Architektura sběrnic PCI, PCI-X a PCI Express

Tomáš Martínek
martinto@fit.vutbr.cz

Obsah

- Úvod
- PCI
- PCI-X
- PCI Express

Vývoj sběrnic typu PCI

- Sběrnice vyvíjena sdružením *PCI-SIG* (***P**eripheral **C**omponent **I**nterconnect **S**pecial **I**nterest **G**roup*)
- Historie vývoje
  - PCI – paralelní sběrnice (dnes už se přestává používat)
  - PCI-X – podobně jako PCI, vyšší výkon i efektivita
  - PCI Express – vysokorychlostní plně duplexní sériové linky, přenos na základě paketové komunikace

Typ	Specifikace	Datum
PCI 33 MHz	2.0	1993
PCI 66 MHz	2.1	1995
PCI-X 66 MHz a 133 MHz	1.0	1999
PCI-X 266 MHz a 533 MHz	2.0	2002
PCI Express (2.5Gb)	1.1	2002
PCI Express (5Gb)	2.0	2007

Porovnání výkonnosti

- **PCI a PCI-X**

Typ	Frekvence	Max. propustnost	Počet slotů na sběrnici
PCI 32-bit	33 MHz	133 MB/s	4-5
PCI 32-bit	66 MHz	266 MB/s	1-2
PCI-X 32-bit	66 MHz	266 MB/s	4
PCI-X 32-bit	133 MHz	533 MB/s	1-2
PCI-X 32-bit	266 MHz	1066 MB/s	1
PCI-X 32-bit	533 MHz	2131 MB/s	1

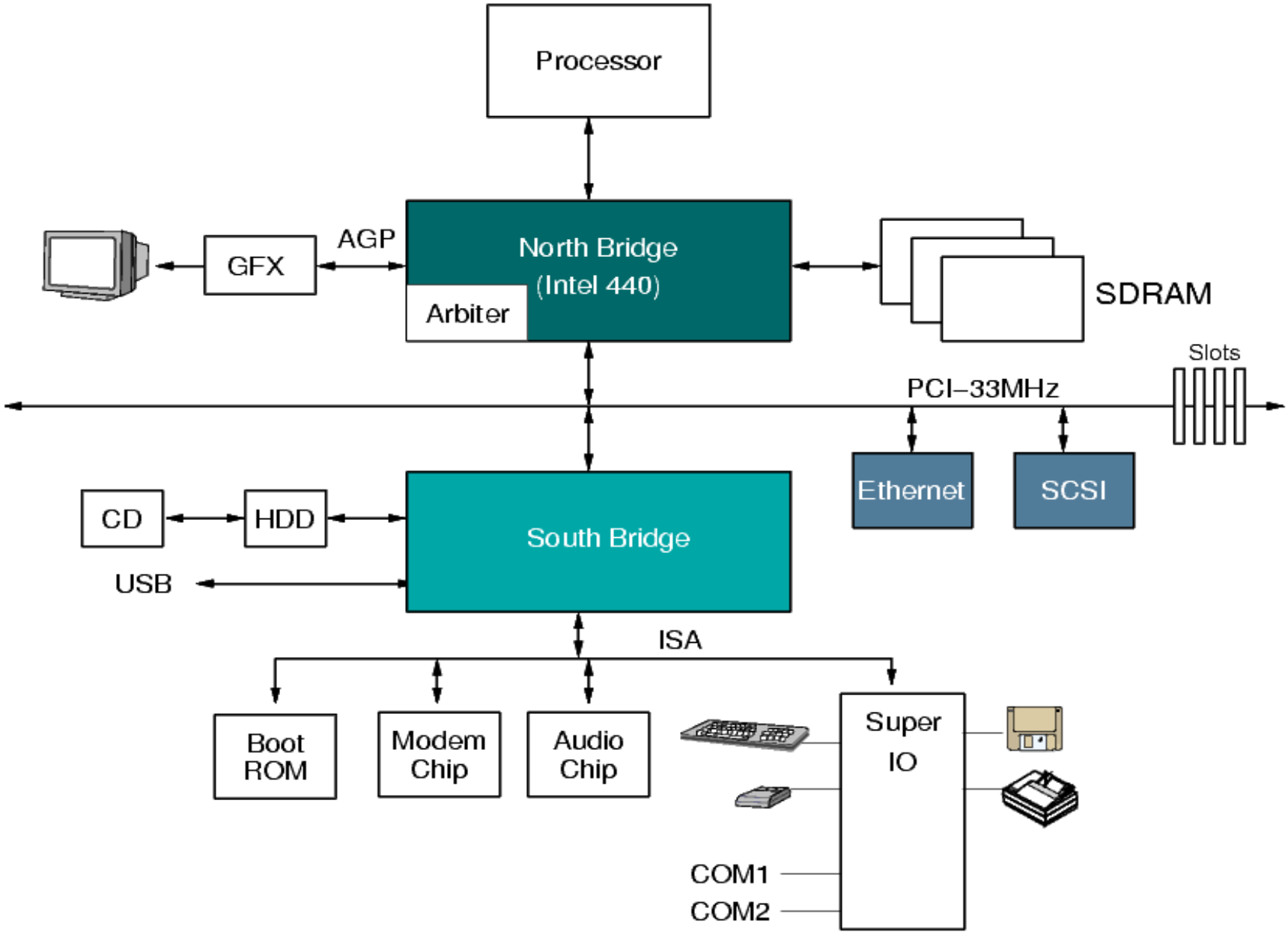
  - pro sběrnice šířky 64-bitů je výkonnost dvojnásobná

- **PCI Express**

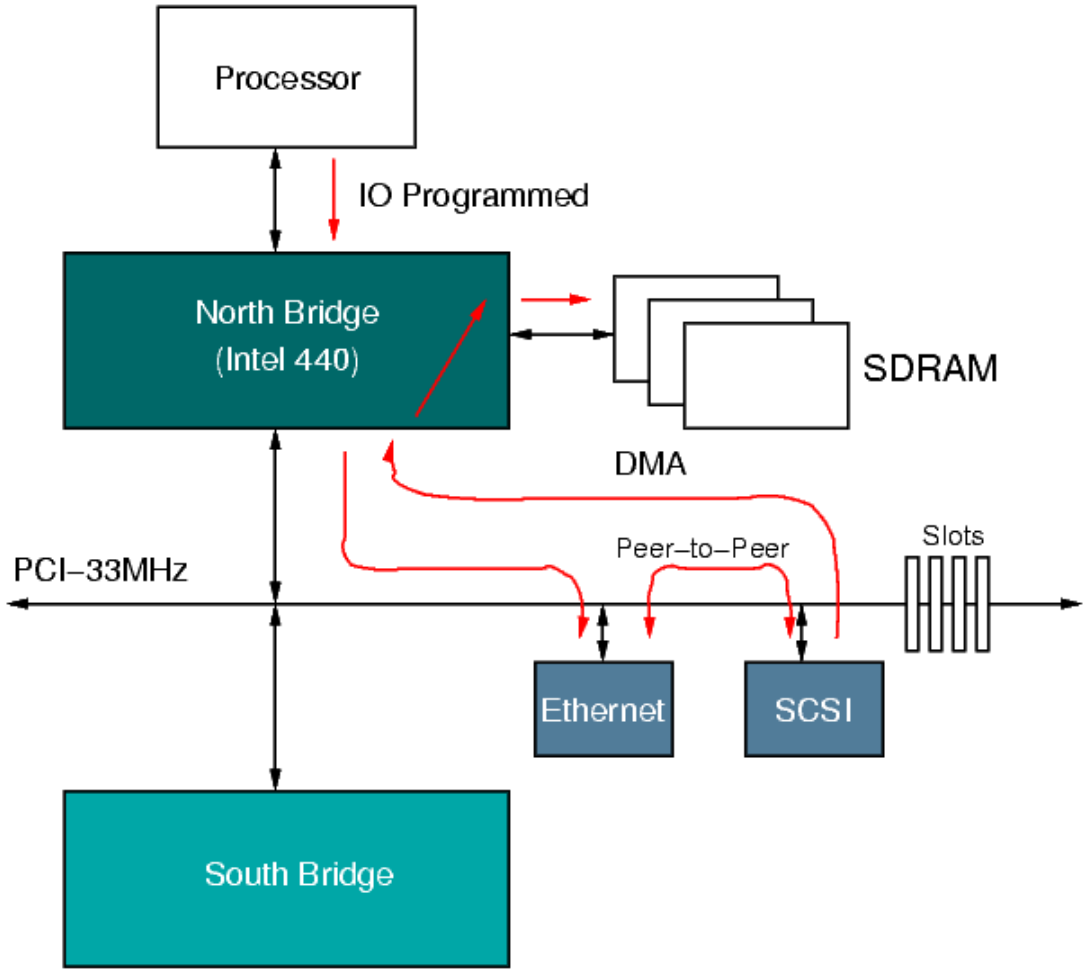
Počet linek	x1	x2	x4	x8	x12	x16	x32
Propustnost GB/s	0.5	1	2	4	6	8	16

  - plně duplexní sériové linky (každá linka 2.5Gb/s oběma směry)
  - potřeba zahrnout vliv kódování 8/10 (20% režie)

Platforma založená na PCI



Modely komunikace na PCI

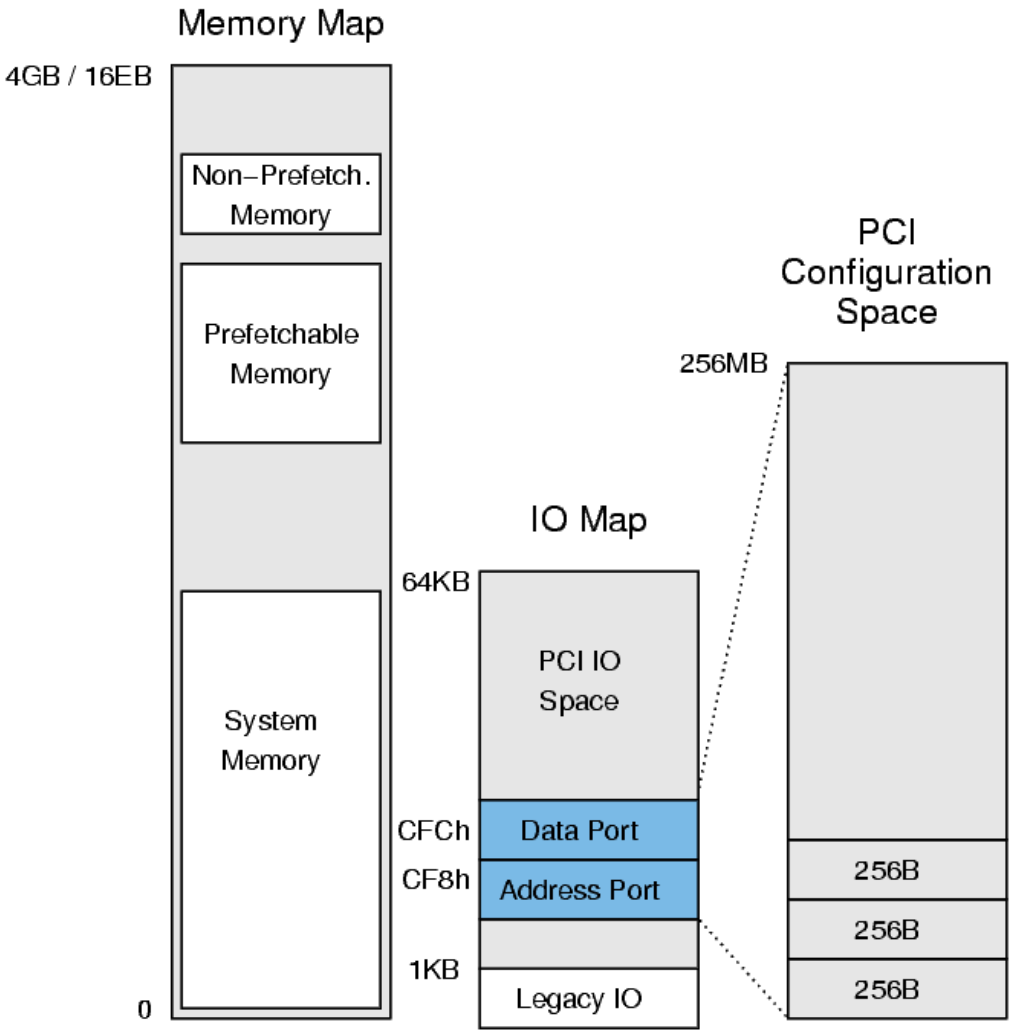


- Typy operací (vybírá signál *C/BE*)
  - Přístup do paměťového prostoru
  - Přístup do prostoru I/O
  - Konfigurační cyklus

Příklady komunikací:

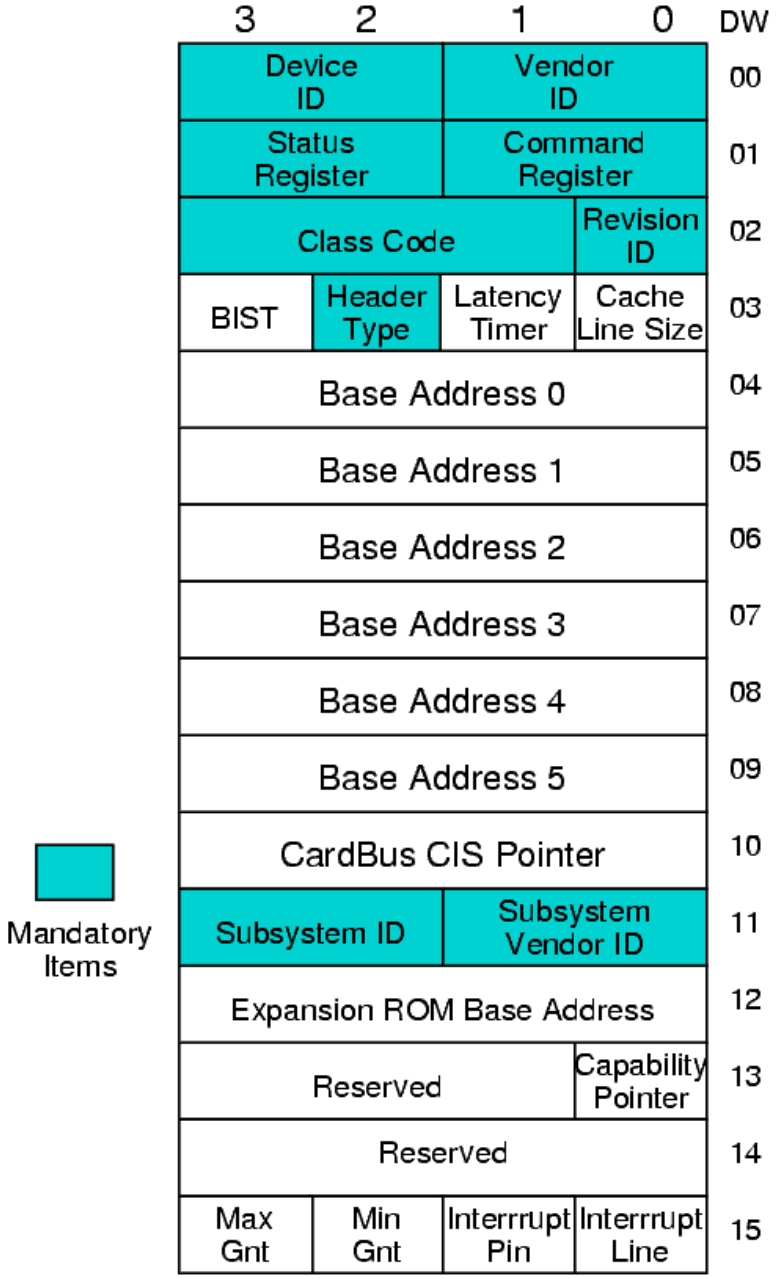
1. Přímý přístup do paměti bez účasti procesoru (DMA): a) zařízení vystaví požadavek přístupu do paměti, b) North Bridge potvrdí příjem operace a c) zprostředkuje přenos dat s pamětí
2. Peer-to-peer – přenos dat mezi dvěma PCI zařízeními: a) Initiator vystaví adresu, b) Target potvrdí transakci
3. IO přístup (do registrů PCI zařízení): a) CPU inicializuje IO operaci na základě sw. instrukcí *IN*/*OUT*, b) North Bridge převezme požadavek a generuje IO transakci na PCI

Struktura paměťového prostoru



- PCI podporuje tři typy paměti:
  - *Paměť*
  - *IO prostor*
  - *Konfigurační prostor*
- Paměť se dále rozlišuje na:
  - *Prefetchable memory* – paměť lze předčítat do cache paměti
  - *Non-prefetchable memory* – do paměti se musí vždy přistupovat přímo
- Konfigurační prostor obsahuje klíčové informace o každém PCI zařízení. Rozděleno do speciálních datových struktur o velikosti 256B. Dva typy:
  - *Typ 0* – pro koncové body
  - *Typ 1* – pro bridge nebo switch
- Kompatibilní v PCI-X a PCI-Express

Konfigurace typu 0

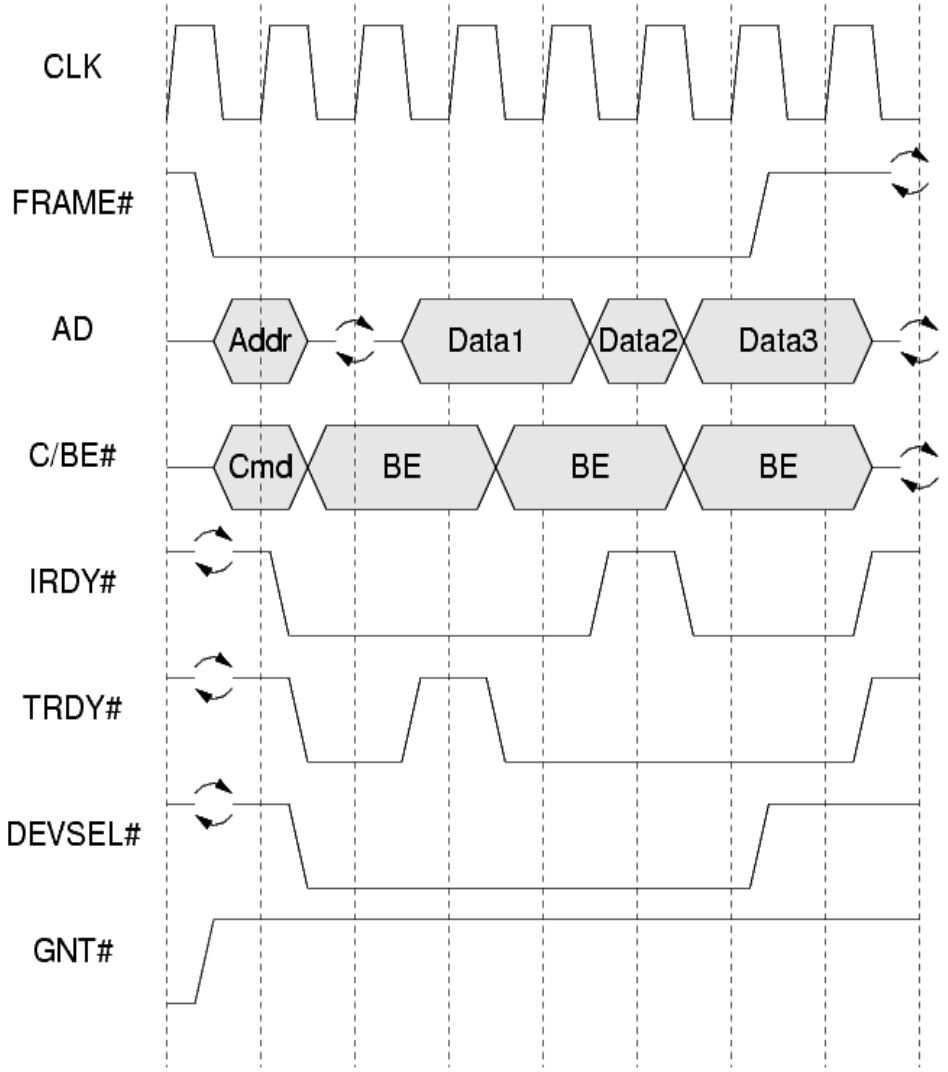


- *Vendor ID* – identifikační číslo výrobce. Přiřazováno centrální autoritou PCISIG
- *Device ID* – identifikace PCI zařízení. Určuje výrobce.
- *Revision ID* – číslo revize. Určuje výrobce.
- *Class Code* – určuje obecný typ zařízení např. multimediální, síťové, wifi zařízení, apod.
- *Subsystem Vendor ID* a *Subsystem ID* – identifikace možného podsystému v rámci daného PCI zařízení.
- *Base Addess (BAR)* – bázová adresa zařízení. Pomocí tohoto registru probíhá alokace paměťového prostoru pro PCI zařízení. Celkem je možné alokovat až 6 bloků paměti. Pokud je použit 64-bitový adresový prostor, použijí se dvě položky.

Obsah

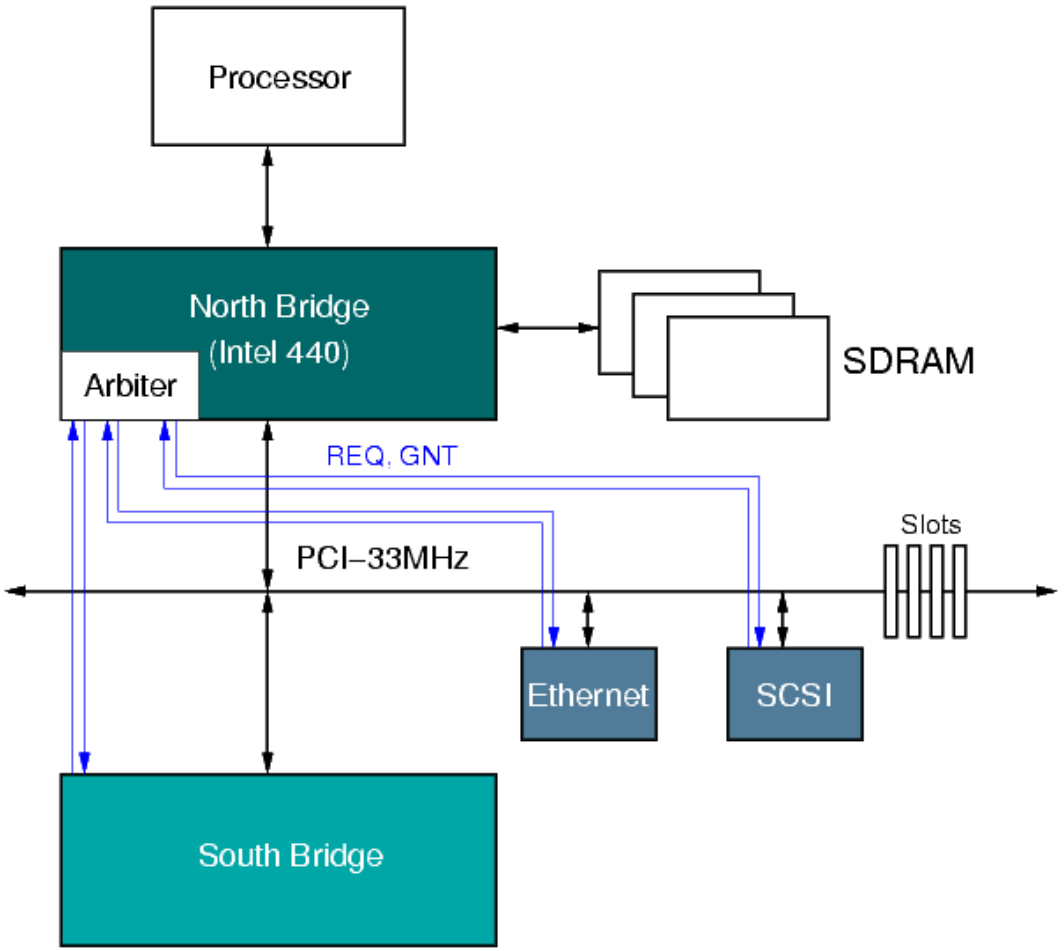
- Úvod
- PCI
- PCI-X
- PCI Express

Komunikační protokol



- Každá operace má dva účastníky
  - *Initiator* – zařízení zakládající operaci
  - *Target* – cílové zařízení
- Na začátku transakce nejprve vystavena adresa, která je pak multiplexovaná s daty => výrazná redukce počtu vodičů
- Zařízení, které pozná svou adresu, potvrdí transakci signálem *DEVSEL*
- Čekací stavy může vkládat Initiator i Target. Jejich připravenost je signalizována signály *TRDY* a *IRDY*
- Obecně blokový přenos dat (Burst) => vede na větší efektivitu. Target nikdy neví kolik dat se přenáší !!
- V případě přerušení transakce, mají oba kontrolu nad stavem sběrnice!

Způsob arbitrace

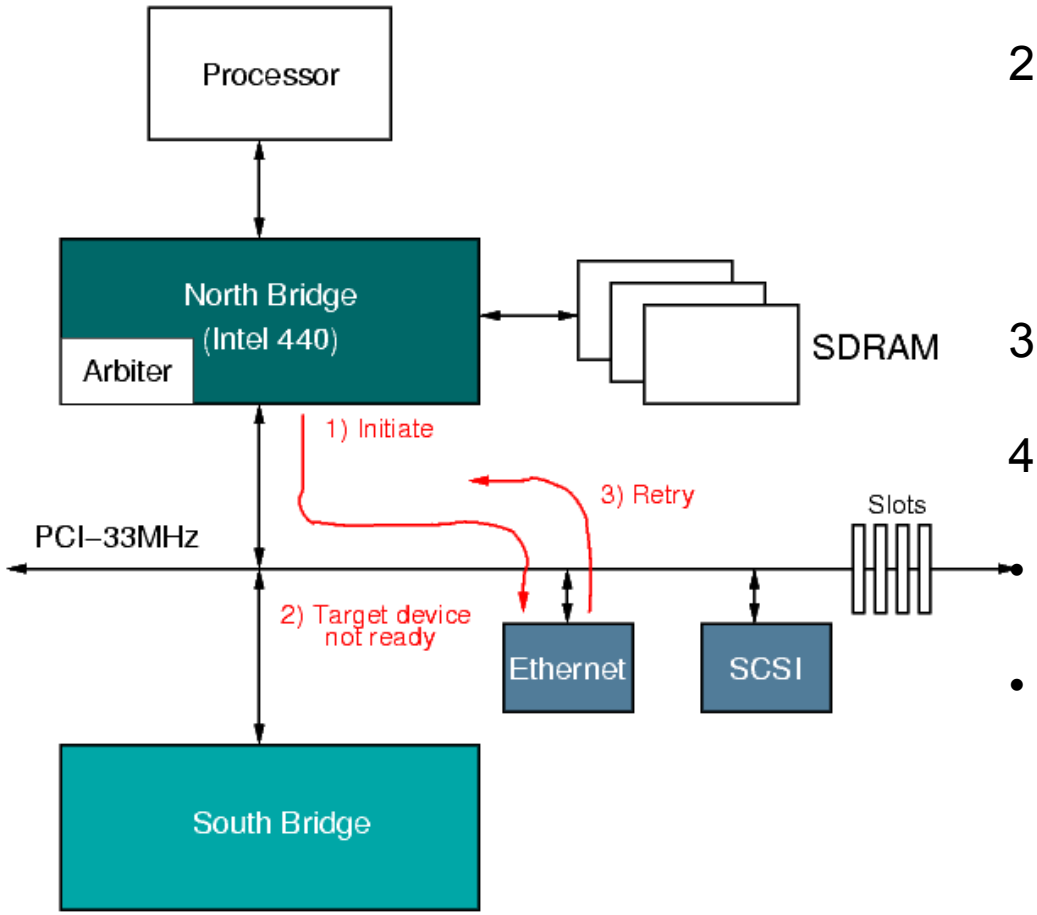


- Před každou transakcí musí Initiátor požádat o sběrnici signálem *REQ*

  O přidělení sběrnice rozhoduje Arbitr umístěný v North Bridge a sběrnici přiděluje nastavením signálu *GNT*

  Každé zařízení má vlastní signál *REQ, GNT* => arbitrace může probíhat na pozadí aktuální operace => vede na efektivnější využití sběrnice. Jedná se o tzv. *„skrytou arbitraci“*

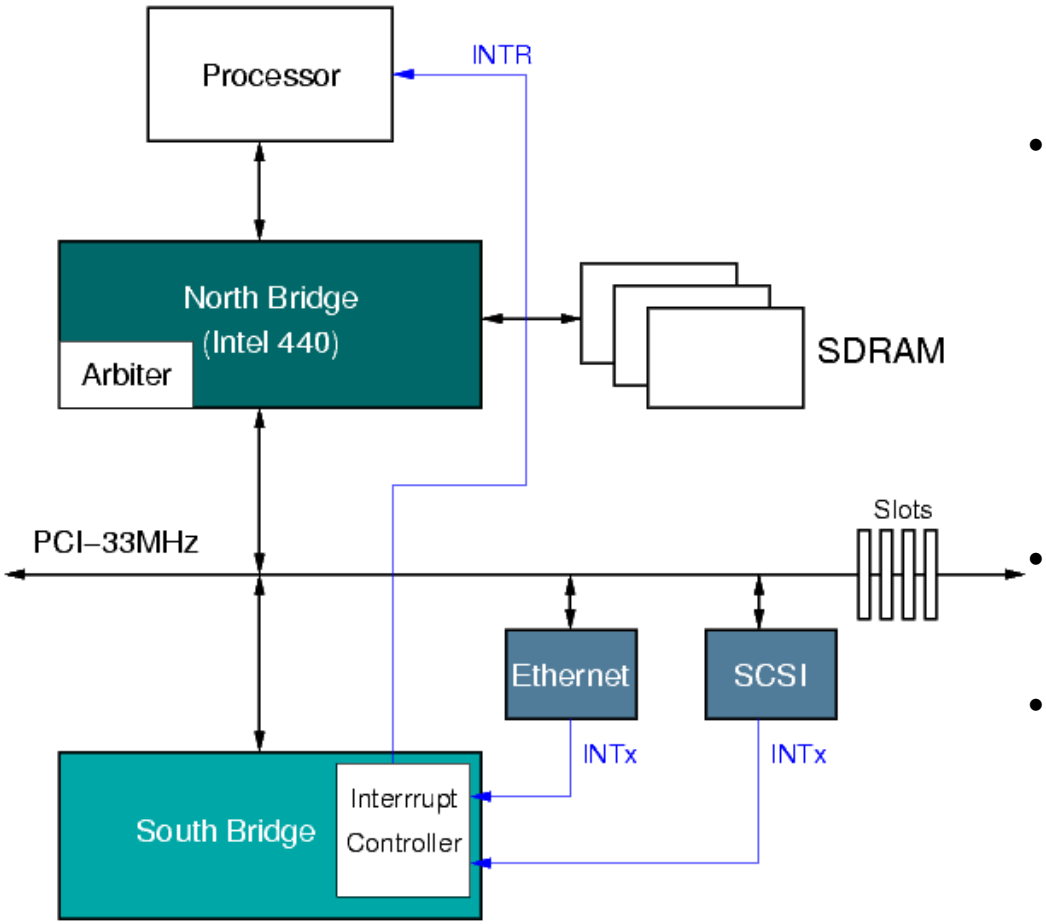
PCI Retry/Disconnect Protokol



1. Initiator inicializuje čtecí operaci směrem k Target zařízení
2. Target rozpozná adresu a potvrdí příjem operace, ale zjistí, že nemá připravena data. (Pokud je schopen data připravit rychle může vložit až 16 čekacích stavů, jinak viz bod 3.)
3. Target odloží operaci příkazem *Retry* a začne si připravovat data
4. Initiator zkusí znovu založit operaci později

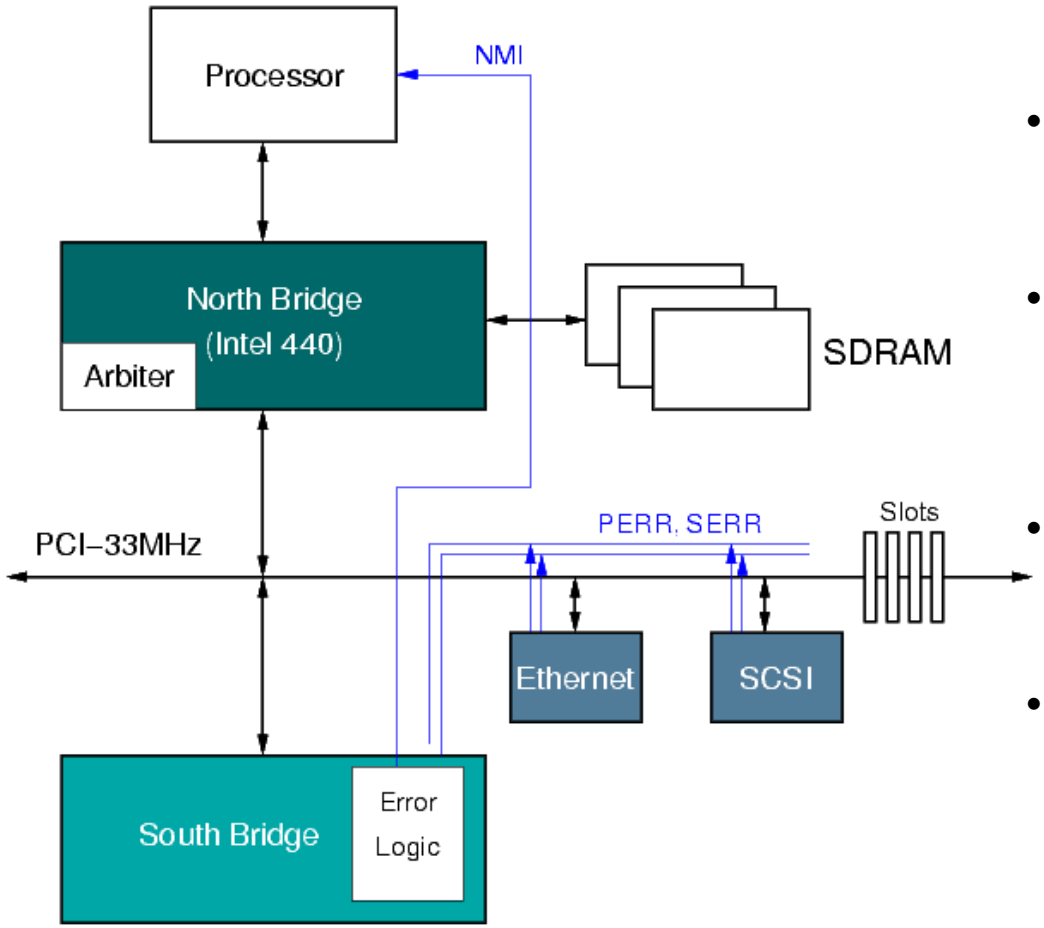
- Cyklus se opakuje dokud nezíská Initiátor požadovaná data
- Podobně, pokud je již část dat přenesena a target v průběhu zjistí, že další nemá k dispozici
  - Target přeruší spojení pomocí příkazu *Disconnect*
  - Initiator zkusí znovu založit operaci později od pozice, kde došlo k rozpojení transakce
- Neefektivní v případě, že dochází k vícenásobnému odložení transakce!

Obsluha přerušení



- PCI zařízení může vyvolat přerušení pomocí signálů *INTA, INTB, INTC, INTD*
- Řadič přerušení umístěný v South Bridge vyhodnotí prioritu a zašle procesoru signál *INTR*. Pokud architektura podporuje APIC (Advanced Programmable Interrupt Controller) začle procesoru místo INTR přímo zprávu
- Procesor přeruší svoji činost a provede obsluhu přerušení
- Signály *INTA-D* jsou sdílené pro všechny PCI zařízení, proto musí OS driver dodatečně prohledat, které zařízení přerušení vyvolalo => neefektivní !!

Obsluha chyb



- V průběhu každé transakce jednotlivé PCI zařízení kontrolují paritu vystavované adresy a dat
- V případě, že je detekována chybná parita, zařízení nastaví signály *PERR* (Parity Error) a *SERR* (System Error)
- V South Bridge je umístěna logika, která vyhodnocuje tyto stavy a generuje nemaskovatelné přerušení do procesoru signálem *NMI*
- Při chybě parity dochází k pádu systému (pokud není tato kontrola explicitně vypnuta)
- Kontrola parity je obecně nedostačující kontrolou (detekuje pouze lichý počet chyb) !!

Shrnutí nevýhod PCI

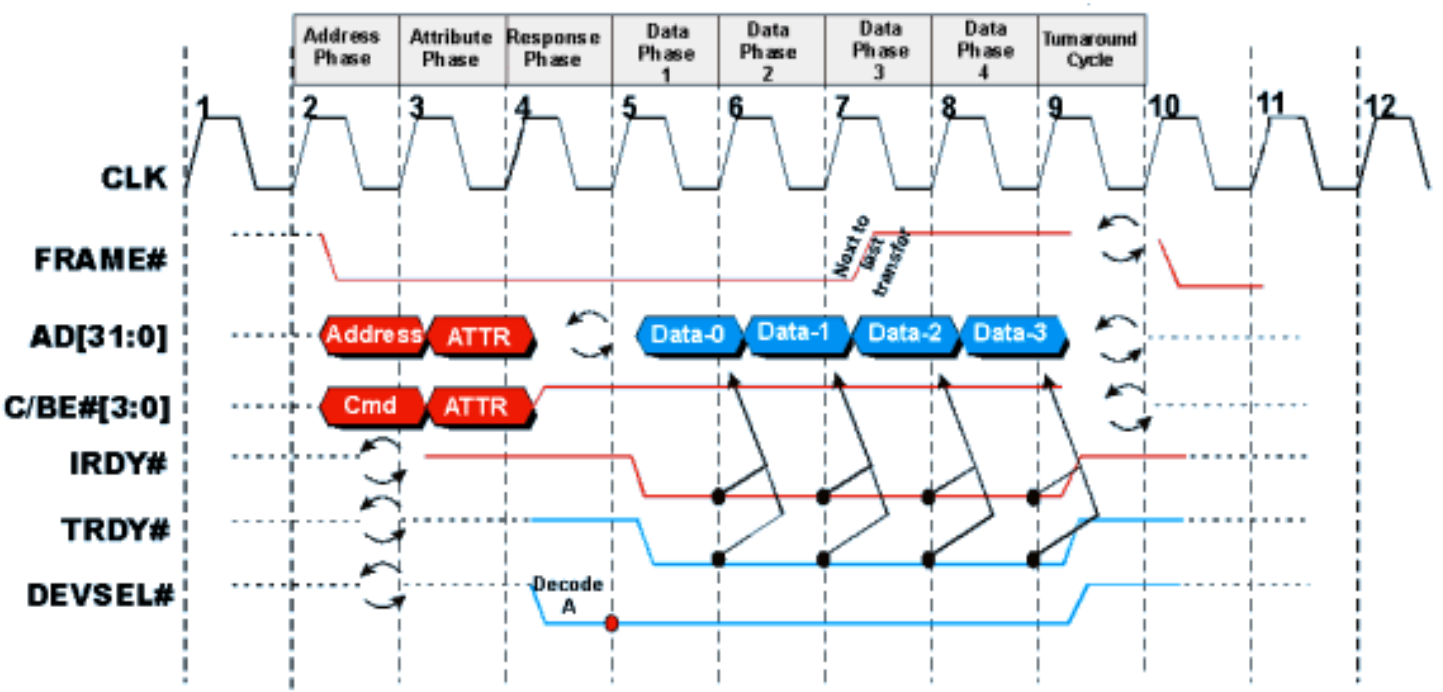
- Malá efektivita využití sběrnice - cca 50%! Režie zahrnuje: cykly potřebné pro arbitraci sběrnice, režie komunikačního protokolu, více-násobné opakování PCI Retry a PCI Disconnect, čekací stavy, apod.
- Nedostačující propustnost – fyzická realizace povoluje max. frekvenci 66 MHz, není možné připojit periferie jako jsou 1Gb Ethernet adaptér, výkonné RAID pole, apod. Velký počet vodičů komplikuje návrh desky a zvyšuje její cenu.
- Při přenosu není definována velikost dat – komplikuje návrh PCI zařízení, nutná správa bufferu
- Zpracování přerušení – sdílené signály, OS driver musí zjistit, kdo vyvolal přerušení.
- Správa chyb – parita je nedostačující, při identifikaci chyby se systém zhroutí a není definován způsob zotavení.
- Není podporována technologie Hot Plug – připojení/odpojení nového zařízení za chodu.
- Není podporováno řízení spotřeby (Power Management)

Obsah

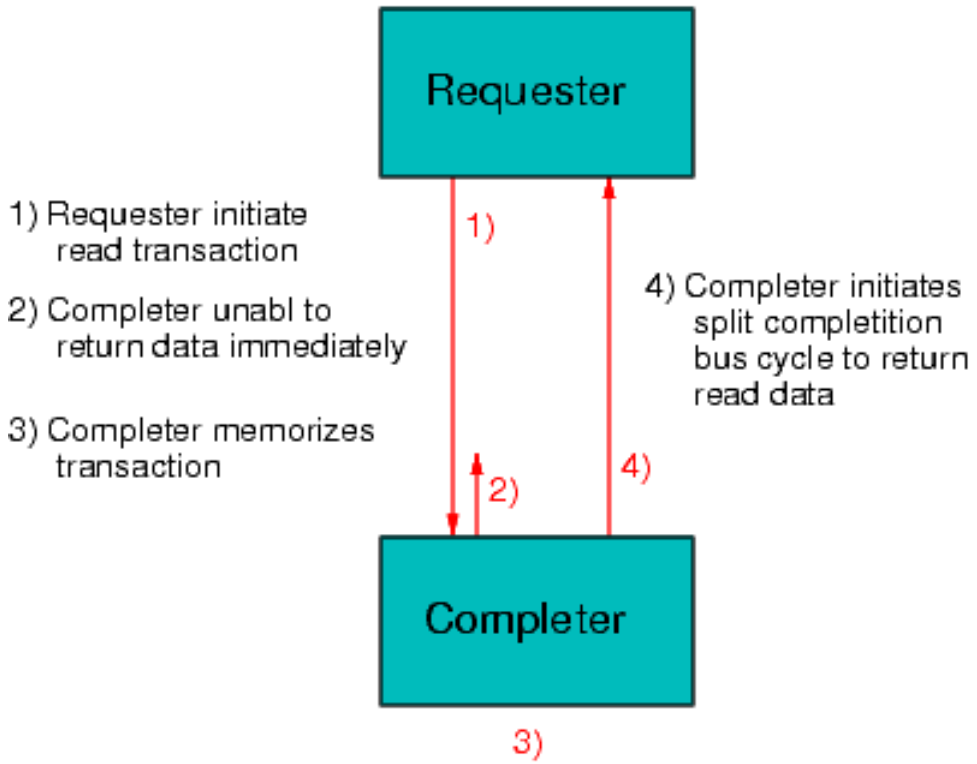
- Úvod
- PCI
- PCI-X
- PCI Express

Komunikační protokol

- Mezi vystavením adresy a dat je vložena ještě jedná fáze ATTRIB – vložení atributu, kde se obvykle posílá velikost dat. Cílové zařízení zná velikost dat => vede na lepší správu bufferů uvnitř PCI zařízení.
- Nepodporuje čekací stavy v okamžiku, když už se začaly přenášet data.
- Minimální velikost bloku dat je 128 bajtů => vyšší efektivita využití sběrnice až 85%



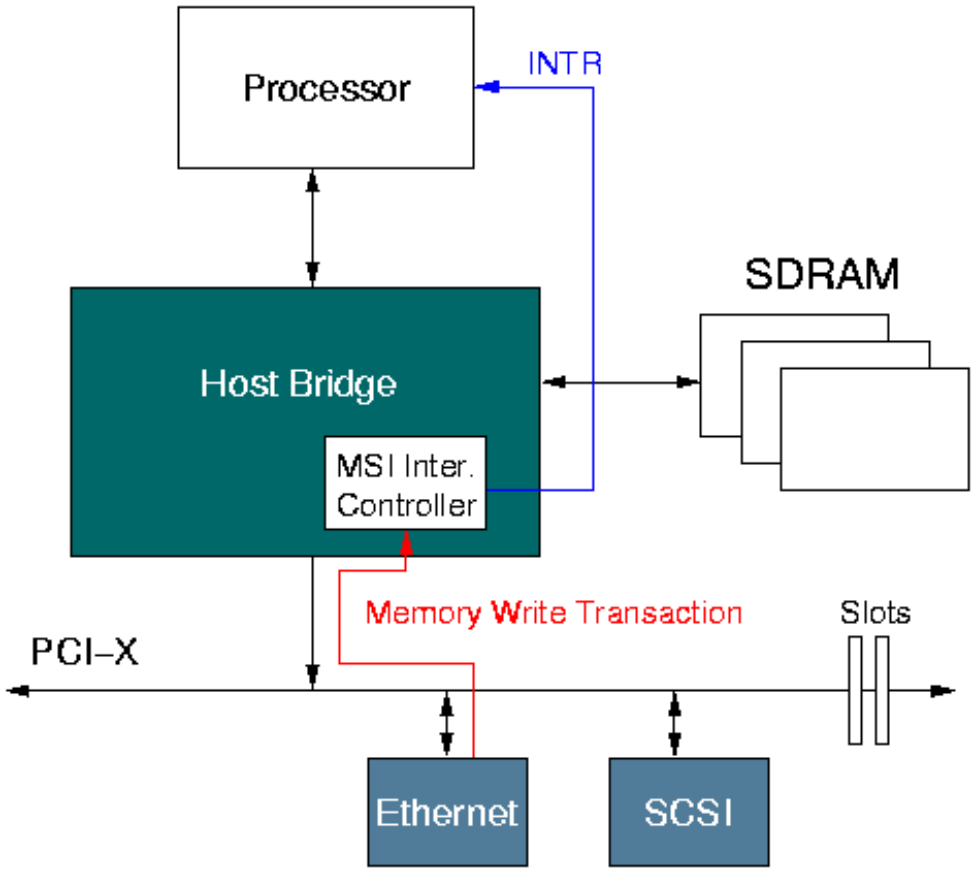
Model rozdělené transakce



1. Requester inicializuje čtecí operaci
2. Completer detekuje svou adresu a potvrzuje transakci. Zjistí ale, že není schopen poskytnout data okamžitě a rozpojuje transakci
3. Completer připraví požadovaná data do vnitřního bufferu (zná velikost)
4. Completer zahájí jednu nebo více zápisových operací, ve kterých pošle požadovaná data Requesteru

- Nedochází k opakovanému vyzývání ze strany Requesteru, pro načtení dat jsou potřeba maximálně dvě operace na PCI-X

Obsluha přerušení



- PCI-X zařízení musí podporovat MSI (Message Signaled Interrupt)

  Přerušení se negeneruje signály INTx, ale pomocí běžné zápisové transakce do prostoru MSI kontroléru

  V době inicializace:
  - PCI zařízení si určí počet přerušovacích vektorů (skrze konfigurační registry)
  - MSI kontrolér provede jejich alokaci a výslednou adresu zapíše zpět do zařízení

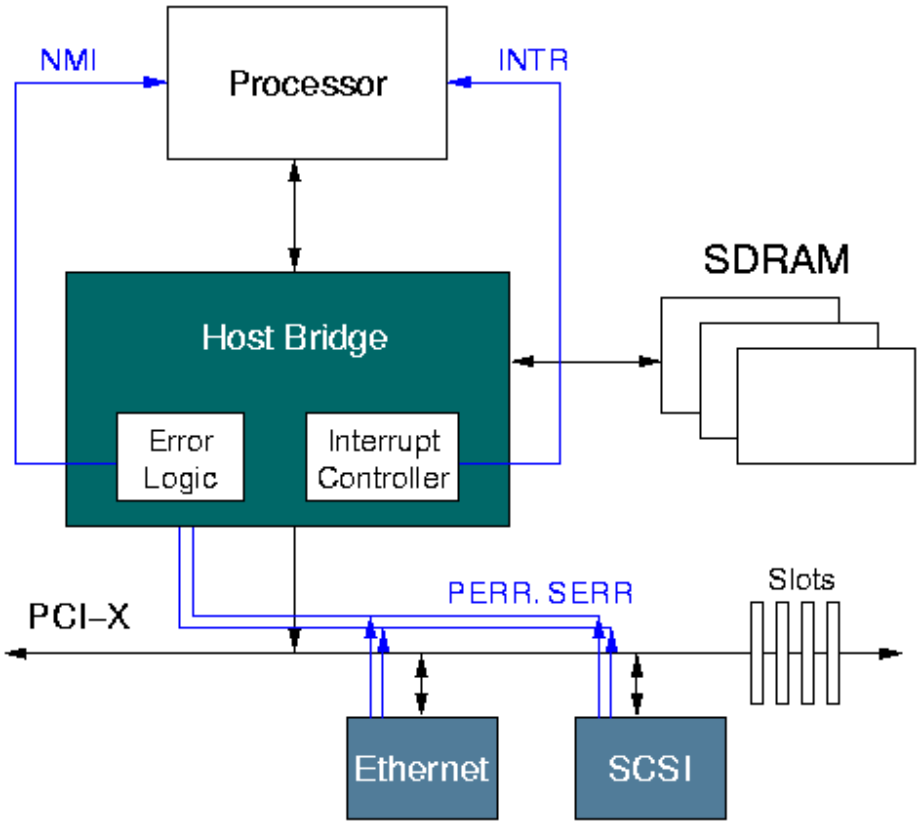
  Při přerušení dostane procesor přímo vektor identifikující obsluhu přerušení

  Odstraněno sdílení přerušení

  Není potřeba zpětně detekovat zdroj

  Pokud je přerušení spojeno s přenosem dat do RAM, generuje se až po ukončení přenosu

Obsluha chyb

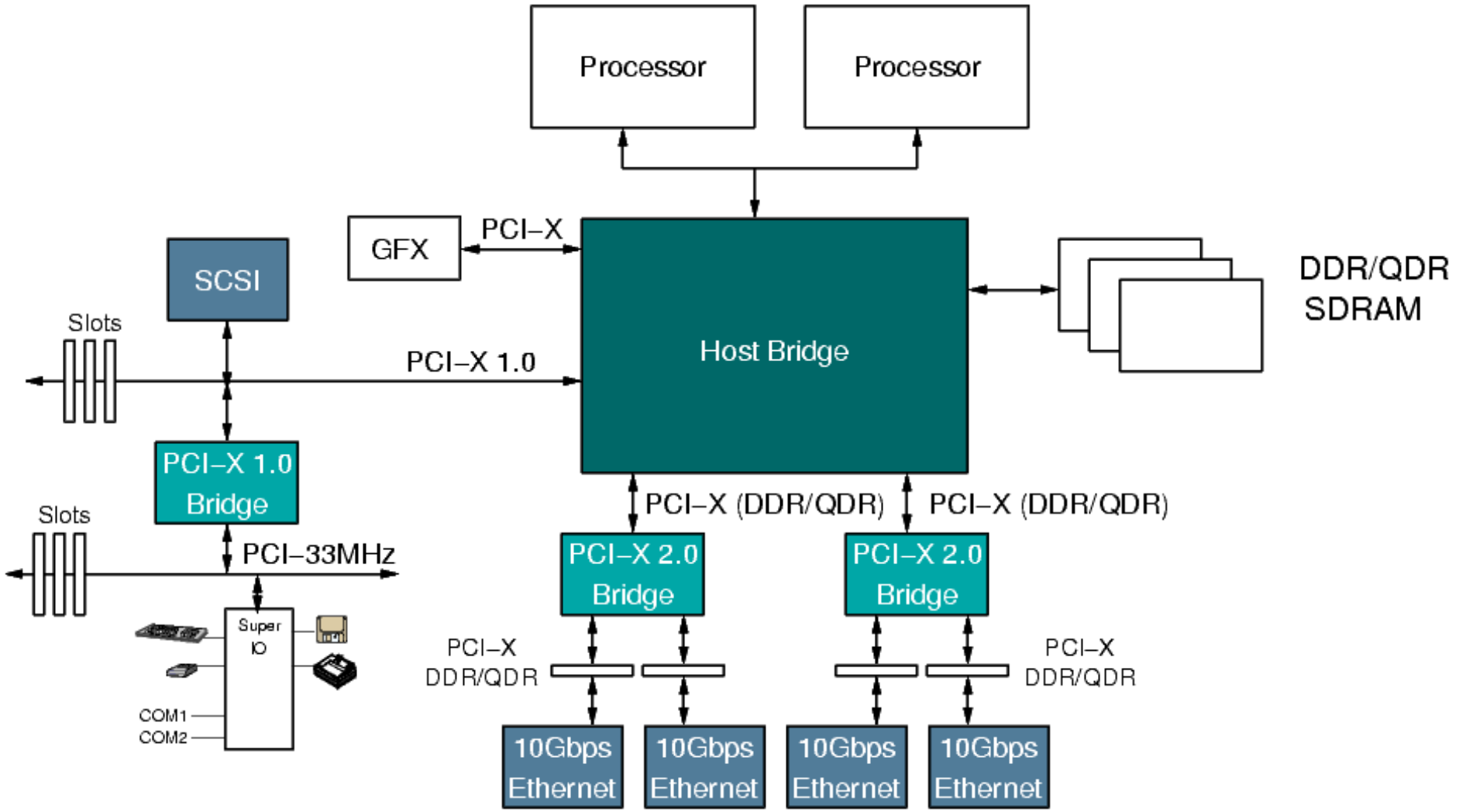


- Host bridge při aktivaci PERR (parity error) signálu už negeneruje NMI
- NMI přerušení generuje pouze při aktivaci SERR (system error) signálu
- Pokud PCI-X zařízení detekuje chybu parity, uloží si tuto informaci do stavového registru a vygeneruje běžné přerušení, které vyvolá obslužnou rutinu v ovladači
- Na ovladači zařízení je pak ponechán způsob obnovy ze vzniklé chyby (přeposlání transakce apod.)
- Pokud ovladač obnovu chyby parity nepodporuje, aktivuje zařízení SERR
- Pokud je do PCI-X připojeno PCI zařízení a je detekována chyba parity, Host bridge generuje NMI – režim jako u PCI

Další vlastnosti PCI-X

- Všechny signály jsou registrované => vede na vyšší rychlost, jednodušší konstrukci základní desky, možnost vložení více slotů
- Pro další zvýšení propustnosti použity technologie (PCI-X 2.0)
  - DDR – dvě datová slova v každém hodinovém taktu
  - QDR – čtyři datová slova v každém hodinovém taktu
- Počet slotů na jedné sběrnici:
  - max. 4 pro PCI-X 1.0
  - max. 1 pro PCI-X 2.0. Další sloty musí být odděleny skrze bridge => Pro PCI-X (2.0) musí být každý slot připojen na vlastní bridge (téměř Point-to-Point spojení). Čtecí operace mají velkou latenci (putují přes bridge) a téměř vždy se uplatňují rozdělené čtecí transakce.

Příklad platformy pro PCI-X

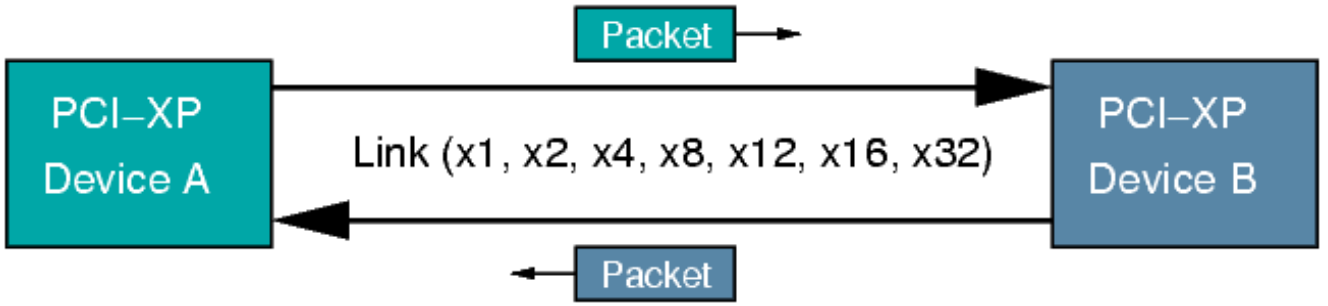


Obsah

- Úvod
- PCI
- PCI-X
- PCI Express

Základní vlastnosti

- PCI Express je založena na rychlých plně duplexních sériových linkách (Lane) s kapacitou 2.5Gb/s (do budoucna 5Gb/s až 10Gb/s)



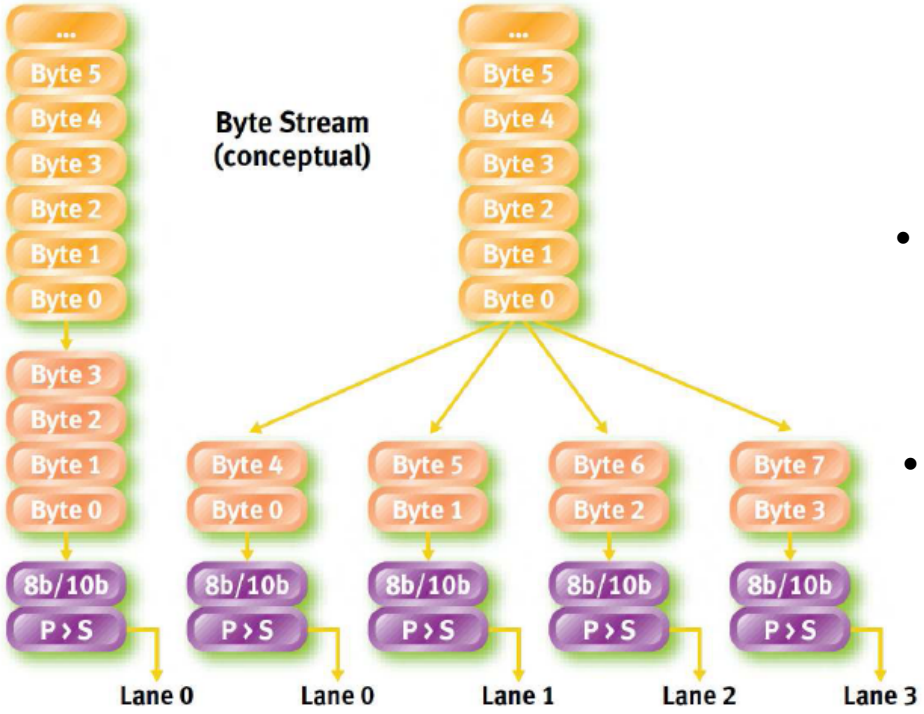
- pro jeden kanál lze paralelně zapojit až 32 těchto linek s přenosovou kapacitou podle následující tabulky

Počet linek	x1	x2	x4	x8	x12	x16	x32
Propustnost GB/s	0.5	1	2	4	6	8	16

- do celkové propustnosti je zahrnut vliv kódování 8/10 (20% režie)

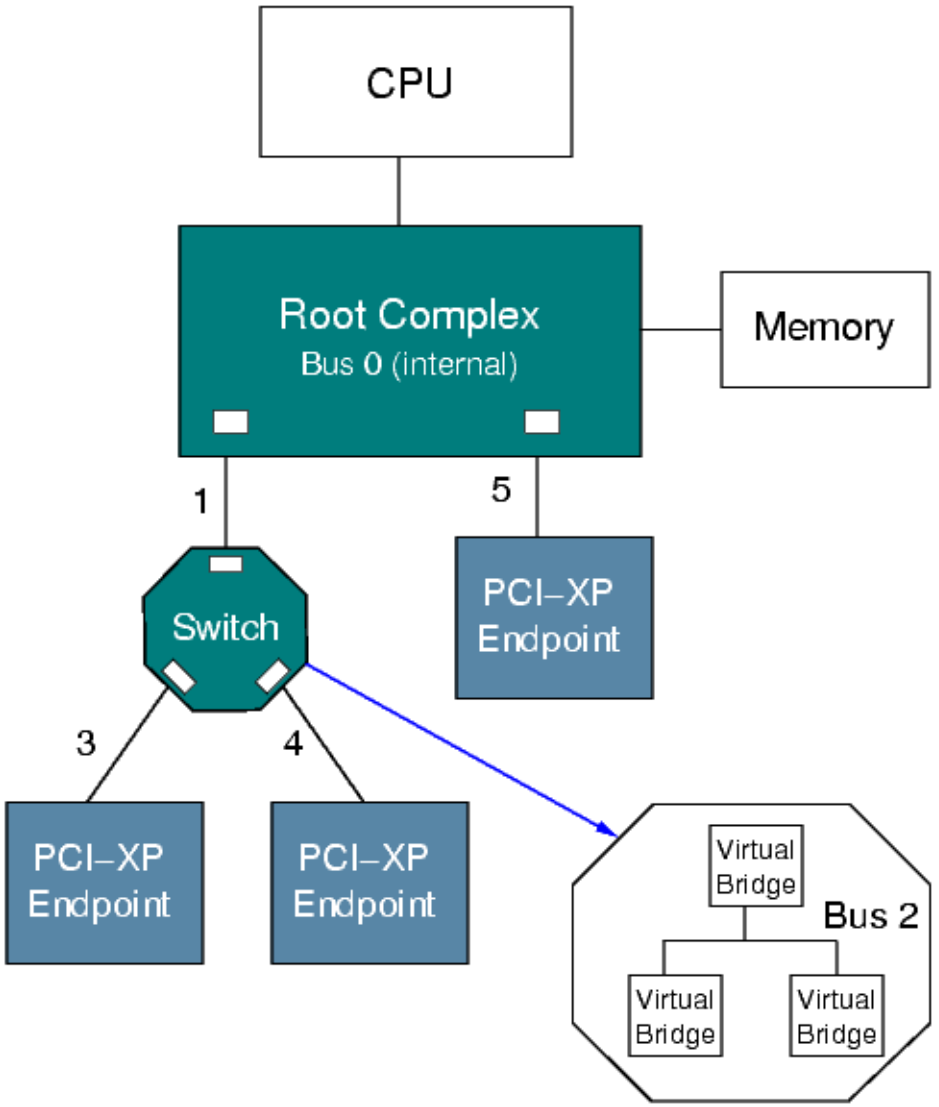
Základní vlastnosti

- Veškeré transakce PCI Express probíhají skrze paketovou komunikaci a data paketu jsou rozdělována rovnoměrně do jednotlivý linek
- Při připojení zařízení se obě strany musí dohodnout na komunikující rychlosti a počtu linek



- *Způsob obsluhy přerušení*
  - přerušení jsou odesílána pomocí MSI podobně jako u PCI-X. Součástí zprávy je informace o zařízení, které přerušení vyvolalo, včetně vektoru.
- *Řízení spotřeby (Power Management)*
  - spotřeba každého zařízení/linky lze individuálně řídit, např. skrze SW pomocí zasílání zpráv, popřípadě se mohou zařízení/linky automaticky uspávat v době snížené aktivity
  - rozlišují se stavy: *pro zařízení*: D0, D1, D2, D3-Hot, D3-Cold, *pro linku*: L0, L1, L2, L3
- *Správa chyb*
  - každý paket je zabezpečen pomocí CRC
  - vzniklé chyby se zapisují do logu a ošetřují na různých úrovních
- *Hot Plug*
  - v průběhu činnosti systému lze připojovat/odpojovat nové PCI zařízení. Pro tyto účely je na desce ke každému slotu speciální tlačítko a dvojice LED diod pro signalizaci stavu napájení.

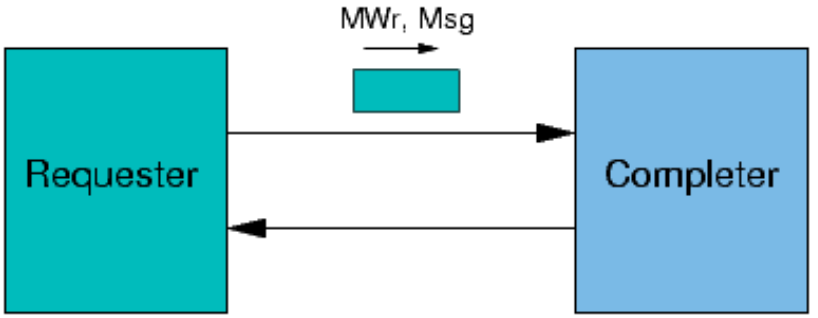
Topologie PCI Express



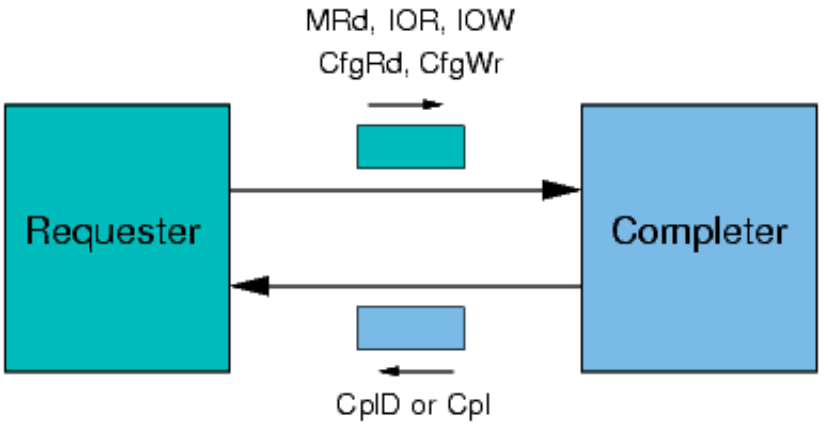
- PCIe = stromová topologie
  - *Root Complex* – centrální bridge mezi CPU, Memory a PCIe sběrnicí
  - *Endpoints* – koncová zařízení
  - *Switch* – přepínač mezi více linkami
  - Každé zařízení je v rámci topologie jednoznačně identifikováno pomocí trojice Bus Number, Device Number a Function Number
- Číslování sběrnic (linek)
  - linka 0 je virtuální uvnitř Root
  - ostatní jsou číslovány do hloubky
  - uvnitř přepínače je také číslována pomyslná virtuální sběrnice připojující jednotlivé porty
- Na každé lince jsou pouze dvě zařízení 0 a 1 (0 je ve směru dolů)
- Každé zařízení může mít až 8 funkcí

Transakce na PCI Express

- Transakce reprezentuje tok dat složený z jednoho nebo více paketů, kde maximální velikost jednoho paketu je 4kB
- V každé transakci vystupuje
  - *Requester* – zařízení, které inicializuje transakci
  - *Completer* – cílové zařízení, které „kompletuje“ transakci
- Komunikace mezi:
  - Root <=> Endpoint,
  - Endpoint <=> Endpoint
- Transakce se dělí na *Posted* (bez odpovědi) na *Non-posted* (s odpovědí)



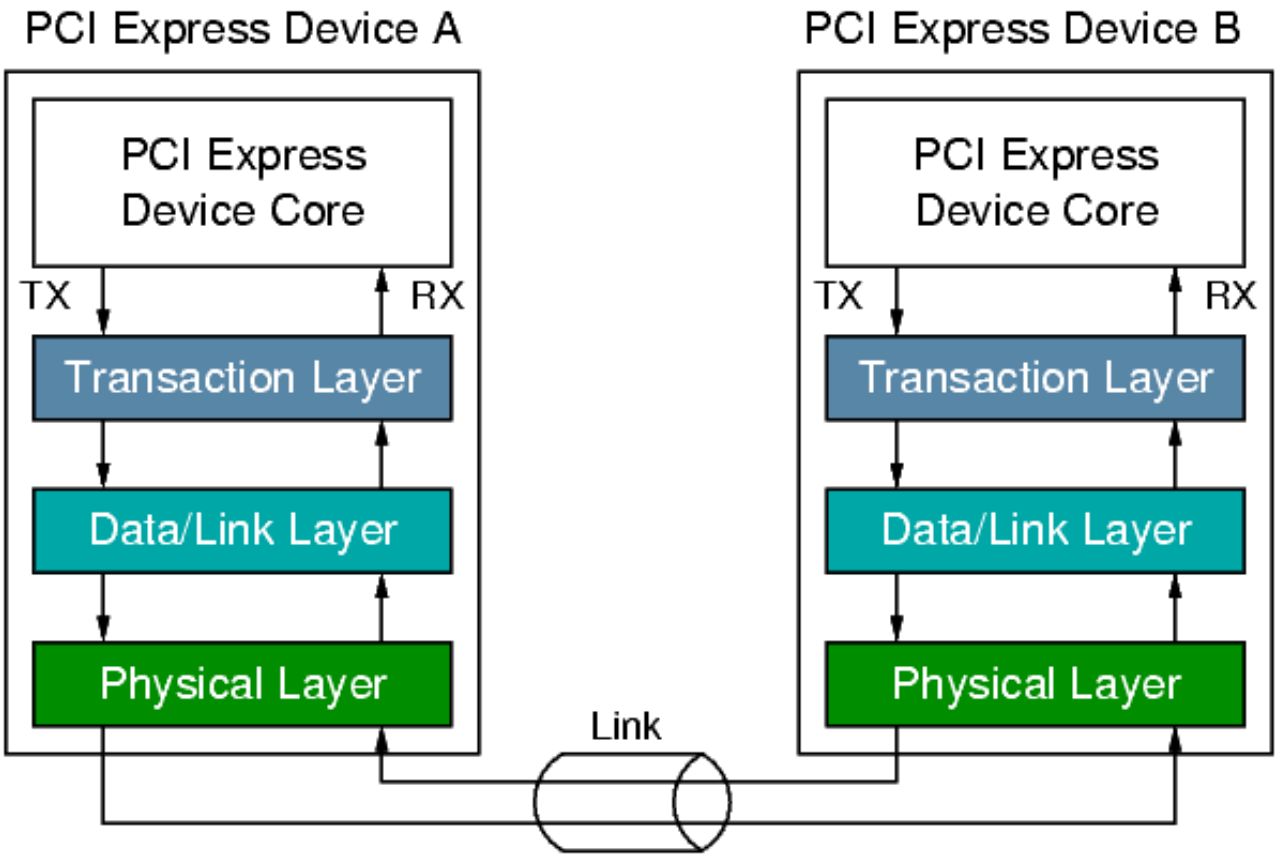
Obr.: Posted transakce



Obr.: Non-posted transakce

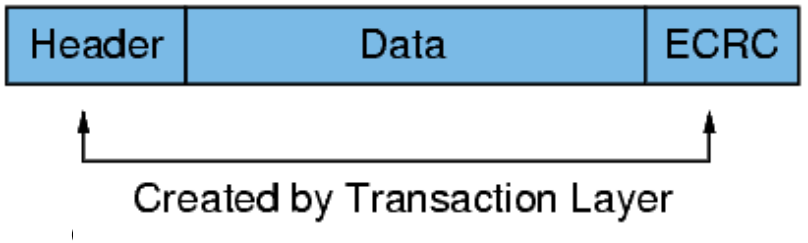
Vrstvový model

- Architektura každého zařízení je logicky členěna do několika vrstev:
  - *Transakční vrstva* – řídí přenos paketu mezi libovolnými uzly
  - *Linková vrstva* – řídí přenos paketu mezi sousedními uzly
  - *Fyzická* – realizuje přenos dat na nejnižší úrovni (digitální i analogové)

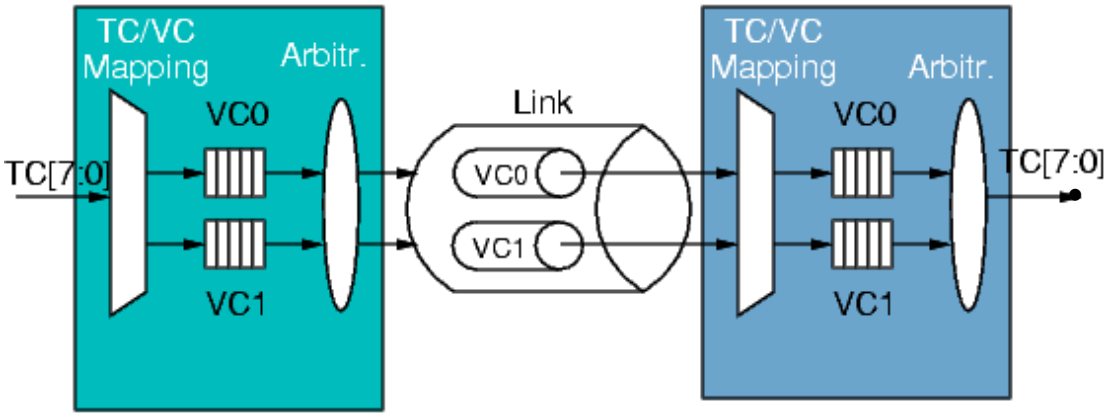


Transakční vrstva

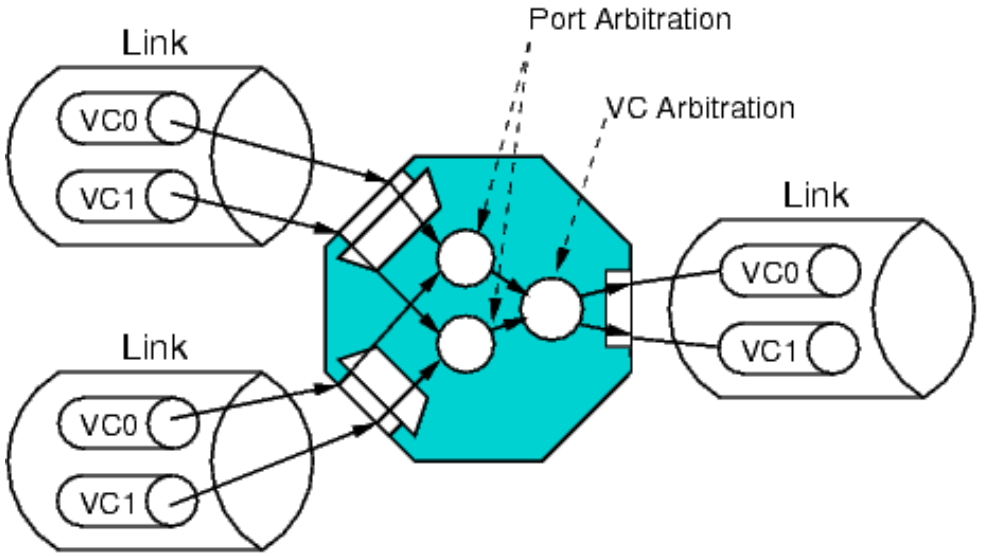
- Stará se o přenos dat mezi PCI zařízeními včetně jejich směrování přes přepínače.
- Na základě informací z jádra PCI zařízení (Identifikace cíle, typ transakce, velikosti dat, samotna data) sestavuje hlavičku paketu, připojuje data paketu a vypočítává volitelné CRC pro aplikace s vysokým požadavkem na spolehlivost.
- Obsahuje buffery pro příchozí a odchozí pakety, rozhoduje o prioritě jejich zpracování (QoS) a výměnu informací o velikostí volného místa mezi jednotlivými zařízeními na lince (Flow Control)
- Zahrnuje řízení spotřeby (Power Magement) pro PCI zařízení. Probíhá automaticky bez účasti SW nebo jádra PCI zařízení. Podporováno ACPI/ PCI
- Řeší konfiguraci PCI zařízení. Součástí je nastavení bázových adres PCI zařízení a velikosti paměťových bloků.



Quality of Services (QoS)



Obr.: Vyhodnocení priority v zařízení

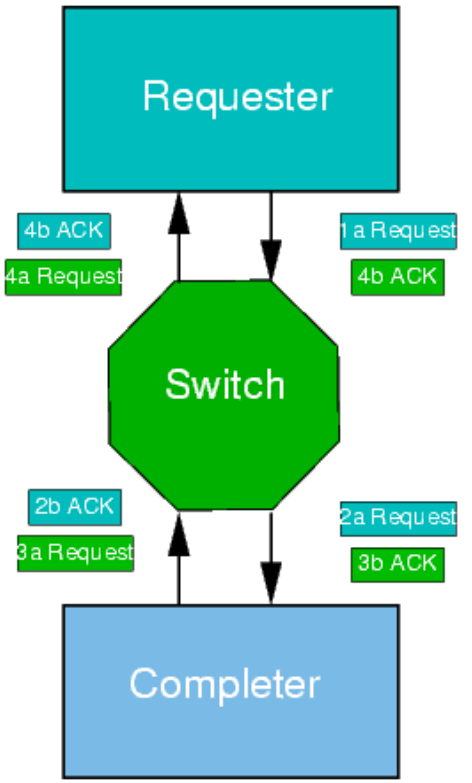


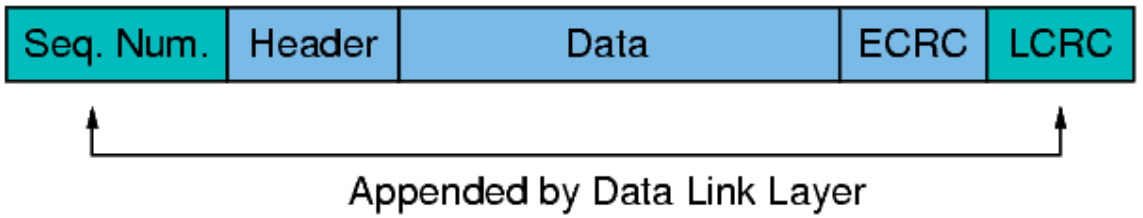
Obr.: Vyhodnocení priority v přepínači

- Priorita mezi pakety se rozlišuje položkou Traffic Class v hlavičce paketu. Může nabývat hodnot TC0-TC7 (7 je nejvyšší priorita).
  Skutečný přenos dat přes jeden fyzický kanál (linku) může probíhat ve více logických kanálech v tzv. Virtual Channels (VC). V rámci jedné linky může být max. 8 kanálů VC0-VC7.
- Na transportní vrstvě probíhá mapování TC => VC. Více TC může být mapováno do jednoho VC.
- Skrze konfigurační registr lze vybrat jeden z následujících způsobů vyhodnocení priority:
  - *statická,*
  - *Round-Robin,*
  - *váhovaný Round-Robin,*
  - *časovaný Round-Robin*

Datová/linková vrstva

- Zajišťuje integritu dat přenesených skrze jednu linku
- Postup při přenosu paketu na linkové vrstvě:
  1. Ke hlavičce a datům paketu je připojeno sekvenční číslo a je vypočítám CRC kód pro zabezpečení dat.
  2. Paket je odeslán skrze fyzickou vrstvu a současně je uložen do *Reply bufferu*.
  3. Na přijímací straně linková vrstva paket přijme, zkontroluje CRC a zkontroluje sekvenční číslo (pakety nesmí přicházet mimo pořadí)
  4. Pokud je paket v pořádku, odešle se *ACK* paket společně se sekvenčním číslem a vysílací strana uvolní paket z Reply bufferu.
  5. Pokud je identifikována chyba, potom se odešle *NACK* paket a vysílací strana musí poslat paket znovu z *Reply bufferu*. Tímto způsobem může být paket přeposlán nejvýše 4 krát, potom je propagována chybová zpráva a informace o chybě je uložena do log. registru.

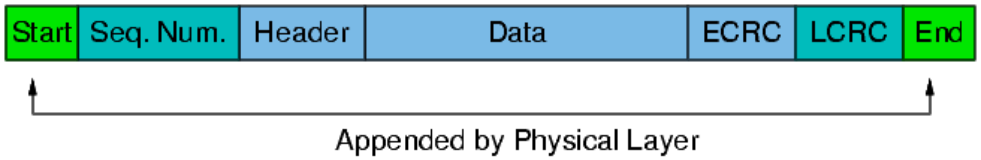




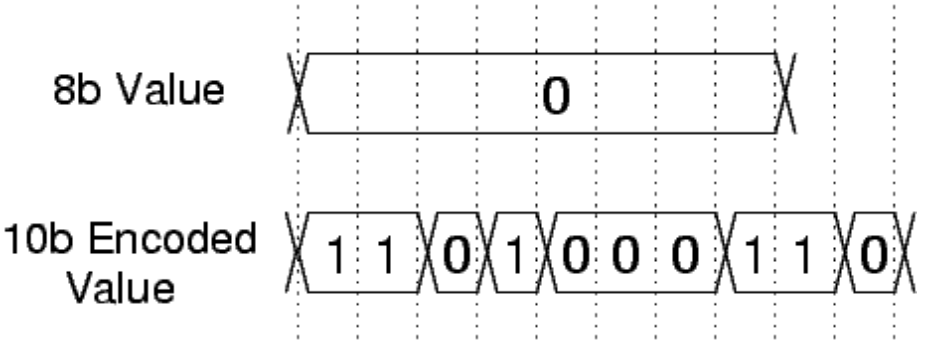
Obr.: Struktura DLLP paketu

Fyzická vrstva

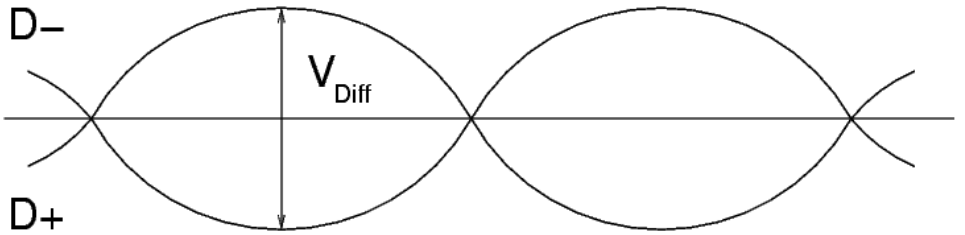
- Realizuje přenos paketu na fyzické vrstvě (Logická i analogová část)
- Postup při přenosu paketu na fyzické vrstvě:
  1. Ke každému paketu jsou připojeny řídící znaky *Start* a *End*
  2. Paket je rozdělen po 8-bitech na jednotlivé linky
  3. Provede se *kódování 8/10* – každých 8 bitů dat je zakódováno na 10 bitů tak, aby byl zachován vhodný počet přechodů *0->1* a *1->0*. Tyto přechody jsou klíčové na přijímací straně pro obnovení hodinového signálu (není potřeba další drát pro přenos hodinového signálu).
  4. Následuje *serializace* dat a jejich odeslání diferenciálním spojem



Obr.: Struktura rámce na fyzické vrstvy



Obr.: Příklad kódování 8/10

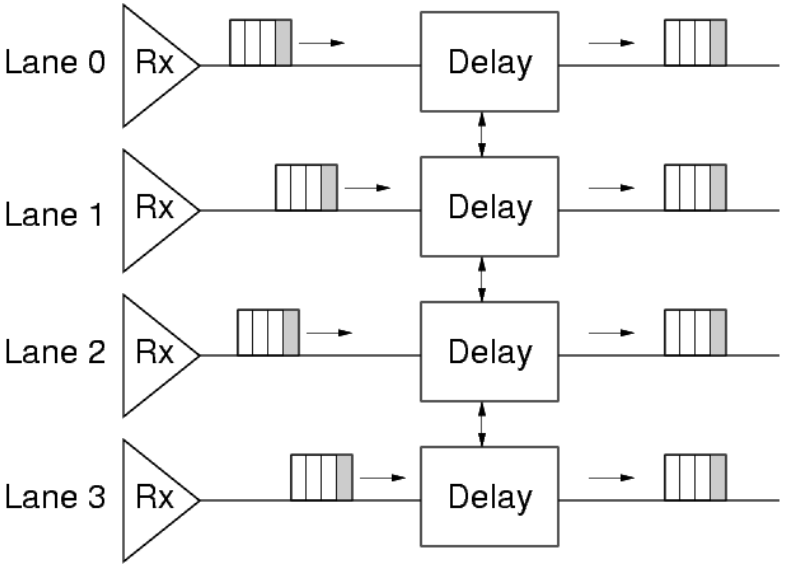


Obr.: Diferenciální signál

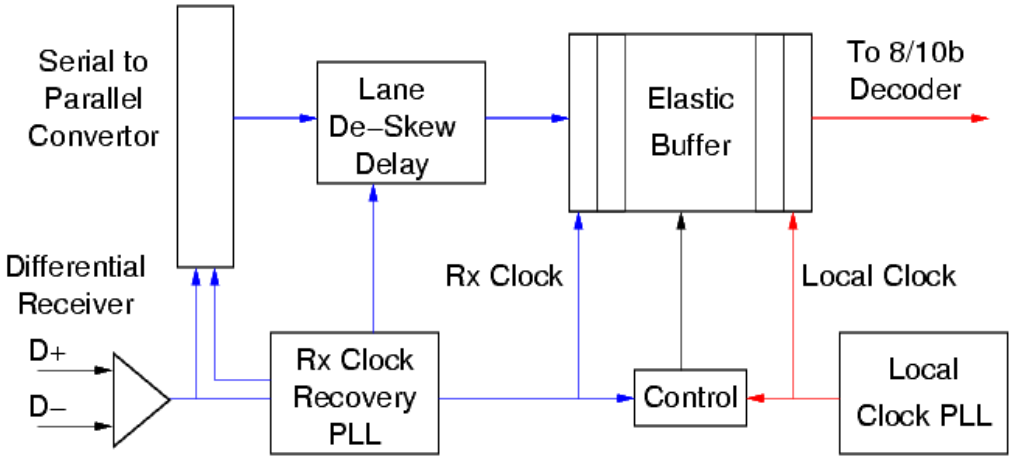
Fyzická vrstva

5. Na přijímací straně je *obnoven hodinový signál*, data jsou deserializována a je vyrovnáno zpoždění dat mezi jednotlivými linkami.
6. Pomocí elastických bufferů jsou data převedena na lokální hodinový signál a provede se *dekódování 8/10*

- Další důležitou funkcí je inicializace a trénování linky. Tento proces probíhá automaticky bez účasti SW nebo jádra PCI zařízení a zahrnuje:
  - detekce počtu linek (obě připojená zařízení se musí dohodnout na počtu linek)
  - detekce přenosové rychlosti (2.5, 5 a 10 Gb/s)
  - detekce polarity diferenciálního spoje popřípadě reverzace pořadí linek

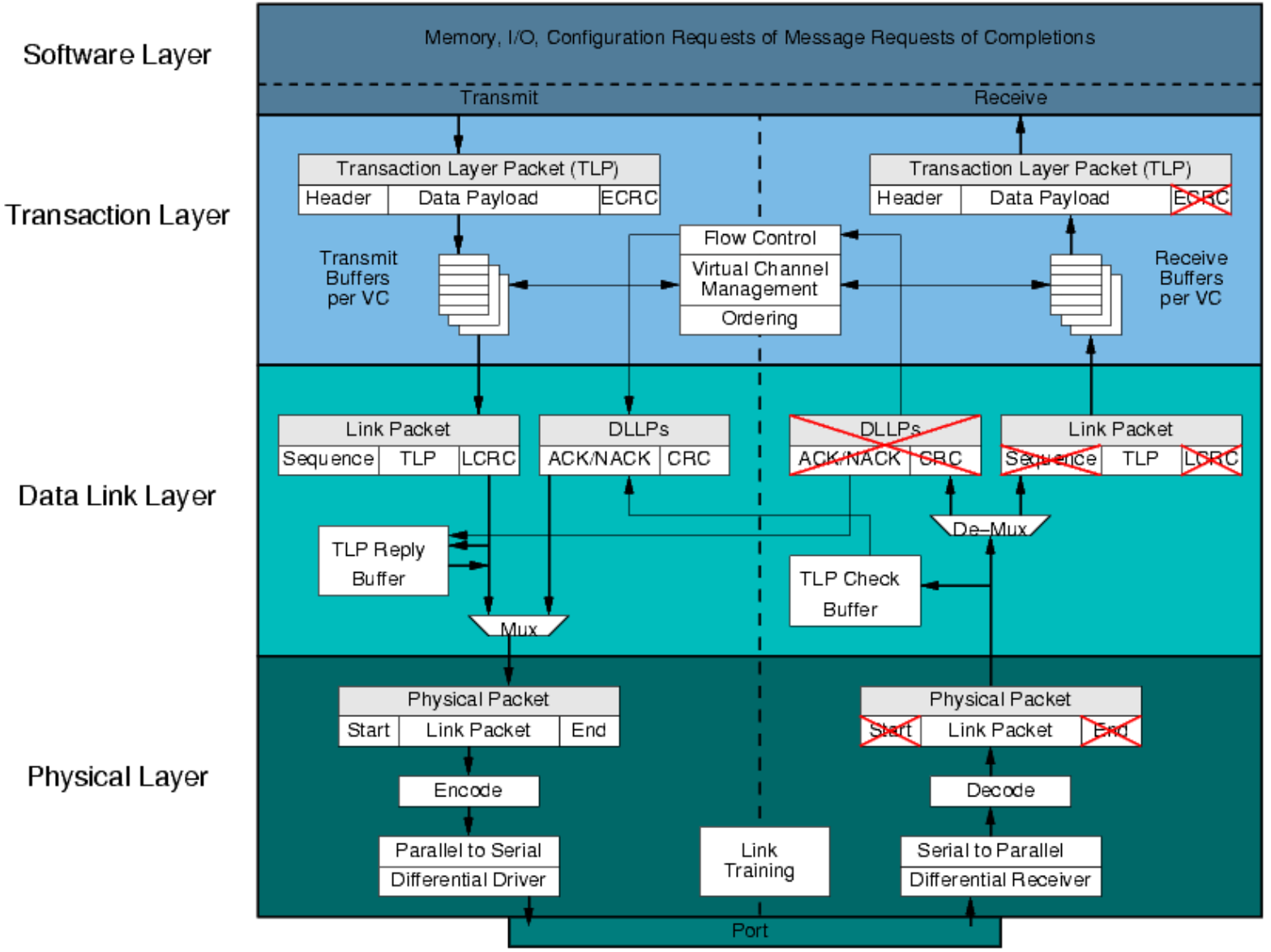


Obr.: Vyrovnávaní zpoždění mezi linkami



Obr.: Blokové schéma přijímací části

Celkové schéma vrstev na PCIe



Režie protokolu PCIe

- Maximální velikost paketu (MTU) je dnes obvykle 128 bajtů
- Každý paket má 20-28 bajtů režie (hlavičky na všech úrovních+CRC)
- Každých 8 bitů je zakódováno pomocí 10-ti bitů
- Příklad:
  - Jaká je efektivita při přenosu 128bajtů dat?
  - (128*8) / (128+20)*10 = 1024/1480 = 0,69 → cca 69%
- Další režie:
  - Každý odeslaný paket (nebo skupina paketů) je potřeba potvrdit zprávou ACK
  - Jednou za určitou dobu je potřeba vložit mezeru pro vyrovnání rozdílu hodinových signálu na lince

Další vývoj PCIe

- Příprava verze 3.0 (konec roku 2010)
- Místo kódování 8/10 použit pouze scrambling – ještě není úplně promyšleno
- Vede na dosažení dvojnásobné propustnosti na datech oproti PCIe 2.0, i když propustnost linky není dvojnásobná

Typ	Propustnost linky	Max. propustnost
PCIe 1.0	2,5Gb	2Gb
PCIe 2.0	5Gb	4Gb
PCIe 3.0	8Gb	8Gb

Shrnutí PCIe

- Výhody:
  - Zcela nová technologie přenosu dat – vysokorychlostní sériové linky
  - Snadná škálovatelnost
  - Na místo spousty vodičů různé typy paketů (nové funkce: Hot Plug, Power Management, atd.)
- Nevýhody:
  - Velká režie přenosu – bude vyřešeno u PCIe 3.0
  - Velká latence přenosu – využití cache pamětí

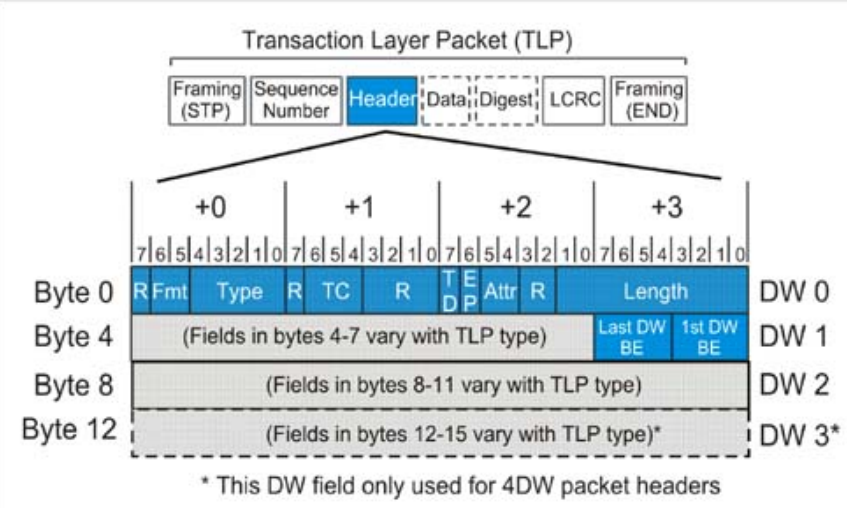
Reference

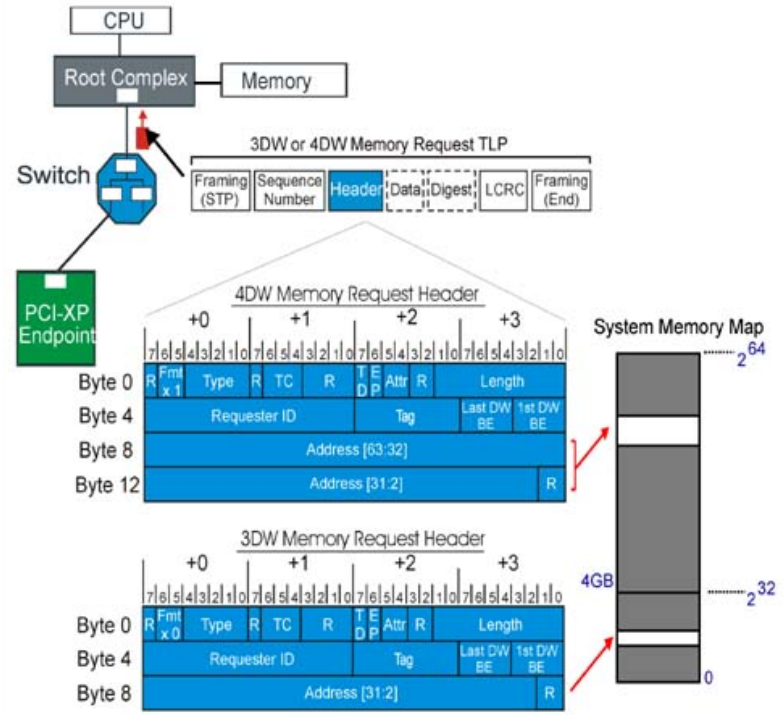
- ***PCI Express System Architecture***, By Don Anderson, Ravi Budruk, Tom Shanley, September 4, 2003

Konec přednášky

Děkuji za pozornost

Formát paketu





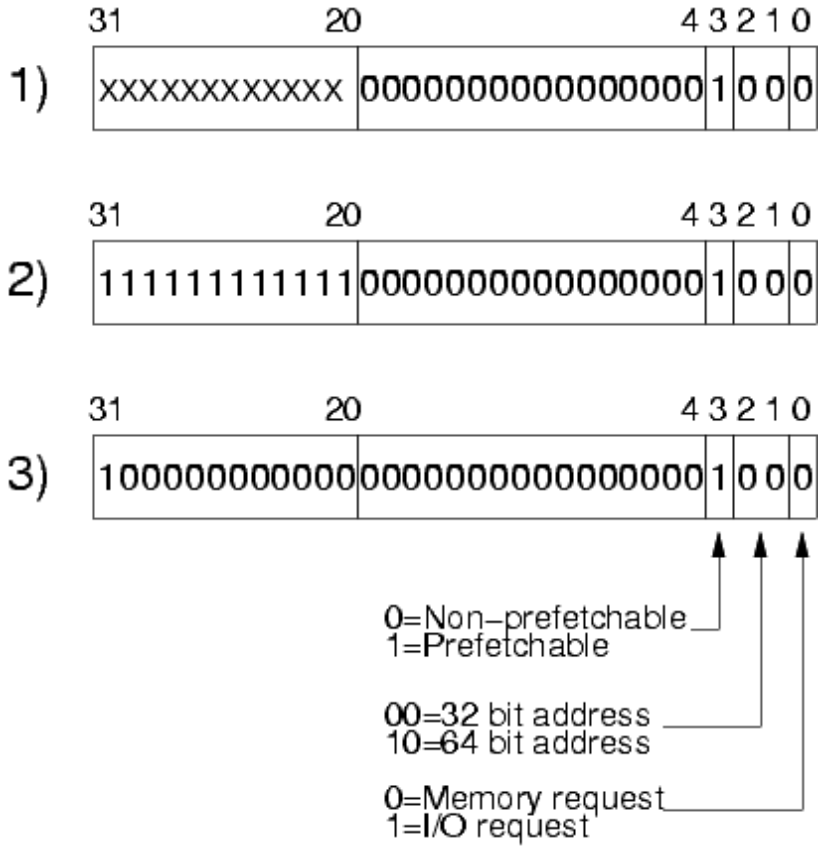
Obecný formát hlavičky paketu:

- *Type* – Identifikuje typ transakce
  - Memory Read/Write/Compl.
  - IO Read/Write,
  - Config. Read/Write
- *TC* – Traffic Class
- *Fmt* – rozlišuje mezi délkou hlavičky 3DW/4DW
- *Length* – určuje délku paketu v počtech DWORD
- *BE* – identifikuje zarovnání prvního a posledního DWORD

Příklad: 32/64-bitová pam. operace

- Tělo hlavičky obsahuje:
  - Cílovou adresu, Identifikaci zdroje
  - TAG

Postup alokace paměti na PCI



Obr.: Příklad alokace 1MB paměti

- Postup při obsluze požadavku na alokaci paměťového prostoru je následující:

0. Systémový SW nejprve zapíše do neinicializovaného BAR registru samé jedničky. Bity, které jsou napevno nastaveny výrobcem, nebudou ovlivněny.
1. Systémový SW si přečte hodnotu BAR registru a zjistí tak požadovaný typ a množství paměti. Po získání všech požadavků SW rozdělí prostor.
2. V posledním kroku provede systémový SW zápis hodnoty do BAR registru, která odpovídá bázové adrese přiděleného prostoru.