

Procesor

Výkonná jednotka počítače se nazývá procesor (CPU). Procesor je ústřední součást, která práci počítače řídí.

Hlavní částí procesoru je řídicí jednotka. Procesor obsahuje i další výkonné jednotky, zejména aritmeticko-logickou jednotku.

Řídicí jednotka procesoru neboli *řadič* vybírá z paměti instrukce a postupně je provádí.

Instrukce jsou elementární příkazy pro procesor. Každá instrukce říká procesoru, že má provést jednu operaci se svými *operandy*: čísla, znaky nebo logickými hodnotami. Operandy jsou typicky uloženy v jednom slově nebo dokonce jen v jednom bytu v operační paměti nebo v registru.

Instrukce také může měnit pořadí provádění následujících instrukcí tím, že procesoru nastaví adresu instrukce, která bude další na řadě.

Registr procesoru je elementární paměťové místo uvnitř procesoru, které má typicky velikost jednoho slova a ukládají se do něho adresy nebo data, s nimiž procesor přímo pracuje. Procesor obsahuje několik registrů (jednotky až desítky). Každý z nich má své označení a toto označení se používá v instrukcích.

Zpracování jedné instrukce

Procesor pracuje v *instrukčních cyklech*. V každém cyklu

- vybere z paměti instrukci,
- instrukci provede, většinou s pomocí aritmeticko-logické jednotky,
- výsledek instrukce zapíše do interního registru nebo do operační paměti,
- zjistí a nastaví adresu další instrukce.

Zpracování jedné instrukce

Každá *instrukce* procesoru říká,

- co se má provést
tj. která operace – aritmetická, logická, řetězcová, řídicí;
- kde se nacházejí operandy,
například instrukce sčítání má dva operandy (sčítance) a část instrukce říká, že první sčítanec je např. v interním registru a druhý na určené adrese v paměti; jiné instrukce mohou mít zase jiný počet operandů neboli jinou *aritu*;
- kam se má zapsat výsledek,
zda do paměti, nebo do interního registru apod.;
- jaká bude adresa následující instrukce
obvykle je to adresa následující po adrese právě provedené instrukce, ale některé instrukce mohou řízení programu změnit – *podmíněné* a *nepodmíněné skoky*.

Příklad: 16bitová instrukce, která do registru R1 přičte číslo z paměti, jehož adresu má v registru R3.

kód instrukce	první operand	druhý operand i výsledek
0110	010 011	000 001
add	@R3+	R1

- Kód instrukce 0110 říká, že se bude přičítat první operand ke druhému.
- Každý z obou operandů je určen šesti bity, rozdělenými do dvou trojic.
- Druhá trojice bitů v operandové části určuje jeden z osmi registrů, R0 až R7. U prvního operandu je to 011, tedy registr R3, u druhého operandu je to 001, tedy registr R1.
- První trojice bitů v operandové části určuje způsob adresování, tj. jak získáme vlastní operand. U prvního operandu je to 010, což znamená, že v registru R3 je *adresa* operandu: operand (sčítanec) je v paměti počítače. Dále způsob adresování říká, že se adresa v registru R3 má po použití zvýšit. U druhého operandu je způsob adresování 000. Znamená, že druhý operand (číslo, k němuž se přičítá) je v registru R1 přímo, tedy bez dalších odkazů do paměti.

Vlastní provádění jedné instrukce se skládá z kroků časovaných *hodinovými takty* procesoru. V každém taktu se propojí vhodné spoje mezi registry a paměťovým systémem a bity v nich se buď zkopírují beze změny, anebo se změní některou logickou operací (*not, and, nand, xor,...*). Propojování a změny bitů na nejnižší úrovni se dějí pomocí *logických hradel*. Logická hradla jsou jednoduché obvody, které provádějí logické operace s bity.

Instrukce z uvedeného příkladu `add @R3+ R1` by například měla kroky:

- zkopírování obsahu registru R3 na adresní vstup paměťového subsystému
- povel paměťovému subsystému pro výběr slova
- zkopírování výstupu paměťového subsystému do registru sčítačky v ALU
- povel ALU pro sečtení s obsahem registru R1
- zkopírování výstupu sčítačky do registru R1
- povel ALU pro zvětšení adresy v registru R3
- povel ALU pro zvětšení adresy v registru PC adres instrukcí
- zkopírování registru PC na adresní vstup paměťového subsystému
- povel paměťovému subsystému pro výběr další instrukce
- načtení další instrukce z výstupu paměťového subsystému

Generace počítačů

V dnešních počítačích je logické hradlo realizováno pomocí transistorů na čipu. Jeden transistor je oblast na křemíkovém plátku o velikosti řádově desítek až stovek nm². V minulosti však mohlo jedno hradlo zabírat i stovky cm³. Podle technologie, na níž jsou logická hradla realizována, se někdy rozlišují tzv. *generace počítačů*.

technická realizace hradel	příklad	rok	generace
mechanické prvky: páčky, kolečka	Babbageův Analytical Engine	1840	1. generace
elektromagnetická relé	Zuseho Z3	1940	
vakuové elektronky	Eniac	1945	2. generace
samostatné (diskrétní) transistory		1955	3. generace
transistory integrované na čipech		1965	
čipy s vysokou hustotou integrace		od 1980	4. generace

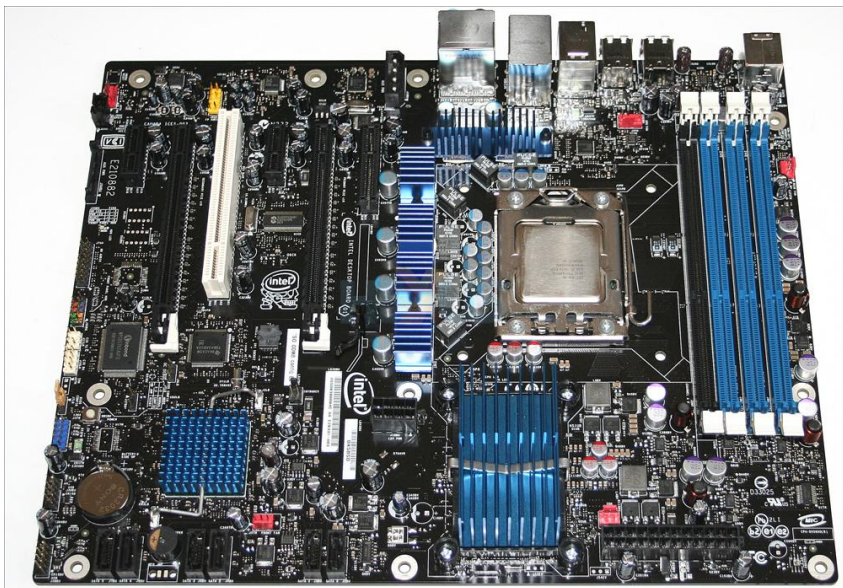
Poznámka: Rozdělení do generací není přesně definované a různé zdroje se drobně liší. Například, pokud se započítá i vývoj software, uvádí se někdy až 6 generací.

Reálné počítače obvykle mívají více spolupracujících procesorů. Navíc každý z nich má kromě jednoho hlavního řadiče více tzv. *jader*, z nichž každé má podřízený vlastní řadič a vlastní aritmeticko-logickou jednotku.

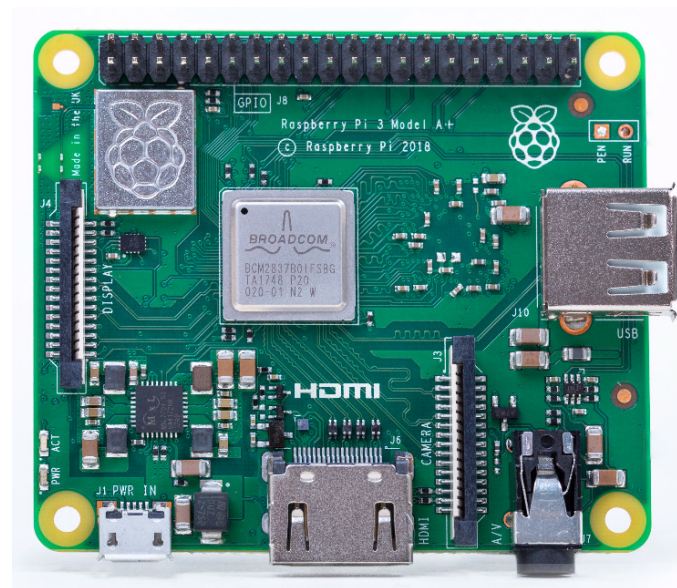
Cvičení: V shellu napište příkaz `less /proc/cpuinfo` a diskutujte o vypsaných položkách. Ve Windows podobně: `wmic cpu list` nebo klávesovou zkratkou -.

Procesory výkonnějších počítačů zpravidla obsazují jeden čip, který má aktivní chlazení.

Procesory malých počítačů mohou sdílet jeden čip spolu s pamětí, sběrnicí a s obvodem pro vstup a výstup. Čip má pasivní chladič.



Procesor i7 (v pravé části desky)



Procesor Broadcom na Raspberry Pi