

Report – HW 3

1. Problem & Purpose

i. Purpose

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

ii. Problem

Problem 1.

■ 1에서 10까지의 Factorial값을 Second operation로 구현하시오

- Factorial값이란 1부터 해당하는 값까지의 곱을 말함
- Ex) $10! = 10 \times 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 3,628,800$
- Factorial값을 지정된 memory 주소(0x40000)에 순서대로 저장

Problem 2.

- Problem 1에서 작성한 코드를 Multiplication operation으로 수정
- Problem 1과 Problem 2의 차이와 성능에 대해서 비교

2. Used Instruction

MOV // END // LDR // STR // LSL // ADD // MUL // RSB

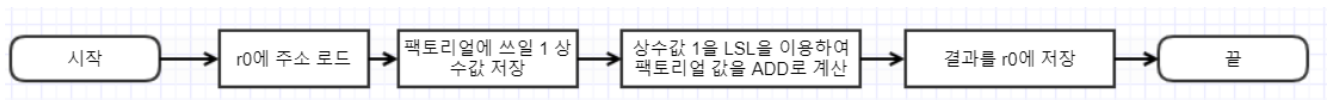
- i. MOV Rd, operand2 : operand2에 있는 값을 Rd에 저장한다.
- ii. END : Assembly code가 끝났음을 의미하는 Instruction
- iii. LDR Rd, addressing : addressing에 있는 값을 Rd에 저장한다.
- iv. STR Rd <address> : Rd에 있는 값을 address에 저장한다.
- v. LSL #number : Number 만큼 왼쪽으로 bit stream을 shift
- vi. ADD Rd, Rn, N : Rd에 $Rn+N$ 값을 저장한다

- vii. MUL Rd, Rm, Rs : Rm에 저장된 값과 Rs값을 곱하여 Rd에 저장한다
- viii. RSB Rd, Rn, N : Rd에 N-Rn을 저장한다

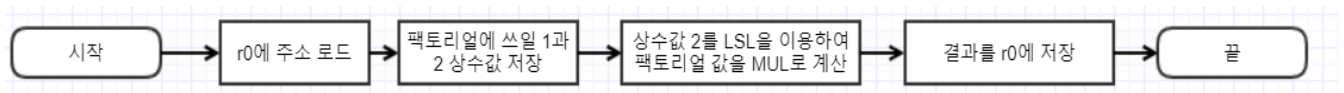
3. Design(Flow chart)

- i. 설계한 내용의 Flow chart

A. Problem 1



B. Problem 2



4. Conclusion

[Problem 1과 Problem 2의 차이와 성능에 대해서 비교]

Problem 1 : Code size - 68

Status - 20

Problem 2 : Code size - 88

Status - 34

Problem 1과 Problem 2의 Code size를 비교했을 땐, Problem 1이 훨씬 작았다. Status를 비교해봤을 땐, Problem 2가 1.5배나 더 컸다. 10 팩토리얼을 계산할 때는 Problem 1을 해결하면서 이용한 Second Operation이 성능이 더 좋았다.

구현하면서 느낀 차이와 어려움 : Problem 2를 해결하면서 이용한 MUL 명령어를 사용하면 10 팩토리얼을 구현하기가 훨씬 쉬웠다. 곱셈을 하는 방식 자체가 인간들이 늘 쓰는 방법과 어셈블리어에서 쓰는 방법이 같았다. MUL은 인간 친화적인 명령어라고 생각한다. Second operation을 이용한 방법은 MUL 명령어를 사용했을 때 보다 코드 구현에 시간이 더 걸렸다. 명령어가 생각보

다 익숙치 않았기도 했고, 자꾸 엉뚱한 값이 나왔다. ADD만 써서는 내가 원하는 값을 구할 수 없고 MOV도 적절히 사용해야 한다는 걸 깨달았다.

Problem1과 Problem2를 풀면서 도대체 재귀함수는 어떻게 구현할 수 있을까? 라는 고민을 했다. 팩토리얼 문제는 재귀함수의 아주 단골 문제가 아니던가! 매개변수는 어떻게 표현할 수 있을까? 재귀함수를 생각하기 힘드니 반복문도 생각해 보았다. 반복문을 쓰면 코드의 재사용성도 높이니 여러모로 좋은 방법 같았다. 그런데 LSL는 상수값만 취급하고 레지스터 값은 취급하지 않았다. 레지스터를 써야 반복문에서 레지스터를 증가시키며 뭐라도 할텐데... 이 방법은 생각하다가 포기하고 unrolling 기법으로 풀었다.

이론수업에서 교수님께 여쭙봤을 때, 무조건적으로 Second operation을 쓸 때 성능이 좋은 것은 아니라고 하셨다. MUL명령어의 max cycle이 16이었나 17이었나 그랬는데, Second operation을 쓰면서 이 max cycle을 넘어서면 그 때는 MUL을 쓰는 것이 낫다고 그러셨다. 이번 과제에서는 Second operation이 성능이 더 좋으니 max cycle을 안 넘겼나 보다. MUL의 최대 cycle을 넘는 경우는 어떤 때일 까 궁금하다. 이걸 알면 성능을 위해 MUL과 Second operation을 골라서 쓸 수 있을 텐데 말이다. 직접 비교하는 수 밖에 없을까? 고민을 유발하는 과제였다.

5. Reference

이번 과제를 하면서 참고한 문서는 없었습니다.