1. **Explain the role of John von Neumann in development of computer.**

→ENIAC designed and constructed at the University of Pennsylvania, was the world's first general purpose electronic digital computer. In the early days of computing, one instruction at a time of a program was executed, with a person (operator) setting up each instruction and also initiating the execution of each instruction by switching some **switches.**

In 1946, von Neumann and his colleagues began the design of a new **stored program computer**, referred to as the IAS computer. The IAS computer is the prototype of all general-purpose computers. The architecture consists of :

• A **main memory**, which stores both data and instructions.
• An **arithmetic and logic unit (ALU)** capable of operating on binary data.
• A **control unit**, which interprets the instructions in memory and causes them to be executed.
• **Input/output (I/O)** equipment operated by the control unit

With some exceptions, all of today's computers have this same general structure and function and are referred as **von Neumann machines.** Neumann Architecture is based on **Stored program concept .A program could be represented in a form suitable for storing in memory with the data stored in same memory. During a execution of a program the stored instruction can be fetched from memory and a program could be set or altered by setting the values of a portion of memory. The fetched instruction can be decoded to set up the necessary data paths and generate the control signals.** This is called **stored program concept**.

Von Neumann computer Organization was a revolutionary concept and held the center stage of computer design for next several decades. The computer organization was based on the von Neumann styles for several decades. Execution of program became faster as manual entry of instruction as well as setting the data and control path or function unit was avoided. The concept of instruction cycle was also raised. One of the short coming of Von Neumann computing arise due to the fact that a single connection exist between processor and memory .That is , at a time only one memory access can occur .

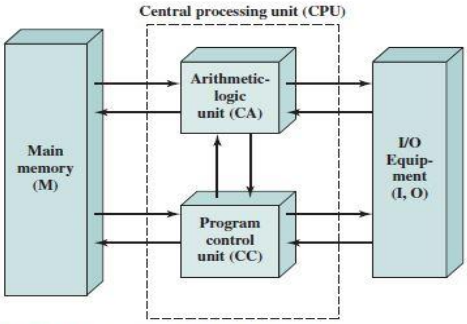
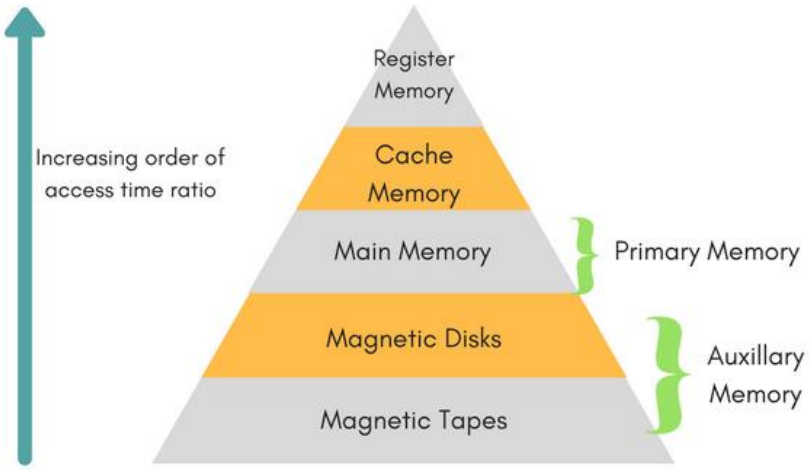


Figure 2.1 Structure of the IAS Computer

2. Explain the memory hierarchy with diagram.



- The total memory capacity of a computer can be visualized as being a hierarchy of component. Memory hierarchy consists of all the storage devices used in a computer system from the slow but high capacity auxiliary memory to the relative faster main memory to a even smaller and faster cache memory accessible to the high speed processing logic. Design constraints on a computer's memory can be summed up by three questions:

➢ How much, how fast, how expensive

There is a relation among capacity, access time, and cost

- Faster access time, greater cost per bit
- Greater capacity, smaller cost per bit
- Greater capacity, slower access time

As one goes down the hierarchy, the following occur :(*show in figure with arrow)*
a. Decreasing cost per bit
b. Increasing capacity
c. Increasing access time
d. Decreasing frequency of access of the memory by the processor
The memory in a computer can be divided into five hierarchies based on the speed as well as use. The processor can move from one level to another based on its requirements. The five hierarchies in the memory are registers, cache, main memory, magnetic discs, and magnetic tapes. The first three hierarchies are volatile memories which mean when there is no power, and then automatically they lose their stored data. Whereas the last two hierarchies are not volatile which means they store the data permanently.

The memory hierarchy in computers mainly includes the following.

Registers:
This is the fastest memory as it remains inside the processor. It is usually used to hold the temporary data .It is also called as processor memory. The Resistors memory like accumulator, Program counter, status word register falls in this category.

Cache Memory:

Cache memory is memory that a computer microprocessor can access more quickly than it can access regular random access memory. It contains the copy of main memory. Cache Memory may be implemented either internal to processor or external .Internal is called as L1 and External is called as L2.

Main Memory

The main memory in the computer is nothing but, the memory unit in the CPU that communicates directly. It is the main storage unit of the computer. This memory is fast as well as large memory used for storing the data and instruction throughout the operations of the computer. This memory is made up of RAM as well as ROM.

Magnetic Disks

A magnetic disk is a storage device that uses a magnetization process to write, rewrite and access data. It is covered with a magnetic coating and stores data in the form of tracks, spots and sectors. Hard disks, zip disks and floppy disks are common examples of magnetic disks.

Magnetic Tape

A magnetic tape drive is a storage device that makes use of magnetic tape as a medium for storage. It uses a long strip of narrow plastic film with tapes of thin magnetizable coating. It is essentially a device which records or perhaps plays back video and audio using magnetic tape, examples of which are tape recorders and video tape recorders.

.

3. Explain the element of cache design

Table 4.2 Elements of Cache Design

Cache Addresses	**Write Policy**
Logical	Write through
Physical	Write back
Cache Size	**Line Size**
Mapping Function	**Number of Caches**
Direct	Single or two level
Associative	Unified or split
Set associative	
Replacement Algorithm	
Least recently used (LRU)	
First in first out (FIFO)	
Least frequently used (LFU)	
Random	

- **Cache Memory**

If the Active portion of the programs and data are placed in a fast and small memory, the average memory access time can be reduced, thus reducing the execution time of the program. Such fast small memory is referred to as cache memory.

The cache contains a copy of portions of main memory. When the processor attempts to read a word of memory, a check is made to determine if the word is in the cache. If so, the word is delivered to the processor. If not, a block of main memory, consisting of some fixed number of words, is read into the cache and then the word is delivered to the processor.

➢ Logical and physical cache :

- Almost all non embedded processors, and many embedded processors, support virtual memory. In essence, virtual memory is a facility that allows programs to address memory from a logical point of view, without regard to the amount of main memory physically available.
- When virtual addresses are used, the system designer may choose to place the cache between the processor and the MMU or between the MMU and main memory. A **logical cache**, also known as a virtual cache, stores data using virtual addresses. The processor accesses the cache directly, without going through the MMU. A **physical cache** stores data using main memory physical addresses.

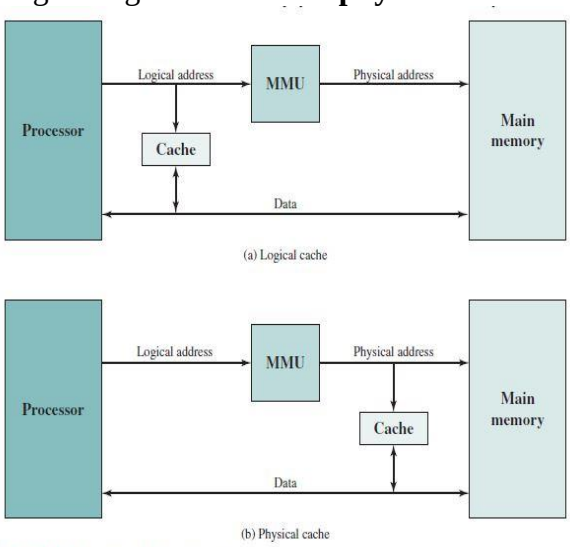


Figure 4.7 Logical and Physical Caches

Cache Size

- We would like the size of the cache to be small enough so that the overall average cost per bit is close to that of main memory alone and large enough so that the overall average access time is close to that of the cache alone.
- There are several other motivations for minimizing cache size. The larger the cache, the larger the number of gates involved in addressing the cache.
- The result is that large caches tend to be slightly slower than small ones—even when built with the same integrated circuit technology and put in the same place on chip and circuit board. The available chip and board area also limits cache size. Because the performance of the cache is very sensitive to the nature of the workload, it is impossible to arrive at a single "optimum" cache size.

Mapping Function

Associative Mapping:

- It stores both the address and content of the memory word. This permits any location in cache to store any word from main memory.
- A CPU address is placed in the argument register and the associative memory is searched for a matching address. If address is found (hit) data is read and sent to the CPU else (miss) the data is searched in main memory.
- To replace the cells of cache we can use FIFO.

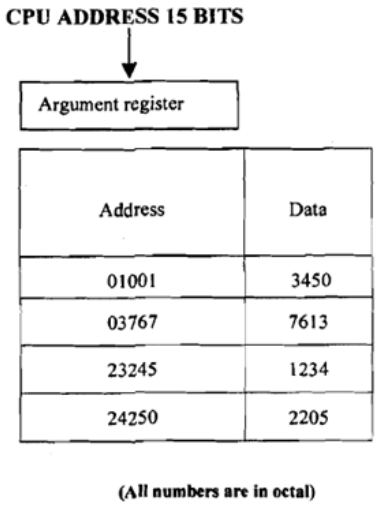


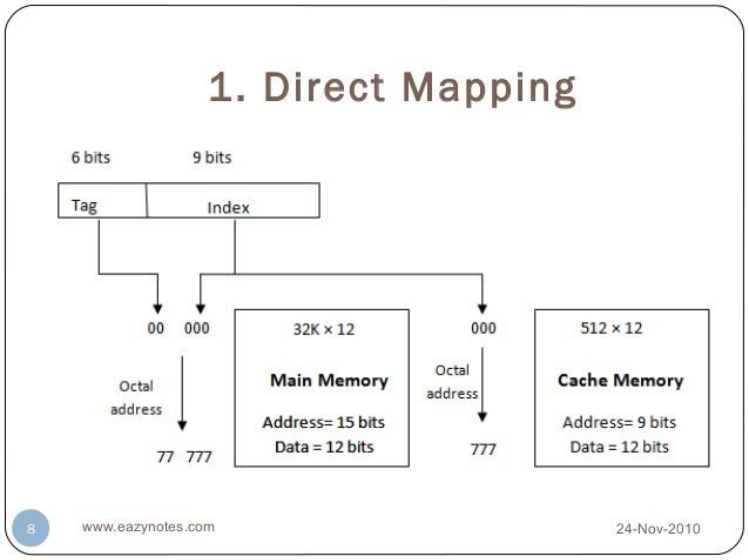
Figure 1 : Associative mapping

Direct Mapping

- The cpu address (15 bit) is divided into two fields i.e. index (9) and tag field (6).
- When a CPU generates a memory request, the index field is used for the address to access the cache.
- The tag field of the CPU address is compared with the tag in the word read from the cache. If two tag match , there is hit and desired word is in cache else searched in main memory .
- After that data will come in main memory.

➢ **Disadvantage**

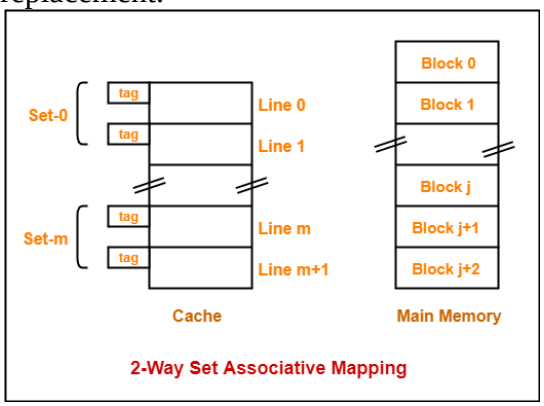
- The hit ratio could drop considerably if two or more words whose addresses have the same index but different tag are accessed repeatedly.



Set Associative Mapping

- Set-associative mapping is a compromise that exhibits the strengths of both the direct and associative approaches while reducing their disadvantages.
- The disadvantage of direct mapping is that two words with the same index in their address but with different tag values cannot reside in cache memory at the same time.
- It is the improvement over the direct mapping organization in that each word of cache can store two or more words of memory under the same index address.
- Each data word is stored together with its tag and the number of tag-data items in one word of cache is said to form a set.

- Common algorithm used for cache replacement are FIFO , LRU (Least Recently Used) and random replacement.



Replacement Algorithm

The most common replacement algorithms are:

Least recently used (LRU)

- This cache algorithm keeps recently used items near the top of cache. Whenever a new item is accessed, the LRU places it at the top of the cache. When the cache limit has been reached, items that have been accessed less recently will be removed starting from the bottom of the cache.This can be an expensive algorithm to use, as it needs to keep "age bits" that show exactly when the item was accessed. In addition, when a LRU cache algorithm deletes an item, the "age bit" changes on all the other items..

First-in-first-out (FIFO)

- The FIFO algorithm selects for replacement the page is loaded into memory, its identification number is pushed into a FIFO stack .FIFO will be full whenever memory has no more empty blocks .When a new page is must be loaded the page least recently brought in is removed. The page to remove is easily determined because its identification number is at the top of the FIFO stack.

- Advantage :
  Easy to implement.
- Disadvantage
  Under certain circumstances pages are removed and loaded from memory too frequently.

Least Frequently Used (LFU)

- This cache algorithm uses a counter to keep track of how often an entry is accessed. With the LFU cache algorithm, the entry with the lowest count is removed first. In LFU we check the old page as well as the frequency of that page and if the frequency of the page is larger than the old page we cannot remove it and if all the old pages are having same frequency then take last i.e FIFO method for that and remove that page.

Random

This cache algorithm chooses any page to replace at random. It assumes the next page to be referenced is random. It can test other algorithms against random page replacement. This type of replacement algorithm is not appropriate in usual case.

4. Explain the internal structure of hard disk.

It is a storage device that used magnetically platters to store data, instruction and information. The Desktop and notebook computer[contains 1 or more hard disks. It has storage capacity from 40 GB to 1.5 TB and more. These are some components of hard disk:

Platters:

It is a circular, metal disk that is mounted inside a hard disk drive. It is made of aluminum, glass, or ceramic and is coated with an alloy material that allows items to be recorded magnetically on its surface. Before writing data, the hard disk must be formatted.

The process of dividing the disk into tracks and sectors, so that the operating system can store and locate data and information on the disk is called Formatting. A narrow recording band that forms a full circle on the surface of the disk is called **Track**. The disk's storage locations consist of pie-shaped sections, which break the racks into small arcs called ***sectors.*** The smallest unit of disk space that stores data and information is called **Cluster** .

Disk case:

The rectangular shaped disk case holds all of the components of a hard disk drive.

Read/Write Head

A read/write head is the mechanism that reads items and writes items in the drive as it barely touches the disk's recording surface .

Cylinder

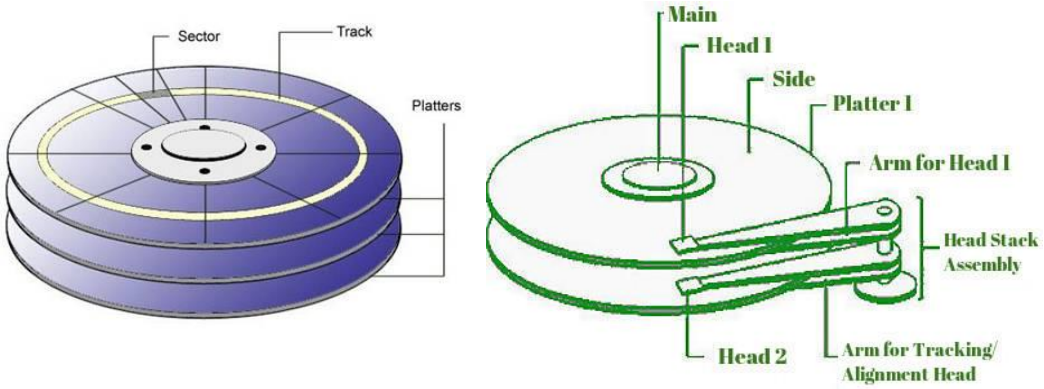
A cylinder is the vertical section of a track that passes through all platters.

Spindle

The spindle holds the platters together and the motor rotates the platters at their designated speed, which is measured in RPM. The platters in the hard disk rotate at a high[rate of speed. This spinning, which usually is 5,400 to 15,000 revolutions per minute (rpm) .

Disk cache

It is sometimes called a buffer, consists of a memory chip(s) on a hard disk that stores frequently accessed item



5. Explain the concept of stored program concept.

- A program could be represented in a form suitable for storing in memory with the data stored in same memory. During an execution of a program the stored instruction can be fetched from memory and a program could be set or altered by setting the values of a portion of memory. The fetched instruction can be decoded to set up the necessary data paths and generate the control signals. This is called stored program concept.
  - We organize a computer with one processor register and the instruction code format with two parts. The first part specifies the operation to be performed and the second specifies the address. The memory address tells the control where to find the operand in the memory .This operand is read from

the memory and used as the data to be operated on together with the data stored in the processor register.

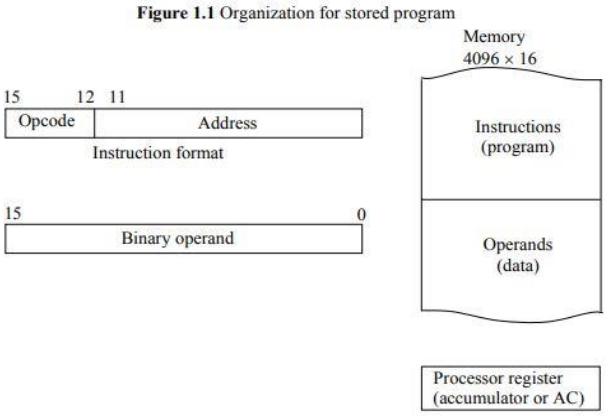


Figure 2: Stored Program Organisation

- The above figure shows the stored program organization implementation. Here instructions are stored in one section of memory and data in another in a same memory. For a memory unit with 4096 words we need 12 bits to specify an address since 2^12 =4096. If we store each instruction code in one 16 bit memory word, we have available 4 bits for the opcode and 12 bits to specify the address of the operand.The control reads a 16-bit instruction, usages 12-bit address to read a 16 bit operand from data portion of memory. Computer that have single-processor register usually assign to it the name accumulator (AC).If operation in instruction doesn't need any operand like (complement AC , increment AC) the rest bits can be used for other purposes.

6. Demonstrate the theory of direct and indirect address with instruction format diagrams.

- Sometime the second part of an instruction code not used as a address but as the actual operand, then the instruction is said to have an immediate operand .When the address part of instruction part specifies the address of an operand the instruction is said to be have a **direct address**. When the second part of the instruction code specifies the address of a memory word in which the address of the operand, the instruction is said to have indirect address. 1 bit of the instruction code can be used to distinguish between direct and an indirect address. So, A basic computers instruction has three parts, indirect address mode designated by one bit I , 3-bit Operational code and a 12-bit Address as shown in figure (a). The mode bit is 0 for a direct address and 1 for indirect address.

- Here in figure (b) the instruction ADD 456 is placed in memory location 22 and I bit is zero recognized as a direct addressing. The control finds the operand in memory 457 and adds it to the content of AC.
- Here in fig c the instruction ADD 300 is placed in memory location 35 has a I bit one recognized as a indirect addressing .the control goes the 300 to find the address of the operand .The address of the operand is 1350. So the operand fond in 1350 is added with the accumulator.

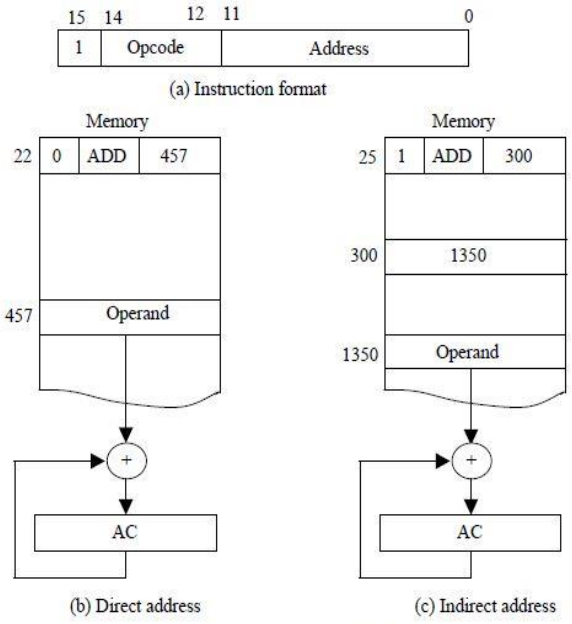


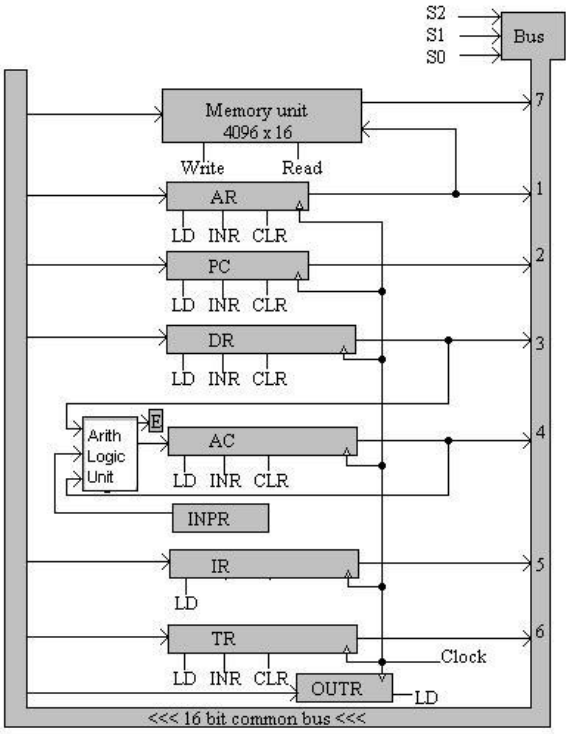
Figure 1-2 Calculation of direct and indirect address.

7. Define register. Explain the basic computer register with their usages.

- Registers are the memory that stores the temporary data .Computer instructions are stored in a memory. The control reads the instruction from memory and executes it. It then continuous by reading the next instruction in sequence and executes it and so on. But computer needs registers for storing instruction code, manipulating data, holding memory address and many other purposes.
- The below figure shows the basic computer Register and memory configuration. The memory unit has a capacity of 4096 words and each word contains 16 bits. Twelve bits of an instruction word are needed to specify the address of an operand. This leaves three bits for the operation part of the instruction and a bit to specify a direct or indirect address. The different register of basic computer with their function are as follows:
  - The **data register (DR) is a 16 bit register that** holds the operand read from memory.
  - The **accumulator (AC) is a 16 bit register that** is a general purpose processing register.
  - The instruction read form memory is placed in 16 bit **instruction register (IR)**.
  - The **temporary register (TR)** is used for holding temporary data during the processing.
  - The memory **address register** (AR) has 12 bits since this is the width of a memory address.
  - The **program counter (PC)** also has 12 bits and it holds address of the next instruction to be read from memory after the current the instruction is executed. The PC goes through a counting sequence and causes the computer to read sequential instructions previously stored in memory.
  - Two registers are used for input and output. The **input register (INPR)** receives an 8-bit character from an input device. The **output register (OUTR)** holds an 8-bit character for an output device.

8. Define Bus system. Explain the bus system construction mechanisms using Multiplexure and three state buffer gates.

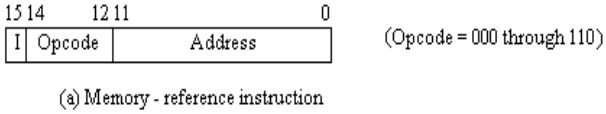
- The basic computer has eight registers a memory unit and a control unit .Path must be provided to transfer information from one register to another and between memory and registers. The number of wires will be excessive if connections are made between the outputs of each register and the input of the other register .So a Common Bus system is used to transfer information is a system with many register and other unit. The output of seven register and memory are connected to a common bus.



- The Figure shows the basic register which is connected in a common bus .It contains eight register and the memory unit connected by a common bus. The output of seven register and memory are connected to the common bus. Five register AR, PC, DR, AC, and TR have three control unit LD(load) , INR(increment) and CLR (clear). Two register have only LD input .The input data and output data of a memory are connected to the common bus, but a memory address is connected to the AR so AR is used to specify a memory address. The sixteen input of AC comes from an adder and logic circuit. Adder Logic circuit has three set of input coming from DR, AC and INPR. The input from DR and AC are for arithmetic and logical micro operation with result storing in the AC and carry in flip-flop E(extended AC bit).INPR (Input register) and OUTR (Output Register) have 8 bit each. INPR receives a character from an input device then transferred to AC.OUTR receives a character from AC and delivers it to an output device. Here four register DR , AC ,IR and TR are 16 bit each . AR and PC have 12 bit since they hold the memory address so when their content are placed in bus the 4 most significant bit are set to 0's.
- The specific output that is selected is determined by selection variables S_0,S_1,S_2.For example if $S_2S_1S_0$=011(3) the Data Register (DR) is selected .The 16 bit outputs of DR are placed in the system bus .The Register whose LD is enabled receives the data from the bus in next clock cycle. Again if $S_2S_1S_0$=111(7) the memory places its 16 bit output on the system bus.

9. Explain the instruction set design issues

- The basic computer has three instruction code formats as shown in figure below. Each instruction is of 16 bits. The opcode part of the instruction contains three bits and the remaining 13 bit depends on the opcode encounter. A **memory referenced instruction** uses 12 bits to specify an address and one bit to specify the addressing mode I. I=0 for the ***direct addressing*** and I=1 for the ***indirect address***.
- A **register reference instruction** is recognized by the operational code 111 with a 0 in the leftmost bit (bit 15) of the instruction. A register reference instruction specifies an operation on or a test of the AC register. An operand from the memory is not needed so the other 12 bits are used to specify the operation or test to be executed.
- Similarly in the **Input Output instruction** does not require the reference to the memory and is recognized by the opcode 111 with 1 in leftmost bit .The remaining 12 bit are used to specify the type of input output operation or test performed.

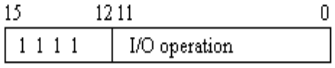


(Opcode = 000 through 110)

(a) Memory - reference instruction

(Opcode = 111, I = 0)

(b) Register - reference instruction



(Opcode = 111, I = 1)

(c) Input - output instruction

Basic Computer Instruction Formats

Basic Computer Organization & Design 15 *Instructions*

BASIC COMPUTER INSTRUCTIONS

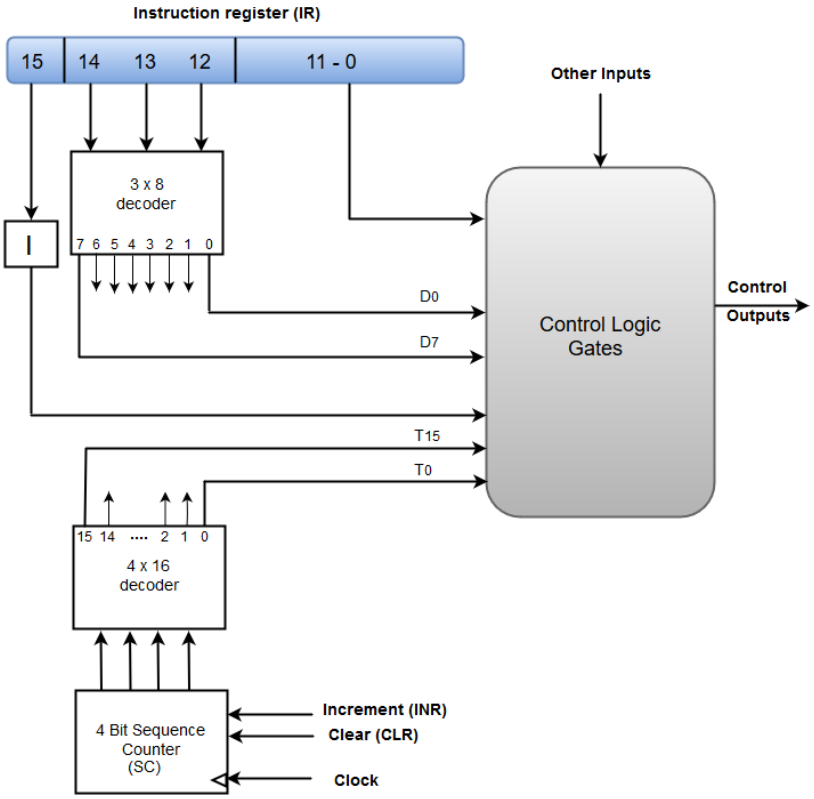
Symbol	Hex Code I = 0	Hex Code I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load AC from memory
STA	3xxx	Bxxx	Store content of AC into memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instr. if AC is positive
SNA	7008		Skip next instr. if AC is negative
SZA	7004		Skip next instr. if AC is zero
SZE	7002		Skip next instr. if E is zero
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

Computer Organization *Computer Architectures Lab*

10. Explain the structure of control unit of basic computer.(Hardwired control unit)

- The below diagram shows the block diagram of control unit. It consists of two decoder, a sequence counter and a number of logic gates. An instruction read from the memory is placed in the Instruction Register (IR) . The IR is divided into three parts: I bit, the operation code and bits 0 through 11 .the operation code in bit 12 to 14 are decoded with a 3*8 decoder .The eight output of decoder are designated by symbols D0 to D7. The subscripted decimal number is equivalent to the binary value of corresponding code. Bit 15 of instruction is transferred to flip-flop. Bit 0-11 is applied to control logic gate. The 4 bit sequence counter can count in binary from 0 through 15.The output of the counter are decoded into 16 timing signal T0 through t15 .Most of the time counter is incremented. When a timer is cleared to zero, causing the next active timing signal to be T0.
- Initially the CLR input of SC is active which in turn activates the T0 out of the decoder .T0 is active during one clock cycle .The positive transition labeled T0 in the figure trigger only those register whose control input are connected to timing signal T0.SC is increased every positive cycle producing T0,T1,T2,T3,T4 and so on up to T15 and back to T0.When timing signal becomes T4 becomes active , the output of the AND gate that implement the control function D3T4 becomes active. This signal is applied to CLR input of SC. On the next positive signal the counter is cleared to zero.

Control Unit of a Basic Computer:

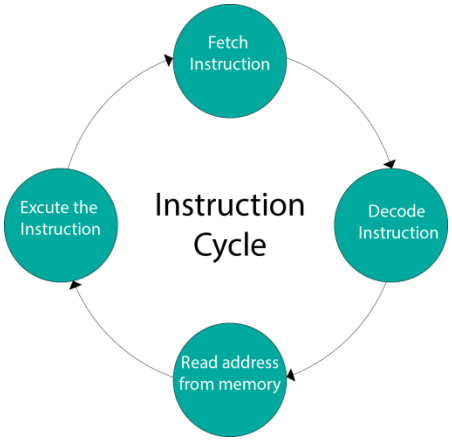


11. Define instruction cycle. Explain the instruction cycle with state diagram.

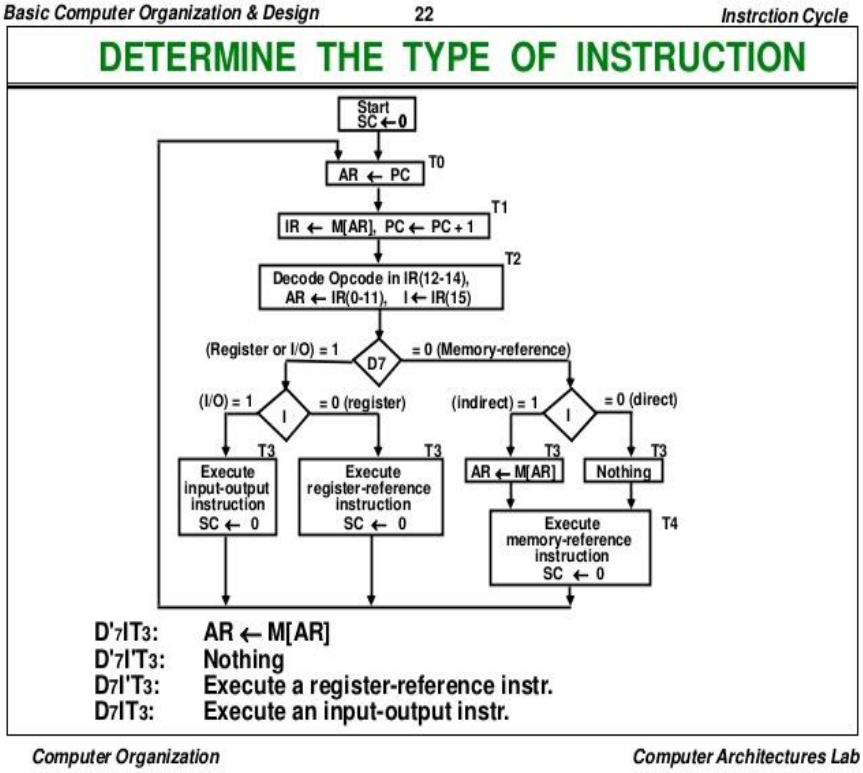
- A program residing in the memory unit of the computer consists of sequences of instruction. In the basic computer each computer instruction cycle consists of following phases :

a) Fetch an instruction from memory
b) Decode the instruction
c) Read the effective address from the memory if the instruction has an indirect address

d) Execute the instruction
Each instruction must pass through this phases that is to be executed and continues until a HALT instruction is encountered.



- A program residing in the memory unit of the computer consists of a sequence of instruction. The program is executed in the computer by going through a cycle for each instruction .Each instruction cycle is divided into number of subclasses and phases.



Fetch an Decode instruction:

Initially the program counter PC is loaded with the address of the first instruction in the program .The Sequence counter Sc is cleared to 0 , providing a decoded timing signal T_0.After each cock SC is incremented by one so that the timing signal must go through T_0 T_1 T_2 and so on . The microoperations for the fetch and decode phases can be specified by the following register transfer instruction.

T0: AR←PC
T1: IR ←M[AR], PC←PC +1
T2:D0,…….,D7←Decode IR(12-14), AR←IR(0-11) , I←IR(15)

In Clock cycle T0 address is transferred from PC to AR. In clock cycle T1 instruction read from memory is placed in IR and Pc is incremented. At time T2 the operational code is decoded, the indirect bit is transferred to I , address of instruction to AR and instruction to AR.

Determining the type of instruction

……yet to come