**PROCESOR**

Procesor je úst řední výkonnou jednotkou po čítače, která čte z paměti instrukce a na jejich základ ě vykonává program. Primárním úkolem procesoru je řídit činnost ostatních částí po čítače včetně vlastních jednotek a provád ět matematické a logické operace s operandy (čísla ve dvojkovém vyjád ření) na základ ě instrukcí.

Protože procesor, který by vykonával program zapsan ý v n ějakém vyšším programovacím jazyku by byl příliš složitý, má každý procesor sv ůj vlastní jazyk - tzv. strojový kód. Ten se podle typu procesoru skládá z jednodušších nebo složit ějších instrukcí, které jsou řadičem procesoru přeloženy (dekódovány) na tzv. mikroinstrukce, pomocí kterých jsou řízeny ostatní části procesoru a počítače. Instrukce, které dokáže procesor rozpoznat a zpr acovat pak tvoří tzv. instrukční sadu procesoru.

Současné procesory jsou dnes realizovány jako integrova ný obvod s vysokou hustotou integrace součástek na čipu.

**Typy procesorů**

1. MCU *(Micro Controller Unit)* - nejjednodušší skupina procesorů, tzv. mikrořadiče, jejichž uplatnění je velmi široké, od běžné spot řební elektroniky až po výkonné po čítače. Mezi výhody t ěchto čipů patří nízká cena, malé rozm ěry a nízká spot řeba energie. Tyto procesory jsou vyráb ěny pro přesně určenou specifickou činnost a mají jen malou možnost rozší ření. Rovněž ve výkonnosti nedosahují vysokých hodnot.
2. CPU *(Central Processor Unit)* - tvoří základní řídící jednotku počítače. Oproti předcházející skupin ě mají mnohem vyšší výkon, větší rozměry, je možné je díky jejich otevřené architektuře a velkému množství vyvedených signál ů lépe rozšiřovat. Samozřejmě tyto výhody jsou zaplaceny vyšší spotřebou a ztrátovým výkonem, rovn ěž cena je vyšší než v p ředchozím případě.
3. DSP *(Digital Signal Processor)* - je určitým kompromisem mezi ob ěma předcházejícími

skupinami procesorů. Signálové procesory se v ětšinou vyznačují vysokým výkonem v oblasti zpracování matematických výpo čtů a schopností zpracovávat velké objemy dat. Součástí t ěchto procesorů jsou často i digitáln ě-analogové a analogově-digitální převodníky. DSP se používají nap říklad v oblasti měřící techniky (digitální osciloskop, atd.), ve zvukových kartách, atd.

1. NPU *(Network Processors Units)* - jsou nedílnou součástí všech p řepínačů, směrovačů a dalších síťových za řízení. Vedoucí pozici ve výrob ě síťových procesor ů má spole čnost Broadcom.
2. GPU *(Graphic Processors Units)* – jedná se o speciální procesory, které jsou sou částí

grafických karet a moderních procesor ů. Jejich úkolem je řídit zpracování obrazových dat a výpo čty fyzikálního modelu 3D scény. Program i data jsou uloženy bu ď v operační paměti nebo paměti grafické karty.

f) APU *(Accelerated Processing Unit)* – nazývá se tak čip, do kterého je v jednom pouzdře integrováno jádro (nebo jádra) CPU a GPU, tedy jak centrální procesorové, tak grafické jednotky. Jedná se o sou časné moderní procesory v osobních i přenosných po čítačích.

**Rozdělení procesorů**

1. *Podle instrukční sady*
   1. CISC (*Complex Instruction Set Computer*) – jedná se o procesor s velkou sadou instrukcí. Instrukční sada je rozšířena o nové a složit ější instrukce tak, aby pro celou operaci stačilo načtení jen jediné instrukce z paměti. Zjednodušeně řečeno platí, že na každou operaci bude existovat odpovídající instrukc e.
   2. RISC (*Reduced Instruction Set Computer*) – jedná se o procesor s redukovanou instrukční sadou. Instrukční sada obsahuje pouze jednoduché a nejčastěji používané instrukce. Instrukcí se provádí b ěhem jednoho strojového cyklu. Mikroinstrukce jsou hardwarově implementovány v procesoru, čímž je velmi výrazn ě zvýšena rychlost jejich provád ění.
2. *Podle ší řky operandu v bitech (resp. ší řka slova)*

Jedním ze základních ukazatel ů procesoru je počet bitů, tj. šířka operandu, který je procesor schopen zpracovat v jednom kroku. Zjednodušeně se dá říci, že nap ř. osmibitový procesor umí počítat s čísly od 0 do 255, 16bitový s čísly od 0 do 65535 atd.

* 1. *4-bitové, 8-bitové procesory* – určené pro velmi jednoduché aplikace (spotřebníelektronika, domácí spot řebiče, kalkulačky, jednodušší periferní zařízení, atd.)
  2. *16-bitové procesory* – určené pro středněsložité aplikace (programovatelné automaty,mobilní telefony, PDA, přenosné videohry, atd.)
  3. *32-bitové, 64-bitové procesory* – určené pro velmi složité aplikace (osobní počítače,servery, tiskárny, řídící počítač automobilu, atd.)

1. *Podle výrobce procesoru*

Na poli procesorů pro osobní počítače dominují v současnosti pouze dva výrobci – firma INTEL a AMD. Existuje však spousta dalších výrobců procesorů určených pro jiné aplikace než osobní počítače – firmy MOTOROLA, ATMEL, NexGen, Texas Instruments, Broadcom, IBM, atd.

**Významné parametry procesoru**

a) Rychlost procesoru – jedná se p ředevším o **frekvenci jádra**procesoru odvozenou od generátoru hodinových impuls ů, který je řízen krystalovým oscilátorem umíst ěným na základní desce (zjednodušen ě řečeno: 1 takt = 1 činnost vykonaná procesorem). Udává se v [Hz], resp. [MHz], [GHz]. Procesor je prostřednictvím patice (socket) připojen k lokální procesorové sb ěrnici (FSB, QPI, popř. HyperTransport), pomocí níž komunikuje s čipovou sadou a tedy s ostatními částmi po čítače. Frekvence procesoru je pak dána násobkem frekvence této procesorové sběrnice.

MIPS (Milion Instructions Per Second) - jednotka označující počet instrukcí vykonaných procesorem za jednu sekundu v násobku milión ů.

FLOPS (Floating Point Operations Per Second) - jednotka označující počet operací s čísly v pohyblivé řádové čárce (reálná čísla) za jednu sekundu.

1. Šířka slova – šířka vnitřní datové sběrnice v bitech. Neboli, jak velké číslo dokáže procesor zpracovat během jedné operace. Podle šířky datové sběrnice se říká, že procesor je 8-bitový, 16-bitový, 32-bitový, 64-bito vý
2. Napájecí nap ětí jádra procesoru
3. TDP (Thermal Design Power) – tepelný výkon proce soru. Jednotkou je watt [W]). AMD

udává TDP jako maximální dosažitelný tepelný výkon, INTEL jako typický, tedy dosažitelný p ři pracovní zát ěži procesoru.

1. Patice (Socket) – slouží k uchycení procesoru na základní desce. N ěkteré jednodušší procesory jsou strojově připájeny k základní desce bez možnosti pozd ější vým ěny. Jednotlivé patice nejsou ve většině případů vzájemn ě kompatibilní (zaměnitelné).

f) Počet fyzických jader uvnit ř procesoru – více fyzických jader procesoru umož ňuje paralelní zpracování instrukcí a dat b ěhem jednoho hodinového taktu. Výrazně zvyšuje výkon procesoru, pokud paralelní zpracování podporu je operační systém a programová aplikace. (jádro = core)

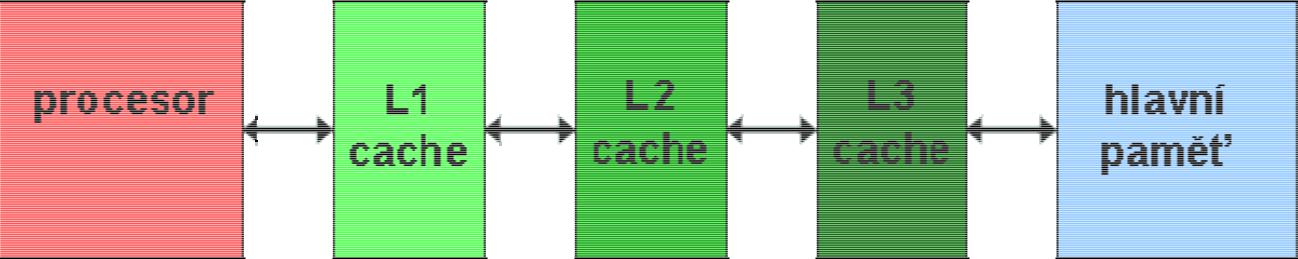
1. Instrukční sada – CISC, RISC, EPIC
2. Velikost adresovatelné paměti – udává velikost opera ční paměti, kterou je procesor

schopen používat (adresovat). Maximální velikost ad resovatelné paměti jsou 4 GB pro 32 bitovou adresovou sběrnici (232 = 4.294.967.296, tedy 4 GB).

1. Velikost vyrovnávací pam ěti (cache paměť) – uvnitř procesoru je umístěna vyrovnávací

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| statická pam ěť | | RAM, | která urychluje | komunikaci procesoru | s opera ční pamětí. | Jde |
| v podstatě | o velmi rychlý pam ěťový | | | zásobník, ur čený k do časnému ukládání dat | | a |
| instrukcí, | které | bude | procesor s největší pravděpodobností | | požadovat. Vyrovnávací | |

paměť procesoru bývá obvykle dvoustup ňová (u n ěkterých procesor ů třístupňová).



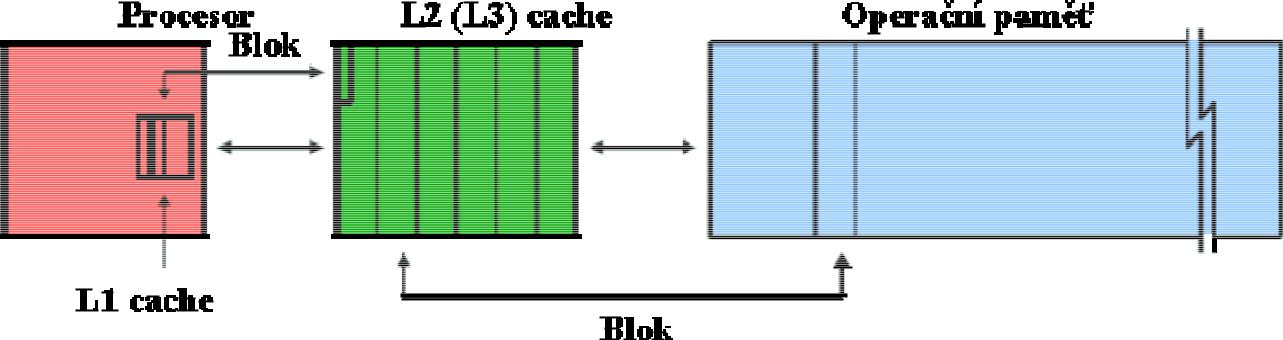
První stupeň (L1) paměti cache má malou kapacitu ( řádov ě desítky kB), je přímo součástí procesoru a je stejně rychlá jako vlastní procesor. D ělí se na instrukční a datovou cache. Slouží k ukládání práv ě zpracovávaných instrukcí a dat v procesoru.

Druhý a t řetí stupeň (L2, L3) paměti cache je pomalejší, ale s větší kapacitou (řádov ě jednotky MB), je mezi L1 cache a operační pamětí (dnes se již umis ťuje do pouzdra společně s jádrem procesoru). Obsahuje data a instrukce, kt eré procesor aktuáln ě nepoužívá, ale pravd ěpodobně bude potřebovat. Protože cena pam ěti stoupá s její rychlostí a kapacitou, je možné tímto uspo řádáním najít kompromis mezi cenou a rychlostí.

Princip vyrovnávací pam ěti L2

Vyrovnávací pam ěť L2 je fyzicky umístěna na datové sběrnici mezi procesorem a operační pamětí. Paměť je rozdělena do bloků o konstantní velikosti, které se nazývají *cache line*. Velikosti těchto blokůjsou různé, v praxi se pohybují od 8 bajtůdo 1 kB.Například vyrovnávací pam ěť o kapacitě 512 kB a velikosti bloku 32 bajtů obsahuje celkem 16384 bloků.

Operační paměť je taktéž rozd ělena na bloky o stejné velikosti, těchto bloků je však mnohem více, než blok ů ve vyrovnávací pam ěti cache. Z tohoto vyplývá, že zdaleka ne všechny bloky operační paměti mohou být obsaženy ve vyrovnávací pam ěti. Na strategii výb ěru bloků je závislá efektivita vyrovnávací pam ěti.

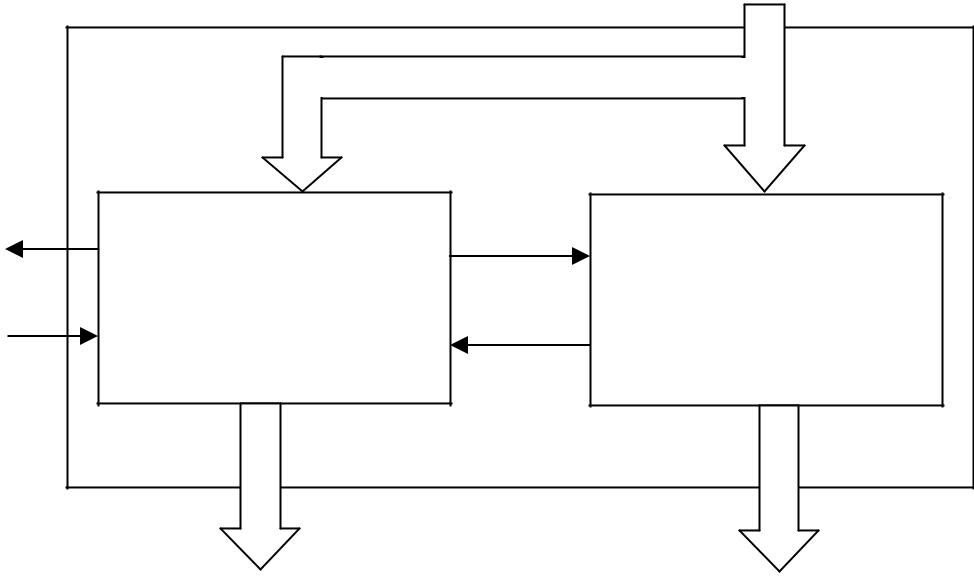


Požaduje-li procesor na číst určitý obsah pam ěťové buňky na určité adrese v operační paměti (může se jednat jak o instrukce, tak o zpracovávaná da ta), provede se kontrola, zda se tato data nachází ve vyrovnávací pam ěti. Pokud ano, jsou data z vyrovnávací paměti přečtena a předána procesoru mnohem rychleji, než v p řípadě čtení dat přímo z operační paměti. Ovšem ve chvíli, kdy potřebná data nejsou nalezena ve vyrovnávací paměti, musí do ní být p řenesena. Dojde k uvolnění bloku ve vyrovnávací pam ěti a data jsou do tohoto bloku načtena (přenáší se vždy celý blok) a dále se pokra čuje v přenosu do procesoru.

**Základníčásti procesoru**

Každý procesor obsahuje 2 základní části: **řadič** a **aritmeticko-logickou jednotku**.

data z OP



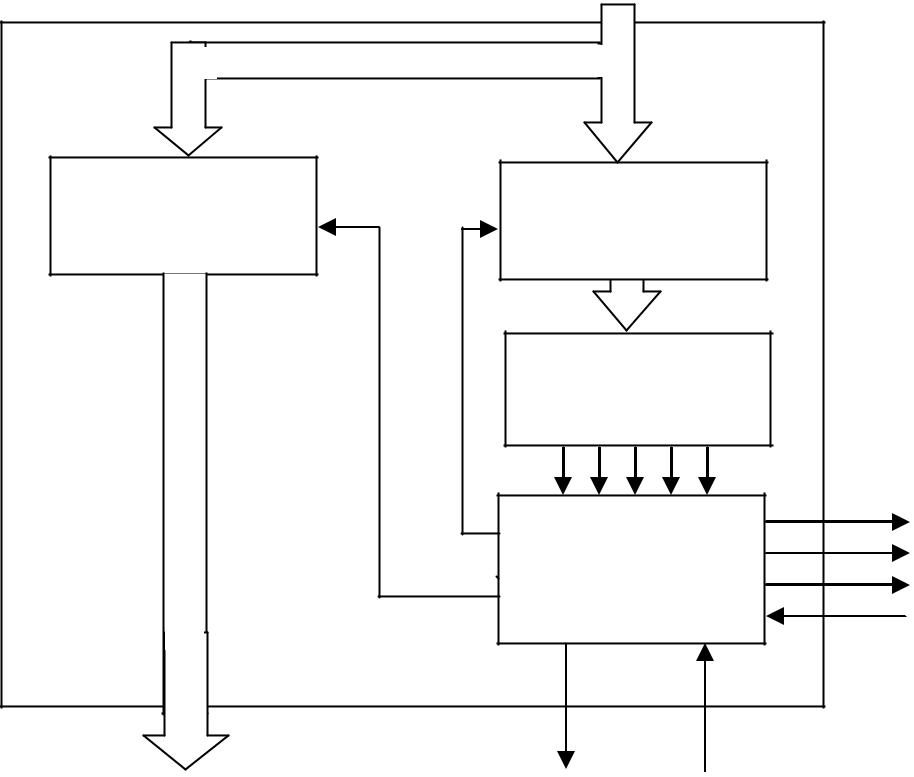
|  |  |  |
| --- | --- | --- |
| **CPU** |  |  |
| ŘADIČ | Aritmeticko-logická |  |
| jednotka |  |
| **(CU)** | **(ALU)** |  |
| řídící signály |  |  |

výstup adres data do OP

Současné procesory obsahují navíc několik jednotek ALU, dále jednotky FPU, které provád ějí operace s reálnými čísly (kdysi zajišťováno pomocí samostatného obvodu na základní desce, který se nazýval *Matematický koprocesor* ). Dále obsahuje Vyrovnávací pam ěti (CACHE) první a druhé úrovn ě urychlující práci uvnit ř CPU (L1) a komunikaci mezi CPU a operační pamětí (L2).

Řadič (CU – Control Unit)

|  |  |  |  |
| --- | --- | --- | --- |
| Je aktivní částí procesoru. Jeho úkolem je | řídit pořadí, v němž jsou provád ěny instrukce programů, | |  |
| dekóduje instrukce, vysílá do ostatních | části po čítače a procesoru řídící signály, čímž instrukce | |  |
| provádí. |  |  |  |
| **Blokové schémař adiče:** | Vstup dat |  |  |
|  |  |  |
| Registr adresy | Registr |  |  |
| instrukce | instrukce |  |  |
|  | Dekodér |  |  |
|  | instrukce |  |  |
|  | Generátor | Řízení ALU |  |
|  | řídících |  |
|  | impulsů |  |  |
| Výstup adres | řídící signály |  |  |



1. *Registr adres instrukci(RAI)* – obsahuječíslo od nuly až do hodnoty adresy poslední

paměťové buňky v OP. Touto hodnotou je omezena velikost OP (operační paměť), kterou lze k CPU připojit. Např. u 32 bitového registru lze adresovat max. 232 = 4 GB. Při zapnutí počítače se RAI nastaví na výchozí hodnotu, zpravidla nulu . Tímto nulovým obsahem se na výstupu RAI objeví číslo, které odpovídá adrese první instrukce uložené v OP. Výstup RAI je spojen s adresovou sběrnicí.

1. *Registr instrukce* – do tohoto registru se po datové sběrnici přivádí z OP instrukce

(nalezená pomocí adresy). Zde se uloží až do okamži ku, než je p řepsaná instrukcí následující.

1. *Dekodér instrukce* – instrukce je přivedena z registru na dekodér, kde se dekóduje apostupuje do generátoru řídících impulsů (GŘI). Procesem dekódování se rozumí nalezení mikrokódu k vykonání instrukce ve vnit řní paměti ROM řadiče.
2. *GŘI* – spuštěním mikrokódu se v určitéčasové posloupnosti generujířídící impulsy doostatních jednotek počítače.

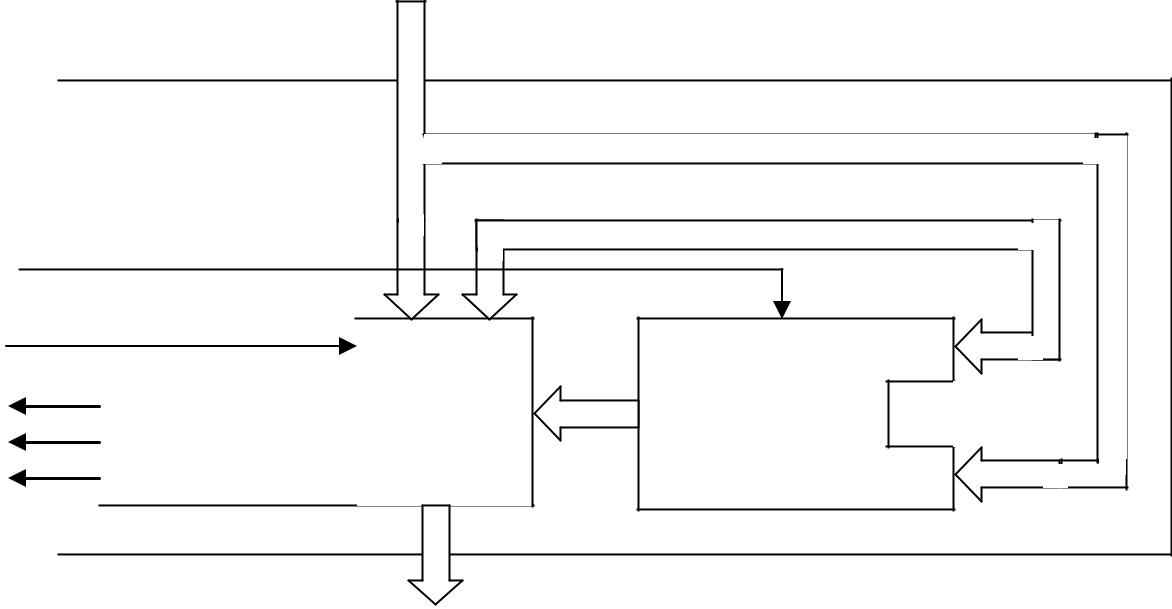
**Princip činnosti:**

*P*o zapnutí počítače nebo restartu se RAI nastaví na výchozí hodnotu a na výstupu RAI se objevíkódové číslo adresy paměťového místa OP s první instrukcí. Toto místo se připojí na datovou sběrnici a instrukce se přenese do řadiče CPU. Zapíše se do RI. Doba zápisu sta čí k tomu, aby se kódovaná instrukce dekódovala v DI. Dekódováním se vygenerovala vnitřní adresa pro spuštění mikrokódu a generují se řídící signály. Je-li instrukce ukon čena nebo vyžadují-li se data k vykonání instrukce, vyšle GŘI impuls do RAI, obsah RAI se zvýší o 1 a na výstup u se objeví kód nové adresy s následující instrukcí nebo daty. P říslušná pam ěťová bu ňka se zase napojí na datovou sběrnici a celý proces se opakuje. V programu m ůžeme provád ět skoky tím, že p říslušnou instrukcí (např.go to) zvýšíme obsah RAI o více než 1.

Aritmeticko-logická jednotka (ALU - Arithmetic-Logi c Unit)

Provádí s daty p říslušné aritmetické a logické operace.

Vstup dat



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  | Registr |  |
| řadič |  | příznaků |  |
|  |  |  |
|  |  |  |  |
|  |  |  |  |

Střadač Operační

blok

Výstup dat

1. *Operační blok* – zpracovává operandy přiváděné na dva vstupy a výsledek se předává jednímvýstupem k dalšímu zpracování.
2. *Střadač* – je registr, v němž se uchovávají data – 1. operand, mezivýsledky, v ýsledky.

3. R*egistr příznaků* – stavový registr. Skládá se z řady jednobitových pam ětí, ve kterých je uložena 0 nebo 1, podle výsledk ů. Obsah stavových registr ů kontroluje řadič, který na n ě příslušně reaguje. Nejčastější příznaky:

* CARRY (příznak přenosu) – signalizuje přeplnění střadače – přenos do vyššího řádu.

· ZERO (příznak nuly) – nastaví se na hodnotu 1 je-li výslede k operace nula.

* SIGN (signum) – příznak znaménka (+,-)

**Princip činnosti ALU**

Činnost ALU je řízena řadičem na základ ě instrukcí přijatých z OP. ALU pracuje ve v ětšině případů na základ ě jednoadresových instrukcí. Chceme-li nap ř. sečíst dvě čísla, musíme k tomu uvést více než jednu instrukci. CPU pot řebuje znát:

* adresu prvního sčítance
* adresu druhého sčítance
* adresu, kde se uloží výsledek

U počítačů s jednoadresovými instrukcemi se první s čítanec načte do střadače, odtud do operačního bloku na první vstup a druhý s čítanec se další instrukcí načte přímo z operační paměti na druhý vstup opera čního bloku. Výsledek se uloží do st řadače. Ze střadače se další instrukcí výsledek uloží na p říslušnou adresu v OP.