

# Paralelní Sběrnice

# Pokus o taxonomii paralelních sběrnic

- asynchronní
- synchronní

- paměťová
- IO

- univerzální
- „vyhrazená“

- monomaster
- multimaster

- jednoduchá
- vícenásobná

Multimaster:

- centralizovaná arbitrace
- decentralizovaná arbitrace

- nemultiplexovaná
- multiplexovaná

# Paralelní sběrnice

- ◆ Sběrnice - propojující IO zařízení s CPU, popř. s pamětí
- ◆ Měření výkonu I/O

**Čím se budeme zabývat:**

**Různé sběrnicevé mechanismy.**

**Jednoduše řečeno, sběrnice představuje propojení různých obvodů/komponent v počítači.**

**Sběrnice zajišťuje řízení a přenos dat mezi těmito různými komponentami.**

- ◆ Spojovací podsystém - „stmeluje“ komponenty počítačového systému
- ◆ Vysokorychlostní hardwarový interface + logické protokoly
- ◆ Sítě, kanály, sběrnice (backplanes)

	<b>Sít'</b>	<b>Kanál</b>	<b>Backplane</b>
<b>Propojuje</b>	Počítače	Zařízení	Obvody
<b>Vzdálenost</b>	>1000 m	10 - 100 m	0.1 m
<b>Šíře pásma</b>	10 - 1000 Mb/s	40 - 1000 Mb/s	320 - 6000+ Mb/s
<b>Latence</b>	velká ( 1ms)	střední	malá (nanosec.)
<b>Spolehlivost</b>	malá Extensivní CRC	střední parita-byte	vysoká parita-byte

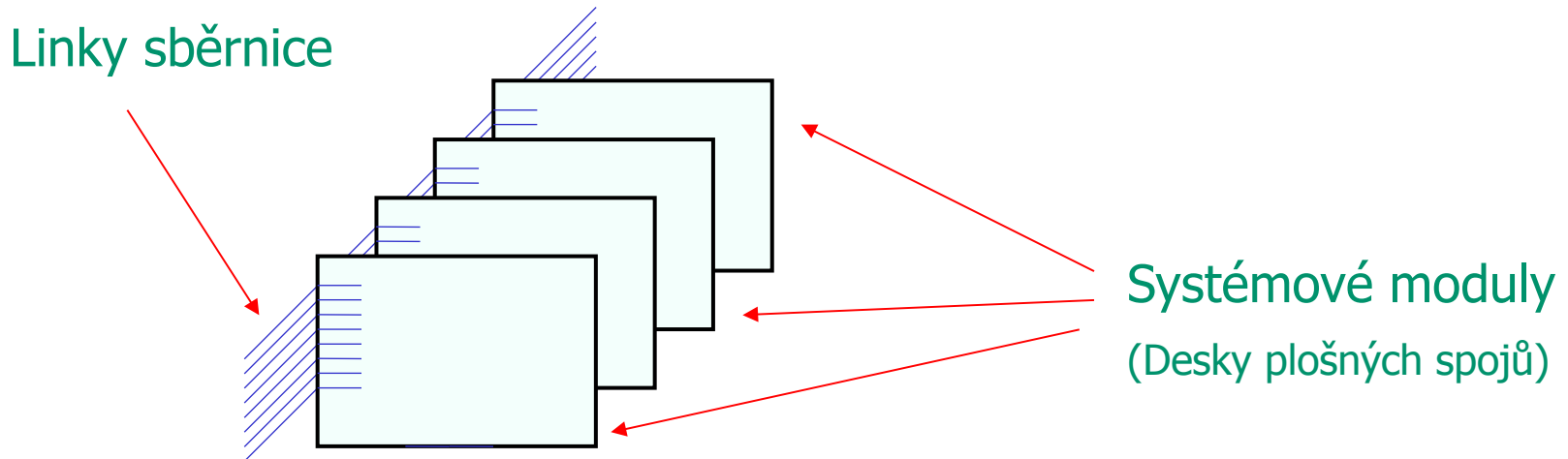
princip předávání zpráv  
nízký počet signál. linek  
distribuovaná arbitrace



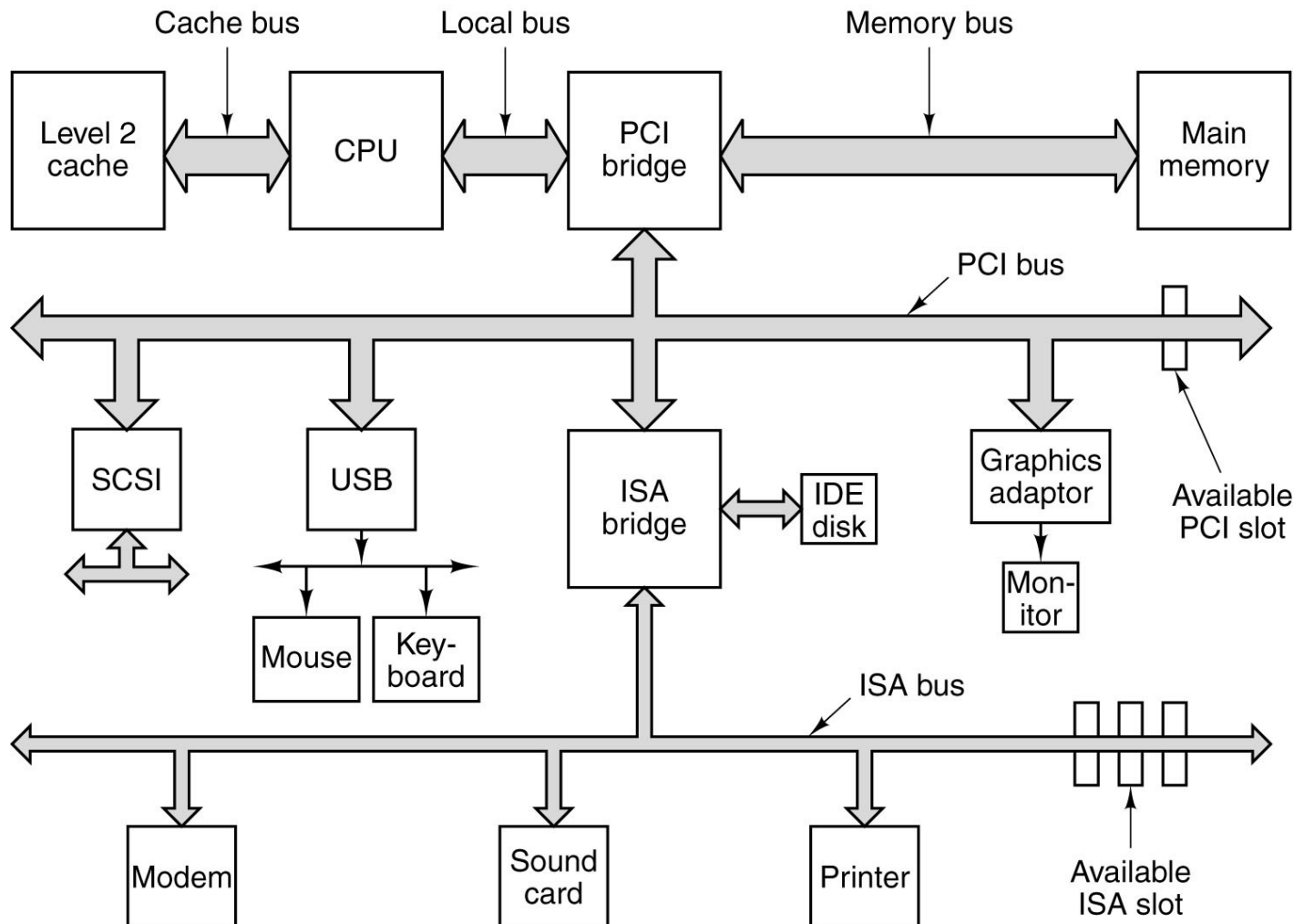
mapování do paměť. prostoru  
široké datové cesty  
centralizovaná arbitrace

# Sběrnice (backplane)

- „Pravá“ systémová sběrnice typu (backplane) obsahuje řadu konektorů
- Systémové moduly se zasouvají do konektorů
- Je definován log. protokol, aby se zabránilo vzniku elektr. konfliktů (zkratů)
  - Použití „třístavových“ budičů (jen jeden smí být v daném okamžiku aktivní)
  - Technologie „otevřený kolektor“ (pasivní jednička přes odpor)



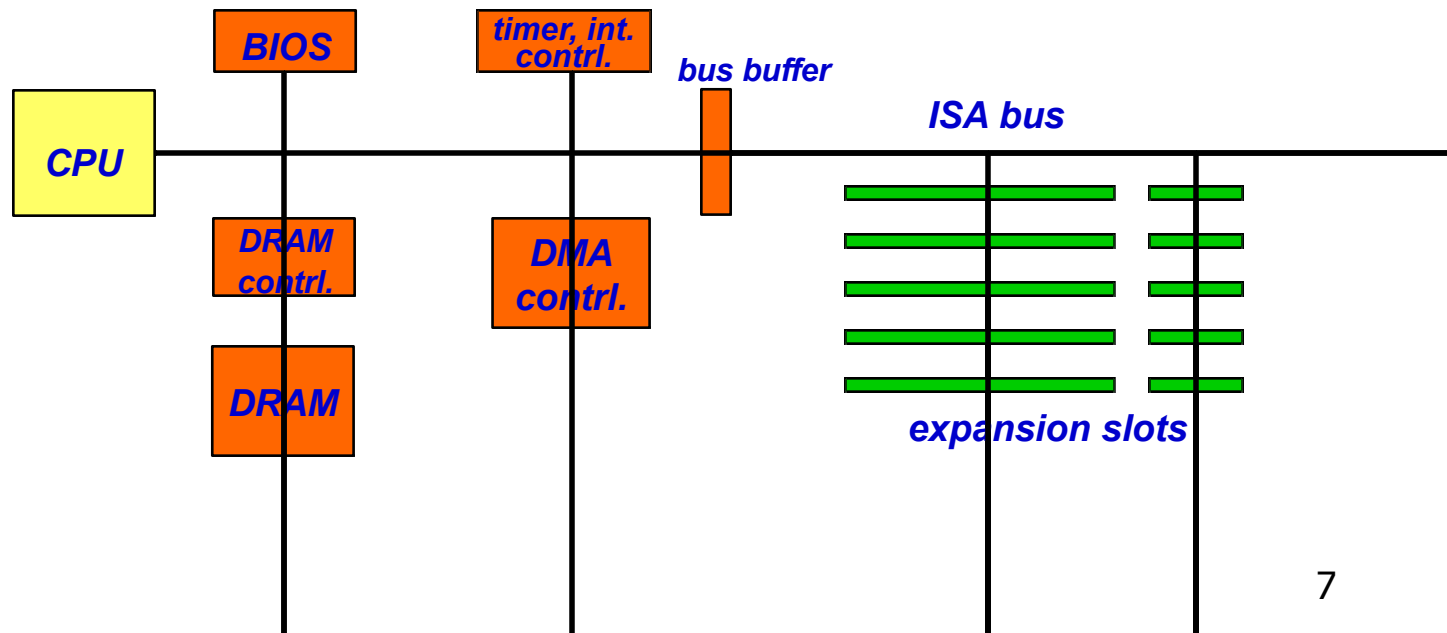
# Starší implementace PC



# Sběrnice v PC-XT a PC-AT

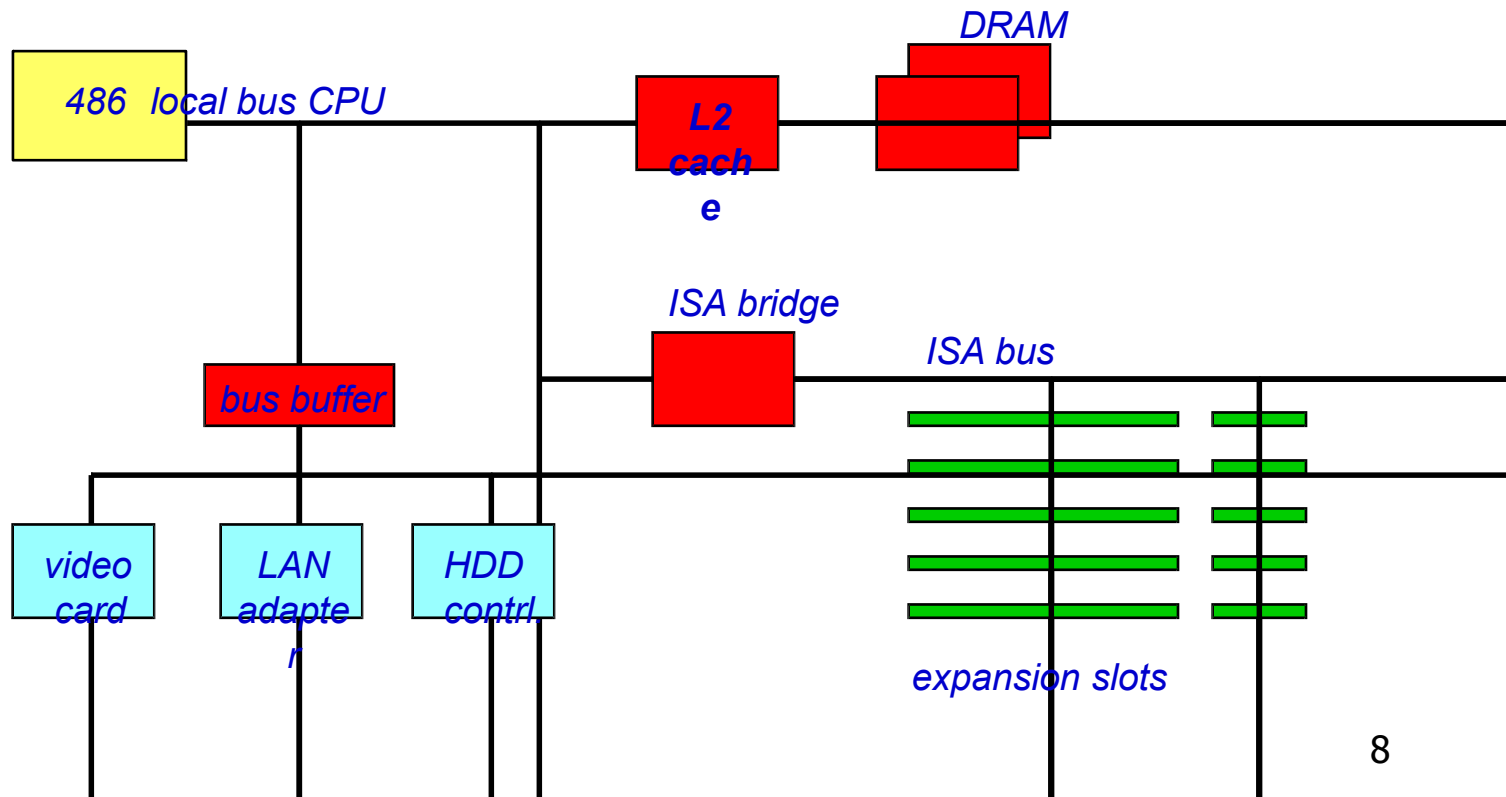
## □ ISA (Industry Standard Architecture)

- ❖ IBM-PC a PC-XT: 8 bitů na 4.77MHz, přímé propojení k 8088, 2-stupňový cykl sběrnice (šířka pásma sběrnice 2.38Mbyte/sec)
- ❖ AT bus: extension slot + 8 bitová ISA
  - 16 bitů na 8.33MHz pro 80286



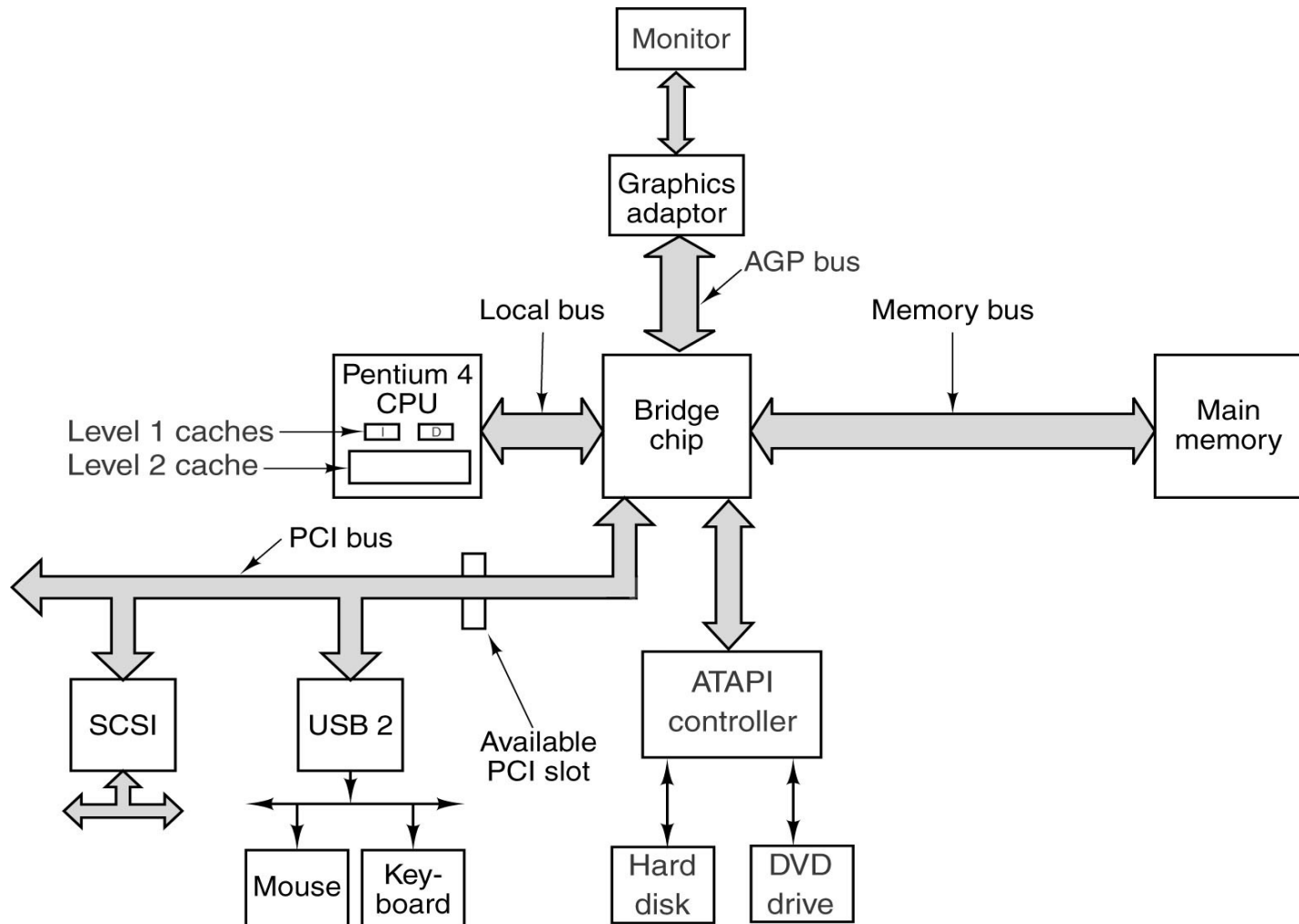
# Sběrnice v PC(486)

- ❑ 16-bitová ISA nemohla podporovat aplikace Windows --- video data
- ❑ VESA LB (local bus) -- vázána na 486 local bus, 33 MHz, 32 bitů



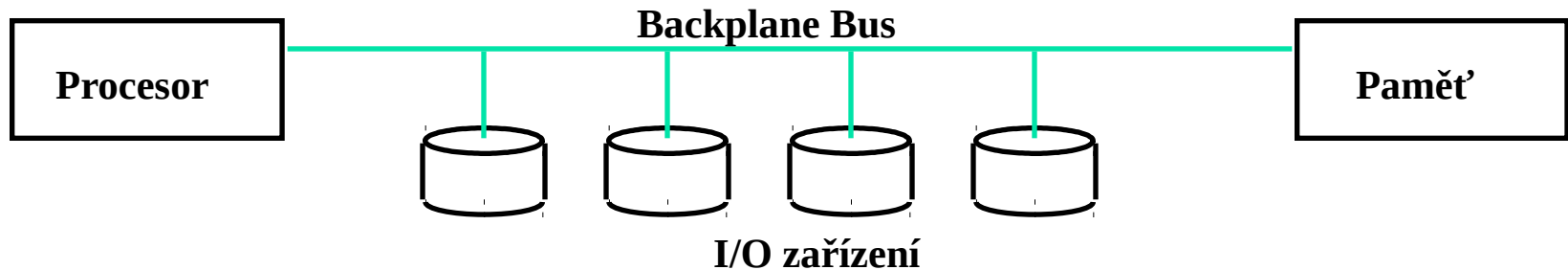


# Novější implementace PC



# Sběrnice

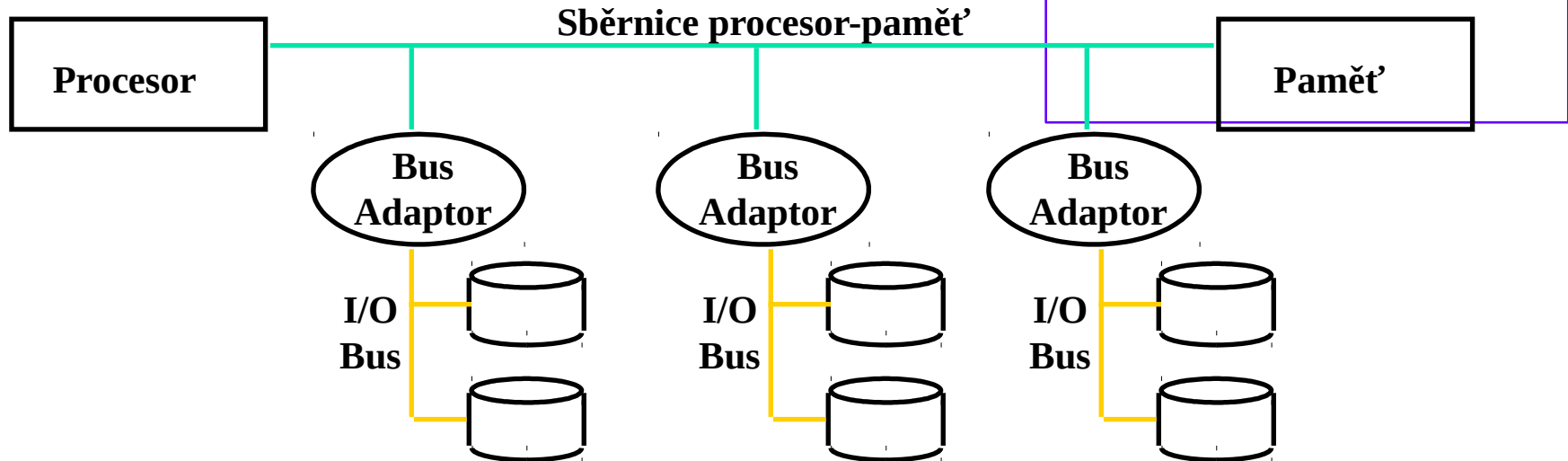
Počítačový systém s jednou sběrnicí:  
Backplane Bus



- ♦ Jednoduchá sběrnice se používá pro:
  - ♦ Komunikace procesoru s paměť
  - ♦ Komunikace mezi I/O zařízeními a paměť
- ♦ Výhody: Jednoduchost a nízká cena
- ♦ Nevýhody: pomalé a sběrnice se může stát úzkým místem systému
- ♦ Příklad: IBM PC - AT

# Sběrnice

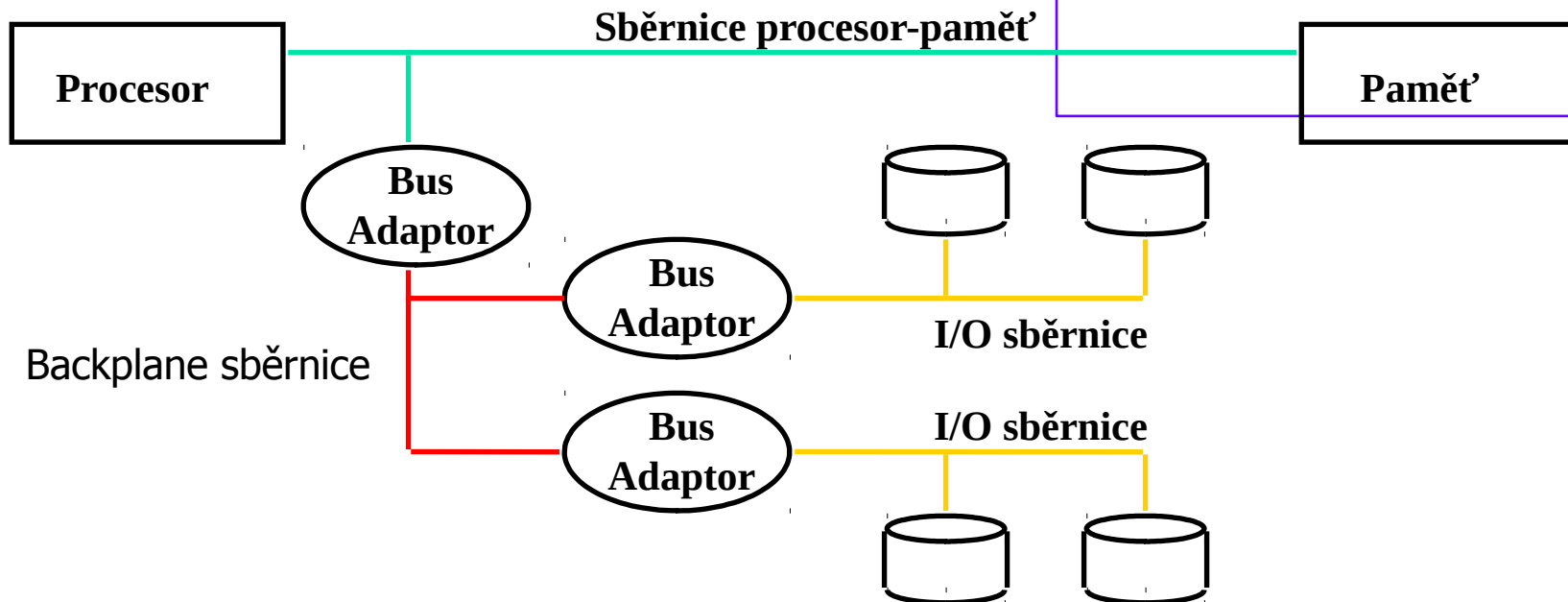
## Dvouběrnicevý systém



- ◆ I/O sběrnice je připojena k procesoru, popř. k paměťové sběrnici pomocí sběrnicevých adaptorů:
  - ◆ Paměťová sběrnice procesoru: určena hlavně pro komunikaci procesor-paměť
  - ◆ I/O sběrnice: obsahuje konektory pro I/O zařízení
- ◆ Apple Macintosh-II
  - ◆ NuBus: Procesor, paměť a několik vybraných I/O zařízení
  - ◆ SCSI sběrnice: zbylá I/O zařízení

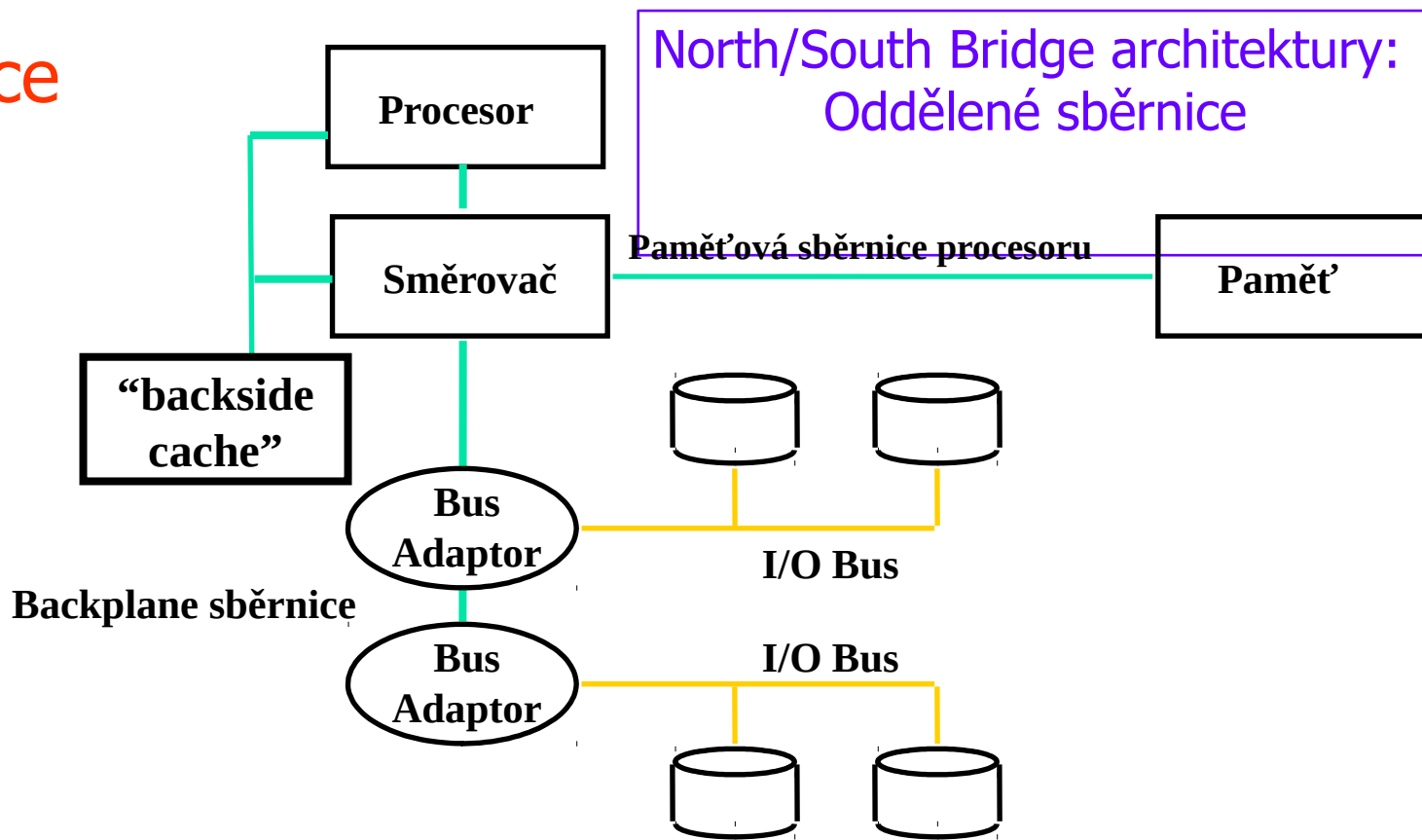
# Sběrnice

System se třemi sběrnici



- ♦ Malý počet „backplane“ sběrnic, napojených na sběrnici procesor-paměť
  - ♦ Pro přenos dat mezi procesorem a pamětí se používá výhradně paměťová sběrnice
  - ♦ I/O sběrnice se připojují k „backplane“ sběrnici
- ♦ Výhoda: minimalizuje se zátěž procesorové sběrnice

# Sběrnice



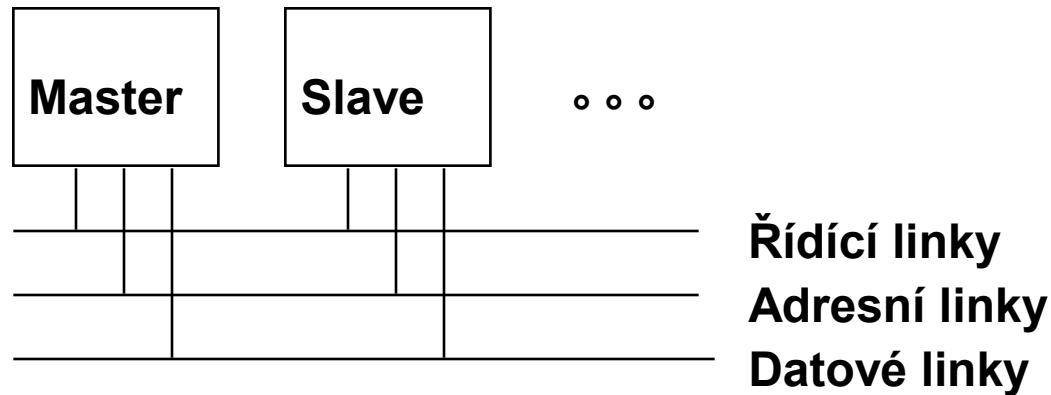
- ◆ Oddělené soubory pinů pro různé funkce
  - ◆ Paměťová sběrnice
  - ◆ Cache paměti
  - ◆ Sběrnice pro připojení grafické karty (rychlý frame buffer)
  - ◆ I/O sběrnice jsou připojeny k „backplane“ sběrnici
- ◆ Výhody:
  - ◆ Sběrnice lze provozovat při různých rychlostech
  - ◆ Mnohem menší celkové zatížení !



- ◆ Synchronní sběrnice:
  - ◆ Mezi řídicími linkami zahrnuje i hodiny
  - ◆ Fixní komunikační protokol, definovaný relativně k hodinám
  - ◆ Výhoda: malé množství logiky a vysoká rychlost
  - ◆ Nevýhody:
    - ◆ Každé zařízení pracující na sběrnici musí pracovat na stejné rychlosti hodin
    - ◆ Aby se potlačil vliv časového skew, sběrnice nesmí být příliš dlouhá, má-li být zachována rychlost
- ◆ Asynchronní sběrnice:
  - ◆ Synchronizaci dat neurčují hodiny
  - ◆ Může obsluhovat širokou škálu zařízení
  - ◆ Délka se může zvětšovat bez ohledu na skew hodinového signálu
  - ◆ Vyžaduje protokol zahrnující „handshaking“

# Sběrnice

Některé pojmy



**Bus Master:** má schopnost řídit sběrnici, zahajuje transakci

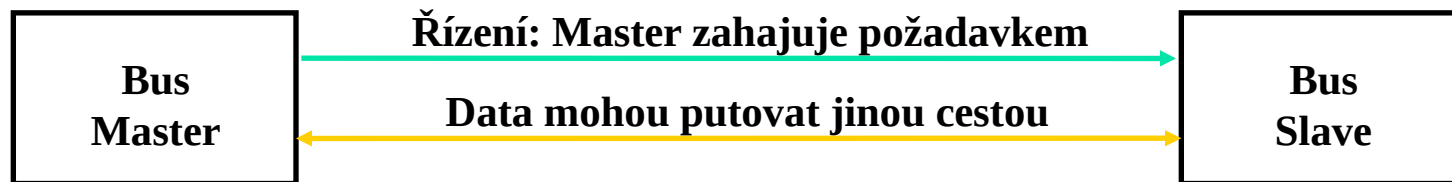
**Bus Slave:** modul aktivovaný zařízením typu master při transakci

**Komunikační protokol sběrnice:** specifikace posloupnosti událostí a časových požadavků při přenosu informace.

**Asynchronní přenosy:** řídicí linky (req, ack) zajišťují synchronizaci událostí.

**Synchronní přenosy:** posloupnost událostí odvozovaná od společných hodin.

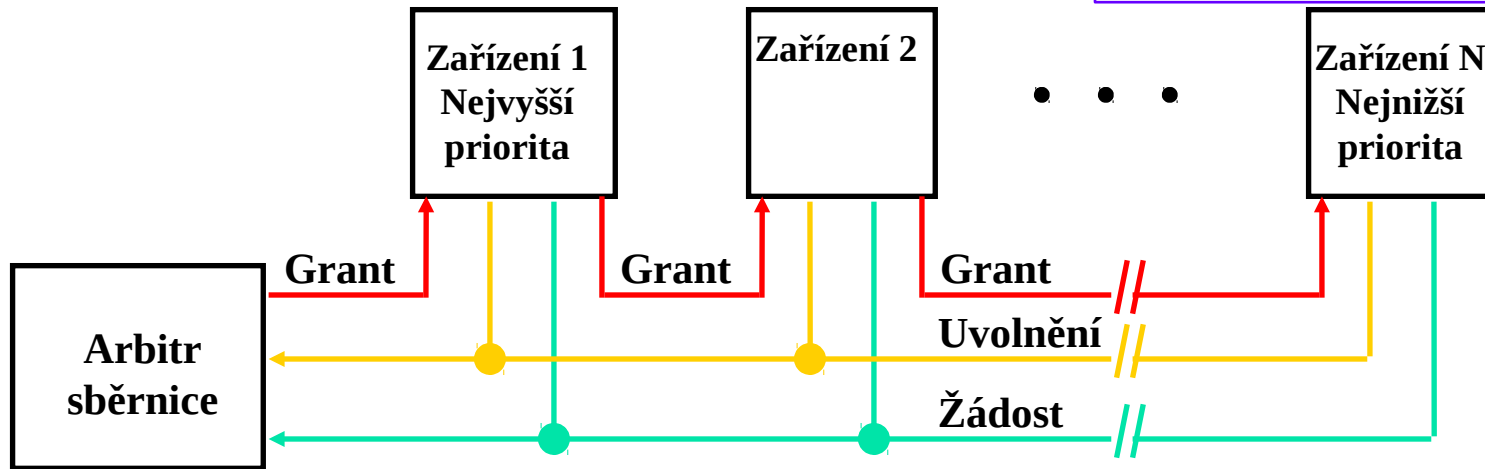




- ◆ Důležitá otázka při návrhu sběrnice:
  - ◆ Jak je sběrnice rezervována pro zařízení, které ji chce použít ?
- ◆ Chaosu je zabráněno využitím organizace master-slave:
  - ◆ Pouze bus master může řídit přístup na sběrnici:
    - Inicializuje a řídí všechny požadavky na sběrnici
  - ◆ Slave odpovídá na požadavky zápisu nebo čtení
- ◆ Nejjednodušší systém:
  - ◆ Procesor je jediným zařízením typu master na sběrnici
  - ◆ Procesor musí řídit všechny požadavky na sběrnici
  - ◆ Hlavní nevýhoda: procesor je zapojen do každé transakce

# Sběrnice

Arbitrační schéma  
„Daisy Chain“

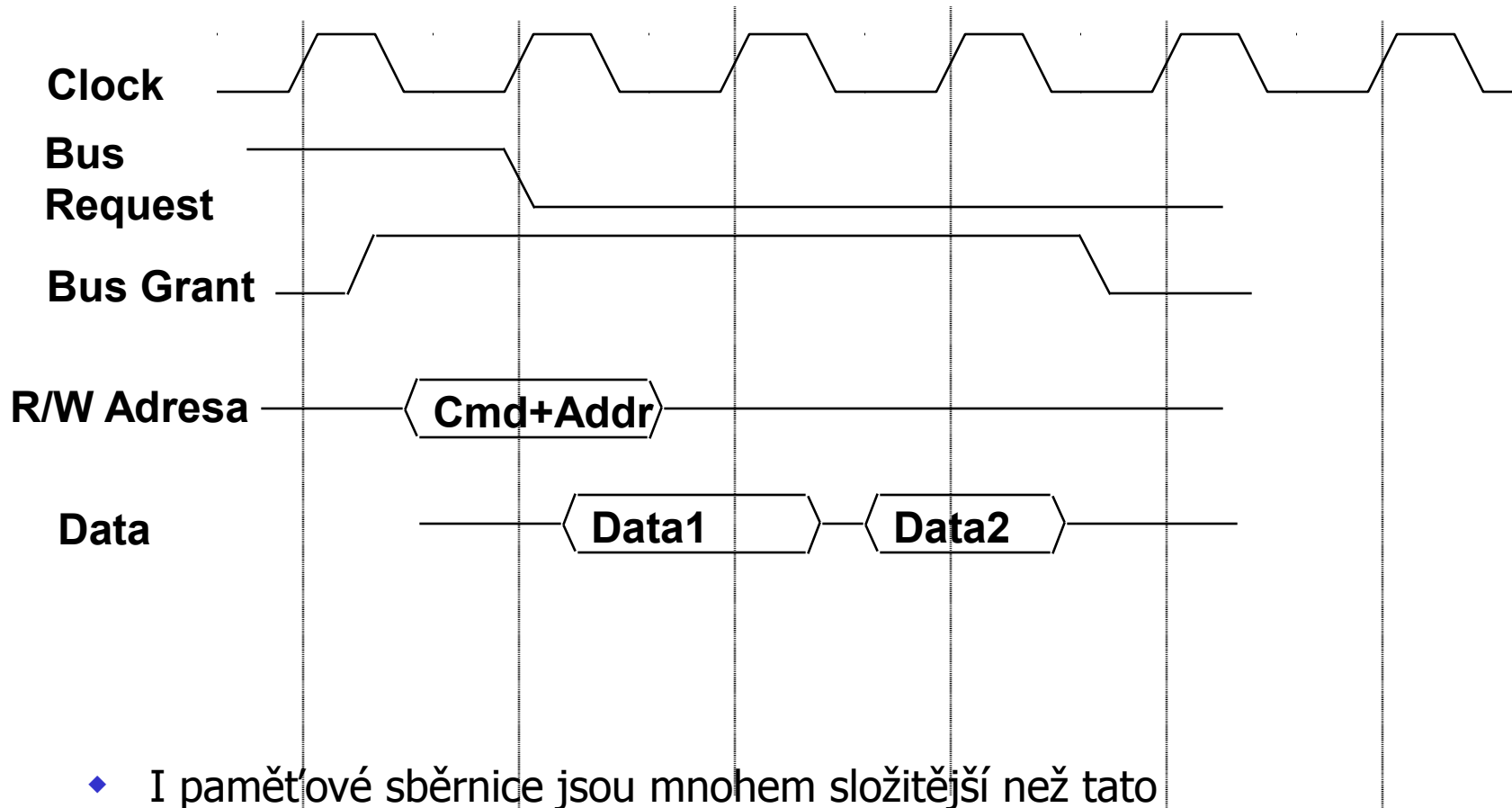


signály typu wired-OR

- ♦ **Výhoda:** jednoduchost
- ♦ **Nevýhody:**
  - ♦ Nemůže zajistit „spravedlivost“:  
Zařízení s nízkou prioritou může být mimo hru neomezeně
  - ♦ Použití mechanismu „daisy chain“ pro signál grant omezuje rychlost

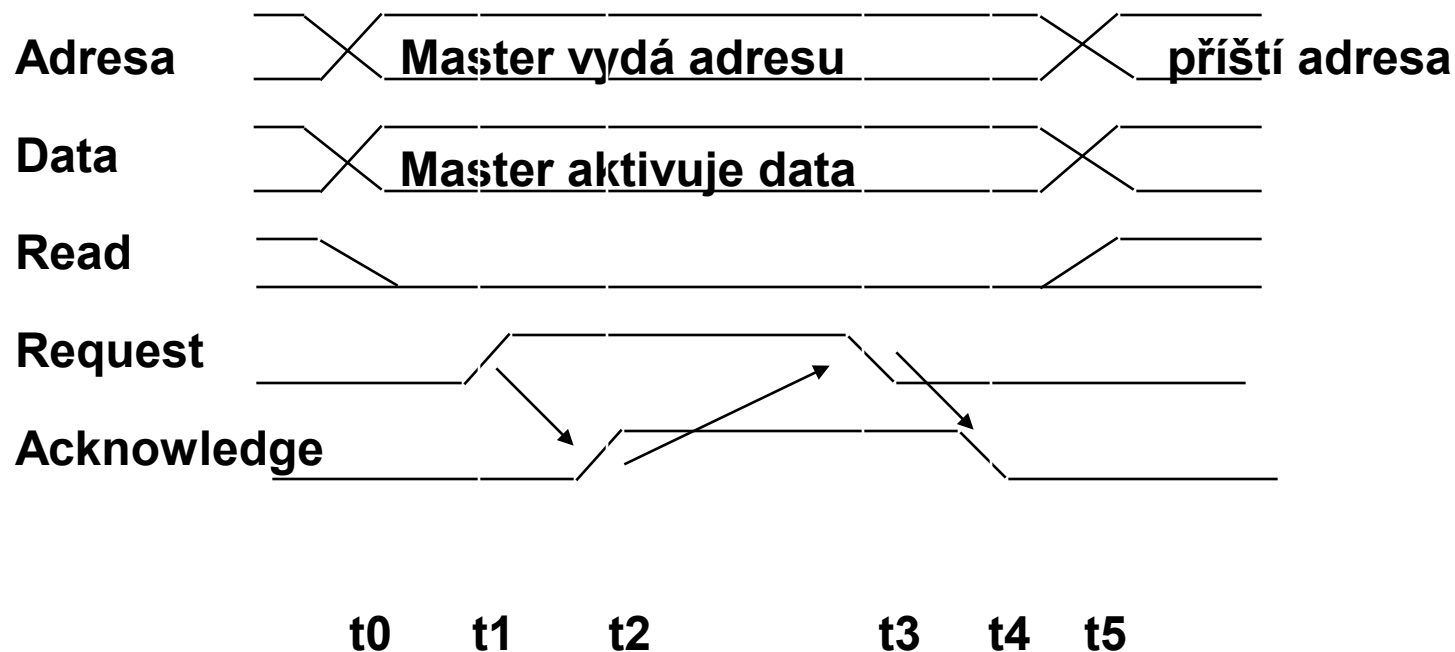
Pořadí je:

1. **Žádost**
2. **Grant**
3. **Uvolnění.**

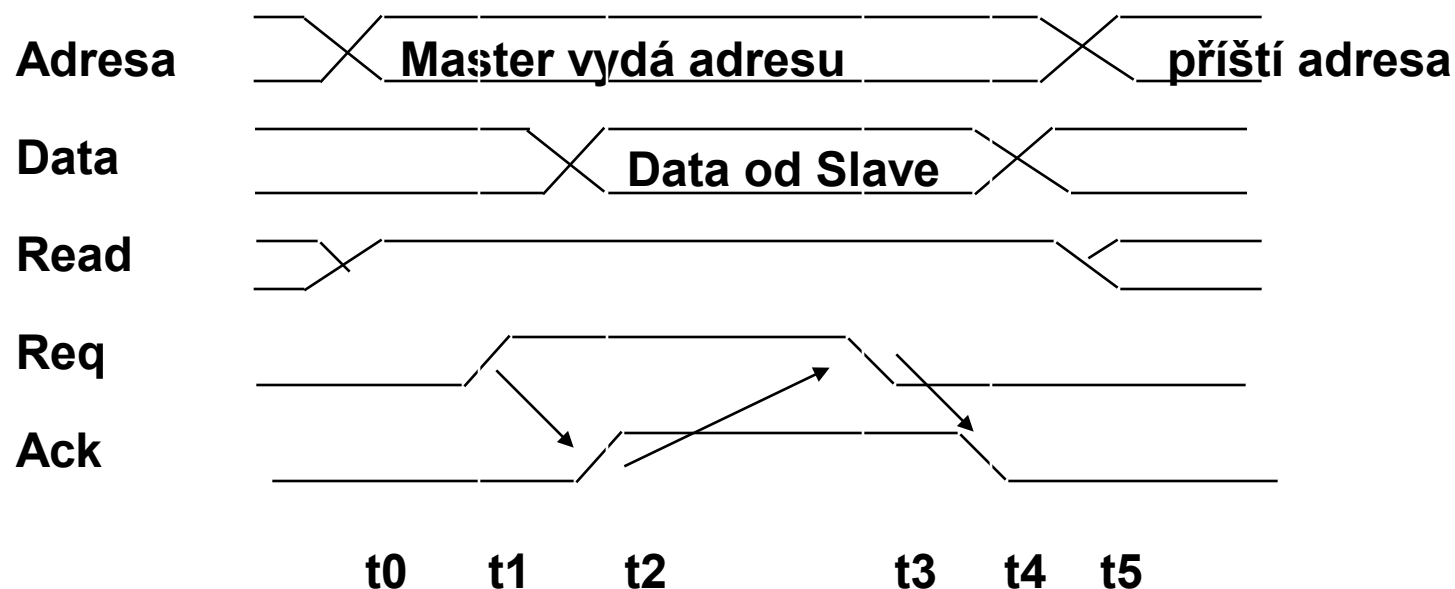


- ◆ I paměťové sběrnice jsou mnohem složitější než tato
  - ◆ paměť (slave) potřebuje dobu pro provedení čtecí operace
  - ◆ je třeba řídit rychlost přenosu dat

### Transakce - zápis



- t0 : Master získal řízení a určuje adresu, směr (not read) i data. Čeká specifikovanou dobu na slave, který musí adresu dekodovat
- t1: Master nastavuje linku **request**
- t2: Slave odpovídá signálem **ack**, čímž potvrzuje přijetí dat
- t3: Master uvolňuje signál **req**
- t4: Slave přestane budit signál **ack**



t0 : Master získal řízení a určuje adresu, směr (read)

Čeká specifikovanou dobu, aby mohl slave dekodovat adresu

t1: Master nastavuje linku **request**

t2: Slave provede čtení a odpovídá signálem **ack**. Naznačuje připravenost dodat data

t3: Master uvolňuje signál **req**, data přijata

t4: Slave přestane budit signál **ack**

Zařízení je nastaveno pro operaci, která se má provádět:

- Čtení nebo zápis
- Velikost přenášených dat
- Poloha zařízení (v systému)
- Adresa v paměti

Inicializace zařízení signálem, aby zahájilo operaci.

Po dokončení operace zařízení aktivuje interrupt.

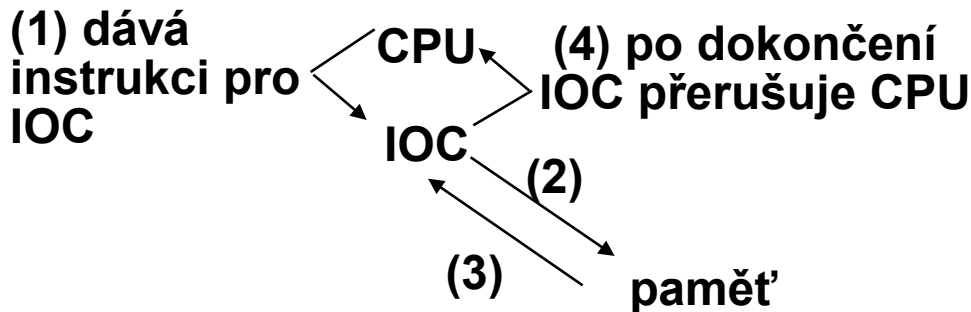
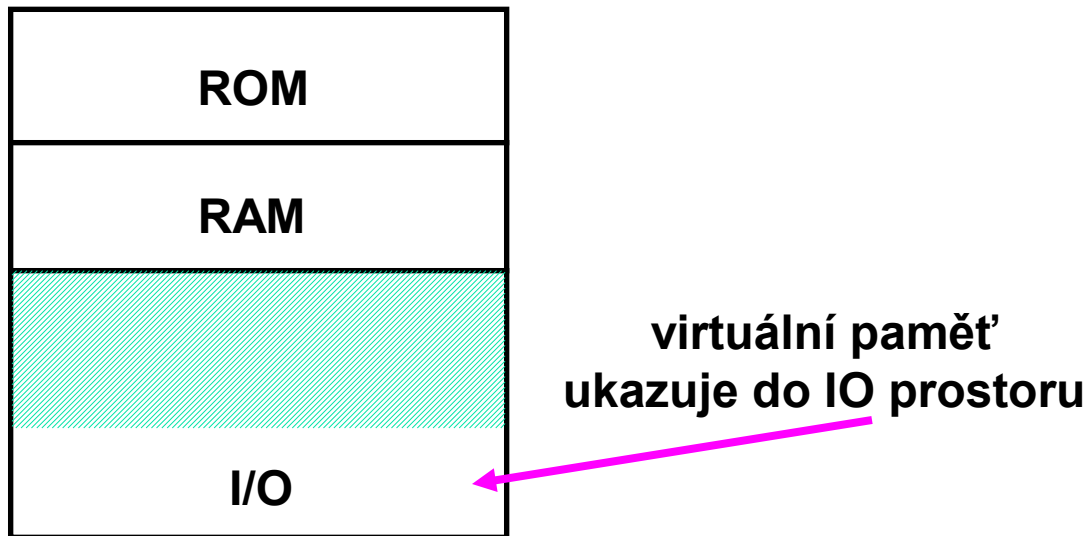
I/O instrukce (in,out) odlišné od instrukcí pro přístup do paměti.  
LDD R0,D,P <-- Load R0 obsahem portu P, nacházejícím se v zařízení D.

Registry zařízení jsou mapovány tak, že se chovají jako regulární paměť:  
LD R0,Mem1 <-- Load R0 obsahem portu P, nacházejícím se v zařízení D.

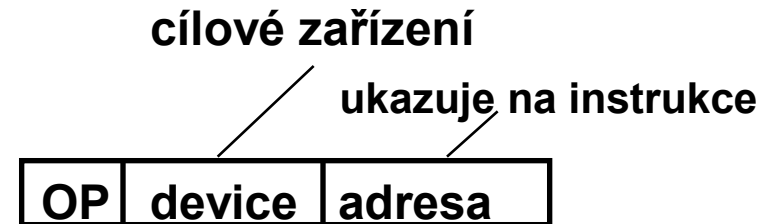
Toto je funkční, protože při inicializaci byly charakteristiky zařízení přiřazeny adrese Mem1.

# Připojení I/O k procesoru

Komunikace mezi CPU a IO



přenosy mezi pamětí a zařízením jsou řízeny přímo pomocí IOC.



IOP čte příkazy z paměti



co se má dělat

kam ukládat data

kolik

speciální funkce

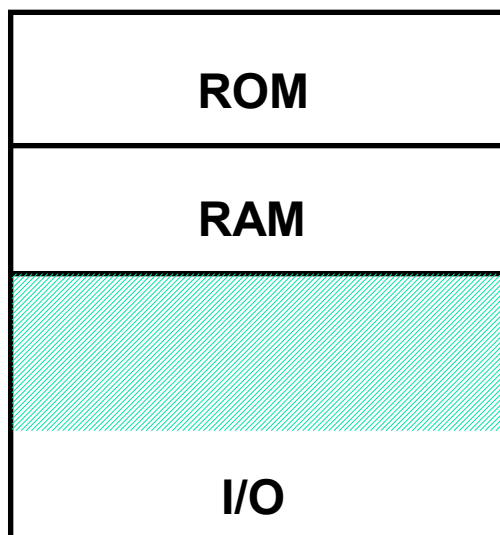
# Připojení I/O k procesoru

I/O mapovaný do  
paměťového prostoru

Některé fyzické adresy (oblast) jsou vyhrazeny.

Neodpovídá jim žádná **REÁLNÁ** paměť.

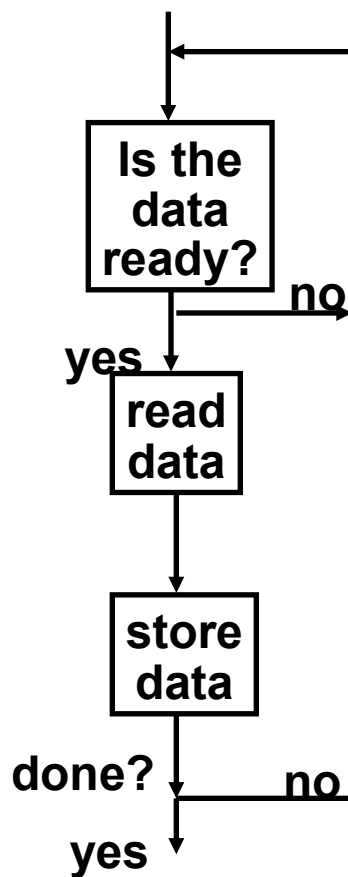
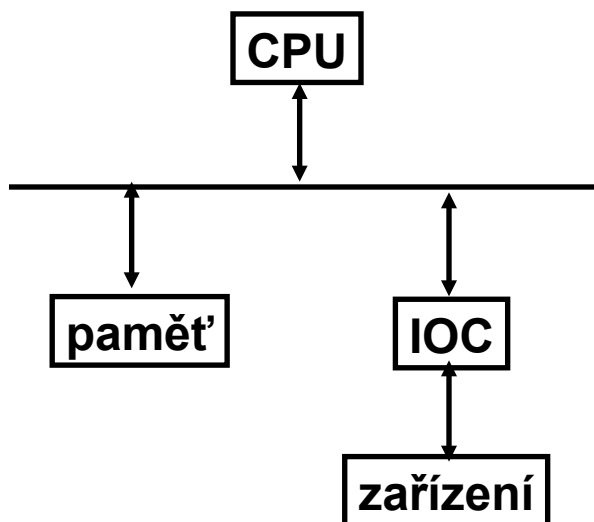
Procesor, který má do této oblasti přístup, „ví“, že uvedená oblast slouží k ukládání instrukcí pro IO procesor.





# Připojení I/O k procesoru

Přenosová metoda 1:  
Programový I/O (Polling)



aktivní čekací smyčka  
není právě efektivní  
využití CPU,  
pokud není zařízení  
velmi rychlé !

kontrola dokončení  
I/O může být ale  
programována  
specificky  
(„intenzivním kódem“)

# Připojení I/O k procesoru

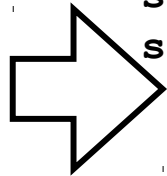
Interrupy od zařízení

- ◆ I/O interrupt se podobá výjimce (exception) s následujícími rozdíly:
  - ◆ I/O interrupt je asynchronní
  - ◆ Je třeba dodat další informaci
- ◆ I/O interrupt je asynchronní vzhledem k provádění instrukcí:
  - ◆ I/O interrupt není asociován s žádnou instrukcí
  - ◆ I/O interrupt nebrání žádné instrukci v dokončení
    - ◆ Lze vybrat vhodný okamžik pro zahájení obsluhy interruptu
- ◆ I/O interrupt je složitější než výjimka:
  - ◆ Je třeba zprostředkovat identitu zařízení, které interrupt generovalo
  - ◆ Požadavek na interrupt může mít různou urgenci:
    - ◆ Interruptům musí být přiřazeny priority

# Připojení I/O k procesoru

## Interrupty od zařízení

Externí interrupt



```
...  
add    $r1,$r2,$r3  
subi   $r4,$r1,#4  
slli   $r4,$r4,#2
```

Hiccup (!)

```
lw     $r2,0($r4)  
lw     $r3,4($r4)  
add    $r2,$r2,$r3  
sw     8($r4),$r2  
...
```

PC uložen  
Zákaz všech Int  
Supervisor Mode

Obnova PC  
User Mode

Zvýšení priority  
Povolení všech Intů  
Uložení registrů

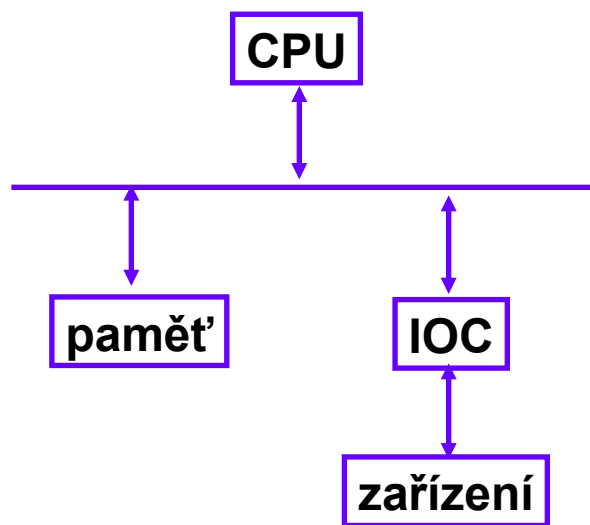
```
...  
lw     $r1,20($r0)  
lw     $r2,0($r1)  
addi   $r3,$r0,#5  
sw     $r3,0($r1)  
...
```

Obnova registrů  
Nulování akt. Intu  
Zákaz všech Intů  
Obnova priority  
RTI

“Obsluha interruptu”

- ♦ Výhoda:
  - ♦ Uživatelský program je pozastaven pouze v době aktuálního přenosu
- ♦ Nevýhoda, je třeba speciální hardware pro:
  - ♦ Vytvoření interruptu (I/O zařízení)
  - ♦ Detekci interruptu (procesor)
  - ♦ Uložení kontextu pro pokračování programu po obsluze interruptu (procesor)

# Připojení I/O k procesoru



**Uživatelský program zastaven jen během vlastního přenosu. Přenos zajišťuje obsluha interruptu.**

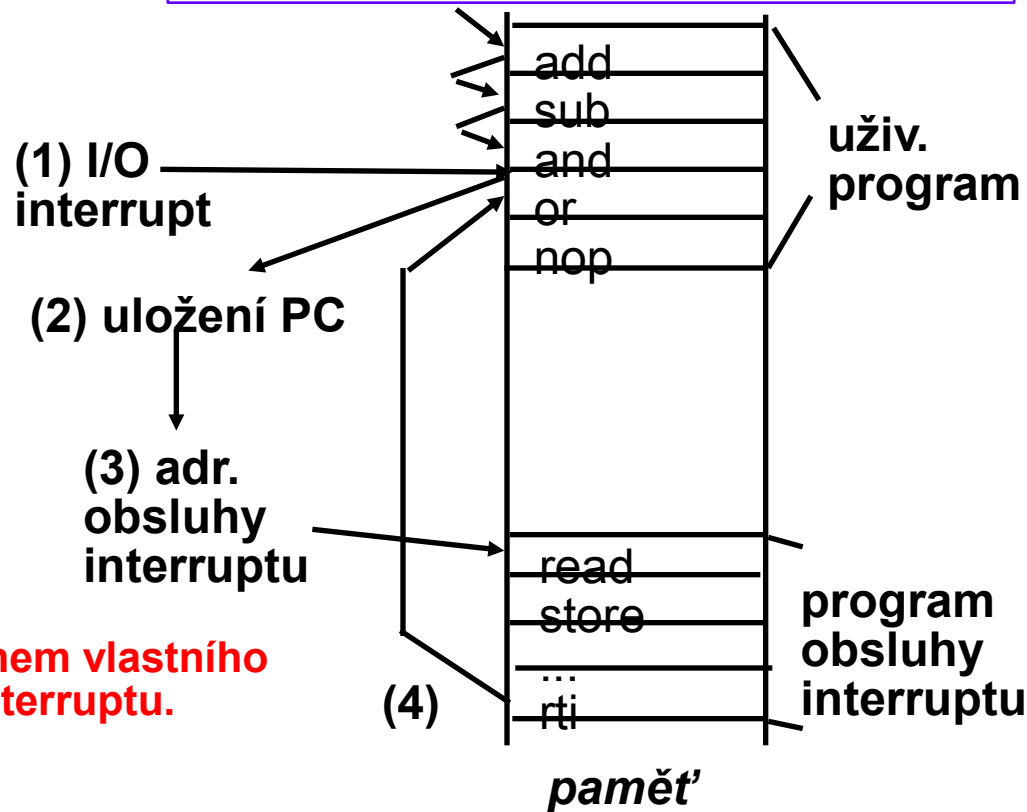
1000 přenosů po 1000 bytech:  
1000 interruptů @ 2  $\mu$ sekundy na interrupt  
1000 obsluha interruptu @ 98  $\mu$ sekund  
každý = 0.1 CPU sekund

Přenosová rychlost zařízení = 10 MByte/sec  $\Rightarrow$   $0.1 \times 10^{-6}$  sec/byte  $\Rightarrow$  0.1  $\mu$ sec/byte  
 $\Rightarrow$  1000 bytů = 100  $\mu$ sekund

1000 přenosů x 100  $\mu$ sekund = 100 ms = 0.1 CPU sekund

**Stále velmi daleko od přenosové rychlosti zařízení. Celá 1/2 času se ztrácí v interuptech.**

Přenosová metoda 2:  
Přenos dat s podporou interruptů

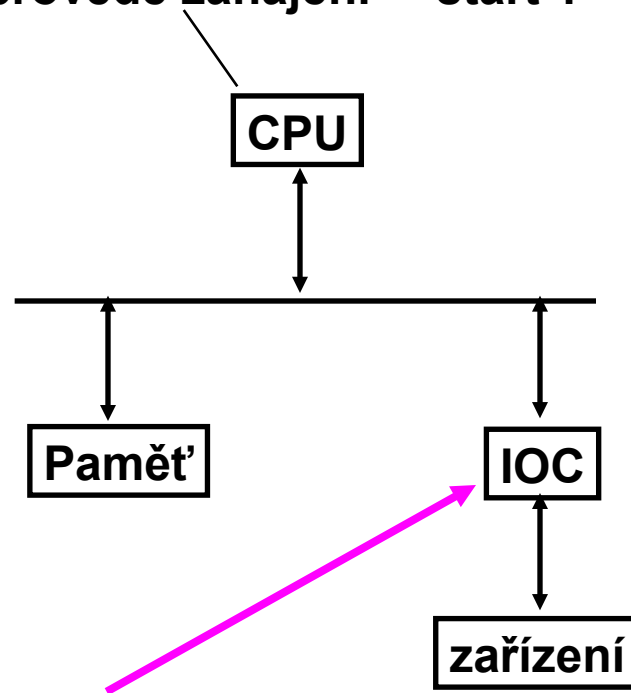


# Připojení I/O k procesoru

## Delegování I/O řídicí funkce z CPU: DMA

**CPU vysílá startovací adresu, směr a délku bloku k IOC. Potom provede zahájení - "start".**

- ♦ Přímý přístup do paměti (DMA):
  - ♦ Odehrává se vně CPU
  - ♦ Pracuje jako master na sběrnici
  - ♦ Přenáší bloky dat z nebo do paměti bez přímé účasti CPU



**IOC zajišťuje signály „handshake“ pro řadič periferie, adresy do paměti a signály „handshake“ pro paměť.**

# Připojení I/O k procesoru

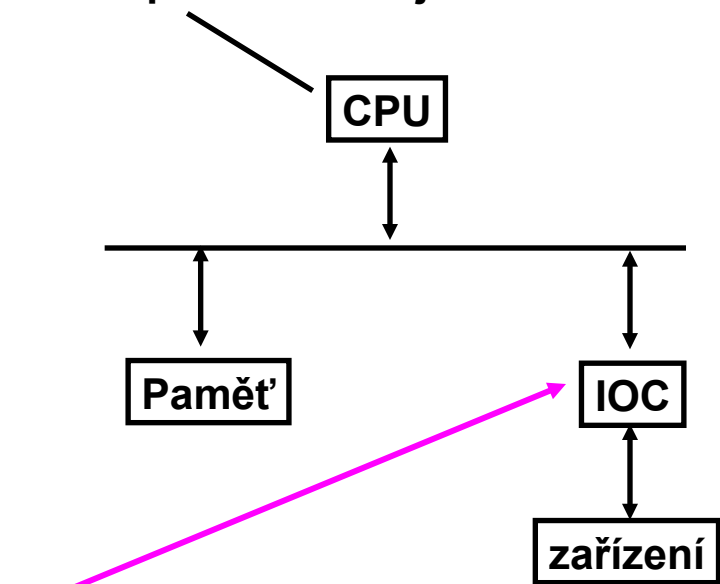
Přenosová metoda 3:  
Přímý přístup do paměti

Čas potřebný k 1000 přenosům po 1000 bytech:

- 1 inicializační sekvence DMA @ 50  $\mu$ sekund
- 1 interrupt @ 2  $\mu$ sekundy
- 1 obsluha interruptu @ 48  $\mu$ sekund

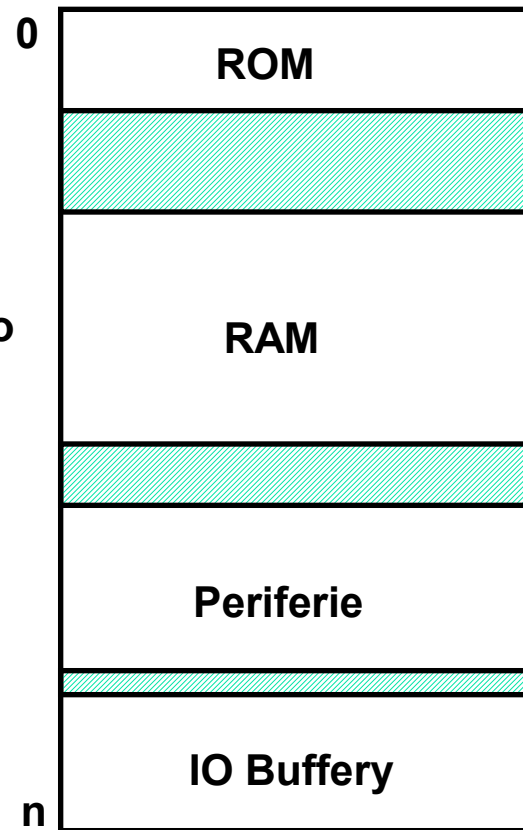
.0001 sekundy času CPU

CPU vysílá startovací adresu, směr a délku bloku do DMAC. Potom provede zahájení - "start".



IOC zajišťuje signály „handshake“ pro řadič periferie, adresy do paměti a signály „handshake“ pro paměť.

I/O mapovaný do paměťového prostoru



# Sběrnice PCI - úvod

- ◆ Architektura PCI Local Bus
- ◆ Signály PCI
- ◆ Základní operace sběrnice PCI
- ◆ Adresování PCI a příkazy sběrnice
- ◆ Konfigurace PCI
- ◆ Specifikace elektrických a časových parametrů
- ◆ Rozšíření na 64-bitů
- ◆ Přehled 66-MHz verze
- ◆ Variace PCI

# Sběrnice PCI

Kompletní dokumentace odpovídající roku 2003 je zachycena v následujících dokumentech:

- PCI Local Bus Specification, Revision 2.2
- PCI-to-PCI Bridge Architecture Specification, Revision 1.1
- PCI Bus Power Management Interface Specification, Revision 1.1
- PCI Hot Plug Specification, Revision 1.1
- Small PCI Specification, Revision 1.5a
- PCI BIOS Specification, Revision 2.1
- PCI Mobile Design Guide, Revision 1.1
- Mini PCI Specification, Revision 1.0
- PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0a
- PCI Standard Hot-Plug Controller and Subsystem Specification, Revision 1.0



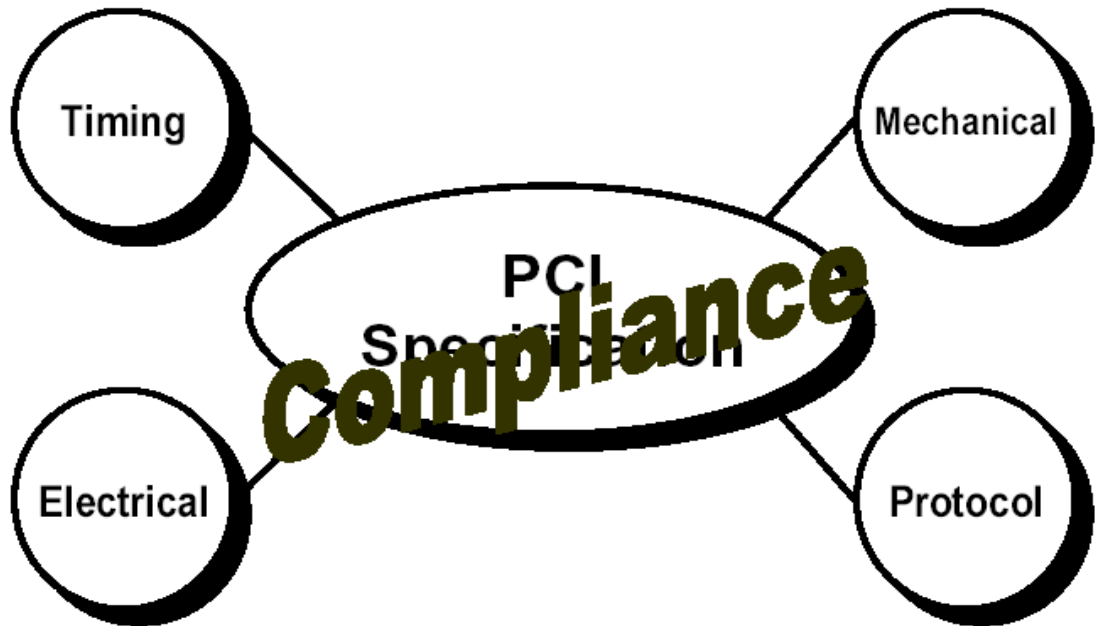
# PCI Local Bus

**Peripheral Component Interconnect (PCI).**

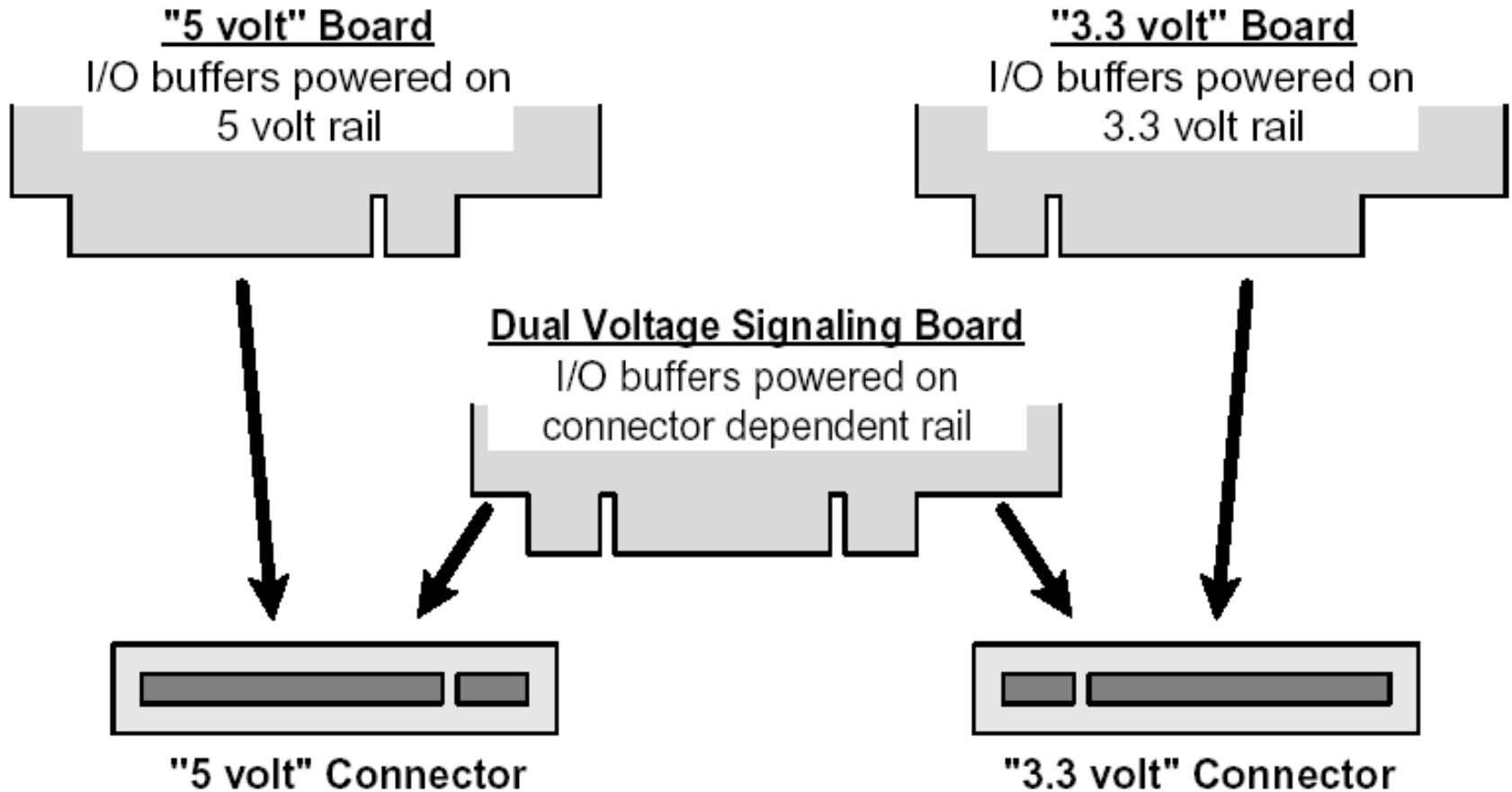
# PCI Local Bus

Definice sběrnice musí pokrývat celou řadu požadavků:

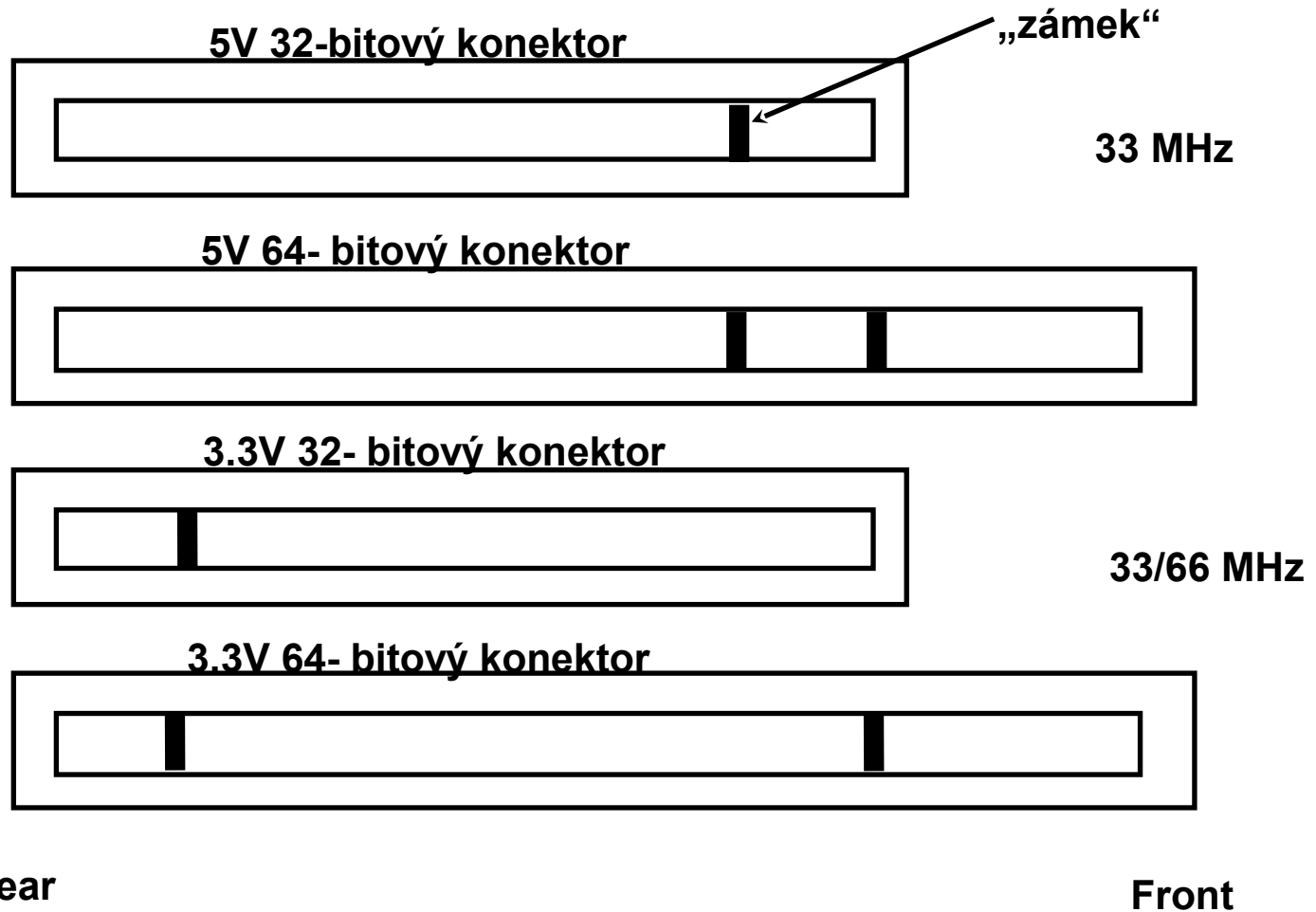
- Časování
- Mechanické parametry
- Elektrické parametry
- Protokol sběrnice



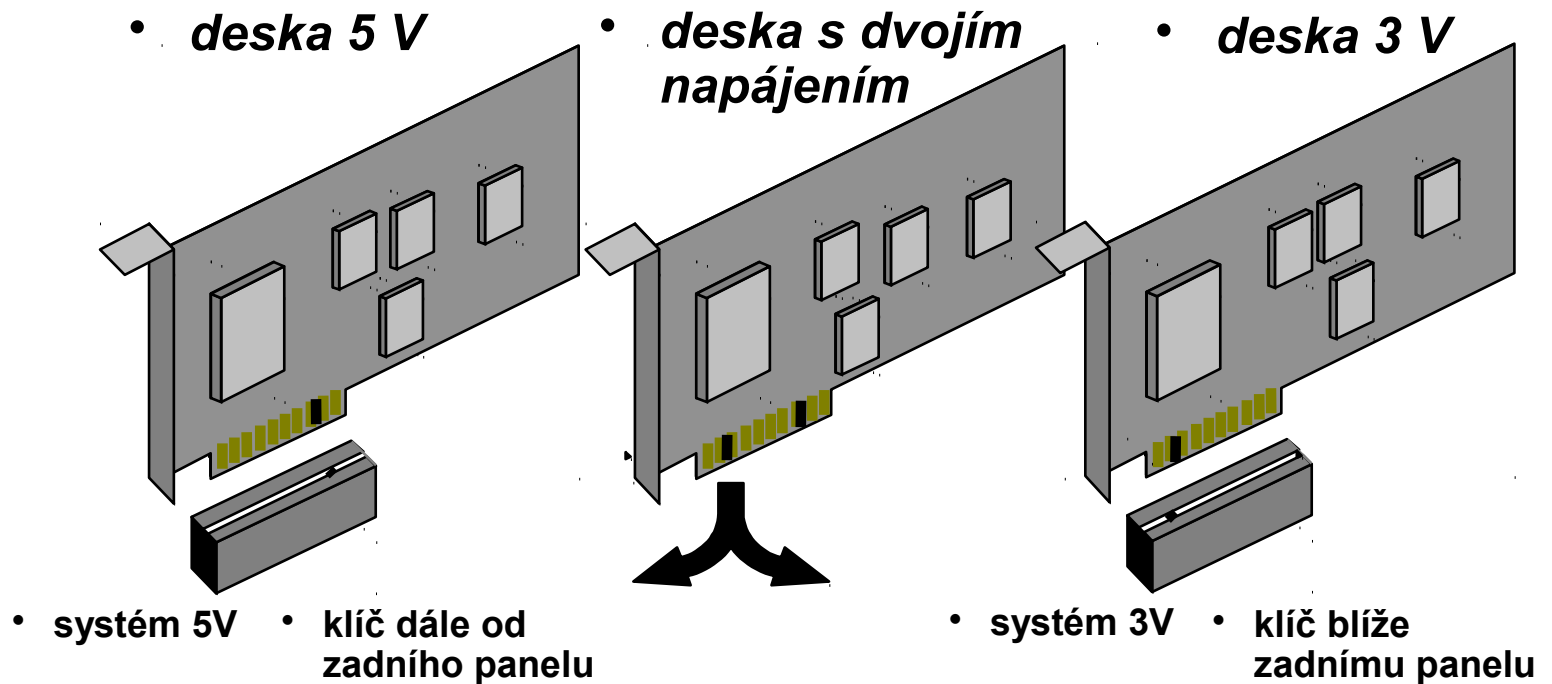
# Ošetření dvojího napájecího napětí



# Konektory desek PCI



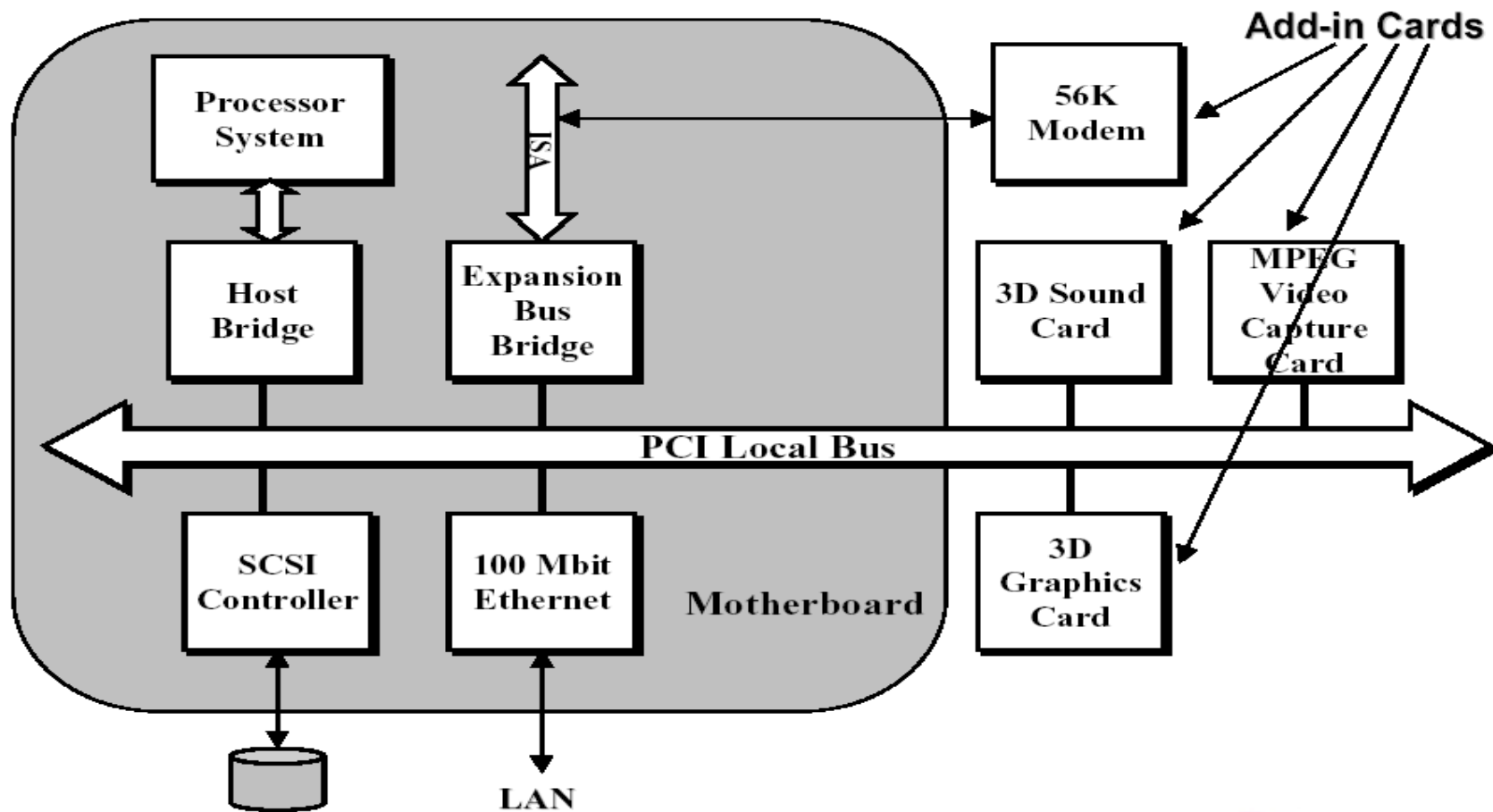
# Migrace z 5 V na 3.3 V



# Koncepce PCI

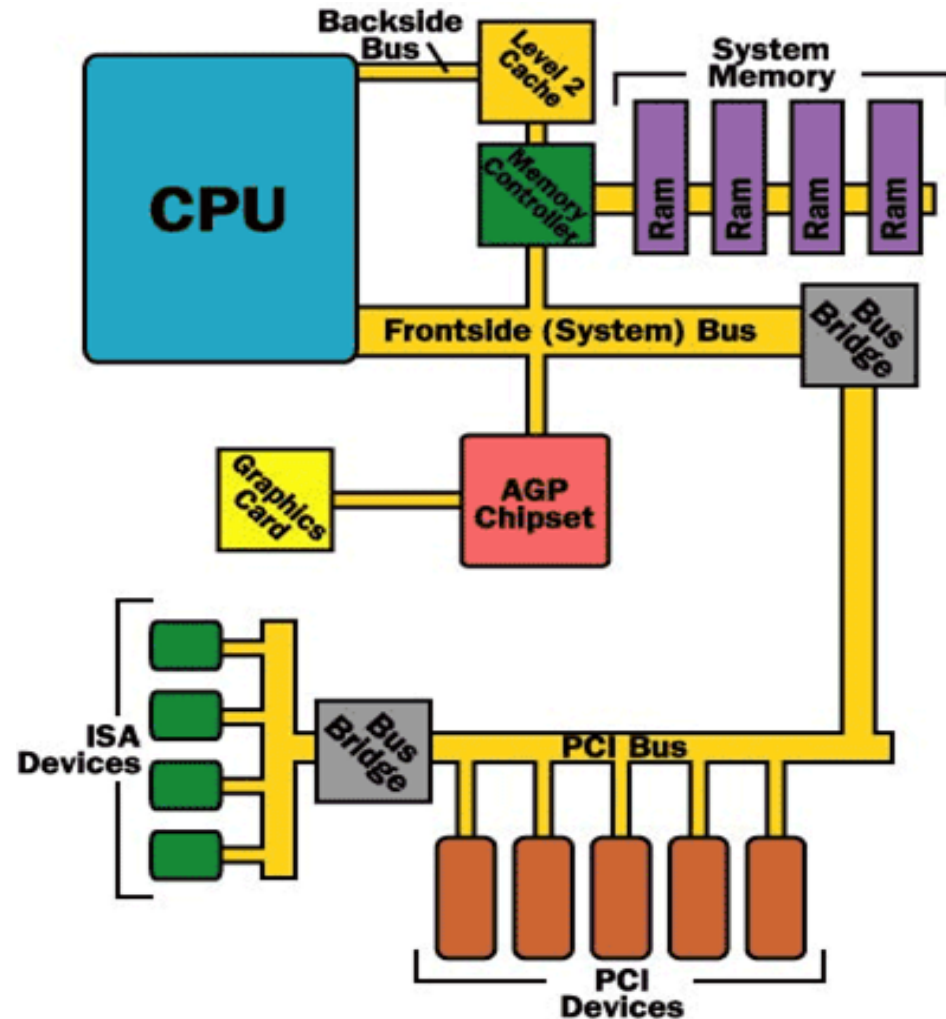
- ◆ Velká šířka pásma
  - ◆ Grafické karty
  - ◆ Pevné disky
- ◆ Rychlé CPU
  - ◆ Ale i obsluha pomalejších IO
- ◆ Sběrnice s vazbou na CPU a paměťovou sběrnici

# Základní architektura sběrnice



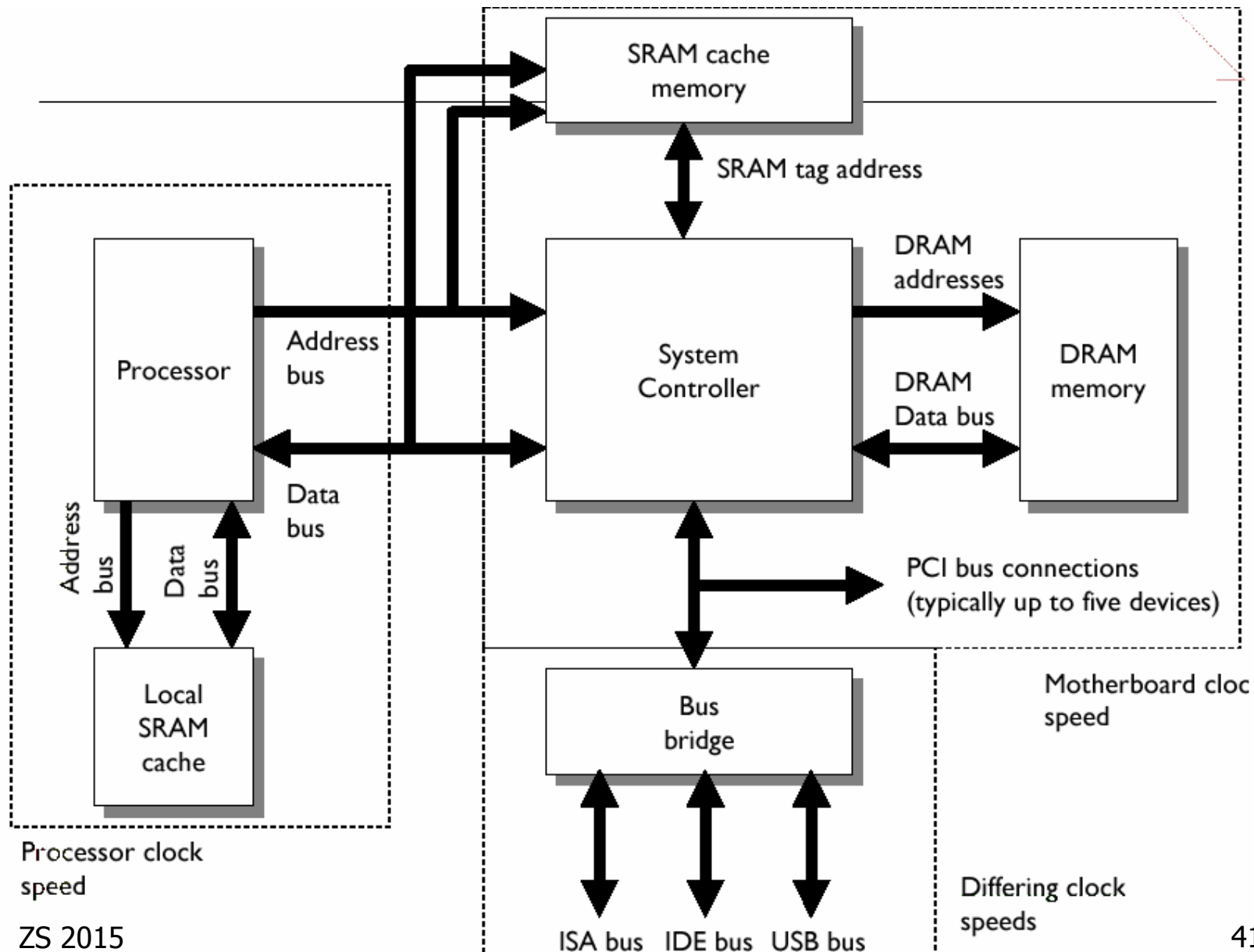
# Duální nezávislá sběrnice

- ◆ Backside Bus
- ◆ Frontside Bus
- ◆ PCI
  - ◆ Přímý přístup k systémové paměti pro připojená zařízení
  - ◆ Používá „bridge“ k připojení „frontside“ sběrnice a tím k CPU

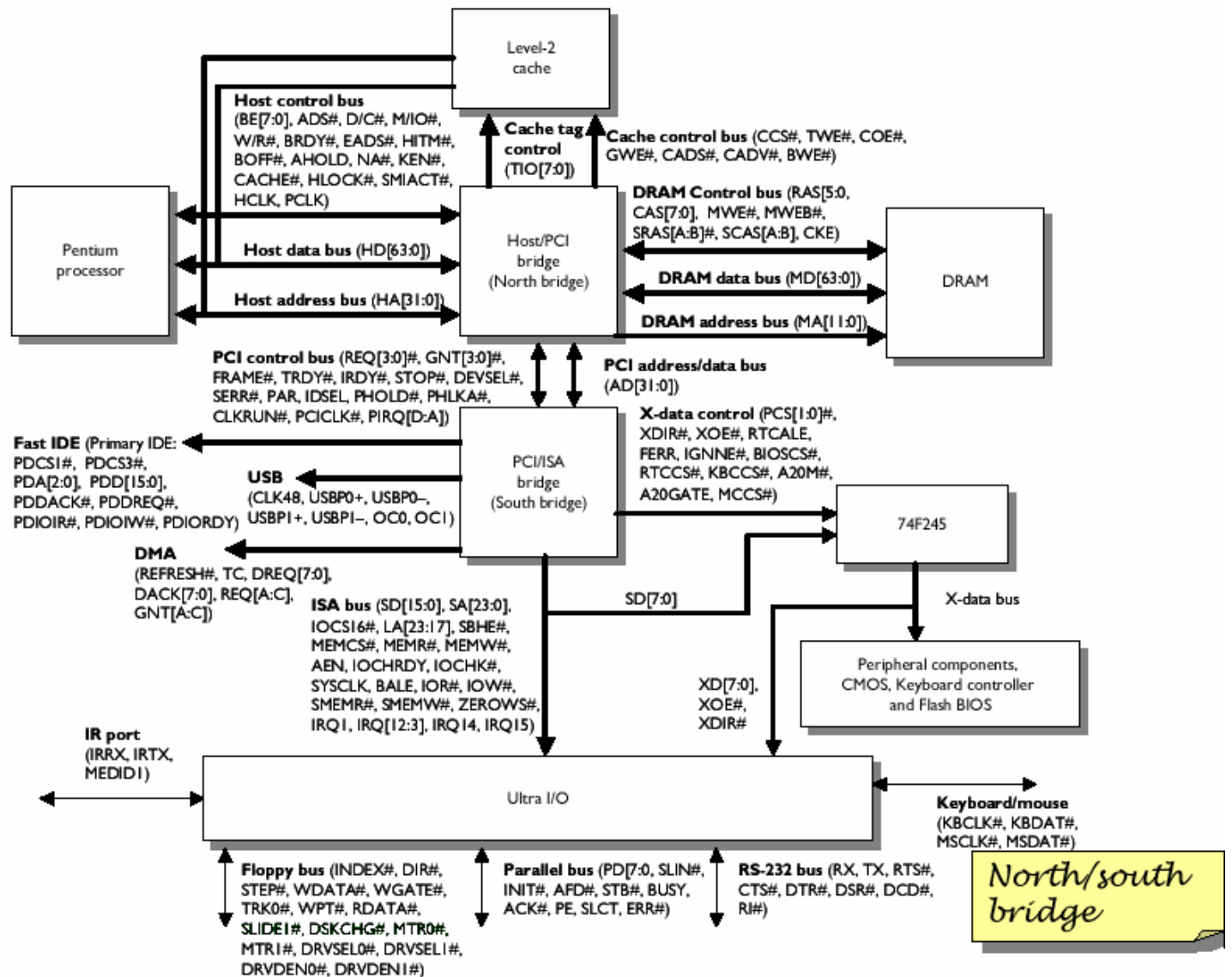




# Různé úrovně sběrnic



# North / South bridge



# Porovnání sběrnic

Typ	Šířka	Frekvence	MB/sec	Rok	
ISA	16 bitů	8 MHz	16 MBps	1984	IBM
EISA	32 bitů	8 MHz	32 MBps	1988	Gang of Nine
VL-bus	32 bitů	25 MHz	100 MBps	1992	VESA Commite (NEC)
VL-bus	32 bitů	33 MHz	132 MBps	1994	VESA Commite (NEC)
PCI	32 bitů	33 MHz	132 MBps	1993	PCI-SIG
PCI	64 bitů	33 MHz	264 MBps	1995	PCI-SIG
PCI	64 bitů	66 MHz	512 MBps	1998	PCI-SIG
PCI	64 bitů	133 MHz	1 GBps	2002	PCI-SIG
PCI express x1 (až x16)	32 bitů	66MHz	250 Mbps (až 4 GBps)	2003	PCI-SIG
...					
PCI express3 x1 až x16	32 bitů		0.5 GBps Až 16 GBps	2010	PCI-SIG
PCI express4	32bitů		Až 32 GBps	2017 ?	

# Typická karta PCI



Karta PCI – 47 pinů

# PCI revize 2.1 - vlastnosti

- ◆ PCI je sběrnice typu Multimaster Bus
- ◆ Všechny transakce *inicializuje* Master
- ◆ Všechny transakce směřují k(od) cílovému(ho) zařízení (*target*)

# PCI revize 2.1 – vlastnosti (pokračování)

- ◆ Nezávislost na procesoru
- ◆ Nízká spotřeba
- ◆ Použití burst módu pro všechny operace čtení a zápisu
- ◆ Rychlost sběrnice až 66 MHz
- ◆ Šířka sběrnice 64-bitů
- ◆ Nízký počet pinů (PCI Target: 47, PCI Initiator: 49 pinů)
- ◆ Paralelně probíhající přenosy na sběrnici (concurrency)
- ◆ Podpora funkce Bus master
- ◆ Arbitrace sběrnice na pozadí
- ◆ Autokonfigurace

# PCI revize 2.1 – vlastnosti (pokračování)

- ◆ Doba přístupu je 60 ns při 33 MHz, když iniciátor parkuje na sběrnici
- ◆ Paralelní činnost částí systému (sběrnice procesoru, PCI, exp.bus)
- ◆ Plná podpora PCI masterů přes bridge
- ◆ Paritní kontrola adresy, dat a povelů
- ◆ Tři adresní prostory
- ◆ Transparentnost z hlediska software
- ◆ Specifikace konektorů i rozměrů přídatných desek

# Úvod do operací sběrnice PCI

## Klíčové pojmy

- ◆ **Initiator**
  - ◆ Jinak **master**
  - ◆ Vlastní sběrnici a zahajuje přenos dat
  - ◆ Každý **initiator** musí být zároveň **target**
- ◆ **Target**
  - ◆ Jinak **Slave**
  - ◆ Cíl operace přenosu data (čtení nebo zápis)
- ◆ **Agent**
  - ◆ Libovolný **initiator/target** nebo **target** na PCI sběrnici
- ◆ **Funkce**
  - ◆ Trochu posunutý význam, znamená spíše port, popř. kanál



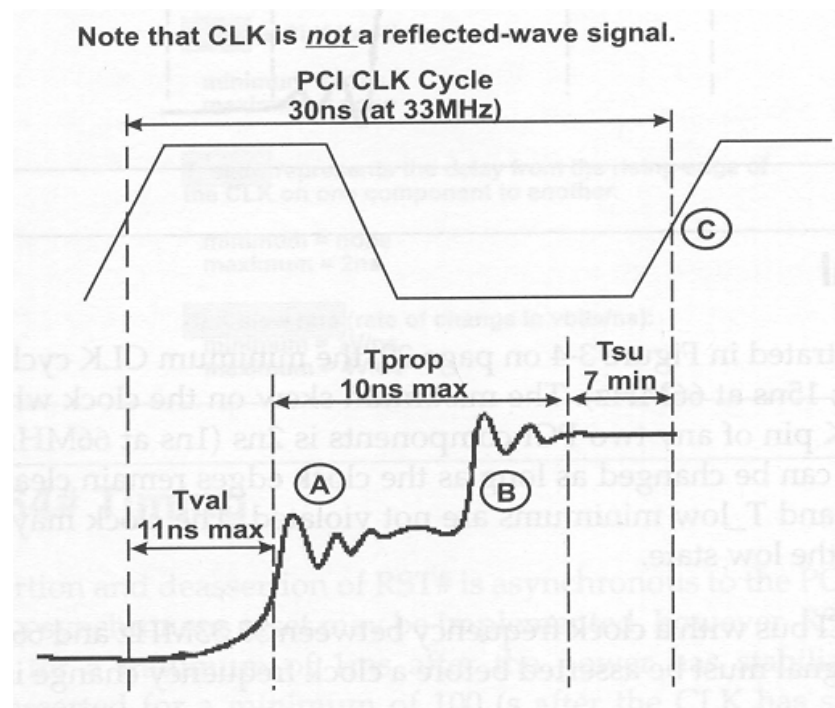
# Úvod do operací sběrnice PCI

## Hodiny PCI sběrnice

- ◆ Všechny akce jsou synchronní vzhledem k PCI hodinám
- ◆ Hodiny se mohou pohybovat v rozsahu od 0 MHz do 33.33 MHz a všechna PCI zařízení musí podporovat tento frekvenční rozsah. Frekvence se může kdykoliv měnit i zastavit.
- ◆ Revize 2.1 definuje rychlosti do 66 MHz
- ◆ **Nevyužívají odraz na vedení !!!**

## Ostatní signály PCI

- ◆ Využívají odraz na vedení



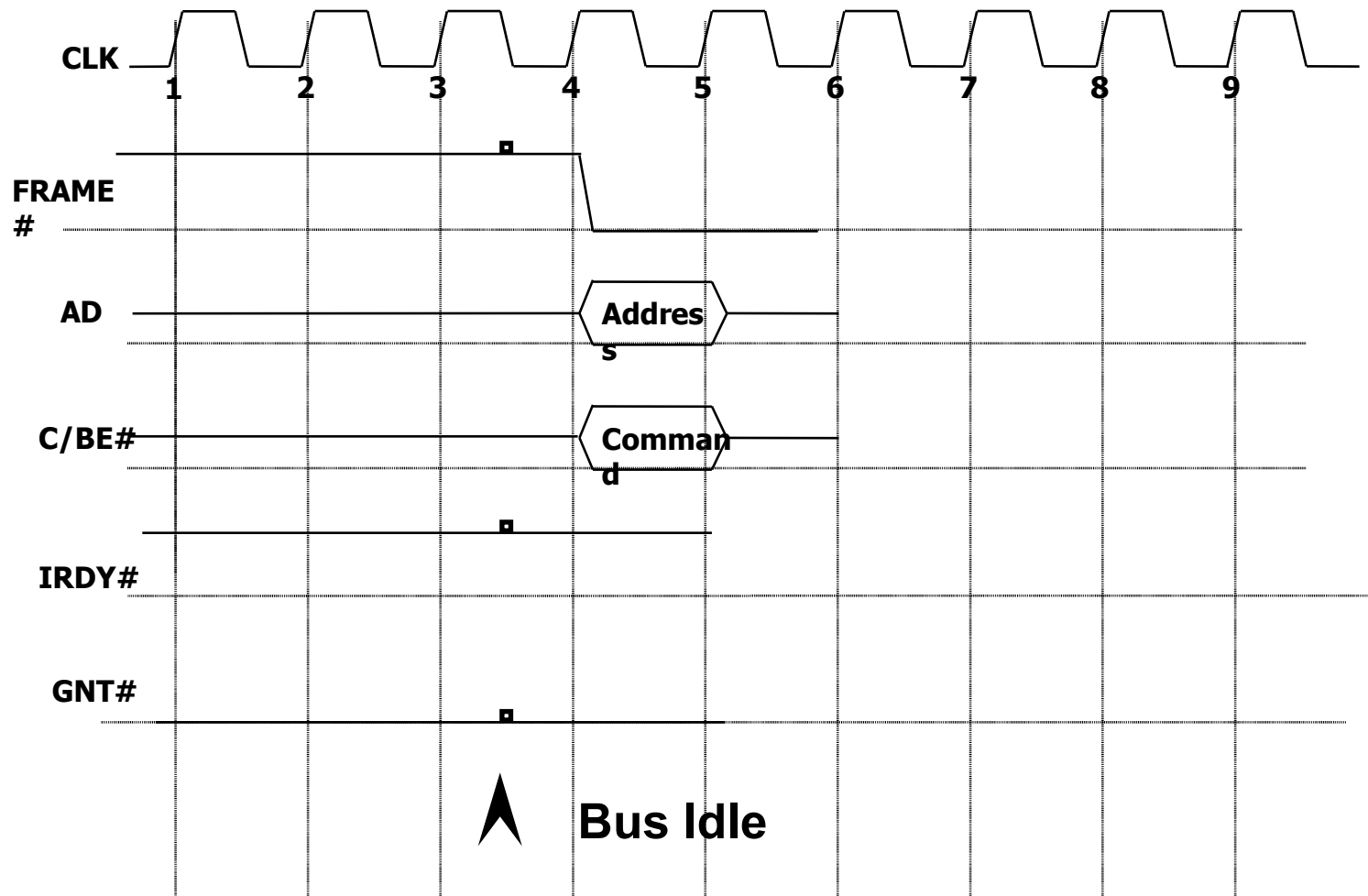
# Přenosy na sběrnici

- ◆ Burst režim – skládá se z jediné adresní fáze a dvou nebo více datových fází.
- ◆ Bus master prochází procesem přidělení pouze jednou.
- ◆ Během adresní fáze se předává startovací adresa a typ transakce.
- ◆ Všechna zařízení na sběrnici si zaznamenají adresu a typ transakce. Obojí ihned dekodují. Cílové zařízení přepíše adresu do adresního čítače a samo provádí pak jeho inkrementaci.

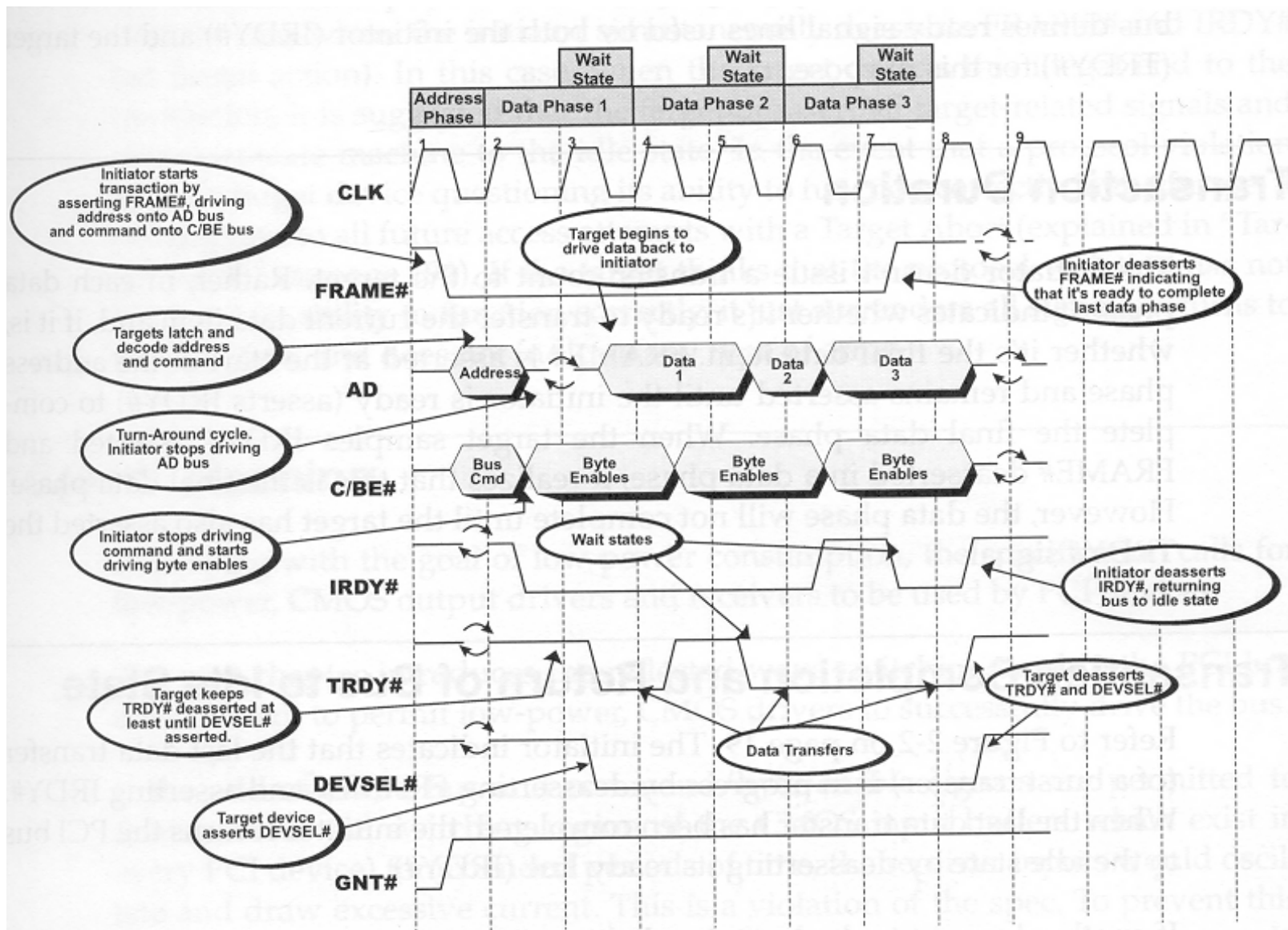
# Přenosy na sběrnici

- ◆ Režim burst není pro zařízení povinný. Jestliže master zkouší provádět burst přenos na takové zařízení, dojde k ukončení přenosu cílovým zařízením.
- ◆ Rozlišují se zařízení s jednou funkcí a multifunkční zařízení.
- ◆ Každá funkce obsahuje vlastní individuálně adresovatelný konfigurační prostor o velikosti 64 dvojslov. Zde jsou implementovány konfigurační registry. V procesu konfigurace se softwarově detekuje přítomnost „funkcí“ na sběrnici a lze tak nastavit bezkonfliktní provoz všech zařízení.

# Start transakce na sběrnici



# Typická transakce PCI



# Úvod do operací sběrnice PCI

## Adresní fáze

- ◆ Ve stejnou dobu iniciátor identifikuje cílové zařízení a typ transakce.
- ◆ Iniciátor aktivuje signál FRAME#.
- ◆ Každé PCI cílové zařízení zachycuje adresu do registru typu latch a dekoduje ji.
- ◆ Adresní fáze trvá jednu periodu hodin (vyjma 64 bitové adresování). Současně s adresou vydává iniciátor typ transakce (Command). Přítomnost platné adresy je doprovázena signálem FRAME#. Na základě těchto signálů může cílové zařízení určit, zda se bude účastnit transakce.
- ◆ Vybrané zařízení odpovídá signálem DEVSEL#.

# Úvod do operací sběrnice PCI

## Datová fáze

- ◆ Počet datových byte, které se přenesou během jedné datové fáze, je určen počtem „Command/Byte Enable“ signálů aktivovaných iniciátorem.
- ◆ Obě zařízení iniciátor i cílové zařízení musí být schopna dokončit datovou fázi.
- ◆ Jsou použity signály IRDY# i TRDY# (iniciátor ready, target ready).

## Trvání transakce

- ◆ Aktivace signálu FRAME# na počátku adresní fáze. Signál zůstane aktivní do konce datové fáze.

# ... znamená aktivitu signálu v nule

# Datová fáze

- ◆ Každá datová fáze trvá alespoň jeden takt. Není předávána délka přenosu, ale připravenost přijmout další položku a také je indikováno, zda se jedná o poslední položku.
- ◆ Poslední datová fáze je provázena  $IRDY\# = 1$  a  $FRAME\# = 0$ .



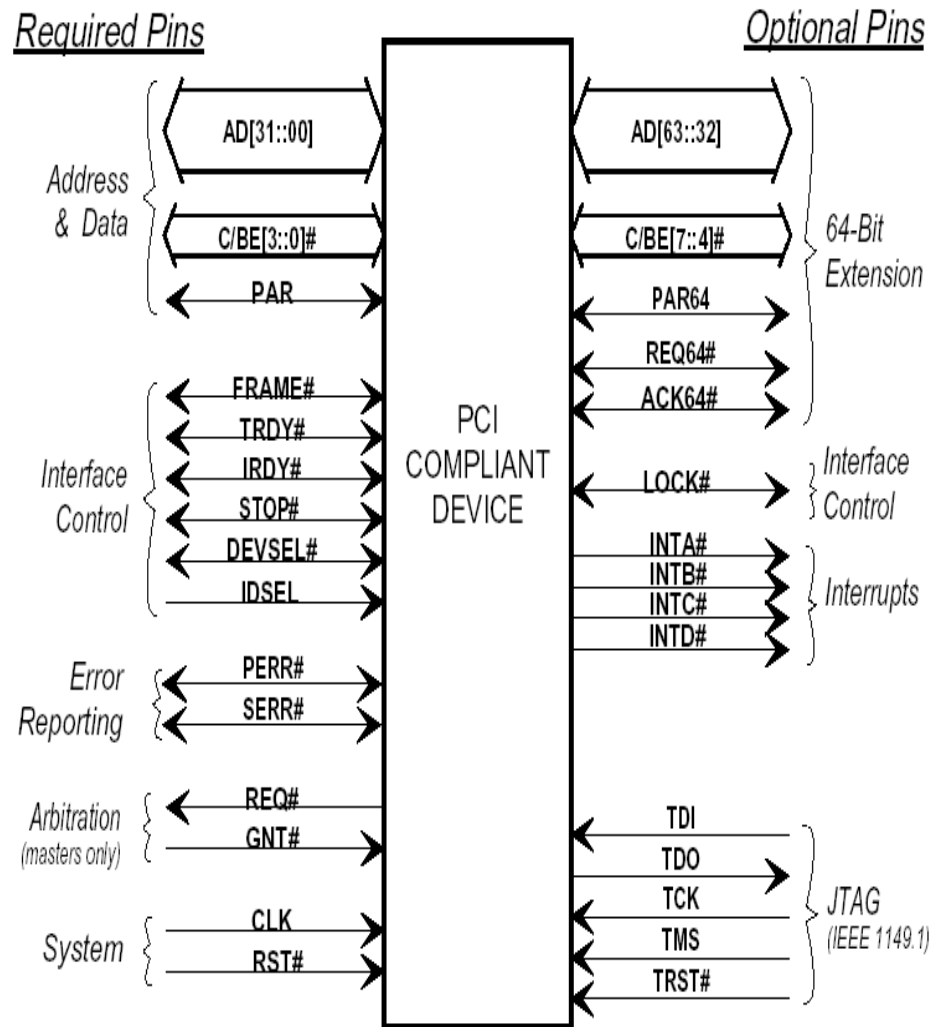
# Datová fáze (pokračování)

## Ukončení transakce a návrat sběrnice do čekacího stavu

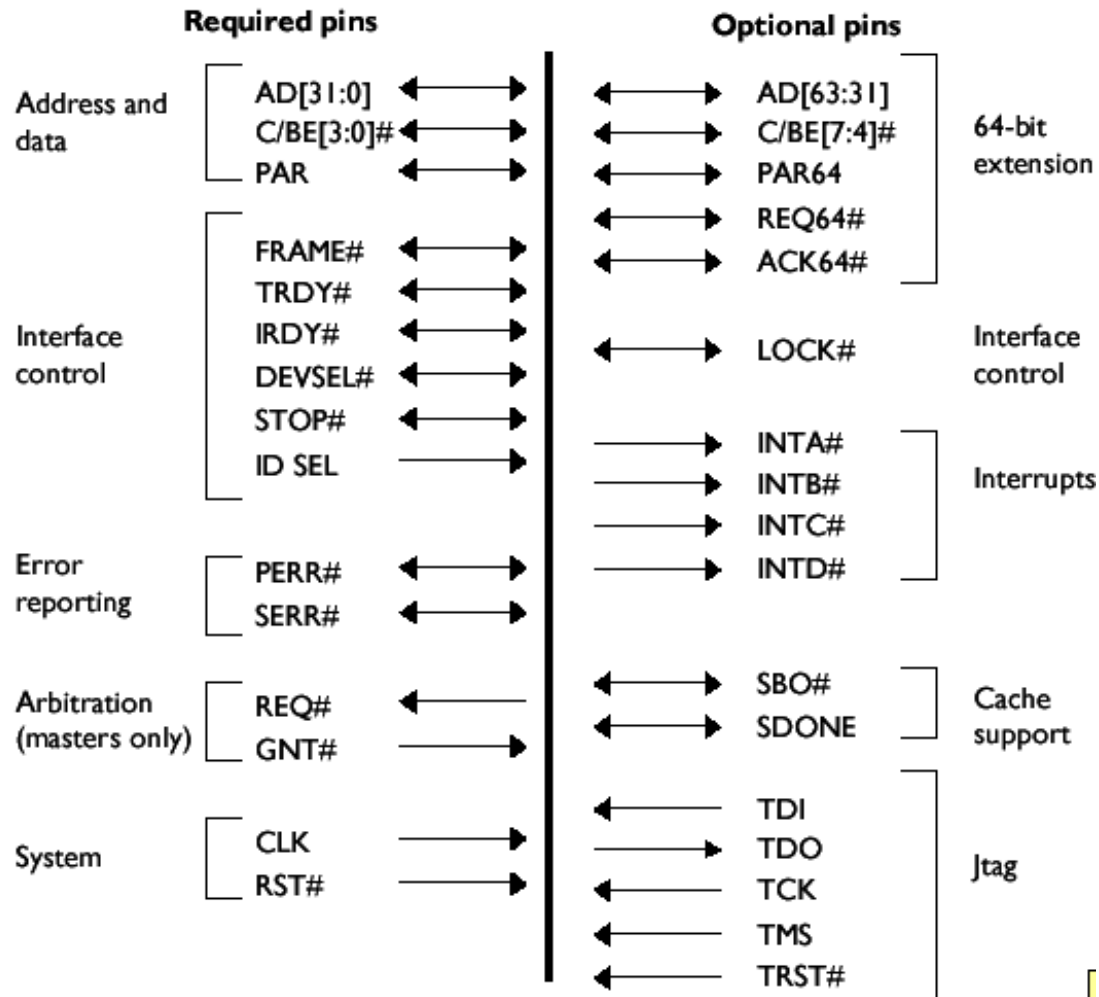
- ◆ Uvolněním signálu FRAME# a aktivací signálu IRDY#
- ◆ Po ukončení posledního datového přenosu vrací iniciátor sběrnici PCI do čekacího stavu (idle state) deaktivací signálu IRDY#

# Signály PCI

- ◆ Hodiny a Reset
- ◆ Řízení transakcí
  1. Signály iniciátoru
  2. Signály cílového zařízení
  3. Konfigurační signály
- ◆ Adresní a datové signály
- ◆ Arbitrační signály
- ◆ Chybové signály



# Signálové linky PCI



# Hodiny a Reset

## ◆ CLK

- ◆ *PCI vstup hodin*
- ◆ *Všechny signály jsou vzorkovány na náběžné hraně hodin*
- ◆ *33 MHz je ve skutečnosti 33.33333 MHz (perioda clk 30ns)*
- ◆ *Frekvence hodin se může pohybovat od 0 do 33 MHz*
  - ◆ *Frekvence se může měnit během provozu (“za jízdy”)*
  - ◆ *Z toho důvodu se nepřipouští použití fázových závěsů (PLL)*

## ◆ RST#

- ◆ *Asynchronní reset*
- ◆ *PCI zařízení musí převést všechny I/O během resetu do stavu vysoké impedance*

# Řízení transakce – signály cílového zařízení

## ◆ TRDY# – I/O

- ◆ *“T-Ready”*
- ◆ *Jestliže „target“ aktivuje tento signál, sděluje iniciátoru, že je schopen vysílat nebo přijímat data*

## ◆ STOP# – I/O

- ◆ *Používán zařízením „target“ . Oznamuje, že target potřebuje ukončit transakci*

# Řízení transakce – signály cílového zařízení

## ◆ DEVSEL# – I/O

- ◆ *Device select*

- ◆ *Část distribuovaného dekódování adresy na PCI*

- ◆ Každý target odpovídá za dekódování adresy, která je spojena s každou transakcí

- ◆ Jestliže target rozpozná svoji adresu, aktivuje DEVSEL# , aby sdělil, že se zúčastní transakce

# Řízení transakce – signály iniciátoru

- ◆ **FRAME# – I/O**

- ◆ Oznamuje počátek a konec transakce

- ◆ **IRDY# – I/O**

- ◆ *“I-Ready”*
- ◆ Aktivace iniciátorem sděluje, že je schopen vysílat nebo přijímat data

# Řízení transakce – konfigurační signály

- ◆ Používají stejné signály jako target, plus . . .
- ◆ **IDSEL – I**
  - ◆ “ID-Sel”
  - ◆ *Individuální výběrový signál zařízení pro konfiguraci – jedinečný signálová linka IDSEL pro každého agenta*
  - ◆ Řeší problém „slepice – vejce“ (“*chicken-and-egg*”)
  - ◆ Dovoluje systémovému hostu konfigurovat agenty ještě před tím, než jsou známy jejich PCI adresy, na které pak musí odpovídat



# Adresy a datové signály

- ◆ **AD[31:0] – I/O**
  - ◆ 32-bitová adresa/data
  - ◆ PCI používá *little endian* (nejnižší numerický index je LSB)
- ◆ **C/BE#[3:0] – I/O**
  - ◆ 4-bitové pole *command/byte enable*
  - ◆ Definuje příkazy PCI během adresní fáze
  - ◆ Indikuje *byte enable* během datových fází
    - ◆ Každý bit koresponduje s odpovídajícím bytem ve dvojitém slově. Např.: C/BE#[0] povoluje AD[7:0]

# Adresy a datové signály

## ◆ PAR – I/O

- ◆ *Paritní bit*
- ◆ *Používán pro korektní přenos adresy/dat a signálů C/BE#[3:0] (command/byte-enable)*
- ◆ *Operace XOR z AD[31:0], C/BE#[3:0] a PAR má dát nulu (sudá parita)*
  - ◆ *Jinak řečeno, počet jedniček na těchto 37 signálech má být sudý*

# Chybové signály

## ◆ **PERR# – I/O**

- ◆ *Indikuje parity error*
- ◆ *Agent deaktivuje svůj PERR# během konfigurace PCI*

## ◆ **SERR# – I/O**

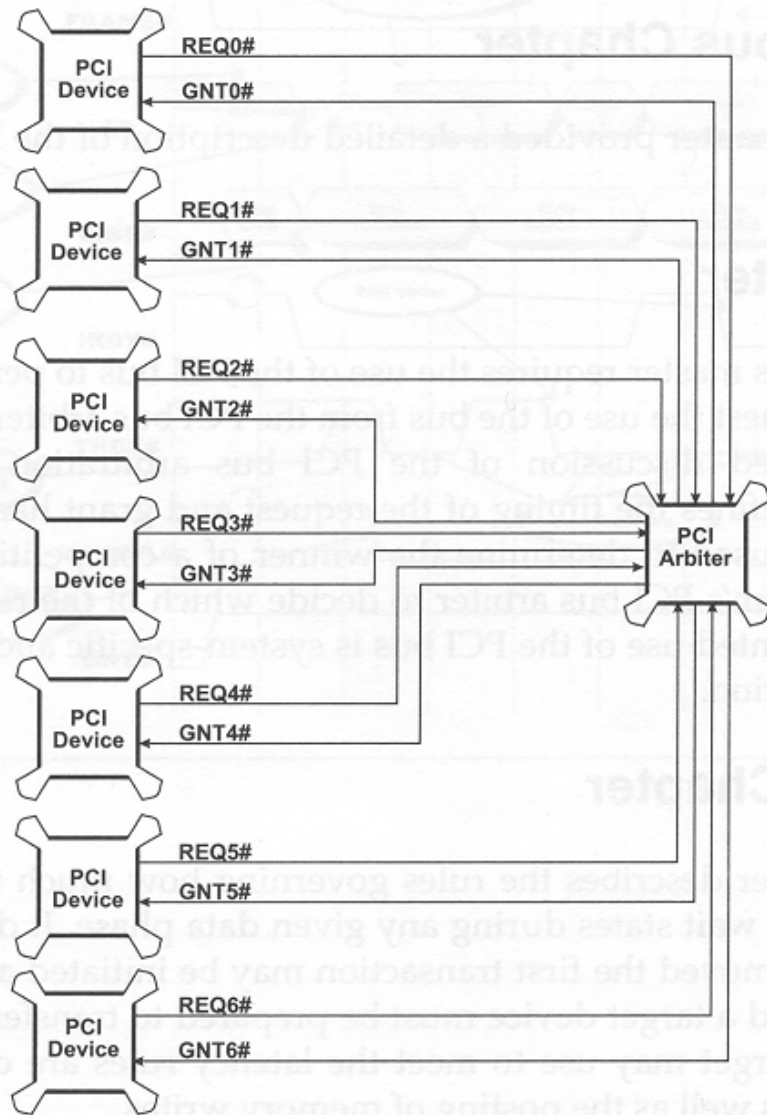
- ◆ *Indikuje vážnou systémovou chybu*
  - ◆ *Příklad: Chyba parity v adrese*
- ◆ *Může vyvolat NMI (nemaskovatelný interrupt, popř. restart) u některých systémů*

# Arbitrace PCI

# Arbitrační signály

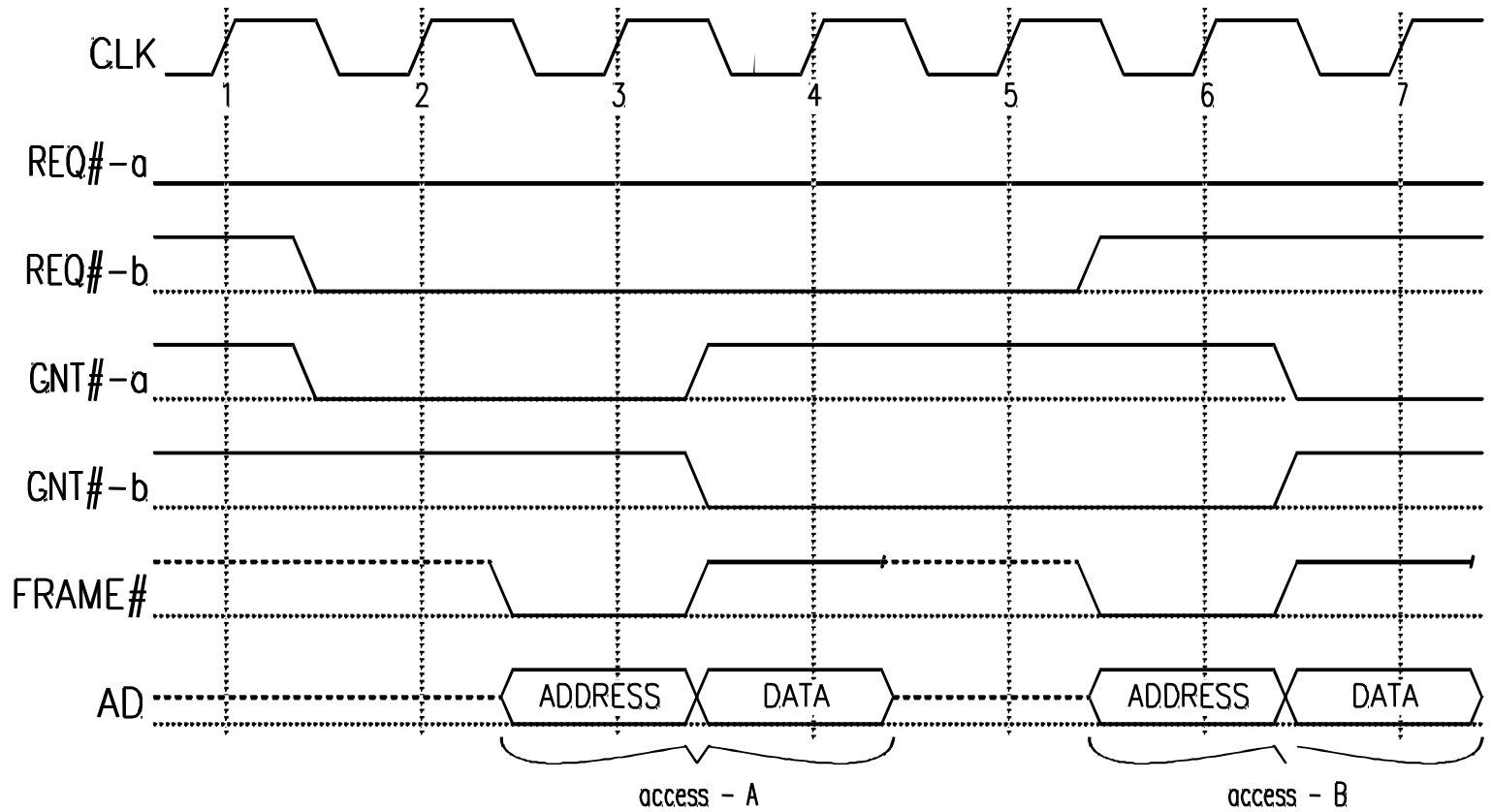
- ◆ **Týkají se pouze iniciátoru !!**
- ◆ **REQ# – 0**
  - ◆ *Aktivován iniciátorem – **žádost o sběrnici***
  - ◆ *Dvoubodový spoj s arbitrem – každý iniciátor má svoji linku REQ#*
- ◆ **GNT# – 1**
  - ◆ *Aktivována systémovým arbitrem – **potvrzení přidělení sběrnice iniciátoru***
  - ◆ *Dvoubodový spoj s arbitrem – každý iniciátor má svoji linku GNT#*

# Arbitrační signály



Každé PCI zařízení má svoji dvojici signálů:  
 $REQx\#$  a  $GNTx\#$

# PCI Bus arbitrace



# Arbitrace

- ◆ Arbitrace je vázána na přístupy
  - ◆ Master musí projít arbitrací pro každý přístup
- ◆ Centrální arbitrační schéma
  - ◆ Každý master má svůj signál žádosti i odpovědi (**request** a **grant**)
- ◆ Arbitrace je skrytá
  - ◆ Probíhá během předcházejícího cyklu sběrnice



# Parkování sběrnice

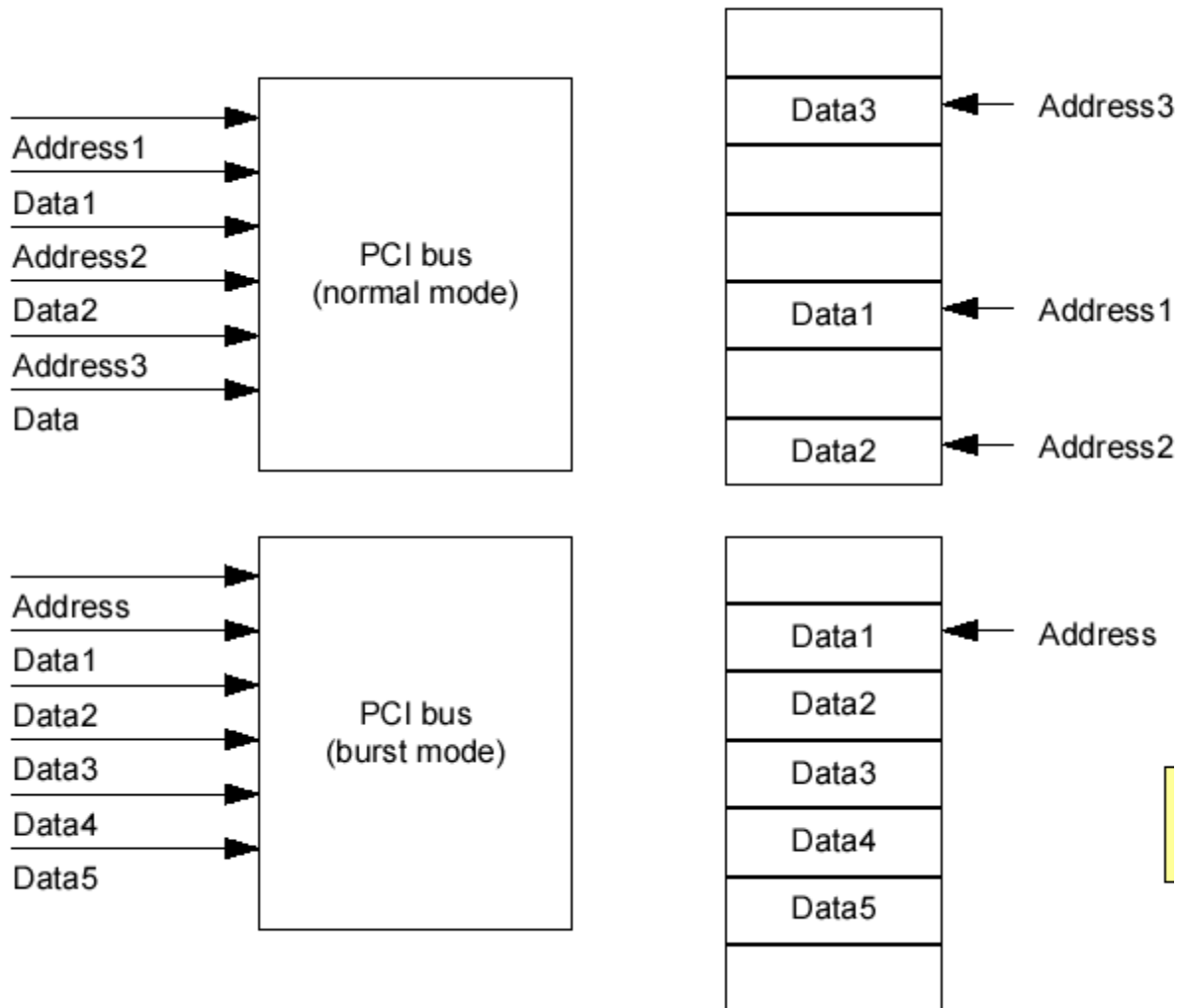
- ◆ Parkování dovoluje arbitru vybrat agenta aktivací jeho signálu GNT#. Pokud žádný jiný agent nežádá o sběrnici, může si ji původní agent ponechat.
- ◆ Arbiter určuje, jakým způsobem je proveden výběr
  - ◆ Fixní, naposledy použitý, ..., nebo žádný

# Základní operace sběrnice

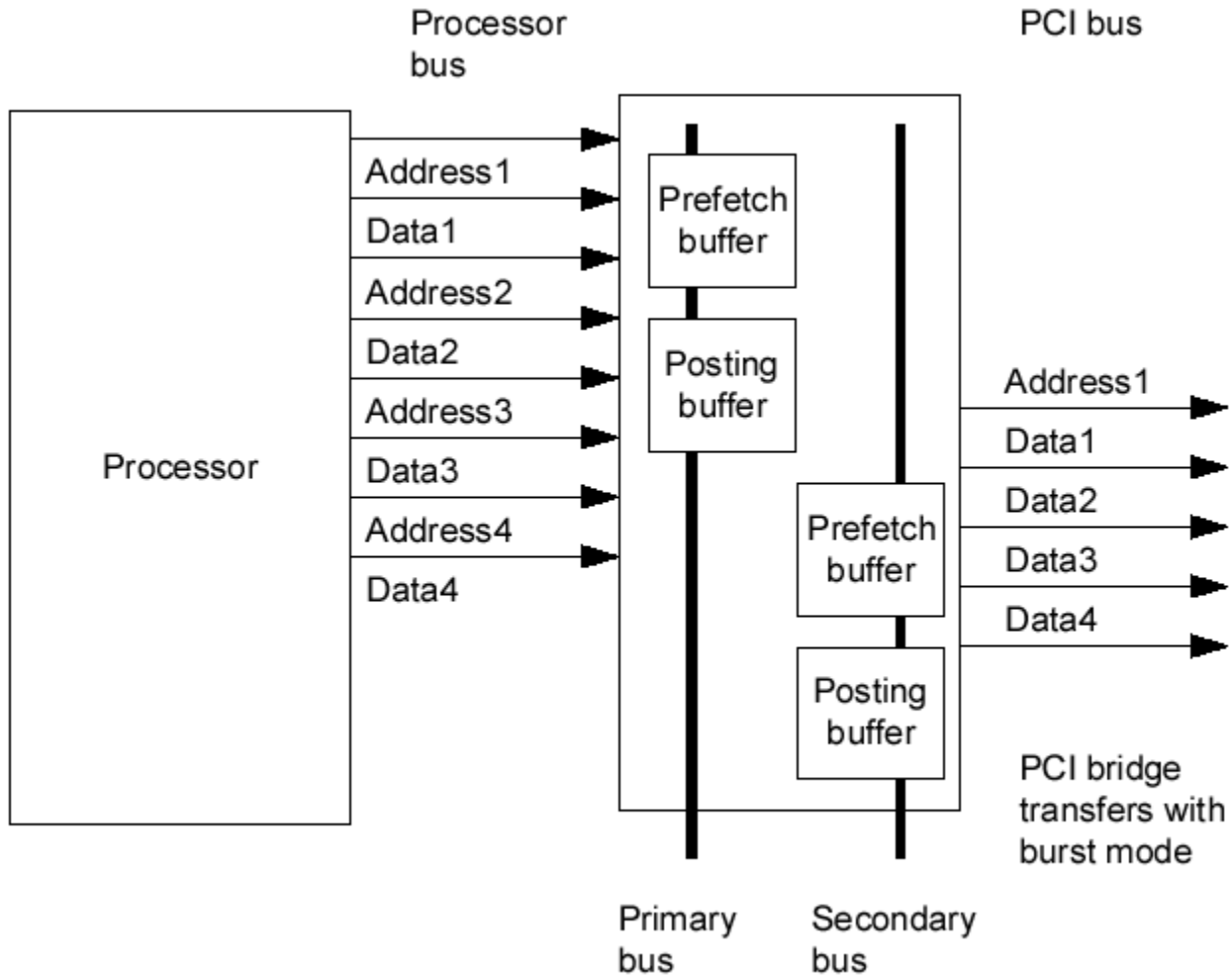
## Pojmy:

- ◆ **Doubleword**
  - ◆ *32 bitů, zkratka "DWORD"*
- ◆ **Quadword**
  - ◆ *64 bitů, zkratka "QWORD"*
- ◆ **Burst transakce**
  - ◆ *Každá transakce, obsahující více než jednu datovou fázi*
- ◆ **Idle state (neaktivní sběrnice)**
  - ◆ *Vstupuje se do něho deaktivací FRAME# a IRDY#*

# Přenosové režimy

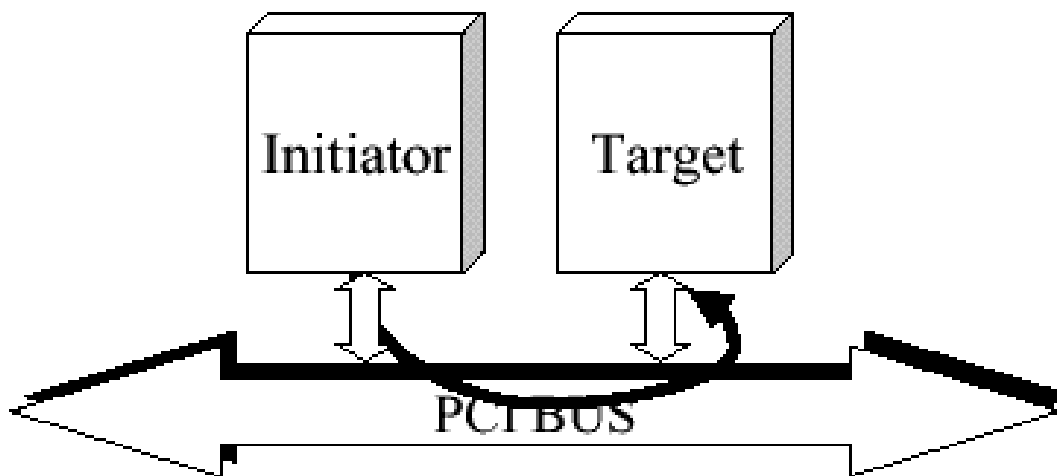


# Režim burst



# Příklad #1 – základní operace zápisu

- ◆ Čtyři - DVOJSLOVA jsou zapsána iniciátorem do cílového zařízení (target)



# Operace zápisu

## 1. Adresní fáze.

- ◆ Aktivace FRAME#.
- ◆ Nastavení linek C/BE a AD31-A0.

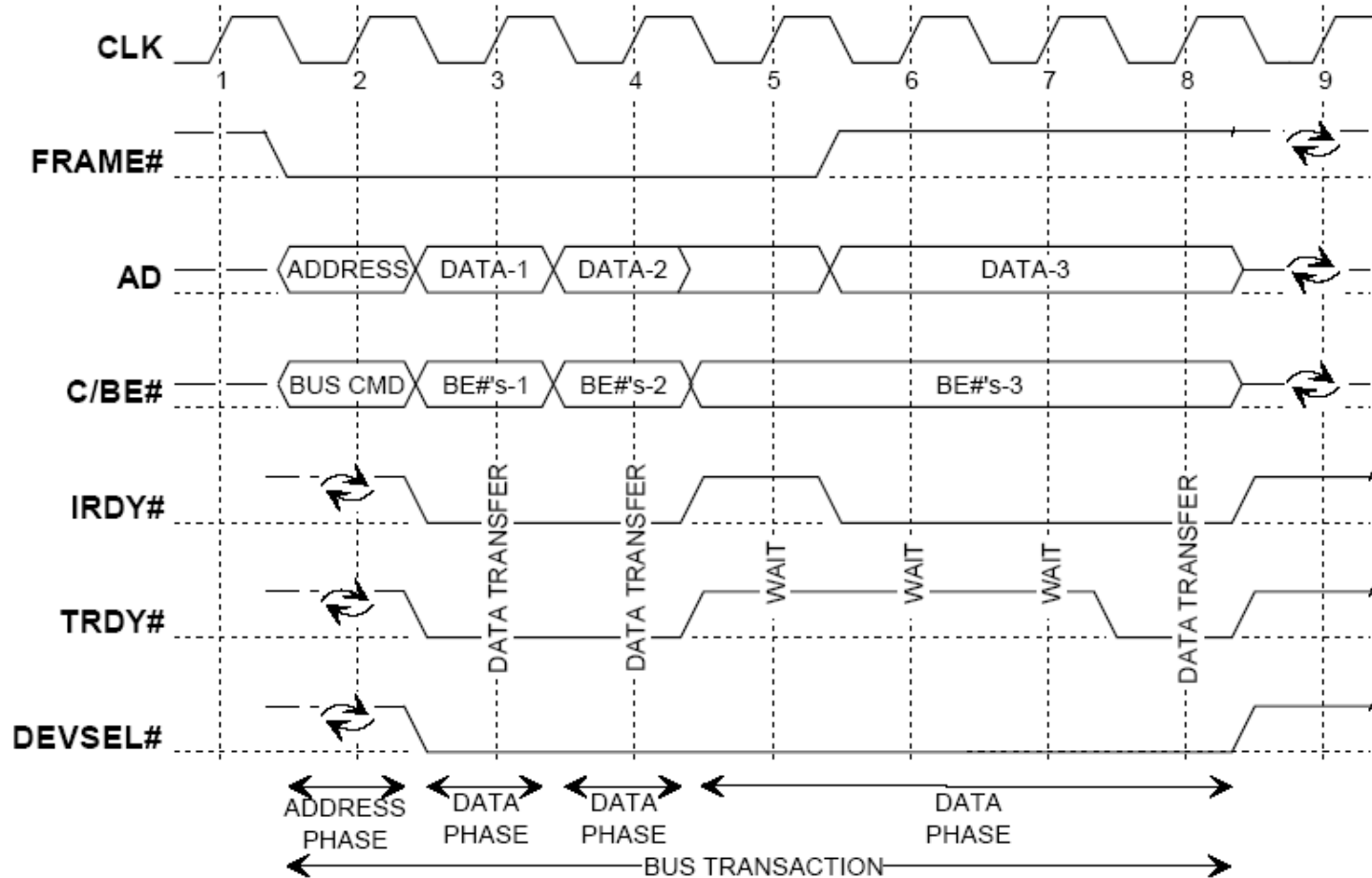
## 2. Linky pro handshake.

- ◆ Target nastavuje TRDY (Target Ready).
- ◆ Iniciátor nastavuje IRDY (Iniciátor Ready).

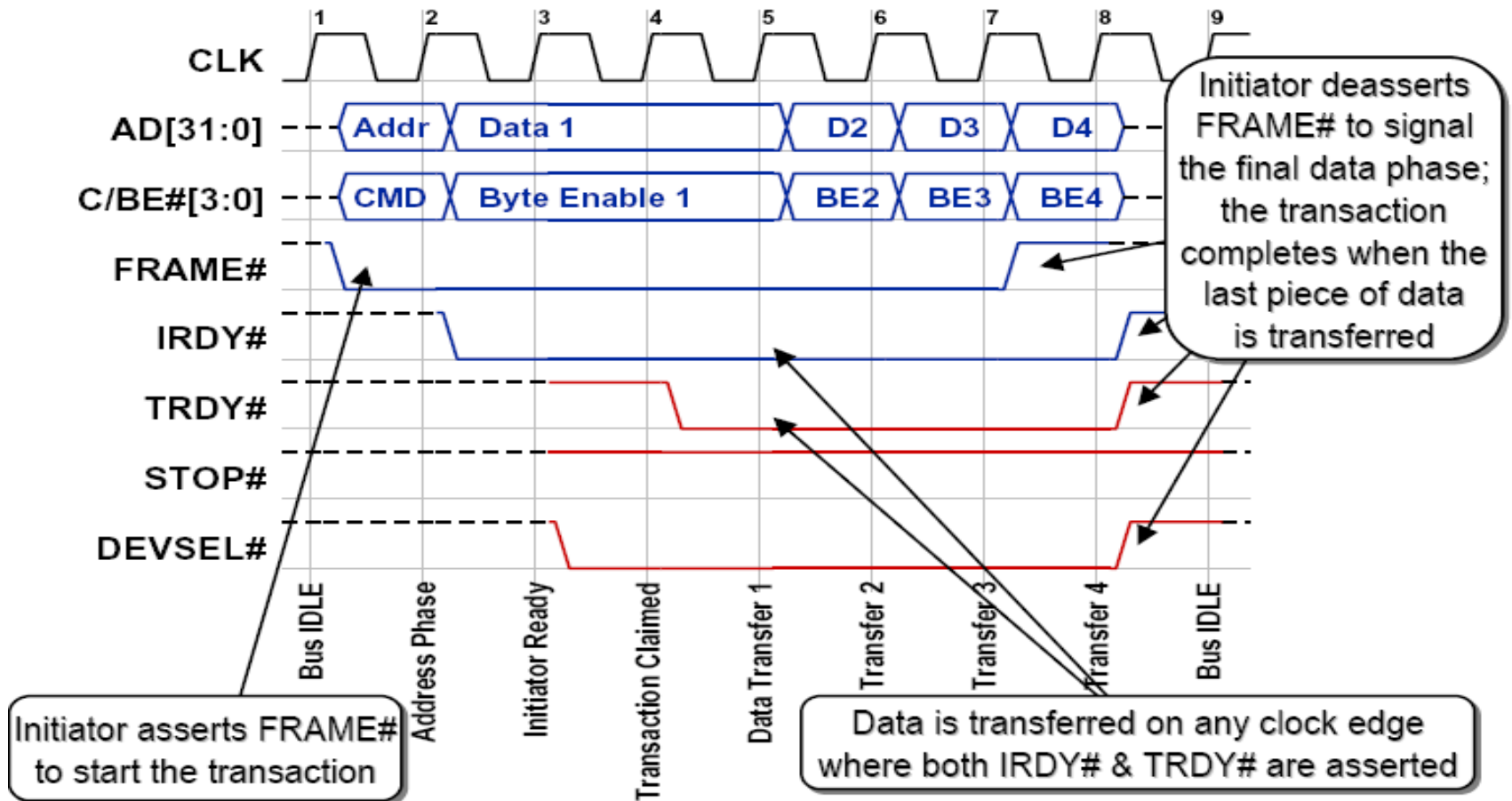
## 3. Datový přenos.

- ◆ Nastavení linek BE, které indikují velikost přenášených dat

# Operace zápisu

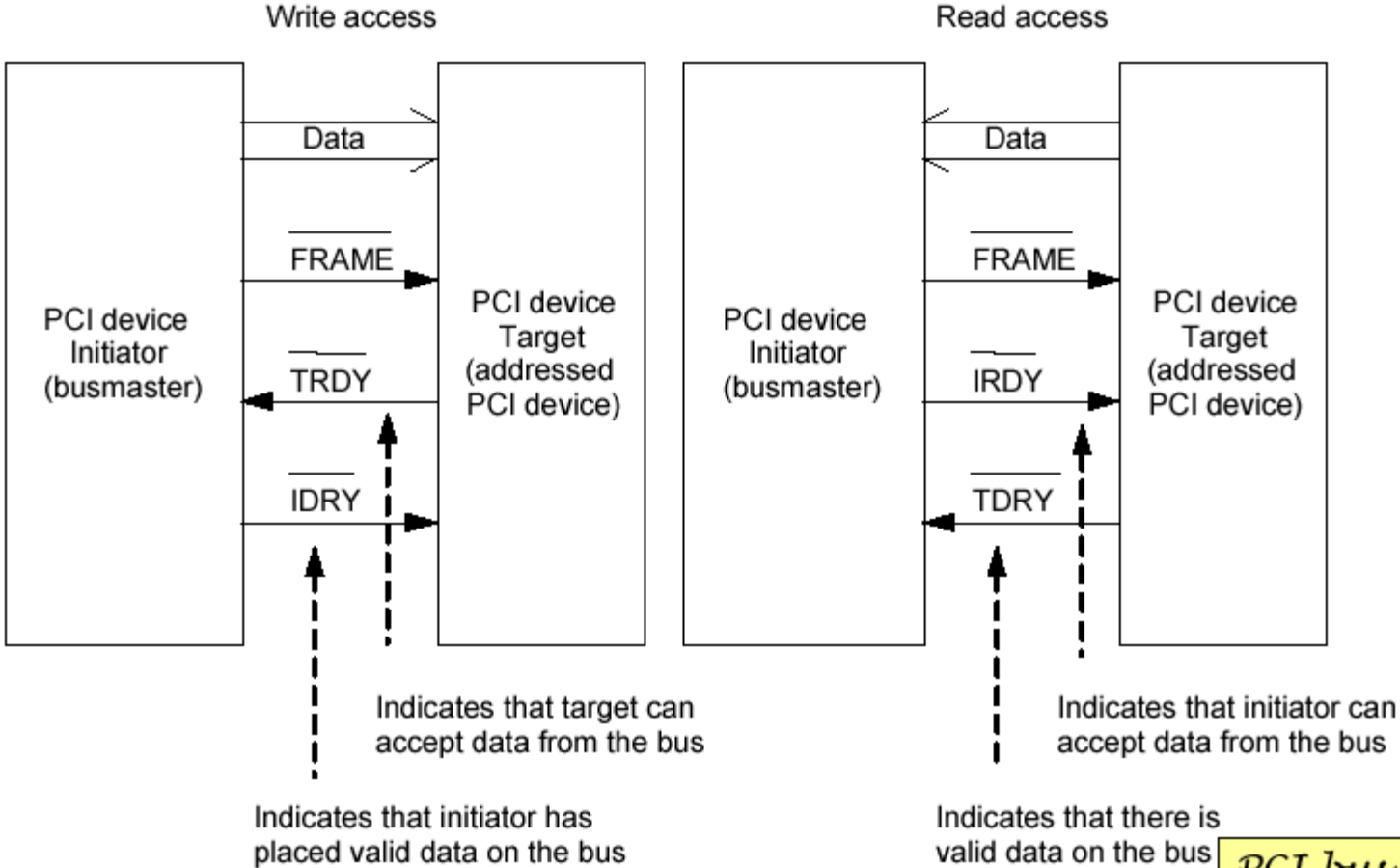


# Operace zápisu





# Sběrníkové cykly PCI



*PCI bus cycles*

# Příklad operace zápisu – poznámky

- ◆ **Iniciátor má profil fází 3-1-1-1**
  - ◆ *První data mohou být přenášena po třech cyklech hodin (idle + adresa + data = “3”)*
  - ◆ *Druhá, třetí a každá další data se přenáší v cyklech (“1-1-1”)*

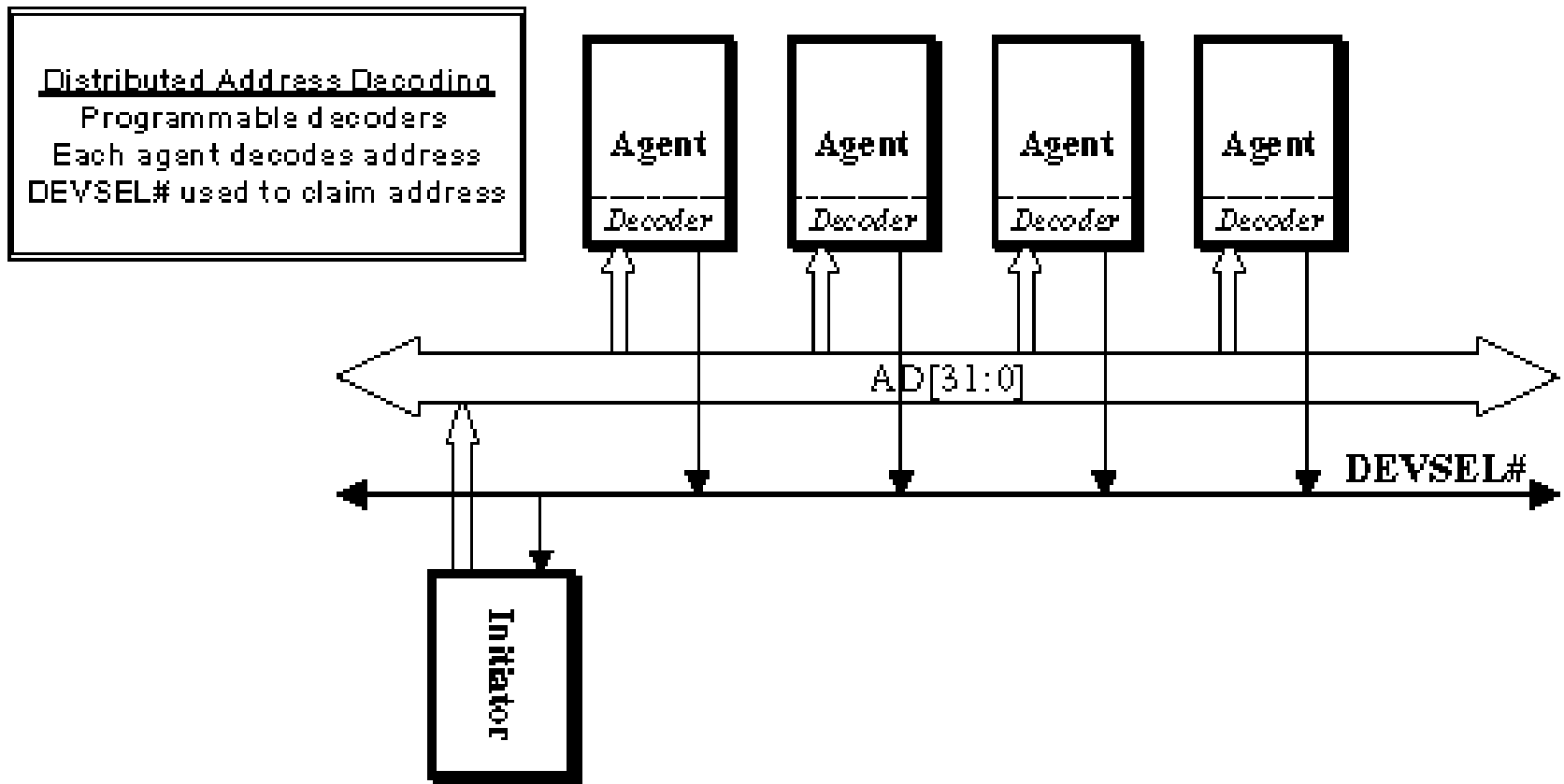
# Příklad operace zápisu – poznámky

- ◆ **Je-li profil fází 5-1-1-1**
  - ◆ *Střední rychlost dekódování – DEVSEL# nastaven ve 2 taktu hodin po FRAME#*
  - ◆ *Latence je 1 perioda hodin (nebo wait state) na počátku přenosu*
  - ◆ DEVSEL# se nastavuje v taktu 3, ale TRDY# není nastaven do taktu 4
  - ◆ *Ideální zápis do cíle (target) je 3-1-1-1*
- ◆ **Celkem 4 datové fáze, ale vyžadují 8 taktů hodin**
  - ◆ *Efektivita jen 50 %*

# Dekódování adresy u zařízení „target“

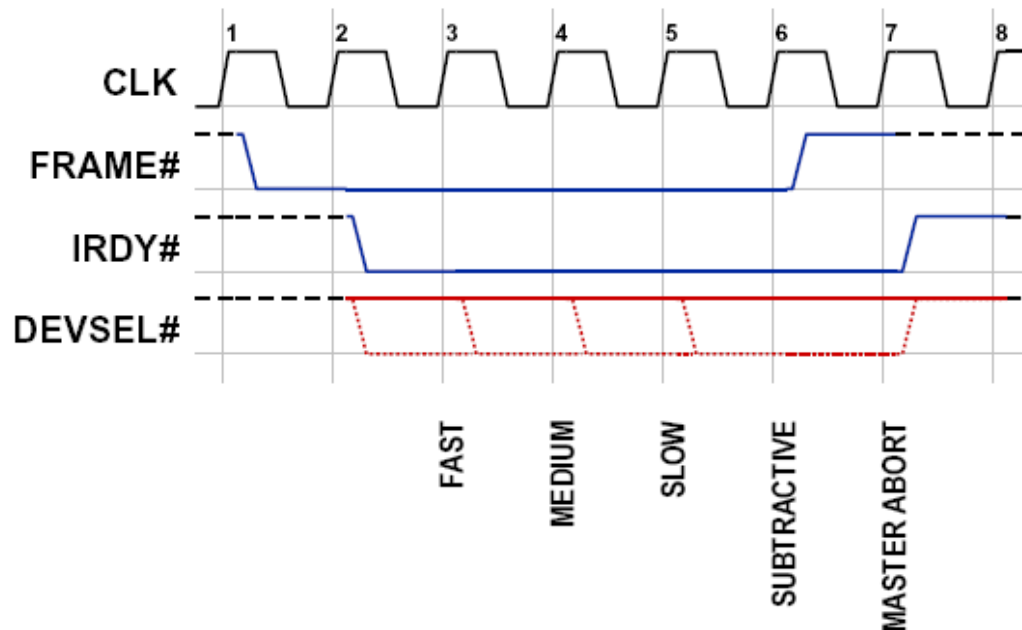
- ◆ **PCI používá distribuované dekodování adresy**
  - ◆ *Začátek transakce na PCI sběrnici*
  - ◆ *Každý potenciální „target“ na sběrnici dekoduje PCI adresu, aby určil, zda patří do jemu vyhrazeného adresního prostoru*
    - ◆ *Jednomu cílovému zařízení může být přiřazen větší adresní prostor než jinému a musí tedy reagovat na více adres*
  - ◆ *Cílové zařízení, které zahrnuje PCI adresu, oznamuje účast na transakci aktivací signálu DEVSEL#*

# Distribuované dekódování adresy



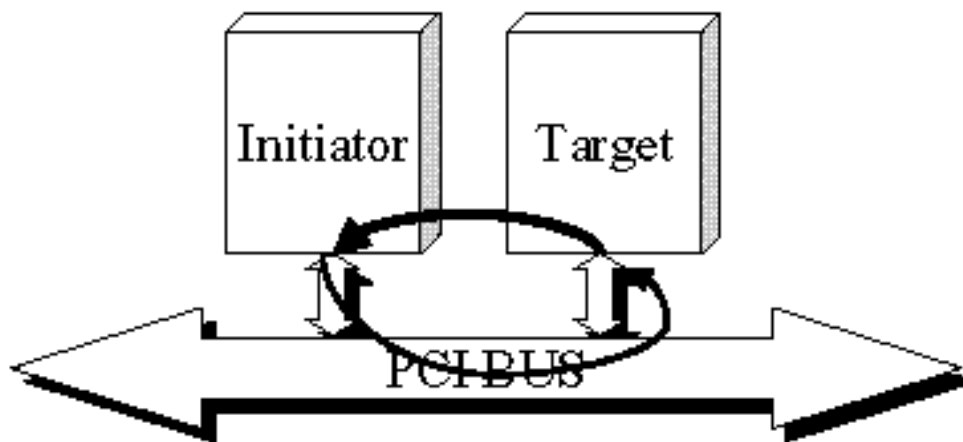
# Dekódování na straně „cíle“

- ◆ Dekódování adresy může mít rozdílnou rychlost
- ◆ Jestliže transakce zůstane bez odezvy (nikdo neaktivuje DEVSEL#), nastává **“Master Abort”**



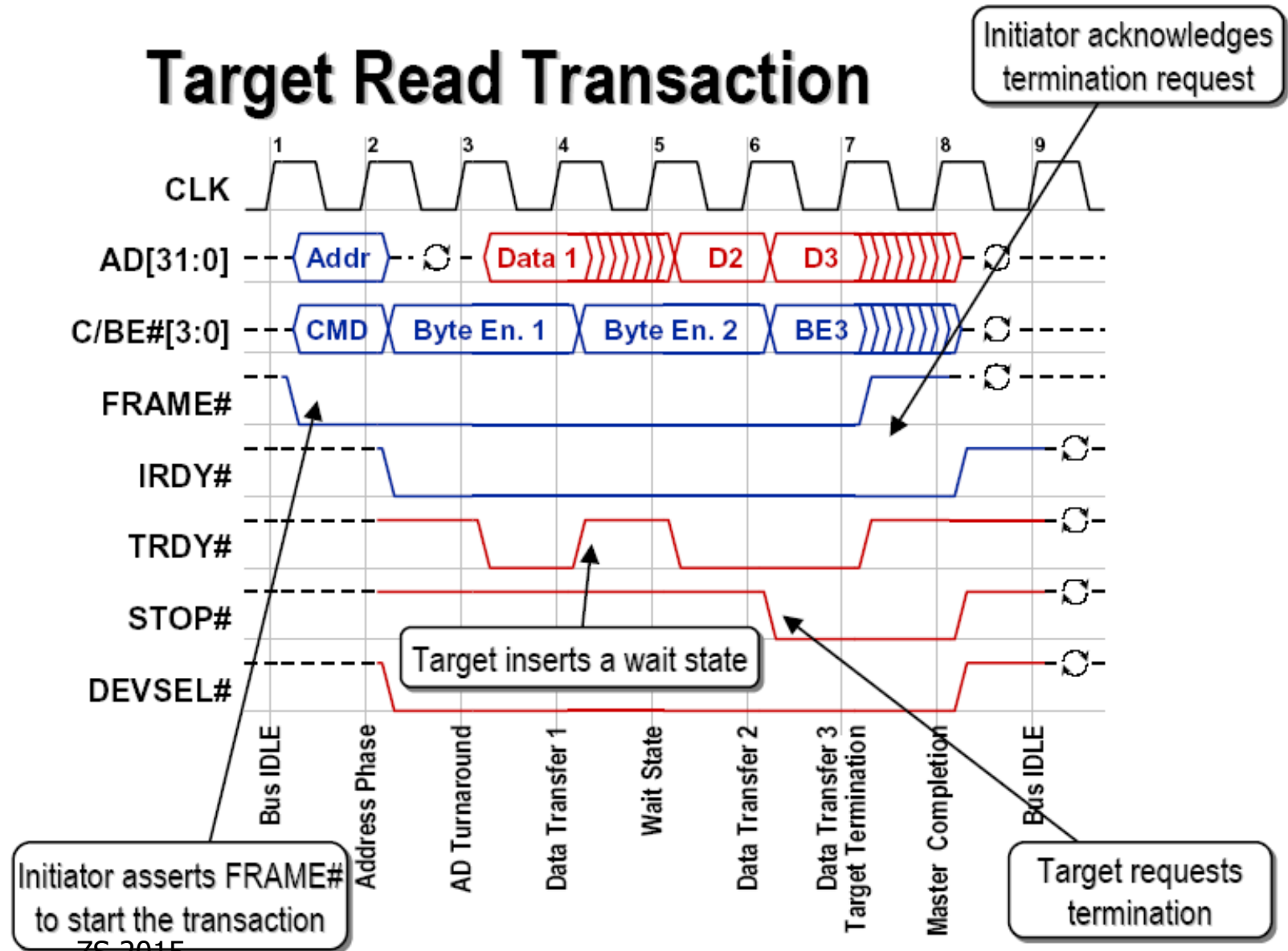
# Příklad #2 – Operace čtení

- ◆ Čtveřice - dvojslov je čtena v režimu burst z cíle do iniciátoru



# Příklad #2 – Operace čtení

## Target Read Transaction





# Další pojmy

- ◆ **Změnový cyklus (Turnaround cycle )**
  - ◆ *“Mrtvý” cyklus sběrnice zabraňující konfliktům na sběrnici*
- ◆ **Čekací stav (Wait state)**
  - ◆ *Cykl sběrnice, ve kterém je možné data přenášet, ale žádná data přenášena nejsou*
  - ◆ *Target deaktivuje TRDY#, aby oznámil nepřipravenost*
  - ◆ *Initiator deaktivuje IRDY#, aby oznámil nepřipravenost*
- ◆ **Ukončení ze strany „cíle“**
  - ◆ *Target aktivuje signál STOP# a tím dává najevo ukončení aktuální transakce*

# Čtení „cíle“ – důležité

- ◆ Čekací stavy mohou být vkládány dynamicky iniciátorem nebo „cílem“ deaktivací signálu IRDY# nebo TRDY#
- ◆ Kterýkoli z obou agentů může signalizovat konec transakce
  - ◆ *Cílové zařízení signalizuje ukončení aktivací STOP#*
  - ◆ *Iniciátor signalizuje dokončení deaktivací FRAME#*

# Žádný nebo jeden čekací stav

- ◆ **Agent (s jedním čekacím stavem) vkládá čekací stav na počátku každé datové fáze**
  - ◆ *To se praktikuje tehdy, jestliže agent – implementovaný starší technologií potřebuje interně zřetězit kritické cesty (nemůže je provést v jednom taktu)*
  - ◆ *Dojde k redukci šířky pásma na 50%*

# Žádný nebo jeden čekací stav

- ◆ **Nutnost vložit čekací stav typicky nastává, když jeden z agentů poskytuje data (zápis iniciátorem nebo čtení ze zařízení typu „target“)**
  - ◆ *Důvod je ten, že takový agent by musel vzorkovat odpovídající  $xRDY\#$  signál aby viděl, že druhý agent akceptoval data, potom by musel aktivovat rozvětvený signál 32 hodin. vstupů pro registry  $AD[31:0]$ , popř. i  $C/BE\#[3:0]$ , . . . potom dodat další položku dat na PCI a to vše během 11 ns!*
  - ◆ K tomu 11 ns činí interní distribuce hodinového signálu

# Typy ukončení transakce „cílovým“ agentem

- ◆ **Target Retry**

*“Nejsem připraven, zkus to později”*

- ◆ **Target Disconnect with Data**

*“Nemohu další sousto . . . OK, jen jedno.”*

- ◆ **Target Disconnect Without Data**

*“Nemohu další sousto . . . doopravdy!”*

- ◆ **Target Abort**

*“Vážné varování!”*

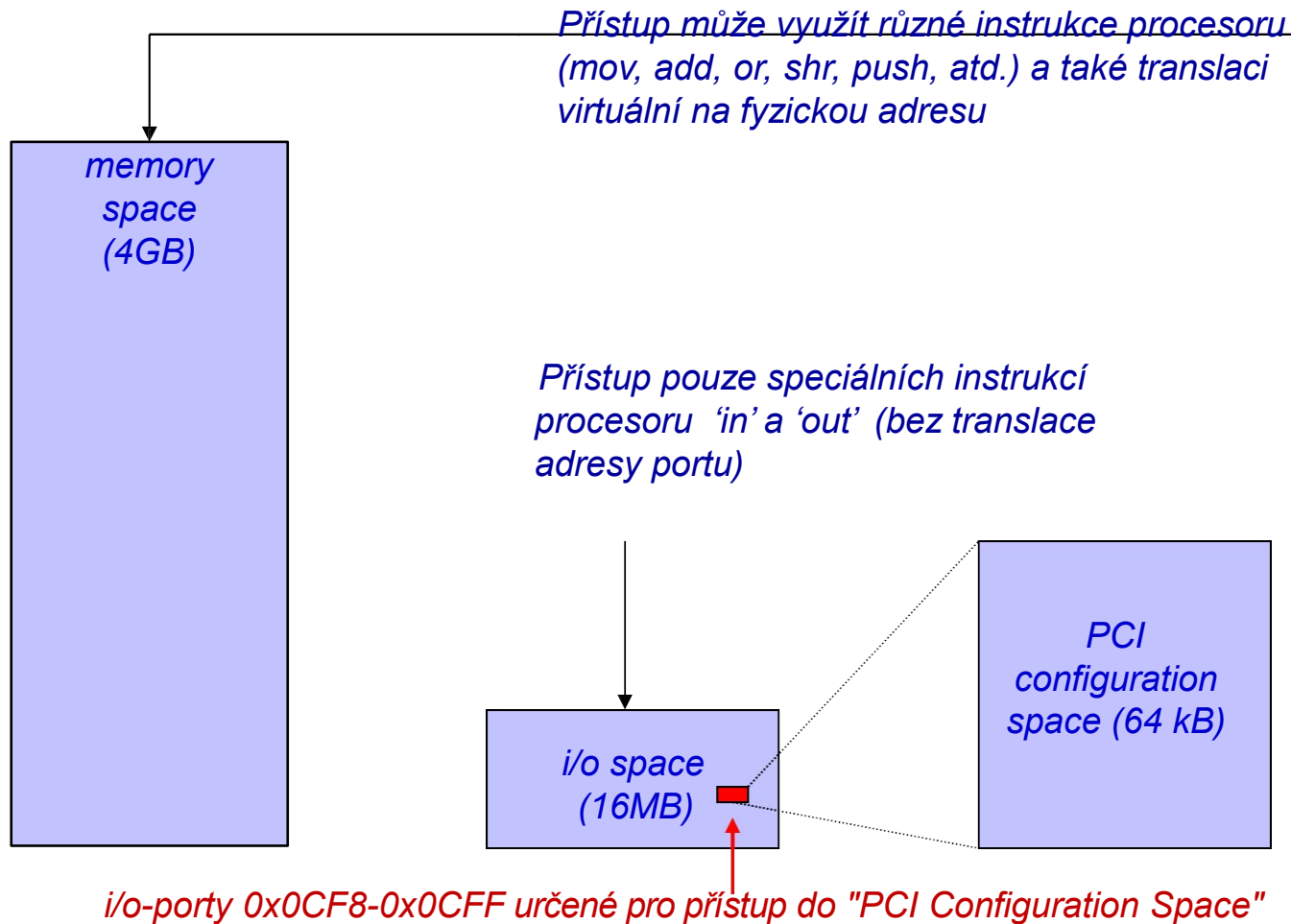
# Adresování PCI a povely sběrnice

# Adresní prostory PCI

**PCI target může zahrnovat až tři různé typy adresních prostorů**

- ◆ *Konfigurační prostor*
  - ◆ Ukládání základních informací o zařízení
  - ◆ Dovoluje centrálnímu zařízení (přes O/S) programovat a nastavovat zařízení
- ◆ *I/O prostor*
  - ◆ Používaný hlavně jen pro PC periferie
- ◆ *Paměťový prostor*
  - ◆ Používaný pro vše ostatní

# Přístup do adresních prostorů





# Adresní prostory PCI

## Konfigurační prostor

- ◆ Obsahuje základní informace o zařízení, např. výrobce a třídu zařízení
- ◆ *Dovoluje „Plug-N-Play“*
  - ◆ Bázové adresní registry umožňují dynamické mapování agenta do paměťového nebo I/O prostoru
  - ◆ Programovatelné přerušovací linky umožňují softwarovému driveru naprogramovat PC desku s IRQ až po startu systému (bez jumperů („šlůsátka“)!)

# Adresní prostory PCI

## Konfigurační prostor (pokračování)

- ◆ *Obsahuje 256 bytů*
  - ◆ Prvých 64 bytů (00h – 3Fh) tvoří standardní konfigurační hlavičku, definovanou specifikací PCI
  - ◆ Zbýlých 192 bytů (40h – FFh) reprezentuje uživatelsky-definovatelný konfigurační prostor
    - ◆ *Tato oblast může uchovávat např. informace, specifické pro použití PC karty softwarovým driverem*

# IO prostor

- ◆ *Do tohoto prostoru se mapují základní periferní zařízení PC (klávesnice, seriové porty, atd.)*
- ◆ *Specifikace PCI připouští, aby agent obsadil I/O prostor ve velikosti od 4 bytů do 2GB*
  - ◆ Pro systémy x86 je maximum 256 bytů v souvislosti se sběrnici ISA

# Paměťový prostor

## Paměťový prostor

- ◆ *Tento prostor je používán pro vše ostatní – jedná se o adresní prostor pro obecné využití*
  - ◆ Specifikace PCI doporučuje, aby zařízení používala tento prostor i když se jedná o periferie
- ◆ *Agent může vyžadovat paměťový prostor v rozsahu od 16 bytů do 2GB*
  - ◆ Specifikace PCI doporučuje, aby agent obsadil paměťový prostor o velikosti nejméně 4kB, aby se redukovala šíře adresních dekodérů agentů

# Povely PCI

- ◆ **PCI rozeznává až 16 různých 4-bitových povelů**
  - ◆ Konfigurační povely
  - ◆ Paměťové povely
  - ◆ I/O povely
  - ◆ Speciální povely
- ◆ **Povel je generován iniciátorem během adresní fáze na linkách C/BE# (prvá aktivace FRAME# během transakce)**

# Pověly PCI

<b>C/BE[3::0]#</b>	<b>Typ povelu</b>	
0000	Interrupt Acknowledge	
0001	Special Cycle	
0010	I/O Read	
0011	I/O Write	
0100	Reserved	
0101	Reserved	
0110	Memory Read	
0111	Memory Write	
1000	Reserved	
1001	Reserved	
1010	Configuration Read	Pomocí IDSEL
1011	Configuration Write	Pomocí IDSEL
1100	Memory Read Multiple	
1101	Dual Address Cycle	
1110	Memory Read Line	
1111	Memory Write and Invalidate	

# Konfigurace PCI

# Inicializace systému

- ◆ Konfigurace dovoluje softwarovou inicializaci systému (BIOS)
- ◆ Každé zařízení má konfigurační registry
- ◆ Po připojení napájení software „prochází sběrnici (scan)
- ◆ Software analyzuje požadavky systému
- ◆ Jednotlivá zařízení se konfiguruje nastavením konfiguračních registrů



# Konfigurační transakce

- ◆ Specifické příkazy sběrnice
  - ◆ čtení konfigurace (C/BE# = 1010)
  - ◆ zápis konfigurace (C/BE# = 1011)
- ◆ Typicky jednoduchá fáze
  - ◆ Připouští se Burst, ale je užíván jen zřídka
- ◆ Typ 0
  - ◆ lokální PCI sběrnice (konfiguruje agenty na stejném segmentu
    - ◆ linky IDSEL určují zařízení
    - ◆ adresové pole ukazuje registr
- ◆ Typ 1
  - ◆ Konfiguruje agenty sběrnice přes PCI-PCI most
    - ◆ adresové pole určuje sběrnici, zařízení a registr

# Konfigurační hlavička typ 00h

31		16		15		0		
Device ID				Vendor ID				00h
Status				Command				04h
Class Code						Revision ID		08h
BIST		Header Type		Latency Timer		Cache Line Size		0Ch
Base Address Registers								10h
								14h
								18h
								1Ch
								20h
Cardbus CIS Pointer								24h
Subsystem ID				Subsystem Vendor ID				28h
Expansion ROM Base Address								2Ch
Reserved						Capabilities Pointer		30h
Reserved								34h
Reserved								38h
Max_Lat		Mln_Gnt		Interrupt Pin		Interrupt Line		3Ch

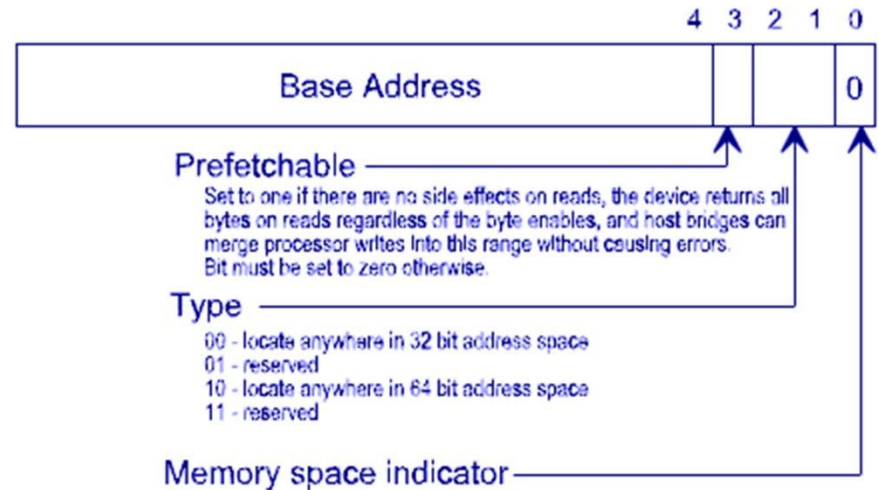
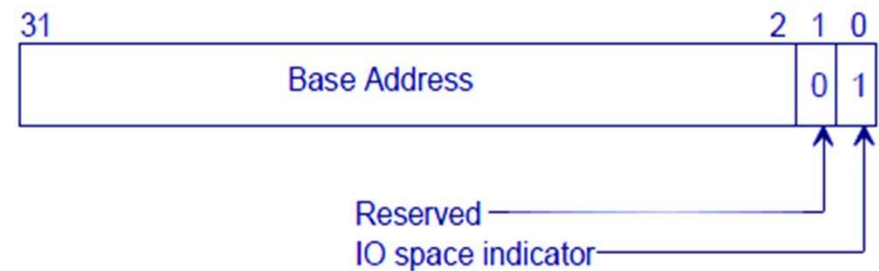


Figure 6-5: Base Address Register for Memory



(PCI Local Bus Specification, Revision 2.2)

# Konfigurační příkazy

## □ Dvě dvojslova v I/O prostoru se používají pro generování konfigurační transakce

- ❖ 0CF8h udává read/write registr, CONFIG\_ADDRESS.
- ❖ 0CFCh udává read/write registr, CONFIG\_DATA.



## □ Enumerace sběrnice

- ❖ Zkouší číst Vendor- a Device ID registr pro každou kombinaci čísla sběrnice a čísla zařízení pro funkci zařízení s #0
- ❖ Tím se dozví, že zařízení existuje a může pak programovat adresy zařízení mapovaných do paměťového nebo I/O prostoru.

# Konfigurační porty PCI na PC

- ◆ Všechny PC založené na x86 užívají IO sběrnici k určení adresy a dat pro konfiguraci PCI
  - ◆ Adresní port je široký 4 byty a nachází se na adrese 0x0CF8
  - ◆ Datový port je široký 4 byty a je umístěn na adrese 0x0CFC
- ◆ Konfigurační cykly nejdříve zapíše (IO IN) kanonickou adresu do adresního portu
  - ◆ Čtením datového portu (IO OUT) se vrátí obsah adresovaného konfiguračního registru o šířce 4 byty
  - ◆ Zápis do datového portu (IO IN) nastaví konfigurační registr



### Base Address for PCI Memory Space

- TYP 00 umístěný kdekoliv v dolních 4GB  
 01 umístěný nad 1MB  
 10 umístěný kdekoliv nad 4GB  
 11 rezervováno



### Base Address for PCI I/O Space

# BAR Management

Base Address Register

- ◆ Zapiš samé 1 do BAR a přečti
  - ◆ Vráti-li se samé 0, pak není použit
  - ◆ Je-li výsledek čtení nenulový, testujeme poslední bit:
    - ◆ Je-li 1 potom se jedná o přiřazení IO
    - ◆ Je-li 0 potom se jedná o přiřazení paměti
    - ◆ Pozice posledního významového bitu se používá pro určení velikosti (tzn. je-li nastaven bit 6, pak dekodér vyžaduje  $2^6 = 64$  byte prostoru)

# BAR Management (pokr.)

- ◆ Jsou-li do BARu zapsány samé 1 a návratová hodnota je 0xFFFF0000, potom:
  - ◆ BAR je dekodér paměti
  - ◆ BAR není s funkcí prefetch
  - ◆ BAR vyžaduje adresu < 4GB
  - ◆ BAR vyžaduje prostor o velikosti  $2^{16}$  neboli 64 KB
- ◆ 0xFFFF0008 by byl stejný ale s funkcí prefetch
- ◆ 0xFFFF000A a navíc by ještě byl 64 bitový

# BAR Management (pokr.)

- ◆ Jsou-li do BARu zapsány samé 1 a návratová hodnota je 0xFFFFF001, potom:
  - ◆ BAR je IO dekodér
  - ◆ BAR vyžaduje adresu  $< 16$  KB
  - ◆ BAR vyžaduje prostor  $2^{12}$  neboli 4 KB
- ◆ Minimální velikost paměti je 16 bytů
- ◆ Minimální velikost IO je 4 byty

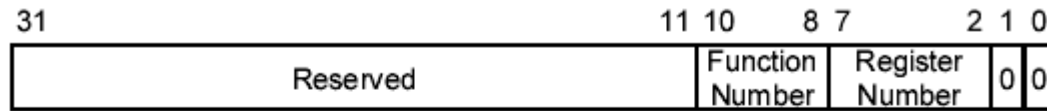


# Koncepce Plug-and-Play

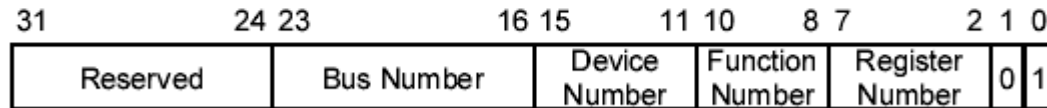
## Plug-and-Play (PNP)

- ◆ *Připouští, aby se karty vkládaly do libovolného konektoru bez změny nastavení přepínačů nebo jumperů*
  - ◆ Mapování adres, IRQ, COM portů, atd., se určuje dynamicky během startu systému
- ◆ *Aby systém PNP pracoval, musí karty obsahovat základní informace pro BIOS a operační systém, např.:*
  - ◆ Typ karty a zařízení
  - ◆ Požadavky na paměťový prostor
  - ◆ Požadavky na přidělení přerušení

# Koncepce Plug-and-Play



**Type 0**



**Type 1**

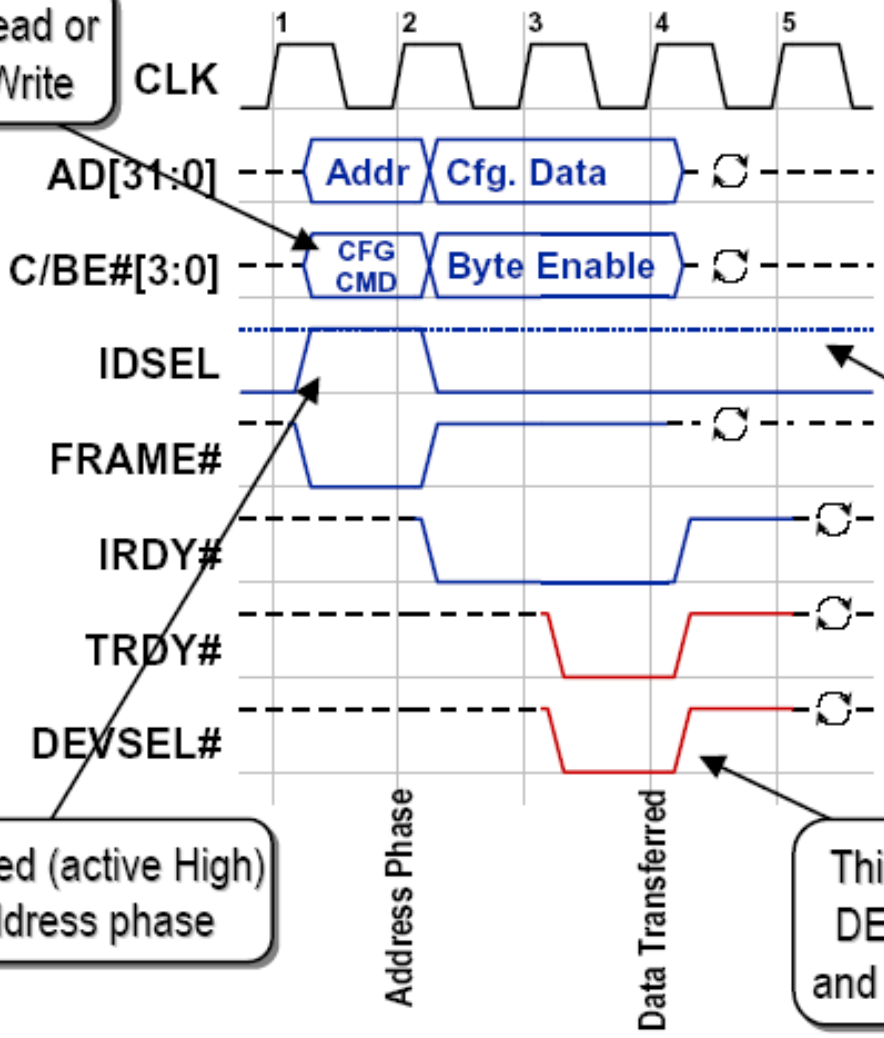
# Konfigurační transakce

- ◆ **Je generována hostem nebo mostem PCI-to-PCI**
- ◆ **Používá signály IDSEL jako signály „chip select“**
  - ◆ *Vyčleněné dekodování adresy*
  - ◆ *Každý agent má svůj jedinečný signál IDSEL*
- ◆ **Jedná se typicky o jednu datovou fázi**
  - ◆ *Burst se přípouští, ale používá se velmi zřídka*
- ◆ **Dva typy (specifikované pomocí AD[1:0] v adr. fázi)**
  - ◆ *Typ 0: Konfiguruje agenty na stejném segmentu sběrnice*
  - ◆ *Typ 1: Konfiguruje napříč mezi mosty PCI-to-PCI*

# Configuration Example

C/BE#	Command
0000	Interrupt Ack.
0001	Special Cycle
0010	I/O Read
0011	I/O Write
0100	Reserved
0101	Reserved
0110	Memory Read
0111	Memory Write
1000	Reserved
1001	Reserved
1010	Config. Read
1011	Config. Write
1100	Memory Read Mult.
1101	Dual Address Cycle
1110	Memory Read Line
1111	Memory Write & Inv.

Configuration Read or Configuration Write



Note that the host can do anything it wants to IDSEL outside of a configuration address phase

IDSEL is asserted (active High) during the address phase

This time, the target asserts DEVSEL# based on IDSEL and not based on the address

# Elektrické a časové specifikace

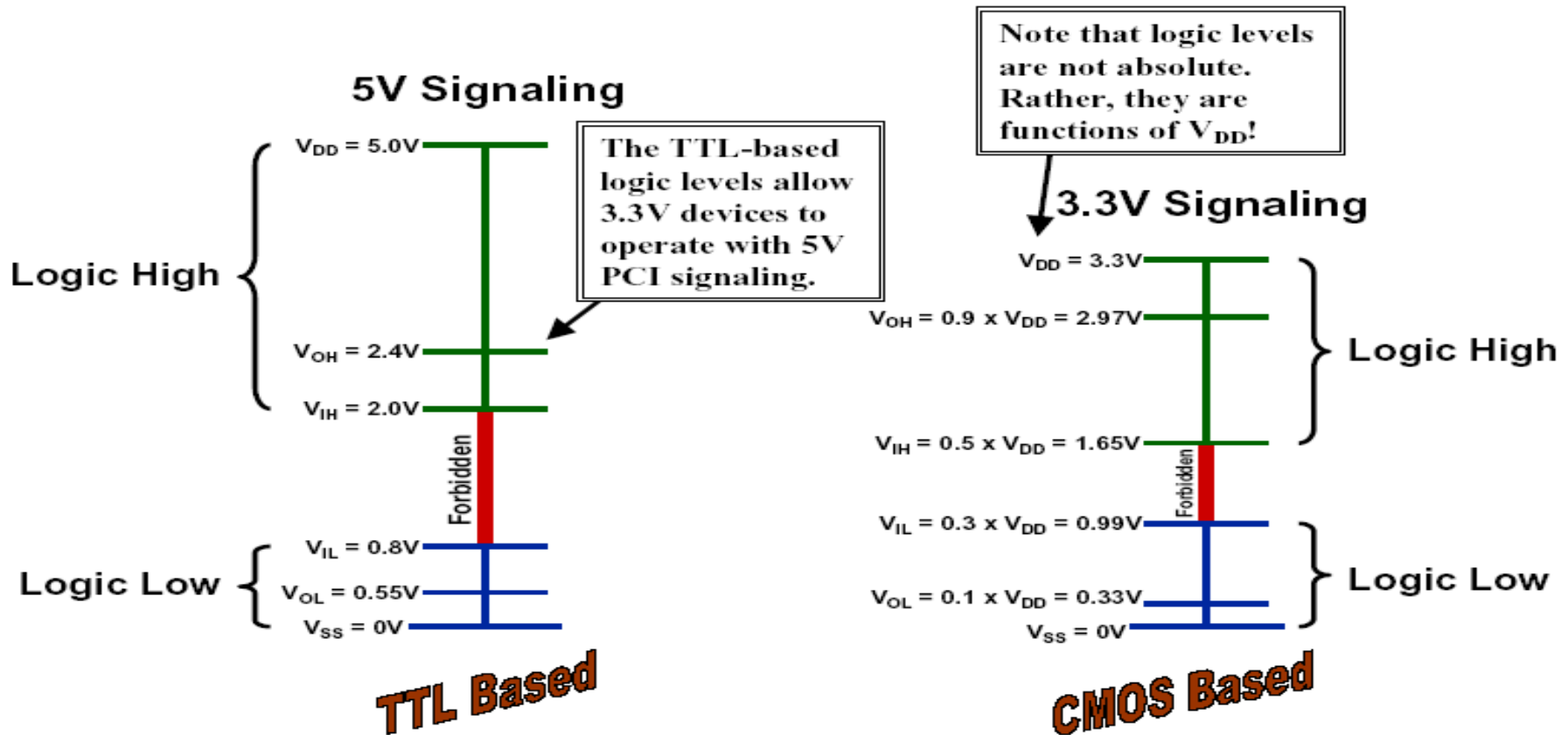
- ◆ **PCI připouští dvě napět'ové úrovně signálů**
  - ◆ *5V úroveň*
  - ◆ *3.3V úroveň*
- ◆ **Technicky nemají tato jména nic společného s aktuálním napájecím napětím**
  - ◆ *Jsou vázány na rozhodovací log. úrovně*

# Signálové úrovně

- ◆ **5V úrovně jsou nejběžnější**
- ◆ **66MHz sběrnice PCI mohou používat jen 3.3V úroveň**
  - ◆ *Poznámka - 33MHz PCI může používat obě*
- ◆ **Některé desky mohou podporovat obě prostředí současně (5V, 3.3V) – nazývají se “univerzální” desky**

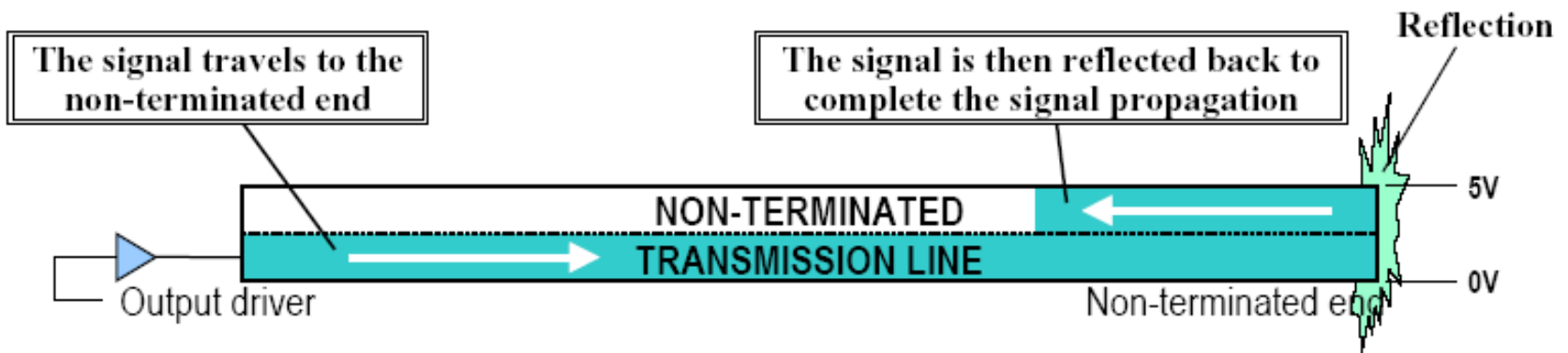
# Signálové úrovně

## Signaling Environments



# Využití odrazů na vedení

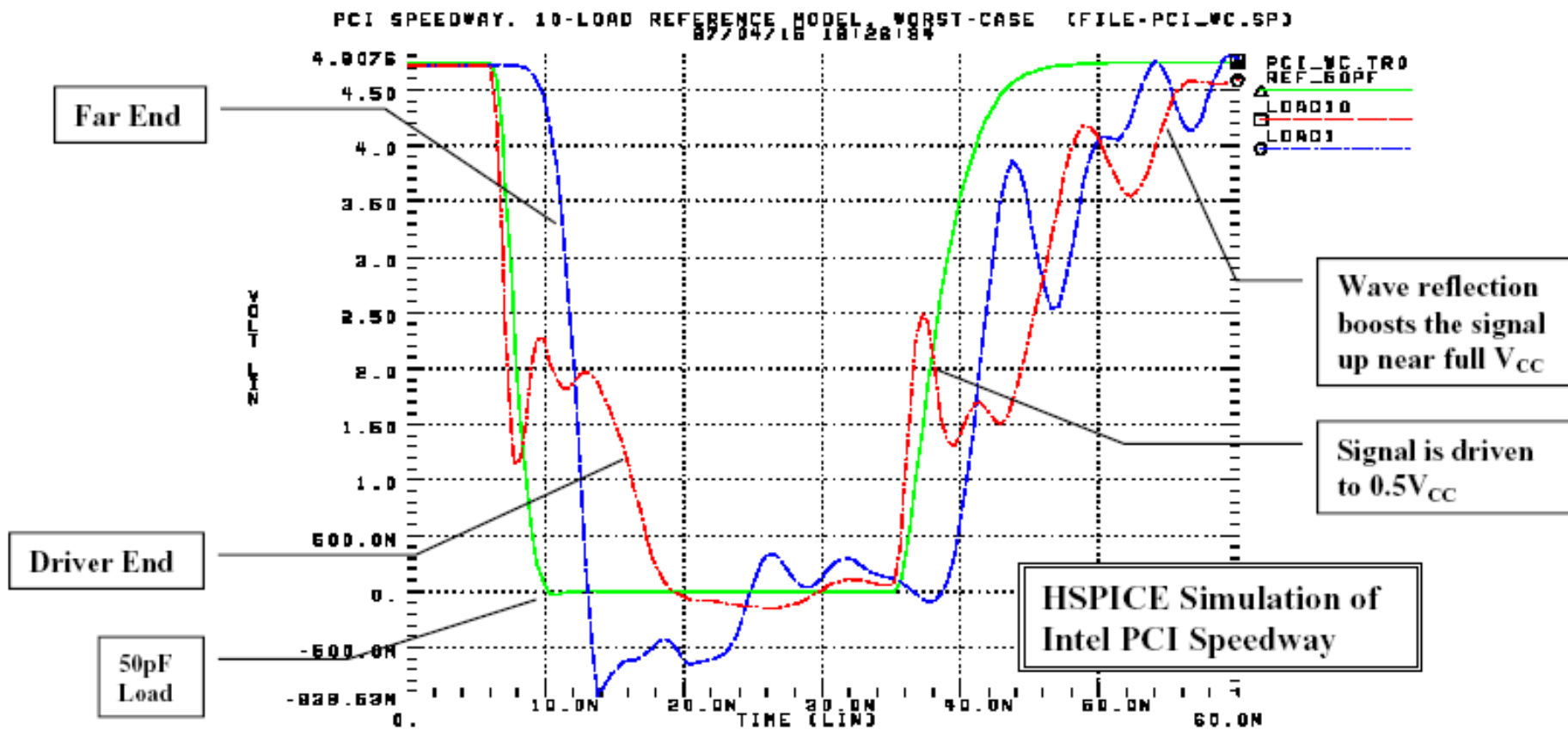
- ◆ **PCI využívá odrazů na koncích signálových linek**
  - ◆ *Každý vodič PCI je nezakončená přenosová linka, jejíž signál se po příchodu na konec odrazí*
  - ◆ *Platné napěťové úrovně jsou získány po jednom odrazu; to snižuje cenu PCI, protože se nemusí používat vysokovýkonné výstupní budiče*





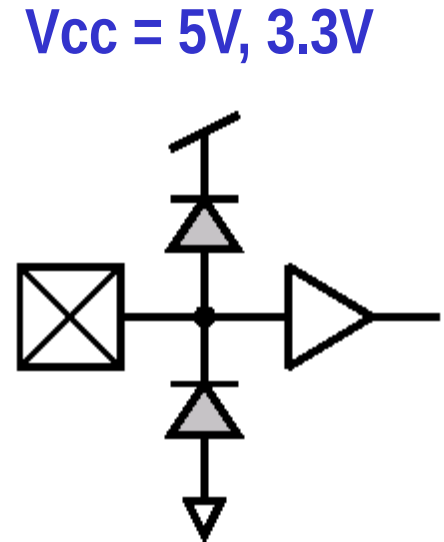
# Průběhy signálů

- ◆ This display shows how a reflection looks

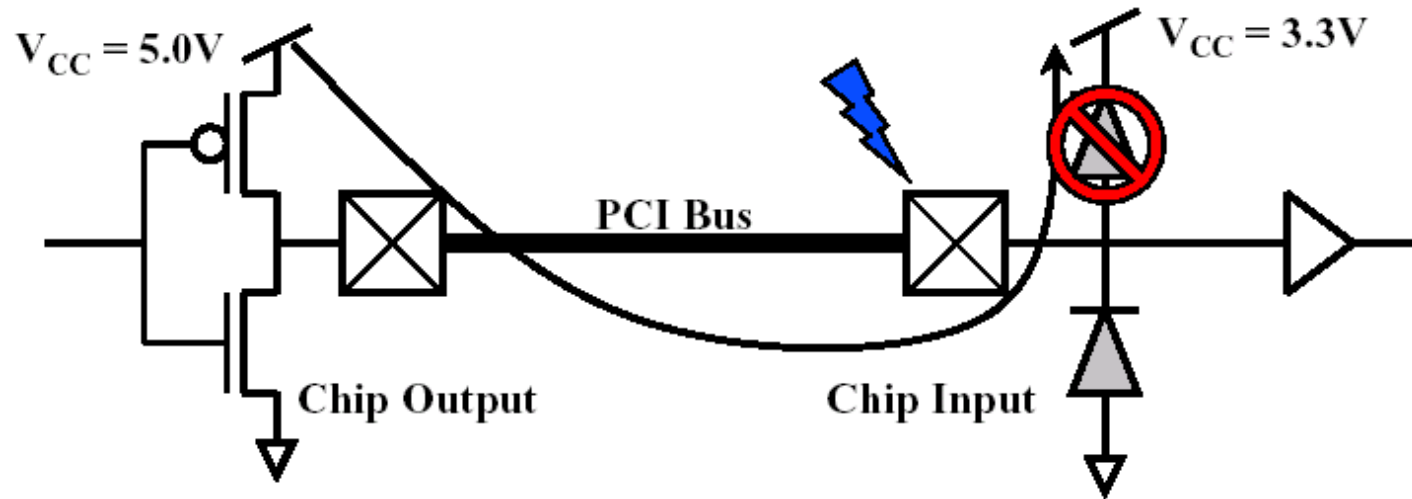


# Elektrická specifikace PCI - vstupy

- ◆ **Ochranné diody chrání vstupy před zkratovými proudy, které vznikají v důsledku zpožděného přechodu výstupů do vysokoimpedančního stavu**
- ◆ **Signálové úrovně 5V**
  - ◆ *Vstupy se „opírají“ o zem*
  - ◆ *Horní dioda „opírající“ se o 5V je nepovinná*
- ◆ **Signálové úrovně 3.3V**
  - ◆ *Vstupy musí být „opřeny“ o zem i o Vcc*
  - ◆ *Vstupy univerzálních desek (nebo jiných 3.3V zařízení), která jsou připojena k 5V PCI, musí být „opřeny“ o napájení 5V, nikoliv o 3.3V*



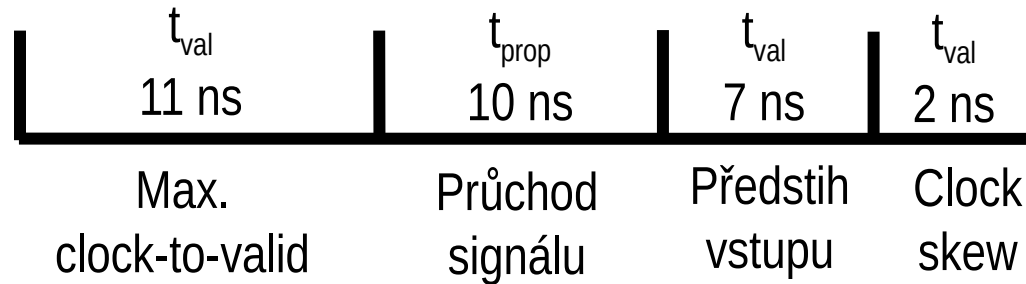
# Špatně provedená ochrana



- ◆ Univerzální deska, jejíž diody jsou „opřeny“ o 3.3V a je zasunutá do PCI s 5V úrovní signálů, způsobuje zkrat mezi napájecím zdrojem 5V a 3.3V základní desky !

# Časová specifikace PCI 33MHz

**\* Cykl sběrnice 30 ns**



$$\Sigma = 30 \text{ ns}$$

## Ostatní požadavky:

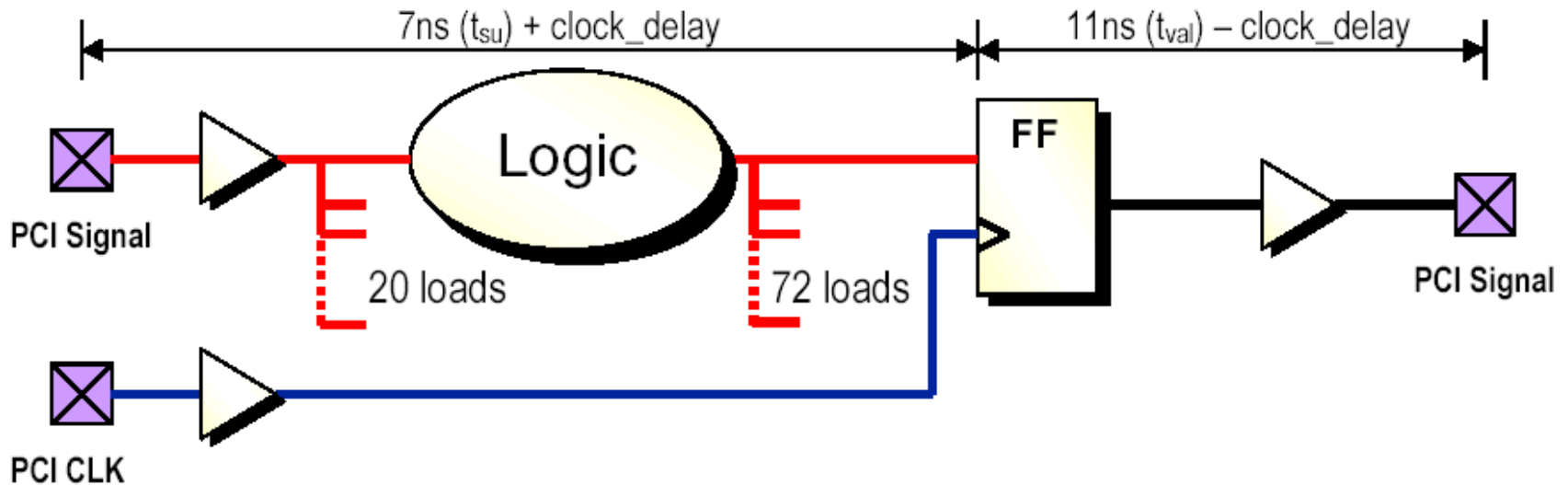
Hold time : 0 ns

Min clock-to-out : 2 ns

Output off time : 28 ns

Všechny časové parametry jsou měřeny na pinech příslušného pouzdra

# Proč je časování PCI tak přísné



- ◆ PCI „handshaking“ se provádí v každém hodinovém cyklu

- ◆ *Na tyto signály nelze uplatnit pipelinning*

**7ns setup + clock\_delay → 100 + MHz! [33 MHz PCI]**

**3ns setup + clock\_delay → 220 + MHz! [66 MHz PCI]**

# Návrh desky agenta

## ◆ Délka signálových tras

- ◆ *Všechny signály 32-bitové PCI sběrnice nesmí přesáhnout délku 1.5 "*

## ◆ Délka vedení hodin musí být přesně 2.5" ( $\pm 0.1$ " )

- ◆ *Vedeny pouze k jedné zátěži*
- ◆ *Nutné pro řízení clock-skew*

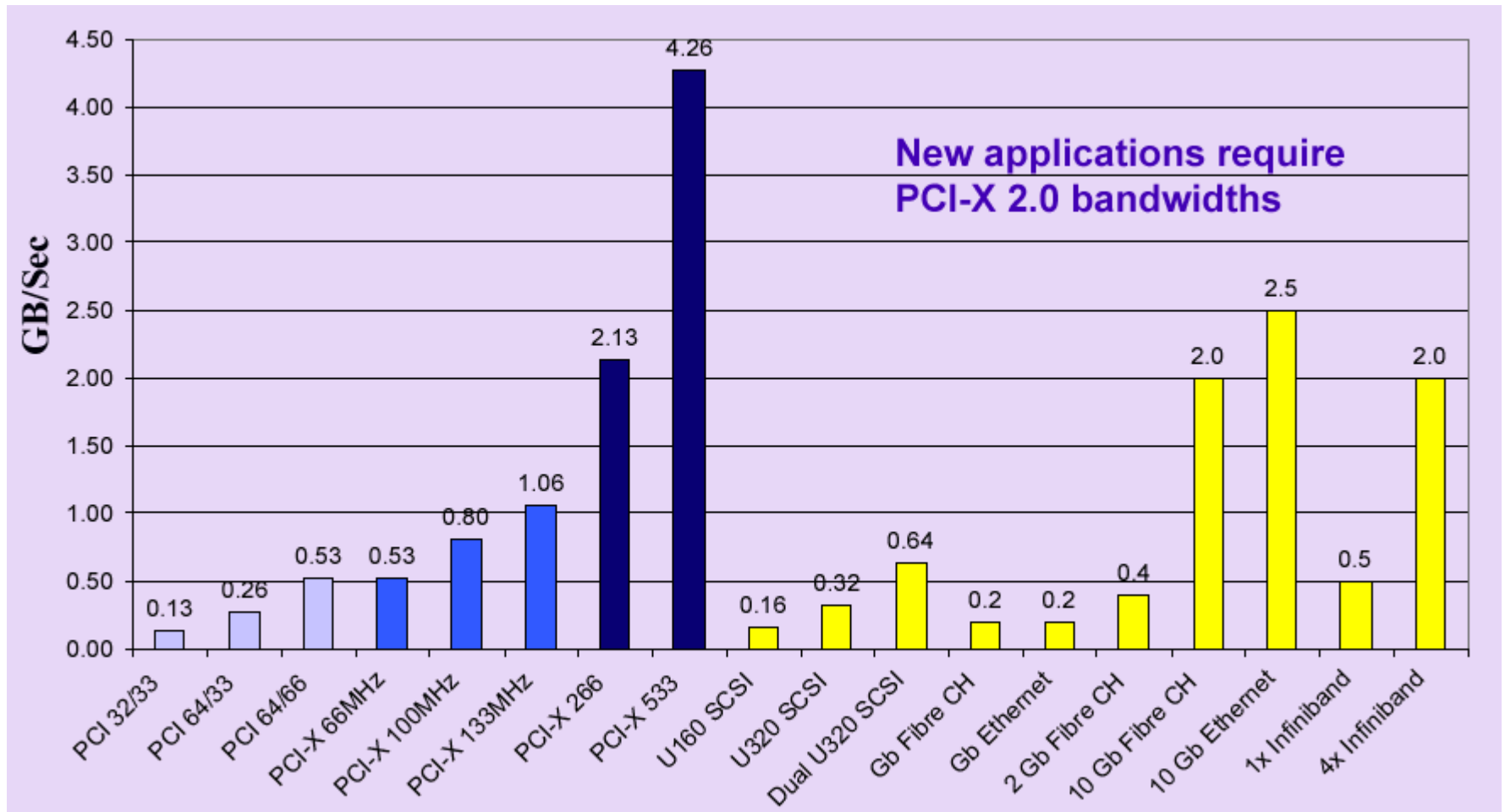
## ◆ Požadavky na PCI zařízení

- ◆ *Jeden signál - jeden pin !*
- ◆ *Maximální vstupní kapacita činí 10pF (je-li zařízení na základní desce, připouští se 16pF)*

# Buzení a zatížení sběrnice

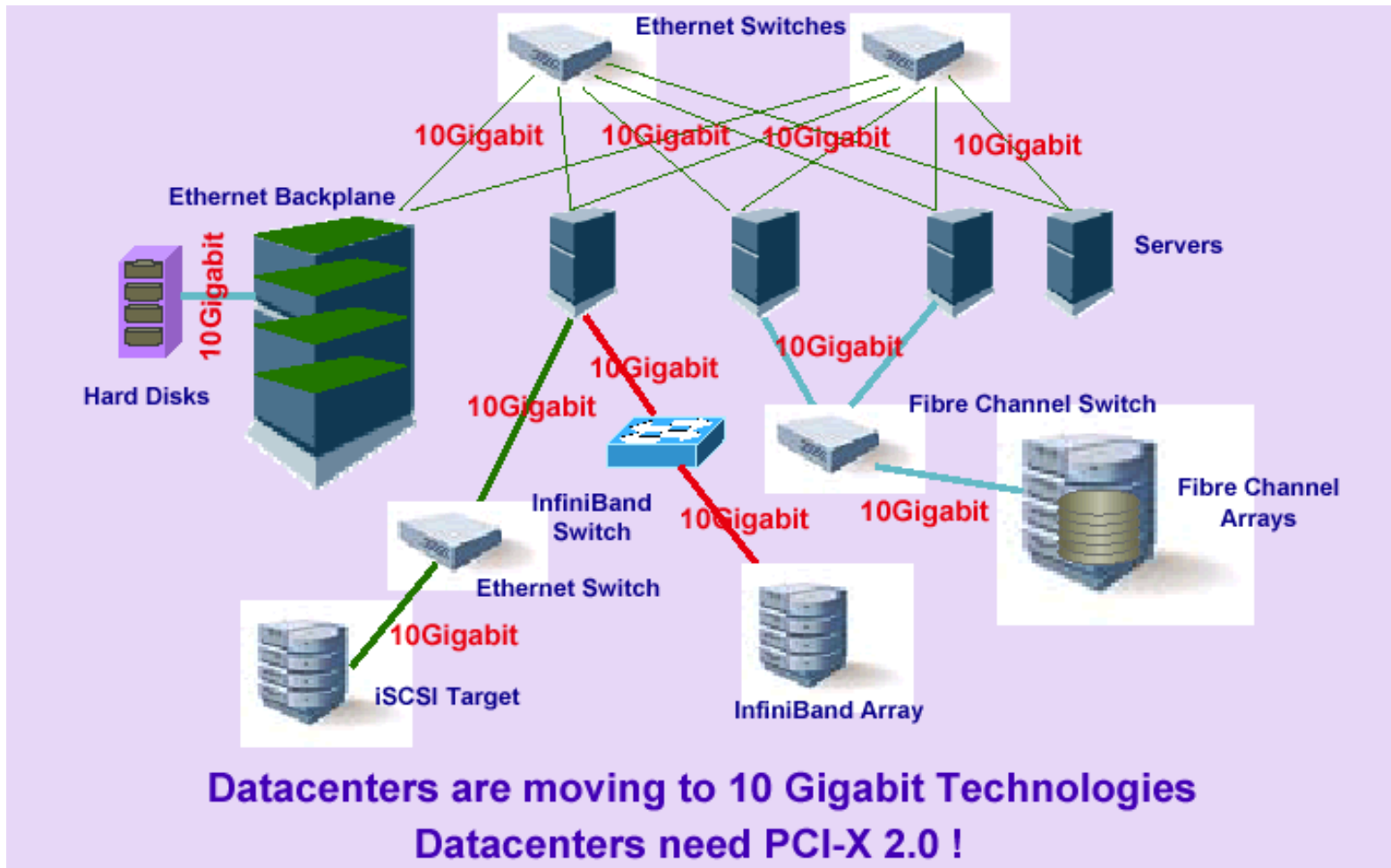
- ◆ **K zatížení PCI se nevyjadřuje žádná specifikace, ale:**
  - ◆ *Budič musí vyhovět době zpoždění 10ns*
- ◆ **Pravidlo je max. 10 zátěží pro 33 MHz**
  - ◆ *Zařízení na základní desce se počítají jako jedna zátěž*
  - ◆ *Každá vložená deska se počítá jako dvě zátěže*
  - ◆ *Protože většina základních desek PC musí mít více než 2 PCI zařízení, nemají tyto obvykle více než 4 konektory*
- ◆ **Více konektorů je k dispozici při použití PCI-to-PCI bridge nebo u systémů peer to-peer PCI**

# Šířka pásma jednotlivých typů sběrníc



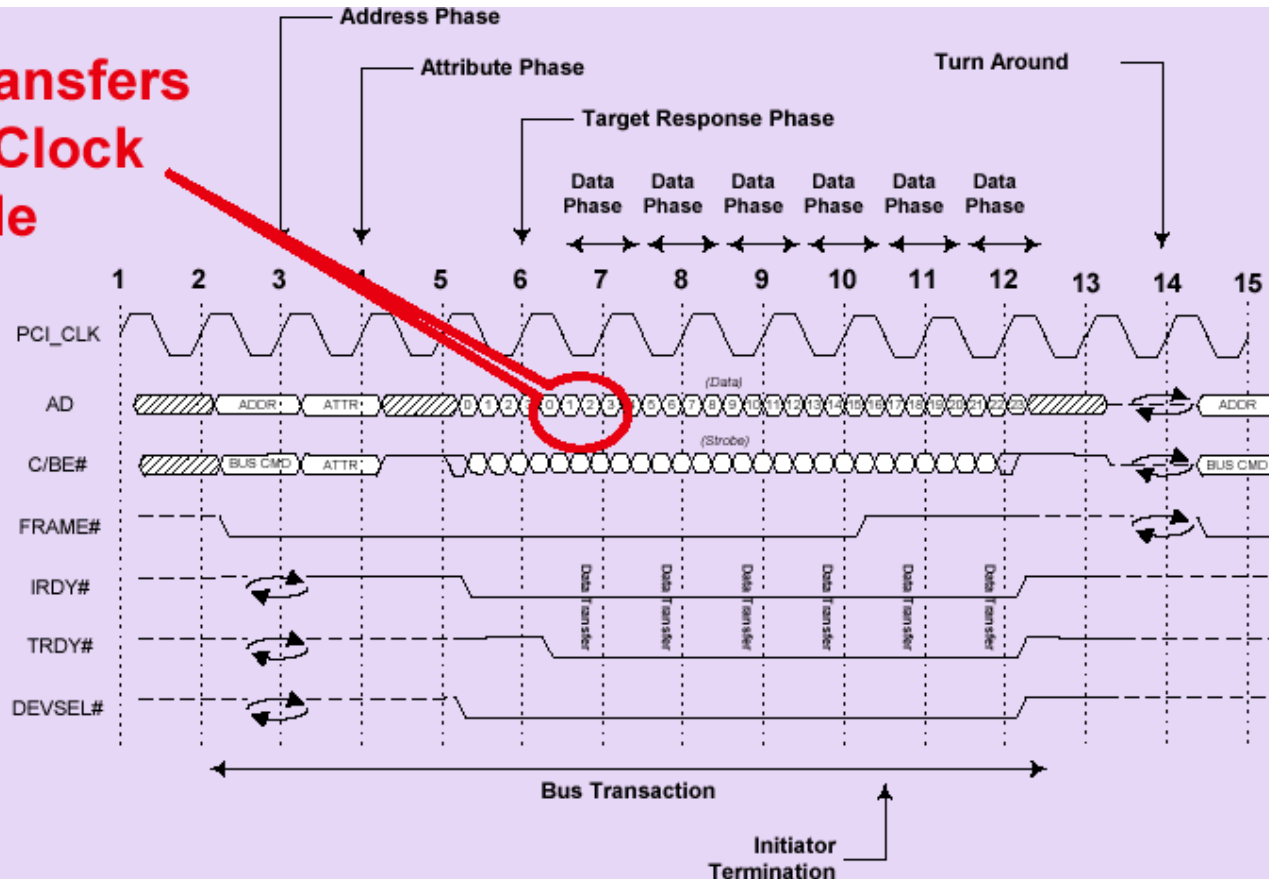


# Proč PCI-X 2.0?



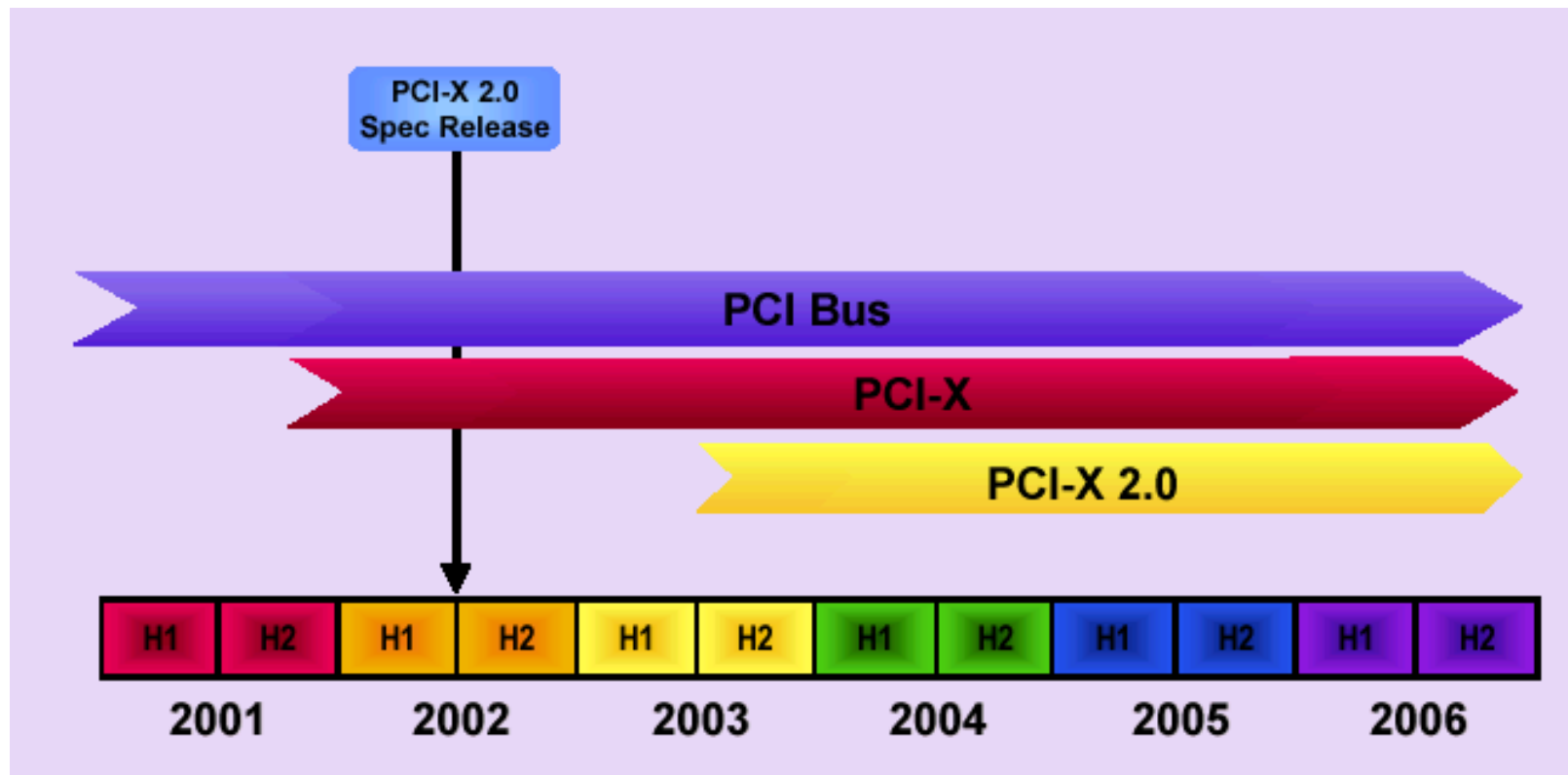
# Příklad zápisu na PCI-X 2.0

**4 Transfers  
per Clock  
Cycle**



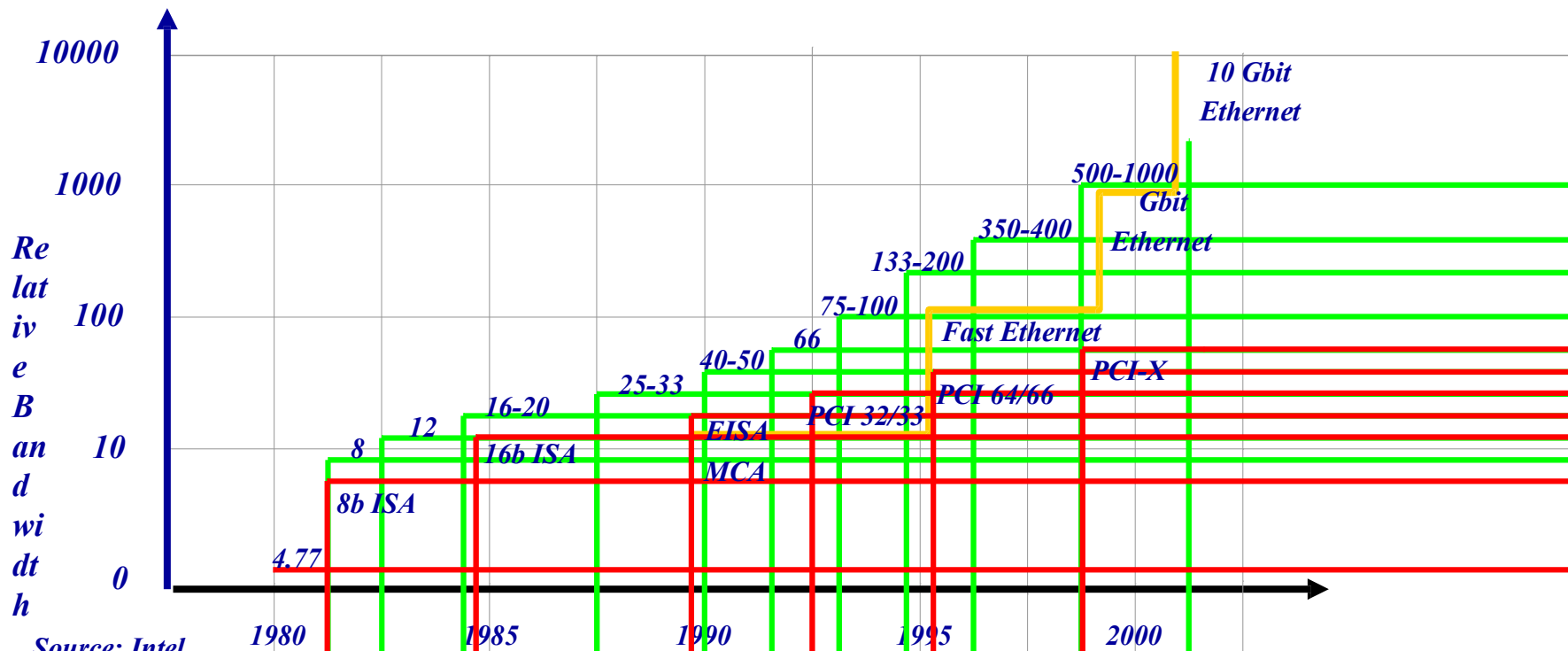
**PCI-X 533 achieves high data rates by transferring 4 QWORDS of data per clock cycle.**

# Časový průběh vývoje

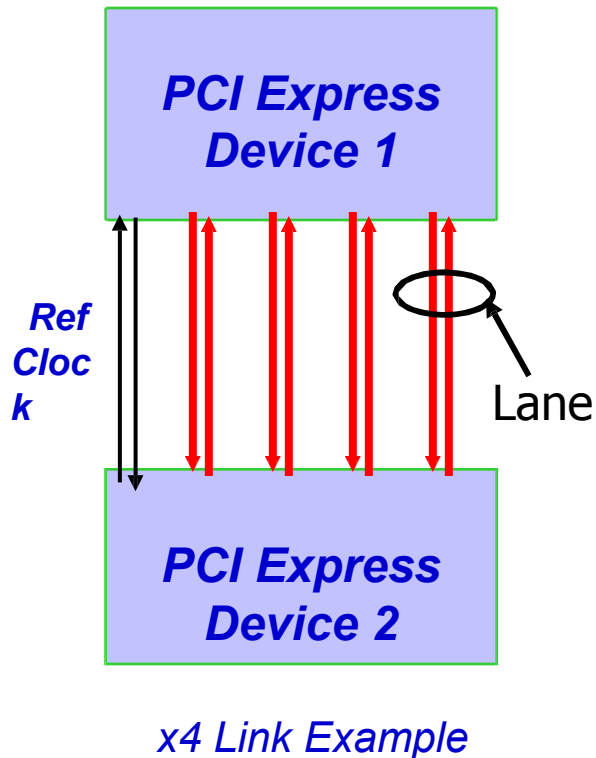


# Vývoj požadavků na přenosové rychlosti

- ❑ Multimediální aplikace vyžadují rychlé efektivní zpracování dat po metalických i bezdrátových spojích
- ❑ Výkon CPU se zdvojnásobuje každých 18 měsíců, zatímco výkon PC sběrnice se zdvojnásobuje jednou za 3 roky



# Základy sběrnice PCI Express



- ❑ **Seriová, point-to-point, Low Voltage Differential Signaling**
- ❑ **2.5GHz full duplex lanes (2.5Gb/s)**
  - ❖ PCIe Gen 2 = 5Gb/s
- ❑ **Scaleable links – x1, x4, x8, x16**
- ❑ **Packet based transaction protocol**
- ❑ **Software compatible but with higher speeds**
- ❑ **Built-in Quality of Service provisions**
  - ❖ Virtual Channels
  - ❖ Traffic Classes
- ❑ **Reliability, Availability and Serviceability**
  - ❖ End-to-End CRC (Cyclic redundant checking)
  - ❖ Poison Packet
  - ❖ Native Hot Plug support
- ❑ **Flow Control**
- ❑ **Advance error reporting**

# Výkon sběrnice PCI Express

Link Width	X1	X2	X4	X8	X12	X16	x32
Bandwidth in Gbits/s (Tx and Rx)	5	10	20	40	60	80	160
Throughput in GB/s (Tx and Rx)	.5	1	2	4	6	8	16
Throughput in GB/s (per direction)	.25	.5	1	2	3	4	8

***Hrubý výkon: Předpokládáme  
100% efektivitu bez ztrát.***

 = PCI 32/66

 = PCI or PCI-X 64/66

 = PCI-X 64/133

# Vrstvy PCIe

- Layered architecture
- Application Data transferred via packets
  - ❖ Transaction Layer Packet (TLP)
- PCIe core usually implement the lower three layers
  - ❖ connection establishing
  - ❖ link control
  - ❖ flow control
  - ❖ power management
  - ❖ error detection and reporting

