

Biblioteka arytmetyki liczb stałoprzecinkowych dowolnej precyzji z wykorzystaniem wewnętrznej reprezentacji U2.

Leszek Błażewski

Karol Noga

Semestr letni 2018/2019

1 Główne założenia projektu

Głównym założeniem projektowym jest implementacji biblioteki liczb w języku C o dowolnej precyzji, która wykorzystuje system U2 jako bazowy. Aby możliwe było osiągnięcie nieskończonej dokładności postanowiliśmy, że dane będą przechowywane w postaci tekstowej. W języku C reprezentowane są one przy pomocy dynamicznie alokowanej tablicy charów.

2 Konwersje

2.1 Opis

Zgodnie z założeniami projektu biblioteka musi operować na danych, których wartości wyrażone są w systemie U2. Liczby dodatnie nie posiadają znaku natomiast liczby ujemne przed konwersją reprezentowane są z znakiem minus. W zależności od wartości ulegającej konwersji liczba reprezentowana jest w systemie U2 z odpowiadającym jej znakiem.

Poniżej zamieszczono listę konwersji do systemu U2, które oferowane są przez bibliotekę:

- Konwersja z systemu 16
- Konwersja z systemu 8
- Konwersja z systemu binarnego
- Konwersja z systemu 10

2.2 Algorytm konwersji

W celu wyznaczenia wartości liczby zapisanej w systemie binarnym z znakiem, przechowywanej jako tekst posłużono się prostym algorytmem polegającym na skanowaniu liczby od tyłu w celu znalezienia pierwszej "1" występującej w liczbie oraz negacji wszystkich następnych bitów.

W skrócie algorytm można zapisać następująco:

1. Skanuj liczbę od tyłu spisując wszystkie "0" aż do momentu wystąpienia pierwszej "1".
2. Znaną liczbę również przepisz do wyniku.
3. Wszystkie kolejne bity liczby konwertowanej neguj.
4. Jeśli liczba nie ma znaku - na początku dopisz "0" z przodu liczby w przeciwnym wypadku dopisz "1".

Liczby zapisane w innych systemach najpierw konwertowane są na reprezentację binarną z znakiem, a następnie według powyższego algorytmu konwertowane są na sytem U2.

3 Podstawowe operacje arytmetyczne

Biblioteka oferuje następujące operacje arytmetyczne na danych reprezentowanych w systemie U2. Operacje te zostaną zaimplementowane przy użyciu instrukcji assemblerowych.

Lista podstawowych operacji artmetycznych:

- Dodawanie
- Odejmowanie
- Mnożenie
- Dzielenie
- Pierwiastkowanie
- Potęgowanie

4 Problemy jakie napotkaliśmy dotychczas

1. Czy reprezentacja w tablicach charów jest odpowiednia. Jeśli nie czy są jakieś lepsze alternatywy ?
2. Jak konwertować liczby z systemu dziesiętnego (trzymane w tablicy charów) na binarny, (część całkowita + ułamkowa), jak usprawnić ten algorytm ?
3. Czy założenie że precyzja liczby jest określana na podstawie wprowadzonej danej jest odpowiednia ?
4. Czy dokładność wyniku działania powinna być zależna od składnika działania który miał większą precyzję ?
5. Co zrobić w przypadku algorytmu pierwiastkowania gdy liczba ma okres który jest bardzo długi i trudno wykrywalny lub nie jest okresowa ?
6. Czy wykonując kod assemblerowy lepiej przekształcić tablice charów z ascii na faktyczne liczby czy zostawić i wykonywać operacje na kodach.