



Adder

Carry Look-Ahead Adder

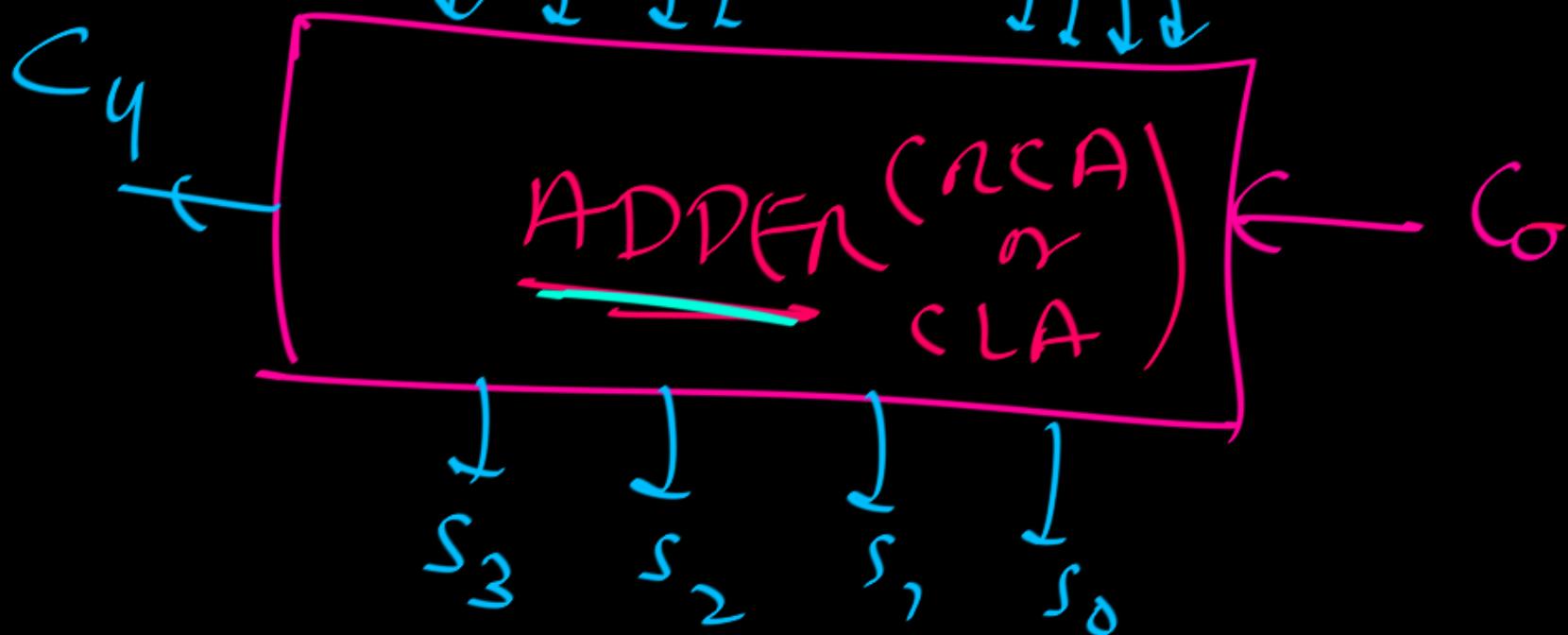
Q: Is C_0 Always 0 ?

Practically; we manufacture address
of 4-bit; 8-bit Addition only.

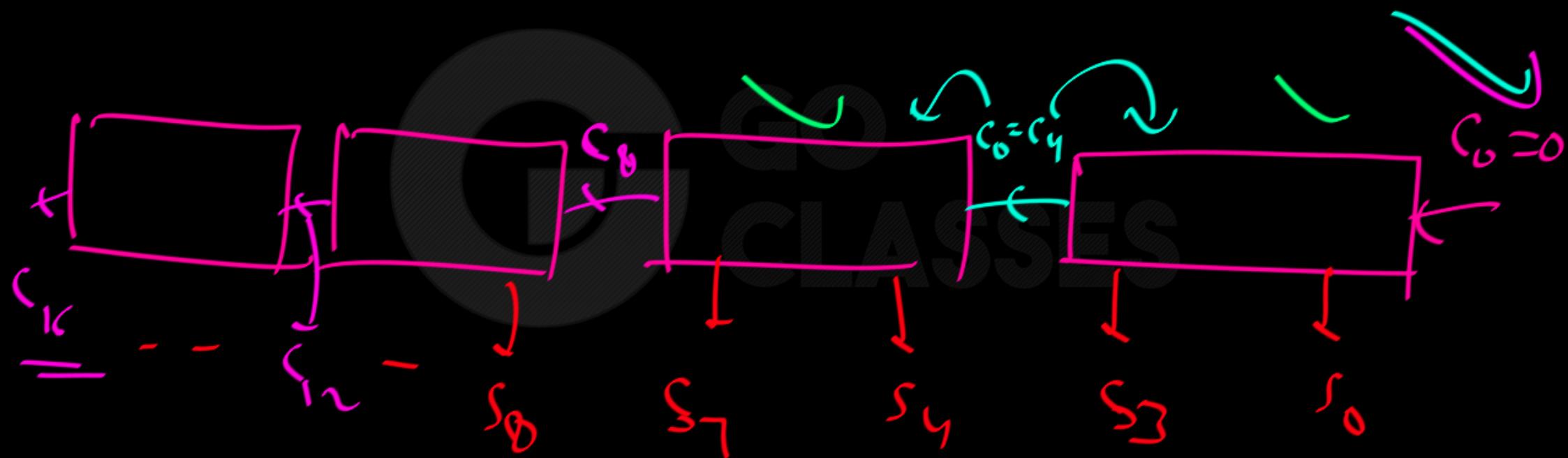
Adder: $\boxed{n_3 \ n_2 \ n_1 \ n_0} \leftarrow C_0$

Adder Available :

$a_3 \ a_2 \ a_1 \ a_0$ $b_3 \ b_2 \ b_1 \ b_0$



16 bit Addition:



Q: If fan-in unlimited for logic gates and we only have Basic Gates (AND, OR, NOT) each gate delay \Rightarrow f

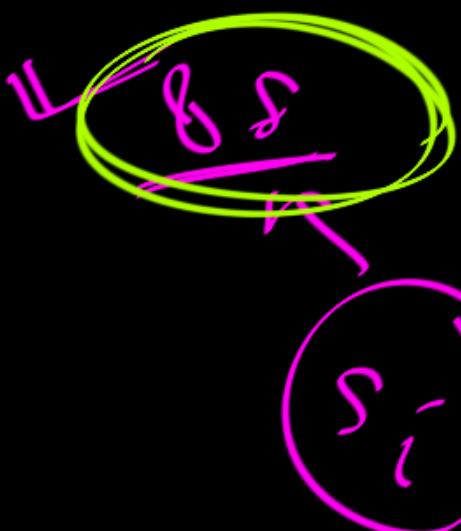
n-bit CLA ; Delay ??

Mys:

8f

\Rightarrow Constant

$t = 3f$ $[P_{n-1} S_{n-1} \dots S_0 P_0]$



3f

$\frac{t}{f}$

$2f \leftarrow$ carry C.C.

$t = 5f$ $[C_n \dots C_1 C_0]$



All sum bits available/valid;

All carry bits available/valid ;

5f.

Q: fan-in is 2. \Rightarrow Delay = f

1 bit CLA using basic gates.

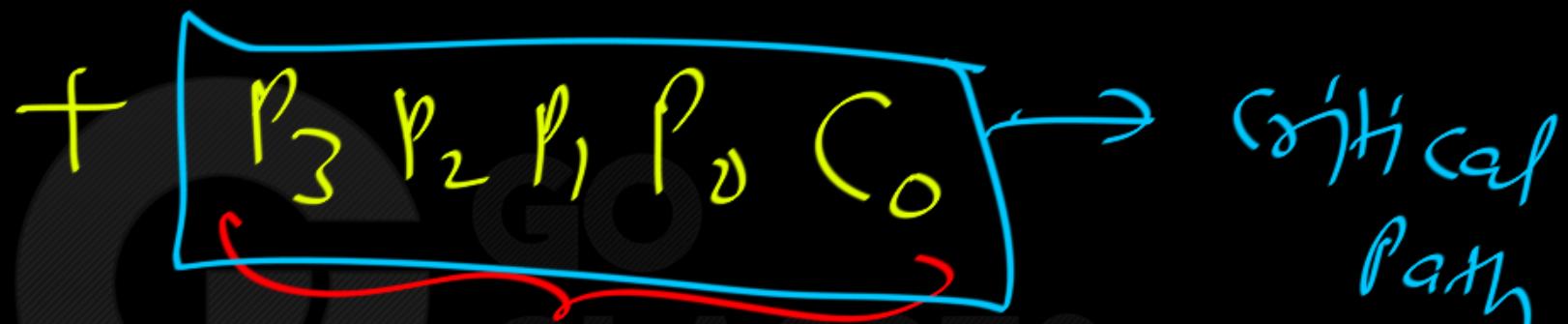
Carry Generator Circuit will take How much delay?



C_0 = Given initially

$C_i \Rightarrow (i+1)$ fan-in on gate }
= $\Rightarrow (i+1)$ fan-in AND gate }

$$\overline{C_4} : G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0$$



5 input
OR gate

AND
5 input
gate



$P_3 P_2 P_1 P_0 C_0$

↓

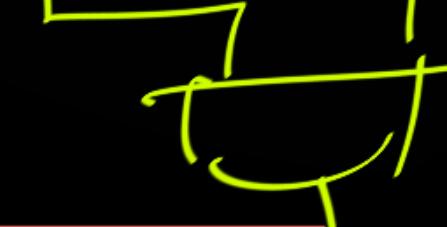
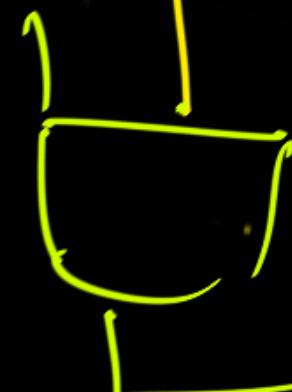
3 stages



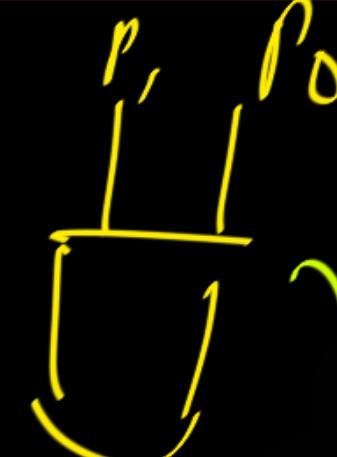
$P_3 P_2$



C_0



$P_1 P_0$



3 levels



5 input OR gate \Rightarrow

3 levels.

Carry - Gen - CK+ \Rightarrow

6 levels

= 6 f.



n-bit CLA :

$C_n =$

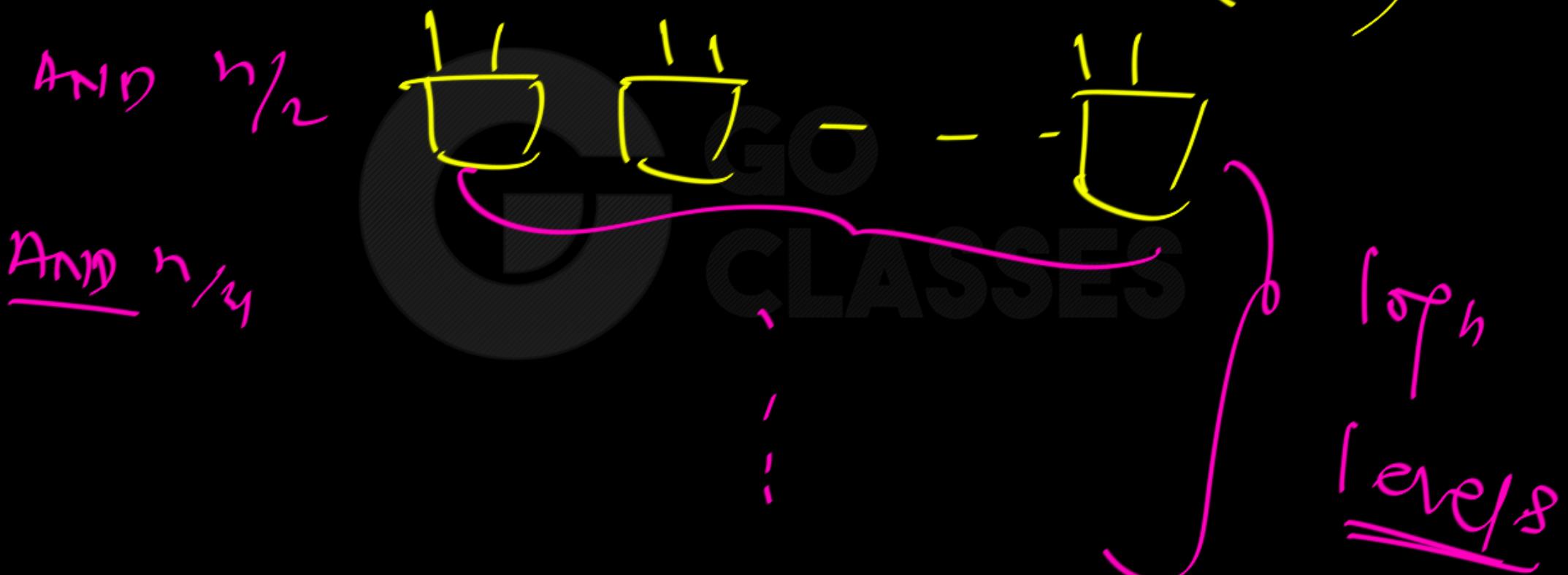


$n+1$ terms

$P_{n-1} P_{n-2} \dots P_0 C_0$

$n + 1$
arrays

$(n+1)$ fan-in AND gate $\Rightarrow \Theta(\log n)$ levels



$(n+1)$ fan-in OR gate $\Rightarrow \mathcal{O}(\log n)$ levels

Total level for Carry connect =
 $\underbrace{\mathcal{O}(\log n)}$ AND + $\underbrace{\mathcal{O}(\log n)}$ OR = $\mathcal{O}(\log n)$ levels.

Carry Gen. CKT delay = $\Theta(\log_2 n)$

When fanin is limited or constant.

$$\Theta(\log_2 n) = \Theta(\log_5 n) = \Theta(\log_{15} n)$$



fan-in of a logic gate:

inputs of logic gate.

fan-out of a logic gate:

outputs of logic gate.

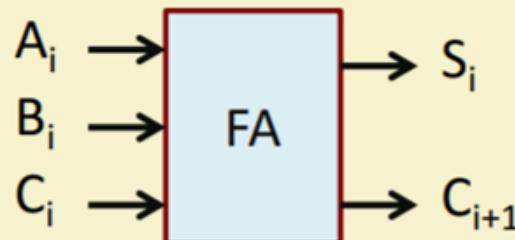
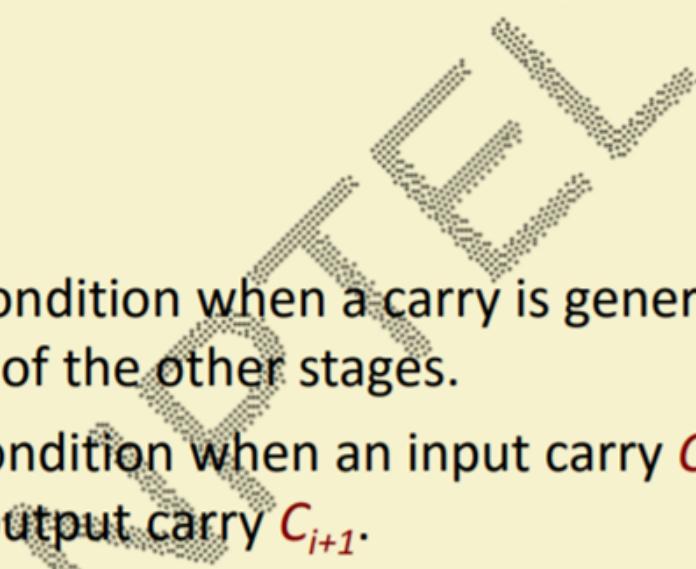


- Consider the i -th stage in the addition process.
- We define the *carry generate* and *carry propagate* functions as:

$$G_i = A_i \cdot B_i$$

$$P_i = A_i \oplus B_i$$

- $G_i = 1$ represents the condition when a carry is generated in stage- i independent of the other stages.
- $P_i = 1$ represents the condition when an input carry C_i will be propagated to the output carry C_{i+1} .
- We can generate the output carry in terms of G_i and P_i .



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



(using carry lookahead logic)

$$P_i = A_i \oplus B_i$$

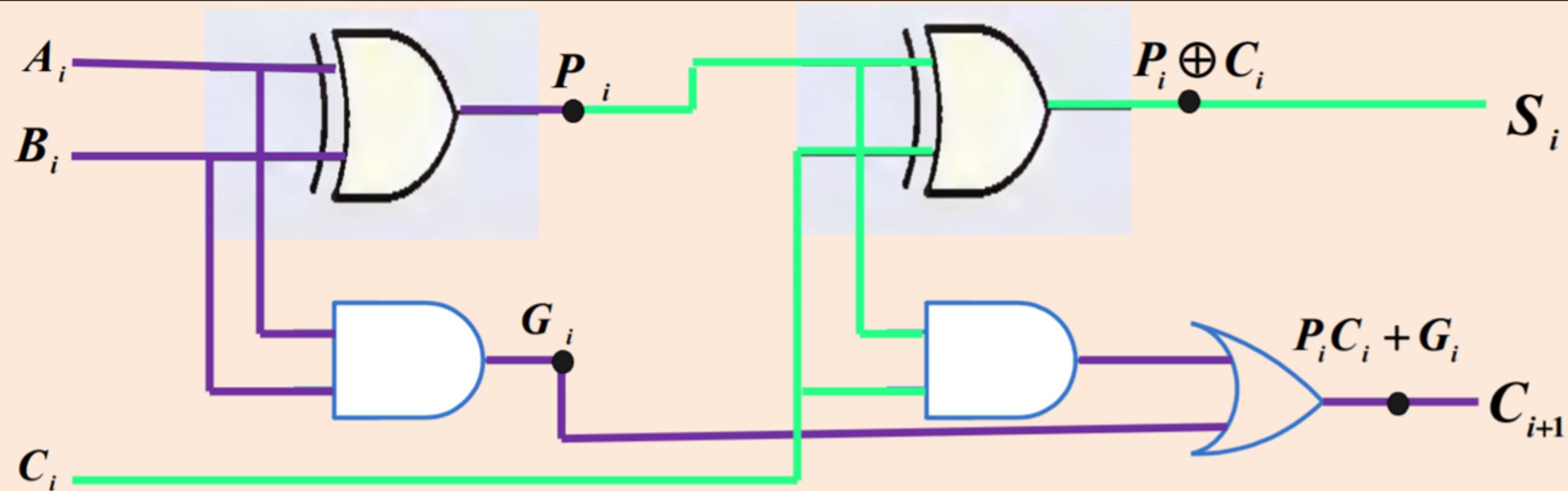
$$G_i = A_i B_i$$

The output sum and carry are:

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

- ✓ G_i -called a **carry generate**, and it produces a carry of **1** when both A_i and B_i are **1**.
- ✓ P_i -called a **carry propagate**, it determines whether a carry into stage **i** will propagate into stage **i + 1**.



Full Adder with P and G



$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

the output sum and carry can respectively be expressed as

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

G_i is called a *carry generate*, and it produces a carry of 1 when both A_i and B_i are 1, regardless of the input carry C_i . P_i is called a *carry propagate*, because it determines whether a carry into stage i will propagate into stage $i + 1$ (i.e., whether an assertion of C_i will propagate to an assertion of C_{i+1}).



- ✓ The **Boolean function** for the carry outputs of each stage and substitute the value of each C_i from the previous equations:

$$\left\{ \begin{array}{l} C_0 = \text{input carry} \\ C_1 = G_0 + P_0 C_0 \\ C_2 = G_1 + P_1 C_1 = G_1 + P_1(G_0 + P_0 C_0) \\ \quad = G_1 + P_1 G_0 + P_1 P_0 C_0 \\ C_3 = G_2 + P_2 C_2 = G_2 + P_2(G_1 + P_1 G_0 + P_1 P_0 C_0) \\ \quad = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \end{array} \right\}$$



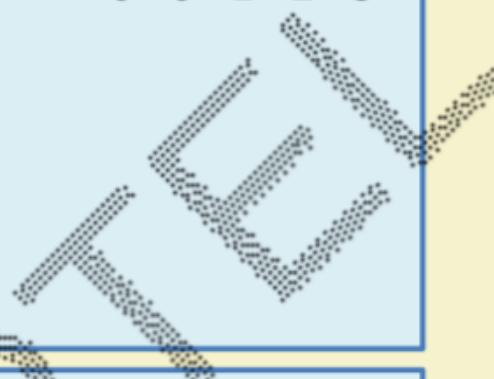
Design of 4-bit CLA Adder

$$C_4 = G_3 + G_2P_3 + G_1P_2P_3 + G_0P_1P_2P_3 + C_0P_0P_1P_2P_3$$

$$C_3 = G_2 + G_1P_2 + G_0P_1P_2 + C_0P_0P_1P_2$$

$$C_2 = G_1 + G_0P_1 + C_0P_0P_1$$

$$C_1 = G_0 + C_0P_0$$

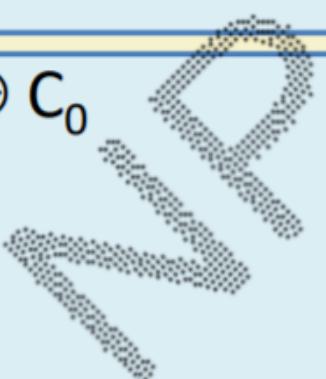


$$S_0 = A_0 \oplus B_0 \oplus C_0 = P_0 \oplus C_0$$

$$S_1 = P_1 \oplus C_1$$

$$S_2 = P_2 \oplus C_2$$

$$S_3 = P_3 \oplus C_3$$



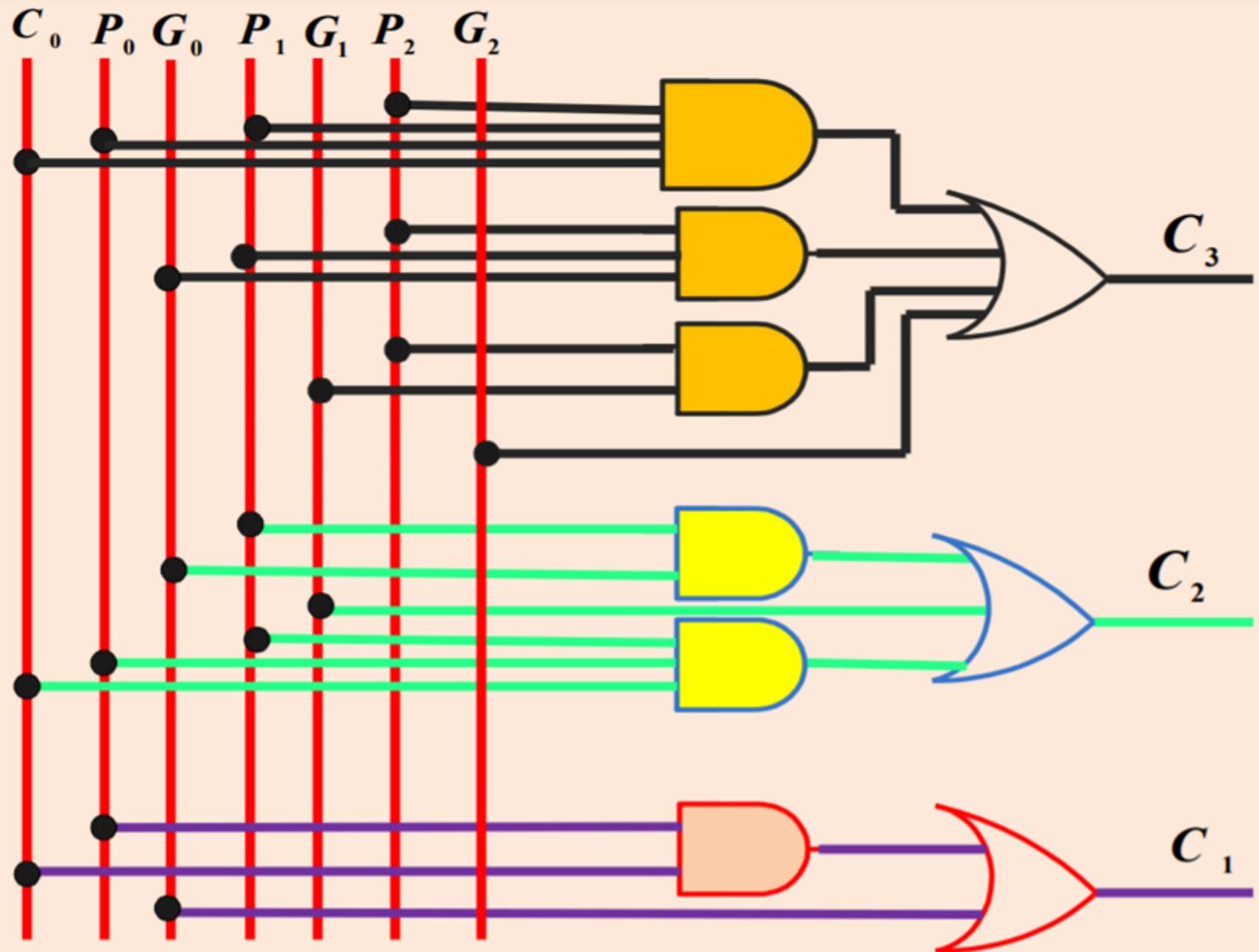


- The three Boolean functions C_1 , C_2 and C_3 are implemented in the ***carry lookahead generator.***

The two level-circuit for the output carry C_4 is not shown, it can be easily derived by the equation.

- C_3 does not have to wait for C_2 and C_1 to propagate, in fact C_3 is propagated at the same time as C_1 and C_2 .

Since the Boolean function for each output carry is expressed in sum-of-products form, each function can be implemented with one level of AND gates followed by an OR gate (or by a two-level NAND).



Logic Diagram for Carry Lookahead Generator

Carry Generator Circuit:

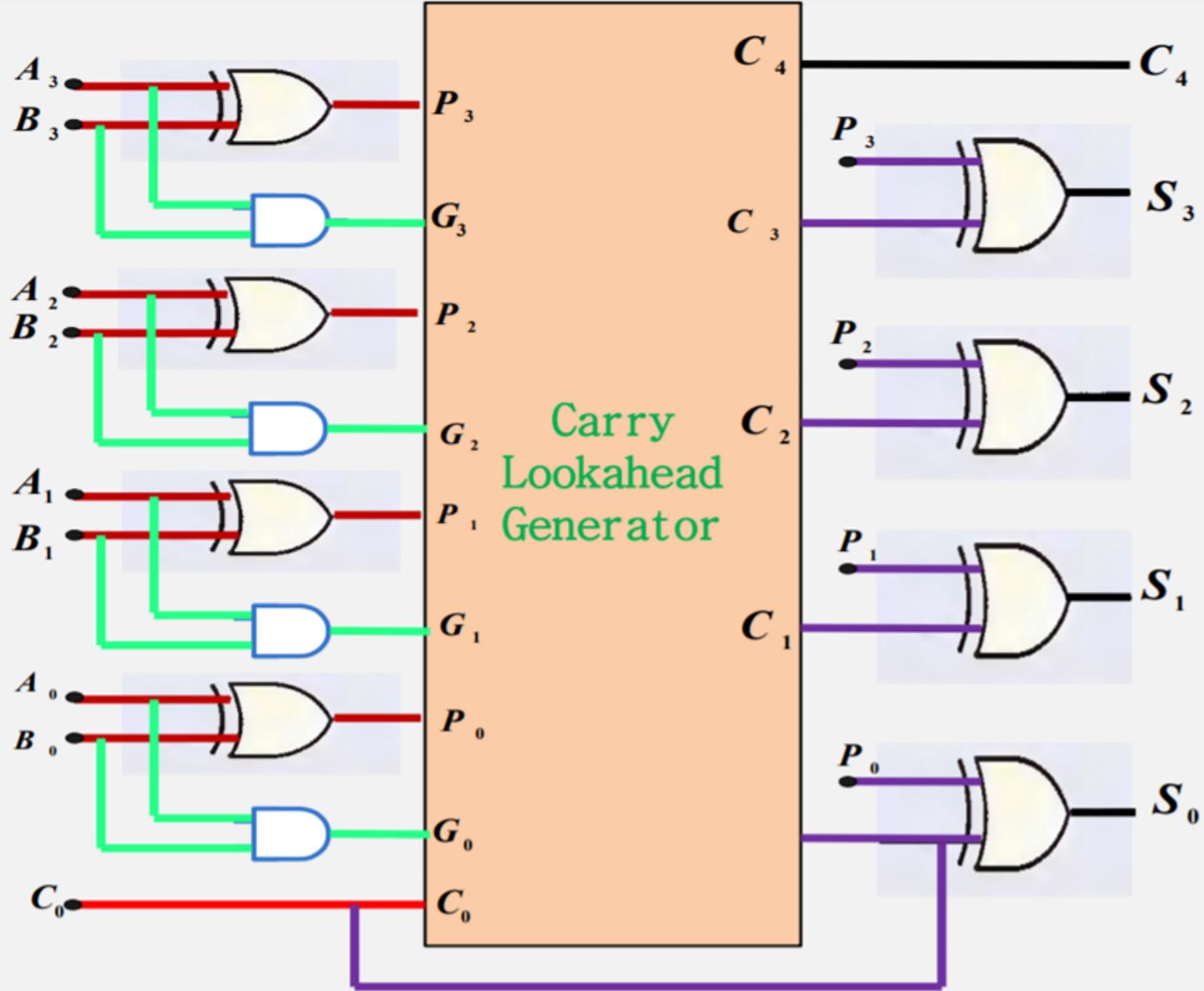
at two levels: AND — OR
implementation

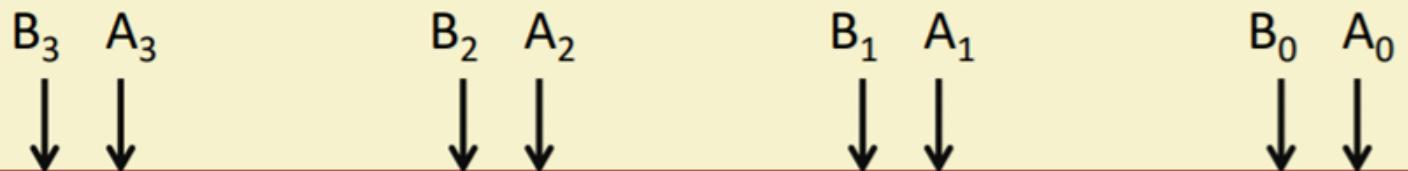


The construction of a **four-bit adder with a carry lookahead scheme** is the following:



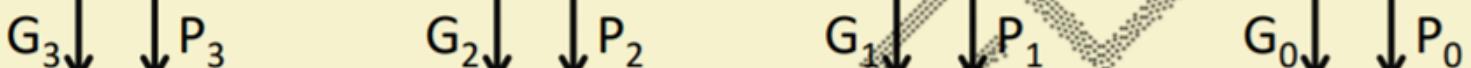
four-bit
adder
with a
carry
look
ahead
scheme





G_i and P_i Generator

3δ



4-bit Carry Look Ahead Circuit

2δ

C_3

C_2

C_1

C_0

3δ

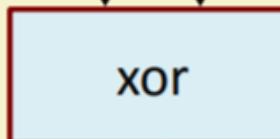
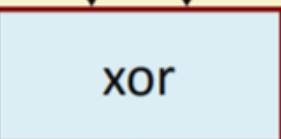
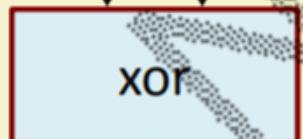
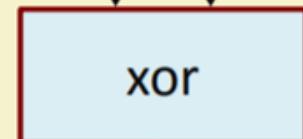
C_4

xor

xor

xor

xor



S_3

S_2

S_1

S_0

Total Time = 8δ

In a look-ahead carry generator, the carry generate function G_i and the carry propagate function P_i for inputs A_i and B_i are given by:

$$P_i = A_i \oplus B_i \text{ and } G_i = A_i B_i$$

The expressions for the sum bit S_i and the carry bit C_{i+1} of the look ahead carry adder are given by:

$$S_i = P_i \oplus C_i \text{ and } C_{i+1} = G_i + P_i C_i, \text{ where } C_0 \text{ is the input carry.}$$

Consider a two-level logic implementation of the look-ahead carry generator. Assume that all P_i and G_i are available for the carry generator circuit and that the AND and OR gates can have any number of inputs. The number of AND gates and OR gates needed to implement the look-ahead carry generator for a 4-bit adder with S_3, S_2, S_1, S_0 and C_4 as its outputs are respectively:

- A. 6, 3
- B. 10, 4
- C. 6, 4
- D. 10, 5

Unlimited fan-in

$$C1 = G0 + C0.P0$$

$$C2 = G1 + G0.P1 + C0.P0.P1$$

$$C3 = G2 + G1.P2 + G0.P1.P2 + C0.P0.P1.P2$$

$C4 = G3 + G2.P3 + G1.P2.P3 + G0.P1.P2.P3 + C0.P0.P1.P2.P3$ // read this as carry is generated in 3rd stage OR carry is generated in 2nd stage AND propagated to 3rd stage OR carry is generated in 1st stage AND carry is propagated through 2nd AND 3rd stage OR carry is generated in 0th stage AND propagated through 1st, 2nd AND 3rd stage OR initial carry is propagated through 0th, 1st, 2nd AND 3rd stage.

4 OR gates are required for $C1, C2, C3, C4$

- 1 AND gate for $C1$
- 2 AND gate for $C2$
- 3 AND gate for $C3$
- 4 AND gate for $C4$
- AND = 10
- OR = 4

Correct Answer: B.



CLA Generator



+
 C_4



+
 C_3



+
 C_2



+
 C_1

C_0

10 AND gate; 4 on sale

4.1.9 Adder: GATE CSE 2016 Set 1 | Question: 33 top ↹<https://gateoverflow.in/39688>

testis.gatecse.in

goclasses.in

testis.gate



Consider a carry look ahead adder for adding two n-bit integers, built using gates of fan-in at most two. The time to perform addition using this adder is

- A. $\Theta(1)$
- B. $\Theta(\log(n))$
- C. $\Theta(\sqrt{n})$
- D. $\Theta(n)$

4.1.9 Adder: GATE CSE 2016 Set 1 | Question: 33 top ↹<https://gateoverflow.in/39688>

test.gatecse.in

goclasses.in

test.gatecse.in



Consider a carry look ahead adder for adding two n-bit integers, built using gates of fan-in at most two. The time to perform addition using this adder is

- A. $\Theta(1)$
- B. $\Theta(\log(n))$ ✓
- C. $\Theta(\sqrt{n})$
- D. $\Theta(n)$

fan-in = 2 $\Rightarrow \Theta(1)$ time ✓

$P_i \Rightarrow a_i \oplus b_i \Rightarrow 3$ levels (NOT \rightarrow AND \rightarrow OR)

Carry len $C_{k+1} = \Theta(\log n)$ Delay

$S_i \Rightarrow P_i \oplus C_i \Rightarrow 3$ levels $\Rightarrow \Theta(1)$ time.
 $(\log n + 3 + 3)$