USN | | | | | | | | | | |

Internal Assessment Test 4 – FeB 2022

Sub:	Advanced Computer Architecture					Sub Code:	17CS72	Branch:	CSE
Date:		Duration:	90 min's	Max Marks:	50	Sem / Sec:	VII SEM D	OBE	

	Answer any FIVE FULL Questions	MARKS	CO	RBT
	1. Explain Flynn's classification of computer architecture.		CO1	L2
	2 .What are the conditions of parallelism? Explain the types of data dependence.	[10]	CO2	L3
	3. What is Amdahl's law? Explain efficiency, utilization, and quality of parallelism.	[10]	CO2	L3
	4. Explain the architecture of a vector supercomputer with a neat diagram.	[10]	CO1	L2
	5. Explain with a diagram the operational model of SIMD supercomputer.	[10]	CO1	L2
	6. Define data dependency. Explain different functions of data dependency with the help of dependency graph.	[5+5]	CO2	L2
(b)	7. Distinguish between typical RISC and CISC processor architectures.	[10]	CO2	L2
	8. Explain the inclusion property and locality of reference along with its types in multilevel memory hierarchy.	[10]	CO2	L2

Explain Flynn's Classification of Computer architecture along with neat diagram.

Michael Flynn introduced a classification of various computer architectures based on notions of instruction and data streams.

Single Instruction Stream Single Data Stream(SISD)

- It is uniprocessor system
- Single instruction is executed by CPU in one clock cycle.
- Instructions are executed sequentially
- Workstations of DEC, MP & IBM, IBM 701, IBM 1620, IBM 7090 etc.

Single Instruction Stream Multiple Data stream(SIMD)

- A single instruction is executed by multiple processing elements or processors. Each processing element operates on different data.
- Data level parallelism is achieved.
- Example: Vector supercomputer in early 1970 like C

Machine CM2, Maspar MP-1, IBM 9000, Cray C90, Fujitsu VP.

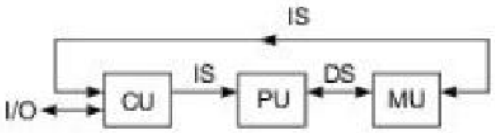
Multiple Instruction Stream Single Data Stream(MISD)

- The same data stream flows through a linear array of processor executing different instruction streams. This architecture is known as systolic arrays for pipelined execution of specific instructions.
- Few actual examples of this class of parallel computer have ever existed. One is the experimental Carnegie Mellon C.MPP computer (1971).
- Least popular model to be applied in commercial machines.

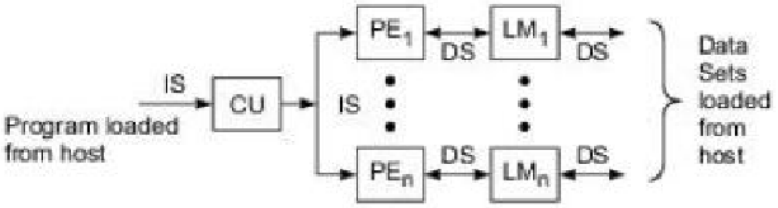
Multiple Instruction Stream and Multiple Data Stream(MIMD)

Most popular computer model

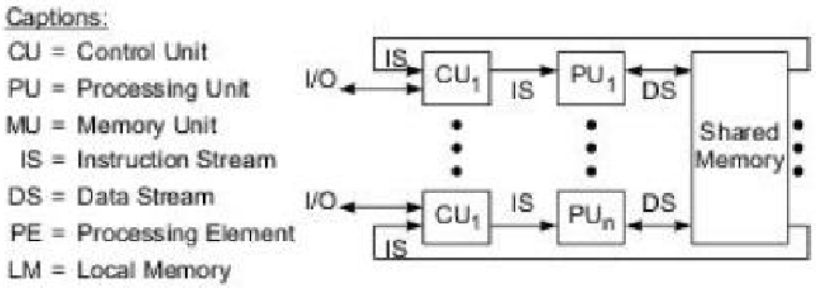
- Every processor may be executing different instruction stream and every processor uses different data stream.
- They are also called as parallel computers.
- Example: IBM 370/168MP, Univac 1100/80
- Parallel computers operate in MIMD mode.
- There are two types of MIMD i.e. shared memory multiprocessor and distributed memory multicomputer.
- In shared memory multiprocessor system all processors have common shared memory and can communicate through shared variables.
- In distributed multicomputer system each computer node has local memory ui with other nodes. Inter-processor communication is done though message passin among the nodes.



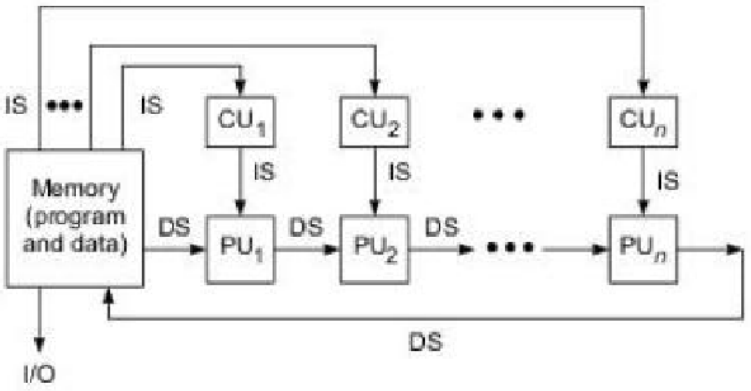
(a) SISD uniprocessor architecture



(b) SIMD architecture (with distributed memory)



(c) MIMD architecture (with shared memory)



(d) MISD architecture (the systolic array)

What are the conditions of parallelism? Explain the types of data dependence

Conditions of Parallelism

It is necessary to detect and exploit the parallelism for parallel computing. The significant progress is needed in following areas

1. Development of computational models for parallel computers (e.g. PRAM model)
2. Inter processor communication in Parallel architecture
3. System integration of parallel architecture in general computing environment.

Data and Resource Dependencies

Segment is collection of program instructions. Several segments can be executed parallel only when segments are independent of each other. The dependence relation among several instructions of the program is shown using dependence graph. The nodes of dependence graph correspond to the program statements [instructions], and the directed edges with different labels show the ordered relations among the statements. The analysis of dependence graphs shows

where opportunity exists for parallelization and vectorization.
Data Dependence: There are five types of Data Dependence as shown below
1. Flow Dependence: A statement S2 is flow dependent on S1 if at least one output of S1 Feeds in as input to S2. Flow Dependence is denoted as S1→ S2

2. Anti-Dependence: Statement S2 is anti-dependent on statement S1 if S2 follows S1 in Program order and if the output of S2 overlaps the input to S1. It is denoted as follows

3. Output Dependence: Two statements are output-dependent if they produce the same Output variable. It is denoted as follows

4. I/O Dependence: The read and write statements are I/O statements. The I/O dependence Occurs when the same file is referenced by both I/O statements.

5. Unknown dependence: The dependence relation between two statements cannot be Determined in the following situations:
The subscript of a variable is itself subscribed (indirect addressing mode)

LOAD R1, @100

- The subscript does not contain the loop index variable.
- A variable appears more than once with subscripts having different coefficients of the loop variable.
- The subscript is nonlinear in the loop index variable.

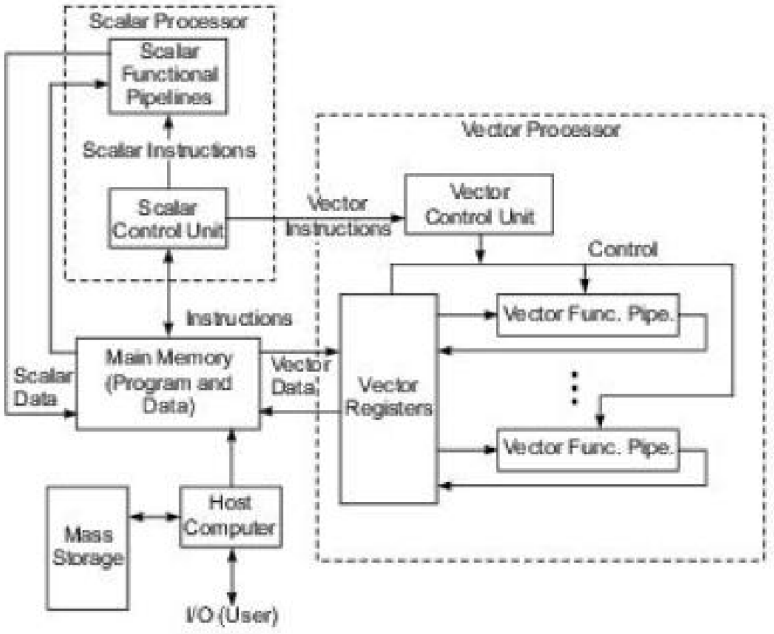
Explain the architecture of a vector supercomputer with a neat diagram.

The program and data are loaded into main memory from the host computer.
- All instructions are first decoded by the scalar control unit. If the decoded instruction is a scalar operation it will be directly executed by the scalar processor using the scalar functional pipelines.

- If the instruction is decoded as vector operation, it will be sent to vector control unit. The vector control unit manages the flow of vector data between vector functional units and main memory.

- There are multiple vector functional units which are pipelined. Data is forwarded from one vector functional unit to another i.e. called vector chaining.

There are two types of vector processor
- Register Register vector processor
- Memory Memory vector processor



Register Register Vector Processor

- Vector registers are used to hold the vector operands, intermediate and final vector results.

- The vector functional pipelines retrieve operands from and put results into the vector registers. All vector registers are programmable for user instructions.

- The length of each vector register is usually fixed, say, sixty-four 64-bit component registers in a vector register in a Cray Series supercomputer Memory to Memory Vector Processor
- Vector operands and results are directly retrieved from and stored into the main memory in super words, say, 512 bits as in Cyber 205.

. Explain with a diagram the operational model of SIMD supercomputer

The operational model of SIMD machine is specified by a 5-tuple

M=(N,C,I,M,R)

- N is the number of processing elements (PEs) in the machine. For example, the Illiac IV had 64 PEs and the Connection Machine CM-2 had 65,536 PEs.
- C is the set of instructions directly executed by the control unit (CU).
- I is the set of instructions broadcast by the CU to all PEs for parallel execution.
- M is the set of masking schemes, where each mask partitions the set of PEs into enabled and disabled subsets.
- R specifies the data routing schemes to be followed during inter PE communication.

Dgm required.

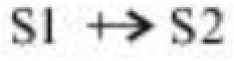
Define data dependency. Explain different functions of data dependency with the help of dependency graph.

Data Dependence: There are five types of Data Dependence as shown below

- Flow Dependence: A statement S2 is flow dependent on S1 if at least one output of S1 feeds in as input to S2. Flow Dependence is denoted as S1→ S2

🎬 Anti Dependence: Statement S2 is anti-dependent on statement S1 if S2 follows S1 in program order and if the output of S2 overlaps the input to S1. It is denoted

as follows



Output Dependence: Two statements are output-dependent if they produce the same output variable. It is denoted as follows

I/O Dependence: The read and write statements are I/O statements. The I/O dependence occurs when the same file is referenced by both I/O statements.

Distinguish between typical RISC and CISC processor architectures.

Architectural Consideration	**CISC**	**RISC**
Instruction set size and format	Large set of instructions with variable format(16-64 bits per instruction)	Small set of instructions with fixed format(32 bit per instruction)
Addressing modes	12-24	3-5
General purpose register and cache design	8-24 GPR and unified cache	32-192 GPR and split cache
CPI	Between 2 and 16	Average CPI<1.5
CPU control	Micro programmed	Hardwired

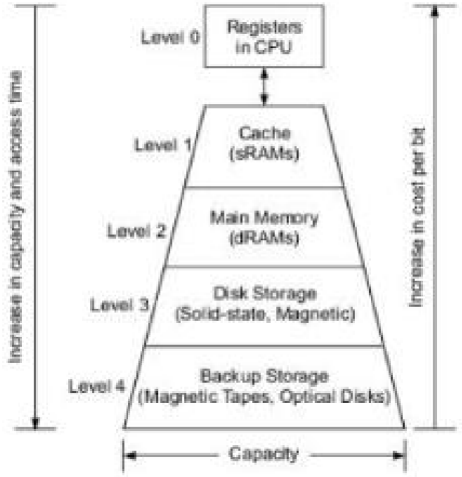
Explain the inclusion property and locality of reference along with its types in multilevel memory hierarchy.

The storage devices like register, cache, main memory, disk devices and backup storage devices are organized in a form of hierarchy as shown below. The cache is at level 1, main memory at level 2, disk at level 3 and backup storage at level 4.

🎬 Memory devices at a lower level are faster to access, smaller in size, and more expensive per byte, having a higher bandwidth and using a smaller unit of transfer as compared with those at a higher level.

🎬 The access time ti refers to the round-trip time from the CPU to the ith-level memory. The memory size si is the number of bytes or words in level i. The cost of the ith-level memory is estimated by the product cisi. The bandwidth bi refers to the rate at which information is transferred between adjacent levels. The unit of transfer xi refers to the grain size for data transfer between levels i and i+1.

🎬 Also ti-1<ti,si-1<si,ci-1>ci,bi-1>bi and xi-1<xi for i=1,2,3,4 in the hierarchy.



Inclusion

The inclusion property is stated as M1⊂M2⊂M3⊂…...⊂MN. The inclusion relationship implies that all information items are originally stored in the outermost level MN. During the processing, subsets of MN, are copied into MN-1. Similarly, subsets of MN-1 are copied into MN-2 and so on. Hence if word is found in Mi then same word can be found in Mi+1, Mi+2 and so on but may not be found in Mi-1

Coherence

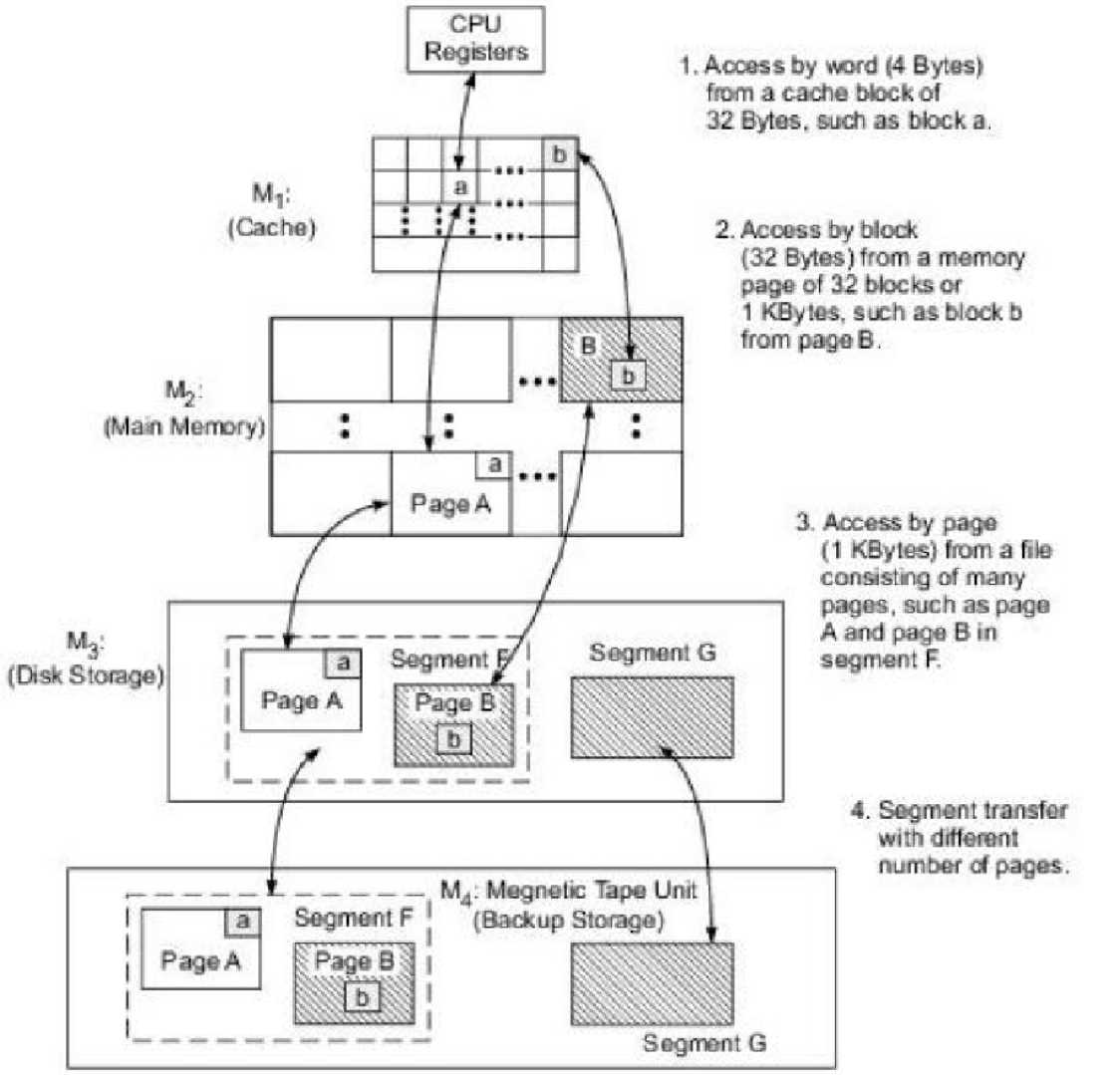


Fig. 4.18 The inclusion property and data transfers between adjacent levels of a memory hierarchy

Coherence

The coherence property requires that copies of the same information item at successive memory levels must be consistent. If a word is modified in the cache, copies of that word must be updated immediately or eventually at all higher levels. In general, there are two strategies for maintaining the coherence in a memory hierarchy. The first method is called write-through (WT), which demands immediate update in Mi+1 if a word is modified in Mi. The second method is write-back (WB), which delays the update in Mi+1 until the word being modified in Mi is replaced or removed from Mi.

Locality of References

The CPU refers memory to either access the instructions or data. The memory references can be clustered according to time, space or ordering. Hence there are three dimensions for locality of reference i.e. temporal, spatial and sequential locality.

Temporal Locality: Recently referenced items i.e. instructions or data are likely to be referenced again in the near future. This is often caused by special program constructs such as iterative loops, process stacks, temporary variables, or subroutines. Once a loop is entered or a subroutine is called, a small code segment will be referenced repeatedly many times.

Spatial Locality: This refers to the tendency for a process to access items whose addresses are near one another. For example, operations on tables or arrays involve accesses of a certain clustered area in the address space.

Sequential Locality: In typical programs, the execution of instructions follows a sequential order unless branch instructions create out-of-order executions. The ratio of in-order execution to out-of-order execution is roughly 5 to 1 in ordinary programs. Besides, the access of large data array also follows a sequential order.