SIC/XE Practice Problems

Sample Addressing Modes :-

**Note :-

According by set flags for object code Generation

① LDA @ ALPHA → indirect-addressing mode
n=1, i=0, x=0, ...
(@ → indirect)

② LDA # 123 → immediate addressing "
n=0, i=1, x=0, b=0, p=0
(# → immediate)

③ LDA ALPHA, X → indexed addressing "
n=1, i=1, x=1, b/p=1
(X → indexed)

④ +LDA ALPHA → format-4 addressing "
(e=1)
(+ → format-4)

⑤ LDA ALPHA → PC/Base relative addressing
n=i=1, p=1 (or) b=1, x=0

Problem : (1) Generate object-Program

CLEAR = B4 , LDT = 74 , TD = E0 , JEQ = 30 ,
LDCH = 50 , WD = DC , TIXR = B8 , JLT = 38 ,
RSUB = 4C
Address of BUFFER is 4033 and Address of LENGTH is 4036

	LOCCTR	Label	opcode	operand	object-code
1)		WRREC	START	405D	
2)	405D		CLEAR	X	B410
3)	405F		LDT	LENGTH	772FD4
4)	4062	WLOOP	TD	OUTPUT	E32011
5)	4065		JEQ	WLOOP	3321FA
6)	4068		LDCH	BUFFER, X	53AFC8
7)	406B		WD	OUTPUT	DF2008
8)	406E		TIXR	T	B850
9)	4070		JLT	WLOOP	3B2FEF
10)	4073		RSUB		4F0000
11)	4076	OUTPUT	BYTE	X'05'	05
12)	4077		END		405D // this is not obj. code

Line - (2) CLEAR, X

opcode	R1	R2
1011 0100	0001	0000
B 4	1	0

= object-code B410

Reg	NO.
A	0
X	1
L	2
B	3
S	4
T	5
F	6
PC	8
SW	9

Line 3 — LDT LENGTH

opcode	n i x b p e	displacement
011101ϕϕ	1 1 0 0 1 0	FD4

Disp = 4036 − 4062 = −2C $\xrightarrow{2\text{'s complement}}$ FD4

PC relative ! −2048 to 2047.

⇒ objcl-code = 772FD4

Line 4 — TD OUTPUT

opcode	n i x b p e	displacement
111000ϕϕ	1 1 0 0 1 0	011

Disp = 4076 − 4065 = 11

∴ PC relative : −2048 to 2047

⇒ objcl-code = E32011

Line 5 — JEQ WLOOP

opcode	n i x b p e	Displacement
0011 00ϕϕ	1 1 0 0 1 0	FFA

Disp = 4062 − 4068 = −6 $\xrightarrow{2\text{'s complement}}$ FFA

∴ PC relative = −2048 to 2047

⇒ objcl-code = 332FFA

Line 6 LDCH BUFFER, X

opcode	n i x b p e	displacement
01010000	1 1 1 0 1 0	FC8

Disp = 4033 − 406B = −38 $\xrightarrow[\text{complement}]{2's}$ FC8

∴ PC relative = −2048 to 2047

⇒ Object-code = 53AFC8

Line 7 WD OUTPUT

opcode	n i x b p e	displacement
11011100	1 1 0 0 1 0	008

Disp = 4076 − 406E = 8

⇒ Object-code = DF2008

Line 8 TIXR T

opcode	R1	R2
10111000	0101	0000

⇒ Object-code = B850

Line 9 JLT WLOOP

opcode	n i x b p e	displacement
00111000	1 1 0 0 1 0	FEF

Disp = 4062 − 4073 = −11 $\xrightarrow[\text{complement}]{2's}$ FEF

⇒ Object-code = 3B2FEF

Line 10 RSUB

opcode	n	i	x	b	pe	displacement
01001100	1	1	0	0	0	000

=> object-code = 4F0000

Line 11 BYTE X'05' // Meaning is: Store 05 in register 'X'.

=> object-code = 05

Line 12 END

For END statement - we dont calculate object-code. In place of object-code we write the start address of source program.

Here it is 405D.

Object-Program

H^WRREC_^00450D^001A
T^00405D^1A^B410^772FD4^E32011^332FFA
^53AFC8^DF2008^B850^3B2FEF
^4F0000^05
E^00450D

length of the Source Program = 4077 - 405D = 001A

Problem ② Generate object-Program

H.W

LDA = 00, STA = 0C, ADD = 18, SUB = 1C

	ADDITION	START	1000
		LDA	ALPHA
		ADD	INCR
Step1		SUB	ONE
Calculate		STA	BETA
object-		LDA	GAMMA
code		ADD	INCR
for		SUB	ONE
each		STA	DELTA
statement			
Step2	ONE	WORD	1
write	ALPHA	RESW	1
object-	BETA	RESW	1
program	GAMMA	RESW	1
H^...	DELTA	RESW	1
T^...			
E^...	INCR	RESW	1
		END	

---

* Do class Problems also

* Before Symbolic operand if '@' is there it is indirect-add.
* " " '#' " immediate add
* with " " 'X' " indexed add
* If before opcode/instruction '+' is there it is format-4. (e=1)
* Remaining all are direct-address. => n=i=1, b or p=1, (flags)

Problem ③ / Convert to object Program

RSUB = 4C

LDX = 04, LDA = 00, LDB = 68, ADD = 18, JLT = 38, TIX = 2C

Source Prog

				Object-code
SUM	START	0		
FIRST	LDX	#0	→	050000
	LDA	#0	→	010000
	+LDB	#TABLE2	→	69101790
	BASE	TABLE2	→	NO object-code as it is BASE, no address also.
LOOP	ADD	TABLE, X	→	1BA013
	ADD	TABLE2, X	→	1BC000
	TIX	COUNT	→	2F200A
	JLT	LOOP	→	3B2FF4
	+STA	TOTAL	→	0F102F00
	RSUB		→	4F0000
COUNT	RESW	1		
TABLE	RESW	2000		
TABLE2	RESW	2000		
TOTAL	RESW	1		
	END	FIRST		

* Write the object-Program seeing the written Object-Codes

H^ ---

T^ ---

E^ ---

SIC/XE Practice Problems

Sample Addressing Modes :-

**Note :-

According to set flags for object code Generation

① LDA @ ALPHA → indirect-addressing mode
indirect- n=1, i=0, x=0, b=0, p=0

② LDA # 123 → immediate addressing "
immediate n=0, i=1, x=0, b=0, p=0

③ LDA ALPHA, X → indexed addressing "
indexed n=1, i=1, x=1, b/p = 1 (b or p)

④ +LDA ALPHA → format-4 addressing "
format-4 (e=1)

⑤ LDA ALPHA → PC/Base relative addressing
n=i=1, p=1 (or) b=1, x=0