HPCA

Assignment 2.

① Why associative memories are called a contents addressable memories?

i) A memory unit accessed by content is called as Associative memory or Content Addressable Memory (CAM).

ii) Associative memory is accessed simultaneously & in parallel on the basis of data content rather by special address of or location.

iii) This memory is capable of finding an empty unused location to store the words.

iv) When a word is to be read from associative memory, the content of word is specified or part of the word is specified.

v) When word is written in associative memory, no address is given.

vi) The memory locates all words which match the specified content & marks them for reading.

vii) Because of its organization, the Associative memory is uniquely suited to do parallel searches by data association

It is widely used in database management system.

ix) Associative memory is expensive than RAM.

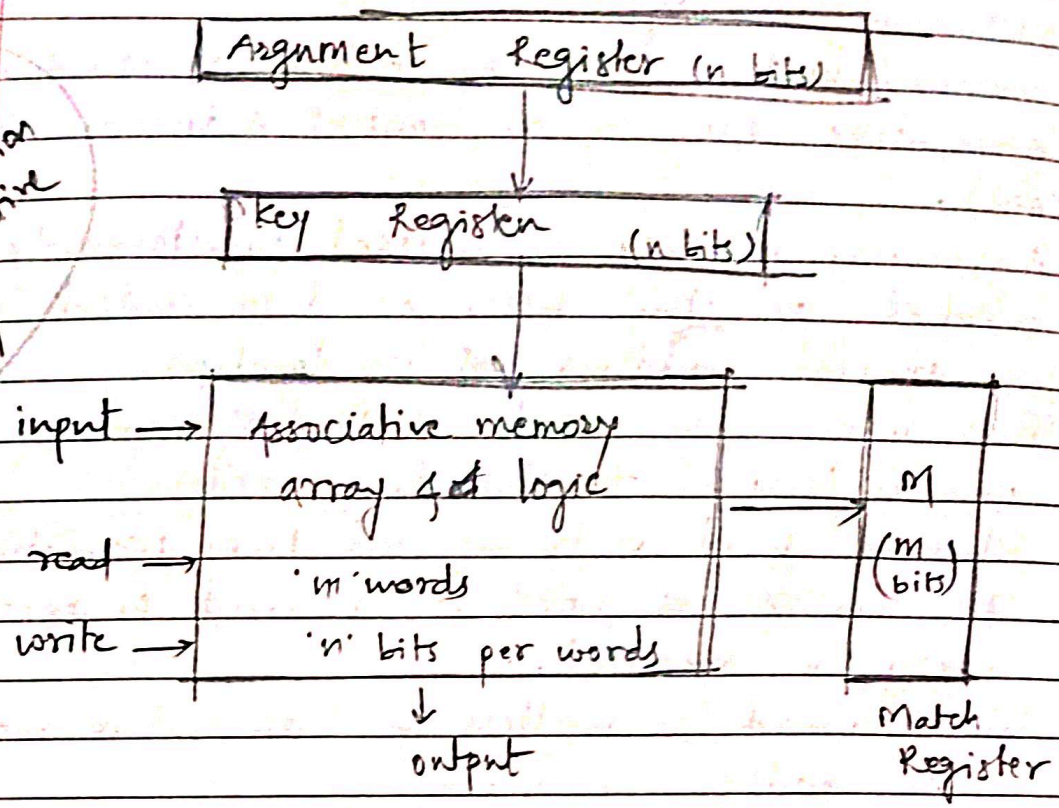
(Logic circuit)

(Neural Network)

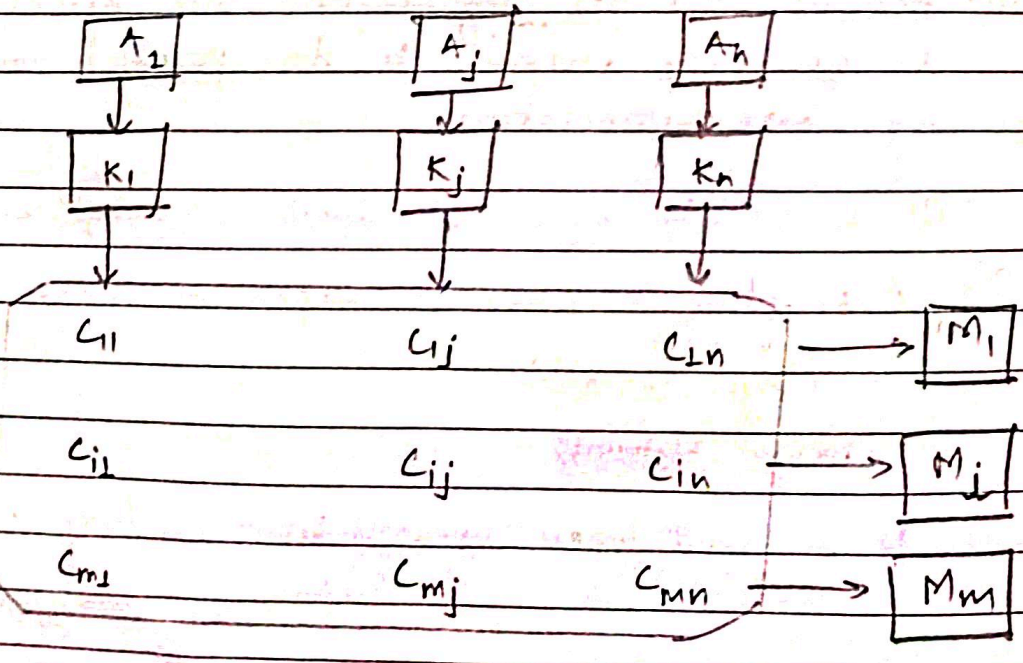
x) It is used where search time is very critical.

Q.2. Draw the architecture of associative memory array, what is Bij? How associative memory works?

Hardware organization of Associative memory



Associative memory of 'm' words, n cells per words:



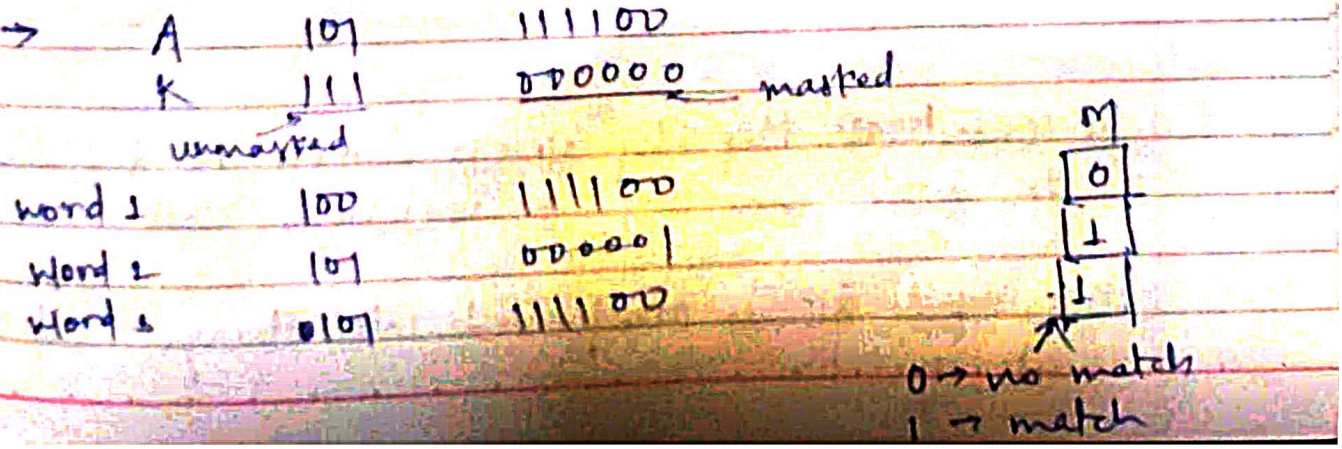
Argument Register: It contains words to be searched
(A) It has 'n' bits (one for each bit of word)

key register (K): It provides mask for choosing a particular key in argument word.
- It specifies which part of the argument word needs to be compared with words in memory.
- If all bits in key register are 1's, the entire word should be compared otherwise, only the bits having 1's in their corresponding position are compared.

Associative Memory Array: - It contains the word that are to be compared with the argument word in parallel.
- It consists of 'm' words with 'n' bits per word.

Match logic (M): It has 'm' bits, one bit corresponds to each word in the memory array.
After the matching process, the bits corresponding to matching words in match register are set to 1.

Reading is accomplished by sequential access in memory for those words whose match bits are set (or 1).



2nd diagram

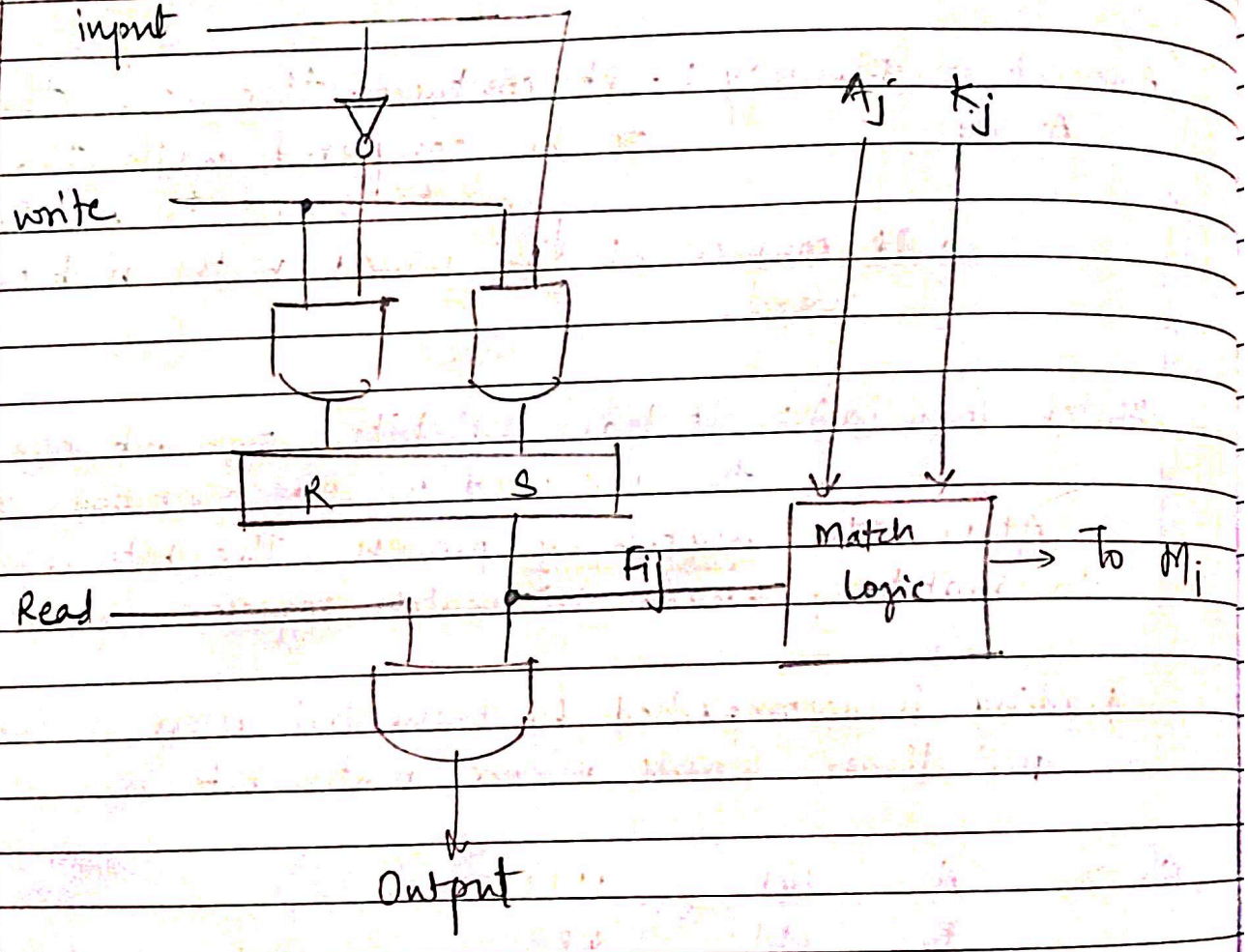
Associative memory of 'm' words, 'n' cells per words. The cells in array is represented by Cij : a cell for bit 'j' in word 'i'

where,

i = word number

j = bit position in word

Internal organisation of cell Cij (Bij might be)



It consist of flip-flop storage element Fij & circuits for sending, writing & matching cell.

The input bit is transferred into storage cell during write operation.

The bit stored is read out during read operation

The Match logic compares the content of the storage cell with the corresponding unmasked bit of the argument & sets the bit in M_i to 1.

Q.3) Draw and explain scalable coherent multiprocessor model with distributed shared memory, why it is called DASH architecture?

Ans → Parallel processing in memory / Parallel computer Architecture
Primary Memory architectures of parallel computer are:

1) Shared Memory Architecture
   - Shared common memory
   - Used in multiprocessors
2) Distributed Memory Architecture
   - Unshared distributed memory
   - Used in Multicomputers
3) Hybrid Memory Architecture
   Combination of both shared & distributed

Scalable Coherent multiprocessor model with DSM:

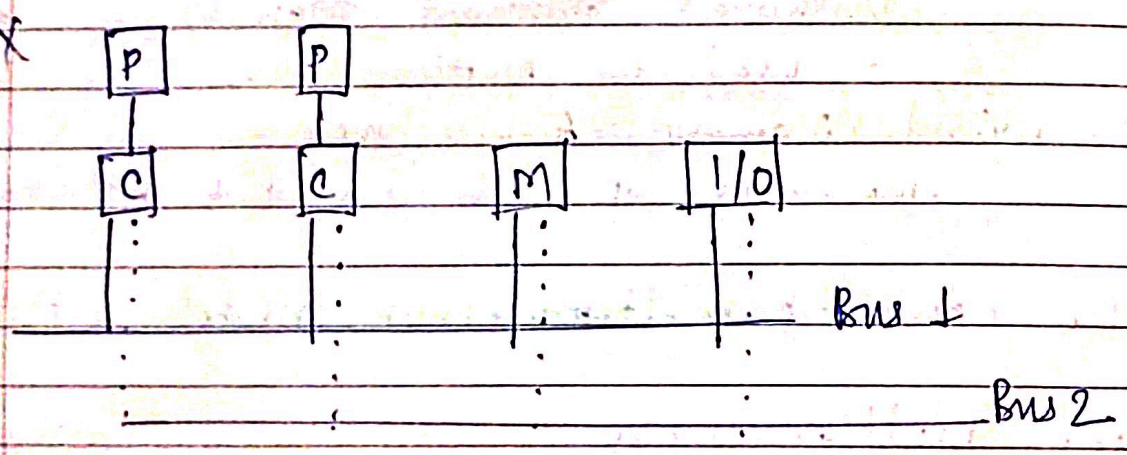
Scalable Multiprocessors –
requirement →
- Must scale the local memory bandwidth linearly
- Must scale the global interprocessor communication bandwidth.

- Scaling memory bandwidth cost effectively requires seperate, distributed memories.

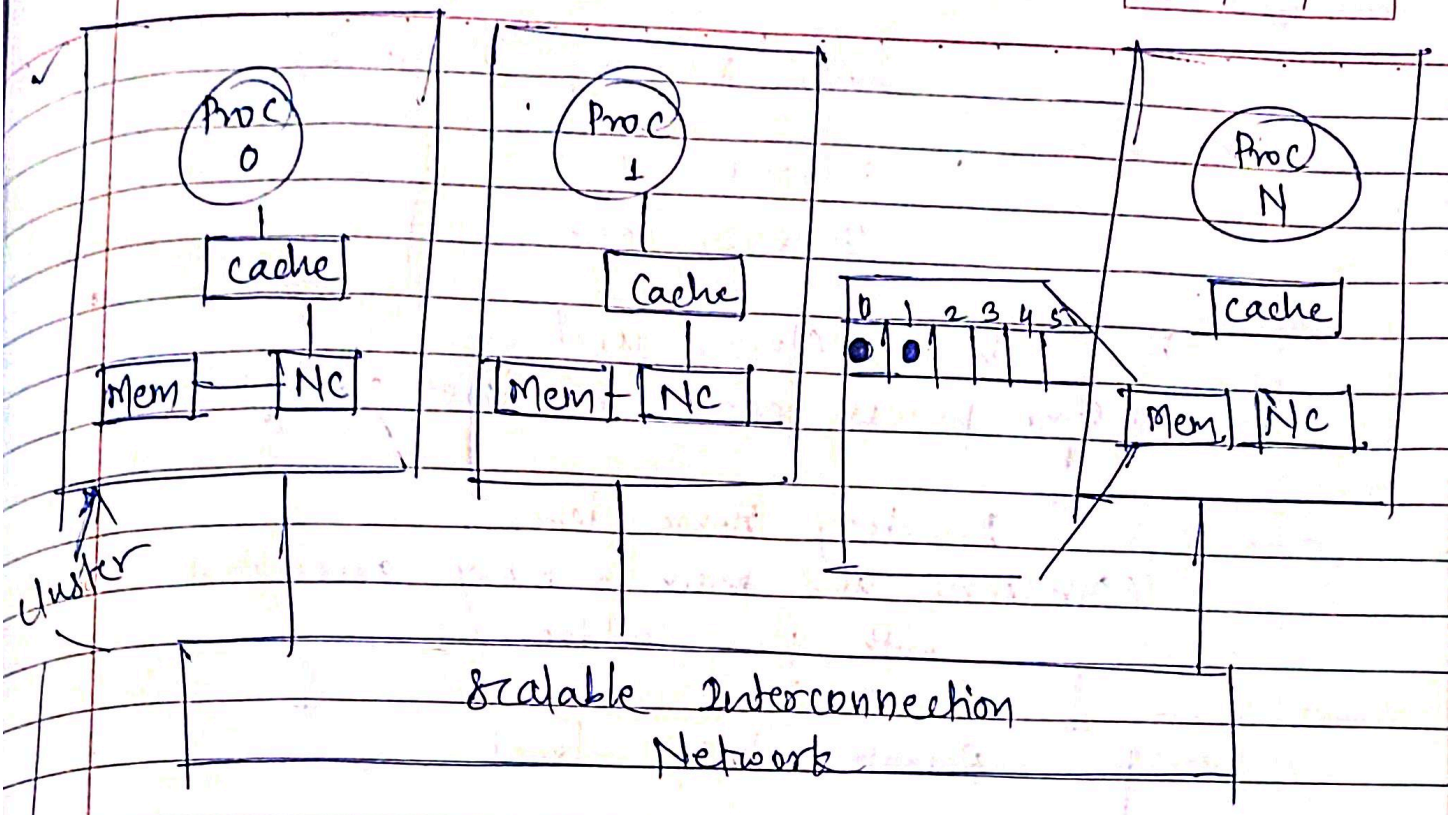
Architectural ~~Alternatives~~
1) ~~Distributed~~, ~~private memories~~
1) Single Address space implemented with physically distributed memories.

It requires →
1) Attention to locality of Access
2) Scalable interconnection technology

Why single Address Space -
(i) Easier to program / build compiler
(ii) → Allows incremental program development method.
(iii) Can efficiently simulate other models (with hardware help)
(iv) performance advantages
communication
Easier to exploit caching
(v) Reduce latency



P → processor
C → cache
M → Memory
I/O → input / output

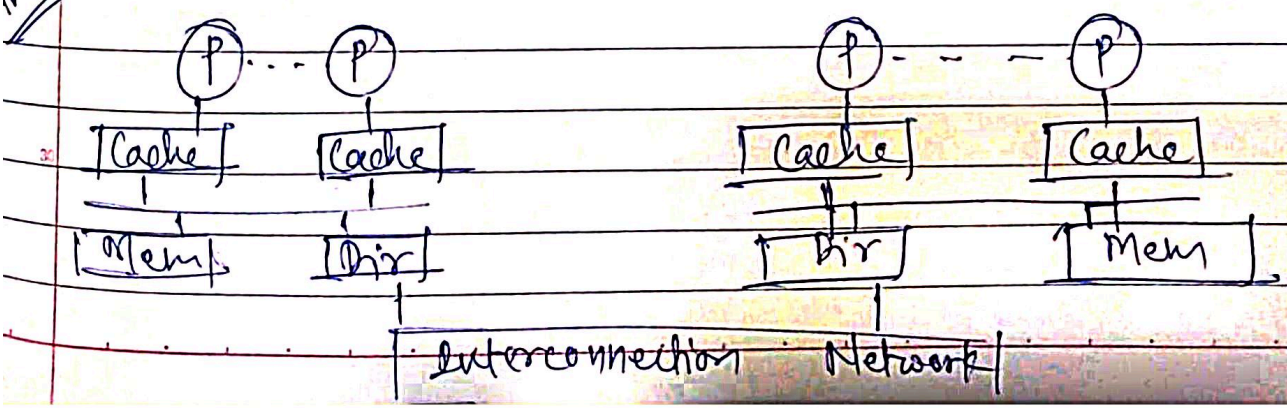


DASH Architecture

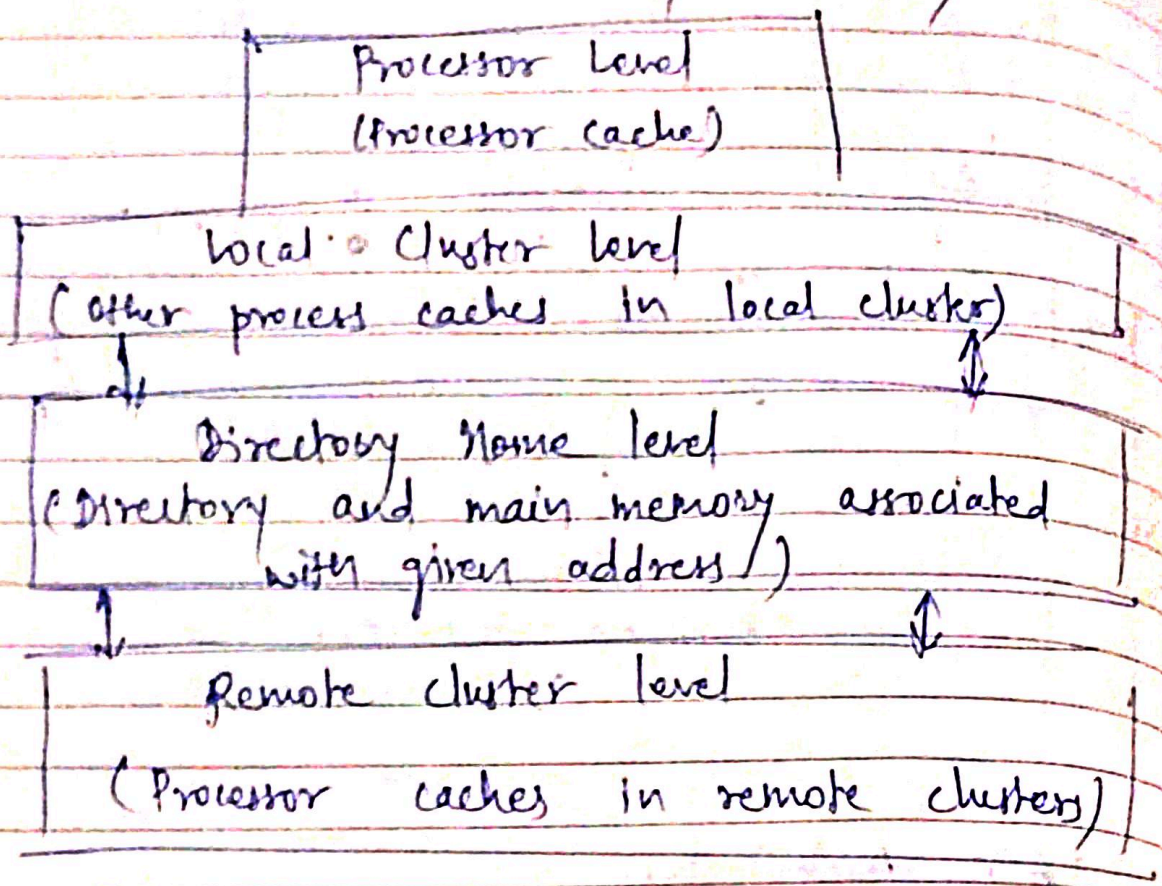
Directory Architecture for Shared Memory

DASH is build around the concept on distributed directory

- Nodes connected by high BW network
- Distributed memory with single address space



DASH Memory Hierarchy



Dash combines -
Scalability advantages of message-passing
programming advantages of single address space

Q.4. Explain the concept of Multithreading in parallel architectures.

Multithreading demands that the processor be designed to handle multiple contexts simultaneously on a context switching basis.

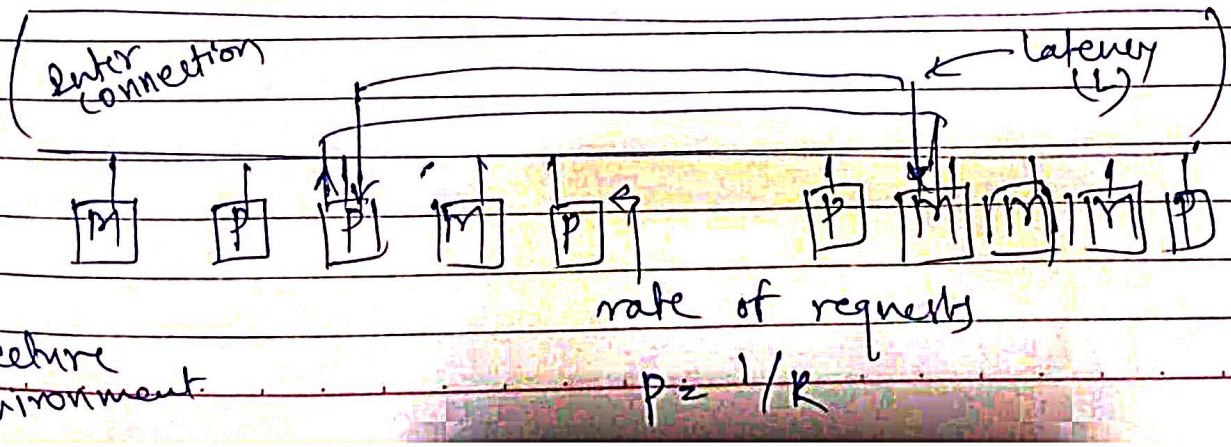
Architectural Environment:
It consist of network of processors and memory nodes.

Latency (L) : This is communication latency on a remote memory access. 'L' includes network delays, cache miss penalty and delays.

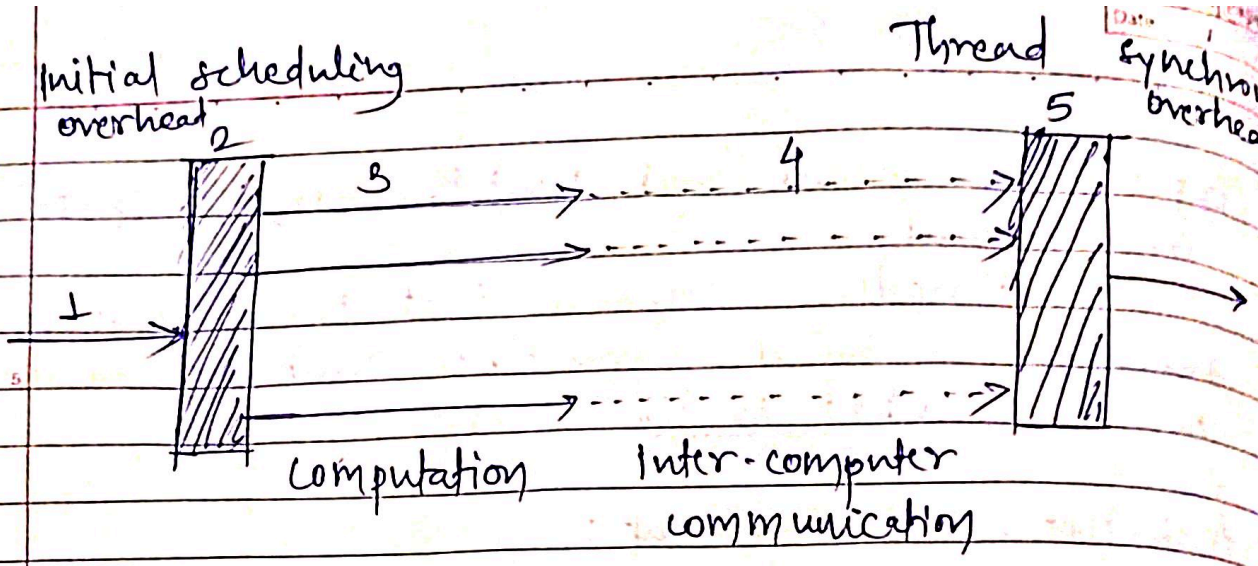
The number of threads (N). This includes no. of threads that can be interleaved in each processor.

The context switching overhead (C) : This refers to cycles lost in performing context switching in processor.

The interval between switches (R) : This refers to cycles between switches triggered by remote reference.



Architecture environment



Multithreaded computation model

Context switching policies

1. switch on cache miss
2. switch on every load
3. switch on every instruction
4. switch on block of instruction.

Assignment No. 4

1. What are dataflow architectures?
Data Driven Mechanism (used by data flow computers) (Data Driven Computers)

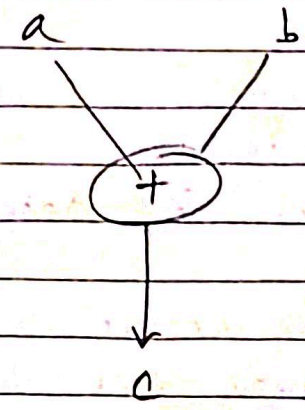
- In data flow computers, the execution of an instruction is driven by data availability instead of being guided by program counter.

- These machines are eager for data to be present If the data is present then the instruction is executed these computer are also known as data driven computers or Eager machines.

- In a data driven program, the instruction are not ordered.
There is no shared memory. Thus data is stored inside instructions.

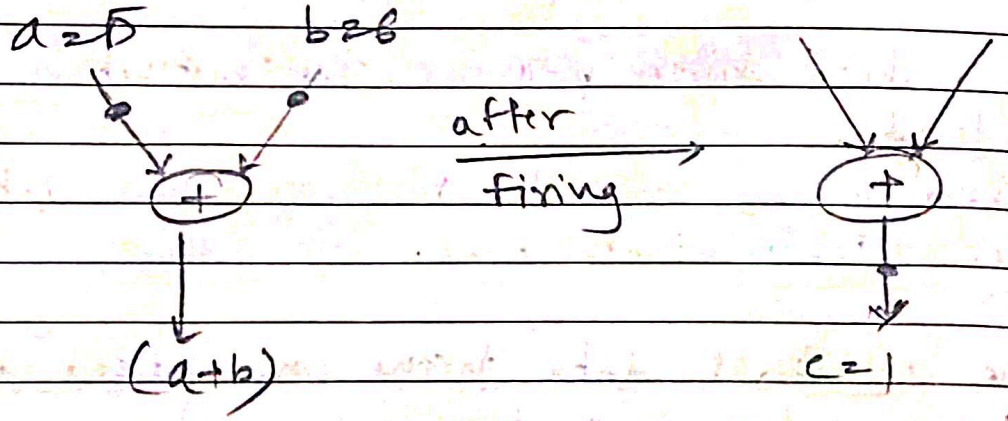
- The result of data tokens are passed directly between the instruction.
Actually a copy of data is passed to instructions Once data is consumed by instruction it will no longer available for reuse by other instructions.

- In a data flow machine, a program consists of data flow nodes i.e. a program is represented using directed acyclic graph (nodes & edges).
- Instructions is represented by a node & the data dependancy relationship is represented by the edge between connected nodes.



A data flow node fires (fetched & executed) when all its inputs are ready i.e. when all inputs have tokens.

Eg. if a=5 b=6



Data Flow features -
- No need for shared memory
- No program counter
- No control sequencer

Special Mechanisms are required to
- detect data availability
- match data tokens
- enable chain reaction of asynchronous instruction execution.

Camlin Page
Date / /

Advantage -
High potential for parallelism and throughput
Freedom from side effects.

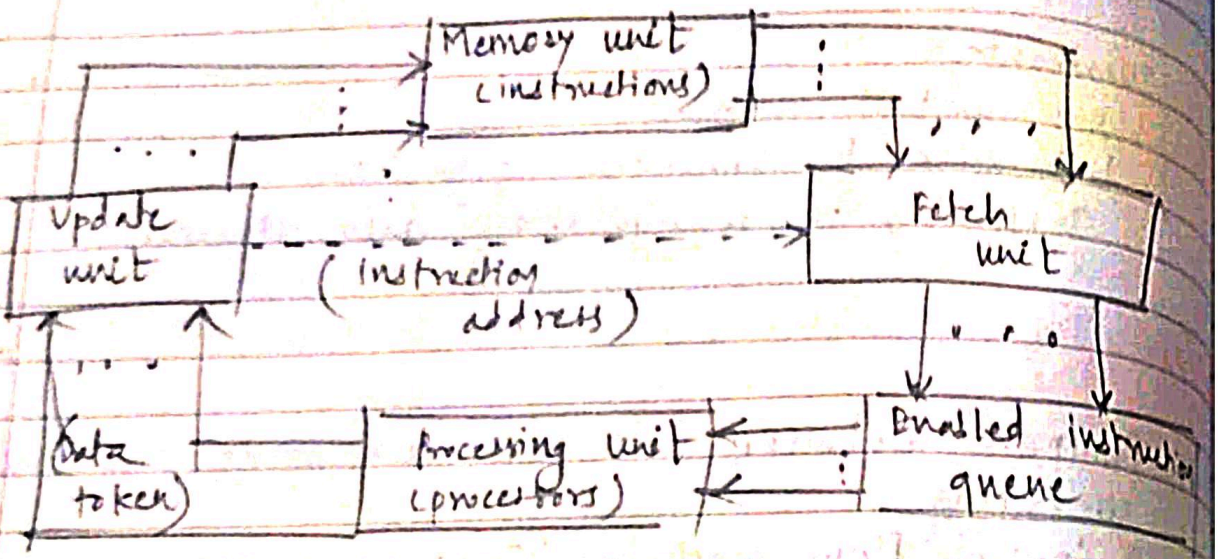
Disadvantage -
High control overhead
lose time for unneeded arguments.
Difficulty in manipulating data structures.

Q.2. Compare between data flow computing and traditional computing.

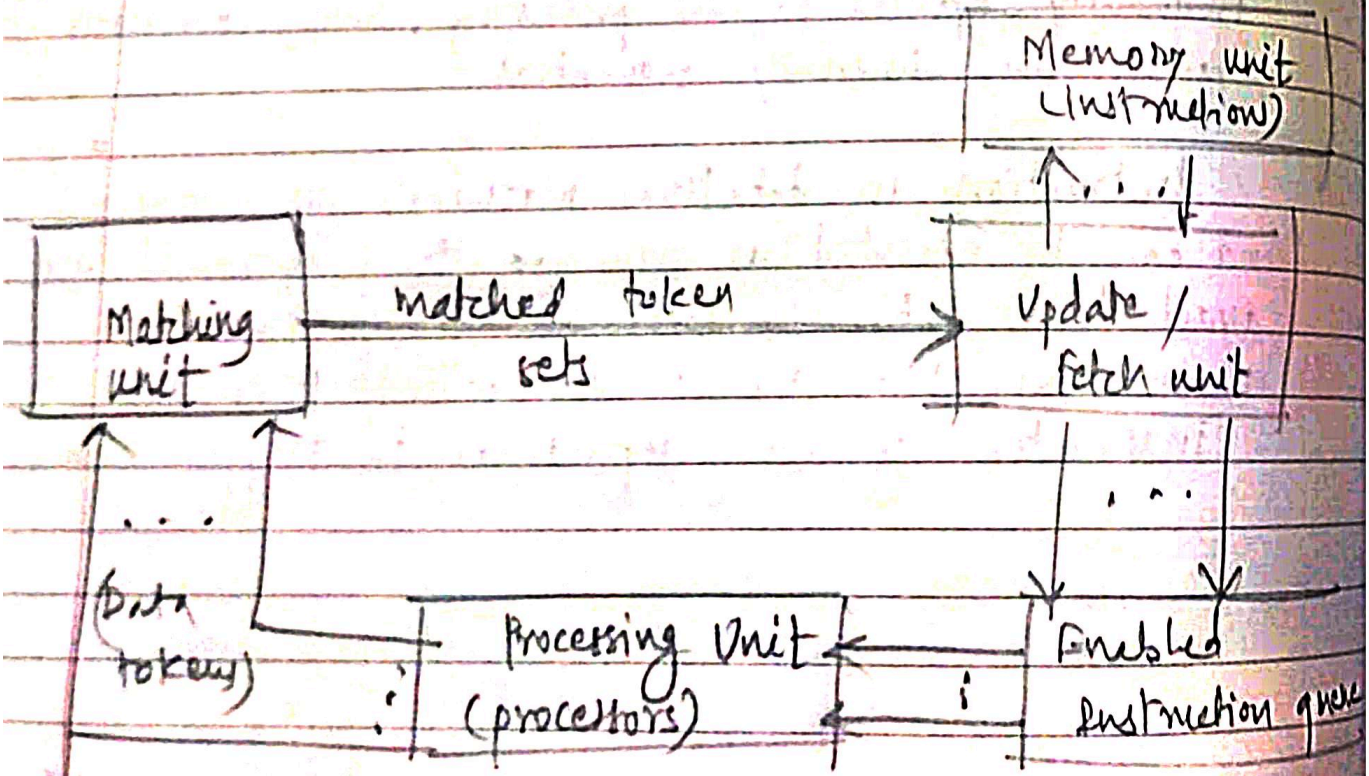
Control flow machines / Traditional computing used shared memory for instructions and data.
Since variables are updated by many instructions, there may be side effects on other instructions. These side effects frequently prevent parallel processing. Single processor systems are inherently sequential.

Instruction in dataflow machines are unordered and can be executed as soon as their operands are available; data is held in the instructions themselves. Data tokens are passed from an instruction to its dependents to trigger execution.

Q.3. Draw and explain static & dynamic dataflow architectures.



(A static data flow computer organisation)



A dynamic computer organization.

Static Data flow computers :

i) Memory section consist of instruction cells which hold instructions & their operands.
ii) Processing section consist of processing unit that perform functional operations of data tokens.
iii) Arbitration network delivers operation packets from the memory section to the processing section.
iv) Control network delivers a control token from the processing section to the memory section.
v) Distribution network delivers data tokens from the processing section to the memory section.

Dynamic Data flow computers :

Q.4. What are dataflow operators? State data flow language properties.

Operators can be - (Each operator denoted by sym
- arithematic → (i) Binary (ii) Unary
- logical →
- relational
- equality
- bitwise
- reduction
- shift
- ~~concatenation~~
- ~~conditional~~

Arithematic -

(i) Binary - Multiply (*)
(0, 1) divide (/) → Truncate fractional part
add (+)
subtract (-)
power (**)
modulus (%)

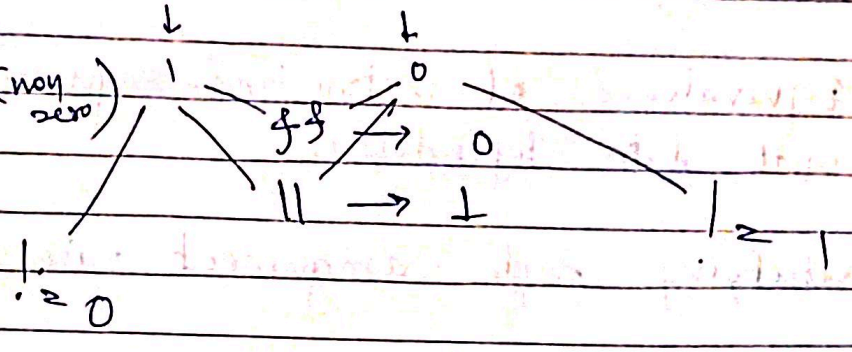
(ii) Unary - (+) & (-)
They are used to specify the positive or negative sign of the operand.

Unary + or - operators have higher precedence than binary + or - operators

-4 // negative 4

2) logical - (Use of parantheris)
logical and (&&) } → binary
logical or (||) }
logical not (!) → unary - single number

Eg: A = 3 B = 0



3) Relational - >, <, >=, <=

4) Equality :
logical equality (==)
logical inequality (!=)
case equality (===)
case inequality (!==)

5) Bitwise operators
negation ~
and &
or |
xor ^
xnor ^~ , ~^

6) Reduction operators
and &
nand ~&
or |
nor ~|
xor ^
xnor ~^ , ^~

7) Shift operators
right shift >>
left shift <<
arithematic right shift >>>
arithematic left shift <<<

Properties of in data flow language

i) Freedom from side effects based on functional programming.

ii) Locality of effect without far-reaching data dependencies.

iii) Equivalence of instruction-sequencing constraints with data dependencies.

iv) Satisfying single-assignment rule with aliasing

v) Unfolding of iterative computations into parallelism

vi) Lack of "history sensitivity" in procedure calls.