

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

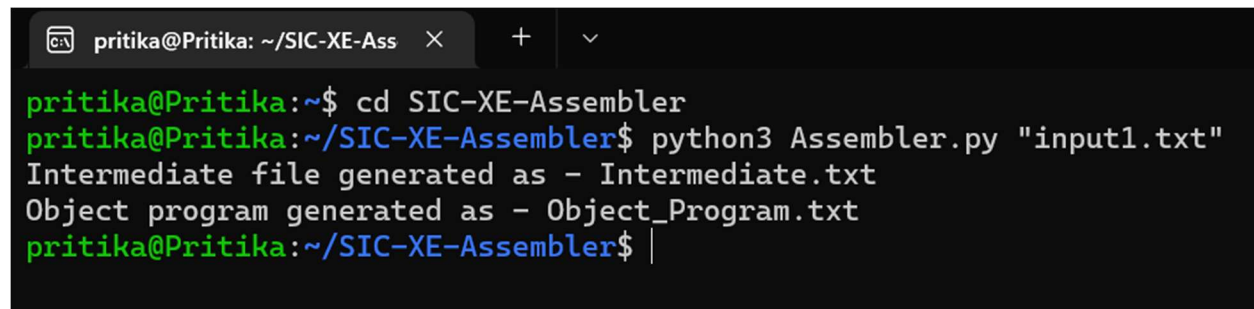
1. After Pass One:

- 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

A terminal window with a dark background. The title bar shows 'pratika@Pritika: ~/SIC-XE-Ass' with window control buttons. The terminal text shows the user navigating to the 'SIC-XE-Assembler' directory and running 'python3 Assembler.py "input1.txt"'. The output indicates that an 'Intermediate file' and an 'Object program' have been generated with their respective filenames.

```
pratika@Pritika:~$ cd SIC-XE-Assembler
pratika@Pritika:~/SIC-XE-Assembler$ python3 Assembler.py "input1.txt"
Intermediate file generated as - Intermediate.txt
Object program generated as - Object_Program.txt
pratika@Pritika:~/SIC-XE-Assembler$ |
```

*Object Program generated

```
Object_Program - Notepad
File Edit View
H^COPY ^000000^00107A
T^000000^1D^172030^692030^4B101039^032029^290000^332007^4B101060^3F2FEC^032010
T^00001D^13^0F2019^010003^0F2010^4B101060^3E2006^454F46
T^001039^1D^B410^B400^B440^75104096^E32019^332FFA^DB2013^A004^332008^57C003^B850
T^001056^1D^3B2FEA^134000^4F0000^F1^B410^774000^E32011^332FFA^53C003^DF2008^B850
T^001073^07^3B2FEF^4F0000^05
M^000007^05
M^000014^05
M^000027^05
E^000000
```