

Mid Semester Examination

①

V. Divya Srinivasan
20CSE1042

- 1) Let Alpha = multiplicand
 Beta = Multiplier
 Gamma = Product.

$$\text{ALPHA} * \text{BETA} = \text{GAMMA}$$

PRODUCT	STAT	8000
	LDA	ZERO.
	STA	GAMMA
	LDX	ZERO
SUM	LDA	ALPHA
	ADD	GAMMA
	STA	GAMMA
	TIX	BETA
	JLT	SUM
OUTLP	TD	OUTDEV
	JEQ	OUTLP
	LDA	GAMMA
	KID	OUTDEV
ALPHA	RESW	1
BETA	RESW	1
GAMMA	RESW	1
ZERO	WORD	0
OUTDEV	BYTE	X'05'
	END	FIRST

<u>Address</u>	<u>Generated in pass 1</u>			<u>Object code</u>	<u>in pass 2</u>
2) 3000	PRODUT	START	3000		00302D
3000		LDA	ZERO		0C302A
3003		STA	GAMMA		04302D
3006		LDX	ZERO		003024
3009	SUM	LDA	ALPHA		18302A
300C		ADD	GAMMA		0C302A
300F		STA	GAMMA		2C3027
3012		TRX	BETA		
3015		JLT	SUM	383009	
3018	OUTLP	TD	OUTDEY	E03030	
301B		JEQ	OUTLP	303018	
301E		LDA	GAMMA	00302A	
3021		WD	OUTDEV	DC3030	.
3024	ALPHA	RESW	1		
3027	BETA	RESW	1		
302A	GAMMA	RESW	1		
302D	ZERO	WORD	0	000000	
3030	OUTDEY	BYTE	X'05'	05	
3031		END	FIRST		

Length = 3031
-3000
0031

Object program :- → Generated in pass 2

M PRODUCT A 003000 A 000031

T A 003000 A 1 E A 00302D A 0C302A A 04302D A 003024 A
 18302A A 0C302A A 2C3027 A 383009 A E03030 A 303018
 T A 00301E A 06 A 00302A A 0DC3030
 T A 00302D A 04 A 000000 A 05
 E A 003000

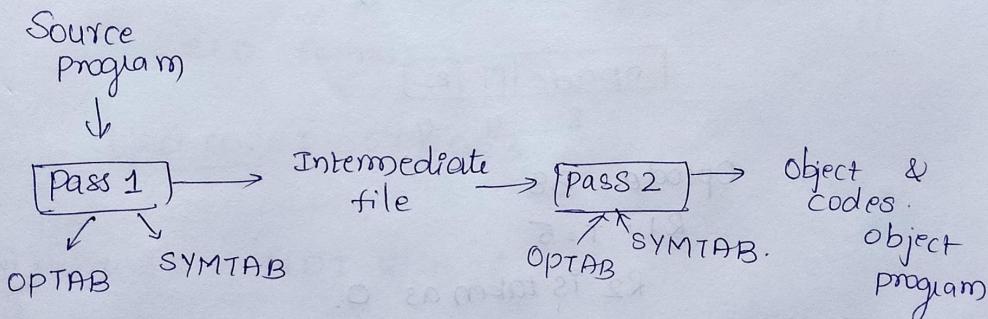
(3)

Pass 1 :

- Addresses are assigned to instructions and symbols.
- Machine codes of corresponding Mnemonic instructions are stored in OPTAB and addresses assigned to all labels are saved in SYMTAB.
- Assembler directives (START, END) are processed.

Pass 2 :

- Object code for the instructions are generated with the help of OPTAB and SYMTAB.
- Object code for constants defined by BYTE,WORD are generated.
- Remaining Assembler directives are processed.
- Object program is generated.



(4)

3. a) 1059 RSUB.

opcode	n	i	b	p	e	Displacement
0100 1111 0000 0000 0000 0000 0000	6	1	1	1	1	12

Disp Address = 0000

Opco = 4C = 0100 11

 $n, i = 1, 1 \rightarrow$ simple addressing $a = 0 \rightarrow$ (Not indexed)

Direct addressing.

 $b = 0, p = 0 \rightarrow$ No relative

Absolute addressing

 $e = 0 \rightarrow$ Format 3.

Object code :- 4 F0000.

b) 1051 TIXR T.

Format 2 instruction

opcode	R1	R2
8	4	4

Opco = B8

R1 = T = 5

R2 is taken as 0.

Object code: B850.

c) 104E STCH BUFFER, X.

opcode	n	i	b	p	e	Displacement
0101 0111 1111 1100 0000 0000 0011	6	1	1	1	1	12

Opco of STCH = 54

Disp Address: let's take PC relative,
 $(TA) - (PC)$
 $= 0036 - 1051$

$$\begin{array}{r}
 \text{1031} \quad 16+1 \\
 -0036 \\
 \hline
 \text{101B}
 \end{array}$$

(5)

$$\begin{array}{r}
 \text{FFFF} \\
 -\text{101B} \\
 \hline
 \end{array}$$

$$\begin{array}{r}
 \text{FFE4} \\
 +1 \\
 \hline
 \text{EFF5}
 \end{array}
 \rightarrow \text{16's complement}$$

length > 12 80 pc relative not possible

Base relative:

$$\begin{aligned}
 \text{Disp} &= [\text{TA}] - [\text{B}] \\
 &= 0036 - 0033 \\
 &= 0003.
 \end{aligned}$$

$m, i = 1, 1$ simple addressing

$\alpha = 1$ indexed addressing

$b, p = b=1, p=0 \Rightarrow$ Base relative addressing

$e=0$ Format 3.

object code: 57 C 0003

d) 103C + LDT #4096

Format 4:

opcode	m	i	α	b	p	e	Displacement
0111 01	0	1	0	0	0	1	0000 0001 0000 0000 0000 0000

6 1 1 1 1 1 20

LDT \Rightarrow 74 opcode (0111 01)

4096 \Rightarrow 1000 in hexadecimal representation

$m=0, i=1 \Rightarrow$ immediate addressing

$\alpha=0, \rightarrow$ Direct

$b=0, p=0 \rightarrow$ No relative

$e=1 \rightarrow$ Format 4.

Object code: 475101000.

e) 002D EOF BYTE C'EOF'.

Value of EOF = 454FU6
 ↓
 constant

Object code: 454FU6.

4.

a)

SIC	SIC/XE						
<u>Memory :-</u> i) Word length = 24 bits ii) Memory size = 2^{15} bytes Any byte can be addressable by 15-bit address	i) word length = 24 bits ii) Memory size = 2^{20} bytes Any byte can be addressable by 20-bit address.						
<u>Registers:-</u> i) It supports 5 registers A, X, L, PC, SW	Registers i) It supports 9 registers. extra registers used in SIC/XE:- B, S, T, F.						
<u>Data formats:</u> It supports - Integers - characters	It supports - integers - characters - Floating point.						
<u>Instruction format:</u> 24 bit length	- 4 types of instruction formats are available in SIC/XE. <u>Format 1 :- 1 Byte</u> <table border="1"> <tr> <td>Opcode</td> <td>x</td> <td>Address</td> </tr> <tr> <td>8.</td> <td>1</td> <td>15</td> </tr> </table>	Opcode	x	Address	8.	1	15
Opcode	x	Address					
8.	1	15					

(7)

Format 2: 2 Byte

Opcode	R1	R2
8	4	4

Format 3:- 3 byte

Opcode	m	i	a	b	P	e	displacement
6	1	1	1	1	1	1	12

Format 4:- 4 bytes

opcode	m	i	a	b	P	e	Address
6	1	1	1	1	1	1	20

Addressing modes:

- It has indexed and direct addressing modes

SIC is the base version

- It has additionally.
- relative addressing
- Base relative
- PC relative

SIC/XE is the advanced version of SIC

b) Relative Addressing is better than absolute addressing.

- 1) In relative addressing, effective address is calculated by adding contents of CPU register with the address part of the instruction. So it is used for writing relocatable code. (i.e supports program relocation) but in absolute addressing it is not possible.
- 2) Relative addressing results in shorter access address field in instruction format as it can be specified in small no. of bits. whereas absolute addressing results in longer access address field.
- 3) Relative addressing can also be used in branch type instructions, as it can directly update the program counter but absolute addressing can not be used.

4*) Relative addressing can also handle recursive functions which cannot be handled by using absolute addressing.

c) In SIC, for program relocation all instructions except RSUB and I/O instructions cause a modifier record to be written because length all instructions use (15 bit) memory address.

- Assembler assumes the starting address as zero and all address specified should be relative to starting address of the program.
- And we have each time the assembler produces an instruction with an address, a modification record is produced.

Text records