HPCA -
ASS - 1

1) unifunction pipeline -

1) A pipeline unit with a fixed and dedicated function such as the floating point adder is called unifunctional.

e.g → 2) The cray-1 has 12 unifunctional pipeline units for various scaler, vector, fixed point and floating point operations.

3) A multifunction pipe may perform different functions, either at diffn times or at the same time.

4) It is interconnecting different subsets of stages in the pipeline.

2) Data dependency Hazards -

1) Hazards are caused by resource-usage ~~conflicts~~ conflicts among various instruction in the pipeline.

2) Such hazards are triggered by interinstruction dependencies.

3) Method to cope with such hazards are needed in any type of lookahead processors for either synchronous-pipeline or asynchronous pipeline.

4) There are three classes of data-dependent hazards, according to various data update patterns : 1) write after Read (WAR) hazards
2) Read after write (RAW) hazards
3) write after write (WAW)

3) Pipeline Throughput :-

1) The no. of results (tasks) that can be completed by a pipeline per unit time is called its throughput.

2) This rate reflects the computing power of a pipeline. In terms of efficiency η and clock period τ of a linear pipeline we define the throughput follows:-

$$W = \frac{n}{k\tau + (n-1)\tau} = \frac{\eta}{\tau}$$

3) When n equals the total no. of tasks being processed during an observation period $k\tau + (n-1)\tau$

4) In the ideal case $\omega = 1/\tau = f$ when $\eta \to 1$.
This means that the maximum throughput of a linear pipeline is equal to its frequency, which corresponds to one output result per clock period.

4) Static pipeline:-

1) A static pipeline may assume only one functional configuration at a time.

2) Static pipeline can be either unifunctional or multifunctional.

3) Pipelining is made possible in static pipes only if instructions of same type are to be

5) most existing computers are equipped with static pipes, either unifunctional or multifunctional.
executed continuously.
ii) The function ~~perm~~ performed by a static pipeline should not change frequently. Otherwise, its performance may be very low.

5) Scalar pipeline-
i) Depending on the instruction or data types, pipeline processor can be also classified as a scalar pipeline and vector pipeline.
2) A scalar pipeline processes a sequence of scalar operands under the control of a Do loop.
3) Instruction in a small Do loop are often prefetched into the instruction buffer.
4) The required scalar operands for repeated scalar instructions are moved into a data cache in order to continuously supply the pipeline operands.
5) The IBM System 1360, Model 91 is a typical example of a machine equipped with scalar pipeline. However, the model 91 does not have a cache.

Q-2) Draw internal arch. of CPU pipeline structure of typical CPU. efficiency of linear pipeline.

~~Q-3) Draw architecture of cray-1~~

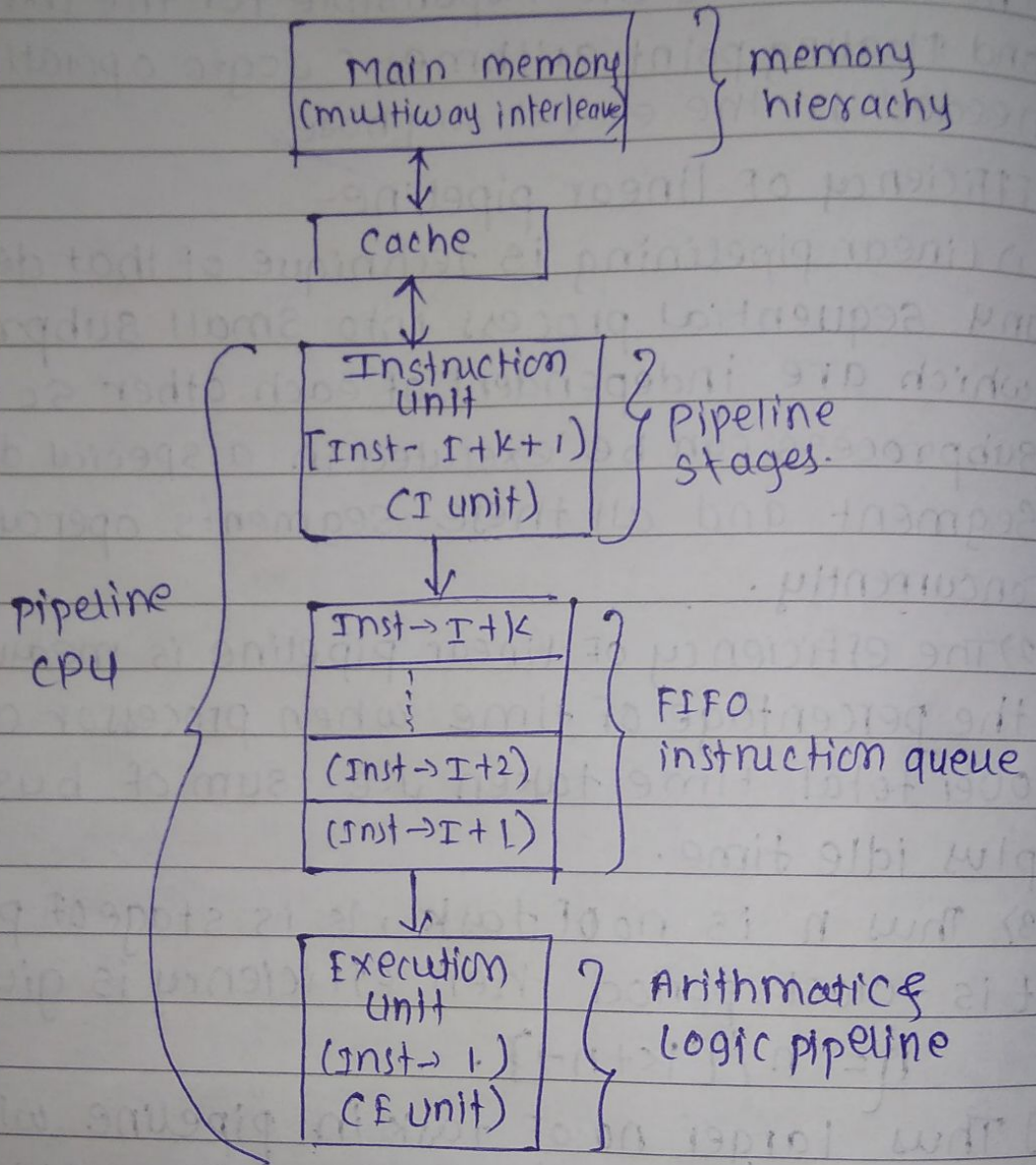


fig. The pipelined structure of typical CPU.

1) The central processing unit (CPU) of a modern digital computer can generally be partitioned into 3 sections → A] The instruction unit B] The instruction queue C] The execution unit.

2) Programs and data reside in the main memory which usually consists of interleaved memory modules.

3) The cache is faster storage of copies of program and data which are ready for execution. It is used to close up the speed gap between main memory and the CPU.

4) The instruction unit consist of pipeline stages for instruction fetch, decode, operand addr calculation and operand fetches.

5) The instruction queue is a FIFO storage area for decoded instruction and fetched operands.

6) The execution unit may contain multiple ~~functional~~ functional pipelines for arithmetic & logical fun^n.

Efficiency of linear pipeline-

1) Linear pipelining is technique of that decomposes any sequential process into small subprocess which are independent of each other so that each subprocess can be executed in a special dedicated segment and all these segments operates concurrently.

2) The efficiency of linear pipeline is measured by the percentage of time when processor are busy over total time taken i.e. sum of busy time plus idle time.

3) Thus n is no of task, k is stage of pipeline & t is clock period then efficiency is given by

$$\eta = n / [k+n-1].$$

4] Thus larger no of task in pipeline will be pipeline busy hence better will be efficiency.

The pipelined structure of a typical central processing unit ⟶

Q-3) Draw architecture of cray-1

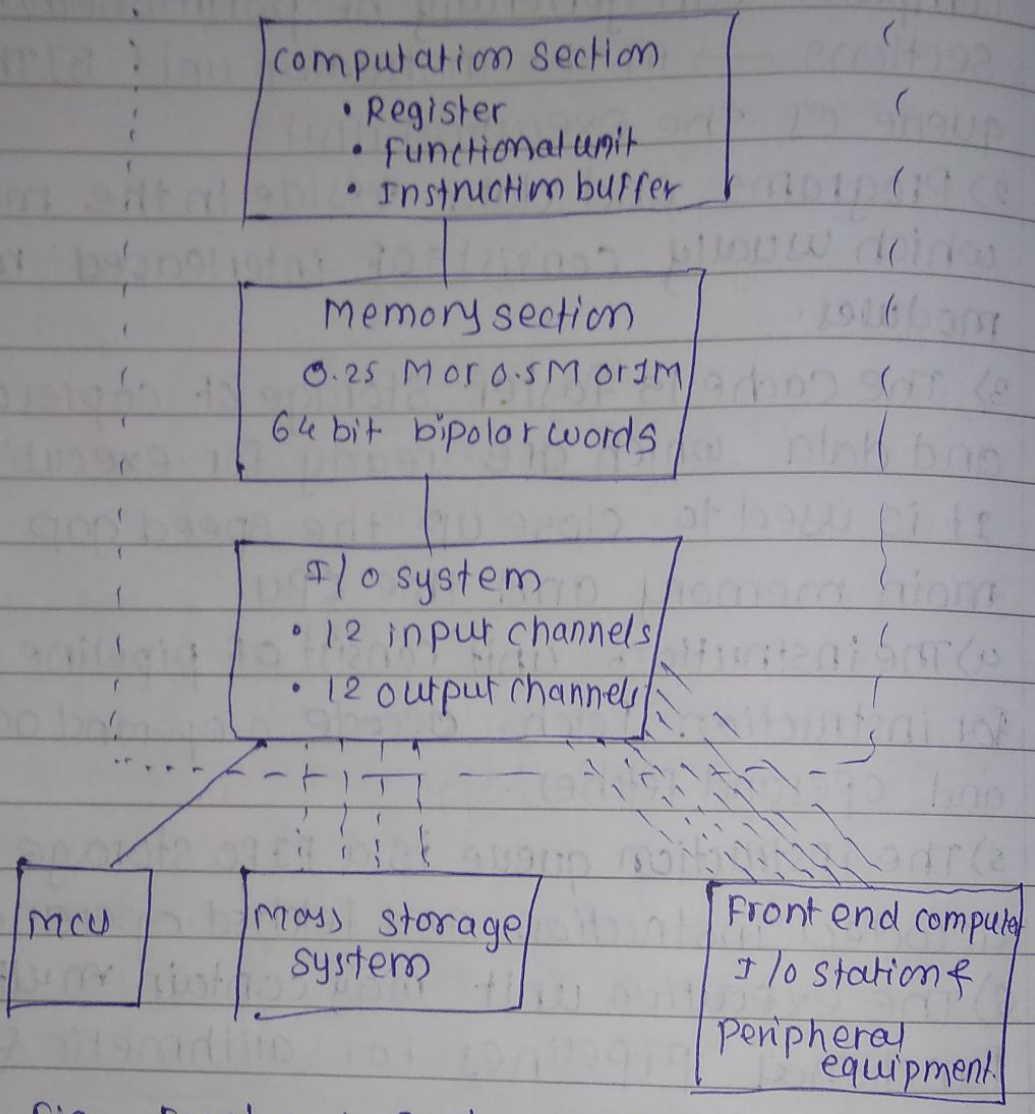


fig. Front end system & Cray-1 connection.

Q-4

=> The cray-1 has been available as the first modern vector processing since 1976. The architecture of cray-1 consist of no. of working registers, large instruction buffer, data buffer and 12 functional pipeline units.

2) with the chaining of pipeline units, interim results are used immediately once they become available.
3) The clock rate in the cray-1 is 12.5 ns
4) The cray is not a "stand alone" computer.
5) The CPU contains a computation section, a memory section and I/O section.
6) 24 I/O channels are connected to the front-end computer. The I/O stations, peripheral equipment the mass storage subsystem, and maintenance control unit (MCU).
7) The front-end system will collect data, present it to the cray-1 for processing, and receive output from the cray-1 for distribution to slower device.

Q-4 What are characteristics of vector processing? How vector instructions are classified?

1) A vector operand contains an ordered set of n elements. where n is called the length of vector.
2) Each element in vector is scalar quantity. which may be a floating point no, an integer, a logical value, or a character.
3) Vector instruction can be classified into four primitive types:-

$f_1 : V \rightarrow V$

$f_2 : V \rightarrow S$

$f_3 : V \times V \rightarrow V$

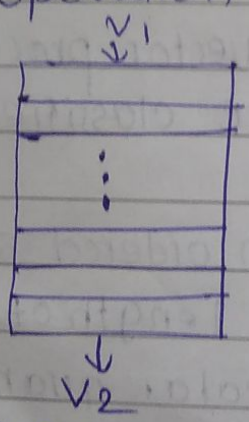
$f_4 : V \times S \rightarrow V$

where V and S denote a vector operand & scalar operand, resp.

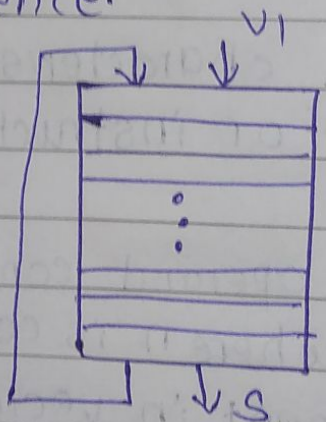
5) The mappings f_1 and f_2 are unary operations and f_3 and f_4 are binary operations.

6) The VSQR (vector square root) is an f_1 operation. VSUM (vector summation) is an f_2 operation. SVP (scalar vector product) is f_4 operation & VADD (vector add) is f_3 operation.

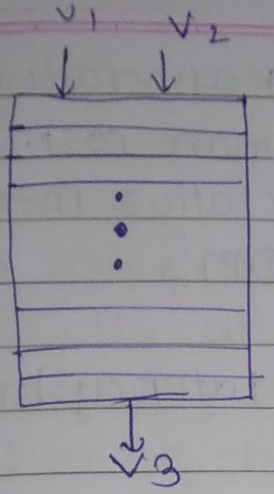
7) The dot product of two vector $V1 \cdot V2 = \sum_i V_{1i} \cdot V_{2i}$ is generated by applying f_3 and then f_2 operation in sequence.



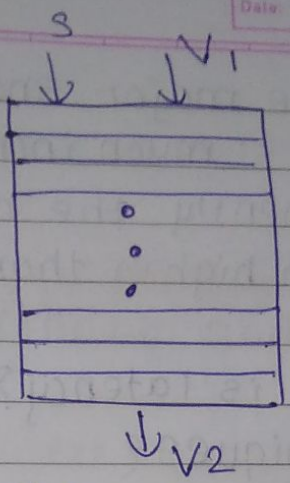
a) $f_1 : V_1 \rightarrow V_2$



b) $f_2 : V_1 \rightarrow S$



c) F3: $V_1 \times V_2 \rightarrow V_3$



d) F4: $S \times V_1 \rightarrow V_2$

ASS-2

Q.1 Explain Associative memory. How it is diffⁿ from Random Access Memory.

1) Data stored in an associative memory are addressed by their contents.
2) In this sense, associative memories have been known as content-addressable memory, parallel Serach memory and multiaccess memory.
3) The major advantage of associative memory over RAM is capability of performing parallel Search and parallel comparison operations.
4) These are frequently needed in many important application, such as the storage and retrival of rapidly changing database, radar signal tracking, image processing.

5) The major shortcoming of associative memory is its much increased hardware cost.
6) Presently the cost of associative memory is much higher than that of RAM's.

Q-2 What is latency? What are latency hiding techniques?

1) The value of latency includes: network delays, cache miss penalty, delays caused by contentions in split transactions.
2) Interval between switches refers to the cycles between switches triggered by remote reference.
3) Latency hiding can be achieved using four methods-
   1) Prefetching technique
   2) using coherent caches
   3) using relaxed memory consistency models
   4) using multiple context support.

1) using prefetching technique-
   which bring instruction or data close to the the processor before they are actually needed.

2) using coherent caches-
Supported by hardware to reduce cache misses.

3) using relaxed memory consistency models-
By allowing buffering and pipelining of memory references.

4) using multiple context support -
To allow a processor to switch from one context to another.

Q-3 Draw and explain cm* architecture in loosely coupled system.

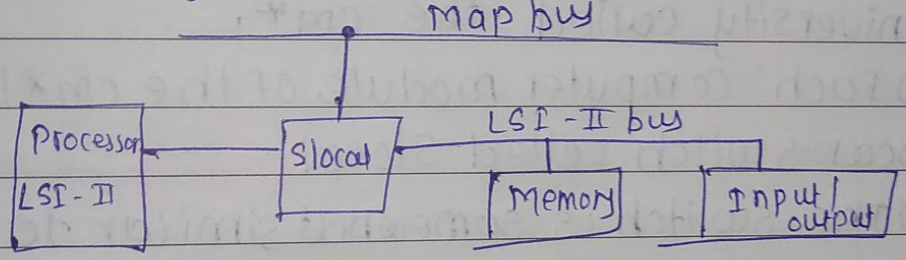


Fig. A] - A computer module.

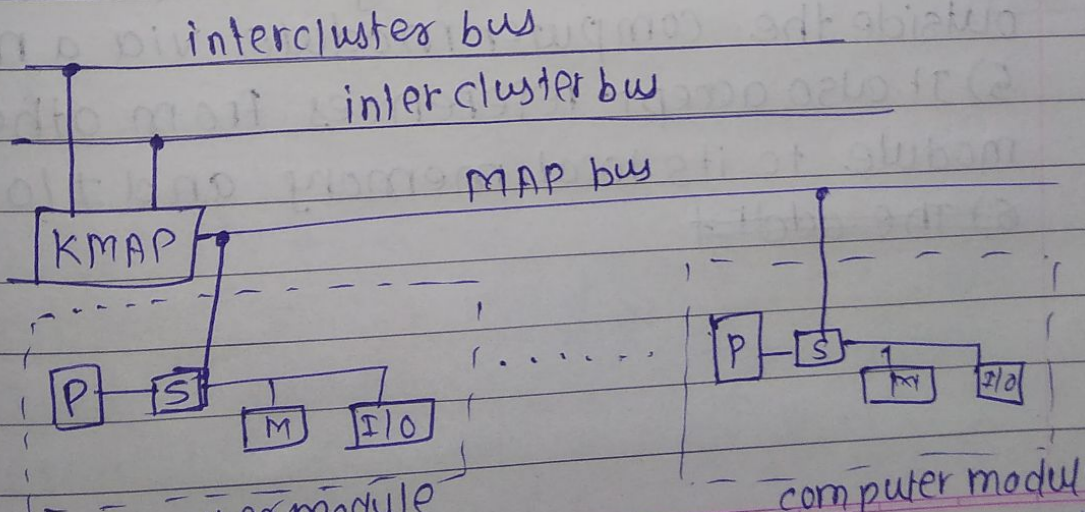
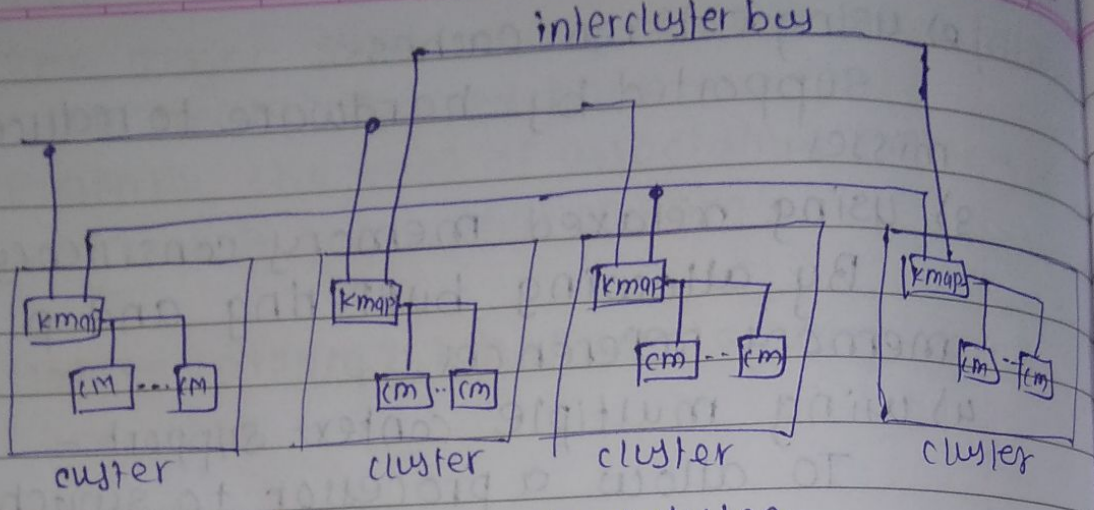


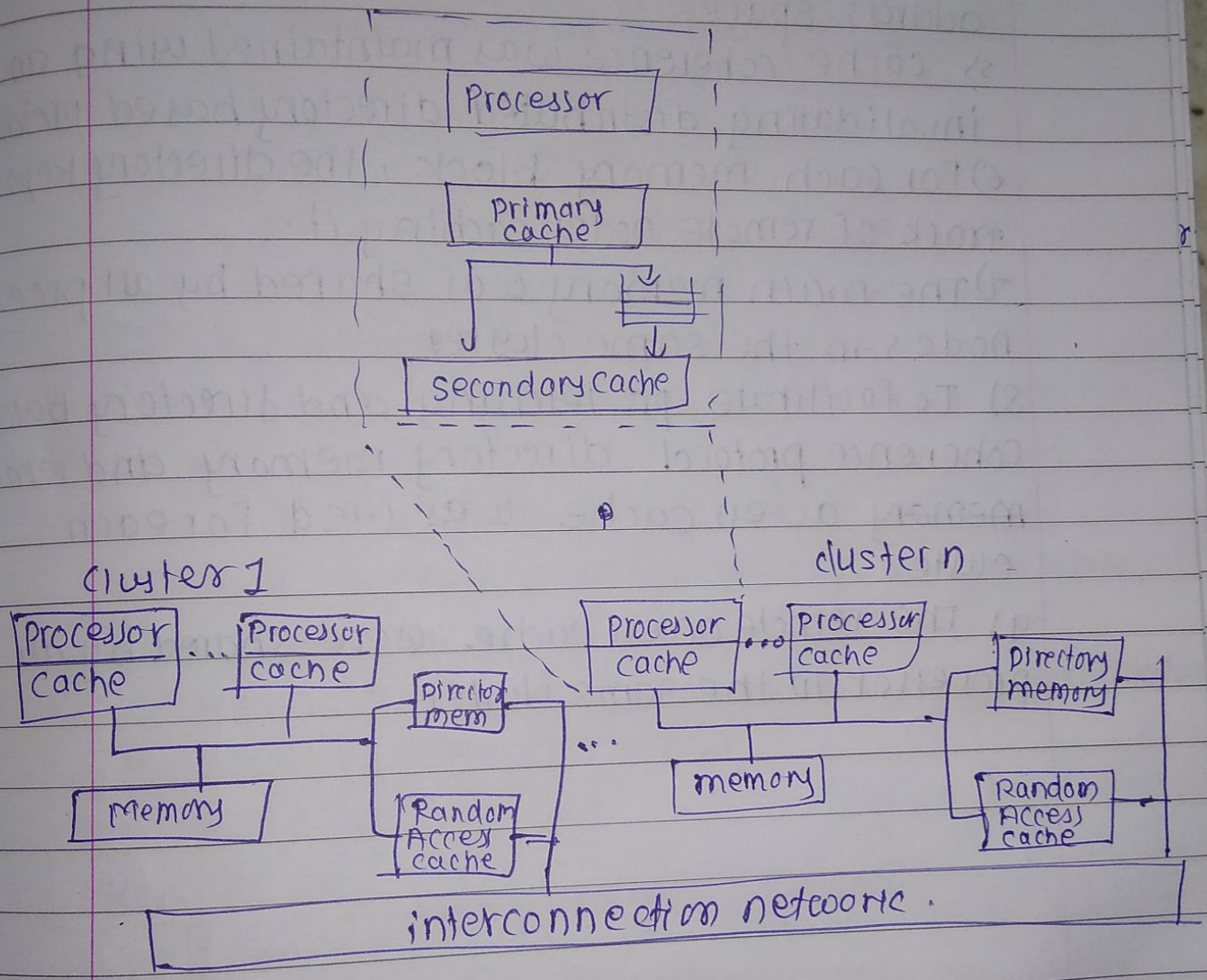
Fig B] A cluster of computer module



c] - A network of cluster.

1) For a hierachical LCS, we take example of a computer system project at Carnegie Mellon university called the CM*.
2) Each computer module of the CM* includes a local switch called Slocal.
3) The switch is somewhat similar to the CAS
4) The slocal intercepts and routes the processor's requests to the memory and I/O devices outside the computer module via a map bus.
5) It also accept references from other compute module to its local memory and I/O devices.
~~6) The addr~~

Q-4 Draw and explain scalable coherent cache multiprocessor with distributed shared memory model.



1) The DASH architecture was a large-scale, cache-coherent, NUMA multi-processor system.
2) It consist of multiple multiprocessor clusters connected through a scalable, low-latency interconnection network.

3) Physical memory was distributed among the processing nodes in various cluster.
4) The distributed memory formed a global address space.
5) cache coherence was maintained using an invalidating, distributed directory based protocol
6) For each memory block, the directory kept track of remote nodes caching it.
7) The main memory was shared by all processing nodes in the same cluster.
8) To facilitate pre fetching and directory based coherence protocol directory memory and remote ~~memory~~ access cache was used for each cluster.
9) The remote access cache were shared by all processor in the same cluster.

Q-1

ASS - 3

Q-1 What are diffⁿ component of Kmap in cm* architecture? State funⁿ of each component.

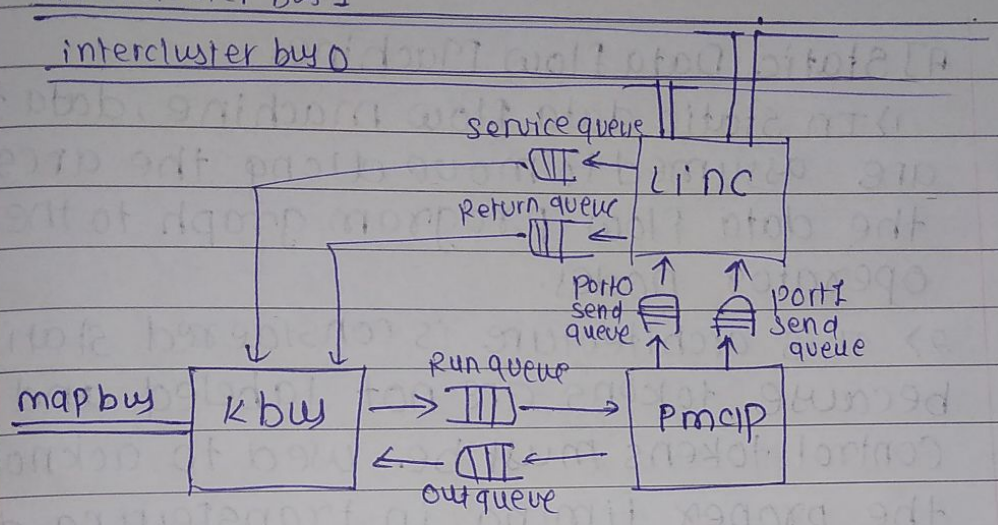


Fig. The component of Kmap in CM*

1) Three processors in the Kmap are the Kbus, Linc and the Pmap.
2) Kbus is the bus controller which arbitrates requests to the map bus.
3) The Linc message communication betn the Kmap and other Kmap.
4) The Pmap is the mapping processor which responds to requests from the Kbus and Linc. It also performs most of the request processing.
5) The Pmap communicates with the compute modules in the cluster via map bus which is a packet switched bus.

Q-2 With block diagram explain dynamic and static data flow architecture.

A] Static Data Flow Machines -

1) In static data flow machine, data tokens are assumed to move along the arcs of the data flow program graph to the operator nodes.

2) This architecture is considered static because tokens are not labeled and control tokens must be used to acknowledge the proper timing in transferring data tokens from node to node.

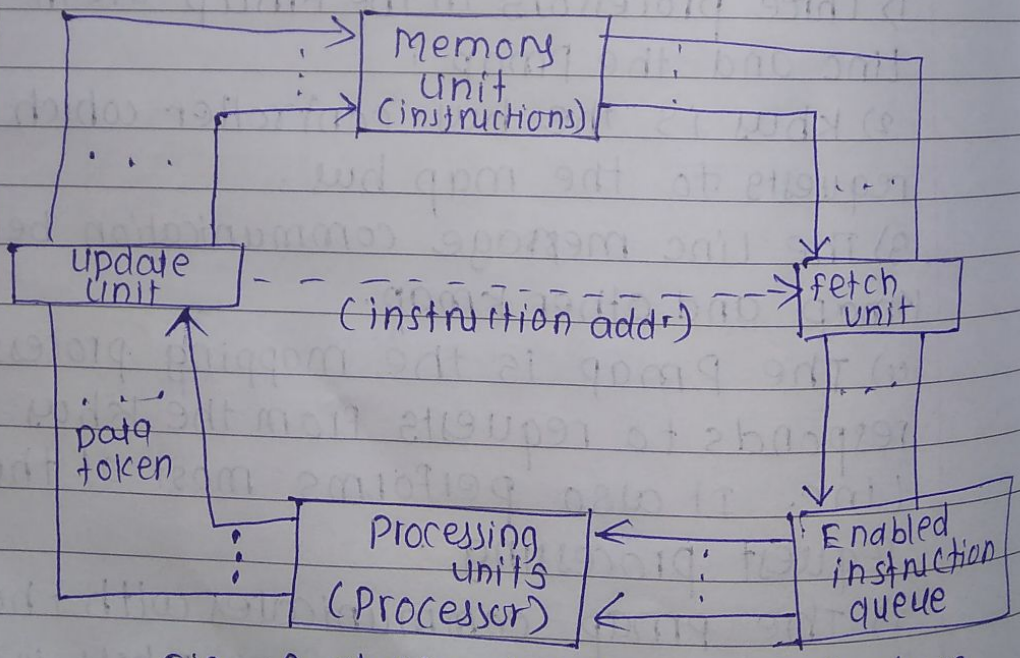


fig. A static dataflow architecture

B] Dynamic Dataflow machine-

1) A dynamic data flow machine uses tagged tokens, so that more than one token can exist in arc.

2) The tagging is achieved by attaching a label with each token which uniquely identifies the context of the at particular tokens.

3) The dynamic data flow allows greater exploitation of parallelism, however, this advantage comes at the expense of the overhead in terms of the generation of tags, larger data tokens, and complexity of the matching tokens.

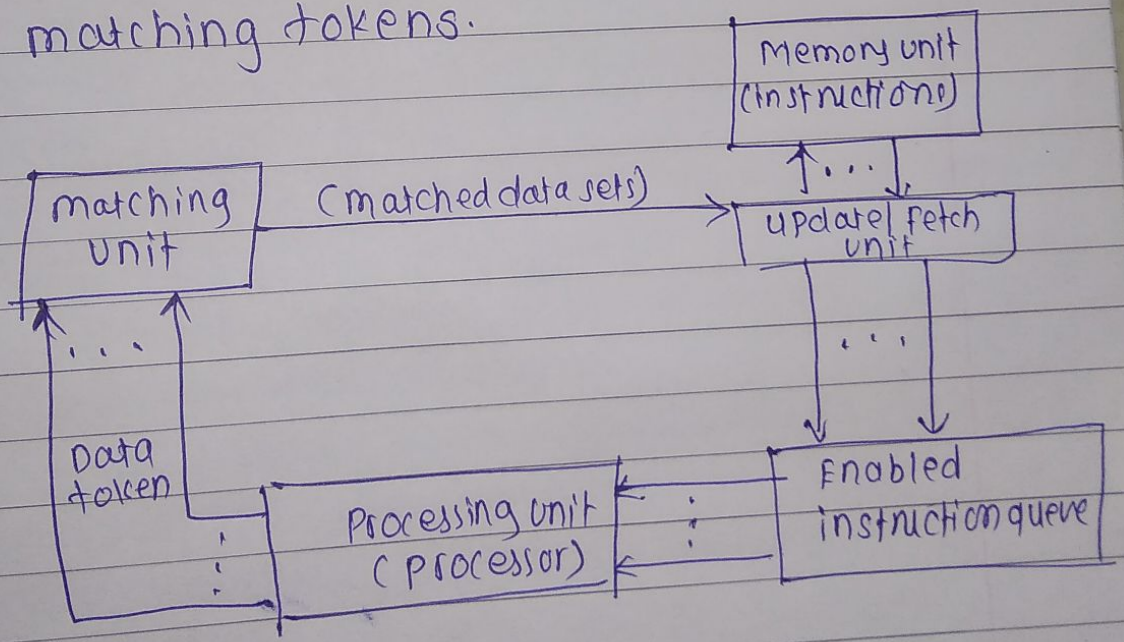


fig. Dynamic dataflow architecture.