B. E. 5th SEMESTER (IT) FINAL EXAMINATIONS 2011
System programming (IT-504)

Full Marks: 70 **Duration – 3 Hours**

Answer question number one and any five from the rest

1. ***Answer any five*** **2x5**

a) Why is it necessary sometimes to use the extended format instructions for SIC/XE machine assembly level program?

b) Consider the following possibility for the storage linking and execution of user's program. *"Store the source program and a linked version that has all external references resolved. Use a relocating loader each time the program is to be executed"*. Under what condition(s) might each of these approaches be appropriate?

c) What is the use of BASE and NOBASE assembler directives?

d) What is load-and-go assembler?

e) What is the advantage of writing the statement

```
MAXLEN    EQU   BUFFEND – BUFFER
```

instead of

```
MAXLEN    EQU   4096          ?
```

f) The process of fixing up a few forward references should involve less overhead than making a complete second pass of the source program. Why don't all assemblers use the one-pass technique for efficiency?

g) What is macro-time looping statement? Give one example.

2. **(2+4+6)**

a) Which register(s) you will use to do the floating point operation in SIC/XE machine? Write the floating point data format supported by the SIC/XE machine.

b) Write the different instruction formats supported by both the SIC and SIC/XE machines.

c) Suppose ALPHA is an array of 100 bytes record that contains the ASCII codes of the characters from '0' – '9'. Write a subroutine for SIC/XE machine that counts the number of occurrences of each character in ALPHA. (Hint: ASCII code of '0' is 30_H)

3. **(4+6+2)**

a) The assembler could automatically use extended format instructions whose operands involve external references. This would eliminate the need for the programmer to code + in such statements. What would be the advantages and disadvantages of doing this?

b) Outline the flow chart for first pass of the two pass assembler.

c) The assembler could simply assume that any reference to a symbol not defined within a control section is an external reference. This change would eliminate the need for the EXTREF statement. Would this be a good idea?

4. **(6+6)**

a) Write the relocatable loader algorithm for SIC machine based assembled program.

b) Determine the missing object codes for the following program segment WRREC and write the object codes in a file using proper format.

Loc	Source statements			Object code
0000	WRREC	START	0	
0000		CLEAR	X	B410
0002		LDT	LENGTH	
0005	WLLOP	TD	=X'25'	
0008		JEQ	WLOOP	332FFA
000B		LDCH	BUFFER, X	
000E		WD	=X'25'	
0012		TIXR	T	B850
0014		JLT	WLOOP	3B2FEE
0017		RSUB		4F0000
001A	LENGTH	RESW	1	
001D	BUFFER	RESB	200	

Opcodes of the required instructions are as follows: LDT → $(74)_H$, TD → $(E0)_H$, LDCH → $(50)_H$, WD → $(DC)_H$

5. (4+8)

a) What is load-on-call? Explain how this load-and-call mechanism happens.

b) Consider the three (separate assembled program) programs each of which consists of a single control section. Each program contains a list of items (LISTA, LISTB, LISTC); the ends of these lists are marked by the labels ENDA, ENDB and ENDC.

Loc	Source statements			Object code
0000	PROGA START 0			
		EXTDEF	LSITA, ENDA	
		EXTREF	LISTB, ENDB, LISTC, ENDC	
		:		
		:		
0020	REF1	LDA	LISTA	????
0023	REF2	+LDT	LISTB-4	????
0027	REF3	LDX	#ENDA-LISTA	????
		:		
		:		
0040	LISTA	EQU	*	
		:		
0054	ENDA	EQU	*	
0054	REF4	WORD	ENDA-LISTA+LISTC	????
		END		

Symbols LISTB, ENDB, LISTC and ENDC are defined in the control sections PROGB and PROGC at locations $(0060)_H$, $(0070)_H$, $(0030)_H$ and $(0042)_H$ respectively similar to PROGA. PROGA has been loaded at starting address 4000 with PROGB and PROGC immediately following. PROGB has length $(7F)_H$ and PROGC has the length $(51)_H$ bytes.

Explain how does the linking loader solve the relocation and linking operation for instruction operands of REF1, REF2, REF3 and REF4?

LDA, LDT and LDX has the machine code $(00)_H$, $(74)_H$ and $(04)_H$ respectively. Determine the object codes after linking operation is performed for the instructions marked as ???? in the object code column.

6. 2+6+4

a) What is a bootstrap loader?

b) Write a bootstrap loader in SIC/XE machine instructions.

c) What are the different data structures used for designing a linking loader? Discuss their usage.

7. 6+4+2

a) Write an algorithm for a *two-pass* macro processor in which all macro definitions are processed in the first pass, and all macro invocations are expanded in the second pass. You do not need to allow for macro definitions or invocations within macro.

b) What is macro time variable? How can it be used in a macro definition? Show with example.

c) Show with example how the concatenation of macro parameters is done?

8. 5+4+3

a) What are the different data structures used to construct a one-pass macro processor? Explain with examples the use of each data structure.

b) What are keyword macro parameters? Explain with example how the keyword macro parameter is used and it is different form positional macro parameters.

c) What is conditional macro expansion? How can it be used in a macro definition? Show with example.

9. **Write short notes on any two of the followings**: 2x6

a) Linking loader
b) Control sections
c) Program block
d) Multi-pass assembler