

1. For a 64 processor system, compare the interconnection network for each of the following topologies. (We normalize the bandwidth of a single link to “1”).

	Bus	Ring	Torus	6-d Hypercube	Fully Connected
Total # of Switches	-				
Links per Switch	-				
Total # of links	1				
Network Bandwidth	1				
Bisection Bandwidth	1				

2. The above table focuses on the overall characteristics of different interconnection networks. If we focus on a single data transmission of n bytes between two processors (the *source* and *destination*), then transmission time is effected by:

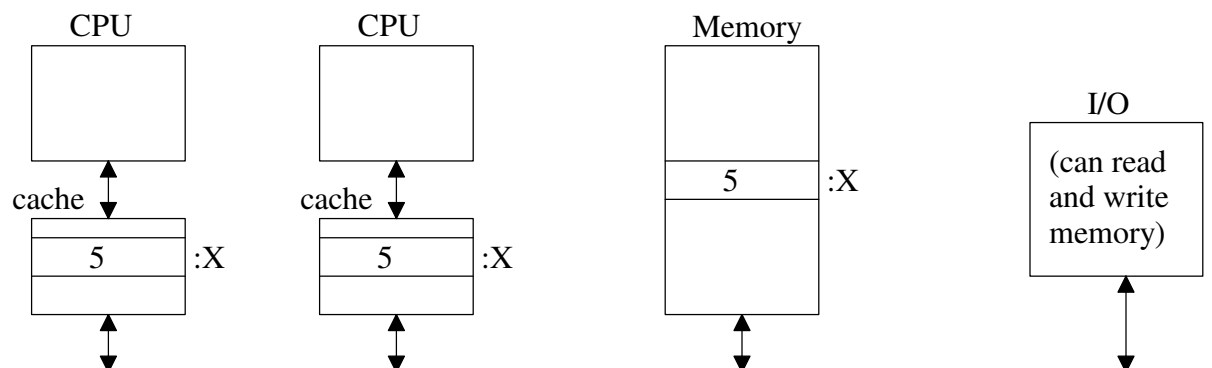
- *latency* (l) - the time that elapses between the source's beginning to transmit the data and the destination's receiving the first byte of data.
 - *bandwidth* (b) - the rate at which the destination receives data after it has started to receive the first byte (i.e., b B/sec.)
- a) What is the formula for transmitting an n bytes message between a *source* and *destination* with a bandwidth of b B/second?

message transmission time =

b) What components in the above table effect the latency?

c) What components in the above table effect the bandwidth?

Cache Coherency Solution - bus watching with write through / Snoopy caches - caches eavesdrop on the bus for other caches write requests. If the cache contains a block written by another cache, it take some action such as invalidating it's cache copy.



The MESI protocol is a common write-back cache-coherency protocol. Each cache line is marked as: Modified, Exclusive, Shared or Invalid.

	Modified	Exclusive	Shared	Invalid
This cache line valid?	Yes	Yes	Yes	No
The memory copy is ...	out of date	valid	valid	-
Copies exist in other caches?	No	No	Maybe	Maybe
A write to this line ...	does not go to the bus	does not go to the bus	goes to the bus and update cache	goes directly to bus

3. How can distributed shared memory machines do cache coherency?

Amdahl's law expresses the limitations of parallelization due to the existence of non-parallelizable computations. If $1/S$ of the computation is inherently sequential, then the maximum speedup performance improvement (speedup) is limited to a factor of S .

$$speedup = \frac{\text{sequential execution time}}{\text{parallel execution time}} = \frac{T_S}{T_P}$$

T_S

s_1	p_1	s_2	p_2	s_3
-------	-------	-------	-------	-------

P processors T_P

s_1	p_1/P	s_2	p_2/P	s_3
-------	---------	-------	---------	-------

"infinite" processors

s_1	s_2	s_3
-------	-------	-------

4. If $1/S$ is the fraction of sequential program that is non-parallelizable, what is the formula for the T_P assuming linear speedup of the parallelizable portion of the sequential program?

$T_P =$

5. What are the four major categories of performance loss that prevents *linear speedup* (i.e., P processors speeding up a computation by a factor of P)?