

BENGAL ENGINEERING AND SCIENCE UNIVERSITY, SHIBPUR
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.** **5x2**
- a) Identify the format types and determine the object codes for the instructions COMPR A, S and LDA #3. Opcodes of the instructions are (A0)_H and (00)_H respectively.
- b) Write the use of ORG assembler directive with example.
- c) What is macro-time variable? Give example.
- d) What will happen if a programmer writing all instructions of an SIC/XE machine based assembly program using extended format?
- e) Consider the following possibility for the storage, linking and execution of a user's program:
"Store the source and object versions of the program; use a linking loader each time the program is to be executed."
- f) What is bootstrap loader and write its functions.
- g) What is the function of SW register of SIC/XE machine. Show with example how this register can be used?
2. **6+6**
- a) Write an SIC/XE or SIC program to find the GCD of two integer numbers.
- b) Write the different addressing modes and instruction formats supported by the SIC and SIC/XE machine.
3. **7+5**
- a) Outline the flowchart for the pass-1 of two pass assembler.
- b) Immediate operands and literals are both ways of specifying an operand value in a source statement. What are the advantages and disadvantages of each? When might each be preferable to the other?
4. **6+6**
- a) Outline the algorithm for the one-pass load-and-go assembler.
- b) Explain with example how the forward reference problem is resolved by an one pass assembler.
5. **5+3+4**
- a) How could an assembler that allows external references avoid the need for an EXTDEF statement? What would be the advantages and disadvantages of doing this?
- b) What do you mean by the relative and absolute expression? Explain with example for each case.
- c) Write an algorithm for SIC re-locatable loader.
6. **2+5+5**
- a) What is the function of a linkage editor?
- b) Suggest a module format suitable for representing a linked programs produced by a linkage editor. Assume that the linked program is not to be reproduced by the linkage editor.
- c) Describe the working principle of dynamic linking.

7.

7+5

a) Consider the following control sections.

Loc		Source statements	
0000	PROGB	START	0
		EXTDEF	LSITB, ENDB
		EXTREF	LISTA, ENDA, LISTC, ENDC
		:	
		:	
0036	REF1	+LDA	LISTA
003A	REF2	LDT	LISTB+4
003D	REF3	+LDX	#ENDA-LISTA
		:	
		:	
0060	LISTB	EQU	*
		:	
0070	ENDB	EQU	*
0070	REF4	WORD	ENDA-LISTA+LISTC
		END	

Similarly, other control sections are PROGA and PROGC. Symbols LISTA, ENDA are defined at the locations $(0040)_H$, $(0054)_H$ in PROGA and LISTC, ENDC are at $(0030)_H$ and $(0041)_H$ in PROGC. The same type of expressions is also defined in PROGA and PROGC like PROGB. For execution of the program all the three control sections are to be loaded consecutive locations starting from PROGB, then PROGA and then PROGC. Object code of PROGA is of length $(0063)_H$ and of PROGC is $(0051)_H$. LDA, LDT and LDX has the machine code $(00)_H$, $(74)_H$ and $(04)_H$ respectively.

Explain how the linking loader solves the relocation and linking operation for the instruction operands of REF1, REF2, REF3 and REF4. Determine the object code before linking operation is performed and after linking operation is performed. Consider the relocation will start from the location $(4000)_H$.

b) What are the different data structures used for designing a linking loader? Discuss with example.

8.

4+4+4

a) It is not possible to define for the body of a macro instruction to contain labels. How can the constraint be removed? Show with example.

b) Explain with example about how the different data structures are used to design a two pass macro processor?

c) Explain with example how the macro time looping statement can be used in a macro body.

9.

3+4+5

a) Write the different stages of a compiler and their basic functions in brief.

b) Write the algorithm for converting a regular expression directly to DFA.

c) Convert the regular expression $(a|b)^*a$ to NFA using Thompson's construction and then convert it into DFA.