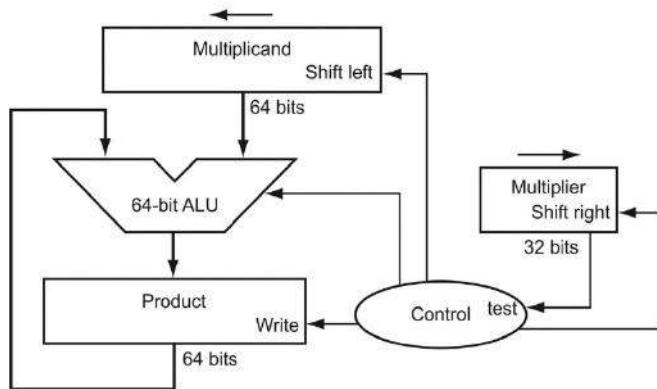
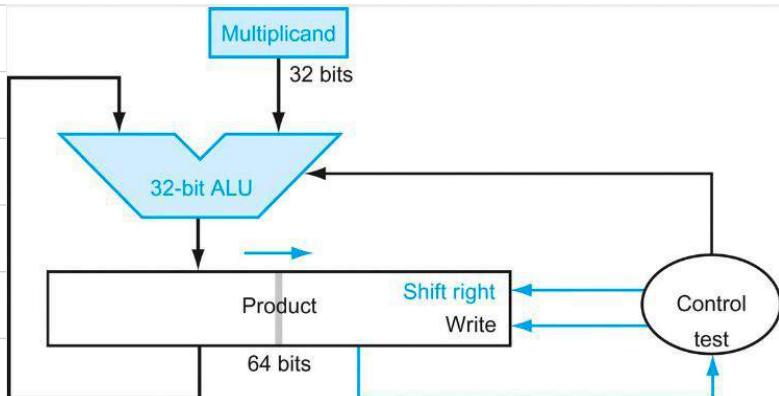


## ASSIGNMENT - 3 SOLUTION

3.14) Time necessary to perform multiply using following approach:



(a) Multiplication Hardware [1]



(b) Multiply example [1]

Given :-

- Integer is 8-bit wide.
- Each step of operation take 4 time units.
- Assume :
  - step 1a: Addition always performed.  
Either with multiplicand or zero.
  - registers have already been initialized.
- Hardware : shifts of multiplicand and multiplier can be done simultaneously.
- Software : Multiplicand and multiplier shifts are done one after another.

Solve for each case.

Solution :-

### Case 1: Hardware

From figure (a), we can see that,

To perform multiplication, the following need to be done for each bit:

1. Addition
2. Shift (multiplier & multiplicand done simultaneously)
3. Decide if complete.

⇒ 3 operations steps are needed for a single bit  
(Note however shift may be considered as 2 different operations when they cannot be done simultaneously.)

We are given that integer is 8-bit wide

⇒ For 8-bits, total number of operations =  $8 \times 3$   
= 24

For 1 operation we are given that 4 time units are required.

→ For 24 operations, we need  $24 \times 4$  time units  
= 96 time units

For hardware, 96 time units are required to perform multiplication.

### Case 2: Software:

For multiplication, each of the following operations are performed for every bit:

1. Decide the component to add
2. Addition
3. Shift multiplicand
4. Shift multiplier
5. Decide if complete

⇒ 5 operations are needed for every bit.

⇒ For 8-bit integer,  $5 \times 8 = 40$  operations are performed.

Since each operation step requires 4 time units

⇒  $4 \times 40 = 120$  time units are required.

For software, 120 time units are required to perform multiplication.

Answer: Time necessary to perform multiply

(i) For hardware : 96 time units

(ii) For software : 120 time units

3.15) Time necessary to perform multiply ?

Given:-

31 adders stacked vertically approach.

8-bit wide integer  
adder takes 4 time units.

Solution:-

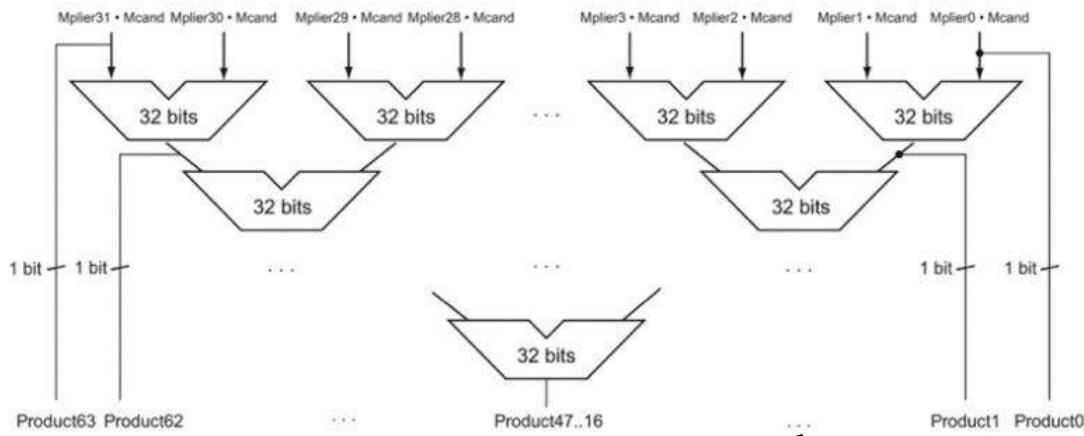
For 8-bit integer 7-adders will be stacked using the approach given in the question.

For 1 adder, we are given 4 time units are required

$\Rightarrow$  For 7 adders,  $7 \times 4 = 28$  time units are required.

Answer: Time necessary to perform multiply using approach mentioned  
 $= 28$  time units.

3.16) Time necessary to multiply using approach shown in the following figure:



(c) Fast Multiplication [1]

Given :-

8-bit wide integer

adder takes 4 time units

## Solution:

The approach involves organizing adders into a tree structure.

This approach reduces number of steps from  $n$  to  $\log_2(n)$  times, where  $n$  is the number of bits in the integer.

⇒ Only  $\log_2(8) = 3$  adder levels are required.

To pass through each adder, 4 time units are required

$$\therefore \text{3 such levels} \Rightarrow 3 \times 4 \text{ time units}$$
$$= \underline{\underline{12 \text{ time units}}}.$$

Answer: Time necessary to perform multiply using approach mentioned  
= 12 time units

B28) Calculate relative performance of adders.

Given :-

- Assume hardware corresponding to any equation containing OR or AND terms have  $p_i$  and  $g_i$  given as follows take one time unit  $T$ .

⇒ Equations for  $C_1, C_2, C_3$  and  $C_4$  take  $2T$  time units (one for AND, one for OR)

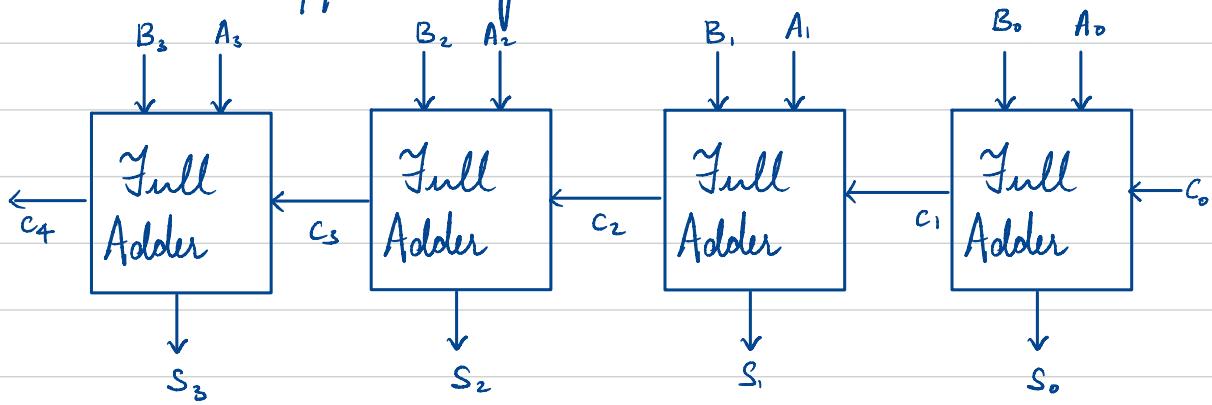
Calculate the numbers and performance ratio for 4-bit adders for both ripple carry and carry lookahead.

Given :-

- Add appropriate delay for terms in the equation governed by other equation.

Include drawing of each adder labelled with the calculated delay and the path of worst-case delay highlighted.

# Case 1 : Ripple Carry Adder (RCA)



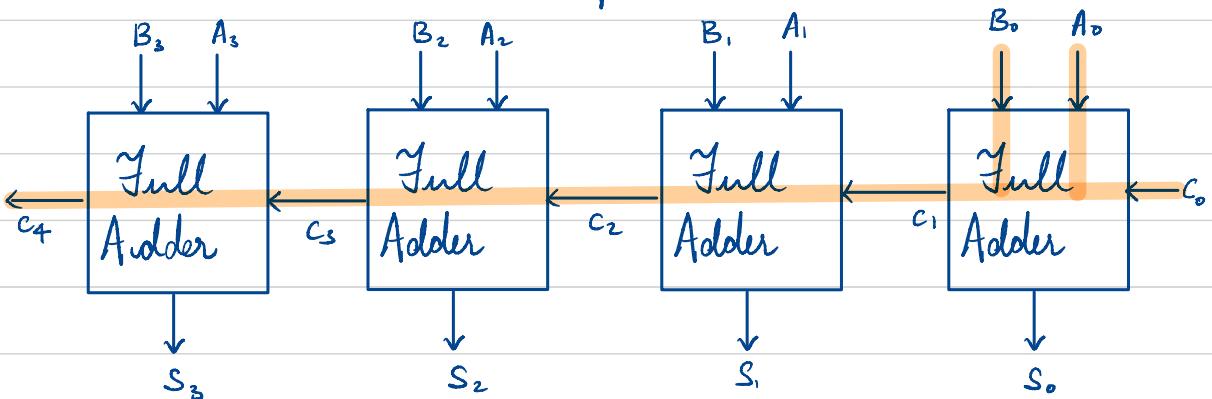
Each full adder is a 1-bit full adder  
 $\Rightarrow$  The above ripple carry adder is for 4-bits.

TERM	TIME TAKEN
$S_0 = (A_0 \oplus B_0) \oplus C_0$ $= p_0 \oplus C_0$	$T + T = 2T$
$C_0$	0
$C_1 = A_0 B_0 + B_0 C_0 + C_0 A_0$	$2T$
$S_1 = (A_1 \oplus B_1) \oplus C_1$ $= p_1 \oplus C_1$	$3T$ ( $A_1 \oplus B_1, C_1$ can be calculated in $2T$ , $1T$ for XOR)
$C_2 = A_1 B_1 + B_1 C_1 + C_1 A_1$	$T + T + 2T = 4T$ (AND) (OR) (C <sub>0</sub> )

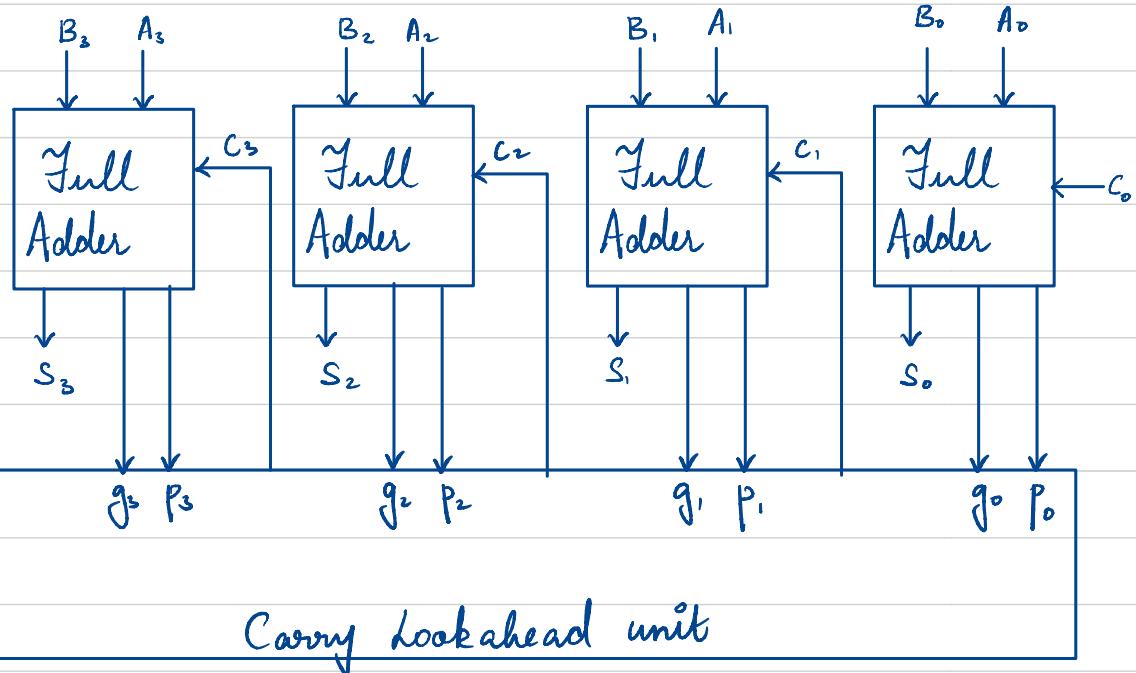
TERM	TIME TAKEN
$S_2 = (A_2 \oplus B_2) \oplus C_2$ = $P_2 \oplus C_2$	$4T + T = 5T$
$C_3 = A_2 B_2 + C_2 B_2 + C_2 A_2$	$T + T + 4T = 6T$
$S_3 = (A_3 \oplus B_3) \oplus C_3$ = $P_3 \oplus C_3$	$7T$
$C_4 = A_3 B_3 + C_3 A_3 + C_3 B_3$	$T + T + 6T = 8T$

From the table, we can see that total time taken by a 4-bit ripple carry adder is 8T time units.

Paths for worst-case delay:



## Case 2: Carry lookahead Adder (CLA)

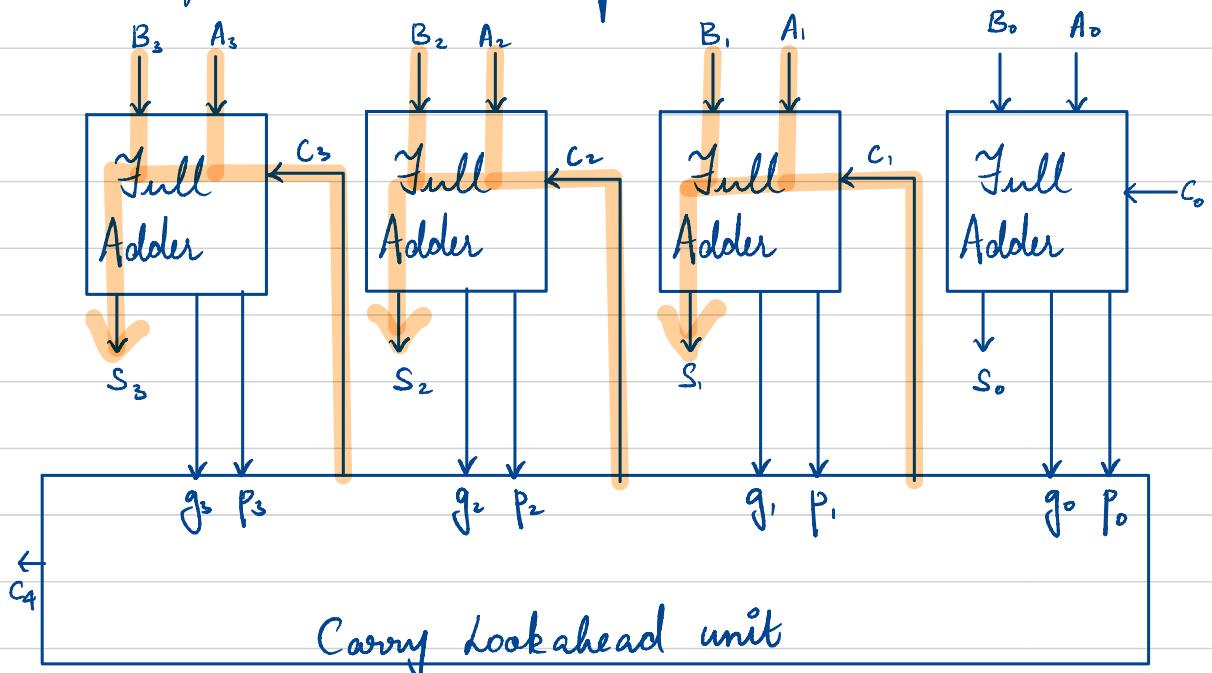


Each full adder is a 1-bit full adder  
⇒ The above carry lookahead adder is for 4-bits.

TERM	TIME TAKEN
$C_0$	0
$S_0 = p_0 \oplus C_0$	$T + T = 2T$
$C_1 = g_0 + C_0 p_0$	$T + T + T = 3T$
$S_1 = p_1 \oplus C_1$	$3T + T = 4T$
$C_2 = g_1 + g_0 p_1 + C_0 p_1 p_0$	$3T$
$S_2 = p_2 \oplus C_2$	$4T$
$C_3 = g_2 + g_1 p_2 + g_0 p_1 p_2 + C_0 p_0 p_1 p_2$	$3T$
$S_3 = p_3 \oplus C_3$	$4T$
$C_4 = g_3 + g_2 p_3 + g_1 p_3 p_2 + g_0 p_1 p_2 p_3 + C_0 p_0 p_1 p_2 p_3$	$3T$

From the table, we can see that total time taken by a 4-bit carry lookahead adder is  $4T$  time units.

# Paths for worst-case delay



## Relative performance

$$\frac{\text{Total time taken by RCA}}{\text{Total time taken by CLA}} = \frac{8T}{4T} = 2$$

⇒ Carry lookahead adder is 2 times faster.

Answer: Relative performance of adder:

CLA is 2 times faster than RCA

Time for CLA =  $4T$

Time for RCA =  $8T$

Performance Ratio: CLA : RCA = 1 : 2

(Time)

B29) Relative speed of 16-bit adders using:

1. Ripple carry only
2. Ripple carry of 4-bit groups that use carry lookahead
3. Carry lookahead.

Solution:

Case 1: Ripple Carry

For a ripple carry adder, each bit takes  $2T$  time units.

$\Rightarrow$  16-bit adder takes  $16 \times 2T = 32T$  time units.  
( $\because$  Cout of RCA is the longest)

Case 2: Ripple carry of 4-bit groups that use carry lookahead

If 4-bits are grouped together into one group, 16-bits will be grouped into 4 groups.

Any individual generate and propagate for ripple carry adder will take  $1T$  time.

For the first block carry  $C_0$  is available at time  $0$ .  
For the second block carry  $C_4$  is available at time  $3T$ .  
Now to calculate  $C_{12}$  of second CLA block, it takes additional  $2T$ .

$$\text{i.e., } C_8 = g_7 + g_6 p_7 + g_5 p_1 p_6 + g_4 p_1 p_6 p_5 + C_4 p_1 p_6 p_5 p_4$$

$\downarrow$   
 $2T \text{ for AND \& OR.}$

$3T$  for  $C_4, p_i, g_i$ .

$\Rightarrow C_8$  is calculated at  $5T$ .

Similarly for  $C_{12}$  takes  $7T$ ,  $C_{16}$  takes  $9T$ .

$C_{12}$  takes  $7T \Rightarrow C_{15}$  takes  $9T$

$$\Rightarrow S_{15} = p_{15} \oplus C_{15}$$

$9T + 1T \text{ (wor)} \\ (C_{15})$

$$\Rightarrow 10T$$

$\Rightarrow$  Total time delay for ripple carry adder of 4-bits groups that use carry lookahead is  $10T$ .

Case 3: Carry lookahead adder:

All the generates and propagates take  $T$  time units.

All carry are product of carry-propagate or generate-propagate terms.

$\therefore$  OR takes  $1T$ , AND takes  $1T$

$$\Rightarrow C_i^o = T + \underset{\substack{\text{(generate OR)} \\ \text{($g_i, p_i$)}}}{T} + \underset{\text{(AND)}}{T} = 3T$$

$$S_i^o = P_i^o \oplus C_i^o = 4T$$

$\Rightarrow$  Total time delay for carry look ahead adder  
is  $4T$ .

### Relative speeds

Time for

$$\text{RCA : Ripple Carry with CLA : CLA} = 32 : 10 : 4 \\ = \underline{16 : 5 : 2}$$

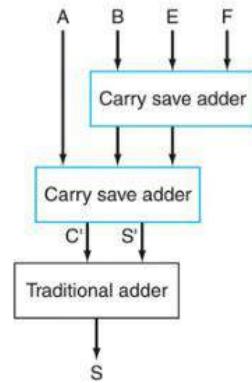
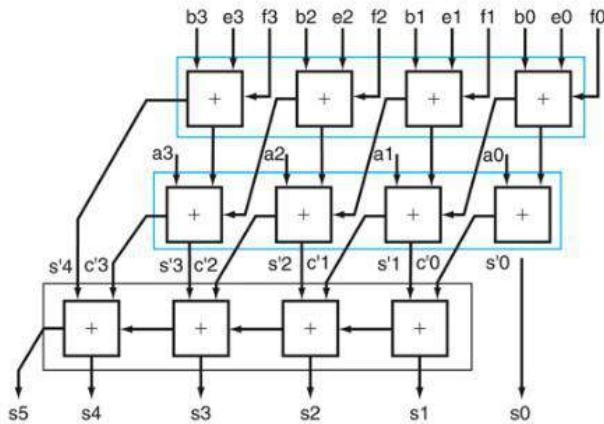
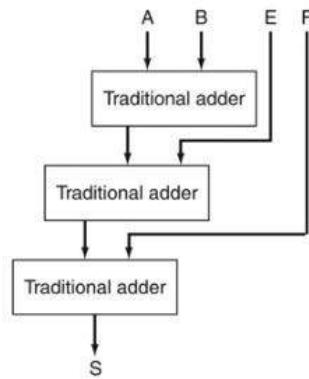
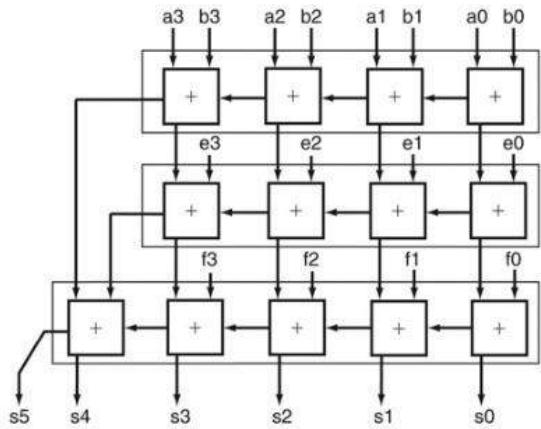
Relative speeds is inverse of time :  $\frac{1}{16} : \frac{1}{5} : \frac{1}{2}$

$$\text{RCA : Ripple with CLA : CLA} \equiv \underline{5 : 16 : 40}$$

B31) Think of adder as product of as a hardware device that can add  $(a_i^i, b_i^i, c_i)$  and produce two outputs  $(S, C_i + 1)$ .

Form  $S'$  and  $C'$ . At the end add  $C'$  and  $S'$  like carrying same addition.

Calculate delay to add four 16-bit numbers using full carry-look ahead adders versus carry save with a carry look ahead adder forming final sum.



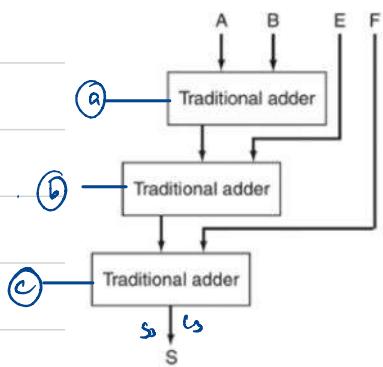
Solution:

If time units are same as that of question B28,  
 $\Rightarrow$  Time delay for carry lookahead adder is  $4T$ .

$\therefore$  We are assuming  $g_i$  &  $p_i$  are formed in IT.

At 1T, wall  $g_i$  and  $p_i$  are formed for  $i=0, \dots, 15$ .

## Case 1 : Full Cover - look ahead reader



For each carry look ahead  
welder, sum takes 4T,  
carry takes 3T.

For @,  $S_a$  is produced at 4T  
of which  $S_{a_0}$  is produced at 2T.  
Others at 4T.

For (b)

$C_0, E$  are available at 0.

$S_{a0}$  is available at 2T.

$\Rightarrow p_0, g_0$  are available at 3T

$\Rightarrow C_1$  is available at 5T.

By 4T, all of  $S_a$  are available to (b)

$\Rightarrow$  At 5T  $p_1, g_1$  are found.

At 6T,  $S_{b1}$  is available

At 5T,  $p_2, g_2, p_3, g_3$  are available to (b)

$\Rightarrow$  At 7T,  $C_2, C_3, C_4$  are available

$\Rightarrow S_3$  is found at 8T

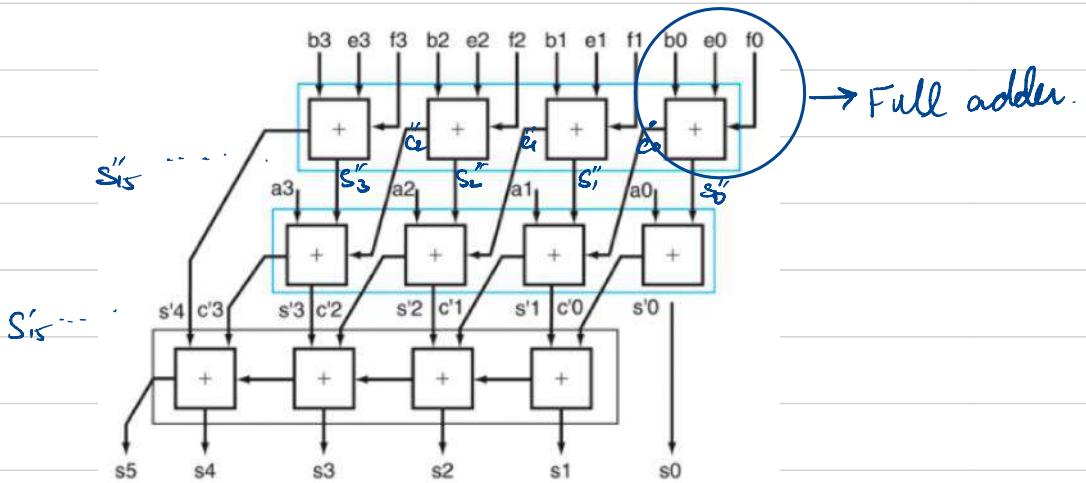
$\Rightarrow$  (b) takes another 4T.

Similarly C takes 4T

$\Rightarrow$  Total time taken = 12T

## Case 2: Carry Save Adder with CLA

For every Full adder in CSA, it takes T units to produce  $s_i$  &  $c_{i+1}$ .



- ⇒ For all,  $s_5 \dots s_0$ , takes 1T to be calculated  
lly  $c_5 \dots c_0$ , takes 1T to be calculated.
- ⇒  $s'_0 \dots s'_5$ , are calculated at 2T.  
lly  $c'_0 \dots c'_5$  are calculated at 2T.

For the last CLA unit, it takes another 3T to calculate  $c_{16}$ .

$\Rightarrow$  At  $6T$ , we get  $S_{15}$ .

$\therefore C''_{15}, C'_{15}, C_{16}$  are available by  $5T$ ,  $S_{16}, S_{17}$  are calculated by  $6T$ .

$\Rightarrow$  Total time delay is  $6T$ .

Answer :  $CLA = 12T$

CSA with  $CLA = 6T$