

어셈블리프로그래밍설계및실습 보고서

과제 주차: 3주차

학 과: 컴퓨터공학과

담당교수: 이형근 교수님

실습분반: 화요일 6, 7

학 번: 2021202058

성 명: 송채영

제 출 일: 2022.09.22(목)

1. Problem Statement

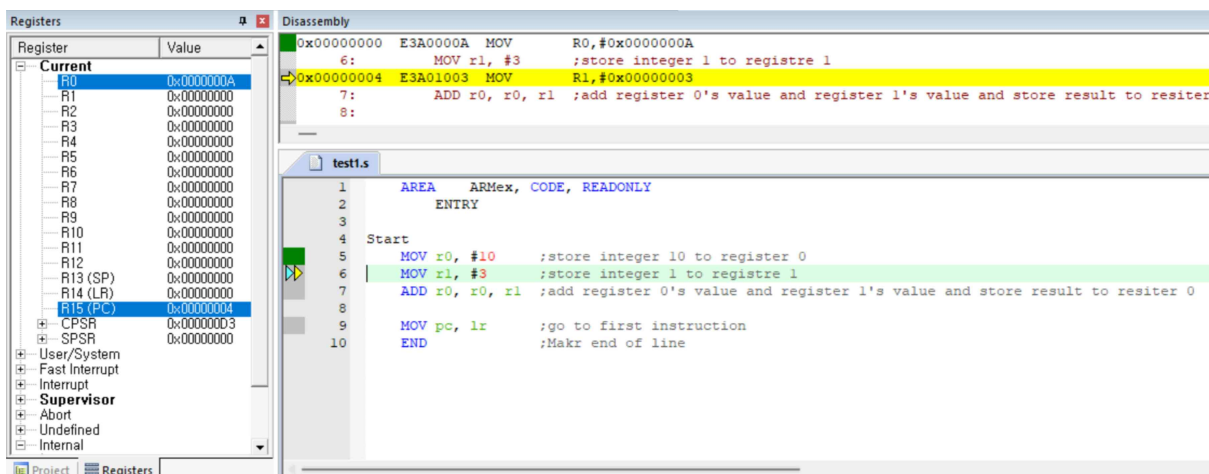
어셈블리의 기본에 대해 알고, 프로그램을 구성하는 방법에 대해 안다. 어셈블리에서 사용하는 ARM과 ARM에 사용법(파일 생성, 디버깅)에 대해 안다. 예제 코드를 수행해 기본적인 동작에 대해 이해하고 register 값을 확인하는 법을 안다.

2. Design

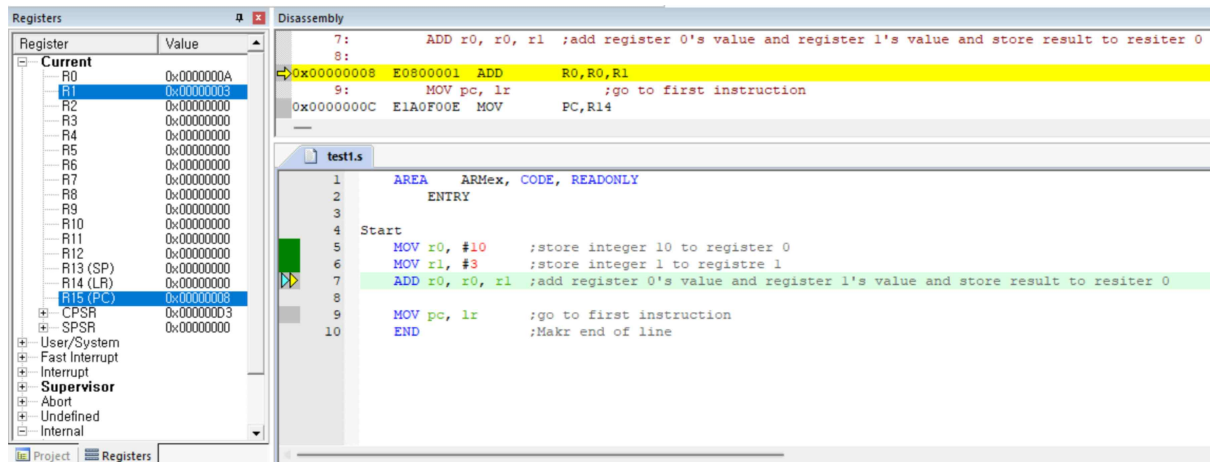
어셈블리어는 기계어를 문자로 기호화 해 나타내는 낮은 레벨의 언어이다. 어셈블리의 기본 명령어는 Register movement, Arithmetic, Logical, Comparison 이렇게 4가지로 구성 되어있다. Register movement에는 MOV와 MVN이 있다. Arithmetic에는 ADC, ADD, RSB, RSC, SBC, SUB 등의 명령어가 있다. Logical은 AND, ORR, EOR, BIC가 있는데, 비트 연산에 관련된 명령어이다. Comparison은 비교와 관련된 명령어 CMN, CMP, TEQ, TST가 있다.

register r0에 정수 10을 저장한다. -> register r1에 정수 3을 저장한다. -> register r0에 저장된 값과 register r1에 저장된 값을 더해 그 값을 register r0에 저장한다. -> lr의 값을 pc에 저장한다 -> 프로그램을 종료한다. 의 순서대로 과제를 수행할 것이다.

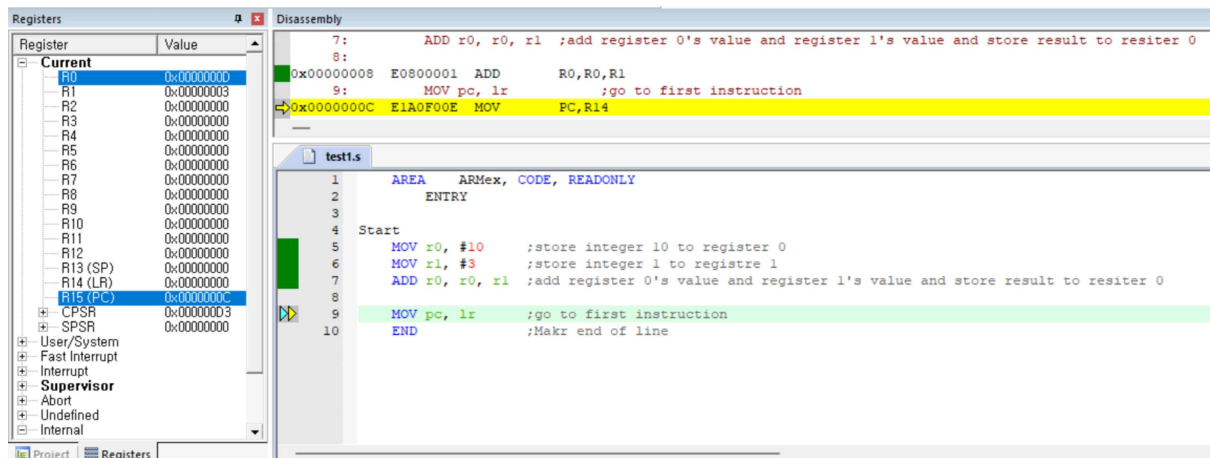
3. Conclusion



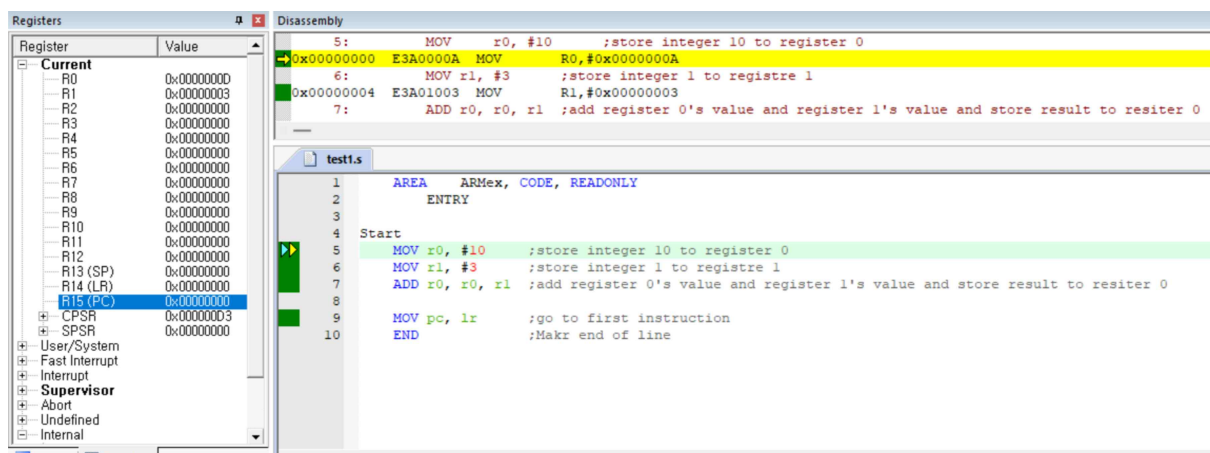
MOV r0, #10을 통해 register r0에 정수 10을 저장한다. R0의 value가 A로 바뀐 것을 확인할 수 있다.



MOV r1, #3d을 통해 register r1에 정수 3을 저장한다. R1의 value가 3으로 바뀐 것을 확인할 수 있다.



ADD r0, r0, r1을 통해 register r0에 저장된 값과 register r1에 저장된 값을 더해 그 값을 register r0에 저장한다. R0의 value가 D로 바뀐 것을 확인할 수 있다.



MOV pc, lr을 통해 r의 값을 pc에 저장한다.

4. Consideration

low 레벨의 언어를 처음 써봐 과제를 수행하면서 첫 줄 부터 막막했다. 수업 때 배웠던 명령어를 참고하여 MOV 명령어를 통해 레지스터에 수동적으로 값을 저장하고, ADD 명령어를 통해 값을 더해주고 레지스터에 더해진 값을 저장하고, END를 통해 코드의 끝을 나타냈다. 실습 시간에 ppt 자료를 통해서 배웠던 것들 (예를 들어 code size를 확인하는 등)도 확인하며 keil에 익숙해지려고 노력한 것 같다. 또한 MOV pc, lr을 통해 lr의 값을 pc의 레지스터에 복사해 함수가 호출된 시점으로 돌아가게 한다는 점이 처음엔 헷갈렸지만 직접 코드를 디버깅 해보며 이해를 한 것 같다.

5. Reference

이형근 교수님/어셈블리프로그래밍설계및실습/광운대학교(컴퓨터정보공학부)/2022

어셈블리어

[/https://terms.naver.com/entry.naver?docId=2180467&cid=43667&categoryId=43667](https://terms.naver.com/entry.naver?docId=2180467&cid=43667&categoryId=43667)