

Assignment No 8.

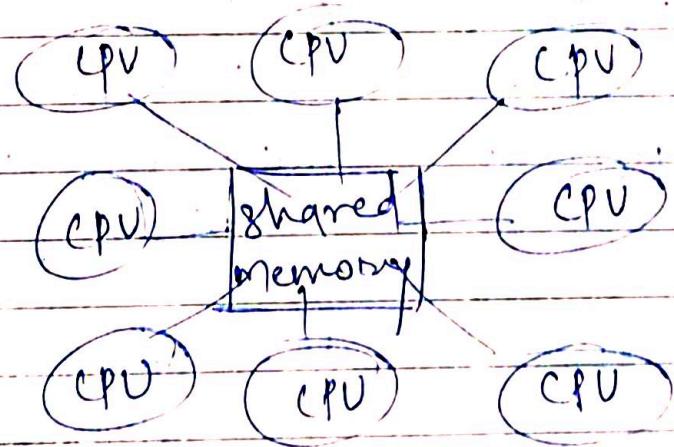
Q.1. What is difference between multiprocessor and multiple computer system? What are advantages of loosely coupled architecture over tightly coupled architecture?

Ans → Multiprocessor: It is computer system with two or more central processing unit (CPUs) share full access to a common RAM.

The main objective of using a multiprocessor is to boost the system's execution speed, with other objective being fault tolerance and application matching.

① Shared memory multiprocessor

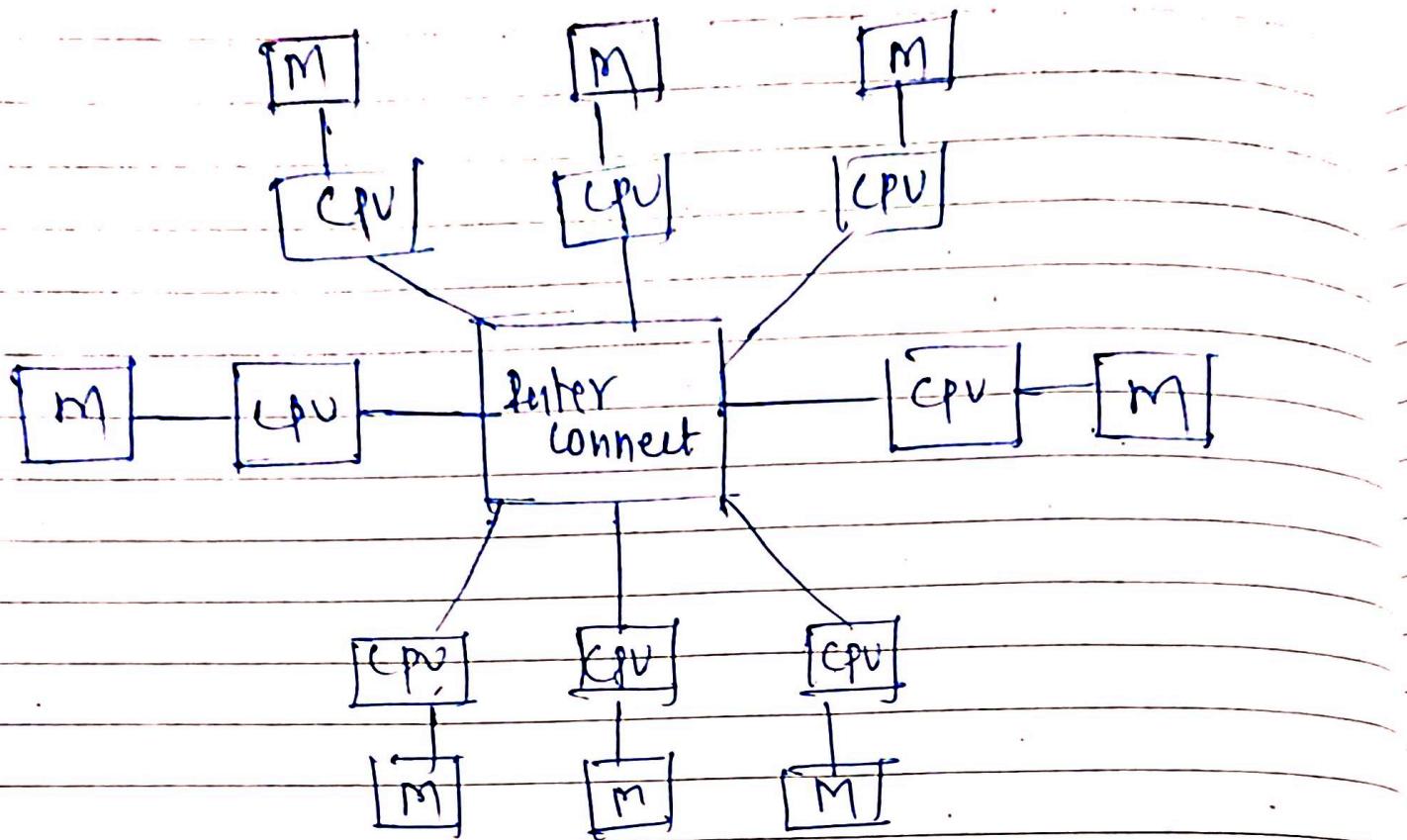
② Distributed memory multiprocessor



Multicomputer -

It is a computer system with multiple processors that are connected together to solve a problem.

Each processor has its own memory and it is accessible by that particular processor and those processors can communicate with each other via interconnection network.



Multiprocessor

A system with two or more CPU's that allows simultaneous processing of program

Easier to process

More difficult and costly to build

Multiprocessors supports parallel computing

Multicomputer

A set of processors connected by the communication network that works jointly to solve computation problem.

Less easy to program

Easier and cost effective to build.

Multicomputer's supports distributed computing.

Advantages of loosely coupled Multiprocessor system.

- 1) In this system, every processor has its own memory module.
- 2) It is efficient when there is less interaction between tasks running on different processors.
- 3) There are no memory conflicts in general.
- 4) It is considered as a Message transfer system (MTS).
- 5) It is less expensive.
- 6) It has a low data rate.
- 7) It has distributed memory.
- 8) They are usually seen in distributed computer systems.

Advantages of tightly coupled architecture over loosely coupled -

- 1) In this system, the processors share memory modules.
- 2) It is efficient when used with real-time processing.
- 3) It provides high speed.
- 4) It has memory conflicts.
- 5) They are connected through networks such as FMIN, IOPIN, ISIN.
- 6) It has high data rate.
- 7) It is expensive.
- 8) It is usually seen in parallel processing systems.

Tightly Coupled

① shared memory

② pmap, iopen, lsn

③ IOPCN helps to connect
processor + I/O devices

④ high data rate

⑤ most costly compare
to loosely coupled

⑥ chance of memory
conflict

⑦ high degree of interaction
between tasks

⑧ Appⁿ → parallel processing
system

Loosely coupled

distributed memory

Message Transfer System

processor + I/O devices
directly connected

low data rate

less costly in compare
to tightly coupled.

No chance Memory
conflict don't take place

low degree of interaction
between tasks

Appⁿ → distributed
computing system

Tightly Coupled Architecture (Shared Memory Archi) System

A multiprocessor is a tightly coupled computer system having two or more processing units (multiple processors) each sharing main memory and peripherals, in order to simultaneously process programs.

The throughput of the hierarchical loosely coupled multiprocessors may be too slow for some applications that require fast response times.

If high speed or real time processing is required the Tightly Coupled system may be used.
(TCS)

Two models - (1) Without cache (2) With cache.

(1) First model consists of -

(i) P-processors (ii) Memory modules (iii) input-output channels.

The above units are connected through a set of three interconnection networks -

PMIN → Processor Memory Interconnection Network

IOPIN → I/O processor interconnection network

ISIN - Interrupt signal Interconnection network

① The PMIN is a switch which can connect every processor to every module.

It has $p \times m$ set of cross points. It is a multistage network. A memory can satisfy only one processor's request in a given memory cycle.

Hence if more than one processor attempt to access the same memory module, a conflict occurs and it is resolved by the PMIN.

Another method to resolve the conflict is to associate a reserved storage area with each processor and it is called as ULM (Unmapped Local Memory). It helps in reducing the traffic in PMDN.

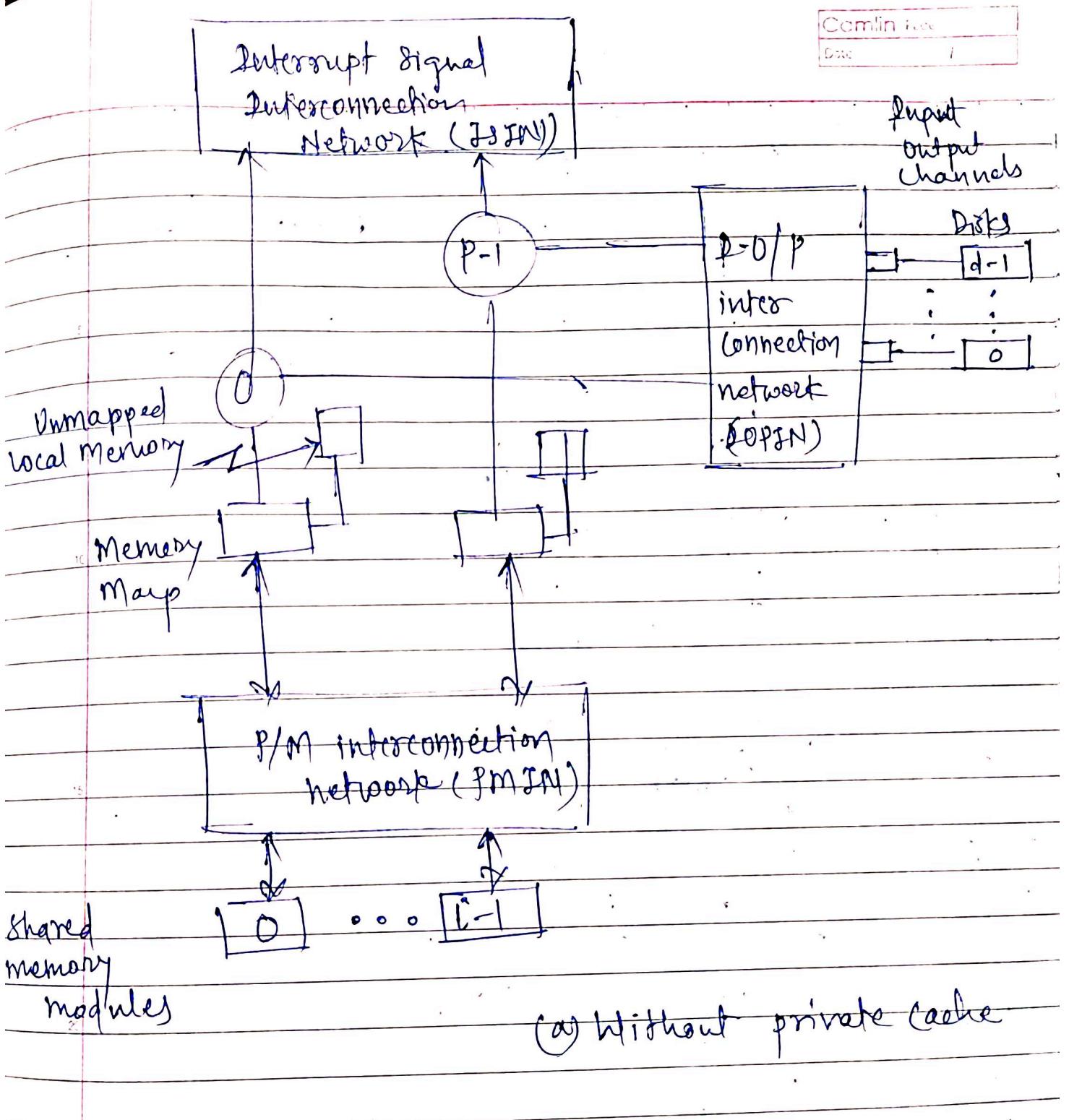
Since each memory reference goes through the PMDN, it encounters delay in the processor's memory switch so this can be overcome by using a private cache with each processor.

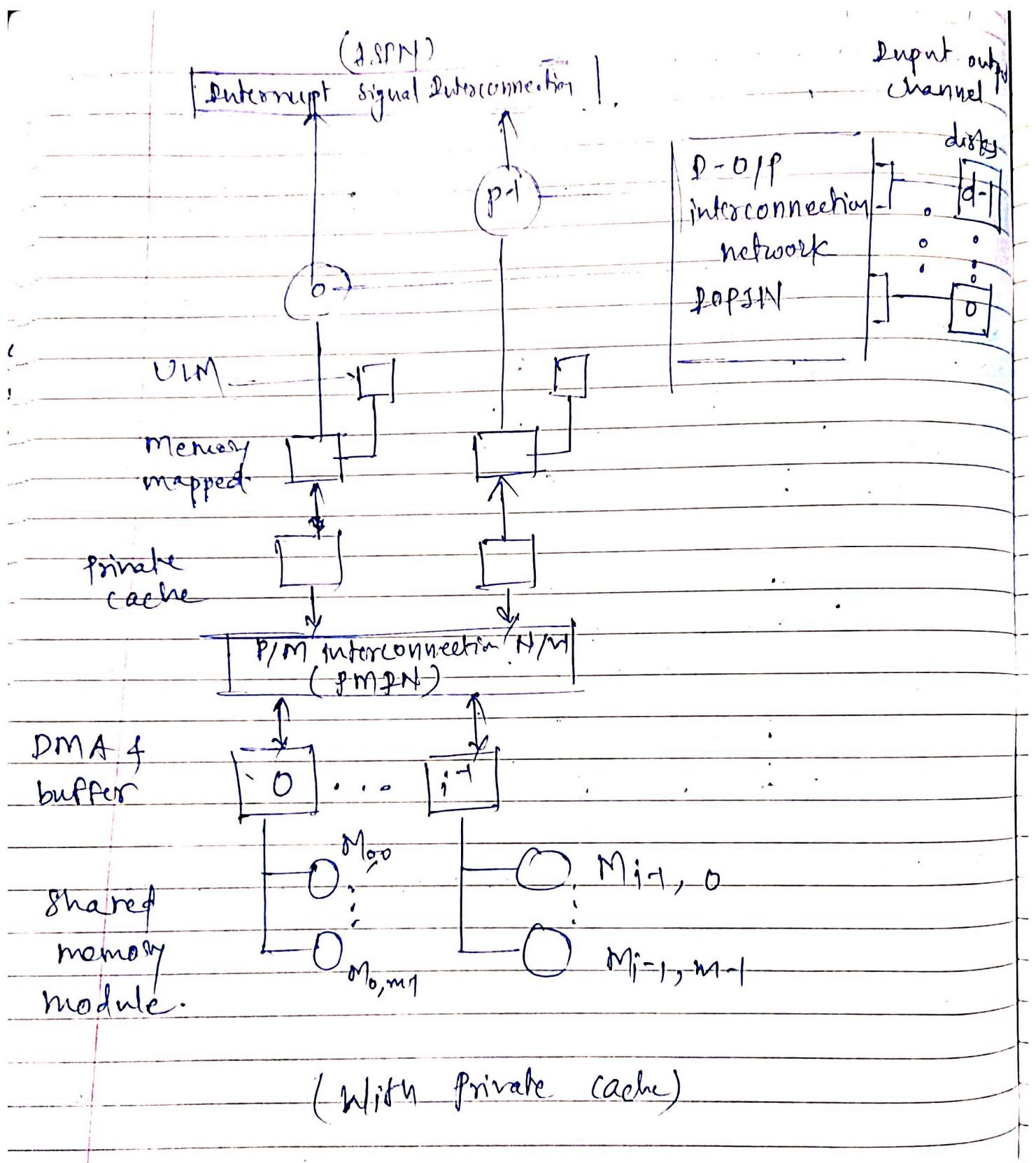
With Cache

Multiprocessor organization with cache encounters the cache coherence problem. More than one consistent copy of data may exist in the system.

In the figure, there is a module attached to each processor that directs the memory reference to either the ULM or the private cache of that processor.

This module is called memory map and is similar in operation to local.





Loosely Coupled Multiprocessor

Each processor has a set of input-output devices and a large local memory.

Computer module: processor, local memory, i/o interface

Processes in different CM communities through MJS

Degree of coupling is loose which is determined by communication topology of MPs.

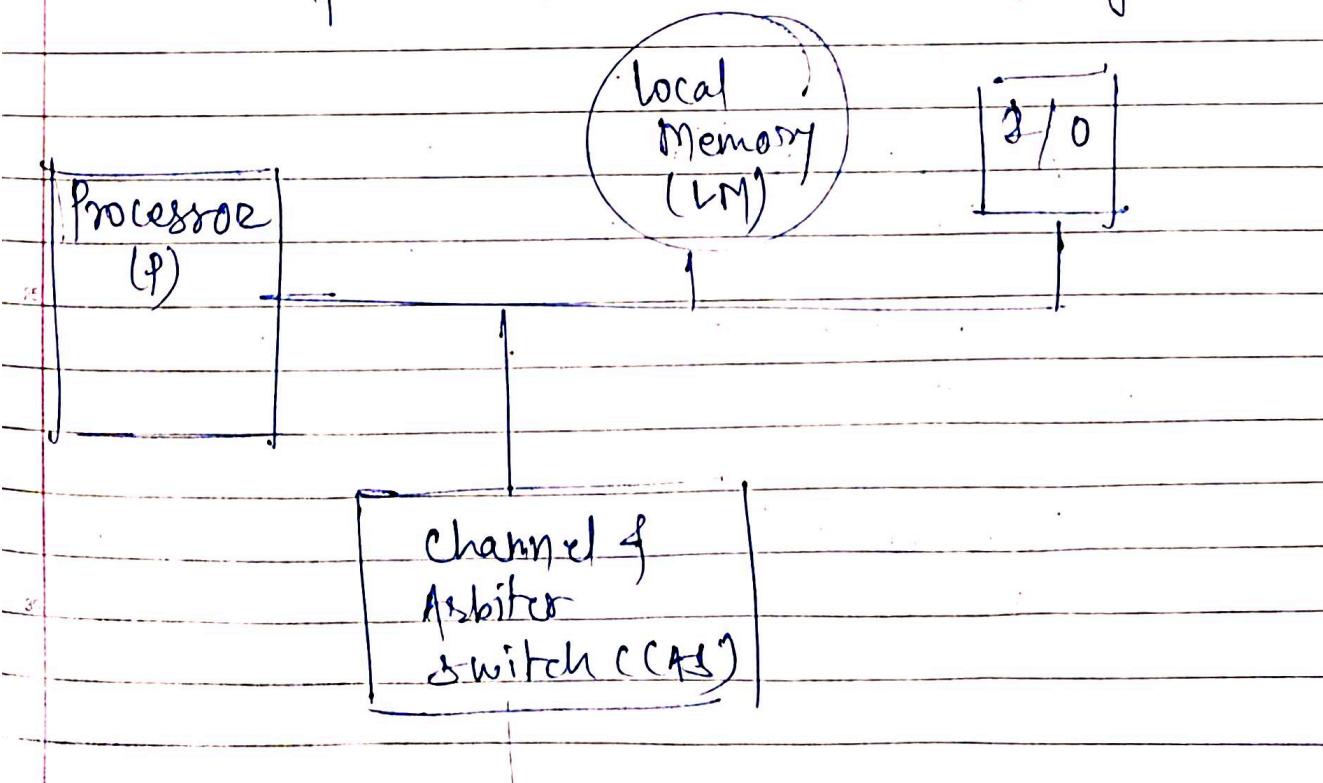
Efficient when interactions are minimum.

Computer Module -

Consist of processor, local memory, local input output devices and interface to other computer modules.

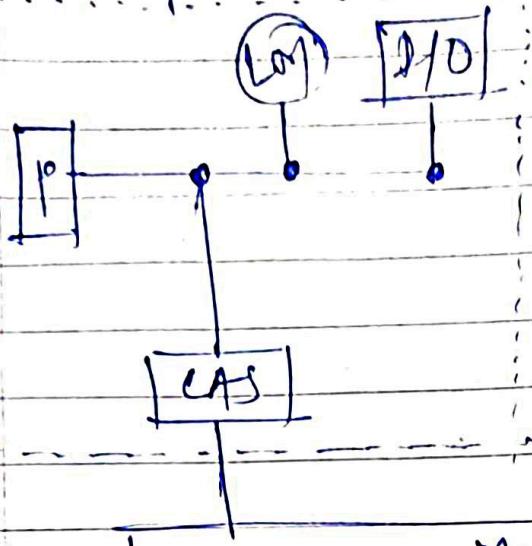
Interface contains a channel and arbiter switch (CAS)

Computer modules are connected through MJS.

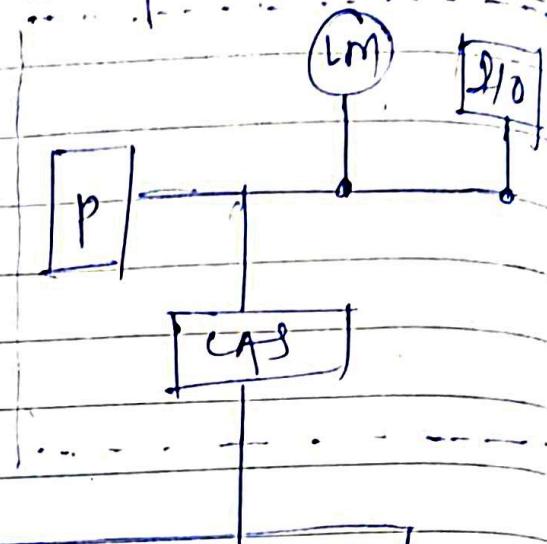


Computer Model connection

Computer module 0



Computer Module N-1



Message transfer system
(MTS)

Role of CAS -

In case of collision while accessing physical segment of the MTS, CAS choose one of the request according to service discipline.

Delays other requests until selected request is serviced completely.

A channel within the CAS may have high speed communicating memory.

Communication memory is used for buffering block transfer of images messages.

Communication memory is accessible to all the processors.

Role of MTS -

It determines the performance of multiprocessor systems.

MTS can be implemented as -

① simple time shared bus - performance is limited by the message arrival rate on the bus, message length and bus capacity.

② shared memory system - can be implemented with set of memory modules & processor memory interconnection networks main memory.

Performance is limited by memory conflict problem.

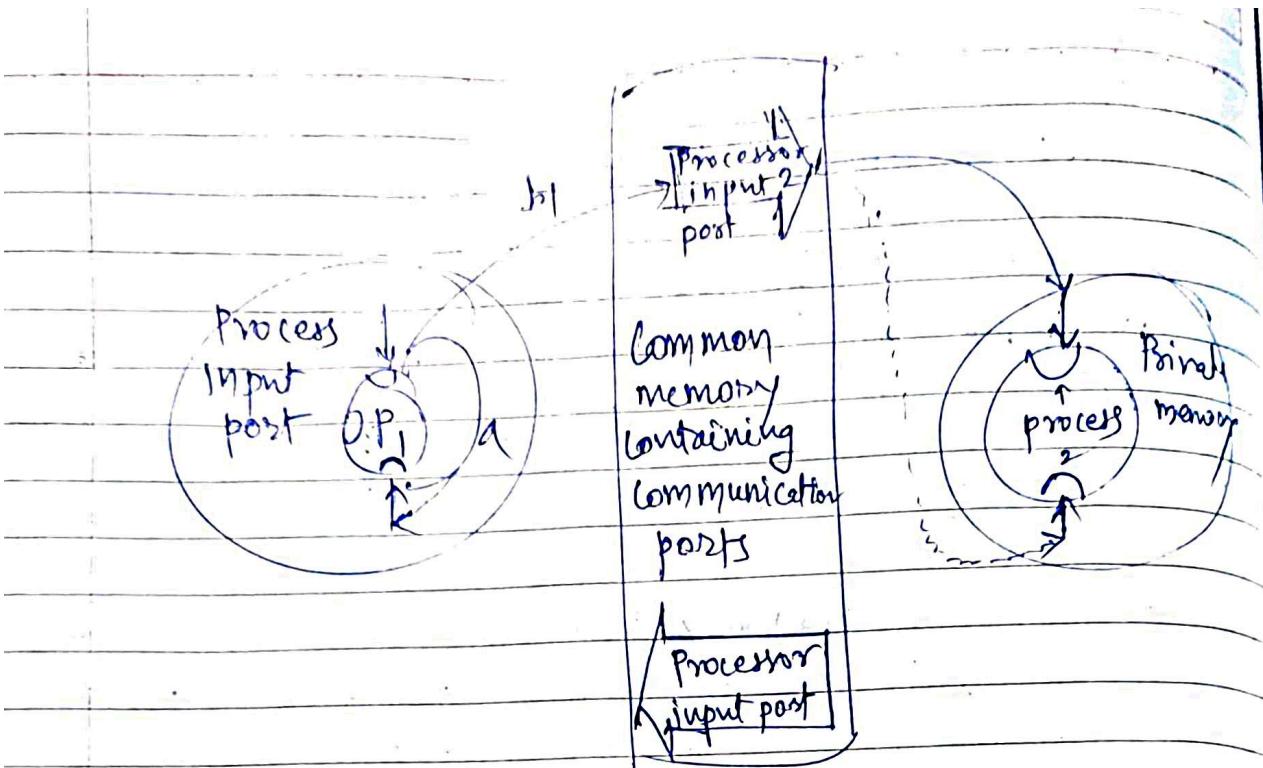
Communication memory types -

1. centralized connected to shared bus
2. As a part of shared memory

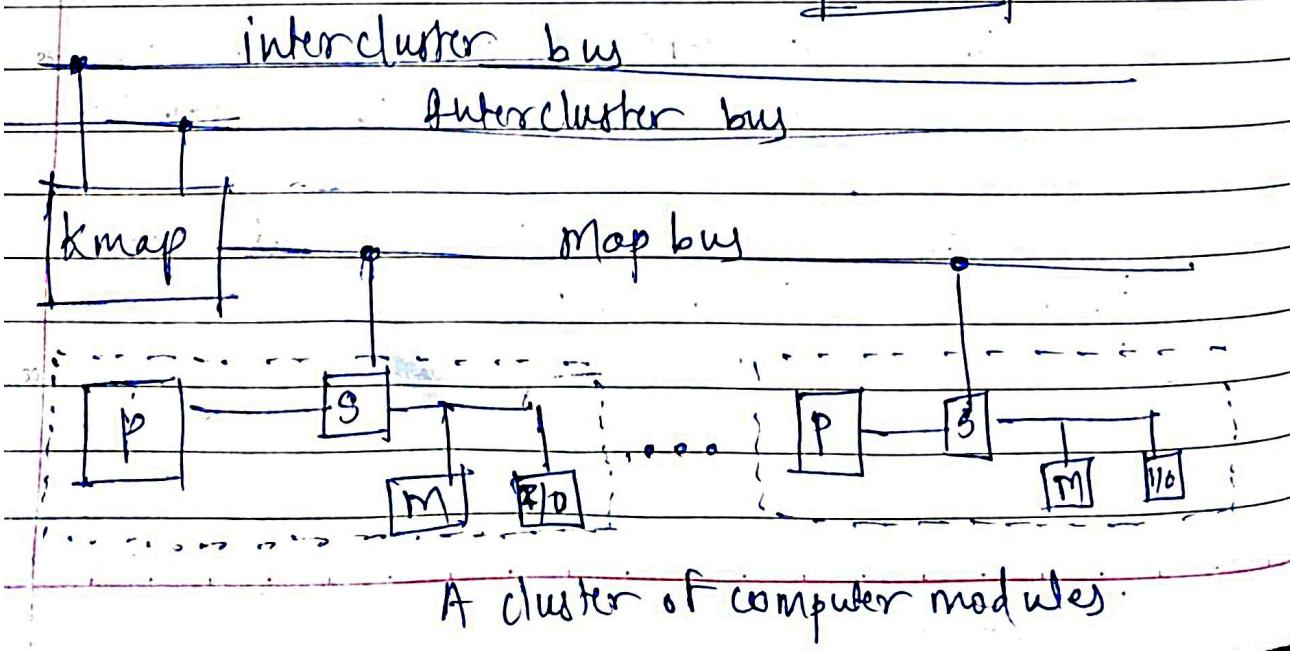
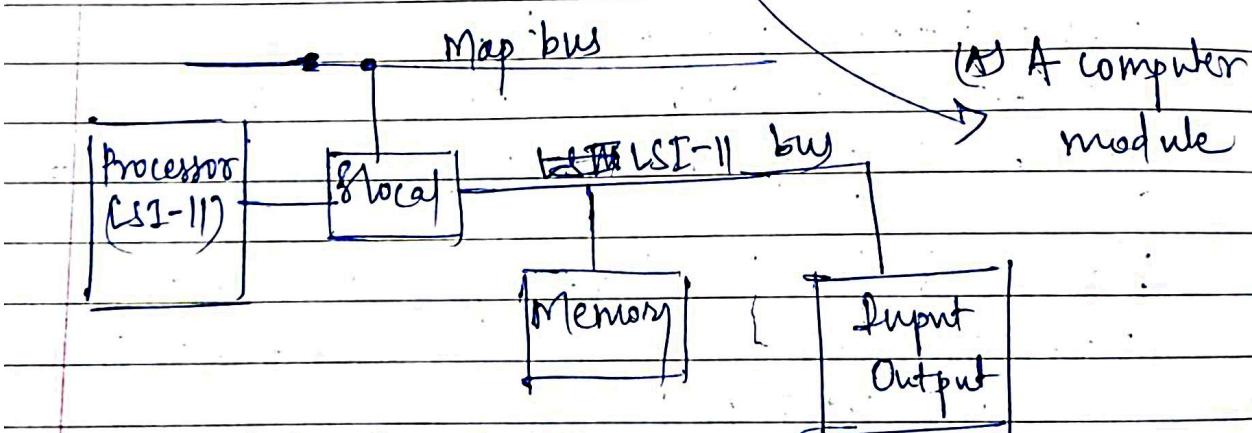
Each task has its input port in local memory. Task in same processor communicate through local input port.

Task allocated to different processors communicate through communication port in communication memory.

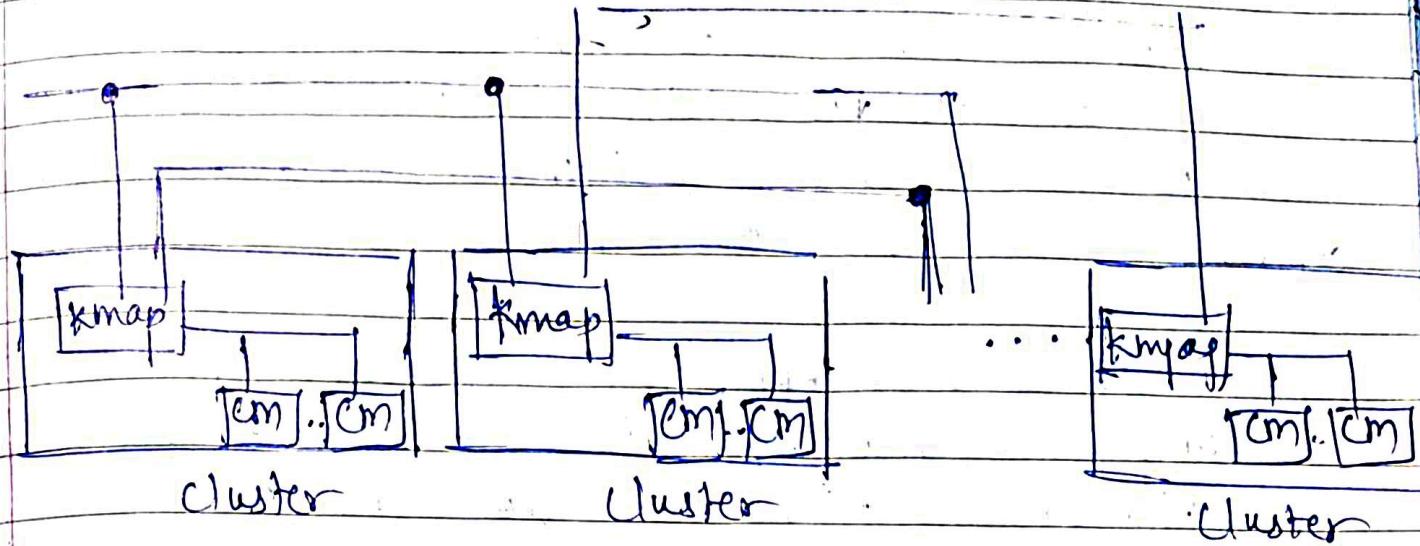
One communication port is associated with each processor as input port.



Draw and explain CM* architecture. What is Kmap? Explain function of Kmap.



Network of clusters



(Q) A network of clusters

Cluster - lowest level consists of computer modules, k-map and map bus

Enhance the cooperative ability of the processors in cluster to operate on shared data with low communication overhead

Provide hardware facilities to execute group of tightly coupled co-operating processes

kmap - handles any nonlocal reference to memory

Clusters' communication handled by inter cluster buses connected between kmaps.

CMX - based on loosely coupled system

8 local - (local switch) like (AS)

It intercepts and routes the processor requests to the memory & I/O devices outside the computer module via the map bus.

It also accepts the references from other computer modules to its local memory of I/O devices. This can be done by using map bus.

Cluster → Computer module + kmap + map bus

Cluster is considered to be low level made up of →

It can enhance cooperative ability of processor in a cluster to operate on shared data with low ~~level~~ communication overhead

It also provides hardware facilities to execute group of tightly coupled cooperative processes

→ Kmap - microprogrammed 150 nanosec. three processor complex with common data memory

- provides address mapping, communication & synchronization functions within the system

3 processors in kmap are

* kbus - bus controller arbitrates the requests to map bus

line - manages communication between kmap and other kmap

pmap - Mapping processor respond to requests from kbus & line

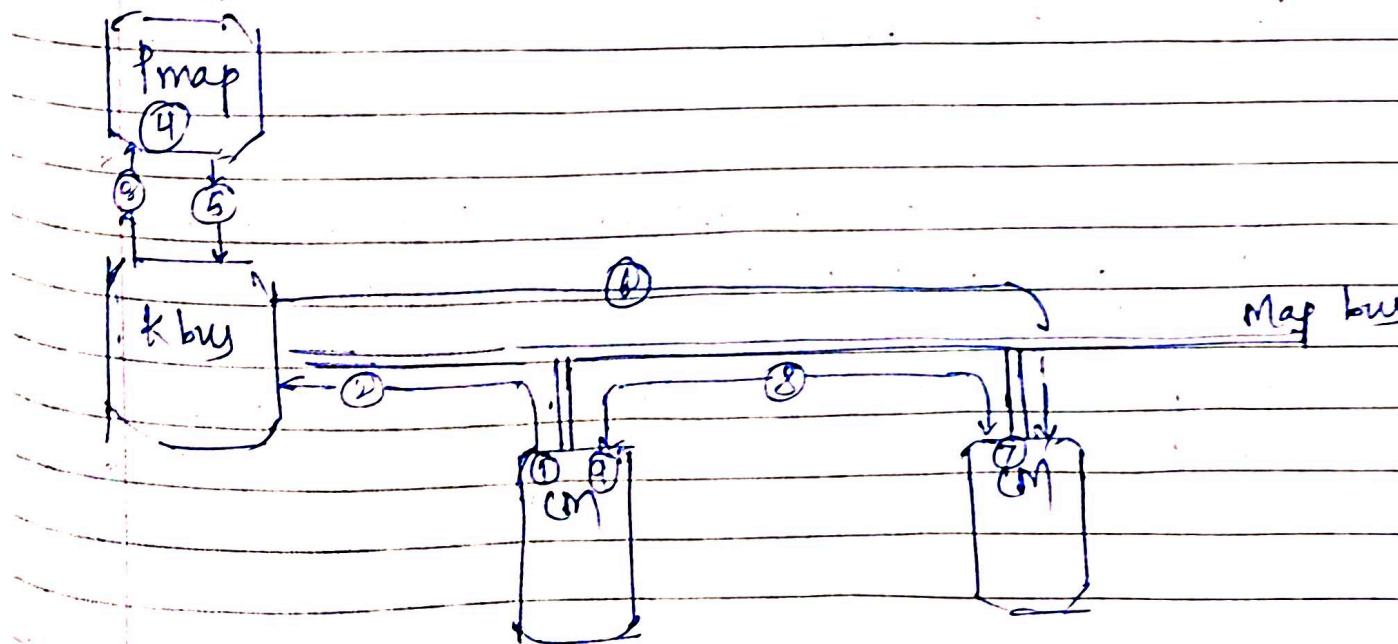
* perform most of request processing

* kmap is multiprogrammed to handle & concurrent requests (partitions) called context.

* Three sets of queues provides interfaces between kbus, line & pmap.

* kmap handles any & nonlocal reference in memory

Q.3. Intracluster communication takes place in com* architecture.



Steps

- ① processor initiate non local memory access
- ② kbus reads virtual address from master cm
- ③ context activation waits in run queue
- ④ pmap microsubroutine performs address translation.
- ⑤ request for memory cycle waits in outqueue
- ⑥ kbus sends physical address to destination cm.
- ⑦ destination cm steals memory cycle from its processor
- ⑧ kbus gates return result back to master cm
- ⑨ process continues.

Whenever processor of computer module makes nonlocal reference - Master cm

In response to service requests, kbus allocates pmap context and fetches virtual address of memory reference via mapbus.

Kbus activates new context containing virtual address.

Pmap context performs virtual to physical address translation.

Context is initiated invoking kbus operation by loading request in to kbus outqueue.

pmap makes context switch to another runnable context.

Kmap devices out requests by sending physical address via mapbus to destination CM.

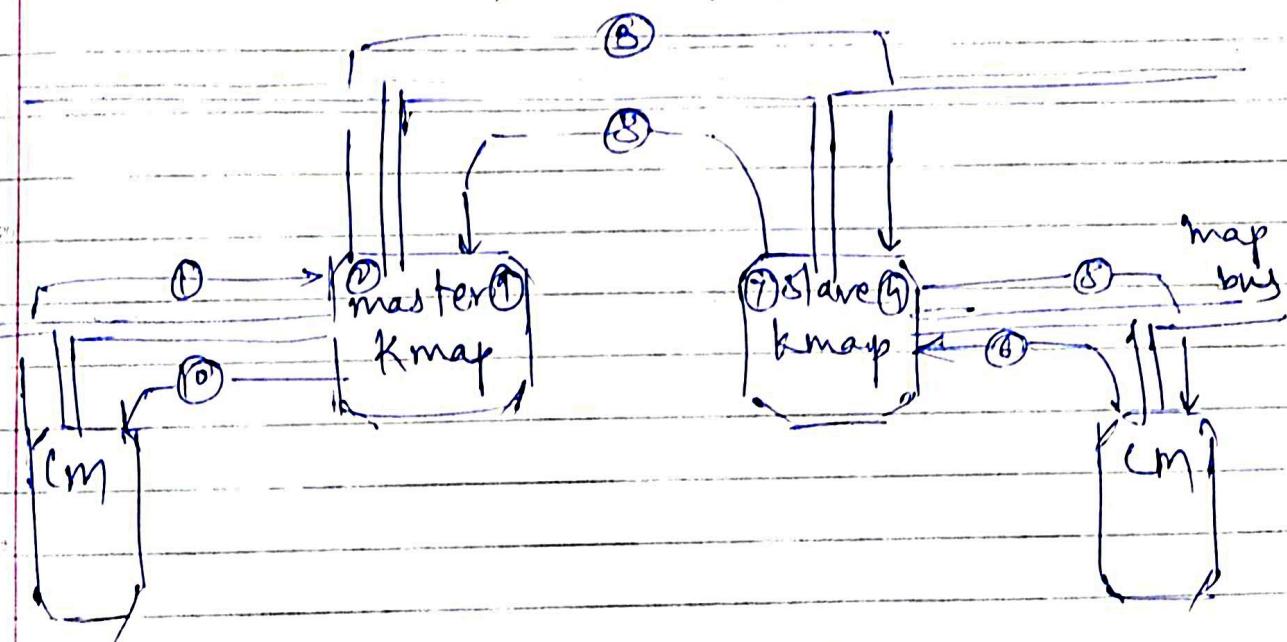
Destination CM completes memory access, signal return requests to Kmap.

* Intercluster memory access

It is done by passing messages via intercluster bus.

Messages are of two types:

1. Postload message - invokes new context at dest Kmap
2. Originating Return message - contain context number of the content to be reactivated.



① Master Kmap receives requests from master CM

② Master Kmap prepares intercluster messages

③ Messages travels to Slave Kmap

④ Slave Kmap decodes request

⑤ Request for memory cycle sent to destination CM

⑥ Return result sent back to Slave Kmap

⑦ Slave kmap prepares return intercluster mess

⑧ Message returns to master kmap

⑨ Master kmap receives return message

⑩ Result sent to master Cm

dynamic data flow

- In dynamic machines, data tokens are tagged to allow multiple tokens to appear simultaneously on any input or on operator node.
- The tagging is achieved by attaching a label with each token which uniquely identifies the scatterer of that token.
- Hence, additional hardware is needed to attach tags onto data tokens & perform tag matching.
- The dynamic dataflow allows greater exploitation of parallelism.