

Abhishek Gupta

Roll No.01

MCA Sem- IV

Parallel & Distributed Computing
Assignment.

Ans. for implementing instructions or block of instructions in parallel, it should be guaranteed that the instructions are independent of each other.

These instructions can be

- (i) Control dependent
- (ii) Resource dependent
- (iii) Data dependent

But we only consider data dependency to check parallelism.

Bernstein condition rely on subsequent two set of variables :-

- (i) Input set (I_1) for process P_1 ,
- (ii) Output set (O_1) for process P_1 .

following are Bernstein conditions :-
for two instructions P_1 & P_2

$$(i) I_1 \cap O_2 = \emptyset$$

$$(ii) I_2 \cap O_1 = \emptyset$$

$$(iii) O_1 \cap O_2 = \emptyset$$

If above three conditions satisfies, processes P_1 & P_2 are parallel.

Bernstein conditions in following example:

	I_i	O_i
$P_1 : A = B \times C$	$\{B, C\}$	$\{A\}$
$P_2 : C = D + E$	$\{D, E\}$	$\{C\}$
$P_3 : C = A + B$	$\{A, B\}$	$\{C\}$
$P_4 : E = F - D$	$\{F, D\}$	$\{E\}$
$P_5 : H = I + J$	$\{I, J\}$	$\{H\}$

Total no. of combinations = ${}^5C_2 = 10$

(i) P_1 & P_2 :-

$$I_1 \cap O_2 = \{C\} \neq \emptyset$$

→ Not parallel

(ii) P_1 & P_3 :-

$$I_1 \cap O_3 = \{C\} \neq \emptyset$$

→ Not parallel

(iii) P_1 & P_4 :-

$$I_1 \cap O_4 = \emptyset$$

$$I_4 \cap O_1 = \emptyset$$

$$O_1 \cap O_4 = \emptyset$$

→ $P_1 \parallel P_4$

(iv) P_1 & P_5 :-

$$I_1 \cap O_5 = \emptyset$$

$$I_5 \cap O_1 = \emptyset$$

$$O_1 \cap O_5 = \emptyset$$

→ $P_1 \parallel P_5$

(V) $P_2 \not\parallel P_3$

$$I_2 \cap O_3 = \emptyset$$

$$I_3 \cap O_2 = \emptyset$$

$$O_2 \cap O_3 = \{C\}$$

$\rightarrow P_2$ not parallel to P_3

(VI) $P_2 \not\parallel P_4$

$$I_2 \cap O_4 = \{E\} \neq \emptyset$$

$\rightarrow P_2$ not parallel to P_4

(VII) $P_2 \not\parallel P_5$

$$I_2 \cap O_5 = \emptyset$$

$$I_5 \cap O_2 = \emptyset$$

$$O_2 \cap O_5 = \emptyset$$

$\rightarrow P_2 \perp\!\!\! \perp P_5$

(VIII) $P_3 \not\parallel P_4$

$$I_3 \cap O_4 = \emptyset$$

$$I_4 \cap O_3 = \emptyset$$

$$O_3 \cap O_4 = \emptyset$$

$\rightarrow P_3$ not parallel P_4

(ix) $P_3 \& P_5$

$$I_3 \cap O_5 = \emptyset$$

$$O_3 \cap I_5 = \emptyset$$

$$O_3 \cap O_5 = \emptyset$$

$\rightarrow P_3 \amalg P_5$

(x) $P_4 \& P_5$

$$I_4 \cap O_5 = \emptyset$$

$$I_5 \cap O_4 = \emptyset$$

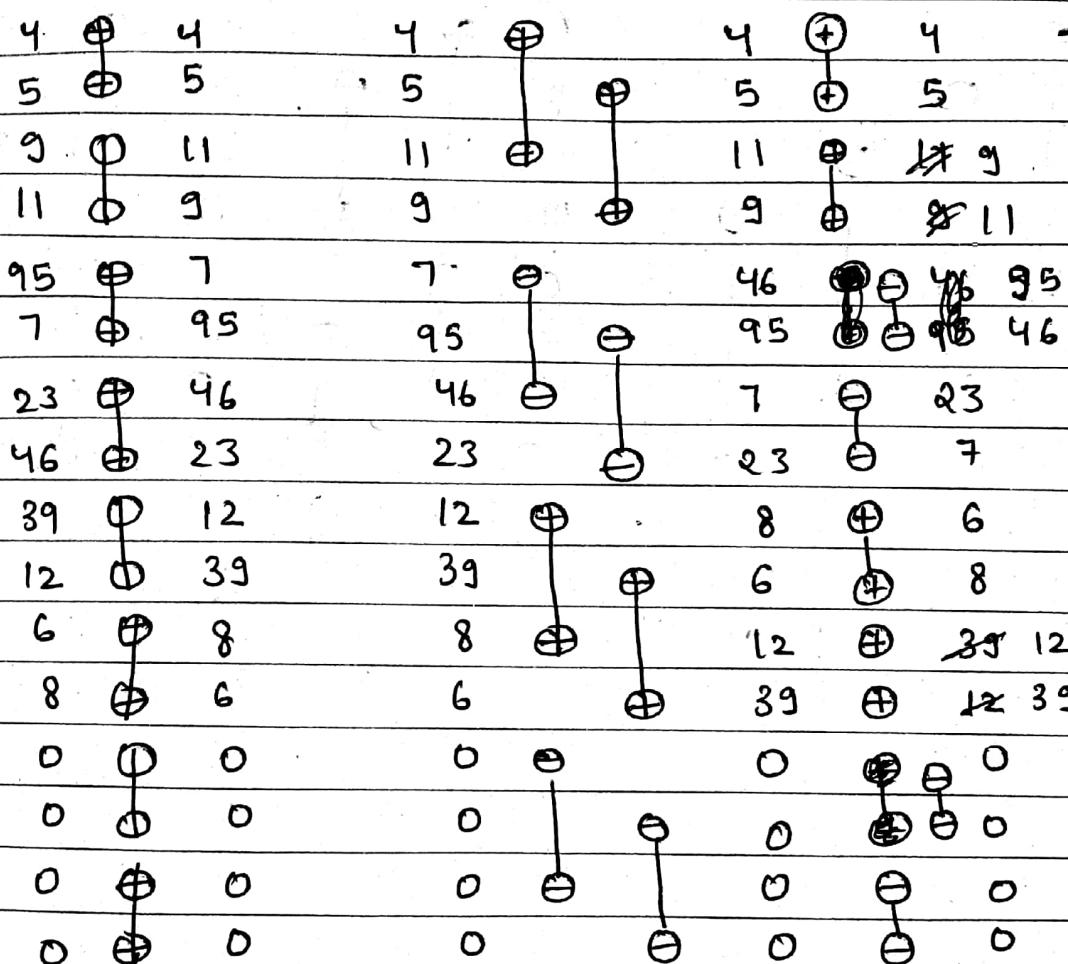
$$O_4 \cap O_5 = \emptyset$$

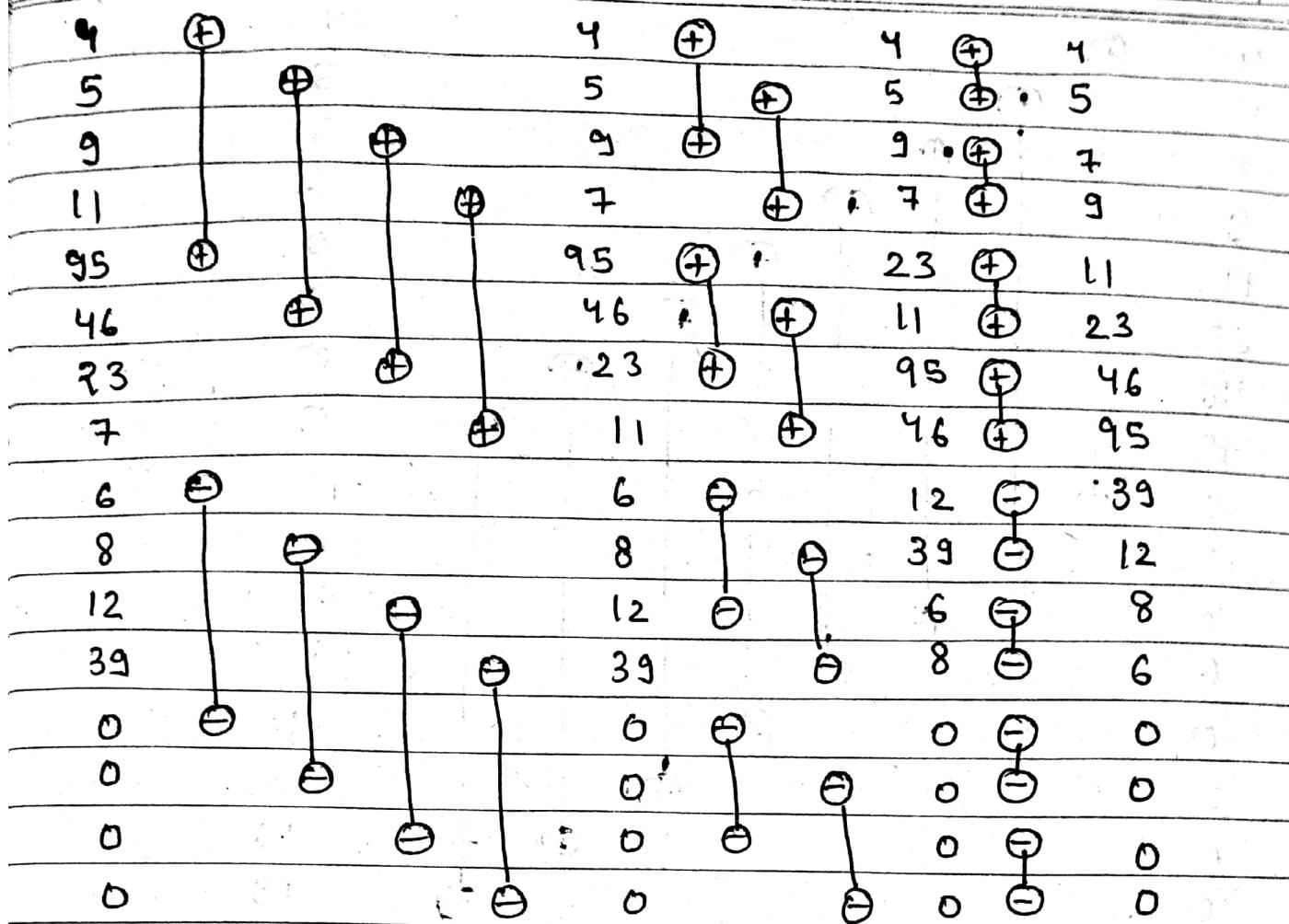
$\rightarrow P_4 \amalg P_5$

$$\underline{A_{42}}. \quad 4, 5, 9, 11, 95, 7, 23; 46, 39, 12, 6, 8$$

First create Bitonic sequence for above sequence

- ① Merge 4 zeroes at the end
 - ② Diagram.

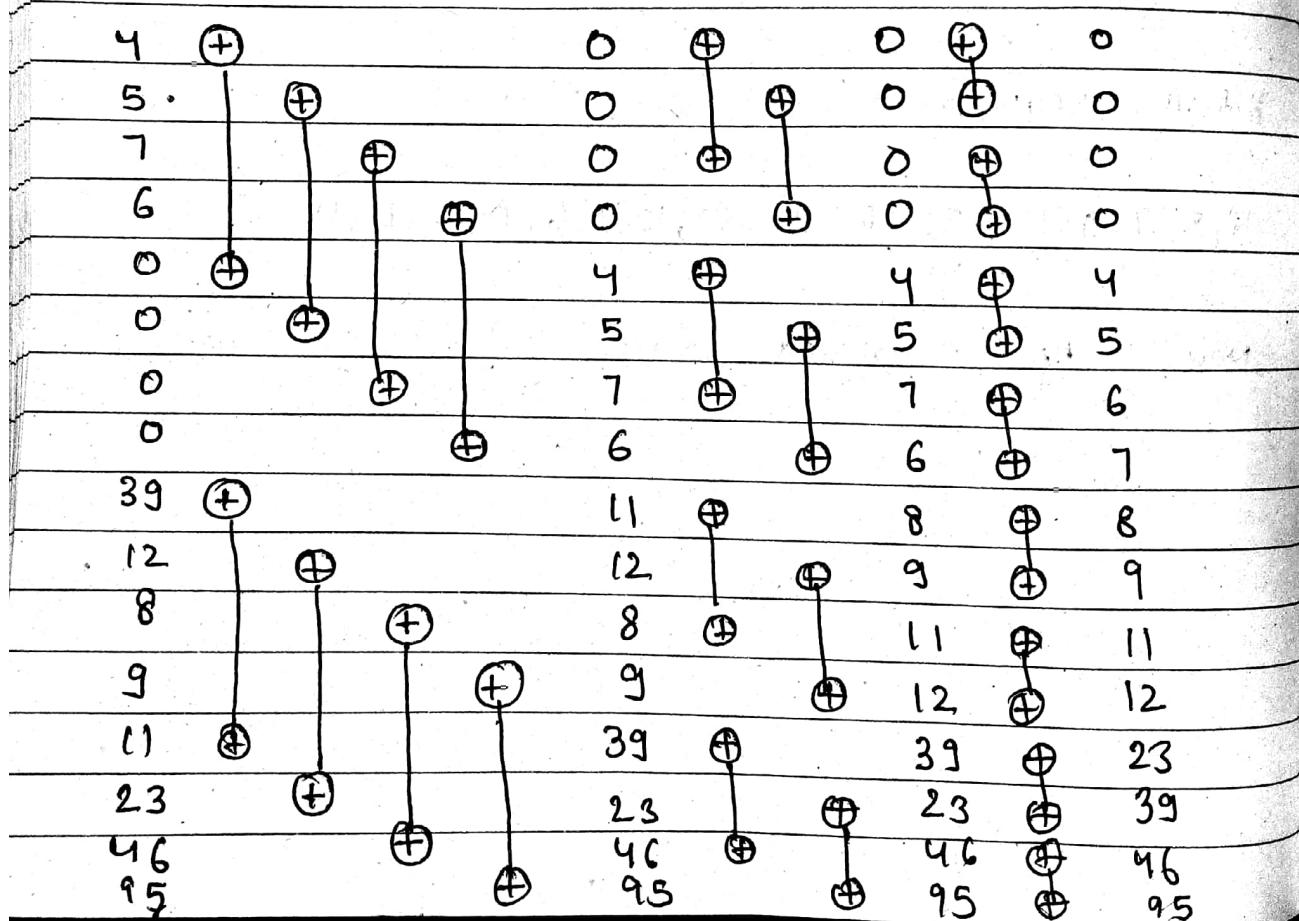
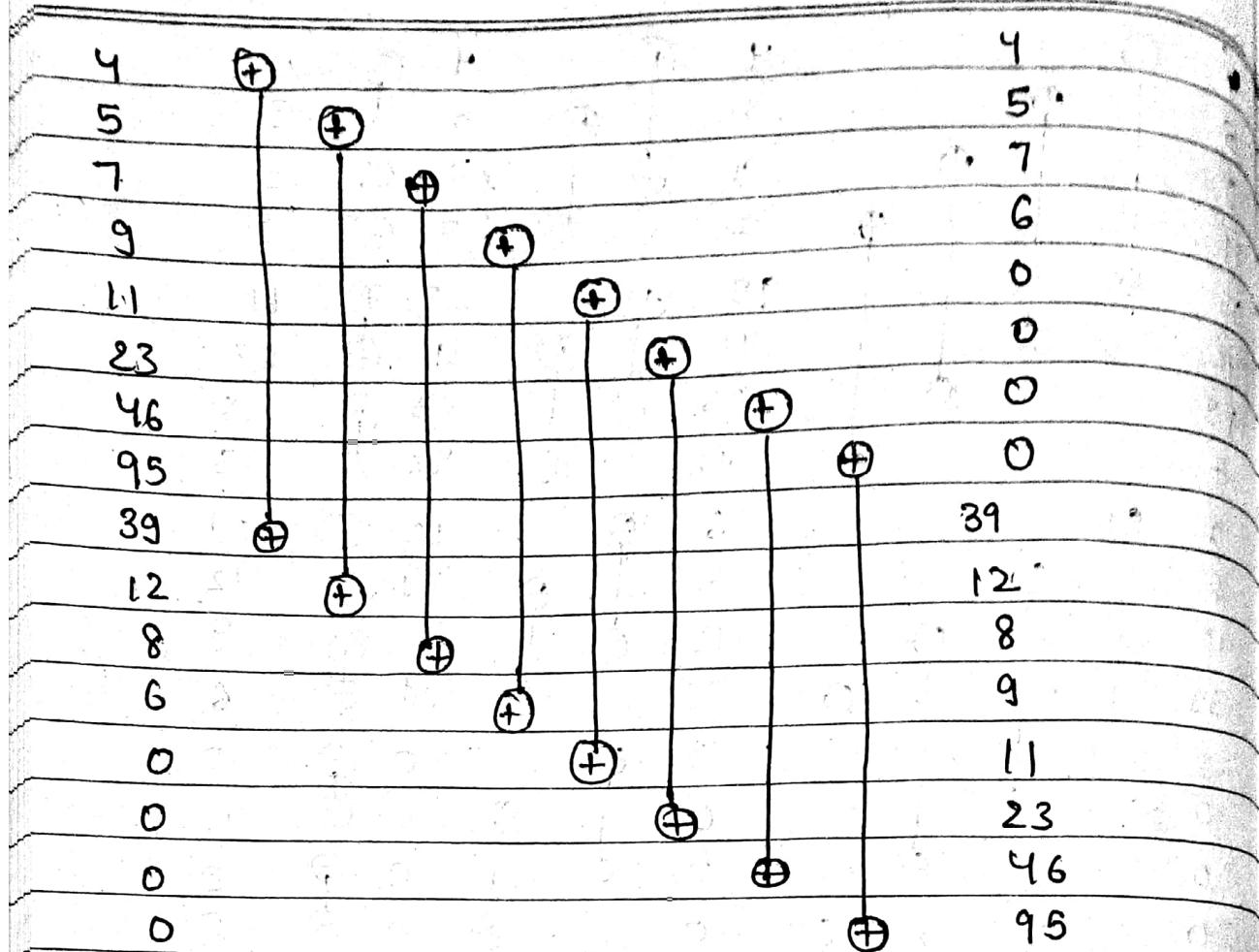




Bitonic Sequence :-

4, 5, 7, 9, 11, 23, 46, 95, 39, 12, 8, 6, 0, 0, 0, 0

Now, Apply Bitonic Sort.



Sorted Sequence :-

4, 5, 6, 7, 8, 9, 11, 12, 23, 39, 46, 95

Avg 3 (a)

(i) Task I, II, III, IV can be executed parallelly

so,

$$\begin{aligned}\text{Degree of concurrency} &= 10 + 10 + 10 + 10 \\ &= 40\end{aligned}$$

(ii) Task V & VI can be executed parallelly

so,

$$\text{Degree of concurrency} = 9 + 6 = 15$$

(iii) Degree of concurrency for Task VII = 8

$$\begin{aligned}\text{Now, Max no of concurrency} &= \max(40, 15, 8) \\ &= 40\end{aligned}$$

$$\begin{aligned}\text{Avg. degree of concurrency} &= \frac{40 + 15 + 8}{10 + 9 + 8} = \frac{63}{27} \\ &= 2.3\end{aligned}$$

(b)

(i) Task I, II, III, IV can be executed parallelly :-

$$\begin{aligned} \text{Degree of concurrency} &= 10 + 10 + 10 + 10 \\ &= 40 \end{aligned}$$

~~(ii) Task~~

Task 3, 6, 7 will get executed serially :-

$$\begin{aligned} \text{Max No of concurrency} &= \text{Max}(40, 6, 11, 7) \\ &= 40 \end{aligned}$$

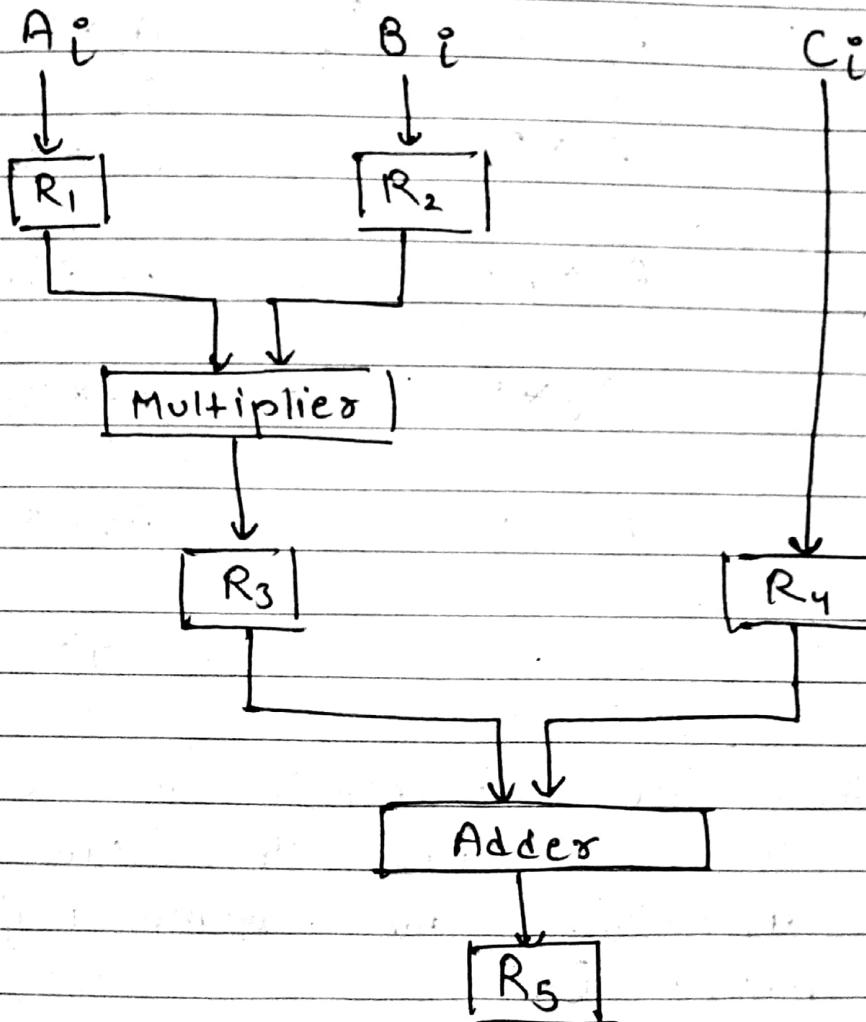
$$\begin{aligned} \text{Avg. degree of } n &= \frac{40+6+11+7}{10+6+11+7} = \frac{64}{34} \\ &= 1.88 \end{aligned}$$

Ay 1. Given expression can be decomposed into 3 segments:

$$R_1 \leftarrow A_i \quad R_2 \leftarrow B_i \quad (\text{Inputs } A_i \text{ & } B_i)$$

$$R_3 \leftarrow A_i * B_i \quad R_4 \leftarrow c_i \quad (\text{mult. & inp. } c_i)$$

$$R_5 \leftarrow R_3 + R_4 \quad (\text{Add } c_i \text{ to product})$$



Segment 1

Segment 2

Segment 3

 $R_1 \quad R_2$ $R_3 \quad R_4$ R_5 $A_1 \quad B_1$ $- \quad -$ $-$ $A_2 \quad B_2$ $A_1 * B_1 \quad C_1$ $-$ $A_3 \quad B_3$ $A_2 * B_2 \quad C_2$ $A_1 * B_1 + C_1$ $A_4 \quad B_4$ $A_3 * B_3 \quad C_3$ $A_2 * B_2 + C_2$ $A_5 \quad B_5$ $A_4 * B_4 \quad C_4$ $A_3 * B_3 + C_3$ $A_6 \quad B_6$ $A_5 * B_5 \quad C_5$ $A_4 * B_4 + C_4$ $A_7 \quad B_7$ $A_6 * B_6 \quad C_6$ $A_5 * B_5 + C_5$ $- \quad -$ $A_7 * B_7 \quad C_7$ $A_6 * B_6 + C_6$ $- \quad -$ $-$ $-$ $A_7 * B_7 + C_7$

Q5. In this case, 8 flops can be performed on 5 cache lines (one for matrix A and four for column major access to matrix B).

This corresponds to a speed of 16 MFLOPS.

Ques 6 In terms of power , the most powerful model is the CRCW (concurrent Read, concurrent write) model , because it allows multiple processors to simultaneously read and write to the same memory location .

In other words , in the CRCW model , multiple processors can write to the same memory location without any conflicts , which means that it can execute certain type of algo. faster than other models.

Ans consider a d -dimensional hypercube with 2^d nodes. Let's fix k bits, where $k \leq d$, and call these the "fixed" bits. The remaining $(d-k)$ bits are called "free" bits.

Suppose we choose two nodes in the hypercube whose labels differ in exactly the free bits. That is, the fixed bits are the same for both nodes, but the free bits differ. These two nodes will be adjacent in the hypercube, since they differ in only one bit.

Now, consider all the nodes that differ from one ~~part~~ of these two nodes only in their free bits. Since, the fix bits are same for all these nodes, they form a $(d-k)$ -dimensional subcube. This subcube has 2^{d-k} nodes, since each of the free bits can take on two possible values.

Moreover, since any two adjacent nodes in the subcube differ in only one free bit, this subcube is also a hypercube in its own right, with edge length 1 in each of its $(d-k)$ dimensions. Therefore, the nodes whose labels differ in the remaining $(d-k)$ bit positions form a $(d-k)$ -dimensional subcube composed of $2^{(d-k)}$ nodes.

Q8 Benz network is a non-blocking network and it is generated by joining two butterfly networks back to back, in such a manner that data flows forward through one and in reverse through the other.

In Benz network to reduce the complexity the middle stage switches can recursively be broken into $\frac{N}{4} \times \frac{N}{4}$ ($\text{then } \frac{N}{8} \times \frac{N}{8}$)

till switch size become 2×2 .

→ Connection in 2×2 switch will either be straight, exchange, lower broadcast or upper broadcast.

→ Benz network uses lesser switches and it provides good connectivity.

→ To find hardware complexity of Benz network let us assume that

$$N = 2^n \Rightarrow n = \log_2 N$$

($\because N = \text{no of stages input}$)

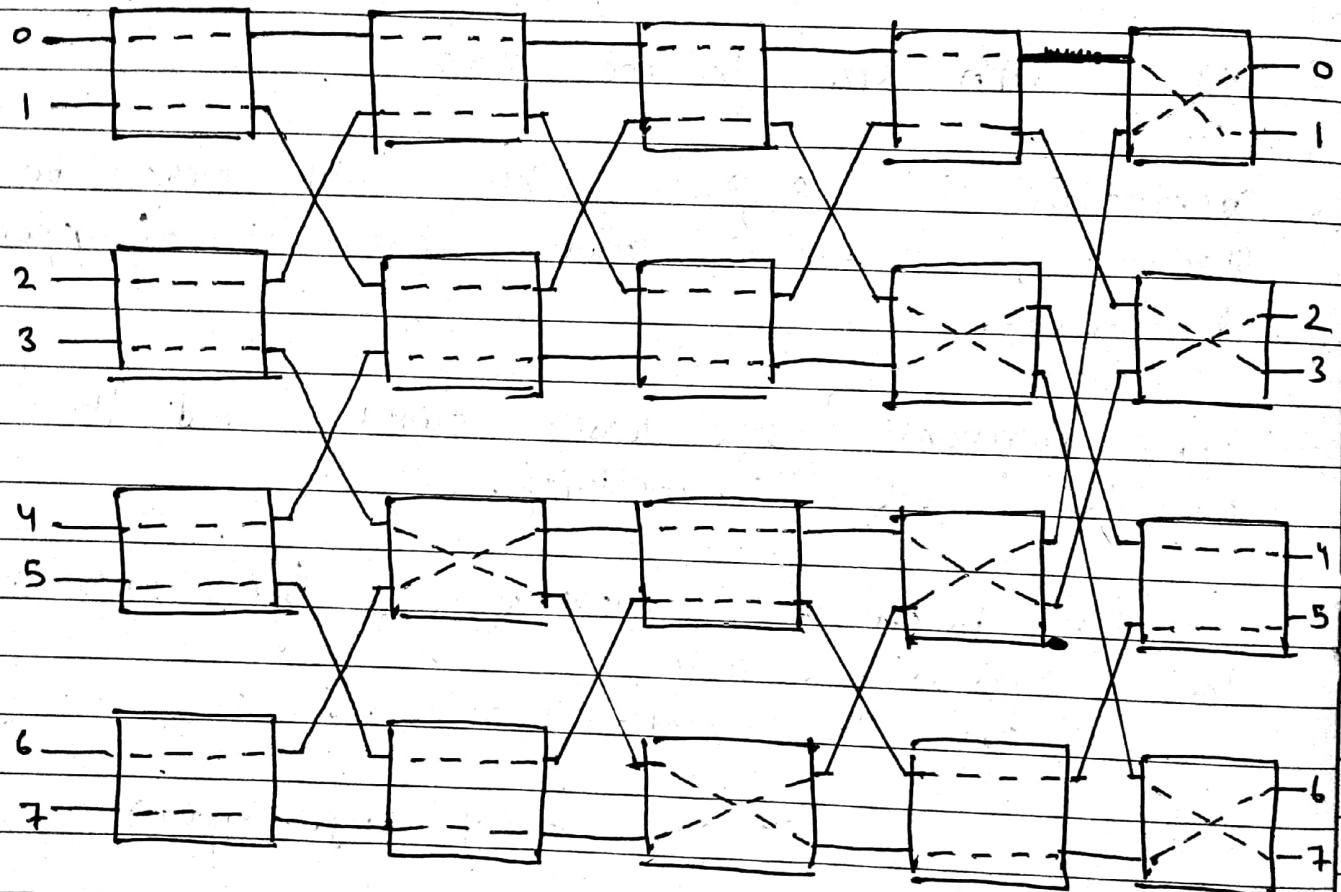
$$2^n - 1 = 2 \log_2 N - 1$$

$$\therefore \text{Total no of cells in network} = \frac{N}{2} (2 \log_2 N - 1)$$

$$= N \log_2 N - \frac{N}{2}$$

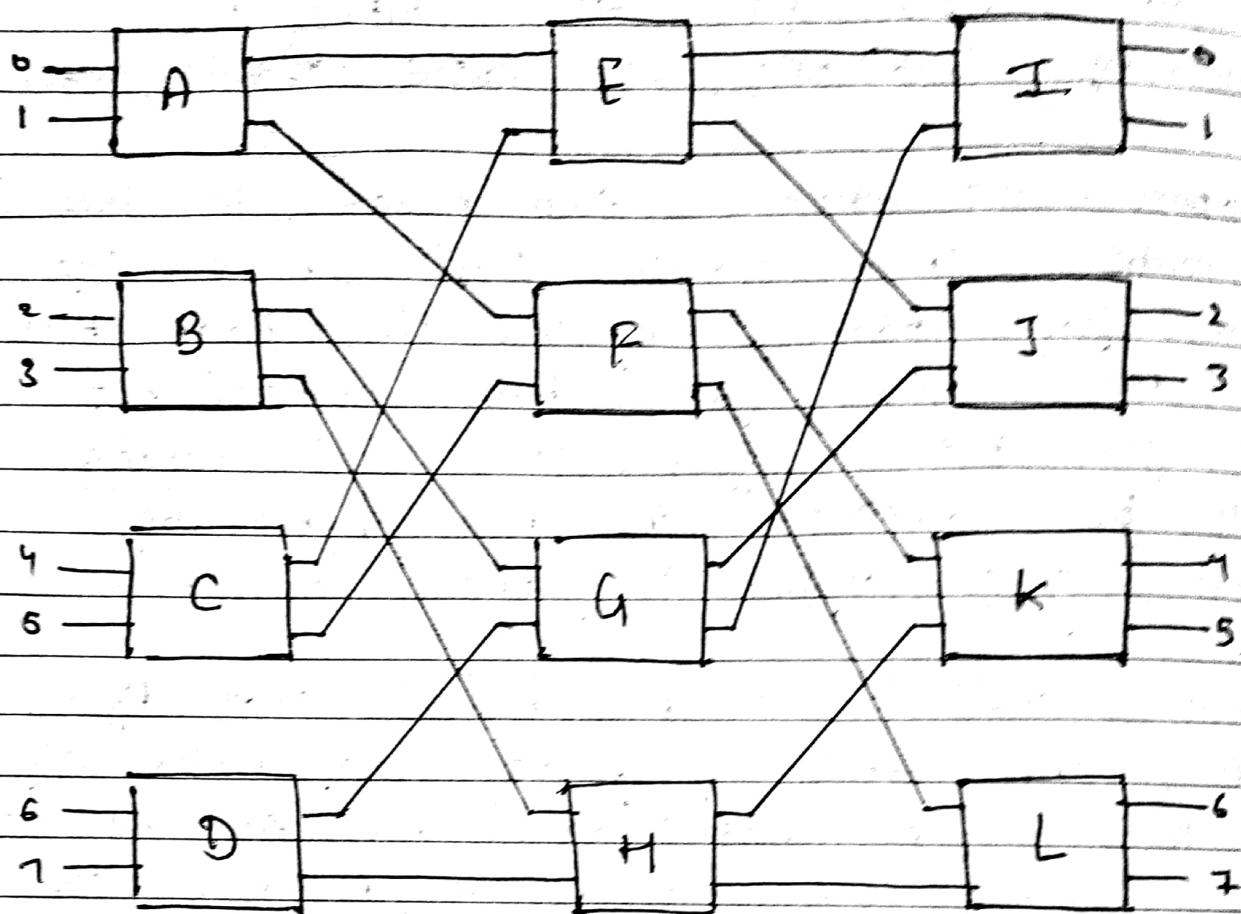
No. of switches in diff. network for various input :-

Input	No. of cross bars	Switches	Benz - Network
2	N	9	4
8	84	69	80
64	4096	1536	1408
256	65536	12888	7680



∴ Require 8×8 benz network of 4 stages.

Ques 9



Routing Algo for omega network :-

- i) Consider the first stage of the of the Omega network to the right.
- ii) It should be noted that the interconnection pattern among stages follow the shuffle operation.
- iii) All four first stage switches send their upper output to switch E and G and their lower output to switches F and H.

- iv) switches E and G both end their output to switch J and I; their data can only reach the network output = 4, 5, 6 & 7.
- v) Each first stage switch must be set so that its upper output has a destination with binary value 000, 001, 010, 011 (0 in the 1st bit).
- vi) Similarly the ^{lower} output of each 1st stage switch must have a 1 in the first bit postn. of its dest. to reach output 100, 101, 110, or 111.
- vii) If two input to 1st stage switch have the same value in the first bit postn, the Omega network can not realize this permutation.
- viii) For second stage, the second bit of the destination determines the setting of switch.
- ix) Similarly, the least significant bit of the destination determine the setting of switches in 3rd stage.
- x) Since, the 3rd stage output are the output of network, the last cannot block a permutation that has been routed successfully by the previous stages.

Q410 Since, we have 6 segments and 200 tasks.

∴ Total time required will be given by

$$= k + (n-1)$$

here,

$$k = \text{no. of segments} = 6$$

$$n = \text{no of tasks} = 200$$

$$\begin{aligned}\text{No of clock cycles} &= 6 + (200 - 1) \\ &= 6 + 199 \\ &= 205\end{aligned}$$