**УДК 681.03**

**А.Я. Горпенюк[1], Н.М. Лужецька[2]**
Національний університет "Львівська політехніка"
[1]кафедра захисту інформації,
[2]кафедра безпеки інформаційних технологій

# ГЕНЕРАТОР ПСЕВДОВИПАДКОВИХ ЧИСЕЛ НА ОБЧИСЛЮВАЧІ КОРЕНЯ КВАДРАТНОГО З ПРОСТОГО ЧИСЛА



**Подано результати синтезу та дослідження генератора псевдовипадкових чисел на базі обчислювача кореня квадратного з простого числа. Показано, що такий генератор має добрі статистичні характеристики. Запропонований алгоритм обчислення функції кореня квадратного забезпечує високу швидкодію генератора.**

**Ключові слова: генератор псевдовипадкових чисел, криптографія, квадратний корінь.**

**The results of pseudorandom numbers generator, based on the calculator square root of a prime number, synthesis and research are given. It is shown that this generator has qualitative statistical properties. The proposed algorithm for computing the square root function provides high speed generator.**

**Key words: generator of pseudorandom numbers, cryptography, square root.**

## Вступ

У математиці відома гіпотеза про нормальність ірраціональних та трансцендентних чисел, зокрема квадратних коренів з простих чисел [1]. Фактично, це означає, що послідовність цифр кореня з простого числа утворює псевдовипадкову послідовність. Згадана гіпотеза є недоведеною і належить до найвідоміших нерозв'язаних математичних задач. Інтерес до обчислення квадратного кореня з простого числа зародився дуже давно. Зокрема, у збірках вавилонських історичних цінностей, що зберігаються в Єльському університеті, є кругла глиняна табличка (рис. 1), що датується 1750 р. до н.е. На ній зображений поділений діагоналями квадрат і чіткими клинописними знаками виписано три цифри. Коли їх прочитали, стало зрозуміло, що майже 4000 років тому у Вавилоні уміли визначати діагональ квадрата за його стороною, перемножуючи її довжину на квадратний корінь з двійки. Помітки на табличці дають наближене значення $\sqrt{2}$ у чотирьох шістдесяткових цифрах, що відповідає 8 десятковим цифрам:

$$1+\frac{24}{60}+\frac{51}{60^2}+\frac{10}{60^3}=1,41421296\,.$$

Задача обчислення якомога більшої кількості розрядів кореня з простого числа, як і чисел $\pi$, $e$, актуальна і сьогодні. Зокрема, одним з відомих сучасних досягнень в цьому напрямі є праця співробітника Відділу математичних методів в інженерній справі при Колумбійському університеті професора Жака Дутки. Він розробив алгоритм і підрахував корінь з двійки до мільйон вісімдесят другого десяткового знака. На початку XXI ст. це була найдовша зі всіх обчислювальних величин за всю історію математики. Актуальність цієї обчислювальної задачі пояснюється як прагненням емпірично переконатися в псевдовипадковості послідовності цифр кореня, так і широкою областю застосування псевдовипадкових чисел, зокрема в криптографії.

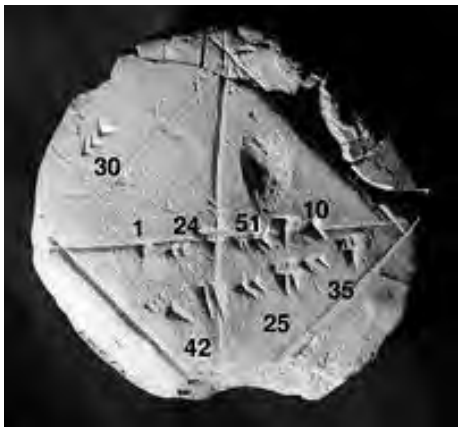


*Рис. 1. Вавилонська глиняна табличка*

Крім псевдовипадковості, послідовність цифр кореня з простого числа має ще одну важливу властивість, яка робить такі послідовності цінними з

погляду сучасної криптографії. Ця властивість полягає в тому, що послідовність цифр кореня з простого числа не має періоду. Тому такі послідовності можуть застосовуватися в потокових шифрах для шифрування великих обсягів інформації, зокрема зображень, відео- та аудіофайлів.

**Аналіз методів обчислення квадратного кореня з простого числа**

Для розрахунку квадратного кореня з простого числа в загальному випадку можуть застосовуватися різні методи, наприклад, метод обчислення квадратного кореня в стовпчик, розклад в ряд Тейлора, метод арифметичного добування квадратного кореня, метод грубої оцінки значення квадратного кореня, геометричний метод обчислення квадратного кореня, табличне обчислення. Можуть застосовуватися спеціалізовані функціональні перетворювачі, наприклад, число-імпульсні, побудовані за класичною чи конвеєрною [2] структурою. Може також застосовуватися ітераційна формула Герона й інші числові методи уточнення значення квадратного кореня. Однак для застосування в генераторах ПВЧ майже усі ці методи непридатні. Це зумовлено двома причинами. Перш за все, генератор ПВЧ повинен, як правило, генерувати довгу послідовність псевдовипадкових бітів. У випадку побудови такого генератора на обчислювачі кореня квадратного з простого числа це означає, що необхідно розрахувати велику кількість розрядів кореня. І навіть більше, кожен черговий розряд результату має бути обчислений точно. Інакше властивість псевдовипадковості згенерованої послідовності може бути втрачена. Крім того, розряди квадратного кореня, які стають бітами псевдовипадкової послідовності і можуть застосовуватися в швидких потокових шифрах, мають розраховуватися швидко.

Отже, для застосування в генераторі ПВЧ на обчислювачі квадратного кореня з простого числа придатний швидкий і точний обчислювач “цифра за цифрою”. Відповідно різноманітні методи грубої оцінки значення квадратного кореня придатні хіба що для розрахунку початкового наближення. Малопридатні також такі методи, як метод обчислення в стовпчик і більшість з наближених числових методів (Герона, хорд, дотичних та інші). Основна причина малопридатності цих методів для побудови генератора ПВЧ – висока складність обчислювальних операцій, яка зростає зі збільшенням кількості обчислених розрядів. Разом з тим зазначимо, що метод обчислення в стовпчик дає можливість знайти результати “цифра за цифрою”.

Зважаючи на згадані особливості сфер застосування, зручним для побудови обчислювача генератора ПВЧ є метод половинного ділення. В цьому методі застосовуються доволі прості обчислювальні операції. Він має повільну збіжність, однак є швидким і за певних умов може бути перетворений на метод “цифра за цифрою”. Щодо швидкодії, то для генератора ПВЧ важливішою є швидкість генерування точного значення чергового біта послідовності, ніж загальна швидкість генерування всієї послідовності без гарантії точності значень окремих бітів.

**Розроблення алгоритму обчислення кореня квадратного для генератора ПВЧ**

Сформулюємо стосовно методу половинного ділення спостереження, яке дає змогу перетворити цей метод на метод “цифра за цифрою” обчислення кореня квадратного.

Якщо ширина відрізка початкового наближення методу половинного ділення (відрізка, на якому локалізовано корінь нелінійного рівняння $x^2 = p$) дорівнює цілому степеню двійки, на кожному кроці методу ми отримуємо черговий правильний біт результату обчислення кореня.

Отже, якщо витримати сформульовані вимоги до початкового наближення кореня, зможемо побудувати алгоритм обчислення квадратного кореня методом “цифра за цифрою”.

Отже, загалом, метод половинного ділення можна застосувати для побудови генератора ПВЧ на обчислювачі кореня квадратного з простого числа. Порівняно з іншими методами обчислення квадратного кореня, метод половинного ділення характеризується меншою складністю обчислювальних операцій, часто поступаючись за швидкістю збіжності. Разом з тим, у методі половинного ділення можна забезпечити такий вибір початкового наближення, який дасть змогу обчислювати значення квадратного кореня з числа “біт за бітом”, отримуючи на кожному кроці алгоритму точне значення чергового біта результату, що критично важливо для генерування псевдовипадкової послідовності бітів.

Ще однією важливою характеристикою генератора ПВЧ є його швидкодія. Невисока складність основних операцій алгоритму половинного ділення здатна забезпечити таку характеристику генератора ПВЧ. Саме тому алгоритм половинного ділення часто рекомендується для обчислення квадратного кореня з довгих чисел. Разом з тим, якщо застосовується обчислювач квадратного кореня, для побудови генератора ПВЧ можна запропонувати ряд вдосконалень методу половинного ділення, які додатково знизять його трудомісткість. Суть вдосконалень така:

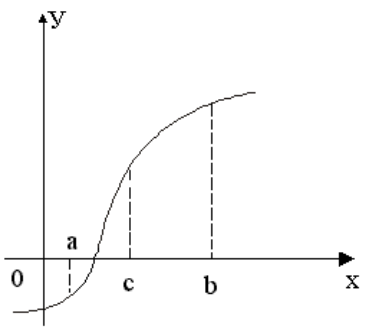


*Рис. 2. Ілюстрація методу половинного ділення*

1. Ширина відрізка початкового наближення має дорівнювати цілому степеню двійки. Це дає змогу отримувати точне значення чергового біта результату на кожному кроці алгоритму. Крім того, як показано далі, це дозволяє в алгоритмі половинного ділення відмовитися від контролю за правою межею відрізка локалізації кореня, а також виконувати обчислення середини відрізка простим дописуванням одиниці в черговий біт.

2. Під час генерування послідовності ПВЧ, яка у випадку застосування обчислювача квадратного кореня з простого числа є неперіодичною, можна почати генерацію з будь-якого біта, вибравши довільне початкове наближення відповідно до вимог пункту 1.

3. Спираючись на пункт 2, пропонуємо починати генерацію з першого дробового біта, прийнявши, що початкове наближення кореня, тобто точки a відрізка локалізації (рис. 2), дорівнює цілій частині результату обчислення кореня. При цьому відповідно до пункту 1 передбачається, що права межа відрізка локалізації кореня (точка b) віддалена від точки a на одиницю. Як показано нижче, це дає змогу істотно спростити обчислення умов локалізації кореня для наступного кроку алгоритму.

**Розроблення рівняння роботи обчислювача генератора ПВЧ**

Рівняння роботи генератора розробляємо, спираючись на метод половинного ділення, враховуючи сформульовані пропозиції щодо вдосконалення методу. Нам потрібно обчислити значення x квадратного кореня з простого числа $p$:

$$x = \sqrt{p} \tag{1}$$

Для цього ми застосовуємо метод половинного ділення, за допомогою якого уточнюємо початкове наближення кореня нелінійного квадратного рівняння:

$$x^2 - p = 0 \tag{2}$$

За початкове наближення $x_0$ (ліву межу відрізка локалізації кореня в методі половинного ділення) вибираємо цілу частину x – найбільше ціле число, квадрат якого менший за число $p$. Очевидно, що в точці початкового наближення значення функції (2) від'ємне. У сформульованих пропозиціях ми, крім початкового наближення $x_0$, запропонували ширину відрізка локалізації – 1. Отже, щоб за таких умов знайти середину відрізка локалізації, нам потрібно до початкового наближення $x_0$ додати половину відрізка локалізації – в нашому випадку $1/2$, тобто $2^{-1}$. Далі потрібно обчислити квадрат середньої точки і порівняти результат з $p$. Квадрат середньої точки, спираючись на початкове наближення і враховуючи прийняті припущення, визначимо за таким співвідношенням:

$$(x_c)^2 = y_c = \left(x_0 + 2^{-1}\right)^2 = x_0^2 + 2x_0 \cdot 2^{-1} + 2^{-2} = x_0^2 + x_0 + 2^{-2} = y_0 + x_0 + 2^{-2} \tag{3}$$

Далі порівняємо обчислене за (3) $y_c$ із $p$. Якщо $y_c < p$, приймаємо $x_1 = x_c; y_1 = y_c$. Якщо ні, $x_1 = x_0; y_1 = y_0$. Далі продовжуємо і на $i$-му кроці алгоритму отримаємо:

$$(x_c)^2 = y_c = \left(x_{i-1} + 2^{-i}\right)^2 = x_{i-1}^2 + 2^{-i+1}x_{i-1} + 2^{-2i} = y_{i-1} + 2^{-i+1}x_{i-1} + 2^{-2i} \tag{4}$$

(4) знову порівнюємо з $p$ і за результатом порівнянь визначаємо черговий біт результату, і так далі.

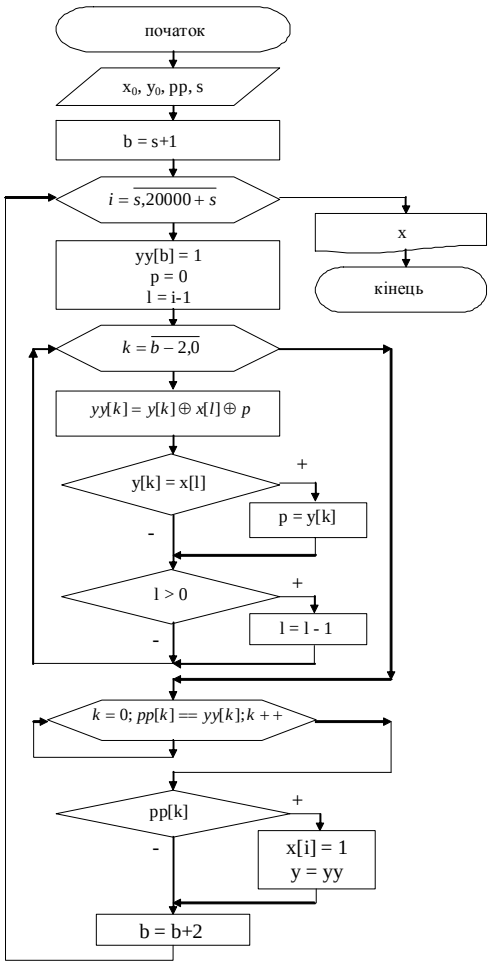


*Рис. 3. Алгоритм функціонування генератора ПВЧ на обчислювачі кореня з простого числа*

Аналізуючи вирази (3) і (4), доходимо висновку, що додавання доданка $2^{-2i}$ на практиці реалізується дописуванням одиниці у відповідний розряд (адже перед виконанням і-го кроку алгоритму всі молодші розряди числа у, починаючи з розряду із вагою $2^{-2i+2}$, – нульові). Доданок $2^{-i+1}x_{i-1}$ формуємо зсувом числа $x_{i-1}$. Отже, для обчислення нового $y_i$ за (4) необхідно виконати запис одиниці у черговий біт, зсув числа $x_{i-1}$ і його додавання до $y_{i-1}$. За складністю сукупність таких операцій близька до одного додавання.

Відтак виконується порівняння розрахованого числа з $p$ і за результатами порівняння встановлюються нові значення $x_i, y_i$. Причому нове значення $x_i$ встановлюється дописуванням нуля або одиниці (чергового *правильного!* біта) у черговий молодший розряд результату.

Отже, якщо не брати до уваги простих операцій запису біта, зсуву і переприсвоєння, запропонований алгоритм потребує всього однієї операції додавання на один правильний біт результату.

**Розроблення алгориту роботи генератора ПВЧ на обчислювачі кореня з простих чисел**

Спираючись на запропонований принцип роботи генератора і виведене рівняння роботи його обчислювача, розроблено алгоритм роботи генератора ПВЧ на основі обчислювача кореня з простих чисел. Структура розробленого алгоритму подана на рис. 3.

У структурі алгоритму на рис. 3 використано такі позначення:

**$x_0$, $y_0$** – початкові наближення результату обчислення x та його квадрата;

**pp** – задане просте число;

**s** – кількість бітів у цілій частині результату;

**yy** – проміжне значення квадрата результату – обчислюється відповідно до (4);

**p** – значення переносу в наступний біт при додаванні.

Розроблений алгоритм (рис. 3) передбачає генерування послідовності довжиною 20000 бітів з метою подальшого дослідження статистичних властивостей згенерованої послідовності за стандартом FIPS 140. Тому величини *x*, *y*, *yy*, які є довгими числами, в алгоритмі на рис. 3 мають вигляд масивів. Причому, враховуючи доданок $2^{-2i}$, який додається до *y* під час розрахунку *yy* відповідно до (4), а також необхідність зсуву *x* відповідно до (4), розмір цих масивів збільшено до 40000+2s.

У розробленому алгоритмі (рис. 3) в основному циклі генеруються 20000 бітів квадратного кореня із заданого простого числа *pp*. Для цього відповідно до (4) обчислюється квадрат середньої точки на поточному відрізку локалізації. Додавання доданка $2^{-2i}$ реалізується записом одиниці в біт *yy*[*b*]. Необхідний за (4) зсув числа *x* реалізується відповідною модифікацією індексу масиву *x*, після чого виконується побітове додавання *y* та зсунутого *x* із застосуванням переносу в наступний розряд. Після обчислення *yy*, відповідно до (4), виконується порівняння *yy* із заданим простим

числом. Якщо просте число більше, відповідно до (2) робиться висновок про від'ємність функції у поточній середній точці відрізка локалізації (за прийнятих умов функція у правій точці відрізка завжди додатна). В цьому випадку ліва межа відрізка локалізації повинна переміститися в середню точку – число $x$ збільшується на величину $2^{-i}$ (дописується одиниця у відповідний розряд), а проміжне уу стає новим у. Якщо ж проміжне уу на цій ітерації більше за просте число рр, ми залишаємося на старій лівій межі відрізка локалізації (вважаючи, що права межа перейшла в середню точку). В обох випадках приріст х для розрахунку наступного біта зменшується вдвічі і ми переходимо до наступної ітерації алгоритму.

**Порівняння розробленого алгоритму з базовим**

Аналізуючи запропонований алгоритм (рис. 3), доходимо висновку, що для генерування одного біта х необхідно виконати одне додавання, одне порівняння і, не завжди, одне переприсвоєння. Натомість в класичному алгоритмі половинного ділення у випадку нашої функції (2) для обчислення кожного біта результату необхідно виконати додавання і зсув для обчислення середньої точки відрізка локалізації, множення для отримання квадрата середньої точки, порівняння і два переприсвоєння.

Отже, запропонований алгоритм порівняно з класичним дає змогу зекономити одне множення на кожен біт результату. Зважаючи на те, що приблизна складність запропонованого алгоритму – одне додавання на один біт результату, такий виграш є істотним, особливо з огляду на довжину послідовностей бітів, які необхідно генерувати в генераторі ПВЧ.

На основі розробленого алгоритму (рис. 3) виконано програмну реалізацію генератора на обчислювачі квадратного кореня з простого числа. Виконана програмна реалізація передбачала також можливість дослідження статистичних властивостей генератора на відповідність вимогам стандарту FIPS 140.

У процесі дослідження статистичних властивостей генератора за допомогою його програмної моделі генерувалася послідовність псевдовипадкових бітів довжиною 20000 бітів. При цьому здійснювалось тестування згенерованої послідовності за стандартом FIPS 140. Результати тестування статистичних властивостей генератора для трьох простих чисел зведено в табл. 1. В таблиці тести серій одиниць подано у верхньому рядку, а тести серій нулів – у нижньому.

*Таблиця 1*

**Результати дослідження статистичних характеристик генератора**

| | Монобітний тест | Блоковий тест | Тест серій | | | | | | Тест довжини серії |
|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | |
| $\sqrt{3}$ | 10036 | 10 | 2433 | 1261 | 632 | 350 | 157 | 143 | 14 |
| | | | 2465 | 1259 | 610 | 336 | 172 | 134 | |
| $\sqrt{17}$ | 10014 | 18 | 2527 | 1240 | 614 | 310 | 163 | 158 | 12 |
| | | | 2526 | 1234 | 621 | 335 | 144 | 153 | |
| $\sqrt{31}$ | 10005 | 24 | 2522 | 1225 | 630 | 339 | 181 | 127 | 11 |
| | | | 2531 | 1215 | 679 | 316 | 132 | 150 | |

Як відомо, прийнятними показниками тестів відповідно до стандарту FIPS 140 є такі:
**монобітовий тест:** 9654 < *n1* (*n2*) < 10346; **блоковий тест :** 1,03< $X_3$ < 57,4;
**тест серій**

| Довжина серії | Необхідний інтервал |
|---|---|
| 1 | 2267 – 2733 |
| 2 | 1079 –1421 |
| 3 | 502 – 748 |
| 4 | 223 – 402 |
| 5 | 90 – 223 |
| 6 | 90 – 223 |

(зі збільшенням довжини серії на 1 кількість серій зменшується приблизно в 2 рази)

Тест довжин серій: максимальна довжина серії не повинна перевищувати значення 34.

Отже, досліджені статистичні характеристики розробленого генератора відповідають вимогам стандарту FIPS 140.

**Висновки**

У роботі запропоновано вдосконалення методу половинного ділення з метою його застосування в генераторі ПВЧ на обчислювачі квадратного кореня з простого числа. Розроблено алгоритм роботи і програмну реалізацію генератора ПВЧ на такому обчислювачі. Досліджено статистичні характеристики розробленого генератора. За результатами проведених досліджень показано, що застосування вдосконаленого методу половинного ділення дає змогу істотно покращити швидкодію методу за рахунок спрощення обчислень на одне множення протягом кожної ітерації. Стосовно побудованого на такому методі генератора ПВЧ це означає, що для обчислення одного біта псевдовипадкової послідовності виконується всього одна операція додавання. Отримані за результатами дослідження розробленого генератора оцінки його статистичних характеристик підтверджують, що якість згенерованих таким генератором псевдовипадкових послідовностей відповідає вимогам стандарту FIPS 140.

*1. Силкин Б.И. С корнем квадратным сквозь историю. – М., 1971. 2. Горпенюк, А.Я. Принципи побудови конвеєрних базових вузлів число-імпульсних вимірювальних перетворювачів [Текст, рисунки] / А.Я. Горпенюк // Контроль і управління в технічних системах (КУТС-97): книга за мат. конференції. Том 2. – Універсум-Вінниця. – 1997. – С. 137–140. 3. Горпенюк А.Я. Імітаційне моделювання конвеєрних число-імпульсних функціональних перетворювачів [Текст] / А.Я. Горпенюк, В.Б. Дудикевич, Н.М. Лужецька // Вісник Нац. ун-ту "Львівська політехніка" "Автоматика, вимірювання та керування". – 2005. – № 530. – С. 66–75.*