Chap. 5 Multistage Cube/Shuffle-Exchange Networks

- based on Cube interconnection functions
- alternatively, based on Shuffle-Exchange functions
- can use in:
  - SIMD
  - multiple-SIMD
  - MIMD
  - partitionable SIMD/MIMD

Multistage Cube Network

- N inputs/outputs
- Log_2N stages
- N/2 switches/stage
- Distributed routing tag control
- Partitionable

OUTLINE

1. multistage cube structure
2. paths through the multistage cube
3. routing tag control for the multistage cube
4. partitioning the multistage cube
5. relationships among multistage cube-type networks

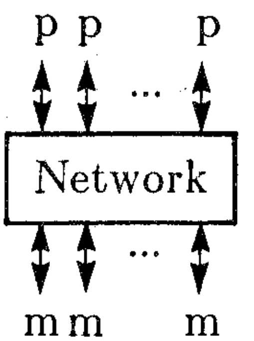
Multistage Cube Network Topology

Aliases
- omega network
- flip network
- indirect binary n-cube network
- SW-banyan network (s=f=2)
- butterfly network
- multistage shuffle-exchange network
- baseline network
- delta network
- generalized cube network

Used/Proposed for
- STARAN
- PASM
- Ultracomputer
- IBM RP3
- BBN Butterfly
- Dataflow Machines
- Cedar

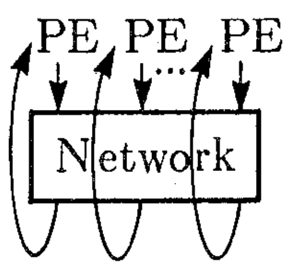
Generalized Cube Network Structure

- conceptually based on two-input/two-output device - *interchange box*
- m = $\log_2$ N stages of boxed for N×N network
- N/2 boxes per stage
- each box individually controlled
- network could be bidirectional



Processor-to-Memory configuration

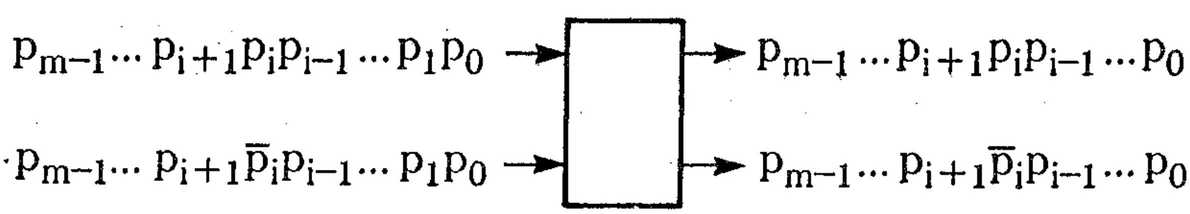
- assume unidirectional and same device at network input j and output j



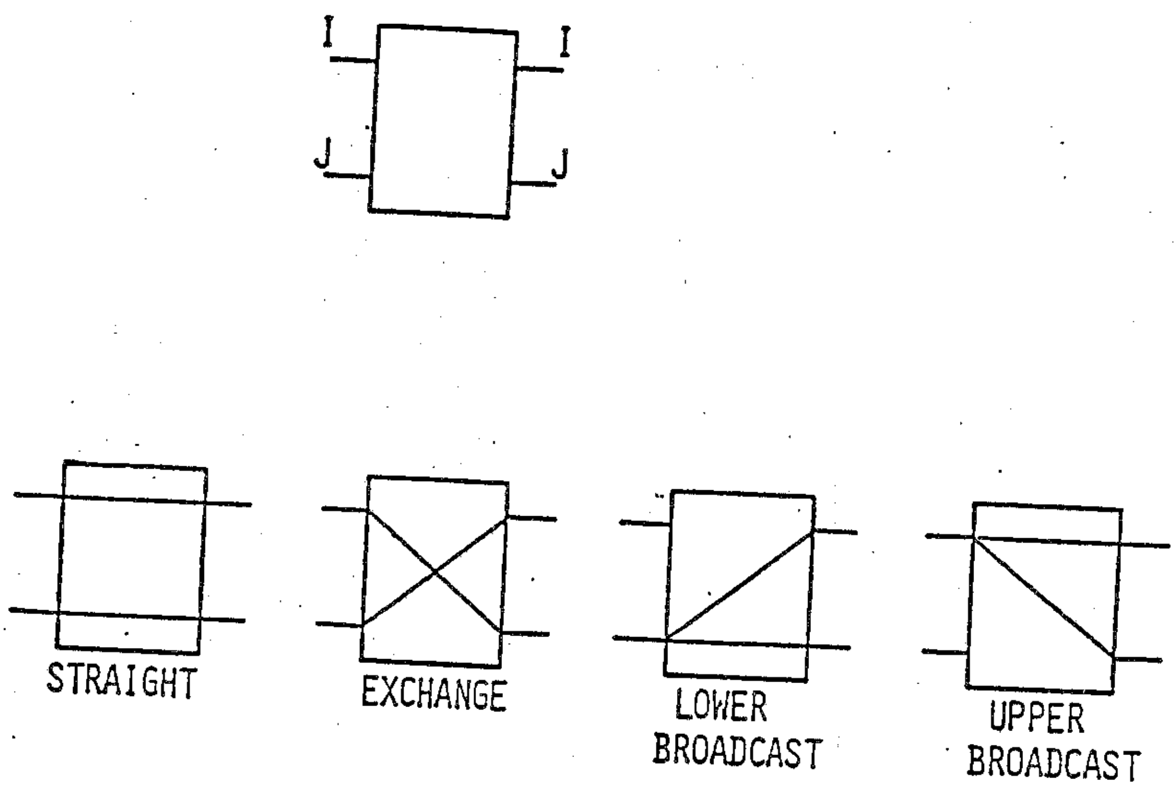
PE: processing element proc./mem. pair

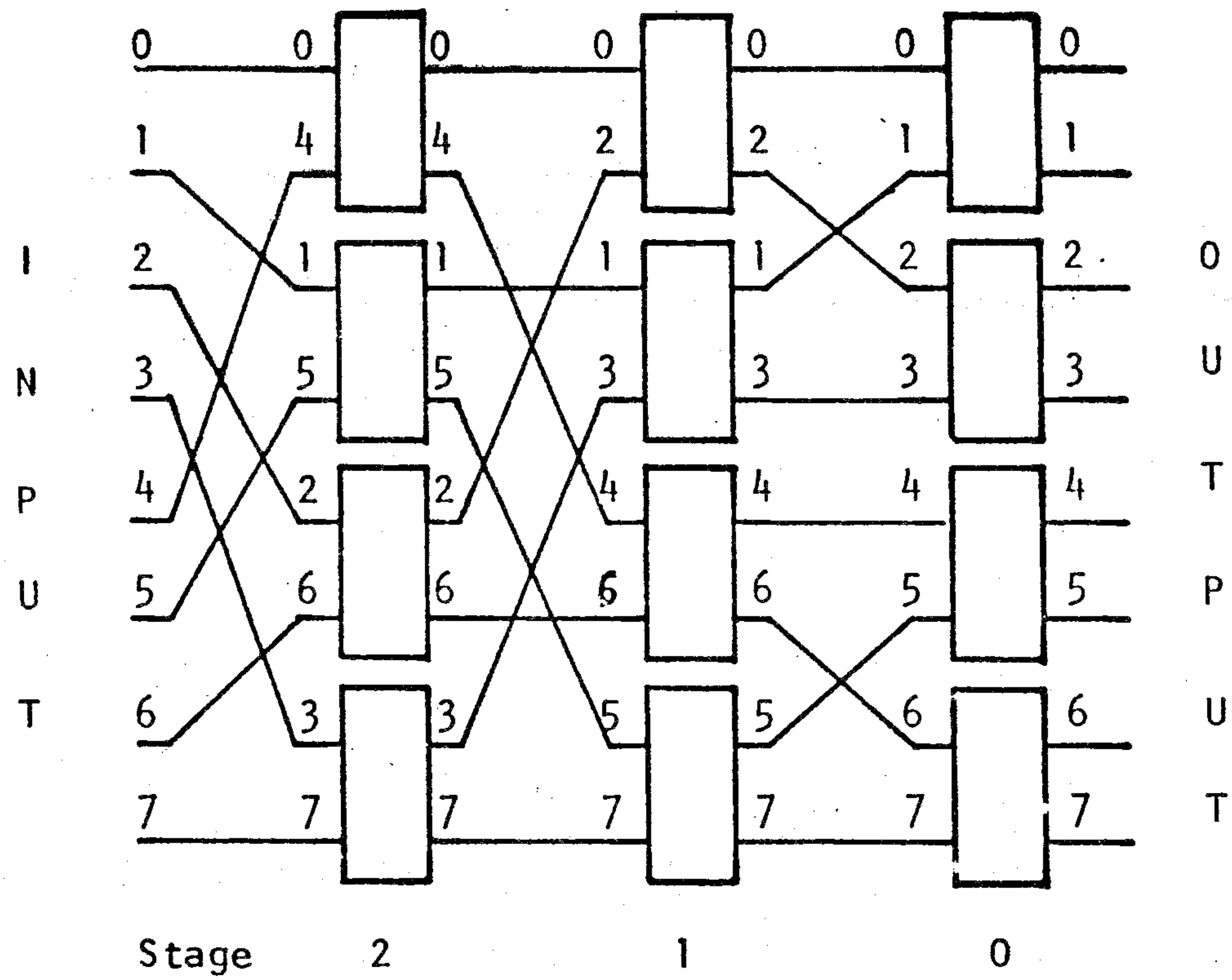
PE-to-PE configuration

- connection pattern between stages: at stage i link labels differ in i^{th} bit

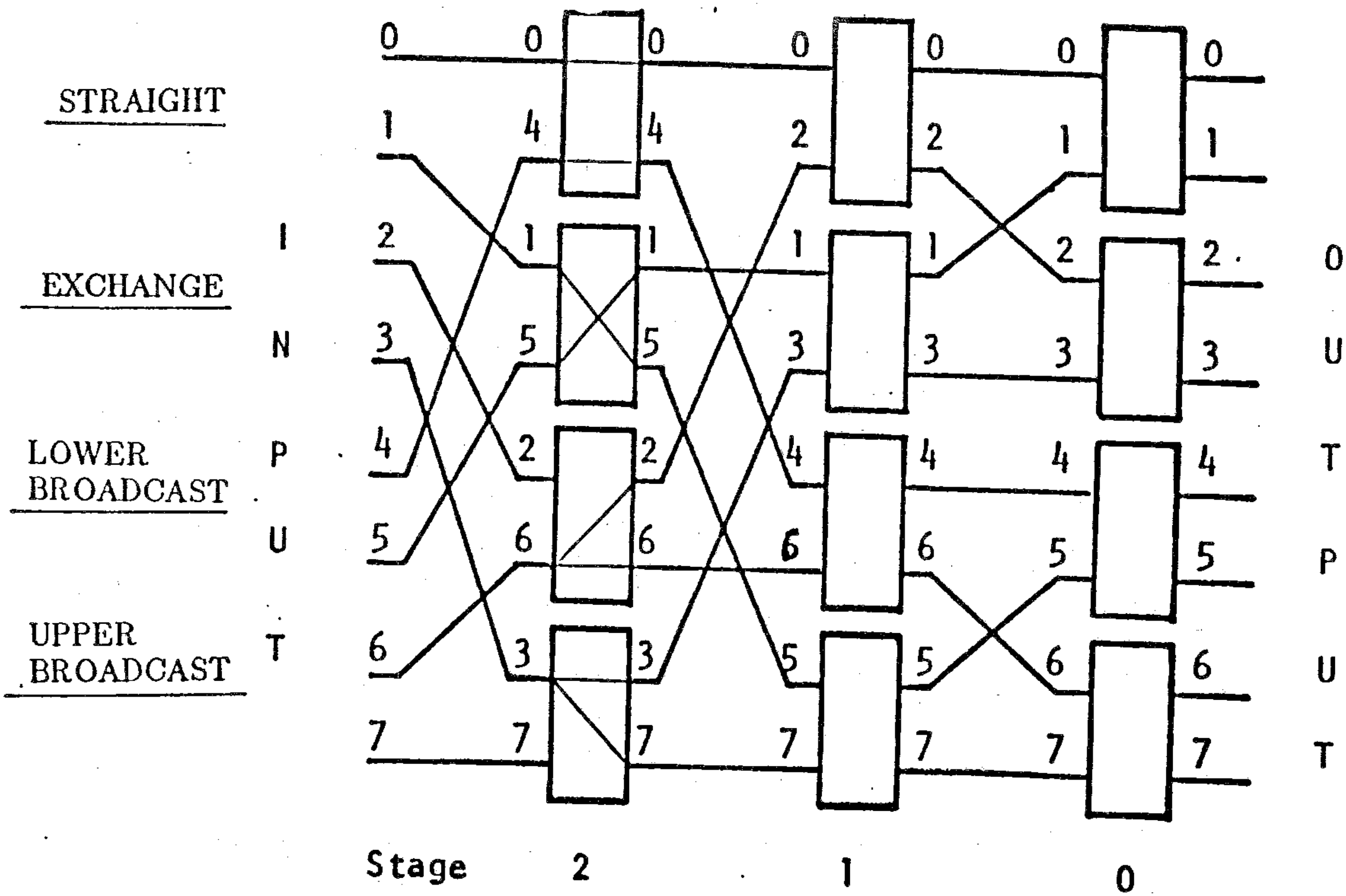


INTERCHANGE BOX
TWO INPUT, TWO OUTPUT DEVICE

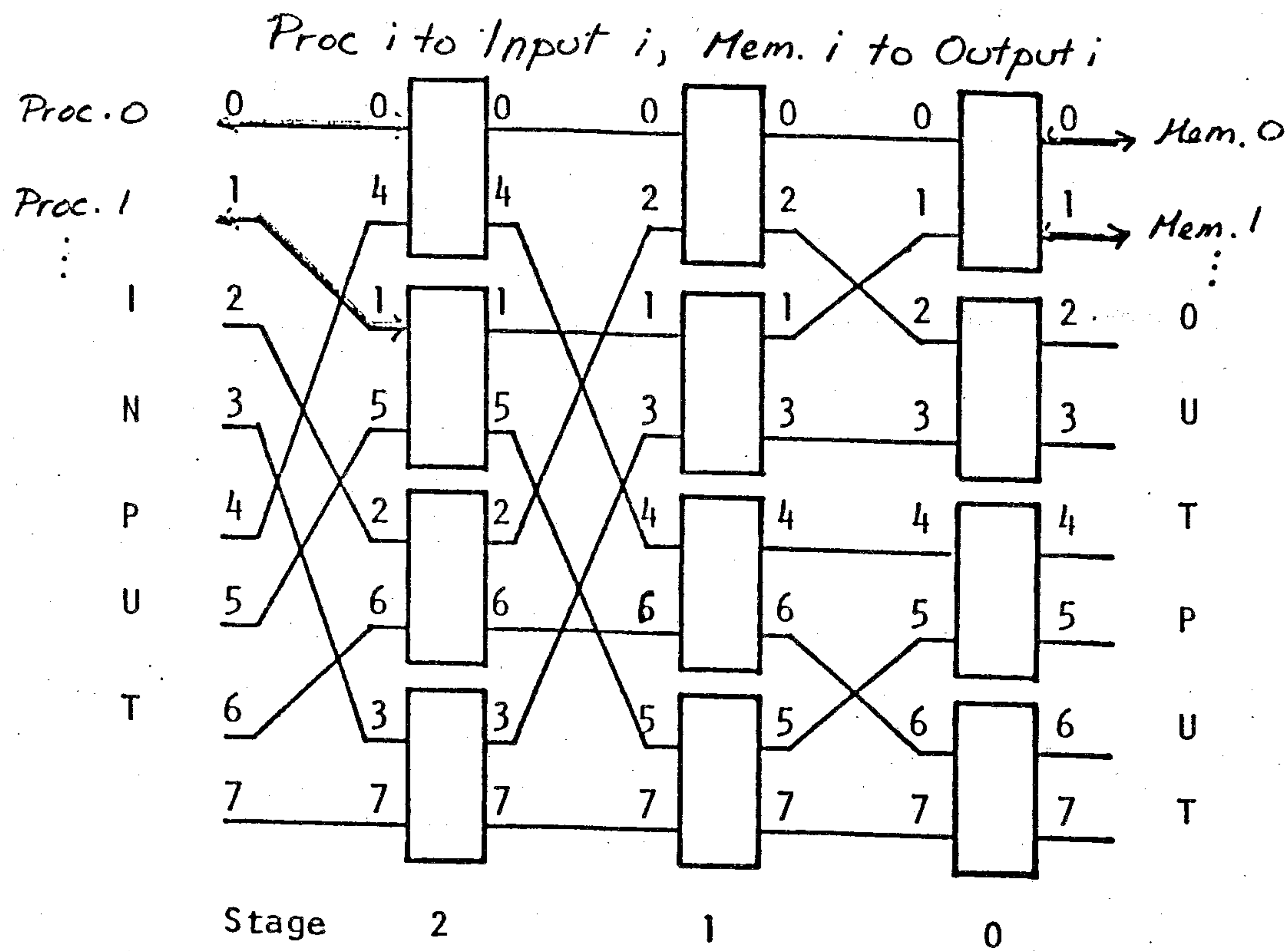




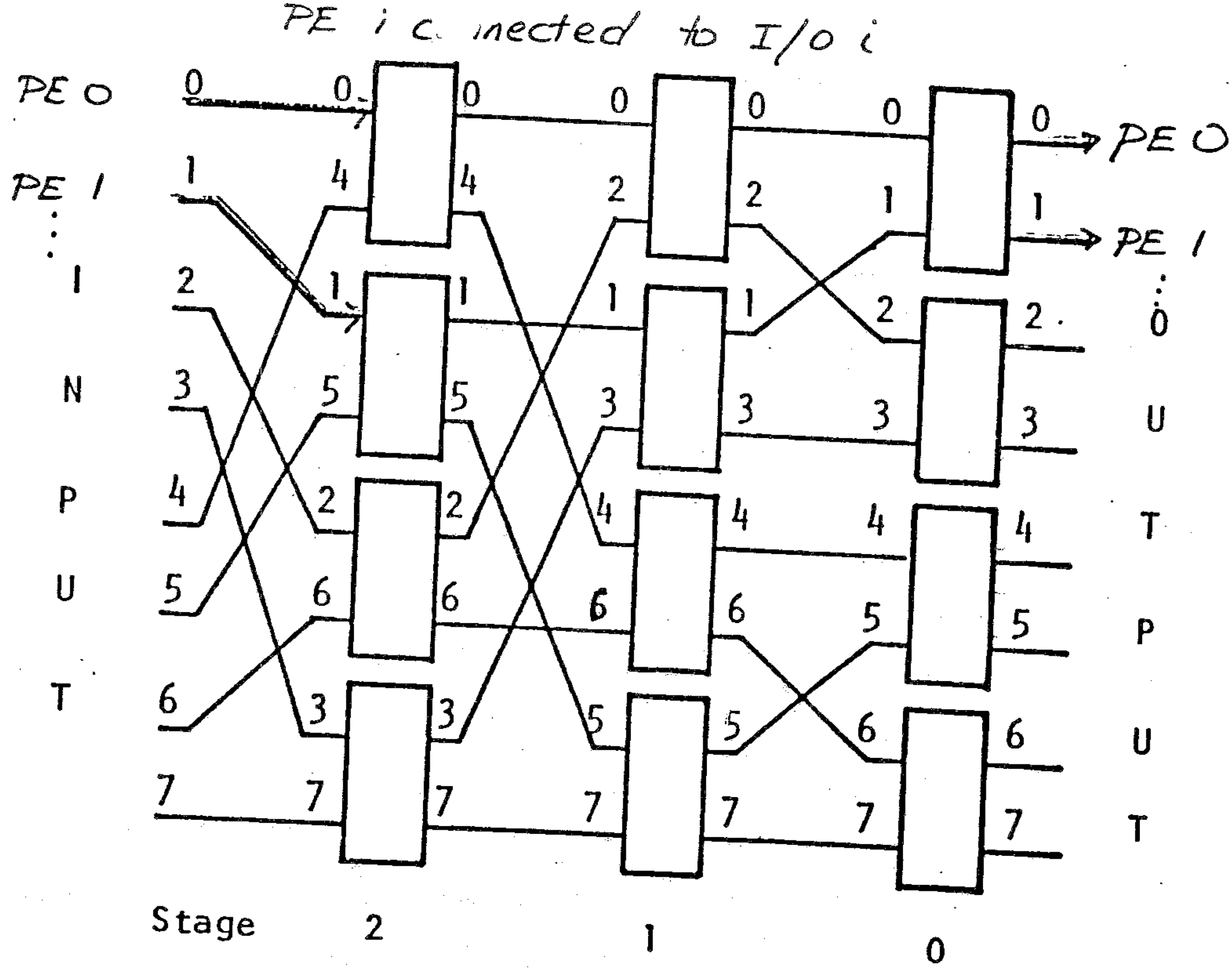
GENERALIZED CUBE TOPOLOGY FOR N = 8



GENERALIZED CUBE TOPOLOGY FOR N = 8



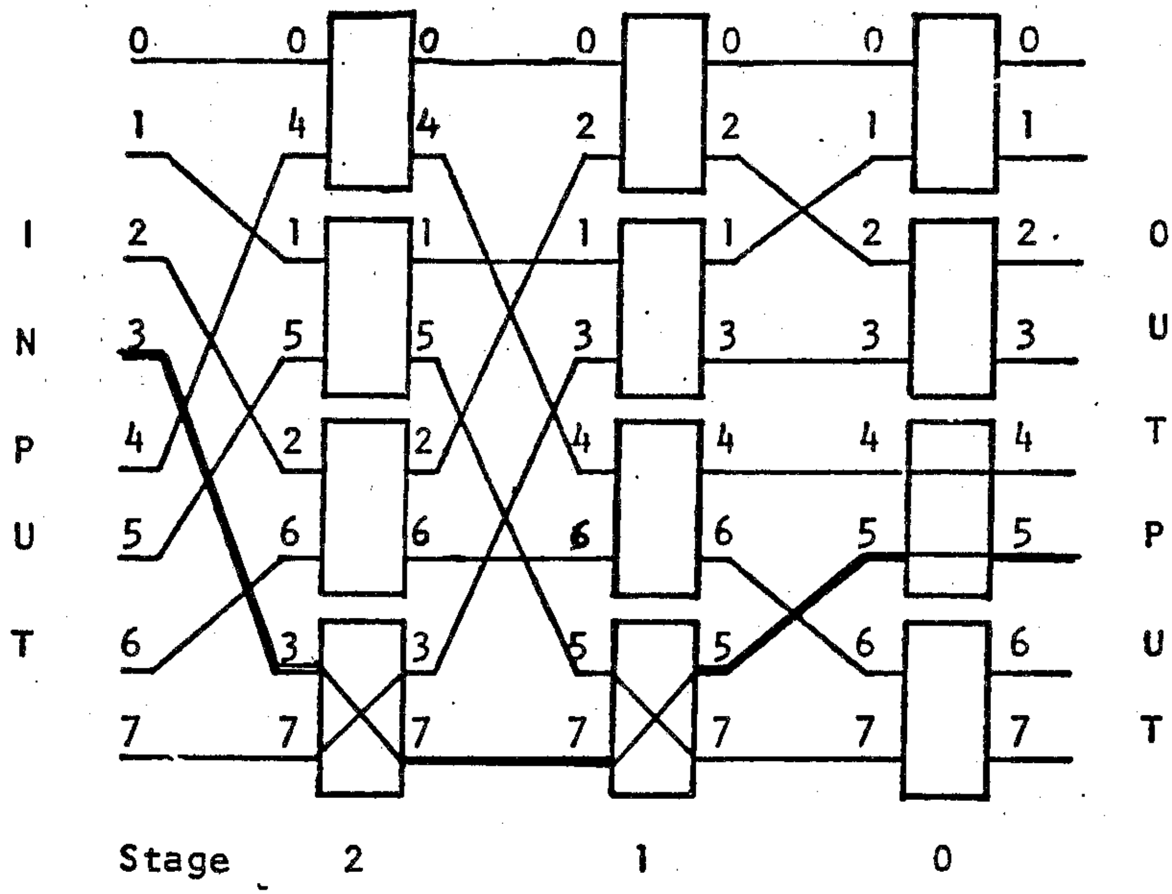
GENERALIZED CUBE TOPOLOGY FOR N = 8



GENERALIZED CUBE TOPOLOGY FOR N = 8

Example 1-to-1 Connection

$3 \rightarrow 5$



GENERALIZED CUBE TOPOLOGY FOR N = 8.

Source $S = s_2 s_1 s_0$ Destination $D = d_2 d_1 d_0$

Example 1-to-1 Connection

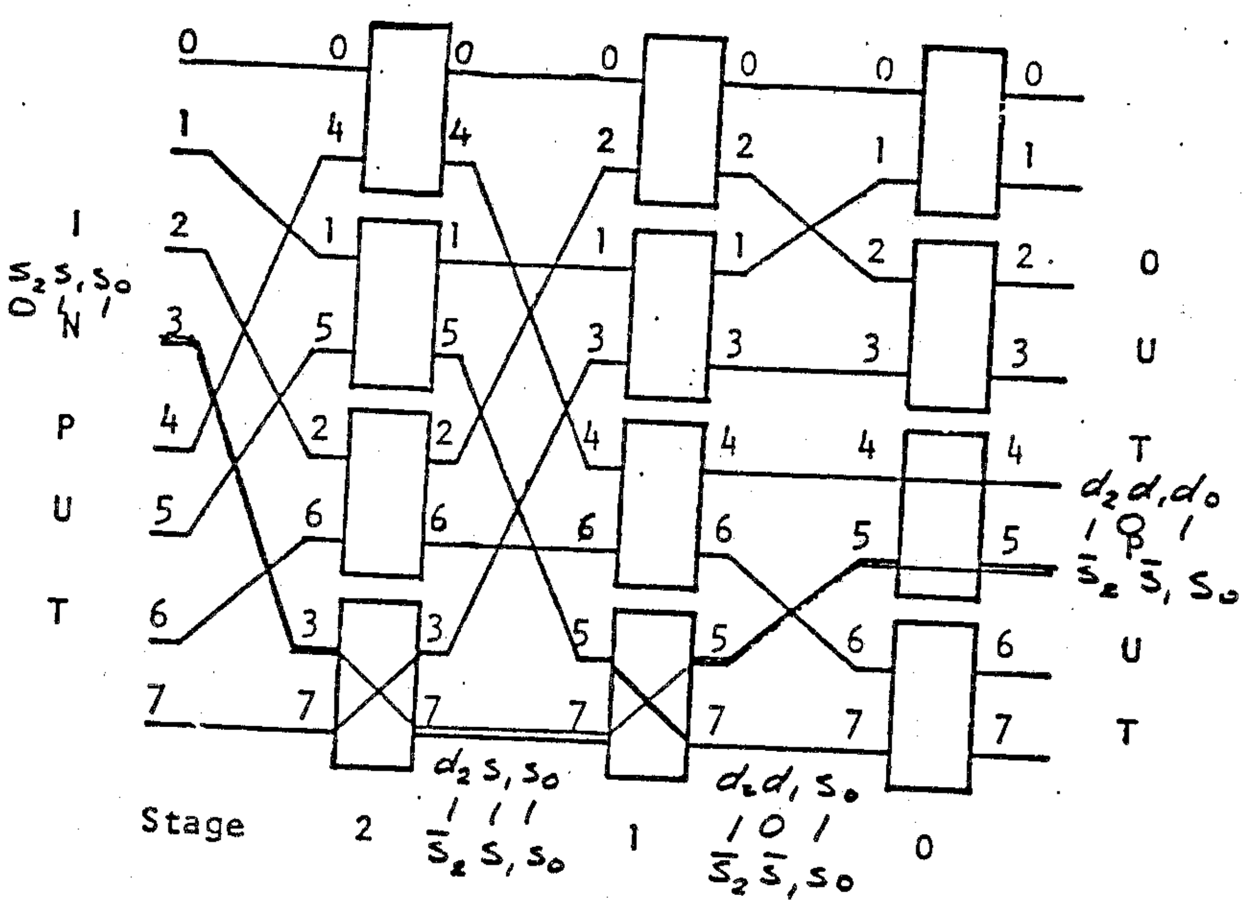
Stage i determines i^{th} bit of destination

$d_i = s_i$ 3→5

$d_i = \bar{s}_i$

$s_2\, s_1\, s_0 \rightarrow d_2\, d_1\, d_0$

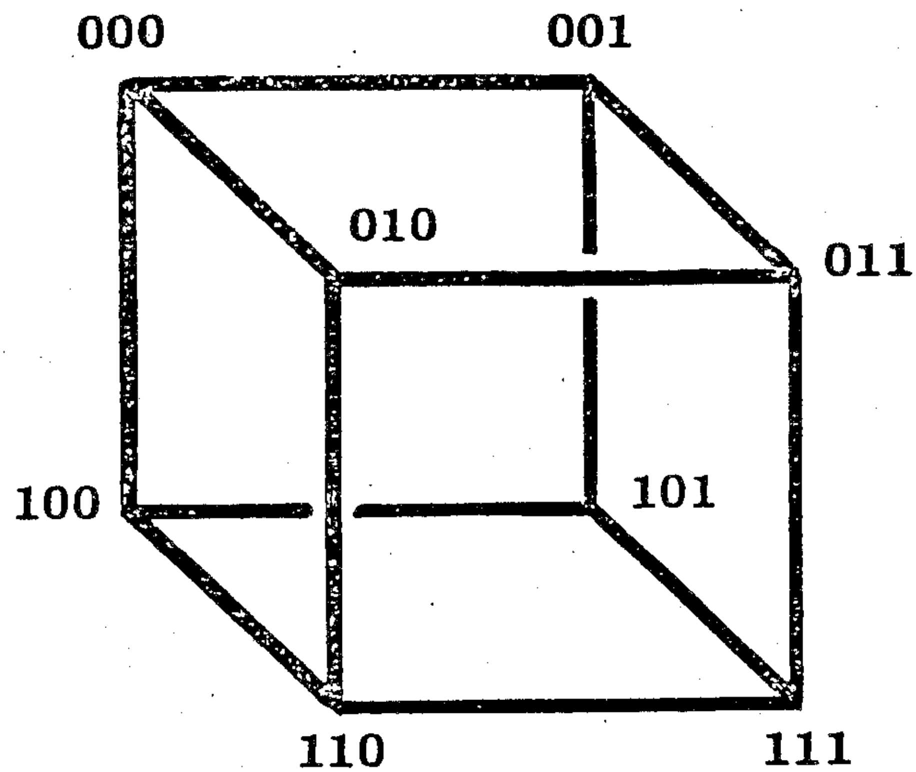
$0\ 1\ 1 \rightarrow 1\ 0\ 1$



GENERALIZED CUBE TOPOLOGY FOR N = 8.

Only one path from a given source to a given destination

Reason Why Called "Cube"



Three Dimensional Cube Structure, with Vertices Labeled from 0 to 7 Binary.

Stage 0

Stage 1

Stage 2

Packet Switching vs. Circuit Switching

Packet — fixed size packet moves from one stage to next

- occupies only 1 stage at a time
- storage for packets at each box

Circuit — establish complete path through network and hold for whole transmission

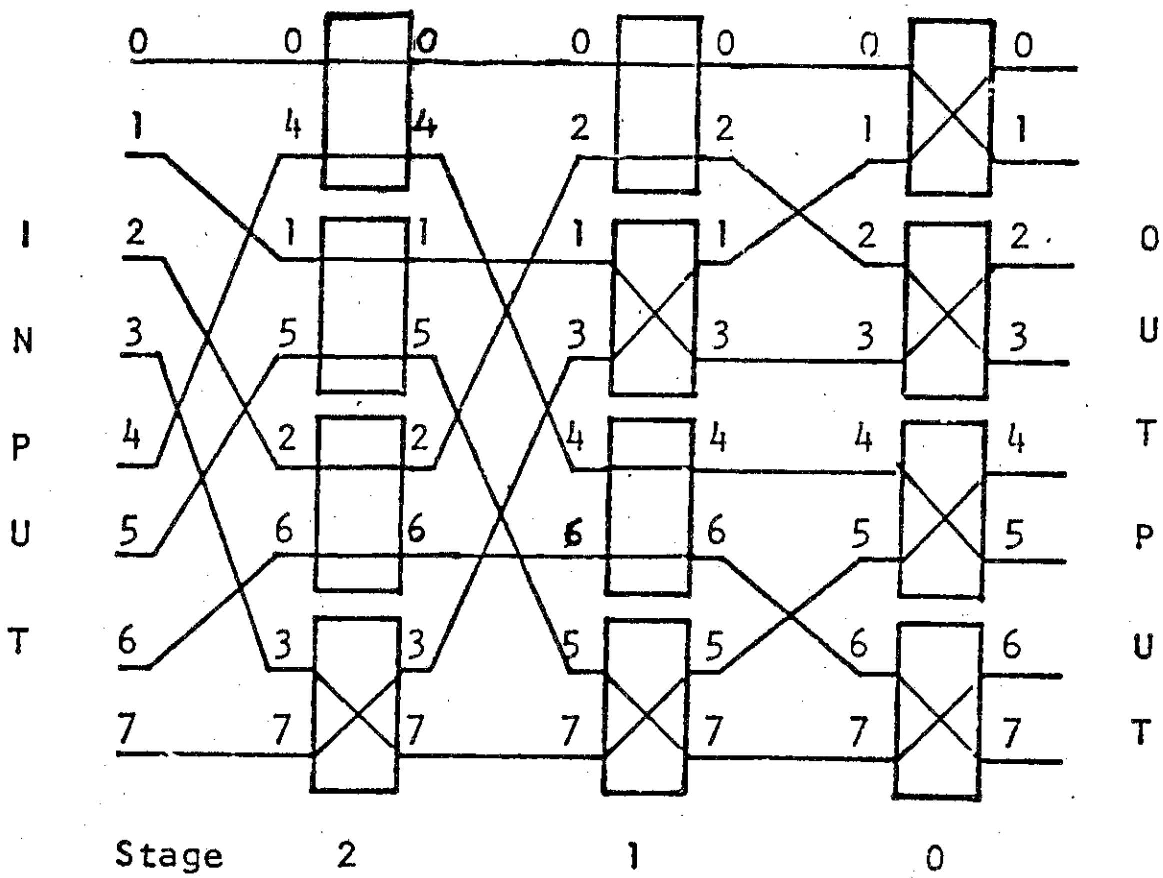
- occupies $\log_2 N$ boxes
- no storage at boxes

Tradeoffs — currently under study, factors involved include:

- implementation details
- protocols
- average message size
- fixed or variable size messages
- network load

Example Permutation

input i to output i+1 mod 8

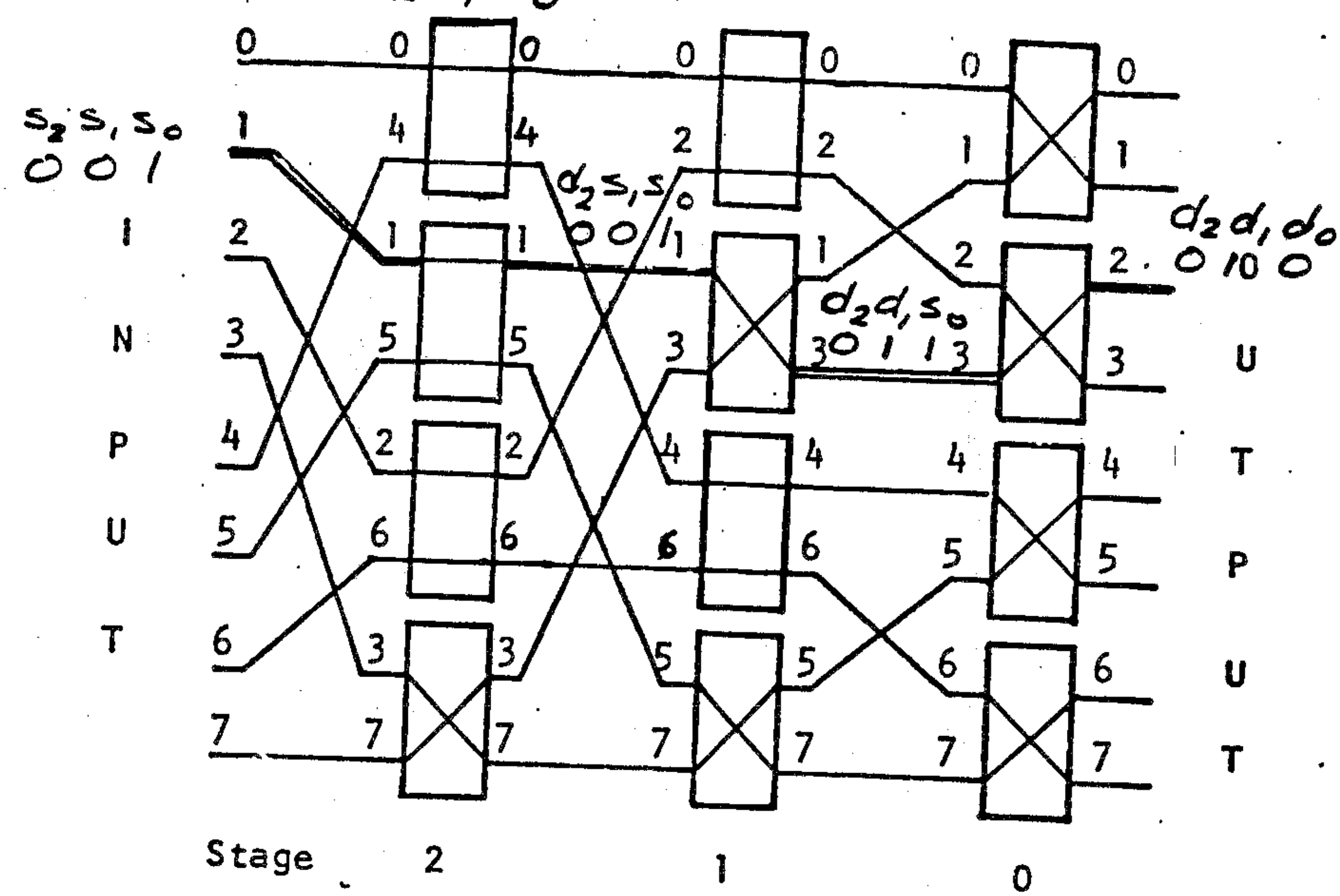


GENERALIZED CUBE TOPOLOGY FOR N = 8.

Example Permutation

input i to output i+1 mod 8

Ex. $S=1=s_2s_1s_0=001$
$D=2=d_2d_1d_0=010$

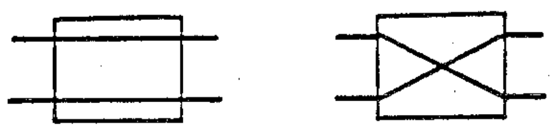


GENERALIZED CUBE TOPOLOGY FOR N = 8.

Number of Permutations:

$\log_2 N * (N/2)$ Boxes

Each Box



$$2^{\log_2 N * (N/2)} << N!$$

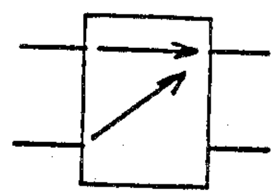
N	CUBE	N!
4	16	24
8	4K	40K

"Useful" Permutations

(SIMD)

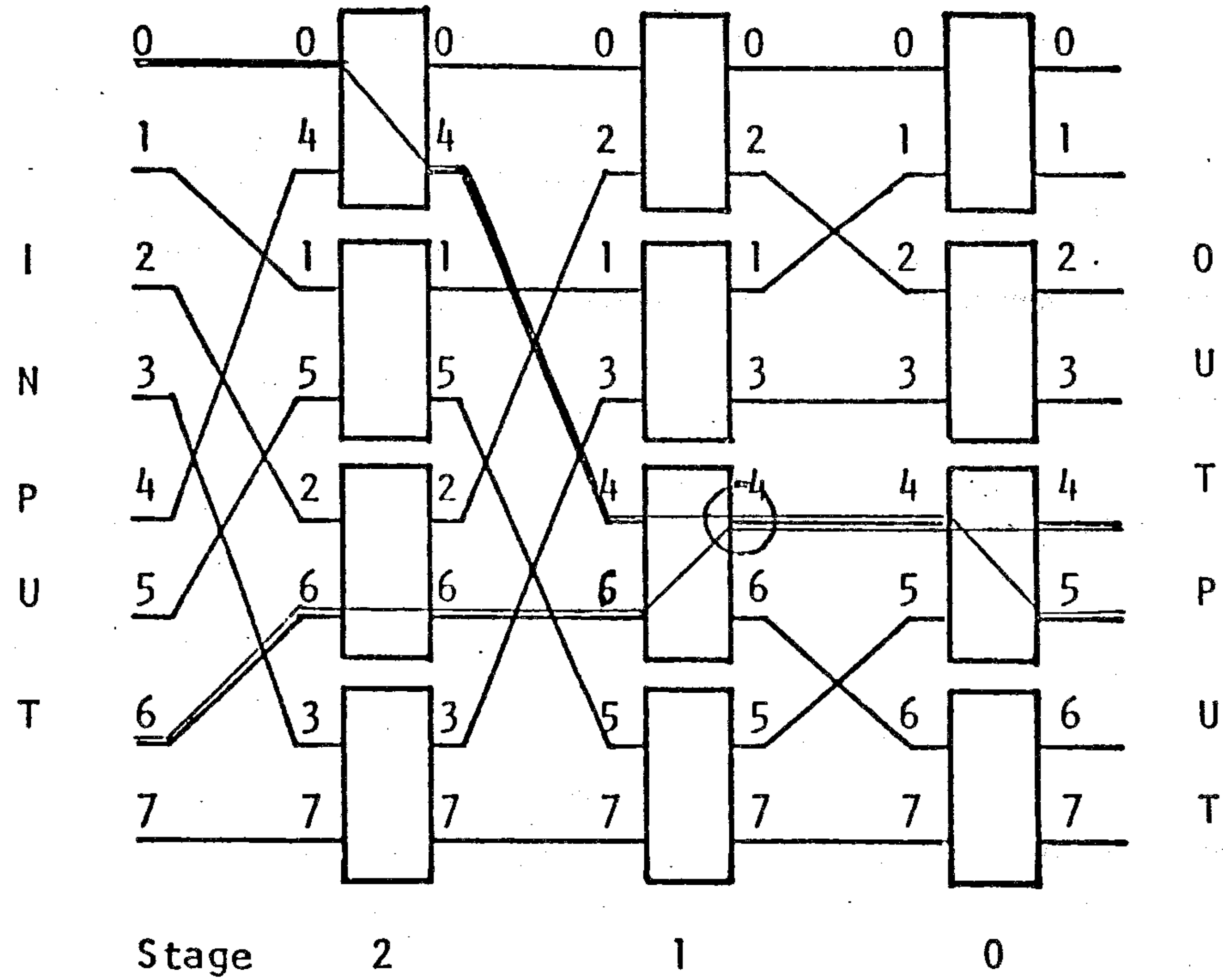
Network Conflicts

MIMD mode (not "passable" permutation)



two inputs desire same output

one must wait

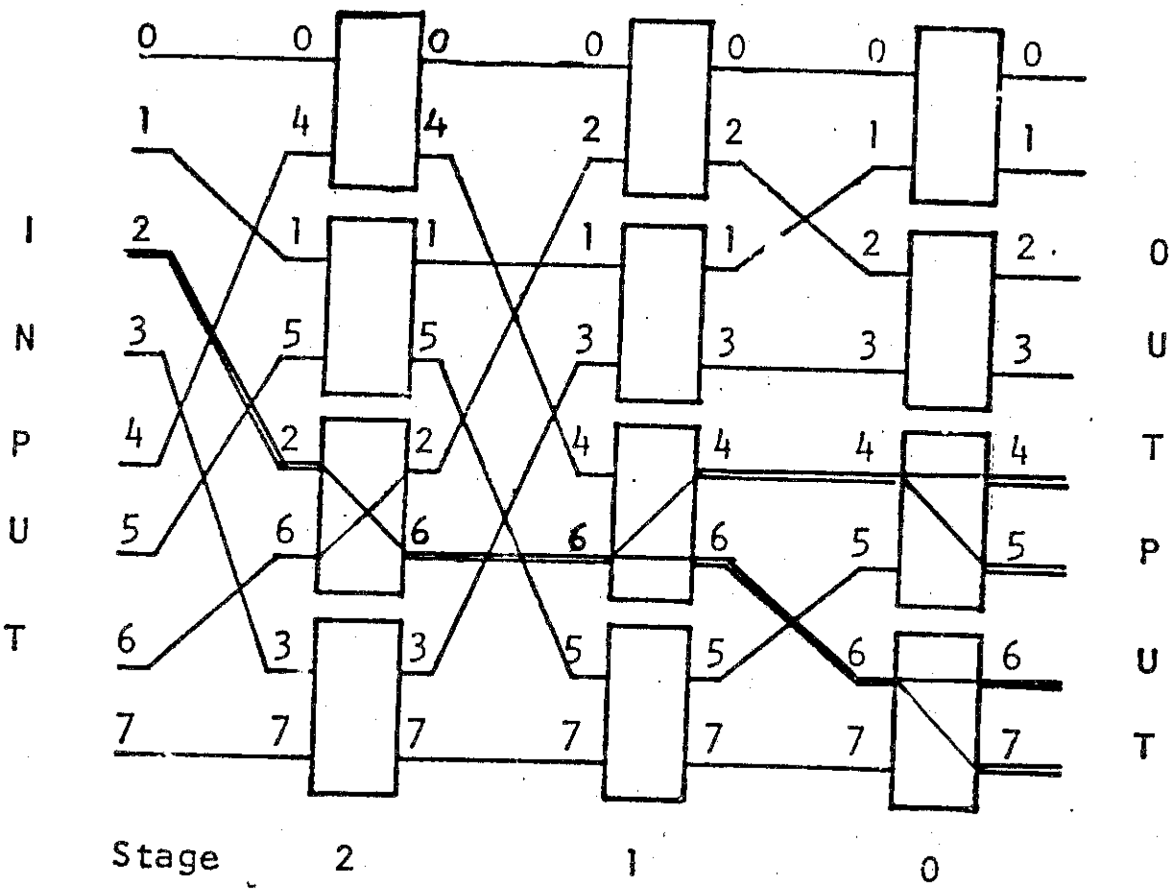


EXAMPLE OF CONFLICT $0 \to 5$, $6 \to 4$

GENERALIZED CUBE TOPOLOGY FOR N = 8

Example Broadcast

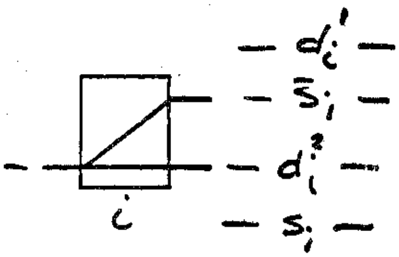
$$2 \rightarrow \{4, 5, 6, 7\}$$

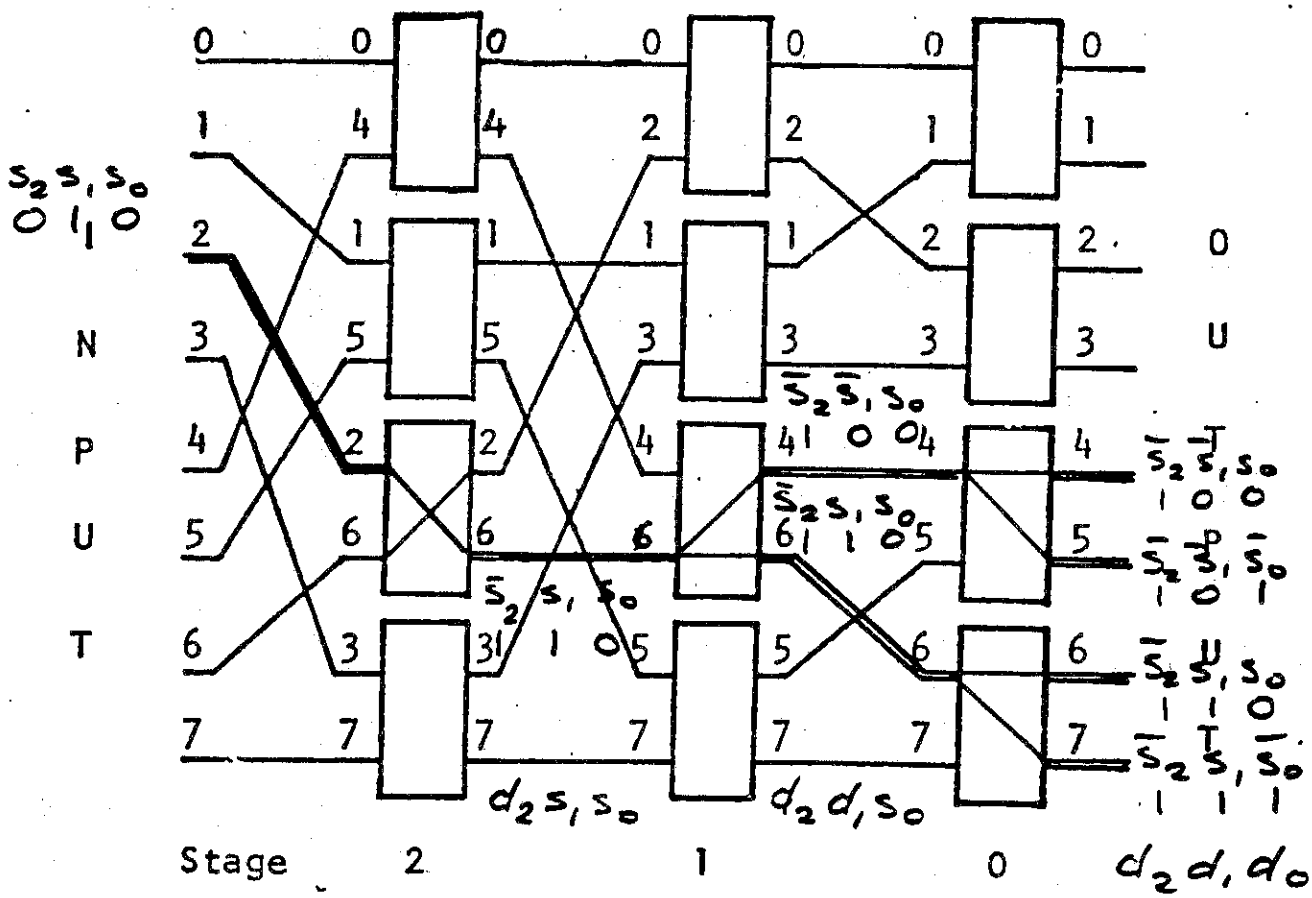


GENERALIZED CUBE TOPOLOGY FOR N = 8.

Example Broadcast

$2 \rightarrow \{4, 5, 6, 7\}$





GENERALIZED CUBE TOPOLOGY FOR N = 8.

Network Control - Routing Tags

- control distributed
- each network input device determines own tag
- tag is header for message
- XOR scheme for 1-to-1

  — $m = \log_2 N$ bits per tag

  — Source $S = s_{m-1} \cdots s_1 s_0$

  — Destination $D = d_{m-1} \cdots d_1 d_0$

  — Tag $T = t_{m-1} \cdots t_1 t_0 = S \oplus D$

  — stage i box examines t_i
  (each box set independently)
  $t_i = 0 \rightarrow$ set straight
  $t_i = 1 \rightarrow$ set exchange

  — use for 1-to-1 or permutations

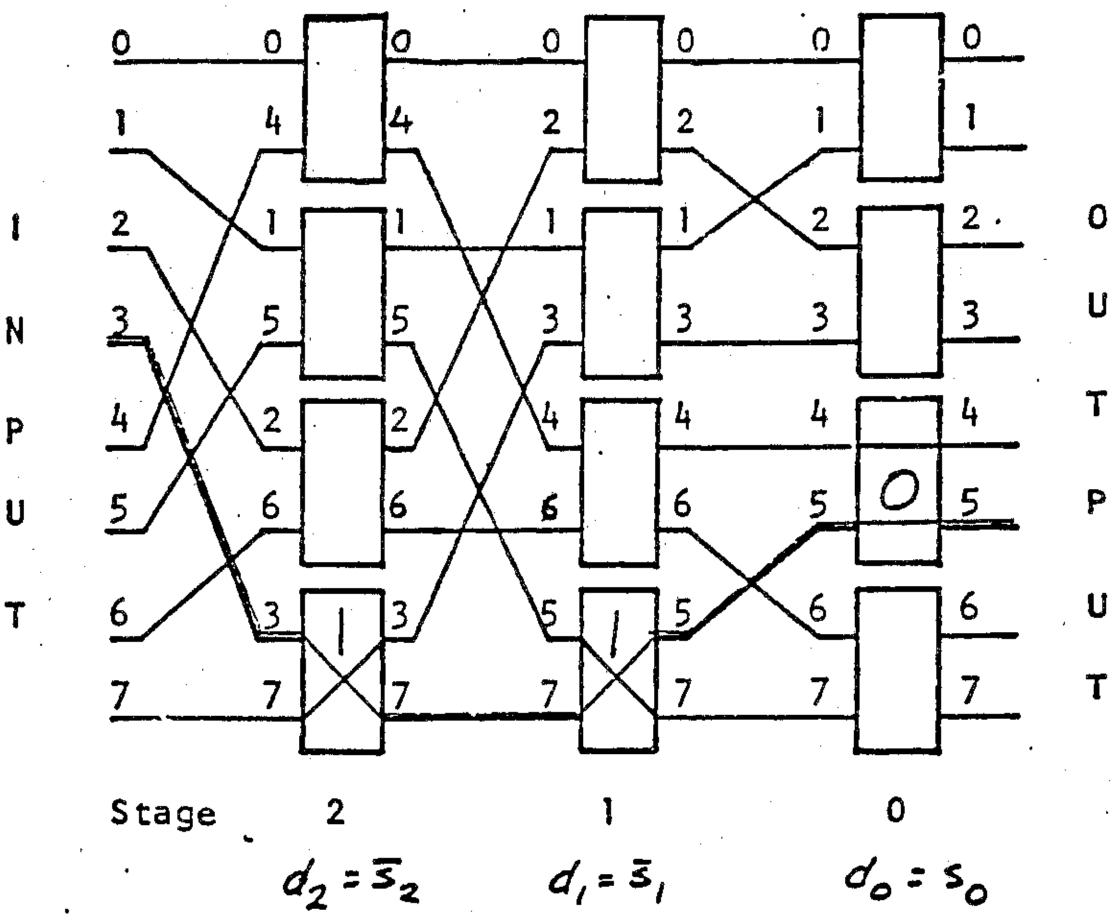
  — add m-bit broadcast mask for broadcasts

  — tag can be used for return message and source info
  $T = S \oplus D = D \oplus S$
  $S = D \oplus T$

Routing Tag Example

$S = 3 = 011 \qquad\qquad D = 5 = 101$

$T = S \oplus D = 110$



GENERALIZED CUBE TOPOLOGY FOR N = 8.

Broadcast Routing Tag

One port to 2^j ports

can be at most j bits that differ between any pair of destination port addresses.

Port S $\rightarrow$ ports $\{D_1, D_2, \ldots, D_{2^j}\}$

Routing $= S \oplus D_I$
info.

Broadcast $= D_I \oplus D_k$ (must differ in j positions)

ex. S =	1100	$D_1 = 0000$	0
	12	$D_2 = 0001$	1
		$D_3 = 0010$	2
		$D_4 = 0011$	3

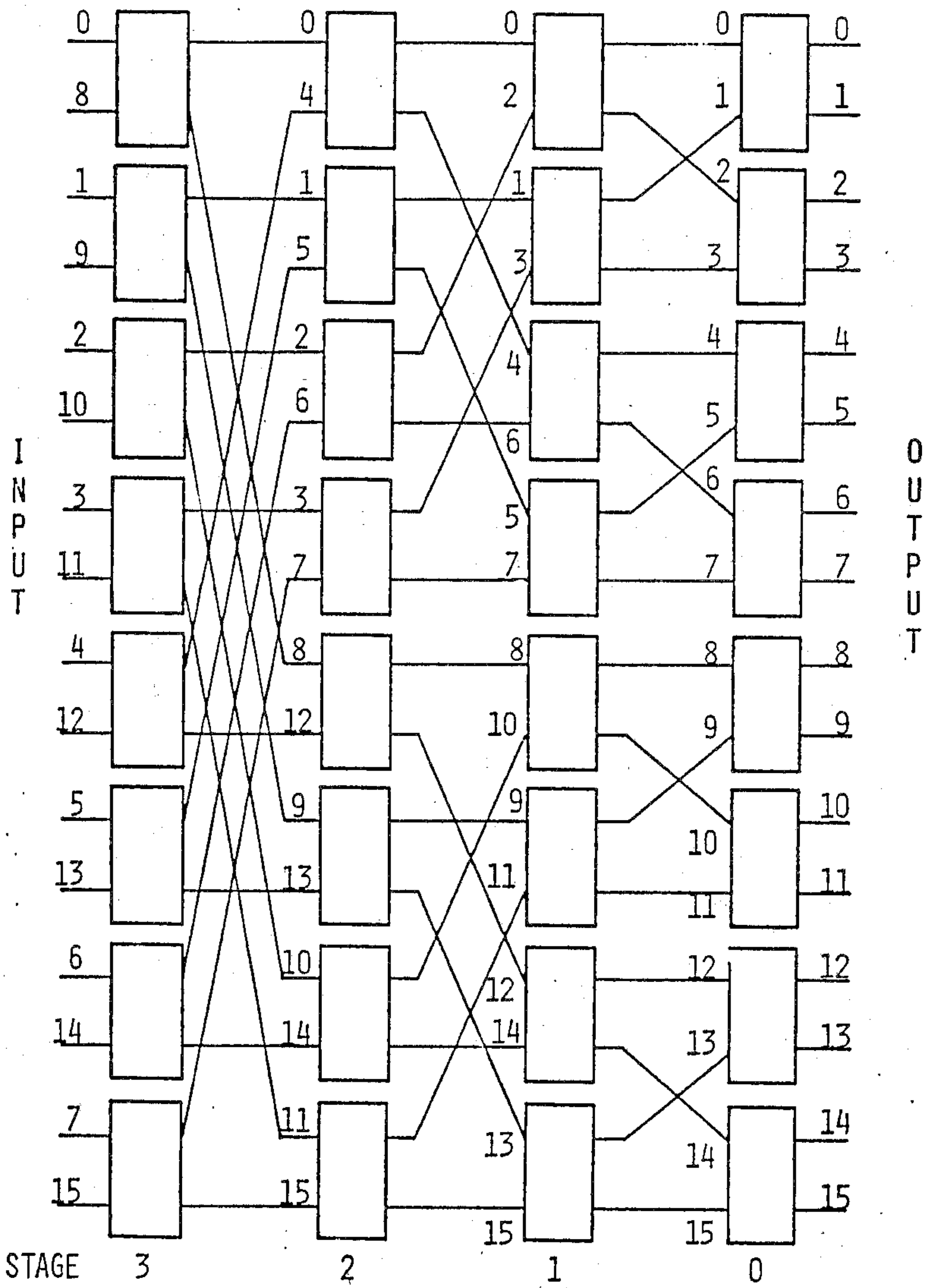
route $= 1101$ $(S \oplus D_2)$

broadcast $= 0011$ $(D_1 \oplus D_4)$

Stage i look at ith bit of route (r_i) and broadcast (b_i)

$b_i = 0$, use r_i: 1 exchange, 0 straight

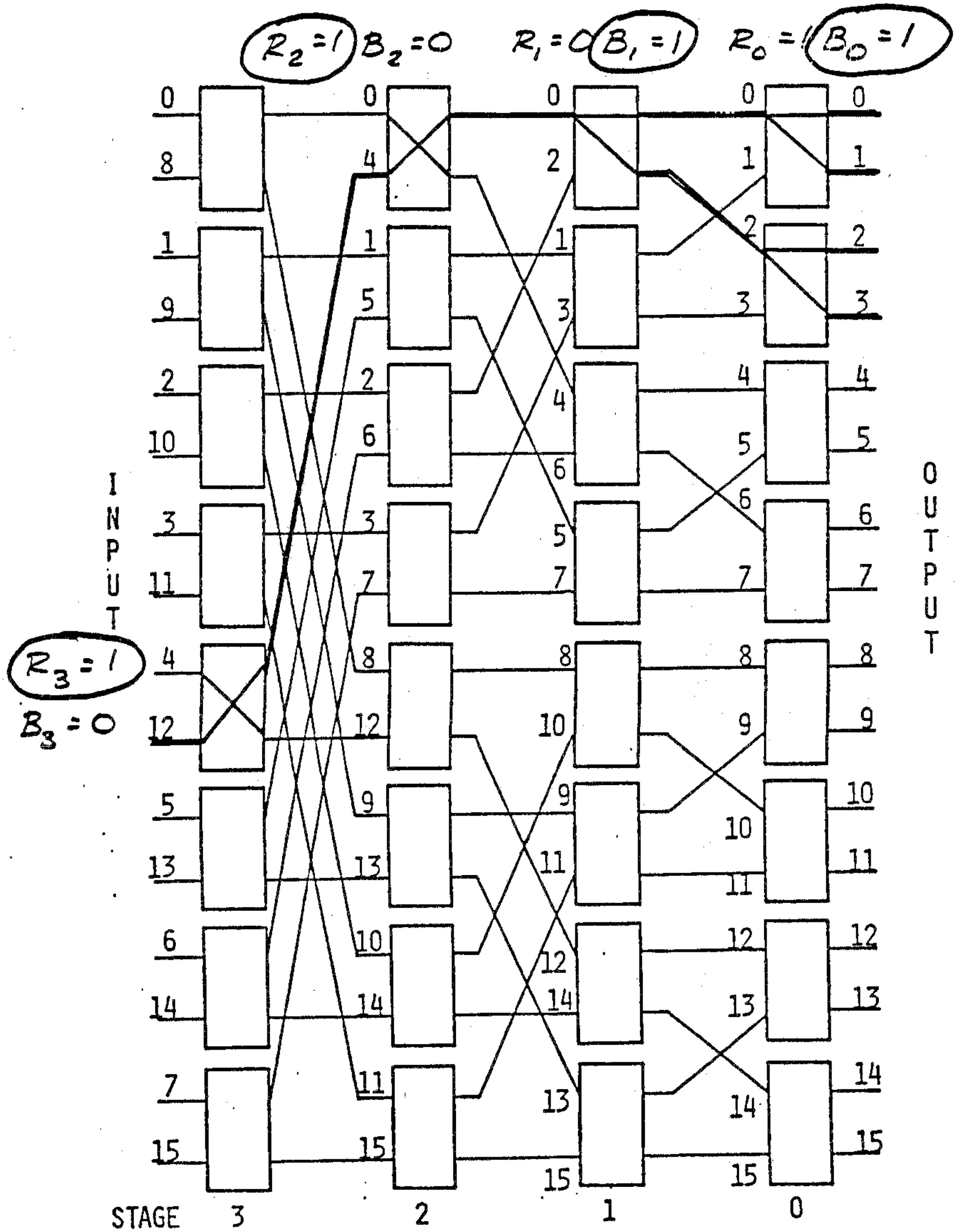
$b_i = 1$: broadcast (ignore r_i)



CUBE for N = 16

Example: 12 → {0, 1, 2, 3}

Route = 1101 Broadcast = 0011

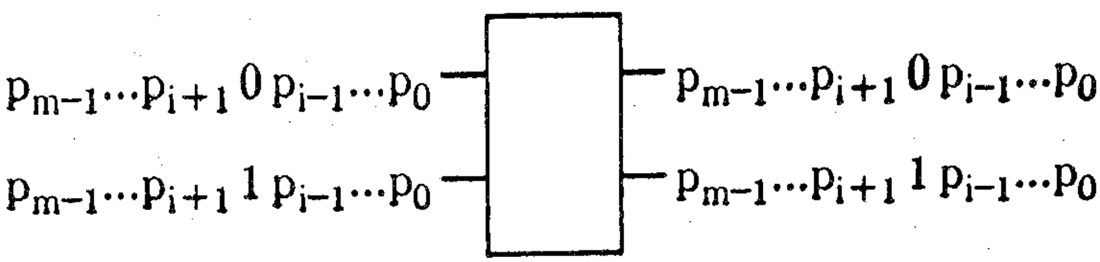


Network Control - Destination Tags

- $m = \log_2 N$ bits per tag
- Tag = Destination $D = d_{m-1} \ldots d_1 d_0$
- Stage i box examines d_i
  (each box set independently)

$$d_i = 0 \rightarrow \text{upper box output}$$

$$d_i = 1 \rightarrow \text{lower box output}$$

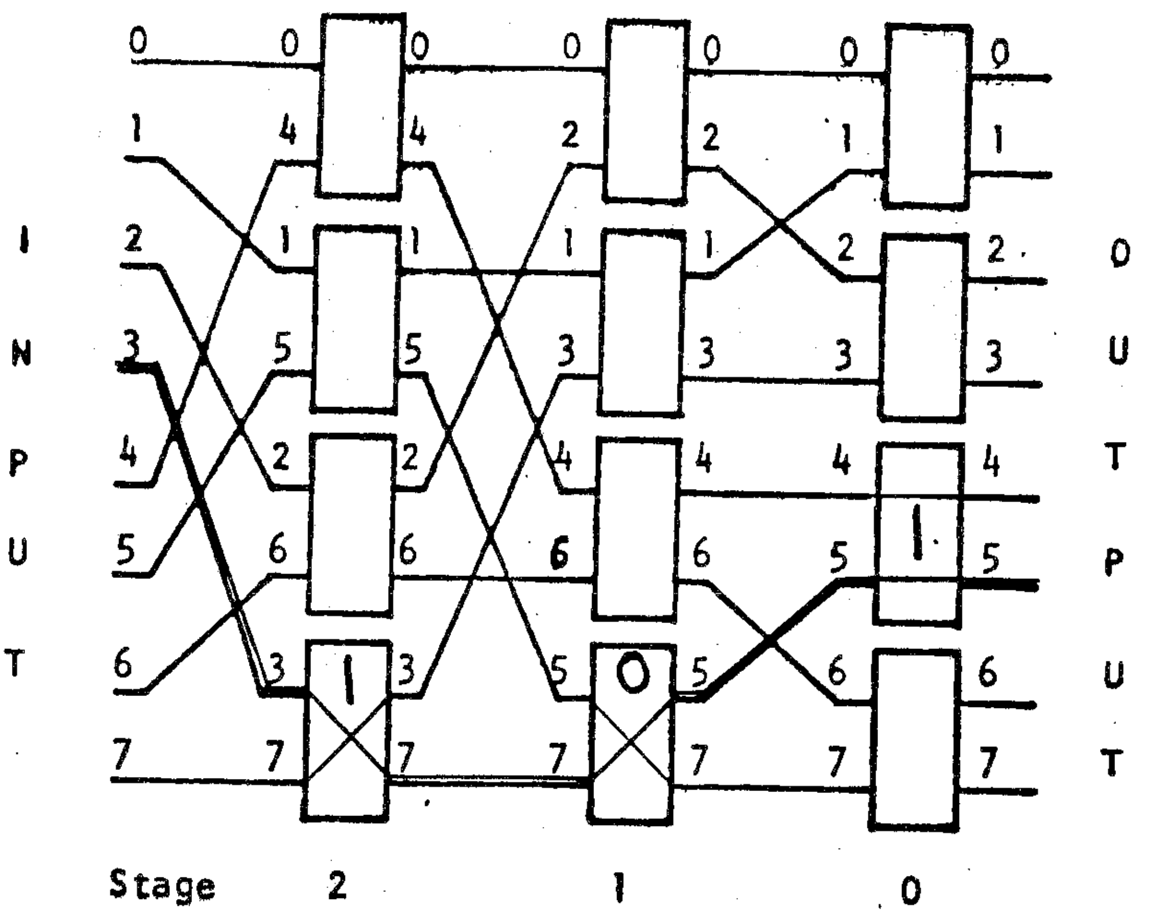


- use for 1-to-1 or permutations
- add m-bit broadcast mask for broadcasts
- tag can be used for check for correct destination

Destination Tag Example

$S = 3 = 011 \qquad D = 5 = 101$

$T = D = 101$



GENERALIZED CUBE TOPOLOGY FOR N = 8

Tag Generation

static

precomputed by compiler
processor fetches from memory
faster algorithm execution
compiler takes longer

dynamic

processor determines destination
processor determines routing tag
tag can be data conditional
process assignment to processor need not be known at compile time

could implement both -

choose most appropriate

Partitioning of Network

- form independent subnetworks
- each subnetwork has properties of Generalized Cube
- each partition size power of two
- partition sizes can vary
- routing tags can still be used
- operating system can use routing tags to enforce partitions
- no need for centralized network control
- many different ways to partition

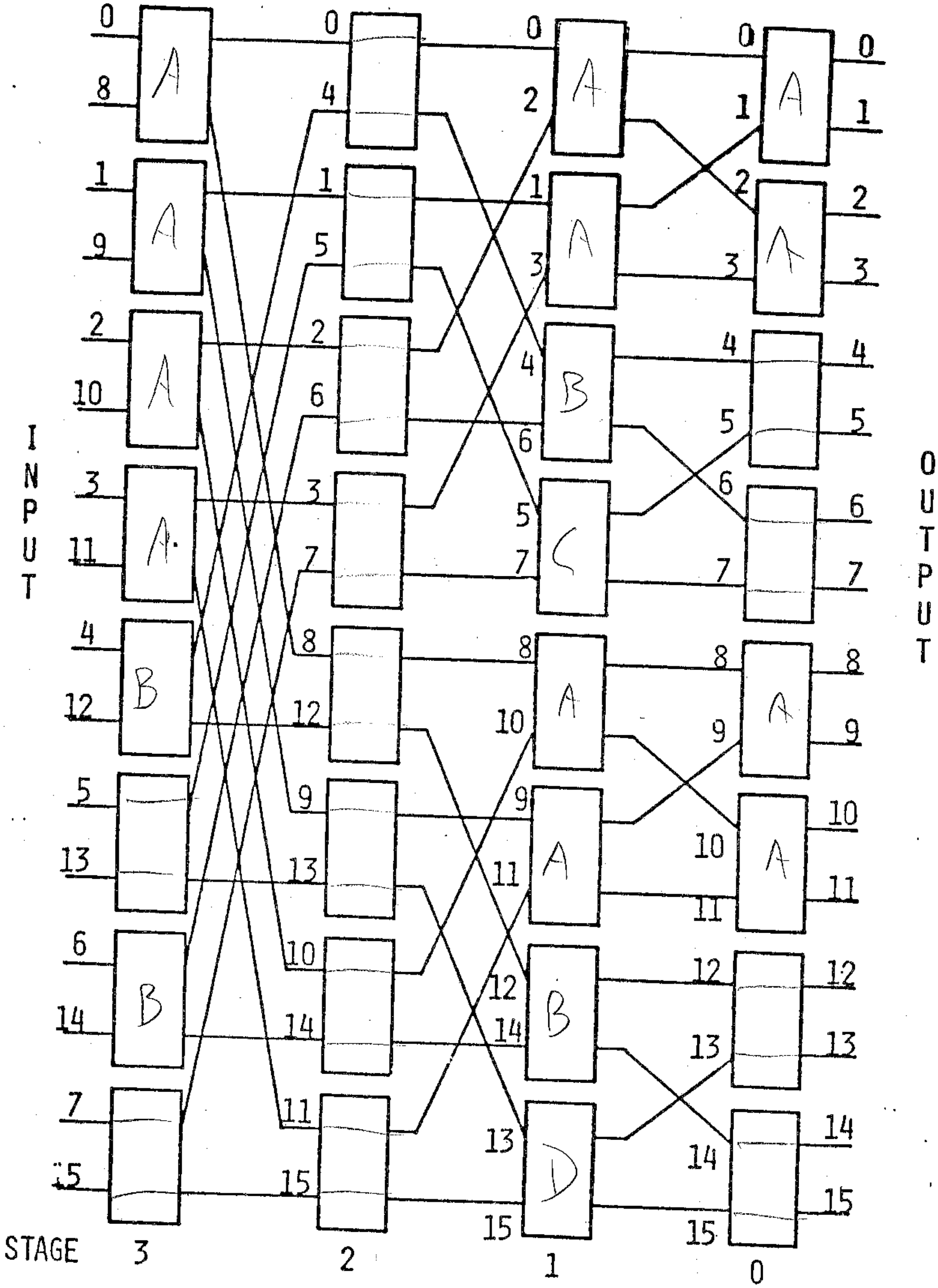
Reasons for Partitioning

- multiple - SIMD machine
  - set of CU's
  - partition PE's into independent SIMD machines

- reconfigurable SIMD/MIMD machines
  - partition system into independent SIMD/MIMD subsystems (PASM)
    - fault tolerance
    - multiple users
    - efficient size
    - program development
    - subtask parallelism

- SIMD machine
  - single CU, same program
  - multiple data sets
  - can improve efficiency

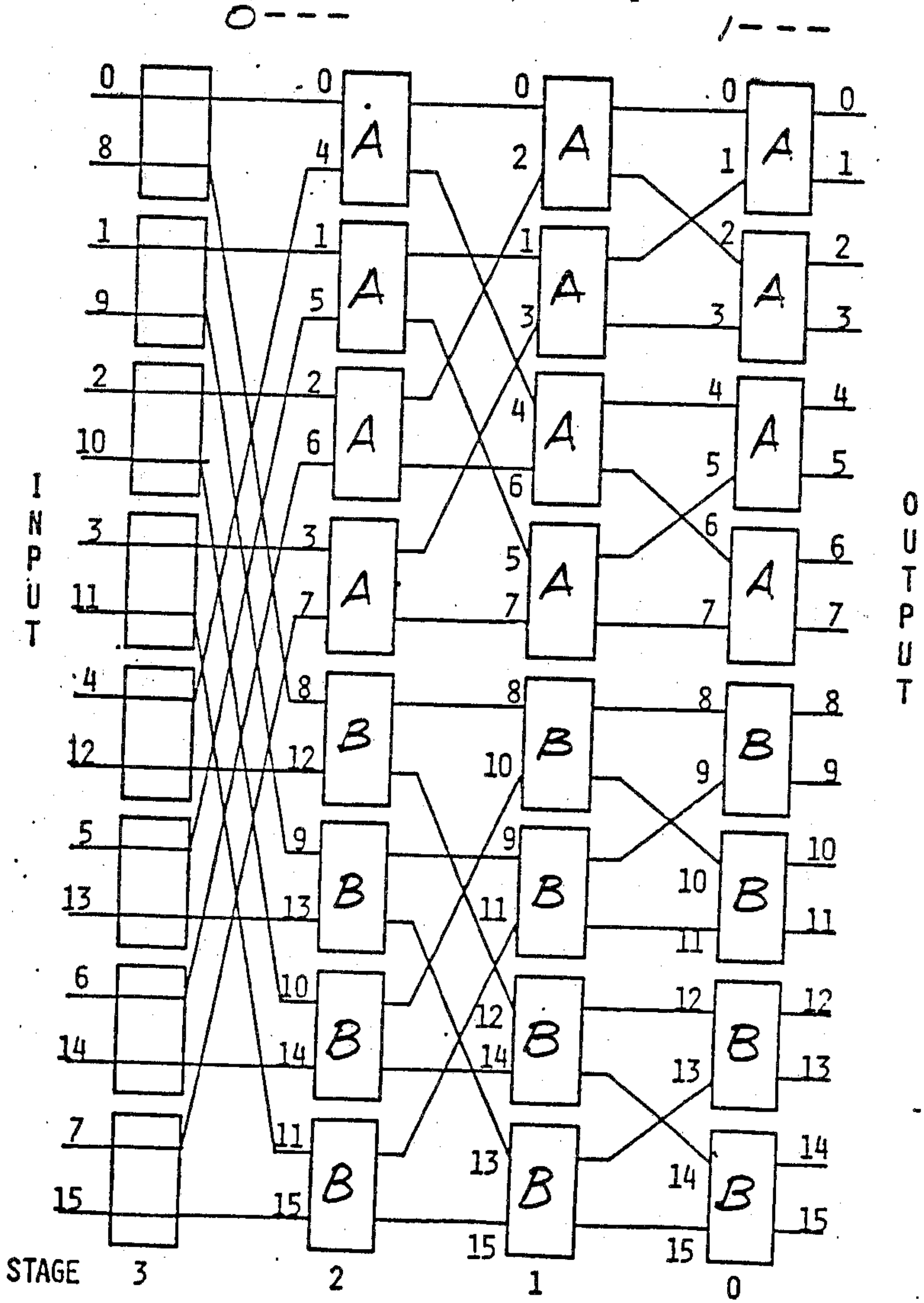
- MIMD machine
  - group PE's which communicate
  - reduce network conflicts



CUBE for N=16

Partitioning Example

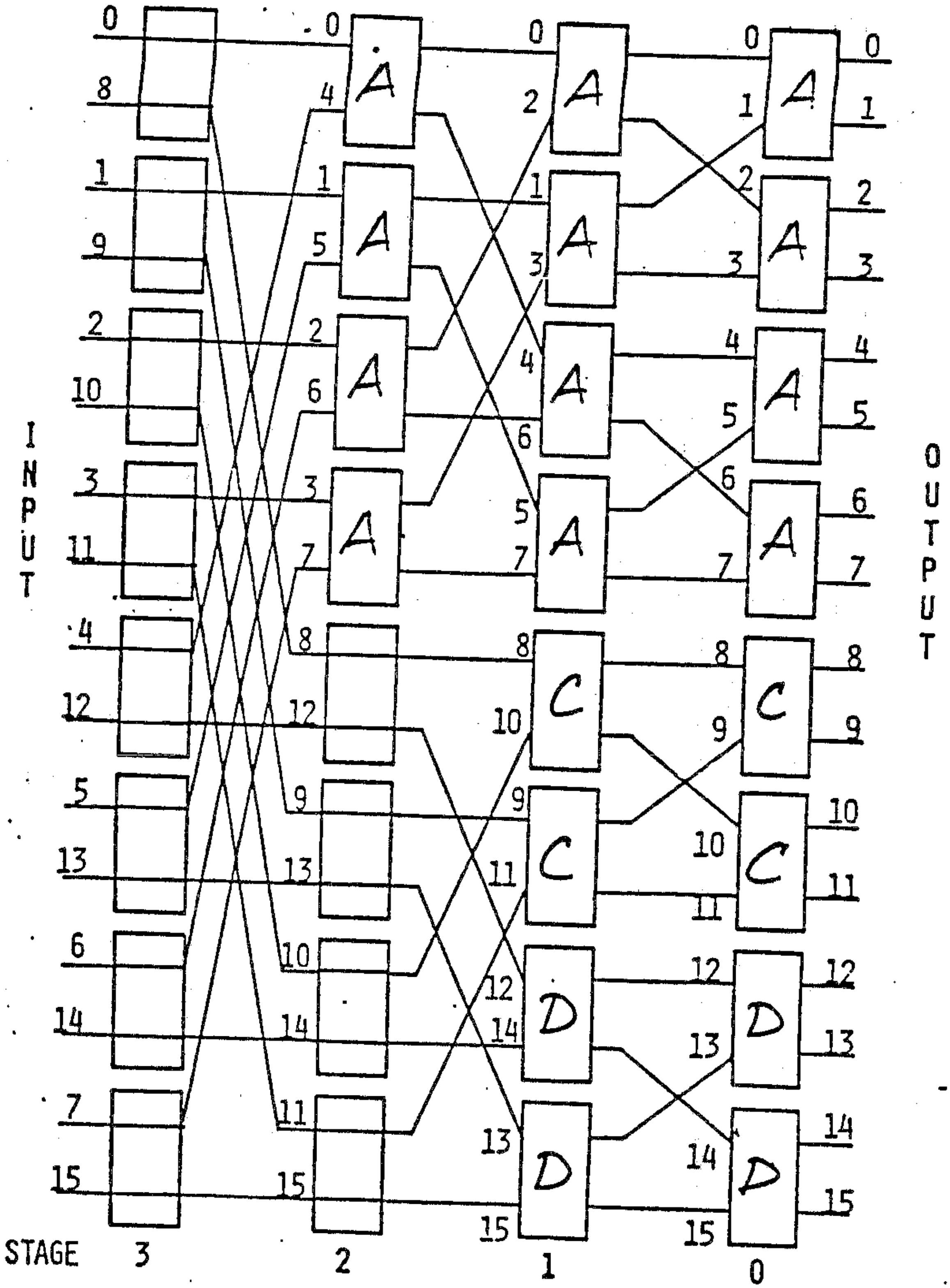
Group A: 0-7 , Group B: 8-15



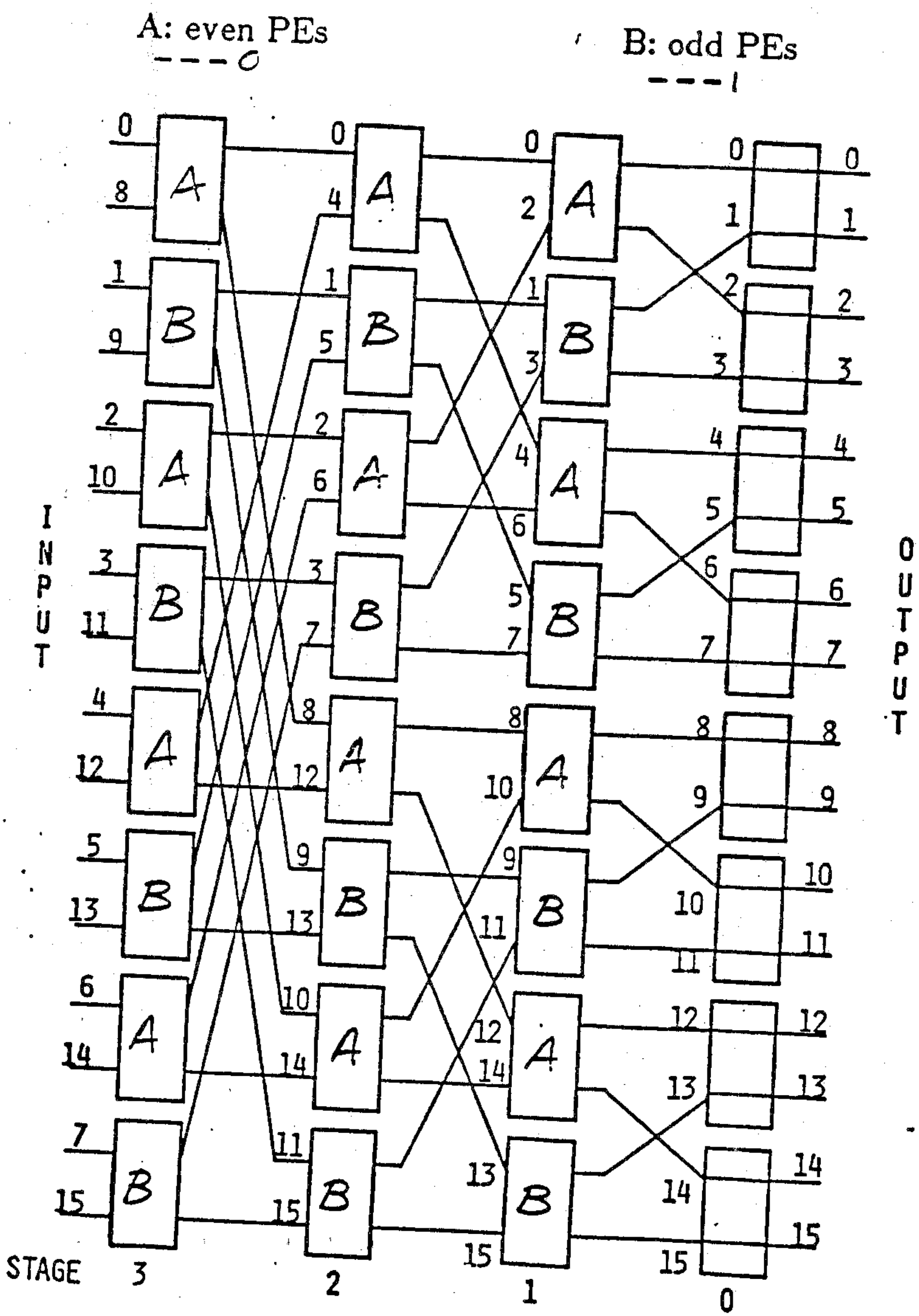
Partitioning Example

0--- 10-- 11--

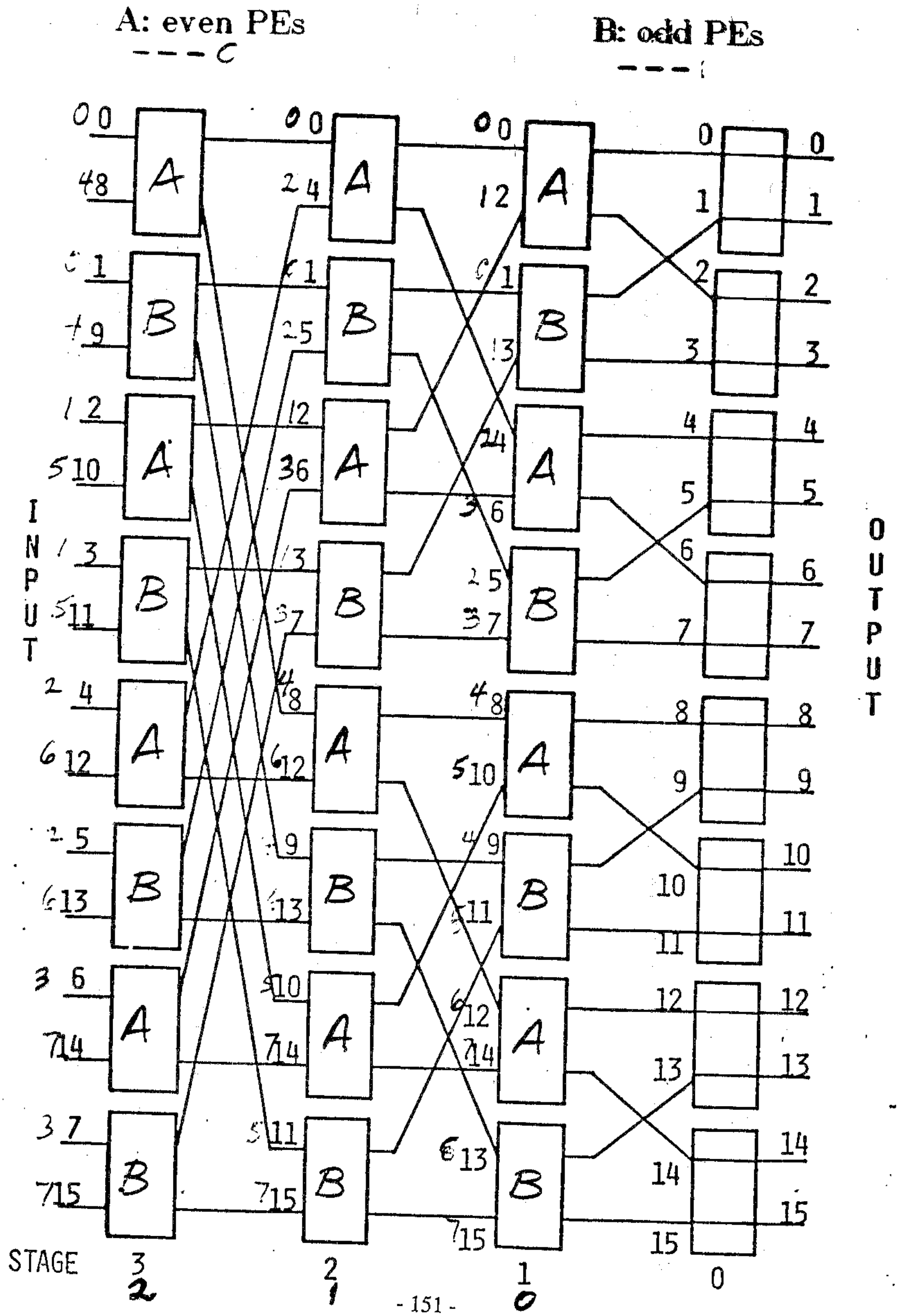
A: 0-7 C: 8-11 D: 12-15



Partitioning Example



Partitioning Example

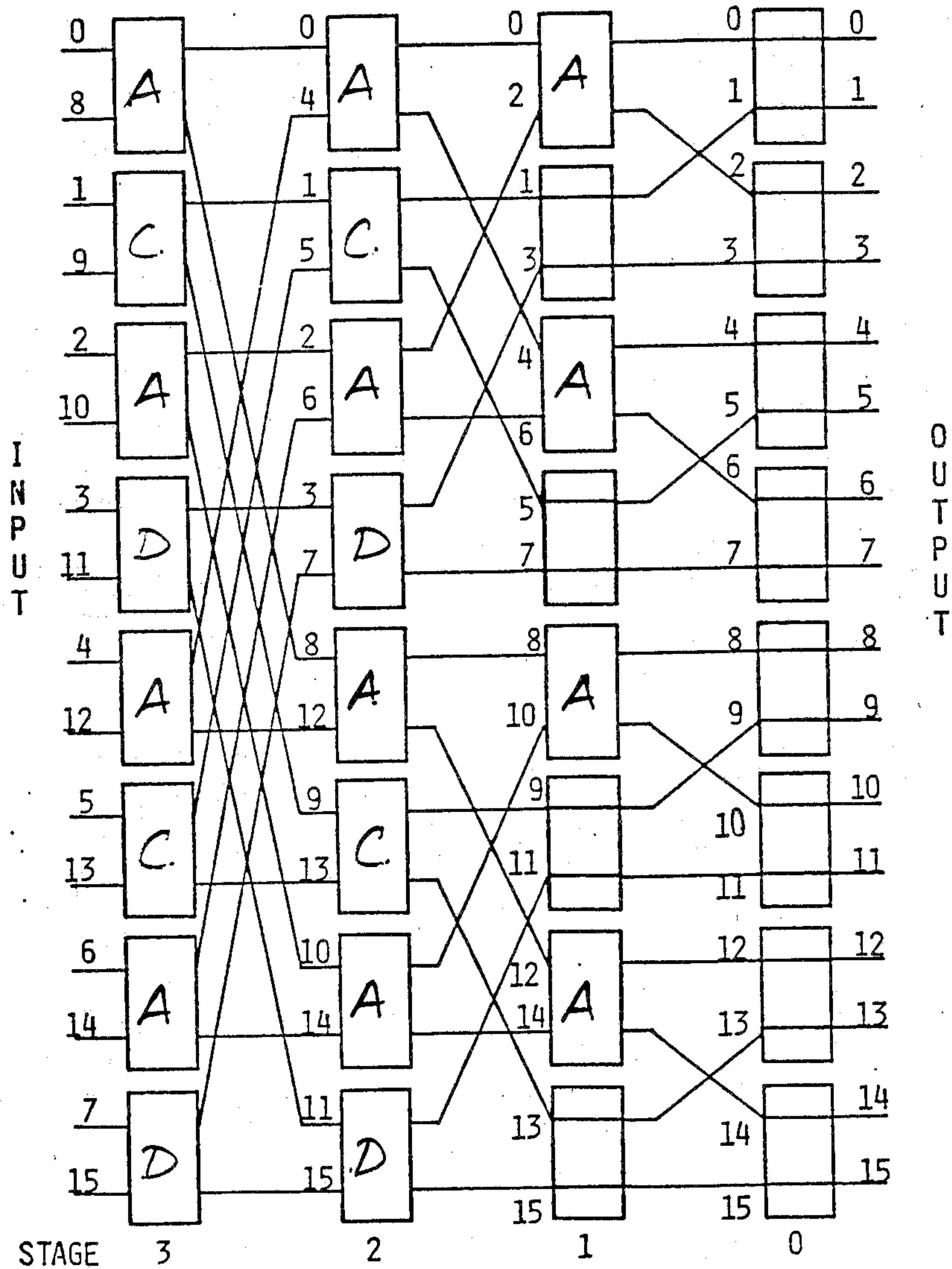


Partitioning Example

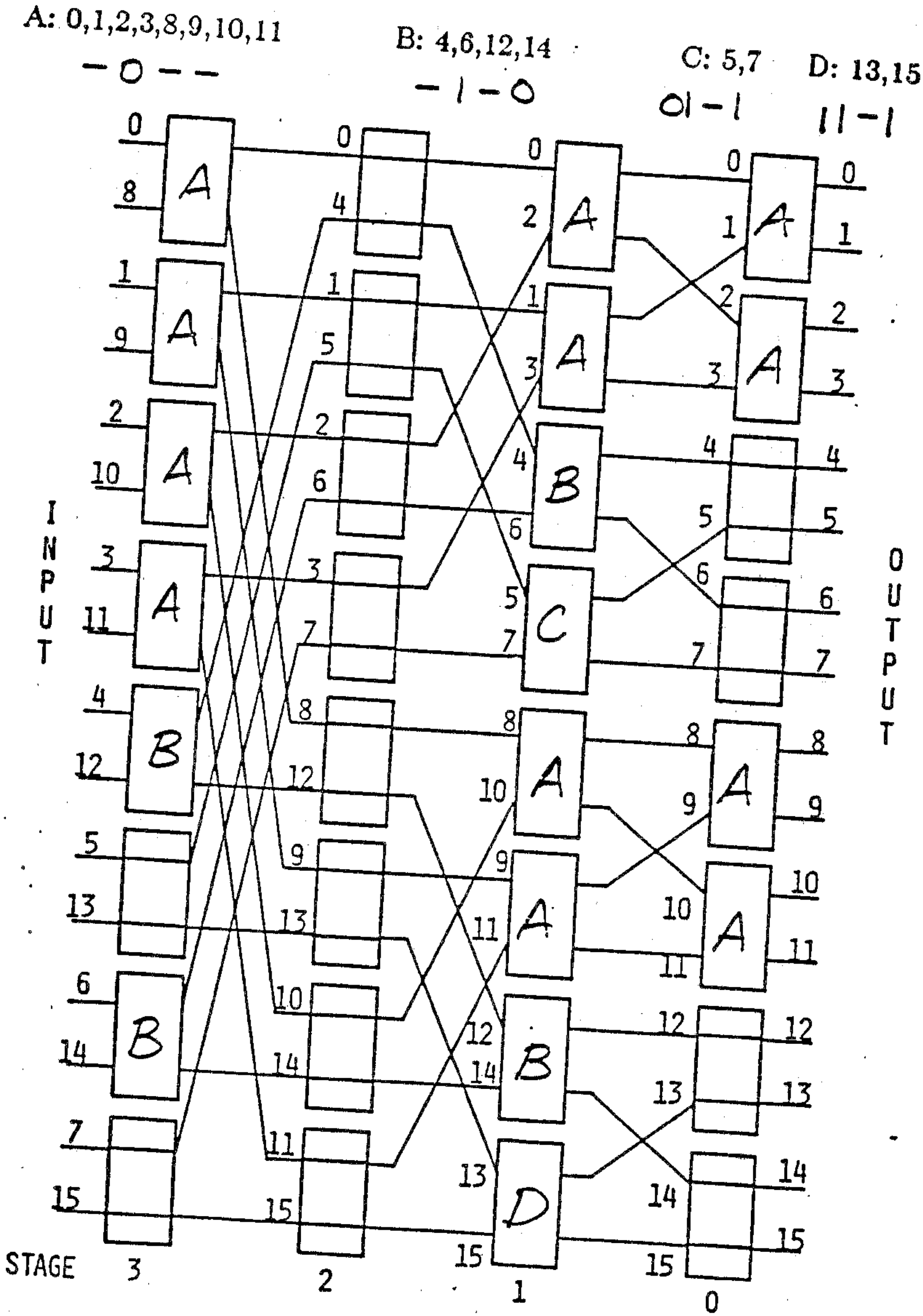
A: even PEs
---0

C: PEs 1,5,9,13
--01

D: PEs 3,7,11,15
--11



Partitioning Example: Groups of 8, 4, 2, 2



Partitioning the Generalized Cube Network

- all I/O ports in subnetwork of size 2^s agree in m−s bit positions
- interchange boxes used by this subnetwork set to straight in stages that correspond to these m−s bit positions
- other s stages make up subnetwork of size 2^s
- partitioning choices
  - — which stage to force to straight to divide (sub)network in half
  - — which subnetwork to further subdivide
- follows from theory of partitioning single stage Cube in Chap. 4
- transverse subnetwork from input to output, i^{th} stage not forced to straight is logical stage s−i, where $1 \leq i \leq s$
- for logical numbering of ports within subnetwork
  - — select from physical port address s bit positions in which ports disagree, in order, to use as logical number
  - — can complement any of the s bit positions as part of the mapping
  - — e.g., N = 16, subnetwork size 4 = {12, 13, 14, 15}
    $p_3p_2p_1p_0 \rightarrow p_1p_0$ or $p_1\overline{p}_0$ or $\overline{p}_1p_0$ or $\overline{p}_1\overline{p}_0$

Partitioning Generalized Cube

Cannot permute bits!

Multistage Cube-Type Networks

Relationship between generalized cube topology and:

1. SW-Banyan Networks
2. Omega (multistage shuffle - exchange) Network
3. STARAN Flip Network
4. Indirect Binary n-Cube Network

Comparison of Multistage Cube-Type Networks

- topology – actual interconnection patterns used to connect a set of N inputs to a set of N outputs

- interchange box type

  2-function: straight or exchange

  4-function: straight, exchange, upper broadcast, or lower broadcast

- control structure

  individual stage control: one control signal sets the state of all boxes in a stage (all are set to same state)

  partial stage control: i+1 control signals set the state of stage i (stage i divided into i+1 sets of boxes, all boxes in same set are in same state)

  individual box control: separate control signal sets the state of each box

Generalized Cube Network

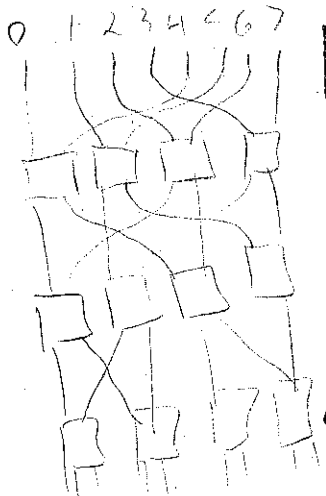
- Generalized Cube topology
- 4-function interchange boxes
- individual box control

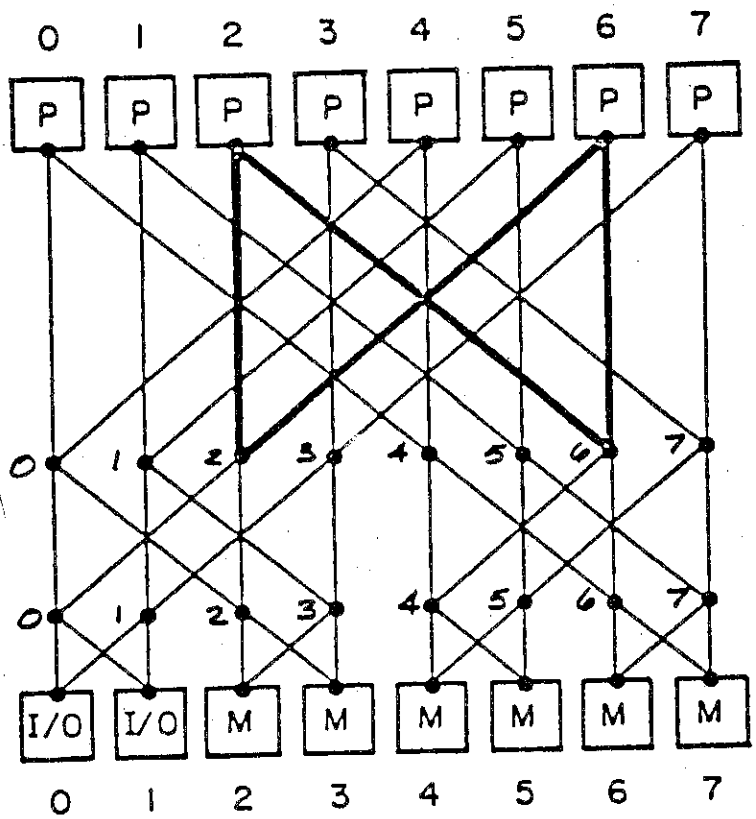
Banyan Networks -

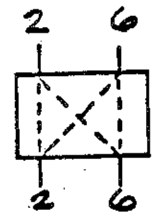
Class of Graphs

SW - Banyan Subclass — really a multistage cube

Spread = 2, Fanout = 2







Stage 2

Stage 1

Stage 0

Relationship Between SW-Banyan ($S = F = 2; L = m$) and Generalized Cube Networks

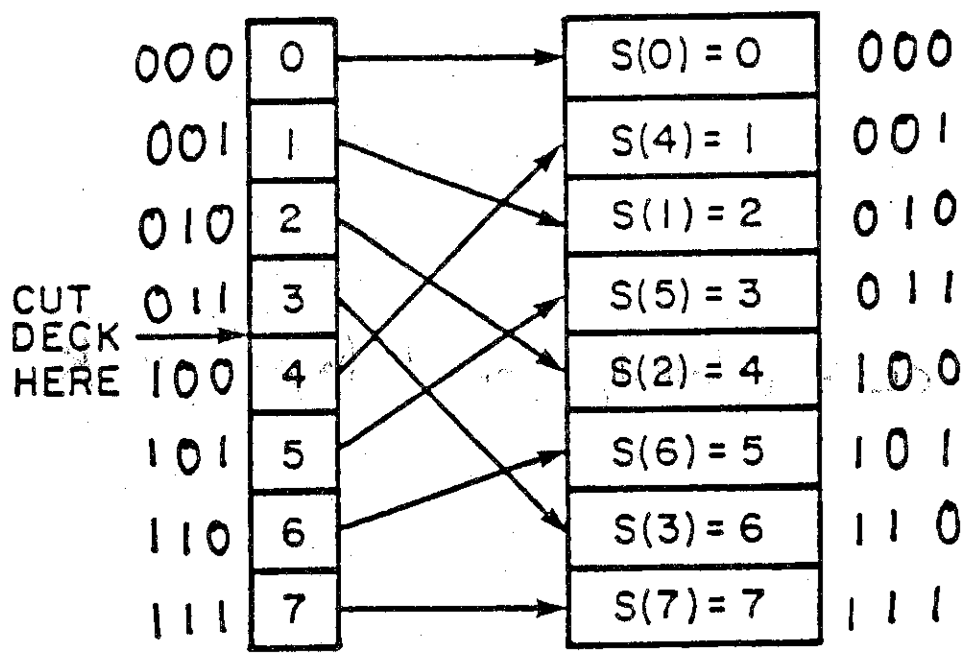
- Topology: equivalent, based on constructive definition of SW-banyans, definition of Generalized Cube, and treating edges as interchange boxes, and nodes as links
- box type: not specified for SW-Banyan (graph)
- control scheme: not specified for SW-Banyan (graph)

Omega Network -

multistage shuffle-exchange network

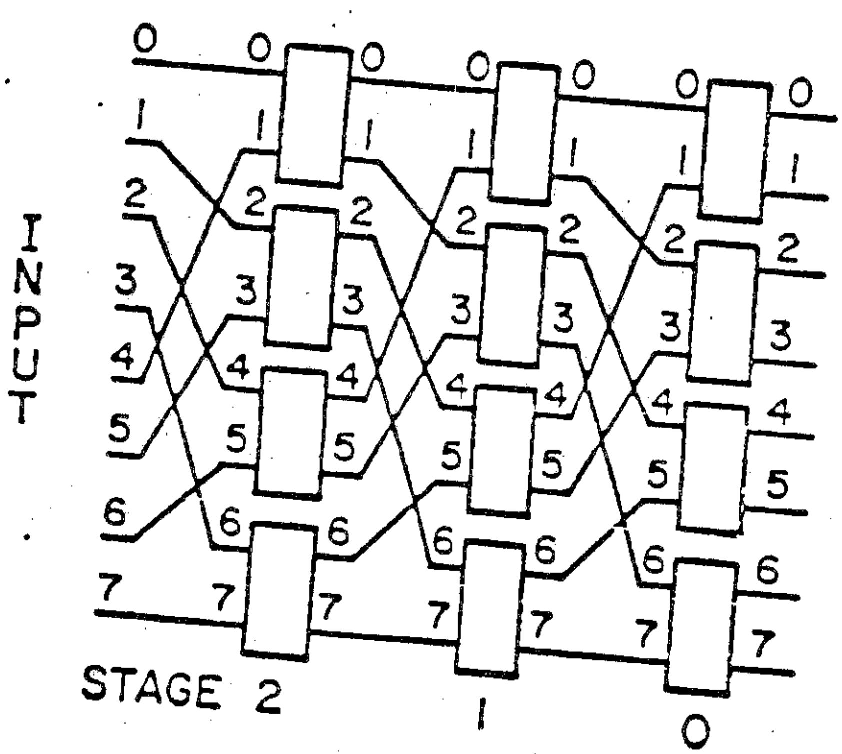
For $N = 2^m$ shuffle connects

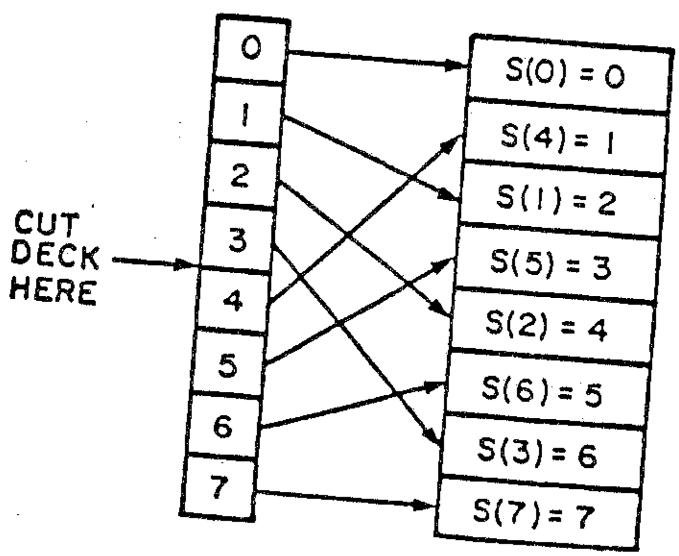
$p_{m-1} \cdots p_1 p_0 \rightarrow p_{m-2} \cdots p_1 p_0 p_{m-1}$



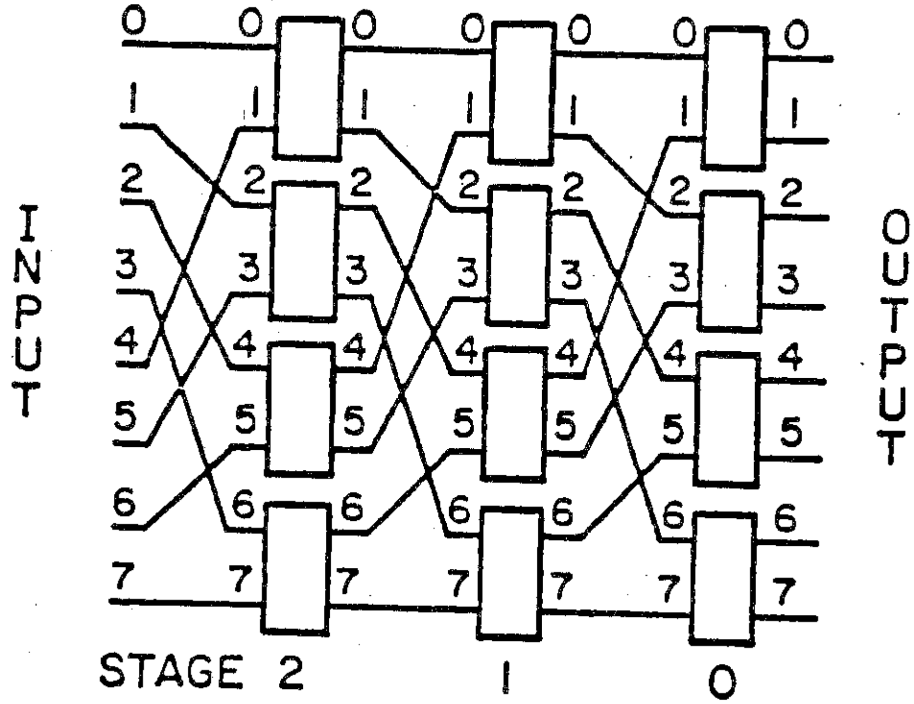
Omega Network for N = 8

Links labelled to show relation to shuffle

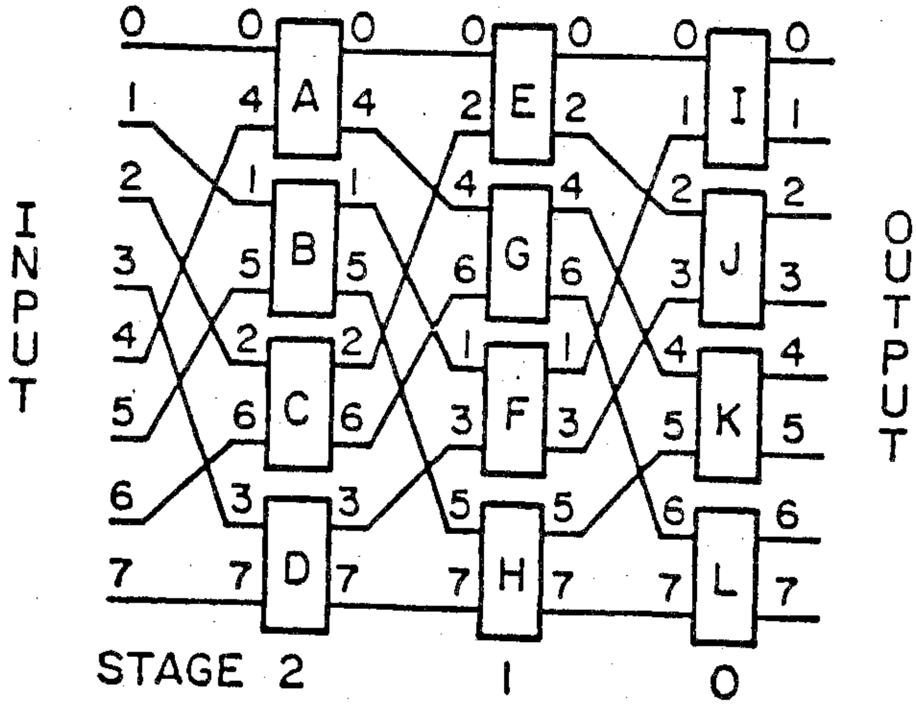




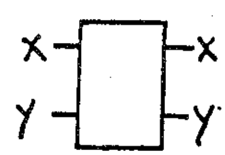
Omega Network for N = 8



labelled to show shuffle

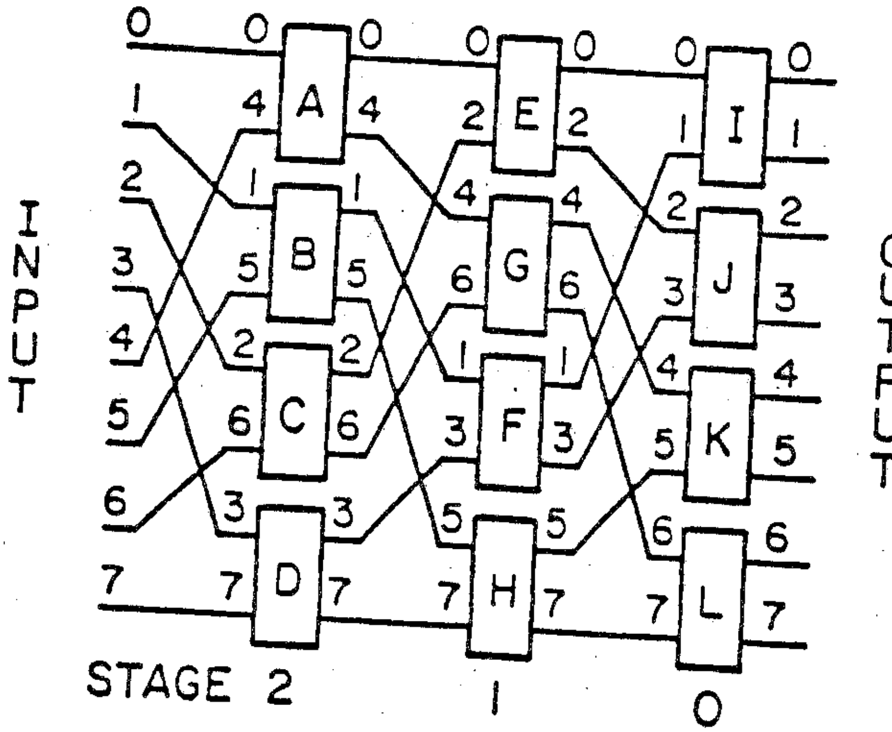


labelled using

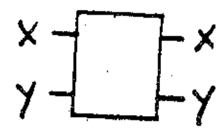


rule (like for cube)

Omega Network for N = 8

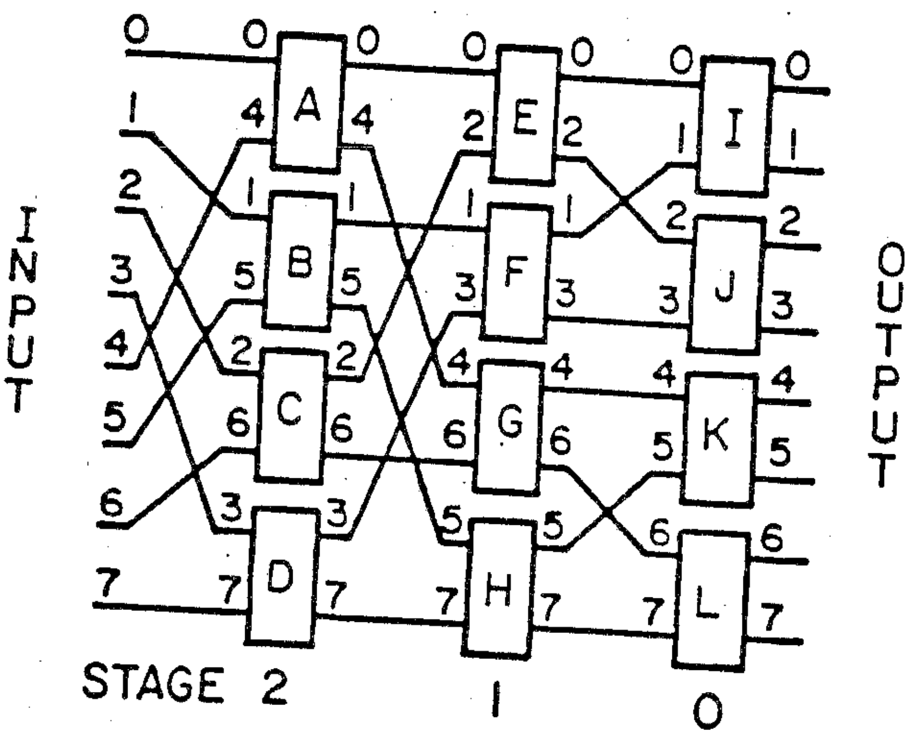


labelled using



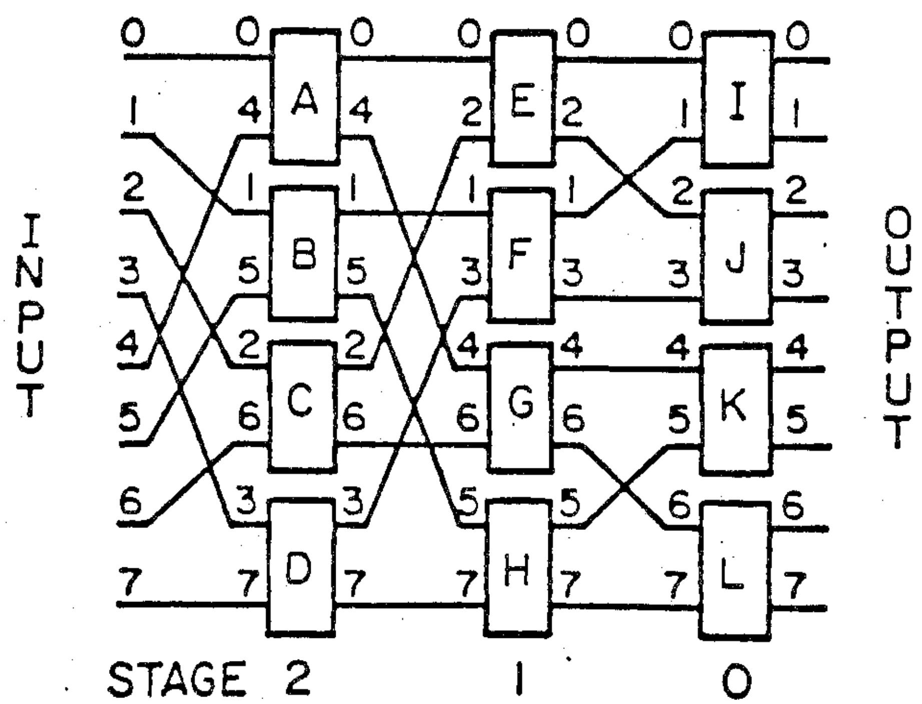
rule

(like for cube)

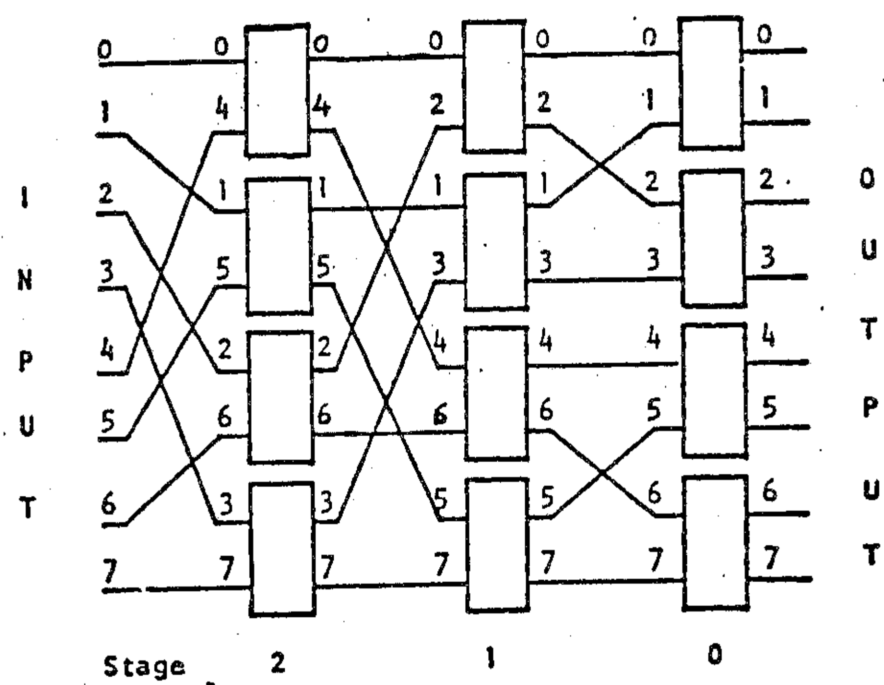


boxes F + G moved to show relation to cube

Omega = Generalized Cube



boxes
F + G
moved
to show
relation
to
cube



GENERALIZED CUBE TOPOLOGY FOR N = 8.

Relationship Between Generalized Cube and Omega Networks

- topology
  - Recall from Chap. 3 Shuffle-Exchange → Cube algorithm

$$\begin{aligned} \text{cube}_j(P) &= \text{shuffle}^j(\text{exchange}(\text{shuffle}^{m-j}(P))) \\ &= p_{m-1/j+1}\overline{p}_j p_{j-1/0} \end{aligned}$$

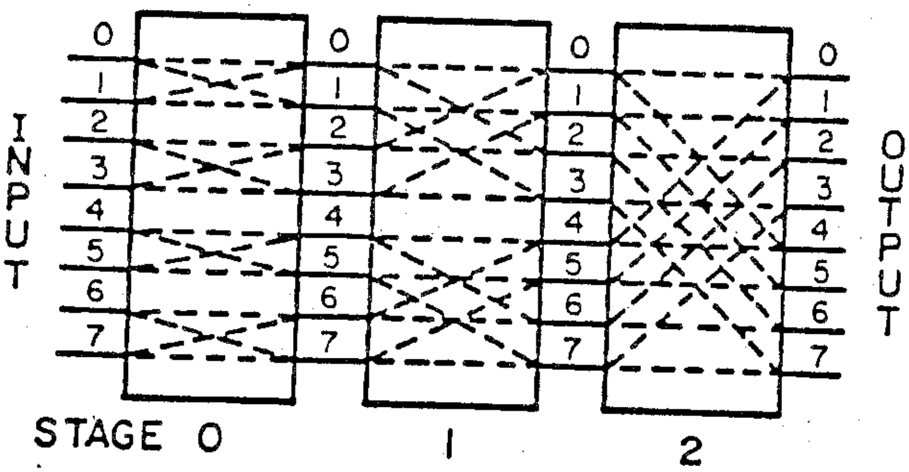
  - data entering stage j box in omega has been shuffled m−j times
  - setting a box to exchange is like performing the exchange function
  - stage j acts like cube_j
  - topologies are equivalent
- box type: 4-function for both
- control scheme: individual box for both

STARAN Flip Network

implemented for N = 256

SIMD Machine

shown for N = 8



Flip control

1 bit controls each stage

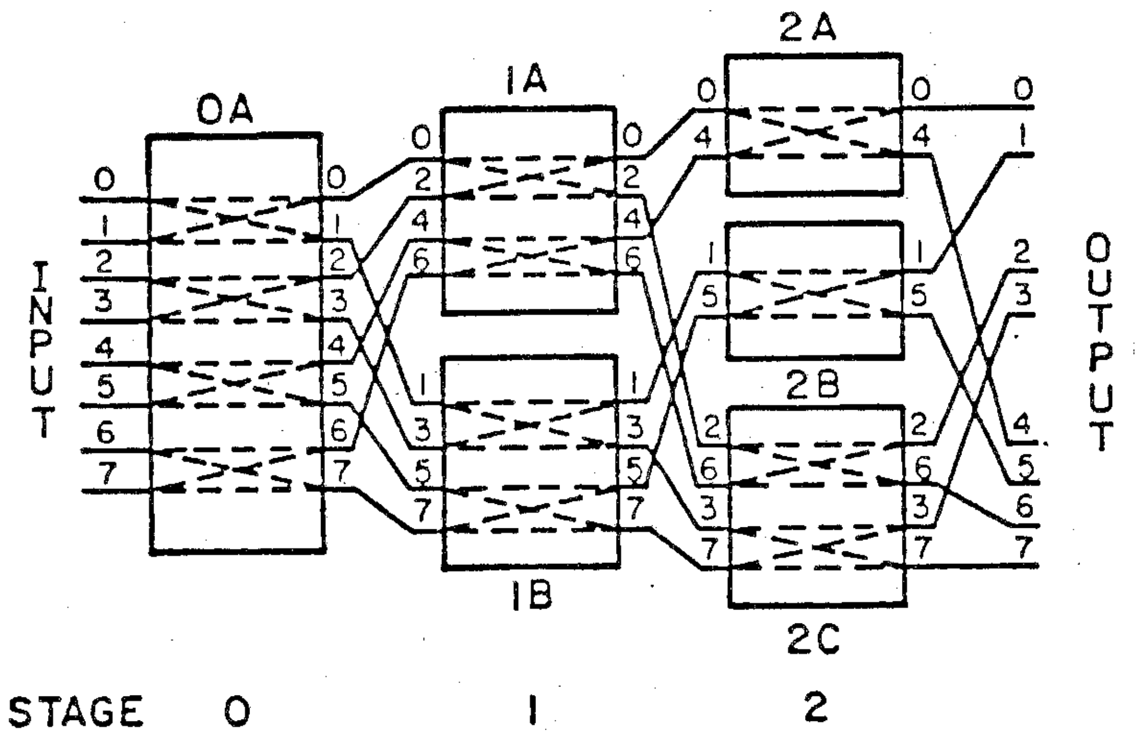
all boxes in a stage either straight or

all boxes exchange

STARAN Flip Network

shift control - $i + 1$ bits for stage i

different types of uniform shifts



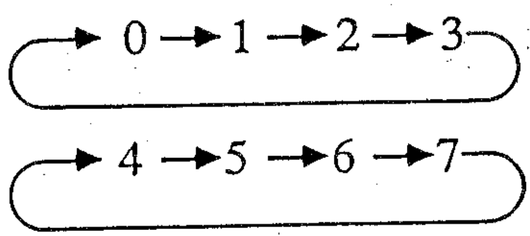
Ex. $x \rightarrow x + 1 \bmod N$

OA exchange	1B straight
1A	2B
2A	2C

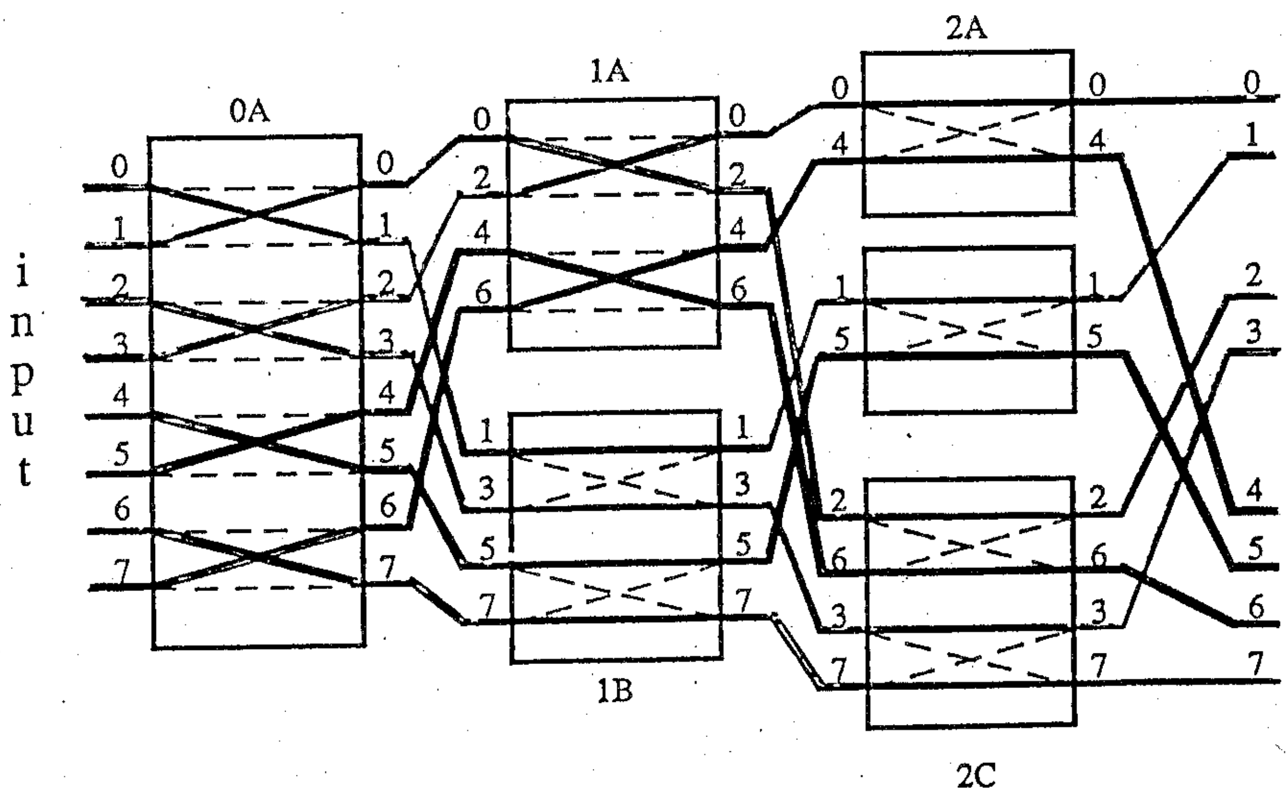
$0 \rightarrow 1, \; 3 \rightarrow 4$

STARAN Network Shift Control

		Control Signals					
Shift	Group Size	0A	1A	1B	2A	2B	2C
+1	8	1	1	0	1	0	0
+2	8	0	1	1	1	1	0
+4	8	0	0	0	1	1	1
→ +1	4	1	1	0	0	0	0 ←
+2	4	0	1	1	0	0	0
+1	2	1	0	0	0	0	0

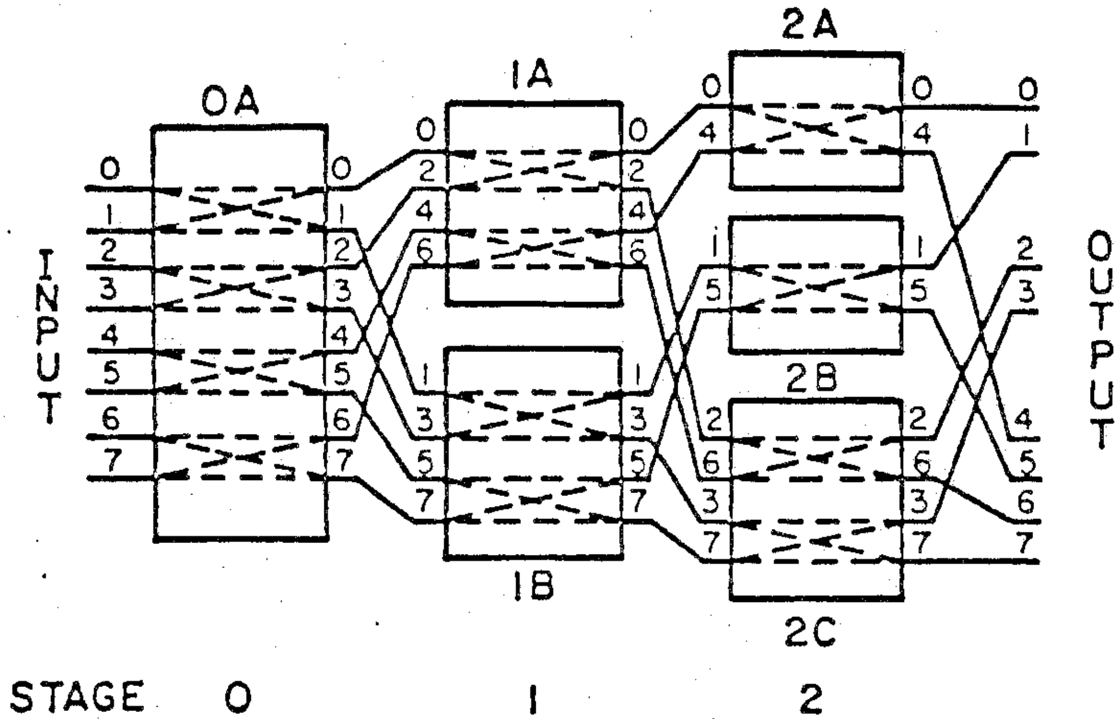


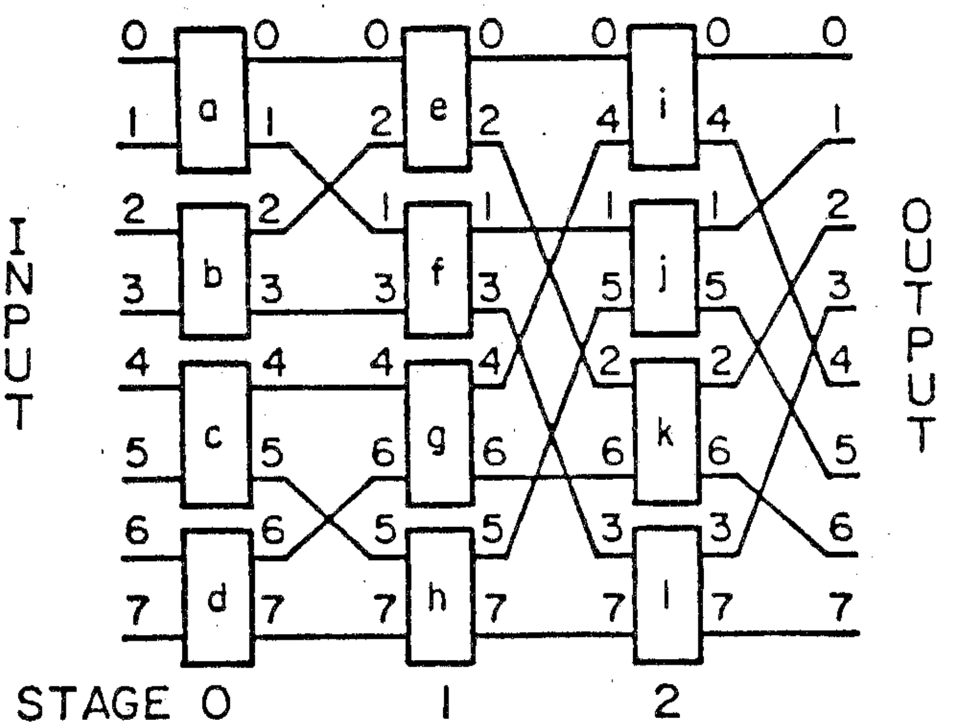
partition on
bit position 2



stage 0 2

STARAN Flip Network





0A: a, b, c, d

1A: e, g
1B: f, h
2A i

2B: j

2C: k, l

Generalized Cube with order of stages reversed

STARAN Shift Control

- related to Chap. 3 Cube $\rightarrow$ PM2I algorithm
  and Chap. 4 Cube partitioning results

- each shift of $+2^i \bmod N \equiv PM2_{+i}$

- Chap. 3 Cube $\rightarrow$ $PM2_{+i}$ algorithm:

  for j = m−1 step −1 to i do

  $cube_j \; [X^{m-j} 1^{j-i} X^i]$

- STARAN flip network does $cube_0$, $cube_1$,...

  for j = i to m−1 do

  $cube_j \; [X^{m-j} 0^{j-i} X^i]$

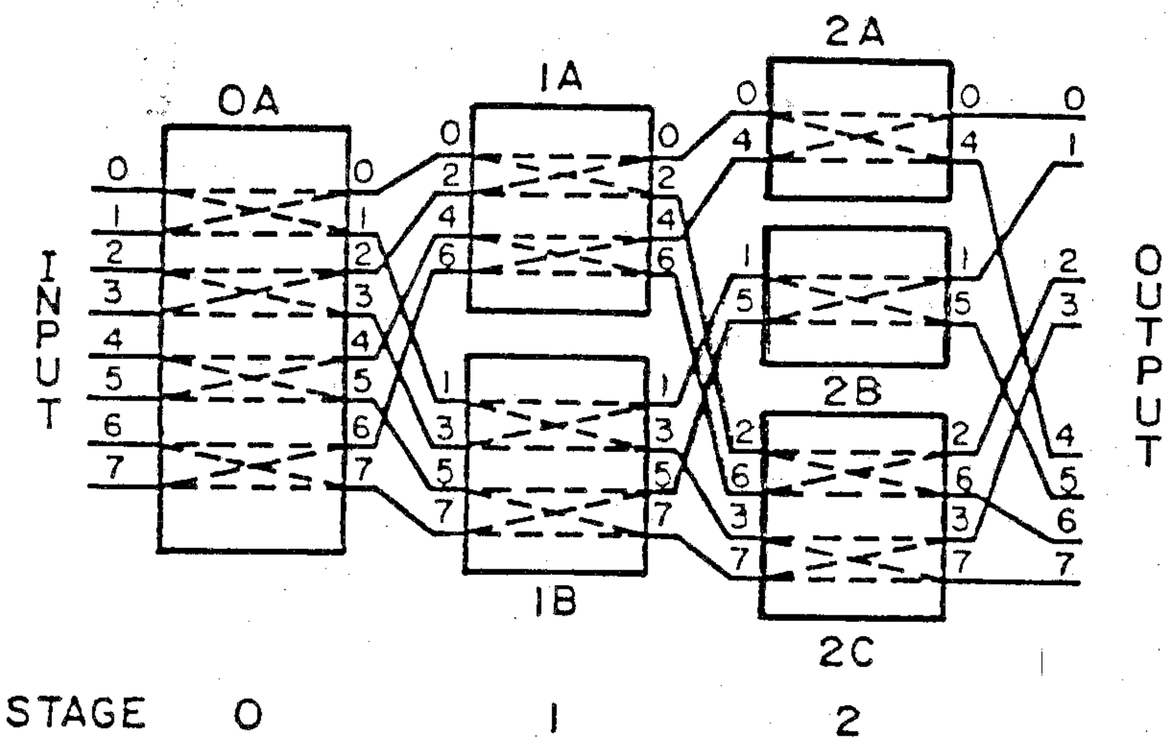
  ex. i = 0, m = 3

  $cube_0$ [XXX]

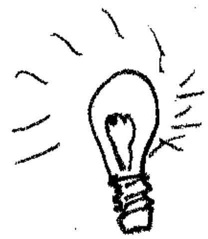
  $cube_1$ [XX0]

  $cube_2$ [X00]

STARAN Shift Controls



0A = [XXX], 1A = [XX0], 1B = [XX1], 2A = [X00], 2B = [X01], 2C = [X1X].



Cube $\rightarrow$ PM2_{+i}

for j = i to m−1 do

  $\text{cube}_j\ [X^{m-j}0^{j-i}X^i]$

N=8:

+1	cube_0	[XXX]
	cube_1	[XX0]
	cube_2	[X00]
+2	cube_1	[XXX]
	cube_2	[X0X]
+4	cube_2	[XXX]

STARAN Shift Control N=8

0:	0A:	XXX	+1		
1:	1A:	XX0	+1	+2	
	1B:	XX1			
2:	2A:	X00	+1	+2	+4
	2B:	X01			
	2C:	X1X			

STARAN Shift Controls

$0A = [XXX]$, $1A = [XX0]$, $1B = [XX1]$, $2A = [X00]$, $2B = [X01]$, $2C = [X1X]$.

$PM2_{+0}$:

$\text{cube}_0 \; [XXX] \equiv 0A = 1$

$\text{cube}_1 \; [XX0] \equiv 1A = 1$

$\text{cube}_2 \; [X00] \equiv 2A = 1$

$PM2_{+1}$:

$\text{cube}_1 \; [XXX] \equiv 1A = 1B = 1$

$\text{cube}_2 \; [X0X] \equiv 2A = 2B = 1$

$PM2_{+2}$:

$\text{cube}_2 \; [XXX] \equiv 2A = 2B = 2C = 1$

N=128

Stage 5
6 signals

SA XX00000 } XX0000X
SB XX00001
SC XX0001X XX000XX
SD XX001XX
SE XX01XXX
SF XX1XXXX

Do THIS FOR TEST →

STARAN for N=16

Shift Controls Needed

0:	0A:	XXXX	+1			
1:	1A:	XXX0	+1	+2		
	1B:	XXX1				
2:	2A:	XX00	+1	+2	+4	
	2B:	XX01				
	2C:	XX1X				
3:	3A:	X000	+1	+2	+4	+8
	3B:	X001				
	3C:	X01X				
	3D:	X1XX				

Cube $\rightarrow$ $PM2_{+i}$

for j = i to m−1 do

cube$_j$ $[X^{m-j}0^{j-i}X^i]$

$sum(w) = \cancel{20}\ w(w+1)/2$

N=1024 $sum(10) = 110/2 = 55$ signals

STARAN Shift Controls

0A = [XXX], 1A = [XX0],

1B = [XX1], 2A = [X00],

2B = [X01], 2C = [X1X].

- for shifting $+2^j$ within groups of size 2^k
  - — all elements in a group numbered consecutively
  - — all elements in a group agree in high-order $m-k$ bit positions
  - — partition by disallowing use of $cube_i$ for $k \leq i < m$
- Ex. $k = 2$, $N = 8$, $j = 0$, $+1$ shift

  $cube_0$ [XXX] $\equiv$ 0A = 1

  $cube_1$ [XX0] $\equiv$ 1A = 1

  $4 \rightarrow 5, 5 \rightarrow 4 \rightarrow 6, 6 \rightarrow 7, 7 \rightarrow 6 \rightarrow 4$
- Ex. $k = 2$, $N = 8$, $j = 1$, $+2$ shift

  $cube_1$ [XXX] $\equiv$ 1A = 1B = 1

  $4 \rightarrow 6, 6 \rightarrow 4, 5 \rightarrow 7, 7 \rightarrow 5$

1A
1B
2A ?
2B ?

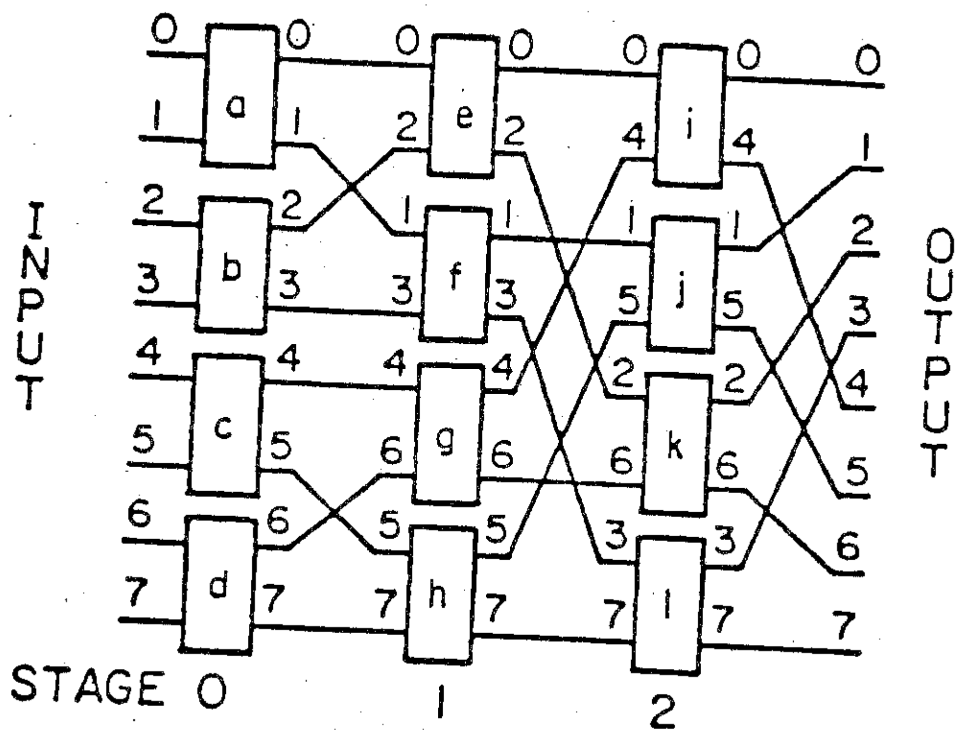
Relationship of STARAN Flip Network to Generalized Cube Network

- topology – STARAN network equivalent to Generalized Cube with stages in reverse order
- box type: STARAN is 2-function

  Generalized Cube is 4-function
- control scheme: STARAN is partial stage and individual stage

  Generalized Cube is individual box

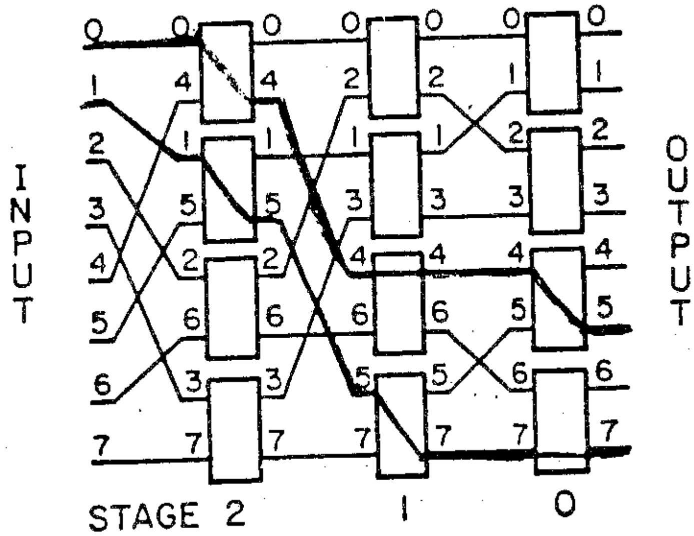
Indirect Binary n-Cube ($n = \log_2 N$) for $N = 8$



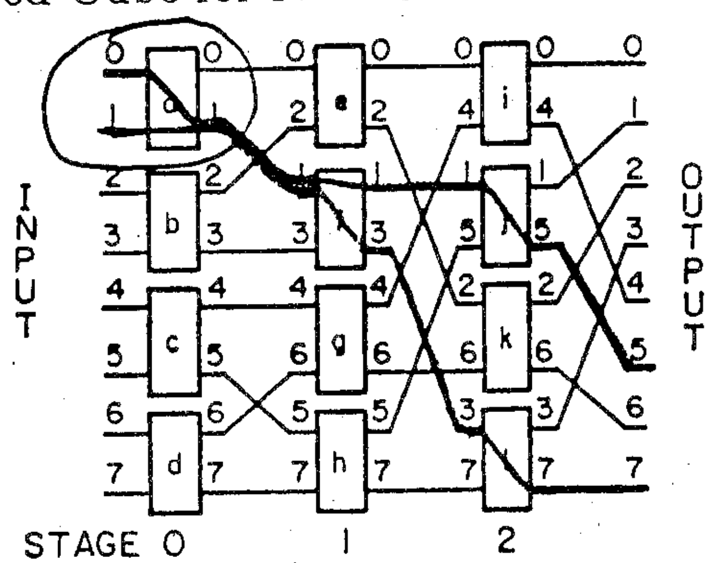
same topology as STARAN flip
individual box control
only straight or exchange
stage order reverse of Generalized Cube

Generalized Cube versus Indirect Binary n-Cube

Permutations - cannot do same permutations due to reversed order stages, e.g., 0 to 5 and 1 to 7



Generalized Cube for N = 8



Indirect Binary n-Cube for N = 8

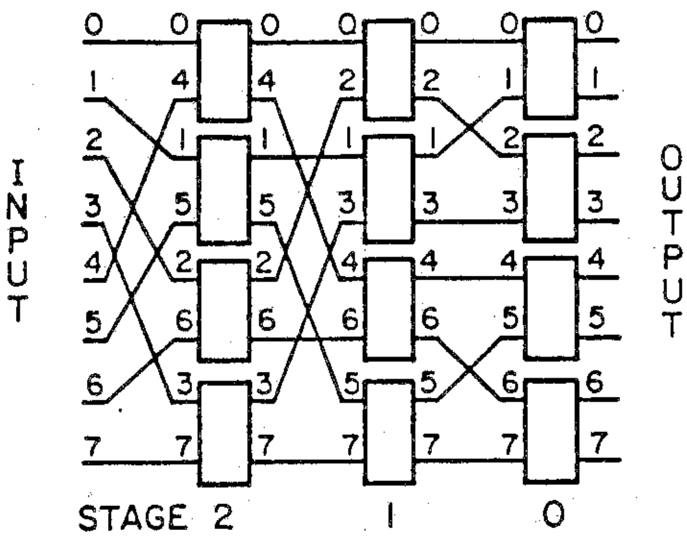
Generalized Cube versus Indirect Binary n-Cube

If Generalized Cube can perform permutation f,

then Indirect Binary n-Cube can perform f^{-1}

$P \rightarrow f(P) \qquad f(P) \rightarrow P = f^{-1}(f(P))$

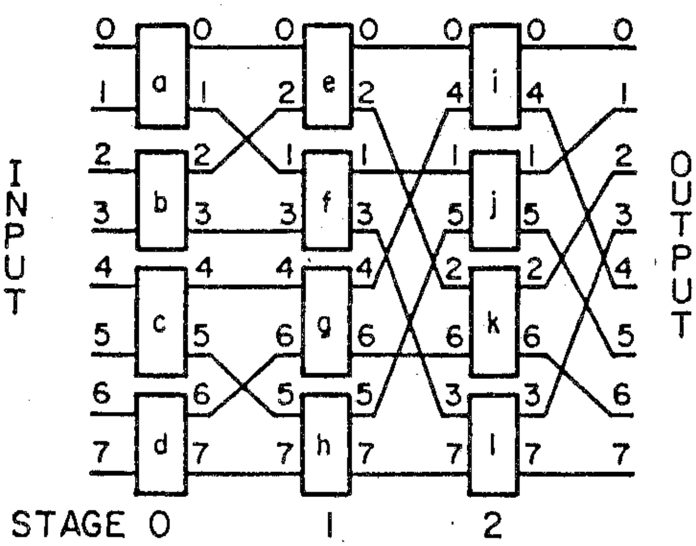
P



f(P)

Generalized Cube for N = 8

f(P)

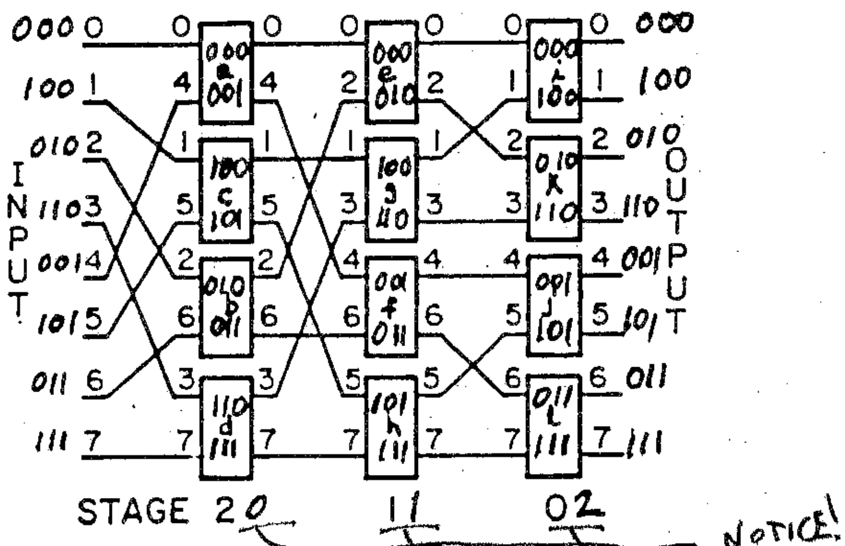


P = f⁻¹(f(P))

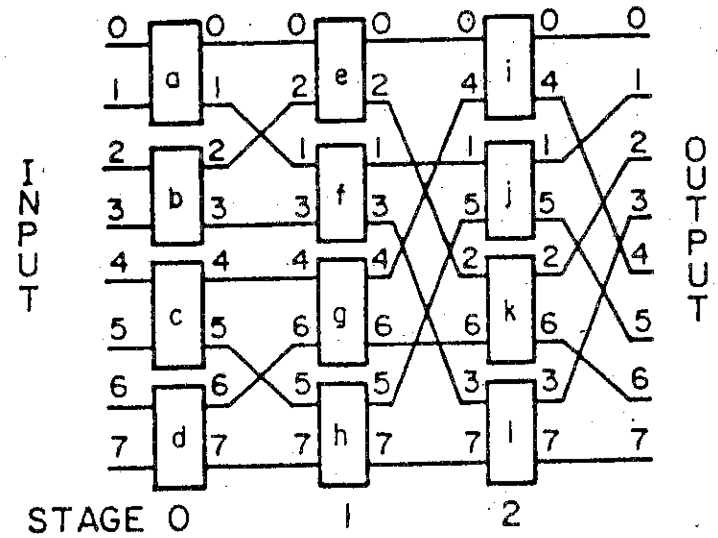
Indirect Binary n-Cube for N = 8

Generalized Cube versus Indirect Binary n-Cube

If logically relabel each I/O port P as Reverse (P), where $\text{Reverse}(p_{m-1} \ldots p_1 p_0) = p_0 p_1 \ldots p_{m-1}$ then can use Generalized Cube to emulate Indirect Binary n-Cube, and vice versa.



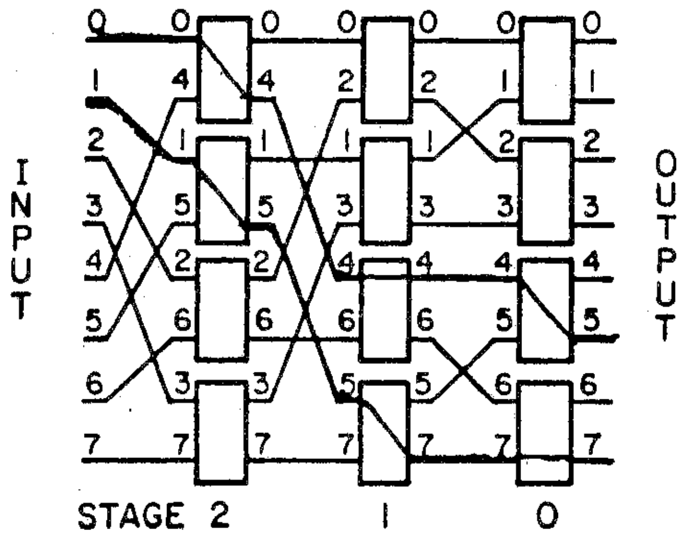
Generalized Cube for N = 8



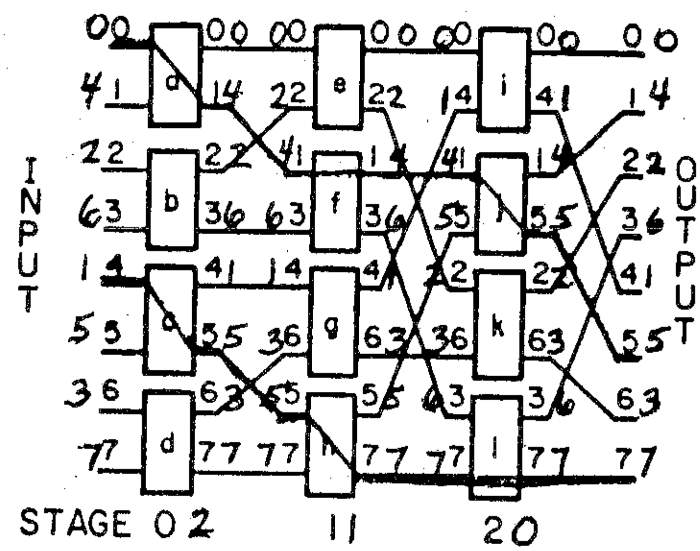
Indirect Binary n-Cube for N = 8

Generalized Cube versus Indirect Binary n-Cube

Permutations - cannot do same permutations due to reversed order stages, e.g., 0 to 5 and 1 to 7



Generalized Cube for N = 8



Indirect Binary n-Cube for N = 8

Fault Detection and/or Location Techniques for Multistage Cube Networks

1. send destination address
2. parity/ECC on data/tags at I/O ports
3. parity/ECC at each interchange box
4. use handshaking protocol
5. timer for timeouts
6. test bit-patterns
7. combinations of above

Techniques for Making Multistage Cube Networks Fault Tolerant

1. extra stage
2. extra links
3. extra switches
4. extra interchange box (switch) complexity
5. extra network
6. extra bits for ECC
7. extra control - bit/byte slice - degrade/spares
parity/ECC across slices
8. extra passes
9. combinations of above

Advantages of Cube Network Include:

- up to N simultaneous transfers
- partitionable into independent subnetworks
- one device can broadcast to all or subset
- distributed network control using routing tags
- variety of implementation options
- can use SIMD in addition to MIMD

EXTRA STAGE CUBE NETWORK

1. network structure - single fault tolerant

2. paths through network

3. routing tag control

4. partitioning

5. multiple fault handling

6. enhancement

Advantages of Cube Network Include:

- up to N simultaneous transfers
- partitionable into independent subnetworks
- one device can broadcast to all or subset
- distributed network control using routing tags
- variety of implementation options
- can use SIMD in addition to MIMD

Disadvantage:

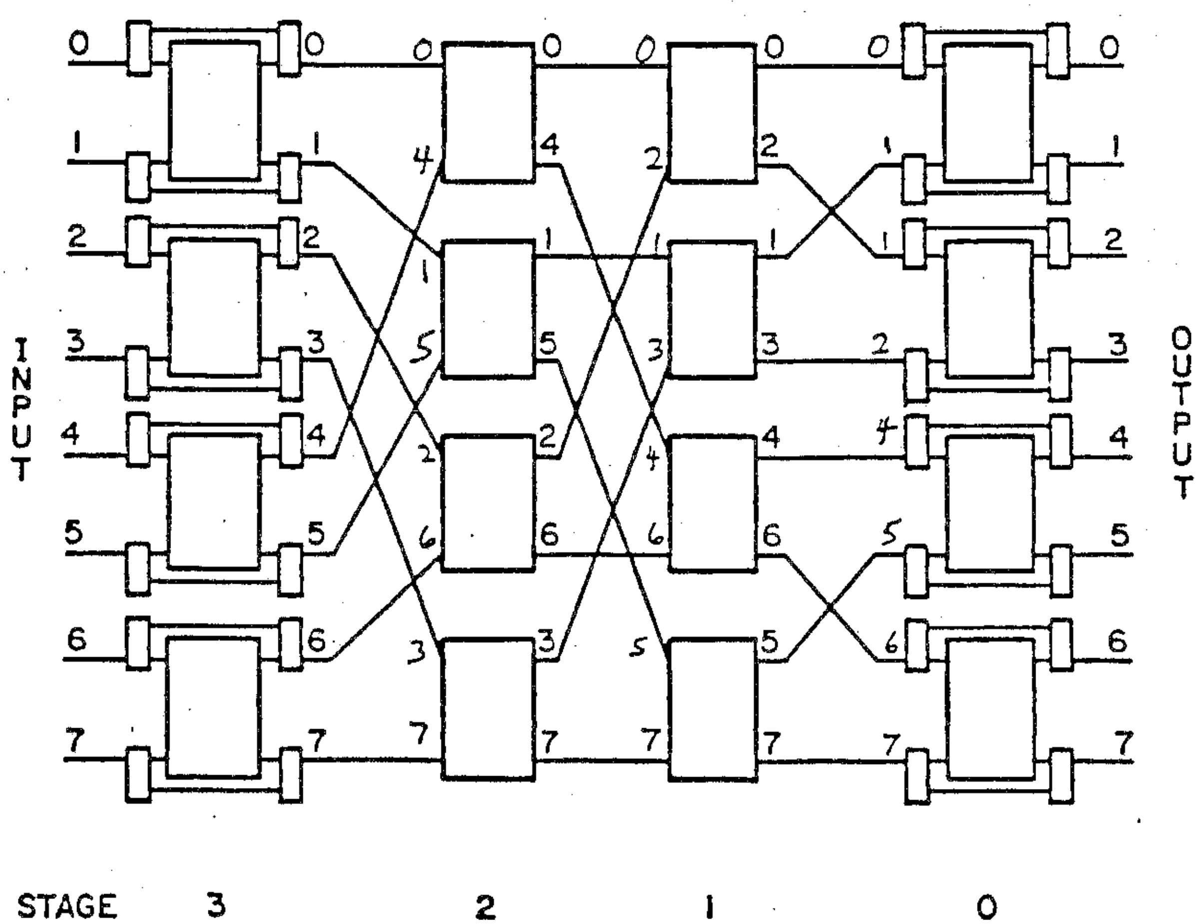
- only one path between given source and given destination - not single fault tolerant

Extra Stage Cube

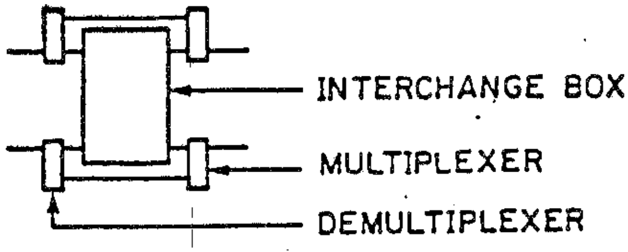
- Based on "popular" multistage cube network
- All advantages of multistage cube network
- Single-fault tolerant
- Robust given two faults
- Techniques for determining if particular multiple faults prevent full functioning, and if so, which I/O ports affected

Extra Stage Cube

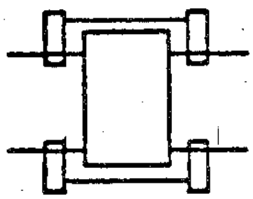
- single-fault tolerant
- add extra stage to input side of Generalized Cube
  - *stage m* (m = $\log_2 N$)
- stage m pairs lines differ in 0^{th} bit (like stage 0)
- simple bypass circuitry for stages m and 0



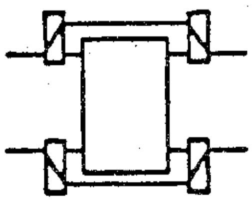
Detail of Stage m and 0 Interchange Box



block diagram



interchange box enabled

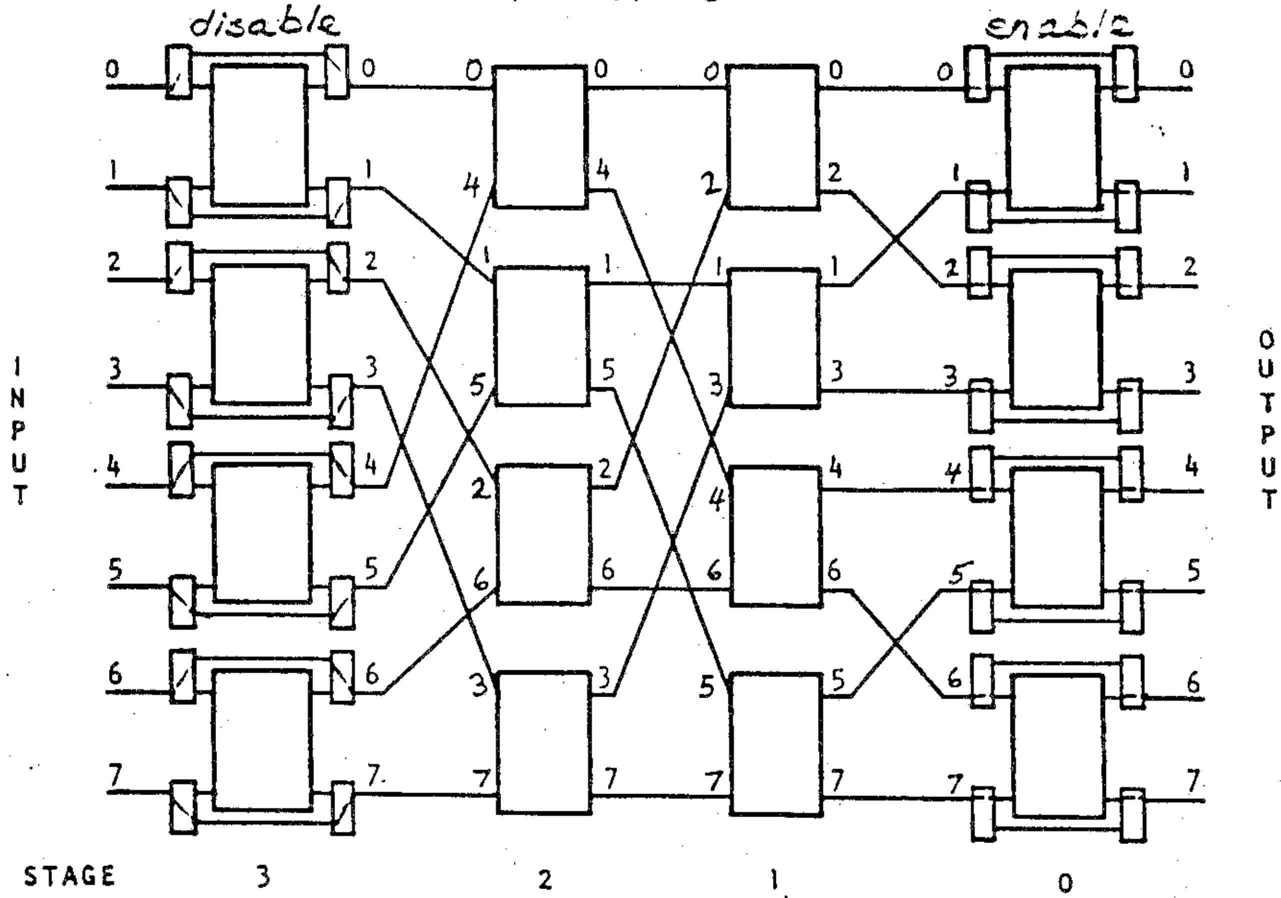


interchange box disabled

- stage m normally disabled
- stage 0 normally enabled

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



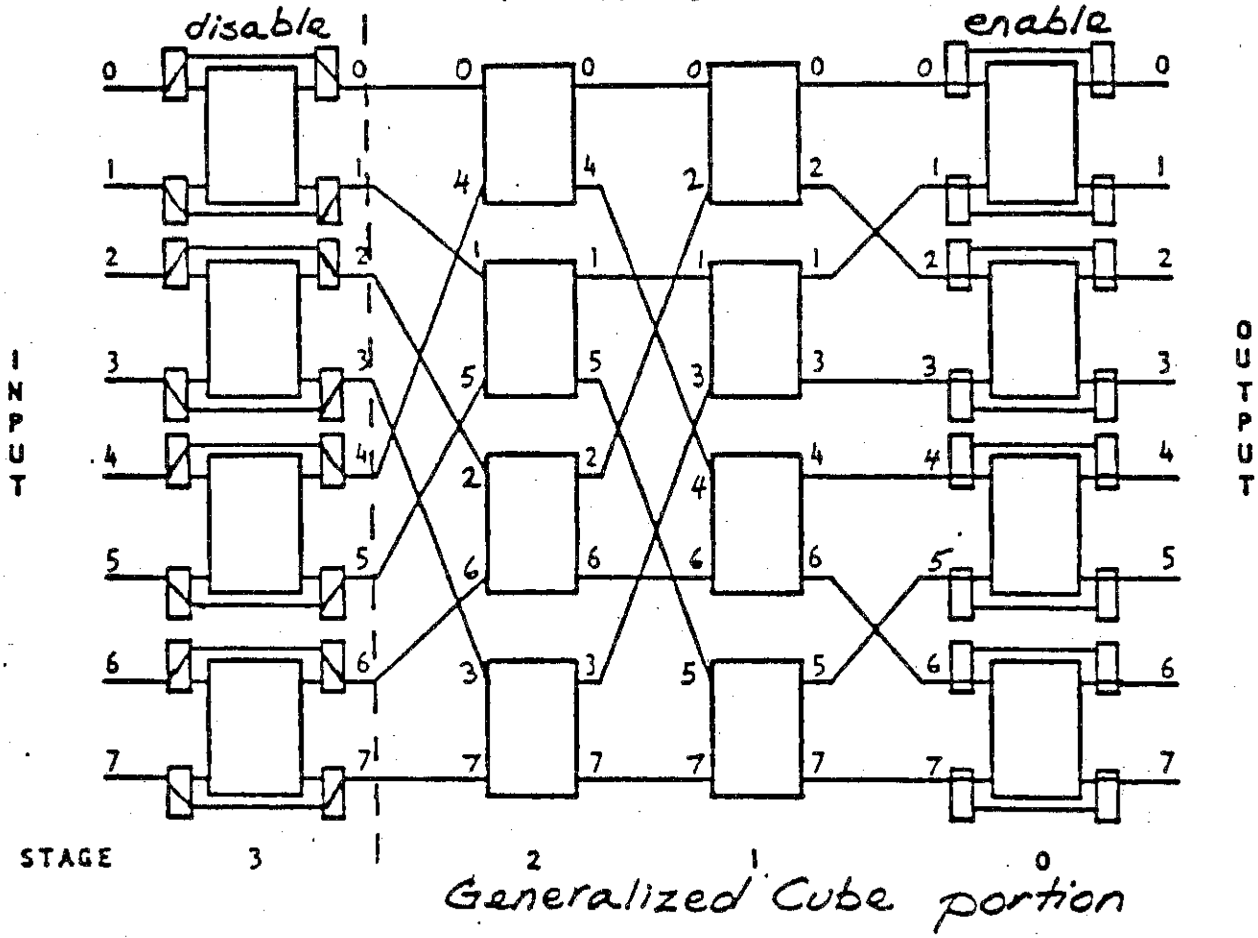
- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

NO FAULTS

STAGE
$m = \log_2 N$
DISABLED

STAGE 0
ENABLED

THE EXTRA STAGE CUBE NETWORK FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

NO FAULTS

STAGE
$m = \log_2 N$
DISABLED

STAGE 0
ENABLED

JUST LIKE GENERALIZED CUBE

Fault Model

- I/O ports and bypass circuits assumed fault-free
- data not passed through a faulty link or interchange box

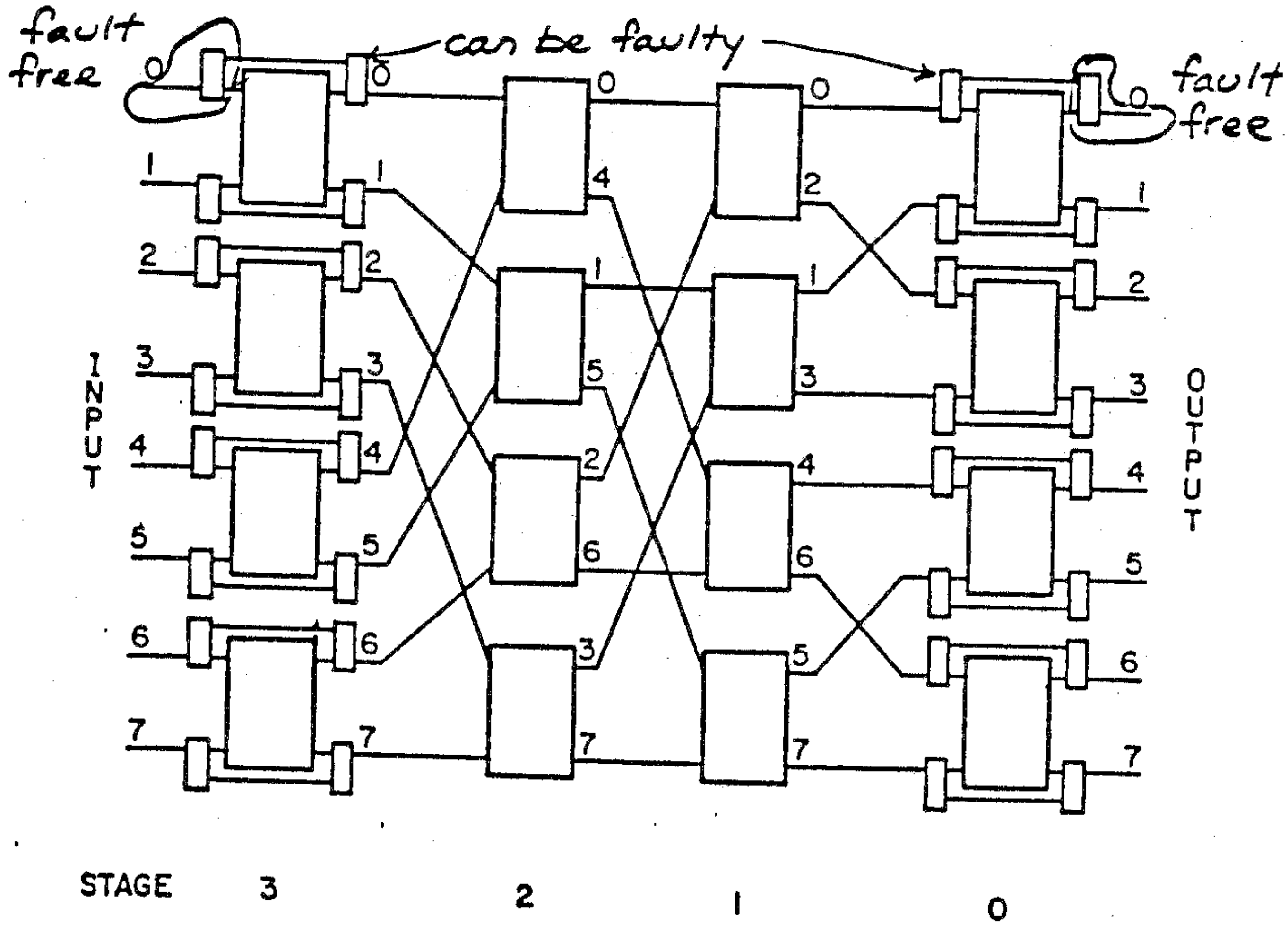
"stuck at" faults may be problem.

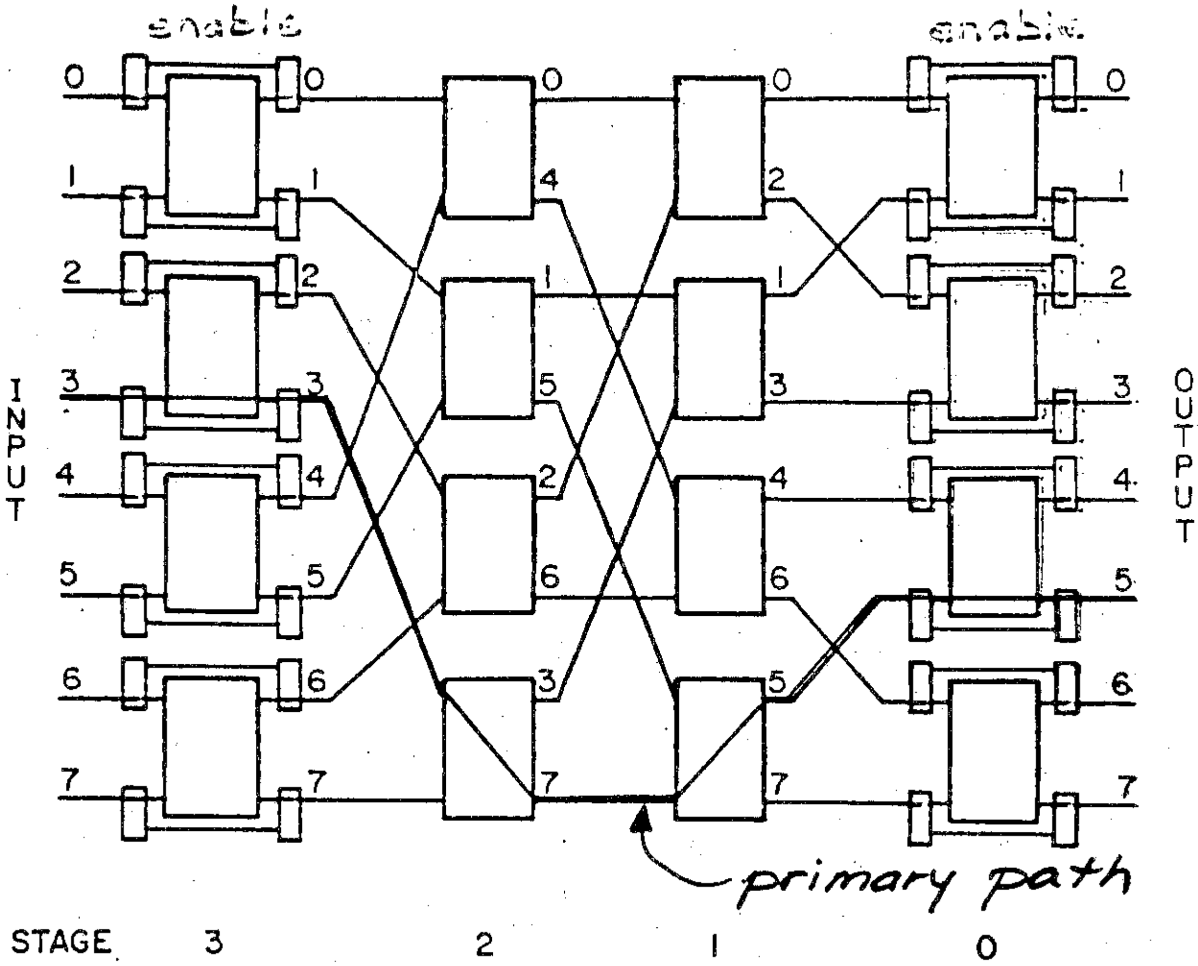
Fault Detection and Location

- test patterns
- dynamic parity checking

Concern

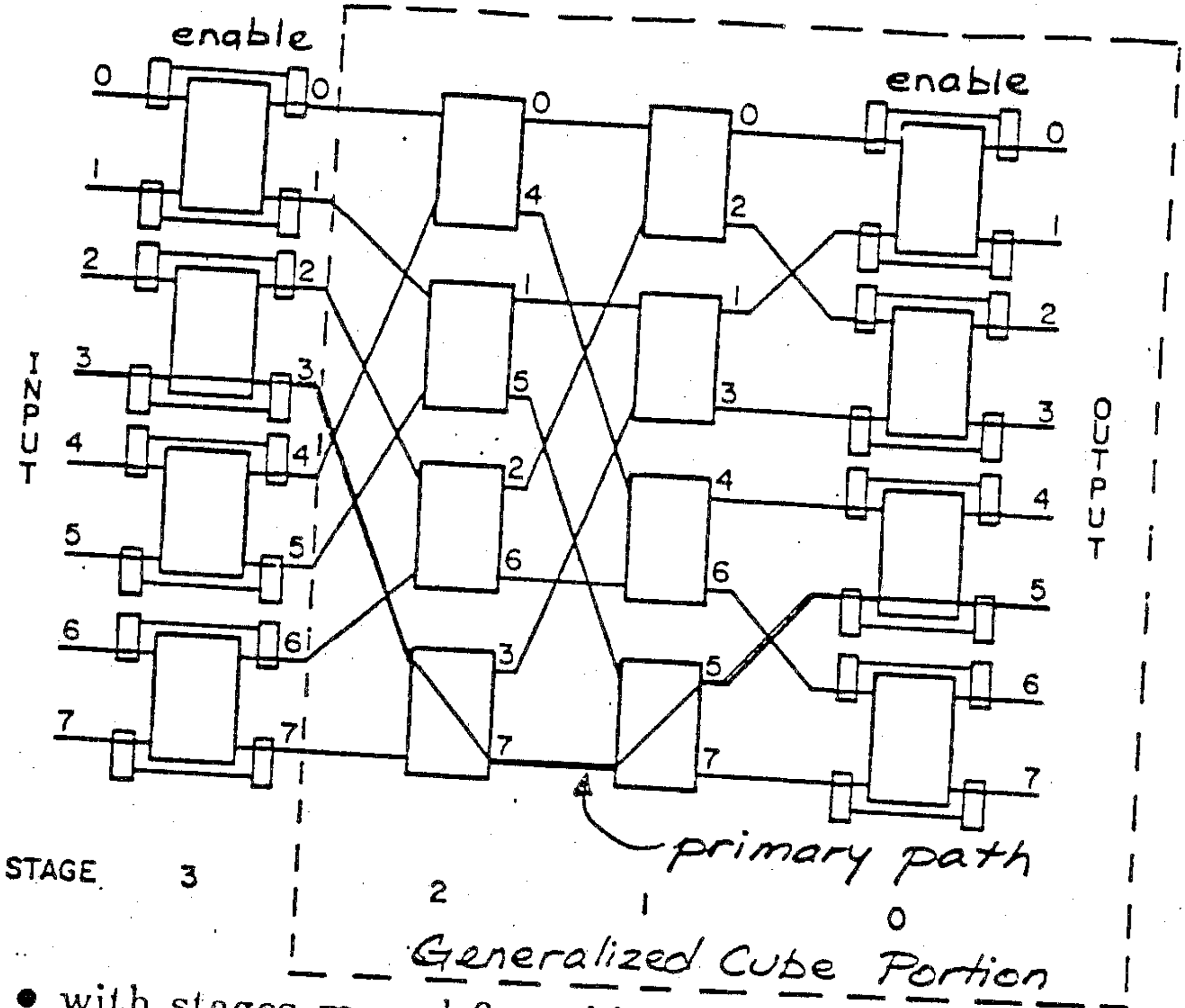
- recovery once fault is located





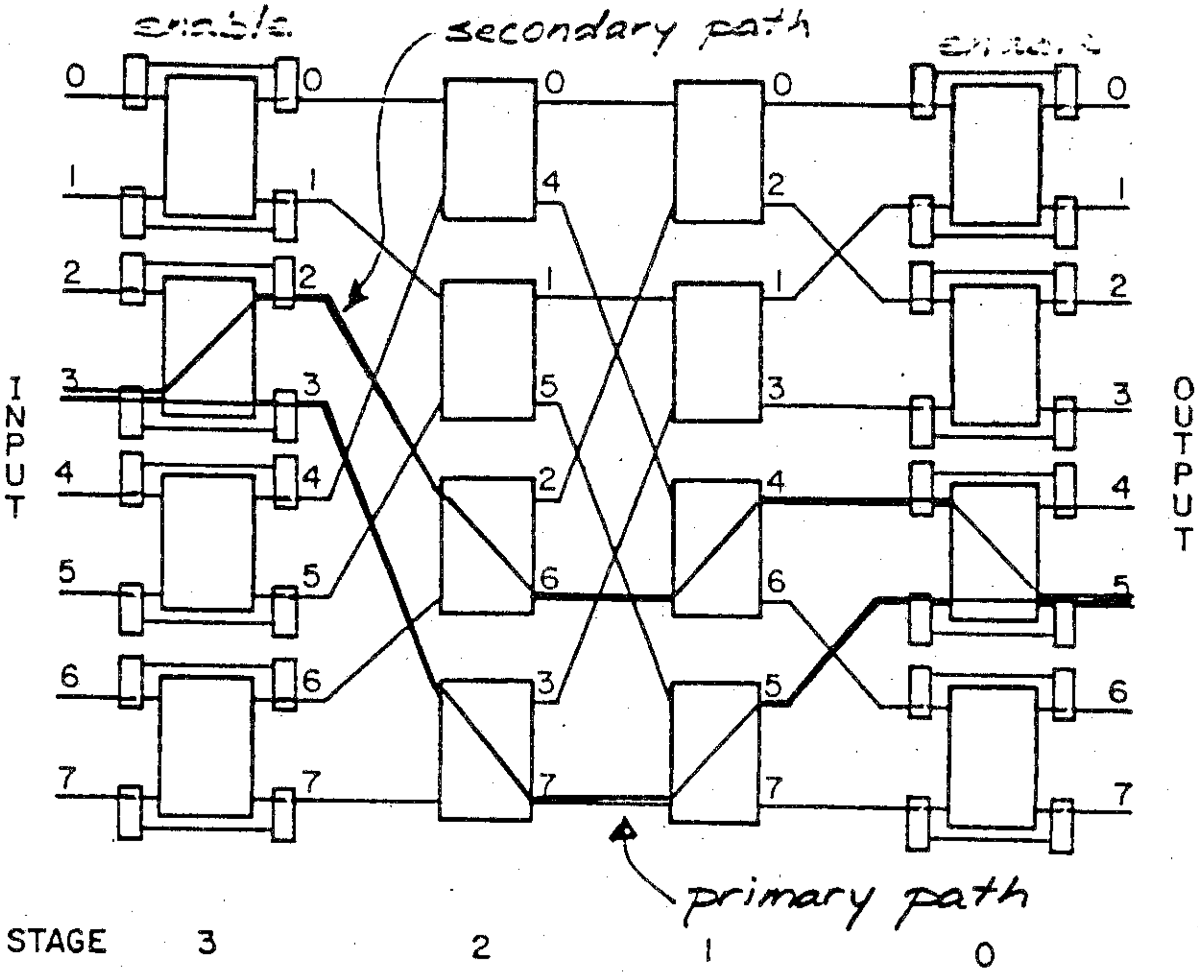
- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

primary path - use if not faulty
(same as Generalized Cube)



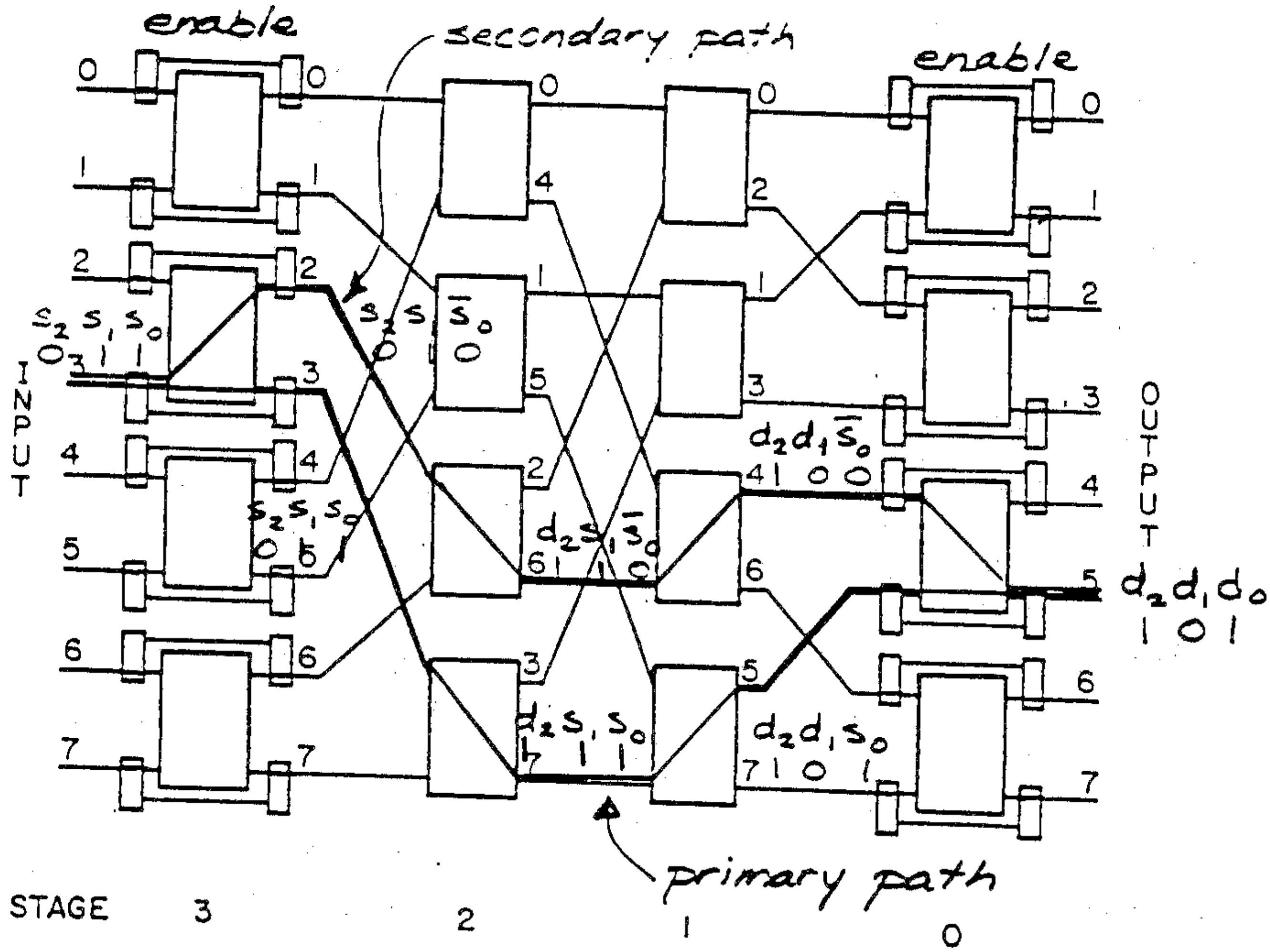
- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

primary path - use if not faulty
(same as Generalized Cube)



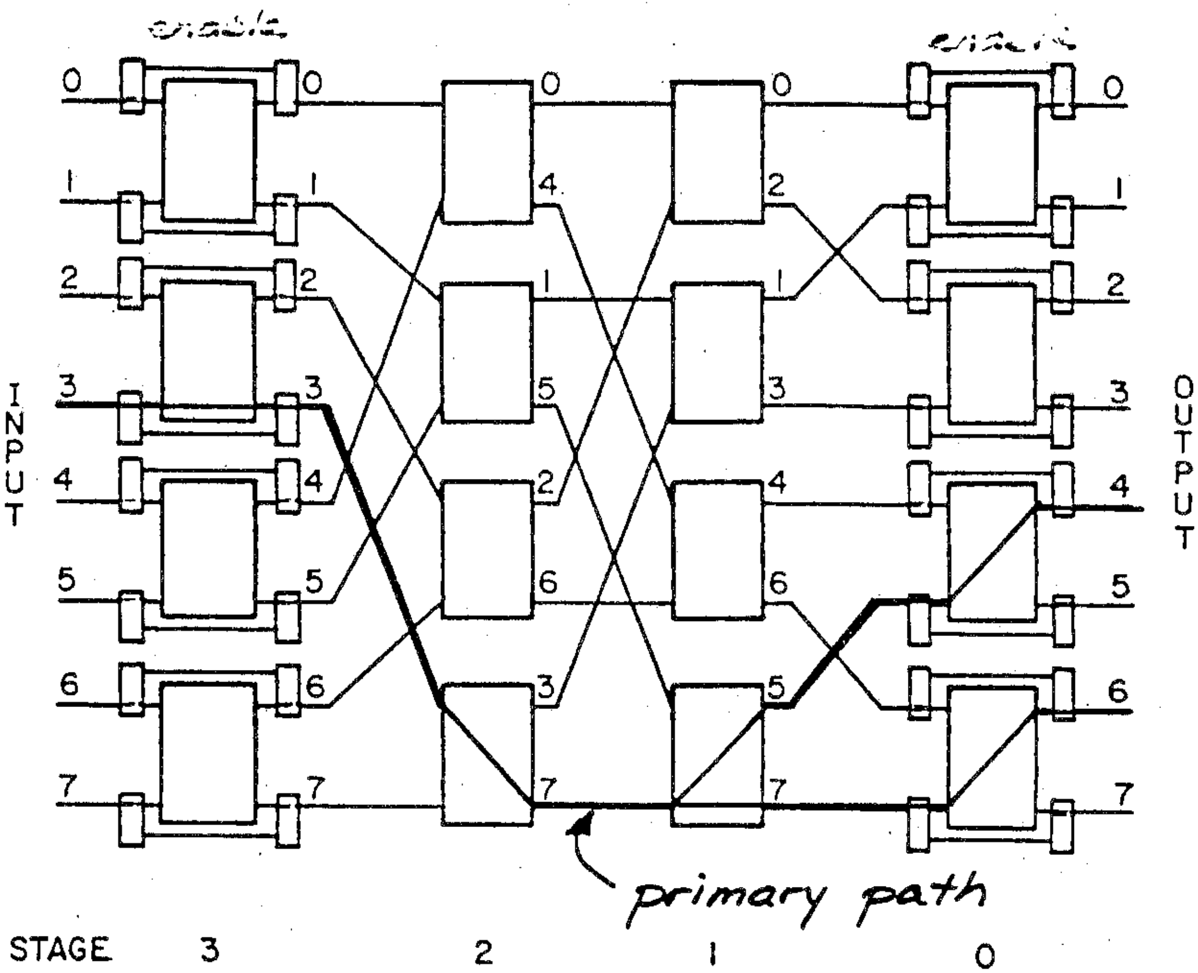
- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

primary path - use if not faulty
(same as Generalized Cube)
secondary path - use if primary path has fault



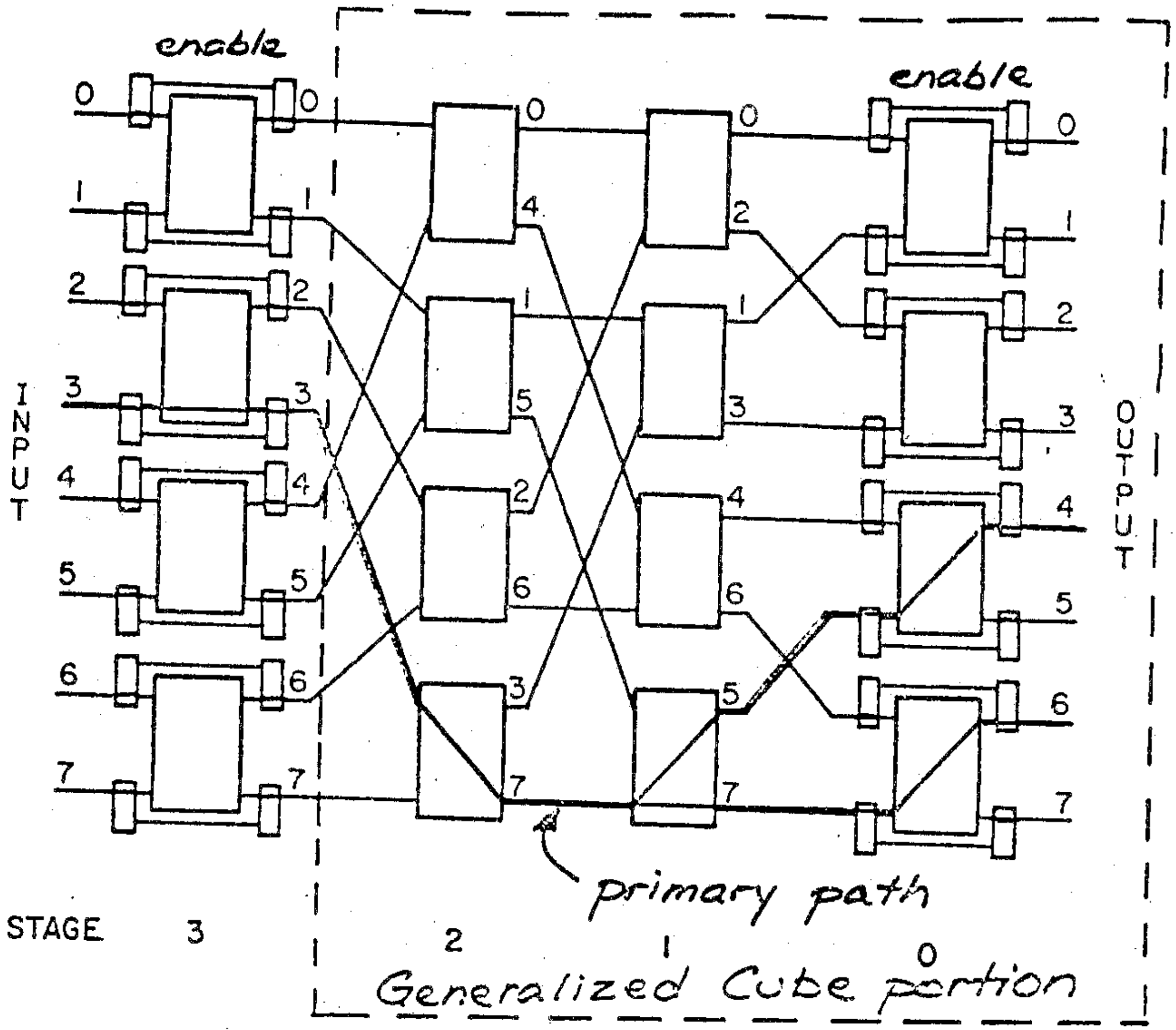
- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

primary path - use if not faulty (same as Generalized Cube)
secondary path - use if primary path has fault



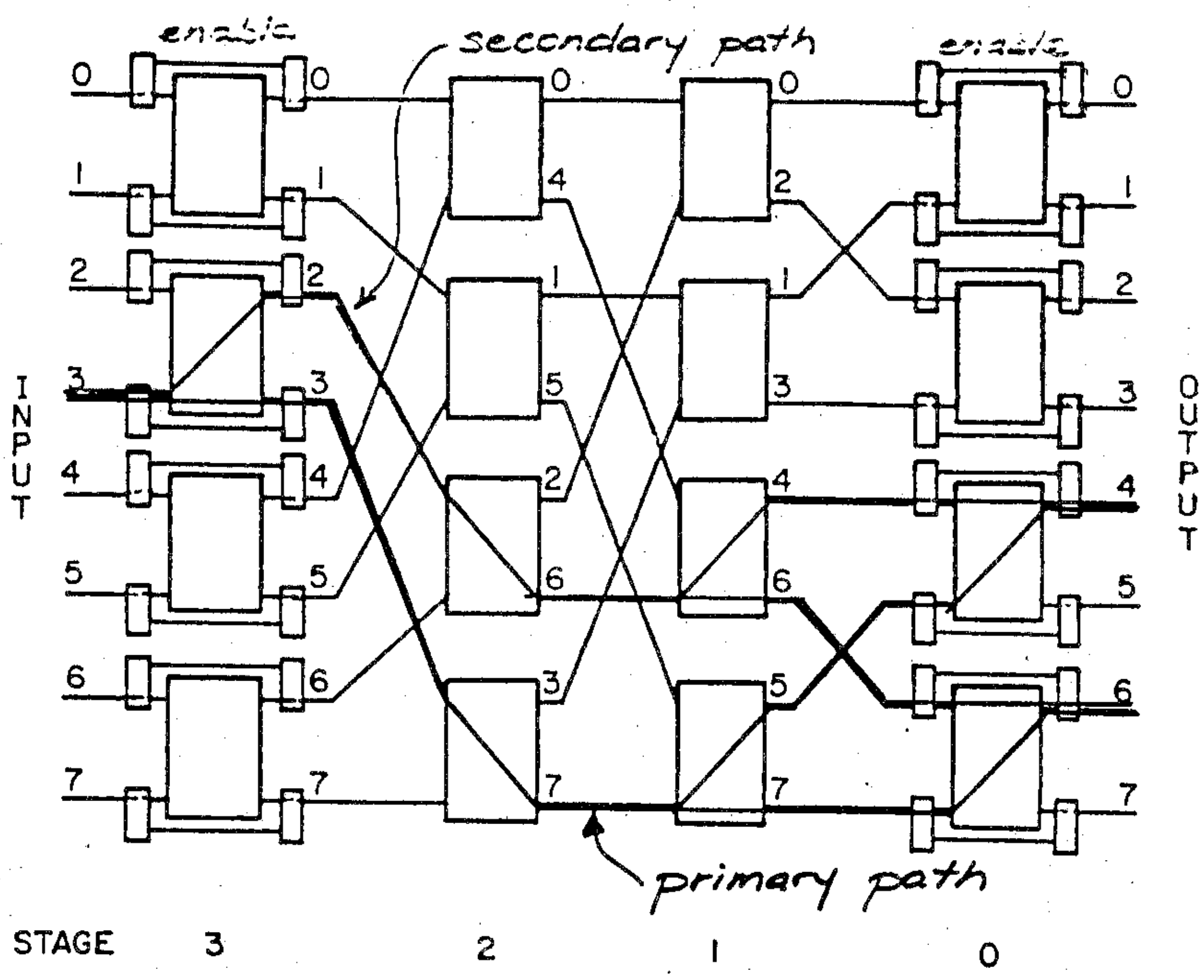
- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

broadcast from 3 to 4 and 6
primary path - use if not faulty
(same as Generalized Cube)



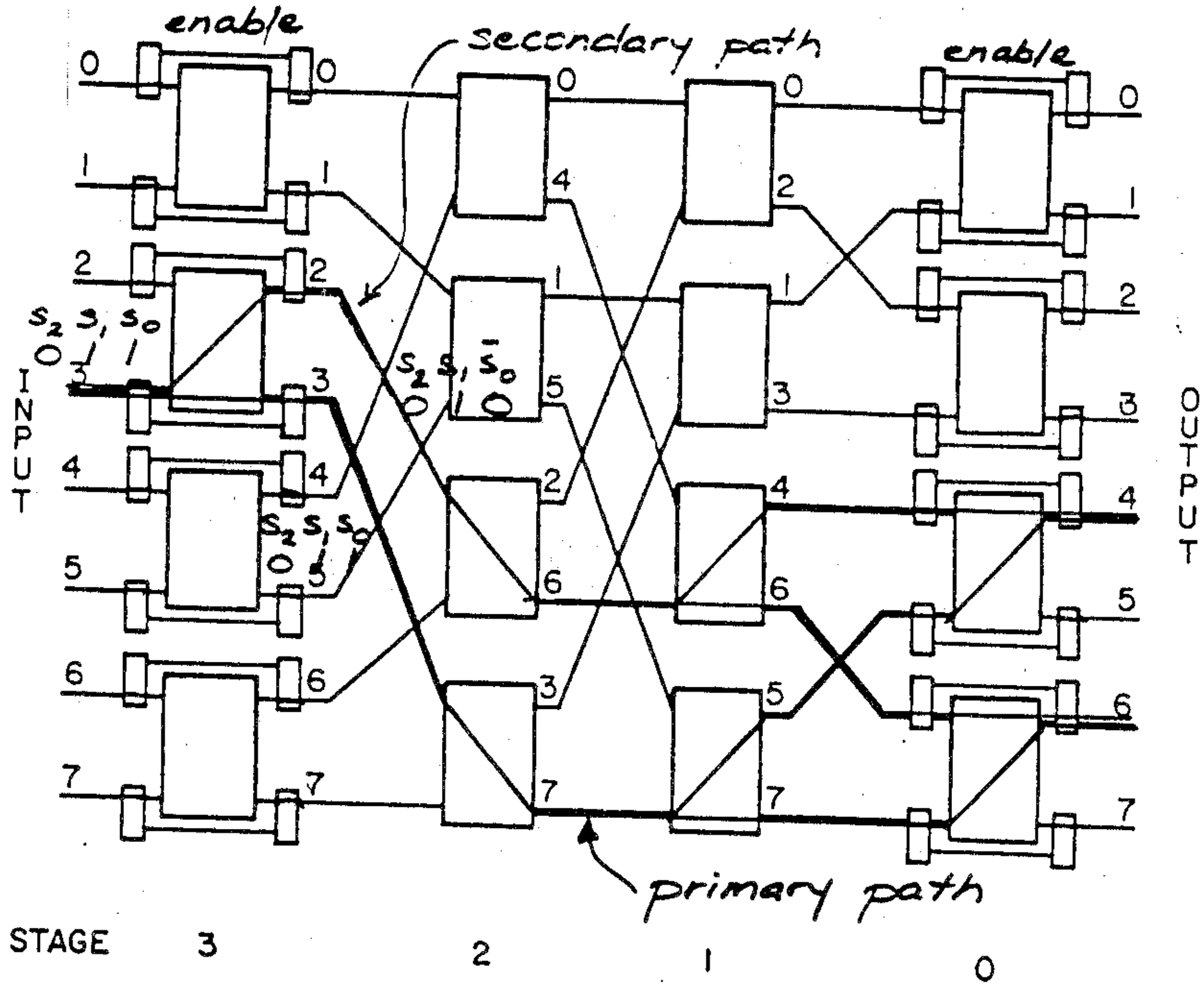
- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

broadcast from 3 to 4 and 6
primary path - use if not faulty
(same as Generalized Cube)



- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

broadcast path: from 3 to 4 and 6
primary path - use if not faulty
(same as Generalized Cube)
secondary path - use if primary path faulty

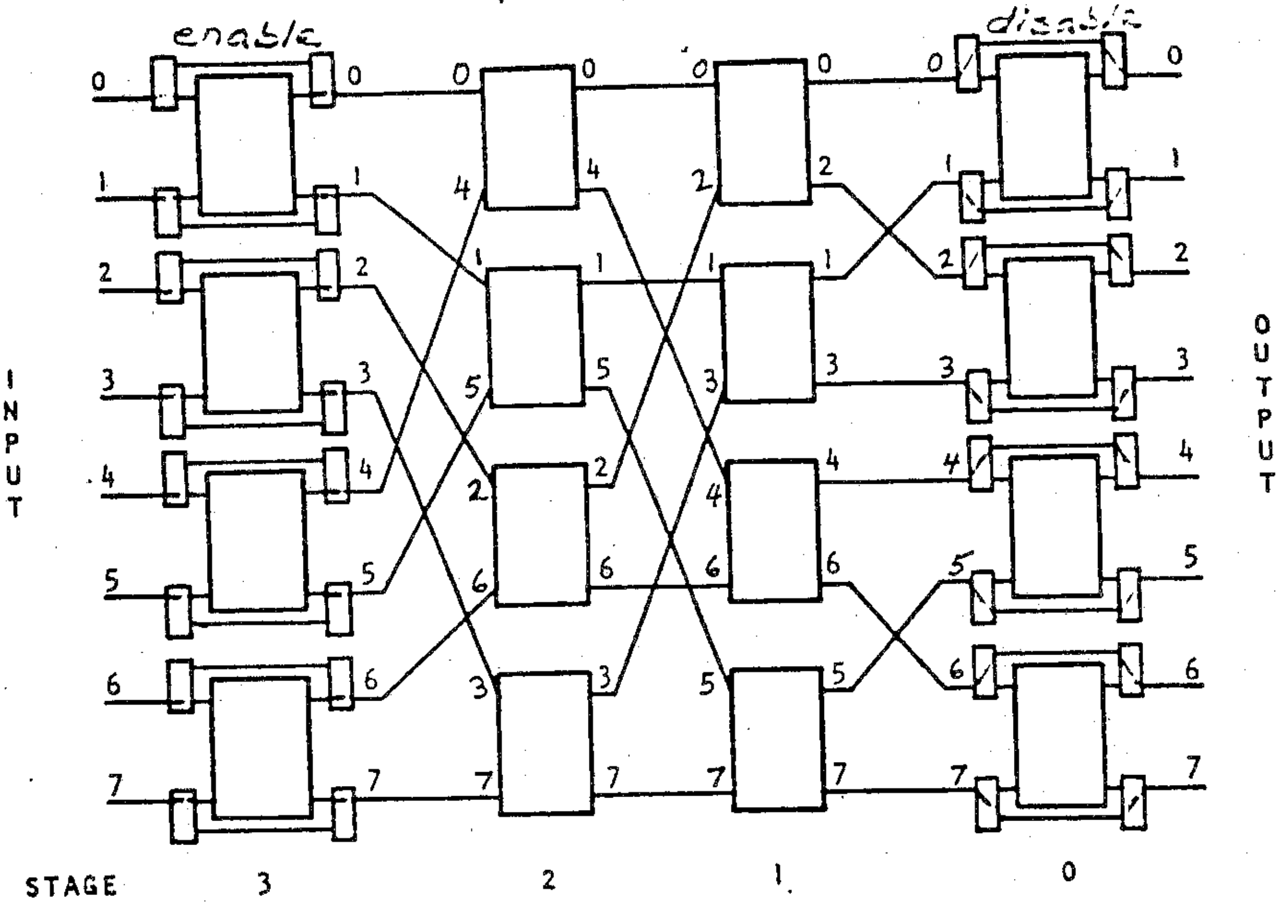


- with stages m and 0 enabled there exist two paths between any source and any destination
- the two paths have no links in common
- excluding stages m and 0, the paths have no boxes in common
- with a single fault there exists at least one fault-free path between any source and destination

broadcast path from 3 to 4 and 6
primary path - use if not faulty
(same as Generalized Cube)
secondary path - use if primary path faulty

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

STAGE 0 BOX FAULT

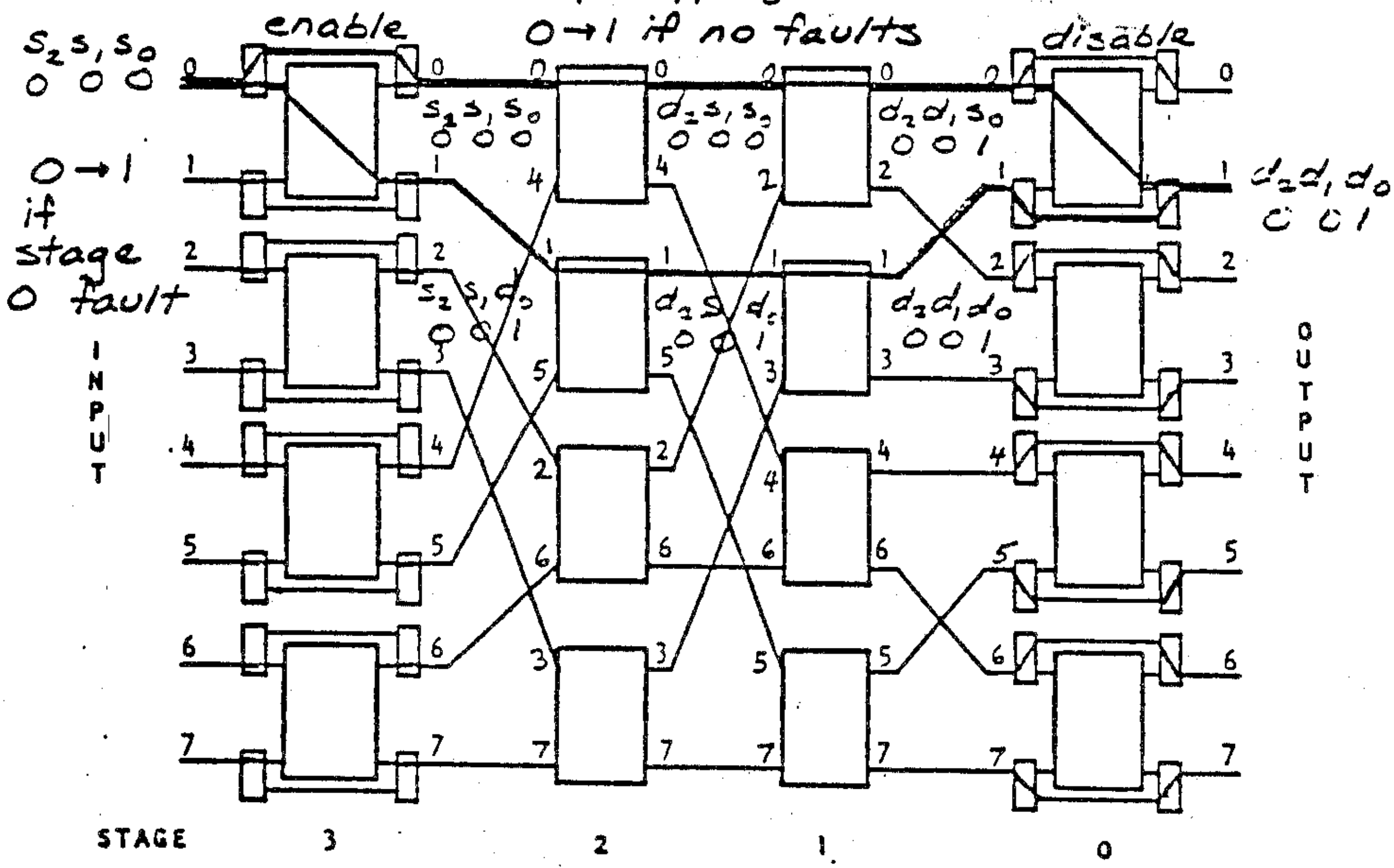
use STAGE $m = \log_2 N$ instead

ENABLE
STAGE m

DISABLE
STAGE 0

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

STAGE 0 BOX FAULT

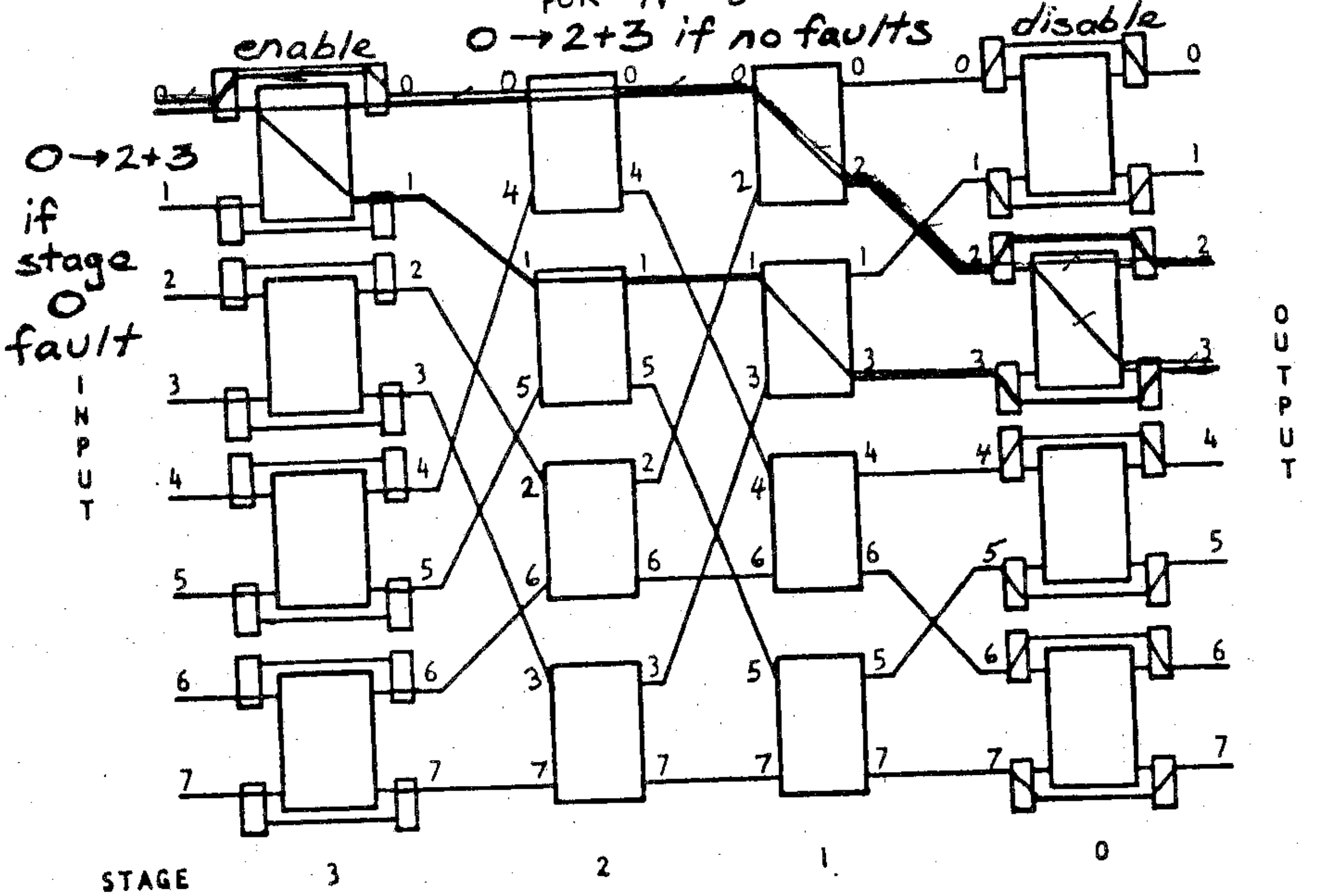
use STAGE $m = \log_2 N$ instead

ENABLE STAGE m

DISABLE STAGE 0

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

STAGE 0 BOX FAULT

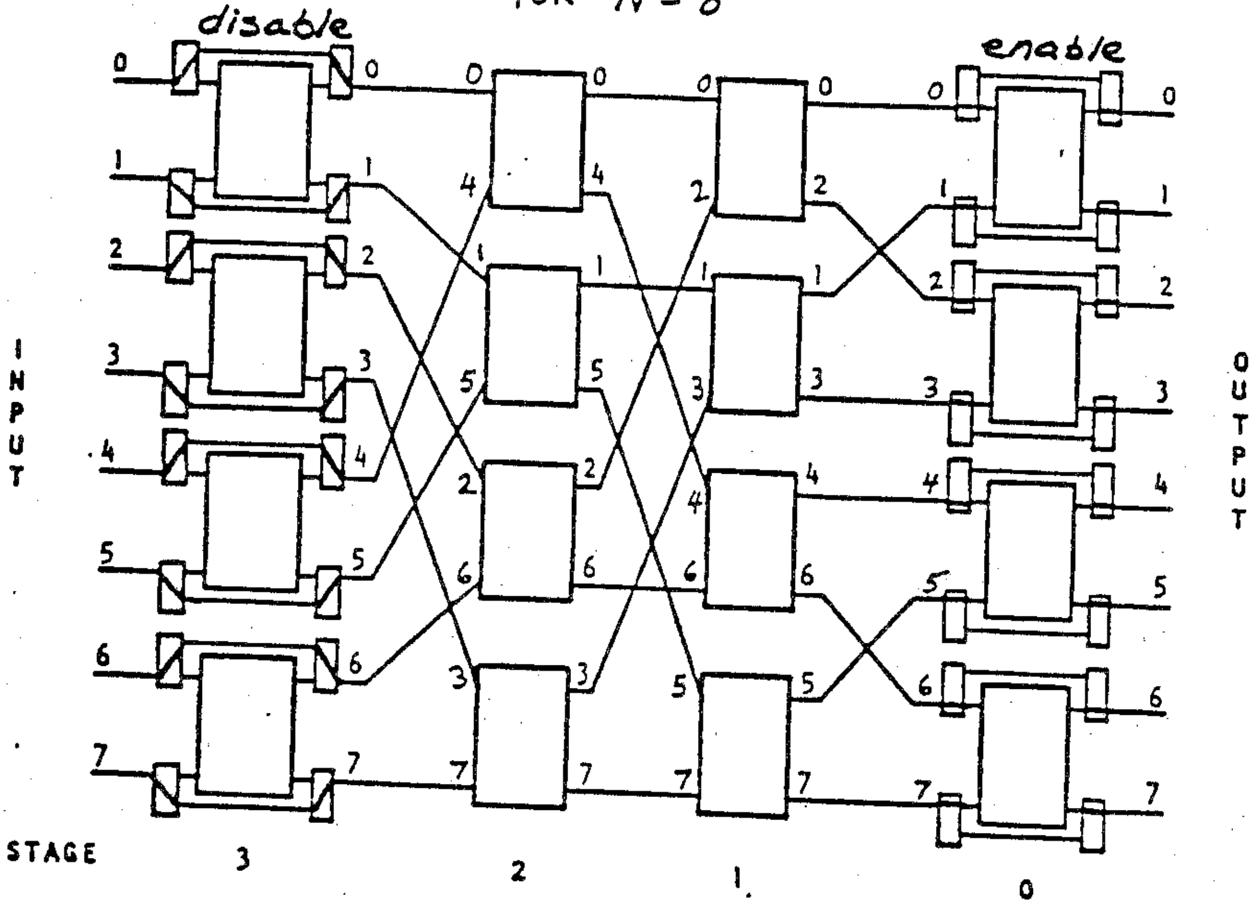
use STAGE $m = \log_2 N$ instead

ENABLE STAGE m

DISABLE STAGE 0

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= $\log_2 N$), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE $\log_2 N$ OR 0

STAGE m BOX FAULT

DISABLE STAGE m

ENABLE STAGE 0

JUST LIKE GENERALIZED CUBE

Permuting with the ESC

Permuting:

routing all N inputs to the N outputs simultaneously

No Faults:

ESC can perform in one pass

all Generalized Cube performable permutations

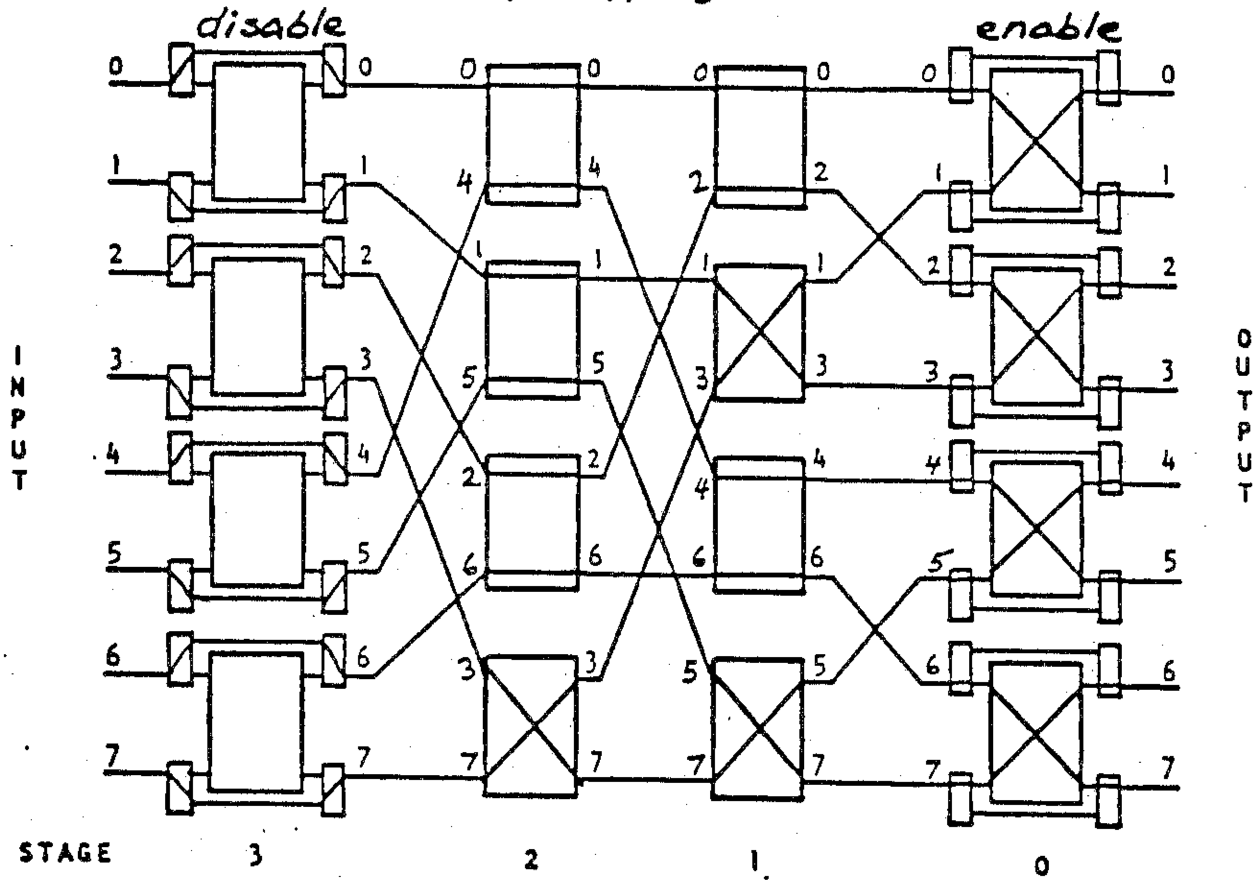
Single Fault:

ESC can perform in at most two passes

all Generalized Cube performable permutations

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



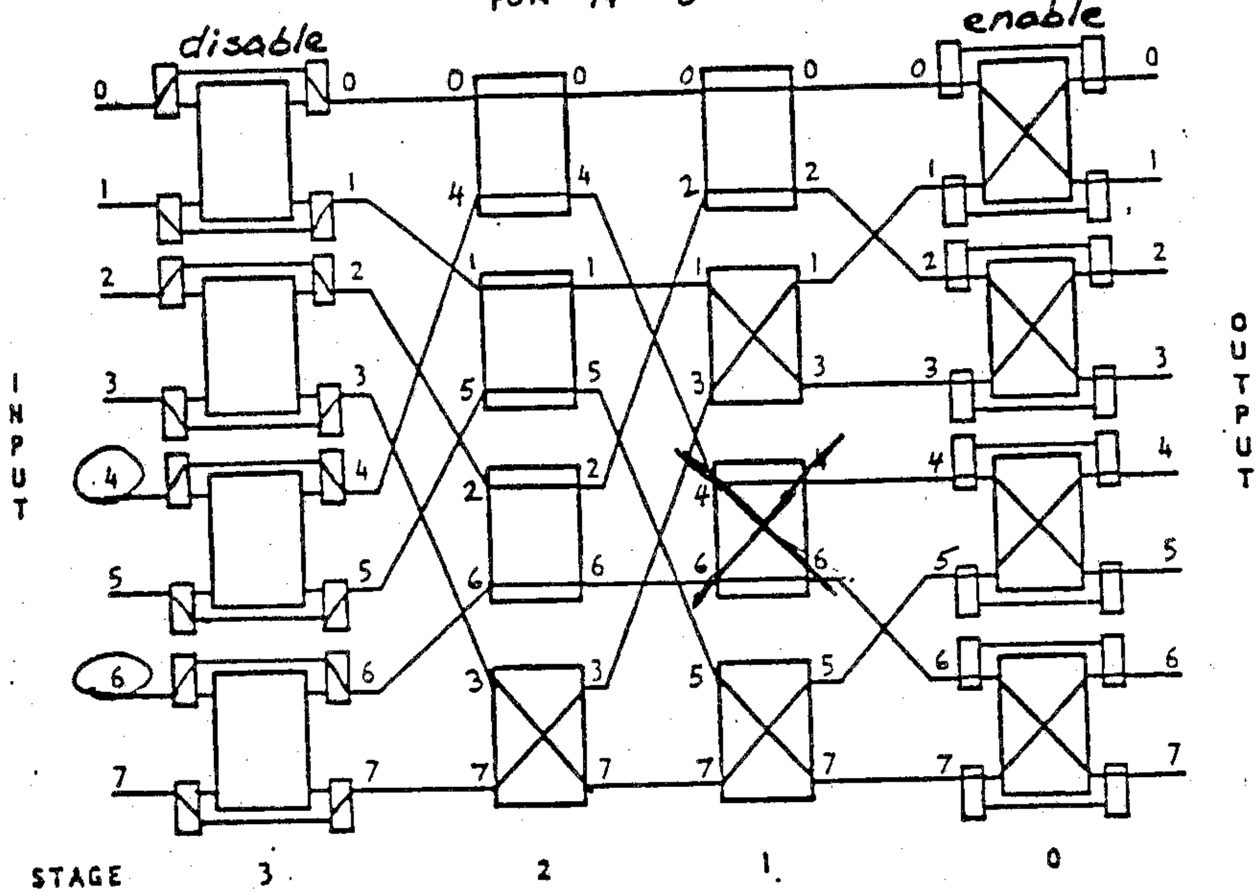
- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

INPUT I TO OUTPUT I + 1
(MOD N)

NO FAULTS

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= $\log_2 N$), IS A COPY OF STAGE 0

- INCLUDES CIRCUITRY TO BYPASS STAGE $\log_2 N$ OR 0

INPUT I TO OUTPUT I + 1 (MOD N)

FAULT

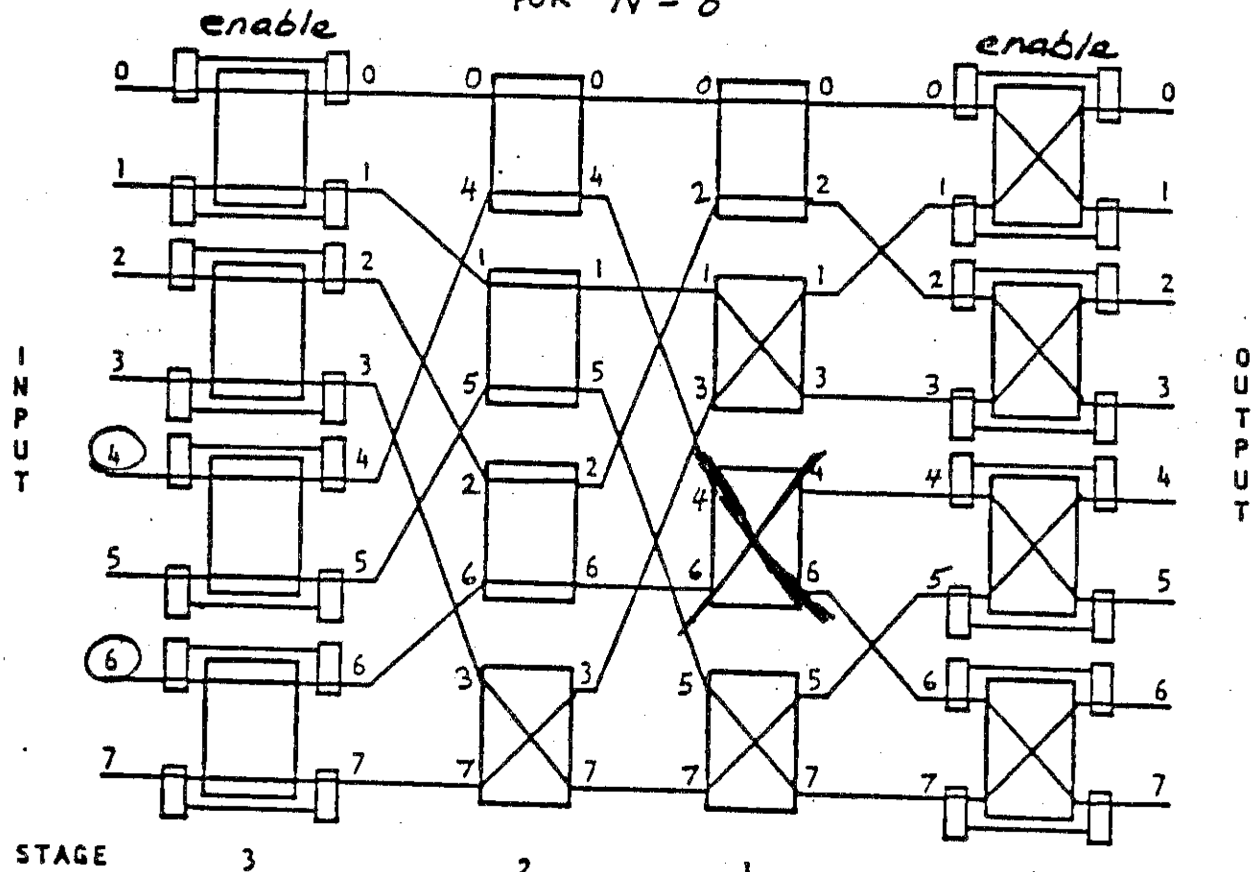
~~NO FAULTS~~

PASS 1: ALL WITH OK ~~PRIMARY~~ PATHS (ALL EXCEPT 4 + 6)

PASS 2: 4+6 USE SECONDARY PATHS

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



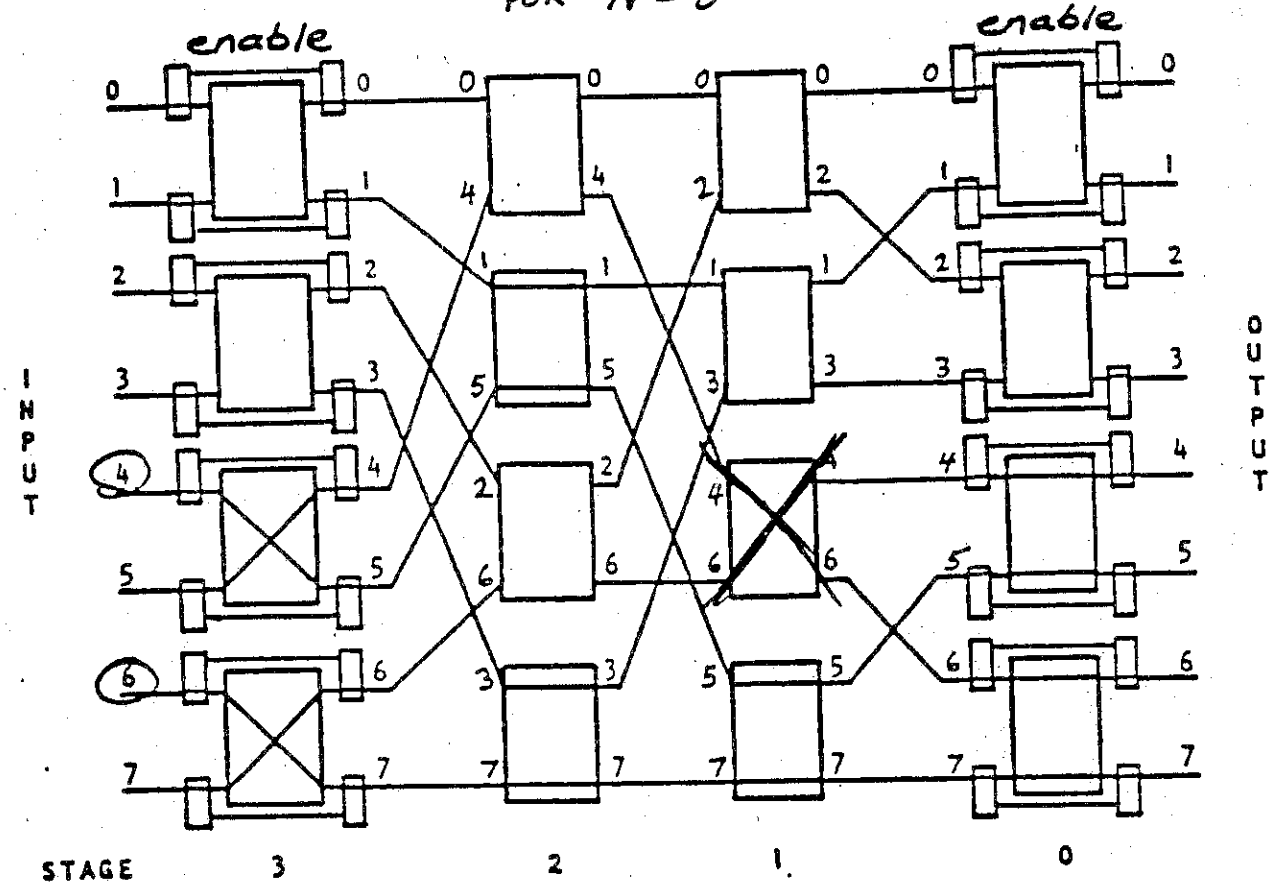
- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

PASS 1: ALL WITH GOOD PRIMARY PATHS
(ALL EXCEPT 4+6)

I TO I+1 MOD N

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= $\log_2 N$), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE $\log_2 N$ OR 0

PASS 2: 4 to 5

INPUTS 4+6

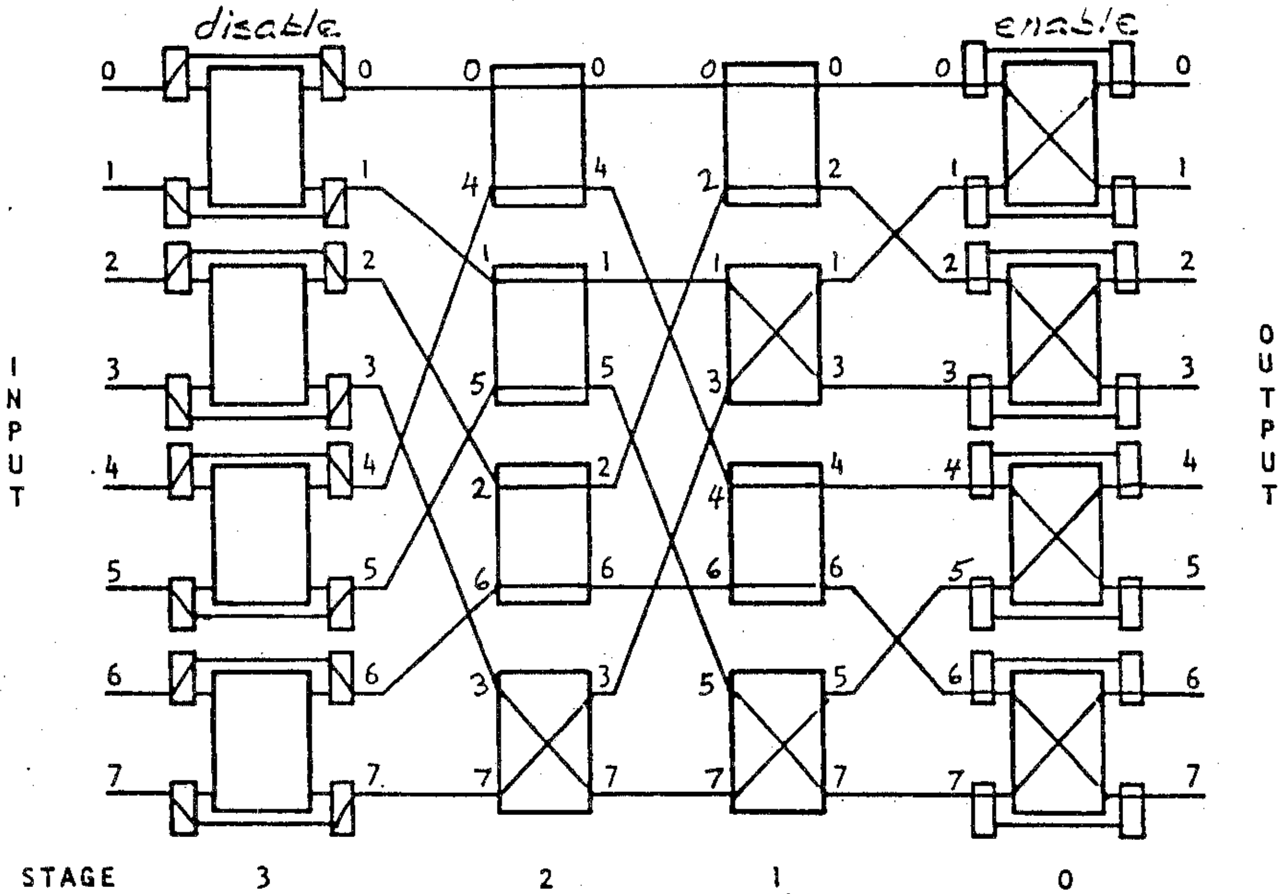
USE SECONDARY 6 to 7

PATHS

NO CONFLICTS

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



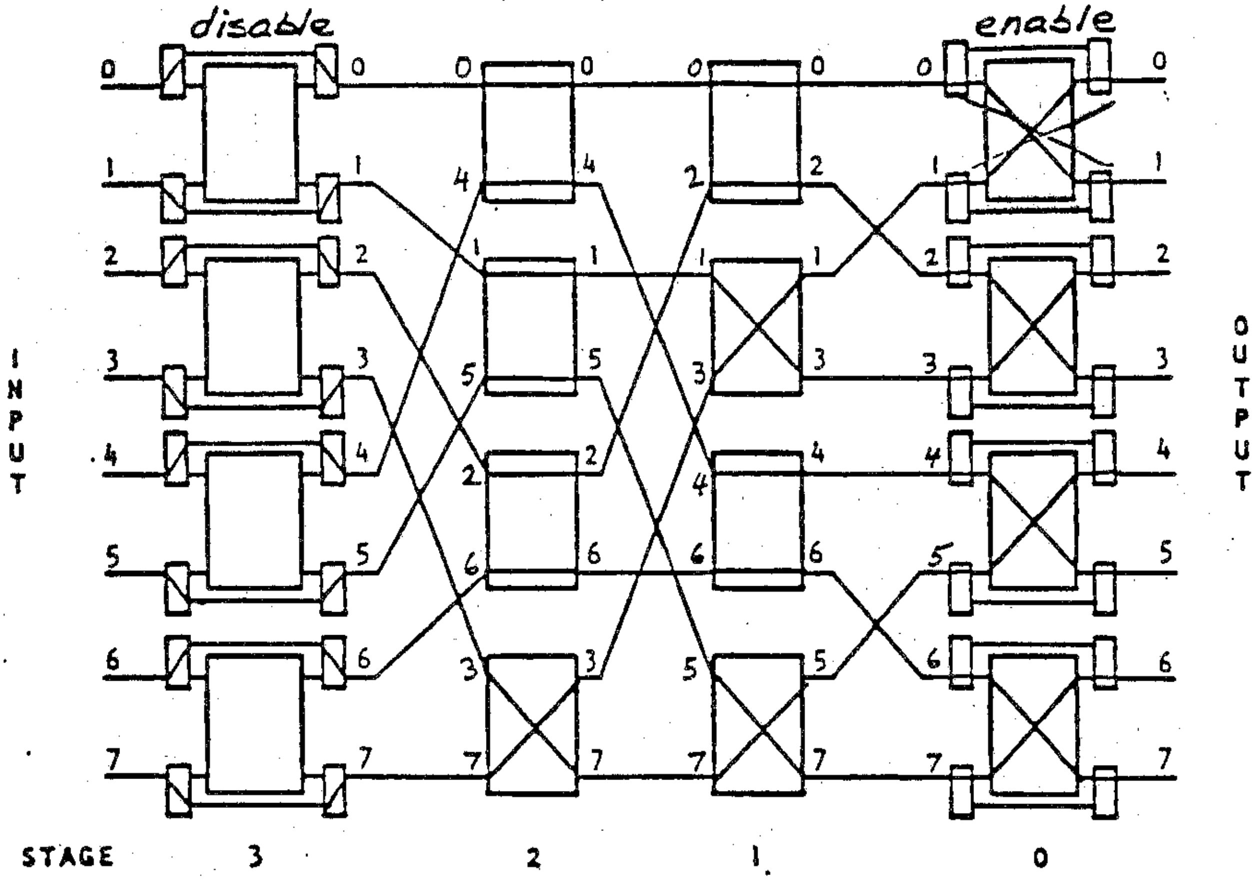
- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

INPUT I TO OUTPUT I+1 (MOD N)

<u>NO</u> FAULTS

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



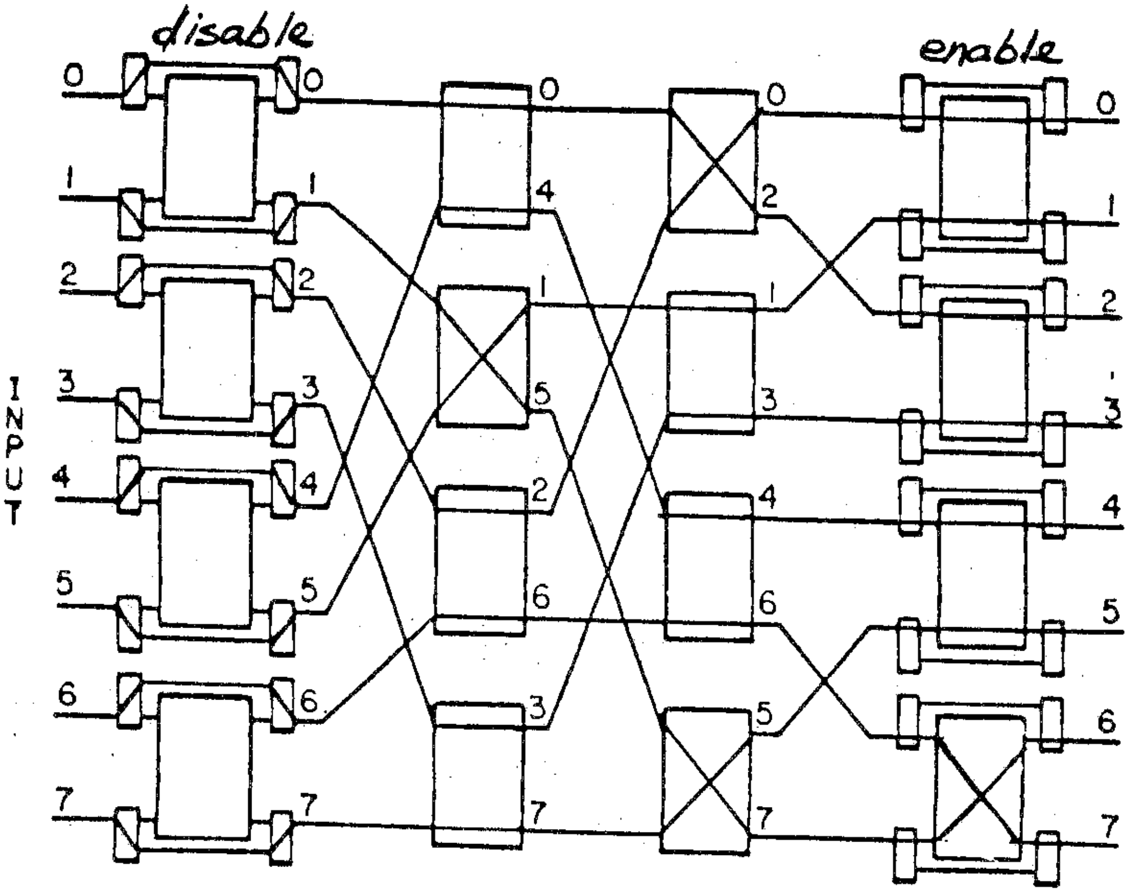
- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

INPUT I TO OUTPUT I+1 (MOD N)

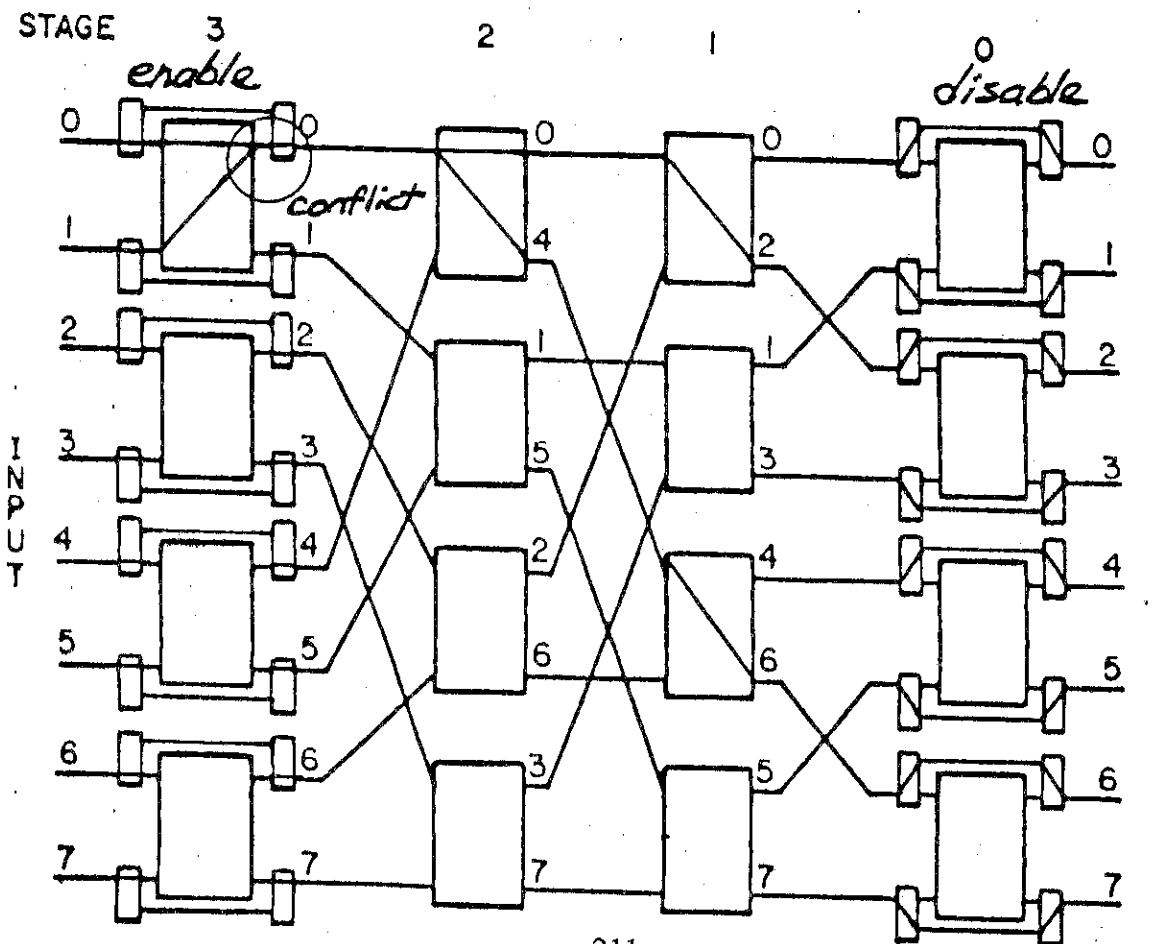
~~NO FAULTS~~

STAGE 0 FAULT
MAY NEED 2 PASSES

IN GENERAL, CAN NOT DO IN ONE PASS



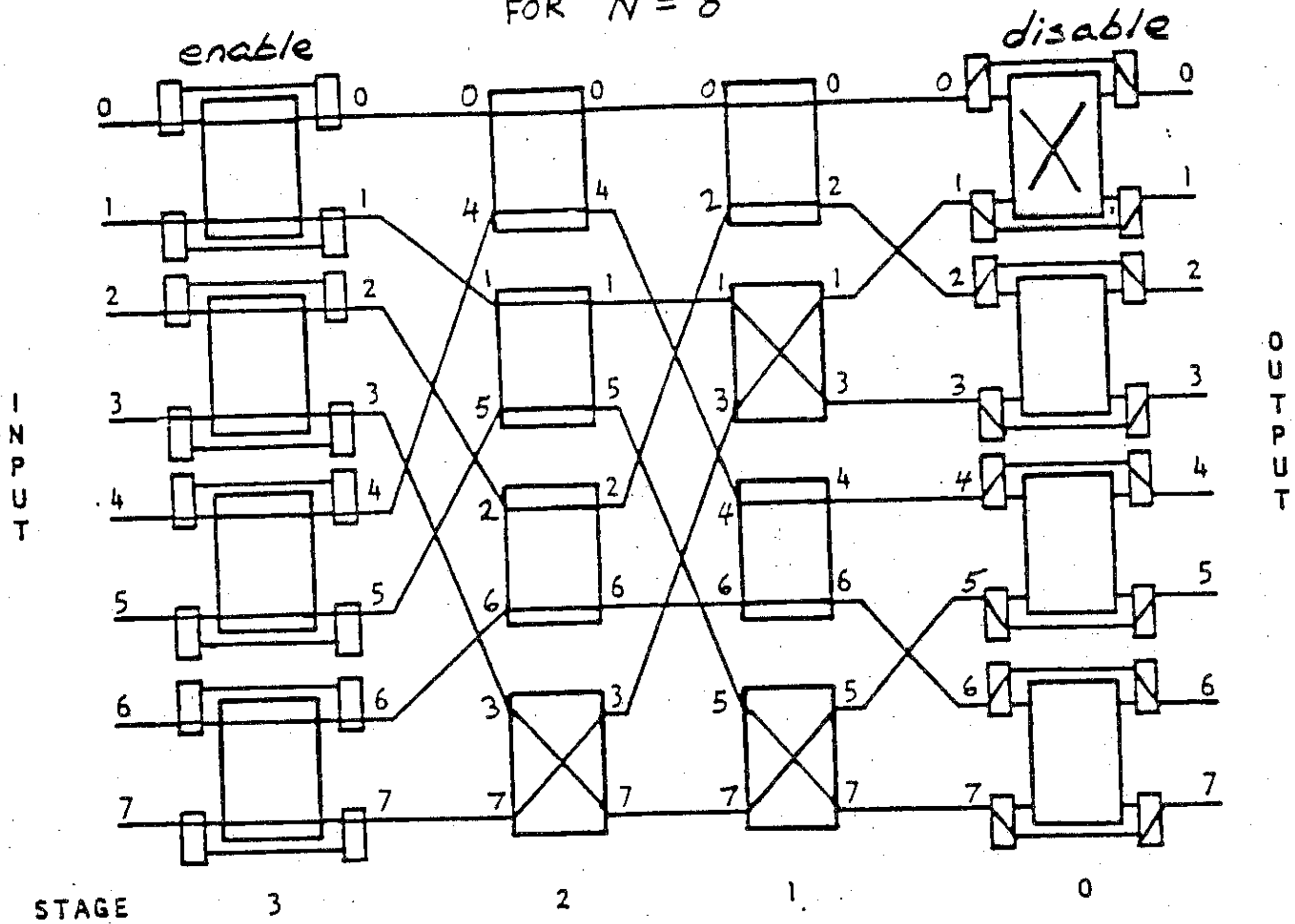
$0 \to 2$
$1 \to 6$
$2 \to 0$
$3 \to 3$
$4 \to 4$
$5 \to 1$
$6 \to 7$
$7 \to 5$



order of stages matter

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



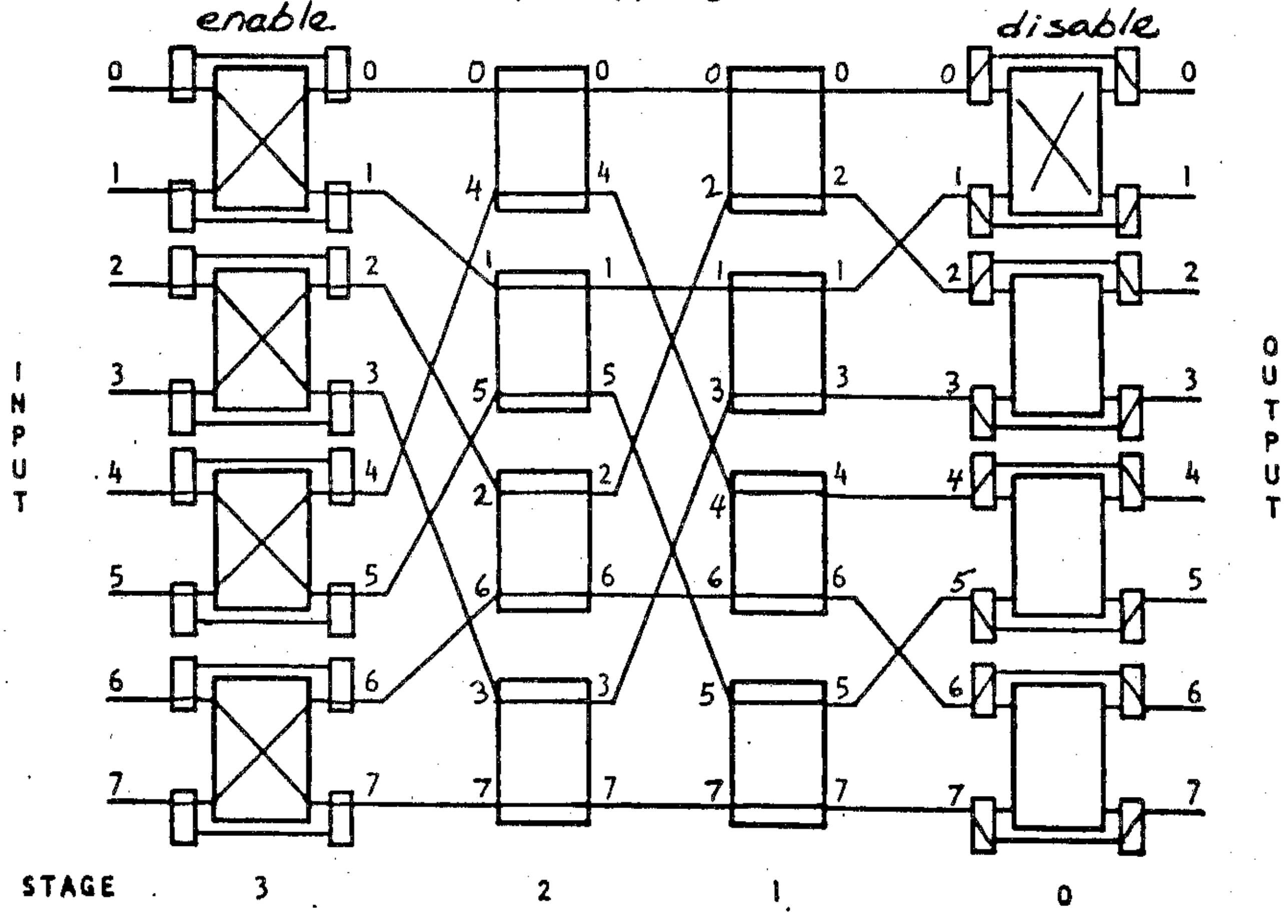
- EXTRA STAGE, STAGE 3 (= $\mathrm{LOG}_2\mathrm{N}$), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE $\mathrm{LOG}_2\mathrm{N}$ OR 0

PASS 1: DO STAGES 2 + 1

(ALL BUT 0)

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

PASS 2: DO STAGE 0
USING STAGE m

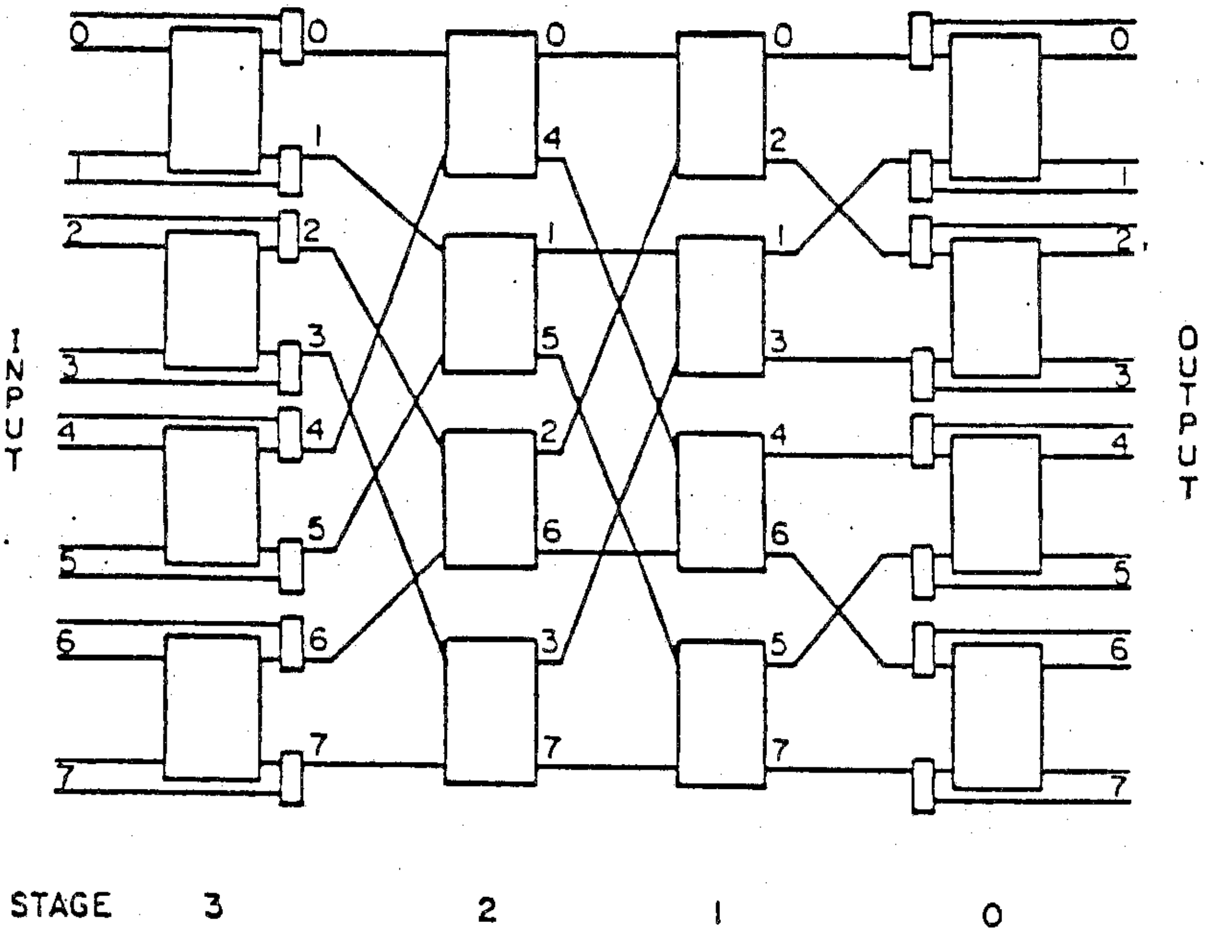
Elimination of fault-free hardware requirements

- ESC required input demuxes and output muxes to be fault free
- Design so there are two physical ports for each logical port to the network
- Single failure no longer denies access to the network

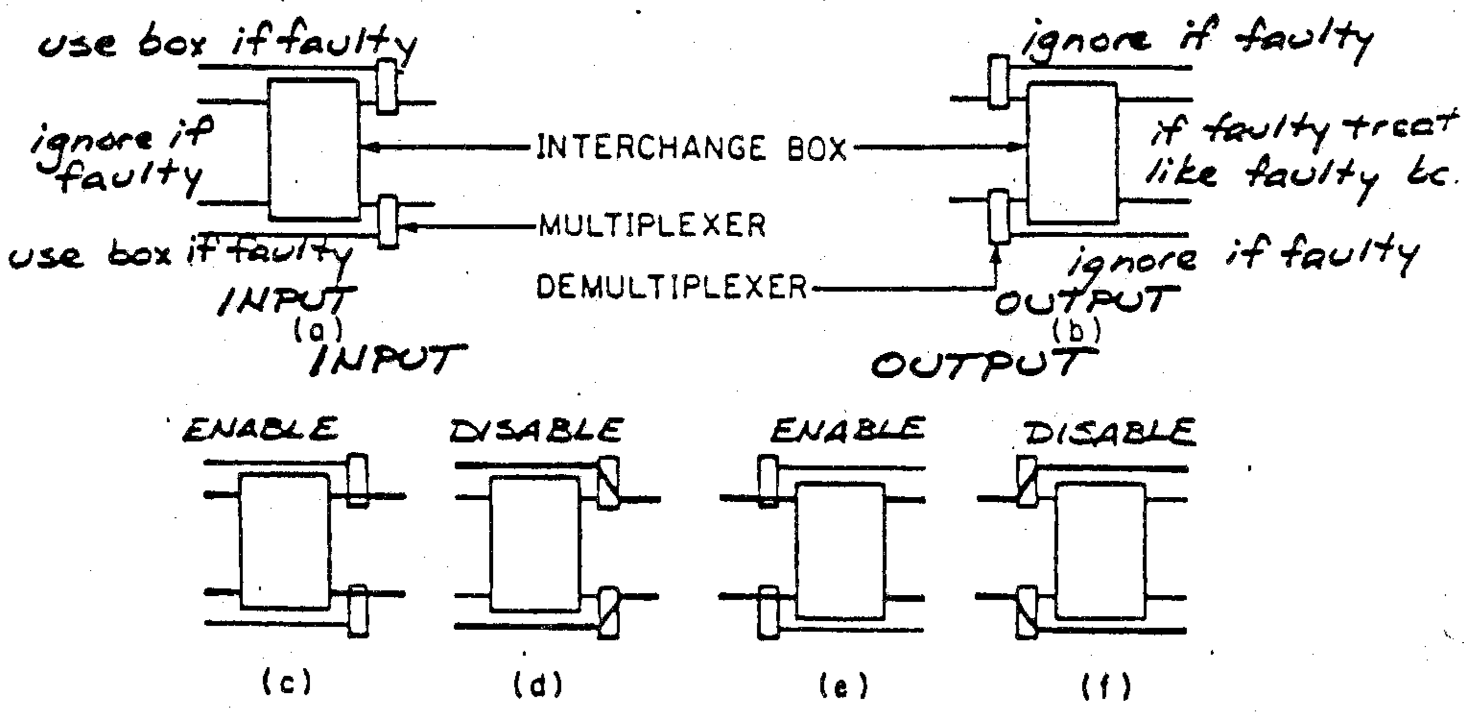
Fault Model

- ~~I/O ports and bypass circuits assumed fault-free~~

Dual I/O ports eliminate need for input DEMUX + output M



DUAL PORTS PROVIDE SINGLE FAULT TOLERANCE



If no fault (or stage m box fault)
    disable stage m, enable stage 0

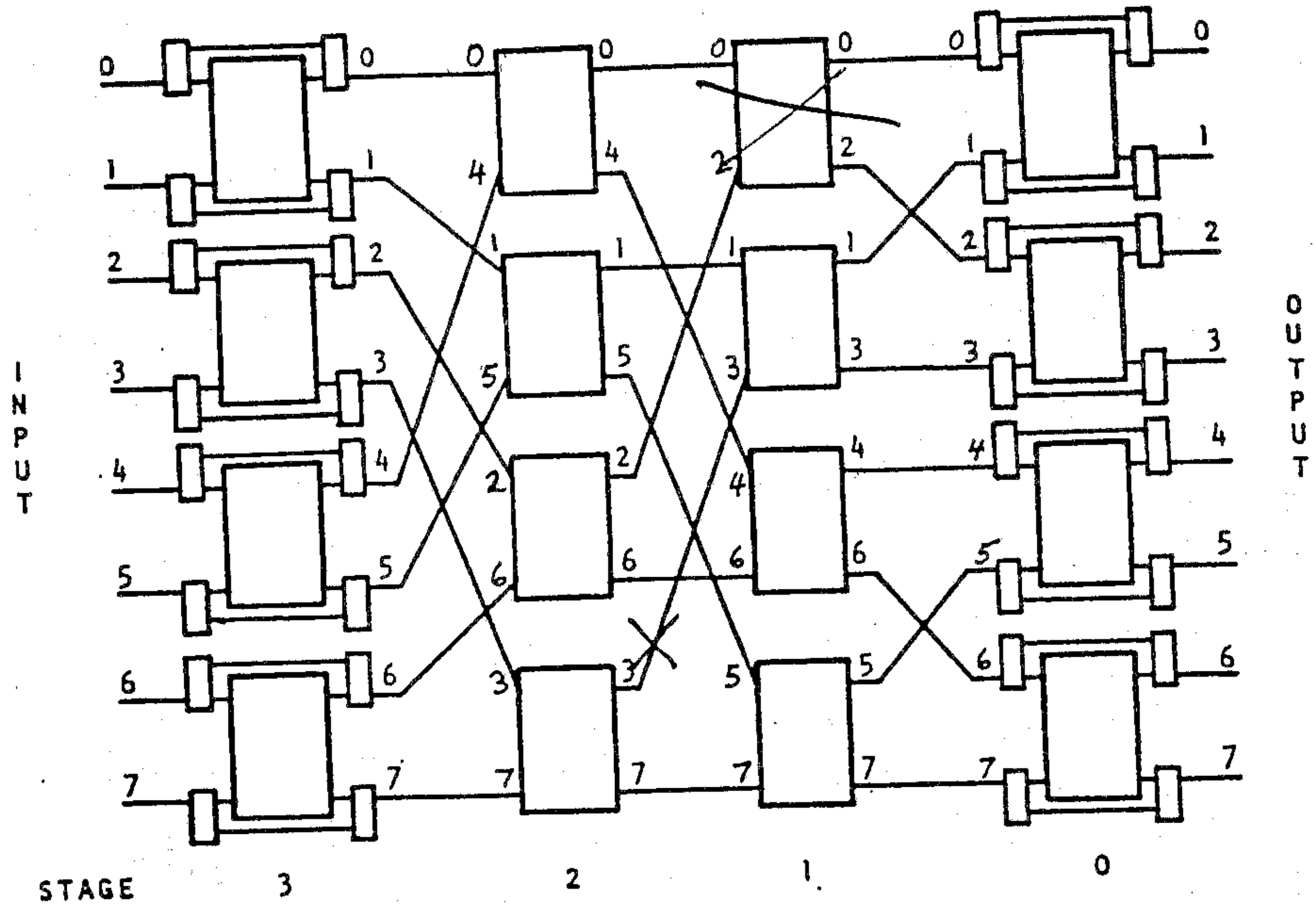
If stage 0 box fault
    enable stage m, disable stage 0

If stage i box fault, $1 \leq i < m$, or link fault
    use primary path if it does not include fault
    use secondary path if primary path includes fault

How is it determined if primary path includes fault?

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0

- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

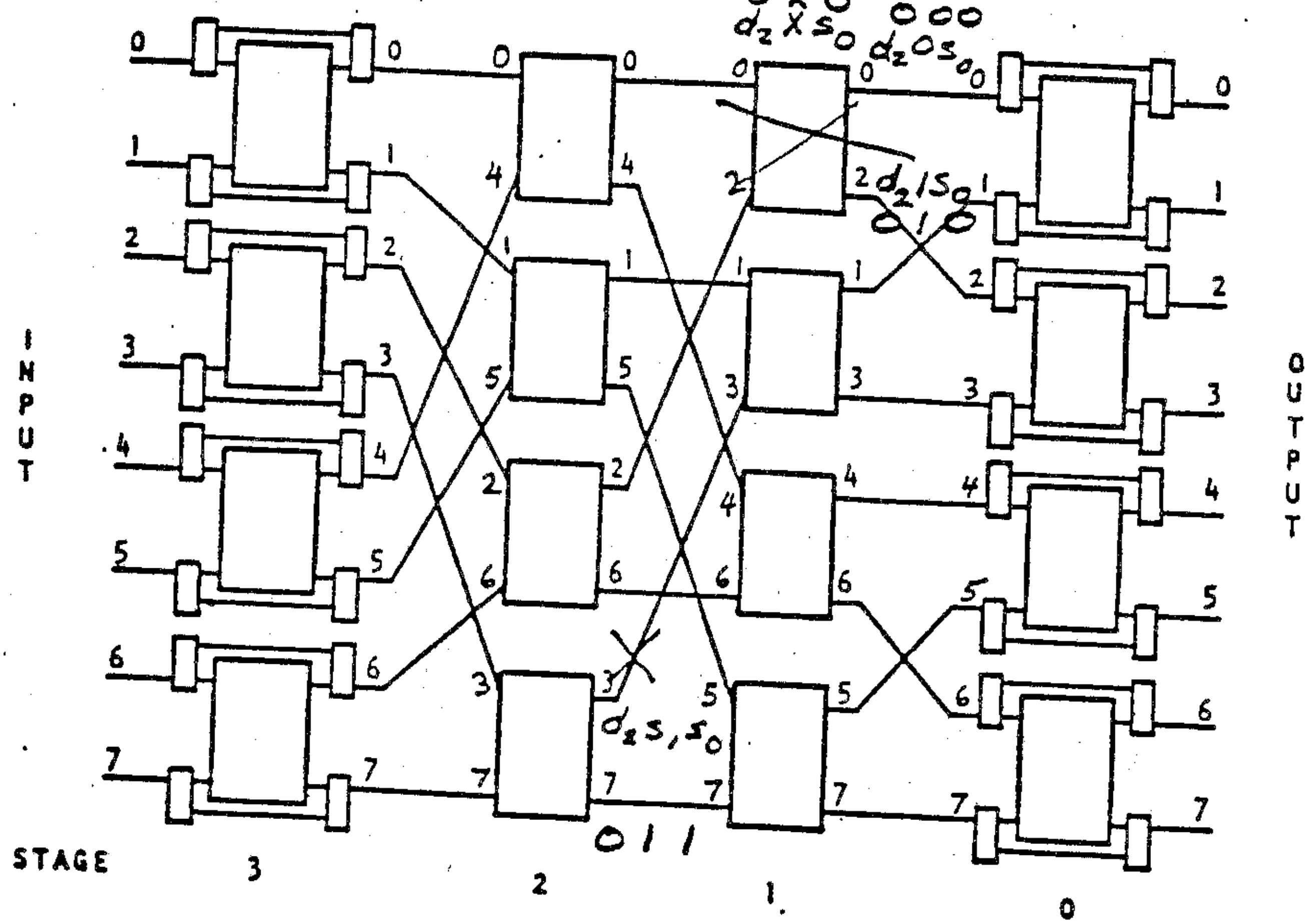
FAULT LABELS SENT TO ALL PEs
(FAULT NOT STAGE m OR 0 BOX)

BOX: PORT LABELS AND STAGE — 000, 010 = 0X0
1

LINK: LINK LABEL AND STAGE — 011
2

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

FAULT LABELS SENT TO ALL PES
(FAULT NOT STAGE m OR 0 BOX)

BOX: PORT LABELS AND STAGE   000, 010 = 0X0
1

LINK: LINK LABEL   011

Given fault label the source forms

Sourc

Destin

1. If stage i box fault

$$d_{m-1} \cdots d_{i+1} X s_{i-1} \cdots s_1$$

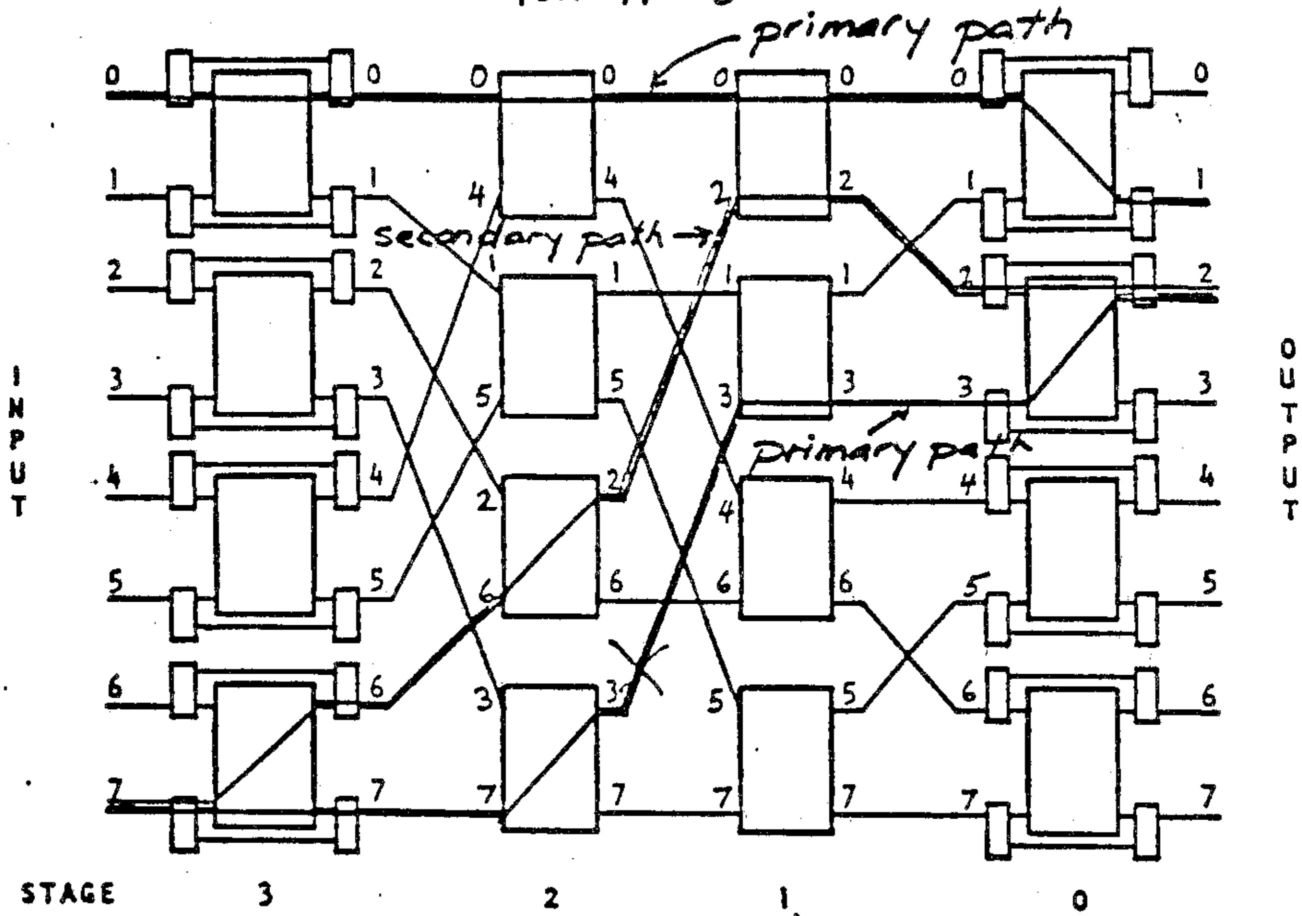
2. If stage i link fault

$$d_{m-1} \cdots d_i s_{i-1} \cdots s_1 s_0$$

If formed value matches th
primary path is faulty.

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= LOG_2N), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE LOG_2N OR 0

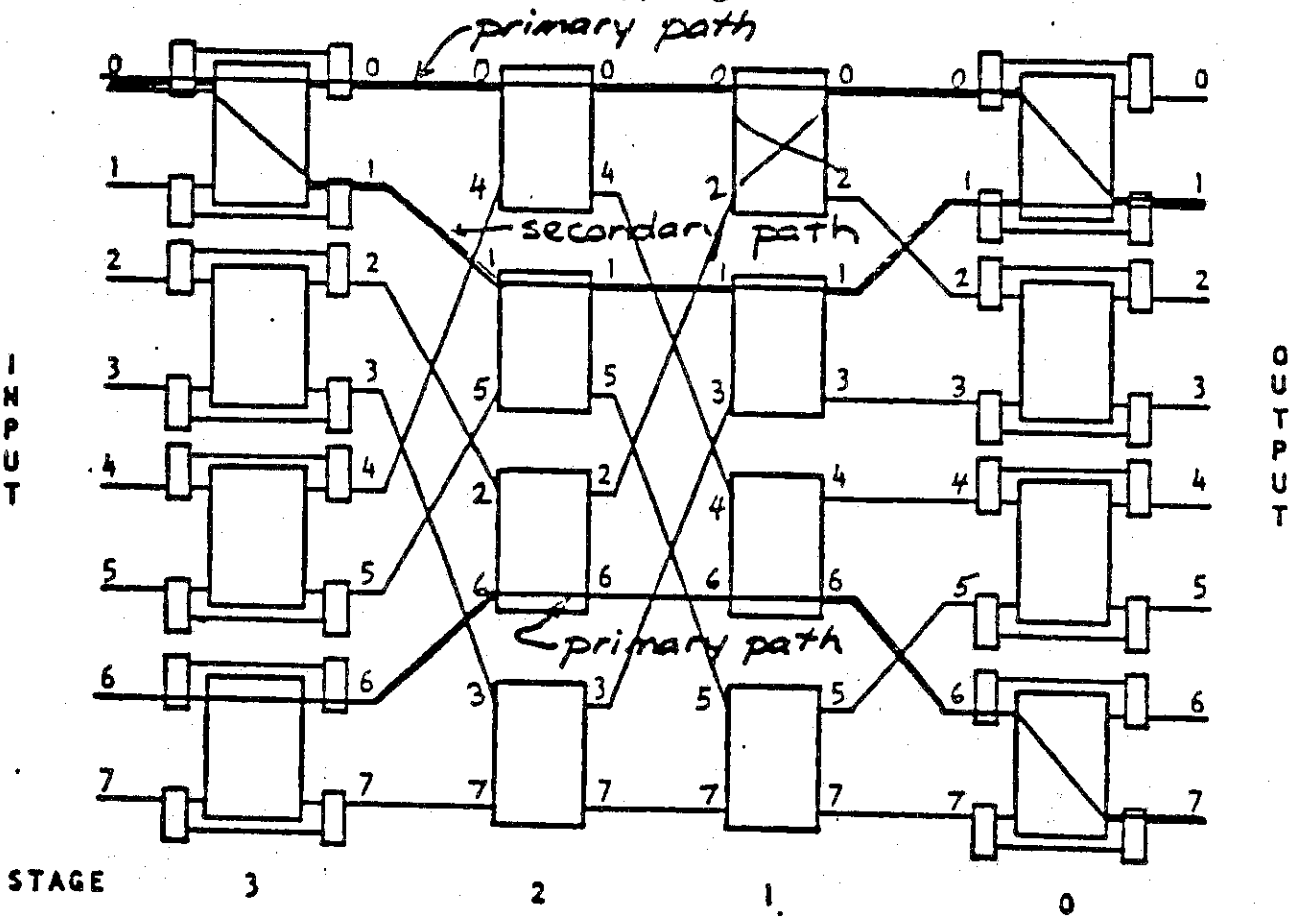
LINK FAULT: 011, 2

$7 \rightarrow 2$ $S = 111$ $D = 010$ $d_2 s_1 s_0 = 011$
match - blocked (primary)

$0 \rightarrow 1$ $S = 000$ $D = 001$ $d_2 s_1 s_0 = 000$
no match - not blocked
(primary)

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= $\log_2 N$), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE $\log_2 N$ OR 0

BOX FAULT: 0X0, 1

$0 \rightarrow 1$ $S = 000$ $D = 001$ $d_2 X s_0 = 0X0$
match - blocked (primary)

$6 \rightarrow 7$ $s = 110$ $D = 111$ $d_2 X s_0 = 1X1$
no match - not blocked
(primary)

Broadcast paths - use routing tag R, broadcast mask B

1. If stage i box fault

1 to 1 $\quad d_{m-1} \cdots d_{i+1} X s_{i-1} \cdots s_0$

broadcast $\quad$ use $W = w_{m-1} \cdots w_1 w_0$ to compare to fault label

$w_{i-1} \cdots w_0 = s_{i-1} \cdots s_0$
$w_i = X$
w_j for $i < j < m$:
if $b_j = 1$ then $w_j = X$
if $b_j = 0$ then $w_j = s_j \oplus r_j$
(common d_j)

2. If stage i link fault

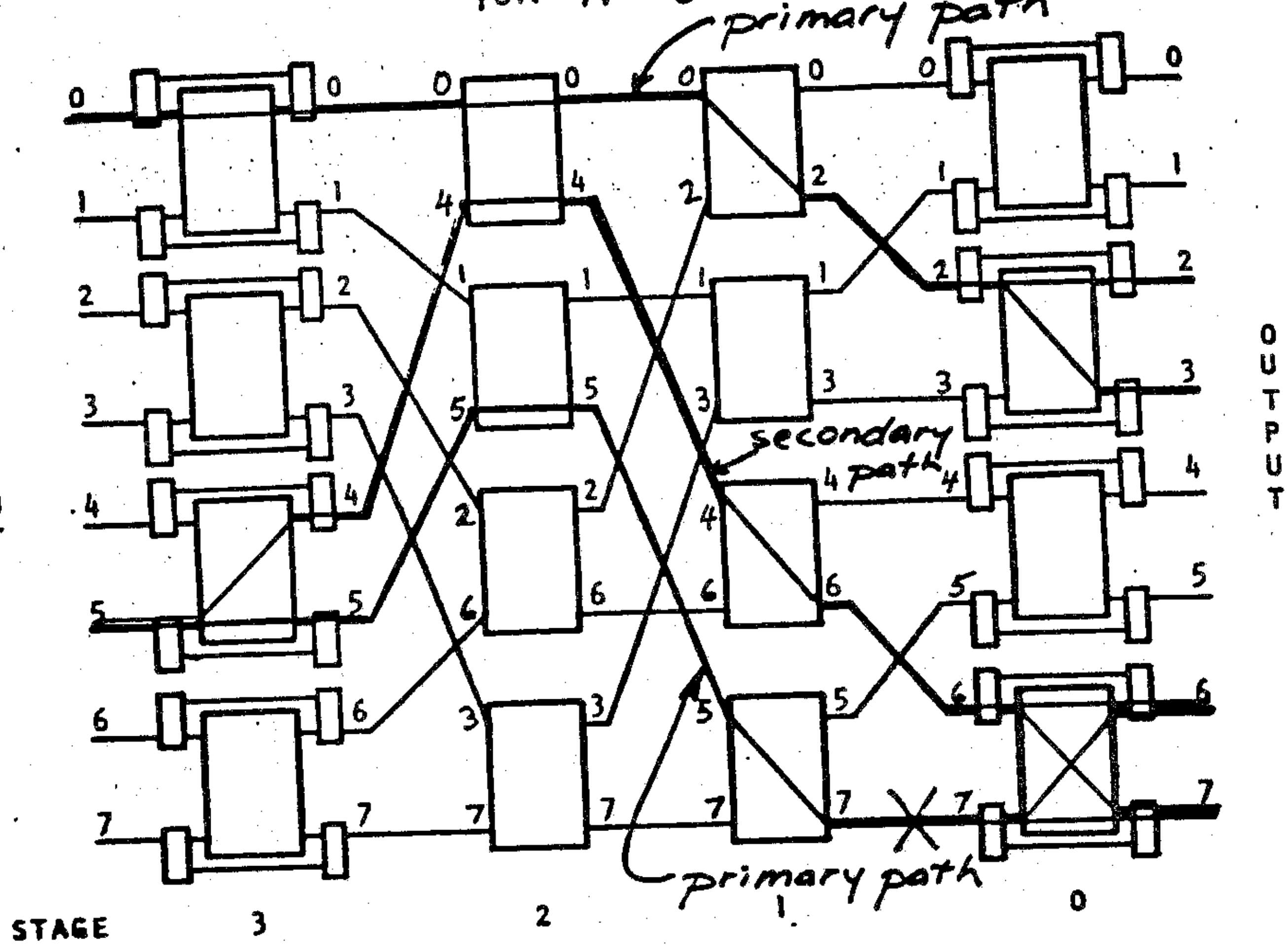
1 to 1 $\quad d_{m-1} \cdots d_i\, s_{i-1} \cdots s_0$

broadcast $\quad$ use $W = w_{m-1} \cdots w_1 w_0$ to compare

$w_{i-1} \cdots w_0 = s_{i-1} \cdots s_0$
w_j for $i \leqslant j < m$:
if $b_j = 1$ then $w_j = X$
if $b_j = 0$ then $w_j = s_j \oplus r_j$
(common d_j)

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= $\mathrm{LOG_2 N}$), IS A COPY OF STAGE 0

- INCLUDES CIRCUITRY TO BYPASS STAGE $\mathrm{LOG_2 N}$ OR 0

LINK FAULT: 111, 1

101 110 111 $\qquad s_2 \oplus r_2 \; s_1 \oplus r_1 \; s_0$

$5 \rightarrow 6 + 7 \quad R = 011 \quad B = 001 \quad W_2 \, W_1 \, W_0 = 111$

match - blocked (primary)

000 010 011 $\qquad s_2 \oplus r_2 \; s_1 \oplus r_1 \; s_0$

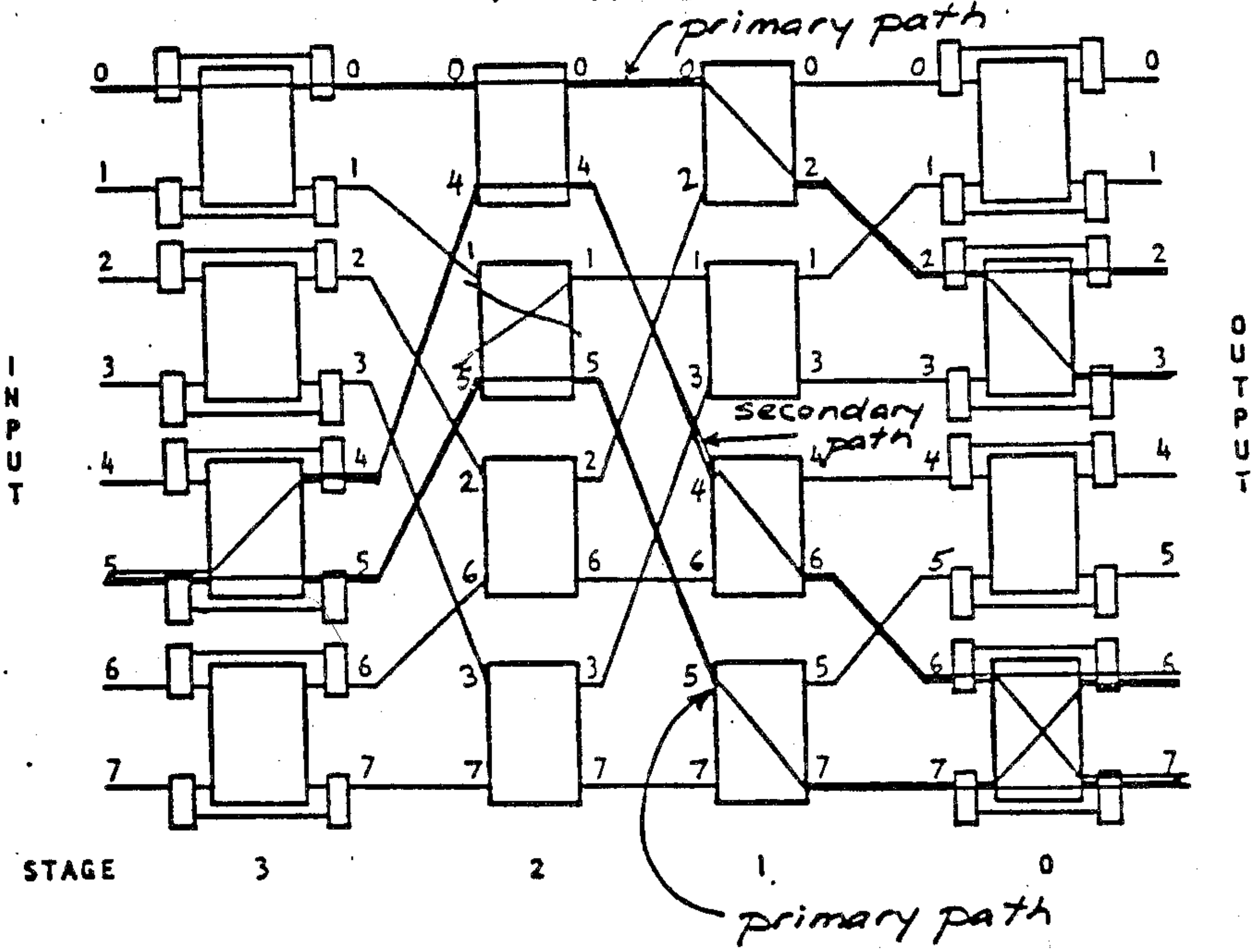
$0 \rightarrow 2 + 3 \quad R = 010 \quad B = 001 \quad W_2 \, W_1 \, W_0 = C$

no match - not blocked

(primary)

THE EXTRA STAGE CUBE NETWORK

FOR $N = 8$



- EXTRA STAGE, STAGE 3 (= $\text{LOG}_2 N$), IS A COPY OF STAGE 0
- INCLUDES CIRCUITRY TO BYPASS STAGE $\text{LOG}_2 N$ OR 0

BOX FAULT: X01, 2

$\overset{101}{5} \rightarrow \overset{110}{6} + \overset{111}{7} \quad R = 011 \quad B = 001 \quad \overset{X}{W_2}\,\overset{S_1}{W_1}\,\overset{S_0}{W_0} = X01$

match - blocked (primary)

$\overset{000}{0} \rightarrow \overset{010}{2} + \overset{011}{3} \quad R = 010 \quad B = 001 \quad \overset{X}{W_2}\,\overset{S_1}{W_1}\,\overset{S_0}{W_0} = X00$

no match - not blocked (primary)

Fast test to determine if primary path *may* be faulty

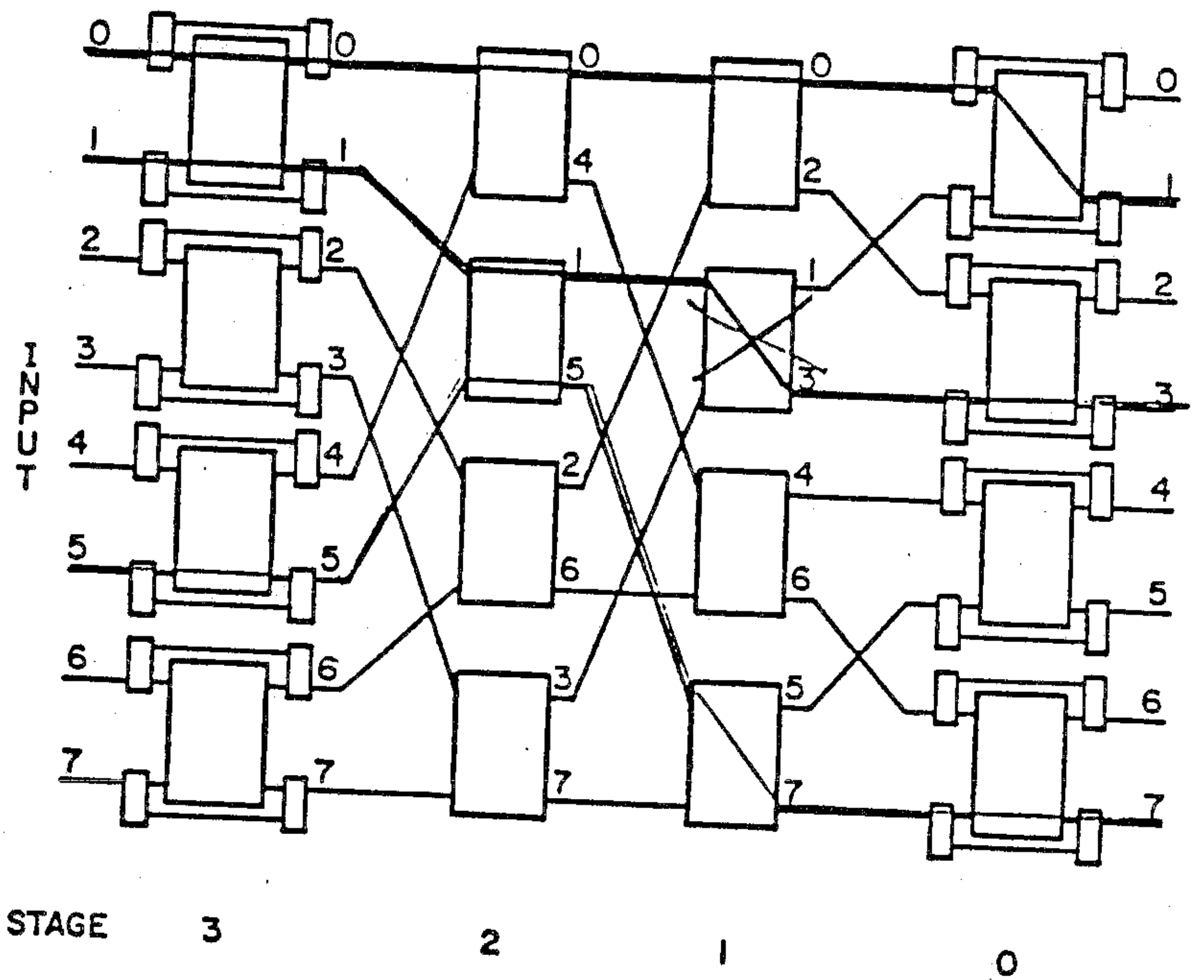
Compare s_0 to low-order bit of fault label

— if different, fault not on primary

— if same, fault *may* be on primary so use secondary (may cause unnecessary use of secondary paths)

BOX FAULT: 0X1

0 → 1 $s_0 = 0$ PRIMARY NOT BLOCKED
1 → 3 $s_0 = 1$ PRIMARY MAY BE BLOCKED (IT IS)
5 → 7 $s_0 = 1$ PRIMARY MAY BE BLOCKED (IT IS NO



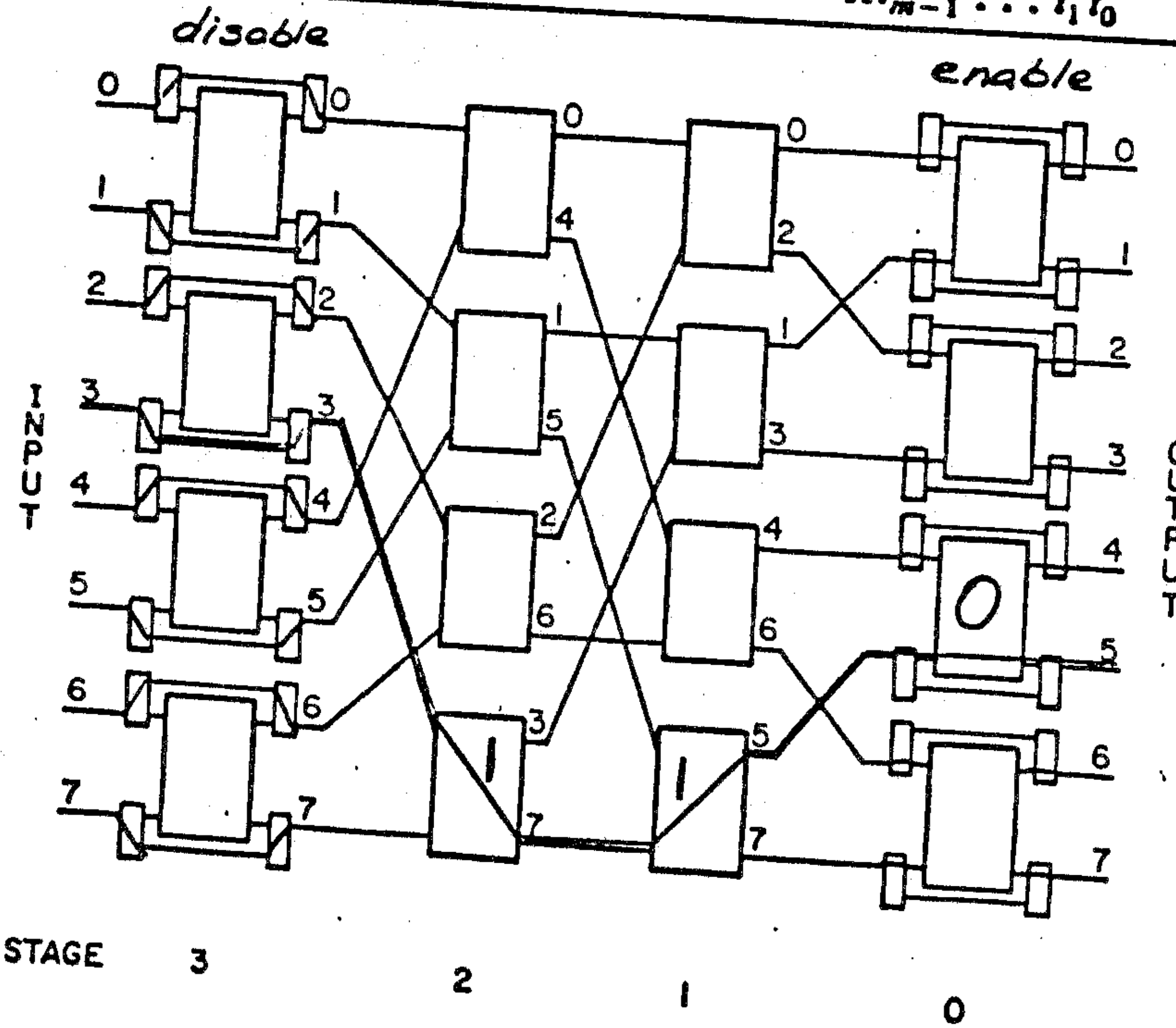
One-to-One Routing Tags for the ESC Network
(X = 0 or 1)

Fault Location	*Routing Tag T^**
No fault	$T^* = Xt_{m-1} \ldots t_1t_0$
Stage 0 box	$T^* = t_0t_{m-1} \ldots t_1X$
Stage i box, $1 \le i < m$, or any link	$T^* = 0t_{m-1} \ldots t_1t_0$ if primary path is fault-free; $T^* = 1t_{m-1} \ldots t_1\bar{t}_0$ if primary path contains fault
Stage m box	$T^* = Xt_{m-1} \ldots t_1t_0$

One-to-One Routing Tags for the ESC Network (X = 0 or 1)

$3 \longrightarrow 5$ $\quad T = 3 \oplus 5 = 110 = t_2 t_1 t_0$
011 $\quad$ 101

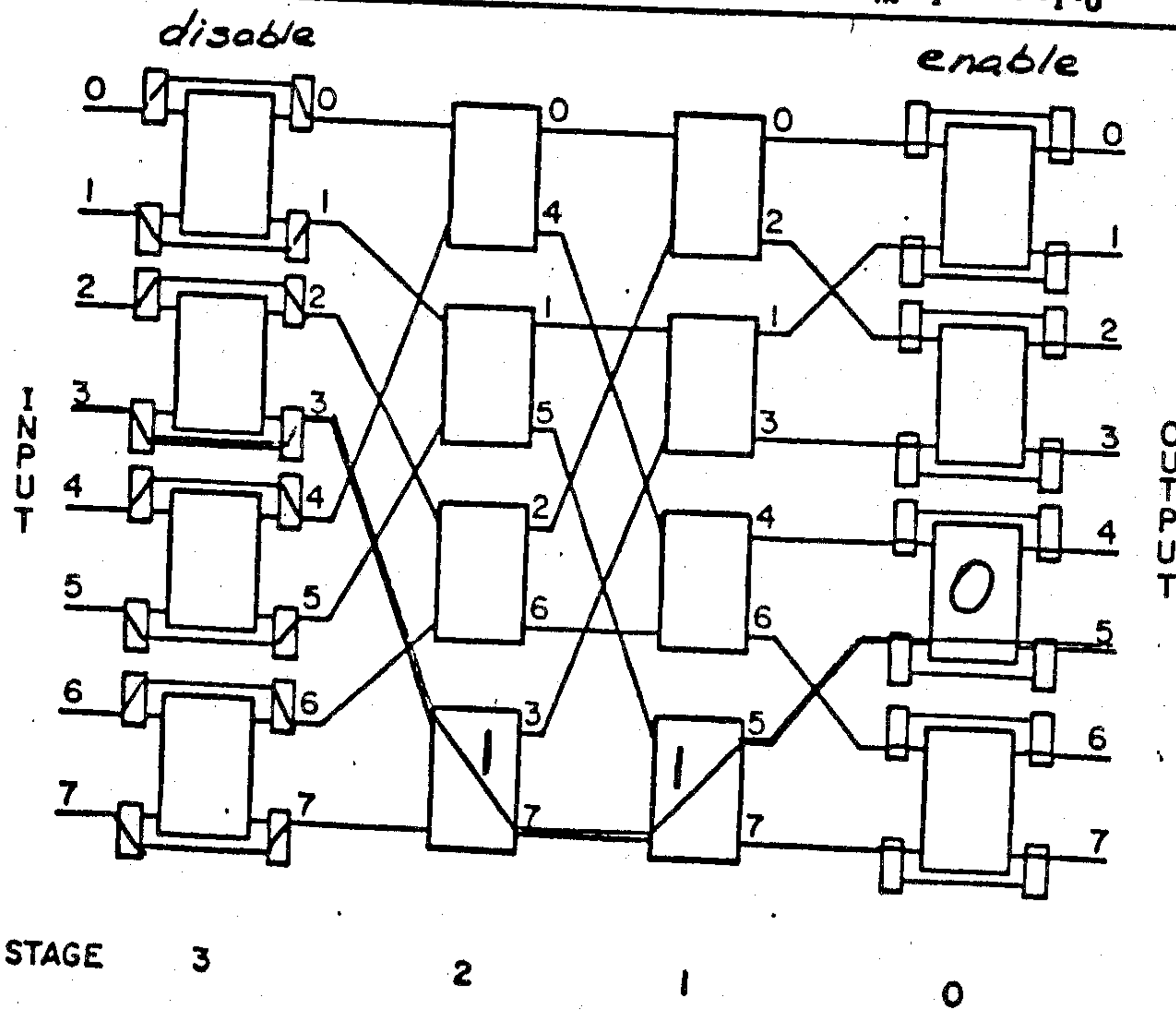
Fault Location	Routing Tag T^*
No fault	$T^* = Xt_{m-1} \ldots t_1 t_0 = X110$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X$
Stage i box, $1 \le i < m$, or any link	$T^* = 0t_{m-1} \ldots t_1 t_0$ if primary path is fault-free; $T^* = 1t_{m-1} \ldots t_1 \bar{t}_0$ if primary path contains fault
Stage m box	$T^* = Xt_{m-1} \ldots t_1 t_0$



One-to-One Routing Tags for the ESC Network (X = 0 or 1)

$\underset{011}{3} \rightarrow \underset{101}{5} \quad T = 3 \oplus 5 = 110 = t_2 t_1 t_0$

Fault Location	Routing Tag T^*
No fault	$T^* = X t_{m-1} \ldots t_1 t_0 = X110$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X$
Stage i box, $1 \leq i < m$, or any link	$T^* = 0 t_{m-1} \ldots t_1 t_0$ if primary path is fault-free; $T^* = 1 t_{m-1} \ldots t_1 \bar{t}_0$ if primary path contains fault
Stage m box	$T^* = X t_{m-1} \ldots t_1 t_0$

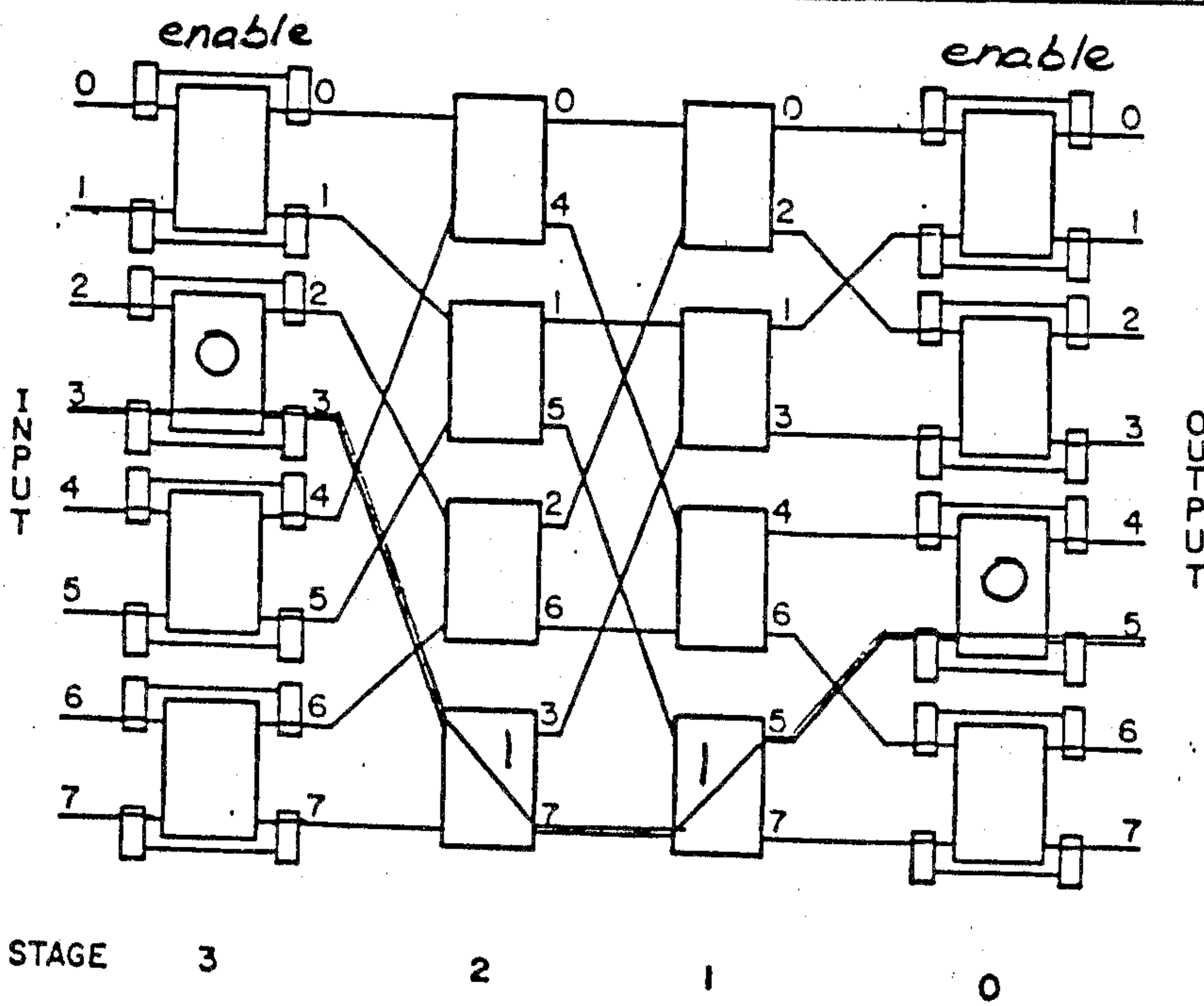


One-to-One Routing Tags for the ESC Network
(X = 0 or 1)

$3 \rightarrow 5 \quad T = 3 \oplus 5 = 110$

$011 \quad 101 \quad t_2 t_1 t_0$

Fault Location	*Routing Tag T^**
No fault	$T^* = Xt_{m-1} \ldots t_1 t_0$ $\quad t_3^* t_2^* t_1^* t_0^*$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X$ $\quad 0\, t_2\, t_1\, t_0$
Stage i box, $1 \leq i < m$, or any link	$T^* = 0t_{m-1} \ldots t_1 t_0 = 0110$ if primary path is fault-free; $T^* = 1t_{m-1} \ldots t_1 \bar{t}_0$ if primary path contains fault
Stage m box	$T^* = Xt_{m-1} \ldots t_1 t_0$

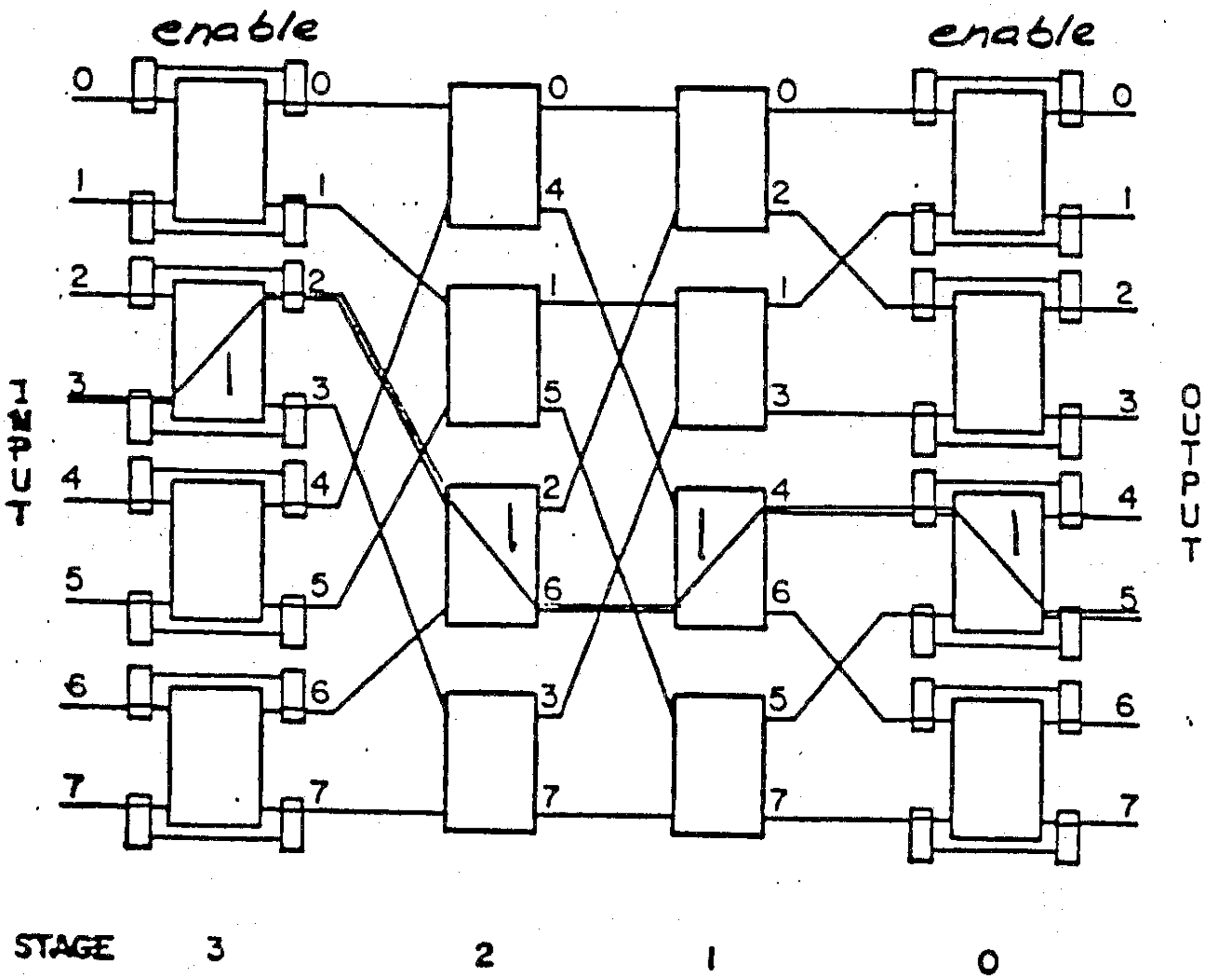


One-to-One Routing Tags for the ESC Network (X = 0 or 1)

$3 \rightarrow 5 \quad T = 3 \oplus 5 = 110$

$011 \quad 101 \quad t_2\, t_1\, t_0$

Fault Location	*Routing Tag T^**
No fault	$T^* = Xt_{m-1} \ldots t_1 t_0$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X$
Stage i box, $1 \leq i < m$, or any link	$T^* = 0t_{m-1} \ldots t_1 t_0$ $\quad t_3^* t_2^* t_1^* t_0^*$ if primary path is fault-free; $1\, t_2\, t_1\, \bar{t}_0$ $T^* = 1t_{m-1} \ldots t_1 \bar{t}_0 = 111\bar{0} = 1111$ if primary path contains fault
Stage m box	$T^* = Xt_{m-1} \ldots t_1 t_0$



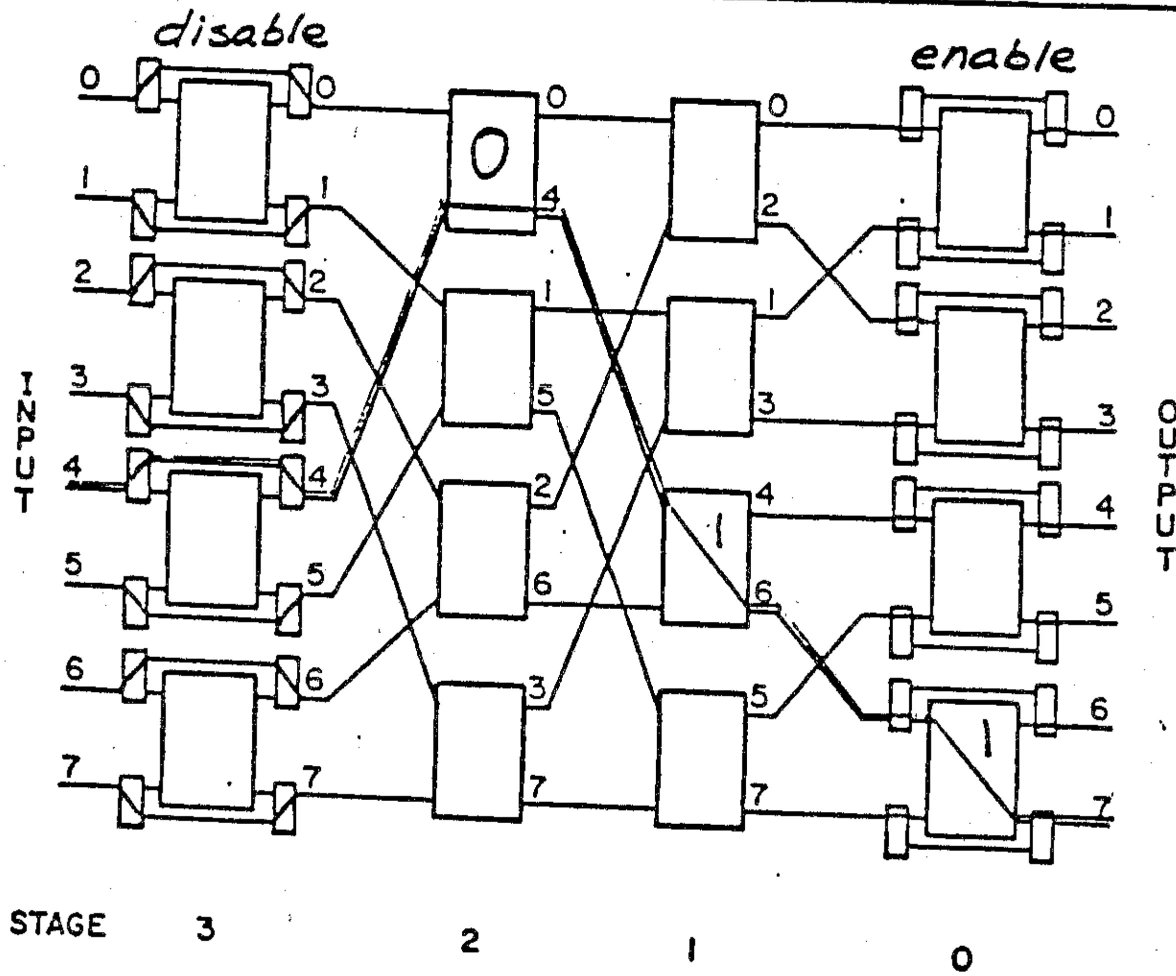
start again after missing Friday

One-to-One Routing Tags for the ESC Network (X = 0 or 1)

$4 \rightarrow 7 \quad T = 4 \oplus 7 = 011$

100 111 $t_2 t_1 t_0$

Fault Location	Routing Tag T^*
No fault	$T^* = Xt_{m-1} \ldots t_1 t_0$ $= X011$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X$
Stage i box, $1 \leq i < m$, or any link	$T^* = 0t_{m-1} \ldots t_1 t_0$ if primary path is fault-free; $T^* = 1t_{m-1} \ldots t_1 \bar{t}_0$ if primary path contains fault
Stage m box	$T^* = Xt_{m-1} \ldots t_1 t_0$

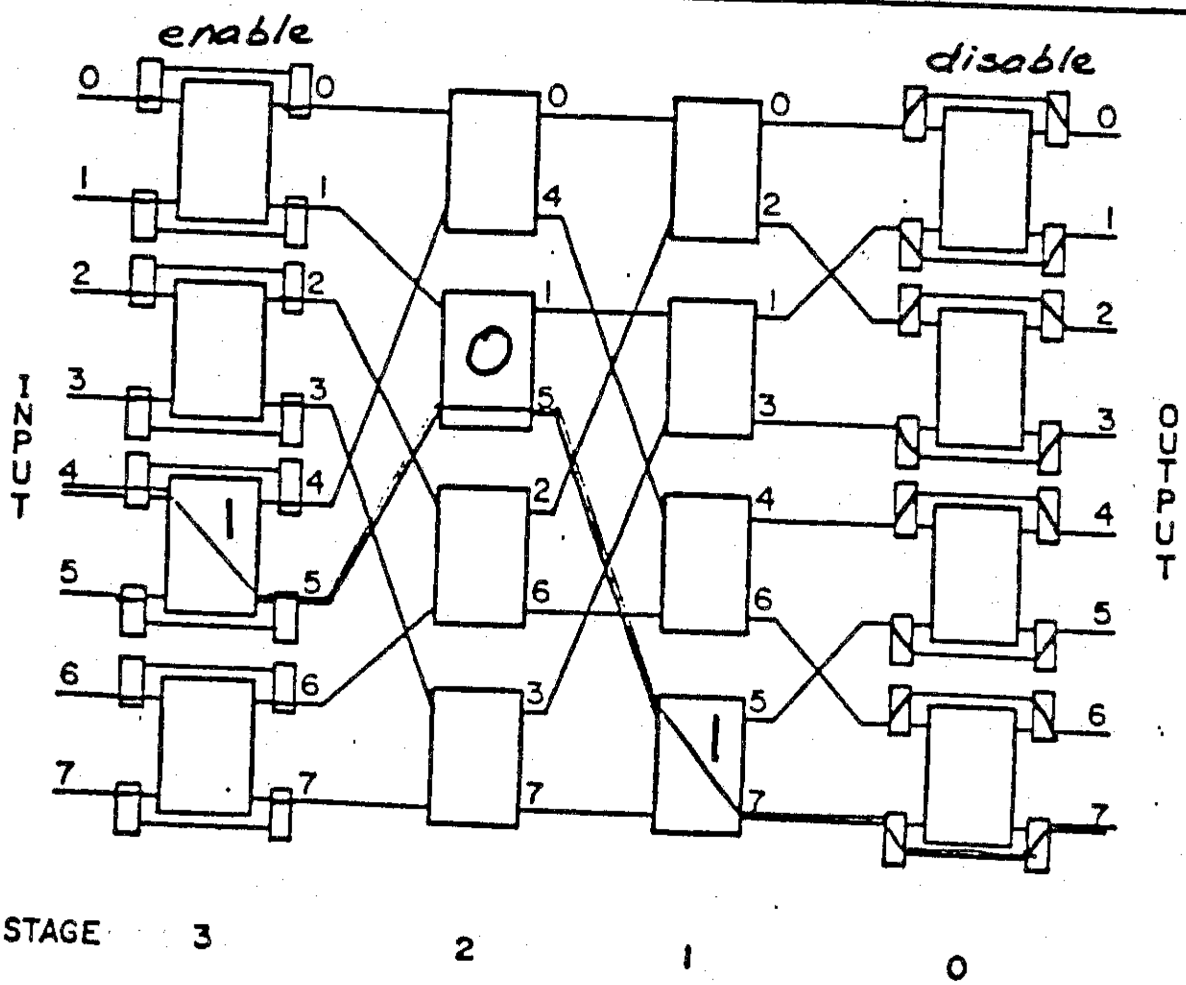


One-to-One Routing Tags for the ESC Network ($X = 0$ or 1)

$4 \rightarrow 7 \quad T = 4 \oplus 7 = 011$

$\underset{100}{4} \quad \underset{111}{7} \quad \underset{t_2\, t_1\, t_0}{011}$

Fault Location	Routing Tag T^*
No fault	$T^* = Xt_{m-1} \ldots t_1 t_0$ $\quad t_3^* t_2^* t_1^* t_0^*$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X = 101X$ $\quad t_0 t_2 t_1 X$
Stage i box, $1 \leq i < m$, or any link	$T^* = 0t_{m-1} \ldots t_1 t_0$ if primary path is fault-free; $T^* = 1t_{m-1} \ldots t_1 \bar{t}_0$ if primary path contains fault
Stage m box	$T^* = Xt_{m-1} \ldots t_1 t_0$

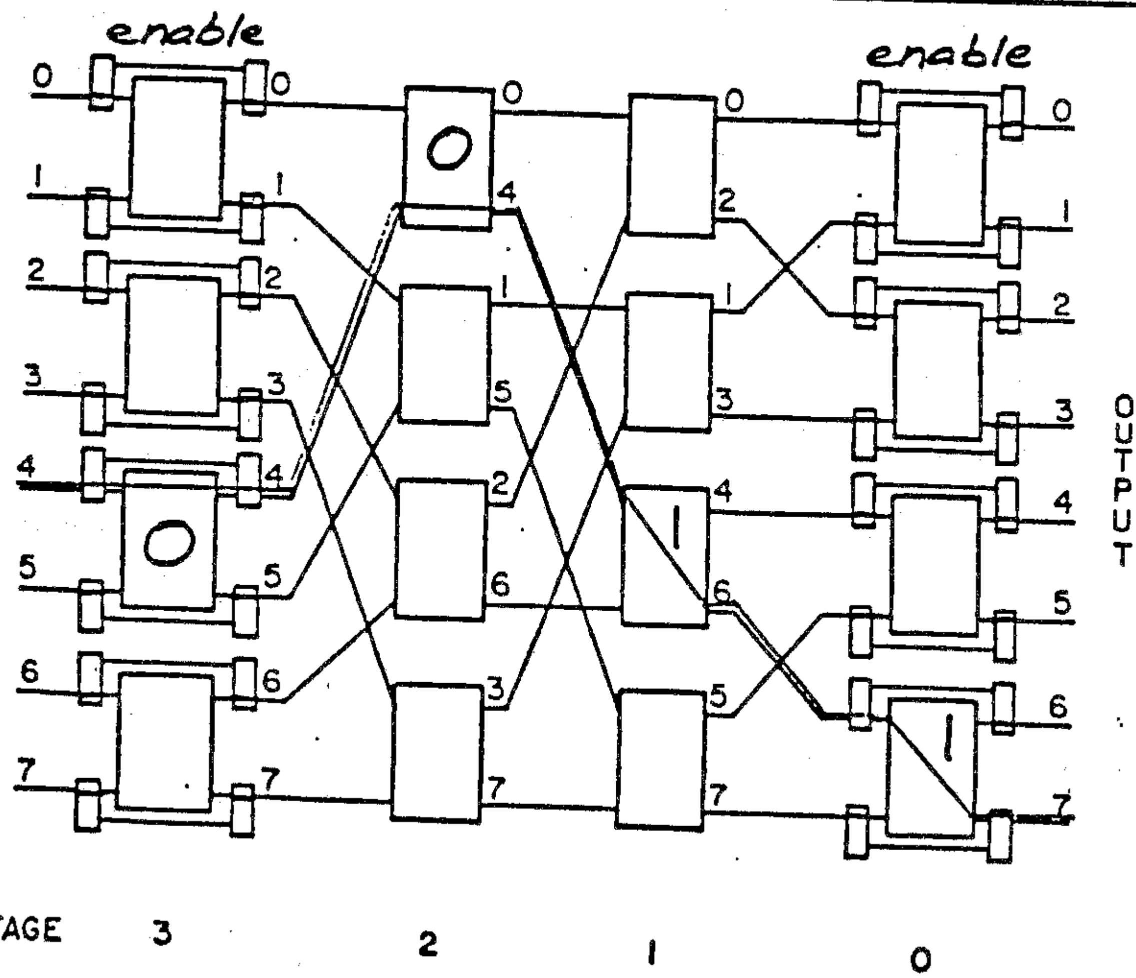


One-to-One Routing Tags for the ESC Network
(X = 0 or 1)

$4 \rightarrow 7 \quad T = 4 \oplus 7 = 011$

$100 \quad 111 \quad t_2 t_1 t_0$

Fault Location	*Routing Tag T^**
No fault	$T^* = X t_{m-1} \ldots t_1 t_0$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X$ $t_3^* t_2^* t_1^* t_0^*$ $0\, t_2 t_1 t_0$
Stage i box, $1 \leq i < m$, or any link	$T^* = 0 t_{m-1} \ldots t_1 t_0 = 0011$ if primary path is fault-free; $T^* = 1 t_{m-1} \ldots t_1 \bar{t}_0$ if primary path contains fault
Stage m box	$T^* = X t_{m-1} \ldots t_1 t_0$

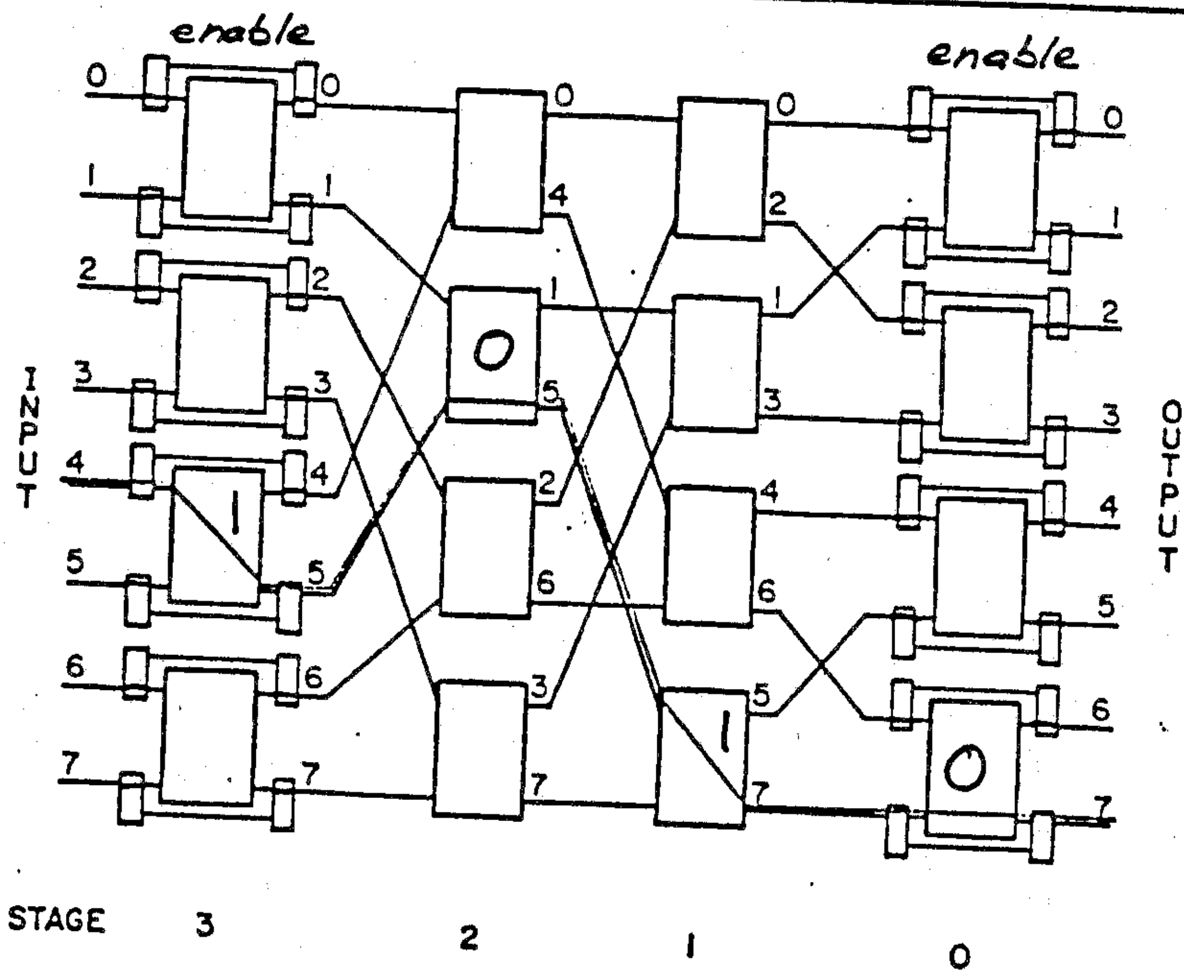


One-to-One Routing Tags for the ESC Network
(X = 0 or 1)

$4 \rightarrow 7 \quad T = 4 \oplus 7' = 011$

$100 \quad 111 \quad t_2 t_1 t_0$

Fault Location	*Routing Tag T^**
No fault	$T^* = Xt_{m-1} \ldots t_1 t_0$
Stage 0 box	$T^* = t_0 t_{m-1} \ldots t_1 X$
Stage i box, $1 \le i < m$, or any link	$T^* = 0t_{m-1} \ldots t_1 t_0$ $t_3^* t_2^* t_1^* t_0^*$ if primary path is fault-free; $1 t_2 t_1 t_0$ $T^* = 1t_{m-1} \ldots t_1 \bar{t}_0 = 101\bar{1} = 1010$ if primary path contains fault
Stage m box	$T^* = Xt_{m-1} \ldots t_1 t_0$



Broadcast Routing Tags for the ESC Network *(X = 0 or 1)*

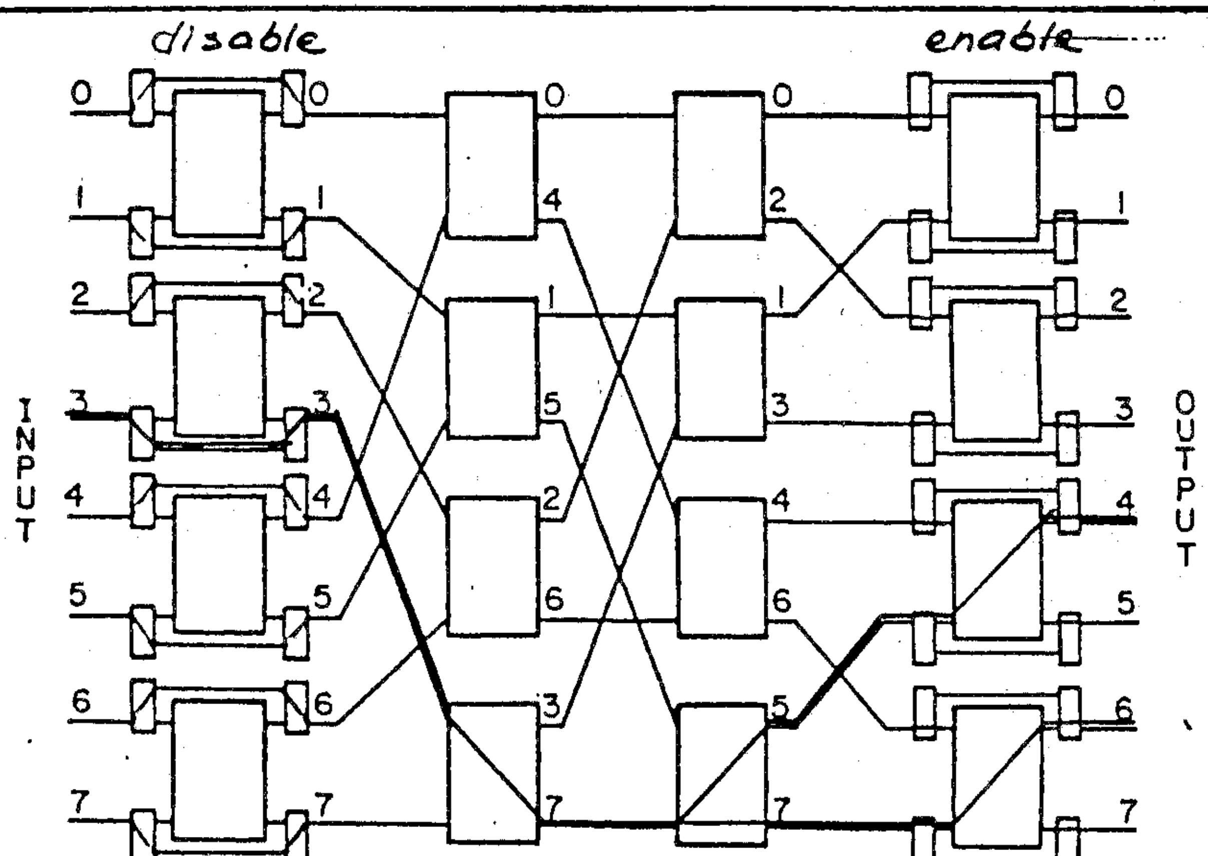
Fault Location	*Routing Tag* R^*, B^*
No fault	$R^* = Xr_{m-1} \ldots r_1 r_0$ $B^* = Xb_{m-1} \ldots b_1 b_0$
Stage 0 box	$R^* = r_0 r_{m-1} \ldots r_1 X$ $B^* = b_0 b_{m-1} \ldots b_1 X$
Stage i box, $1 \leq i < m$, or any link	$R^* = 0r_{m-1} \ldots r_1 r_0$ $B^* = 0b_{m-1} \ldots b_1 b_0$ if primary path is fault-free; $R^* = 1r_{m-1} \ldots r_1 \overline{r_0}$ $B^* = 0b_{m-1} \ldots b_1 b_0$ if primary broadcast path contains fault
Stage m box	$R^* = Xr_{m-1} \ldots r_1 r_0$ $B^* = Xb_{m-1} \ldots b_1 b_0$

Broadcast Routing Tags for the ESC Network *(X = 0 or 1)*

$3 \rightarrow 4 + 6$ $\quad R = 3 \oplus 4 = 111$ $\quad B = 4 \oplus 6 = 010$

$011 \quad 100 \quad 110 \qquad r_2 r_1 r_0 \qquad b_2 b_1 b_0$

Fault Location	*Routing Tag* R^*, B^*
No fault	$R^* = Xr_{m-1} \ldots r_1 r_0 = X111 = r_3^* r_2^* r_1^* r_0^*$ $B^* = Xb_{m-1} \ldots b_1 b_0 = X010 = b_3^* b_2^* b_1^* b_0^*$
Stage 0 box	$R^* = r_0 r_{m-1} \ldots r_1 X$ $B^* = b_0 b_{m-1} \ldots b_1 X$
Stage i box, $1 \leq i < m$, or any link	$R^* = 0r_{m-1} \ldots r_1 r_0$ $B^* = 0b_{m-1} \ldots b_1 b_0$ if primary path is fault-free; $R^* = 1r_{m-1} \ldots r_1 \bar{r}_0$ $B^* = 0b_{m-1} \ldots b_1 b_0$ if primary broadcast path contains fault
Stage m box	$R^* = Xr_{m-1} \ldots r_1 r_0 = X111$ $B^* = Xb_{m-1} \ldots b_1 b_0 = X010$



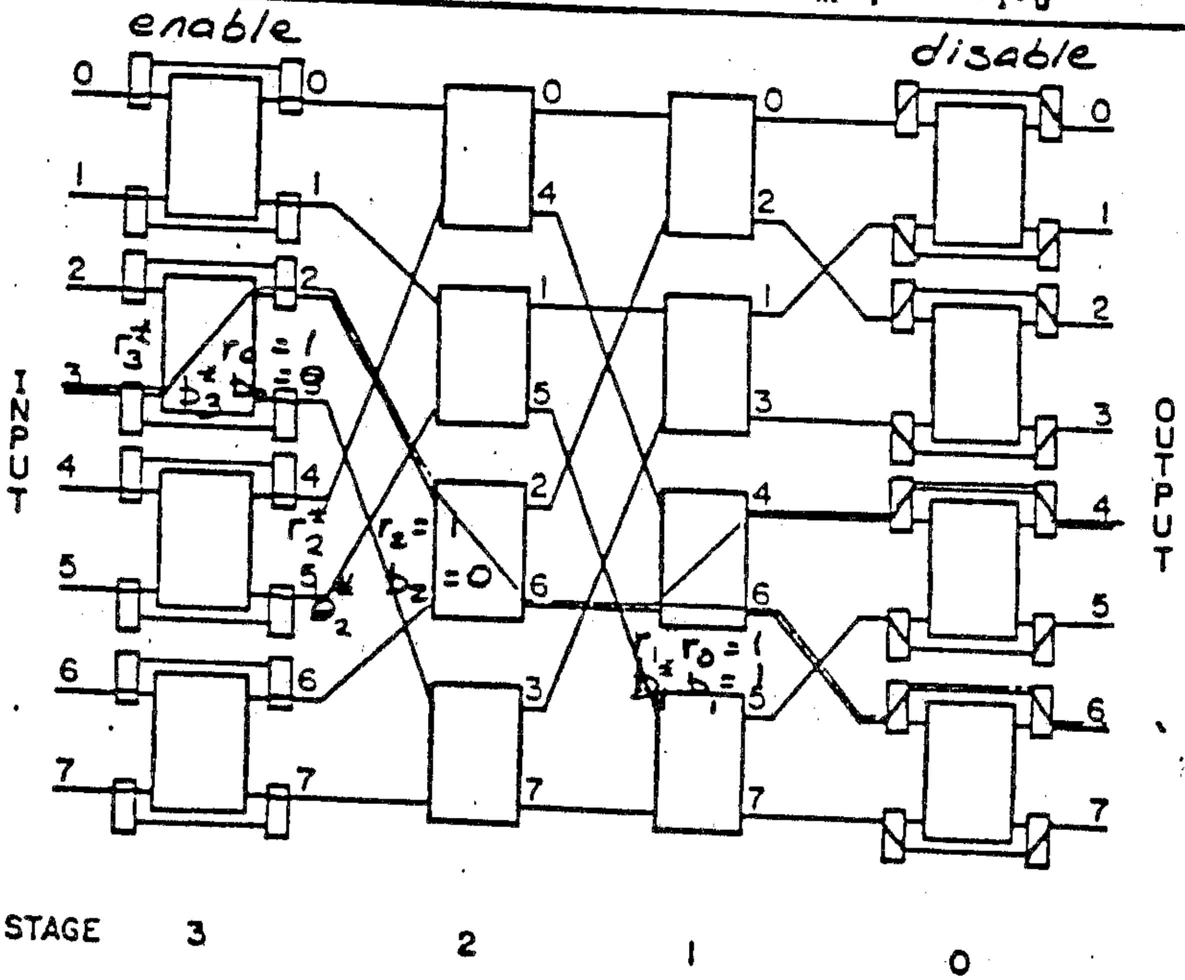
$r_2^* \; r_2 = 1 \qquad r_1^* \; r_1 = 1 \qquad r_0^* \; r_0 = 1$

STAGE 3 $b_2^* \; b_2 = 0$ 2 $\quad b_1^* \; b_1 = 1 \quad b_0^* \; b_0 = 0$

Broadcast Routing Tags for the ESC Network (X = 0 or 1)

3 → 4 + 6   R = 3 ⊕ 4 = 111   B = 4 ⊕ 6 = 010
011   100 110 $r_2 r_1 r_0$ $b_2 b_1 b_0$

Fault Location	*Routing Tag* R^*, B^*
No fault	$R^* = Xr_{m-1} \ldots r_1 r_0$ $B^* = Xb_{m-1} \ldots b_1 b_0$
Stage 0 box	$R^* = r_0 r_{m-1} \ldots r_1 X = 111X = r_3^* r_2^* r_1^* r_0^*$ ($r_0 r_2 r_1 X$) $B^* = b_0 b_{m-1} \ldots b_1 X = 001X = b_3^* b_2^* b_1^* b_0^*$ ($b_0 b_2 b_1 X$)
Stage i box, $1 \leq i < m$, or any link	$R^* = 0r_{m-1} \ldots r_1 r_0$ $B^* = 0b_{m-1} \ldots b_1 b_0$ if primary path is fault-free; $R^* = 1r_{m-1} \ldots r_1 \overline{r_0}$ $B^* = 0b_{m-1} \ldots b_1 b_0$ if primary broadcast path contains fault
Stage m box	$R^* = Xr_{m-1} \ldots r_1 r_0$ $B^* = Xb_{m-1} \ldots b_1 b_0$

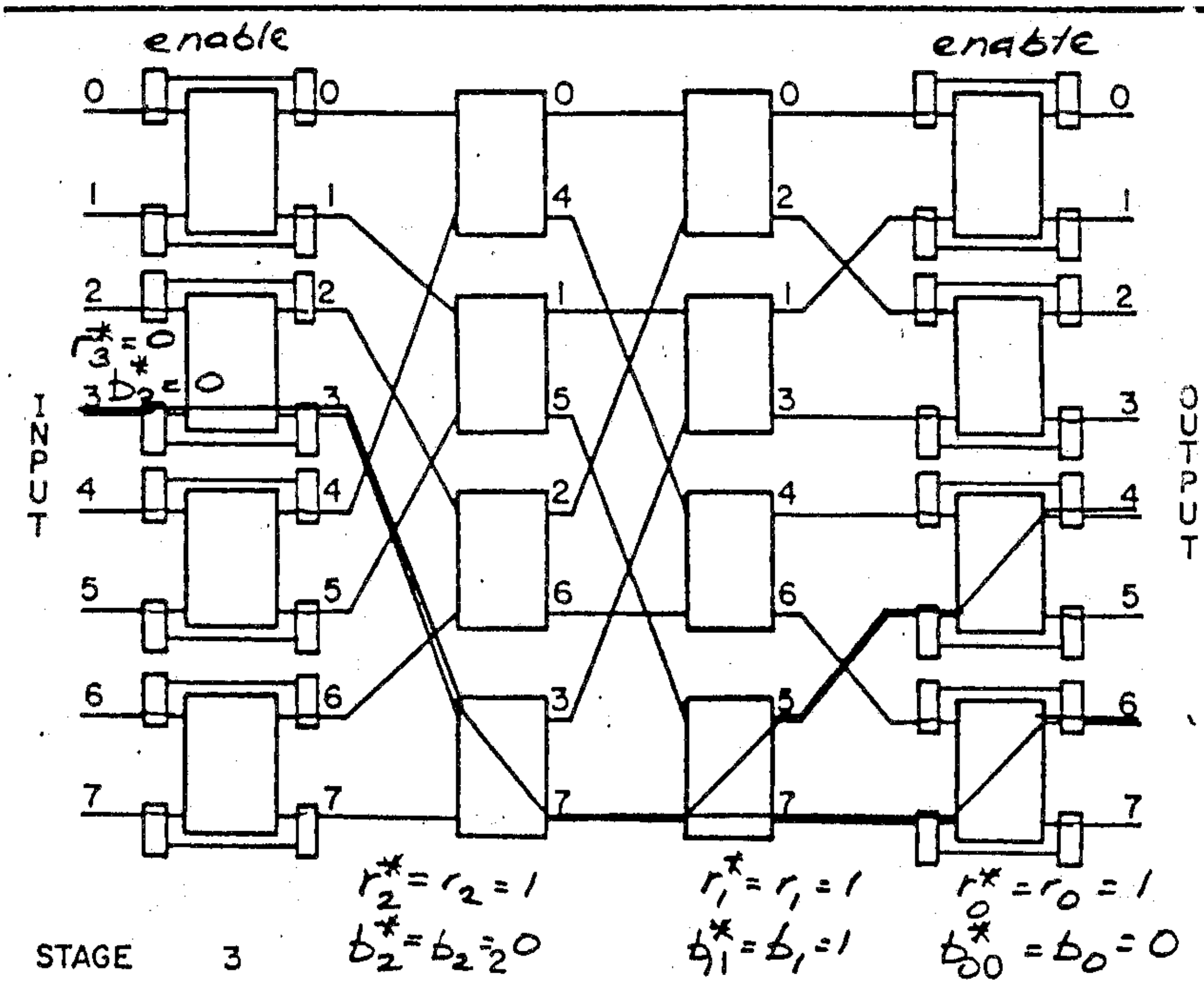


Broadcast Routing Tags for the ESC Network *(X = 0 or 1)*

$3 \rightarrow 4+6 \quad R = 3 \oplus 4 = 111 \quad B = 4 \oplus 6 = 010$

$011 \quad 100 \; 110 \quad r_2 r_1 r_0 \quad b_2 b_1 b_0$

Fault Location	*Routing Tag* R^*, B^*
No fault	$R^* = Xr_{m-1} \ldots r_1 r_0$ $B^* = Xb_{m-1} \ldots b_1 b_0$
Stage 0 box	$R^* = r_0 r_{m-1} \ldots r_1 X$ $B^* = b_0 b_{m-1} \ldots b_1 X$
Stage i box, $1 \leq i < m$, or any link	$R^* = 0r_{m-1} \ldots r_1 r_0 = 0111 = r_3^* r_2^* r_1^* r_0^*$ $B^* = 0b_{m-1} \ldots b_1 b_0 = 0000 = b_3^* b_2^* b_1^* b_0^*$ if primary path is fault-free; $R^* = 1r_{m-1} \ldots r_1 \bar{r}_0$ $B^* = 0b_{m-1} \ldots b_1 b_0$ if primary broadcast path contains fault
Stage m box	$R^* = Xr_{m-1} \ldots r_1 r_0$ $B^* = Xb_{m-1} \ldots b_1 b_0$

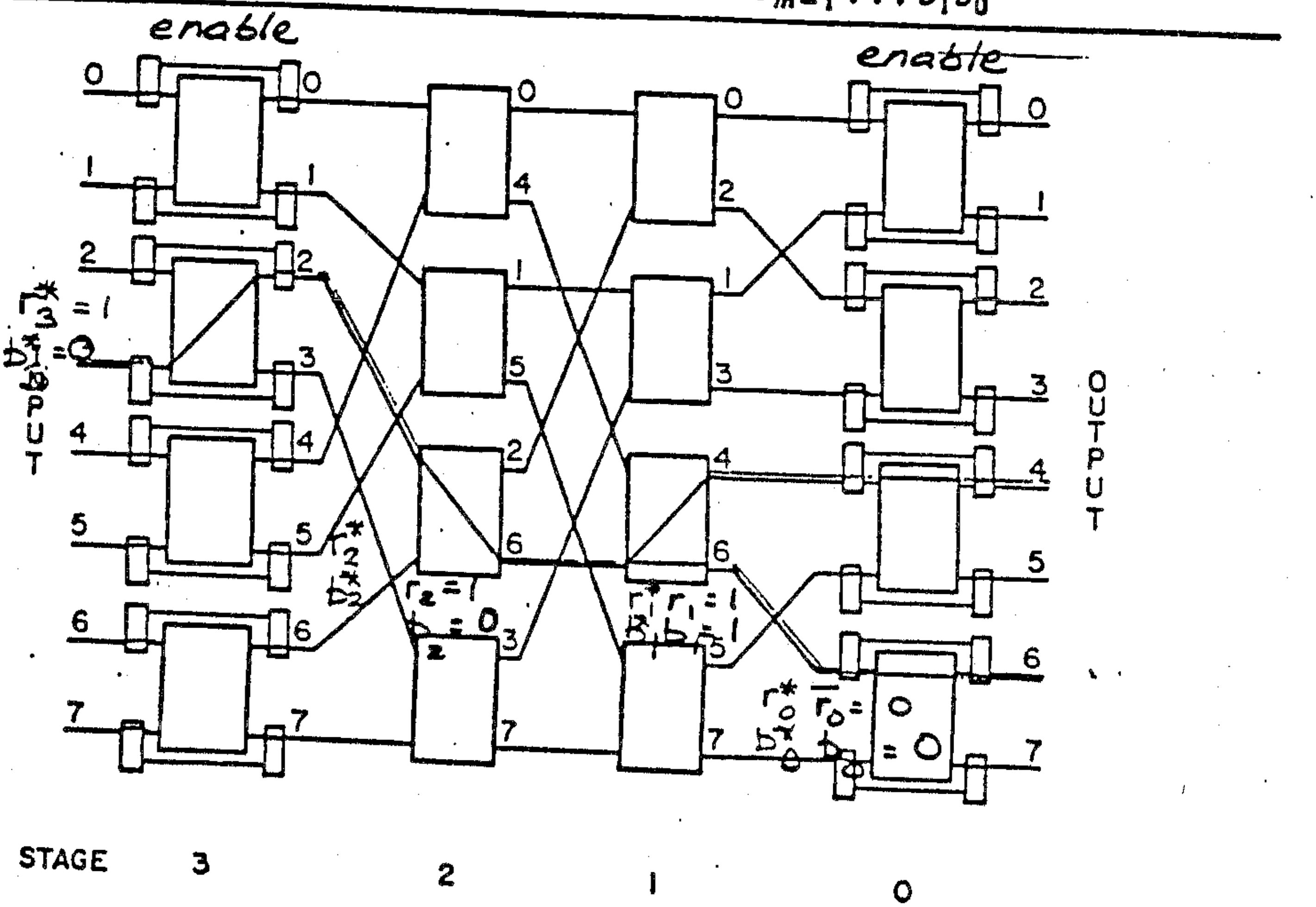


Broadcast Routing Tags for the ESC Network $(X = 0 \text{ or } 1)$

$3 \rightarrow 4 + 6$ $\quad R = 3 \oplus 4 = 111$ $\quad B = 4 \oplus 6 = 010$

$011 \quad 100\ 110 \qquad r_2 r_1 r_0 \qquad b_2 b_1 b_0$

Fault Location	*Routing Tag* R^*, B^*
No fault	$R^* = X r_{m-1} \ldots r_1 r_0$ $B^* = X b_{m-1} \ldots b_1 b_0$
Stage 0 box	$R^* = r_0 r_{m-1} \ldots r_1 X$ $B^* = b_0 b_{m-1} \ldots b_1 X$
Stage i box, $1 \le i < m$, or any link	$R^* = 0 r_{m-1} \ldots r_1 r_0$ $B^* = 0 b_{m-1} \ldots b_1 b_0$ if primary path is fault-free; $= r_3^* r_2^* r_1^* r_0^*$ $R^* = 1 r_{m-1} \ldots r_1 \overline{r_0} = 111\overline{1} = 1110$ $B^* = 0 b_{m-1} \ldots b_1 b_0 = 0010 = b_3^* b_2^* b_1^* b_0^*$ if primary broadcast path contains fault
Stage m box	$R^* = X r_{m-1} \ldots r_1 r_0$ $B^* = X b_{m-1} \ldots b_1 b_0$



Fault Handling in Extra Stage Cube

- if no fault
  - disable stage m, enable stage 0
  - use routing tag $T^* = Xt_{m-1}...t_1t_0$
- stage 0 box fault
  - disable stage 0, enable stage m, notify devices
  - use routing tag $T^* = t_0t_{m-1}...t_1X$
- stage i box fault, $1 \leq i < m$, or link fault
  - enable stage m and 0
  - send devices fault label

    stage i, link J $\rightarrow$ (i, J, link)

    stage i, box with link J $\rightarrow$ (i, J, box)
  - link: compare $d_{m-1}...d_{i+1}d_is_{i-1}...s_1s_0$ to J

    box: compare $d_{m-1}...d_{i+1}Xs_{i-1}...s_1s_0$ to J

    if no match use $T^* = 0\ t_{m-1}...t_1t_0$

    if match use $T^* = 1\ t_{m-1}...t_1\overline{t_0}$
- permutations similar - two passes
- broadcasting similar - need m+1 bit broadcast mask

One-to-One Destination Tags for the ESC Network ($X = 0$ or 1)

Fault Location	Destination Tag D^*
No fault	$D^* = X\, d_{m-1} \ldots d_1 d_0$
Stage 0 box	$D^* = d_0 d_{m-1} \ldots d_1 X$
Stage i box, $1 \leq i < m$ or any link	$D^* = s_0 d_{m-1} \ldots d_1 d_0$ if primary path is fault-free $D^* = \bar{s}_0 d_{m-1} \ldots d_1 d_0$ if primary path contains fault
Stage m box	$D^* = X\, d_{m-1} \ldots d_1 d_0$

TRY EXTRA Extra stage

$N=8$ $7 \to 1$

C_0 C_1 C_2 C_0

Primary 7 7 3 1 1

Sec 7 6 2 0 1

$1 \searrow$ 1 $\nearrow 5$

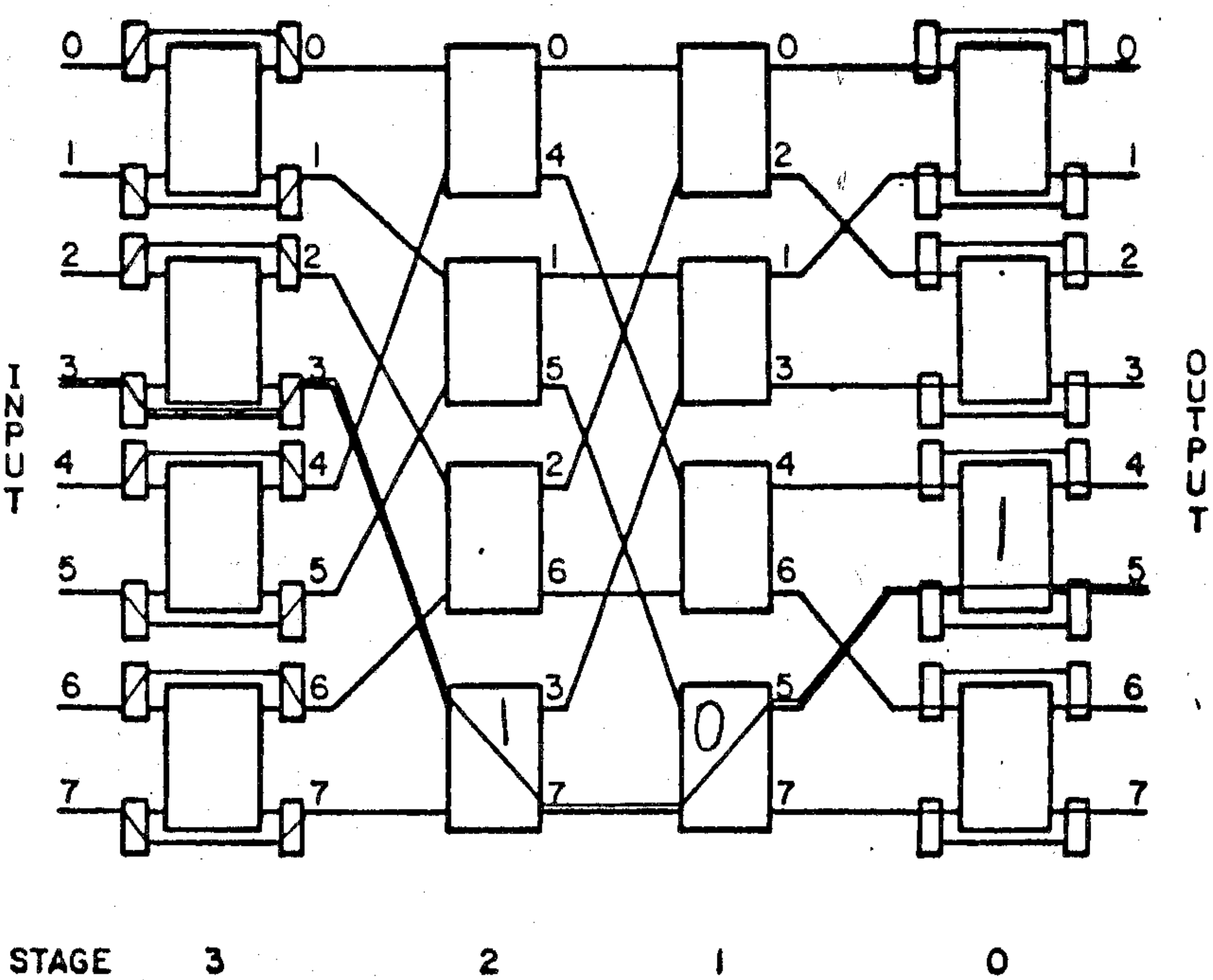
∴ Cannot handle two faults.

One-to-One Destination Tags for the ESC Network (X = 0 or 1)

$3 \longrightarrow 5$ $\qquad D = 101 = d_2 d_1 d_0$

Fault Location	Destination Tag D^*
No fault	$D^* = X\, d_{m-1} \ldots d_1 d_0 = X101$
Stage 0 box	$D^* = d_0 d_{m-1} \ldots d_1 X$
Stage i box, $1 \leq i < m$	$D^* = s_0 d_{m-1} \ldots d_1 d_0$
or any link	if primary path is fault-free
	$D^* = \bar{s}_0 d_{m-1} \ldots d_1 d_0$
	if primary path contains fault
Stage m box	$D^* = X d_{m-1} \ldots d_1 d_0 = X101$

disable $\qquad$ enable

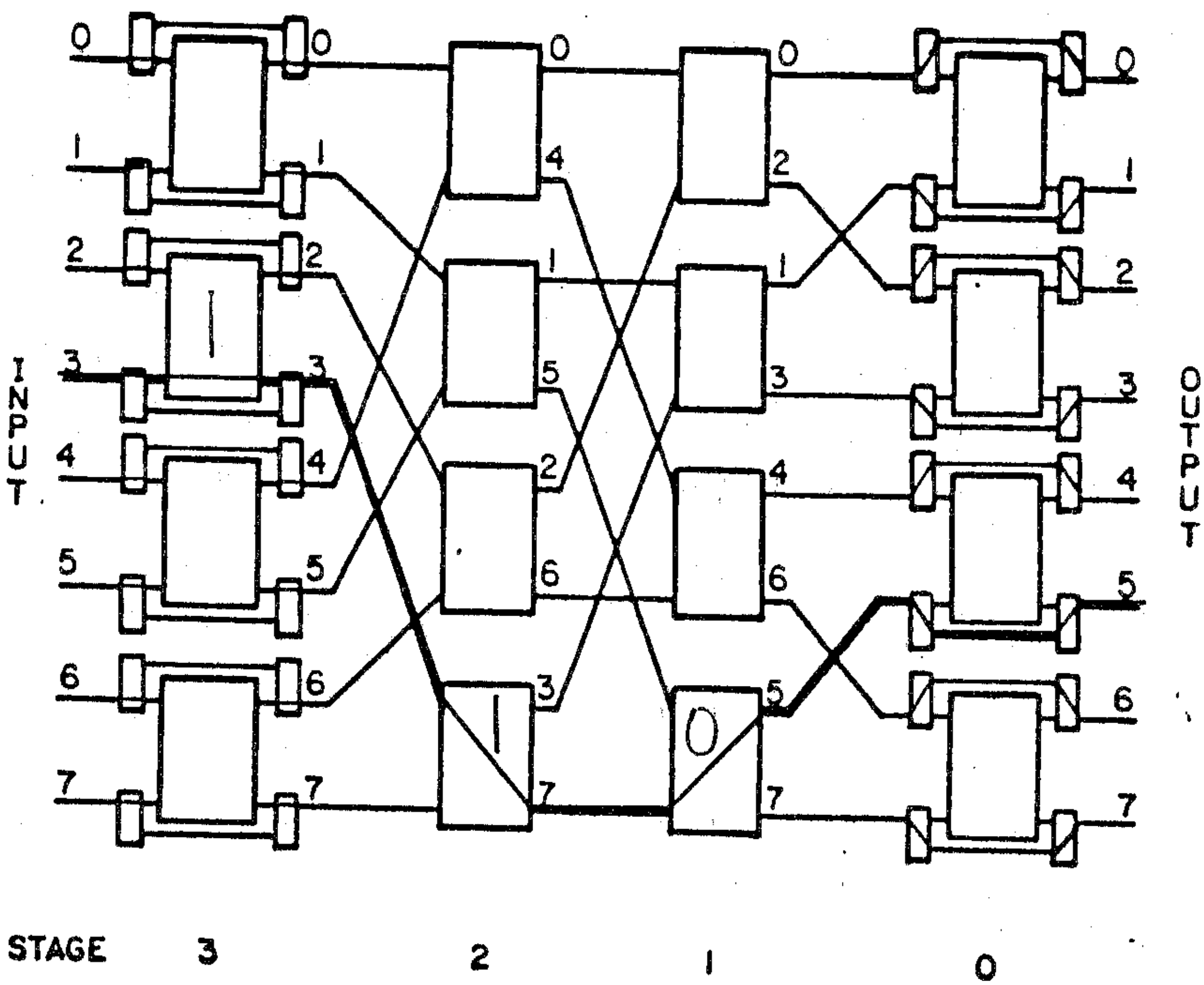


One-to-One Destination Tags for the ESC Network (X = 0 or 1)

3 → 5 $\quad$ $D = 101 = d_2 d_1 d_0$

Fault Location	Destination Tag D^*
No fault	$D^* = X\, d_{m-1} \ldots d_1 d_0$ $\quad d_0\, d_2\, d_1\, X$
Stage 0 box	$D^* = d_0 d_{m-1} \ldots d_1 X = 1\ 1\ 0\ X$
Stage i box, $1 \leq i < m$ or any link	$D^* = s_0 d_{m-1} \ldots d_1 d_0$ if primary path is fault-free; $D^* = \bar{s}_0 d_{m-1} \ldots d_1 d_0$ if primary path contains fault
Stage m box	$D^* = X d_{m-1} \ldots d_1 d_0$

enable $\qquad$ disable

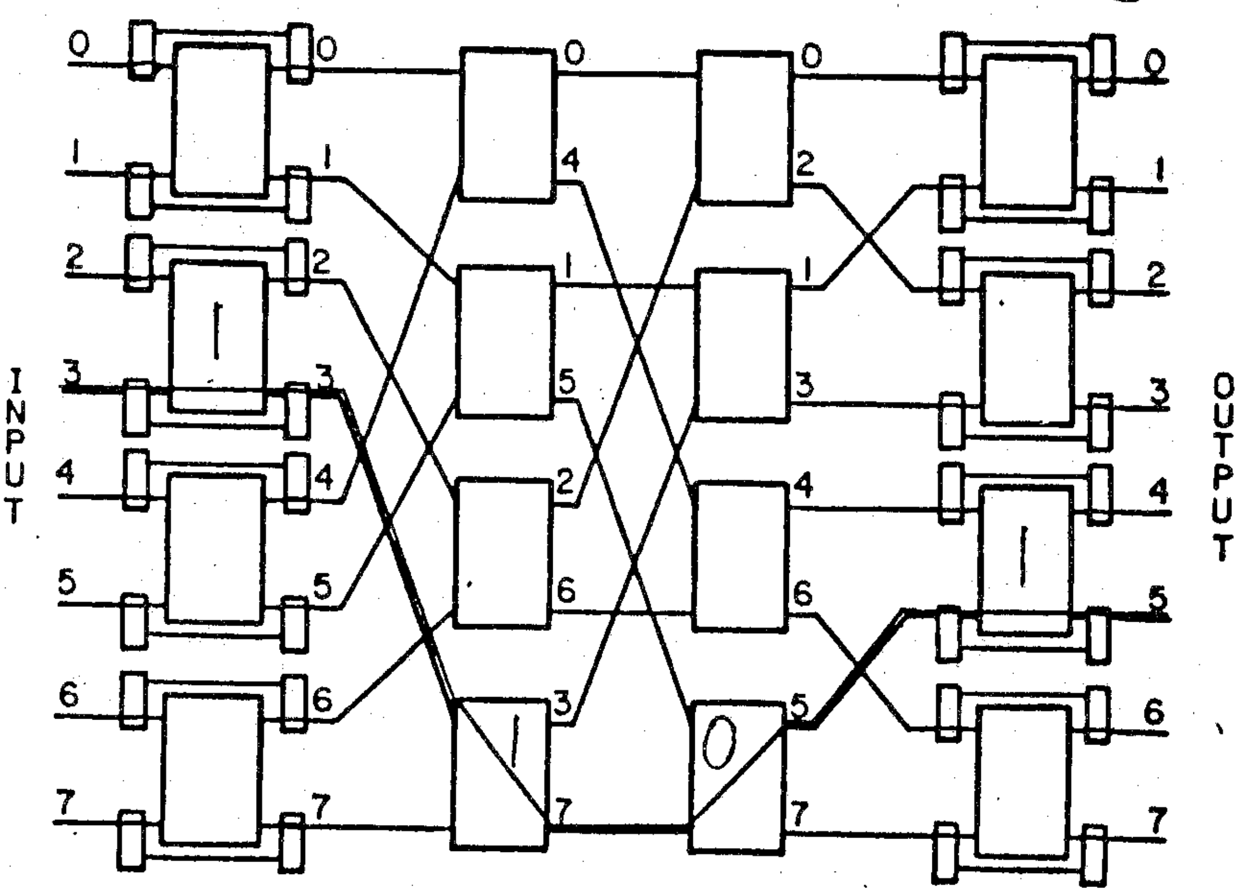


One-to-One Destination Tags for the ESC Network (X = 0 or 1)

3 → 5 $\qquad S = 011 = s_2 s_1 s_0 \quad D = 101 = d_2 d_1 d_0$

Fault Location	Destination Tag D^*
No fault	$D^* = X\, d_{m-1} \ldots d_1 d_0$
Stage 0 box	$D^* = d_0 d_{m-1} \ldots d_1 X$ $\quad s_0 d_2 d_1 d_0$
Stage i box, $1 \le i < m$ or any link	$D^* = s_0 d_{m-1} \ldots d_1 d_0 = 1\,1\,0\,1$ if primary path is fault-free
	$D^* = \bar{s}_0 d_{m-1} \ldots d_1 d_0$ if primary path contains fault
Stage m box	$D^* = X\, d_{m-1} \ldots d_1 d_0$

enable $\qquad\qquad\qquad\qquad$ enable

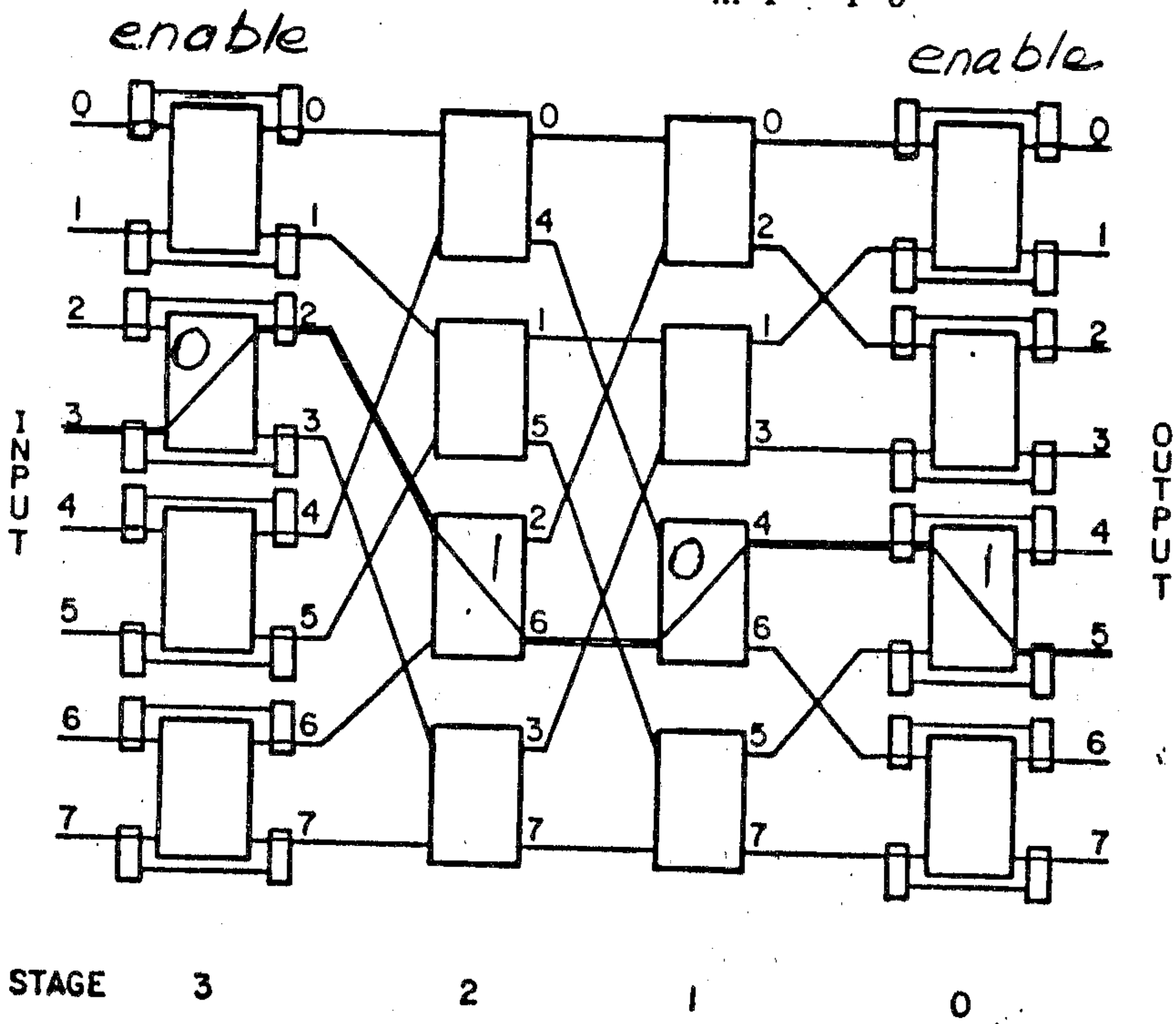


STAGE 3 2 1 0

One-to-One Destination Tags for the ESC Network (X = 0 or 1)

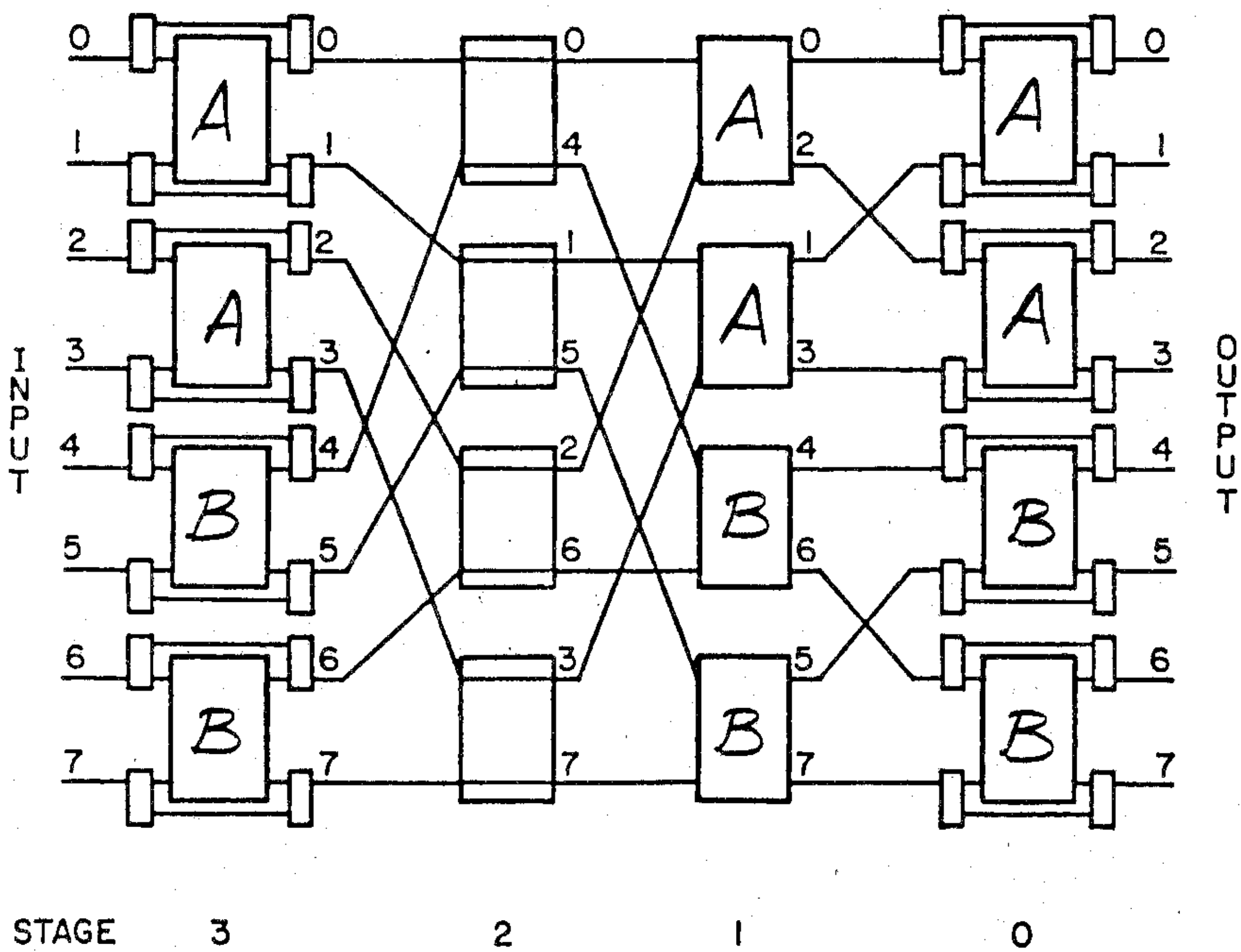
$3 \rightarrow 5 \quad S = 011 = s_2 s_1 s_0 \quad D = 101 = d_2 d_1 d_0$

Fault Location	Destination Tag D^*
No fault	$D^* = X\, d_{m-1} \ldots d_1 d_0$
Stage 0 box	$D^* = d_0 d_{m-1} \ldots d_1 X$
Stage i box, $1 \leq i < m$ or any link	$D^* = s_0 d_{m-1} \ldots d_1 d_0$ if primary path is fault-free $\bar{s}_0 d_2 d_1 d_0$
	$D^* = \bar{s}_0 d_{m-1} \ldots d_1 d_0$ = 0 1 0 1 if primary path contains fault
Stage m box	$D^* = X d_{m-1} \ldots d_1 d_0$



Extra Stage Cube Partitioning

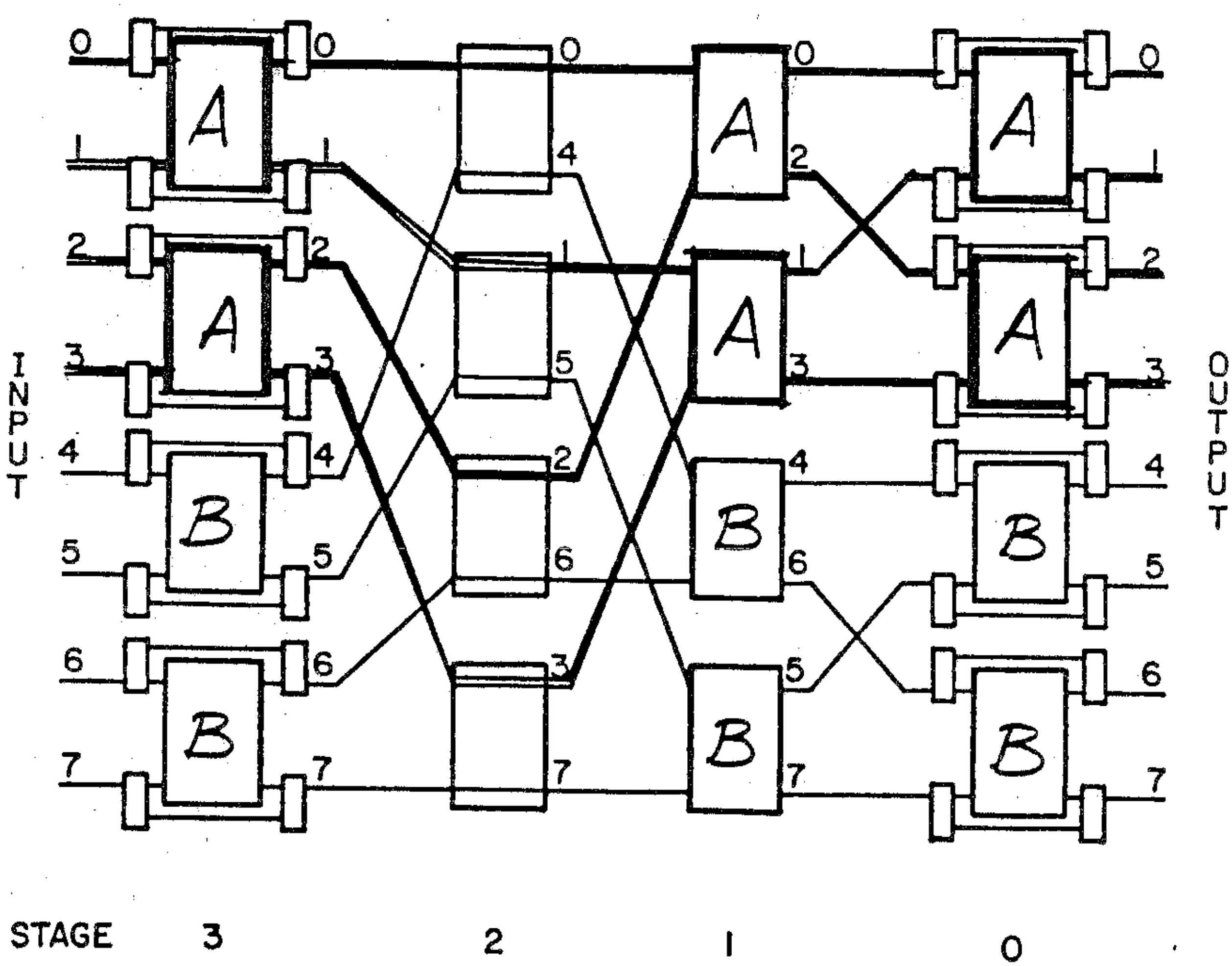
- similar to Generalized Cube
- example - Group A: 0-3   Group B: 4-7



Extra Stage Cube Partitioning

- similar to Generalized Cube
- example - Group A: 0-3 Group B: 4-7.

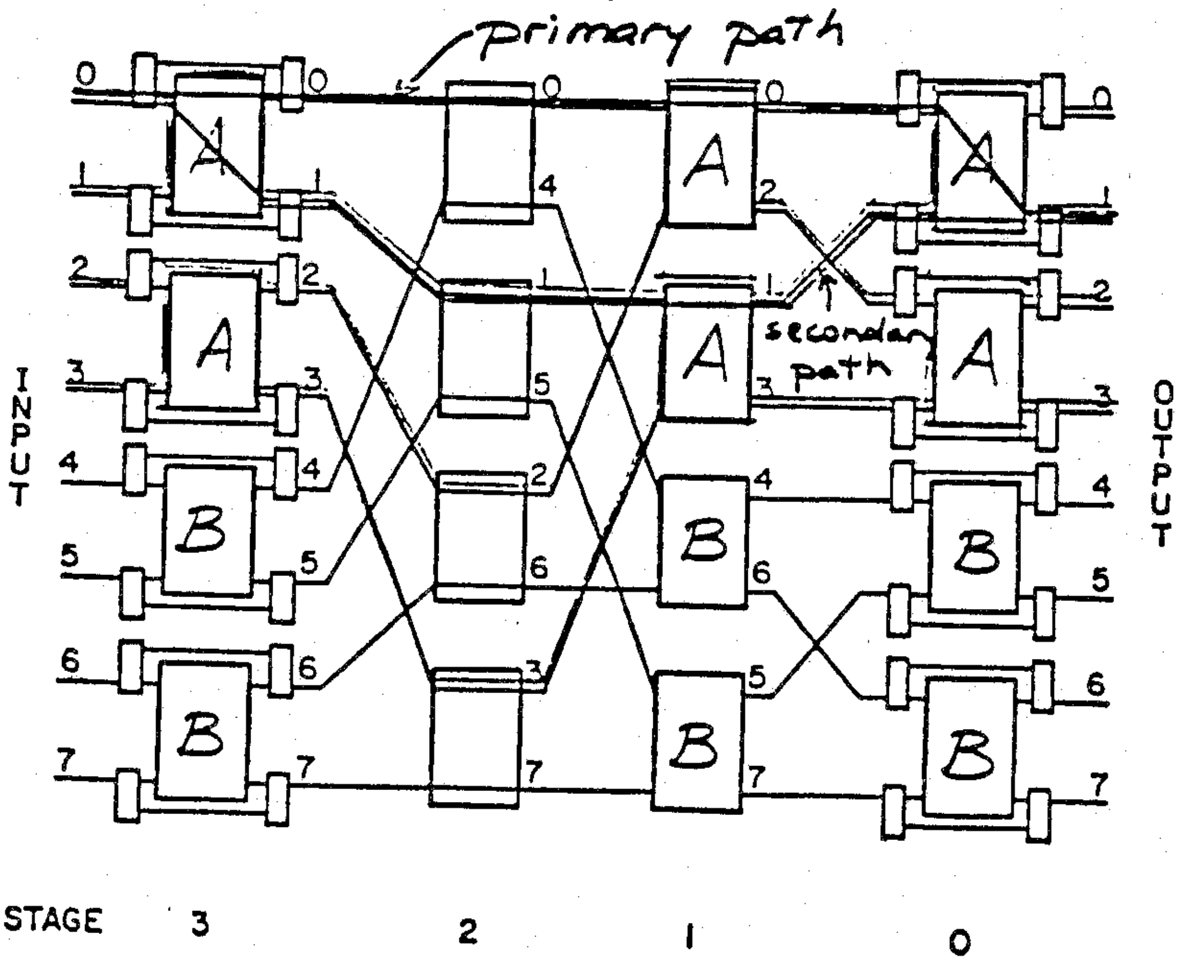
red shows independence of groups



Extra Stage Cube Partitioning

- similar to Generalized Cube
- example - Group A: 0-3   Group B: 4-7

red shows independence of groups
· primary and secondary paths exist within partition



Extra Stage Cube Partitioning

- same as Generalized Cube, except cannot partition on stage 0
- each subnetwork is independent ESC network
- each subnetwork is single fault tolerant (if boxes are bypassed individually)
- if box forced to straight is faulty, two partitions affected (single fault in each)

Multiple Faults

1. If one fault in stage 0 and one fault elsewhere, then some source/destination pairs not possible.

2. If one fault in stage m and one fault elsewhere, then some source/destination pairs not possible.

3. No faulty stage 0 or stage m boxes:
   Fault Labels $(a_{m-1} \cdots a_1 a_0, i)$
   $(b_{m-1} \cdots b_1 b_0, j)$ $1 \leq j \leq i < m$

   If $a_{m-1} \cdots a_i \neq b_{m-1} \cdots b_i$
   OR $a_{j-1} \cdots a_1 \overline{a}_0 \neq b_{j-1} \cdots b_1 b_0$ then there is a fault-free path for all source/destination pairs

   If both equal - no connection between these pairs:

   $$s_{i-1} \cdots s_1 = a_{i-1} \cdots a_1$$
   $$d_{m-1} \cdots d_j = b_{m-1} \cdots b_j$$

   $(s_{m-1} \cdots s_i, s_0, d_{j-1} \cdots d_0$ arbitrary$)$

   (system control unit checks this)

Multiple Fault Tolerance for ESC

- assume fault label A is from stage i,

  and fault label B is from stage j, $j \leq i$

- consider primary and secondary paths from S to D

- stage i output link: primary — $d_{m-1/i}s_{i-1/0}$ and secondary — $d_{m-1/i}s_{i-1/1}\overline{s}_0$

- stage j output link: primary — $d_{m-1/j}s_{j-1/0}$ and secondary — $d_{m-1/j}s_{j-1/1}\overline{s}_0$

- at stages i and j the primary and secondary paths both have $d_{m-1/i}$ and $s_{j-1/1}$ and are complements in bit position 0

- if $a_{m-1/i} \neq b_{m-1/i}$ both paths not blocked (since

  if $a_{m-1/i} = d_{m-1/i}$, then $b_{m-1/i} \neq d_{m-1/i}$ and vice versa)

- similar if $a_{j-1/1}\overline{a}_0 \neq b_{j-1/0}$

- only if $a_{m-1/i} = b_{m-1/i}$ and $a_{j-1/1}\overline{a}_0 = b_{j-1/0}$ are some S/D pairs blocked

- these S/D pairs are:

  $d_{m-1/j} = b_{m-1/j}$ ($\rightarrow d_{m-1/i} = a_{m-1/i}$)

  $s_{i-1/1} = a_{i-1/1}$ ($\rightarrow s_{j-1/1} = b_{j-1/1}$)

  $\rightarrow$ $d_{m-1/j}s_{j-1/1} = b_{m-1/1}$ and $d_{m-1/i}s_{i-1/1} = a_{m-1/1}$

  and $s_0 = a_0 = \overline{b}_0$ or $\overline{s}_0 = a_0 = \overline{b}_0$

- $s_{m-1/i}$ and $d_{j-1/0}$ and s_0 may vary: $2^{(m-i)+1+j}$ pairs

Consider probability that *two* arbitrary faults will cause loss of full functioning capability

2 box faults
2 link faults
1 box fault and 1 link fault

From:

"Modifications to Improve the Fault Tolerance of the Extra Stage Cube Interconnection Networks," Adams and Siegel, *1984 Int'l. Conf. Parallel Processing.*

NOT IN BOOK,

* BUT HE WILL HAND OUT PAPER *

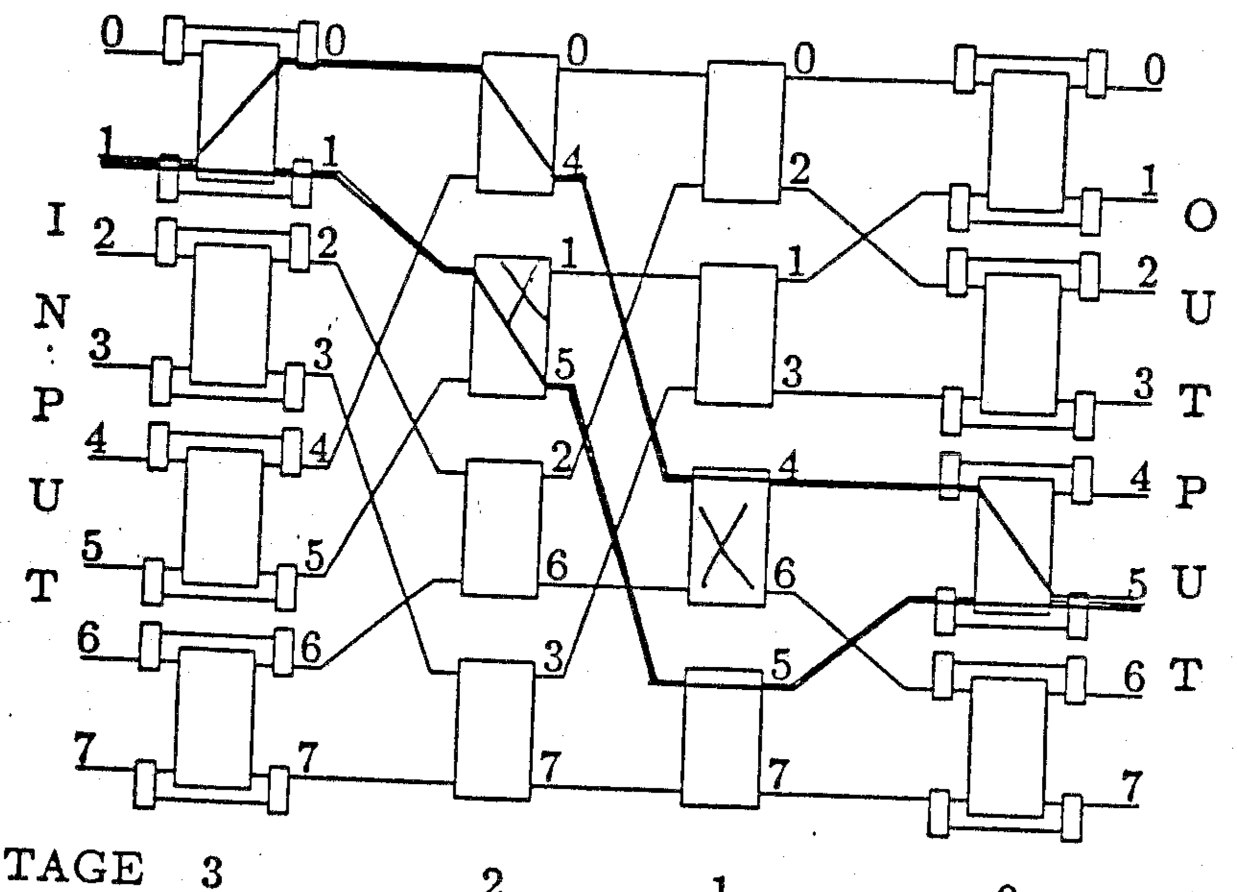
Consider probability that *two* arbitrary network faults will cause loss of full functioning capability

2 box faults

2 link faults

1 box fault and 1 link fault

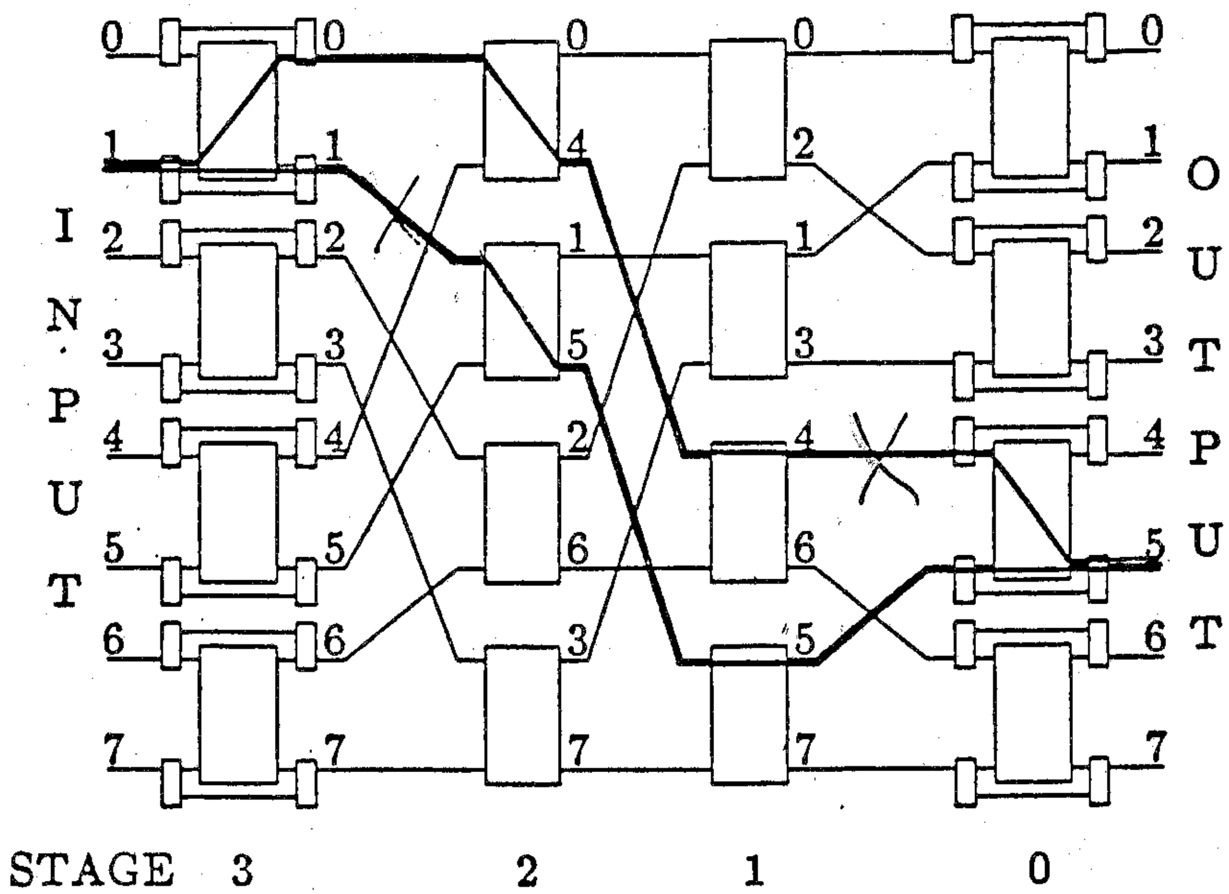
Must block both primary and secondary path for an input/output pair



2 faulty boxes

Consider probability that *two* arbitrary network faults will cause loss of full functioning capability

2 box faults
2 link faults
1 box fault and 1 link fault



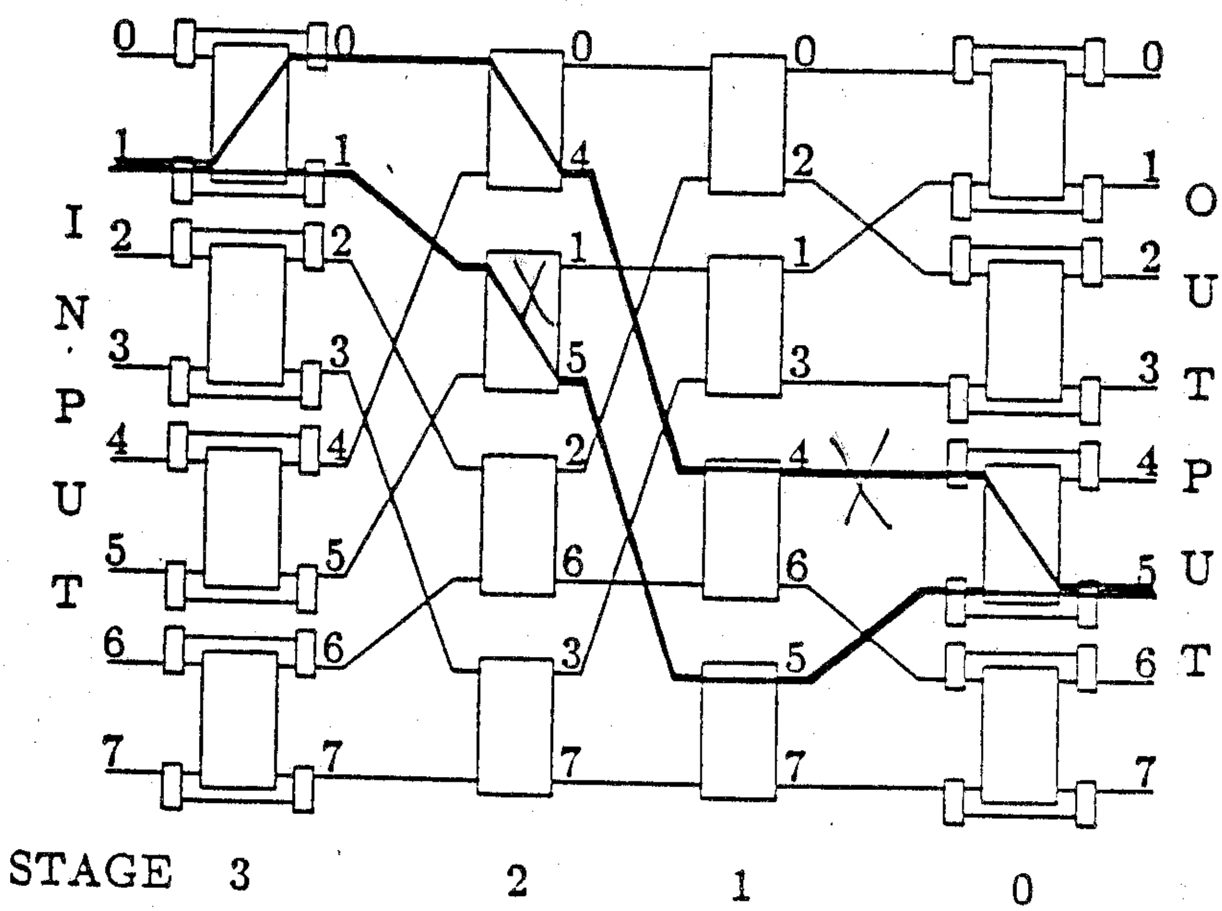
2 faulty links

Consider probability that *two* arbitrary network faults will cause loss of full functioning capability

2 box faults

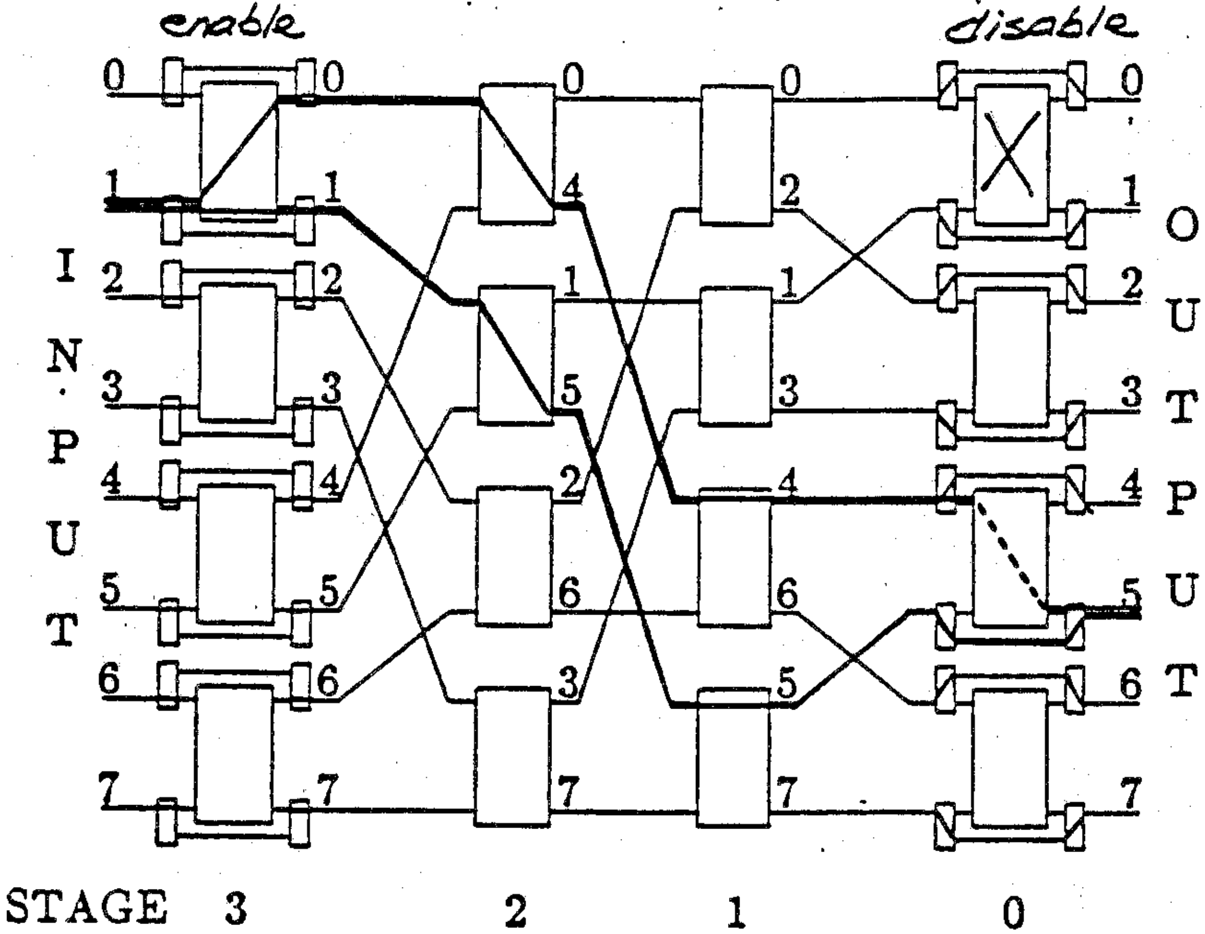
2 link faults

1 box fault and 1 link fault



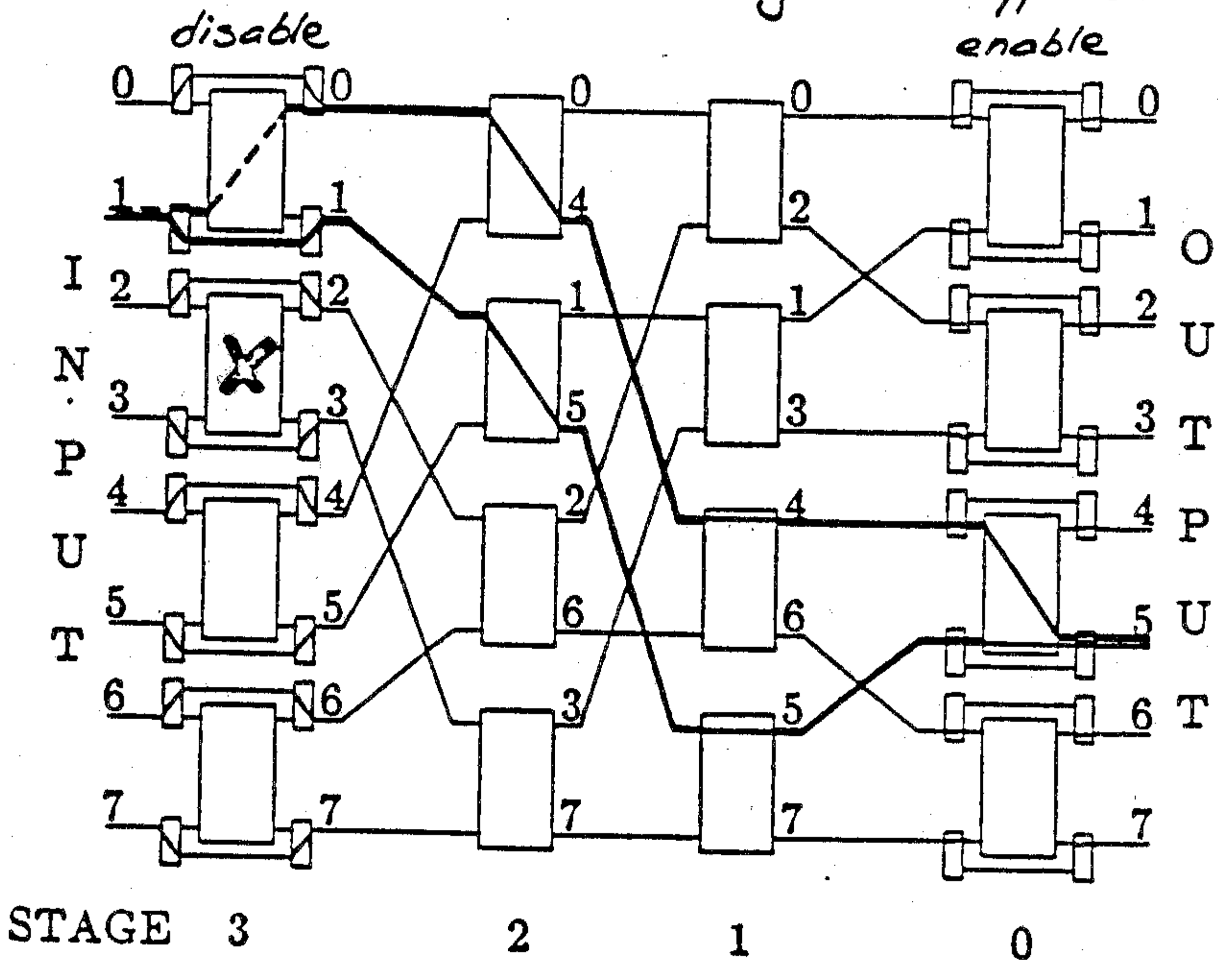
1 faulty box
and
1 faulty link

Stage bypassing - 1 stage 0 box fault, entire stage 0 bypassed



with stage 0 disabled, only one path between any source / destination
any other single fault prevents full functioning

Stage bypassing - 1 stage m box fault,
entire stage m bypassed



with stage m disabled, only one
path between any source/destination
any other single fault prevents
full functioning

Enhancement to ESC

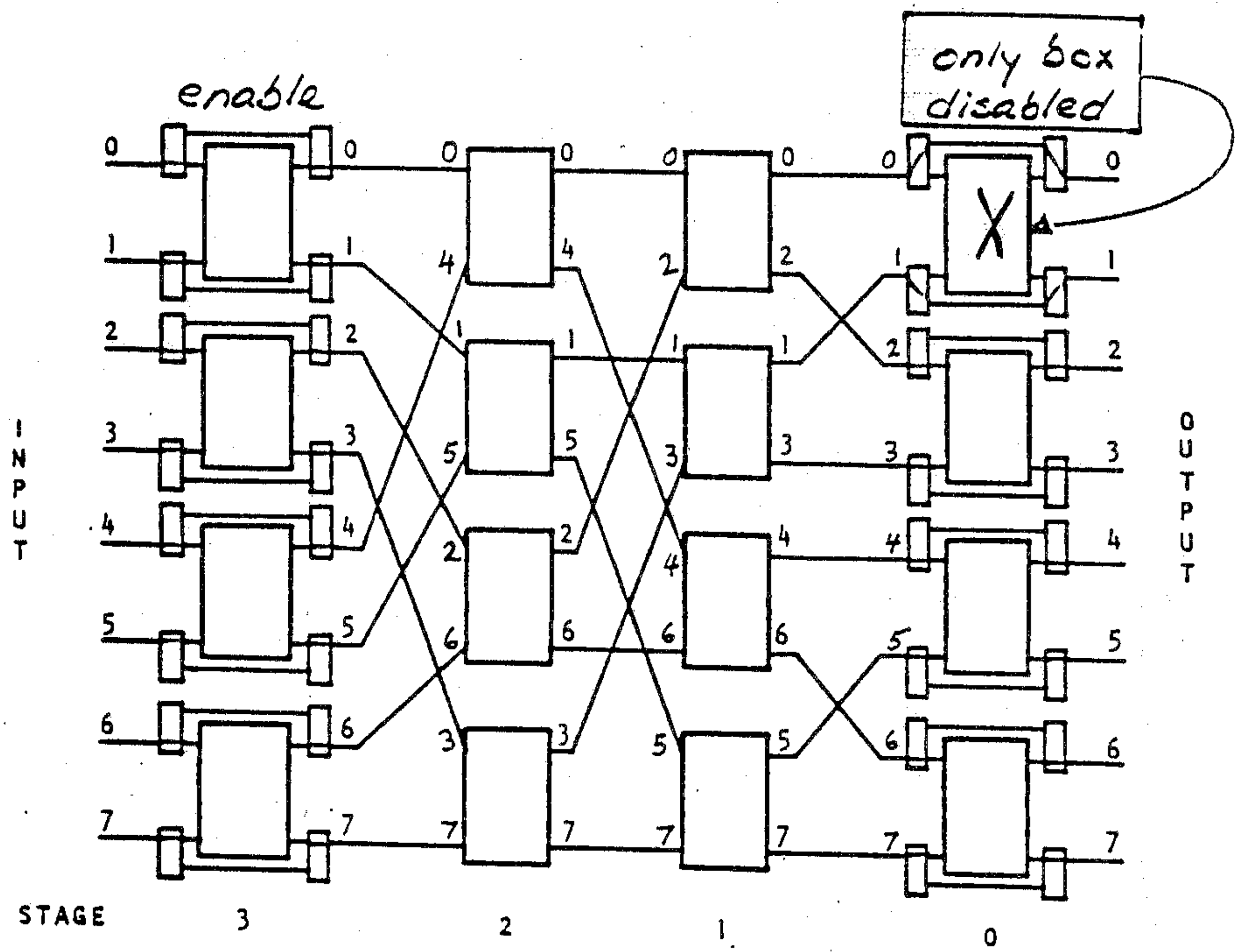
Box bypassing - (vs. stage bypassing)

Stage 0 faulty box -
- bypass (disable) only faulty box
  (not all of stage 0)
- enable all of stage m

Stage m faulty box -
- bypass (disable) only faulty box
  (not all of stage m)
- enable all of stage 0

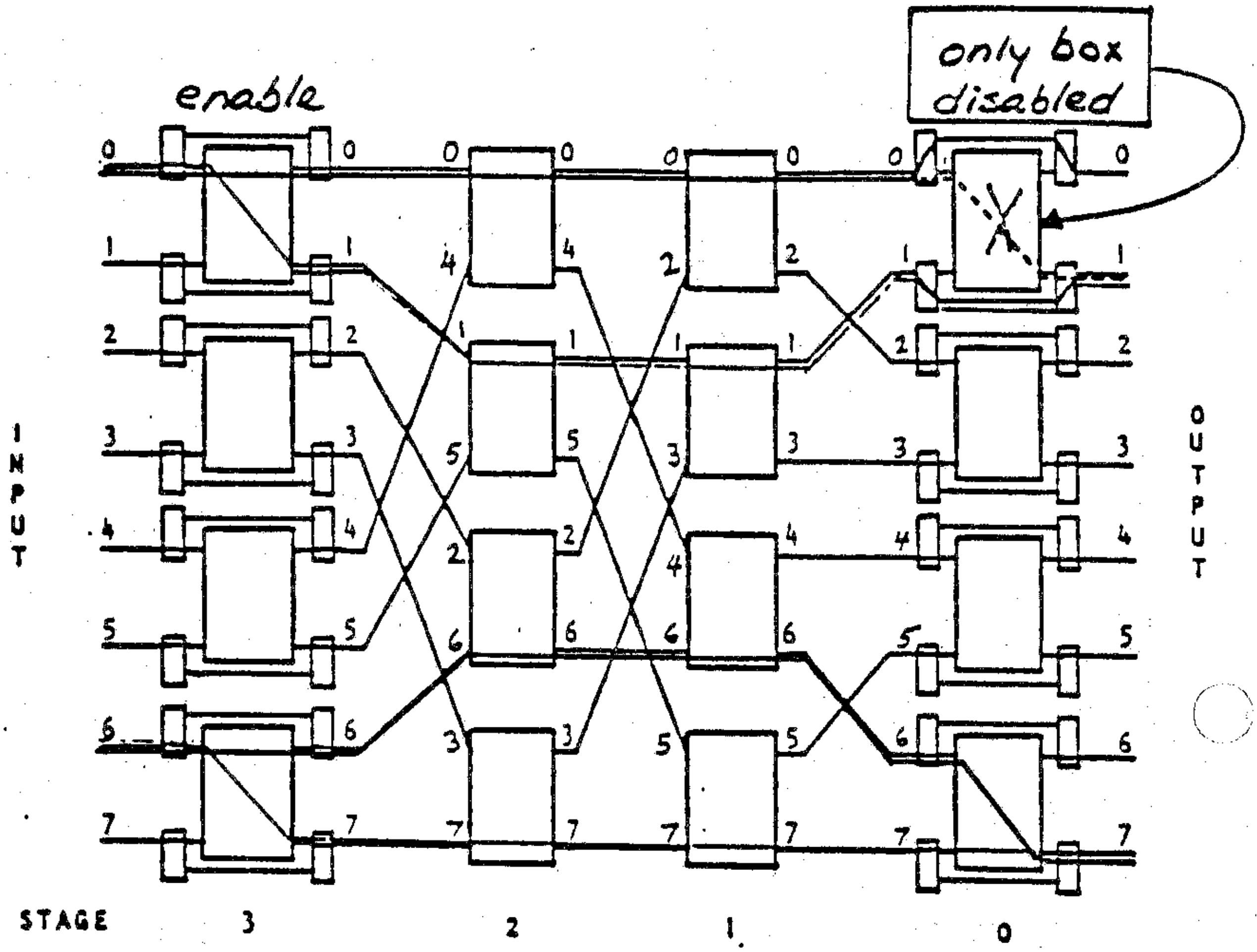
All other faults -
- handle same way as before

Box bypassing - Stage 0



only paths that need faulty box blocked

Box bypassing - Stage 0



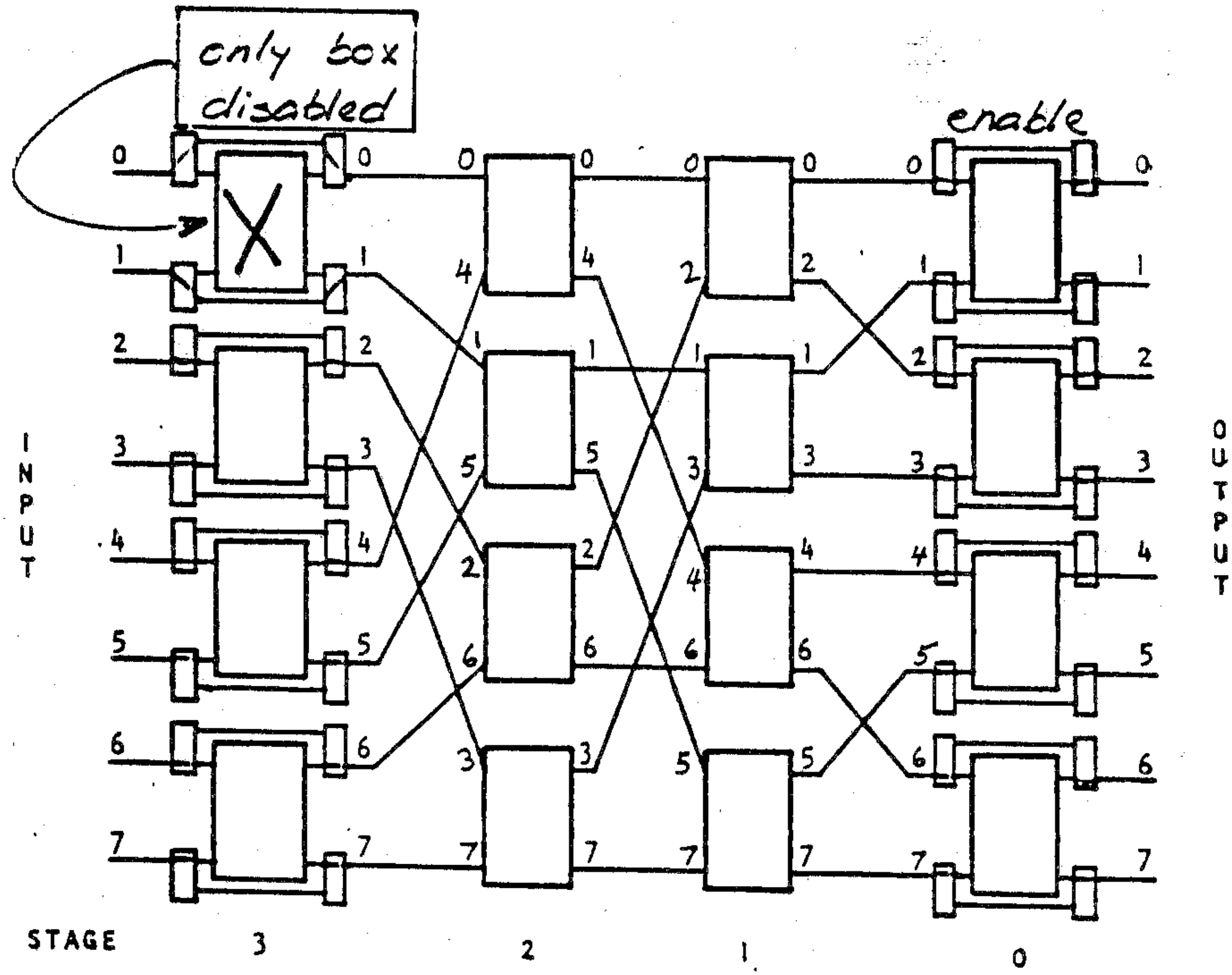
only paths that need faulty box blocked

Ex. one 0 → 1 path blocked

Ex. no 6 → 7 paths blocked (one would be by stage bypassing)

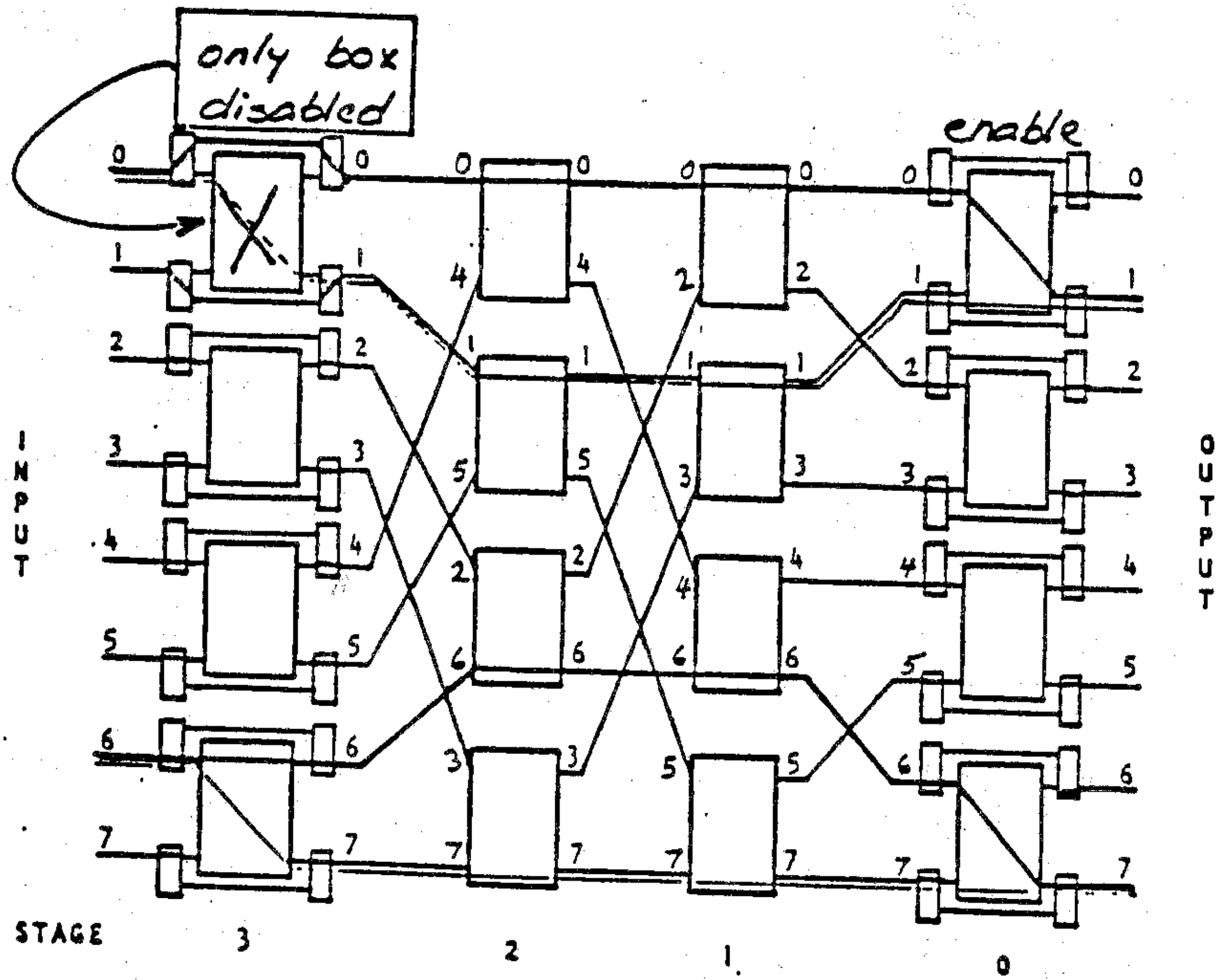
improves chance to survive multiple faults

Box bypassing - stage m



only paths that need faulty box blocked

Box bypassing - stage m



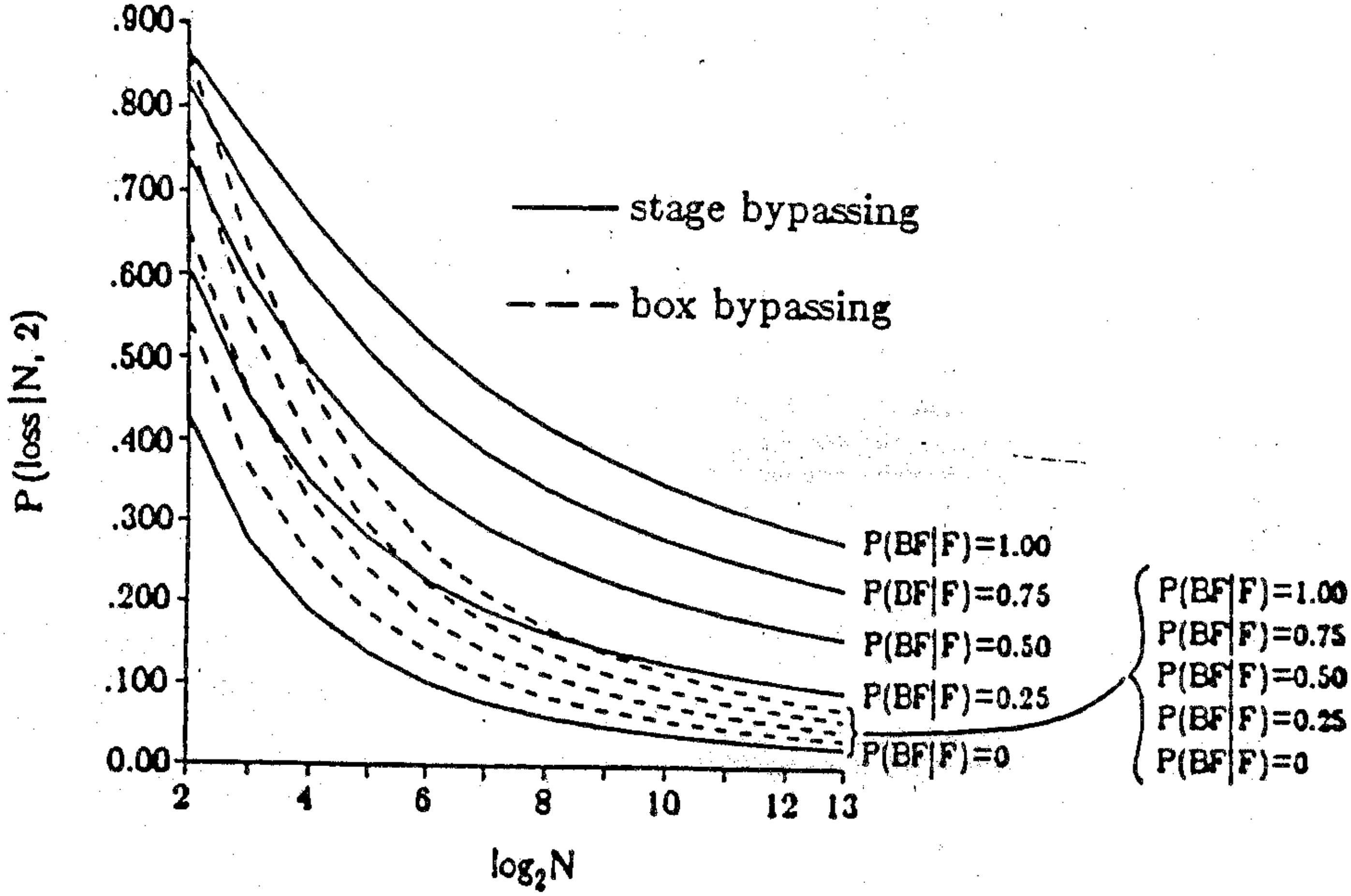
only paths that need faulty box blocked

Ex. one $0 \rightarrow 1$ path blocked
Ex. no $6 \rightarrow 7$ paths blocked
(one would be by stage bypassing)

improves chance to survive multiple faults

Probability of Loss of

Full Functioning Capability Given 2 Faults Occur



$P(BF \mid F) \equiv$ probability box fault, given a fault

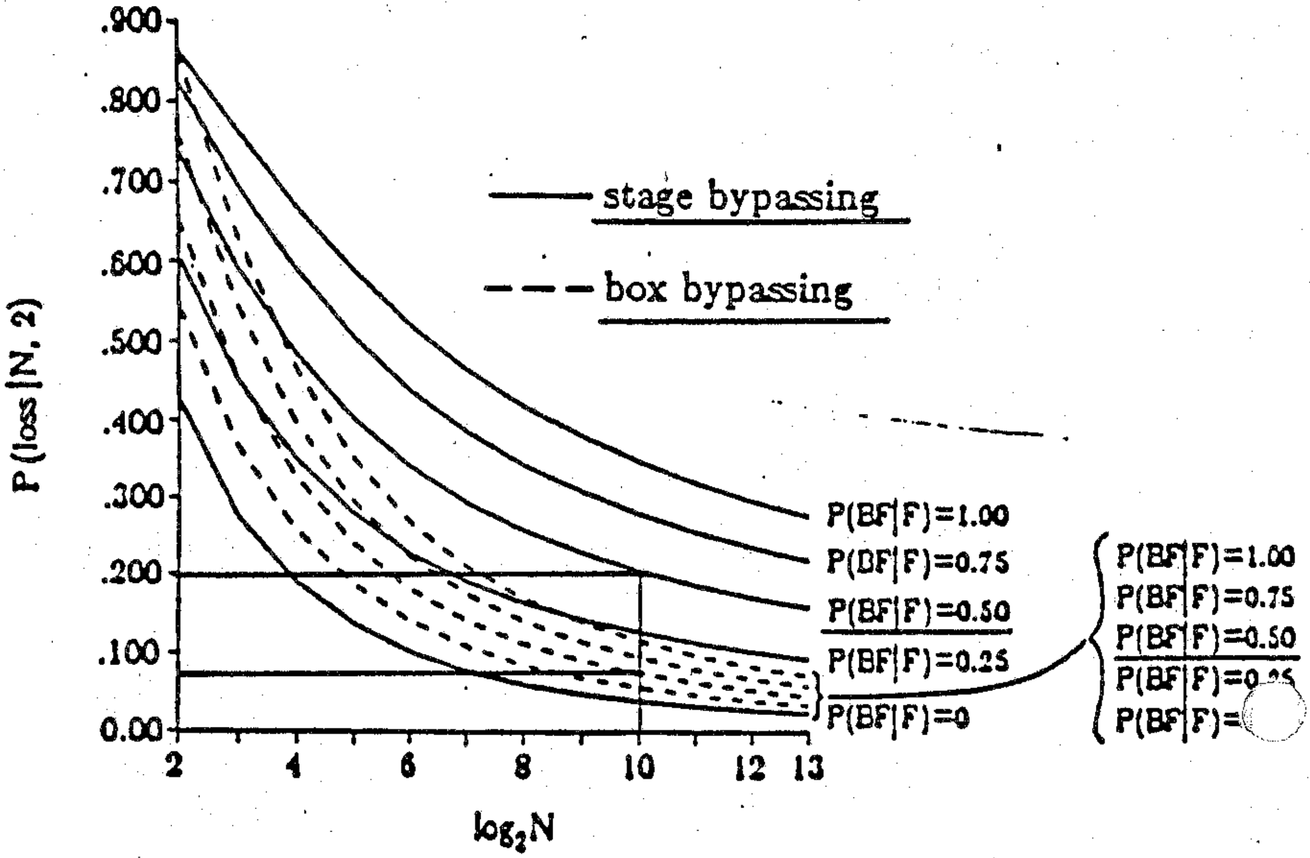
$P(LF \mid F) \equiv$ probability link fault, given a fault

$$P(LF \mid F) = 1 - P(BF \mid F)$$

$P(\text{loss} \mid N,2) \equiv$ probability loss of full functioning capability given 2 faults occur in a network of size N

Full Functioning Capability $\equiv$ can connect any input to any output

Probability of Loss of Full Functioning Capability Given 2 Faults Occur



P(BF | F) ≡ probability box fault, given a fault

P(LF | F) ≡ probability link fault, given a fault

$$P(LF \mid F) = 1 - P(BF \mid F)$$

P(loss | N,2) ≡ probability loss of full functioning capability given 2 faults occur in a network of size N

Full Functioning Capability ≡ can connect any input to any output

Analysis:

Given two faults and stage bypassing

$$P(\text{loss}|N,2) = \left[\frac{(4Nm-2N)+(4N-6m-2)}{N(m+1)^2-2(m-1)}\right] * P^2(BF|F) +$$
$$\left[\frac{2Nm+(4N-4m-4)}{Nn^2+Nm}\right] * 2 * P(BF|F) * P(LF|F) +$$
$$\left[\frac{4N-3m-4}{Nm^2-m}\right] * P^2(LF|F)$$

Given two faults and box bypassing

$$P(\text{loss}|N,2) = \left[\frac{14N-6m-18}{N(m+1)^2-2(m-1)}\right] * P^2(BF|F) +$$
$$\left[\frac{8N-4m-8}{Nm^2+Nm}\right] * 2 * P(BF|F) * P(LF|F) +$$
$$\left[\frac{4N-3m-4}{Nm^2-m}\right] * P^2(LF|F)$$

Verified by simulation for N = 4, 8, 16, and 32

Extra Stage Cube Advantages

1. all advantages of multistage cube
   - distributed network control using routing tags
   - partitionable into independent subnetworks
   - one device can broadcast to all or subset
   - can use for SIMD in addition to MIMD
   - variety of implementation options

2. single fault tolerant
   (1 to 1, broadcasts, 2 passes for permutations)

3. each partition single fault tolerant (box bypassing)

4. robust for 2 faults (box bypassing)
   (any S to any D ~90%)

5. when multiple faults occur
   — degradation (if any) determinable:
   amount and which S/D pairs affected