

1. Z jakých částí se skládá sběrnice a co je účelem jednotlivých částí?

Datová sběrnice – slouží pro přenos dat v počítači, data jsou vždy přenášena mezi dvěma bloky např. přenos z paměti do mikroprocesoru. Mezi hlavní parametry datové sběrnice patří šířka a časování. Šířka udává kolik bitů je schopna převést naráz.

Adresová sběrnice – pokud chce procesor číst nebo zapisovat, musí sdělit místo čtení/zápisu – toto místo je adresa, která je přenášena po adresové sběrnici. Počet bitů adresové sběrnice odpovídá počtu bitů adresy.

Řídící sběrnice – souhrn signálů, aktivní v různý čas, s různým významem a různým zdrojem. Signály jsou generovány mikroprocesorem nebo ostatními bloky.

Nejčastější signály – RESET – uvedení mikroprocesoru do výchozího stavu, Memory Read, Memory Write, Input/Output Read/Write, Ready – určuje připravenost obvodu

2. Co to je adresní dekodér a kdy je potřeba jej použít?

Při přenosu dat s mikroprocesorem je nutné rozhodnout, **které zařízení je ke komunikaci určeno**, protože paměťový prostor bývá **obsazen** více než jednou fyzickou pamětí nebo periferními zařízeními. Tento úkol právě řeší **adresní dekodér**.

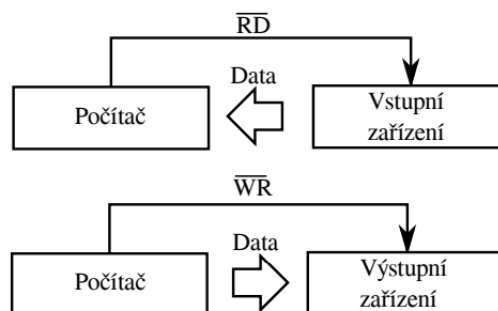
Adresní dekodér může být stavěn jako dekodér pro:

- **Úplné dekódování adresy** – každé unikátní adrese ADR_i je přiřazen pouze jeden signál $DEADR_i$ a obráceně kde $i = 1, 2, 3, \dots$
- **Neúplné dekódování adresy** – jistému signálu $DEADR_i$ přísluší adresový prostor ADR_i (je levnější ale vyžaduje pozornější přiřazení adres)
- **Lineární přiřazení adresy** – jednotlivé řády adres jsou pokládány za signály dekodéru $DEADR_i = ADR_i$ (nejlevnější dekodér ale možnost připojit pouze m přídatných zařízení u adresy s n řády)
- **Univerzální přiřazení dekódovaných adres**

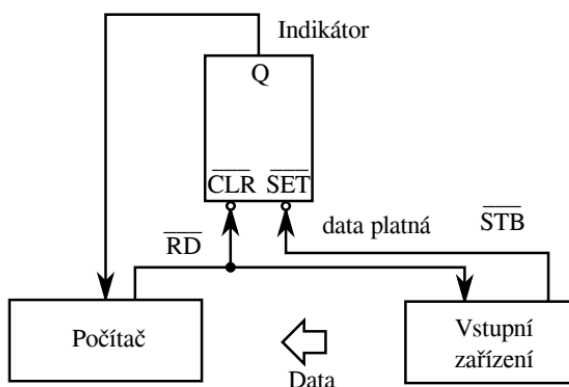
3. Jaký je princip komunikace s perifériemi pomocí V/V bran?

V/V brána zprostředkovává předávání dat mezi sběrnici počítače a perifériemi. Máme možnost použít brány s pamětí Základem bran bývá **záchytný registr (latch)** s třístavovým výstupem nebo bez paměti což jsou výkonné zesilovače (budiče) řízené jednosměrně nebo obousměrně.

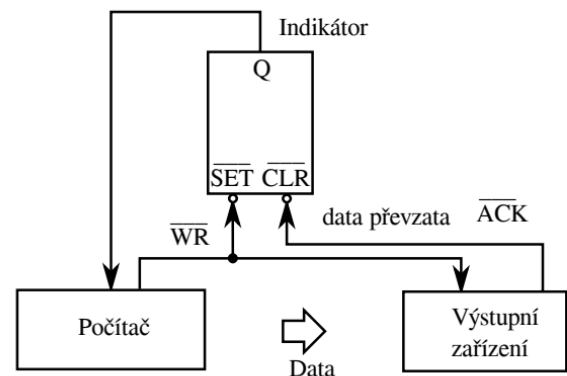
Při vstupu vyšle pc signál RD (read data), čímž přikáže vstupnímu zařízení předat data do **vstupní brány**. Při výstupu počítač vyšle současně data i signál WR a výstupní zařízení převezme data.



Obrázek 5: Technika nepodmíněného vstupu a výstupu dat



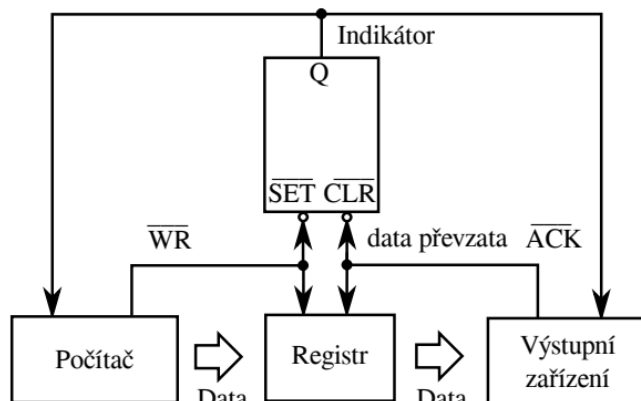
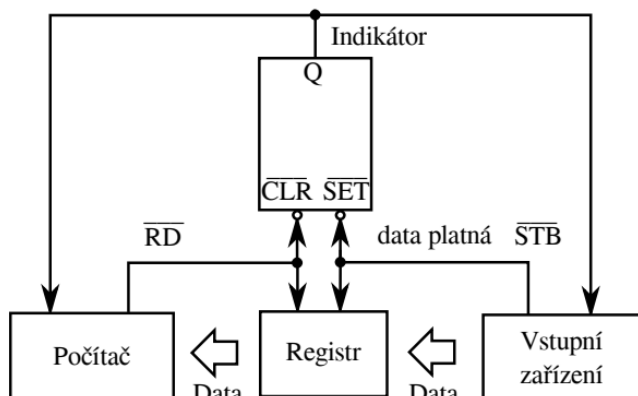
Obrázek 6: Technika podmíněného vstupu dat



Obrázek 7: Technika podmíněného výstupu dat

Jsou-li poskytována platná data ze vstupního zařízení pak se pomocí impulsu STP (strobe) nastaví $Q=1$. Q je označován jako indikátor (flag). Pokud je $Q=1$ platný, jsou předány data počítačem pomocí impulsu RD. Po ukončení je indikátor vynulován.

Při výstupu pošle počítač impuls WR pro přepis dat do výstupního zařízení. Výstupní zařízení po převzetí dat impulsem ACK nastaví indikátor na nulovou hodnotu. Touto hodnotou je počítači sděleno že může vyslat další data.



Obrázek 8: Technika vstupu dat s vyrovnávací pamětí Obrázek 9: Technika výstupu dat s vyrovnávací pamětí

Při vstupu dat s vyrovnávací pamětí semafor informuje procesor o připravenosti dat ve vyrovnávací paměti. Pro vstupní periférii tento semafor představuje informace, zda je možno do vyrovnávací paměti zaslat další data, či obsah paměti nebyl ještě procesorem přečten.

U výstupu se význam signálu semaforu pro procesor a periférii opačný než v předchozím případě.

4. K čemu slouží u komunikace V/V bran indikátor a jaké přináší výhody?
5. Popište, jak probíhá přenos dat pomocí V/V brány s indikátorem.

Pomočí indikátoru jsou sdělovány informace pouze počítači o zahájení nebo ukončení přenosu a kdy vysílač dat je povinen data udržovat.

Je to jednosměrný přenos – jednodušší na konstrukci, méně částí

Indikátor - flag (Q) se při vstupu dat nastaví na Q=1 pokud jsou poskytována platná data. Pokud je tento stav platný jsou data předány a po uskutečnění přenosu je indikátor vynulován.

Při výstupu počítač pošle impuls pro přepsání dat do výstupního zařízení a nastavení indikátoru. Zařízení po přepsání dat indikátor nastaví na nulu a tím řekne, že je možno vyslat další data.

6. Jaký je rozdíl mezi programově řízenou komunikací s perifériemi a pomocí přerušení?

Programově řízená komunikace (pooling) – komunikace s procesorem a perifériemi je řízena programovými prostředky.

Program pomocí stavové vstupní brány testuje zda a která vstupně/výstupní zařízení jsou připravena přijmout nebo odeslat data.

Při zjištění připravenosti procesor volá V/V podprogram (driver) a ten zajistí vstup/výstup dat.

Je výhodné z hlediska úspory obvodu avšak je pomalé – počítač je zatěžován testováním periodických stavů.

Přerušením (interrupt) – Jakmile periferní zařízení požaduje obsluhu od počítače, zašle mikroprocesoru přerušovaný signál. Po zjištění žádosti o obsluhu procesor přeruší právě probíhající program a přejde do obslužného programu. Po dokončení se procesor vrátí zpět do hlavního programu a pokračuje v přerušené práci.

7. Jaké výhody přináší řízení komunikace s využitím přerušení?

Především usnadňuje řízení spolupráce s periferním zařízením ve fázi zjišťování žádosti o obsluhu.

Avšak u tohoto řízení nastává problém při obsluze více periferních zařízení – je potřeba rozsáhlejšího přerušovacího systému.

8. Z jakých částí se skládá řadič DMA?

Čítač přesunů – počet slov, které mají být přesunuty v rámci jednoho spojení mezi hlavní pamětí a periférií

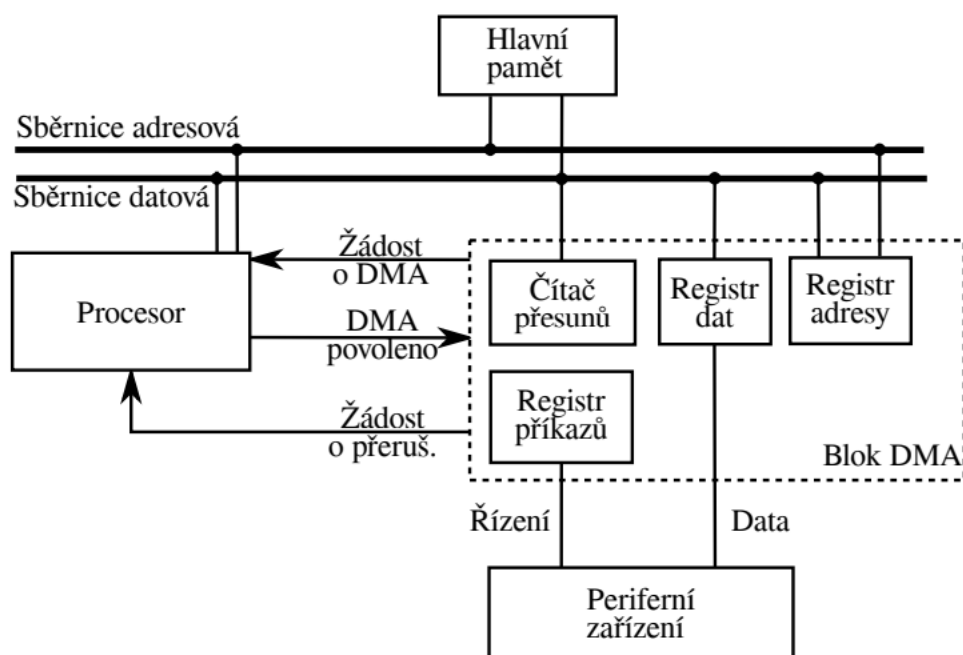
Registr dat – obsahuje slovo, které má být přesunuto z periferie do hlavní paměti nebo naopak

Registr adresy – slouží pro uchování adresy hlavní paměti na kterou má být slovo zapsáno nebo přečteno

Registr příkazů

9. Jak probíhá přenos dat s použitím DMA?

1. Procesor naprogramuje řadič DMA
2. DMA spustí periférii a čeká až bude připravena přijmout/vyslat data
3. Procesor ve vhodnou chvíli zareaguje na požadavek DMA
4. a) Procesor pošle DMA info o povolení přenosu a uvolní sběrnici
b) DMA pošle adresu na adresovou sběrnici, data na datovou sběrnici a čeká na přenos
c) DMA inkrementuje registr adresy a dekrementuje čítač přesunů
d) Pokud není čítač přesunů nulový, DMA zkontroluje zda-li má zařízení připravena další data a pokračuje od b) jinak předá řízení CPU



Obrázek 10: Struktura bloku pro přímý přístup do paměti (DMA)

Von Neumann

- skládá se z řídicí a logické jednotky (cpu), paměti a vstupu/výstupu – toto celé tvoří počítač, pokud něco chybí nelze říct, že se jedná o funkční počítač
- pro uložení instrukcí, čísel, znaků, adres se používá dvojková soustava
- počítač je programován obsahem paměti
- instrukce jsou procesorem zpracovány sekvenčně – pokud chceme změnit pořadí zpracování používají se skoky
- následující krok je závislý na předchozím
- počítač je univerzální a není závislý na typu řešené úlohy
- paměť je organizována do buněk stejné velikosti (většinou 1B) a očíslovány jsou adresami
- počítač je zařízení co zpracovává instrukce a pracuje s daty

Výhody

- pouze 1 sběrnice a paměť – jednodušší na výrobu
- uložení dat a programu určuje programátor – neplýtvá se jí
- jednotný přístup po 1 sběrnici

Nevýhody

- jedna paměť pro data i programy – hrozí přepsání
- jediná sběrnice tvoří úzké místo

Harvardská architektura

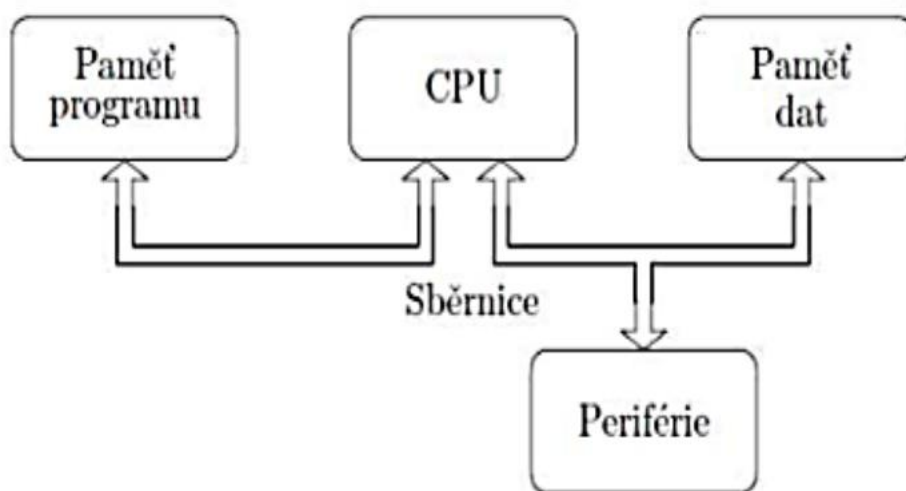
- 2 sběrnice a oddělené paměti pro data a program
- jinak princip stejný jako u Von Neumanna

Výhody

- dvě sběrnice umožňují současně přistupovat pro instrukce a data
- oddělené paměti znamenají že nehrozí přepsání programu
- paměti mohou být vyrobeny odlišnými technologiemi

Nevýhody

- nelze použít paměť pro program na data a naopak
- dvě sběrnice a dvě paměti – nákladnější a složitější konstrukce



Dnešní pc

- řadič, ALU je na jednom čipu (společně s cache a někdy grafickým čipem – APU)
- více jader – podpora paralelismu
- CPU umožňují zavádět do paměti jen potřebné části (on-demand) a tak jde optimalizovat lépe paměť

RISC

- RISC procesory obsahují redukovanou sadu instrukcí (ty základní)
- složitější instrukce se pak tvoří kombinací těchto instrukcí
- čtení a zápis do paměti provádí dvě specializované instrukce STORE, LOAD
- instrukce mají pevný formát a délku, který vymezuje význam jednotlivých bitů
- je použit vyšší počet registrů
- Nevýhody – základní instrukce – délka programů se zvětšila – to se ale v praxi nepotvrdilo

Využívá zřetězení instrukcí (pipeline)

- instrukce je rozdělena na více kroků, které jsou zpracovány samostatnou jednotkou - zásadním způsobem zrychluje práci procesoru
- přínosem je zpracování více instrukcí najednou (jedna se dekóduje, druhá se načítá, třetí se vykonává)
- jednotlivé fáze zpracování by měli být stejně časově náročné – nejpomalejší článek bude brzdit všechny ostatní

Datové hazardy

- když rozpracovaná instrukce potřebuje data předchozí instrukce ale ta ještě nejsou k dispozici – tyto problémy se řeší už v překladači

Strukturální hazardy

- problém omezených prostředků procesoru a počítače jako celku
- např. komunikace po sběrnici, která bývá pouze jedna, nelze povolit více jednotkám přistupovat k sběrnici současně – přístup je tedy potřeba koordinovat – přináší zpomalení práce

Aritmetické zřetězení - součet čísel s plovoucí řádovou čárkou – porovnání exponentu, posun, sečtení normalizace výsledku

Instrukční zřetězení – načtení instrukce, dekódování, výpočet adresy operandu, výběr operandů, provedení operace, zapsání výsledku (retire), úprava IP

CISC

Větší počet instrukcí, nejednotná délka, instrukce řeší několik operací naráz, dnes nelze říct že je procesor čistě RISC nebo CISC dnešní procesory mají prvky z obou, x86 od pentia pro

Výhody – Specializované instrukce pro provedení složitých operací

Nevýhody – složitost návrhu procesoru a instrukční sady

RISC – skok, predikce skoku

větvení kódu – vzniká problém, pokud dojde ke skoku musí se vymazat celá pipeline – zpomalení – proto existují predikce skoku – předpovídají jestli se skok provede nebo ne

Bit predikce skoku – v instrukce se vyhradí 1 bit jestli se skok provedl nebo ne

Statická – bity jsou vloženy již při kompilaci nebo přímo programátorem

Dynamická – při každém skoku se zaznamená jestli se provedl nebo ne – výhodnější ale vyžaduje bit zapisovat což může být v některých případech komplikované až nemožné

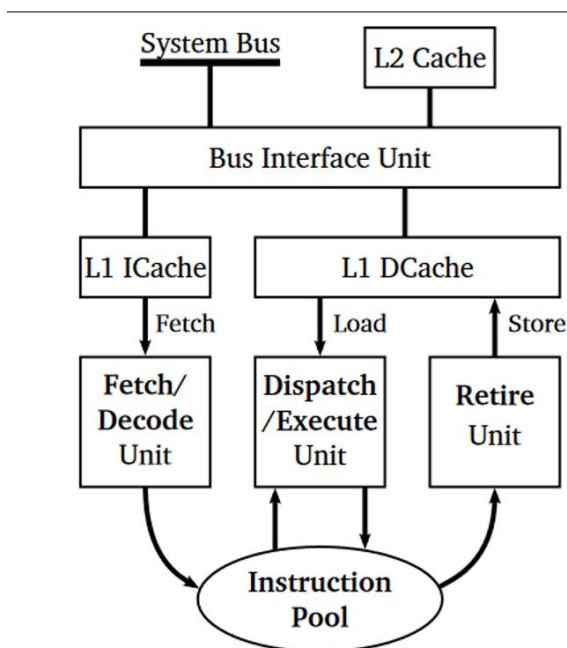
Vylepšení je predikce dvoubitová kdy můžeme uložit až 4 stavy

Zpoždění skoku – po skokové instrukci bychom měli zastavit činnost procesoru jinak se rozpracované instrukce provedou a to změní logiku našeho programu

Řešení 1 – můžeme vyplnit místo za skokem prázdnými instrukcemi NOP – logika programu se sice nezmění ale nezvýší se ani efektivita

Řešení 2 – najít v programu instrukce nesouvisející se skokem a vyhodnocení jeho podmínky a ty přeskočit a zpracovat

Paměť skoku – součást procesoru je tabulka prováděných skoků – do tabulky se ukládá adresa skoků – můžeme aplikovat s bitovou predikcí – nevyžaduje úpravu instrukcí a není součástí činnosti překladače



- 150-200 MHz
 - Datová sběrnice 64 bit
 - Adresová sběrnice 36 bit
 - 2 čipy – v jednom je jádro L1 cache v druhém L2
- Přímo na procesoru je implementována L2 cache. Fetch/Decode unit vybírá z paměti instrukce a dekoduje je na RISC – dále jsou inst. ukládány do Instr. poolu (až 40) odkud si je může dispatch/execute unit vybírat v libovolném pořadí out-of-order. Provedené instrukce jsou opět ukládány do inst. poolu odkud jsou vybírány retire unit. Dále pak do registrů a paměti L1, potom přes rozhraní sběrnice do L2 a do paměti pc.

DRAM

Data jsou uloženy ve formě náboje v kondenzátoru (C) – z důvodu miniaturní velikosti si informaci neuchová napořád je proto nutné obnovovat napětí (C) – občerstvení – obnovují se celé řady – jsou implementovány speciální obvody

Princip – adresní multiplexing – je kontrolován řídícími signály RAS, CAS

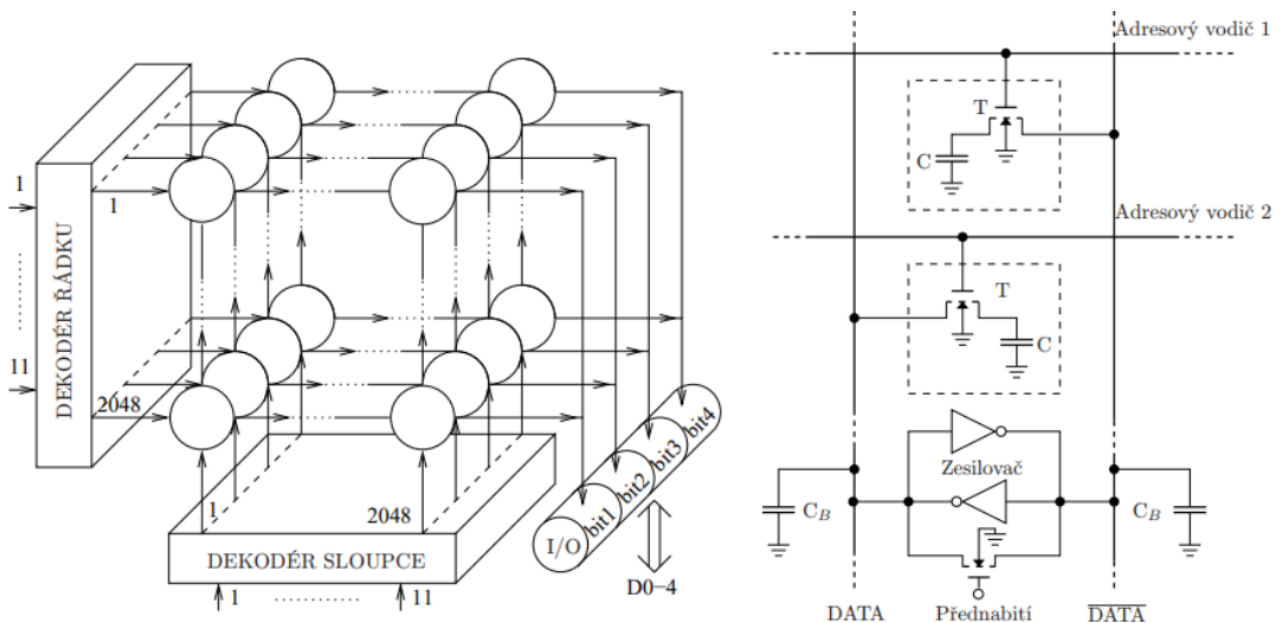
Paměťový kontrolér pošle adresu řádku a zároveň aktivuje RAS. RAS informuje čip, že se jedná o adresu řádku. Nyní kontrolér aktivuje adresový buffer aby získal adresu a přesune ji do dekodéru řádku – dekóduje ji. – stejný postup u adresy sloupce (CAS). Adresovaná buňka předá na výstup uložená data, která jsou zesílena a přesunuta do I/O bufferu. I/O poskytne informace jako Dout přes datové vývody. Čtení je destruktivní, proto musí být buňky řádku občerstveny.

Zápis – kontrolér aktivuje WE signál a přesune data Din do I/O bufferu. Pomocí čtecích zesilovačů je informace zesílena, přesunuta od adresové paměť. buňky a v ní uložena.

Stránkování – do buňky je nejprve zadána adresa řádku. Pokud se další přístup do paměti vztahuje na stejný řádek ale jiný sloupec, není nutné dekódovat adr. řádku ale pouze sloupce. – přístupová doba je kratší až o 50%

Během vývoje byly zavedeny různé formy organizace např. jeden Mbit čip má organizaci 1Mword na 1 bit tzv čip se skládá z 1M slov o šířce 1bit na každá vývod

jeden Mbit chip ma organizaci 256 Kword x 4 bitova organizace tzv. maji 4 vyvody.



SRAM

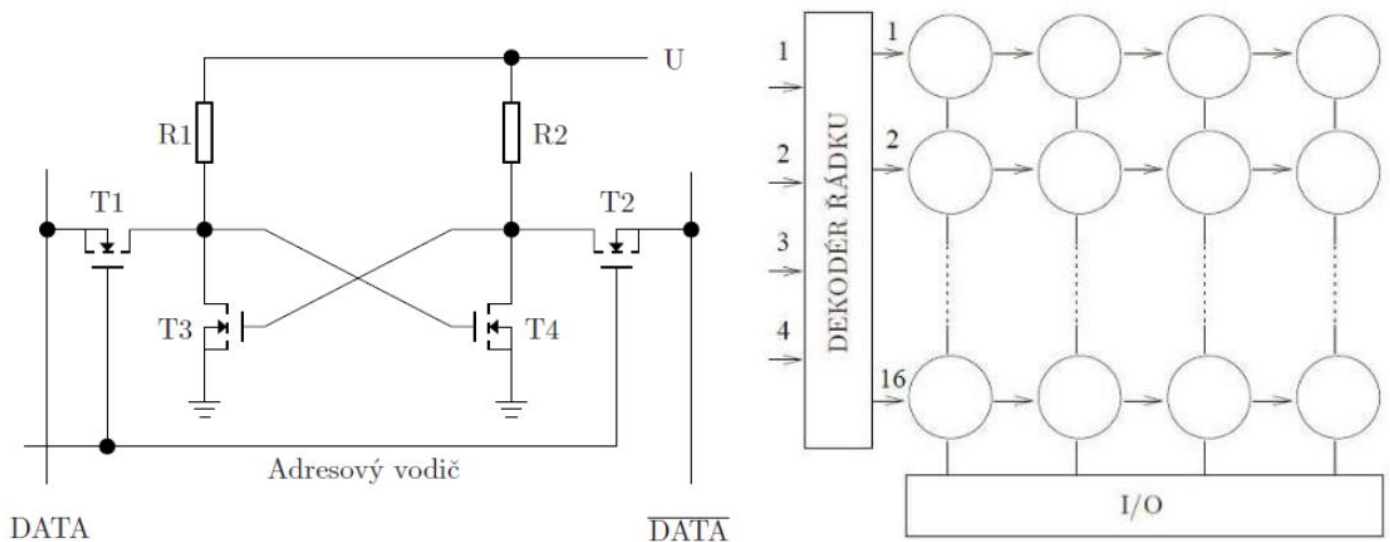
Informace je uložena stavem klopného obvodu. Realizace pomocí 4 nebo 6 tranzistorů. Buňky jsou organizovány do 2D mřížky. Paměť není potřeba občerstvovat – stav klopného obvodu je udržen dokud je SRAM čip napájen. Rychlost přístupu je v ns.

Je dražší a v porovnání s DRAM - jednotky až desítky GB je fyzicky větší na to, kolik může pojmout dat (SRAM – jednotky MB). Proto se používá u malých a rychlých cache.

Princip – nepoužívá se adresový multiplexing

Při čtení/zápisu SRAM aktivuje dekodér řádku odpovídající adresový vodič. Při čtení se přístupové tranzistory T1, T2 zapnou a propojí klopný obvod s vodiči DATA a DATA. Po stabilizaci dat vybere dekodér sloupec a předá na I/O data a tím do vnějších obvodů.

Při vstupu se přes I/O zapisovaná data zavedou na čtecí zesilovač. Aktivuje se dekodér řádku a adresní vodič aktivuje přístupový tranzistor. Datovým vodičům DATA se poskytne datový signál, který odpovídá zapisovaným datům.



1955 – feritové paměti na principu zmagnetizovaných feritových jader

Bubnové paměti – magnetický materiál byl nanesen na mag. buben, který se otáčel vysokou rychlostí

Bublinové paměti – mag. paměti na principu velkokapacitních magnetických posuvných registrů

Polovodičové – jednobitové a vícebitové posuvné registry

1970 – DRAM, 1971 SRAM

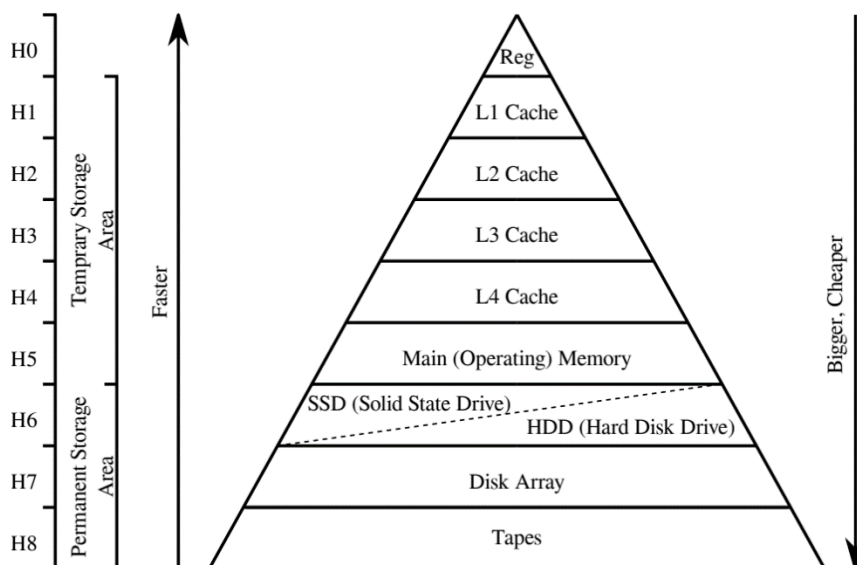
Dělení

Podle přístupu – RAM (random), SAM (serial), Speciální přístup (fronta, zásobník)

Podle možnosti zápisu/čtení – RWM(Read Write Mem), ROM(Read Only Mem), WOM(Write Only Mem), WORM(Write Once-Read many times) – optické

Podle přístupu elementární buňky – SRAM(statické), DRAM(Dynamické), ,,PROM, EEPROM, EPROM, FLASH – programovatelné

Podle uchování informace po odpojení napájení – Non Volatile (xxROM), Volatile (DRAM,SRAM)

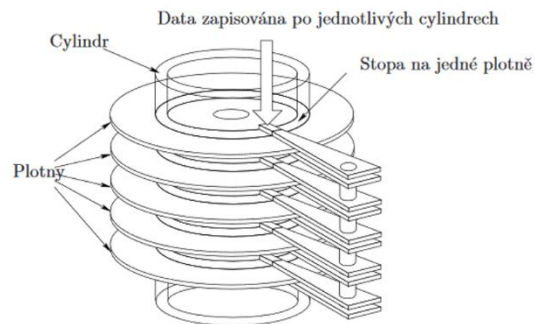
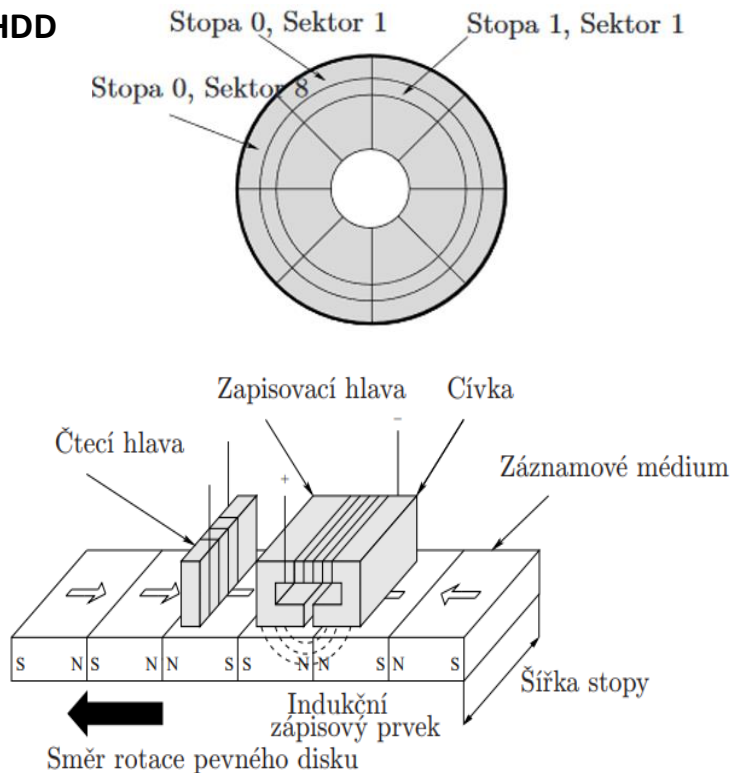


EPROM – informace se uchovává pomocí elektrického náboje, ten kvalitně je izolovaný a tak uchovává informaci i po odpojení napětí. Naprogramování je pulsem o délce 50 ms (5V). Je možné ji vymazat ultrafialovým zářením. Doba pamatování je 10-20 let.

EEPROM – programování stejné jak u EPROM. U vymazání se obrátí polarita pulzu. 10-20let

FLASH – podobná struktura jako EEPROM. doba uchování 10–100let. Mazání a programování je pulzem trvající 10 ns a napětím. Možnost vykonat 10000 a více programovacích a mazacích cyklů.

HDD



- data jsou ukládána v bajtech
- bajty jsou uloženy do skupin po 512
- tyto skupiny jsou sektory – sektor je nejmenší jednotka dat, která lze na disk zapsat nebo číst
- sektory jsou seskupeny do stop
- stopy jsou uspořádány do skupin nad sebou neboli do cylindrů

Princip

Data jsou uloženy na magnetickém médiu, které se pohybuje stejnou rychlostí. Zápis probíhá tak, že hlava je těsně nad médiem. Hlava obsahuje magnetický obvod – cívka, štěrba (jádro). Magnetický proud proudí obvodem a ovlivňuje i nejbližší okolí. Pokud se mění směr proudu v cívce, mění se i směr mag. toku a tím i smysl mag. vrstvy na médiu. Tímto vznikají magnetizované oblasti tím či oným směrem (log 1, log 0).

Podélný – jednotlivé bity jsou interpretovány jako malá mag. pole, které se uchovávají v podobě vektoru, který je rovnoběžný s plotnou – takto však dosáhneme menší hustoty uchovaných dat (150 gigabitů na čtvereční palec)

Kolmý – vektory jsou orientovány kolmo na plotnu – zvýšila se hustota uchovávaných dat až na 1 terabit na čtvereční palec – tato technologie si však vyžádala úpravu hlavy a přidání vrstvy navíc pod datovou vrstvu

Optické disky

Data jsou na disk zaznamenávána do stopy mající tvar spirály. Stopa je rozdělena na sektory. Sektory se dělí do rámců.

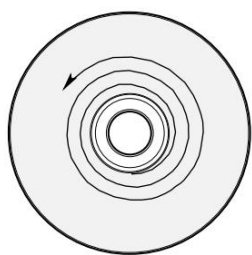
CD-ROM – Nejrozšířenější optická paměť. Data jsou zapsána lisováním podle matrice. Záznam se provádí v podobě pitů a polí. Log 1 je přechod mezi pitem a polem, log 0 se nezapisují. (650-700MB)

Při čtení prochází laserový paprsek polopropustným zrcadlem, následně je zaostřen čočkou a dopadá na povrch disku. Laser se odráží od povrchu a je snímán fotodetektozem. Pokud se v místě odrazu od povrchu disku nachází pit, dojde k rozptýlení paprsku (kvůli rozptýlení nedochází k vytvoření elektrického signálu), což vyvolá na čtecím senzoru jinou odezvu než když se paprsek odrazí od pole (na fotodetektor dopadá většina světla čímž dochází k vytvoření elektrického signálu).

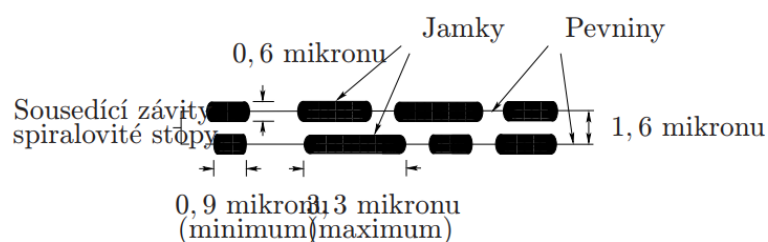
Velkým technologickým krokem vpřed byly paměti CD-WORM – na ty může zapsat uživatel jen jednou. Zápis je prováděn laserovým paprskem, který má vyšší výkon než čtecí. Paprsek zamíří na požadované místo a krátkým impulsem provede záznam dat. Impulz zahřeje a změní vlastnosti kovové vrstvy a tím vznikne obdoba pitu. Princip čtení je podobný jako u CD-ROM.

DVD – záznam stejný jako u CD (pity a pole) liší se akorát v rozměrech – jsou menší kvůli dosažení větší kapacity (4,7 – 17GB).

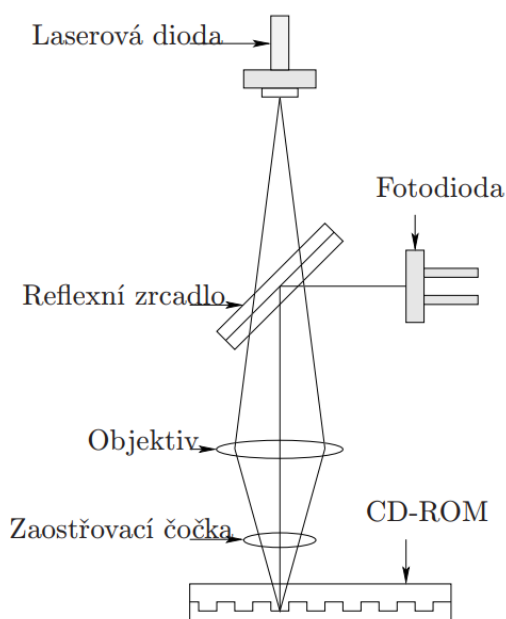
Mechaniky – slouží pro práci s optickými médii. Laser v mechanice snímá zaznamenaný vzor na disku. Laser je umístěn rovnoběžně s diskem a paprsek je na něj odrážen zrcadlem přes dvě čočky. Fotodetektor měří intenzitu světla. Mechanika musí měnit rychlost otáčení disku z důvodu spirálovitému uložení dat (spirálovitá stopa). Čte se od vnitřních stop po vnější.



Obrázek 7: Spirálovitá stopa



Obrázek 6: Jamky a pevniny u CD médií



Obrázek 5: Princip optického záznamu

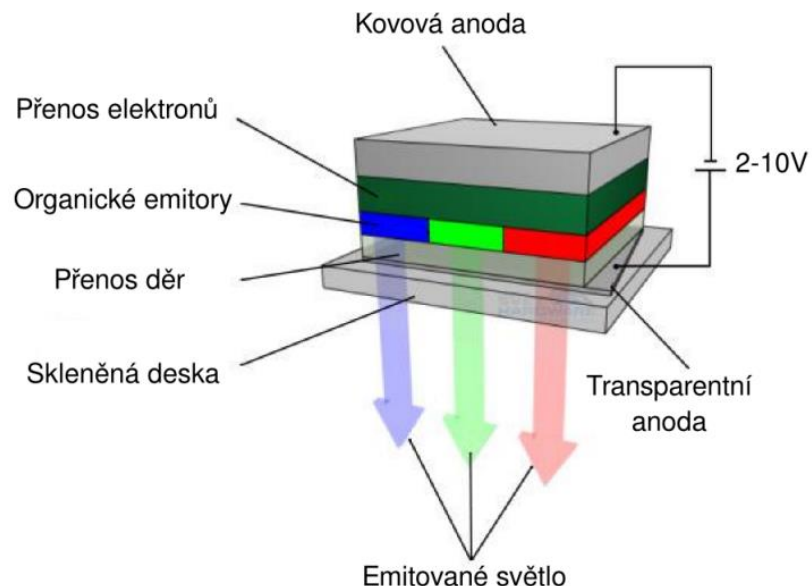
OLED

Hlavní prvek je organická dioda emitující světlo. Tato technologie selektivně vytváří světlo.

Princip – Po přivedení napětí se začnou hromadit elektrony (záporné částice) na straně blíže anodě. Díry (kladné částice) se hromadí blíže katodě. V organické vrstvě začíná docházet ke srážkám elektronů a děr a tím i k jejich eliminaci. Důsledkem tohoto je vyzáření energie ve formě fotonu. Jev se nazývá rekombinace.

Výhody – vysoký kontrast, věrohodné barvy, dobré pozorovací úhly, snadná výroba, pružný podklad, skoro bez zpoždění, velmi tenký

Nevýhody – vyšší cena, využívá se spíše na malé displeje (mobily)



LCD

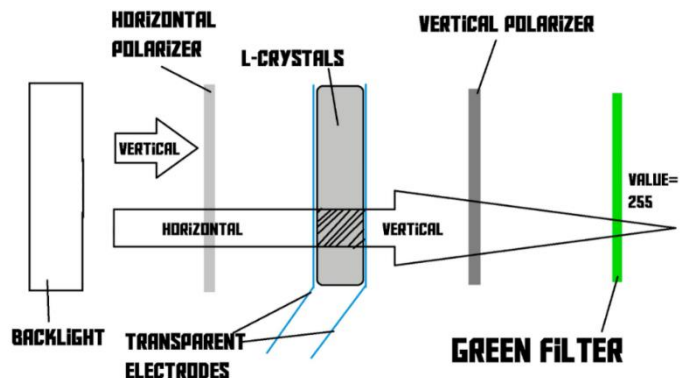
Hlavní prvek jsou tekuté krystaly. Tato technologie selektivně blokuje světlo.

Princip – nepolarizované světlo projde prvním polarizačním sklem a zpolarizuje se. Poté prochází vrstvou tekutých krystalů, které otočí světlo o 90 stupňů. Následně projde druhým polarizačním sklem které je otočené o 90 stupňů oproti prvnímu. Takto se LCD chová v klidovém stavu – propouští světlo.

Jakmile začne krystaly protékat slabý proud, začnou se orientovat podle směru proudu. Otočené krystaly tímto přestanou otáčet světlo a druhá polarizační vrstva jej nepropustí neboť je o 90 stupňů odlišná. Intenzitou proudu pak můžeme určit intenzitu propouštěného světla.

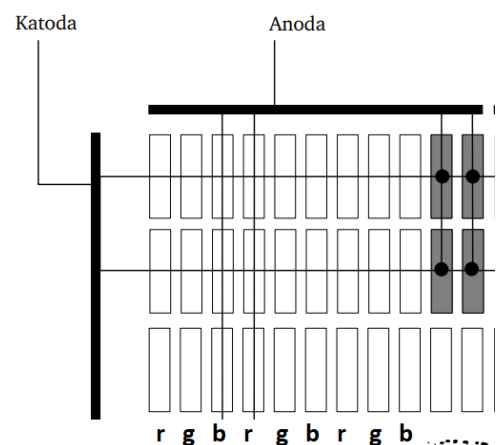
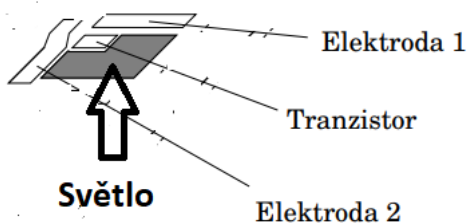
Výhody – životnost, spotřeba energie, bez emisí, kvalita obrazu

Nevýhody – ne tolik věrohodné barvy, vadné pixely, doba odezvy, pevné rozlišení



Pasivní (s pasivní maticí) – Body displeje jsou organizovány do pravoúhlé matice. Každý bod je aktivován dvěma na sebe kolmými elektrodami, které prochází celou výškou a šířkou matice. Nevýhodou je vysoká odezva – 150 – 250ms a rozostřený obraz. Použití u malých displejů (hodinky).

Aktivní (s aktivní maticí) – Každý průsečík v matici obsahuje tranzistor nebo diodu. Díky tomu je možné přesně a rychle ovládat svítivost každého bodu. Výhodou je že nevznikají problémy při částečně aktivních bodech – tranzistory izolují jeden bod od ostatních.

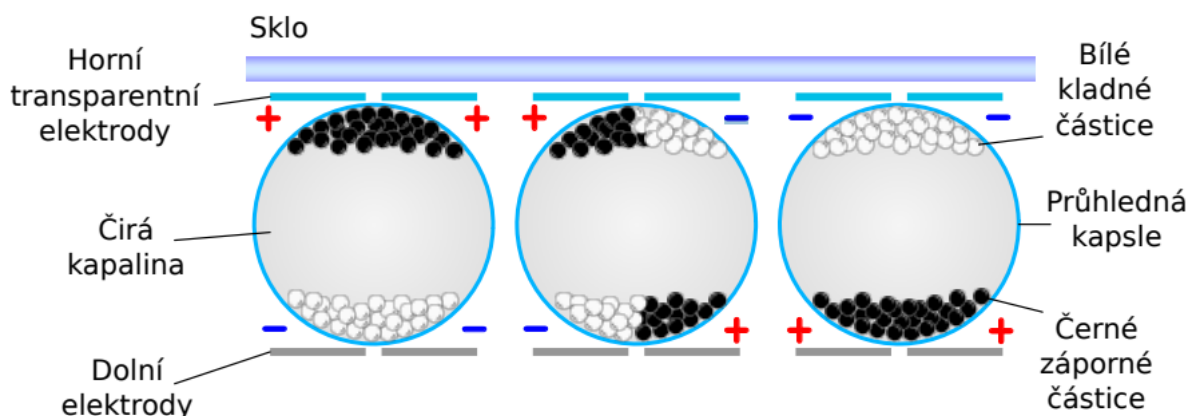


E-ink

Princip - Kapsle jsou umístěny mezi elektrody a po přivedení napětí na tyto elektrody se částice podle svého náboje přitáhnou k elektrodě s opačnou polaritou. Tímto získáme bílé bod, černé body, nebo body s odstíny šedi. Kapalina v kapslích je hydrokarbonový olej který díky svým vlastnostem zaručuje udržení částic v aktuálních pozicích i po odpojení napájení.

Výhody – dobrý kontrast, vysoké rozlišení, čitelnost na slunci, není nutné podsvětlení, dobré pozorovací úhly, tenké, malá spotřeba

Nevýhody – vysoká odezva při překreslování, malé barevné rozlišení (používají se barevné filtry jako u LCD, pouze 4096 barev), málo odstínů šedi (pouze 16)



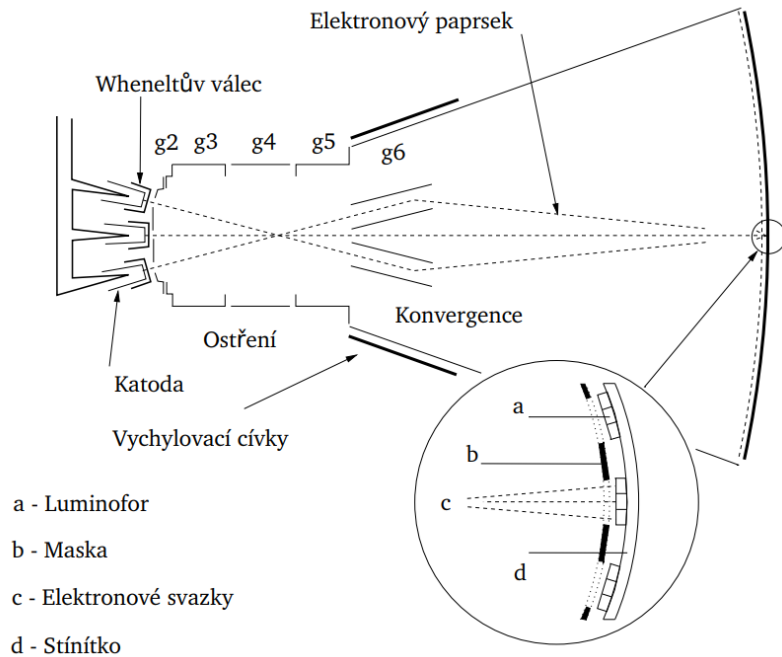
CRT

Princip – Elektronové dělo po zahřátí katody vystřeluje vysokou rychlostí proudy elektronů pro jednu ze tří (RGB) barev jejichž vlastností je záporný náboj který slouží ke správnému nasměrování částic. Elektrony cestou k obrazovce projdou filtrem (mřížka). Filtér propustí jen požadované množství elektronů a tím řídí jejich intenzitu. Elektronové svazky dále prochází Wheneltovým válcem – řízením napětí na válci se kontroluje intenzita elektronů – některé se odkloní k anodě v horní části. Elektrony dále procházejí přes mřížky (g2-g6) které je zaostří a elektronové svazky se zde postupně sbíhají. K setkání svazků dojde u masky, která propustí jen jejich část a dopadne na luminofor. Svazky jsou vychylovány horizontálně i vertikálně.

Vykreslování začíná v levém horním rohu, následně se sníží o řádek a takhle se pokračuje až do pravého dolního rohu.

Výhody – ostrost, věrohodné barvy, doba odezvy, pozorovací úhly

Nevýhody – vyzařování, spotřeba, hluk, velikost



Monolity

Počítač který má CPU, paměť a V/V periferie a tyto komponenty jsou uzavřeny v jednom pouzdře (monolitu). Používá Harvardskou arch. protože je oddělena paměť programu a dat. Převážně nese rysy RISC arch.

Paměti – dva základní typy - RWM-RAM, která se využívá pro data. Pro program se využívá nejčastěji EPROM, Flash EEPROM

RESET – uvedení mikropočítače do výchozího stavu, po provedení se nastaví počáteční hodnota čítače instrukcí (buď 0 nebo samé 1)

Organizace paměti

pracovní registry – aktuálně zpracovávaná data a výsledky operací, nejsou určeny pro dlouhodobé ukládání dat.

univerzální zápisníkové registry – ukládání nejčastěji používaných dat

paměť RWM – ukládání rozsáhlejších nebo méně používaných dat

Periferie

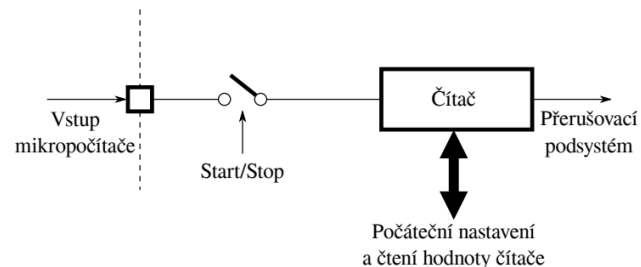
V/V brány – nejjednodušší a nejčastější rozhraní pro vstup a výstup dat (paralelní brána – port). Je organizována jako skupina 4 nebo 8 jednobitových vývodů kde lze současně číst i zapisovat log 1, 0. Lze jednotlivě nastavit které vývody budou vstupní nebo výstupní. Tím

Čítače – registr o N bitech, který čítá vnější události, při přečtení čítače se automaticky předává výzva do přerušovacího systému mikropočítače

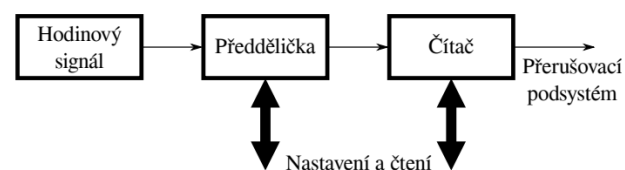
Časovač – je inkrementován vnitřním hodinovým signálem používaným pro řízení samotného mikropočítače.

A/D převodníky – do počítače vstupují fyzikální veličiny (napětí, proud, odpor) reprezentovány analogovou formou. Ke zpracování počítačem je ale potřebujeme v digitální formě, které převádí analogovou veličinu na číslicovou.

Obvody RTC – v mnoha aplikacích je potřeba přesnou časovou souvislost řízených událostí (řízení v reálném čase). RTC slouží tedy pro udržení skutečného času (hodiny, minuty, sekundy ale i dny, týdny, měsíce, roky). Proto RTC potřebuje záložní zdroj (baterii) pro počítání času po odpojení napájení. Hodnoty se na pozadí mění v čase – řeší se vhodným programovým řízením nebo pomocným registrem (do toho se uloží čas před čtením a během čtení se nemůže měnit)



Obrázek 7: Čítač vnějších událostí



Obrázek 8: Časovač

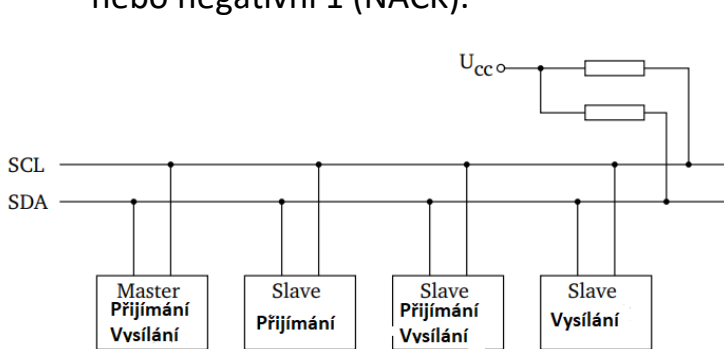
I2C

Dvou vodičová (SDA, SCL) sběrnice. SDA slouží pro obousměrný přesun dat. SCL je hodinový signál pro synchronizaci. Tyto linky pracují na logické úrovni 1 a 0. V klidovém stavu jsou na úrovni 1. Na sběrnici může být připojeno více zařízení. Jedno je označováno jako Master, které řídí komunikaci. Ostatní jsou Slaves. Slaves po zavození od Master, přijímá nebo vysílá data.

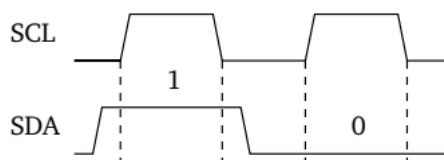
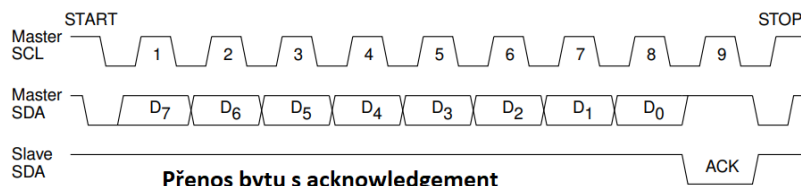
Vysílání bitů je synchronní. Synchronizaci provádí hodinový signál. Jeden bit je vyslán v jednom hodinovém cyklu. Datový signál SDA může být změněn pouze, když je SCL signál 0. Pokud je 1 a SDA je změněn, jedná se o kontrolní signál

Kvůli limitovanému počtu vodičů má I2C pouze 2 kontrolní signály (START, STOP). Sestupná hrana SDA je signál START a náběžná je STOP. Po signálu STOP je I2C v klidovém stavu.

Přenos dat je rozdělen do bytů. MSB bit přenesen první a LSB poslední. Spolehlivost přenosu je také velmi důležitá. Tohle řeší potvrzení o každém přeneseném bytu. Toto potvrzení je acknowledgement a je to devátý bit v přenosu. Může být buď pozitivní 0 (ACK) nebo negativní 1 (NACK).



Základní schéma I2C



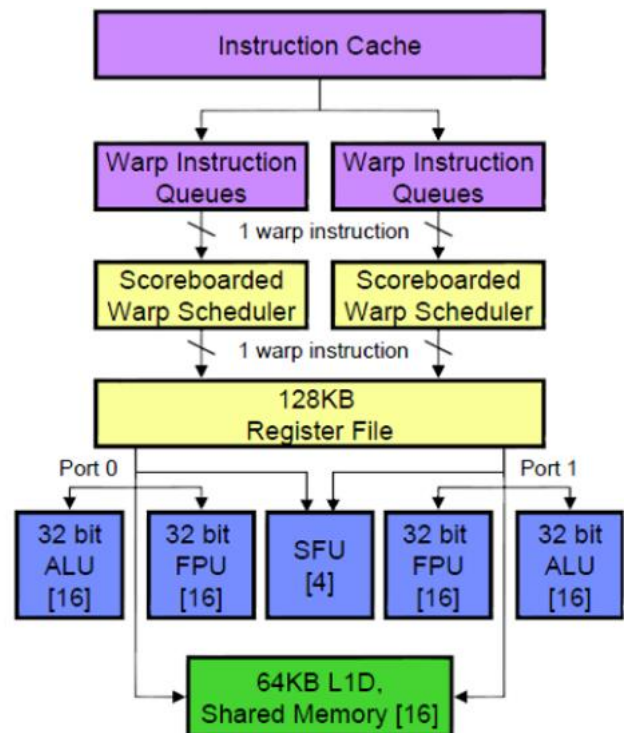
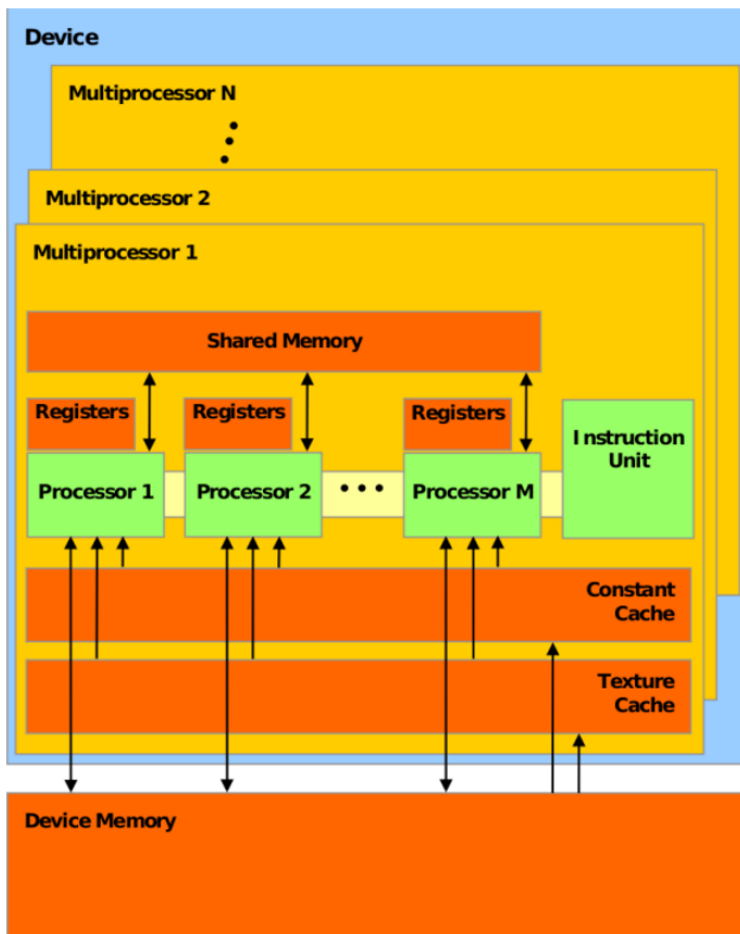
CUDA

Představena v roce 2007. Výpočetní architektura pro výpočet na grafických kartách. Díky technologii jednodušší programování - programátor nepotřebuje znát technické detaily o grafické kartě, odpadá nutnost znalosti např. OpenGL. Funguje pouze na kartách NVIDIA.

Výhody – velký výpočetní výkon (masivní paralelismus), jednotlivá vlákna běží nezávisle

Několik sdružených CUDA jader (procesorů) na multiprocesoru – podle výkonu grafické karty je multiprocesorů více. Multiprocesory mezi sebou komunikují pouze prostřednictvím paměti grafického adaptéru -> skoro o sobě nevědí, takže jsou nezávislé

Paměť grafického adaptéru se plní pouze přes sběrnici, ta je poměrně pomalá



CUDA

Je třeba mít v hlavní paměti data, která chci zpracovat (obrázek) + mít napsanou funkci, která se zavolá z procesoru a vykoná se na grafické kartě

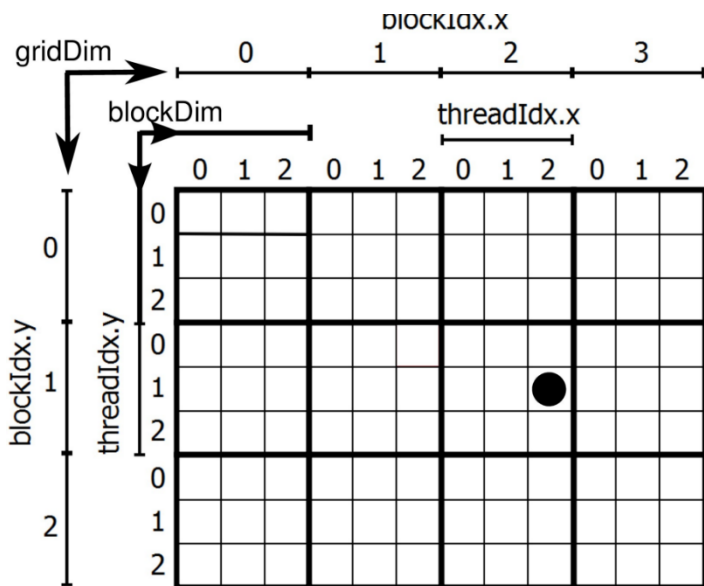
1. Přenos dat z hosta (paměť počítače) do paměti grafické karty
2. Start výpočtu na grafické kartě (spuštění vláken)
3. Průběh výpočtu (vlákna)
4. Vracení dat z paměti GPU do paměti hosta pomocí Unified Memory automatizmem nebo na požádání programátora přes funkci

Vlákna musí být na sobě nezávislé, nelze zaručit jejich pořadí vykonávání.

Efektivita výpočtu – Omezení počtu přenosu mezi device a hostem, vyhnout se používání smyček a skoků, počet vláken by neměl být zbytečně velký – nebudou využity, ideálně když GPU sdílí paměť s počítačem

Rozšíření C/C++

`__global__` - kod má být pro gpu a hlavička funkce pro CPU, `__device__`, `__host__`, `kernel` – funkce vykonávána gpu, `name<<<pocet_bloku, pocet_vlaken>>>(params)` - volání kernel



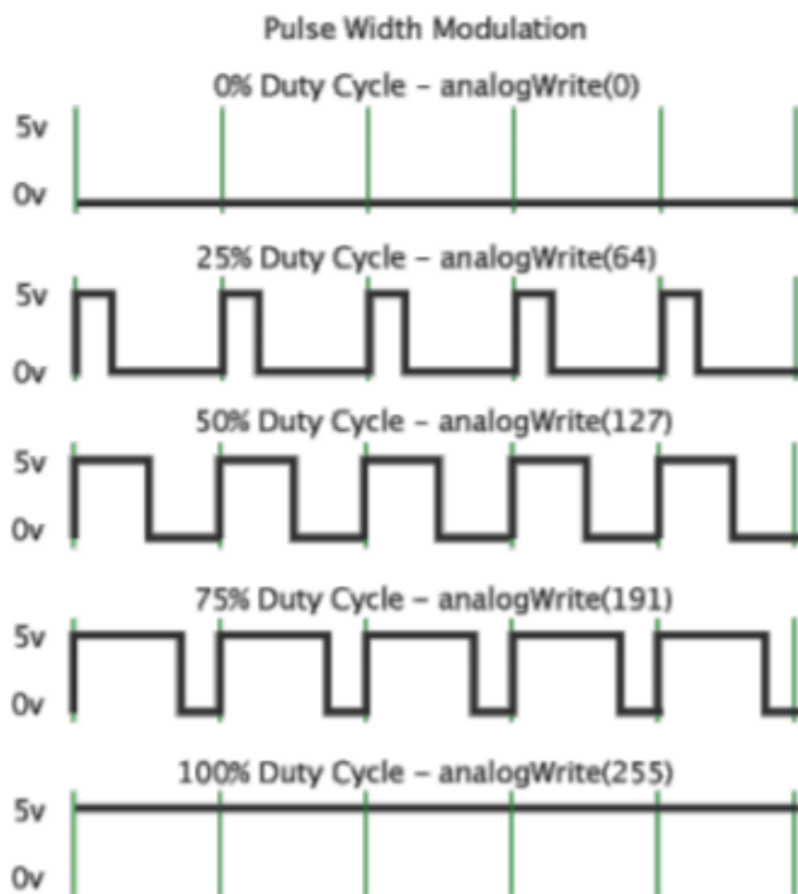
PWM

Typ modulace pro přenos analogového signálu pomocí dvouhodnotového signálu. Jako dvouhodnotová veličina může být napětí a proud. Je většinou realizováno čítačem.

Princip – signál který se přenáší může nabývat hodnot zapnuto, vypnuto (log 1, 0). Hodnota přenášeného signálu je v přenosu reprezentována jako poměr mezi stavy zapnuto a vypnuto (střída). Přenos jedné střídy – perioda. Periody se provádí většinou v ms.

Využití pro změnu jasu LED nebo regulaci otáček elektromotoru.

Log 1 je že LED svítí a log 0 nesvítí. Regulací střídy říkáme kdy je LED zapnutá a vypnutá. Tím že přenos 1 periody je ms, tak lidské oko nevidí že led bliká ale mění se pouze její jas.



ASSEMBLER

SSE – registry je možno použít ve funkcích a není potřeba obnovovat jejich obsah (XMM0-XMM15)

Pro vstupní parametry typu float, double (XMM0 – XMM7)

Při použití těchto registrů je počet parametrů uložen v AL registru.

Celočíselná jednotka

Předávání parametrů do funkce – RDI, RSI, RDX, RCX, R8, R9

Pro navrácení je registr RAX (akumulátor)

Skoky

Nepodmíněné skoky pomocí `JMP cíl`. Pro podmíněné skoky `Jcc cíl` (JGE, JG, JLE, JE).

OSTATNÍ

Amdahlův zákon – pravidlo používané v informatice k vyjádření maximálního předpokládaného zlepšení systému poté, co je vylepšena pouze některá z jeho částí. využívá se např. u víceprocesorových systémů k předpovězení teoretického maximálního zrychlení při přidávání dalších procesorů

Flynnova klasifikace – SIMD (single instruction, multiple data), SISD, MISD, MIMD, SIMT (multiple instruction, multiple thread – CUDA), MSIMD (multiple single instruction stream, multiple data stream – několik SIMD systémů pracuje nezávisle na sobě), SPMD (same program, multiple data stream – všechny procesory provádějí stejný program nezávisle na sobě)

Dělení paralelních systému podle organizace paměti

Paměť – sdílený prostor – procesy mají přístup k jedné paměti

Distribuovaná paměť – každý proces má vlastní adresní paměť, komunikují pomocí zpráv

Paměť	Ne (sdílené)	Ano (sdílené)
Ne (distribuované)	klasické 1x CPU	UMA, SMP (desítky CPU)
Ano (distribuované)	Clustery, multipočítače	NUMA

SMP (systémy se sdílenou pamětí) – paměť a procesor jsou na sdílené sběrnici, paměť je rozdělena na menší kousky, do kterých je přístupováno paralelně

DMP (systémy s distribuovanou pamětí) – k přístupu do vzdálené paměti probíhá zasíláním zpráv, nutné při větším počtu procesorů

Historie intel x86

Intel 8080 – 2MHz, dat. Sb. 8bit, adres. 16bit

8bitový procesor

8086 – 4 - 7MHz, dat. Sb. 16bit, adres. 20bit

první 16bitový procesor, dovoľoval adresovat až 1MB paměti(64kB bloky)

80186/188– 6 - 25MHz, dat. Sb. 16bit, adres. 20bit

DMA, časovače, hodiny a porty.

80286– 6 - 25MHz, dat. Sb. 16bit, adres. 24bit

Možnost taktování. Až 16MB paměti

80386DX– 16- 40MHz, dat. Sb. 32bit, adres. 32bit

První plně 32bitový procesor řady x86. Implementována Cache L1.

8087/287/387

Výpočty s desetinnými čísly

80486DX/i486DX– 16 -100MHz, dat. Sb. 32bit, adres. 32bit

Pětinásobný počet tranzistorů. L1 Cache přímo v procesoru

Intel Pentium–60 - 300MHz, dat. Sb. 64bit, adres. 20bit

Dvě paralelní ALU jednotky.

Pentium Pro

Intel Pentium II– 233 - 533MHz, dat. Sb.

Intel Pentium III– 450MHz –

Intel Pentium 4– 1,3 – 3,8GHz, dat. Sb.

Intel Pentium M– 900MHz – 2,2G

Pro přenosné počítače

Intel Core, Core Duo, Core Solo

Dvě jádra na jednom čipu

Core 2– 1 – 3,3GHz, dat. Sb. 64bit, adres.

64bitová architektura

Intel Atom– 800MHz – 2GHz, dat. Sb.

Nová architektura Bonnell. Implementována technologie Hyper Threading, směřováno na minimální spotřebu, poloviční výkon oproti Pentium M.