Architektura počítačů

Milan Kolář, KES

Literatura

[1] Hlavička, J.: Architektura počítačů. ČVUT FEL Praha, 2001.
[2] Dvořák, V. – Drábek, V.: Architektura procesorů. VUT Brno, 1999
[3] Ličev, L.: Architektura počítačů I. VŠB TUO, Ostrava, 1999
[4] Skalický P.: Přístrojové aplikace mikropočítačů. ČVUT FEL Praha, 2004
[5] Pluháček A.: Projektování logiky počítačů. ČVUT FEL Praha, 2000
[6] Pinker, J.: Mikroprocesory a mikropočítače. BEN, Praha 2004
[7] Smékal, Z. – Sysel, P.: Signálové procesory. Sdělovací technika, Praha 2006

Definice architektury

Architektura – metoda vytváření počítačových systémů z menších celků

Architektura je globální pohled na všechny podstatné vlastnosti počítačů, zahrnuje:

- **strukturu** – popis propojení jednotlivých funkčních bloků
- **organizaci** – popis dynamických interakcí FB a řízení styku mezi nimi
- **realizaci** – popis návrhu a vnitřní struktury jednotlivých FB
- **funkci** – popis chování počítače jako celku

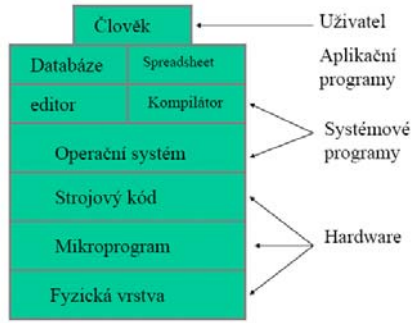
Znalost architektury má být prostředkem pro vytváření nových systémů, má být podkladem pro hodnocení kvality výsledku

Počítač - definice

Stroj na číslicové zpracování informací (dat).

Zařízení, které provádí výpočty nebo řídí operace, které jdou popsat čísly nebo logickými výrazy (Oxfordský slovník)

Virtuální (rozšířený) počítač – HW + OS

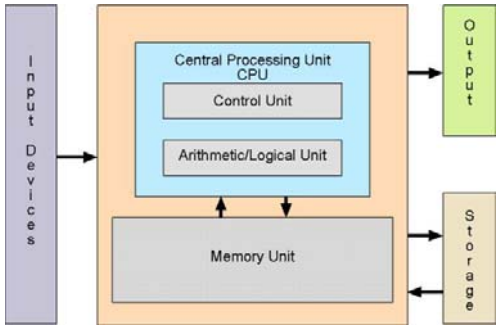


Počítač (elektronický číslicový)

CPU (Central Processing Unit) - procesor

ALU (Arithmetical and Logical Unit) – aritmeticko-logická jednotka

CU (Control / Central Unit) – řadič, řídicí jednotka



Počítač (elektronický číslicový)

I/O (Input / Output) Devices – vstupně / výstupní zařízení

Memory – paměť (operační)

Storage – paměť (archivní) – disk, karta (flash), páska

vše propojují **sběrnice** :

- datové
- adresové
- řídicí

Šířka toku dat (šířka sběrnice) – počet bitů, které se po datové sběrnici přenášejí současně (může být rozdílná uvnitř a vně procesoru)

Kategorie počítačů

Členění do 4 kategorií:

mikropočítače – určeny pro osobní použití (spec. jednočipové mikropočítače a řadiče)

minipočítače – sdíleny více uživateli (např. pomocí terminálů)

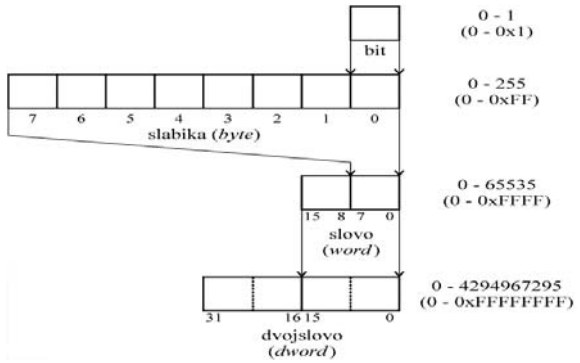
střediskové počítače (mainframe) – velký výkon, který slouží především k vědeckým výpočtům, velký počet V/V zařízení pro hromadný sběr dat

superpočítače – velmi velký výkon (armáda, meteorologie, seismologie, atomová fyzika)

Základní pojmy

Bit (binary digit) – základní jednotka informace (0 nebo 1)

Slovo (word) – skupina slabik, které se v počítači zpracovávají jako celek



Instrukce

Specifikace jednoduché činnosti, kterou má provést technický prostředek (nejčastěji procesor)

Binární tvar instrukce se skládá většinou na začátku z instrukčního kódu (určuje typ instrukce) a z parametrů, které mohou být:

- do instrukce zakódované konstanty
- označení registrů, odkud vzít hodnotu, příp. kam zapsat
- adresa paměti, odkud načíst hodnotu, příp. kam zapsat
- adresa paměti, kam má skočit další provádění programu

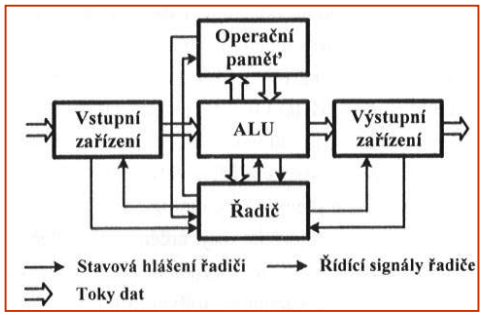
Instrukce lze zapisovat v různých tvarech:

- strojový zápis (11010000 10001010)
- častěji zápis v jazyku symbolických adres (mov ah, 56h)

Soubor instrukcí tvoří program

Von Neumannova architektura

představena v roce 1946



Von Neumannova architektura

Charakteristické vlastnosti lze shrnout do následujících bodů:

1) struktura počítače je nezávislá na typu řešené úlohy, počítač se programuje obsahem paměti
2) instrukce a operandy jsou v téže paměti
3) paměť je rozdělena do buněk stejné velikosti, jejich pořadová čísla se používají jako adresy
4) program je tvořen posloupností elementárních příkazů (instrukcí), které se provádějí jednotlivě v pořadí, v němž jsou zapsány do paměti
5) změna pořadí provádění instrukcí se vyvolá instrukcí podmíněného nebo nepodmíněného skoku

Von Neumannova architektura

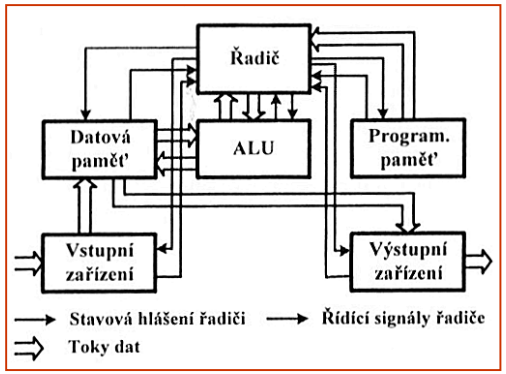
6) pro reprezentaci instrukcí i čísel se používají dvojkové signály a dvojková číselná soustava
7) programem řízené zpracování dat probíhá v počítači samočinně (tok dat řídí řadič)
8) zpracování dat probíhá v tzv. diskrétním režimu (během výpočtu nelze s počítačem komunikovat)
9) vstupy (resp. výstupy) jsou koncipovány jako datové zdroje (resp. výsledky) a jsou tedy přímo napojeny na ALU

Nevýhody:

možnost mylně interpretovat data jako program

Harvardská architektura

vznikla v roce 1943 (koncepce IBM Harvard MARK1)



Harvardská architektura

Základní principy (rozdíly vůči von Neumannově archit.):

1) paměť programu je oddělena od paměti dat
   - možnost ve stejném okamžiku načítat instrukci a přistupovat k datové paměti
   - datová a programová paměť mohou mít odlišnou organizaci
2) oddělené sběrnice
3) řízení procesoru je odděleno od řízení vstupních a výstupních jednotek (nejsou napojeny přímo na ALU)

- možnost rychlejšího zpracování většího objemu dat

CISC vs. RISC

CISC (*Complex Instruction Set Computer*)

- instrukční soubor se složitými operacemi s variabilitou různých adresovacích módů
- zpracování instrukcí ve více strojových cyklech
- řídící obvody zabírají na čipu přibližně 60% místa

RISC (*Reduced Instruction Set Computer*)

- instrukční sada obsahuje malý počet jednoduchých instrukcí (většinou pevné délky)
- instrukce jednocyklové (výjimkou může být komunikace s pamětí)
- řídící obvody zabírají pouze 6-10% místa (místo se využívá pro větší soubory registrů)

Historie počítačů

30. léta 20. stol. – spíše mechanické kalkulátory

1938 – **Z1** (Konrád Zuse) elektromechanický
kolíčková paměť na 16 čísel, nespolehlivý

1940 – první plně elektronický počítač (releový)
ABC (Atanasoff - Berry Computer)
paměť – 60 slov (50 bitů) v podobě kondenzátorů
taktovací kmitočet 60 Hz, velká chybovost (0,001%)

1941 – **Z3** (programovatelný kalkulátor)
pracuje s čísly s plovoucí desetinnou čárkou
(14-bitová mantisa, 7-bitový exponent, znaménkový bit)
paměť pro 64 čísel, 2600 relé

Historie počítačů

1943 – **IBM ASCC MARK I** (Harvardská univerzita)
16 m dlouhý, hmotnost 5 tun
800 km drátových spojů
¾ mil. součástek

1946 – **ENIAC** (Pennsylvanská univerzita)
hmotnost 30 tun, zabíral 15 m^2, příkon 174kW, 17460 elektronek, 1500 relé, 70000 odporů a 10000 kondenzátorů, pracovní frekvence 100 kHz.
Stroj nepracoval binárně, nýbrž dekadicky.

Generace počítačů

každá generace je charakteristická svou konfigurací, rychlostí počítače a základním stavebním prvkem

Generace	0	1	2	3	4
Rok	1940	1951	1957	1964	1981
Prvky	relé	elektronky	tranzistory	SSI	LSI
Hlavní paměť		buben	ferity	ferity	LSI
Kapacita paměti		1 KB	10 KB	1 MB	10 MB
MIPS	0,001	0,01	0,1	1	10
Příklad	Mark I	Univac 1	IBM 7090	IBM 360	Intel 4004

Čtvrtá generace

její vývoj prožíváme dodnes

základem je centrální procesorová jednotka (CPU) označovaná jako mikroprocesor (vesměs z křemíku)

IO LSI a VLSI (až 10^{18} tranzistorů na čipu)

malé rozměry (technologie 90 nm)

velká rychlost – využití paralelismu a zavádění programovacích prostředků, které paralelismus podporují

Začíná se mluvit o 5. generaci – orientace na využití umělé inteligence, přímý styk s uživatelem na úrovni přirozeného jazyka, textu a obrazů

Hodnocení výkonnosti počítačů

Základní požadavek kladený na počítač je schopnost provádět zpracování informací. Tuto schopnost označujeme jako **výkonnost počítače**.

Výkonnost je obtížné hodnotit jediným číslem – objektivnější je použít tzv. **vektor výkonnosti**, jehož struktura se vyvíjí.

Základem je **počet operací za sekundu**, buď v pevné nebo pohyblivé řádové čárce.

Dále **propustnost systému**, která udává počet úloh, které je systém schopen zpracovat za jednotku času

Dalšími složkami mohou být doba odezvy, stupeň využití, aj.

Hodnocení výkonnosti by mělo být podkladem pro optimalizaci

Výkonnost

Výkonnost $P_T(T)$ – inverzní hodnota doby T provedení 1 úkonu (programu)

$$P_T(T) = \frac{1}{T}$$

Výkonnost $P_R(n,T)$ – n úkonů za čas T

$$P_R(n,T) = \frac{n}{T}$$

Doba jedné instrukce

Výkonnost: $P = 1/T_p$ [µs, MIPS]

T_p … čas potřebný na provedení jedné průměrné strojové instrukce;

Pro zjištění T_p je třeba sestavit tabulku četnosti výskytu jednotlivých instrukcí při „běžném provozu“ počítače. Každá instrukce má přiřazenu svoji váhu a_i, která vyjadřuje pravděpodobnost výskytu instrukce v programu

$$T_p = \frac{\sum_{i=1}^{n} a_i t_i}{\sum_{i=1}^{n} a_i} \quad [s]$$

t_i … doba provádění i-té instrukce

a_i … váha i-té instrukce

n … počet instrukcí zařazených do mixu

Počet operací za jednotku času

Počet operací **s pevnou řádovou čárkou**:
KIPS, **MIPS**, GIPS (kilo, mega, giga instructions per second)
v amer. liter.: BIPS (billion instructions per second) = GIPS

Pro výpočty **s pohyblivou řádovou čárkou (FP)**:
FLOPS (floating-point operations per second), příp. flops
KFLOPS (10^3), **MFLOPS** (10^6), GFLOPS (10^9), TFLOPS (10^{12})

Jelikož se operace v pohyblivé a pevné řádové čárce provádějí v různých ALU, jsou na sobě nezávislé (u téhož systému může být počet operací v pevné čárce větší nebo menší než v pohyblivé)

Amdahlův zákon

Významný zákon informatiky

Popisuje výpočet výkonového zisku (zrychlení S) dosaženého vylepšením nějaké části počítače

Zrychlení S je číslo, které udává kolikrát je rychlejší běh úlohy na počítači s vylepšením oproti běhu stejné úlohy na původním počítači

$$S = \frac{\text{výkonnost při využití vylepšení}}{\text{výkonnost bez využití vylepšení}} = \frac{P_{NEW}}{P_{OLD}}$$

$$S = \frac{\text{doba výpočtu bez využití vylepšení}}{\text{doba výpočtu při využití vylepšení}} = \frac{T_{OLD}}{T_{NEW}}$$

Poměry F_E a S_E

Definujme si poměry:

F_E … udává, jakou část výpočtu lze vylepšit

$$F_E = \frac{\text{původní doba výpočtu zlepšené části úlohy}}{\text{původní celková doba výpočtu}} \leq 1$$

S_E … udává, kolikrát se zrychlil výpočet zlepšené části úlohy

$$S_E = \frac{\text{původní doba výpočtu zlepšené části úlohy}}{\text{doba výpočtu zlepšené části úlohy}} > 1$$

Doba výpočtu

Doba výpočtu na vylepšeném počítači se bude skládat z:

$(1 - F_E)\, T_{OLD}$ = doba výpočtu té části úlohy, kterou nelze vylepšit

$\frac{F_E}{S_E} T_{OLD}$ = doba výpočtu vylepšené části úlohy

Tedy **doba výpočtu** T_{NEW} na vylepšeném počítači je:

$$T_{NEW} = T_{OLD}\left((1 - F_E) + \frac{F_E}{S_E}\right)$$

Celkové zrychlení (Amdahlův z.)

Celkové zrychlení $S_{OVERALL}$ odpovídající danému vylepšení:

$$S_{OVERALL} = \frac{T_{OLD}}{T_{NEW}} = \frac{1}{(1 - F_E) + \frac{F_E}{S_E}}$$

Celkové zrychlení multiprocesorového systému, který má p procesorů, a jehož část programu f může být provedena pouze jediným procesorem, je dáno:

$$S = \frac{1}{f + \frac{(1 - f)}{p}}$$

Příklady

1) Předpokládejme vylepšení procesoru pro web. Nový CPU je 10-krát rychlejší pro webové aplikace než nynější. Dále víme, že nyní je CPU zaměstnán ze 40% výpočty a 60% času čeká na I/O operace. Jaké bude celkové zrychlení po plánovaném vylepšení?

2) Předpokládejme, že při FP výpočtech v počítačové grafice operace odmocniny FPSQRT odpovídá 20% a všechny FP instrukce odpovídají 50% doby výpočtu úlohy. Úkolem je rozhodnout, zda je výhodnější 10x zrychlit provádění operace FPSQRT nebo 1,6x zrychlit provádění všech instrukcí.

Instrukční mixy

Nejstarší způsob hodnocení propustnosti číslicových systémů

Instrukční mixy jsou seznamy (tabulky) nejfrekventovanějších instrukcí ohodnocených pravděpodobnostmi jejich výskytu v rámci daného typu zátěže

Nevýhody:

- instrukční mixy jsou zpravidla závislé na konkrétním procesoru a architektuře – obtížná přenositelnost
- frekvence použití jednotlivých instrukcí závisí subjektivně na programátorech, na druhu zpracovávaných úloh i na souborech vstupních zpracovávaných dat
- dobu instrukcí ovlivňuje i operační systém

Nejznámější mixy

Instrukční mix Gibson I vznikl v roce 1965
obsahoval 29 nejpoužívanějších operací

Postupem času se upravoval:
Gibson III E – pro operandy jednoduché délky (32 bitů)
Gibson III D – pro operandy dvojnásobné délky (64 bitů)

Vhodné pro oblast vědeckotechnických výpočtů

Instrukční mix GPO-WU II (General Post Office)
původně vyvinut pro potřeby britského ministerstva pošt

Do výpočtu výkonnosti byly zahrnuty i periferní jednotky, instrukce celočís. sčítání a ukládání do paměti a čtení z ní

Vhodné pro plánovací a ekonomické úlohy

Četnost výskytu instrukcí

Rozlišujeme mezi statickou a dynamickou četností:

Statická četnost reprezentuje výskyt jednotlivých instrukcí v programu (tak jak je uložen v paměti) – nevystihuje kolik jakých instrukcí procesor skutečně provedl

Dynamická četnost je frekvence instrukcí tak, jak je vykonává procesor

Rozdíl mezi statickou a dynamickou četností je dán zejména existencí příkazu skoku (větvení, cykly) – většinou používáme snadněji zjistitelnou statickou četnost, ale vzniklá chyba není příliš velká (max. jednotky %)

Zjišťování výkonnosti

Analytické testování - spočívá v měření tzv. hrubého výkonu, kde se například zjišťuje, kolik vykreslí grafická karta trojúhelníků za sekundu. Tento typ testování informuje spíše o teoretických možnostech systému.

- s pevnou řádovou čárkou
- s pohyblivou řádovou čárkou

Aplikační testy - zjišťují, jak si daná komponenta vede při použití v reálných aplikacích. Například jak rychle dokáže procesor zkomprimovat dokument metodou ZIP nebo provede konverzi videa. Tento typ testování reálně odráží výkonnost v daných aplikacích a je tedy objektivnější.

Zkušební úlohy (benchmarky)

Zkušební úloha je vzorek zátěže, který má ověřit propustnost počítače v rámci určité aplikační oblasti

Výhodou je komplexnost – na jejich běhu se nepodílí jen procesor (jako u mixů), ale jsou ovlivněny i operačním systémem, překladačem, vstupy a výstupy, atd.

Skupiny zkušebních úloh:

- přirozené (převzaté z běžného provozu počítače)
- jádra převzatá z řešení složitých problémů
- umělé (vzniklé pouze za účelem zjišťování výkonnosti) (v současnosti nejčastější)

Whetstone

Umělá zkušební úloha, která využívá celočíselné výpočty, výpočty v pohyblivé řádové čárce, transcendentní funkce, podmíněné skoky, volání procedur a indexování polí

Program vytvořen v roce 1976 na Oxfordské univerzitě, původně napsán v jazyku ALGOL, kde se zkoumala četnost tzv. whetstonských instrukcí (whetstonský mix)

Whetstone zkušební úloha přestavuje 963 tis. whetst. instrukcí

Je určena k měření propustnosti u vědeckotechnických výpočtů

Výsledky udávány ve Whetstonech za sekundu nebo prostě jen Whetstonech (někdy zkracováno WHIPS - Whetstone instructions per second nebo MWIPS)

Whetstone - charakteristika

- používá velké množství FP dat a operací nad nimi; použita je i menší část celočíselných výpočtů
- lokálních proměnných je velmi málo (neodráží umístění lokálních proměnných do registrů)
- většina proměnných je globální (skalární proměnné a jednorozměrná pole)
- výsledný kód Whetstone je malý (může se vejít do cache)
- existují dvě varianty Whetstone – pro výpočty prováděné s jednoduchou a dvojitou přesností

Dhrystone

- zkušební úloha vytvořená v USA v roce 1984
- k dispozici ve třech jazycích: C, Pascal, Ada
- založena na provádění „typické“ sady celočíselných výpočtů (datové typy integer, string, array, pointer)
- udávána v Dhrystonech za sekundu DIPS (Dhrystone Instructions per Second), příp. jen v Dhrystonech
- vhodná zejména pro oblast systémového programování (nehodnotí PC z hlediska numerických výpočtů)

Dhrystone - charakteristika

- neobsahuje žádné FP operace
- základní datové typy: integer (54%), character (20%), výčtový typ (12%)
- cykly v Dhrystone (na rozdíl od Whetstone) mají zpravidla počet opakování rovný jedné (menší vliv vyrovnávací paměti procesoru)
- malý výskyt globálních proměnných (8%), většina proměnných je lokální (49%)

LINPACK

měří výkonnost počítačů v oblasti výpočtů s pohyblivou řádovou čárkou

vychází ze souboru podprogramů pro řešení soustav lineárních rovnic (odtud název)

Udává se v jednotkách Linpack MFLOPS

Jádrem je řešení soustavy 100 lineárních rovnic

Linpack má více verzí:

- single / double (jednoduchá nebo dvojnásobná přesnost)
- rolled / unrolled (nerozvinutý / rozvinutý tvar cyklů)

LINPACK - charakteristika

- vysoké procento FP operací, ale jen několik typů (nepoužívá se např. FP dělení, exp(x) aj.)
- díky provádění poměrně jednoduché funkce může ovlivňovat cache procesoru
- jádro výpočtu omezeno jen na několik instrukcí
- data jsou rozprostřená ve velkém prostoru (matice 100x100)

Whetstone, Dhrystone či Linpack hodnotí převážně výkonnost jen základní jednotky

SPEC2000

Standardní benchmarkový test SPEC CPU 2000 pro srovnání výkonnosti počítačů vycházející z aktuálních aplikací (komprese dat, zpracování obrazu, databáze, kompilátory)
SPECint2000 – pro celočíselné výpočty

SPECfp2000 – výpočty v pohyblivé řádové čárce

Procesor	SPECint2000	SPECfp2000
Intel Pentium 4 - 1.8GHz	612 bodů	678 bodů
AMD AthlonXP - 2200+	765 bodů	671 bodů
AMD AthlonXP - 2800+	898 bodů	782 bodů
IBM PowerPC 970 - 1.8GHz	937 bodů	1051 bodů
Intel Pentium 4 - 3.066 GHz	1130 bodů	1103 bodů
IBM PowerPC 970 - 2.5GHz	1301 bodů	1460 bodů
Intel Pentium 4 - 3.6 GHz zapnutá technologie Hyperthreading	1510 bodů	1627 bodů

Systémy pro zpracování transakcí

Transakce je obecně jakýsi zásah do dat (uložených v paměťovém systému) na základě požadavku

Výkonnost systémů pro zpracování transakcí udáván v jednotkách „transakce za sekundu“ (TPS)

Není definována standardní transakce => nejednotné, velmi závisí na architektuře, na typu vstupů a výstupů, aj.

Pokusy o jednotné definování funkcí nad databází, např. simulace reálného bankovního provozu (testovací funkce Scan, Sort, Debit-Credit)

Monitorování výkonnosti počítačů

Na rozdíl od zkušebních úloh cílem monitorování je zjištění skutečného chování počítače v čase prostřednictvím hodnot několika předem zvolených stavových proměnných (např. stav procesoru, paměti, sběrnice, V/V zařízení)

Programový monitor – snadná realizace, ale nepatrně zdržuje vlastní činnost počítače

Obvodový monitor – samostatný funkční blok, který neovlivňuje počítač a je s ním spojen sondami

Kombinovaný monitor

Monitory mohou sledovat nejen činnost OS, ale i požadavky uživatele nebo vyváženost konfigurace počítače