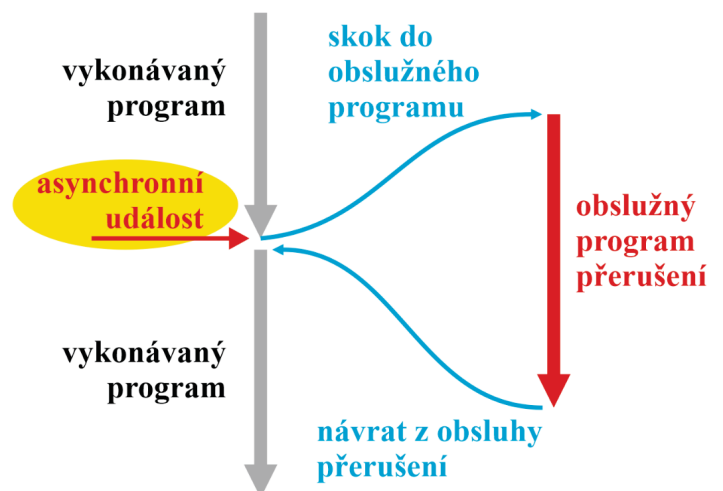


Přerušovací systém

Přerušování a výjimky (1)

- Při výskytu určité události procesor přeruší vykonávání „hlavního“ programu a začne vykonávat obslužnou proceduru pro danou událost.
- Po dokončení obslužné procedury pokračuje výpočet „hlavního“ programu.



Přerušování a výjimky (2)

- Přerušování (**Interrupt**)
 - Událost nastane mimo procesor (např. v IO řadiči).
 - Indikace vzniku události je do procesoru zavedena speciálním přerušovacím signálem.
- Výjimky (**Exceptions**)
 - Událost nastane přímo v procesoru (dělení nulou, trap, výpadek stránky, programové přerušování, ...).

Maskovatelná a nemaskovatelná přerušování:

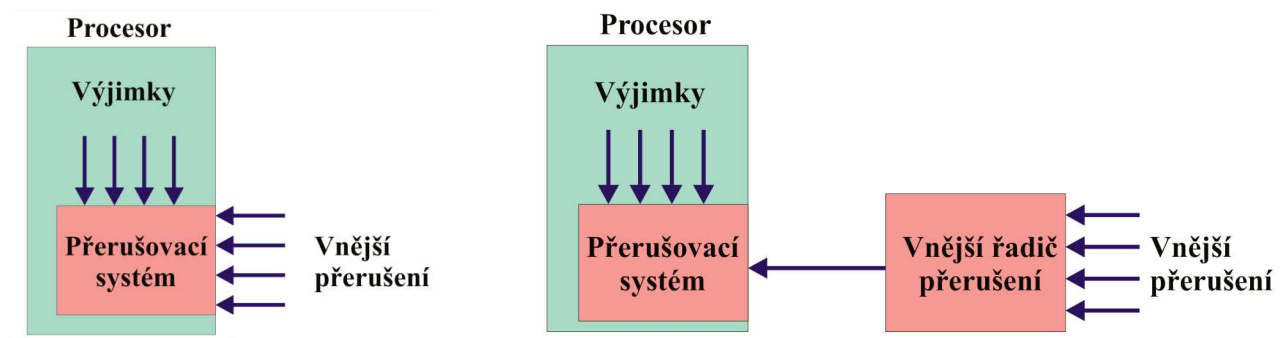
- **Maskovatelná přerušování:** Lze je programově (nastavením určitého registru v CPU resp. speciální instrukcí) povolit nebo zakázat.
- **Nemaskovatelná přerušování:** Nelze je zakázat, tj. jsou vždy povolena.

Požadavky na přerušovací systém

1. Globální povolení nebo zákaz všech přerušování (kromě NMI).
2. Selektivní povolování/zákaz jednotlivých přerušování.
3. Prioritní systém – povolení pouze přerušování s určitou prioritou.
4. Prioritní systém – při současném vzniku více přerušování se nejprve obslouží přerušování s nejvyšší prioritou.
5. Vazba každého přerušování na příslušný obslužný program.
6. Možnost vnořených přerušování.

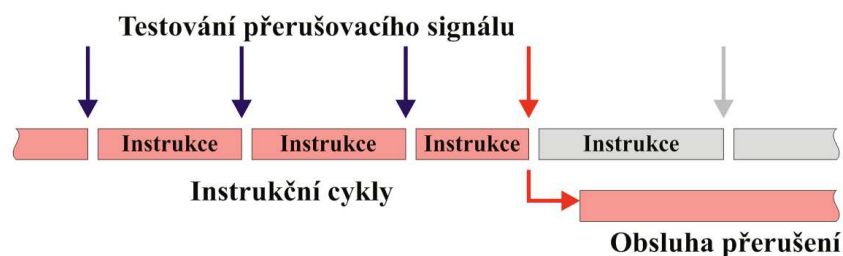
Řadič přerušení

- Řadič vnějších přerušení může být kompletně vestavěn v procesoru (např. H8S),
- nebo může být připojen jako samostatný obvod (k přerušovacímu systému procesoru).



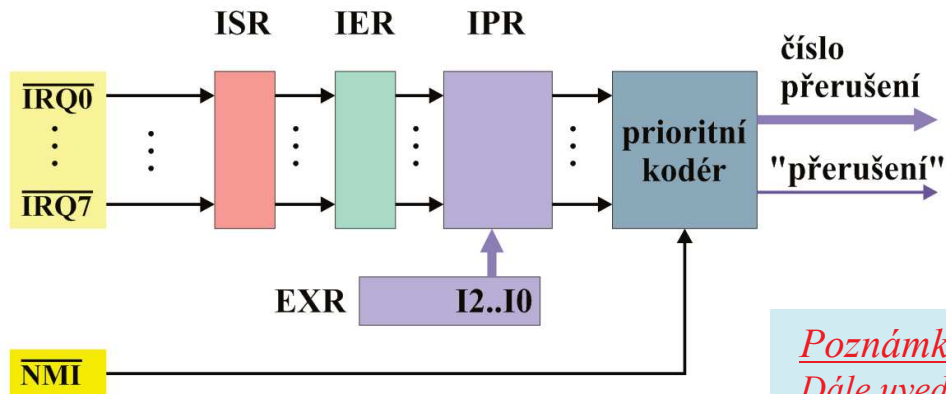
Uplatnění přerušení

- Aktivita požadavku na přerušování se testuje vždy po dokončení instrukce.
 - Není-li požadavek na přerušování aktivní, pokračuje procesor další instrukcí.
 - Je-li požadavek na přerušování aktivní (a přerušování jsou povolena), zahájí se operace pro jeho obsluhu.
- Ve speciálních případech (výpadek stránky, porušení ochrany paměti – ne u H8S) může přerušování/exception nastat i v průběhu provádění instrukce.



Přerušovací systém procesoru H8S (1)

- Několik desítek přerušení od periferních řadičů přímo na čipu (nebudeme je dále uvažovat).
- 8 vnějších přerušení (/IRQ7 - /IRQ0).
- Nemaskovatelné přerušování (/NMI).

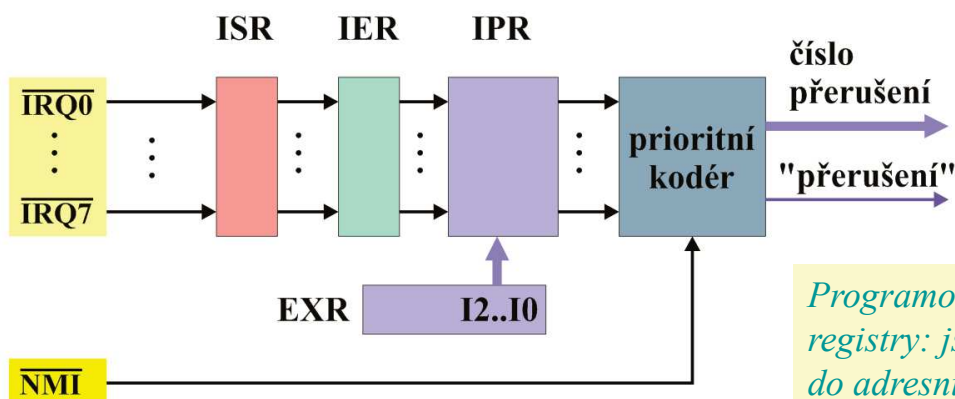


*Poznámka:
Dále uvedený popis
odpovídá přerušování v
MODE 2.*

7

Použití registrů přerušovacího systému

- **ISR** – (Interrupt Status Register) Nastavuje se automaticky, ale lze jej modifikovat i programově.
- **IER** – (Interrupt Enable Register) Nastavuje se programově podle potřeby.
- **IPR** – (Interrupt Priority Register) Nastavuje se programově podle potřeby.
- **EXR** – Nastavuje se programově, při obsluze přerušování se modifikuje automaticky.

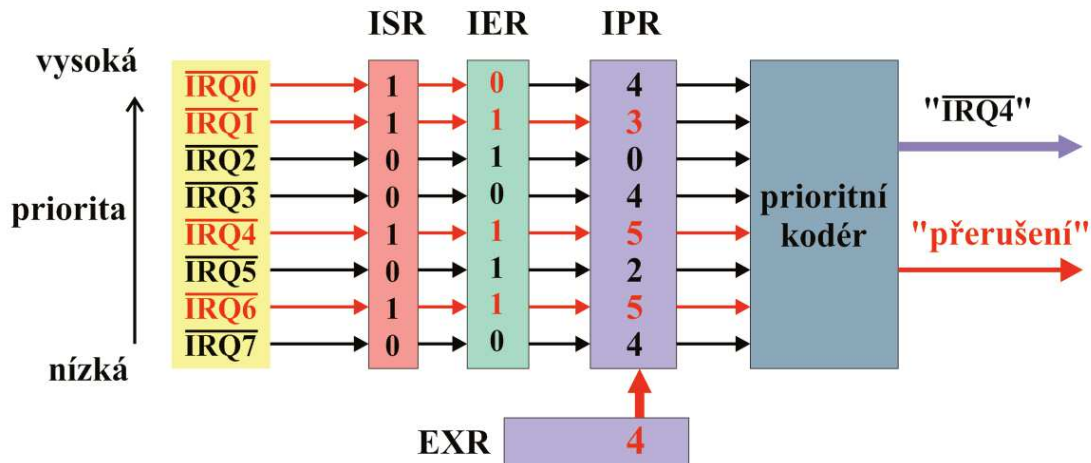


*Programově přístupné
registry: jsou mapovány
do adresního prostoru
paměti.*

8

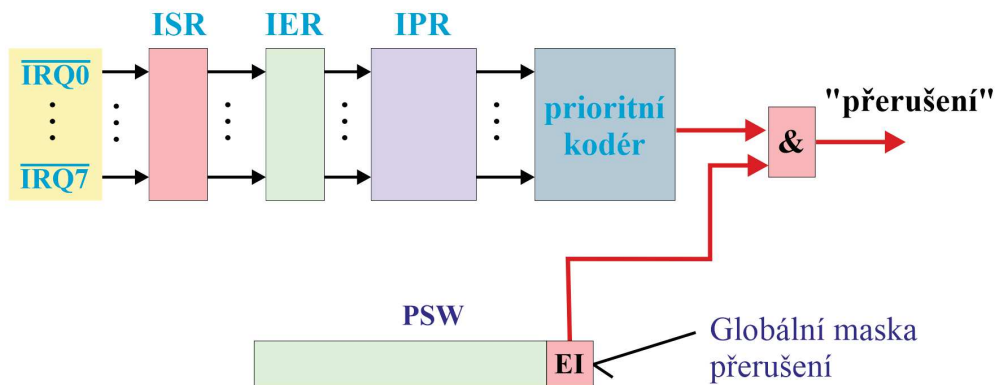
Přerušovací systém procesoru H8S (2)

- **ISR** – Registr aktivních žádostí o přerušení.
- **IER** – Povolení jednotlivých přerušení (1 = povoleno, 0 = zakázáno).
- **IPR** – Priorita přerušení (0 = nízká, 7 = vysoká).
- **EXR** – Bity **I2 . . I0** indikují úroveň povolených přerušení (musí být $IPR > EXR$, tj. při $EXR = 7$ jsou přerušení zakázána).



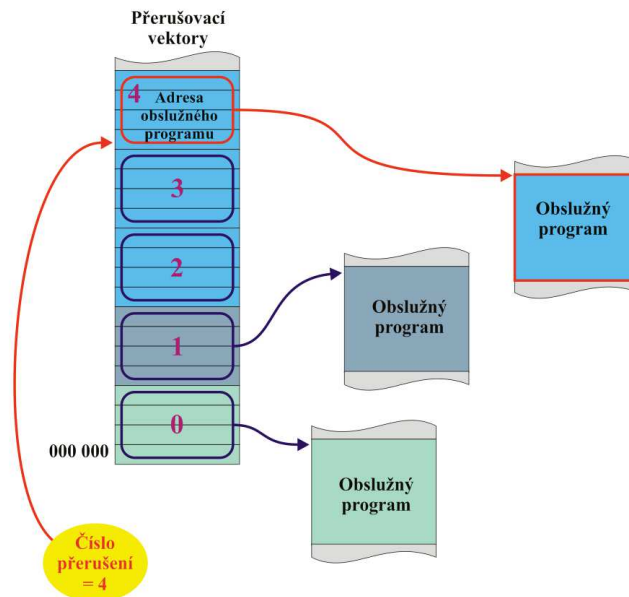
Globální zákaz / povolení přerušení

- U **H8S** v **Mode2**: musí být $IPR > EXR$, tj. při $EXR = 7$ jsou všechna přerušení zakázána.
- U mnoha jiných procesorů: Stavový registr procesoru obsahuje globální masku přerušení (např. bit **EI**, tj. Enable Interrupt), kterou lze programově nastavit na 0 nebo na 1.
 - Při akceptování přerušení se maska automaticky nastaví na „Disable Interrupt“.



Určení adresy obslužného programu (H8S)

- Na začátku paměti je tabulka s adresami obslužných programů (přerušovací vektory).
 - U některých procesorů je tabulka vektorů na jiné adrese nebo je přemístitelná.
- Každé přerušení má pevně přidělenou určitou položku v tabulce.



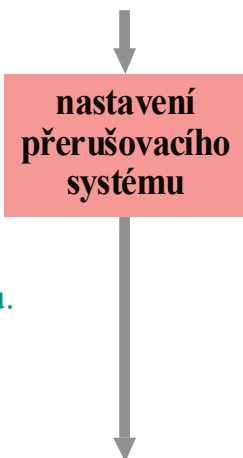
K.D. - přednášky POT

11

Obsluha přerušení (H8S) (1)

1. Inicializace přerušovacího systému

- nastavení přerušovacích vektorů,
 - staticky před spuštěním nebo za běhu programu,
- nastavení periferních řadičů,
- nastavení priorit,
- povolení přerušení
 - Obecně: speciální instrukce „Enable Interrupt“ nebo nastavením určitého bitu ve stavovém registru procesoru.
 - U H8S: zápisem zvolené priority < 7 do EXR.

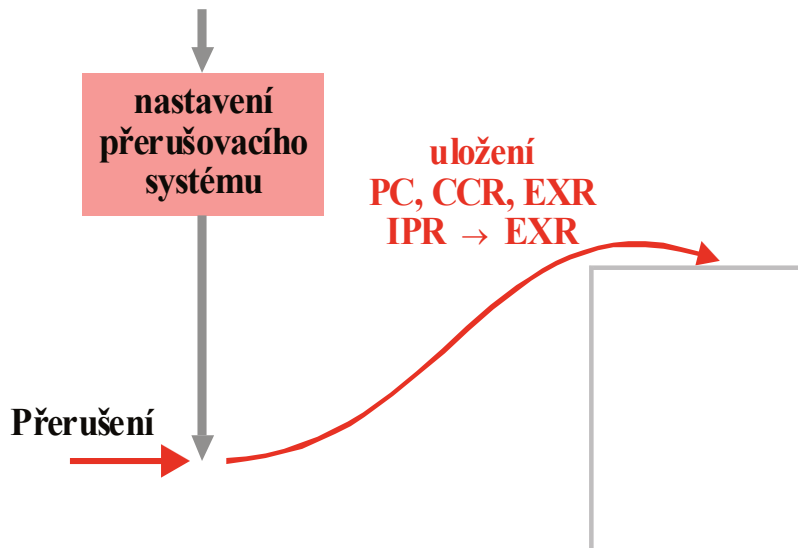


K.D. - přednášky POT

12

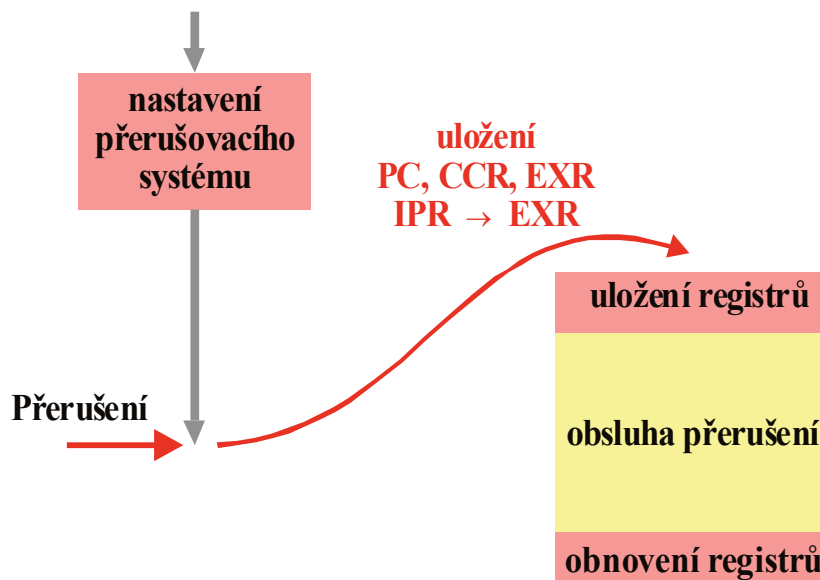
Obsluha přerušení (H8S) (2)

2. **Hardwarově** provedené operace při vzniku přerušení:
- uložení PC, CCR a EXR do zásobníku,
 - nastaví se IPR → EXR,
 - přečte se příslušný přerušovací vektor z paměti a podle něj se nastaví PC.



Obsluha přerušení (H8S) (3)

3. **Softwarově** provedené operace v obslužné proceduře přerušení:
- uložení používaných registrů do zásobníku,
 - obsluha přerušení (obsluha I/O řadiče, ...),
 - obnovení obsahu uložených registrů.

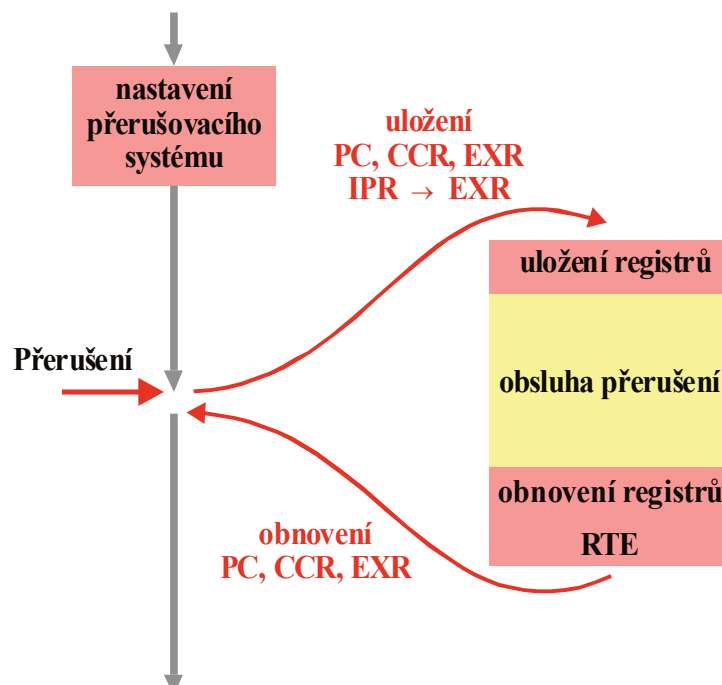


Obsluha přerušení (H8S) (4)

3. **Softwarově** provedené operace v obslužné proceduře přerušení:
- uložení používaných registrů do zásobníku,
 - obsluha přerušení (obsluha V/V řadiče, ...),
 - obnovení obsahu uložených registrů.
-
- Při obsluze určitého přerušení mohou být ostatní přerušení podle celkové koncepce programového vybavení:
 - **zakázána** – obvykle se zakáží automaticky, povolí se na konci obslužného programu (automaticky nebo programově),
 - **povolena** – obvykle se musí programově povolit na začátku obslužného programu (např. speciální instrukcí „Enable Interrupt“),
 - **povolena** pokud mají vyšší prioritu než obsluhované přerušení. Tato koncepce je podporována některými typy řadičů přerušení (např. u H8S).

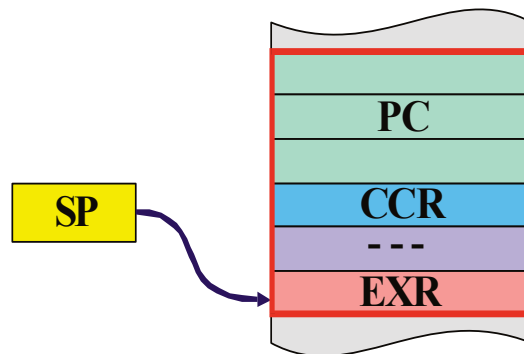
Obsluha přerušení (H8S) (5)

4. Provedení instrukce RTE na konci obslužného programu:
- ze zásobníku se postupně vybere a obnoví původní stav EXR, CCR, PC (provede se **hardwarově**).



Obsah zásobníku při obsluze přerušování (H8S)

- Do zásobníku se při obsluze přerušování hardwarově uloží:
 - PC (3 byte),
 - CCR (1 byte),
 - EXR (1 byte, 1 byte zásobníku je nevyužit).
- V obslužné proceduře se na začátku programově ukládají (a na konci obnovují) používané registry.



Detaily časování přerušování u H8S

- Spuštění obslužného programu:
 - Uložení PC, CCR a EXR do zásobníku → zápis 6 bytů,
 - Přečtení přerušovacího vektoru → čtení 4 bytů.
- Návrat z obslužného programu:
 - Čtení instrukce RTE → čtení 4 bytů.
 - Provedení RTE (obnovení EXR, CCR, PC) → čtení 6 bytů.

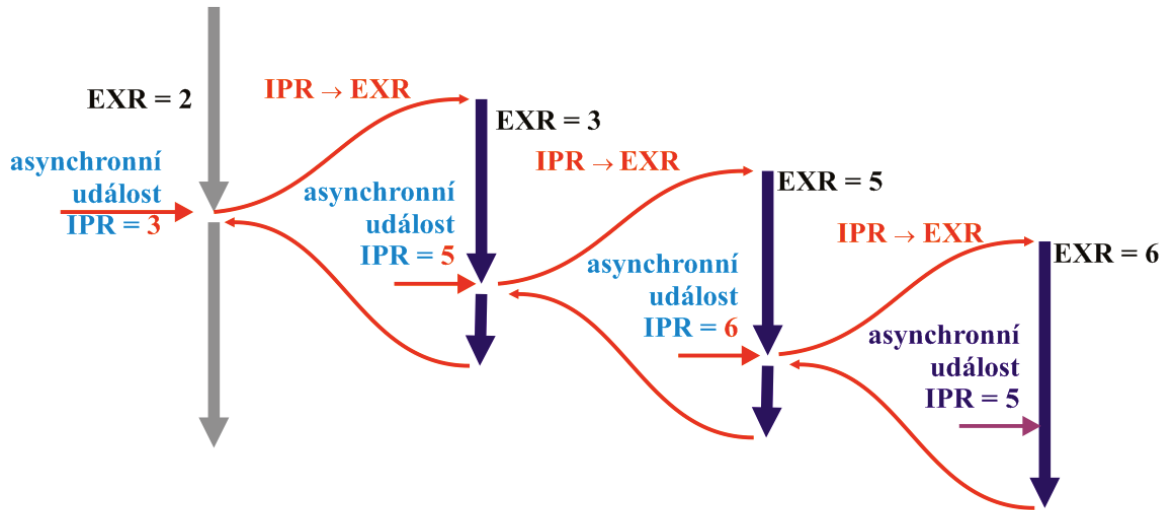


Minimální „overhead“ je **20** 8bitových nebo **10** 16bitových přístupů do paměti

- odpovídá $20 \times 3 \times 40 = 2400$ ns pro přístup 8/3 (8bitový, 3 takty),
- $10 \times 1 \times 40 = 400$ ns pro přístup 16/1 (16bitový, 1 takt).

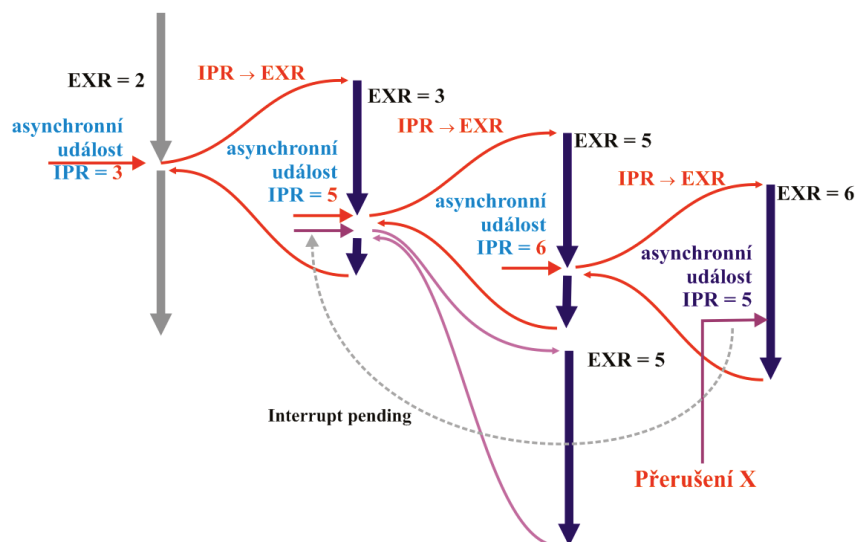
Vnořená přerušeni

- Obslužný program přerušeni je znovu přerušen
 - při akceptování přerušeni se nastaví $IPR \rightarrow EXR$, tj. další přerušeni musí mít vyšší prioritu než právě obsluhované přerušeni,
 - po dokončení obsluhy druhého přerušeni pokračuje obsluha prvního přerušeni.



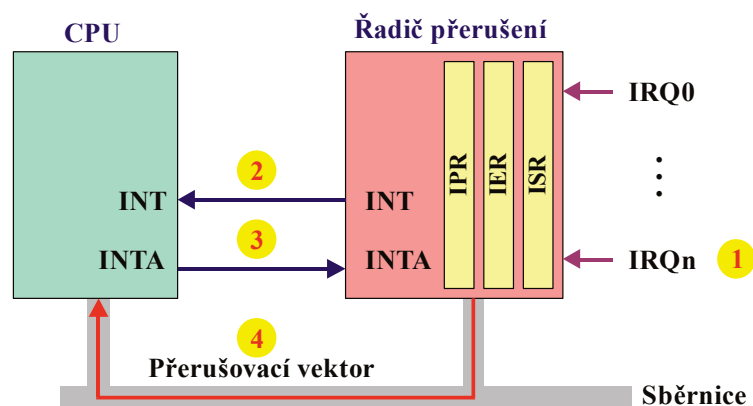
„Interrupt pending“

- Situace „Interrupt pending“
 - Přerušeni X má nízkou prioritu ($IPR = 5$) \Rightarrow nemůže se obsloužit okamžitě.
 - Přerušeni X se obslouží až po nastavení $EXR = 3$.



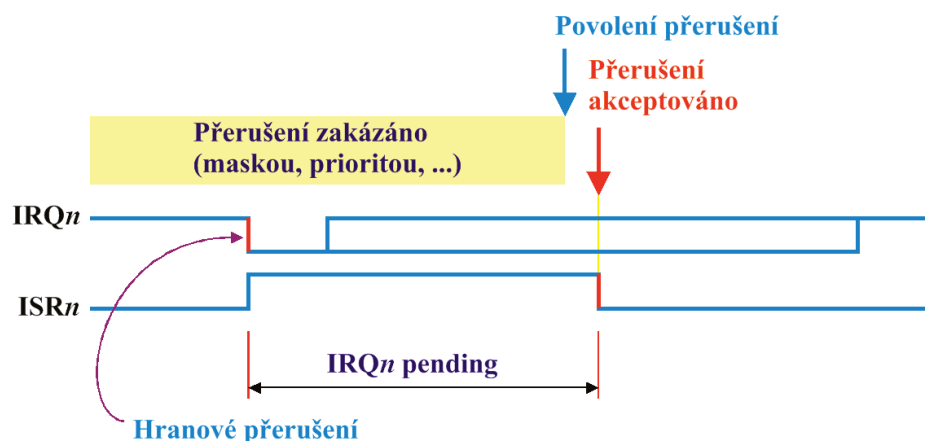
Vnější řadič přerušení

- Vnější řadič přerušení
 - obsahuje masky pro jednotlivá vnější přerušení,
 - řadí vnější přerušení podle priorit.
- Při aktivitě některého IRQ:
 1. Je aktivní IRQ_n .
 2. Je-li IRQ_n v řadiči povoleno, generuje řadič signál INT do procesoru.
 3. Jsou-li v procesoru povolena přerušení, vyšle procesor signál INTA (Interrupt Acknowledge).
 4. Řadič vyšle na sběrnici vektor s identifikací akceptovaného IRQ.



Hranové přerušení

- Přerušování se aktivuje (nastaví se ISR_n) hranou \downarrow nebo \uparrow .
- ISR_n se nuluje (automaticky) při akceptování přerušování.



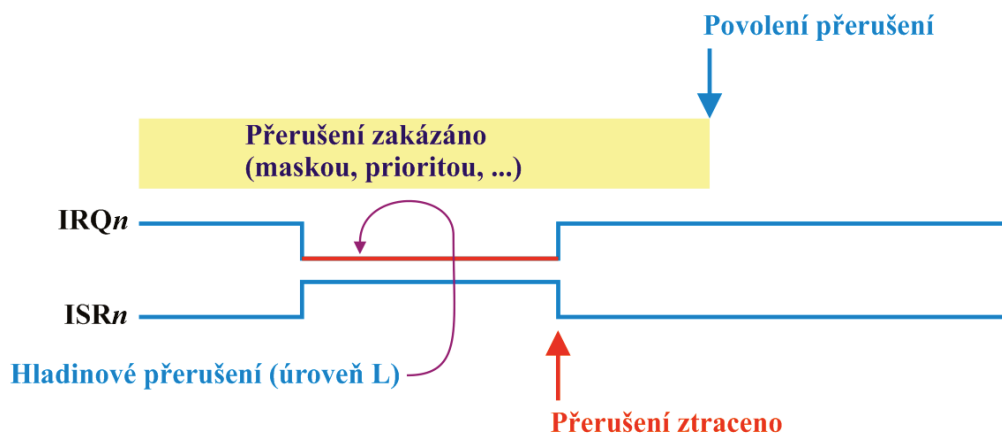
Hladinové přerušování (1)

- Přerušování se aktivuje (nastaví se **ISR_n**) úrovní H nebo L.
- **ISR_n** je aktivní až do zrušení aktivity **IRQ_n** (například programovou obsluhou periferie).



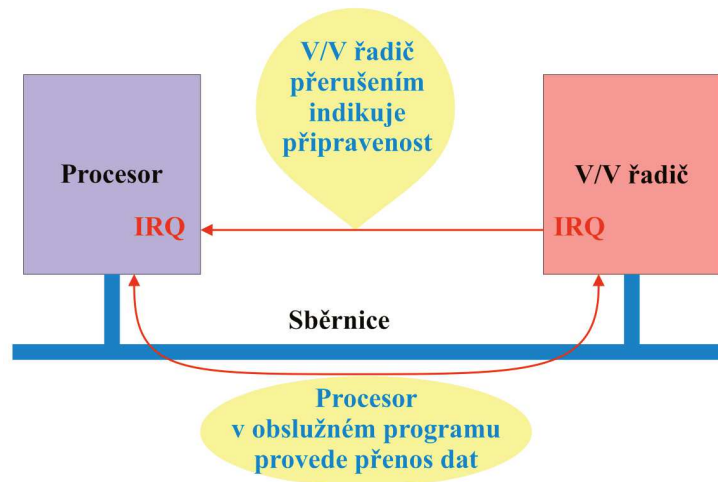
Hladinové přerušování (2)

- Přerušování se aktivuje (nastaví se **ISR_n**) úrovní H nebo L.
- Skončí-li aktivita **IRQ_n** před akceptováním přerušování, může se přerušování ztratit (jen u některých systémů, ne u H8S).



Použití přerušování k řízení IO operací

- Přerušování lze použít k řízení IO operací:
 - Řadič IO zařízení musí být navržen tak, aby indikoval připravenost speciálním signálem, který se zavede na přerušovací vstup procesoru.
 - Připravenost IO systému k přenosu způsobí přerušování.
 - V obslužné proceduře se provede zápis/čtení dat do/z IO řadiče.



Použití přerušování k řízení IO operací

- Charakteristiky IO systému s přerušováním:
 - Procesor **nemusí** periodicky testovat stav IO systému (může provádět jiný výpočet).
 - Maximální rychlost přenosu je omezena především rychlostí reakce na přerušovací signál (vyžaduje řadu operací).

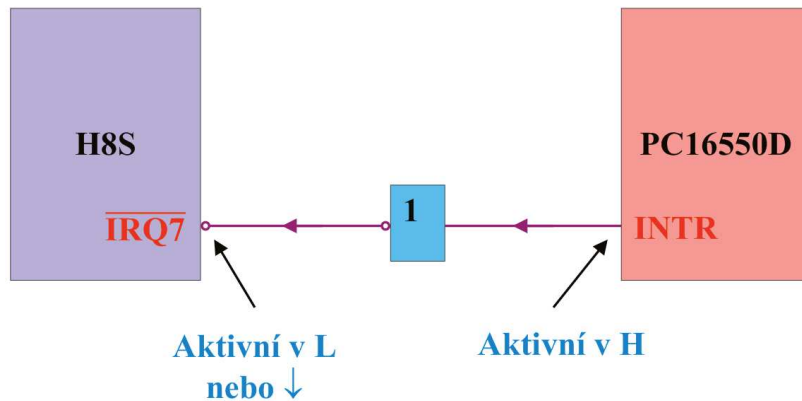


IO systém s přerušováními je vhodný především pro pomalé přenosy „znak_po_znaku“.

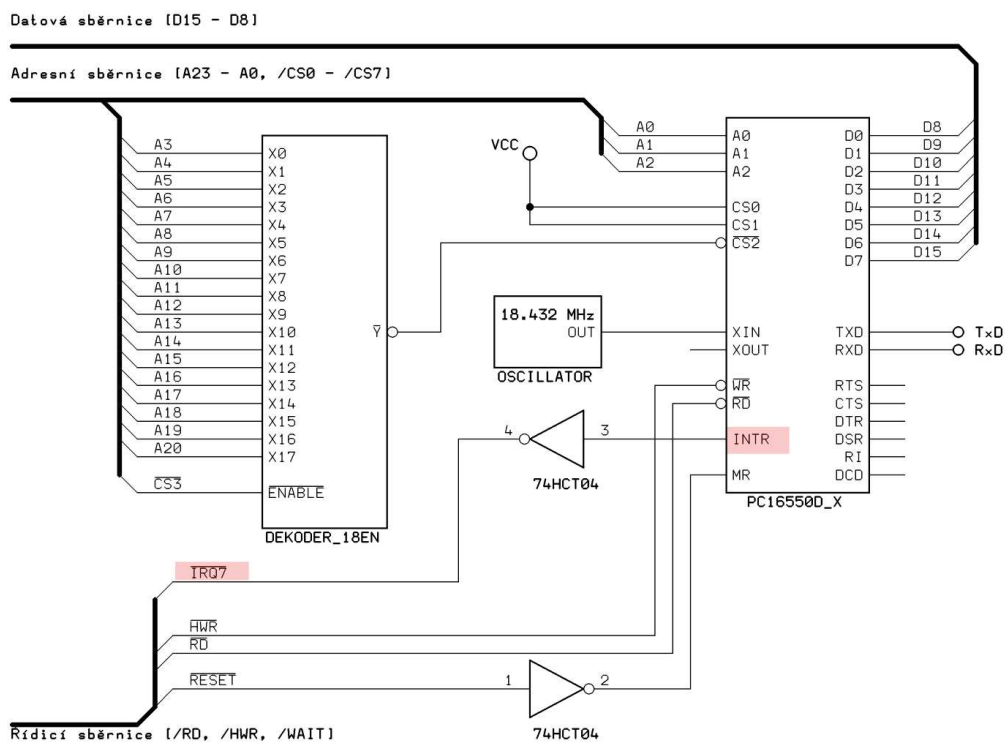
Příklad použití přerušování

Vysílání dat sériovou linkou s PC16550D s využitím přerušování.

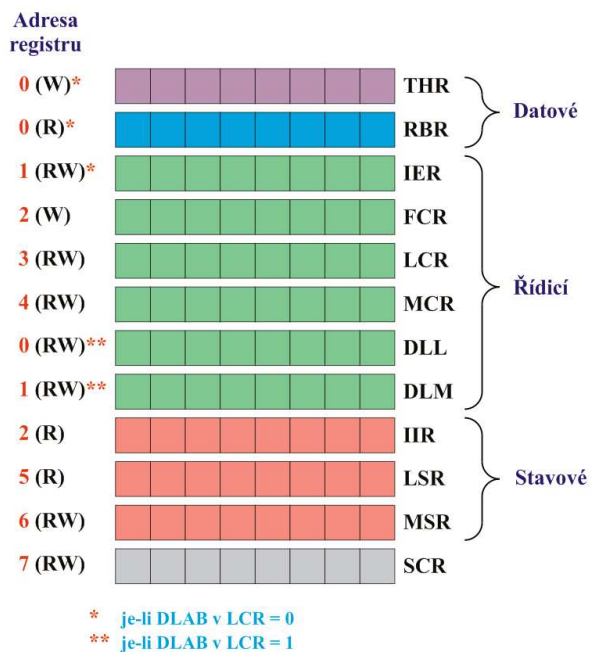
- Signál INTR obvodu PC16550D připojíme na některý vstup pro přerušování H8S (je nutné použít negaci signálu INTR).



Příklad použití přerušování



Registry PC16550D



- **THR** – data pro vysílání.
- **RBR** – přijatá data.
- **IER (Interrupt Enable Register)** – povolení přerušení.
- **FCR (FIFO Control Register)** – řízení fronty FIFO.
- **LCR (Line Control Register)** – řízení přenosu.
- **MCR (Modem Control Register)** – řízení modemu.
- **DLL, DLM** – nastavení přenosové rychlosti.
- **IIR – (Interrupt Identification Register)** – identifikace příčiny přerušení.
- **LSR (Line Status Register)** – stav přenosu.
- **MSR (Modem Status Register)** – stav modemu.

Přerušení generované obvodem PC16550D (1)

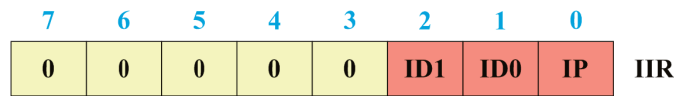
- PC16550D generuje přerušení v následujících situacích:
 - Při příjmu dat došlo k chybě PE, FE nebo OE nebo byl detekován BREAK.
 - Registr s přijatým znakem (RBR) je plný.
 - Registr pro vysílaný znak (THR) je prázdný.
 - Změna logické úrovně některého signálu pro řízení modemu (/CTS, /DSR, /RI, /DCD).
- Generování přerušení pro každou z příčin lze samostatně povolit nebo zakázat v registru IER:



- EDSSI – Enable Data Set Status Interrupt (1 = povoleno, 0 = zakázáno),
- ELSI – Enable Line Status Interrupt (1 = povoleno, 0 = zakázáno),
- ETBEI – Enable Transmit Buffer Empty Interrupt (1 = povoleno, 0 = zakázáno),
- ERBFI – Enable Receiver Buffer Full Interrupt (1 = povoleno, 0 = zakázáno).

Přerušování generované obvodem PC16550D (2)

- Příčinu přerušování lze zjistit v registru IIR (Interrupt Identification Register):



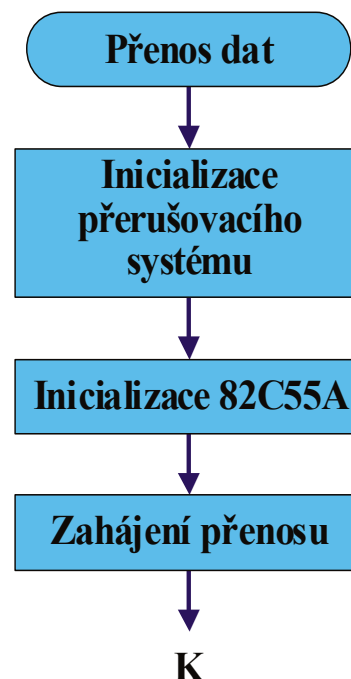
IP - Interrupt Pending:
 0 = požadavek na přerušování v PC16550D je aktivní,
 1 = není požadavek na přerušování.

ID1, ID0 - identifikace příčiny přerušování:

ID1	ID0	Příčina	Priorita
1	1	Chyba při příjmu	vysoká ↑ ↓ nízká
1	0	RBR naplněn	
0	1	THR prázdný	
0	0	Změna signálů z modemu	

Inicializace přenosu

- Inicializace přerušovacího systému
 - nastavení přerušovacího vektoru,
 - nastavení priorit IPR a EXR,
 - nastavení typu /IRQ7 (hladinové).
- Inicializace PC16550D
 - nastavení parametrů přenosu v LCR,
 - povolení přerušování od Tx.
- Zahájení přenosu
 - nastavení ukazatele na data,
 - povolení přerušování /IRQ7.



Poznámka: Pro přerušování použijeme vstup /IRQ7.

Přerušovací vektory u H8S

Table 5-4 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*		Priority
			Advanced Mode	IPR	
NMI	External pin	7	H'001C		High
IRQ0		16	H'0040	IPRA6 to 4	
IRQ1		17	H'0044	IPRA2 to 0	
IRQ2		18	H'0048	IPRB6 to 4	
IRQ3		19	H'004C		
IRQ4		20	H'0050	IPRB2 to 0	
IRQ5		21	H'0054		
IRQ6	22	H'0058	IPRC6 to 4		
IRQ7	23	H'005C			
SWDTEND (software activation interrupt end)	DTC	24	H'0060	IPRC2 to 0	
WOVI0 (interval timer)	Watchdog timer 0	25	H'0064	IPRD6 to 4	
CMI	Refresh timer	26	H'0068	IPRD2 to 0	
PC break	PC break	27	H'006C	IPRE6 to 4	
ADI (A/D conversion end)	A/D	28	H'0070	IPRE2 to 0	
WOVI1 (interval timer)	Watchdog timer 1	29	H'0074		
Reserved	—	30	H'0078		

K.D. - přednášky POT

Registry pro řízení přerušení H8S (1)

- Registry jsou mapovány do paměťového adresního prostoru.

Table 5-2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	Address* ¹
System control register	SYSCR	R/W	H'01	H'FDE5
IRQ sense control register H	ISCRH	R/W	H'00	H'FE12
IRQ sense control register L	ISCR L	R/W	H'00	H'FE13
IRQ enable register	IER	R/W	H'00	H'FE14
IRQ status register	ISR	R/(W)* ²	H'00	H'FE15
Interrupt priority register A	IPRA	R/W	H'77	H'FEC0
Interrupt priority register B	IPRB	R/W	H'77	H'FEC1
Interrupt priority register C	IPRC	R/W	H'77	H'FEC2
Interrupt priority register D	IPRD	R/W	H'77	H'FEC3
Interrupt priority register E	IPRE	R/W	H'77	H'FEC4
Interrupt priority register F	IPRF	R/W	H'77	H'FEC5
Interrupt priority register G	IPRG	R/W	H'77	H'FEC6
Interrupt priority register H	IPRH	R/W	H'77	H'FEC7
Interrupt priority register I	IPRI	R/W	H'77	H'FEC8

K.D. - přednášky POT

Registry pro řízení přerušení H8S (2)

- IPRC (Interrupt Priority Register C)

Table 5-3 Correspondence between Interrupt Sources and IPR Settings

Register	Bits	
	6 to 4	2 to 0
IPRA	IRQ0	IRQ1
IPRB	IRQ2 IRQ3	IRQ4 IRQ5
IPRC	IRQ6 IRQ7	DTC
IPRD	Watchdog timer 0	Refresh timer
IPRE	PC break	A/D converter, watchdog timer 1
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRI	8-bit timer channel 0	8-bit timer channel 1
IPRJ	DMAC	SCI channel 0
IPRK	SCI channel 1	SCI channel 2

Registry pro řízení přerušení H8S (2)

- IPRC (Interrupt Priority Register C)

5.2.2 Interrupt Priority Registers A to L, O (IPRA to IPRL, IPRO)

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	:	0	1	1	1	0	1	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

Registry pro řízení přerušení H8S (3)

- IER (Interrupt Enable Register)

5.2.3 IRQ Enable Register (IER)

Bit	:	7	6	5	4	3	2	1	0
		IRQ7E	IRQ6E	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 to 0—IRQ7 to IRQ0 Enable (IRQ7E to IRQ0E): These bits select whether IRQ7 to IRQ0 are enabled or disabled.

Bit n	IRQnE	Description
	0	IRQn interrupts disabled (Initial value)
	1	IRQn interrupts enabled

(n = 7 to 0)

Registry pro řízení přerušení H8S (4)

- ISCRH (Interrupt Sense Control Register H)

5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

ISCRH

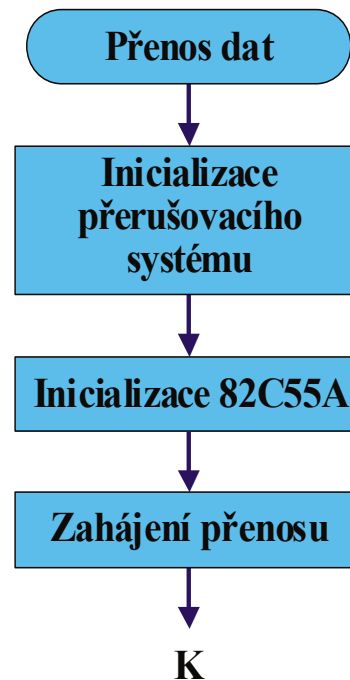
Bit	:	15	14	13	12	11	10	9	8
		IRQ7SCB	IRQ7SCA	IRQ6SCB	IRQ6SCA	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0

IRQ7SCB to IRQ0SCB	IRQ7SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input low level (initial value)
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ7}}$ to $\overline{\text{IRQ0}}$ input

Inicializace přenosu (1)

- Inicializace přerušovacího systému
 - nastavení přerušovacího vektoru,
 - nastavení priorit IPR a EXR,
 - nastavení typu /IRQ7 (hladinové).
- Inicializace PC16550D
 - nastavení parametrů přenosu v LCR,
 - povolení přerušování od Tx.
- Zahájení přenosu
 - nastavení ukazatele na data,
 - povolení přerušování /IRQ7.



Inicializace přenosu (2)

```

.equ  BASE16550, 0x600000    ;bazova adresa PC16550D
.equ  THR,      BASE16550    ;transmit register
.equ  IERT,     BASE16550+1  ;interrupt control
.equ  IIR,     BASE16550+2  ;interrupt identification
.equ  LCR,     BASE16550+3  ;line control register
.equ  LSR,     BASE16550+5  ;line status register
.equ  DLL,     BASE16550    ;nast. frekvence low
.equ  DLM,     BASE16550+1  ;nast. frekvence high
...
TEXT:  .asciz  „AHOJ“        ;text pro vysilani + 0x00
TX_PTR: .space  4           ;pointer na vysílaná data
...
START: JSR    @INIT_IRQ7    ;inicializace v H8S
       JSR    @INIT16550    ;inicializace PC16550D
       JSR    @OPEN_TX      ;procedura pro vysilani
...
; tady procesor provádí výpočet, který je přerušován
; přerušním od PC16550D
...

```

Inicializace přenosu (3)

```

INIT_IRQ7: MOV.L    #SERV_7,ER0    ;prerusovaci vektor
           MOV.L    ER0,@0x0000005C ;na spravne misto
;
           MOV.B    @ISCRH,R0L    ;typ preruseni
           AND.B    #0x3F,R0L    ;[7..6]=00 -> hladinove
           MOV.B    R0L,@ISCRH    ;zapis
;
           MOV.B    @IPRC,R0L    ;priorita
           AND.B    #0x0F,R0L    ;nulovani [7..4]
           OR.B     #0x40,R0L    ;priorita IPR = 4
           MOV.B    R0L,@IPRC    ;zapis
;
           LDC.B    #0x03,EXR    ;priorita EXR = 3
           RTS

```

Inicializace přenosu (4)

```

INIT16550: MOV.B    @LCR,R0L    ;puvodni hodnota LCR
           OR.B     R0L,#0x80    ;DLAB nastaven na 1
           MOV.B    R0L,@LCR    ;zapis do LCR
           MOV.W    #120,R0    ;delici pomer do R0
           MOV.B    R0L,@DLL    ;zapis dolni casti
           MOV.B    R0H,@DLM    ;zapis horni casti
           MOV.B    #0x07,R0L    ;8bitu, 2stop, bez parity
           MOV.B    R0L,@LCR    ;nastaveni LCR, DLAB = 0
           MOV.B    #0x02,R0L    ;ETBEI -> R0L
           MOV.B    R0L,@IERT*1) ;zapis do IER
           RTS                    ;navrat

```

***1)** Od tohoto okamžiku je aktivní signál INTR → IRQ7 .

7	6	5	4	3	2	1	0	
0	0	0	0	EDSSI	ELSI	ETBEI	ERBFI	IER

Inicializace přenosu (5)

```

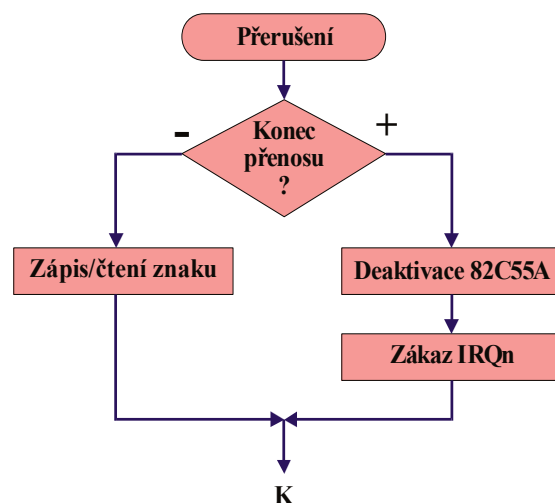
;
OPEN_TX:  MOV.L   #TEXT,ER6      ;adresa textu do ER6
          MOV.L   ER6,@TX_PTR    ;ulozeni pointeru na data
          MOV.B   @IER,R0L       ;povoleni IRQn
          OR.B    #0x80,R0L      ;IRQ7 povoleno
          MOV.B   R0L,@IER*2)    ;zapis
          RTS

```

***2)** Po zápisu do IER nastane první přerušení .

Obsluha přerušení od 16550D (1)

- Není-li konec přenosu:
 - zápis dalšího znaku,
 - posun pointeru.
- Na konci přenosu:
 - zákaz přerušení v PC16550D,
 - zákaz přerušení /IRQn v CPU.



Obsluha přerušování od 16550D (2)

```

SERV_7:   PUSH      R0                ;ulozeni R0
          PUSH.L   ER6              ;ulozeni ER6
          MOV.L    @TX_PTR,ER6      ;pointer na data do ER6
          MOV.B    @ER6,R0L        ;cteni dat z pameti
          BNE     LAB01             ;(data != 0x00) => neni konec
;
          XOR.B    #R0L,R0L        ;vse preneseno, 0 -> R0L
          MOV.B    R0L,@IERT       ;zakaz preruseni v 16550
          MOV.B    @IER,R0L        ;povoleni/zakaz IRQn
          AND.B    #0x7F,R0L       ;IRQ7 zakazano
          MOV.B    R0L,@IER        ;zapis
;
          Zde by měla být indikace konce přenosu operačnímu systému apod.
          BRA     LAB02             ;na konec
;
LAB01:   MOV.B    R0L,@THR          ;zapis znaku do THR
          INC.L   ER6              ;posunuti pointeru
          MOV.L   ER6,@TX_PTR      ;uklizeni pointeru
LAB02:   POP.L   ER6              ;obnoveni ER6
          POP     R0               ;obnoveni R0
          RTE                      ;navrat

```

K.D. - přednásky PO1

Poznámky k příkladu

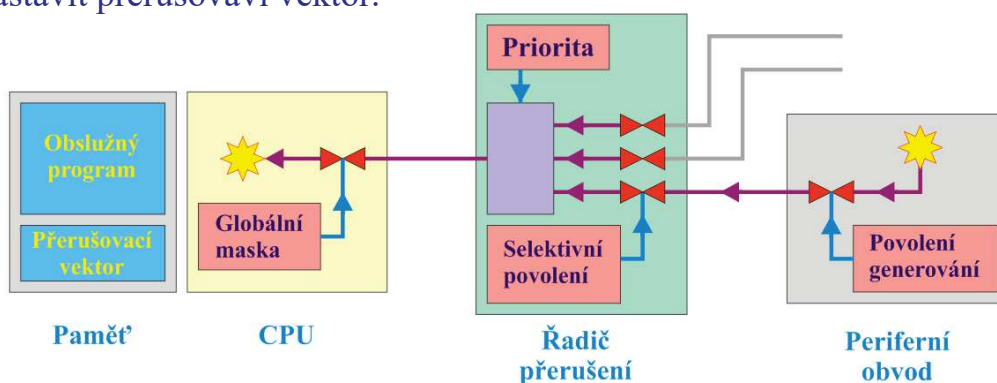
1. Registr se standardním názvem IER existuje v PC16550D i v H8S. Pro rozlišení je v programu pro registr v PC16550D použit název IERT.
 - Název registru v programu není důležitý, podstatná je adresa, na které je mapován do adresního prostoru.
 - **Ale:** k většině procesorů existují standardní hlavičkové soubory, které definují adresy jejich registrů – musí se potom použít standardní názvy.
2. Protože je v PC16550 povoleno jen přerušování ETBEI, není nutné v obslužné proceduře číst IIR a zjišťovat příčinu přerušování.
 - Pokud by bylo povoleno i jiné přerušování, musela by se obsluha doplnit o zjištění příčiny a ošetření všech možných případů.
3. Odstraněním příčiny přerušování (zápisem dat do THR) přestane být aktivní přerušovací signál INTR.
4. Obslužný program obsahuje 11 instrukcí. Předpokládáme-li 1 instrukce $\approx 1 \mu\text{s}$, trvá obsluha přerušování cca. $13 \mu\text{s}$ (včetně operací před vstupem do obslužného programu). Maximální přenosová rychlost je potom cca 75 kB/s .

Přerušování H8S v MODE 0

- Kromě popsaného MODE 2 má H8S jednoduchý přerušovací režim MODE 0
- Priority:
 - nelze nastavit priority přerušování v registru IPR,
 - nepoužívá se registr EXR,
- Povolení/zákaz přerušování:
 - jednotlivá přerušování lze povolit/zakázat v registru IER,
 - všechna přerušování se povolují/zakazují společně bitem I v registru CCR.

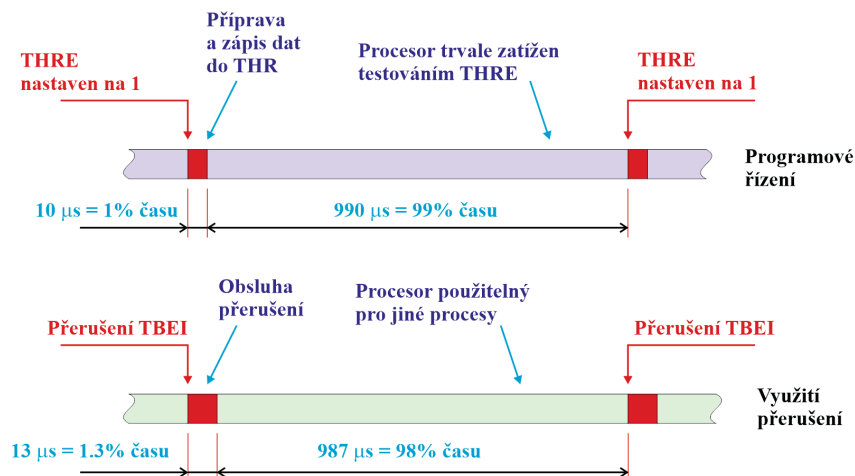
Shrnutí

- Pro aktivaci přerušování je nutné:
 1. povolit generování přerušování v periferním obvodu,
 2. povolit přerušování v řadiči přerušování,
 3. nastavit správnou prioritu přerušování,
 4. globálně povolit přerušování v CPU.
- Pro správnou obsluhu přerušování je nutné:
 1. napsat obslužný program,
 2. nastavit přerušovací vektor.



Porovnání časových poměrů

- Porovnání využití CPU při programovém řízení IO a při využití přerušování
 - Předpoklad: řízení sériového rozhraní s rychlostí 9k6.

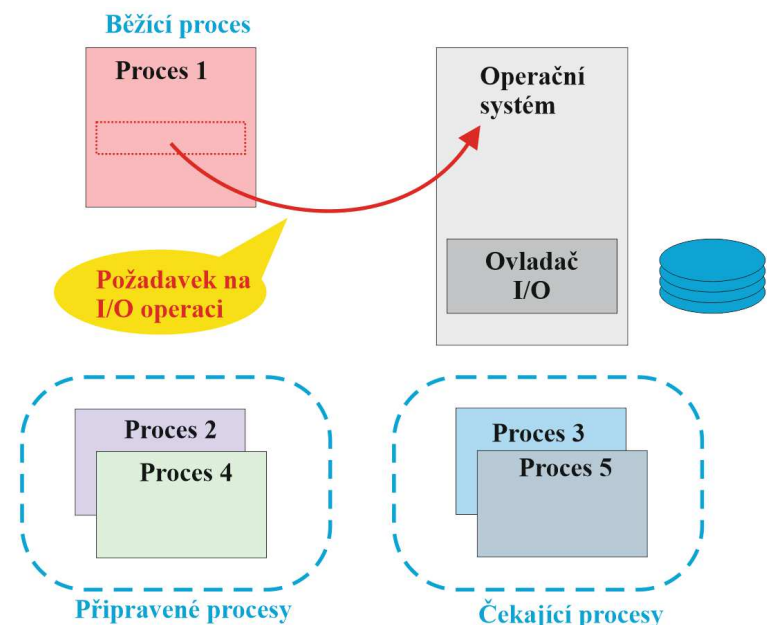


IO operace a procesy v OS (1)

- IO operace obvykle řídí ovladače, které jsou součástí OS.
- Úloha (proces) musí OS požádat o provedení IO přenosu (**voláním služeb OS**).

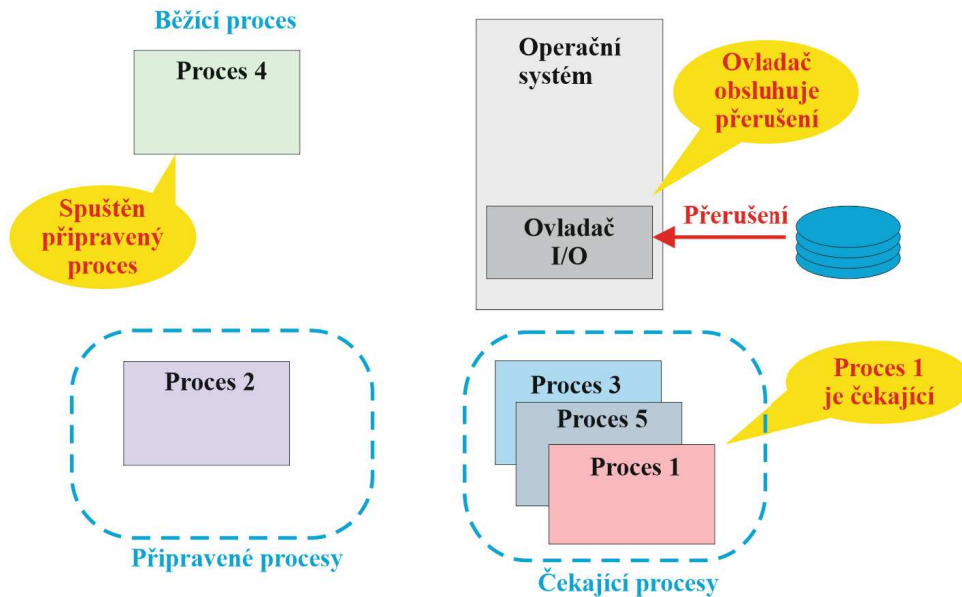
Čekající procesy: čekají na dokončení IO operace (nebo na jinou událost).

Připravené procesy: jsou schopné spuštění, čekají jen na procesor.



IO operace a procesy v OS (2)

- Proces který čeká na dokončení IO operace je zařazena mezi čekající procesy.
- OS spustí některý z připravených procesů.



IO operace a procesy v OS (3)

- Po dokončení IO operace zařadí OS **Proces 1** mezi připravené.

