

Program

- Periferie počítače z hlediska programátora
- DMA – jak to funguje co to je
- IRQ – co to je, řadič přerušení
- IO – co je to ioport
- PCI address space
- RTC, Klávesnice, řadič přerušení
- BCD kód
-
- Cvičení: výpis data a času z RTC, blikání s LED klávesnice, výběr počítačové sestavy

Nakupuje počítač

- V čem je třeba mít jasno
 - typické práce, které se budou s počítačem vykonávat
 - cenový strop, alias peníze co jsme ochotni do sestavy investovat
- Kritéria výběru:
 - Monitor
 - CRT versus LCD
 - tj klasický monitor
 - 17 palců (úhlopříčka obrazu)
 - 19 palců (úhlopříčka obrazu)
 - LCD
 - zpoždění displaye
 - pozorovací úhel
 - velikost úhlopříčky

Nakupujeme počítač II

- CASE
 - tower, minitower
 - výkon zdroje
- základní deska
 - kolik lze osadit paměti
 - jaké jsou možnosti rozšíření
 - jaké periferie jsou na desce integrovány
 - jaká patice pro procesor je použita
- Procesor
 - výkonostní kritéria
 - jak dobře jde procesor uchládit
- Disky mechaniky
 - cena za gigabyte
 - DVD RW „region free“

Sestava

- Skupina1
 - 33588 – Asus deluxe A8V, kingstone 2x 512 DDR433, AMD64 (3000) 3500 socket 939, chladic zelman, S3, ASUS DVDRW dual, case 400W eurocase, disky 200GB 3x maxtor, 8MB kes
- Skupina2
 - 34500 – Abit AV8, apace 2x 512MB ECC DDR 400, Athl 64, 3200,socket 939, box chladic, XXX, Plextor SATA 12x dual, eurocase 400W, Seagate 200GB sata 8MB kes 3x
- Skupina3
 - 33423 – Asus K8N deluxe, 2x 512MB 400MHZ Adata, 3GHz A64, socket 754, chladic coolermaster, geforce mx440 64mb, pionyr 16x 4x rw dual, 350W case, 2x seagate 200GB 8MB kes SATA, UPS apc.
- Skupina4
 - 26000 – microcstar kt6d 2x4, 256DDR400Mhz, AMD K7 2200+, box, microstar ms8911, gsa 4160B 2vrstvy DVDRW, KME 350W, Maxtor 200GB SATA 8MB

Sestava

- Skupina1
 - 30683 – Asus deluxe A8V, 256MB DDR433, AMD64 (3000) socket 939, chladic zelman, S3, ASUS DVDRW dual, case 400W eurocase, diský 200GB 3x maxtor, 8MB kes
- Skupina2
 - 33100 – Abit AV8, apace 256MB DDR 400, Athl 64, 3200,socket 939, chladic articfreezer, gxm64 geforce, Plextor SATA 12x dual, eurocase 400W, Seagate 200GB sata 8MB kes 3x , UPS 400W
- Skupina3
 - 33972 – Asus deluxe A8V, 256MB 400MHZ Adata, 3GHz A64, socket 939, chladic coolermaster, geforce mx440 64mb, pionyr 16x 4x rw dual, 350W case, 3x seagate 200GB 8MB kes SATA, UPS apc.
- Skupina4
 - 32460 – microcstar kt6d 2x4, 256DDR400Mhz, AMD K7 3000+, vortext cooling acc72, microstar ms8911, gsa 4160B 2vrstvy DVDRW, KME 350W, Maxtor 200GB SATA 8MB 3x, UPS

Periferie z hlediska programátora

- Drtivá většina procesorů má stejnou instrukci pro přístup do paměti i do „periferního zařízení“.
- Konfigurační registry zařízení jsou namapována do adresního prostoru procesoru
 - počítač 32bit adresová sběrnice
 - max 4GB RAM.
 - např. posledních 128MB adres je vyhrazeno pro per. zařízení
 - zápisem a čtením z jistých adres se mění konfigurace zařízení
- I/O – vstup výstup
- Rodina procesorů x86 však adresový prostor pro periferie od přístupu do paměti oddělila
 - mají speciální instrukce pro komunikaci s periferními zařízeními
 - jednotlivé adresy se jmenují „porty“
 - je však možné použít i první metodu a namapovat periferii do paměti

Mapa IO portů typického počítače

```
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-006f : keyboard
0070-0077 : rtc
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
01f0-01f7 : ide0
0290-0297 : HW monitoring IO
02f8-02ff : serial
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial
0cf8-0cff : PCI conf1
9400-94ff : 0000:00:0c.0
9400-94ff : 8139too
```

```
cat /proc/ioproports
```

Ve windows viz správce zařízení

Mapa adresního prostoru

```
00000000-0009fbff : System RAM
0009fc00-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c7fff : Video ROM
000f0000-000fffff : System ROM
00100000-0ffffbfff : System RAM
    00100000-002d3886 : Kernel code
    002d3887-0038437f : Kernel data
0fffc000-0ffffefff : ACPI Tables
0ffff000-0fffffff : ACPI Non-volatile Storage
ed800000-ed8000ff : 0000:00:0c.0
    ed800000-ed8000ff : 8139too
```

```
cat /proc/iomem
```


Přerušeni

- Procesor má implementován (je vybaven) mechanismem podporující událost přerušeni
- při přerušeni se přestane provádět dosud prováděný program, dočasně se začne provádět program pro obsluhu přerušeni. Po jeho dokončení se pokračuje od místa kde byl původní program přerušeni.
- přerušeni
 - hardwarová
 - softwarová (instrukce INT)
- IRQ
 - interrupt service request
 - zařízení žádá prostřednictvím přerušeni o obsluhu
 - na platformě PC bylo vyhrazeno 16 přerušovacích „linek“ , viz sběrnice ISA
 - IRQ0 - IRQ15

Obsazenost přerušení

Standard Interrupt Assignments

IRQ	Priority	Standard Function
0	1	System Timer
1	2	Keyboard Controller
2	N/A	Programmable Interrupt
3*	11	Communications Port (COM2)
4*	12	Communications Port (COM1)
5*	13	Sound Card (sometimes LPT2)
6	14	Floppy Disk Controller
7*	15	Printer Port (LPT1)
8	3	System CMOS/Real Time Clock
9*	4	ACPI Mode when used
10*	5	IRQ Holder for PCI Steering
11*	6	IRQ Holder for PCI Steering
12*	7	PS/2 Compatible Mouse Port
13	8	Numeric Data Processor
14*	9	Primary IDE Channel
15*	10	Secondary IDE Channel

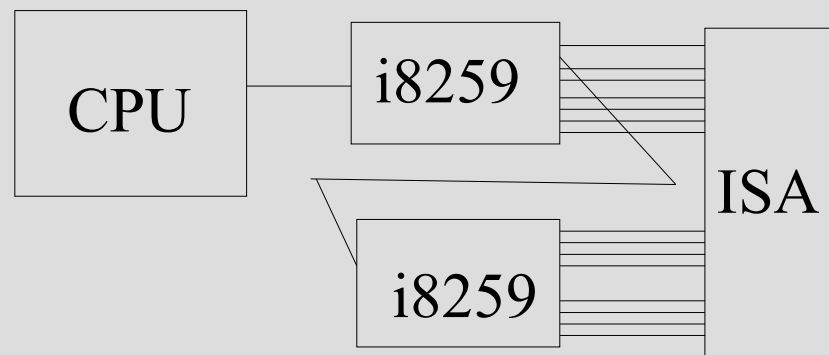
*These IRQs are usually available for ISA or PCI devices.

Přerušeni

```
ruik@ruik:~/board$ cat /proc/interrupts |less
          CPU0
 0:      205072258          XT-PIC  timer
 1:         14059          XT-PIC  i8042
 2:              0          XT-PIC  cascade
 4:              6          XT-PIC  serial
 5:              5          XT-PIC  ohci_hcd, eth1
 8:              4          XT-PIC  rtc
 9:              0          XT-PIC  ohci_hcd
10:      4796478          XT-PIC  eth0, CMI8738-MC6
11:     17427732          XT-PIC  r128@PCI:1:0:0
12:      132690          XT-PIC  i8042
14:      194182          XT-PIC  ide0
```

Přerušení a sběrnice ISA

- Ve sběrnici skutečně vyhrazeno 16 vodičů pro potřeby přerušení
- Procesor má však jen jeden přerušovací vstup.
- Mezi procesorem a sběrnicí je umístěn řadič přerušení
 - 2x intel 8259A
 - je programovatelný porty 0x20 a 0x21 a 0xA0 a 0xA1
-

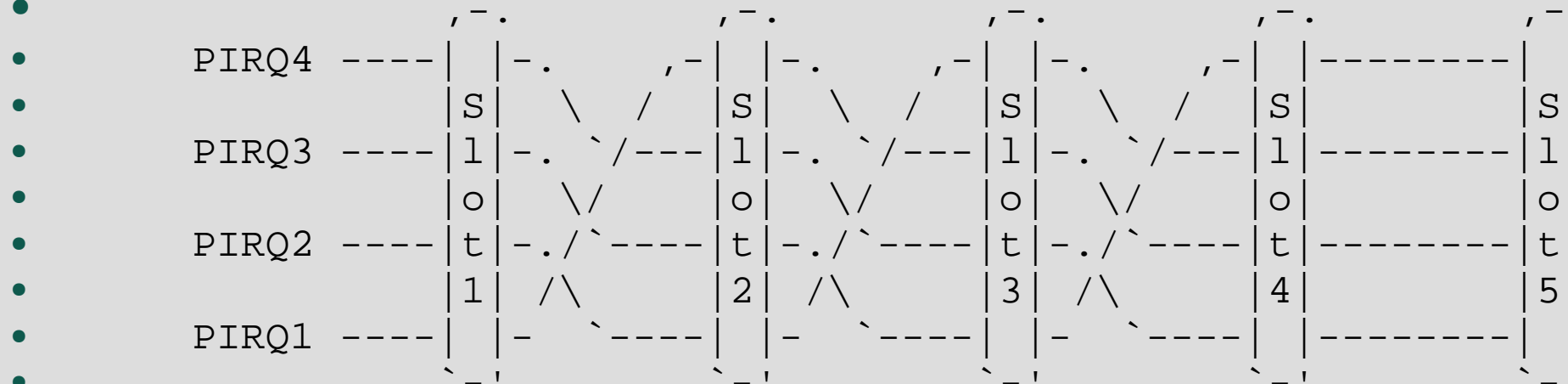


Přerušování a sběrnice PCI

-
- INTD-- | |
- | S |
- INTC-- | 1 |
- | 0 |
- INTB-- | t |
- | x |
- INTA-- | |
- | - |

Z PCI karty vedou 4 přerušovací linky, většina používá jen jedno IRQ tj INTA

- Do chipsetu jsou vyvedeny 4 piny, které se připojí na některé z
- IRQ0 – IRQ15



Instrukce pro I/O

- Čtení:
 - in registr1,registr1
 - registr1 – 8, 16, 32 bitů – vždy AL AX EAX – načtená hodnota
 - registr2 – vždy 16 bitový a vždy DX – číslo portu se kterým se komunikuje
 - in ax,dx;načte do AX šestnáctibitovou hodnotu z portu, jehož číslo je uloženo v registru DX
 - in in al,dx
 - in al,port (0x0-0xFF)
 - in al,0x20
- Zápis:
 - obdobně
 - out dx,al ;zapíše hodnotu AL do portu specifikovaném v DX
 - out dx,ax
 - out dx,eax

Přístup do paměti CMOS

- Do paměti CMOS přistupujeme pomocí dvojice portů
- 0x70 – zápis adresy
- 0x71 – zápis/čtení dat
-
- čtení 0 bajtu
 - `mov al,0x0`
 - `out 0x70,al`
 - `in al, 0x71`
- zápis je příliš nebezpečný
 - myslím že na něj každý přijde
 - adresa 0x2E obsahuje kontrolní součet – její přepsání je to samé co `cmospwd /k`

Struktura paměti CMOS

Offset Hex	Offset Dec	Field size	Function
00h	0	1 byte	RTC seconds. (BCD Format)
01h	1	1 byte	RTC seconds alarm. (BCD Format)
02h	2	1 byte	RTC minutes. (BCD Format)
03h	3	1 byte	RTC minutes alarm. ((BCD Format)
04h	4	1 byte	RTC hours. (BCD Format)
05h	5	1 byte	RTC hours alarm. (BCD Format)
06h	6	1 byte	RTC day of week.(1 .. 7, sunday=1)
07h	7	1 byte	RTC date day. (BCD Format)
08h	8	1 byte	RTC date month. (BCD Format)
09h	9	1 byte	RTC date year. (BCD Format)
12h	18	1 byte	Hard Disk Types Bits 7-4 = Hard disk 0 type Bits 3-0 = Hard disk 1 type 0000 = No drive installed 0001 = Type 1 installed 1110 = Type 14 installed 1111 = Type 16-47 (defined later in 19h)

BCD kód

- BCD – binary coded digit
 - v jednom bajtu dvě číslice
 - 4 bity kódují číslice 0-9
 -
 - 0x12 je 12 desítkově 0001 0010
- BCD na číslo
 - vrchní 4 bity vynásobíme deseti a přičteme spodní 4 bity
- Číslo na BCD
 - číslo vydělíme 10, zbytek zapíšeme v dolních 4 bitech, výsledek v horních 4.
- zkuste napsat jednoduchý podprogram, nebo aspoň vývojový diagram

Úkol na cvičení

- Vypsát datum a čas na obrazovku.
- Zjistit typ disku

Klávesnice připojení

- klávesnice je s PC propojena sériovým rozhraním
 - DIN5 – velký konektor
 - PS/2 konektor – malý konektor
 - +5V, GND, CLOCK, DATA
- klávesnice posílá počítači tzv scankódy (scancodes)
 - obsahují číslo stisknuté klávesy
 - pokud je stisknuto víc kláves odvysílají se postupně
 - pokud klávesu držíme posílá se jiný kód, než když klávesu právě uvolníme
 - scancode poslední stisknuté klávesy je možné přečíst z IO portu 0x60
 - klávesnice rozumí různým příkazům
 - rozsvít' LED diody
 - nastav interval opakování
 - ...

Klávesnice prog. pohled

- v PC je keyboard controller – řadič klávesnice
 - porty 0x60 a 0x64
 - jsou definovány příkazy pro komunikaci buď
 - i8042 mikroprocesor – což je řadič klávesnice
 - přímo příkazy pro klávesnici které zašle řadič
 - blikáme s LED
- pokud budeme posílat příkazy řadiči klávesnice musíme se ujistit že je přijal (status register port 0x64)

```
MSb                                                    LSb
+-----+
| PERR | RxTO | TxTO | INH | A2 | SYS | IBF | OBF |
+-----+
```

OBF (Output Buffer Full) - Indicates when it's okay to write to output buffer.

0: Output buffer empty - Okay to write to port 0x60

1: Output buffer full - Don't write to port 0x60

Klávesnice prog. pohled

- pokud budeme posílat příkazy řadiči klávesnice musíme se ujistit že je přijal.

kbWrite:

WaitLoop:

```
in    al, 64h      ; Read Status byte
and   al, 01b      ; Test OBF flag (Status<0>)
jnz   WaitLoop     ; Wait for OBF = 0
out   60h, cl      ; Write data to output buffer
```

DMA

- http://en.wikipedia.org/wiki/Direct_memory_access
- Direct memory access (DMA)
 - přímý a nezávislý přístup periferií o paměti bez CPU
 - hardware může číst nebo zapisovat do paměti bez toho aby se používal procesor - instrukce IN a OUT – tzv PIO režim
 - Co používá DMA
 - Hard Disk Controller,
 - Disk Drive Controller,
 - (Graphics Card)
 - Soundcard.
- Jak funguje:
 - přenos DMA kopíruje blok paměti do periferního zařízení (nebo naopak)
 - např. zvuková data
 - naprogramuje se přenos (instrukcemi pro IO)
 - zvuková karta přehrává data která do ní proudí přes DMA
 - jakmile se oblast paměti „přehraje“ vyvolá se přerušení a v něm se naprogramuje nový přenos, CPU se tak může „věnovat“ např. počítačové hře (kterou vykonává) a jen občas se zabývat zvuk. kartou

DMA

- Jak DMA probíhá?
 - musí být naprogramován přenos DMA
 - pokud je, požádá řadič DMA procesor o přístup k paměťové sběrnici
 - DRQ (DMA REQUEST)
 - jakmile procesor vyhoví, řadič DMA začne sběrnici používat a generovat čtení nebo zápisy do paměti a zapisovat nebo číst z daného zařízení (zvuk. karta)
 - Přístup ke sběrnici procesor dovolí jen ve chvílích kdy zpracovává instrukci, tj na začátku a na konci strojového cyklu