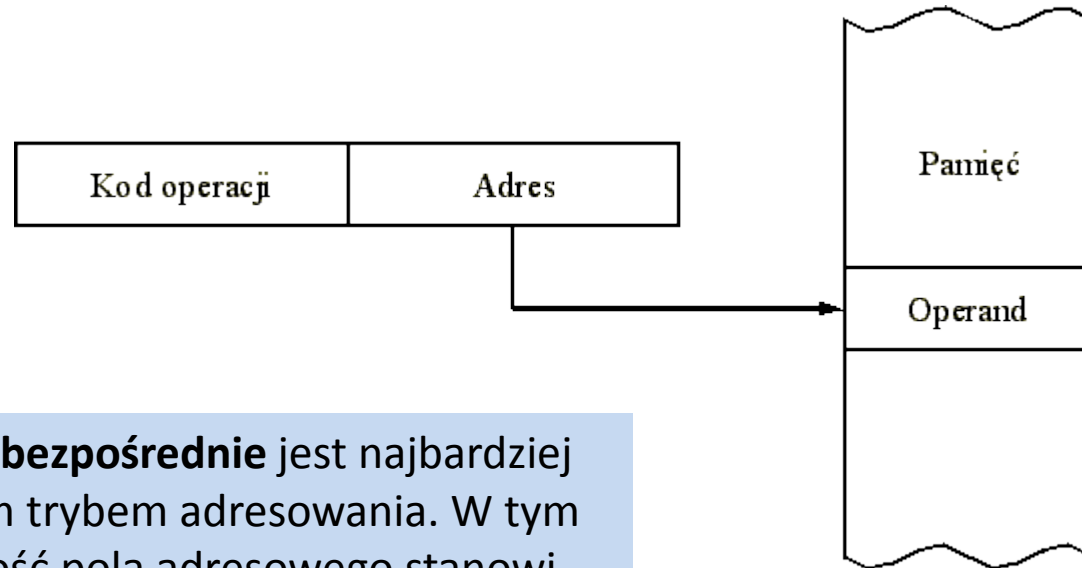


Adresowanie



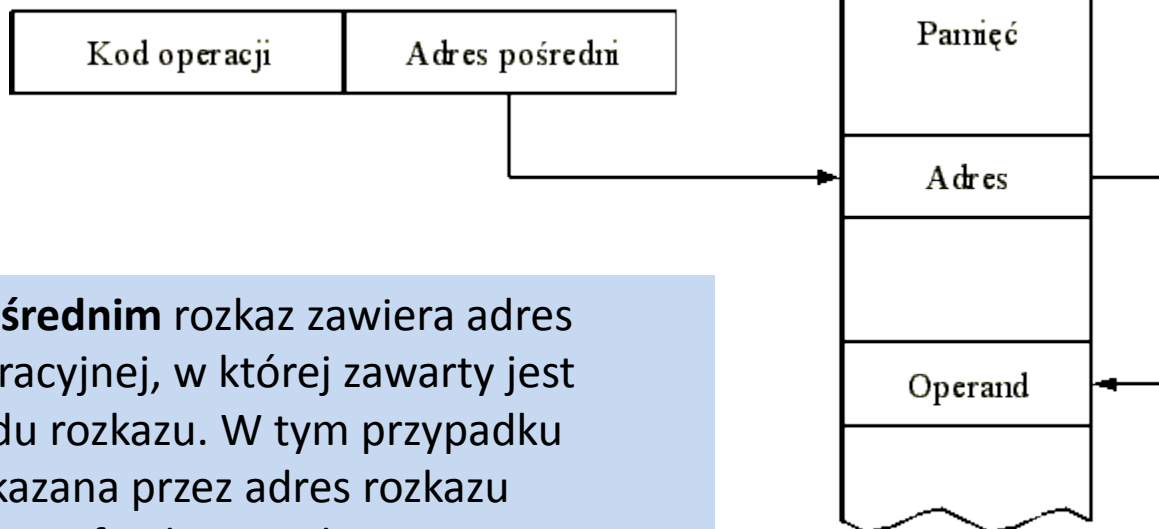
- W **trybie natychmiastowym** pole adresowe zawiera bezpośrednio **operand** czyli daną dla rozkazu.
- Wada: rozmiar argumentu ograniczony do rozmiaru pola adresowego

Adresowanie



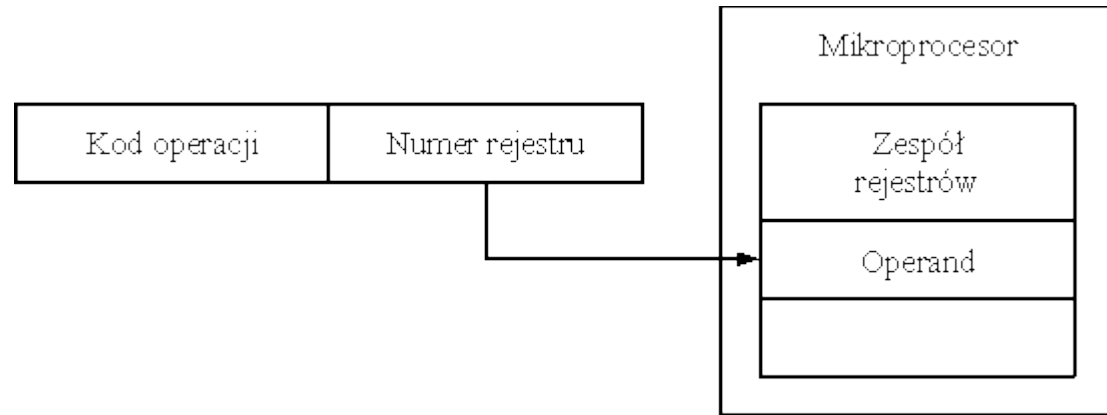
- **Adresowanie bezpośrednio** jest najbardziej podstawowym trybem adresowania. W tym trybie zawartość pola adresowego stanowi już finalny adres argumentu rozkazu w pamięci operacyjnej i nie podlega przekształceniu.
- Ten tryb stosujemy, gdy nie zależy nam na tym, aby nasz program był przesuwalny w pamięci operacyjnej, lecz jest przeznaczony do wykonania przy zapisie w ściśle określone miejsce w pamięci.

Adresowanie



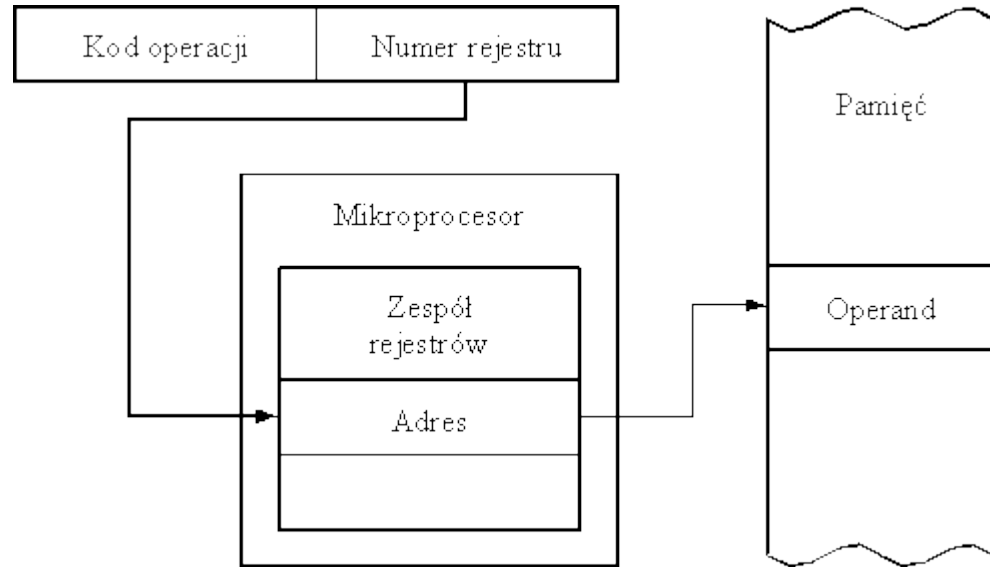
- Przy **adresowaniu pośrednim** rozkaz zawiera adres komórki pamięci operacyjnej, w której zawarty jest finalny adres operandu rozkazu. W tym przypadku komórka pamięci wskazana przez adres rozkazu pośredniczy w określeniu finalnego adresu.
- Ten tryb stosujemy, gdy chcemy, aby finalny adres operandu rozkazu mógł być dynamicznie wstawiony do komórki pośredniczącej w adresowaniu w czasie wykonywania programu. Może tak być, gdy ten adres zależy od jakichś testów na wyniku operacji poprzedzającego rozkazu.

Adresowanie



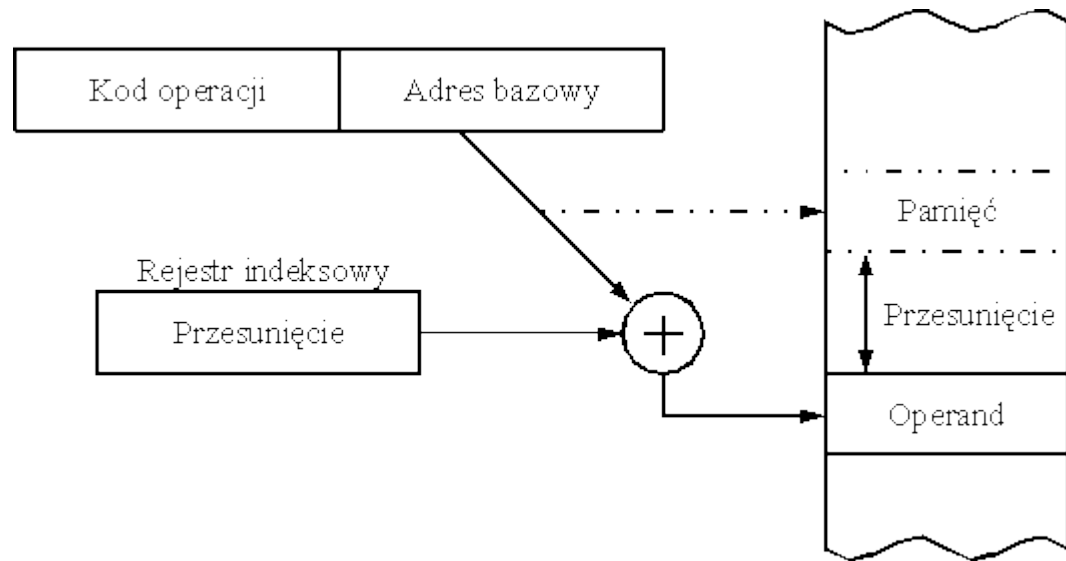
- **Adresowanie rejestrowe** stosuje się, gdy dana dla rozkazu jest przechowywana w rejestrze.
- Pole adresowe odnoszące się do rejestrów ma zwykle 3 lub 4 bity i może odwoływać się do 8 lub 16 rejestrów.

Adresowanie



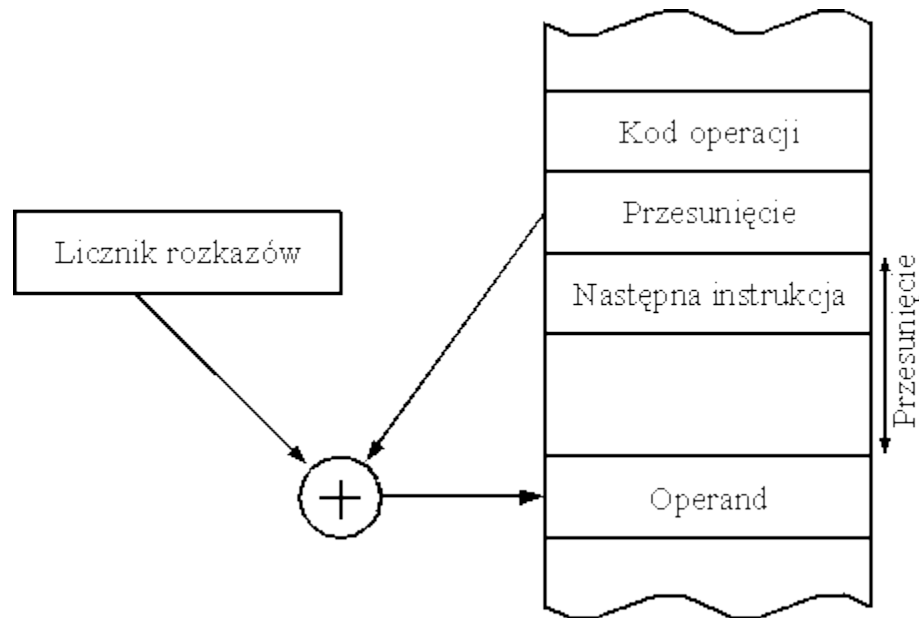
- **Adresowanie pośrednie rejestrowe** polega na tym, że jako miejsce pobrania finalnego adresu operandu rozkazu stosuje się rejestr procesora, którego identyfikator umieszczony jest w polu adresowym rozkazu.
- Przy pomocy tego trybu adresowania można dynamicznie określić finalny adres operandu poprzez odpowiednie załadowanie zawartości rejestru, w zależności od przebiegu obliczeń w programie.

Adresowanie



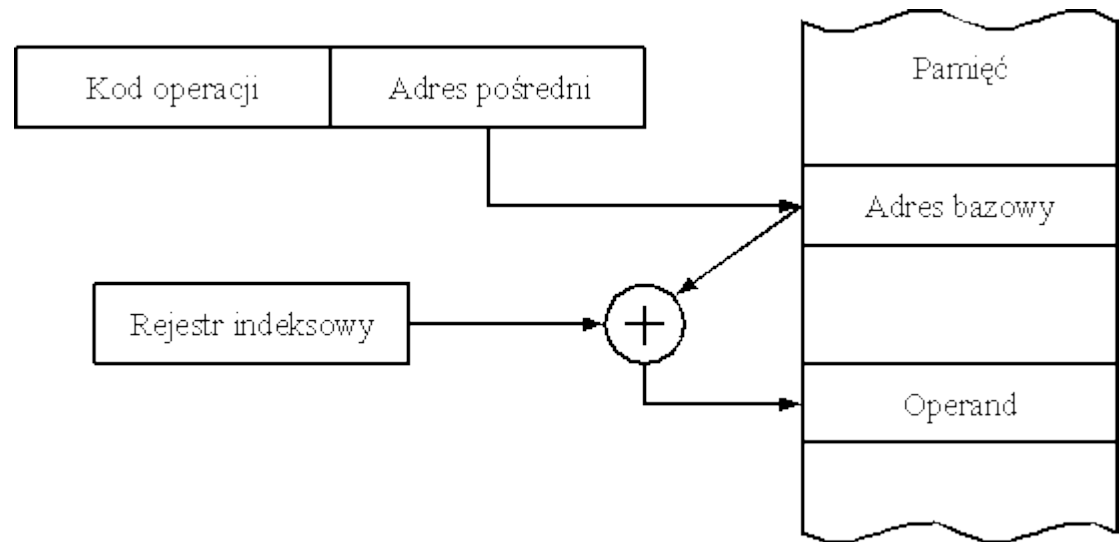
- **Adresowanie indeksowe** jest inaczej nazywane **modyfikacją adresu przez indeksowanie**. W tym trybie wykorzystuje się specjalne rejestry procesora tzw. **rejestry indeksowe (index registers)**, które zawierają przesunięcie, który trzeba dodać do adresu istniejącego w rozkazie aby wyliczyć adres finalny operandu. Ten tryb adresowania pozwala przesunąć adres zawarty w rozkazie o wartość rejestru indeksowego.
- Dzięki operacji indeksowania rozkazów programu, wszystkie adresy operandów zostaną przesunięte o tę samą wartość - przesunięcie. Taka organizacja przesuwalności programu w pamięci nosi nazwę dynamicznej relokacji programu w pamięci.

Adresowanie



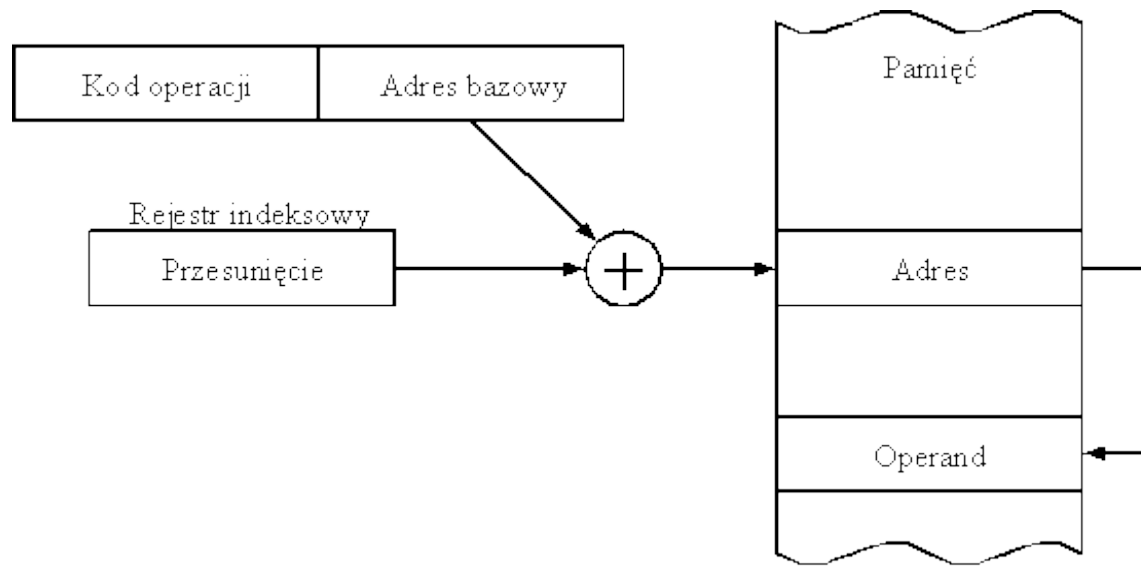
Adresowanie względne polega na modyfikacji adresu zawartego w rozkazie przez aktualną zawartość licznika rozkazów. Ten tryb adresowania dostarcza innego sposobu osiągnięcia dynamicznej przesuwalności adresów dostępu do danych, tj. gdy nie chcemy lub nie możemy znać przesunięcia całości programu w stosunku do adresu zerowego. Przy tym trybie adresowania, finalny adres danej jest wyliczany względem bieżącej zawartości licznika rozkazów,

Adresowanie



- **Adresowanie pośrednie indeksowe** zapewnia jednoczesną możliwość zastosowania w programie adresowania pośredniego z modyfikacją adresu odczytanego z komórki pośredniczącej poprzez zawartość rejestru indeksowego. Umieszczony w rozkazie adres wskazuje na komórkę przechowującą adres danej, który może być tam wstawiany dynamicznie jako wynik obliczeń programu. Do tego adresu stosowane jest następnie indeksowanie poprzez zawartość rejestru indeksowego.
- Adresowanie pośrednie indeksowe pozwala na dwupoziomowe dynamiczne określanie adresu danych w wyniku obliczeń wykonanych w poprzedzających fragmentach programu.

Adresowanie



Adresowanie indeksowe pośrednie zapewnia najpierw modyfikację adresu zawartego w rozkazie przez zawartość rejestru indeksowego a następnie tak otrzymany adres jest stosowany do wskazania komórki pamięci, w której jest przechowywany finalny adres operandu rozkazu.

Adresowanie

rozkaz

Adresowanie stosowe – adres operandu domyślny, rejestr wierzchołka stosu