

Welcome to ACE Engineering Academy - online live class

Subject: **Computer Organization and Architecture**

Faculty: **Y.V. Ramaiah**

Subject

Computer organization & Architecture

Chapters (Topics)

I. Computer Arithmetic ✓

II. Memory Organization

III. Secondary Memories

IV. Basic processor organization and Design

V. Pipeline organization

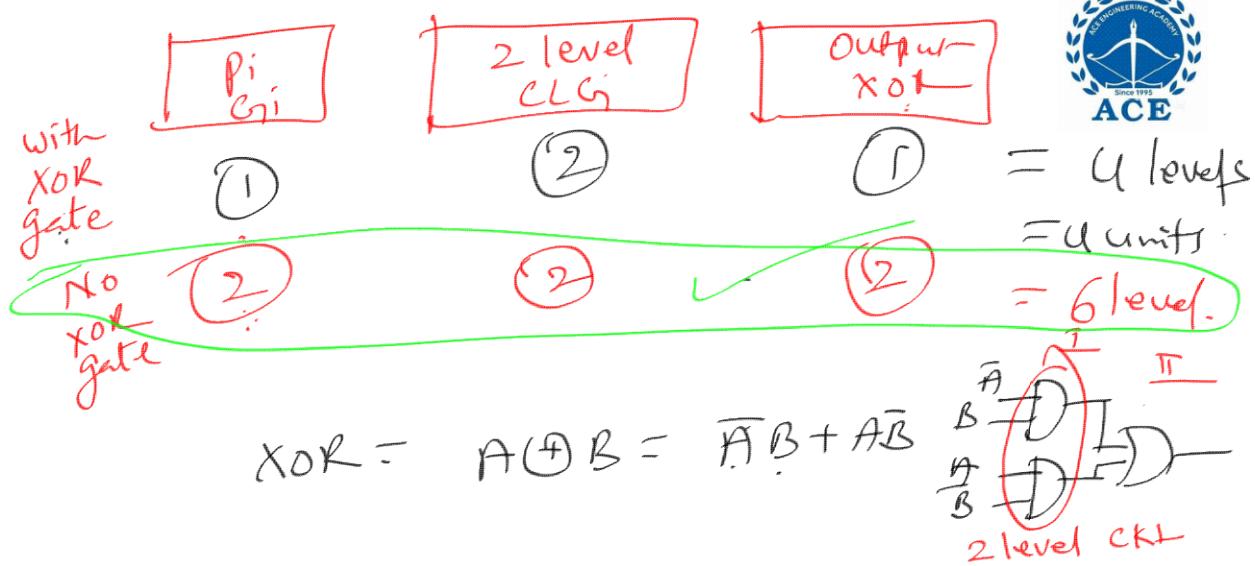
VI. Control unit Design

VII. IO Organization

Chapter - 1

Computer Arithmetic

1. Signed binary data representation (Integer) ✓
 2. Overflow concept in signed 2's complement representation ✓
 3. Different Arithmetical operations ✓
 4. Booth's Algorithm ✓
 5. Floating point representation ✓
 6. IEEE standards for floating point representation
 7. Ripple carry Adder
 8. Carry lookahead generator and Adder



Q. In a lookahead 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 $P_i = A_i \oplus B_i$ and $G_i = A_i B_i$. The expression for the sum bit S_i and carry bit C_{i+1} of the look ahead carry adder are given by $S_i = P_i \oplus C_i$ and $C_{i+1} = G_i + P_i C_i$, where C_0 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

4 bit
 $OK = 4$

AND

4x5
2
= 10.



$P_i = A_i \oplus B_i, G_i = A_i \cdot B_i$
 $S_i = A_i \oplus B_i \oplus C_i = P_i \oplus C_i$
 $C_{i+1} = C_i (A_i \oplus B_i) + A_i \cdot B_i = C_i P_i + G_i$
 P_i, G_i and C_0 inputs are directly available from Source without having any Delay.
 $C_L G_i$ is used to Generate C_1, C_2, C_3 and C_4 .
 $C_1 = C_0 P_0 + G_0 = P_0 C_0 + G_0$
 $C_2 = C_1 P_1 + G_1 = P_1 C_1 + G_1$
 $\therefore C_2 = P_1 (C_0 P_0 + G_0) + G_1 = P_1 P_0 C_0 + P_1 G_0 + G_1$
 $C_3 = P_2 C_2 + G_2 = P_2 (P_1 P_0 C_0 + P_1 G_0 + G_1) + G_2$
 $\therefore G_3 = P_2 P_1 P_0 C_0 + P_2 P_1 G_0 + P_2 G_1 + G_2$
 $\therefore C_3 = P_2 P_1 P_0 C_0 + P_2 P_1 G_0 + P_2 G_1 + G_2$
 $C_4 = P_3 C_3 + G_3$
 $= P_3 (P_2 P_1 P_0 C_0 + P_2 P_1 G_0 + P_2 G_1 + G_2) + G_3$
 $C_4 = P_3 P_2 P_1 P_0 C_0 + P_3 P_2 P_1 G_0 + P_3 P_2 G_1 + P_3 G_2 + G_3$

Final Result :-

$C_4 S_3 S_2 S_1 C_0$

$S_3 = P_3 \oplus C_3, S_2 = P_2 \oplus C_2, S_1 = P_1 \oplus C_1, S_0 = P_0 \oplus C_0$

Note: All P_3, P_2, P_1, P_0 and G_3, G_2, G_1, G_0 outputs are available after only one gate Delay.

Q) 0x FF D000 00 is a 32 bit data represented in IEEE 32 bit notation. Its Value in Decimal is

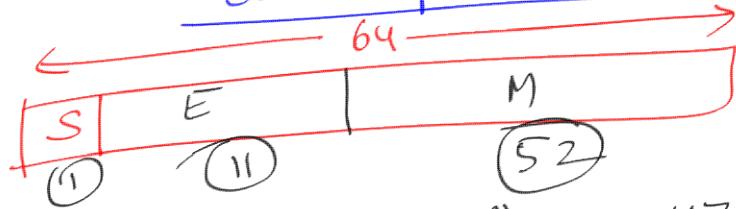


a b c d	-157.25 $+157.25$ Infinite NAN	s <div style="border: 1px solid black; padding: 5px; display: inline-block;"> $1111111101000000\cdots$ $E = 255$ $M \neq 0$ </div>
--------------------------	-------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



IEEE 64 bit Notation

Double precision



$E_{\min} = 0$, $E_{\max} = 2^{11} - 1 = 2047$
These are used for
Special Values.

$$b = 1023 \checkmark$$

$$= 2^{11} - 1 = 1023$$

$$= \frac{2^{11}}{2} = 1024$$

$$\frac{2^{11}}{2} = 2^{10}$$

$$K-1 = 10$$

$$K = 11$$

Expression Value = $(-1)^S \times 1.M \times 2^{E-1023}$



E range for Implicit normalized
number: $-1 \leq E \leq 2046$



S (1)	E (11)	M (52)	Representation
0/1	0	0	± 0
0 1	1111111111 2047	0	\pm Infinitive
0 1	1111111111	$M \neq 0$	NAN
0 1	$1 \leq E \leq 2046$	xxx...x	Implicit Normalized Number
0 1	0	$M \neq 0$	Denormal number



Ripple carry Adder

→ It is used to add 2no. of n bit data but it takes longer time, since the carry signal is rippled from lower adder to its next higher adder.
 It performs addition operation at slower rate.

$$A = \begin{array}{c} \text{MSD} \\ A_3 \ A_2 \ A_1 \ A_0 \end{array} \quad \begin{array}{c} \text{FA}, \text{FA} \\ \hline \text{LSB} \end{array}$$

$$B = \begin{array}{c} B_3 \ B_2 \ B_1 \ B_0 \end{array} \quad \begin{array}{c} \text{HA}_0 \\ \hline \end{array}$$

$$+ \begin{array}{c} C_3 \ C_2 \ C_1 \ \vdots \\ \hline C \end{array}$$

$$\begin{array}{c} S_3 \ S_2 \ S_1 \ S_0 \\ \hline \end{array}$$

final Result =

$$C_4 \left(\begin{array}{c} S_3 \ S_2 \ S_1 \ S_0 \\ \hline \end{array} \right)$$

\downarrow

SUM

carry

ACE
Since 1975

$$\begin{array}{c} \text{FA} \ | \ \text{FA} \ | \ \text{FA} \ | \ \text{I} \ | \ \text{I} \\ + \ | \ | \ | \ 1 \ | \ | \\ \hline \text{I} \ | \ \text{I} \ | \ \text{I} \ | \ 0 \end{array} = A$$

$$= B$$

$$\text{HA}$$

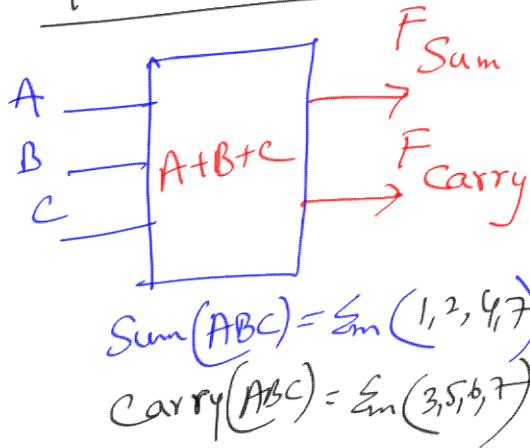
$n=4$
 $\text{HA}=1$
 $\text{FA}=3$

→ Let there is $C_{in} = \text{initial}$
carry for LSB input; then
LSB adder is also
Full adder

- 1Q) To add 2no. of 16 bit data
(without C_{in}), the no. of HA required is 1 and no. of FA required is 15
- 2Q) To add 2no. of n bit data (with C_{in})
the no. of HA required is 0
and no. of FA required is n



Full Adder



	ABC	F _{Sum}	F _{CY}
0	000	0	0
1	001	1	0
2	010	1	0
3	011	0	1
4	100	1	0
5	101	0	1
6	110	0	1
7	111	1	1



$$F_{\text{Sum}}(A, B, C) = \Sigma m(1, 2, 4, 7)$$

$$\bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

$$\overline{P} (B \oplus C) + A (\overline{B} \oplus \overline{C}) = P \oplus Q = A \oplus B \oplus C$$

$$F_{\text{CY}}(A, B, C) = \Sigma m(3, 5, 6, 7)$$

$$= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$C(A \oplus B) + AB$$



$$\text{Sum } (A \oplus B) \oplus C$$

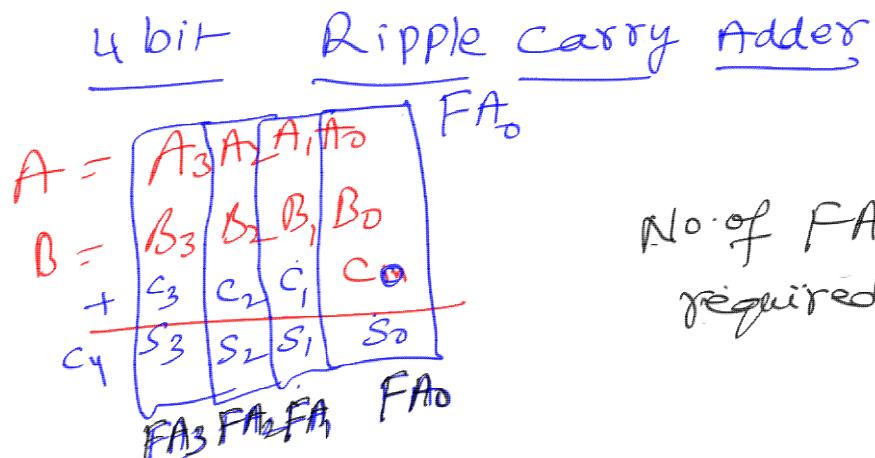
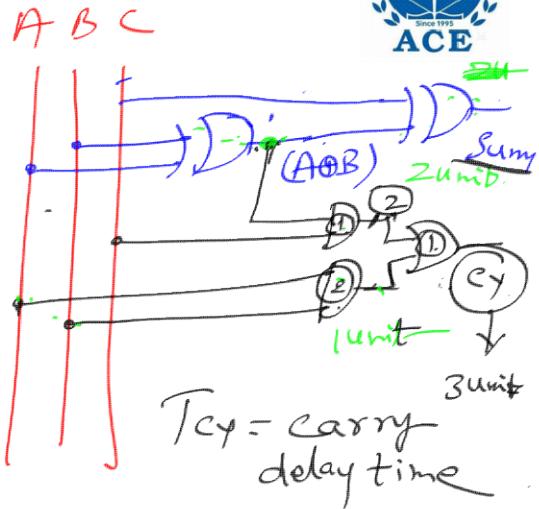
$$C_y (ABC) = C \cdot (A \oplus B) + AB$$

$T_{XOR} = 1 \text{ unit time}$

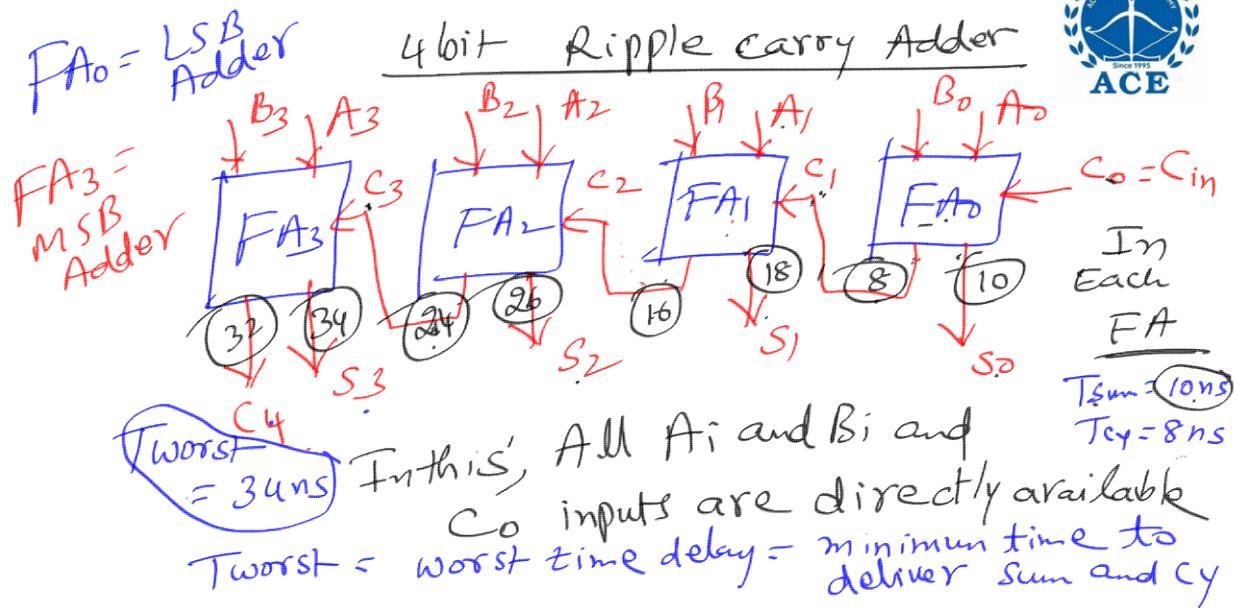
$T_{AND} = 1 \text{ unit}$

$T_{OR} = 1 \text{ unit}$

$T_{sum} = \frac{\text{Sum delay time}}{3}$

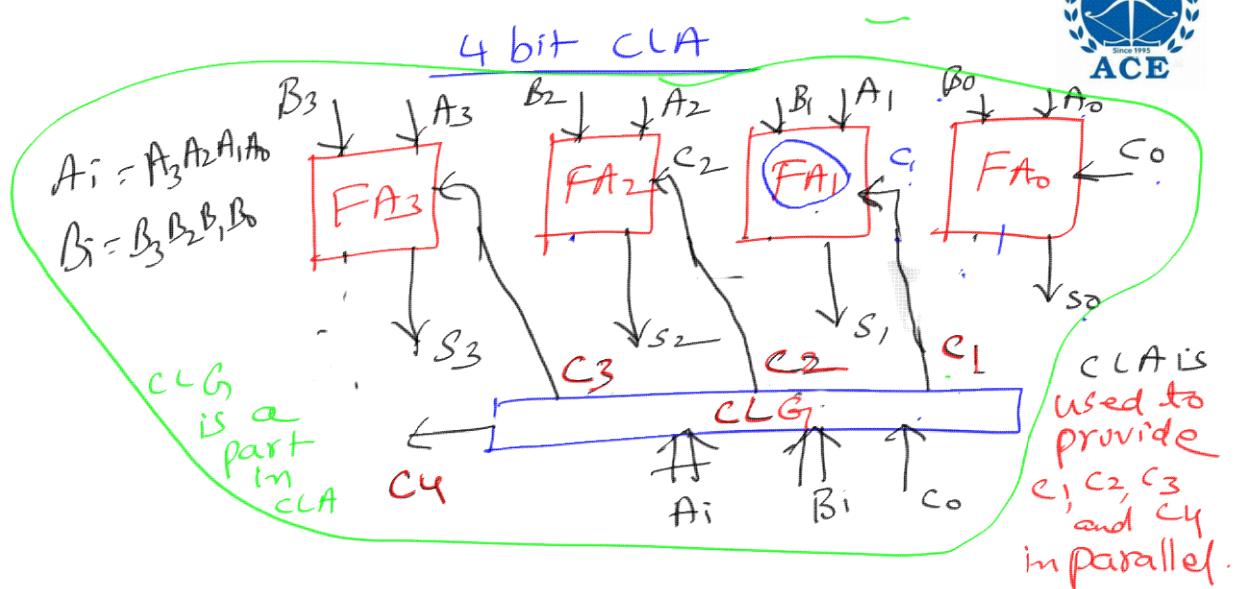


No. of FA required = 4



carry Lookahead Generator (CLG) and carry Lookahead Adder (CLA)

→ CLG is used to supply instant carry signals from external source to all Full Adders, hence it reduces the Ripple carry signal propagation delay.



- CLA is used to perform perfect parallel Addition.
- CLA is used to generate instant Carry Signal in parallel
- CLG ckt design is costlier but it is faster
- Generally 2 level circuit is used to design CLG circuit with multiple inputs AND gate OR gate

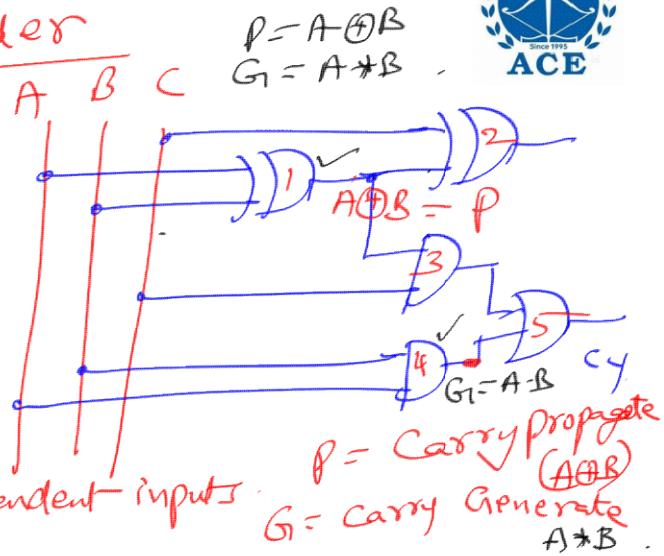


Full Adder

$$S = A \oplus B \oplus C$$

$$C_y = C(A \oplus B) + AB$$

only gate ① and gate ④ are having independent inputs



$$\text{Sum} = (A \oplus B) \oplus C = P \oplus C$$

$$\text{Carry} = C(A \oplus B) + AB = CP + G_1 = PC + G_1$$

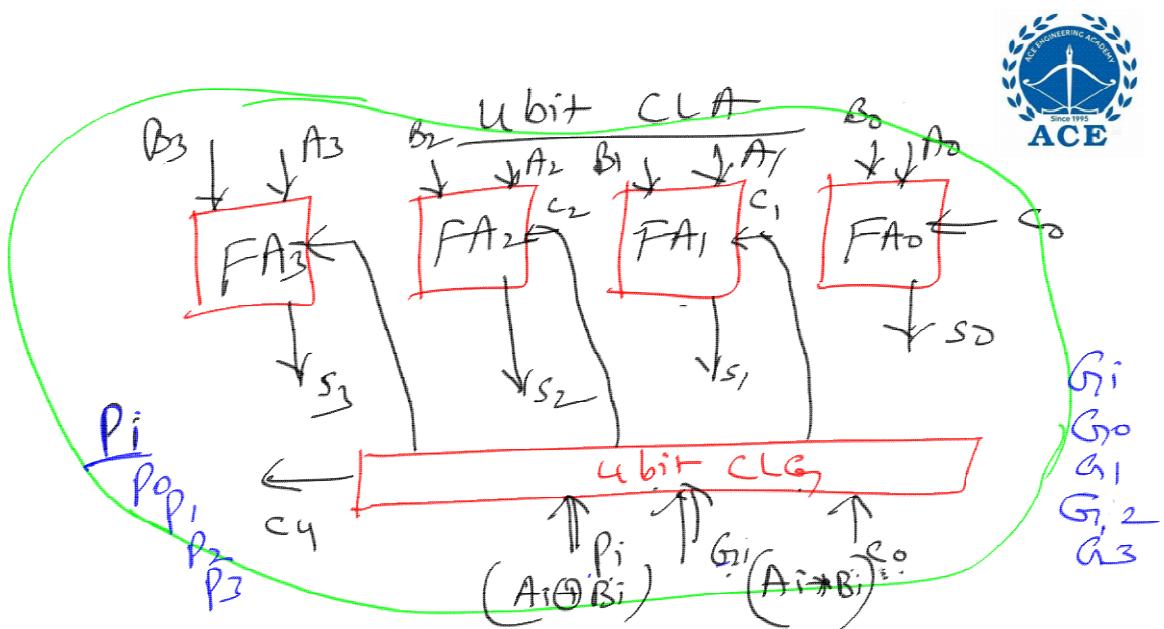
→ 4 bit CLA is aimed to provide the result $C_4 S_3 S_2 S_1 S_0$

→ CLA is aimed to generate C_1, C_2, C_3 and C_4 .

$S_0 = P_0 \oplus C_0$
$S_1 = P_1 \oplus C_1$
$S_2 = P_2 \oplus C_2$
$S_3 = P_3 \oplus C_3$



$C_1 = C_0 P_0 + G_0$, $C_2 = C_1 P_1 + G_1$
 $C_3 = C_2 P_2 + G_2$, $C_4 = C_3 P_3 + G_3$
 → Since CLG permits only P_i , G_i and C_0
 it is not possible to use other
 carry signals in C_2, C_3, C_4 except C_0





$$P_3 = A_3 \oplus B_3$$

$$P_2 = A_2 \oplus B_2$$

$$P_1 = A_1 \oplus B_1$$

$$P_0 = A_0 \oplus B_0$$

$$P_i = A_i \oplus B_i$$

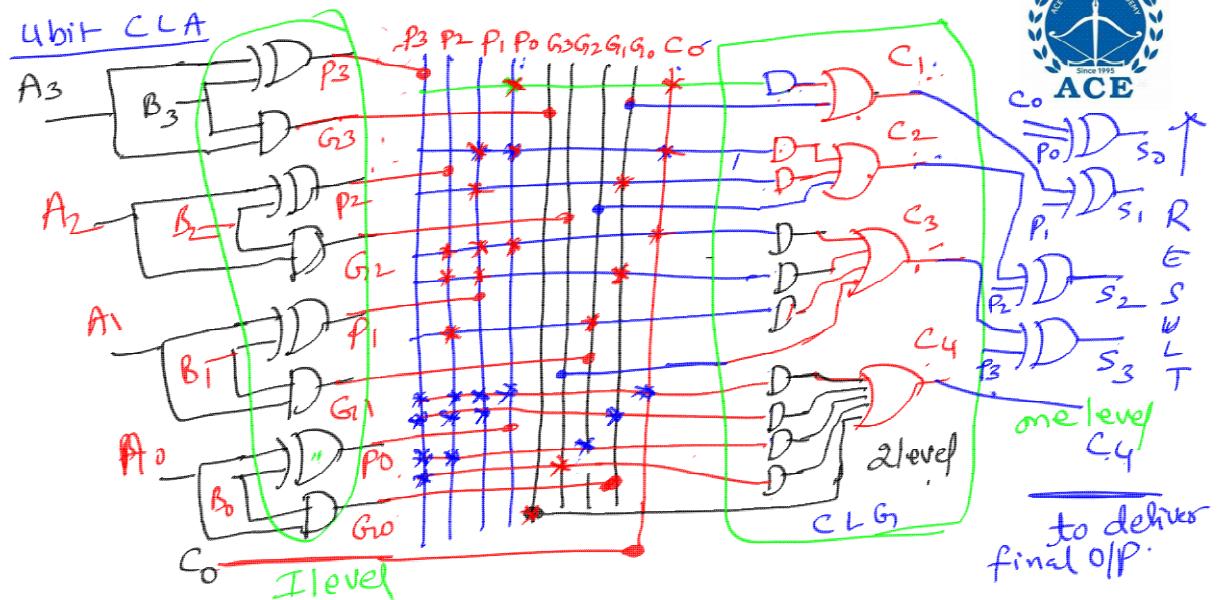
$$G_3 = A_3 * B_3$$

$$G_2 = A_2 * B_2$$

$$G_1 = A_1 * B_1$$

$$G_0 = A_0 * B_0$$

$$G_i = A_i * B_i$$





When P_i , G_i and C_0 are available

for n bit CLG;

	<u>No. of AND</u>	<u>OR gates</u>	
for C_1	1	1	$\frac{n(n+1)}{2}$
for C_2	2	-	
for C_3	3	1	sum of Natural numbers
for C_4	4	1	
<hr/>		10	4
for C_n	n	1	

$$\text{Total no. of AND gates} = \frac{n(n+1)}{2}$$

To design n bit CLG, when P_i , G_i and C_0 are available



1) No. of AND gates required = $\frac{n(n+1)}{2}$

2) No. of OR gates required = n .



$$C_1 = (P_0 C_0 + G_{10})$$

$$C_2 = P_1 C_1 + G_{11} = (P_1 P_0 C_0 + P_1 G_{10} + G_{11}) \checkmark$$

$$C_3 = P_2 C_2 + G_{12} = (P_2 P_1 P_0 C_0 + P_2 P_1 G_{10} + P_2 G_{11} + G_{12})$$

$$C_4 = P_3 C_3 + G_3 = \underline{P_3 P_2 P_1 P_0 C_0} + \underline{P_3 P_2 P_1 G_{10}} +$$

$$\underline{P_3 P_2 G_{11}} + \underline{P_3 G_{12}} + G_3$$

→ Generally CLG is designed with
2 level CK — with multiple
input gates

Fan-in: no. of inputs permitted
by the gate



Generally Fan-in is infinite for the
AND gates and OR gates which
are used in CLG design



- only one gate delay time is sufficient to get all P_i and G_{i1}
- CLG requires 2 no. of tpd's (2 level) after receiving P_i and G_{i1}
- To generate the final O/P from XOR gates, one more level is needed
Hence total levelings (tpd)
 $\frac{P_i G_{i1}}{1} \quad \text{CLG} \quad \frac{\text{O/P}}{2} = 4 \text{ level}$