Parul® University

Computer Organization & Architecture - 203105253

Prof. Madhuri M. Thakkar, Assistant Professor
Electronics & Communication Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-1

FUNCTIONAL BLOCKS OF A COMPUTER

Parul® University

DIGITAL LEARNING CONTENT

Introduction

- **Computer:** A computer is a combination of **hardware and software** resources which integrate together and provides various functionalities to the user.
- **Hardware**: These are the physical components of a computer like the monitor, keyboard, processor, memory devices, etc.
- **Software**: These are the set of instructions or programs that are required by the hardware resources to function in a specific manner properly.

- **Digital Computer:** A digital computer can be defined as a programmable machine which reads the binary data passed as instructions, processes this binary input data, and displays a calculated digital output.

Parul® University

DIGITAL LEARNING CONTENT

Difference (computer architecture & organization)

- Architecture is those attributes visible to the programmer
  - Instruction set, number of bits used for data representation, I/O mechanisms, addressing techniques.
  - e.g. Is there a multiply instruction?

- Organization is how features are implemented
  - Control signals, interfaces, memory technology.
  - e.g. Is there a hardware multiply unit or is it done by repeated addition?

Parul® University

DIGITAL LEARNING CONTENT

Functional Components of a Computer

- The components, which aids the working-cycle of a computer i.e. the Input-Process- Output Cycle, are called as the **functional components** of a computer.
- Input unit takes the input.
- CPU (central processing unit) does the processing of data.
- Output unit produces the desired output.
- Memory unit holds the data and instructions during the processing. It includes temporary as well as permanent memory.

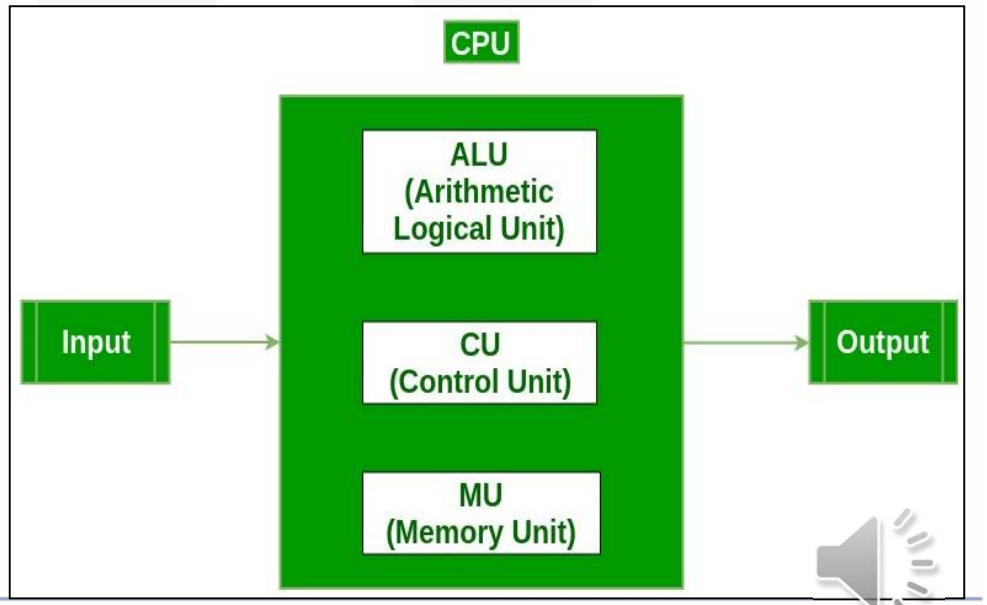


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

Functional Components of a Computer

- **Input Unit**: The input unit consists of input devices that are attached to the computer. These devices take input and convert it into series of 0s and 1s, i.e. binary language, which the computer understands. Some of the common input devices are mouse, keyboard, joystick, scanner etc.
- **Central Processing Unit (CPU)**: Data is fetched from memory or input device. Thereafter CPU executes or performs the required computation which in turn generates output. The output can then either be stored in memory or can be displayed on the output device.
  - Three main components of CPU are : Arithmetic Logic Unit (ALU), Control Unit (CU) and Memory registers

Parul® University

DIGITAL LEARNING CONTENT

Functional Components of a Computer

- **Arithmetic and Logic Unit (ALU):** IT performs mathematical calculations like addition, subtraction, multiplication and division and takes logical decisions like comparison and others.
- **Control Unit:** The Control unit coordinates and controls the data flow in and out of CPU, controls all the operations of ALU, memory registers and also input/output units. It decodes the fetched instruction, interprets it and sends control signals to input/output devices according to the operation to be performed.
- **Memory Registers:** Registers are used to store the data which is directly used by the processor. Registers can be of different sizes(16 bit, 32 bit, 64 bit and so on) and each register inside the CPU has a specific function like storing data, storing an instruction, storing address of a location in memory etc. Accumulator (ACC) is the main register in the ALU and contains one of the operands of an operation to be performed in the ALU.

Parul® University

DIGITAL LEARNING CONTENT

Functional Components of a Computer

- **Memory:** Memory attached to the CPU is used for storage of data and instructions and is called internal memory, The internal memory is divided into many storage locations, each of which can store data or instructions. Each memory location is of the same size and has an address. With the help of the address, the computer can read any memory location easily without having to search the entire memory. The internal memory is also called the Primary memory or Main memory.
- **Output Unit:** The output unit consists of output devices that are attached with the computer. It converts the binary data coming from CPU to human understandable form. The common output devices are monitor, printer, plotter etc.,

Parul® University

DIGITAL LEARNING CONTENT

Generations of Computer

There are various generations of computer were invented.

FIRST GENERATION (1945 – 1955):

- Program and data reside in the same memory (stored program concepts – John von Neumann).
- ALP was used to write programs.
- Vacuum tubes were used to implement the functions (ALU & CU design).
- Magnetic core and magnetic tape storage devices are used.
- Electronic vacuum tubes were being used as the switching components.

Parul® University

DIGITAL LEARNING CONTENT

Generations of Computer

SECOND GENERATION (1956 – 1964):

- Transistor were used to design ALU & CU.
- HLL was started being used (FORTRAN).
- To convert HLL to MLL compiler were used.
- Separate I/O processor were developed to operate in parallel with CPU, thus improving the performance.
- Invention of the transistor which was faster, smaller and required considerably less power to operate.

Generations of Computer

SECOND GENERATION (1956 – 1964):

- A **high-level language** (**HLL**) is a programming language such as C, FORTRAN, or Pascal that enables a programmer to write programs that are more or less independent of a particular type of computer. Such languages are closer to human languages and further from machine languages.
- Difference between High Level and Low level languages
- Both **High level language** and **low level language** are the programming languages's types.
- The main difference between **high level language** and **low level language** is that, Programmers can easily understand or interpret or compile the high level language in comparison of machine. On the other hand, Machine can easily understand the low level language in comparison of human beings.

Parul® University

DIGITAL LEARNING CONTENT

Generations of Computer

SR.	HIGH LEVEL LANGUAGE	LOW LEVEL LANGUAGE
1.	It is programmer friendly.	It is a machine friendly.
2.	It is less memory efficient.	It is high memory efficient.
3.	It is easy to understand.	It is tough to understand.
4.	It is simple to debug.	It is complex to debug.
5.	It is simple to maintain.	It is complex to maintain.
6.	It is portable.	It is non-portable.
7.	It can run on any platform.	It is machine-dependent.
8.	It needs compiler or interpreter for translation.	It needs assembler for translation.

Parul® University

DIGITAL LEARNING CONTENT

Generations of Computer

THIRD GENERATION (1965-1971):

- IC (Integrated circuit) technology improved
- Improved IC technology helped in designing low cost, high speed processor and memory modules
- Multiprogramming, pipelining concepts were incorporated
- DOS allowed efficient and coordinate operation of computer system with multiple users
- Cache and virtual memory concepts were developed
- More than one circuit on a single silicon chip became available

Generations of Computer

THIRD GENERATION (1965-1971):

- An integrated circuit, or IC, is small chip that can function as an amplifier, oscillator, timer, microprocessor, or even computer memory. An IC is a small wafer, usually made of silicon, that can hold from hundreds to millions of transistors, resistors, and capacitors.
- DOS (Disk Operating System) is an operating system that runs from a hard disk drive. The term can also refer to a particular family of disk operating systems, most commonly MS-DOS (Microsoft Disk Operating System).

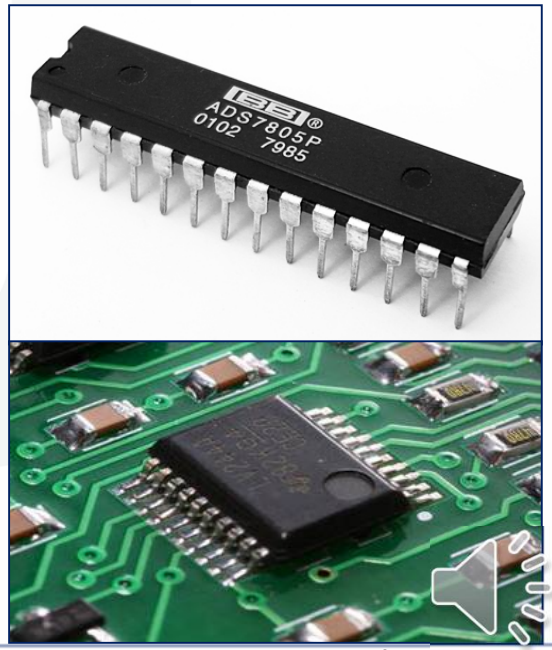


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

Generation of Computer

FOURTH GENERATION (1972-1985):

- CPU – Termed as microprocessor
- INTEL, MOTOROLA, TEXAS,NATIONAL semiconductors started developing microprocessor.
- Workstations, microprocessor (PC) & Notebook computers were developed.
- Interconnection of different computer for better communication: LAN, MAN, WAN.
- Specialized processors like Digital Signal Processor were also developed.

Difference between LAN, MAN and WAN

- **LAN-** Local Area Network is a group of network devices which allow communication between connected devices.
- **MAN** -Metropolitan Area Network, Covering the largest area than LAN such as: City.
- **WAN** -Wide Area Network, Covering large area than LAN as well as MAN such as: Country/Continent etc.

Generation of Computer

BEYOND THE FOURTH GENERATION (1985 – TILL DATE):

- E-Commerce, E- banking, home office
- ARM, AMD, INTEL, MOTOROLA
- High speed processor - GHz speed
- Because of submicron IC technology, a lot of features were added in small size.
- They will be intelligent like human beings and will use Artificial Intelligence (AI) for working.

Generation of Computer

- **e Banking**
- E banking or online banking is nothing but allowing a customer to use internet to access his account anytime he wishes sitting in the comfort of his home or office or anywhere else. E banking, which started slowly has today become a need and also allows banks to cut down on expenditure involved with extra staff.

- **e Commerce**
- E commerce is the name given to trading activities that are conducted using the power of internet. E commerce is simply online transactions. Buying and selling of goods and services using money through internet.

Interconnection between Functional Components

- A computer consists of input unit that takes input, a CPU that processes the input and an output unit that produces output.
- A bus is a transmission path, made of a set of conducting wires over which data or information in the form of electric signals, is passed from one component to another in a computer.
- The bus can be Address bus, Data bus and Control Bus.

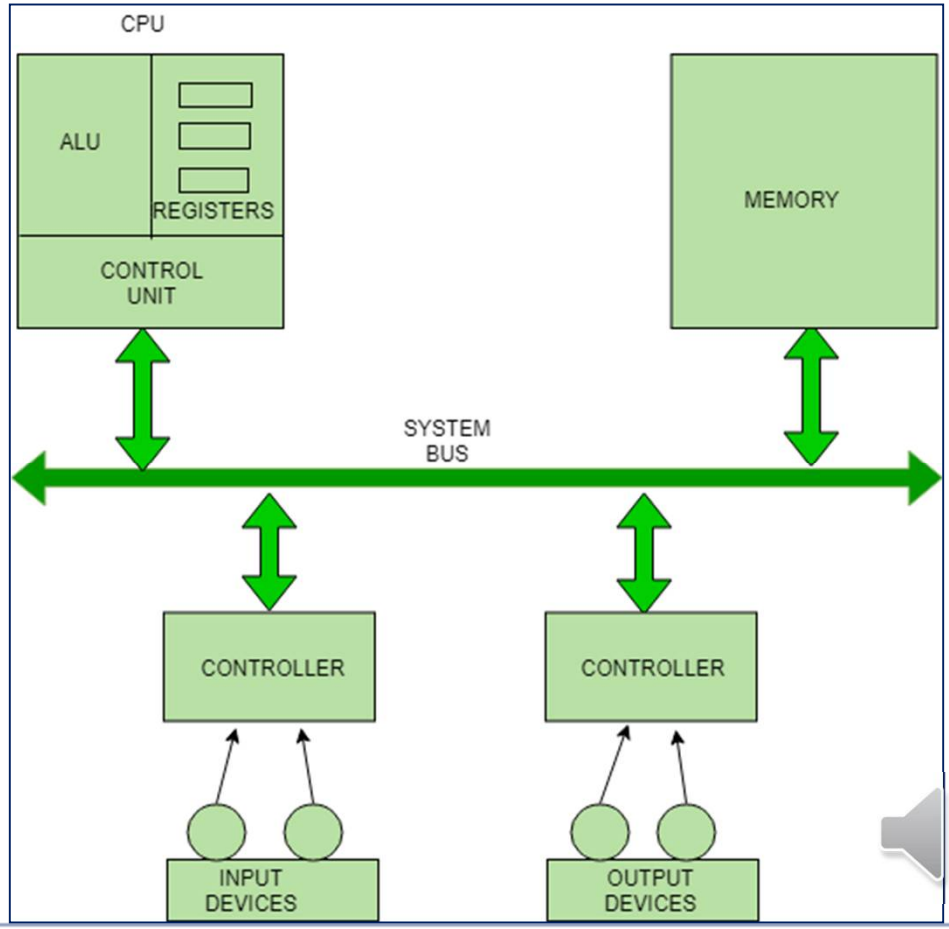


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

BUS STRUCTURE - Connecting CPU and memory

- The CPU and memory are normally connected by three groups of connections, each called a bus: data bus, address bus and control bus. Functional units may be connected by a group of parallel wires. The group of parallel wires is called a bus.
- Types of Bus:
  - System Bus/Memory Bus
  - Data Bus
  - Address Bus
  - Control Bus
  - I/O Bus

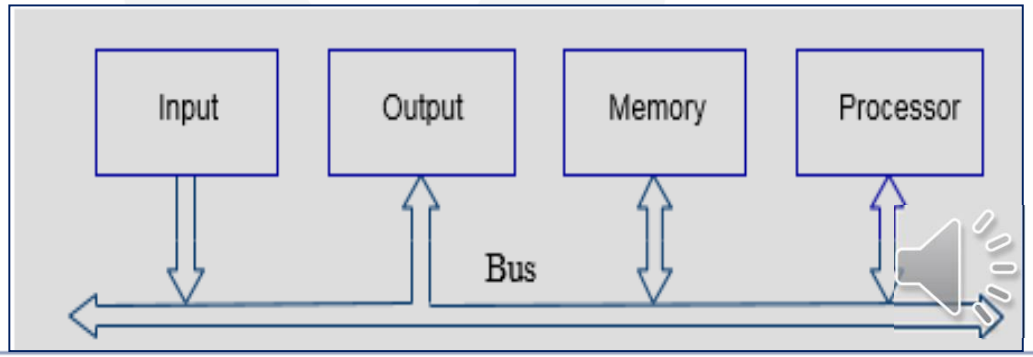


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

BUS STRUCTURE - Connecting CPU and memory

- The address bus carries the address location of the data or instruction.
- The data bus carries data from one component to another.
- The control bus carries the control signals.
- The system bus is the common communication path that carries signals to/from CPU, main memory and input/output devices. The input/output devices communicate with the system bus through the controller circuit which helps in managing various input/output devices attached to the computer.
- Figure shows the interconnection of Processor Functional units to memory and I/O subsystem .

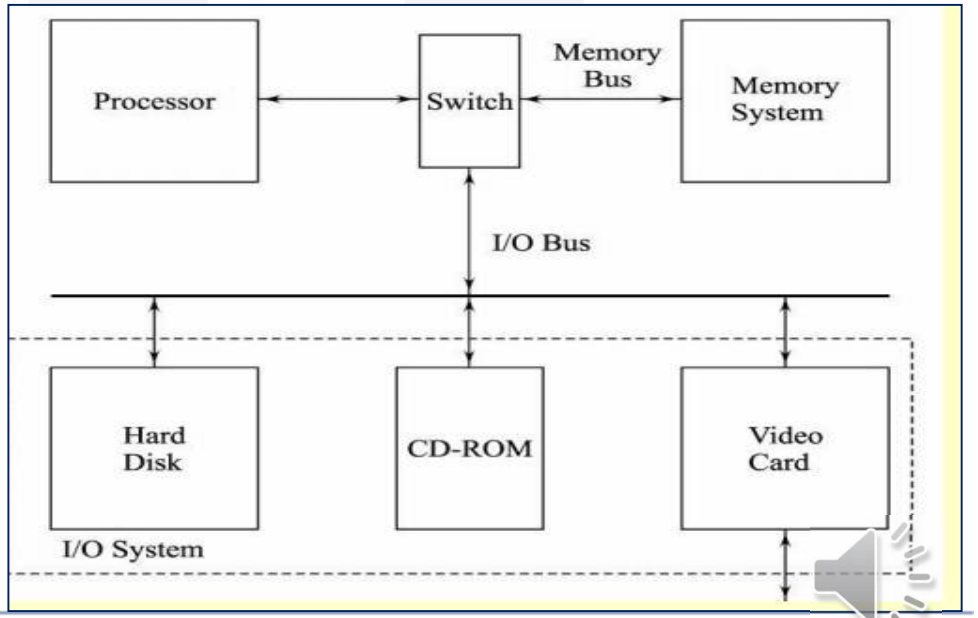


Image source : Google

Bus organizations

Single Bus organization and Two Bus organization

Single Bus Organization

- Three units share the single bus. Information can be transferred between any two units at a time.
- I/O units use the same memory address space. i.e. Memory mapped I/o. So, no special instructions are required to address the I/O, it can be accessed like a memory location.
- Due to difference in the operating speed of the devices in the system, there may be lack of synchronism between transmitter and receiver.
- So, buffer registers are used to hold the data during transfers. For eg, Communication between the processor and printer.

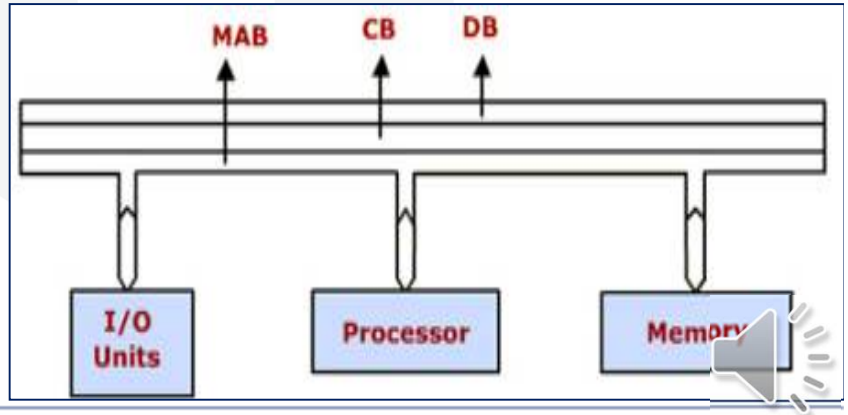


Image source : Google

Bus organizations

Two Bus Organization

- In two bus organization, various units are connected through two independent buses. I/O units are connected to the processor through an I/O bus and memory is connected to the processor through the memory bus.
- I/O bus consists of address, data and control bus, memory bus. In this type of organization, processor completely supervises the transfer of information to and from I/O units. All the information is first taken to processor and from there to the memory. This kind of transfer is known as program controlled transfer.

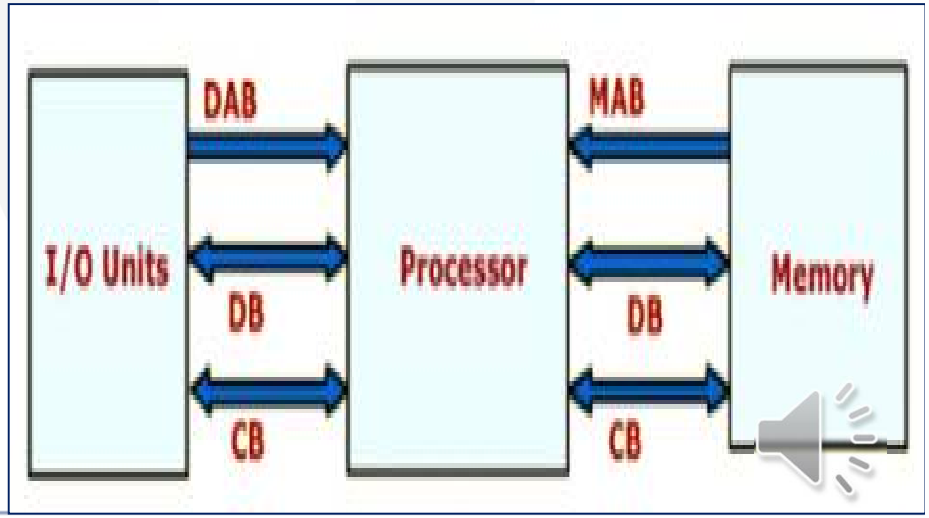


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

Classification of Computer

Computers can be classified in three types:

1. Analog Computers
2. Digital Computers
3. Hybrid Computers

- Analog computers work on the principle of measuring physical magnitudes, such as voltages, resistances or currents, to represent the quantities being manipulated rather than directly dealing with the numbers.
- Digital computers are those that operate with information in binary input or information represented in a digital form. Digital computers process data into a digital value (in 0s and 1s). They give more accurate and faster result.
- Hybrid computers incorporate the measuring feature of an analog computer and counting feature of a digital computer.

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set Architecture (ISA)

- The Instruction Set Architecture (ISA) is the part of the processor that is visible to the programmer or compiler writer.
- The ISA serves as the boundary between software and hardware
- The 3 most common types of ISAs are:
    1. Stack - The operands are implicitly on top of the stack.
    2. Accumulator - One operand is implicitly the accumulator.
    3. General Purpose Register (GPR) - All operands are explicitly mentioned, they are either registers or memory locations.

Functioning of computer

- Program
  - A program is a set of instructions that specify the operations, operands and the sequence by which processing has to occur.
- Computer Instruction
  - A computer instruction is a binary code that specifies a sequence of microoperations for the computer.
  - The computer reads each instruction from memory & places it in a control register.
  - The control then interprets the binary code of the instruction and proceeds to execute it by issuing a sequence of microoperations.
- Instruction Code
  - It is a group of bits that instruct the computer to perform a specific operation.
  - Example

Parul® University

DIGITAL LEARNING CONTENT

Functioning of computer

- Operation Code (Opcode)
  - The operation code of an instruction is a group of bits that define such operations as add, subtract, multiply, shift, and complement.
  - The number of bits required for the operation code of an instruction depends on the total number of operations available in the computer.
  - The operation code must consist of at least n bits for a given 2n (or less) distinct operations.

Stored Program Organization

- In stored program organization, one processor register(AC) and an instruction code format with two parts are there.
- The first part specifies the operation (opcode) to be performed and the second specifies an address (operand).
- The memory address tells the control where to find an operand in memory.
- This operand is read from memory and used as the data to be operated on together with the data stored in the processor register.

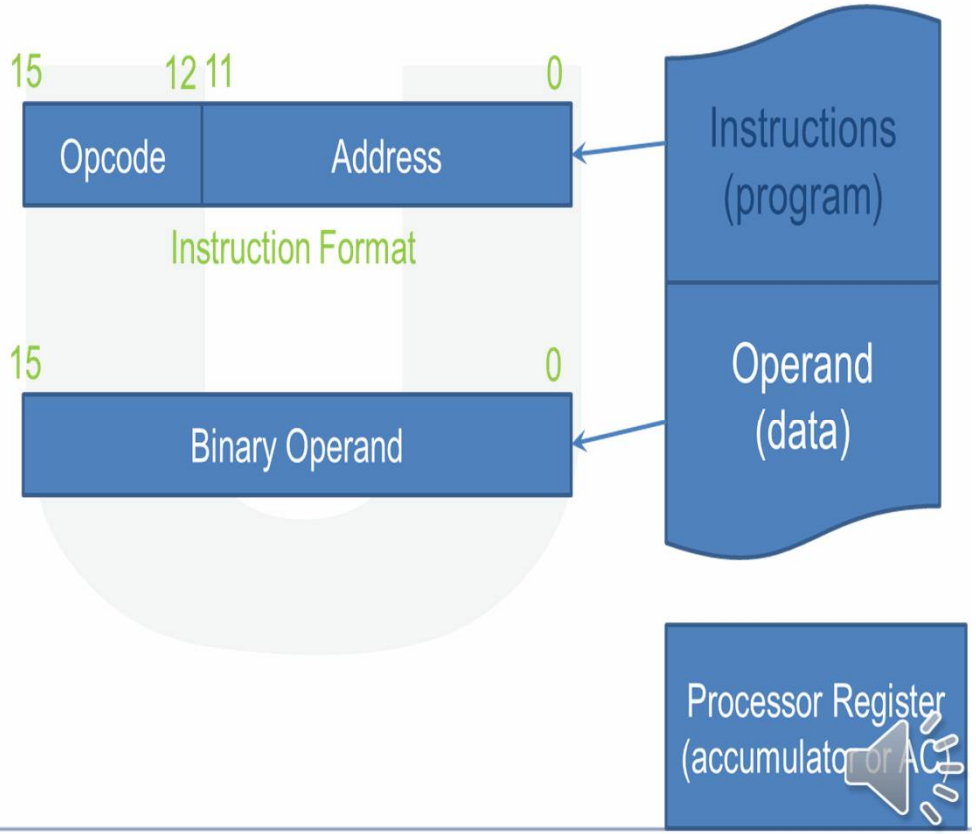


Image source : Google

Instruction format of basic computer

- Basic computer has word size of 16 bits and address size of 12 bits.
- In the format, bit 0 to 11 specifies address, 12 to 14 specifies opcode and bit 15 specifies direct or indirect memory in case of opcode ranging from 000 to 110.
- Bit 15 can specify whether the instruction is register reference or I/O reference in case of opcode is 111.

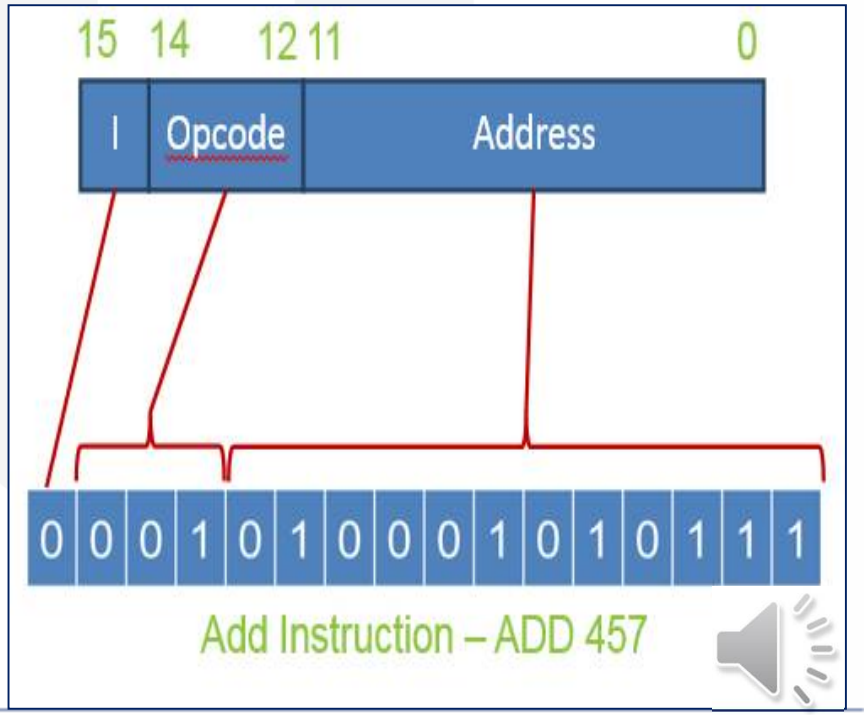


Image source : Google

Direct & Indirect Addressing of Memory

- If the second part of an instruction format specifies the address of an operand, the instruction is said to have a direct address.
- In Indirect address, the bits in the second part of the instruction designate an address of a memory word in which the address of the operand is found.
- The bit no 15 (mode bit) is 0 for a direct address and 1 for an indirect address.

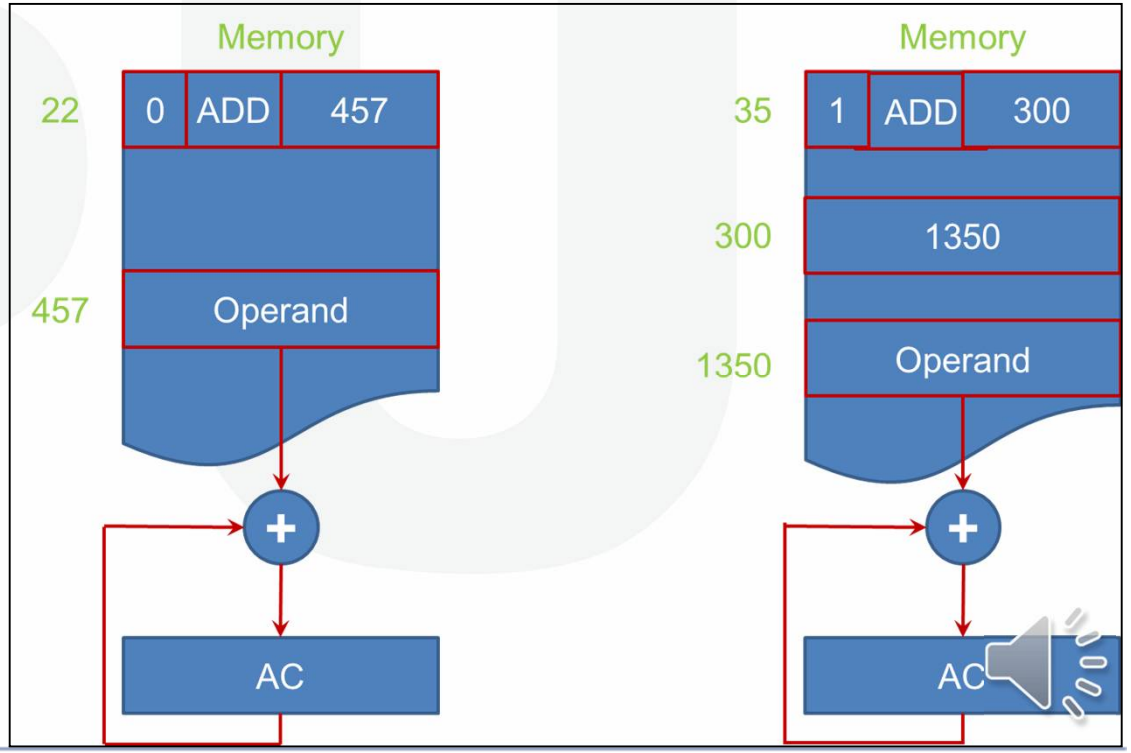


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

Computer Registers:

- In computer, instruction sequencing needs a counter to calculate the address of the next instruction after execution of the current instruction is completed.
- It is also necessary to provide a register in the control unit for storing the instruction code after it is read from memory.
- The computer needs processor registers for manipulating data and a register for holding a memory address.
- Due to these requirements, various kinds of registers are required in computer architecture. The register configuration is shown here.

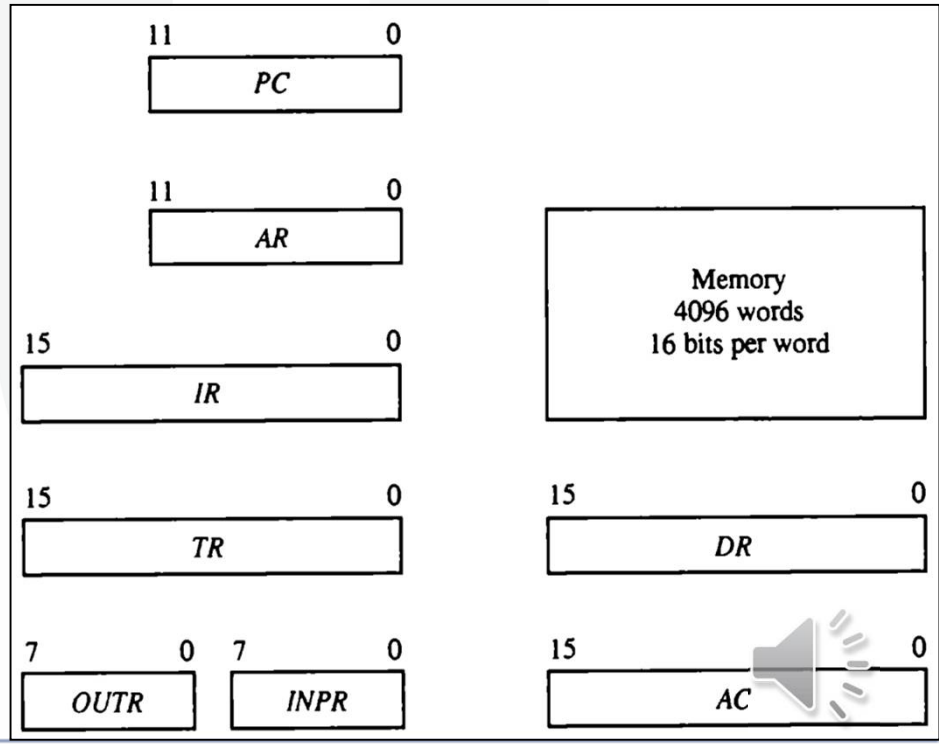


Image source : Google

Common Bus system:

- Outputs of all the registers except the OUTR are connected to the common bus. The output selected depends upon the binary value of variables S^2, S^1 and S^0.
- A register receives the information from bus when its LD/Write input is enabled.
- The contents of memory are placed onto the bus when its Read input is activated.
- DR, AC, IR and TR have 16 bits
- AR (which specifies address always) and PC have 12 bits. INPR and OUTR have 8 bits each. 5 registers have 3 control inputs LD (load), INR (increment) and CLR (clear).

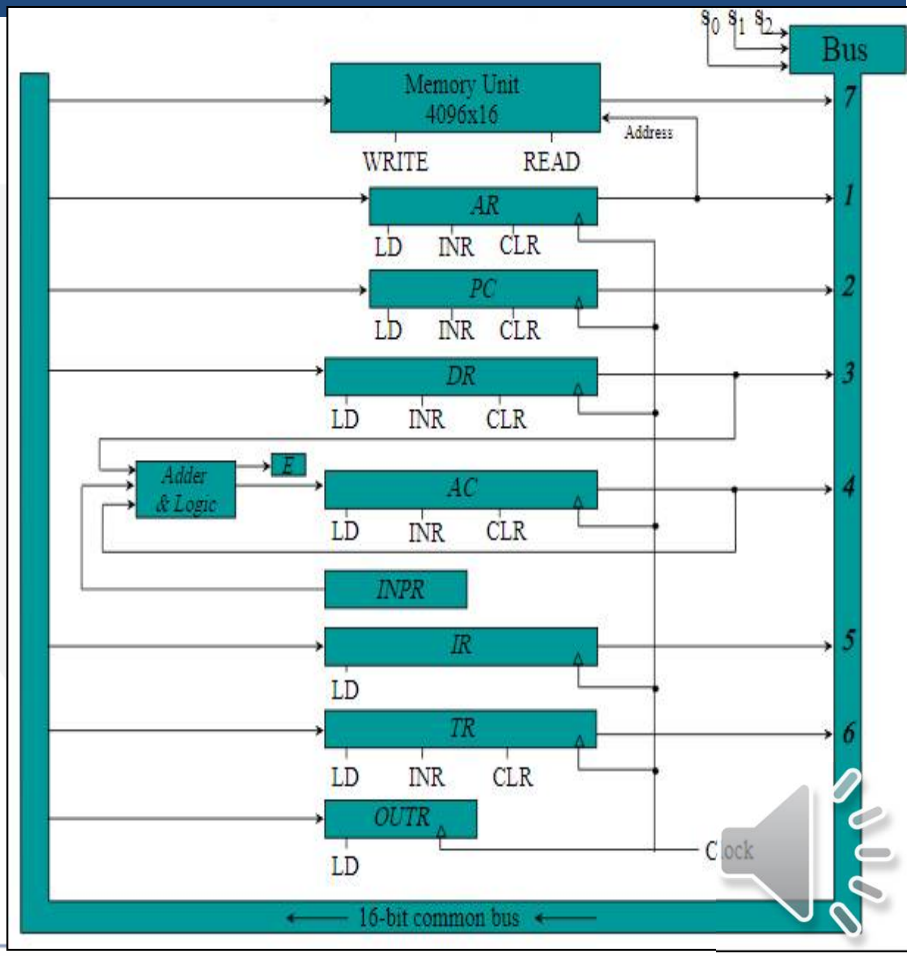


Image source : Google

Computer Instructions

- The basic computer has three instruction code formats as shown in figure below. Each format has 16 bits. The opcode part of instruction contains three bits and the meaning of the remaining 13 bits depends on the operation code encountered.
- A memory reference instruction uses 12 bits to specify an address and one bit to specify the addressing mode I. I is equal to 0 for direct address and to 1 for indirect address.
- In memory reference instructions, opcode can be from 000 to 110 only, 111 is reserved for Register reference or Input-output instruction.
- Register reference instruction has 0 at the leftmost bit in the format, where as I/O instruction has 1 in the said position.

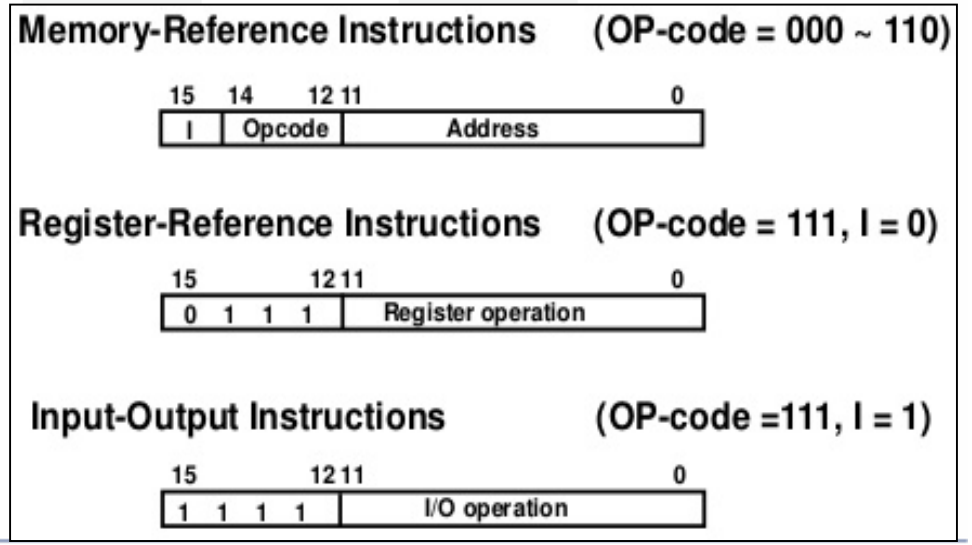


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

Computer Instructions

- Memory reference instructions →
- Register Reference instruction→
- I/O instructions→

Symbol	Hexadecimal code $I = 0$	Hexadecimal code $I = 1$	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store content of AC in 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 instruction if AC positive
SNA	7008		Skip next instruction if AC negative
SZA	7004		Skip next instruction if AC zero
SZE	7002		Skip next instruction if E is 0
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

Image source : Google

Timing and control

- The timing for all registers in basic computer is controlled by a master clock generator. The control signals are generated in the control unit and provide control inputs for the multiplexers in the common bus, control inputs in processor registers, and microoperations for the accumulator.
- There are two major types of **control organization: Hardwired control and microprogrammed control.**
- In the hardwired organization, the control logic is implemented with gates, flip-flops, and other digital circuits. Its advantage is that it can be optimized to produced a fast mode of operation & The disadvantage is that it requires to modify the circuit if we need any changes.
- In the microprogrammed organization, the control information is stored in a control memory. The control memory is programmed to initiate the required sequence of microoperation. In this case, updating the microprogram is required in the control memory if we need any changes.

Timing and control

- The control unit (CU) is a component of a computer's central processing unit (CPU) that directs the operation of the processor. It tells the computer's memory, arithmetic and logic unit and input and output devices how to respond to the instructions that have been sent to the processor.

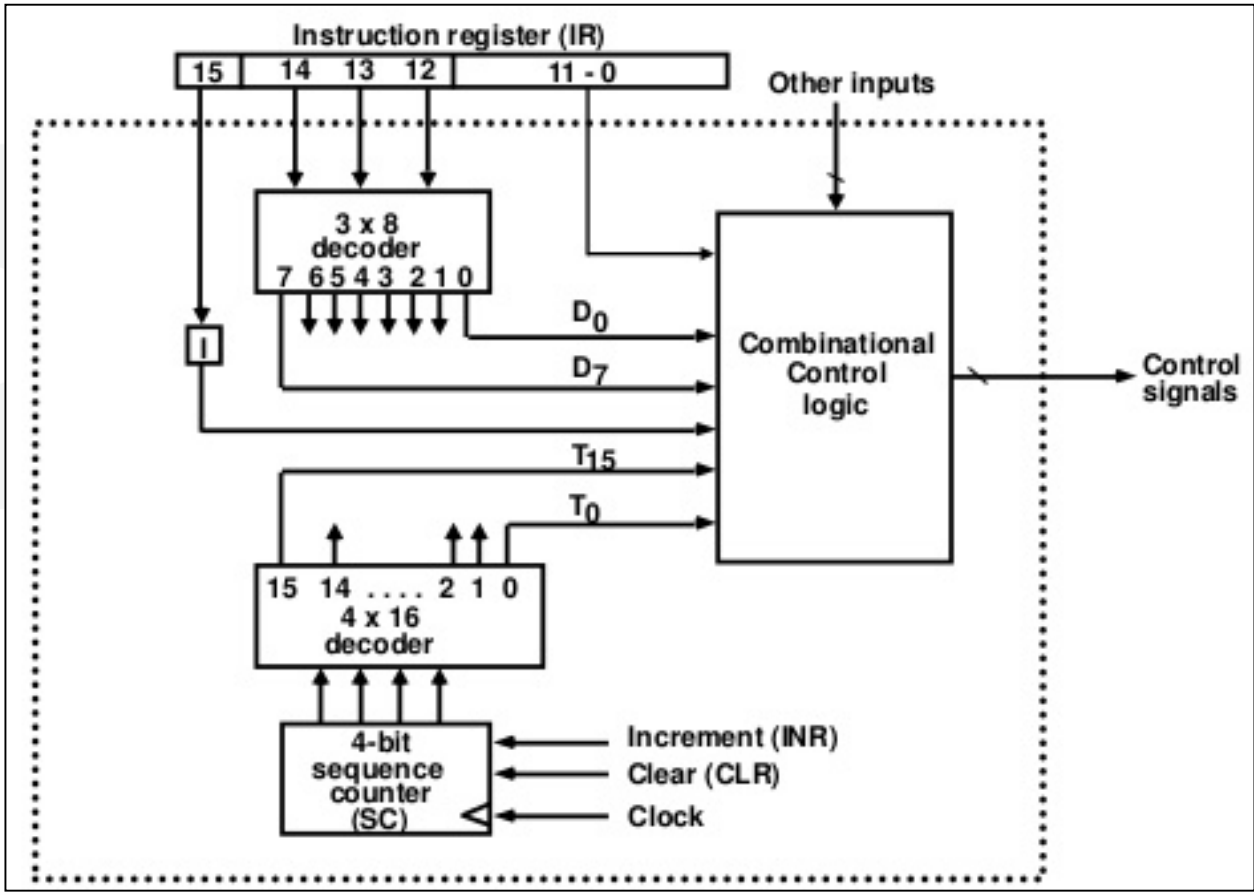


Image source : Google

Timing and control

- A Hard-wired Control consists of two decoders, a sequence counter, and a number of logic gates.
- An instruction fetched from the memory unit is placed in the instruction register (IR).
- The component of an instruction register includes; I bit, the operation code, and bits 0 through 11.
- The operation code in bits 12 through 14 are coded with a 3 x 8 decoder.
- The outputs of the decoder are designated by the symbols D0 through D7.
- The operation code at bit 15 is transferred to a flip-flop designated by the symbol I.
- The operation codes from Bits 0 through 11 are applied to the control logic gates.
- The Sequence counter (SC) can count in binary from 0 through 15.

Timing and control

Timing Diagram

- Fig shows the time relationship of the control signals.
- The SC responds to the positive transition of the clock. Initially, the CLR input of SC is active.
- The first positive transition of the clock clears SC to 0, which in turn activates the timing T0 out of the decoder.
- T0 is active during one clock cycle. SC is incremented with every positive clock transition, unless its CLR input is active.
- This procedures the sequence of timing signals T0, T1, T2, T3 and T4, and so on. If SC is not cleared, the timing signals will continue with T5, T6, up to T15 and back to T0. [4]

Timing and control

- Assume, at time T4, SC is cleared to 0 if decoder output D3 is active.
- The last three waveforms shows how SC is cleared when D3T4 = 1. Output D3 from the operation decoder becomes active at the end of timing signal T2.
- When T4 becomes active, the output of the AND gate that implements the control function D3T4 becomes active.
- This signal is applied to the CLR input of SC. On the next +ve clock transition, the SC is cleared to 0. This causes the timing signal T0 to become active instead of T5 that would have been active if SC were incremented instead of cleared.

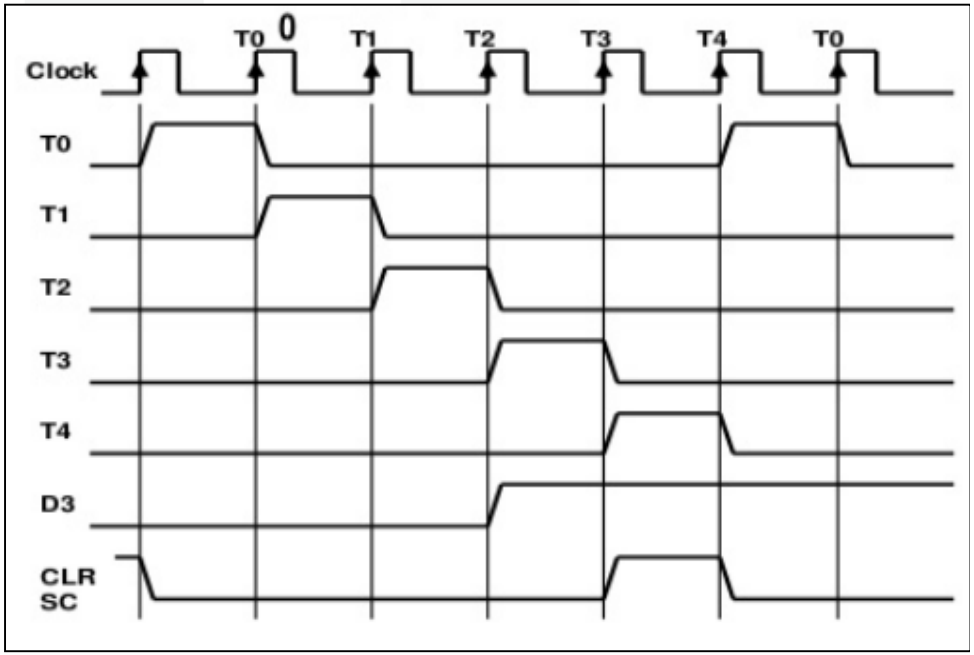


Image source : Google

Instruction Execution Cycle

- In a basic computer, each instruction cycle consists of the following phases:
  1. Fetch instruction from memory.
  2. Decode the instruction.
  3. Read the effective address from memory.
  4. Execute the instruction.

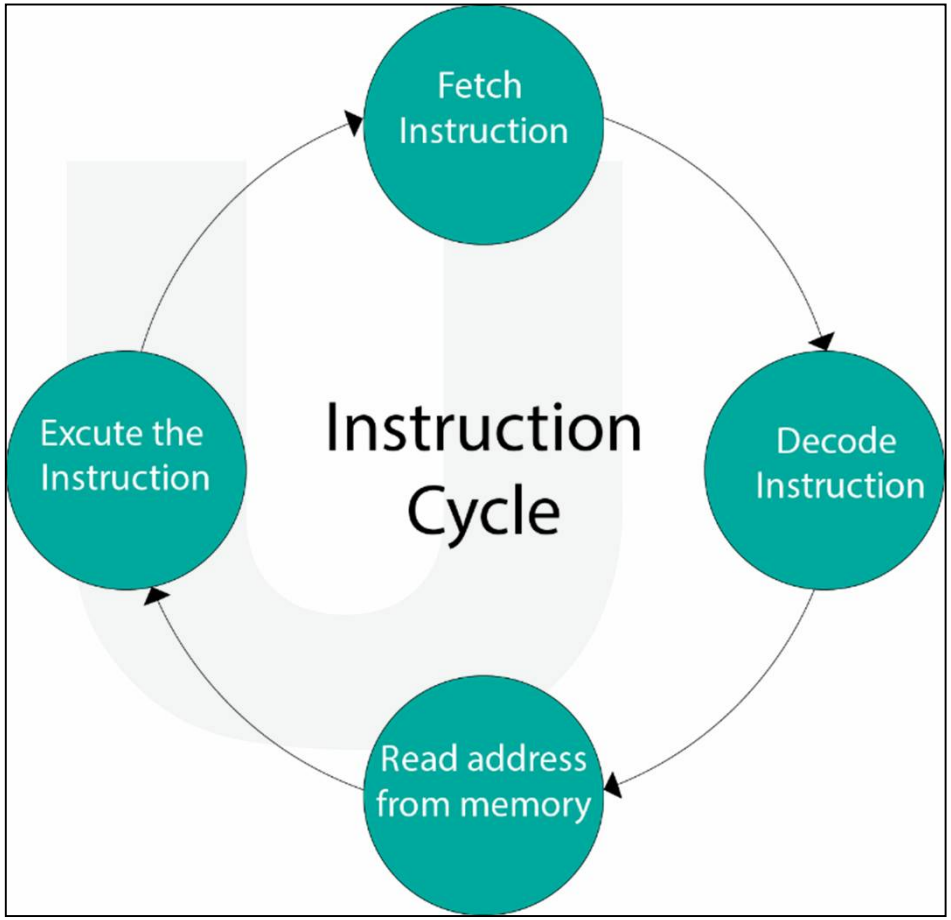


Image source : Google

Parul® University

DIGITAL LEARNING CONTENT

Instruction Cycle

- A program residing in the memory unit of the computer consists of a sequence of instructions.
- In the basic computer each instruction cycle consists of the following phases: Fetch, Decode, Find the effective address and Execute the instruction
- After step 4 of execution, control goes back to step 1 to fetch another instruction until the HALT instruction is executed.

Instruction Cycle

- The flowchart presents an initial configuration for the instruction cycle and shows how the control determines the instruction type after the decoding.

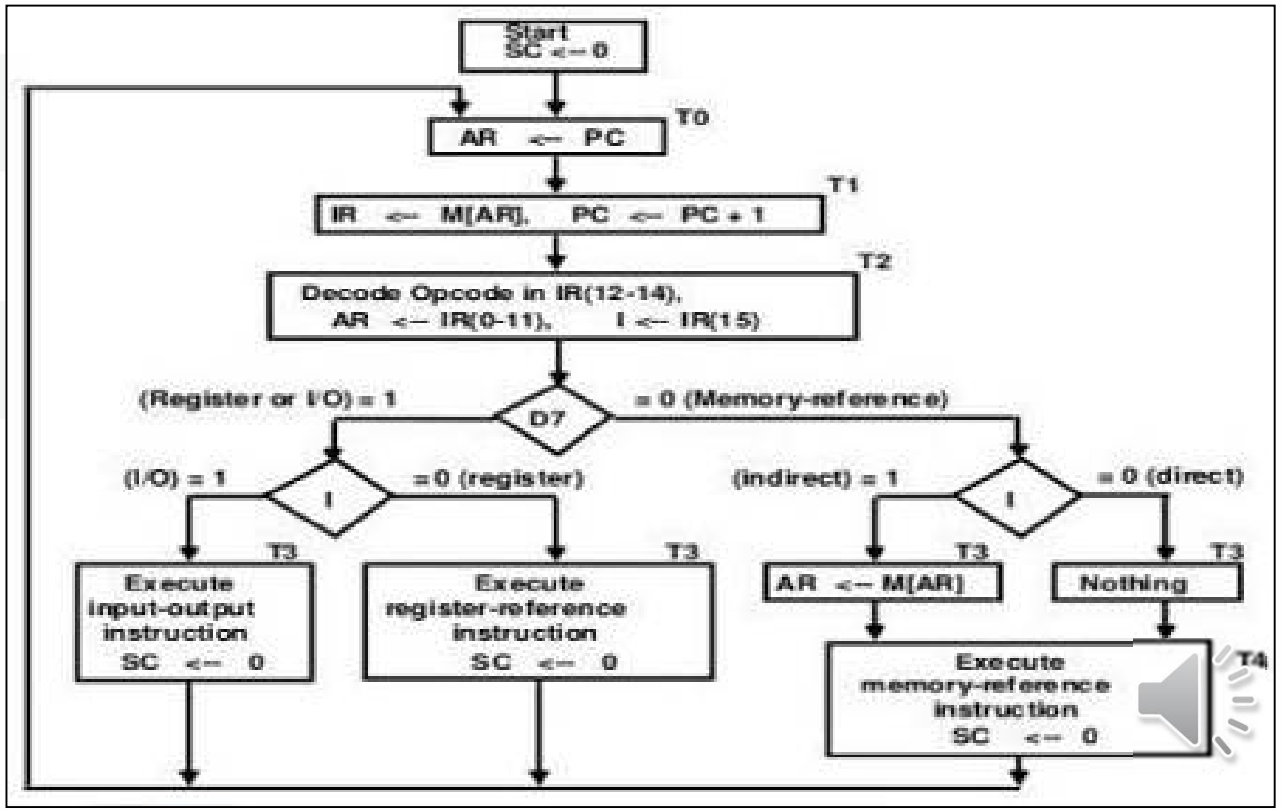


Image source: Google

Parul® University

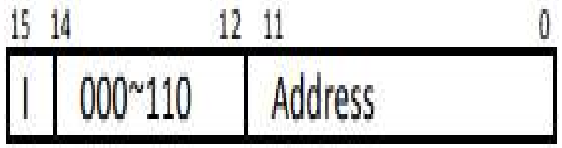
Register reference instructions

- When the register-reference instruction is decoded, D7 bit is set to 1.
- Each control function needs the Boolean relation D7 I' T3
- The basic format of register reference instruction and basic 12 register reference instructions are shown here.

CLA	rB11	AC←0
CLE	rB10	E←0
CMA	rB9	AC←AC’
CME	rB8	E←E’
CIR	rB7	AC ← shr AC, AC(15) ← E, E ← AC(0)
CIL	rB6	AC ← shl AC, AC(0) ← E, E ← AC(15)
INC	rB5	AC ← AC + 1
SPA	rB4	if (AC(15) = 0) then (PC ← PC+1)
SNA	rB3	if (AC(15) = 0) then (PC ← PC+1)
SZA	rB2	if (AC = 0) then (PC ← PC+1)
SZE	rB1	if (E = 0) then (PC ← PC+1)
HLT	Rb0	S ← 0 (S is a start-stop flip-flop)

Memory reference instructions

- When the memory-reference instruction is decoded, D7 bit is set to 0.
- The basic format of memory reference instruction and basic 7 memory reference instructions are shown here.



Symbol	Operation Decoder	Symbolic Description
AND	D_0	AC←AC^M[AR]
ADD	D_1	AC←AC+M[AR], E←C_{OUT}
LDA	D_2	AC←M[AR]
STA	D_3	M[AR]←AC
BUN	D_4	PC←AR
BSA	D_5	M[AR] ←PC, PC←AR+1
ISZ	D_6	M[AR] ←M[AR]+1, IF M[AR]+1=0 then PC←PC+1

BSA

- BSA: Branch and Save Return Address
- This instruction is useful for branching to a portion of the program called a subroutine or procedure.
- When executed, the BSA instruction stores the address of the next instruction in sequence (which is available in PC) into a memory location specified by the effective address..
- Figure shows the example of BSA instruction.

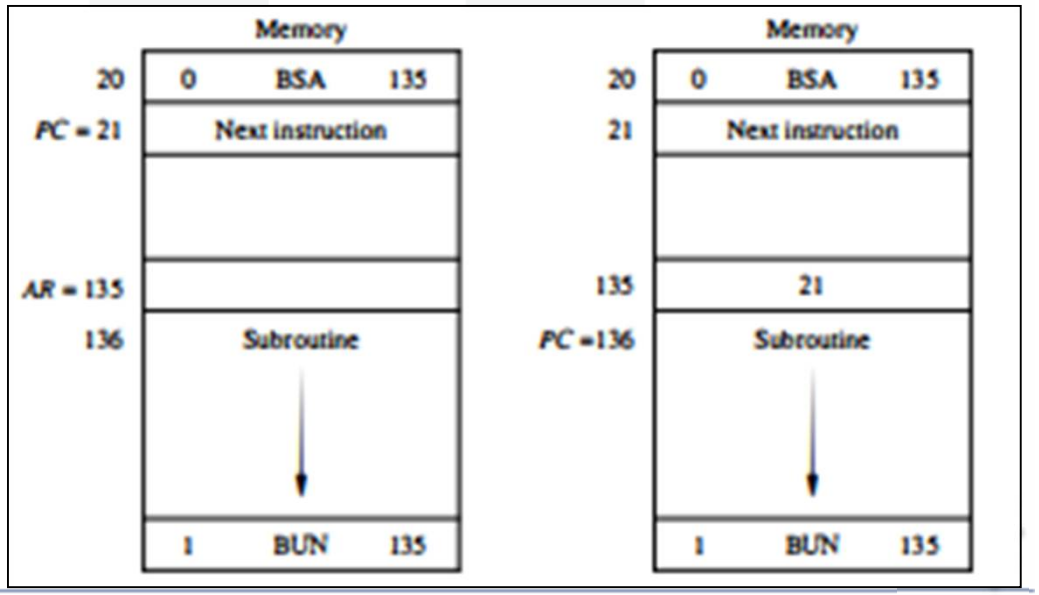


Image source: Google

Parul® University

DIGITAL LEARNING CONTENT

Memory reference instructions internal steps

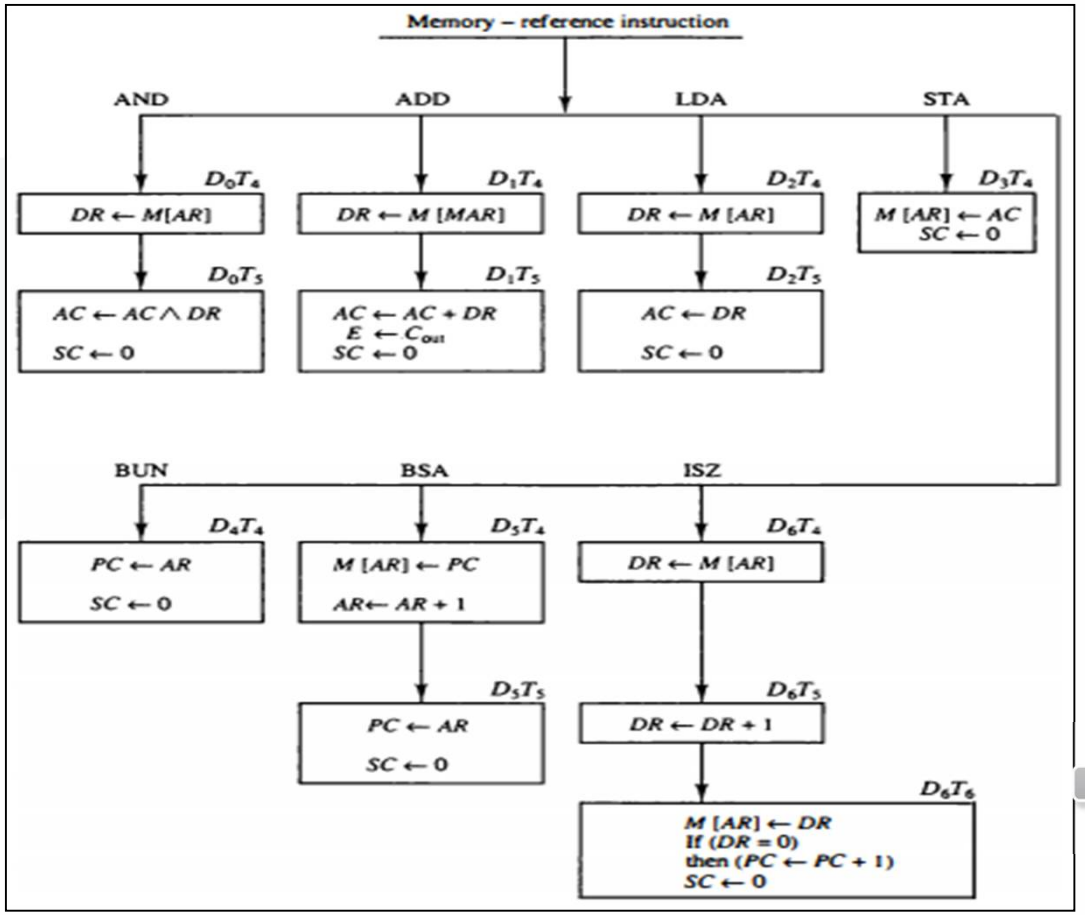


Image source: Google

Parul® University

DIGITAL LEARNING CONTENT

Input-Output instructions

- Input and output instructions are needed for transferring information to and from AC register, for checking the flag bits, and for controlling the interrupt facility.
- Input-output instructions have an operation code 1111 and are recognized by the control when D7 = 1 and I = 1.
- The remaining bits of the instruction specify the particular operation.

Symbol	Symbolic Description	Symbolic Description
INP	AC(0-7) ←INPR, FGI←0	Input data to AC
OUT	OUTR←AC(0-7), FG0 ←0	Output char from AC
SKI	IF FGI=1 then PC←PC+1	Skip on input flag
SKO	IF FGI=0, Then (PC←PC+1)	Skip on output flag
ION	IEN←1	Interrupt enable on
IOF	IEN←0	Interrupt enable off

Addressing Modes:

- 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:
  1. 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.
  2. To reduce the number of bits in the addressing field of the instruction.

Parul® University

DIGITAL LEARNING CONTENT

Addressing Modes:

There are basic 10 addressing modes supported by the computer.

1. Implied Mode
2. Immediate Mode
3. Register Mode
4. Register Indirect Mode
5. Auto-increment or Auto-decrement Mode
6. Direct Address Mode
7. Indirect Address Mode
8. Relative Addressing Mode
9. Indexed Addressing Mode
10. Base Register Addressing Mode

Parul® University

DIGITAL LEARNING CONTENT

Addressing Modes:

1. **Implied Mode**
   - Operands are specified implicitly in the definition of the instruction.
   - For example, the instruction “complement accumulator (CMA)” is an implied-mode instruction because the operand in the accumulator register is implied in the definition of the instruction.
   - All register reference instructions that use an accumulator and zero address instructions are implied mode instructions.
2. **Immediate Mode**
   - Operand is specified in the instruction itself.
   - An immediate-mode instruction has an operand field rather than an address field, like initialization of register.
   - The operand field contains the actual operand to be used in conjunction with the operation specified in the instruction.
   - MOV R1, 05H; This instruction copies immediate number 05H to R1 register.

Addressing Modes:

3. **Register Mode**
   - Operands are in registers that reside within the CPU.
   - The particular register is selected from a register field in the instruction.
   - E.g. MOV AX,BX; This instruction moves value from BX to AX register
4. **Register Indirect 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.
   - E.g. MOV [R1], R2
   - value of R2 is moved to the memory location specified in R1.

Addressing Modes:

5. **Auto-increment or Auto-decrement Mode**
   - This is similar to the register indirect mode expect that the register is incremented or decremented after (or before) its value is used to access memory.
   - When the address stored in the register refers to a table of data in memory, it is necessary to increment or decrement the register after every access to the table. This can be achieved by using the increment or decrement instruction.
6. **Direct Address 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.
   - E.g. ADD 457

Addressing Modes:

7. **Indirect Address Mode**
   - The address field of the instruction gives 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
8. **Relative Address Mode**
   - In this mode the content of the program counter is added to the address part of the instruction in order to obtain the effective address.
   - The address part of the instruction is usually a signed number which can be either positive or negative.
   - Effective address = address part of instruction + content of PC

Addressing Modes:

9. Indexed Addressing Mode

- The content of an index register is added to the address part of the instruction to obtain the effective address.
- The indexed register is a special CPU register that contain an index value.
- The address field of the instruction defines the beginning address of a data array in memory. Each operand in the array is stored in memory relative to the begging address.
- Effective address = address part of instruction + content of index register

Addressing Modes:

10. Base Register Addressing Mode

- The content of a base register is added to the address part of the instruction to obtain the effective address.
- A base register is assumed to hold a base address and the address field of the instruction gives a displacement relative to this base address.
- The base register addressing mode is used in computers to facilitate the relocation of programs in memory.
- Effective address = address part of instruction + content of base register

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set:

Instruction sets are instruction codes to perform some task. It is classified into five categories.

1. Control Instructions
2. Logical Instructions
3. Branching Instructions
4. Arithmetic Instructions
5. Data Transfer Instructions

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Control Instructions

Opcode	Operand	Meaning	Explanation
NOP	None	No operation	No operation is performed, i.e., the instruction is fetched and decoded.
HLT	None	Halt and enter wait state	The CPU finishes executing the current instruction and stops further execution. An interrupt or reset is necessary to exit from the halt state.
DI	None	Disable interrupts	The interrupt enable flip-flop is reset and all the interrupts are disabled except TRAP.
EI	None	Enable interrupts	The interrupt enable flip-flop is set and all the interrupts are enabled.
RIM	None	Read interrupt mask	This instruction is used to read the status of interrupts.
SIM	None	Set interrupt mask	This instruction is used to implement the interrupts.

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Logical Instructions

Opcode	Operand	Meaning	Explanation
CMP	R M	Compare the register or memory with Acc	The contents of the operand (register or memory) are M compared with the contents of the Acc.
CPI	8-bit data	Compare immediate with Acc	The second byte data is compared with the contents of Acc.
ANA	R M	Logical AND register or memory with Acc	The contents of Acc are logically AND with M the contents of the register or memory, and the result is placed in Acc.
ANI	8-bit data	Logical AND immediate with Acc	The contents of Acc are logically AND with the 8-bit data and the result is placed in Acc.
XRA	R M	Exclusive OR register or memory with Acc	The contents of Acc are Exclusive OR with M the contents of the register or memory, and the result is placed in Acc.

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Logical Instructions

Opcode	Operand	Meaning	Explanation
XRI	8-bit data	Exclusive OR immediate with Acc	The contents of Acc are Exclusive OR with the 8-bit data and the result is placed in Acc.
ORA	R M	Logical OR register or memory with Acc	The contents of Acc are logically OR with M the contents of the register or memory, and result is placed in Acc.
ORI	8-bit data	Logical OR immediate with Acc	The contents of the Acc are logically OR with the 8-bit data and the result is placed in Acc.
RLC	None	Rotate Acc left	Each binary bit of the Acc is rotated left by one position. Bit D7 is placed in the position of D0 as well as in CY. CY is modified according to bit D7.
RRC	None	Rotate Acc right	Each binary bit of the Acc is rotated right by one position. Bit D0 is placed in the position of D7 as well as in the CY. CY is modified according to bit D0.

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Logical Instructions

Opcode	Operand	Meaning	Explanation
RAL	None	Rotate Acc left through carry	Each binary bit of the Acc is rotated left by one position through the CY. Bit D7 is placed in CY, and CY is placed in the least significant D0. CY is modified according to D7.
RAR	None	Rotate Acc right through carry	Each binary bit of Acc is rotated right by one position through CY. Bit D0 is placed in CY, & CY is placed in the most significant position D7. CY is modified according to bit D0.
CMA	None	Complement Acc	The contents of Acc are complemented. No flags are affected.
CMC	None	Complement carry	CY is complemented. No other flags are affected.
STC	None	Set Carry	Set Carry

Instruction Set->Branching instructions

Opcode	Operand	Meaning	Explanation
JMP	16-bit address	Jump unconditionally	The program sequence is transferred to the memory address given in the operand.

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Branching instructions

Opcode	Description	Flag status			
JC	Jump on Carry	CY=1	16-bit address	Jump conditionally	The program sequence is transferred to the memory address given in the operand based on the specified flag of the PSW.
JNC	Jump on no Carry	CY=0			
JP	Jump on positive	S=0			
JM	Jump on minus	S=1			
JZ	Jump on zero	Z=1			
JNZ	Jump on no zero	Z=0			
JPE	Jump on parity even	P=1			
JPO	Jump on parity odd	P=0			

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Arithmetic Instructions

Opcode	Operand	Meaning	Explanation
ADD	R M	Add register or memory, to Acc	The contents of the register or memory are added to the contents of Acc and the result is stored in Acc. Example − ADD K.
ADC	R M	Add register to Acc with carry	The contents of the register or memory & M the Carry flag are added to the contents of Acc and the result is stored in Acc. Example − ADC K
ADI	8-bit data	Add the immediate to Acc	The 8-bit data is added to the contents of Acc and the result is stored in Acc. Example − ADI 55K
ACI	8-bit data	Add the immediate to Acc with carry	The 8-bit data and the Carry flag are added to the contents of Acc and the result is stored in Acc. Example − ACI 55K
LXI	Reg. pair, 16bit data	Load the register pair immediate	The instruction stores 16-bit data into the register pair designated in the operand. Example − LXI K, 3025M

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Arithmetic Instructions

Opcode	Operand	Meaning	Explanation
SUB	R M	Subtract the register or the memory from Acc	The contents of the register or the memory are subtracted from the contents of Acc, and the result is stored in Acc. Example − SUB K
SBB	R M	Subtract the source and borrow from Acc	The contents of the register or the memory & M the Borrow flag are subtracted from the contents of Acc and the result is placed in Acc. Example − SBB K
SUI	8-bit data	Subtract the immediate from Acc	The 8-bit data is subtracted from the contents of Acc & the result is stored in Acc. Example − SUI 55K
SBI	8-bit data	Subtract the immediate from Acc with borrow	The contents of register H are exchanged with the contents of register D, and the contents of register L are exchanged with the contents of register E. Example − XCHG
INR	R M	Increment the register or the memory by 1	The contents of the designated register or the memory are incremented by 1 and their result is stored at the same place. Example − INR K

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->Arithmetic Instructions

Opcode	Operand	Meaning	Explanation
INX	R	Increment register pair by 1	The contents of the designated register pair are incremented by 1 and their result is stored at the same place. Example − INX K
DCR	R M	Decrement the register or the memory by 1	The contents of the designated register or memory are decremented by 1 and their result is stored at the same place. Example − DCR K
DCX	R	Decrement the register pair by 1	The contents of the designated register pair are decremented by 1 and their result is stored at the same place. Example − DCX K

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->DATA TRANSFER INSTRUCTION

Opcode	Operand	Meaning	Explanation
MOV	Rd, Sc M, Sc Dt, M	Copy from source (Sc) to destination(Dt)	This instruction copies the contents of the source register into the destination register without any alteration. Example – MOV K, L
MVI	Rd, data M, data	Move immediate 8-bit	The 8-bit data is stored in the destination register or memory. Example – MVI K, 55H
LDA	16-bit address	Load Acc	The contents of a memory location, specified by a 16-bit address in the operand, are copied to Acc. Example – LDA 2034H
LDAX	B/D Reg. pair	Load Acc indirect	The contents of the designated register pair point to a memory location. This instruction copies the contents of that memory location into Acc. Example – LDAX K

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->DATA TRANSFER INSTRUCTION

Opcode	Operand	Meaning	Explanation
LXI	Reg. pair, 16-bit data	Load the register pair immediate	The instruction loads 16-bit data in the register pair designated in the register or the memory. Example – LXI K, 3225H
STA	16-bit address	16-bit address	The contents of Acc are copied into the memory location specified by the operand. This is a 3-byte instruction, the second byte specifies the low-order address and the third byte specifies the high-order address. Example – STA 3251H

Parul® University

DIGITAL LEARNING CONTENT

Instruction Set->DATA TRANSFER INSTRUCTION

Opcode	Operand	Meaning	Explanation
OUT	8-bit port address	Output the data from Acc to a port with 8bit address	The contents of Acc are copied into the I/O port specified by the operand. Example – OUT K9L
IN	8-bit port address	Input data to accumulator from a port with 8-bit address	The contents of the input port designated in the operand are read and loaded into Acc. Example – IN5KL

DIGITAL LEARNING CONTENT

Parul® University

www.paruluniversity.ac.in