EDDI
Electronic Design
Development Institute

에디로봇아카데미

임베디드 마스터 Lv1 과정

제 5기

2023. 05. 19

최성효

확인 사항

EDDI
Electronic Design
Development Institute

cltd, shr, lea 확인 필요

어셈블리어 분석(1)

EDDI
Electronic Design Development Institute

ARM 어셈블리어

```
=> 0x00010420 <+0>:     push    {r11, lr}
   0x00010424 <+4>:     add     r11, sp, #4
   0x00010428 <+8>:     sub     sp, sp, #16
   0x0001042c <+12>:    mov     r3, #5
   0x00010430 <+16>:    str     r3, [r11, #-8]
   0x00010434 <+20>:    mov     r3, #0
   0x00010438 <+24>:    str     r3, [r11, #-16]
   0x0001043c <+28>:    mov     r3, #1
   0x00010440 <+32>:    str     r3, [r11, #-12]
   0x00010444 <+36>:    b       0x1047c <main+92>
   0x00010448 <+40>:    ldr     r3, [r11, #-12]
   0x0001044c <+44>:    cmp     r3, #0
   0x00010450 <+48>:    and     r3, r3, #1
   0x00010454 <+52>:    rsblt   r3, r3, #0
   0x00010458 <+56>:    cmp     r3, #1
   0x0001045c <+60>:    bne     0x10470 <main+80>
   0x00010460 <+64>:    ldr     r2, [r11, #-16]
   0x00010464 <+68>:    ldr     r3, [r11, #-12]
   0x00010468 <+72>:    add     r3, r2, r3
   0x0001046c <+76>:    str     r3, [r11, #-16]
   0x00010470 <+80>:    ldr     r3, [r11, #-12]
   0x00010474 <+84>:    add     r3, r3, #1
   0x00010478 <+88>:    str     r3, [r11, #-12]
   0x0001047c <+92>:    ldr     r2, [r11, #-12]
   0x00010480 <+96>:    ldr     r3, [r11, #-8]
   0x00010484 <+100>:   cmp     r2, r3
   0x00010488 <+104>:   ble     0x10448 <main+40>
   0x0001048c <+108>:   ldr     r2, [r11, #-16]
   0x00010490 <+112>:   ldr     r1, [r11, #-8]
   0x00010494 <+116>:   ldr     r0, [pc, #16]   ; 0x104ac <main+140>
   0x00010498 <+120>:   bl      0x10304 <printf@plt>
   0x0001049c <+124>:   mov     r3, #0
   0x000104a0 <+128>:   mov     r0, r3
   0x000104a4 <+132>:   sub     sp, r11, #4
--Type <RET> for more, q to quit, c to continue without paging--c
   0x000104a8 <+136>:   pop     {r11, pc}
   0x000104ac <+140>:   andeq   r0, r1, r12, asr #10
```

코드

```
#include <stdio.h>

int main(void)
{
        int  End=5;
        int  sum=0;
        for (int  i=1; i<=End; ++i){
                if(i%2==1){
                        sum+=i;
                }
        }
        printf("%d까지의 홀수의 합은 %d\n",End,sum);
        return 0;
}
```

어셈블리어 분석(2)

EDDI
Electronic Design Development Institute

어셈블리어

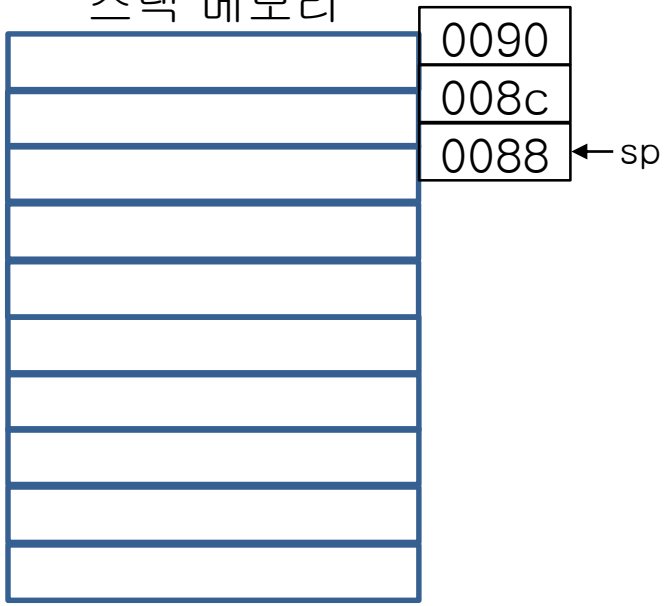
```
0x00010420 <+0>:     push    {r11, lr}
0x00010424 <+4>:     add     r11, sp, #4
0x00010428 <+8>:     sub     sp, sp, #16
0x0001042c <+12>:    mov     r3, #5
0x00010430 <+16>:    str     r3, [r11, #-8]
0x00010434 <+20>:    mov     r3, #0
0x00010438 <+24>:    str     r3, [r11, #-16]
0x0001043c <+28>:    mov     r3, #1
0x00010440 <+32>:    str     r3, [r11, #-12]
0x00010444 <+36>:    b       0x1047c <main+92>
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
```

레지스터

r11	0x20f14	134932
r12	0x3ff97000	1073311744
sp	0x40800090	0x40800090

r11	0x20f14	134932
r12	0x3ff97000	1073311744
sp	0x40800088	0x40800088

스택 메모리



push {r1-r9}
->push 안의 레지스터 개수 만큼 fp증가

push {r11,lr} 2*4[byte]= 8 [byte] 만큼 감소

어셈블리어 분석(3)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x00010420 <+0>:     push    {r11, lr}
0x00010424 <+4>:     add     r11, sp, #4
0x00010428 <+8>:     sub     sp, sp, #16
0x0001042c <+12>:    mov     r3, #5
0x00010430 <+16>:    str     r3, [r11, #-8]
0x00010434 <+20>:    mov     r3, #0
0x00010438 <+24>:    str     r3, [r11, #-16]
0x0001043c <+28>:    mov     r3, #1
0x00010440 <+32>:    str     r3, [r11, #-12]
0x00010444 <+36>:    b       0x1047c <main+92>
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
```

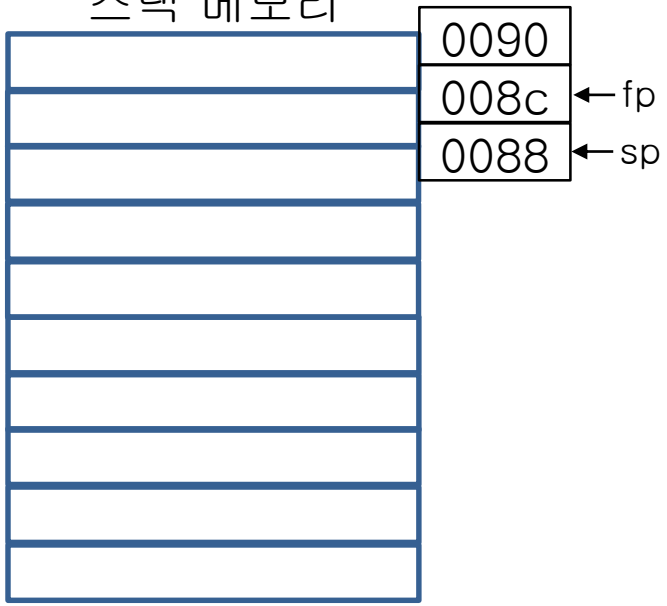
레지스터

r11	0x20f14	134932
r12	0x3ff97000	1073311744
sp	0x40800088	0x40800088

↓

r11	0x4080008c	1082130572
r12	0x3ff97000	1073311744
sp	0x40800088	0x40800088

스택 메모리



add r11,sp,#4

어셈블리어 분석(4)

EDDI
Electronic Design Development Institute

어셈블리어

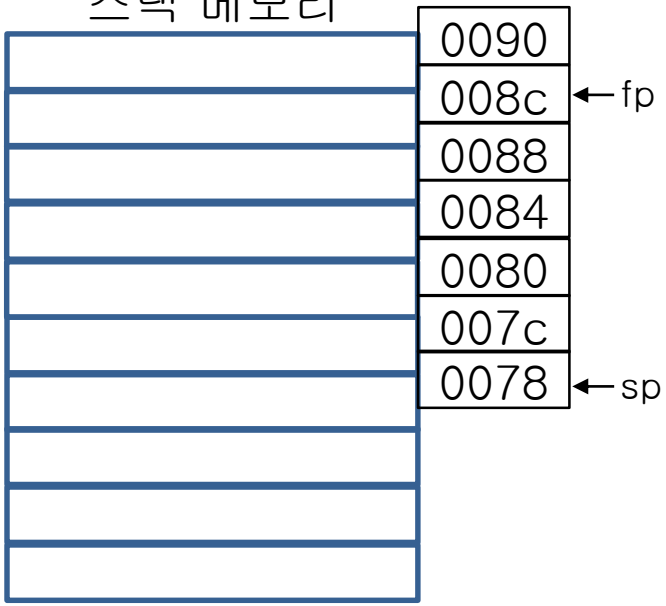
```
0x00010420 <+0>:     push    {r11, lr}
0x00010424 <+4>:     add     r11, sp, #4
0x00010428 <+8>:     sub     sp, sp, #16
0x0001042c <+12>:    mov     r3, #5
0x00010430 <+16>:    str     r3, [r11, #-8]
0x00010434 <+20>:    mov     r3, #0
0x00010438 <+24>:    str     r3, [r11, #-16]
0x0001043c <+28>:    mov     r3, #1
0x00010440 <+32>:    str     r3, [r11, #-12]
0x00010444 <+36>:    b       0x1047c <main+92>
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
```

레지스터

r11	0x4080008c	1082130572
r12	0x3ff97000	1073311744
sp	0x40800088	0x40800088

r11	0x4080008c	1082130572
r12	0x3ff97000	1073311744
sp	0x40800078	0x40800078

스택 메모리



sub sp,sp,#16

sp -=16

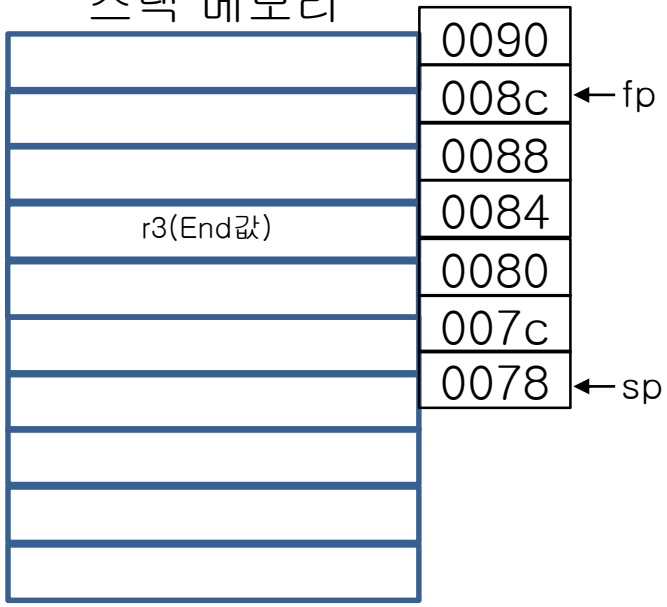
어셈블리어 분석(5)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x00010420 <+0>:     push    {r11, lr}
0x00010424 <+4>:     add     r11, sp, #4
0x00010428 <+8>:     sub     sp, sp, #16
0x0001042c <+12>:    mov     r3, #5
0x00010430 <+16>:    str     r3, [r11, #-8]
0x00010434 <+20>:    mov     r3, #0
0x00010438 <+24>:    str     r3, [r11, #-16]
0x0001043c <+28>:    mov     r3, #1
0x00010440 <+32>:    str     r3, [r11, #-12]
0x00010444 <+36>:    b       0x1047c <main+92>
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
```

스택 메모리



r3에 5대입 [End값]

r11-8=0084에 대입

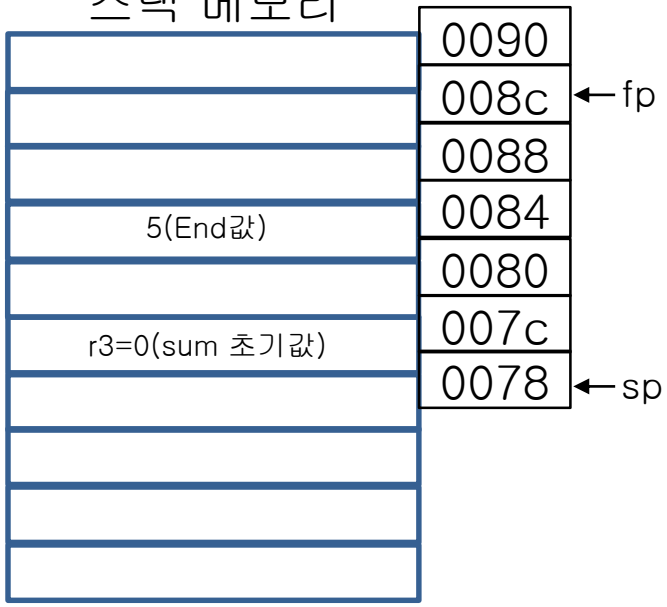
어셈블리어 분석(6)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x00010420 <+0>:     push    {r11, lr}
0x00010424 <+4>:     add     r11, sp, #4
0x00010428 <+8>:     sub     sp, sp, #16
0x0001042c <+12>:    mov     r3, #5
0x00010430 <+16>:    str     r3, [r11, #-8]
0x00010434 <+20>:    mov     r3, #0
0x00010438 <+24>:    str     r3, [r11, #-16]
0x0001043c <+28>:    mov     r3, #1
0x00010440 <+32>:    str     r3, [r11, #-12]
0x00010444 <+36>:    b       0x1047c <main+92>
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
```

스택 메모리



r3에 0대입 [sum 초기값]

r11-8=007c 에 대입

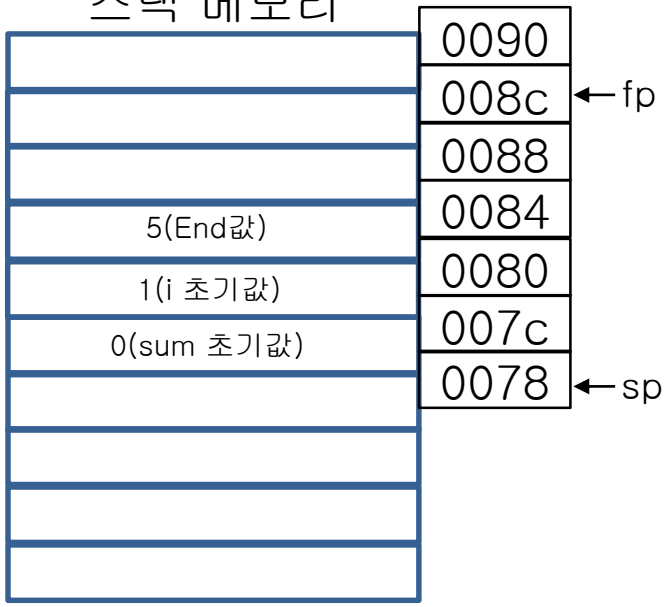
어셈블리어 분석(7)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x00010420 <+0>:     push    {r11, lr}
0x00010424 <+4>:     add     r11, sp, #4
0x00010428 <+8>:     sub     sp, sp, #16
0x0001042c <+12>:    mov     r3, #5
0x00010430 <+16>:    str     r3, [r11, #-8]
0x00010434 <+20>:    mov     r3, #0
0x00010438 <+24>:    str     r3, [r11, #-16]
0x0001043c <+28>:    mov     r3, #1
0x00010440 <+32>:    str     r3, [r11, #-12]
0x00010444 <+36>:    b       0x1047c <main+92>
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
```

스택 메모리



r3에 1대입 [i 초기값]

r11-12=0080 에 대입

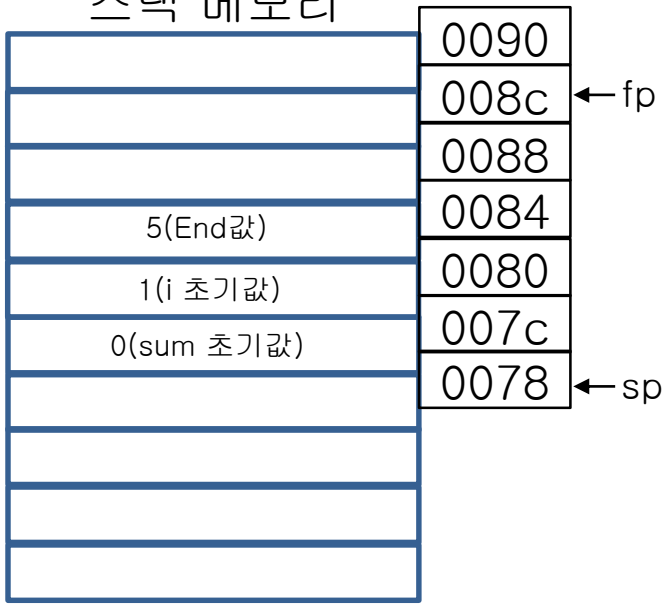
어셈블리어 분석(8)

어셈블리어

```
0x00010420 <+0>:     push    {r11, lr}
0x00010424 <+4>:     add     r11, sp, #4
0x00010428 <+8>:     sub     sp, sp, #16
0x0001042c <+12>:    mov     r3, #5
0x00010430 <+16>:    str     r3, [r11, #-8]
0x00010434 <+20>:    mov     r3, #0
0x00010438 <+24>:    str     r3, [r11, #-16]
0x0001043c <+28>:    mov     r3, #1
0x00010440 <+32>:    str     r3, [r11, #-12]
0x00010444 <+36>:    b       0x1047c <main+92>
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
```

```
0x0001047c <+92>:    ldr     r2, [r11, #-12]
0x00010480 <+96>:    ldr     r3, [r11, #-8]
0x00010484 <+100>:   cmp     r2, r3
0x00010488 <+104>:   ble     0x10448 <main+40>
```

스택 메모리



r2에 1(i 초기값)대입
r3에 5(End값) 대입
r2, r3 비교
왼쪽(r2)가 오른쪽(r3)보다 작거나 같으면 점프

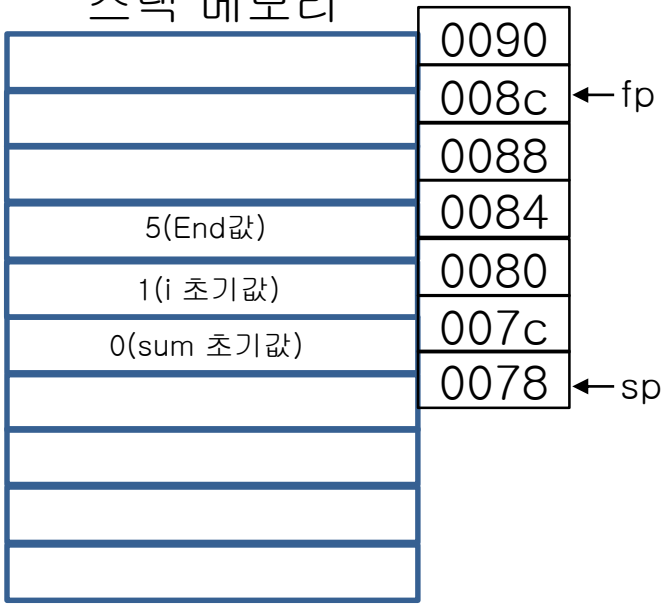
어셈블리어 분석(9)

어셈블리어

```
0x0001047c <+92>:    ldr     r2, [r11, #-12]
0x00010480 <+96>:    ldr     r3, [r11, #-8]
0x00010484 <+100>:   cmp     r2, r3
0x00010488 <+104>:   ble     0x10448 <main+40>
```

```
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
0x0001045c <+60>:    bne     0x10470 <main+80>
```

스택 메모리



r3에 1(i 값)대입
r3와 0 비교 (?)
r3 와 1 and 연산
-> r3가 홀수이면 r3가 1이되고 짝수이면 0이 된다.
rsblt r3, r3, #0 (reverse subtract)
-> r3=0-r3 (i값 음수처리부분)
bne (r3와 1이 같지 않으면 점프)
-> i%2==1 부분 , r3가 1과 같음으로 계속 진행

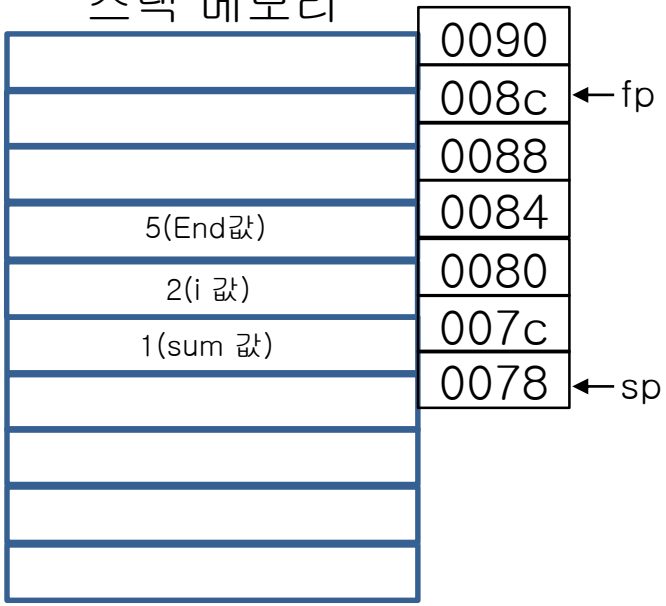
어셈블리어 분석(10)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
0x0001045c <+60>:    bne     0x10470 <main+80>
0x00010460 <+64>:    ldr     r2, [r11, #-16]
0x00010464 <+68>:    ldr     r3, [r11, #-12]
0x00010468 <+72>:    add     r3, r2, r3
0x0001046c <+76>:    str     r3, [r11, #-16]
0x00010470 <+80>:    ldr     r3, [r11, #-12]
0x00010474 <+84>:    add     r3, r3, #1
0x00010478 <+88>:    str     r3, [r11, #-12]
0x0001047c <+92>:    ldr     r2, [r11, #-12]
0x00010480 <+96>:    ldr     r3, [r11, #-8]
0x00010484 <+100>:   cmp     r2, r3
0x00010488 <+104>:   ble     0x10448 <main+40>
```

스택 메모리



r2에 0(sum 초기값) 대입
r3에 1(I 값)대입
r3 +=r2 (sum+=I 부분)후 r11-16에 대입
r3에 1(I 값) 대입
r3에 1 더한 후 (i++부분) r11-12에 대입
r2값에 2(i 값)대입
r3에 5(End값) 대입
r2(i),r3(end) 비교 후 r2가 작거나 같으면 점프

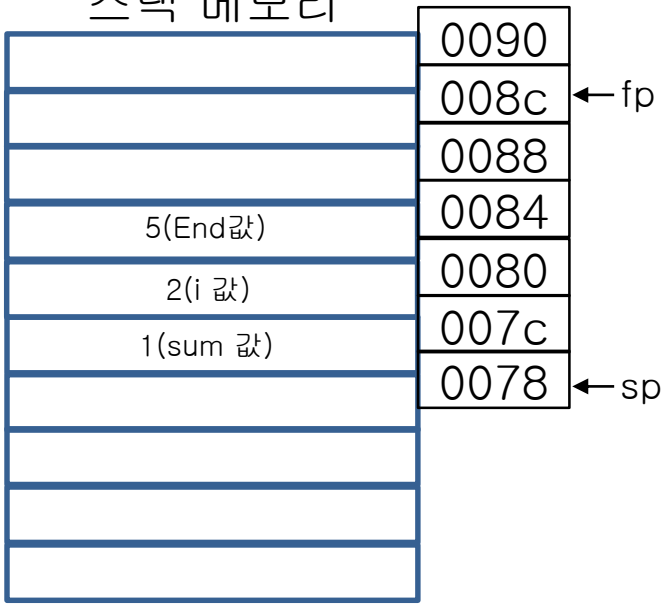
어셈블리어 분석(11)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x00010448 <+40>:    ldr     r3, [r11, #-12]
0x0001044c <+44>:    cmp     r3, #0
0x00010450 <+48>:    and     r3, r3, #1
0x00010454 <+52>:    rsblt   r3, r3, #0
0x00010458 <+56>:    cmp     r3, #1
0x0001045c <+60>:    bne     0x10470 <main+80>
0x00010460 <+64>:    ldr     r2, [r11, #-16]
0x00010464 <+68>:    ldr     r3, [r11, #-12]
0x00010468 <+72>:    add     r3, r2, r3
0x0001046c <+76>:    str     r3, [r11, #-16]
0x00010470 <+80>:    ldr     r3, [r11, #-12]
0x00010474 <+84>:    add     r3, r3, #1
0x00010478 <+88>:    str     r3, [r11, #-12]
0x0001047c <+92>:    ldr     r2, [r11, #-12]
0x00010480 <+96>:    ldr     r3, [r11, #-8]
0x00010484 <+100>:   cmp     r2, r3
0x00010488 <+104>:   ble     0x10448 <main+40>
```

스택 메모리



i가 2일 경우
sum 부분 없이 i++

i=3,4,5일 경우는 생략

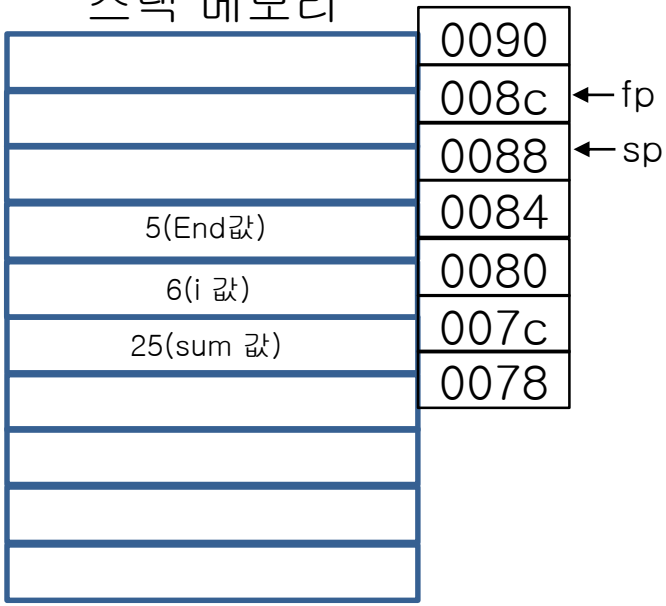
어셈블리어 분석(12)

EDDI
Electronic Design Development Institute

어셈블리어

```
0x00010460 <+64>:    ldr     r2, [r11, #-16]
0x00010464 <+68>:    ldr     r3, [r11, #-12]
0x00010468 <+72>:    add     r3, r2, r3
0x0001046c <+76>:    str     r3, [r11, #-16]
0x00010470 <+80>:    ldr     r3, [r11, #-12]
0x00010474 <+84>:    add     r3, r3, #1
0x00010478 <+88>:    str     r3, [r11, #-12]
0x0001047c <+92>:    ldr     r2, [r11, #-12]
0x00010480 <+96>:    ldr     r3, [r11, #-8]
0x00010484 <+100>:   cmp     r2, r3
0x00010488 <+104>:   ble     0x10448 <main+40>
0x0001048c <+108>:   ldr     r2, [r11, #-16]
0x00010490 <+112>:   ldr     r1, [r11, #-8]
0x00010494 <+116>:   ldr     r0, [pc, #16]   ; 0x104ac <main+140>
0x00010498 <+120>:   bl      0x10304 <printf@plt>
0x0001049c <+124>:   mov     r3, #0
0x000104a0 <+128>:   mov     r0, r3
0x000104a4 <+132>:   sub     sp, r11, #4
0x000104a8 <+136>:   pop     {r11, pc}
0x000104ac <+140>:   andeq   r0, r1, r12, asr #10
```

스택 메모리



i가 6일 경우,
r2(i값)이 r3(End값)보다 큼으로, +108번 줄 실행
r2에 25(sum값) 대입
r1에 5(End값) 대입
r0에 print 후 돌아올 주소 저장 후 print
r3에 0대입
r0에 r3(0)대립
sp=r11-4

어셈블리어 분석(13)

EDDI
Electronic Design Development Institute

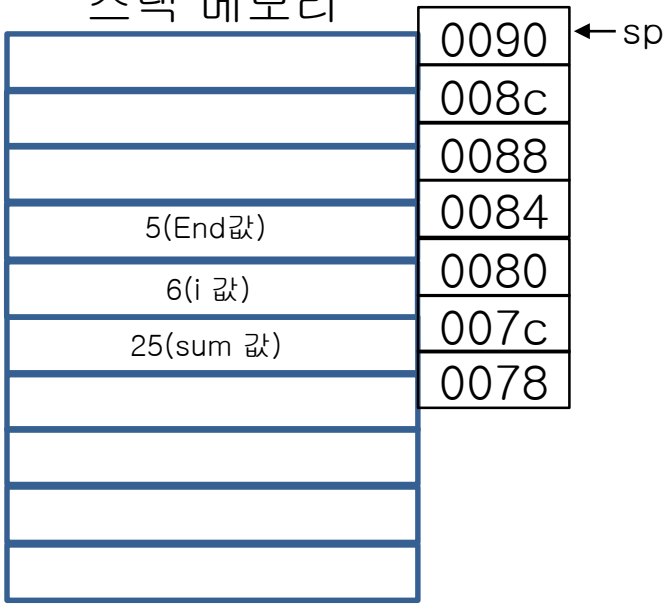
어셈블리어

```
0x00010460 <+64>:    ldr     r2, [r11, #-16]
0x00010464 <+68>:    ldr     r3, [r11, #-12]
0x00010468 <+72>:    add     r3, r2, r3
0x0001046c <+76>:    str     r3, [r11, #-16]
0x00010470 <+80>:    ldr     r3, [r11, #-12]
0x00010474 <+84>:    add     r3, r3, #1
0x00010478 <+88>:    str     r3, [r11, #-12]
0x0001047c <+92>:    ldr     r2, [r11, #-12]
0x00010480 <+96>:    ldr     r3, [r11, #-8]
0x00010484 <+100>:   cmp     r2, r3
0x00010488 <+104>:   ble     0x10448 <main+40>
0x0001048c <+108>:   ldr     r2, [r11, #-16]
0x00010490 <+112>:   ldr     r1, [r11, #-8]
0x00010494 <+116>:   ldr     r0, [pc, #16]   ; 0x104ac <main+140>
0x00010498 <+120>:   bl      0x10304 <printf@plt>
0x0001049c <+124>:   mov     r3, #0
0x000104a0 <+128>:   mov     r0, r3
0x000104a4 <+132>:   sub     sp, r11, #4
0x000104a8 <+136>:   pop     {r11, pc}
0x000104ac <+140>:   andeq   r0, r1, r12, asr #10
```

r11	0x4080008c	1082130572
r12	0x81010101	-2130640639
sp	0x40800088	0x40800088
lr	0x1049c	66716
pc	0x104a8	0x104a8 <main+136>

r11	0x20f14	134932
r12	0x81010101	-2130640639
sp	0x40800090	0x40800090
lr	0x1049c	66716
pc	0x3fe2d958	0x3fe2d958

스택 메모리



pop 명령어 후 sp는 0088+8 =0090으로 이동
pc에는 복귀 주소값(lr) 저장

andeq??????