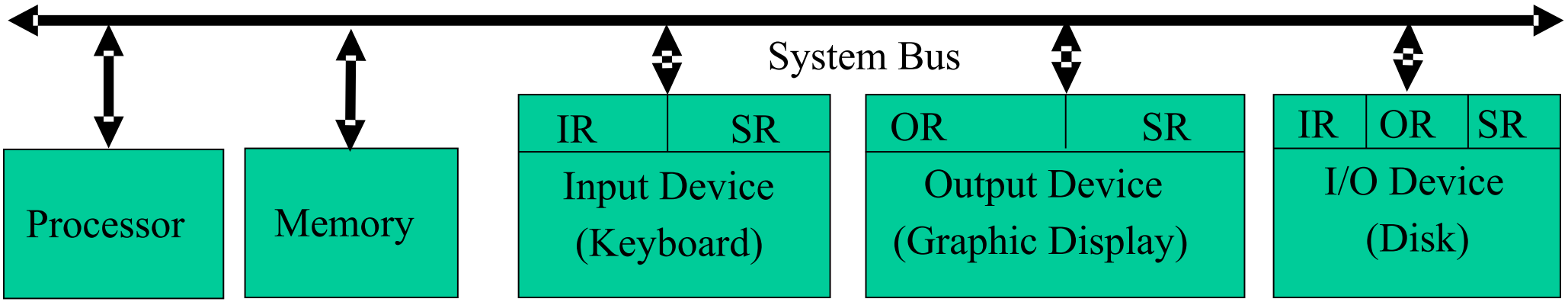
Vstupno/výstupný (Input/Output) podsystém

(Riadenie - obsluha periférií)



Riadiace obvody I/O modulov sú k zbernici počítača pripojené podobne ako pamäťové obvody (vývodmi pre údajovú, adresovú/adresovaciu a riadiacu zbernicu). Procesorom ich môžeme riadiť cez registre, a to:

-zápisom do **riadiaceho registra** určujeme režim (spôsob) činnosti I/0

- čítaním zo **stavového registra** sledujeme stav I/O

- čítaním z **údajového registra** a zapisovaním do údajov registra prebieha prenos informácií medzi CPU (pamäťou) a I/O zariadením

Základný charakteristický znak periférnych obvodov:

Prenosová rýchlosť I/O zariadení:

- Klávesnica: „cca“ 10 znakov (B)/sek. Charakteristika prenosu: Komunikácia procesor – klávesnica. Väčšinou sa náhodné prenášajú byty, resp. niekoľko bytov. CPU musí neustále *testovať „zatlačenie“* klávesnice. Kód zatlačeného znaku sa uloží do *Input register* a radič klávesnice nastaví príznak pripravenosti znaku. Procesor znak prečíta a zruší príznak => znak sa prečíta len raz. Procesor trávi veľa času testovaním: => použiť Prerušenie/*Interrupt*.

Opačný proces možno badať smerom k displeju, resp. k tlačiarni..

- Laserová tlačiareň: cca. 100 000 znakov/sek,
- Grafický displej: cca. 30 000 000 znakov/sek,
- Pevný disk: cca milióny B/sek. Charakteristika prenosu: Prenos veľkého množstva údajov organizuje procesor po bytoch, resp. slovách. => Strata času. Výhodnejšie je prenos organizovať po blokoch (zabezpečuje hardware) : => Rýchlejší prenos. Procesor „oddychuje“, resp. robí inú prácu. => *DMA*.

Pripojenie I/O obvodov k zbernici počítača

Z uvedeného je zrejmé že procesor musí vedieť príslušné I/O zariadenia adresovať.

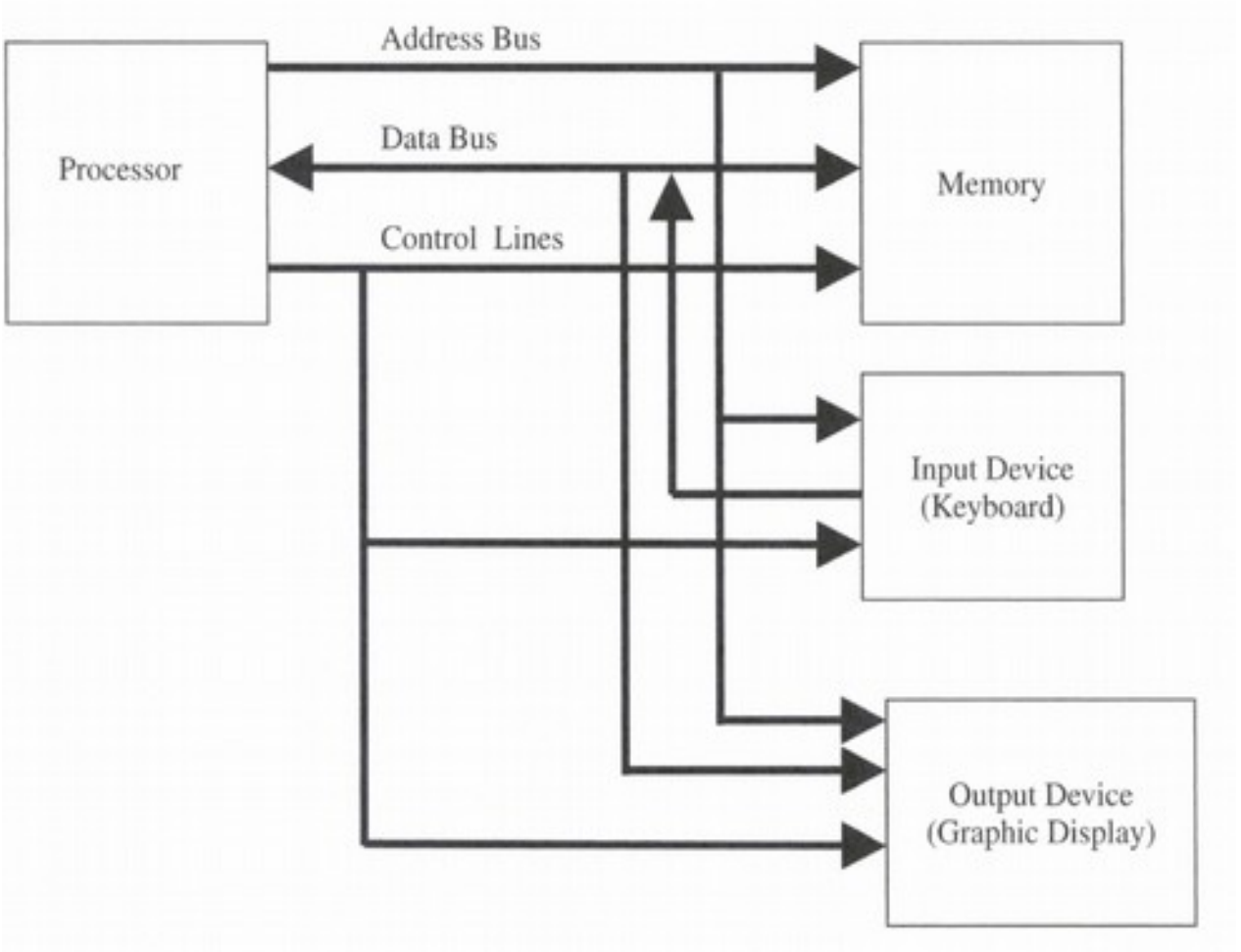
- **Pamäťovo mapované I/O**

  Pre I/O zariadenia možno použiť celú pamäť

– Periférie a pamäť zdieľajú ten istý pamäťový priestor

– Žiadny špeciálny príkaz pre I/O

– (Motorola).



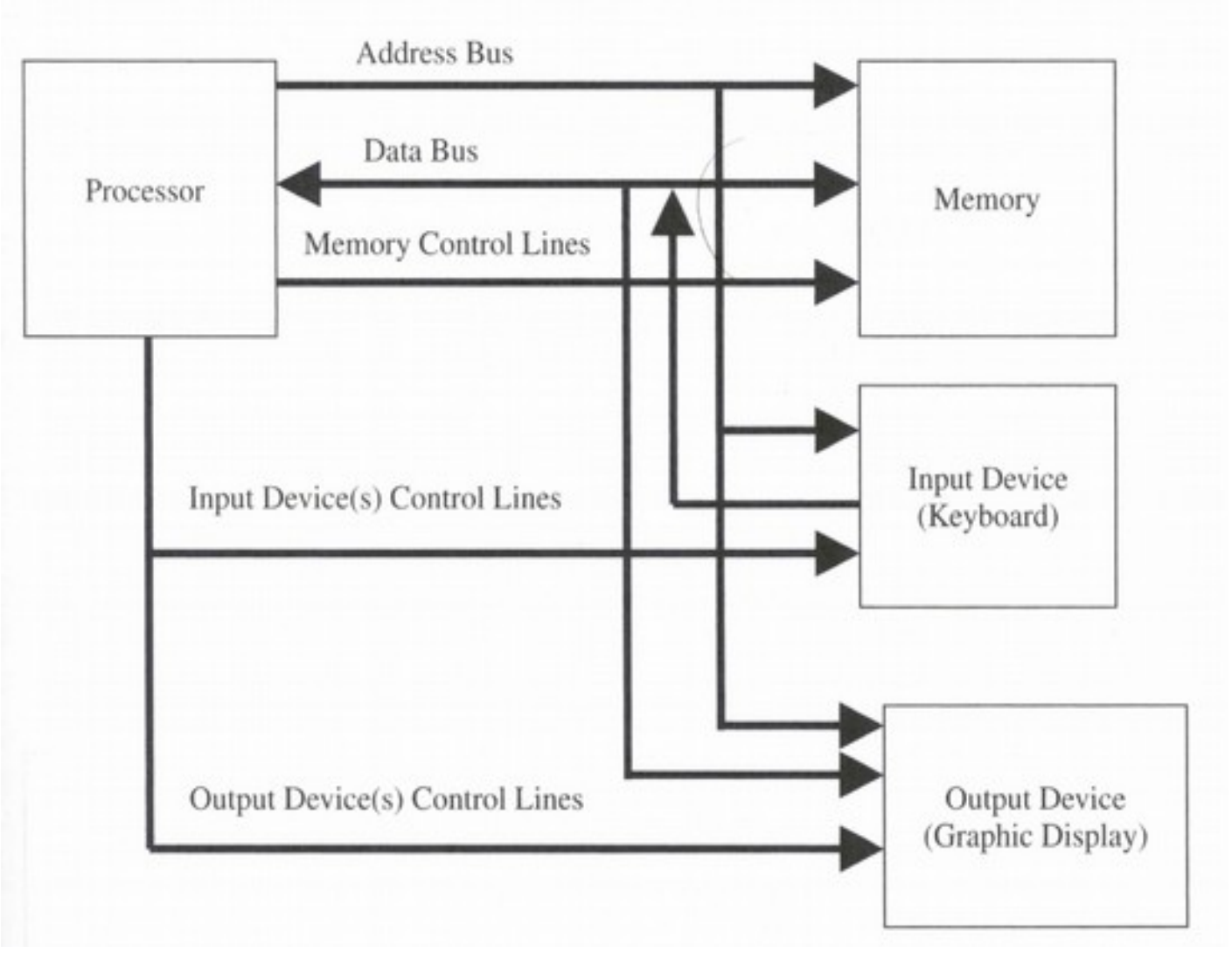
- **Samostatné mapované I/O zariadenia**

– Oddelený adresný priestor

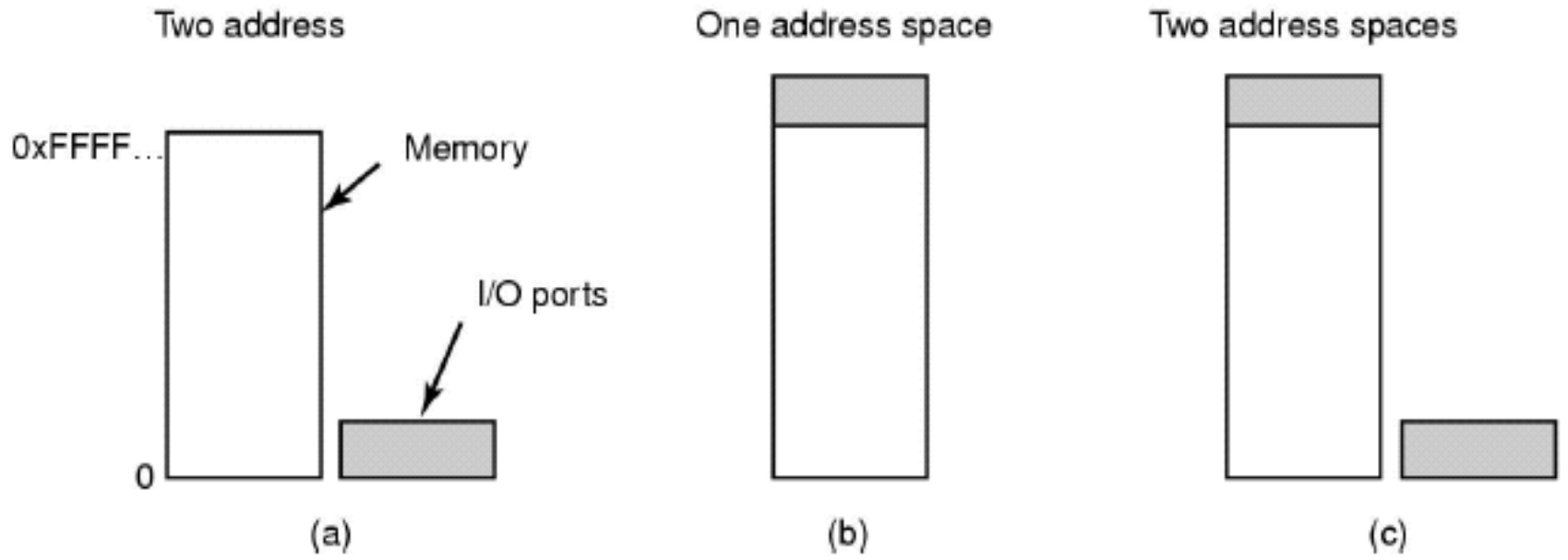
– Špeciálne vodiče pre výber I/O a pamäť

– Špeciálne príkazy I/O

– (Intel x86
a kompatibilné
=nahrádzajúce)



Mapovanie I/O obvodov, I/O Mapping



- Separate I/O and memory space — Pôvodné IBM PC
- Memory-mapped I/O — 8051
- Hybrid — Dnešné PC,AVR

Riadenie komunikácie s I/O podsystémom

Tri základné spôsoby :

S účasťou procesora

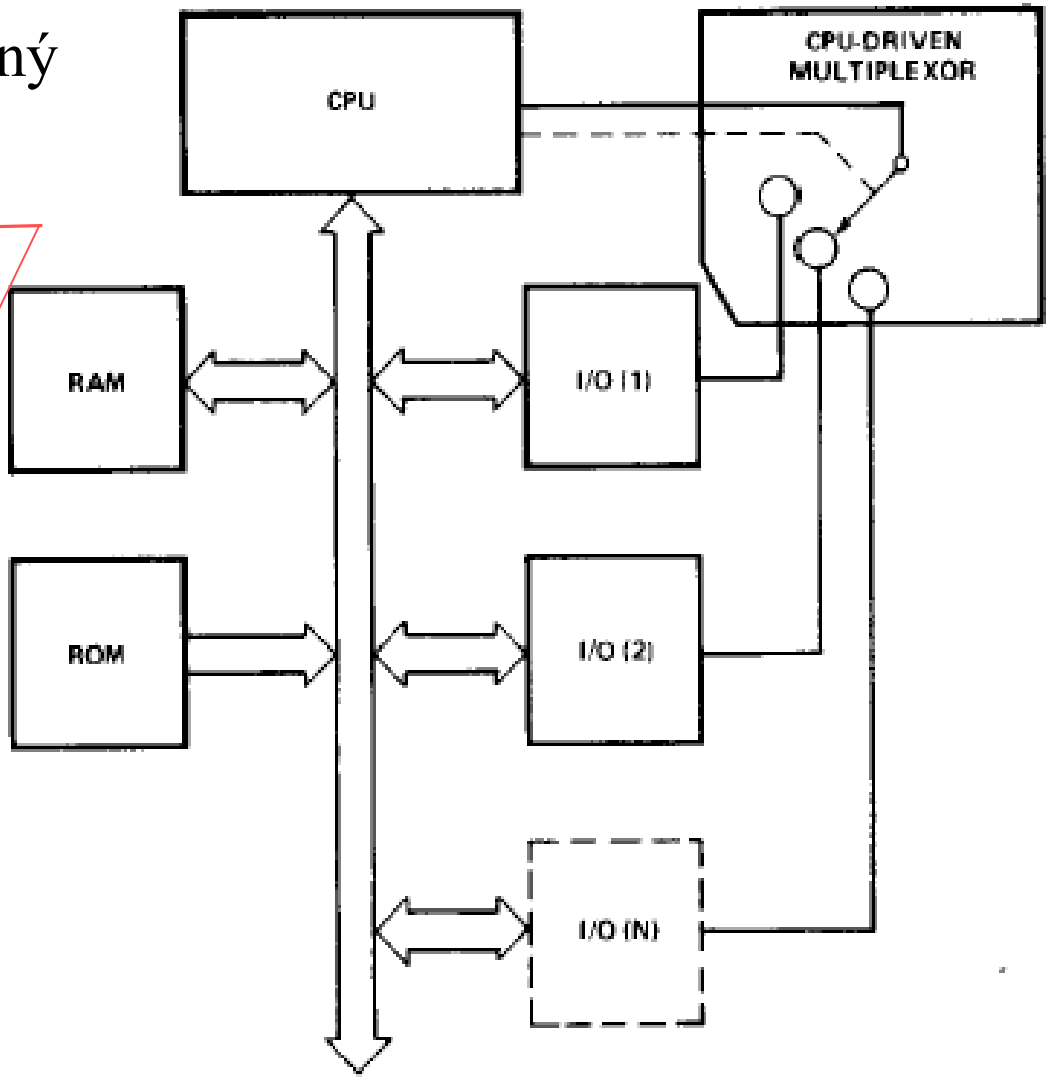
Polled Method/Polling – **dopytovacia metóda**
Interrupt Method – **prerušenia**

Bez účasti procesora

DMA – priamy prenos medzi pamäťou a perifériami

1. *Obsluha periférií:* ***Polled Method*** **– opytovacia metóda**

Spôsob komunikácie – protokol je riadený procesorom pomocou programu.
Nech I/O(2) má „pripravený“ znak pre CPU.
CPU adresuje I/O(2) a prečíta znak do ACU.
Atď.
Ak nie je rýchlosť CPU a I/O(2) rovnaká, CPU preberie znak niekoľko krát.
Riešenie je v použití:
Status registra (bitu).



Polling je rýchlejší, ak v najmenej 50% prípadoch dáta budú k dispozícii každú periódu pollingu **Low-latency Ethernet device polling By Jonathan Corbet** **May 21, 2013** https://lwn.net/Articles/551284/ Napr. pri dnešnom zaťažení siete

2. Obsluha periférií: ***Interrupt Method***

Niekedy je potrebné prerušiť beh programu, napr.: pokles napájania.

Prerušením behu programu môžeme tiež informovať CPU o ukončení udalosti: .

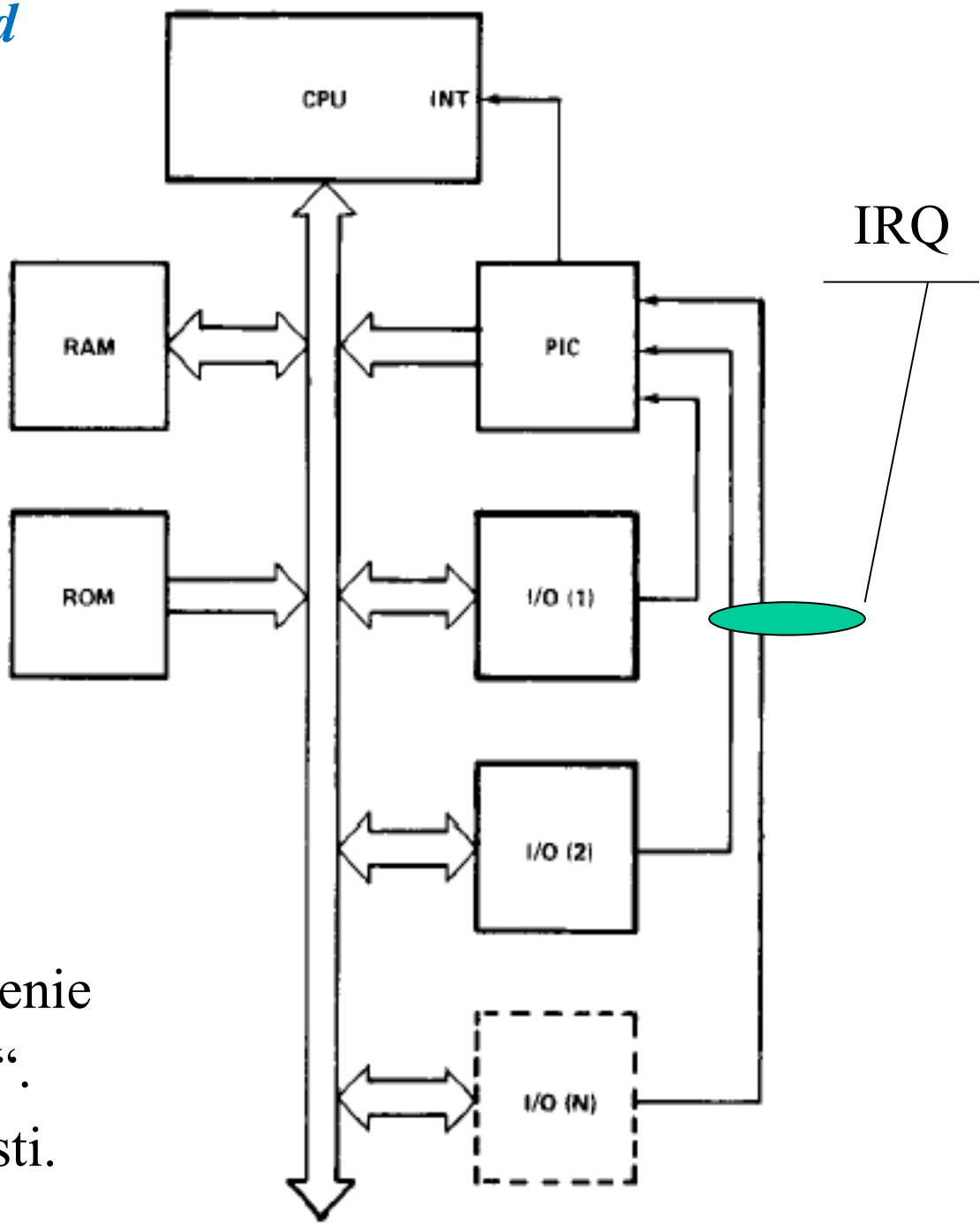
Napr odvysielanie znaku Cez USART.

Moderné CPU majú inštrukcie, ktoré sa správajú podobne ako hardvérové prerušenie.

T.j. Ak je prerušená činnosť CPU, riadenie sa dočasne odovzdá inému „programu“.

Po skončení sa vráti k pôvodnej činnosti.

Viacnásobné prerušenia: Priorita.



Typy prerušení

Prerušenia
– **hardvérové**
- **softvérové**

Hardvérové – vyvolané **vonkajšou udalosťou** z hľadiska procesora.
Časová vzdialenosť medzi udalosťami musí byť dostatočne veľká
Udalosti asynchrónne alebo periodické s nízkou frekvenciou

Softvérové – v inštrukčnom súbore sú inštrukcie na vyvolanie sekvencie, ako pri hardvérovom prerušení (využívajú, že začiatok obsluhy prerušenia je definovaný typom procesora).

Vybavenie procesora

- **vývody na požiadavky na prerušenie**
  - **RESET** – (nuluje stav procesora – počítača – signál býva distribuovaný aj na iné obvody mikropočítača), - nastavuje PC na preddefinovanú hodnotu, ale neodkladá návratovú adresu
  - **NMI** – (NonMasked/NonMaskable Interrupt) – prerušenie, ktoré sa nedá zakázať
  - **INT** - aspoň jeden vývod pre požiadavku na maskovateľné hardvérové prerušenie
  - **INTA** – (Interrupt Acknowledge) – vývod na signalizáciu povolenia prerušenia

- **schopnosťou realizovať prerušenie**
  - vedieť **zakázať/ povoliť** prerušenie programom
  - po dokončení vykonávanej inštrukcie odložiť **návratovú adresu (zásobník)**
  - odložiť **stavové slovo** (stavový register + akumulátor= register výsledkov) – nie je nutné
- **zakázať** ďalšie prerušenie

Typy prerušovacích signálov

-**statické** – požiadavka musí byť signalizovaná trvalou dohodnutou logickou úrovňou až do akceptácie prerušenia
-**dynamické** - požiadavka je signalizovaná impulzom s definovanou minimálnou šírkou, jedna z hrán je považovaná za signál požiadavky
-typ signálu môže byť **programovateľný**

Obsluha prerušenia

- má atribúty **podprogramu**
- používa špeciálne inštrukcie na **návrat z obsluhy prerušenia**
- musí byť **transparentná** – musí uchovať všetky registre, ktoré používa vrátane stavového registra
- obsluha hardvérového prerušenia **nemá parametre**, softvérového môže mať parametre – obvykle len registrové

Vlastnosti prerušovacieho systému

Prerušovací systém

- **nevektorový** – na INT sa pripája logický súčet požiadaviek na prerušenie
  (má charakter dopytovacej metódy I/O, obslužný podprogram zisťuje
  zdroj prerušenia)

-**vektorový** – systém zabezpečuje, že do **PC** sa vloží adresa obsluhy prerušenia
pre daný zdroj prerušenia, ktorý bol povolený
(vektory prerušenia – pevné adresy podprogramov AVR, alebo tabuľka
adries podprogramov Ix86) – realizácia radičmi prerušení

Prioritný systém

- **časový** – požiadavky na obsluhu sa vybavujú v poradí
  v akom boli vygenerované
- **úrovňový** – každý zdroj prerušenia má definovanú prioritu

Úrovňový prioritný systém

- **pevný** – priorita sa nedá meniť
- **rotujúci** – po každej obsluhe prerušenia sa cyklicky mení priorita
- **programovateľný**
  - ako cyklický, len prvý môže byť definovaný ľubovolný zdroj
  - možnosť povolenia **dokončenia obsluhy s nižšou prioritou**

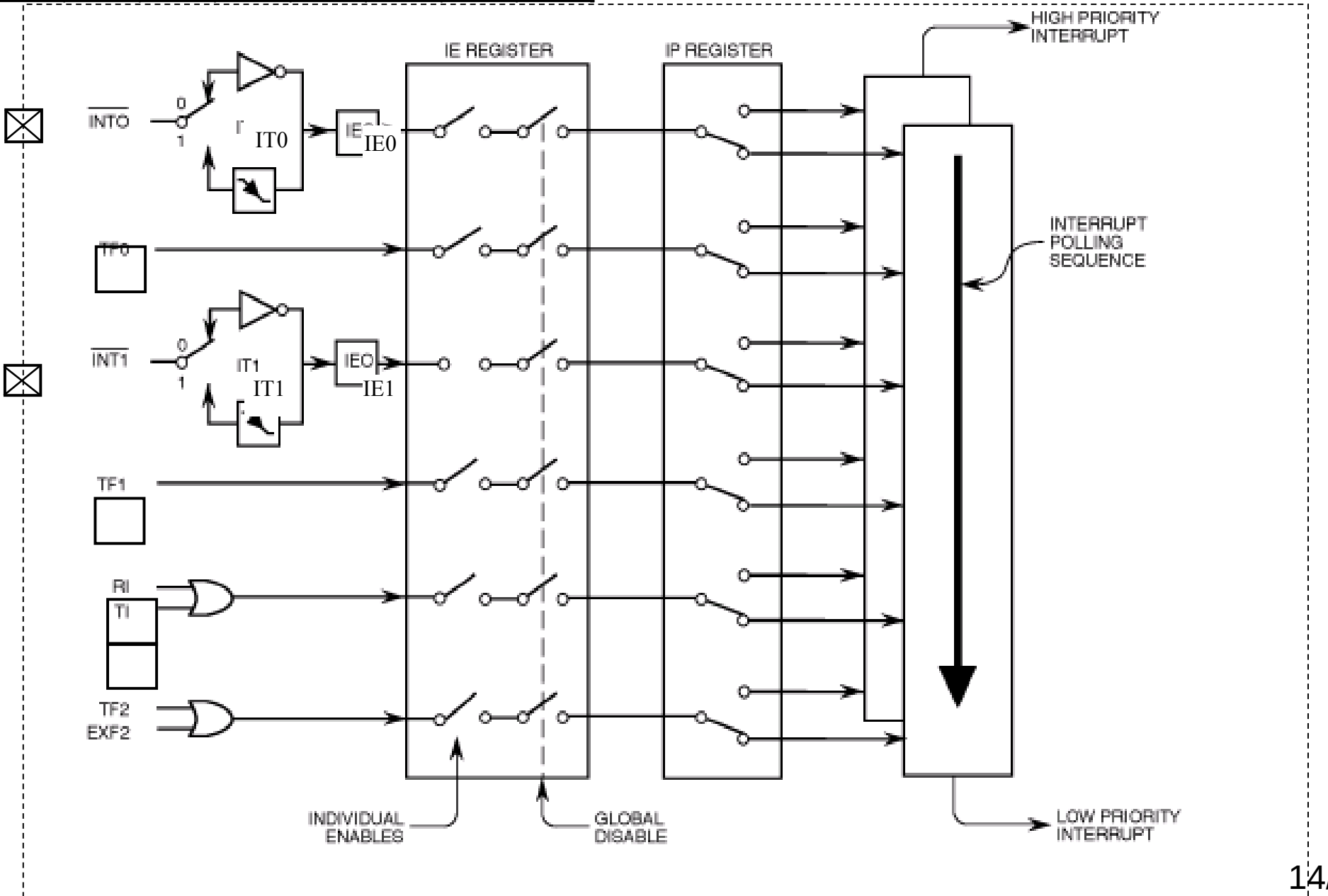
Úrovňový prioritný systém

- **jednoúrovňový** (môže v jednom čase bežať len jedna obsluha prerušenia)
- **vnorený** (môže v jednom čase viacero obslúh prerušení)
- **úplný vnorený** – teoreticky môžu bežať všetky obsluhy prerušení
  (ale každá iba raz)

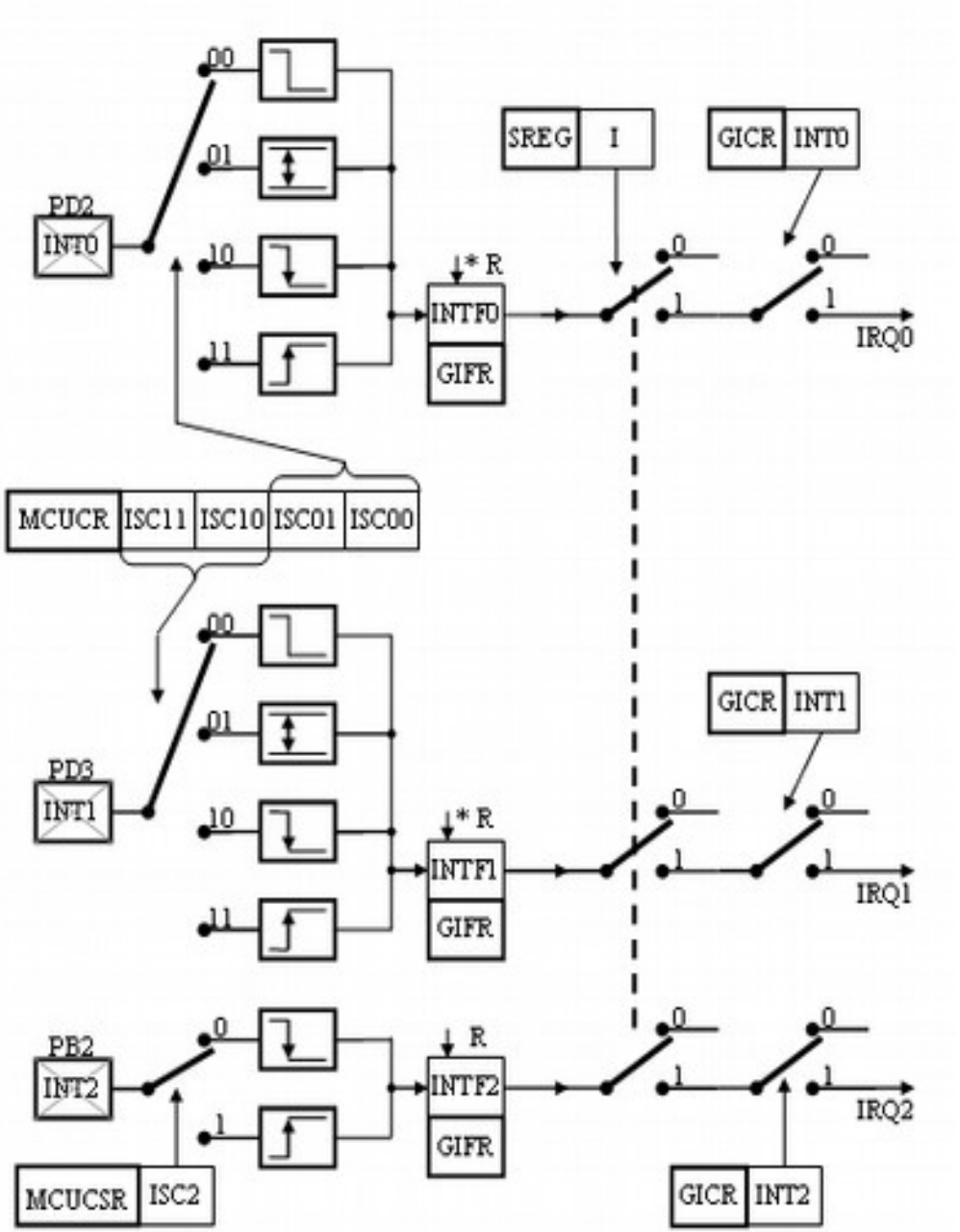
Prerušovací podsystém

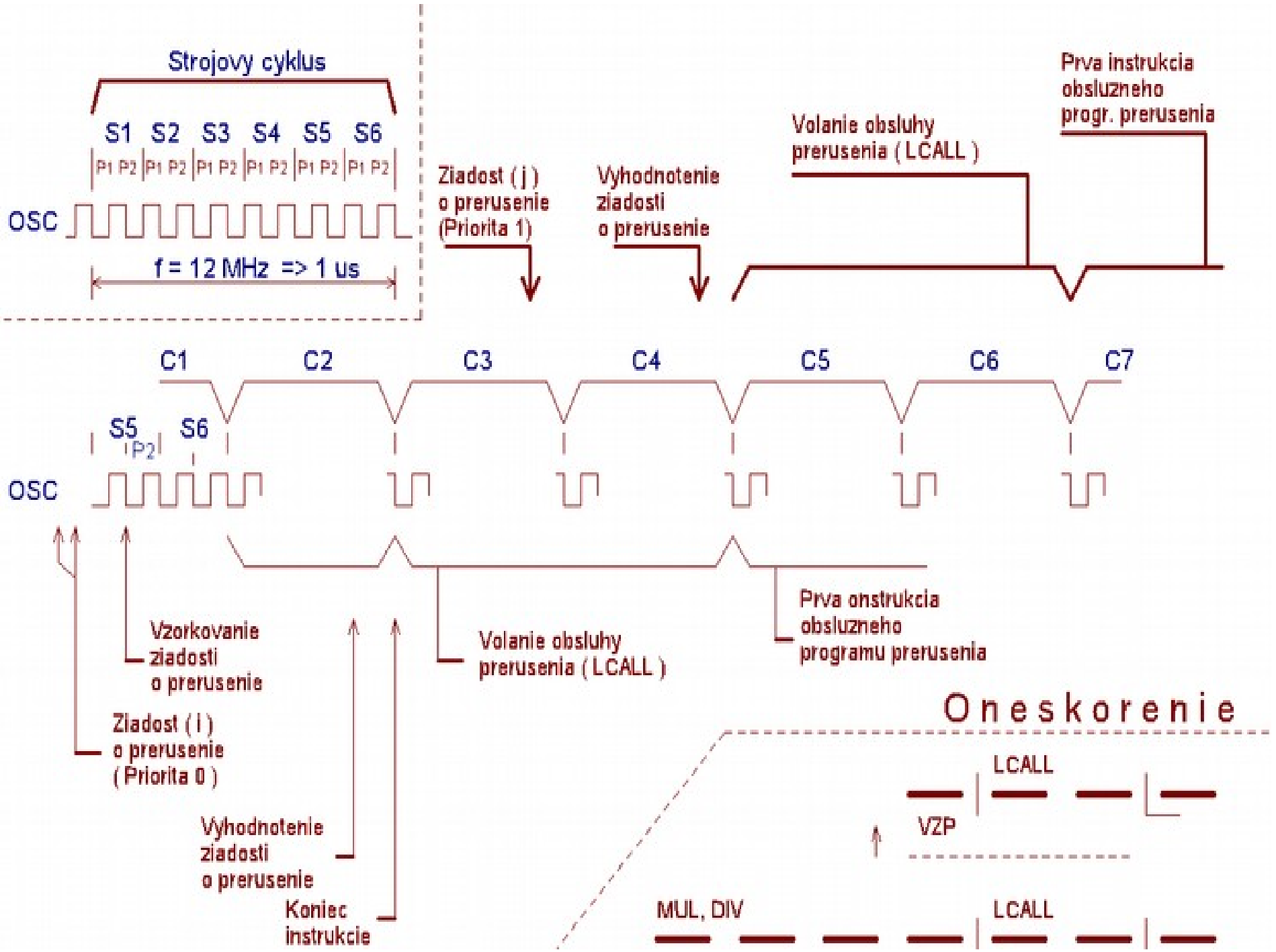
TCON = 88H

				IE1	IT1	IE0	IT0



Prerušovací podsystém




Strojovy cyklus
S1 S2 S3 S4 S5 S6
P1 P2 P1 P2 P1 P2 P1 P2 P1 P2 P1 P2
OSC
f = 12 MHz => 1 us
Ziadost (j)
o prerusenie
(Priorita 1)
Vyhodnotenie
ziadosti
o prerusenie
Volanie obsluhy
prerusenia (LCALL)
Prva instrukcia
obsluzneho
progr. prerusenia
C1 C2 C3 C4 C5 C6 C7
S5 S6
P2
OSC
Vzorkovanie
ziadosti
o prerusenie
Ziadost (l)
o prerusenie
(Priorita 0)
Vyhodnotenie
ziadosti
o prerusenie
Koniec
instrukcie
Volanie obsluhy
prerusenia (LCALL)
Prva onstrukcia
obsluzneho
programu prerusenia
Oneskorenie
LCALL
VZP
MUL, DIV
LCALL

Interrupt Priority	Vector No.	Program Address	Source	Interrupt Definition
High	1	$000	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	TIMER2 COMP	Timer/Counter2 Compare Match
	5	$008	TIMER2 OVF	Timer/Counter2 Overflow
↑	6	$00A	TIMER1 CAPT	Timer/Counter1 Capture Event
|	7	$00C	TIMER1 COMPA	Timer/Counter1 Compare Match A
|	8	$00E	TIMER1 COMPB	Timer/Counter1 Compare Match B
|	9	$010	TIMER1 OVF	Timer/Counter1 Overflow
|	10	$012	TIMER0 OVF	Timer/Counter0 Overflow
|	11	$014	SPI, STC	Serial Transfer Complete
|	12	$016	USART, RXC	USART, Rx Complete
|	13	$018	USART, UDRE	USART Data Register Empty
|	14	$01A	USART, TXC	USART, Tx Complete
|	15	$01C	ADC	ADC Conversion Complete
↓	16	$01E	EE_RDY	EEPROM Ready
	17	$020	ANA_COMP	Analog Comparator
	18	$022	TWI	Two-wire Serial Interface
	19	$024	INT2	External Interrupt Request 2
	20	$026	TIMER0 COMP	Timer/Counter0 Compare Match
Low	21	$028	SPM_RDY	Store Program Memory Ready

- ***Vonkajšie*** **– asynchrónne (hardvérové).Vyvolané hardvérovým zariadením, napr. I/O kartou.**

µC(keď nemáme grécke písmená tak uC) má dva vstupy:

- Nemaskovateľné prerušenie (Non Mascable Interrupt)

Vstup procesora: NMI

Využíva sa na signalizáciu :

- poklesu napájania,

- chyby parity

Toto prerušenie sa nedá zakázať.

- Maskovateľné prerušenie (Mascable Interrupt)

vstup procesora: INTR,

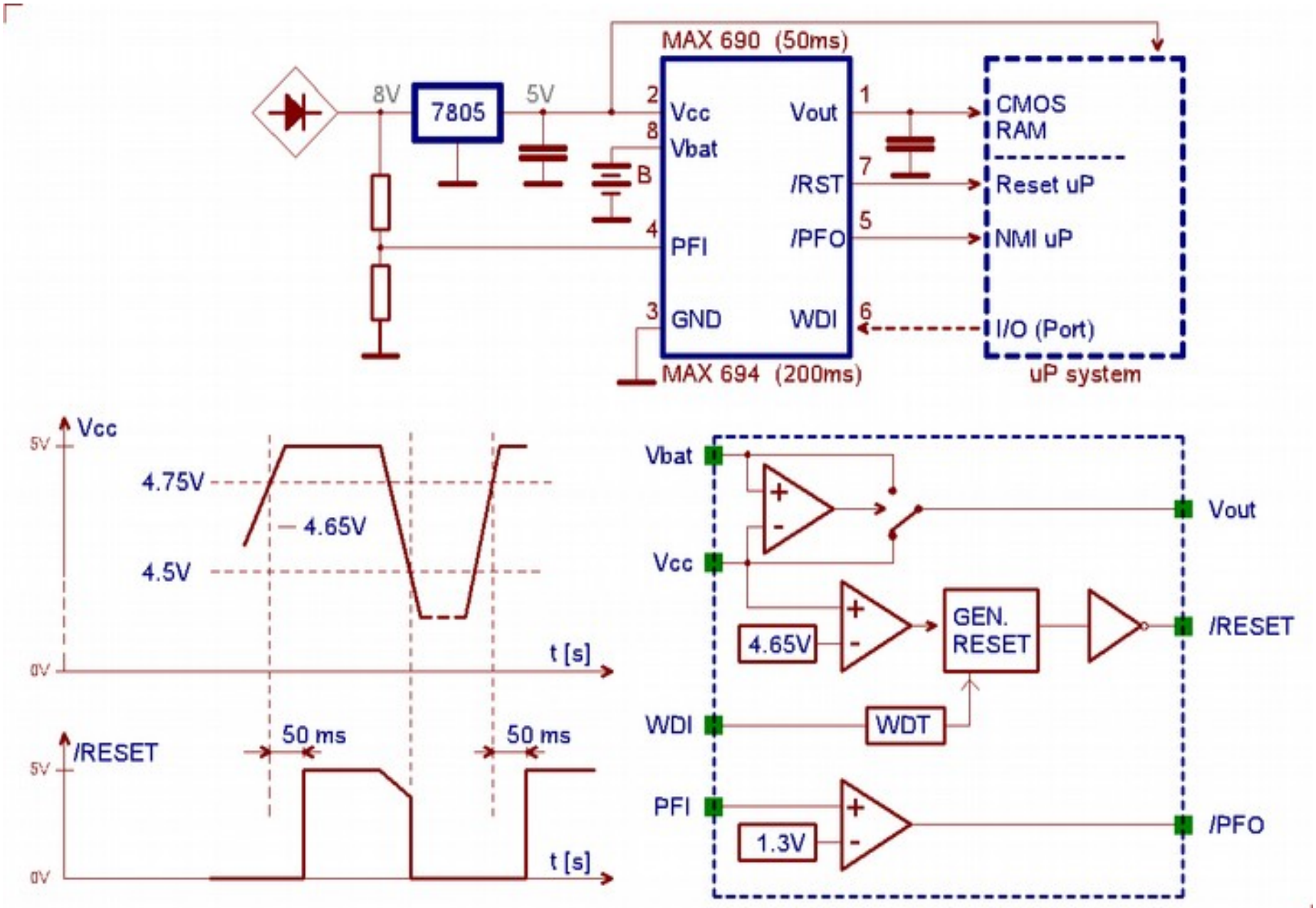
aktivovaný požiadavkou o prerušenie IRQ
Cez externý obvod PIC (Priority Interrupt Controller napr. i8259) Obsluhu tohto prerušenia možno programovo zakázať vynulovaním bitu **IF** (Interrupt Flag), resp. povoliť nastavením tohto bitu.

- ***Vnútorné* – synchrónne**
  - **Softwarové. *Cielené* s**púšťané inštrukciou **INT.**
  - **Výnimky** (Exceptions)
    - **Fault:** táto výnimka sa vyvolá skorej ako sa dokončí príkaz.

      Do zásobníka sa uloží obsah čítača inštrukcií EIP/IP – adresa príkazu, ktorý výnimku vyvolal. Napr. Ak bol výnimkou segment neprítomný v pamäti, procesor ho môže načítať a inštrukciou IRET už v pamäti existujúci príkaz vykonať.
    - **Trap: vyvolá výnimku potom ako sa vykoná** (správne vykoná). Do zásobníka sa uloží adresa nasledujúcej inštrukcie.

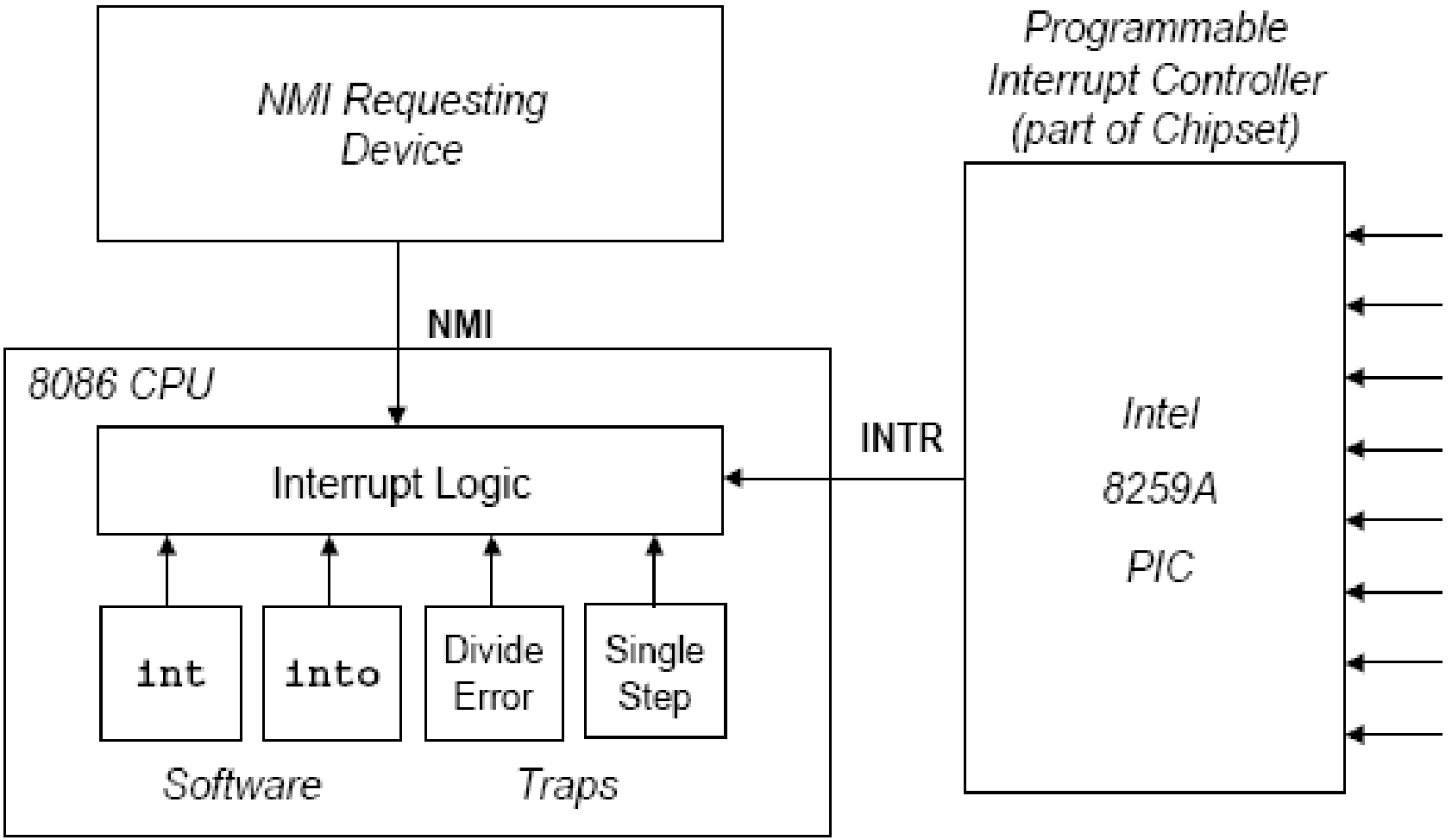
      Typickým príkladom je ladenie programu pomocou debuggeru/odvšivovača. Program sa „zastaví“ a debugger kontroluje obsahy registrov.
    - **Abort:** Táto výnimka väčšinou neposkytuje adresu miesta vzniku chyby. Signalizuje sa napr.: chyba hardvéruru, alebo neplatnosť systémových tabuliek.

NMI, RST


MAX 690 (50ms)
8V
7805
5V
2
Vcc
8
Vbat
B
4
PFI
3
GND
Vout
/RST
/PFO
WDI
1
7
5
6
CMOS RAM
Reset uP
NMI uP
I/O (Port)
uP system
MAX 694 (200ms)
Vcc
5V
4.75V
4.65V
4.5V
0V
t [s]
/RESET
50 ms
50 ms
Vbat
Vcc
4.65V
GEN. RESET
WDT
WDI
PFI
1.3V
/RESET
/PFO

8086 Interrupt Connections

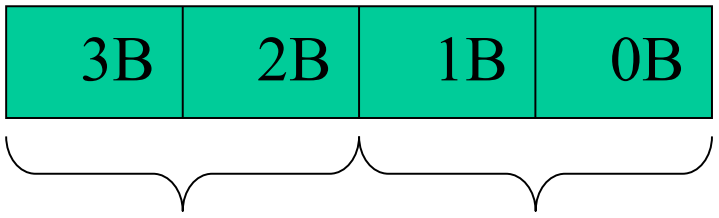
NMI - *Non-Maskable Interrupt* **INTR** - *Interrupt Request*



Prerušenia a architektúra procesorov INTEL

Procesor i8086(i80186,i80286… Pentium/K5=i586..) rozlišuje 256 rôznych prerušení.

31	0	Adresa	Čis.Pr.vekt.
Segment	Offset	0:03FC	INT 0FFh
Segment	Offset	0:000C	INT 3
Segment	Offset	0:0008	INT 2
Segment	Offset	0:0004	INT 1
Segment	Offset	0:0000	INT 0



V pamäti uložené ako:

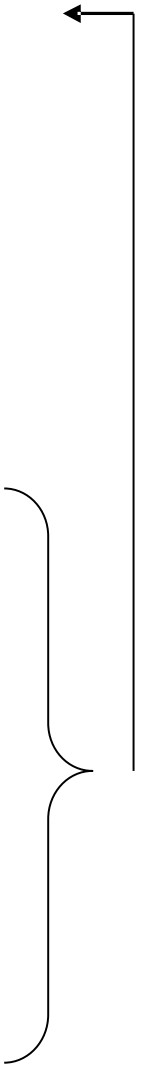
0000: 0B

0001: 1B

0002: 2B

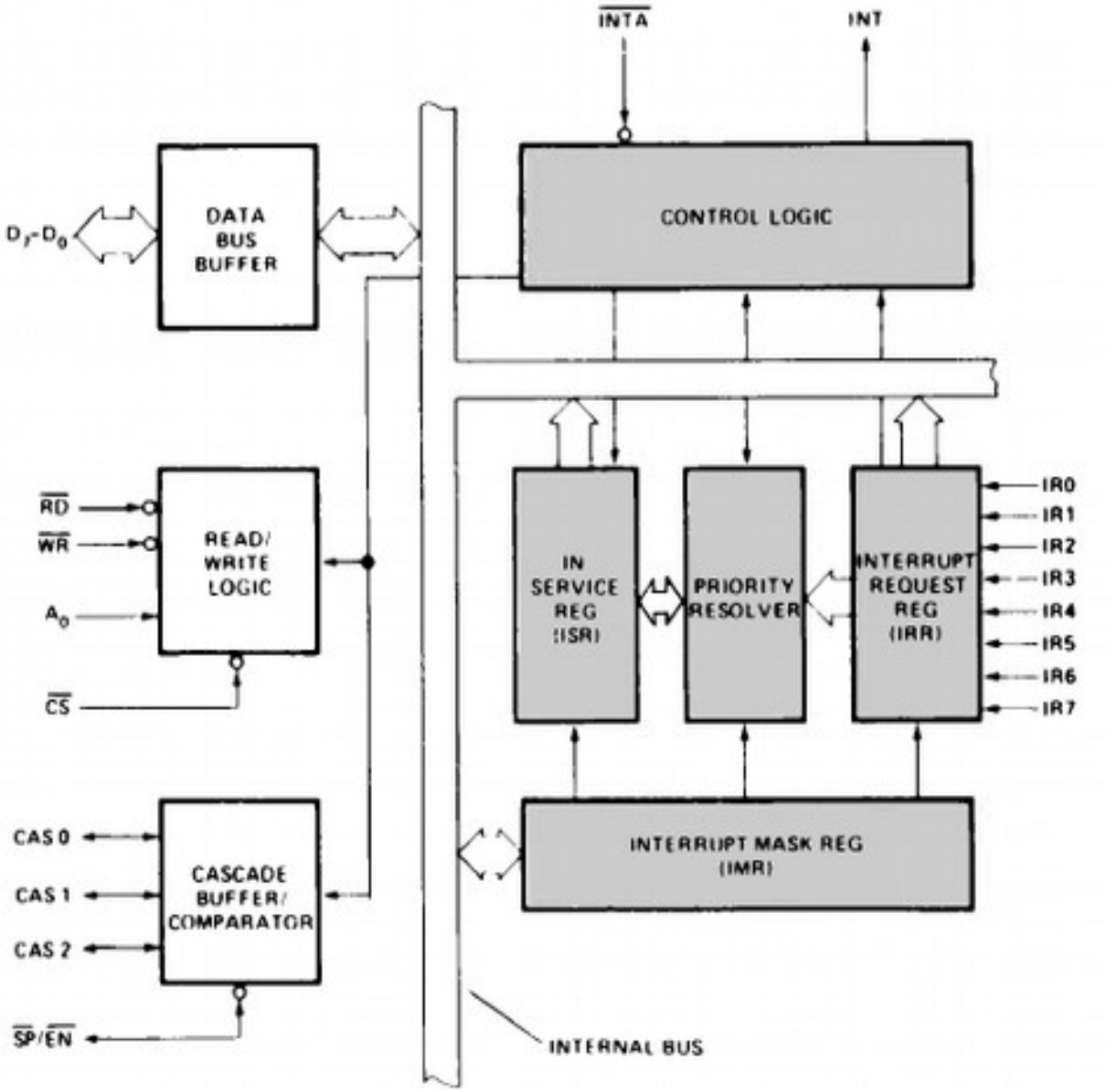
0003: 3B

Typ INT	Adresa INT vect.	Typ IRQ	Význam v PC AT	
00 H – 01H			Delenie nulou, krokovanie programu	Exceptions
02H		NMI		**Hardware**
03H - 07H				Exceptions
08H	20H	IRQ0	HW časovač 18.2Hz	Hardware
09H	24H	IRQ1	Klávesnica	Hardware
0AH	28H	IRQ2	Do kaskády zapojený 2. PIC	Hardware
0BH	2CH	IRQ3	COM2	Hardware
0CH	30H	IRQ4	COM1	Hardware
0EH	38H	IRQ6	Disketa	Hardware
0DH	34H	IRQ5	LPT2	Hardware
0FH	3CH	IRQ7	LPT1	Hardware
10H – 6FH			Softwarové prerušenia	
70H	1C0H	IRQ8	Real Time Clock – RTC	Hardware
71H	1C4H	IRQ9		Hardware
72H	1C8H	IRQ10		Hardware
73H	1CCH	IRQ11		Hardware
74H	1D0H	IRQ12		Hardware
75H	1D4H	IRQ13	Math's Co-Processor	Hardware
76H	1D8H	IRQ14	Radič HardDisk	Hardware
77H		IRQ15		Hardware
78H - FFH	1DCH		Softwarové prerušenia	



PIC Programmable Interrupt Controller - Programovateľný radič prerušení
Dnes sa používa APIC=Advanced PIC

Dokáže riadiť až osem zdrojov prerušenia **IRQ** $_i$. Prideľovať prioritu a lokálne povoliť, resp. zakázať jednotlivé prerušenie….



Kaskádne zapojenie:
Max.64 IRQ

PC XT : 1* PIC
PC AT : 2* PIC

PIC – 8259A/8259A-2:

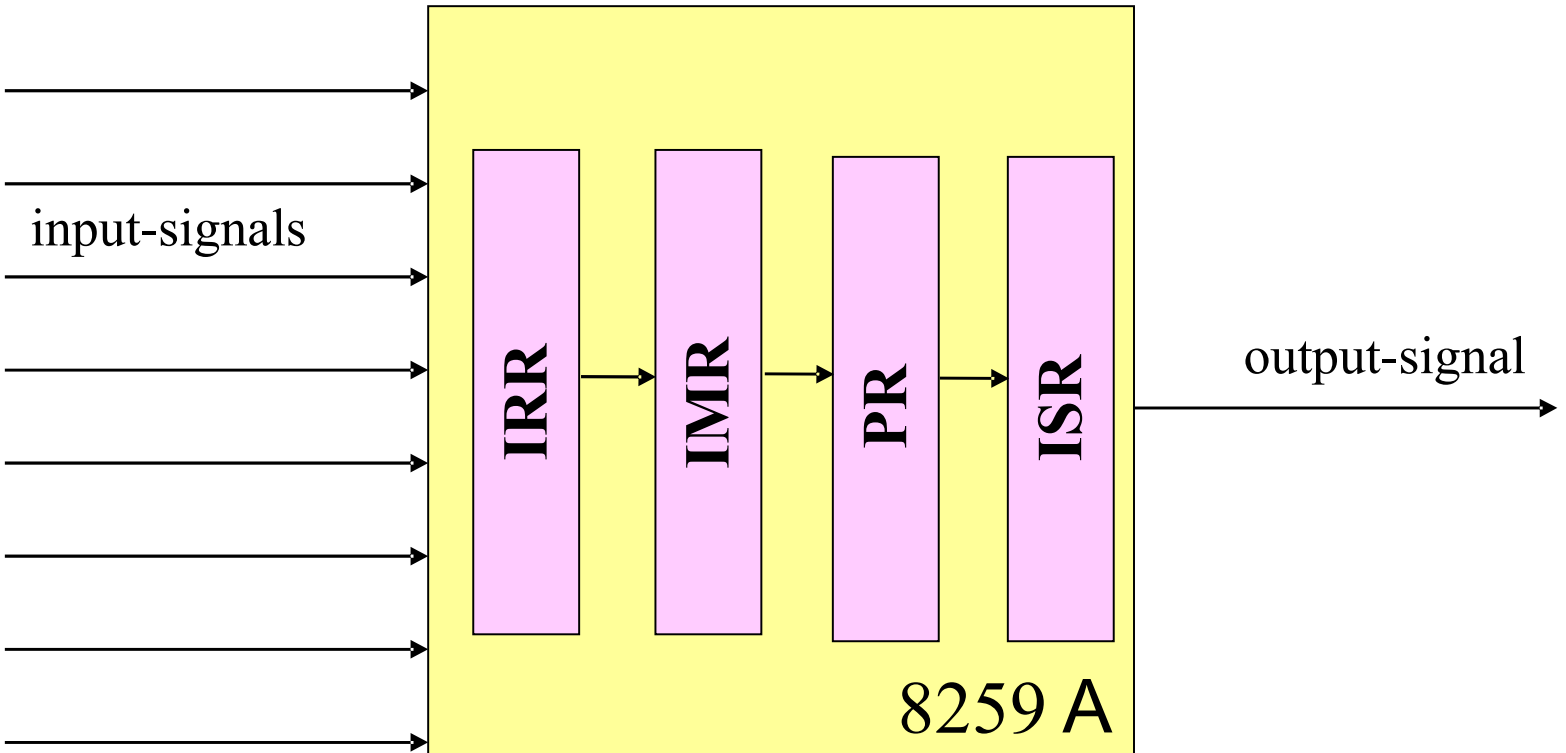
- ***IRR*** (Interrupt request register) ***- register požiadaviek IRQ***

-***IMR*** (Interrupt mask register) ***- register masiek požiadaviek IRQ***

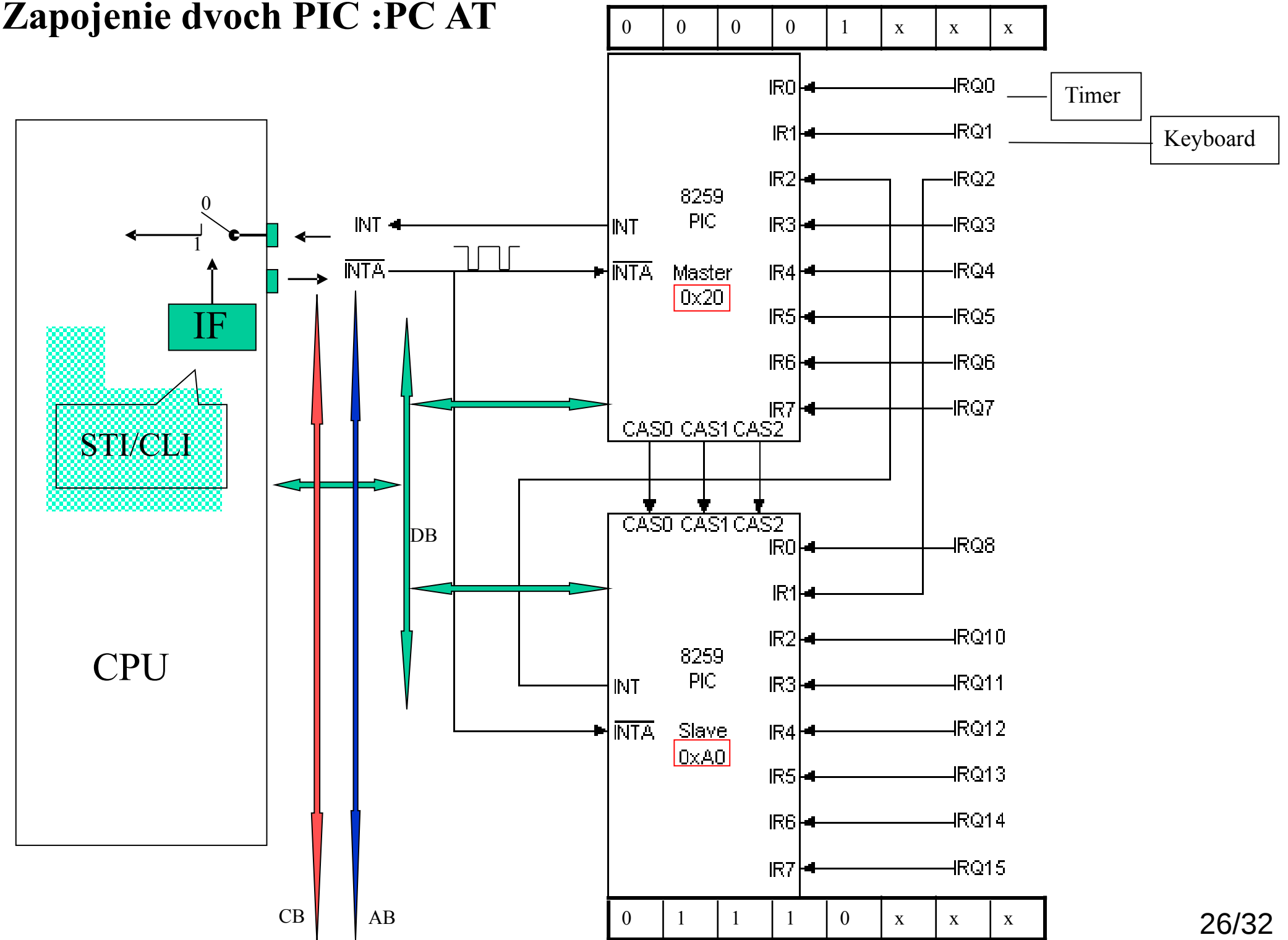
Nastavený bit zakazuje odpovedajúce prerušenie
a vynulovaný bit povoluje odpovedajúce prerušenie.

- ***ISR*** (In/Interrupt service register) ***- stavový register požiadaviek na obsluhu prerušení .*** *Obsah tohoto registra vyhodnocuje procesor ako požiadavku o prerušenie.*

Priority coder (priority resolver - PR) - ***logika vyhodnocovania priorít***



Zapojenie dvoch PIC :PC AT

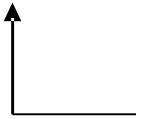


Ovládanie obvodov PIC:

1. Inicializácia, podstatnú časť robí BIOS/EFI
2. Pomocou OCW1 vykonáme:

Povolenie, resp. zakázanie prerušenia:

Povolenie **IRQ3:** `outportb(0x21,inportb(0x21) & 0xF7 );`



Bázová adresa+1

Zakázanie **IRQ3:** `outportb(0x21,inportb(0x21)| 0x08 );`

Microsoft Windows a Visual Studio,najskôr kam,potom čo

```
static void outportb(int portNum, int val)
```

Unixy (MAC OS X,Linux)- najskôr hodnota, potom kam

```
void outb(unsigned char value, unsigned short int port);
unsigned char inb(unsigned short int port);
```

3. Zrušenie príznaku akceptovaného prerušenia s najvyššou prioritou

`outportb(0x20,0x20)` – End of Interrupt (EOI) pre PIC1

`outportb(0xA0,0x20)` – EOI pre PIC2

Obsluha prerušenia - Interrupt service routine (ISR) 1

1. **Vznik požiadavky- IRQ** (Interrupt Request)
   - IRQ vyvolané hranou (*nábežnou)*
   - IRQ vyvolané úrovňou (*log. 1, ?ako dlho?)*
   - zapamätanie + vyhodnotenie priority.
2. Pre najvyššiu prioritu generuje obvod PIC signál do procesora **INTR.** Vetvenie vyhodnocovania: IF = 0 alebo IF = 1. Ak je povolené, dokončí sa rozpracovaná inštrukcia a procesor pokračuje bodom 3.
3. Procesor pošle dva INTA (*Interrupt Acknowledge),*
4. Po dátovej zbernici vyšle **PIC** osem bitovú hodnotu – typ prerušenia – INT_vector. PraktickyINT8 až INT15 a INT112 až INT119.
5. Vyvolá sa obsluha prerušenia ISR.
   – Zakáže sa prerušenie.
   – Do zásobníka sa uloží obsah registra príznakov.
   – Do zásobníka sa uloží návratová adresa (Segment:Offset)
   – 4*INT_Vector => IP, CS = adresa ISR
   –Uloženie dôležitých registrov.

Obsluha prerušenia - Interrupt service routine (ISR) 2

6. „Vlastná obsluha prerušenia“ alebo naplánovanie vlákna s ňou
7. Vynulovanie IRQ. Oznámenie obvodu PIC ukončenie prerušenia.
   - „Obnova“ registrov.
   - „Vynulovanie“ žiadosti o prerušenie na úrovni hardvéru periférie.
8. Inštrukcia IRET (povolenie prerušenia). Iným spôsobom povolenia prerušenia je hneď po vstupe do ISR použiť inštrukciu STI.

Moving interrupts to threads By Jake Edge October 8, 2008
https://lwn.net/Articles/302043/
Processing interrupts from the hardware is a major source of latency in the kernel, because other interrupts are blocked while doing that processing. For this reason, the realtime tree has a feature, called threaded interrupt handlers, that seeks to reduce the time spent with interrupts disabled to a bare minimum—pushing the rest of the processing out into kernel threads. But it is not just realtime kernels that are interested in lower latencies, so threaded handlers are being proposed for addition to the mainline

Block-layer I/O polling November 11, 2015
https://lwn.net/Articles/663879/

- I/O polling made no sense for the block layer as long as storage was dominated by rotating media.
- Solid-state drives are different, though; I/O completion times are tiny and even a low-end drive can complete huge numbers of operations per second. With such a drive, the case for doing some other work while waiting for an I/O completion interrupt is rather weaker.

Linux 4.4 has been released on Sun, 10 Jan 2016.
http://kernelnewbies.org/Linux_4.4

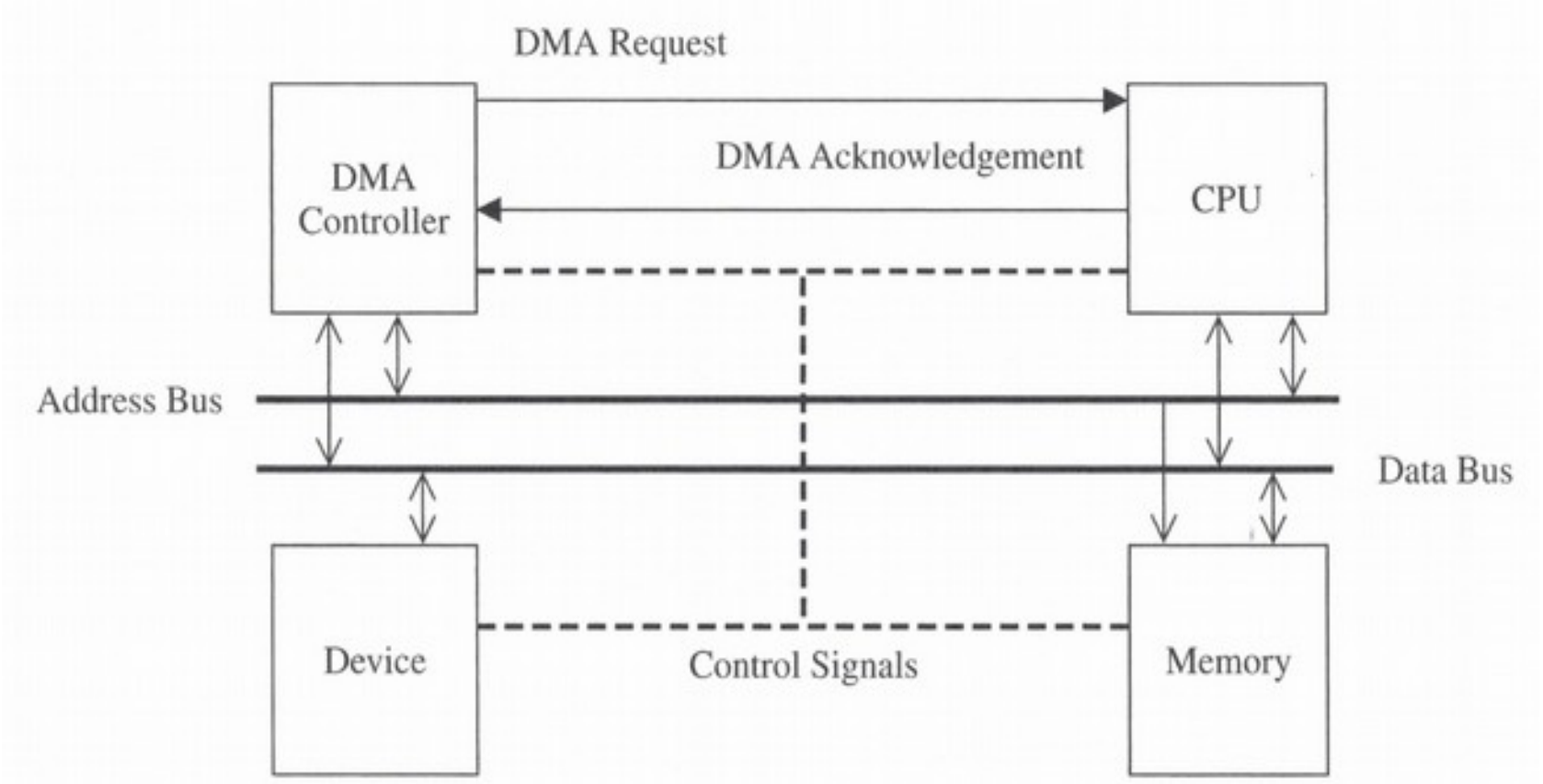
- Prominent features
  - Block polling support

tthis release adds basic support for polling for specific IO to complete, which can improve latency and throughput in very fast devices. This support is only intended for testing, in future releases stats tracking will be used to auto-tune this. For now, for benchmark and testing purposes, we add a sysfs file (io_poll) that controls whether polling is enabled or not.

3. **Direct Memory Access** (DMA) – priamy prenos medzi pamäťou a periférami. Väčšina zariadení má „pridelený“ DMA kanál.

Dôvod zavedenia DMA: Vynechať CPU ako sprostredkovateľa pri prenose dát medzi operačnou pamäťou a periférami. Ak umožníme priamy prenos: periféria ⟷ pamäť, CPU môže vykonávať inú činnosť.

DMA – hardware riadiaci prenos medzi pamäťou a I/O zariadeniami.



Prenos DMA kanálom sa môže realizovať ako:

– prenos po slovách.

– prenos celého bloku.

Riadenie DMA prenosu:

1. DMA controller je inicializovaný CPU.
   – Nastaví sa počiatočná adresa
     – „odkiaľ“, „kam“ a
     – počet prenášaných údajov.

   Mód prenosu sa nastavuje do riadiaceho registra.
2. Dáta sa presúvajú (adresa pamäte sa inkrementuje, a presunie sa určité množstvo bytov, slov).
3. Keď počítadlo prenesených údajov dosiahne nulu, DMA vyvolá prerušenie.
4. CPU prevezme riadenie nad prístupom k pamäti.