어셈블리프로그램 설계 및 실습 보고서

실험제목: Data transfer from/to memories

실험일자: 2023년 10월 5일 (목)

제출일자: 2023년 10월 11일 (수)

학 과: 컴퓨터정보공학부

담당교수: 이형근 교수님

실습 분반: 화 5, 목 6,7

학 번: 2020202031

성 명: 김재현

1. 제목 및 목적 (3%)
   1. 제목

Data transfer from/to memories

* 1. 목적

기본 명령어 사용 예제를 통해 어셈블리어 프로그래밍을 이해한다.

ARM 조건부 실행 코드 보는 방법을 이해, 이를 어셈블리어 프로그래밍 능력을 습득한다.

원하는 데이터를 메모리로 저장 및 가져올 수 있도록 어셈블리어 프로그래밍 능력을 습득한다.

1. 설계 (Design) (50%)

problem 1)

* 1. Pseudo code

r0 = 00001000

r1 = 0x11

r2 = 0xA

store r1 data to address 00001000

load one byte data from address 00001000 to r3

compare r3, r2

if (r3 > r2) r5 = 1

if (r3 < r2) r5 = 2

if (r3 == r2) r5 = 3

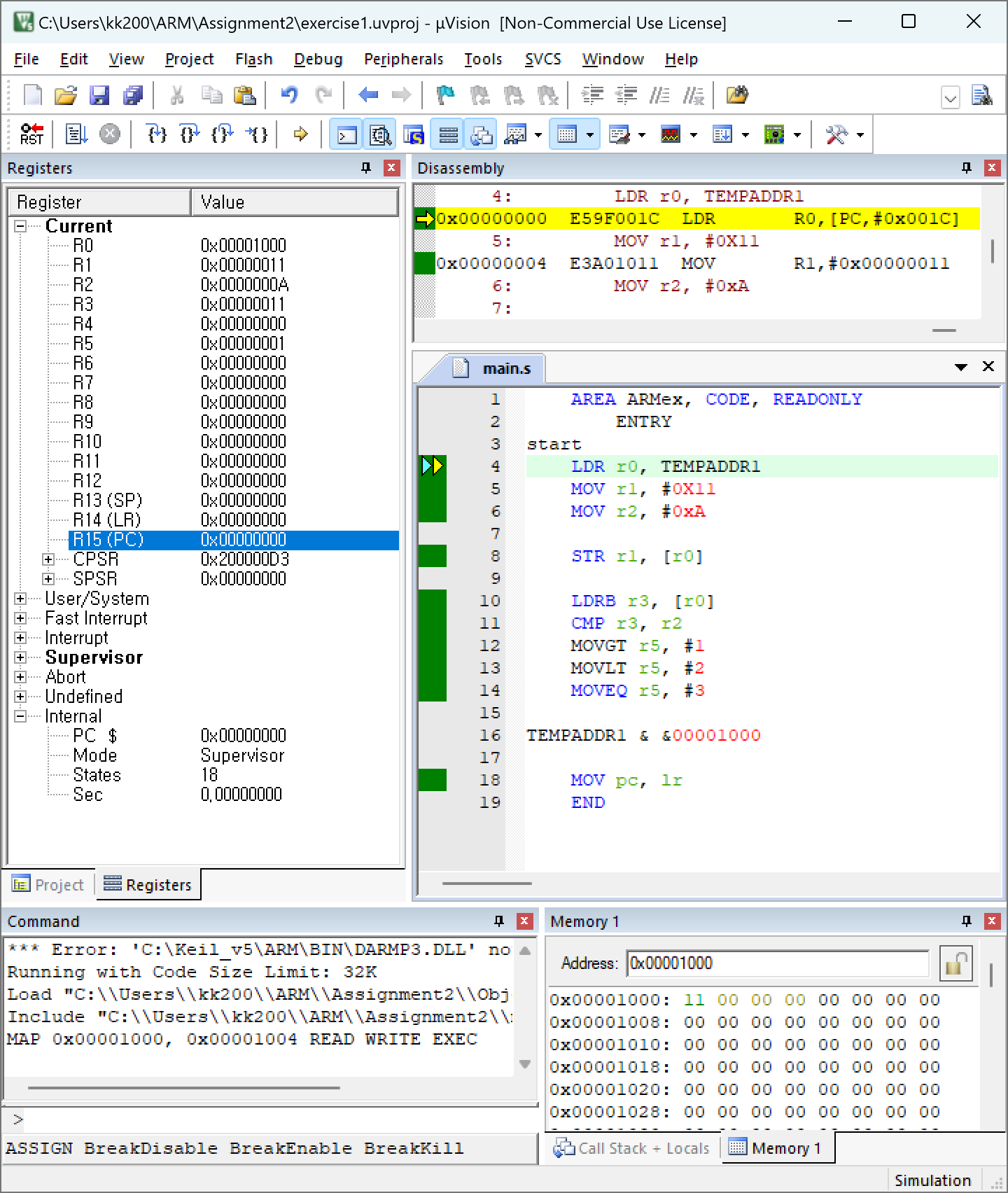
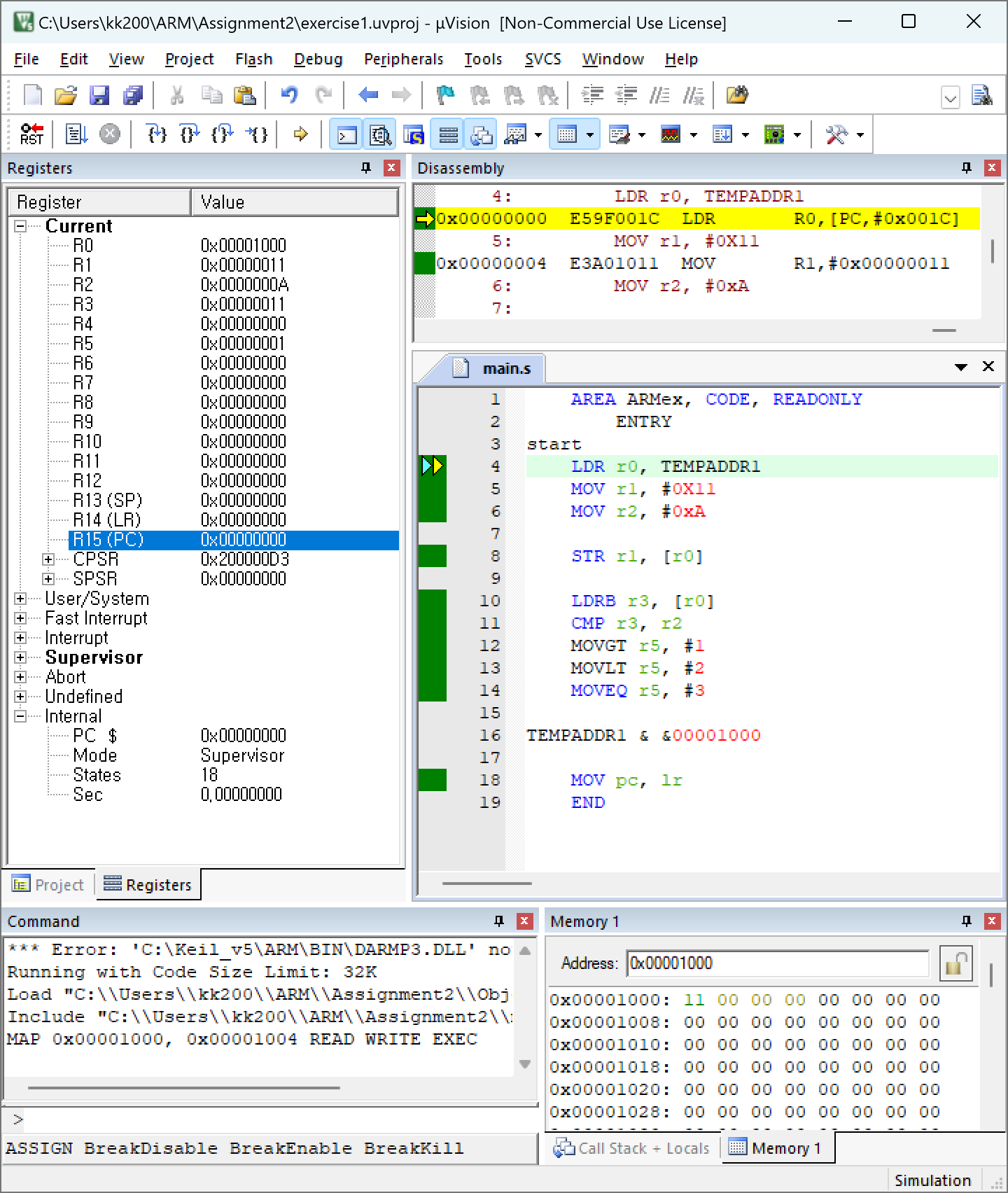
return to start

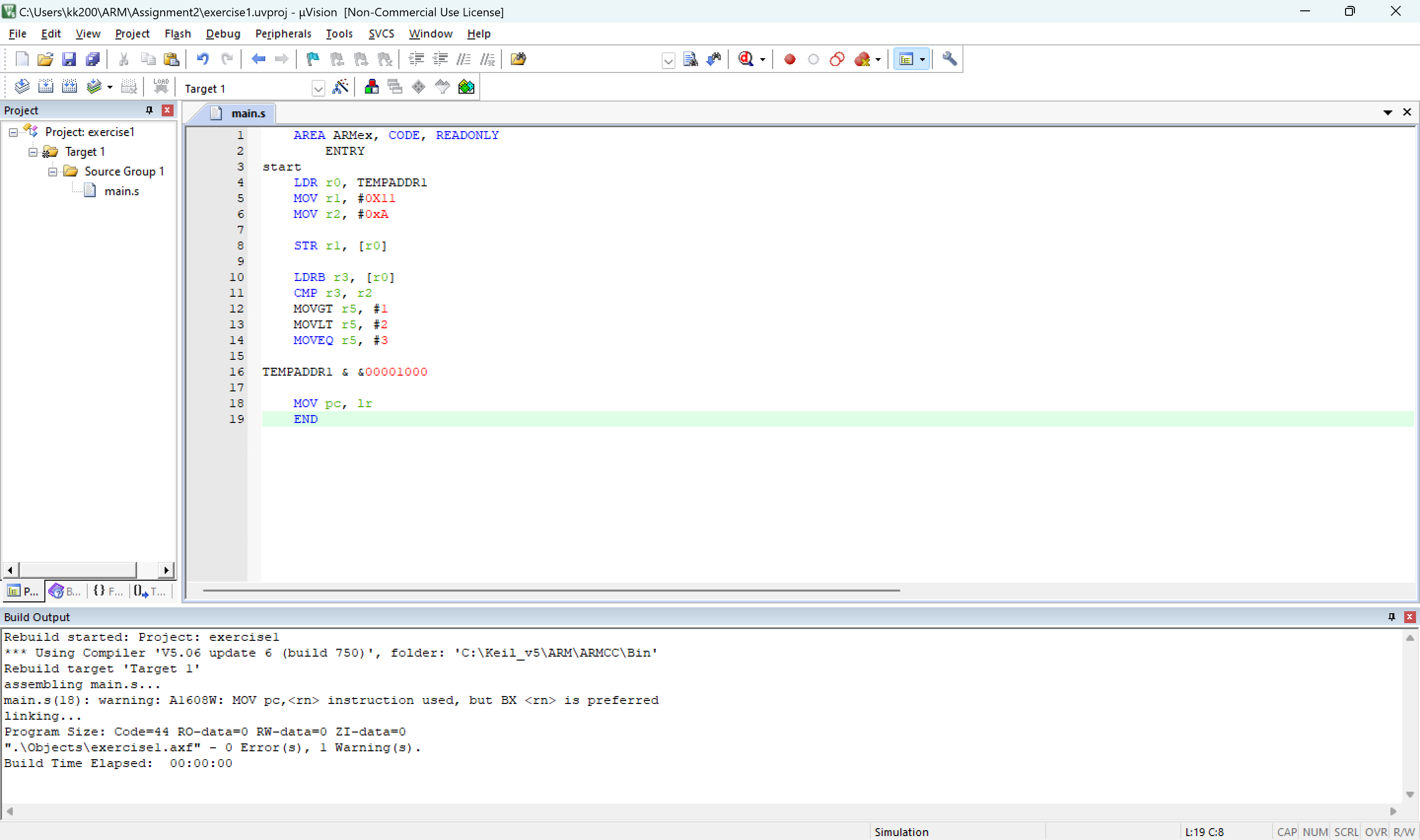
* 1. Flow chart 작성

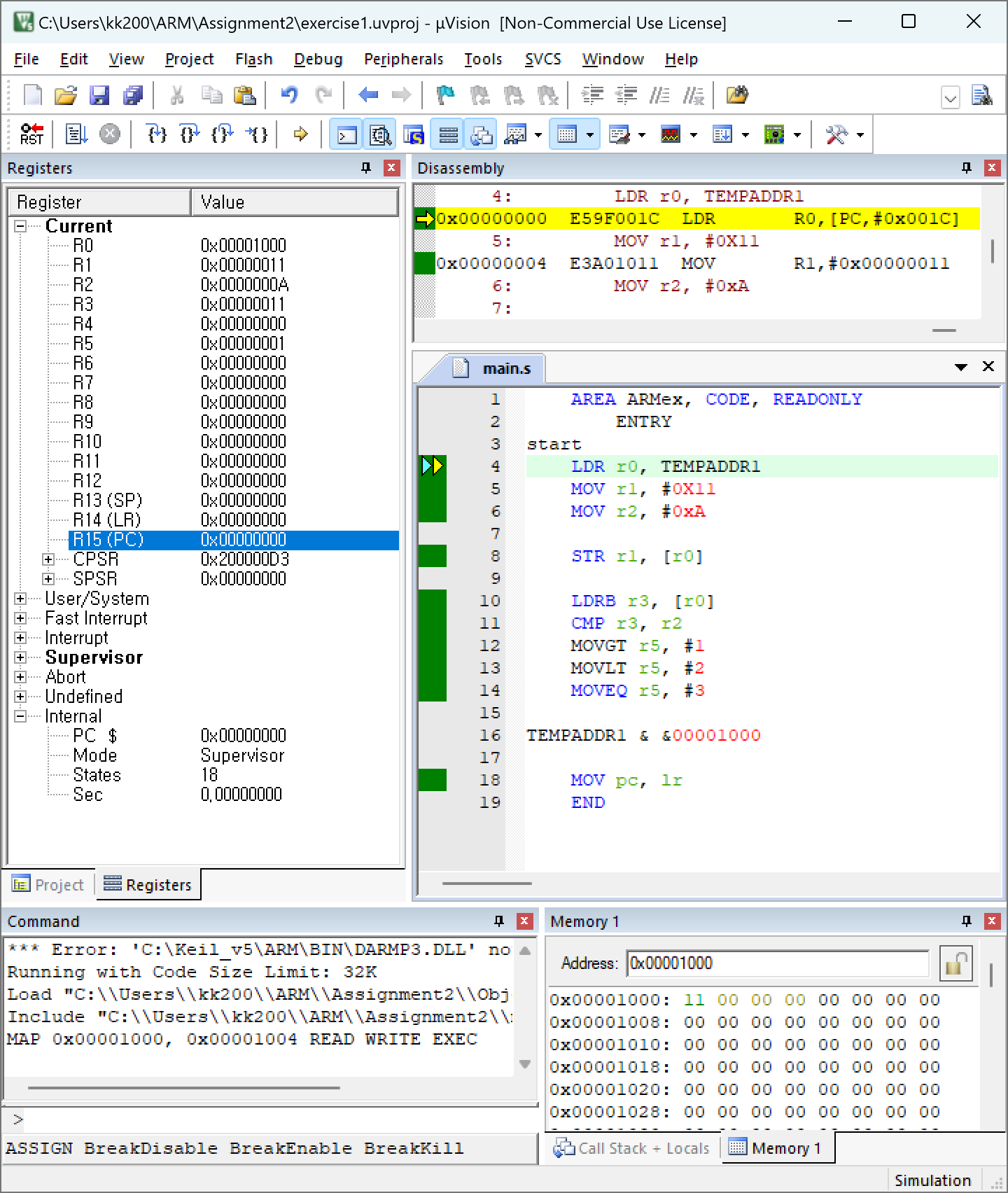
텍스트, 도표, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

* 1. Result



* 1. Performance

code size: 44Byte

states: 18

problem 2)

1. Pseudo code

r0 = 1

r1 = 2

r3 = 3

r4 = 4

r2 = 00001000

store r0 data to address 00001000

store r1 data to address 00001001

store r3 data to address 00001002

store r4 data to address 00001003

load four bytes data from address 00001000 to r5

store r4 data to address 00001000

store r3 data to address 00001001

store r1 data to address 00001002

store r0 data to address 00001003

load four bytes data from address 00001000 to r6

return to start

1. Flow chart 작성

텍스트, 영수증, 흑백이(가) 표시된 사진

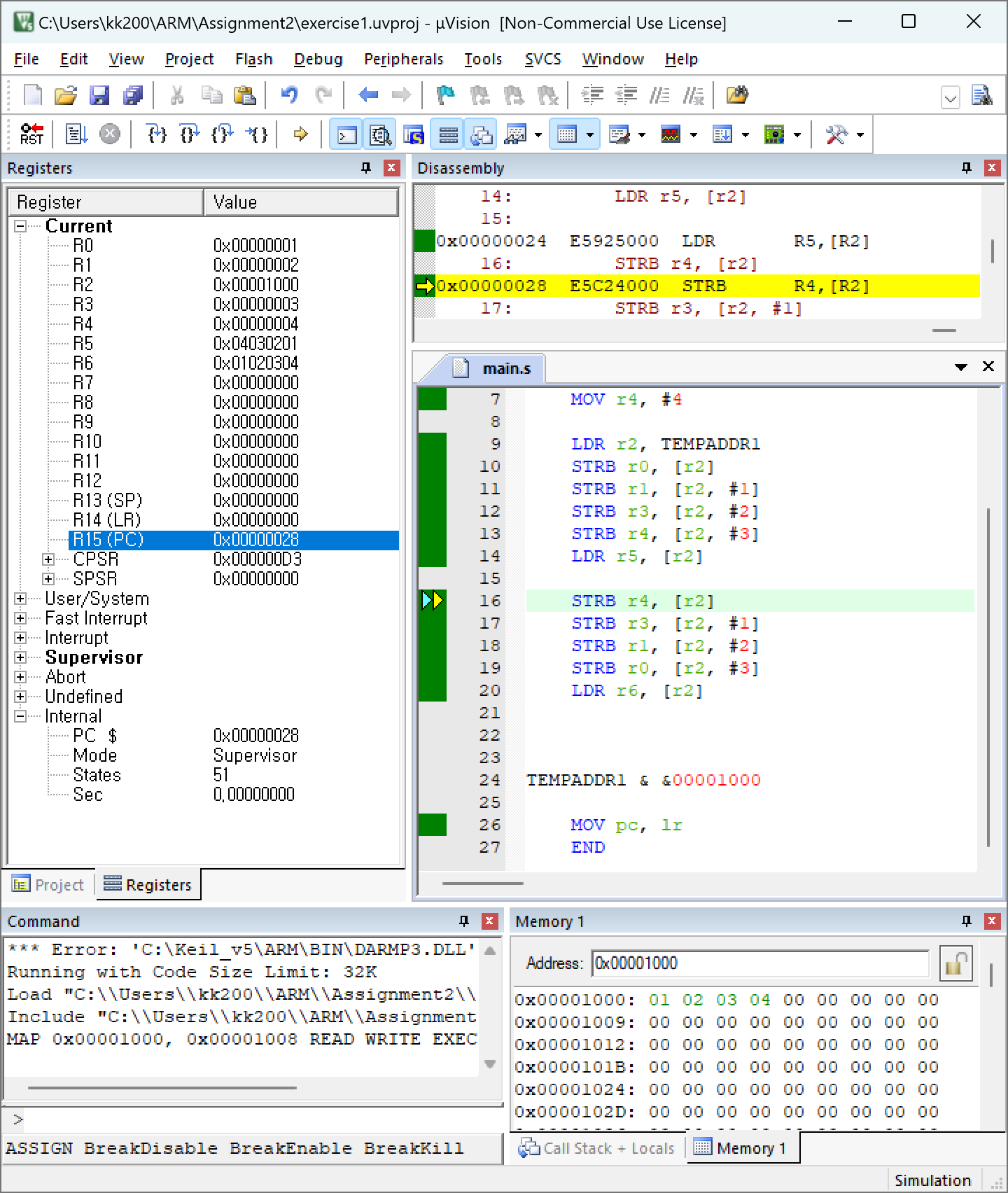
자동 생성된 설명

1. Result

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

r0, r1, r3, r4에는 문제에 주어진 값들을 저장하고, r2에 메모리 주소 값을 저장했습니다.



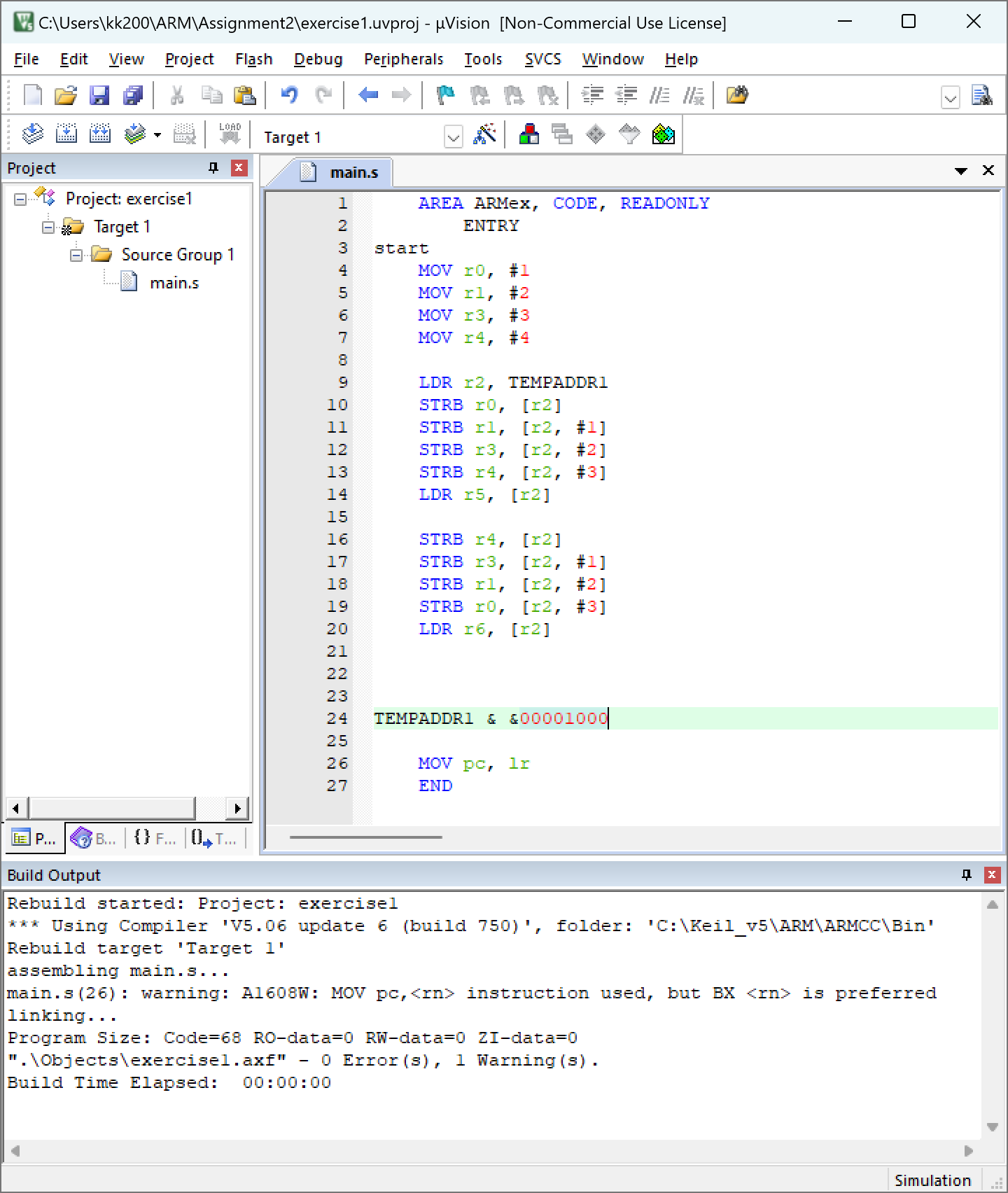
STRB를 사용하여 레지스터에 저장된 값들을 메모리 한 바이트 단위로 저장하고, LDR을 통해 네 바이트를 한번에 불러들여 r5에 0x04030201을 저장했습니다.

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

마찬가지 방법이지만 레지스터 값들을 반대 순서로 메모리에 저장해줌으로써 r6에 0x01020304를 저장했습니다.

1. Performance



code size: 68Byte

`텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

states: 33

1. 고찰 및 결론
   1. 고찰 (35%)

레지스터에 값을 대입할 때 MOV 명령어를 사용했었는데, 메모리 주소 값을 대입할 때는 LDR 명령어를 쓰는 이유에 대해 궁금증이 생겼습니다. 이에 레지스터에 MOV 명령어를 사용하여 메모리 주소 값을 대입하려고 시도했는데 오류가 났습니다. 이에 레지스터에 저장할 값의 자리수를 하나씩 줄였더니 16진수로 최대 2자리수까지만 가능했습니다. 이를 통해 MOV 명령어를 통해서는 1바이트의 수만을 대입 가능하며 그 이상의 크기를 가지는 수에 대해서는 LDR 명령어를 사용해야 함을 알게 됐습니다.

* 1. 결론 (10%)

이번 실습을 통해 LDR, STR 명령어를 통해 메모리에 접근하여 값을 불러오고, 저장하는 법을 알게 됐고, 명령어 뒤에 condition B, H를 붙여 접근할 메모리의 크기를 설정하는 법 또한 익힐 수 있어 유익한 실습이었던 것 같습니다.

1. 참고문헌 (2%)

이형근 교수님/어셈블리프로그램설계및실습/광운대학교(컴퓨터정보공학부)/2023