Introduction: Instruction Format

- Instruction word should have the complete information required to fetch and execute the instruction

- Fields of an instruction word
  - Opcode of the operation to be carried out
    - Varying length (CISC)
    - Fixed length (RISC)
  - Size of the operands:
    - Byte, Word, Longword, Quadword for integer operands
    - Float, Double for real operands
  - Addressing mode (AM) of each operand
  - Specification of each operand involves specifying one or more of the following:
    - General purpose register
    - Value of an immediate operand
    - Address of operand
    - Base register
    - Index register
    - Displacement

Instruction Representation

- 3-operand CISC instruction format:

ADD dst, src1, src2

Opcode	Size of operands	AM of dst	Specification of dst	AM of src1	Specification of src1	AM of src2	Specification of src2

Opcode	Size of operands	AM of dst	Specification of dst	AM of src1	Specification of src1	AM of src2	Specification of src2

0011	01	000	00011	101	00001	00000	110	0011 0010	00010

Instruction Representation

- Examples of RISC instructions:

ADD.w R2, R0, R1

Opcode	Size of operands	Specification of dst	Specification of src1	Specification of src2
000111	01	00010	00000	00001

LOAD.w R2, [R1][R0]

Opcode	Size of operands	Specification of dst	AM of src	Specification of src	
010011	01	00010	101	00001	00000

Registers

- The **registers set** stores intermediate data used during the execution of the instructions

- **Processor register** is a small amount of very fast computer memory used to speed the execution of computer programs by providing quick access to commonly used values—typically, the values being in the midst of a calculation at a given point in time. (AC)

- **Data registers** are used to store integer numbers (DR)

- **Address registers** hold memory addresses and are used to access memory. (AR)

- **General Purpose registers** (**GPR**s) can store both data and addresses, i.e., they are combined Data/Address registers

- **Floating Point registers** (**FPR**s) are used to store floating pointComputer arithmetic

- **Constant registers** hold read-only values (e.g., zero, one, pi, ...).

- **Vector registers** hold data for vector processing

- **Special Purpose registers** store internal CPU data, like the program counter which indicates where the computer is in its instruction sequence

- **Control Registers** which ctrl the general behavior of the CPU

- Program counter: Holds address for instruction (i.e. address of the next instruction after execution of the current instruction is completed (PC)
- Instruction Register: holds the instruction code. (IR)
- Temporary register: holds temporary data (TR)
- Input register: holds input character (INPR)
- Output registers: holds output character (OUTR)

Instruction codes

- The **internal organization of a digital system** is defined by the **sequence of micro operations** it performs on data stored in its registers.

- **Digital computer** is capable of executing various micro operations & can be instructed as to **what sequence of operations it must perform.**

- The **user of a computer can control** the process by means of a program.

- A **program is a set of instructions** that specify the operations, operands, and the sequence by **which processing has to occur**.

- The **data processing task maybe altered** by specifying a new program with **different instructions** or specifying the same instructions with **different data**.

- A **computer instruction is a binary code** that specifies a sequence of micro operations for the computer. **Instructions codes together with data are stored in memory.**

- The **computer reads each instruction from memory** and 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 micro operations** .

- Every **computer has its own instruction set**. The ability to store and execute , the stored program concept, is the most important property of a general purpose computer.

- **An instruction code is a group of bits that instruct the computer to perform a specific operation.** It usually divided into two parts, each having its own particular interpretation.

- The most basic part of an instruction code is its **operation part**. The operation code of an instruction is a group of bits that **define such operations as add, subtract, multiply, shift and complement.**

- As an illustration, consider a **computer with 64 distinct operations**. One of them being an **ADD operation**. When this **operation code is decoded in the control unit**, the computer **issues control signals to read an operand from memory and add the operand to a processor register.**

- The relationship between a computer operation and a micro operation. **An operation is a part of an instruction** stored in computer memory. It is a binary code that **tells the computer to perform a specific operation.**

- The control unit receives the instruction from memory and interprets the computer code bits.

- **It then issues a sequence of control signals to initiate micro operations in internal computer registers.**

- ***For every operation code, the control issues a sequence of micro operations needed for the hardware implementation of the specified operation.***

- For this reason, an operation code is sometimes called a macro operations because it specifies a set of micro operations.
- The operation part of an instruction code specifies the operation to be performed. This operation must be performed on some data stored in processor

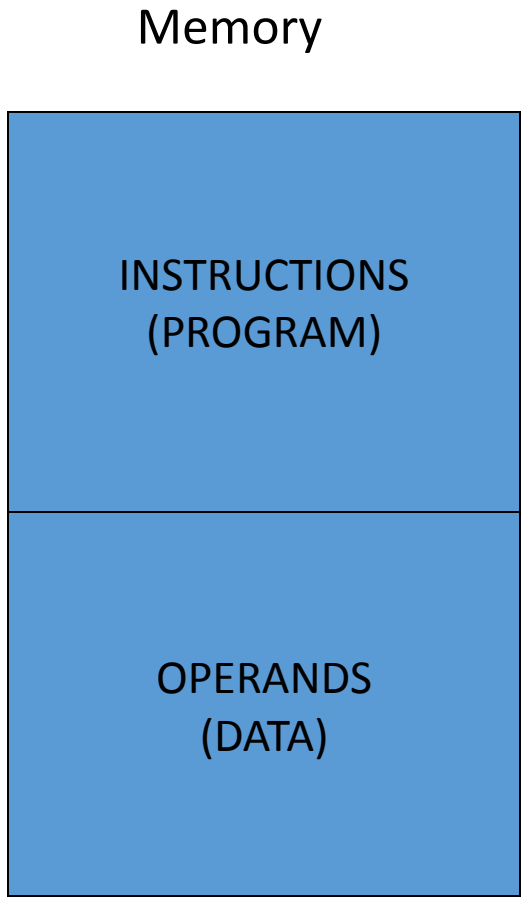
- Instruction Code specifies operation and registers where the operands are to be found.

- Instruction Code format with two parts

- Opcode specifies the operation to be performed

- Address tells the control where to find an operand in memory.

Instruction Format



STORED PROGRAM ORGANIZATION

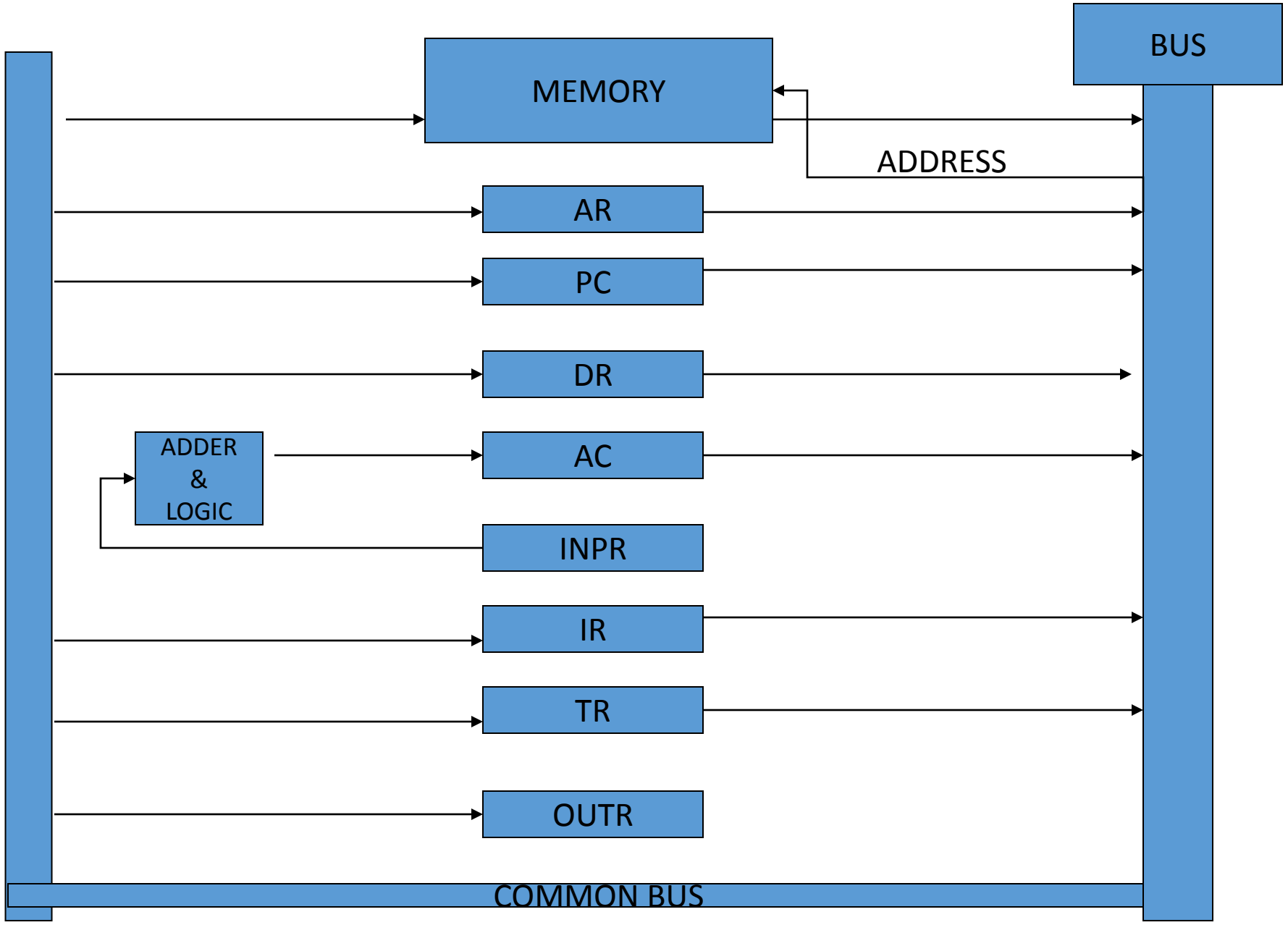
COMMON BUS SYSTEM

- BASIC COMPUTER HAS
  - **EIGHT REGISTERS**
  - **MEMORY UNIT**
  - **CONTROL UNIT**

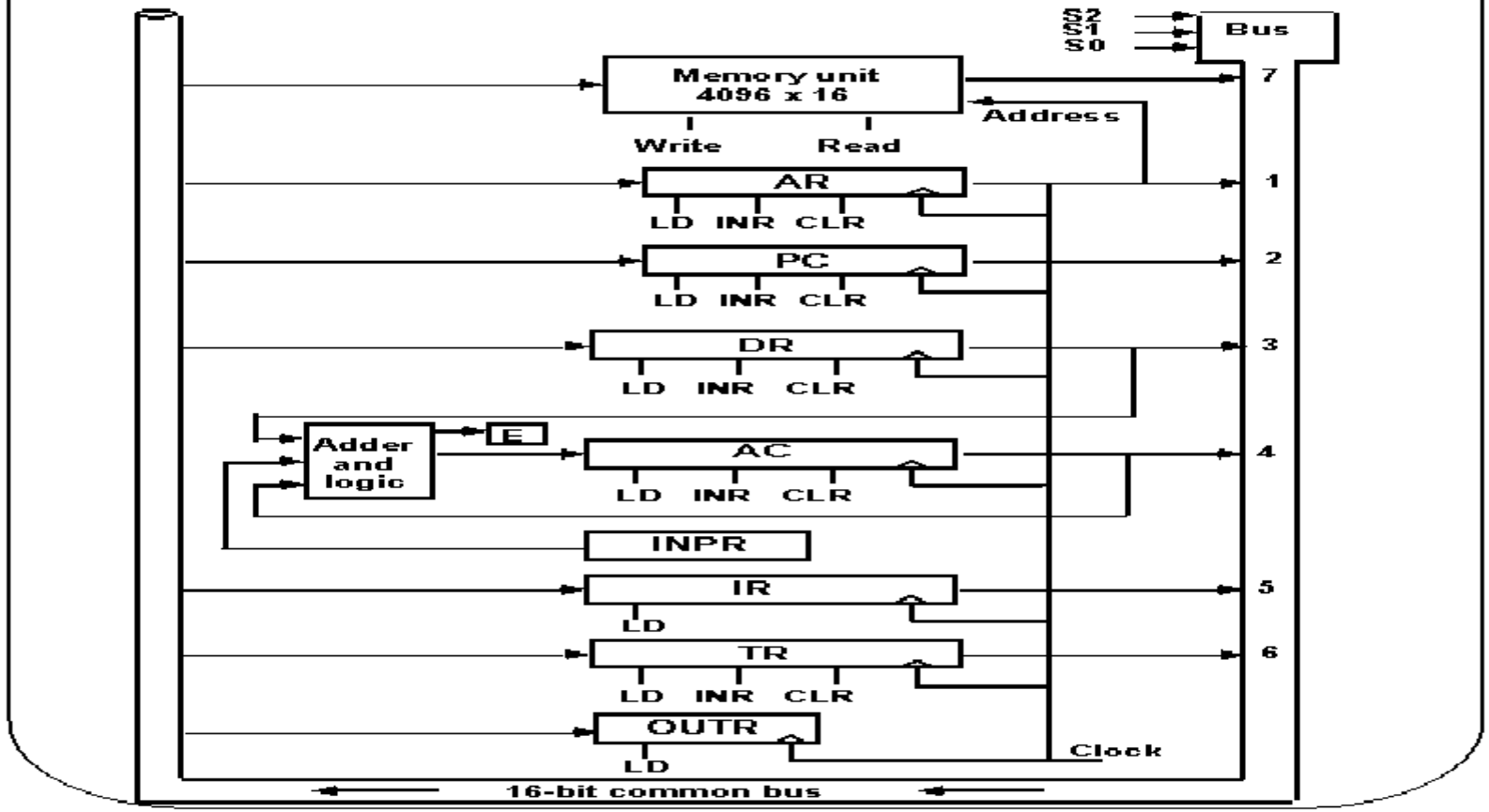
- Path must be provided to transfer information from one register to another and between memory and registers.

- The number of wires will be excessive if connections are made between the o/p of each register and i/p of other registers.

- A more efficient scheme of transferring the information in a system with many registers is to use a common bus


BUS
MEMORY
ADDRESS
AR
PC
DR
ADDER
&
LOGIC
AC
INPR
IR
TR
OUTR
COMMON BUS

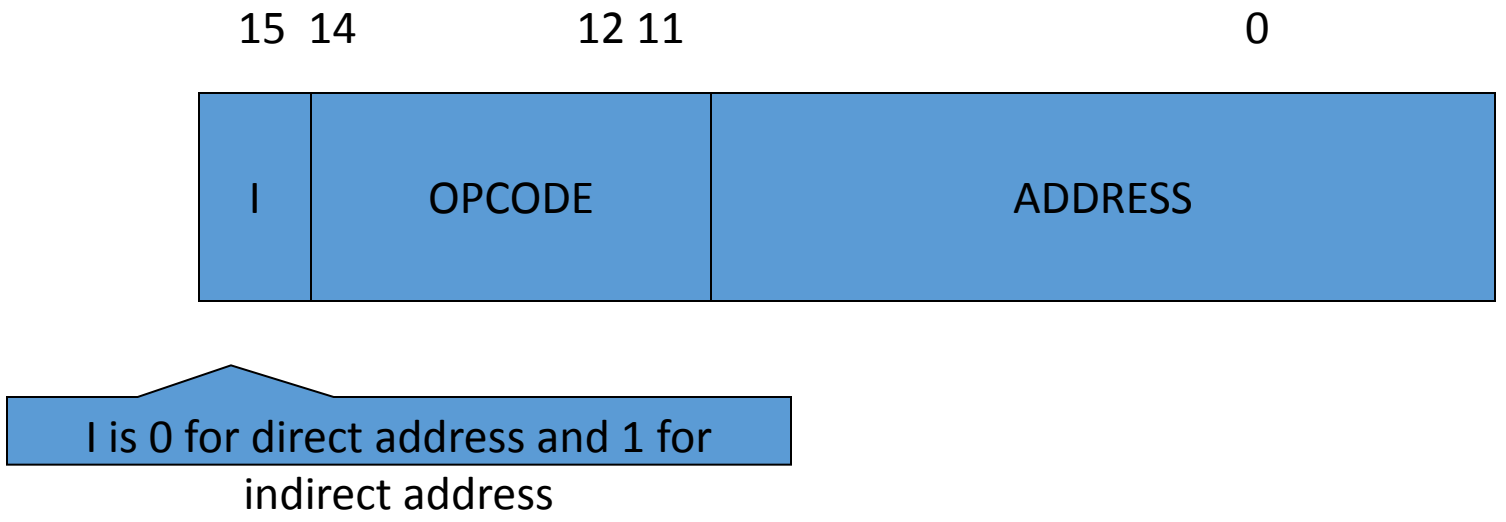
COMMON BUS SYSTEM



Computer Instructions

- Computer instruction code format has 16 bits

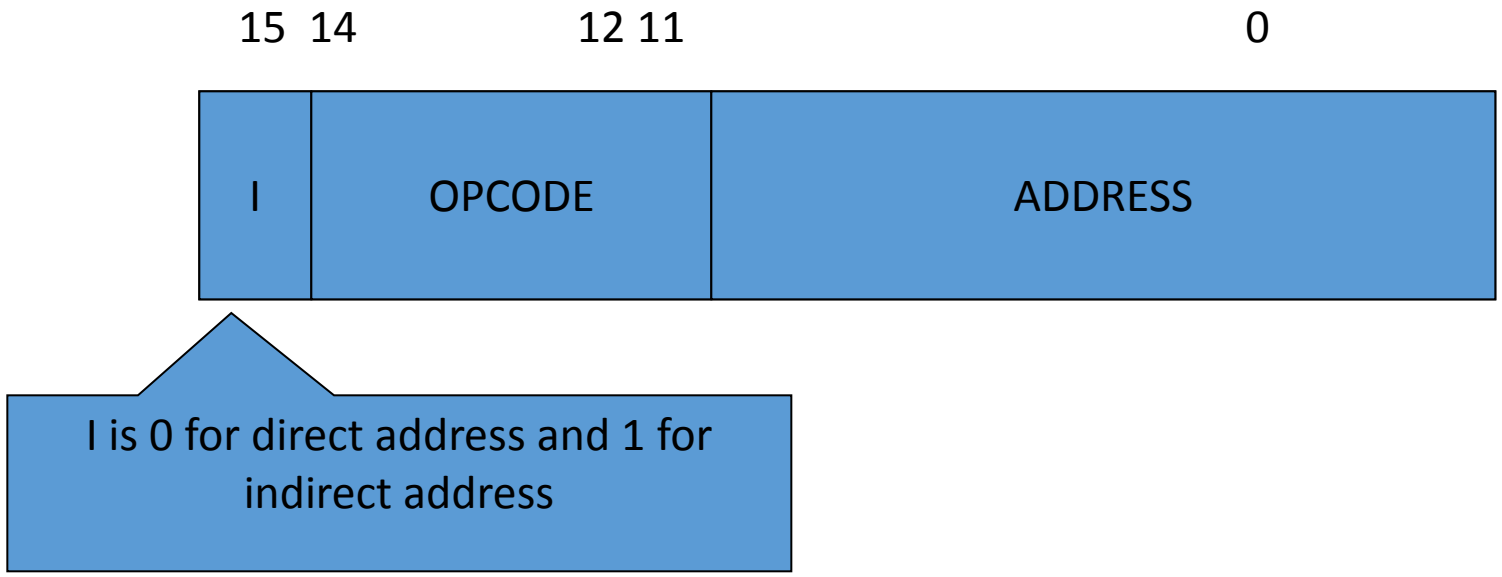
- OPCODE : part of the instruction contains three bits and the meaning of the 13 bits depends upon the operation code encountered.



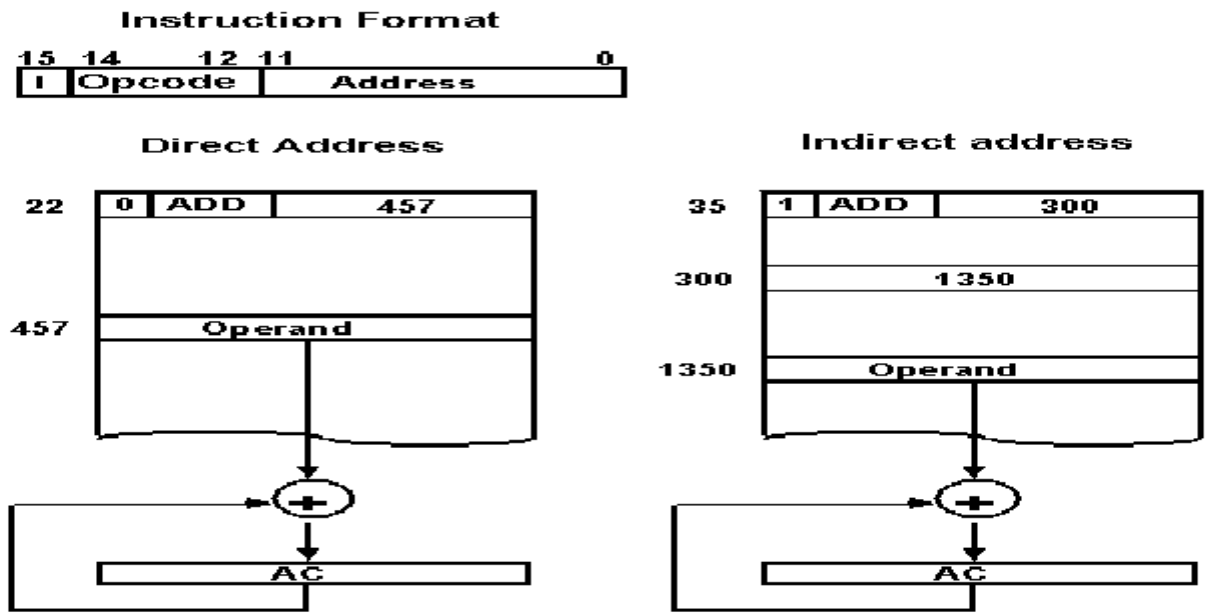
Computer instruction are of three types

- Memory reference instruction
- Register reference instruction
- Input-output instruction

Memory reference instruction



INDIRECT ADDRESS



Effective Address(EFA, EA)

The address, that can be directly used without modification to access an operand for a computation-type instruction, or as the target address for a branch-type instruction

Register reference instruction

- They are recognized by the OPCOde 111 and 0 with the left most bit
- The other 12 bits specifies the operation.
- Register reference instruction specifies operation on register.
- So, does not need any reference to memory

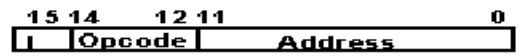
Input-output instruction

- They are recognized by the OPCOde 111 and 1 with the left most bit
- The other 12 bits specifies the operation.
- Input-output instruction does not need any reference to memory

COMPUTER(BC) INSTRUCTIONS

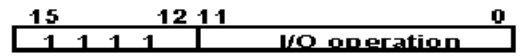
Basic Computer Instruction code format

Memory-Reference Instructions(OP-code = 000 ~ 110)



Register-Reference Instructions(OP-code = 111, I = 0)

Input-Output Instructions (OP-code =111, I = 1)



Assignment

Explain Instruction set format.
What do you mean by register.