어셈블리 프로그래밍 설계 및 실습

실험제목: Pseudo Instructions

실험일자: 2020 년 11 월 03 일 (화)

제출일자: 2020 년 11 월 09 일 (화)

학　　과: 컴퓨터공학과
담당교수: 이준환 교수님
학　　번: 2019202052
성　　명: 김 호 성

1. 제목 및 목적
   A. 제목
      Pseudo Instructions
   B. 목적
      - Pseudo instruction 이 assembler 에 의해 어떻게 실제 instruction 으로 변환되는지 직접 확인하고 이해한다.
      - Disassembly 를 해석하는 방법을 이해한다. (32bit 명령어들의 구조 이해)

2. 설계 (Design)

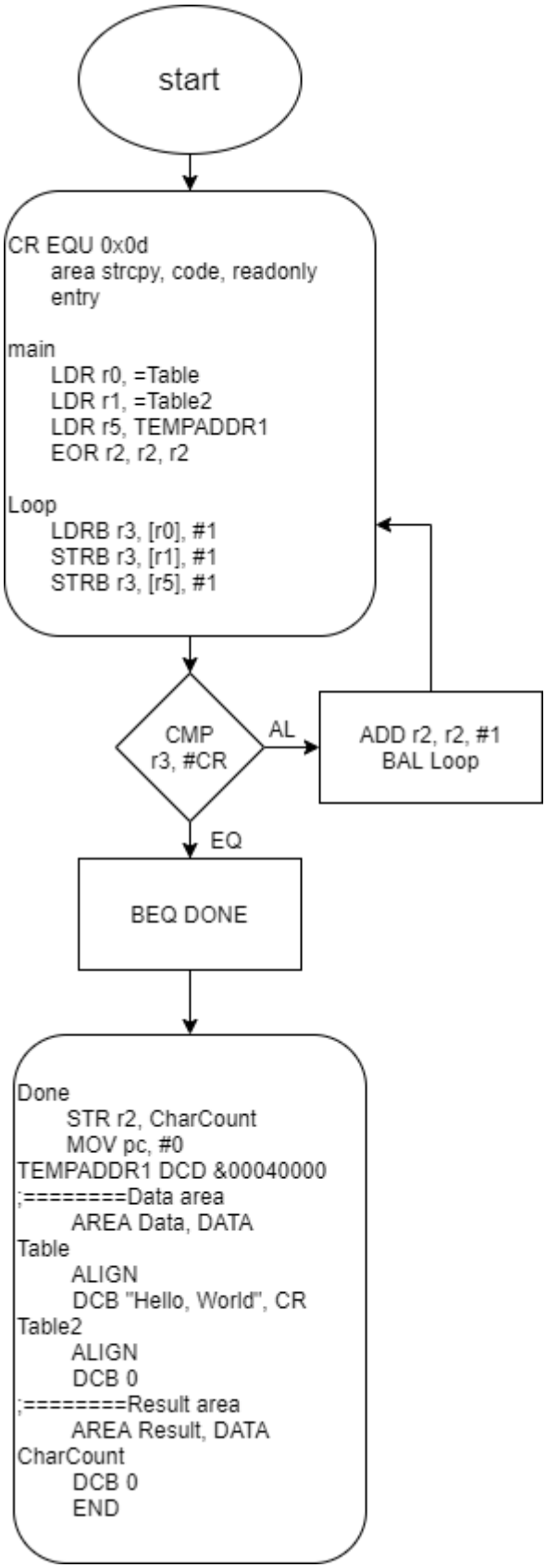
Problem1

A. Pseudo code

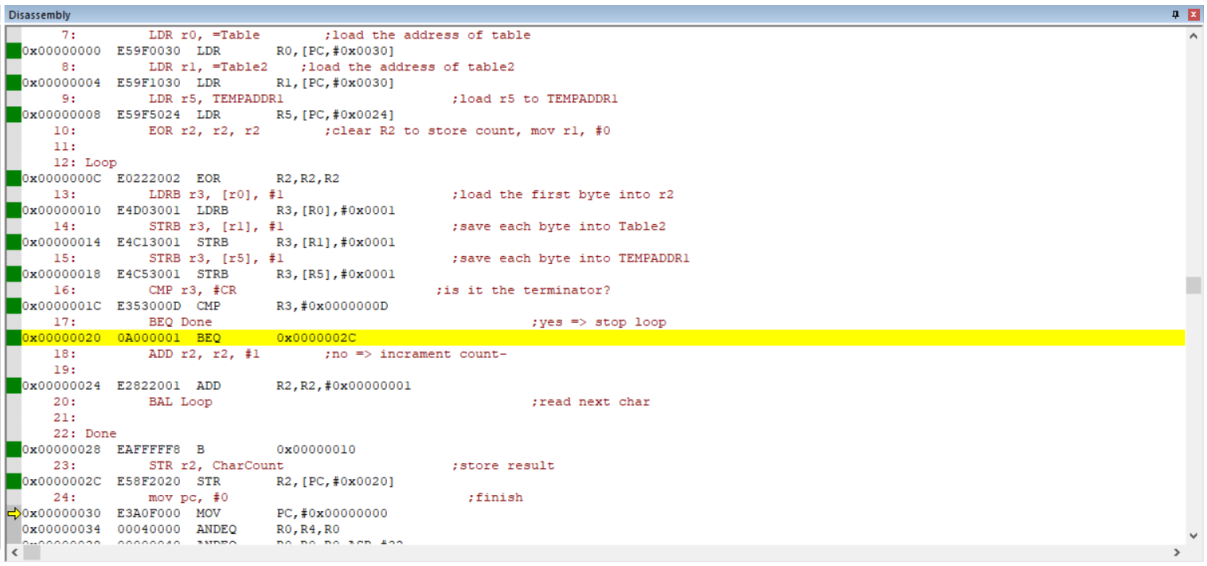
```
CR EQU 0x0d
        Area strcpy, code, readonly
                entry
main
        LDR r0 ← =Table
        LDR r1 ← =Table2
        LDR r5 ← TEMPADDR1
        EOR r2 ← r2, r2      ;clear R2 to store count, mov r1, #0
Loop
        LDRB r3 ← [r0], #1
        STRB r3 → [r1], #1
        STRB r3 → [r5], #1
        CMP r3, #CR
        BEQ Done
        ADD r2 = r2 + #1
        BAL Loop
Done
        STR r2 → CharCount
        MOV pc ← #0
TEMPADDR1 DCD &00040000
```

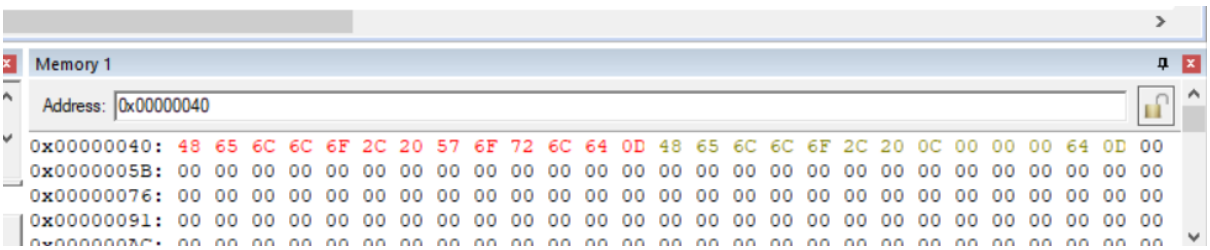
```
;========Data area
                AREA Data, DATA
Table
                ALIGN
                DCB “Hello, World”, CR
Table2
                ALIGN
                DCB 0
;========Result area
                AREA Result, DATA
CharCount
                DCB 0
                END
```

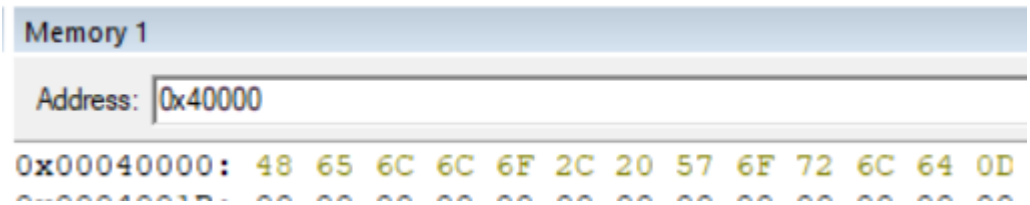
B. Flow chart 작성



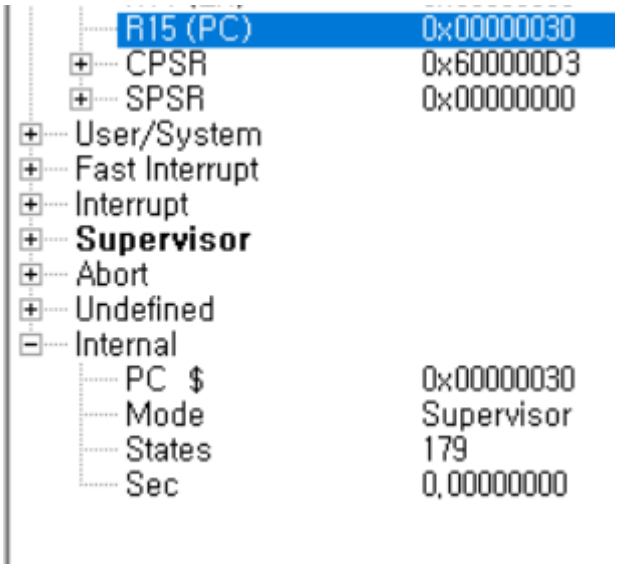
C. Result


Disassembly
7: LDR r0, =Table ;load the address of table
0x00000000 E59F0030 LDR R0,[PC,#0x0030]
8: LDR r1, =Table2 ;load the address of table2
0x00000004 E59F1030 LDR R1,[PC,#0x0030]
9: LDR r5, TEMPADDR1 ;load r5 to TEMPADDR1
0x00000008 E59F5024 LDR R5,[PC,#0x0024]
10: EOR r2, r2, r2 ;clear R2 to store count, mov r1, #0
11:
12: Loop
0x0000000C E0222002 EOR R2,R2,R2
13: LDRB r3, [r0], #1 ;load the first byte into r2
0x00000010 E4D03001 LDRB R3,[R0],#0x0001
14: STRB r3, [r1], #1 ;save each byte into Table2
0x00000014 E4C13001 STRB R3,[R1],#0x0001
15: STRB r3, [r5], #1 ;save each byte into TEMPADDR1
0x00000018 E4C53001 STRB R3,[R5],#0x0001
16: CMP r3, #CR ;is it the terminator?
0x0000001C E353000D CMP R3,#0x0000000D
17: BEQ Done ;yes => stop loop
0x00000020 0A000001 BEQ 0x0000002C
18: ADD r2, r2, #1 ;no => increment count-
19:
0x00000024 E2822001 ADD R2,R2,#0x00000001
20: BAL Loop ;read next char
21:
22: Done
0x00000028 EAFFFFF8 B 0x00000010
23: STR r2, CharCount ;store result
0x0000002C E58F2020 STR R2,[PC,#0x0020]
24: mov pc, #0 ;finish
0x00000030 E3A0F000 MOV PC,#0x00000000
0x00000034 00040000 ANDEQ R0,R4,R0


Memory 1
Address: 0x00000040
0x00000040: 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 0D 48 65 6C 6C 6F 2C 20 0C 00 00 00 64 0D 00
0x0000005B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000076: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000091: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00


Memory 1
Address: 0x40000
0x00040000: 48 65 6C 6C 6F 2C 20 57 6F 72 6C 64 0D

D. Performance



State = 179

```
Total RO  Size (Code + RO Data)                64 (   0.06kB)
Total RW  Size (RW Data + ZI Data)             24 (   0.02kB)
Total ROM Size (Code + RO Data + RW Data)      88 (   0.09kB)
```

Code size = 64

64 * 179 = 11,456

3. 고찰 및 결론

A. 고찰

- LDR r0, =Table => LDR R0, [PC, #0x0030] 현재 PC 값이 0x00000000 이므로 PC 와 0x =0030 를 더한 값인 0x00000030 이 R0 에 저장되는 명령과 같다.

- LDR r1, = Table2 => LDR R1, [PC, #0x0030]

- LDR r5, TEMPADDR1 => LDR R5, [PC, #0x0024]

B. 결론

명령어들은 다음 과 같이 이루어져 있다.

- LDR E59F0030, E59F1030, E59F5024
- EOR E0222002
- LDRB E4D03001
- STRB E4C13001, E4C53001

- CMP E353000D
- BEQ 0A000001
- ADD E2822001
- B EAFFFFF8
- STR E58F2020

모든 명령어들은 32bit 를 가지고 있는데 이중에서 가장 앞쪽의 4bit(MSB)는 CPSR 의 flag 를 뜻한다. N, Z, C, V 순이며, 만약 1110 이면 N, V, C flag 를 CPU 가 실행하도록 결정하는 것이다. 위의 그림에서 LDR, EOR, LDRB, STRB, CMP, ADD, B, STR 는 맨 앞의 수가 E 인 것을 확인할 수 있는데, 이것을 2 진법으로 나타내면 1110 이다. 따라서 N, Z, C flag 가 실행되도록 하는 명령어이다.
또, 명령어의 마지막 8bit(LSB)는 CPSR(current program status register)를 의미한다. 즉 현재 프로그램 상태를 의미하는 것인데, 0~4bit 는 M4~M0(Mode bit)를, 5 번째 bit 는 T(Thumb mode bit), 6 번째 bit 는 F(FIQ mask bit), 7 번째 bit 는 I(IRO mask bit)를 나타낸다.

4. 참고문헌

이준환교수님/디지털논리회로 2/광운대학교(컴퓨터정보공학부)/2020

공영호교수님/디지털논리회로 2/광운대학교(컴퓨터정보공학부)/2020