**Date: 20-04-2021**

**Branch: CSE (III Year) and EE (III Year)**

**Topic:** Basic Computer Organization and Design Instruction codes,

Computer registers and computer instructions

**Time: 08:00 AM -09:00 AM**

**Basic computer organization and design**

Computer organization refers to the operational unit and their interconnection that realise the architectural specification.

Computer organization deals with how different part of a computer is organized and how various operations are performed between different part to do a specific task.

The organization of the computer is defined by its internal registers, timing and control structure, and set of instruction that is uses.

Instruction & Instruction Codes

Instruction:-

Computer instructions are a set of machine language instructions that a particular processor understands and executes. A computer performs tasks on the basis of the instruction provided.

An instruction comprises of groups called fields. These fields include:

* The Operation code (Opcode) field which specifies the operation to be performed.
* The Address field which contains the location of the operand, i.e., register or memory location.
* The Mode field which specifies how the operand will be located.

Computer Instructions

A basic computer has three instruction code formats which are:

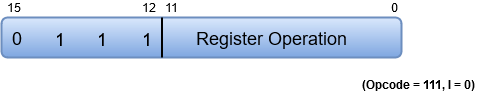
1. Memory – reference instruction
2. Register – reference instruction
3. Input-Output instruction

Memory – reference instruction



In Memory-reference instruction, 12 bits of memory is used to specify an address and one bit to specify the addressing mode ‘I’.

Register – reference instruction

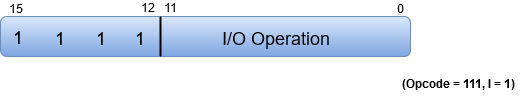


The Register-reference instructions are represented by the Opcode 111 with a 0 in the leftmost bit (bit 15) of the instruction.

Note: The Operation code (Opcode) of an instruction refers to a group of bits that define arithmetic and logic operations such as add, subtract, multiply, shift, and compliment.

A Register-reference instruction specifies an operation on or a test of the AC (Accumulator) register.

Input-Output instruction



Just like the Register-reference instruction, an Input-Output instruction does not need a reference to memory and is recognized by the operation code 111 with a 1 in the leftmost bit of the instruction. The remaining 12 bits are used to specify the type of the input-output operation or test performed.

Note

* The three operation code bits in positions 12 through 14 should be equal to 111. Otherwise, the instruction is a memory-reference type, and the bit in position 15 is taken as the addressing mode I.
* When the three operation code bits are equal to 111, control unit inspects the bit in position 15. If the bit is 0, the instruction is a register-reference type. Otherwise, the instruction is an input-output type having bit 1 at position 15.

**Instruction Set Completeness**

A set of instructions is said to be complete if the computer includes a sufficient number of instructions in each of the following categories:

* Arithmetic, logical and shift instructions
* A set of instructions for moving information to and from memory and processor registers.
* Instructions which controls the program together with instructions that check status conditions.
* Input and Output instructions

Arithmetic, logic 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 the memory unit, but all computations are done in processor registers. Therefore, one must possess the capability of moving information between these two units.

Program control instructions such as branch instructions are used change the sequence in which the program is executed.

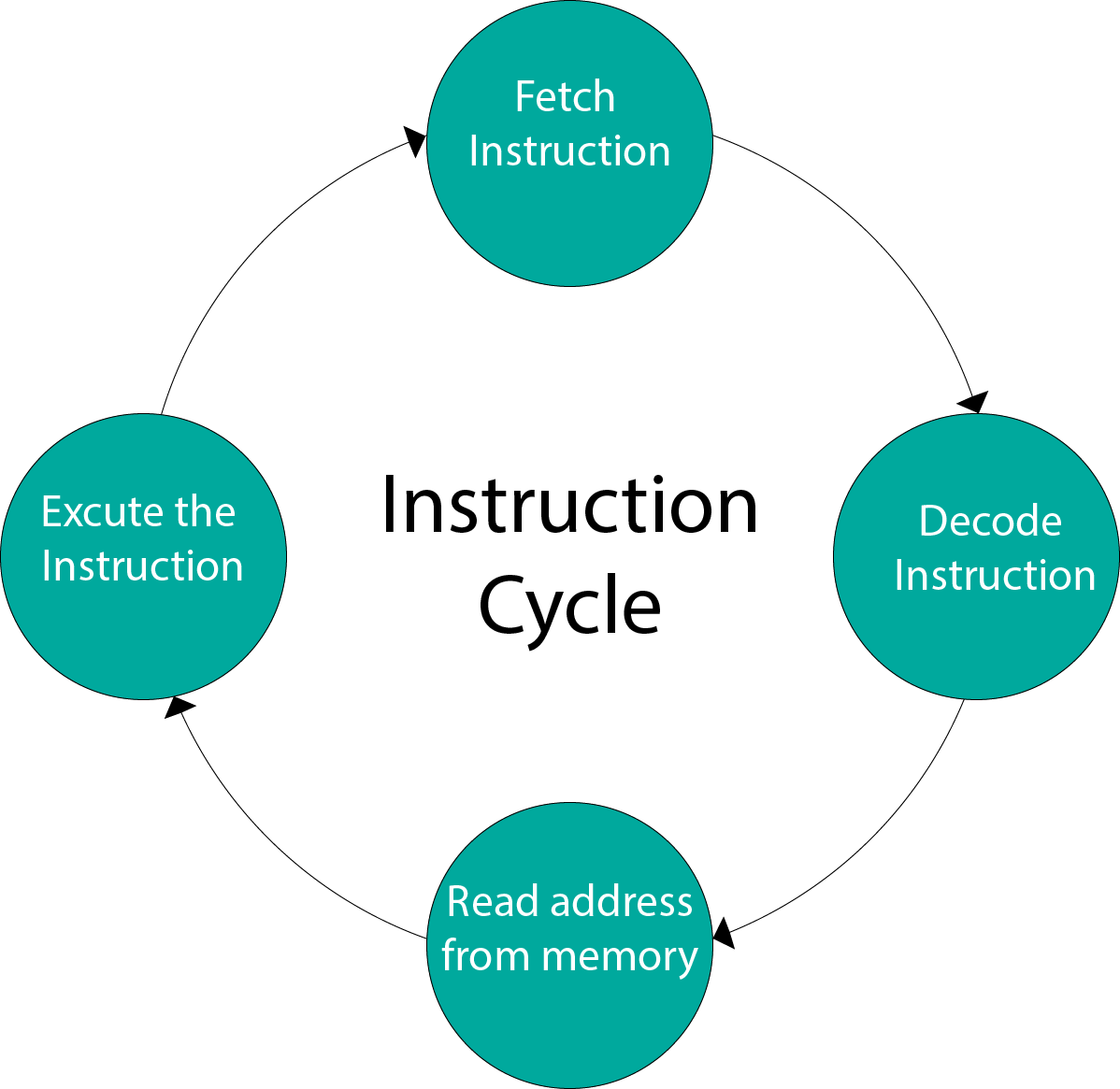
Input and Output instructions act as an interface between the computer and the user. Programs and data must be transferred into memory, and the results of computations must be transferred back to the user.

**Instruction Cycle**

A program residing in the memory unit of a computer consists of a sequence of instructions. These instructions are executed by the processor by going through a cycle for each instruction.

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.

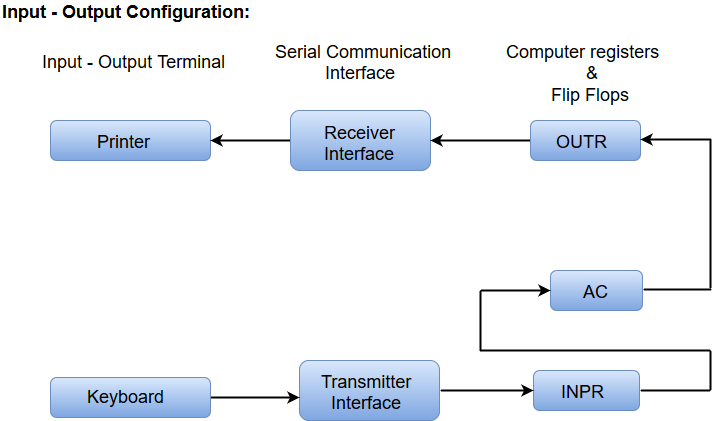


Input-Output Configuration

In computer architecture, input-output devices act as an interface between the machine and the user.

Instructions and data stored in the memory must come from some input device. The results are displayed to the user through some output device.

The following block diagram shows the input-output configuration for a basic computer.



* The input-output terminals send and receive information.
* The amount of information transferred will always have eight bits of an alphanumeric code.
* The information generated through the keyboard is shifted into an input register ‘INPR’.
* The information for the printer is stored in the output register ‘OUTR’.
* Registers INPR and OUTR communicate with a communication interface serially and with the AC in parallel.
* The transmitter interface receives information from the keyboard and transmits it to INPR.
* The receiver interface receives information from OUTR and sends it to the printer serially.

**Design of a Basic Computer**

A basic computer consists of the following hardware components.

1. A memory unit with 4096 words of 16 bits each
2. Registers: AC (Accumulator), DR (Data register), AR (Address register), IR (Instruction register), PC (Program counter), TR (Temporary register), SC (Sequence Counter), INPR (Input register), and OUTR (Output register).
3. Flip-Flops: I, S, E, R, IEN, FGI and FGO

Note: FGI and FGO are corresponding input and output flags which are considered as control flip-flops.

1. Two decoders: a 3 x 8 operation decoder and 4 x 16 timing decoder
2. A 16-bit common bus
3. Control Logic Gates
4. The Logic and Adder circuits connected to the input of AC.

**Computer Registers**

Registers are a type of computer memory used to quickly accept, store, and transfer data and instructions that are being used immediately by the CPU. The registers used by the CPU are often termed as Processor registers.

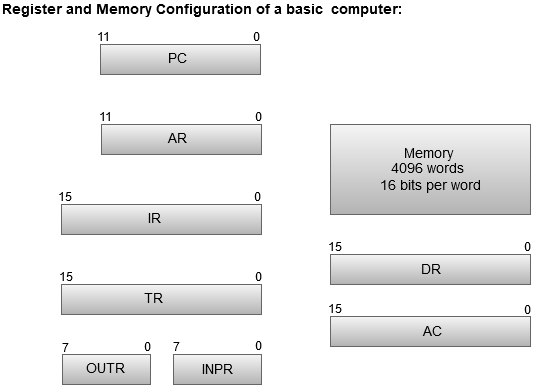
A processor register may hold an instruction, a storage address, or any data (such as bit sequence or individual characters).

The computer needs processor 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 the current instruction is completed.

Following is the list of some of the most common registers used in a basic computer:

|  |  |  |  |
| --- | --- | --- | --- |
| **Register** | **Symbol** | **Number of bits** | **Function** |
| Data register | DR | 16 | Holds memory operand |
| Address register | AR | 12 | Holds address for the memory |
| Accumulator | AC | 16 | Processor register |
| Instruction register | IR | 16 | Holds instruction code |
| Program counter | PC | 12 | Holds address of the instruction |
| Temporary register | TR | 16 | Holds temporary data |
| Input register | INPR | 8 | Carries input character |
| Output register | OUTR | 8 | Carries output character |

The following image shows the register and memory configuration for a basic computer.

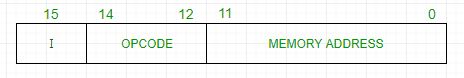


* The Memory unit has a capacity of 4096 words, and each word contains 16 bits.
* The Data Register (DR) contains 16 bits which hold the operand read from the memory location.
* The Memory Address Register (MAR) contains 12 bits which hold the address for the memory location.
* The Program Counter (PC)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

Computer Instructions

The basic computer has 16-bit instruction register (IR) which can denote either memory reference or register reference or input-output instruction.

1. **Memory Reference –** These instructions refer to memory address as an operand. The other operand is always accumulator. Specifies 12-bit address, 3-bit opcode (other than 111) and 1-bit addressing mode for direct and indirect addressing.

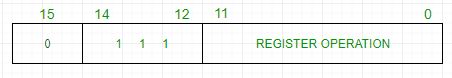


**Example –**  
IR register contains = 0001XXXXXXXXXXXX, i.e. ADD after fetching and decoding of instruction we find out that it is a memory reference instruction for ADD operation.

Hence, DR ← M[AR]

AC ← AC + DR, SC ← 0

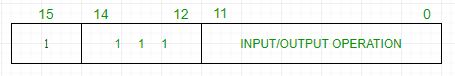
1. **Register Reference –** These instructions perform operations on registers rather than memory addresses. The IR(14 – 12) is 111 (differentiates it from memory reference) and IR(15) is 0 (differentiates it from input/output instructions). The rest 12 bits specify register operation.



**Example –**  
IR register contains = 0111001000000000, i.e. CMA after fetch and decode cycle we find out that it is a register reference instruction for complement accumulator.

Hence, AC ← ~AC

1. **Input/Output –** These instructions are for communication between computer and outside environment. The IR(14 – 12) is 111 (differentiates it from memory reference) and IR(15) is 1 (differentiates it from register reference instructions). The rest 12 bits specify I/O operation.



**Example –**  
IR register contains = 1111100000000000, i.e. INP after fetch and decode cycle we find out that it is an input/output instruction for inputing character. Hence, INPUT character from peripheral device.

The set of instructions incorporated in16 bit IR register are:

1. Arithmetic, logical and shift instructions (and, add, complement, circulate left, right, etc)
2. To move information to and from memory (store the accumulator, load the accumulator)
3. Program control instructions with status conditions (branch, skip)
4. Input output instructions (input character, output character)

| Symbol | Hexadecimal Code | | 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 AC content in memory |
| BUN | 4xxx | Cxxx | Branch Unconditionally |
| BSA | 5xxx | Dxxx | Branch and Save Return Address |
| ISZ | 6xxx | Exxx | Increment and skip if 0 |
| CLA | 7800 | | Clear AC |
| CLE | 7400 | | Clear E(overflow bit) |
| 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 > 0 |
| SNA | 7008 | | Skip next instruction if AC < 0 |
| SZA | 7004 | | Skip next instruction if AC = 0 |
| SZE | 7002 | | Skip next instruction if E = 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 |