

Microcontroladores

Prof. Marcos Chaves

Interrupções Externas

Fonte:

PICmicro Microcontroladores - José Carlos Fonseca
Desbravando o PIC Ampliado e Atualizado – David José de Souza

Interrupções

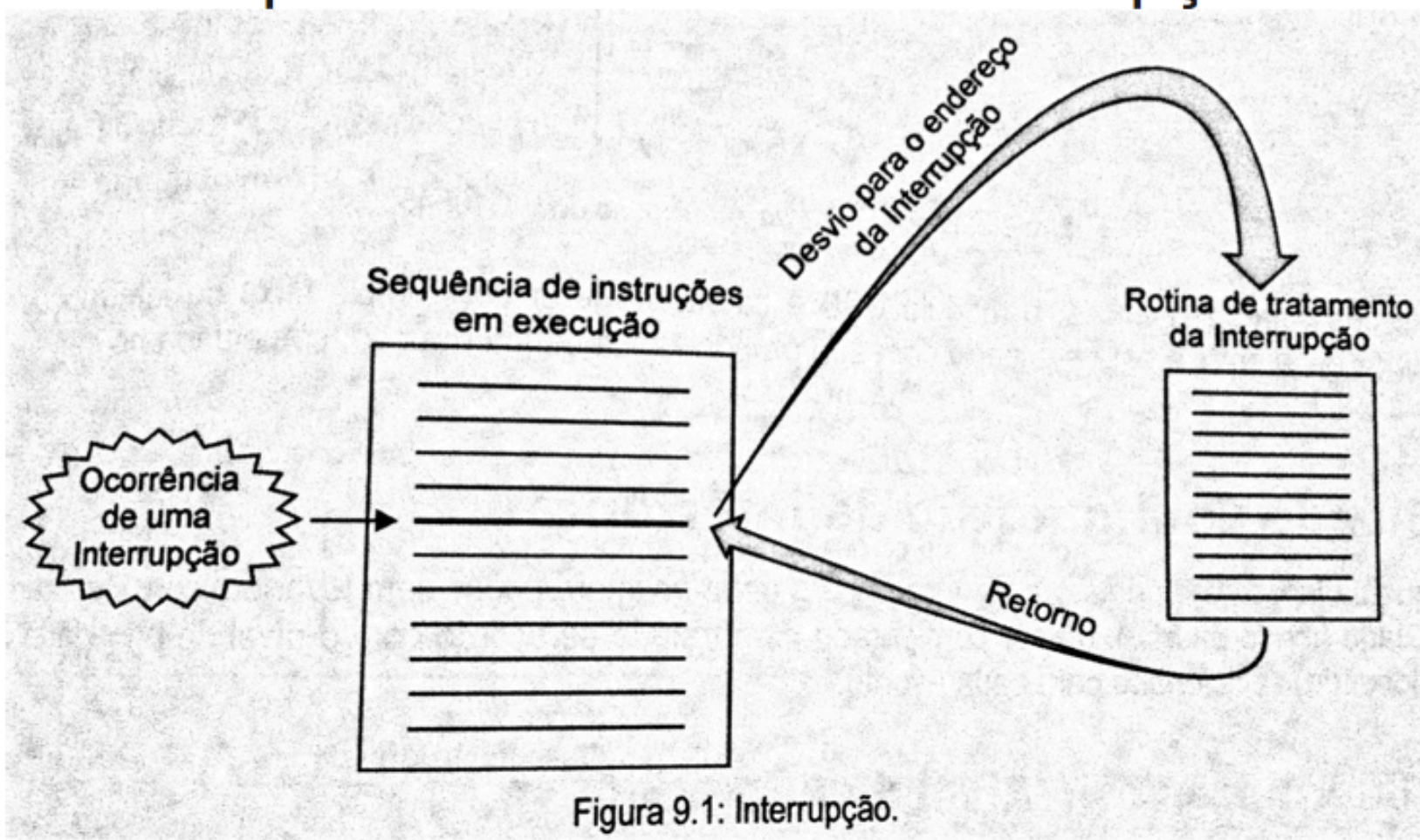
As interrupções são causadas através de eventos assíncronos (podem ocorrer a qualquer momento) causando um desvio no processamento. Este desvio tem como destino um endereço para tratamento da interrupção. Uma boa analogia para melhor entendermos o conceito de

interrupção é a seguinte: você está trabalhando digitando uma carta no computador quando o seu telefone toca. Neste momento você, interrompe o que está fazendo, para atender ao telefone e verificar o que a pessoa do outro lado da linha está precisando. Terminada a conversa, você coloca o telefone no gancho novamente e retoma o seu trabalho do ponto onde havia parado.

Observe que não precisamos verificar a todo instante, se existe ou não alguém na linha, pois somente quando o ramal é chamado, o telefone toca avisando que existe alguém querendo falar com você.

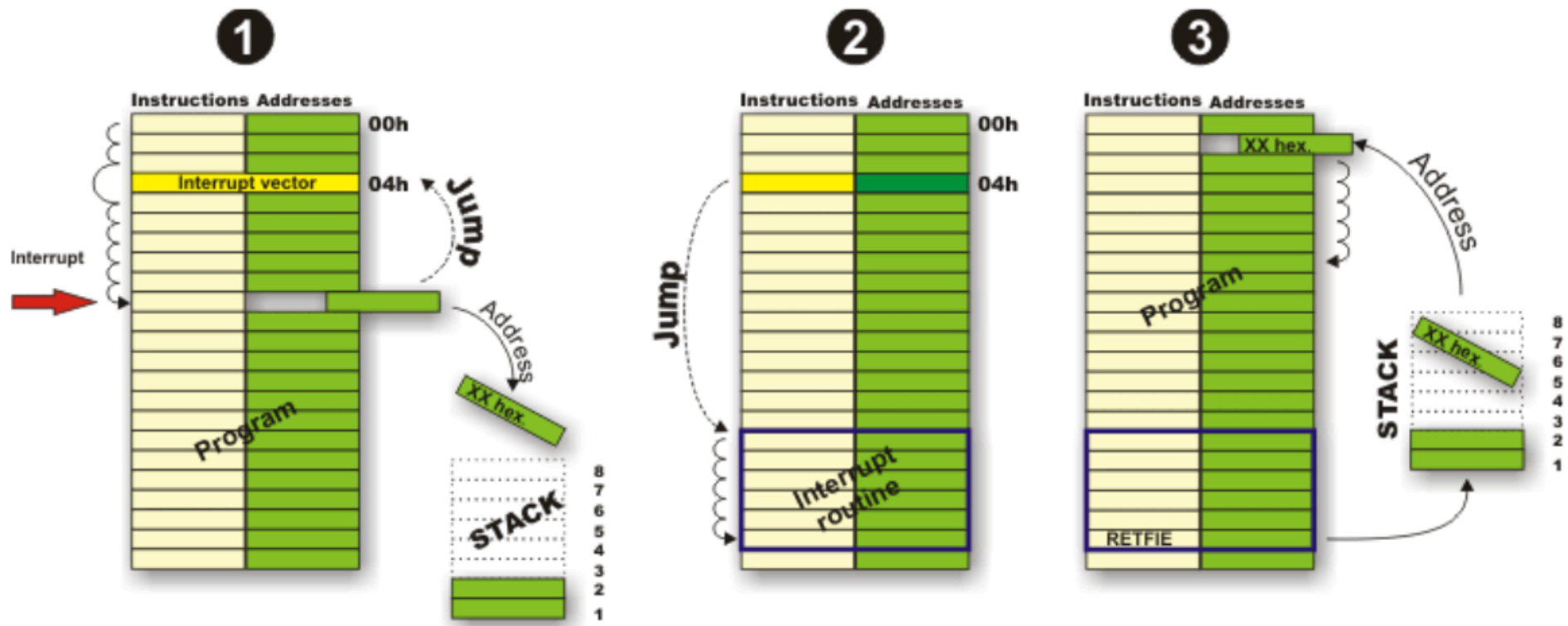


Comportamento de uma interrupção



Uma interrupção suspende a execução de uma aplicação, salva as informações de contexto e desvia o controle para uma rotina de serviço de interrupção (ISR) para que o evento possa ser processado. Ao finalizar as informações são restauradas e a execução normal é retomada.

Fases de atendimento de uma interrupção



1 – Quando a interrupção ocorre, o programa principal é interrompido; O endereço atual é salvo na pilha (STACK); O contador de programa(PC) é carregado com o endereço do vetor de interrupção específica ativada.

2 – No endereço do vetor interrupção deve ocorrer a chamada da rotina de tratamento da interrupção (Interrupt routine).

3 – Ao final da rotina de interrupção, o PC é carregado com o último endereço salvo na pilha, retornando ao ponto original do programa principal.

Vetor	Endereço	Fonte	Definição
1	0x0000	RESET	Reset Ext., Power-On Reset,
2	0x0002	INT0	Int. Externa no pino INT0
3	0x0004	INT1	Int. Externa no pino INT1
4	0x0006	PCINT0	Mudança estado pino Req. 0
5	0x0008	PCINT1	Mudança estado pino Req. 1
6	0x000A	PCINT2	Mudança estado pino Req. 2
7	0x000C	WDT	Watchdog Time-out
8	0x000E	TIMER2 COMPA	T/C2 Compare Match A
9	0x0010	TIMER2 COMPB	T/C2 Compare Match B
10	0x0012	TIMER2 OVF	T/C2 Overflow
11	0x0014	TIMER1 CAPT	T/C1 Evento de Captura
12	0x0016	TIMER1 COMPA	T/C1 Compare Match A
13	0x0018	TIMER1 COMPB	T/C1 Compare Match B
14	0x001A	TIMER1 OVF	T/C1 Overflow
15	0x001C	TIMER0 COMPA	T/C0 Compare Match A
16	0x001E	TIMER0 COMPB	T/C0 Compare Match B
17	0x0020	TIMER0 OVF	T/C0 Overflow
18	0x0022	SPI, STC	SPI Transfer. completa
19	0x0024	USART, RX	USART Recep. completa
20	0x0026	USART, UDRE	USART Reg. dados vazio
21	0x0028	USART, TX	USART Transm. completa
22	0x002A	ADC	Conv. A/D completada
23	0x002C	EE_READY	EEPROM Pronta
24	0x002E	ANALOG COMP	Comparador Analógico
25	0x0030	TWI	2-Wire serial interface
26	0x0032	SPM READY	Memória Flash Pronta

Interrupções no Atmega328

Vetores de Interrupções no Atmega328 em C

Código (função)

```
int main()                //aqui vai o programa principal                }
ISR(INT0_vect)            //interrupção externa 0                        }
ISR(INT1_vect)            //interrupção externa 1                        }
ISR(PCINT0_vect)          //interrupção 0 por mudança de pino           }
ISR(PCINT1_vect)          //interrupção 1 por mudança de pino           }
ISR(PCINT2_vect)          //interrupção 2 por mudança de pino           }
ISR(WDT_vect)             //estouro do temporizador Watchdog            }
ISR(TIMER2_COMPA_vect)    //igualdade de comparação A do TC2            }
ISR(TIMER2_COMPB_vect)    //igualdade de comparação B do TC2            }
ISR(TIMER2_OVF_vect)      //estouro do TC2}                               }
ISR(TIMER1_CAPT_vect)     //evento de captura do TC1                    }
ISR(TIMER1_COMPA_vect)    //igualdade de comparação A do TC1            }
ISR(TIMER1_COMPB_vect)    //igualdade de comparação B do TC1            }
ISR(TIMER1_OVF_vect)      //estouro do TC1                               }
ISR(TIMER0_COMPA_vect)    //igualdade de comparação A do TC0            }
ISR(TIMER0_COMPB_vect)    //igualdade de comparação B do TC0            }
ISR(TIMER0_OVF_vect)      //estouro do TC0                               }
ISR(SPI_STC_vect)         //transferência serial completa - SPI         }
ISR(USART_RX_vect)        //USART, recepção completa                    }
ISR(USART_UDRE_vect)      //USART, limpeza do registrador de dados      }
ISR(USART_TX_vect)        //USART, transmissão completa                 }
ISR(ADC_vect)             //conversão do ADC completa                   }
ISR(EE_READY_vect)        //EEPROM pronta                                }
ISR(ANALOG_COMP_vect)     //comparador analógico                        }
ISR(TWI_vect)             //interface serial TWI                        }
ISR(SPM_READY_vect)       //armazenagem na memória de programa pronta}
```

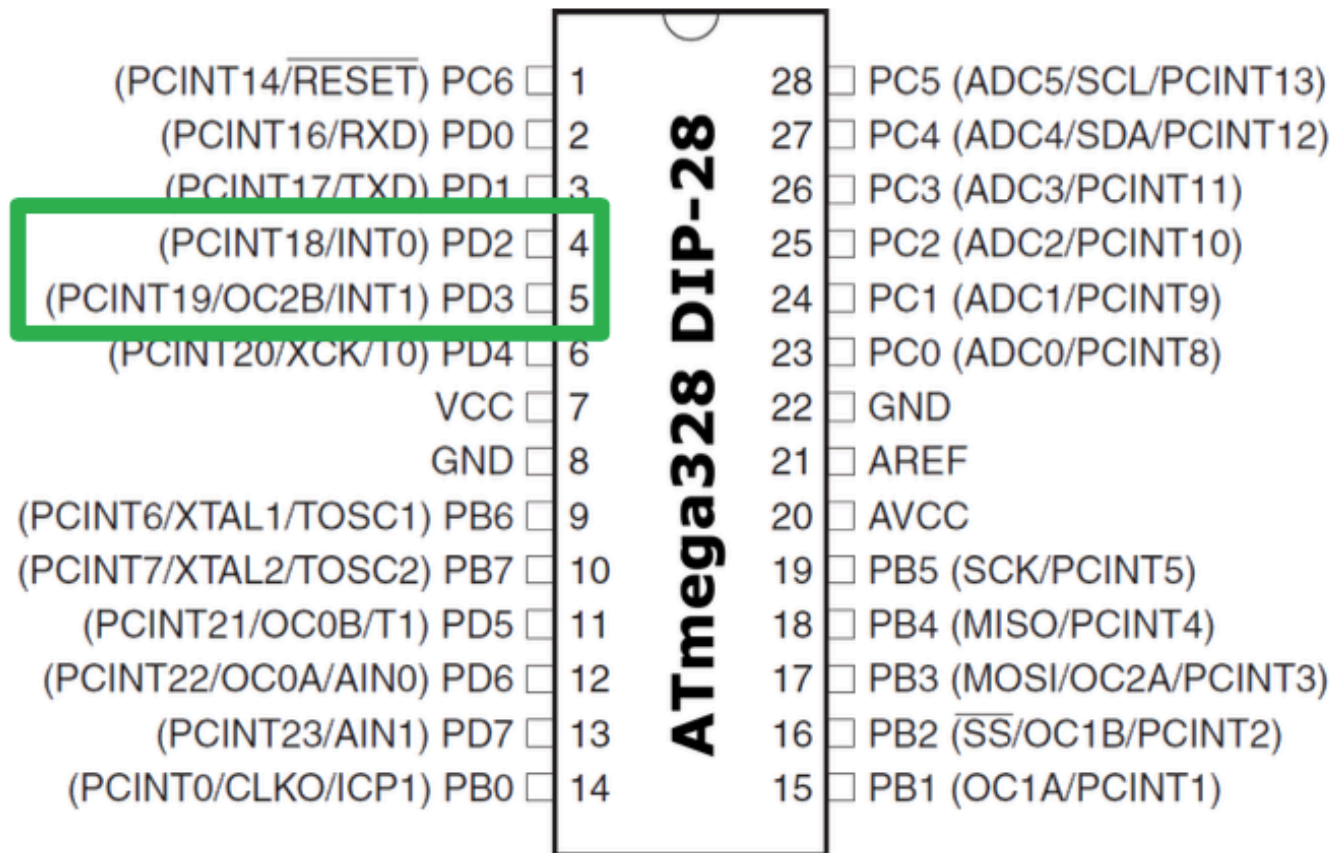
Vetores de Interrupções no Atmega328 em Assembly

Address	Labels	Code	Comments
0x0000		jmp RESET	; Reset Handler
0x0002		jmp EXT_INT0	; IRQ0 Handler
0x0004		jmp EXT_INT1	; IRQ1 Handler
0x0006		jmp PCINT0	; PCINT0 Handler
0x0008		jmp PCINT1	; PCINT1 Handler
0x000A		jmp PCINT2	; PCINT2 Handler
0x000C		jmp WDT	; Watchdog Timer Handler
0x000E		jmp TIM2_COMPA	; Timer2 Compare A Handler
0x0010		jmp TIM2_COMPB	; Timer2 Compare B Handler
0x0012		jmp TIM2_OVF	; Timer2 Overflow Handler
0x0014		jmp TIM1_CAPT	; Timer1 Capture Handler
0x0016		jmp TIM1_COMPA	; Timer1 Compare A Handler
0x0018		jmp TIM1_COMPB	; Timer1 Compare B Handler
0x001A		jmp TIM1_OVF	; Timer1 Overflow Handler
0x001C		jmp TIM0_COMPA	; Timer0 Compare A Handler
0x001E		jmp TIM0_COMPB	; Timer0 Compare B Handler
0x0020		jmp TIM0_OVF	; Timer0 Overflow Handler

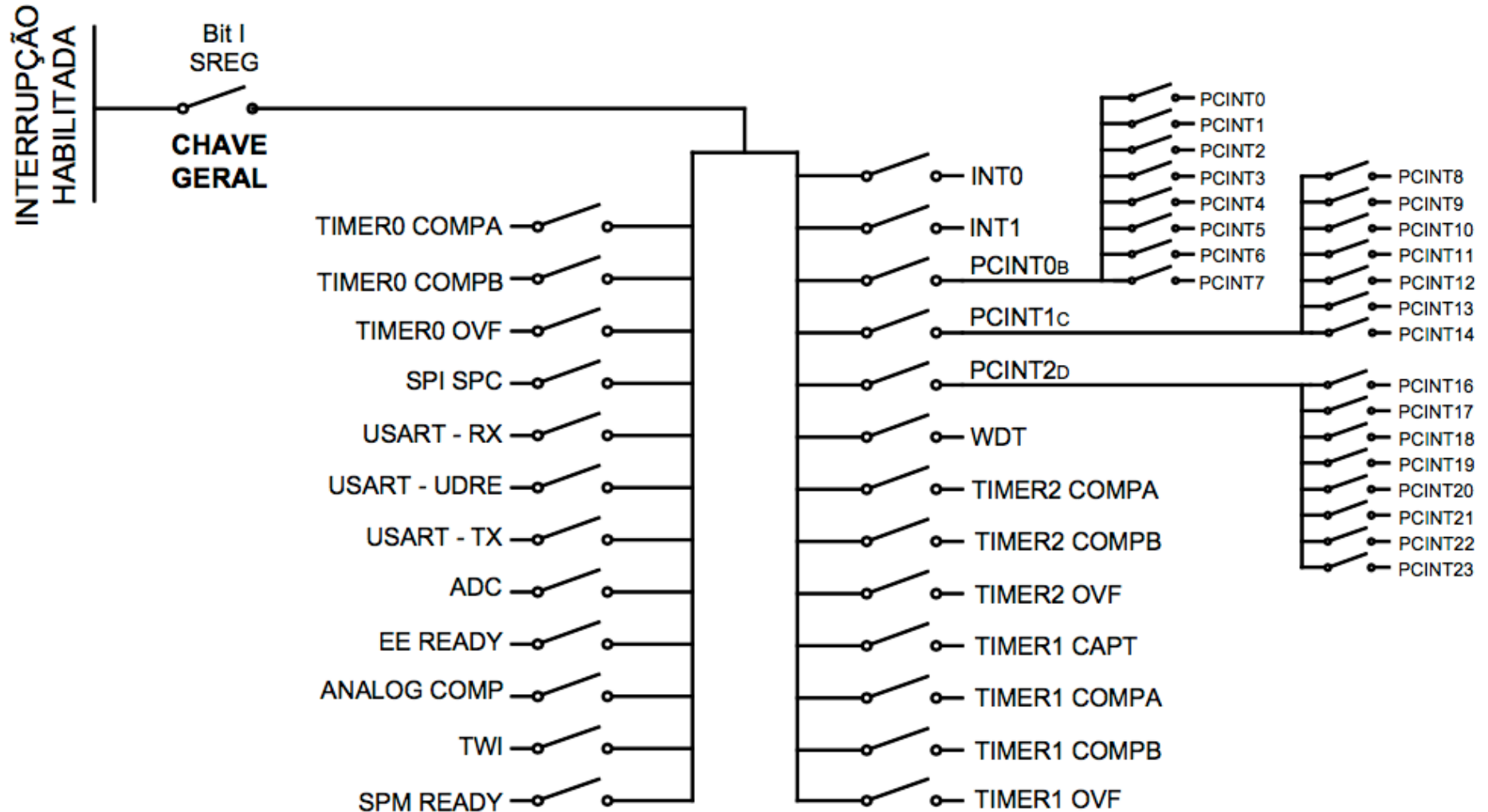
Vetores de Interrupções no Atmega328 em Assembly

```
0x0022      jmp     SPI_STC           ; SPI Transfer Complete Handler
0x0024      jmp     USART_RXC        ; USART, RX Complete Handler
0x0026      jmp     USART_UDRE       ; USART, UDR Empty Handler
0x0028      jmp     USART_TXC        ; USART, TX Complete Handler
0x002A      jmp     ADC              ; ADC Conversion Complete Handler
0x002C      jmp     EE_RDY           ; EEPROM Ready Handler
0x002E      jmp     ANA_COMP         ; Analog Comparator Handler
0x0030      jmp     TWI              ; 2-wire Serial Interface Handler
0x0032      jmp     SPM_RDY          ; Store Program Memory Ready Handler
;
0x0033RESET: ldi      r16, high(RAMEND); Main program start
0x0034      out     SPH,r16          ; Set Stack Pointer to top of RAM
0x0035      ldi      r16, low(RAMEND)
0x0036      out     SPL,r16
0x0037      sei                          ; Enable interrupts
0x0038      <instr>  xxx
...      ...      ...      ...
```

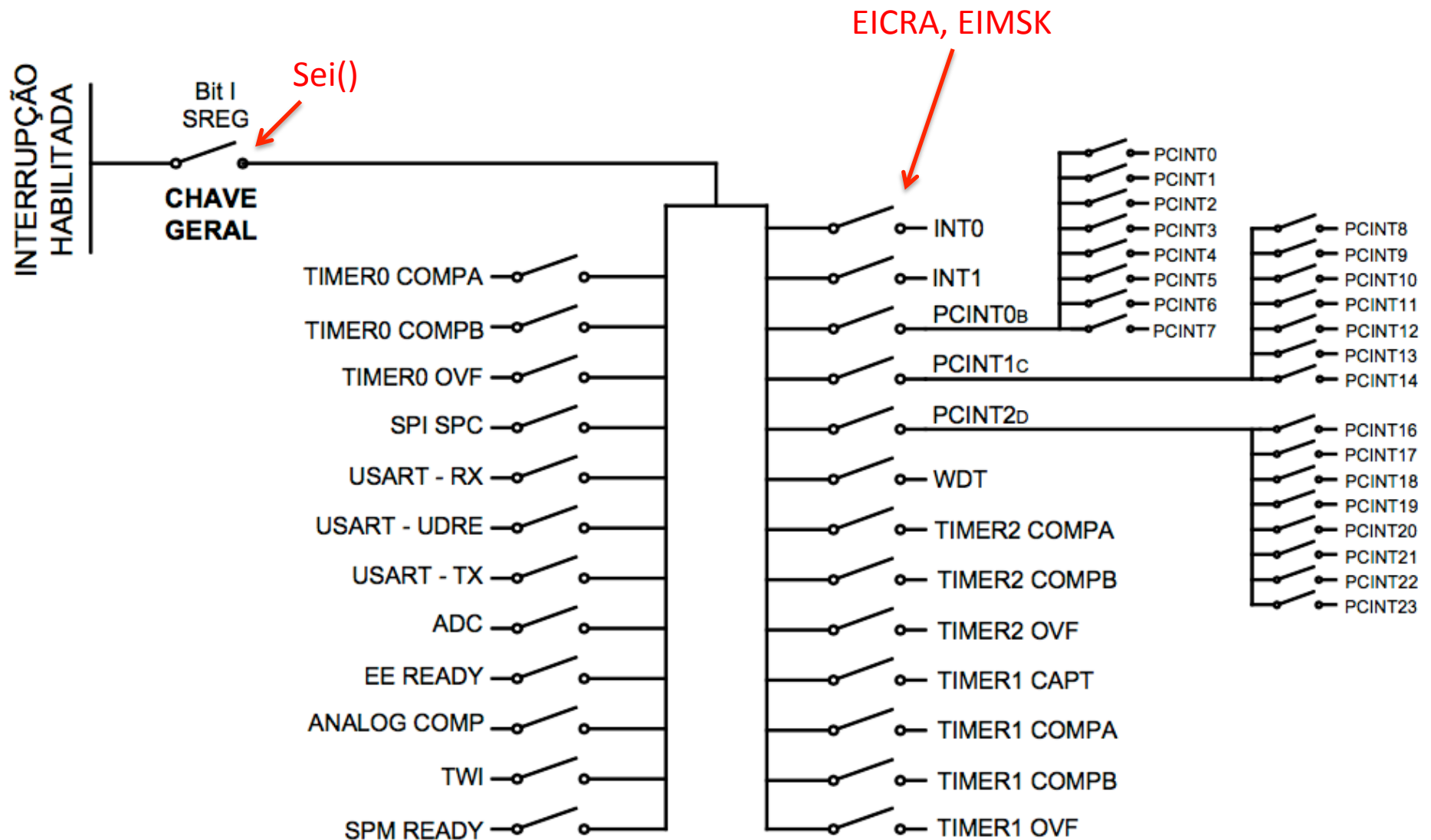

- O ATmega328 possui dois pinos dedicados para interrupções externas:
 - INT0 e INT1



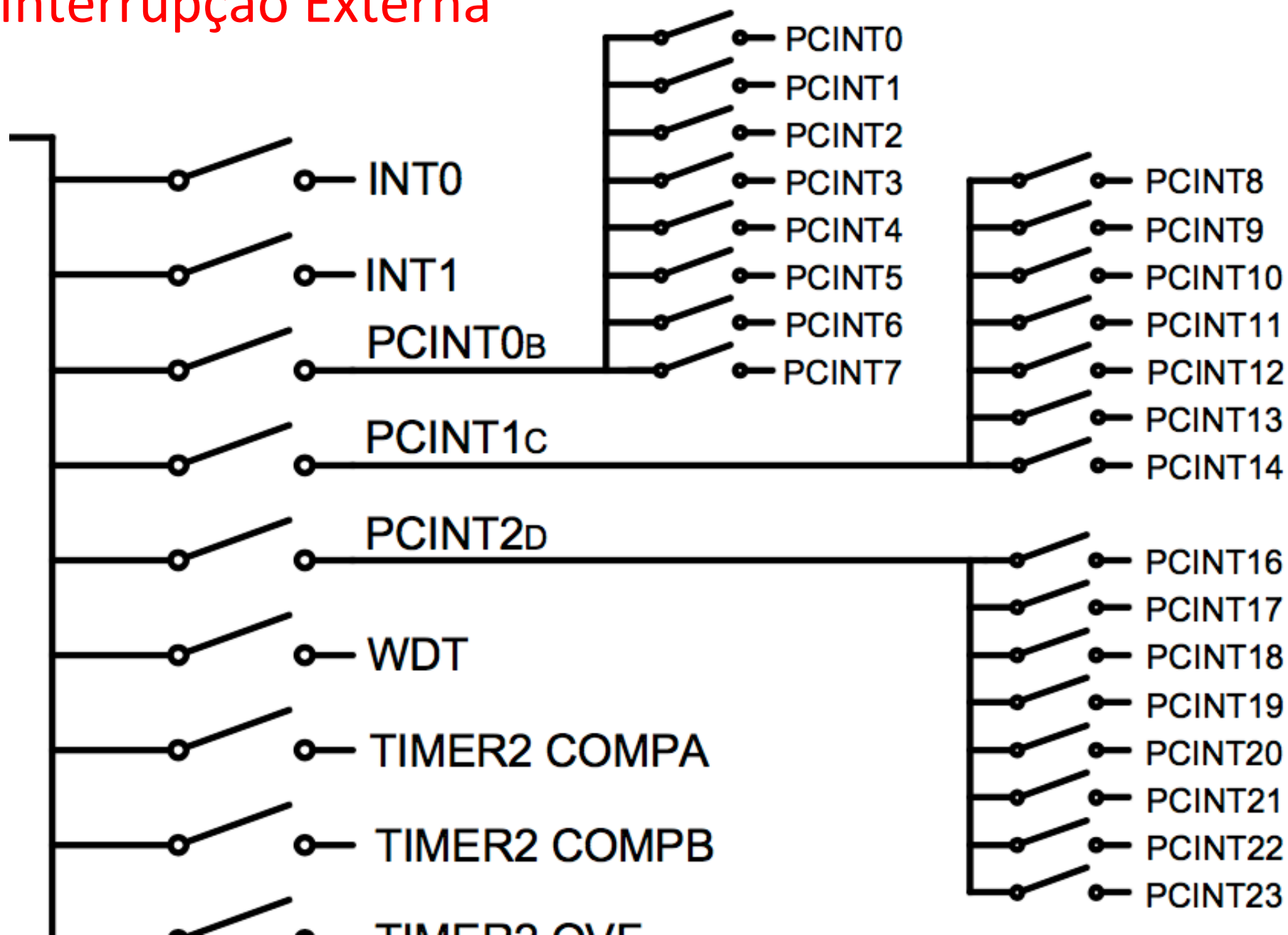
Interrupção Externa



Interrupção Externa



Interrupção Externa

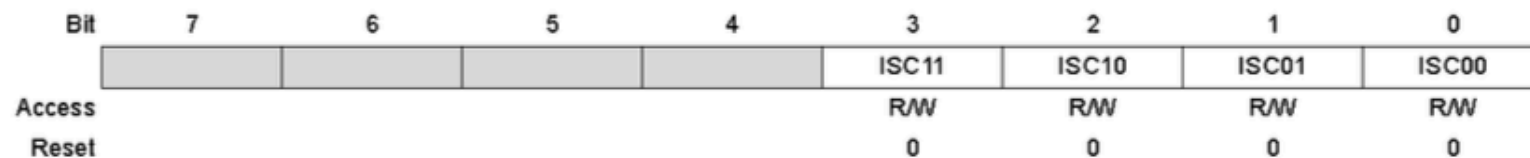


Registradores Interrupção Externa

- ✓ Há 5 interrupções externas:
 - INT1 e INT0;
 - PCI2, PCI1 e PCIO.
- ✓ As interrupções externas são gatilhadas a partir dos pinos INT[1:0] (INT1 e INT0), PCINT[23:16] (PCI2), PCINT[14:8] (PCI1) e PCINT[7:0] (PCIO).

Registadores Interrupção Externa EICRA

Name: EICRA
Offset: 0x69
Reset: 0x00
Property: -



- ✓ **Bits 1 e 0 - ISC01 e ISC00** → Indica o evento (nível e borda) no pino INT0 que gerará uma interrupção.

Value	Description
00	The low level of INT0 generates an interrupt request.
01	Any logical change on INT0 generates an interrupt request.
10	The falling edge of INT0 generates an interrupt request.
11	The rising edge of INT0 generates an interrupt request.

'0' por pelo menos
a execução da
instrução atual.

> 1 ciclo
de clock

Tab. 6.2 – Bits de configuração da forma das interrupções nos pinos INT1 e INT0.

ISC11	ISC10	Descrição
0	0	Um nível baixo em INT1 gera um pedido de interrupção.
0	1	Qualquer mudança lógica em INT1 gera um pedido de interrupção.
1	0	Uma borda de decida em INT1 gera um pedido de interrupção.
1	1	Uma borda de subida em INT1 gera um pedido de interrupção.
ISC01	ISC00	Descrição
0	0	Um nível baixo em INT0 gera um pedido de interrupção.
0	1	Qualquer mudança lógica em INT0 gera um pedido de interrupção.
1	0	Uma borda de decida em INT0 gera um pedido de interrupção.
1	1	Uma borda de subida em INT0 gera um pedido de interrupção.

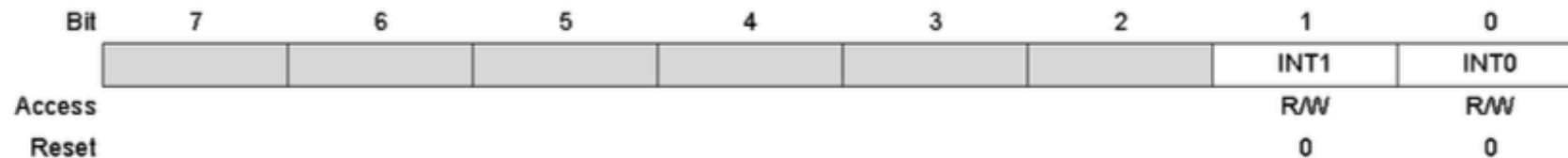
Registadores Interrupção Externa EIMSK

Name: EIMSK

Offset: 0x3D

Reset: 0x00

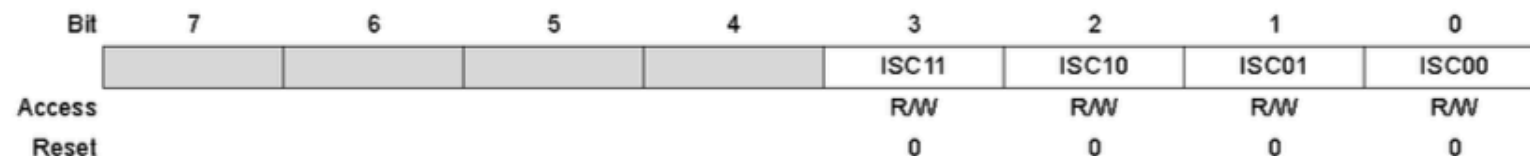
Property: When addressing as I/O Register: address offset is 0x1D



- ✓ **Bits 1 e 0 - INT1 e INT0** → Habilita, respectivamente, as interrupções INT1 e INT0. Se I = 1, uma interrupção ocorrerá quando um determinado evento ocorrer no respectivo pino.
- ✓ **Nota:** mesmo se o pino for configurado como saída, uma requisição de interrupção poderá ocorrer se uma atividade (evento configurado em EICRA) ocorrer.

Registadores Interrupção Externa EICRA

Name: EICRA
Offset: 0x69
Reset: 0x00
Property: -



- ✓ **Bits 3 e 2 - ISC11 e ISC10** → Indica o evento (nível e borda) no pino INT1 que gerará uma interrupção.

Value	Description
00	The low level of INT1 generates an interrupt request.
01	Any logical change on INT1 generates an interrupt request.
10	The falling edge of INT1 generates an interrupt request.
11	The rising edge of INT1 generates an interrupt request.

'0' por pelo menos
a execução da
instrução atual.

> 1 ciclo
de clock

Registadores Interrupção Externa PCICR

Name: PCICR
Offset: 0x68
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						PCIE2	PCIE1	PCIE0
Access						R/W	R/W	R/W
Reset						0	0	0

- ✓ **Bit 2 - PCIE2** → Habilita interrupção de mudança no pino. Qualquer mudança em qualquer um dos pinos (PCINT[23:16]) habilitados causará uma interrupção (PCI2).
 - ✓ **Bit 1 - PCIE1** → idem para os pinos PCINT[14:8] e interrupção PCI1.
 - ✓ **Bit 0 - PCIE0** → idem para os pinos PCINT[7:0] e interrupção PCI0.
-

Registadores Interrupção Externa PCIFR

Name: PCIFR

Offset: 0x3B

Reset: 0x00

Property: When addressing as I/O Register: address offset is 0x1B

Bit	7	6	5	4	3	2	1	0
						PCIF2	PCIF1	PCIF0
Access						R/W	R/W	R/W
Reset						0	0	0

- ✓ **Bit 2 - PCIF2** → Indica a ocorrência da mudança de sinal em um dos pinos PCINT[23:16].
- ✓ **Bit 1 - PCIF1** → Indica a ocorrência da mudança de sinal em um dos pinos PCINT[14:8].
- ✓ **Bit 0 - PCIF0** → Indica a ocorrência da mudança de sinal em um dos pinos PCINT[7:0].
 - Esses flags são zerados automaticamente quando a ISR for executada ou, alternativamente, via software escrevendo '1'.

Registadores Interrupção Externa PCMSK2

Name: PCMSK2
Offset: 0x6D
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

- ✓ Seleciona se a interrupção de mudança no pino está habilitada no correspondente pino de E/S - PCI2.
- ✓ Ex.: se PCINT16 = 1, qualquer mudança nesse pino (1→0 ou 0→1), gerará uma interrupção se PCIE2 de PCICR for '1' e I=1. O bit PCIF2 de PCIFR será setado.

Registadores Interrupção Externa PCMSK1

Name: PCMSK1

Offset: 0x6C

Reset: 0x00

Property: -

Bit	7	6	5	4	3	2	1	0
		PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

- ✓ Seleciona se a interrupção de mudança no pino está habilitada no correspondente pino de E/S - PCI1.
- ✓ Ex.: se PCINT14 = 1, qualquer mudança nesse pino (1→0 ou 0→1), gerará uma interrupção se PCIE1 de PCICR for '1' e I=1. O bit PCIF1 de PCIFR será setado.

Registadores Interrupção Externa PCMSK0

Name: PCMSK0
Offset: 0x6B
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

- ✓ Seleciona se a interrupção de mudança no pino está habilitada no correspondente pino de E/S - PCIO.
- ✓ Ex.: se PCINT3 = 1, qualquer mudança nesse pino (1→0 ou 0→1), gerará uma interrupção se PCIE0 de PCICR for '1' e I=1. O bit PCIF0 de PCIFR será setado.

Exemplo de uma operação das duas interrupções externas, INT0 e INT1

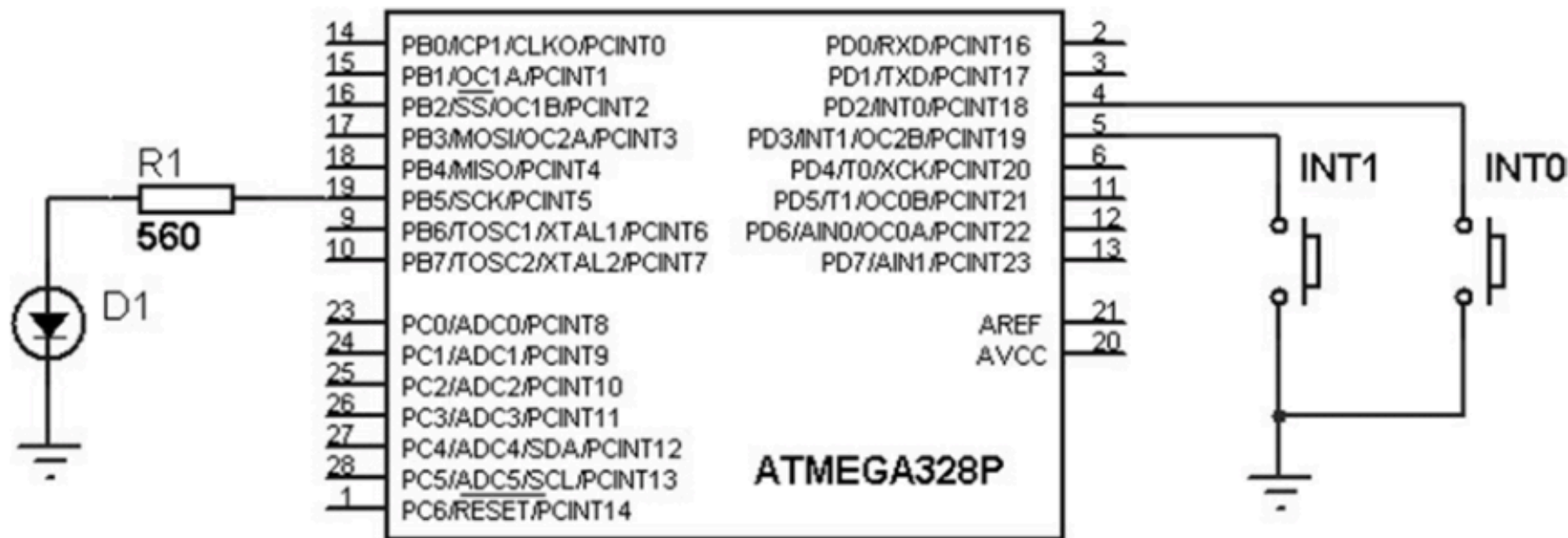


Fig. 6.2 – Circuito para emprego das interrupções externas INT0 e INT1.

```
=====
; DEFINITIONS
```

```
;=====
```

```
.equ vermelho = pb0
```

```
.equ amarelo = pb1
```

```
.equ verde = pb2
```

```
;=====
```

```
; VARIABLES
```

```
;=====
```

```
;=====
```

```
; RESET and INTERRUPT VECTORS
```

```
;=====
```

```
.org 0x0000
```

```
    ; Reset Vector
```

```
    rjmp Start
```

```
.org 0x0002 ; porta que pode receber o bot,,o deinterrupÁ,,o
```

```
    rjmp trata_int0
```

```
.org 0x0004 ; porta que pode receber o bot,,o deinterrupÁ,,o
```

```
    rjmp trata_int1
```

```
.org 0x050
```



```
.include "lib328Pv01.inc"
```

```
Start:
```

```
    ; Write your code here
```

```
    sbi ddrb,0
```

```
    sbi ddrb,1
```

```
    sbi ddrb,2
```

```
    sei ; habilita as interrupções no microcontrolador
```

```
    sbi EIMSK,INT0 ; habilita a interrupção externa 0
```

```
    sbi EIMSK,INT1 ; habilita a interrupção externa 0
```

```
Loop:
```

```
    sbi portb,vermelho
```

```
    ldi delay_time,1
```

```
    rcall delay_seconds
```

```
    cbi portb,vermelho
```

```
    sbi portb,verde
```

```
    ldi delay_time,1
```

```
    rcall delay_seconds
```

```
    cbi portb,verde
```

```
    sbi portb,amarelo
```

```
    ldi delay_time,1
```

```
    rcall delay_seconds
```

```
    cbi portb,amarelo
```

```
    rjmp Loop
```

;=====

trata_int0:

 cbi portb,vermelho

 cbi portb,verde

blink: ; pisca o amarelo

 sbi portb,amarelo

 ldi delay_time,1

 rcall delay_seconds

 cbi portb,amarelo

 ldi delay_time,1

 rcall delay_seconds

 rjmp blink

 ;reti

trata_int1:

 cbi portb,amarelo

 cbi portb,verde

 sbi portb,vermelho

 sei

 rjmp trata_int1