

## Module - I

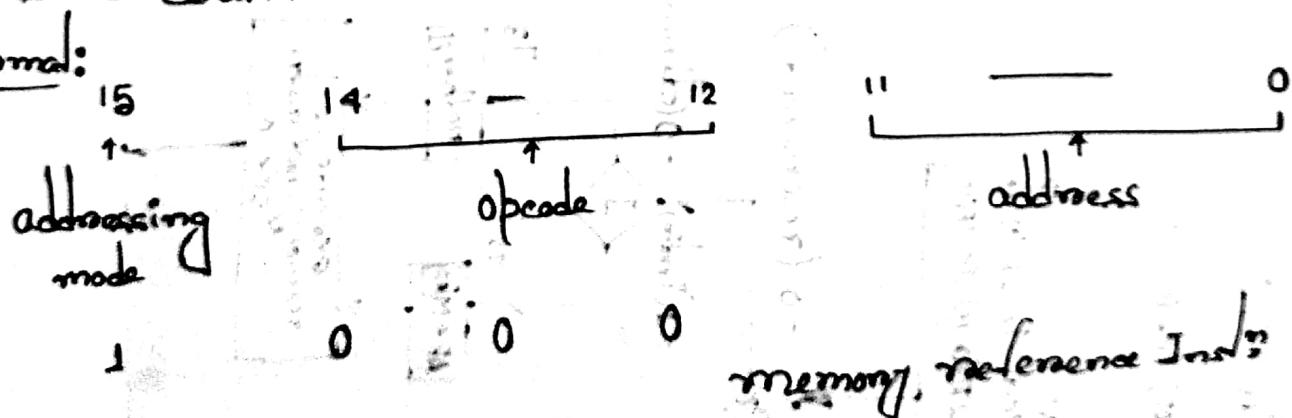
### Inst's execution Mechanism Details:

In a basic Computer each inst's cycle consists of following phases

1. Fetch an inst's from mem.
2. Decode the inst's
3. Read the effective address from mem. if the inst's has an indirect address
4. Execute the inst's

upon completion of step-4, the control goes back to step-1 and execute the next inst's. This process continues indefinitely unless a HALT inst's is encountered.

Format:



0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1      Register reference Inst's

0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1      I/O operation  
 $I = \begin{cases} 0 & \rightarrow \text{direct address} \\ 1 & \rightarrow \text{An } " \end{cases}$

Initially the PC is loaded with the address of the first inst<sub>n</sub> in the program. The sequence Counter SC is cleared to 0 & providing a timing signal T<sub>0</sub>. After each clock pulse SC is increased by 1 so the timing signals go through T<sub>0</sub>, T<sub>1</sub>, T<sub>2</sub> and so on.

T<sub>0</sub>: AR ← PC      } Fetch

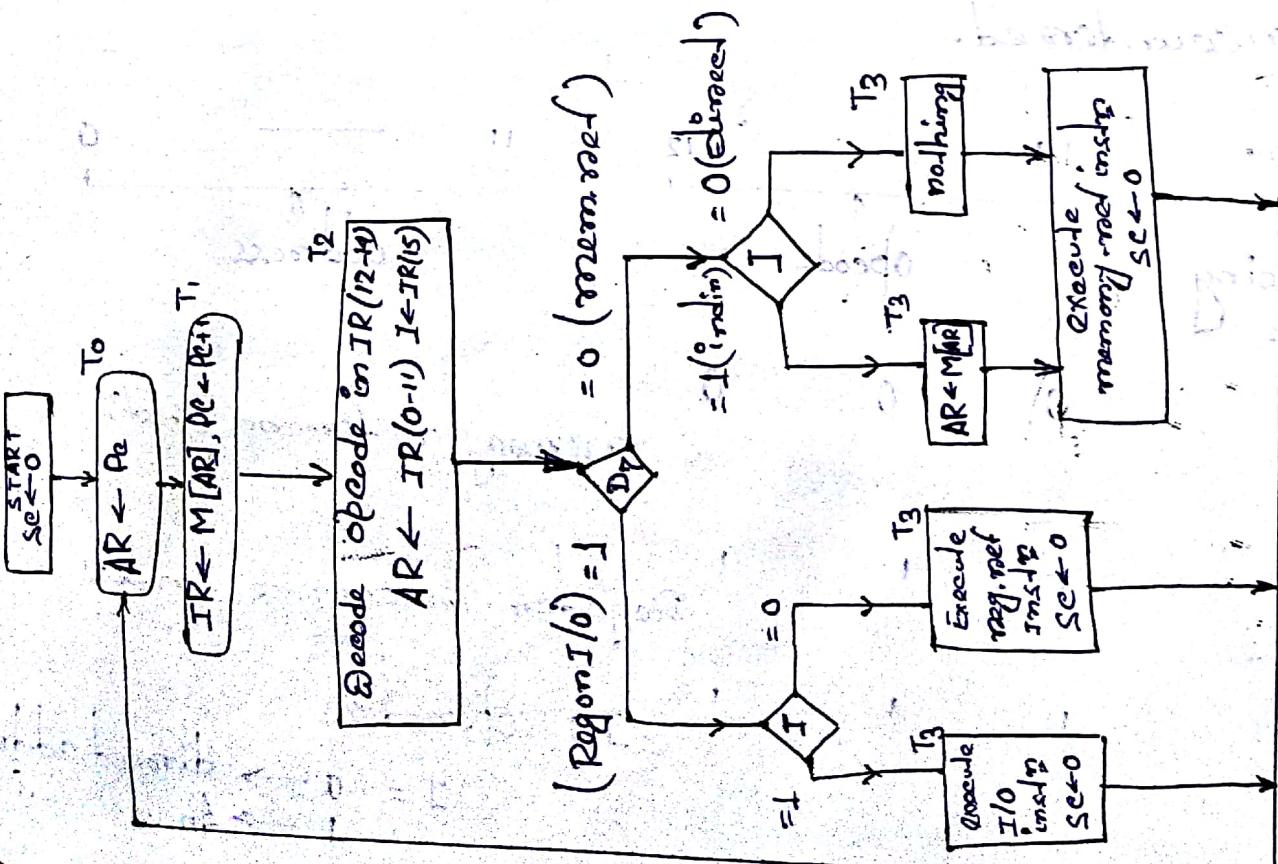
T<sub>1</sub>: IR ← M[AR], PC ← PC + 1

T<sub>2</sub>: decode opCode IR(12-19)      } decode

AR ← IR(0-11), I ← IR(15)

T<sub>3</sub>: decision I(indirect) → next clock pulse  
0 (direct) → T<sub>3</sub> itself

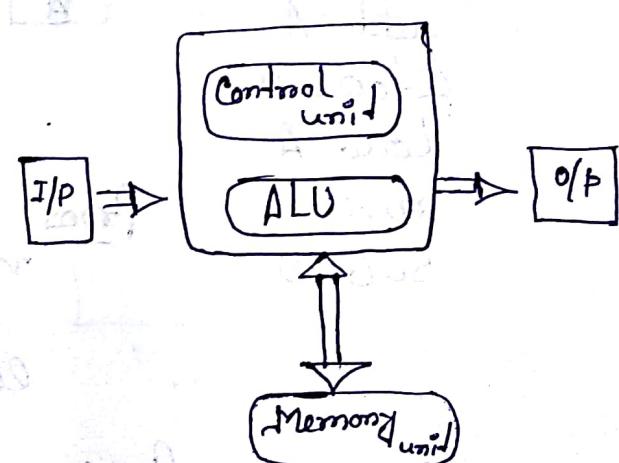
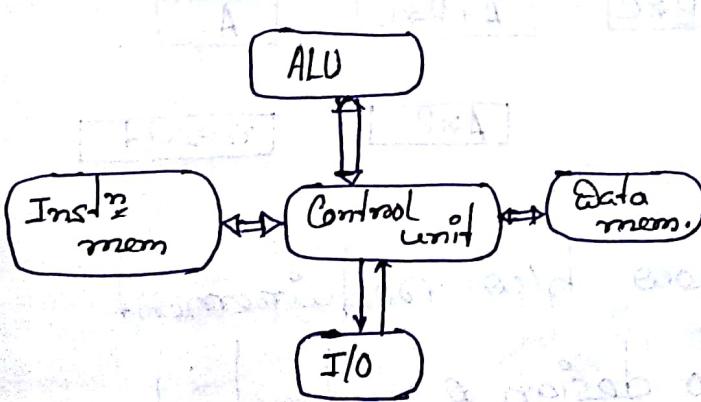
flowchart:



# Difference between Harvard & Von-Neumann architecture

June

Harvard	Von-Neumann
i) It requires two mem. for their instr & Data	i) Here requires only one
ii) Design is complicated	ii) Design is simple
iii) Here requires separate bus	iii) Requires only one bus
iv) Processors can complete instr in one cycle	iv) Processors complete instr in two cycles.
v) comparatively high cost	v) comparatively low



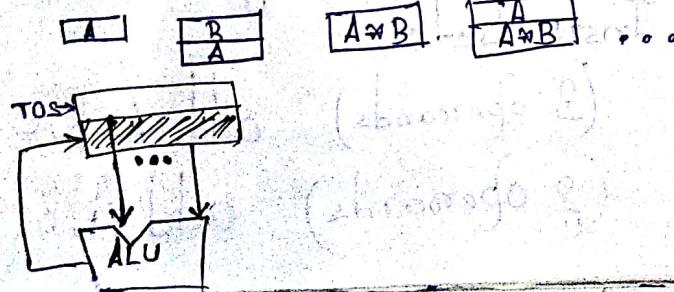
## Stack architecture:

- Instr Set: Add, Sub, Mul, Div  
Push A, Pop A

ex:  $A * B - (A + C * B)$

Push A  
Push B  
Mul  
Push A  
Push B  
Push C

MUL  
ADD  
Sub  
Result



Pros Good code density

Low h/w requirements

Cons Stack becomes the bottleneck  
Little ability for Pipelining

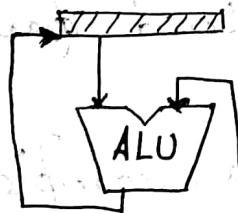
Accumulators architecture:

Inst<sup>n</sup> Set

add A, sub A, mult A, div A

Load A, Store A

ex:  $A \times B - (A + C \times B)$



Load B

mul C

add A

Store D

Load A

mult B

sub D

B

B \* C

A + B \* C

A

A \* B

result

Pros

very low h/w requirements

easy to design & understand.

Cons

Accumulators becomes the bottleneck

High memory traffic

Memory-Memory Architecture

Inst<sup>n</sup> Set:

(3 operands) add A, B, C sub A, B, C mul A, B, C

(2 operands) add A, B sub A, B mul A, B

$$\text{ex: } A * B - (A + C * B)$$

3 operands

mul D, A, B

mul E, C, B

add E, A, E

Sub E, D, E

2 operands

mov D, A

mul D, B

mov E, C

mul E, B

add E, A

sub E, D (D থেকে E কে

Pros Reduce Sessn Inst<sup>n</sup> (especially if 3 operands)  
Easy to write Compilers for ( " " " " )

Cons very high mem. traffic

variable no. of clocks per Inst<sup>n</sup>

### Load Store architecture

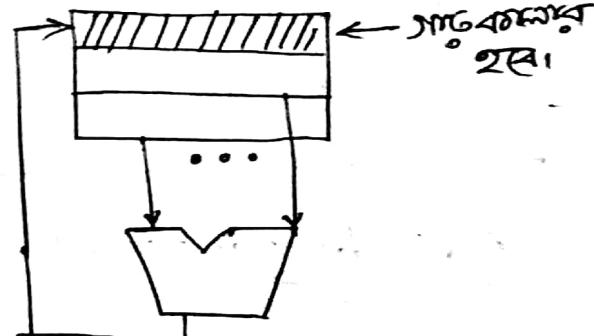
Inst<sup>n</sup> Set

add R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub> Sub R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>

mul R<sub>1</sub>, R<sub>2</sub>, R<sub>3</sub>

load R<sub>1</sub>, &A Store R<sub>1</sub>, &A

move R<sub>1</sub>, R<sub>2</sub>



$$R_3 = R_1, +, -, \times, / R_2$$

$$\text{ex: } A * B - (A + C * B)$$

load R<sub>1</sub>, &A

load R<sub>2</sub>, &B

load R<sub>3</sub>, &C

mul R<sub>7</sub>, R<sub>2</sub>, R<sub>3</sub>      | A \* B |

mul R<sub>9</sub>, R<sub>1</sub>, R<sub>2</sub>

sub R<sub>10</sub>, R<sub>9</sub>, R<sub>8</sub>

add R<sub>8</sub>, R<sub>7</sub>, R<sub>1</sub>

| A \* B - (A + C \* B) |

Pros: Simple fixed length instr<sup>n</sup> encoding  
Instr<sup>n</sup>s takes similar no. of cycles.

Cons: Higher Instr<sup>n</sup> Count  
Dependent on good Compiler.

### RISC

- ① Stands for "Reduced Instr<sup>n</sup> Set Computer"
- ② Emphasis On S/cos.
- ③ Include Single CLK
- ④ Calculations are faster
- ⑤ Execution time is very less

### CISC

- ① Stands for "Complex Instr<sup>n</sup> Set Computer"
- ② Emphasis on H/cos include multi CLK calculations are slower
- ③ Execution time is high

MIPS Architecture: An acronym for "Microprocessors without Interlocked Pipeline Stages". It's a RISC archi. developed by MIPS Technologies. The early version was 32 bit, 64 bit comes later. There are multiple versions of MIPS MIPS I, II, ... V as well as five releases for 32 bit & 64 bit. As of April 2017 they ~~had~~ release their current versions. (MIPS 32/64 Releases)

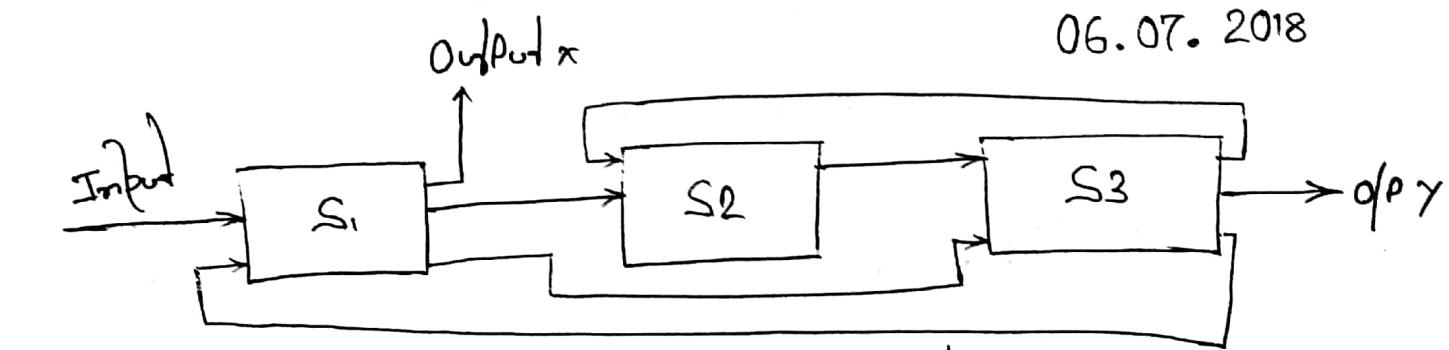
Inst<sup>n</sup> format

Inst<sup>n</sup> are divided into 3 types, R, I, J

Every Inst<sup>n</sup> start with 6 bit of code. In addition to opcode R type inst<sup>n</sup> specify 3 registers, a shift field, a function field

I type instructions specify two registers and 16 bit immediate value. J type follow the opcode with a 26 bit jump target.

06.07.2018



→  $S_1, S_2, S_3 = \text{Stage}$   $1, 2, \dots, 8 = \text{clk cycle}$

Reservation technique for X

1	2	3	4	5	6	7	8
$S_1$	x				x		x

$S_2$		x		x			
$S_3$			x		x		x

The evaluation time = 8 [no. of columns]

Latency is the gap b/w 2 check mark

1011010

0 ← Permissible latency

initial Collision vectors.

①, 3, 6 Position are permissible

0101101

← Right shift with 0

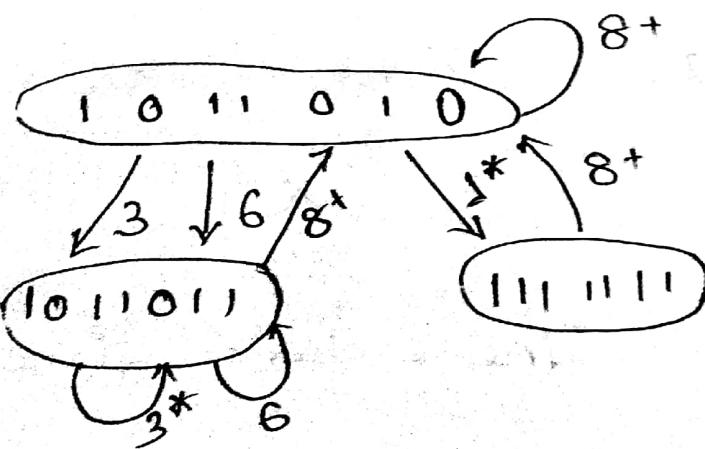
Now bitwise OR

.....111111

1, ③, 6

0001011  
1011010  
-----  
1011011

1, 3, ⑥



$$\begin{array}{r}
 0\ 0\ 0\ 1\ 0\ 1\ 1 \\
 1\ 0\ 1\ 1\ 0\ 1\ 0 \\
 \hline
 1\ 0\ 1\ 1\ 0\ 1\ 1
 \end{array} \quad \leftarrow \text{36:1 form 3}$$

$$\begin{array}{r}
 0\ 0\ 0\ 0\ 0\ 0\ 1 \\
 1\ 0\ 1\ 1\ 0\ 1\ 0 \\
 \hline
 1\ 0\ 1\ 1\ 0\ 1\ 1
 \end{array}$$

cycles:  $\overbrace{(3)}^{\text{simple}}, \overbrace{(6)}^{\text{greedy}}, \overbrace{(8)}^{\text{simple}}, \overbrace{(1,8)}^{\text{greedy}}, \overbrace{(3,8)}^{\text{greedy}}$   
 $\overbrace{(6,8)}^{\text{greedy}}, \overbrace{(3,3,8)}^{\text{greedy}}, \overbrace{(3,6)}^{\text{greedy}}, \overbrace{(6,3,8)}^{\text{greedy}}, \overbrace{(3,8,8)}^{\text{greedy}}$

simple cycle  $\leftarrow$  (no one of the state visited twice)  
 except initial state

greedy cycle:  $(1,8) (3)$  (average latency is less)

$$\text{MAL} = \frac{(1+8)/2}{2} = 4.5, 3 \quad \text{one double collision vector}$$

so 3 will answers

$$\frac{1+8}{2} = 9/2 = 4.5 \leq 4 \\
 3 < 4$$

Diagram  $\rightarrow$  RT table  $\rightarrow$  Collision vector

Consider the class example  
 forbidden latency

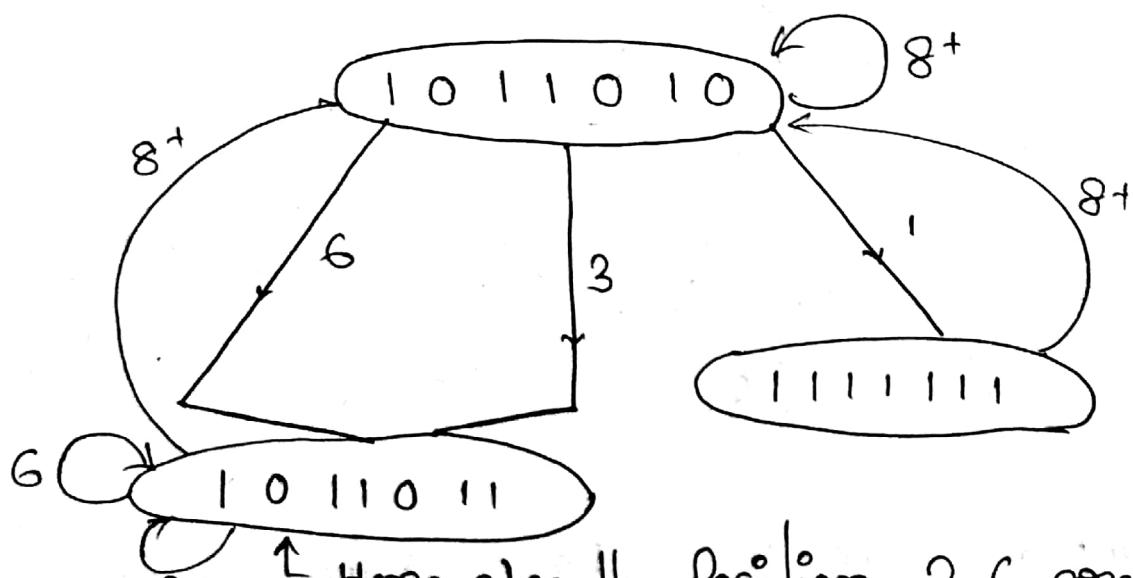
$$= \{(6-1), (8-1), (8-6), (4-2) \\
 \quad (5-3), (8-3), (9-5)\}$$

$$= \{5, 7, 2, 2, 2, 4, 2\} = \{2, 4, 5, 7\}$$

- state diagram  $\rightarrow$  permission latency
- ① simple cycles latency
  - ② greedy cycles
  - ③ MAL (minimal Average latency)

Hence the highest no is 7 So the vectors will be in 7 bit

$c_7 \ c_6 \ c_5 \ c_4 \ c_3 \ c_2 \ c_1$  ← Collision vectors  
 ↓ 0 ↓ 1 0 1 0



Here also the position 3, 6 are permissible

$$\begin{array}{r}
 36 \text{ bit} \\
 \begin{array}{r}
 1.011010 \\
 0001011 \\
 \hline
 1011011
 \end{array}
 \end{array}$$

$$\begin{array}{r}
 6 \text{ bit} \\
 \begin{array}{r}
 1011010 \\
 0000001 \\
 \hline
 1011011
 \end{array}
 \end{array}$$

↑ as it's same with Collision vectors so self loop of 3 formed

Prob:  $T_1 = 60 \text{ ns}$

$T_2 = 90 \text{ ns}$

$T_3 = 50 \text{ ns}$

$T_4 = 40 \text{ ns}$

10 ns is the latch delay. Calculate the

speed up of the pipeline system?

uniform time 75 ns.

what will be the speed up?

$$S_K = \frac{60 + 90 + 50 + 40}{T_{\max} + T_L} = \frac{240}{100} = 2.4$$

Uniform time 75 ns

$$\frac{75+75+75+75}{75+10} = \frac{300}{85}$$

⦿ Pipeline Concept from CO copy previous Semesters.

## Module 2 : Vector Processing

In many Science & Engg. applications, the problems can be formulated in terms of vectors and matrices that lend themselves to vector processing.

Computers with vector processing capabilities are in demand in specialized applications.

- Long range weather forecasting
- Petroleum exploration
- Medical diagnosis
- Image Processing
- Artificial Intelligence

**vector processor:** An Com. Science a vector processor is a CPU that implements an inst<sup>n</sup>.

That operate on one dimensional arrays of data called vectors.

Compared to scalar processors whose inst<sup>n</sup> operate on single data item.

☒ Component of vector processors —

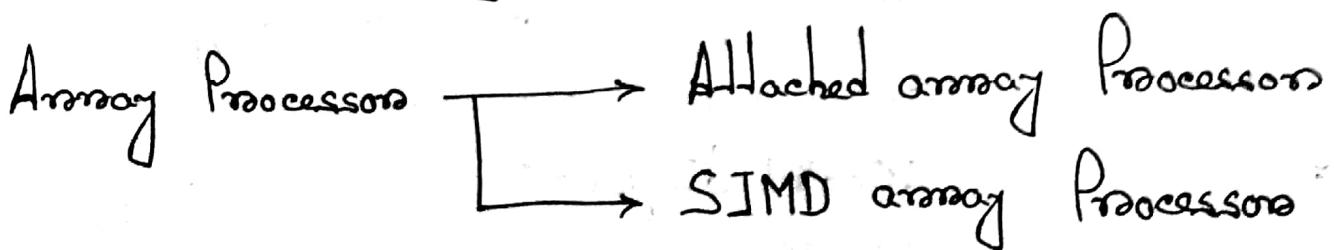
Vector registers  $\Rightarrow$  fixed length bank holding a single vector

- has at least 2 read and 1 write ports
- typically 8-16 vector registers, each holding 64-128 bit elements

vector fun<sup>n</sup> unit  $\Rightarrow$  fully pipelined, start new operation in every clock

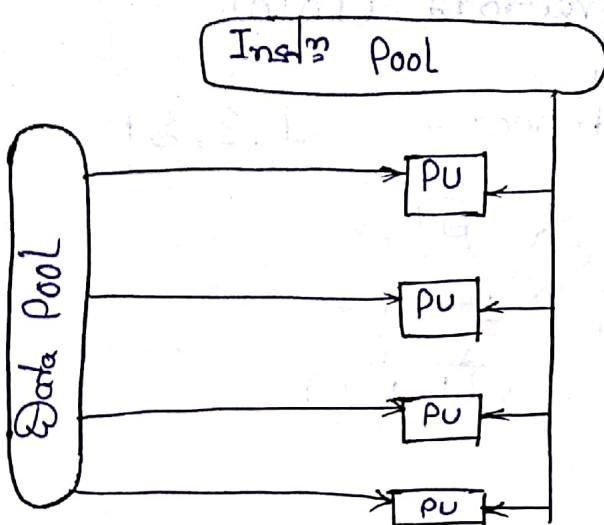
- typically 4 to 8: FP add, FP mult, FP reciprocal ( $1/x$ ), integers add, logical, shift

vector load & store unit  $\Rightarrow$  fully pipelined unit to load or store vectors. And ④ scalar Reg.



SIMD Array Processors  $\Rightarrow$  An SIMD array Processor is a computer with multiple processing units operating in parallel. The processing units are synchronized to perform same operation under the common control unit thus provide a single  $\frac{1}{2}$  streams with multiple data stream organization.

## SIMD



Processor unit = Pu

example:

Illiac - Iv  
STARAN

## SIMD Matrix Multiplication $\Rightarrow$

The algorithm is to perform  $A \times B$

for  $i=1$  to  $n$  do

    for  $k=1$  to  $n$  do

$c_{ik} = 0$  [vector load]

    for  $j=1$  to  $n$  do

        for  $K=1$  to  $n$  do

$c_{ik} = c_{ik} + a_{ij} * b_{jk}$

            [vector multiply]

    end of  $j$  loop

end .. "

Time Complexity:  $O(n^2)$

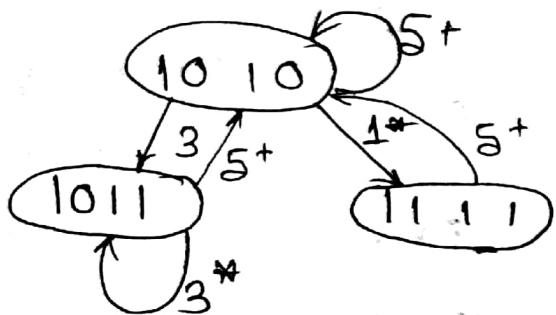
Pass loop =  $O(1)$

09.07.18

	1	2	3	4	5	6
$s_1$	X					
$s_2$			X			
$s_3$		X		X		X

Initial Collision vectors  $\langle 1010 \rangle$

Permissible latencies —  $1, 3, 5^+$



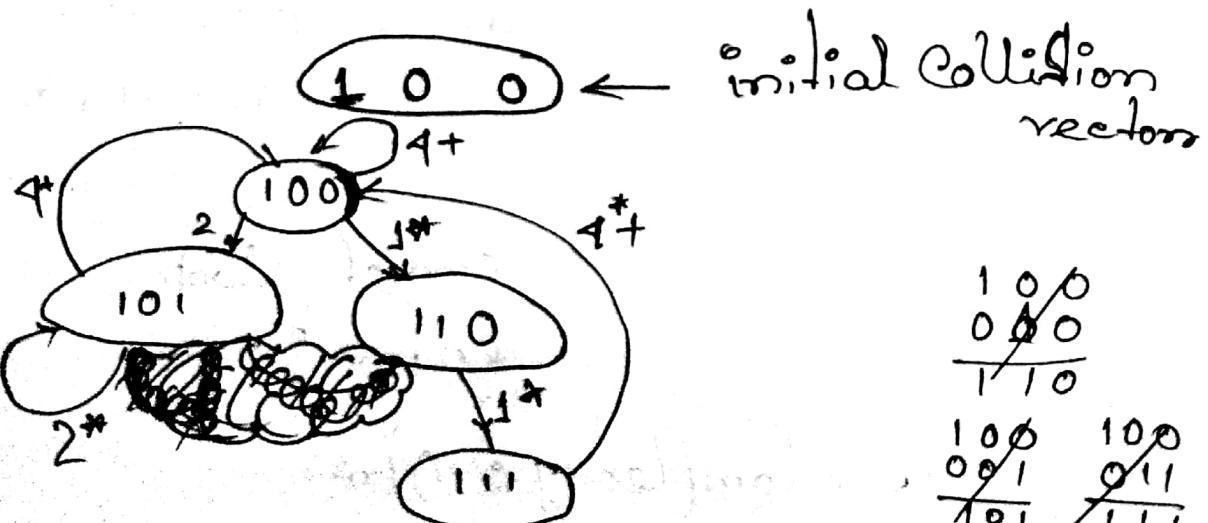
Simple cycles =  $(3), (5), (3, 5), (1, 5)$

greedy cycle =  ~~$\nabla$~~   $3, (1, 5)$

$S_1$       1      2      3      4  
      x                  x

$S_2$                   x

$S_3$                   x



value of  $\text{MAL} = 2$

next Page

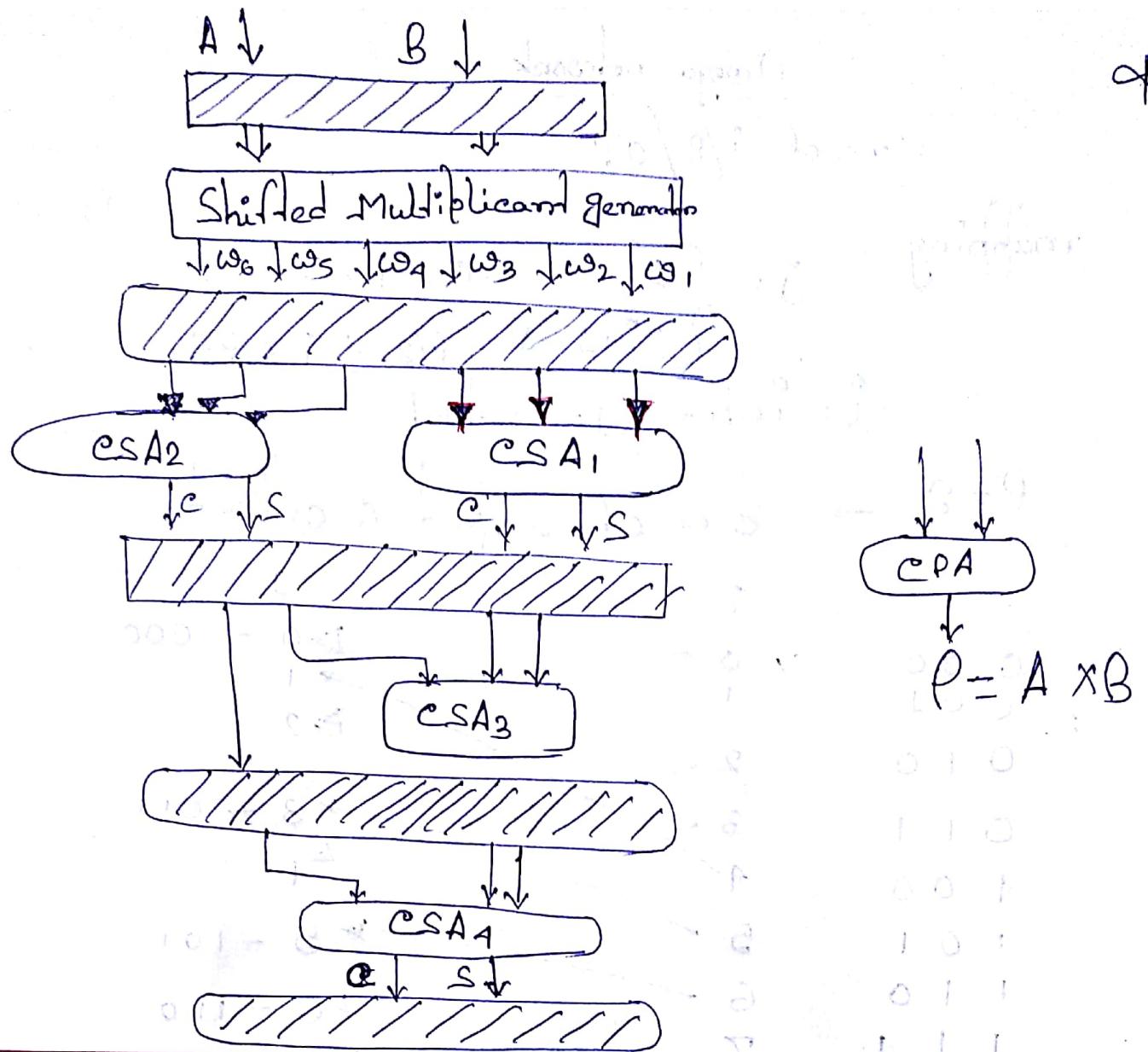
1	0	0
0	0	0
1	1	0

1	0	0
0	0	1
1	0	1

1	0	0
0	0	0
1	1	1



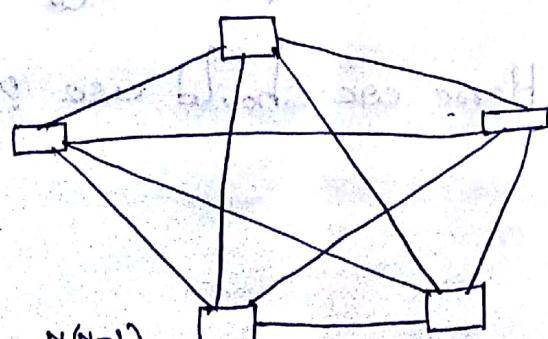
### Interconnection Networks

In Parallel Computing we use multiple no. of Processors ( $N$ )

- Cost effective
- Time "

Hence we use mesh

topology to connect the  
n/c



$$\text{No. of Links} = \frac{N(N-1)}{2}$$

## Omega network

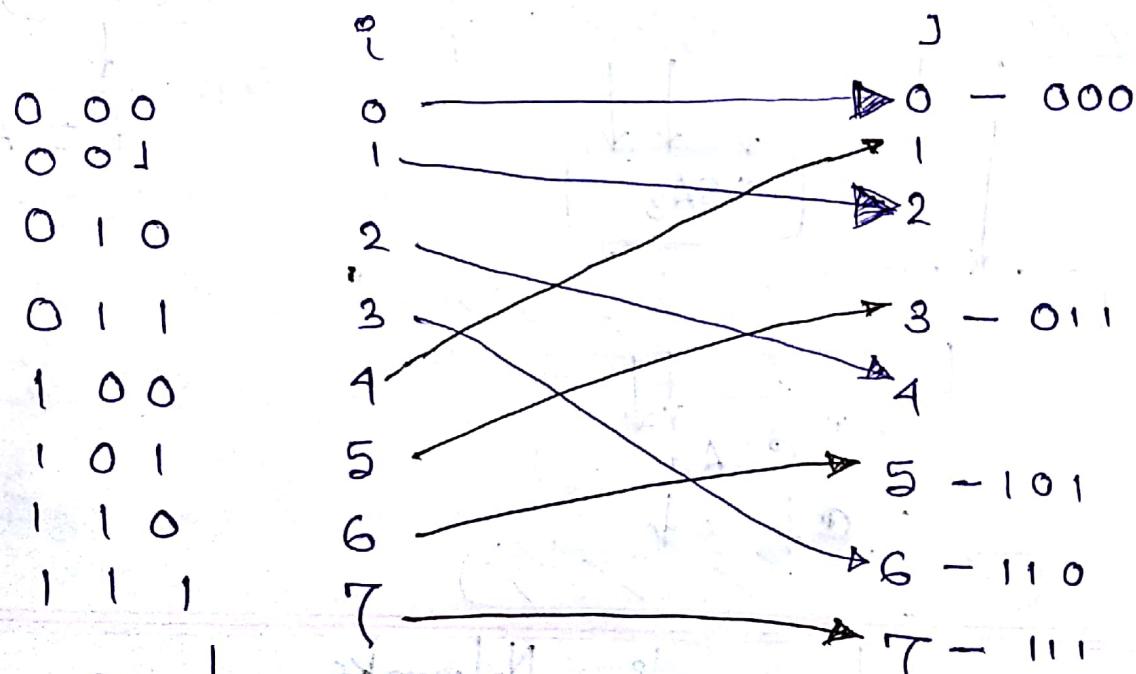
P no. of i/p / o/p

mapping

$$j = \begin{cases} 2i & 0 \leq i \leq P/2 - 1 \\ 2i + P & P/2 \leq i \leq P - 1 \end{cases}$$

$i = \text{input}, j = \text{output}$

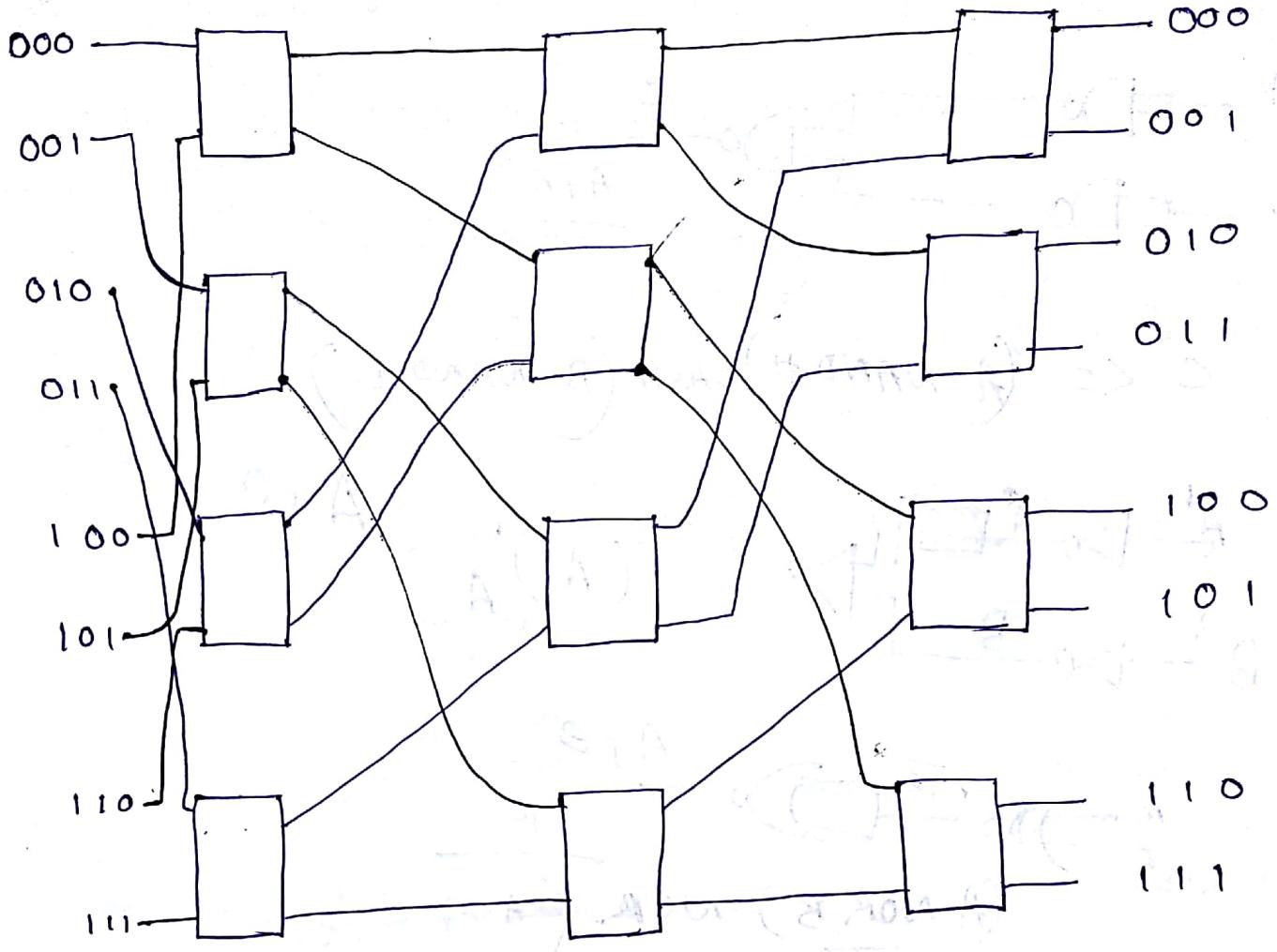
$P = 8 \rightarrow 8 \text{ no. of input \& output}$



→ No. of stages creating Omega n/w

Hence 8 no. of input & output present. So, the no of stages =  $\log_2 8 = 3$

Hence we should use  $2 \times 2$  switches



assignment operators  $\frac{ab}{c \leftarrow a \text{ OR } b};$

AND using NAND

OR " NAND

NOT "

AND "

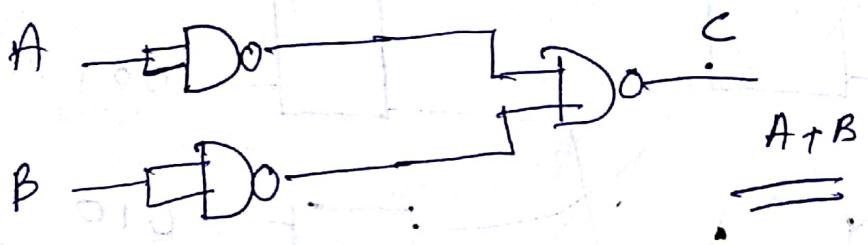
OR "

NOT "

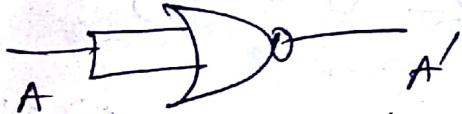
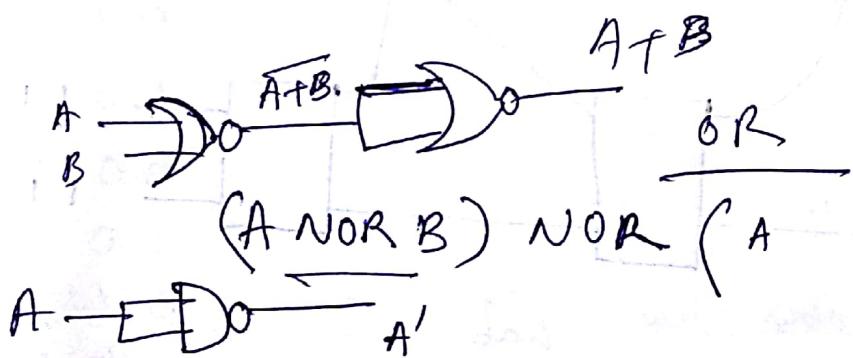
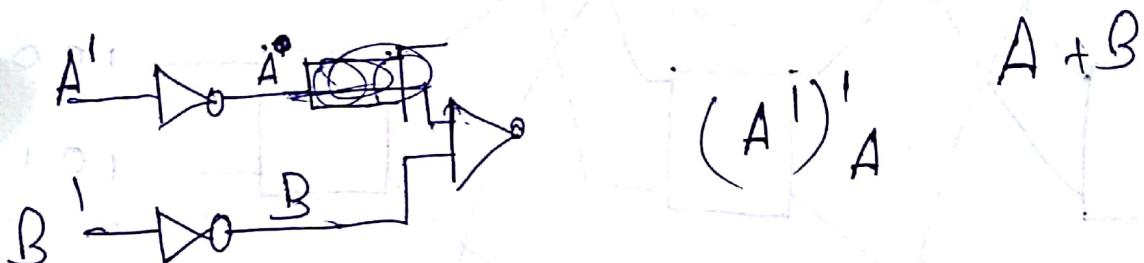
$$C \leftarrow (A \text{ NAND } B) \text{ NAND } (A \text{ NAND } B);$$

$(AB)'$





$$C \leq (A \text{ NAND } A) \text{ NAND } (B \text{ NAND } B)$$



vector Processing

① vectors to vectors

11.07.18

4 types are there

②  $\mathbf{v} \times \mathbf{v} \rightarrow \mathbf{v}$

$$B(I) \leftarrow \sqrt{A(I)}$$

Cross Product

③  $\mathbf{Y} \times \mathbf{S} \rightarrow \mathbf{V}$



## Masking Vectors:

$$X = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$B = \{0, 1, 1, 0, 1, 1, 0, 1\}$$

$$Y = X(B) = \{2, 3, 5, 6, 8\}$$

1 ← active  
0 ← passive

## Boolean vectors:

$$X = \{2, 3, 9, 15\} \quad Y = \{10, 1, 3, 20\}$$

$$A = X \gtreqless Y$$

$$= \{0, 1, 1, 0\}$$

Length of vectors

How many data processes in unit time by a vector.

opcode, base address  
add. offset  
add. increment

Memory to memory architecture

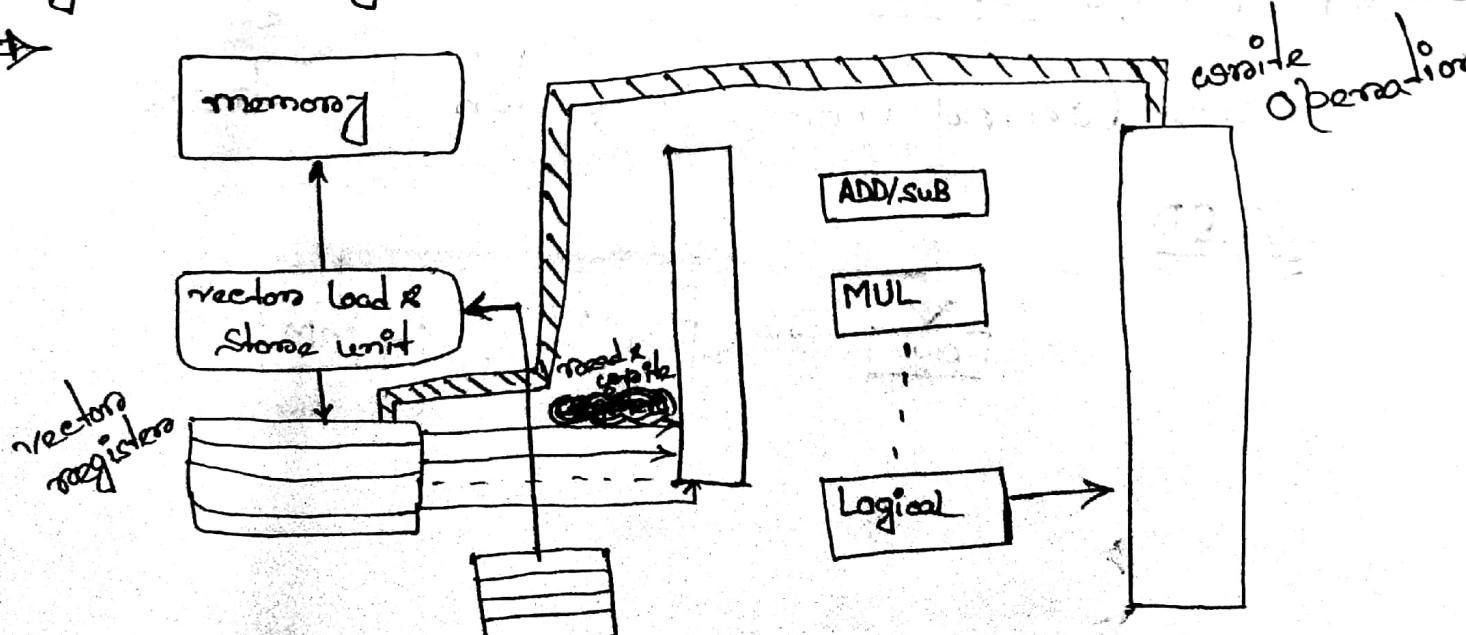
→ It's not widely used

Registers to Registers architecture

→

VLR ← <sup>vectors</sup> length rego

These two types of  
vectors architecture



vectors stride  $\Rightarrow$  it's a gap b/c 2 mem. which should be adjacent.

Strid<sup>h</sup> mining  $\Rightarrow$  The length of vectors is greater than vectors neg. 0, 1, 1, 0

■  $\text{Do } 100 \cdot I = 1, N$

where  $A(I) = B(I) + C(I)$

$100 \cdot B(I) = 2 * A(I+1)$

$\Rightarrow$  Initialize  $I=1$

Read  $B(I)$

Read  $C(I)$

ADD  $B(I), C(I)$

STORE  $A(I) \leftarrow B(I) + C(I)$

Read  $A(I+1)$

Multiply 2,  $A(I+1)$

STORE  $B(I) \leftarrow 2 * A(I+1)$

Increment  $I$  by 1

IF  $I \leq N$ , go to 10  
STOP

$$A(1:N) = B(1:N) + C(1:N)$$

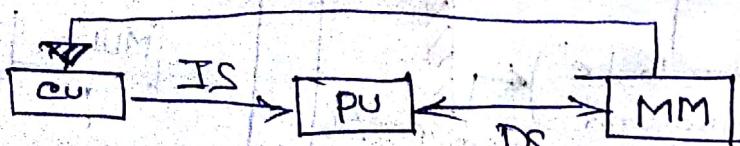
$$\text{TEMP}(1:N) = A(2:N+1)$$

$$B(1:N) = 2 * \text{TEMP}(1:N)$$

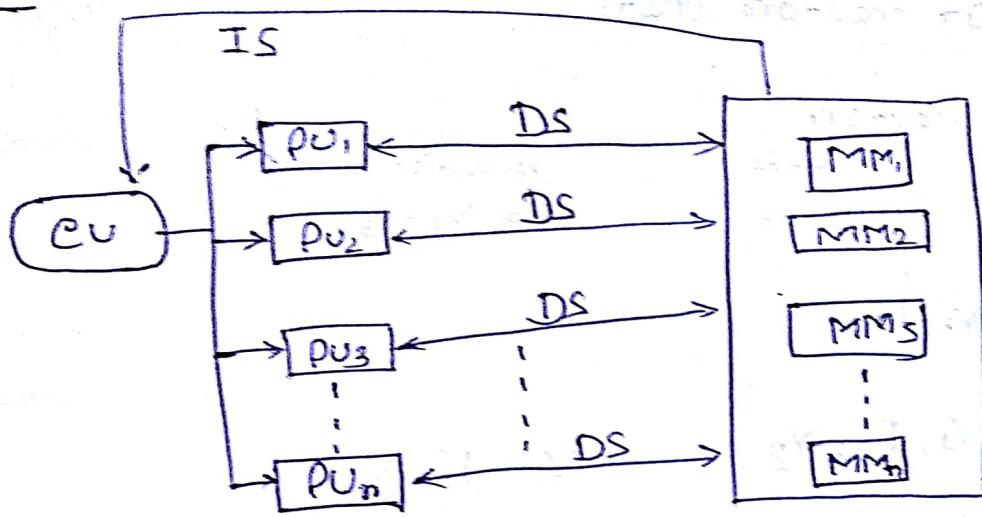
■ Flynn's Computer

CU  $\Rightarrow$  Control unit  $\Rightarrow$  fetch the Inst<sub>n</sub>

SISD

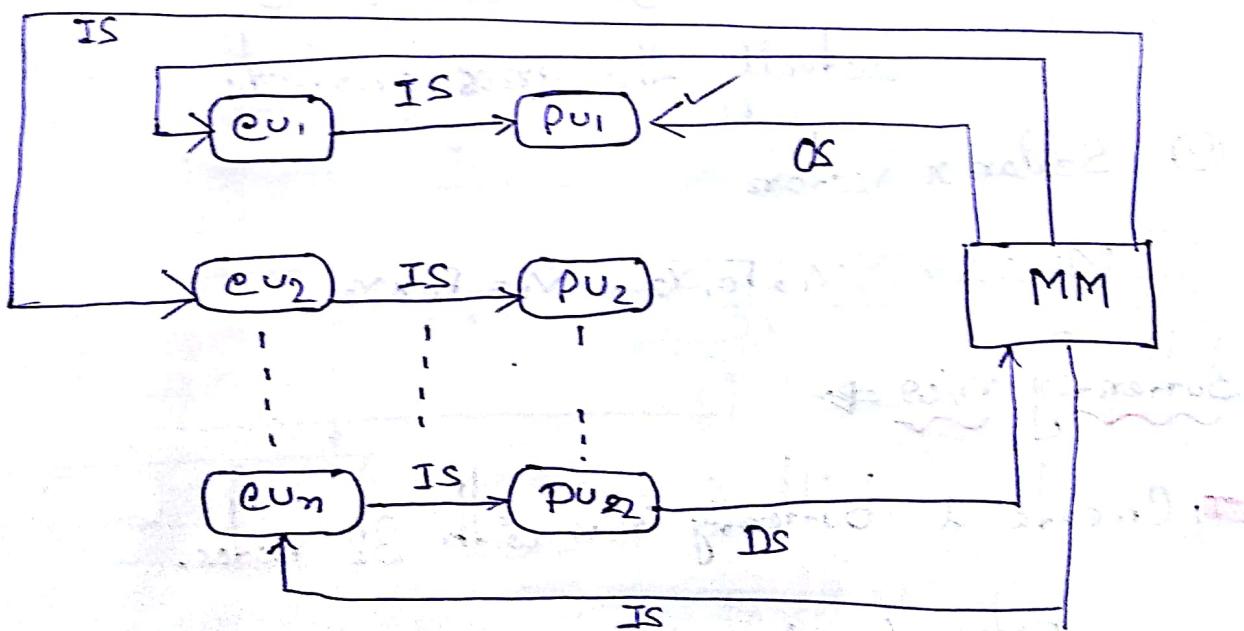


## SIMD

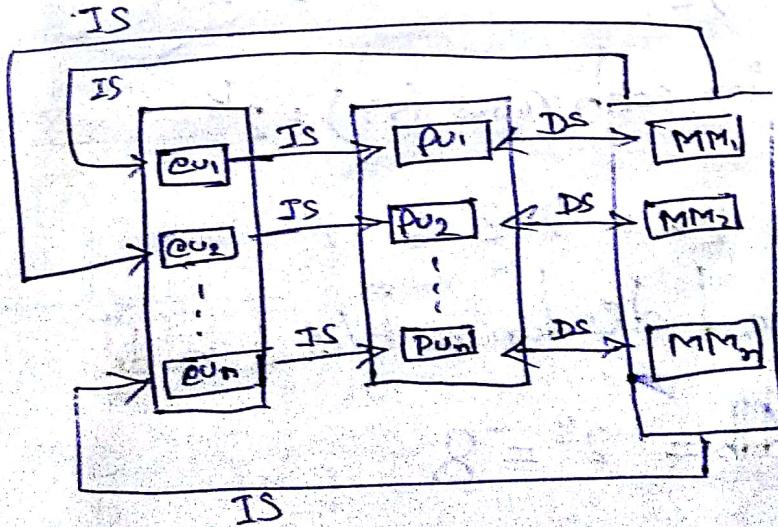


## MISD

Has no longer used practically



## MIMD



## \* 4 types of vector instrn

### ① vectors + vectors

inst<sup>n</sup> operands

ADDV  $v_1, v_2, v_3$

operation

$$v_1 = v_2 + v_3$$

### ② Scalars + Vectors

ADDSV  $v_1, F_0, v_2$

$$v_1 = F_0 + v_2$$

### ③ Vectors x Vectors

MULTY  $v_1, v_2, v_3$   $v_1 = v_2 \times v_3$

actually the cross product.

### ④ Scalars x vectors

MULTSV  $v_1, F_0, v_2$   $v_1 = F_0 \times v_2$

Butterfly  $\eta/\omega \Rightarrow$

Create a butterfly tree with 32 nodes.

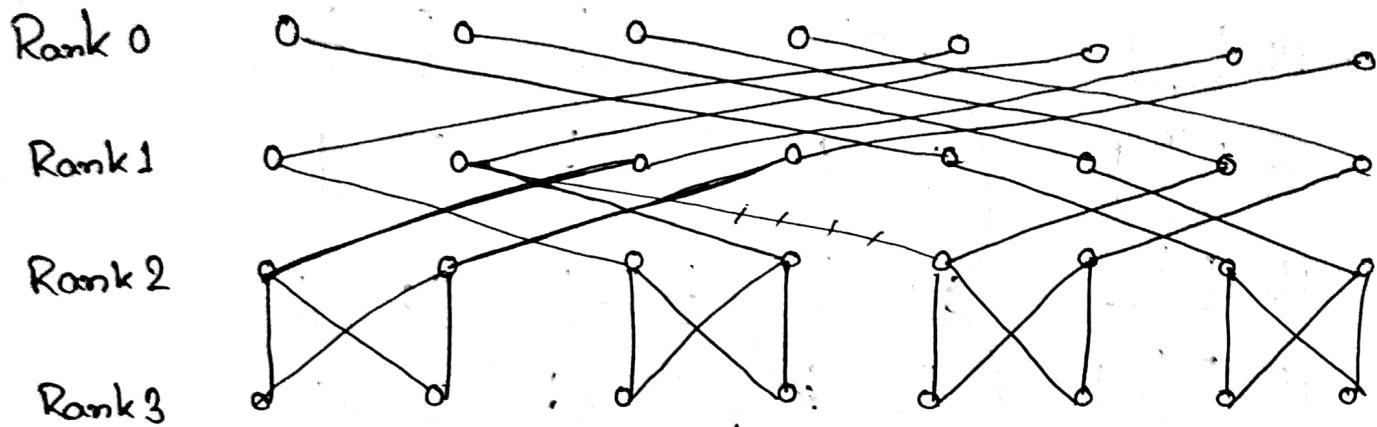
$$\text{nodes} = (K+1)2^K \quad \text{rows} = K+1$$

$$\text{elements in each row } (n) = 2^K$$

$$\text{diameter} = 2^K = 2 \times 3 = 6$$

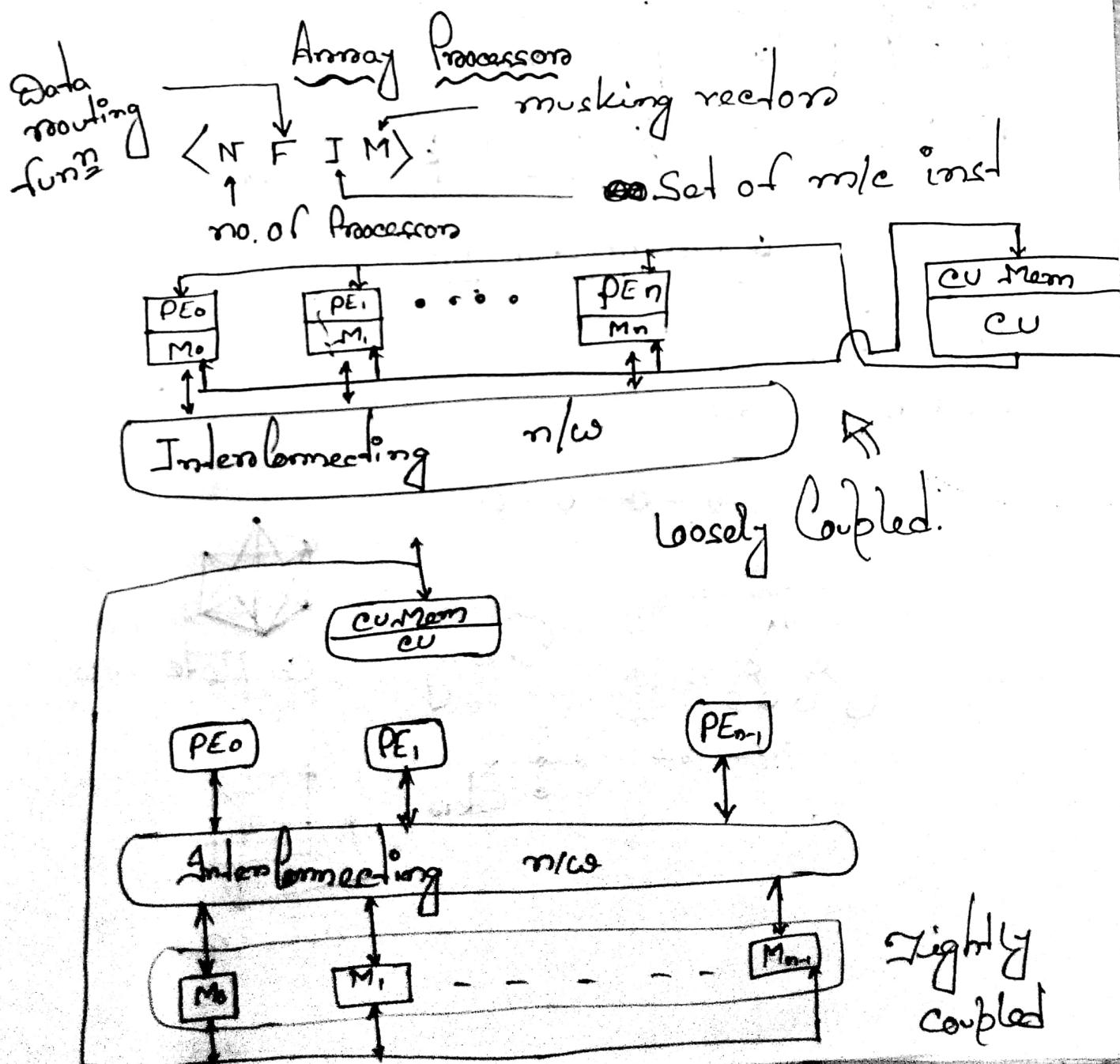
↑ Largest distance b/w 2 node

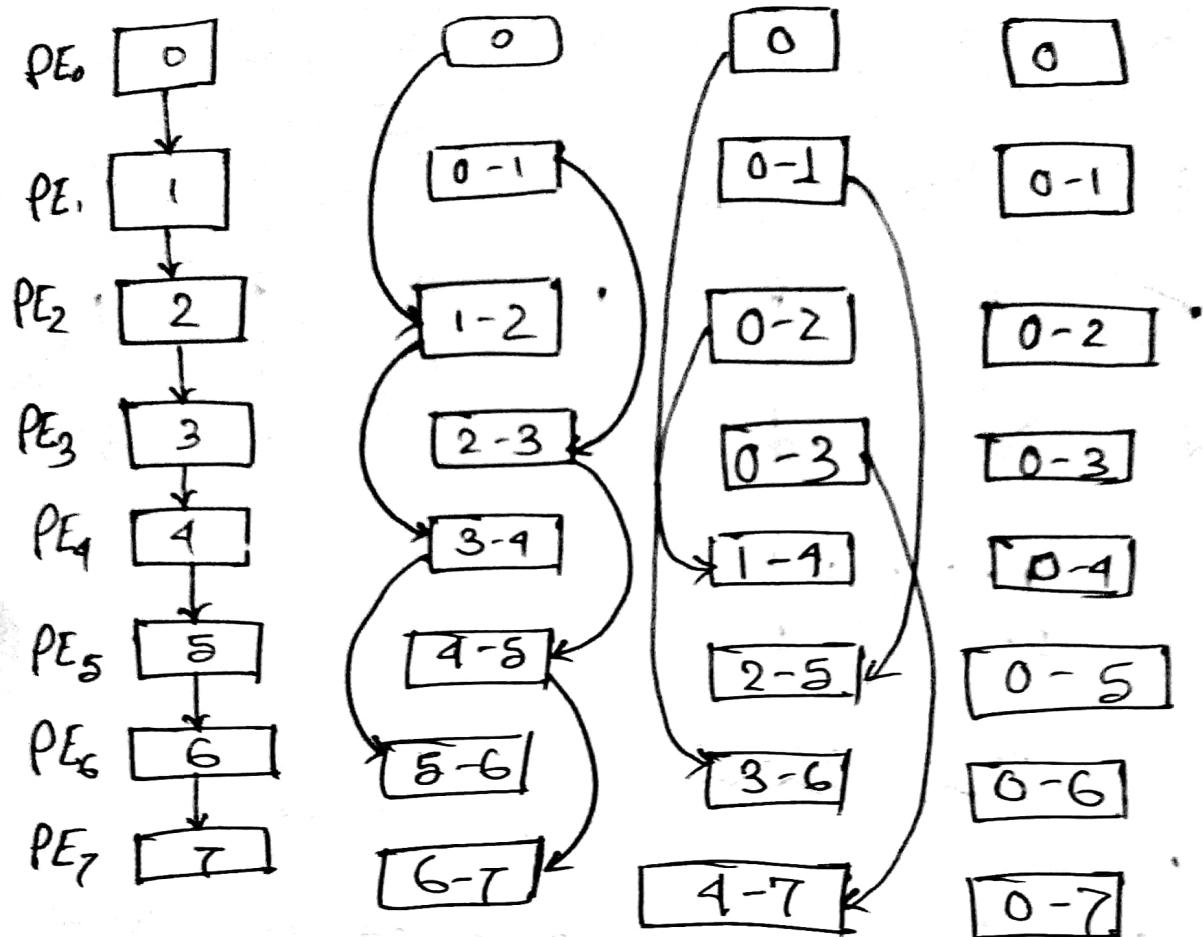
$$\text{bisection width} = 2^3 = 8$$



$$4 \rightarrow 2 \rightarrow 1$$

$$2^2 \quad 2^1 \quad 2^0$$

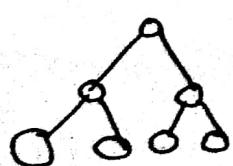




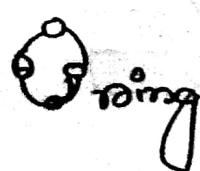
Interconnection n/w

Single stage

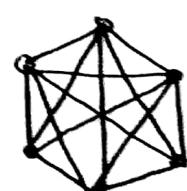
one dimensional n/w



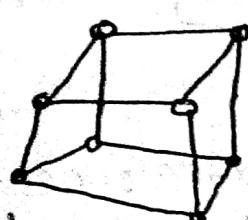
Tree n/w



Ring



Complete n/w

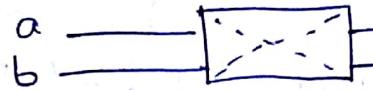


Star

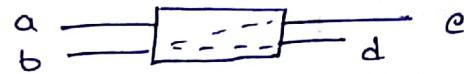
Cube n/w

Switch boxes  
( $2 \times 2$ )

① straight



② upper broadcast

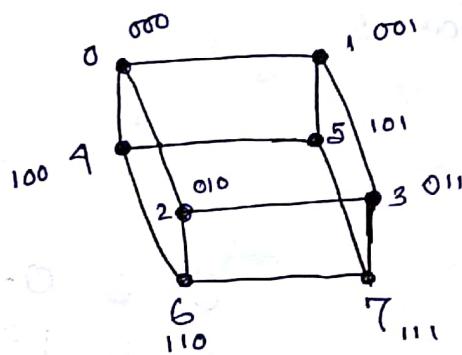


③ lower broadcast cart

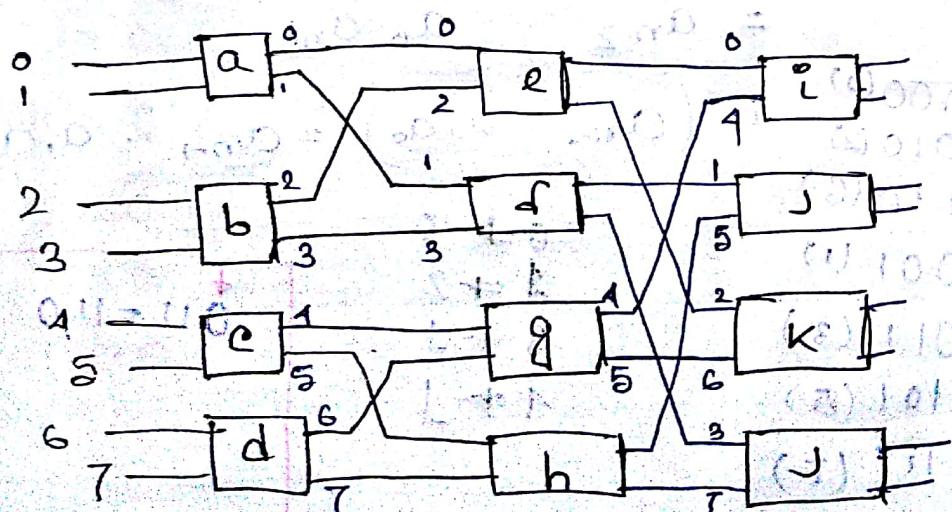
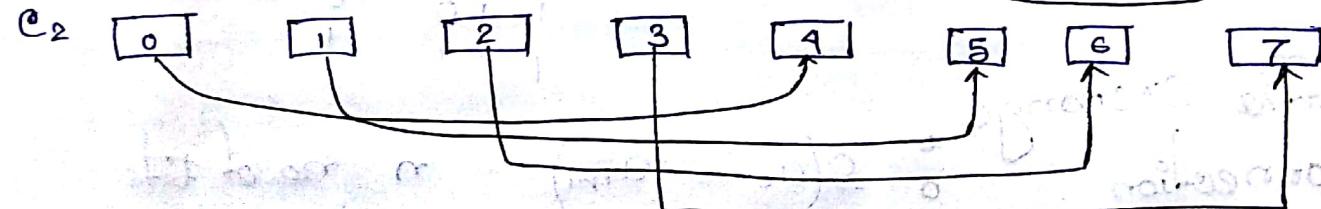
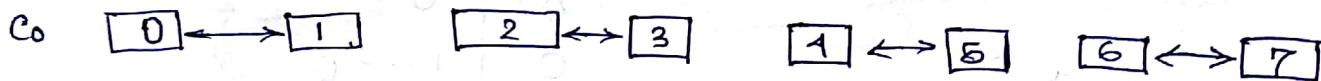


one-sided Switch box

Multistage n/cos



routing fun<sup>n</sup> = dimension  
= 3



ক্রমাগত  
0  $\rightarrow$  3  
1  $\rightarrow$  2  
3  $\rightarrow$  7

গ্রাম্য গ্রাম  
মিল (3, 7).  
গ্রাম পুরো ক্রম

# Behavioral model

Process (a,b)

begin

if ( $A = '0'$  and  $B = '0'$ ) Then

and otherwise  $C \leftarrow '0'$ .

else

$C \leftarrow '1'$ ;

end if.

end process

XOR

0	0	0
0	1	1
1	0	1
1	1	0

0 0 0 - 0

0 0 1 - 1

0 1 0 - 1

0 1 1 - 0

1 0 0 - 1

1 0 1 - 0

1 1 0 - 0

Shuffle exchange

Connection

$S(a_0 - a_{n-1})$       n no. of bits



$= a_{n-2} \dots a_0 a_{n-1}$

0 000  $\rightarrow$  000 (0)

$E(a_{n-1} \dots a_1 a_0) = a_{n-1} \dots a_1 a_0$

1 001  $\rightarrow$  010 (2)

2 011  $\rightarrow$  110 (6)

4 100  $\rightarrow$  001 (1)

0  $\rightarrow$  0

1  $\rightarrow$  2

3  $\rightarrow$  1

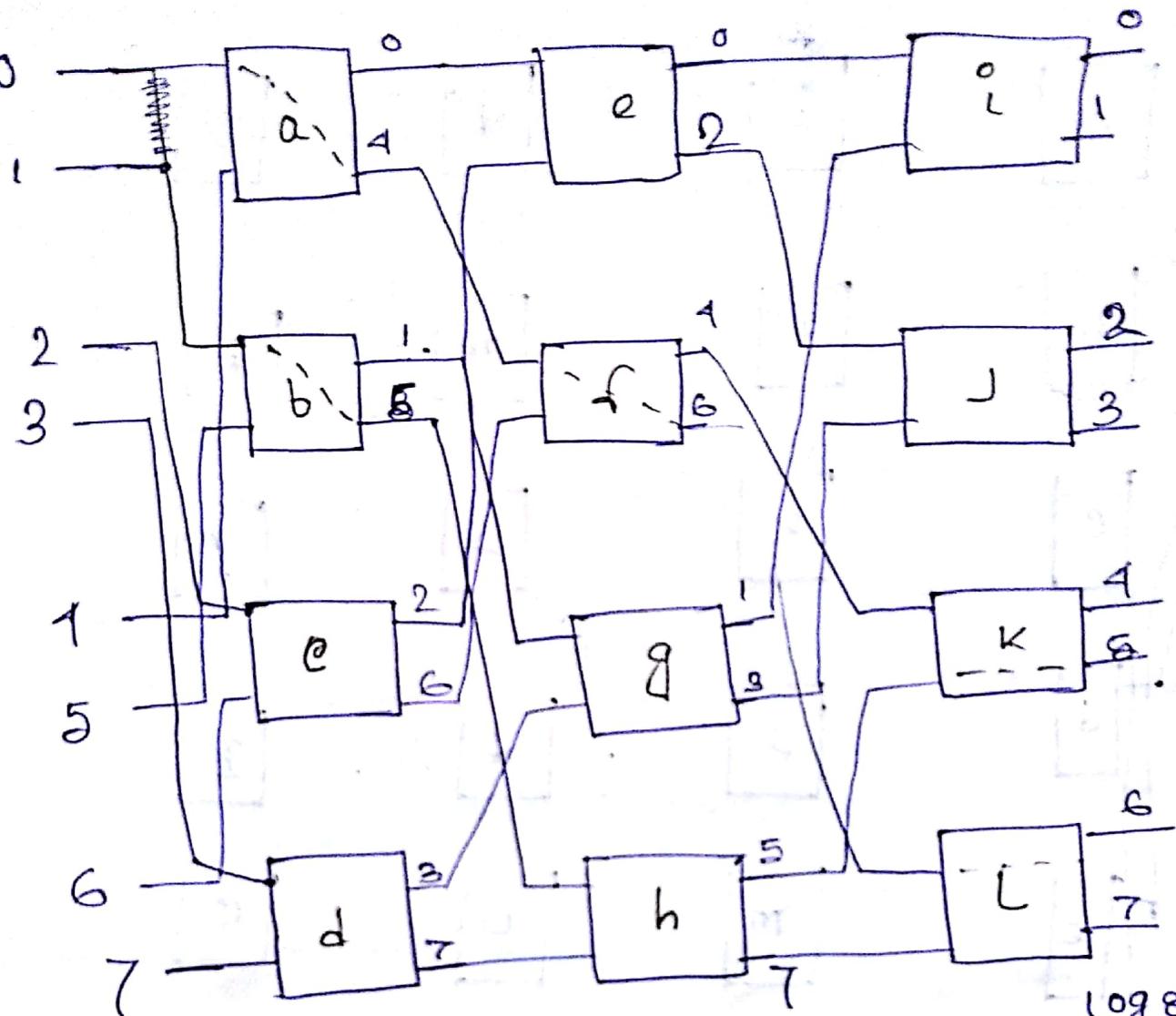
4  $\rightarrow$  1

5 101  $\rightarrow$  011 (3)

6 110  $\rightarrow$  101 (5)

7 111  $\rightarrow$  111 (7)

011 = 110



$$\log_2 8 = \log_2 2^3$$

### 16 i/p shuffle connection:

$$0 \rightarrow 0000 \Rightarrow 0000 (0)$$

$$1 \rightarrow 0001 \Rightarrow 0010 (2)$$

$$2 \rightarrow 0010 \Rightarrow 0100 (4)$$

$$3 \rightarrow 0011 \Rightarrow 0110 (6)$$

$$4 \rightarrow 0100 \Rightarrow 1000 (8)$$

$$5 \rightarrow 0101 \Rightarrow 1010 (10)$$

$$6 \rightarrow 0110 \Rightarrow 1100 (12)$$

$$7 \rightarrow 0111 \Rightarrow 1110 (14)$$

$$8 = 1000 = 0001 (1)$$

$$9 = 1001 = 0011 (3)$$

$$10 = 1010 = 0101 (5)$$

$$11 = 1011 = 0111 (7)$$

$$12 = 1100 = 1001 (9)$$

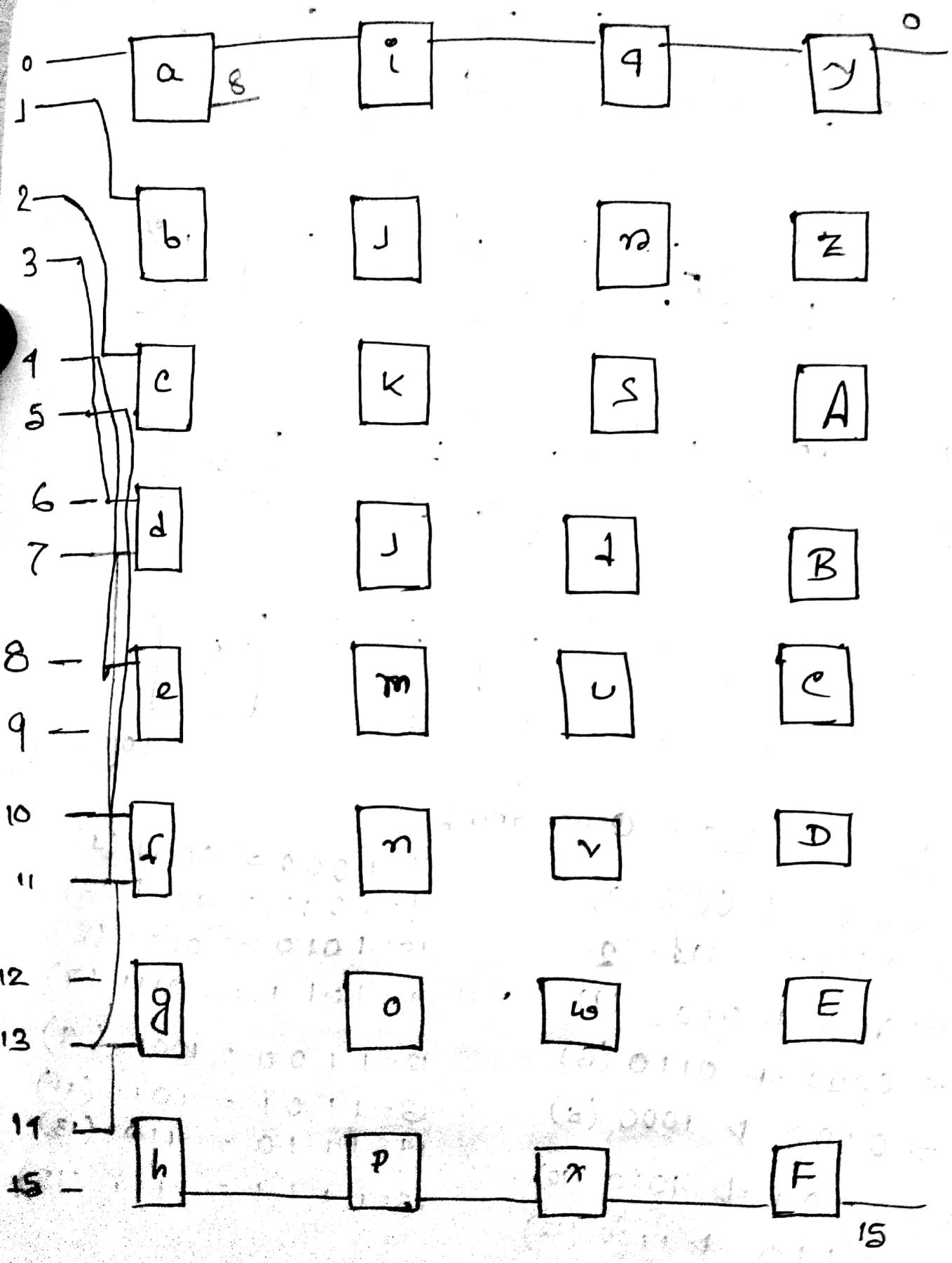
$$13 = 1101 = 1011 (11)$$

$$14 = 1110 = 1101 (13)$$

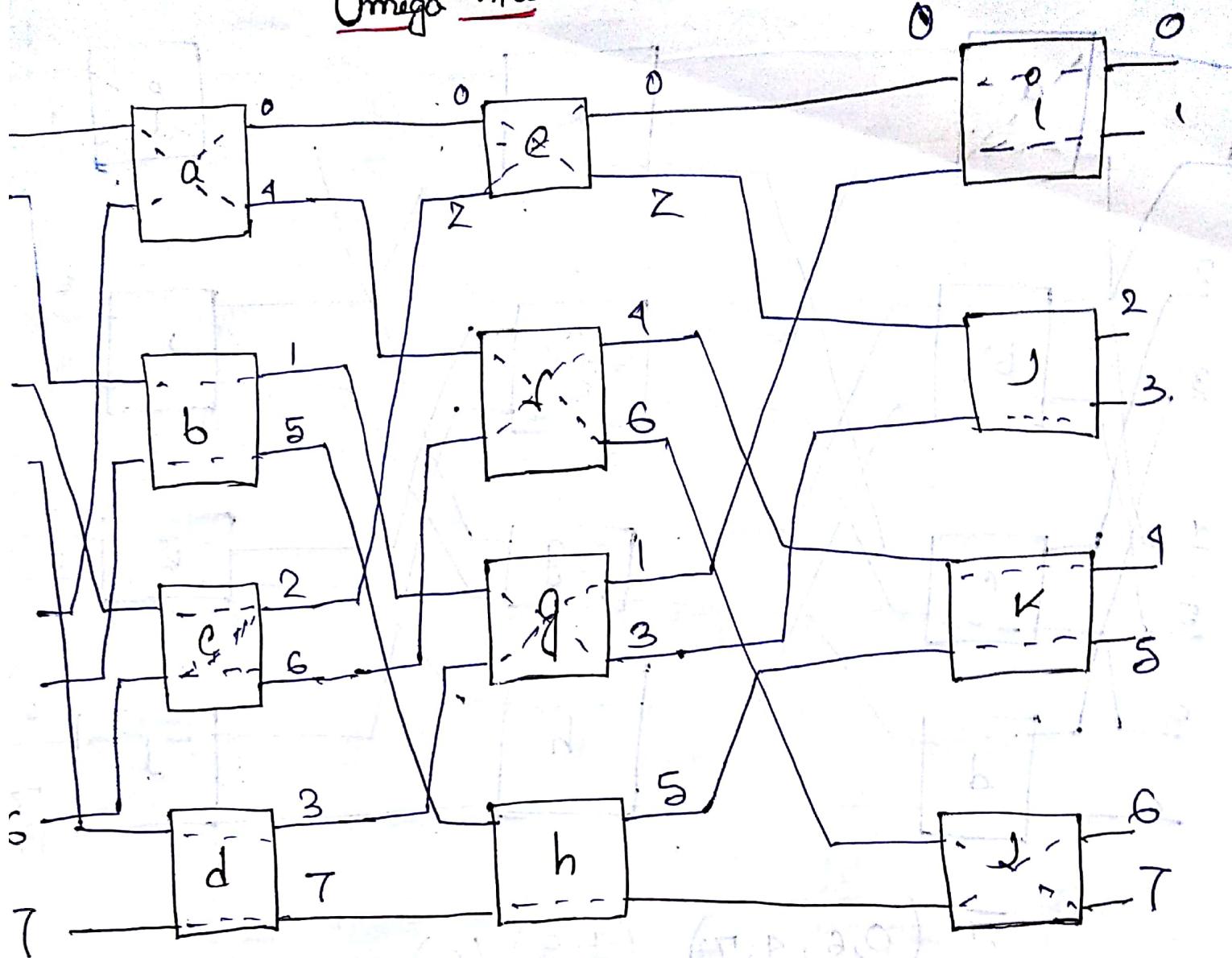
$$15 = 1111 = 1111 (15)$$

0-6 }  
1-5 }

2-5 } block  
Omega



Omega  $n/w$



$$\pi = (0, 7, 6, 4, 2) \quad (1, 3) \quad (5)$$

$$0 \rightarrow 7 \quad 7 \rightarrow 6 \quad 6 \rightarrow 4 \quad 5 \rightarrow 5 \\ 4 \rightarrow 2 \quad 2 \rightarrow 0 \quad 1 \rightarrow 3 \quad 3 \rightarrow 1$$

$$0 \rightarrow 0 \\ 1 \rightarrow 2$$

$$2 \rightarrow 4$$

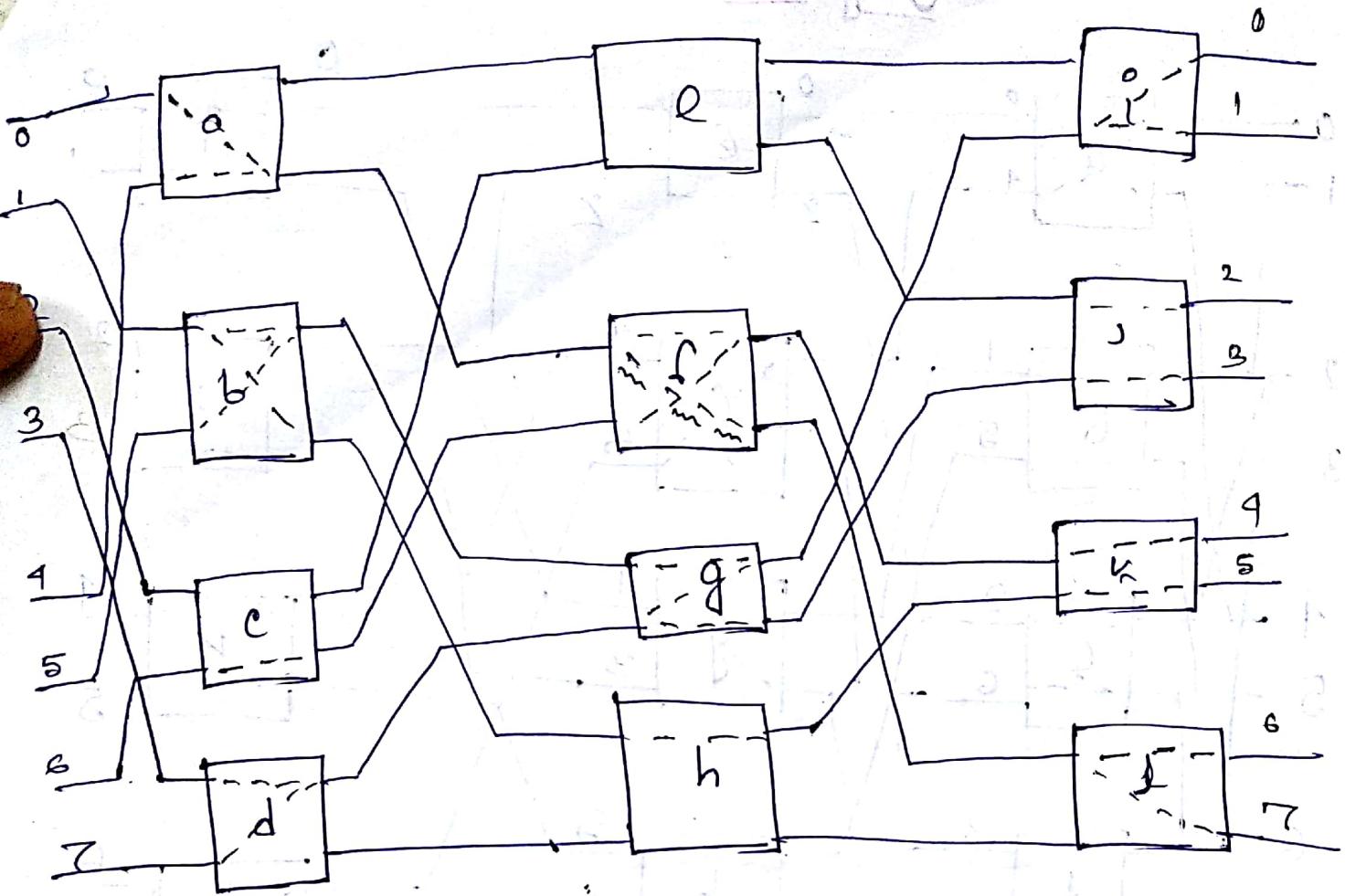
$$3 \rightarrow 6$$

$$4 \rightarrow 1$$

$$5 \rightarrow 3$$

⊗ 8 %

$$= \log_2 8 = \log_2 2^3 = \frac{3}{\text{column}}$$



$$\pi: (0, 6, 4, 7, 3) \quad (1, 5) \quad (2)$$

$$0 \rightarrow 6 \quad 6 \rightarrow 4 \quad 4 \rightarrow 7 \quad 7 \rightarrow 3 \quad 3 \rightarrow 0 \\ 1 \rightarrow 5 \quad 5 \rightarrow 1 \quad 2 \rightarrow 8 \quad 8 \rightarrow 2$$

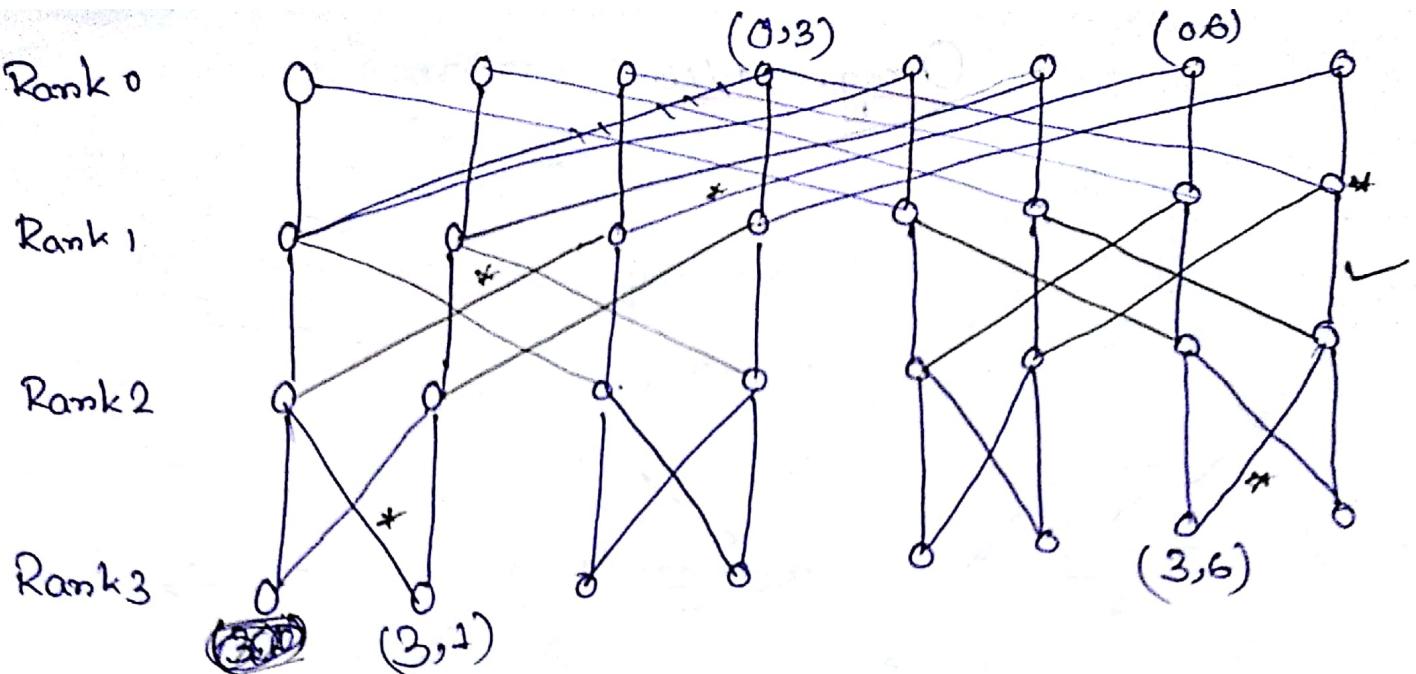
Butterfly n/c

$$k=3$$

$$(k+1)2^k = 32 \text{ nodes}$$

$$row = (3+1)^2 = 4$$

- |                         |                                 |
|-------------------------|---------------------------------|
| • Cube network          | • Shuffle exchange              |
| • Omega network         | • Butterfly network             |
| • Delta                 | • Systolic array                |
| • Bit-reversed shuffles | • Mesh Antidiagonal connections |



$$0 \leq i \leq k \quad 0 \leq j \leq 2^k$$

$i,j \rightarrow i > 0 \quad (i-1)j, (i+1)j$  ← for straight conn.

A node  $(i,j)$  will be Conn. to a node  $(i-1)m$

$m = \text{invert } i^{\text{th}} \text{ msb of } j$

$(3,0)$

$(2,0)$

$(3,1)$

$(2,1)$

$(3,2)$

$(2,2)$

2nd bit change  
from R.2

0003

0103

0013

0113

3rd bit  
change  
for R.3

000

001

010

011

Rank 1  
 $\downarrow$   
 $(1,0) \quad 000 \quad \left\{ \begin{array}{l} \\ \end{array} \right. \quad \textcircled{A}$

Send data  $(0,3) \rightarrow (3,6)$

011

110

$j-1$

$i-1$

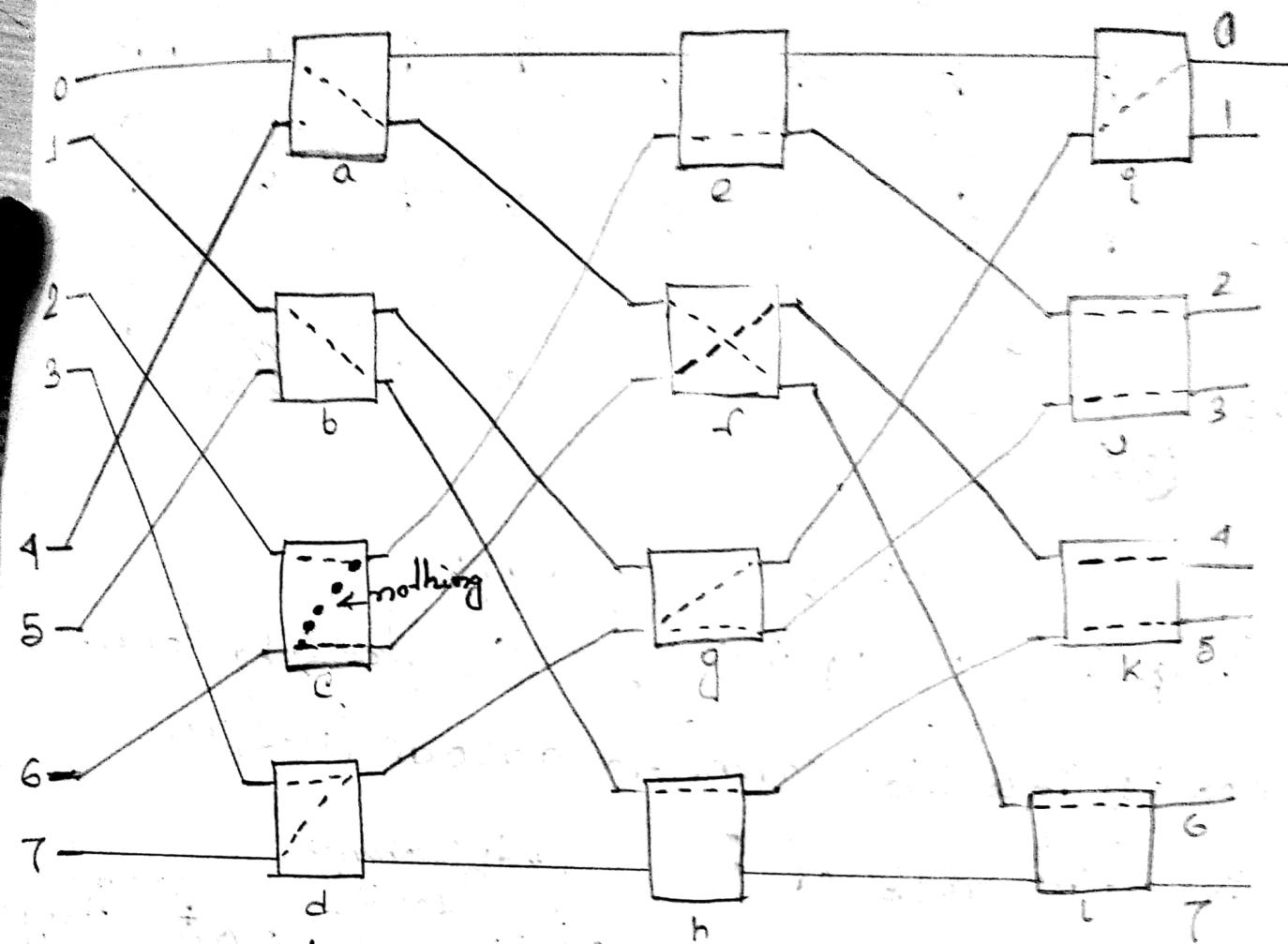
$i-0$

$(0,6) \rightarrow (3,1)$

110

001

Omega n/cw

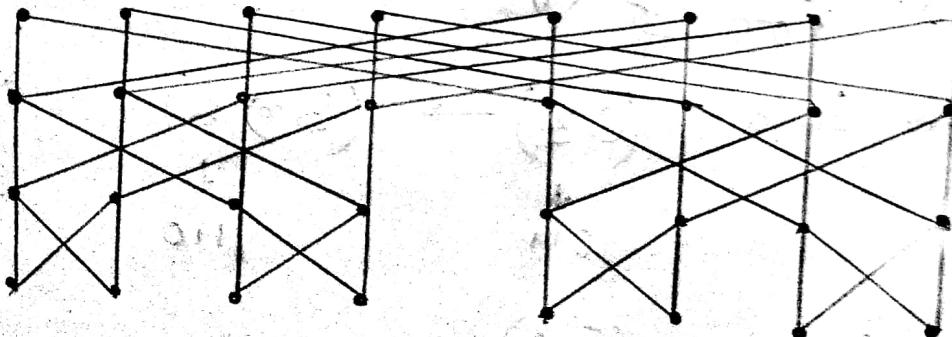


$$\pi, (0,6,4,7,3) (1,5)(2)$$

$0 \rightarrow 6$     $6 \rightarrow 9$     $4 \rightarrow 7$     $7 \rightarrow 3$     $3 \rightarrow 0$   
 $1 \rightarrow 5$     $5 \rightarrow 1$  X   X   Overlap  
 $2 \rightarrow 2$

Overleaf ଶ୍ରୀନାୟିକା ମହାପାତ୍ର,

Rank 0  
Rank 1  
Rank 2  
Rank 3



# Line mapping (Odd-even transposition)

for  $i=1$  to  $\lceil \frac{n}{2} \rceil$  do

    for all  $\text{PE}_j$  ( $0 \leq j < n/2$ ) do

        if  $\text{Even}(j)$  then

$$j' \leftarrow a_{j+1}$$

$$a_{j+1} \leftarrow \min(a_j, j')$$

$$a_j \leftarrow \max(a_j, j')$$

    endif

end for

$\text{PE}_0$	$\text{PE}_1$	$\text{PE}_2$	$\text{PE}_3$	$\text{PE}_4$	$\text{PE}_5$	$\text{PE}_6$	$\text{PE}_7$
7	8	14	8	1	9	10	5

$i=1$

3     8     9

9

$i \rightarrow 1, 10, 9$   
 $(j \leq n-1)$

$j=14$       $j=1$       $j=10$

odd

7     3     14     1     8     9     10     5  
 even      $j=3$       $j=1$       $j=9$       $j=5$   
 3     7     1     14     8     9     5     10

$i=2$

3     7     1     14     8     9     5     10  
 $j=1$       $j=8$       $j=5$

odd = 3 7 8 14 5 9 10  
 even      $j=1$       $j=8$       $j=5$       $j=10$   
 2 7 8 5 14 9 10

for  $i = 1 \dots 10$  [n/2]

for all  $PE_j$  ( $0 \leq j \leq n-1$ ) do

if  $j < n-1$  and odd( $j$ ) then

$$d_j \leftarrow a_{j+1}$$

$$a_{j+1} \leftarrow \max(a_j, d_j)$$

$$a_j \leftarrow \min(a_j, d_j)$$

end if

③ 7 ⑧ 5 ⑭ 9 10

$$d_j = 7 \quad d_j = 5 \quad d_j = 9$$

① 3 ⑦ 5 ⑧ 9 ⑭ 10

$$d_j = 3 \quad d_j = 5 \quad d_j = 9 \quad d_j = 10$$

③ 5 ⑦ ⑧ ⑨ 10 ⑭

$$d_j = 5 \quad d_j = 8 \quad d_j = 10$$

① 3 ⑤ ⑦ ⑧ 9 ⑩ ⑭

$$d_j = 3 \quad d_j = 7 \quad d_j = 9 \quad d_j = 14$$

1 3 5 7 8 9 10 14

// odd वाले अंगड़े अंगड़े last ele. Consider  
एवेन, इन्हें even, एवेन, इन्हें

odd

even

$a_j = \text{Position element}$

odd pos.  
 $\downarrow$   
 $a \rightarrow d$   
 $\downarrow$   
start

Linear array [odd - even / Bubble Sort]

Given array: 3, 2, 3, 8, 5, 6, 4, 1

Total length = 8

So hence will  $8/2 = 4$  no. of step

1. odd: 3 2 3 8 5 6 4 1

$$a_j = 2, 8, 6 \quad b_j = 3, 5, 4$$

even: 3 2 3 5 8 4 6 1 PEo ... So hence first 3 considered

$$a_j = 3, 3, 8, 6 \quad b_j = 2, 5, 4, 1$$

2, 3, 3, 5, 4, 8, 1, 6

2. odd:

$$a_j = 3, 5, 8 \quad b_j = 3, 4, 1$$

2, 3, 3, 4, 5, 1, 8, 6

even:

$$a_j = 2, 3, 5, 8 \quad b_j = 3, 4, 1, 6$$

2, 3, 3, 4, 1, 5, 6, 8

3. odd:

$$a_j = 3, 4, 5 \quad b_j = 3, 1, 6$$

2, 3, 3, 1, 4, 5, 6, 8

even:

$$a_j = 2, 3, 4, 6 \quad b_j = 3, 1, 5, 8$$

2, 3, 1, 3, 4, 5, 6, 8

4. odd:

$$a_j = 3, 3, 5 \quad b_j = 1, 4, 6$$

2, 1, 3, 4, 5, 6, 8

even:

$$a_1 = 2, 3, 4, 6 \quad j_1 = 1, 3, 5, 8$$

$$1, 2, 3, 3, 4, 5, 6, 8$$

Delta network:

Dimensions of switch box  $a^n b^n$  here

It's a multi stage m/w.

$$\textcircled{1} 4^2 \times 3^2$$

$$\textcircled{2} 2^3 \times 2^3$$

$$\textcircled{1} 4^2 \times 3^2$$

$$a = 4$$

$$b = 3$$

$$n = \text{no of stages} = 2$$

1st stage:

$$\text{no. of switch box} = a^{n-1}$$

Last stage:

$$\text{no. of switch box} = b^{n-1}$$

if  $a = b$  then total no. of switch boxes

$$n \cdot b^{n-1}$$

if

$$a \neq b$$

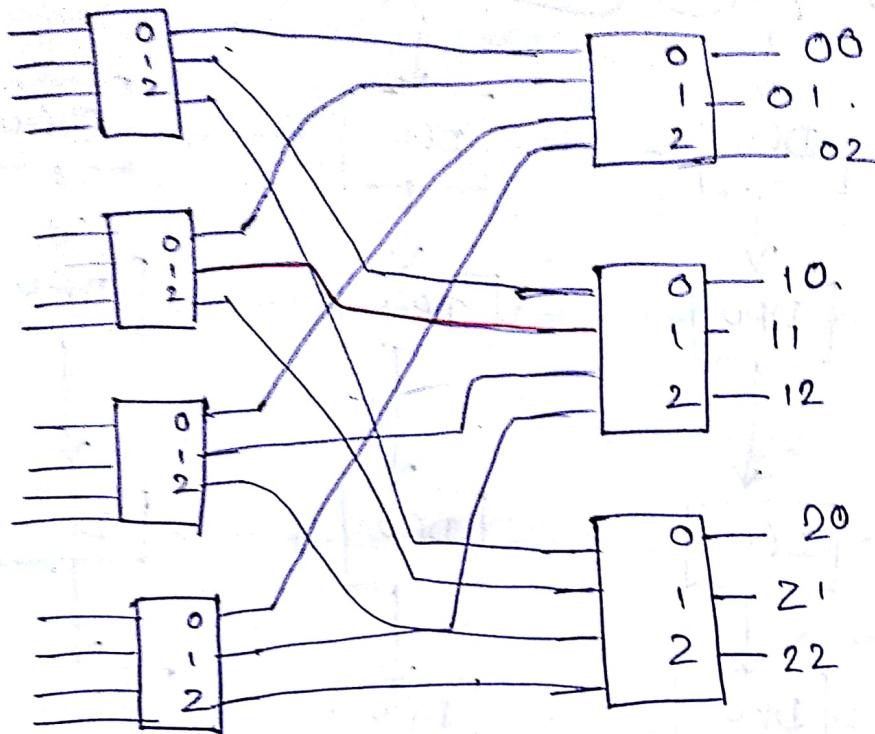
$$\frac{a^n - b^n}{a - b}$$

Here total s. box = 7

$$n = 2 = \text{Stage}$$

1st stage: 1

Last stage: 3



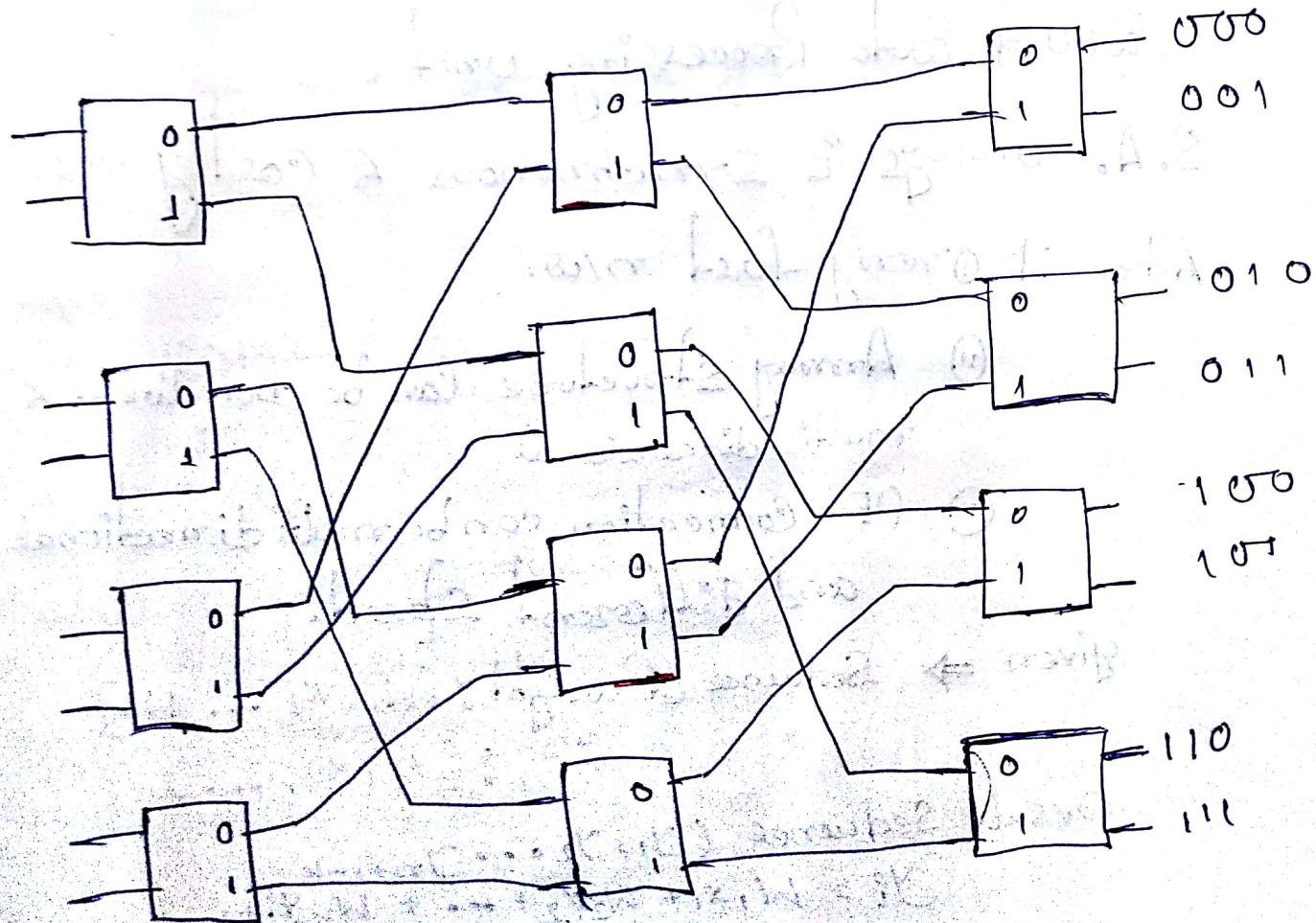
প্রথম ০-০, ১-১

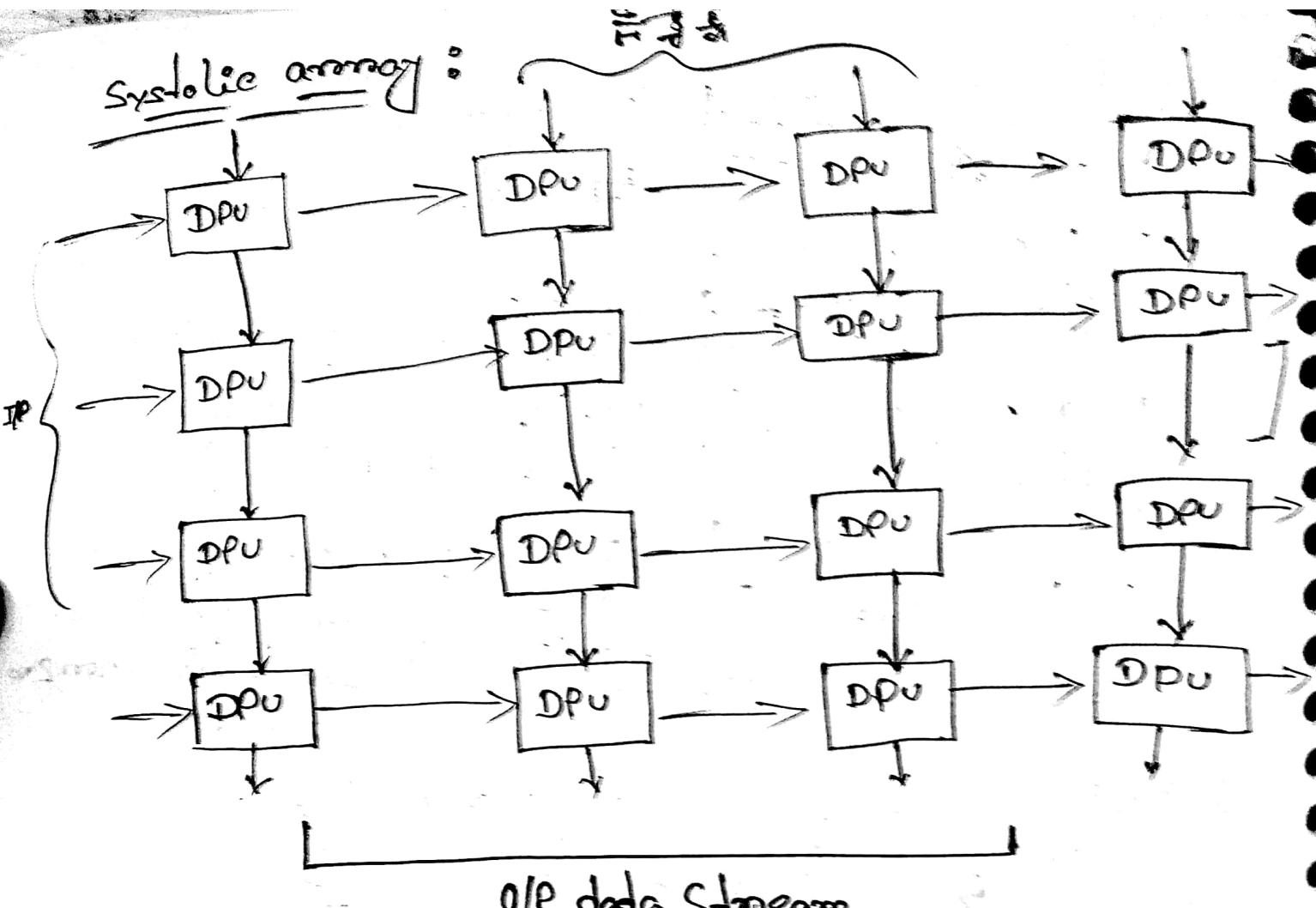
তাহে দ্বিতীয় পর্যায়  
১ম এবং ২য় পর্যায় তাহে ২য়  
এবং ৩য় পর্যায়

$$\text{ii) } 2^3 \times 2^3$$

$$\begin{aligned} &\text{total S. Box} \\ &\Rightarrow 3 \times 2^2 = 12 \end{aligned}$$

$n = 3 = \text{Stage}$





DPU  $\Rightarrow$  Data Processing unit

S.A. always is synchronous & costly one.

Adv.  $\Rightarrow$  very fast n/w.

(i) Array structure can be non-linear & multi dimensional

(ii) PE Connection can be multidirectional and different speed.

Given  $\Rightarrow$  Sequence of weight :  $\{w_1, w_2 \dots w_k\}$

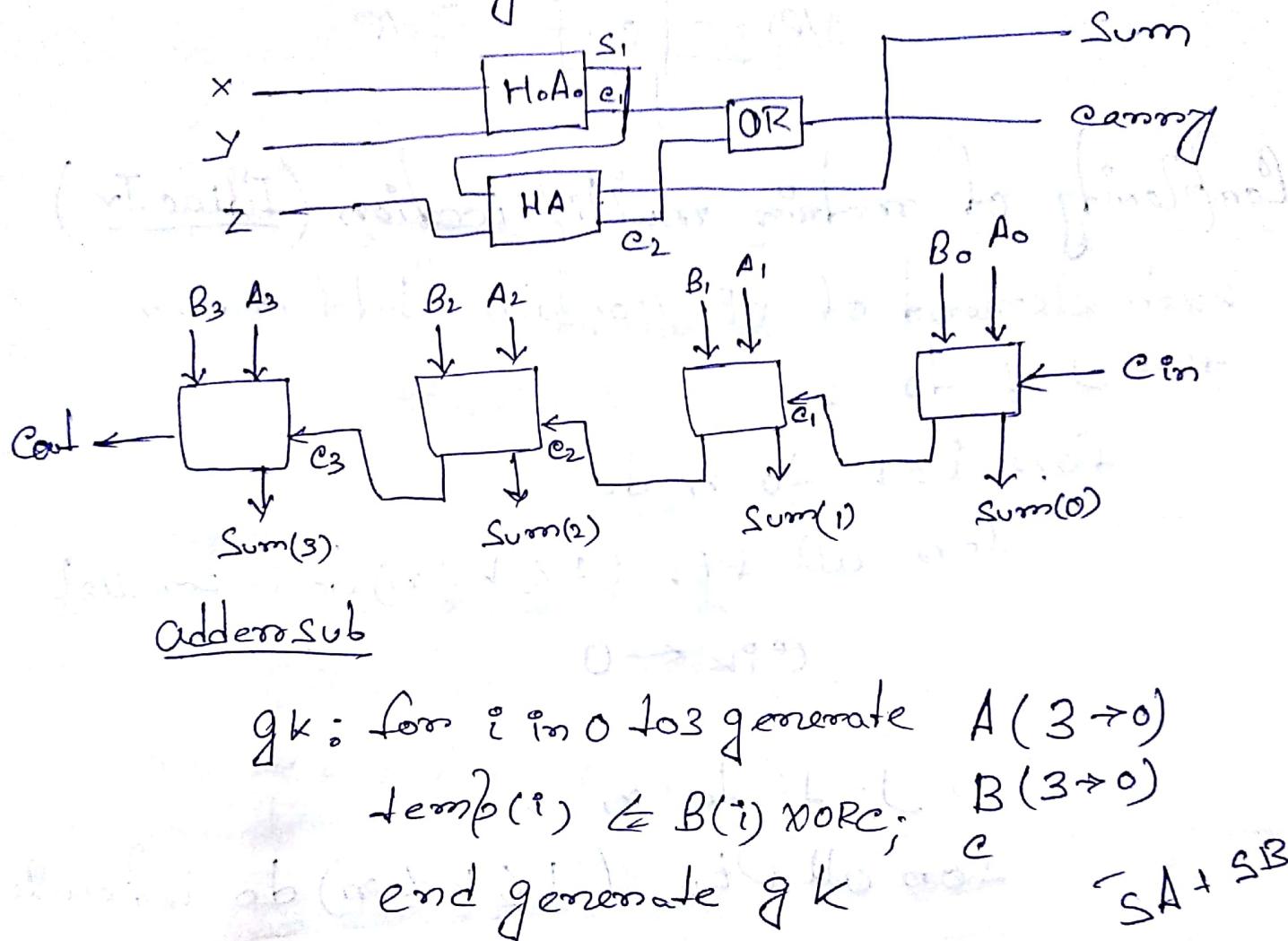
" " I/P :  $\{x_1, x_2, \dots x_n\}$

Result Sequence ( $y_1, y_2, \dots y_{n+k-1}$ )

$$y_i = w_1 x_1 + w_2 x_2 + \dots + w_k x_{i+k-1}$$

### Lab 31

Create a f/f/a using 2 H/a



### addersub

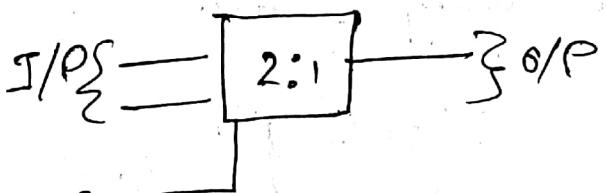
gk: for  $i$  in 0 to 3 generate  $A(3 \rightarrow 0)$   
 $temp(i) \leftarrow B(i) \oplus C$ ;  $B(3 \rightarrow 0)$   
 end generate gk

$$SA + SB$$

begin

$y \leftarrow A(0)$  when  $S = "00"$  else  $A(1)$  when  $S = "01"$  else  $A(2)$  when  
 $S = "10"$  else  $A(3)$  when  $S = "11"$ ;

end datapath;



## Complexity of matrix multiplication (Illiad IV)

each element of PE contains total column.

for  $J = 1 \text{ to } x$

for  $i = 1 \text{ to } n \text{ do}$

for all  $PE_k (1 \leq k \leq n)$  do in parallel

$$C_{ik} \leftarrow 0$$

for  $J = 1 \text{ to } n$

for all  $PE_k (1 \leq k \leq n)$  do in parallel

$$C_{ik} \leftarrow C_{ik} + a_{ik}^{\circ} \times b_{jk}$$

end for)

end for i

So Complexity is  $O(n^2)$

### Phase - I

for  $k = 1 \text{ to } (n-1) \text{ do}$

for all PEs in parallel do

if  $i > k$  then

rotate  $A(i,j) \in A(i, j+1)$

endif

if  $j > k$  then

rotate  $B(i,j) \leftarrow B(i+1,j)$

endif

end for

## ~~Phase-I~~ Phase-II

For all  $PE_{i,j}$  in Parallel do

$$C(i,j) \leftarrow 0$$

end for

for  $k=1$  to  $n$  do

for all  $PE_{i,j}$  in Parallel do

$$C(i,j) = C(i,j) + A(i,j) * B(i,j)$$

rotate  $A(i,j) \leftarrow A(i,j+1)$

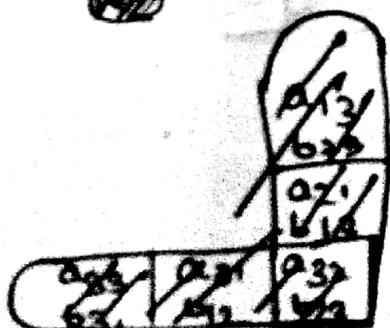
rotate  $B(i,j) \leftarrow B(i+1,j)$

end for (Parallel)

end for ( $K$ )

end

④



		a <sub>13</sub> b <sub>23</sub>
		a <sub>21</sub> b <sub>13</sub>
		a <sub>31</sub> b <sub>32</sub>
a <sub>11</sub> b <sub>11</sub>	a <sub>12</sub> b <sub>12</sub>	a <sub>13</sub> b <sub>13</sub>
a <sub>21</sub> b <sub>21</sub>	a <sub>22</sub> b <sub>22</sub>	a <sub>23</sub> b <sub>23</sub>
a <sub>31</sub> b <sub>31</sub>	a <sub>32</sub> b <sub>32</sub>	a <sub>33</sub> b <sub>33</sub>

$K=2$  (array change  
only 3 row)

			K=1
			3
a <sub>11</sub> b <sub>11</sub>	a <sub>12</sub> b <sub>22</sub>	a <sub>13</sub> b <sub>23</sub>	a <sub>13</sub> b <sub>23</sub>
a <sub>22</sub> b <sub>22</sub>	a <sub>23</sub> b <sub>32</sub>	a <sub>21</sub> b <sub>33</sub>	a <sub>21</sub> b <sub>33</sub>
a <sub>32</sub> b <sub>32</sub>	a <sub>33</sub> b <sub>33</sub>	a <sub>31</sub> b <sub>12</sub>	a <sub>31</sub> b <sub>12</sub>
a <sub>12</sub> b <sub>12</sub>			2
a <sub>21</sub> b <sub>21</sub>			1

$$i = 2, 3 \quad j = 2, 3$$

str 2, 3 rows &

## Phase II

(k=3)

$a_{12}$	$a_{13}$	$a_{11}$
$b_{21}$	$b_{32}$	$b_{13}$
$a_{23}$	$a_{21}$	$a_{22}$
$b_{31}$	$b_{12}$	$b_{23}$



$0+$ $a_{11} * b_{11}$	$0+$ $a_{12} * b_{22}$	$0+$ $a_{13} * b_{33}$
$0+$ $a_{22} * b_{21}$	$0+$ $a_{23} * b_{32}$	$0+$ $a_{21} * b_{13}$
$0+$ $a_{33} * b_{31}$	$0+$ $a_{31} * b_{12}$	$0+$ $a_{32} * b_{23}$



$" +$ $a_{12} * b_{21}$	$" +$ $a_{13} * b_{32}$	$" +$ $a_{11} * b_{13}$
$" +$ $+ a_{23} * b_{31}$	$" +$ $a_{21} * b_{12}$	$0 + "$ $+ a_{22} * b_{23}$
$" +$ $+ a_{31} * b_{11}$	$" +$ $a_{32} * b_{22}$	$" +$ $a_{33} * b_{33}$

A

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \times \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Ques:

$$\begin{bmatrix} 2 & 3 & 7 \\ 23 & 55 & 7 \\ 8 & 90 & 10 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

Phase I

2, 1 (1,1)	3, 2 (1,2)	7, 3 (1,3)
23, (2,1)	55, (2,2)	7, (2,3)
8, (3,1)	90, (3,2)	10, (3,3)

$K=1$

2 1	3 5	41 6
55 1	7 8	23 9
90 7	10 2	8 3

$K=2$

2 1	3 5	41 9
55 1	7 8	23 3
10 7	8 2	90 6

Phase II

$K=1$

$0+(2 \times 1)$	$0+(3 \times 5)$	$0+(41 \times 6)$
$0+(55 \times 1)$	$0+(7 \times 8)$	$0+(23 \times 9)$
$0+(90 \times 1)$	$0+(10 \times 2)$	$0+(8 \times 3)$

41, 9

$K=2$

3 "	4 "	23 "
" 7	8 "	55 "
8 1	90 5	10 9

(K=2)

<del><math>(3 \times 1) + (3 \times 4) + (4 \times 7)</math></del>	<del><math>(3 \times 5) (4 \times 8)</math></del>	<del><math>(4 \times 6) (2 \times 3)</math></del>
	<del><math>(2 \times 2) (2 \times 5)</math></del>	
$(10 \times 7)$	$(8 \times 1)$	$(8 \times 3) (10 \times 9)$



$(2 \times 1)$	$(3 \times 5)$	$(4 \times 9)$
$(55 \times 1)$	$(7 \times 8)$	$(23 \times 3)$
$(10 \times 7)$	$(8 \times 2)$	$(90 \times 6)$

$K=1$

$(3 \times 4)$	$(4 \times 8)$	$(2 \times 3)$
$(7 \times 7)$	$(23 \times 2)$	$(55 \times 6)$
$(8 \times 1)$	$(90 \times 5)$	$(10 \times 9)$

$K=2$

$(4 \times 7)$	$(2 \times 2)$	$(3 \times 6)$
$(23 \times 1)$	$(55 \times 5)$	$(7 \times 9)$
$(90 \times 1)$	$(10 \times 8)$	$(8 \times 3)$

$K=3$

- ⊕ Best Policy math calculation এর মানের  
পর ইন্দোর প্রেস / সেন্টেন্স এবং

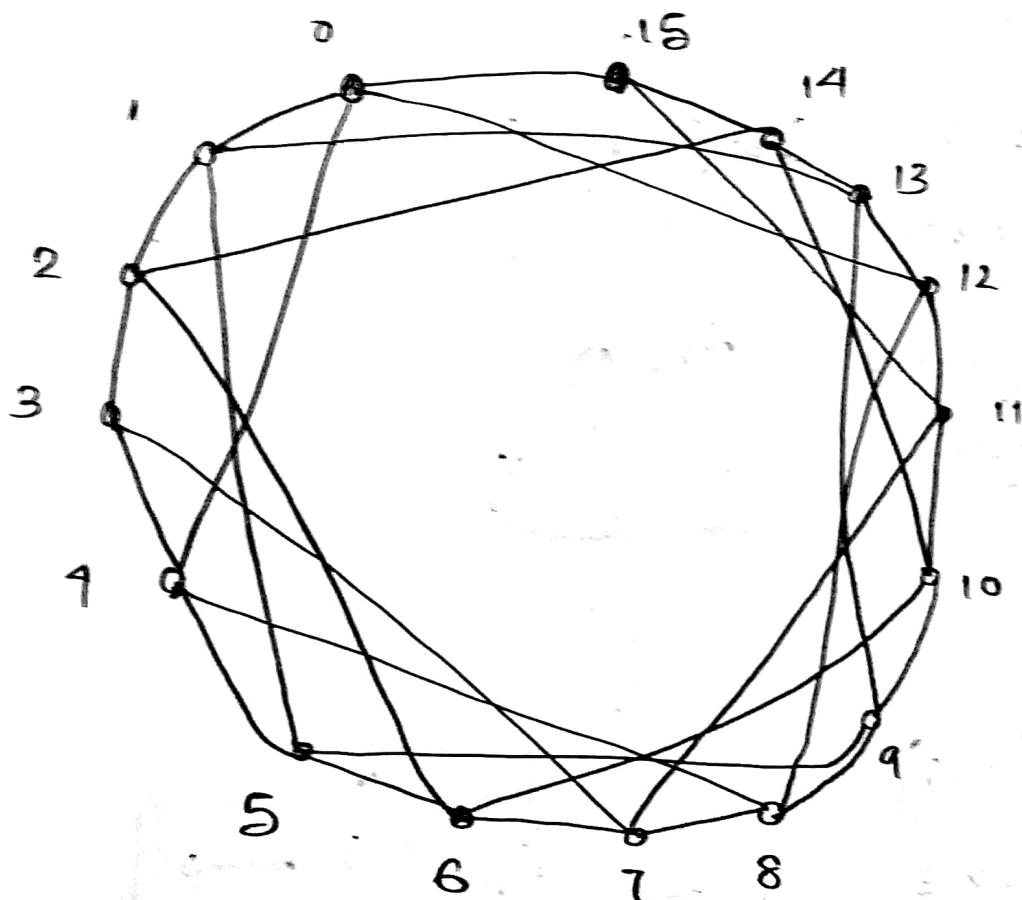
Max interconnection n/w

$$D = \sqrt{n}$$

value  $n = 16$

$$D = 4$$

beside 2 & skip 4

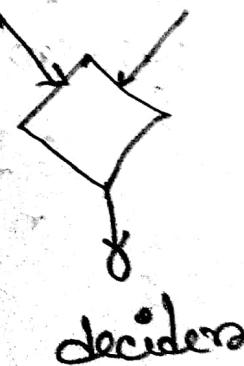
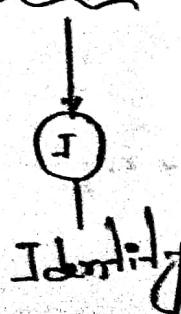
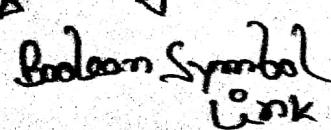


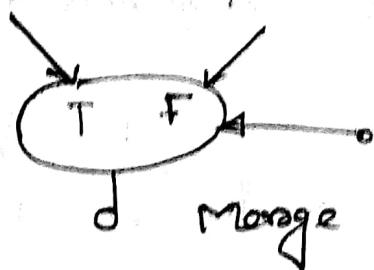
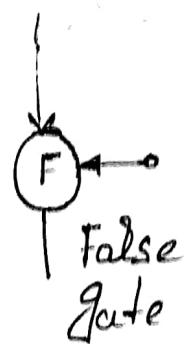
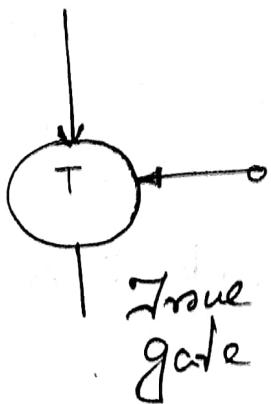
### Barcode Shiften

$$B_i(j) = (j + 2^i) \bmod N$$

$$B_{-i}(j) = (j - 2^i) \bmod N$$

### data flow architecture





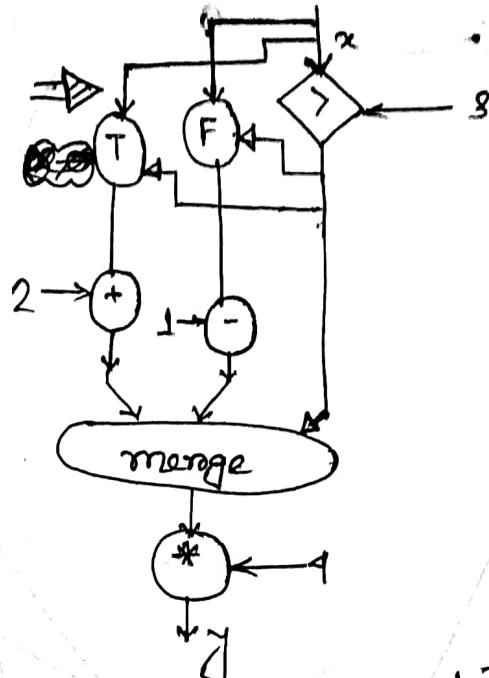
$\text{if } (x > 3)$

$$y = x + 2$$

else

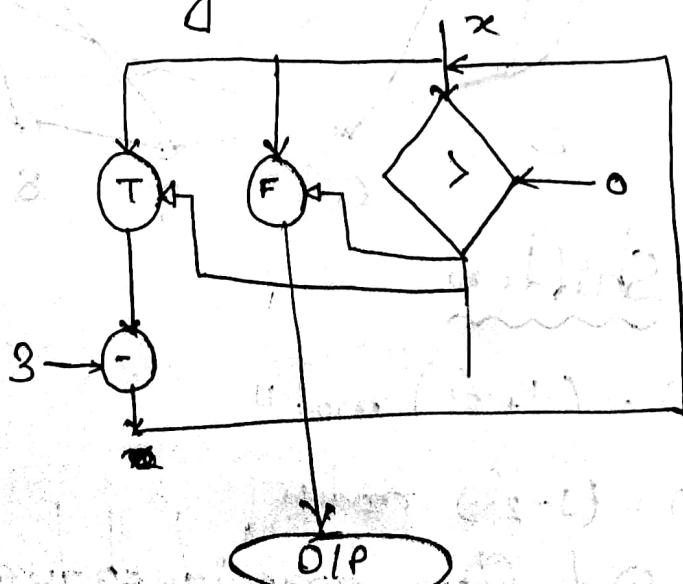
$$y = x - 1$$

$$y = y * 4$$



while ( $x > 0$ )

$$x = x - 3$$



$$\text{fact} = 1$$

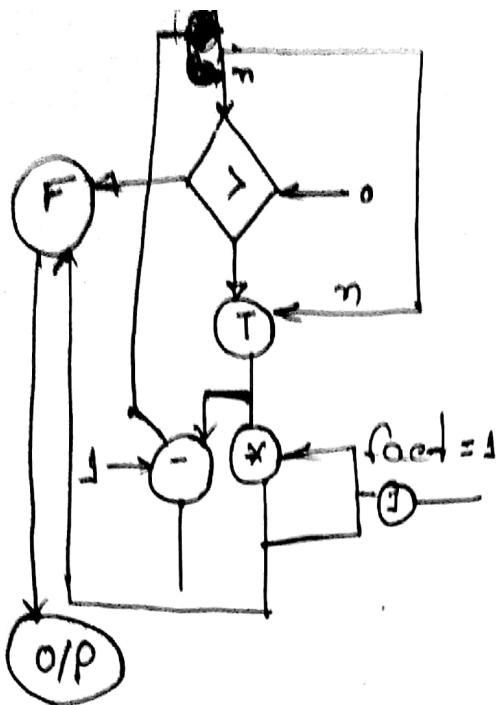
while ( $n > 0$ )

{

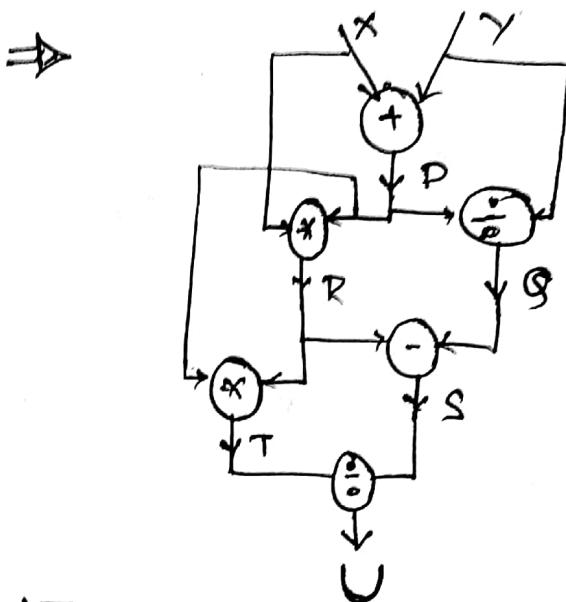
$$\text{fact} = \text{fact} * n;$$

$$n = n - 1;$$

}



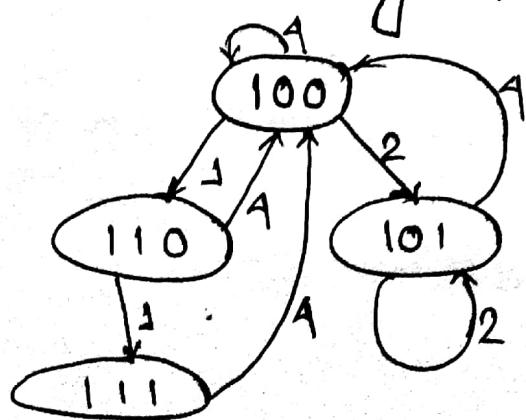
$$\begin{aligned}
 P &= X+Y \\
 Q &= P-Y \\
 R &= X*P \\
 S &= R-Q \\
 T &= R*P \\
 U &= S-T
 \end{aligned}$$



Collision Vectors = 100

→ Hence forbidden latency = 3

Permissible latency 2, 1



$$\begin{array}{r}
 \begin{array}{c} 1'S \\ 100 \\ 010 \\ \hline 110 \end{array} &
 \begin{array}{c} 2'S \\ 100 \\ 001 \\ \hline 101 \end{array} \\
 \begin{array}{c} 011 \\ \hline 100 \\ \hline 111 \end{array} &
 \begin{array}{c} 001 \\ 100 \\ \hline 101 \end{array}
 \end{array}$$

Simple cycles: 2, 4, (1, 4), (1, 1, 4), (2, 4)

Greedy " : 2, (1, 1, 4)

M A L O C

for architecture test      out  
Data source → Add → Simulation source  
create file      name - db

Simulation Source Sim1      Port I/O declare.  
anchi  
begin \$entity (Component)  
end component

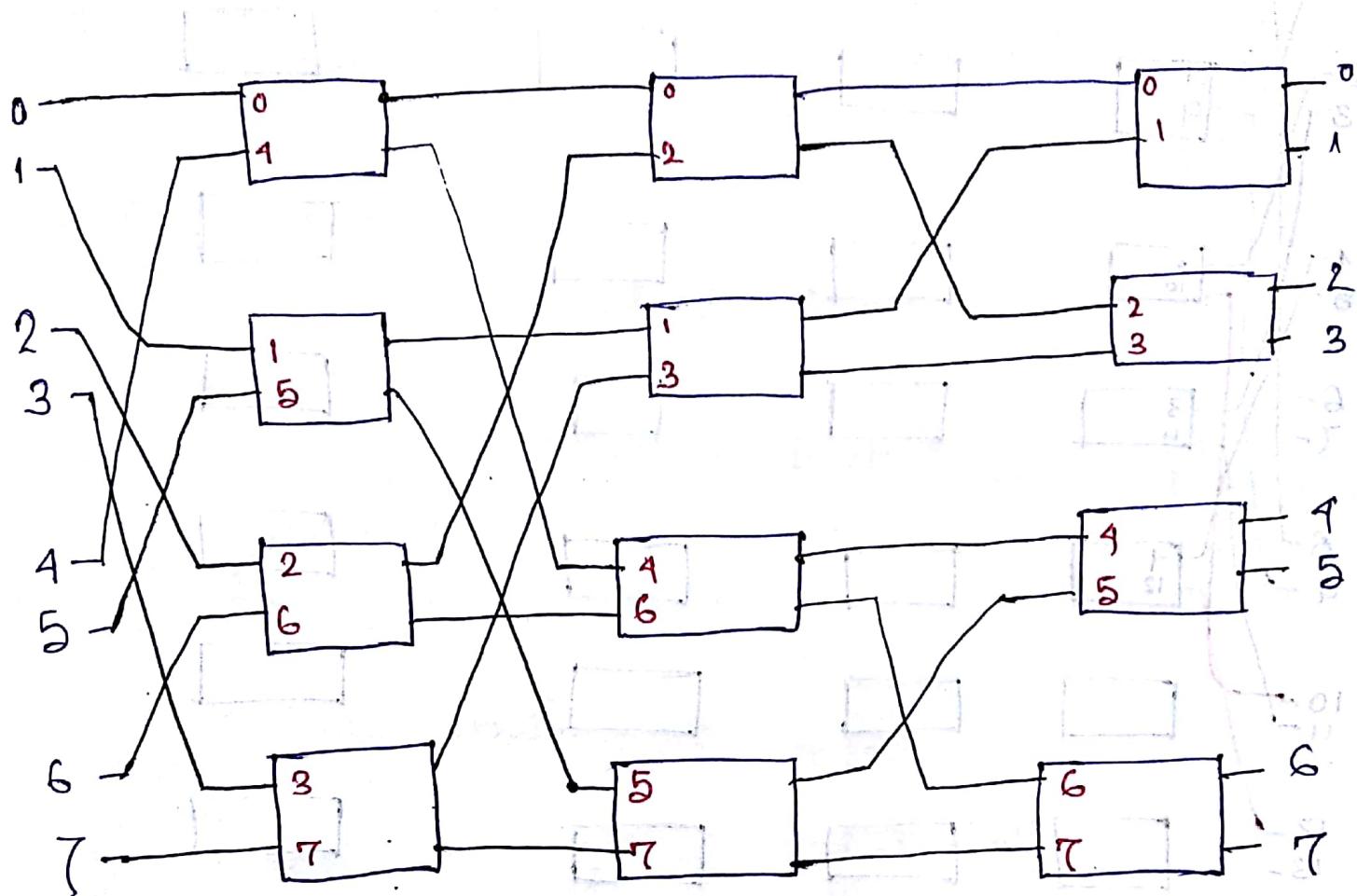
Signal S1 : STD-Logic = '0';  
begin  
on1 : on\_dataflow Portmap (A => S1, );  
shim\_proc: process  
begin  
wait for 150 ns  
S1 <= '0';  
end

## Banyan Network

8 i/p n/o

$\log_2 n$

when  $n=8$  then 3 stages.



1) MSB bit, Second bit, LSB

0 0 1

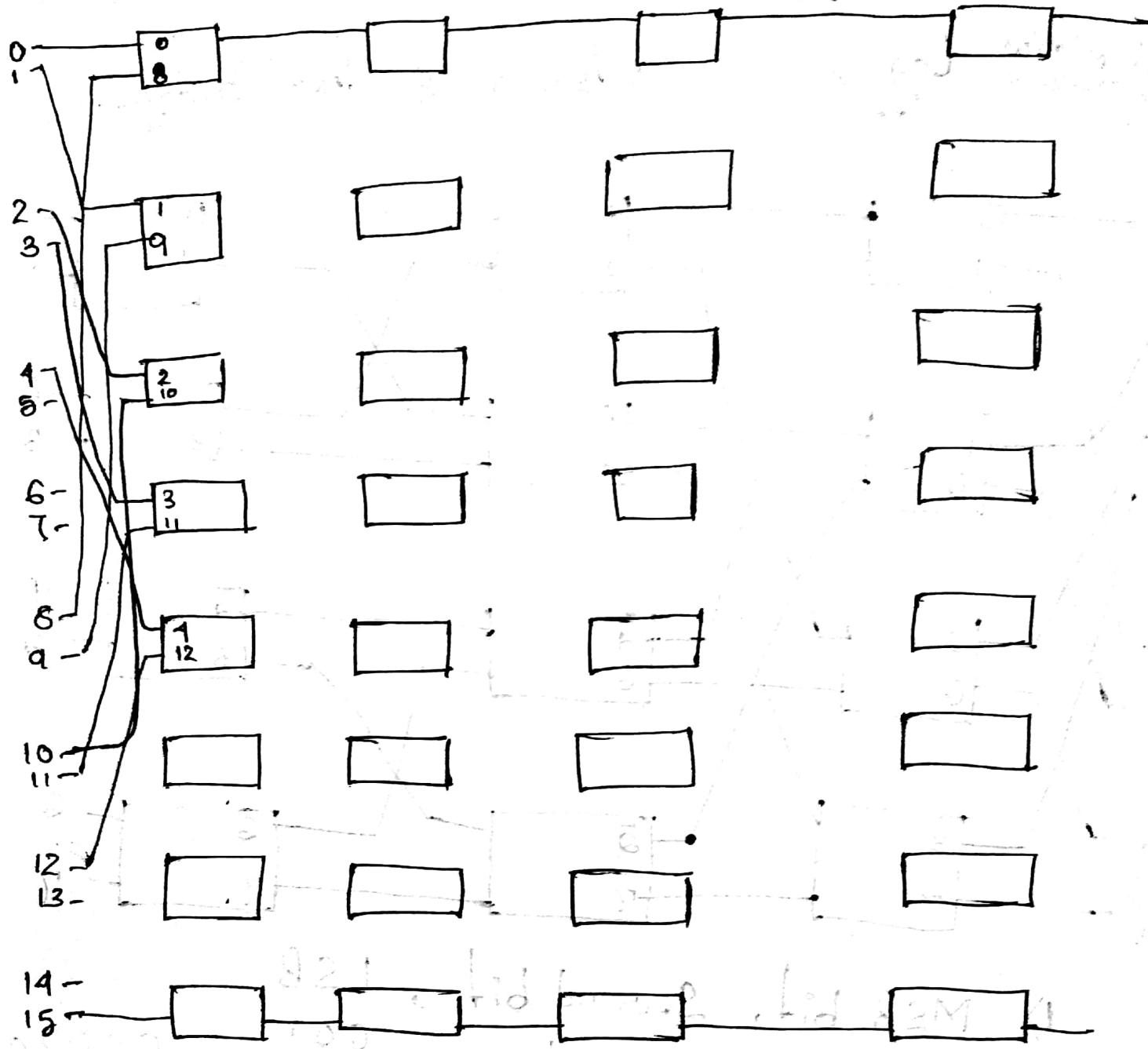
0 0 1

0 0 1

0 0 0 } 0 1  
1 0 0 } 1 1

16 input banyan n/o

$\log_2 16$ . as  $n=16$  so 4 stages



0 -	0000	→	1000
1 -	0001	→	1001
2 -	0010	→	1010
3 -	0011	→	1011
4 -	0100	→	1100
5 -	0101	→	1101
6 -	0110	→	1110
7 -	0111		
8 -	1000		
9 -	1001		
10 -	1010		
11 -	1011		
12 -	1100		
13 -	1101		
14 -	1110		
15 -	1111		

16 8 92±  
1 000

## Super Scaler Architecture

A superscalar processor is a CPU that implements a form of parallelism called inst<sup>n</sup>-level parallelism, within a single processor.

An normal Pipeline

$$T(1,1) = K + N - 1$$

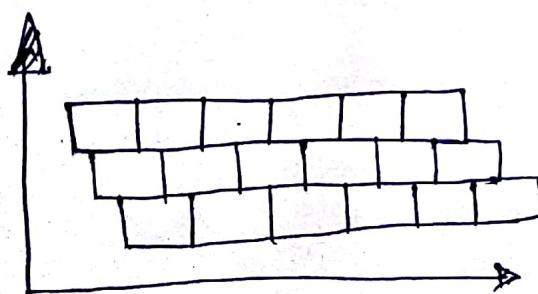
Hence

$$T(m,1) = K + \frac{N-m}{m}$$

So the speed up hence

$$\begin{aligned} S(m,1) &= \frac{T(1,1)}{T(m,1)} \\ &= \frac{m(N+K-1)}{N+m(K-1)} \end{aligned}$$

## Super Pipeline Architecture



$$\text{An general } \Rightarrow K+N-1$$

Hence,  $T(1,n) = \boxed{\text{X}} \times \frac{1}{n}(N-1)$

$\uparrow_{\substack{\text{sub} \\ \text{clk pulse}}}$

$$\text{Speed up } S(1,n) = \frac{T(1,1)}{T(1,n)} = \frac{n(K+N-1)}{nk+N-1}$$

## Super Scalars Super pipeline Architecture

$$T(m, n) = K + \frac{N-m}{mn}$$

$$\text{Speed up} = \frac{T(1, 1)}{T(m, n)}$$
$$= m \frac{mn(K + N - 1)}{mnK + N - m}$$

