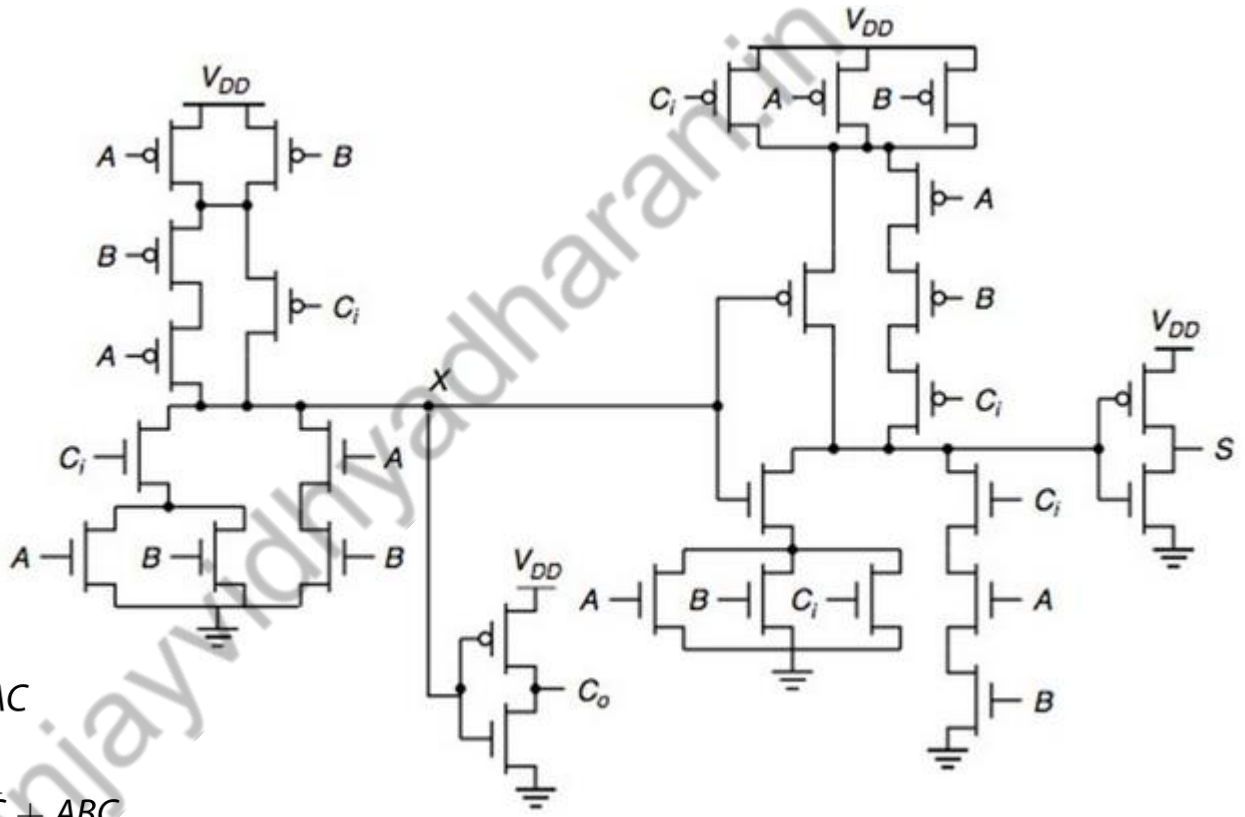
# Advanced VLSI Design: 2022-23
# Lecture 11
# Arithmetic Circuits: Part-1

**By Dr. Sanjay Vidhyadharan**

# CMOS 28T Adder

| A | B | Carry In | Sum | Carry out |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

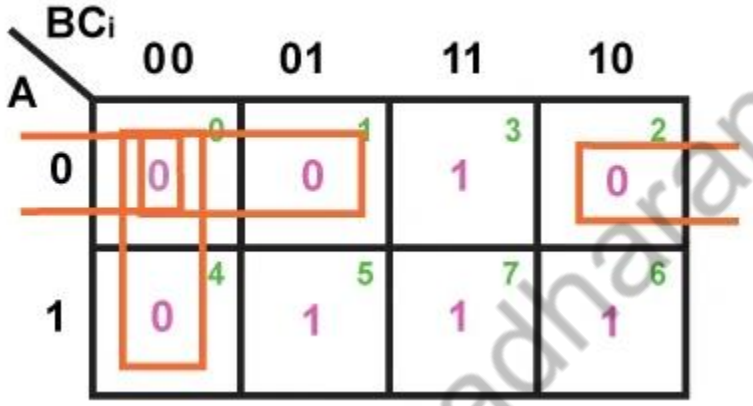



$$Carry(CY) = AB + BC + AC$$

$$Sum(S) = \bar{A}\bar{B}C\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC$$

$$\left.\begin{aligned} CY &= AB + C(A+B) \\ S &= \overline{CY}(A+B+C) + ABC \end{aligned}\right\} \text{Simplified Expressions}$$

# CMOS 28T Mirror Adder

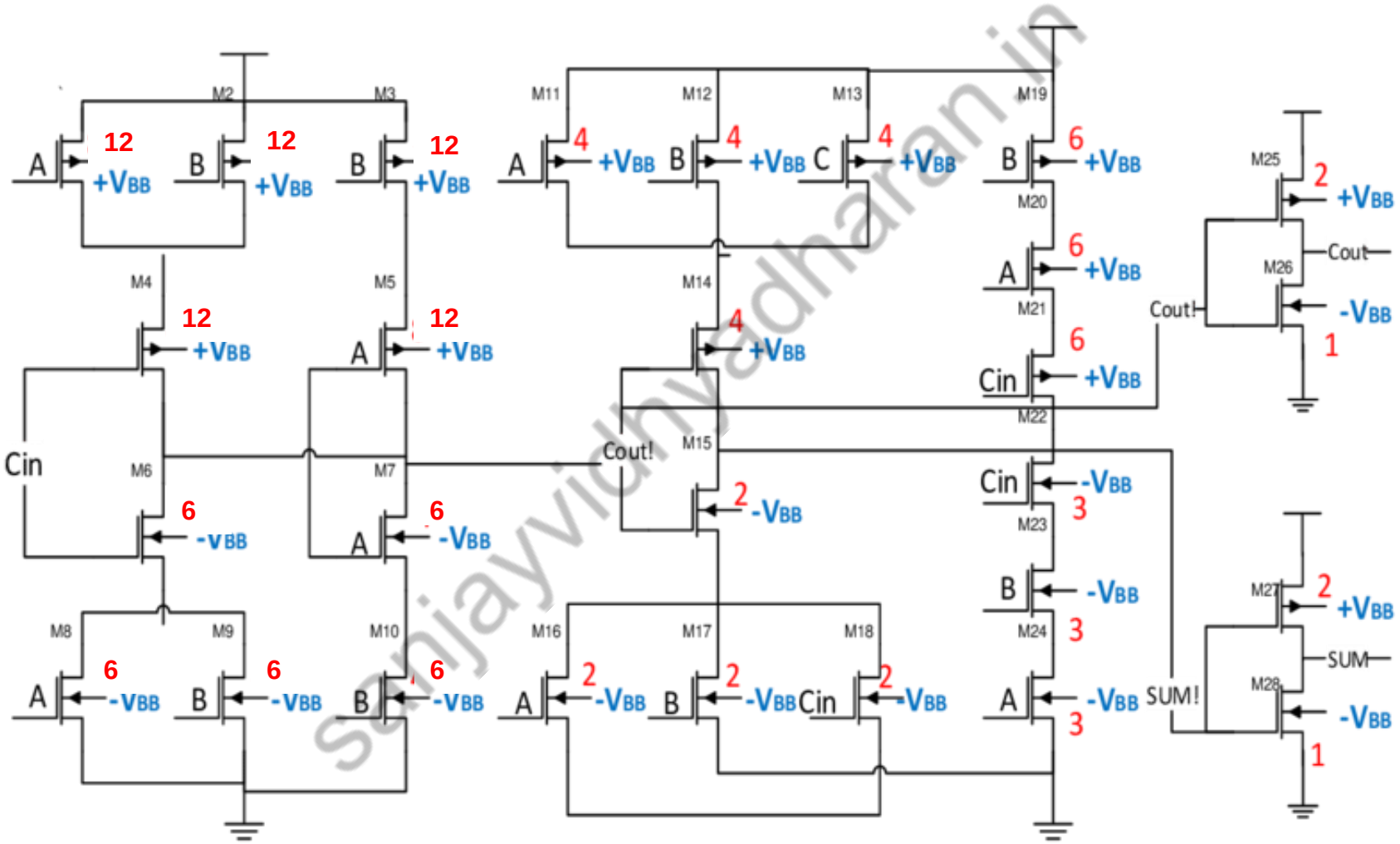



$$Carry = AB + BC + CA = AB \; + \; C(A + B)$$

$$Pull - Down\ Network\ for\ Carry\ Bar = AB \; + \; C(A + B)$$

$$Pull - UP\ Network\ for\ Carry\ Bar = A'B' + B'C' + C'A'$$
$$= A'B' + \; C'(A' + B')$$

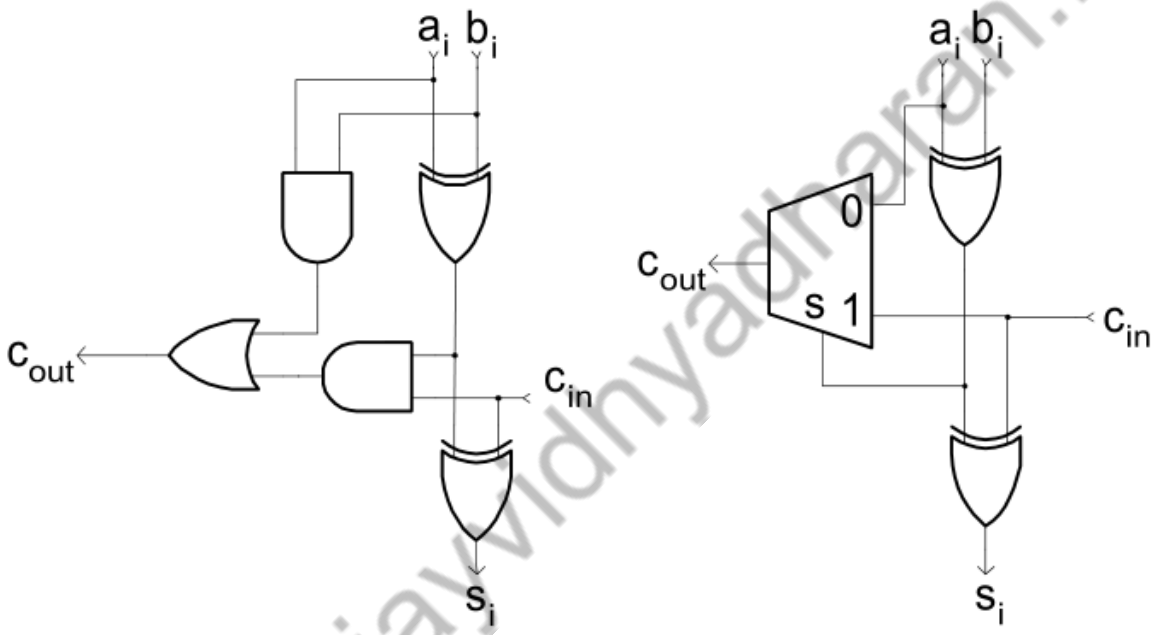
# CMOS 28T Mirror Adder




Carry Delay is 2 and Sum Gate Delay is 3

# Full Adder

Multiplexer based FA




$$s_i = a_i\bar{b}_i\bar{c}_i + \bar{a}_i b_i\bar{c}_i + \bar{a}_i\bar{b}_i c_i + a_i b_i c_i = a_i \oplus b_i \oplus c_i$$

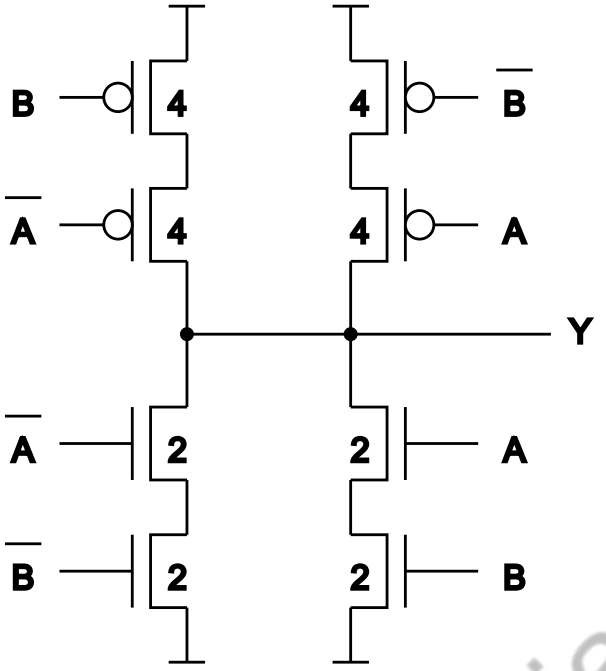
$$c_{i+1} = \bar{a}_i b_i c_i + a_i\bar{b}_i c_i + a_i b_i\bar{c}_i + a_i b_i c_i$$

Sum 2 Gate Delays

Carry 2 Gate Delays

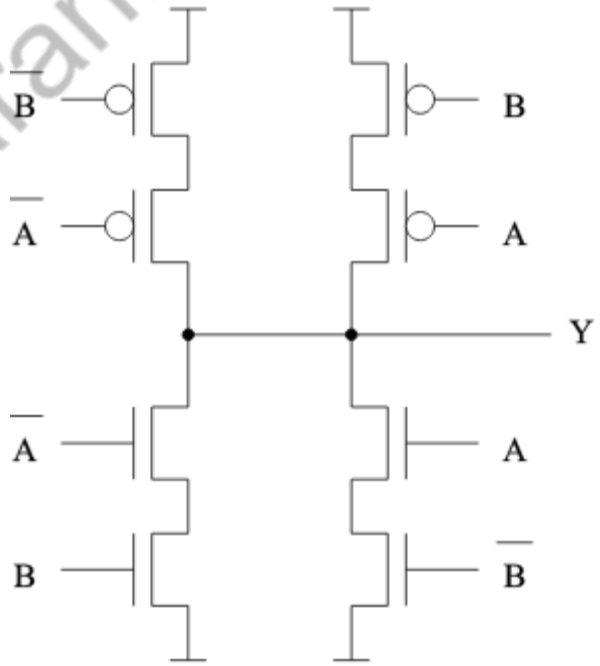
# CMOS XOR and XNOR

Matched 2-input XOR gate.

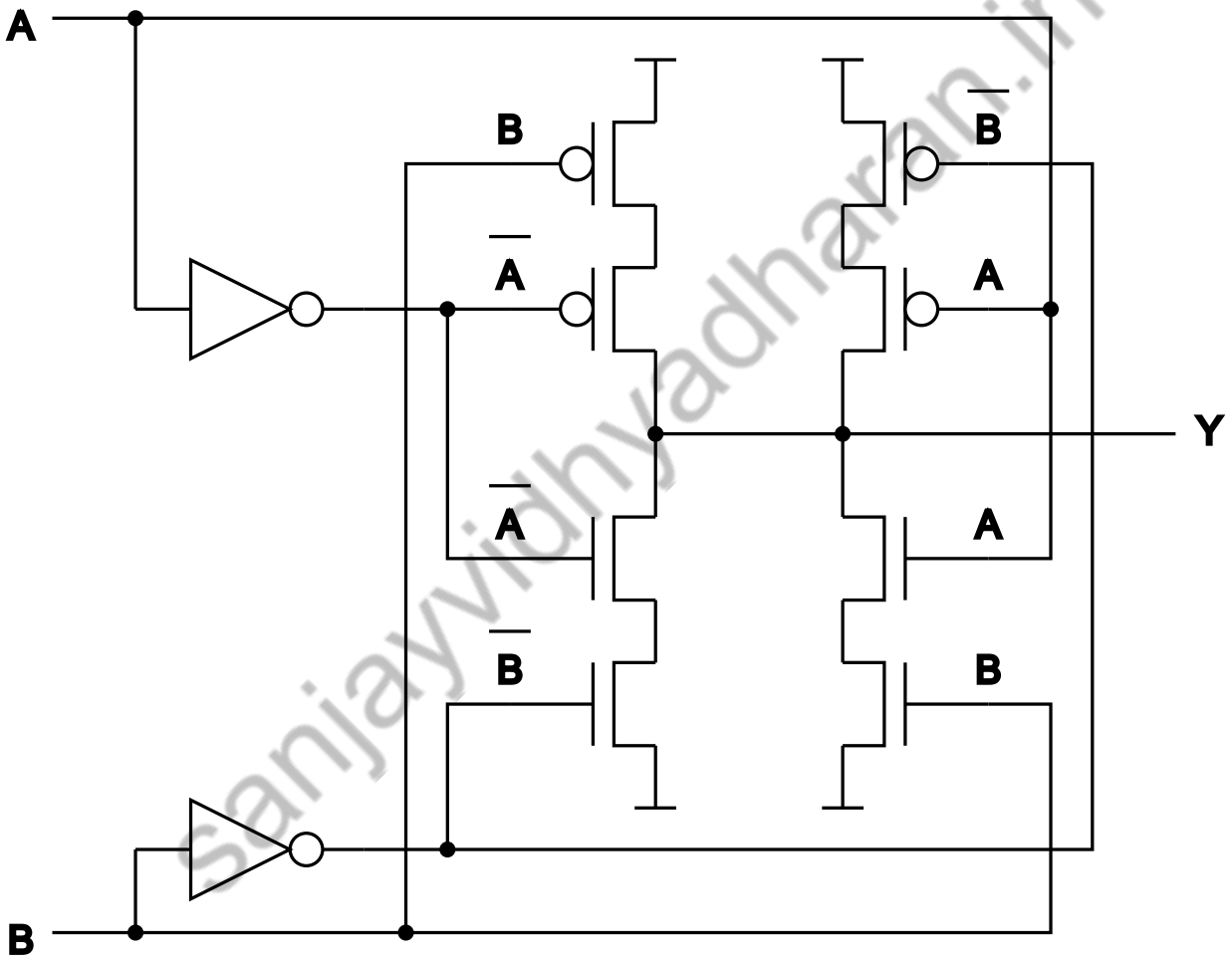



| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

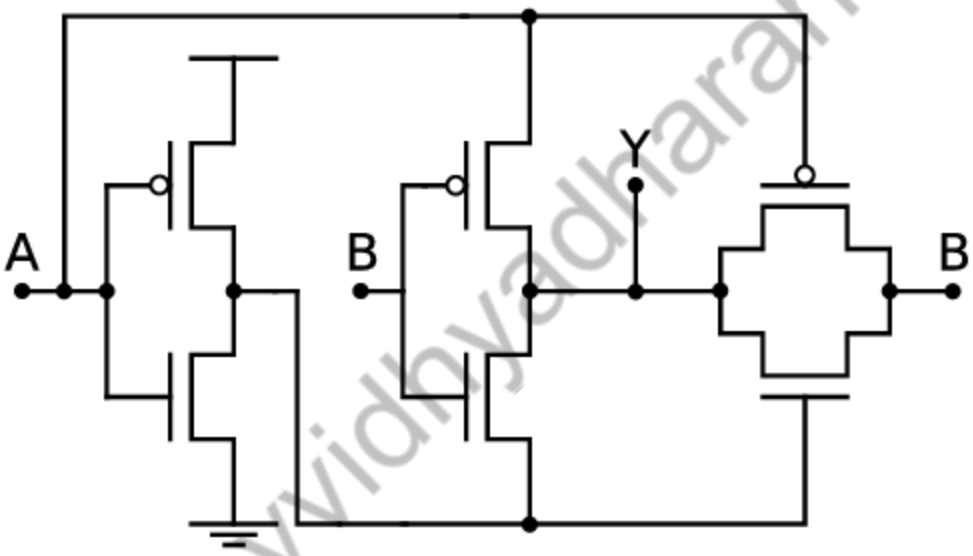
2-input XNOR gate.



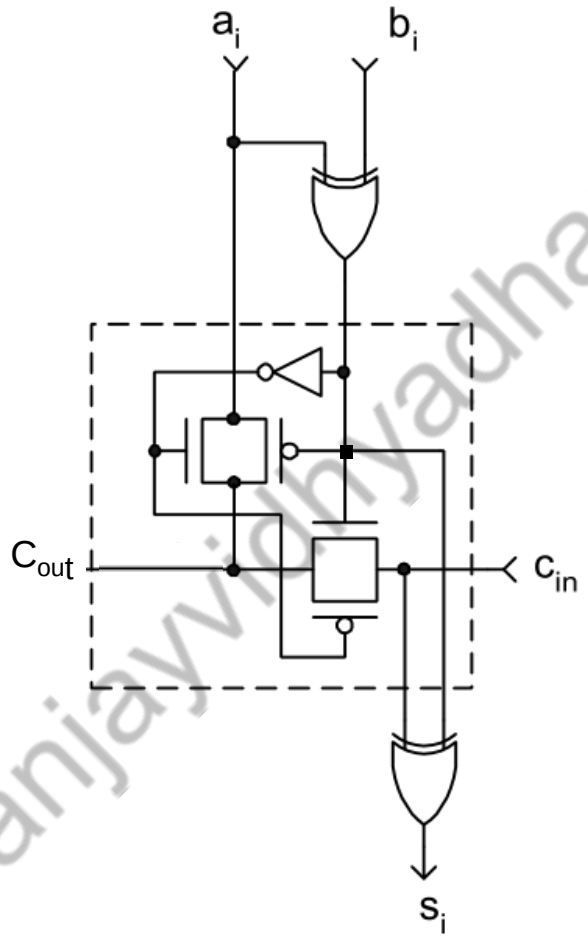
# CMOS XOR



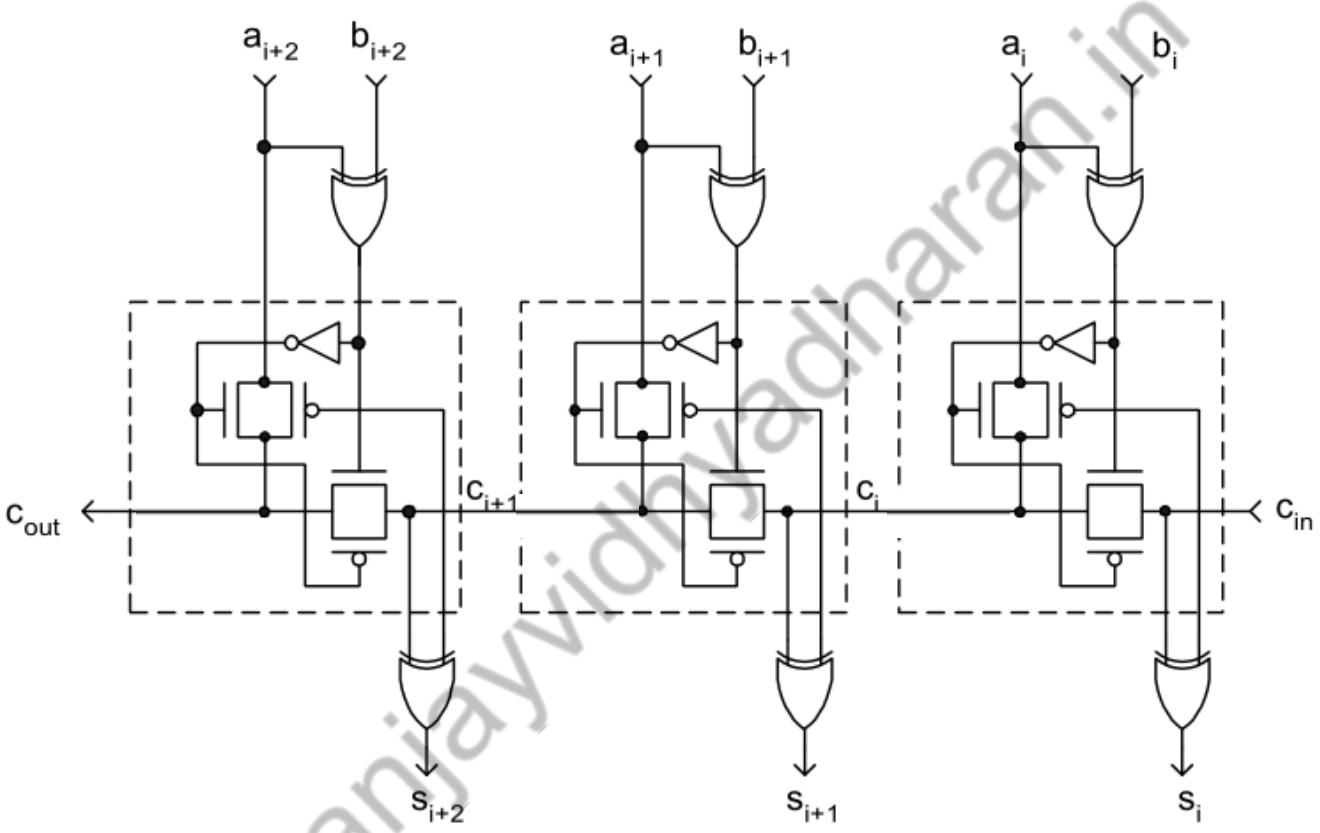
# Transmission Gate XOR



# Full Adder



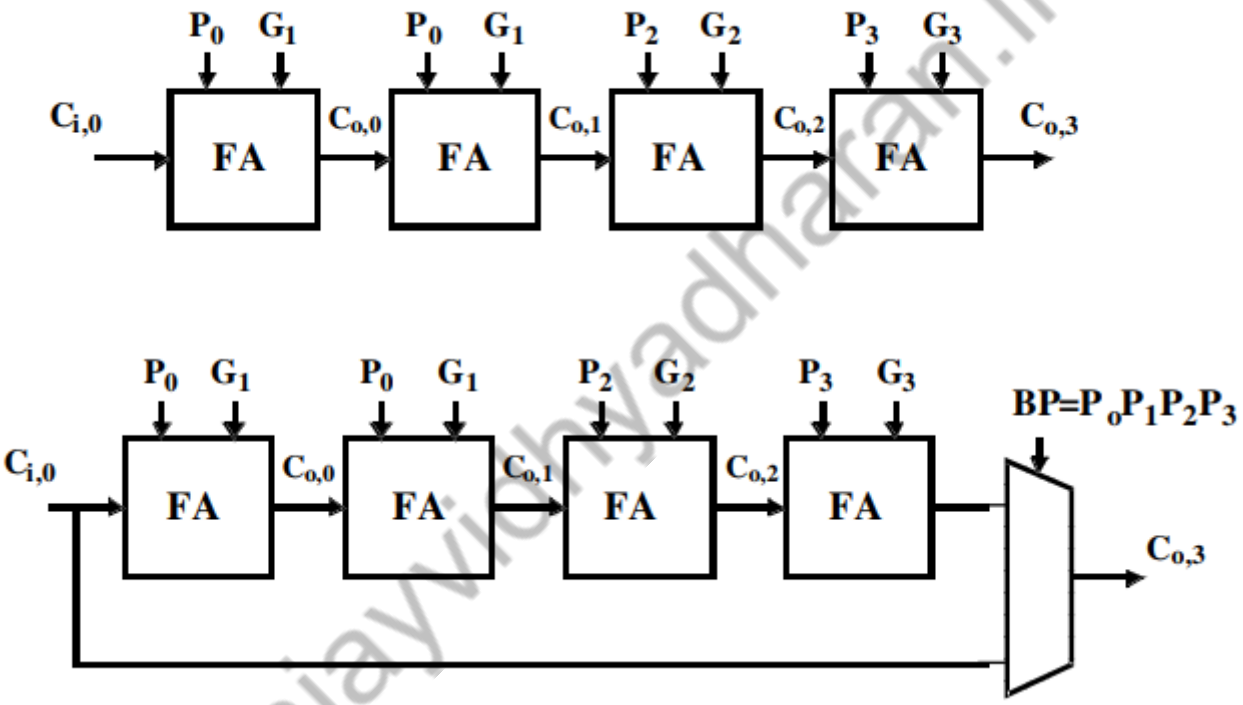
# Ripple Carry Adder



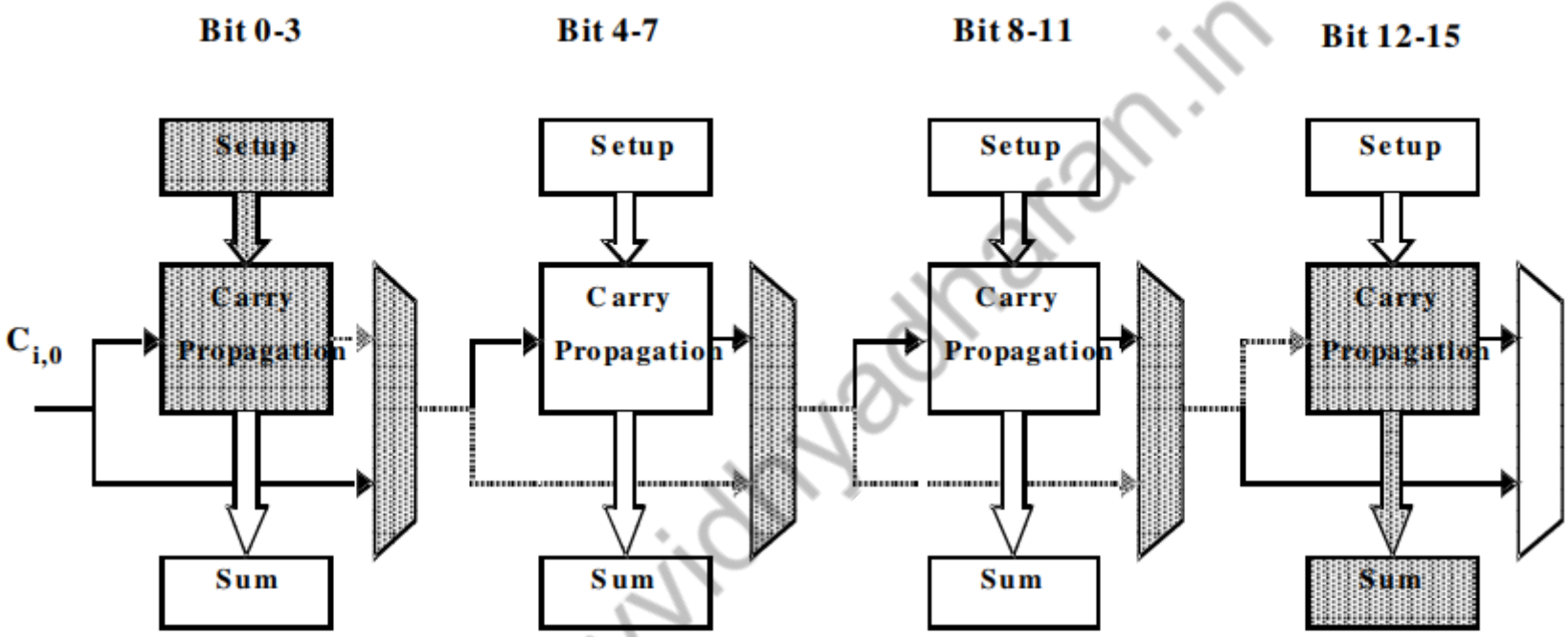



Delay of such an adder is 2N gate delays from $C_{in}$ to $C_{out}$

# Carry Bypass or Carry Skip Adder



# Carry Bypass or Carry Skip Adder





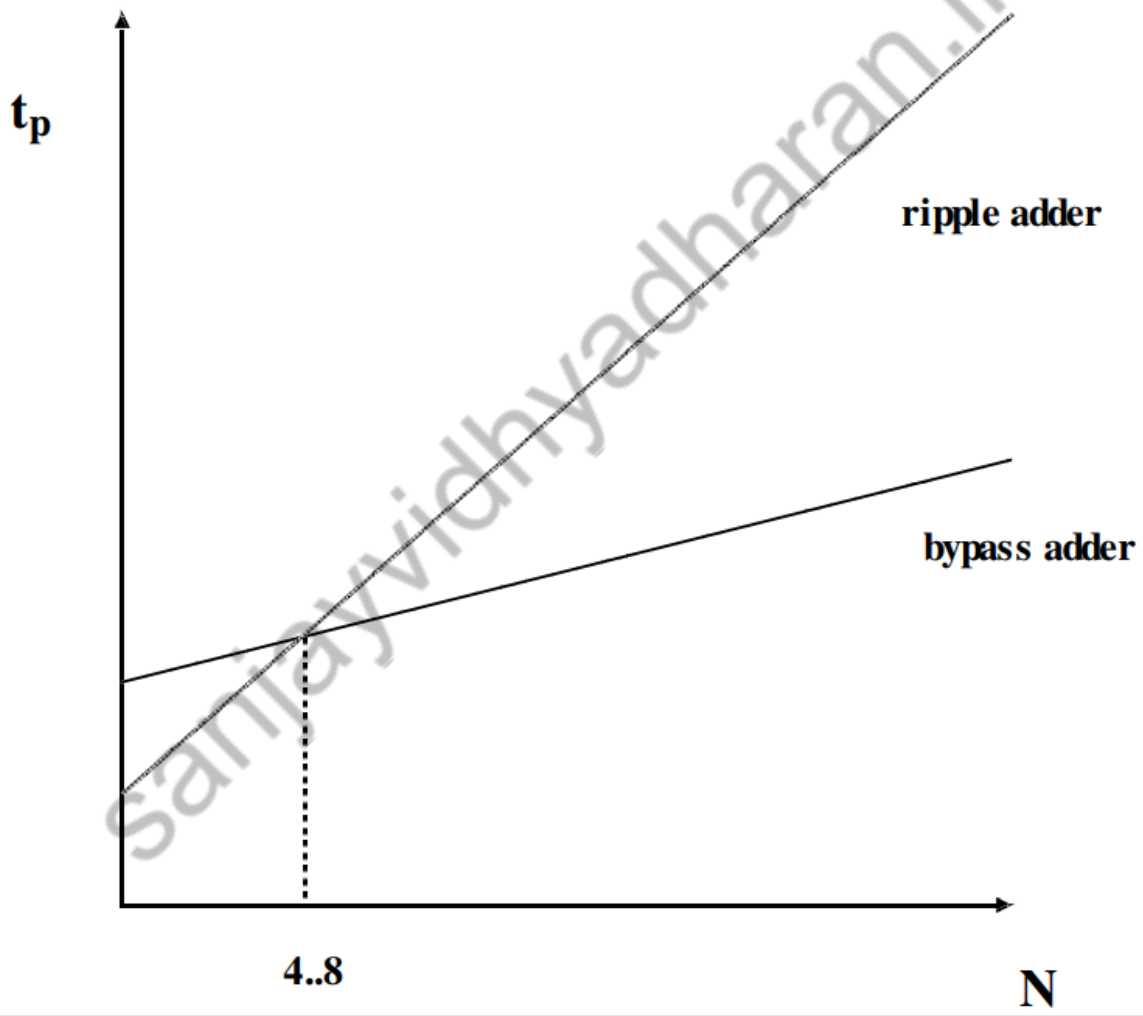

Design N-bit adder using N/M equal length stages

e.g. N = 16, M = 4

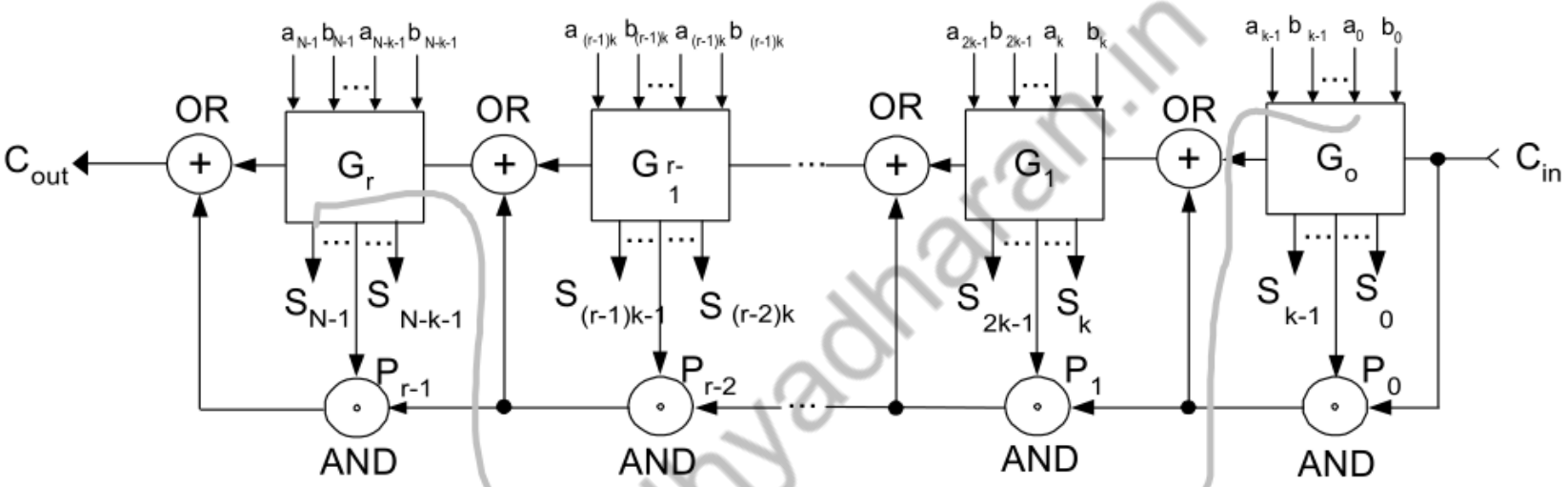
What is the critical path?

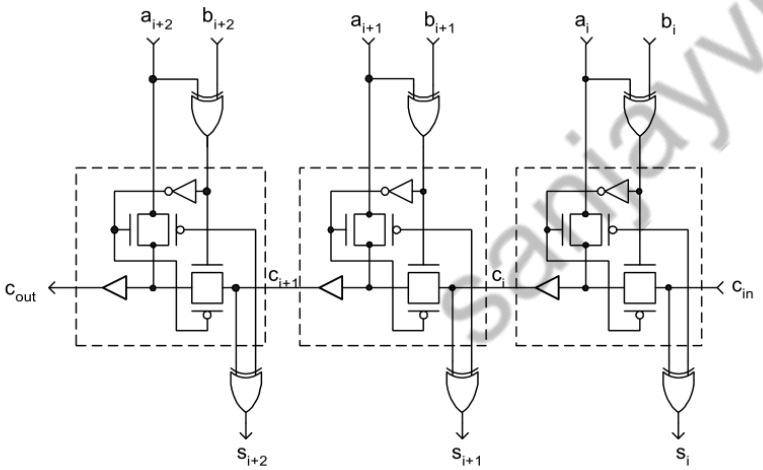
$$t_p = t_{setup} + Mt_{carry} + (N/M-1)t_{bypass} + Mt_{carry} + t_{sum}\text{ , i.e. O(N)}$$

# Carry Ripple versus Carry Bypass



# Carry Bypass or Carry Skip Adder



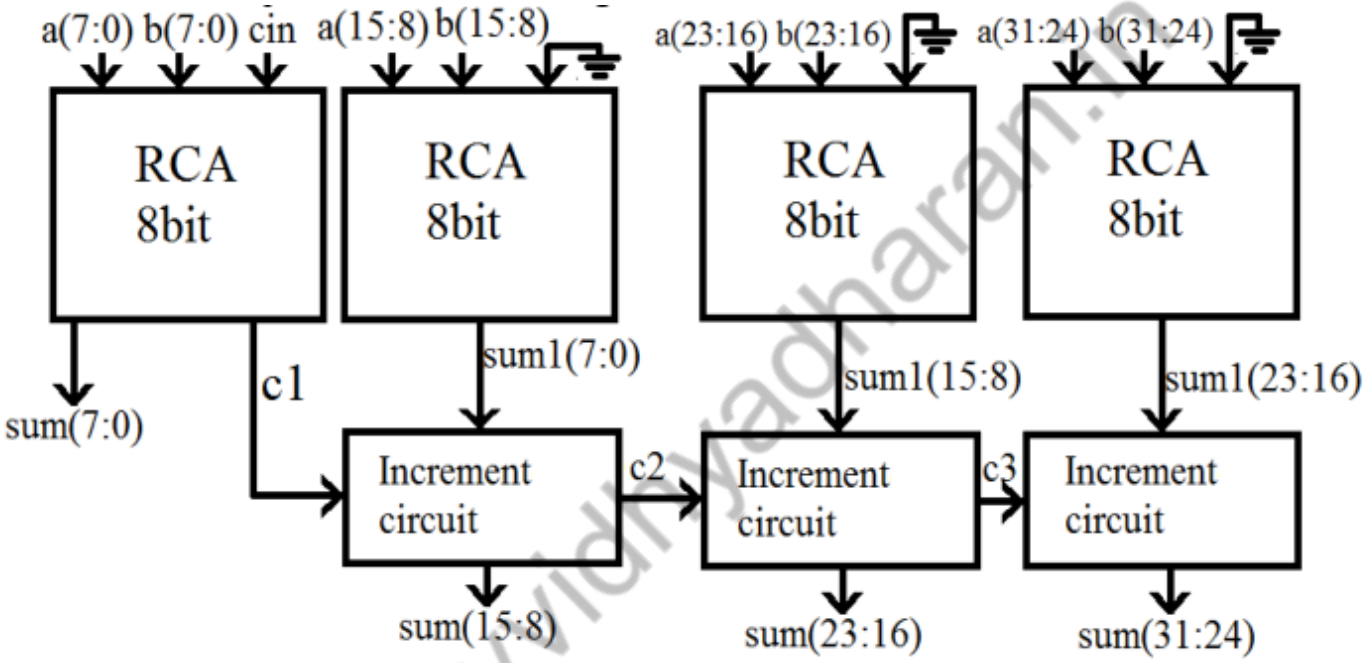




$$P = p_i \bullet p_{i+1} \bullet p_{i+2} \bullet \ldots\ldots \bullet p_{i+k}$$

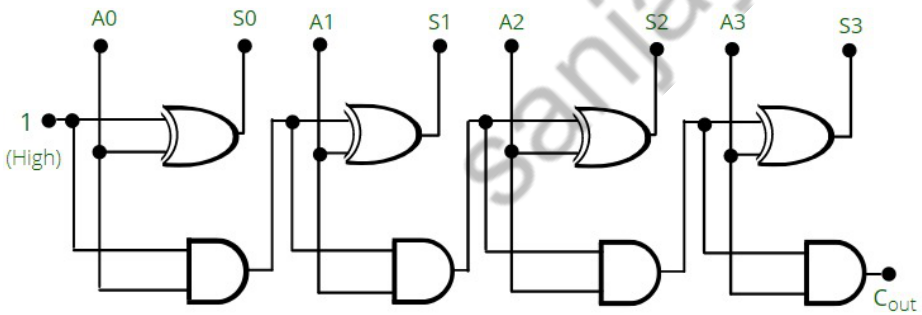
$$Delay = k * T_{Carry} + \left(\frac{N}{k} - 1\right) T_{Skip} + k * T_{Sum}$$

$$T_{Skip} = Genetaing\ P\ and\ one\ AND\ gate\ Delay$$

# Carry Increment Adder



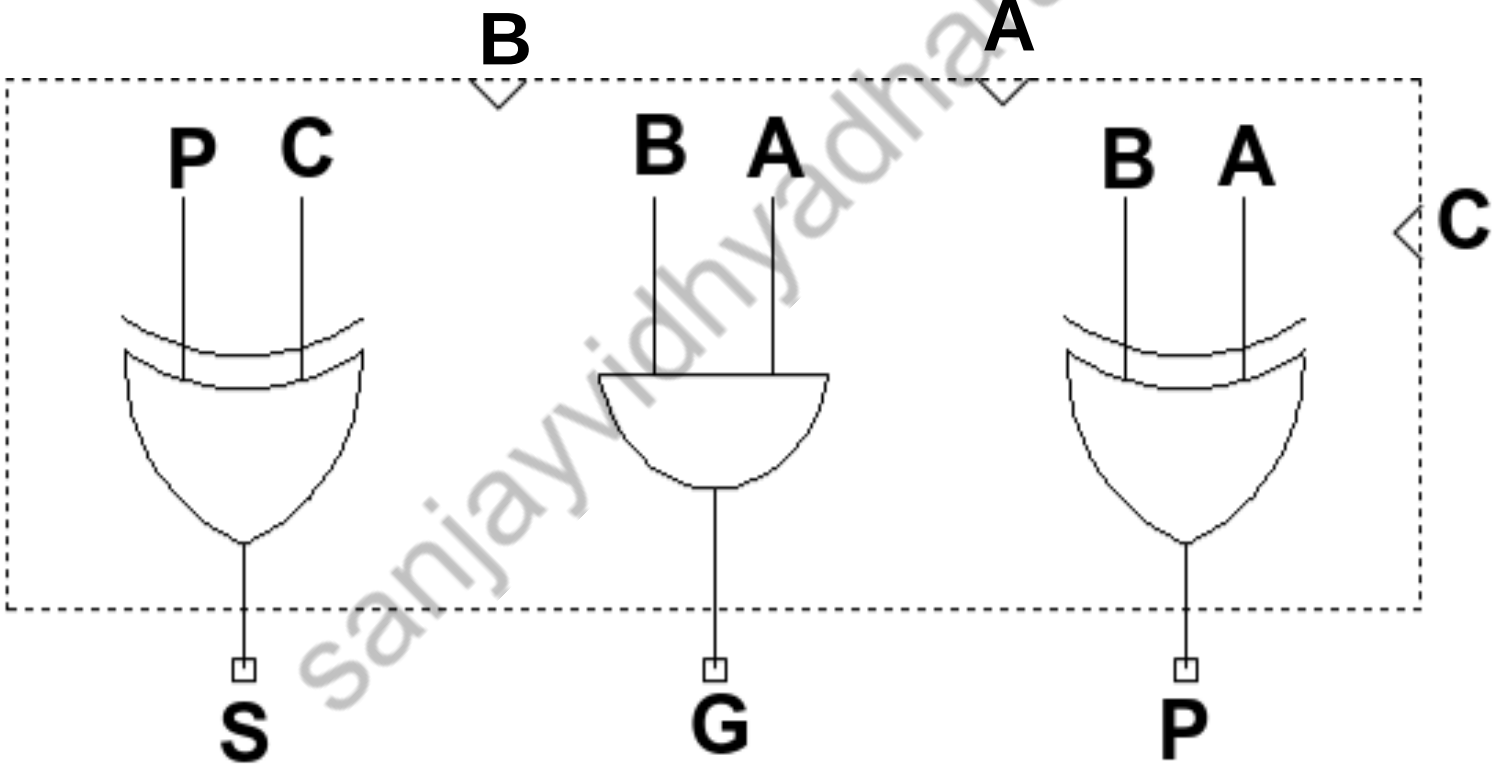

RCA - 64




CIA-16+8+8+8=40

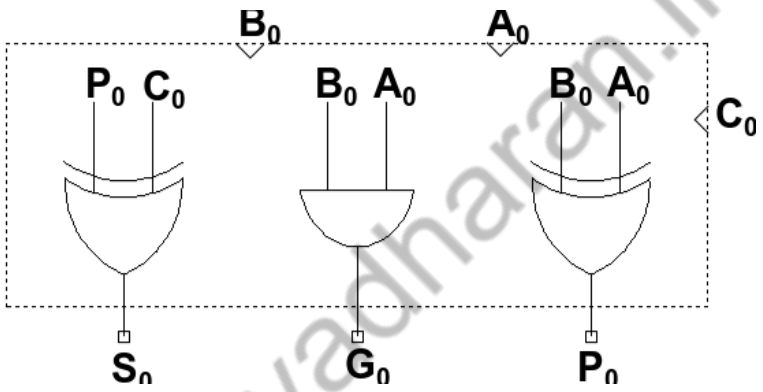
# Carry Look-Ahead Adder

1-bit CLA

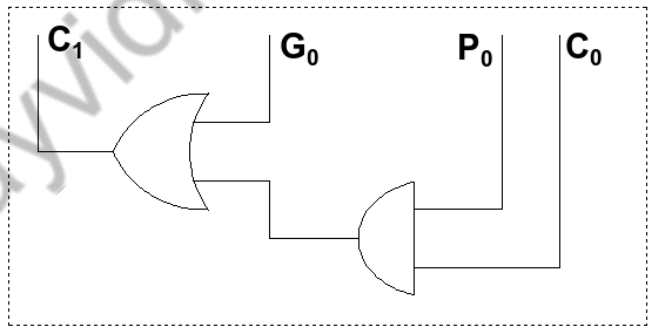


# Carry Look-Ahead Adder

CLA

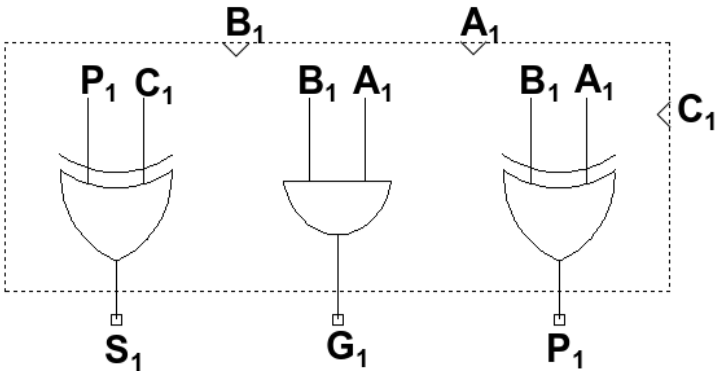


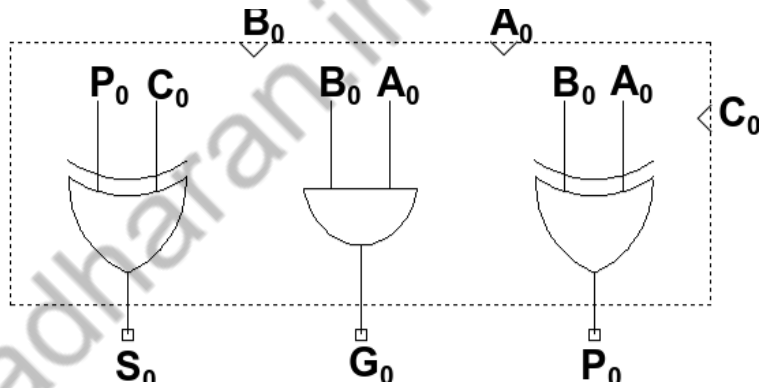

CLLB




$C_1=G_0+P_0C_0$

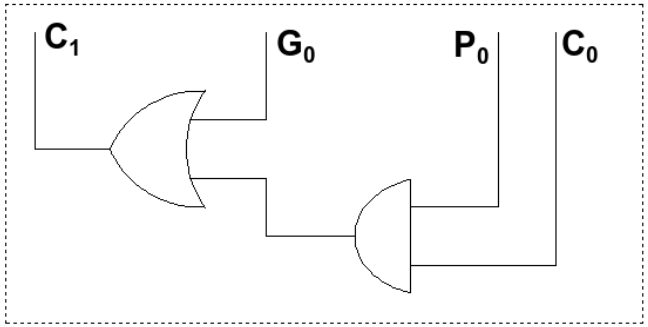
# Carry Look-Ahead Adder



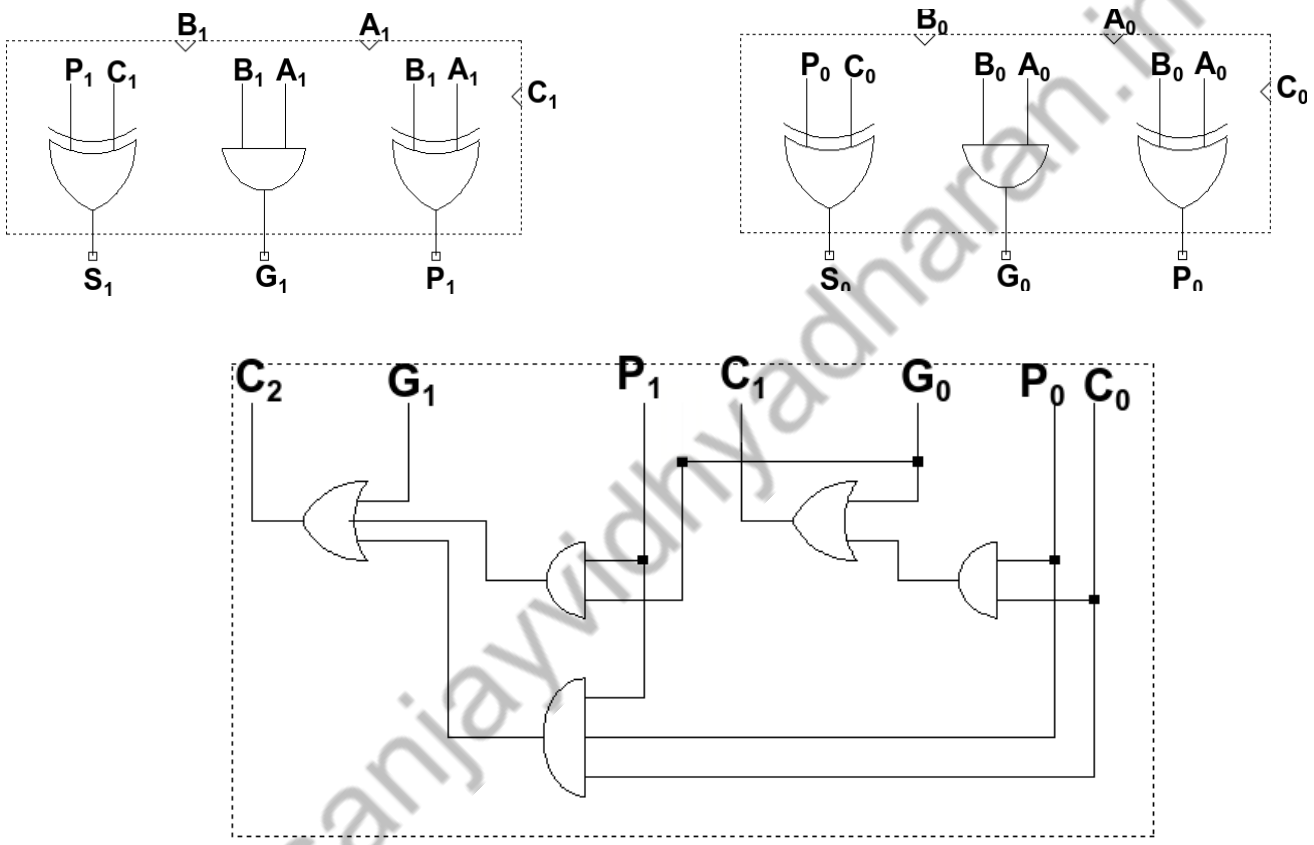




$$C_2 = G_1 + P_1 C_1$$

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



# 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

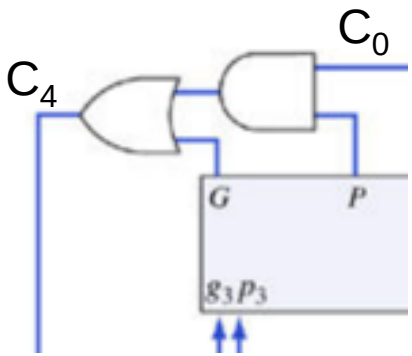
$$C_1=G_0+P_0C_0$$

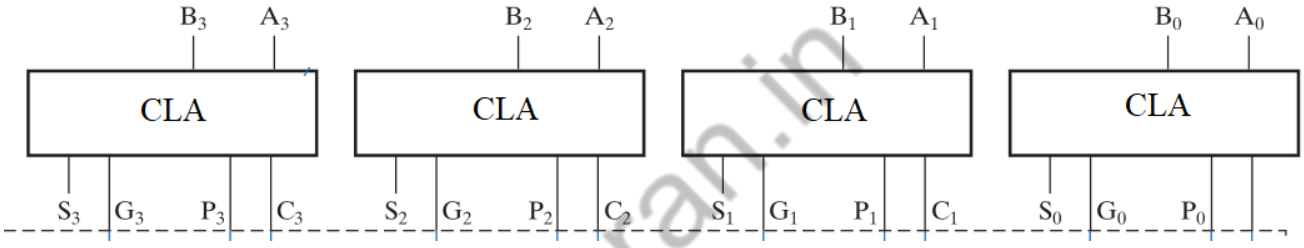
$$C_2= G_1+P_1G_0+P_1P_0C_0$$

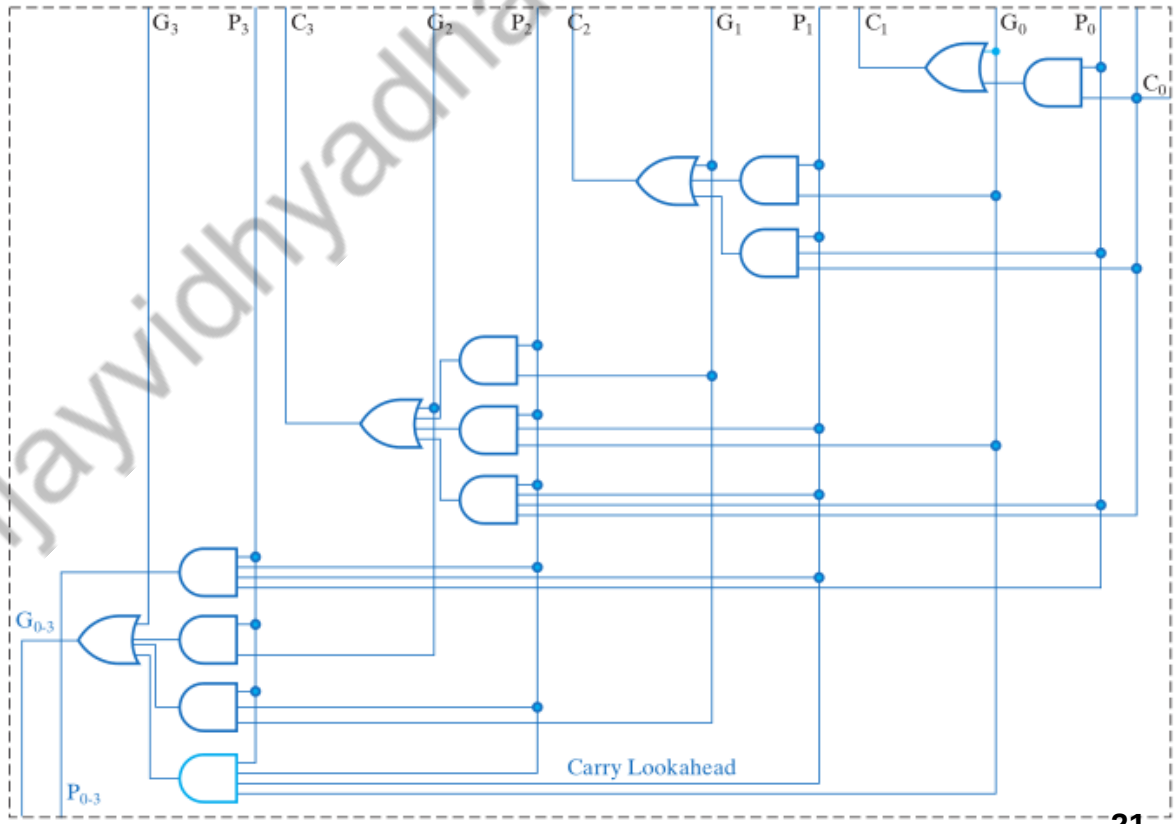
$$C_3=G_2+P_2G_1+ P_2P_1G_0 +P_2P_1P_0C_0$$

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

# Carry Look-Ahead Adder



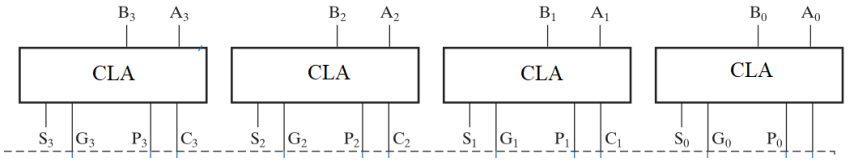



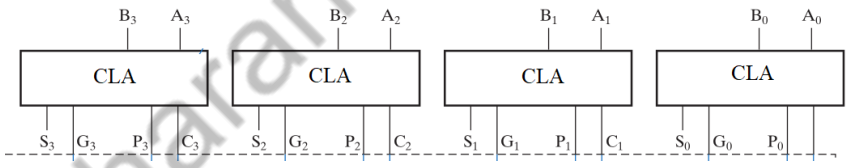

3 Gate Delay for $G_3$
2 Gate Delay for $P_3$

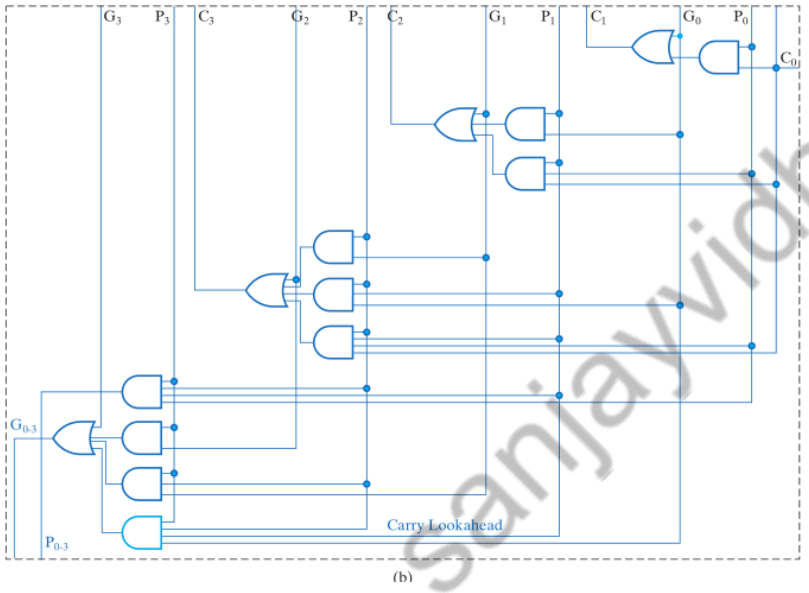
4 Gate Delay for $C_4$
4 Gate Delay for $S_3$

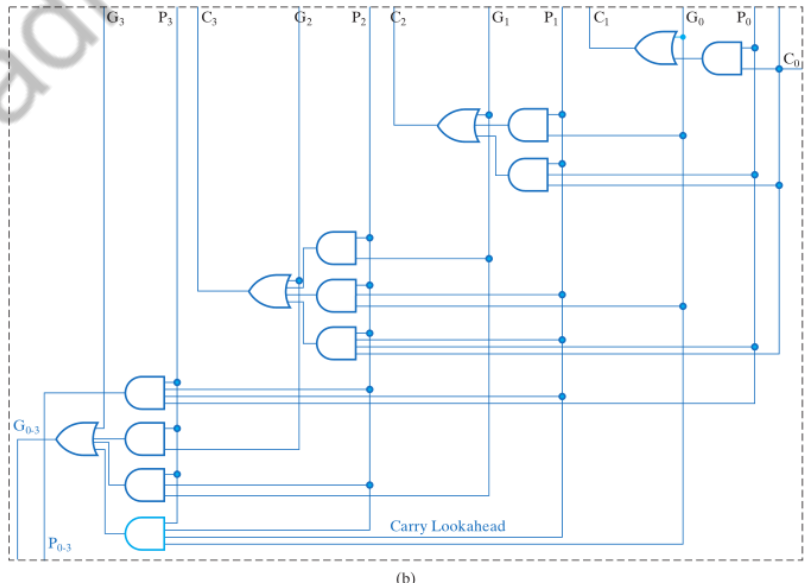
# Carry Look-Ahead Adder

## 8 Bit Full Adder



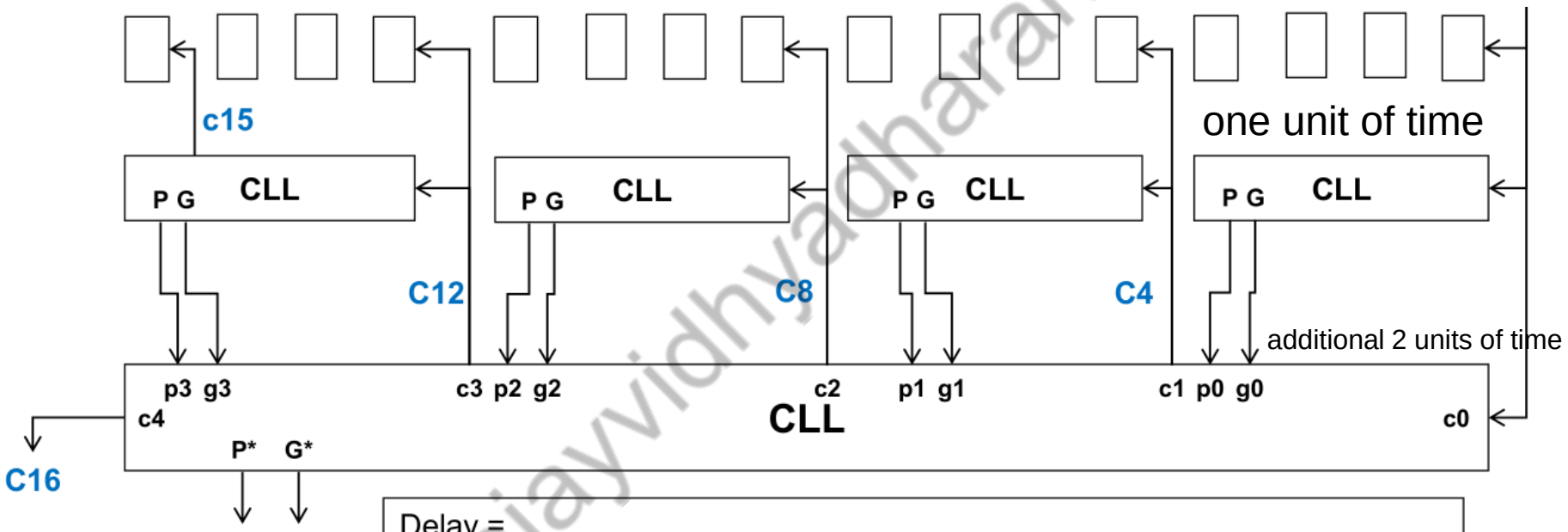



Delay = 4 + 4 = 8 Gate Delay for $C_7$

= 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, 6 Delays for C16
= 7 = Delay in producing c15
= 8 = Delay in producing S15

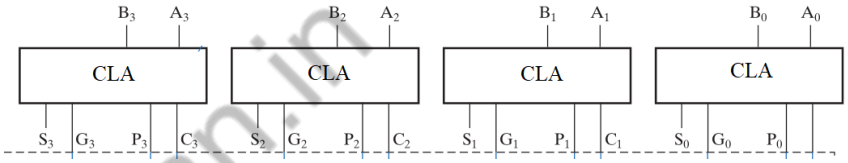
# Carry Look-Ahead Adder

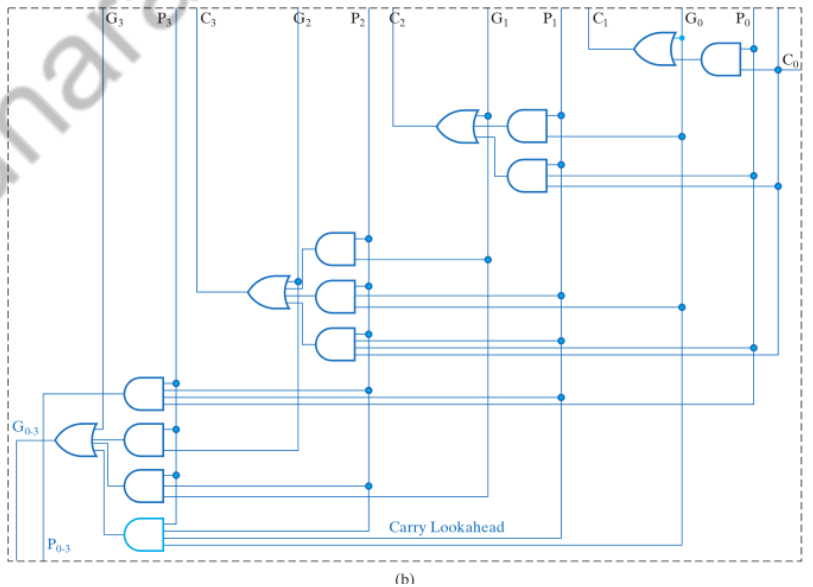
## Delay of a k-bit Carry-Lookahead Adder

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

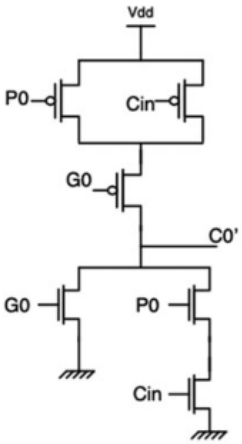
| k | $T_{\text{lookahead-adder}}$ | $T_{\text{ripple-carry-adder}}$ |
|---|---|---|
| 4 | 4 | 8 |
| 16 | 8 | 32 |
| 32 | 12 | 64 |
| 64 | 12 | 128 |
| 128 | 16 | 256 |
| 256 | 16 | 512 |

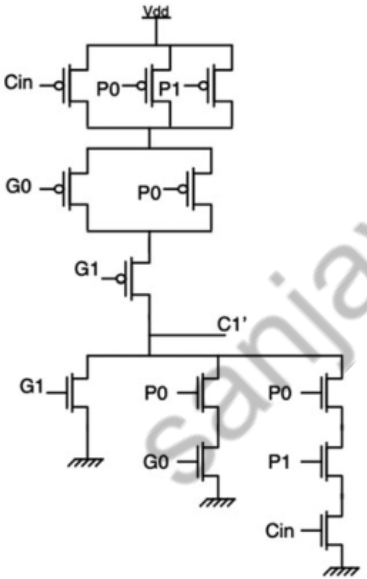
# Carry Look-Ahead Adder

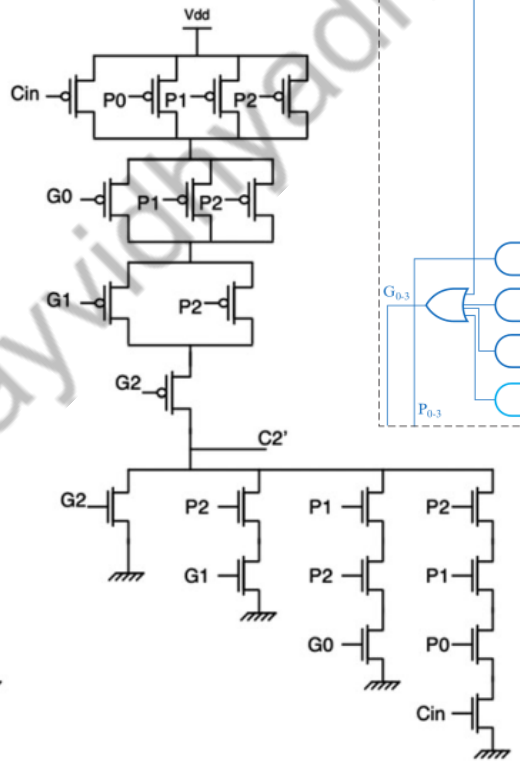


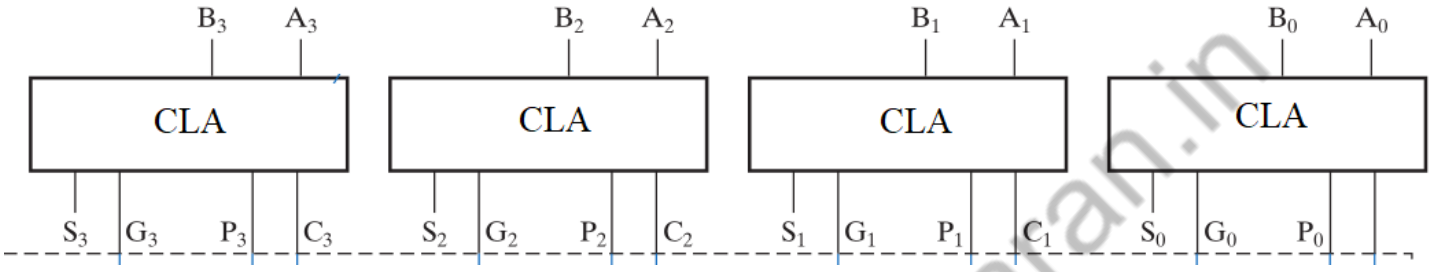

(b)

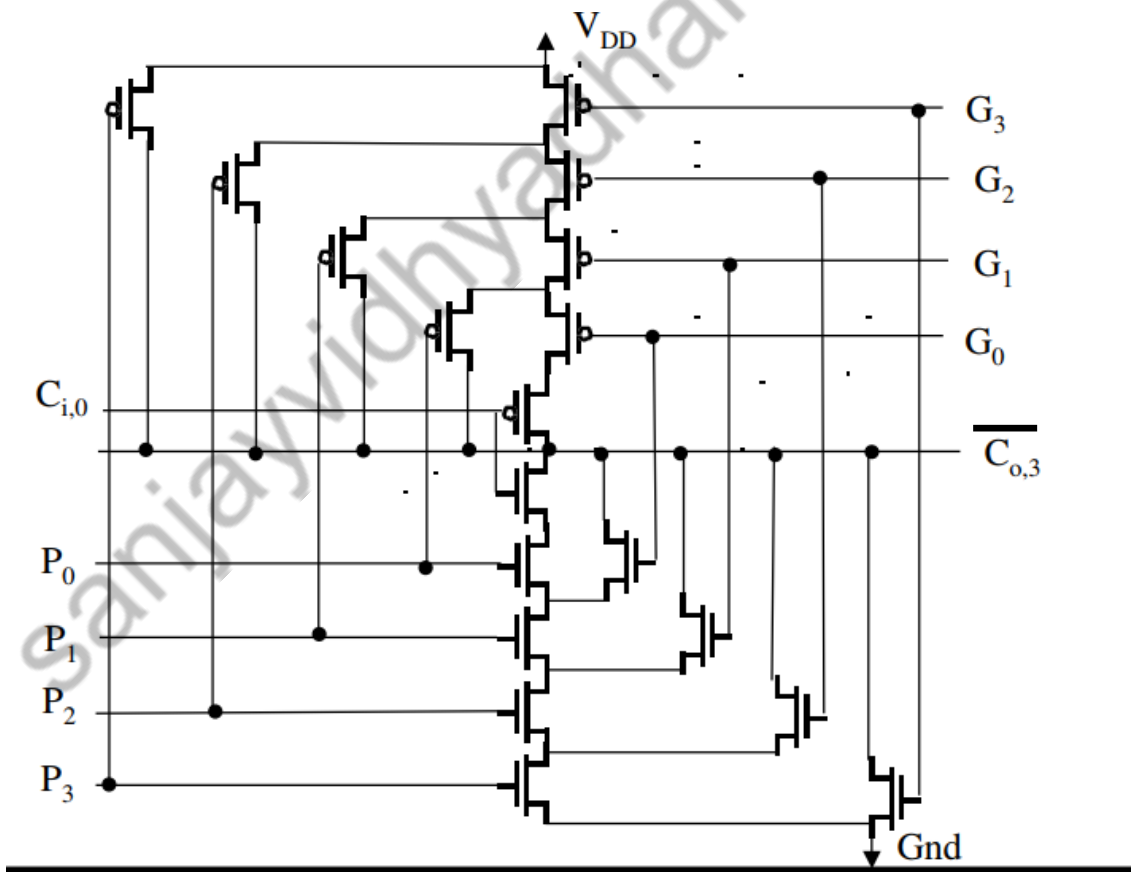


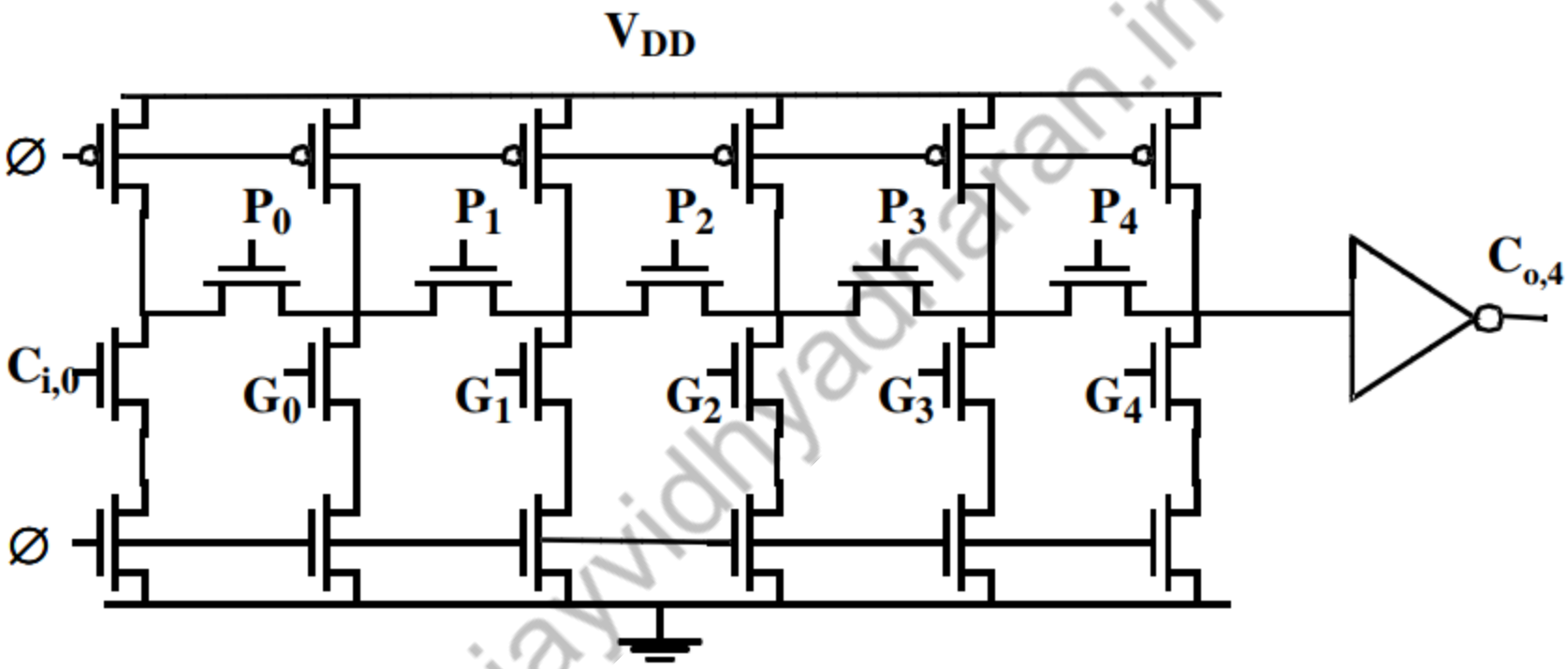



# Carry Look-Ahead Adder

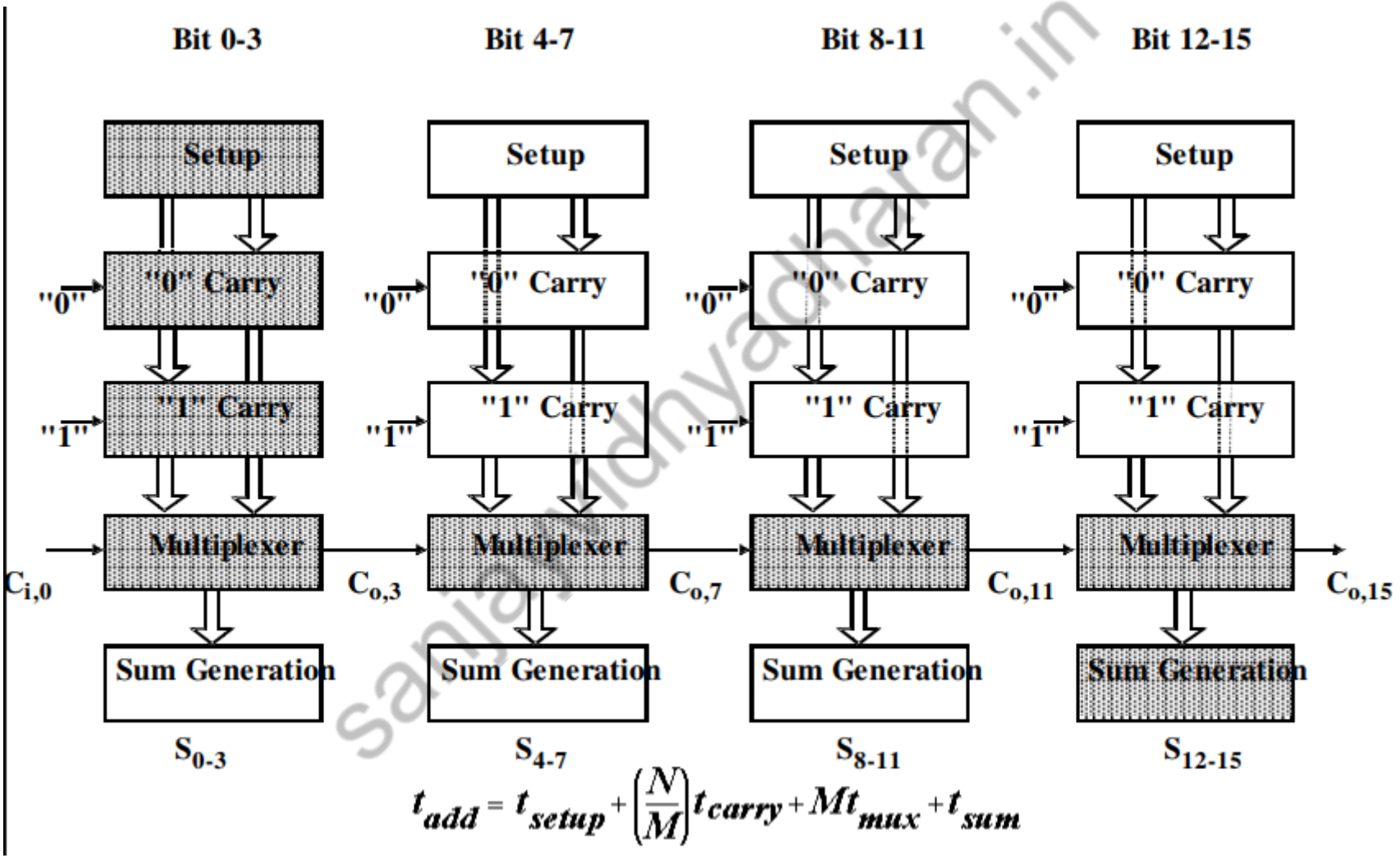


# Manchester Carry Chain

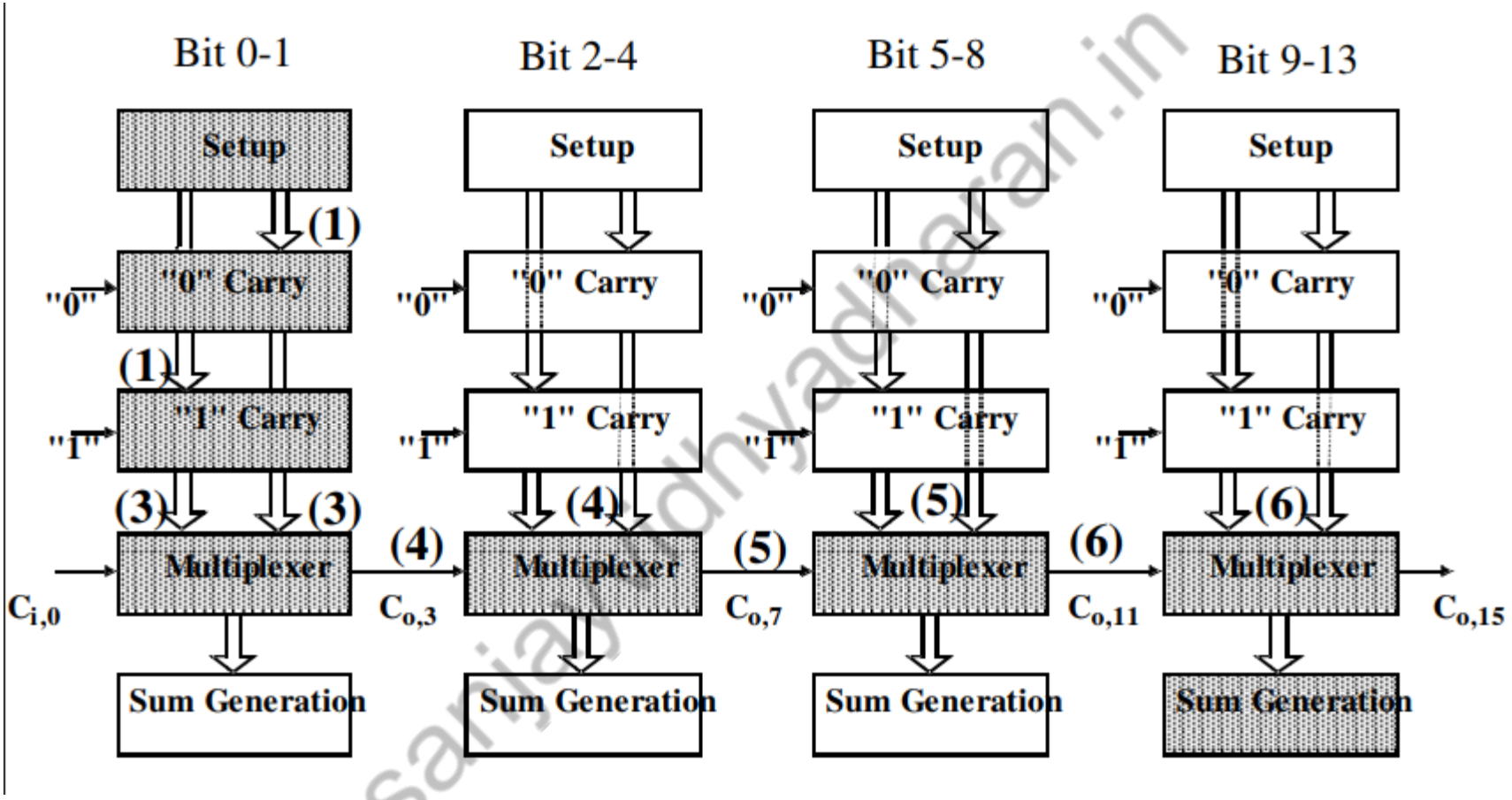


# Linear Carry-Select Adder




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

# 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)

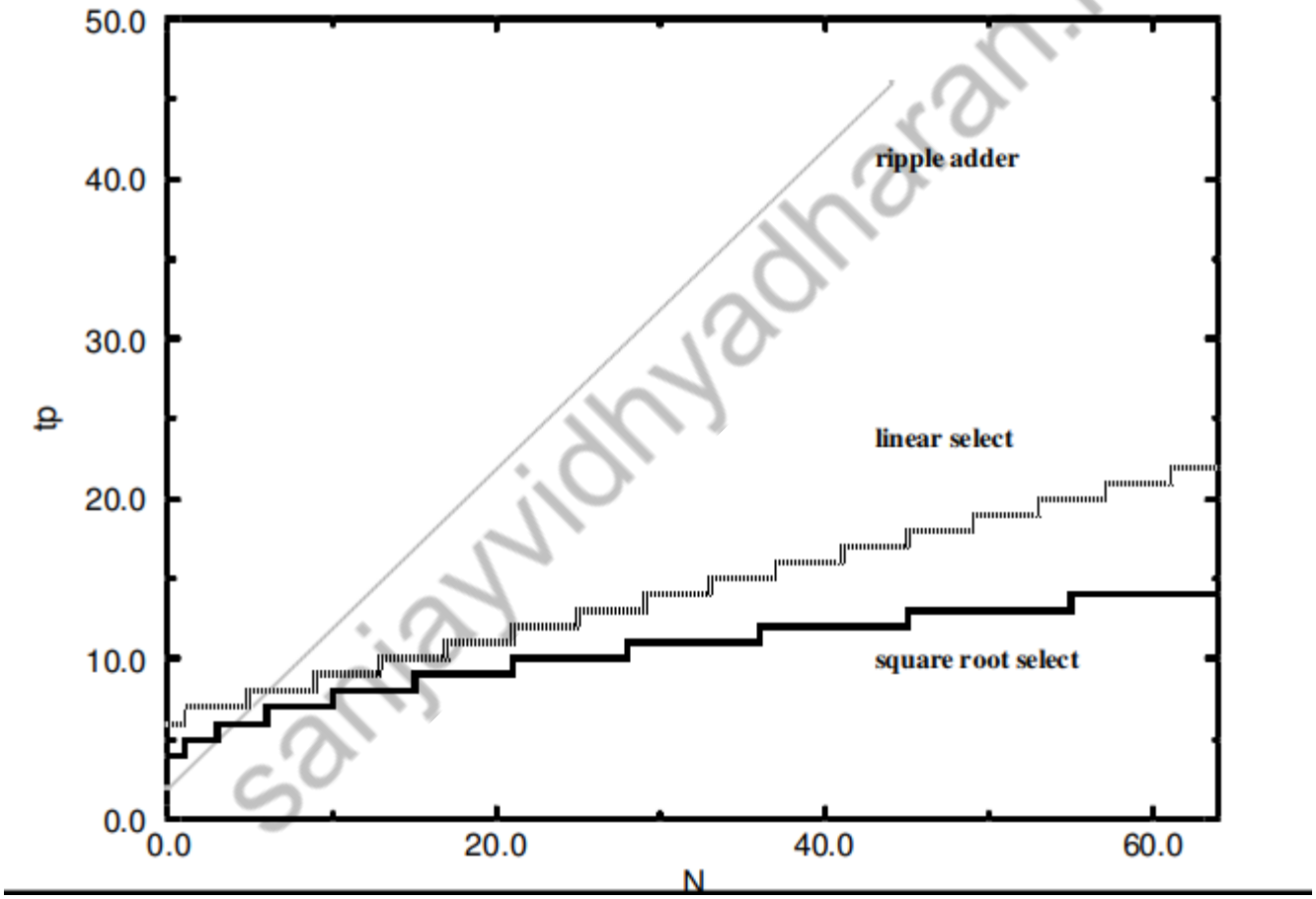
$$N = MP + \frac{P(P-1)}{2}$$

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

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

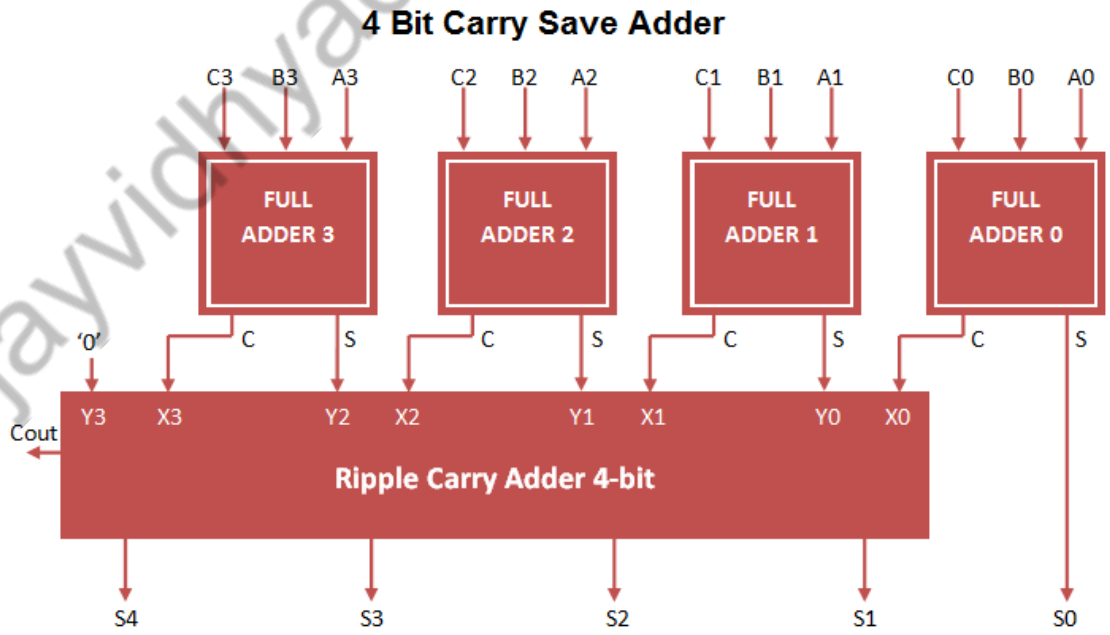
$$N \approx \frac{P^2}{2}$$

# Adder Delays - Comparison

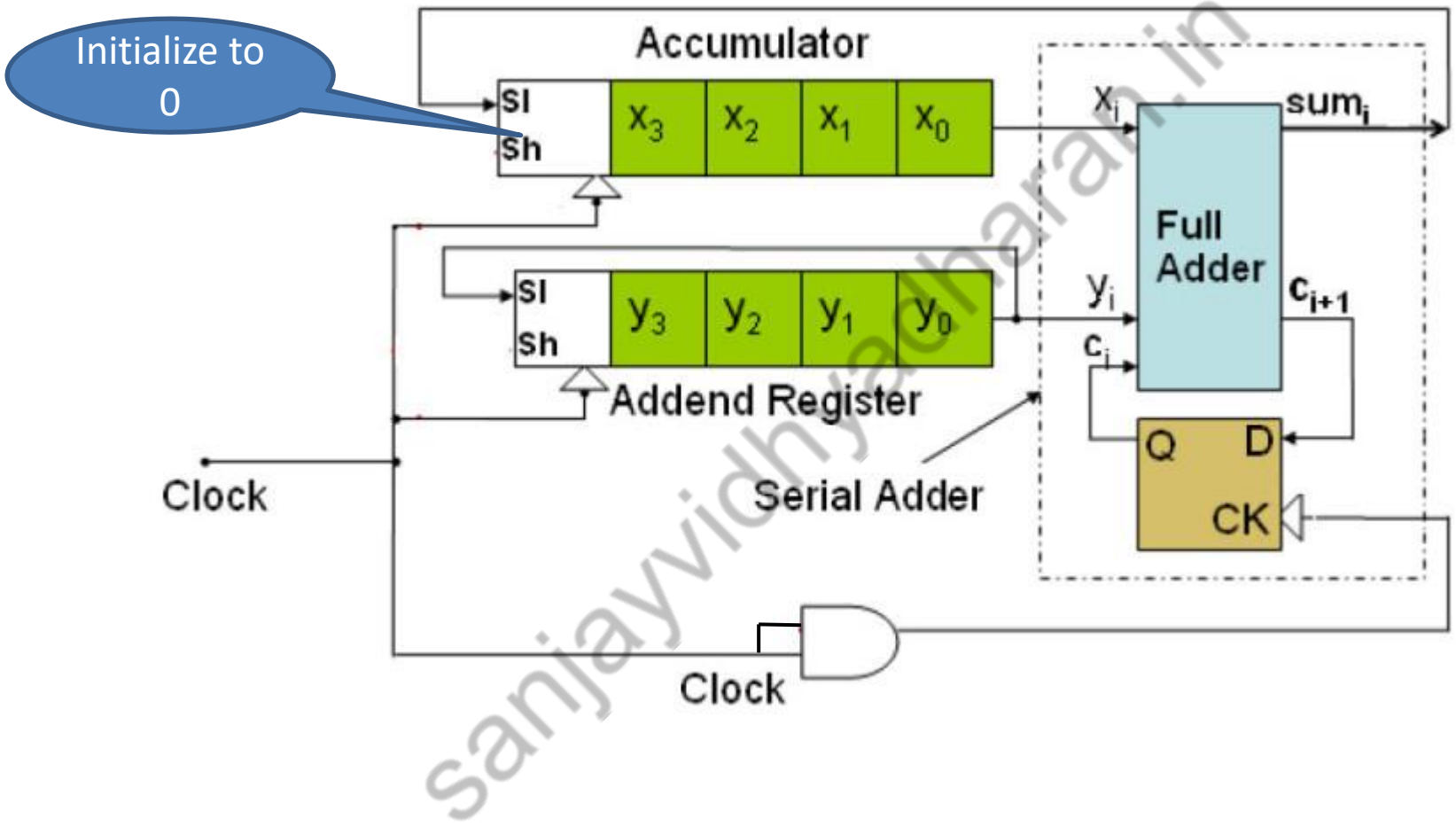


# Carry Save Adder

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

# Parallel Prefix Adder

Dot Operator

$$(G_0, P_0) \circ (G_1, P_1) = (G_{0:1}, P_{0:1})$$

$$G_{0:1} = G_1 + P_1 G_0$$

$$P_{0:1} = P_0 P_1$$

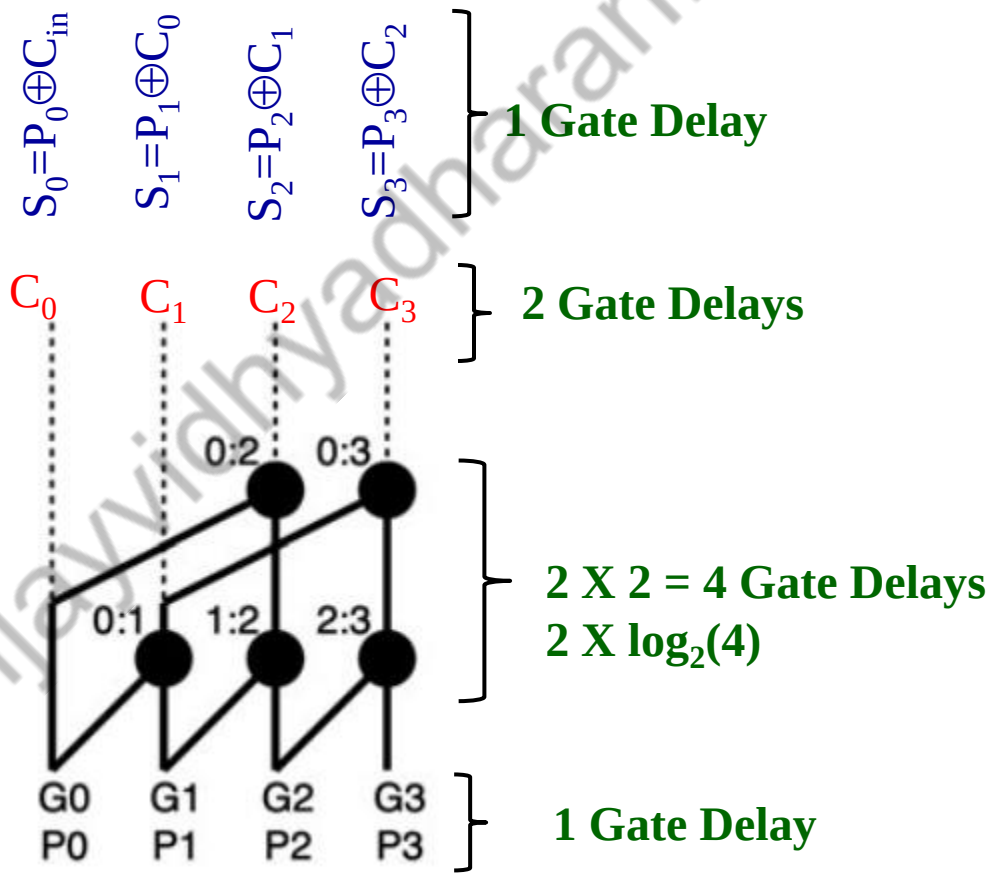
$$(G_0, P_0) \circ (G_1, P_1) = (G_1 + P_1 G_0, P_0 P_1)$$

$$C_1 = G_1 + P_1 G_0 + P_0 P_1 C_{\text{in}}$$

$$(G_{1:3}, P_{1:3}) \circ (G_{2:5}, P_{2:5}) = (G_{1:5}, P_{1:5})$$

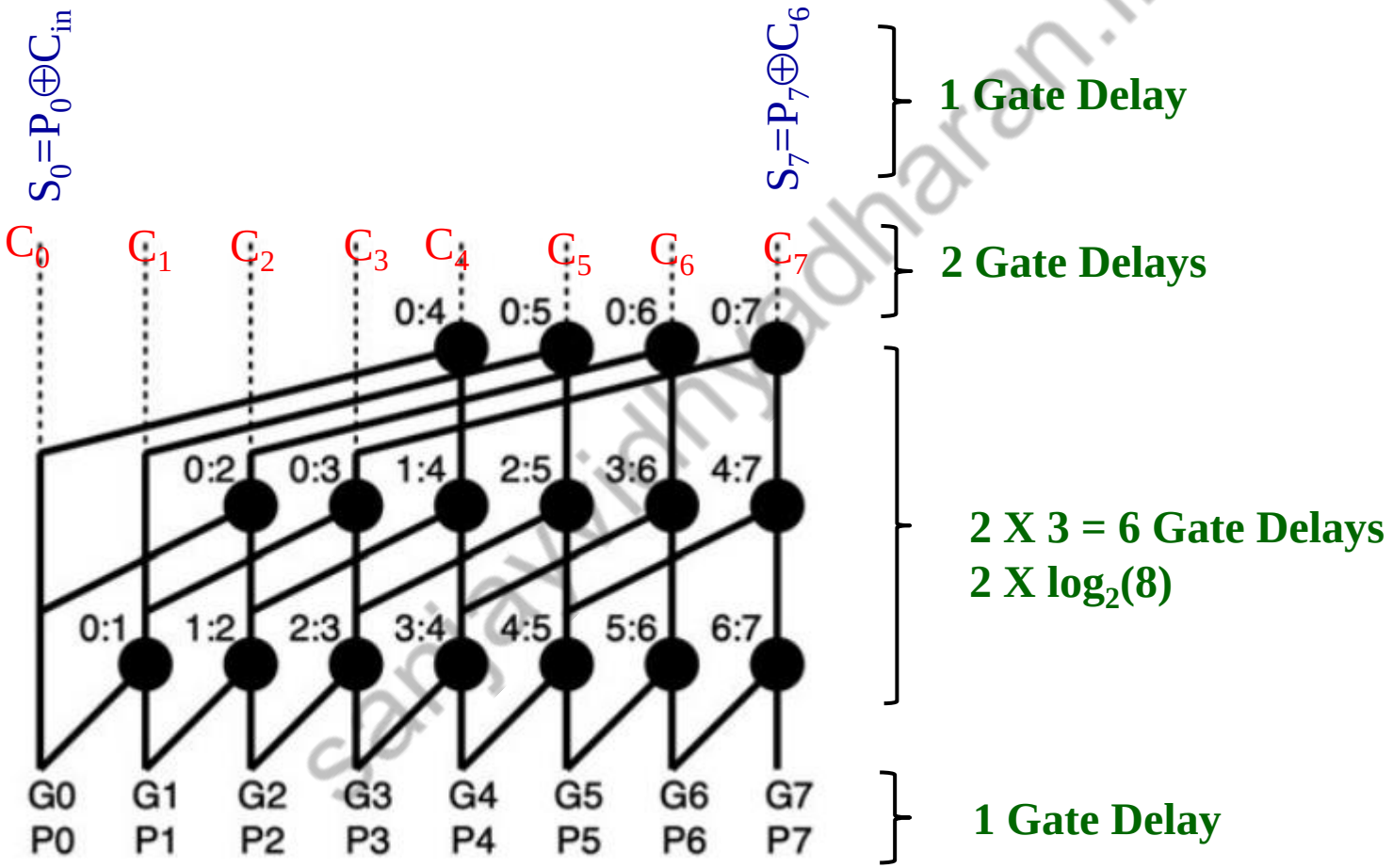
# Parallel Prefix Adder

Kogge–Stone



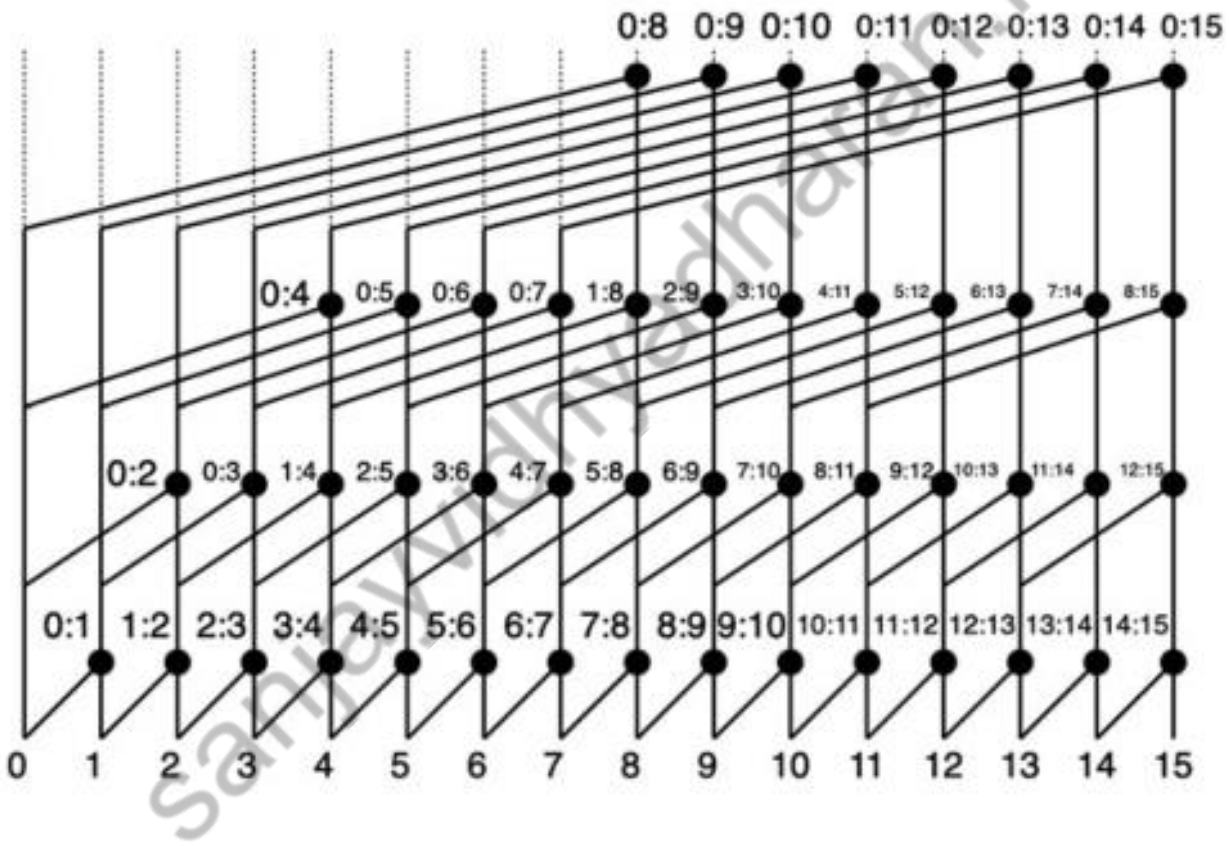
# Parallel Prefix Adder

Kogge–Stone



# Parallel Prefix Adder

Kogge–Stone

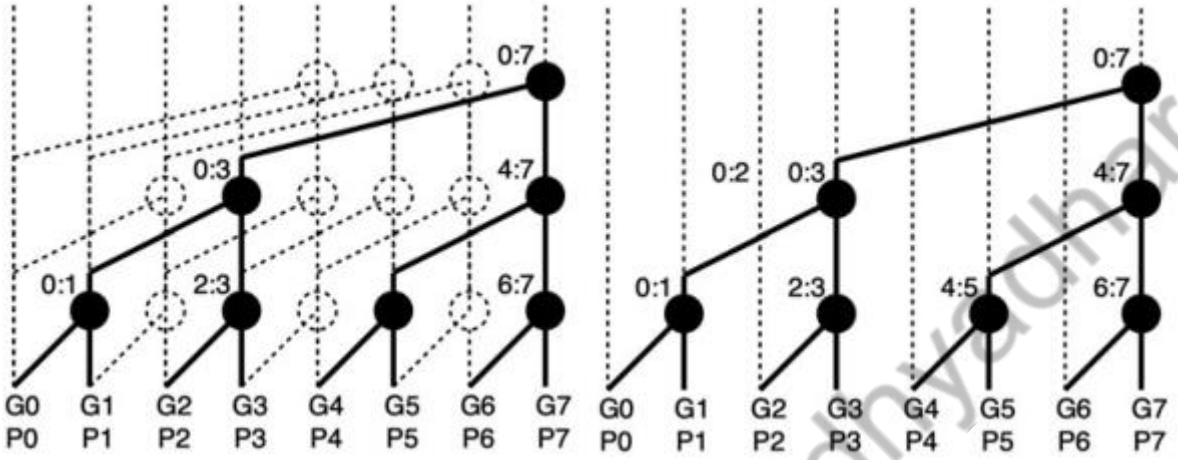





Routing is Complex
Large Fanout

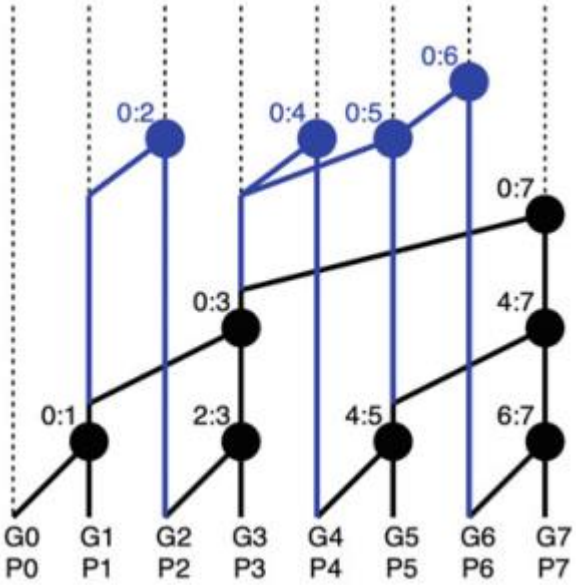
# Parallel Prefix Adder

Kogge–Stone




8-bit KS adder calculating only $C_7$

8-bit Brent–Kung adder



# Thank you