

어셈블리프로그래밍설계 및 실습 보고서

실험제목: Data transfer from/to memories

실험일자: 2023년 09월 19일 (화)

제출일자: 2023년 09월 20일 (수)

학 과: 컴퓨터공학과

담당교수: 이준환 교수님

실습분반: 화 6,7 목 5

학 번: 2022202065

성 명: 박나림

1. 제목 및 목적 (3%)

A. 제목

Data transfer from/to memories

B. 목적

ARM software development tool을 설치하여 사용 방법을 익히도록 한다. 어셈블리어를 공부하여 코드를 작성한다. 그 결과로, 메모리에 데이터가 전송되는 과정을 확인할 수 있도록 한다.

2. 설계 (Design) (50%)

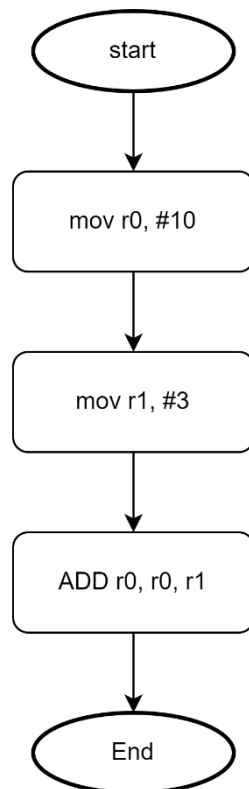
A. Pseudo code

```
AREA    ARMex, CODE, READONLY ;해당 코드를 읽기전용으로 정의
ENTRY  ;프로그램 진입점
start ;프로그램 시작 label

    MOV    r0, #10 ;r0에 10을 저장
    MOV    r1, #3  ;r1에 3을 저장
    ADD    r0, r0, r1 ;r0과 r1의 저장된 값을 더하여 r0에 저장

    END ;파일 끝
```

B. Flow chart 작성



C. Result

Register	Value
Current	
R0	0x0000000D
R1	0x00000003
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000

처음에 MOV 명령으로 r0에 10 (0xA)이 저장되고, r1에 3 (0x3)이 저장된다. 그 다음 ADD 명령을 통해 r0와 r1에 저장된 10과 3이 더해져서 그 결과인 13 (0xD)이 r0에 저장된 걸 확인할 수 있다. 또한 PC(program Counter)는 다음 실행될 명령어의 주

소를 가지는 역할이므로, 명령어가 실행될 때마다 4byte씩 증가되어 MOV, MOV, ADD
를 모두 실행한 다음에는 0x0000000C로 업데이트된 걸 볼 수 있다.

3. 고찰 및 결론

A. 고찰 (35%)

주어진 코드를 첫번째로 실행했을 때는 오류가 발생했었다. 코드를 다시 살펴본 결과
첫번째 줄을 작성할 때 공백을 넣지 않았다는 걸 발견하여, 수정 후 돌려 보니 정상적으
로 작동하였다. label을 제외한 코드들은 앞에 공백을 넣어줘야 하기 때문에 오류가 발생
했던 것 같다.

B. 결론 (10%)

명령어를 실행하면서 라인 별로 PC 레지스터의 값이 변하는 걸 확인하면서, 각 명령어
의 크기는 4byte라는 것을 다시 한번 깨달을 수 있었다. 또한 최종적으로 결과가 나왔을
때, 덧셈을 해도 여전히 r1의 값은 3인걸로 보아 완전한 이동이 아니라 복사하여 이동한
다는 점도 알 수 있었다.

4. 참고문헌 (2%)

이준환 / 어셈블리프로그래밍설계및실습 강의자료 / 광운대학교(컴퓨터공학과) / 2023