

Report – HW 1

1. Problem & Purpose

i. Purpose

예제 코드를 수행하여 기본적인 동작에 대해 이해하고, register 값을 확인하는 법을 숙지하기 위해 해당 과제를 진행한다.

ii. Problem

Problem 1.

메모리에 저장된 숫자 3개를 1 바이트 단위로 읽어서 해당 가져온 데이터의 값에 따라 R5의 값을 바꾸는 프로그램을 작성하라.

➤ 0x0A보다 클 경우 1을, 0x0A보다 작을 경우 2를, 0x0A와 같을 경우 3을 R5에 저장한다. *단, 반드시 조건부 실행 명령어를 사용해야 한다.

✧ 읽어들 메모리의 번지는 구현자가 임의로 설정

✧ 또한 최초에 메모리에 저장되는 값은 구현자가 임의로 입력 후 시작한다.

✧ ex) 메모리 번지 0x00001000에 0x11를 저장하고 시작

Problem 2.

R0, R1, R2, R3 레지스터에 다음과 같은 값들이 저장돼 있다고 했을 때, 현재 little-endian 방식임을 감안하여 R5와 R6에 다음을 저장하라.

➤ R0=1, R1=2, R2=3, R3=4

➤ R5 = 0x04030201

➤ R6 = 0x01020304

✧ (힌트) 메모리를 사용하시오

2. Used Instruction

MOV, MOVL, MOVGT, MOVEQ // END // CMP // LDR // STRB

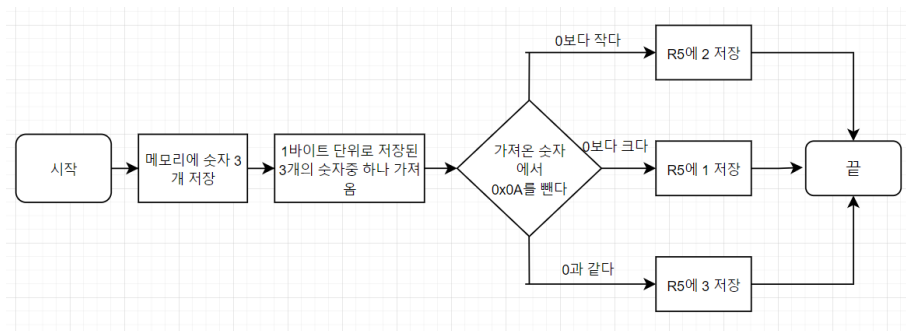
i. MOV Rd operand2 : operand2에 있는 값을 Rd에 저장한다.

- A. MOVLT Rd operand 2 : CPSR에 있는 N, V flag가 서로 다를 때 실행한다.
Comparison 명령어가 비교한 결과가 부호가 있고 0보다 작을 때 실행한다.
- B. MOVGT Rd operand 2 : CPSR에 있는 Z flag 가 0이고, N, V flag가 서로 같을 때 실행한다. Comparison 명령어가 비교한 결과가 부호가 있고 0보다 클 때 실행한다
- C. MOVEQ Rd operand 2 : CPSR에 있는 Z flag 가 1일 때 실행한다. Comparison 명령어가 비교한 결과가 0일 때 실행한다.
- ii. END : Assembly code가 끝났음을 의미하는 Instruction
- iii. CMP Rn operand2 : Rn와 operand2에 있는 값을 빼서 그 결과에 따라 CPSR에 있는 N, Z, C, V flag를 설정한다.
- iv. LDR Rd addressing : addressing에 있는 값을 Rd에 저장한다.
- v. STRB Rd <address> : 1Byte에 대해 Rd에 있는 값을 address에 저장한다.

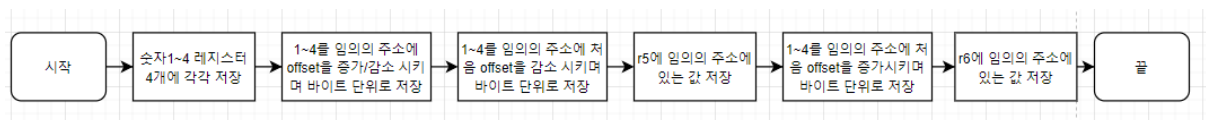
3. Design(Flow chart)

- i. 설계한 내용의 Flow chart

A. Problem 1



B. Problem 2



4. Conclusion

처음 어셈블리어를 이용해봤다. 과제를 보고 엄청 막막했는데 교안과 책이나 인터넷 정보를 이용하여 해결했다. Byte단위로 load & store하는 것이 이론 수업만 들었을 땐 잘 와닿지 않았었는데 실습을 통하여 어느정도 습득했다. Store를 하며 직접 메모리에 있는 값을 볼 수 있었다. Little Endian이나 Big Endian도 실습을 해보니 이해가 더 잘 되었다. 또한 pre indexing, post indexing을 을 배우며 C나 Java같은 고급언어에서 반복문이 이렇게 돌아가는 거구나 하고 신기했다. CMP같은 comparison 명령어를 통한 CPSR의 flag들을 잘 이해하지 못하여 애를 먹었다. 교안과 외부 자료를 학습하며 완전히는 아니지만 어느정도 알게 되어서 기쁘다. 계속 이론수업을 듣고 실습을 하다 보면 많이 알 수 있을 것 같다. 컴퓨터의 작동방식을 좀 더 알게 되었다.

5. Reference

플로우 차트에 대한 이해와 활용: <https://www.tina.com/ko/flowchart-editor-and-debugger/>

STRB, LRD 명령어에 대한 이해: <https://ch4njun.tistory.com/>

전반적인 명령어 이해 : 강의교안,

Pre Indexing, Post Indexing : <https://kyuhyuk.kr/article/simple-arm-operating-system/2019/03/04/Simple-ARM-Operating-System-Chapter-2>

V flag와 C flag : https://www.youtube.com/watch?v=9cXe_T9nL4

<https://blog.naver.com/heobk1/221397307858>

리틀 엔디안 & 빅 엔디안 : https://tcpschool.com/c/c_refer_endian\