Arytmetyka binarna

Działania na liczbach stałopozycyjnych
→ liczba reprezentowana jednym słowem

Działania na liczbach zmiennopozycyjnych
→ liczba reprezentowana trzema słowami

$L(A)$ - wartość liczbowa słowa A
zależna od sposobu kodowania

Naturalny kod binarny - NKB

Liczby całkowite

$A = a_{n-1} \dots a_1 a_0$

$L(A) = \sum_{i=0}^{n-1} a_i 2^i$

Liczby ułamkowe

$A = a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m}$

$L(A) = \sum_{i=-m}^{n-1} a_i 2^i$

tylko liczby nieujemne

Konwersja liczby ułamkowej na NKB
- pomnożyć przez 2
- odjąć cyfrę jedności

$0.17_{10} \simeq 0.00101_2$

$0.00101_2 = 0 \cdot \frac{1}{2} + 0 \cdot \frac{1}{4} + 1 \cdot \frac{1}{8} + 0 \cdot \frac{1}{16} + 1 \cdot \frac{1}{32} = \frac{5}{32}$

$0.17_{10} \rightarrow 0$
$0.34 \rightarrow 0$
$0.68 \rightarrow 0$
$1.36 \rightarrow 1$
$0.72 \rightarrow 0$
$1.44 \rightarrow 1$
...

Zapis znak-moduł (ZM)

$A = a_n a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m}$

$$L(A) = \begin{cases} +\sum_{i=-m}^{n-1} a_i 2^i & \text{dla } a_n = 1 \\ -\sum_{i=-m}^{n-1} a_i 2^i & \text{dla } a_n = 0 \end{cases}$$

- Pierwszy bit określa znak
- Zakres $-(2^{n-1}-1)$ do $+(2^{n-1}-1)$
- Dwie reprezentacje zera
- Stałopozycyjny
- Zmiana znaku - negacja bitu znaku

Zapis uzupełnień do 1 (U1)

$A = a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m}$

$L(A) = -a_{n-1}(2^{n-1}-1) + \sum_{i=-m}^{n-2} a_i 2^i$

- Pierwszy bit ma ujemną wagę
- Zamiana znaku - negacja bitów
- Zakres $-(2^{n-1}-1)$ do $+(2^{n-1}-1)$
- Dwie reprezentacje zera
- Stałopozycyjny

Zapis uzupełnień do 2 (U2)

$A = a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m}$

$L(A) = -a_{n-1} 2^{n-1} + \sum_{i=-m}^{n-2} a_i 2^i$

- Pierwszy bit ma ujemną wagę
- Naturalne dodawanie i odejmowanie
- Zamiana znaku - negacja bitów i dodanie 1
- Zakres -2^{n-1} do $2^{n-1}-1$
- Jedna reprezentacja zera
- Stałopozycyjny

Zapis Bias

$A = a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m}$

$L(A) = -2^{n-1} + \sum_{i=-m}^{n-1} a_i 2^i$

- Wszystkie wartości przesunięte o stałą
- Zakres -2^{n-1} do $2^{n-1}-1$
- Jedna reprezentacja zera
- Zmiana znaku - negacja bitów i dodanie 1

Zapis Binary coded decimal (BCD)

- Zapis dziesiętny, każda cyfra kodowana oddzielnie na 4 bitach
- Nadaje się do prezentacji, nie do obliczeń

Zapis minus-dwójkowy (negabinarny)

$A = a_{n-1} \dots a_1 a_0$

$L(A) = \sum_{i=0}^{n-1} a_i (-2)^i$

- Historyczny

Dodawanie i odejmowanie ZM
1. Jeśli znaki są takie same - dodać moduły
2. Jeśli znaki są przeciwne
   - wybrać większy moduł (komparator)
   - odjąć mniejszy moduł od większego (układ odejmujący)
   - przyjąć znak liczby o większym module

Dodawanie i odejmowanie U1
1. Dodać oba argumenty
2. Wyznaczyć przeniesienie z najbardziej znaczącego bitu
3. Dodać przeniesienie

Wynik = A + B + carry

Dodawanie liczb w zapisie Bias
1. Dodanie argumentów
2. Odjęcie polaryzacji (2^{n-1})

Nadmiar w operacjach arytmetycznych (overflow)

Wynik może wykroczyć poza zakres dla danej liczby bitów.
Układ wykonawczy powinien sygnalizować wystąpienie nadmiaru.
Sposób reagowania na nadmiar zależy od dostępnych zasobów
Można na etapie projektu zwiększyć długość słów.

Mnożenie

Wynik mnożenia liczby n-bitowej i m-bitowej zajmuje n+m bitów

Klasyczny algorytm
- mnożenie mnożnej przez każdy z bitów mnożnika
- zsumowanie iloczynów cząstkowych

Wygodniej jest sumować iloczyny cząstkowe w każdym kroku

Mnożenie sekwencyjne
- Rejestr mnożnej MN
- Rejestr wyniku RH RL
  - podwójna długość
  - RL zawiera mnożnik
- Sumator

Algorytm
1. Dodać do RH mnożną lub 0 zależnie od bitu mnożnika
2. Przesunąć R w prawo

liczba kroków równa liczbie bitów argumentów

Mnożenie w kodzie ZM
- moduł - iloczyn modułów w NKB
- znak - XOR znaków argumentów

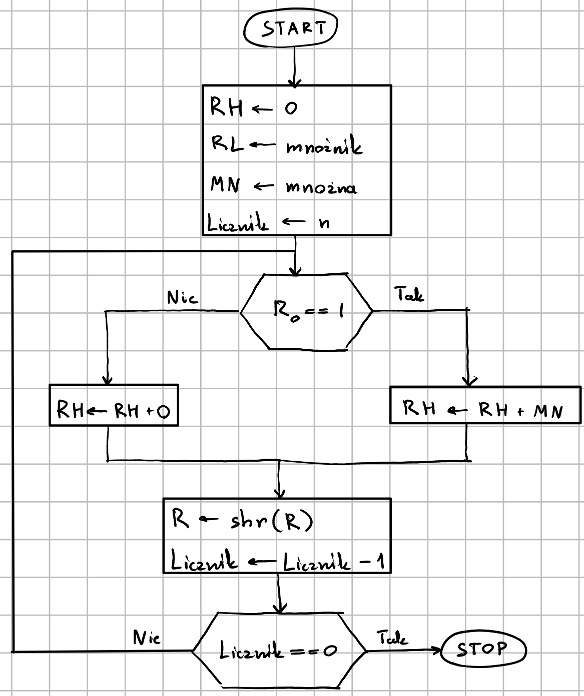
Mnożenie w kodzie U1 i U2
- dla dodatniego mnożnika tak samo jak NKB
- kroki korekcyjne dla ujemnego mnożnika

Kodowanie U2 jest najbardziej praktyczne
- prosta zamiana znaku
- proste dodawanie i odejmowanie
- prostsze mnożenie niż U1
- pojedyncza reprezentacja zera

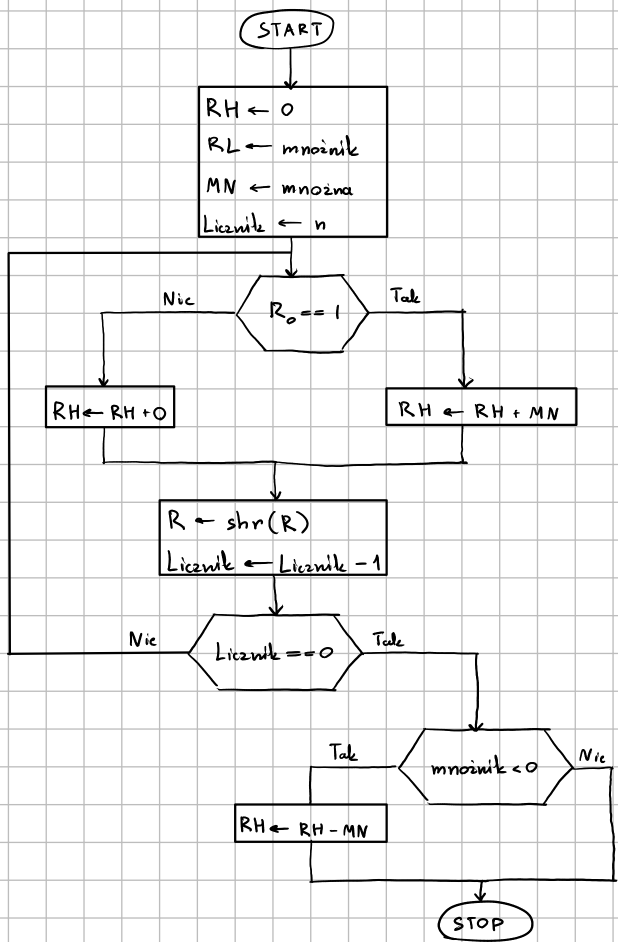
Algorytm Bootha
na mnożenie w kodzie U2

Wykorzystuje dodawanie i odejmowanie
W każdym kroku bierze pod uwagę 2 bity,
przez co unika kroku korekcyjnego

Mnożenie NKB



Mnożenie U2



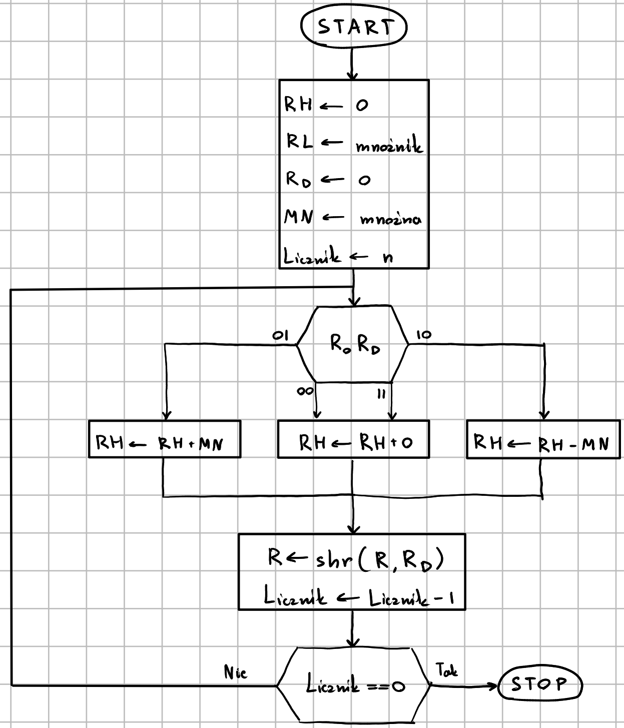
Dzielenie

Wynik dzielenia to iloraz i reszta
Iloraz zostaje w RL a reszta w RH

Przy dzieleniu liczby podwójnej długości przez liczbę pojedynczej długości iloraz i reszta są pojedynczej długości

A / B = I (R)
R = A - IB

Algorytm Bootha



Algorytm dzielenia w NKB