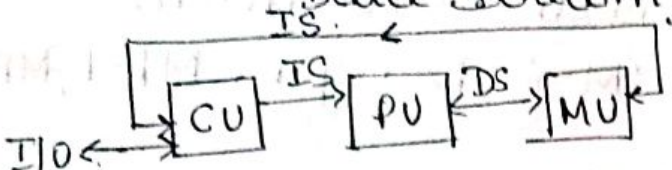
ACA

Assignment-1 Vaibhavi.M
1JT17CS056

Module-1.

1. Describe the Flynn's classifications of computer architectures.

Ans: 1* SISD (Single Instruction Stream over a Single Data Stream).

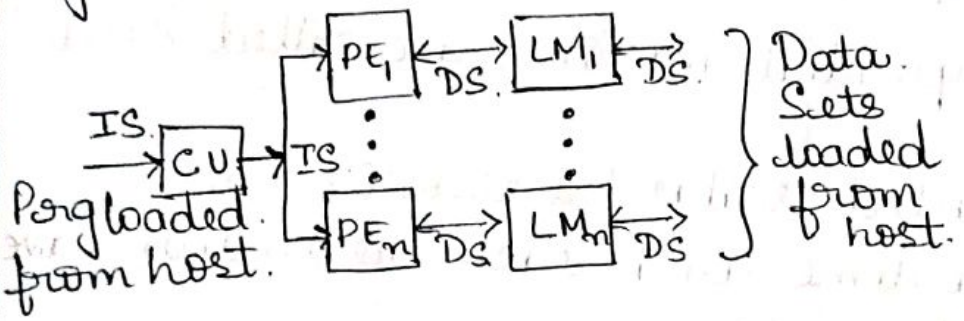


* Conventional sequential machines are called SISD computers.
* They are also called scalar processor i.e, one instruction at a time and each instruction have only one set of operands.
* Single Instruction :- Only one instruction stream is being acted on by the CPU during any one clock cycle.
* Single Data :- Only one data stream is being used as i/p during any one clock cycle.
* Deterministic execution.
* Egs :- most PC's, single CPU workstations & mainframes

2* SIMD (Single Instruction stream over Multiple Data streams)

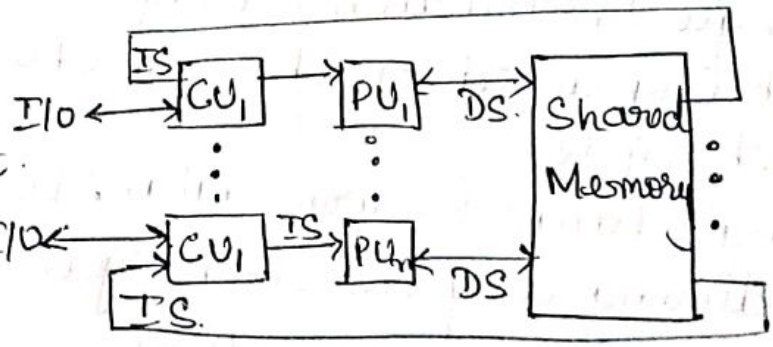
* Single Instruction :- All processing units execute the same instruction issued by the control unit at any given clock cycle.

* Multiple data:- Each processing unit can operate on a different data element. The processors are connected to shared memory or interconnection N/w providing multiple data to processing unit.
* Thus single instruction is executed by different processing unit on different set of data.
* Synchronous & deterministic execution.
* Eg:- Connection Machine CM-2, Maspar MP-1, MP-2.



3* MIMD (Multiple Instruction streams Over Multiple Data).

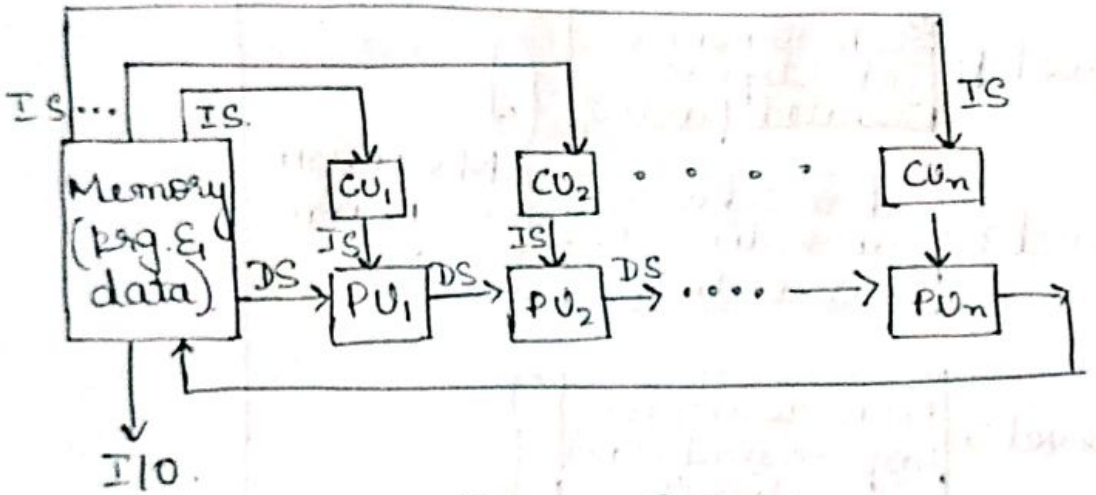
CU - Control Unit
PU - Processing Unit
MU - Memory Unit
IS - Instruction Stream
DS - Data Stream
PE - Processing Element
LM - Local Memory



* A Single data stream is fed into multiple processing units.
* Each processing unit operates on the data independently via independent instruction streams

* A single data stream is forwarded to different processing unit which are connected to different control unit & execute instruction given to it by control unit to which it is attached.
* This architecture is also known as Systolic Arrays for pipelined execution of specific instructions.

* MISD (Multiple Instruction streams & a Single Data Stream).

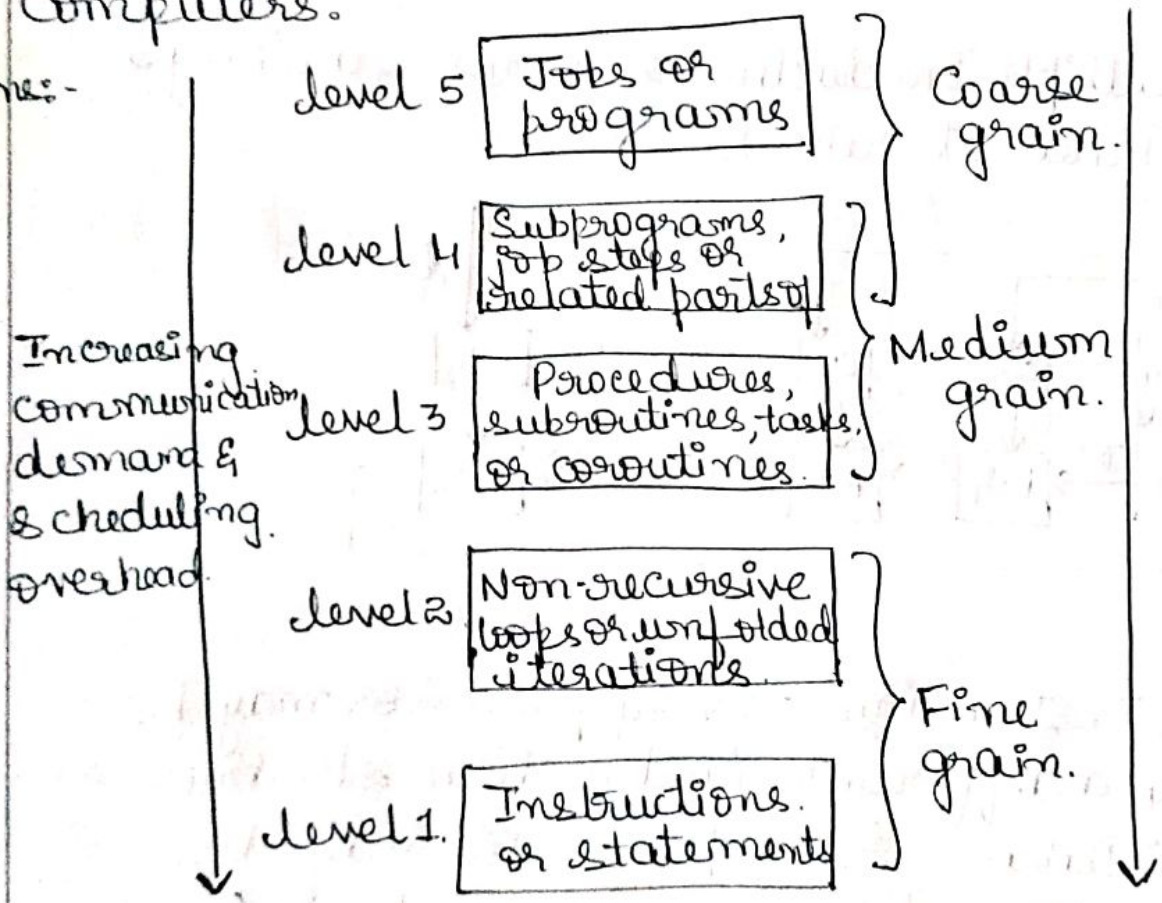


* Multiple Instructions :- Every processor may be executing a different instruction stream
* Multiple Data :- Every processor may be working with a different data stream, multiple data stream is provided by shared memory.
* Can be organised loosely coupled or tightly coupled depending on sharing of data and control.
* Execution can be synchronous or asynchronous deterministic or non-deterministic.

* There are multiple processors each processing different tasks.
* Eg:- most current supercomputers etc.

2. Write the neat diagram explain the levels of parallelism in program execution on modern computers.

Ans:-



(i) Instruction Level Parallelism :-

* This fine-grained, or smallest granularity level typically involves less than 20 instructions per grain.
* The no of candidates for parallel execution varies from 2 to thousands, with about 5 instructions or statements being the average level of parallelism.

(i) Advantages :- There are usually many candidates for parallel execution. Compilers can usually do a reasonable job of finding this parallelism.

(ii) Loop-level Parallelism :-

* Typical loop has less than 500 instructions. If a loop has less than 500 instructions. If a loop operation is independent b/w iterations, it can be handled by a pipeline, or by a SIMD machine.
* Most optimized program construct to execute on a parallel or vector machine.
* Some loops are difficult to handle. Loop-level parallelism is still considered fine grain computation.

(iii) Procedure-level Parallelism :-

* Medium-sized grain, usually less than 2000 instr.
* Detection of parallelism is more difficult than with smaller grains, inter procedural dependence analysis is difficult & history-sensitive.

(iv) Subprogram-level Parallelism :-

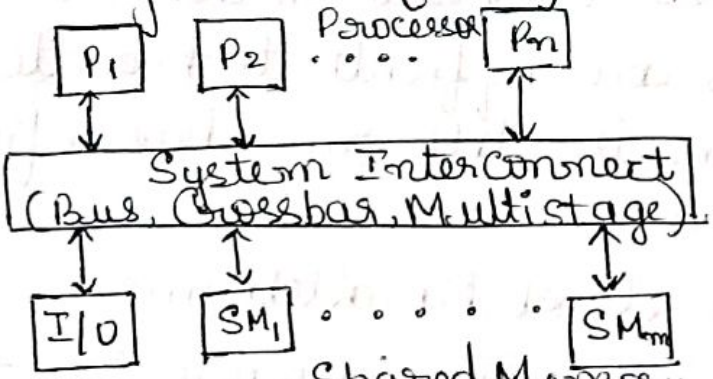
* Job step level, grain typically has 1000's of instructions; medium- or coarse-grain level.
* Job steps can overlaps across different jobs. Multiprogramming conducted at this level.

(v) Job or Program-Level Parallelism :-

* Corresponds to execution of essentially independent jobs or programs on a parallel computer.
* This is practical for a machine with a small no of powerful processors, but impractical for a machine with a large no of simple processors.

3. Describe the UMA, NUMA, COMA & CC-NUMA models for multiprocessor systems.

(i) Uniform Memory-Access (UMA) model :-



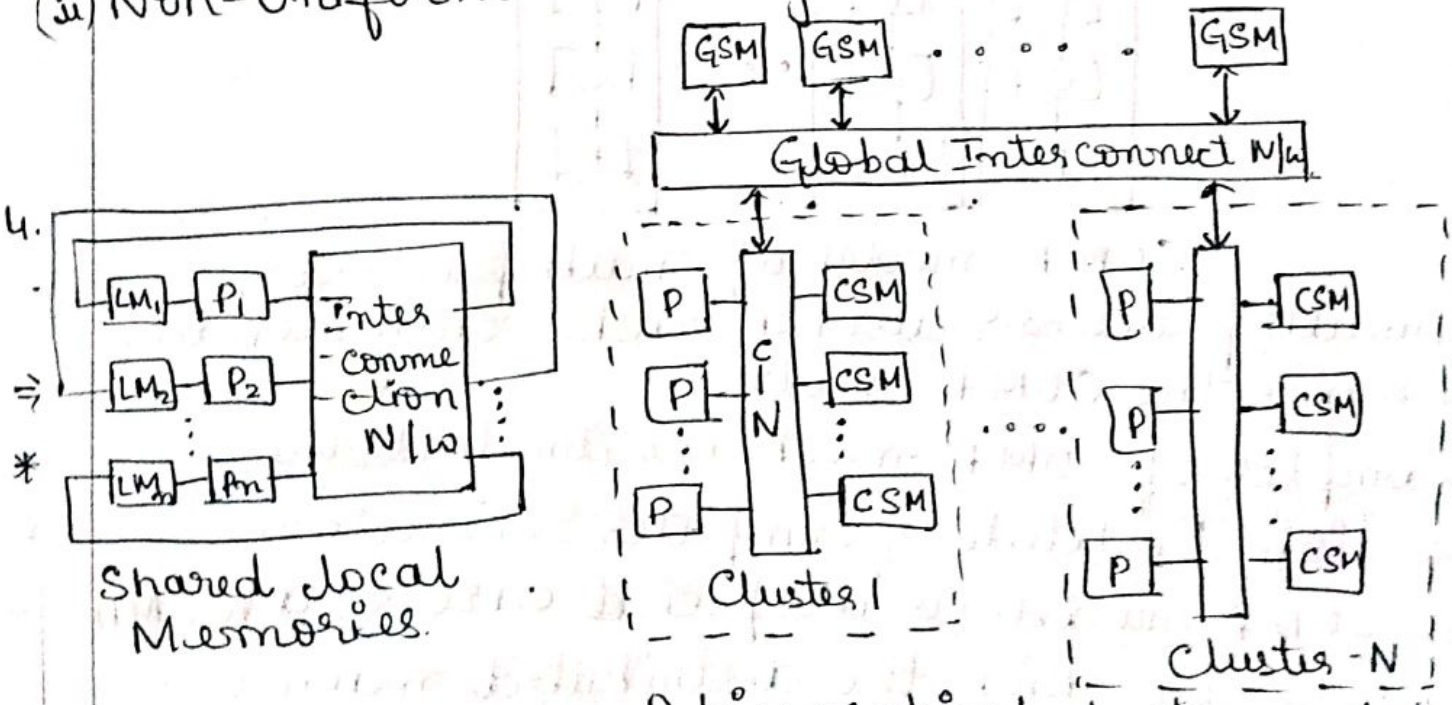
* In a UMA multiprocessor model, the physical memory is uniformly shared by all the processors.
* All processors have equal access time to all memory words, which is why it is called uniform memory access.
* Each processor may use a private cache. Peripherals are also shared in some fashion.
* Multiprocessors are called tightly coupled systems due to the high degree of resource sharing.

* Most computer manufactures have multiprocessor extensions of their uniprocessor product line.

* The UMA model is suitable for general-purpose & timesharing applications by multiple users. It can be used to speed up the execution of a single large program in Time Critical Application.

* When all processors have equal access to all peripheral devices, the system is called a Symmetric multiprocessor.

(ii) Non-Uniform-Memory Access (NUMA) Model:



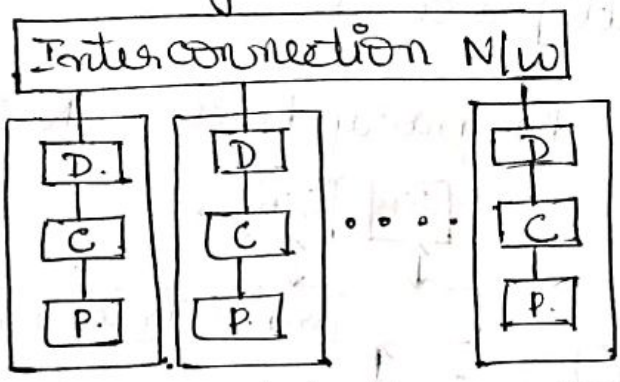
A hierarchical cluster model.

* A NUMA multiprocessor is a shared-memory system in which the access time varies with the location of the memory word.

* Two NUMA machine models are depicted in the fig above.

* The shared-memory is physically distributed to all processors, called local memories.
* The collection of all local memories forms a global address space accessible by all processors.
* It is faster to access a local memory with a local processor. The access of remote memory attached to other processor takes longer due to the added delay through the interconnection N/w.

(iii) Cache-Only Memory Architecture (COMA) model.



COMA model of multiprocessor.

* A multiprocessor using cache-only memory assumes the COMA model.
* Examples of COMA machines include the Swedish Institute of Computer Science's.
* The COMA model is a special case of a NUMA machine, in which the distributed main memories are converted to caches.
* There is no memory hierarchy at each processor node. All the caches form a global address space.
* Remote cache access is assisted by the distributed cache directories.

(iv) CC-NUMA Model

Besides the UMA, NUMA & COMA models specified above, other variations exist for multiprocessor

Eg:- a cache-coherent non-uniform memory access (CC-NUMA) model can be specified with distributed shared memory & cache directories.

Early examples of the CC-NUMA model include the Stanford Dash & the MIT Alewife to be studied in Chapter 9.

A cache-coherent COMA machine is one in which all cache copies must be kept consistent.

4. What are the conditions of Parallism? Explain the types of data dependence.

⇒ Conditions of Parallism:-

* The ability to execute several program segments in parallel requires each segment to be independent of the other segments. We use a dependence graph to describe the relations

* The nodes of a dependence graph correspond to the program statement & directed edges with different labels are used to represent the ordered relations among the statements.

* The analysis of dependence graphs shows where opportunity exists for parallelization & vectorization.

⇒ Data Dependences:-

(i) Flow dependence :- A statement S2 is flow dependent on S1 if an execution path exists from S1 to S2 & if at least one o/p of S1 feeds in as i/p to S2. also called as RAW hazard & denoted as $S_1 \rightarrow S_2$.

(ii) Antidependence :- Statement S2 is antidependence on the statement S1 if S2 follows S1 in the program order & if the o/p of S2 overlaps the i/p to S1 also called as RAW hazard & denoted as $S_1 \dashrightarrow S_2$.

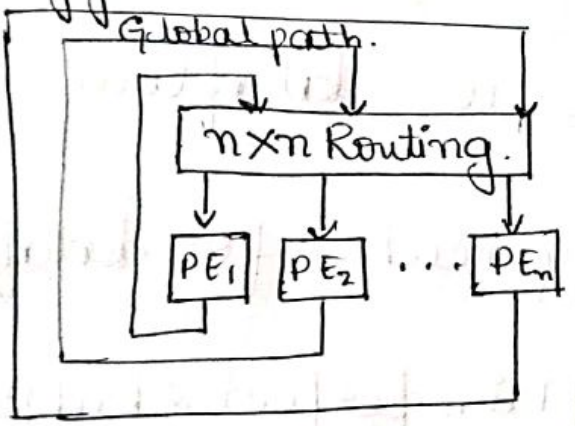
(iii) Output dependence :- Two statements are o/p dependent if they produce the same o/p variable, also called WAW hazard & denoted as $S_1 \circ\!\!\rightarrow S_2$.

(iv) I/O dependence :- Read & write are I/O statements. I/O dependence occurs not because the same variable is involved but because the same file referenced by both I/O statement.

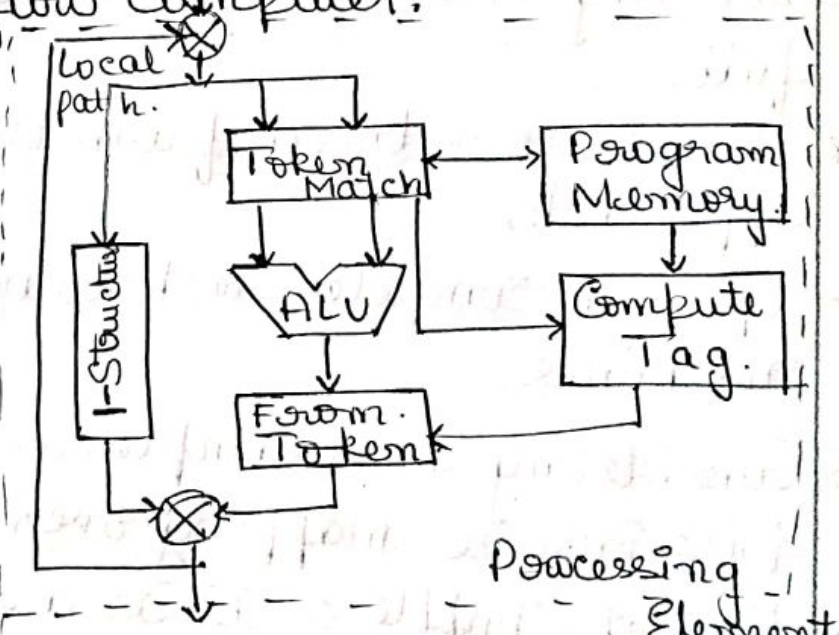
(v) Unknown dependence :- The dependence relation b/w two statements cannot be determined by * The subscript of a variable is itself subscribed. * The subscript does not contain the loop index variable. * A variable appears more than once with subscripts having different co-efficients of the loop variable. * The subscript is non-linear in the loop index variable.

5. With neat diagram, explain the operation of tagged token data flow computer.

A :-



Global architecture



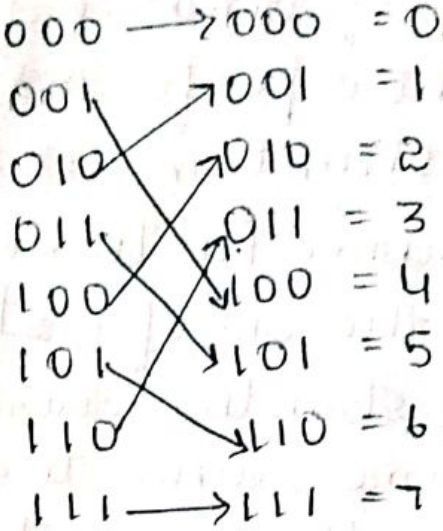
* The Arwind machine (MIT) has N PEs & an N-by-N interconnection N/w.
* Each PE has a token-matching mechanism that dispatches only instructions with data tokens available.
* Tagged tokens enter PE through local path & can also be communicated to other PEs through the routing N/w.

* Instruction address (ds) effectively replace the ~~frame base register~~ program counter in a control flow machine.
* Context identifier effectively replaces the frame base register in a control flow machine.
* Since the dataflow machine matches the data tags from one instruction with successors, synchronized instruction execution is implicit.
* Each word of the I-structure has a two-bit tag indicating whether the value is empty, full.
* This is a retreat from the pure dataflow approach.
* Special compiler technology needed for dataflow machines.

6. Considering an example; explain perfect shuffle & its inverse mapping over eight objects.

Ans- Perfect Shuffle :- It is a special permutation function suggested by Harold Stone for parallel processing applications. The mapping corresponding to a perfect shuffle is shown in figs below. It's inverse is shown on the right-hand side.

Perfect shuffle

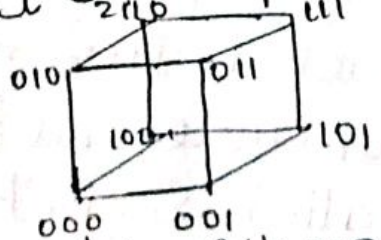


Inverse perfect shuffle

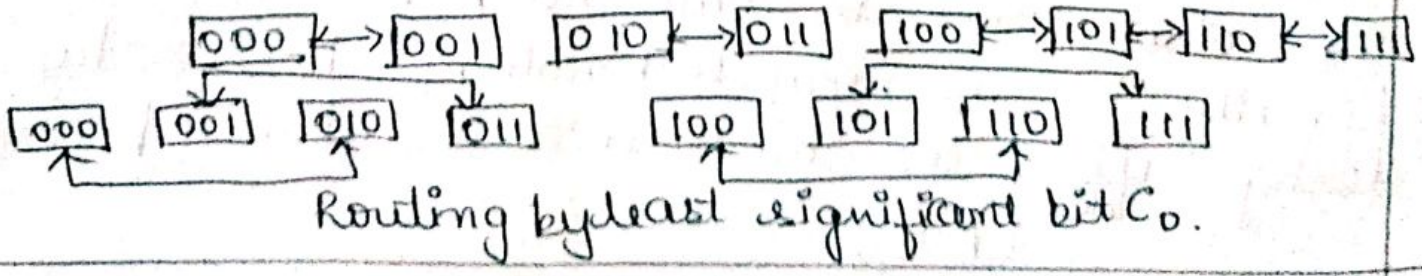
7. Considering a 3D binary cube N/w, explain the hyperCube routing functions for all 3 bits.

Ans: A 3D binary cube N/w is shown in fig. 3 routing functions are defined by three bits in the node address. Eg:- one can exchange the data b/w adjacent nodes which differ in the least significant bit C_0.

III^{dy}. two other routing patterns can be obtained by checking the middle bit C_1 & the most significant bit C_2 respectively.



A 3 cube with nodes $C_2 C_1 C_0$



8. List the performance factors & system attributes Explain how performance factors are influenced by stepsystem attributes.

⇒ Performance Factors :-

* Let I_c be the no of instructions in a given prg. or the instruction count.
* The CPU time needed to execute the prg is estimated by finding the product. , $T = I_c \times CPI \times \tau$.
* The execution of an instruction requires going through a cycle of events involving the instr fetch, decode, operand fetch, execution & store results.
* In this Cycle, only the instruction decode & execution phases are carried out in the CPU.
* Depending on the instruction type, the complete instruction cycle may involve one to four memory references.

$\therefore T = I_c \times (p + m \times k) \times \tau$.

⇒ System Attributes :-

* The above five performance factors (I_c, p, m, k, τ) are influenced by four system attributes :.
* The instruction-set architecture affects the prg length & processor cycle needed (p).
* The CPU implementation & control determine the total processor time needed.
* Finally the memory technology & hierarchy design affect the mem access latency processor. (k.τ).

System Attributes	Performance Factors				
	Instr. Count	Average Cycles per Inst CPI			Processor Cycle Time T
		Processor Cycles per Inst, p	Memory References per Instn	Memory-Access Latency, k	
Instruction set Architecture	X	X			
Compiler Technology	X	X	X		
Processor Implementation & Control		X			X
Cache & Memory Hierarchy				X	X

9. Explain Amdahl's law for a fixed workload & Gustafson's law for scaled problems.

⇒ Amdahl's Law says that for fixed workloads, the upper bound of the speedup expected from moving to a more parallelized environment gets dominated by the sequential portion of the code as the parallel portion gets further divided into smaller & smaller chunks.

Amdahl's law is often used in parallel computing to predict the theoretical speedup. $\left(\frac{1}{1-p}\right)$ is the formula used to compute theoretical speedup.

The execution time of whole task is denoted as T

$T = (1-p)T + pT.$

$T(s) = (1-p)T + \frac{p}{s}T$ ⇒ theoretical execution, T(s)

$S_{latency}(s) = \frac{Tw}{T(s)w} = \frac{T}{T(s)} = \frac{1}{1-p+\frac{p}{s}}$ ⇒ Theoretical Speedup.

⇒ Gustafson's law :-

Law says that if you apply 'p' processors to a task that has serial fraction f, scaling the task to take the same amount of time as before the speedup.

The execution workload of the whole task before the improvement of the resources of the system is denoted by 'w'. The fraction of the execution workload that would benefit from the improvement of the resources is denoted by 'p'.

The fraction concerning the part that would not benefit from it is ∴ 1 - p.

$$\therefore w = (1-p)w + pw.$$

The theoretical execution workload w(s) of the whole task after the improvement of resources is

$$w(s) = (1-p)w + spw.$$

The theoretical speedup in latency of the execution of the whole task at fixed time.

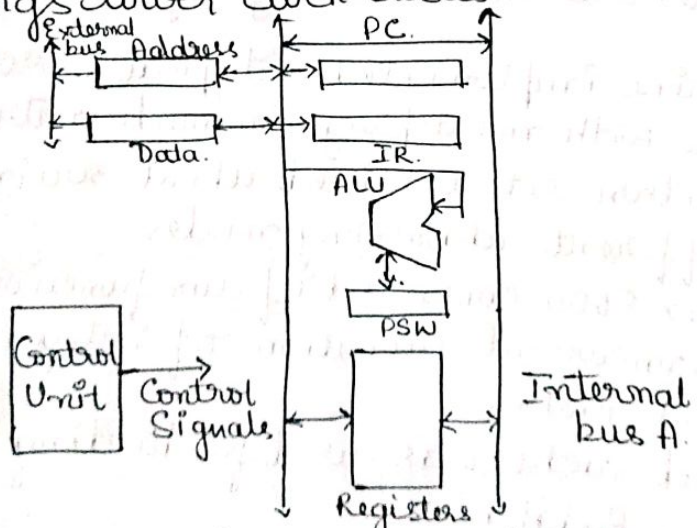
T,

$$S_{latency}(s) = \frac{Tw(s)}{Tw} = \frac{w(s)}{w} = 1 - p + sp.$$

Module - 2.

1. With diagrams, explain the models of a basic scalar computer system.

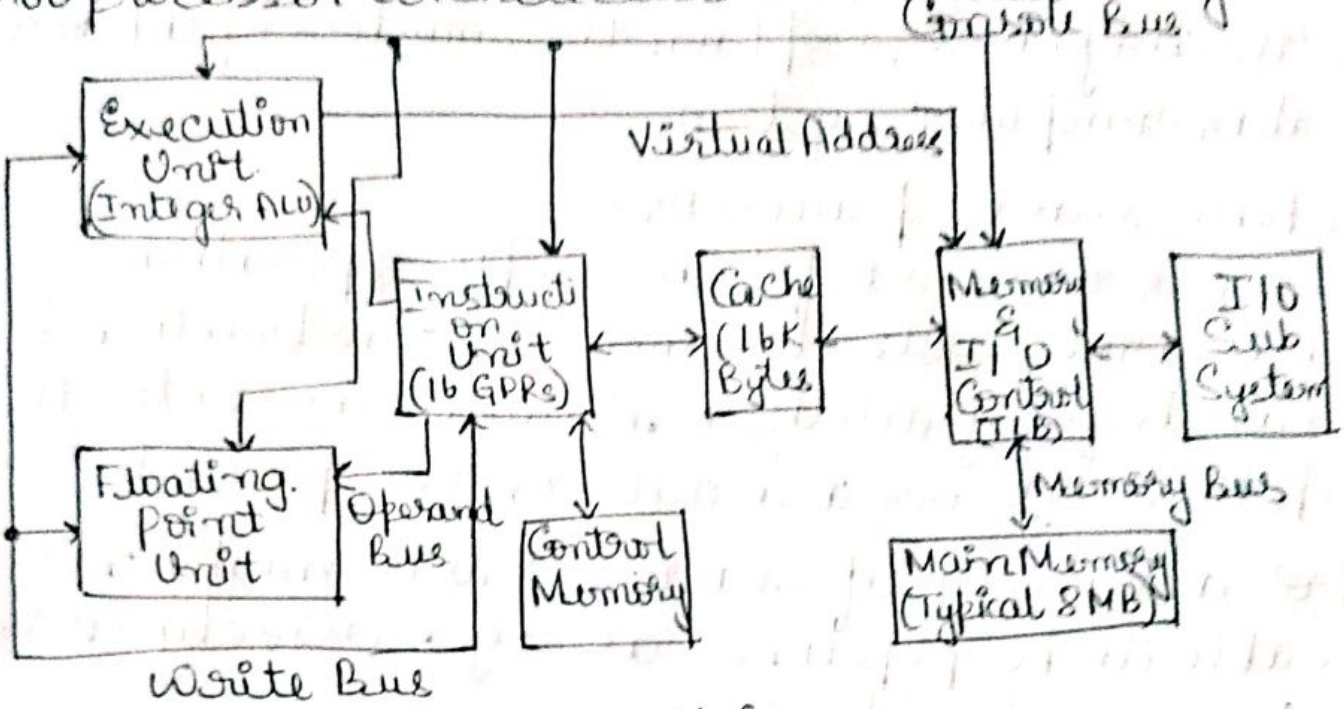
* A base scalar processor :-
  → issues one inst for a simple operation.
  → has a one-cycle latency b/w instruction issues
  → can be fully utilized if inst can enter the pipelines at a rate on one per cycle.
* For a variety of reasons, instr might not be able to be pipelines as aggressively as in a base scalar processor.
* CPI rating is 1 for an ideal pipeline. Underpipelined systems will have higher CPI ratings lower clock rates or both.



* Fig shows the data path architecture & Control unit of a typical, simple scalar processor.
* Main memory, I/O controllers etc are connected to external bus.

2. Explain VAX8600, Motorola MC68040, Intel i860 processor architecture with neat diagram.

(a)



CPU - Central Processor Unit
TLB - Translation Lookaside Buffer.
GPR - General Purpose Register.

* The VAX 8600 was introduced by Digital Corporation in 1985.
* This machine implemented a typical CISC architecture with microprogrammed control
* The instruction set contained about 300 instr. with 20 different addressing modes.
* The CPU VAX 8600 consisted of two functional units for concurrent execution of integer & floating point instr.
* The unified cache was used for holding both instructions & data.
* There were 16 GPRs in the instruction unit. Instruction pipelining was built with six stages in the VAX 8600, as in most elec machines.

* The instruction unit prefetched & decoded instr. handled branching operations, & supplied operands to the two functional units in a pipelined fashion.
* A Translation Lookaside Buffer (TLB) was used in the memory control unit for fast generation of a physical address from a virtual address.
* Both integer & floating point units were pipelined

3. Give the architectural distinctions & characteri-stics of CISC & RISC architectural.

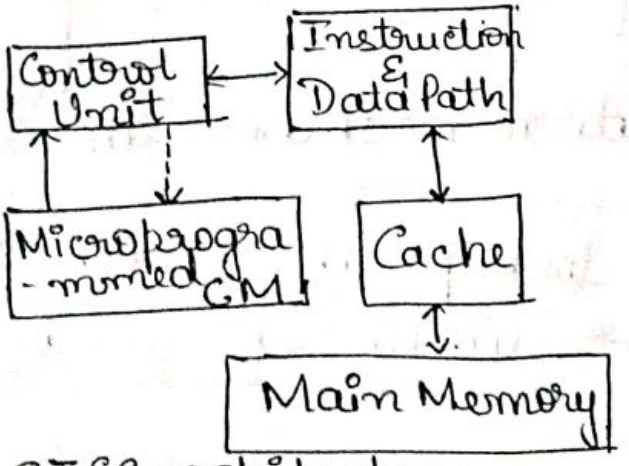
⇒ Architectural Distinctions :-

CISC :-

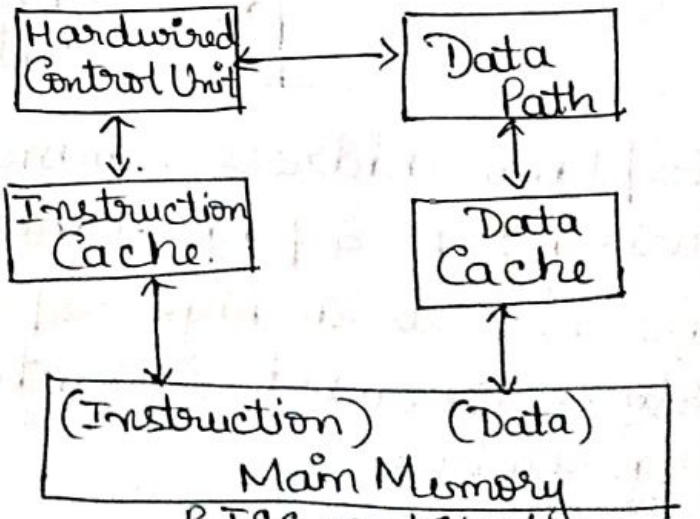
* Unified cache for instructions & data.
* Microprogrammed control units & ROM in earlier processors.

RISC :-

* Separate instruction & data caches.
* Hard-wired units.



CISC architecture



RISC architecture

=> Characteristics :-

Architectural Characteristic	Complex Inst Set Computer (CISC)	Reduced Inst Set Computer (RISC)
Instruction-set size & instruction formats	Large set of inst with variable formats.	Small set of inst with fixed format & most registered - based inst.
Addressing modes	12-24	Limited 3-5.
General-purpose register & Cache design	8-24 GPR's, originally with a unified cache for instructions & data, recent designs also use split caches.	Large no of GPR's with mostly split data. cache & instruction cache.
CPI	CPI b/w 2 & 15	One cycle for almost all instructions & an avg CPI <1.5
CPU Control	Earlier microcoded using control memory, but modern CISC also used hardwired control	Hardwired without control memory.

4. Explain address translation mechanism using TLB & page table.

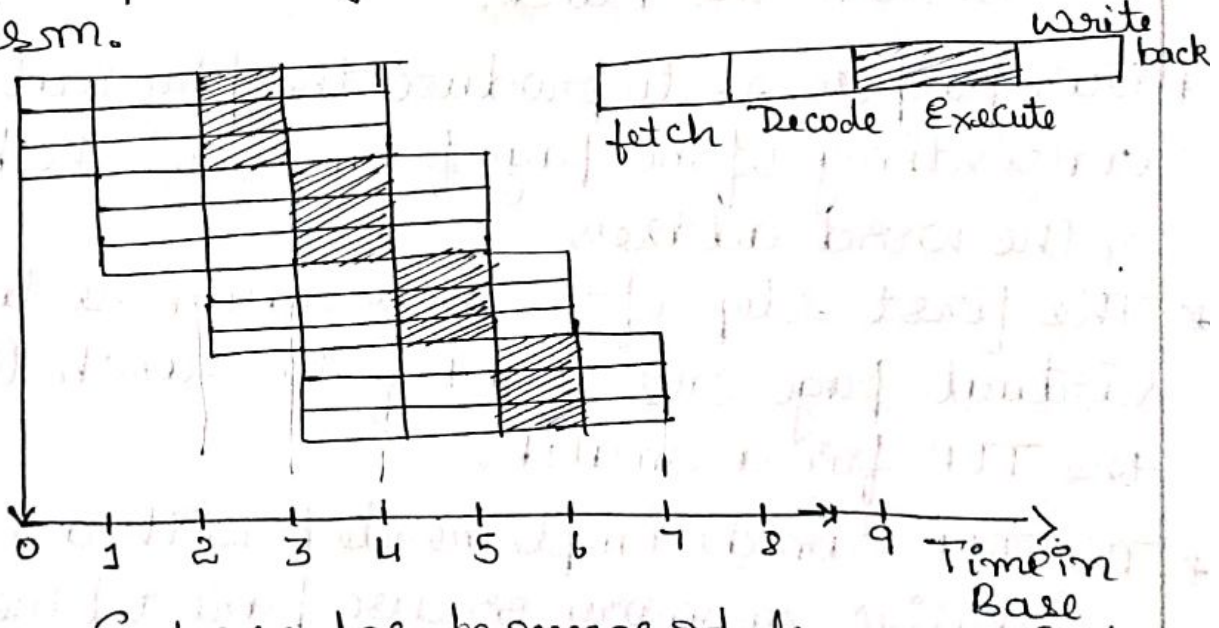
* The TLB is a high-speed lookup table which stores the most recently or likely referenced page entries.

* A page entry consists of essentially a pair. It is hoped that pages belonging to the same working set will be directly translated using the TLB entries.
* The use of a TLB & PTs for address translation is shown in given fig. Each virtual address is divided into 3 fields:
  → The leftmost field holds the virtual page no.
  → the middle field identifies the cache block no.
  → the rightmost field is the word address within the block.
* Our purpose is to produce the physical address consisting of the page frame no, the block no, & the word address.
* The first step of the translation is to use the virtual page no as a key to search through the TLB for a match.
* The TLB can be implemented with a special associative memory or use part of the cache memory.
* In case of a match in the TLB, the page frame no is retrieved from the matched page entry. The cache block & word address are copied directly.

* In case the match cannot be found in the TLB, a hased is used to identify one of the page tables where the desired page frame no. can be retrived.

5. Explain the pipelining in superscalar processor with degree = 3.

* The fundamental structure of a 3-issue super-scalar pipeline shown in below fig.
* Superscalar processors were originally developed as an alternative to vector processors, with a view to exploit higher degree of inst level parallelism.



Superscalar processor of deg. m=3. Cycle

* A superscalar processor of degree 'm' can issue 'm' instr. per cycle.
* The base scalar processor, implemented either in RISC or CISC, has m=1.
* In order to fully utilize a superscalar processor of degree m, m-instr must be executable in parallel.

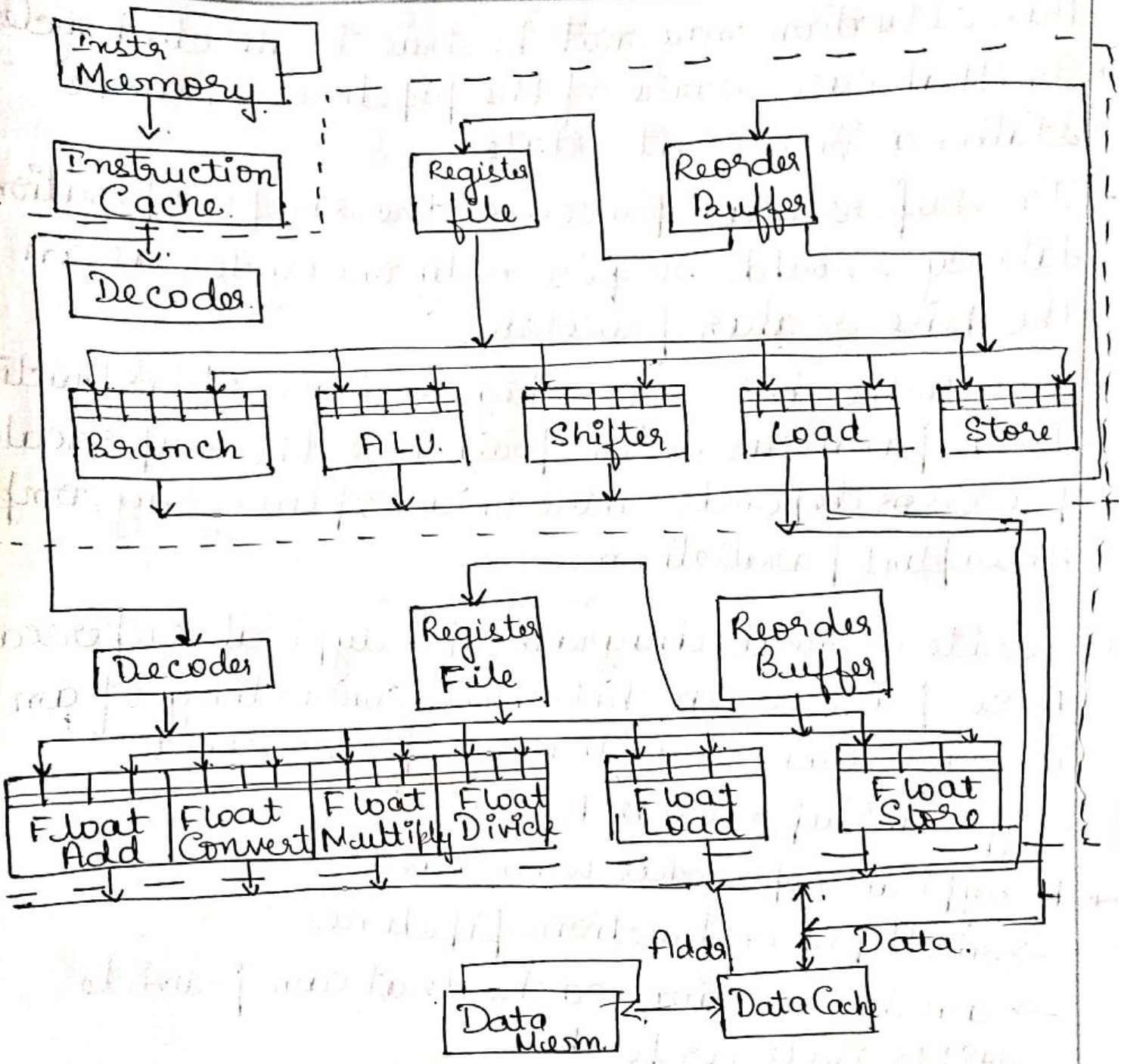
This situation may not be true in all clock cycles.

* In that case, some of the pipelines may be stalling in a wait state.
* In superscalar processor, the simple operation latency should require only one cycle, as in the base scalar processor.
* Due to the desire for a higher degree of instruction level parallelism in programs, the superscalar processor depends more on an optimizing compiler to exploit parallelism.

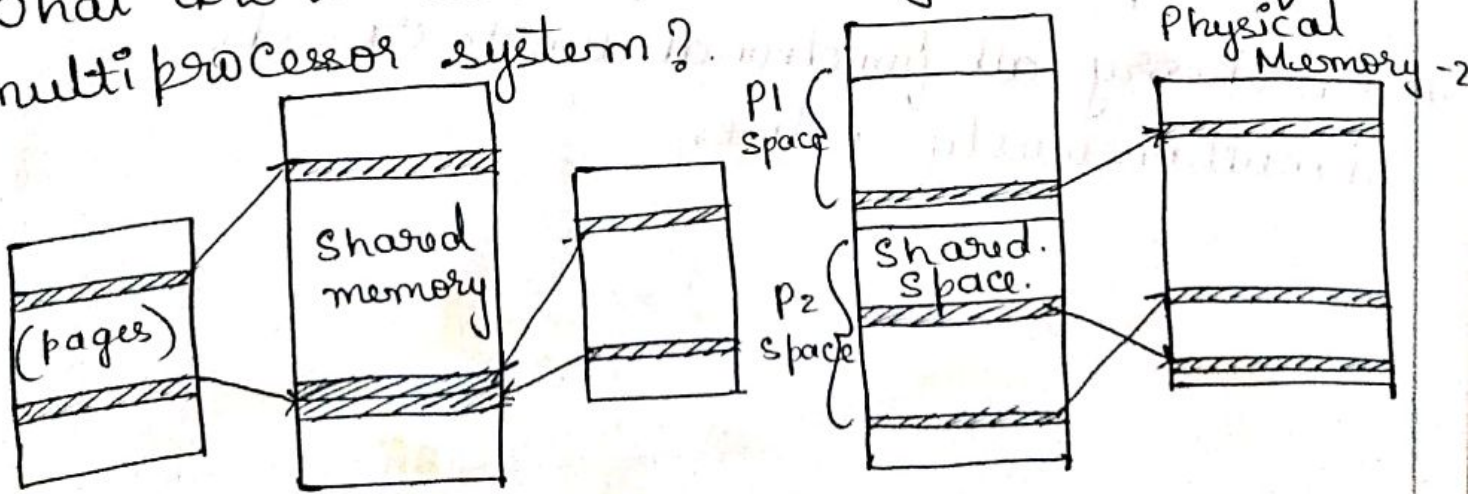
6. Write a neat diagram of a typical superscalar RISC processor architecture consisting of an integer unit & a floating point unit.

⇒ Typical Superscalar Architecture :-

* A typical superscalar will have
  → multiple instruction pipelines
  → an instruction cache that can provide multiple instr per fetch.
  → multiple buses among the function units.
* In theory, all functional units can be simultaneously active.



7. What are the virtual memory models of multiprocessor system?



(i) Private Virtual Memory :-

* In this scheme, each processor has a separate virtual address space, but all processors share the same physical address space.

=> Advantages :-

(i) Small processor address space.
(ii) Protection on a per-page or per-process basis.
(iii) Private memory maps, which require no locking.

=> Disadvantages :-

(i) The synonym problem - different virtual addresses in different / same virtual spaces point to the same physical page.

(ii) Shared Virtual Memory :-

* All processors share a single shared virtual address space, with each processor being given a portion of it.
* Some of the virtual address can be shared by multiple processor.
* ~~Some of the virtual~~

=> Advantages.

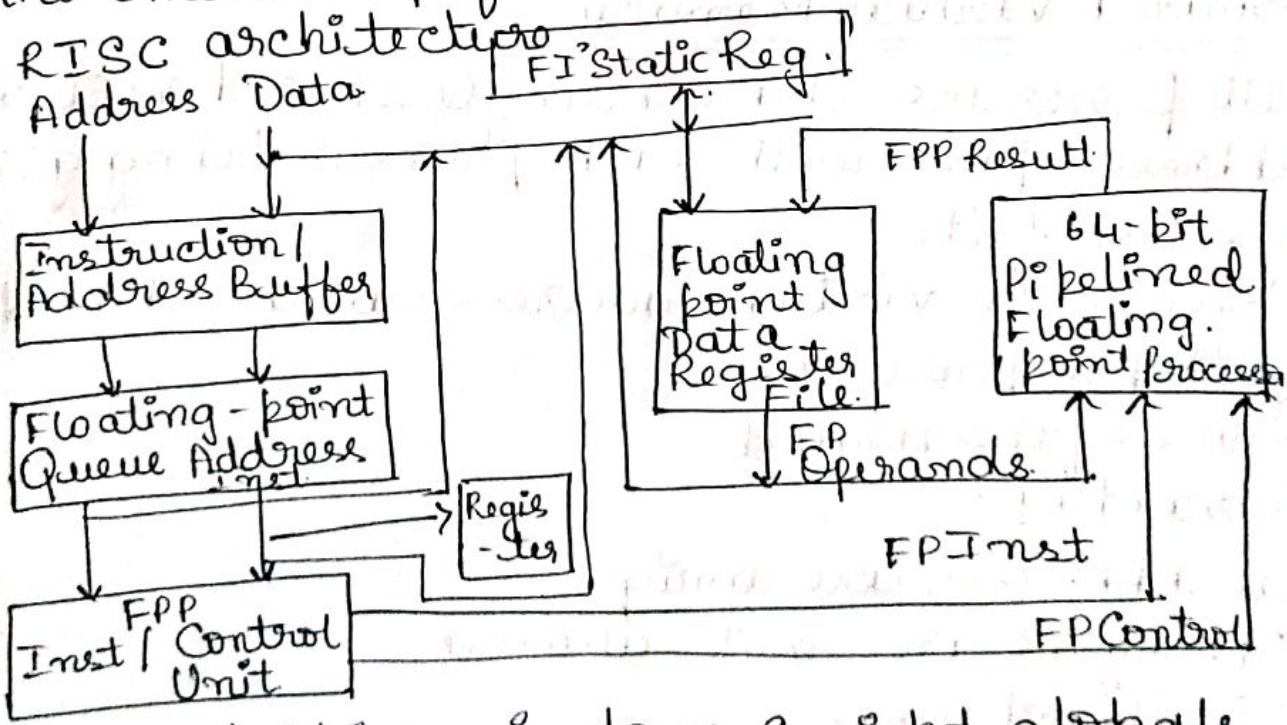
(i) All addresses are unique.
(ii) Synonyms are not allowed.

=> Disadvantages :-

(*) Processors must be capable of generating large virtual addresses.
(ii) Segmentation must be used to confine each proce
to its own address space

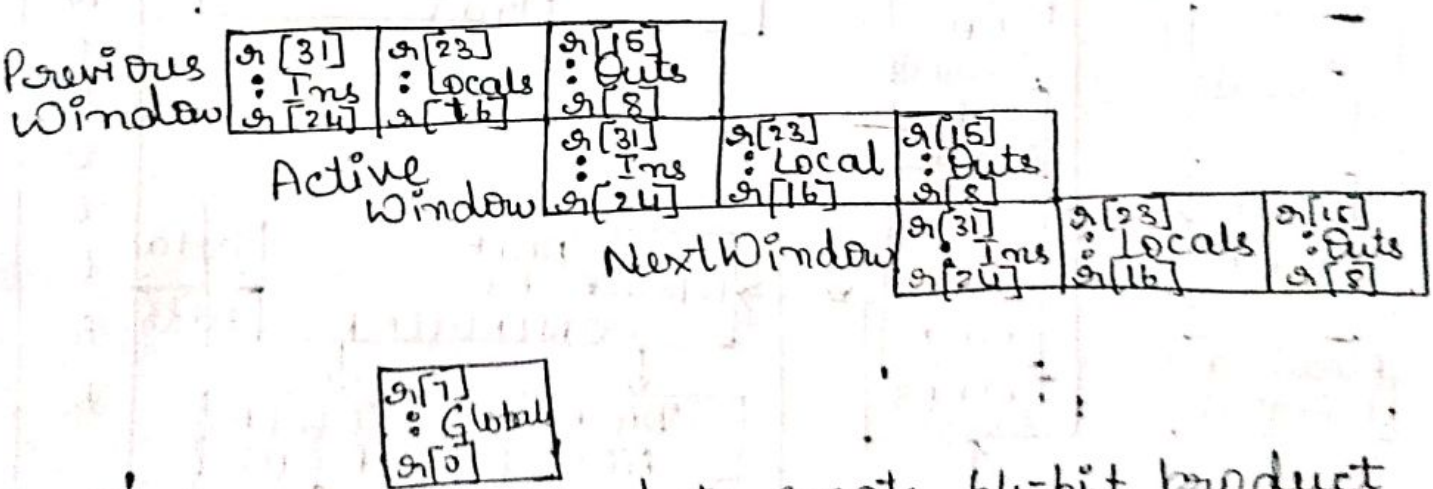
8. Explain the Sun Microsystems SPARC. architecture.

* SPARC family chips produced by Cypress Semiconductors as shown in fig.
* The Sun SPARC instruction set contains 69 basic instr.
* The SPARC. runs each procedure with a set of thirty-two 32-bit IU reg.
* Eight of these registers are global regs. shared by all procedures, & the remaining 24 are window reg. associated with only procedure.
* The concept of using overlapped reg. windows is the most imp feature introduced by the RISC architecture



* Eight overlapping windows & eight globals with a total of 136 reg. as implemented in the Cypress 601.

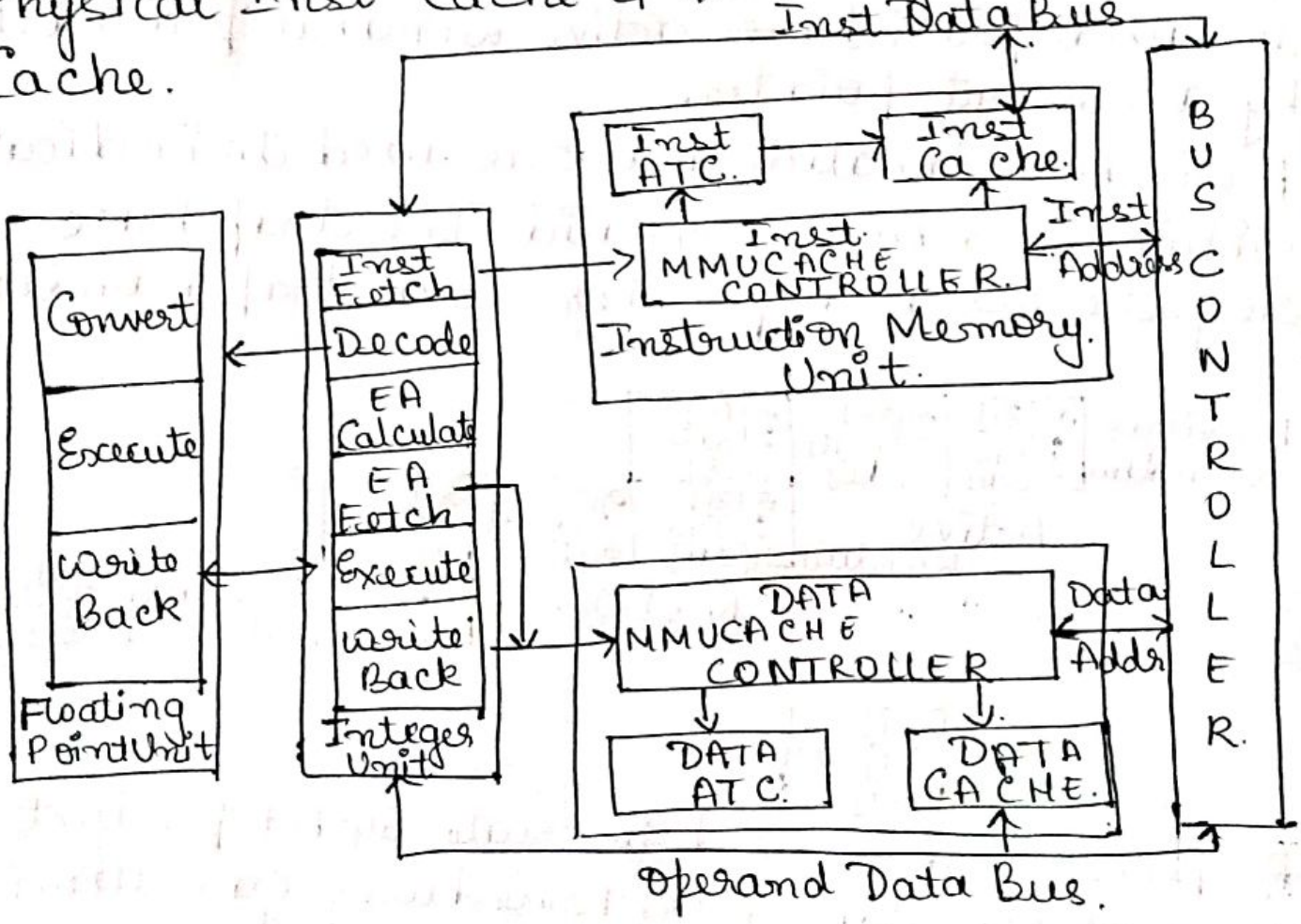
* Each register window is divided into three eight-reg. sections, labeled Ins, Locals & Outs.
* The local reg. are only locally addressable by each procedure. The Ins & Outs are shared among procedures.
* The calling procedure passes parameters to the called procedure via its Outs reg. which are the Ins reg. of the called procedure.
* The Window of the currently running procedure is called ~~procedure~~ active window pointed to by a current pointer.
* A window invalid mask is used to indicate which window is invalid. The trap base reg. serves as a pointer to a trap handler.



* A special reg. is used to create 64-bit product in multiple step instr. Procedures can also be called without changing the window.

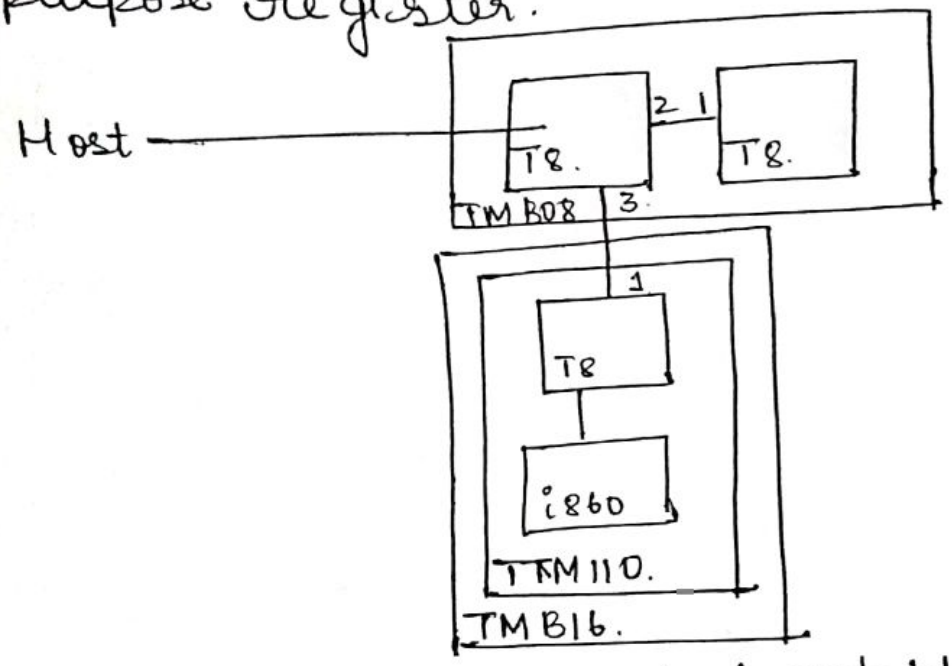
2(b) Motorola MC68040 Architecture :

* The Motorola 68040 is a 32-bit microprocess-or from Motorola released in 1990.
* MC68040. Data width - 32 bits & Address width - 32 bits.
* Main features of the MC68040 are 6 stage Pipeline, Independent Instruction & Data MMU's.
* It has Simultaneously Accessible 4KB Physical Inst Cache & 4KB Physical Data Cache.



2(c) Intelli 860 processor architecture :-

* Intel i 860 was a RISC microprocessor design introduced by Intel in 1989.
* It has 32/64 bits, it has 32, 32 bit General purpose register.



* All the buses were at least 64 bits wide. The i 860 combined a no of features that were unique at the time, most notably its very long inst word (VLIW). architecture.
* The internal memory bus to the cache, for instance, was 128 bits wide.
* One unusual feature of the i 860 was that the pipelines into the functional units were program - accessible, requiring the compilers to order instructions carefully in the object code to keep the pipeline filled.