

CACHE COHERENCE

① Cache Coherence -

Two processors can have two different values for the same memory location.

Memory system is coherent if any read of a data item returns the most recently written value of data item.

Coherence defines values returned by a read, Consistency determines when a written value will be returned by a read.

→ Basic schemes for enforcing coherence one -

Migration and Replication.

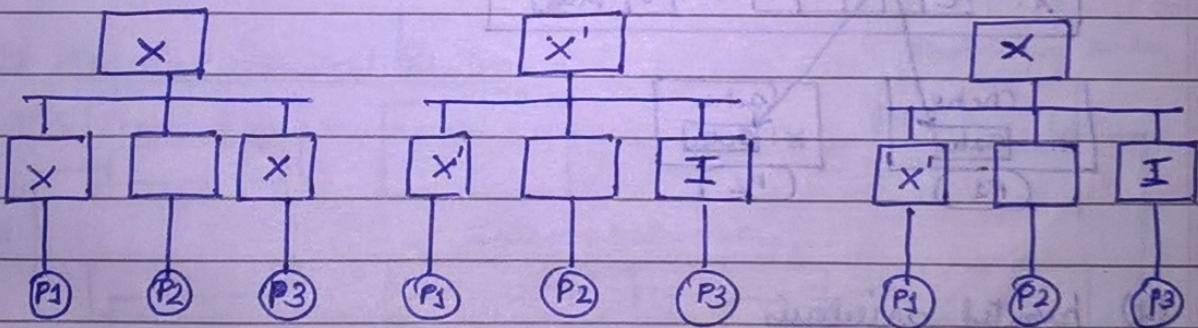
② Cache coherence protocols -

Protocols to maintain coherence for multiple processors. Key to implementing a cache coherence protocol is tracking the state of any sharing of a data block. Two different techniques are -

→ Snoopy protocols - No centralized directory, designed for bus connected system. Two types -

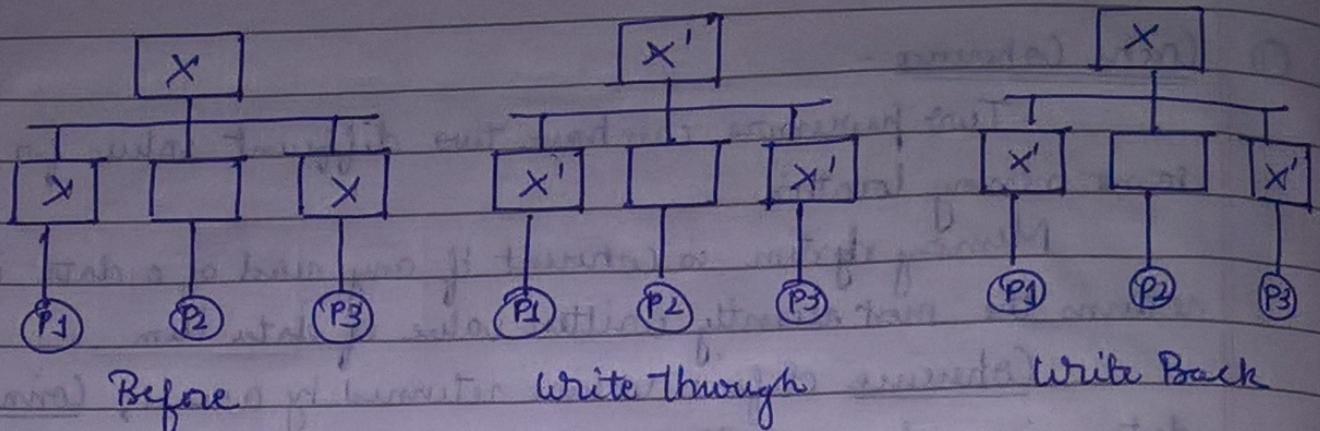
(1) Write-invalidate-

The processor that is writing data causes copies in the caches of all other processors in the system to be rendered invalid before it changes its local copy.



(2) Write-update - (Write-Broadcast)

The processor that is writing the data broadcasts the new data over the bus. All caches that contained copies of the data are then updated.



→ Problem with Snoopy bus protocol -

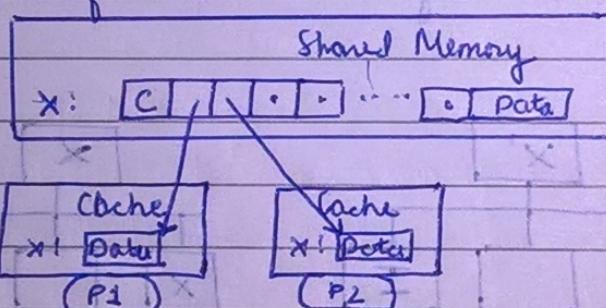
- (1) Cannot be used with for a multistage network
- (2) System bus is not available for snooping
- (3) Snoopy bus protocols at a remote node increase delays there
- (4) This increases latency and reduces memory bandwidth.

→ Directory based Protocols -

Sharing status of a block of physical memory is kept in just one location, called the directory. Applied to network connected system. Three types -

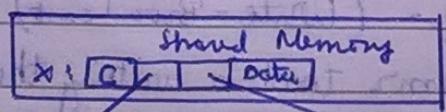
(1) Full-map directories -

Each directory entry can identify all processes with cached copies of data.



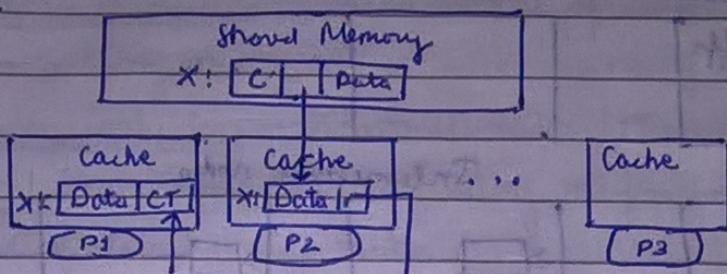
(2) Limited directories -

Each entry has a fixed number of process identifiers, regardless of the system size.



(3) Chained directories -

Emulate full-map directories by distributing entries among the caches.

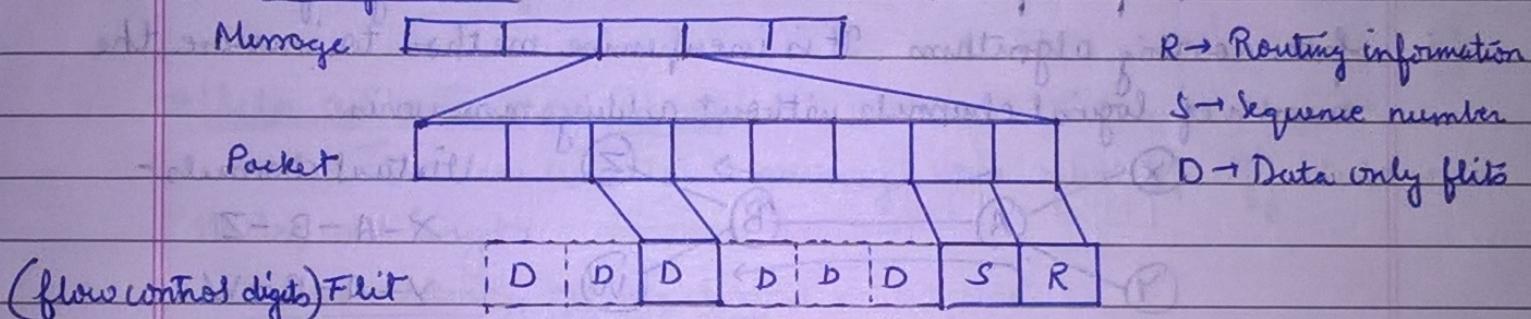


→ Limitations of directory-based protocols -

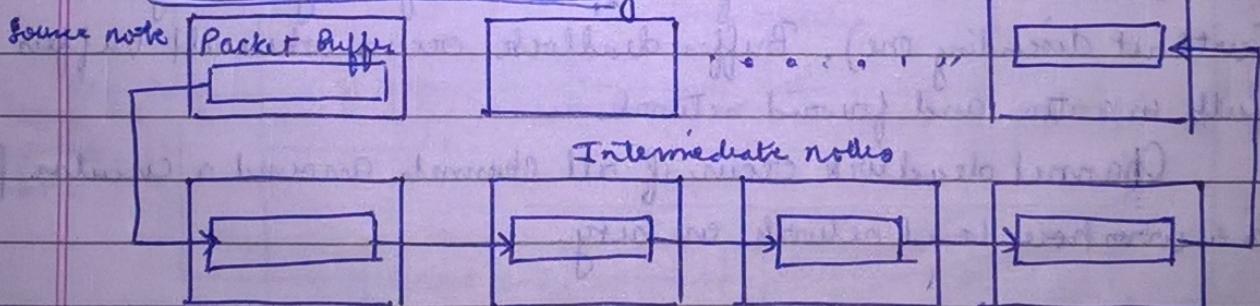
- (1) limited capacity for replication
- (2) Cost of complex design implementation when using hardwired controls
- (3) limitations on physical address space to map the information.

(3) Message Routing Schemes in multicomputer network -

→ Message formats -



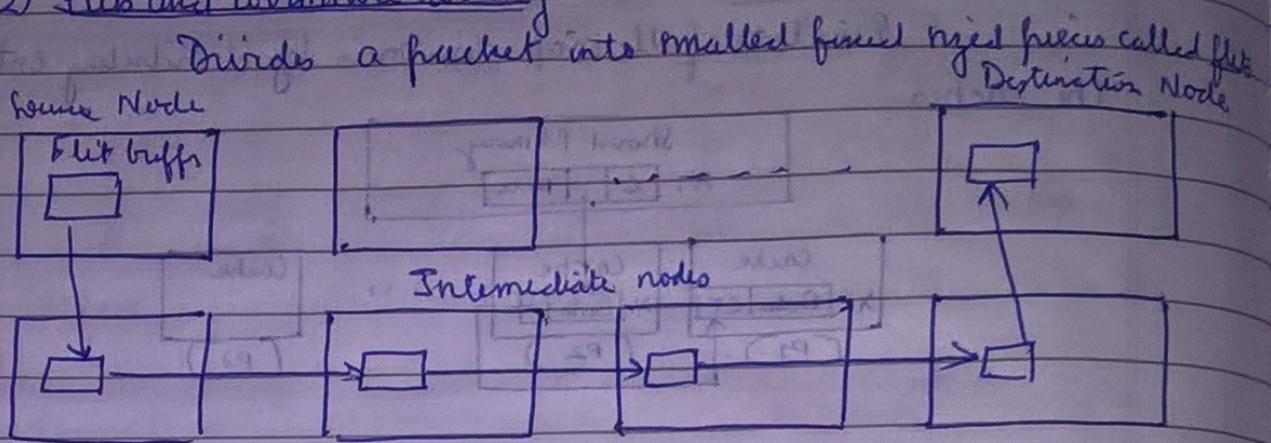
(1) Store and Forward Routing -



Advantages → simple, suitable for interactive traffic, bandwidth on demand

Disadvantages → Buffers for every packet, potential long latency, potential deadlock.

(2) Flit and wormhole routing -



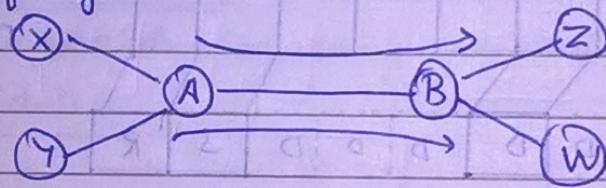
Advantages - Good for long messages, reduced need for buffering, reduced effect of path length.

Disadvantages - Possibility for deadlock, inability to support backtracking.

(4) Deadlock and virtual channels -

→ Virtual channels

A principle introduced to allow the design of deadlock free routing algorithms. It is inexpensive method to increase the number of logical channels without adding more wires.



Virtual channels -

X-A-B-Z

Y-A-B-W.

→ Deadlock -

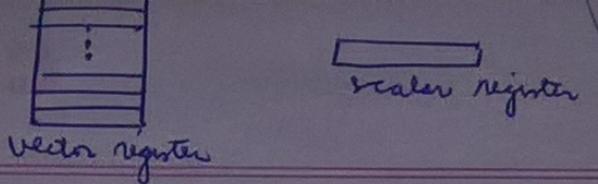
Deadlock can occur if it is impossible for any messages to move (without discarding one). Buffer deadlock occurs when all buffers are full in a store and forward network.

Channel deadlock occurs if all channels around a circular path in a wormhole-based network are busy.

(5) Vector processing principles -

→ Vector instruction types -

① Vector-Vector instructions - One or ~~more~~ ^{true} vector operands are fetched from the respective vector registers.



$$f_1: V_i \rightarrow V_j$$

$$f_2: V_j \times V_k \rightarrow V_l$$

② Vector-scalar instruction - obtain one operand from scalar register and one from a V register.

③ Vector-memory instruction - transmit data between memory and V register

④ Vector-reduction instructions - finding maximum, minimum, sum, mean value of elements in a vector

⑤ Gather and scatter instructions -

Gather - fetches from memory the non-zero elements of a sparse vector using indices that themselves are indexed

Scatter - storing into memory a vector in a sparse vectors whose non-zero entries are indexed.

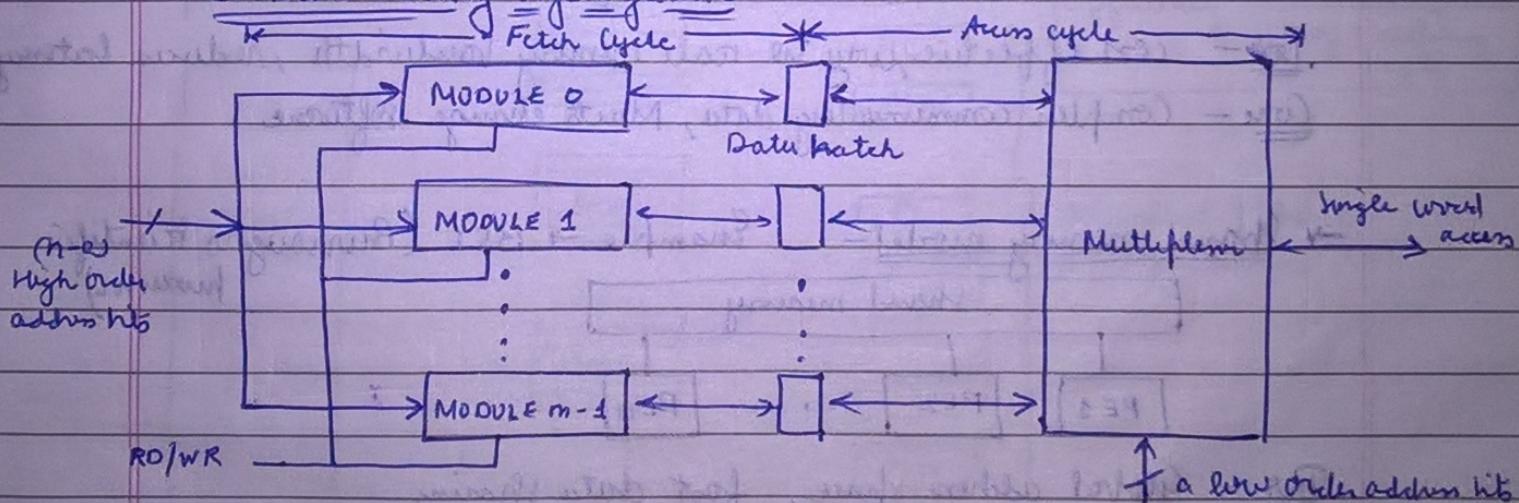
⑥ Masking instruction -

The mask vector is used to compress or to expand a vector to a shorter or longer index vector.

→ Vector address memory schemes -

To access a vector in memory, one must specify its base, stride, and length.

S-Access Memory organization



S-Access organization for an m-way interleaved memory

⑥ Vector supercomputer architecture -

Most supercomputers are clusters of SIMD multiprocessors, each processor of which is SIMD.

A SIMD processor executes the same instruction on more than one set of data at the same time.

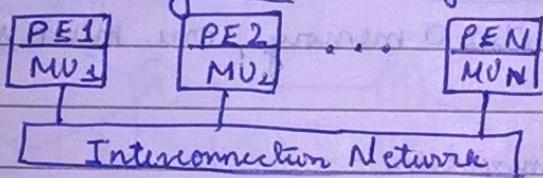
MIMO is employed to achieve parallelism, by using a number of processors that function asynchronously and independently.

Features -

- (1) More than one CPU
- (2) Large storage capacity
- (3) Very fast I/O capability.
- (4) Cryogenic fluids are used for cooling.
- (5) Unix / Linux operating system used.
- (6) FORTRAN language is preferred.

⑦ SIMD organization -

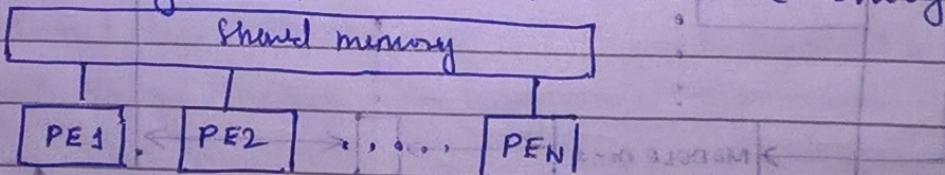
→ Distributed memory model - Eg - Illiac IV.



Pro - cost effective, way to scale memory bandwidth, reduces latency.

Con - Complex communicating data, Must change software.

→ Shared memory model - Example → BSI (Burrough's Scientific Computer)



Pro - Global address space, fast data sharing.

Con - lack of scalability, responsibility for synchronization, expensive.

⑧ Principles of multithreading -

Software multithreading - software that is aware of more than one processor/core and can use these to be able to simultaneously complete multiple tasks.

Hardware multithreading - Allows multiple to share the functional units of a single processor in an overlapping fashion.

⑨ Multithreading Issues and Solutions -

→ Problem of asynchrony -

Triggers two fundamental latency problems are remote loads and synchronization loads.

Solution of remote loads - cost of thread switching should be much smaller than that of the latency of the remote load.

Solution of synchronization load ~~distributed caching~~

A large continuation name space is provided to name an adequate number of threads waiting for remote responses.

⑩ Multiple Context Processors -

Multithreaded systems are constructed with multiple context processors.

