

Metody připojování periférií

BI-MPP

Přednáška 2

Ing. Miroslav Skrbek, Ph.D.

Katedra číslicového návrhu

Fakulta informačních technologií

České vysoké učení technické v Praze

Miroslav Skrbek ©2010-2021

ZS2021/22



EVROPSKÁ
UNIE

Evropský sociální fond

Praha & EU: Investujeme do vaší budoucnosti

Agenda

- Adresní prostory PC
- Překlad adres
- V/V operace
- Přímý přístup do paměti

Literatura

- Gook, M.: Hardwarová rozhraní – Průvodce programátora. Computer Press, Brno 2006. ISBN 80-251-1019-2

Adresní prostor (opakování)

- Souvislý rozsah adres generovaný procesorem
- Velikost dána počtem adresních bitů, velikost je vždy mocnina dvou
- Nejmenší adresovatelná datová jednotka v adresním prostoru může být:
 - 1 bit (ve speciálních případech)
 - 1 slabika (1 byte), nejčastější (PC)
 - 1 slovo 16-bitů, 32-bitů, 64-bitů atp.
- Do adresního prostoru se mapuje fyzická paměť (tj. paměť v paměťových čípech) nebo registry
- Adresní prostor nemusí být vždy celý vyplněn fyzickou pamětí
- Pokud procesor podporuje více adresních prostorů, neznamená to, že každý prostor bude mít svou vlastní adresovou sběrnici. Adresní prostory obvykle sdílí jednu adresovou sběrnici a jednotlivé adresní prostory jsou odlišeny oddělenými čtecími a zápisovými signály



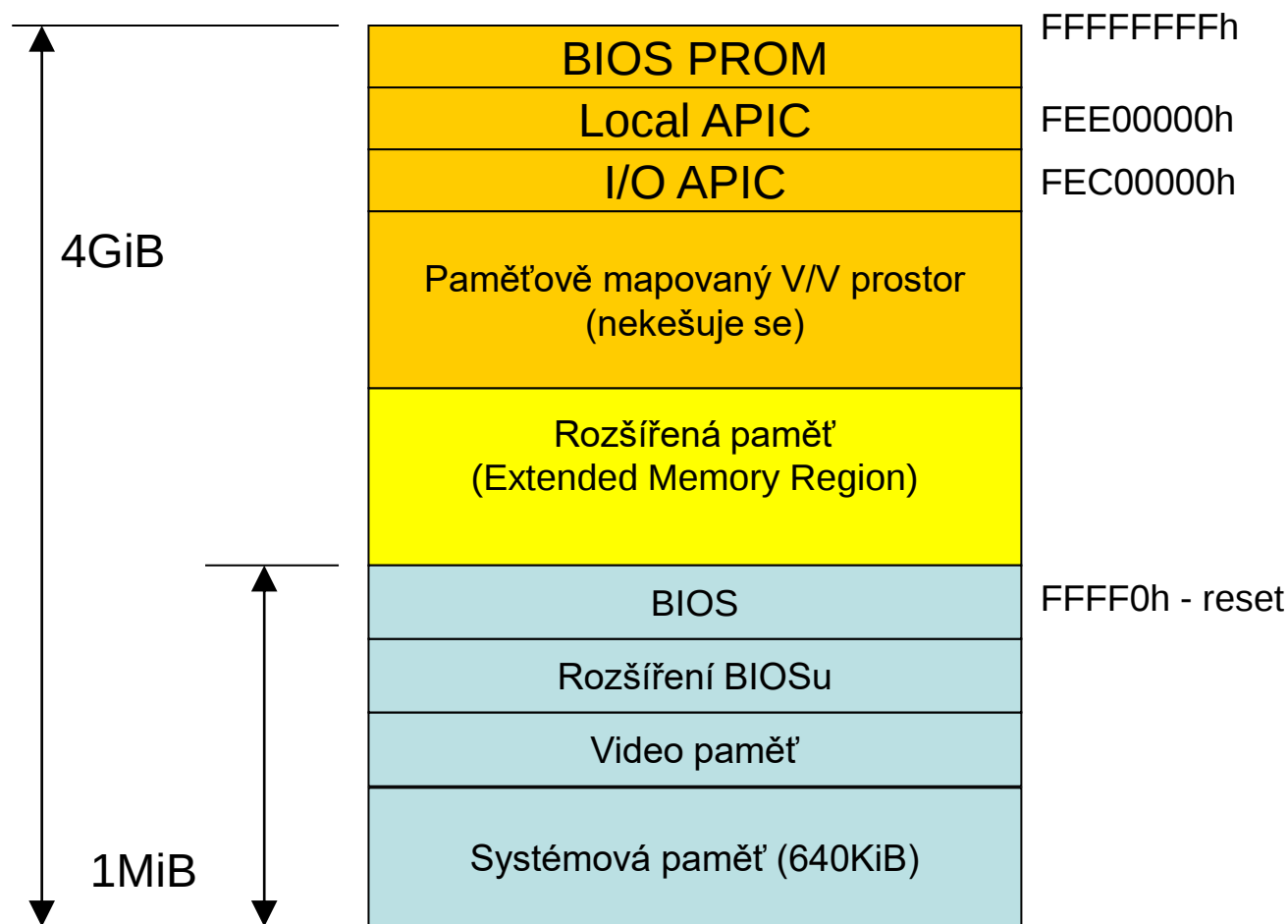
Adresní prostory PC

- Fyzický paměťový prostor
 - Do tohoto prostoru jsou mapovány paměťové moduly
- Virtuální paměťový prostor
 - V tomto prostoru adresují aplikace
 - Každá aplikace má svůj adresní prostor
 - Operace čtení a zápis v tomto prostoru podléhají překladu adres (v pořadí segmentace, stránkování)
 - Do virtuálního adresního prostoru se
 - mapují části fyzické paměti
 - části fyzické paměti odložené na disku (stránkovací soubor, swap partition)
 - Zbytek je neobsazen a jakýkoliv přístup vede k výjimce
- Vstupně/výstupní
 - Velikost 64KiB, mapují se sem registry periférií

Fyzický adresní prostor

- V tomto adresním prostoru je mapována operační paměť
- Na 32 bitové architektuře má tento prostor velikost 4GiB
- Do tohoto adresního prostoru se mapují paměťové bloky periférií (např. videokarty)
- K fyzickému paměťovému prostoru má přístup pouze jádro operačního systému
- Operační systém mapuje části fyzického adresního prostoru do virtuálních prostorů aplikací

Mapa fyzické paměti PC



Virtuální paměťový prostor

- Typicky paměťový adresní prostor přiřazený aplikacím.
- Do tohoto prostoru se mapují části fyzického adresního prostoru metodou
 - Segmentace
 - Stránkování
 - Segmentací s následným stránkováním
- Překlad adres z virtuálního paměťového prostoru do fyzického je hardwarový
 - Překládá se na základě překladových tabulek
 - Tabulky upravuje operační systém

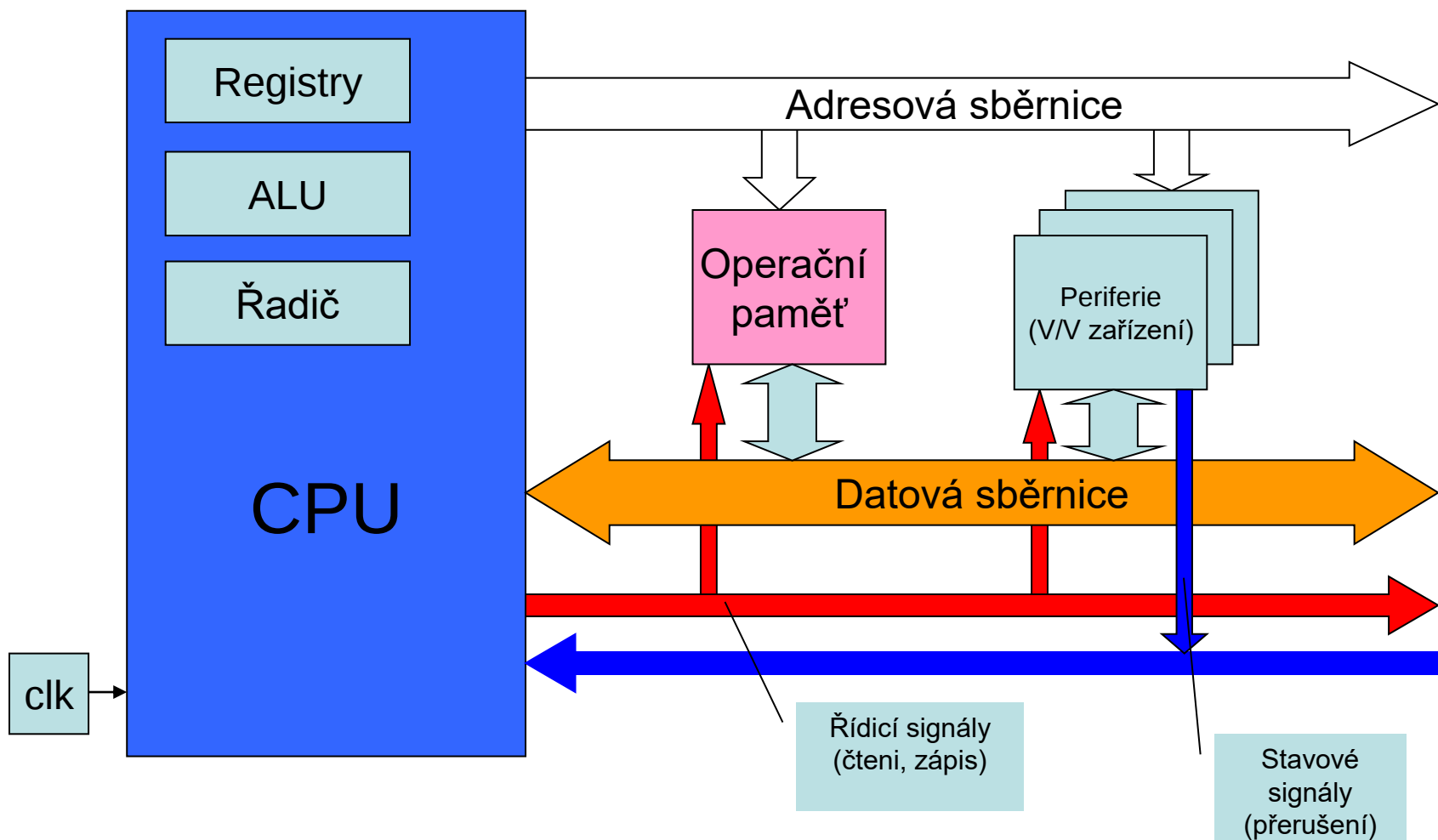
Segmentace

- Segment je část virtuálního (při vypnutém stránkování přímo fyzického) adresního prostoru určená svým počátkem a délkou
- Segment je určen záznamem v tabulce deskriptorů
- Procesor obsahuje segmentové registry, které obsahují ukazatele do tabulky deskriptorů
 - CS registr (Code segment, čtení instrukcí)
 - DS registr (Data segment, proměnné programu)
 - SS registr (Stack segment, zásobník programu)
 - ES, FS, GS registr (další segmentové registry)
- Registry jako IP (Instruction pointer) jsou posunutím (offsetem) v rámci segmentu CS, např. EBX je posunutím DS (MOV DS:[EBX], EAX; MOV [EBX], EAX)
- Při zapnutém stránkování jsou segmenty mapovány do fyzické paměti stránkovacím mechanismem.
- Operační systémy jako Linux obchází segmentaci (nelze totiž vypnout) nastavením segmentů na celý adresní prostor.

Stránkování

- Fyzický adresní prostor je rozdělen na stejně velké stránky (typicky 4KiB)
- Jednotlivé stránky se v různém pořadí mapují do virtuálních prostorů aplikací (souvislý úsek paměti ve virtuálním prostoru nemusí být souvislým úsekem ve fyzické paměti)
- Mapování stránek se provádí přes tabulku stránek, která je umístěna ve fyzické paměti
- Tabulku stránek vytváří a modifikuje operační systém
- Překlad adresy probíhá hardwarově, na jednu operaci např. MOV EAX, [EBX] (čtení dat ve virtuálním prostoru) musí předcházet několik čtecích operací v tabulkách stránek. Pro urychlení překladu se používá asociativní paměť TLB (Translation Lookaside buffer, neplést keší).

Blokové schéma počítače



Zápis a čtení v paměťovém prostoru

Jazyk C

```
long* adr = 0xC0000000;  
long v = *adr;
```

Pozor ! Adresujeme ve
virtuálním adresovém
prostoru

Asembler

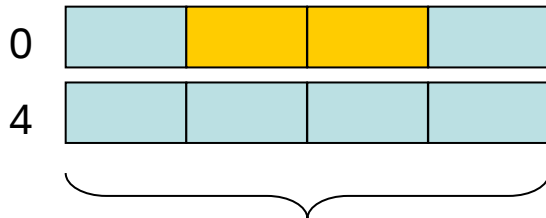
```
MOV EBX, 0xC0000000  
MOV EAX, [EBX]
```

Pokud nejsme v kontextu jádra,
ale aplikace, pak to patrně skončí
výjimkou, protože pro použitou
adresu není mapována fyzická
paměť

Nezarovnané V/V a paměťové operace

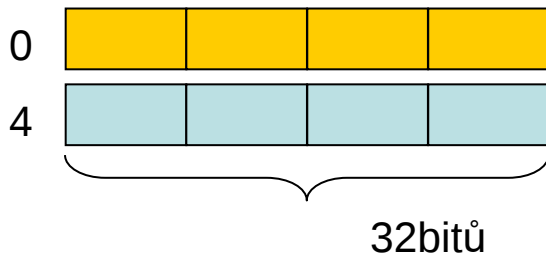
```
mov DX, 1  
in AX, DX
```

Jeden čtecí cyklus



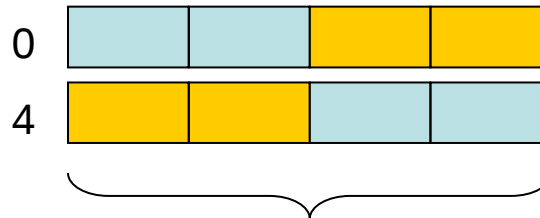
```
mov DX, 0  
in EAX, DX
```

Jeden čtecí cyklus



```
mov DX, 2  
in EAX, DX
```

Dva čtecí cykly



Pozor ! Často V/V registry nedovolují nezarovnaný zápis, případně zarovnaný zápis menší jednotky než je 32 bitů.

Příklad: adresy CF8h a CFCh, které se používají pro konfiguraci PCI karet, vyžadují zarovnané 32bitové operace.

Čtení a zápis v IO prostoru

Jazyk C - linux

```
Long v = inb(io_addr)
outb(v, io_addr)
```

Jazyk C – Windows (HAL)

```
READ_PORT_UCHAR(port)
WRITE_UCHAR(port, value)
```

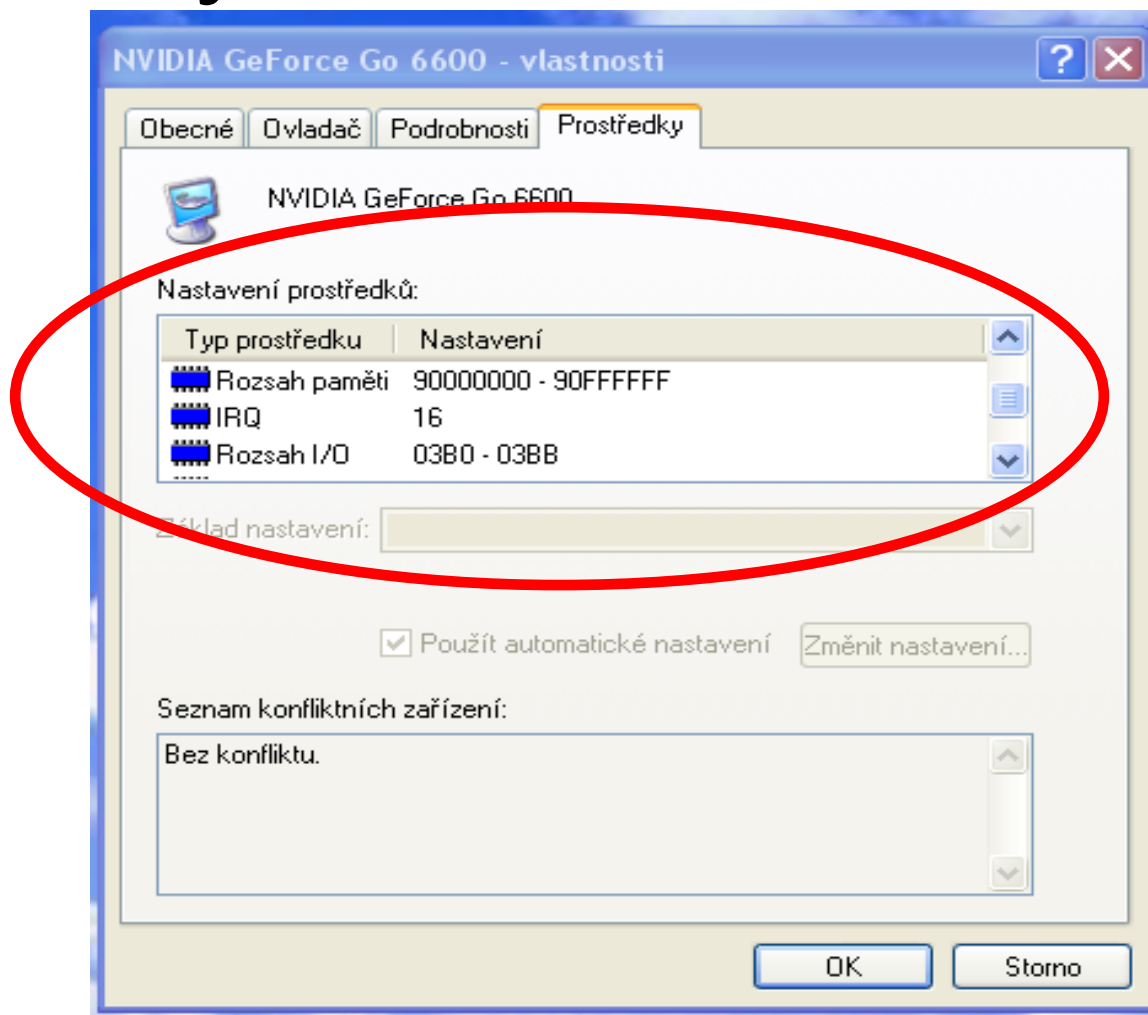
Na úrovni aplikace pokus o tyto operace končí výjimkou.

Na Linuxu je možné povolit funkcí iopl. Ve windows je nutné použít např. giveio

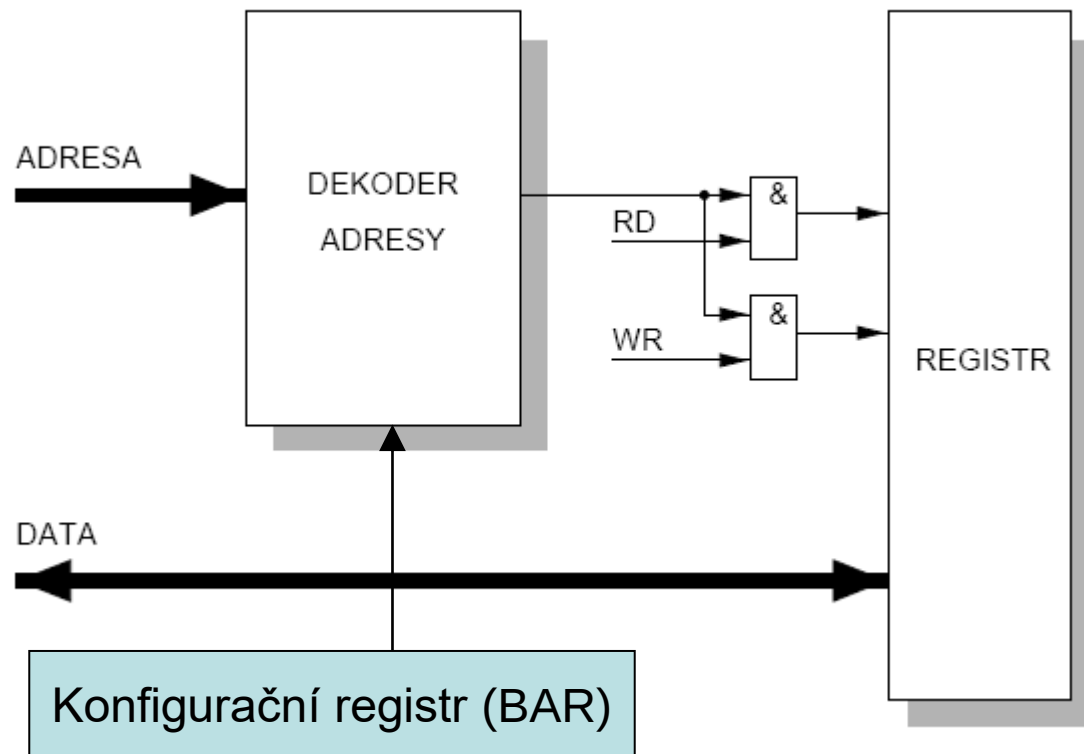
Pozor ! odblokování přístupu do I/O prostoru je v principu bezpečnostní díra !

Při přístupu k portům z aplikace je třeba uvažovat s přepínáním kontextu (např. kdyby někdo chtěl generovat časový průběh)

Prostředky alokované zařízením



Mapování registrů do V/V prostoru

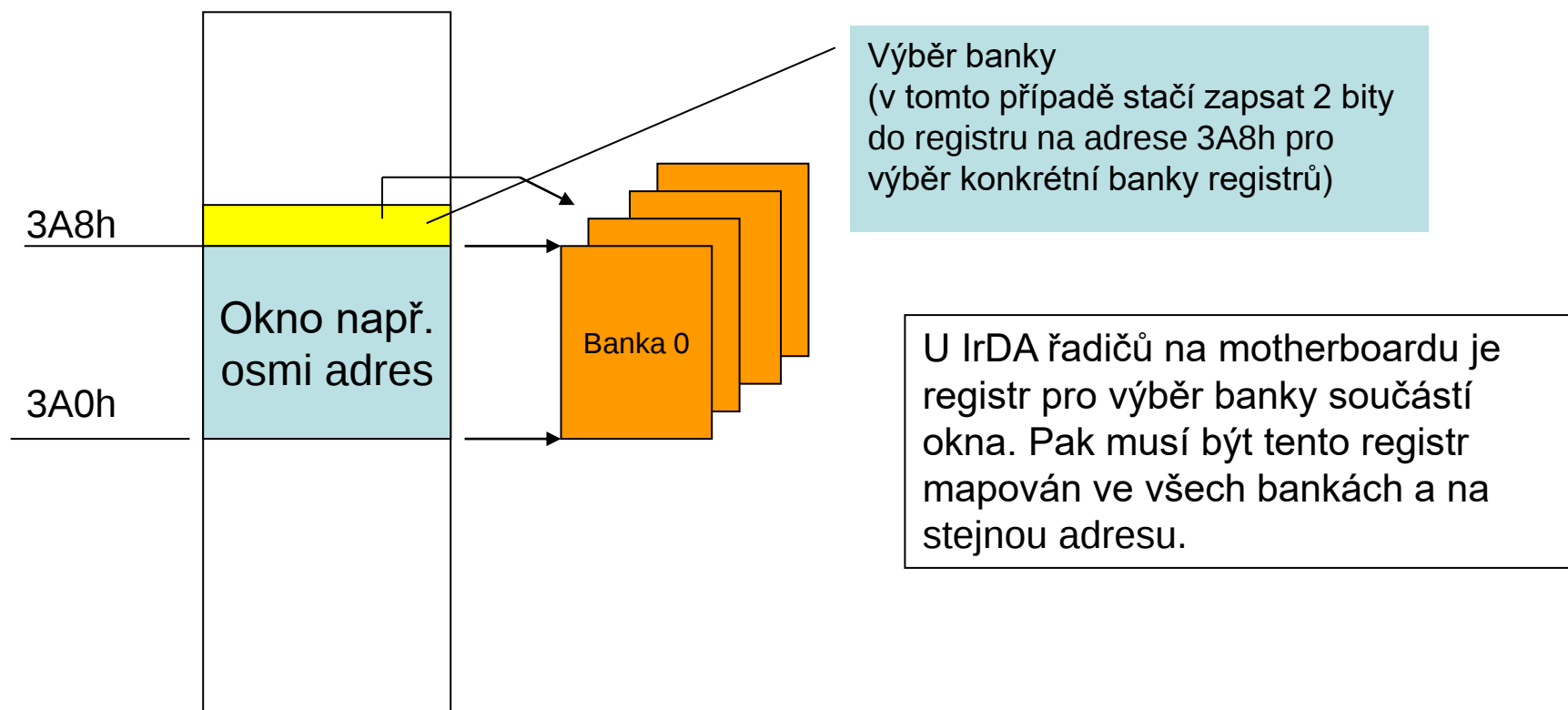


Konfigurační registry PCI karet

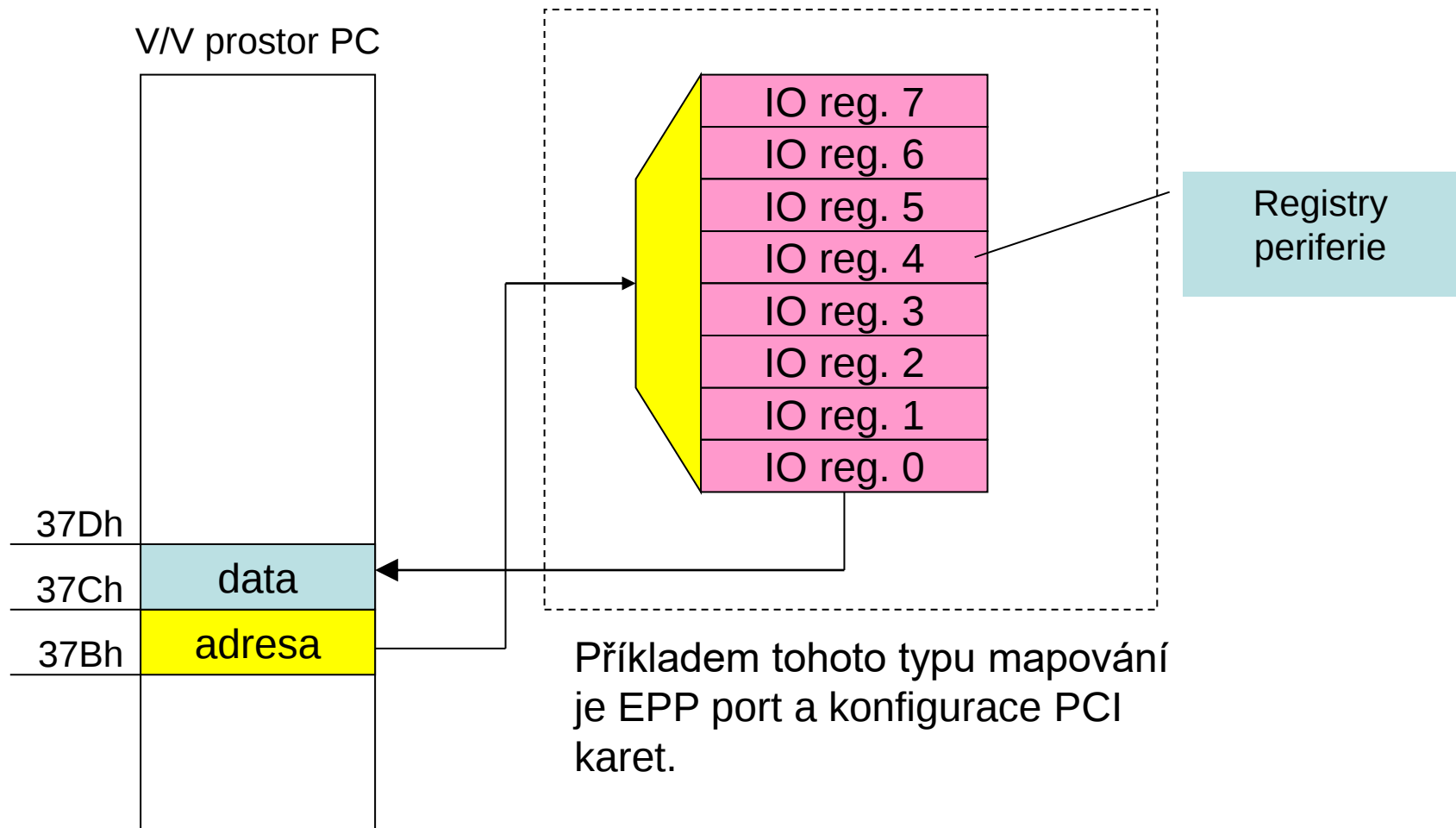
Device ID (31..16)		Vendor ID (15..0)	
Status (31..16)		Command (15..0)	
Class Code (31..8)		Revision ID (7..0)	
BIST (31..24)	Header Type (23..16)	Latency Timer (15..8)	Cache Line Size (7..0)
Base Address Register 0			
Base Address Register 1			
Base Address Register 2			
Base Address Register 3			
Base Address Register 4			
Base Address Register 5			
Cardbus CIS Pointer			
Subsystem ID (31..16)		Subsystem Vendor (15..0)	
Expansion ROM Base Address			
Reserved			
Reserved			
Max_Lat (31..24)	Min_Gnt (23..16)	Interrupt pin (15..8)	Interrupt line (7..0)

Mapování bank registrů

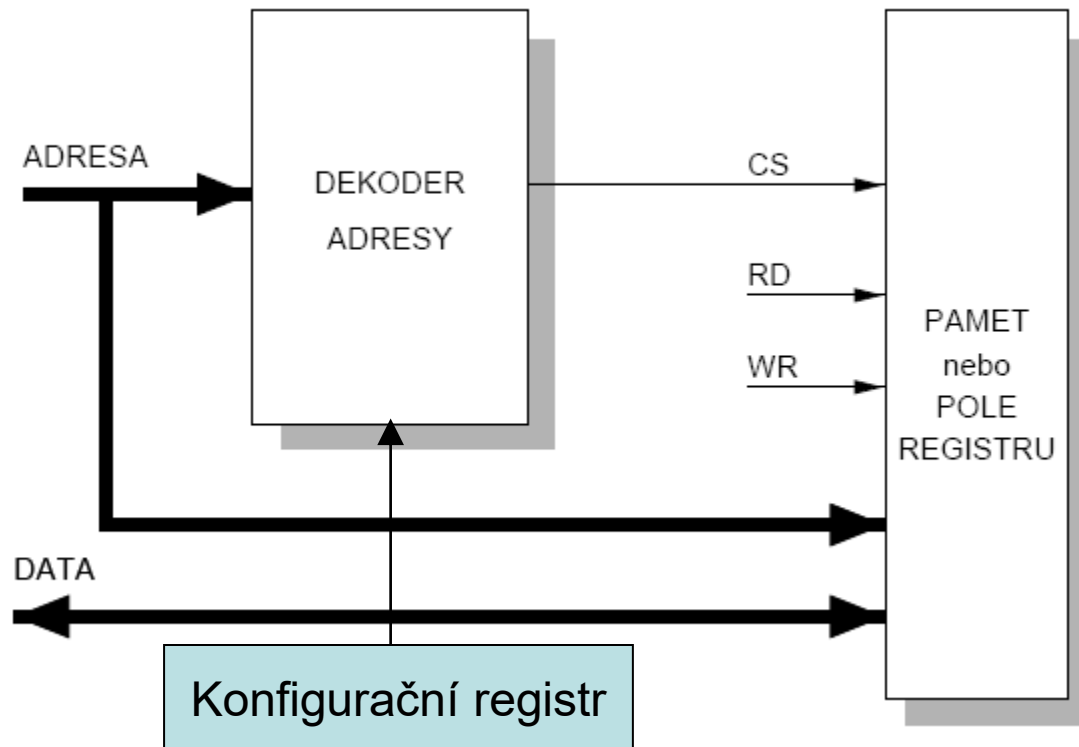
Pokud má periferie větší množství registrů, pak použije metodu mapování bank, aby se ušetřily adresy ve V/V adresním prostoru.



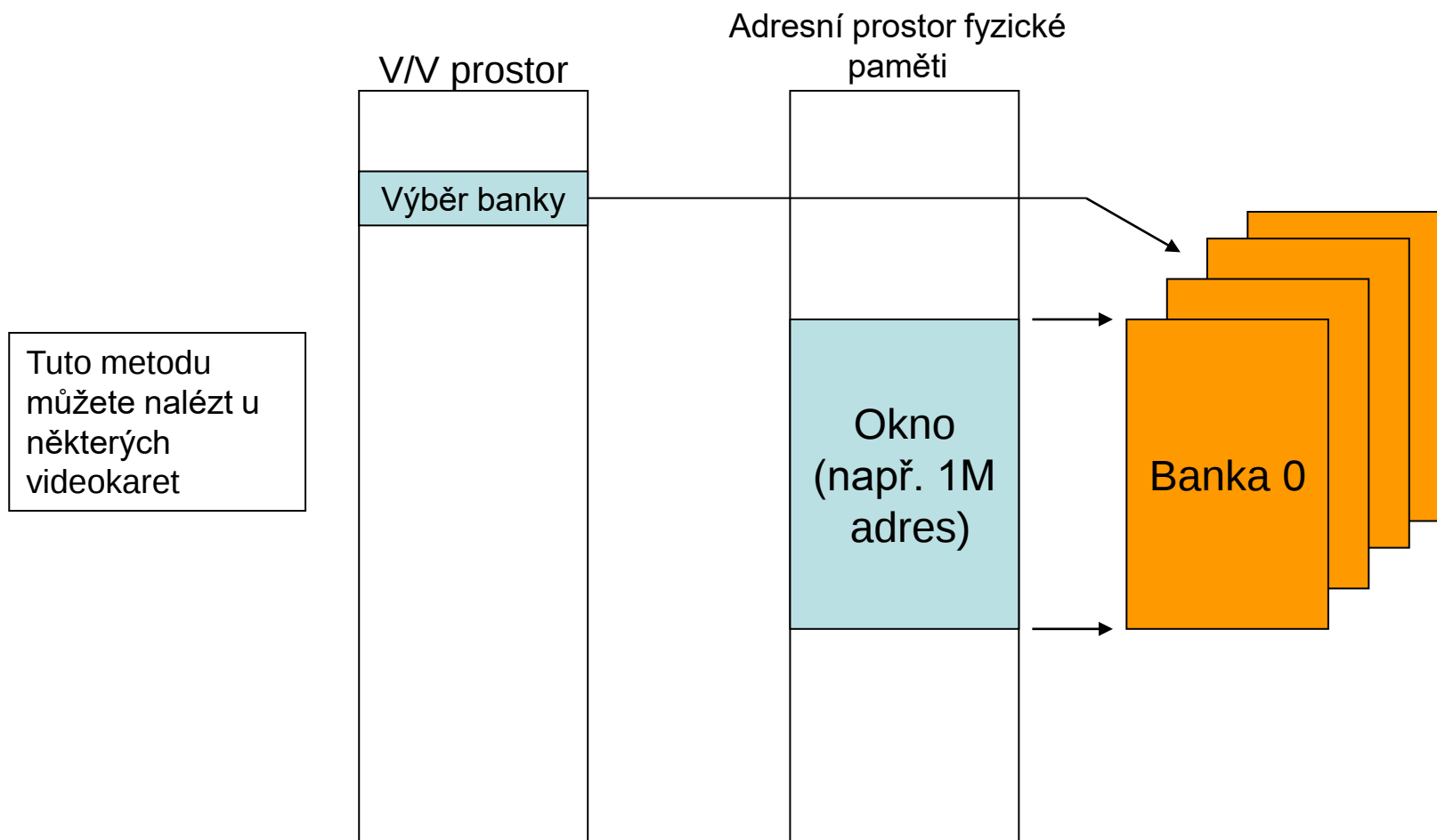
Adresační přístup k registrům



Mapování registrů nebo paměti do paměťového adresového prostoru



Mapování paměťových bank



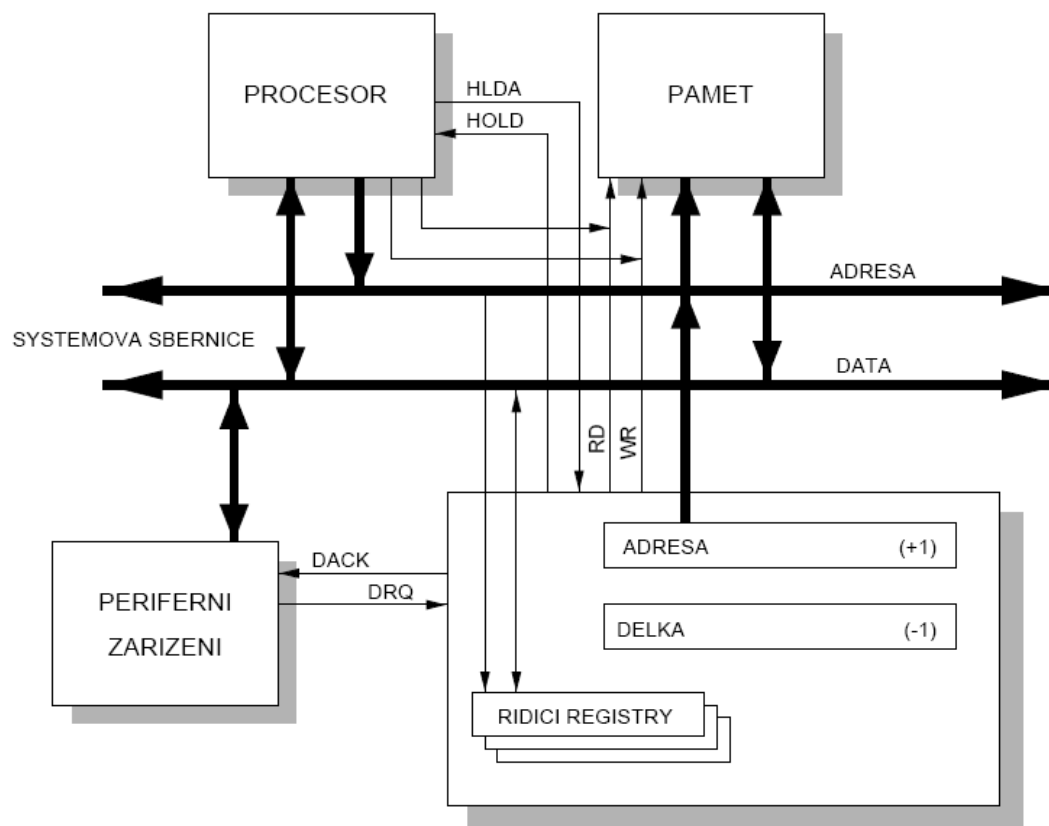
PnP přístupy při mapování registrů a paměťových bloků

- Rozmanitost V/V zařízení nedovoluje předem určit pevné adresy registrů V/V zařízení (velká pravděpodobnost kolize)
- Ruční konfigurace je uživatelsky nepohodlná
- Dnešním řešením je programová konfigurace zařízení (princip PnP)

V/V adresy v PC

- Pevné – dány historicky, udržovány z důvodu zpětné kompatibility. Např. čítač, původní řadič přerušení, DMA řadiče apod.
- Pevné – zajišťující geografické adresování periférií a přístup do konfiguračních prostorů (typicky PCI karet)
- Konfigurovatelné – bloky adres, které jde umístit téměř libovolně v adresových prostorech. Za technickou realizaci mapování je zodpovědná periferie, za konfiguraci pak setup nebo operační systém.

DMA přenos s externím řadičem



Dnes je již udržováno jen z důvodu kompatibility.

DMA typu Bus-mastering

