

# Computer Science & Engineering Department

I. I. T. Kharagpur

Compiler Lab (CS39003)

3rd Year CSE

5th Semester

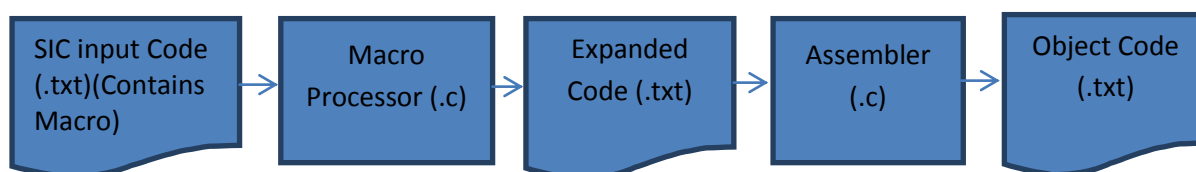
## Assignment 5: Implementation of Macro Assembler

**Submission Deadline: 7<sup>th</sup> September 2013, Midnight.**

**Marks: 15**

In this assignment, you are asked to design a macro assembler which consists of two individual modules; first module implements a macro processor and the second one implements an assembler. The macro processor should take the SIC assembly language program (containing Macro instructions) as an input and generate the Expanded SIC code in .txt format. This Expanded SIC code would be the input to the assembler (which you designed in Assignment 2) and the assembler would generate the object code as the final output.

Similar to Assignment 2, here also you can safely assume that one Text record in the object code is sufficient to contain the entire translated instructions and data. So there will be only one Text record in the translated object code. The diagram given below describes the desired input-output format for this assignment.



Create a single makefile for the macro assembler using the Macro Processor module (./macro) and Assembler module (./assm) to obtain the object code from the input SIC assembly language program.

Your macro processor must support the simple as well as nested macro definitions. It should also perform the basic syntax checks for the macro definitions and invocations (matching MACRO and MEND, matching parameters and arguments etc).

Submit the source and executable codes (macro.c, macro and assm.c, assm respectively) along with the makefile and auxiliary tables. The submission should be accompanied by a Readme file which should contain the detailed specifications of the functionalities supported by your macro processor.

Two sample input SIC programs are provided below (for the testing purpose).

1. Swap two elements

| Label | Instruction | Operand |
|-------|-------------|---------|
| TEST  | START       | 2000h   |
| SWAP  | MACRO       | &I, &J  |
|       | LDA         | &I      |
|       | STA         | TEMP    |

|      |      |       |
|------|------|-------|
|      | LDA  | &J    |
|      | STA  | &I    |
|      | LDA  | TEMP  |
|      | STA  | &J    |
|      | MEND |       |
|      | LDA  | ONE   |
|      | STA  | I     |
|      | LDA  | FIVE  |
|      | STA  | J     |
|      | SWAP | I, J  |
| I    | RESW | 1     |
| J    | RESW | 1     |
| TEMP | RESW | 1     |
| ONE  | WORD | 1     |
| FIVE | WORD | 5     |
|      | END  | FIRST |

## 2. Addition, Subtraction and Multiplication using nested macro

| Label  | Instruction | Operand |
|--------|-------------|---------|
| TEST   | START       | 2000h   |
| SPOPT  | MACRO       | &OP     |
| PEROPT | MACRO       | &X      |
|        | OP          | &X      |
|        | MEND        |         |
|        | MEND        |         |
|        | LDA         | TWO     |
|        | SPOPT       | ADD     |
|        | PEROPT      | FIVE    |
|        | SPOPT       | MUL     |
|        | PEROPT      | NINE    |
|        | STA         | REST    |
| TWO    | WORD        | 2       |
| FIVE   | WORD        | 5       |
| NINE   | WORD        | 9       |
| REST   | RESW        | 1       |
|        | END         | FIRST   |