14. Operandy operácií

Popíšte typy a príklady jednotlivých typov inštrukcie CPU:

Inštrukcia = najjednoduchší úkon CPU (prikazuje mu čo má vykonať so vstupnými údajmi a kam ich má uložiť

Assembler = ***Jazyk symbolických inštrukcií*** – programovací jazyk, prekladá inštrukcie do strojového kódu

Typy inštrukcií: aritmeticko-logické inštrukcie **– ADD, SUB, MUL, DIV, ABS, NEG, INC,DEC**
– AND, OR, NOT, EXOR

prenosné inštrukcie – prenos dát medzi registrami **/ MOV**
– prenos údajov z registra na vrch zásobníka **/ PUSH**
– prenos údajov z vrchu zásobníka do registra **/ POP**
– ukladanie dát do registra alebo do RAM **/ STR**
– načítavajú dáta z registra alebo z pamäte RAM do registra **/ LOAD**

bitové inštrukcie – umožňujú meniť bity v registroch **/SET, CLR, ROL, SAL, SHL**
– umožňujú meniť bity v registroch /**BTS, BSP** (Bit Test/Set Prime)

riadiace inštrukcie – Inštrukcie, ktoré riadia **program**, alebo **skoky**, alebo **cykly**, alebo **prerušenie**, alebo **zastavia vykonávanie** programu

systémové inštrukcie

Uveďte formát inštrukcie a spôsoby dekódovania inštrukcie:

Inštrukcie môžu mať **rovnakú**, alebo **pohyblivú** dĺžku

Podľa počtu adries delíme I
- s **3** adresami
- s **2** adresami
- s **1** adresami
- **bez** adresy

PEVNÁ DĹŽKA	PREMENLIVÁ DĹŽKA
Rovnaká dĺžka všetkých I	***Rôzna*** dĺžka všetkých I
I sú delené do **jednotlivých polí**	Delenie I do **polí je rôzne**
Dekódovanie I prebieha **súčasne**	Dekódovanie I prebieha **postupne**

Dekóder: **1.** dekódery polí pracujú paralelne

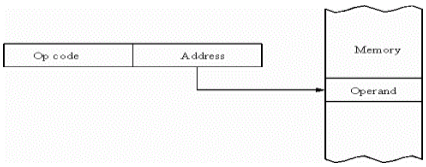
2. každý dekóder generuje 1 aktívny signál (spúšťa vykonanie operácie v PC)

Dekódovanie je riadené: **1.** časovými signálmi, z generátora hod. sign. Prechod do RJ
2. dekódovaním bitových polí I

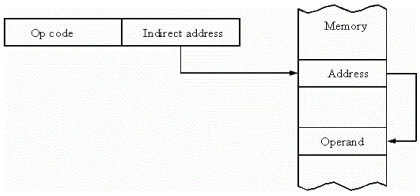
Načrtnite adresné režimy operandov v inštrukcií:

1.) Okamžité adresovanie – adresa operandu

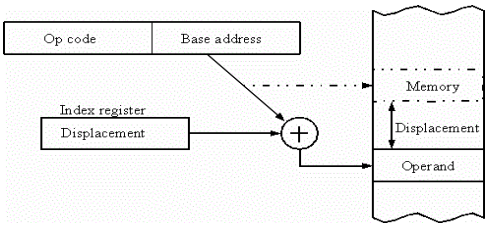
2.) Priame adresovanie

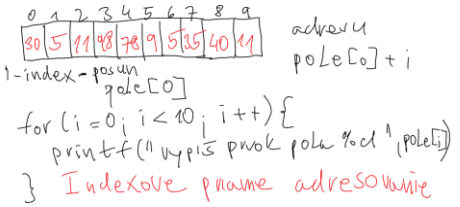


3.) Nepriame adresovanie

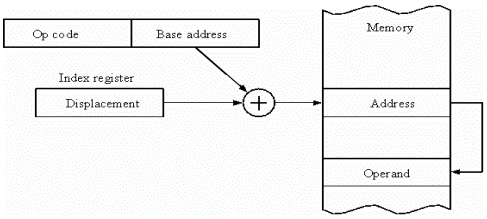


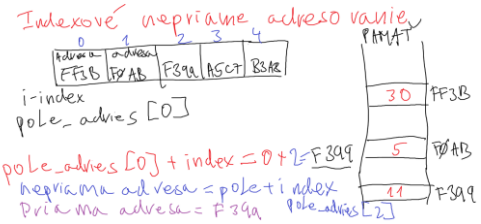
4.) Indexové adresovanie



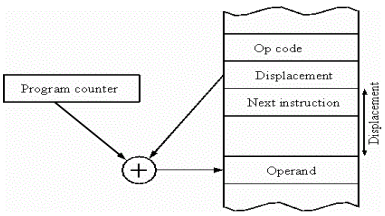


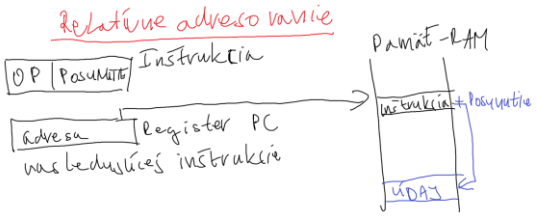
5.) Indexové nepriame adresovanie



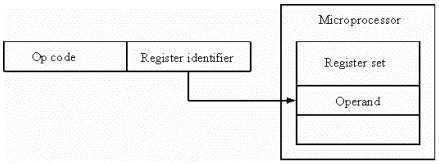


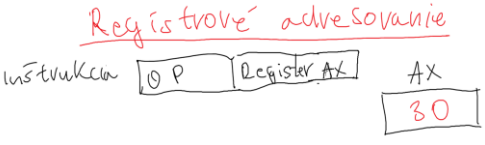
6.) Relatívne adresovanie



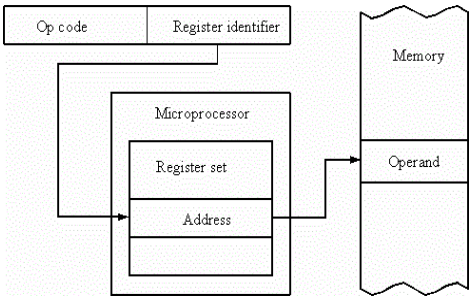


7.) Registrové adresovanie





8.) Registrové nepriame adresovanie



Funkcie

Definícia funkcie – vytvorenie, napísanie funkcie

Typ hodnoty, ktorý funkcia vráti ako výsledok

Argumenty - vstupné údaje, ktoré bude funkcia spracovávať

návratový_typ **názov_funkcie** (typ arg1, typ arg2,..) — hlavička funkcie {

- definície a príkazy - inak povedané, všetko čo funkcia robí — telo funkcie
- return – príkaz, pomocou ktorého funkcia vracia hodnotu a je ním ukončená

}

Ak funkcia nemá argumenty alebo nemá návratovú hodnotu, vo funkcii zapíšeme - void.

Deklarácia funkcie – je informácia pre kompilátor, že funkcia existuje a pomocou deklarácie kompilátor kontroluje správnosť volania funkcie.

návratový_typ **názov_funkcie** (typ arg1, typ arg2,..);

Volanie funkcie – ak funkciu chceme použiť, zavoláme ju.

názov_funkcie (hodnoty – ktoré funkcii posielame);

Ak funkcii neposielame žiadne hodnoty, zátvorky ostávajú pri volaní funkcie prázdne.

Napíšte v jazyku C funkciu average, ktorá má na vstupe tri celé čísla a vráti ich priemer:

```
float average (int x, int y, int z)
{
  int pomocna = x + y + z;
  float vysledok = pomocna/3;
  return vysledok;
}
```

Zdôvodnite použitie typu návratovej hodnoty vo funkcií average:

*Výsledok **nevychádza v celých číslach**, preto je potrebné použiť **návratovú hodnotu** typu **float**, nakoľko sa jedná o číslo s **desatinnou čiarkou**.*

Vysvetlite funkciu elektronického prepínača v osciloskope:

*Súvisí so vst. citlivosťou, umožňuje porovnať signály súčasne —> **majú ho AO aj DO***

***Umožňuje nám mať viackanálové osciloskopy**, bez toho aby sme museli robiť všetko nanovo pre každý kanál (omnoho lacnejšie). V podstate je to vysokofrekvenčný prepínač (súbor vysokofrekvenčných tranzistorov). **Nájdeme na viackanálových osciloskopoch (AO aj DO)**. Umožňuje nám zobraziť signál z jednotlivého kanálu, ich aritmetický súčet (SUM) alebo ich môže zobraziť viacero naraz (ALT/CHOP).*

***ALT** = režim ČZ – používame na rýchle priebehy*
***CHOP** = režim ČZ – používame na pomalé priebehy*

***Alernate** – vykresľujú sa kanáli v zapätí jeden po druhom **CHOP** sa vykresľujú „naraz“ (prepína sa medzi kanálmi extrémne rýchlo) - celá táto veta je pre analógový, pri digitálnom sa signály vždy vykresľujú naraz //bez úvodzoviek, fakt naraz!*