

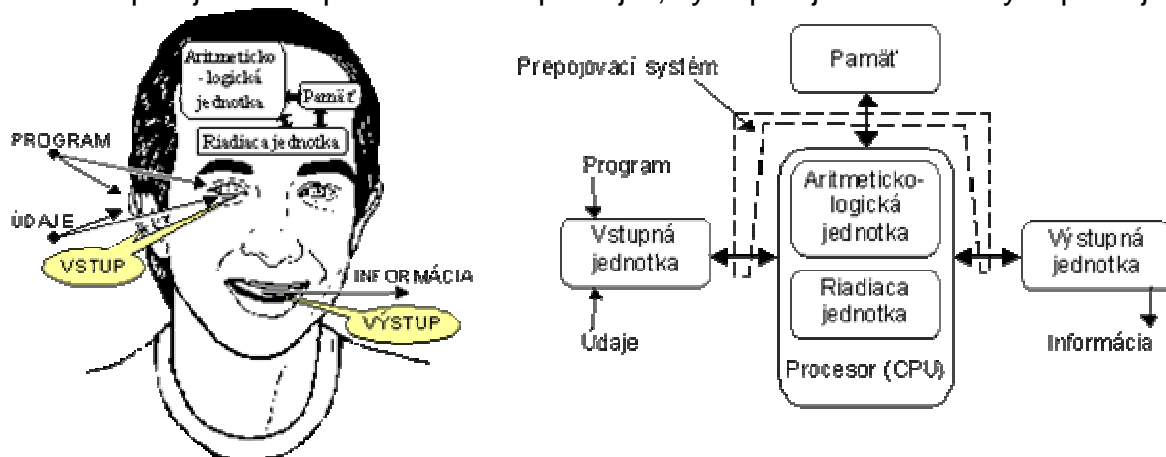
Základná koncepcia počítača

Základný koncept počítača, ktorý vykonáva postupnosť operácií na dosiahnutie konečného výsledku, je známy viac ako 150 rokov. Bol použitý v mechanických dekadických počítačoch strojoch, ktoré navrhol a čiastočne zostrojil už v úvode spomínaný Charles Babbage. Jeho analytický stroj (Analytical Engine) z r. 1834 obsahuje centrálnu procesorovú jednotku (mechanickú, s dekadickou aritmetikou), pamäť (mechanickú) a vstupnú a výstupnú jednotku (pre dierne karty), t.j. všetky základné časti moderných počítačov. Program a údaje pre tento stroj sú na diernych kartách. Samozrejme, vtedajšia mechanická technológia nedovolila úspešnú realizáciu funkčných zariadení, takže jeho myšlienky zostali viac ako 100 rokov nevyužitú. Až vytvorenie elektronických obvodov v 40. rokoch 20. storočia ich umožnilo zaviesť do praxe. V tomto čase **John von Neumann** predložil základný princíp počítača riadeného tokom inštrukcií, ktorý sa stal základom jednej triedy súčasných počítačov. Táto trieda počítačov býva tiež označovaná ako von Neumannovské počítače. Vyznačuje sa tým, že jednotlivé inštrukcie programu sa vykonávajú postupne za sebou, tak ako sú uložené v pamäti. V súčasnosti existujú aj výpočtové systémy, v ktorých sa inštrukcie nevykonávajú v poradí, v akom sú uložené v pamäti (napr. počítače riadené tokom údajov, neurónové počítače a iné). V tomto počítači sa vykoná práve tá inštrukcia, ktorá má pripravené údaje, pričom nezáleží od jej poradia

Počítače riadené tokom inštrukcií (von Neumannovské počítače)

Pre vysvetlenie princípu nám posluží nasledujúci obrázok. Základné črty von Neumannovského počítača sú:

- Pamäť použitá na uloženie inštrukcií aj údajov
- Riadiaca jednotka použitá na výber inštrukcií z pamäte.
- Aritmeticko-logická jednotka použitá na vykonávanie špecifikovaných operácií nad údajmi.
- Vstupná jednotka použitá na vstup údajov, výstupná jednotka na výstup údajov.



Obr.1 Analógia a bloková koncepcia von Neumannovského počítača

Pamäť je množina rovnakých buniek, z ktorých každá je samostatne identifikovateľná svojim poradovým číslom-**adresou**. Inštrukcie a údaje, uložené v pamäti, sú zakódované dvojkovým kódom.

Inštrukcia (príkaz pre riadiacu jednotku) určuje, aká operácia sa má vykonať a s ktorými údajmi. Inštrukcie sa vykonávajú postupne za sebou, tak ako sú uložené

v pamäti. Výnimkou sú skokové inštrukcie. Implicitne sa predpokladá pripravenosť údajov, ktoré vykonávaná inštrukcia požaduje. Dvojkovo zakódované inštrukcie sú označované ako *strojové inštrukcie*. Operácie, špecifikované v strojových inštrukciách, sú obyčajne iba jednoduché, napr. aritmetické a logické operácie, posuvy atď. , čo poskytuje najväčšiu flexibilitu. Zložitejšie operácie potom vytvára používateľ ako postupnosť inštrukcií. Z danej množiny strojových inštrukcií (inštrukčný súbor procesora) používateľ vyberá inštrukcie na vykonanie požadovaného výpočtu. Táto postupnosť vybraných inštrukcií sa nazýva *strojový program počítača*.

Riadiaca jednotka a aritmetická jednotka sú zvyčajne realizované ako jeden funkčný blok, ktorý sa nazýva *centrálne procesorová jednotka* (CPU) alebo skrátene *procesor*. Ak je procesor integrovaný na jedinom polovodičovom čipe, nazýva sa *mikroprocesor*. Procesor obsahuje niekoľko *registrov*, ktoré sú použité na uchovávanie špecifických operandov (použitých pri výpočte), adres a riadiacich informácií. Počet registrov a ich funkcia závisí od konkrétneho procesora, ale niektoré registre sú prítomné v každom procesore von Neumannovského počítača. Typickým registrom je programové počítadlo. Obsahuje adresu nasledujúcej inštrukcie, ktorá sa bude vykonávať

Popis činností jednotlivých súčastí počítača von Neumannovho typu.

1. Riadiaca jednotka nahrá program, ktorý sa má vykonávať prostredníctvom zbernice zo vstupného zariadenia do operačnej pamäte.
2. Počas vykonávania programu sa jednotlivé inštrukcie z operačnej pamäte postupne prenášajú do procesora, ktorý ich spracováva.
3. Spracované výsledky sa uložia späť do pamäte.
4. Počas vykonávania programu môže procesor komunikovať so vstupno-výstupnými zariadeniami a zobrazovať, tlačiť a prijímať nové údaje.

Zbernice

Funkčné jednotky výpočtového zariadenia možno prepojiť tak, že sa každá dvojica zariadení, medzi ktorými môže dôjsť k výmene správ, navzájom spojí komunikačným kanálom (prenášajú sa ním adresy, údaje i riadiace signály). „Kanálový“ počítač je však zväčša potrebné tvoriť „na objednávku“ zákazníkovi, je teda drahý. V prípade zbernicovej štruktúry nám postačuje jediný prenosový kanál, na ktorý sú napojené všetky zariadenia. Nazývame ho **zbernica**, anglicky **bus**. Je to lacnejšie riešenie a ľahšie sa do počítača pridávajú ďalšie zariadenia – stačí ich nakonfigurovať a pripojiť k príslušnej zbernici, ak už jej kapacita nebola vyčerpaná.

Zbernica teda slúži na prepojenie procesora s ostatnými súčastami počítača. Je rozdelená do troch častí:

- **Adresná** - časť zbernice, na ktorú sa nastaví adresa zariadenia, s ktorým chce procesor komunikovať
- **Riadiaca** - časť zbernice, cez ktorú do zariadenia posielajú príkazy, ktoré zariadenie vykonáva
- **Dátová** - časť zbernice, cez ktorú sa prenášajú údaje

Prenosová rýchlosť zbernice udáva množstvo údajov, ktoré je možné preniesť po zbernici za jednotku času. V jednom cykle zbernice sa vykoná prenos jediného údaju po zbernici. Napríklad ak je šírka dátovej časti zbernice 8 bitov, v 1 cykle sa preniesie 1 bajt.

Pamäť

Všetko, čo používame v počítači, musí byť niekde uložené. Údaje sa ukladajú do pamäte. Za minimum u pamäte považujeme jednu cifru dvojkového čísla – binary digit, skrátene bit, značka *b*. *Bit* môže obsahovať iba číslo 0 alebo 1. Dobré sa dá realizovať v pamäťovom médiu, ktoré sa môže nachádzať v dvoch stavoch – stav zapnutý/vypnutý, zmagnetizovaný jedným smerom alebo iným... Kapacita pamäte býva udávaná v bajtoch *B* (*byte*). $1B = 8b$. Jeden bajt je teda tvorený ôsmimi bitmi. V dvojkovej aritmetike to znamená, že doň vieme uložiť číslo 0 až 255. To stačí na zakódovanie celej anglickej abecedy a niekoľkých znakov vrátane riadiacich. Pamäť môže byť typu RWM – Read Write Memory, t.j. dá sa z nej čítať i do nej zapisovať. Občas sa používa aj pamäť typu ROM – Read Only Memory, pamäť iba na čítanie. **Najrýchlejšou pamäťou je procesorový register.** Register je množina pamäťových buniek. Je veľmi rýchly, ale jeho kapacita je len niekoľko bajtov. Napríklad 32 bitový mikroprocesor má kapacitu jeho registrov je iba 32 bitov – 4 bajty. Procesor vie pracovať s 32 bitovými operandmi inštrukcií. Väčšou kapacitou oplýva pamäť RAM. Je možné ju adresovať po bajtoch od 0 až po jej maximálnu kapacitu. Skratka RAM znamená, že ide o pamäť s náhodným prístupom (Random Access Memory) – po prečítaní bajtu z adresy X je možné v ďalšom cykle prečítať bajt z ľubovoľnej inej adresy Y. Sekundárne pamäťové médiá sú ostatné pamäte – pevné disky, diskety, CD-ROM, magnetické pásky, USB čítačky. Oproti operačnej pamäti RAM majú veľkú kapacitu, presnejšie lepší pomer medzi cenou a pamäťovou kapacitou. Až na výnimky pre ne platí, že nie sú typu RAM. To znamená, že napríklad disky poskytujú údaje po veľkých blokoch (512B). Keď chce počítač prečítať napríklad tretí bajt, musí do operačnej pamäte načítať celý blok o veľkosti 512B alebo dokonca viac blokov. Potom v tomto bloku vyľadá tretí bajt a ten použije.

Procesor (CPU)

Hlavnými súčasťami procesora (CPU) sú:

- Riadiaca jednotka (Control Unit)
- Aritmeticko logická jednotka (ALU)
- Pamäťové registre

Riadiaca jednotka

Riadiaca jednotka riadi tok údajov a vykonávanie operácií v inštrukčnom cykle. Riadiaca jednotka pozostáva z časti:

- **Register inštrukcii**, ktorý uchováva operačný znak inštrukcie počas jej vykonania.
- **Dekodér inštrukcii**, ktorý obsah dekoduje a generuje riadiace signály pre procesor.

Aritmeticko-logická jednotka (ALU)

aritmeticko-logická jednotka ako súčasť (mikro)procesora vykonáva aritmetické (+, -, *, /) a logické operácie (súčet, súčin, negácia). Požadované operandy pre ALU dodáva riadiaca jednotka – riadič inštrukcií. Všetky aritmetické operácie možno redukovať na posunutie, vytvorenie doplnku a sčítanie. Odčítanie je súčet doplnku kladného čísla, násobenie je opakované sčítavanie a delenie je opakované odčítavanie... Základnými operačnými prvkami ALU sú binárne sčítačky a negátory. Operácie sa môžu vykonávať sériovo a paralelne. Sériová organizácia vystačuje s

jednou sčítačkou. Paralelné metódy sú rýchlejšie, ale vyžadujú veľký počet zhodných súčastí. Riadiaca jednotka dodáva ALU riadiace signály potrebné na správne vykonanie aritmetických operácií. V riadiacej jednotke bývajú implementované algoritmy násobenia a delenia. (Napríklad pomocou mikrokódu.) Riadiaci obvod operácií dozerá na správny priebeh operácií vykonávaných radičom. Do **stavového registra** zapisujú obvody kontrolujúce, či výpočtové operácie prebehli správne, alebo či poruchy neznehodnotili výsledok (kontrolný bit, prekročenie rozsahu, prerušenie).

Registre

Register je extrémne rýchla pamäťová bunka nachádzajúca sa priamo v procesore. Jej obsah je neustále prístupný pre inštrukcie. Používa sa ako zdroj dát počas výpočtov v procesore. Akýkoľvek procesor dokáže vykonávať výpočty iba s obsahmi registrov. Môže mať v inštrukčnej sade aj inštrukcie pre výpočty s premennými nachádzajúcimi sa v operačnej pamäti, no aj tieto premenné si musí pred výpočtom skopírovať aspoň do nejakého „tajného“ registra. Registre slúžia na rýchle ukladanie medzivýsledkov, adries alebo často používaných hodnôt. Majú veľmi krátku dobu prístupu, avšak je ich málo. Ich vhodným použitím možno znížiť dobu výpočtu programov, preto dobré prekladače počítačových jazykov disponujú optimalizátormi kódu optimalizujúcimi okrem iného aj využitie registrov pre intenzívne využívané premenné.

Popis procesu spracovania inštrukcie procesorom.

Spôsoby spracovania môžeme rozdeliť na:

- prúdové
- dynamické

Cieľom **prúdového spracovania inštrukcií** (*pipelining*) je urýchliť činnosť procesora tým, že sa inštrukcie vykonávajú paralelne. Vychádzame z toho, že spracovanie inštrukcie možno rozložiť na (spravidla) päť jednoduchších úkonov, ktoré na seba nadväzujú:

1. prenos inštrukcie z pamäte do procesora (*Instruction Fetch - IF*),
2. dekódovanie (*Instruction Decode - ID*) - inštrukcia sa konvertuje do jednoduchých povelov (mikrooperácií), ktoré výkonné jednotky procesora dokážu vykonať,
3. výber operandu z pamäte (*Data Access - DA*),
4. samotné vykonanie operácie (*Execution - EX*),
5. zápis výsledku do pamäte (*Write Back - WB*).

Každý úkon sa vykonáva v samostatnom funkčnom bloku, prípadne sa rozloží na ešte jednoduchšie operácie a vykonáva sa v niekoľkých funkčných blokoch.

Funkčný blok (*functional unit*) je skupina logických obvodov, ktoré vykonávajú spoločnú prácu (napr. aritmeticko-logická jednotka). Pri klasickom spracovaní údajov by sa ďalšia inštrukcia začala spracovávať až po ukončení spracovania predchádzajúcej inštrukcie. Pri takomto spôsobe práce by funkčné bloky väčšinu času zaháľali a čakali na príchod ďalších údajov. Prúdové spracovanie údajov prebieha tak, že funkčné bloky sú zoradené logicky za sebou a tvoria **kanál** (*pipe*). Akonáhle spracovanie inštrukcie postúpi z prvého stupňa (funkčného bloku) do druhého, môže prísť ďalšia inštrukcia a vstúpiť do prvej fázy svojho spracovania (obr. 1). Tým sa dosiahne, že až na niekoľko výnimiek sa v každom hodinovom cykle ukončí jedna inštrukcia.

Dynamické spracovanie inštrukcií je súbor metód, ktorých cieľom je zvýšiť efektívnosť práce procesora tým, že sa predpovedá a mení poradie vykonávania inštrukcií.

Predpovedanie skokov zohráva svoju úlohu v prípade nesequenčného vykonávania programu v dôsledku príkazu skoku. Pretože príkaz skoku prenáša riadenie do iného miesta programu, môže sa stať, že príslušné inštrukcie nie sú v okamihu potreby preložené a pripravené na vykonanie. Úlohou predpovedania skokov je na základe správania sa programu v minulosti predvídať, na akej inštrukcii bude vykonávanie programu pokračovať, aby sa ďalšie inštrukcie mohli vybrať z pamäti a dekodovať skôr, ako sa vykoná inštrukcia skoku. Táto snaha je opodstatnená, lebo skoky predstavujú 15 až 25 % všetkých vykonávaných inštrukcií.

Význam vybraných parametrov procesora

Procesor načítava program a jeho údaje do pamäte, vykonáva inštrukcie programu a komunikuje s ostatnými zariadeniami prostredníctvom zbernice. Rýchlosť a výkon procesora ovplyvňuje niekoľko faktorov:

- taktovacia frekvencia - počet krokov vykonaných za jednu sekundu,
- veľkosť registrov resp. dĺžka slova - počet bitov spracovávaných pri jednej operácii,
- typ inštrukčnej sady - počet rôznych inštrukcií, ktoré je procesor schopný vykonať,
- efektivita mikrokódu - počet krokov potrebných na vykonanie jednej inštrukcie (napr. vynásobenie dvoch čísel),
- počet jadier - počet samostatných inštrukčných dekóderov,
- počet inštrukčných kanálov - maximálny počet inštrukcií, ktoré sa dajú vykonať v jednom kroku,
- veľkosť vyrovnávacích pamätí - počet inštrukcií, ktoré má procesor k dispozícii na spracovanie,
- veľkosť adresovateľnej pamäte - veľkosť pamäte, ktorú je schopný procesor používať.

Taktovacia frekvencia

Taktovacia frekvencia určuje počet operácií, ktoré je procesor schopný vykonať za jednu sekundu. Udáva sa v násobkoch jednotky Hertz napríklad 3GHz.

Dĺžka slova

Pod šírkou spracovávaného slova myslíme maximálny počet bitov, ktorý je mikroprocesor schopný naraz spracovať. Procesor má v sebe zabudované tzv. registre, do ktorých sa ukladajú údaje s ktorými má procesor vykonať nejakú operáciu. Registre napríklad vstupujú ako operandy do matematických inštrukcií (sčítanie, odčítanie...). Veľkosť registrov je zhodná s veľkosťou slova. V súčasnosti sa najviac používajú 64 bitové procesory.

Inštrukčná sada

Inštrukčnou sadou rozumieme súbor inštrukcií, ktoré môže procesor vykonávať. Na základe toho, z akého typu inštrukcií pozostáva inštrukčná sada, môžeme rozdeliť procesory na:

- **CISC**(*Complex Instruction Set Computer - Počítač s úplnou inštrukčnou sadou*) predstavuje zložité inštrukcie pozostávajúce z mikroinštrukcií, jedinou inštrukciou je možné vykonať zložitú operáciu, čo sa môže zdať výhodné, no

takýto procesor vyžaduje zložitý dekóder inštrukcií a spracovanie inštrukcie trvá viacero cyklov

- **RISC**(*Reduced Instruction Set Computer - Počítač s obmedzenou inštrukčnou sadou*) inštrukčná sada pozostáva z malého počtu elementárnych inštrukcií, ktorými je možné vytvoriť zložitejšie operácie, jednoduché inštrukcie rovnakej dĺžky znamenajú, že procesor má jednoduchý dekóder inštrukcií, taktiež RISC procesory obsahujú veľký počet interných registrov s univerzálnym použitím, čo urýchlí výkon programu, pretože väčšina operácií prebieha medzi registrami a ALU

Počet jadier a inštrukčných kanálov procesora

Moderné procesory obsahujú viacero jadier, z ktorých každé môže mať viacero inštrukčných kanálov. To umožňuje použiť paralelné výpočty.

Veľkosť vyrovnávacej pamäte

Vyrovňavacia pamäť (hovorovo zásobník, angl. buffer) je spravidla vyhradená časť pamäte určená na dočasné ukladanie dát pred spracovaním (na vyrovnanie rýchlostí odosielajúcej jednotky a prijímacej jednotky alebo na prevenciu rôznych výpadkov pri prenose dát). Slúži na vyrovnanie oneskorenia zbernice oproti rýchlosti procesora. Keby procesor nemal vyrovnávaciu pamäť, musel by čakať pokým sa k nemu údaje dostanú z pamäte cez niekoľko násobne pomalšiu zbernicu. V prípade vyrovnávacej pamäte jedna časť procesora načíta naraz celý blok údajov priamo do procesora a druhá časť môže zatiaľ vykonávať operácie.

Adresovanie pamäte

Maximálna veľkosť pamäte je obmedzená adresným priestorom procesora. V súčasnosti je to 48bitov, čo umožňuje adresovať 2^{48} bajtov (t.j. 256 TB). Veľkosť adresovateľnej pamäte teda udáva veľkosť pamäte, ktorú je schopný procesor používať. V programe býva zadaná ako tzv. logická (virtuálna) adresa. Táto virtuálna adresa sa potom určitým mechanizmom (závislým na type procesora) prevádza na tzv. fyzickú adresu, ktorá je už adresou ukazujúcou na konkrétnu pamäťovú bunku v operačnej pamäti.

Zdroje:

Schmotzer Milan: Architektúra počítača

<http://maturitazinf.mrazovci.eu/koncepcia-pocitaca-von-neumannovho-typu>

[Sobota Branislav: Počítače a ich sieťové služby](#)