CSE305
Project 2. Building a Simple MIPS Emulator Report

201611057 김준우

1. Introduction

이 프로그램은 MIPS(Big-endian) 어셈블리 코드를 바이너리 코드로 변환된 'object.o'파일을 실생시켜주는 MIPS ISA 에뮬레이터를 구현한 것이다.

2. Environment

VMWare Workstation 16을 사용하여 Ubuntu 20.04.4를 구동하였으며, Ubuntu 내부에서는 VSCode로 코딩을 한 후, 터미널을 통해 구동을 확인하였다. 사용한 언어는 C++이며 컴파일 환경은 c++ 9.4.0 version이다. 프로그램은 Ubuntu의 Terminal 콘솔창에서 c++ -o runfile project2.cpp 명령어로 컴파일 후 실행이 가능하다.

3. Explanation

해당 프로그램의 코드는 main함수 영역과 연산처리 과정을 돕는 함수로 구성되어있으며, 사용한 STL은 [iostream, fstream, vector, bitset, string, algorithm, vector, sstream, stdlib.h, cstring, cmath] 이다.

각각의 MIPS operator에 따른 연산처리 과정은 아래의 세 함수를 통해 연산이 처리된다.

void r_type_oper(string str)	string type인 str을 input으로 받아, operator를 구분하여 각 함수에서 적절한 연산처리를 한다.
void I_type_oper(string str)	
void j_type_oper(string str)	

그리고 연산된 결과는 메모리 또는 레지스터에 저장되는데, 이는 vector구조로 구현하였다.

또한 연산에 도움을 주는 간단한 함수를 아래와 같이 정의하였다.

int complement2_16bit(string bits)	16비트인 2의 보수를 정수로 변환
string hex_to_bi(string hexnum)	16진수를 2진수의 string으로 변환
string detect_oper(string input_str)	바이너리코드를 받아 operator를 구분해주는 함수

메인함수에서는 바이너리 코드의 파일을 line별로 구별하여 각각의 알맞은 vector로 분류를 실행한 후, operator의 연산이 진행이 된다. 연산이 진행이 되며, '-d'옵션이 있는 경우 및 '-m' 옵션이 있는 경우를 고려하여 코드를 구성했다. 또, '-d'옵션이 없는 경우에 연산과정이 종료된 이후에 레지스터와 해당되는 메모리 값을 출력하도록(-m옵션이 있는 경우) 설정하였다.

프로그램은 우분투의 Terminal 콘솔창에서 c++ -o runfile project2.cpp로 컴파일 후 실행이 가능하다. 컴파일을 진행한 후 [$./runfile -m 0x40000:0x40040 -n 25 sample.o] , [$./runfile -m 0x40000:0x40040 -d -n 25 sample.o] 와같은 명령어로 실행할 수 있다. 이때 각 실행 옵션에 따라 알맞은 출력결과가 콘솔창에 나타난다.

4. Results

프로그램은 정상적으로 실행되었다.