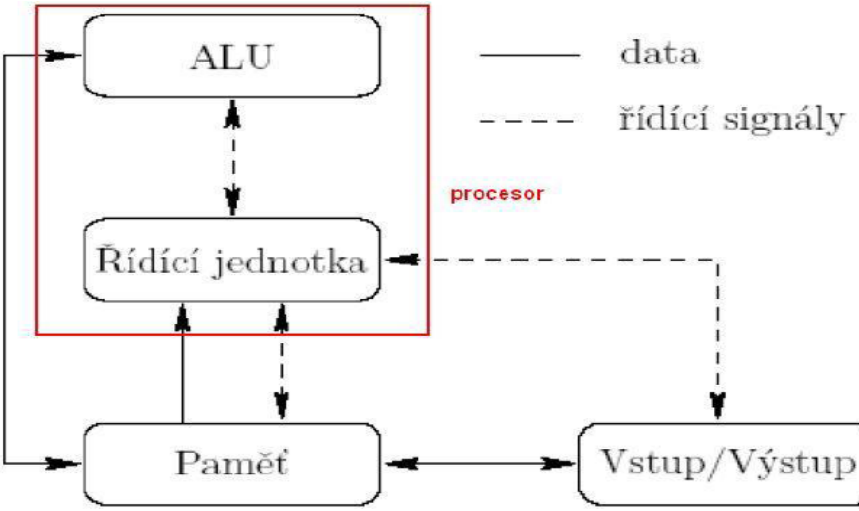
Von Neumanova architektura (1945)

John Von Neumann definoval v roce 1945 základní koncepci počítače (EDVAC) řízeného obsahem paměti.
Od té doby se objevilo několik odlišných modifikací, ale v podstatě se počítače v dnešní době konstruují podle tohoto modelu.
Ve svém projektu si von Neumann stanovil určitá kritéria a principy,které musí počítač splňovat, aby byl použitelný univerzálně. Můžeme je ve stručnosti shrnout do následujících bodů:

Základní principy fungování počítače:

1. Počítač se skládá z paměti, **řídící jednotky, aritmeticko-logické jednotky, vstupní a výstupní jednotky**.



ALU (aritmeticko-logická jednotka) – jednotka provádějící veškeré aritmetické výpočty a logické operace. Obsahuje sčítačky, násobičky a komparátory.
Operační paměť – slouží k uchování zpracovávaného programu, zpracovávaných dat a výsledků výpočtu
Řídící jednotka – řídí činnost všech částí počítače. Toto řízení je prováděno pomocí řídících signálů, které jsou zasílány jednotlivým modulům. Dnes řadič spolu s ALU tvoří jednu součástku, a to procesor neboli CPU (Central Processing Unit).
Vstup/ Výstup – zařízení určené pro vstup dat, a výstup zpracovaných výsledků.

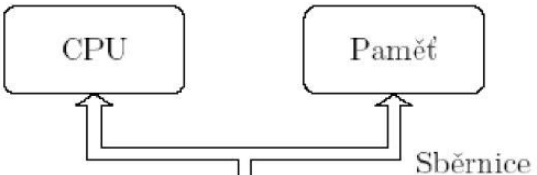
2. Struktura pc je nezávislá na typu řešené úlohy (univerzálnost), počítač se programuje obsahem paměti.
3. Následující krok počítače je závislý na kroku předešlém.
4. Instrukce a data jsou v téže paměti.
5. Paměť je rozdělena do paměťových buněk stejné velikosti (Byte), jejichž pořadová čísla se využívají jako adresy.
6. Program je tvořen posloupností instrukcí, které se vykonávají jednotlivě v pořadí, v jakém jsou zapsány do paměti.
7. Změna pořadí prováděných instrukcí se provádí skokovými instrukcemi (podmíněné nebo nepodmíněné skákání na adresy)
8. Čísla, instrukce, adresy a znaky se značí ve dvojkové soustavě.

Nevýhody Von Neumana ve srovnání s dnešními pc:
Podle von Neumannova schématu počítač pracuje vždy nad jedním programem. Toto vede k velmi špatnému využití strojového času. Dnes je obvyklé, že počítač zpracovává paralelně více programů zároveň - tzv. multitasking
Počítač může mít i více jak jeden procesor.
Podle Von Neumanova schématu mohl počítač pracovat pouze v tzv. diskrétním režimu, kdy byl do paměti počítače zaveden program, data a pak probíhal výpočet. V průběhu výpočtu již nebylo možné s počítačem dále interaktivně komunikovat.
Dnes existují vstupní/ výstupní zařízení, např. pevné disky a páskové mechaniky, které umožňují vstup i výstup.
Program se do paměti nemusí zavést celý, ale je možné zavést pouze jeho část a ostatní části zavádět až v případě potřeby.

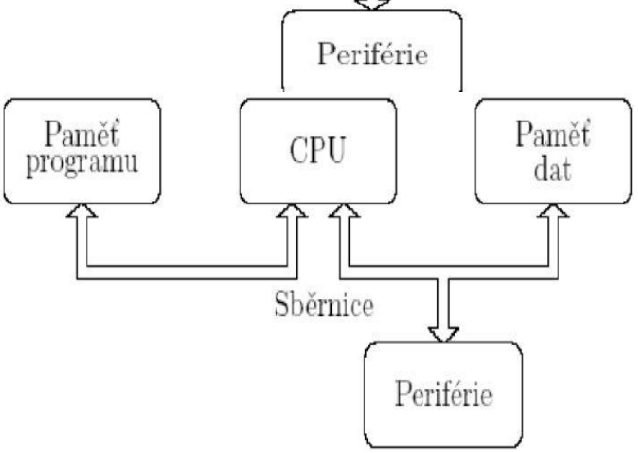
Harvardská koncepce

Několik let po von Neumannovi, přišel vývojový tým odborníků z Harvardské univerzity s vlastní koncepcí počítače, která se sice od Neumannovy příliš nelišila, ale odstraňovala některé její nedostatky. V podstatě jde pouze o **oddělení paměti pro data a program**. Abychom si mohli obě koncepce porovnat, můžeme vycházet ze zjednodušených schémat.

Von Neumanova koncepce
+ rozdělení paměti pro kód a data určuje programátor, řídící jednotka přistupuje pro data i instrukce jednotným způsobem
+ jedna sběrnice -> jednodušší levnější výroba
– jedna sběrnice je omezující
– společné uložení dat a kódu může mít za následek přepsání vlastního programu



Harvardská koncepce
+ program se nepřepíše (oddělené paměti pro data a program)
+ dvě sběrnice umožňují paralelní načítání instrukcí a dat
+ paměti mohou být vyrobeny odlišnými technologiemi a každá může mít jinou nejmenší adresovací jednotku (8 bitů pro instrukce a 8, 16 nebo 32 pro data)
– 2 sběrnice mají vyšší nároky na vývoj řídící jednotky a jsou také dražší a složitější na výrobu
– paměť je rozdělena už od výrobce
– nevyužitou část dat nelze využít po program a obráceně



Základním nedostatkem obou koncepcí je **sekvenční vykonávání instrukcí,** které sice umožňuje snadnou implementaci systému, ale nepovoluje paralelní zpracování. Paralelizmy se musí simulovat až na úrovni operačního systému. Sběrnice zas nedovolují přistupovat do více míst paměti současně a navíc dovolují v daném okamžiku přenos dat jen jedním směrem.

Procesory
Procesor je **integrovaný obvod** označovaný jako CPU (ALU+řadič). Tvoří srdce a mozek celého počítače. Do značné míry ovlivňuje celý jeho výkon (čím rychlejší procesor, tím rychlejší pc). Bývá umístěn na základní desce. Obsahuje rychlá paměťová místa malé kapacity zvané registry.

1. RISC
- **Reduced Instruction Set Computer = počítač s redukovaným souborem instrukcí**
- počet instrukcí a způsobů adresování je malý, ale zůstává úplný, aby bylo možno provést vše
- instrukce jsou vytvořeny pomocí obvodu → jednodušší na výrobu
- širší sběrnice, rychlejší tok instrukcí a dat do procesoru
- instrukce jen nad registry
- navýšen počet registrů → delší program
- instrukce mají jednotný formát – délku i obsah
- komunikace s pamětí pouze pomocí instrukcí LOAD/ STORE
- každý strojový cyklus znamená dokončení jedné instrukce
- používá se zřetězené zpracování instrukcí
- řešení problému s frontou instrukcí
- mikroprogramový řadič může být nahrazen rychlejším obvodovým
- přenášejí složitost technologického řešení do programu (překladače)
- předtavitelé : ARM, MOTOROLA 6800, INTEL i960, MIPS R6000

2. CISC
- **Complex Instruction Set Computer = počítač se složitým souborem instrukcí**
- každý problém se snaží provést jednou instrukcí -> kratší program, malý počet registrů, ale velká sada instrukcí vede pak ke komplikovanější a dražší výrobě
- mají různě dlouhé strojové instrukce, jejichž vykonání trvá různě dlouhou dobu -> problémy u zřetězeného zpracování
- představitelé: Pentium, Pentium PRO, Pentium 2-4, Pentium M (CORE DUO)

ARM procesor (Advanced RISC Machine)
- 32 bitová mikroprocesorová architektura typu RISC
- energeticky úsporné vlastnosti → mobilní telefony, mp3 přehrávače, notebooky
- Procesory ARM podporují **dva adresové módy**. Můžeme adresovat buď prostřednictvím čítače instrukcí, nebo pomocí bázové adresy uložené v jednom z vnitřních registrů.
- Do paměti lze přistupovat pouze instrukcemi Load/Store což výrazně zjednodušuje výkonnou jednotku procesoru, protože pouze několik instrukcí pracuje přímo s pamětí. Většina instrukcí pracuje s vnitřními registry.
- ARM procesory podporují dvě úrovně priority přerušení s dvěma zaměnitelnými bankami registrů. Nejkratší doba provedení požadavku na přerušení je poskytována režimem rychlého přerušení **FIQ (Fast Interrupt Request)**. Druhý typ přerušení je **IRQ (Interrupt Request)** se používá pro obsluhu přerušení nevyžadujících extrémně krátké doby odezvy.
- Procesor ARM obsahuje 44 základních instrukcí s jednotnou šířkou 32 bitů. V jednom taktu se vykonávají pouze

instrukce pracující s ALU, s registry nebo s přímými operandy.
- Procesor pracuje ve čtyřech základních režimech:
    uživatelský režim USR
    privilegovaný režim supervizora SUP
    privilegovaný režim přerušení IRQ
    privilegovaný režim rychlého přerušení FIQ
- Instrukční soubor můžeme rozdělit na skupiny:
    instrukce zpracování údajů - zpracování registrových operandů, zpracování přímých operandů, nastavení podmínkového kódu a instrukce aritmeticko-logické
    instrukce jednoduchého přenosu údajů - jsou použity k přenosu dat mezi pamětí a souborem registrů (Load) a naopak (Store).
    instrukce blokového přenosu údajů - zabezpečují přenos několika registrů jednou instrukcí, která obsahuje pole bitů
    instrukce větvení a větvení s uchováním návratové adresy
    instrukce přechodu do privilegovaného režimu supervizora, které zahrnují i programové přerušení

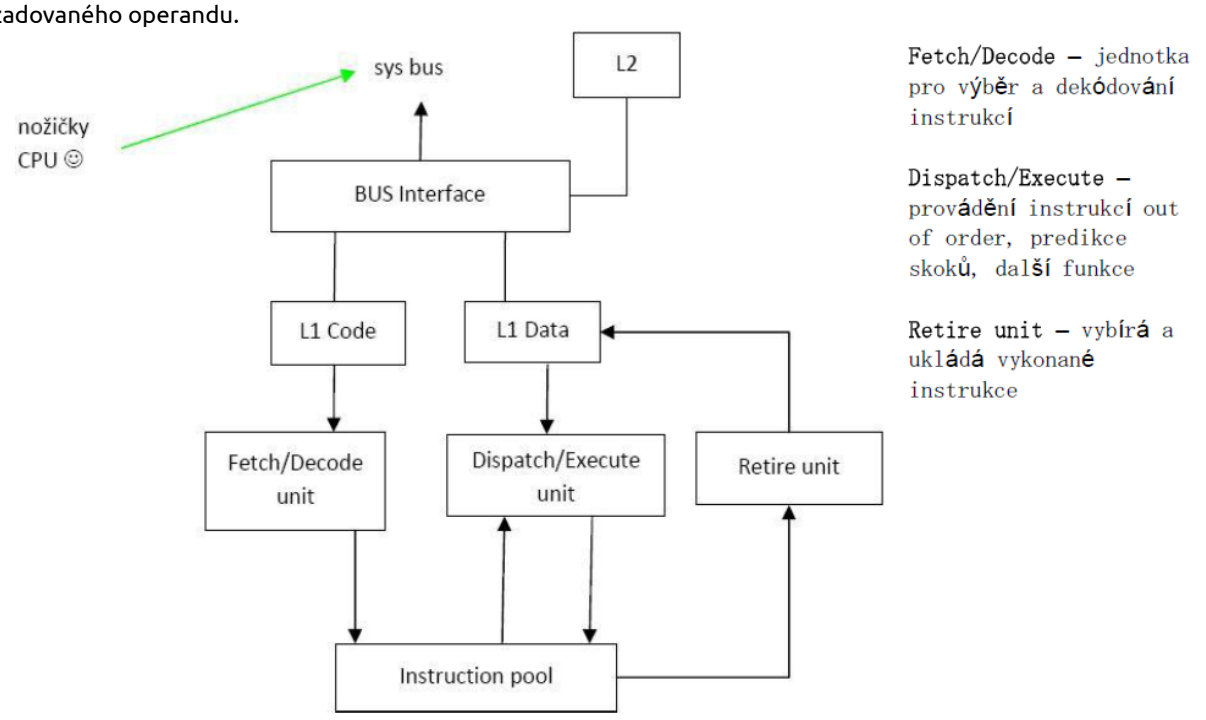
Pentium PRO

Koncem roku 1995 uvádí firma Intel na trh další generaci procesorů řady 80x86. Novinkou jeho architektury je integrace externí cache paměti o kapacitě 256 kB (512 kB) přímo do pouzdra procesoru. Tato cache paměť není součástí čipu procesoru, ale je tvořena samostatným čipem umístěným v jednom pouzdru s čipem procesoru. Čip procesoru Intel Pentium pro je ekvivalentem asi 5,5 mil. tranzistorů a čip jeho externí cache paměti obsahuje cca 15 mil. tranzistorů. Pentium Pro je až 2x rychlejší než Pentium. Nový design procesoru vykazuje 4 pipelines na paralelní zpracování příkazů a integrovanou primární a sekundární cache paměť (L1 a L2 cache).

Vlastnosti

externí cache v pouzdře s čipem
spekulativní provádění instrukcí mimo pořadí
instrukce RISC, sběrnice: A32/D64
3 celočíselné jednotky (14° zřetězení), 1 jednotka pro pohyblivé řadové čárky
vlastní soubor instrukcí – **mikrooperace** (předtím jednotky zpracovávaly přímo instrukce z instrukčního souboru 80x86), zatímco teď jsou jednotlivé instrukce (80x86) překládány do jedné nebo několika mikrooperací, které jsou dále předávány ke zpracování jednotlivým prováděcím jednotkám.
využití tzv. **přejmenování registrů** při provádění mikrooperací, kdy 40 záložních registrů mohou být přejmenovány na 8 z libovolných univerzálních registrů. Podpora pro multiprocesorové pc.

Instruction Pool je v podstatě sada 40 speciálních registrů, přičemž do každého z nich se vejde právě jedna mikroinstrukce.
Pentium PRO umožňuje vykonávání instrukcí tzv. "**out of order**" čili je nemusí brát postupně z fronty, ale z instruction pool, který zrovna chce. Např. v případě že požadovaná data nejsou v cache paměti Pentium PRO nečeká na načtení z pomalejší operační paměti, ale provádí další instrukce. Instrukce mimo pořadí vykonává také, dokud nedojde k výpočtu požadovaného operandu.



Pentium 4 – novější verze Pentia, používá architekturu **NetBurst**, která zdvojnásobuje hloubku zřetězeného zpracování, systémová sběrnice pracuje v rozsahu 400 – 533 MHz což umožňuje přenosy s rychlostí až 3,2 GB/s, externí cache umožňuje uložit až 12 kB mikrooperací a interní cache má 8kB, má 2 ALU s dvojnásobným taktem oproti vnitřní frekvenci procesoru (celočíselné a logické operace jsou prováděny během ½ taktu)

Zřetězené zpracování instrukcí (pipelining)

Na dosažení zřetězení je nutné rozdělit úlohu do posloupnosti dílčích úloh, z nichž každá může být vykonána samostatně, např. oddělit načítaní a ukládání dat z paměti od provádění výpočtu instrukce…a tyto části pak mohou běžet souběžně.

To znamená, že musíme osamostatnit jednotlivé části sekvenčního obvodu tak, aby každému obvodu odpovídala jedna fáze zpracování. Všechny fáze musí být stejně časově náročné, jinak je rychlost degradována na nejpomalejší z nich.
Fáze zpracování je rozdělena minimálně na 2 úseky:

- načtení a dekódování instrukce
- provedení instrukce a případné uložení výsledku

Zřetězení se stále vylepšuje a u novějších procesorů se již můžeme setkat stále s více řetězci rozpracovaných informací (více pipelines), dnes je standardem 5 pipelines.

Problém

Největší problém spočívá v plnění zřetězené jednotky, hlavně při provádění podmíněných skoků, kdy během stejného počtu cyklů se vykoná více instrukcí.
U pipelingu se instrukce následující po skoku vyzvedává dřív než je skok dokončen. Primitivní implementace vyzvedává vždy následující instrukci, což vede k tomu že se vždy mýlí, pokud je skok nepodmíněný. Pozdější implementace mají jednotku předpovídání skoku (1 bit), která vždy správně předpoví nepodmíněný skok a s použitím cache se záznamem předchozího chování programu se pokusí předpovědět i cíl podmíněných skoků nebo skoků s adresou v registru nebo paměti.
V případě, že se predikce nepovede, bývá nutné vyprázdnit celou pipeline a začít vyzvedávat instrukce ze správné adresy, což znamená relativně velké zdržení. Souvisejícím problémem je přerušení.

Plnění fronty instrukcí

Pokud se dokončí skoková instrukce která odkazuje na jinou část kódu musejí být instrukce za ní zahozeny (problém plnění fronty instrukcí).

- u malého zřetězení neřešíme
- používání bublin na vyprázdnění pipeline, naplnění prázdnými instrukcemi
- **predikace skoku** – vyhrazen jeden bit předurčující, zda se skok provede či nikoliv
  - statická – součást instrukce, řeší programátor
  - dynamická
    - jednobitová – zaznamenává jestli se skok provedl či ne (1/ 0)
    - dvoubitová – metoda zpožděného skoku, v procesoru řeší se např. tabulkou s 4kB instr.

- **superskalání architektura (zdvojení)** – když nastane podmíněný skok, začnou se vykonávat instrukce obou variant, nepotřebná část se pak zahodí. Tento způsob, pak vyžaduje vyřešit ukládání výsledku.

Hazardy

- **datové** - chceme pracovat s nehotovým výsledkem (předtím než se data uloží)
  a = b + c + d (c + d musíme nejdříve uložit a potom teprve sčítat)

- **strukturální** – důsledek konstrukce počítače, omezený počet zdrojů

Monolitické počítače (mikroprocesory)

- mikroprocesory, mikrokontroléry, minipočítače jsou další názvy pro monolitické počítače
- jsou to malé počítače **integrované v jediném pouzdře** (all in one)
- mají širokou oblast využití
- využívá se **Harvardské koncepce**, což umožňuje aplikovat paměti pro data a program různých technologií
- zjednodušené rysy architektury RISC
- INTEL 8051 (standart), ATMEL, MICROCHIP PIC

Paměti

pro **data** používáme většinou paměti energeticky závislé typu RWM-RAM (Read-Write Memory - Random Access Memory), tedy paměť s libovolným přístupem určenou pro čtení i zápis. Tyto paměti jsou vyráběny jako **statické** (uchování paměti po celou dobu napájení), jejich paměťové buňky jsou realizovány jako klopné obvody.
pro **program** se používají paměti, které si svůj obsah zachovají i po odpojení napájení, tedy jde o paměti typu ROM určeny především ke čtení. Nejčastěji paměti EPROM, EEPROM a Flash. Nesmíme také zapomenout na výrobky s pamětí PROM (Programmable EPROM)

Organizace paměti

Střadačové (pracovní) registry - ve struktuře procesoru jsou obvykle 1-8-16 základních pracovních registrů, jsou nejpoužívanější. Ukládají se do nich aktuálně zpracovávaná data a jsou nejčastějším operandem strojových instrukcí (to na co se instrukce v závorkách odkazují). A také se do nich nejčastěji ukládají výsledky operací. Nejsou určeny pro dlouhodobé ukládání dat.

Univerzální zápisníkové registry – jsou jich desítky až stovky. Slouží pro ukládání nejčastěji používaných dat. Instrukční soubor obvykle dovoluje, aby se část strojových instrukcí prováděla přímo s těmito registry. Formát strojových instrukcí ovšem obvykle nedovoluje adresovat velký rozsah registru, proto se implementuje několik stejných skupin registru vedle sebe, s možností mezi skupinami prepínat - registrové banky.

Paměť dat RWM - slouží pro ukládání rozsáhlejších nebo méně používaných dat (z těch předešlých nejméně používaný). Instrukční soubor obvykle nedovoluje s obsahem této paměti přímo manipulovat, kromě instrukcí přesunových. Těmi se data přesunou např. do pracovního registru. Některé procesory dovolují, aby data z této paměti byla použita jako druhý operand strojové instrukce, výsledek ale nelze zpět do této paměti uložit přímo.

Zdroje synchronizace

krystal (křemenný výbrus) – jsou drahé ale přesné
obvod LC – méně časté
obvod RC – snadno integrovatelný
zásuvka 230V/50Hz

Ochrana proti rušení

Na prvním místě většinou jde o **ochranu mechanickou**. Odolávat náhodným rázům, nebo i trvalým vibracím nebo elektromagnetickým vlivům z okolí. Pro odstranění chyb, které nastanou působením vnějších vlivů nebo chyby programátora, je v mikropočítačích implementován speciální obvod nazývaný **WATCHDOG** (provede pomocí vnitřního RESETu reinicilazaci mikropočítače) – patří k ochraně elektrické.
K elektrické ochraně se také řadí **BROWN-OUT** ochrana proti podpětí (→ reset).

Typické periferie

Periferie – obvody, které zajišťují komunikaci mikropočítače s okolím

1. **Vstupní a výstupní brány**

Nejjednodušší a nejčastěji používané rozhraní pro vstup a výstup informací je u mikropočítačů paralelní brána – **port**. Bývá obvykle organizována jako 4 nebo 8 jednobitové vývody, kde lze současně zapisovat i číst logické informace 0 a 1. U většiny bran lze jednotlivě nastavit, které bitové vývody budou sloužit jako vstupní a které jako výstupní. Na vstupu je Schmittův klopný obvod. U mnoha mikropočítačů jsou brány implementovány tak, že s nimi instrukční soubor může pracovat jako s množinou vývodu, nebo jako s jednotlivými bity.

2. **Čítače a časovače**

Do skupiny nejpoužívanějších periférií mikropočítače určitě patří čítače a časovače.
Časovač se od čítače příliš neliší. Není ale inkrementován vnějším signálem, ale přímo v**nitřním hodinovým signálem** používaným pro řízení samotného mikropočítače. Lze tak podle přesnosti zdroje hodinového signálu zajistit řízení událostí a chování v reálném čase. Při přetečení časovače se i zde může automaticky předávat signál do přerušovacího podsystému mikropočítače.

3. **Sériové linky**

Sériový přenos dat je v praxi stále více používán. Dovoluje efektivním způsobem přenášet data na relativně velké vzdálenosti při použití minimálního počtu vodičů. Hlavní nevýhodou je však nižší přenosová rychlost, a to že se data musí kódovat a dekódovat.

USART (RS232) +/-12V jet transformována na TTL /RS422/RS485

I2C (Philips) komunikace mezi integrovanými obvody (přenos dat uvnitř elektronického zařízení)
SPI

4. **A/D a D/A převodníky**

Fyzikální veličiny, které vstupují do mikropočítače, jsou většinou reprezentovány analogovou formou (napětím, proudem, nebo odporem). Pro zpracování počítačem však potřebujeme informaci v digitální (číselné) formě. K tomuto účelu slouží analogově–číslicové převodníky. Existuje několik základních typu těchto převodníků.

A/D převodníky – velký počet součástek, malá rozlišovací schopnost, velmi rychlá
D/A převodníky – PWM (šířková modulace pulzu), vyroben pomocí odporů, jednoduchá konstrukce, rychlá odezva, velká rozlišovací schopnost, vyrobit sadu odporů vyžaduje velkou přesnost

Princip PWM (pulse-width modulation = pulsně šířková modulace)

PWN je diskrétní (nespojitá) modulace pro přenos analogového signálu pomocí dvouhodnotového signálu. Jako dvouhodnotová veličina může být použito například napětí, proud, nebo světelný tok. Signál je přenášen pomocí střídy, což je poměr mezi stavy zapnuto/vypnuto (log.1, log.0). Cyklu, kdy dojde k přenosu jedné střídy se říká perioda (součtem doby zapnuto a vypnuto).
V mikropočítači lze realizovat pomocí technických prostředků, nejčastěji pomocí čítače, nebo i programově. Počet bitů

použitého čítače N určuje rozlišovací schopnost převodníku. Výstupní hodnota muže mít 2N různých úrovní.
Pro převod šířkově modulovaných pulzů na analogovou veličinu slouží RC článek. Pro správnou činnost filtru je nutné, aby časová konstanta RC byla výrazně větší, než T. Musí tedy platit: R · C >> T

Technologie výroby číslicových obvodů

Integrované obvody je možné vyrábět pomocí různých technologií, z nichž každá má svůj základní stavební prvek a díky němu poskytuje specifické vlastnosti

Bipolární PNP,NPN – přenosu se účastní elektrony i díry
Unipolární MOS-FET – přenosu náboje se účastní pouze jeden druh nosičů

1. **Bipolární technologie**

   RTL technologie (Rezistor-Transistor-Logic) – základem tranzistor a rezistor

   DTL technologie (Diode-Transistor-Logic) – základem tranzistor a dioda

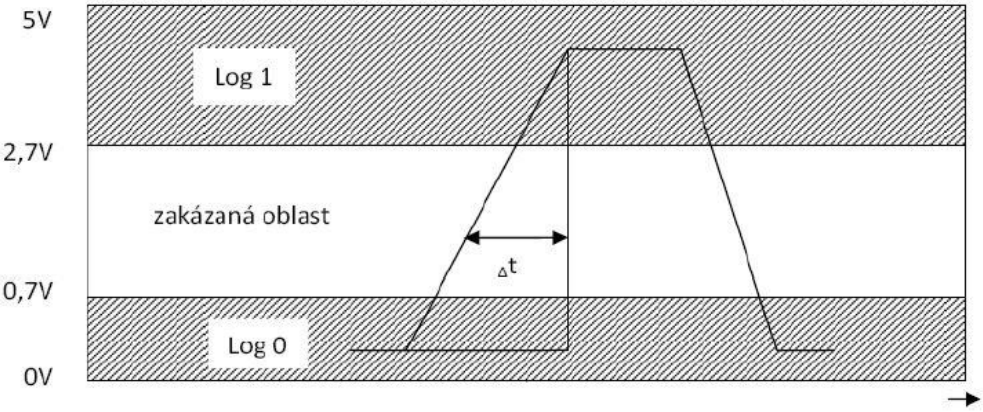
   TTL technologie (Transistor-Transistor-Logic) – jejím základním stavebním prvkem je bipolární tranzistor.

   Její nevýhodou je velká spotřeba elektrické energie a z toho vyplývající velké zahřívání takovýchto obvodů**.** Nejrozšířenější při výrobě obvodu SSI a MSI (s malou a střední integrací).

je co

klopný *ignoruje oblast, šikminy čára, dokud*

zakázanou pamatuje si hodnotu



Δt *– snaha nejblíže k 0*
Schmittův obvod *– zakázanou místo vertikální nepřede celou oblast původní*

Technologie IIL / I2L / I3L – rychlejší než TTL, jen pro vyšší integraci uvnitř číslicových obvodů (LSI, VLSI)

Technologie ECL (Emitor Coupled Logic) – velmi vysoká rychlost (1-2 ns), tranzistory pracují v pracovní oblasti → malá odolnost proti rušení, logika se využívá mezi chipsetem a dynamickou pamětí

2. **Unipolární technologie**

   MOS-FET – nadbytek elektronů -> rychlejší, minimální proud 0,00 mA

   PMOS – nadbytek děr, technologie používající unipolární tranzistor MOS s pozitivním vodivostním kanálem P. MOS tranzistory jsou řízeny elektrickým polem a nikoliv elektrickým proudem jako u technologie TTL -> redukuje nároky na spotřebu elektrické energie. Jedná se však o pomalou a dnes nepoužívanou technologii.

   NMOS – základním prvkem je unipolární tranzistor MOS s negativním vodivostním kanálem N, 3x rychlejší než PMOS, levnější a efektivnější technologii než TTL.

   CMOS - technologie spojující v jednom návrhu prvky tranzistorů PMOS i NMOS. Tyto obvody mají malou spotřebu a tato technologie je používána pro výrobu velké části dnešních moderních integrovaných obvodů. Používají se jako paměti BIOSU (obsah se nemění)

   BICMOS – CMOS v bipolární technologii

   FAMOS, FLOTOX – trvalá paměťová buňka EPROM (mazání UV světlem) a EEPROM (mazání elektronicky)

   Flash PROM – jednovrstevná (SLC), vícevrstevná (MLC), maže se jen část ne celá paměť

Paměti

Paměť počítače je zařízení, které slouží k ukládání programů a dat, s nimiž počítač pracuje. Paměti v PC jsou např. vnitřní paměti procesoru, registry, zásobníky, fronty, paměť mikroprogramů v řadiči procesoru, hlavní paměť atd. Paměti lze rozdělit podle mnoha kriterií:

Dělení podle přístupu

RAM (Random Acces Memory) – paměti sl ibovolným přístupem
SAM (Serial Acces Memory) – paměti se sériovým přístupem
Paměti se speciálními způsoby přístupu – asociativní paměť, FIFO, LIFO atd.

Podle možnosti zápisu

RWM (Read Write Memory) – paměti pro čtení i zápis
ROM (Read Only Mmory) – pouze pro čtení dat

Podle principu elementární buňky

SRAM – statické paměti
DRAM – dynamické paměti
PROM, EPROM, EEPROM, FLASH – programovatelné paměti

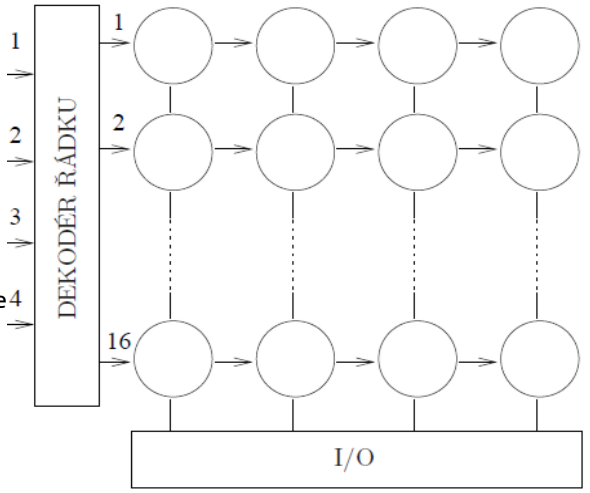
Statické paměti = SRAM (Static Random Acess Memory) – informace je uložena **stavem klopného obvodu**, mezi dvěma stavy lze přepínat pomocí vnějšího signálu. Buňka se skládá z tranzistorů.

SRAM čip je dražší a může obecně pojmout méně dat než DRAM čip (kvůli menší hustotě paměťových buněk na jednotku místa).

SRAM paměti se používají pro malé a rychlé cache paměti. Paměťové buňky jsou uspořádány v 2D matici, řádky a sloupce této matice se vybírají pomocí dekodéru.

V paměťovém kontroléru SRAM jsou adresy řádků a sloupců poskytnuty současně.

Není potřeba refresh (občerstvování). Nepoužívá se zde multiplexu.



Dynamické = DRAM (Dynamic Random Acess Memory) – informace je uložena ve formě **náboje v kondenzátoru** (nabitý = log 1 nebo vybitý = log 0).

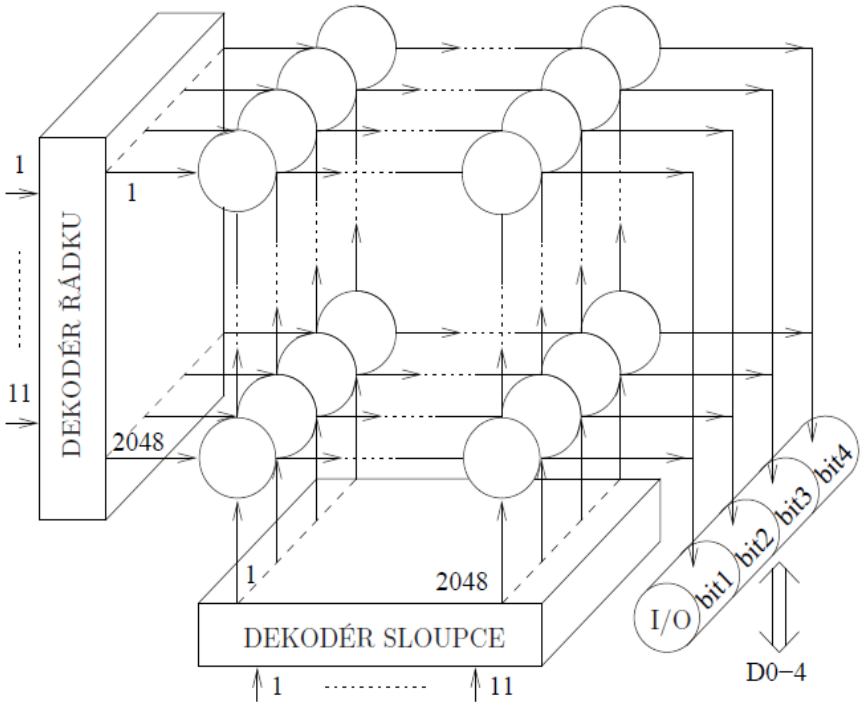
Kondenzátory jsou uloženy ve čtvercové matici → počet vodičů je poloviční.

Kapacita kondenzátorů je velmi malá, to znamená, že i velmi malý proud tekoucí do nebo z kondenzátoru vyvolá velké změny jeho napětí v krátkém čase → nutná obnova (refresh).

V běžných osobních počítačích je operační paměť v současnosti vytvořena pomocí dynamických pamětí DRAM.

DRAM čip může obecně pojmout více dat než SRAM čip (kvůli větší hustotě paměťových buněk na jednotku místa).

DRAM čipy se používají pro velké a relativně pomalé hlavní paměti (RAM)



Typy

SDRAM – Synchronous DRAM, přístupová doba 8-15 ns
DDR SDRAM (double-data-rate synchronous dynamic random access memory) – data přenášené na náběžné i sestupné hraně

Jednotliví představitelé pamětí

Registry procesoru – SRAM, velikost jednotky až desítky, nejrychlejší
L1 cache – uvnitř CPU – SRAM, klopné obvody, velikost až desítky kB
L2 cache – není v jádře, SRAM, jednotky MB
Hlavní paměť – DRAM, velikost jednotky GB
HDD – magnetická paměť, velikost až stovky GB
CD/DVD – tvarová /optická paměť, velikost 700MB – 15 GB

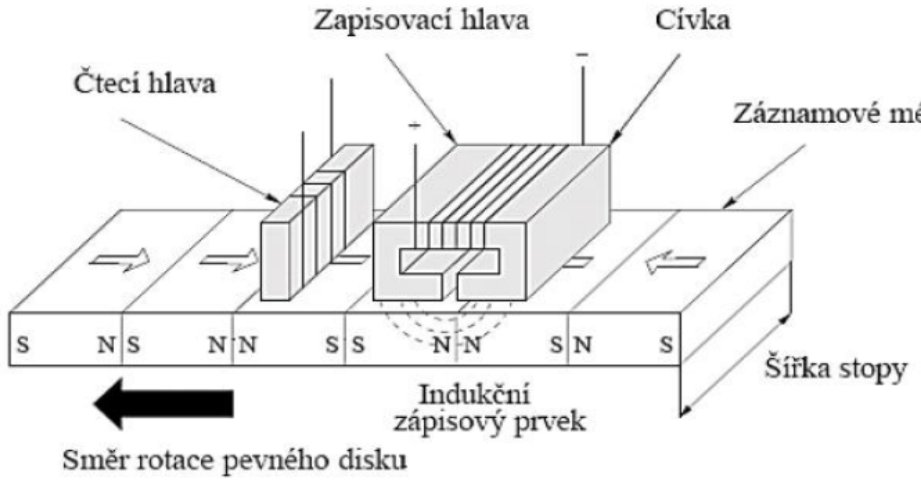
Pásky – nejpomalejší, nejlevnější

Diskové paměti

1. **Magnetické paměti**
   záznamové médium má tvar kruhové desky nebo dlouhé pásky a je pokryto magnetickou vrstvou rozdělenou na jednotlivé sektory, u nichž lze měnit pomocí cívky orientaci, podle toho zda-li chceme vyjádřit 1 nebo 0 -> nastavujeme S a J
   materiál nese informaci trvale
   mezi hlavní představitele magnetické paměti dnes patří pevný disk a magnetofonové pásky

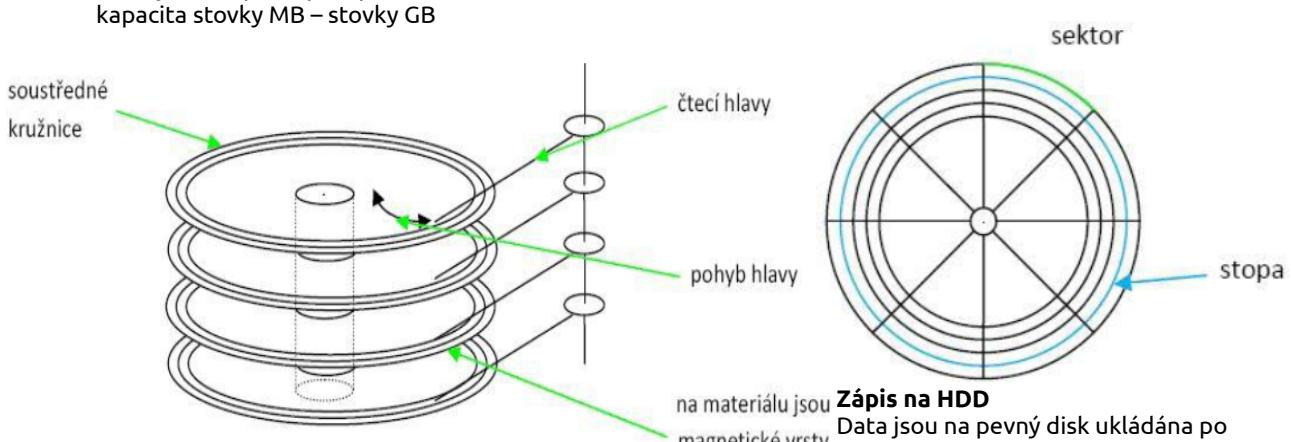
Princip magnetického záznamu
Zapisovací hlavu obvykle tvoří magneticky měkký materiál, na němž je navlečená cívka. Tato hlava má u styku s médiem štěrbinu. Prochází-li cívkou hlavy proud, vzniká v jejím jádru magnetické pole (toto pole v místě štěrbiny z jádra „vystupuje“ a magnetuje aktivní vrstvu záznamového materiálu (pásek), který se před štěrbinou rovnoměrně posouvá. Při změně směru elektrického proudu v cívce se mění i směr magnetického toku jádrem a tím smysl magnetizace vrstvy. Tak vznikají oblasti magnetizované tím či oním směrem a mezi nimi místa magnetických změn (magnetických reverzací), které představují zapsanou informaci (na obr. šipky vlevo či vpravo jsou buď 1 nebo 0).
Čtení je realizováno také pomocí cívky, ve které se při pohybu nad různě orientovanými zmagnetizovanými místy indukuje elektrický proud.

Pevné disky
Pevné disky jsou média pro uchování dat s vysokou kapacitou záznamu (řádově stovky MB až desítky GB). V současnosti jsou pevné disky standardní součástí každého PC. Jedná se o pevně uzavřenou



nepřenosnou jednotku. Uvnitř této jednotky se nachází několik nad sebou umístěných rotujících kotoučů (disků). Tyto disky se otáčejí po celou dobu, kdy je pevný disk připojen ke zdroji elektrického napájení nezávisle na tom, zda se z něj čte (na něj zapisuje). Rychlost otáčení bývá 3600 až 7200 ot./min. Díky tomuto otáčení se v okolí disků vytváří tenká vzduchová vrstva, na níž se pohybují čtecí/zapisovací hlavy.

Schéma pevného disku
plotna – kruhovitá, kovová či skleněná deska pokrytá magnetickou vrstvou, u pevných disků jsou plotny neohebné (u disket ohebné), nad každým povrchem plotny je čtecí hlava, počtu ploten odpovídá počet hlav
sektor – data jsou na disk ukládána v bytech (1 byte = 8 bitů), byty jsou uspořádány do skupin po 512 = sektor, u většiny pevných disků je počet sektorů na všech stopách stejný → plýtvání médiem protože na vnější straně kruhu může být umístěno víc sektorů
stopa – rozdělení disků do soustředných kružnic, počet kružnic neustále narůstá
cylindr = souhrn stop v jedné poloze hlav, válec který tvoří stopy nad sebou (více ploten nad sebou)
přenosová rychlost – 10-100MB/s
otáčky – 5400/7200/10K/15K
kapacita stovky MB – stovky GB



Zápis na HDD
Data jsou na pevný disk ukládána po jednotlivých cylindrech tak, že se zaplní

sektory celého 1. cylindru, a pak druhého. Tento způsob dovoluje, aby se čtecí (zapisovací) hlavy podílely na čtení (zápisu) paralelně. Ukládání dat po jednotlivých discích by bylo podstatně pomalejší, protože v daném okamžiku by vždy mohla pracovat právě jedna hlava.

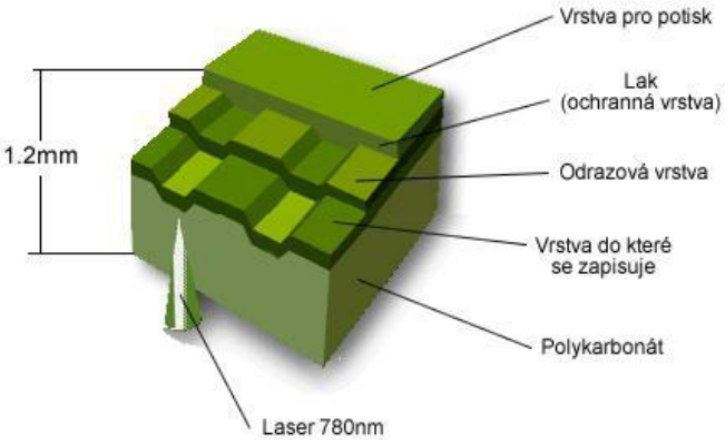
2. Optické paměti

CD-ROM / DVD

médium, které je určeno pouze ke čtení informací
kapacita CD-ROM 74 minut = 700 MB, 22.000 závitů, délka stopy je cca 6 km
kapacita DVD, 50.000 závitů, 12 km stopy
průměr 12 cm
polykarbonát
data nejsou ukládána do soustředných kružnic jako u pevného disku, ale do jedné dlouhé spirály, která začíná u středu média a rozvíjí se postupně až k jeho okraji
záznam je pouze na spodní straně disku
u klasických CD mechanik je konstantní rychlost čtení, ale rychlost otáčení CD-ROM disku se různí – je kontinuálně přizpůsobována podle toho, zda se čtení provádí blíže kraji nebo středu disku (u středu je rychlost otáčení vyšší na okraji naopak nižší).

Záznam a čtení

Záznam se provádí v podobě prohlubní a ostrůvků, **pitů** a **polí**, což reprezentuje jedničky a nuly.
Při čtení dopadá laserový paprsek celou tloušťkou průhledného polykarbonátu, odráží se od vnitřní strany kotoučku a znovu prochází celou jeho tloušťkou ke čtecímu senzoru. Prochází tedy celé médium dvakrát. Pokud se v místě odrazu svazku od horního povrchu nachází pit, dojde k částečnému rozptýlení odrazu, a to vyvolá na čtecím senzoru jinou odezvu, než v případě odrazu čistého povrchu.
Protože šířka stopy spirály je velmi malá, data jsou uložena s poměrně velkou hustotou a vlastní CD-ROM nosič není ničím chráněn, je velká pravděpodobnost, že i při běžné manipulaci může dojít ke špatnému přečtení některých uložených bitů. Proto informace uložené na médiu CD-ROM jsou silně redundantní (nadbytečné) a mechanika má obvody realizující na základě těchto nadbytečných informací poměrně složité algoritmy pro korekturu chyb vzniklých při čtení.



CD-RW - povrch tepelně měnitelný → může se obsah smazat

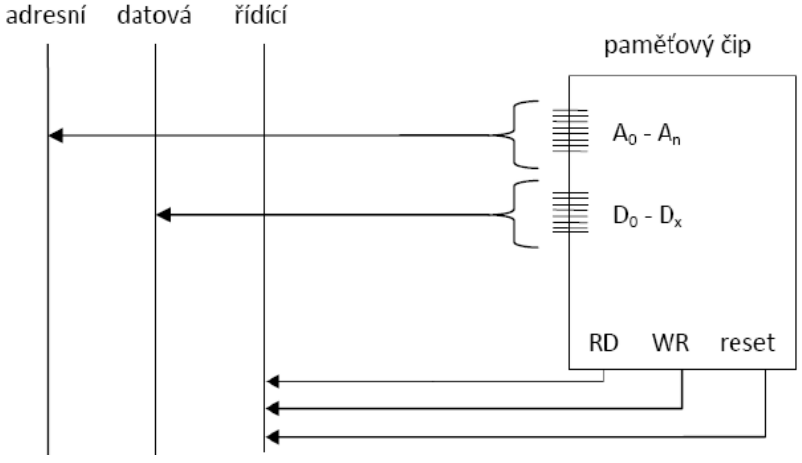
DVD – Základní rozdíl mezi CD a DVD je v tom, že CD má pouze jeden polykarbonátový disk, zatímco DVD je slepen ze dvou. Ten nejdůležitější rozdíl, je samozřejmě v kapacitě média. Ta je dána hustotou drážek na polykarbonátovém kotouči.

Sběrnice

Pod pojmem sběrnice obecně rozumíme soustavu paralelních vodičů, která umožňuje přenos signálů mezi jednotlivými částmi počítače → komunikace a přenášení dat.

Zařízení jako jsou procesor, cache paměť, operační paměť a některá další zařízení jsou propojena tzv. **systémovou sběrnicí (CPU bus)**. Osobní počítače musí být navrženy tak, aby bylo možné jejich snadné rozšiřování o další zařízení (zvukové karty, síťové karty, řadiče disků apod.). Takovéto rozšiřování je velmi často uskutečňováno pomocí tzv. **rozšiřující sběrnice** (bus) počítače (označované pouze jako sběrnice), na kterou se jednotlivá zřízení zapojují.

Rozdělení signálu sběrnice



Rozdělení signálu sběrnice:

- ***datová*** (urč. zda adresa patří datům nebo V/V)
- ***řídící*** (READ, WRITE, RESET, přerušení, SLEEP, WAKE UP, DMA)
- ***adresní*** (urč. zda jsou data adresy)
- ***nápájení*** (+5V, -5V)

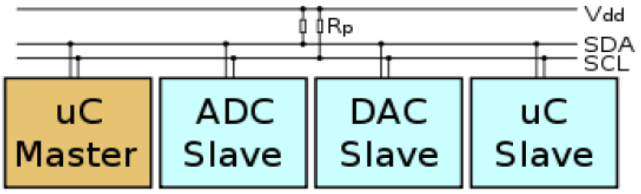
I²C Sběrnice

I²C Sběrnice slouží pro komunikaci a přenos dat mezi jednotlivými integrovanými obvody většinou v rámci jednoho zařízení. Je často používána k připojení periferních zařízení jako jsou například EEPROM, RTC, AD/DA převodníků a dalších. Je to speciální sběrnice vyvinutá firmou Philips, pro účely připojování periferních zařízení (SLAVE) k jednomu řídícímu (MASTER).

Sběrnice rozděluje připojená zařízení na

Master (řídící) – zahajuje a ukončuje komunikaci; generuje hodinový signál SCL

Slave (řízené) – zařízení adresované masterem, každé SLAVE zařízení na sběrnici má svou unikátní adresu, která je složena z pevné části vypálené uvnitř součástky a často také z volitelné části, kterou může konstruktér odlišit stejné odvody na jedné sběrnici



Umožňuje propojení až 128 různých zařízení s pomocí pouze dvou obousměrných vodičů. Jeden tvoří hodinový signál SCL (Synchronous Clock) a druhý datový kanál SDA (Synchronous Data). Přenosová rychlost sběrnice je pro většinu aplikací dostatečná i v základní verzi, kde je frekvence hodin 100kHz.
Rychlost přenosu pak musí být přizpůsobena pochopitelně "nejpomalejšímu" čipu na sběrnici.

Princip přenosu
Jeden z integrovaných obvodů (většinou mikrokontrolér) je nastaven jako MASTER a všechny ostatní obvody jsou SLAVE. Master při jakémkoli přenosu generuje hodinový signál na vodiči SCL. Když jeden čip vysílá, přijímají všechny ostatní a pouze podle adresy určují, zda jsou data určena jim. Čip, který chce vyslat/přijmout data musí nejprve definovat adresu čipu, s kterým chce komunikovat a zda půjde o příjem nebo vysílání - tedy o čtení nebo zápis. To určuje R/W (read/write) bit, který je součástí adresy.

Přenos probíhá kombinováním následujících celků
stav klidu – je zajištěn logickými jedničkami na obou vodičích, master negeneruje hodinový signál a neprobíhá žádný přenos. Logické jedničky jsou na obou vodičích zajištěny pull-up rezistory (Rp).

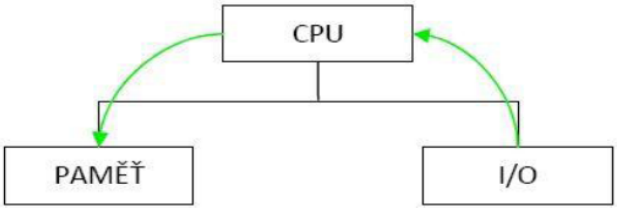
start bit – zahajuje přenos nebo jeho další část. Je vygenerován tak, že se změní úroveň SDA z 1 na 0 zatímco je SCL v logické 1.

stop bit – ukončuje přenos. Je vygenerován podobně jako start bit. Logická úroveň SDA se změní z 0 na 1 zatímco je SCL v logické 1. Stop bit může být generován pouze po "nepotvrzení přenosu", tedy pouze po přijmutí Ack v logické 1.
přenos dat - Data jsou přenášena po 1Byte tedy 8 po sobě jdoucích bitů od nejvyššího po nejnižší. Při přenosu dat se může logická úroveň na SDA měnit pouze pokud je SCL v logické 0. Při každém pulzu na SCL je přenesen jeden bit.

potvrzující bit Ack – tento bit slouží k potvrzení správného přijmutí dat. Ack bit se odesílá stejným způsobem jako by se odesílal devátý bit dat, ale generuje ho čip, který přijímal (přijímač) a nikoliv ten který data odesílal. Pokud přenos proběhl v pořádku a přijímač je připraven přijímat dalšího byte -> logická 0, pokud přenos selhal -> logická 1, pokud má dojít k ukončení přenosu, tak "neodešle nic".

Komunikace s periferiemi

DMA (Direct Memory Acess)
Pokud je třeba přenést data z I/O (z periferií a V/V zařízení) do paměti, je CPU degradováno na „přenašeče“ těchto dat.



Toto lze realizovat výhodněji formou přímého přístupu do paměti označované zkratkou DMA. Princip DMA spočívá v přímém přesunu informací bez účasti procesoru (mezi vyrovnávacím registrem periferního zařízení a hlavní pamětí), který tudíž nemusí ukončovat nebo přerušovat své aktuálně běžící programy. Jedinou nutnou podmínkou je „uvolnění— sběrnice pro procesor, to znamená, že procesor na dobu přesunu přepne všechny budiče sběrnic do třetího (vysokoimpedančního) stavu. Sběrnice nemohou zůstat po dobu přesunu bez řízení, proto existuje další blok, který je schopný generovat adresu a určovat okamžiky přesunu dat po datové sběrnici. Tento blok se označuje jako blok DMA, někdy též označovaný kanál nebo **řadič DMA**.

Jednotlivé kroky
1. nastavíme adresu, čítač a směr (do/z paměti)
2. spustíme DMA řadič
3. DMA řadič se dotáže CPU na uvolnění sběrnice

4. CPU potvrdí DMA – ACK
5. DMA řadič pošle adresu na Ao = Am, pošle signál R/W, pošle Do –Dm
6. uvolní sběrnici – DMA-REQ
7. adresa++, čítač - -
8. pokud čítač == 0 → konec, jinak zpět na 3

Monitory

Monitory jsou základní výstupní zařízení počítače. Slouží k zobrazování textových i grafických informací. K počítači je připojen grafickou kartou.

Typy monitorů

- CRT
- LCD
- Plazmové monitory
- OLED
- E-Ink

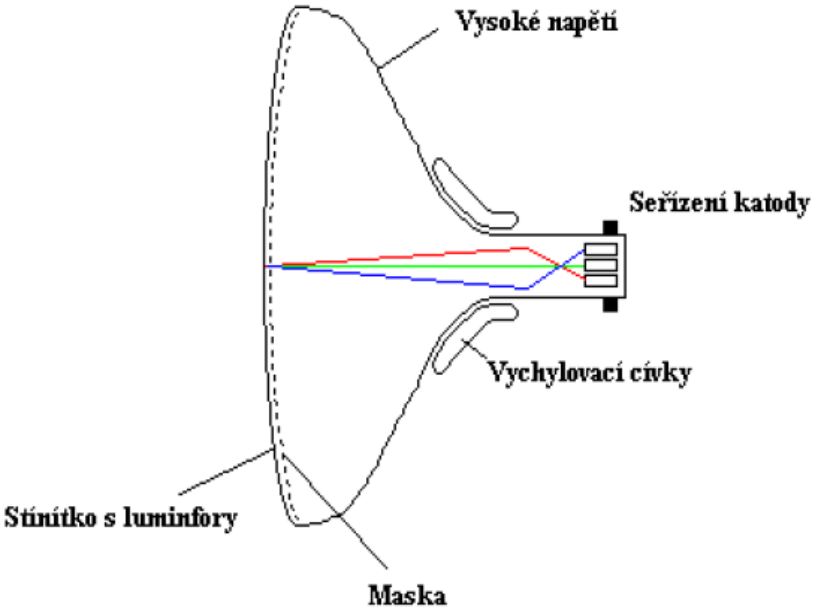
Grafická karta

Skládá se ze tří základních bloků

- grafický procesor (řídící obvod), GPU – zobrazuje data dodaná procesorem počítače, na základě dodaných dat propočítá jednotlivé body obrazu a uloží je do videopaměti
- videopaměť - sestavuje zobrazovaná data grafickým procesorem, na její velikosti závisí výsledné rozlišení,
  SEQ – kontrolér pro sekvenční adresovaní videopaměti
- D/A convertor (RAMDAC) – převádí obsah paměti na obrazovku, jeho rychlost ovlivňuje celkovou rychlost karty a obnovovací frekvenci

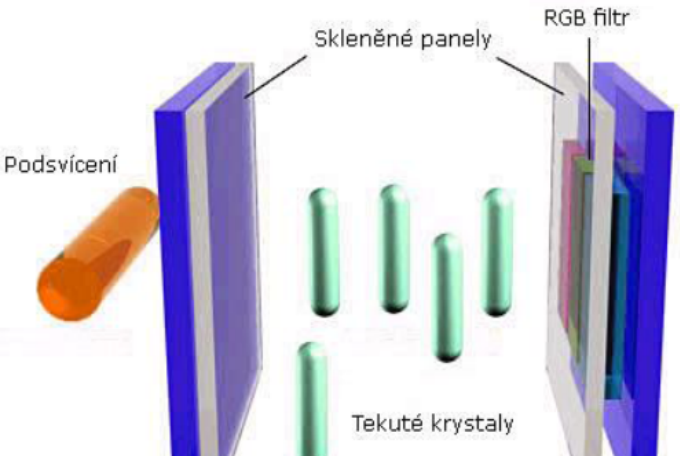
CRT monitor (Cathode Ray Tube)

CRT monitory pracují na principu katodové trubice. Hlavní částí každého monitoru je obrazovka, kterou tvoří sklenění baňka, na jejímž stínítku se zobrazují jednotlivé pixely. Monitor je připojen přímo k videokartě zasílající patřičné informace, které budou na obrazovce zobrazeny.
Při práci obrazovky jsou ze tří katod emitovány **elektronové svazky**, které jsou pomocí jednotlivých mřížek taženy až na **stínítko obrazovky**. Na zadní stěně stínítka obrazovky jsou naneseny vrstvy tzv. **luminoforů** (látka přeměňující kinetickou energii na energii světelnou). Tyto luminofory jsou ve třech základních barvách RGB. Vlastní elektronové svazky jsou bezbarvé, ale po dopadu na příslušné luminofory dojde k rozsvícení bodu odpovídající barvy.
Protože elektronový svazek je z částic stejného náboje (záporného), mají tyto částice tendenci se odpuzovat a vlivem toho dochází k rozostřování svazku. Proto těsně před stínítkem obrazovky se nachází maska obrazovky. Je to v podstatě mříž, která má za úkol propustit jen úzký svazek elektronů. Elektronové svazky jsou vychylovány pomocí **vychylovacích cívek** tak, že vzniká světelná stopa. Paprsek elektronů začne v levém horním rohu obrazovky, postupně dojde na pravý horní roh, vypne se, pak se sníží o jeden řádek a opět pokračuje zleva doprava. Obrazovka se překresluje nejmíň 75x /s což je (75Hz) může být až 100Hz.



LCD monitor (Liquid crystal display)

Je tenké, ploché zobrazovací zařízení skládající se z určitého počtu barevných nebo monochromatických pixelů seřazených před zdrojem světla (**podsvětlující katody**). Vyžaduje poměrně malé množství elektrické energie.
Každý pixel LCD se skládá z **molekul tekutých krystalů** uložených mezi dvěma průhlednými elektrodami (skleněné panely) a mezi dvěma **polarizačními filtry a barevným filtrem** (pro červenou, zelenou a modrou). Každý pixel je také aktivně ovládán jedním tranzistorem



kontrolujícím velikost napětí, které prochází mezi vyrovnávacími vrstvami.
Tato technologie je založena na elektromagnetických vlastnostech tekutých krystalů. Pomocí napětí na elektrodách jsou molekuly tekutých krystalů usměrňovány do příslušné polohy, přes které prochází polarizované světlo.
Pokud je tekutý krystal v základním stavu (bez procházejícího napětí), molekuly jsou srovnány do spirálek. V tomto případě točí polarizaci procházejícího světla o 90 stupňů, takže světlo může projít i druhým polarizačním filtrem a v konečném důsledku prochází plný jas podsvětlujících katod.
Druhý případ je kdy pixelem prochází veškeré možné napětí. Pak se molekuly srovnají vedle sebe a procházející světlo je polarizováno kolmě k druhému filtru → pohlcení polarizačním filtrem. Důsledkem toho by měl být černý pixel.
Pomocí ovlivnění stočení krystalů v pixelu lze kontrolovat množství procházejícího světla, a tudíž i celkovou svítivost pixelu. Tímto způsobem lze krystal regulovat v několika desítkách až stovkách různých stavů a tak vzniká výsledný jas barevných odstínů.

Plazmové displeje
Plazma = skupenství, složené z iontů a elementárních částic. Není to plyn, kapalina ani pevná látka (proto se někdy nazývá čtvrtým skupenstvím)
Celý plazma displej je tvořen maticí miniaturních fluorescentních buněk, ovládaných sítí elektrod. **Řádky = adresovací elektrody, sloupce = zobrazovací (výbojové) elektrody**. Všechny pixely se u barevných plazma displejů skládají ze 3 barevných **sub-pixelů (R,G,B)**.

E-Ink
Zařízení, které k zobrazení statické informace nepotřebuje elektrický proud (označení EPD – Electronic Paper Device)
Jednotlivé body jsou tvořeny malými uzavřeným **kapslemi**. Tyto obsahují elektricky separovatelný roztok – v tomto průsvitném a chemicky stálém roztoku jsou obsaženy záporně nabité černé částice (inkoust) a kladně nabité bílé částice. Kapsle jsou umístěny mezi elektrody a když přivedeme napětí, částice se podle svého náboj přitáhnou k elektrodě s opačnou polaritou. E-Ink poskytuje velmi kontrastní obraz, který je dobře čitelný i na slunečním světle, dobrý pozorovací úhel (téměř 180°), nízkou spotřebu elektrické energie. Ovšem nenabízí velkou škálu odstínů šedi (jen 16 odstínů) a také je E-Ink nevhodný k zobrazování videosekvencí.
Existujei **barevný E-Ink**, stejná metoda jako u LCD (barevné filtry – RGB filtr umístěný před každou kapslí). Barevná hloubka 4096 barev.

CUDA technologie
CUDA (**C**ompute **U**nified **D**evice **A**rchitecture) – GPGPU (**G**eneral **P**urpose **G**raphical **P**rocessing **U**nit)
- od 1970 → 1980 firma IBM (éra karet se svou vlastní GPU)
- 1993 – založena NVidia
- 1994 – 3dfx
- 1996 – VooDoo Graphics
- 1999 – GeForce 256 geometrická transformace objektů
- zlomový bod 2000/2002 GeForce 4 (pixel/vertex shadery) → GeForce 8 → **únor 2007 (CUDA)**
- 2008 GeForce 280
- 2010 GeForce 480 – první GPU postavené pro **obecné výpočty GPGPU**

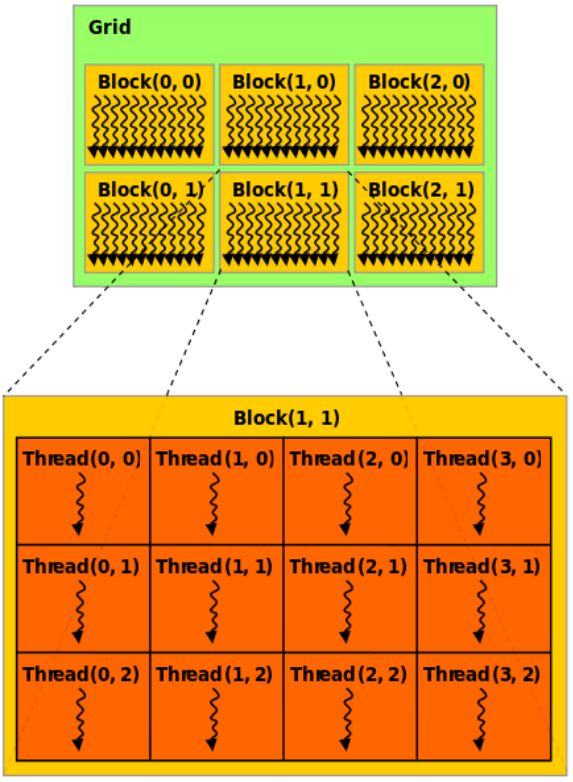
Výhody
- GPU může zpracovávat stovky vláken (virtuálně i statisíce vláken)
- vlákna jsou nezávislá – není zaručeno pořadí, v jakém budou zpracovávána
- vhodné pro intenzivní výpočty s malým počtem podmínek
- optimalizováno pro sekvenční přístup do paměti, přenosová rychlost = stovky GB/s

Podstata GPU
- CUDA = programátorské rozhraní nad grafickými kartami NVidia (únor/2007)
- založeno na jednoduchém rozšíření jazyka C/C++, lze programovat i v jazyku Fortran
- je nutné znát GPU, aby byl kód co nejefektivnější

Komunikace mezi CPU a GPU
- komunikace přes PCI-Express je velmi pomalá (5 GB/s)
- při časté komunikaci použijeme pipeline (= pipelining)

Efektivní kód
- redukovat data, která potečou mezi CPU a GPU
- optimalizovat přístup do globální paměti
- volit správnou velikost bloku vláken

Hierarchie vláken



- „masivní paralelismus“
- grid = mřížka, která do sebe shlukuje hromadu bloků (1D – 1rozměrné pole, 2D – 2rozměrné pole nebo i třeba 3D)
    matice, každý blok si nese pozici
    v mřížce a velikost
    každý blok má svůj rozměr, nese vlákna (každé vlákno má opět svou pozici)
    bloky musí být stejné

CUDA & C / kernel

- CUDA rozšiřuje jazyk C/C++ a umožňuje tak programátorovi definovat funkce v jazyce C/C++, nazvané kernely a když dojde k volání těchto funkcí, tak jsou kernely jsou spuštěny N krát v N různých CUDA vláknech. Právě tohoto nejsme v obyčejném C/C++ schopni dosáhnout při volání funkcí.
- abychom naznačili, že chceme, aby daná funkce byla kernelem, musí tato funkce být uvozena _ _gobal_ _ (= kontaktní blok mezi CPU a GPU)

Například:

```
// definice kernelu
__global__ void VecAdd(float * A, float * B, float * C)
{
    int i = threadIdx.x;
    C[i] = A[i] + B[i];
}

int main()
{
    ...
    // vyzvání kernelu s N vlákny
    VecAdd<<<1, N>>>(A, B, C);
}
```

Průběh výpočtu GPGPU

1. Vyhrazení paměti na GPU
2. Přesun dat z hlavní paměti RAM do paměti grafického akcelerátoru
3. Spuštění výpočtu na grafické kartě
4. Přesun výsledků z paměti grafické karty do hlavní RAM paměti