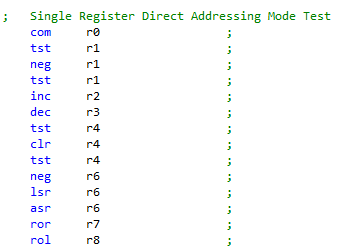
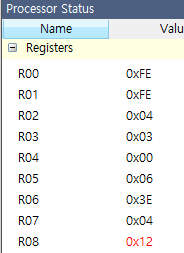
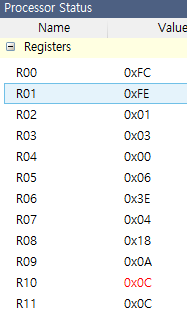
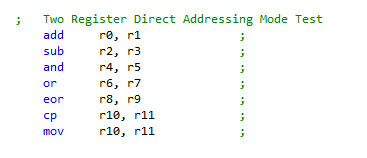
**Single Register Direct Addressing Mode Test**

|  |  |  |  |
| --- | --- | --- | --- |
| **ASM code** | **Register value**  **before execution** | **Register value after execution** | **Meaning of**  **instruction** |
| com r0 ; | 0000 0001 | 1111 1110 | 1’s complement of the contents in r0 |
| tst r1 ; | 0000 0010 | 0000 0010 | Tests if a register is  zero or negative |
| neg r1 ; | 0000 0010 | 1111 1110 | 2’s complement of the contents in r1 |
| tst r1 ; | 1111 1110 | 1111 1110 | Tests if a register is  zero or negative |
| inc r2 ; | 0000 0011 | 0000 0100 | 1 increase in r2 |
| dec r3 ; | 0000 0100 | 0000 0011 | 1 decrease in r3 |
| tst r4 ; | 0000 0101 | 0000 0101 | Tests if a register is  zero or negative |
| clr r4 ; | 0000 0101 | 0000 0000 | Clear the value in r4 (set 0) |
| tst r4 ; | 0000 0000 | 0000 0000 | Tests if a register is  zero or negative |
| neg r6; | 0000 0111 | 1111 1001 | 2’s complement of the contents in r1 |
| lsr r6; | 1111 1001 | 0111 1100 | Shift all bits in r6 one place to the right |
| asr r6 ; | 0111 1100 | 0011 1110 | Shift all bits in r6 one place to the right, bit 7 is held constant |
| ror r7 ; | 0000 1000 | 0000 0100 | Shift all bits in r7 one place to the right, bit 7,0 is determined by Carry flag. |
| rol r8 ; | 0000 1001 | 0001 0010 | Shift all bits in r8 one place to the right, bit 7, 0 is determined by Carry flag. |

**Two Register Direct Addressing Mode Test**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ASM code** | **register value before execution** | | **register value after execution** | **meaning of instruction** |
|  |  |
| add r0, r1 ; | 1111 1110 | 1111 1110 | 1111 1100 | add r0 and r1, store to r0 |
| sub r2, r3 ; | 0000 0010 | 0000 0011 | 0000 0001 | sub r2 and r3, store to r2 |
| and r4, r5 ; | 0000 0000 | 0000 0110 | 0000 0000 | operation and r4 with r5, store r4 |
| or r6, r7 ; | 0011 1110 | 0000 0100 | 0011 1110 | operation or r6 with r7, store r6 |
| eor r8, r9 ; | 0001 0010 | 0000 1010 | 0001 1000 | operation exclusive or r8 with r9, store r8 |
| cp r10, r11 ; | 0000 1011 | 0000 1100 | 0000 1011 | Compare r10 with r11, no change |
| mov r10, r11 ; | 0000 1011 | 0000 1100 | 0000 1100 | copy r11 and paste to r10 |

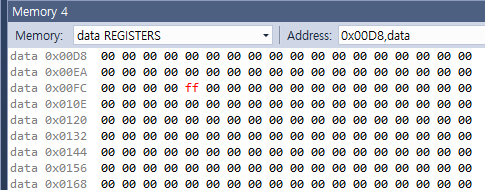
 

**Data Direct Addressing Mode Test**

먼저 ldi temp, $ff ; 를 통해 temp 변수에 해당하는 r16레지스터의 주소값 0xFF가 할당된다.

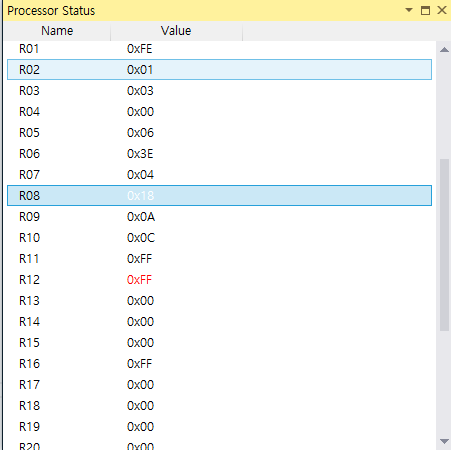
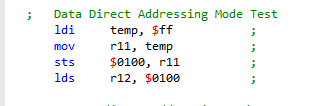
mov r11, temp ; r11레지스터에 주소값 0xFF가 복사된다.

sts $0100 r11 ; r11레지스터값을 데이터의 0100번지에 저장한다. 즉 0xFF가 저장된다.



lds r12, $0100 ; 데이터의 0100 번지 값을 r12레지스터에 저장한다. 즉 0xFF가 저장된다.

결론 : 메모리의 $0100에 0xFF 값이 저장되고 r12레지스터에도 0xFF가 저장된다.



**Data Indirect Addressing Mode Test**

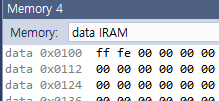
ldi temp, $fe ; temp 변수, 즉 r16레지스터에 주소값 0xFE를 저장한다.

mov r13, temp ; r13레지스터에 0xFE를 복사한다.

ldi XH, high($0101) ; X레지스터의 상위비트에 0x01을 저장한다.

ldi XL, low($0101) ; X레지스터의 하위비트에 0x01을 저장한다.

st X, r13 ; 메모리의 $0101에 저장되어 있는 값 0xFE를 r13레지스터에 저장한다.



ld r14, X ; 메모리 $0101에 저장되어 있는 값 0xFE를 r14레지스터에 저장한다.

결론: 메모리의 $0101에는 0xFE가 저장되어 있고 r14레지스터에 역시 0xFE가 저장된다.

