어셈블리프로그램 설계및실습 보고서

과제 주차: 3주차

학 과: 컴퓨터정보공학부

담당교수: 이형근 교수님

실습분반: 화요일 6, 7

학 번: 2019202050

성 명: 이강현

제 출 일: 2022.09.20(화)

1. Problem Statement

   어셈블리언어에 대해 기본예제를 통해 학습하고 프로그램을 어떻게 구성하고 있는지 알아본다. ARM에서 코드를 작성해보고 빌드, 디버깅을 통해 register의 값을 컴퓨터가 어떻게 처리하는지 기본적인 이해를 기른다.

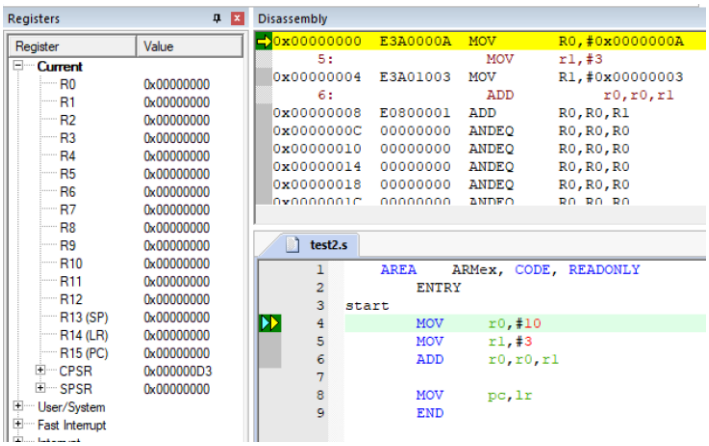
2. Design

   MOV 명령어는 레지스터에 값을 저장하는 것이고 ADD 명령어는 두 값을 더해서 레지스터에 저장한다. END명령어로 코드를 종료한다. Basic example 예제는 명령어를 통해 값이 어떻게 저장되는지 눈으로 확인하고 학습한다.

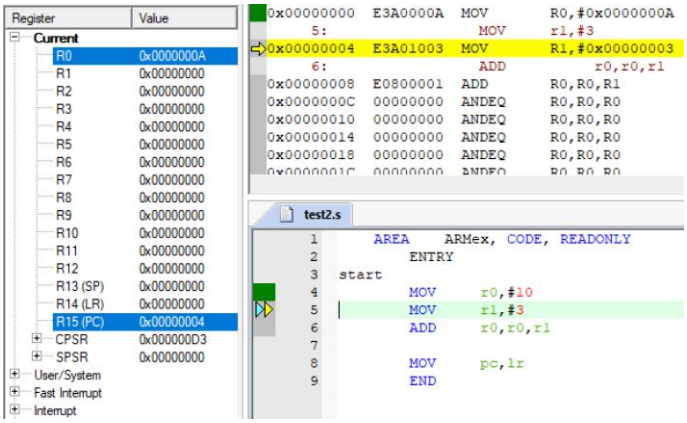
   코드의 내용을 간략하게 설명하면

   1. Register0에 정수 10을 저장.
   2. Register1에 정수 3을 저장
   3. Register0에 저장된 값과 register 1에 저장된 값을 더해 register0에 저장한다.
   4. Pc에 lr값을 저장한다.

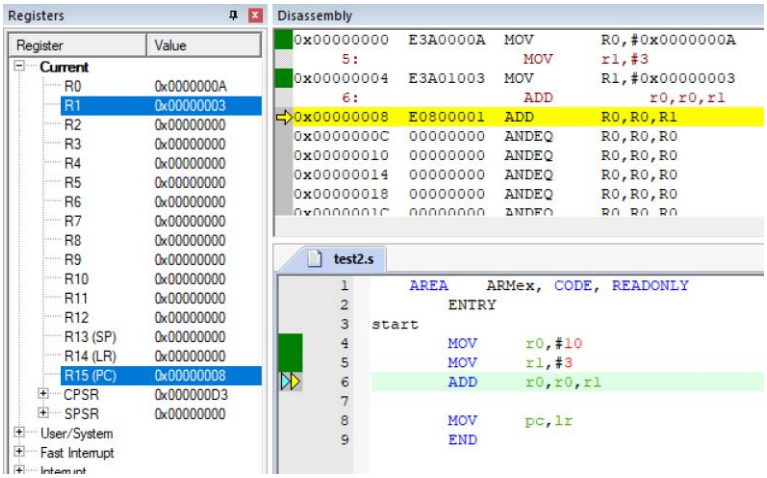
3. Conclusion



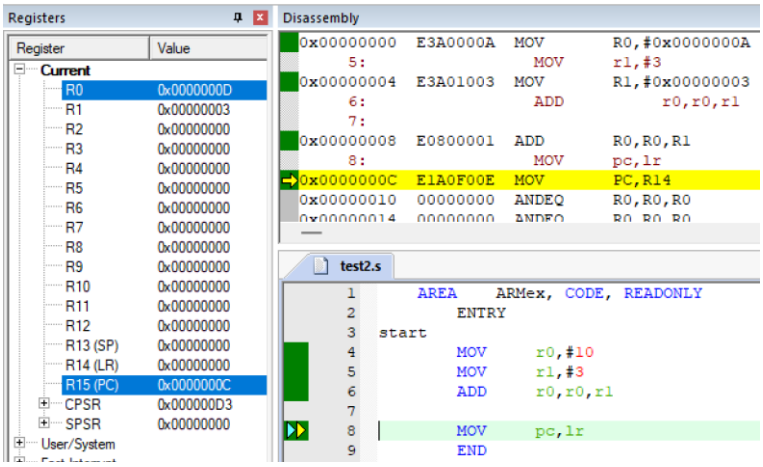
Register 0값이 0x00000000으로 되어있는 초기 모습



Register 0에 MOV명령어를 통해 10이 저장된 모습 16진수이므로 A로 표현되어 있다.

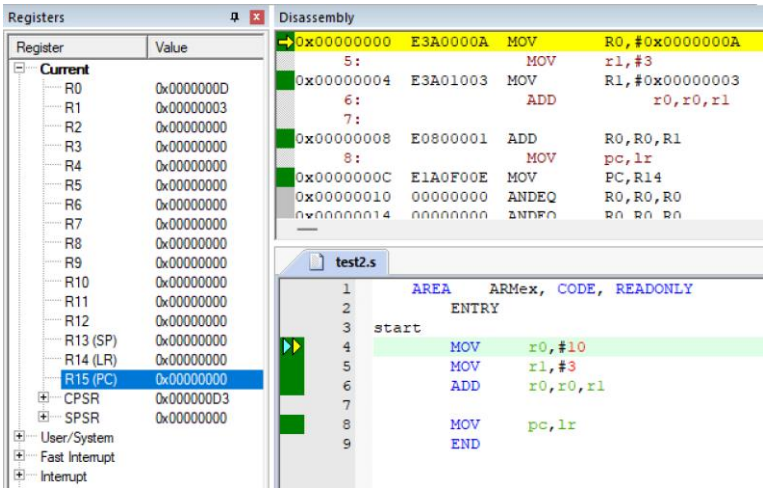


Register1에 정수 3이 저장



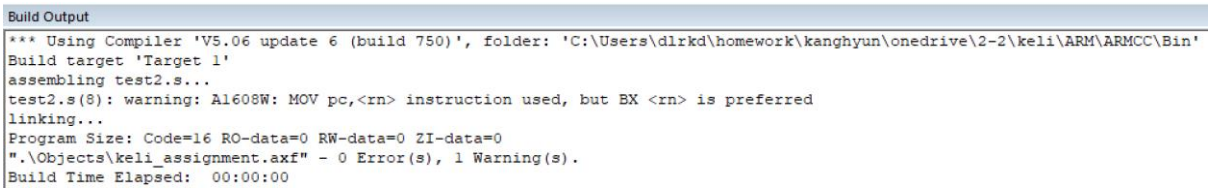
Register0에 register 0과 register 1의 값이 더해져 저장된 모습

16진수 이므로 A(=10)+3=D(=13)가 저장됨.



Pc에 값이 0x00000000인 lr값이 들어가면서 0x00000000주소를 가지는 첫번째 명령어 MOV가 다시 실행되는 모습

이로써 계속 코드가 반복될 것을 유추할 수 있다



또한 코드크기가 16으로 명령어 4개가 코드 크기란 것을 알게 되었다.

4. Consideration

코드를 구현하면서 제일 먼저 start라는 코드가 AREA와 같은 라인에 들여쓰기 되어있었는데 오류가 나는 것을 확인했다 이는 start 코드를 가장 앞에 두는것으로 해결하였다. 또한 위와 같은 예제로 pc의 주소를 건들면 명령어가 순차적으로 실행되지 않고 특정 구간으로 점프하는 것을 확인했는데 이를 memory.ini파일로 메모리 접근 구간을 정해두는 것으로 해결할 수 있었다.

5. Reference

이형근/어셈블리프로그램 설계 및 실습/광운대학교(컴퓨터정보공학부)/2022