# VLSI SYSTEMS AND ARCHITECTURE

## 2021-22

## Lecture 3

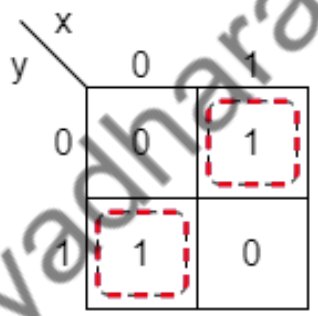
## Computer Arithmetic Algorithms and Implementations

**By Dr. Sanjay Vidhyadharan**

# Half Adder

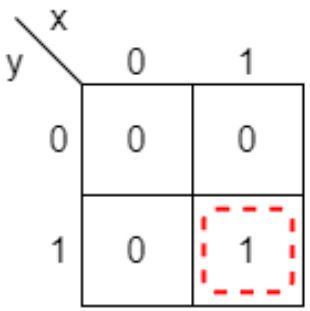
| Inputs | | Output | |
|---|---|---|---|
| A | B | Carry | Sum |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

K-map for Sum

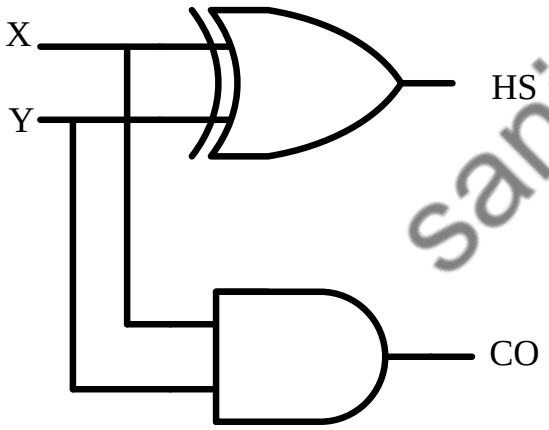



S = xy' + x'y

K-map for Carry




C = xy




Sum = X•Y' + X'•Y = X ⊕ Y
Carry = X•Y

Time Delay for the Half Adder?

1 gate delay
A gate delay of an xor for the half sum
A gate delay of an AND gate for the carry out

# Full Adder

**Full-adder can also implemented with two half adders and one OR gate.**

$S = A \oplus B \oplus C_{in}$

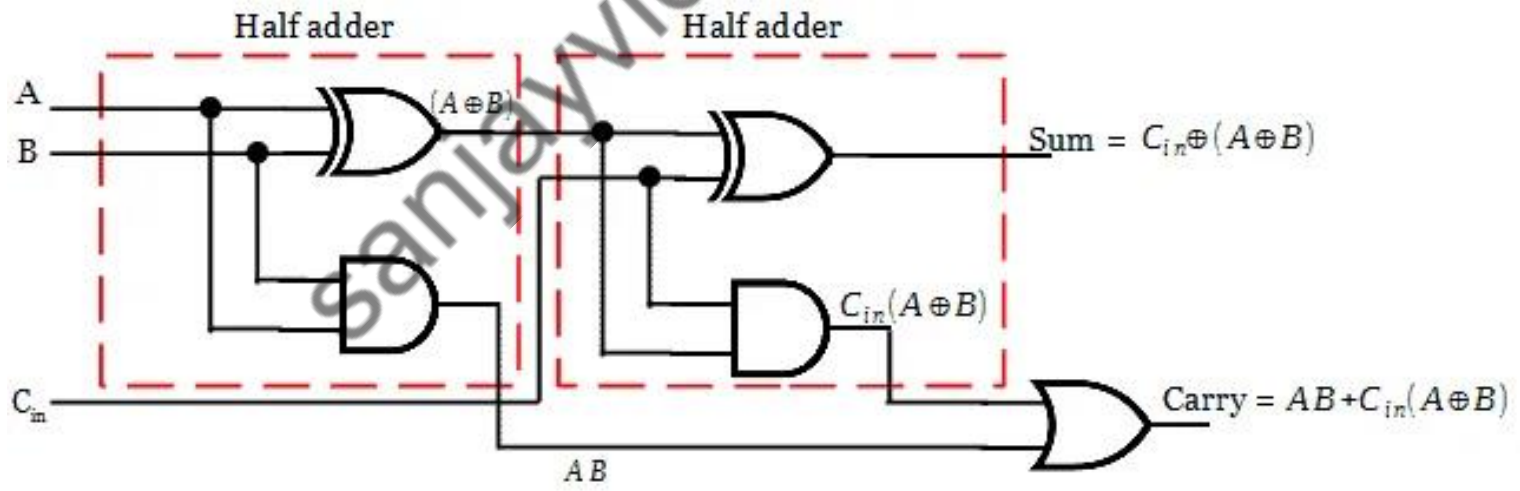
$C_{out} = AB + BC_{in} + AC_{in}$

$= AB + C_{in}(A+B)$

$= AB + C_{in}(A \oplus B + AB)$

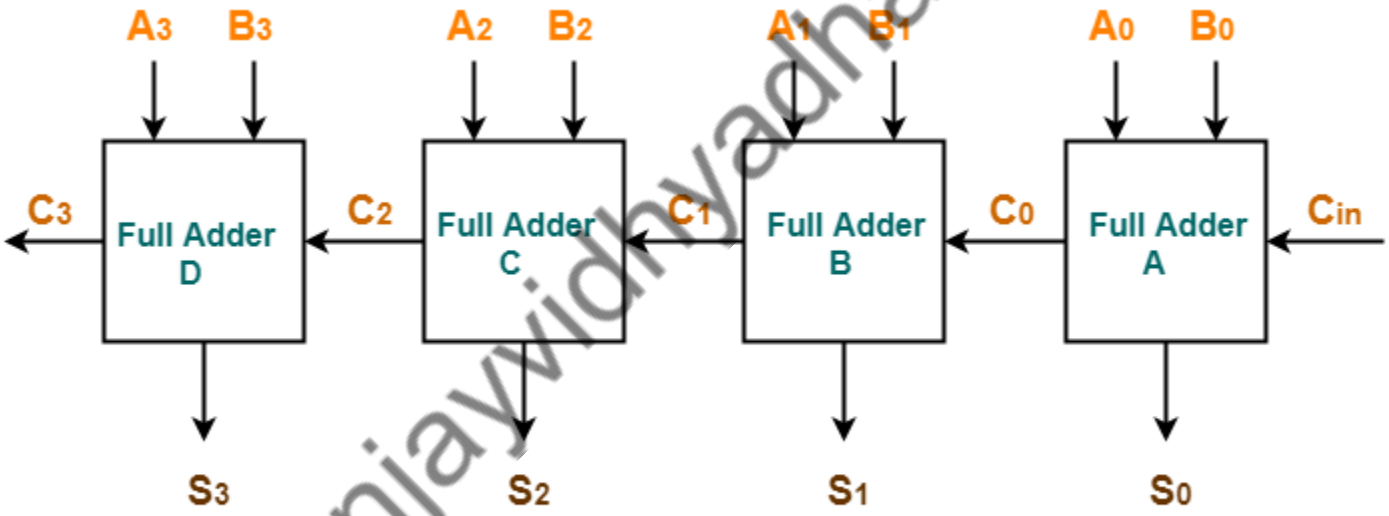
$= AB + C_{in}(A \oplus B) + C_{in}AB = AB(1+C_{in}) + C_{in}(A \oplus B)$

$= AB + C_{in}(A \oplus B)$



# Ripple Carry Adder

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



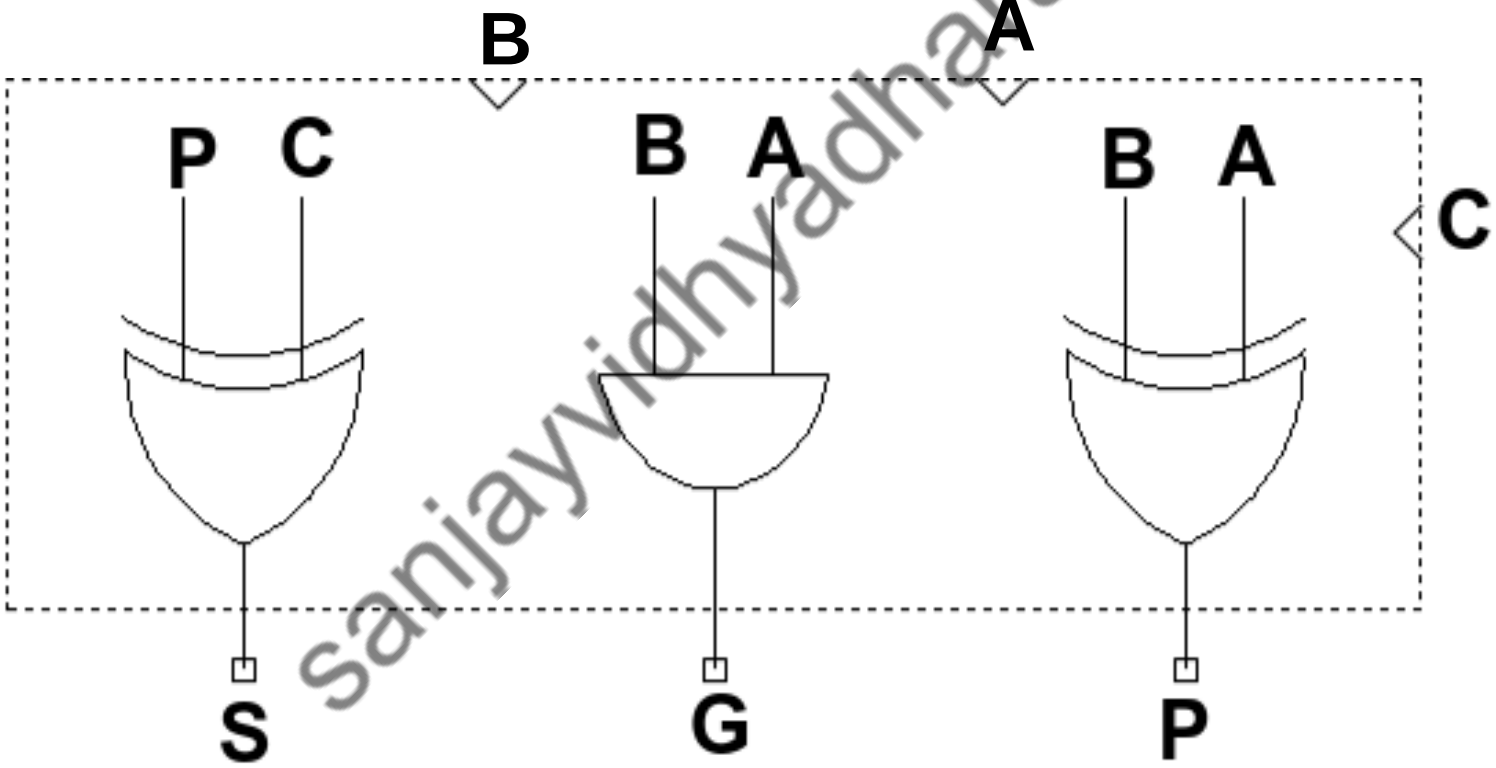

4-bit Ripple Carry Adder



Delay= 4 X Full Adder Delay =12 Gate Delays

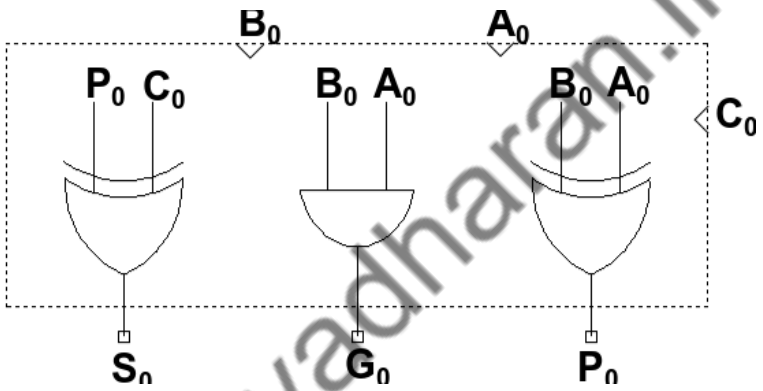
# Carry Look-Ahead Adder

1-bit CLA

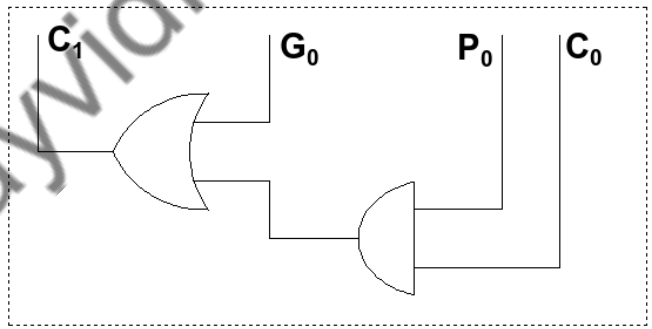


# Carry Look-Ahead Adder

CLA

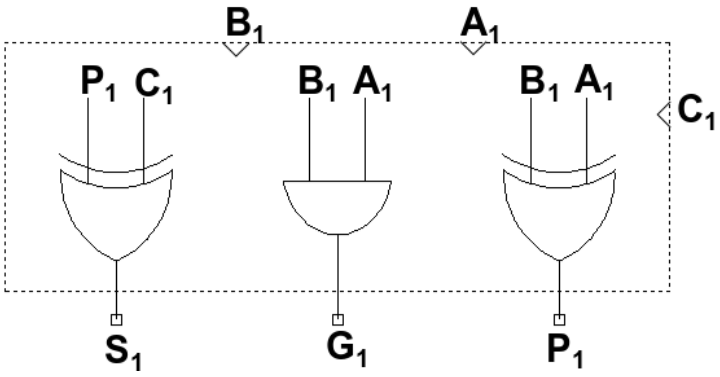


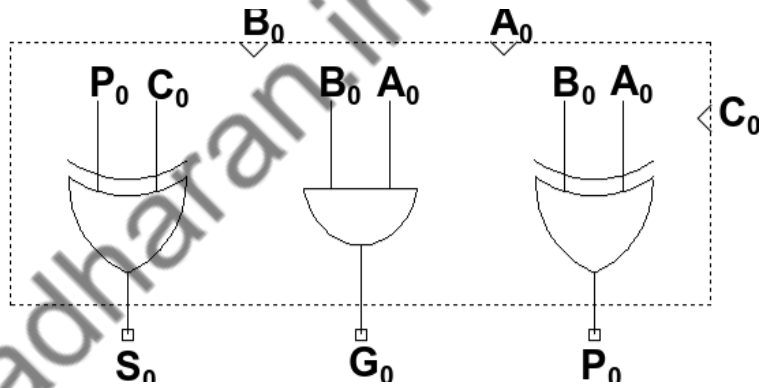

CLLB




$C_1=G_0+P_0C_0$

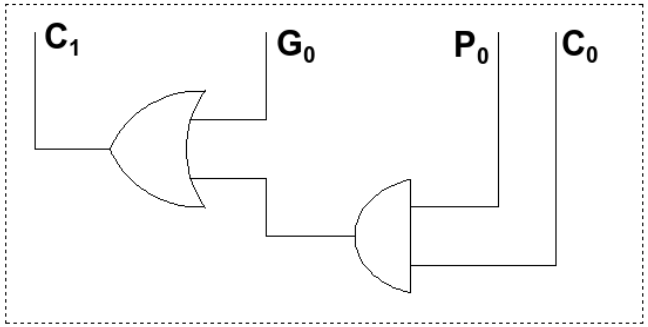
# Carry Look-Ahead Adder



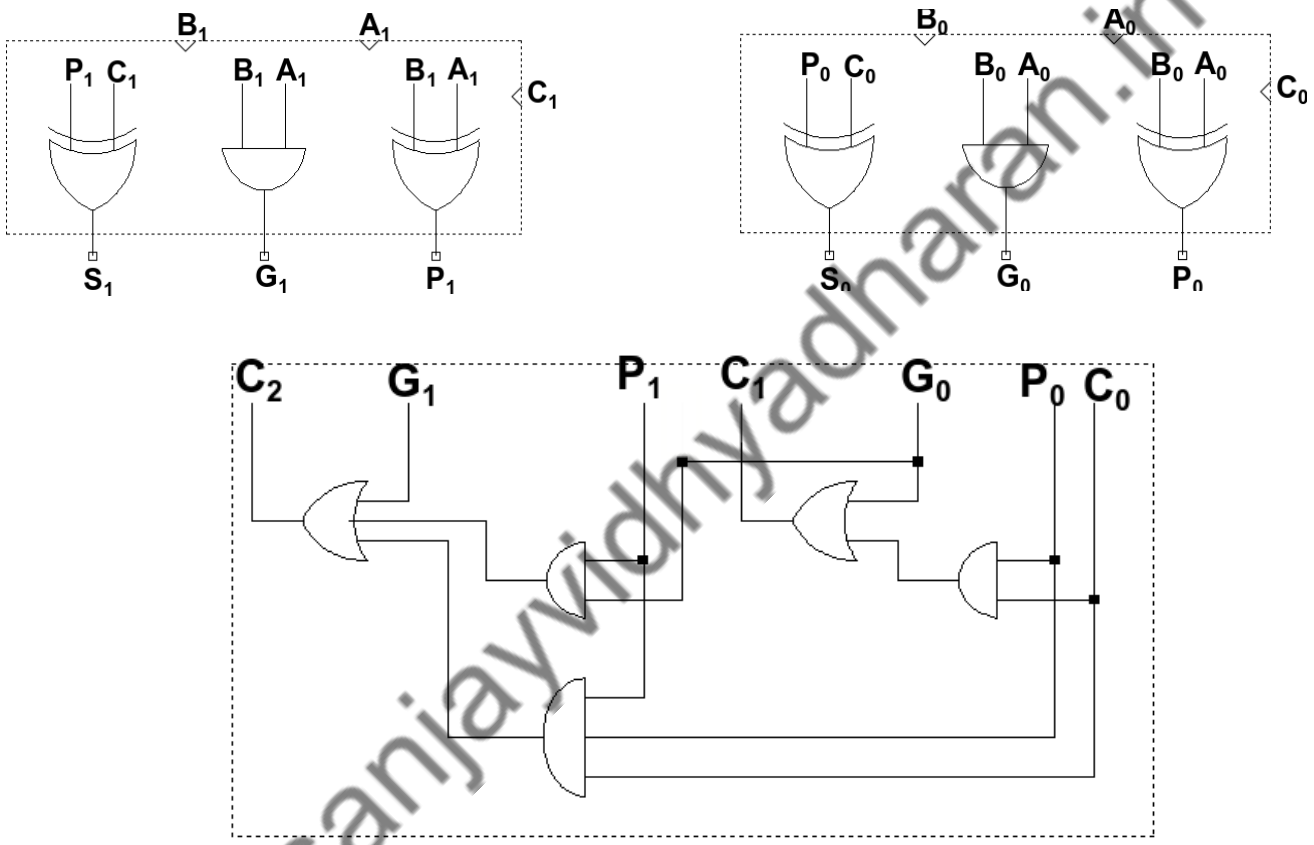




$$C_2 = G_1 + P_1 C_1$$

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



# 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

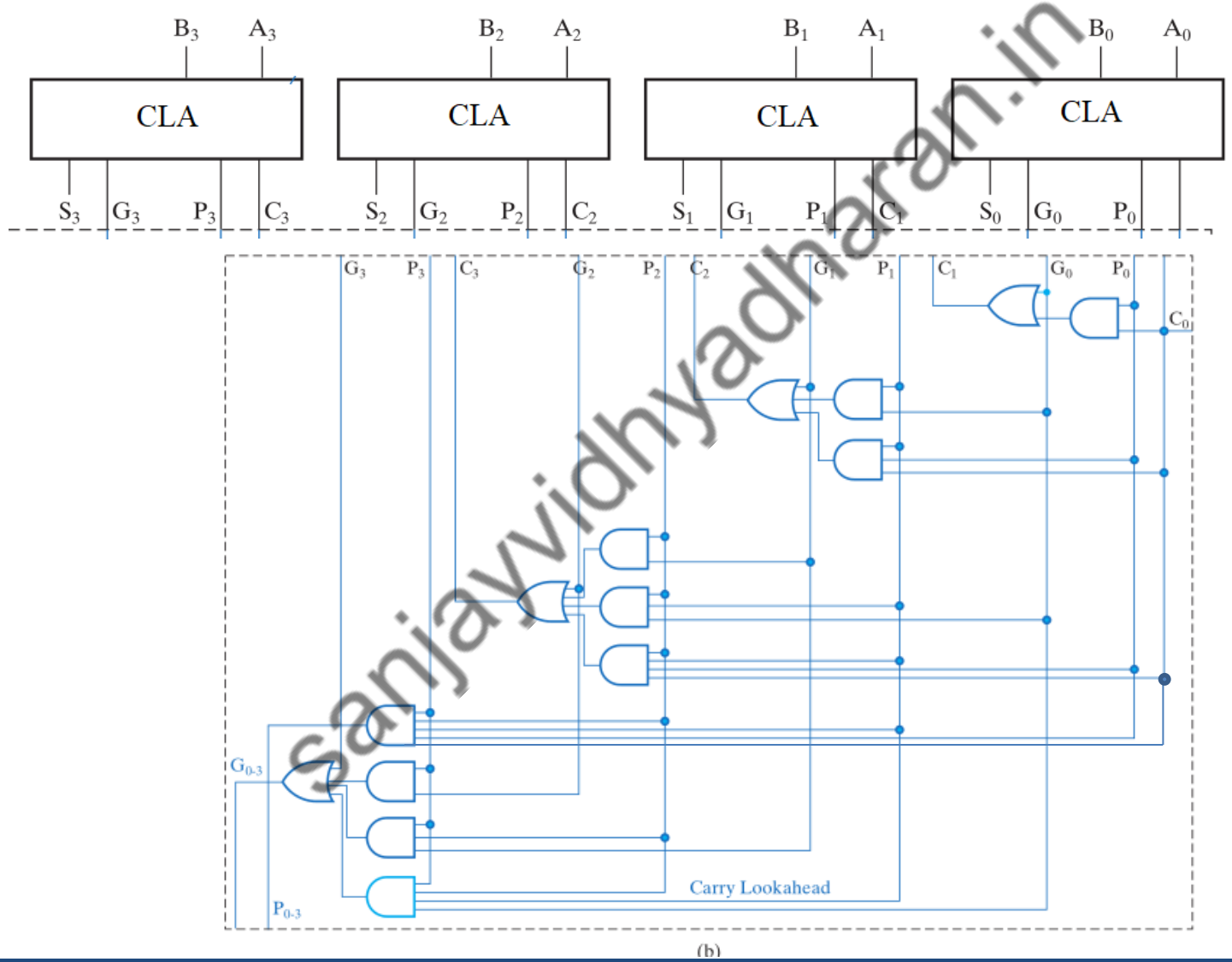
$$C_1=G_0+P_0C_0$$

$$C_2= G_1+P_1G_0+P_1P_0C_0$$

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

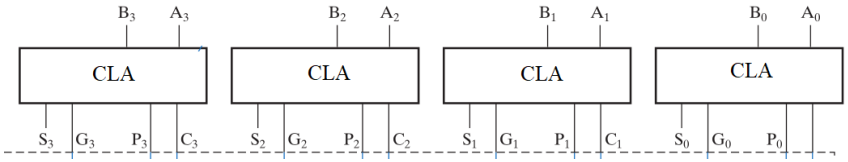
$$C_4=G_3+P_3G_2+ P_3P_2G_1 +P_3P_2P_1G_0 +P_3P_2P_1P_0C_0$$

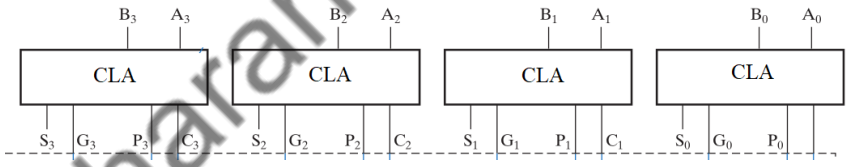
# Carry Look-Ahead Adder

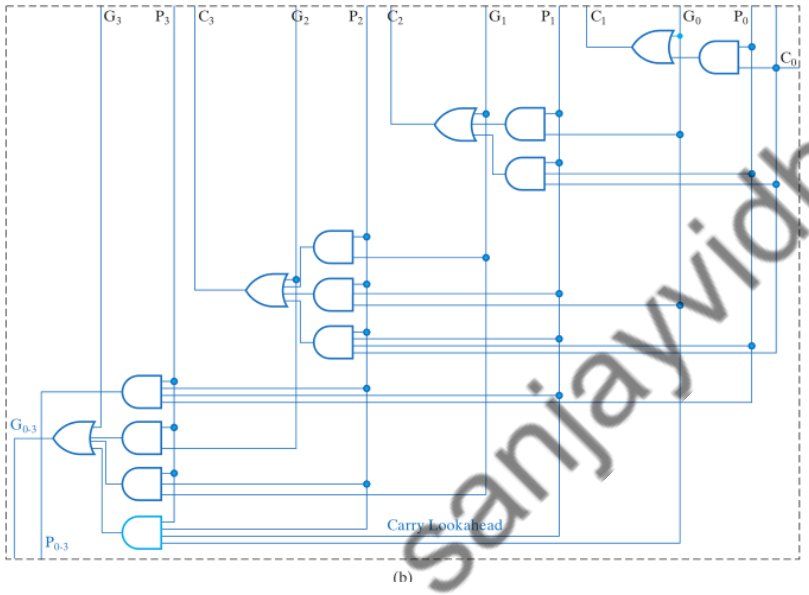


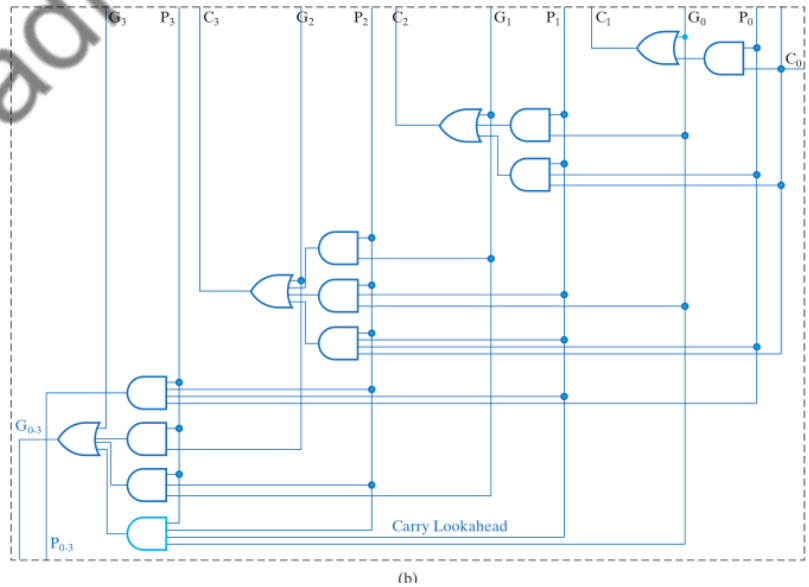
# Carry Look-Ahead Adder

## 8 Bit Full Adder



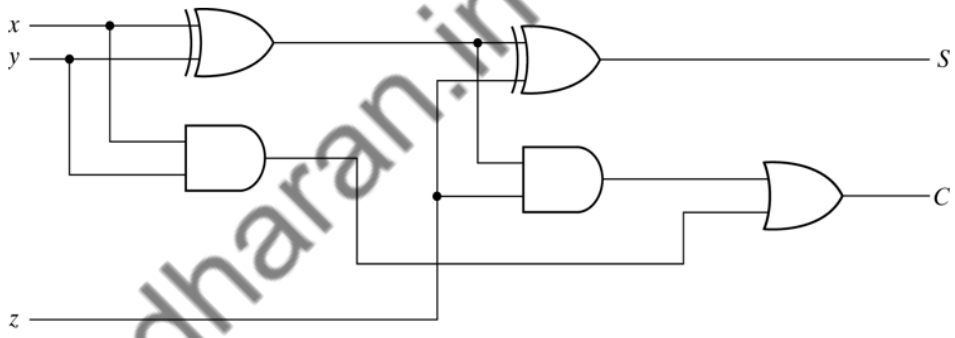



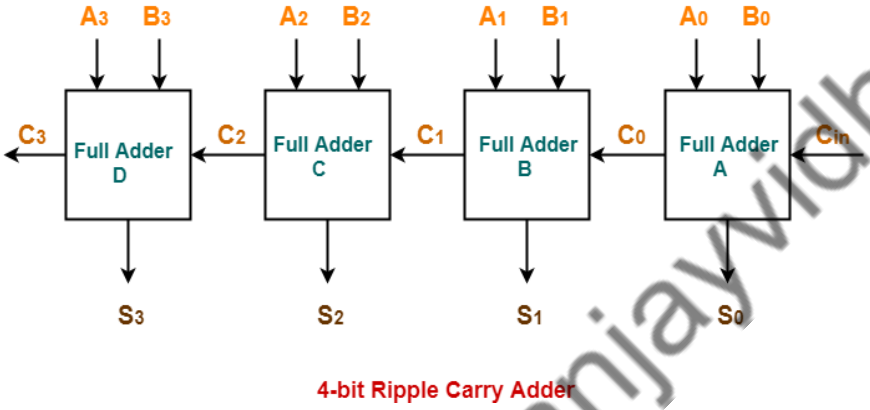




Delay = 2 X 3 =6 Gate Delay

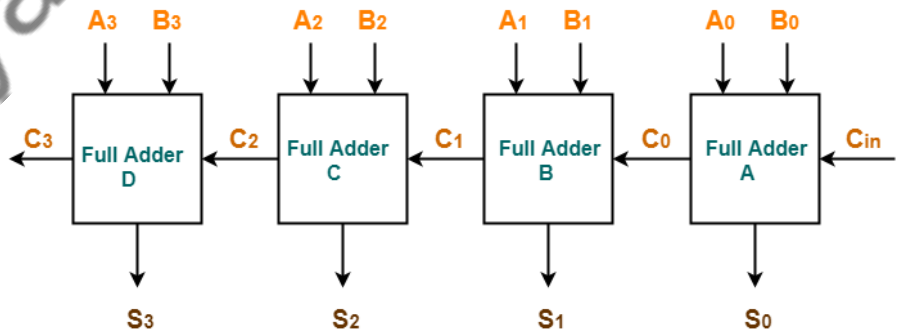
# Ripple Carry Adder

8 Bit Full Adder



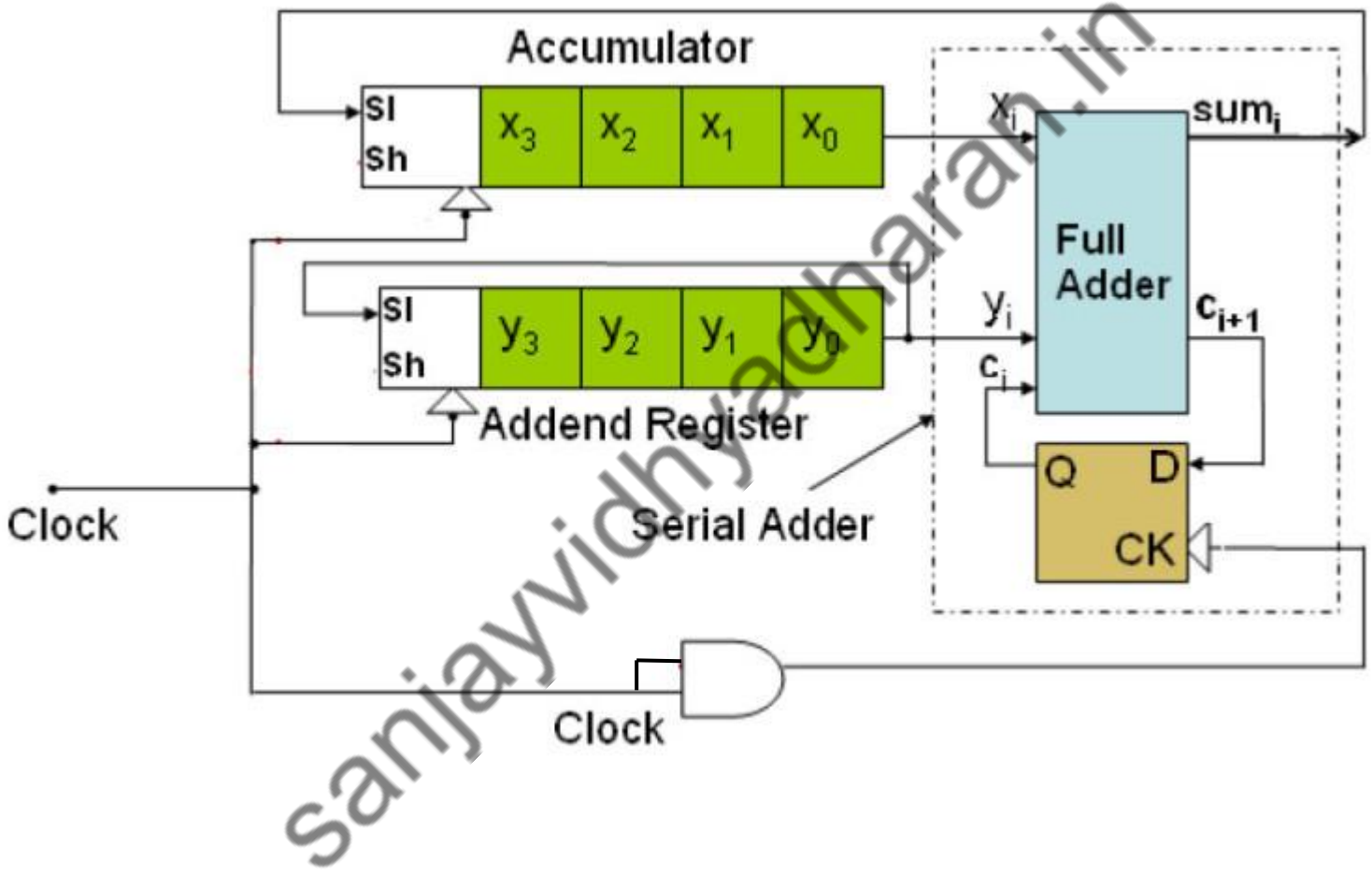

4-bit Ripple Carry Adder




4-bit Ripple Carry Adder

Delay = 8 X 3 Gate Delay

# 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**

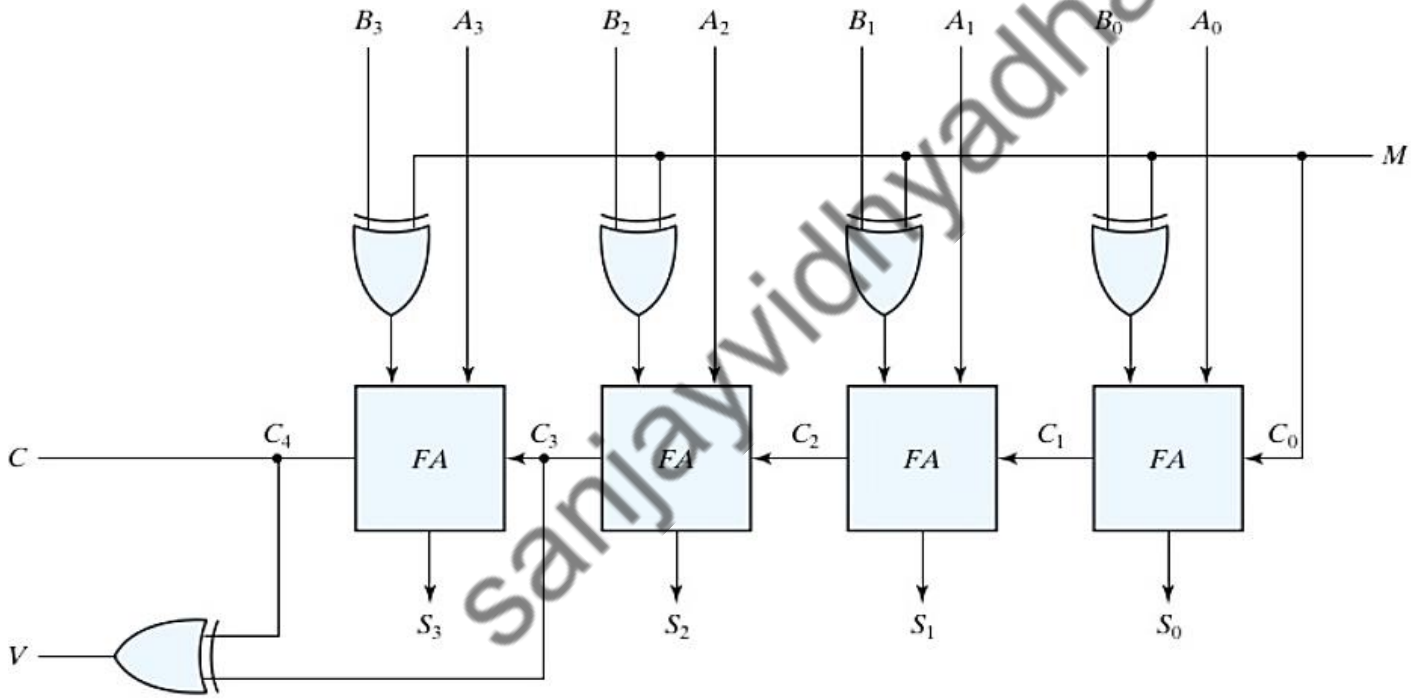
1100
1011
1 0111

**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 Multiplier

| | | 2 | 3 |
|---|---|---|---|
| | x | 5 | 2 |
| | | 4 | 6 |
| 1 | 1 | 5 | |
| 1 | 1 | 9 | 6 |

| | 1 | 0 |
|---|---|---|
| x | 1 | 1 |
| | 1 | 0 |
| 1 | 0 | |
| 1 | 1 | 0 |

2 x 3 = 6

# Binary Multiplier (2-bit x 2-bit)

$B_1 \quad B_0$

$x \quad A_1 \quad A_0$

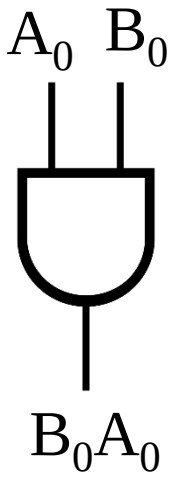
$B_1A_0 \quad B_0A_0$

$B_1A_1 \quad B_0A_1$

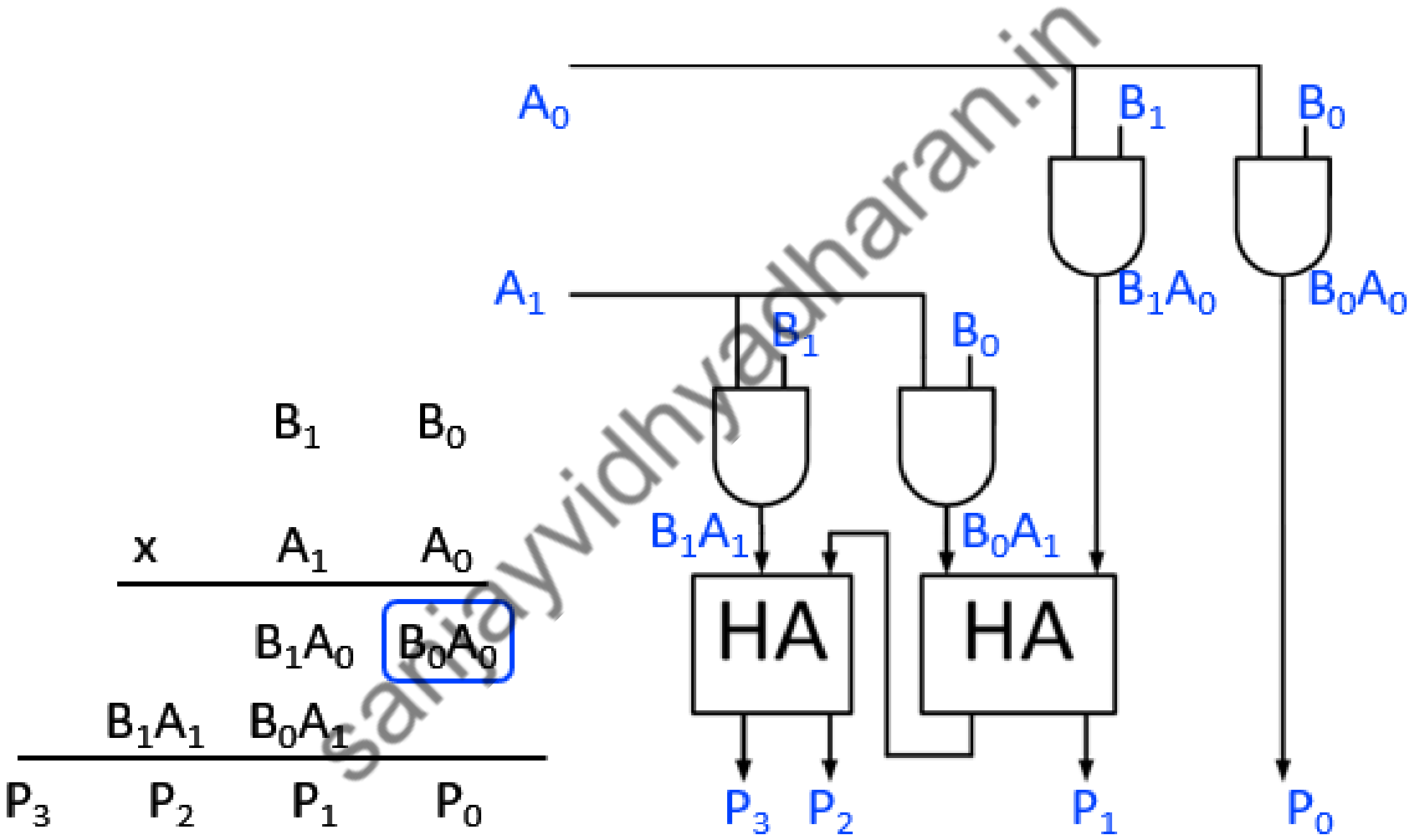
$P_3 \quad P_2 \quad P_1 \quad P_0$

Gate for 1-bit Multiplication ($B_0A_0$) ?

| $A_0$ | $B_0$ | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |



# Binary Multiplier (2-bit x 2-bit)



# Binary Multiplier (4-bit x 4-bit)

(Multiplicand) $B_3\ B_2\ B_1\ B_0$

(Multiplier) $A3\ A_2\ A_1\ A_0$

---

$A_0B_3$ $A_0B_2$ $A_0B_1$ $A_0B_0$

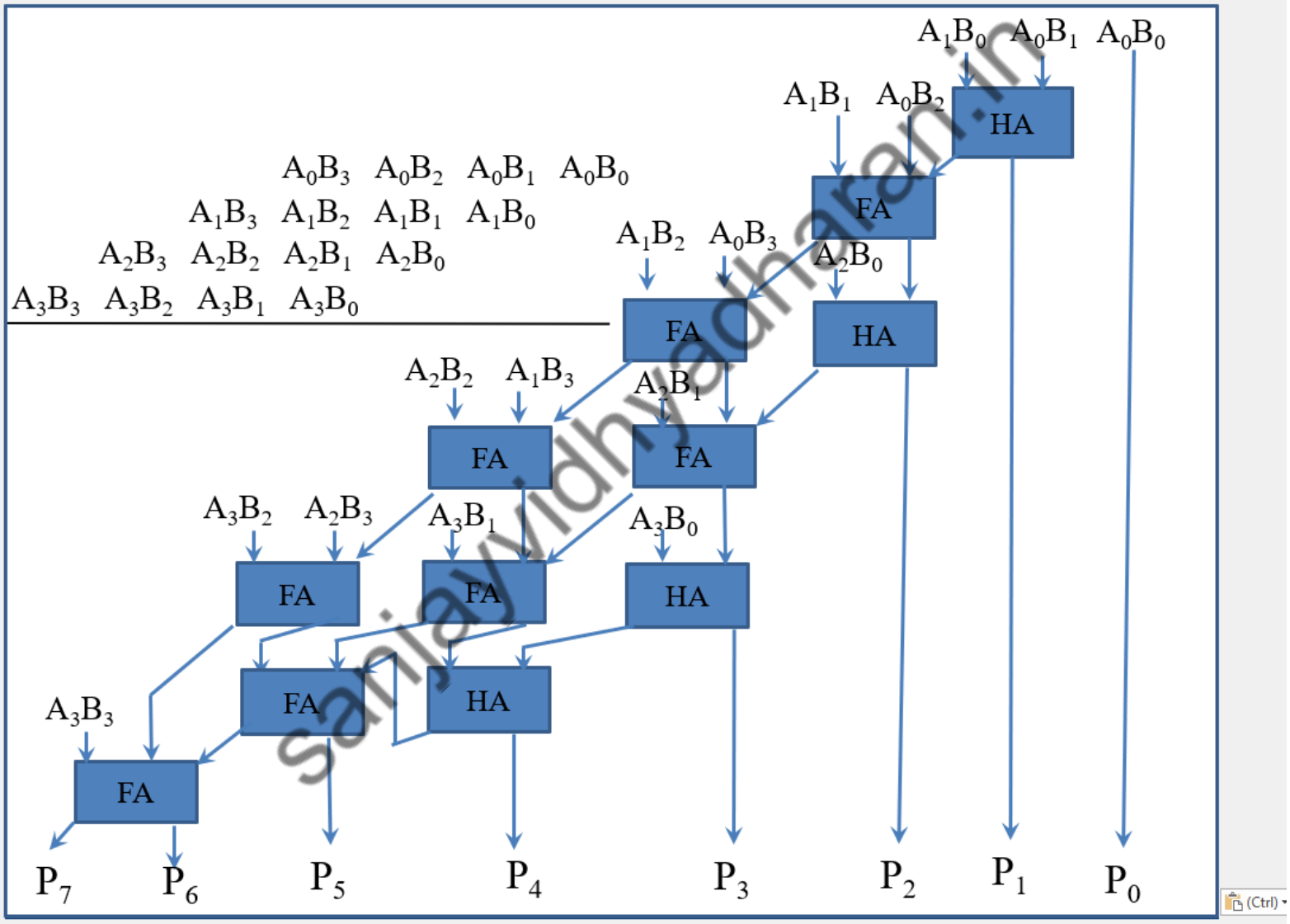
$A_1B_3$ $A_1B_2$ $A_1B_1$ $A_1B_0$

$A_2B_3$ $A_2B_2$ $A_2B_1$ $A_2B_0$

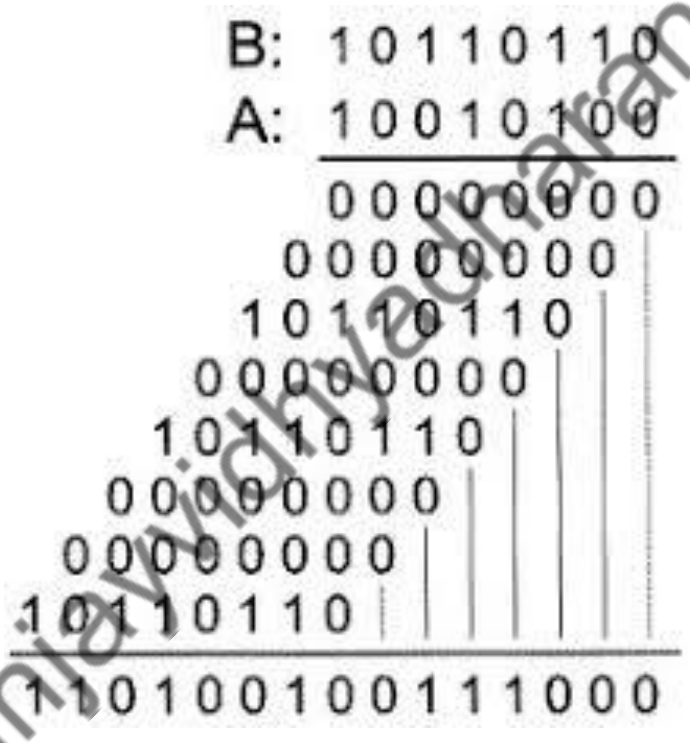
$A_3B_3$ $A_3B_2$ $A_3B_1$ $A_3B_0$

---

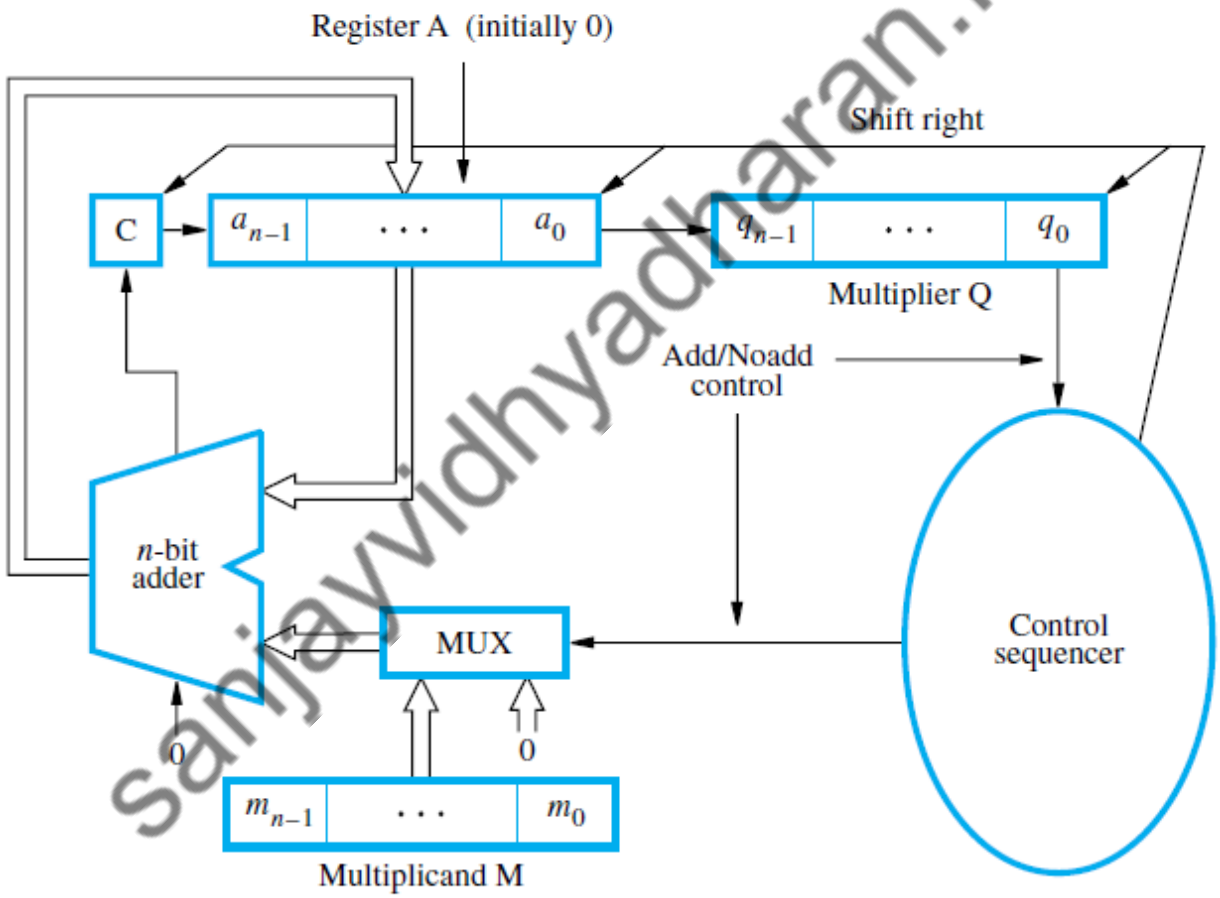
# Binary Multiplier (4-bit x 4-bit)



# Booth Algorithm



# Booth Algorithm




(a) Register configuration

# Booth Algorithm

| 3x5 | | | | |
|---|---|---|---|---|
| | M | A | Q | |
| Initial Condition | 0011 | 00000 | 0101 | |
| Clk ↑ (Load) | 0011 | 00011 | 0101 | 1 |
| Clk ↓ (Shift Right) | 0011 | 00001 | 1010 | |
| Clk ↑ (Load) | 0011 | 00001 | 1010 | 2 |
| Clk ↓ (Shift Right) | 0011 | 00000 | 1101 | |
| Clk ↑ (Load) | 0011 | 00011 | 1101 | 3 |
| Clk ↓ (Shift Right) | 0011 | 00001 | 1110 | |
| Clk ↑ (Load) | 0011 | 00001 | 1110 | 4 |
| Clk ↓ (Shift Right) | 0011 | 00000 | 1111 | |

# Thank you