

BALLARI INSTITUTE OF TECHNOLOGY & MANAGEMENT

(Autonomous Institute under Visvesvaraya Technological University, Belagavi)

USN

--	--	--	--	--	--	--	--	--	--

Course Code

2	2	C	S	4	6	1
---	---	---	---	---	---	---

Fourth Semester B.E. Degree Examinations, July 2026

SYSTEM SOFTWARE

Duration: 3 hrs

(Computer Science & Engineering)

Max. Marks: 100

Note: 1. Answer any FIVE full questions choosing ONE full Question from each Module.

2. Missing data, if any, may be suitably assumed

<u>Q. No</u>	<u>Question</u>	<u>Marks</u>	<u>(RBTL:CO:PI)</u>
Module-1			
1.	a. Define system software. Distinguish the system software with application software and give examples.	05	(1 : 1 : 1.6.1)
	b. Write a sequence of instruction for SIC to set all 100 elements to zero if ALPHA is an array of 100 words.	05	(2 : 1 : 1.6.2)
	c. The contents of registers B, PC, and X are 006000, 003000, and 000090. What will happen if the following instructions are executed? Show the target address calculation. (i) 032600 (ii) 03C300 (iii) 022030 (iv) 010030 (v) 003600	10	(2 : 1 : 1.6.1)
(OR)			
2.	a. List the steps to translate SIC source program to object program.	05	(1 : 1 : 1.6.2)
	b. Write a sequence of instructions for SIC to clear a 20-byte string to all blanks.	05	(2 : 1 : 1.6.3)
	c. Explain different instruction formats and addressing modes used in SIC/XE machine.	10	(2 : 1 : 1.6.1)
Module-2			
3.	a. Explain the algorithm of PASS-1 of two-pass assembler.	05	(1 : 2 : 1.6.1)
	b. Explain absolute and relative expressions. How these are processed by an assembler?	05	(2 : 2 : 1.6.1)
	c. When is multi-pass assembler required? Show step by step procedure to evaluate the following statements. Show the symbol table after each scan. (i) HALFSZ EQU MAXLEN/2 (ii) MAXLEN EQU BUFEND-BUFFER (iii) PREVBT EQU BUFFER-1 (iv) BUFFER RESB 4096 (v) BUFEND EQU *	10	(2 : 2 : 1.6.1)
(OR)			
4.	a. Explain the algorithm of PASS-2 of two-pass assembler.	05	(1 : 2 : 1.6.1)
	b. Define program block. How multiple program blocks are handled by assemblers?	05	(2 : 2 : 1.6.1)

Note: (RBTL - Revised Bloom's Taxonomy Level: CO - Course Outcome: PI- Performance Indicator)

- c. Generate the complete object program for the following assembly level SIC/XE program. **10 (2 :2 : 1.7.1)**

```

SUM      START    4000
FIRST    CLEAR    X
          LDA      #0
          + LDB    #TOTAL
          BASE     TOTAL
LOOP     ADD      TABLE, X
          TIX      COUNT
          JLT      LOOP
          STA      TOTAL
COUNT   RESW     1
TABLE    RESW     2000
TOTAL    RESW     1
          END      FIRST

```

Assume below Opcode (all in hexadecimal)

CLEAR – B4 LDA – 00 LDB – 68 ADD – 18 TIX – 2C
JLT – 38 STA – 0C

Module-3

5. a. List the basic functions of a loader. **05 (1 :3 : 1.6.1)**
b. Explain the algorithm of PASS 1 linking loader **05 (2 :3 : 1.6.1)**
c. Write a SIC/XE program for a bootstrap loader. **10 (2 :3 : 1.6.1)**

(OR)

6. a. Differentiate between linking loader and linkage editor. **05 (1 :3 : 1.6.1)**
b. Write and describe the algorithm of PASS 2 linking loader. **05 (2 :3 : 1.6.1)**
c. Explain dynamic linking with a neat figure. Discuss its applications. **10 (2 :3 : 1.6.1)**

Module-4

7. a. What is Macro? With an example explain the MACRO and MEND assembler directives. **05 (1 :4 : 1.7.1)**
b. What are the data structures used in the macro processor design? **05 (2 :4 : 1.7.1)**
c. Explain the following machine independent macro processor features **10 (2 :4 : 1.6.1)**
(i) The generation of unique labels.
(ii) The concatenation of Macros- Parameters.

(OR)

8. a. Explain the macro processor algorithm. **05 (1 :4 : 1.6.1)**
b. List the advantages and disadvantages of general purpose macro processors. **05 (2 :4 : 1.6.1)**
c. Explain the following Machine Independent macro processor features **10 (2 :4 : 1.6.1)**
(i) Conditional Macro expansion.
(ii) Keyword Macro Parameter

Module-5

9. a. Write a Lex program to count the number of words, characters, blanks and lines in a given file. 05 (2 :5 : 1.7.1)
- b. Write a YACC program to recognize a valid arithmetic expression that uses +, -, *, /. The tokens are passed from a Lex program. 05 (2 :5 : 1.7.1)
- c. What is regular expression? Briefly explain the meta characters used in regular expression with an example. 10 (2 :5 : 1.7.1)

(OR)

10. a. Write a YACC program to check whether the given string $a^n b^n$ ($n > 0$) is accepted by the grammar or not. 05 (2 :5 : 1.7.1)
- b. Explain the shift-reduce and reduce-reduce conflicts in YACC with example. 05 (2 :5 : 1.6.1)
- c. Explain the structure of LEX and YACC programs. Show the procedure to get a parser using LEX and YACC tools. 10 (2 :5 : 1.6.1)

** ** *