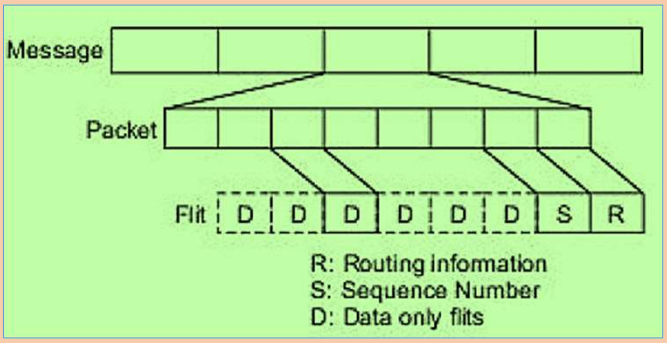
MODULE 4

MESSAGE PASSING MECHANISMS

INTRODUCTION

- Message passing in a multicomputer network needs special hardware and software
- A message is a logical unit of internode communication
- It is assembled from fixed length packets

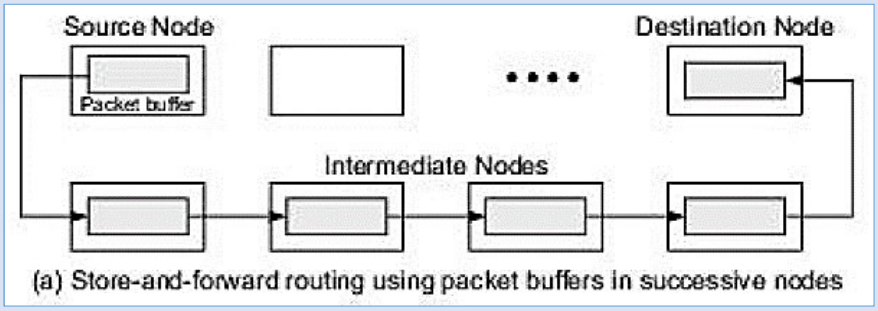


- A packet is the basic unit containing the destination address for routing purposes
- A sequence number is needed in each packet to allow reassembly of the message transmitted.
- A packet can be Further divided into a number of fixed-length flits(flow control digits).
- Routing information(destination) and sequence number occupy the header flits. The remaining flits are the data elements of a packet.
- Typical packet lengths range from 64 to 512 bits. The sequence number may occupy one to two flits depending on the message length.

MESSAGE ROUTING SCHEMES

1. Store- and - Forward Routing

- Packets are the basic unit of information transformation. Each node uses packet buffer
- A packet is transmitted from a source node to a destination node through a sequence of intermediate nodes
- The latency is directly proportional to the distance between the source and the destination (number of hops)

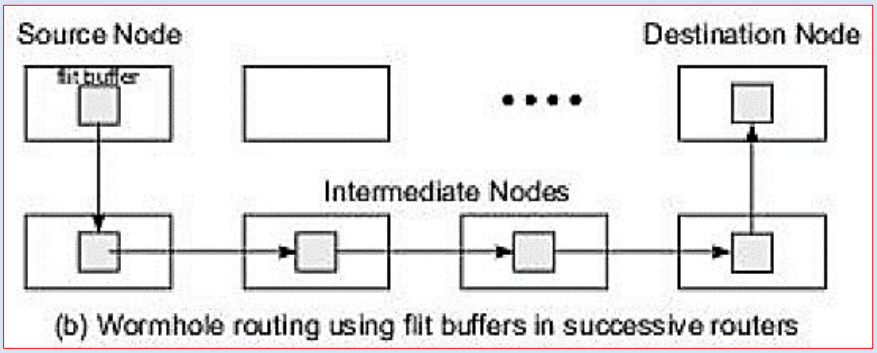


(a) Store-and-forward routing using packet buffers in successive nodes

2. Wormhole Routing

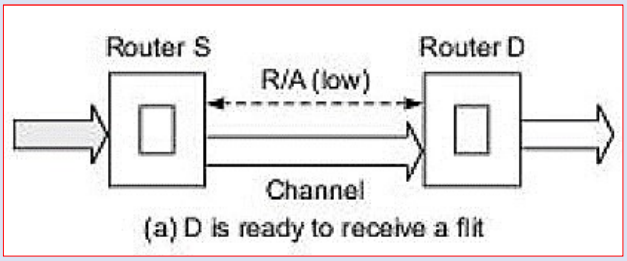
- Packets are subdivided in to flits. Flit buffers are used in the hardware routers attached to nodes
- The transmission from the source node to the destination node is done through a sequence of routers

- All the flits in the same packet are transmitted in order as inseparable companions in a pipelined fashion.
- Only the header flits knows where the packet is going. All the data flits must follow the header flit. The flits from different packet cannot be mixed up.



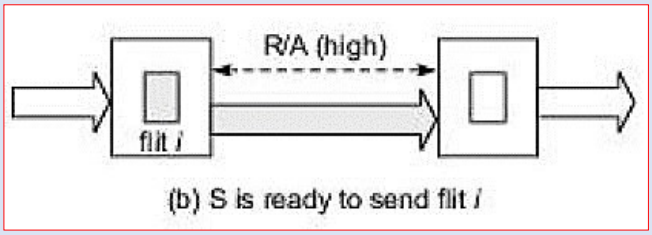
(b) Wormhole routing using flit buffers in successive routers

- The pipelining of successive flits in a packet is done asynchronously using a handshaking protocol
- A path, (a 1-bit ready/request(R/A))line is used between adjacent routers.
- When the receiving router D is ready to receive a flit, it pulls the R/A line low



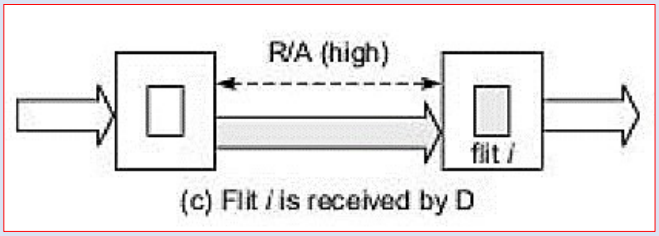
(a) D is ready to receive a flit

- When the sending router S is ready , it raises the line high and transmits flit i through the channel.



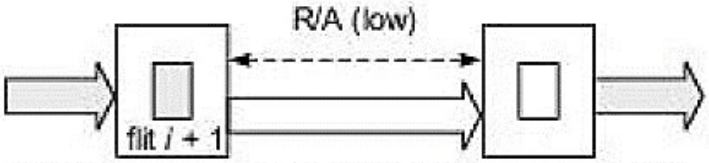
(b) S is ready to send flit *i*

- While the flit is being received by D, the R/A line is kept high.



(c) Flit *i* is received by D

- After flit i is removed from D's buffer the cycle repeats itself for the transmission of the next flit i + 1 until the entire packet is transmitted.



(d) Flit *i* is removed from D's buffer and flit *i* + 1 arrives at S's buffer

The communication latency for a store-and-forward network is expressed by

$$T_{SF} = \frac{L}{W}(D+1)$$

The latency for a wormhole-routed network is expressed by

$$T_{WH} = \frac{L}{W} + \frac{F}{W} \times D$$

Where

L – Packet length

W- channel bandwidth

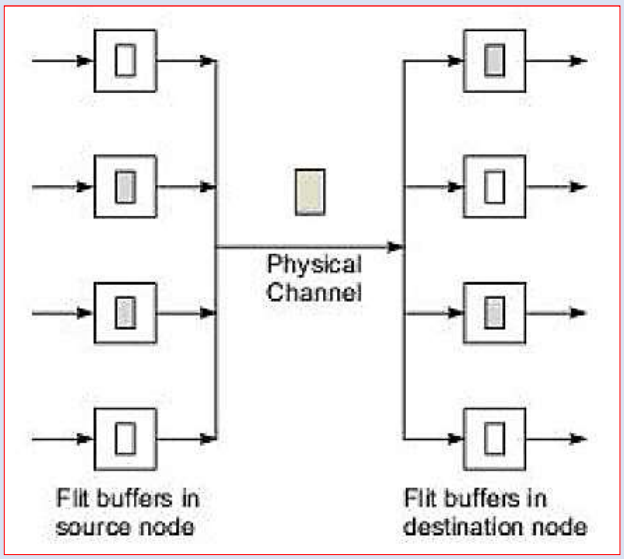
D – distance

F- flit length (in bits)

❖Virtual Channel

- In wormhole routing, the physical channel is shared between nodes. Here a virtual channel is a logical link between two nodes.
- It is formed by a flit buffer in the source node, a physical channel between them, and a flit buffer in the receiver node
- One source buffer is paired with one receiver buffer to form a virtual channel when the physical channel is allocated for the pair.
- The sharing of a physical channel by a set of virtual channels is conducted by time-multiplexing on a flit-by-flit basis

❖Virtual Channel



FLOW CONTROL STRATEGIES

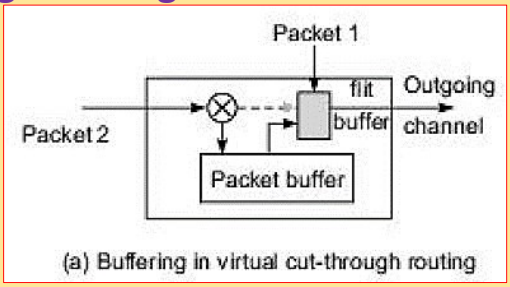
- Flow control strategies are developed to control smooth network traffic flow without causing congestion or deadlock situations.

Packet Collision Resolution

- To move a flit between adjacent nodes in a pipeline of channels, three elements must he present:
  - Source buffer holding the flit
  - The channel being allocated
  - The receiver buffer accepting the flit.
- When two packets reach the same node, they may request the same receiver buffer or the same outgoing channel

- There are four methods for resolving the conflict between two packets competing for the use of The same outgoing channel

1. **virtual cut-through routing**



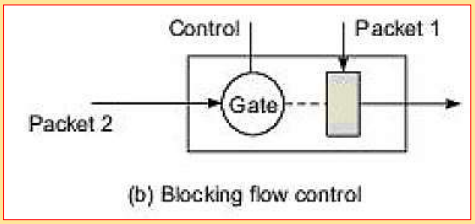
(a) Buffering in virtual cut-through routing

- Packet 2 is temporarily stored in a packet buffer. When the channel becomes available later, it will be transmitted then
- One advantage is that it does not waste the resource already allocated, but it need a large buffer to hold the entire packet.

- The virtual cut-through method combines the store-and-forward and wormhole rooting schemes
- When collisions do not occur, it perform Wormhole routing , otherwise store and forward routing.

2. Blocking Flow Control

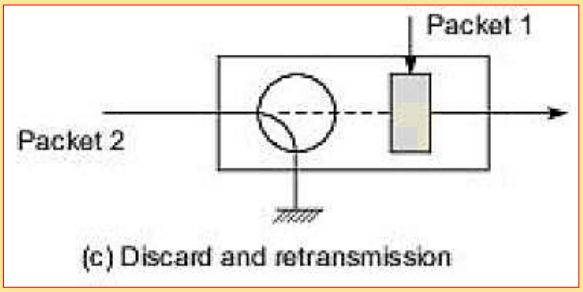
- Pure wormhole routing uses a blocking policy in case of packet collision. The second packet is being blocked from advancing; however, it is not being abandoned.



(b) Blocking flow control

3. Discard and Retransmission

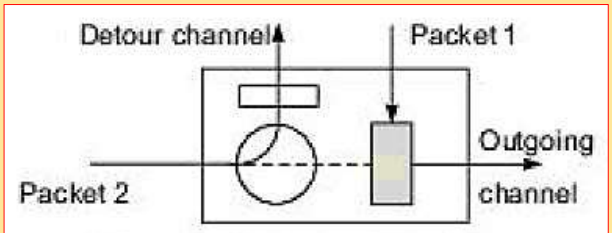
- It simply drops the packet being blocked from passing through.
- The discard policy may result in a severe waste of resources and it demands packet retransmission and acknowledgment.
- This policy is rarely used now



(c) Discard and retransmission

4. Detour

- The blocked packet is routed to a detour channel.
- It may waste more channel resources



(d) Detour after being blocked

UNICAST (ONE – TO – ONE) ROUTING

The two main unicast packet routing algorithms are

➢ Deterministic Routing Algorithm

- Here routing path is uniquely predetermined in advance

➢ Adaptive Routing Algorithm

- It depends on network conditions and alternate paths are possible

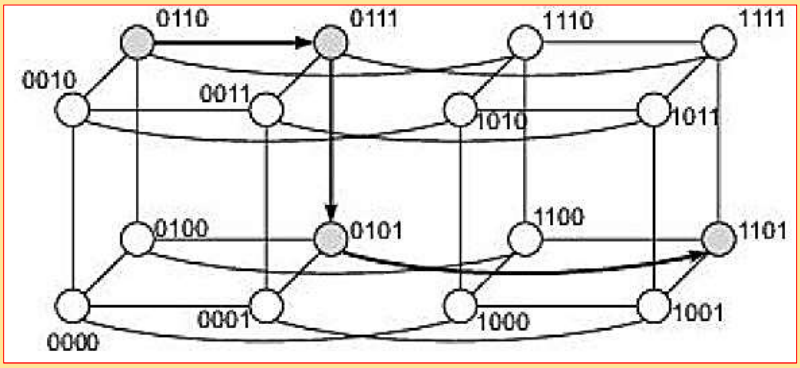
❖Deterministic Routing Algorithm

- These algorithms are based on a concept called "dimension order routing"
- In dimension order routing, the selection of successive channels follow a specific order

1. E-cube Routing on Hypercube

- Hyper cube is a binary n-cube architecture. An n-cube consist of N = 2^n nodes
- We want to determine a route from **s** to d with a minimum number of steps.

Eg:



n = 4 (4 dimensional Hypercube)

s = 0110

d = 1101

(a) Find direction bit " r "

Truth Table		
B	A	Q
0	0	0
0	1	1
1	0	1
1	1	0

$s_4\ s_3\ s_2\ s_1$

s = 0 1 1 0

$d_4\ d_3\ d_2\ d_1$

d = 1 1 0 1

r = 0 1 1 0 ⊕ (XOR)

1 1 0 1

1 0 1 1

$r_4\ r_3\ r_2\ r_1$

1 0 1 1

(b) Find route from s (current node) to next node (v)

Next node $v = s \oplus 2^0$

s → 0 1 1 0 ⊕

2^0 → 0 0 0 1

v → 0 1 1 1

Prepared By Mr. EBIN PM, AP, IESCE EDULINE 21

➢ We can take 0111, because r_1 = 1. If the value is 0, just discard the node.

To find the next node from $v = v \oplus 2^1$

v → 0 1 1 1 ⊕

2^1 → 0 0 1 0

new v is 0 1 0 1

$r_4\ r_3\ r_2\ r_1$

1 0 1 1

We can take 0101 node , because r_2 = 1

The value of r_3 =0. So we can discard the dimension 3

Prepared By Mr. EBIN PM, AP, IESCE EDULINE 22

To find the next node from new v = v ⊕ 2^3

new v → 0 1 0 1 ⊕

2^3 → 1 0 0 0

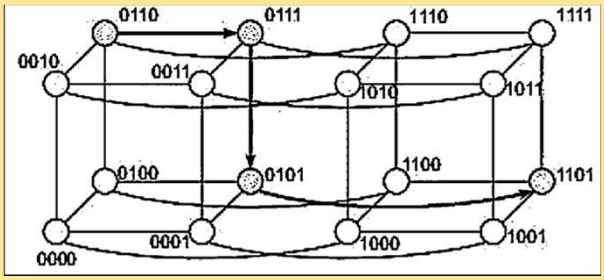
new v is 1 1 0 1

r_4	r_3	r_2	r_1
1	0	1	1

We can take 1101 node , because r_4 = 1

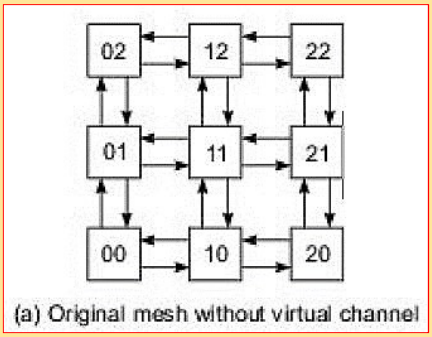
So the route is

0110 → 0111 → 0101 → 1101

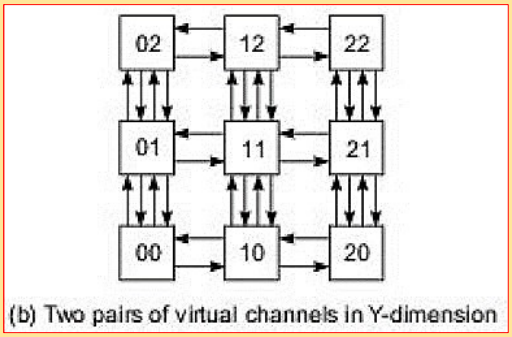


❖Adaptive Routing Algorithm

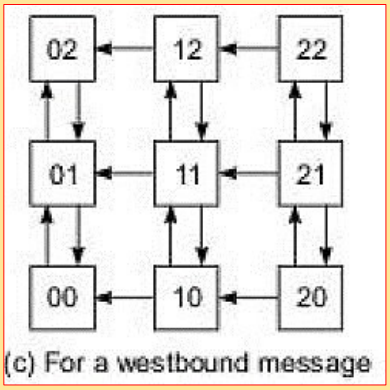
- The purpose of adaptive routing algorithm is to achieve efficiency and avoid deadlock.
- It uses the concept of virtual channels. The idea is to provide virtual channels in all connections along the same dimension of a mesh connected network.



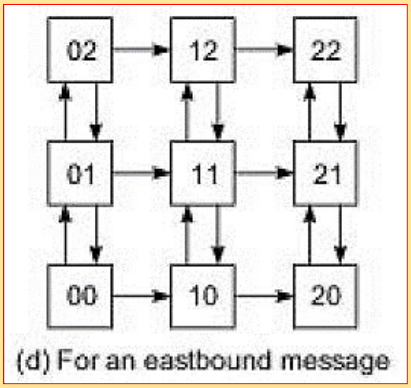
(a) Original mesh without virtual channel



(b) Two pairs of virtual channels in Y-dimension



(c) For a westbound message



(d) For an eastbound message

➢For westbound traffic, the virtual network can be used to avoid deadlock because all east bound X-channels are not in use.

MULTICAST (ONE-TO-MANY) ROUTING

➢The 4 types of communication patterns in multi cast computer networks are

- One to one unicast pattern with one source and one destination
- Multicast pattern (one to many communication) in which one source send the same message to multiple destinations.
- Broadcast pattern of one to all communication
- Many –to-many conference communication

Routing Efficiency

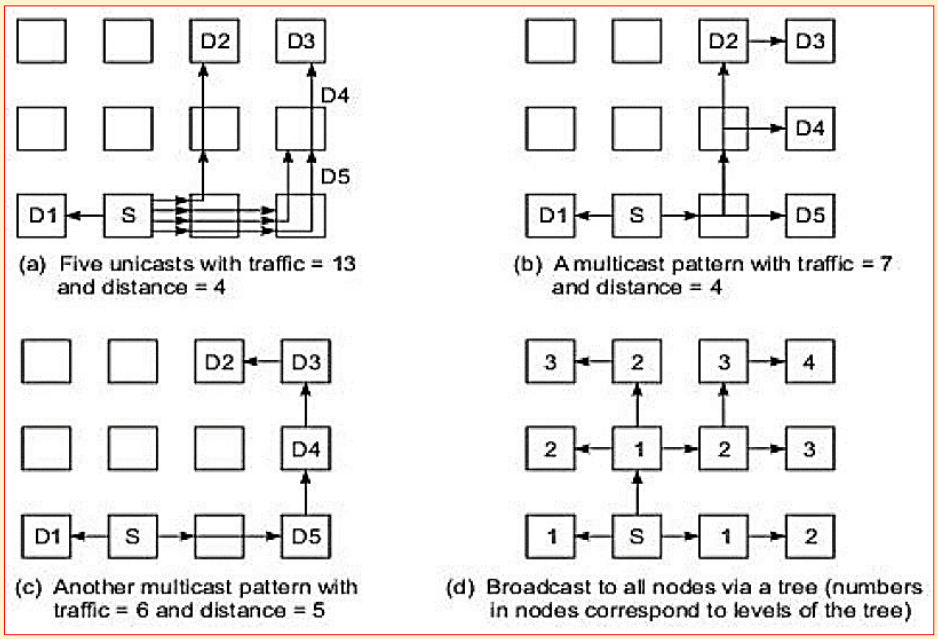
➢Two commonly used efficiency parameters are channel bandwidth and communication latency

- The channel bandwidth at any time instant indicates the effective data transmission rate achieved to deliver the message.
- The latency means packet transmission delay
- Latency is the most important issue in a store – and – forward network

❖Multicast and broadcast on a mesh-connected computer

- The following figure shows multicast routing implementation on a 3×4 mesh computer. "S" is source and "D" is destination.
- In figure(a), the x-y routing traffic requires the use of 1+3+4+3+2=13 channels and the latency is 4 for the longest path leading to D_3

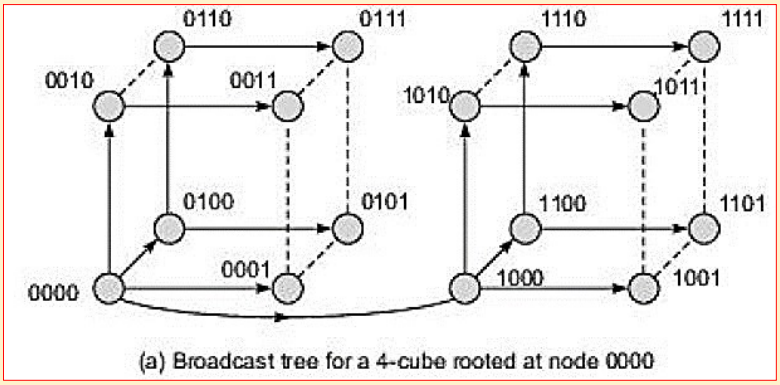
Fig:



- In figure(b) and (c) , multicast can be implemented by replicating the packet at an intermediate node.
- On a wormhole routed network, Fig(c) is better, and for a store-and-forward network, fig(b) is better (shorter latency).
- In Fig(d), 4-level spanning tree is used to broadcast a packet. Nodes reached at level "i" of the tree have latency "i". This tree have minimum latency and minimum traffic.

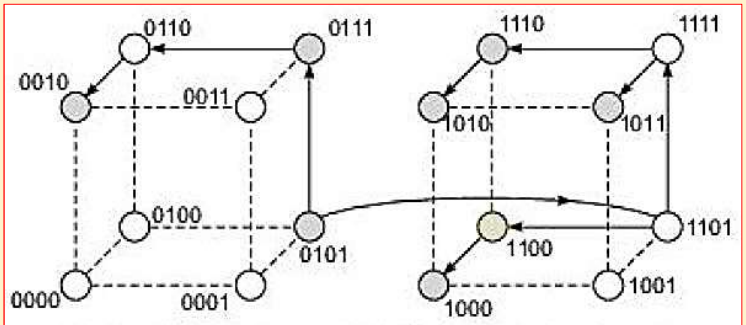
❖Multicast and broadcast on a hypercube computer

- In Fig(a), to broadcast on an n-cube, a spanning tree is used to reach all nodes with in a latency of 'n'. The root node is 0000

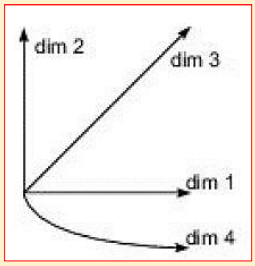


(a) Broadcast tree for a 4-cube rooted at node 0000

- In Fig(b), a greedy multicast tree is shown. Greedy multicast algorithm send the packet through the dimensions which can reach the most number of remaining destinations.



(b) A multicast tree from node 0101 to seven destination nodes 1100, 0111, 1010, 1110, 1011, 1000, and 0010



Prepared By Mr. EBIN PM, AP, IESCE EDULINE 31

- From source node S = 0101, there are two destinations via dimension 2, and 5 via dimension 4

  ie, 0101→0111

  0101→1101

- From node 1101, there are 3 destinations via dimension 2, and 4 via dimension 1

  ie, 1101→1111

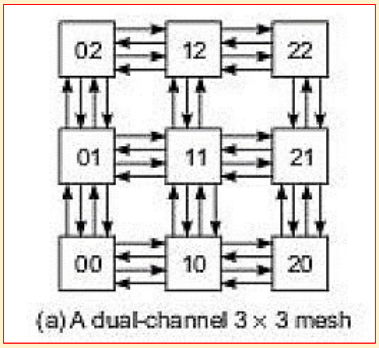
  1101→1100 & 0111→0110

- Before selecting certain dimensions, we should compare the reachability to obtain minimum coverset for the remaining nodes.

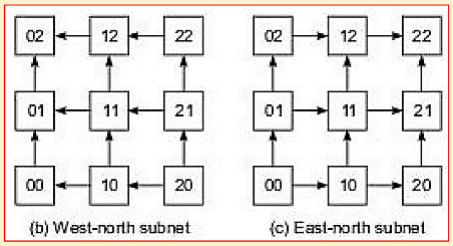
Prepared By Mr. EBIN PM, AP, IESCE EDULINE 32

❖Virtual Networks

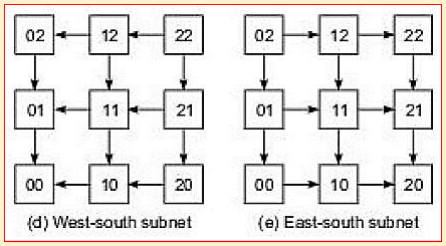
➢Virtual channels can be used to generate virtual networks. The 4 possible virtual network in a 3×3 mesh is shown below.



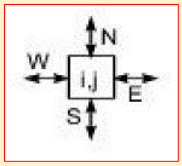
(a) A dual-channel 3 × 3 mesh



(b) West-north subnet (c) East-north subnet



(d) West-south subnet (e) East-south subnet

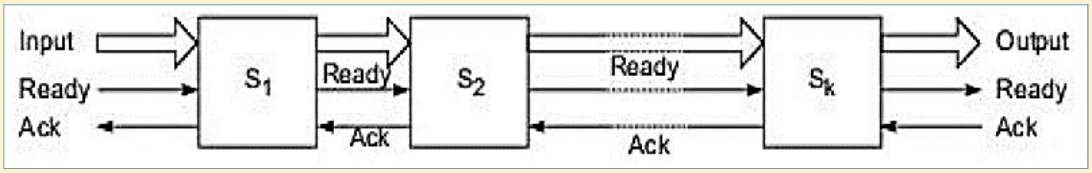


LINEAR PIPELINE PROCESSORS

- A linear pipeline processor is a cascade of processing stages which are linearly connected to perform a fixed function over a stream of data flowing from one end to the other.
- linear pipelines are applied for instruction execution, arithmetic computation, and memory-access operations.
- A linear pipeline processor has K processing stages. External inputs (operands) are fed into the first stage S_1
- The processed results are passed from stage S_i to stage S_{i+1} for all i= 1, 2,...K-1
- The final result emerges from the pipeline at the last stage S_k

- Depending on the control of data flow linear pipelines can be classified in to 2
  - Asynchronous
  - Synchronous

❖**Asynchronous Model**



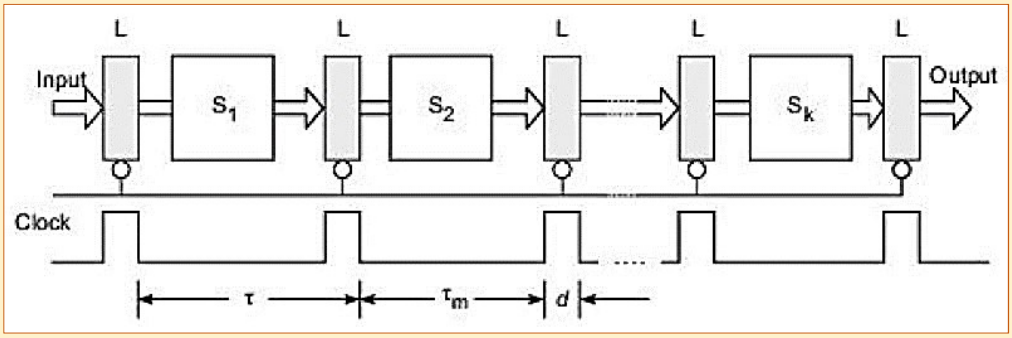
- Data flow between adjacent stages is controlled by a handshaking protocol.

- When stage S_i is ready to transmit, it sends a ready signal to stage S_{i+1} .
- After stage S_{i+1} receives the incoming data, it returns an acknowledge signal to S_i
- Asynchronous pipelines may have a variable throughput rate.
- Different amounts of delay may be experienced in different stages.

❖**Synchronous Model**

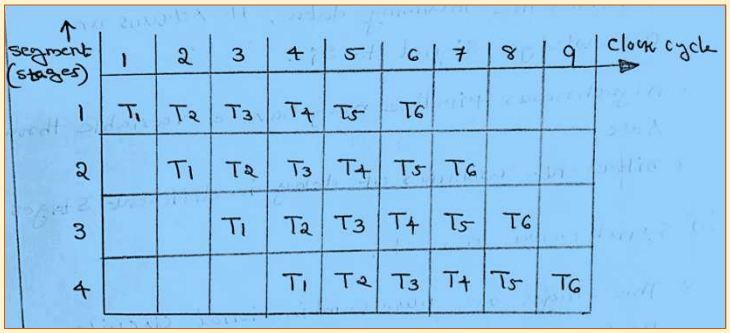
- The pipeline stages are pure combinational logic circuits performing arithmetic or logical operations over the data stream flowing through the pipe.

- The stages are separated by high speed interface latches
- Latches are fast registers (made by master slave flip-flop) for holding the intermediate results between the stages.
- Up on arriving the clock pulse all latches transfer data to the next stage simultaneously.



Space – time Diagram

- It illustrate the overlapped operations in a linear pipeline processor.



- The above figure shows the space –time diagram for 4 segment(stage) and 6 tasks.
- Once the pipeline is filled up, it will output one result per clock period independent of the number of stages in the pipe.

➢**Clock period**

Let d be the time delay of latch

k is the number of stages

τ_m is maximum stage delay

τ_i is the time delay of the circuitry in each stage

Clock period (Cycle time) $\tau = \tau_m + d$

➢The reciprocal of clock period is called frequency f

$f = 1/\tau$

Prepared By Mr. EBIN PM, AP, IESCE EDULINE 39

Example

Suppose the time delay of 4 stages are τ_1 = 60ns, τ_2= 50ns, τ_3=90ns and τ_4= 80ns and the interface latch has the delay of d=10ns. The cycle time of this pipe line is

$\tau = \tau_m + d$

$= 90+10 = 100ns$

This means that the clock frequency of the pipe line can be set to

$f = 1/\tau = 1/100 = 10$ MHz

Prepared By Mr. EBIN PM, AP, IESCE EDULINE 40

➢Speed up

Ideally , a linear pipeline with k stages can process n tasks in

$T_k = [k+(n-1)]\times$ **clock periods**

Where K cycles can used to fill up the pipeline and n-1 cycles are needed to complete the remaining n-1 task.

Total time required is $T_k = [k+(n-1)]\times \tau$

where τ is clock period

In a non pipelined processor the amount of time to execute n task is

$T_1 = n\times k\tau$ where $k\tau$ is delay.

➢The Speed up of a k-stage pipeline processor over an equivalent non pipe line processor as

$S_k = T_1 / T_k$

$= (n* k\tau) / (k+(n-1))\times \tau = nk / k+(n-1)$

➢ Efficiency

Let n = number of tasks

k = number of pipe line stages

τ = clock period

Efficiency $\eta = nk\tau / k[k\tau + (n-1)\tau] = n / k+ (n-1)$

➢When $n\rightarrow\infty$ the efficiency approaches 1. ie, the larger the number of task flowing through the pipeline, the better is its efficiency.

➢**Throughput**

- The number of tasks that can be completed by a pipeline per unit time is called its throughput.

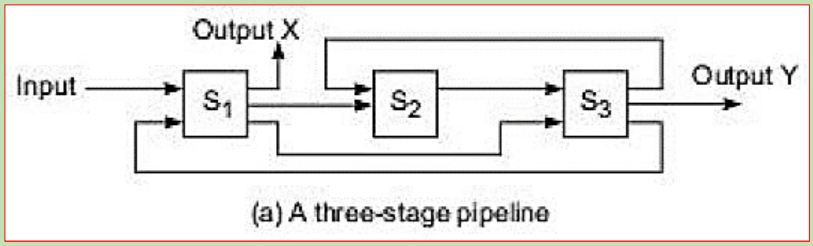
Throughput $w = n/ k\tau + (n-1)\tau = \eta / \tau$

➢The throughput reflects the computing power of a pipeline.

➢n equals the total number of tasks being processed during an observation period $k\tau + (n-1)\tau$

NONLINEAR PIPELINE PROCESSORS

- Linear pipeline processors are static because they are used to perform fixed functions.
- Nonlinear pipelines are dynamic pipelines. It allows feed forward and feedback connections in addition to the streamline connections



(a) A three-stage pipeline

Stream line connections

From S_1 to S_2

From S_2 to S_3

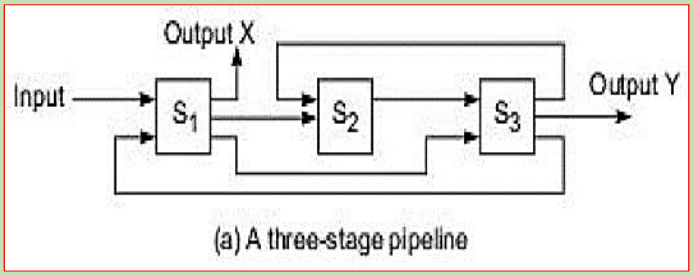
Feed forward connections

From S_1 to S_3

Feedback connections

From S_3 to S_2

From S_3 to S_1



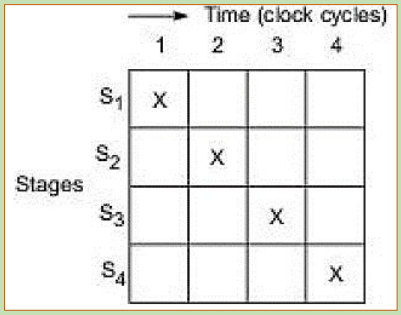
(a) A three-stage pipeline

- With these connections the output of the pipeline is not necessarily from the last stage.

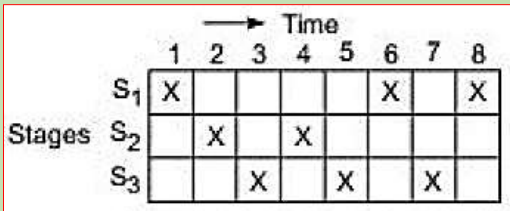
❖ Reservation Table

➢The utilization of pattern of successive stages in a pipeline is specified by a reservation table.

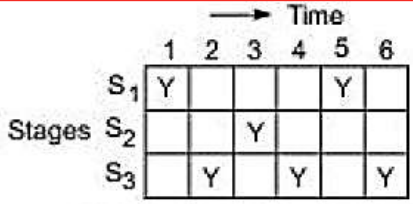
Reservation table for a 4 stage linear pipeline



➢Reservation table for a nonlinear pipeline processors follows a nonlinear pattern. On the evaluation of different functions multiple reservation table can be generated.



(b) Reservation table for function X



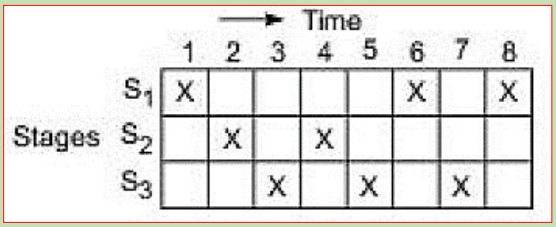
(c) Reservation table for function Y

- Multiple check marks in a row means repeated usage of the same stage in different cycles
- Contiguous check marks in a row means , extended usage of a stage over more than one cycle.

❖ Latency Analysis

- An initiation refers to the start of a single function evaluation.
- The number of cycles between two initiations of a pipeline is a latency between them.
- A latency of k means , two initiations are separated by k clock cycles. Latency value must be non negative integers
- When two or more initiations attempt to use the same stage at the same time , a collision results
- Some latencies will cause collisions. Latencies that cause collisions are called Forbidden latencies.

❖Forbidden latency



Consider the reservation table.

Forbidden latencies are

S_1 = {(6-1), (8-1), (8-6)}

S_2 = {(4-2)}

S_3 = {(5-3), (7-5), (7-3)}

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

Forbidden latency = {2, 4, 5, 7}

Prepared By Mr. EBIN PM, AP, IESCE EDULINE 49

❖Collision Vector

Forbidden latency = {2, 4, 5, 7}

- 7 is the largest number in forbidden latency. So we require 7 bits to represent collision vector.

Collision Vector = {C_7 C_6 C_5 C_4 C_3 C_2 C_1}

Permissible latency is the Non-forbidden latency = {1, 3, 6}

We will not take 8 in permissible latency because we need only 7 bits to represent Collision Vector.

Now, Forbidden latency is {2, 4, 5, 6}

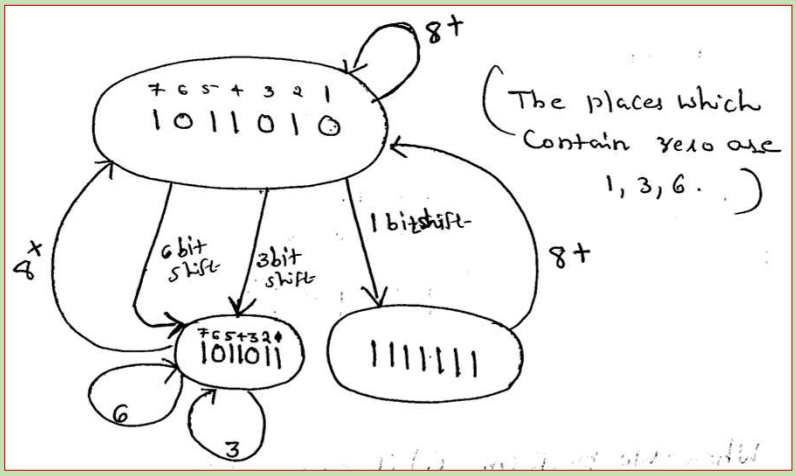
Collision Vector = {C_7 C_6 C_5 C_4 C_3 C_2 C_1} = **(1011010)**

1 0 1 1 0 1 0

Prepared By Mr. EBIN PM, AP, IESCE EDULINE 50

❖ State Diagram

➢From collision vector we can draw state diagram



Shifting

1011010 – 1 bit shift

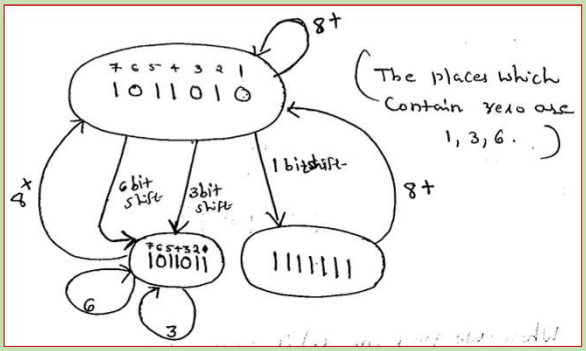
0101101

Then do Bitwise OR

1011010 - collision vector

0101101 - shifted answer

1111111



Shifting

1011010 – 3 bit shift

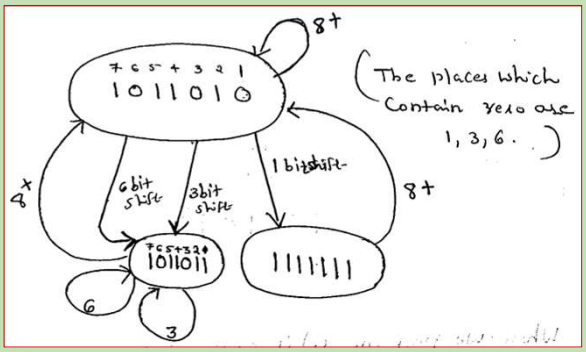
0001011

Then do Bitwise OR

1011010 - collision vector

0001011 - shifted answer

1011011



Shifting

1011010 – 6 bit shift

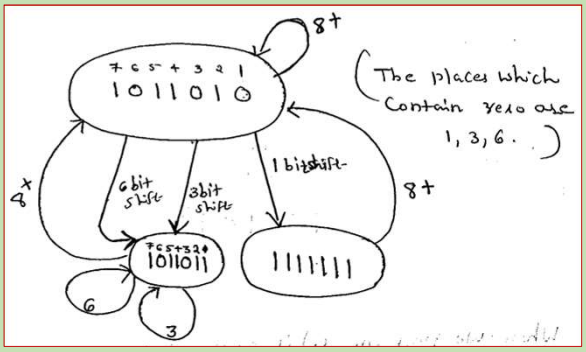
0000001

Then do Bitwise OR

1011010 - collision vector

0000001 - shifted answer

1011011



- When we perform 6 bit and 3 bit shift on 1011011, we will get the same answer. So we can draw two self loop
- If we perform 8 bit shift or more than 8 bit shift we will get same collision vector 1011010. So we can add loop.

❖Simple cycle

- Simple cycle is a latency cycle in which each state appears only once.

Eg: (3), (6), (8), (1,8) , (3,8), (6,8)

❖Greedy Cycle

Some simple cycles are greedy cycle. Greedy cycle edges are all made with minimum latencies from their starting state.

Eg: (1,8) , (3)

The average latency of (1,8) is (1+8)/2 = 4.5 which is lower than that of the simple cycle (6,8) = (6+8)/2 = 7

❖Minimum Average Latency (MAL)

- ➢At least one of the greedy cycle will lead to the MAL.
- ➢The greedy cycle (3) has a constant latency which equals the MAL for evaluating function X without causing a collision.