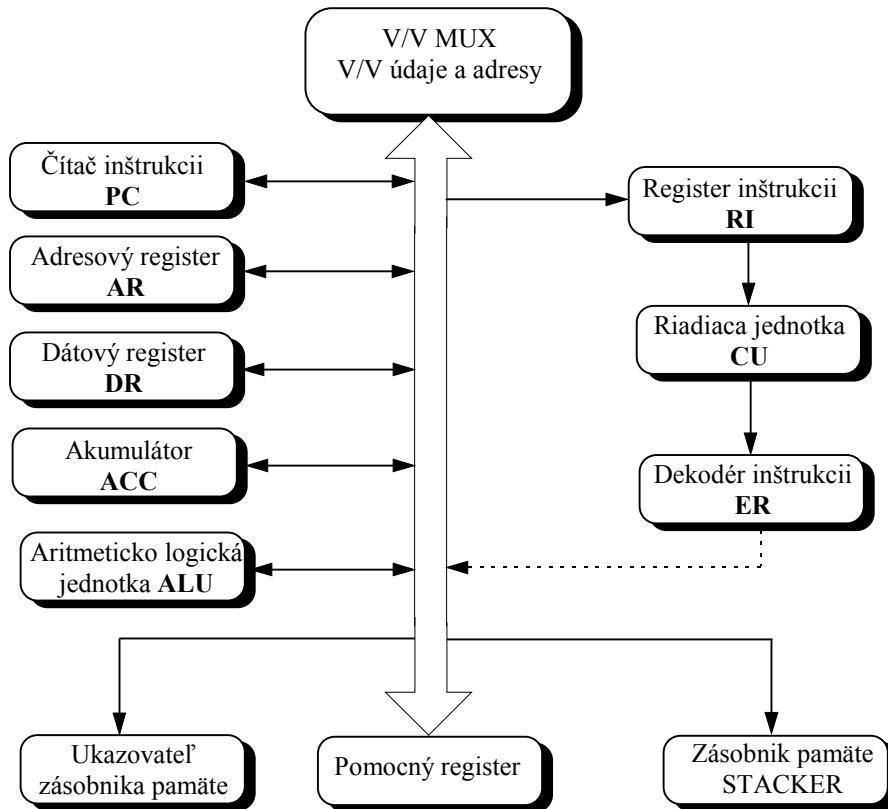


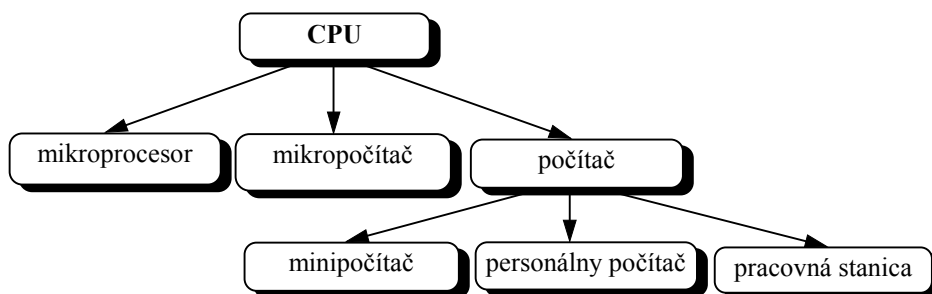
ZÁKLADNÉ POJMY

Mikroočítač: je konštrukčne riešený ako jeden integrovaný obvod, umiestnený v puzdre. Skladá sa z centrálnej jednotky procesorového systému (CPU), z aritmetickej jednotky, z riadiacej jednotky a z rôznych ďalších pracovných jednotiek.



Bloková schéma mikroprocesora

Mikroočítač vznikne z mikroprocesora pridaním ďalších obvodov, napr. pamäti, V/V, prerušení atď. Predstavuje minimálnu konfiguráciu systému, ktorý je schopný samostatne pracovať. Ďalšími obvodmi je možno podstatne zvýšiť jeho výkonnosť, takže vzniknú zariadenia, ktoré sa svojou výkonnosťou rovnajú minipočítačom. Mikroočítač je určený na spracovanie údajov s určitou presnou dĺžkou slova, pre ktorú je najvhodnejší. Každé slovo sa môže interpretovať ako inštrukcia, adresa, alebo časť údajov.



Štruktúra rastu využitia CPU

SPÔSOBY KOMUNIKÁCIE PC S PRÍDAVNÝMI ZARIADENIAMI

Súčinnosť prenosu údajov medzi PC a periférnym zariadením sa realizuje použitím troch základných princípov:

- programovým riadením
- systémom prerušovania programu
- metódou priameho prístupu do pamäte

Programové riadenie styku

Pri tomto spôsobe je styk procesora s prídavným zariadením riadený výlučne len programovými prostriedkami. Program prostredníctvom testovania stavu zariadenia postupne zisťuje, či a ktoré vstupné zariadenia sú pripravené vyslať údaje a ktoré výstupné zariadenia sú pripravené údaje prijať. Pri programovom riadení styku rozlišujeme:

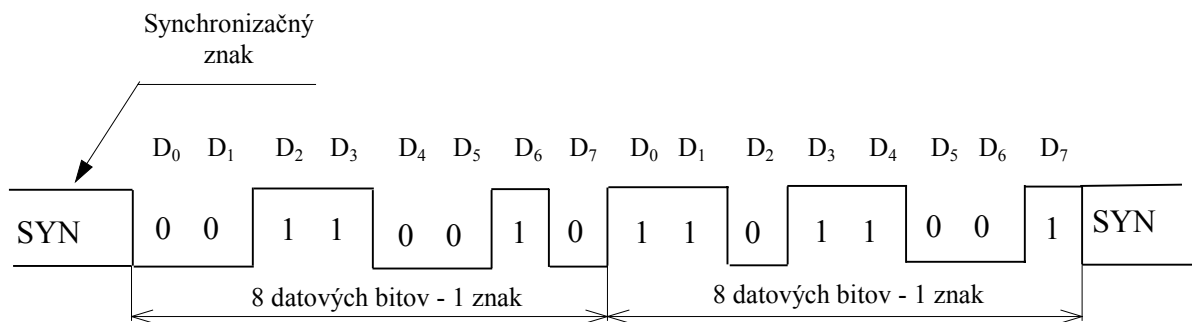
- nepodmienený prenos údajov
- podmienený prenos údajov
- prenos s využitím prerušenia

SÉRIOVÝ PRENOS NA PERSONÁLNO M POČÍTAČI

Synchronný serióvy prenos

Pri synchronnom presnose sú obyčajne prenášané celé bloky znakov. Datové bity jednotlivých znakov pritom nasledujú tesne po sebe, bez akýchkoľvek časových odstupov a nie sú prekládané žiadnymi štart, alebo stop bitmi (môžu byť doplnené jedným paritným bitom). Začiatok prenosu je indikovaný jedným, alebo niekoľkými špeciálnymi synchronizačnými znakmi (tzv. znaky SYN), ktorých hlavným zmyslom je zaistiť potrebnú časovú synchronizáciu odosielateľa a príjemcu, teda pomôcť príjemcovi presne stanoviť časové okamžiky, v ktorých má vyhodnocovať jednotlivé dátové bity. Blok znakov je potom opäť ukončený synchronizačnými znakmi, ktoré môžu (ale nemusia) byť nepretržite vysielané až do začiatku nasledujúceho dátového bloku.

Sériový prenos je obyčajne rýchlejší než asynchronný, pretože nie je zaťažený režiou pripadajúcou na štart a stop bity. Jeho technika a programova realizácia však býva o niečo zložitejšia ako pri prenose asynchronnom.



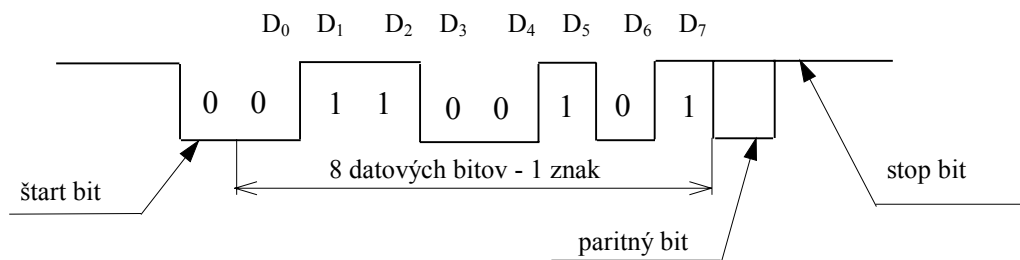
Synchronný prenos znaku

Asynchronný serióvy prenos

Pri asynchronnom prenose môžu byť jednotlivé znaky prenášané s ľubovlnými časovými odstupmi medzi sebou. Príjemca potom však nemôže dopredu vedieť, kedy začína ďalší znak a preto musí byť schopný jeho príchod podľa vhodného príznaku rozpoznať. Týmto príznakom je tzv. štart bit (niekedy sa to nazýva aj rozbehový prvok), ktorým začína každý asynchrónne prenesený znak. Príchod štart bitu je pre príjemcu súčasne aj možnosťou správne si nastaviť svoje merítko času (svoju časovú základňu). To je nutné preto, aby príjemca správne určil časové okamihy, kedy má vyhodnocovať stav jednotlivých dátových bitov, ktoré po štart bite nasledujú.

Za vlastnými dátovými bitmi môže nasledovať jeden tzv. paritný bit a tzv. stop bit (záverný prvok), ktorého dĺžka obyčajne zodpovedá dĺžke jedného, alebo dvoch dátových bitov. Stop bit ne nesie žiadnu informáciu, jeho zmyslom je iba zaistiť určitý minimálny odstup medzi jednotlivými znakmi - vyslanie nasledujúceho znaku môže začať najskôr po odvysielaní celého predchádzajúceho znaku, teda vrátane stop bitu.

Asynchrónnemu prenosu sa niekedy hovorí aj prenos štart - stop.



Asynchrónny prenos znaku

Bios PC pre asynchrónny prenos

- bios PC umožňuje obsluhu 4 asynchrónnych komunikačných portov označených zvyčajne ako COM 1 až COM 4.
- prvý z nich môže byť označovaný aj ako AUX
- bios PC inicializuje štandardne iba porty COM1 a COM2, ktoré majú priradené štandardné adresy I/O adresného priestoru a štandardné úrovne prerušenia

port	bázové adresy	adresy portov	úroveň prerušenia	vektor prerušenia
COM1	0x400	0x3f8 - 0x3ff	IRQ4	0x0c
COM2	0x402	0x2f8 - 0x2ff	IRQ3	0x0b
COM3	0x404	0x3e8 - 0x3ef*	IRQ10*	0x12*
COM4	0x406	0x2e8 - 0x2ef*	IRQ11*	0x13*

* - tieto údaje nie sú štandardne priradené bios-om počítača

Pridelenie adres a prerušení pre jednotlivé asynchrónne porty

- porty COM3 a COM4 musíme inicializovať samostatne, pretože nepatria do štandardnej výbavy počítača. Adresa a úroveň prerušenia sa nastavuje prepínačmi na karte
- bios PC je schopný obsluhovať iba jeden komunikačný port. Port si môže vyžiadať obsluhu bios-om pomocou priradeného prerušenia na základe rôznych stavov závisiacich na hodnote uloženej v registry povolenia prerušenia

Označenie jednotlivých vývodov konektora DB10S a DB25S pri asynchrónnom prenose RS-232

DB10S	Názov signálu	RS 232C	DB25S
1	DCD (Data Carrier Detect), detekovanie požiadavky RI	CF	8
2	RX, RXT (Receive Data Terminal), prijímanie dát	BB	3
3	TX, TXT (Transmit Data Terminal), vysielanie dát	BA	2
4	DTR (Data Terminal Ready), hlásenie o stave, že data sú pripravené k vyslaniu	CD	20
5	GND (Signal Ground), uzemnenie	AB	7
6	DSR (Data Set Ready), potvrdenie prijatia dát	CC	6
7	RTS (Request To Send), požiadavka pre vyslanie dát	CA	4
8	CTS (Clear to Send), potvrdenie požiadavky pre vysielanie	CB	5
9	RI (Ring Indicator), vyzváňanie, príprava k k prenosu	CE	22

PROGRAMOVANIE SÉRIOVÉHO ROZHRAŇIA RS 232C

IO obvod 8251A má k dispozícii osem osem bitových registrov postupne uložených na adresách BASE + 0 až BASE + 7. Adresa BASE je všeobecná adresa prvého registra (v našom prípade môže byť 0x3f8 pre com1). Význam jednotlivých registrov je nasledovný:

<i>adresa</i>	operácia	význam
BASE + 0	zápis	výstupný register dát, ak je bit D7 na adrese BASE + 3 rovný 0
	zápis	nižší byte deliteľa určujúceho (spolu s vyšším byte) prenosovú rýchlosť
	čítanie	vstupný register prijímaných dát

Baud	50	75	110	133,5	150	300	600	1200	1800
Deliteľ	2304	1536	1047	857	768	384	192	96	64

Baud	2000	2400	3600	4800	7200	9600	19200	38400	56000
Deliteľ	58	48	32	24	16	12	6	3	2

Tabuľka deliteľa pre danú prenosovú rýchlosť

BASE + 1	zápis	povoľovací register prerušenia, ak bit D7 na adrese BAS + 3 je rovný 0	
	D7 - D4	nemá význam	
	D3	povolenie prerušenia pri zmene stavu modemu (zmena hodnôt signálov CTS, DSR, RI, RLS D)	
	D2	povolenie prerušenia prostredníctvom prijímacej linky - chyba, alebo prijatý znak BREAK	
	D1	povolenie prerušenia pri vyprázdnení buffra vysielača	
	D0	povolenie prerušenia pri prijatí platných dát	
BASE + 2	zápis	vyšší byte deliteľa určujúceho prenosovú rýchlosť	
	čítanie	identifikačný register prerušení. Pri prerušení je možné na základe tohto registra určiť jeho príčinu.	
	D7 - D3	0 0 0 0 0	
	D2 - D1	0 0 - prerušenie vyvolané prijímačom - preplnenie, chyba parity, chyba dát, BREAK. Nuluje sa čítaním stavu linky - register BASE + 5	
		0 1 - vo vstupnom buffri sú pripravené data. Nuluje sa čítaním vstupného registra - register BASE + 0	
		1 0 - buffer vysielača je prázdny. Nuluje sa zápisom do výstupného buffra - register BASE + 8	
		1 1 - došlo k zmene stavu modemu - signály CTS, DSR, RI, RSLD. Nuluje sa čítaním stavu modemu BASE + 6	
	D0	ak je 0 - není požiadavka na prerušenie. Tento bit je možné využiť pri opakovanom testovaní sériových portov.	
	BASE + 3	zápis	riadiaci register linky

	D7	DLAB - určuje význam registrov BASE + 0 a BASE + 1. ak DLAB = 0 - dátový register BASE + 0 a povoloovací register prerušení BASE + 1 ak DLAB = 1 - registre delitel'a násobku prenosovej rýchlosti
	D6	povolenie kontroly BREAKu. 1 - po BREAKu začni vysielat' medzery
	D5	0 - parita príjmača je rovnaká, ako vysieláča 1 - parita príjmača je opačná ako u vysieláča. (Bios nepoužíva)
	D4 - D3	parita: X 0 - žiadna 0 1 - nepárna 1 1 - párna
	D2	počet stop bitov 0 - jeden 1 - dva (ak je dĺžka slova 5 bitov, potom je to 1,5 bitu)
	D1 - D0	dĺžka slova: 0 0 - 5 bitov 1 0 - 7 bitov 0 1 - 6 bitov 1 1 - 8 bitov
BASE + 4		zápis riadiaci register modemu
	D7 - D5	nevyužitý
	D4	aktivácia slučky pre diagnostiku (DTR ⇒ DSR; RTS ⇒ CTS; OUT1 ⇒ RI; OUT2 ⇒ RLSP)
	D3	hodnota signálu OUT2
	D2	hodnota signálu OUT1
	D1	hodnota signálu RTS
	D0	hodnota signálu DTR
BASE + 5		čítanie register stavu linky
	D7	0 - nevyužitý
	D6	1 - register vysieláča je prázdny
	D5	1 - vysieláč vyslal znak a je pripravený prijať nový
	D4	1 - indikuje prítomnosť znaku BREAK
	D3	1 - chyba konštrukcie znaku - prijatý chybný stop bit
	D2	1 - chyba parity. Nuluje sa čítaním stavu linky
	D1	1 - preplnenie. Došlo k strate prijatého znaku
	D0	1 - data v registri príjmača sú platné
BASE + 6		čítanie register stavu modemu
	D7	1 - signál DCD je aktívny
	D6	1 - signál RI je aktívny
	D5	1 - signál DSR je aktívny
	D4	1 - signál CTS je aktívny
	D3	1 - signál DCD zmenil hodnotu
	D2	1 - signál RI zmenil hodnotu
	D1	1 - signál DSR zmenil hodnotu
	D0	1 - signál CTS zmenil hodnotu
BASE + 7		pri RS 232C nevyužitý

Programovanie sériového rozhrania v jazyku C

- funkciou *bioscom*
- funkciami *inport, inportb, outport, outportb*

Funkcia bioscom

syntax: *int bioscom(int cmd, char abyte, int port)*
prototyp: *bios.h*

Funkcia využíva prerušenie BIOSu 0x14 pre nastavenie a komunikáciu cez zadaný sériový port. Port sa určuje parametrom *port*, ktorý môže nadobúdať nasledujúce hodnoty:

- 0 - pre COM1
- 1 - pre COM2
- 2 - pre COM3
- 3 - pre COM4

hodnota *cmd* určuje čo bude vykonávať funkcia *bioscom* a môže mať hodnoty:

- 0 - nastaví komunikačné parametre daného portu hodnotou danou v *abyte*
- 1 - posíla znak uložený v *abyte* na komunikačnú linku
- 2 - prijíma znak z komunikačnej linky
- 3 - vracia aktuálny stav komunikačného portu

riadiace slovo *abyte* môže nadobúdať nasledujúce hodnoty pre *cmd = 0* (nastavenie parametrov prenosu):

0x02	7 bitové data	0x00	110 baud
0x03	8 bitové data	0x20	150 baud
		0x40	300 baud
0x00	1 stop bit	0x60	600 baud
0x04	2 stop bity	0x80	1200 baud
		0xa0	2400 baud
0x00	bez parity	0xc0	4800 baud
0x08	nepárna parita	0xe0	9600 baud
0x18	párna parita		

Slovo *abyte* je súčtom všetkých parametrov a môže mať tvar:

0xeb ≈ 0xe0|0x08|0x00|0x03

Bioscom vracia 16 bitové číslo pre všetky hodnoty *cmd*. Horných 8 bitov vyjadrujú stav a dolných 8 bitov je závislých na hodnote *cmd*. Horné bity majú význam:

Bit 15	time out
Bit 14	Posuvný register vysielača je prázdny
Bit 13	Vyrovňovací register vysielača je prázdny
Bit 12	Zistené rozpojenie - BREAK
Bit 11	Chyba rámca
Bit 10	Chyba parity
Bit 9	Chyba preplnenia
Bit 8	Data sú pripravené

Ak nebolo možné vyslať hodnotu abyte, tak bit 15 je nastavený na 1 a ostatné bity neplatia. Ináč sú bity platné.

Ak cmd je nastavený na 2, prečítaný byte je v dolných 8 bitoch platný, ak všetky horné bity majú nulovú hodnotu.

Ak cmd má hodnotu 3, potom horné bity majú rovnaký význam, ako v predchádzajúcich prípadoch a dolné bity majú význam:

Bit 7	Detekcia signálu príjmacej linky RLSD
Bit 6	Indikátor vyzváňania RI
Bit 5	Pripravenosť k prenosu - modem je pripojený - DSR
Bit 4	Pripravenosť k vysielaniu - modem je schopný vysielat' - CTS
Bit 3	Zmena signálu príjmacej linky
Bit 2	Detekovaný koniec vyzváňania
Bit 1	Zmena signálu DSR
Bit 0	Zmena signálu CTS

Príklad:

```
#include <bios.h>
#include <conio.h>

#define COM1 0
#define DATA_READY 0x100
#define TRUE 1
#define FALSE 0
#define SETTINGS (0x80|0x02|0x00|0x00)

int main(void)
{
    int in,out,status, DONE=FALSE;

    bioscom(0,SETTINGS,COM1);
    cprintf(„...BIOSCOM {ESC} to exit ...\\n“);
    while(!DONE)
    {
        status=bioscom(3,0,COM1);
        if(status & DATA_READY)
            if((out=bioscom(2,0,COM1) & 0x7F) != 0)
                putchar(out);
                if(kbhit())
                {
                    if((in=getch())== '\\x1b')
                        DONE=TRUE;
                    bioscom(1,in,COM1);
                }
    }
    return 0;
}
```

Funkcie inport, inportb, outport, outportb

tieto funkcie robia priamy pístup na danú adresu.

syntax: *int inport(int port)*

prototyp: *dos.h*

význam: funkcia číta slovo(16 bitov) zo špecifikovaného portu parametrom *port*.

syntax: *int inportb(int port)*

prototyp: *dos.h*

význam: funkcia byte(8 bitov) slovo zo špecifikovaného portu parametrom *port*.

syntax: *void outport(int port, int value)*

prototyp: *dos.h*

význam: funkcia zapisuje slovo dané hodnotou *value* na port špecifikovaný parametrom *port*.

syntax: *void outportb(int port, int value)*

prototyp: *dos.h*

význam: funkcia zapisuje byte dané hodnotou *value* na port špecifikovaný parametrom *port*.

Príklad:

```
#include <stdio.h>
```

```
#include <dos.h>
```

```
int main(void)
```

```
{
```

```
    int result;
```

```
    int port = 0x3f8;
```

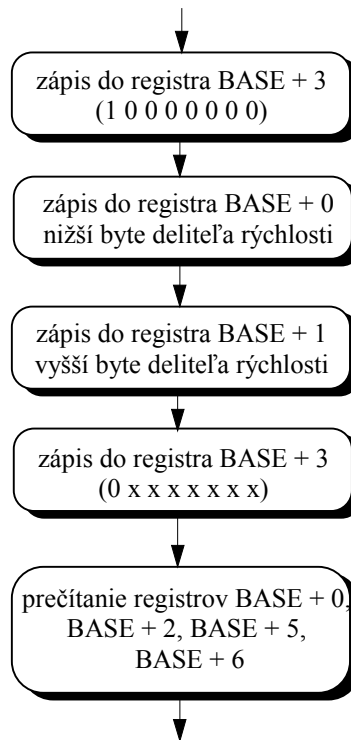
```
    result = inport(port);
```

```
    printf( "Word read from port %d = 0x%X\n", port,result);
```

```
    return 0;
```

```
}
```

Pripravenie obvodu 8251A na prenos



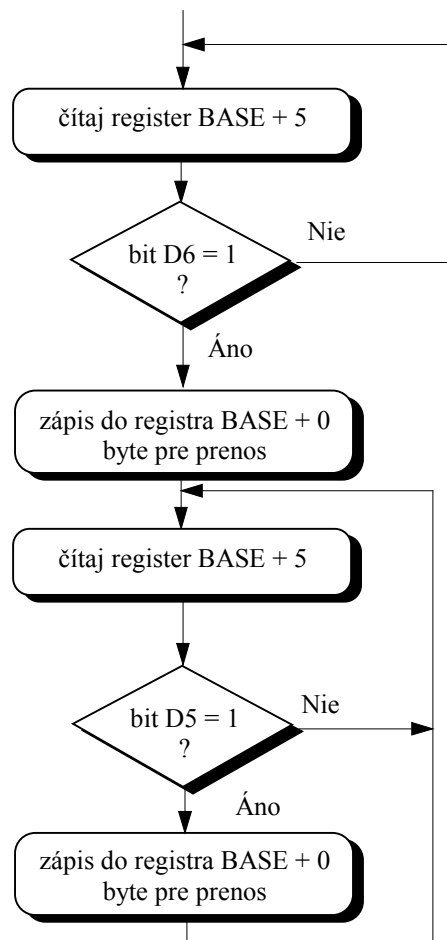
Príklad:

pre funkciu bioscom: bioscom(0, 0x80|0x02|0x00|0x00,0)

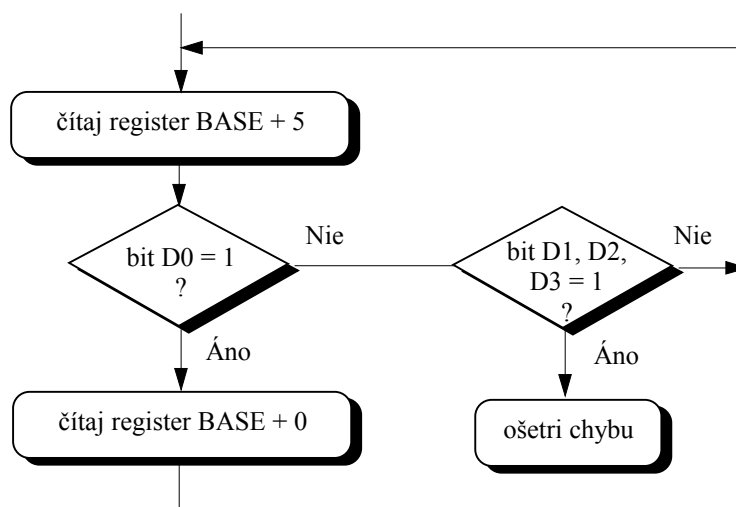
pre funkcie inportb, outportb:

```
outportb(0x3fb, 0x80);  
outportb(0x3f8, 0x0c);  
outportb(0x3f9, 0x00);  
outportb(0x3fb, 0x02);  
inportb(0x3f8);  
inportb(0x3fa);  
inportb(0x3fd);  
inportb(0x3fe);
```

Pripravenie obvodu 8251A pre vysielanie



Pripravenie obvodu 8251A pre príjem



Sériové rozhranie RS 422/485

Tento typ sériového rozhrania sa inštaluje ako externá karta na pozície COM3 a COM4. Priemyselný štandard pre inštalovanie tohto rozhrania do PC AT je karta firmy Advantech - PCL 745. Rozhranie RS 422/485 je priemyselné rozhranie s možnosťou komunikácie až do 1.5 km, rýchlosťou 56 000 baud. Umožňuje pripojenie viacerých rozhraní k jednému miestu čo sa javí ako vysoká prednosť pred rozhraním RS 232C, napr. môže byť prepojených až 12 počítačov spolu a ich obsluha je robená na základe komunikačného protokolu. Tento typ rozhrania umožňuje tri základné spôsoby komunikácie:

- plný duplex
- polovičný duplex
- simplex

Režim plný duplex je charakterizovaný tým, že každý signál má svoju dvojicu vodičov a údaje idú po nich len jedným smerom od vysielateľa zariadenia 1, k prijímaču zariadenia 2. Súčasne však prostredníctvom druhej dvojice vodičov vysielateľ zariadenia 2 môže posielateľ údaje prijímaču 1.

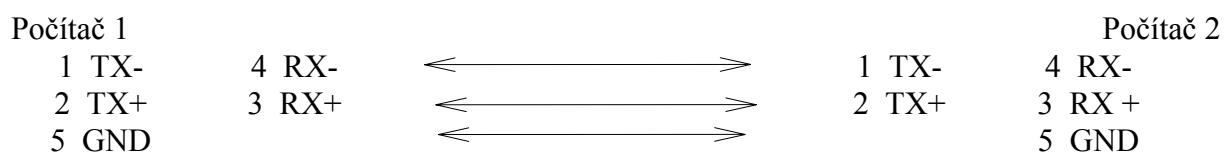
Režim polovičný duplex umožňuje zariadeniam po tej istej linke (dvojici vodičov) údaje prijímať aj vysielateľ. Údaje však nie je možné vysielateľ naraz, lebo len jeden z vysielateľov oboch zariadení môže byť aktívny v danom časovom okamihu, aby nedošlo ku konfliktným stavom na linke. Týmto spôsobom je možné zapojiť aj väčší počet zariadení, pričom sa však musí zabezpečiť, aby vždy visielal v danom čase len jeden vysielateľ a ostatné zariadenia prijímajú, resp. ignorujú správu.

Režim simplex je podmnožinou plného duplexu, kde údaje sa prenášajú len jedným smerom.

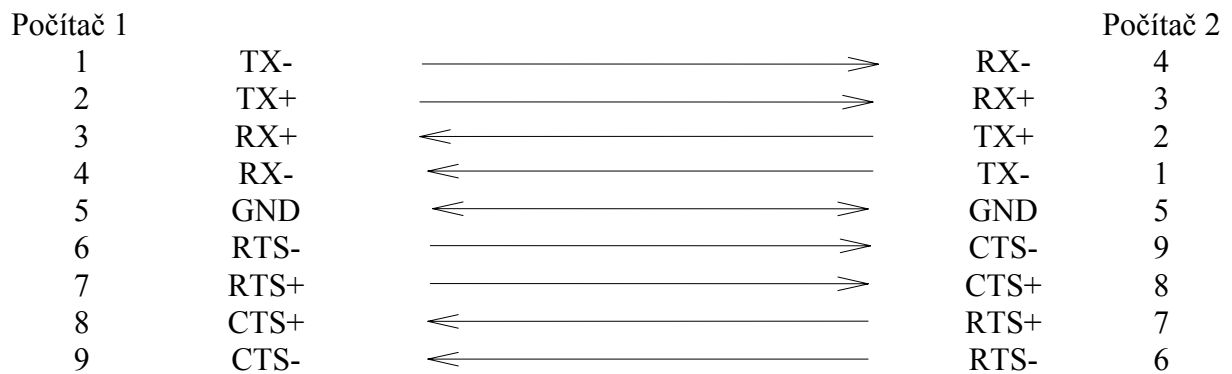
Prenos RS 422 a RS 485 sa nazýva aj diferenciálny prenos, vzhľadom na spôsob prenosu. Karta PCL 745 môže byť inštalovaná na adresách 0x200 až 0x3f8 a s prerušovacou úrovňou 2, 3, 4, 5, 6, 7. Adresa aj úroveň prerušenia sa nastavuje prepínačmi umiestnenými na karte. Karta PCL 745 je ukončená 9 pinovým konektorom, ktorý má nasledujúce rozloženie signálov:

Číslo pinu	Význam
1	TX-, send data -
2	TX+, send data +
3	RX+, receive data +
4	RX-, receive data -
5	GND
6	RTS-, ready to send-
7	RTS+, ready to send +
8	CTS+, clear to send +
9	CTS -, clear to send -

Prepojenie počítačov pri polovičnom duplexe:



Prepojenie počítačov pri plnom duplexe:



Programovanie prenosu RS 422/485

Programovanie tohto prenosu je úplne analogické s programovaním prenosu RS 232C. Inštalovaná karta má rovnaké registre ako obvod 8251A. Pri tomto spôsobe prenosu register BASE +7 (pri RS 232C nevyužitý) má funkciu prepínača medzi prijímačom a vysielačom, čo umožňuje voliť si spôsoby prenosu, ako boli popísané vyššie.

BASE + 7	zápis	bit 0 - 0 vysielač je zakázaný 1 vysielač je povolený
		bit 1 - 0 prijímač je zakázaný 1 prijímač je povolený

Obsadenosť adries v štandardnej zostave PC - AT:

I/O adresa (hex)	použitie adresy
000 - 1FF	použité procesorovou kartou
200	nepoužité
201	game karta
202 - 277	nepoužité
278 - 27F	LPT2t
280 - 2F7	nepoužité
2F8 - 2FF	COM2
300 - 377	nepoužité
378 - 37F	LPT1
380 - 3AF	nepoužité
3B0 - 3BF	monochrome a tlačiareň
3C0 - 3CF	nepoužité
3D0 - 3DF	farba a grafika
3E0 - 3EF	nepoužité
3F0 - 3F7	floppy mechaniky
3F8 - 3FF	COM1

PRERUŠENIE

Na PC existujú dva spôsoby prerušení:

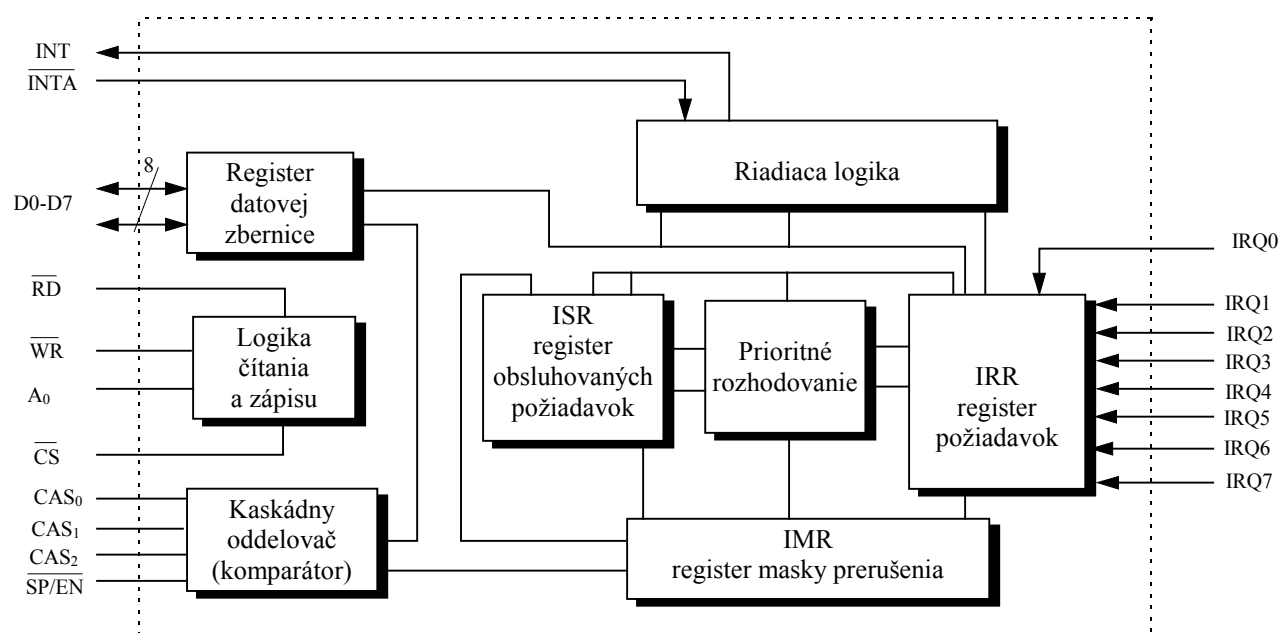
Hardwarové - je priamo riaditeľné vonkajšími signálmi a je nezávislé od behu programu (sú určené vektorom prerušenia, úrovňou prerušenia IR a prioritou)

Softwarové - sú generované programovo. Používajú sa pri inštalácii rezidentných programov (sú určené vektorom prerušenia)

V štandardnej konfigurácii PC je možné nastaviť až 256 vektorov prerušenia. Prvých n vektorov zaberajú hardwarové prerušenia (n je závislé na počte kaskádne riadených radičov prerušenia 8259A, v PC sú štandardne 2, čiže prvých 16 vektorov). Približne ďalších 20 prerušení je vyhradených pre operačný systém MS DOS a prerušenia s vektorom väčším ako 50 sú užívateľské prerušenia. Obslužné programy pre hardwarové prerušenia, ale aj pre prerušenia MS DOS je možné presmerovať a napísať si vlastné rutiny, ktoré budú inštalované na požadovanom vektore prerušenia.

RADIČ PRERUŠENIA INTEL 8259A

Prerušovací systém väčších mikropočítačov je možné vytvoriť použitím obvodu 8259A. Jeden obvod 8259A môže riadiť až osem žiadostí o prerušenie systémom úrovnových priorit. Je možné vytvoriť kaskádny systém jedného nadradeného obvodu a až ôsmich podriadených, čím dostaneme prerušovací podsystem pre 64 žiadostí o prerušenie. Nadradený obvod má signál SP v úrovni H a podradený obvod má tento signál v úrovni L. Vývody kaskádneho riadenia u nadradeného obvodu sú zapojené ako vstupné pretože cez nich nadradený obvod posielajú trojbitový identifikačný znak k prídomom CAS_2 až CAS_0 podriadených obvodov. Bloková schéma obvodu je na obrázku



Bloková schéma obvodu prerušenia 8259A

Register IMR obsahuje masku na selektívne zakázanie jednotlivých zdrojov prerušení IR_{0-7} . Aktívne a povolené žiadosti sú zachytené v registri IRR. Prioritné rozhodovanie vyberá žiadosť z registra IRR, ktorej priorita je vyššia, ako priorita práve obsluhovanej žiadosti a

pomocou obvodu riadiacej logiky žiada signálom INT o prerušenie procesora. Mikroprocesor signálom INTA indikuje, že akceptoval žiadosť a očakáva od obvodu prerušovací vektor na rozpoznanie zdroja prerušenia. Vstupno-výstupné obvody I/O zbernicovej strany umožňujú komunikáciu programu s registrami IMR, IRR a ISR na zisťovanie stavu obvodu. V registri ISR sú žiadosti o prerušenie, ktoré sa práve obsluhujú. Postup činnosti pri výskyte žiadosti o prerušenie je:

- Ak sa na niektorom vstupe IR_i objaví nábehová hrana, nastaví sa príslušný bit v registri IRR, pokiaľ táto žiadosť nie je maskovaná stavom zodpovedajúceho bitu v registri IMR ($IMR_i = 1$ zakazuje prerušenie od IR_i , $IMR_i = 0$ povoľuje prerušenie od IR_i)
- Obvod vyhodnotí požiadavku s najvyššou prioritou spomedzi aktívnych žiadostí v IRR a ak je táto vyššia ako prioritá práve obsluhovanej žiadosti (podľa stavu ISR), tak sa generuje signál INT
- mikroprocesor po akceptovaní signálu INT vysiela impulzy signálu INTA, čím si od obvodu 8259A vyžiada jednu až tri identifikačné slabiky na rozpoznanie zdroja prerušenia.
- obvod 8259A po prijatí impulzu INTA nastaví bit registra ISR podľa čísla požiadavky IR_i , ktorá prerušenie vyvolala a nuluje rovnaký bit registra IRR. Do procesora sa pri tomto prvom impulze neposkytuje žiadny kód
- mikroprocesor vyšle druhý impulz INTA a počas neho obvod poskytuje cez dátový kanál 8 bitový ukazovateľ do tabuľky adres obslužných programov. Mikroprocesor podľa tohto ukazovateľa vyberie z tabuľky adresu skoku, uplatní ju a riadenie sa preniesie do obslužného programu, ktorý obsluží žiadosť o prerušenie.
- nastavený bit v registri ISR sa nuluje osobitným povelením z obslužného programu, alebo automaticky pri druhom impulze INTA, podľa nastaveného režimu práce

V prípade 16 bitového mikroprocesora je tabuľka skokových adres prerušenia nemenne umiestnená od adresy nula a jej položky majú rovnakú dĺžku 4 slabiky. Nie je potrebné zadávať tieto parametre riadiacim slovom, tak ako v prípade 8 bitového mikroprocesora.

Obsah identifikačného vektora vysielaného do procesora je v nasledujúcej tabuľke:

Obsah identifikačného vektora pre mikroprocesor 8086								
IR_i	D7	D6	D5	D4	D3	D2	D1	D0
0	A15	A14	A13	A12	A11	0	0	0
1	A15	A14	A13	A12	A11	0	0	1
2	A15	A14	A13	A12	A11	0	1	0
3	A15	A14	A13	A12	A11	0	1	1
4	A15	A14	A13	A12	A11	1	0	0
5	A15	A14	A13	A12	A11	1	0	1
6	A15	A14	A13	A12	A11	1	1	0
7	A15	A14	A13	A12	A11	1	1	1

Adresové bity A_{11} - A_{15} sa zadávajú do obvodu pri inicializácii riadiacim slovom ICW2.

Programovanie a čítanie stavu obvodu.

Riadenie činnosti obvodu 8259A sa dosahuje zadávaním riadiacich slov z programu. Niektoré slová sa zadávajú pri začiatku činnosti a majú charakter inicializácie obvodu. Označujú sa ICW_1 , ICW_2 , ICW_3 a ICW_4 (Initialization Command Word). Prvé dve sa zadávajú vždy, ďalšie dve len vtedy ak je to potrebné. Ďalšie riadiace slová sa zadávajú počas

činnosti obvodu a modifikujú jeho činnosť v rámci režimu práce nastaveného slovami ICW. Tieto slová sa označujú OCW₁, OCW₂, OCW₃ (Operation Control Word). Sprístupnenie riadiacich a stavových hlásení je v nasledujúcej tabuľke:

Sprístupnenie riadiacich a stavových slov obvodu 8259A					
A0	D4	D3	RD	WR	sprístupnené prvky
0	-	-	0	1	IRR, ISR, žiadajúca úroveň IR _i
1	-	-	0	1	IMR
0	1	x	1	0	ICW ₁
0	0	1	1	0	OCW ₃
0	0	0	1	0	OCW ₂
1	x	x	1	0	OCW ₁ , ICW ₂ , ICW ₃ , ICW ₄

Master radič prerušení v PC-AT obsadzuje adresy 0x20 a 0x21 (je to dané adresným bitom A₀, pre 0x20 A₀ = 0 a pre 0x21 A₀ = 1) I/O adresového priestoru a SLAVE radič prerušenia je na adresách 0xA0 a 0xA1 I/O adresového priestoru.

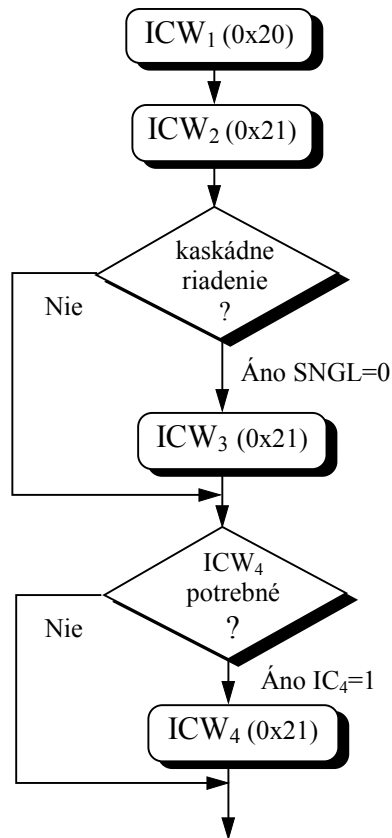
Zápisom ICW₁ sa zaháji inicializačná sekvencia, ktorú je nutné celú dokončiť a v prípade potreby zmien je nutné ju prejsť celú znova. Súčasne so zápisom ICW₁ sa automaticky:

- interná logika sa nastaví k prijatiu takej postupnosti ICW, akú definuje ICW₁, bity SNGL a IC₄. Blokova schéma zápisu je uvedená nižšie
- nulujú sa registre ISR a DMR, register špeciálneho maskovaného módu a klopný obvod automatického módu EOI
- pri nulovom IC₄ v ICW₁ sa anulujú všetky funkcie ICW₄, t.j. nastaví sa mód I 8080/85
- zvolí sa mód FNM s počiatočným nastavením najvyššej priority 0 pre IR₀ a najnižšej 7 pre IR₇
- hranovo citlivý klopný obvod pre každý vstup IR sa nuluje, takže k uplatneniu požiadavky je nutná vzostupná hrana príslušného signálu IR_x. Táto vlastnosť sa ovplyvňuje aj bitom LTIM v IC₁

bit	názov	význam
D7 - D5	A ₇ -A ₅	adresové bity - len v spojení s mikroprocesorom INTEL 8080A a INTEL 8085
D4	1	1
D3	LTIM	1 - citlivosť k úrovni 0 - citlivosť k hrane
D2	ADI	1 - interval 4 slabík pri 8080 0 - interval 8 slabík pri 8080
D1	SNGL	1 - jeden obvod INTEL 8259A 0 - kaskádne riadenie obvodov 8259A
D0	IC ₄	1 - ICW ₄ je nutné 0 - ICW ₄ nie je nutné

Tabuľka významu bitov v riadiacom slove ICW₁

Riadiace slovo ICW₁ sa zapíše na adresu 0x20 a ostatné riadiace slová sa zapíšu na adresu 0x21.

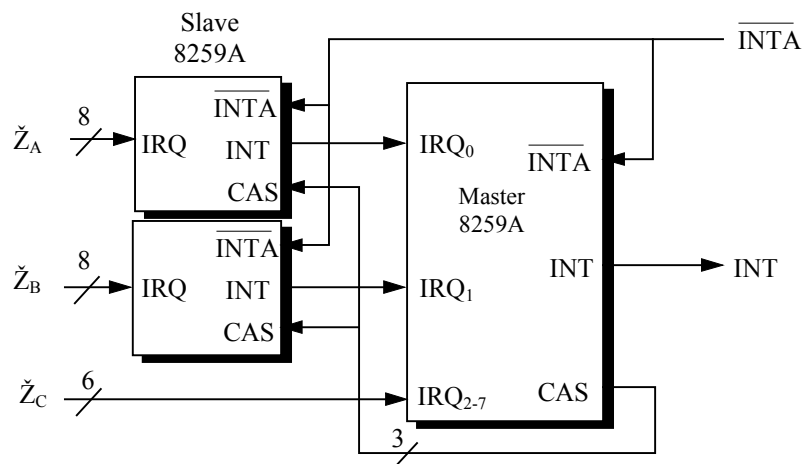


Postup inicializácie radiča prerušenia 8259A

D7-D0		pre mikroprocesory INTEL 8080A a INTEL 8085 A ₁₅ až A ₈ oslužnej rutiny
D7-D3		pre mikroprocesory INTEL 8086 a INTEL 8088 bity A ₁₅ až A ₁₁ typu prerušenia podľa tabuľky uvedenej vyššie

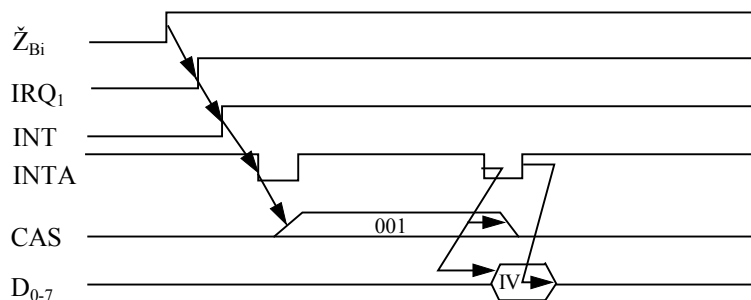
Tabuľka významu bitov v riadiacom slove ICW₂

Slovo ICW₃ sa zadáva, len ak je viac obvodov 8259A zapojených tak, že jeden je nadradený a ostatné sú podradené, čo bolo určené bitom SNGL v ICW₁. Bloková schéma kaskádneho zapojenia viacerých obvodov 8259A je na nasledujúcom obrázku.



Bloková schéma spojenia dvoch podradených obvodov s jedným nadradeným

Dva podradené obvody 8259A prijímajú 8 požiadavok prerušenia \check{Z}_A a 8 požiadavok prerušenia \check{Z}_B . Ich požiadavky o prerušenie sa privádzajú na dva vstupy IR nadradeného obvodu. Ostatné vstupy nadradeného obvodu sa využívajú ako riadne žiadosti mechanizmov alebo stykových obvodov. Nadradený obvod potom generuje celkový signál INT, ktorý je privedený do mikroprocesora. Mikroprocesor si impulzmi INTA vyžiada kód, podľa ktorého skočí na príslušný obslužný program. Pri prvom impulze INTA je aktívny nadradený obvod a poskytne kód inštrukcie CALL, alebo neprenáša žiadnu slabiku. Pri druhých dvoch impulzoch vysiela podradený obvod inštrukcie CALL (pre 8080, resp. identifikačný vektor v druhom impulze INTA (pre 8086). Preto nadradený obvod vyberie trojbitovým vektorom CAS_{0-2} ten podradený obvod, ktorý má mikroprocesoru odpovedať pri sekvencii impulzov INTA. Časovú postupnosť signálov ilustruje obrázok:



Časový sled signálov pri prerušení vyvolanom v kaskádnom radiči

Ak má nadradený obvod 8259A spolupracovať s podradenými obvodmi, treba oznámiť nadradenému obvodu na ktorých vstupoch má pripojené podradené obvody a podradeným na ktorý vstup nadradeného je každý z nich pripojený. To sa dosiahne zaslaním riadiaceho slova ICW_3 do všetkých obvodov. Toto slovo očakáva nadradený aj podradené obvody, avšak štruktúra slova ICW_3 je pre každý z nich odlišná. Pri nadradenom obvode $ICW_{3i}=1$ znamená, že vstup IR_i je pripojený podradený obvod. Stav $ICW_{3i}=0$ znamená, že vstup IR_i má jednoduchú žiadosť z niektorého stykového obvodu.

D7-D0	$S_7 - S_0$	1- vývod IR použitý pre SLAVE obvod 0 - vývod IR použitý priamo
-------	-------------	--

Riadiace slovo ICW_3 pre obvod MASTER

Pri podriadanom obvode v bitoch ICW_{30-2} je zakódovaný index vstupu IR_i na ktorý je podradený obvod pripojený. Podradený obvod porovnáva prijatý kód ICW_{30-2} s kódom na vstupoch CAS_{0-2} a keď sa zistí zhoda, vysiela podradený identifikačný vektor. Bity ICW_{33-7} sú pri podriadanom obvode bez významu.

D7-D3		00000
D2-D0		ID podriadaného obvodu INTEL 8259A

Riadiace slovo ICW_3 pre obvod SLAVE

Pri spolupracujúcich obvodoch 8259A sú vývody CAS_{0-2} nadradeného výstupné a v podriadanom sú vstupné. Na vývode SP/EN sa obvodu signalizuje, či pracuje ako nadradený, alebo podradený. Ak $SP=1$ obvod je nadradený, ak $SP=0$, tak je podradený. Ak obvod 8259A pracuje s oddelovacím budičom na údajovej zbernici, je vývod SP/EN výstupný a riadi sa ním činnosť budiča. Vtedy SP/EN neslúži na rozlíšenie nadradenosti a podradenosti, ale rozlíšenie sa urobí v riadiacom slove ICW_4 .

D7-D5		000
D4	SFNM	typ riadenia vstupov IR: 1 - mód SNF 0 - mód NSNF
D3	BUF	spolupráca z budičom 1 - budič sa riadi výstupom EN 0 - bez budiča
D2	M/S	postavenie obvodu v systéme viacerých obvodov, ak EN je výstup a ICW4 ₃ = 1: 1 - nadradený 0 - podradený
D1	AEOI	nulovanie bitov ISR: 1 - mód AEOI (automaticky) 0 - mód EOI (programom)
D0		typ mikroprocesora: 1 - 8086 0 - 8080

Význam bitov riadiaceho slova ICW₄

Ak sa riadiace slovo ICW₄ nezadá, potom sa obvod správa tak, akoby prijal slovo ICW₄ s nulovými bitmi. Po zadaní riadiacich slov ICW₁ a ICW₄ sa nastaví základný režim práce. V rámci neho je možné obvodu zasielať povely na jeho ovládanie povelovými slovami OCW₁, OCW₂, OCW₃, ktoré sú odlišené adresným signálom A₀.

Povelovými signálmi je možné dosiahnuť prácu v niektorom z nasledujúcich režimov:

- úplné úrovňové priority
- rotácia priorít
- špeciálny maskovací mód
- postupné programové prehliadanie

Riadiacim slovom OCW₁ sa zadáva slabika maskovacích bitov do obvodu. Kódovanie je prirodzené, bitom OCW₁₀ sa maskuje vstup IR₀, bitom ICW₁₁ sa maskuje vstup IR₁ a pod. Slabika maskovacích bitov sa vloží do registra IMR, pričom stav IMR_i = 1 blokuje vstup a IMR_i = 0 umožňuje prerušenie od vstupu IR_i.

Obsah slova OCW1		
D7-D0	M7-M0	maska prerušenia: 1 - zakázané prerušenie 0 - povolené prerušenie

Pri použití úrovňového prioritného systému treba obvod nastaviť do režimu úplných úrovňových priorít. Po zadaní slov ICW sa obvod nachádza práve v tomto režime (Fully Nested Mode).

Jednotlivé pracovné módy sa nastavujú pracovným slovom OCW₂.

Obsah slova OCW2		
D7	R	povel na cyklickú zmenu úrovne
D6	SL	povel SEOI ukončenie obsluhy špecifikovanej žiadosti
D5	EOI	povel EOI, ukončenie obsluhy prerušenia
D4		0
D3		0
D2	L2	najvyšší bit čísla vstupu IR
D1	L1	stredný bit čísla vstupu IR
D0	L0	najnižší bit čísla vstupu IR
Obsah 5, 6 a 7 bitu slova OCW2		

7 bit	6 bit	5 bit	význam
0	0	0	ruší sa cyklické pridelovanie priorít pri automatickom nulovaní bitov ISR - automatická rotácia v mode AEOI-RESET
0	0	1	ukončenie obsluhy pri úrovňových prioritách, nuluje sa bit ISR s aktívnou hodnotu a najvyššou prioritou - nešpecifikovaný príkaz EOI
0	1	0	žiadna činnosť - prázdna operácia NOP
0	1	1	nuluje sa bit registra ISR, ktorého kód je v 0 až 2 bite OCW ₂ - špecifikovaný príkaz EOI
1	0	0	obvod sa nastaví na cyklické pridelovanie priorít pri automatickom nulovaní bitov ISR - automatická rotácia v móde AEOI-SET
1	0	1	ukončenie obsluhy pri cyklickom pridelovaní priorít, nuluje sa aktívny bit ISR s najvyššou prioritou a obsluženému vstupu sa pridelí najnižšia priorita - automatická rotácia na nešpecifikovaný príkaz EOI
1	1	0	nastavenie najnižšej priority vstupu, ktorého kód je v 0 až 2 bite OCW ₂ - príkaz nastavenia priority špecifickej rotácie
1	1	1	ukončenie obsluhy pri cyklickom pridelovaní priority, nuluje sa aktívny bit ISR s najvyššou prioritou a najnižšia priorita sa pridelí vstupu IR, ktorého kód je v 0 až 2 bite OCW ₂ - špecifická rotácia na špecifický príkaz EOI

Popis jednotlivých pracovných módov:

Mód s úplným vnorením FNM(Full Nested Mode)

Je základným módom v ktorom je radič prerušenia INTEL 8259A ihneď po inicializačnej sekvencii, pokiaľ nie je vyžiadaný iný mód činnosti. V ňom sa najčastejšie prevádzkujú jednotlivé radiče 8259A. V ňom má prerušovací vstup IR₇ najnižšiu prioritu a IR₀ má najvyššiu prioritu. Ak sa na vstupoch IR objaví jedna, alebo viac požiadaviek o prerušenie, prioritné rozhodovanie rozpozná z nich najvyššiu prioritu a ak táto je vyššia ako najvyššia priorita práve obsluhovanej žiadosti, tak obvod žiada na výstupe INT o prerušenie. Súčasne obvod nastaví príslušný bit registra ISR, ktorý zostane nastavený, pokiaľ mikroprocesor nevyšle ukončovací príkaz EOI. Počas tohto sú zablokované všetky požiadavky prerušenia rovnakej a nižšej priority. Systém úplných úrovňových priorít je vhodný pre nerovnocenné zariadenia, ktoré je možné jednoznačne zoradiť podľa naliehavosti ich obsluhy.

Príkaz nešpecifického módu EOI

Oznamuje radiču prerušenia, že obsluha požiadavky skončila, ale nešpecifikuje o ktorú úroveň išlo. Tento spôsob by mal byť využívaný iba v prípadoch, keď posledné obsluhované prerušenie má najvyššiu prioritu z akceptovaných požiadavok. Potom obvod 8259A určí bit najvyššej priority v ISR a nuluje ho. Nešpecifikované ukončenie by nemalo byť použité pre prípady, keď sa menia priority v priebehu podprogramov a pri použití módu špeciálnej masky.

Príkaz špecifického EOI:

oznamuje obvodu 8259A, že končí obslužný program s úrovňou špecifikovanou bitmi L2 až L0 v OCW₂.

Mód automatického ukončenia prerušenia - AEOI

V tomto móde nemusí procesor vysielat' príkaz OCW₂ oznamujúci koniec podprogramu obsluhy prerušenia, pretože obvod 8259A sám vykoná nešpecifický EOI okamžite po vzostupnej hrane posledného impulzu INTA, to znamená, že nie na konci, ale na začiatku podprogramu prerušenia. V tomto prípade vzniká riziko porušovania priorit a pravidiel vnorovania sa podprogramov v móde FNM a aj riziko neobmedzeného preplňovania zásobníkovej pamäte.

Automatická rotácia priority

Ak sú prídavne zariadenia prevažne rovnocenné, je záujem aby sa priorita ich obsluhy rozdelila rovnomerne. Obvod 8259A umožňuje cyklické pridelovanie najvyššej priority v závislosti od vybavenia požiadavky na prerušenie. Je to dané spôsobom, že ak sa obsluhuje prerušenie IR₆, potom to má najvyššiu prioritu. Po jeho obsluhu je udelená najvyššia priorita pre IR₇ a najnižšia pre IR₆ atď. Znamená to, že po obslúžení požiadavky IR_i táto dostane najnižšiu prioritu a IR_{i+1} dostane najvyššiu prioritu.

Príkaz špecifickej priority

Priraduje špecifikovanému vstupu IR_x najnižšiu prioritu a ostatné priority sa samočinne upravia v súlade s módom FNM.

Špecifická rotácia

funguje ako kombinácia nastavenia špecifickej priority a špecifického EOI. Špecifickému vstupu IR_x sa priradí najnižšia priorita, súčasne sa špecifikovaná úroveň nuluje v registri ISR.

Mód špeciálnej masky

Nastavuje sa v slove OCW₃ bitom SMM. V tomto prípade je povolené prerušenie na všetkých úrovniach okrem tej, ktorá má v maske jedničku. Týmto je možné v priebehu podprogramu pre obsluhu prerušenia umožniť uplatnenie požiadavky nielen vyššej, ale aj nižšej priority, ktorá odpovedá aktuálnemu podprogramu. V tomto prípade radič už nie je v móde FNM, ale môžeme selektívne povoľovať prerušenia ľubovoľnej úrovne.

Obsah slova OCW3		
D7	ESMM	nemá význam
D6		určuje platnosť 5 bitu: 0 - nezáleží na stave bitu 5 1 - bit 5 sa uplatní
D5	SMM	typ maskovania: 0 - riadne maskovanie 1 - režim špeciálnej masky
D4-D3	P	01
D2		vyhodnotenie stavu vstupov IR: 0 - bez funkcie 1 - nasledujúcim čítaním sa získa kód najvyššej aktívnej žiadosti v režime programového prehliadania stavu zariadení
D1	RR	platnosť bitu 0: 0 - na stave bitu 0 nezáleží 1 - bit 0 sa uplatní
D0	RIS	0 - číta sa register IRR 1 - číta sa register ISR

Bitom OCW3₂ sa dá dosiahnuť rozpoznanie aktívnej požiadavky s najvyššou prioritou. Operáciou čítania na adrese 0x20 sa získa stavové slovo, ktoré obsahuje požadované informácie a má nasledujúci tvar:

Význam bitov stavového slova		
D7		0 - žiadny vstup nie je aktívny 1 - aspoň jeden vstup je aktívny
D6-D3		nemajú význam
D2-D0		kód aktívnej žiadosti D0 je najnižší bit a D2 je najvyšší bit

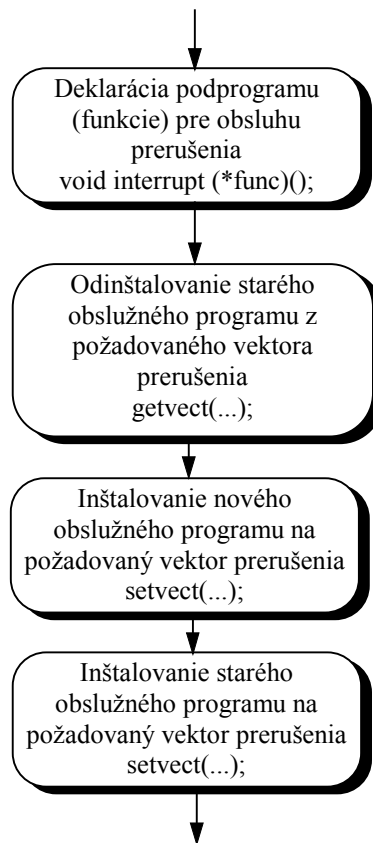
Po zadaní povelu OCW3₂=1 sa vyhodnotí stav vstupov IR a zmrazí sa stav obvodu až do najbližšieho čítania čísla aktívneho vstupu. Toto čítanie sa interpretuje ako akceptovanie prečítaného čísla žiadosti a preto sa nastaví príslušný bit registra ISR, ako pri akceptovaní žiadostí o prerušenie.

Nulovanie, resp. nastavenie počiatočných podmienok - RESET sa prevádza u radiča 8259A pomocou vloženia novej postupnosti riadiacich slabík ICW. Týmto sa obvod nastaví na hodnoty:

- nuluje sa register IMR
- vstup IR získa najnižšiu prioritu
- podradený obvod získa adresu 7, t.j. akoby bol pripojený na vstup IR₇ nadradeného obvodu
- ruší sa špeciálny maskovací mód
- čítanie je nastavené na register IRR

Úroveň	Zariadenie
NMI	chyba parity dát
0	časovač číslo 0
1	výstupný buffer klávesnice
2	pripojenie SLAVE radiča 8259A
8	hodiny reálneho času
9	voľný, softwarovo presmerovaný na IR ₂
10	voľný
11	voľný
12	voľný
13	matematický koprocessor INTEL 80x87
14	radič pevného disku
15	voľný
3	asynchrónny sériový port 2
4	asynchrónny sériový port 1
5	paralelne rozhranie CENTRONICS 2
6	radič diskiet
7	paralelné rozhranie CENTRONICS 1

Programovanie prerušenia v jazyku C



Vývojový diagram pre inštalovanie vlastnej obslužnej rutiny prerušenia

getvect

syntax: `void interrupt (*getvect(int interruptno))();`
 prototyp: `dos.h`

Funkcia číta hodnotu vektora prerušenia daného parametrom *interruptno* (napr. 0x0c) a vracia túto hodnotu ako ukazovateľ na funkciu prerušenia. Hodnota *interruptno* môže byť od 0 do 255.

setvect

syntax: `void setvect(int interruptno, void interrupt(*isr)());`
 prototyp: `dos.h`

funkcia nastavuje hodnotu vektora prerušenia daného parametrom *interruptno* na novú hodnotu *isr*, ktorá je vzdialeným ukazovateľom, ktorý obsahuje adresu novej funkcie prerušenia. Adresu C rutiny je možné vkladať do *isr* iba ak táto rutina je deklarovaná ako rutina interrupt.

interrupt

je kľúčové slovo, ktoré definuje funkcie majúce vzťah k prerušeniu.

Príklad:


```

#include <stdio.h>
#include <dos.h>
#include <conio.h>

#define INTR 0x1c          /* prerusenie od tiky hodin*/

void interrupt (*oldhandler)(void);
int count = 0;

void interrupt handler(void)
{
    count++;
    oldhandler();
}

int main(void)
{
    oldhandler=getvect(INTR);          /*ulozi stary vektor prerusena*/
    setvect(INTR, handler);           /*instaluje novy obsluzny program*/
    while(count<20)
        printf("pocitadlo je &d\n", count);
    setvect(INTR, oldhandler);        /*nastavi stary obsluzny program*/
    return 0;
}

```

enable

syntax: *void enable(void);*

prototyp: *dos.h*

Makro enable povoľuje prerušenia, čím umožňuje výskyt ľubovoľných prerušení.

disable

syntax: *void disable(void);*

prototyp: *dos.h*

Makro disable zakazuje prerušenia. V tomto prípade je povolené iba NMI (nemaskovateľné prerušenie) od ľubovlného externého zariadenia.

int86

syntax: *int int86(int intno, union REGS *inregs, union REGS *outregs);*

prototyp: *dos.h*

Funkcia prevádza softwarové prerušenie 80x86 špecifikované argumentom intno. Pred prevedením softwarového prerušenia kopíruje registrové hodnoty z inregs do registrov. Po návrate zo softwarového prerušenia kopíruje intáž aktuálne registrové hodnoty do outregs, kopíruje stav príznaku prenosu do poľa x.cflag v outregs a kopíruje hodnotu príznakového registra 80x86 do poľa x.flags v outregs. Ak je príznak prenosu nastavený, obyčajne označuje, že sa vyskytla chyba.

int86x

syntax: *int int86x(intno, union REGS *inregs, union REGS *outregs, struct SREGS *segregs);*

prototyp: *dos.h*

intdos

syntax: *int intdos(union REGS *inregs, union REGS *outregs);*

prototyp: *dos.h*

Funkcia prevádza prerušenie 0x21 DOSu, aby sa vyvolala špecifikovaná funkcia DOSu určená parametrom *inregs->h.ah*.

intdosx

syntax: *int intdosx(union REGS *inregs, union REGS *outregs, struct SREGS *segregs);*

prototyp: *dos.h*

intr

syntax: *void intr(int intno, struct REGPACK *preg);*

prototyp: *dos.h*

Funkcia je alternatívnym rozhraním pre vykonávanie softwarovýc prerušení. Generuje prerušenia špecifikované v *intno*.

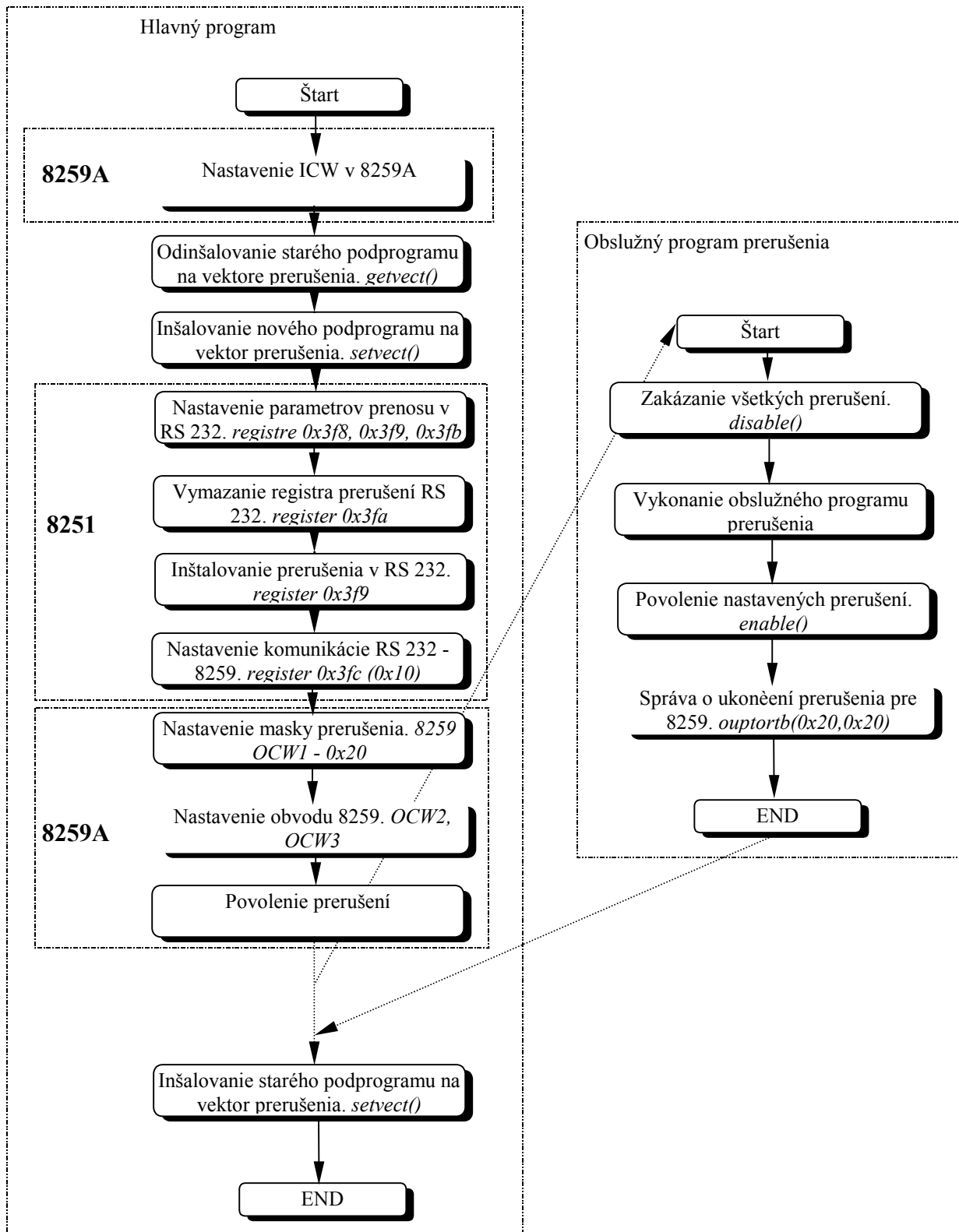
geninterrupt

syntax: *void geninterrupt(int intr_num);*

prototyp: *dos.h*

Makro spúšťa softwarové prerušenia dané parametrom *intr_num*.

Inštalovanie prerušenia od RS 232



Vývojový diagram pre obsluhu prerušenia RS 232

```

#include <stdio.h>
#include <dos.h>
#include <conio.h>
  
```

