Přerušení

- Důvod – obsluha asynchronních událostí (CPU mnohem rychlejší než pomalé periferie – „má klávesnice nějaké znaky?“)
- Zdroje přerušení – interrupt source
  - HW – periferie (UART, Disk, časovače apod.)
  - SW (někdy – MS-DOS, multiprocesorové systémy)
- Realizace – CPU při vzniku přerušení skočí na předem danou adresu (podprogram)
  - Vznik přerušení („klávesnice má znak ke zpracování“)
  - CPU – dokončení právě vykonávané instrukce
  - CPU – vykonání podprogramu pro obsluhu přerušení („vyzvedne znak z klávesnice a uloží jej do paměti“) – asynchonnost
  - návrat na původní místo v programu a jeho pokračování
- lze globálně povolit/zakázat (všechny zdroje)
  - obvykle jeden bit ((global) interrupt enable – EA)

Tabulka vektorů přerušení

- interrupt vector table
- CPU při vzniku přerušení skáče na pevně dané místo v paměti (obvykle začátek CODE seg.)

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	Hardware Pin, Power-on Reset and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT1	Timer/Counter1 Capture Event
5	$004	TIMER1 COMP1	Timer/Counter1 Compare Match
6	$005	TIMER1 OVF1	Timer/Counter1 Overflow
7	$006	TIMER0 OVF0	Timer/Counter0 Overflow
8	$007	UART, RX	UART, RX Complete
9	$008	UART, UDRE	UART Data Register Empty
10	$009	UART, TX	UART, TX Complete
11	$00A	ANA_COMP	Analog Comparator

(AT90S2313)

- všechny vektory – blízko sebe (nelze podprogram)
- Na místa odskokových adres – jmp na podprogram pro obsluhu přerušení

```
      jmp RESET          ; Reset
      jmp Vektor1
      jmp Vektor2
            |
      jmp VektorN
; „hlavni“ program
RESET:
```

```
        <instrukce>
        ; atd.
; podprogram pro obsluhu přerušeni od vektoru1
Vektor1:
        <instrukce>
        ; atd.
```

- program přerušení nevyužívá (EA off) – lze vypustit

Typy přerušení

- maskovatelná (maskable interrupt)
  - lze vymaskovat (zakázat/povolit)
  - v interrupt mask register – kromě EA každý bit jedno přerušení (0/1 – on/off INT)
- nemaskovatelná (non-maskable interrupt)
  - nelze zakázat – pro řešení krit. stavů a chyb (zejména HW – chyby paměti apod)

Podprogram pro obsluhu přerušení

- Interrupt service routine, interrupt handler
- vykoná činnost pro dané přerušení
- požadavky na kód
  - volán asynchronně – nutné důsledné uschování/obnova používaných registrů
  - musí být zakončen instrukcí pro návrat z přerušení (reti, rti, iret) – = ret + obnova stavu logiky přerušení (na různých uP různé)

```
TIM_OVF1:
        push SREG
        push r0
        <instrukce>
        pop r0
        pop SREG
        reti
```

Priorita přerušení

- = co dělat, když:
  - vznik více přerušení současně
  - přerušení právě probíhajícího přerušení
- dána:
  - pevně (např. nižší vektory = vyšší priorita)
  - programovatelně – dle typu uP

Přerušení na AVR

- dle typu až 34 zdrojů přerušení

Nastavení přerušení

- registru SREG → bit I = Globální povolení (či zákaz) přerušení
  - I = 1 ⇒ přerušení povolena (instrukce SEI)
  - I = 0 ⇒ přerušení zakázána (instrukce CLI)

Bit	7	6	5	4	3	2	1	0	
	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- ATmega mají možnost tzv. bootloaderu → nutnost přesměrovat tabulku vektorů přerušení → registr GICR (nebudeme probírat, podrobnosti viz. datasheet)
- Každá periferie → přerušení ve svých registrech
  - Příklad: čítače/časovače → registr TIMSK

Bit	7	6	5	4	3	2	1	0	
	OCIE2	TOIE2	TICIE1	OCIE1A	OCIE1B	TOIE1	OCIE0	TOIE0	TIMSK
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

TOIEx → T/C overflow (přerušení přetečení)

OCIEx → T/C přerušení v režimu compare (nebudeme probírat)

- Mechanismus přerušení
  - Žádost o přerušení
  - Pokud povoleno globálně + vektor → PC = PC + x (x dle následující instrukce)
  - PC → STACK
  - Adresa vektoru → PC
  - **Zakáže se globálně přerušení (I = 0) !**
  - Vektor v PC = vykonání podprogramu, konec = RETI
  - **Povolí se globálně přerušení (I = 1) !**
  - Obnova PC ze STACK = pokračování programu
- priorita přerušení

- je určena pořadím – nejvyšší má RESET.

  Tabulka vektorů přerušení u ATmega32

Vector No.	Program Address(2)	Source	Interrupt Definition
1	$000(1)	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
2	$002	INT0	External Interrupt Request 0
3	$004	INT1	External Interrupt Request 1
4	$006	INT2	External Interrupt Request 2
5	$008	TIMER2 COMP	Timer/Counter2 Compare Match
6	$00A	TIMER2 OVF	Timer/Counter2 Overflow
7	$00C	TIMER1 CAPT	Timer/Counter1 Capture Event
8	$00E	TIMER1 COMPA	Timer/Counter1 Compare Match A
9	$010	TIMER1 COMPB	Timer/Counter1 Compare Match B
10	$012	TIMER1 OVF	Timer/Counter1 Overflow
11	$014	TIMER0 COMP	Timer/Counter0 Compare Match
12	$016	TIMER0 OVF	Timer/Counter0 Overflow
13	$018	SPI, STC	Serial Transfer Complete
14	$01A	USART, RXC	USART, Rx Complete
15	$01C	USART, UDRE	USART Data Register Empty
16	$01E	USART, TXC	USART, Tx Complete
17	$020	ADC	ADC Conversion Complete
18	$022	EE_RDY	EEPROM Ready
19	$024	ANA_COMP	Analog Comparator
20	$026	TWI	Two-wire Serial Interface
21	$028	SPM_RDY	Store Program Memory Ready

- Sejde-li se naráz více přerušení, je vykonáno to s největší prioritou. Další přerušení se vykonají postupně po vykonání předešlých dle priority.
- Aktuálně vykonávané přerušení nemůže být přerušeno již žádným jiným, a to ani přerušením s vyšší prioritou ! (Velký rozdíl oproti 8051 !). Důvod → CLI po vstupu do přerušovací rutiny.
- Pokud po vstupu do rutiny SEI a dojde-li během vykonávání aktuálního přerušení k požadavku na další přerušení → přerušení aktuálního přerušení bez ohledu na prioritu

Příklad programu

Přerušení od T1 každou 1s při f_{clk} = 4MHz

```
; 16b T/C pretece za 16,384 ms
; s delickou /64 je perioda 1,048576 -> nutno odecist 3036 period osc

.INCLUDE "m32def.inc"

; definice konstant
.EQU  pin    = PORTD0
.EQU  DDRport = DDRD
.EQU  OUTport      = PORTD
.EQU  INport       = PIND

.EQU  PRESET       = 3036
```

```
; definice aliasu
.DEF  temp  = r16   ;pomocny
.DEF  temp1 = r17   ;pomocny
.DEF  T1L   = r18
.DEF  T1H   = r19
.DEF  stav  = r0

.CSEG
      .ORG  $000
      rjmp  main        ; RESET
      .ORG  OVF1addr
      rjmp  T1_ovf            ; preteceni T1

; hlavni program
main: ; nastaveni T1
      ldi   temp, (1 << CS11) | (1 << CS10)
      out   TCCR1B,temp    ; vstupni preddelic nastaven na clk/64

      ; prednastaveni casovace, dodrzet poradi pristupu !
      ldi   T1H,HIGH(PRESET)
      out   TCNT1H,T1H
      ldi   T1L,LOW(PRESET)
      out   TCNT1L,T1L  ; perioda nastavena na 1s

      ; nastaveni portu "pin" na vystup
      ldi   temp,(1<<pin)
      out   DDRport,temp

      ; nastaveni stacku
      ldi   temp,HIGH(RAMEND)
      out   SPH,temp    ; min $60,roste k nizsim adresam
      ldi   temp,LOW(RAMEND)
      out   SPL,temp    ; min $60,roste k nizsim adresam


      ; nastaveni preruseni
      ldi   temp, 1 << TOIE1  ;nastaveni masky preruseni casovace
      out   TIMSK, temp
      sei               ; globalni povoleni preruseni
end:  rjmp  end         ; infinity loop

T1_ovf:
      ; ulozeni
      push  temp
      in    temp,SREG
      push  temp        ; = push SREG
      ; prednastaveni T1
      out   TCNT1H, T1H
      out   TCNT1L, T1L
      ; bliknuti pinu
      in    temp, INport
      ldi   temp1, 1 << pin
      eor   temp, temp1
      out   OUTport, temp
      ; obnova registru
      pop   temp
      out   SREG, temp
      pop   temp
      reti
```