

- a. Write the Interrupt Vector Change Enable (IVCE) bit to one.
- b. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE.

Interrupts will automatically be disabled while this sequence is executed. Interrupts are disabled in the cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the Status Register is unaffected by the automatic disabling.

**Note:** If Interrupt Vectors are placed in the Boot Loader section and Boot Lock bit BLB02 is programmed, interrupts are disabled while executing from the Application section. If Interrupt Vectors are placed in the Application section and Boot Lock bit BLB12 is programmed, interrupts are disabled while executing from the Boot Loader section. Refer to the section ["Boot Loader Support – Read-While-Write Self-Programming, ATmega88P, ATmega168P and ATmega328P" on page 277](#) for details on Boot Lock bits.

This bit is not available in ATmega48P.

## • Bit 0 – IVCE: Interrupt Vector Change Enable

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts, as explained in the IVSEL description above. See Code Example below.

### Assembly Code Example

```
Move_interrupts:
    ; Enable change of Interrupt Vectors
    ldi r16, (1<<IVCE)
    out MCUCR, r16
    ; Move interrupts to Boot Flash section
    ldi r16, (1<<IVSEL)
    out MCUCR, r16
    ret
```

### C Code Example

```
void Move_interrupts(void)
{
    /* Enable change of Interrupt Vectors */
    MCUCR = (1<<IVCE);
    /* Move interrupts to Boot Flash section */
    MCUCR = (1<<IVSEL);
}
```

This bit is not available in ATmega48P.