Name :- Tanishq Chauhan

Roll No :- 21C3184

Subject :- Computer Organization & Architecture (CER36G1)

Semester :- IIIrd Semester

Branch :- Computer Science & Engineering

Tanishq Chauhan
21C3184
CS-B

Chauhan
Date
Page 01

Computer Organization & Architecture (CER36I1)

Q.1. On which concept Flynn's classification is based ? Explain MIMD in detail with proper example.

Answer of Q. No. 1

- M.J Flynn proposed a classification for the organization of a computer system by the number of instruction and data items that are manipulated simultaneously.
- Flynn's classification is based on the multiplicity of Instruction Streams and Data Streams.

Instruction Stream

- The sequence of instructions read from memory constitutes an instruction stream.

Data Stream

- The operation performed on the data in the processor constitutes a data stream.
- A data stream is sequence of data having input, partial or temporary result.
- Based on the multiple data and instruction streams Flynns classify the digital computers in four categories.

Tanishq Chauhan
21C3184
CS-B

1. Single Instruction Stream, Single Data Stream (SISD)
2. Single Instruction Stream, Multiple Data Stream (SIMD)
3. Multiple Instruction Stream, Single Data Stream (MISD)
4. Multiple Instruction Stream, Multiple Data Stream (MIMD)

		Number of Data Streams	
		Single	Multiple
Number of Instruction Streams	Single	SISD	SIMD
	Multiple	MISD	MIMD

Figure 1.1 : Flynn's Classification

* Multiple Instruction Stream, Multiple Data Stream (MIMD)

- MIMD stands for "Multiple Instruction and Multiple Data Stream".
- In this organization, all processors in a parallel computer can execute different instruction and operate on various data at the same time.
- In MIMD, each processor has a separate program and an instruction stream is generated from each program.
- Each PE in the MIMD model has separate instruction and data streams; therefore machines built using this model are capable to any kind of application.
- Unlike SIMD and MISD machines, PEs in MIMD machines work asynchronously.
- MIMD machines are broadly categorized into shared-memory MIMD and Distributed-Memory MIMD based on the way PEs are coupled to the main memory.

Tanishq Chauhan
21C3184
CS-B

- Most multiprocessor and multicomputer systems can be classified in this category.
- When multiple SISD works together than its called MSISD, which comes under category of MIMD.
- If number of instructions are high than it's known as tightly coupled else known as loosely coupled.
- It contains multiple processing units.
- Execution of multiple instructions on multiple data.

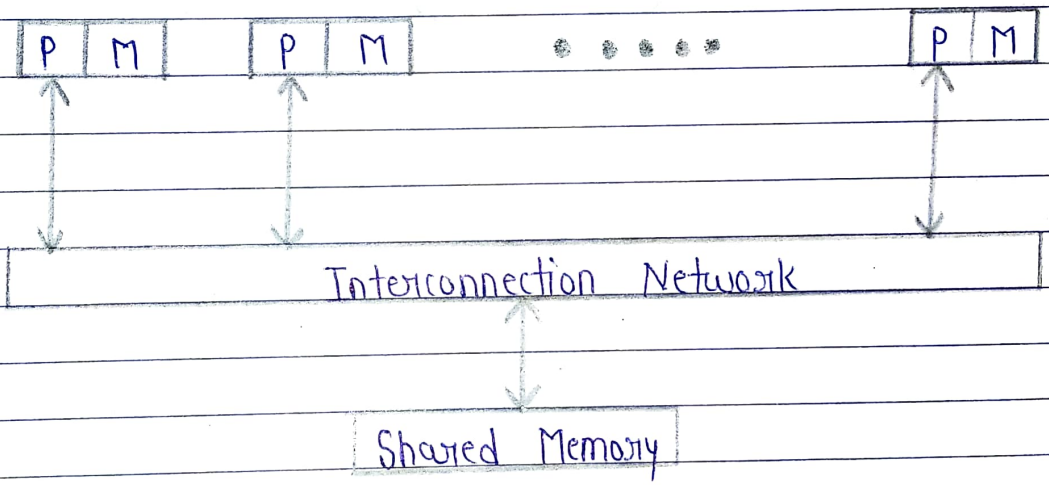


Figure 1.2 : MIMD Organization

- Examples :- IBM-SP2, Cray-2, Cray T90, IBM370.

Tanishq Chauhan
21C3184
CS-B

Chauhan

Date
Page 04

Q.2. Differentiate between Computer Organization and Computer Architecture in tabular format only.

Answer of Q. No. 2

S.No.	Computer Organization	Computer Architecture
1.	This is the design of computer and functional block using which computer systems are built.	It consists of those attributes of the system that are visible to the programmer.
2.	Computer organization describes how computer does it.	Architecture describes what the computer does.
3.	Computer organization deals with structural relationship.	Computer architecture deals with functional behaviour of the system.
4.	Its deals with low-level design issue.	It deals with high-level design issue.
5.	For designing a computer, organization is decided after its architecture.	For designing a computer, its architecture is fixed first.
6.	Computer organization is frequently called as micro architecture.	Computer architecture is also called as instruction set architecture
7.	Computer organization handles the segments of the network in a system.	Architecture coordinate between the hardware and software of the system.
8.	Computer organization consists of physical units like circuit designs, peripherals and adders.	Computer architecture comprises logical function such as instruction sets, registers, data types and addressing modes.
9.	Organization indicates its performance.	Architecture indicates its hardware.

Tanishq Chauhan
21C3184
CS-B

Chauhan
Date 05
Page

Q.3. Why addressing modes are required ? Explain Immediate, Register, Indirect, Direct addressing mode and Indirect addressing mode how effective address is calculated in each case ?

Answer of Q. No. 3

- The different ways of specifying the location of an operand is an instruction are called as addressing modes.
- The way the operands are chosen during program execution is dependent on the addressing mode of the instruction.
- The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is actually referenced.
- Computers use addressing mode techniques for the purpose of accommodating one or both of the following provisions:

(i) To give programming versatility to the user by providing such facilities as pointers to memory, counters for loop control, indexing of data, and program relocation.

(ii) To reduce the number of bits in the addressing field of the instruction.

Immediate Addressing Mode (Symbol #):

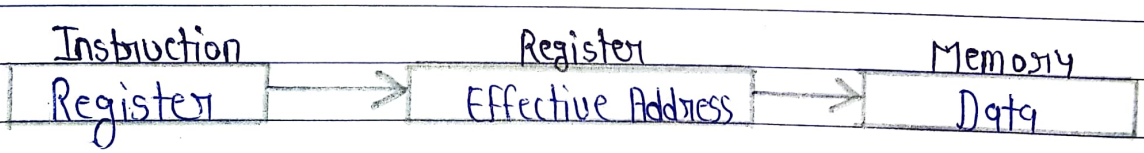
- In this mode the operand is specified in the instruction itself. In other words, an immediate - mode instruction has an operand field rather than an address field.
- The operand field contains the actual operand to be used in conjunction with the operation specified in the instruction.

Tanishq Chauhan
21C3184
CS-B

- Immediate mode of instructions is useful for initializing register to constant value.
- Examples -
  (i) ADD 10, will increment the value stored in the accumulator by 10. 10 is operand here.
  (ii) MOV R #20 initializes register R to a constant value 20.

Register Indirect Addressing Mode :

- In this mode the instruction specifies a register in the CPU whose contents give the address of the operand in memory.
- Before using a register indirect mode instruction, the programmer must ensure that the memory address of the operand is placed in the processor register with a previous instruction.
- The advantage of this mode is that address field of the instruction uses fewer bits to select a register than would have been required to specify a memory address directly.



- Example :-
  ADD R will increment the value stored in the accumulator by the content of memory location specified in Register R.

$$AC \leftarrow AC + [[R]]$$

- This addressing mode is similar to indirect addressing mode. The only difference is address field of the instruction refers to a CPU register.

Tanishq Chauhan
21C3184
CS-B

Chauhan
Date
Page 07

Direct Addressing Mode (Symbol [ ]):

- In this mode the effective address is equal to the address part of the instruction. The operand resides in memory and its address is given directly by the address field of the instruction.
- Only one reference to memory is required to fetch the operand. It is also called as Absolute Addressing Mode.

- Example :-

(i) ADD X, will increment the value stored in the accumulator by the value stored at memory location X.

$$AC \leftarrow AC + [X]$$

(ii) ADD R1, 4000 - In this the 4000 is effective address of operand.

Indirect Addressing Mode :

- In this mode the address field of the instruction give the address where the effective address is stored in memory.
- Control fetches the instruction from memory and uses its address part to access memory again to read the effective address. The effective address in this mode is obtained from the following computational :

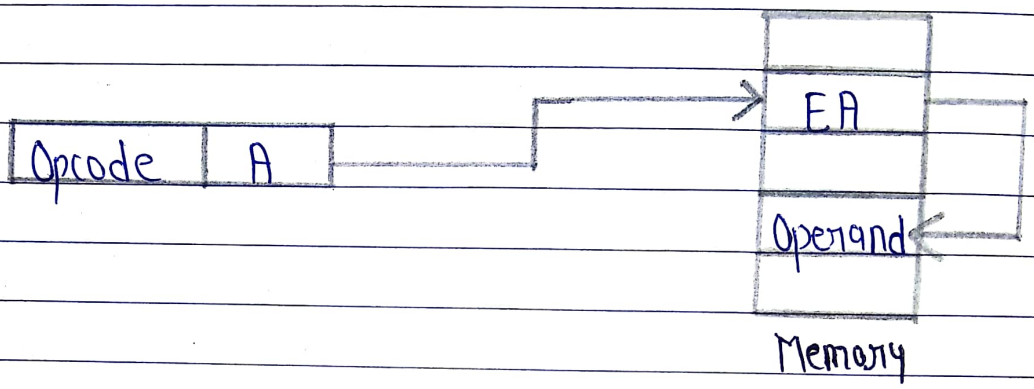
- **Effective Address = Address Part of Instruction + Content of CPU Register**

Tanishq Chauhan
2103184
CS-B

Chauhan

Date
Page 08

- Two references to memory are required to fetch the operand.



- Example :-

  ADD X will increment the value stored in the accumulator by the value stored at memory location specified by X.

$$AC \leftarrow AC + [[X]]$$

- In immediate, effective address is address of instruction operand.
- In register indirect, effective address is stored in register.
- In direct mode, effective address is memory address specified in operand.
- In indirect mode, effective address is stored either in register or memory as specified operand.