NATIONAL INSTITUTE OF TECHNOLOGY GOA

SYSTEM PROGRAMMING ASSIGNMENT

SIC/XE ASSEMBLER THAT SUPPORTS PROGRAM RELOCATION AND SYMBOL DEFINING STATEMENTS

Team member 1: Pritika Barshilia

Team member 2: Shreya Maniksha Parsekar

Computer Science and Engineering, Batch of 2024

SIC/XE ASSEMBLER THAT SUPPORTS PROGRAM RELOCATION AND SYMBOL DEFINING STATEMENTS

Introduction

BASIC SIC/XE

- SIC/XE stands for **Simplified Instructional Computer Extra Equipment or Extra Expensive**. This computer is an advanced version of SIC. Both SIC and SIC/XE are closely related to each other that's why they are Upward Compatible.

SIC/XE machine architecture:

1. Memory

- Memory consists of 8 bit-bytes and the memory size is 1 megabyte (2^{20} bytes). The memory size of a Standard SIC machine is very small (2^{15} bytes). This change in the memory size leads to change in the instruction formats as well as addressing modes. Three consecutive bytes form a word (24 bits) in SIC/XE architecture.
- All addresses are byte addressable and words are addressed by the location of their lowest numbered byte.

2. Registers

- SIC/XE Machine contains 9 registers (5 SIC registers + 4 additional registers). The four additional registers are:

	Mnemonics	Use of Register
1.	B	Base register
2.	S	General working register
3.	T	General working register
4.	F	Floating-point accumulator

3. Data Formats:

- Integers are represented by Binary numbers.
- Characters are represented using ASCII codes.
- Floating points are represented using 48-bits.

4. Instruction formats:

- In SIC/XE architecture, there are 4 types of addressing formats available
- The bit **'e'** is used to distinguish between Formats 3 and Formats 4; e=0 implies Format 3 addressing and e=1 implies Format 4 addressing is being used

5. Addressing Modes:

Format 3 addressing is relative. It can be base relative or PC (program counter) relative

Mode	Indication	Target Address (TA)
Base Relative	b=1, p=0	Base (B) + displacement
PC Relative	b=0, p=1	PC value (PC) + displacement

Target address is the effective address of the instruction.

6. Instruction Set:

- In SIC/XE, all the instructions from the SIC architecture are retained, however the additional floating point data format provides Floating point Arithmetic functions as well.

*Compilation Procedure

Steps to compile and run the program:

1. Open the terminal.

2. Change the directory to the SIC-XE-Assembler folder.

3. Run the command – python3 Assembler.py "input1.txt".

4. Intermediate file will be generated in the directory as "Intermediate.txt" and object program will be generated as "Object_Program.txt".

*Generated Files and Data Structures

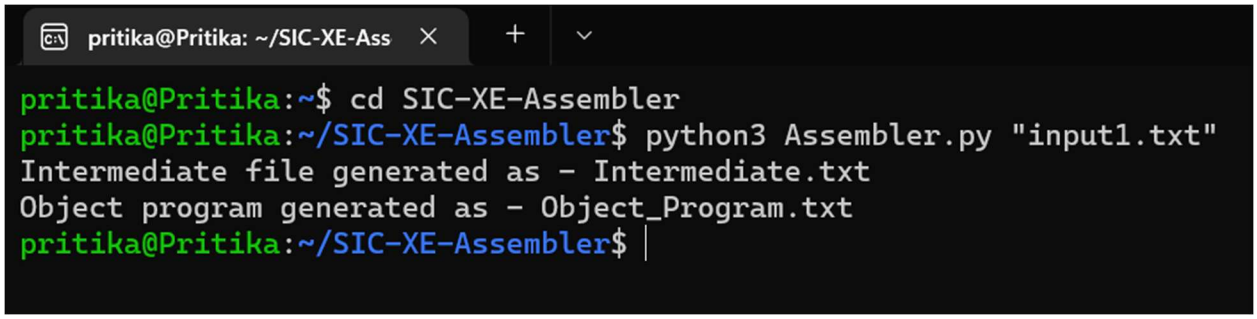
1. After Pass One:

- Intermediate.txt: contains the address of each instruction generated after pass one.
- SYMTAB: Python dictionary that contains all the symbols defined in the program by the user.

2. After Pass Two:

- Object_Program.txt: text file generated which reflects all the object codes generated in a sequential order in ASCII format.
- OBJ_CODES: Python dictionary that contains all the instructions in the program with their corresponding object codes as their values

*Screenshots of Compilation Process for Reference



*Object Program generated