어셈블리 프로그래밍 설계 및 실습

실험제목: Block Data Transfer & Stack

실험일자: 2017년 10월 11일 (수)

제출일자: 2017년 10월 25일 (수)

학 과: 컴퓨터공학과

담당교수: 이형근 교수님

실습분반: 월 5, 수 6, 7

학 번: 2014722075

성 명: 이 동 준

1. 제목 및 목적 (3%)
   1. 제목

Block Data Transfer & Stack

* 1. 목적

register와 메모미간 block 단위의 데이터 저장 및 가져오기를 이해하고 stack명령어의 필요성과 효용성을 알아본다.

1. 설계 (Design) (50%)
   1. Pseudo code

\* program1

store value in register

ri = i (0=<i=<7)

decide stack pointer(sp)

store r0~r7 in stack by using empty ascending(push)

load value in stack to each register by using empty ascending(pop)

\* program2

main

store value in register

ri = 10 + i (0=<i=<7)

r9 = 160

doRegister()

doGCD()

r10 = GCD + r4

store value in memory

doRegister

decide stack pointer(sp)

store r0~r7 in stack by using empty ascending(push)

load value in stack to each register by using empty ascending(pop)

copy(using stack)

ri = ri + i (0=<i=<7)

r10 = r0 + ~ + r7

load data before update from r0 to r7(pop)

go to main

doGCD

compare r10,r9

if(r10>r9) r10 = r10- r9

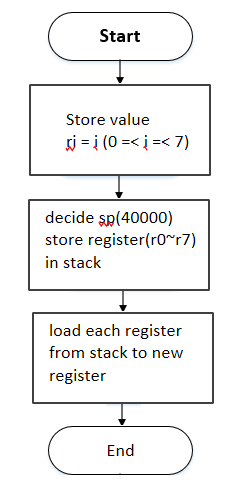
if(r9>r10) r9 = r9 – r10

if(r10 != r9) go to doGCD

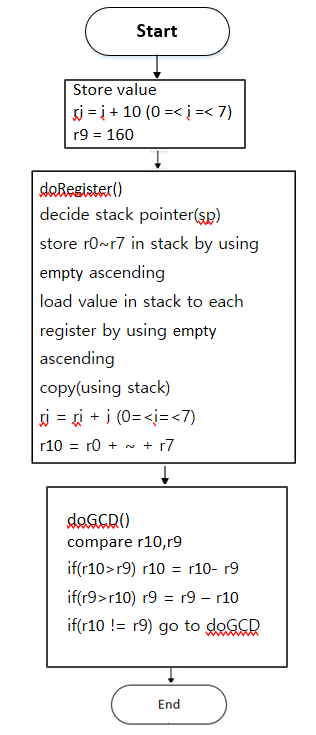
if(same) go to main

* 1. Flow chart 작성

\* program1



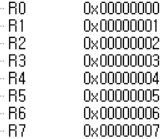
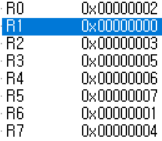
\* program2

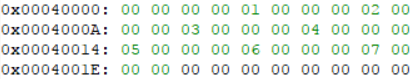


* 1. Result

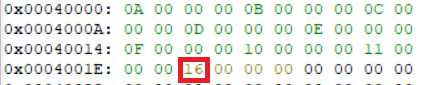
\* program1

- 바꾸기 전 - 바뀐 후



\* program2



GCD = 8, before r4 = 14 -> 8 + 14 = 22 = 16(16진수)

나머지 값들은 stack을 이용한 copy에 사용됨

* 1. Performance

\* program1

- codesize : 60 Bytes, state : 37

\*program2

- codesize : 168 Bytes, state : 130

1. 고찰 및 결론
   1. 고찰 (35%)

Stack을 이용하여 register의 있는 값을 바꾸는 실습과 함수를 만들어서 프로그램을 작성하는 실습을 해보았다. 처음에는 Stack 명령어에 대해 이해하기가 어려웠는데 중간고사를 준비하면서 Stack에 대해서 좀 더 자세히 알게 되었고 이번 실습에 필요한 문법도 익혔다. 더불어 함수를 이용하여 특정 식이나 기능을 수행하였는데 다양한 함수를 구현할 수 있도록 해야 한다.

* 1. 결론 (10%)

register중 sp(stack pointer)의 개념을 확실히 이해해야 하고, stack명령어의 각 기능을 명확하게 구분하여야 한다. 한편 함수를 만들어서 프로그램을 작성할 때 lr(link register)에 대해 좀 더 정확하게 알아야한다.

1. 참고문헌 (2%)

광운대학교 – 어셈블리프로그램설계및실습 – 이형근교수님 – 강의자료