МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АЕРОНАВІГАЦІЇ, ЕЛЕКТРОНІКИ ТА ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА ЕЛЕКТРОНІКИ, РОБОТОТЕХНІКИ І ТЕХНОЛОГІЙ
МОНІТОРИНГУ ТА ІНТЕРНЕТУ РЕЧЕЙ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач випускової кафедри
_____ Шутко В.М.
«____»_____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА
ЗІ СПЕЦІАЛЬНОСТІ 171 «ЕЛЕКТРОНІКА»
ОСВІТНЬО-ПРОФЕСІЙНОЇ ПРОГРАМИ «ЕЛЕКТРОННІ СИСТЕМИ»

Тема: «Апаратно-програмний модуль поточного шифрування інформації**»**
Виконавець

студент групи ЕС-238М _____ Ковальчук Арсен Віталійович

Керівник
к.т.н., доцент _____ Сініцин Рустем Борисович

Консультант розділу
«Охорона Праці» _____ Козлітін Олексій Олександрович

Консультант розділу
«Охорона навколишнього
середовища» _____ Радомська Маргарита Мирославівна

Нормоконтролер _____ Сініцин Рустем Борисович

КИЇВ 2022

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет аеронавігації, електроніки і телекомунікацій

Кафедра електроніки, робототехніки і технологій моніторингу та інтернету речей

Напрям (спеціальність) 171 «Електроніка»

(шифр, найменування)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Шутко В.М.

«____»_____ 2022 р.

ЗАВДАННЯ

на виконання дипломної роботи

Ковальчуку Арсену Віталійовичу

(прізвище, ім'я, по батькові випускника в родовому відмінку)

1. Тема дипломної роботи: «Апаратно-програмний модуль поточного шифрування інформації».

затверджена наказом ректора від «09» вересня 2022р. №1351/ст.

2. Термін виконання роботи : з «09» вересня 2022р. по «17» листопада 2022р.

3. Вихідні дані до роботи: створення сучасного алгоритму шифрування, апаратної та програмної реалізації потокового шифрування інформації на основі узагальнених матриць Галуа.

4. Зміст пояснювальної записки: теоретичний опис способу шифрування та роботи технологічних пристроїв, розробка програмного і апаратного рішення, результати аналізу та тестування.

5. Перелік обов’язкового ілюстративного матеріалу: ПВП в конфігурації Галуа, Побудова структурної схеми генератора Галуа четвертого порядку, Структурна схема генератора Фібоначчі, Схема взаємозв'язку елементів повної множини класичних матриць Галуа, Структурна схема «прямого» узагальненого базового генератора Галуа, Схема взаємозв’язку повної множини узагальнених матриць

Галуа, Реперна сітка алгоритму синтезу НП, Шифр XOR, Схема Галуа-64, Послідовність станів генератора ПВП, Рекурентні обчислення станів класичних Галуа-генераторів ПВП, Взаємозв'язок матриць Галуа, Етапи обрахунку ПВП.

6. Календарний план-графік

№ пор.	Завдання	Термін виконання	Підпис керівника
1	Написати заяву кваліфікаційної роботи	09.09.2022	
2	Ознайомитися та обґрунтувати актуальність обраної теми	10.09.2022 - 21.09.2022	
3	Сформулювати мету. Сформулювати завдання кваліфікаційної роботи	22.09.2022 - 24.09.2022	
4	Ознайомитися з загальними положеннями про дипломне проектування та оформлення	24.09.2022 - 25.09.2022	
5	Здійснити бібліографічний пошук	26.09.2022 - 27.09.2022	
6	Написати вступ та загальні відомості	28.09.2022 - 06.10.2022	
7	Написати 2-й та 3-й розділ кваліфікаційної роботи	07.10.2022 - 16.10.2022	
8	Створення інтерфейсу програми	17.10.2022 - 29.10.2022	
9	Написати програмного забезпечення	20.10.2022 - 24.10.2022	
10	Реалізація Галуа - 64	25.10.2022- 29.10.2022	
11	Апаратна реалізація Arduino	30.10.2022 - 04.11.2022	
12	Реалізація RSA	05.11.2022 - 11.11.2022	
14	Усунути недоліки.	12.11.2022- 16.11.2022	
15	Подати кваліфікаційну роботу на кафедру	17.11.2022	

7. Консультація з окремого(мих) розділу(ів):

Назва розділу	Консультант (посада, П.І.Б.)	Дата, підпис	
		Завдання видав	Завдання прийняв
Загальні відомості	к.т.н., доцент Сініцин Р. Б.		

8. Дата видачі завдання: «09» вересня 2022р.

Керівник дипломної роботи (проекту)_____Сініцин Р. Б

(підпис керівника) П.І.Б.)

Завдання прийняв до виконання_____ Ковальчук А. В.

(підпис випускника) (П.І.Б.)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Апаратно-програмний модуль поточного шифрування інформації»: 12 с., 13 с., 25 с., 28 с., 36 с., 60 с., літературних джерела.

Об’єкт дослідження: апаратний модуль і потокове шифрування інформації на основі узагальнених матриць Галуа.

Мета роботи: розробити апаратно-програмний модуль для забезпечення шифрування каналу зв’язку, дослідити різновиди матриць Галуа та створити алгоритм шифрування інформації на основі узагальнених матриць Галуа.

Методи дослідження: обробка літературних джерел, абстрактно-логічний, порівняльний аналіз, проведення обрахунків.

Результати магістерської роботи рекомендується використовувати під час проведення лекцій, наукових досліджень та в практичній діяльності для шифрування даних.

В першому розділі описано фундаментальні основи теорії криптографії.

У другому розділі описано класичні матриці Галуа і Фібоначчі та їх алгоритм синтезу, класичні генератори Галуа і Фібоначчі, ефективний алгоритм обчислення класичних генераторів ПВП.

У третьому розділі описано алгоритм синтезу узагальнених матриць Галуа, алгоритм синтезу незвідних поліномів лінійної складності, другий варіант захисту він атаки Берлекемпа - Мессі, принцип роботи шифратора Галуа – 64.

У четвертому розділі описано апаратну реалізацію на платформі Arduino Uno

МАТРИЦІ ГАЛУА, УЗАГАЛЬНЕНІ МАТРИЦІ ГАЛУА, НЕЗВІДНІ ПОЛІНОМИ, ЗАХИСТ ІНФОРМАЦІЇ, КРИПТОГРАФІЯ.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ

ПВП – Псевдовипадкова послідовність;

ЛРЗ – лінійний регістр зсуву;

РЗЛЗЗ – регістр зсуву з лінійними зворотними зв'язками ;

НП – незвідні поліноми;

ПрП – примітивні поліноми;

ПНП прості незвідні поліноми;

РЗ – регістр зсуву;

КМГ – класичні матриці Галуа;

УЕ – утворюючий елемент;

УМГ – узагальнені матриці Галуа;

НСК – найменше спільне кратне

НСД – найбільший спільний дільник

WPF – Windows Presentation Foundation

IDE - Integrated development environment

ВСТУП

Метою даної роботи є створення апаратно програмної реалізації потокового байт орієнтованого алгоритму шифрування інформації на основі узагальнених матриць Галуа. Практична значимість даної роботи полягає у забезпеченні захисту каналів зв’язку.

Процес перетворення звичайного і зрозумілого тексту в нерозбірливий і навпаки називається кодуванням, а наука яка цим займається - криптографією. За допомогою цієї галузі дані зберігаються та передаються у спеціальній формі, щоб їх могли зрозуміти та обробки лише довірені персони. Важливість криптографії полягає в тому, що вона захищає дані від злому та зміни, одночасно роблячи їх корисними для автентифікації користувачів. Пилип Циммерман визначає криптографію як "науку про використання математики для шифрування та дешифрування даних". Брюс Шнайер говорить: "Криптографія - це мистецтво і наука про збереження повідомлень у безпеці".

Люди об'єднувалися в племена, королівства і групи в міру розвитку цивілізацій, що призвело до появи думок про політику і панування. Це підживлювало потребу в таємному спілкуванні один з одним, що в сучасному розумінні називається криптографією. Використання "ієрогліфів" та "шифру зсуву Цезаря" було одне з перших спроб людини зашифрувати інформацію. В перший половині XX століття більша частина криптографічних робіт проводилася у військових цілях, з метою приховування секретної інформації. Основною метою криптографії в історичні часи було не приховати повідомлення, а лише змінити форму “таємного листа”, що він містив у собі трансформацію початкового тексту. Таким чином в світових цивілізаціях, таких як: давньоєгипетська, вавилонська, греко-римська та візантійська вже був покладений фундамент криптографії. Клод Шеннон, відомий як “батько математичної криптографії”, оскільки він опублікував статтю під назвою "Математична теорія криптографії", яка була опублікована в 1949 році. Його робота надихнула подальші дослідження криптографії, а його праці були згадані як такі, що мали великий вплив на різних криптографів світу. Так почалася

сучасна криптографія, з появою стандарту шифрування, розробкою відкритого ключа, хешуванням, криптографічної політики та сучасного криптоаналізу.

До цього часу криптографія займалася виключно забезпеченням конфіденційності повідомлень - перетворенням повідомлень із зрозумілої форми в незрозумілу і зворотне відновлення на стороні одержувача, роблячи його неможливим для прочитання для того, хто перехопив або підслухав без секретного знання (а саме ключа, необхідного для дешифрування повідомлення). В останні десятиліття сфера застосування криптографії розширилася і включає не лише таємну передачу повідомлень, але і методи перевірки цілісності повідомлень, довготривале збереження інформації на різних системах збереження, ідентифікування відправника/одержувача (аутентифікація), цифрові підписи, інтерактивні підтвердження.

В теорії та практиці криптографічного захисту інформації однією з ключових проблем є побудова генераторів псевдовипадкових послідовностей (ГПВП) максимальної довжини (періоду) з прийнятними статистичними властивостями, які зазвичай реалізуються регістрами лінійного зсуву (РЛЗ) з лінійним зворотним зв'язком (ЛЗЗ) в конфігурації (залежно від схеми) Галуа або Фібоначчі. Структура і логічні схеми класичних генераторів РЛФП ШІМ однозначно визначаються формуючими поліномами, за допомогою яких формуються одноконтурні зворотні зв'язки в ЛРЗ. Відомо, що відповідний поліном зворотного зв'язку повинен бути примітивним, щоб регістр зсуву міг бути регістром максимального періоду.

Генерація псевдовипадкових послідовностей є надзвичайно важливою для криптографічного захисту інформації, оскільки від якості згенерованої послідовності залежить надійність та цілісність зашифрованої інформації.

Генератори та їх реалізації повинні відповідати ряду вимог: Вихідні значення повинні бути рівномірно розподілені, непередбачувані та максимально тривалі. Періодом генератора в криптографії називають кількість чисел, які не повторюються при незмінних комбінаціях вхідних значень. Алгоритм генерації ключів повинен мати якомога більшу кількість ненульових внутрішніх станів.

РОЗДІЛ 1
ТЕОРЕТИЧНІ ОСНОВИ РОЗРОБКИ

1.1 Дослідження шифрування

Шифрування - це процес, коли інформація кодується для запобігання несанкціонованому доступу, щоб тільки довірені користувачі мали доступ до зашифрованої інформації. Більшість користувачів не знають, що значна частина інформації вже захищена технологією шифрування. Наприклад, інтернет-магазини та онлайн-банкінг не працювали б без хорошого алгоритму шифрування. Для захисту грошей та персональних даних використовується шифрування. У корпоративному середовищі шифрування має використовуватися для захисту інтелектуальної власності та інноваційних розробок компанії, а також інших конфіденційних даних. Ще одним завданням криптографії є розробка алгоритмів завадостійкого зв'язку.

Генерація псевдовипадкових послідовностей є надзвичайно важливою для криптографічного захисту інформації, оскільки якість згенерованої послідовності визначає надійність та цілісність зашифрованої інформації.

Генератори та їх реалізація повинні відповідати ряду вимог: Вихідні значення повинні бути рівномірно розподіленими і непередбачуваними та мати якомога більший період. Періодом генератора в криптографії називають кількість чисел, які не повторюються при незмінних комбінаціях вхідних значень.

1.2 Мета шифрування

Сьогодні шифрування використовується не тільки для передачі конфіденційної інформації, а й для зберігання важливих даних у ненадійних джерелах та передачі їх незахищеними каналами зв'язку. Така передача даних складається з двох взаємно зворотних процесів:

- Шифрування даних перед їх передачею або зберіганням по каналах зв'язку.
- Розшифрування та відновлення вже зашифрованих вихідних даних без втрати інформації.

1.3 Шифр. Ключ. Види ключів

Шифр - сукупність алгоритмів криптографічних перетворень (алгоритмів шифрування), які відображають множину можливих відкритих даних у множину можливих зашифрованих даних, та їх зворотних перетворень. Таким чином, безпека всіх даних, зашифрованих за допомогою криптографічних алгоритмів шифрування, безпосередньо залежить від якості ключів криптографічного шифрування та складності алгоритму шифрування, а важливим параметром будь-якого шифру є ключ.

Ключ - параметр криптографічного алгоритму, який дозволяє вибрати перетворення з множини можливих перетворень для цього алгоритму. У сучасній криптографії прийнято вважати, що вся таємниця криптографічного алгоритму полягає в ключі, а не в деталях самого алгоритму. Ключ - це секретна інформація, яка використовується криптографічним алгоритмом при шифруванні або дешифруванні інформації. При використанні одного і того ж алгоритму результат шифрування залежить від ключа. Завдяки сучасним стійким криптографічним алгоритмам втрата ключа практично унеможливлює дешифрування інформації. Криптографічний ключ - це рядок бітів, який використовується криптографічними алгоритмами для шифрування та дешифрування даних. Відповідно, криптографічний ключ використовується для виконання криптографічних операцій і є видом вхідних даних для алгоритмів шифрування, що використовуються в криптографічних системах.

Залежно від використовуваного алгоритму шифрування криптографічні ключі поділяються на симетричні та асиметричні. У симетричних алгоритмах шифрування і дешифрування даних відбувається за допомогою одного і того ж ключа. В асиметричних алгоритмах для криптографічного прямого і зворотного перетворення використовуються різні ключі. Пари відкритого та закритого ключів називаються ключовими парами. При цьому особистий ключ відомий лише його власнику і зберігається в таємниці. В результаті обчислення функції від закритого ключа визначається відкритий ключ, який публікується в сертифікаті його власника і відомий всім. Обсяг інформації, що зберігається в ключі, визначається довжиною

ключа, яка зазвичай вимірюється в бітах. Чим більша довжина ключа, тим надійніший алгоритм шифрування і вищий рівень захисту.

1.4 Поліном

Многочленом називається многочлен, який складається з алгебраїчної суми кількох одночленів, кожний з яких є членом многочлена. Многочлени, що складаються з одного члена, також вважаються одночленами. Многочленом називається многочлен виду:

$$a_0x^n + a_1x^{n-1} + ... + a_{n-1}x + a_n, \qquad (1.1)$$

де a_0, a_1,…, a_n – елементи поля, x - змінна, n – коефіцієнт многочлена, a_0 - старший коефіцієнт.

1.5 Незвідні поліноми

У теорії полів Галуа, яка лежить в основі алгебри завадостійкого кодування, криптографії та проектування сучасних електронних систем передачі інформації, ключовим є поняття незвідного полінома (НЗ).

Незвідні многочлени (НЧ) мають сильну схожість з простими числами, які мають лише тривіальні дільники, тобто коли вони не діляться на многочлен нижчого степеня в заданій області. НП мають два позначення, перше - це так звана "поліноміальна форма", яку ми будемо називати алгебраїчною формою: $f(x) = \sum_{k=0}^{n} a_k x^k = a_n x^n + a_{n-1}x^{n-1} + ... + a_k x^k + ... + a_1 x + a_0$, *(1.2)*

а в якості другої служить *векторна форма*, що є сукупністю коефіцієнтів a_k полінома, включаючи нульові коефіцієнти відсутніх мономів ряду:

$$f = a_n\, a_{n-1} ... a_k ... a_1\, a_0. \qquad (1.3)$$

1.6 Примітивні поліноми

Поліноми поділяються на примітивні поліноми (ПрП) і поліноми, які не є примітивними. Останні для зручності будемо називати простими незвідними поліномами (ПНП). Підмножини $Q_{ПрП}$ і $Q_{ПНП}$ — непересічні підмножини повної множини Q поліномів одного і того ж ступеня n, тобто $Q = Q_{ПрП} \cup Q_{ПНП}$, причому $Q_{ПрП} \cap Q_{ПНП} = 0$. Якщо немає необхідності в конкретизації: чи є поліном ПрП або ПНП, для простоти будемо називати його незвідним поліномом. Кожен примітивний

поліном обов’язково має бути незвідним, тоді як незвідний не завжди має бути примітивним.

Існують різні варіанти визначення поняття «примітивного полінома». В алгебрі, теорії чисел і полів Галуа незвідним поліном f_n степені *n* називається примітивним над *GF(p)*, *p* – обрана система числення в тому і тільки в тому випадку, якщо він не дорівнює нулю і його порядок $Ord(f_n) = p^n - 1$. В криптографії примітивним вважається такий незвідний поліном $f_n(x)$, який ділить без залишку двочлен $e^n - 1$, за умови, що мінімальне значення *e* задане співвідношенням $e = p^n - 1$.

Поліноми характеризуються рядом основних параметрів. Одним з таких є *ступінь полінома (deg)* - це максимальний ступінь, який входить в поліном монома з ненульовим коефіцієнтом. *Порядок полінома (ord)* - це таке найменше натуральне число *m*, при якому даний поліном *f(x)* ділить без залишку двочлен $x^m - 1$.

1.7 RSA

RSA (Rivest-Shamir-Adleman) - криптосистема з відкритим ключем, яка широко використовується для безпечної передачі даних. Вона також є однією з найстаріших. Абревіатура "RSA" походить від прізвищ Рона Рівеста, Аді Шаміра та Леонарда Адлемана, які публічно описали алгоритм у 1977 році. Аналогічна система була таємно розроблена в 1973 році в GCHQ (британському агентстві радіотехнічної розвідки) англійським математиком Кліффордом Коксом (Clifford Cocks). Ця система була розсекречена в 1997 році.

У криптосистемі з відкритим ключем ключ шифрування є відкритим і відрізняється від ключа дешифрування, який тримається в таємниці (приватним). Користувач RSA створює та публікує відкритий ключ на основі двох великих простих чисел, а також допоміжного значення. Прості числа тримаються в секреті. Повідомлення можуть бути зашифровані будь-ким за допомогою відкритого ключа, але можуть бути розшифровані тільки тим, хто знає прості числа.

Безпека RSA ґрунтується на практичній складності піднесення до степеня добутку двох великих простих чисел, "проблемі піднесення до степеня". Злам

шифрування RSA відомий як проблема RSA. Чи є вона такою ж складною, як і проблема факторингу, залишається відкритим питанням. Не існує опублікованих методів перемоги над системою, якщо використовується досить великий ключ.

RSA є відносно повільним алгоритмом. Через це він не часто використовується для безпосереднього шифрування даних користувача. Частіше RSA використовується для передачі загальних ключів для криптографії з симетричним ключем, які потім використовуються для масового шифрування-розшифрування.

Алгоритм RSA складається з чотирьох етапів: генерація ключів, розподіл ключів, шифрування та дешифрування.

Основним принципом RSA є спостереження про те, що практично можливо знайти три дуже великих натуральних числа e, d і n, таких, що при піднесенні до степеня за модулем для всіх цілих чисел m (при $0 \leq m < n$):

$$(m^e)^d \equiv m \ (mod\ n) \qquad (1.4)$$

і що, знаючи e і n, або навіть m, буває надзвичайно важко знайти d. Потрійна риска (≡) тут означає модулярну конгруентність, тобто при діленні $(m^e)^d$ на n і m діленні на n вони обидва дають однакову остачу.

Крім того, для деяких операцій зручно, що порядок двох піднесень до степеня можна змінювати і що це співвідношення також має на увазі

$$(m^d)^e \equiv m \ (mod\ n). \qquad (1.5)$$

RSA включає в себе відкритий і закритий ключі. Відкритий ключ може бути відомий кожному і використовується для шифрування повідомлень. Мета полягає в тому, що повідомлення, зашифровані за допомогою відкритого ключа, можуть бути розшифровані за розумний проміжок часу тільки за допомогою закритого ключа.

Відкритий ключ представлений цілими числами n та e, а закритий ключ - цілим числом d (хоча число n також використовується в процесі розшифрування, тому його можна вважати частиною закритого ключа). m представляє повідомлення (заздалегідь підготовлене за певними правилами).

Відкритий ключ складається з модуля n та відкритого показника (або показника шифрування) e. Закритий ключ складається з закритого показника (або показника розшифрування) d, який повинен зберігатися в таємниці. p, q та λ(n) також повинні зберігатися в таємниці, оскільки вони можуть бути використані для обчислення d. Насправді, всі вони можуть бути відкинуті після обчислення d.

В оригінальній статті RSA,[5] замість λ(n) для обчислення приватного показника d використовується функція Ейлера тотієнт:

$$\varphi(n) = (p - 1)(q - 1). \qquad (1.6)$$

Оскільки φ(n) завжди ділиться на λ(n), алгоритм також працює. Можливість використання тотієнтної функції Ейлера випливає також з теореми Лагранжа, застосованої до мультиплікативної групи цілих чисел за модулем pq. Таким чином, будь-яке d, що задовольняє $d \cdot e \equiv 1 \pmod{\varphi(n)}$, також задовольняє $d \cdot e \equiv 1 \pmod{\lambda(n)}$. Однак, обчислення d за модулем φ(n) іноді дає результат, який є більшим, ніж потрібно (тобто d > λ(n)). Більшість реалізацій RSA приймають показники, згенеровані будь-яким з методів (якщо вони взагалі використовують приватний показник d, а не оптимізований метод розшифровки, заснований на китайській теоремі про залишок) але деякі стандарти, такі як FIPS 186-4, можуть вимагати, щоб $d < \lambda(n)$. Будь-які "великі" приватні експоненти, що не відповідають цьому критерію, завжди можуть бути зменшені за модулем λ(n) для отримання меншої еквівалентної експоненти.

Оскільки будь-які спільні множники $(p-1)$ та $(q-1)$ присутні у факторизації

$$n - 1 = pq - 1 = (p - 1)(q - 1) + (p - 1) + (q - 1), \qquad (1,7)$$

рекомендується, щоб (p - 1) та (q - 1) мали лише дуже малі спільні множники, якщо такі є, окрім необхідних 2.

Автори оригінальної статті про RSA здійснюють генерацію ключа, вибираючи d і потім обчислюючи e як модулярну мультиплікативну величину, обернену до d по модулю $\varphi(n)$, тоді як більшість сучасних реалізацій RSA, таких як ті, що слідують за PKCS#1, роблять навпаки (вибирають e і обчислюють d). Оскільки обраний ключ

може бути малим, тоді як обчислений ключ зазвичай не є таким, алгоритм RSA оптимізує розшифрування порівняно з шифруванням, тоді як сучасний алгоритм оптимізує шифрування.

1.9 Факторизація

У теорії чисел цілочисельна факторизація - це розкладання складеного числа на добуток менших цілих чисел. Якщо ці множники додатково обмежуються простими числами, процес називається факторизацією простими числами. Рис.1.1

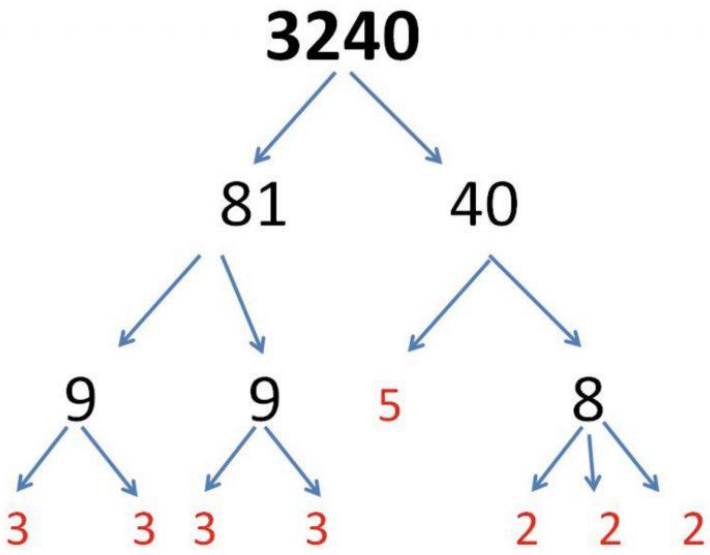

Рис. 1.1 «Факторизація»

Коли числа досить великі, не відомо жодного ефективного неквантового алгоритму факторизації цілих чисел. Однак не доведено, що такого алгоритму не існує. Передбачувана складність цієї проблеми важлива для алгоритмів, що використовуються в криптографії, таких як шифрування з відкритим ключем RSA та цифровий підпис RSA.[4] Багато областей математики та інформатики були залучені до вирішення проблеми, включаючи еліптичні криві, алгебраїчну теорію чисел та квантові обчислення.

Не всі числа заданої довжини однаково важко обчислити. Найскладнішими прикладами цих проблем (для відомих на сьогоднішній день методів) є напівпрості числа, тобто добуток двох простих чисел. Коли вони обидва великі, наприклад, більше двох тисяч біт, випадково вибрані і приблизно однакового розміру (але не дуже близькі, наприклад, щоб уникнути ефективної факторизації методом Ферма),

навіть найшвидші алгоритми факторизації простих чисел на найшвидших комп'ютерах можуть зайняти достатньо часу, щоб зробити пошук недоцільним; тобто, зі збільшенням кількості розрядів простих чисел, що факторизуються, кількість операцій, необхідних для виконання факторизації на будь-якому комп'ютері, різко збільшується.

Багато криптографічних протоколів засновані на складності факторизації великих складених цілих чисел або на суміжній проблемі - наприклад, проблема RSA. Алгоритм, який ефективно факторизує довільне ціле число, зробить криптографію з відкритим ключем, засновану на RSA, незахищеною.

РОЗДІЛ 2
МАТРИЦІ ГАЛУА І ГЕНЕРАТОРИ ПВП

2.1 Матриця

Матриця - математичний об'єкт, записаний у вигляді прямокутної таблиці чисел (чи елементів кільця), він допускає операції (додавання, віднімання, множення та множення на скаляр). Зазвичай матриці мають вигляд двовимірних прямокутних таблиць, це значно спрощує простоту запису і її розуміння.. Іноді розглядають багатовимірні матриці або матриці непрямокутної форми.

Горизонталі елементи матриці називають елементами рядків, вертикальні – відповідно елементами стовпців. В позначеннях розмірності матриці першим йде індекс який вказує кількість рядків, другий кількість стовпців. Наприклад, запис $n \times m$ вказує на те, що матриця має n рядки і m стовпців.

2.2 Матриця Галуа

Терміни «матриці Галуа», які ми позначимо - G , як і пов'язані з ними оператором правостороннього транспонування «матриці Фібоначчі», позначені як - F , запозичені з теорії криптографії, в якій широко використовуються генератори псевдовипадкових послідовностей за схемами Галуа і Фібоначчі. Особливість даних матриць полягає в тому, що за допомогою їх можуть бути програмно обчислені такі ж самі бінарні послідовності, як і послідовності, що формуються генераторами псевдовипадкових послідовностей побудованих на регістрах зсуву з лінійними зворотними зв'язками.

В роботі використовуються так звані сполучені матриці G^* і F^* , які були утворені лівостороннім (класичним) транспонуванням базових (вхідних матриць) G і F відносно головної діагоналі вхідних матриць відповідно. Також використовуються обернені до базових матриць G і F матриці $\underline{G}$ і $\underline{F}$ для яких, також можна утворити зворотні матриці $\underline{G^*}$ і $\underline{F^*}$. Вище зазначена сукупність матриць Галуа і Фібоначчі, в сумі містить вісім різник матриць, таке різноманіття можна назвати множиною генераторів (матриць) Галуа.

2.3 Класичні генератори Галуа

До підмножини класичних генераторів псевдовипадкових послідовностей з максимальним періодом віднесено генератори, побудовані на основі лінійних регістрів зсуву з одним колом зворотного зв'язку, який є виключно функцією примітивного полінома (ПрП), що виконує роль формуючого полінома генератора.

D-тригери зазвичай використовуються як розряди РЗЛЗЗ, що перезаписують вхідний сигнал в момент подачі синхроімпульсу на вихід тригера. Приклад генератора Галуа четвертого порядку, де зворотний зв'язок формується примітивним поліномом $f = 1'0011$ четвертого степеня, наведено на рис. 2.1.

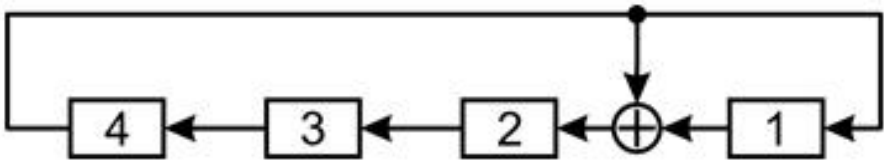

Рис. 2.1 «Структурна схема генератора ПВП в конфігурації Галуа»

Як видно зі структурної схеми генератора (рис. 1), зворотний зв'язок в класичних генераторах (регістрах) Галуа максимального періоду однозначно визначається обраним ПрП f_n ступеня n і формується наступним чином: Виходи кожного розряду (D - тригера) регістра зсуву (SR) подаються на входи наступних розрядів, будучи для них функціями збудження. Крім того, вихід старшого розряду регістра (за схемою XOR) подається на входи тих розрядів, номери яких відповідають номерам ненульових мономів СР. При цьому молодший одночлен, що лежить в правій частині полінома f_n, і молодший розряд регістра відповідають числу 1.

За допомогою рис. 2.1 виведемо мономіальне правило, за яким створюються структурні схеми класичних регістрів зсуву з лінійними зворотними зв'язками генераторів псевдовипадкових послідовностей в конфігурації Галуа. Для цього доповнимо схему пунктирними лініями, які розмістимо в тих точках схеми, де відсутні оператори XOR в нижньому ряду D-тригерів. Потім над суцільними вертикальними лініями (лініями зворотного зв'язку) ставимо цифри 1, а над пунктирними - цифри 0. Приходимо до рис. 2.2, який співпадає з рис. 2.1.

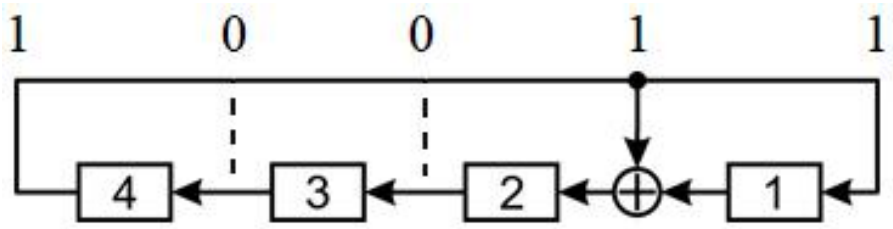

Рис. 2.2 «Побудова структурної схеми генератора Галуа четвертого порядку»

Як показано на рис. 2.2, одиниці примітивного полінома у векторній формі визначають лише положення вертикальних ліній в одноконтурному ланцюзі зворотного зв'язку класичного генератора РЗЛЗЗ-Галуа ПВП. Методика застосування сформульованого правила побудови структурної схеми генератора ПСП з максимальним періодом в конфігурації Галуа проілюстрована на прикладі побудови генератора, що генерується восьмим ступенем ПрП $f = 1'0110'0101$. Вирішення проблеми передбачає наступні два етапи синтезу.

Крок 1. Формуємо восьми розрядний кільцевий регістр зсуву (рис. 2.3), у вузлах лінії зворотного зв'язку якого розряди обраного примітивного полінома розташовані рівновіддалено:

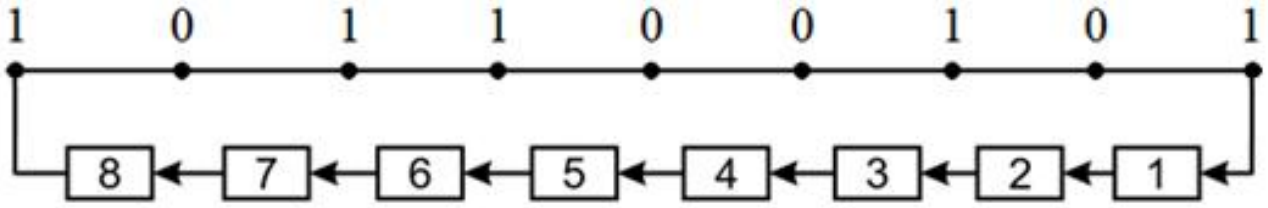

Рис. 2.3 «Основа схеми восьми розрядного Галуа-генератора ПВП»

Крок 2. об'єднавши окремі вузли лінії зворотного зв'язку оператором XOR, як на рис. 2.4, завершуємо побудову класичного РЗЛЗЗ-генератора Галуа, утвореного ПрП $f = 1'0110'0101$.

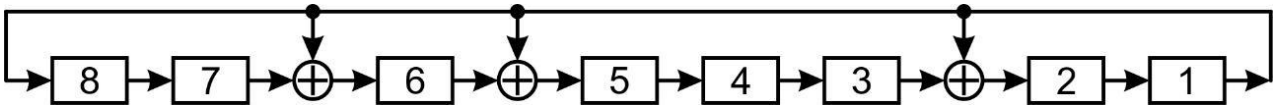

Рис. 2.4 «Структурна схема Галуа-генератора ПВП»

2.4 Класичні генератори Фібоначчі

Крім класичних генераторів РЗЛЗЗ на основі Галуа, в криптографії часто використовуються генератори в конфігурації Фібоначчі, які утворюються з

генераторів Галуа шляхом розвороту ланцюга зворотного зв'язку по вертикальній і горизонтальній осях на 180°, при збереженні колишньої нумерації елементів регістра зсуву. Структурна схема генератора Фібоначчі, утвореного ПрП $f = 1'0110'0101$, наведена на рис. 2.5.

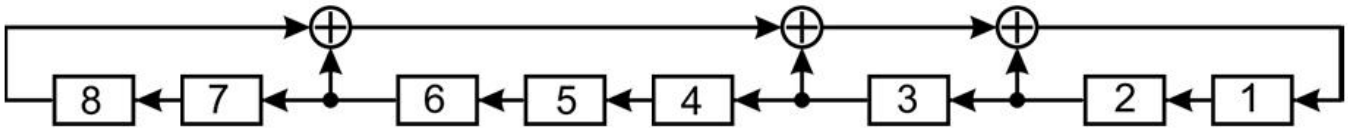

Рис. 2.5 «Структурна схема генератора Фібоначчі»

Кожному генератору Галуа або Фібоначчі РЗЛЗЗ відповідають однозначно задані матриці, які ми позначимо як відповідні генератори і позначимо символами G і F відповідно. Нехай $S(k)$ - стан n-розрядного генератора ПВП в конфігурації Галуа, що генерується ПрП f після k-го синхроімпульсу (на k-му кроці зсуву регістра), схема розрахунку якого представлена матричним виразом

$$S(k+1) = S(k) \cdot G_f^{(n)},\ k = 0, 1, \ldots S(0) = \underbrace{00\ldots01}_{n\ біт}. \qquad (2.1)$$

Наше завдання полягає в обчисленні для заданого примітивного полінома ступеня n матриці Галуа n-го порядку $G_f^{(n)}$,, за допомогою якої за співвідношенням (2.1) формується така ж послідовність псевдовипадкових чисел, як і фізичним генератором ПВП, який побудований на основі лінійного регістра зсуву, охопленого ланцюгом зворотного зв'язку, утвореним ПрП f

$$f = 1a_{n-1}, a_{n-2}\ldots a_k \ldots a_1\, 1,\ a_k \in GF(2) = \{0,1\},\ k = \underline{1, n-1}. \qquad (2.2)$$

Спочатку спробуємо розв'язати цю задачу для матриць малого порядку n. Візьмемо схему генератора ШПФ (рис. 2.6), зображену на рис. 1.

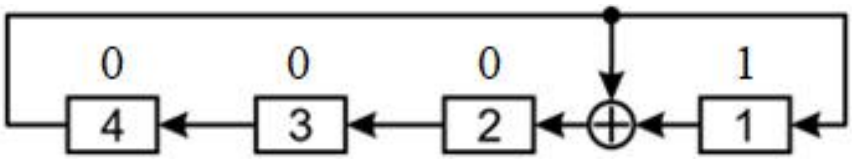

Рис. 2.6 «Ілюстрація початкового стану генератора ПВП за схемою Галуа»

Цифри над розрядами генератора вказують на рівень логічного сигналу на виході відповідного елемента (тригера) регістра. При надходженні імпульсу синхронізації одиниця з молодшого (правого) розряду генератора переміщується до

старшого (лівого) розряду, як показано на рис.2.7 для перших трьох імпульсів синхронізації.

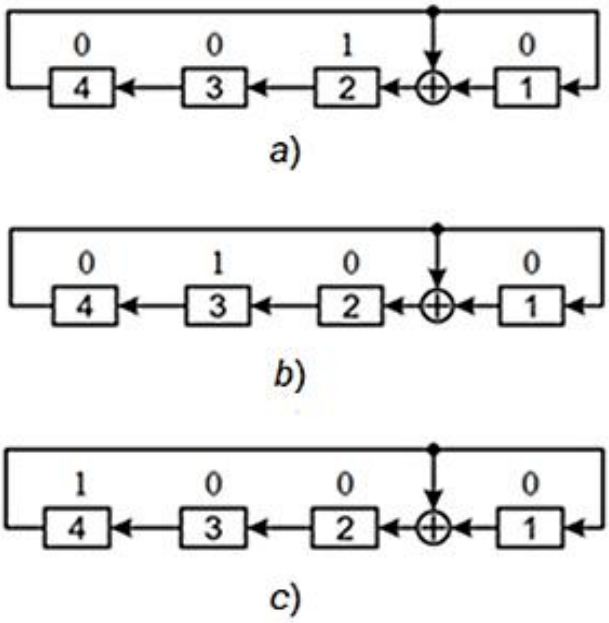

Рис. 2.7 «Стан генератора ПВП після:

a) – першого, b) – другого, c) – третього синхроімпульсу»

Звідси випливає рис.7, що після третього синхроімпульсу вони надходять на входи першого та другого тригерів і відповідно з'являються на виходах цих тригерів на четвертому кроці формування ПВП (рис. 2.8).

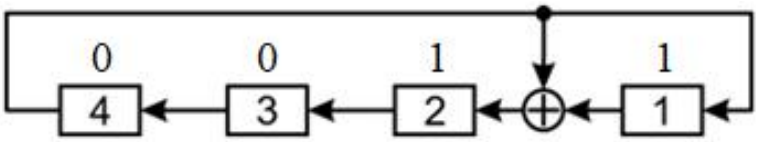

Рис. 2.8 «Стан генератора ПВП після четвертого синхроімпульсу»

Побудуємо матрицю $G_{13}^{(4)}$ з множини векторних станів S(k), в які переходить генератор Галуа після перших чотирьох синхронних імпульсів, розмістивши вектори в матриці, починаючи з їх найнижчого рядка $i = 1$.

$$\boldsymbol{G}_{13}^{(4)} = \begin{array}{c} \uparrow i \\ \left(\begin{array}{cccc} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{array}\right) \\ \leftarrow j \quad 4 \quad 3 \quad 2 \quad 1 \end{array} \begin{array}{c} \\ 4 \\ 3 \\ 2 \\ 1 \end{array}. \qquad (2.3)$$

Звернемо увагу на те, що нижній індекс 13 в позначенні матриці $G_f^{(n)}$ в (2.3) є ніщо інше, як 16-ричная запис ПрП $f = 1'0011$. Крім того, візьмемо до уваги, що нумерація рядків (i) матриці Галуа здійснюється знизу-вгору, а стовпців (j) - справа-наліво, що відрізняються від загальноприйнятих. Обраний спосіб нумерації рядків і стовпців матриці $G_f^{(n)}$ спрощує окремі завдання побудови структурно-логічних схем РЗЛЗЗ-генераторів ПВП.

Легко переконатися в тому, що, по-перше, рядки матриці (2.3) складають безліч лінійно незалежних векторів, в силу чого, що матриця $G_{13}^{(4)}$ виявляється невиродженою. По-друге, матриця $G_{13}^{(4)}$, будучи підставленою в рівняння (2.1), формує послідовність четирьохрозрядних кодових комбінацій (табл. 2.1), що є мультиплікативною групою поля, породжуваного ПрП $f = 1'0011$.

Зауважимо, що нижній індекс 13 в позначенні матриці $G_f^{(n)}$ в (2.3) є не що інше, як 16-розрядне позначення ПрП $f = 1'0011$. Крім того, слід зазначити, що нумерація рядків (i) матриці Галуа йде знизу вгору, а стовпців (j) - справа наліво, що відрізняється від загальноприйнятої нумерації. Обраний спосіб нумерації рядків і стовпців матриці $G_f^{(n)}$ спрощує деякі задачі побудови структурно-логічних схем генераторів РЗЛЗЗ.

Легко бачити, що, по-перше, рядки матриці (2.3) є набором лінійно незалежних векторів, так що матриця $G_{13}^{(4)}$ не є виродженою. По-друге, матриця $G_{13}^{(4)}$, підставлена в рівняння (2.1), утворює послідовність чотирирозрядних кодових

комбінацій (табл. 2.1), яка є мультиплікативною групою поля, що генерується ПрП $f = 1'0011$.

Таблиця 2.1

Послідовність станів генератора ПВП

Крок (k)	Розряди ЛРЗ					Крок (k)	Розряди ЛРЗ			
	4	3	2	1			4	3	2	1
0	0	0	0	1		8	1	0	0	1
1	0	0	1	1		9	0	1	0	0
2	0	1	0	1		10	1	1	0	0
3	1	1	1	1		11	1	0	1	1
4	1	1	1	0		12	0	0	1	0
5	1	1	0	1		13	0	1	1	0
6	1	0	0	0		14	1	0	0	0
7	0	1	1	1		15	0	0	0	1

І, по-третє, нарешті, верхній рядок матриці $G_{13}^{(4)}$ є не що інше, як четвертий степінь ПрП $f = 1'1111$, де вилучається старший елемент, а старший (лівий) елемент усіченого полінома є коефіцієнтом a_{n-1}.

Виходячи з аналізу матриці $G_{13}^{(4)}$, приходимо до наступного правила побудови (алгоритму синтезу) класичних матриць Галуа (КГМ) *n*-го порядку $G_f^{(n)}$, породжених примітивними поліномами *n*-го ступеня.

2.5 Алгоритм синтезу КМГ

Помістимо елемент форми (УЕ) класичної матриці Галуа, який співпадає з мінімальним примітивним елементом $\theta = 10$ мультиплікативної групи поля $GF(2^n)$, породженої ПрП f степеня n, у правий нижній кут утвореної матриці $G_f^{(n)}$. Всі інші розряди в рядку зліва від елемента заповнюються нулями. Кожен наступний рядок матриці (у напрямку знизу вгору) формується шляхом зсуву на один біт вліво від попереднього рядка, а у вільну комірку праворуч записується нуль. Оскільки при

формуванні верхнього рядка КМГ його старша одиниця (при зсуві попереднього рядка на один розряд вліво) виходить за лівий край матриці, то цей рядок, який раніше став $(n+1)$-розрядним, необхідно звести до залишку по модулю незвідного полінома f і таким чином повернути в межі матриці.

Першу частину матриці $G_{13}^{(4)}$ можна записати в більш компактному вигляді:

$$G_f^{(n)} = (\blacktriangleleft f\ E\ 0), \qquad (2.4)$$

де **E** - одинична матриця $^{(n-1)}$ -го порядку; **0** - нульовий вектор-стовпець довжини $^{n-1}$; и $\blacktriangleleft$ - показник розміщення коефіцієнтів a_k ПрП ^{f} , стрілка якого спрямована в бік розташування старшого коефіцієнта a_{n-1}.

Загальна форма КМГ $G_f^{(n)}$ виглядає наступним чином:

$$\boldsymbol{G}_f^{(n)} = \begin{pmatrix} \boldsymbol{\alpha}_{n-1} & \boldsymbol{\alpha}_{n-2} & \boldsymbol{\alpha}_{n-3} & \cdots & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_1 & \mathbf{1} & n \\ \mathbf{1} & 0 & 0 & \cdots & 0 & 0 & \mathbf{0} & n-1 \\ 0 & \mathbf{1} & 0 & \cdots & 0 & 0 & \mathbf{0} & n-2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \mathbf{0} & 3 \\ 0 & 0 & 0 & \cdots & \mathbf{1} & 0 & \mathbf{0} & 2 \\ 0 & 0 & 0 & \cdots & 0 & \mathbf{1} & \mathbf{0} & 1 \\ n & n-1 & n-2 & \cdots & 3 & 2 & 1 & \end{pmatrix} \qquad (2.5)$$

У матриці $G_f^{(n)}$ (2.5) жирним шрифтом для наочності виділені елементи головної діагоналі одиничної матриці E і оздоблюючі елементи цієї матриці (праворуч - нульовий стовпець **0**, а зверху - рядок, що є укороченим на один розряд ліворуч примітивним поліномом f, що породжує КМГ $G_f^{(n)}$).

У матриці $G_f^{(n)}$ (2.5) для наочності напівжирним шрифтом виділено елементи головної діагоналі одиничної матриці E та кінцеві елементи цієї матриці (праворуч нульовий стовпець **0** та вгорі рядок, що є примітивним поліномом f, усіченим на один розряд вліво, який породжує КМГ $G_f^{(n)}$).

2.6 Алгоритм синтезу матриць Фібоначчі

Компактні форми матриць Фібоначчі $F_f^{(n)}$ взаємопов'язані з матрицями Галуа $G_f^{(n)}$ оператором правостороннього транспонування

$$G_f^{(n)} \ \perp\!\!\leftrightarrow \ F_f^{(n)} = (\Theta\, f\, E\, \blacktriangledown), \qquad (2.6)$$

де Θ - нульовий вектор-рядок $^{(n-1)}$-го порядку.

Наведемо, для прикладу (див. 2.7), вирази для матриць G і F, породжуваних ПрП восьмої ступені $f = 1'0110'0101$. Структурно-логічні схеми Галуа і Фібоначчі РЗЛЗЗ-генераторів ПВП, що відповідають співвідношенням (6), показані вище на рис. 4 і 5 відповідно.

$$\boldsymbol{G} = \begin{array}{c} \\ \begin{pmatrix} 0&1&1&0&0&1&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \end{pmatrix} \\ j\quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} i \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array}, \quad \boldsymbol{F} = \begin{array}{c} \\ \begin{pmatrix} 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&1 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&1 \\ 0&0&0&0&0&1&0&1 \\ 0&0&0&0&0&0&1&0 \end{pmatrix} \\ j\quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} i \\ 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array} \qquad (2.7)$$

Доповнивши символічні форми (2.4) і (2.6) матриць Галуа G і F Фібоначчі відповідними їм сполученими матрицями G^* і F^*, утвореними лівостороннім транспонуванням базових матриць,

$$G(F)\ T \leftrightarrow\ G^*(F^*) = (\blacktriangle\, E\, f\, \Theta)((0\, E\, f\, \blacktriangleright)), \qquad (2.8)$$

приходимо до схеми взаємозв'язку (рис. 2.9) елементів підмножини «прямих» матриць Галуа, яке позначимо $\{G\}$.

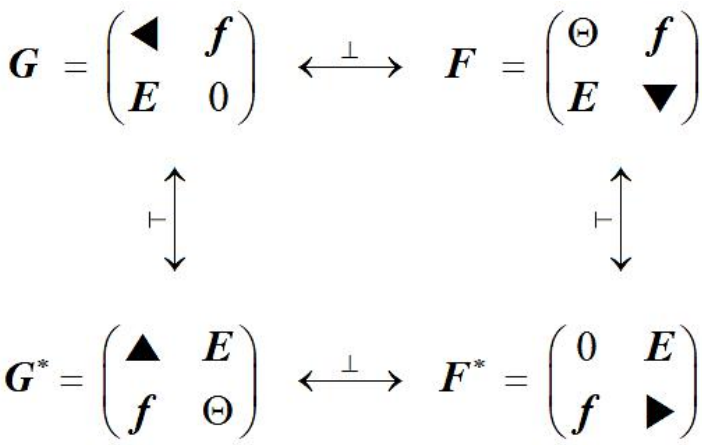

Рис. 2.9 «Схема взаємозв'язку базових і сполучених «прямих» матриць Галуа»

Зв'язані матриці Галуа G^* і F^* Фібоначчі восьмого порядку, породжувані перетвореннями (2.8) і матрицями (2.7), і мають вигляд:

$$G^* = \begin{array}{c} i \\ \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 1&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \end{pmatrix} \\ j \quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array}, \quad F^* = \begin{array}{c} i \\ \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&1&0&0&1&1&0 \end{pmatrix} \\ j \quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array} \qquad (2.9)$$

а відповідні їм схеми РЗЛЗЗ-генераторів ПВП представлені на рис. 2.10, утворені $f = 1'0110'0101$.

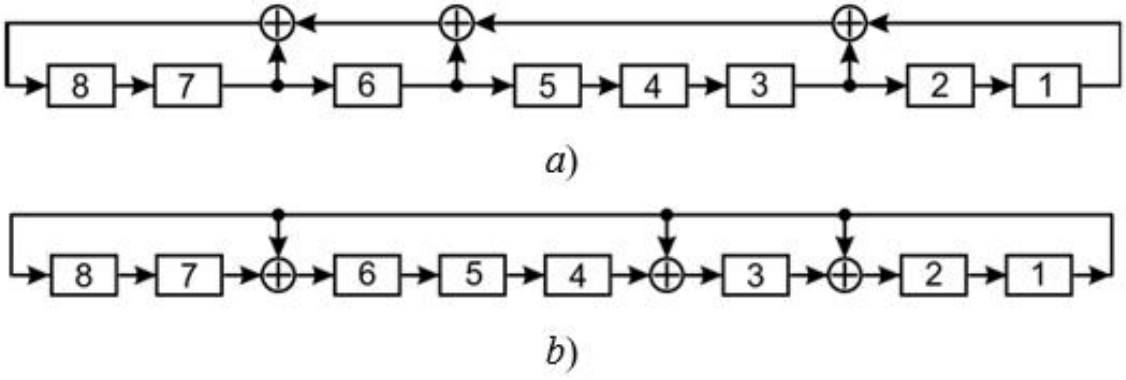

Рис. 2.10 «Структурні схеми сполучених генераторів ПСП в конфігураціях Галуа і Фібоначчі»

Кожній «прямій» матриці з підмножини $\{G_g\} \in (G, F, G^*, F^*)$, зображеній на рис. 9, відповідають так звані «обернені» матриці, множина яких утворює підмножину $\{\underline{G}\} \in (\underline{G}, \underline{F}, \underline{G}^*, F^*)$. Доповнимо контур матриці $\{\underline{G}\}$, зображений на рис. 5 (назвемо його внутрішнім контуром), іншим, який назвемо зовнішнім контуром, у вузлах якого містяться матриці підмножини $G_f^{(n)}$. Проблема має тривіальне рішення. Як видно з (2.9), матриці Галуа $G_f^{(n)}$ та їх складові елементи θ пов'язані відношенням ізоморфізму. Звідси випливає, що дві матриці Галуа, породжені одним і тим же незвідним (примітивним - для КМГ) поліномом, стають взаємно оберненими, якщо їх складові елементи взаємно обернені. Таким чином, для побудови матриці $\underline{G}_f^{(n)}$, оберненої до $G_{f,\theta}^{(n)}$, достатньо на етапі синтезу матриці $\underline{G}_f^{(n)}$ замінити УЕ в матриці $G_f^{(n)}$ на її обернене значення $\underline{\theta}$. Звідси випливає наступний взаємозв'язок:

$$\underline{G}_{f,\theta}^{(n)} = G_{f,\theta}^{(n)} \qquad (2.10)$$

Для КМГ, утворених ПрП f та формоутворюючим елементом $\theta = 10$, інваріантне означення виконується. Якщо ω - ненульовий елемент розширеного поля Галуа, то $\underline{\omega}$ - обернена до ω величина, якщо виконується умова $(\omega \cdot \underline{\omega}) mod\, f = 1$. Нехай поліном $f = 1a_{n-1}a_{n-2}, ..., a_1 1$ - примітивний двійковий поліном степеня n та формуючий елемент $\theta = 10$ матриці $G_{f,\theta}^{(n)}$. Тоді $\theta = 1a_{n-1}a_{n-2}, ..., a_1$, оскільки добуток $\underline{\theta} \cdot \theta$, що дорівнює $\theta = 1a_{n-1}a_{n-2}, ..., a_1 0$, зводиться до залишку за модулем f і утворює залишок 1, як і вимагається для пари обернених. Це дає наступний загальний вигляд оберненого КМГ:

$$\bar{\boldsymbol{G}}_f^{(n)} = \begin{pmatrix} \mathbf{0} & \mathbf{1} & 0 & \cdots & 0 & 0 & 0 \\ \mathbf{0} & 0 & \mathbf{1} & \cdots & 0 & 0 & 0 \\ \mathbf{0} & 0 & 0 & \cdots & 0 & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & 0 & 0 & \cdots & 0 & \mathbf{1} & 0 \\ \mathbf{0} & 0 & 0 & \cdots & 0 & 0 & \mathbf{1} \\ \mathbf{1} & \boldsymbol{\alpha}_{n-1} & \boldsymbol{\alpha}_{n-2} & \cdots & \boldsymbol{\alpha}_3 & \boldsymbol{\alpha}_2 & \boldsymbol{\alpha}_1 \end{pmatrix} \begin{matrix} n \\ n-1 \\ n-2 \\ \cdots \\ 3 \\ 2 \\ 1 \end{matrix}$$
$$n \quad n-1 \quad n-2 \quad \cdots \quad 3 \quad 2 \quad 1 \qquad (2.11)$$

яку також можливо представити в більш компактній формі

$$\underline{G}_f^{(n)} = (0\ E\ f\ \blacktriangleleft) \qquad (2.12)$$

Зв'язування обернених матриць Галуа $\{\underline{G}\}$ визначається тими ж операторами (лівим і правим транспонуванням) і відбувається за тією ж схемою, що і для прямих матриць $\{G\}$ на рис. 5. Якщо внутрішній цикл породжується матрицею (2.4), то зовнішній цикл породжується матрицею (2.9). Схема з'єднання елементів множини $\{Q\} = \{G\} \cup \{\underline{G}\}$ показана на рис. 2.11 [2].

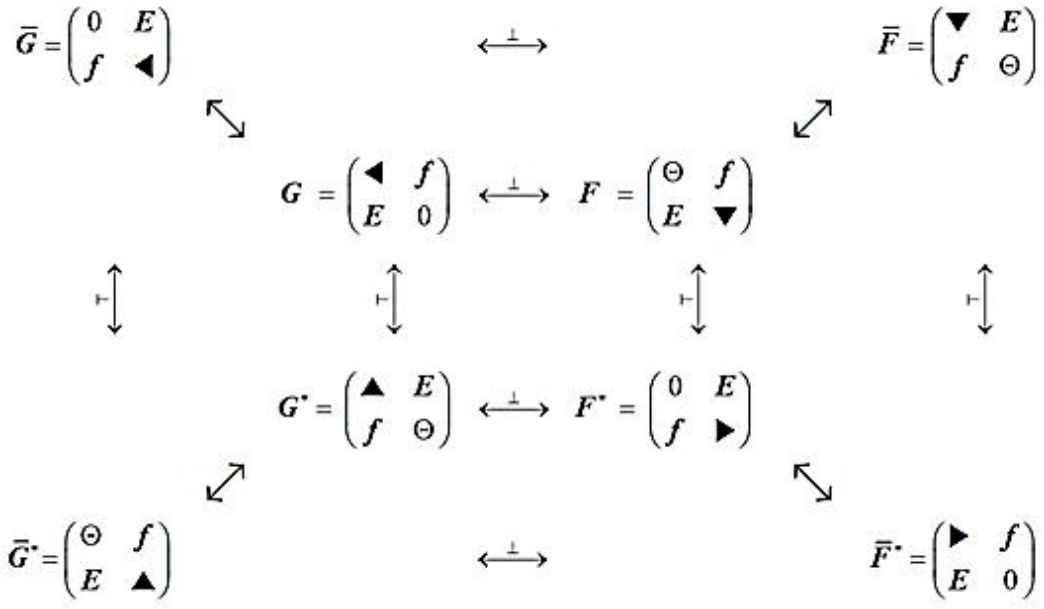

Рис. 2.11 «Схема взаємозв'язку елементів повної множини класичних матриць Галуа»

Сукупність зворотних матриць Галуа і Фібоначчі, породжуваних ПрП $f =$ 1'0110'0101 зведена в систему співвідношень:

$$\bar{G} = \begin{array}{c} i \\ \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&1&1&0&0&1&0 \end{pmatrix} \\ j \quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array}; \quad \bar{F} = \begin{array}{c} i \\ \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 1&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \end{pmatrix} \\ j \quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array}, \qquad (2.13)$$

а відповідні їм структурно-логічні схеми РЗЛЗЗ-генераторів ПВП зображені на рис.2.12.

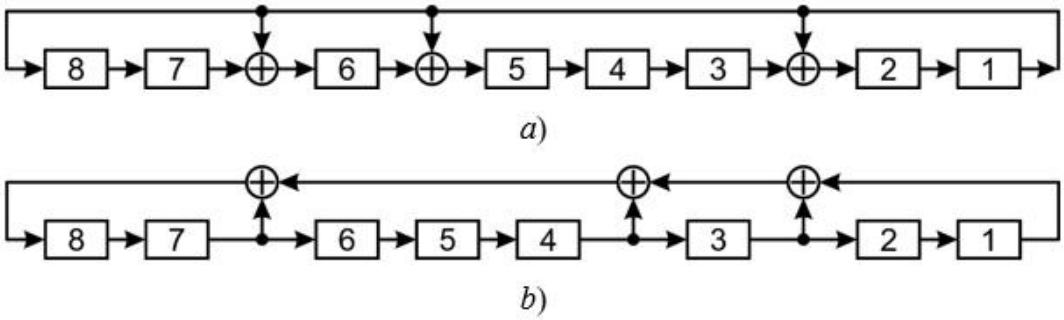

Рис. 2.12 «Структурні схеми зворотних генераторів ПСП в конфігураціях:

a) Галуа і b) Фібоначчі»

»

Зв'язані зворотні матриці Галуа $\underline{G}^*$ і Фібоначчі $\underline{F}^*$ восьмого порядку мають такий вигляд:

$$\bar{G}^* = \begin{array}{c} i \\ \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&1&1&0&0&1&0 \end{pmatrix} \\ j \quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array}; \quad \bar{F}^* = \begin{array}{c} i \\ \begin{pmatrix} 0&1&0&0&0&0&0&0 \\ 1&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0 \\ 1&0&0&0&0&1&0&0 \\ 1&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0 \end{pmatrix} \\ j \quad 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1 \end{array} \begin{array}{c} 8 \\ 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \end{array}. \qquad (2.14)$$

Структурно-логічні схеми зворотних сполучених РЗЛЗЗ-генераторів ПВП, що відповідають матрицями (2.14) представлені на рис. 2.13.

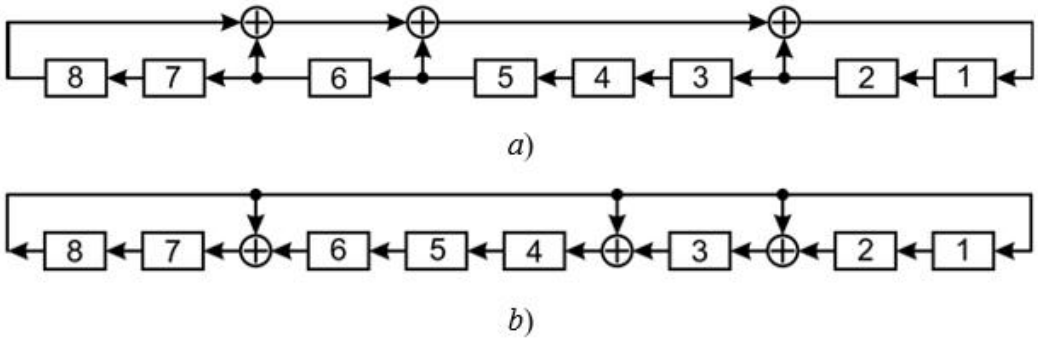

Рис. 2.13 «Структурні схеми зворотних сполучених генераторів ПВП в конфігураціях: a)Галуа і b)Фібоначчі»

Зауважимо, що якщо в прямих базових РЗЛЗЗ -генераторах на транзисторах в конфігураціях Галуа (рис. 4) та Фібоначчі (рис. 5) зворотні зв'язки розрядів регістрів «закручені» за годинниковою стрілкою, а у відповідних зв'язаних генераторах (рис. 10) - проти годинникової стрілки, то в інверсних генераторах - навпаки. Дійсно, зворотні зв'язки в інверсних базових генераторах ПВП (рис. 12) закручені проти годинникової стрілки, а в інверсних зв'язаних генераторах (рис. 13) закручені за годинниковою стрілкою.

2.7 Ефективні алгоритми обчислення станів класичних генераторів ПВП

Складність алгоритму обчислення стану одного з восьми класичних генераторів псевдовипадкових послідовностей на рис. 11, згідно співвідношення (2.1) становить $O(n^2)$, тобто зростає квадратично в залежності від порядку *n* класичних матриць Галуа, що відповідають генераторам ПВП. На основі структур КМГ (в основному за рахунок їх складових - одиничних матриць порядку $E\ (n-1)$ вдається суттєво скоротити обчислювальний час, що витрачається на розрахунок стану генератора ПВП на наступному $(k+1)$ обчислювальному кроці.

Для спрощення вводимо систему позначень, яка дещо відрізняється від тієї, що використовувалася до цього часу. $V_k = \{v_{k-1}, v_{n-2}, ..., v_1, v_0$ - вектор ПВП на k-му кроці генерації, де в фігурних дужках вказані двійкові складові вектора; $f = \{a_n = 1, a_{n-1}, ..., a_1, a_0 = 1\}$ - примітивний поліном степеня *n*, який генерує ШМ.

Остаточні співвідношення, що визначають вектори v_{k+1} для різних КМГ, наведені в табл. 2.2.

Таблиця 2.2

Рекурентні обчислення станів класичних Галуа-генераторів ПВП

Матриця Галуа	V_{k+1}
G	$\oplus \frac{V_{n-1} \cdot f\,(a_{n-1}) \setminus a_n, a_0}{V_k(n-2)_{\text{ъ}n-1\ біт}}, V_{n-1}$
F	$V_k(n-2)_{\text{ъ}n-1\ біт}, \oplus\,(V_k \uparrow \otimes f \downarrow)_{\text{ъ}1\ біт}$
G^*	$\oplus\,(V_k \uparrow \otimes f \uparrow)_{\text{ъ}1\ біт}, V_k((n-1) \setminus v_0)_{\text{ъ}n-1\ біт}$
F^*	$v_0, \oplus \frac{V_k((n-1) \setminus v_0)}{v_0 \cdot \tilde{f}(a_{n-1}) \setminus a_n, a_0}_{\text{ъ}n-1\ біт}$
$\underline{G}$	$v_0, \oplus \frac{V_k((n-1) \setminus v_0)}{v_0 \cdot \tilde{f}(a_1) \setminus a_n, a_0}_{\text{ъ}n-1\ біт}$
$\underline{F}$	$\oplus\,(V_k \uparrow \otimes f \downarrow)_{\text{ъ}1\ біт}, V_k((n-1) \setminus v_0)_{\text{ъ}n-1\ біт}$
$\underline{G}^*$	$V_k(n-2)_{\text{ъ}n-1\ біт}, \oplus\,(V_k \uparrow \otimes f \uparrow)_{\text{ъ}1\ біт}$
$\underline{F}^*$	$\oplus \frac{v_{n-1} \cdot \tilde{f}\,(a_1) \setminus a_n, a_0}{V_k(n-2)_{\text{ъ}n-1\ біт}}$

У табл. 2.2 прийняті такі позначення: $f(a_{n-1}) \setminus a_n, a_0 = (a_{n-1}, a_{n-2} \ldots, a_1)$ вектор, який не містить коефіцієнтів a_n, a_0 ; $\tilde{f}(a_{n-1}) \setminus a_n, a_0 = (a_1, a_2 \ldots, a_{n-1})$ - вектор, інверсний вектору НП f, в якому видалені коефіцієнти a_n, a_0; $V_k(n-2) = \{v_{n-2}, v_{n-3}, \ldots, v_1, v_0\}$ - зсунутий на один розряд вліво вектор V_k, з якого виключений старший розряд V_{n-1}; $V_k(n-1) \setminus v_0 = \{v_{n-1}, v_{n-2}, \ldots, v_2, v\}$; стрілки ↑ і ↓ в середній групі формул в табл. 2 вказують на розташування старших елементів векторів, записаним зліва від стрілок; і наприкінці,

У таблиці 2.2 використані наступні позначення: $f(a_{n-1}) \setminus a_n, a_0 = (a_{n-1}, a_{n-2} \ldots, a_1)$ - вектор, який не містить коефіцієнтів a_n, a_0 ; $\tilde{f}(a_{n-1}) \setminus a_n, a_0 = (a_1, a_2 \ldots, a_{n-1})$ - вектор, обернений до НП-вектора f з вилученням коефіцієнтів a_n, a_0; $V_k(n-2) = \{v_{n-2}, v_{n-3}, \ldots, v_1, v_0\}$ - вектор V_k, зсунутий на один біт вліво, з

якого виключається старший біт V_{n-1} , $V_k(n-1) \setminus v_0 = \{v_{n-1}, v_{n-2}, ..., v_2, v\}$; стрілки ↑ та ↓ в середній групі формул в табл. 2 вказують на розташування старших елементів векторів зліва від стрілок; і в кінці,

$(V_k \uparrow \cdot f \uparrow) = (v_{n-1} \cdot a_{n-1}\, v_{n-2} \cdot a_{n-2} \vdots v_0 \cdot a_0)$; $(V_k \uparrow \cdot f \downarrow) = (v_{n-1} \cdot a_0\, v_{n-2} \cdot a_1 \vdots v_0 \cdot a_{n-1})$; *(2.15)*

Проводимо аналіз виразів для векторів v_{k+1} , після чого робимо висновок, що запропоновані алгоритми формування ПВП значно простіші, а обчислювальна складність становить $O(n)$, тобто лінійно залежить від порядку матриць Галуа, які генерують генератори двійкових псевдовипадкових послідовностей.

Основним недоліком класичних ПВП -генераторів Галуа в апаратному та програмному плані є те, що вони не захищені від атаки Бурлекампа-Массі. У цій роботі ми пропонуємо два шляхи протидії таким атакам. Перший з них стосується переходу від класичних до узагальнених генераторів псевдовипадкових послідовностей. Такий перехід супроводжується розширенням різноманітності генераторів як за рахунок збільшення кількості формуючих елементів, так і за рахунок того, що узагальнені генератори синтезуються не тільки на основі примітивних поліномів (як у випадку класичних генераторів), а й поліномів, які не обов'язково є примітивними. Другим конструктивним способом захисту від атаки Бурлекампа-Мейсі є використання перетворень подібності матриць форми в класичних або узагальнених генераторах псевдовипадкових послідовностей.

РОЗДІЛ 3
УЗАГАЛЬНЕНІ ГЕНЕРАТОРИ І МАТРИЦІ ГАЛУА

3.1 Узагальнені генератори

До множини узагальнених Галуа-генераторів ПВП максимального періоду будемо відносити генератори, побудовані на основі лінійних регістрів зсуву, охоплених багатоконтурною схемою зворотного зв'язку, яка залежить як від незвідного полінома *f* ступеня *n* (зовсім не обов'язково примітивного), що грає роль породжуваного полінома генератора, так і утворюючого елемента $\theta > 10$, що є примітивним елементом поля $GF(2^n)$, породжуваного НП *f*.

Матрицю Галуа $G_{f,\theta}^{(n)}$, за допомогою якої програмно формується така ж ПВП, як і послідовність, утворена узагальненим РЗЛЗЗ-генератором, називається узагальнена матриця Галуа (УМГ).

Розглянуто сукупність узагальнених максимально періодичних генераторів Галуа для генераторів, побудованих на основі лінійних регістрів зсуву, охоплених багатоконтурною схемою зворотного зв'язку, яка залежить як від незвідного полінома *f* ступеня *n* (не обов'язково примітивного), що відіграє роль породжуваного полінома генератора, так і від формуючого елемента $\theta > 10$, який є примітивним елементом поля $GF(2^n)$,, породжуваного НП *f*.

Матриця Галуа $G_{f,\theta}^{(n)}$, яка використовується для програмної генерації такого ж ПВП, як і послідовність, що генерується узагальненим генератором РЗЛЗЗ, називається узагальненою матрицею Галуа (УГМ). 3.2

3.2 Правило синтезу УМГ

Вибраний примітивний елемент $\theta > 10$ поля $GF(2^n)$, породженого НП f степеня *n*, розміщується у правому нижньому куті породженої матриці $G_f^{(n)}$. Елемент виступає як формуючий елемент матриці $G_{f,\theta}^{(n)}$. Всі розряди в рядках зліва від елемента θ заповнюються нулями. Кожен наступний рядок матриці в напрямку знизу вгору формується шляхом зсуву на один розряд вліво в попередньому рядку. У рядку, що звільнився в результаті зсуву, нуль записується в комірці праворуч.

Якщо на кроці ітерації старша ненульова цифра рядка виходить за лівий край матриці, то цей рядок модулюється із залишком і таким чином повертається до меж матриці. Далі процес продовжується за описаною схемою до заповнення всіх рядків матриці $G_{f,\theta}^{(n)}$.

Далі розглянемо приклади синтезу підмножини прямих примітивних УМГ $\{G_g\} \in (G_g, F_g, G_g^*, F_g^*)$ та побудови на їх основі генераторів ПВП з максимальним періодом. В якості незвідних виберемо двійковий поліном четвертого степеня $f = 1'1111$, який не є примітивним, та примітив UE $\theta = 111$. Побудовані матриці з даними параметрами набувають наступного вигляду

$$\boldsymbol{G}_g = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}; \qquad \boldsymbol{F}_g = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix};$$

$$\boldsymbol{G}_g^* = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}; \qquad \boldsymbol{F}_g^* = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}. \qquad (3.1)$$

На наступному рис 3.1 представлена структурна схема що відповідає узагальненій матриці Галуа G_g, узагальненого чотирьохрозрядного генератора Галуа, Вертикально розташовані регістри генераторів, відмічені зверху символом $\otimes$, реалізують операцію порозрядного множення, а регістри, що містять внутрішній символом $\oplus$ - операцію складання вмісту регістра по модулю 2.

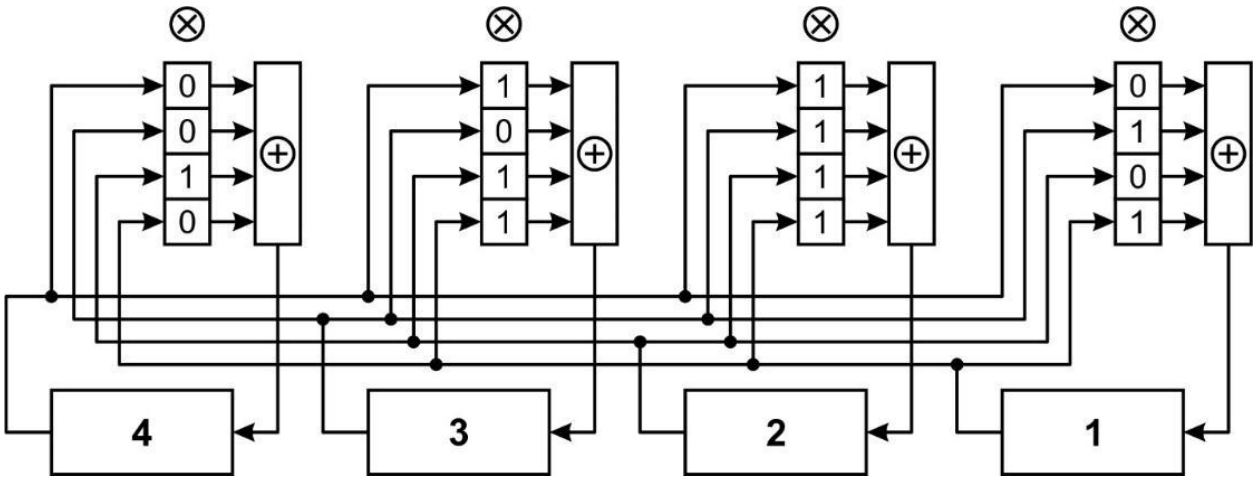

Рис. 3.1 «Структурна схема «прямого» узагальненого базового генератора Галуа»

Замінивши в рис. 14 комірки вертикальних регістрів зворотного зв'язку елементами матриці F_g з системи (3.1), отримаємо схему «прямого» узагальненого генератора ПВП в конфігурації Фібоначчі. Структурно-логічна схема генератора ПВП, сполучена схемою розглянутого генератора Галуа (рис. 3.1), представлена на рис. 3.2.

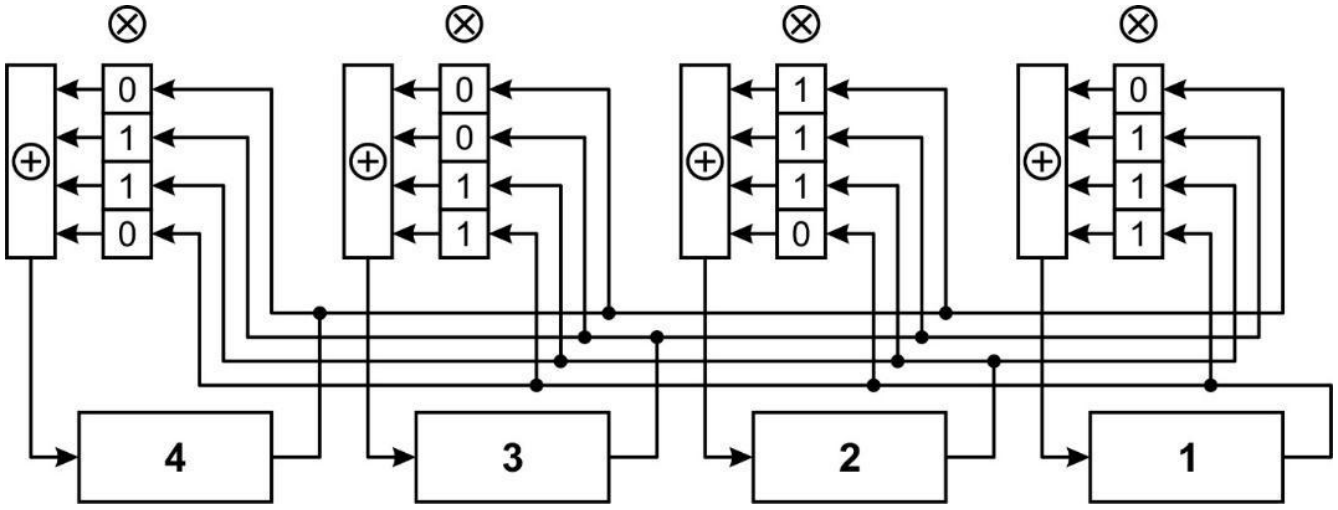

Рис. 3.2 «Структурна схема «прямого» сполученого узагальненого генератора Галуа»

Якщо в схемі на рис. 15 замінити вміст комірок регістрів зворотного зв'язку на елементи матриці F^* із системи (3.1), то прийдемо до "прямого" зв'язку генератора узагальненого ПВП в конфігурації Фібоначчі. Блок-схеми, наведені на рис. 3.1 та 3.2 - приклади регістрів з багатоконтурним зворотним зв'язком.

Друга множина узагальнених матриць складається з матриць, що містяться в підмножині узагальнених "обернених" матриць Галуа $\left\{\underline{G}_g\right\} \in \left(\underline{G}_g, \underline{F}_G, \underline{G}_g^*, \underline{F}_g^*\right)$.

Множина матриць $\{G_g\}$ та $\left\{\underline{G}_g\right\}$ разом утворюють повну множину узагальнених матриць Галуа $\left\{\underline{Q}_g\right\} = \{G_g\} \cup \left\{\underline{G}_g\right\}$. Всі матриці підмножини $\left\{\underline{G}_g\right\}$ однозначно визначаються перетворенням матриці $\underline{G}_g$ лівим та правим транспонентами. Матриця $\underline{G}_g$ в свою чергу розраховується за наведеним вище правилом синтезу УМГ, де в якості УЕ θ слід використовувати її обернену $\underline{\theta}$.

Для примітивного утворює елемента $\theta = 111$ і довічного породжувального НП $f = 1'1111$ зворотне значення $\underline{\theta} = 1001$. Сукупність зворотних матриць підмножини $\left\{\underline{G}_g\right\}$ представлена системою:

$$\bar{\boldsymbol{G}}_g = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}; \qquad \bar{\boldsymbol{F}}_g = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix};$$

$$\bar{\boldsymbol{G}}_g^* = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}; \qquad \bar{\boldsymbol{F}}_g^* = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}. \qquad (3.2)$$

Структурно-логічні схеми генераторів РЗЛЗЗ ПВП, що відповідають матрицям $\underline{G}_g$ та $\underline{F}_g$ із системи (3.2), узгоджуються зі схемами на рис. 15, а для матриць $\underline{G}_g^*$ і $\underline{F}_g^*$ - на рис. 14.

Головною відмінністю від класичних матриць $\{Q\}$ від узагальнені матриці Галуа $\left\{\underline{Q}_g\right\}$ полягає в наступному. В узагальнених матрицях не має можливості обрати одиничну матрицю $(n-1)$ - го порядку Е, тоді як у випадку з КМГ можемо явно виділити нульовий вектор **0** (або вектор-рядок Θ). Звідси приходимо до висновку, що для матриць множини Q_g не існує запису в компактній формі,

подібних до форм групи матриць $\{Q\}$. Взаємозв'язок між узагальненими матрицями Галуа показано на рис. 3.3 Для спрощення нижній індекс у назвах матриць опущено.

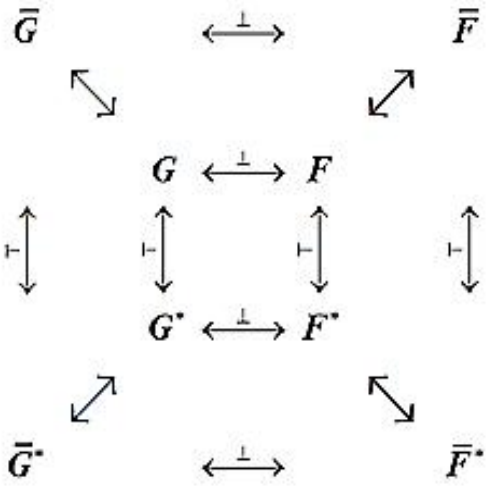

Рис. 3.3 «Схема взаємозв'язку повної множини узагальнених матриць Галуа»

Схема взаємозв'язку елементів множин класичних $\{Q\}$ і $\left\{\underline{Q}_g\right\}$ узагальнених матриць Галуа , представлених на рис. 11 і 16, відображена в табл. 3.1.

Звертаючись до матричної системи (2.14), можемо звернути увагу на те, що утворюючий елемент матриці більший за значення 10. Однак це зовсім не означає того, що матриця $\underline{G}$ в (2.14) вважається узагальненою. Оскільки вона обернена до КМГ, то матриця G також належить до групи класичних матриць Галуа. Однозначною матриця Галуа належить до сімейства КМГ, якщо можна прослідкувати в її контурі одиничку матрицю $E(n-1)$ та примикання нульового вектора рядка Θ (або вектор стовпця **0**) такого самого порядку, який і порядок матриці Е. Наприклад, одиничні матриці в явному вигляді містяться у співвідношеннях (2.7), (2.9), (2.12) та (3.1), які входять в повну множину (див. рис. 11) класичних матриць Галуа $\{Q\}$ (ПВП -генераторів).

Схему зв'язку елементів множин класичних $\{Q\}$ і $\left\{\underline{Q}_g\right\}$ узагальнених матриць Галуа показано на рис. 11 та 16, наведено в таблиці 3.1.

Таблиця 3.1

Взаємозв'язок матриць Галуа

	$\boldsymbol{G}$	$\boldsymbol{F}$	$\boldsymbol{G}^*$	$\boldsymbol{F}^*$
$\boldsymbol{G}$	–	$\perp$	T	$\perp T$
$\boldsymbol{F}$	$\perp$	–	$\perp T$	T
$\boldsymbol{G}^*$	T	$\perp T$	–	$\perp$
$\boldsymbol{F}^*$	$\perp T$	T	$\perp$	–

Перехід від класичних до узагальнених Галуа-генераторам ПВП являється одним із можливих варіантів захисту генератора він атаки Берлекемпа-Мессі.

3.3 Перетворення подібності матриць Галуа

Існує можливість розширення різноманітності генераторів псевдовипадкових послідовностей Галуа за вдяки можливості зміни подібності матриць Галуа, які генеруються генераторами ПВП, що відповідно підвищує їх криптографічну стійкість. Таку трансформацію запроваджує відношення:

$$\widetilde{G} = P^{-1} \cdot G \cdot P, \qquad (3.3)$$

з виразу отримуємо матриця $\widetilde{G}$, множини $\{Q\}$ або Q_g що вважається подібною матриці G, а P - матриця перетворення подібності, невироджена того ж порядку, як і порядок матриці G. Для матриць P - зручно застосовувати переставні матриці, оскільки для них досить просто обчислюється обернена матриця перетворення подібності P^{-1}, а саме: $P^{-1} = P^{-T}$.

Перетворення подібності матриць Галуа пропонують другу можливість захисту від атаки Бурлекампа-Массі на генератори ГПСЧ. Основним недоліком перетворень подібності матриць Галуа, що використовуються в класичних генераторах ГПСЧ, є наступне. Такі перетворення в матрицях Галуа руйнують окремі матриці Е, що виключає можливість розробки швидких алгоритмів (див. табл. 2) для обчислення послідовностей, що генеруються PRT-генераторами. При цьому

перетворення подібності в узагальнених генераторах матриць Галуа не впливають на час обчислення ДПФ, оскільки їх матриці Галуа не містять матриць Е.

3.4 Генерація поліномів

Концептуальні правила синтезу незвідних поліномів. Процес синтезу незвідних поліномів розпочинається з перевірки ряду простих положень (аксіом), доведені шляхом емпіричних досліджень, які суттєво полегшують процес обчислення НП [1]:

Аксіома 1. Векторна форма незвідного полінома має форму $f = 1a_n\, a_{n-1}...a_k...a_1\, 1$, тобто незвідний поліном повинен обрамлятись зліва та справа одиницями.

Аксіома 2. Вага внутрішніх коефіцієнтів полінома повинна бути непарним числом, тому що в противному випадку f_n ділиться без залишку на поліном першого ступеня $f_1 = 11$ і, тим самим, тестований поліном являється звідним.

Аксіома 3. Максимальний порядок $L_n^{\sim}$ НП f_n, визначається виразом:

$$L_n^{\sim} = 2^n - 1. \qquad (3.4)$$

Аксіома 4. Якщо f - незвідний поліном, то і двосторонній йому теж є незвідним.

Аксіома 5. Примітивним є НП максимального порядку

Аксіома 6. Порядок L_n незвідного полінома f_n збігається з порядком елемента $\theta = 10$ поля $GF(2^n)$ породжуваного НП f_n.

Аксіома 7. Порядок L_n незвідного полінома f_n є дільником максимального порядку $L_n^{\sim}$ тобто дотримується співвідношення:

$$L_n \mid L_n^{\sim} = (2^n - 1) \qquad (3.5)$$

Введемо ряд числових параметрів (див. табл. 2.2), «пов'язавши» їх з характеристиками так званої реперною сітки (рис. 3.4), що складається із сукупності паралельних прямих ліній (сходинок сітки). Число ступенів сходів збігається зі ступенем тестованого на незвідність полінома.

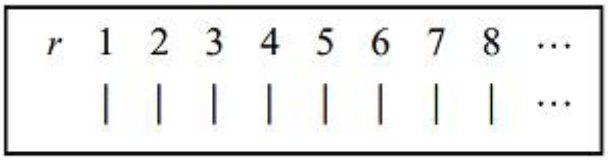

Рис. 3.4 «Реперна сітка алгоритму синтезу НП»

В таблиці 3.2 прийняті такі позначення: *r*- номер сходинки реперною сітки;

t_r - ступінь двійкового полінома CV_r, назвемо його координатним вектором (Coordinate Vector), лівий розряд якого дорівнює 1, а решта заповнені нулями, тобто: CV_r = 100…0.

Таблиця 3.2

Допоміжні числові параметри

r	1	2	3	4	5	6	7	8	…
t_r	1	3	7	15	31	63	127	255	…
Δ_r	1	2-3	4-7	8-15	16-31	32-63	64-127	128-255	…

Взаємозв'язок векторів CV_r з інтервалами Δ_r ступенів ТП для $r = 3$ можна простежити по таблиці 3.3, в якій змінні $b \in \{0, 1\}$ вибираються такими, щоб їх сукупна вага в поліномах дорівнював (за аксіомою 2) непарному числу.

Таблиця 3.3

Взаємозв’язок векторів

CV_3		1	0	0	0	0	0	0	0
		Структура тестуємого полінома							
Δ_3	4	1	*b*	*b*	*b*	1			
	5	1	*b*	*b*	*b*	*b*	1		
	6	1	*b*	*b*	*b*	*b*	*b*	1	
	7	1	*b*	*b*	*b*	*b*	*b*	*b*	1

Твердження. Необхідною умовою незвідності двійкового полінома ступеня *n* є виконання обов’язкового порівняння(співвідношення):

$$1\ (0)^{2^n-1} \equiv 1\ (\bmod f_n),\ n \geq 2. \qquad (3.6)$$

Представимо формулу (3.5) в наступному виді:

$$(10)^{L_n^{\sim}} = 1\ (\bmod f_n) \qquad (3.7)$$

Рівняння (3.5) і (3.6) є аналогами аксіоми 7. Ліва компонента $(10)^{L_n^{\sim}}$ порівняння (5) являє собою координатний вектор: CV_n = 100…0 } $2^n - 1$ біт

У свою чергу бінарний вектор, який відповідає порядку $L_n^{\sim}$ складається виключно з n одиниць. Назвемо цей вектор вектором одиниць (Unit Vector): UV_n = *11…11* } $2^n - 1$ біт. Десяткове значення вектора CV_n на одиницю більше значення вектора UV_n. Тому якщо дотримується умова (3.5), а саме $L_n \mid L_n^{\sim} = (2^n - 1)$, то тим самим підтверджується і порівняння (3.6). На цьому закінчується доказ твердження.

Відобразимо реперну сітку, відповідну поліному f_n, вектором $1^{[n]}$, що містить n одиниць, тобто нехай $1^{[n]}$ = 11…11 } n біт. Кожна r – на одиниця в $1^{[n]}$ символізує координатний вектор CV_r. Закон зміни порядків нульових розрядів векторів CV_r можна легко встановити, аналізуючи дані середнього рядка в таблиці 2, а саме:

$t_r = 2 \cdot t_{r+1}$, $t_0 = 0$, $r = \underline{1, n}$.

Нехай $S_r = Res(CV_r)_f$ - залишок координатного вектора CV_r по модулю полінома *f*. Запропонований алгоритм становить фундаментальну основу тестування двійкових поліномів на незвідність, яке зводиться до простих рекурентних обчислень: $S_r = Res(S_{r-1} \cdot s_r)_f$, $S_0 = 1$, $s_r = S_{r-1}$ 0, $r = r = \underline{1, n}$, де

s_r – розширений (додатковий нуль) залишок координатного вектора CV_{r-1}. При досягненні індексом r останньої n – ї сходинки реперної сітки, якщо виявиться, що $S_n = 1$, то це буде означати, відповідно до твердження , виконання необхідних умов незвідності поліному, що тестується. Кінцевий алгоритм синтезу незвітних поліномів: $S_r = Res(CV_r)_f = Res(CV_{r-1}CV_{r-1}0)_f = Res(S_{r-1}S_{r-1}0)_f$.

реперну сітку, що відповідає поліному f_n ,, зобразимо вектором $1^{[n]}$ з n одиницями, тобто $1^{[n]}$ = 11…11 } n розрядів. Кожне r - по одному в $1^{[n]}$ символізує вектор координат CV_r. Закон впорядкування нульових розрядів векторів CV_r можна легко визначити, проаналізувавши дані середнього рядка табл. 2, а саме:

$t_r = 2 \cdot t_{r+1}$, $t_0 = 0$, $r = \underline{1, n}$.

Нехай $S_r = Res(CV_r)_f$ - залишок координатного вектора CV_r за модулем полінома *f*. Запропонований алгоритм є базовою основою для перевірки бінарних поліномів на незвідність, яка зводиться до простих рекурентних обчислень:

$S_r = Res(S_{r-1} \cdot s_r)_f,\ S_0 = 1,\ s_r = S_{r-1}\ 0,\ r = r = \underline{1, n},$

де s_r - розширений (додатковий нульовий) залишок вектора координат CV_{r-1}. Якщо індекс *r* досягає останнього *n*-го кроку опорної сітки і виявляється, що $S_n = 1$, то згідно з твердженням це означає виконання необхідних умов незвідності полінома, що перевіряється. Кінцевий алгоритм синтезу незвідних поліномів:

$S_r = Res(CV_r)_f = Res(CV_{r-1}CV_{r-1}0)_f = Res(S_{r-1}S_{r-1}0)_f$

Приклад. Виберемо в якості тестованого поліному один з перевірених незвідних поліномів четверної ступені $f_4 = 10011$.

Напишемо одиничний вектор $f_4 = 10011$, виділивши в ньому жирним тоном перших три молодший біти, тому що їм спільно відповідають сім нулів, тобто вони утворюють вектор 10000000. Двох молодших жирних одиничок буде недостатньо, оскільки вони породжують чотирьох бітний вектор 1000, недостатній для обрахування остатку п’яти розрядним тестовим поліномом (ТП).

Напишемо для прикладу одиничний вектор $f_4 = 10011$, ньому трьом молодшим бітам відповідають сім нулів, тобто вони утворюють вектор 10000000. Зазначимо, що двох молодших бітів буде недостатньо, вектор який вони породжують буде недостатньої довжини, а саме чотирьох бітний вектор 1000, для можливості обрахування остатку від п’яти розрядного тестового поліномому (ТП).

Таблиця 3.4

Знаходження залишку Res_3

1	0	0	0	0	0	0	0
1	**0**	**0**	**1**	**1**		**ТП**	
			1	1	0	0	0
			1	**0**	**0**	**1**	**1**
				1	0	1	1

Залишок $\underline{Res}_4$, утворений четвертою (нежирною) одиничкою, буде дорівнювати Res_3, праворуч від якого дописується один нулик, тобто $\underline{Res}_4 = 10110$.

Перемноживши Res_3 і $\underline{Res}_4$ отримаємо $Res_4 = Res_3 \cdot \underline{Res}_4$.

Таблиця 3.5

Знаходження залишку $\underline{Res}_4$

		x	1	0	1	1	0
			1	0	1	1	
			1	0	1	1	0
		1	0	1	1		
1	0	1	1				
1	0	0	0	1	0	1	0

Тепер потрібно Res_4 привести до залишку по модулю ТП = 10011, отримаємо

Таблиця 3.5

Знаходження залишку Res_4

1	0	0	0	1	0	1	0
1	0	0	1	1			
			1	0	0	1	0
			1	0	0	1	1
							1

Таким чином, якщо ТП є НП, то залишок по модулю НП на n кроці обчислення буде рівним Res = 1.

Таблиця 3.6

Реперна сітка f = 10011

S_1 = 10;
S_2 = 1000;
S_3 = 1011;
S_4 = 1;

Перевіримо алгоритм на практиці вибравши для тестування перевірений незвідний поліном 12-го ступеня f_{12} 1000000001111. Значення залишку S_r векторів CV_r по модулю f_{12} зведені в таблицю 3.7.

Таблиця 3.7

Реперна сітка поліному f = 1000000001111

S_1 = 10;	S_5 = 101010011110;	S_9 = 110111111100;
S_2 = 1000;	S_6 = 110101111101;	S_{10} = 110100000100;
S_3 = 10000000;	S_7 = 110101111110;	S_{11} = 11111100010;
S_4 = 1111000;	S_8 = 110101110100;	S_{12} = 1

Той факт, що залишок S_{12} виявився рівним 1, є свідченням виконання необхідних умов незвідності полінома.

РОЗДІЛ 4
АПАРАТНА ЧАСТИНА

4.1 Arduino Uno

Arduino Uno - це мікроконтролерна плата на базі 8-розрядного мікроконтролера ATmega328P. Поряд з ATmega328P до її складу входять інші компоненти, такі як кварцовий генератор, послідовний зв'язок, стабілізатор напруги тощо, для підтримки роботи мікроконтролера. Arduino Uno має 14 цифрових входів/виходів (з яких 6 можуть бути використані як ШІМ-виходи), 6 аналогових входів, USB-роз'єм, роз'єм живлення, заголовок ICSP і кнопку скидання.

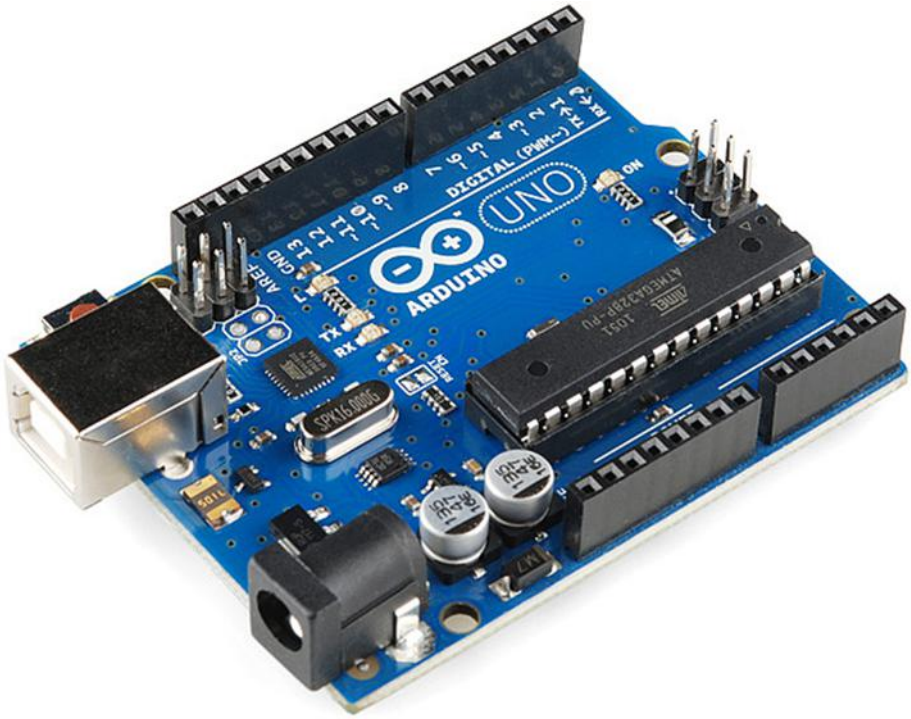

Рис. 4.1 «Arduino Uno»

Технічні характеристики

МІКРОКОНТРОЛЕР - ATmega328P

РОБОЧА НАПРУГА - 5В

ВХІДНА НАПРУГА (РЕКОМЕНДОВАНА) - 7-12В

ВХІДНА НАПРУГА (ГРАНИЧНЕ) - 6-20В

ЦИФРОВІ ВИВОДИ - 14 (з них 6 забезпечують ШІМ-вихід)

ШІМ ЦИФРОВІ ВХОДИ/ВИХОДИ - 6

АНАЛОГОВІ ВХІДНІ ВИВОДИ - 6

СТРУМ ПОСТІЙНОГО ТИПУ НА КЛЮЧ ВХОДУ/ВИХОДУ -20 мА

СТРУМ ПОСТІЙНОГО ТИПУ НА ПІН 3,3 В - 50 мА

FLASH ПАМ'ЯТЬ - 32 КБ (ATmega328P), з яких 0,5 КБ використовується завантажувачем

SRAM - 2 КБ (ATmega328P)

EEPROM - 1 КБ (ATmega328P)

CLOCK SPEED - 16 МГц

LED_BUILTIN - 13

ДОВЖИНА - 68.6 мм

ШИРИНА - 53.4 мм

вага - 25 г

Зображена на рис 4.1. розпіновка універсальної електронної плати Uno 14 цифрових виводів вводу/виводу можуть бути використані як виводи вводу або виводу за допомогою функцій pinMode(), digitalRead() і digitalWrite() в програмуванні Arduino. Кожен вивід працює при напрузі 5 В і може подавати або приймати струм максимум 40 мА, а також має внутрішній підтягуючий резистор 20-50 кОм, який за замовчуванням відключений. З цих 14 виводів деякі мають специфічні функції, які перераховані нижче:

- Послідовні виводи 0 (Rx) і 1 (Tx): Виводи Rx та Tx використовуються для прийому та передачі послідовних даних TTL. Вони з'єднані з відповідною мікросхемою ATmega328P USB to TTL.
- Виводи зовнішнього переривання 2 і 3: Ці виводи можуть бути налаштовані на спрацьовування переривання по низькому значенню, наростаючому або спадаючому фронту або зміні значення.
- Виводи ШІМ 3, 5, 6, 9 і 11: Ці виводи забезпечують 8-розрядний ШІМ-вихід за допомогою функції analogWrite().
- Виводи SPI 10 (SS), 11 (MOSI), 12 (MISO) і 13 (SCK): Ці виводи використовуються для зв'язку SPI.

- Вбудований світлодіодний вивід 13: Цей вивід з'єднаний з вбудованим світлодіодом, коли вивід 13 знаходиться на високому рівні - світлодіод світиться, а коли на низькому рівні - не світиться.
- Поряд з 14 цифровими виводами, є 6 аналогових вхідних виводів, кожен з яких забезпечує 10 біт роздільної здатності, тобто 1024 різних значень. Вони вимірюють від 0 до 5 вольт, але ця межа може бути збільшена за рахунок використання виводу AREF з аналоговою функцією Reference().
- Аналогові виводи 4 (SDA) і 5 (SCA) також використовуються для TWI-зв'язку за допомогою бібліотеки Wire.

Arduino Uno також має декілька інших виводів:

- AREF: Використовується для забезпечення опорної напруги для аналогових входів за допомогою функції analogReference().
- Reset Pin: Переведення цього виводу в низький рівень скидає мікроконтролер.

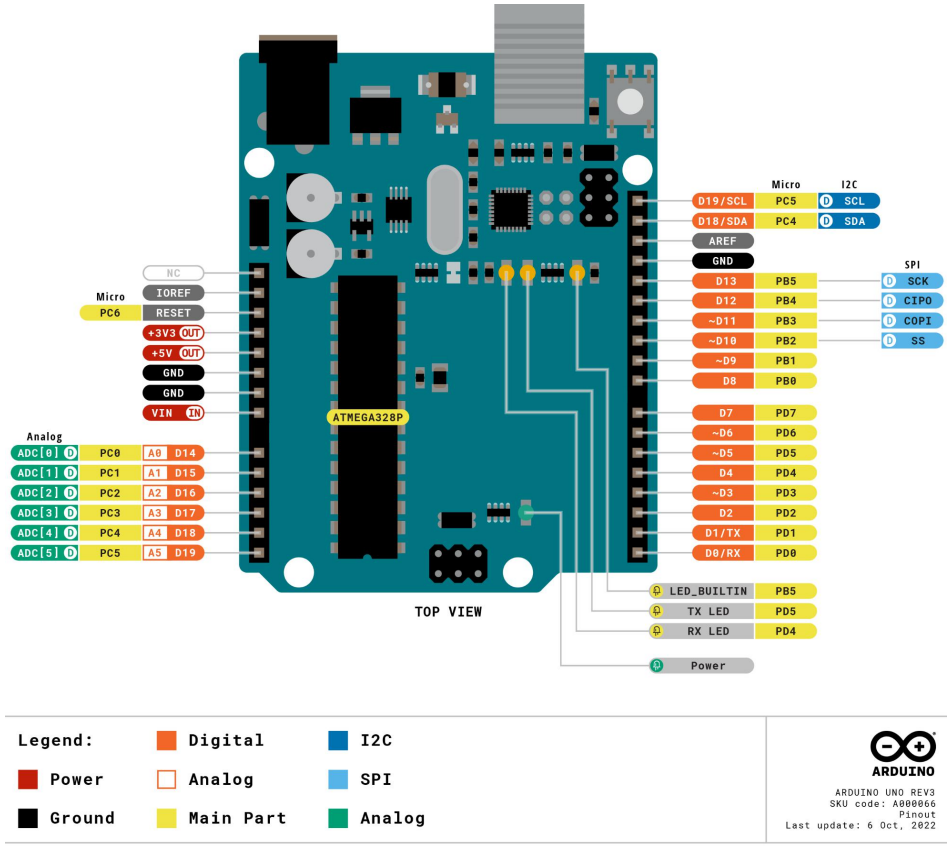

Рис. 4.2 «»

4.1.1 Комунікація

Arduino можна використовувати для зв'язку з комп'ютером, іншою платою Arduino або іншими мікроконтролерами. Мікроконтролер ATmega328P забезпечує послідовний UART-TTL зв'язок (5 В), який може здійснюватися через цифровий вивід 0 (Rx) і цифровий вивід 1 (Tx). ATmega16U2 на платі передає цей послідовний зв'язок через USB і виглядає як віртуальний порт для програмного забезпечення на комп'ютері. Прошивка ATmega16U2 використовує стандартні драйвери USB COM, тому зовнішній драйвер не потрібен. Однак, в операційній системі Windows потрібен файл .inf. Програмне забезпечення Arduino включає в себе послідовний монітор, який дозволяє відправляти прості текстові дані на плату Arduino і з неї. На платі Arduino є два світлодіоди, RX і TX, які блимають, коли дані передаються на комп'ютер через мікросхему USB-to-serial і USB-з'єднання (не тоді, коли послідовний зв'язок на контактах 0 і 1). Бібліотека SoftwareSerial дозволяє здійснювати послідовний зв'язок через будь-який з цифрових виводів Uno. ATmega328P також підтримує I2C (TWI) і SPI зв'язок. Програмне забезпечення Arduino включає в себе бібліотеку Wire для спрощення використання шини I2C.

4.1.2 ATmega238P

ATmega48A/PA/88A/PA/168A/PA/328/P - це малопотужний CMOS 8-розрядний мікроконтролер на базі AVR розширеної RISC-архітектури. Виконуючи потужні інструкції за один такт, мікроконтролери ATmega48A/PA/88A/PA/168A/PA/328/P досягає продуктивності, що наближається до 1 MIPS на МГц, що дозволяє розробнику системи дозволяє розробникам систем оптимізувати енергоспоживання по відношенню до швидкості обробки даних.

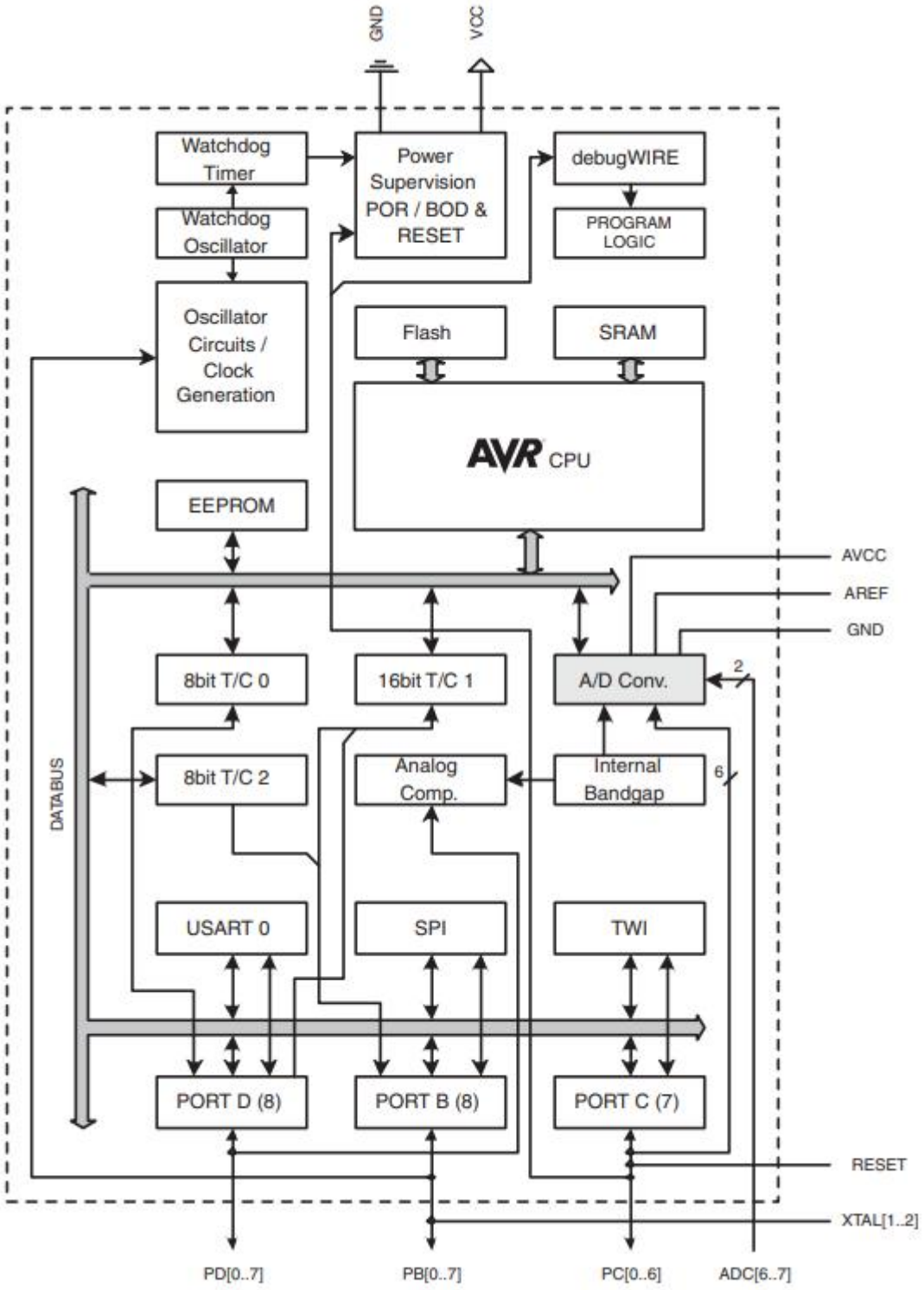

Рис. 4.3 «»

Ядро AVR поєднує в собі багатий набір інструкцій з 32 робочими регістрами загального призначення. Всі 32 регістри безпосередньо підключені до арифметико-логічного пристрою (ALU), що дозволяє отримати доступ до двох незалежних регістрів в одній до двох незалежних регістрів в одній інструкції, що виконується за

один такт. Отримана архітектура є більш ефективною з точки зору коду, досягаючи при цьому пропускної здатності до десяти разів більшої, ніж у звичайних мікроконтролерів CISC.

ATmega48A/PA/88A/PA/168A/PA/328/P має наступні характеристики: 4К/8Кбайт внутрішньосистемної програмованої флеш-пам'яті з можливістю читання і запису, 256/512/512/1Кбайт EEPROM, 512/1К/1К/2Кбайт SRAM, 23 лінії вводу/виводу загального призначення, 32 робочих регістра загального призначення, три гнучких таймера/лічильника з режимами порівняння режимами порівняння, внутрішні та зовнішні переривання, послідовний програмований інтерфейс USART, байт-орієнтований 2-провідний послідовний інтерфейс, послідовний порт послідовний порт SPI, 6-канальний 10-розрядний АЦП (8 каналів в корпусах TQFP і QFN/MLF), програмований Watchdog таймер з внутрішнім генератором і п'ять програмних режимів енергозбереження. Режим простою зупиняє роботу центрального процесора дозволяючи ОЗП, таймеру/лічильникам, USART, 2-провідному послідовному інтерфейсу, порту SPI та системі переривань продовжувати роботу. Режим Power-down зберігає вміст регістрів, але заморожує генератор, відключаючи всі інші функції мікросхеми до тих пір, поки не функції мікросхеми до наступного переривання або апаратного скидання. У режимі енергозбереження асинхронний таймер продовжує асинхронний таймер продовжує працювати, дозволяючи користувачеві підтримувати базу таймера, поки решта пристрою перебуває в режимі сну. Режим зменшення шуму АЦП зупиняє роботу центрального процесора і всіх модулів вводу/виводу, крім асинхронного таймера і АЦП, для мінімізації шуму перемикання під час АЦП перетворень. У режимі очікування кристалічний/резонаторний генератор працює, в той час як решта пристрою перебуває в режимі сну. Це забезпечує дуже швидкий запуск в поєднанні з низьким енергоспоживанням.

Atmel® пропонує бібліотеку QTouch® для вбудовування ємнісних сенсорних кнопок, повзунків і коліщаток в мікроконтролери AVR®. Запатентована технологія збору сигналу заряду-переносу забезпечує надійне зчитування і включає в себе

повністю з повним відображенням сигналів сенсорних клавіш і включає технологію придушення сусідніх клавіш (AKS™) для однозначного виявлення ключових подій. виявлення ключових подій. Простий у використанні набір інструментів QTouch Suite дозволяє досліджувати, розробляти і налагоджувати власні сенсорні додатки.

Пристрій виготовлено з використанням технології енергонезалежної пам'яті Atmel високої щільності. Вбудована флеш-пам'ять ISP Flash дозволяє перепрограмувати програмну пам'ять в системі через послідовний інтерфейс SPI, за допомогою звичайного програматора енергонезалежної пам'яті або за допомогою вбудованої завантажувальної програми, що працює в ядрі AVR. Завантажувальна програма може використовувати будь-який інтерфейс для завантаження прикладної програми в пам'ять Application Flash. Програмне забезпечення в розділі Boot Flash буде продовжувати працювати, поки оновлюється прикладна флеш-пам'ять, забезпечуючи справжній режим читання під час запису. Об'єднавши 8-розрядний RISC-процесор з внутрішньосистемною самопрограмованою флеш-пам'яттю на монолітному кристалі, мікросхеми Atmel.

ATmega48A/PA/88A/PA/168A/PA/328/P - це потужні мікроконтролери, які забезпечують гнучкі та економічно ефективні рішення для багатьох вбудованих систем керування.

ATmega48A/PA/88A/PA/168A/PA/328/P AVR підтримується повним набором засобів розробки програм і систем, включаючи: Компілятори C, макроасемблери, відладчики/симулятори програм, внутрішньосхемні емулятори та оціночні набори.

4.2 Технологія бездротового зв'язку

Bluetooth - технологія бездротового зв'язку, розроблена у 1998 році групою компаній: Ericsson, IBM, Intel, Nokia, Toshiba. Наразі розвиток Bluetooth здійснюється під керівництвом Bluetooth SIG (Special Interest Group), до якої також входять Lucent, Microsoft та інші компанії, що займаються мережевими технологіями. Основним призначенням Bluetooth є забезпечення економічного (з точки зору енергоспоживання) та дешевого бездротового зв'язку між різними типами електронних пристроїв, таких як мобільні телефони та аксесуари, ноутбуки

та настільні комп'ютери, принтери та інші. Крім того, велика увага приділяється компактності електронних компонентів, щоб Bluetooth можна було використовувати в невеликих пристроях розміром з наручний годинник.

Інтерфейс Bluetooth дозволяє передавати як голос (зі швидкістю 64 Кбіт/с), так і дані. Для передачі даних можуть використовуватися асиметричний (721 Кбіт/с в один бік та 57,6 Кбіт/с в інший) та симетричний (432,6 Кбіт/с в обидва боки) способи. Трансивер (Bluetooth-чіп) працює на частоті 2,4 ГГц і забезпечує зв'язок в радіусі 10 або 100 метрів. Різниця у відстанях, звичайно, велика, але з'єднання в межах 10 метрів дозволяє зберегти низьке енергоспоживання, компактні розміри і відносно низьку вартість компонентів. Наприклад, малопотужний передавач споживає всього 0,3 мА в режимі очікування і в середньому 30 мА при обміні інформацією. Стандарт Bluetooth передбачає шифрування переданих даних ключем з ефективною довжиною від 8 до 128 біт і можливість вибору між односторонньою і двосторонньою аутентифікацією. Крім шифрування на рівні протоколу, може використовуватися шифрування на програмному рівні.

Технологія Bluetooth працює за принципом FHSS (frequency-hopping spread spectrum). Якщо коротко, то передавач розбиває дані на пакети і передає їх за псевдовипадковим алгоритмом стрибкоподібної зміни частоти (1600 разів на секунду) або за шаблоном, що складається з 79 підчастот. "Розуміти" один одного можуть тільки пристрої, налаштовані на одну і ту ж схему передачі - для сторонніх пристроїв передана інформація є звичайним шумом. Найважливішим структурним елементом мережі Bluetooth є так званий "піконет" - група від 2 до 8 пристроїв, які працюють за однаковою схемою. У кожному піконеті один пристрій працює як активний (головний) учасник, а інші - як пасивні (підлеглі) учасники. Активний пристрій визначає шаблон, за яким працюють всі пасивні пристрої в його рісо-мережі, і синхронізує свою роботу. Стандарт Bluetooth передбачає з'єднання незалежних, а також несинхронізованих рісо-мереж (до 10) у так званий "скатернет" (англ. to scatter звучить як "розкидати"). Для цього кожна пара однорангових мереж повинна мати принаймні один спільний пристрій, який є активним в одній і пасивним в іншій. Таким чином, в межах однієї розсіяної мережі з інтерфейсом

Bluetooth може бути підключено максимум 71 пристрій одночасно, але ніхто не обмежує використання для тривалого зв'язку пристроїв воріт, які мають спільний інтернет.

Діапазон частот Bluetooth є безліцензійним у більшості країн, але у Франції, Іспанії та Японії необхідно використовувати інші частоти, ніж ті, що згадані вище, через законодавчі обмеження.

4.2.1 Bluetooth

Загалом, технологія Bluetooth проста у використанні, легка в налаштуванні і відносно дешева. Bluetooth є гнучким інструментом, який може бути використаний у багатьох різних рішеннях і проектах.

Кожен пристрій має бути оснащений мікрочіпом (трансивером), який здійснює передачу та прийом у частоті 2,4 ГГц, яка доступна в усьому світі (з деякими варіаціями пропускної здатності в різних країнах). Крім інформації, доступні три канали голосового зв'язку.

Bluetooth працюють між частотами 2,4 ГГц і 2,4835 ГГц. Кожна смуга має ширину 1 МГц, і кожна смуга розташована з інтервалом в 1 МГц. У стандартному Bluetooth є 79 смуг і є захисні смуги 2 МГц в нижній частині частотного спектру і 3,5 МГц у верхній частині частотного спектру.

Пристрої з підтримкою Bluetooth працюють у необмеженому діапазоні 2,4 гігагерца (ГГц) у промисловому, науковому та медичному (ISM) діапазоні. Діапазон ISM знаходиться в межах від 2,400 ГГц до 2,483 ГГц. Пристрої з підтримкою BWT використовують сімдесят дев'ять 1-мегагерцових частот (від 2,402 до 2,480 ГГц) в діапазоні ISM, як показано на малюнку 1. Пристрої з підтримкою Bluetooth використовують техніку, яка називається стрибкоподібною зміною частоти, щоб мінімізувати підслуховування і перешкоди від інших мереж, які використовують діапазон ISM(Доступний частотний діапазон). При скачкоподібній перебудові частоти дані діляться на невеликі частини, які називаються пакетами. Передавач і приймач обмінюються пакетом даних на одній частоті, а потім переходять на іншу частоту для обміну іншим пакетом. Вони повторюють цей процес до тих пір, поки всі дані не будуть передані.

4.2.2 Пошук пристроїв

Процес виявлення в Bluetooth-пристрої відносно швидкий і простий. Один пристрій, сподіваючись встановити з'єднання, розсилає "пакети запитів" у різні канали з певною швидкістю. Другий пристрій, який вважається виявленим, сканує канали з іншою швидкістю в пошуках пакетів запитів. Коли другий пристрій отримує пакет запиту, він відповідає пакетом синхронізації зі скачкоподібною перебудовою частоти (FHS), який містить інформацію, необхідну для встановлення з'єднання. Перший пристрій отримує пакет FHS, коли він повертається на відповідний канал і з'єднує пристрої. Після з'єднання пристрої можуть обмінюватися даними і синхронно переходити на різні канали для ефективного використання пропускної здатності.

4.2.3 Передачі інформації

Стандартний Bluetooth використовує скачкоподібну зміну частоти і "комутацію пакетів" для передачі інформації від одного кінцевого користувача до іншого. "Комутація пакетів" - це метод зв'язку, при якому відправник передає свої дані багатьма невеликими пакетами з порядковим номером. Кожен пакет знаходить найшвидший шлях до одержувача, а одержувач повторно впорядковує пакети, як тільки вони всі прибувають.

Для синхронізації існує один пристрій, який є головним, що диктує такт синхронізації, а всі інші пристрої слідують за головним тактовим генератором та схемою стрибкоподібної перебудови частоти для зв'язку. Метод частотної модуляції та комутації пакетів означає, що дані від відправника розбиваються на пакети встановленої кількості байт і кожен пакет передається в одному з 79 діапазонів частот. Процес стрибкоподібної зміни частоти забезпечує ефективне використання смуги пропускання і розподіляє трафік по всій смузі.

4.2.4 Безпека та завадостійкість блютуз

Безпека bluetooth є складною, але в основному прозорою і простою для користувачів. bluetooth використовує три типи механізмів безпеки: аутентифікація, авторизація та шифрування.

Аутентифікація перевіряє ідентичність пристрою BWT, який намагається з'єднатися з вашим пристроєм. Після завершення аутентифікації ваш bluetooth-пристрій надає (авторизує) іншому bluetooth-пристрою доступ до певного сервісу.

Шифрування переводить дані у формат, який може бути прочитаний тільки іншим пристроєм з таким же ключем шифрування. Реалізація цих механізмів відбувається на 3-х рівнях (режимах) безпеки:

Режим 1 - Без захисту; будь-хто може користуватися пристроєм. Цей режим використовується за замовчуванням для загальнодоступних пристроїв, таких як принтери.

Режим 2-У цьому режимі дозвіл на доступ до пристрою залежить від служби (служб), які ви дозволили (безпека на рівні служби). Наприклад, за допомогою КПК ви можете дозволити іншому пристрою обмінюватися електронними візитними картками і заборонити йому доступ до контактної інформації та записів календаря.

Режим 3 - У цьому режимі пристрої повинні бути з'єднані в пару, перш ніж вони зможуть встановити з'єднання та передавати дані (захист на рівні з'єднання).

Пристрої bluetooth випадковим чином перемикаються між частотами до 1600 разів на секунду - набагато швидше, ніж інші типи пристроїв, які використовують діапазон ISM. Це означає, що якщо інший пристрій, наприклад, бездротовий телефон, що працює на частоті 2,4 ГГц, створює перешкоди для мережі Bluetooth на певній частоті, перешкоди тривають лише протягом 1/1600 секунди, поки пристрої Bluetooth не переключаться на іншу частоту. Це робить мережі Bluetooth дуже стійкими до перешкод з боку інших пристроїв, що працюють на частоті 2,4 ГГц. Існує три класи радіопристроїв bluetooth, кожен з яких має різну максимальну дальність дії:

Клас 1 (100 метрів); Клас 2 (50 метрів); Клас 3 (10 метрів).

4.2.5 Bluetooth 4.0

У грудні 2009 консорціум Bluetooth SIG анонсував стандарт Bluetooth 4.0 для електронних пристроїв. Новий стандарт призначений для передачі коротких пакетів даних обсягом по 8-27 байт зі швидкістю 1 Мбіт/с. Для порівняння, Bluetooth 3.0,

розробка якого була завершена в квітні 2008, дає змогу передавати дані зі швидкістю до 24 Мбіт/с, але і призначений він для іншої сфери застосування Bluetooth 4.0 використовують в мініатюрних сенсорах, що розміщуються на тілі пацієнтів, у спортивному взутті, тренажерах тощо. Сенсори на базі нового стандарту можуть передавати різну інформацію з навколишнього світу — температуру, тиск, вологість, швидкість пересування і так далі — на різні пристрої контролю, включаючи мобільні телефони. За словами представників консорціуму, окремий стандарт був розроблений у зв'язку з тим, що Bluetooth 3.0 і більш ранні версії не в змозі забезпечити необхідний низький рівень енергоспоживання. Перший чип з одночасною підтримкою Bluetooth 4.0 і 3.0 випустив ST-Ericsson. У липні 2010 року специфікація була затверджена Bluetooth Special Interest Group.

4.3 Bluetooth Module HC-05

Bluetooth Module HC-05 - модуль широкого застосування для з'єднання пристроїв через Bluetooth-підключення. Модуль Bluetooth керується через UART, тож по суті є преобразователем UART-to-Bluetooth. На відміну від модуля HC-06 може працювати як в режимі Master так і в режимі Slave. Являє собою чип Bluetooth HC-05(06), розпаяний на платі, що містить DC-DC перетворювач напруги живлення і схему перетворення рівнів сигналів RX і TX. Сам модуль HC-05(06) живиться напругою 3.3 вольта, але завдяки перетворювачам, ви можете цю збірку під'єднати до пристрою, що працює від напруги 5 вольт.

HC-05 призначений для організації двостороннього зв'язку за протоколом Bluetooth. Bluetooth є популярним типом зв'язку між пристроями на коротких дистанціях. Особливістю передачі даних по Bluetooth є стійкість до широкосмугових перешкод. Цей модуль можна використовувати для керування роботами, освітленням та іншими пристроями на відстані.

Bluetooth HC-05 має можливість працювати у веденому і ведучому режимах. Завдяки цим двом режимам модуль може самостійно виявити і налагодити зв'язок з пристроєм. Модуль Bluetooth HC-05 спроектований на основі мікросхеми CSR BC417, яка підтримує зв'язок Bluetooth версії 2.0. Швидкість передачі даних модуля становить до 3 Мбіт/с.

Рис. 4.4 «»

Технічні характеристики модуля HC-05:

Чип Bluetooth: HC-05 (BC417143)

Діапазон частот радіозв'язку: 2,4-2,48 ГГц

Підтримувана швидкість передачі даних: 9600, 19200, 38400, 57600, 115200, 230400, 460800.

Потужність передачі: 0,25-2,5 мВт

Чутливість: -80 dBм

Напруга живлення: 3,3-5 В

Споживаний струм: 50 мА

Радіус дії: до 10 метрів

Інтерфейс: послідовний порт

Режими: master, slave

Робочий діапазон температур: -25...75 °C

Габарити: 27 x 13 x 2,2 мм.

Таблиця 4.1

HC-05 Конфігурація розводки виводів

Номер пін	Ім'я пін	Опис

1	Enable / Key	Цей вивід використовується для перемикання між режимом даних (низький рівень) і режимом команд AT (високий рівень). За замовчуванням він знаходиться в режимі даних
2	Vcc	Живить модуль. Підключіть до напруги живлення +5В
3	Ground	Контакт заземлення модуля, підключіть до заземлення системи.
4	TX – Transmitter	Передає послідовні дані. Все, що отримується через Bluetooth, буде видаватися цим контактом як послідовні дані.
5	RX – Receiver	Отримання послідовних даних. Кожні послідовні дані, передані на цей штифт, будуть транслюватися через Bluetooth
6	State	Вивід стану підключений до бортового світлодіоду, його можна використовувати як зворотній зв'язок для перевірки правильності роботи Bluetooth. ● коли модуль активний світлодіод блимає ● коли зв'язок встановлено - горить
7	LED	Показує стан модуля ● Блимає один раз на 2 секунди: Модуль перейшов у командний режим ● Повторне блимання: Очікування з'єднання в режимі передачі даних ● Блимає двічі за 1 сек: Успішне з'єднання в режимі передачі даних
8	Button	Використовується для керування контактом

		Key/Enable для перемикання між режимом передачі даних та режимом команд

Підключення блютуз модуля HC-05 до Arduino:

EN - вкл / викл, якщо подати сюди логічну одиницю (або просто 5В), то модуль вимкнеться, якщо логічний нуль (або просто не підключати цей пін) буде працювати. На деяких модулях для включення режиму АТ-команд, треба затиснути кнопку і подати живлення +5 В, а через 2-3 сек відпустити кнопку. На інших модулях (особливість прошивки) АТ-команди можна використовувати в режимі очікування (світлодіод блимає 1 раз в секунду) при натиснутій (і утримуваній) кнопці. Перед введенням команд, в моніторі посл. порту необхідно вибрати режим переведення рядка (CR + LF) і швидкість з'єднання 9600 бод.

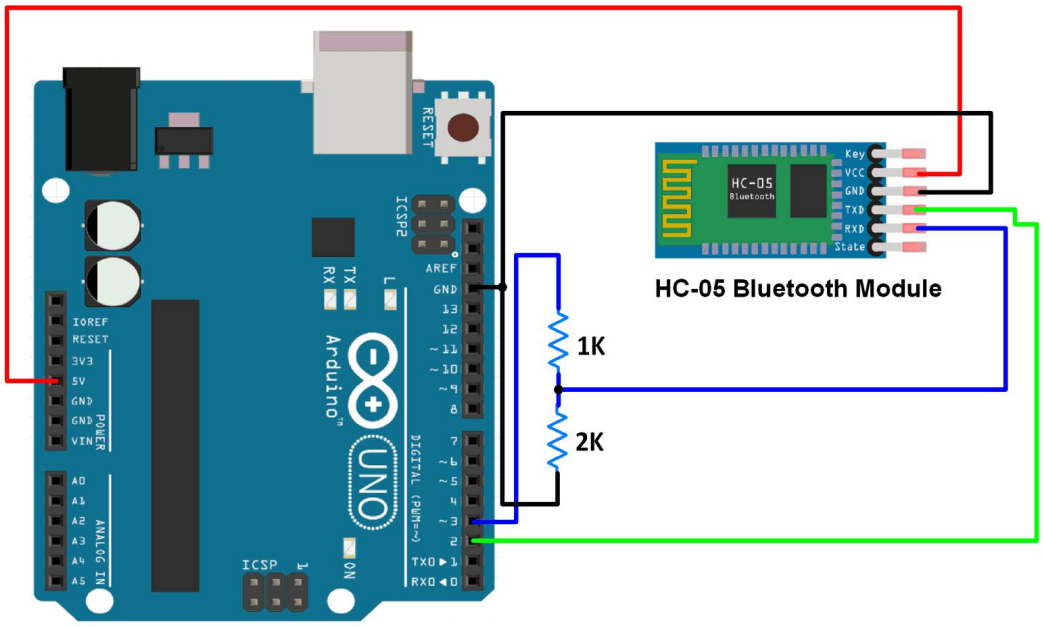

Рис. 4.5 «»

Bluetooth модуль HC-05 під'єднується по шині UART і здатний виконувати АТ-команди. АТ-команда - це рядок, що починається з букв "AT" (від англійського attention - "увага"). Модуль виконує команду, що надійшла, і відправляє назад відповідь (результат виконання команди), яка також є рядком. У Bluetooth модулях

HC-05 кожна команда (як і відповідь) повинна закінчуватися символами перекладу рядка "\r\n".

Підключення:

Керувати Bluetooth модулем HC-05 можна або з комп'ютера, або через мікроконтролер, наприклад, Arduino. Вивід RX модуля підключається до виводу TX, а вивід TX модуля підключається до виводу RX того пристрою, з якого він буде управлятися.

Для під'єднання модуля до комп'ютера (без мікроконтролерів) знадобиться адаптер USB-UART, або адаптер RS232-UART, або програматор із виводами TX RX, замість адаптера USB-UART можна використати плату Arduino. Для надсилання команд у модуль потрібно встановити програму термінал. Одним з таких терміналів є вільно розповсюджувана програма Termite.

Для підключення модуля до Arduino можна скористатися апаратною або програмною шиною UART. У разі використання апаратної шини, модуль під'єднують до виводів TX і RX, зазначених на платі. У разі використання програмної шини, модуль під'єднується до призначуваних виводів TX і RX Arduino.

Налаштування:

Модуль Bluetooth HC-05 зможе приймати команди тільки якщо правильно налаштовані такі параметри послідовного порту:

Номер порту: Його можна дізнатися експериментально, вимкніть адаптер або Arduino, подивіться, які порти доступні. Підключіть адаптер або Arduino і знову подивіться, які порти доступні. Порт, що з'явився, і є той самий.

Швидкість передачі даних: У звичайному режимі Bluetooth-модуль HC-05 зберігає останню встановлену швидкість передачі даних, але за замовчуванням вона дорівнює 38400 біт/сек (рідко 9600 біт/сек).

Параметри передавання даних: Модуль зберігає останні встановлені параметри передавання даних. Значення параметрів за замовчуванням: кількість біт у пакеті - 8, розмір стопового біта = 1, без перевірки парності.

Текст, що передається: Потрібно встановити пункт "додавати символи CR & LF(NL)" - це символи перекладу рядка "\r\n", які Ви не зможете ставити самостійно в кінці AT-команд.

У разі використання Arduino, номер порту вказується у вкладці "Інструменти". Параметри передачі даних використовуються за замовчуванням. Для додавання символів NL & CR скористайтеся меню в правому нижньому кутку монітора послідовного порту.

Перевірка:

Після кожного під'єднання живлення або перезавантаження модуля, до того як надсилати команди, потрібно короткочасно натиснути на кнопку модуля. Якщо у модуля немає кнопки, то короткочасно подати високий рівень на вивід K. Після чого модуль залишиться у звичайному режимі, але буде сприймати AT-команди. Крім звичайного режиму, модуль може працювати в режимі AT-команд. Про те, як увійти в цей режим і чим він відрізняється від звичайного, розказано нижче, у розділі примітка.

Для перевірки зв'язку з Bluetooth модулем надішліть тестову команду AT (введіть текст AT і натисніть Enter). Якщо зв'язок встановлено коректно, то модуль відповість OK. Після цього можна надсилати інші AT-команди.

AT-команди:

Якщо в програмі термінал вказано додавати символи CR & LF або NL & CR, то символи "\r\n" у командах ставити не потрібно!

Команди можуть бути звичайними: AT+КОМАНДА\r\n, запитами: AT+КОМАНДА?\r\n, або установками: AT+КОМАНДА=ПАРАМЕТР(и)\r\n.

4.4 Display

1,3-дюймовий дисплей має розширення 240x240, 16-бітних повнокольорових пікселів і є IPS-дисплеєм, тому кольори чудово виглядають під кутом до 80 градусів від осі в будь-якому напрямку. Драйвер TFT (ST7789) дуже схожий на популярний ST7735, і бібліотека Arduino добре його підтримує.

Рис. 4.6 «»

Це 1,3-дюймовий рідкокристалічний дисплей IPS LCD високої чіткості. Його мікросхема драйвера - ST7789 та вихід SPI. Як зазначалося вище, контролер TFT-дисплея ST7789 працює тільки з напругою 3,3 В (лінії живлення і управління). На дисплейний модуль подається напруга 3,3 В (між VCC і GND), яка надходить з плати Arduino. Всі вихідні виводи плати Arduino UNO мають напругу 5В, підключення виводу 5В до TFT-дисплея ST7789 може призвести до пошкодження його контролера.

Для підключення Arduino до модуля дисплея я використовував дільник напруги для кожної лінії, тобто є 4 дільника напруги. Кожен дільник напруги складається з резисторів 2.2k і 3.3k, що дозволяє знизити напругу з 5В до 3В, чого цілком достатньо.

Якщо дисплейний модуль має вивід CS (Chip Select), то його слід підключити до цифрового виводу Arduino 10 через інший дільник напруги.

Отже, TFT дисплей ST7789 підключається до плати Arduino наступним чином (кожен через дільник напруги):

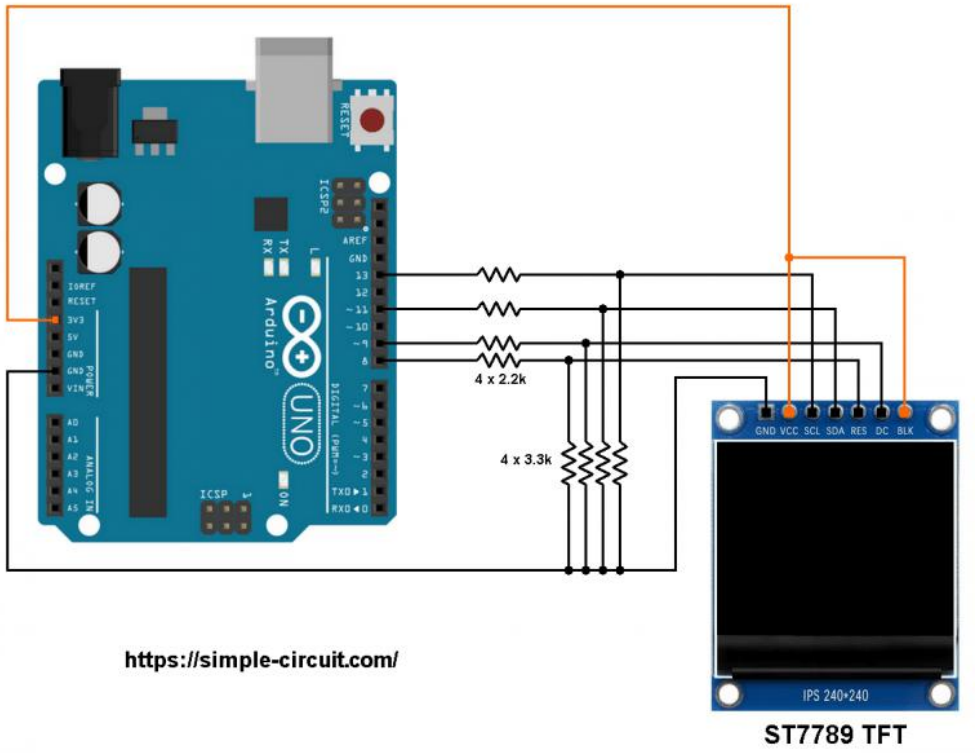

Рис. 4.7 «»

Вивід RST підключається до цифрового виводу Arduino 8,

вивід DC підключений до цифрового виводу Arduino 9,

вивід SDA підключений до цифрового виводу Arduino 11,

вивід SCL підключений до цифрового виводу Arduino 13.

Інші виводи підключаються наступним чином:

Вивід VCC підключений до виводу Arduino 3V3,

вивід GND підключений до виводу Arduino GND,

вивід BL (світлодіод) підключений до виводу Arduino 3V3 (необов'язково).

4.5 Інтерфейси (передачі даних)

Послідовний периферійний інтерфейс (Serial Peripheral Interface, SPI) - це специфікація синхронного послідовного інтерфейсу зв'язку, що використовується для зв'язку на короткі відстані, в першу чергу у вбудованих системах. Інтерфейс був розроблений компанією Motorola в середині 1980-х років і став стандартом де-факто. Типові застосування включають захищені цифрові карти і рідкокристалічні дисплеї.

Пристрої SPI взаємодіють в повнодуплексному режимі, використовуючи архітектуру "ведучий-ведений", як правило, з одним ведучим (хоча деякі пристрої

Atmel підтримують зміну ролей на льоту в залежності від зовнішнього (SS) виводу). Ведучий пристрій (контролер) ініціює кадри для читання і запису. Кілька підлеглих пристроїв можуть підтримуватися за допомогою вибору за допомогою окремих ліній вибору мікросхеми (CS), які іноді називають лініями вибору підлеглого (SS).

Іноді SPI називають чотирипровідною послідовною шиною, на відміну від трьох-, двох- та однопровідних послідовних шин. SPI можна точно описати як синхронний послідовний інтерфейс,[6] але він відрізняється від протоколу Synchronous Serial Interface (SSI), який також є чотирипровідним синхронним послідовним протоколом зв'язку. Протокол SSI використовує диференціальну сигналізацію і забезпечує лише один симплексний канал зв'язку. Для будь-якої заданої транзакції SPI - це зв'язок між одним ведучим і декількома веденими.

Інтерфейси SPI можуть мати тільки один головний і можуть мати один або кілька підвузлів. На малюнку 1 показано з'єднання SPI між головним і підвузлом.

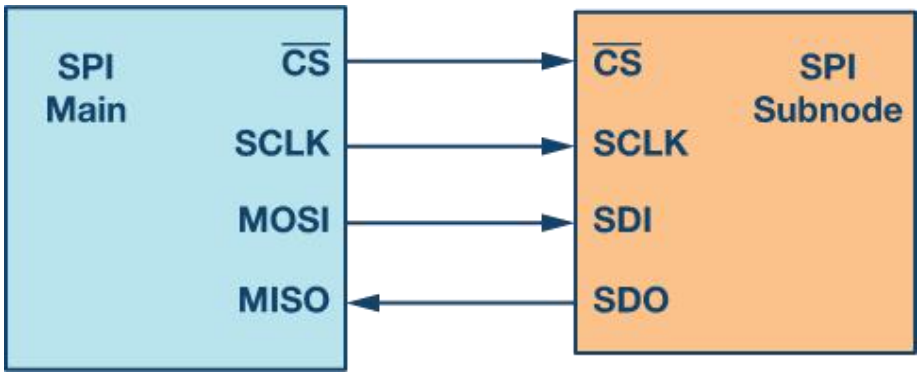

Рис. 4.8 «»

4-х провідні SPI пристрої мають чотири сигнали:

- Тактовий (SPI CLK, SCLK)
- вибір мікросхеми (CS)
- основний вихід, вхід підвузла (MOSI)
- основний вхід, вихід підвузла (MISO)

Пристрій, який генерує тактовий сигнал, називається головним. Дані, що передаються між головним і підвузлом, синхронізуються з тактовим сигналом, що генерується головним. Пристрої SPI підтримують набагато вищі тактові частоти в

порівнянні з інтерфейсами I2C. Користувачі повинні звернутися до специфікації тактової частоти інтерфейсу SPI в технічному паспорті продукту.

Сигнал вибору мікросхеми з основного використовується для вибору підвузла. Зазвичай це активний низький рівень сигналу, який піднімається до високого рівня, щоб від'єднати підвузол від шини SPI. Коли використовується декілька підвузлів, для кожного підвузла потрібен окремий сигнал вибору мікросхеми від головного. У цій статті сигнал вибору мікросхеми завжди є активним низьким сигналом.

MOSI і MISO - це лінії передачі даних. MOSI передає дані від головного до підвузла, а MISO передає дані від підвузла до головного.

Передача даних

Щоб розпочати SPI-зв'язок, головний пристрій повинен послати тактовий сигнал і вибрати підвузол, увімкнувши сигнал CS. Зазвичай вибір мікросхеми є активним низьким сигналом; отже, головний пристрій повинен послати логічний 0 на цей сигнал, щоб вибрати підвузол. SPI є повнодуплексним інтерфейсом; як головний, так і підвузол можуть одночасно відправляти дані по лініях MOSI і MISO відповідно. Під час SPI-зв'язку дані одночасно передаються (послідовно виводяться на шину MOSI/SDO) і приймаються (дані на шині (MISO/SDI) вибираються або зчитуються). Послідовний фронт тактового сигналу синхронізує зсув і вибірку даних. Інтерфейс SPI надає користувачеві гнучкість у виборі переднього або заднього фронту тактового сигналу для вибірки та/або зсуву даних. Будь ласка, зверніться до технічного паспорту пристрою, щоб визначити кількість бітів даних, що передаються за допомогою інтерфейсу SPI.

Полярність і фаза тактового генератора

У SPI основним може бути вибір полярності та фази тактового генератора. Біт CPOL встановлює полярність тактового сигналу під час простою. Стан простою визначається як період, коли CS є високим і переходить до низького рівня на початку передачі, і коли CS є низьким і переходить до високого рівня в кінці передачі. Біт CPHA вибирає фазу синхронізації. Залежно від біта CPHA, зростаючий або спадаючий фронт тактового сигналу використовується для вибірки та/або зсуву даних. Головний повинен вибрати полярність і фазу тактового сигналу відповідно

до вимог підвузла. Залежно від вибору бітів CPOL і CPHA, доступні чотири режими SPI. У таблиці 1 показано чотири режими SPI.

Таблиця 4.2

Режими SPI з CPOL та CPHA

SPI Mode	CPOL	CPHA	Полярність синхронізації	Фаза тактового генератора, що використовується для вибірки та/або зсуву даних
0	0	0	Logic low	Дані вибираються на зростаючому фронті і зсуваються на спадаючому фронті
1	0	1	Logic low	Дані, що відбираються на спадаючому фронті і зсуваються на зростаючому фронті
2	1	0	Logic high	Дані вибираються на зростаючому фронті і зсуваються на спадаючому фронті
3	1	1	Logic high	Дані, що відбираються на спадаючому фронті і зсуваються на зростаючому фронт

На рисунках 4.9 - 4.12 показано приклади обміну даними в чотирьох режимах SPI. У цих прикладах дані показані на лінії MOSI та MISO. Початок і кінець передачі позначено пунктирною зеленою лінією, фронт дискретизації позначено помаранчевим кольором, а фронт зсуву позначено синім кольором. Зверніть увагу, що ці малюнки наведені лише для ілюстрації. Для успішної передачі даних через SPI користувачі повинні звернутися до технічного паспорту виробу і переконатися, що часові характеристики для деталі дотримані.

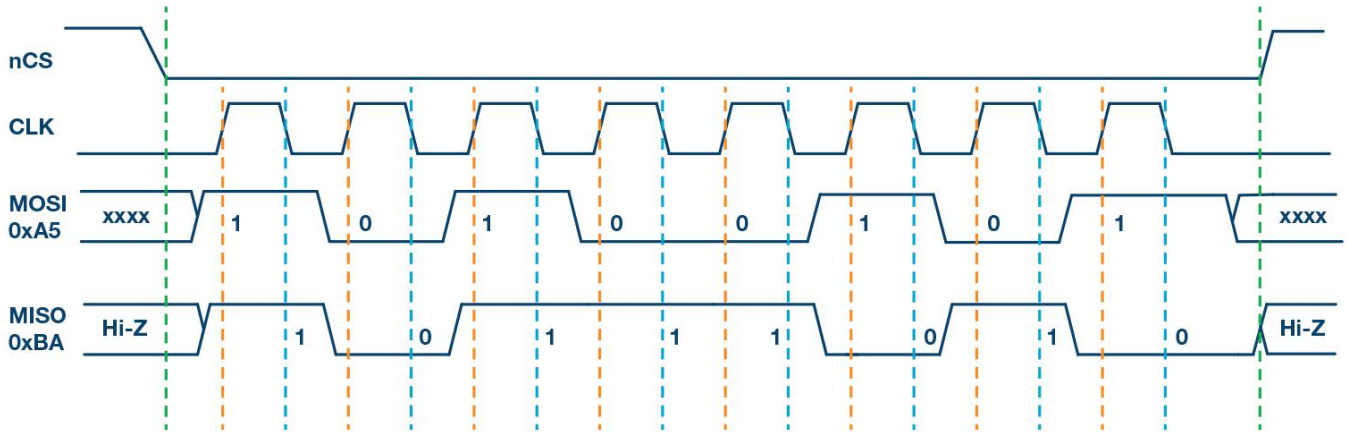

Рис. 4.9 «Режим SPI 0, CPOL = 0, CPHA = 0: стан холостого ходу CLK = низький, дані відбираються по передньому фронту і зсуваються по задньому фронту»

На рисунку 4.10 показана часова діаграма для режиму SPI 1. У цьому режимі полярність тактового сигналу дорівнює 0, що вказує на те, що в стані спокою рівень тактового сигналу низький. Фаза тактового сигналу в цьому режимі дорівнює 1, що вказує на те, що вибірка даних здійснюється по спадаючому фронту (показано помаранчевою пунктирною лінією), а зсув даних - по зростаючому фронту (показано синьою пунктирною лінією) тактового сигналу.

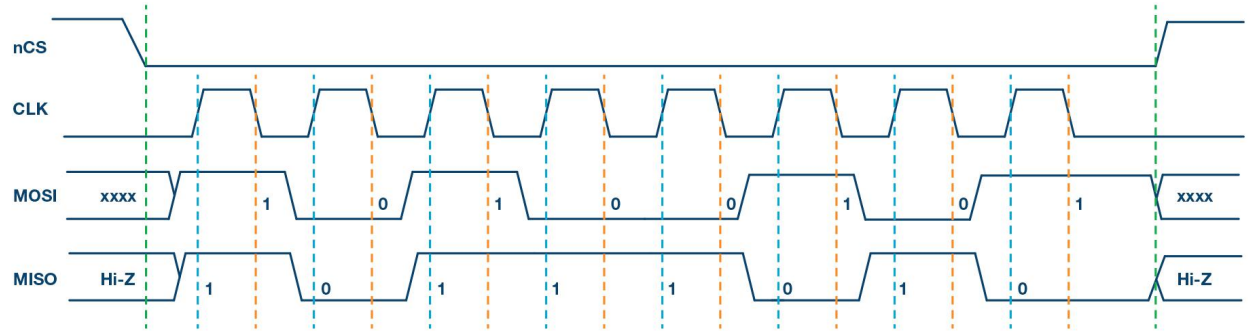

Рис. 4.10 «Режим SPI 1, CPOL = 0, CPHA = 1: стан холостого ходу CLK = низький, дані відбираються по спадаючому фронту і зсуваються по зростаючому фронту»

На рисунку 4.11 показана часова діаграма для режиму SPI 3. У цьому режимі полярність тактового сигналу дорівнює 1, що вказує на те, що стан спокою тактового сигналу є високим. Фаза тактового сигналу в цьому режимі дорівнює 1, що вказує на те, що вибірка даних здійснюється по спадаючому фронту (показано помаранчевою пунктирною лінією), а зсув даних - по зростаючому фронту (показано синьою пунктирною лінією) тактового сигналу.

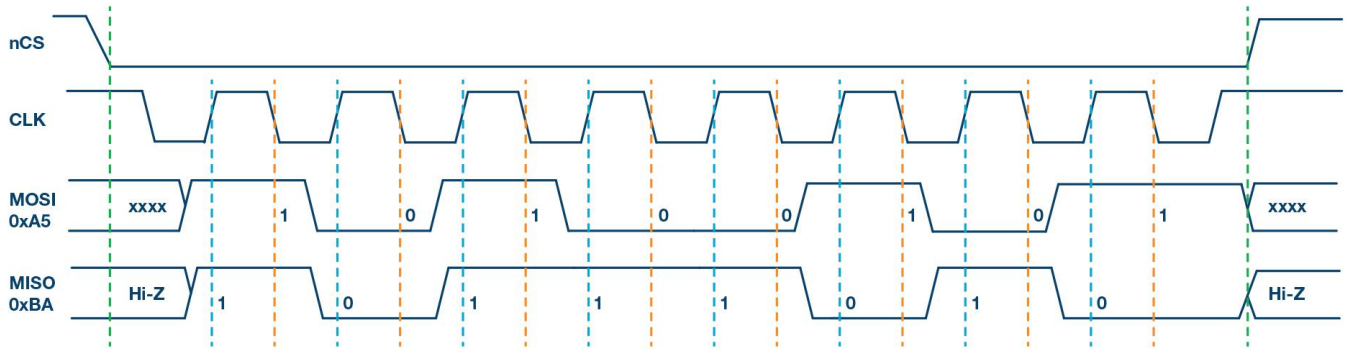

Рис. 4.11 «Режим SPI 3, CPOL = 1, CPHA = 1: стан холостого ходу CLK = високий, дані відбираються по спадаючому фронту і зсуваються по зростаючому фронту»

На рисунку 4.12 показана часова діаграма для режиму SPI 2. У цьому режимі полярність тактового сигналу дорівнює 1, що вказує на те, що стан спокою тактового сигналу є високим. Фаза тактового сигналу в цьому режимі дорівнює 0, що вказує на те, що вибірка даних здійснюється по передньому фронту (показано помаранчевою пунктирною лінією), а зсув даних - по задньому фронту (показано синьою пунктирною лінією) тактового сигналу.

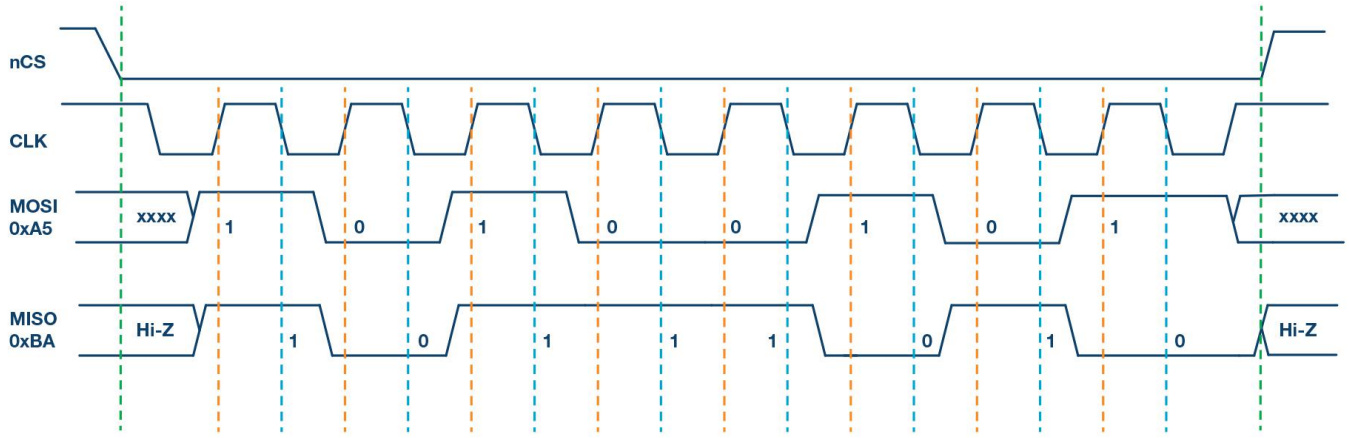

Рис. 4.12 «Режим SPI 2, CPOL = 1, CPHA = 0: стан холостого ходу CLK = високий, дані відбираються по передньому фронту і зсуваються по задньому фронту»

4.6 Апаратна реалізація

Апаратна реалізація програмного модуля включає в себе мікроконтролерну плату Arduino Uno на базі 8-розрядного мікроконтролера ATmega328P до якої підключено Bluetooth модуль HC-05, а також екран ST7789 (Рис. 4.13).

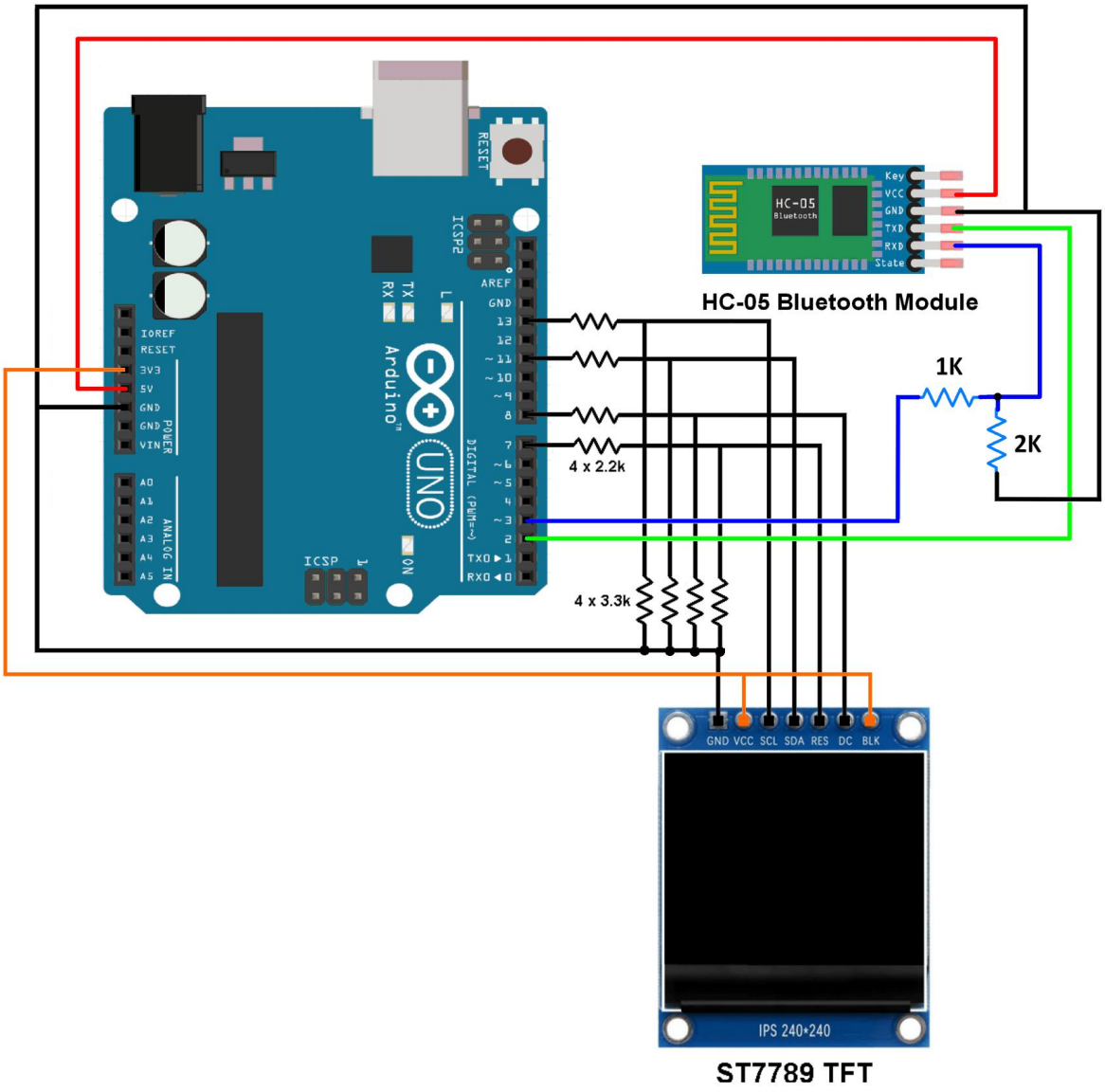

Рис. 4.13 «Апаратна реалізація»

ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

5.1 Алгоритм пересилання ключів за допомогою RSA

Алгоритм генерації відкритого і закритого ключів є найскладнішою частиною криптографії RSA. Генерацію ключів можна умовно розділити на декілька етапів:

1. Необхідно згенерувати два великих простих числа, p і q, за допомогою алгоритму перевірки на простоту перевірити їх. Для того, щоб ускладнити факторизацію, p і q повинні бути обрані випадковим чином, бути схожими за величиною, але відрізнятися за довжиною. Прості цілі числа можна ефективно знайти за допомогою тесту на простоту. Важливо зазначити, що значення p та q слід обов'язково тримати в таємниці.
2. Наступним кором буде обчислення модуля n за формулою:

$$n = p * q \qquad (5.1)$$

надалі n буде використовується одразу як спільний модуль як для двох ключів, а саме у відкритому так і для закритого.

3. Обчислити $\lambda(n)$, де λ - функція Ейлера. Оскільки $n = p * q$, то $\lambda(n) = НСК(\lambda(p), \lambda(q))$, а оскільки p і q прості, то $\lambda(n) = \varphi(p) = p - 1$, і аналогічно $\varphi(q) = q - 1$. Звідси $\lambda(n) = НСК(p - 1, q - 1)$. НСК можна обчислити за евклідовим алгоритмом, оскільки:

$$НСК(a, b) = \frac{|ab|}{НСД(a,b)} \qquad (5.2)$$

4. Тепер, коли ми маємо результат функції Ейлера для наших простих чисел, настав час визначити наш відкритий ключ. Відповідно до RSA, відкриті ключі складаються з простого числа e, а також модуля n. Число e повинно задовольняти умові:

$$1 < e < \lambda(n) \qquad (5.3)$$

і НСД(e, λ(n)) = 1, тобто e і λ(n) є взаємно простими числами. *e* може набувати будь-яких значень між 1 і значенням λ(n). Оскільки відкритий ключ надається відкрито, не так важливо, щоб e було випадковим числом. На практиці зазвичай встановлюється на рівні

$$e = 2^{16} + 1 = 65537, \qquad (5.4)$$

оскільки при випадковому виборі набагато більших чисел це робить шифрування набагато менш ефективним. Найменшим а також і найшвидшим можливим значення $e = 3$, але таке мале значення не можливо вважати безпечним за деяких умов використання алгоритму.

5. Визначимо d як $de = 1(mod\ \lambda(n))$, тобто d є модулем, оберненим до e за модулем $\lambda(n)$. Це означає: розв'язати для d рівняння $de \equiv 1/e(mod\ \lambda(n))$. d можна ефективно обчислити за допомогою розширеного алгоритму Евкліда, оскільки завдяки тому, що e та $\lambda(n)$ є взаємно простими, зазначене рівняння є формою тотожності Безу, де d є одним з коефіцієнтів.

Значення d тримається у секреті, як показник степені закритого ключа.

5.2 Шифрування RSA

Після того, як було згенеровано відкритий і закритий ключ ми можемо відправити відкритий ключ через незахищений канал зв’язку рис 5.1, отримавши відкритий ключ від Arduino ми можемо зашифрувати повідомлення M і відправити його закритим каналом зв’язку.

Рис. 5.1 «Отримання ключа відритим каналом зв’язку»

Шифрування відбувається наступним чином, для цього ми будемо використовувати відкритий ключ Key(e, n), який складається з двох чистин ступеня e та модуля n. Для початку необхідно перетворити повідомлення M (нерозшифрований відкритий текст) у ціле число m (розшифрований відкритий

текст), таке, що 0 ≤ m < n, використовуючи узгоджений оборотний протокол, відомий як схема доповнення.

Потім за допомогою формули (5.5) обчислює зашифрований текст *c*, використовуючи відкритий ключ

$$c \equiv m^e \ (mod\ n) \qquad (5.5)$$

Схематично шифрування зображено на наступному рис.5.2

Рис. 5.2 «Шифрування RSA»

Це операція виконується досить швидко, навіть для дуже великих чисел, використовуючи модульне піднесення до степеня. Наступним кроком зашифрований шифротекст передаємо для Arduino через Bluetooth.

5.3 Розшифрування RSA

Після отримання повідомлення з шифротекстом, яке було зашифроване відкритим ключем шифрування ми можемо його відновити рис.5.3. Для того, щоб виконати операцію розшифрування повідомлення m за допомогою свого закритого ключа Key(d, n), використовуючи показник степеня d, та модуль n необхідно виконати обрахунки за наступною формулою

$$m = c^d \ (mod\ n) \qquad (5.6)$$

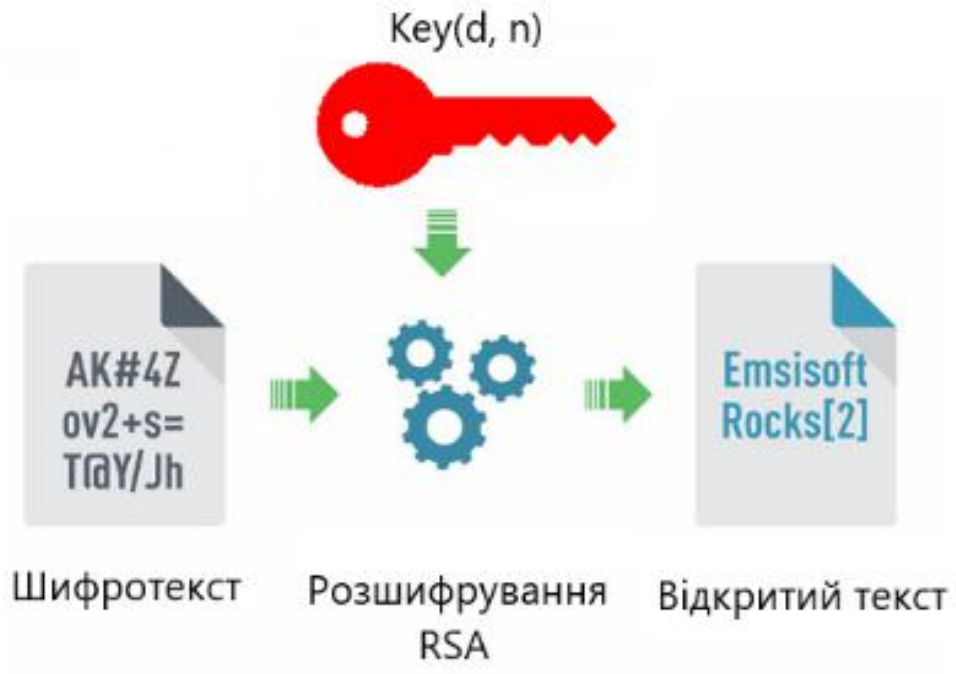

Рис. 5.3 «Розшифрування RSA»

5.4 Синхронізація Галуа-64

Важливим етапом перед початком передачі повідомлень між Arduino та пристроєм керування є синхронізація ключів шифру Галуа-64. Оскільки у двох сторін немає можливості безпечно передати початкові значення ключів, через те, що між ними, ще не створено закритого каналу зв’язку. Пристрої знаходяться на відстані і єдиний спосіб взаємодії через не захищений відкритий канал. Для того аби вирішити дану проблему використовується додатковий алгоритм шифрування, який має змогу встановити закритий канал зв’язку і безпечно передати необхідні значення для безпечної та повноцінної роботи потокового байт орієнтованого шифру.

У нашій ситуації таким шифром було вибрано RSA. Він дозволяє користувачам встановити закритий канал зв’язку і виконує повідомлення за допомогою відкритого ключа свого кореспондента, так що вони можуть бути розшифровані тільки за допомогою відповідного закритого ключа.

Незважаючи на можливість RSA безпечно працювати і створювати закритий зв'язок, на практиці ми не можемо його використовувати для шифрування всіх

повідомлень - це було б занадто неефективно, через математичну складність алгоритму і недостатньою для цього потужністю Arduino.

Тому RSA використовуємо тільки на етапі передачі і синхронізації ключів шифру Галуа-64. Схема синхронізації виглядає наступним чином рис.5.4

Рис. 5.4 «Синхронізація ключів»

перед відправлення повідомлення відкритим ключем по черзі шифруються такі параметри, як незвідний поліном, утворюючий елемент і вектор ініціалізації, дані відправляються закритим каналом зв’язку одержувач Arduino залишається за допомогою секретного закритого ключа виконати операцію розшифрування.

Після завершення обміну ключи RSA більше не використовуються і вони можуть бути знищені.

5.5 Шифратор узагальненого Галуа-64

У основі потокового шифру є шифр XOR - метод симетричного шифрування, що полягає в накладанні послідовності, що складається з випадкових чисел, на відкритий текст. Послідовність випадкових чисел називається гамма-послідовністю і використовується для зашифровування і розшифрування даних. В свою чергу в основі шифру XOR знаходиться логічна та побітова операція порозрядного додавання за модулем 2 (XOR), що набуває значення «істина» тоді й лише тоді, коли значення «істина» має суто один з її операндів, таблиця істинності оператора (табл.8). Шифрування шифром XOR відбувається за формулою:

$$X = I \oplus K, \qquad (5.7)$$

де X – зашифрований байт, I – байт шифрування, K – байт ключа. Для того, щоб розшифрувати інформацію необхідно на зашифрований байт накласти байт ключа.

Таблиця 5.1

Істиності оператора XOR

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

Шифр, побудований на основі узагальнених матриць Галуа, базується на шифрі XOR, з тією різницею, що гама для шифру формується за одним з обраних алгоритмів, який є криптографічно стійким генератором (з використанням ключа) псевдовипадкової послідовності.

У синхронному потоковому шифрі потік псевдовипадкових цифр генерується незалежно від відкритого і зашифрованого тексту повідомлення, а потім поєднується з відкритим текстом (для шифрування) або зашифрованим текстом (для дешифрування). У найбільш поширеному вигляді використовуються двійкові цифри (біти), а ключовий потік поєднується з відкритим текстом за допомогою XOR, як показано на рисунку 5.5.

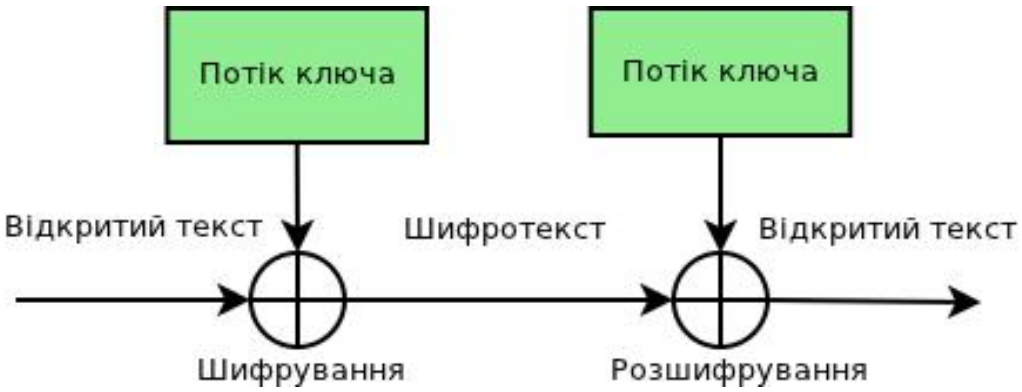

Рис. 5.5 «Шифр XOR»

Відомий спосіб криптографічного перетворення, заснований на представленні послідовності інформації у вигляді 64-бітних блоків, які піддаються ітераційній обробці примітивними криптографічними перетвореннями:

Спочатку вихідні 64-розрядні блоки представляються у вигляді байтів {b0,b1,...b7} і додаються операцією XOR за модулем 2 для отримання гами. Тоді отриманий байт ключа є гамою з одним байтом інформації, блок-схема шифру Галуа-64 наведена на рис. 5.6.

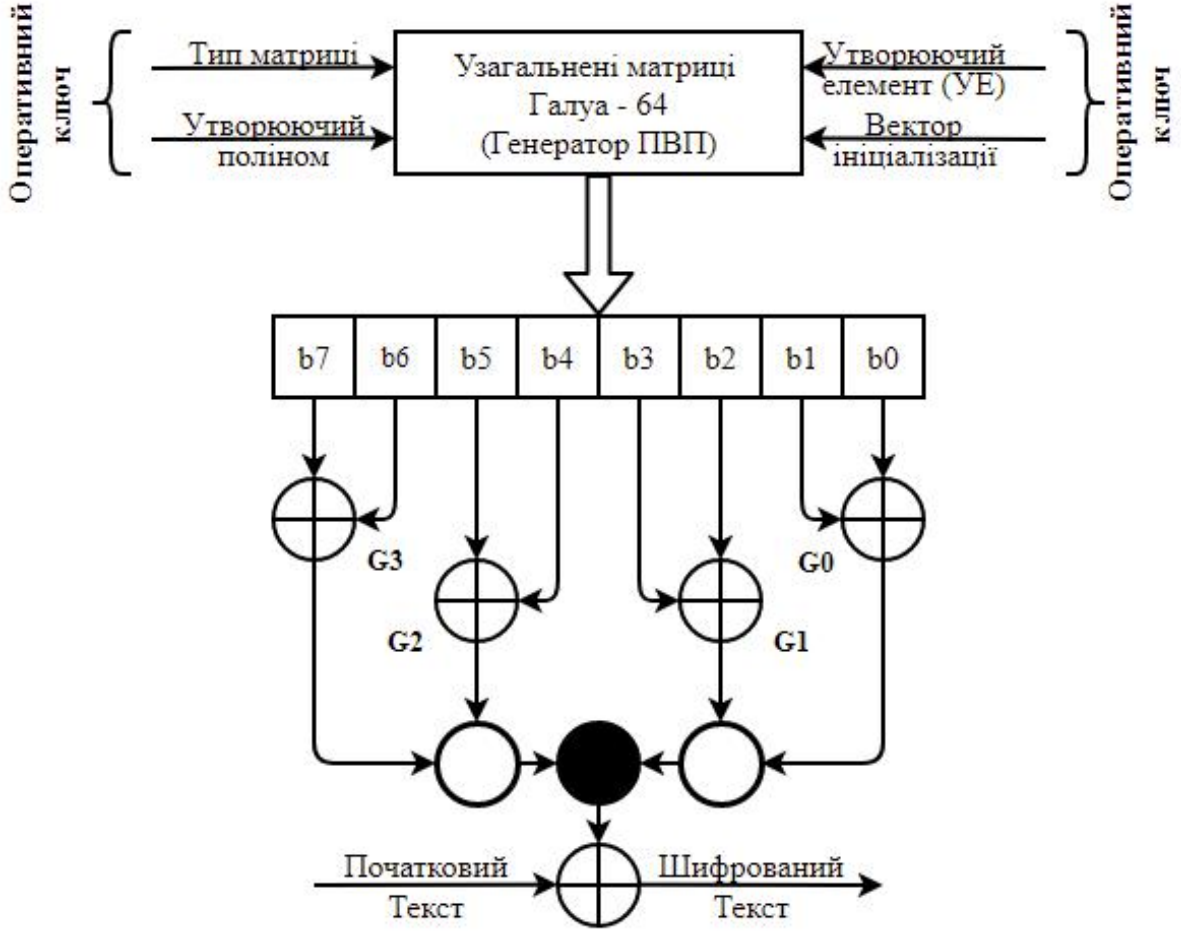

Рис. 5.6 «Схема Галуа-64»

Важливими параметрами операторного ключа Галуа шифратора є:

- тип матриці;
- незвідний поліном f 64-степені;
- утворюючий примітивний елемент повинен бути $\theta > 10$;
- вектор ініціалізації може набувати будь яких значень в діапазоні $2^1, \ldots, 2^{64}$ обирається стохастично;

Після визначення необхідних вхідних значень обрана узагальнена матриця Галуа формується незвідним поліномом *f* та формуючим елементом. На наступному етапі відбувається формування ПВП:

$$S(t+1) = S(t) \cdot G_{f,\theta}^{(n)}, t = 1,2, \ldots, \qquad (5.8)$$

де S(*t)* – значення регістру в дискретний момент часу t

$$S(t+1) = S(t) \cdot G_{f,\theta}^{(n)} = a_{n-1}\, a_{n-2} \,:\, a_1\, a_0 \;\cdot\; (n\, n-1 \,:\, 2\, 1) \qquad (5.9)$$

Ітеративні етапи розрахунку ПВП за формулою (5.9) наведено в таблиці 5.2.

Таблиця 5.2

Етапи обрахунку ПВП

t	Разряды, $\boldsymbol{S}(t)$							
	8	7	6	5	4	3	2	1
0	0	0	0	0	0	0	0	1
1						ω		
2				$\boldsymbol{S}(2) = \boldsymbol{S}(1) \cdot \boldsymbol{G}$				
3			$\boldsymbol{S}(3) = \boldsymbol{S}(2) \cdot \boldsymbol{G}$					
⋮	…							
k	0	0	0	0	0	0	0	1

На кожному *k*-му етапі шифрування обчислюється черговий елемент ПВП, після чого вхідна інформація шифрується гамма-ключем. Зворотний процес відбувається за тією ж схемою, за винятком того, що в якості вхідної інформації отримується шифротекст.

5.6 Програмна реалізація

Комп’ютерна версія програми розроблена за допомогою мови програмування C#. Це сучасна об'єктно-орієнтована мова програмування загального призначення, розроблена компанією Microsoft і схвалена Європейською асоціацією виробників комп'ютерів (ECMA) та Міжнародною організацією зі стандартизації (ISO).

Мова C# розроблена для спільної мовної інфраструктури (Common Language Infrastructure, CLI), яка складається з виконуваного коду та середовища виконання, що дозволяє використовувати різні мови високого рівня на різних комп'ютерних платформах та архітектурах. Широке розповсюдження мови C# як професійної мови зумовлено наступними причинами

- це сучасна, універсальна мова програмування
- Вона є об'єктно-орієнтованою.
- Компонентно-орієнтована.
- Легко вивчається.
- Це структурована мова.
- Створює ефективні програми.
- Його можна компілювати на різних комп'ютерних платформах.
- Входить до складу .Net Framework.

5.7 Інтерфейс програми

Інтерфейс програмного забезпечення був створений за допомогою мови програмування C# та технології .NET Framework і WPF рис 5.7.

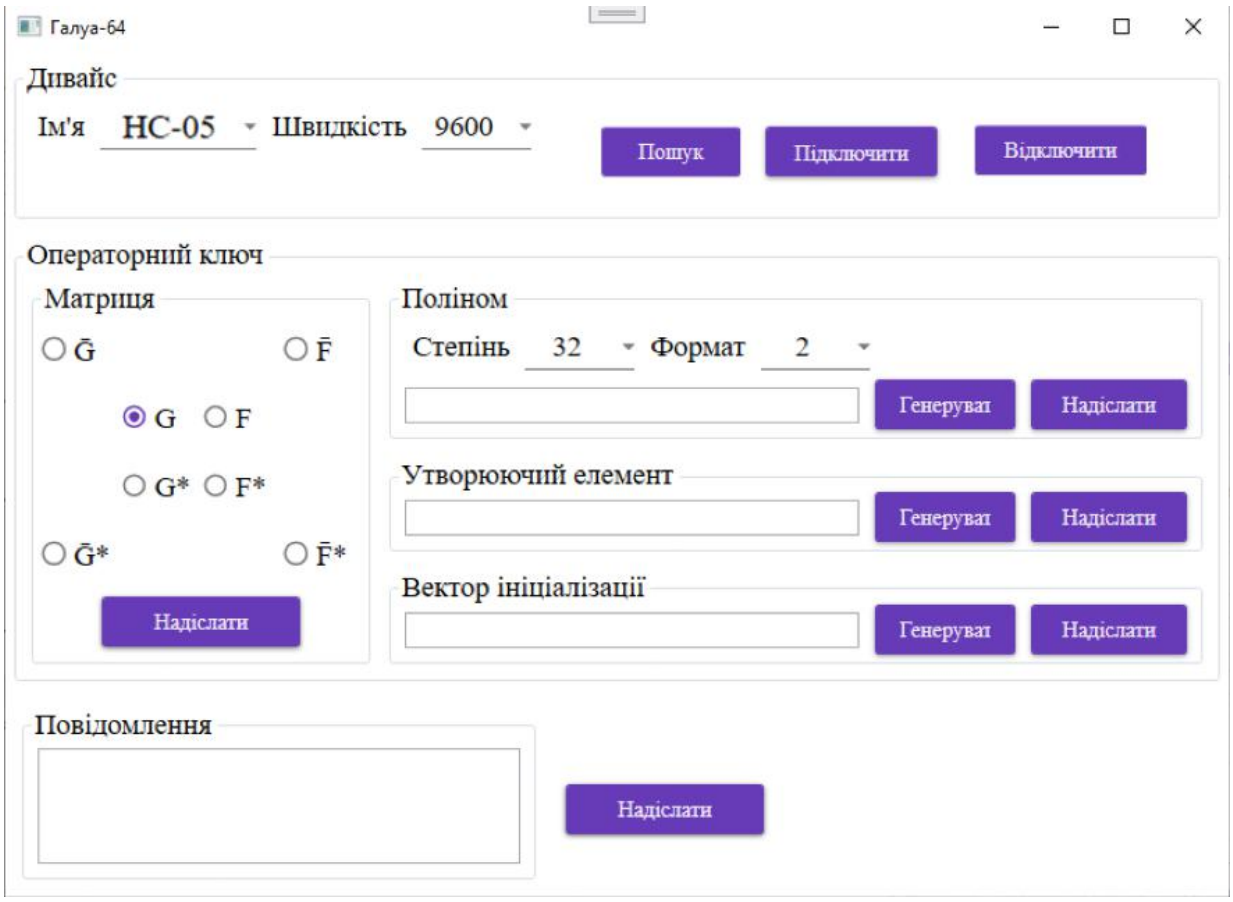

Рис. 5.7 «Інтерефейс програмного продукту»

Інтерфейс програми має всі необхідні компоненти для підключення і забезпечення управлінням шифрування.

Windows Presentation Foundation, - це середовище для створення клієнтських додатків для настільних комп'ютерів (UI - user interface). Платформа розробки WPF підтримує широкий набір функцій розробки додатків, включаючи модель додатка, ресурси, елементи управління, графіку, макет, прив'язку даних, документи і безпеку. WPF є частиною .NET Framework, використовує розширювану мову розмітки додатків XAML, щоб забезпечити декларативну модель для програмування додатків.

WPF дозволяє розробляти додатки з використанням як розмітки, так і коду, з яким повинні бути знайомі розробники. Основна поведінка програми полягає в реалізації функціональності, яка реагує на дії користувача. Наприклад, натискання на меню або кнопку, і виклик у відповідь бізнес-логіки і логіки доступу до даних. У WPF ця поведінка реалізується в коді, який пов'язаний з розміткою. Цей тип коду відомий як code-behind.

Таке розділення зовнішнього вигляду і поведінки має наступні переваги:

- Витрати на розробку і підтримку знижуються, оскільки специфічна для зовнішнього вигляду розмітка не тісно пов'язана з кодом, специфічним для поведінки.
- Розробка є більш ефективною, оскільки дизайнери можуть реалізовувати зовнішній вигляд програми одночасно з розробниками, які реалізують поведінку програми. Спрощується глобалізація і локалізація для WPF додатків.

Елементи керування найчастіше виявляють і реагують на введення користувача. Система введення WPF використовує як прямі, так і маршрутизовані події для підтримки введення тексту, управління фокусом та позиціонування миші.

Додатки часто мають складні вимоги до введення. WPF надає систему команд, яка відокремлює дії, що вводяться користувачем, від коду, який відповідає на ці дії. Система команд дозволяє декільком джерелам викликати одну і ту ж логіку команд. Наприклад, візьмемо загальні операції редагування, що використовуються різними додатками: Копіювати, Вирізати та Вставити. Ці операції можуть бути викликані за допомогою різних дій користувача, якщо вони реалізовані за допомогою команд.

5.7.1 Елементи керування

Взаємодія з користувачем, яка забезпечується моделлю додатку, є сконструйованими елементами управління. У WPF елемент управління є загальним терміном, який застосовується до категорії класів WPF, що мають наступні характеристики:

- Розміщуються або у вікні, або на сторінці.
- Мають користувацький інтерфейс.
- Реалізують деяку поведінку.

При створенні користувальницького інтерфейсу елементи управління розташовуються за розташуванням і розміром, утворюючи компонування. Ключовою вимогою будь-якого макета є адаптація до змін розміру вікна та налаштувань відображення. Замість того, щоб змушувати вас писати код для

адаптації макета в цих умовах, WPF надає вам першокласну розширювану систему макетів.

В системі компонування є відносне позиціонування, яке збільшує здатність адаптуватися до мінливих умов вікна і дисплея. Система компонування також керує узгодженням між елементами керування для визначення макета. Узгодження - це двоетапний процес: по-перше, елемент управління повідомляє своєму батькові, яке розташування та розмір він вимагає. По-друге, батько повідомляє елементу управління, який простір він може мати.

Система компонування піддається впливу дочірніх елементів управління через базові класи WPF. Для поширених макетів, таких як сітки, укладання та стикування, WPF включає кілька елементів управління макетами.

5.7.2 Мова розмітки

XAML - це мова розмітки на основі XML, яка декларативно реалізує зовнішній вигляд програми. Зазвичай її використовують для визначення вікон, діалогових вікон, сторінок і елементів керування користувача, а також для їх заповнення елементами керування, фігурами та графікою.

У застосуванні до моделі програмування .NET Core, XAML спрощує створення інтерфейсу користувача для програми .NET Core. Ви можете створювати видимі елементи інтерфейсу в декларативній розмітці XAML, а потім відокремити визначення інтерфейсу від логіки виконання за допомогою файлів code-behind, які приєднуються до розмітки через часткові визначення класів. XAML безпосередньо представляє екземпляри об'єктів в певному наборі типів підтримки, визначених в збірках. Це відрізняється від більшості інших мов розмітки, які, як правило, є інтерпретованою мовою без такої прямої прив'язки до системи типів підкладки. XAML забезпечує робочий процес, в якому окремі сторони можуть працювати над інтерфейсом і логікою програми, використовуючи потенційно різні інструменти.

Представлені у вигляді тексту, файли XAML є файлами XML, які зазвичай мають розширення .xaml. Файли можуть бути закодовані будь-яким кодуванням XML, але типовим є кодування UTF-8.

5.8 Мікроконтролерна реалізація

Мікроконтролерна реалізація Arduino виконана на мові програмування C та C++ в інтегрованому середовищі розробки Arduino. Arduino Software (IDE) - містить текстовий редактор для написання коду, область повідомлень, текстову консоль, панель інструментів з кнопками для загальних функцій і ряд меню. Воно підключається до апаратного забезпечення Arduino для завантаження програм і спілкування з ними.

Програми, написані за допомогою IDE, називаються скетчами. Такі скетчі пишуться в текстовому редакторі і зберігаються з розширенням файлу .ino. Редактор має функції вирізання/вставки та пошуку/заміни тексту. Область повідомлень надає зворотній зв'язок при збереженні та експорті, а також відображає помилки. Консоль відображає текст, що виводиться Arduino Software, включаючи повні повідомлення про помилки та іншу інформацію. У правому нижньому куті вікна відображається сконфігурована плата та послідовний порт. Кнопки панелі інструментів дозволяють перевіряти і завантажувати програми, створювати, відкривати і зберігати скетчі, відкривати монітор послідовного порту.

РОЗДІЛ 6
ОХОРОНА ПРАЦІ

6.1 Охорона праці та техніка безпеки

В магістерській дипломній роботі детально розглядаються основні аспекти правового регулювання охорони праці та навколишнього середовища, безпечні та здорові умови трудової роботи, шкідливі та небезпечні виробничі фактори, психосоціальні аспекти охорони праці, нещасні випадки на виробництві та професійні захворювання, їх розслідування та облік. Також в роботі проаналізовано наукову та навчальну літературу, а також практику з метою виявлення найбільш актуальних проблем у сфері охорони праці.

Запропоноване більш комплексне визначення охорони здоров'я та безпеки на робочому місці, а також аргументується важливість врахування психологічних факторів на роботі та посилення права працівників на охорону їх психічного здоров'я та захист від психологічних утисків.

Оскільки електричні схеми та програми розробляються, моделюються та програмуються за допомогою комп’ютера і мікрокомп'ютера, тому потрібно дотримуватися загальних положень при роботі з ЕОМ. Приміщення де виконувався проект, обладнано чотирма персональними комп'ютерами, оргтехнікою, шафами для зберігання паперів. Це окреме приміщення загальною площею 32м2, об'ємом 32 м3 і дозволяє розмістити 4 інженера (8 кв. м на 2 особи), при цьому на одного інженера та розробника припадає по 27 м3. Аналіз умов праці та розробка заходів з охорони праці для інженера з електротехніки.

6.2 Основні положення

Дія інструкції поширюється на всі підрозділи підприємства, де виконуються робота з ЕОМ. Інструкція розроблена відповідно до Положення про розробку інструкцій з охорони праці, затвердженого наказом Державного комітету України з промислової безпеки, охорони праці та гірничого нагляду від 29.01.1998 № 9, Типового положення про порядок проведення навчання і перевірки знань з питань охорони праці, затвердженого наказом Державного комітету України з промислової

безпеки, охорони праці та гірничого нагляду від 26 лютого 1998 року. 03.03.2010 № 65, Державних санітарних правил і норм роботи з електронно-обчислювальними машинами ДСанПіН 3.3.2.007-98, затверджених постановою Голови Державної санітарно-епідеміологічної служби України від 10.12.1998 № 7, Загальних вимог до роботодавців щодо забезпечення безпеки і гігієни праці працівників під час виконання робіт, затверджених наказом Міністерства надзвичайних ситуацій України від 25.01.2012 № 67 (НПАОП 0.00-7.11-12).

Згідно з цією інструкцією працівник, який користується персональним комп'ютером (далі - користувач), повинен бути ознайомлений перед початком роботи (первинний інструктаж) і через кожні шість місяців після нього (повторний інструктаж). Результати інструктажу заносяться до журналу реєстрації інструктажів з питань охорони праці на робочому місці (журнал повинен бути підписаний інструктором та користувачем).

Користувач несе відповідальність за власну безпеку та здоров'я, а також за безпеку та здоров'я інших осіб під час роботи та перебування у приміщенні.

До роботи з персональними комп'ютерами допускаються особи, які пройшли інструктаж з охорони праці та пожежної безпеки.

Користувач зобов’язаний:

- Дотримуватися правил внутрішнього трудового розпорядку;
- Користувач повинен дотримуватися внутрішніх правил компанії і не допускати на робоче місце сторонніх осіб;
- Не допускати жодних осіб на територію підприємства без попередньої письмової згоди роботодавця;
- знати правила надання першої медичної допомоги;
- знати місцезнаходження та правила користування вогнегасниками;
- вміти працювати з комп'ютером.

Приміщення з ЕОМ повинні мати природне та штучне освітлення.

6.3 Перелік небезпечних та шкідливих факторів у робочій зоні

При роботі з технічними пристроями по ГОСТ'у 12.0.003-74 під час роботи впливають основні небезпечні та шкідливі фактори:

1. Несприятливий мікроклімат:

Небезпечні та шкідливі для здоров'я умови праці на робочому місці згідно з ГОСТ 12.1.005-88 " «Загальні санітарно-гігієнічні вимоги до повітря робочої зони" у виробничих приміщеннях під час виконання робіт пов'язаних з нервово-емоційним напруженням, необхідно дотримуватися оптимальні значення температури повітря 22-24°С, при її відносній вологість 60-40% і швидкість руху (не більше 0,1 м/с).

2. Запиленість робочої зони:

Найважливішим параметром складу повітря є запиленість. Пил у робочому приміщенні - це аерозоль, що розкладається, який утворюється при механічному стиранні частинок ґрунту, внесених у приміщення, на взутті та одязі. Концентрація пилу в приміщенні не перевищує 0,1 мг/м2 при розмірі частинок не більше 3 мкм, що відповідає нормі. Рівень пилу повинен бути зменшений шляхом вологого прибирання приміщення.

3. Промислова радіація:

При роботі з комп'ютером інфрачервоне, ультрафіолетове та інші види випромінювання, опромінюється до 50% поверхні тіла користувача комп'ютера, Згідно з ГОСТ 12.1.005-88 інтенсивність теплового випромінювання не повинна перевищувати 70 Вт/м2 .

4. Підвищений рівень шуму, спричинений технічним обладнанням;

Основні характеристики та допустимі рівні шуму на робочих місцях визначені в ДСН 3.3.6.037-99 "ССБТ. Шум. Загальні вимоги безпеку". Допустимі рівні звукового тиску для аналітичних та вимірювальні роботи - 60 ДБА.

5. Недостатнє освітлення робочої зони. Вимоги до освітлення викладені в ДБН В.2.5-28-2006 "Природне і штучне освітлення". Збереження зору людини, стану центральної нервової системи значною мірою залежить від освітлення. Розрахунок освітленості робочого місця обмежується вибором системи. системи освітлення,

визначення необхідної кількості світильників, їх типу та розміщення. У приміщенні використовується система загального штучного освітлення, з використанням люмінесцентних ламп в світильниках. загальне освітлення. В якості світильників загального освітлення використовуються люмінесцентні лампи типу ЛП071-4х18-571/Мілан з індексом передачі кольору не менше 70 (R>=70), в якості світильників - світильники типу УСП-35-4х18 з наступними характеристиками можливість плавного регулювання яскравості. У нормах викладені вимоги до освітлення на робочому місці інженера. на робочому місці інженера, оскільки рекомендована освітленість поверхні робочого столу в зоні робочого документа становить від 300 до 500 лк. Місцеве освітлення не повинно засліплювати поверхню екрана або збільшувати освітленість екрана понад 300 лк. Джерелами штучного освітлення є люмінесцентні лампи. Їх переваги полягають у наступному:

економічність, тривалий термін служби, рівномірна освітленість в полі зору,

спектр випромінювання близький до спектру природного кольору.

У приміщенні, де розташоване робоче місце інженера, використовується наступне освітлення.

Змішане освітлення, тобто поєднання природного та штучного освітлення. 6Високоякісне природне бокове освітлення через вікна. Штучне освітлення використовується при недостатньому природному освітленні.

6. Підвищення статичної електрики;
7. електричний струм;
8. навантаження на зір та увагу;
9. тривалі статичні навантаження.

6.4.1 Вимоги щодо забезпечення пожежної безпеки

Загальні вимоги до систем протипожежного захисту та пожежогасіння регулюються спеціальними нормативно-технічними документами. Згідно з класифікацією приміщень за ступенем пожежної небезпеки, робоче місцеінженера знаходиться в класі В, який характеризується наявністю твердих матеріалів. горючих і важкогорючих речовин і матеріалів та легкозаймистих речовин і

матеріалів. матеріалів. У зв'язку з цим можна виділити декілька протипожежних заходів:

1. не палити і не користуватися нагрівальними приладами в приміщенні з ПК;
2. не від'єднувати і не підключати кабелі, усунення несправностей при
3. Не знімайте і не відключайте електроприлади, якщо в мережі є напруга;
4. не визначати наявність напруги в ланцюзі шляхом замикання роз'ємів.

В електронно-обчислювальній техніці пожежну небезпеку становлять такі фактори обладнання, що нагрівається, електро- та радіоелементи. Вони нагрівають навколишнє повітря та прилеглі компоненти і провідники. Все це може призвести до займання цих елементів, руйнування ізоляції та короткого замикання.

На випадок пожежі повинен бути один (не менше одного) вогнегасник.
(не менше 2 вуглекислотних вогнегасників на кожні 20 м2) вуглекислотний вогнегасник. ВВК-5. Технологічні об'ємні перекриття повинні бути виконані з негорючих або важкогорючих матеріалів з межею вогнестійкості не менше 0,5 год. Підземні приміщення з об'ємними рівнями нижче розділені негорючими перегородками з наступною межею вогнестійкості не менше 0,75 г при площі поверхні не більше 250 м2 .

6.4.2 Вимоги техніки безпеки перед початком роботи

До роботи на обладнанні допускається оператор, який оглянув об'єкт, технічну інструкцію, поточну інструкцію, а також пройшли перевірку щодо охорона праці та пожежна безпека. Перед початком робіт необхідно

1. переконайтеся, що пристрій підключено правильно;
2. переконатися, що в робочій зоні немає зайвих предметів;
3. перевірити з'єднувальні ланцюги і кабелі, місця відключення і перевірте електропроводку та з'єднання і переконайтеся, що вони знаходяться в належному стані;
4. при виявленні несправності в обладнанні або інструменті, що використовується при виконанні наступних операцій. що застосовуються під час роботи, повідомити бригадира.

6.4.3 Вимоги безпеки під час виконання робіт

Під час виконання робіт необхідно:

1. Використовуйте тільки придатний до використання термометр і тільки термометр

2. Використовуйте тільки відповідний термометр і використовуйте термометр тільки за призначенням;

3. Замінюйте запобіжник тільки при номінальному струмі і тільки за призначенням, тільки при вимкненому приладі;

4. У разі іскріння, короткого замикання, запаху гару або диму, негайно вимкнути прилад і з'ясувати можливі причини. вогонь;

5. Регулярно перевіряйте електричний ланцюг приладу, щоб встановити причини пожежі;

6. Прибрати з робочого місця зайві предмети, що заважають роботі. і може призвести до травмування інженера.

6.4.4 Вимоги безпеки після закінчення роботи

По закінченню роботи необхідно підбити підсумки:

1. Вимкніть машину, коли оператор або технічний персонал залишає своє робоче місце;

2. Приберіть своє робоче місце;

3. Дотримуватися санітарних норм та правил особистої гігієни;

4. Інформувати керівника про будь-які недоліки в роботі обладнання.

6.4.5 Вимоги безпеки під час аварійних ситуацій

1. При виникненні аварії негайно відключити все обладнання, що знаходиться під напругою, від мережі, не допускати в небезпечну зону сторонніх осіб, повідомити керівника робіт.

2. У разі виникнення пожежі негайно викликати пожежну команду.

Негайно викличте пожежну команду. Приступайте до гасіння пожежі самостійно до її прибуття та рятуйте людей і надавайте їм допомогу;

3. У разі виникнення пожежі відключіть прилад від електромережі;

4. Обов'язкова наявність засобів пожежогасіння в приміщенні. засобів пожежогасіння;

6.5 Розрахунок освітлення робочої зони

Вимоги до освітлення містяться в ДБН В.2.5-28-2006 "Природне і штучне освітлення". Збереження зору людини, стан її центральної нервової системи значною мірою залежить від освітлення [7].

Розрахунок освітлення робочого місця зводиться до вибору системи освітлення, визначення необхідної кількості світильників, їх типу і розташування. Приміщення має систему загального штучного освітлення з люмінесцентними лампами в світильниках загального освітлення. В якості загального освітлення використовуються люмінесцентні лампи типу ЛП071-4х18-571/Мілано з індексом передачі кольору не менше 70 (R>=70) та лампи типу УСП-35-4х18 з можливістю плавного регулювання яскравості.

Вимоги до освітлення робочого місця інженера визначені в нормативних документах. Рекомендована освітленість на поверхні столу в зоні розташування робочих документів становить 300 - 500 лк. Місцеве освітлення не повинно викликати відблисків на поверхні екрана і не повинно підвищувати освітленість екрана вище 300 лк.

Джерелами штучного світла є люмінесцентні лампи. Їх переваги полягають у наступному: Ефективність, тривалий термін служби, рівномірне освітлення поля зору, спектр випромінювання близький до природного колірного спектру.

У приміщенні, де розташоване робоче місце інженера, використовується змішане освітлення, тобто поєднання природного та штучного світла. В якості природного бокового освітлення через вікна. Штучне освітлення використовується при недостатньому природному освітленні. Тип світильника: світильник підвісний жалюзійний для громадських місць LP071-4x18-571/Milano з умовним номером групи 1.

Для планування штучного освітлення використовуємо метод світлового потоку, так як розраховуємо рівномірне загальне освітлення приміщення при освітленні тільки в горизонтальній площині.Визначаємо індекс приміщення:

$$i = \frac{s}{h(A+B)}$$

$$i = \frac{s}{h(A+B)} = \frac{32}{2.1(4+8)} = 1.26$$

Коефіцієнти відбиття від стелі 70%, від стін – 50%, від підлоги – 10% Вибираємо характеристики, що відповідають типу лампи – ЛЕЦ(754) напруга 110 +/- 11В, тривалість горіння – 5200 годин, світловий потік після 100 годин горіння 4000 лм.

Розраховуємо потрібну кількість світильників:

$$N_c = \frac{100 \cdot E \cdot S \cdot K_3 \cdot Z}{n_i \cdot \Phi \cdot \eta \cdot K_y}$$

$$N_c = \frac{100 \cdot 300 \cdot 32 \cdot 1.4 \cdot 1.3}{4 \cdot 4000 \cdot 40 \cdot 0.9} = 3.79$$

З розрахунків видно, що для даного приміщення з ПЕОМ потрібно N = 4 світильників типу ЛГ1071-4х18-57.

РОЗДІЛ 7
ОХОРОНА НАВРОЛИШНЬОГО СЕРЕДОВИЩА

7.1 Вступ

Завданням природоохоронного законодавства є регулювання відносин у галузі охорони, використання і відтворення природних ресурсів, забезпечення екологічної безпеки, запобігання і ліквідація негативного впливу господарської та іншої діяльності на навколишнє природне середовище, охорона природних ресурсів, генетичної спадщини диких тварин і рослин, ландшафтів та інших природних об'єктів, унікальних територій та природних об'єктів історико-культурної спадщини.

Конституція України передбачає раціональне використання земельних, лісових, атмосферних та водних ресурсів. У сфері охорони навколишнього природного середовища наразі діє низка нормативних актів: Закон України "Про охорону навколишнього природного середовища"; Постанова Уряду України "Про затвердження Порядку визначення нормативів збору за забруднення навколишнього природного середовища і стягнення цього збору та його лімітів" тощо.

7.2 Охорона довкілля

Під довкіллям розуміється цілісна система взаємопов'язаних природних і антропогенних об'єктів та явищ, під впливом і при безпосередньому використанні яких відбуваються праця, побутова діяльність і відпочинок людей. Поняття довкілля включає в себе соціальні, природні та антропогенні фізичні, хімічні та біологічні фактори, тобто все, що впливає на життя і діяльність людини. Природне середовище є невід'ємною частиною навколишнього природного середовища. Роль сучасного суспільства полягає не лише у захисті природи, а й у запобіганні негативним наслідкам господарської діяльності людини у майбутньому.

Охорона навколишнього середовища - складна і комплексна проблема, яка стосується як суспільства в цілому, так і кожного окремого громадянина.

Йдеться про вирішення життєво важливої проблеми - захист і збереження здоров'я нинішнього та майбутніх поколінь від шкідливого впливу науково-технічної та виробничої діяльності.

На початку своєї історії людина задовольняла лише прості фізіологічні потреби (їжа, одяг, житло). З розвитком суспільства зростало використання природних ресурсів для задоволення матеріальних потреб. Сьогодні людина дуже активно впливає на природу.

Одним з небажаних, але очевидних наслідків технологічних процесів є забруднення навколишнього середовища побічними продуктами виробничо-технологічної діяльності.

Стан довкілля в Україні викликає серйозне занепокоєння, оскільки є наслідком економічних помилок та екологічних прорахунків, оскільки природне середовище вже не здатне самовідновлюватися та очищатися, а природні ресурси активно погіршуються та небезпечно знищуються.

У цьому контексті викладено основні шляхи, якими Україна може подолати серйозну екологічну кризу:

Розробка комплексних природоохоронних програм на основі результатів моніторингу;

Збільшення витрат на охорону природи та прискорення будівництва природоохоронних об'єктів.

Заборона виходу з екологічно шкідливих проектів тощо.

7.3 Чинники забруднення

Оскільки основними джерелами забруднення довкілля є технологічні процеси та неналежне поводження з відходами, головним завданням є розробка безвідходних або маловідходних технологій, використання сучасних транспортних засобів, впровадження організаційних заходів з екологічної безпеки, таких як збори за викиди в атмосферу та воду, продаж квот на викиди в навколишнє середовище, раціональне розміщення підприємств з урахуванням щільності населення тощо.

На сучасному етапі, коли монітори та комп'ютери використовуються масово, неможливо не враховувати їх вплив на навколишнє середовище на всіх етапах: при виготовленні, під час використання та після закінчення терміну експлуатації. Наразі існують екологічні стандарти, які визначають вимоги до виробництва обладнання та

матеріалів, що використовуються при його будівництві. Вони не повинні містити фреонів, хлоридів або бромідів і тому подібних.

7.2 Компоненти апаратної частини

Матеріали, що використовуються для виробництва комп'ютерних мікросхем, у своїй більшості є рідкісними. Мало того, що вони надходять з різних джерел, так ще й існує багато етапів перетворення їх на компоненти і доставки їх туди, де вони потрібні для складання.

Найбільш вагомий внесок у забруднення довкілля і власне проблемою є забезпечення таких ключових елементів: гафній, хром, кобальт, платина, тантал і паладій.

Тантал і паладій зазвичай використовуються в транзисторах або конденсаторах і часто поєднуються або нашаровуються з кремнієм. Вони складають значну частину графічних карт (GPU) або оперативної пам'яті (пам'яті), а також багато інших матеріалів. Нашарування двох матеріалів таким чином може поліпшити зберігання даних на менших чіпах, що робить їх більш продуктивними з точки зору продуктивності, а також більшою ємністю, хоча і з меншим профілем.

На відміну від флеш-пам'яті, звичайний жорсткий диск з рухомими частинами потребує магнітного диска, виготовленого з різних металів. Деякі поширені матеріали, що використовуються для цього, включають оксид заліза, кремній, мідь, цинк, нікель, алюміній та інші. Однак присутні у його складі і більш рідкісні елементи хром, кобальт і платину.

Інші звичайні та дорогоцінні метали також використовуються у виробництві комп'ютерних чіпів, включаючи цинк, залізо, нікель, золото, мідь та алюміній. Різні кількості кожного матеріалу використовуються в повній системі, оскільки комп'ютери в цілому складаються з багатьох різних компонентів, починаючи від материнської плати і закінчуючи периферійними пристроями вводу/виводу.

У випадку з настільними комп'ютерами набагато легше побачити, як безліч компонентів збираються і з'єднуються для створення кінцевого продукту. Однак з меншими пристроями, такими як ноутбуки, планшети і смартфони, це не завжди так

очевидно. У цих мініатюрних пристроях все ще є багато компонентів - вони просто розроблені для того, щоб бути більш компактними і часто мають унікальні форми або форм-фактори.

Найпоширеніші компоненти, які використовуються при створенні комп'ютерних мікросхем це друковані плати, конденсатори, транзистори і резуистори.

7.2.1 Друковані плати

Хоча самі по собі друковані плати не є матеріалом для комп'ютерних мікросхем, вони є одним з найбільш затребуваних компонентів будь-якого процесу виготовлення комп'ютерних мікросхем. Вони утворюють каркас, на якому закріплюється, встановлюється і з'єднується все інше обладнання.

Сьогодні друковані плати виготовляються з використанням найрізноманітніших матеріалів, включаючи пластмаси. Вони також можуть бути виготовлені різними способами. Наприклад, адитивне виробництво (3D-друк) стає все більш поширеним, оскільки воно доступне і використовує регулярно доступні матеріали. Термопласти або скловолокно є основними матеріалами, а струмопровідна мідь часто використовується для пайки та електричних доріжок.

7.2.2 Конденсатори

Конденсатори, Призначені для регулювання електроенергії, конденсатори дозволяють пропускати змінний струм, одночасно блокуючи постійний струм від входу в ланцюг. Фактична робота трохи складніша, але всередині комп'ютерів та електроніки вони забезпечують відповідну напругу для різного обладнання, такого як відеокарти, процесори та жорсткі диски. Конденсатори також можуть використовуватися для зберігання електроенергії, хоча б протягом мінімального періоду.

Вони складаються з таких матеріалів, як тантал, паладій і рутеній. Вони також відповідають за ланцюжок поставок і проблеми дефіциту через те, звідки беруться матеріали. Як правило, відповідні матеріали є в надлишку, але в неспокійних регіонах, таких як райони, зруйновані війною або конфліктними ситуаціями.

7.2.3 Транзистори

Призначені для посилення або перемикання електричних сигналів, транзистори використовуються майже в усій електроніці - вони є основним будівельним блоком. Оскільки вони складаються з напівпровідників, клем і припою, вони створюються з різних матеріалів. До них відносяться кремній, германій, арсенід галію та індій.

7.2.4 Резистори

Щоб допомогти краще контролювати потік електричних струмів і створювати опір там, де це можливо, резистори майже завжди встановлюються в ланцюгах. У найпростішому вигляді вони являють собою електроди, виготовлені зі сплавів металів, таких як платина, срібло і паладій.

7.2.5 Припій

Припій - це провідний матеріал, який використовується для зв'язування різних компонентів або з'єднання їх в ланцюзі. Найпоширенішим матеріалом колись був свинець, але через його небезпечні властивості сьогодні широко використовуються інші метали, такі як вісмут, цинк, індій, сурма, мідь, срібло і золото.

7.3 Рідкісні компоненти

Матеріали, що використовуються для виробництва комп'ютерних мікросхем, у своїй більшості є рідкісними. Мало того, що вони надходять з різних джерел, так ще й існує багато етапів перетворення їх на компоненти і доставки їх туди, де вони потрібні для складання. Сонячні панелі, дрони, 3D-принтери і смартфони містять до 30 різних елементів, які постачаються з усього світу.

Найбільш вагомий внесок у забруднення довкілля і власне проблемою є забезпечення таких ключових елементів: гафній, хром, кобальт, платина, тантал і паладій.

Проблема полягає в тому, що сучасні технології не працюють без так званої критичної сировини. Яскравим прикладом є літій з Чилі, який необхідний для виробництва акумуляторів для електромобілів.

Ніхто, навіть, не буде заперечувати, що видобуток літію завдає величезної шкоди навколишньому середовищу, маючи на увазі штучні озера, які компанії створюють при вимиванні металу з підземних соляних резервуарів. Процес використовує величезну кількість води, тому в кінцевому підсумку ви отримуєте ці величезні затоплені території, де літій осідає.

Такий спосіб видобутку призводить до руйнування та забруднення природної водної системи. Унікальні рослини і тварини втрачають доступ до підземних вод і водопоїв. Також надходять повідомлення про засолення прісних вод через великі обсяги кислих стічних вод під час видобутку літію.

Але літій - не єдина сировина, яка завдає шкоди. За словами Гюнтера Хільперта, керівника відділу досліджень Азії німецького аналітичного центру SWP, видобуток лише однієї тонни рідкоземельних елементів призводить до утворення 2000 тонн токсичних відходів, що спустошило великі регіони Китаю.

За його словами, тамтешні компанії застосували процес розпилення кислоти над районами видобутку, щоб відокремити рідкоземельні елементи від інших руд, і що видобуті райони часто залишаються занедбаними після видобутку корисних копалин.

"Вони більше не є життєздатними для сільськогосподарського використання", - сказав Хілперт. "Природа була надмірно експлуатована".

Китай - не єдина країна з низькими екологічними стандартами і поганим управлінням ресурсами. На Мадагаскарі, наприклад, процвітаючий нелегальний сектор видобутку дорогоцінних каменів і металів пов'язаний з виснаженням тропічних лісів і знищенням природних місць проживання лемурів.

Такі держави, як Мадагаскар, Руанда і ДРК, мають низькі показники за Індексом екологічної ефективності, який оцінює 180 країн за такими факторами, як охорона природи, якість повітря, управління відходами та викидами. Тому екологи особливо занепокоєні тим, що ці країни видобувають високотоксичні матеріали, такі як берилій, тантал і кобальт.

7.4 Утилізація електронних пристроїв

Окрім ефективного захисту конфіденційної інформації на електронних пристроях, важливо дотримуватися найкращих практик утилізації електронних пристроїв. Комп'ютери, смартфони та камери дозволяють тримати велику кількість інформації під рукою, але при утилізації, пожертвуванні або переробці пристрою ви можете ненавмисно розкрити конфіденційну інформацію, яка може бути використана кіберзлочинцями.

Типи електронних пристроїв включають:

Комп'ютери, смартфони та планшети - електронні пристрої, які можуть автоматично зберігати та обробляти дані; більшість з них містять центральний процесор та пам'ять.

Цифрові медіа - ці електронні пристрої створюють, зберігають і відтворюють цифровий контент. Цифрові медіа-пристрої включають такі елементи, як цифрові камери та медіаплеєри;

Зовнішні апаратні та периферійні пристрої - апаратні пристрої, які забезпечують введення та виведення даних для комп'ютерів, такі як принтери, монітори та зовнішні жорсткі диски; ці пристрої містять постійно збережені цифрові символи.

Ігрові консолі - електронні, цифрові або комп'ютерні пристрої, які виводять відеосигнал або візуальне зображення для відображення відеоігор.

Деякі з цих невеликих електронних компонентів, таких як мікросхеми і жорсткі диски, зберігають конфіденційну та ідентифікаційну інформацію. Це ще одна причина, чому важливо переробляти і ці компоненти. Якщо ви просто викинете ці компоненти, інформація, яка зберігається на цих пристроях, може бути отримана будь-ким, хто випадково натрапить на ваші старі пристрої. Переробка цих пристроїв сертифікованою компанією з переробки гарантує, що інформація буде стерта або пристрій буде повністю знищений, що зробить інформацію на 100% безповоротною.

Згідно з доповіддю Коаліції з повернення електроніки (Electronics TakeBack Coalition), у 2008 році в США було утворено близько 3,16 млн. тонн електронних

відходів, з яких лише 14 відсотків було перероблено. Електронні відходи можуть завдати великої шкоди: свинець, ртуть і антипірени, що використовуються для виробництва електронних компонентів, вимиваються, коли продукція потрапляє на звалища, забруднюючи навколишнє середовище. Проте, багато міст і муніципалітетів активізують переробку електроніки, щоб відновити багато компонентів з повсякденних приладів і гаджетів, які можна переробити.

Електронні відходи (іноді їх називають електронними відходами) - це термін, який використовується для опису електроніки, що наближається до кінця свого терміну експлуатації та підлягає викиданню, пожертвуванню або переробці. Хоча пожертвування та переробка електронних пристроїв зберігає природні ресурси, існують додаткові можливості утилізувати електронні відходи через контактні центри місцевих НДО.

Переробка електроніки має низку переваг, таких як зменшення потреби у видобутку природних ресурсів, економія енергії та безпека даних. Однак, як щодо переробки електронних компонентів, таких як кабелі, мікросхеми, схеми та інші компоненти, що містяться в електронних пристроях.

Всі види електронних пристроїв, включаючи дрібні електронні компоненти, вважаються і розглядаються як електронні відходи, коли вони більше не використовуються або просто досягли кінця свого життєвого циклу. Саме в цих невеликих електронних компонентах часто можна знайти природні ресурси, такі як золото, залізо, мідь, платина і паладій. Коли ви переробляєте ці пристрої, дорогоцінні метали, такі як ці, витягуються з електронних відходів, таким чином заощаджуючи природні ресурси, зменшуючи забруднення та заощаджуючи енергію. Це пов'язано з тим, що видобуток цих металів у надрах землі вимагає великої кількості енергії для видобутку, переробки та транспортування, що призводить не тільки до викидів парникових газів, а й до забруднення навколишнього середовища.

Висновок

У розділі розглянуто проблеми забруднення навколишнього середовища.

ВИСНОВКИ

Одним з основних результатів даної роботи є розробка повних сукупностей класичних і узагальнених варіантів Галуа-генераторів ПВП в апаратній (на основі одно- і багатоконтурних РЗЛЗЗ) або програмному (матричному) виконанні. Відмітна особливість матричних Галуа-генераторів ПВП полягає в тому, що за допомогою використовуваних в цих генераторах матриць, множина яких містить по вісім як класичних, так і узагальнених матриць Галуа, можуть бути програмно отримані такі ж самі бінарні послідовності, як і послідовності, що формуються апаратними РЗЛЗЗ-генератором ПВП.

Запропоновано рекурентні оцінки станів класичних матричних Галуа-генераторів ПВП, що забезпечили суттєве підвищення швидкості обчислювальних процесів. І що особливо важливо, отримані рекурентні оцінки станів позначених генераторів є лінійно складними, тоді як обчислювальна складність відомих Галуа-генераторів ПВП не нижче квадратичної.

В роботі коротко позначені два варіанти захисту від атак Берлекемпа-Мессі, до яких схильні класичні Галуа-генератори ПВП. Першим з них пропонується заміна класичних генераторів ПВП на узагальнені генератори ПВП, а другий зводиться до перетворення подібності матриць Галуа. Звісно, що другий варіант захисту більш застосовний до матричних генераторів ПВП. Потужність різноманіття матриць Галуа зростає в факторіальній залежності ($n!$) від порядку матриць, що і забезпечує значне підвищення криптостійкості генераторів ПВП.

В даній дипломній роботі описано системне проектування та програмування системи апаратно програмного модуля шифрування інформації на базі мікрокомп'ютера Arduino Uno з використанням модуля Bluetooth HC-05. Описано метод, який забезпечую безпечне відправлення повідомлень через відкриті катали зв’язку.

Запропонований пристрій, призначений для використання в приміщенні або на відкритому повітрі, спроектований відповідно до наступних вимог.

Вимоги СНИП 2.09.04-87 " Адміністративні й побутові будинки".

Також були розглянуті всі небезпеки при роботі та описані правила і вимоги безпеки при роботі з пристроєм.

СПИСОК ЛІТЕРАТУРИ

1. Білецький А. Я., Ковальчук А. В., Новіков К. А., Полторацький Д. А. Алгоритм синтеза неприводимых полиномов линейной сложности: Захист інформації/ А. Я. Белецкий, А. В. Ковальчук, К. А. Новиков, Д. А. Полторацкий // Буд-во НАУ. Київ. – 2020. - ТОМ 22. - №2. – С. 74-87. – Бібліогр.: 2 назв.

2. Білецький А. Я. Generalized Pseudorandom Generators of the Galois and Fibonacci Sequences: Telecommunications and Radio Engineering / Anatoly Beletsky. – 2020.

3. Білецький А. Я., Ковальчук А. В., Новіков К. А., Полторацький Д. А. Семейство обобщённых матриц Галуа и генераторов псевдослучайных последовательностей/ А. Я. Белецкий, А. В. Ковальчук, К. А. Новиков, Д. А. Полторацкий // Буд-во НАУ. Київ. – 2020.

4. International Association for Cryptologic Research
Factorization of a 768-bit RSA modulus

5. Rivest, R.; Shamir, A.; Adleman, L
A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, *February 1978*

6. Serial Peripheral Interface (SPI) Master
Cypress Semiconductor Corporation, 2011.

7. ГОСТ 12.1.003–76 “Шум. Загальні вимоги безпеки”

ДОДАТКИ