**Von Neumann -** 1 paměť pro data i instrukce, Instrukce sekvenčně(pořadí lze měnit skoky), dvojková soustava, struktura nezávislá na typu řešené úlohy, program dán obsahem paměti, následující krok závislý na předchozím, paměť rozdělena do buněk-pořadová čísla jako adresy, program=posloupnost instrukcí

* ALU, řadič, Paměť, I/O periférie
* *Výhody* – jednotný přístup pro data-instrukce, programátor určuje rozdělení paměti pro kód-data(neplýtvá se jí), 1 sběrnice=jednodušší výroba
* *Nevýhody* – 1 sběrnice=tvoří úzké místo, 1 technologie pro paměti, přepis kódu

**Harvardská -** oddělené paměti, CPU, Paměť dat, Paměť pro instrukce, Periferie

* *Výhody* – oddělená paměť dat/programu => nemůže přepsat sám sebe, paměť výroba různé technologie, paměť jiná velikost, 2sběrnice=paralelismu(přístup pro data-instrukce současně)
* *Nevýhody* – větší výrobní náklady, vyšší nároky na řídicí jednotkuCPU, nevyužitou paměť dat nelze využít pro instrukce a opačně

**Dnešní PC –** superskalární architektura, paralelismu, více vláken

* Rozdíl – řadič, ALU na 1 čipu(s cache někdy i grafický čip APU=CPU+GPU), více CPU na 1 čipu

**Zřetězení** – instrukce=několik kroků=>paralelní zpracování, když se instrukce nezpracovávají stejně rychle=>kroky odděleny registry=> mezivýsledky=>vyrovnávají časové rozdíly

* Aritmetické – sčítání floatu, FPU – součet 2 čísel (porovnání exponentu, posuv, sečtení…)
* Instrukcí – načtení z paměti, dekódování, výpočet adres operandů, výběr operandů, provedení operace s nimi, zapsání výsledků; větvení problem->predikce skoku
* Procesů – několik CPU provádí stejný kód nad různými daty (BubbleSort na 4 částech pole)
* *Hazardy*
  + Datové – práce s daty výsledků jiné instrukce; čeká na 2pipeline kvůli výsledku
  + Strukturální – 2pipleline přistupují najednou ke sdílené paměti, řešení - NOP
  + Řídící – při skocích umlčuje instrukce

**RISC** –70.léta, cíl redukovat složitost výroby + zpracování 1instrukce/1cyklus

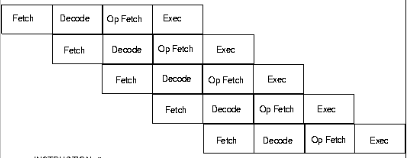
* *Výhody* - Nejzákladnější instrukce(složité=jejich kombinace), instrukce=pevná délka/jednotný formát, větší dočasná paměť(hodně registrů+cache), přístup do paměti-LOAD/STORE (načíst/uložit), zřetězení-pipeline pro zrychlení, *Zástupci* – ARM, MIPS
* *Nevýhody* – základní instrukce=větší délka kódu(v praxi NE)

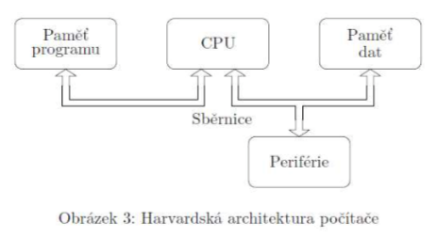
**CISC** – velké množství sad instrukcí=nejednotná délka, některé=řeší více operací naráz, malý počet registrů

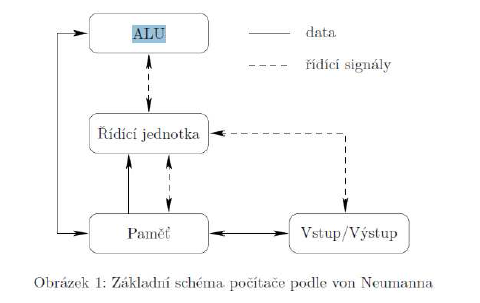
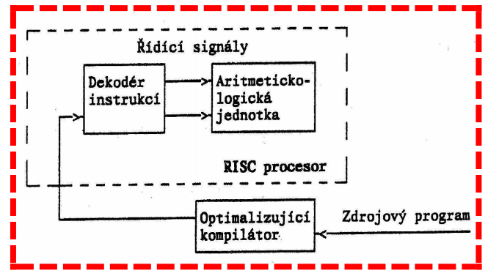
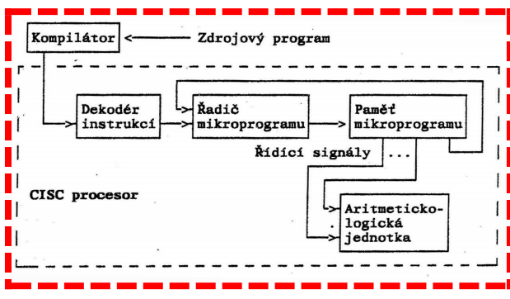
* *Výhody* – specializované instrukce pro složité úkoly
* *Nevýhody* – velká složitost(naprogramovány mikroprogramově), problémy s překladem
* Zástupci – Intel x86(Pentium Pro a dál, ve skutečnosti ale kompatibilní RISC)

**ARM** – 32bit, nejpoužívanější pro mobily, nízká spotřeba=malý odběr, LOAD/STORE, 25 registrů

* Soubor= 44 instrukcí, 2typy přerušení – FIQ(neodkladné události), IRQ(pro NE extrem fast doba odezvy), 3 stupně zřetězení, 4 režimy(uživatelský, supervizor, IRQ, FIQ), aktuální režim 2bit PSW

**MIPS** – 32/64bit, non-interlocked pipeline, LOAD/STORE, 32 registrů/32 bit, herní konzole, set-top box, tiskárny, taky mobily, kompilátor->obvod s důrazem na rychlost



****

**Rodina P6** – Pentium Pro, instrukce ukládány do Instruction pool(až 40 instrukcí i ty hotové), 4pipeline, FPU integrován na čipu, 2 čipy – 1.jádro&L1chache, 2.L2cache

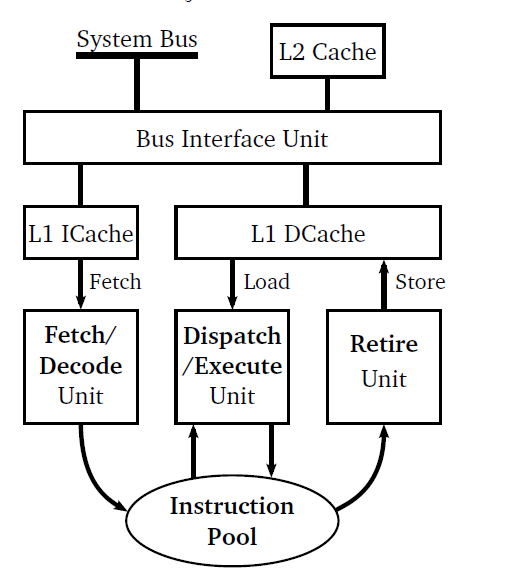
* Bus interface unit – komunikace procesoru se systémovou sběrnicí
* Paměťový subsystém–hlavní paměť,L1(interní, datová&instrukční sběrnice),L2(externí, 64bit sběrnice)
* Fetch/Decode unit – načítá CISC instrukce z L1->dekóduje na RISC mikrooperace->pool; 3dekódery(2for simple i., 1for hard), predikce skoků
* Dispatch/execute unit – 2xALU, 1xFPU, 2xAGU(jednotka generování adres), 2xFPU\_SISMD, memory interface unit(přístup do paměti), vybírá z poolu v optimálním pořadí instrukce(out-of-order=prováděné i mimo pořadí), po provedení vrácené do poolu
* Retire Unit – vybírá hotové mikrooperace z poolu, data uloženy z L1 do L2

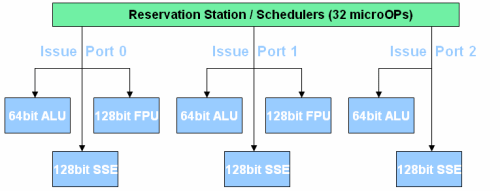
**Intel Core Duo** – data 64bit, adresy 36bit, 2006, BiCMOS(65nm), 2xFPU128+SSE(sse pracuje s obrázky), 3xALU64(komplex/simple/simple), 3xSSE128, 3xIssuesPort (načítání dat/ukládání/načítání adres) pracují s **TLB(vyrovnávací paměť)**, 2 jádra-káždé L1+společná L2, out-of-order(vykonávaní i. mimo pořadí), až 4operace/cyklus, přejmenování registrů=pomáhá paralelismu

* *Dekóder*–instrukce->mikrooperace, 3x simple dekóder, 1x komplex(až 4 mikroOP), až 7mO/takt
* *Macro-fusion* – sloučení 2 instrukcí do 1(jejich kombinace), 4x dekóder, až 5instrukcí/takt
* *Micro-fusion*- sloučení 2 mikrooperací do 1
* *Loop detector* – predikce skoku, hledá podmínky ukončující cyklus
* *Spekulativní předvykonávání* – asymetrický multithreading=pomocná vlákna dopředu vypočtou adresy a načítají data z RAM do L2 cache(prefetch) pro hlavní vlákno
* *Prefetch* – do cache načítá okolní data, kdyby je náhodou potřebovala=>zrychlení

**Intel Core i** - modularita, škálovatelnost, integrovaný paměť. Řadič, přiřazení vlastní části paměti + NUMA

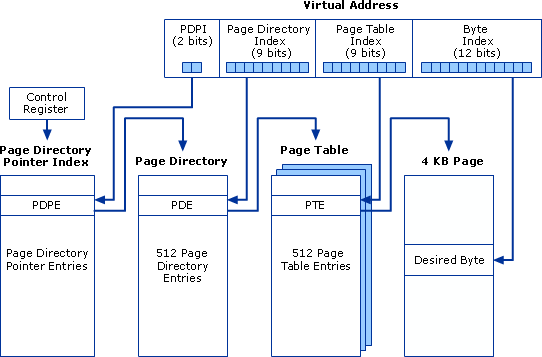
* *QPI* – rychlejší sys.sběrnice, sériová linka-2x 20bit, 12,8GB/s, až 4xQPI pro komunikaci s periferiemi
* *TurboBoost* – dočasné zvýšení taktu jader, dokud dovolí teplota-monitorováná=>bezpečná mez
* *Hyper-Threading* – lepší využití=zaměstná FPU i ALU, vše 2x=>30%nárust, 2 jádra->4 vlákna
* *Virtualization T.-* root VMM(Virtual Machine Monitor) běží na ploše a vytváří non-root VM(nový systém)->komunikuje s HW v protected reřimu
* *Digital Media Boost*–lepší výkon FPU(plovoucí desetinná čárka) a SSE(multimediální výpočty)
* *Trusted Execution T.* – větší bezpečnost=proti spyware, jedná appka nevidí co dělá druhá
* *AES New Instructions* – nové instrukce pro AES, HW podpora 6 instrukcí pro šifrování
* *SpeedStep T.*-snižuje takt procesoru pokud není využíván, snižuje hlučnost, šetří výdrž baterie u NTB
* *Execute Disable Bit* – ochrana, paměť.oblasti, ve kterých smí nesmí být spouštěn kód
* *SSE 4.2* – 7 nových instrukcí pro SSE(rozpoznání hlasu, sekvence DNA, kontrola integrity dat)

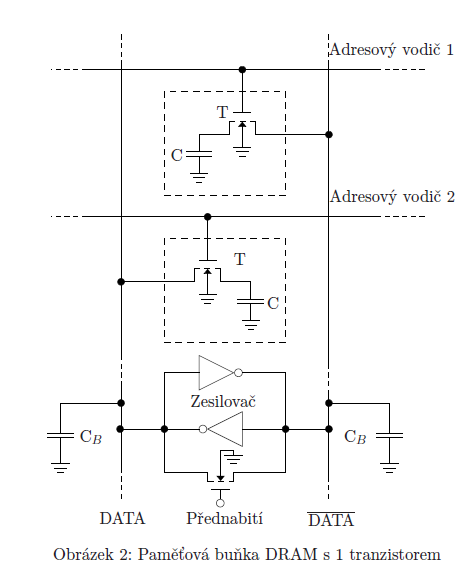
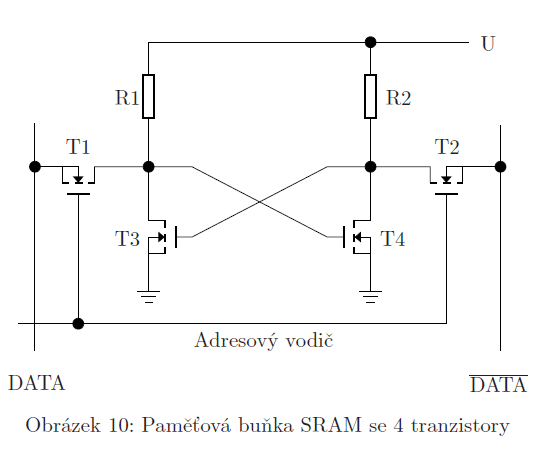
**Paměti**

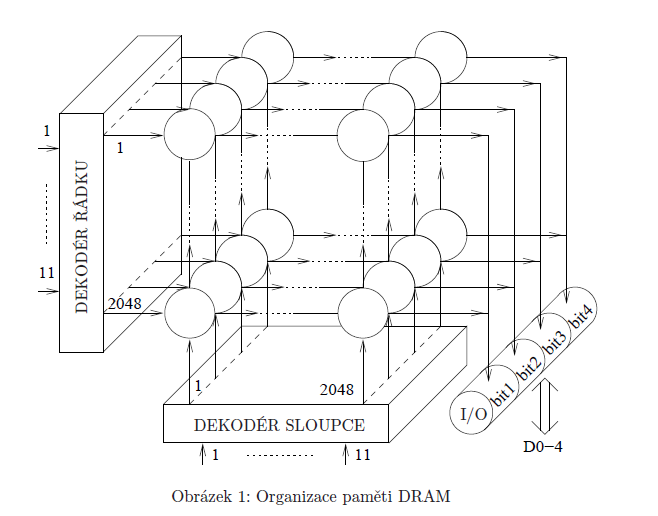
* Podle umístění – vnitřní, vnější
  + Vnitřní – uvnitř PC (registry, cache, RAM) – polovodičové SSD
  + Vnější – externí komponenty, mechanické HDD
* Podle přístupu – RAM, SAM
  + RAM – náhodný přístup, pomalejší, pohodlnější
  + SAM – seriový přístup, k datům popořadě, rychlejší – fronta, zásobník
* Podle Možnosti zápisu/čtení – ROM,RWM,NVRAM,WOM,WORM
  + ROM(čtení), RWM(čtení+zápis), WOM(pouze zápis=černá skříňka)
  + NVRAM(RWM+EEPROM=elektro.mazatelná RAM), WORM(1zápis, opakovatelné čtení)
* Podle principu – SRAM, DRAM, PROM
  + SRAM – static, malá kapacita, rychlé, klopné obvody
  + DRAM – dynamic, velká kapacita, pomalejší, kondenzátory
  + PROM–programovatelné, uživatel vypálí informace do paměti=přepálení pojistek
  + EPROM–mazatelná PROM, elektronický náboj-5V, mazání pomocí UV
  + EEPROM–elektronicky EPROM, programování a mazání elektrický puls
  + FLASH – 10-100let pamatování info, data rozdělená do sektorů; 0->paměť(0)->flash paměť
* Hierarchie – Registry, L1, L2, DRAM, HDD, SDD, CD, DVD

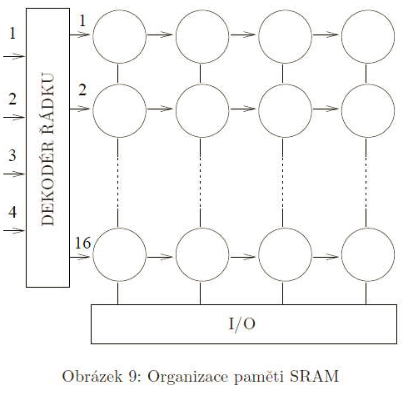
Hodnoty proměnných

* + Registry–nejrychlejší, přímo na procesoru, 10-100kyB, rychlost závisí na procesoru
  + L1 cache – v CPU, desítky KB pro data in instrukce, rychlost=1 ns
  + L2 cache – v CPU, 1-10 MB, pro data, rychlost jednotky ns
  + DRAM – desítky GiB, přístupová doba v jednotkách ns
  + HDD – TiB paměti, přístupová doba 1-10 ms
  + SDD – stovky GiB paměti, rychlost čtení/zápisu ve stovkách GiB/s
  + CD, DVD, pásky – nejpomalejší a nejlevnější
* *Dynamické p.*-uložena ve formě NÁBOJE kondenzátoru(1nabitý/0vybitý), samovolné vybíjení=>REFRESH, buňky v matici, k informacím přístup pomocí tranzistorů
  1. Poskytne se adresa řádku, tím se aktivuje celý řádek (adresní multiplexing)
  2. Přes datové vodiče, vede náboj do Cb a poté se vybere daný sloupec a ten se pošle na výstup
  3. Náboj zůstane v CB a uloží se zpět do dané buňky, nastane obnovení celého řádku
  + Page Mode – adresa 1 řádku a 4 sloupců za sebou, o 50% kratší přístupová doba
  + Extended Data Out – podržení sloupce dokud není nový, o 30% rychlejší než PageMode
  + SDRAM (synchronous) – malá přístupová doba, synchronně se sys. Frekvencí, Busrt mod=zadá se adresa řádku a sloupce a poté se načítají data z přilehlých adres
  + DDR SDRAM (double data rate)-2x přenos rychlost dat, sestupná i náběžná hrana
* Statické p.- info ve stavu kloného obvodu, 2x NMOS přístupové, 2x NMOS paměťové tranzistory
  + Dražší, méně paměti, rychlejší, lehčí čtení, u cache, buňky do matice, adresa(řádek+sloupec)
  + Asynchronní – není synchronizována se sys. Hodinama, procesor čeká na data
  + Synchroní burst – synchronizovaná se sys. Hodinami, max frekvence 66 MHz
  + Pipeline burst SRAM – požadavky zřetězeny, 75 MHz a vyšší
* Virtuální p. - přidělování paměti procesům aby využívaly celý adresní prostor + aby byla lineární, sdílení paměti, velikost stránky 4 kiB->velká->kompenzace dlouhé přístupové doby, MMU(memory managment unit)
* Virtuální adresa – rozdělena 10/10/12(řádek v adresáři stránek, řádek v stránkovací tabulce, posuv v rámci stránky)
* Translation look-aside buffer – pro urychlení, pamatuje si 256 naposledy vypočtených fyzických adres a hned je může poslat, bez nutnosti výpočtu
* Segmentace – kousek rozdělené paměti se stejnou délkou, má(začátek, limit, směr práva, atributy), adres(virtuální adresa=segment+lineární adresa), procesy k dispozici několik segmentů
* Selektor(16bit) – prvních 13bit=index tabulky deskriptorů, 1bit GDT/LDT, 2bit-oprávnění
* Deskriptor-32bit počáteční adresa, 20bit limit segmentu, typ segmentu(kódový, datový, systémový), přístupová práva(read/write/execute)

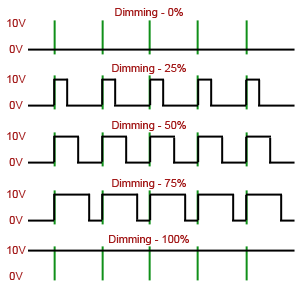








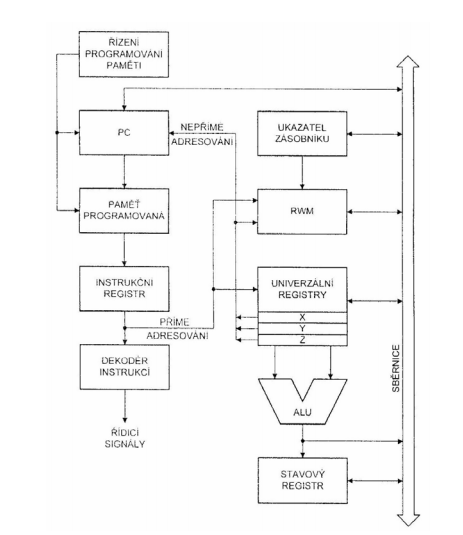
**Monolitické PC** – obsahují procesor, paměť, vstupní/výstupní periferie integrované v 1 pouzdře, RISC zjednodušené znaky, Harvardská, ROM-E(E)PROM-Flash

* *Paměť*
  + Pracovní registry – 1-2, obsahují právě zpracovaná data a výsledky
  + Univerzální registry – nejčastěji používané data, registrové banky
  + Data – RWM, rozsáhlejší=pro hodně/velká data, pro méně používanější data
* *Čítač instrukcí* – pointer na vykonávanou instrukci
* *Zásobník* – pro ukládání návratových adres, ukazatel na vrchol zásobníku je REGISTR
* *Reset* – výchozí stav procesoru
* *Ochrana proti rušení* – mechanická, elektromagnetická, watchdog(hlídá zacyklení, při po/přetečení RESET),obvod hlídající pokles napětí-brownout(při poklesu RESET)
* *Přerušovací podsystém* – dovoluje ON/OFF přerušení, detekuj IRQ(zdroj přerušení)
* *Zdroj synchronizace* – časový základ pro synchronní práci, křemen, keramický rezonátor(nejpřesnější, nejdražší RC obvod; LC obvod – levné, oscilace se mění s teplotou/U)
* *Periferie* – obvody zajišťující komunikaci procesoru s okolím
* *V/V brány*–nejjednodušší způsob komunikace, paralelní brána–4or8bit vývodů for read/write, vstupy – pull-down/up resitory pro udržení logické 0 a 1 při nezapojeném vstupu
* Čítač/časovač–inkrementované registry čítačem(vnější události)/časovačem(vnitřním signálem)

Při přetečení vyvolají přerušení

* *A/D převodníky*
  + Komparační–přímé porovnání U s referenčním U, rychlé, potřebný nárůst s vyšším rozlišením
  + Sledovací – porovnává U se vstupem, Else: zvedne nebo klesne U
  + Aproximační – půlení intervalu, max. n počet bitů rozlišení převodníku
  + Integrační – dvojí integrace, externí kondenzátor, U a -U ==0
  + S RC článkem – doba nabití/vybití kondenzátoru závisí na vstupním odporu
* *D/A převodníky*
  + PWM – výstupní U se mění díky poměru doby 1/0 na výstupu, převod zajišťuje RC
  + Paralelní – přímý převod čísla na proud přes odporovou síť R-2R(hodnoty R a 2xR)
* *Obvody RTC*- potřebují záložní zdroj(baterii), slouží k počítání času, atomicita(během čtení nelze měnit)
* *POWER PC* – 3 výkonné jednotky(IU,FPU,BPU), pracují součastně, obsahuje i : instrukční jednotka, paměťová jednotka, vyrovnávací paměť, jednotka MMU, systémové rozhraní; až 3instrukce/takt
  + IU – celočíselné operace, zápis/čtení paměti, moduly(dekódování, provedení, zápis result)
  + FPU-obsahuje sčítačka s 2xpřestnost, simple násobička, dělička a pole registrů, moduly(čekací, dekódování, 2 prováděcí, zápis výsledku)
  + BPU – pro řízení skoků, sčítačku pro výpočet adres skoku

**Přerušení**

* =odezva procesoru na požadavek o programové obsloužení události
* 1)dokončí instrukci, 2)dočasně vypne ostatní IR, 3)převezme vektor(adresu podprogramu) a spustí ho, 4)vrátí se na původní místo a pokračuje
* *Obsluha priority přerušení a identifikace žadatelu o přerušení*
  + Programová identifikace–vyvolá se program, který přejde do požadované větve, nenáročné způsob, pomalé, klade nároky na program
  + Seriová obvodová identifikace – očekává identifikační znak s prioritou, procesor pošle potvrzení po sběrnici(zařízení přepošle/vrátí) a podle priority vyhoví
* *Řadič přerušení* – obvod, identifikuje zařízení žádající o help
  + DMA-způsob přístupu dat/perifeií do OP paměti(bez CPU), zrychlujePC, při hodně datech
* *Způsob řízení*
  1. řadič->signál->procesor HOLD->uvolní sběrnici + řadič přenese data->řízení do z5 procesoru
  2. po ukončení cyklu úplné odpojení procesoru pro část dat – zpomaluje
  3. synchonní přenosy se uskutečňují , když procesor pracuje+nepoužívá sběrnici - nezpomaluje
  + Registr dat – obsahuje slovo, které má být přesunuto z periferie do paměti(i naopak)
  + Registr adresy –adresu paměti, do které se mají přesouvat data z periferie
  + Čítač přesunů – kolik přesunů zbávý
* *Průběh přenosu dat pomocí DMA*
  + Procesor naprogramuje radič DMA->spustí periferii-> čeká na data->CPU pošle info o povolení přenosu-> DMA pošle adresu a data a čeká na přenos->DMA ++registr & --čítač přesunů ->if čítač!=0 check next data?
* *Kanál* – multiplex/selektor, přenosový procesor se stará o I/O mezi CPU a periferiemi
* *Programové čítače* – čítač=registr, uchovává adresy instrukcí, po načtení libovolné instrukce se ++ a ukazuje na další instrukci, která se vykonání aktuální načtě
* *Atmel T90* – 8bit CMOS mikrořadič, 1 hod cyklus/1 instrukce, Harvard, po 16bit
  + 1úrovňové překrývání(1.vykonává/2.načítá), 32 pracovních registrů připojené k ALU
  + Obsahuje RWM zásobník, analogový komparátor, rozhraní SPI, 8bit čítač/časovač, watchdog, může obsahovat UART, 10bit modul PWM, 16bit časový modul/EEPROM
* *Atmega 32* – 40pinů(2xnapájecí, 2xpro oscilátor, 1xRESET, 3x pro napájení, 32x pro I/O operace)
  + Piny umí fungovat v analog mod, 3 vnitří timery, aproximační ADC(analog-to-digital convertor)
  + USART, rozhraní SPI, 32 kiB programovatelné Flash, 1kiB EEPROm, 2kiB interní SRAM, Frekvence 1-16 MHz, two wire inteface
  + lze programovat pomocí SPI paralelně nebo pomocí JTAGu(slouží k testování plošných spojů a interních obvodů) – lze nastavit fuse bity
* *Microchip PIC* – monolitické 8bit, RISC, Harvard, pro kontrolní a řídící úlohy, 1 klíčový registr(W)=>realizace ALU operací, hodně datových registrů, HW zásobník

**Architektura grafických zobrazovačů, SVGA** – rozlišení 640x480 – 1280x1024, min 512kiB

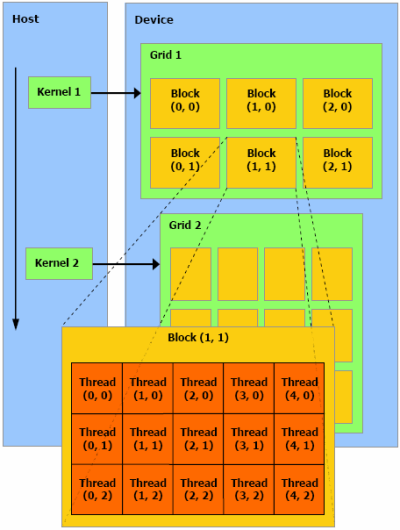
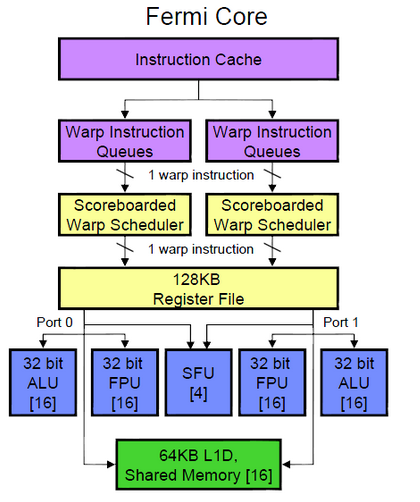
* *Obrazový kontroler* – generuje horizontální a vertikální synchronizační signály a signály pro obnovení dynamických oblastí, podporuje kurzor a podtržení
* *Sekvencer* – generuje časovací signály pro RAM a tvary znaků v textovém režimu
* *Adresový a datový multiplexer* – řídí přístup ke stránkám RAM, multiplexer=kombinuje analogové signály nebo digitální datové toky do 1
* *Grafický kontroler* – komunikace mezi mikroprocesoru-obrazovou paměti-atributovým kontrolerem
* *Atributový kontroler* – data z grafického konroleru -> atributová data v textovém režimu/sériová data v grafickém režimu, zařizuje blikání, podtržení a generuje kurzor
* *Číslicově analogový převodník* – D/A převod dat, RGB kanály

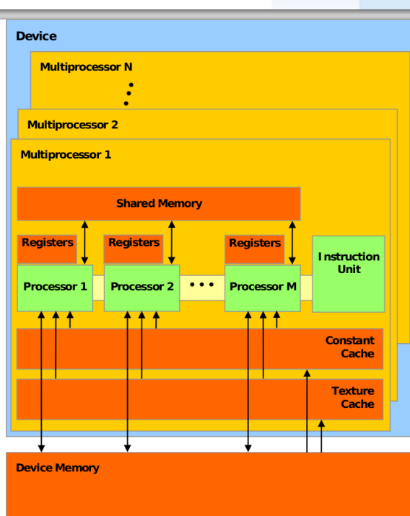
**Grafické procesory** – slouží k zobrazení dat dodaných procesorem, počítá body obrazu->ukládá do videopaměti, specializace na grafické výpočty(geometrie, rastr, textury)

* *Paměti* – DRAM(1 V/V brána-čtení/zápis), VRAM(video - 1brána čtení/vstupní, 1brána zápis/výstupní), WRAM(window – VRAM+podpora 3D transformace a zpracování videa)
* *Obnovovací freknvence* – počet snímku překreslených na monitoru/sekundu
* *Horizontální rozkladová frekvence* – počet řádku vykreslených/sekundu
* *DAC*- rychlý aby zvládal konvertovat data D->A, min.rychlost podle obnovovací frekvence/rozlišení/dodatečné kapacity pro řídící signály

**CUDA** – unifikovaná architektura pro výpočty na grafických kartách, není potřeba znát architektutu GPU

* Problémy se neřeší v jazyce GPU, only NVIDIA, rozhraní – C/C++,Fortran,Python; optimalizace na mat. náročné výpočty, hodně ALU, málo řídicích obvodů, několik SM-multiprocesorů, několik procesorů(vlastní registry), instrukční jednotka(rozděluje práci procesorům)
* *Typická operace –* Copy dat CPU->GPU, run vlákna na GPU, do výpočty na GPU, Copy dat GPU->CPU
* *Vlákna* – musí být na sobě nezávislé, nelze zaručit pořadí vykonávání, organizována do bloku->grid->dimenze, skupiny vláken(warpy) opakovaně spouštěny schedulerem a přepínají se mezi sebou(paralelismus), všechny multiCPU mají přístup k globální/konstantní/texturové paměti
* *FERMI - GF100* – 3 miliardy tranzistoru, 512 jader, 64bit adresní sběrnice, komunikace s CPU přes host interface(8GiB/s), 6x64bit řadič GDDR5, optimalizace pro 2x přestnost výpočtu, FMA instrukce(1 takt SP, 2 takty DP), ECC ochrana
* *GigaThreadEngine* – stará se o rozdělení práce mezi jednotlivé streaming multiprocesory(SM)
* *Streaming multiprocessor –* 32 zřetězených jader, warp scheduler=přiřazuje vláknům instukce (2/takt), 16 load/store, 4 jednotky SFU, programovatelná cache, pamět pro textury,konstanty
* *CUDA jádro* – 1x FPU, 1x ALU(integer unit)
* *Rozšíření C/C++* - kvantifikátory funkcí a proměnných(\_\_global\_\_,\_\_device\_\_,\_\_host\_\_), spouštění kernelů(fn<<<počet\_bloku,počet\_vlaken>>>(params)), datové typy(intx,charx: x číslo)





**TTL, bipolární/unipolární t., CMOS** – technologie pro vyrábění číslicových obvodů, nápájení 5V

* *TTL* – tranzistor s vícenásobným emitorem=>logické funkce
  + L-TTL(low-power->slow, měnší P-příkon), H-TTL(high-speed->rychlejší,větší P), S-TTL (schottkyho dioda),LS-TTL(nejčastější;low-power-schotttky),LV-TTL(low-voltage 3.3V),ASTTL(rychlé, malé P)
* *Bipolární technologie* – přenos náboje elektron i díra, rychlejší, levnější, větší spotřeba energie, menší stupeň integrace
* *Unipolární technologie* – přenosu náboje se účastní buď elektron OR díra, FET tranzistory
  + MOS-N – unipolární tranzistor s kanálem N(elektrony), 3x fast než MOS-P, sloučitelný s TTL,5V
  + *HMOS* – lepší, zmenšení o ½->8x zrychlení, úprava délky kanálu/šířka oxidové vrstvy/ hloubka difuze/překrývání elektrod
  + *CMOS* – spínací prvek(MOS-N), zátěž(MOS-P), extrem nízký P, napájecí 3-18V, kompatibilita s TTL, na výstupu generuje 0.005-4.95V(krajní hodnoty)
  + *BiCMOS* – Intel->Pentium, kombinace CMOS a bipolárního spínače

**PCI-Express**–skládá se z linek o rychlosti 250(new500)MiB/s,100%z5 kompatiblita s PCI, hot plug/how swap

Přidávat/vyměňovat zařízení za běhu PC

* *Link-*2 nízkonapěťové differen. páry, komunikační kanál–full duplex,linky obousměrné(vysílat/přijímat)
* *Rychlost* – 500MiB/s(oba směry dohromady), až x32 při 1směru->při 2směrech 2xvíc
* *Zařízení*-root komplex(propojení mezi zařízeními/hostitelským systémem/pamětí), switches(rozšíření a větvení sběrnice), Bridges(přemostění na jiný typ sběrnice), Endpoints(koncová zařízení)
* *Vrstvy*
  + Transaction layer – komprese a dekomprese paketů, paket=nese info o operaci
  + Data link layer – zajišťuje integritu dat – detekce, oprava, kontrola chyb
  + Physical layer – zajišťuje komunikace mezi linkem a linkovou vrstvou, připojení HW(peer-to-peer), full-duplex rozhraní, změna formátování dat/přenosová rychlost->only change physical

**PCI vs PIC-Express** –

* *Srovnání*
  + PCI – paralelní, hodně vodičů, half-duplex, 132MB/s, NO hot plug, NO řízení spotřeby, kontrola parity, DMA, IO, peer-to-peer  
    PCI-Express – sériová, málo vodiču, full-duplex, 250(500)MB/s, Hot Plug, řízení spotřeby, CRC, ošetření na různých úrovních, paketová komunikace
* *Sériový přeno*s – přenos po 1vodiči, slow, komunikace větší vzdálenosti, načtené znaky sériově přeneseny k přijímači, odstranění redundantních bitů, externí/interní kominukace
* *Asynchronní komunikace* – kdykoliv zahájení přenosu znaku, start bit(hrana=0), paritní bity, stop bit(), frame=všechny odeslané bity
* *Synchronní komunikace*–nepoužívá,komunikace synchronizována hodinovm signálem,no start/stop bit
* *USART* – i synchronně(sychronous/asynchronous)
* *UART* – pouze asynchroně(universal asyhcnronous receiver/transmitter)

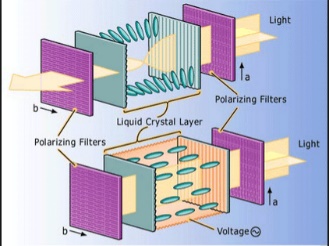
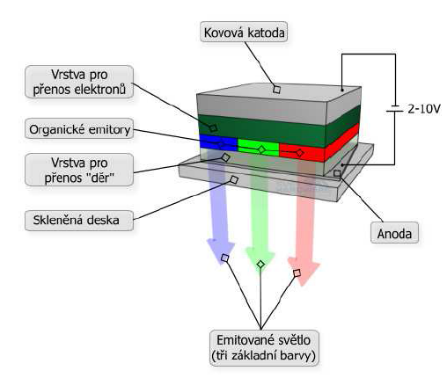
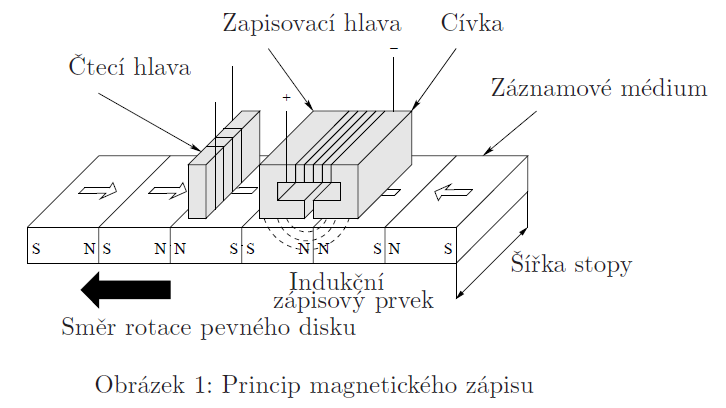
**SATA, PATA** – rozhraní pro komunikaci s pevnými disky,default-IDE,max. přenos rychlost 133MB/s

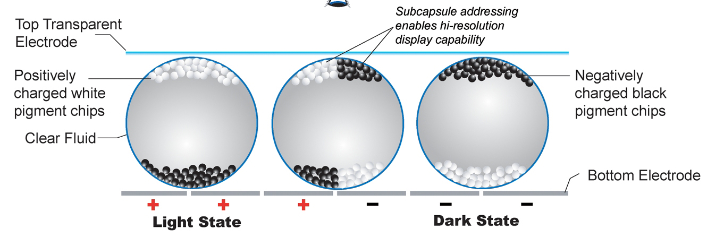
* *PATA*(parallel)–hodně pinů, vyšší-5V,problém s velikostí kabelů&frekvence,40vodičů-plochý kabel
* *SATA*(serial)-novější, nižší U, vysoká rychlost, menší kabely, žádný master/slave, podpora velkých disků, hot swap, NCQ, port multiplier
* *NCQ*-řadí příkazy do fronty, optimalizuje rychlost a šetří disk, pořadí čtení->co nejméně posunů hlavy
* Port multiplier – 1 řadič dokáže obsloužit více pevných disků
* I2C–2drátová2vodičová sběrnice se sériovým přenosem, slave/master obvody, připojit až 128 zařízení
* Adresa zařízení – skládá se ze 7bitů(horní 4 výrobce, dolní 3 jdou nastavit libovolně)
* Signály – SCL(synchronous clock), SDA(synchronous data)
* Náběžná hrana – vezmi si data – write
* Sestupná hrana – změň data - read
* Komunikace – klidový stav(SCL=SDA=1),START(SDA-dolu/SCL-hore),KONEC(SDA-hore/SCL-dolu), start->slave adresa(A-adresa/7bit;M-read(1)/write(0))->ACK 0(ok)/ACK 1(slave nereaguje)->NACK(1)

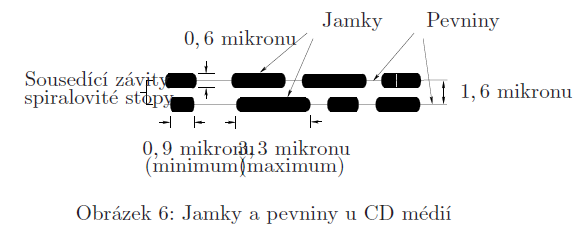
**Monitory**

* *CRT* – vzduchoprázdná skleněná buňka, přední část luminofor(luminiscenční látka)
  + Princip-elektronové dělo vystřeluje(zahřátím katody) proudy elektronů(každá barva/dělo), elektronové svazky->Wheneltový válec->mřížky(zaostří)->maska(propustní jen část->dopadnou na luminofor)->svazky vychylovány horizon./vertik. pomocí cívek(left-up corner->right-up, řádek++), obnovovací f-60Hz

Nekvalitní obraz

* + *Invar* – otvory v masce jsou kruhové(uspořádány do trojúhelníku), nevýhoda velká plocha,
  + *Trinitron* – od sony, maska=pevně natažené horizont.drátky, body více září(proniká více elektronů), obrazovka=širokoúhlá, náchylná interference(prolínání)eltmagnet.pole
  + *Vlastnosti* – úhlopříčka, rozlišení, horizontální/vertikální frekvence
  + *Výhody* – ostrost, věrné barvy, doba odezvy, pozorovací úhly
  + *Nevýhody* – velikost, vyzařování, spotřeba
* *LCD monitor*–jádrem TN struktura, ohraničena 2stran skleněnými deskami s polarizačními filtry
  + *Princip* – světlo->polarizační filtr->otočení o 90st->struktura tekutých krystalů->otočení o 90st->2.polarizační filtr->displej svítí(default), el.proud konrolujeme orientaci krystalů(I určuje intenzitu světla), nutnost podsvětlovat displej Ledkami
  + *Barevné LC*D – bod(3šásti=subpixely)->RGB jako filtry, které skládají výslednou barvu
  + *Pasivní displeje*–mřížka vodičů(body=průsečíky,aktivuje se když I-run), hodně bodů-> rosteU->rozostřený obraz a velká doba odezvy(3FPS),zařízení s malým displejem(hodinky)
  + *Aktivní displeje*-každý průsečík v matici obsahuje tranzistor OR diodu, která řídí činnost bodu(možné velmi přesně a rychle ovládat svítivost každého bodu), 0.2-0.3mm
  + *Výhody* – kvalita obrazu, živnost, spotřeba energie, odrazivost a oslnivost, bez emisí
  + *Nevýhody* – citlivost na teplotu, pevné rozlišení, vadné pixely, doba odezvy
* *Plazmové monitory*-plazma=skupenství iontů a elementárních částic(4skupenství)
  + Klid stav=směs vzácných plynů(argon,neon,xenon)->el.proud->volné elektrony + plyn =srážka->vznik +Ionty->plazma; nabité častice->opačné póly->náraz=uvolnění fotonů, dispej=matice buněk(řádky-adresovací elektrody, sloupce-zobrazovací elektrody), buňky uzavřeny v 2skleněných tabulkách(kondenzátor+3elektrody), pixely->subpixely RGB, ovládání intenzity(PWM), snímky->podsnímky->světlo
  + Výhody-kvalitní&kontrastní obraz, bez podsvícení, velké pozorovací úhly, min.hloubka&váha
  + Nevýhody-paměťový efekt(obraz se může vypálit-nePC), cena, energetická náročnost
* *OLED* - hlavní prvek organická dioda emitující světlo
  + v buňce je kovová katoda/vrstva pro přenos elektronů/organické emitory/vrstva pro přenos děr/průhledná anoda; po přivedení U na obě elektrody->hromazení elektronů u ANODY(vrstva pro elektrony) & hromazení děr u KATODY(vrstva pro přenos děr)-> v organické vrstvě Srážky(elektrony/díry)->jejich vzájemná eliminace->vyzáření fotonu
  + PMOLED - pasivní, 2 na sebe kolmé elektrody procházejí celou výšku i šířku displeje
  + AMOLED – active matrix OLED, každá buňka má tranzistor
  + Výhody – nepotřebuje podsvícení, vysoký kontrast, velmi tenké, nízká spotřeba, dobrý pozorovací úhel, min.zpožední, snadná výroba, možnost->pružné displeje
  + Nevýhody – žádné, body se rády vypalují
* *E-INK* – elektronický inkoust->tvořen mikrokapslemi(10-100 mikrometrů)
  + každá buňka obsahuje kapsli,která je naplněna roztokem, který obsahuje bílé(oxid titaničitý,kladný náboj) a černé částice(uhlík,záporný náboj),kapsle zespodu/svrchu ohraničené elektrodami,které podle potřeby přitahují na protikladně nabité elektrody
  + Energie se spotřebovává pouze při překreslení zobrazovaných bodů, molekuly vydrží velmi dlouhou dobu zůstat na místě(roztok-hydrokarbonový olej->častice na místě i bez napájení)
  + Výhody-vysoké rozlišení, dobrá kontrast, pozorovací úhly, čitelnost na slunci, bez podsvícení, možnost->pružný podklad, minimální spotřeba při překresleni(0 bez překreslení)
  + Nevýhody–pouze 16 odstínu šedi,špatné barevné rozlišení,zpoždění při překreslování
* *Princip multiplexu na zobrazovací zařízení* – nejsou ovládané naráz->po částech(řádky,znaky,pixely)
  + Data posílaná ne naráz-> za sebou(řádek,sloupec,znak), časový multiplex(zobrazuje naráz pouze část dat, strašně rychle blikání-30-50/s->lidské oko->souvislý obraz)
* *Znakově orientované displeje*-7segmentové displeje, vykreslují se po 1 znaku
  + Matice=několik pozic pro znaky, vykreslení->pozice(řádek,sloupec)+jaký znak (adresa), znaky jsou uložené v paměti displeje, m+n vodiču(m-segmenty, n-znaky)
* *Pixelově orientované displeje*-matice pixelů, ovládání jednotlivě OR po řádcích/sloupcích
  + LCD (s aktivní maticí)si pamatují poslední stav->zůstává svítit ikdyž není aktivně ovládán
  + Výhody – méně vodičů, jednodušší elektronika, snížená spotřeba

**Pevný disk** – záznamové médium=disk, páska s magnetickou vrstvou

* + Zápis/čtení pomocí hlavička, magnetický obvod=jádro+cívka na jádře, pokud ním prochází proud vzniká magnetický tok->ovlivňuje magnetickou vrstvu(změna směru I=>změna směru magnetického toku), First naležato->nyní svislé záznamy, záznamy jsou uloženy na magnetické vrstvě, data se zapisují tak, že přes magnetické jádro hlavy
  + Pro přečtení sektoru se musí hlava přesunout nad správnou stopu, a následně jej přečte. (prakticky se disk otáčí tak rychle, že není potřeba říct, že čeká)
* *Magnetické reverzace* – místo, kde se mění orientace magnetizace média, při čtení se hlavička pohybuje stále 1směrem, pokud hlavička narazí na magnetickou rezervaci dojde k vyvolání mag.toku a vzniku pamětového pulzu; data jsou kódovány
* Části–plotny disku,hlavy pro čtení/zápis+pohon,vzduchové filtry,pohon otáčení ploten disku,řídící deska
* *Stopa*-rozdělené plotny na kružnice, moderní disky-statisíce
* *Cylindr*-stopy nad sebou tvoří pomyslný válec
* *Sektor*-nejmenší adresovatelná jednotka(4096B), číslování od1, na začátku hlavička s číslem selektoru, na konci zakončení selektoru s kontrolním součtem, mezisektorové skoky(před&za)->nelze uložit data
* Čtení – 1)hlava se přesune nad stopu(nejdelší), 2)disk se natočí do požadované pozice
* *Optická média*
  + CD-ROM – polykarbonát, 1zapsány pomocí přechodů mezi pity(prohlubně) a landy(ostrůvky), 0 se nezapisují; zápis se provádí pomocí silně zostřeného paprsku, který způsobí lokální ohřátí a změnu vlastnosti materiálu, čtení je založené na odrážení paprsku od pitů a landů.
    - U CD/DVD je stopa oproti pevnému disku spirálovitá (výjimku tvoří DVD-RAM), data->do stop->vine se směrem k okraji disku, stopa(dělí na sektory)
    - Části-laserová/optická hlava, fotodetektor, servomechanismus, ovládací čip, vyrovnávací paměť, elektronika pro detekování signálu a řízení procesů
  + CD-R – WORM, 1vypálit data, silným laserovým paprskem(větší výkon než při čtení) dojde k zahřátí a změně vlastnosti materiálu -> vznik pitu

(24B data; 1B subkod; 8B korekce)

* + CD-RW-data lze zapisovat a opět vymazat pomocí změny krystalické struktury, 33B rámec
  + DVD–vysokokapacitní, zmenšení rozměrů pitů, odstranění zbytečného kódování, změna vlnové délky paprsku, 2064B rámec - 2048B data, 4B identifikátory, 2B korekce pro identifikátory ,6B ochrana autorských práv, 4Bkorekce dat
  + DVD-RAM-práce jako s pevným diskem, v pouzdře
  + DVD-RW, DVD+RW – formát pro opakovatelně přepisovatelné DVD disky, konkurence
  + Vlastnosti-CD-ROM(650-700MiB,červený laser),DVD-ROM(oboustraný záznam, 2 vrstvy, 4,7-? GiB)

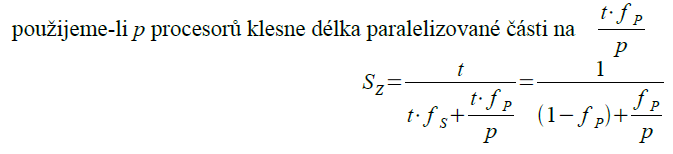
**Flynnova klasifikace, paralelní systémy**

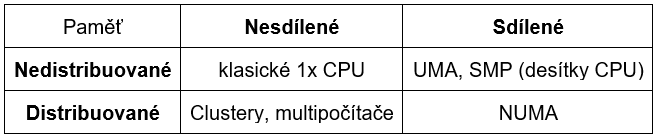
* Flynovou klasifikací lze třídit paralelní systémy podle počtu toků instrukcí a dat
* SMP - symetrický multiprocessing, všechny jádra procesoru jsou stejně výkonné, každý procesor vykonává jiné instrukce - MIMD
  + *Statické sítě*: cesta, kružnice, 2x kružnice, binární strom, krychle
  + *Dynamické sítě*: křížové, promíchání s výměnou, sběrnice
* SIMD – single instuction, multilple data => MMX, SSE, vektorové instrukce
* SISD - single instuction, single data =>klasický přístup
* MISD - multiple instuction, single data => nepoužívá se moc
* MIMD - multilple instuction, multilple data=> vícejádrový systém, nejčastější
* SIMT - single instuction, multilple THEAD => CUDA
* MSIMD – multiple single instuction, multilple data => několik SIMD pracuje nezávisle na sobě
* SPMD-same i., multilple data=> modifikace SIMD, všechny CPU dělají stejný program nezávisle na sobě

**Amdalův zákon**-slouží k vypočtení zlepšení výkonu při upgradu some části/zvětšením počtu procesorů

* SZ-součinitel zrychlení (poměr doby, která by byla potřebná, kdyby celý výpočet probíhal na 1

procesoru k době potřebné při částečné paralelizaci)

* fS podíl sériové délky, fP podíl paralelní délky (fs + fp = 1), t celková doba sériového výpočtu

**Paměť**

* *Sdílený prostor* – procesy mají přístup k 1 paměti
* *Distribuovaná paměť* – každý proces má vlastní adresní paměť
* *UMA*=uniform memory access, SMP=symetrický multiprocessing, všechny procesy stejně výkonné
* *NUMA*=non-uniform memory access
* *Statické sítě*: cesta, kružnice, 2x kružnice, binární strom, krychle
* *Dynamické sítě*:
  + křížové: nezávislost cest, lze vytvořit kolik je I/O, drahé!
  + promíchání s výměnou: jsou sestaveny z el. přepínačů, dělí se na 1 a více urovňové
  + sběrnice: nejjednodušší, ale přetížitelné
* *Systém se sdílenou pamětí*-přistup do vzdálené paměti se posílá zasíláním zpráv, nutné při hodně procesorech, 1)mřížka sběrnic-každý procesor připojen na 2->posílají si zprávy, 2)křížový přepínač-drahé, komunikace po ethernetu, 3)vícevrstvá přepínač-není potřeba posílat zprávy, každý procesor má přístup všude, 4)přímá síť – procesory propojeny přímo
* *VLIW*-PC s dlouhým instrukčním slovem
* Zálohované systémy-výpočet na několika procesorech zároveň, SISD(stejná data i instrukce)
* *Systémy řízené tokem událostí*-když okolnosti dovolí dochází k určité operaci, operace neprobíhají na základě instrukcí(redukční počítače)
* *Systémy bez centrálního řízení*-neexistuje v nich program ani řadič
* *Systolické systémy*-jednoúčelové, operace je daná vnitřní strukturou
* *Model Neuronové sítě*-umělá inteligence, vzdaluje se od Von Neumanna(BEZ řadič, procesor, pamět)

**Úrovně granularity**–určuje jak velké úseky zpracování informace probíhají součastně, 1nejmenší

* Stupně paralelismu
  + *Pracovní*-poskytování činností jako soubrou nezávislýchúloh
  + *Programový*-řeší se v kódu(př.rozdělení matice na 4 submatice-každou vypočte 1CPU)
  + *Instrukční*-stará se o něj procesor nebo překladač, je plně transparentní
  + *Arimetický a bitový*-paralelismus na nízké úrovni, doména návrhářů ALU a FPU

**HPC** –superpořítače k vědeckotechnickým výpočtům, zapoužití výpočetních clusterů

* Clustery=z multiporocesorových SIMD setav, používají heterogenní součásti, které jsou na úrovni běžných uživatelských strojů+je jich hodně, clustery jsou propojeny vysokorychlostní sítí(Etherner, Infiniband)+sdílejí úlohy pomocí technik SMP a NUMA
* *Migrace procesů*=přesunutí procesu z jednoho procesoru na jiný v rámci clusteru
  + Preemptivní-proces lze přesunou za jeho běhu, složitější, často se používá
  + Nepreemptivní-proces lze přesunou pouze než začne, jednodušší implementace
  + Zvyšuje početní výkon tím, že přesouvá procesy na nevyužité procesory(load blancing), lze přesunout dlouhodobě trvající proces ze selhávajícího uzlu na jiný uzel, lze restartovat uzly(jeho procesy se přesunou jinam)
* *Checkpoint-restart*-systém pro zastavení procesoru a jeho následné obnovení

