УДК 681.03

А.Я. Горпенюк , Н.М. Лужецька

Національний університет “Львівська політехніка”

кафедри “Захист інформації”, “Безпека 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.

Keywords: generator of pseudorandom numbers, cryptography, square root.

Вступ

В математиці відомою є гіпотеза про нормальність ірраціональних та трансцендентних чисел, зокрема квадратних коренів з простих чисел [1]. Фактично, це означає, що послідовність цифр кореня з простого числа складає псевдовипадкову послідовність. Згадана гіпотеза є недоведеною і відноситься до найбільш відомих невирішених математичних задач. Інтерес до обчислення квадратного кореня з простого числа зародився дуже давно. Зокрема у збірках Вавілонських історичних цінностей, що зберігаються в Єльському університеті, є кругла глиняна табличка (Рис.1), що відноситься до 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 тисячі років назад у Вавілоні уміли визначати діагональ квадрата за його стороною, перемножуючи її довжину на квадратний корінь з двійки. Помітки на табличці дають наближене значення $\sqrt{2}$ в 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 більшого числа розрядів кореня з простого числа, як і чисел π, e, актуальна і сьогодні. Зокрема одним з відомих сучасних досягнень в цьому напрямку є робота співробітника Відділу математичних методів в інженерній справі при Колумбійському університеті професора Жака Дутки. Він розробив алгоритм і підрахував величину кореня з двійки до мільйон

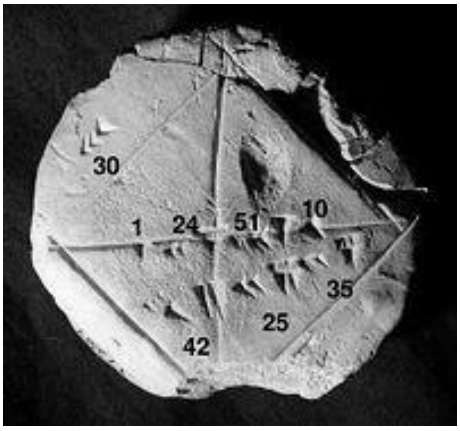

Рис.1. Вавилонська глиняна табличка

вісімдесят другого десяткового знаку. На початок 21 століття 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.

Крім псевдовипадковості, послідовність цифр кореня з простого числа має ще одну важливу властивість, яка робить такі послідовності цінними з точки зору 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 об'ємів інформації, зокрема зображень, відео- та аудіо файлів.

Аналіз методів обчислення квадратного кореня з простого числа

Для розрахунку квадратного кореня з простого числа в загальному випадку можуть застосовуватися різні методи. Може застосовуватися 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 і інші чисельні методи уточнення значення квадратного кореня. Однак для застосування в генераторах ПВЧ майже усі ці методи є непридатними. Це обумовлено 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. Більше того, 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.

Таким чином, 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. Мало придатні також такі методи, як метод обчислення в стовпчик і більшість з наближених чисельних методів (Герона, хорд, дотичних і інші). Основна причина малопридатності цих методів для побудови генератора ПВЧ – висока складність обчислювальних операцій, яка зростає із збільшенням кількості обчислених розрядів. Разом з тим відзначимо, що метод обчислення в стовпчик дає можливість обчислення результату "цифра за цифрою".

Зважаючи на згадані особливості області 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 важливішим 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 дозволяє перетворити цей метод в метод "цифра за цифрою" обчислення кореня квадратного.

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

Таким чином, якщо витримати сформульовані вимоги до початкового наближення кореня, ми зможемо побудувати алгоритм обчислення квадратного кореня методом "цифра за цифрою".

Отже в цілому, метод половинного ділення може бути застосований 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 швидкістю збіжності. Разом з тим в методі половинного ділення можна забезпечити такий вибір початкового наближення, який дасть можливість обчислювати значення квадратного кореня з числа “біт за бітом”, отримуючи на кожному кроці алгоритму точне значення чергового біту результату, що критично важливо при генеруванні псевдовипадкової послідовності бітів.

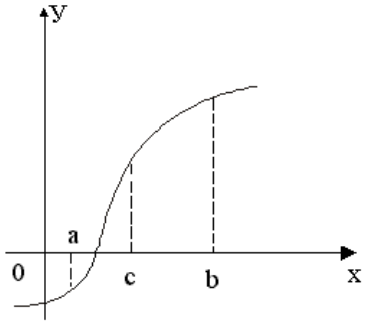

Рис.2. Ілюстрація методу половинного ділення

Ще одною 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 у випадку застосування обчислювача квадратного корення для побудови генератора ПВЧ можна запропонувати ряд вдосконалень методу половинного діленя, які дозволять додатково знизити його трудоємність. Суть вдосконалень полягає в наступному.

1. Ширина відрізку початкового наближення має дорівнювати цілій степені двійки. Це дозволяє отримувати точне значення чергового біта результату на кожному кроці алгоритму. Крім того, як буде показано далі, це дозволяє в алгоритмі половинного ділення відмовитися від контролю за правою межею відрізку локалізації кореня, а також виконувати обчислення середини відрізку простим дописуванням одиниці в черговий біт.
2. При генеруванні послідовності ПВЧ, яка у випадку застосування обчислювача квадратного кореня з простого числа є неперіодичною, ми можемо почати генерацію з будь-якого біта, вибравши довільне початкове наближення відповідно до вимог пункту 1.
3. Спираючись на пункт 2 пропонується починати генерацію з першого дробового біта, забезпечивши початкове наближення кореня, тобто точки a відрізку локалізації (Рис.2), рівним цілій частині результату обчисленя кореня. При цьому відповідно до пункту 1 передбачається, що права межа відрізку локалізації кореня (точка b) віддалена від точки a на одиницю. Як буде показано нижче, це дозволяє 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) є від’ємним. В сформульованих пропозиціях ми, крім початкового наближення x_0 запропонували ширину відрізку локалізації – одиниця. 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 визначається таким співвідношенням:

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

Далі ми порівнюємо 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 ми продовжуємо і на і-ому кроці алгоритму отримаємо:

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

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

Аналізуючи вирази (3) і (4) приходимо до висновку, що додавання доданку 2^{-2i} на практиці реалізується дописуванням одиниці у відповідний розряд (адже перед виконанням і-ого 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}. Таким чином, для обчислення нового y_i за (4) нам 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.

Далі 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 *вірного!* біта) у черговий молодший розряд результату.

Отже, якщо не брати до уваги простих операцій запису біту, зсуву і переприсвоєння, запропонований алгоритм вимагає всього одної операції додавання на один вірний біт результату.

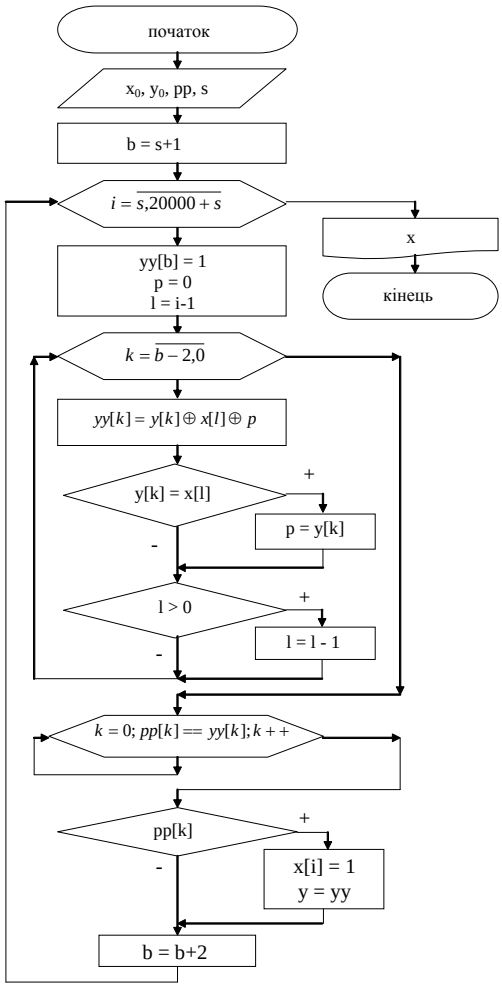

Рис.3. Алгоритм функціонування генератора ПВЧ на обчислювачі кореня з простого числа

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 базі обчислювача кореня з простих чисел. Структура розробленого алгоритму подана на Рис.3.

В структурі алгоритму на Рис.3 застосовано такі позначення:

x_0, y_0 – початкові наближення результату обчислення x та його квадрату;
pp – задане просте число;
s – кількість біт в цілій частині результату;
yy – проміжне значення квадрату результату - обчислюється відповідно до (4);
p – значення переносу в наступний біт при додаванні;

Розроблений алгоритм (Рис.3) передбачає генерування послідовності довжиною 20000 біт 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 при розрахунку yy відповідно до (4), а також враховуючи необхідність зсуву x відповідно до (4), розмір цих масивів збільшено до 40000+2s.

В розробленому алгоритмі (Рис.3) в основному циклі генеруються 20000 біт квадратного кореня із заданого простого числа *pp*. Для цього відповідно до (4) обчислюється квадрат середньої точки на біжучому відрізку локалізації. При цьому додавання доданка 2^{-2i} реалізується записом одиниці в біт yy[b]. Необхідний за (4) зсув числа x реалізується відповідною модифікацією індекса 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 в біжучій середній точці відрізка локалізації (за прийнятих умов функція в правій точці відрізку завжди додатна). В цьому випадку ліва межа відрізку локалізації повинна перемістится в середню точку – число x збільшується на величину 2^{-i} (дописується одиниця у відповідний розряд), а проміжне yy стає новим y. Якщо-ж проміжне yy на даній ітерації більше за просте число pp, ми залишаємося на старій лівій межі відрізку локалізації (вважаючи, що права межа перейшла в середню точку). В обох випадках приріст x для розрахунку наступного біта зменшується вдвічі і ми переходимо до наступної ітерації алгоритму.

Порівняння розробленого алгоритму з базовим

Аналізуючи запропонований алгоритм (Рис.3) приходимо до висновку, що для генерування одного біта x нам 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 біту результату необхідно виконати додавання і зсув для обчислення середньої точки відрізку локалізації, множення для отримання квадрату середньої точки, порівняння і два переприсвоєння.

Таким чином, запропонований алгоритм порівняно з класичним дозволяє зекономити одне множення на кожен біт результату. Зважаючи на те, що приблизна складність запропонованого алгоритму складає одне додавання на один біт результату, такий виграш є істотним, особливо з огляду на довжини послідовностей біт, які необхідно генерувати в генераторі ПВЧ.

На основі розробленого алгоритму (Рис.3) було 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.

В процесі дослідження статистичних властивостей генератора, за допомогою його програмної моделі генерувалася послідовність псевдовипадкових бітів довжиною 20000 біт. При цьому здійснювалось тестування згенерованої послідовності за стандартом FIPS 140. Результати тестування статистичних властивостей генератора для трьох простих чисел зведено в Таблиці 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

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

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

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

Висновки

В 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 дозволяє 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] / Горпенюк А.Я. // : "Контроль і управління в технічних системах" (КУТС-97). Книга за матеріалами конференції: Том2.-: "Універсум-Вінниця". - 1997. - С. 137-140. 3. Горпенюк, А. Я. Імітаційне моделювання конвеєрних число-імпульсних функціональних перетворювачів [Текст] / Горпенюк А.Я., Дудикевич В.Б., Лужецька Н.М. // Вісник НУ "Львівська політехніка" – "Автоматика, вимірювання та керування". - 2005. - №530. - С. 66-75.