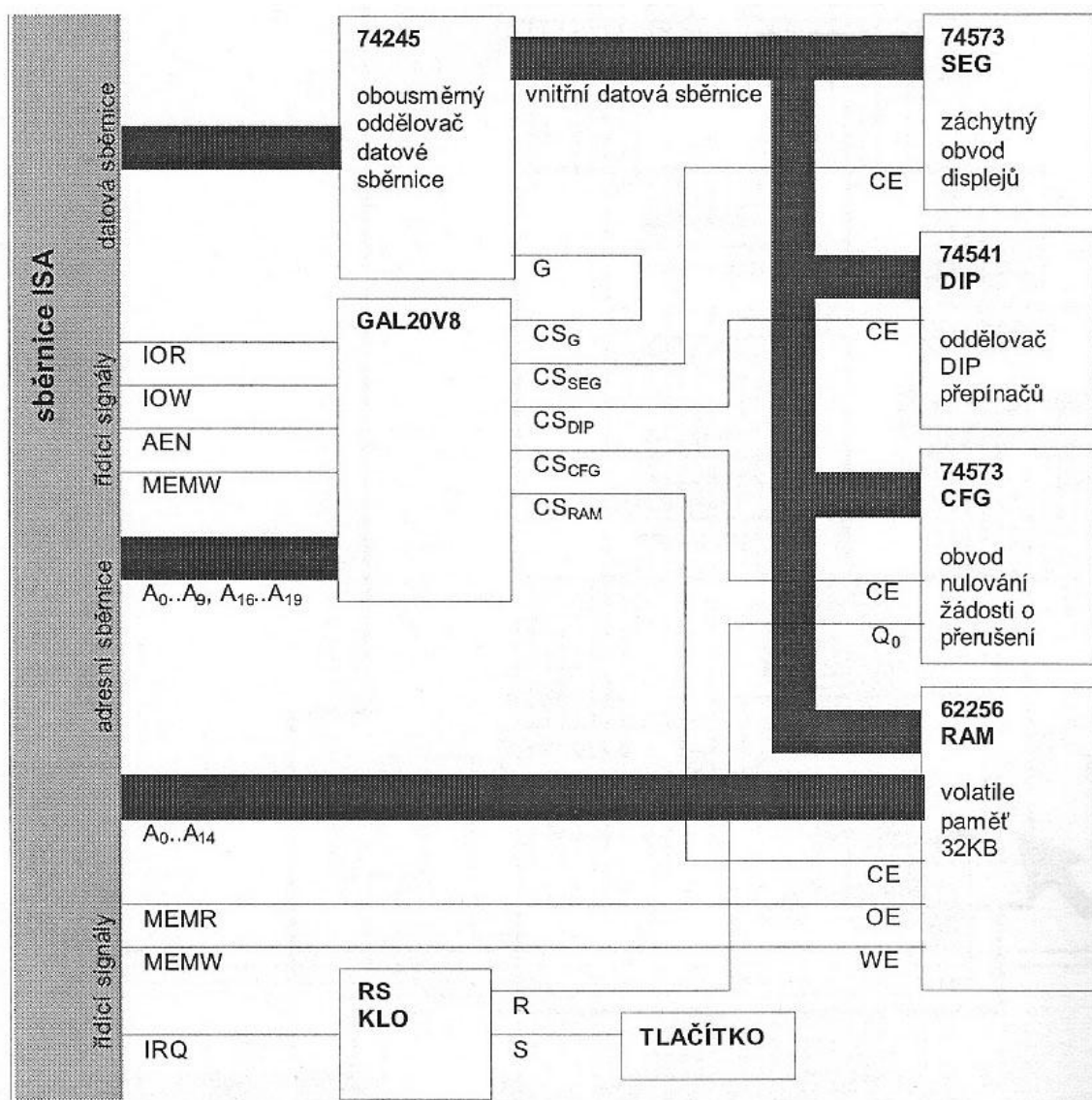


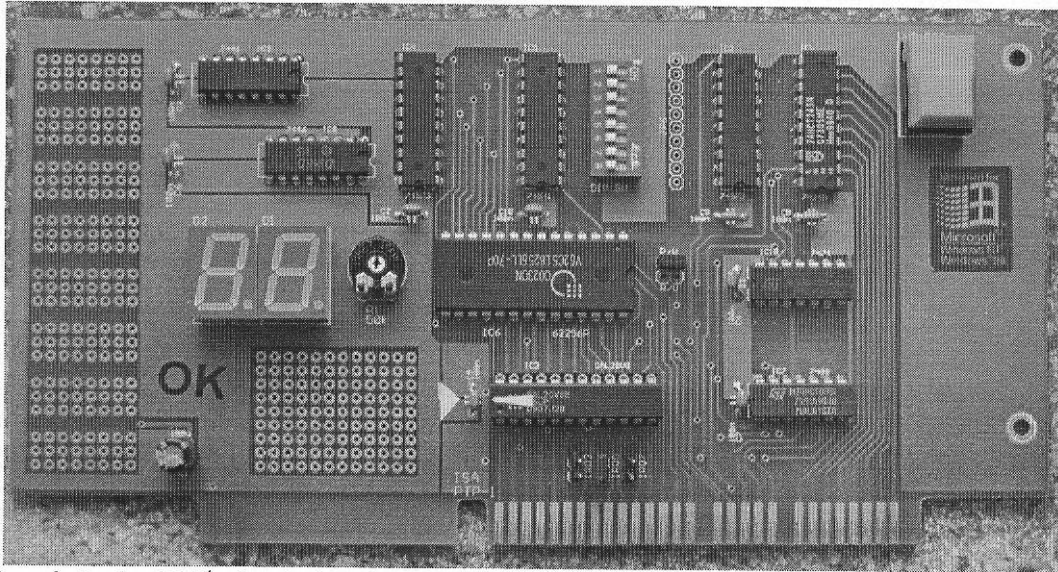
Zadání:

1. Seznamte se s konstrukcí cvičné zásuvné adaptérové desky do PC, zejména pak s konstrukcí generátoru přerušení.
2. Vytvořte program v jazyku C včetně obslužné rutiny hardwarového přerušení, který bude vhodným způsobem demonstrovat událost příchodu přerušení z adaptérové desky.

Blokové schéma adaptérové desky



Adaptérová deska



Adaptérová deska ISA¹ obsahuje následující:

- Programovatelný adresový dekodér (GAL20V8²)
- Paměť RAM (32KB, 32768x8b, IO 62256)
- Vstupní port DIP spínačů
- Výstupní port LED segmentového displeje
- Výstupní port potvrzení hardwarového přerušení
- Tlačítko pro generování hardwarového přerušení

Výstupní port potvrzování hardwarového přerušení je na adrese 300h, výstupní port LED segmentového displeje má adresu 301h, vstupní port DIP spínačů má adresu 302h. Paměť RAM je v segmentu D000h.

Vstupně/výstupní porty

Na adaptérové desce jsou 2 výstupní a 1 vstupní port. Všechny mají šířku 8 bitů.

Na výstupním portu na adrese 301h jsou připojeny dva sedmissegmentové displeje LED s šestnáctkovými dekodéry. Tento port slouží jak k jednoduchému ověření funkce adresového dekodéru, tak i k demonstraci sběrnicevého cyklu „zápis na výstupní port“. Data vložená na tento port jsou ihned zobrazena na displeji. Aby byla zapsaná hodnota zobrazena i po skončení sběrnicevého cyklu, je tento port realizován osmibitovým registrem typu latch 74573.

Druhý výstupní port na adrese 300h slouží jako port potvrzení (nulování) žádosti o přerušení. Krátkým nastavením nejnižšího bitu portu na hodnotu 0 se asynchronně vynuluje vzniklá žádost o přerušení. Poté se nejnižší bit portu musí nastavit opět do 1.

Čtením vstupního portu na adrese 302h lze získat stav osmi přepínačů DIP. K realizaci tohoto portu je použit pouze třístavový budič 74541.

¹ ISA = Industry Standard Architecture, 16bitová sběrnice pro PC AT, navržena v roce 1984

² Programovatelné logické pole, EECMOS

Zdroj přerušení

Zdrojem maskovatelného hardwarového přerušení (IRQ) je tlačítko, které je ošetřeno proti zákmitům. Požadavek na přerušení, generovaný stiskem tlačítka je uschován v klopném obvodu typu RS. Sběrnice ISA reaguje na hladinu H na přerušovací vodiči, proto je potřeba jako potvrzení obsluhy žádosti nulovat klopný obvod.

Pomocí propojky na desce je možno volit různé číslo požadavku na přerušení. Na výběr je IRQ 3, 5 a 7. Omezení výběru plyne z mapy obsazení jednotlivých přerušovacích linek standardními periferiemi PC.

Přerušovací systém PC

Na PC/AT je 15 linek hardwarového přerušení (IRQ) zapojených přes 2 kaskádně spojené řadiče přerušení typu 8259. HW přerušení jsou mapována do 256ti vektorů přerušení (INT) procesorů Intel řady x86 podle následující tabulky:

IRQ	INT	Popis
IRQ0	08	Timer (55 ms intervals, 18.2 per sekund)
IRQ1	09	Keyboard service requiered
IRQ2	0A	Slave 8259 or EGA/VGA vertical retrace
IRQ8	70	Real time clock
IRQ9	71	Software redirected to IRQ2
IRQ10	72	Reserved
IRQ11	73	Reserved
IRQ12	74	Mouse interrupt
IRQ13	75	Numeric coprocessor error
IRQ14	76	Fixed disk controller
IRQ15	77	Reserved
IRQ3	0B	COM2 or COM4 service requiered
IRQ4	0C	COM1 or COM3 service requiered
IRQ5	0D	Fixed or data request from LPT2
IRQ6	0E	Floppy disk service required
IRQ7	0F	Data request from LPT1

První (master) řadič přerušení 8259 obsluhuje IRQ0 až IRQ7, druhý (slave) pak IRQ8 – IRQ15. Řadiče jsou zapojeny do kaskády, slave řadič je připojen na master přes IRQ2, na ně je směrováno IRQ9.

Každý řadič má dvě adresy v I/O prostoru. Pro řadič master je to 20h a 21h, pro slave pak A0h a A1h.

Je důležité po obslužení přerušení (nejlépe na konci obslužné rutiny) poslat na port 20h číslo 20h. Tím se dá řadiči najevo, že obsluha přerušení je u konce a může se tudíž vyvolat další.

Povolení či zakázání některé linky žádosti o přerušení se děje modifikací příslušného bitu na portu 21h. Má-li být přerušení povoleno, musí být příslušný bit nulován. Hodnoty ostatních bitů musí zůstat nezměněny!

Funkce pro obsluhu přerušení

Překladače C a C++ pro DOS umožňují, abychom si definovali své vlastní funkce pro obsluhu přerušení. Každá taková funkce musí být typu **void interrupt**. Její parametry mohou být registry procesoru 80x86, a to v pořadí:

bp, di, si, ds, es, dx, cx, bx, ax, ip, cs, flags.

Deklarujeme je jako parametry typu unsigned. Za registrem flags mohou následovat další parametry.

Funkce pro obsluhu přerušení vždy nejprve uloží na zásobník obsah registrů AX, BX, CX, DX, ES, DS, SI, DI a BP. (Registry CS, IP a FLAGS uložil již procesor před jejím voláním.)

Parametry funkce pro obsluhu přerušení pak zpřístupňují v těle funkce právě tyto uložené hodnoty.

Formální parametry bp, ..., cs, flags, odkazují právě na tyto uložené obsahy registrů. (Proto je rozumné použít uvedené identifikátory a dodržet jejich pořadí.)

Před návratem z funkce se obsahy registrů automaticky obnoví. Pokud změníme ve funkci pro obsluhu přerušení hodnotu některého z parametrů bp, ..., flags, použije se změněná hodnota.

Pokud ve funkci pro obsluhu přerušení chceme užívat aritmetické operace s reálnými čísly, musíme nejprve uložit stav koprocessoru 80x87 a před návratem jej obnovit.

Použití

K instalaci funkce pro obsluhu přerušení slouží funkce setvect(). Chceme-li získat ukazatel na aktuálně instalovanou funkci pro obsluhu určitého přerušení, použijeme funkci getvect().

Poznámky

- Překlad funkce pro obsluhu přerušení končí instrukcí IRET (návrat z přerušení). Tato instrukce způsobí skok na vzdálenou adresu uloženou na vrcholu zásobníku a přitom ze zásobníku vyjme návratovou adresu a obsah registru FLAGS. Při překladu ostatních funkcí se na PC užívá instrukce RET, která nezahrnuje manipulaci s registrem příznaků.
- Funkce pro obsluhu přerušení můžeme volat i jako obyčejné funkce.

Postup práce:

1. Nakopírujte si soubor **irq.c** z adresáře **C:\IRQ** do Vašeho pracovního adresáře **C:\PISECEK**. Tato šablona obsahuje důležité části programu, které můžete využít při řešení úlohy. Zbylé části viditelně označené je třeba doplnit. Důkladně šablonu prostudujte.
2. Vytvořte program, který nastaví správně vektor přerušení
 - a. Úschova původního vektoru přerušení.
 - b. Nastavení nového vektoru přerušení.
 - c. Nulování klopného obvodu zápisem 0 a následně 1 na nejnižší bit výstupního portu 300h.
 - d. Povolení přerušení v řadiči přerušení (adresa 21h).
 - e. Čekat na stisk klávesy
 - f. Zakázání přerušení v řadiči přerušení (adresa 21h).
 - g. Vrátit původní vektor přerušení.
3. Vytvořte obslužnou rutinu hardwarového přerušení.
 - a. Vyvolání obslužné rutiny se musí nějak projevit (vypsání něčeho na obrazovku nebo pípnutí apod.)
 - b. Potvrzení obsluhy přerušení v adaptéru (krátké vynulování nejnižšího bitu na portu 301h)
 - c. Potvrzení obsluhy přerušení v řadiči přerušení.
4. Přeložení vašeho **irq.c** můžete provést buď přímo v Turbo C (spouští se příkazem **tc**) nebo pomocí příkazu **tcc irq.c**

Některé užitečné funkce:

```
void outportb(int port, unsigned char value); //zápis na port, definováno v <dos.h>
unsigned char inportb(int port); //čtení z portu, definováno v <dos.h>
void interrupt (*getvect( int intr_num ))(); //získání ukazatele na funkci obsluhy přerušení
void setvect( int intr_num, void interrupt(*isr)()); //nastavení vektoru přerušení
int getch(); //čtení znaku z klávesnice
```

Detailní schéma adaptérové desky

