Computer Architecture & Organization

Compiled by:- Er. Nagendra Karn

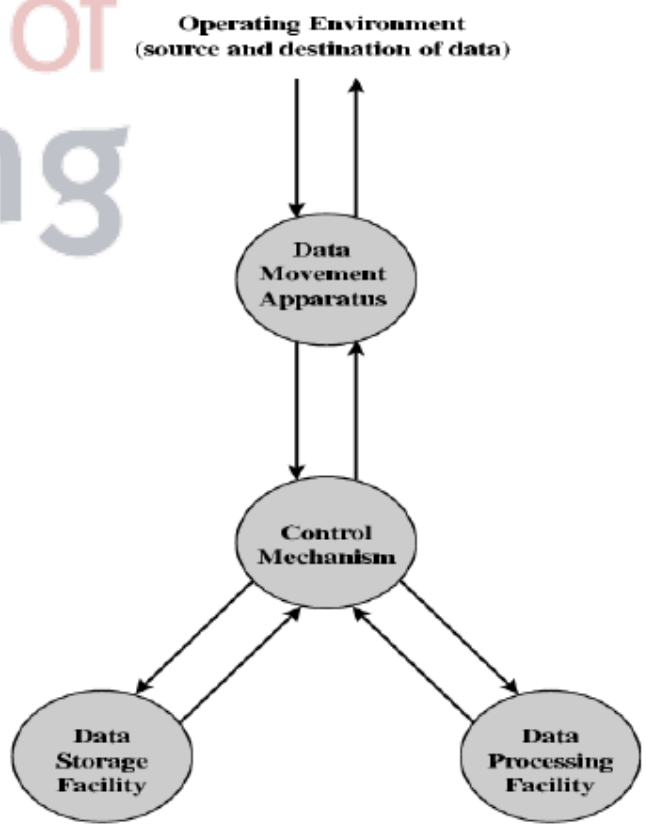
Computer Architecture and Organization:-

- Computer Architecture refers to those attributes of the system that have direct impact on the logical execution of the program. Architecture is those attributes that are visible to the programmer. Examples-
  - The instruction set
  - The number of bits used to represent various data types
  - I/O mechanism
  - Memory addressing techniques
- Computer Organization refers to the operational unit and their interconnections that realize the architectural specification. Organization is how features are implemented. Examples-
  - Control signal
  - Interface between computer and peripherals
  - The memory technology being used
- For example, the fact that multiply instruction is available is computer architecture issue. How that multiply is implemented is a computer organization issue.

Structure & Function

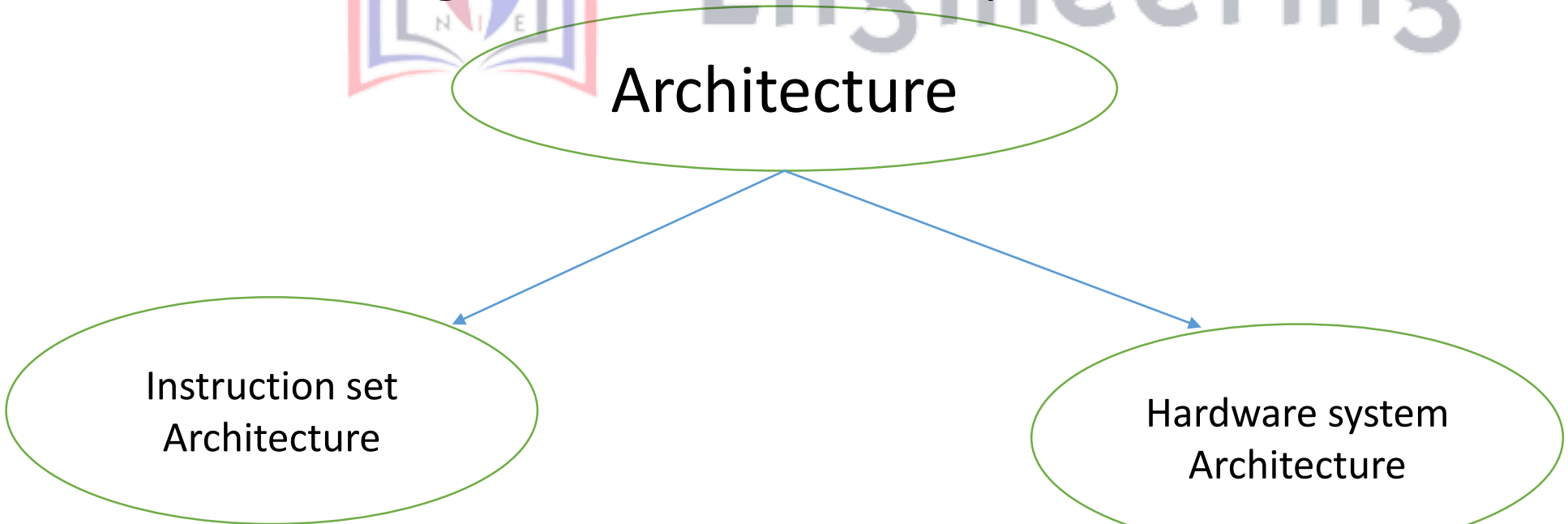
- Structure is the way in which components relate to each other.
- Function is the operation of individual component as part of the structure.
- All computer functions are:-
  - ✓Data Processing- Computer must be able to process
    take wide variety of forms and range of processing.
  - ✓Data storage- Computer stores data either temporari
  - ✓Data movement- computer must be able to move da
    and the

    outside world.
  - ✓Control- there must be control of the above three fur



Computer Architecture

- It is the design of computers including their instruction sets, hardware components and system organization.
- Instruction set architecture (ISA) include the specification that determine how machine language programs will interact with the computer.
- Hardware system architecture (HAS) deals with computer major hardware subsystem like cpu, storage, i/o etc. it includes both the logical design and the data flow organization of the sub systems and determines the efficiency.

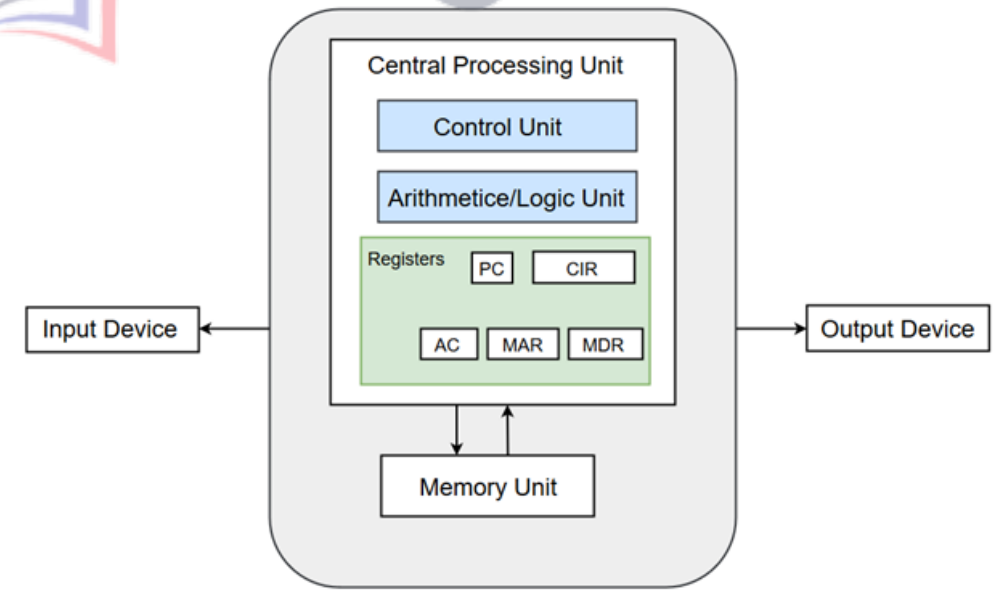


History of Computer Architecture

- The first document of the computer architecture was a correspondence between Charles Babbage and Ada Lovelace that describes the analytical engine.
- The example of other important machines : John Von Neumann and Alan turing.
- Computer Architecture is the art of determining the need of the user of a structure and then designing to meet those needs as effectively as possible with economic status and as well as technological constraints.
- In ancient period, computer architecture was prepared on the paper and then directly build into the final hardware form.
- In today's computer architecture, prototypes were physically built in the form of a transistor logic (TTL) computer such as prototypes of the 6800 and the PA-RISC tested and tweaked before committing to the final hardware form.

Von Neumann Architecture

- The modern computers are based on a stored program concept introduced by John Von Neumann.
- In this architecture program data and instruction data are stored in a same memory and instructions are fetched from memory one at a time in order(serially) and executes program following the fetch-decode-execute cycle.
- It consisted of a control unit, Arithmetic and Logic unit(ALU), Registers and I/Os.



Central Processing Unit

- The part of the computer that performs bulk of data processing operations is called CPU. It can also defined as an electric circuit responsible for executing the instructions of a computer program.
- The major Components of CPU are Arithmetic and logic unit (ALU), Control unit (CU), and Registers.
- The ALU performs required micro-operations for executing the instructions. ALU allows arithmetic (add, subtract etc) and logical (AND, OR, NOT etc) operations to be carried out.
- Control unit of a computer system controls the operations of a component like ALU, Memory, Input Output devices. The control unit consists of a Program counter that contains address of the instruction to be fetched and an instruction register into which instructions are fetched from memory for execution.
- Registers refers to high speed storage area in the cpu. The data processed by the CPU are fetched from the register.

Register :-

- **Memory Address Register** (MAR)- Hold the memory location of data that need to be accessed.
- **Memory Data Register** (MDR) – Hold the data that is being transferred to or from memory.
- **Accumulator** (AC)- where arithmetic and logic results are stored.
- **Program Counter** (PC)- contains the address of the next instruction to be executed.
- **Instruction Register** (IR)- contains the current instruction during processing.

Buses :-

- Buses are the means by which data is transmitted from one part of a computer to another connecting all major internal components to the CPU and Memory.
- **Address Bus** – carries the addresses of the data between the processor and memory.
- **Data Bus-** Carries data between the processor, memory unit and I/O devices
- **Control Bus** – Carries control signal from cpu in order to control and coordinate all the activities within the computer.

Advantage:-

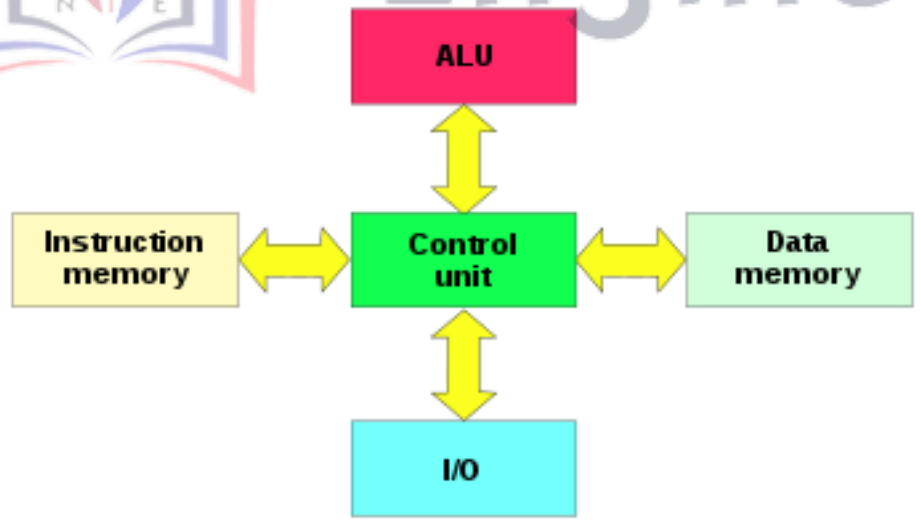
- Control unit retrieves data and instruction in the same manner from one memory. Design development of control unit is simplified cheaper, and faster.
- Organization of memory is done by programmers which allows them to utilize the memory's whole capacity.

Disadvantage:-

- Parallel implementation of program is not allowed due to sequential instruction processing.
- Instruction can only carried out one at a time and sequentially.

Harvard Architecture:-

- It contains separate storage and separate buses (signal path) for instruction and data.
- It was basically developed to overcome the bottleneck of Von Neumann architecture.
- The main advantage of having separate buses for instructions and data is that cpu can access instructions and read/write data at the same time.



Memory Hierarchy

- In the design of computer system, a processor, as well as large amount of memory devices, has been used. The main problem is, these parts are expensive. So the memory organization of the system can be done by memory hierarchy.
- Memory hierarchy is an enhancement of to organize the memory such that it can minimize the access time. The memory hierarchy was developed based on a program behavior known as locality of reference.
- Memory hierarchy design is divided into two main types-

  1. External / secondary Memory- Comprising of magnetic disk, optical disk, magnetic tape i.e. peripheral storage devices which are accessible by the processor via I/O module.

  2. Internal or Primary memory- comprising of main memory & CPU registers. This is directly accessible via processor.

- Based on a speed and use memory in a computer can be divided into five hierarchies. The processor can move from one level to another level based on its requirement.
- The five hierarchies in the memory are registers, cache, main memory, magnetic discs, and magnetic tape. The first three hierarchies are volatile memories where are the last two hierarchies are non volatile.

Increase
Speed
&
Cost per bit

Increase
Size

Processor

Registers

Primary
Cache
L1

Always on the
Processor
chip

Secondary Cache
L2

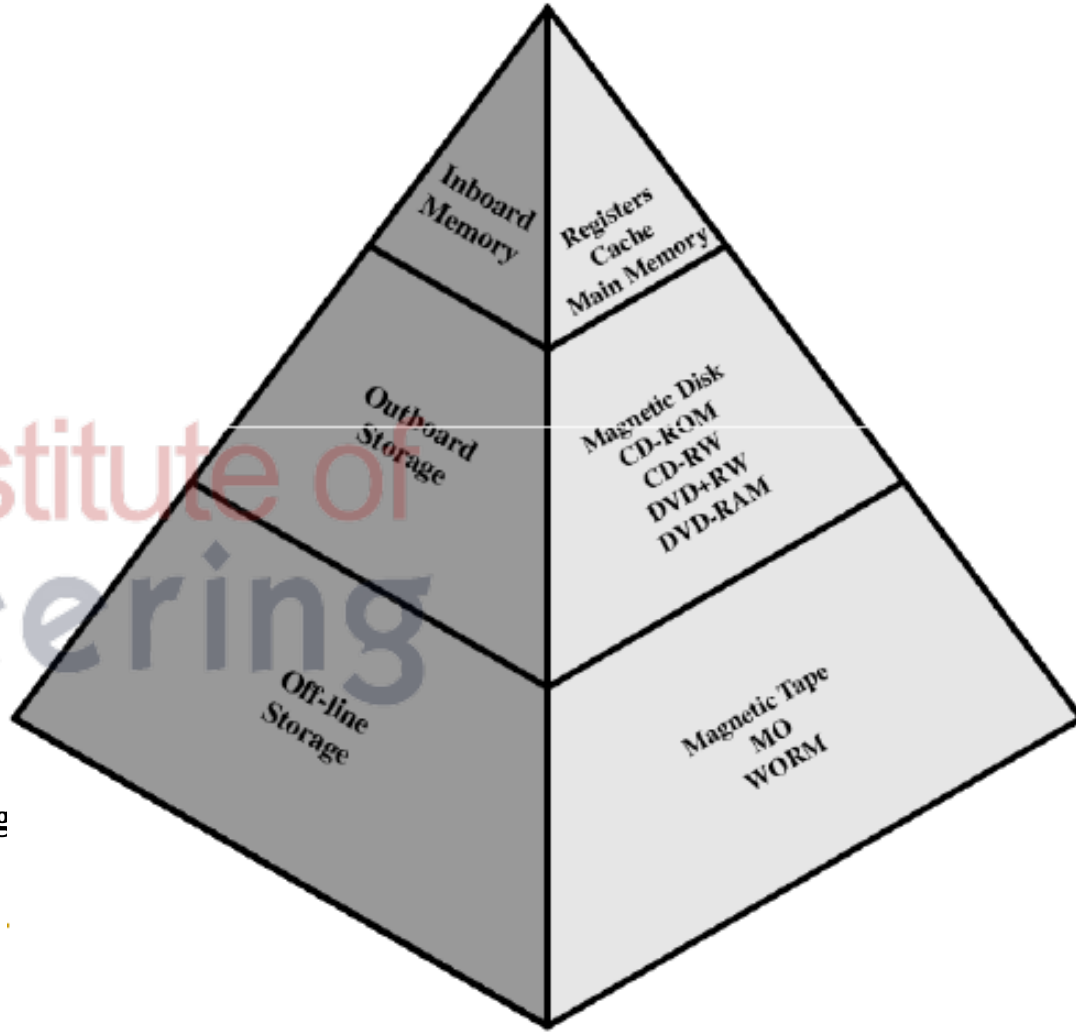
May also be on
the processor chip

Main Memory

Cache usually
SRAM – faster
but more
expensive

Main usually
DRAM – cheap
enough to be larg

Secondary
Storage



Characteristics of Memory Hierarchy:-

1. Performance :- previously designing of computer system was done without memory hierarchy, and the speed gap among the main memory as well as the cpu registers enhances because of the huge disparity in access time, which will cause the lower performance of the system. The enhancement of this was designed in the memory hierarchy model due to system performance increase.
2. Ability :- the ability of memory hierarchy is the total amount of data the memory can store.
3. Access Time :- the access time in a memory hierarchy is the interval of time among the data availability as well as request to read or write. When we move from top to bottom inside the memory hierarchy , then access time will increase.
4. Cost per bit :- When we move from top to bottom inside the memory hierarchy, then cost per bit will decrease.

Memory Hierarchy

- Capacity cost and speed of different types of memory plays a vital role while designing a memory system for a computer. If the memory has larger capacity more applications run smoothly. Moreover for practical system cost should be reasonable.
- There is a trade-off between these three characteristics cost, speed and access time. One cannot achieve all these quantities in same memory module because-
  - ❖If capacity increases access time increases (slower) due to which cost per bit decreases.
  - ❖If access time decreases(faster) capacity decreases and due to which cost per bit increases.
- Memory hierarchy is to obtain highest possible access speed while minimizing the total cost of the memory system.
- The memory unit that directly communicate with cpu is called is called the main memory.
- Devices that provide backup storage is called auxiliary memory.
- The memory hierarchy system consists of all storage devices employed in a computer system from the slow by high capacity auxiliary memory to a relatively faster main memory, to an even smaller and faster cache memory.
- The main memory occupies a central position by being able to communicate directly with the cpu and with auxilary memory devices through an I/O processor.

Memory Hierarchy Design:-

- The memory hierarchy in the computer includes the following:-

1. Registers :- Register is the Static RAM in the processor of the computer which is used for holding the data word which is 64 bits or 128 bits. The program counter register is most important which is found in all processors.
2. Cache Memory :- cache memory can also be found in the processor, however it may be another IC which is separated into levels. The cache holds the chunk of data which are frequently used from main memory. Present multicore processor will be having three, 2 levels for each one core and one level is shared.
3. Main memory :- it directly communicates with cpu and with the auxillary devices through an I/O processor. Main memory is less expensive than cache memory and larger in size. The memory is implemented by using DRAM.
4. Secondary Storage :- Secondary storage devices like magnetic disk are present at level 3. they are used as backup storage. They are cheaper than main memory and larger in size.
5. Tertiary Storage :- tertiary storage devices like magnetic tapes are present at level 4. they are used to store removable files and are cheapest and largest in size.

Internal or Main Memory

- Central unit of a computer system. It is relatively large and fast memory to store programs and data during the computer operation. These memories employ semiconductor integrated circuits. The basic element of semiconductor memory is memory cell.
- The memory cell has three functional terminals which carries electrical signal.
  - ❖The select terminal – it selects the cell.
  - ❖The data terminal – it is used to input data as 0 or 1 and data out or sense terminal is used for the output of the cell's state.
  - ❖Control terminal – it controls the function. i.e. it indicates read and write.

(a) Write (b) Read

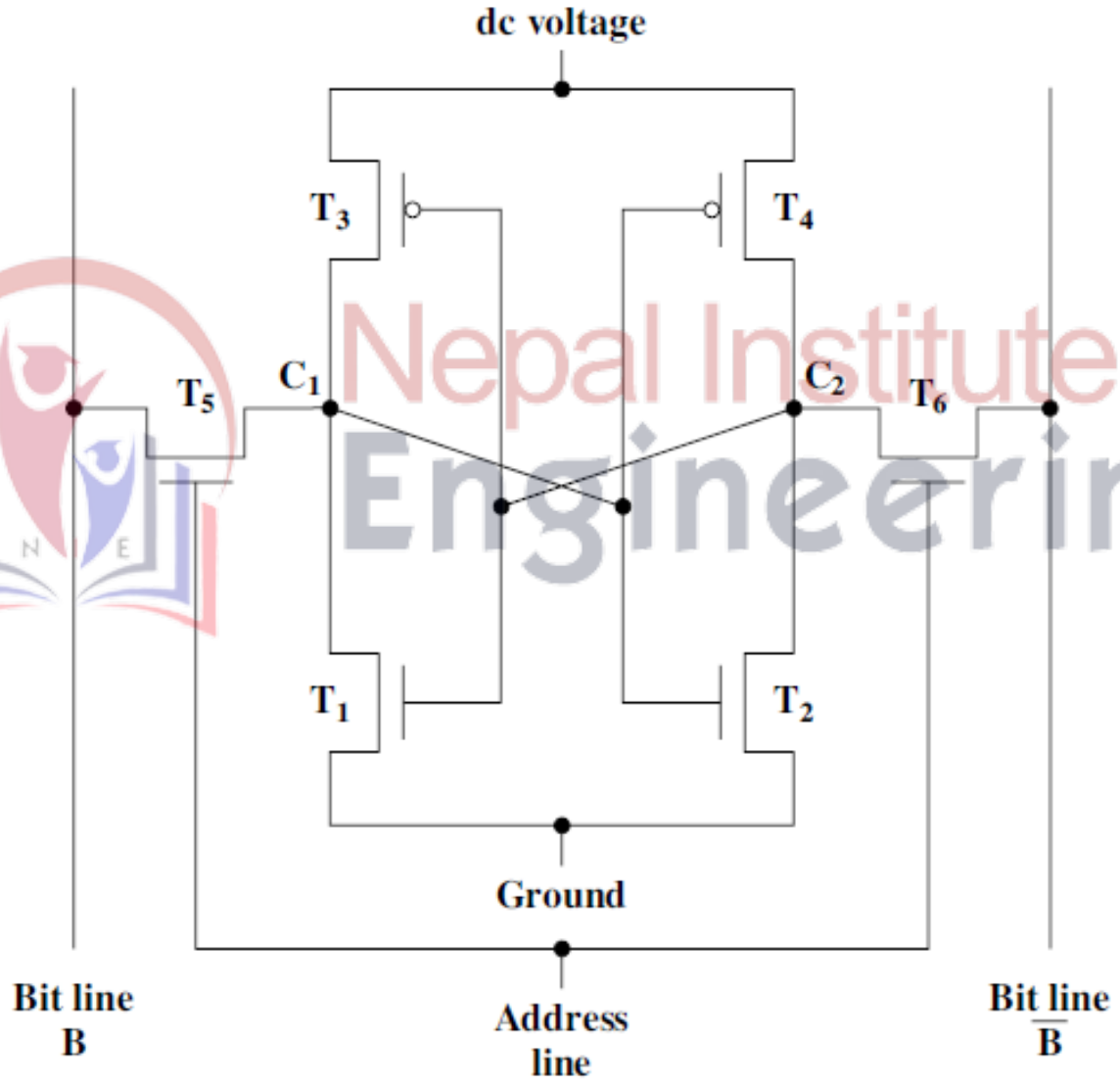
Random Access Memory(RAM)

- Memory cells can be accessed for information transfer from any desired random location.
- The process of locating a word in a memory is the same and requires an equal amount of time no matter where the cells are located physically in memory thus named “Random Access”.
- RAM are available in two possible operating modes, static and Dynamic.

Static Random Access Memory(SRAM)

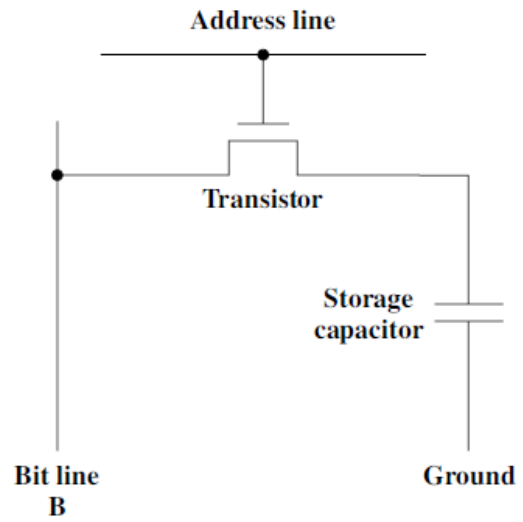
- The static RAM consists of flip-flop that stores binary information and this stored information remains valid as long as power is applied to the unit.
- Bit stored as on/off switches. No refreshing needed when powered.
- More complex construction and more expensive.
- Faster than DRAM.
- Used in Cache.

(Cont...)Static RAM



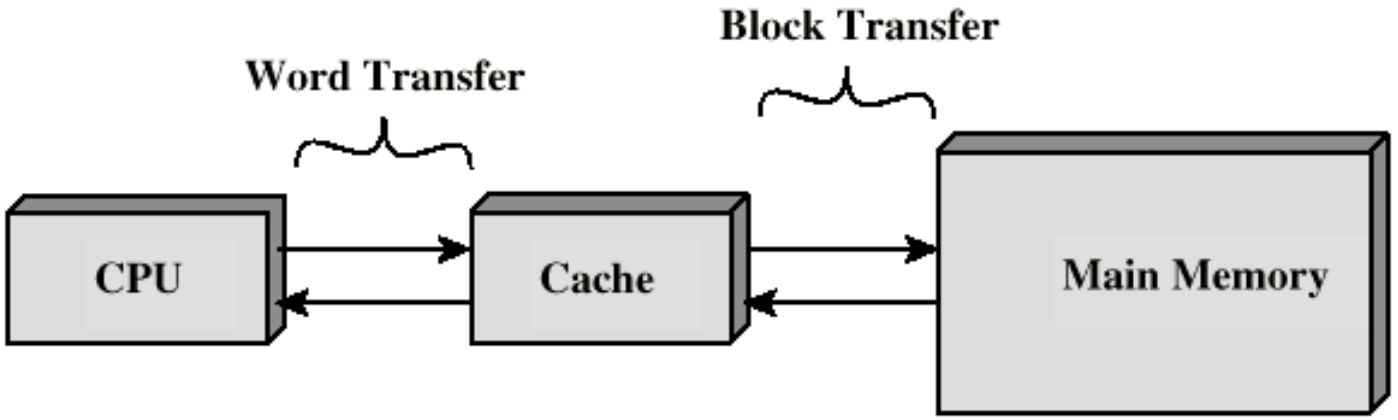
Dynamic RAM

- The dynamic RAM stores the binary information in the form of electrical charges and capacitor is used for this puropose.
- Since the charge stored in a capacitor discharges with time capacitor must be periodically recharged and which is also called refreshing memory.
- Simple construction, highly dense, less expensive and slower compared to SRAM.
- Used as main memory.



Cache Memory

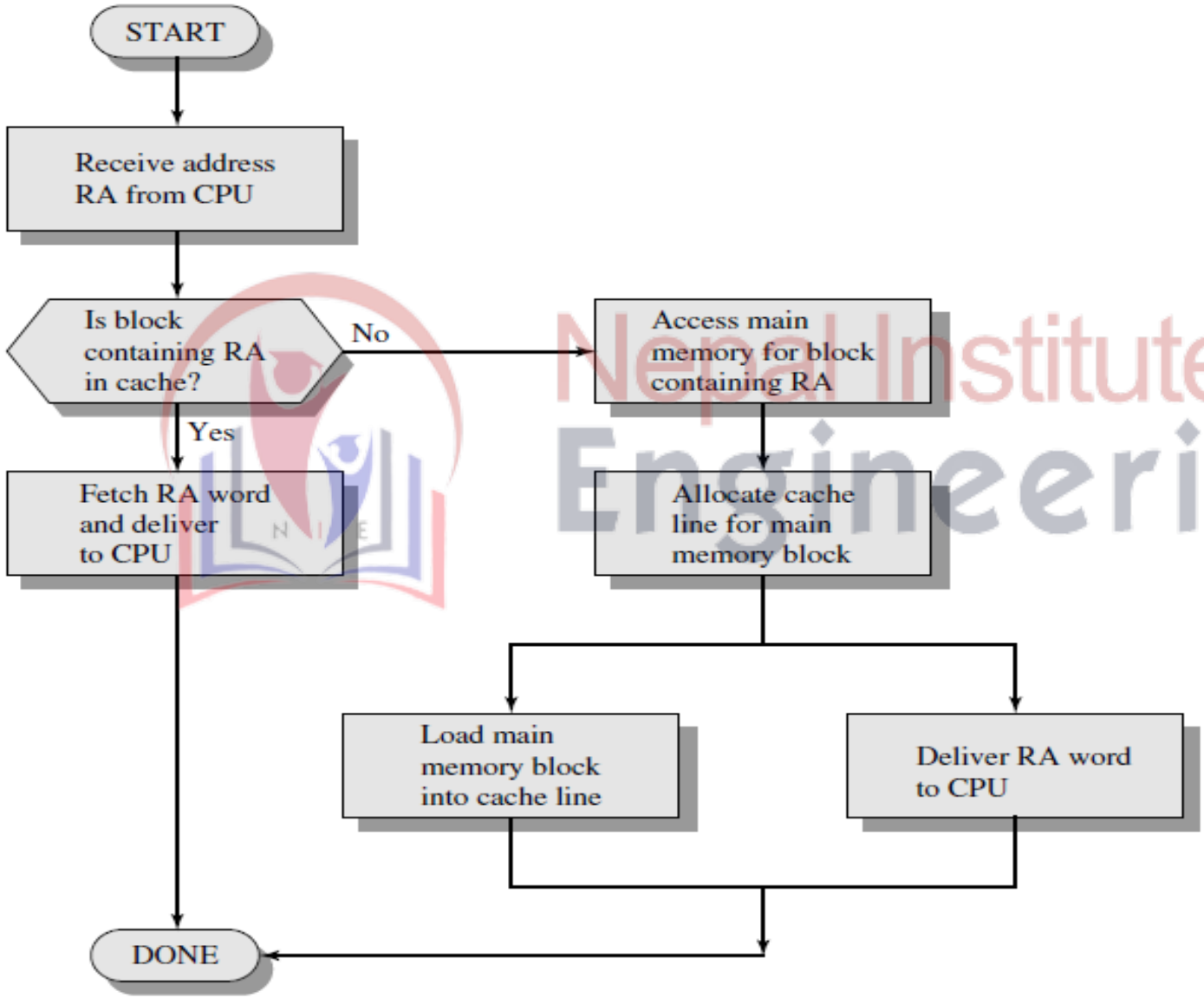
- The data or content of the main memory that are used frequently by the cpu are stored in the cache so that processor can easily access the data in shorter time.
- Whenever cpu needs to access memory it first checks the cache memory, then the cpu moves into main memory.
- Cache memory is placed between the cpu and the main memory. The block diagram for a cache memory can be represented as :



Cache Operation :-

- When the cpu needs to access memory, the cache is examined, if the word is found in the cache, it is read from the main memory.
- If the word addressed by the cpu is not found in the cache, the main memory is accessed to read the word.
- A block of word one just accessed is then transferred from main memory to cache memory.
- The performance of cache is frequently measured in terms of quantity called **Hit ratio**.
- When the cpu refers to memory and find the word in the cache, it is said to produce a hit.
- If the word is not found in the cache, it is in main memory and it counts as a miss.
- The ratio of number of hits divided by the total number of cpu references to memory is the hit ratio.

$$\text{Hit Ratio} = \frac{Number\ of\ Hit}{Tota\ number\ of\ CPU\ refrence}$$

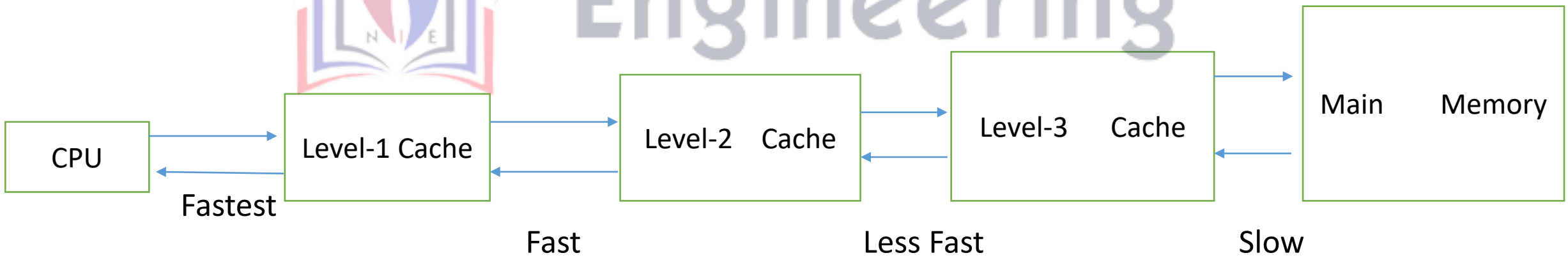


Locality of Reference :-

- Also known as principle of locality/ reference of locality.
- Two types :-

  1. Temporal :- Any memory location that has been accessed now may be accessed again in near future.

  2. Spatial :- If any particular memory location has been accessed then there is possibility that near by memory location may also be accessed.



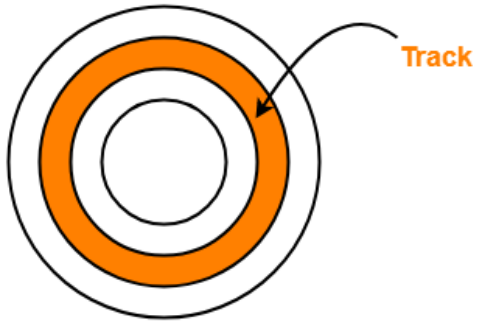
Hard disk

- It is a storage device that used magnetically platters to store data, instructions and information.
- It is a storage device that is used to write rewrite and access the data. It uses a magnetization process.
- Organization :-

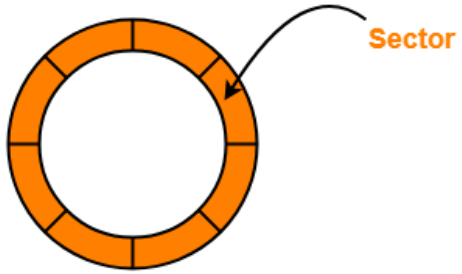
1.The entire disk is divided into platters.

2.Each platters consists of a concentric circles called as tracks.

3.These tracks are further divided into sectors which are the smallest divisions in the disk.

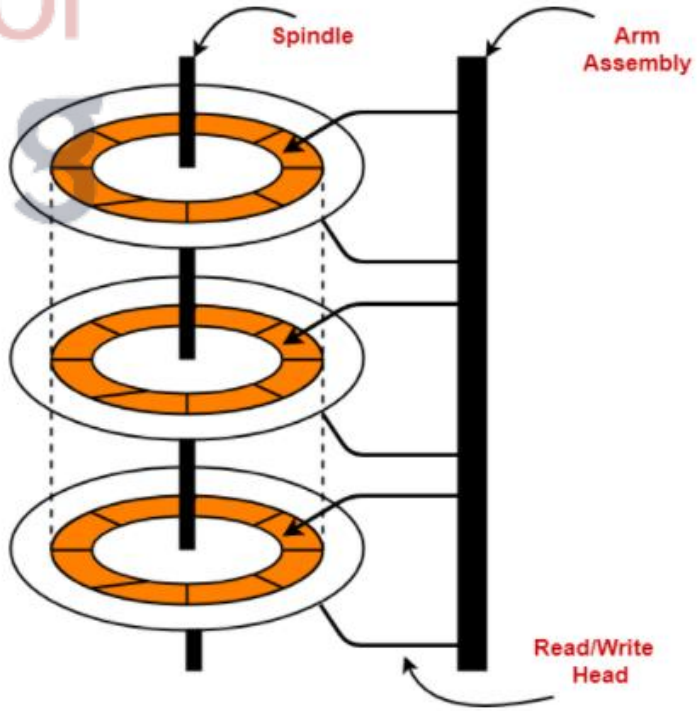


Disk divided into tracks



Track divided into sectors

- A cylinder is formed by combining the tracks at a given radius of a disk pack.
- There exist a mechanical arm called read/write head.
- It is used to read from and write to the disk.
- Head has to reach at a particular track and then wait for the rotation of the platter.
- The rotation causes the required sector of the track to come under the head.
- Each platter has 2 surfaces –top and bottom and both the surfaces are used to store the data.
- Each surface has its own read/write head.



Disk performance parameters :-

- The time taken by the disk to complete an I/O request is called as disk access time.
- Disk Access time = Seek time + Rotation latency + transfer time.

1. Seek time :- the time taken by the read/write head to reach the desired track is called seek time. The lower the seek time, faster the I/O operation.
2. Rotation latency :- the time taken by the desired sector to come under the read/write head is called as rotation latency.
3. Transfer time = $\frac{\text{data to be transfer}}{transfer\ rate}$;
4. Transfer rate = No. of heads(surface) * capacity of one track * No of rotation in

   one second
5. Disk size = Total number of surface * Number of tracks per surface * number of sectors per track * storage capacity of one sector.

Instruction Cycle

- Time required to execute and fetch an entire instruction is called instruction cycle. A program residing in the memory unit of the computer consists of a sequence of instrucions.
- The program is executed in the computer by going through the cycle for each instruction. Each instructions cycle in turn subdivided into sequence of a sub cycle. Each instrucions cycle consists of following phases :-
  1. Fetches instructions from memory.
  2. Decode the instruction.
  3. Read the effective address from memory.
  4. Execute the instruction.

Instruction Codes

- An instruction code is a group of bits that instruct computer to perform a specific operation.
- A computer usually has a variety of instruction code format. It is the function of the control unit within the cpu to interpret each instruction code and provide the necessary control functions needed to process the instruction.
- The bits of the instruction are divided into groups called fields. The most common fields in the instruction formats are –
  - An **operation code** field that specifies the operation to be performed.
  - An **Address field** that designates memory address or a processor register.
  - A **Mode** field that specifies the way the operand or effective address is determined.

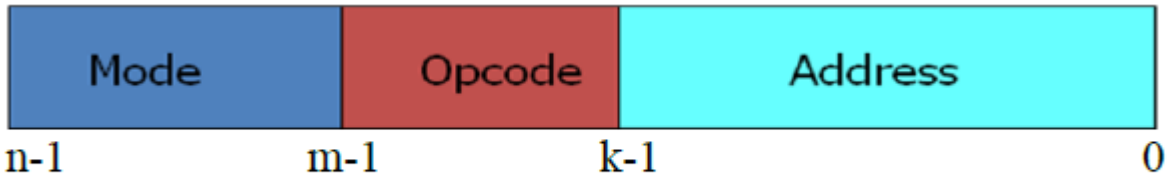


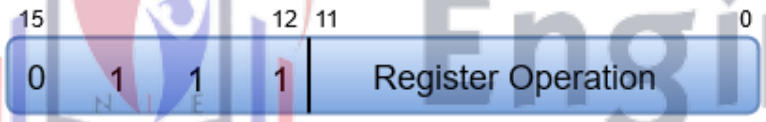
Fig: Instruction format with mode field

- The Instructions are stored in a computer memory in the same manner that data is stored.
- The control unit interprets these instructions and uses these operations code to determine the sequence of micro operations that must be performed to execute the instruction.
- The operands are specified by indicating the registers and/or memory locations in which they are stored.
- The simplest design is to have one processor register and two fields in the instruction, one for the opcode and one for the operand.
- Any operation that does not need memory operand frees the other bits to be used for the purposes, such as specifying different operations.
- A basic computer has three instruction code format –
    1. Memory – reference instruction
    2. Register – reference instruction
    3. Input-output instruction

- Memory Reference Instruction – In memory reference instruction, 12 bit of memory is used to specify a address and 1 bit to specify the addressing mode ‘I’. If I=0, it represents direct and if I=1 then it represents indirect addressing mode.

(Opcode = 000 through 110)

- Register Reference Instruction – the register reference instruction are represented by the opcode 111 with a 0 in the left most bit of the instruction.



(Opcode = 111, I = 0)

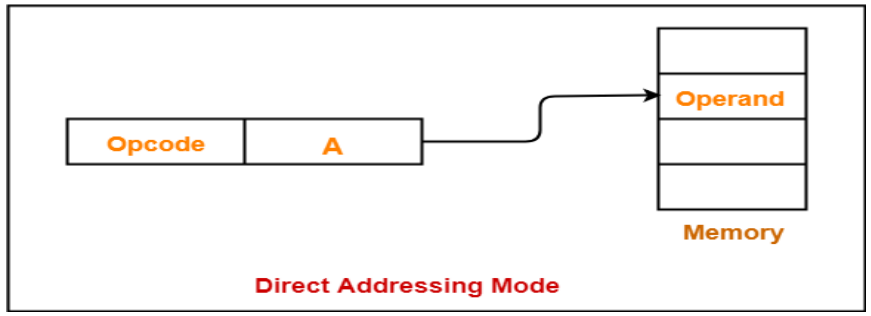
- Input- output Instruction – just like register reference instruction, an input-output instruction does not need a reference to memory and is recognized by the opcode 111 with 1 in the left most bit of the instruction. The remaining 12 bit is used to specify the type of the input output operation or test performed.

(Opcode = 111, I = 1)

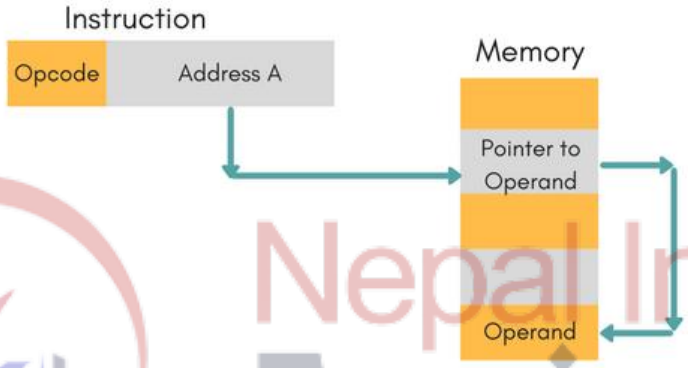
- The three operation code bits in positions 12 through 14 should be equal to 111. otherwise the instruction is memory reference type, and the bit is in position 15 is taken as the addressing mode ,I.
- When the three operation codes bits are equal to 111, control unit inspects the bit in position 15. of the bit is 0, the instruction is a register reference type otherwise the instruction is a input-output reference type.

Addressing Modes:-

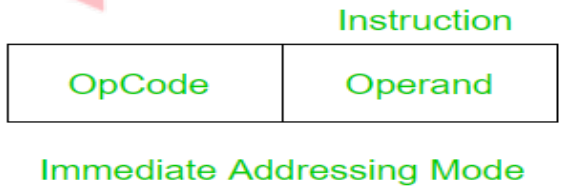
- There are four different types of operands that can appear in an instruction :
- **Direct operand-** an operand stores in a register or in the memory location specified.



- **Indirect operand-** an operand whose address is stored in the register or in the memory location specified.



- Immediate operand- an operand whose value is specified in the instruction. e.g. MVI B,50H



- Implied operand- the operands are specified implicitly in the definition of the instruction. E.g. CMA (complement Accumulator), the operand in the accumulator register is implied in the definition of the instruction.

Instruction-set Completeness

- A set of Instruction set is said to be complete if the computer includes a sufficient number of instructions in each of the following categories-
  - ❖Arithmetic, logic and shift instructions
  - ❖A set of instructions for moving information to and from memory and processor registers.
  - ❖Instruction which controls the program together with instruction that check status conditions
  - ❖Input and output instructions.
- Arithmetic, logical and shift instructions provide computational capabilities for processing the type of data the user may wish to employ.
- A huge amount of binary information is stored in a memory unit but all computations are done in a processor registers. Therefore one must process the capability of moving information between these two units.
- Program control instructions such as branch instructions are used change the sequence in which program is executed.

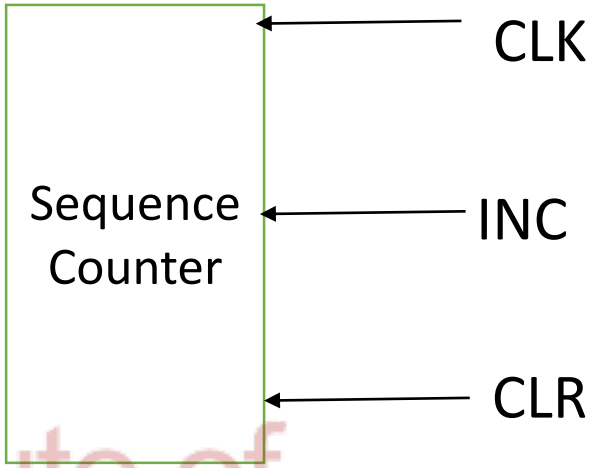
➢Input and output instructions acts as an interface between the computer and user. Program and data must be transferred into memory, and the results of computations must be transferred back to the user.

Basic Memory Reference Instructions

Symbol	Hexadecimal code		Descriptions
	I=0	I=1	
AND	0xxx	8xxx	AND mem. Word to AC
ADD	1xxx	9xxx	ADD mem. Word to AC
LDA	2xxx	Axxx	Load mem. Word to AC
STA	3xxx	Bxxx	Store content of memory word to AC
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address.
ISZ	6xxx	Exxx	Increment and skip if zero.

Memory Reference Instructions :-

1. AND :- D_0T_4: DR←m[AR]
   D_0T_5: AC←AC ^ DR , SC←0
2. ADD :- D_1T_4: DR←m[AR]
   D_1T_5: AC←AC + DR , E←C_{out} , SC←0
3. LDA :- D_2T_4: DR←m[AR]
   D_2T_5: AC← DR , SC←0
4. STA :- D_3T_4: m[AR]←AC , SC←0,
5. BUN :- D_4T_4: PC←AR, SC←0
6. BSA :- D_5T_4: m[AR]←PC , AR←AR+1
   D_5T_4: PC←AR, SC←0
7. ISZ :- D_6T_4 : DR ←m[AR]
   D_6T_4 : DR ←DR+1 if DR=0, then PC ← PC+1, SC ←0;



Basic Register-Reference Instruction

Symbol	Hex Code	Description
CLA	7800	Clear AC
CLE	7400	Clear E
CMA	7200	Complement AC
CME	7100	Complement E
CIR	7080	Circulate right AC & E.
CIL	7040	Circulate left AC & E
INC	7020	Increment AC
SPA	7010	Skip next instruction if AC is positive
SNA	7008	Skip next instruction if AC is negative.
SZA	7004	Skip next instruction if AC is zero
SZE	7002	Skip next instruction if E is zero.
HLT	7001	Halt computer

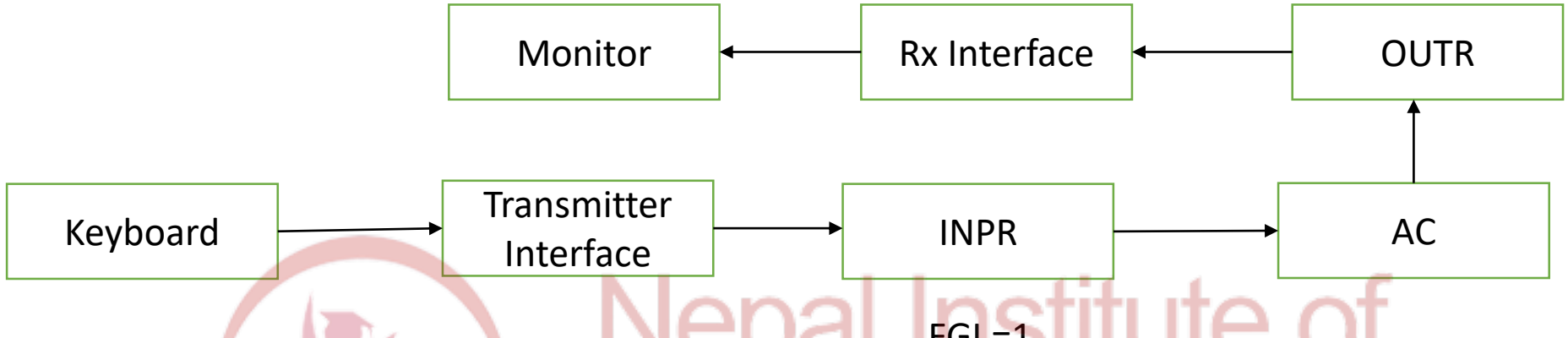
Register Reference Instruction :-

- CLA : rB_{11} : AC←0
- CLE : rB_{10} : E←0
- CMA : rB_9 : AC←AC’
- CME : rB_8 : E←E’
- CIR : rB_7 : AC←shr AC, E ←AC(0), AC(15) ←E
- CIL : rB_6 : AC←shl AC, E ←AC(15), AC(0) ←E
- INC : rB_5 : AC←AC+1
- SPA : rB_4 : if(AC(15)=0) then PC←PC+1
- SNA : rB_3 : if(AC(15)=1) then PC←PC+1
- SZA : rB_2 : if(AC=0) then PC←PC+1
- SZE : rB_1 : if(E=0) then PC←PC+1
- HLT : rB_0 : S←0

Basic input output instructions:-

Symbol	Hex Code	Description
INP	F800	Input character to AC
OUT	F400	Output character from AC
SKI	F200	Skip to input flag
SKO	F100	Skip on output flag
ION	F080	Interrupt on
IOF	F040	Interrupt off

I/O Reference Instructions :-



- INP : AC(0-7) ←INPR, FGI ←0
- OUT : OUTR←AC(0-7), FGo ←0
- SKI : If (FGI=1) then PC ←PC+1
- SKO : if (FGO=1) then PC ←PC+1
- ION : IEN ←1
- IOF : IEN ← 0

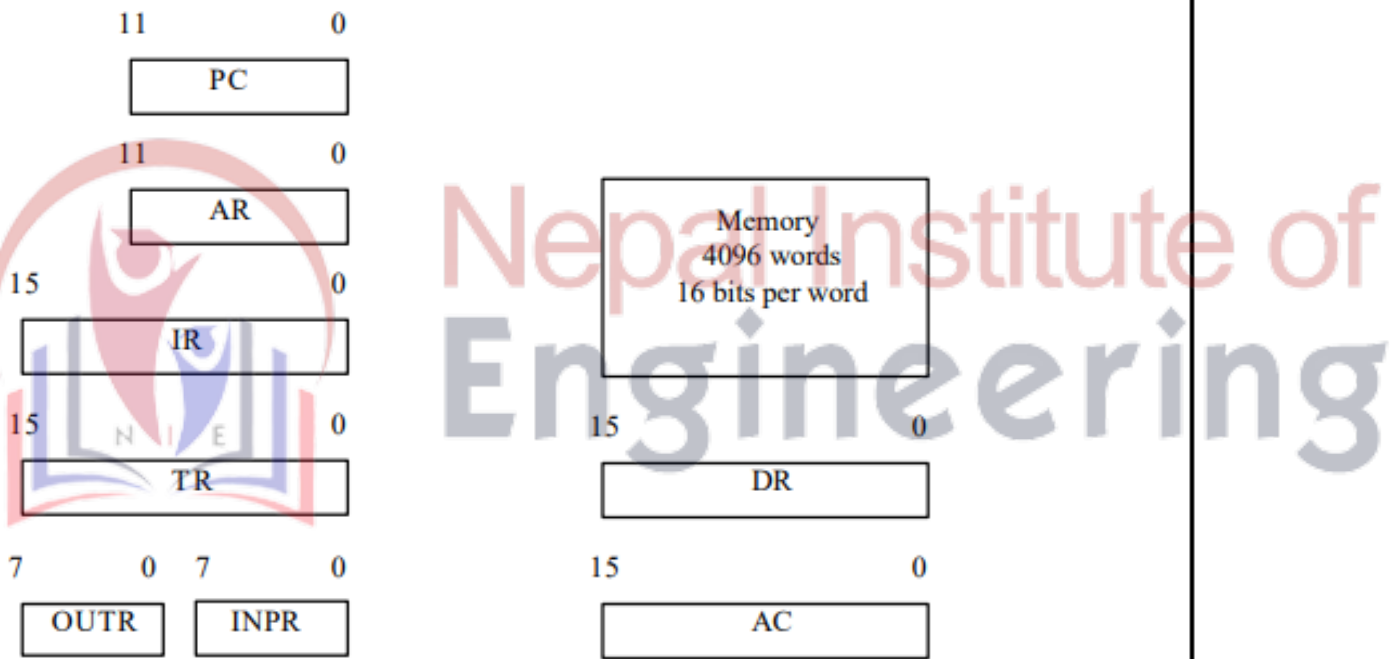
Registers:-

- Register is a type of computer memory used to quickly accept, store and transfer data and instructions that are being used immediately by the cpu.
- The register may hold an instruction, storage address, or any data (such as bit sequence or any individual character).
- Computer need registers for manipulating data and a register for holding a memory address. The register holding the memory location is used to calculate the address of the next instruction after the execution of current instruction is completed.

List of Registers for the Basic Computer

Register Symbol	# of Bits	Register Name	Function
DR	16	Data Register	Holds memory operand
AR	12	Address Register	Holds mem. address
AC	16	Accumulator	Processor Reg.
IR	16	Instruction Register	Holds instruction code
PC	12	Program Counter	Holds instruction address
TR	16	Temporary Register	Holds temporary data
INPR	8	Input Register	Holds input character
OUTR	8	Output Register	Holds output character

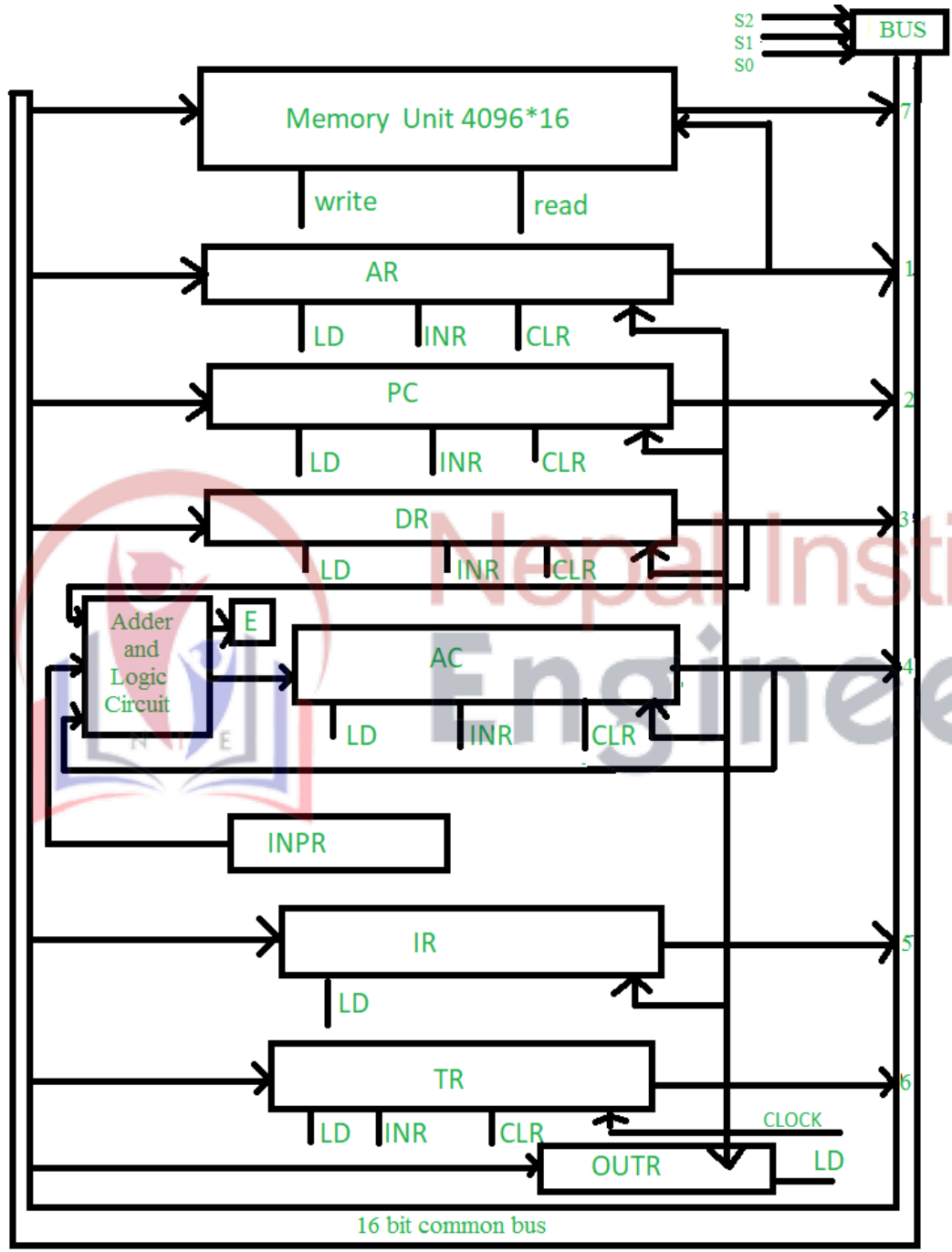
Basic Computer Registers and Memory



- The memory unit has the capacity of 4096 words, and each words contains 16 bits.
- The Data Register (DR) contains 16 bits which holds the operand read from the memory location.
- The memory address register (MAR) contains 12 bits which holds the address for the memory location.
- The program counter (PC) also contains 12 bits which hold the address of the next instruction to be read from the memory after current instruction is executed.
- The Accumulator (AC) is a general purpose processing register.
- The instruction read from the memory is placed in the Instruction Register.
- The Temporary register is used for holding the temporary data during the processing.
- The Input registers (IR) holds the input character given by the user.
- The output registers (OR) holds the output after processing the input data.

Common BUS System

- To avoid excessive wiring, memory and all the registers are connected via a common Bus.
- The specific output that is selected for the Bus is determined by $S_2S_1S_0$.
- The Register whose LD(Load) is enable receives the data from the bus.
- Register Can be incremented by setting the INR control input and can be cleared by setting the CLR control input.
- The Accumulator input must come via Adder and Logic circuit. This allows the accumulator and data register to swap data simultaneously.
- The address of any memory location being accessed must be loaded in the Address Register.
- All except Input Register (INPR) and Adder are connected to clock pulse and INPR is also not connected to Bus as input is given through input device.


S2
S1
S0
BUS
Memory Unit 4096*16
write
read
7
AR
LD
INR
CLR
1
PC
LD
INR
CLR
2
DR
LD
INR
CLR
3
Adder and Logic Circuit
E
AC
LD
INR
CLR
4
INPR
IR
LD
5
TR
LD
INR
CLR
6
CLOCK
OUTR
LD
16 bit common bus

Timing and Control

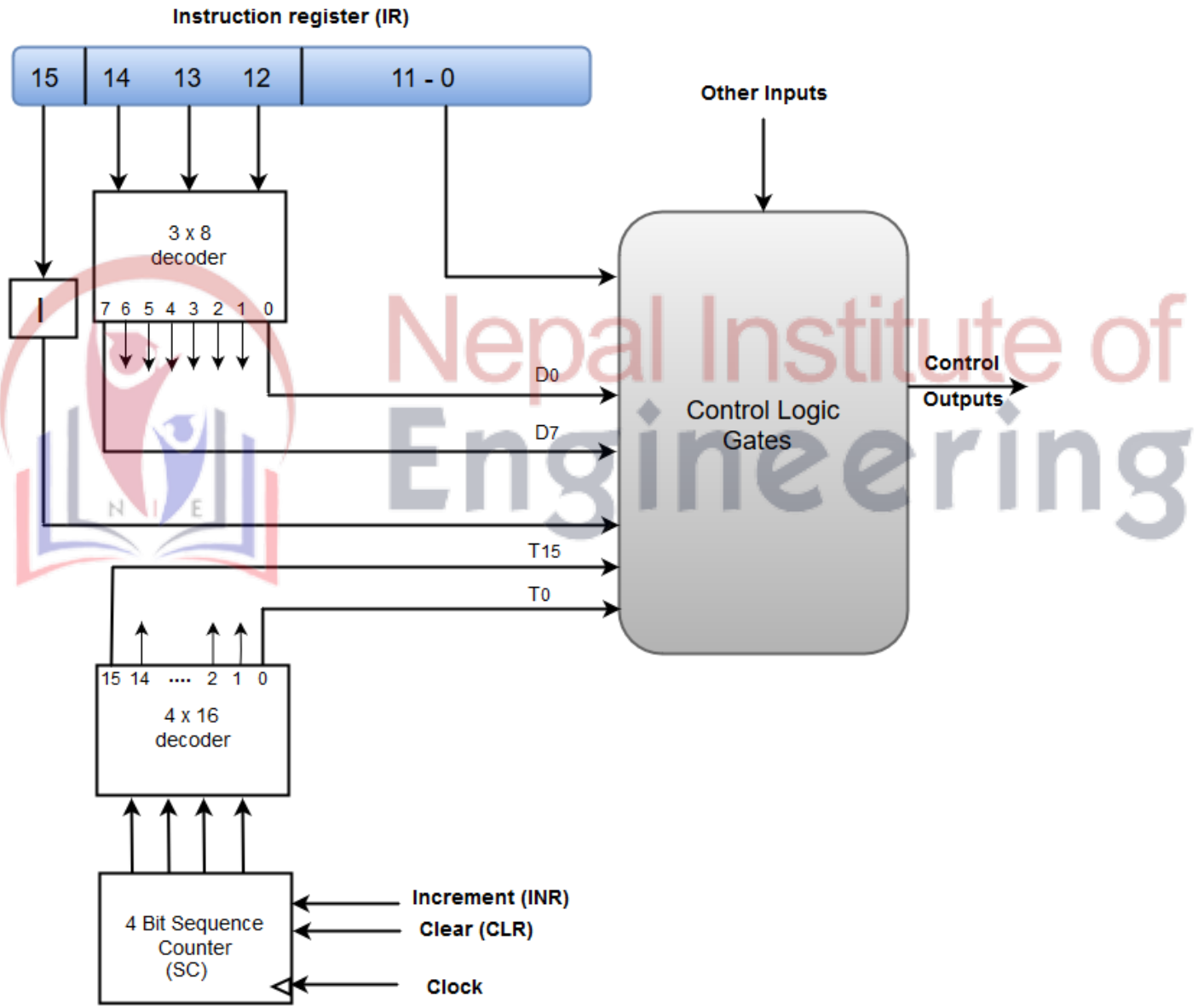
- The timing for all register in a basic computer is controlled by a master clock generator. The clock pulses are applied to all flip-flops and the register in the system, including the flip-flops and register in the control unit.
- The clock pulses do not changes the state of the register unless the register is enabled by a control signal.
- The control signals are generated in the control unit and provide control inputs for the multiplexers in the common bus, control inputs in processor register and micro operations for the accumulator.
- There are two major types of control organization:-
    1. Hardwired control unit.
    2. Microprogrammed control unit.

- **In hardwired organization**, the control logic is implemented with gates, flip-flops decoders and other logical circuits. It has the advantage that it can be optimized to produce a fast mode of operation.
- In Microprogrammed organization, the control information is stored in control memory. The control memory is programmed to initiate the required sequence of micro operations.
- A hardwired control requires changes in the wiring among the various component if the design has to be modified or changed.
- In microprogrammed control, any required changes or modifications can be done by updating the microprogram in the control memory.
- It consists of a two decoders,
    1. A sequence counter, and
    2. A number of logic gates

Timing Signal

- Timing signals are generated by the sequence counter (SC), which receives as inputs the clock pulse, increment and clear.
- The sequence counter's outputs are decoded into 16 timing signals T_o through T_{15} , which are used to control the sequence of operations.
- The statement D_3T_4 :SC ← 0 reset the sequence counter to zero; the next timing signal is T_0

Control Unit of a Basic Computer:



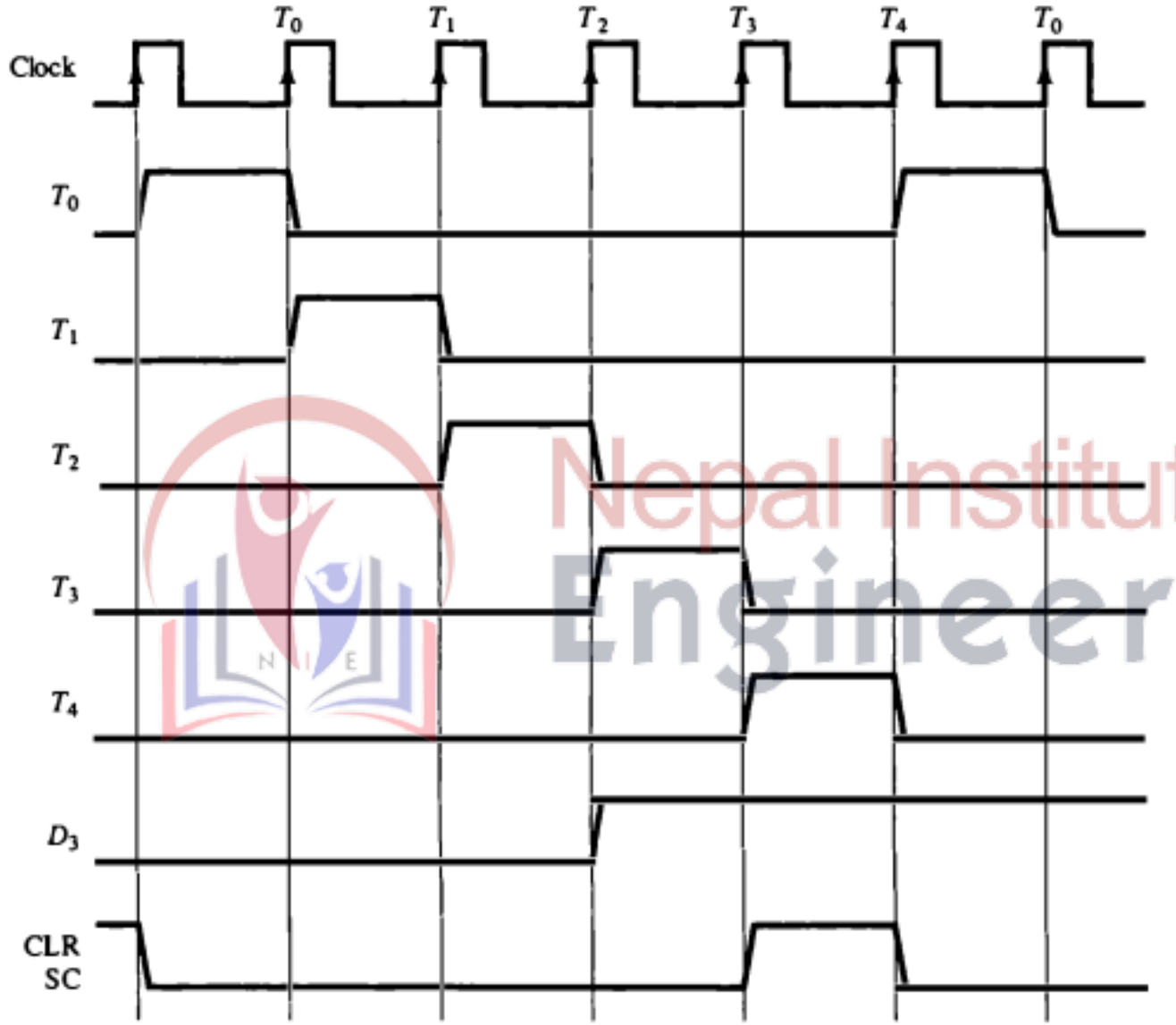


Figure 5-7 Example of control timing signals.