어셈블리프로그래밍 설계 및 실습

Lab #3

Control flow & Data processing

Kwangwoon University
Embedded System Architecture Lab

학습 목표

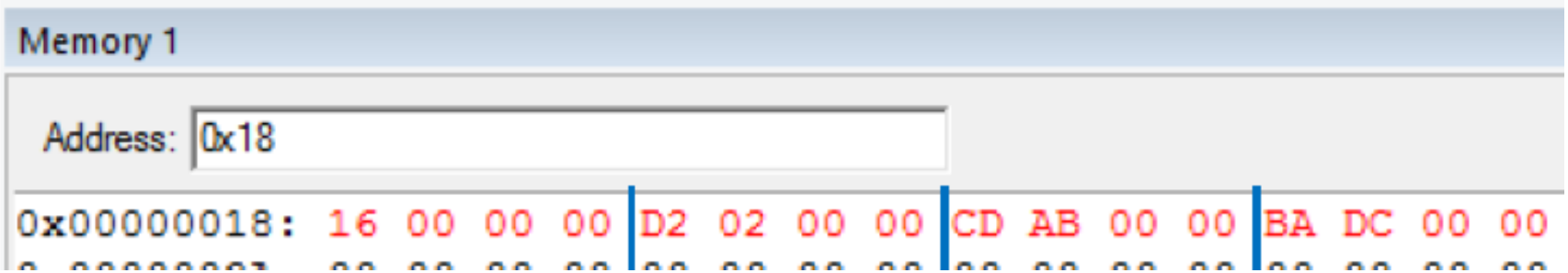
- Memory로부터 값을 받아 이를 이용해 원하는 연산을 수행해 보자
- Assembly code를 작성할 때 performance를 생각하며 작성해 보자
- Branch 명령어와 conditional execution의 차이를 알아보자
- Memory를 할당하여 data를 저장하여 보자

Instruction (1/3)

- MUL{cond} Rd, Rm, Rs
  - Rm에 저장된 값과 Rs값을 곱하여 Rd에 저장
  - Ex) mov r0,#1; mov r1,#2; MUL r2,r0,r1; ➔r2=1*2 =2
- LSL #number
  - Number만큼 왼쪽으로 bit stream을 shift
  - Ex) mov r0,#2; mov r0,r0,LSL #2; ➔r0 = 2*4 = 8
  - Ex) lsl r0, r1, r2 ➔ r0 = r1<<r2
- CMP Rd,Rn
  - Rd- Rn연산을 통해 flag값을 update시켜 줌
- B{cond} label
  - 작성한 label의 첫 번째 instruction으로 이동

Instruction (2/3)

- DCD
  - **4byte 단위**로 메모리를 할당 및 해당 값으로 초기화
  - {label} **DCD** expr{,expr} or {label} **&** expr{,expr}
    - expr : 숫자들 또는 숫자 식
    - 숫자 식 : **숫자 상수**, **숫자 변수**, **바이너리 연산자** 및 **괄호**의 조합으로 구성되는 식
      - Ex) (a*22)
  - Ex)
    - Values DCD 1,2,3,4
    - Values DCD 2_1101, 2_1000, 2_1111, 2_0010
    - Values DCD 0x16, 0x2d2, 0xabcd, 0xdcba



Instruction (3/3)

- DCB
  - **1byte 단위**로 메모리를 할당 및 해당 값으로 초기화
  - {label} **DCB** expr{,expr} or {label} **=** expr {,expr}
    - $-128 \leq \mathbf{expr} \leq 255$
  - DCB 다음에 명령어가 나오면 ALIGN 지시어를 사용하여 명령어를 정렬
  - Ex) C_string DCB "Hello_world",0
    - 0을 넣어준 이유는 문자열의 끝을 표현하기 위함이다.

Exercise (1/2)

- 주어진 함수를 branch문과 condition execution문을 이용하여 각각 작성하시오

  - $f(x) = \begin{cases} x^5 - x^3 (x \geq 0) \\ x^6 + x^4 (x < 0) \end{cases}$
  - Array[6] = {3,2,-3,-2,10,0}
    - 해당 배열을 DCD를 이용하여 배열에 저장 후 LDR을 이용해 불러옴
  - Store address = 0x40000000
    - 각 결과값을 STR을 통해 위의 주소로 각각 저장

Exercise (2/2)

- Conditional execution

```
1     AREA ARMex, CODE, READONLY
2         ENTRY
3  Main
4     LDR r0, =Value1
5     LDR r1, Address1
6     MOV r6, #6            ;입력 할 개수
7
8  Loop
9     LDR r2, [r0], #4
10    CMP r2, #0            ;x - 0?
11
12    ;x>=0
13    MULGE r3, r2, r2      ;r3 = x^2
14    MULGE r4, r3, r3      ;r4 = x^4
15    MULGE r7, r3, r2      ;r7 = x^3
16    MULGE r8, r4, r2      ;r8 = x^5
17    SUBGE r2, r8, r7      ;r2 = x^5 - x^3
18
19    ;x<0
20    MULLT r3, r2, r2      ;r3 = x^2
21    MULLT r4, r3, r3      ;r4 = x^4
22    MULLT r7, r3, r4      ;r7 = x^6
23    ADDLT r2, r7, r4      ;r2 = x^6 + x^4
24
25    STR r2, [r1], #4
26    ADD r5, r5, #1        ;loop 수 계산
27    CMP r5, r6            ;loop수 = 6?
28    BEQ Endline           ;6번 loop -> 종료
29    B Loop                ;loop
30
31 Endline
32    MOV pc, lr
33
34 Value1 DCD 3, 2, -3, -2, 10, 0
35    ;DEC  216, 24, 810, 80, 99000, 0
36    ;HEX   D8, 18, 32A, 50, 182B8, 0
37 Address1 DCD &40000000
38    END
```

- Using branch instruction

```
1     AREA ARMex, CODE, READONLY
2         ENTRY
3  Main
4     LDR r0, =Value1
5     LDR r1, Address1
6     MOV r6, #6            ;입력 할 개수
7
8  Loop
9     LDR r2, [r0], #4
10    CMP r2, #0            ;x - 0?
11
12    BLT Less
13    ;x>=0
14    MUL r3, r2, r2   ;r3 = x^2
15    MUL r4, r3, r3   ;r4 = x^4
16    MUL r7, r3, r2   ;r7 = x^3
17    MUL r8, r4, r2   ;r8 = x^5
18    SUB r2, r8, r7   ;r2 = x^5 - x^3
19    B Finish
20
21 Less ;x<0
22    MUL r3, r2, r2   ;r3 = x^2
23    MUL r4, r3, r3   ;r4 = x^4
24    MUL r7, r3, r4   ;r7 = x^6
25    ADD r2, r7, r4   ;r2 = x^6 + x^4
26
27 Finish
28    STR r2, [r1], #4
29    ADD r5, r5, #1        ;loop 수 계산
30    CMP r5, r6            ;loop수 = 6?
31    BEQ Endline           ;6번 loop -> 종
32    B Loop                ;loop
33
34 Endline
35    MOV pc, lr
36
37 Value1 DCD 3, 2, -3, -2, 10, 0
38    ;DEC  216, 24, 810, 80, 99000, 0
39    ;HEX   D8, 18, 32A, 50, 182B8, 0
40 Address1 DCD &40000000
41    END
```

Condition Field {cond}

Suffix	Flags	Meaning
EQ	Z set	Equal
NE	Z clear	Not Equal
CS/HS	C set	Carry set/Unsigned higher or same (unsigned >=)
CC/LO	C clear	Carry clear/Unsigned lower (unsigned <)
MI	N set	Minus/Negative
PL	N clear	Plus/Positive or zero
VS	V set	Overflow
VC	V clear	No overflow
HI	C set and Z clear	Higher (unsigned >)
LS	C clear or Z set	Lower or same (unsigned <=)
GE	N and V the same	Signed >=
LT	N and V different	Signed <
GT	Z clear, and N and V the same	Signed >
LE	Z set, or N and V different	Signed <=
AL	Any	Always (This suffix is normally omitted.)
NV	Reserved	

Problem (1/2)

- Problem 1
  - Strcmp 함수 만들기
  - 만약 두 문자열이 같은 경우에는 0x0A를, 그렇지 않은 경우에는 0x0B를 0x40000000 메모리 위치에 저장
- Problem 2
  - 배열을 정렬하여 0x4000000번지부터 저장하시오
  - Array[10] = {10,9,8,7,6,5,4,3,2,1}
    - 정렬 알고리즘을 사용하지 않아도 되며, 올바르게 정렬된 값 {1,2,3,4,5,6,7,8,9,10}순서로 저장되기만 하면됨.

Problem (2/2)

- Problem 3
  - 11+13+15+…27+29를 계산하여 결과값을 0x4000000번지에 저장
    - Loop을 이용한 방법
      - 11을 MOV연산으로 넣는 것이 아니라 shift 연산을 이용해야 함
        - 숫자는 #1만 사용
    - 일반화된 식인 n(n+10) 공식을 이용한 방법
      - 10을 사용하기 위해 Shift 연산을 이용해야함!
    - Unlooping을 이용한 방법
  - 위의 방법으로 각각 구현해 보고 각 방법을 비교해 보자

Homework

- 제출기한
  - Soft copy
    - 2019.9.27(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_2_2012722069.zip
    - 하드카피 제출 X

Q&A