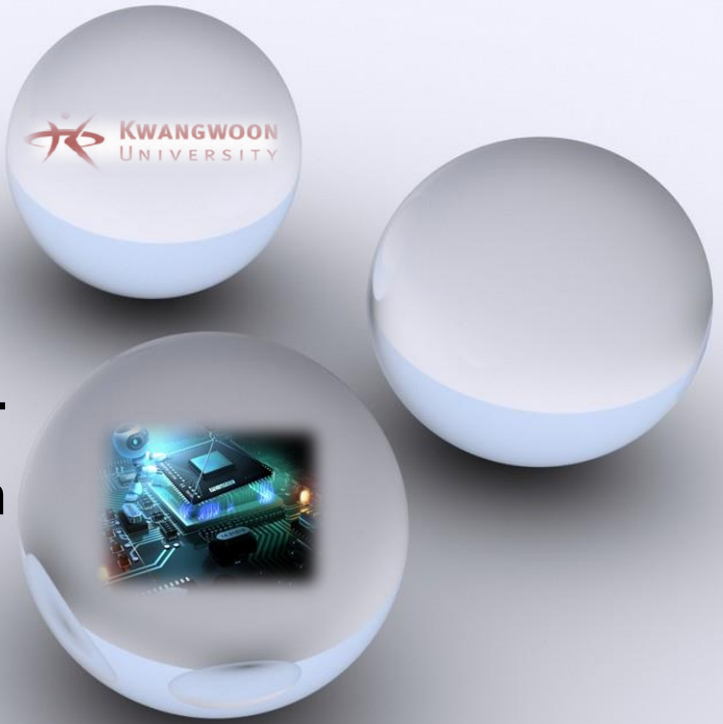
어셈블리프로그램 설계 및 실습



Lab #4

Second Operand & Multiplication

Kwangwoon University
Embedded System Architecture Lab

학습 목표

- Multiplication operation을 사용하는 것과 Second operand를 사용한 코드의 성능 차이를 비교해 보자
- Operand의 순서에 따른 성능의 차이에 대해 알아보자

Second operand & Multiplication operation

- Second operand
  - Add 연산과 Shift 연산 만을 이용해 Multiplication 연산을 수행
  - Ex) add r0,r0,r0, lsl #2 ; -> r0 + (r0<<2) = r0 x 5

- Multiplication operation
  - MUL instruction을 이용하여 Multiplication연산을 수행

Instruction

- MUL{cond} Rd, Rm, Rs
  - Rm에 저장된 값과 Rs값을 곱하여 Rd에 저장
  - Rd는 Rm과 같은 레지스터를 가질 수 없음
  - Ex) mov r0,#1 ; mov r1,#2 ; MUL r2,r0,r1; → r2=1*2 =2
- LSL #number
  - Number 만큼 왼쪽으로 bit stream을 shift
  - Ex) mov r0,#2; mov r0,r0,LSL #2 → r0 = 2*4 = 8
- CMP Rd,Rn
  - Rd와 Rn을 비교하여 flag값을 변경시켜 준다

Exercise(1/2)

- 구구단 7단을 Second operand만을 사용하여 작성하시오
  - 저장되는 Memory의 시작 주소 : 0x40000000
  - 오름차순으로 순서대로 저장할것

- 위에서 작성한 구구단 코드를 Multiplication Operation으로 수정

Exercise (2/2)

Second operand

```
    AREA ARMex, CODE, READONLY
    ENTRY

start
    ldr r0, tempaddr
    mov r1,#7                ;r1=7
    add r2,r1,lsl #1         ;r2=7*2
    add r3,r1,r1,lsl #1      ;r3=7*3
    add r4,r1,lsl #2         ;r4=7*4
    add r5,r1,r1,lsl #2      ;r5=7*5
    add r6,r2,r4             ;r6=7*6
    rsb r7,r1,r1,lsl #3      ;r7=7*7
    mov r8,r1,lsl #3         ;r8=7*8
    add r9,r1,r1,lsl #3      ;r9=7*9

    str r1,[r0],#4
    str r2,[r0],#4
    str r3,[r0],#4
    str r4,[r0],#4
    str r5,[r0],#4
    str r6,[r0],#4
    str r7,[r0],#4
    str r8,[r0],#4
    str r9,[r0]

    mov pc,lr

tempaddr & &40000

    end
```

Multiplication

```
    AREA ARMex, CODE, READONLY
    ENTRY

start
    ldr r0, tempaddr
    mov r1,#7                ;r1=7
    mov r10,#2
    mul r2,r1,r10            ;r2=7*2
    add r10,r10,#1
    mul r3,r1,r10            ;r3=7*3
    add r10,r10,#1
    mul r4,r1,r10            ;r4=7*4
    add r10,r10,#1
    mul r5,r1,r10            ;r5=7*5
    add r10,r10,#1
    mul r6,r1,r10            ;r6=7*6
    add r10,r10,#1
    mul r7,r1,r10            ;r7=7*7
    add r10,r10,#1
    mul r8,r1,r10            ;r8=7*8
    add r10,r10,#1
    mul r9,r1,r10            ;r9=7*9

    str r1,[r0],#4
    str r2,[r0],#4
    str r3,[r0],#4
    str r4,[r0],#4
    str r5,[r0],#4
    str r6,[r0],#4
    str r7,[r0],#4
    str r8,[r0],#4
    str r9,[r0]

    mov pc,lr

tempaddr & &40000

    end
```

Problem

- Problem 1
  - 1에서 10까지의 Factorial값을 Second operand로 구현하시오
    - Factorial값이란 1부터 해당하는 값까지의 곱을 말함
    - ex) 10! = 10x9x8x7x6x5x4x3x2x1 = 3,628,800
    - Factorial값을 지정된 memory 주소(0x40000)에 순서대로 저장
- Problem 2
  - Problem 1에서 작성한 코드를 **Multiplication operation**으로 수정
  - Problem 1과 Problem2의 차이와 성능에 대해서 비교

Homework

- 제출기한
  - Soft copy
    - 2019.10.4(금) 16시 29분 59초까지 u-campus에 제출
    - 압축 파일은 각 과제 파일을 모아서 제출(ini파일 반드시 포함)
    - 압축파일 형식
      - Assignment_(번호)_(학번).zip
        - ex) Assignment_3_2012722069.zip
    - 하드카피 제출 X

Q&A

Thank you