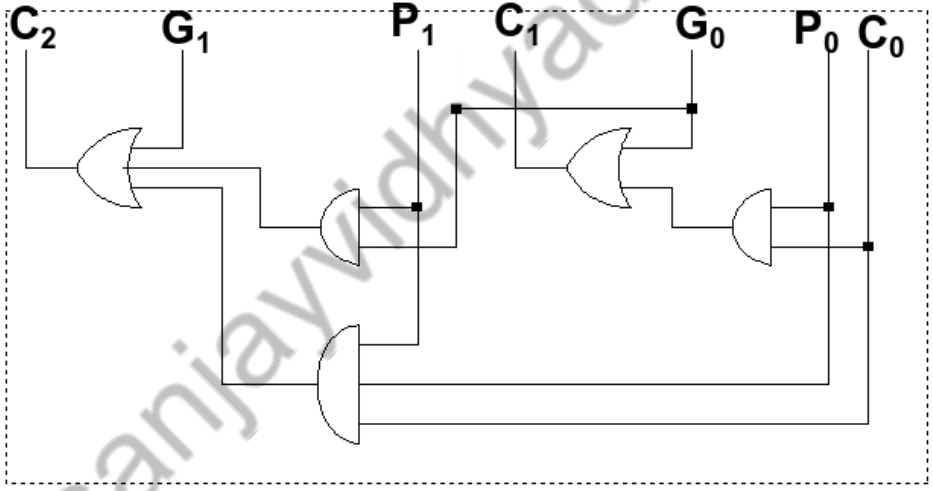
$$C_2 = G_1 + P_1 C_1$$

$$C_2 = G_1 + P_1(G_0 + P_0 C_0)$$
$$= G_1 + P_1 G_0 + P_1 P_0 C_0$$



Carry Look-Ahead Adder





$C_2 = G_1 + P_1G_0 + P_1P_0C_0$

$C_1 = G_0 + P_0C_0$

Carry Look-Ahead Adder

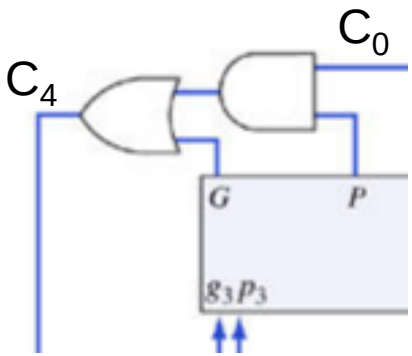
$$C_1=G_0+P_0C_0$$

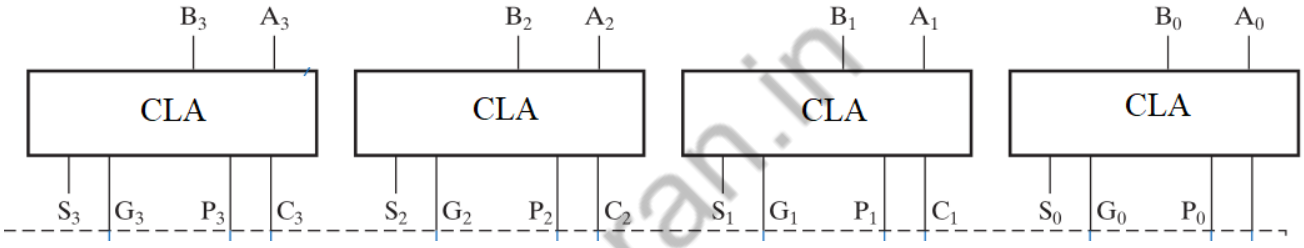
$$C_2= G_1+P_1G_0+P_1P_0C_0$$

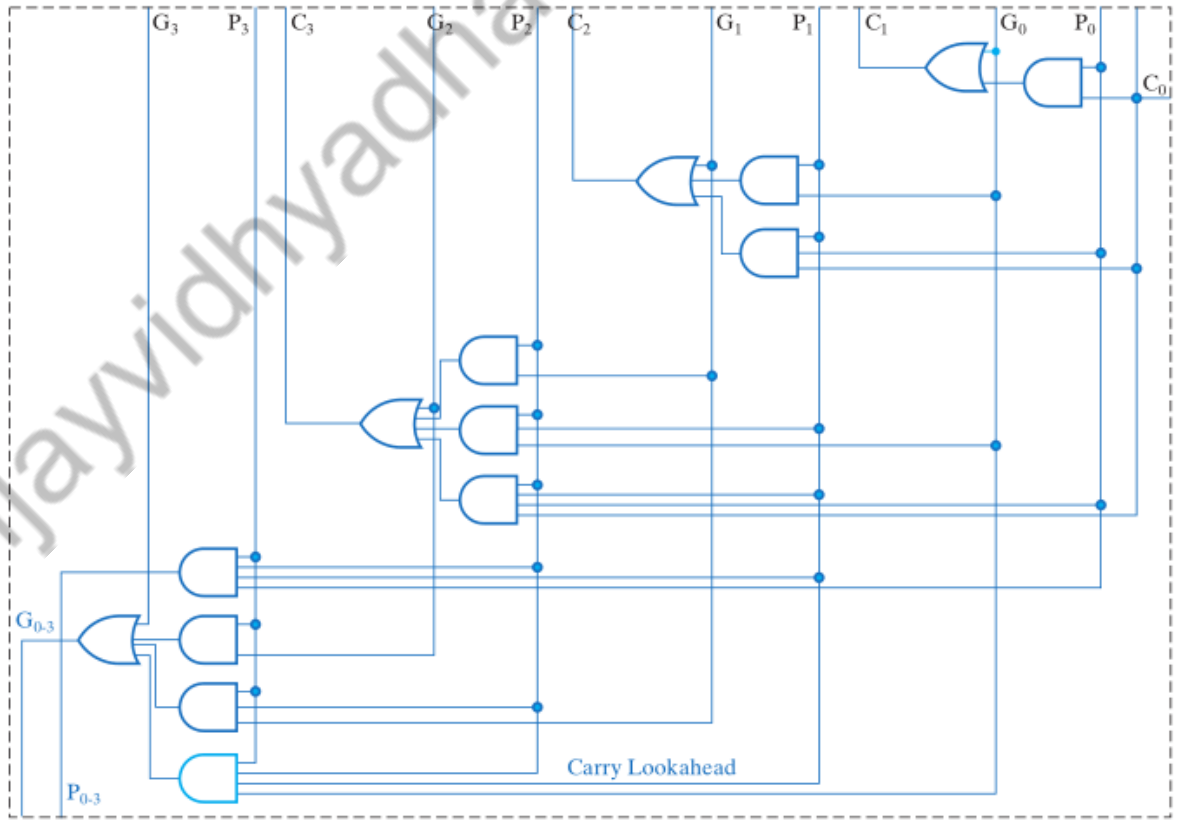
$$C_3=G_2+P_2G_1+ P_2P_1G_0 +P_2P_1P_0C_0$$

$$C_4=G_3+P_3G_2+ P_3P_2G_1 +P_3P_2P_1G_0 +P_3P_2P_1P_0C_0$$

Carry Look-Ahead Adder





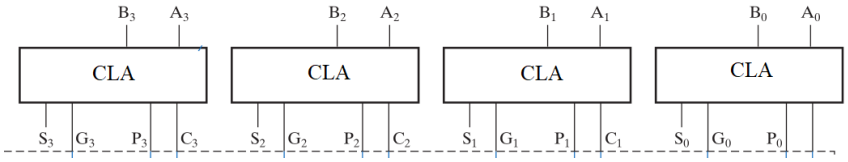


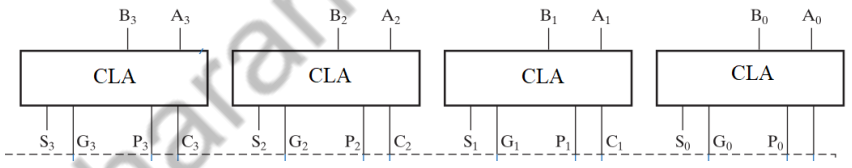
3 Gate Delay for G_3
2 Gate Delay for P_3

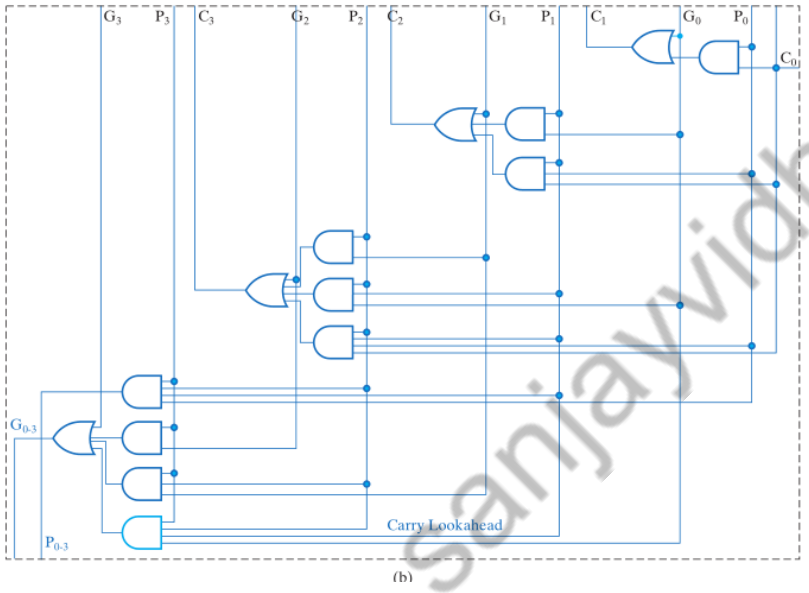
4 Gate Delay for C_4
4 Gate Delay for S_3

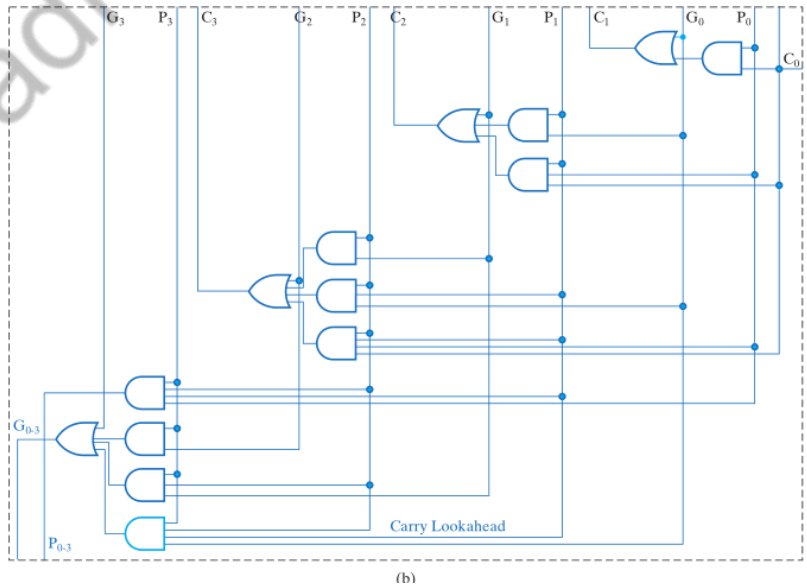
Carry Look-Ahead Adder

8 Bit Full Adder







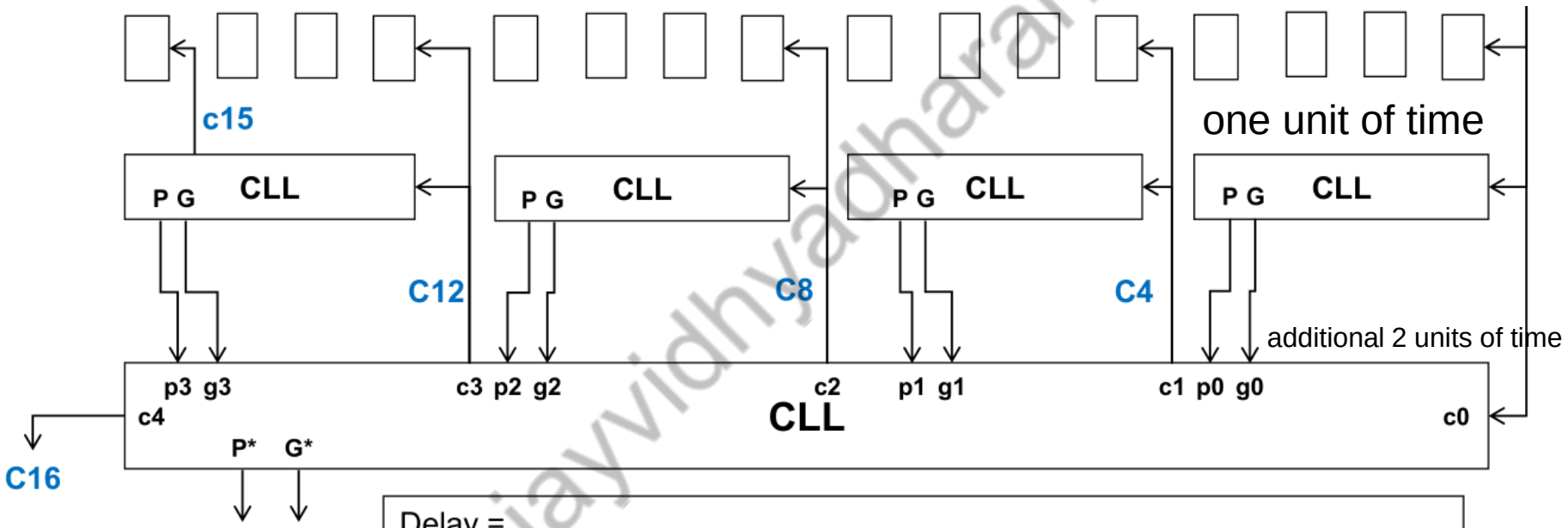


Delay = 4 + 4 = 8 Gate Delay for C_8

= 4 + 4 = 8 Gate Delay for S_7

Carry Look-Ahead Adder

16 Bit Full Adder



Delay =
= 3 = Delay in producing Pi,Gi
= 5 = Delay in producing Pi*,Gi*
= 5 = Delay in producing C4,C8,C12
= 7 = Delay in producing c15
= 8 = Delay in producing S15

6 Delays for C16

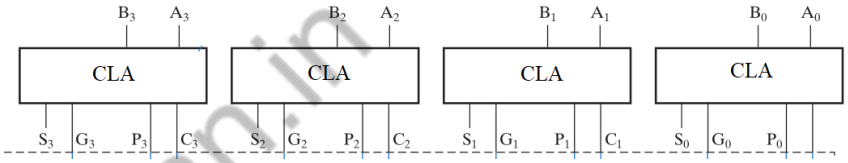
Carry Look-Ahead Adder

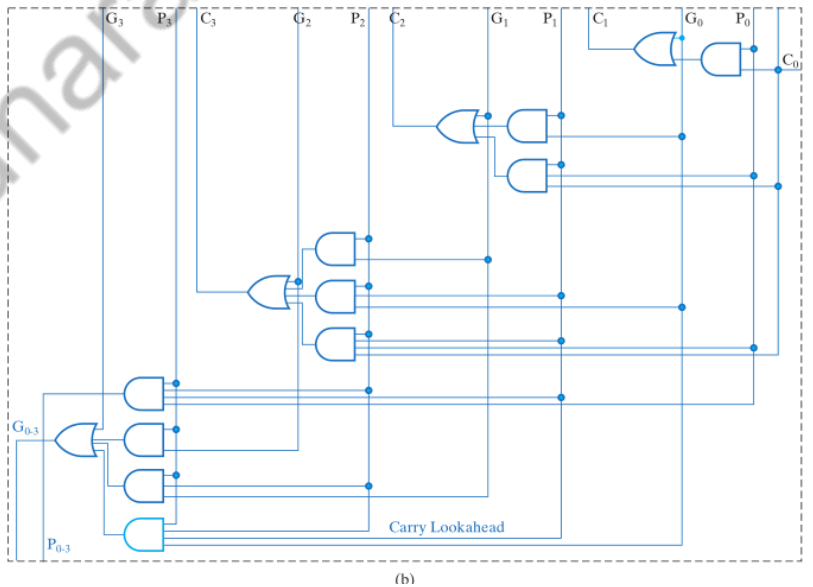
Delay of a k-bit Carry-Lookahead Adder

$$T_{\text{lookahead-adder}} = 4\lceil \log_4 k \rceil$$

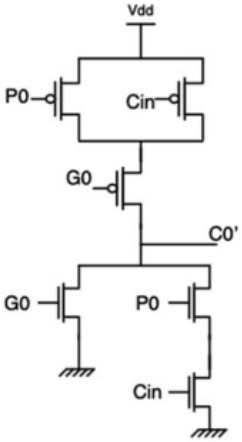
k	$T_{\text{lookahead-adder}}$	$T_{\text{ripple-carry-adder}}$
4	4	8
16	8	32
32	12	64
64	12	128
128	16	256
256	16	512

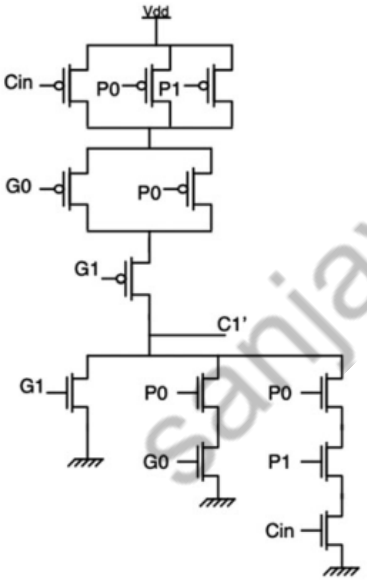
Carry Look-Ahead Adder

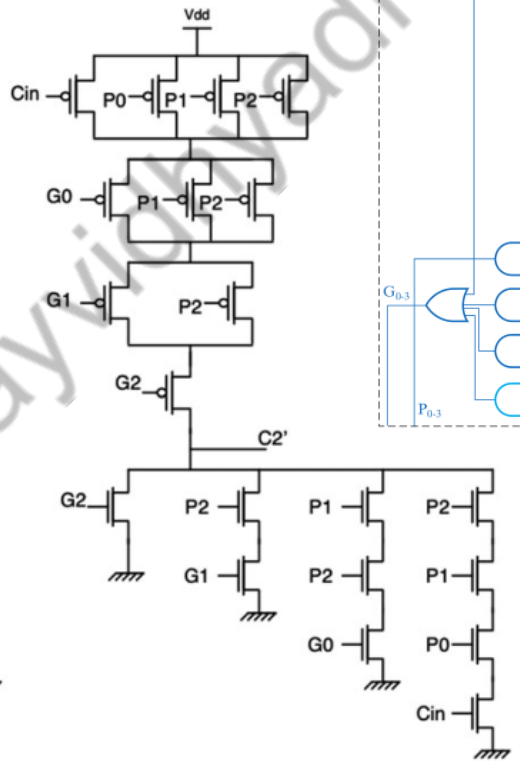




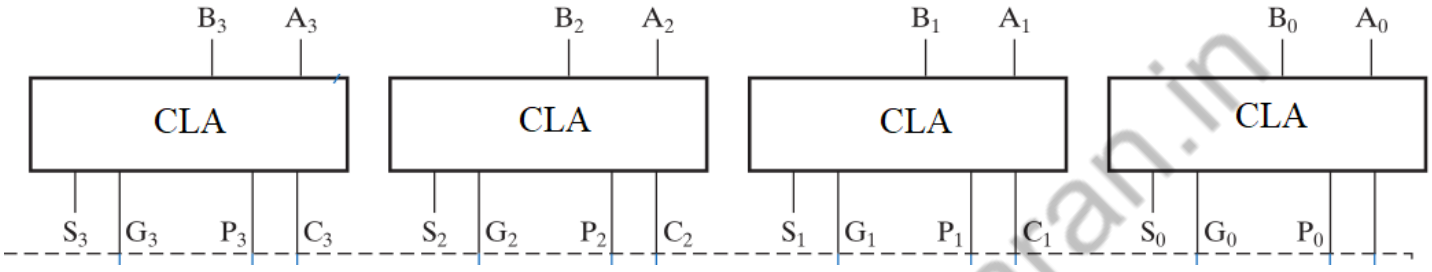
(b)

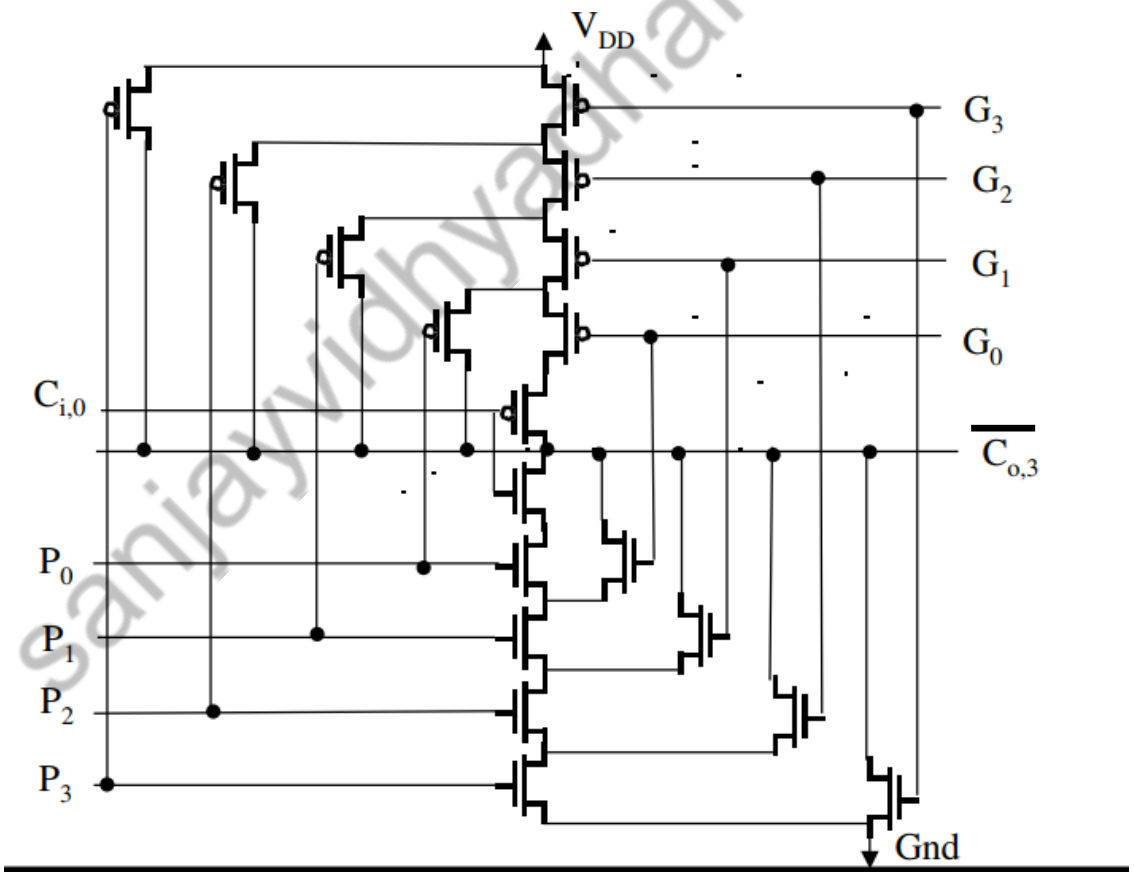




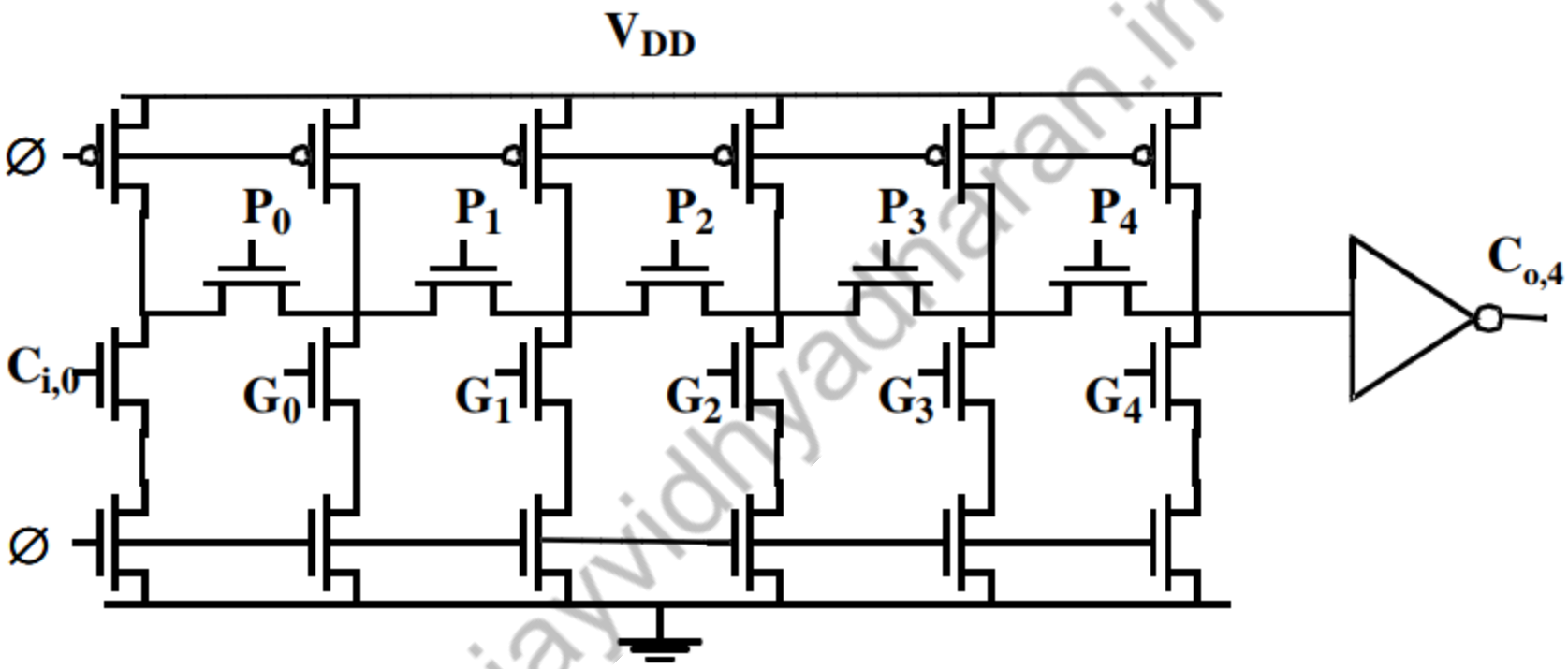


Carry Look-Ahead Adder

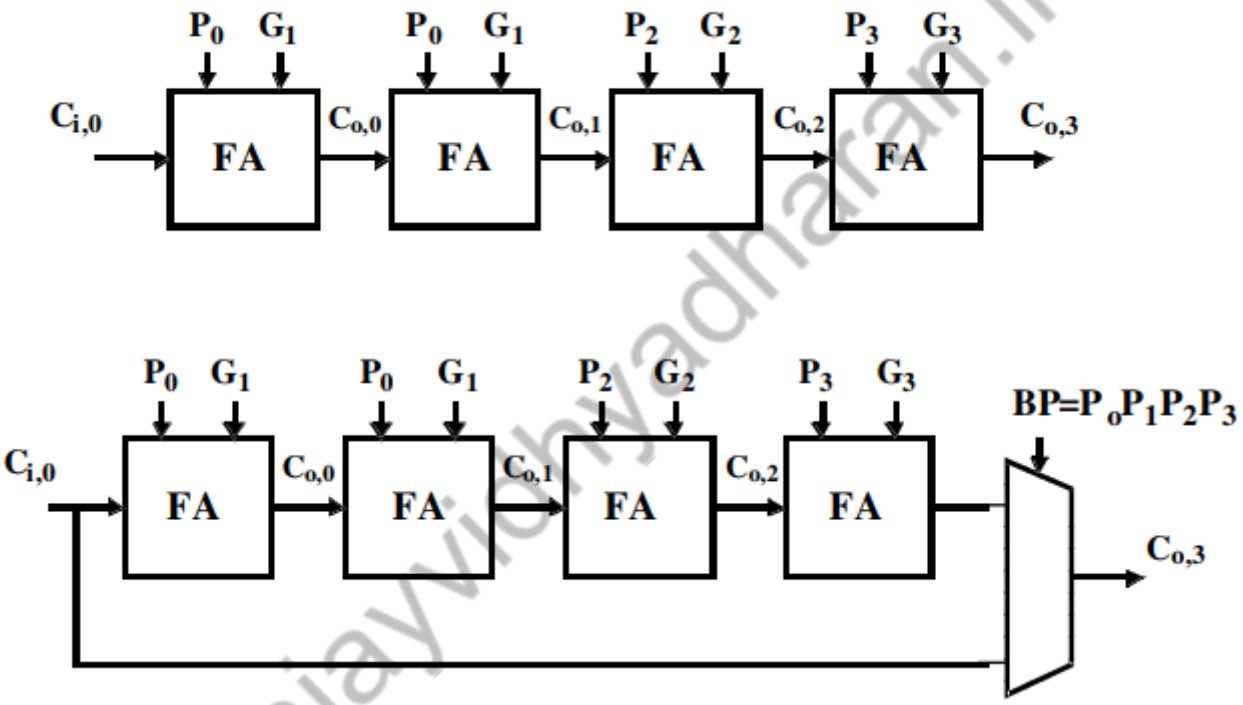




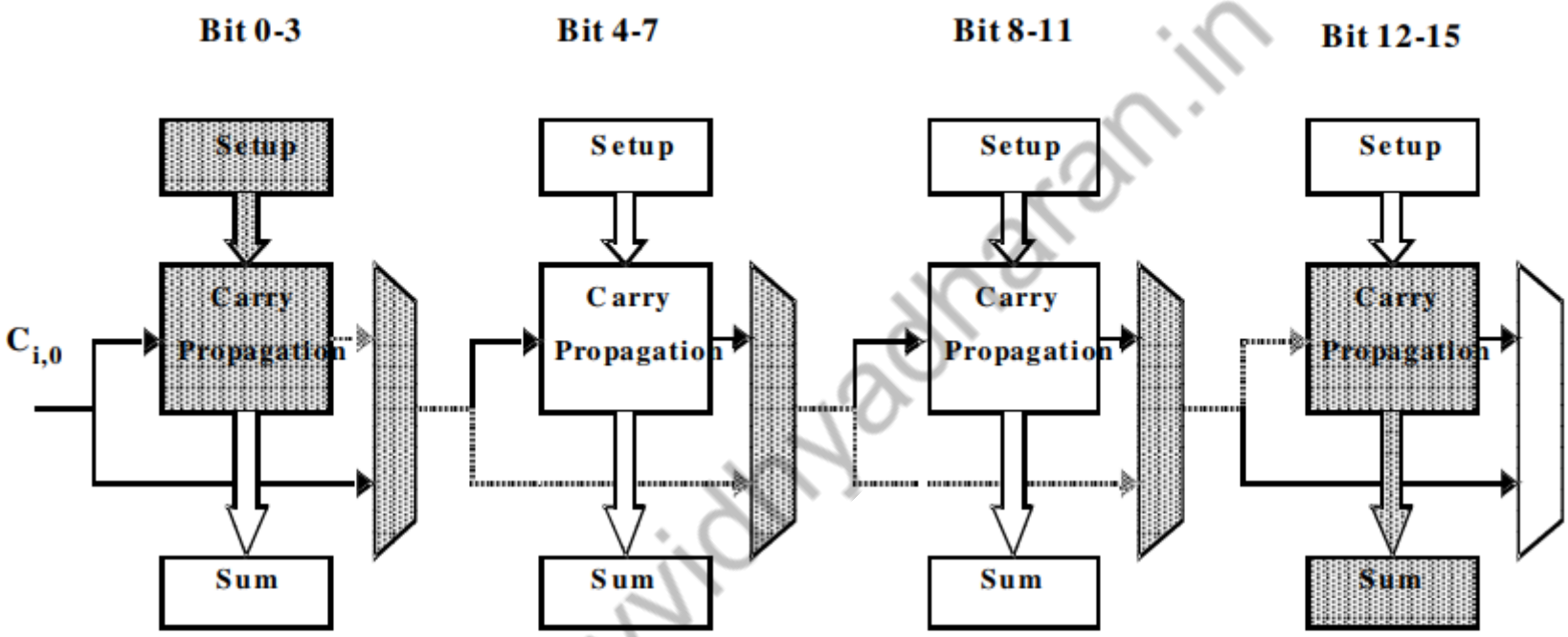
Manchester Carry Chain



Carry Bypass or Carry Skip Adder



Carry Bypass or Carry Skip Adder



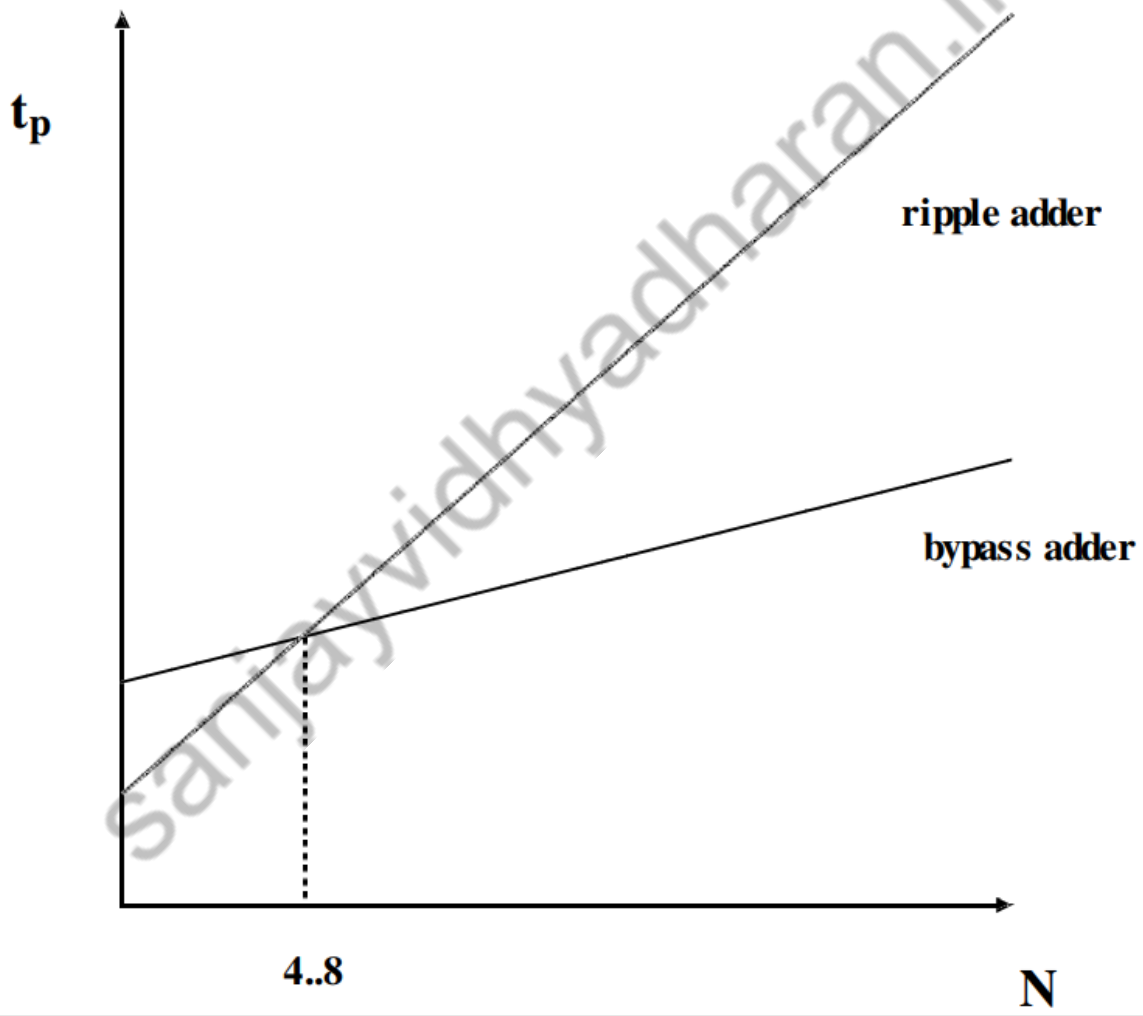
Design N-bit adder using N/M equal length stages

e.g. N = 16, M = 4

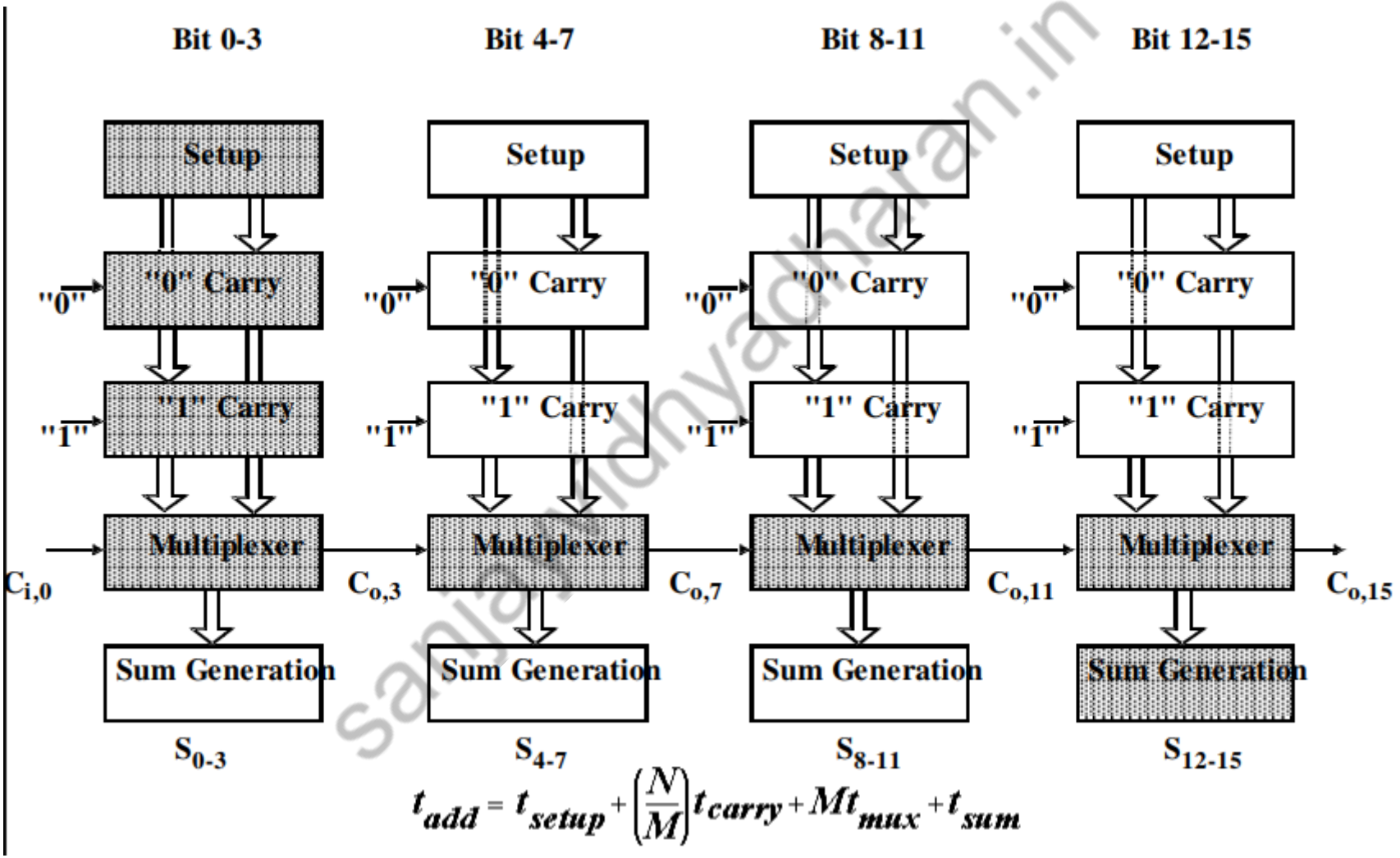
What is the critical path?

$$t_p = t_{setup} + Mt_{carry} + (N/M-1)t_{bypass} + Mt_{carry} + t_{sum}\text{ , i.e. O(N)}$$

Carry Ripple versus Carry Bypass

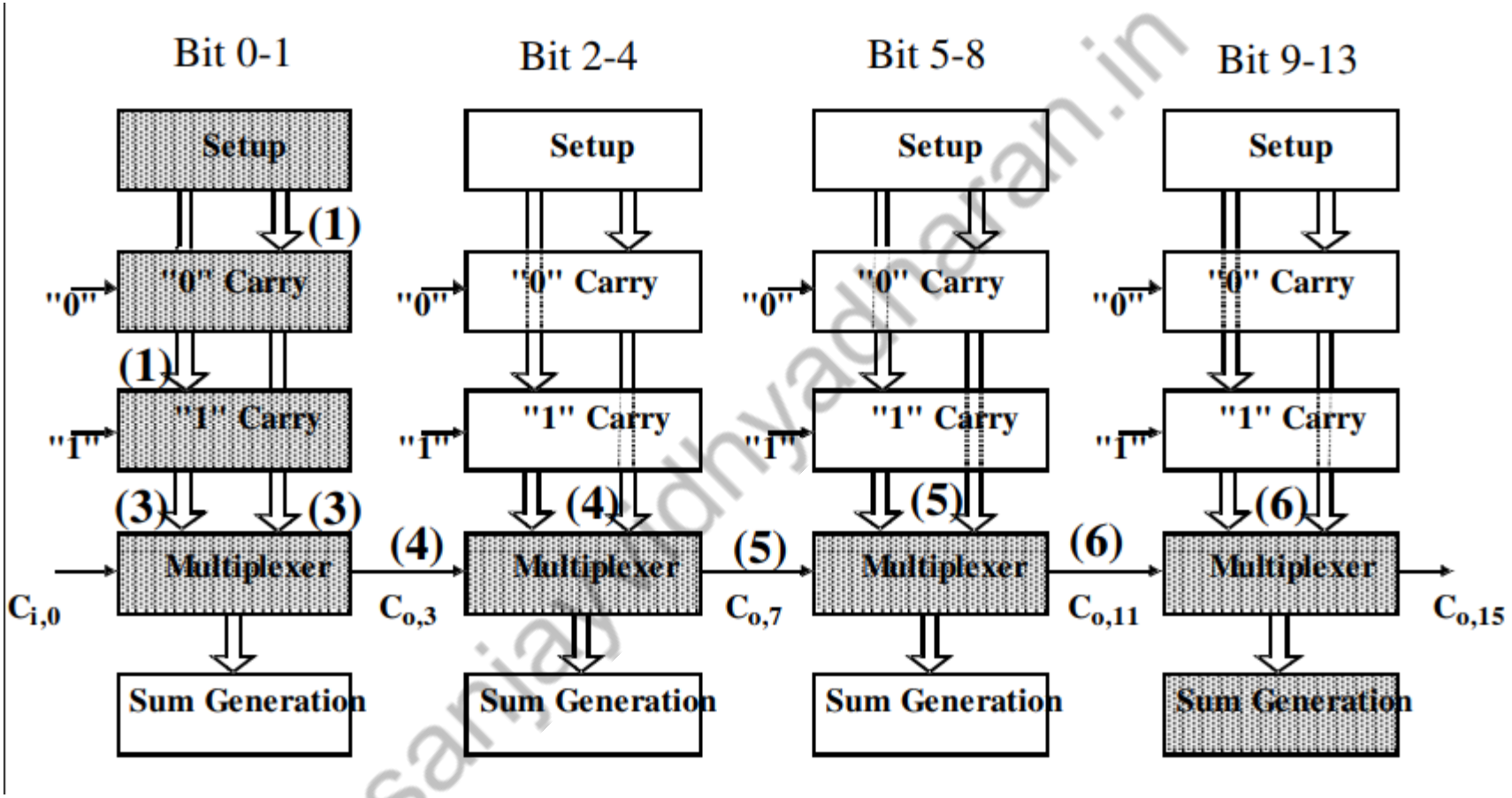


Linear Carry-Select Adder



$$t_{add} = t_{setup} + \left(\frac{N}{M}\right) t_{carry} + M t_{mux} + t_{sum}$$

Square Root Carry-Select Adder



$$t_{add} = t_{setup} + Mt_{carry} + \left(\sqrt{2N}\right)t_{mux} + t_{sum}$$

Square Root Carry-Select Adder

N *Bit adder,* $M -$ *Bits in First Stage ,* $P -$ *Number of Stages*

$$N = M + (M+1) + (M+2) + (M+3) + \cdots . + (M + P - 1)$$

Series: a, a+d, a+2d,……,a+(n-1)d

s_n = n/2(2a + (n-1)d)

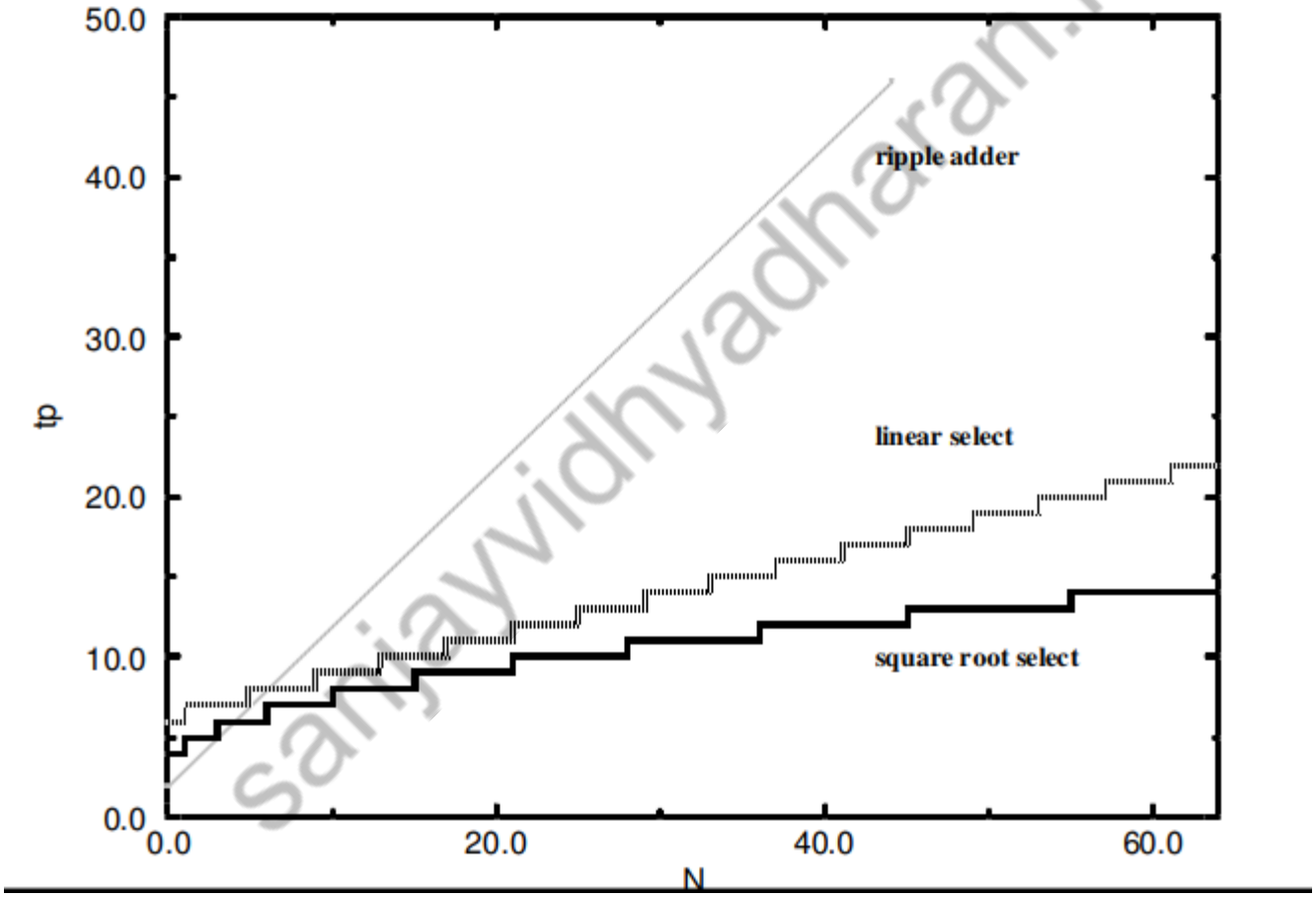
$$N = MP + \frac{P(P-1)}{2}$$

$$N = \frac{P^2}{2} + P(M - \frac{1}{2})$$

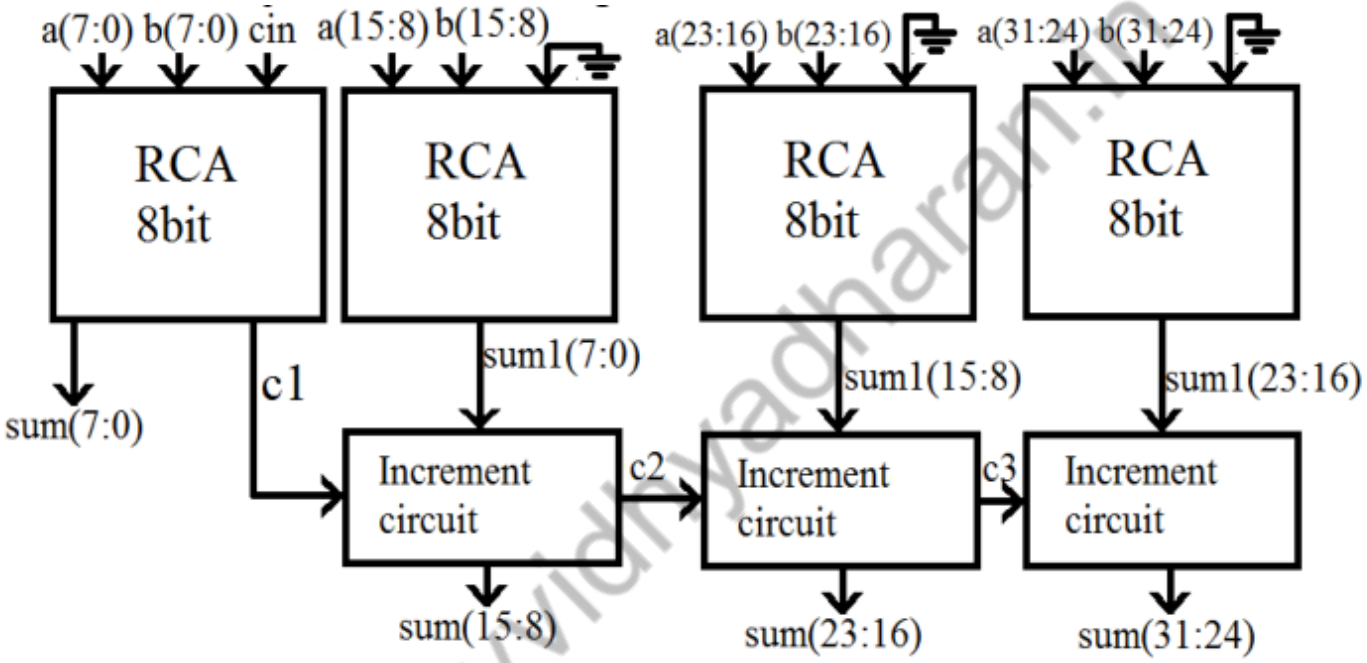
$$M \ll N \; (e.M = 2 \; and \; N = 64)$$

$$N \approx \frac{P^2}{2}$$

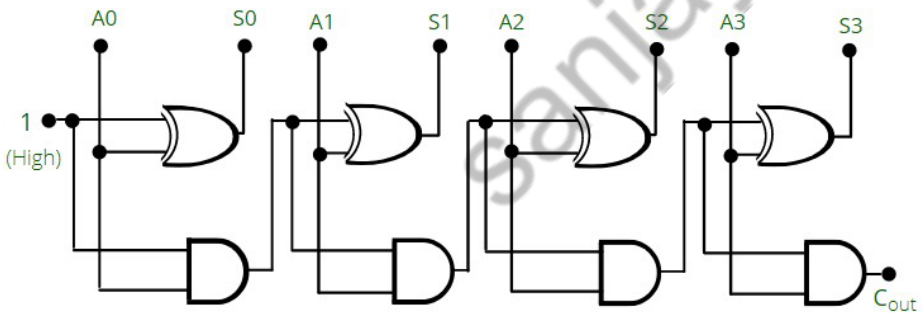
Adder Delays - Comparison



Carry Increment Adder



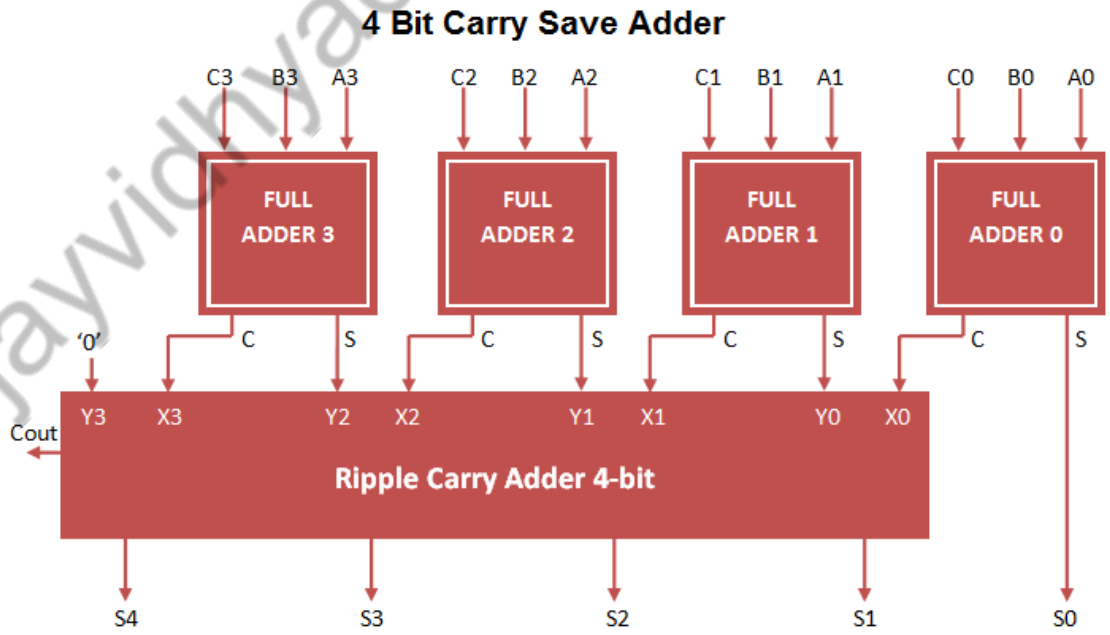
RCA - 65



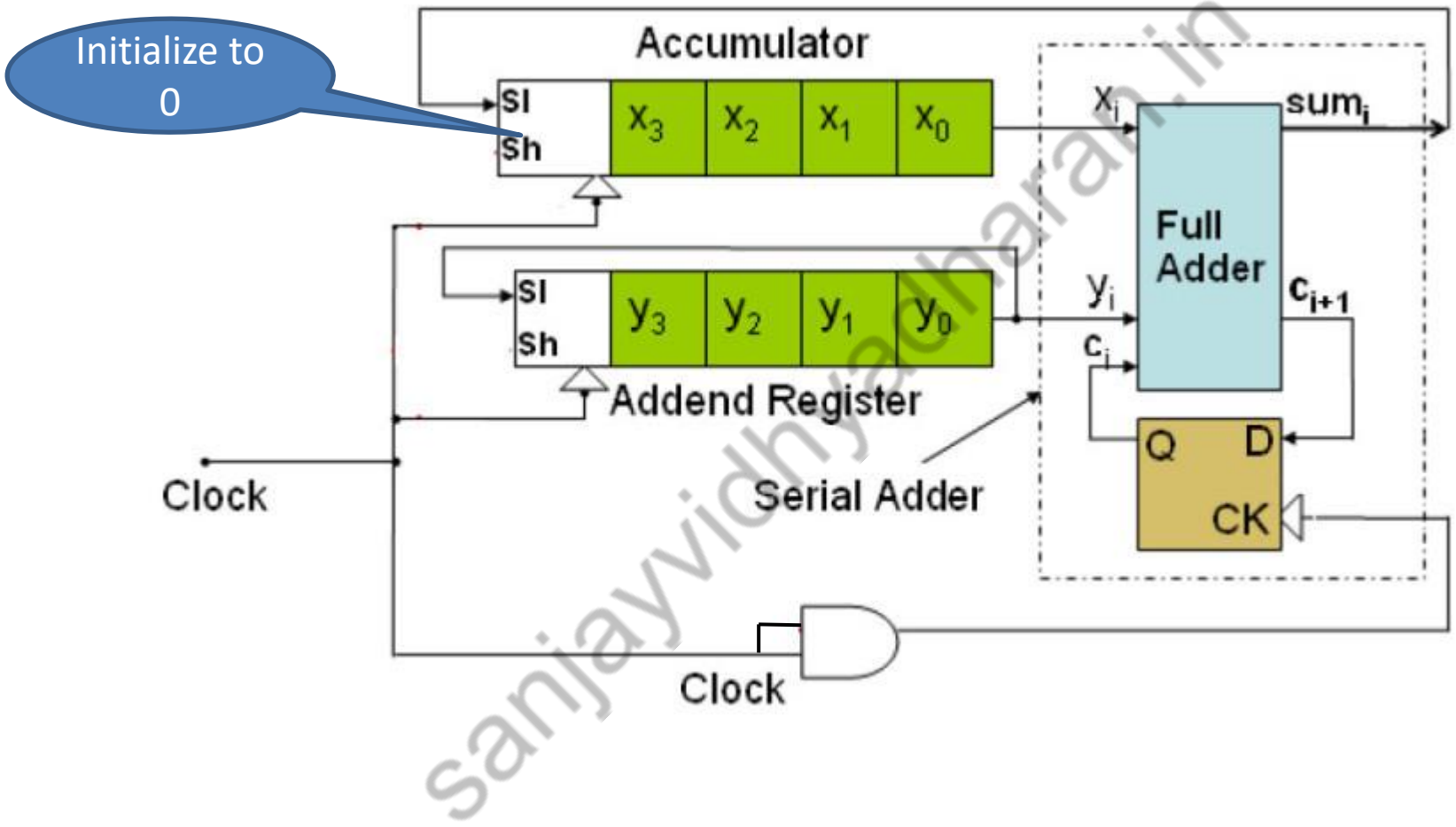
CIA-16+8+8+8=40

Carry Save Adder

X: 1 0 0 1 : 9
Y: 1 0 0 1 : 9
Z: 1 0 1 1 : 11
S: 1 0 1 1
C: 1 0 0 1
Sum: 1 1 1 0 1 : 29



Serial Adder



Block Diagram of a 4-bit Serial Adder with Accumulator

4 Bit-Adder Subtractor

Add 4 & -3

0100
1101
1 0001

Add -4 & -5

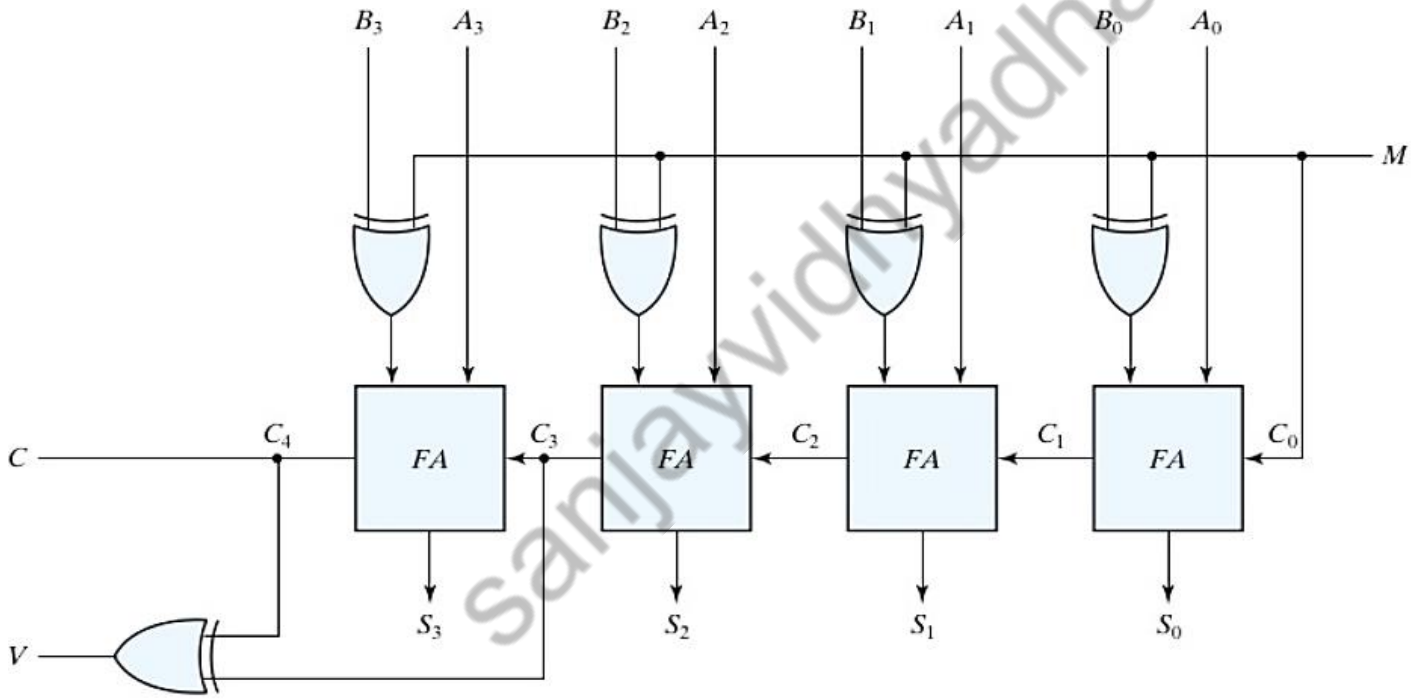
1100
1011
1 0111

Add -8 & 4

1000
0100
1100

Add 4 & 4

0100
0100
1000



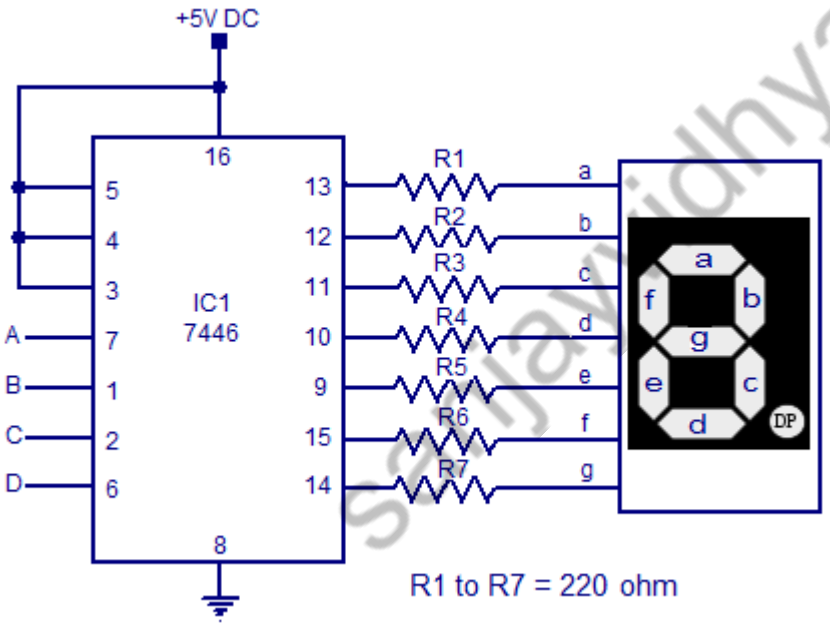
Overflow

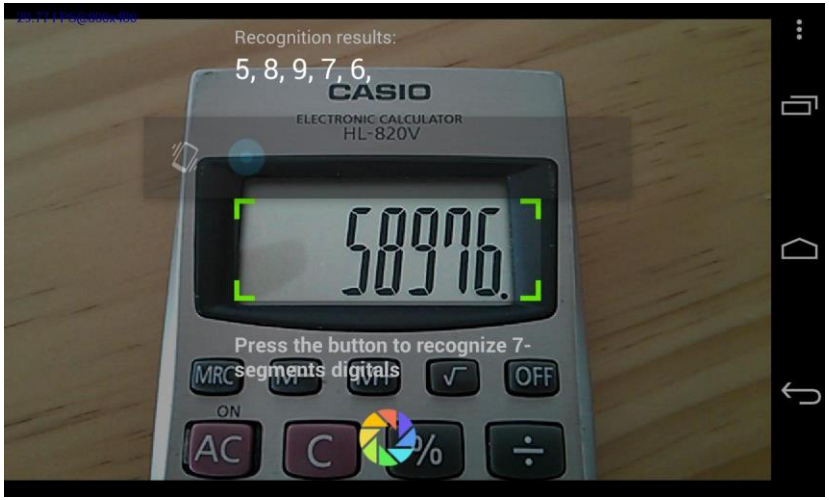
Decimal	2's comp.
7	0111
6	0110
5	0101
4	0100
3	0011
2	0010
1	0001
0	0000
-0	–
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

Binary Coded Decimal

General digital systems

User enters decimal → BCD i/p→ Binary i/p → compute in binary → Binary o/p → BCD o/p → Decimal output shown to user





Binary Coded Decimal

BCD addition

4 + 5	4 0 1 0 0	
	5 0 1 0 1	
	9 1 0 0 1	Expected Result
4 + 8	4 0 1 0 0	
	8 1 0 0 0	
	1 1 0 0	Is this expected Result ?

Expected answer is BCD of 12 0001 0010

Binary Coded Decimal

BCD addition

4 + 8

4 0 1 0 0

8 1 0 0 0

Greater than 9 1 1 0 0

0 1 1 0

0 0 0 1 0 0 1 0

1 2

Add correction of +6

= To skip 6 invalid states (10 - 15) BCDs

Binary Coded Decimal

BCD addition

9 + 9

9 1 0 0 1

9 1 0 0 1

Carry out generated 1 0 0 1 0 Expected result ?

0 1 1 0 Add correction of +6

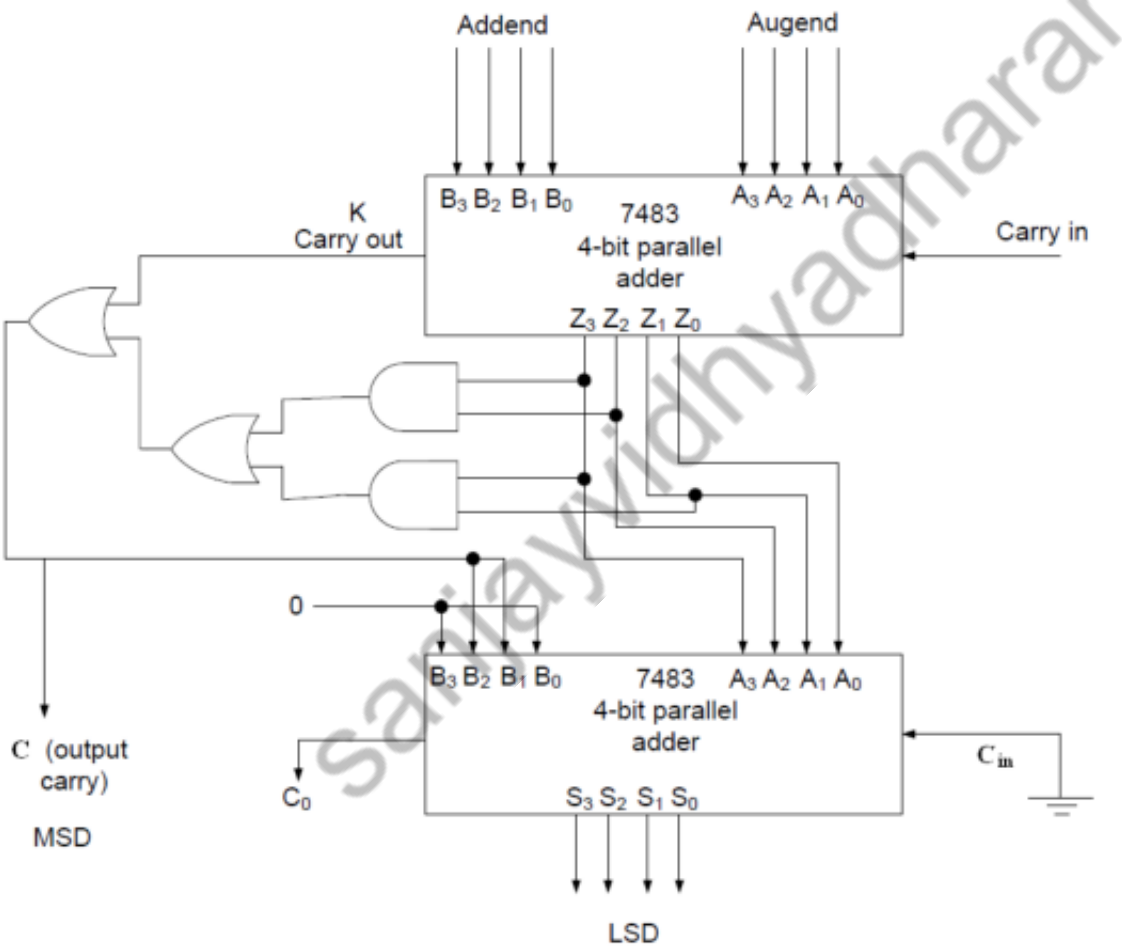
0 0 0 1 1 0 0 0

1 8

After addition if carry out is generated or if sum is greater than 9 there is need for correction

Binary Coded Decimal

BCD addition



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

Thank you