

Sběrnice a periferní zařízení

INP 2019
FIT VUT v Brně



Přehled přednášky

- **Sběrnice**

- Motivace, princip
- Základní pojmy
- Varianty sběrnic
- Arbitrace na sběrnici

- **Periferní zařízení**

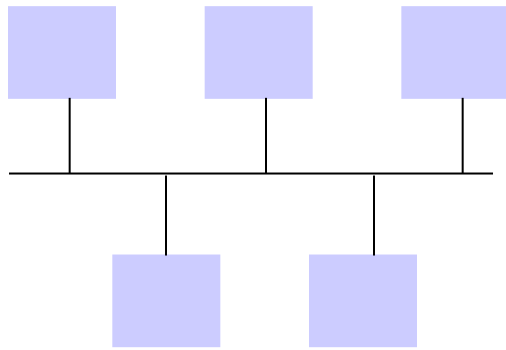
- Připojování
- Obsluha
- Systém USB

Zkratky:

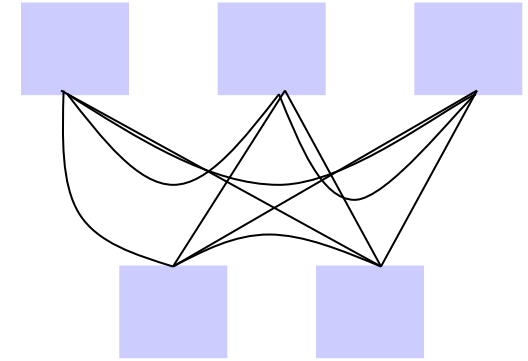
- I/O = V/V – vstup/výstup
- PZ – periferní zařízení
- RVP – rychlá vyrovnávací paměť
- M – hlavní paměť
- IOP – I/O procesor

Motivace: jak zajistit komunikaci více prvků v systému?

Propojením. Rozhoduje poměr výkon / cena!



Vybrané varianty propojení



Sběrnice:

- všechna zařízení sdílí relativně nízký počet vodičů
- dlouhé vodiče => nižší kmitočty
- způsob komunikace: jedna komponenta vysílá, všechny mají možnost přijímat
- relativně nízká propustnost pokud je připojen vyšší počet zařízení

Point-to-point linky:

- samostatné vodiče pro každé zařízení (vyšší cena)
- relativně vysoký počet vodičů
- krátké vodiče => vyšší kmitočty
- komunikace: point to point
- vysoká propustnost (souběžná komunikace mezi několika páry jednotek je možná)

Základní pojmy

Sběrnice: propojovací soustava umožňující komunikaci mezi více než jedním *zdrojem dat a přijímači dat*.

Komunikační rozhraní: soustava vodičů s definovaným významem a elektrickými charakteristikami.

Komunikační protokol: určuje pořadí změn hodnot signálů rozhraní tak, aby komponenta mohla komunikovat s okolím.

Transakce (cyklus): posloupnost kroků (hodnot signálů rozhraní) realizující danou operaci (funkci), např. přečtení dat z dané adresy.

Způsoby komunikace:

- *synchronní – asynchronní*
- *sériová – paralelní*
- *simplexní – poloviční (half) duplex – plný (full) duplex*

Varianty sběrnic:

- *dle způsobu komunikace: viz výše*
- *dle účelu: adresová, datová, řídicí
multiplexovaná, dedikovaná*
- *dle umístění, přístupu: interní – externí*
- *atd.*

**Parametry sběrnice
podstatným způsobem
ovlivňují výkonnost systému!**

Příklady nejčastěji vyskytujících se standardů sběrnic

[http://cs.wikipedia.org/wiki/Sběrnice](http://cs.wikipedia.org/wiki/Sb%C4%99nice)

- **ISA** - starší typ pasivní sběrnice, šířka 8 nebo 16 bitů, přenosová rychlost < 8 MB/s
- **PCI** - novější typ „inteligentní“ sběrnice, šířka 32 nebo 64 bitů, burst režim, přenosová rychlost < 130 MB/s (260 MB/s)
- **AGP** - jednoúčelová sběrnice určená pro připojení grafického rozhraní (karty) k systému, přenosová rychlost 260 MB/s - 2 GB/s
- **PCI-X** - zpětně kompatibilní rozšíření sběrnice PCI
- **PCI-Express (PCIe)** - nová sériová implementace sběrnice PCI
- **USB** - sériová polyfunkční sběrnice, 2 diferenciální datové vodiče + 2 napájecí vodiče 5 V/500 mA, široké použití, verze 1.1 přenosová rychlost 12 Mb/s (~1,43 MB/s), 2.0 přenosová rychlost 480 Mb/s (~57 MB/s), 3.0 přenosová rychlost 4800 Mb/s (~572 MB/s)
- **FireWire** - sériová polyfunkční sběrnice, široké použití, 50 MB/s
- **RS-485** - sériová průmyslová sběrnice (někdy jako proudová smyčka), do prostor s vysokým elektromagnetickým rušením
- **I²C** - sériová sběrnice, < 100 kb/s, adresace 32 zařízení, komunikace a řízení v elektronických zařízeních

Sériová vs. paralelní komunikace (sběrnice)

Paralelní: v jednom kroku (taktu) je přeneseno více bitů

Bitová šířka:

- datová sběrnice – počet bitů významně ovlivňuje výkonnost (propustnost)
- adresová sběrnice – počet bitů určuje max. paměťovou kapacitu systému

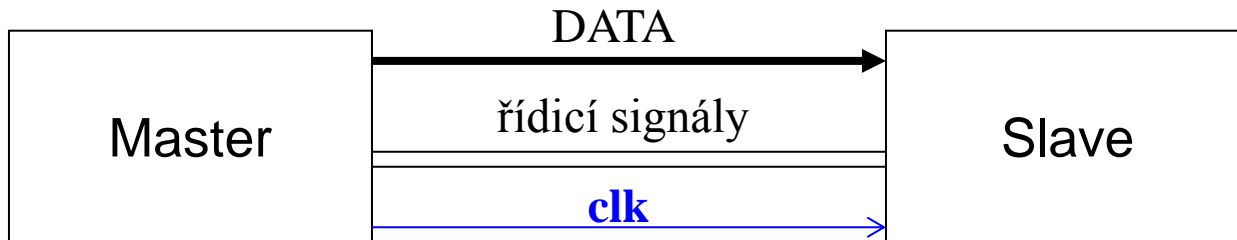
Problém při vzrůstající frekvenci: interference paralelních vodičů, problém se synchronizací (angl. skew)

Sériová: komunikace probíhá po jednotlivých bitech (jeden bit za takt)

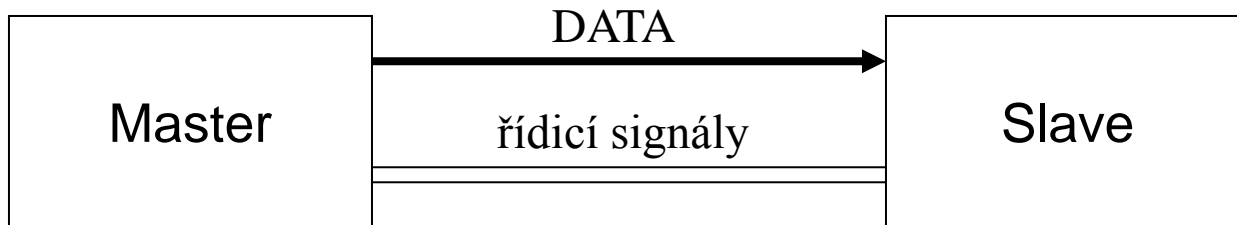
Výhoda: značně nižší počet vodičů, eliminace výše uvedených problémů

Synchronní vs. asynchronní přístup

Synchronní: existuje **vyhrazený vodič** pro přenos hodinového signálu, veškeré děje jsou synchronizovány tímto signálem. Musí existovat prvek, který tento signál generuje (obvykle tzv. master).



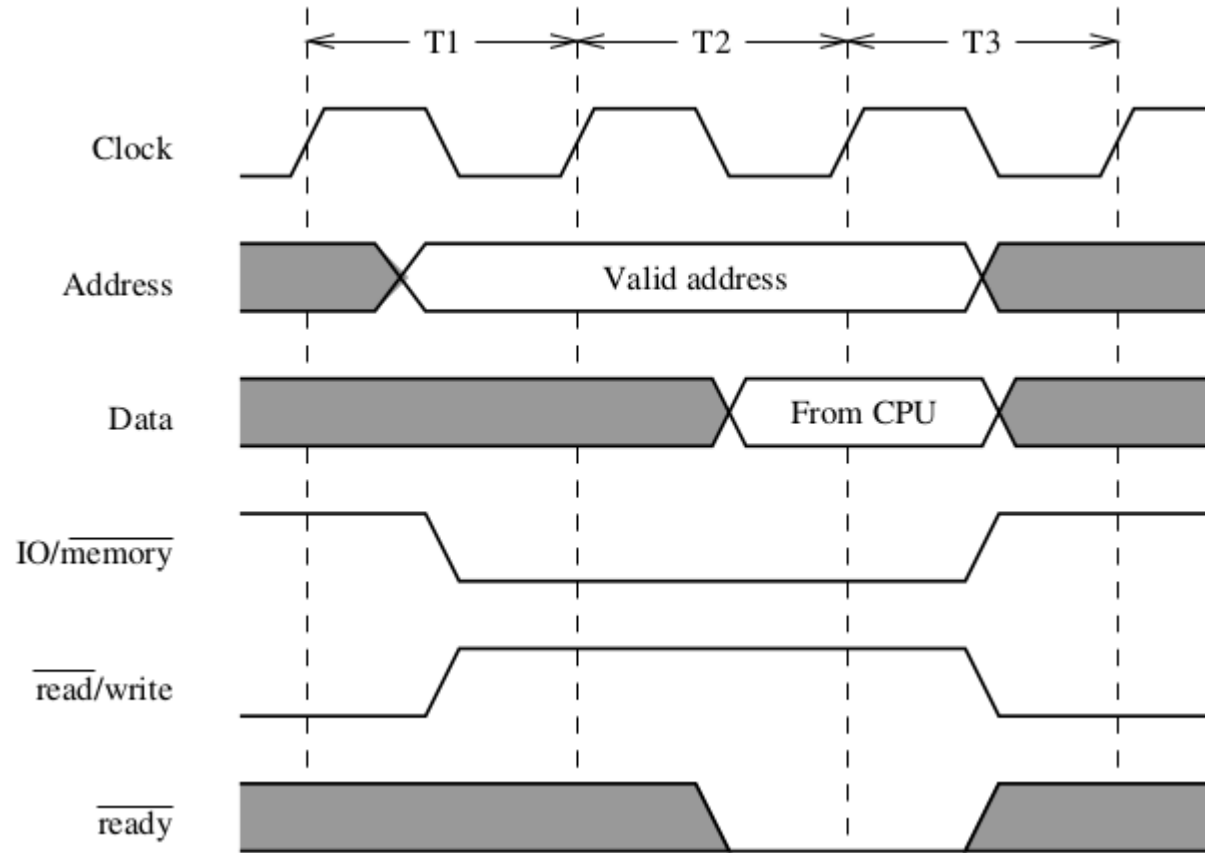
Asynchronní: součástí dějů **není** synchronizační signál, generování nějakého signálu je vázáno na výskyt události (změnu hodnoty) na určitém signálu. Průběh komunikace je řízen **komunikačním protokolem**:



Základní možnosti:

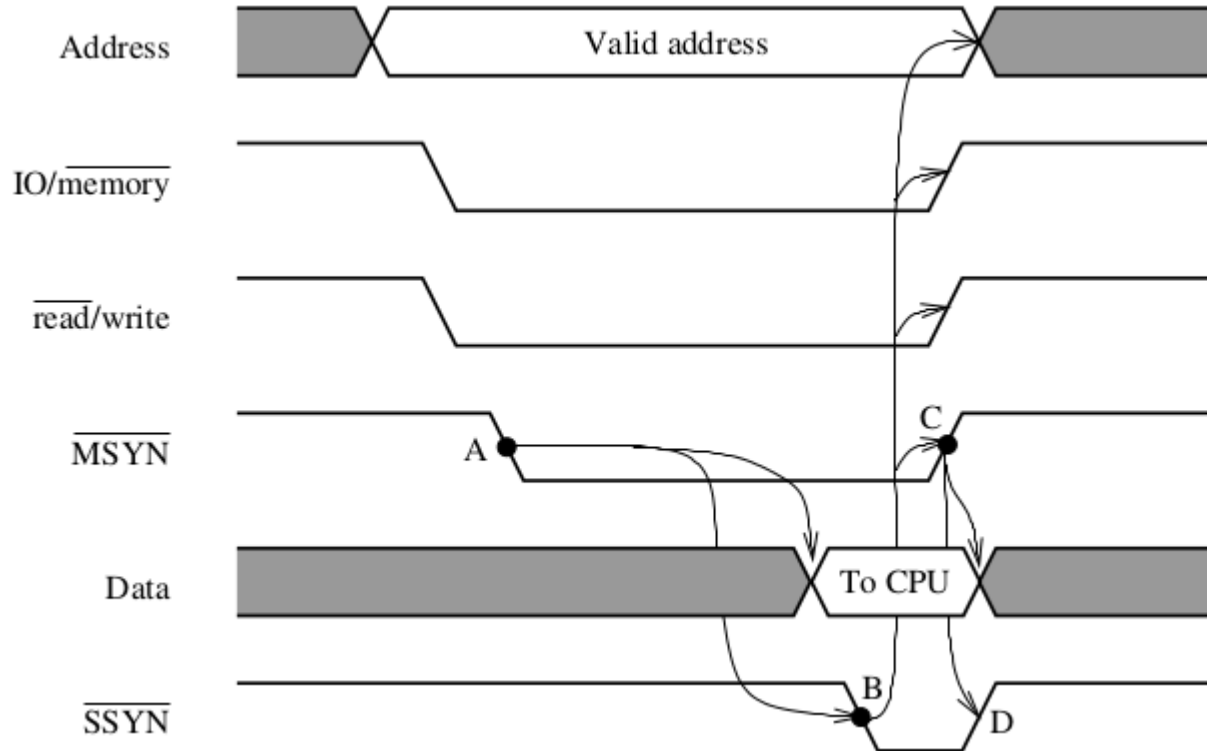
- Jednostranně – přenos nezahrnuje potvrzovací signál (méně časté)
- Oboustranně – každý přenos je potvrzen nastavením k tomu určeného signálu (tzv. **korespondenční protokol**, handshake)

Příklad: zápis do paměti přes synchronní sběrnici



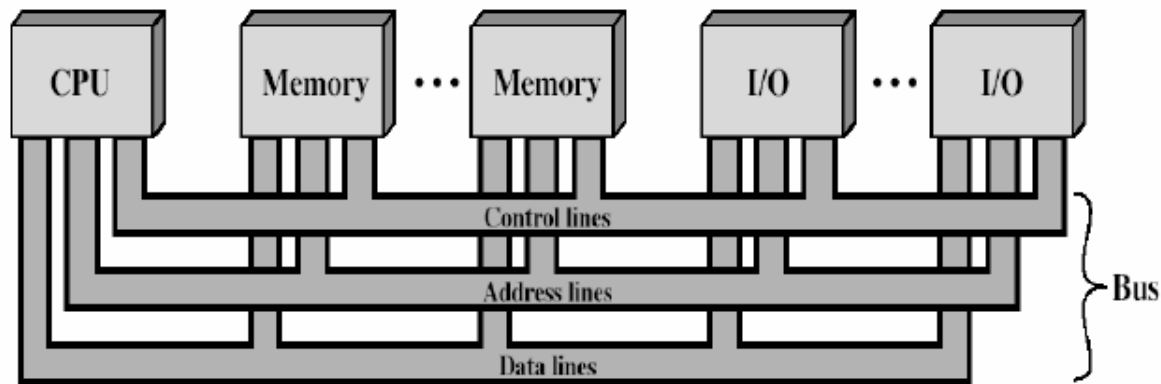
- Děje na sběrnici (změny hodnot signálů) jsou řízeny cykly hodin
- Volba zápisu do paměti je nastavena aktivací signálu memory (jiná možnost: zápis do periférie – izolovaný IO, viz dále)
- Nastavení čtení zápisu aktivací signálu write
- Je-li paměť připravena (signál ready), CPU vystaví data, která jsou zapsána s nástupnou hranou signálu Clock

Příklad: čtení z paměti na asynchronní sběrnici s potvrzováním (handshake)



- Potvrzování je realizováno signály MSYN a SSYN; podle toho, kterou stranou je operace vyvolána, je aktivován jeden z těchto signálů a druhý po vykonání operace slouží k potvrzení.
- Postup v příkladu: (1) master nastaví platnou adresu a aktivuje signály memory a read, (2) aktivace MSYN zahajuje činnost paměti, (3) paměť informuje pomocí SSYN dostupnost dat na sběrnici, (4) po přečtení dat procesor deaktivuje MSYN, což je potvrzení paměti o přečtení.

Základní uspořádání počítače s jedinou sběrnici



- Řídicí signály (někdy též nazývané řídicí sběrnice) typicky zahrnují:
 - Reset (systémové nulování)
 - Clock (systémová synchronizace)
 - Memory Write (zápis do paměti)
 - Memory Read (čtení obsahu paměti)
 - I/O Write (zápis do registru PZ)
 - I/O Read (čtení obsahu registru PZ)
 - Transfer ACK (potvrzení příjmu)
 - Bus Request (žádost o přidělení sběrnice)
 - Bus Grant (přidělení sběrnice)
 - Interrupt Request (žádost o přerušení)
 - Interrupt ACK (potvrzení žádosti o přerušení)
 - a další

Způsoby rozhodování o přidělení sběrnice při současné žádosti několika jednotek

Rozhodování (arbitrace):

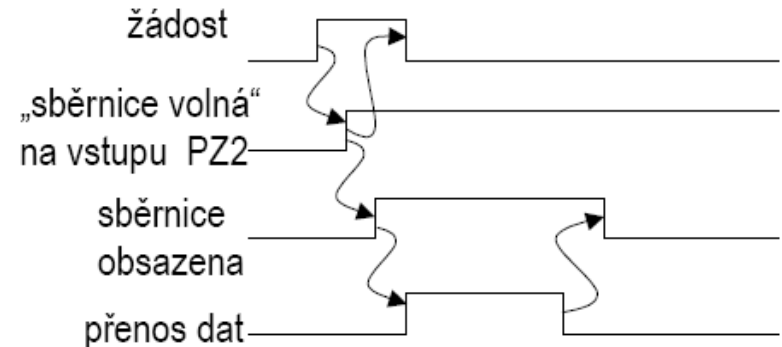
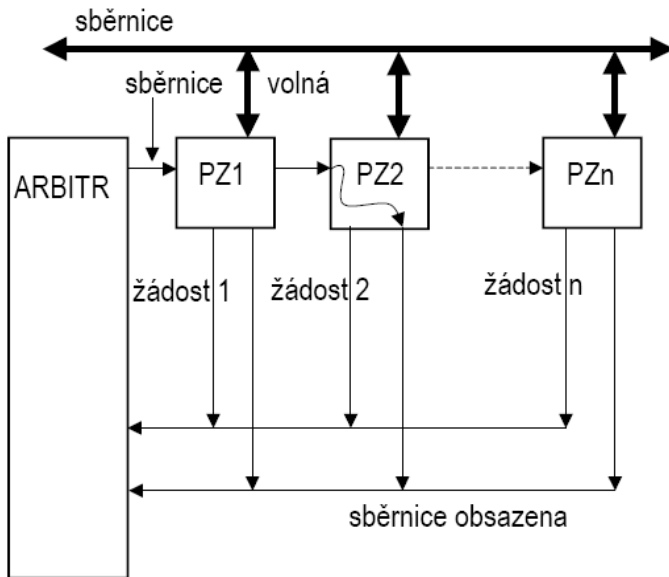
Centralizované vs. decentralizované

Prioritní vs. spravedlivé

U *centrálního řízení* je použita centrální *rozhodovací jednotka* (**arbitr**), která rozhoduje buď *podle priority*, nebo *spravedlivě*, např. podle pořadí vzniku žádostí (udržuje se fronta požadavků), cyklicky (pomocí centrálního čítače), nebo náhodně.

Prioritní dekodér je základem centrálního rozhodovacího členu, který po vyhodnocení priority žádostí oznámí jednotce s nejvyšší prioritou, že získala sběrnici. Vlastní mechanismus oznámení může využívat vyzývání (polling), kdy se na adresovou, datovou, nebo rozhodovací (vyzývací) sběrnici vyšle číslo vítězné jednotky. Jednotka, která zjistí svoje číslo, může zahájit přenos po sběrnici.

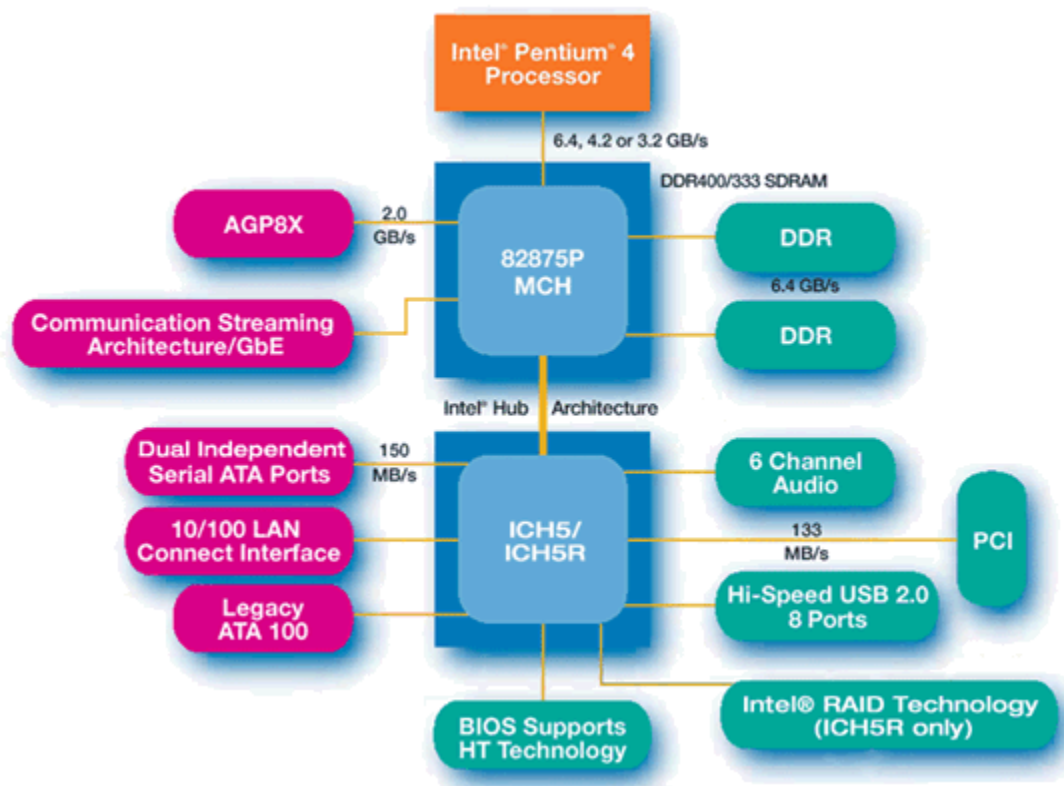
Příklad: prioritní linka



- Jednotlivá zařízení vysílají signály „žádost x“ do společného vodiče.
- Arbitr odpoví vysláním signálu „sběrnice volná“, který je jednotlivými zařízeními postupně přijímán a vyhodnocen.
- Zařízení, které vyslalo „žádost“, zablokuje odeslání signálu „sběrnice volná“ do následujícího PZ a vyšle signál „sběrnice obsazena“.
- Jakmile je ukončen přenos dat, signál „sběrnice obsazena“ je shozen.
- Prioritní systém je uplatněn pořadím zařízení na kabelu „sběrnice volná“ – zařízení, která jsou blíže arbitra, mají vyšší prioritu.

Sběrníková hierarchie ve starších PC (~ r. 2000)

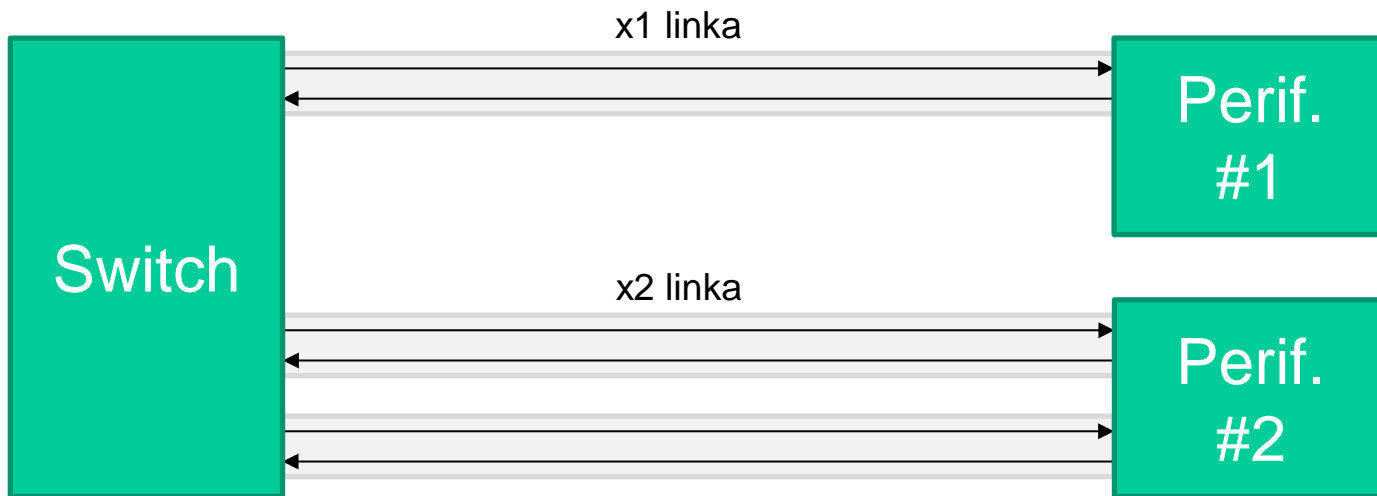
Typickým rysem byla kombinace starší univerzální sběrnice **PCI** a dedikované **AGP** (Accelerated Graphics Port). Jednalo se o paralelní sběrnice s omezenou propustností. Ještě dříve byly používány architektury s PCI a asynchronní sběrnicí ISA. Dle propustnosti různých sběrnic byly tyto využívány k připojování zařízení podle jejich výkonnostních požadavků.



Architektura moderních PC s PCI-Express

V moderních systémech byla sběrnice **PCI(-X)** nahrazena sběrnicí **PCI-Express**. Přechází se od paralelního na sériový přenos dat.

- **PCI** je vlastně jedna sběrnice se sdílenými adresovým, datovým a řídicími signály.
 - Rychlost přenosu je dána nejpomalejším členem.
 - Jen jedno zařízení může komunikovat v jeden okamžik jedním směrem.
 - Celá sběrnice je arbitrovaná.
- **PCIe** - je založena na point-to-point spojeních (linky) mezi zařízením a kořenem (root complex)
 - Mezi dvěma zařízeními je možné komunikovat pomocí plně duplexní komunikace. Ostatní zařízení mohou spolu během této komunikace komunikovat také.
 - Méně kritické zařízení jsou spojeny přes křížový přepínač (switch).



Architektura moderních PC s PCI-Express



PCI-X (64bit)

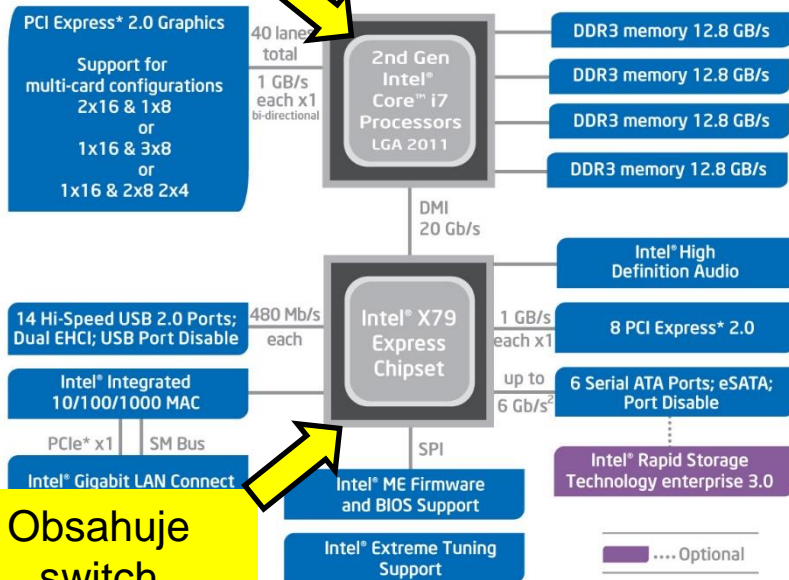
PCI Express x8

PCI

..

PCI Express x16

Root complex
(součást CPU)



Obsahuje
switch

PCI Express je hlavní komunikační sběrnici v moderních systémech, kde kritické prvky počítače (GPU a operační paměť) jsou připojeny přímo ke kořenové části (Root Complex) a méně kritické prvky jsou připojeny přes přepínač.

Poznámky: Severní můstek bývá nyní součástí procesoru; jižní můstek typicky nalezneme na motherboardu. Pro komunikaci mezi můstkami se používá proprietární rychlá sběrnice.

¹theoretical maximum bandwidth
²All SATA ports capable of 3 Gb/s. 2 ports capable of 6 Gb/s.
Intel® X79 Express Chipset Block Diagram

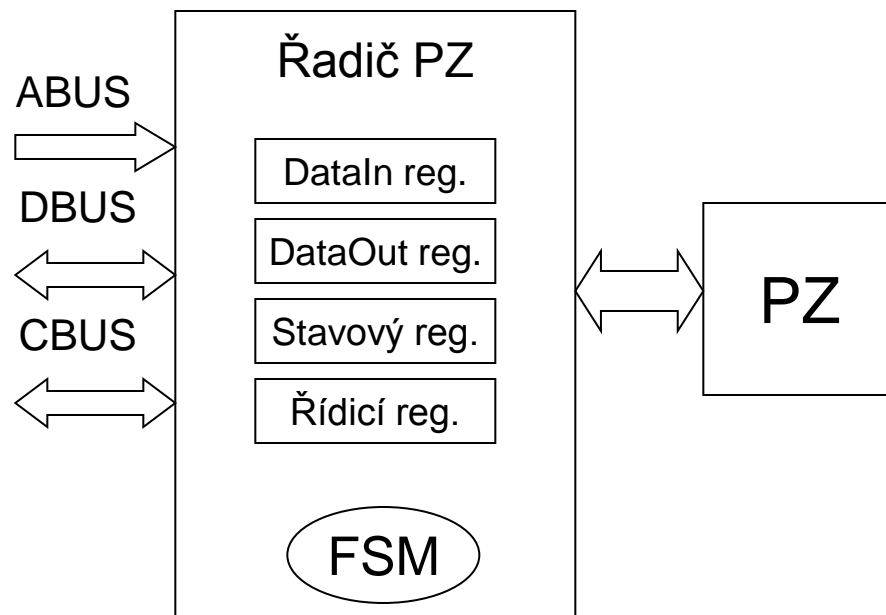
Způsoby připojování a řízení periferních zařízení

- Procesor komunikuje s registry řadiče periferního zařízení.
- Používá k tomu buď instrukce pro práci s pamětí nebo speciální instrukce pro ovládání V/V.
- Poznámky:
 - Např. není možné přímo realizovat přenos mezi registrem řadiče a pamětí – adresy obou by musely být ve stejném okamžiku vystaveny na sběrnici.
 - Přímá komunikace mezi periferií a pamětí (bez účasti procesoru) je možná zavedením dalších obvodů – např. DMA.

Řadič periferního zařízení

Funkce řadiče:

komunikace s CPU,
vyvolání přerušení,
řízení a časování operací PZ,
realizuje vyrovnávací paměť,
detekuje a reportuje poruchu



Registry jsou programovatelné ze strany procesoru.

Datový registr - přes něj se přenášejí data.

Řídicí registr - určuje způsob provedení operace PZ.

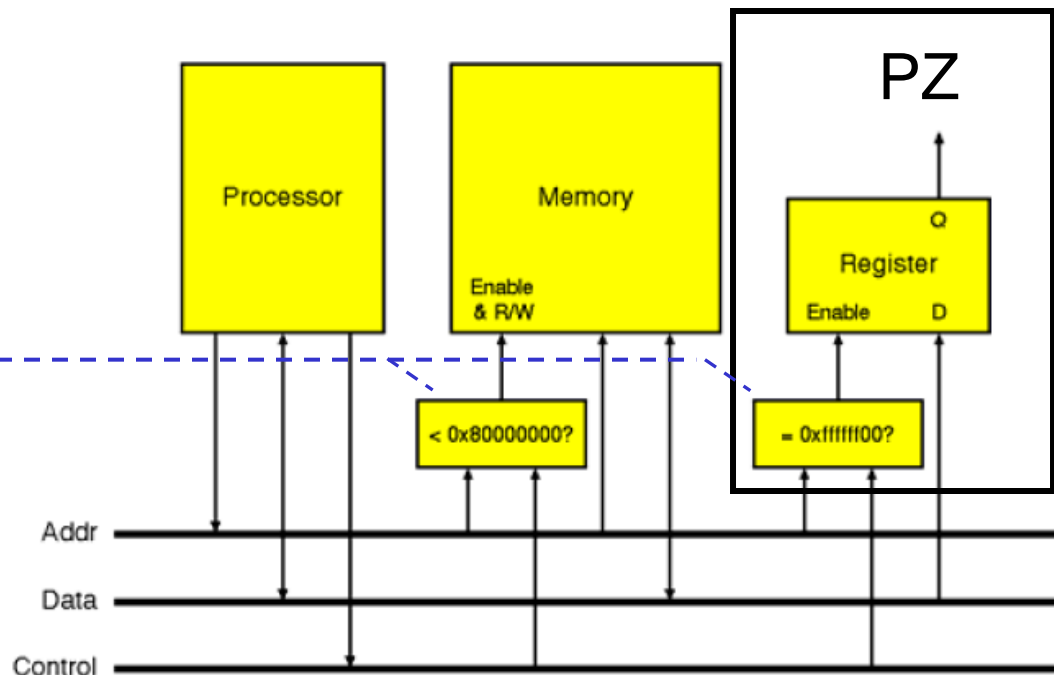
Stavový registr - stav řadiče a PZ.

Možnosti připojení PZ: 1) mapovaný V/V

- Registry PZ jsou namapovány do na určité adresy (hlavní) paměti.
- PZ a paměť **sdílí** stejný adresový prostor.
- Operace s PZ se provádí stejně jako operace s pamětí (instrukcemi pro čtení a zápis).

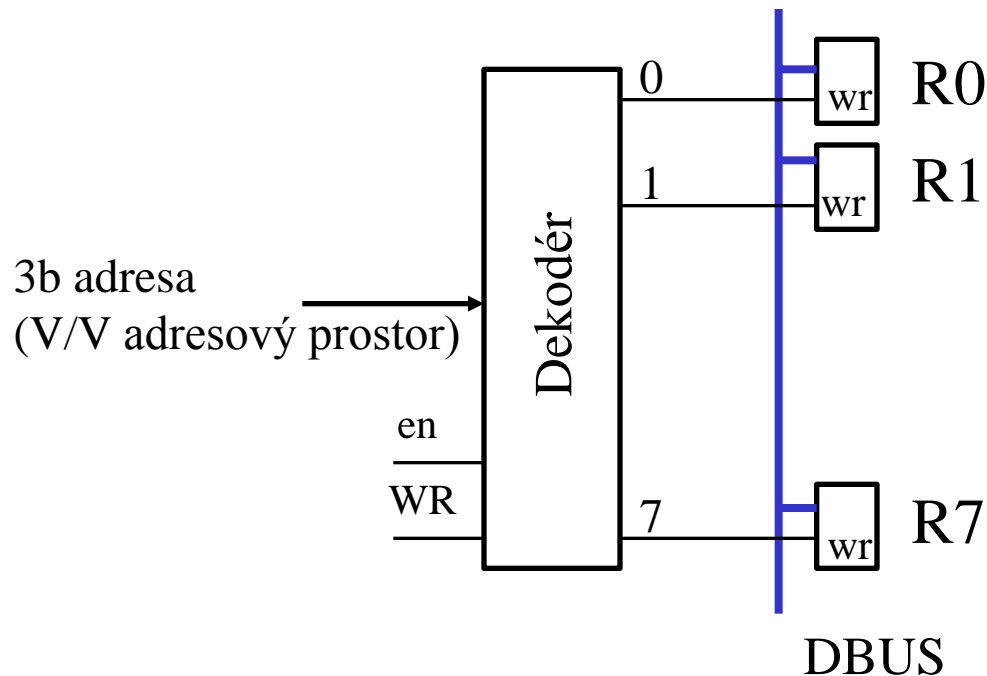
00000000	Up to 2^{31} bytes of Memory
...	
7fffffff	
80000000 ffffffef	Unused
fffffff00 fffffffef	Up to 256 I/O Ports

Řešení je založeno
na použití
adresového
dekodéru.



Možnosti připojení PZ: 2) izolovaný V/V

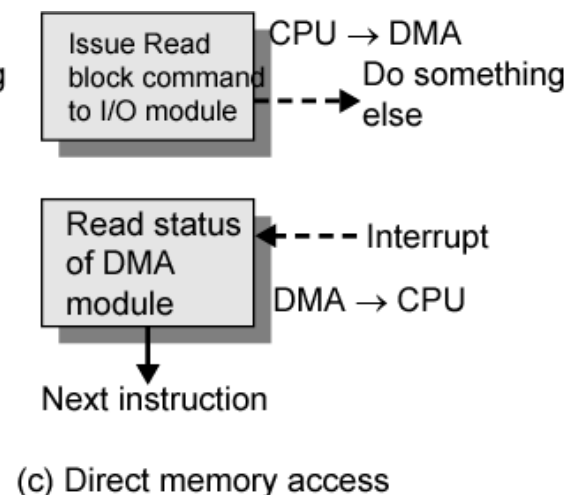
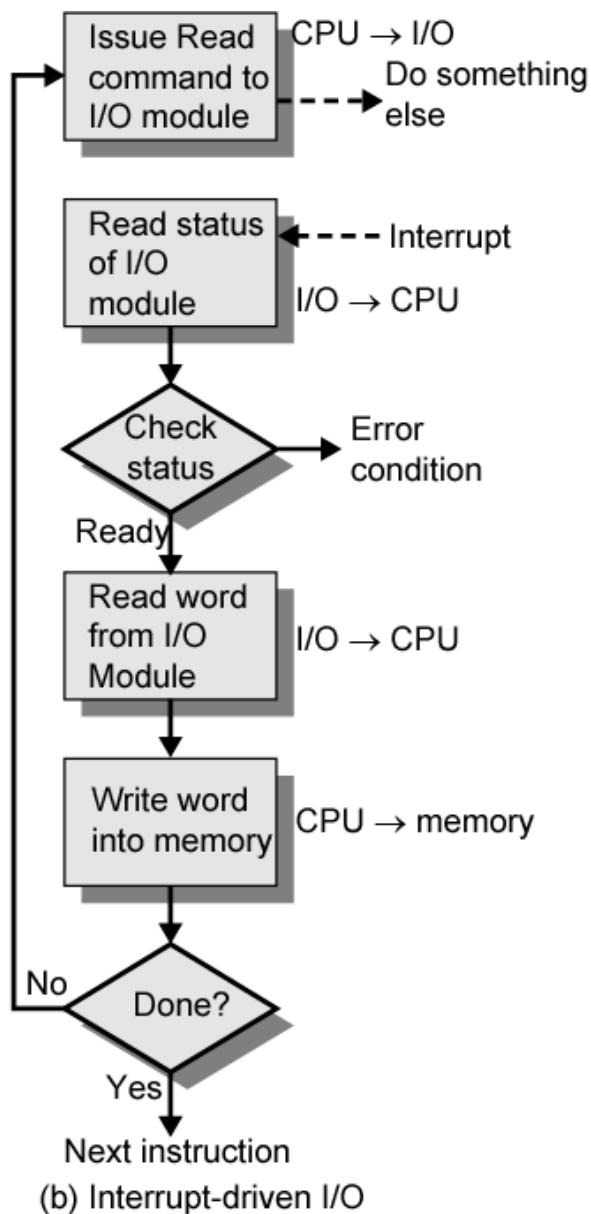
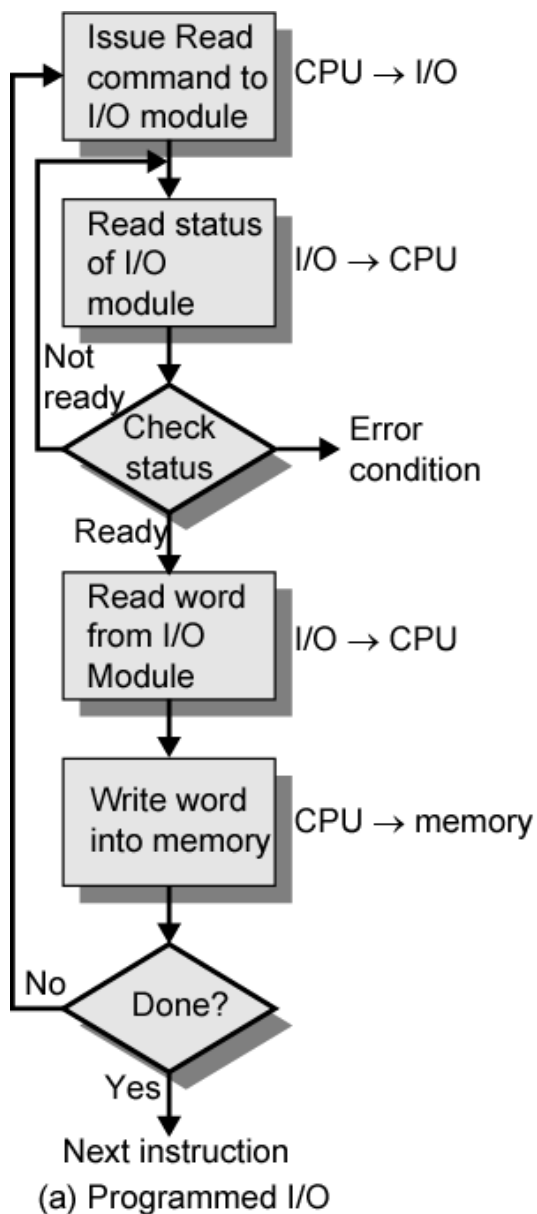
- Oddělení adresového prostoru paměti a periférií
 - Registry periferních zařízení mají svůj **vlastní adresový prostor** (uveden příklad, kdy je možné adresovat až 8 registrů v periferních zařízeních)
 - Operace s PZ se provede pomocí speciální instrukce **IN** a **OUT** (např. signál „zápis do registru“ (viz wr na obrázku) je odvozen od instrukce OUT).



Způsoby obsluhy periférií

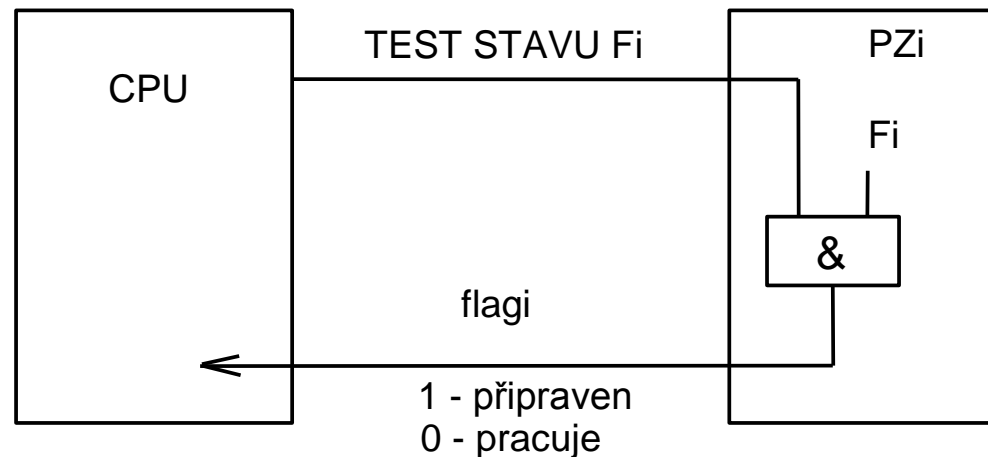
- Programový V/V
 - Procesor testuje ve smyčce stav všech PZ (stisknuta klávesa, přišla data ze sítě?), což je neefektivní.
- Využití přerušení
 - Pokud PZ potřebuje obsluhu vyvolá přerušení a procesor jej obslouží.
- Využití obvodů pro řízení blokových přenosů DMA (direct memory access)
 - Přenos větších objemů dat mezi PZ a pamětí zajistí DMA bez použití procesoru
- Využití IO procesorů
 - Zobecnění konceptu DMA

Porovnání přístupů (př: čtení dat z PZ do paměti)

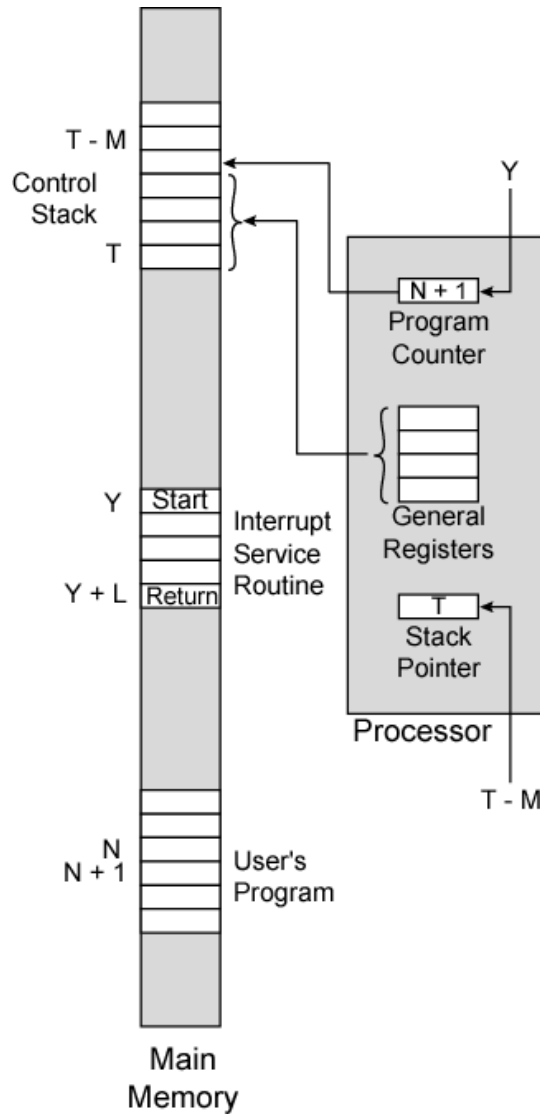


A. Programovaný V/V (polling)

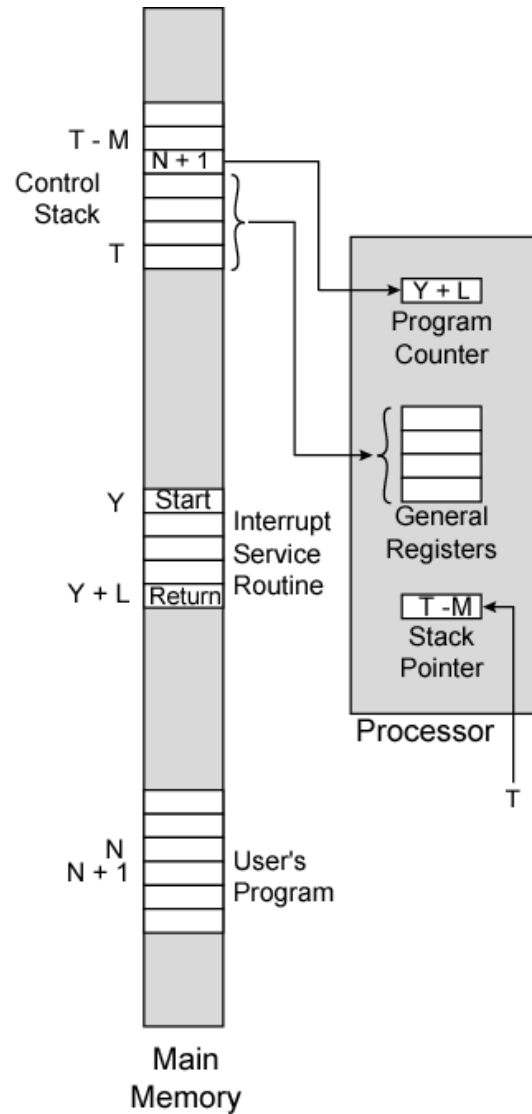
Procesor testuje ve smyčce stav PZ (byla stisknuta klávesa?). Pokud je možné provést obsluhu PZ (klávesa stisknuta), provede obsluhu (přečte kód znaku a zpracuje ho), jinak opět testuje ve smyčce stav PZ (byla stisknuta klávesa?). Je zřejmé, že čekací smyčka zatěžuje procesor, který pak nemůže provádět užitečnější práci. Rovněž sběrnice je zbytečně zatěžována.



B. Obsluha využívající přerušení: O obsluhu žádá PZ!!!



Přerušení programu po vykonání instrukce na adrese N.



Návrat z obsluhy přerušení

Př. Pokud byla stisknuta klávesa, procesor přeruší uživatelský program, uschová obsahy registrů a návratovou adresu na zásobník a obslouží klávesnici (rutina na adrese Y). Potom obnoví obsahy registrů a vrátí se do uživatelského programu.

Jak se řeší priorita, když dvě a více PZ žádají o obsluhu?

- Viz předchozí přednášky

C. Přenos DMA (Direct Memory Access)

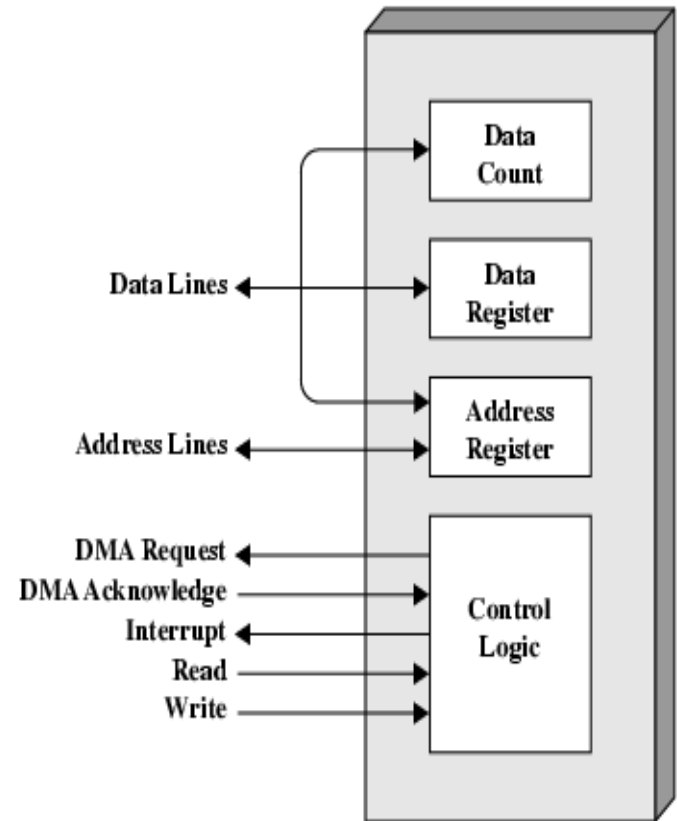
Koncept přerušení je nevýhodný, pokud je třeba přenášet větší objemy dat. Při obsluze přerušení se procesor podílí na datových přenosech, což ho zatěžuje => zaveden DMA.

Řadič DMA je speciální obvod zajišťující blokové přenosy mezi PZ a pamětí (popř. mezi pamětí a pamětí).

- CPU zadá požadavek obvodu DMA, tj. **adresu PZ**, **adresu v paměti** (ta se uloží do Address Register v DMA) a **počet slov** (ten se uloží do Data Count v DMA).
- DMA obvod provede přesun dat, slovo za slovem, bez zásahu CPU.

- data buď prochází přes řadič DMA
- nebo data neprochází přes řadič DMA

- CPU mezitím může vykonat další kód (je však omezen ve využívání sběrnice).
- Po ukončení přenosu (popř. při vzniku chyby) obvod DMA vyvolá přerušení.



Obvod řadiče DMA

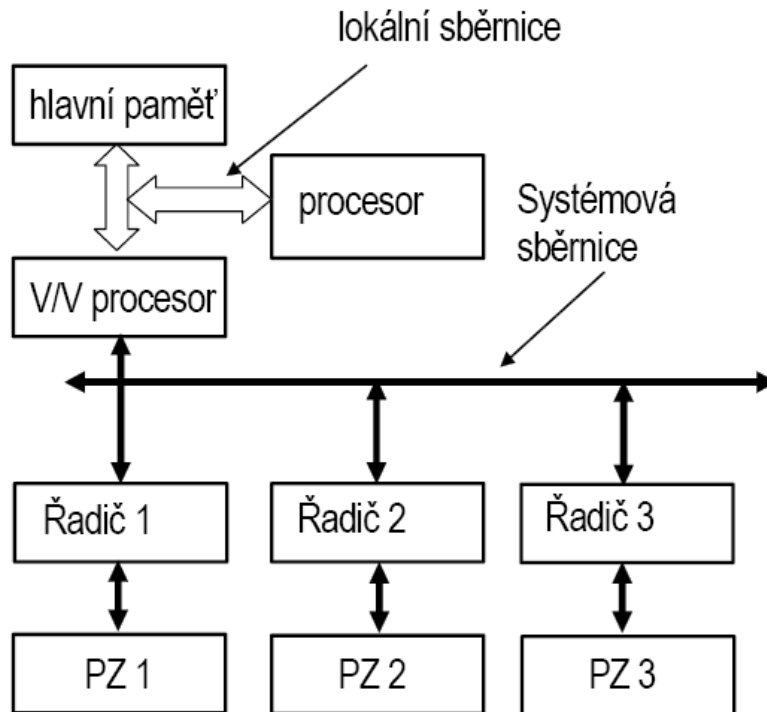
- typické komponenty a rozhraní
- je připojen ke sběrnici

D. IO procesory

Řadič DMA je v podstatě jednoduchý FSM s několika registry.

Rozšířením principu DMA se dospělo ke koncepci **IO procesoru**.

IO procesor (V/V procesor) je další procesor počítače, který má ale speciální instrukční soubor (zejména IO instrukce).



- Procesor nemá přístup k registrům řadičů PZ.
- IO procesor komunikuje s řadiči jednotným způsobem.
- Procesor zadá IO procesoru program, který má vykonat. Program je uložen v hlavní paměti.

USB

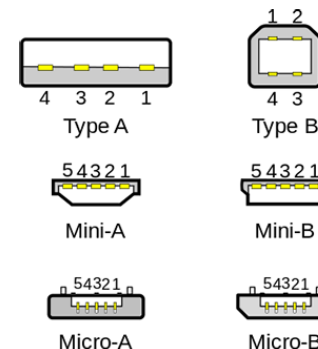
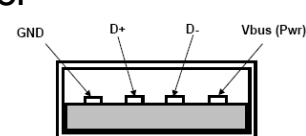
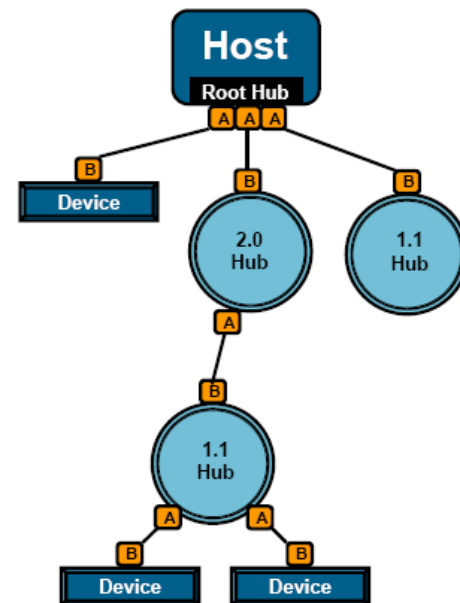
www.usb.org

- Universal Serial Bus – standard rozhraní pro připojování perif. zařízení, jehož cílem je především:
 - jednotný způsob připojování periférií a komunikace s nimi,
 - odstranit nutnost zavádění dalších konektorů pro připojení více PZ,
 - možností připojování a odpojování zařízení za běhu počítače,
 - Možnost napájení periferních zařízení přímo z rozhraní.
- Jedná se o specifikaci koncepce, hardwaru, elektrických a výkonových charakteristik a požadavků na komunikaci se zařízeními různých kategorií
 - ⇒ nutnost řešení automatické identifikace, konfigurace a zavedení ovladače zařízení po připojení k PC
 - ⇒ podpora různých typů datových přenosů (z pohledu latence, garantované propustnosti, odolnosti vůči chybám)



Struktura rozhraní USB

- Sběrnice **master-slaves** (host – devices, počítač – periferní zařízení)
 - Lze připojit až 127 zařízení (pomocí tzv. rozbočovačů – USB hubs)
 - USB hub se snaží komunikovat s kořenem nejvyšší možnou rychlostí, přestože periferie jsou pomalejší => nedochází k degradaci připojením pomalého zařízení vedle rychlého
 - **Master** řídí připojování a odpojování zařízení, jejich konfiguraci (tzv. **enumerace**) a dále veškerou komunikaci (přenos dat)
- Vývoj parametrů, režimů komunikace a fyzické realizace konektorů
- USB 2.0: Low-speed (do 1.5 Mbps)
Full-speed (do 12 Mbps)
High-speed (do 480 Mbps)
- USB 3.0: SuperSpeed (~ 5 Gbps)
- USB 3.1: Superspeed+ (~ 10 Gbps)



type **C**

Komunikace na rozhraní USB

- Veškerou činnost USB řídí hostitelský systém (**master**, typicky PC)

Na fyzické vrstvě je to **USB řadič** a HW periferního zařízení, na vyšších (logických vrstvách) je to **ovladač** podsystému USB, ovladač dané periferie a aplikační SW – viz obr. na další straně.

- Přenos informací z/do zařízení probíhá na tzv. **endpointech** (EP) – adresovatelných místech uvnitř zařízení sloužících pro vysílání/příjem řídicích informací či dat

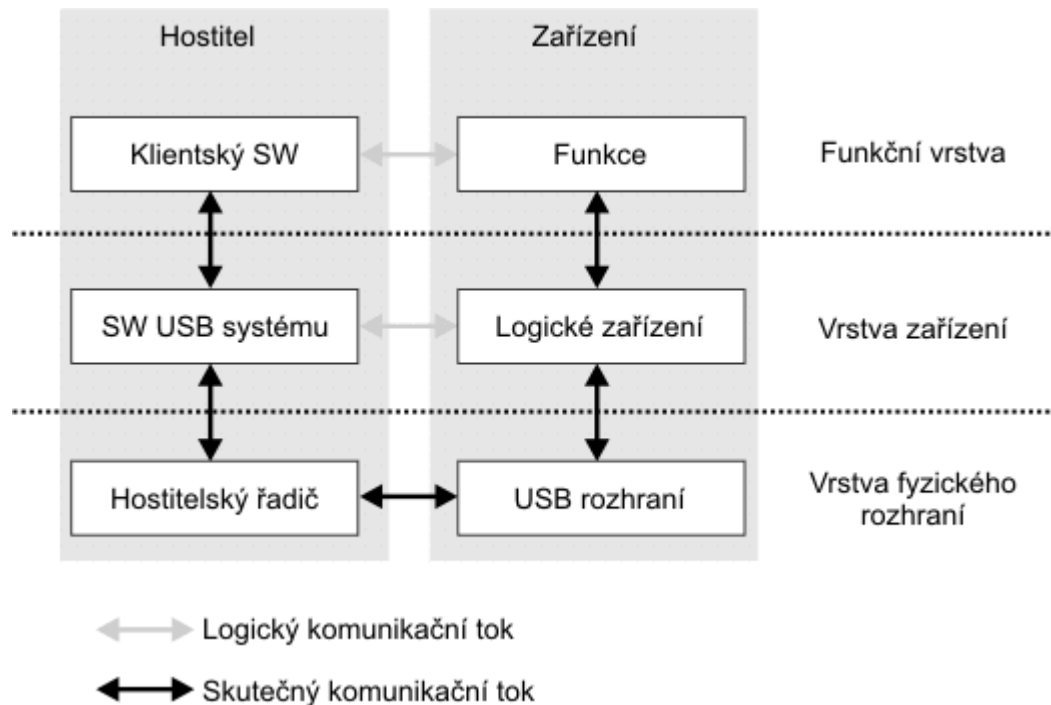
SW (ovladač) v počítači komunikuje přes tzv. **roury** (USB pipes), což jsou log. asociace s určitými endpointy zařízení.

- Základním prvkem komunikace na USB je **paket** – posloupnost bitů určité délky přesně daného formátu, která nese obsah komunikace (příkazy, data).

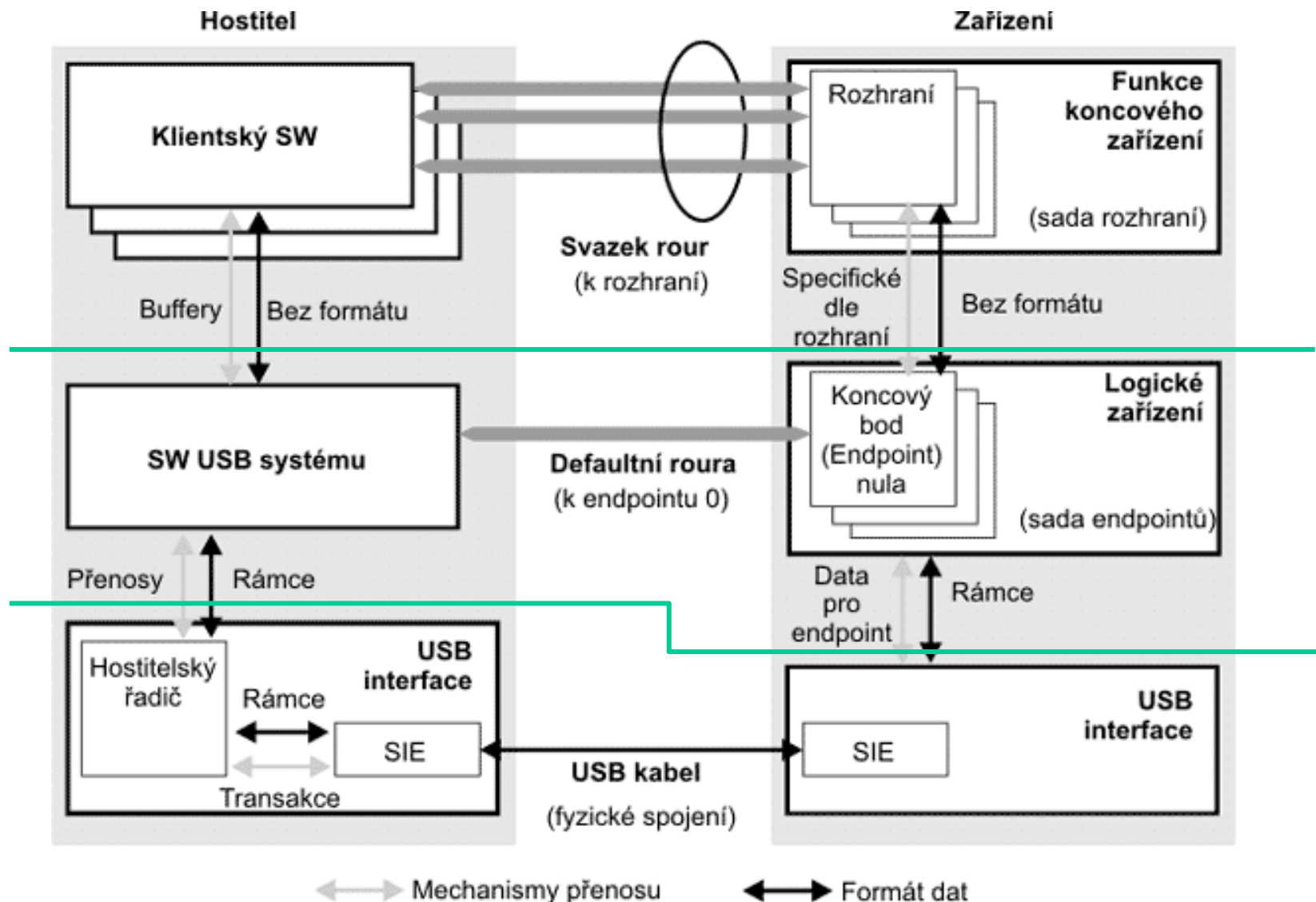
USB Hub skládá pouze pakety za sebe a přeposílá je (na nejvyšší možné rychlosti)

USB z pohledu vývojáře

- Zařízení jsou identifikována pomocí
 - VID a PID (výrobce a typ)
 - Zařízení specifikuje, jaké endpointy má k dispozici
- Komunikace je založena na vrstvách
 - Záleží, na jaké úrovni aplikace komunikaci zprostředkováváme
 - Typicky nás zajímají vyšší vrstvy
 - Nižší vrstvy řeší knihovny, ovladače a integrované obvody



Princip rozhraní USB: endpointy, roury

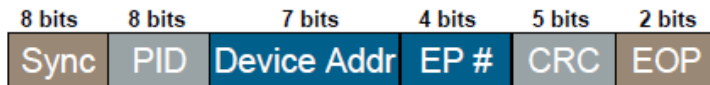


Realizace funkce USB zařízení

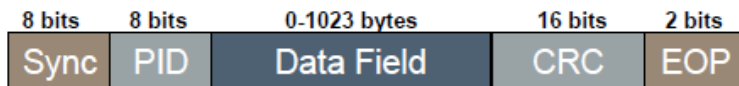
<https://cours.etsmtl.ca/ele784/labo/usb-in-a-nutshell.pdf>

- **Přenos** (transfer) je proces zajišťující kompletní vykonání určitého příkazu od počítače USB zařízením. Je tvořen posloupností **transakcí** (přenos informace na endpointu). Transakce sestává z tzv. **paketů**, přesně definovaných struktur, jejichž formát a význam je dán specifikací USB.
- Přenos sestává obecně ze třech částí: zahájení (**setup**), komunikace (**data**), potvrzení (**status**) – viz příklad dále
- Různé typy paketů

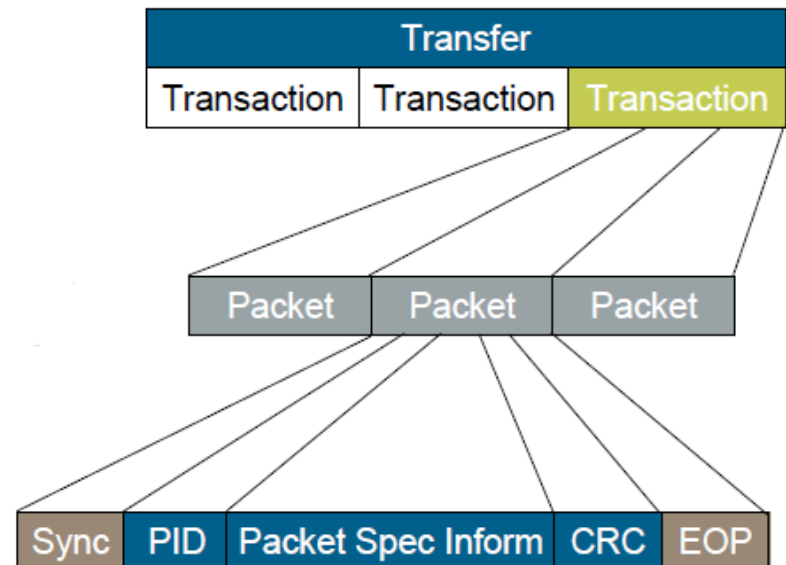
Token Packet



Data Packet



Handshake/Special Packet



Přenos na EP0 – příklad zpracování žádosti GET_DESCRIPTOR od PC

Setup stage

1. Setup Token	Sync	PID	ADDR	ENDP	CRC5	EOP	Address & Endpoint Number
2. Data0 Packet	Sync	PID	Data0		CRC16	EOP	Device Descriptor Request
3. Ack Handshake	Sync	PID	EOP				Device Ack. Setup Packet

Data stage

1. In Token	Sync	PID	ADDR	ENDP	CRC5	EOP	Address & Endpoint Number
2. Data1 Packet	Sync	PID	Data1		CRC16	EOP	First 8 bytes of Device Descriptor
3. Ack Handshake	Sync	PID	EOP				Host Acknowledges Packet

1. In Token	Sync	PID	ADDR	ENDP	CRC5	EOP	Address & Endpoint Number
2. Data0 Packet	Sync	PID	Data0		CRC16	EOP	Second 8 bytes of Device Desc
3. Ack Handshake	Sync	PID	EOP				Host Acknowledges Packet

1. In Token	Sync	PID	ADDR	ENDP	CRC5	EOP	Address & Endpoint Number
2. Data1/0 Packet	Sync	PID	Data0/1		CRC16	EOP	Last 8 bytes of Device Descriptor
3. Ack Handshake	Sync	PID	EOP				Host Acknowledges Packet

Status stage

1. Out Token	Sync	PID	ADDR	ENDP	CRC5	EOP	Address & Endpoint Number
2. Data1 Packet	Sync	PID	Data1		CRC16	EOP	Zero Length Packet
3. Ack Handshake	Sync	PID	EOP				Function Ack. Entire Transactions

Literatura

- Drábek, V.: Výstavba počítačů. Skriptum VUT Brno, 1995
- S.P.Dandamudi: Fundamentals of Computer Organization and Design. Springer, 2002
- Kotásek, Z.: Periferní zařízení. Texty k přednáškám. FIT VUT Brno, 2006
- Stallings, W.: Computer Organization and Architecture, 7th ed., Pearson Prentice Hall, 2006
- PCI Express – An Overview of the PCI Express Standard, dostupné na: <http://www.ni.com/white-paper/3767/en/>
- USB in a Nutshell, dostupné na: <https://cours.etsmtl.ca/ele784/labo/usb-in-a-nutshell.pdf>
- <http://cs.wikipedia.org/wiki/Sběrnice>