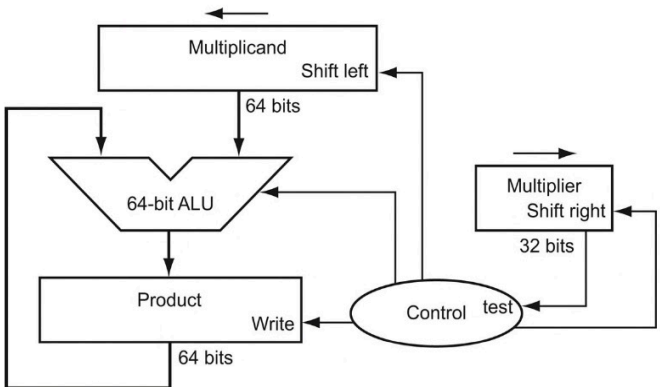
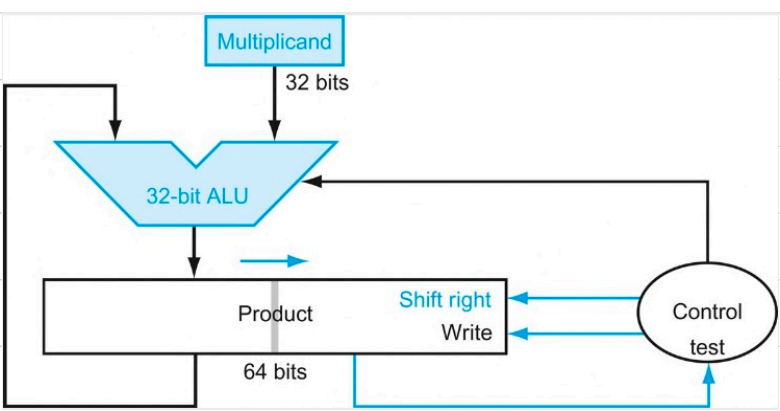
ASSIGNMENT-3 SOLUTION

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 × 3

= 24

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

⇒ For 24 operations, we need 24×4 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

⇒ $4 \times 40 = 160$ time units are required.

For software, 160 time units are required to perform multiplication.

Answer: Time necessary to perform multiply

(i) For hardware: 96 time units

(ii) For software: 160 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:-

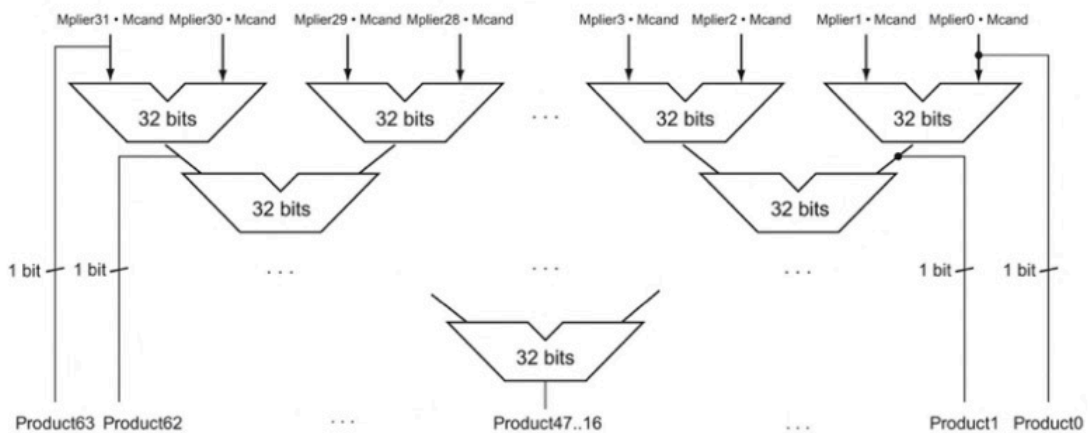
For 8-bit integer 7-adders will be stacked using the approach given in the question.

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

⇒ 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.

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

To pass through each adder, 4 time units are required

$\therefore$ 3 such levels $\Rightarrow$ 3 x 4 time units
= 12 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 C1, C2, C3 and C4 take 2T time units (one for AND, one for OR)

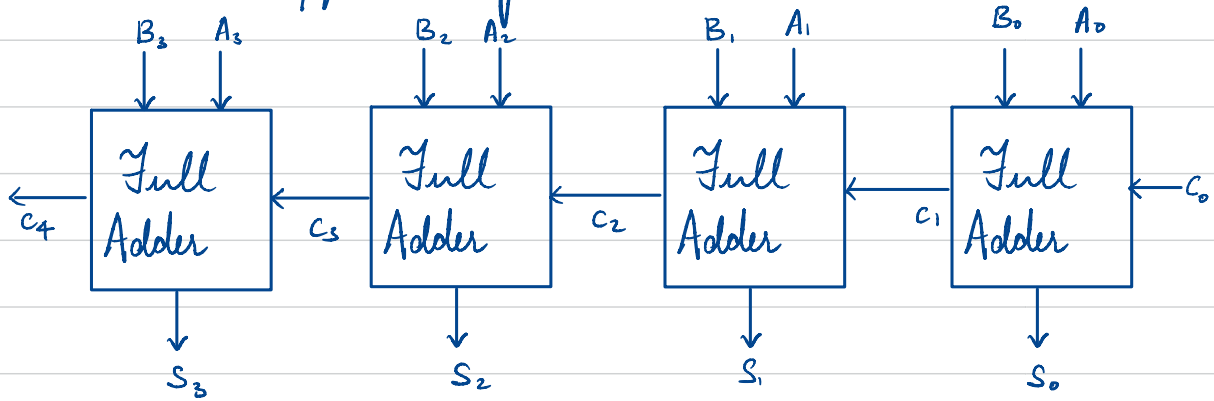
Calculate the numbers and performance ratio for 4-bit adders for both ripple carry and carry lookahead.

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)



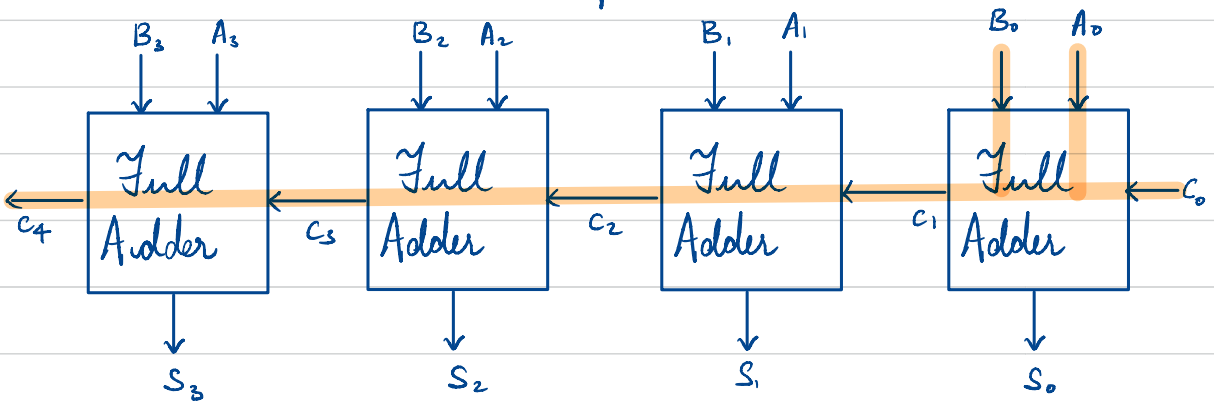
Each full adder is a 1-bit full adder
⇒ The above ripple carry adder is for 4-bits.

TERM	TIME TAKEN
$S_0 = (A_0 \oplus B_0) \oplus C_0$ $= p_0 \oplus C_0$	$T + T = 2T$
C_0	0
$C_1 = A_0B_0 + B_0C_0 + C_0A_0$	$2T$
$S_1 = (A_1 \oplus B_1) \oplus C_1$ $= p_1 \oplus C_1$	$3T$ ($A_1 \oplus B_1$, C_1 can be calculated in $2T$, $1T$ for XOR)
$C_2 = A_1B_1 + B_1C_1 + C_1A_1$	$T + T + 2T = 4T$ (AND) (OR) (C1)

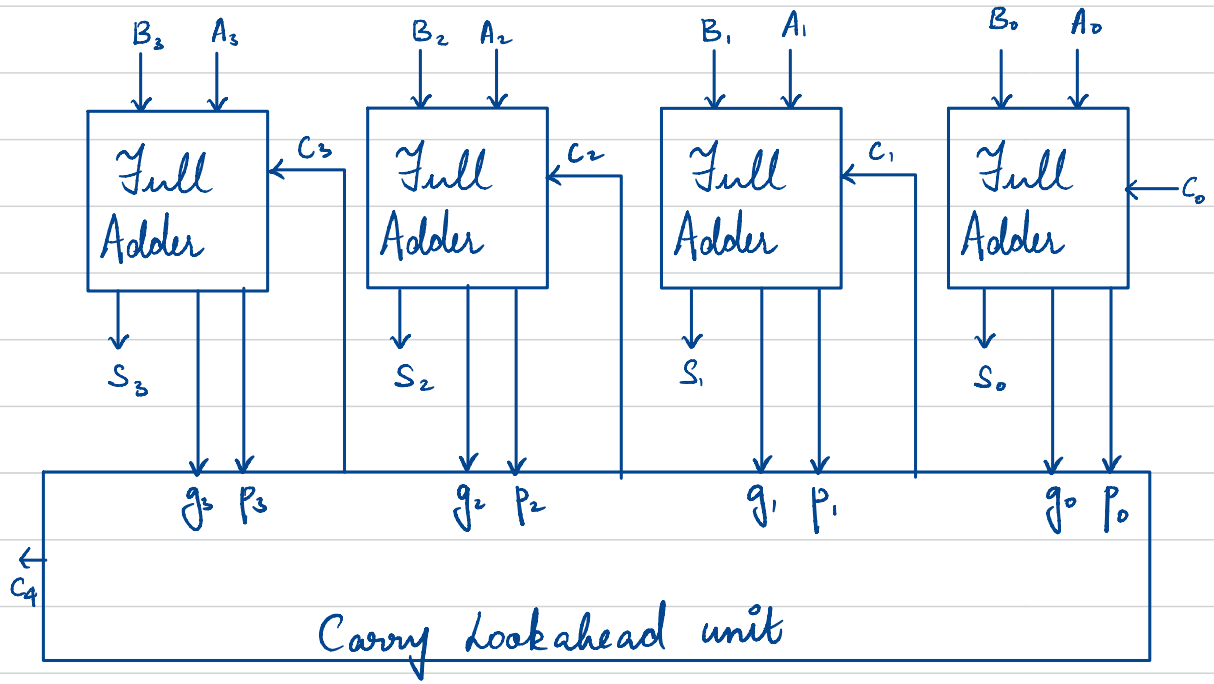
TERM	TIME TAKEN
$S_2 = (A_2 \oplus B_2) \oplus C_2$ $= p_2 \oplus C_2$	$4T + T = 5T$
$C_3 = A_2B_2 + C_2B_2 + C_2A_2$	$T + T + 4T = 6T$
$S_3 = (A_3 \oplus B_3) \oplus C_3$ $= p_3 \oplus C_3$	$7T$
$C_4 = A_3B_3 + C_3A_3 + C_3B_3$	$T + T + 6T = 8T$

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)



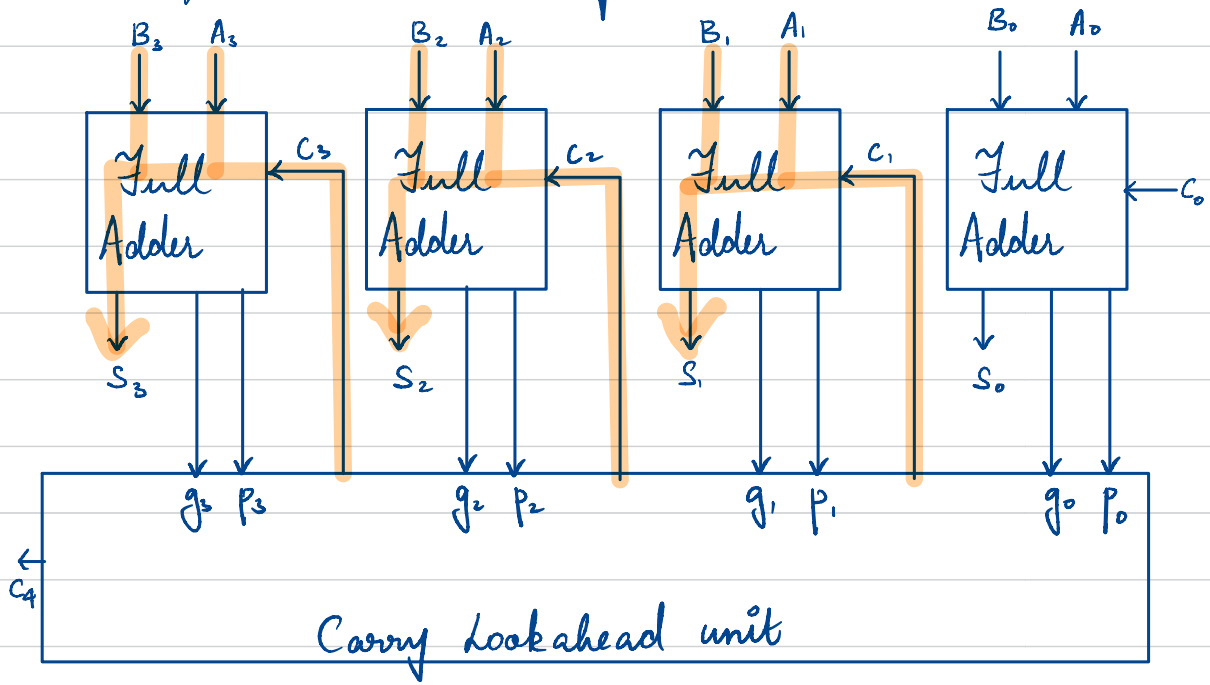
Each full adder is a 1-bit full adder

⇒ 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

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

$\Rightarrow$ 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.

⇒ 16-bit adder takes $16 \times 2T = 32T$ time units.
(C: 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 Cout of second CLA block, it takes additional 2T.

i.e., $C_8 = g_7 + g_6p_7 + g_5p_7p_6 + g_4p_7p_6p_5 + C_4p_7p_6p_5p_4$

↓

2T 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 (to OR)

(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.

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

$\Rightarrow C_i = T + T + T = 3T$

(generate g_i, p_i) (OR) (AND)

$S_i = P_i \oplus C_i = 4T$

$\Rightarrow$ Total time delay for carry lookahead adder is $\underline{\underline{4T}}$.

Relative speeds

Time for

RCA : Ripple Carry with CLA : CLA = 32 : 10 : 4

= 16 : 5 : 2

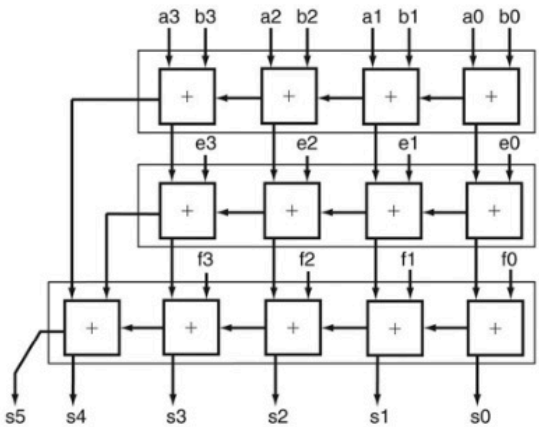
Relative speeds is inverse of time : $\frac{1}{16} : \frac{1}{5} : \frac{1}{2}$

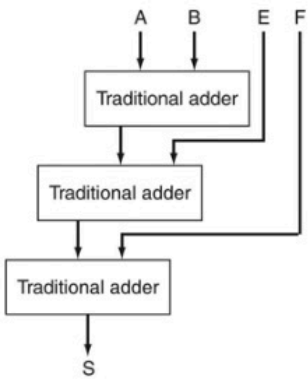
RCA : Ripple with CLA : CLA $\equiv$ 5 : 16 : 40

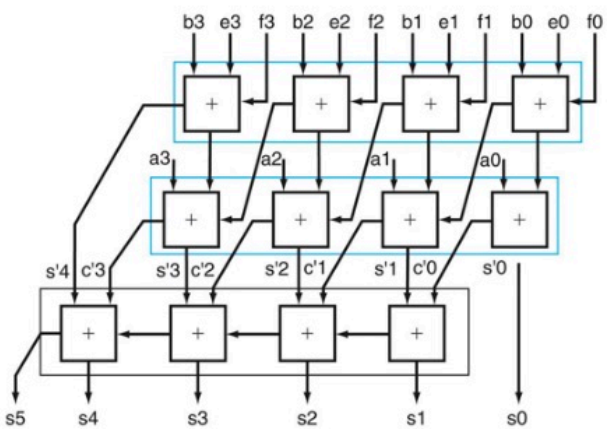
B 31) Think of adder as product of as a hardware device that can add (a_i, b_i, c_i) and produce two outputs $(S, C_i + 1)$.

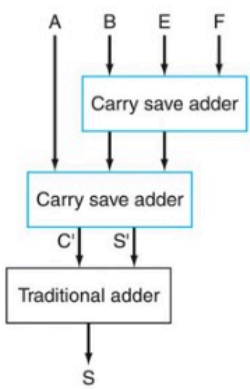
Form S' and C'. At the end add C' and S'. like carry save addition.

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.


a3 b3 a2 b2 a1 b1 a0 b0
e3 e2 e1 e0
f3 f2 f1 f0
s5 s4 s3 s2 s1 s0


A B E F
Traditional adder
Traditional adder
Traditional adder
S


b3 e3 f3 b2 e2 f2 b1 e1 f1 b0 e0 f0
a3 a2 a1 a0
s'4 c'3 s'3 c'2 s'2 c'1 s'1 c'0 s'0
s5 s4 s3 s2 s1 s0


A B E F
Carry save adder
Carry save adder
C' S'
Traditional adder
S

Solution:

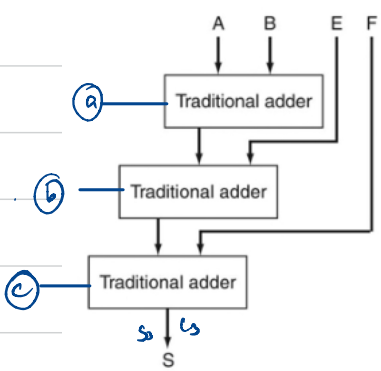
If time units are same as that of question B28,

⇒ Time delay for carry lookahead adder is 4T.

∵ We are assuming g_i & p_i are formed in 1T.

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

Case 1: Full Carry – look ahead adder



For each carry look ahead adder, 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_{a_0} 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_{b_1} 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

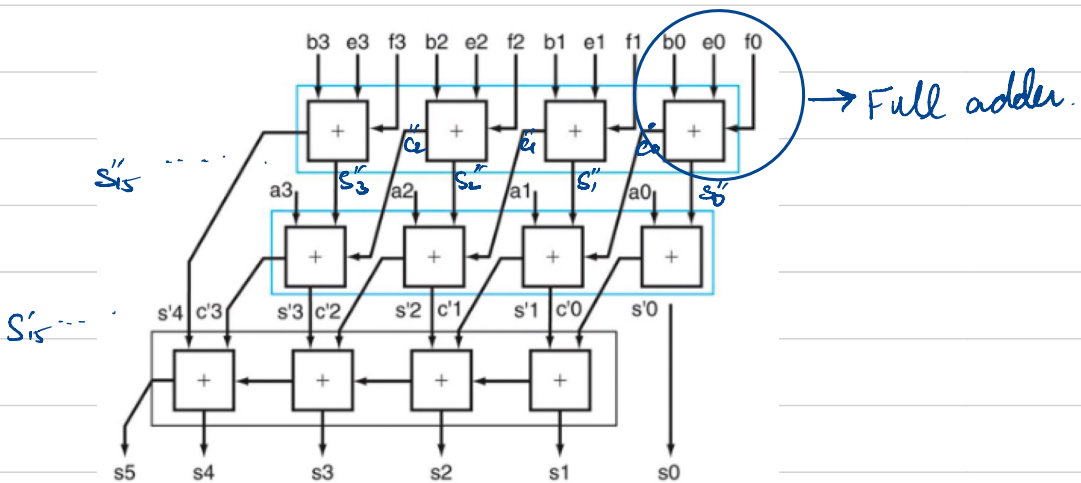
$\Rightarrow$ (b) takes another 4T.

Similarly C takes 4T

$\Rightarrow$ Total time taken = 12T

Case 2: Carry Save Adder with CLA. (CSA)

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



⇒ For all, $S^0_0 \ldots S''_{15}$, takes 2T to be calculated
Illy $C^0_0, \ldots C'_{16}$, takes 2T to be calculated.

⇒ $S'_0 \ldots S'_{15}$, are calculated at 4T.
Illy $C'_0 \ldots C'_{16}$ are calculated at 4T.

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

$\Rightarrow$ At 8T, we get S_{15}.

$\because$ C''_{15}, C'_{15}, C_{16} are available by 7T, S_{16}, S_{17} are calculated by 8T.

$\Rightarrow$ Total time delay is 8T.

Answer : CLA = 12T

CSA with CLA = 8T