Table 9-1. Reset and Interrupt Vectors in ATmega48P (Continued)

Vector No.	Program Address	Source	Interrupt Definition
24	0x017	ANALOG COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM READY	Store Program Memory Ready

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega48P is:

```
Address  Labels Code                 Comments
0x000           rjmp   RESET          ; Reset Handler
0x001           rjmp   EXT_INT0       ; IRQ0 Handler
0x002           rjmp   EXT_INT1       ; IRQ1 Handler
0x003           rjmp   PCINT0         ; PCINT0 Handler
0x004           rjmp   PCINT1         ; PCINT1 Handler
0x005           rjmp   PCINT2         ; PCINT2 Handler
0x006           rjmp   WDT            ; Watchdog Timer Handler
0x007           rjmp   TIM2_COMPA     ; Timer2 Compare A Handler
0x008           rjmp   TIM2_COMPB     ; Timer2 Compare B Handler
0x009           rjmp   TIM2_OVF       ; Timer2 Overflow Handler
0x00A           rjmp   TIM1_CAPT      ; Timer1 Capture Handler
0x00B           rjmp   TIM1_COMPA     ; Timer1 Compare A Handler
0x00C           rjmp   TIM1_COMPB     ; Timer1 Compare B Handler
0x00D           rjmp   TIM1_OVF       ; Timer1 Overflow Handler
0x00E           rjmp   TIM0_COMPA     ; Timer0 Compare A Handler
0x00F           rjmp   TIM0_COMPB     ; Timer0 Compare B Handler
0x010           rjmp   TIM0_OVF       ; Timer0 Overflow Handler
0x011           rjmp   SPI_STC        ; SPI Transfer Complete Handler
0x012           rjmp   USART_RXC      ; USART, RX Complete Handler
0x013           rjmp   USART_UDRE     ; USART, UDR Empty Handler
0x014           rjmp   USART_TXC      ; USART, TX Complete Handler
0x015           rjmp   ADC            ; ADC Conversion Complete Handler
0x016           rjmp   EE_RDY         ; EEPROM Ready Handler
0x017           rjmp   ANA_COMP       ; Analog Comparator Handler
0x018           rjmp   TWI            ; 2-wire Serial Interface Handler
0x019           rjmp   SPM_RDY        ; Store Program Memory Ready Handler
;
0x01ARESET:     ldi    r16, high(RAMEND); Main program start
0x01B           out    SPH,r16        ; Set Stack Pointer to top of RAM
0x01C           ldi    r16, low(RAMEND)
0x01D           out    SPL,r16
0x01E           sei                   ; Enable interrupts
0x01F           <instr>  xxx
  ...     ...     ...  ...
```