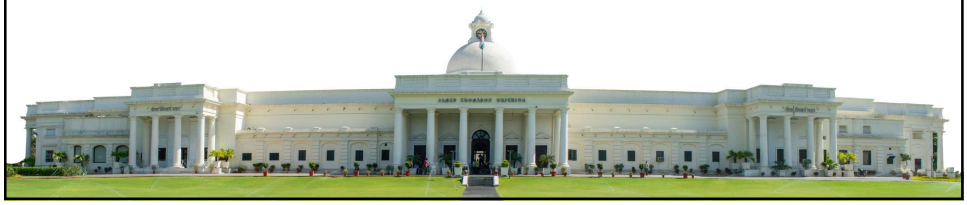
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

System Software
CSN-252
Introduction



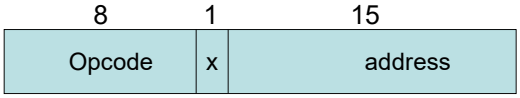
Simplified Instructional Computer (SIC)

- **Memory**
  - Consists of 8-bit bytes
  - Word (24 bits)
  - All addresses are byte addresses
  - Words are addressed by the location of their lowest numbered byte
  - Total 2^{15} bytes in computer memory

- **Registers**
  - 5 registers (24 bit)

Mnemonic	Number	Use
A	0	Accumulator
X	1	Index register
L	2	Linkage register;
PC	8	Program Counter
SW	9	Status Word

- **Data Formats**
  - Integers are stored as 24-bit binary numbers
  - 2's complement representation is used for negative values
  - Characters are stored using their 8-bit ASCII codes
  - No floating point hardware
- **Instruction Formats**



- **Addressing Modes**

  Two addressing modes indicated by the setting of the x bit in the instruction.

Direct	TA = address
Indexed	TA = address + (X)

- **Instruction Set**
  - load and store registers (LDA, LDX etc.)

    LDA m
  - integer arithmetic operations (ADD, SUB etc.) – all arithmetic operations involve register A and a word in memory, with the result being left in the register.

    ADD m
  - Instruction COMP – compares the value in register A with a word in memory and sets a condition code CC (6-7 bits)
  - Conditional jump instructions (JLT, JEQ etc.) can test CC and jump accordingly.
  - JSUB – places the return address in register L; RSUB

    JSUB m

- **Input and Output**
  - Input and output are performed by transferring 1 byte at a time to or from the rightmost 8 bits of register A.
  - Each device is assigned a unique 8-bit code.
  - Three I/O instructions (TD, RD and WD)

    RD m — A [rightmost byte] ← data from device specified by (m)

  - A program that needs to transfer data must wait until the device is ready then execute RD or WD
  - < :device is ready
  - = :device is not ready

Programming

```
            LDA     FIVE    0
            STA     ALPHA   3
            LDCH    CHARZ   6
            STCH    C1      9
            :
ALPHA       RESW    1       c
FIVE        WORD    5       f
CHARZ       BYTE    C'Z'    12
C1          RESB    1       13
```

- Space needed ?
- Time needed ?
- Memory references ?

Defining Storage

- **RESW** Reserves one or more word of storage
- **WORD** Reserves one word of storage which is initialized to a value defined in the operand field
- **RESB**
- **BYTE** Reserves one or more byte initialized to …