

<컴퓨터 구조 pa1 보고서>

1. How do you translate the instructions

과제 명세서를 보면 MIPS instruction 의 format 은 R, I 두개 format 이 있다. 따라서 translate function 에서 instruction_type 함수 호출을 통해 type 이 r 인지 i 인지 구분하게 하였다. 이때 strcmp 를 통해서 입력될 수 있는 모든 경우들을 비교한 뒤에 R type 이면 r 이 return 되고 I type 이면 i 가 return 되게 함으로써 두 format 이 구분된다.

이후 instruction_type 을 통해서 token 의 첫번째가 r type 인지 I type 인지 구분한 뒤에는 R type instruction 일 경우에 R type 을 두가지로 구분해서 rtype_instruction 을 호출하였는데 R 타입인 sll, srl, sra 의 경우에는 rs 의 값이 0 이고, shamt 값이 존재하기 때문이다. 반면 add sub 같은 format 은 shamt 는 0 이고 rs 의 값이 존재한다. 따라서 두가지 케이스를 구분하였다. 마찬가지로 I type 인 경우에는 itype_instruction 을 호출한다.

이후 register_address 함수를 이용해 들어오는 토큰을 각 register 에 알맞는 Binary string 으로 변환해주는데, R type 일 경우 opcode 는 0 으로 세팅을 하고 각 레지스터에 해당하는 Binary 로 이루어진 string 을 만들어서 rs, rt, rd 를 결정하고, getBin 함수를 통해 shamt 의 값을 결정해주고, funct 값을 세팅해 준 뒤 string 을 strtimax 를 통해서 정수형태로 바꿔준 후에 이를 Bit 연산을 통해서 구해야 하는 32 bit binary 로 변환하였고 I type 일 경우에도 비슷하게 opcode 는 iMap 에서 대응되는 string 을 결정해주고 getBin 을 이용하여 immediate value 를 결정해주며 마찬가지로 rs rt 의 값을 결정해준뒤에 다시 정수형태로 변환된 값을 총 32bit 형태의 구하고자하는 binary 로 만들었다. 따라서 마지막 출력시 이를 16 진수 형태로 출력하는 형태로 코드의 흐름이 구성된다.

2. How do you translate the register names to corresponding register numbers

string 단위로 토큰을 끊어서 타입을 구분하였으니 토큰으로 나뉘어 들어오는 register 들에 따라서 각 register 에 맞는 Binary 형태의 string 으로 바꿔준 뒤 이 string 을 다시 이진수로 바꾸고, 추후 32bit 형태로 이루어진 binary 를 16 진수로 바꿔야 한다. 따라서 토큰으로 들어오는 register 들을 register_address 함수를 통해 구조체 배열로 선언된 registerMap 에서 각 register 에 대응되는 binary 가 나타날때 까지 반복문을 실행하며 각 register name 에 대응되는 binary string 으로 바꿔준다.

또한 r type 일때의 funct 값과 I type 일때의 opcode 를 알기 위해서 rMap 과 iMap 을 선언하였다. 결과적으로 토큰으로 구분된 string 에 대한 알맞은 opcode, rs, rt, rd, funct 가 binary 형태의 string 으로 만들어지게 된다.

3. How do you convert shamt/immediate values from strings to corresponding numbers

shamt 와 immediate values 는 각각 R type 일 경우 5 bit, I type 일 경우 16 bit 이다. 따라서 shamt 와 immediate values 에 대응되는 tokens[3]를 알맞게 바꿔줘야 하는데 16 진수 또는 정수 형태로 들어오는 이 string 을 strtimax 를 이용해 long 형태의 값을 int 로 type conversion 을 해서 바꿔준 뒤 이를 getBin 의 parameter 로 넘기게 하였다.

getBin 은 r type 일 경우 5bit 를 세팅해줘야 하기 때문에 padding 이 5 이고, I type 일 경우 16bit 로 padding 이 결정되어서 들어간다. 따라서 각 경우에 mask 를 이용하여 mask 가 0 이 될때까지 mask 를 shift 하며 mask 와 정수로 변환된 토큰을 AND 연산을 통해 clamping 시킨 뒤 값을 경우 '0'에서 1 을 더한 아스키코드값인 '1'을 str 에 저장해주게 하면서 shamt 와 immediate values 를 string 으로 변환하는 함수이다.

따라서 이 함수를 통해서 shamt 와 immediate values 가 이진수 형태의 string 으로 변환되기 때문에 이를 다시 정수로 바꾸고 bit 연산을 통해서 구하고자하는 32bit 의 이진수를 만들 수 있게 된다.

4. Lesson learned (if you have any)

이번 과제를 수행하면서 인터넷에서 많은 검색도 해보고 코드 구현을 하며 수업에서 배운 MIPS assembly 를 machine instruction 으로 변환하는 과정을 복기해볼 수 있었다. 또한 bit 연산에서 !!연산이 0,1 로의 clamping 이라는 것을 알게 되었다. 또한 strtimax 와 atoi 함수를 사용해볼 수 있었다.