The Hashemite University
Department of Computer Engineering

Systems Programming
Simulator Project Report

Team work :

مجد الدين معتصم عبد الكريم كتوعة.....2030254
زيد محمد عبد اللطيف سمامعة 1933504
حمزة موفق أحمد سمامعة 1837557
محمد رائد محمد شفيق داود 2030154
رزان ياسر محمد ابراهيم سدر 1838058

1 Introduction..... 4
1.1 about standerd SIC and SIC/XE……..……4
1.2 Standard SIC and SIC/XE Format ……..5

1 Introduction

In this project , we design and implement by C++ the simulator for SIC and SIC/XE

1.1 About standard SIC and SIC/XE

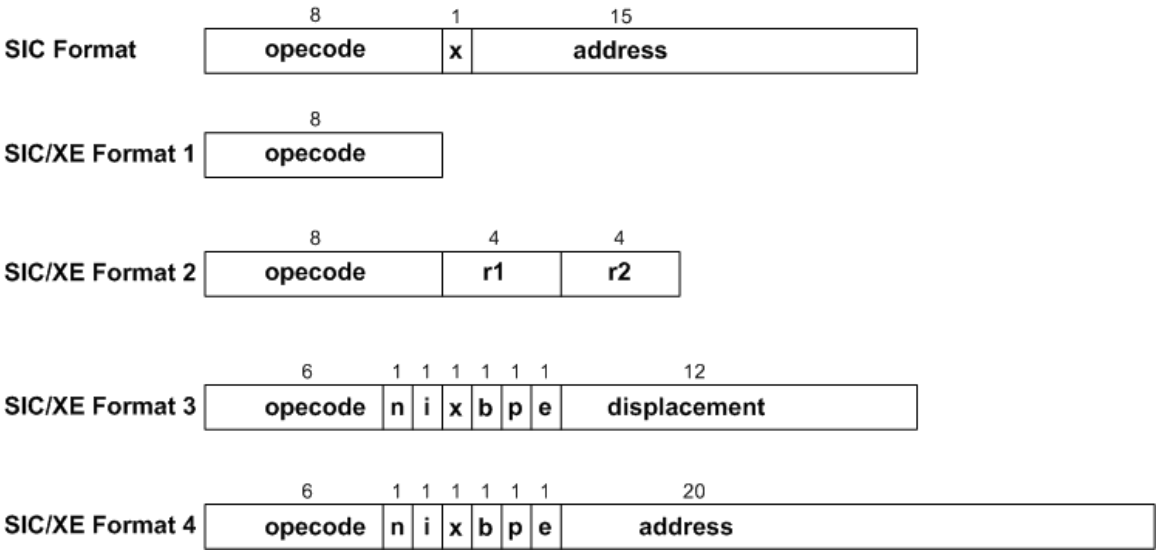
The SIC machine has basic addressing, storing most memory addresses in hexadecimal integer format. Similar to most modern computing systems, the SIC architecture stores all data in binary and uses the two's complement to represent negative values at the machine level. Memory storage in SIC consists of 8-bit bytes, and all memory addresses in SIC are byte addresses. Any three consecutive bytes form a 24-bit 'word' value, addressed by the location of the lowest numbered byte in the word value. Numeric values are stored as word values, and character values use the 8-bit ASCII system. The SIC machine does not support floating-point hardware and has at most 32,768 bytes of memory. There is also a more complicated machine built on top of SIC called the Simplified Instruction Computer with Extra Equipment (SIC/XE). The XE expansion of SIC adds a 48-bit floating point data type, an additional memory addressing mode, and extra memory (1 megabyte instead of 32,768 bytes) to the original machine. All SIC assembly code is upwards compatible with SIC/XE .

- **SIC machines have several registers, each 24 bits long and having both a numeric and character representation:**
    - **A (0)**: Used for basic arithmetic operations; known as the accumulator register.
    - **X (1)**: Stores and calculates addresses; known as the index register.
    - **L (2)**: Used for jumping to specific memory addresses and storing return addresses; known as the linkage register.
    - **PC (8)**: Contains the address of the next instruction to execute; known as the program counter register.
    - **SW (9)**: Contains a variety of information, such as carry or overflow flags; known as the status word register.

- **In addition to the standard SIC registers, there are also four additional general-purpose registers specific to the SIC/XE machine:**
    - **B (3)**: Used for addressing; known as the base register.
    - **S (4)**: No special use, general purpose register.
    - **T (5)**: No special use, general purpose register.
    - **F (6)**: Floating point accumulator register (This register is 48-bits instead of 24).

1.2 Standard SIC and SIC/XE Format

We have 5 Instruction Format for SIC machine , 1 for standard SIC and 4 for SIC/XE .



Flag Bit Description :

FLAG Bit	Description
n,i	(n,i) = (0,1) → immediate addressing (n,i) = (1,0) → indirect addressing (n,i) = (0,0) or (1, 1) → simple addressing
x	x=1 indicates index addressing TA= (x) + address
p	p=1 indicates PC relative addressing TA = (PC) + disp -2048 ≤ disp ≤ 2047
b	b=1 indicates base-relative addressing TA= (B) + disp 0 ≤ disp ≤ 4095
e	Format-3 (e=0) or Format-4 (e=1)

2 User Guide

The programmer guide should discuss the following topics in sufficient details

- SIC/XE Machine
  - Registers (A,X,L,PC,SW,B,S,T,F)
  - Addressing modes (simple)
- Running the simulator
  - Installation (just run the program and watch output.txt)
  - Input files and output file (input : Input.txt , output: Output.txt)
- Debugging
  - Wrong Opcode
  - Wrong Format

3 Programmer Guide

The programmer guide should discuss the following topics in sufficient details

- Overview of the program flow
  The eprogram fetching command ,start and last from input.txt then partitionate commands into hexCode,opcode,n,I,x,p,b,e,address and target address , then run execute function which is execute the command and update the corresponding register with its new value
  ****note : output in decimal NOT hexa**
- Data structure
  - Classes (Command Class)
  - Procedures/methods (execute,loadInput,decode,hex_to_bin,convert_dec_to_hexadd_hex…etc)

4 Testplan

List the features of the design. Describe how those features are tested.

Feature	**Register**
LDA	A
LDB	B
LDL	L
LDS	S
LDT	T
LDX	X
ADD	A
COMP	C
STA	MEM