<Computer architecture assignment – project #1 Simple calculator>

컴퓨터 구조론 2분반 32191597 박민규

**1. 과제 개요**: 이 과제는 컴퓨터 구조론 시간에 배운 Instruction Set Architecture(ISA)를 활용하여, 간단한 계산기 프로그램을 만드는 것이다. ISA란 CPU가 인식해서 기능을 이해하고 실행할 수 있는 기계어 명령어를 말한다. CPU가 이해할 수 있는 instruction set 이 memory에 저장되어 있고, CPU 안에 있는 program counter(pc)가 명령어를 가리키면, 그 명령어를 CPU가 알아들을 수 있도록 decoding 하여 실행하는 구조를 말한다. 컴퓨터엔 다양한 instruction 들이 저장되어 있고, MIPS CPU를 예로 들면 ADD, SW, LW, JUMP, BEQ 등이 있다. 이 instruction 들은 단독으로 사용되기도, 혹은 같이 사용되기도 하며 사용자가 작성한 코드 또는 다양한 명령들을 수행한다. 하나의 instruction을 실행하는데 걸리는 시간을 넉넉히 잡아 두고, 1cycle 안에 모든 instruction이 실행될 수 있도록 하는 컴퓨터를 Single-cycle Machine이라 하고 1개의 cycle을 잘게 나누어서 1개의 instruction이 여러 cycle 안에 실행될 수 있도록 하는 Multi-cycle Machine라고 한다. 현대에는 대부분 후자인 Multi-cycle Machine을 사용하는데, instruction마다 실행되는 실제 시간을 유연하게 조정할 수 있기 때문이다.

**2. 프로그램 개요**: 이 프로그램은 “input.txt” 파일 안에 있는 여러가지 명령들을 받아와서 실행시킨다. 여러가지 명령이란 여기서 사칙연산, 두 값을 비교하는 compare, 원하는 위치로 이동하는 Jump, 조건이 맞다면 Jump를 하는 Branch 등이 있다. 또한 “gcd.txt” 파일에서 앞서 선언한 instruction 들을 활용하여 gcd계산을 하는 프로그램도 만들었다. gcd 계산을 하던 중에 두 값이 같은 지 비교하는 instruction이 추가로 필요 해져 same instruction을 만들었다.

**3. 어려웠던 점 및 아쉬운 점**: 이 과제를 하면서 몇가지의 난관이 있었는데, 첫 난관은 Jump instruction을 만들 때였다. 처음엔 Jump를 어떻게 구현할 지 막막해서 1주동안 그 고민만 한 것 같다. 그런데 어느 순간 아이디어가 떠올라서 구현은 하였으나, 애초 생각했던, 교수님이 원하는 형태의 Jump instruction은 아닐 것 같다는 생각이 들었다. 당초 생각했던 모습은 .txt 파일 각 행의 문자열 시작 부분을 가리키는 pointer array를 만들고, pc의 역할을 하는 변수가 값이 바뀌면서 해당 줄을 실행하는 모습으로 구현하고 싶었고, 이게 교수님의 의도인 것 같았으나 코드로 작성하는 과정에서 방법이 도저히 떠오르지 않아 다른 방법으로 선회했다. 두 번째 난관은 .txt 파일에 입력이 16진수 숫자의 형태(0xa)가 아닌 register 형태(R0)가 올 때(M R1 0x3 이 아닌 M R1 R3 과 같은 형태) 처리를 어떻게 해야 할 지 고민했으나, switch-case 문을 이용해 모든 상황별로 case를 나누어 해결했다. 마지막으로 아쉬운 점은, 코드가 실행되는 것 만을 목표로 삼아 spaghetti code가 되어 버렸다는 것이다. 코드에 이렇다 할 체계가 없고, 몇가지의 모순점들도 발견되었다. 3주는 충분한 시간이었으나, 이런 디테일까지 손보지 못했다는 아쉬움도 남는 과제였다. 하지만 교수님이 요구한 사항은 모두 구현하였고, 중간에 어려웠던 점을 혼자 해결했다는 부분에서 내 스스로 의미가 큰 과제였던 것 같다.

**4. 주요 코드 설명**: 프로그래밍 언어는 C언어를 선택하였고, windows 환경에서 작성하였다. 먼저 소스코드는 크게 main.c, instcution.c 로 나뉜다. 처음엔 모든 instruction 들을 instruction.c 파일에 넣고자 하였으나, Move instruction을 만들 때 C언어에는 JAVA에서의 puclic static 과 같이 다른 파일에서도 함께 쓸 수 있는 변수가 없다 보니, 몇 개의 instruction들은 main.c에 같이 들어가 있다. 사실 이 부분도 main 부분과 instruction 부분을 더 깔끔하게 나눌 수 있는 방법이 있지 않았을까 하는 생각이 든다. 주요한 대부분의 코드들은 main.c에 있고 여기서 .txt파일의 입력을 받아 여러 명령어들을 실행한다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 화면이(가) 표시된 사진

자동 생성된 설명

main.c 파일은 .txt파일의 라인의 총 개수를 반환해주는 함수와, move instruction을 실행하기 위한 move 함수로 시작된다. move 함수는 char type 과 long type 두 개의 파라미터를 받아와 원하는 register에 대입을 해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명openAndReadFile 함수는 이 프로그램에서 main logic을 맡고 있는 함수이다. Int형 매개변수를 하나 받아와서 몇 번째 줄부터 실행시킬 것인지 판단한다. 이는 Jump instruction을 처리하기 위한 방법으로 사용된다. 현재 currentLine의 값이 매개변수로 받은 line값보다 작다면 반복문을 넘어가게 해 사용자가 원하는 줄부터 실행할 수 있게 하였다. 밑의 while문 안에서 공백을 기준으로 나눈 문자열을 포인터 변수에 저장시키고, 각각의 문자열들을 operation 2차원 배열에 텍스트이(가) 표시된 사진

자동 생성된 설명저장하였다.

위 코드에서 calculator instruction을 처리하였는데, operation[0][0]값이 사칙연산이라면, 밑의 코드를 실행시킨다. 여기서 사칙연산을 할 때 쓰는 입력 register는 무조건 R1, R2를 이용하게 하였다. R1 또는 R2에 값이 들어가 있다면 바로 더한 값을 R0에 대입시켜주고, 16진수 숫자가 들어온다면 R1, R2에 문자열을 16진수로 변환해 준 값을 대입해 준 후 계산해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명다음으로 move를 처리하는 과정이 나오는데, 앞서 어려웠던 점에서 말한 move를 처리할 때 16진수 형태가 아닌 register형태가 나올 때를 처리해주기 위해서 operation[2][1]값에 따라 switch-case문을 사용하였다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

다음으로 짠 코드는 compare instruction과 jump instruction이다. Compare는 R1, R2레지스터에 저장되어 있는 값을 비교해 앞의 값이 크거나 같다면 1을, 작다면 0을 R0에 대입해 준다. 다음으로 Jump는 몇 번째 줄로 뛰어 넘을 건지 받아온 후, 그 값이 .txt파일 전체 줄 수 보다 작거나 같은 경우에 현재 파일을 닫고 openAndReadFile 함수에 R9값을 매개변수로 넣어 호출한다. 그러면 원하는 위치부터 파일을 다시 읽기 시작하기 때문에 jump가 된 것이라 볼 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트, 화면이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명branch는 R0레지스터 값이 1인 경우에만 jump를 한다. 다음으로 gcd연산을 하기 위해 same instruction을 만들었는데, R1, R2레지스터의 값이 일치한다면 1을 return하고 같지 않다면 0을 return한다.

메인 함수에서는 매개변수 1을 넣어준 openAndReadFile함수를 호출한다.

**5. 실행 화면**

텍스트이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명실행 화면을 보기 위해 input.txt 파일과 gcd.txt파일이 어떻게 구성되어 있는지 살펴본다.

왼쪽의 파일이 input.txt, 오른쪽 파일이 gcd.txt이다. Input.txt파일을 먼저 보면 3번째 줄로 jump한 이후, 쭉 실행하다 6번째 줄에서 R0에 1을 대입하고, 7번째 줄에서 branch를 만나 9번째 줄로 jump하게 된다. 다음 gcd.txt파일은 1번째 줄에 15와 5의 최대공약수를 구하기 위해 앞서 구성한 instruction들을 활용하였고R0에 15와 5의 최대공약수인 5가 들어가게 하였다.

위텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명 첫 번째 사진이 input.txt를 읽어 실행한 결과이고, 아래의 사진이 gcd.txt를 읽어 실행한 결과이다. 두 개 모두 정상적으로 실행된 것을 알 수 있다.

**6. 마치며**: 이 과제를 하며 C언어 에서의 파일 입력 처리, 문자열 처리에 대한 개념을 확실히 알게 되었고 전체적인 프로그래밍 실력이 한단계 성장하는 계기가 되었던 것 같다. 도중에 막히는 부분도 있었고 하고 난 후 아쉬웠던 점도 분명 있었지만 고민하던 문제를 해결했을 때의 기분을 알게 된 나에게 뜻깊었던 과제였다.