

Unit-I

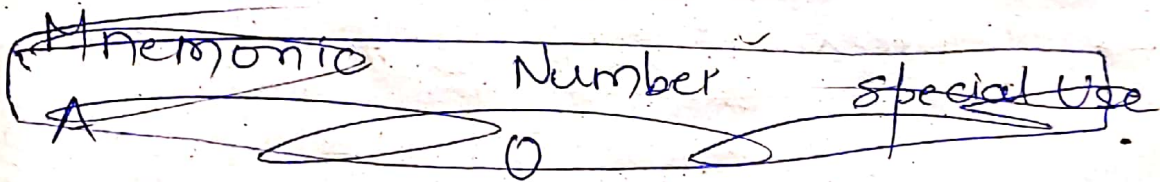
Q.) Discuss the SIC machine Architecture.

→ Memory:

Memory consists of 8-bit bytes. ~~any~~
All addresses on SIC are byte addresses. Words are ~~associated~~ located by the address of their lowest numbered byte. There are 2^{15} byte in computer memory.

→ Registers:

Five registers, each 24 bits in length.



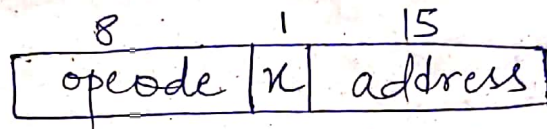
Mnemonic	No.	Special Use
A	0	Accumulator, used for arithmetic op ⁿ
X	1	Index Register, used for addressing
L	2	Linkage register; the jump to subroutine (JSUB) instruction stores the return address in this register
PC	8	Program counter; contains the address of next inst ⁿ to be fetched for execution
SW	9	Status word; contains a variety of info, including a Cond ⁿ code (cc)

→ Data formats

Integers are stored as 24 bit binary numbers. 2's complement^{rep.} is used for negative values. Characters are stored using their 8-bit ASCII codes.

There is no floating point hardware on standard version of SIC.

→ Instruction formats



flag bit n is used to indicate indexed-addressing mode.

→ Addressing Modes

Mode	Indication	Target add. Calc ⁿ
Direct	X = 0	TA = address
Indexed	X = 1	TA = address + (X)

→ Instrⁿ set

(LDA, LD_X, STA, ST_X etc) basic task.

integer arithmetic ~~to~~ opⁿ (ADD, SUB, MUL, DIV.)

All arithmetic operations involve register A & a word in memory, with the result being left in the register.

→ Input & Output

On the standard version of SIC, input & 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. There are 3 three I/O instrⁿ, each of which specifies the device code as an operand

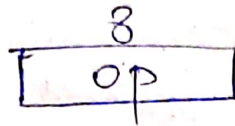
Test Device (TD) instrⁿ tests wheather the addressed device is ready to send or receive a byte of data.

⊙ CC is set to indicate the result of this test.

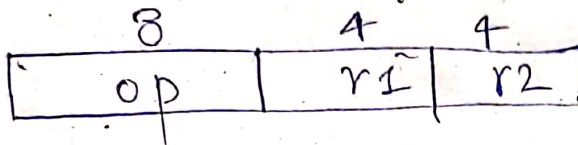
A program needing to transfer data must

Q.) Instrⁿ format of SIC/XE

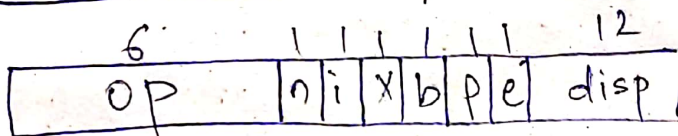
→ Format 1 (1 byte):



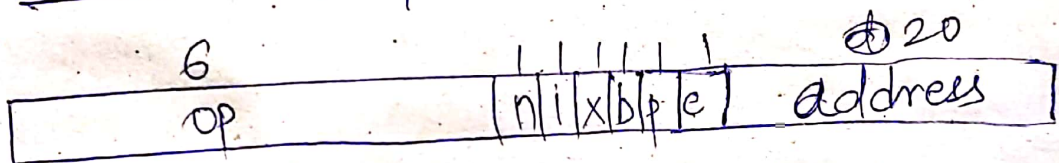
Format 2 (2 bytes):



Format 3 (3 bytes):



Format 4 (4 bytes):



Q.) Addressing modes

Two more addressing modes are available for use ~~with instructions~~

Mode	Indication	Target address calc ⁿ
Base relative	$b=1, p=0$	$TA = (B) + disp \quad (0 \leq disp < 4095)$
Program-counter relative	$b=0, p=1$	$TA = (PC) + disp \quad (-2048 \leq disp < 2047)$

Q.) Discuss first pass

→ first pass does little more than scan the source program for label definitions & assign addresses

→ General description of the functions
Pass 1 (define symbols):

- 1.) Assign addresses to all statements in the ~~to~~ program.
- 2.) Save the values (addresses) assigned to all labels for use in ~~label 2~~ pass 2.
- 3.) Perform some processing of assembler directives.

Pass 2 (assemble instrⁿ & generate obj. program):

- 1.) Assemble instructions
- 2.) Generate data values defined by BYTE, WORD & etc.
- 3.) Perform processing of assembler directives if not done during Pass 1.
- 4.) Write the object program & the assembly listing.