BIRLA INSTITUTE OF TECHNOLOGY, MESRA, RANCHI
(END SEMESTER EXAMINATION)

CLASS: BE **SEMESTER : VI**
BRANCH: CSE **SESSION : SP/19**

SUBJECT: CS6103-SYSTEM PROGRAMMING

TIME: 3.00 Hrs. **FULL MARKS: 60**

INSTRUCTIONS:
1. The question paper contains 7 questions each of 12 marks and total 84 marks.
2. Candidates may attempt any 5 questions maximum of 60 marks.
3. The missing data, if any, may be assumed suitably.
***4*. Before attempting the question paper, be sure that you have got the correct question paper.**
5. Tables/Data hand book/Graph paper etc. to be supplied to the candidates in the examination hall.

Various formats for all SIC and SIC/XE related questions:
- 1 word = 3 bytes. 1 hexadecimal digit = half byte.
- SIC Instruction format: 24 bits = 8-bits Opcode + x + 15-bits address.
- SIC/XE Instruction format: 24 bits = 6-bits Opcode + n + i + x + b + p + e + 12-bits displacement.
- SIC/XE Instruction format: 32 bits = 6-bits Opcode + n + i + x + b + p + e + 20-bits address.
- 1-bit flags: x=index, n=indirect, i = immediate, b=base-relative, p=pc-relative, e=extended.
- The displacement in an instruction can be negative (2's complement representation).
- SIC/XE 6-bits Opcode: 00 implicitly added as last 2 bits to make it 8-bits.
- In Assembly file, all numerical operands are in decimal (other than the address in 1st line and the addresses in the Listing, which are in hexadecimal).
- In Object file, everything is in hexadecimal.
- Header Record: H.<prog-name>.<3-byte-starting-address>.<3-byte-prog-length>
- Text Record: T.<3-byte-starting-address>.<1-byte-length>.<sequence of instruction/data>
- End Record: E or E.<3-byte-starting-address>
- Modification Record: M.<3-byte-address-to-be-modified>.<1-byte-length-to-be-modified (in # of half-bytes)>.< + or - ><external-label or prog-name>
  * used for relocation and linking purposes.
- Define Record: D.<sequence of label-name.3-byte-label-address>
  * for labels defined in this control section and used outside; used for linking purposes.
- Refer Record: R.<sequence of label-names>
  * for external references appearing in this control section; used for linking purposes.
- Note that DOT (.) is used in the records for readability. It may be omitted also.

Q.1(a) Differentiate between RISC and CISC systems. [2]

Q.1(b) State the target address and value loaded in register A for the Hex code 022030 and 032600 for SIC(XE) architecture. Let assume (B)=006000, (PC)=003000 and (X)=000090. [4]

Memory Status:

Address	Contents
3030	003600
3600	103000
6390	00C303
C303	003030

Q.1(c) The variables ALPHA, BETA and GAMMA are arrays of 100 words each. Write a subroutine in SIC/XE to add together the corresponding elements of ALPHA and BETA and store the results in the elements of GAMMA. [6]

Q.2(a) What data structure should be mandatory to write a *dis-assembler?* Explain in brief. [2]

PTO

Q.2(b) Reverse-engineer the 3-byte SIC/XE instruction (Object Code in Hex) and obtain the equivalent Assembly language source instruction. Also explain what it does. [4]

072FFD

Assume the following:
(i) this instruction is located at 2000 (hex). So PC = 2003 (hex)
(ii) B-register = 2020 (hex)
(iii) Opcodes (in Hex): LDA=00, LDX=04, LDL=08, STA=0C
(iv) SIC/XE instruction format is as given in beginning.
(v) Note that displacement can be negative number.

Q.2(c) Consider an array of 20 words. Assume that you want to store some value in the 1st index (array[0]=value). A simple SIC **relocatable** (denoted by start address of 0) program is given below. [6]

```
test    start   0
first   ldx     index
        lda     value
        sta     array,X
array   resw    20
value   word    64
index   word    0
```

Convert this assembly program into Object program (output of the Assembler).
Show only the Text and Modification records. Assume the following:
(i) Formats of records are as given in the beginning.
(ii) Opcodes (in Hex): LDX=04, LDA=00, STA=0C.
(iii) resw reserves 20 words.

Q.3(a) What is the format of Modification records? [2]

Q.3(b) Consider 2 SIC instructions at shown memory locations: [4]

```
2000    var1    word    25
2003    var2    EQU     25
```

What will be the entries in the Symbol Table at the end of Pass-1 of the Assembler? Note that Symbol Table contains only one entry for each label.

Q.3(c) State and explain machine independent assembler features and Write short notes on SPARC Assembler. [6]

Q.4(a) The Linking-Loader links different Control Sections (appearing in 1 or more object files) and loads them into Memory. Some changes are required for the Linkage-Editor, which creates a single Control Section (in a single object file) after linking different Control Sections. There is no loading. Describe only the changes (if any) to the Text and Modification records in the output (single) Control Section from those in the input (multiple) Control Sections. [2]

Q.4(b) Consider a very simple Relocatable SIC/XE Object Program: [4]

```
H.COPY .000000.001077
T.00001D.0D.0F2016.010003.000500.4B10105D
T.001070.07.3B2FEF.4F0000.09
M.000024.04.+COPY
M.000027.05.+COPY
E.00001D
```

Format is as given in the beginning. Suppose that at the time of loading, the Operating System returns an address of 4000 (Hex) as Load Point. Answer the following:
(i) Show the locations and contents of the 2 T-records mentioned above. Pay attention to the M-records when showing the contents.
(ii) What is the PC (Program Counter) set to at the end of loading?

Q.4(c) Consider TWO different SIC/XE Object Programs/Control Sections (PROGA, PROGB), which are presented to the Linking-Loader: [6]

H.PROGA .000000.000063	H.PROGB .000000.00007F
D.LISTA .000040.ENDA .000054	D.LISTB .000060.ENDB .000070
R.LISTB .ENDB	R.LISTA .ENDA
T.000020.07.03201D.77100004	T.000036.07.03100000.772027
T.00005D.06.000014.FFFFC0	T.000079.06.FFFFF0.000060
M.000024.05.+LISTB	M.000037.05.+LISTA
M.00005D.06.-ENDB	M.000079.06.+ENDA
M.00005D.06.+LISTB	M.000079.06.-LISTA
E.000020	E

Formats are as given in the beginning. The following is known:
(i) 1st T-record in each has 2 instructions. Also the 2nd instruction is LDT (opcode=74) and is referring to the same memory location in both control sections.
(ii) 2nd T-record in each has data (2 words). Note that they can be negative (2's complement).
(iii) The Operating System returns an address of 1000 (hex) as Load Point.

Answer the following questions:
(1) Draw the External-Symbol Table (ESTAB) generated by the Pass-1 of the Linking-Loader. It should have Symbol Name and Address for each External Symbol and Control Section.
(2) Show the 4 items (2 instructions + 2 data) in memory for both control sections. Pay attention to M-records when writing the contents.
(3) Are there any duplicate data items in the 2 control sections?
(4) Prove that 2nd instruction is loading the T-register from same memory in each control section.

Q.5(a) What is dynamic linking? [2]
Q.5(b) Explain a simple boot strap loader and discuss on it's design issues. [4]
Q.5(c) What are the different machine independent loader features? Explain. [6]

Q.6(a) Differentiate between Function and Macro. [2]
Q.6(b) Explain the process of macro definition and expansion by means of an example giving emphasis on the data structures involved in the process. [4]
Q.6(c) Write an algorithm for a 2pass macro processor in which all macro definitions are processed in first pass and all macro invocations are expanded in 2nd pass. [6]

Q.7(a) What is SRS? [2]
Q.7(b) What are the different testing strategies can be followed for producing successful system software's? [4]
Q.7(c) How will you make a modular design for macro-processor? State the name of modules and their jobs with interaction diagram. Justify your design principle. [6]

:::::26/04/2019 E::::