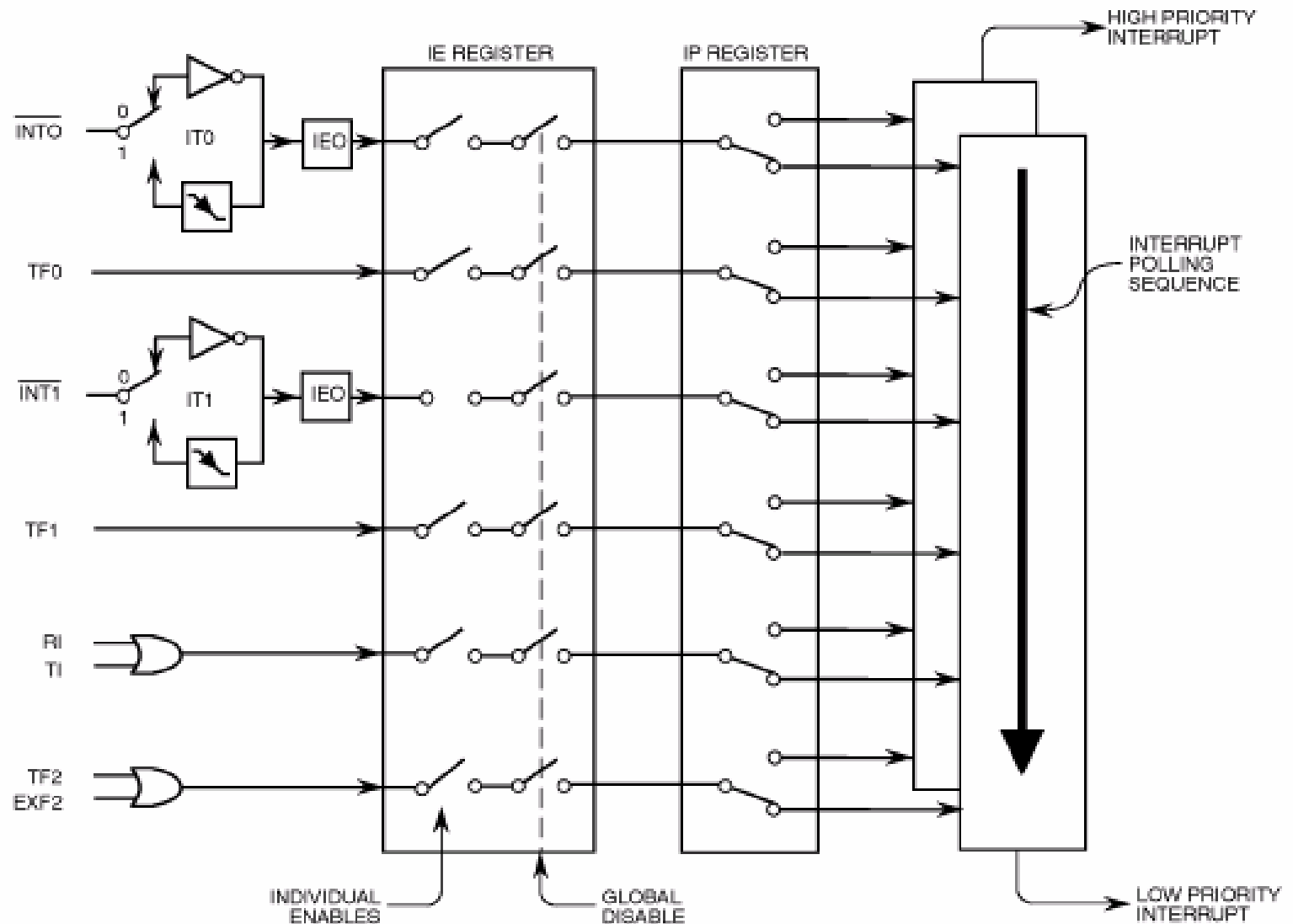


Prerušovací podsystem



Vstupno/výstupný podsystem (Riadenie - obsluha periférií)

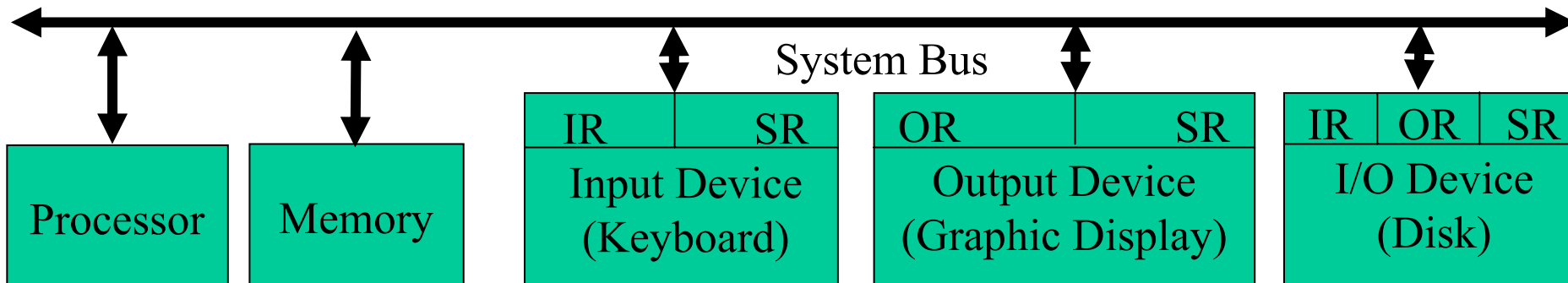
Základný charakteristický znak periférnych obvodov:

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: => *Interrupt*.

Opačný proces možno badať smerom k displayu, resp. k tlačiarňi..

- Laserová tlačiareň: cca 100 000 znakov/sek,
- Grafický display: cca 30 000 000 znakov/sek,



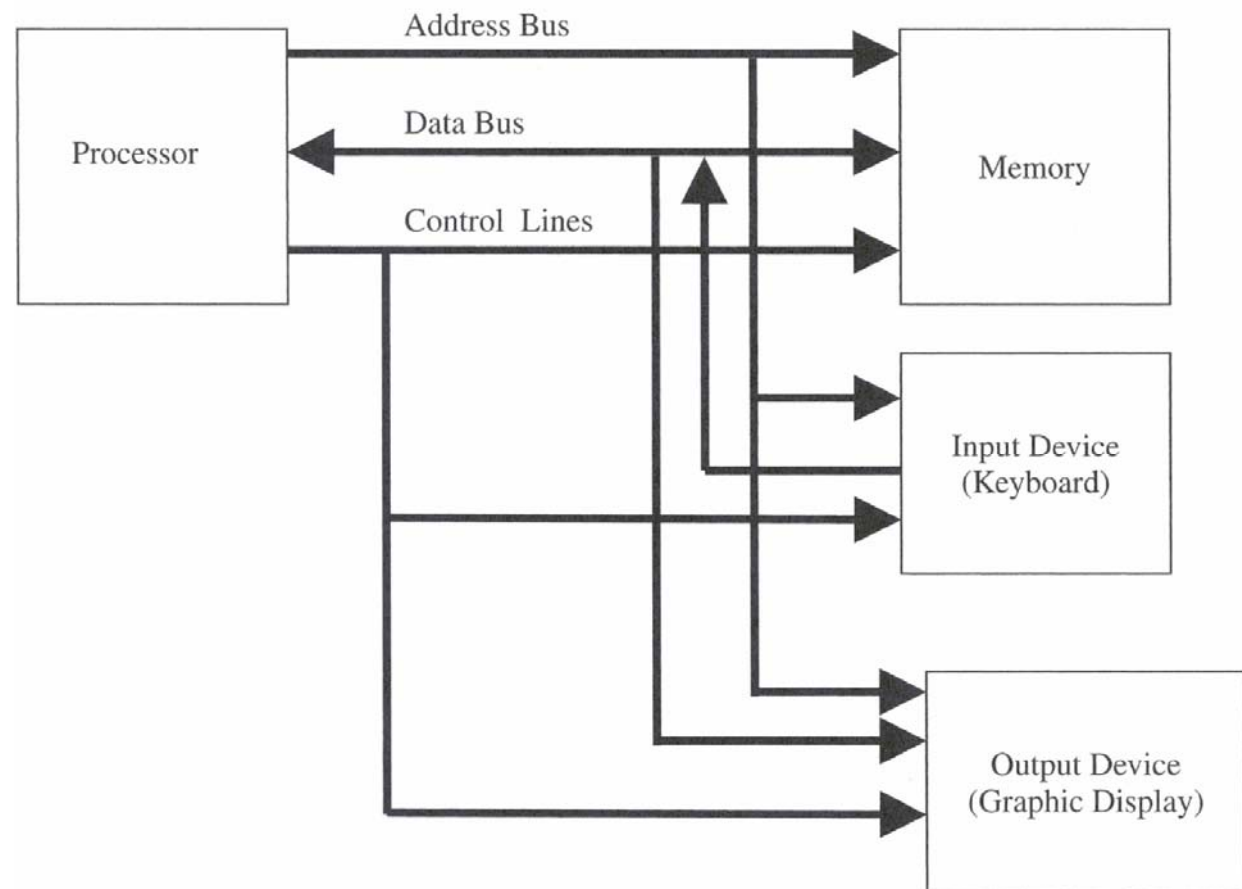
Vstupno/výstupný podsystem (Riadenie - obsluha periférií)

- 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*.

Mapovanie I/O obvodov, I/O Mapping

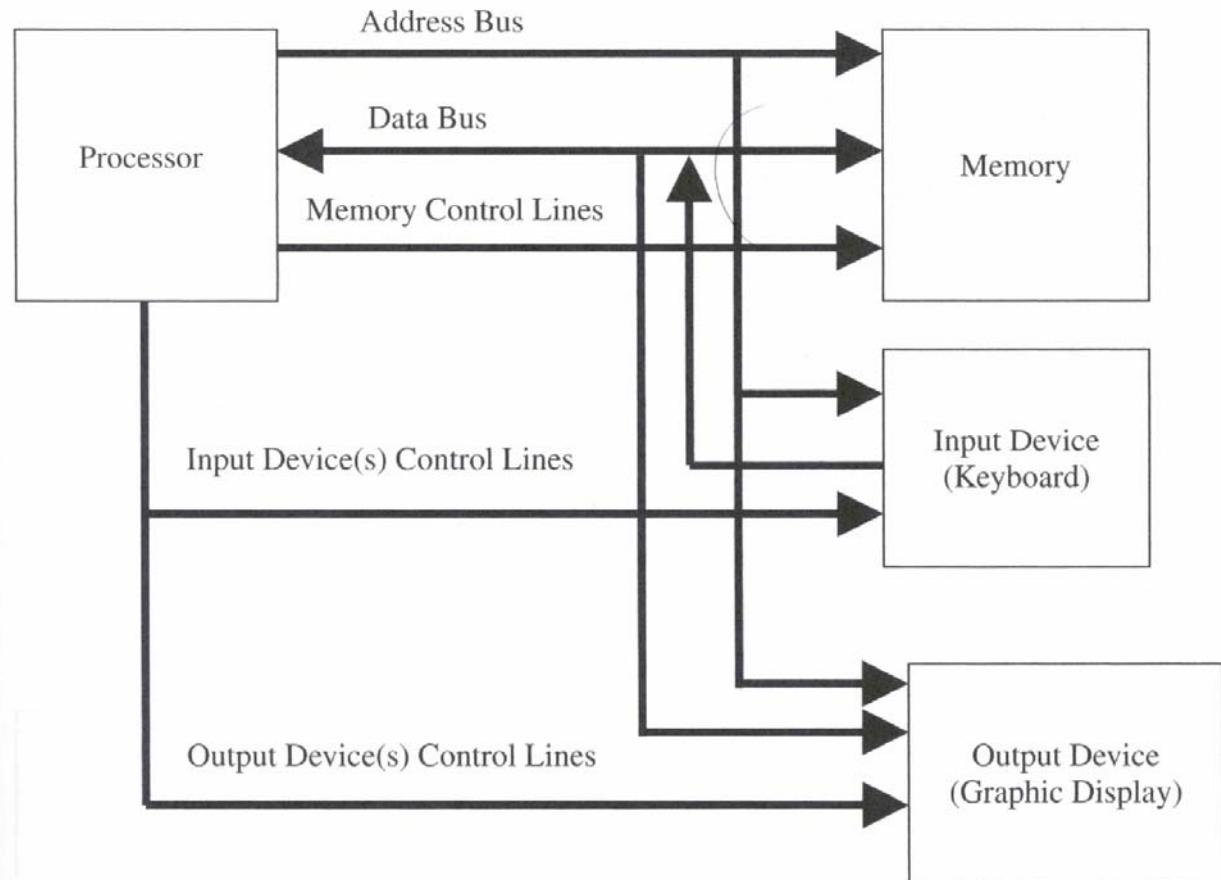
Z uvedeného je zrejmé že procesor musí vedieť príslušné I/O zariadenia adresovať.

- **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).

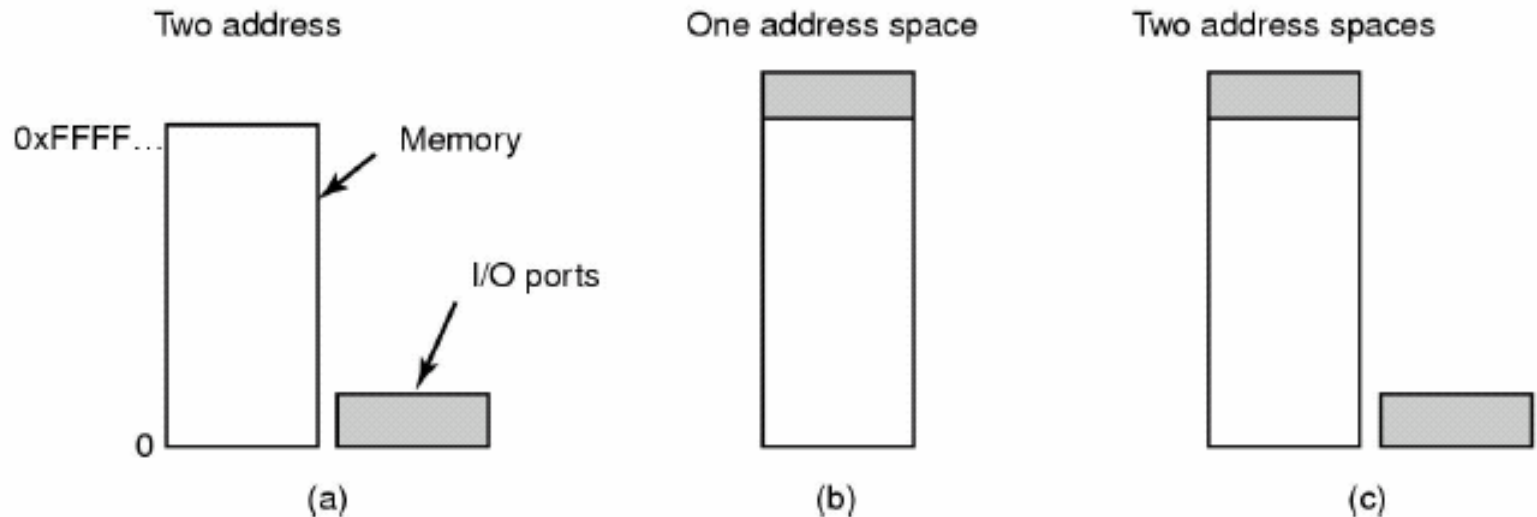


Mapovanie I/O obvodov, I/O Mapping

- **Samostatné mapované I/O zariadenia**
 - Oddelený adresný priestor
 - Špeciálne vodiče pre výber I/O a pamäť
 - Špeciálne príkazy I/O
 - (Intel)



Mapovanie I/O obvodov, I/O Mapping



- Separate I/O and memory space PC
- Memory-mapped I/O 8051
- Hybrid AVR

Tri základné spôsoby komunikácie s I/O podsystemom:

Polled Method – opytovacia metóda

Interrupt Method – prerušenia

DMA – priamy prenos medzi pamäťou a perifériami

1. Obsluha periférii: Polled Method – opytovacia metóda

Spôsob komunikácie – protokol je riadený procesorom pomocou programu.

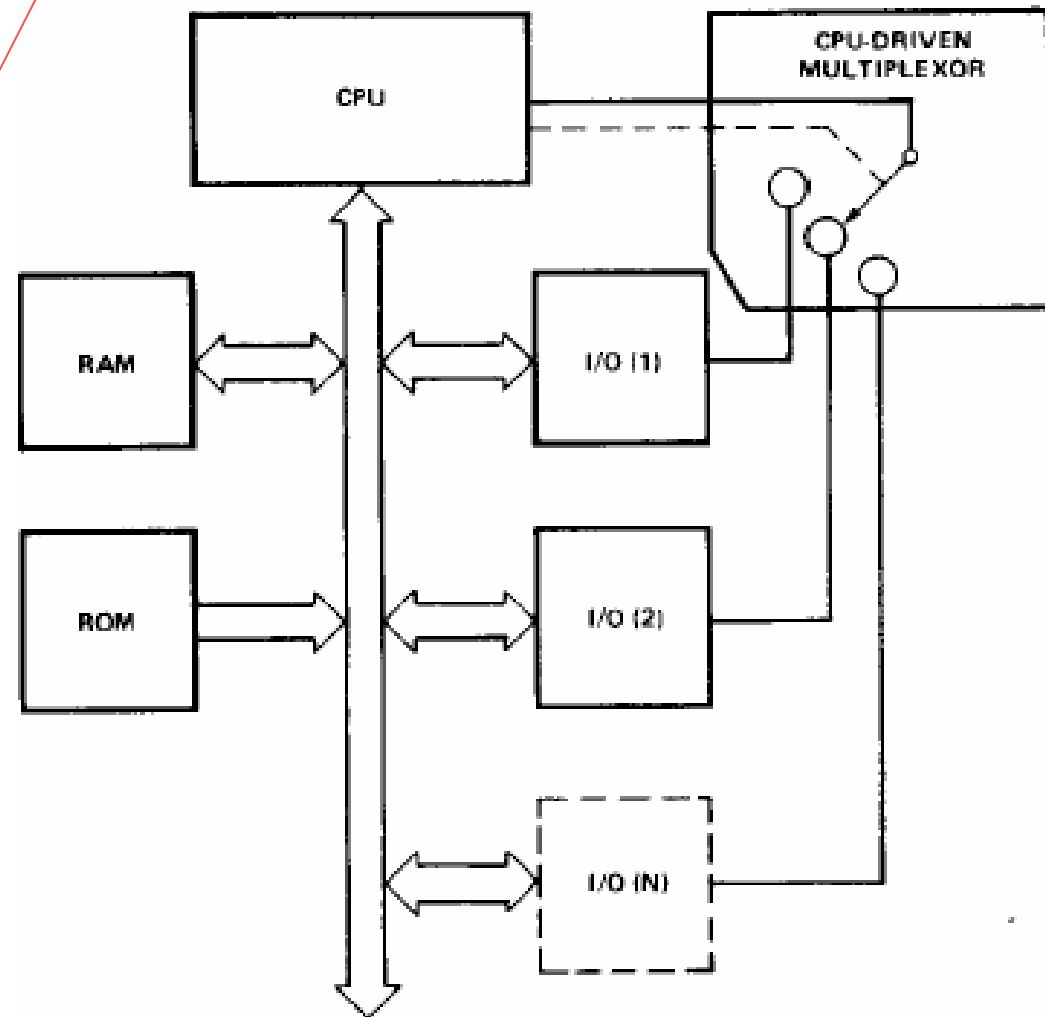
Nech I/O2 má „pripravený“ znak pre CPU.

CPU zaadresuje I/O2 a prečíta znak do ACU.

Atd'.

Ak nie je rýchlosť CPU a I/O2 rovnaká, CPU preberie znak niekoľko krát.

Riešenie je v použití:
Status registra (bitu).

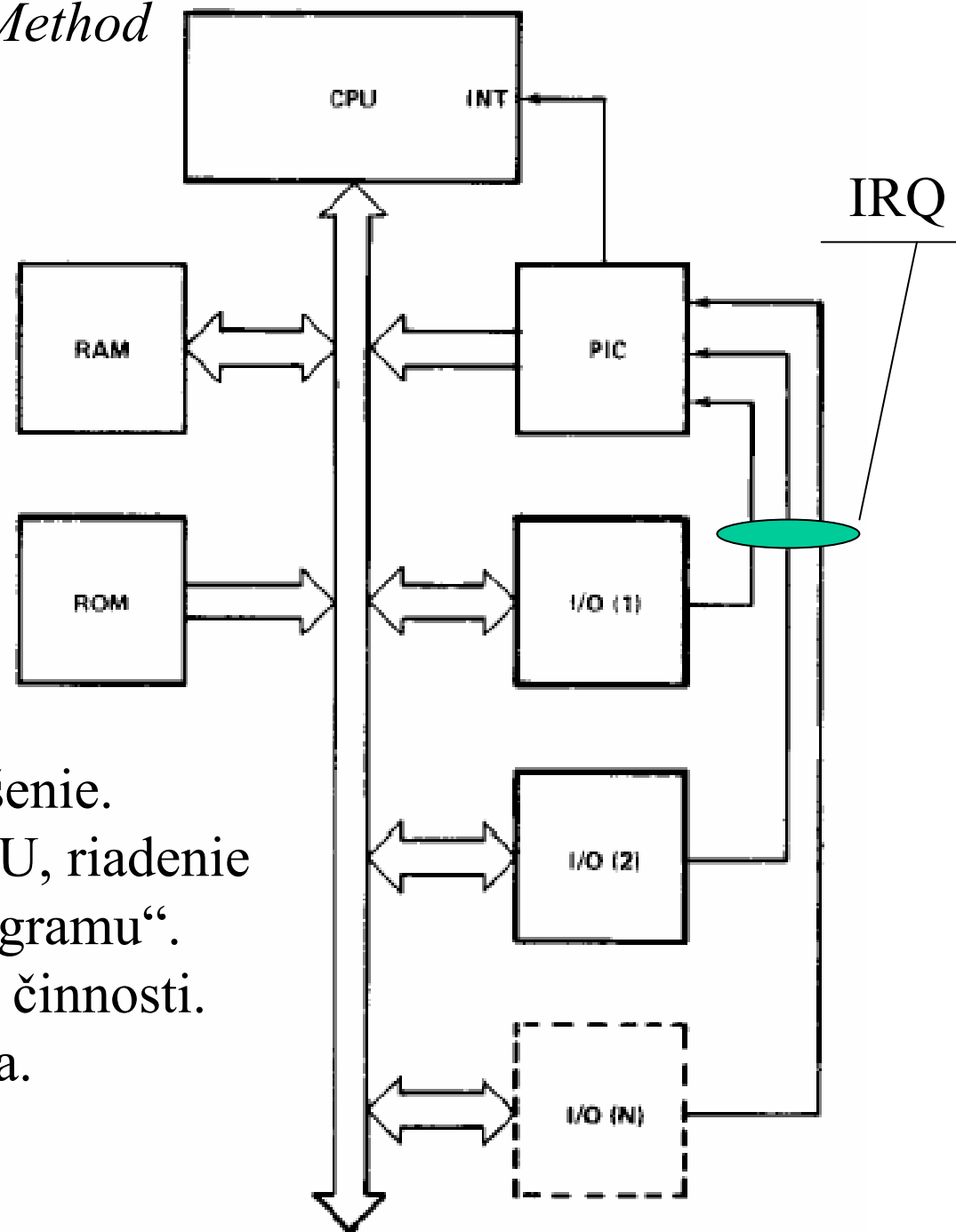


2. Obsluha periférií: Interrupt Method

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 hardwarové prerušenie. T.j. Ak je prerušená činnosť CPU, riadenie sa dočasne odovzdá inému „programu“. Po skončení sa vráti k pôvodnej činnosti. Viacnásobné prerušenia: Priorita.



•Prerušená (Interrupts) a výnimky (Exceptions)

Prerušenie: Udalosť vyvolaná impulzom od externého zariadenia, alebo špeciálnou inštrukciou - programom.

Ak sa dokáže externé zariadenie identifikovať tak, že pošle do CPU svoj kód, ktorý odpovedá počiatočnej adrese obsluhy prerušenia (ISR) daného zariadenia, hovoríme, že k danému zariadeniu je priradený vektor prerušenia.

Procesor musí vedieť obslúžiť súčasné prerušenie od viacerých zdrojov:

- Priorita je daná umiestnením, elektrickým zapojením. Nedá sa meniť.
- Špeciálny obvod vyhodnocuje nastavenú prioritu jednotlivých zariadení.

Podľa zdroja (kto prerušenie aktivuje) rozdeľujeme prerušenia na:
Vonkajšie – asynchronné (hardwarové).
Vnútorne – synchronné (programové)

- **Vonkajšie – asynchronné (hardwarové). Vyvolané hardwarovým zariadením, napr. I/O kartou.**

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. 8259) Obsluhu tohto prerušenia možno programovo zakázať vynulovaním bitu **IF** (Interrupt Flag), resp. povoliť nastavením tohto bitu.

▪ **Vnútorne – synchronné**

- **Softwarové. Cielené** spúšťané inštrukciou **INT**.
- **Výnimky** (Exceptions)
 - **Fault:** táto výnimka sa vyvolá skorej ako sa dokončí príkaz. Do zásobníka sa uloží obsah čítača inštrukcií EIP – 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. Program sa „zastaví“ a debugger kontroluje obsahy registrov.
 - **Abort:** Táto výnimka väčšinou neposkytuje adresu miesta vzniku chyby. Signalizuje sa napr.: chyba hardwaru, alebo neplatnosť systémových tabuliek.

Prerušená a architektúra procesorov INTEL

Procesor 8086 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



CS reg.

IP reg.

V pamäti uložené ako:

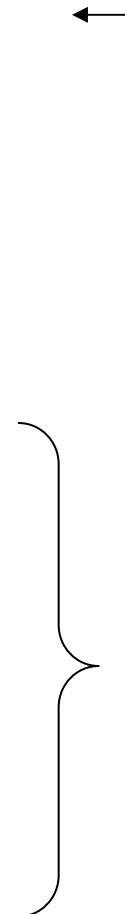
0000: 0B

0001: 1B

0002: 2B

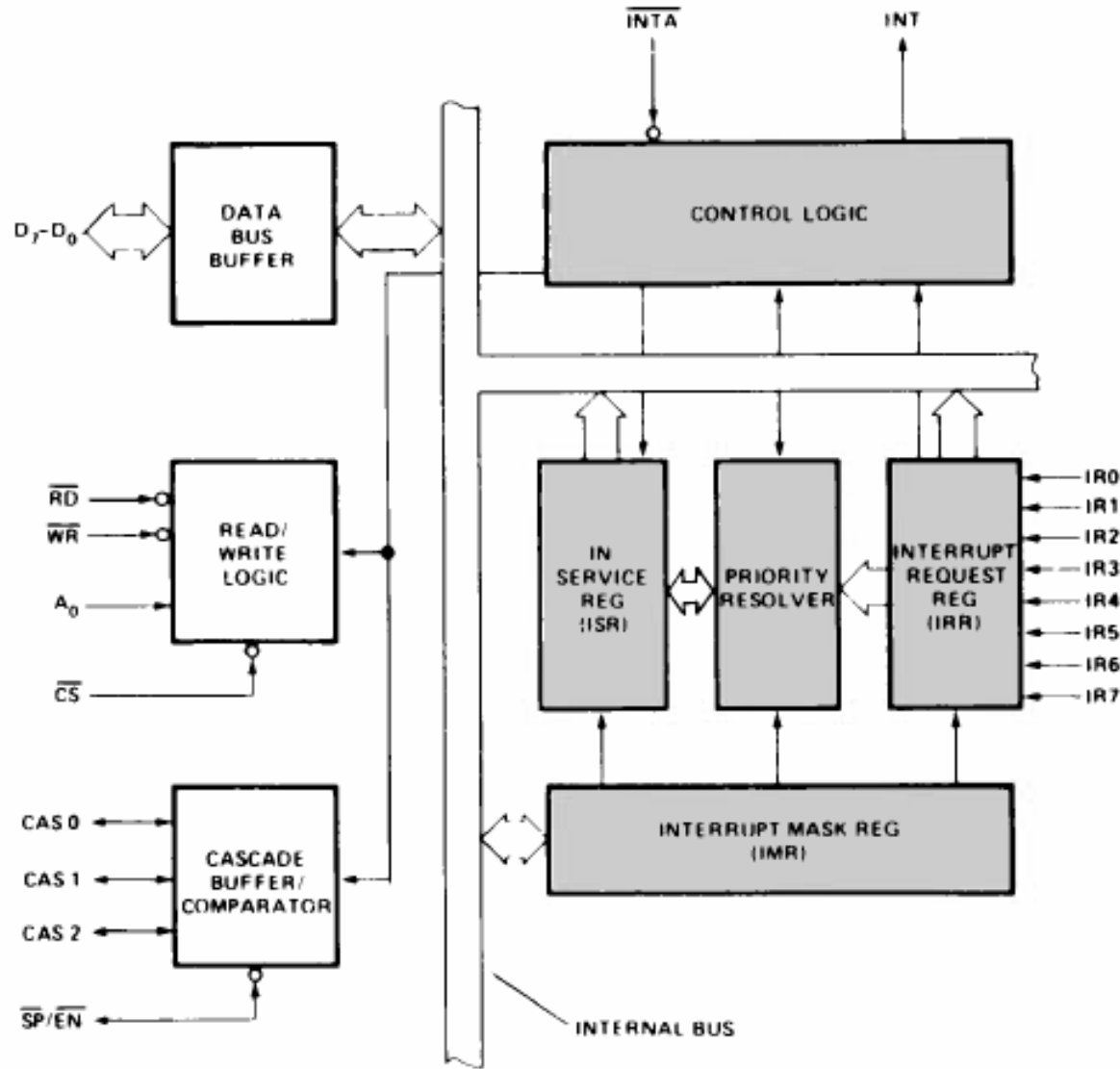
0003: 3B

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
0DH	34H	IRQ5	LPT2	Hardware
0EH	38H	IRQ6	Disketa	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	1DCH	IRQ15		Hardware
78H - FFH			Softwarové prerušenia	



PIC Programmable Interrupt Controller - Kontrolér prerušenia

Dokáže riadiť až osem zdrojov prerušenia **IRQ_i**. Pridel'ovať prioritu a lokálne povoliť, resp. zakázať jednotlivé prerušenie.



Kaskádne zapoj.:
Max.64 IRQ

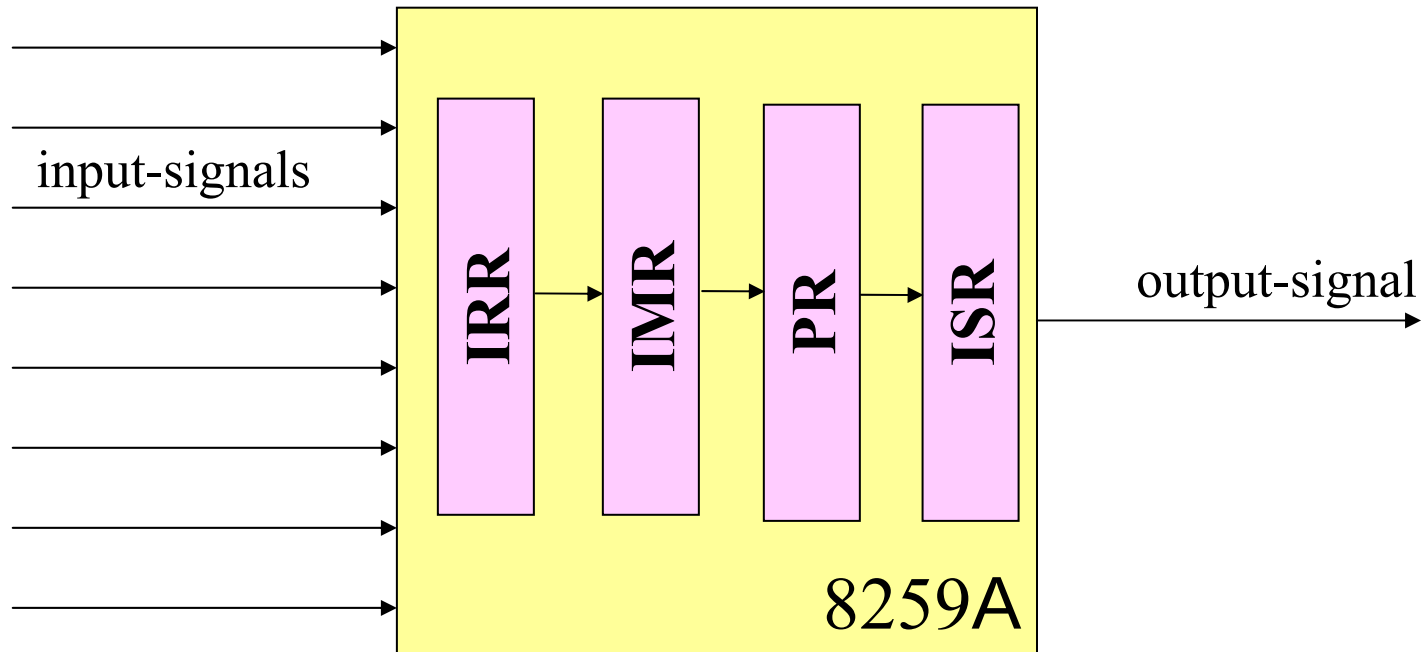
PC XT : 1x PIC

PC AT : 2x PIC

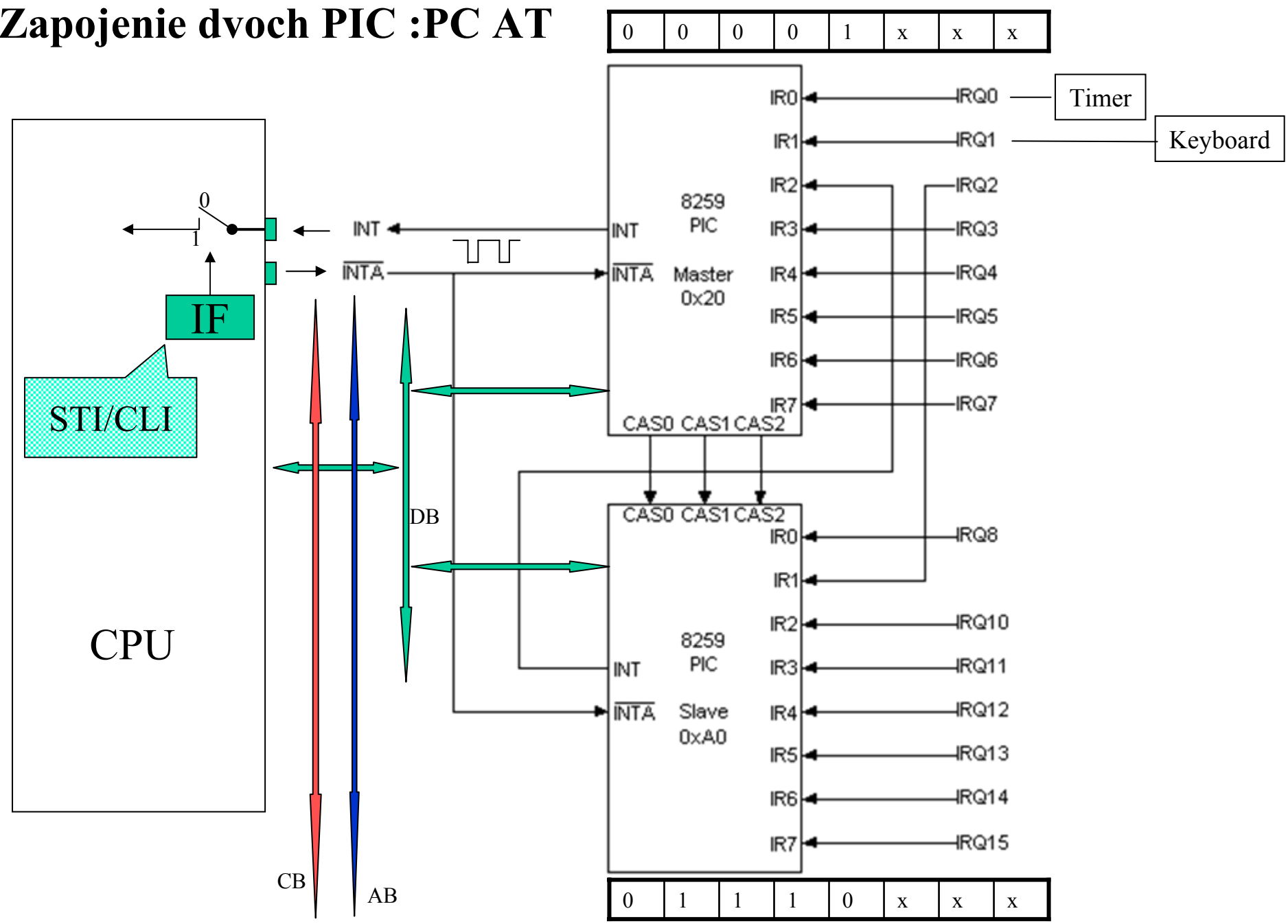
PIC – 8259A/8259A-2:

- **IRR** (Interrupt request register) - *register požiadaviek IRQ*
- **IMR** (Interrupt mask register) - *register masiek požiadaviek IRQ*
Nastavený bit zakazuje odpovedajúce prerušenie a vynulovaný povoľuje.
- **ISR** (In 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

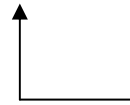


Ovládanie obvodov PIC:

1. Inicializácia, podstatnú časť robí BIOS
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 );`

3. Zrušenie príznaku akceptovaného prerušenia s najv. 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. Prakticky INT8 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)
- $4 * \text{INT_Vector} \Rightarrow \text{IP}$, CS = adresa ISR
- Uloženie dôležitých registrov.

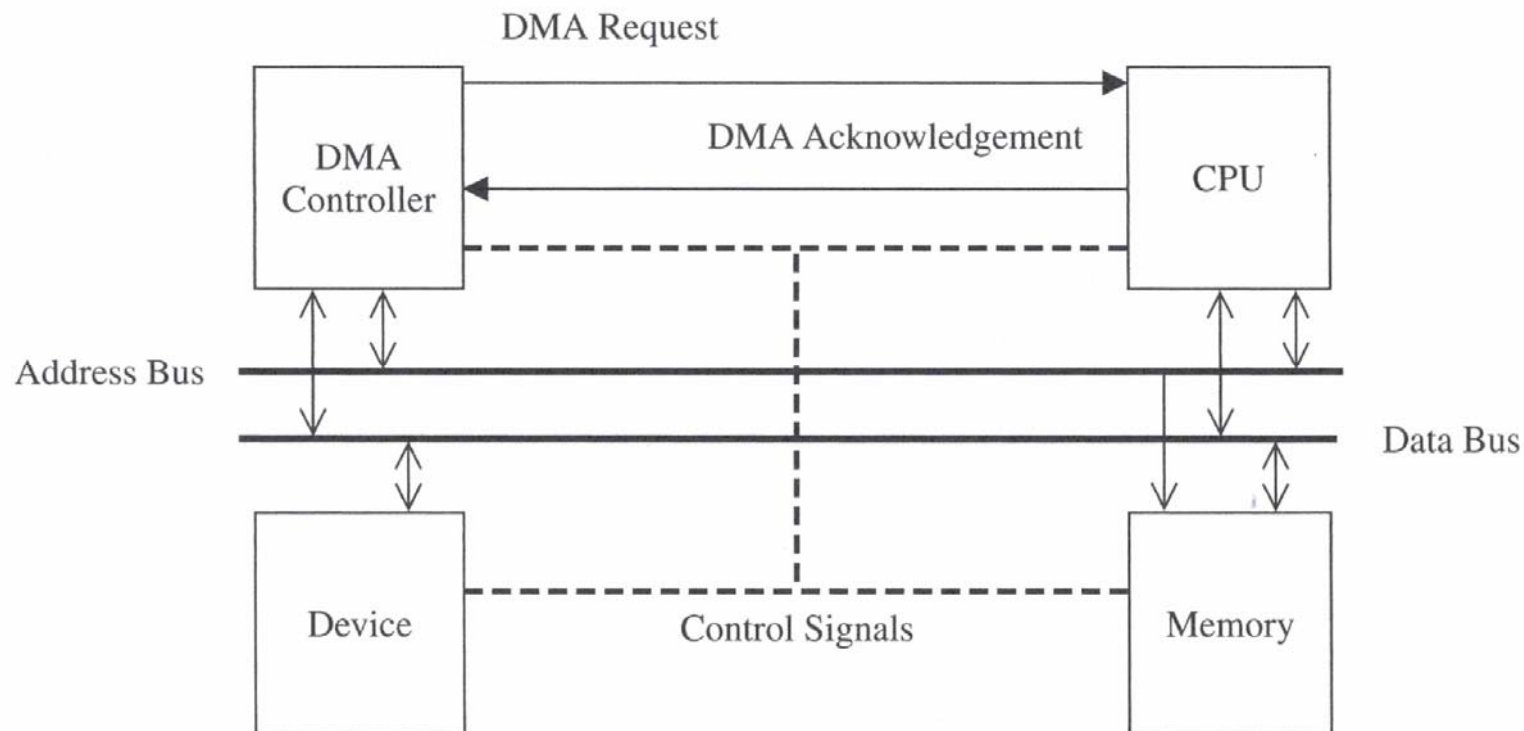
Obsluha prerušenia - Interrupt service routine (ISR) 2

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

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

Dôvod zavedenia DMA: Vynechať CPU ako sprostredkovateľa pri prenose dát medzi operačnou pamäťou a perifériami. Ak umožníme priamy prenos: periféria $\leftrightarrow$ 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.