

**B711222 박조은**

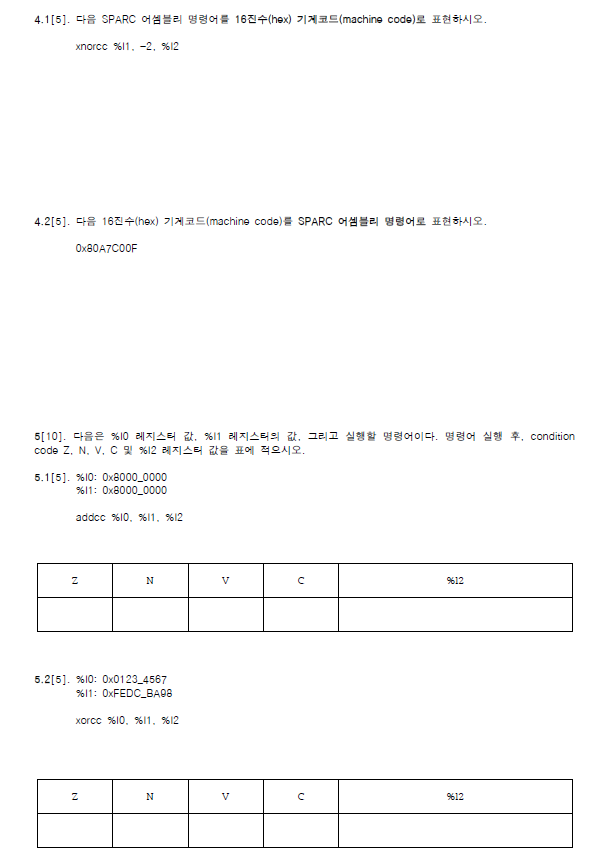
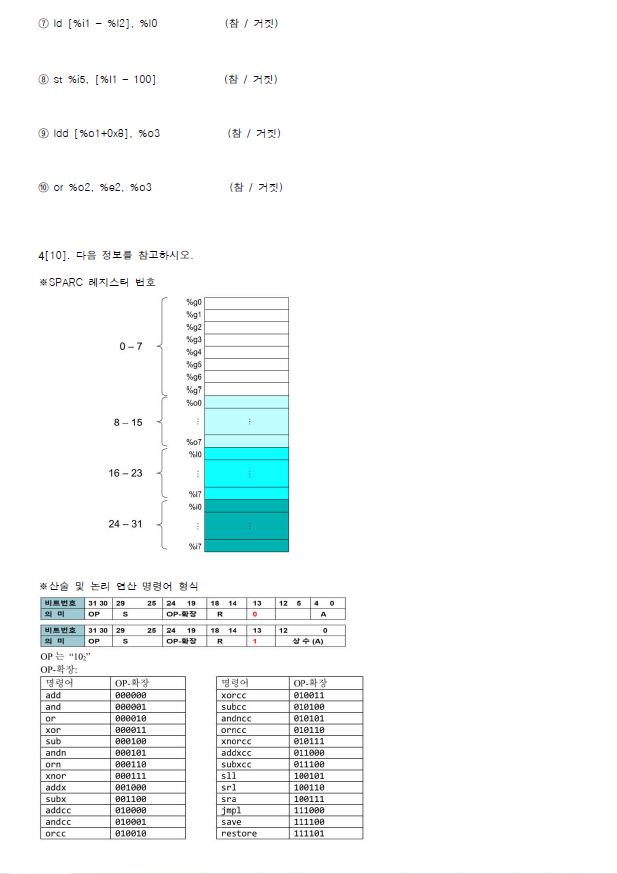
SPARC assembly level에서는 덧셈을 표현하기 위해 3개의 operand가 필요하며, 곱셈은 subroutine과 nop를 사용해야 한다. 또한 그 연산 값을 변수에 할당하기 위해서는 2개의 operand가 필요하다. 즉 ⑥번 식과 같은 간단한 연산도 여러 줄의 logic을 취할 수밖에 없다.

원래 순서: st regrd, [address]  
오답 수정: st %o0, [%l0 + 0]

원래 순서: ld [address], regrd  
오답 수정: ld [%l1 + 1], %l0

원래 순서: add regrs1, reg\_or\_imm, regrd  
오답 수정: add %l0, 10, %l1

ld 명령어의 형식은 다음과 같아야 하기 때문이다.  
ld [R + A], S



%r0부터 %r31까지의 32개 register들은 순서대로  
%g0~%g7, %o0~%o7, %l0~%l7, %i0~%i7의 다른 이름으로 표현 가능하다. 즉 e로 시작하는 register 이름은 없다.

0

0xFFFF\_FFFF

0

0

1

0

1

1

0x0000\_0000

1

subcc %i7, %o7, %g0

0xA4BC7FFE

