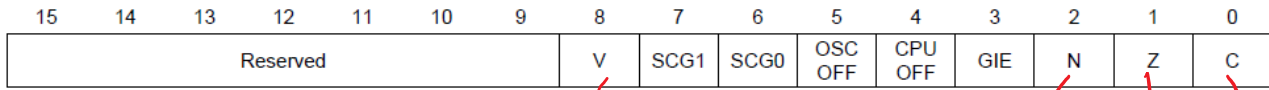


Status Register



Program Counter	PC/R0
Stack Pointer	SP/R1
Status Register	SR/CG1/R2
Constant Generator	CG2/R3
General-Purpose Register	R4
General-Purpose Register	R5
General-Purpose Register	R6
General-Purpose Register	R7
General-Purpose Register	R8
General-Purpose Register	R9
General-Purpose Register	R10
General-Purpose Register	R11
General-Purpose Register	R12
General-Purpose Register	R13
General-Purpose Register	R14
General-Purpose Register	R15

# Status Register (SR)



overflow

negative zero carry

V, N, Z, C bits can be set (=1) or (cleared) reset (=0) after an instruction.

## Z bit : (Zero bit)

Set when the result of a byte or word operation is 0 and cleared when the result is not 0.

```

mov.w #0x0003, R10
mov.w #0x0003, R11
sub.w R10, R11
    
```

status bits not affected

→ Z bit is set (=1) after this instruction

$$\text{sub.w } s, d \equiv d - s$$

```

mov.w #0x0003, R10
mov.w #0x0002, R11
sub.w R10, R11
    
```

cleared

→ Z bit is reset (=0) after this instruction

## N bit (Negative bit)

Set when the result of a byte or word operation is negative and cleared when the result is not negative.

Word operation: N is set to the value of bit 15 of the result.

Byte operation: N is set to the value of bit 7 of the result.

mov.w	#0x0004, R10		
mov.w	#0x0002, R11		
sub.w	R10, R11	$2 - 4 = -2 < 0$	; status bit N set
mov.w	#-4, R10		
mov.w	#-0x0008, R11	$-4 + (-8) = -12 < 0$	
add.w	R10, R11		; status bit N set
mov.w	#-4, R10		
mov.w	#0x0008, R11	$-4 + 8 = 4 > 0$	
add.w	R10, R11		; status bit N reset (cleared)
mov.w	#2, R10		
mov.w	#2, R11	$2 - 2 = 0$ (not $< 0$ )	
sub.w	R10, R11		; status bit N reset (cleared)

# C bit (Carry bit)

Set when the result of a byte or word operation produced a carry and cleared when no carry occurred.

```

11000001 = -63
+ 11000001 = -63
-----
110000010 = -126 carry (with no overflow)
    
```

```

mov.b #11000001b, R10
mov.b #11000001b, R11
add.b R10, R11
    
```

; status bit C set

```

10000001 = -127
10000001 = -127
-----
100000010 = 2 carry (with overflow)
    
```

```

mov.b#10000001b, R10
mov.b#10000001b, R11
add.b R10, R11
    
```

; status bit C set

```

11000000 = -64
- 11111110 = -(-2)
-----
    
```

```

=>      11000000 = -64
        + 00000010 = 2
        -----
        110000010 = -62 no carry (no overflow)
    
```

```

mov.b #11111110b, R10
mov.b #11000000b, R11
sub.b R10, R11
    
```

; R10 contains -2  
; R11 contains -64  
; status bit C reset

# V bit (Overflow Bit)

This bit is set when the result of an arithmetic operation overflows the signed-variable range.

ADD(.W, .B) Set when:

Positive + Positive = Negative } overflow  
 Negative + Negative = Positive } overflow  
 Otherwise reset → no overflow

SUB(.W, .B),CMP(.W, .B) Set when:

Positive - Negative = Negative } overflow  
 Negative - Positive = Positive } overflow  
 Otherwise reset → no overflow

11000001 = -63  
 + 11000001 = -63  
 -----

110000010 = -126 carry (with no overflow)

```
mov.b #11000001b, R10
mov.b #11000001b, R11
add.b R10, R11
```

; status bit V cleared

10000001 = -127  
 10000001 = -127  
 -----

100000010 = 2 carry (with overflow)

```
mov.b #10000001b, R10
mov.b #10000001b, R11
add.b R10, R11
```

; status bit V set