어셈블리프로그래밍설계및실습 보고서

실험제목: block data transfer and stack

실험일자: 2016년 10월 13일 (목)

제출일자: 2016년 10월 26일 (수)

학 과: 컴퓨터공학과

담당교수: 이형근 교수님

실습분반: 화5, 목6,7

학 번: 2013722095

성 명: 최 재 은

1. 제목 및 목적
   1. 제목

Block data transfer and stack

* 1. 목적

Sp(r13)의 성질을 이해하고 이를 활용하는 opcode stm, ldm을 사용하는 방법을 배운다.

이를 통해 스택의 역할과 중요성에대해 고찰하고 어셈블리 관점에서의 반복 루틴/서브루틴의 차이점을 이해한다.

1. 설계 (Design)
   1. Pseudo code

**1) problem1**

R1~r7까지 1~7값을 저장

R2를 스택에 저장

R0,r3를 스택에 저장

R5-r7을 스택에 저장

R1, r4를 스택에 저장

스택에 저장된 값을 레지스터로 복사.

**2) problem 2**

R0~r7에 10~17값을 저장

스택에 r0~r7의 값들을 저장

현위치를 lr에 저장하고, doRegister함수 호출

Return 값을 r9에 저장

Return 값을 왼쪽으로 31bit이동시켜서 0인지 확인(홀/짝 확인)

짝수면 even 레이블로 이동

아니면 r9(return)에서 index가 홀수인 값들을 빼주고 결과를 0x40000에 저장

- even

r9(return)에서 index가 짝수인 값들을 빼주고 결과를 0x40000에 저장

-doRegister

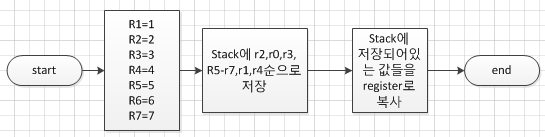
Sp가 가리키는 주소의 값을 r9에 복사하고 sp를 증가시킴

R9\*r8(index)값을 r11(return)에 하고 index를 1 증가시킴(8회 반복)

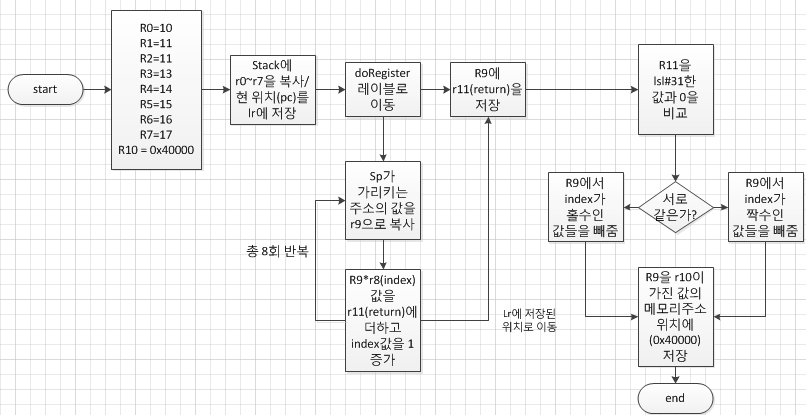
Lr에 저장된 위치로 이동.

* 1. Flow chart 작성

**1) problem1**



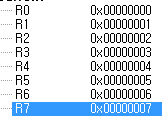
**2) problem2**



* 1. Result

**1) problem1**

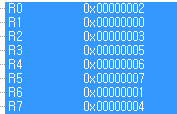
- 레지스터에 임의의 입력값을 저장함.



- 원하는 순서대로 값들이 stack에 저장됨.

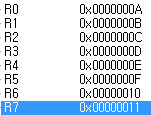


- stack에 있는 값들을 레지스터로 복사.



**2) problem2**

**-** register에 지정된 값들을 저장



-stack에 register들의 값이 복사됨.



- register\* index값들의 합인 return



- return의 홀/짝 여부를 확인하고, 해당되는 연산을 진행한 후의 값을 메모리에 저장



* 1. Performance

1) problem1





- code size는 56이고 state는 30개입니다. Score는 code size \* state^2이므로

이 프로그램의 스코어는 50400이 나오게 됩니다.

2) problem2





- code size는 240이고 state는 93개입니다. Score는 code size \* state^2이므로

이 프로그램의 스코어는 2075760이 나오게 됩니다.

1. 고찰 및 결론
   1. 고찰 및 결론

위의 실험을 진행하면서 (예를 들어) stmfa와 ldmfa가 왜 다른 방식으로 sp이 이동하는가에 대해서 사실상 멘탈이 아스라히 가루가 돼버리는 느낌을 받았지만, 이를 친구와 토의하고 pdf 자료를 다시 한번 확인하면서 개념을 확고하게 안착시켰습니다. 그 외에 홀/짝 여부를 판단하는 것에 있어서 여러가지 방편을 구해보려 했으나 홀/짝 여부가 1의자리 bit값에 따라 판가름 난다고 판단하여 return값을 따로 복사한 뒤 이를 왼쪽으로 31bit이동시켜 그 값이 0인가 아닌가로 홀/짝을 구분하였습니다. 그 이외에는 state를 줄이기 위해서 doRegister를 루핑하지 않고 풀어서 코딩하였습니다.

Stm과 ldm이 한 쌍처럼 사용된다고 개념을 잡았는데, 이를 적절하게 판단해서 실제 코드에 사용해 보았습니다.

4. 참고문헌