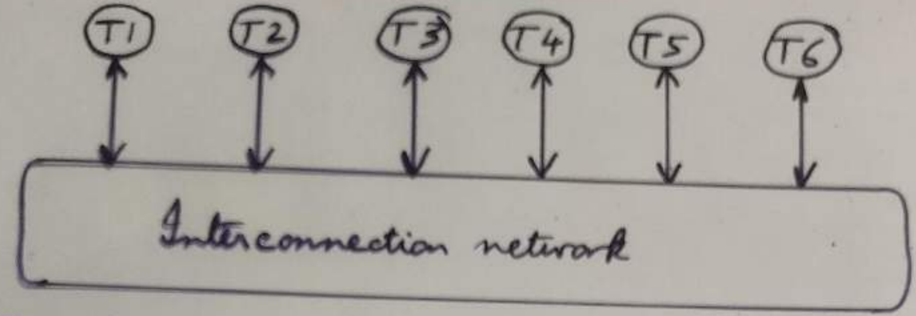
Notes-08

INTERCONNECTION NETWORKS

Informal: As we increase the number of processors, a single-bus architecture becomes impractical. The shared bus becomes a bottleneck and performance goes down. The solution is to use multistage networks to connect the processors.

The three main issues are topology, routing, and flow control. Topology decides the structure of the network: the links and nodes. Routing determines how data can be transferred between two specified nodes. Flow control determines how network resources are allocated to packets on their way from source to destination.

INTERCONNECTION NETWORKS



FUNCTIONAL VIEW of an interconnection network. Terminals (T1.. T6) are connected to the network using channels.

An interconnection network is a programmable system that transports data between terminals.
When terminal T3 wishes to communicate some data with terminal T5, it sends a message containing the data into the network and the network delivers the message to T5.
The network is programmable in the sense that it makes different connections at different points in time. The network in the figure may deliver a message from T3 to T5 in one cycle and then use the same resources to deliver a message from T3 to T1 in the next cycle.
The network is a system because it is composed of many components: buffers, channels, switches, and controls that work together to deliver data.

Slide 8.1

TOPOLOGY, ROUTING, and FLOW CONTROL

TOPOLOGY: The interconnection network is implemented with a collection of shared router nodes connected by shared channels. The connection pattern of these nodes defines the network's topology. A message is then delivered between terminals by making several hops across the shared channels and nodes from its source terminal to its destination terminal.

ROUTING: Once a topology has been chosen, there can be many possible paths (sequences of nodes and channels) that a message could take through the network to reach its destination. Routing determines which of these possible paths a message actually takes.

FLOW CONTROL: Flow control dictates which messages get access to particular network resources over time.

Slide 8.2

CIRCUIT SWITCHING

Circuit Switching is a form of bufferless flow control that operates by first allocating channels to form a circuit from source to destination and then sending one or more packets along this circuit. When no further packets need to be sent, the circuit is deallocated.

NON-BLOCKING NETWORKS

A network is said to be non-blocking if it can handle all circuit requests that are a permutation of the inputs and outputs. That is, a dedicated path can be formed from each input to its selected output without any conflicts (shared channels). Conversely, a network is blocking if it cannot handle all such circuit requests without conflicts.

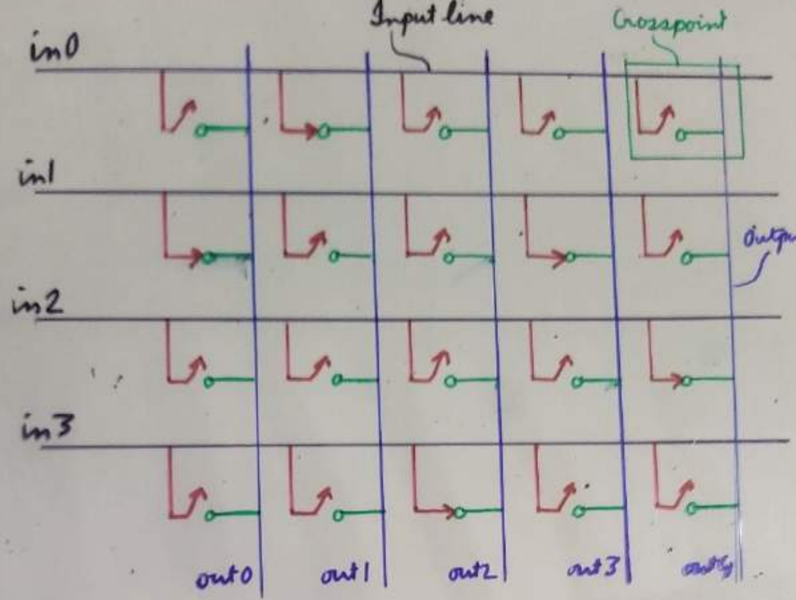
STRICTLY NON-BLOCKING NETWORKS

A network is strictly non-blocking if any permutation can be set up incrementally, one circuit at a time, without the need to reroute (or rearrange) any of the circuits that are already set up.

REARRANGEABLY NON-BLOCKING (or REARRANGEABLE) NETWORKS: Such a network can route circuits for arbitrary permutations, but incremental construction of a permutation may require rearranging some of the early circuits to permit later circuits to be set up.

Slide 8.3

CROSSBAR NETWORKS

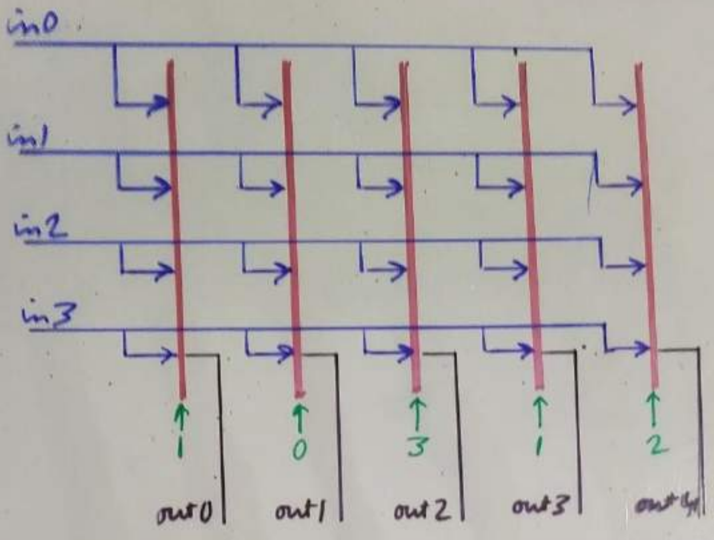


CONCEPTUAL MODEL OF A 4×5 CROSSBAR.
It consists of 4 input lines, 5 output lines, and 20 crosspoints. Each output may be connected to at most one input, while each input may be connected to any number of outputs. This switch has inputs 1, 0, 3, 1, 2 connected to outputs 0, 1, 2, 3, 4, respectively.

An $n \times m$ crossbar or crosspoint switch directly connects n inputs to m outputs with no intermediate stages. In effect, such a switch consists of m $n:1$ multiplexers, one for each output.

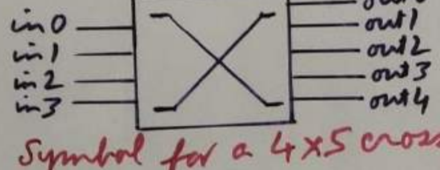
Slide 8.4

CROSSBAR : Implementation



A 4×5 crossbar switch as implemented with 5 4:1 multiplexers. Each multiplexer selects the input to be connected to the corresponding output. Here the connection is {1, 0, 3, 1, 2} → {0, 1, 2, 3, 4}

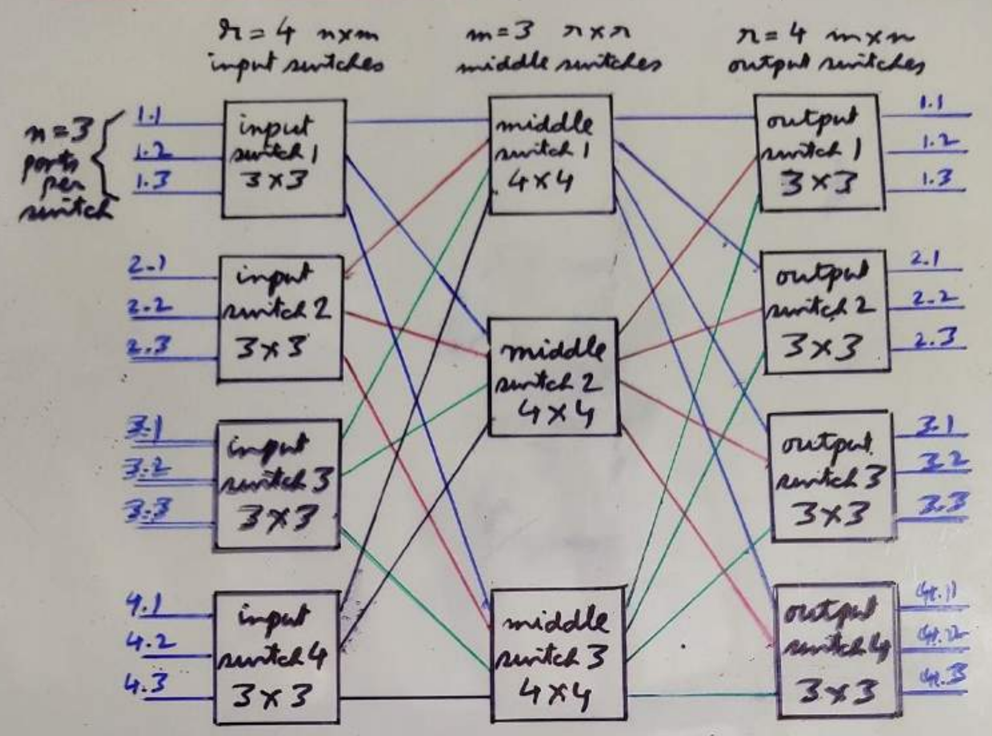
Each of the n input lines connects to one input of m n:1 multiplexers. The outputs of the multiplexers drive the m output ports. The multiplexers may be implemented with tri-state gates or wired-OR gates driving an output line, or with a tree of logic gates to realize a more conventional multiplexer.



Symbol for a 4×5 crosspoint switch

Slide 8.5

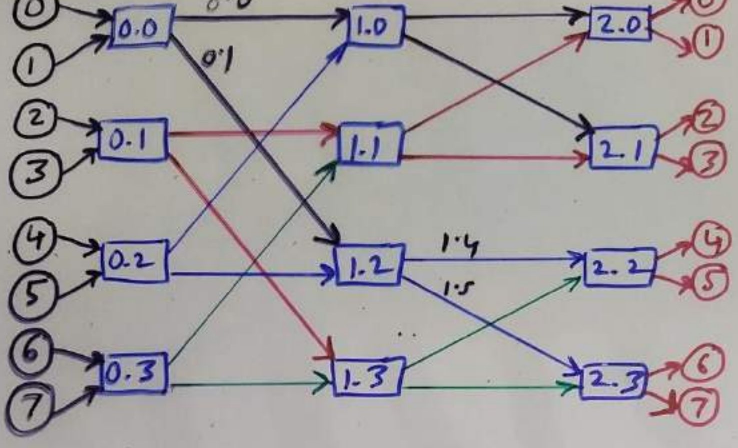
CLOS NETWORKS



An (m=3, n=3, r=4) symmetric Clos network.
All switches are crossbars.

A Clos network is a three-stage network in which each stage is composed of a number of crossbar switches. A symmetric Clos is characterized by a triple (m, n, r) where m is the number of middle-stage switches, n is the number of input (output) ports on each input (output) switch and r is the number of input and output switches. In a Clos network, each middle stage switch has one input link from every input switch and one output link to every output switch.

BUTTERFLY NETWORKS



A 2-ary 3-fly.

A k-ary n-fly network consists of k^n source terminal nodes, n stages of k^{n-1} $k \times k$ crossbar switch nodes, and finally k^n destination terminal nodes.

Routing: $6 \rightarrow 1$

$$6 \rightarrow \boxed{0.3} \xrightarrow[\text{(lower)}]{0} \boxed{1.1} \xrightarrow[\text{(lower)}]{0} \boxed{2.0} \xrightarrow[\text{(upper)}]{1} 1$$

FLOW CONTROL

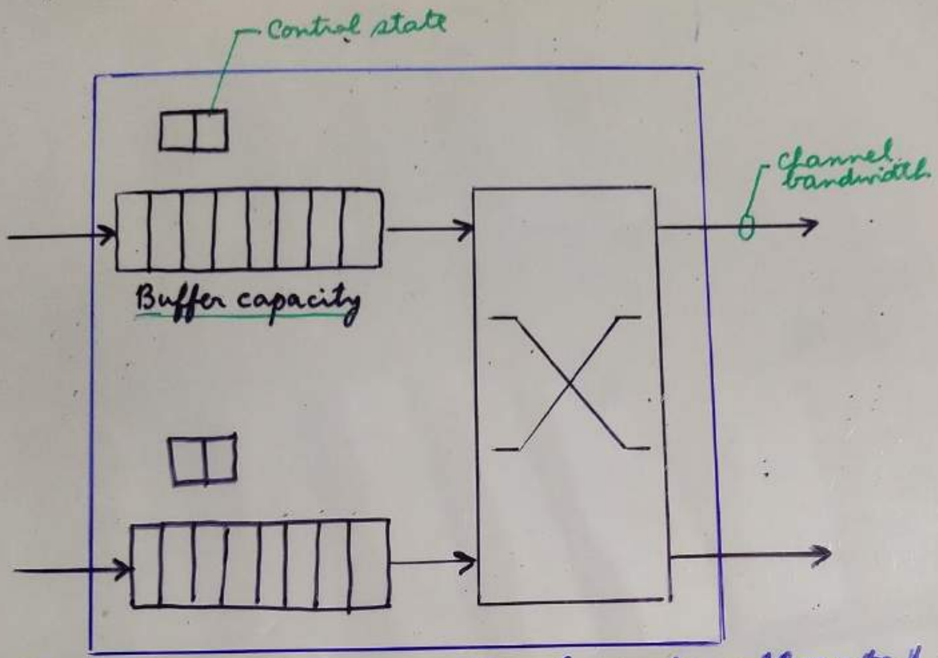
- Flow control determines how a network's resources, such as channel bandwidth, buffer capacity, and control state, are allocated to packets traversing the network.

- A good flow-control method allocates these resources in an efficient manner so the network achieves a high fraction of its ideal bandwidth and delivers packets with low, predictable latency.

- One can view flow control as either a problem of resource allocation or one of contention resolution. From the resource allocation perspective, resources in the form of channels, buffers, and state must be allocated to each packet as it advances from the source to the destination. The same problem can be viewed as one of resolving contention. For example, two packets arriving on different inputs of a router at the same time may both desire the same output. In this situation, the flow-control mechanism resolves this contention, allocating the channel to one packet and somehow dealing with the other, blocked packet.

Slide 8.8

INTERCONNECTION NETWORKS: RESOURCES



Resources within one network node allocated by a flow control method.

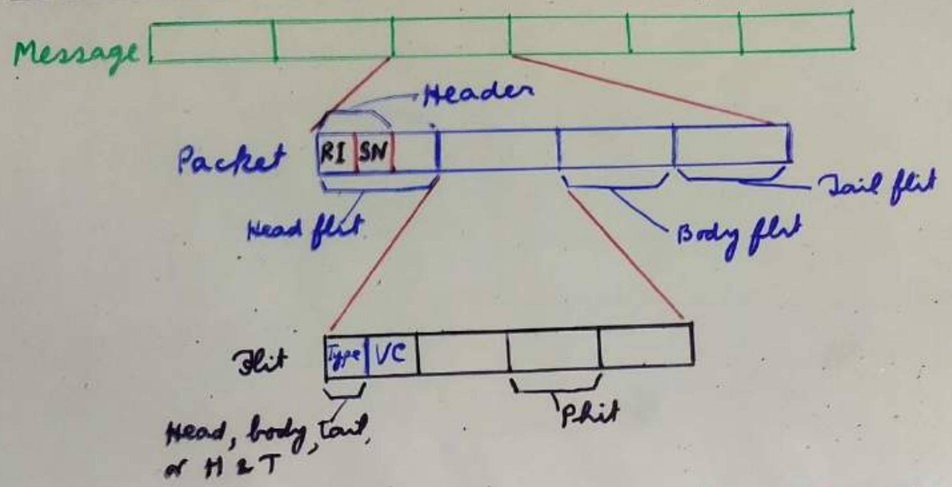
- The control state tracks the resources allocated to the packet within the node and the state of the packet's traversal across the node.

- To advance to the next node, the packet must be allocated bandwidth on an output channel of the node.

- As the packet arrives at a node, it is temporarily held in a buffer while awaiting channel bandwidth. However, some flow control methods do not allocate buffers.

Slide 8.9

INTERCONNECTION NETWORKS: Allocation Units



- At the top level, a message is a logically contiguous group of bits that are delivered from a source terminal to a destination terminal.
- Because messages may be arbitrarily long, resources are not directly allocated to messages. Instead, messages are divided into one or more packets that have a restricted maximum length. By restricting the length of a packet, the size and time duration of a resource allocation is also restricted, which is often important for the performance and functionality of a flow control mechanism.
- A packet is the basic unit of routing and sequencing. A packet consists of a segment of a message to which a packet header is prepended. The packet header includes routing information (RI) and, if needed, a sequence number (SN).

INTERCONNECTION NETWORKS: ALLOCATION UNITS [contd.]

- A packet may be further divided into flow control digits or flits. A flit is the basic unit of bandwidth and storage allocation used by most flow control methods.

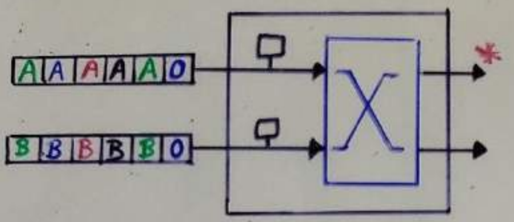
- Flits carry no routing and sequence information and thus must follow the same path and remain in order. However, flits may contain a virtual-channel identifier (VCID) to identify which packet the flit belongs to in systems where multiple packets may be in transit over a single physical channel at the same time.

- A flit is itself subdivided into one or more physical transfer digits or phits. A phit is the unit of information that is transferred across a channel in a single clock cycle. Although no resources are allocated in units of phits, a link level protocol must interpret the phits on the channel to find the boundaries between flits.

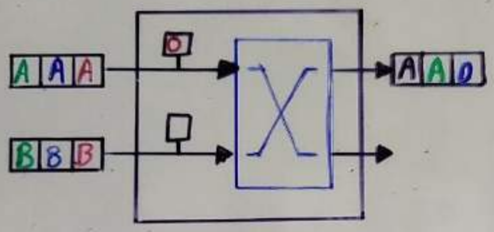
ALLOCATION UNIT	BIT-LENGTH		
	MIN	TYPICAL	MAX
Phit	1	8	64
Flit	16	64	512
Packet	128	1K	512K

Slide 8.11

BUFFERLESS FLOW CONTROL

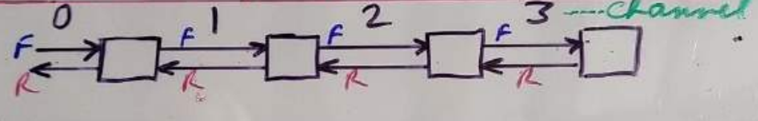


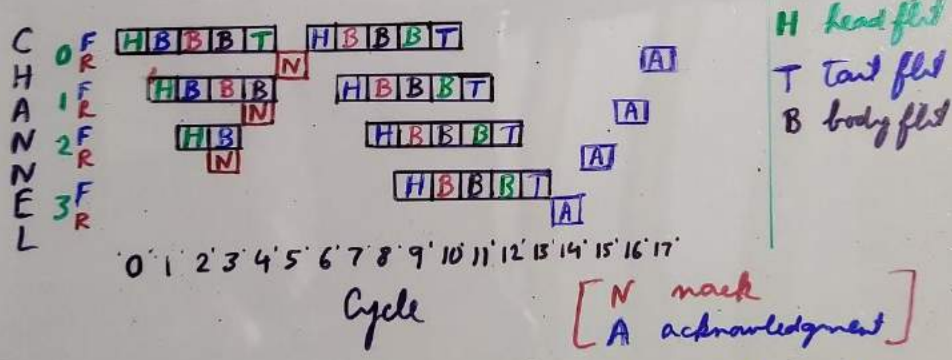
(a) Two packets, A and B arrive at a network node. Both request output channel O*



(b) Dropping flow control: A acquires channel O and B is dropped. B must be retransmitted from the source.

DROPPING FLOW CONTROL WITH EXPLICIT NEGATIVE ACKNOWLEDGMENT





A five-flit packet is sent along a four-hop route. The first transmission of the packet is unable to allocate channel 3 and is dropped. A nack (negative acknowledgment) triggers a retransmission of the packet which succeeds.

Slide 8.12

BUFFERED FLOW CONTROL

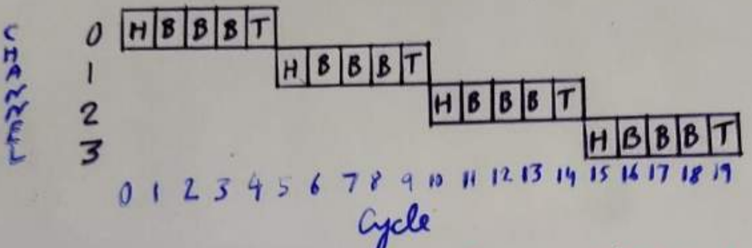
Buffered flow control is more efficient than bufferless flow control. This is because a buffer decouples the allocation of adjacent channels. Without a buffer, the two channels must be allocated to a packet (or flit) during consecutive cycles, or the packet must be dropped or misrouted. There is nowhere else for the packet to go. Adding a buffer gives us a place to store the packet (or flit) while waiting for the second channel, allowing the allocation of the second channel to be delayed without complications. For example, a flit can be transferred over the input channel on cycle i and stored in a buffer for a number of cycles j until the output channel is successfully allocated on cycle $i+j$. We can approach 100% channel utilization with buffered flow control

Slide 8.13

PACKET-BUFFER FLOW CONTROL

Method 1: Store-and-forward
Method 2: Cut-through

STORE-AND-FORWARD FLOW CONTROL

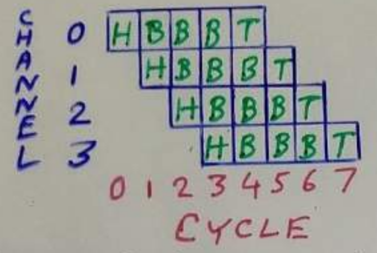


Time-space diagram showing store-and-forward flow control used to send a 5-flit packet over 4 channels.

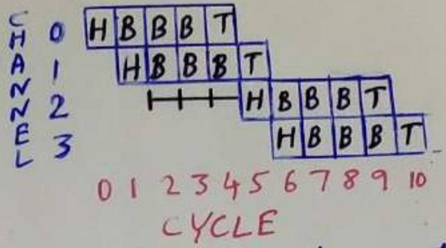
With store-and forward flow control, each node along a route waits until a packet has been completely received (stored) and then forwards the packet to the next node. The packet must be allocated two resources before it can be forwarded: a packet-sized buffer on the far side of the channel and exclusive use of the channel. Once the entire packet has arrived at a node and these two resources are acquired, the packet is forwarded to the next node. While waiting to acquire resources, if they are not immediately available, no channels are being held idle and only a single packet buffer on the current node is occupied.

PACKET-BUFFER FLOW CONTROL (contd.)

Method 2: CUT-THROUGH FLOW CONTROL



(a) Packet proceeds without contention



(b) Packet encounters contention for three cycles before it is able to allocate channel 2.

Time-space diagram showing cut-through flow-control sending a 5-flit packet over 4 channels.

- Store-and-forward — HIGH LATENCY.
- Cut-through — forwards a packet as soon as the header is received and resources (buffer and channel) are acquired, without waiting for the entire packet to be received.

Shortcomings of packet-buffer flow-control

- Buffers are allocated in units of packets. This results in a very inefficient use of buffer storage. [Efficient method — FLIT BUFFER]

- Contention latency is increased. A high-priority packet colliding with a low-priority packet must wait for the entire low-priority packet to be transmitted before it can acquire the channel.

SLIde 8.15

FLIT-BUFFER FLOW CONTROL

(I) WORMHOLE FLOW CONTROL

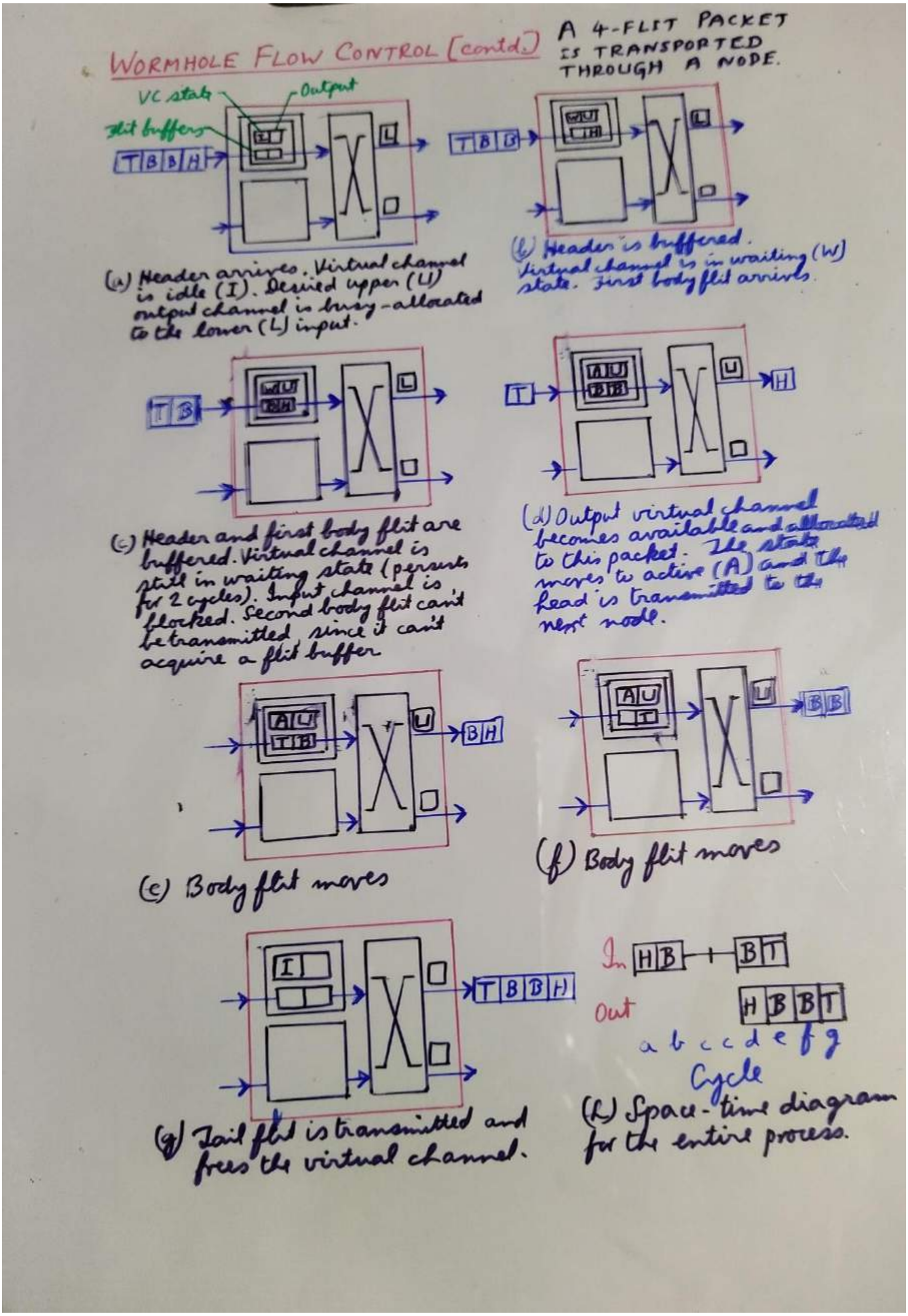
Wormhole flow control operates like cut-through, but with channel and buffers allocated to flits rather than packets. When the head flit of a packet arrives at a node, it must acquire three resources before it can be forwarded to the next node along a route:

- a virtual channel (channel state) for the packet.
- one flit buffer
- one flit of channel bandwidth

Body flits of a packet use the virtual channel acquired by the head flit and hence need only acquire a flit buffer and a flit of channel bandwidth to advance. The tail flit of a packet is handled like a body flit, but also releases the virtual channel as it passes.

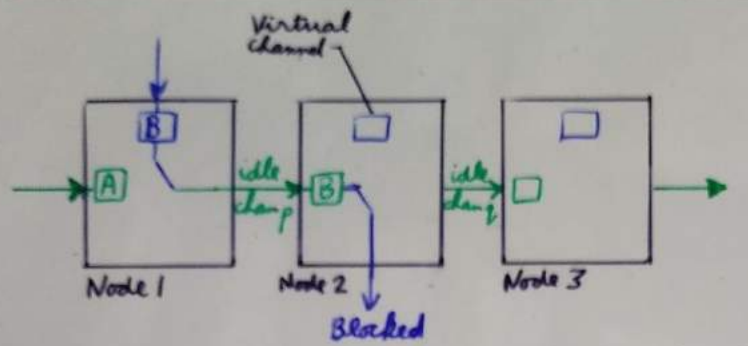
A virtual channel holds the state needed to coordinate the handling of the flits of a packet over a channel. (Output channel, state, etc.)

Slide 8.16

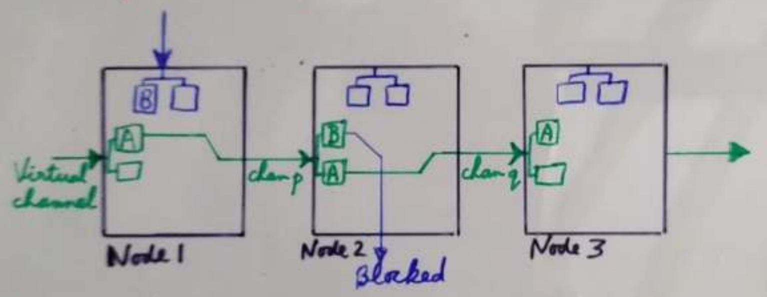

WORMHOLE FLOW CONTROL [contd.]
A 4-FLIT PACKET IS TRANSPORTED THROUGH A NODE.
VC state
Output
Flit buffers
T B B H
T B B
(a) Header arrives. Virtual channel is idle (I). Desired upper (U) output channel is busy-allocated to the lower (L) input.
(b) Header is buffered. Virtual channel is in waiting (W) state. First body flit arrives.
T B
(c) Header and first body flit are buffered. Virtual channel is still in waiting state (persists for 2 cycles). Input channel is blocked. Second body flit can't be transmitted since it can't acquire a flit buffer
T
H
(d) Output virtual channel becomes available and allocated to this packet. The state moves to active (A) and the head is transmitted to the next node.
B H
(e) Body flit moves
B B
(f) Body flit moves
T B B H
In H B B T
Out H B B T
a b c c d e f g
Cycle
(g) Tail flit is transmitted and frees the virtual channel.
(h) Space-time diagram for the entire process.

Slide 8.17

VIRTUAL-CHANNEL FLOW CONTROL



(a) WORMHOLE. When a packet B blocks while holding the sole virtual channel associated with channel p, channels p and q are idled even though packet A requires use of these idle channels.



(b) VIRTUAL-CHANNEL. Packet A is able to proceed over channels p and q using a second virtual channel associated with channel p on node 2.

Virtual-channel flow control, which associates several virtual channels (channel state and flit buffers) with a single physical channel, overcomes the blocking problems of wormhole flow control by allowing other packets to use the channel bandwidth that would otherwise be left idle when a packet blocks.

Slide 8.18

BUFFER MANAGEMENT

All of the flow control methods that use buffering need a means to communicate the availability of buffers at the downstream nodes. Then the upstream nodes can determine when a buffer is available to hold the next flit (or packet for store-and-forward or cut-through) to be transmitted. This type of buffer management provides backpressure by informing the upstream nodes when they must stop transmitting flits because all of the downstream flit buffers are full.

Common low-level flow control mechanisms that provide backpressure:

(i) credit-based;
(ii) on/off
(iii) ack/nack.

CREDIT-BASED FLOW CONTROL

With credit-based flow control, the upstream router keeps a count of the number of free flit buffers in each virtual channel downstream. Then, each time the upstream router forwards a flit, thus consuming a downstream buffer, it decrements the appropriate count. If the count reaches zero, all of the downstream buffers are full and no further flits can be forwarded until a buffer becomes available. Once the downstream router forwards a flit and frees the associated buffer, it sends a credit to the upstream router, causing a buffer count to be incremented.

Slide 8.19