

SIC/XE Simulator의 구현

System Programming Project #2

Jaeyoung Choi

choi@ssu.ac.kr

Contents

1. 프로젝트 목표
2. 필수 구현기능
3. 프로그램 전체 구조
4. GUI 환경 구현 예제
5. 코드 예시

1. 프로젝트 목표

- SIC/XE 기계의 동작을 수행할 시뮬레이터 개발
- SIC/XE Simulator
 - ◆ 입력
 - SIC/XE 어셈블러를 통해 만들어진 object code
 - ◆ 출력
 - 해당 Object 코드에 따른 프로그램 실행의 결과

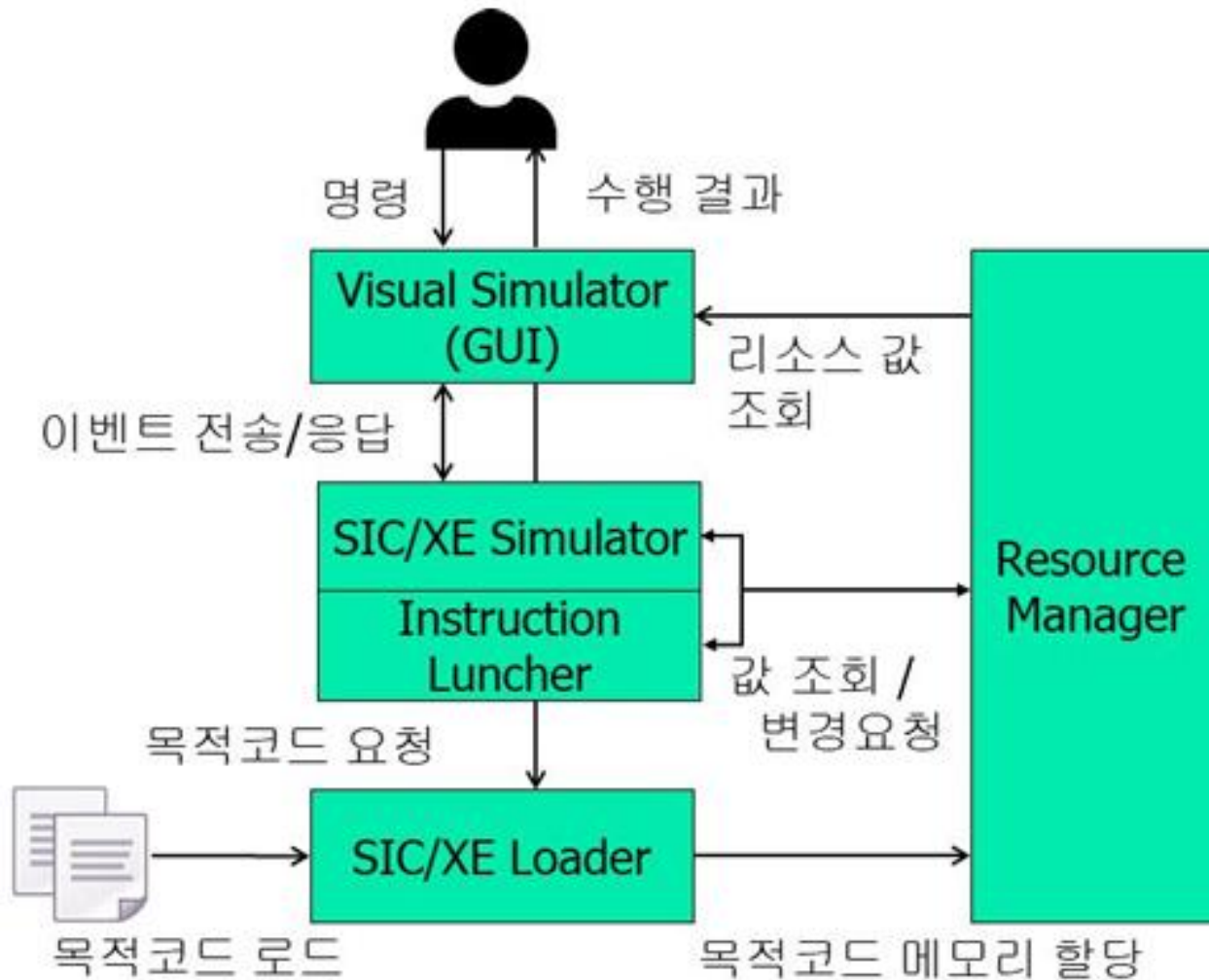
2. 필수 구현 기능

- 교재에 나온 모든 SIC/XE 명령어 처리
- 모니터링 구현
 - ◆ Object Code 별 모니터링
 - ◆ 단계별 Instruction 수행과정
 - ◆ Register 모니터링

3.1 프로그램 전체 구조 설명

- Visual Simulator
 - ◆ 사용자 인터페이스
 - ◆ 이벤트를 SIE/XE Simulator에게 전달
- SIC/XE Simulator
 - ◆ 명령어의 실질적인 수행
 - ◆ SIC/XE Loader 및 Resource Manager 컨트롤
- Instruction Launcher
 - ◆ SIC/XE 머신이 명령어에 따라 수행할 동작 정의
- SIC/XE Loader
 - ◆ 가장 먼저 수행되는 모듈
 - ◆ Object code를 memory 영역에 로드
- Resource Manager
 - ◆ 가상의 메모리, 레지스터, 디바이스 관리

3.2 프로그램 전체구조도



3.3 단계별 실행과정

1. SIC/XE Loader가 Object code를 메모리에 로딩
2. SIC/XE Simulator가 instruction을 가져와서 해석
(fetch/decode)
3. Instruction의 요구 데이터를 Resource Manager 에게 요청
4. 받은 데이터를 이용하여 SIC/XE Simulator가 연산 수행
 - ◆ 이 과정에서 resource (register, memory, device)의 상태 값 변경 발생
5. Object code가 남아 있으면 1~4번과정 반복

4.1 GUI 환경 예제 – main 화면

FileName :

H (Header Record)

Program name :

Start Address of Object Program :

Length of Program :

E (End Record)

Address of First instruction in Object Program :

Start Address in Memory :

Target Address :

Register

	Dec	Hex
A (#0)	<input type="text"/>	<input type="text"/>
X (#1)	<input type="text"/>	<input type="text"/>
L (#2)	<input type="text"/>	<input type="text"/>
B (#3)	<input type="text"/>	<input type="text"/>
S (#4)	<input type="text"/>	<input type="text"/>
T (#5)	<input type="text"/>	<input type="text"/>
F (#6)	<input type="text"/>	<input type="text"/>
PC (#8)	<input type="text"/>	<input type="text"/>
SW (#9)	<input type="text"/>	<input type="text"/>

Instructions :

사용중인 장치

Log (명령어 수행 관련):

4.2 GUI 환경 예제 – 로드 결과

FileName :

H (Header Record)

Program name :

Start Address of Object Program :

Length of Program :

E (End Record)

Address of First Instruction in Object Program :

Start Address in Memory :

Target Address :

Register

	Dec	Hex
A (#0)	<input type="text" value="0"/>	<input type="text" value="0"/>
X (#1)	<input type="text" value="0"/>	<input type="text" value="0"/>
L (#2)	<input type="text" value="0"/>	<input type="text" value="0"/>
B (#3)	<input type="text" value="0"/>	<input type="text" value="0"/>
S (#4)	<input type="text" value="0"/>	<input type="text" value="0"/>
T (#5)	<input type="text" value="0"/>	<input type="text" value="0"/>
F (#6)	<input type="text" value="0"/>	<input type="text" value="0"/>
PC (#8)	<input type="text" value="0"/>	<input type="text" value="0"/>
SW (#9)	<input type="text" value="0"/>	<input type="text" value="0"/>

Instructions :

172027
4B101033
032023
290000
332007
4B10105E
3F2FEC
032016
0F2016
010003
0F200A

사용중인 장치

Log (명령어 수행 관련):

4.3 GUI 환경 예제 - 실행화면

FileName :

H (Header Record)

Program name :

Start Address of Object Program :

Length of Program :

E (End Record)

Address of First Instruction in Object Program :

Start Address in Memory :

Register

	Dec	Hex
A (#0)	<input type="text" value="70"/>	<input type="text" value="000046"/>
X (#1)	<input type="text" value="3"/>	<input type="text" value="000003"/>
L (#2)	<input type="text" value="39"/>	<input type="text" value="000027"/>
B (#3)	<input type="text" value="0"/>	<input type="text" value="000000"/>
S (#4)	<input type="text" value="0"/>	<input type="text" value="000000"/>
T (#5)	<input type="text" value="3"/>	<input type="text" value="000003"/>
F (#6)	<input type="text" value="000000"/>	
PC (#8)	<input type="text" value="0"/>	<input type="text" value="000000"/>
SW (#9)	<input type="text" value="000002"/>	

Instructions :

Log (명령어 수행 관련) :

JEQ

LDCH

WD

TIXR

JLT

RSUB

J

5.1 코드 예시 – GUI 이벤트 처리

```
open_btn = new JButton("open");
open_btn.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(final MouseEvent e) {
        final FileDialog fd = new FileDialog((Frame) thisobject, "파일열기", FileDialog.LOAD);
        fd.setVisible(true);
        if(fd.getDirectory() != null){
            rMgr = new ResourceManager_m();
            object_code_file = new File(fd.getDirectory()+fd.getFile());

            //all_btn, one_btn을 누를 수 있게 만든다.
            all_btn.setEnabled(true);
            one_step_btn.setEnabled(true);

            //object file을 open하면 initialize함수를 호출.
            initialize(object_code_file, rMgr);
            index = 0;

            update(); //화면에 보여주는 내용을 갱신.
        }
    }
});
```

5.2 코드 예시 – 로더 입력처리

```
else if(line.charAt(o) == 'H') {  
    //H와 관련된 정보를 Resource manager를 통해서 저장  
    rMgr.setProgname(line.substring(1, 7), currentSection);  
    rMgr.setProgLength(line.substring(13,19), currentSection);  
    rMgr.setStartADDR(currentSection);  
  
    // SYMTAB 등록  
    define.varName[define_record] = line.substring(1, 7);  
    define.varAddress[define_record++] =  
    Integer.parseInt(rMgr.getStartADDR(currentSection), 16);  
  
    currentSection++;  
    modify_record = 0;  
}
```

5.3 코드 예시 – instruction 처리(1)

```
if(op_int == 0) { //LDA
    // nixbpe에 유의하여 구현
    if(e_flag == 1)
        rMgr.setregister(7, (rMgr.getRegister(7) +
            Integer.parseInt(objcode.substring(loc, loc + 5),16)));
    else
        rMgr.setregister(7, (rMgr.getRegister(7) +
            Integer.parseInt(objcode.substring(loc, loc + 3),16)));

    ...(다음장에서 계속)
```

5.4 코드 예시 – instruction 처리(2)

```
switch(ni_bit) {  
    case 1 :  
        if(e_flag == 1)  
            rMgr.setregister( 0,(Integer.parseInt (objcode.substring(  
                loc, loc + 5),16)));  
        else  
            rMgr.setregister( 0, (Integer.parseInt(objcode.substring(  
                loc, loc + 3), 16)));  
            rMgr.setregister(7, (rMgr.getRegister(7)));  
            break;  
    case 2 :  
        break;  
    case 3 :  
        rMgr.setregister(0, Integer.parseInt(rMgr.getmemory().substring(  
            rMgr.getRegister(7) * 2, Mgr.getRegister(7) * 2 + 6), 16));  
        break;  
}  
}
```