



연습문제

exercise

<3장 연습 문제 참고만 바랍니다>

1. 컴파일러와 어셈블리의 차이점을 설명하라.

컴파일러는 고급언어로 작성된 원시 프로그램을 입력으로 받아 목적프로그램을 만드는 프로그램이다. 어셈블러는 어셈블리어를 기계어 형태의 오브젝트 코드로 변환해주는 프로그램이다.

2. 소스 코드와 목적 코드의 차이점을 설명하라.

소스코드는 프로그램을 작성하는 프로그래머에 의해 작성된 프로그램 문장들로 구성되며, 컴파일되기 이전의 프로그램이다.

목적코드는 프로세서가 이해할 수 있는 명령어의 형태를 가지며 컴파일된 파일이다. 따라서 목적코드는 프로그래머가 목적코드를 읽거나 수정하기 어렵다.

3. 의사명령어와 어셈블리 명령어의 차이점은 무엇인가?

어셈블리 의사명령어(지시어)는 컴퓨터가 무엇을 하도록 지시하는 것이 아니고, 어셈블러에 대한 지시 사항을 전달하며, 기계어로 번역되지 않는다.

어셈블리 명령어는 컴퓨터로 하여금 무엇을 수행하라고 직접 명령하는 것이고, 기계어로 번역된다.

4. 프로그램 카운터와 명령어 로케이션 카운터는 어떻게 다른가?

프로그램 카운터는 프로세서가 실행할 다음 명령어의 주소를 저장하고 있는 레지스터이다.

명령어 로케이션 카운터는 어셈블러 내부 변수로 명령어가 할당된 메모리 위치를 나타내고 있으며, 각 명령어 오프셋 크기에 의해 증가된다.

5. 레이블과 주소의 차이점을 설명하라.

레이블은 주소를 나타내는 심볼이며 레이블로 지정된 주소는 어셈블리 동안 계산된다.

주소는 메모리의 물리 위치를 나타내는 것으로 데이터가 저장되어 있는 위치를 나타낸다.

6. 니드닉과 16진수 표현의 차이를 설명하라.

니드닉은 어셈블리 언어를 기계어 명령을 알기 쉬운 기호로 표시한 기호를 니드닉(Mnemonic)이라 한다. 어셈블리 언어의 니드닉으로 ADD A,B와 같이 나타낸다. 어셈블리 언어를 구성하는 니드닉은 라벨, 명령어코드,오프셋, 주석으로 구성되어 있다. 16진수 표현은 16을 기수로하는 표현하는 수체계이다.

7. 프로그래머가 어셈블리어로 프로그램을 작성할 때 반복되는 연산을 효과적으로 하기 위한 방법은 어떠한 것들이 있는가?

동일한 부분을 반복적으로 프로그램을 작성해야만 하는 불편함을 덜어주기 위해 매크로 정의 기능과 매크로 호출 기능을 사용한다.
프로그램 작성에서 프로그램이 간단해지고 반복적인 코딩의 건조로움을 피하고 효과적으로 프로그램을 할 수 있다.

8. 리틀 엔디언과 빅 엔디언의 차이점을 설명하라.

Endian은 메모리에 여러 바이트의 내용을 저장할 때 상위 바이트 및 하위 바이트의 위치를 결정하는 방식으로 리틀엔디언은 메모리 주소가 낮은 주소에서 하위 바이트부터 저장된다는 것이고 빅엔디언은 반대로 메모리에서 낮은 주소에서 상위 바이트부터 저장되는 것을 말한다.

9. 간단한 컴퓨터에서 두 개의 숫자를 서로 교환하는 프로그램을 작성하라.

<생략>

10. 간단한 컴퓨터에서 논리적 OR 연산을 구현하라.

<생략>

11. 간단한 컴퓨터에서 반전(negation) 연산을 구현하라.

<생략>

12. 간단한 컴퓨터에서 $Z \leftarrow X \times Y$ 의 곱셈 연산을 수행하는 어셈블리어 프로그램을 작성하라. 여기서 X, Y, Z는 메모리의 위치를 나타낸다.

<생략>

13. MDR과 MAR을 이용해서 메모리에 대한 읽기 동작을 수행하는 절차를 설명하라.

(1) 메모리의 임을 데이터워드 주소를 MAR(Memory Address Register)에 적재한다.

(2) 프로세서에서 'Read'라는 신호를 메모리로 전달하며 MAR에 위치한 데이터, 즉 워드를 읽어 나와 MDR(Memory Data Register)로 적재한다.

(3) 프로세서는 MDR의 저장된 데이터를 사용하게 된다.

14. MDR과 MAR을 이용해서 메모리에 대한 쓰기 동작을 수행하는 절차를 설명하라.

(1) 프로세서는 메모리에 저장한 데이터를 먼저 특수한 레지스터인 MDR에 적재한다.

(2) 프로세서는 데이터를 저장할 메모리 위치, 즉 주소를 MAR에 적재한다.

(3) 프로세서의 'Write' 제어신호는 MDR에 있는 데이터를 메모리 주소 MAR 위치에 저장한다.

15. ARM 프로세서의 ISA에 대해서 설명하라.

<생략>

16. ARM에는 범용 레지스터가 몇 개 있는가?

ARM7인 경우 31개의 범용 레지스터가 있으나 레지스터 키워드는 R0 ~ R15까지 16개로 유저모드에서 사용자가 한번에 사용할 수 있는 레지스터는 16개이다. 이 중에 프로그램 카운터와 스택 포인터도 포함되어 있다.

17. ARM에서 CPSR이 하는 역할은 무엇인가?

Current Program Status Register(CPSR)는 현재 동작중이 프로세서 상태를 나타내며 모든 동작 모드에서 공유하여 사용할 수 있다.
ARM에서 Z 비트의 역할은 무엇인가?

Current Program Status Register(CPSR)에서 Z 비트는 ALU 연산 결과 Zero인지를 나타내는 플래그 비트이다.

19. ARM에서 프로그램 카운터는 어디에 있는가?

프로그램 카운터는 R15(PC)이며, 비트 2~31에 저장된다.

20. 다음 연산이 이뤄지고 나면 ARM의 상태 워드는 어떻게 설정되었는가?

- 1) $2 - 3 ==>$ 음수 $==>$ NZCV = 1000
- 2) $-2^{32} + 1 - 1 ==>$ 오버플로우 $==>$ NZCV = 0001
- 3) $-4 + 5 ==>$ Borrow가 발생 $==>$ NZCV = 0010

ARM7에서 상태레지스터에서 플래그 비트는 4개로 다음과 같다. N: Negative/Less Than Flag, Z : Zero Flag, C: Carry/Borrow/Extend Flag, V:Overflow Flag

21. 다음 C 언어의 할당문을 ARM 어셈블리어로 구현하라.

- 1) $x = a + b;$
- 2) $y = (c - d) + (e - f);$
- 3) $z = a \times (b + c) - d \times e;$

<생략>

22. 다음 ARM 조건 코드의 의미는 무엇인가?

- 1) EQ : 영과 일치 Z=1
- 2) NE : 영과 불일치 Z=0
- 3) MI : 음수 N=1
- 4) VS : 오버플로우 V=1
- 5) GE : 유부호 음 Z=0, N==V
- 6) LT : 유부호 같거나 같음 Z==1 or N!=V

23. 다음 명령어에서 R1에 저장되는 값은 무엇인가?

```
MOV R1, #300
R1 = 300
```

24. 다음 C 언어의 조건문을 ARM 어셈블리어로 구현하라.

```
if (x - y < 3) {
    a = b - c;
    x = 0;
}
else {
    y = 0;
    d = e + f + g;
}
```

<생략>

25. ARM에서 BL 명령어의 동작을 설명하라.

Branch 명령은 PC 상대적으로 분기하며 분기할 주소는 분기명령어 주소 + 8+부호-확장이다.
Branch and Link 명령으로 Branch의 동작과 동일하며 분기 전에 LR에 BL 다음 명령어의 주소를 기록한다.
할수 호출에 사용하며 복귀할 경우 MOV PC, LR과 같은 명령어를 사용한다.