1. (Dennard scaling). According to Dennard scaling, why, in every technology generation, transistor integration doubles while system power consumption stays the same?

device dimension scale by $\frac{1}{k}$

power density = $\frac{VI}{A}$

voltage V scale by $\frac{1}{k}$

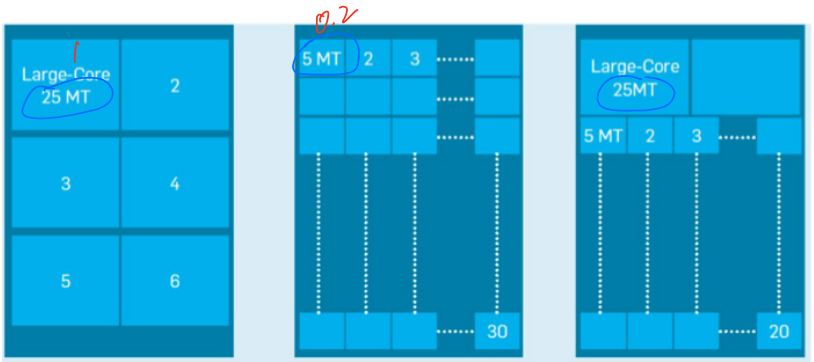
Current I scale by $\frac{1}{k}$

area scale by $\frac{1}{k^2}$

∴ power density scale by 1

power density stays constant, so that the power consumption stays in proportion with area.

2. (Why many 'small' cores?) Figure 1 shows three scenarios for integrating 150 million logic transistors into cores. The large core (25 MT, 25 million transistors) throughput is given in scenario (a). Calculate small-core (5 MT, 5 million transistors) throughput and the total throughput for scenarios (b) and (c). [This figure is modified from Shekhar Borkar and Andrew A. Chien, "The future of microprocessors", Communications of the ACM | May 2011.]



Large-Core Homogeneous	
Large-core throughput	1
Small-core throughput	
Total throughput	6

(a)

Small-Core Homogeneous	
Large-core throughput	
Small-core throughput	?
Total throughput	?

(b)

Small-Core Homogeneous	
Large-core throughput	1
Small-core throughput	?
Total throughput	?

(c)

Figure 1. Three scenarios for integrating 150-million logic transistors into cores

for(b): small-core throughput = $1 \div \sqrt{5} = 0.447$

total throughput $= 0.2 \times 30 = 13.34$

for(c): small-core throughput = $1 \div \sqrt{5} = 0.447$

total throughput $= 0.447 \times 20 + 1 \times 2 = 10.94$

3. (Pipelined bus protocol) Figure 2 shows a high-performance bus pipelining protocol for Read and Write transactions. We assume that a read transaction has 2 cycles of processing delay.

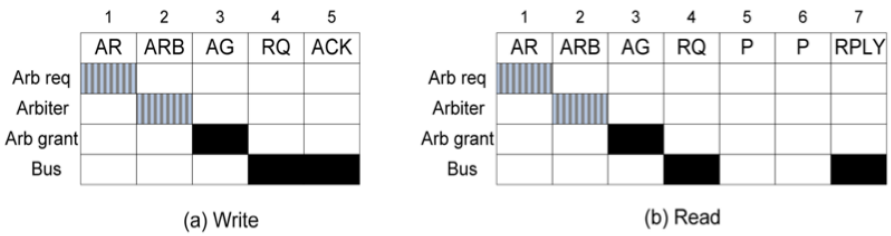
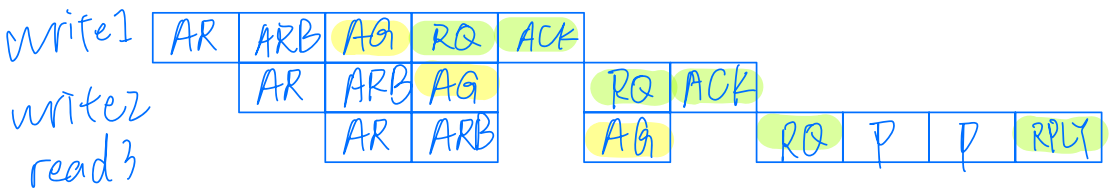


Figure 2. Pipeline sequences and reservation tables

Based on the pipeline sequences and reservation tables, draw a timing diagram for the following two transaction sequences:

(1) Write 1, Write 2, Read 3
(2) Write 1, Read 2, Read 3

(1)



(2)

4. (Split transaction protocol). Redo the above task 3, but using the split-transaction bus protocol.

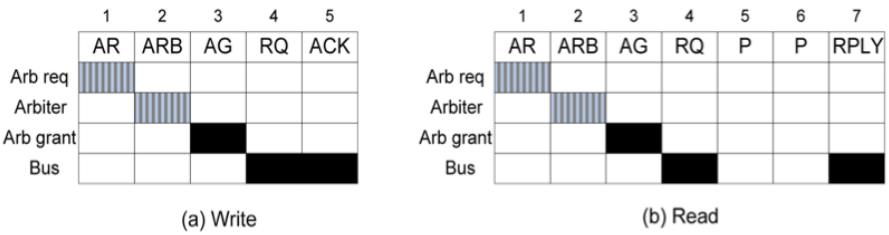


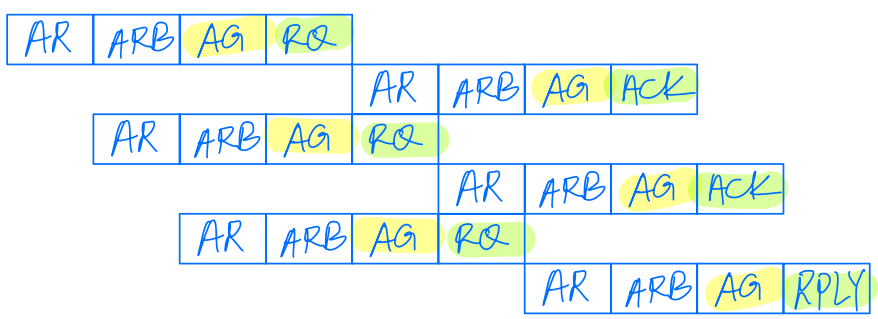
Figure 2. Pipeline sequences and reservation tables

Based on the pipeline sequences and reservation tables, draw a timing diagram for the following two transaction sequences:

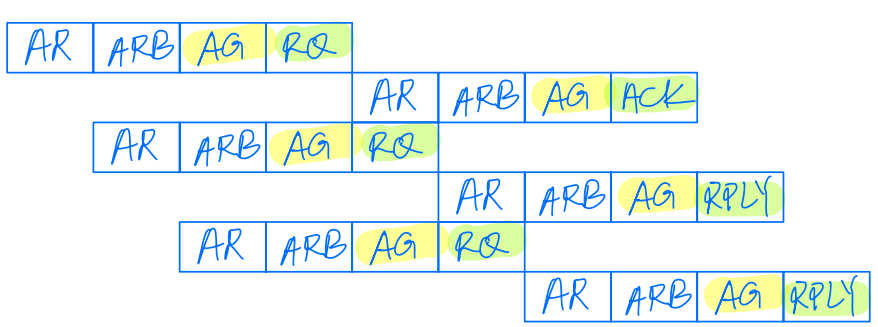
(1) Write 1, Write 2, Read 3
(2) Write 1, Read 2, Read 3

AR

(1)
write 1
Ack 1
write 2
Ack 2
Read 3
RPLY 3



(2)
write 1
Ack 1
Read 2
RPLY 2
Read 3
RPLY 3



5. (Bus protocols) We have learnt three bus protocols, from bus non-pipelining, bus pipelining and split-transaction.
(1) Which problem of the bus non-pipelining protocol is addressed by the bus pipelining protocol and how it is addressed? Which problem of the bus pipelining protocol is addressed by the split transaction and how it is addressed?
(2) Which protocols assume in-order completion? Which protocol allows out-of-order completion?
(3) Give one example to show the performance benefits of out-of-order completion by comparing the average delay of at least 3 transactions. (Inclusion of arbitration steps is not necessary)
(4) If the transactions complete out-of-order, how to identify which reply matches which request?

(1) problem of bus non-pipelining: low performance
↗ module by module, wait for a long time
addressed by bus pipelining
module 2 follows module 1 1 cycle later

problem of bus pipelining: Bus idling between
low bus utilization RQ/RPLY of a transaction
⇐ is long (because a bus can only address one RQ/RPLY (ACK) at one time)

addressed by split-transaction by
Spliting each transaction into 2 transections:
'request' and 'reply'

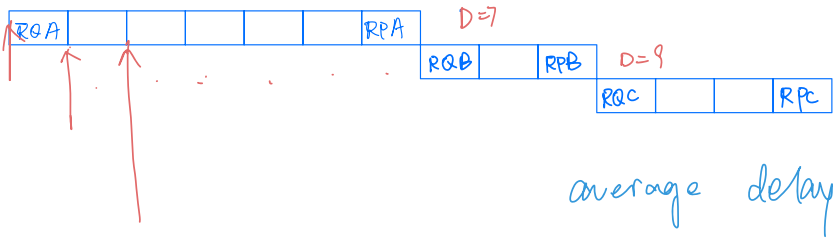
even though the latency for individual operation is increased, but bus utilization is also increased by eliminating P cycles

(2) in-order: bus non-pipelining, bus pipelining
out-of-order: split-transaction

(3)

in-order:

(no transaction overlapping)



average delay = $\frac{7+9+12}{3} = 9.3$

out-of-order:

average delay = $\frac{7+3+4}{3} = 4.7$

(4) need "tag" to match sender / receiver

6. (Channel load and ideal throughput on mesh network) There are two methods of calculating ideal network throughput. One way is the *hop count bound*, which assumes that all utilized channels are perfectly load balanced. The other is the *bi-section bound*, which counts the load on the bisection channels. Strictly speaking, the bi-section bound considers the load on *bottleneck channels* as the limiting factor for network throughput. Thus it can be considered as a bottleneck channel method. Note that the two methods may give different results which mean different tightness.
Consider a 4 x 4 network as shown below. All channels have equal bandwidth, b. For the following cases, use both the *hop count* bound and the *bi-section* or *bottleneck channel* bound to calculate the ideal network throughput. Discuss why, if the two methods give different results.

k-ary n-cube (torus)

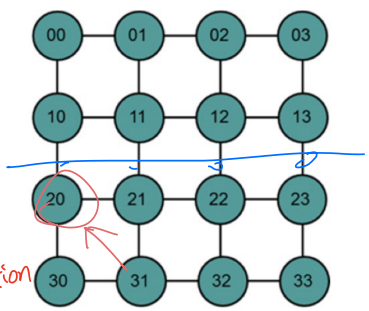
$H_{min} = \begin{cases} \frac{nk}{4} & k \text{ even} \\ n(\frac{k}{4} - \frac{1}{4k}) & k \text{ odd} \end{cases}$

k-ary n-mesh

$H_{min} = \begin{cases} \frac{nk}{3} \\ nC\frac{k}{3} \end{cases}$

no self-transaction

$\times \frac{16}{12}$



"4x4"-Mesh
(4-ary 2-mesh)

a) Assume ~~uniform random traffic~~, i.e., each node sends traffic to all other nodes with equal probability using shortest path.

b) Assume all-to-one traffic, i.e., all other nodes send traffic to the upper left corner node using the shortest path. If a node has more than one shortest-path directions to choose when routing packets, it randomly chooses one direction, i.e., equal probability.

uniform random traffic

a) ① Hop count bound

$$H_{min} = \frac{nk}{3} = \frac{4\times 2}{3} = \frac{8}{3}$$

$$\gamma_{max} = \frac{H_{min} N}{C} = \frac{\frac{8}{3}\times 16}{24\times 2} = \frac{8}{9}$$

$$\theta_{ideal} = \frac{b}{\gamma_{max}} = \frac{9}{8} b$$

② Bisection bound

$$\gamma_{max} = \frac{N}{2BC} = \frac{16}{2\times 8} = 1$$

$$\theta_{ideal} = \frac{b}{\gamma_{max}} = \frac{b}{1} = b$$

b) hop-count:

$$H_{min} = \frac{(1\times 2 + 2\times 3 + 3\times 4 + 4\times 3 + 5\times 2\times 6\times 1)\times 4 + (1\times 3 + 2\times 4 + 3\times 4 + 4\times 3 + 5\times 4)\times 8 + (1\times 4 + 2\times 6 + 3\times 4 + 4\times 1)\times 4}{16^2} = \frac{5}{2}$$

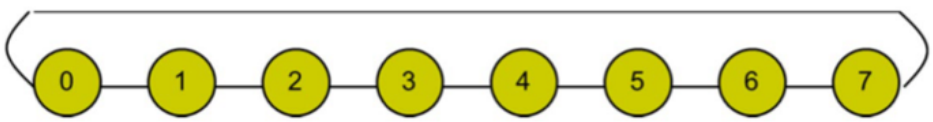
$$\gamma_{max} = \frac{H_{min} \cdot N}{C} = \frac{\frac{5}{2}\times 16}{24\times 2} = \frac{5}{6}$$

$$\theta_{ideal} = \frac{b}{\gamma_{max}} = \frac{6}{5} b$$

② $\gamma_{max} = \frac{\text{total load}}{\text{bottleneck channels}} = \frac{15}{2}$

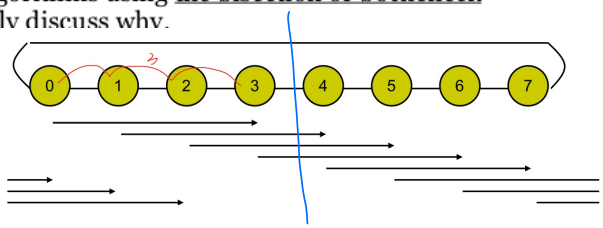
④ $\Theta_{ideal} = \frac{b}{\gamma_{max}} = \frac{b}{7.5} = \frac{2}{15} b$

7. (Channel load, ideal throughput and routing algorithms) In Lecture 4, we illustrate the complication of routing algorithms using a motivating example of the 8-node ring network with the tornado traffic pattern, for which we calculate the ideal throughput using the hop count bound for the three routing algorithms: greedy, uniform random, weighted random.



Recalculate the ideal throughput for the three algorithms using the bisection or bottleneck channel bound. If it gives a different result, briefly discuss why.

a) Greedy algorithm
b) Uniform random
c) Weighted uniform random



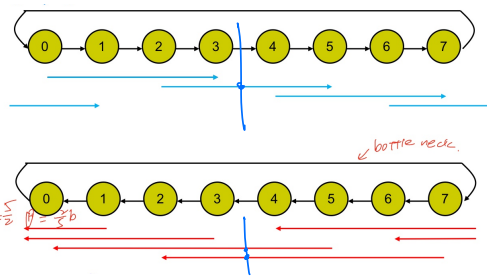
a)

$\gamma_{max} = \frac{b}{2} = 3$

$\Theta_{ideal} = \frac{1}{3} b$

b) $\gamma_{max} = \frac{5}{2}$

$\theta_{ideal} = \frac{2}{5} b$

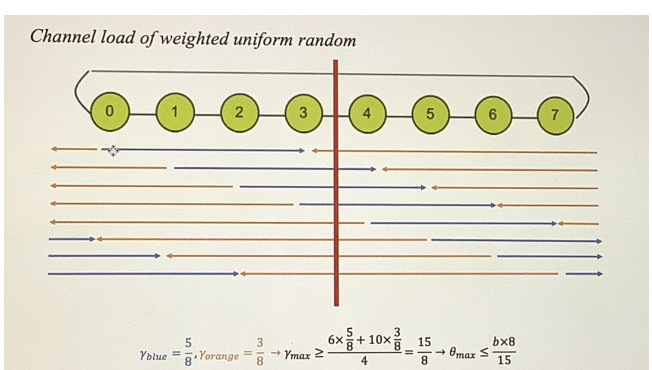


c) counter-clockwise:

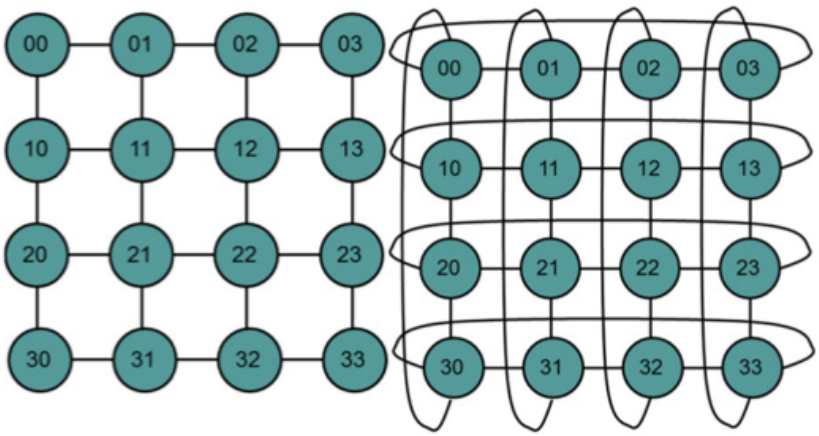
$\gamma_{max} = 5 \times \frac{3}{8} = \frac{15}{8}$, $\theta_{ideal} = \frac{8}{15} b$

clockwise:

$\gamma_{max} = 3 \times \frac{5}{8} = \frac{15}{8}$, $\theta_{ideal} = \frac{8}{15} b$



8. (Topology and network scalability) Consider n-by-n mesh and torus networks. Two 4-by-4 networks are shown below as examples. Consider uniform random traffic. Use the formulas on the cover page of Dally and Towles's book "Principles and Practices of Interconnection Networks" to compute the average minimum hop count H_{min} for the two kinds of networks.



"4x4"-Mesh
(4-ary 2-mesh)

"4x4"-Torus
(4-ary 2-cube)

a) Use the bisection bound to calculate their ideal throughput.
b) Use the hop count bound to calculate their ideal throughput.
c) Draw a graph to show how the ideal throughput would change with network size. (Select a few points for k-ary-2-mesh and k-ary-2-cube (torus), k=2, 4, 8, 16, 24, 32, 48, 64, to draw the graph.) Compare the ideal throughput in both networks as the value of k increases.

(a) for 4x4 mesh:

$\gamma_{max} = \frac{N}{2B_c} = \frac{16}{2\times 8} = 1$

$\theta_{ideal} = 16$

for 4x4 torus

$\gamma_{max} = \frac{N}{2B_c} = \frac{16}{2\times 16} = \frac{1}{2}$

$\theta_{ideal} = 2\times 16$

b) for 4×4 mesh:

$$H_{min} = \frac{nk}{3} = \frac{4\times 2}{3} = \frac{8}{3}$$

$$\gamma_{max} = \frac{H_{min}\, N}{C} = \frac{\frac{8}{3}\times 16}{24\times 2} = \frac{8}{9}$$

$$\theta_{ideal} = \frac{b}{\gamma_{max}} = \frac{9}{8}b$$

for 4×4 Torus

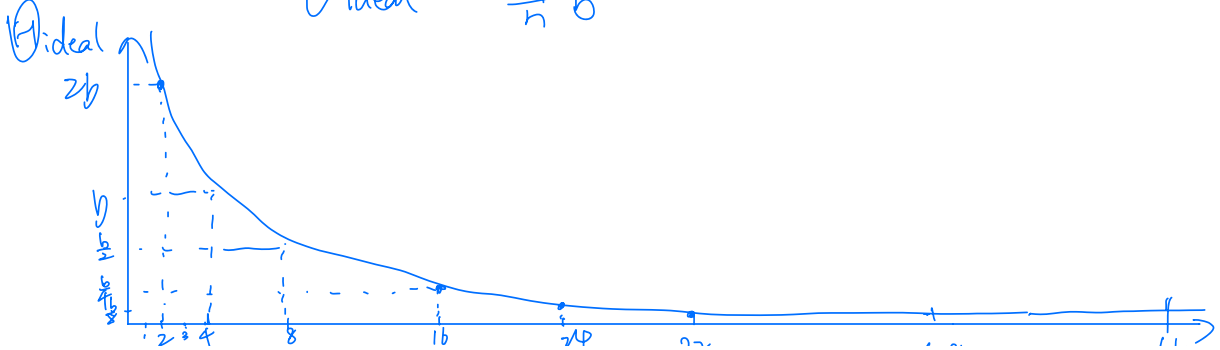
$$H_{min} = \frac{nk}{4} = \frac{4\times 2}{4} = 2$$

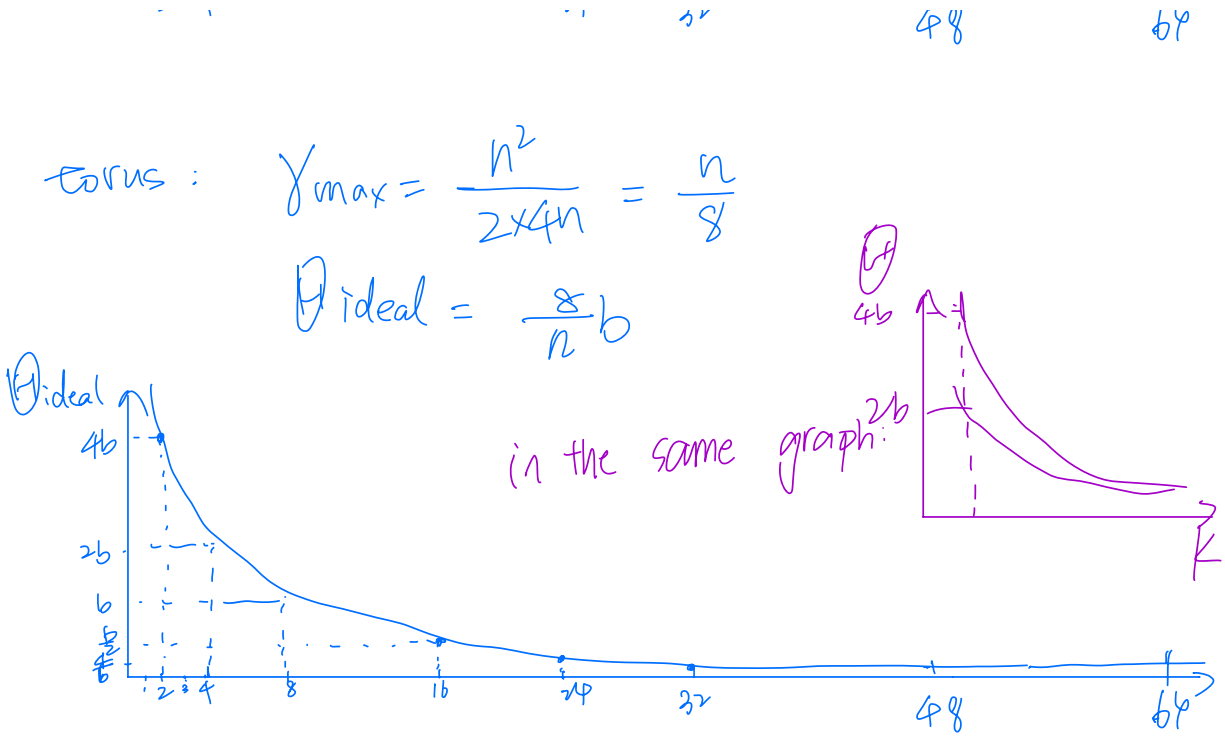
$$\gamma_{max} = \frac{H_{min}\cdot N}{C} = \frac{2\times 16}{4\times 8\times 2} = 0.5$$

$$\theta_{ideal} = \frac{b}{\gamma_{max}} = 2b$$

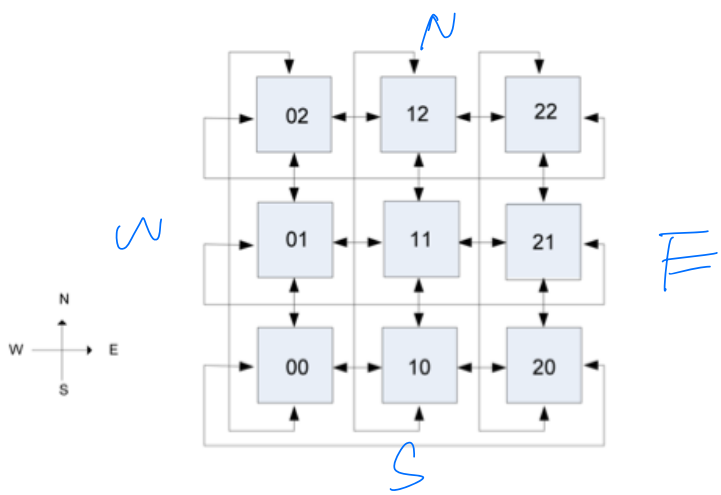
c) mesh: $\gamma_{max} = \frac{n^2}{2Bc} = \frac{n^2}{2\times 2n} = \frac{n}{4}$

$$\theta_{ideal} = \frac{4}{n}b$$





9. (Routing mechanism) Given a 3-by-3 torus network below. Packets are all delivered along the shortest paths.



a) Assume node-table routing, draw the routing table for node 11 and node 20. If there are multiple shortest paths, the table should allow selecting one path randomly.

b) Assume source-table routing, give routes for all packets starting from node 11, and node 22, with destinations to all other nodes.

a) routing table for node 11

	From			
to	11		20	
00	W	S	E	\
10	S	\	W	\
20	E	S	X	X
01	W	\	E	N
11	X	X	W	N
21	E	\	N	\
02	W	N	E	S
12	N	\	W	S
22	E	N	S	\

b) source routing table for node 11

Destination	Route 0	Route 1

00	SWX	WSX
10	SX	NNX
20	ESX	SEX
01	WX	EEX
11	X	X
21	EX	WWX
02	WNX	NWX
12	NX	SSX
22	ENX	NEX

for node 22

Destination	Route 0	Route 1
00	NWWX	WWNX
10	NWX	WNX
20	NX	SSX
01	NWWWX	WWWNX
11	NWX	WNX
21	SX	NNX

02	WWX	EX
12	WX	EEX
22	X	X

10. (Flow control) A partial network with three nodes is shown in Figure 3. A 4-flit packet (1 Head flit, 2 Body flits and 1 Tail flit) is to be sent from node R1 to node R3, as indicated in Figure 3a. Sending head flits to the downstream router takes 3 cycles, and sending other flits takes 1 cycle.

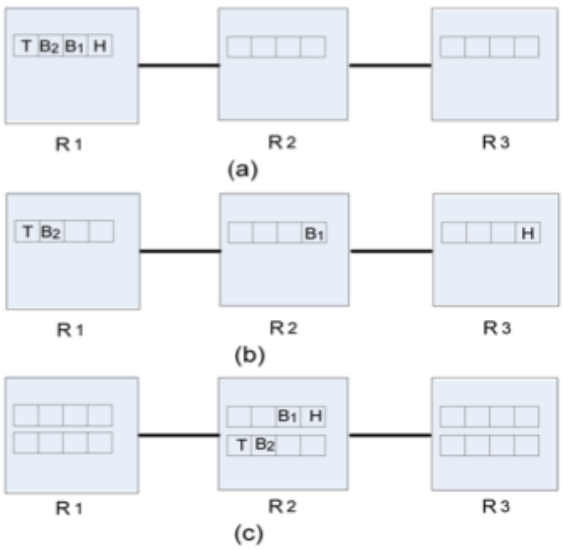


Figure 3. Flow control in a partial network with 3 nodes

a) If the network uses Store-and-Forward flow control, will the situation in Figure 3b happen?
b) If the network uses Cut-through flow control, will the situation in Figure 3b happen? If no contention, how many cycles does it take until R3 receive the tail flit?
c) If the network uses Wormhole flow control, will the situation in Figure 3b happen? If no contention, how many cycles does it take until R3 receive the tail flit?
d) If the network uses Virtual channel flow control, will the situation in Figure 3c happen?
e) If all other packets in the network are of 20 flits long, what is the minimum buffer size (in flits) for Store-and-Forward, Cut-through, Wormhole, and Virtual Channel flow control, respectively?

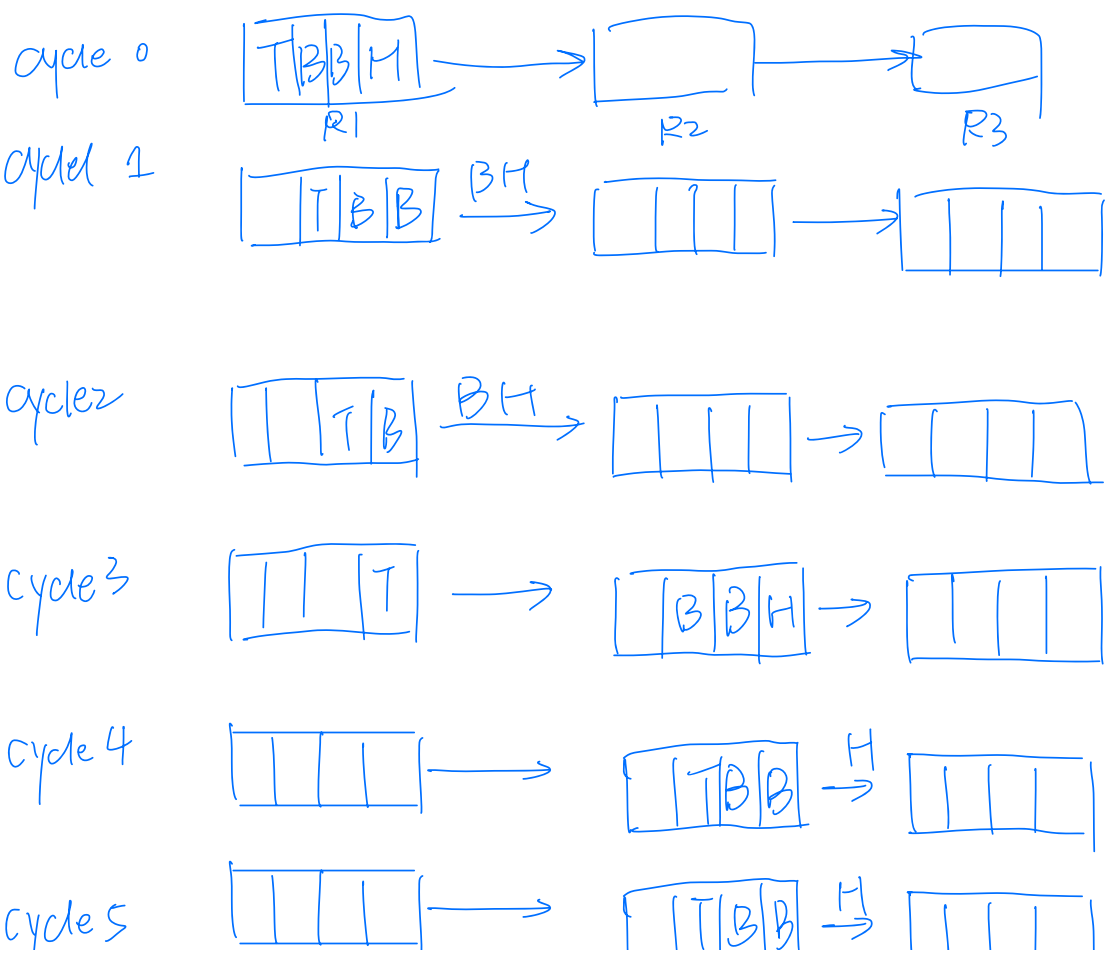
a) NO. in SAF flow control, each node along a route waits until a packet is completely received and the packet is forward to the next node. ✓

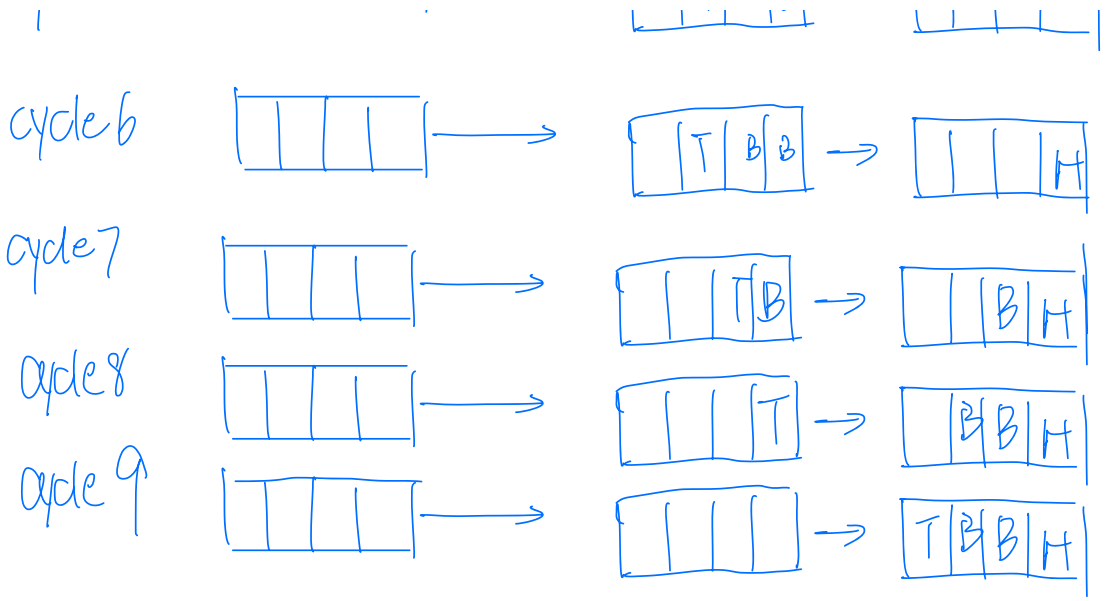
b) NO. In Cut-Through flow control, the transmission on

the next channel starts as soon as possible without waiting for the entire packet to be received. But it still operates on the whole packet. But by the time H reaches R3, T and R1 should be in R2 instead of R1

9 cycles

c) NO. Wormhole flow control operates like cut-through. but with channel and buffers allocated to flits rather than packets. Same reason as b)





9 cycles

d) NO. In virtual channel flow-control, several virtual channels are associated with a single physical channel. But the whole packet should always stay in the same channel.

e) store-and-forward: 20 flits

Cut-through: 20 flits

Wormhole: 1 flit

Virtual channel flow control: 2 1-flit buffer