

학번: 201302423 이름: 신종욱

과제 1 Lab2-1.s

```
;201302423 신종욱
LDR R0, =0x5      ; R0 레지스터에 5를 할당
MOV R1, #25        ; R1 레지스터에 25를 할당
LDR R3, =0x2        ; R3 레지스터에 2를 할당
ADD R4, R0, R1      ; R4 레지스터에 R0와 R1의 합을 할당
LSR R5, R4, R3      ; R5 레지스터에 R4를 R3의 값만큼 오른쪽으로 SHIFT연산
```

실행 결과

Registers	
Register	Value
Core	
R0	0x00000005
R1	0x00000019
R2	0x00000000
R3	0x00000002
R4	0x0000001E
R5	0x00000007
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000200
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x000000D4
+ xPSR	0x01000000
+ Banked	
+ System	
- Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	5
Sec	0,00000042

과제 2 Lab2-2.s

```

;201302423 신종욱
LDR R1, =0x0      ; R1 레지스터에 0을 할당
LDR R2, =0x0      ; R2 레지스터에 0을 할당

L1
CMP R2, #10        ; R2와 상수 10값을 비교 R2-10으로 계산하여 상태플래그로 전달
BGE L2             ; 계산된 값이 00이상이면 L2로 분기
ADD R1, R1, R2      ; R1 레지스터에 R1과 R2의 합을 할당
ADD R2, R2, #1      ; R2레지스터에 R2와 상수 1의 합을 할당 (INCREMENT)
BLT L1             ; 상태플래그가 음수라면 L1으로 분기

L2

```

실행 결과

Registers	
Register	Value
Core	
R0	0x00000000
R1	0x0000002D
R2	0x0000000A
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000200
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x000000D4
+ xPSR	0x61000000
+ Banked	
+ System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	76
Sec	0,00000633

과제 3 Lab2-3.s

```

;201302423 신종욱
MOV R1, #10      ;R1레지스터에 상수 10을 저장
MOV R2, #7       ;R2레지스터에 상수 7을 저장
MOV R0, #5       ;R0레지스터에 상수 5을 저장
MUL R1,R1,R2     ;R1레지스터에 R1과 R2의 곱을 저장

CMP R0,#5        ;R0-5를 계산하여 비교값을 기억
BEQ L3           ;CMP에서 계산한값과 비교하여 같으면 밑으로 분기하고 다르다면 바로 다음 명령어로 간다.
ADD R1,R1,R0     ; R1레지스터에 R1와 R0의 합을 할당
SUB R1,R1,R2     ; R1레지스터에 R1-R2의 값을 할당
L3

```

실행 결과

Register	Value
Core	
R0	0x00000005
R1	0x00000046
R2	0x00000007
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000200
R14 (LR)	0xFFFFFFFF
R15 (PC)	0x0000000A
+ xPSR	0x61000000
+ Banked	
+ System	
- Internal	
Mode	Thread
Privilege	Privileged
Stack	MSP
States	8
Sec	0,00000067

느낀점 : 이전에 배웠던 어셈블리어 지식을 프로그래밍으로 코딩해봐서 신기했다. 상태플래그는 어디에 저장되는지 궁금하다.