

Principy počítačů a operačních systémů

Architektura počítačového systému

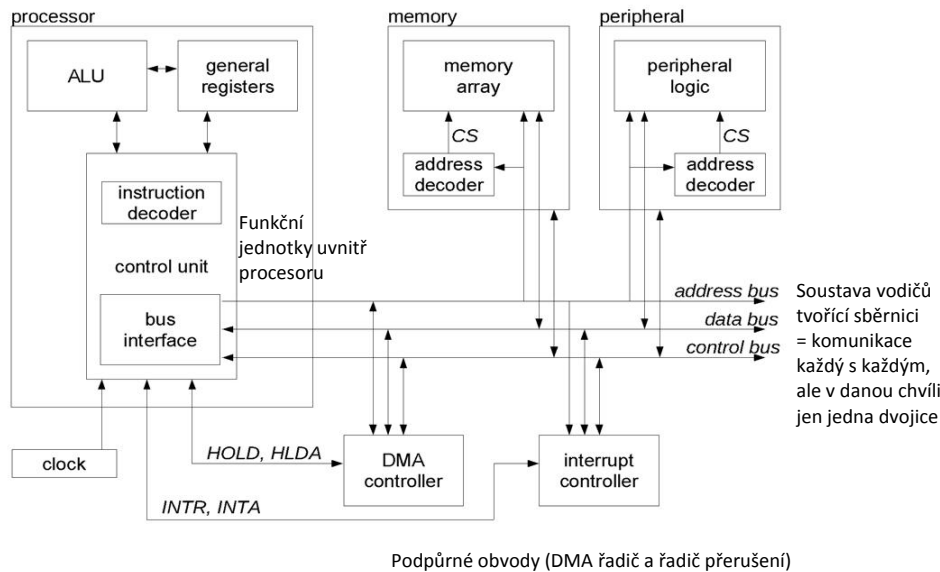
Zimní semestr 2007/2008

Architektura počítačového systému

Uspořádání na úrovni systému

- funkční bloky
 - procesory, paměti
 - periferní zařízení
- propojovací systém (propojovací systém je třeba sběrnice)
 - přenos dat v rámci systému a mimo něj
- podpůrné obvody
 - přidělování prostředků
 - přerušení a přímý přístup do paměti

Architektura PC (přelom 80. a 90. let)



Propojovací systém

Dvoubodové spoje

- **přímé spojení (port)** Velice specifické, umožňuje komunikační zkratky, víme s kým komunikujeme, umožňuje rychlé, nepřilíš flexibilní spojení;
 - není nutné adresovat
- **křížový přepínač (crossbar switch)** Dá možnost propojit každého s každým pomocí dvoubodového systému, ale funguje jako telefonní ústředna; je omezení co jde spojit (např. jen spojí jednu dvojici periferií)
- **propojovací síť (switch fabric)**
 - topologie Např. na zajištění komunikaci dvou dvojic...

Vícebodové spoje

- **účastníci sdílejí přenosové médium**
 - broadcast: 1 vysílač, více příjemců
- **sběrnice: sada sdílených vodičů**
 - adresové/datové vodiče, řídicí vodiče

Na sdílené vodiče jsou připojeni všichni účastníci; omezení je, že může být jen jeden vysílač; Např. na jednom vodiči běží data, na druhém řídicí signály...

Sběrníkové systémy

Výhody Relativně jednoduché řešení.

- nízká cena a flexibilita
- lehké zvládnutí složitosti systému

Nevýhody

- potenciální úzké hrdlo v systému Pokud není dobře dimenzovaná (např. ISA přestala vyhovovat)
- rychlost omezena délkou a počtem zařízení Fyzická délka drátu..
- potřeba spojit různorodá zařízení
 - různé rychlosti, různé objemy přenášených dat

Typicky je navržena obecně, nějakým protokolem, a jakékoliv zařízení, které splní protokol může být připojeno.. Na jednu stranu výhoda, ale zase na jednu sběrnici máme různé rychlosti a různé objemy přenášených dat, je třeba dobrého návrhu, aby nic nezpomalovala...

Sběrníkové systémy

Princip fungování

- vystavení adresy cílového zařízení (address) Iniciátor spojení vystaví adresu s kým chce komunikovat; pak řekneme co chceme
- potvrzení cíle a výběr typu přenosu (control)
- přenos dat (data) Po potvrzení dojde k samotnému přenosu dat ^^ Poté, co periferie zjistí, že s ní někdo chce hovořit, tak zareaguje, pokud bude požadavek timeoutován, žádná taková periferie asi neexistuje

Fyzická charakteristika sběrnice

- vodiče, napěťové úrovně, frekvence hodin, ...

Logická charakteristika sběrnice

^ pokud je synchronní a řízená hodinovým signálem

- sběrníkový protokol Ten říká jakým způsobem probíhá cyklus sběrnice ->vystavení adresy, potvrzení, přenos dat.. Říká kolik taktů musí trvat odpověď, do kdy se musí zahájit datový přenos... Poměrně komplikovaný dokument říkající co všechno se může stát a jak případně řešit

Sběrníkové systémy

Sběrníkový protokol

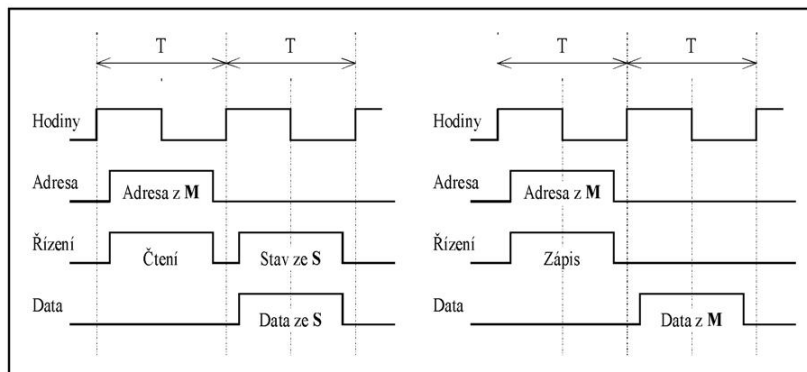
- definuje povolený průběh transakcí na sběrnici
 - logický význam signálů na vodičích Vodiče mohou být sdíleny, a podle času se určuje k čemu zrovna slouží
 - popis pomocí stavových a časových diagramů
- synchronní vs. asynchronní protokol
- centralizované vs. distribuované přidělování
 - o přidělení sběrnice rozhoduje centrální arbitr resp. zařízení v rámci distribuovaného protokolu

Rozděluje, aby právě jeden účastník byl vysílač
 Je potřeba systém přidělování sběrnice; o rozhodování se stará tzv. arbitr

Asynchronní - veškerá zařízení reagují okamžitě; žádný synchronizační element ve formě taktu; velice striktní požadavky načasování
Synchronní - na jednom vodiči nám běží hodinový signál; a s ním běží všechny operace; výhodou je, že se celá řada věcí dá dělat implicitně.. Každé zařízení si sleduje takt; S požadavkem na rychlost jsou téměř všechny synchronní
 U procesorů čistě synchronní

Sběrníkové systémy

Synchronní čtení/zápis



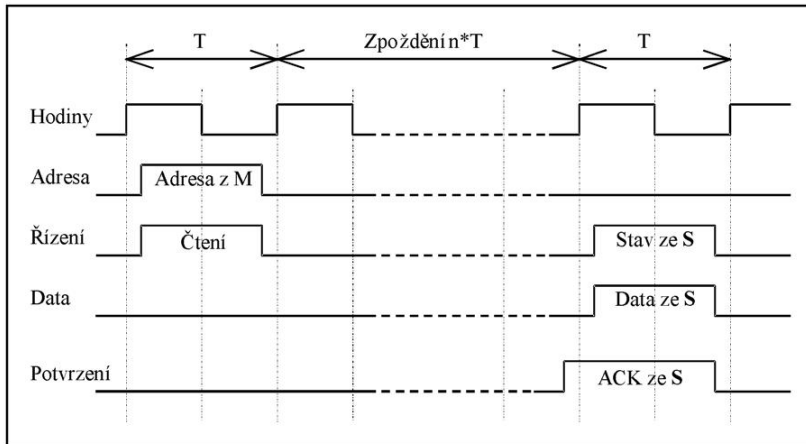
T... takt
 Hodiny, 1,0...

Zde vystavil data
 současně s
 odpovědí svého
 stavu

Sběrníkové systémy

Synchronní čtení s čekáním

Wait state



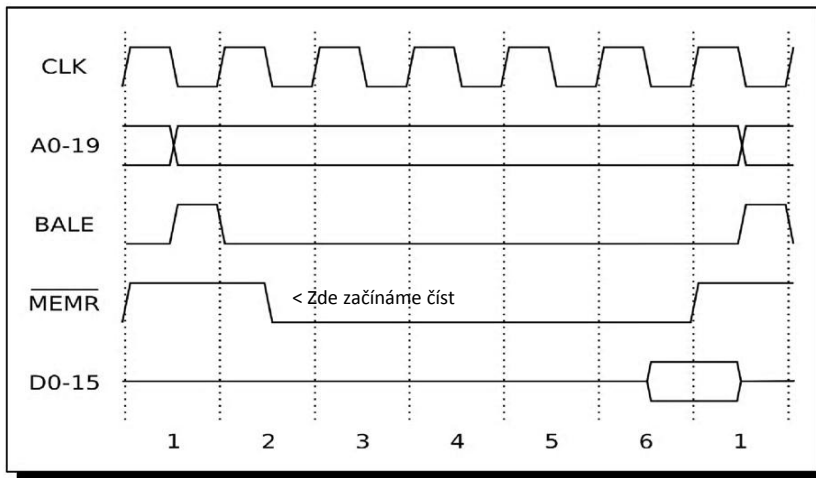
Zde se vlastně promrhá několik cyklů

V asynchronním by tam jen nebyl hodinový signál

Sběrníkové systémy

Křížení znamená změnu... zde, u ISA byla adresa vystavena celou dobu, u PCI už tomu tak není

Čtení z paměti na sběrnici ISA



Hodinový signálů

20 adresových vodičů

Bus adres - označuje zapamatování platné adresy

Řídící signál memory read .. Čára nad ním znamená, že je aktivní v signálu 0 a neaktivní při A

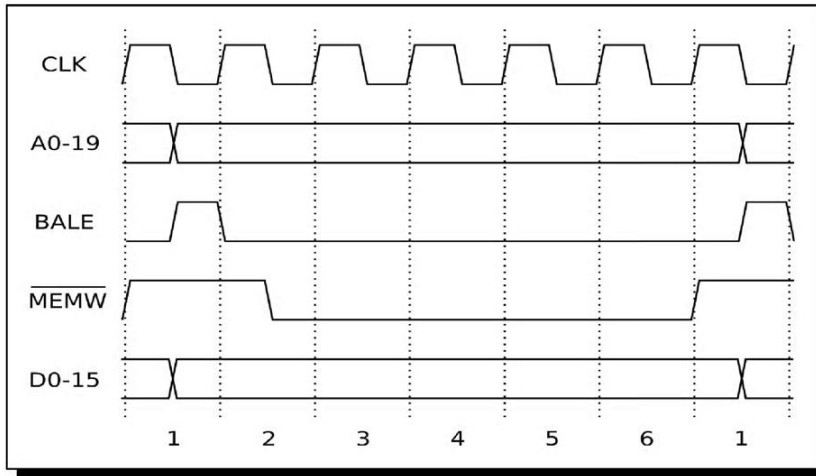
D - 16 datových vodičů

Tady jsou data vystavena na sběrnici

1 cyklus měl 6 taktů

Sběrníkové systémy

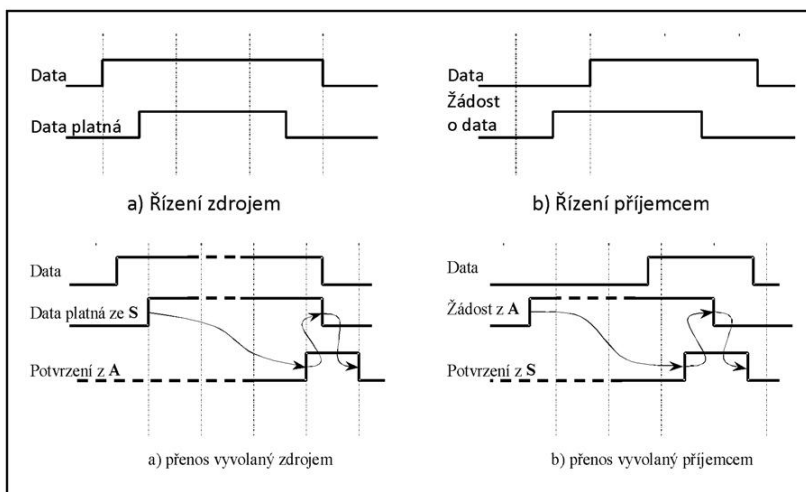
Zápis do paměti na sběrnici ISA



Sběrníkové systémy

Asynchronní čtení/zápis

Šipky definují návaznosti signálů na sebe



Data platná říká, kdy je možné je číst..proto prvně
vystavím data a pak dám signál data jsou platná
Až se vystaví signál potvrzení, tak se shodí signál data
platná a data

Sběrnice systémy

Centrální přidělování sběrnice

- fyzické uspořádání
 - samostatné žádosti, cyklické výzvy (polling)
 - prioritní zřetězení (daisy chain)
- řešení současných požadavků
 - náhodné, dle pořadí vzniku, prioritní

Distribuované přidělování sběrnice

- kolizní (CSMA/CD)
- token bus
- adresová priorita

Stanice mohou být uspořádány do kruhu, a právo posílání koluje mezi stanicemi

Všichni mohou sběrnici připojit na logickou 0, a když dáme 1, tak nic

Pokud vysílám adresu 0, tak je dominantní

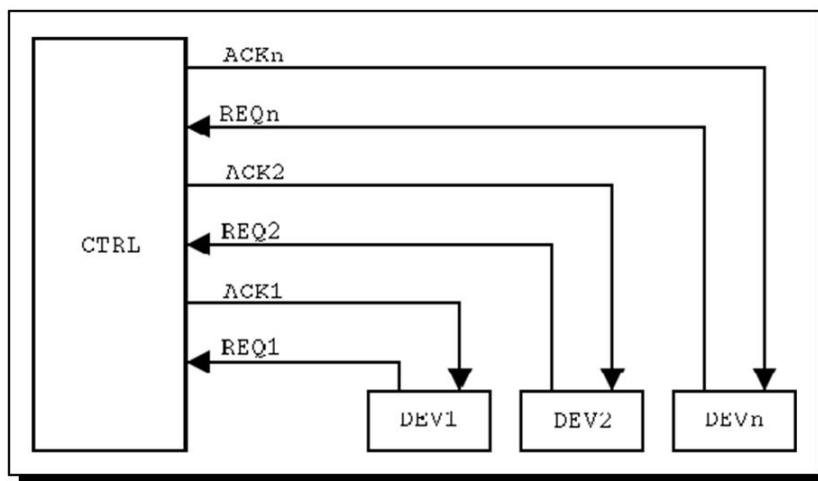
Není žádný centrální arbitr, který by rozhodoval, skoro na principu kdo dřív přijde, ten dřív bere

Když se chce vysílat, koukneme, jestli někdo nevysílá, pokud ne, tak chvíli počkáme a pak pošleme

Centrální přidělování sběrnice

Nezávislé žádosti

Toto schéma je použito u PCI sběrnice
Sběrnice PCI může mít maximálně 10 účastníků, takže to je proveditelné; u PCI se počítá i s odrazem signálu na konci, aby nebyly tak vysoké proudové špičky

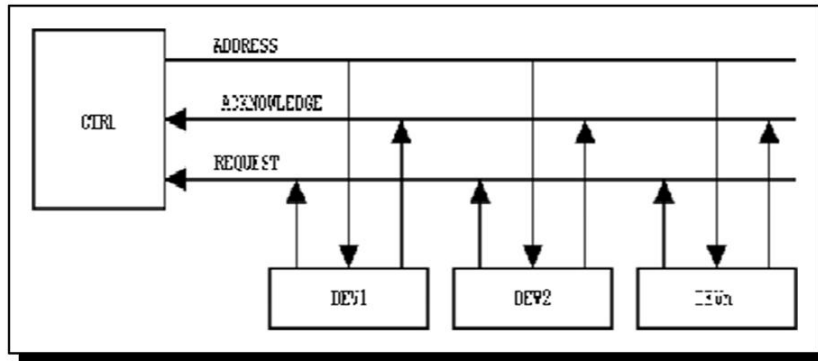


N různých zařízení je n spojení

A CTRL je arbitr, který rozhoduje, kdo dostane sběrnici

Centrální přidělování sběrnice

Cyklické výzvy (polling)



Pro ušetření vodičů se používá cyklických výzev.. Minimalizovat počty vývodů u integrovaných obvodů

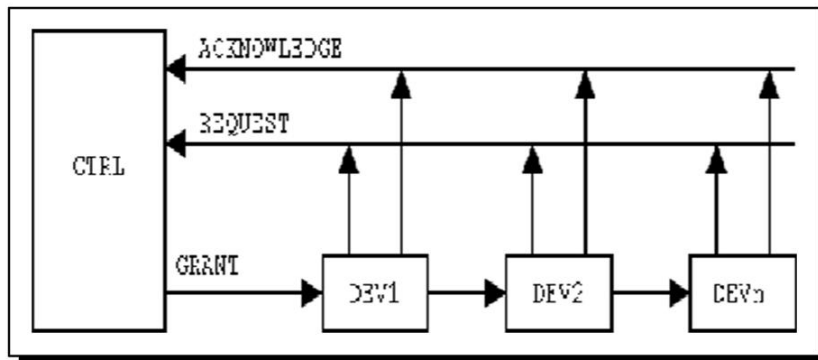
Všichni sdílejí vodiče a je tu ještě jeden vodič adresa, který ukazuje kdo zrovna vysílá

Kdo chce sběrnici, pošle na logickou 0 na request a arbitr pak posílá adresy až najde zařízení, které žádalo; pokud žádalo zařízení s malou prioritou, tak může trvat dlouho hledání; pokud se zařízení najde a ozve se, pošle Acc signál, a pak dostane sběrnici

Je třeba nějak ošetřit, aby jedno zařízení nebralo vždy prioritu

Centrální přidělování sběrnice

Prioritní zřetězení (daisy chain)



Každé zařízení má právě tři vývody,

Zařízení zažádá na request, sběrnice, pokud je volná, tak pošle signál grant, pošle jej prvnímu zařízení, které je připojeno, to pak pomocí jednoho logického obvodu buď sběrnici pošle Acknowledge, nebo pošle grant signál dál.. Časem se tak najde zařízení které žádalo.. Problém je tu fixní priority zařízení

Ovládání periferních zařízení

Periferie ovládány programem Periferie neovládá přímo procesor, ale program na procesoru

- vykonání specifické operace
 - výběr zařízení Vystavení adresy zařízení,
 - zápis příkazů specifických pro zařízení
 - čtení stavové informace specifické pro zařízení

U Intel procesorů se rozlišují dva různé adresové prostory, jeden pro paměť (4GB), druhý pro periferie - 64kB

Rozhraní paměť/procesor/periferie

- sdílený adresový prostor zařízení a paměti
 - zařízení mapována do paměti (memory-mapped I/O)
 - přístup pomocí běžných instrukcí typu load/store
- oddělený adresový prostor zařízení a paměti
 - přístup pomocí speciálních instrukcí typu in/out

Jako bychme četli z paměti, on ani nemusí vědět, jestli jsou to přímo periferie; většina zařízení je mapována do paměti, je to jednodušší

Komunikace procesoru se zařízením

Informace o stavu zařízení

- bud
- **polling** Program stále kontroluje periodicky stav zařízení z registru zařízení;
 - program periodicky kontroluje stav zařízení
 - značná režie v případě pomalých zařízení
- nebo
- **interrupt-driven I/O** Vstup/výstup přerušení; žádá procesor o přerušení toho co dělá, vlastně o pozornost;
 - zařízení signalizuje procesoru změnu stavu
 - obslužná rutina přerušení reaguje na stav zařízení

Přenos dat z/do zařízení (z/do paměti)

- s účastí procesoru (programmed I/O)
- bez účasti procesoru (DMA, bus mastering)

Zařízení přímo přistupuje do paměti bez pomoci procesoru

Program samotný obsahuje smyčku, která čte data.. Pokud chceme číst z disku velký rozsah, tak tou smyčkou jedeme dokud nemáme vše; procesor je zde jako prostředník

Podpora interrupt-driven I/O

Procesor

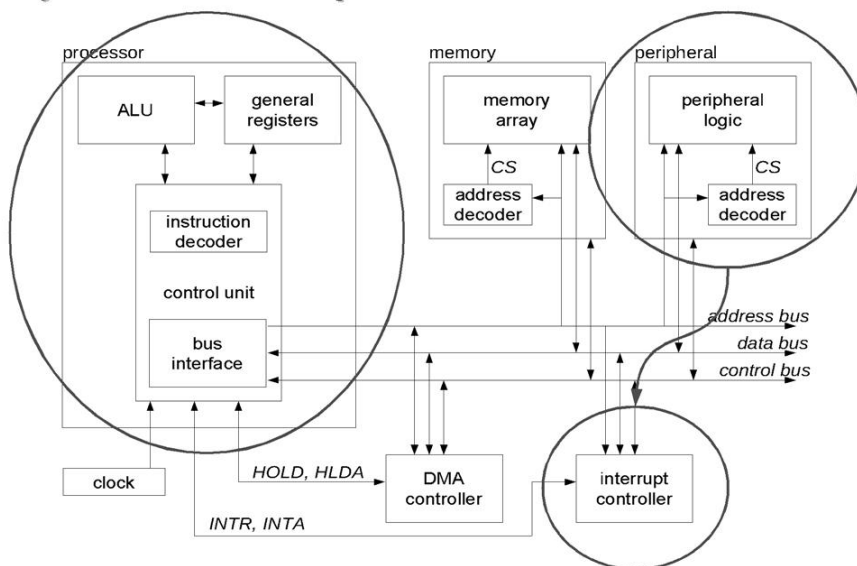
- signalizace požadavku na přerušeni
 - jeden nebo více signálů
- identifikace zdroje přerušeni Program, nebo procesor, musí být schopen poznat kdo žádá
 - pevná identifikace odvozená od signálových vodičů
 - dynamická identifikace
- výběr obslužné rutiny přerušeni Prvně je poskytuje BIOS, poté je nahradí operační systém
 - statická adresa vs. vektor přerušeni

System

- řadič přerušeni
 - více přerušovacích signálů, vyhodnocení priority

Dělá pořádek, aby se periferie "nepřekřikovali"
Pokud je zpracovávat priority

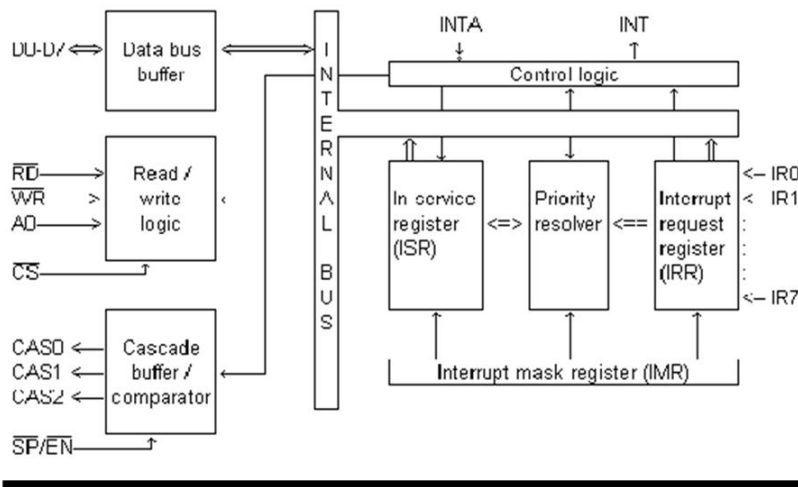
System s řadičem přerušeni



Příklad řadiče přerušení (i8259)

nezkouší

8259 internal block diagram



Dnes na deskách přítomno v southbridge (pozůstalost z dob dřívějších)

Vstup 8 signálů

Řadič se sám chová jako periferie

Zpracování přerušení s využitím i8259

Vyvolání a obsluha přerušení

Pokud periferie požaduje přerušení

1. Aktivován signál IR_x, nastaví bit v IRR
- poté 2. Řadič signalizuje přerušení procesoru pomocí INT
3. CPU potvrdí příjem INT pomocí #INTA Po dokončení rozpracované instrukce, řadič pošle #INTA, čím potvrdí příjem (nastaví ho na 0)
 - Řadič nastaví v ISR bit s nejvyšší prioritou, vynuluje odpovídající bit v IRR Tam jsou žádosti a říká co se bude zpracovávat
4. CPU podruhé signalizuje #INTA Dáno konvencí, že procesor zase potvrdí; to je signál pro řadič, aby poslal na sběrnici vektor přerušení
 - Řadič vyšle na sběrnici číslo vektoru přerušení
5. CPU zavolá obslužnou rutinu odpovídající vektoru Zjistí si adresu podle vektoru, který vynásobí, hrábne si do paměti pro adresu obslužné rutiny a skočí na tu adresu, která obsahuje nějakou funkci; pak se rutina vrátí z obsluhy
6. Ostatní přerušení blokována, dokud řadič neobdrží příkaz "Non-Specific EOI"

Na konci obsluhy je třeba řadiči říct, že už je obsluženo

Obslužná rutina musí schovat obsah registrů, přepnout na jiný zásobník, dá do fronty že je třeba přečíst data, po obslužení periferie se vrátí, obnoví zpět obsah registrů, vrátí zpět registry

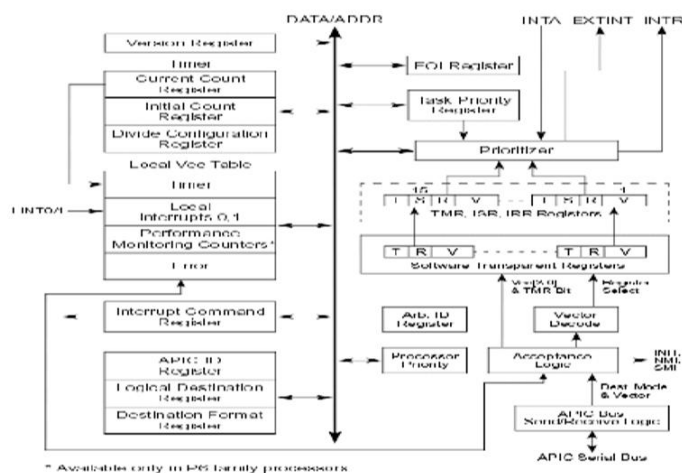
Zpracování přerušení na procesoru MIPS

Vyvolání a obsluha přerušení

1. Aktivován signál INT_x Cause... proč, EPC Exception program counter
2. Procesor nastaví registry Cause a EPC a skočí na obsluhu General Exception na adrese 0x80000180
 - dojde k uschování stavu procesoru
3. Obsluha GE zjistí příčinou a vyvolá specifickou obslužnou rutinu Program si z Cause přečte proč k přerušení došlo a zpracuje
 - pokud bylo přerušení požadováno více zdroji, jsou tyto obslouženy v pořadí určeném obslužnou rutinou
4. Při návratu je obnoven stav procesoru a skočí se na adresu v EPC

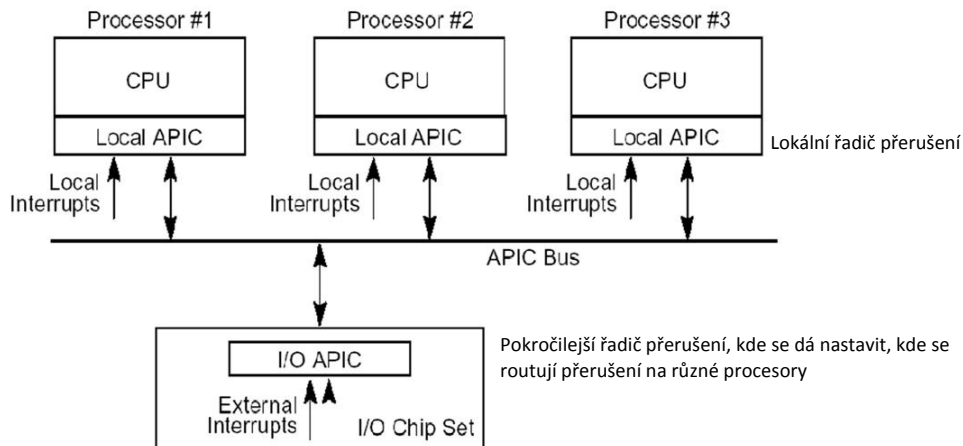
Moderní přerušovací subsystém

Řadič integrovaný na procesoru



Moderní přerušovací subsystém

Přerušování ve víceprocesorových systémech



Přenos dat po sběrnici

Za účasti procesoru

- přenos iniciován při změně stavu zařízení
 - polling, interrupt-driven I/O
- přenos realizován cyklem v programu
 - přečíst slovo z periferie/paměti
 - zapsat slovo do paměti/periferie
 - „programmed I/O“
- rychlý přenos, vytěžuje hlavní procesor
 - procesor by měl dělat něco užitečnějšího

Snahou je, aby procesor dělal něco užitečnějšího

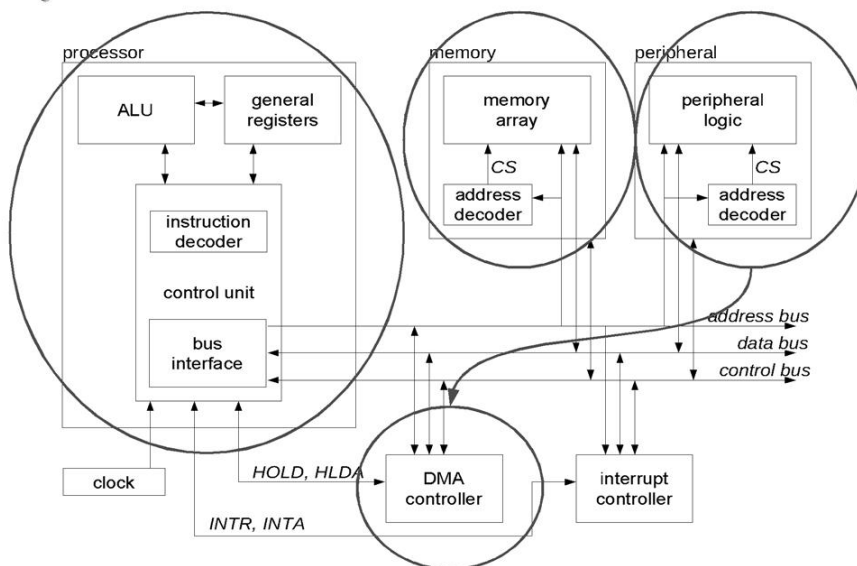
Přenos dat po sběrnici

Bez účasti procesoru

- přenos iniciován při změně stavu zařízení
- přenos realizován řadičem nebo zařízením
 - program musí pouze nastavit parametry přenosu
- přenos zařízení/paměť
 - direct memory access
 - řadič DMA, bus mastering
- přenos zařízení/zařízení
 - bus mastering Dnešní zařízení PCI

buď za využití řadiče DMA, nebo to periferie umí sama (tedy umí být BUS masterem)

System s řadičem DMA



Řadič DMA je obvod, který má schopnost řídit přenos po sběrnici; generuje adresy a generuje řídicí signály.. Dělá to, co normálně dělal procesor. Zároveň musí zajistit, aby procesor mu do toho "nehrabal". Dříve (když procesory neměli cache na instrukce, tak neměli co dělat) Když chce program zajistit, aby data přenášeli do periferie, musí tak naprogramovat periferii i řadič.. Kam se data mají uložit, a kolik toho budeme přenášet.

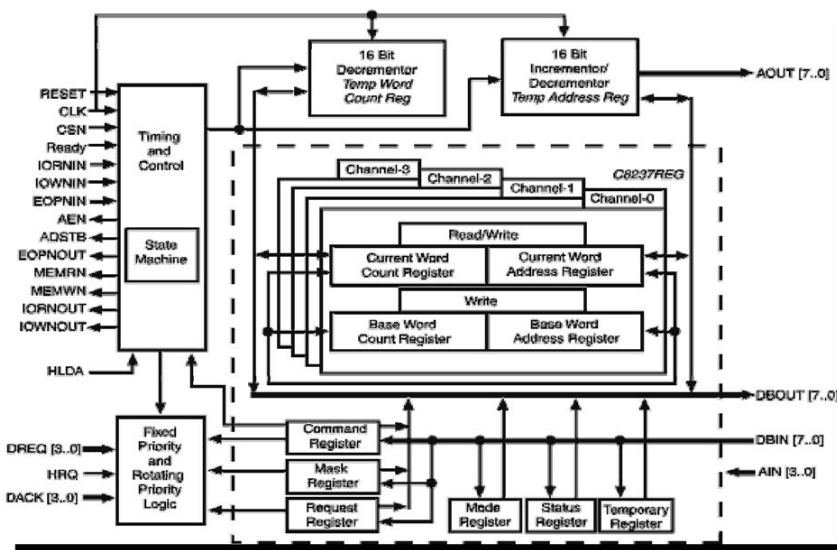
Řadič DMA zajišťuje správu paměti
Dříve byl BUS masterem procesor, který vše řídil.

Řadič DMA

Obvod pro řízení přenosů na sběrnici

- generuje adresy paměti a periferie, generuje řídicí signály pro čtení/zápis
 - při programmed I/O toto dělá procesor
- generuje signály pro procesor, aby zajistil, že procesor nepřistupuje (nezapíše) na sběrnici
- řadič sám se chová jako periferie
 - program nastavuje parametry přenosu, tj. odkud se bude přenášet, kam, a kolik (2 čítače, kanál DMA)
 - zařízení připojena na kanál DMA, při přenosu je cílové zařízení aktivováno řadičem, nikoliv vystavením adresy

Příklad řadiče DMA (i8237)



Channel 0-3 ..
Zvládá si
pamatovat stav
4 přenosů

Přenos dat pomocí DMA

Posloupnost událostí:

1. Program nastaví řadič a periferii povolí přenos
2. Aktivací signálu DREQx periferii požádá řadič DMA o přenos dat z/do paměti
3. Řadič DMA zkontroluje nastavení kanálu vyhodnotí prioritu žádosti
4. Aktivací signálu HOLD řadič DMA požádá CP o přidělení sběrnice
5. Pokud CPU nepotřebuje sběrnici, odpojí se od sběrnice a signalizuje HLDA
 - i. CPU testuje HOLD na začátku strojového cyklu
6. Po přijetí HLDA řadič připraví sběrnici pro přenos - vystaví adresu v paměti a řídicí signály pro čtení/zápis z/do paměti/periferie
7. Řadič DMA aktivuje signál DACKx, kterým vyzve periferii k vystavení/přečtení dat na / ze sběrnice
8. V závislosti na režimu buď přenos končí (pustí signál HOLD, procesor zase pokračuje dál), nebo (burst režim) pokračuje dalším slovem dokud je DREQx aktivní. Až dojde signál na nulu, tak
9. Při posledním slově řadič aktivuje signál EOP
10. Při ukončení přenosu řadič uvolní signál HOLD
11. Procesor uvolní HLDA a připojí se ke sběrnici

Přenos dat po sběrnici

Přenos bloku dat pomocí DMA (i8237)

Posloupnost
událostí

1. program nastaví řadič a periférii a spustí přenos
2. na začátku cyklu sběrnice řadič signalizuje HOLD
3. CPU testuje HOLD na začátku strojového cyklu
 - pokud nepotřebuje sběrnici, signalizuje HLDA a uvolní ji
4. po přijetí HLDA řadič přenesení slovo z/do paměti
 - vystaví adresu v paměti a řídicí signály, a aktivuje periférii aby přečetla/zapsala slovo z/na datovou sběrnici
5. na konci cyklu sběrnice řadič uvolní HOLD
6. procesor uvolní HLDA a připojí se ke sběrnici

Přenos dat po sběrnici

Transparentní režim přenosu s pomocí DMA

- přenos probíhá nezávisle na procesoru
- řadič používá sběrnici v době, kdy ji nepoužívá procesor
 - Ve starších procesorech
 - sběrnice má 2 zdroje hodinového signálu, vzájemně fázově posunuté
 - použije se náběžná i sestupná hrana hodinového signálu
 - efektivně se zdvojnásobí propustnost sběrnice

Přenos dat po sběrnici

Bus mastering

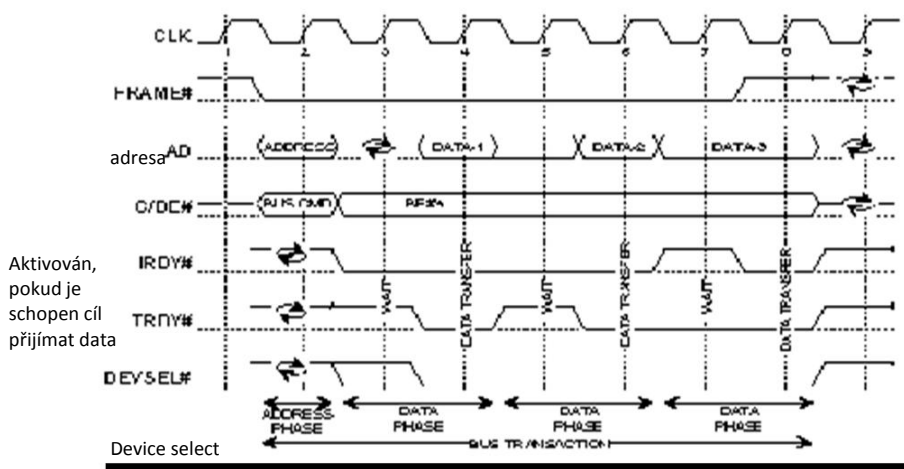
Velké množství podobně schopných účastníků, a pokud nějaký potřebuje přenášet, tak tu sběrnici může řídit libovolný účastník.. Arbitr tohoto řízení bývá na ní integrován v Northbridge

- zobecnění konceptu DMA
- sběrnici může řídit libovolný účastník
 - nutno zažádat o přidělení sběrnice (arbitrace)
 - procesor v roli normálního účastníka
- přenos dat mezi libovolnými účastníky
 - paměť/periferie, periferie/periferie Už nepotřebuji procesor aby viděl všechny data a kopíroval je
 - stále je nutné přenos nastavit z programu
- dávkový režim přenosu (burst mode)
 - 1 adresový cyklus na blok dat Tím ušetříme na adresovém cyklu (vystavím jednou adresu, a pak vystavím větší množství dat)
- přenos nesouvislých bloků (scatter/gather)

Některé síťové karty se dají naprogramovat odkud mají vzít kolik bytů do seznamů (deskriptorů) a potom může vygenerovat burst přenosy a procesor tak nemusí kopírovat vše

Přenos dat po sběrnici

Burst transakce na sběrnici PCI



Přenos dat po sběrnici

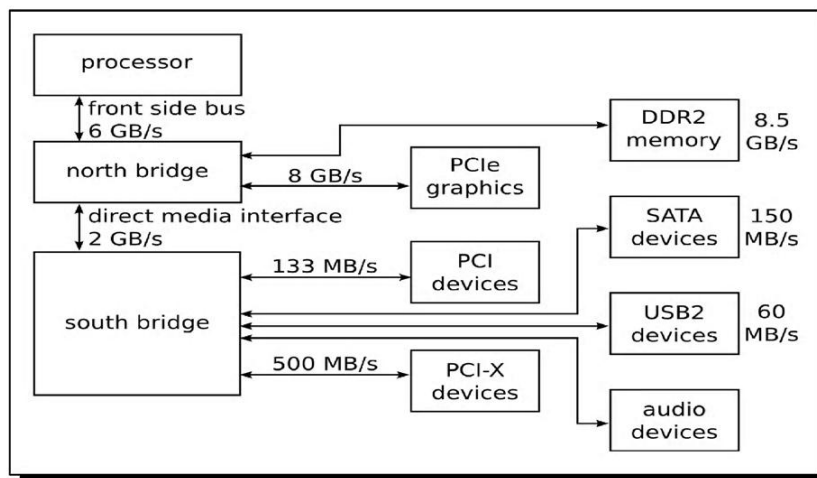
Obecné poznámky k DMA

- přenos pomocí DMA není nutně rychlejší než PIO ^{Programmed I/O}
 - umožňuje procesoru dělat něco užitečnějšího
- systém musí zajistit platnost dat v cache
 - write-back cache může mít novější data než paměť
 - v paměti mohou být novější data než v cache
 - procesor sleduje, provoz z/do paměti

Dát pozor, aby data co se dávají do paměti, jsou aktuální vůči cache procesoru (procesor si musí hlídat zápisy do paměti)

System sběrnic v moderním PC

Propojení různě rychlých sběrnic



CPU

|
Northbridge [k procesoru, nejrychlejší způsob, jsou potřeba často a rychle] - RAM; AGP

|
Southbridge [níže, nejsou tak třeba rychle] - PCI bus; real time clock; APM /power management/; USB

Sběrnice pro připojování periférií

Historické

- **PC Bus** Pro PC/XT (Intel 8088); DMA0-3; už zde se používal režim DMA,
 - 8bit datová, 20bit adresová sběrnice, 8MHz
 - **ISA – Industry Standard Architecture (AT Bus)** Jednoduše ovládatelná pro připojení různorodých zařízení; využívala PC BUS (konektor 62) + vlastní konektor navíc (zpětná kompatibilita)
 - 16bit datová, 24bit adresová sběrnice, 8MHz
 - **MCA – MicroChannel Architecture** Nebylo kompatibilní s ISA, proto se nikdy tak široce neprozšířila
 - 16/32/64bit data, 24/32bit adresy, 10MHz
 - **EISA – Extended Industry Standard Architecture**
 - 32bit data, 32bit adresy, **8MHz**, programová konfigurace Pro snazší konfiguraci (do té doby ručně přímo na HW)
 - **VL Bus – VESA Local Bus** Pro podporu grafiky, pro zvýšení propustnosti obrazu
 - 32bit data, 32bit adresy, až 50MHz, na syst. sběrnici
- Na rozdíl od ostatních periférií byla připojena přímo na sběrnici procesor <> paměť

ATA/ATAPI

AT Attachment / with Packet Interface

16bit, UltraDMA

- Problém: zachování kompatibility se starými procesory 80286

Seriál-ATA

- Od 1,5 Gb/s
- Při návrhu i jednoznačnost zapojení konektorů
- Přejít na čistě asynchronní režim
- Není master/slave; řešila problém s ukončováním sběrnice (když kšanda nebyla uzavřena u ATA)

UltraDMA

- Zrychlení frekvence
- Zavedení CRC (pro kontrolu dat při přenosu)
- Od ATA/ATAPI-5 Mode 3-4 je už třeba 80 žilového kabelu (aby nebyly přeslechy mezi jednotlivými kabely)

SCSI

- Small computer systém interface
- Paralelní interface, od. R 1986 ANSI standard - což dost svědčí i o kvalitě návrhu, že přežil dodnes
- Inteligentní zařízení (náročnější na provoz)
- Zřetězení až 15 zařízení
- Je třeba ukončovací prvek, terminátor sběrnice, aby nevznikaly odrazy na konci sběrnice

Sběrnice pro připojování periférií

PCI – Peripheral Component Interconnect

- připojení k systémové sběrnici přes můstek
- synchronní, multiplex dat a adres
- 32bit data, 32/64bit adresy
- 1992 v1.0: 33MHz, 1995 v2.1: 66MHz Nyní ve verzi 3
- podpora dávkových přenosů (burst mode)

PCI-X Není PC Express

- rozšíření na 64bit data, podpora ECC
- v1.0: 66 a 133MHz, v2.0: 266 a 533 MHz
- zpětná kompatibilita HW i SW

PCI Express (PCIe)

- Sériová, paketový protokol
- x16 8GB/s
- QoS, hot-plug (u předchozích sběrnic se to nemělo dělat), power management

Sběrnice pro připojování periférií

AGP – Advanced Graphics Port

- dvoubodový spoj Ale navrhovaný jako sběrnice - "point-to-point"
- 4x frekvence PCI, tj. 133MHz
- pro přenos dat využita náběžná i sestupná hrana Což umožňuje přenášet větší objemy dat
- pipelining Přímo na sběrnici

Sběrnice pro připojování periferií

PCI Express (PCIe)

- sériová, paketový protokol
- 256 MB/s (PCIe x1) – 8 GB/s (PCIe x16 duplex)
 - pro srovnání PCI/32bit/33MHz - 133MB/s
- QoS, power management, hot-plug...

Sběrnice pro komunikaci mezi IO

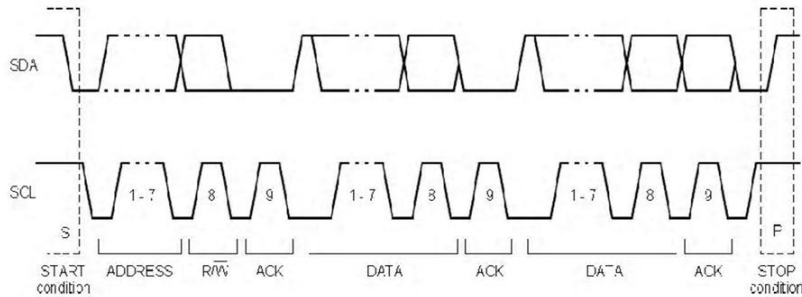
I²C – Inter IC

- dvou vodičová (SDA, SCL) sériová sběrnice
 - v klidovém stavu udržováno kladné napětí (pull-up)
- přenos dat po bajtech, potvrzení příjmu
- rychlosti 0- 100/400/1000/3400 kbit/s
- master/slave řízení
- možnost multi-master konfigurace
 - synchronizace hodin + arbitrace
- www.philipslogic.com/i2c

Sběrnice pro komunikaci mezi IO

Formát transakce na sběrnici I2C

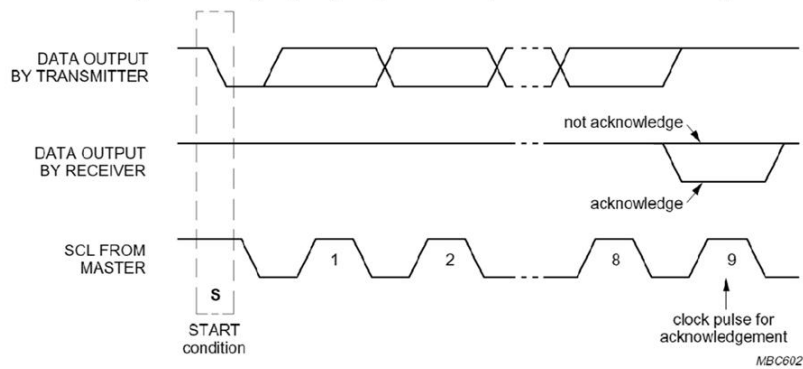
- adresa zařízení, směr přenosu, potvrzení výběru
- data, potvrzení, data, potvrzení, ...



Sběrnice pro komunikaci mezi IO

Přenos dat po sběrnici I2C

- master generuje SCL a provádí adresaci dat
- slave potvrzuje přijetí, může pozdržet hodiny

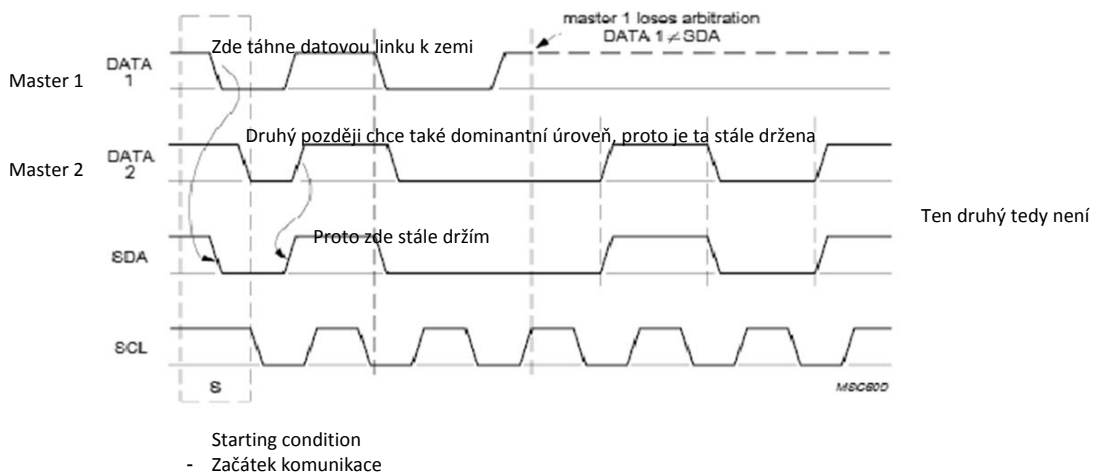


Sběrnice pro komunikaci mezi IO

Arbitrace na sběrnici I²C

- preference dominantního vysílače

Zde první master pozná, že tam je ještě někdo a rozhodne se, že se odpojí (dáno specifikací); dá se na recesivní úroveň - aktivní je jen dominantní úroveň,



Sběrnice pro připojování (nejenom) disků

ATA/ATAPI – AT Attachment/w Packet Interface

- řadič připojený k systémové sběrnici
 - přenos 16-bit dat: PIO, Multiword DMA, UltraDMA
 - 2.1 – 16.7/4.2 – 16.7/16.7 – 100MB/s
 - UltraDMA: přenos doplněn o CRC
- 2 zařízení na kanál (kabel)
 - omezená délka kabelů (nevhodné vlastnosti)

SATA – Serial ATA

- sériový přenos, 150MB/s, 300MB/s
- dvoubodový spoj, podpora hot-plug, ...

Sběrnice pro připojování (nejenom) disků

SCSI – Small Computer System Interface

- paralelní rozhraní, ANSI standard od r. 1986
- 8/16 bit data, až 640MB/s (160MHz, DDR)
- 15 zařízení na 1 sběrnici, podpora hot-plug
 - více logických jednotek na 1 zařízení
- inteligentní zařízení
- podmožinu příkazů používá i ATAPI

SAS – Serial Attached SCSI

- sériový přenos, dvoubodový spoj
- 300 MB/s (1bit, 3GHz)

Sběrnice pro připojování externích periférií

Motivace pro nový typ sběrnice

- uživatel nemusí zařízení nijak konfigurovat
- uživatel nemusí otevřít počítač
- jeden typ kabelu pro všechna zařízení
 - napájení z připojovacího kabelu
- možnost připojení velkého počtu zařízení
- podpora real-time zařízení (zvuk, video)
- instalace zařízení za chodu, bez nutnosti restartovat operační systém
- levná výroba, “fool-proof” design

Sběrnice pro připojování externích periferií

USB – Universal Serial Bus Až 127 zařízení;

- 1,5/12/480 Mb/s
- Control, Bulk, Interrupt, Isochronous přenosy
- stromová struktura – koncová zařízení / hub
- plug-and-play

Umožňuje 4 režimy, pro různé přenosy

IEEE 1394 – FireWire (i.Link)

- synchronní i asynchronní
 - 100/200/400 Mb/s (IEEE 1394b: ~800Mb/s)
 - paketový přenos
- Nominálně nižší než usb, tak přesto je lepší na přenos videa (rychlejší) díky optimalizaci na video

Sběrnice pro připojování externích periferií

Standardizace do posledního detailu...

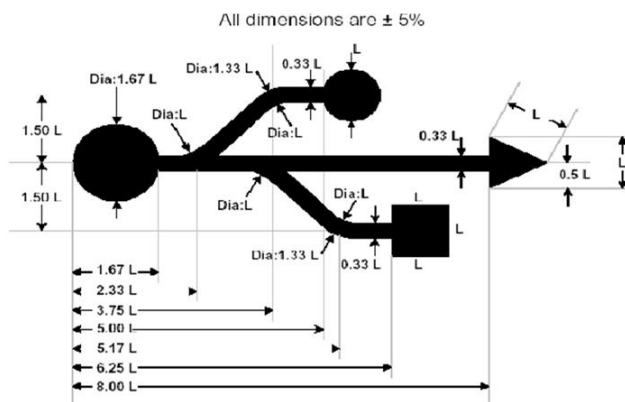


Figure 6-5. USB Icon

Sériová komunikace

- Dvoubodové spoje
- Vícebodové spoje
 - Wired-OR - jediný signál s dominantní úrovní dává dominantní úroveň na sběrnici
 - Když všichni vysílají recesivní, tak je recesivní, jinak stačí alespoň jeden dominantní pro dominantní

RS-232C

- Vysílaný signál je vztažen k zemnímu potenciálu vysílače, přijímaný signál vyhodnocován vzhledem k zemnímu potenciálu přijímače [problém různých zemí]
- Původní návrh určen pro přenos max 19kbs na vzdálenost 20m

Sériové sběrnice

- CAN - Bosch (u aut); v metru takto připojeny displeje písíčí stanice
- SPI - Motorola
- I2C - Philips
 - Definováno až na úroveň přenášených dat

Architektura počítačového subsystému

- Dvoudrátová - data, čas
 - V klidovém stavu udržováno kladné napětí pomocí pull-up, když chceme číst, tak se spojí se zemí, to má menší odpor, než to co když na to nikdo nesáhá (recesivní úroveň), když někdo chce, tak spojíme se zemí /dominantní úroveň/
 - Jedno zařízení je master (tiká hodinami) a ostatní komunikují skrze něj)
- MicroWire - National Semiconductor