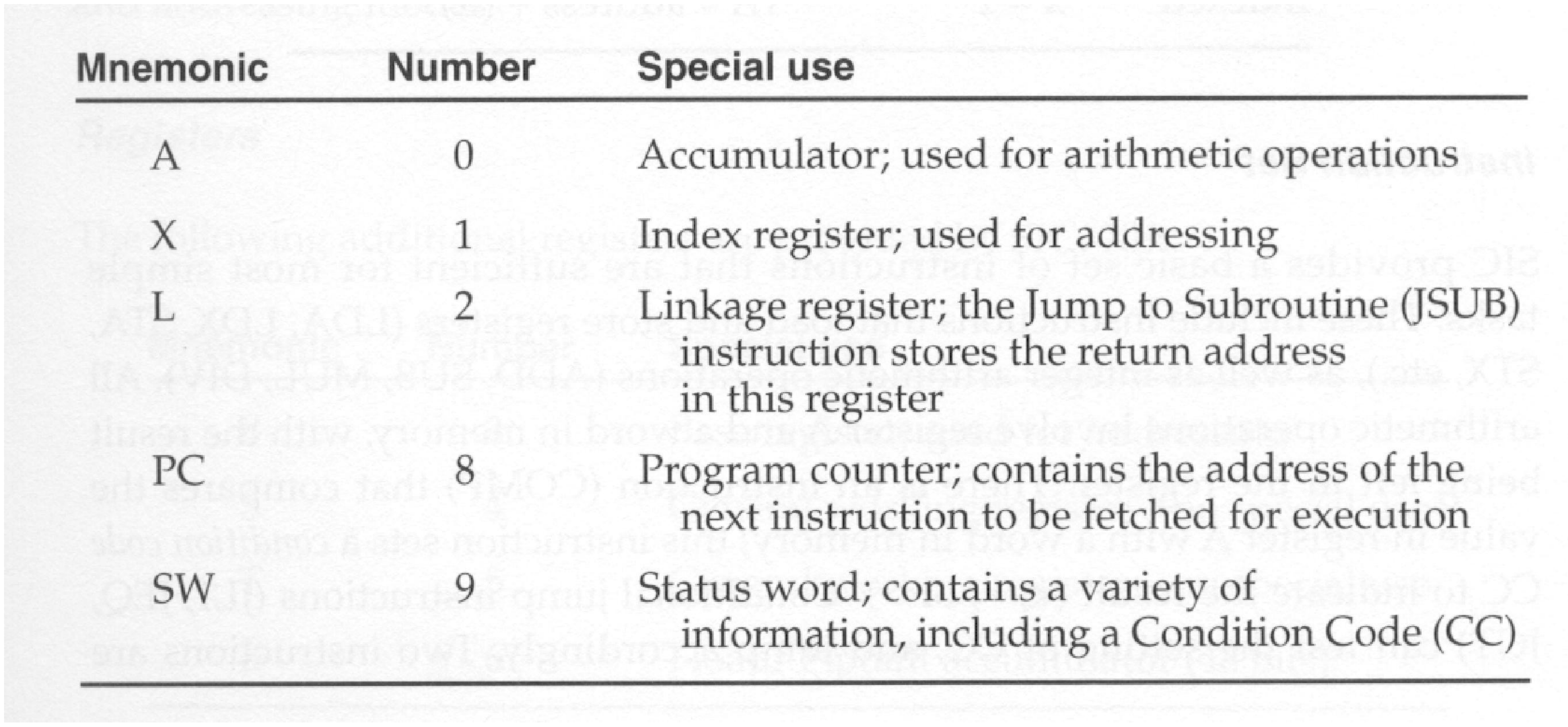
Simplified Instructional Computer (SIC)

SIC Architecture

- Two versions: SIC and SIC/XE (extra equipments). SIC program can be executed on SIC/XE.
- Memory consists of 8-bit bytes. 3 consecutive bytes form a word (24 bits)
- In total, there are 2^15 bytes in the memory.
- There are 5 registers. Each is 24 bits in length.

Five Registers



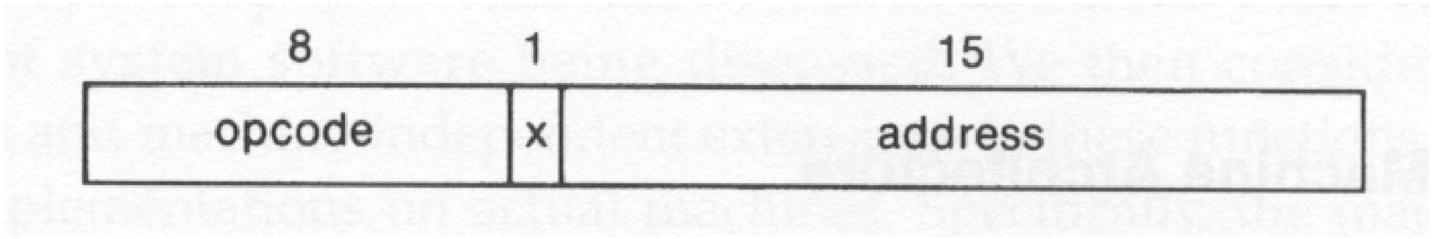
Mnemonic	Number	Special use
A	0	Accumulator; used for arithmetic operations
X	1	Index register; used for addressing
L	2	Linkage register; the Jump to Subroutine (JSUB) instruction stores the return address in this register
PC	8	Program counter; contains the address of the next instruction to be fetched for execution
SW	9	Status word; contains a variety of information, including a Condition Code (CC)

Data Format

- Integers are stored as 24-bit binary numbers; 2’s complement representation is used for negative numbers.
- Characters are store using their 8-bit ASCII codes.
- There is no floating-point hardware on SIC.

Instruction Format

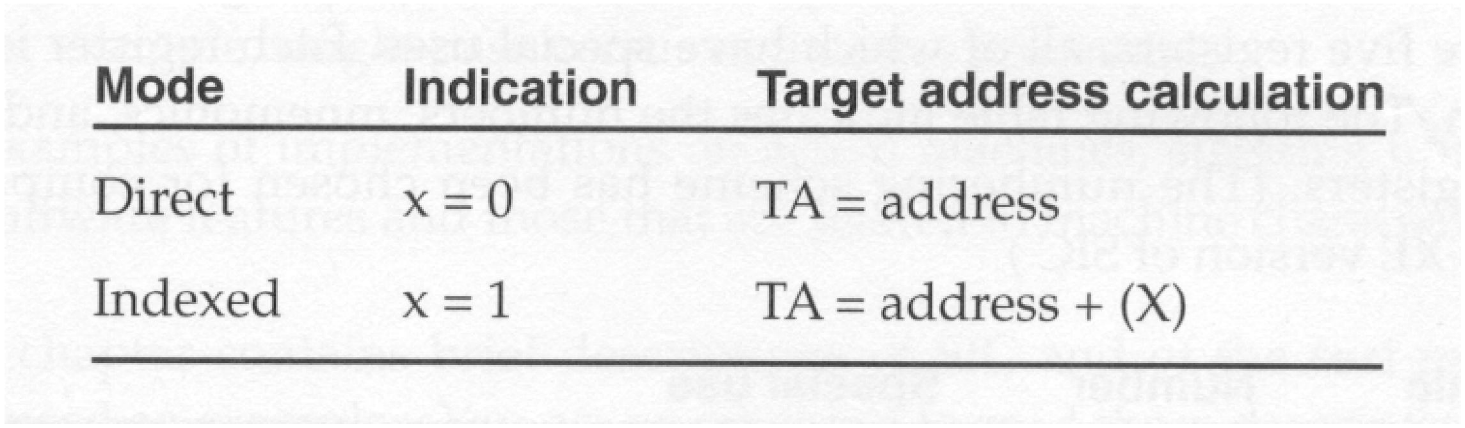
- All machine instructions on SIC has the following 24-bit format.



X is used to indicate indexed-addressing mode.

Addressing Modes

- Only two modes are supported:
  - Direct
  - Indexed



Mode	Indication	Target address calculation
Direct	x = 0	TA = address
Indexed	x = 1	TA = address + (X)

() are used to indicate the content of a register.

Instruction Set

- Load and store registers (LDA, LDX, STA, STX)
- Integer arithmetic (ADD, SUB, MUL, DIV), all involve register A and a word in memory.
- Comparison (COMP), involve register A and a word in memory.
- Conditional jump (JLE, JEQ, JGT, etc.)
- Subroutine linkage (JSUB, RSUB)

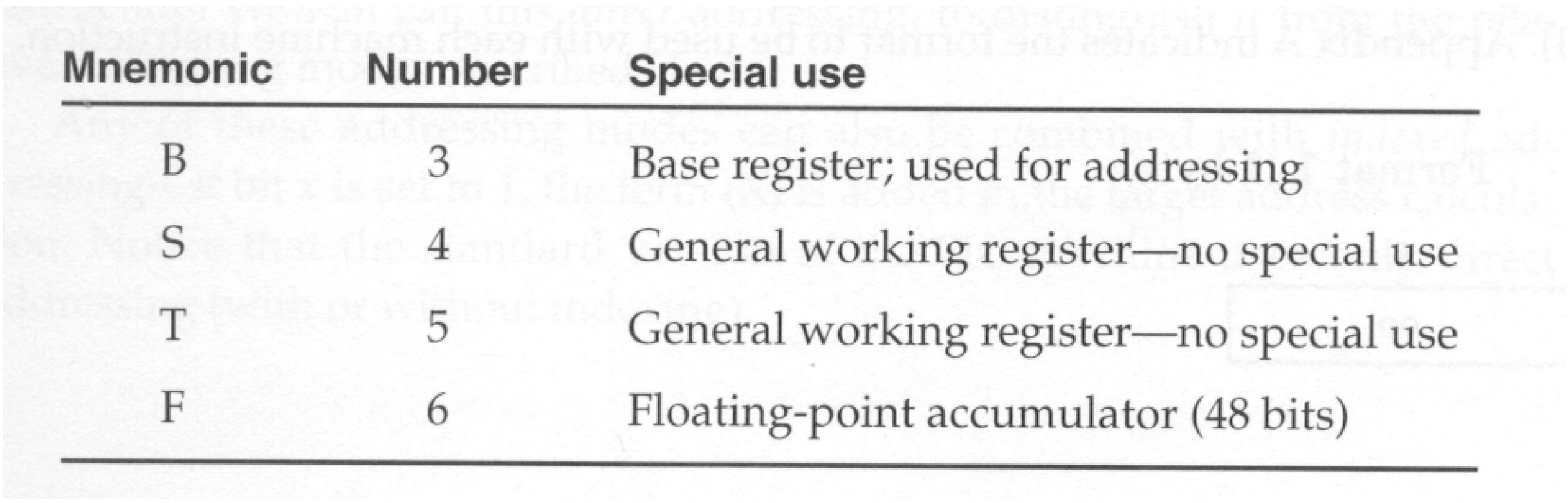
Input and Output

- One byte at a time to or from the rightmost 8 bits of register A.
- Each device has a unique 8-bit ID code.
- Test device (TD): test if a device is ready to send or receive a byte of data.
- Read data (RD): read a byte from the device to register A
- Write data (WD): write a byte from register A to the device.

SIC/XE Architecture

- Memory: 1 megabytes (2^20 bytes)

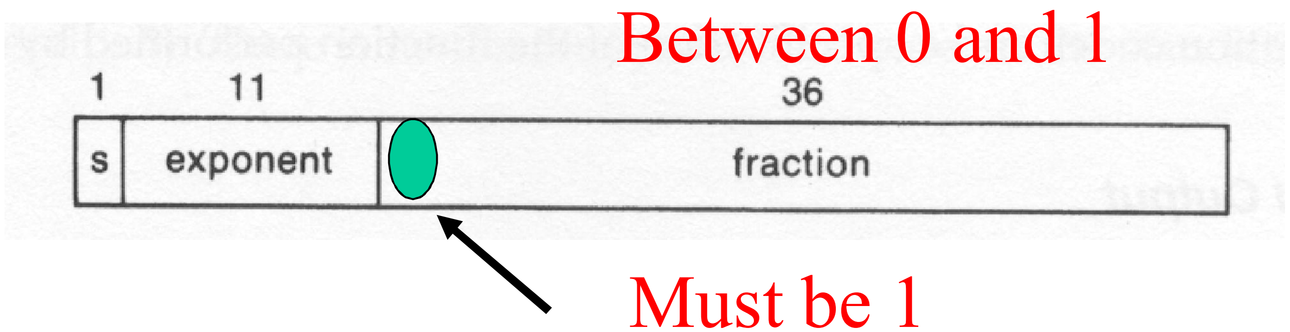
Additional Registers



Mnemonic	Number	Special use
B	3	Base register; used for addressing
S	4	General working register—no special use
T	5	General working register—no special use
F	6	Floating-point accumulator (48 bits)

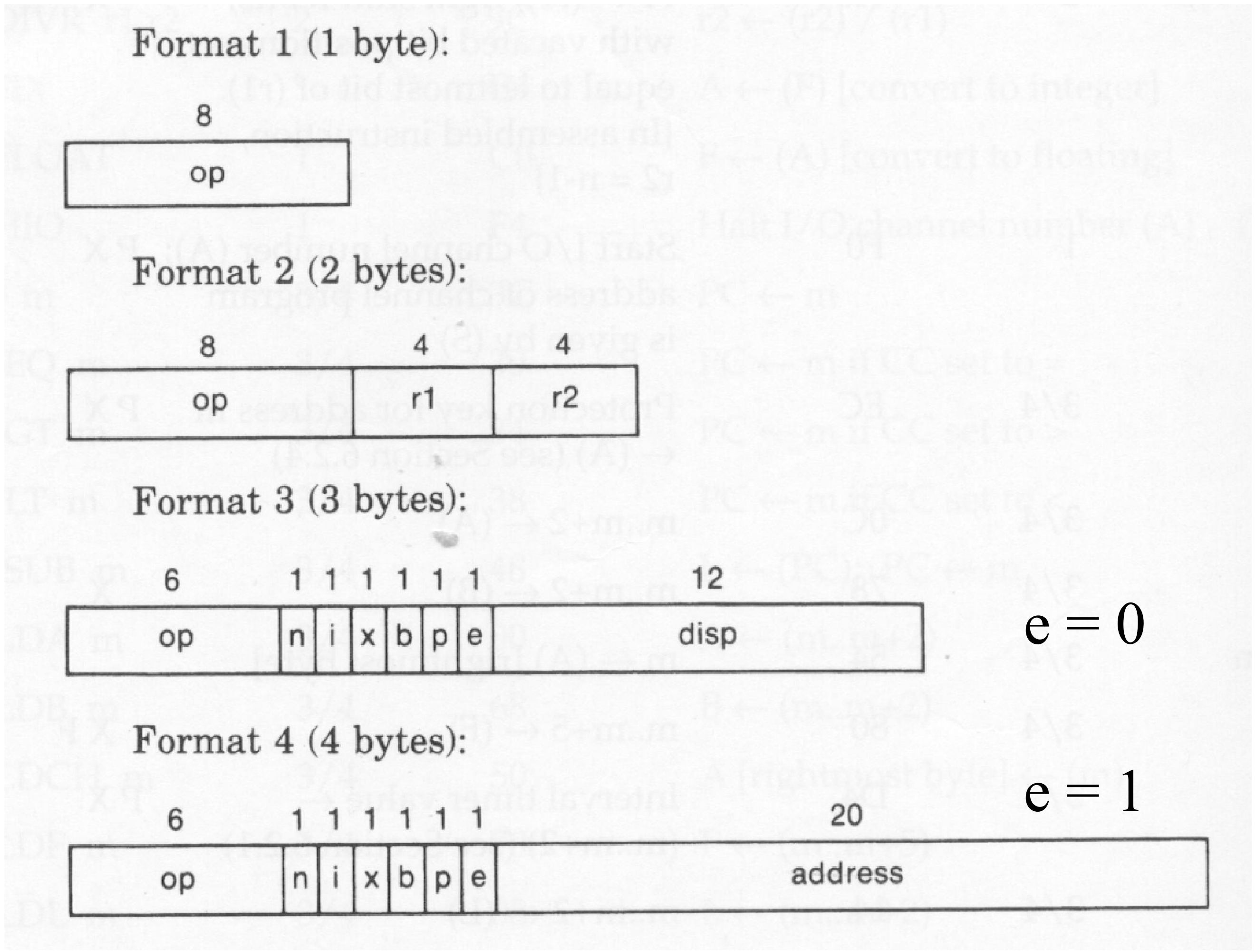
Data Format

- The same as that of SIC.
- There is a floating-point data type with the following format:



- The value represented by the above format is (-1)^s * f * 2 ^(e – 1024)

Instruction Formats

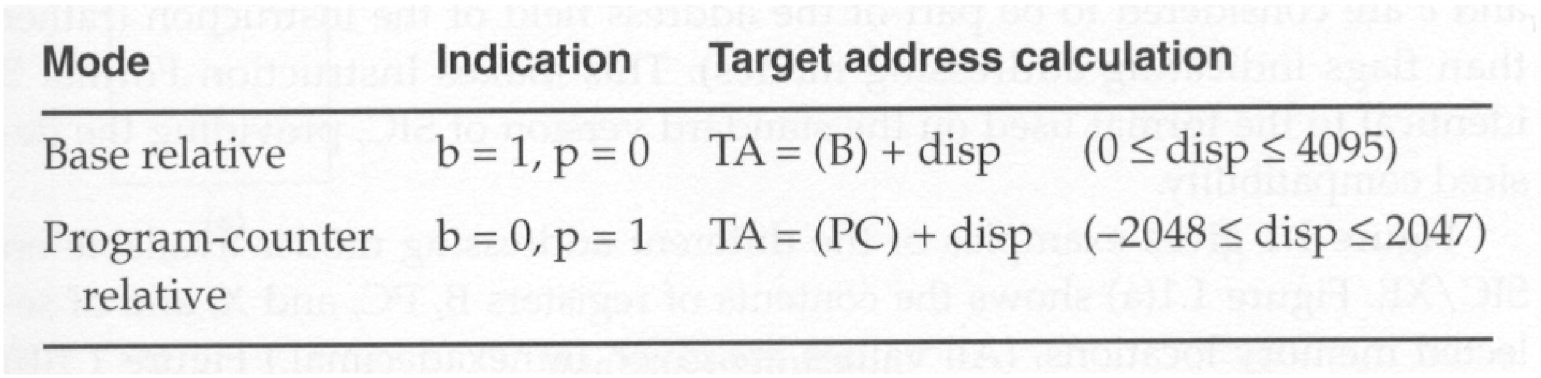


e = 0

e = 1

Addressing Mode

- Two additional modes are introduced for format 3:



Mode	Indication	Target address calculation	
Base relative	b = 1, p = 0	TA = (B) + disp	$(0 \le \text{disp} \le 4095)$
Program-counter relative	b = 0, p = 1	TA = (PC) + disp	$(-2048 \le \text{disp} \le 2047)$

In format 3, if both bits b and p are set to 0, the disp (or address) is taken as the target address. This is called direct addressing mode.

Both modes can be combined with indexed addressing – if bit x is set to 1, the value of register X is added in the target Address calculation.

Addressing Modes

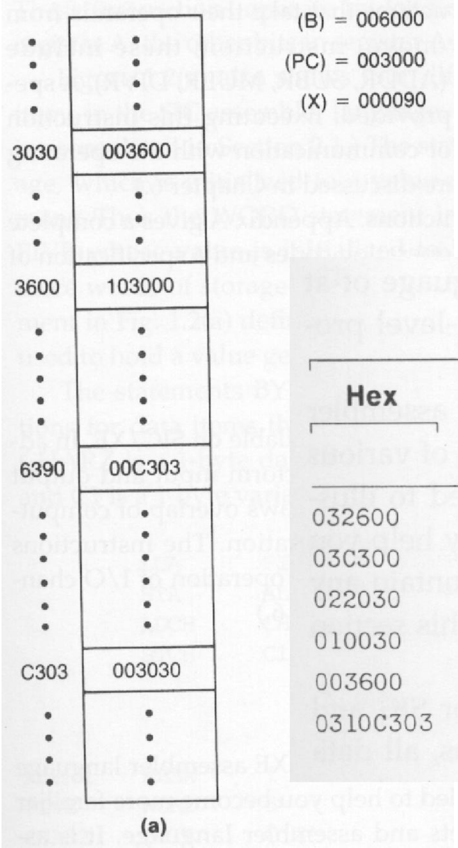
For format 3 and 4:

- (i =1, n = 0): immediate addressing mode. The target address is used as the operand.
- (i = 0, n = 1): indirect addressing mode. The word at the location given by the target address is fetched; the value contained in this word is then used as the address of the operand value.
- (i = 0, n = 0) used by SIC, (i=1, n=1) used by SIC/XE: simple addressing mode. The target address is taken as the location of the operand.

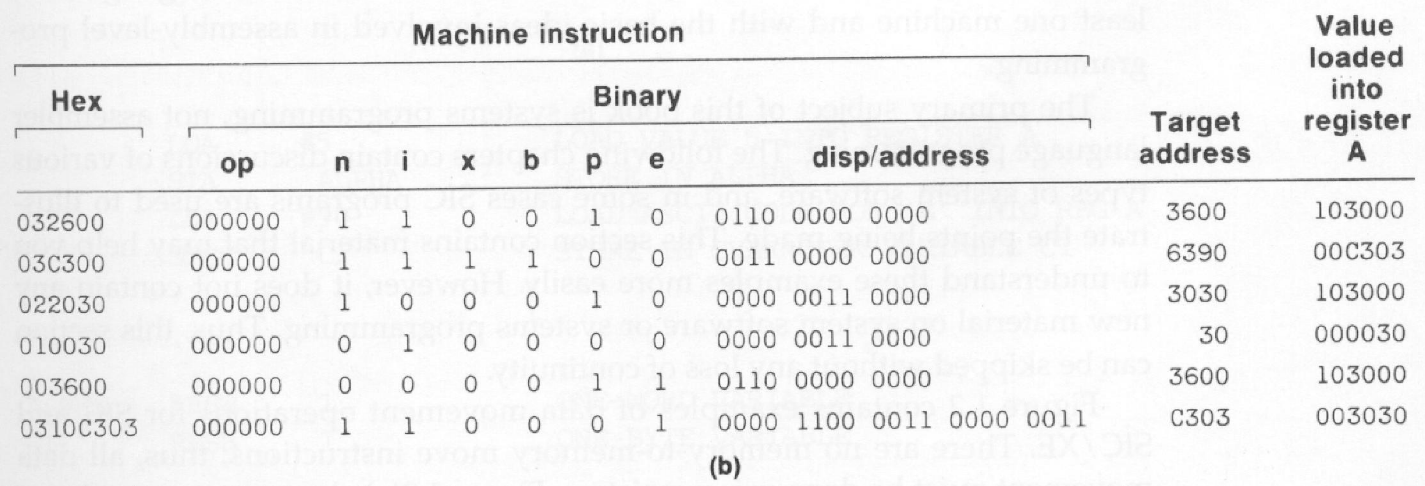
Indexing mode cannot be used with immediate or indirect modes.

Addressing type	Flag bits n i x b p e	Assembler language notation	Calculation of target address TA	Operand	Notes
Simple	1 1 0 0 0 0	op c	disp	(TA)	D
	1 1 0 0 0 1	+op m	addr	(TA)	4 D
	1 1 0 0 1 0	op m	(PC) + disp	(TA)	A
	1 1 0 1 0 0	op m	(B) + disp	(TA)	A
	1 1 1 0 0 0	op c,X	disp + (X)	(TA)	D
	1 1 1 0 0 1	+op m,X	addr + (X)	(TA)	4 D
	1 1 1 0 1 0	op m,X	(PC) + disp + (X)	(TA)	A
	1 1 1 1 0 0	op m,X	(B) + disp + (X)	(TA)	A
	0 0 0 - - -	op m	b/p/e/disp	(TA)	D S
	0 0 1 - - -	op m,X	b/p/e/disp + (X)	(TA)	D S
Indirect	1 0 0 0 0 0	op @c	disp	((TA))	D
	1 0 0 0 0 1	+op @m	addr	((TA))	4 D
	1 0 0 0 1 0	op @m	(PC) + disp	((TA))	A
	1 0 0 1 0 0	op @m	(B) + disp	((TA))	A
Immediate	0 1 0 0 0 0	op #c	disp	TA	D
	0 1 0 0 0 1	+op #m	addr	TA	4 D
	0 1 0 0 1 0	op #m	(PC) + disp	TA	A
	0 1 0 1 0 0	op #m	(B) + disp	TA	A

Example



All of these instructions are LDA.



Machine instruction										
Hex	Binary								Target address	Value loaded into register A
	op	n	i	x	b	p	e	disp/address		
032600	000000	1	1	0	0	1	0	0110 0000 0000	3600	103000
03C300	000000	1	1	1	1	0	0	0011 0000 0000	6390	00C303
022030	000000	1	0	0	0	1	0	0000 0011 0000	3030	103000
010030	000000	0	1	0	0	0	0	0000 0011 0000	30	000030
003600	000000	0	0	0	0	1	1	0110 0000 0000	3600	103000
0310C303	000000	1	1	0	0	0	1	0000 1100 0011 0000 0011	C303	003030

(b)

Instruction Set

- LDB and STB
- Floating-point operations (ADDF, SUBF, MULF, DIVF)
- Register move (RMO)
- Register-to-register operations (ADDR, SUBR, MULR, DIVR)
- Supervisor call (SVC) for generating system calls into the operating system.
- I/O channel operation (SIO: start, TIO: test, HIO: halt), similar to DMA.