

Tryby adresowania

prefixy przed adresami w przykładowych komendach, są częścią standaryzacji trybów adresowania

Natychmiastowy:

W części adresowej znajduje się argument, np. add #5

- nie odnosimy się do pamięci w celu pobrania argumentu
- wielkość argumentu jest ograniczona przez rozmiar pola adresowego

Adresowanie bezpośrednie:

Pole adresowe zawiera adres operandu, np. add 127, gdzie 127 to adres komórki w pamięci

- jedno odniesienie do pamięci
- zakres adresacji jest ograniczony przez rozmiar pola adresowego

Adresowanie pośrednie:

Pole adresowe zawiera adres komórki pamięci w której znajduje się adres operandu, np. add @127

- zwiększa przestrzeń adresową (ograniczenie długością słowa w pamięci)

Adresowanie rejestrowe:

Argument znajduje się w rejestrze wskazanym przez pole adresowe, np. add R1

- ograniczone przez liczbę rejestrów
- szybki dostęp do argumentu

Adresowanie pośrednie rejestrowe:

Argument znajduje się w komórce pamięci wskazywanej przez zawartość rejestru, np. add (R1)

- duża przestrzeń adresowa (ograniczenie długością słowa w rejestrze)

Adresowanie względne:

A – wartość bazowa

PC - przesunięcie

Adres efektywny to $A + PC$, np. ADD #A - w powyższym przypadku adres argumentu to $A + PC$

Adresowanie indeksowe:

R – rejestr z wartością przesunięcia

Adres efektywny to argument rozkazu + R np. ADD adr(R), odnosi się do komórki o adresie równym: $adr + \text{zawartość rejestru R}$

- możliwe autoindeksowanie adres efektywny = $adr + R$; R++

Adresowanie z rejestrem bazowym:

R – rejestr bazowy, może zawierać adres pośredni

Adres efektywny to zawartość rejestru bazowego + argument rozkazu