CSA Notes

* Register Transfer and Micro Operations :-

Introduction to Registers :-

- Registers are used to store data temporarily during the execution of Program.
- It is also called temporarily storage unit with high speed.
- Registers are group of flip-flops, where each flip-flip is capable of storing one bit of information.
- Generally, there are two types of registers in CPU :-
  - (i) General purpose registers.
  - (ii) Special purpose registers.

(i) General Purpose registers :- are used for multiple purposes and assigned to variety of functions by the Programmer.

(ii) Special Purpose Register :- Special Purpose Registers are restricted to only specific function. They are used by the microprocessor itself.

→ Accumulator is special purpose register of the ALU of CPU, which contains one of the operands, and stores result of most arithmetic and logical operations.

→ Another Special purpose registers also include PC, IR, MAR, MDR.

(i) PC :- Program Counter (PC) is used to hold the address of the next instruction to be executed.

(ii) IR :- Instruction Register (IR) is used to hold the instruction that is currently being executed.

(iii) MAR :- Memory Address registers (MAR) holds the address of main memory to or from which data is to be transferred.

(iv) MDR :- Memory data register (MDR) contains the data to be written into memory or read from memory.

* Instruction Format :-

- Instruction is a command given to a computer to perform a specified operation on some given data.
- The format in which the Instruction is specified is known as Instruction format.

→ The important fields of an instruction format are :-

- Opcode field :- It specifies which type of operation is to be performed.
- Address field :- It contains the location of the operand i.e. register or memory location.
- mode field :- It specifies how the operand will be located.

Mode	Opcode	Address of operand

* Types of Instructions

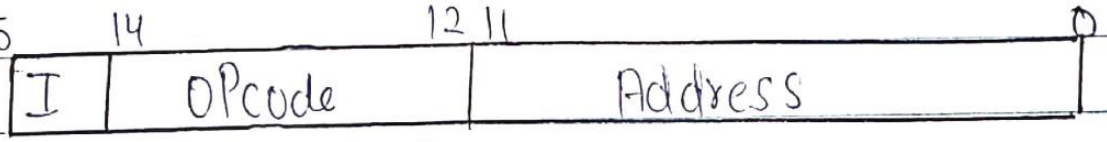
The basic computer has three types of instruction :-

1. Memory Reference Instructions.
2. Register Reference Instructions
3) Input/Output Instructions.

} 16 bit instruction code

1. Memory Reference Instructions :-

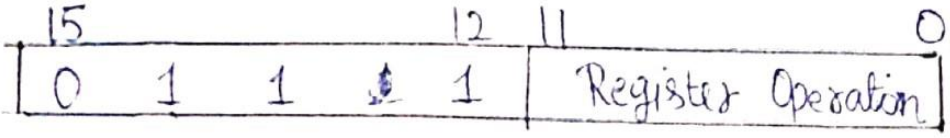
- These instructions refer to memory address as an operand.
- It specifies 12-bit address, 3-bit opcode and 1-bit addressing mode I.



→ Bits (0-11) specify an address.
→ Bits (12-14) Specify operation code (Operan
→ Bit(15) Specify the addressing mode I.
I = 0 for direct address.
I = 1 for indirect address.

2. Register Reference Instructions :-

- Register reference instructions are those instructions in which the operand to be operated upon is present in a register.
- It does not need a reference to memory; therefore Bits (0-11) are used to specify the register operation.
- The register-reference instructions are represented by the Opcode 111 with a 0 in the left most bit and rest of 12 bits represent the operation to be performed.



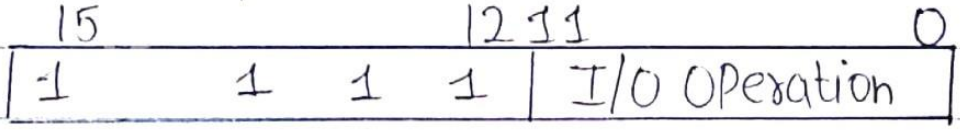
- In Register Reference Instruction :-

Bits (0-11) specify the register operation.
Bits (12-14) equals to 111.
Bits (15) is 0.
Bits (15-12) of an instruction are always 0111 which is equal to hexadecimal 7.

3. Input - Output Instructions :-

- Just like the Register - reference instruction, an input - Output instruction does not need a reference to memory and it 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 I/O Operations or test performed.



→ (The Instruction code format of I/O instruction)

→ First 12 bits (0-11) specify the I/O operation.
→ The next three bits equal to 111 specify opcode.
→ The last mode bit of the instruction is 1.
→ Bits (15-12) of an instruction are always 1111 which is equal to hexadecimal F.