***Zapis zmiennoprzecinkowy* (11.12.2022)**

Zapis zmiennoprzecinkowy lub notacja zmiennoprzecinkowa (czasem mówimy zapis zmiennopozycyjny) to jeden z najczęściej stosowanych w mikroprocesorach kodów numerycznych. Angielskie nazwy to *floating point notation* lub *scientific notation*.

Zasadniczy pomysł jest bardzo prosty. Liczby wymierne bardzo duże i bardzo małe wygodnie jest zapisywać w następującej wykładniczej postaci:

np. , , lub ogólniej w postaci

(\*)

gdzie , jest pewną liczbą wymierną, a *e* jest liczbą całkowitą. Ogólnie rzecz biorąc możemy zakładać, że ale w praktyce wykorzystujemy z reguły lub . Liczba nosi nazwę *bazy* (ang. *base*) zapisu zmiennoprzecinkowego.

W dalszym ciągu będziemy zakładać, że baza i będziemy rozważać zapis zmiennoprzecinkowy binarny ze stałą długością słowa kodowego (tzn. taki, w którym do zapisu słowa kodowego używamy liter alfabetu a słowo kodowe ma ustaloną długość np. 32 b czy 64 b). Liczba reprezentuje znak liczby *x*. Zgodnie ze wzorem (\*) oznacza liczbę dodatnią a liczbę ujemną.

Liczbę nazywamy mantysą (lub częścią ułamkową) angielskie nazwy to *mantissa* lub *fraction*. Liczbę  nazywamy wykładnikiem (lub cechą) angielska nazwa to *exponent*. Tak więc liczbę możemy reprezentować jako trójkę uporządkowaną czyli trójkę .

Oczywiście jeśli nie narzucimy dodatkowych warunków na *f* i *e* to takie przedstawienie nie jest jednoznaczne. Liczby *f* i *e* muszą być ponadto zapisane w ustalonych kodach numerycznych. Zakładamy w dalszym ciągu, że są to kody numeryczne binarne ze stałą długością słowa kodowego.

Przedstawienie liczby różnej od zera staje się jednoznaczne jeśli przyjmiemy, że liczba ma tzw. *postać znormalizowaną*. Mówimy, że liczba reprezentowana w zapisie zmiennoprzecinkowym ma postać znormalizowaną jeśli mantysa *f* zapisana jest w zapisie stałoprzecinkowym NKB bez znaku (z przecinkiem za pierwszym bitem) i ma postać:

1.ciąg skończony bitów o ustalonej długości

czyli

(*jedynka przed przecinkiem jako pierwszy bit*, *przecinek* (czyli *kropka*), *ciąg skończony bitów*)

co odpowiada wartości *f* z przedziału [1,2). Ponieważ liczby zapamiętujemy w postaci znormalizowanej nie ma potrzeby zapamiętywania pierwszego bitu mantysy *f*.

Umawiamy się, że liczbę 0 reprezentujemy za pomocą słowa kodowego złożonego z samych zer.

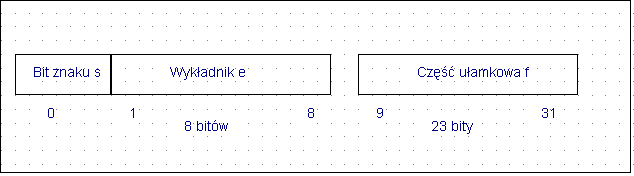
Regułą jest, że po każdej operacji realizowanej przez jednostkę arytmetyczną zmiennoprzecinkową (ang. FPU od *floating point unit*) wynik operacji jest normalizowany o ile jest to możliwe.

Najczęściej zapis zmiennoprzecinkowy używany jest zgodnie ze standardem IEEE 754. Poniżej opiszemy formaty tego standardu dla słów 32 i 64 bitowych.

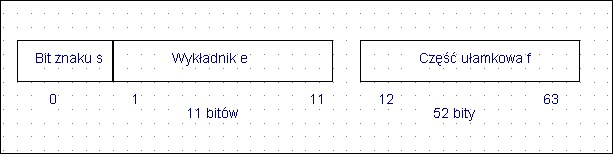
Rzadko Gęsto Rzadko

-Min 0 Max

Rys.1 Rozmieszczenie na osi liczbowej liczb reprezentowanych w zapisie zmiennoprzecinkowym. Warto zwrócić uwagę na charakterystyczne nierównomierne rozmieszczenie reprezentowanych liczb wymiernych. Odległość sąsiednich liczb jest najmniejsza w pobliżu zera i rośnie wraz ze wzrostem odległości od 0



Rys.2 Format standardu zapisu zmiennoprzecinkowego IEEE 754 dla pojedynczej precyzji (ang. *Single Precision Floating Point Notation*).). Słowa są 32 bitowe.



Rys.3 Format standardu zapisu zmiennoprzecinkowego IEEE 754 dla podwójnej precyzji (ang. *Double Precision Floating Point Notation*). Słowa są 64 bitowe.

Część ułamkowa *f* zgodnie uwagą o postaci znormalizowanej pozbawiona jest w standardzie IEEE 754 pierwszej 1. Chcąc więc odtworzyć rzeczywistą wartość *f*  należy poprzedzić ciąg bitów *f* jedynką i odczytać (1.*f*) jako liczbę zapisaną w zapisie stałoprzecinkowym NKB bez znaku.

Z kolei bit znaku *s*=1 oznacza liczb ujemną a równy 0 liczbę dodatnią.

W dosyć specjalnym kodzie zapisany jest wykładnik *e*  jest to mianowicie tzw. obciążony (lub przesunięty) kod NKB. W ogólnym przypadku dla słowa *n* bitowego jest to kod numeryczny przyporządkowujący liczbom całkowitym ze zbioru kolejne słowa kodowe kodu *n* bitowego NKB. Dokładniej, oznaczając przez *k* definiowany kod mamy:

gdzie oznacza zwykły *n* bitowy kod NKB

Ściśle rzecz biorąc wartości wykładnika *e* są liczbami całkowitymi ze zbioru . Wykładnik (czyli *n* jedynek)zarezerwowany jest dla reprezentowania obiektów specjalnych w kodzie zmiennoprzecinkowym: (plus nieskończoność), (minus nieskończoność) oraz NaN (nie liczba ang. *Not a Number*).

W standardzie IEEE 754 dla słowa 32 bitowego na zapamiętanie *e* mamy 8 bitów więc stosujemy do kodowania wykładnika *e* obciążony 8 bitowy kod NKB. Wykładnik *e* liczby reprezentowanej należy więc do zbioru .

W standardzie IEEE 754 dla słowa 64 bitowego na zapamiętanie *e* mamy 11 bitów więc stosujemy do kodowania wykładnika *e* obciążony 11 bitowy kod NKB. Wykładnik *e* liczby reprezentowanej należy więc do zbioru .

W standardzie IEEE 754 mamy jeszcze 2 obiekty specjalne i . Pierwszy reprezentowany jest przez słowo kodowe złożone z samych zer a drugi z 1 na najbardziej znaczącej pozycji(pozycji znaku) i samych zer na dalszych pozycjach.

Warto jeszcze wspomnieć o 80 bitowym (10 bajtowym) standardzie zapisu zmiennoprzecinkowego o nazwie *Double Extended Precision Floating Point Notation*, który jest stosowany we wszystkich mikroprocesorach rodziny x86 firmy Intel ale nie wchodzi do standardu IEEE 754. Podział 80 bitowego słowa na pola *s*. *e*, *f* : pole znaku *s* ma 1b (bit 79), pole wykładnika *e* ma 15 bitów (bity 64-78), pole mantysy ma 64 bity (bity 0 – 63).

Sposób rozmieszczenia liczby zmiennoprzecinkowej w pamięci operacyjnej mikroprocesora rodziny x86 jest typowy dla standardu Intela o nazwie *Little Endian*. Adres wskazujący liczbę wskazuje na najmniej znaczący bajt mantysy. Wszystkie pozostałe bardziej znaczące bajty liczby są po kolei umieszczone w bajtach o wyższych adresach.

**Przykład 1**

Przedstawić liczbę zgodnie z 32 bitowym standardem zapisu zmiennoprzecinkowego IEEE 754.

1.Przedstawiamy moduł liczby w zwykłym stałoprzecinkowym zapisie NKB. Dostajemy 10000000001 co można przedstawić jako . Zatem 23 bitowa mantysa *f* ma postać

*f* = 00000000010000000000000

a cecha *e* jest równa 10 co zapisujemy w 8 bitowym kodzie obciążonym NKB jako

*e* = 10001001

Bit znaku to oczywiście 0 ponieważ zapisywana liczba jest nieujemna. ■

**Przykład 2**

Przedstawić liczbę zgodnie z 64 bitowym standardem zapisu zmiennoprzecinkowego IEEE 754. Rozumujemy analogicznie jak w przykładzie 1 i otrzymujemy mantysę równą

0000000001000000000000000000000000000000000000000000

i cechę równą

10000001001

Bit znaku jest równy 0 ponieważ zapisywana liczba jest nieujemna.■

Warto pamiętać, że operacje zmiennoprzecinkowe mają (ze względu na wprowadzane zaokrąglenia) pewne osobliwości. Np. może się zdarzyć, że nie zachodzi prawo łączności dodawania i mnożenia:

i prawo rozdzielności mnożenia względem dodawania:

gdzie i są odpowiednio dodawaniem i mnożeniem zmiennoprzecinkowym. Uważny Czytelnik proszony jest o podanie odpowiednich przykładów.