

Procesory a jejich strojový kód

Cílem této kapitoly je seznámit se s procesory (mikroprocesory), které tvoří „mozek“ a „srdce“ celé počítačové sestavy. Poznat stavbu, základní části a funkce procesorů. Seznámit se se základními pojmy používanými u procesorů.

Klíčové pojmy:

Processor, mikroprocesor, aritmeticko-logická jednotka, řadič, registr, instrukce, strojová instrukce, vyrovnávací paměť, cache paměť, koprocessor, CISC, RISC, operand, assembler, překladač, kompilátor, interpreter, takt, interpretace instrukcí, instrukční cyklus, zásobník, příznak, RESET, signál přerušeni, priorita

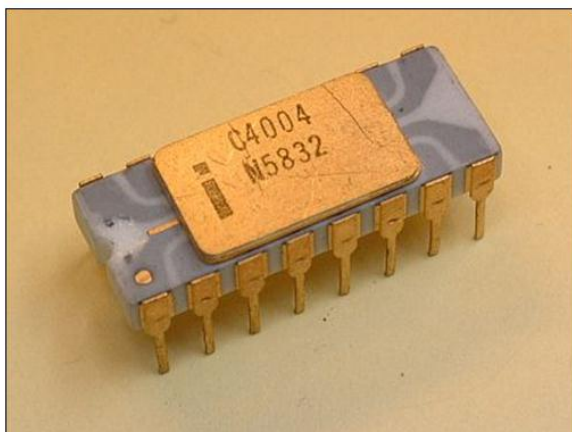
Procesory

Processor (CPU – Central Processor Unit) je základní jednotka počítače, tj. logický automat pro zpracování informací obsahující – řadič (řídící část, CU – Control Unit), aritmeticko-logickou jednotku (výkonná část, ALU – Arithmetic Logic Unit) a registry (rychlé paměti realizované klopnými obvody). Bývá umístěn na základní desce počítače.

Mikroprocesor (μ CPU) je procesor neboli programovatelný automat umístěný v jednom pouzdře (integrovaný obvod VLSI). Dnes se předpona mikro už nepoužívá.

Historie mikroprocesorů

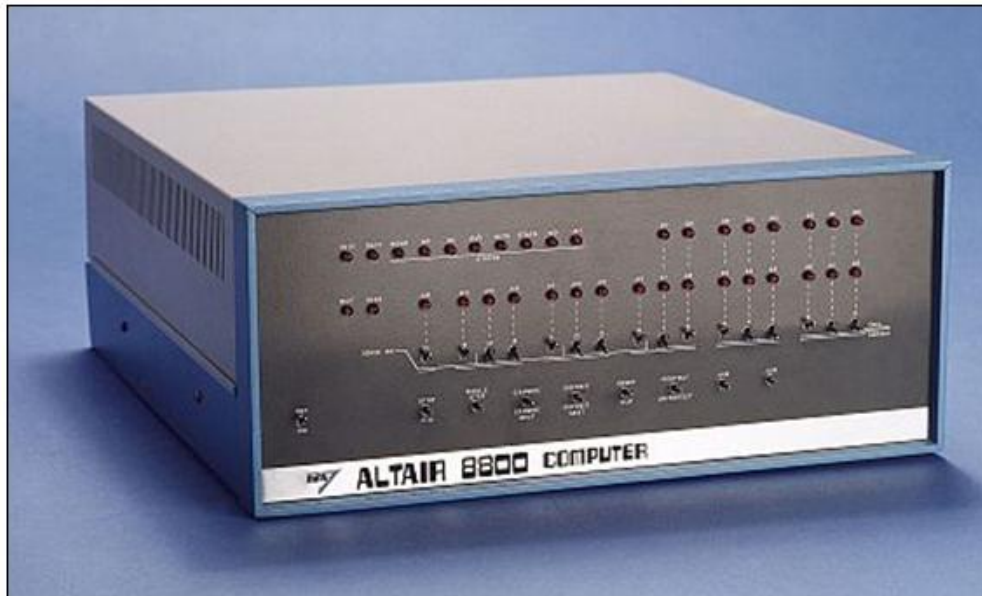
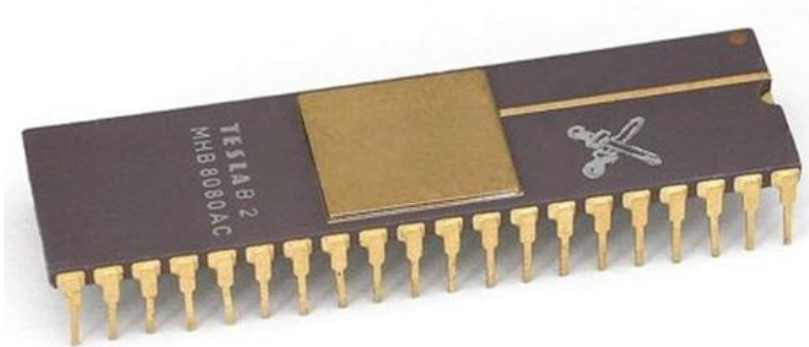
- 1971 - Intel 4004 - první mikroprocesor - 4bitový



Mikroprocesor 4004, úplně první mikročip od Intelu

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- 1972 - Intel 8008 - 8bitový mikroprocesor
- 1974 - Intel 8080 - 8bitový mikroprocesor, který se stal základem prvních 8bitových osobních počítačů



Altair 8800, první úspěšný domácí mikropočítač...

- 1975 - MOS Technology 6502 - 8bitový mikroprocesor, montovaný do AppleII, Commodore 64 a Atari
- 1975 - Motorola 6800 - první procesor firmy Motorola
- 1975 - AMD nastupuje na trh s řadou Am2900
- 1976 - TI TMS 9900 - 16bitový mikroprocesor
- 1976 - Zilog Z80 - 8bitový mikroprocesor, s rozšířenou instrukční sadou Intel 8080, frekvence až 10 MHz

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



Originální britský osmibitový počítač Sinclair ZX Spectrum

- 1978 - Intel 8086 - 16bitový mikroprocesor, první z architektury x86
- 1978 - Intel 8088 - 16bitový mikroprocesor s 8bitovou sběrnicí, který byl použit v prvním IBM PC v roce 1981



Slavný pradědeček současných počítačů: IBM PC.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

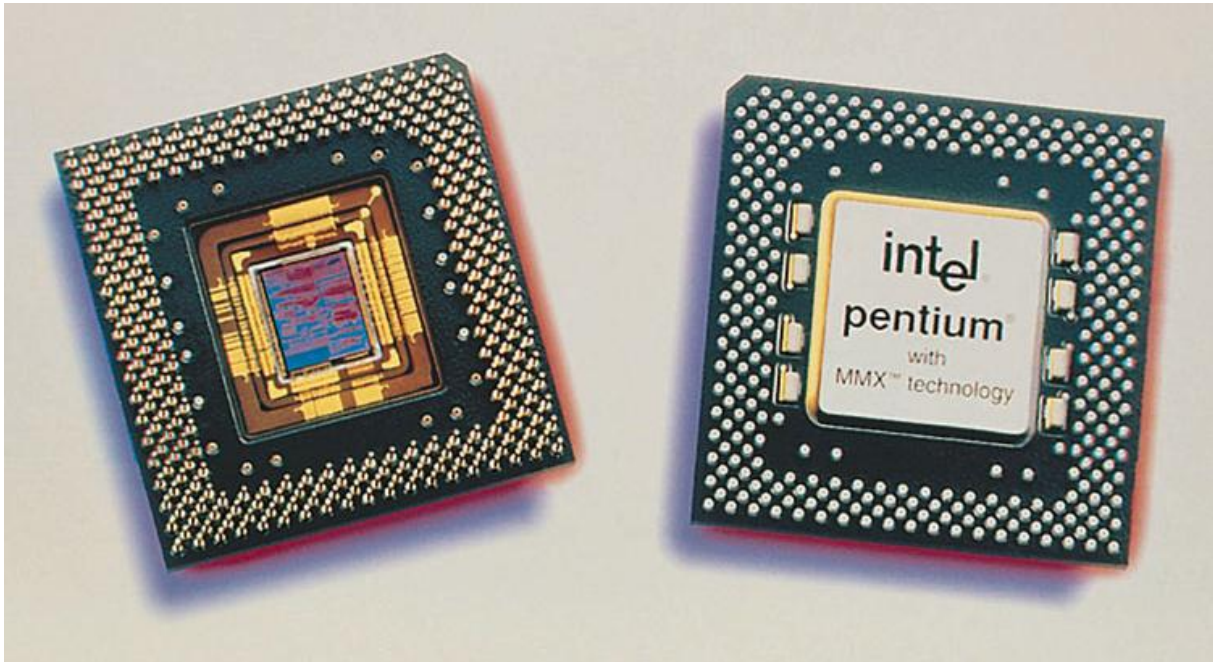
V roce 1981 představila firma IBM svůj osobní počítač PC/XT s operačním systémem MS-DOS. Procesor 8088 běžel na frekvenci 4,77 MHz.

- 1979 - Motorola 68000 - 32/16bitový mikroprocesor
- 1979 - Zilog Z8000 - 16bitový mikroprocesor
- 1980 - IBM 801 - 24bitový experimentální procesor s revoluční RISC architekturou dosahující vynikajícího výkonu
- 1980 - Intel 8051 - 8bitový mikroprocesor se základní sadou periférií pro embedded systémy
- 1982 - Intel 80286 - 16bitový mikroprocesor

Firma IBM později otevřenost architektury omezila. Cesta k dominanci IBM PC, ale už v tomto bodě byla zřejmá. Potvrdily ji i následující modely, IBM XT (1982) a IBM AT (1984) s procesorem 80286. V roce 1985 dominanci této architektury ještě potvrdila firma Microsoft, která pro operační systém MS-DOS vyvinula grafické rozhraní Windows.

- 1983 - TMS32010 - první DSP firmy Texas Instruments
- 1985 - Intel 80386 - 32bitový mikroprocesor (měl 275 000 tranzistorů)
- 1986 - Acorn ARM - 32bitový RISC mikroprocesor, z Advanced RISC Machine, původně Acorn RISC Machine, použit i v domácích počítačích
- 1989 - Intel 80486 - 32bitový mikroprocesor s integrovaným matematickým koprocesorem
- 1989 - Sun SPARC - 32bitový RISC mikroprocesor, z Scalable (původně Sun Processor ARChitecture)
- 1992 - DEC Alpha - 64bitový RISC mikroprocesor
- 1992 - Siemens 80C166 - 16bitový mikroprocesor pro průmyslové embedded systémy s bohatou sadou periférií
- 1993 - Intel Pentium - 32bitový mikroprocesor nové generace (3,3 milionu tranzistorů)

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ



- 1995 - Intel Pentium Pro - 32bitový mikroprocesor nové generace pro servery a pracovní stanice (5,5 milionu tranzistorů)
- 1995 - Sun UltraSPARC - 64bitový RISC mikroprocesor
- 1997 - Intel Pentium II - 32bitový mikroprocesor nové generace s novou sadou instrukcí MMX (7,5 milionu tranzistorů)
- 1997 - Sun picoJava - mikroprocesor pro zpracování Java bytekódu
- 1999 - Intel Celeron - 32bitový mikroprocesor odvozený původně od Intel Pentium II pro nejlevnější PC
- 1999 - Intel Pentium III - 32bitový mikroprocesor nové generace s novou sadou instrukcí SIMD (9,5 milionu tranzistorů)
- 2000 - Intel Pentium 4 - 32bitový mikroprocesor s řadou technologií orientovaných na dosažení vysoké frekvence



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- 2001 - Intel Itanium - 64bitový mikroprocesor nové generace pro servery
- 2001 - AMD Opteron - 64bitový mikroprocesor nové generace pro servery od AMD. Jedná se o historicky nejvyšší procesor, jaký kdy AMD vyrobilo.
- 2003 - AMD Athlon 64 - 64bitový mikroprocesor nové generace pro desktopy s instrukční sadou AMD64, zpětně kompatibilní s x86
- 2006 - Intel Core - 64bitová architektura, na které jsou postaveny procesory Core Duo, Core 2 Duo, Core Solo, Core 2 Quad



- 2007 - AMD uvádí novou řadu procesorů Phenom
- 2008 - Intel Core i7 - nová řada CPU od Intelu pod názvem Nehalem a AMD Phenom II, který staví na 45 nm výrobě
- 2010 - Intel vydává slabší a ořezanější procesory Core i3 a Core i5 postavené na architektuře Nehalem a AMD vydává svůj první šestijádrový procesor Phenom II X6
- 2011 - Intel vydává novou architekturu Sandy Bridge a AMD vydává první procesory s integrovanou grafikou

Parametry mikroprocesoru:

- **Rychlost** - Počet operací provedených za jednu sekundu [Hz]. Rychlost je také ovlivněna architekturou procesoru.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- **Efektivita mikrokódu** - Efektivita, se kterou jsou napsány jednotlivé mikroprogramy provádějící jednotlivé instrukce procesoru. Je to počet kroků potřebných pro provedení jedné instrukce (např.: vynásobení dvou čísel).
- **Numerický koprocessor** - Přítomnost (nepřítomnost) speciální jednotky pro přímé provádění výpočtů v pohyblivé desetinné čárce (FPU – Floating Point Operatoins).
- **Počet instrukčních kanálů** - Udává maximální počet instrukcí proveditelných v jednom taktu procesoru.
- **Šířka slova** - Maximální počet bitů, které je možné zpracovat během jediné operace [b]. Podle tohoto parametru se udává kolikabitový procesor je.
- **Šířka přenosu dat** - Maximální počet bitů, které je možné během jediné operace přenést z (do) procesoru [b]. Je dána velikostí datové sběrnice.
- **Interní cache paměť** - Kapacita rychlé interní cache paměti integrované přímo na čipu procesoru [b]. Jedná se o vyrovnávací paměť umístěnou mezi operační paměť a procesor, která slouží k zrychlení přenosu dat. Velikost musí být menší než velikost operační paměti, jinak by její použití ztrácelo význam.
- **Velikost adresovatelné paměti** - Velikost operační paměti, kterou je procesor schopen adresovat (používat) [B].
- **Počet jader** - Počet procesorů integrovaných v jednom pouzdře (čipu). Zvyšování počtu jader je v podstatě vynuceno fyzikálními omezeními.

Chlazení procesorů

Chlazení počítačů má za úkol odvést z vnitřku počítače ztrátové teplo, vznikající činností aktivních elektrotechnických součástek – hlavně procesoru (dnes se napájecí napětí snižují, aby se snížily tepelné ztráty), které v počítači zajišťují jeho funkčnost. Odvod tepla je způsoben díky vhodnému použití materiálu (dnes je neoblíbenější buď měď nebo s kombinací hliníku, kvůli pevnosti, lehkosti a hlavně cenně), pouze kovový chladič (pasivní) nebo přidání ventilátoru, dále také za použití kapaliny (nejčastěji destilovaná voda nebo voda upravená tak, aby se nestala elektricky vodivá) v oběhu nebo kapaliny pod bodem mrazu (tekutým dusíkem nebo oxidem uhličitým). Každá firma se snaží vyvinout, co nejlepší tvar, jak u chladiče pasivního/aktivního, tak u chladiče vodního, kde jde o problém, jak co nejlépe tvarovat kanálky v chladicím bloku.

Typy chlazení:

-**Pasivní chlazení** - pasivní chladič je kovová nepohyblivá součástka, která má na sobě navařená žebra pro zajištění co největší plochy z důvodu lepšího předávání tepla okolnímu vzduchu. U menších verzí chladičů může jít o výrobek zhotovený z

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

jednoho kusu kovu. Jedná se o součástku vyrobenou frézováním, upravenou laserem nebo jinak. Pasivní chlazení je zcela bezhlučné, neobsahuje žádný ventilátor. Aby bylo dosaženo stejného chladicího efektu, musí být chladič podstatně větších rozměrů, než by tomu bylo u chladiče aktivního. Díky tomu může někdy být výsledná cena pasivního chladiče vyšší než u chladiče aktivního.



Tvar pasivního chladiče

-Aktivní chlazení - je prováděno proudícím vzduchem. Proud vzduchu je obvykle vytvářen ventilátorem, nejvíce používané ventilátory v počítačích mívají rozměry o hraně 80, 90/92 nebo 120 mm a 550-4000 otáčkách za minutu, dnes jsou preferovány pomaluootáčkové ventilátory 700-1500 RPM a větší velikosti. Aktivní chlazení je použito pro chlazení procesoru.



Aktivní chladič ze zdroje

-Kombinované chlazení - je použito nejčastěji, jde o pasivní chladič, na kterém je nasazen chladič aktivní, který vytváří proud vzduchu procházející pasivním chladičem. Použití např. na procesorech (CPU, GPU) nebo i na Northbridge.

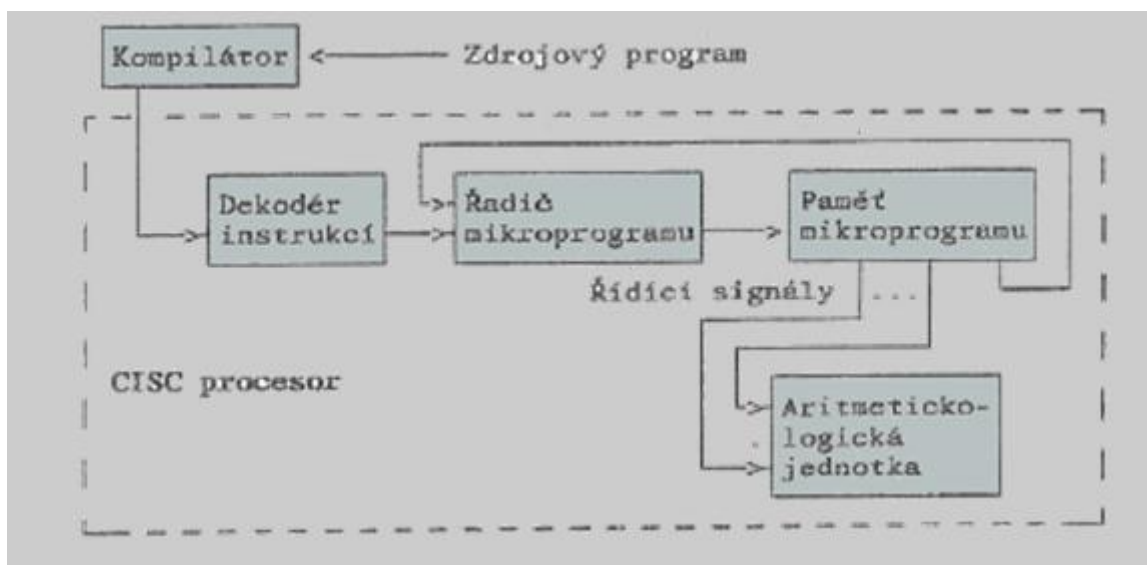
-Vodní chlazení - vzniklo z nedostatku vhodného chlazení pro počítače s nástupem výkonných sestav někdy po roce 2000. Nyní jeho časté používání zpomalují levnější avšak stejně výkonné kombinované chladiče hlavně čtyř velkých firem: Zalman, Thermaltake, PrimeCooler a částečně Nexus.

Rozdělení mikroprocesorů:

- *CISC* (Complex Instruction Set Computer)

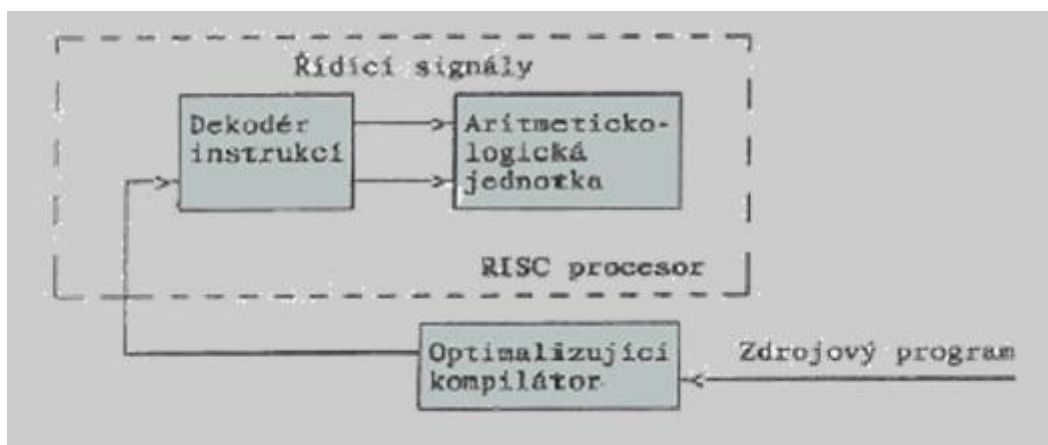
Procesory s co nejuplnějším instrukčním souborem (sadou). Instrukční sada je množina všech základních funkcí, které vyžaduje program pro realizaci aplikačních problémů. Počet instrukcí neustále stoupal a jednotlivé instrukce se stávaly stále složitější, a tak bylo nutná mikroprogramová implementace instrukčního souboru. Mikroprogram je uložen v paměti mikroprocesoru.

Tyto procesory jsou univerzální a používají se u PC.



- *RISC* (Reduced Instruction Set Computer)

Instrukční soubor tohoto typu procesoru obsahuje jen pár základních instrukcí. Cílem je dosáhnout co nejkratší doby vykonávání jedné instrukce, pokud možno během jediného strojového cyklu. Instrukce jsou vytvořeny obvodově a tudíž se většinou provádějí rychleji než u mikrokódového řešení.



Bylo zjištěno, že až 80% strojového času procesory používají pouze asi 20 instrukcí. Tyto procesory jsou jednodušší konstrukce, levnější a rychlejší. Používají se např. laserových tiskáren, serverů a v poslední době u PC.

Processor

Instrukce je předpis k provedení nějaké (většinou jednoduché) činnosti realizovatelný přímo technickým vybavením počítače (např. přičtení jedničky, uložení hodnoty do paměti apod.)

Strojová instrukce (M) je v informatice označení elementární operace procesoru, který je součástí počítače. Strojová instrukce je základní jednotkou *strojového kódu*. Každý typ procesoru má vlastní sadu strojových instrukcí, kterou je schopen přímo vykonávat.

Strojový kód- každý počítač dokáže zpracovávat jen určitý soubor instrukcí jemu vlastní. (V novější době sdílí často několik příbuzných typů počítačů tentýž strojový kód; mluvíme pak o *rodině* počítačů). Program ve strojovém kódu se skládá z jednoduchých příkazů - *instrukcí*. Instrukce (v podstatě se jedná o 1 až 3-Bytové slovo, mění se velikost adresní části, která závisí na způsobu adresování operandu) má většinou dvě základní části - kód operace, který udává, co se má udělat, a adresy (někdy i více adres), který říká, s jakými daty (*operandy*) se má operace provést. Instrukce jsou zapsány čísly (navíc zapsanými ve dvojkové či šestnáctkové soustavě), strojový kód je proto pro člověka velmi nesrozumitelný. Z toho důvodu se v něm programuje jen zcela výjimečně, a byla postupně vytvořena řada *programovacích jazyků*, které jsou pro člověka přece jenom srozumitelnější. Program zapsaný v takovém jazyce ovšem počítač přímo vykonávat neumí, a musí být proto speciálním programem - tzv. *překladačem* - přeložen do strojového kódu, který se teprve může provést (vytvoří se spustitelný soubor – např. v OS WINDOWS soubor typu .exe).

1B

1-2B

OZ	operand
-----------	----------------

**operační znak
(kód instrukce)**

adresní část instrukce

-Jazyky nižší úrovně

Tímto termínem nazýváme jazyky, které svou srozumitelností a snadností programování stojí sice nad čistým strojovým kódem, ale zůstávají vázány na konkrétní počítač. Programy napsané v nich nejsou přenositelné na počítač jiný:

· **Jazyky symbolických adres (JSA) – *assembly*** (assembler správně znamená překladač z nižšího programovacího jazyka do strojového kódu počítače). U těchto jazyků jsou adresy nahrazeny názvy, každému názvu odpovídá určitá konkrétní adresa, kterou obvykle přiřazuje překladač, a ne programátor. Téměř vždy se pro zvýšení srozumitelnosti v symbolickém tvaru zapisují i kódy operací, např. instrukce přesunu obsahu z registru A do C se zapisuje MOV C,A.

Assembly jsou daleko nejrozšířenější (v současné době prakticky jediné používané) jazyky nižší úrovně.

-**Jazyky vyšší úrovně (vyšší programovací jazyky)** – např. Pascal, C, C++ apod. Programy napsané ve zdrojovém kódu těchto jazyků se musí pomocí překladače přeložit do strojového kódu počítače. Existují dva typy překladačů:

- *Kompilátor*- celý program se celý přeloží do strojového kódu počítače (.exe) a potom teprve jej lze spustit.
- *Interpreter*- program se překládá krok po kroku a ihned se provádí.

Každý procesor se dodává s instrukční sadou (počet instrukcí závisí na typu procesoru). Typy instrukcí:

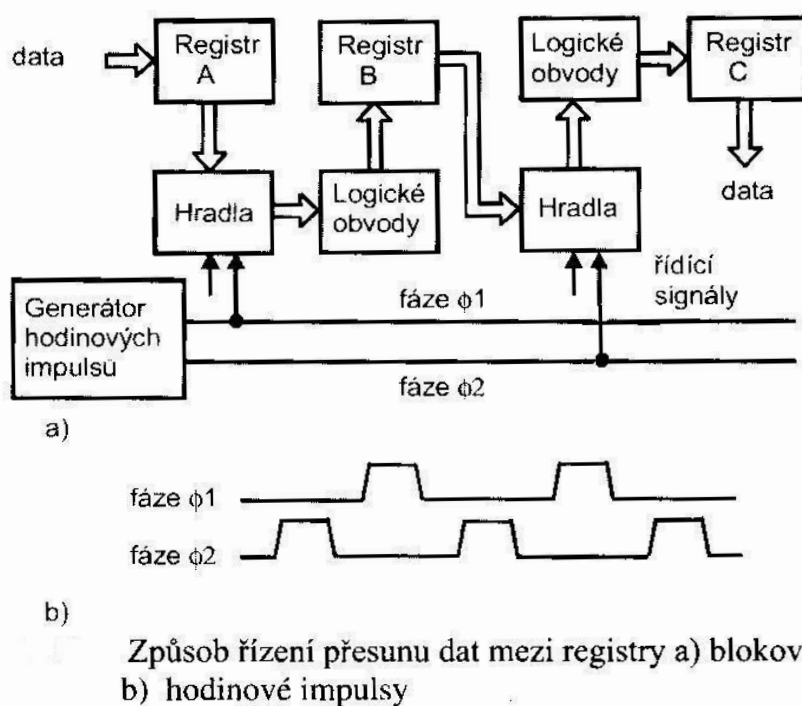
- aritmetické (sčítání, odčítání atd.)
- logické (logický součet, logická součin, negace atd.)
- instrukce přesunu (přesuny mezi registry, přesuny mezi procesorem a operační pamětí atd.)
- instrukce posuvu a rotace (posuv jednotlivých bitů střadače vpravo nebo vlevo atd.)
- instrukce skoku (skok podmíněný a skok nepodmíněný)
- řídicí instrukce (prázdná instrukce, čekání atd.)
- vstupní a výstupní instrukce (vstup z klávesnice, výstup na monitor)

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Takt (T) – nejmenší část strojové instrukce, kdy se vykonávání instrukce rozdělí na jednotlivé kroky podle taktu (hodin, kterými je řízen) procesoru Φ_1 , Φ_2 . Většinou se používají dvoufázové hodiny.

Příklad provedení instrukce přenesení obsahu z registru A do registru C – MOV C,A:

Přesuny dat probíhají v přesně stanovených časových intervalech – taktech Φ_1 a Φ_2 , které jsou určeny synchronizačními hodinovými impulzy (ústřední generátor hodinových impulzů je řízen krystalovým oscilátorem). Přesun dat mezi registry umožňuje otevřené hradlo. Vícefázové hodinové impulzy řídí tok signálů mezi jednotlivými bloky ALU. V taktu Φ_1 se přesouvají data z registru A do registru B, v taktu Φ_2 se přesunou data z registru B do registru C.



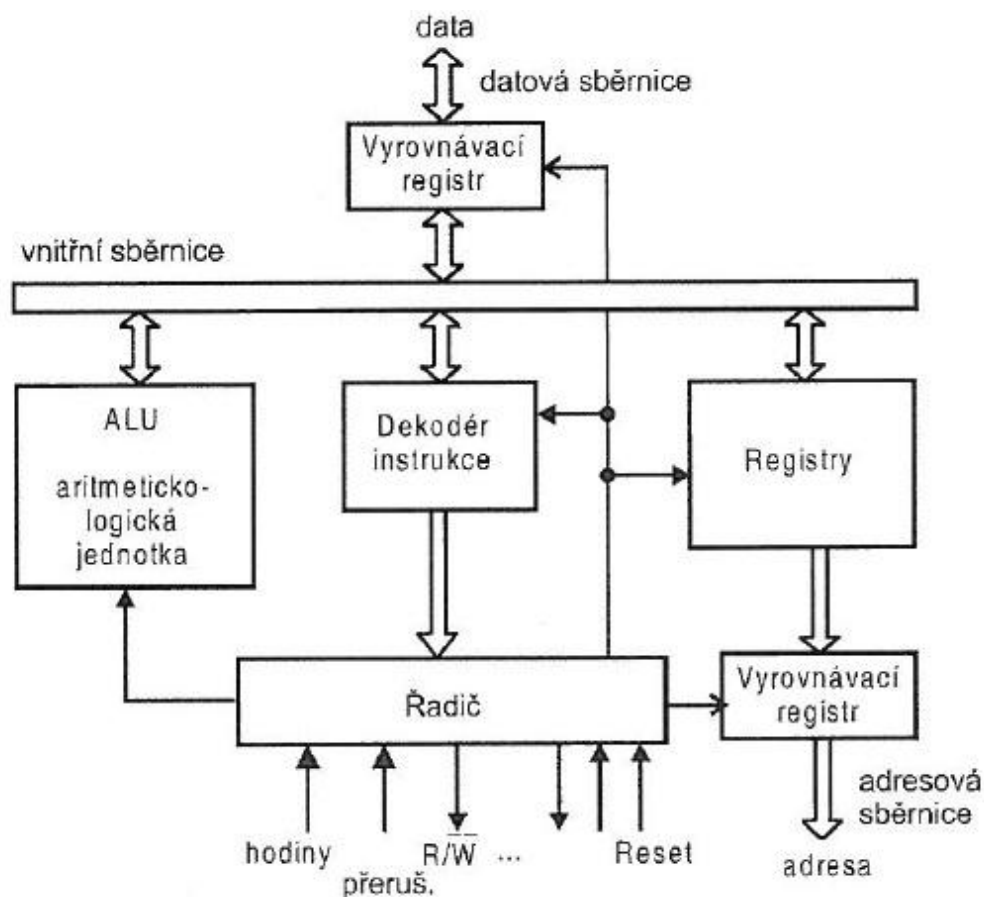
Blokové schéma mikroprocesoru

Procesor obsahuje tyto části:

- Řadič (řídící část – podle dekodéru instrukcí posílá řídicí signály ke všem částem počítače)
- ALU (výkonná část – vykonává veškeré aritmetické a logické operace)

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- Dekodér instrukcí (dekóduje jednotlivé instrukce – podle operačního znaku určí o jakou instrukci se jedná a podle adresní části, kde se vezme operand)
- Čítač instrukcí (PC- Program Counter) – registr, kde se ukládají adresy následující instrukce v operační paměti. Instrukce jsou v operační paměti uloženy na adresách následujících po sobě, a proto se obsah registru instrukcí automaticky zvyšuje o 1 ($PC=PC+1$)
- Registry adresy a instrukcí
- Pomocné řídicí a časovací obvody (tyto obvody taktují procesor-řadič a obsluhují veškerou komunikaci řadiče s okolím)
- Vyrovnávací registry (jsou zapotřebí pro obousměrný přenos dat a adres)



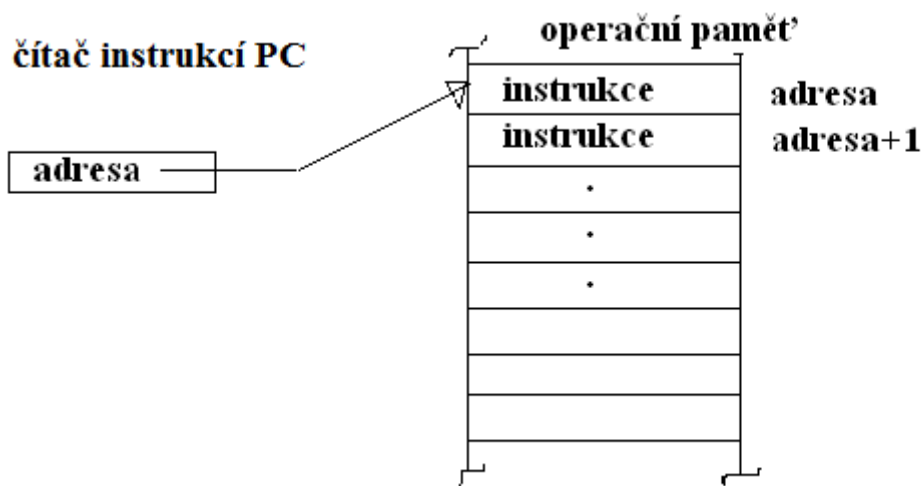
Toto schéma procesoru se nazývá sběrnicové, protože veškerá komunikace v procesoru probíhá po vnitřní sběrnici.

Činnost procesoru

Základní úkolem procesoru je *interpretace instrukcí* v posloupnosti dané programem. Interpretací instrukcí rozumíme:

1. převzetí instrukce z operační paměti
2. dekódování instrukce
3. provedení předepsané operace
4. příprava k převzetí další instrukce.

Procesor zajišťuje opakované provádění operací – *instrukční cyklus*.



Tuto činnost můžeme popsat takto:

Na začátku provádění programu je do čítače instrukce buď uložena 0 nebo adresa, od které je program ve strojovém kódu uložen v operační paměti (záleží na typu procesoru). Pomocí čítače instrukcí se na příslušné adrese vybere instrukce (ad.1), kterou dekóduje dekodér instrukcí (ad.2). podle operačního znaku se v ALU provede operace s operandy, jejichž adresy jsou uloženy v adresní části instrukce (ad.3). Po provedení instrukce se obsah čítače instrukcí automaticky zvýší o 1 (s výjimkou instrukce skoku, kdy je do čítače instrukce uložena adresa, na které program pokračuje) a k vyzvednutí z operační paměti je připravena další instrukce (ad.4).

-Aritmeticko-logická jednotka (ALU - Arithmetic Logic Unit):

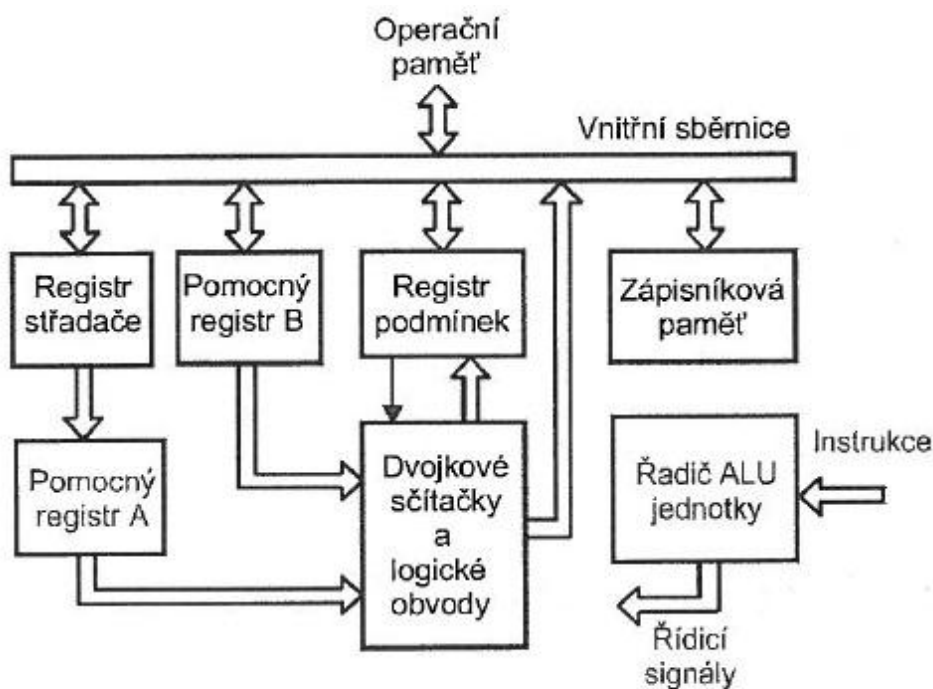
- Část procesoru, která provádí výpočty.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

- Prováděné operace jsou na základní úrovni (sčítání, odčítání a základní logické operace).
- Jádrem ALU je operační blok, který zpracovává operandy přivedené na vstupy. Na jeho výstupu se výsledek operace předá k dalšímu zpracování.

ALU jednoduchých procesorů umí pouze sčítat, všechny ostatní operace se řeší aplikačními programy výrobce, ale musí umět všechny základní logické operace. Výkonnější procesory umí všechny základní aritmetické operace. Operandy se zpracovávají paralelně nebo sério-paralelně. ALU musí být doplněna podpurnými obvody.

Blokové schéma aritmeticko-logické jednotky



ALU obsahuje:

- Speciální registr A (střadač – akumulátor), který slouží k uložení operandu se kterým se bude provádět operace a k uložení výsledku operace (akumulace hodnot). Výsledek operace musí být po ukončení dílčího výpočtu, buď přenesen přímo na výstup (standardním výstupním zařízením je monitor), uložen v zápisníkové paměti pro další zpracování nebo do operační paměti, protože při provádění další instrukce je obsah střadače přepsán novou hodnotou.

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

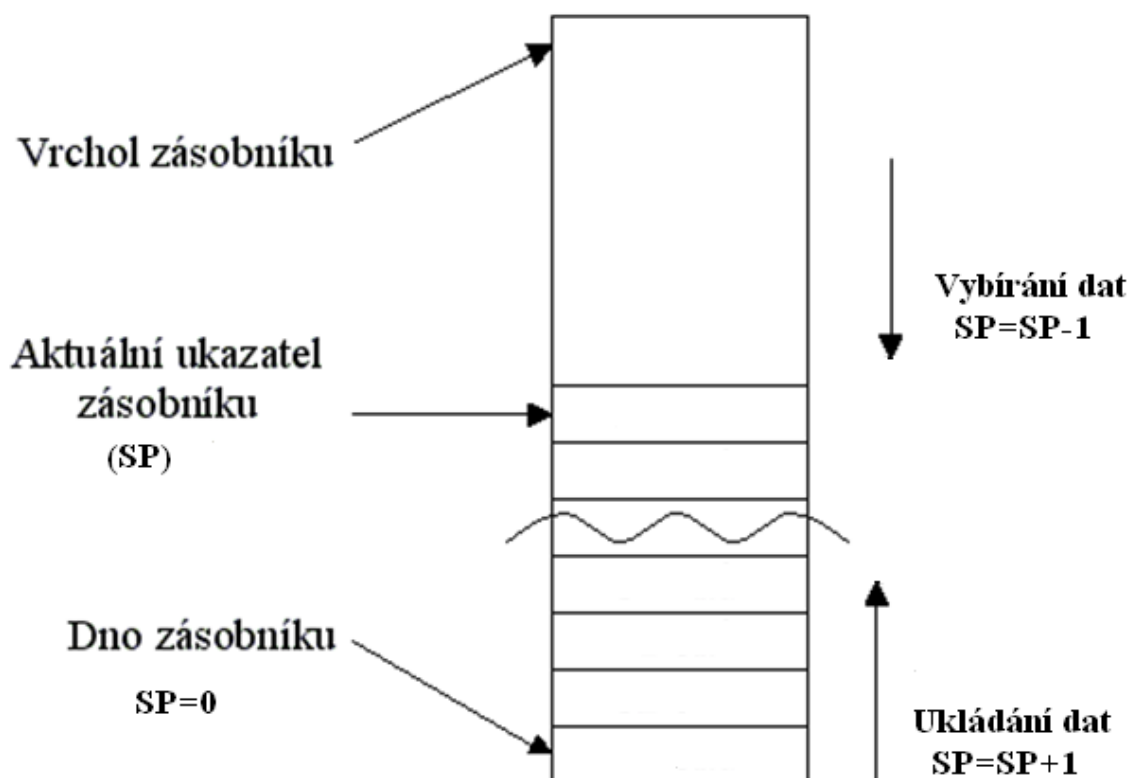
-Řadič ALU řídí provádění jednoduchých aritmetických operací, přesuny dat apod. Ostatní činnost řídí řadič procesoru.

-Blok pomocných registrů

-Zásobník (Stack) je část operační paměti, která má jednoduché adresování pomocí ukazatele zásobníku (registr SP-Stack Pointer). Používá se pro:

- Ukládání hodnot registrů
- Ukládání návratových adres funkcí při jejich volání
- Předávání parametrů ve vyšších programovacích jazycích
- Ukládání lokálních proměnných.

Organizace přístupu do této paměti se nazývá LIFO (Last In First Out – „poslední přijde první odejde“).



V registru SP je uložena adresa na hodnotu, která byla naposled vložena do zásobníku. Část zásobníku "nad" touto adresou (tj. paměť s nižšími adresami) je prázdná; je připravena na vložení dalších hodnot. Část zásobníku "pod" ukazatelem

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

SP (tedy na vyšších adresách) obsahuje hodnoty uložené dříve. Čím "hlouběji" daná hodnota leží, tím dříve tam byla vložena.

Ukládání do zásobníku probíhá způsobem "ulož hodnotu" (push value); vybírání hodnot ze zásobníku se provádí způsobem "vytáhni hodnotu" (pop value). Pokud uložíme hodnotu do zásobníku, bude tato hodnota uložena *vždy* na vrchol zásobníku. Hodnota ukazatele zásobníku se automaticky zvyšuje o 1 ($SP=SP+1$). Při vybírání bude také pouze z vrcholu zásobníku vybrána. Hodnota ukazatele zásobníku se automaticky snižuje o 1 ($SP=SP-1$).

Existuje také organizace FIFO (First In First Out – „první přijde první odejde“), kterou reprezentuje „fronta“.

-Zápisníková paměť slouží k ukládání mezivýsledků operací. U procesorů také bývá nahrazena soustavou univerzálních a jednoúčelových registrů.

-Registr podmínek (podmínkový, příznakový registr – Flags) slouží k ukládání dodatečných informací (*příznaků*) o výsledku operace. Tyto příznaky potom lze použít jako podmínky k větvení programu. Základní příznaky (jednotlivé bity) jsou:

OV (Overflow) – přetečení výsledku aritmetické operace

Z (Zero) – nulový výsledek

C (Carry) – přenos do vyššího řádu

AC (Auxiliary Carry) – přenos při operacích s BCD čísly

S (Sign) – znaménko výsledku

P (Parity) – parita

Podle výsledku operace (hodnot jednotlivých bitů) nastavujeme podmínky pro větvení programu např. „číslo je větší než nula“.

Pokud budeme provádět operace v ALU, musí se hodnota do registrů buď načíst ze zápisníkové paměti, operační paměti nebo ze standardního vstupního zařízení počítače – klávesnice. Budeme-li provádět operaci v ALU se dvěma operandy A a B, proběhne výpočet v obvodech sčítačky ve dvou taktech:

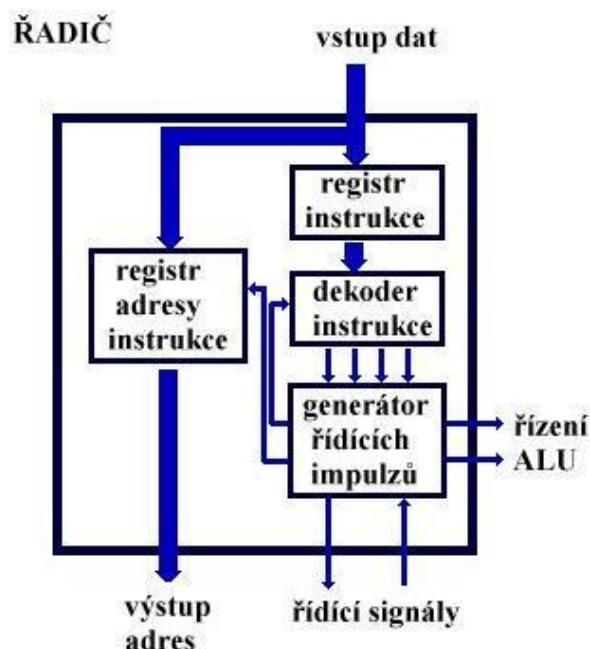
- 1. Obsah střadače se přenesou do pomocného registru A a do pomocného registru B je přiveden po vnitřní sběrnici operand B.
- 2. Obsah registru A se sečte s obsahem registru B a výsledek se uloží do registru střadače ($A=A+B$).

-Řadič (Control Unit):

- určuje pořadí, ve kterém jsou prováděny instrukce
- dekóduje instrukce a případně je modifikuje
- vysílá ostatním částem počítače řídicí signály potřebné pro provádění instrukcí

Řadič dekóduje instrukci v dekodéru instrukcí – určí o jaký typ instrukce se jedná a určí odkud se budou do ALU přesunovat operandy. Podle toho posílá řídicí signály .

Blokové schéma řadiče



Zapojení řadiče určuje instrukční soubor a zpracování instrukcí. Řadič může být řešen dvěma různými způsoby:

1. pevný řadič – výhodou je rychlost prováděných operací a nevýhodou je instrukční soubor omezený pouze na jednodušší operace. Tyto řadiče jsou používány např. v procesorech RISC.
2. mikroprogramový řadič – odpovídá procesorům CISC. Každá instrukce je vykonávána jako posloupnost jednodušších mikroinstrukcí. Tato posloupnost mikroinstrukcí představuje mikroprogram. Mikroprogramy všech instrukcí jsou uloženy v paměti mikroprogramů, do které mikroprocesor vstupuje při dekódování

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

a provádění jednotlivých instrukcí. Mikroprogramový řadič používá k vytváření řídicích signálů řídicí paměť, ve které jsou uloženy mikroprogramy složené z mikroinstrukcí. Mikroinstrukce jsou uloženy v paměti a při dekódování se určuje adresa další instrukce, takže není potřeba čítač mikroinstrukcí. Mikroinstrukce se provádí za sebou s výjimkou mikroinstrukce skoku, který se řeší mikroinstrukcemi skoku nebo modifikací adresy následující mikroinstrukce.

Zpracování instrukcí

Po připojení mikroprocesoru k napájecímu zdroji se do čítače instrukcí předává technickými prostředky počáteční adresa. V paměťovém místě s touto adresou musí být zapsána první instrukce tzv. inicializačního programu, který uvede mikropočítač do počátečního vnitřního stavu. Jedná se o stav nazývaný *RESET* a je vyvolán zapnutím počítače nebo signálem na stejnojmenné vstupní svorce procesoru. Procesor vykonává instrukční cyklus (bylo popsáno výše) a průběžně testuje *signál přerušení*. Pokud je detekován požadavek na přerušení, dojde k pozastavení zpracování instrukcí. Procesor obslouží přerušení a vrátí se ke zpracování následující instrukce. Žádost o přerušení může zaslat např. periferní zařízení, které chce odeslat data. Žádosti mají různou naléhavost (*prioritu*) a podle toho jsou obsluhovány.

Shrnutí:

Základním stavebním prvkem počítačů je procesor (dnes mikroprocesor), který se skládá z ALU, řadiče a registrů. Procesor provádí činnost podle programu, což je posloupnost instrukcí.

Použité zdroje informací:

- [1] ANTOŠOVÁ, M. - DAVÍDEK, V. Číslicová technika: učebnice. 1.vyd. České Budějovice, KOPP, 2004. 286 s. ISBN 80-7232-206-0.
- [2] KESL, J. Elektronika III: číslicová technika. 1.vyd. Praha, BEN, 2003. 112s. ISBN 80-7300-076-8.
- [3] BLATNÝ, J. a kol. Číslicové počítače. 1.vyd. Praha, SNTL, 1980, 496s.
- [4] JANSEN, H. a kol. Informační a telekomunikační technika. 1.vyd. Praha, Europa-Sobotales cz.s.r.o, 2004, 400s. ISBN 80-86706-08-7.
- [5] HÄBERLE, G. a kol. Elektrotechnické tabulky pro školu i praxi. 1.vyd. Praha, Europa-Sobotales cz.s.r.o, 2006, 460s. ISBN 80-86706-16-8.
- [6] LIČEV, L. – MORKEŠ, D. Procesory: architektura, funkce, použití. 1.vyd. Praha, Computer Press, 1999. 260 s. ISBN 80-7226-172-X.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

[7] PINKER, J. Mikroprocesory a mikropočítače. 1.vyd.Praha, BEN, 2004. 160 s. ISBN 80-7300-110-1.

[8] HORÁK, J. Hardware: učebnice pro pokročilé. 4.vyd. Brno, Computer Press, 2007, 360s. ISBN 978-80-251-1741-5.

[9] <<https://www.feec.vutbr.cz/>>

[10] <<https://www.fit.vutbr.cz/>>

[11] <<http://www.fi.muni.cz/usr/pelikan/>>