**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**КАФЕДРА** **КОМП’ЮТЕРИЗОВАНИХ СИСТЕМ ЗАХИСТУ ІНФОРМАЦІЇ**

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Ільєнко

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 р.

На правах рукопису

УДК 004.056.5:510.22(043.3)

**МАГІСТЕРСЬКА АТЕСТАЦІЙНА РОБОТА**

**ВИПУСКНИКА ОСВІТНЬОГО СТУПЕНЯ**

**«МАГІСТР»**

**Тема**:Комбінуючий генератор симетричної криптографічної система на програмуючій логічній матриці

|  |  |
| --- | --- |
| **Автор:** | Б.Л. Йовенко |
| **Науковий керівник:** к.т.н., доц. | О.М. Кулініч |
| **Нормоконтролер:** асист. | С.В. Єгоров |

**Київ 2020**

**НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ**

**Факультет:** Кібербезпеки, комп’ютерної та програмної інженерії

**Кафедра:** Компютеризованих систем захисту інформації

**Освітній ступінь:** Магістр

**Спеціальність:** 125 «Кібербезпека»

**Освітньо-професійна програма**: «Безпека інформаційних і комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_\_\_\_\_\_\_\_\_ С.В. Казмірчук

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**

**на виконання магістерської атестаційної роботи**

**магістранта Йовенка Богдана Леонiдовича**

1. Тема: *Комбінуючий генератор симетричної криптографічної системи на програмуючій логічній матриці*

затверджена наказом ректора від 02.10 2019р. №2265/ст*.*

1. Термін виконання з 14.10.2019р. по 09.02.2020р*.*
2. Вихідні дані: Проаналізувати загальні відомості про генератори псевдовипадкових послідовностей; реалізація генератора псевдовипадкових послідовностей, а саме його проектування; на основі аналізу застосувати, зпроектувати інженерну схему на основі ПЛМ.
3. Зміст пояснювальної записки: загальні відомості про генератори псевдовипадкових посідовностей, принципи роботи систем автоматизованого проектування на програмованих логічних матриць а саме САПР і їхня роль; проектування спеціалізованих інтегральних схем на основі ПЛМ, а саме система MAX+plus II; проектування і реалізація генератора псевдовипадкових полсідовностей.

**КАЛЕНДАРНИЙ ПЛАН**

**виконання магістерської роботи**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Етапи виконання магістерської роботи** | **Термін виконання етапів** | **Примітка** |
|  | Уточнення постановки задачі | 16.10.2019 | *Виконано* |
|  | Аналіз літературних джерел | 16.10.2019- 22.10.2019 | *Виконано* |
|  | Обґрунтування вибору рішення | 22.10.2019- 24.10.2019 | *Виконано* |
|  | Збір інформації | 25.10.2019- 15.11.2019 | *Виконано* |
|  | Дослідження сучасних систем і методик аналізу та оцінки криптографічного захисту інформації | 16.11.2019- 27.11.2019 | *Виконано* |
|  | Розробка методики та структури системи генератора псевдовипадкових послідовностей | 28.11.2019- 03.12.2019 | *Виконано* |
|  | Розробка алгоритму та програмного забезпечення системи автоматизованого проектування MAX+Plus II | 04.12.2019- 08.12.2019 | *Виконано* |
|  | Апробація розробки генератора псевдовипадкових послідовностей за допомогою системи автоматизованого проектування MAX+Plus II | 09.12.2019- 12.12.2019 | *Виконано* |
|  | Перевірка на антиплагіат | 13.12.2019- 14.01.2020 | *Виконано* |
|  | Оформлення і друк пояснювальної записки | 14.01.2020- 15.01.2020 | *Виконано* |
|  | Оформлення презентації | 16.01.2020- 17.01.2020 | *Виконано* |
|  | Отримання рецензій від рецензента | 18.01.2020 | *Виконано* |
|  | Захист в ЕК | 19.01.2020 | *Виконано* |

Магістрант Б. Йовенко

(підпис, дата)

Науковий керівник (підпис, дата) А. Ільєнко)

**ПРИКЛАД НАПИСАННЯ РЕФЕРАТУ АТЕСТАЦІЙНОЇ РОБОТИ**

**РЕФЕРАТ**

Магістерська атестаційна робота складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел, додатків і має 96 сторінки основного тексту, 33 рисунка, 3 таблиць. Список використаних джерел містить 33 найменування і займає 4 сторінок. Загальний обсяг роботи 105 сторінок.

Метою роботи є розробка генератора псевдовипадкових послідовностей за допомогою системи автоматизованого проектування MAX+plusII.

В роботі вирішено задачу побудови і розробки системи автоматизованого проектування MAX+PlusII.

В роботі розроблено реалізація генератора псевдовипадкових послідовносте та його проектування. Побудована схема структури схеми пристрою в САПР

Результатом цієї роботи є може застосовувати в подальшому для навчання молодших спеціалістів у галузі кібербезпеки, а також для безпосереднього створення ключових послідовностей для апаратури шифрування.

Ключові слова: генератор псевдовипадкових послідовностей, криптографічна інформація, інформаційна безпека, інформаційно-комунікаційна система, САПР, програмована логічна матриця.

**ЗМІСТ**

[**ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 7**](#_Toc530933186)

[**ВСТУП 8**](#_Toc530933187)

[**РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ 9**](#_Toc530933188)

[1.1 Сутність та роль інформаційної безпеки у сучасному суспільстві 9](#_Toc530933189)

[1.2 Категорійно-понятійна система інформаційної безпеки 18](#_Toc530933190)

[1.3 Криптографічний захист інформації як основа інформаційної безпеки 21](#_Toc530933191)

[1.3.1 Історія криптографічного захисту інформації 22](#_Toc530933192)

[1.4 Сучасна криптографія 27](#_Toc530933193)

[1.4.1 Симетричне шифрування 27](#_Toc530933194)

[1.4.2. Асиметричне шифрування 28](#_Toc530933195)

[1.5 Атака на шифр і стійкість шифру 31](#_Toc530933196)

[Висновки до розділу 32](#_Toc530933197)

[**РОЗДІЛ 2. ЗАГАЛЬНІ ВІДОМОСТІ ПРО ГЕНЕРАТОРИ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ 34**](#_Toc530933198)

[2.1 Розвиток випадкових послідовностей 34](#_Toc530933199)

[2.2 Генератори псевдовипадкових послідовностей 35](#_Toc530933200)

[2.3 Методи отримання псевдовипадкових послідовностей 39](#_Toc530933201)

[2.4 Принципи побудови генераторів псевдовипадкових послідовностей 42](#_Toc530933202)

[2.5. Генератори, що використовують фізичні квантові випадкові процеси 44](#_Toc530933203)

[2.5.1 Фазовий квантовий шум в лазерному промені. (рис. 2.1) 44](#_Toc530933204)

[2.5.2 Матриця фотокамери (рис. 2.2) 45](#_Toc530933205)

[2.6 Гнератори, що використовують інші кванові випадкові процеси 46](#_Toc530933206)

[2.6.1 Тепловий шум(рис. 2.4.) 46](#_Toc530933207)

[2.6.2 Цифрова схема з невизначеним станом 48](#_Toc530933208)

[2.6.3 Лавинний шум (шум лавинного множення) 48](#_Toc530933209)

[2.6.4 Фазове тремтіння в кільцевих генераторах 49](#_Toc530933210)

[2.3 Методики тестування генераторів псевдовипадкових послідовностей 49](#_Toc530933211)

[2.3.1 Методика тестуванння NIST STS 53](#_Toc530933212)

[Висновки до розділу 58](#_Toc530933213)

[**3. Принципи роботи систем автоматизованого проектування та програмованих логічних матриць 59**](#_Toc530933214)

[3.1. Основні поняття про системи автоматизованого проектування 59](#_Toc530933215)

[3.1.1. Місце та роль автоматизованого проектування серед інформаційних технологій 59](#_Toc530933216)

[3.1.3. Основні відомості про системи автоматизованого проектування 62](#_Toc530933217)

[3.1.4. Переваги застосування інженерних САПР та їх роль у галузі матеріального виробництва 63](#_Toc530933218)

[3.1.5. Стан ринку інженерних САПР 64](#_Toc530933219)

[3.1.7. Класифікація САПР і їх користувачів 65](#_Toc530933220)

[3.2 Проектування спеціалізованих інтегральних схем на основі ПЛМ 67](#_Toc530933221)

[3.2.1 Загальні відомості про систему MAX+plus II 67](#_Toc530933222)

[3.2.2 Редактори системи MAX+plus II 70](#_Toc530933223)

[Висновки до розділу 78](#_Toc530933224)

[**РОЗДІЛ 4. РЕАЛІЗАЦІЯ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ 79**](#_Toc530933225)

[4.1 Вибір технічного рішення та обгрунтування структурної та функціональної схем 79](#_Toc530933226)

[4.2 Загальна методика проектування цифрових пристроїв 82](#_Toc530933227)

[4.3.Проектування генератора псевдовипадових послідовностей 93](#_Toc530933228)

[4.3.1.Загальні відомості 93](#_Toc530933229)

[4.3.2.Структурна схема пристрою 94](#_Toc530933230)

[4.3.3.Принципова схема пристрою та епюри функціонування 96](#_Toc530933231)

[Висновки до розділу 103](#_Toc530933232)

[**ВИСНОВКИ 104**](#_Toc530933233)

[**Список використаних джерел 105**](#_Toc530933234)

# ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ЕОМ — електронно-обчислювальна машина;

PRNG — pseudorandom number generator;

ГВП — генератор випадкових послідовностей;

ГПВЧ — генератор псевдовипадкових чисел;

ГПВП — генератор псевдовипадкових послідовностей;

ДЕ — джерело ентропії;

САD — система конструювання (Computer Aided Design);

САЕ — аналітично-розрахункова підсистема (Computer Aided Engineering);

САПР — система автоматизованого проектування;

ПЛМ — програмована логічна матриця;

ВІС — велика інтегральна схема;

НВІС — надвелика інтегральна схема;

AHDL — Altera Hardware Description Language;

ОБ — операційний блок;

БК — блок керування;

# ВСТУП

Невпевненість у генераторі випадкових чисел може призвести до серйозної шкоди криптографічним протоколам, і їх вразливості можуть бути використані зловмисниками. Проблема створення якісного генератора існувала ще на початку ХХ століття .

Під «випадковим числом» мається на увазі число, яке є випадковим на практиці (тобто непередбачуваним, і таким, яке повторити неможливо). Може здатися, на перший погляд, досить просто, але набагато складніше це відтворити. Є велика різниця між тим, коли ми говоримо про окреме випадкове число та про випадкову послідовність. Окреме випадкове число є одним з набору можливих значень з рівною ймовірністю, коли у випадку з послідовністю, кожне нове значення повинно бути статистично незалежним одне від одного. Ми говоримо, що А статистично незалежне від В, не зважаючи відбулося В, чи ні; і не має значення як часто відбувається А. Статистично незалежне означає, що жодне з них не несе інформації про інше.

З появою все новіших засобів шифрування та, відповідно, засобів дешифрування почали з’являтися більш різноманітні способи створення таких послідовностей, що спитаються на процеси, які відбуваються у природі та власне псевдовипадкові процеси.

Об’єкт дослідження — генератор псевдовипадкових послідовностей.

Предмет дослідження — система автоматизованого проектування MAX+PlusII.

Мета роботи — розробка генератора псевдовипадкових послідовностей за допомогою системи автоматизованого проектування MAX+plusII.

# РОЗДІЛ 1. ОСНОВНІ ПОНЯТТЯ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

У першому розділі будуть розглядатися основні поняття на які опирається і на чому базується захист інформації. Описана вага інформаційної безпеки, визначено понятійний апарат, яким повинен керуватися кожен спеціаліст у сфері кібербезпеки. Ознайомлення з криптографією розпочато з історичної довідки, також розмежовано поняття симетричної та асиметричної криптографії, порівняно їх переваги, недоліки та довжини ключів.

## 1.1 Сутність та роль інформаційної безпеки у сучасному суспільстві

Характерною ознакою сучасного етапу економічного та науково-технічного прогресу е стрімкий розвиток інформаційних технологій, їх якнайширше використання як у повсякденному житті, так і управлінні державою. Інформація і інформаційні технології все більше визначають розвиток суспільства та слугують новими джерелами національної могутності. Становлення інформаційного суспільства радикально змінює політичну, екологічну та соціальну сфери життєдіяльності людства. У цих умовах формування інформаційного суспільства змінює предмет праці на інформацію і знання. У свою чергу основою глобалізації стають інтеграція інформаційних систем різних держав до єдиної загальносвітової інформаційної системи, формування єдиного інформаційного простору, створення глобальних інформаційно-телекомунікаційних тенет, інтенсивне впровадження нових інформаційних технологій в усі галузі суспільного життя, включаючи і державне управління.

Глобальний процес інформатизації суспільства охопив практично всі країни світу і нині є стрижнем науково-технічного і соціально-економічного розвитку.

Інформатизація становить собою організаційний соціально-економічний і науково-технічний процес створення оптимальних умов для всебічного задоволення інформаційних потреб і реалізації прав громадян суспільства, органів державної влади й управління на основі формування і використання інформаційних ресурсів і використання інформаційних систем, мереж, ресурсів і інформаційних технологій із використанням обчислювальної і комунікаційної техніки.

Основними завданнями інформатизації є:

1. всебічне інформаційне забезпечення потреб суб'єктів інформаційних відносин;
2. створення єдиного безпечного інформаційного простору;
3. створення, впровадження і використання інформаційних систем, інформаційних технологій і інформаційних продуктів загального значення;
4. підготовка кадрів, підвищення їх кваліфікації у сфері інформатизації.

Осягнення сутності змісту поняття "інформаційна безпека" є важливим завданнями наукового аналізу. Будь-яке вчення лише тоді досягає зрілості і досконалості, коли розкриває сутність досліджуваних явищ, має можливість передбачати майбутні зміни не лише у сфері явищ, а й у сфері сутностей. Пізнання сутності інформаційної безпеки можливо лише на основі абстрактного мислення, створення теорії досліджуваного предмета, усвідомлення внутрішнього змісту, виявлення характерних ознак, розкриття сутнісних характеристик поняття, що вивчається.

В історичному процесі складається структура предмета, тобто єдність внутрішнього змісту і зовнішніх проявів, співпадаючих і неспівпадаючих суперечливих сутностей. Сутність — сукупність глибинних зв'язків, відносин і внутрішніх законів, які визначають основні риси і тенденції розвитку системи. Сутність може вважатися пізнаною, коли відомі причини виникнення і джерела розвитку розглядуваного об'єкта, шляхи його формування або технічного репродукування, якщо в теорії або на практиці створена його достовірна модель. Одна й та сама сутність може мати множину різних явищ.

Сутність проявляється і осягається в дефініції\* яке виражає родове поняття. Таким щодо інформаційної безпеки е поняття безпеки, яке характеризує певний процес управління загрозами та небезпеками. Відповідно видове поняття "інформаційна безпека" означає процес управління загрозами та небезпеками в інформаційній сфері.

Саме тому інформаційна безпека е невід'ємною частиною загальної безпеки, чи то національної, чи то регіональної, чи то міжнародної. Аналіз інформаційної безпеки передбачає розгляд сукупності таких об'єктивних чинників:

1. потреб громадян, суспільства і держави і світового співтовариства;
2. уразливість індивідів, суспільства і держави від цифрових технологій;
3. наявність широкого кола загроз і небезпек, якими має управляти система забезпечення інформаційної безпеки.

Інформаційна безпека є складовим компонентом загальної проблеми інформаційного забезпечення людини, держави і суспільства. Вона орієнтована на захист значимих або вже згаданих суб'єктів інформаційних ресурсів, законних інтересів. Зміст поняття "інформаційна безпека" розкривається у практичній діяльності, наукових дослідженнях, а також нормативно-правових документах.

Так, визначення інформаційної безпеки було дано у Федеральному Законі Росії "Про участь у міжнародному інформаційному обміні". У даному законі інформаційна безпека розглядалася як стан захищеності інформаційного середовища суспільства, який забезпечує його формування, використання і розвиток в інтересах громадян, організацій і держави. Ця трактовка виходить з того, що захист інформації та інформаційної інфраструктури становить содою зміст інформаційної безпеки. При цьому наголос робиться на технічний бік проблеми.

Дещо інше визначення інформаційної безпеки міститься у Доктрині інформаційної безпеки Російської Федерації, де вона визначається як стан захищеності її національних інтересів у інформаційній сфері, які визначаються сукупністю збалансованих інтересів особи^ суспільства і держави. З цього визначення випливає, що зміст поняття безпеки базується на інтересах суб'єктів суспільних відносин в інформаційній сфері, від збалансованості яких залежить рівень загроз.

Слід зазначити, що у науковій літературі поки бракує єдиного консолідованого погляду на зміст поняття "інформаційна безпека". Для одних воно відображає стан, для інших процес, діяльність, здатність, систему гарантій, властивість, функцію. Відтак постає необхідність в угрупуванні напрямів визначення аналізованого поняття.

Так, наприклад, представник першого напряму за нашою умовною класифікацією, Ю.А. Фісун, який працює в Академії управління МВС РФ характеризує інформаційну безпеку як "стан захищеності інформаційного середовища, який відповідає інтересам держави, за якого забезпечується формування, використання і можливості розвитку незалежно від впливу внутрішніх та зовнішніх інформаційних загроз". Такої ж позиції притримуються і розробники концепції інформаційної безпеки центру Разумкова, а також деякі українські дослідники, які вважають за необхідне визначати інформаційну безпеку як стан захищеності. Так наприклад, Гасеський В.К., Авраменко ВА. визначають інформаційну безпеку як стан захищеності життєво важливих інтересів особи, суспільства та держави, який виключає можливість заподіяння їм шкоди через неповноту, невчасність і недостовірність інформації, через негативні наслідки функціонування інформаційних технологій або внаслідок поширення законодавчо забороненої чи обмеженої для поширення інформації, у той час як О.Г. Додонов визначає інформаційну безпеку як стан захищеності інформаційного простору, який забезпечує формування та розвиток цього простору в інтересах особистості, суспільства та держави.

Аналогічного погляду дотримується і інший російський дослідник /. Панарін, роблячи більший акцент на ролі політичної еліти, яка може протистояти інформаційному впливу. На його думку, інформаційна безпека — стан інформаційного середовища суспільства і політичної еліти, який забезпечує її формування і розвиток в інтересах керівництва країни, громадян і суспільства.

Дещо в іншому ракурсі трактує інформаційну безпеку АА. Тер-Акопов, який репрезентує позицію другого напряму. Під інформаційною безпекою він розуміє стан захищеності інформації, яка забезпечує життєво важливі інтереси людини. У рамках даного напряму існує визначення інформаційної безпеки як стану, тенденції розвитку, умови життєдіяльності соціуму, його структур, інститутів і установ, при яких забезпечується збереження їх якісної з об'єктивними обумовленими інноваціями в ній, і вільне, відповідне власній природі і її функціонування. Ряд представників цього напряму розглядають інформаційну безпеку як стан, який характеризується відсутністю небезпеки, тобто чинників і умов, які загрожують безпосередньо індивіду, спільноті, державі з боку інформаційно-комунікаційного середовища. Прибічники такого підходу розглядають інформаційну безпеку як стан і процес захищеності особи, суспільства, держави від реальних або потенційних загроз. Водночас, на нашу думку, розглядати безпеку лише як стан є не зовсім точним, і не відображає динамізму як самої безпеки, так і тої системи, для якої безпека виступає як функція її подальшого розвитку та існування.

Поняття процес відрізняється від поняття стан. Поняття процес означає послідовність станів, пов'язаність стадій їх зміни і розвитку, тобто на відміну від поняття "стан", поняття "процес" акцентує увагу на моменті спрямованості в зміні об'єкта, цілепокладанні, тоді як "стан" відображає лише один момент, певну мить безпеки, а отже не вичерпує її повністю.

Представник третього напрями В.І. Ярочкін визначає безпеку як "стан захищеності особи, суспільства і держави від зовнішніх та внутрішніх небезпек і загроз, який базується на діяльності людей, суспільства, держави, світового співтовариства з виявлення (вивчення), попередження, послаблення, ліквідації і відбиття небезпек і загроз, здатних загубити їх, лишити фундаментальних матеріальних і духовних цінностей, завдати неприйнятної шкоди, закрити шлях для прогресивного розвитку". Застосування діяльнісного підходу, ва наш погляд, є більш адекватним при описуванні інформаційної безпеки, і ми в певній мірі можемо підтримати дане визначення у загальному плані, однак не погодитись із деталізацією напрямів діяльності, які з часом змінюватимуться, а отже закладатимуть потенціал нестійкості як до самого визначення, так і до функціонування відповідних суб'єктів.

М.П. Хрипков вважає, що діяльність по забезпеченню особи, суспільства і держави виникає в ході вирішення суперечності між такою об'єктивною реальністю, як небезпека, і потребою розумної сутності, соціального індивіда, соціальної групи попередити ЇЇ можливі шкідливі наслідки. Водночас за даного випадку функціонування системи забезпечення інформаційної безпеки зводиться лише до реагування, тоді як превенція лишається поза увагою.

Саме тому, на наше переконання, інформаційна безпека становить собою діяльність органів державного управління. Звідси витікає важливий висновок, що слід діяти активно, здійснюючи вплив на джерела інформаційної небезпеки. При цьому щодо змісту інформаційної безпеки доцільно використовувати не поняття "інтереси", а більш фундаментальне поняття "цінності", через те, що у цінностях знаходять вираз інтереси суб'єктів суспільних відносин, зіткнення яких породжує загрози.

Наступний погляд передбачає, що у самому загальному вигляді під інформаційною безпекою можна розуміти здатність суб'єкта зберігати свої системостворюючі властивості, основні характеристики при патогенних дезорганізуючих, деструктивних впливах на кіберпростір, інформаційно-комунікаційні технології.

На думку прибічників цього погляду, безпека і забезпечення безпеки становлять собою різні поняття, через те, що безпека виражає характеристику стану соціальної спільноти, тоді як забезпечення безпеки — діяльнісну характеристику, тобто діяльність органів державної влади і управління з підтримання безпеки. У цьому плані безпека усвідомлюється як основа цілепокладання політики, а забезпечення безпеки — як діяльність з досягнення безпечного стану суспільства або соціальної групи. Цієї ж думки дотримується і автор.

Цікавою є думка відомого українського дослідника проблем інформаційної безпеки РЛ. Калюжного, який вважає, що інформаційна безпека — вид суспільних інформаційних правовідносин щодо створення, підтримки, охорони та захисту бажаних для людини, суспільства і держави безпечних умов життєдіяльності; суспільних правовідносин, пов'язаних із створенням, розповсюдженням, зберіганням та використанням інформації.

У цілому ж інформаційна безпека покликана забезпечити реалізацію національних інтересів за допомогою усього арсеналу засобів, що є в її розпорядженні. У цьому сенсі ми вважаємо, що найвищий сенс політики інформаційної безпеки — вільний розвиток і процвітання суспільства.

Отже, інформаційна безпека являє собою одне з найважливіших понять у науці і різних сферах людської діяльності. Сутність і комплексність цього поняття виявляється характером сучасного інформаційного суспільства. Аналіз різних підходів до визначення змісту поняття "інформаційна безпека" надає можливість зауважити про недоцільність суворого обрання тієї чи іншої позиції. Наведеш вище погляди, а вірніше підходи до визначення поняття інформаційної безпеки дають змогу розглядати дану проблему більш комплексно і системно, додати знань про цей багатогранний феномен. Більш того, на наше переконання, найбільш прийнятним є інтегральний підхід, за якого інформаційна безпека визначатиметься за допомогою окреслення найбільш важливих її сутнісних ознак з урахуванням постійної динаміки інформаційних систем.

Такий підхід надав можливість дійти висновку, що інформаційна безпека не може розглядатися лише як окремий стан. Безперечно, що це є і властивістю, атрибутом інформаційного суспільства, діяльністю і результатом діяльності людини, спрямованої на забезпечення певного рівня безпеки в інформаційній сфері. Інформаційна безпека має враховувати майбутнє, а отже, вона не є станом, а становить собою процес. Таким чином, інформаційну безпеку слід розглядати крізь органічну єдність ознак, таких як стан, властивість, а також управління загрозами і небезпеками, за якого забезпечується обрання оптимального шляху їх усунення і мінімізації впливу негативних наслідків.

Також наголосимо, що не підтримуємо позицій тих дослідників, які зводять інформаційну безпеку лише до захисту інформації. Інформаційна безпека за своєї суттю є більш широким поняттям. Отже, інформаційна безпека — багатогранна область діяльності, в якій успіх може принести лише системно-комплексний підхід.

Дослідження сутності інформаційної безпеки має враховувати той факт, що сутність є внутрішнім змістом предмету, який знаходить вираз у стійкій єдності усіх багатоманітних і суперечливих формах буття.

Базовою характеристикою інформаційної безпеки слід вважати імовірність появи загрози підвищеного ризику реалізації загрози або небезпеки для індивіда, суспільства та держави.

Критерієм ефективності забезпечення інформаційної безпеки є високий рівень безпеки при мінімумі відповідних затрат.

Отже, можна говорити про структуру поняття інформаційної безпеки. Основним її елементом є життєво важливі інтереси соціальної системи, які співвідносяться із зовнішніми чинниками у вигляді інтересів наднаціональних або інших національно-державних структур в рамках міжнародного співтовариства. Зсередини національно-державного утворення його життєво важливі інтереси перебувають у взаємодії з інтересами елементів, які складають дане утворення. В якості останніх виступають соціальні групи, еліта, організації, партії, релігійні та етнічні утворення, рухи тощо. Сукупність внутрішніх і зовнішніх інформаційних загроз створюють передумови для порушення безпечного функціонування системи державного управління.

Значимість інформаційно-комунікаційних процесів у сучасному світі дає підстави розглядати забезпечення інформаційної безпеки як одне з глобальних і пріоритетних завдань політики національної безпеки.

Як нами вже зазначалося, національні інтереси в інформаційній сфері є похідними від національних цінностей. Отже, інтереси інформаційної безпеки витікають із таких цінностей, як права людини, свобода, економічне процвітання. Саме тому головним інтересом для України є її виживання як вільної незалежної нації при збереженні фундаментальних цінностей і інститутів безпеки.

Інформаційна безпека виступає як характеристика стабільного, стійкого стану системи, яка при впливі внутрішніх та зовнішніх загроз та небезпек зберігає суттєво важливі характеристики для власного існування. Відтак інформаційна безпека може описуватися за допомогою терміну "гомеостазис".

До характеристик, за допомогою яких можна описати дану систему, належать:

1. доступність — можливість за прийнятний час отримати шукану інформаційну послугу будь-яким суб'єктом виконавчої влади;
2. цілісність — актуальність і несуперечливість інформації, її захищеність від руйнування і несанкціонованої зміни;
3. конфіденційність — захист від несанкціонованого ознайомлення.

Сутність і зміст інформаційної безпеки проявляються по-особливому на кожному з рівнів державного управління, зокрема на:

1. стратегічному — Кабінет Міністрів України;
2. тактичному — центральні органи виконавчої влади;
3. оперативному — місцеві органи виконавчої влади, провідне місце серед яких посідають місцеві державні адміністрації.

Таким чином можна говорити і про прояви інформаційної безпеки у самому процесі її забезпечення, таким чином можна виділити наступні рівні:

1. нормативно-правововий — закони, нормативно-правові акти тощо;
2. адміністративний — дії загального характеру, які вживаються органами державного управління;
3. процедурний — конкретні процедури забезпечення інформаційної безпеки;
4. програмно-технічний — конкретні технічні заходи забезпечення інформаційної безпеки.

Для розкриття сутності та змісту інформаційної безпеки важливим є зв'язок останньої із політикою держави. Складовою частиною політики держави як регулятора суспільних відносин відповідно до гуманістичних начал є обов'язок забезпечення інформаційної безпеки особи, суспільства та держави.

Інформаційна безпека як одна з характеристик стійкого розвитку виступає в якості базової цінності держави. Водночас, ціннісні орієнтації, що ґрунтуються на уявленнях про інформаційну безпеку у різних суспільних груп і окремих осіб, почасти не співпадають. Саме у цьому знаходить свій безпосередній вираз вплив держави, яка за допомогою системи методів виражає загальні цінності у сфері інформаційної безпеки.

## 1.2 Категорійно-понятійна система інформаційної безпеки

Основні поняття та визначення інформаційної безпеки описані та затверджені на законодавчому рівні в Законі України «Про основні засади забезпечення кібербезпеки України». Оперування основними термінами та поняттями необхідна складова для вміння кожного кваліфікованого спеціаліста формувати свої професійні думки та грамотно їх виражати. Також вузько-спеціалізована лексика допомагає в спілкуванні та взаємному розумінні між спеціалістами однієї сфери.

Законі України «Про основні засади забезпечення кібербезпеки України» визначає правові та організаційні основи забезпечення захисту життєво важливих інтересів людини і громадянина, суспільства та держави, національних інтересів України у кіберпросторі, основні цілі, напрями та принципи державної політики у сфері кібербезпеки, повноваження державних органів, підприємств, установ, організацій, осіб та громадян у цій сфері, основні засади координації їхньої діяльності із забезпечення кібербезпеки.

У цьому Законі наведені нижче терміни вживаються в такому значенні:

1. індикатори кіберзагроз — показники (технічні дані), що використовуються для виявлення та реагування на кіберзагрози;
2. інформація про інцидент кібербезпеки — відомості про обставини кіберінциденту, зокрема про те, які об’єкти кіберзахисту і за яких умов зазнали кібератаки, які з них успішно виявлені, нейтралізовані, яким запобігли за допомогою яких засобів кіберзахисту, у тому числі з використанням яких індикаторів кіберзагроз;
3. інцидент кібербезпеки (далі - кіберінцидент) — подія або ряд несприятливих подій ненавмисного характеру (природного, технічного, технологічного, помилкового, у тому числі внаслідок дії людського фактора) та/або таких, що мають ознаки можливої (потенційної) кібератаки, які становлять загрозу безпеці систем електронних комунікацій, систем управління технологічними процесами, створюють імовірність порушення штатного режиму функціонування таких систем (у тому числі зриву та/або блокування роботи системи, та/або несанкціонованого управління її ресурсами), ставлять під загрозу безпеку (захищеність) електронних інформаційних ресурсів;
4. кібератака — спрямовані (навмисні) дії в кіберпросторі, які здійснюються за допомогою засобів електронних комунікацій (включаючи інформаційно-комунікаційні технології, програмні, програмно-апаратні засоби, інші технічні та технологічні засоби і обладнання) та спрямовані на досягнення однієї або сукупності таких цілей: порушення конфіденційності, цілісності, доступності електронних інформаційних ресурсів, що обробляються (передаються, зберігаються) в комунікаційних та/або технологічних системах, отримання несанкціонованого доступу до таких ресурсів; порушення безпеки, сталого, надійного та штатного режиму функціонування комунікаційних та/або технологічних систем; використання комунікаційної системи, її ресурсів та засобів електронних комунікацій для здійснення кібератак на інші об’єкти кіберзахисту;
5. кібербезпека — захищеність життєво важливих інтересів людини і громадянина, суспільства та держави під час використання кіберпростору, за якої забезпечуються сталий розвиток інформаційного суспільства та цифрового комунікативного середовища, своєчасне виявлення, запобігання і нейтралізація реальних і потенційних загроз національній безпеці України у кіберпросторі;
6. кіберзагроза — наявні та потенційно можливі явища і чинники, що створюють небезпеку життєво важливим національним інтересам України у кіберпросторі, справляють негативний вплив на стан кібербезпеки держави, кібербезпеку та кіберзахист її об’єктів;
7. кіберзахист — сукупність організаційних, правових, інженерно-технічних заходів, а також заходів криптографічного та технічного захисту інформації, спрямованих на запобігання кіберінцидентам, виявлення та захист від кібератак, ліквідацію їх наслідків, відновлення сталості і надійності функціонування комунікаційних, технологічних систем;
8. 11) кіберпростір -— середовище (віртуальний простір), яке надає можливості для здійснення комунікацій та/або реалізації суспільних відносин, утворене в результаті функціонування сумісних (з’єднаних) комунікаційних систем та забезпечення електронних комунікацій з використанням мережі Інтернет та/або інших глобальних мереж передачі даних;
9. 15) критична інформаційна інфраструктура — сукупність об’єктів критичної інформаційної інфраструктури;
10. 16) критично важливі об’єкти інфраструктури (далі - об’єкти критичної інфраструктури) — підприємства, установи та організації незалежно від форми власності, діяльність яких безпосередньо пов’язана з технологічними процесами та/або наданням послуг, що мають велике значення для економіки та промисловості, функціонування суспільства та безпеки населення, виведення з ладу або порушення функціонування яких може справити негативний вплив на стан національної безпеки і оборони України, навколишнього природного середовища, заподіяти майнову шкоду та/або становити загрозу для життя і здоров’я людей;
11. 17) Національна телекомунікаційна мережа — сукупність спеціальних телекомунікаційних систем (мереж), систем спеціального зв’язку, інших комунікаційних систем, які використовуються в інтересах органів державної влади та органів місцевого самоврядування, правоохоронних органів та військових формувань, утворених відповідно до закону, призначена для обігу (передавання, приймання, створення, оброблення, зберігання) та захисту національних інформаційних ресурсів, забезпечення захищених електронних комунікацій, надання спектра сучасних захищених інформаційно-комунікаційних (мультисервісних) послуг в інтересах здійснення управління державою у мирний час, в умовах надзвичайного стану та в особливий період, та яка є мережею (системою) подвійного призначення з використанням частини її ресурсу для надання послуг, зокрема з кіберзахисту, іншим споживачам;
12. 18) національні електронні інформаційні ресурси (далі - національні інформаційні ресурси) — систематизовані електронні інформаційні ресурси, які містять інформацію незалежно від виду, змісту, форми, часу і місця її створення (включаючи публічну інформацію, державні інформаційні ресурси та іншу інформацію), призначену для задоволення життєво важливих суспільних потреб громадянина, особи, суспільства і держави. Під електронними інформаційними ресурсами розуміється будь-яка інформація, що створена, записана, оброблена або збережена у цифровій чи іншій нематеріальній формі за допомогою електронних, магнітних, електромагнітних, оптичних, технічних, програмних або інших засобів;
13. 19) об’єкт критичної інформаційної інфраструктури — комунікаційна або технологічна система об’єкта критичної інфраструктури, кібератака на яку безпосередньо вплине на стале функціонування такого об’єкта критичної інфраструктури;

## 1.3 Криптографічний захист інформації як основа інформаційної безпеки

Тривалий час під криптографією розумілось лише шифрування — процес перетворення звичайної інформації (відкритого тексту) в незрозуміле «сміття» (тобто, шифротекст). Дешифрування — зворотний процес відтворення інформації із шифротексту. Шифром називається пара алгоритмів шифрування/розшифрування. Дія шифру керується як алгоритмами, та, в кожному випадку, ключем. Ключ — секретний параметр (в ідеалі, відомий лише двом сторонам) для окремого контексту під час передачі повідомлення. Ключі мають велику важливість, оскільки без змінних ключів алгоритми шифрування легко зламуються і непридатні для використання в більшості випадків. Історично склалось так, що шифри часто використовуються для шифрування та дешифрування, без виконання додаткових процедур, таких як аутентифікація або перевірка цілісності.

В англійській мові слова криптографії та криптології інколи мають однакове значення, в той час, як деколи під криптографією може розумітись використання та дослідження технологій шифрування, а під криптологією — дослідження криптографії та криптології.

Дослідження характеристик мов, що будь-яким чином стосуються криптології, таких як частоти появи певних літер, комбінацій літер, загальні шаблони тощо, називається криптолінгвістикою.

### 1.3.1 Історія криптографічного захисту інформації

До нашого часу криптографія займалася виключно забезпеченням конфіденційності повідомлень (тобто шифруванням) — 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, ідентифікування відправника/одержувача (аутентифікація), цифрові підписи, інтерактивні підтвердження, та технології безпечного спілкування тощо.

Найперші форми тайнопису вимагали не більше, ніж аналог олівця та паперу, оскільки в ті часи більшість людей не могли читати. Поширення писемності, або писемності серед ворогів, викликало потребу саме в криптографії. Основними типами класичних шифрів є перестановочні шифри, які змінюють порядок літер в повідомленні, та підстановочні шифри, які систематично замінюють літери або групи літер іншими літерами або групами літер. Прості варіанти обох типів пропонували слабкий захист від досвідчених супротивників. Одним із ранніх підстановочних шифрів був шифр Цезаря, в якому кожна літера в повідомленні замінювалась літерою через декілька позицій із абетки. Цей шифр отримав ім'я Юлія Цезаря, який його використовував, зі зсувом в 3 позиції, для спілкування з генералами під час військових кампаній, подібно до коду EXCESS-3 в булевій алгебрі.

Шляхом застосування шифрування намагаються зберегти зміст спілкування в таємниці, подібно до шпигунів, військових лідерів, та дипломатів. Збереглися також відомості про деякі з ранніх єврейських шифрів. Застосування криптографії радиться в Камасутрі як спосіб спілкування закоханих без ризику незручного викриття. Стеганографія (тобто, приховування факту наявності повідомлення взагалі) також була розроблена в давні часи. Зокрема, Геродот приховав повідомлення — татуювання на поголеній голові раба — під новим волоссям. До сучасних прикладів стеганографії належать невидимі чорнила, мікрокрапки, цифрові водяні знаки, що застосовуються для приховування інформації.

Шифртексти, отримані від класичних шифрів (та деяких сучасних), завжди видають деяку статистичну інформацію про текст повідомлення, що може бути використано для зламу. Після відкриття частотного аналізу (можливо, арабським вченим аль-Кінді) в 9-му столітті, майже всі такі шифри стали більш-менш легко зламними досвідченим фахівцем. Класичні шифри зберегли популярність, в основному, у вигляді головоломок (див. криптограмма). Майже всі шифри залишались беззахисними перед криптоаналізом з використанням частотного аналізу до винаходу поліалфавітного шифру, швидше за все, Альберта Леоном-Баттіста приблизно в 1467 році (хоча, існують свідчення того, що знання про такі шифри існували серед арабських вчених). Винахід Альберті полягав у тому, щоб використовувати різні шифри (наприклад, алфавіти підстановки) для різних частин повідомлення. Йому також належить винахід того, що може вважатись першим шифрувальним приладом: колесо, що частково реалізовувало його винахід (див. Шифрувальний диск Альберті). В поліалфавітному шифрі Віженера (англ. Vigenère cipher), алгоритм шифрування використовує ключове слово, яке керує підстановкою літер в залежності від того, яка літера ключового слова використовується. В середині 1800-тих, Чарльз Беббідж показав, що поліалфавітні шифри цього типу залишились частково беззахисними перед частотним аналізом.

Енігма — автомат, варіанти якого використовувались німецькими військовими починаючи з другої половини 1920-х і до кінця Другої світової війни. Цей автомат реалізовував складний електро-механічний поліалфавітний шифр для захисту таємних повідомлень. Злам шифру Енігми в Бюро Шифрів, та, слідом за цим, дешифрування повідомлень в Блетчі Парк (англ. Bletchley Park), було важливим чинником перемоги Союзників у війні.

Хоча частотний аналіз є потужною та загальною технікою, шифрування, на практиці, часто було ефективним; багато із криптоаналітиків не знали цю техніку. Дешифрування повідомлень без частотного аналізу практично означало необхідність знання використаного шифру, спонукаючи, таким чином, до шпигунства, підкупу, крадіжок, зрад, тощо для отримання алгоритму. Згодом, в 19-му столітті, було визнано, що збереження алгоритму шифрування в таємниці не забезпечує захист від зламу; насправді, було встановлено, що будь-яка адекватна криптографічна схема залишається у безпеці, навіть за умови доступу сторонніх. Збереження в таємниці ключа має бути достатньою умовою захисту інформації нормальним шифром. Цей фундаментальний принцип було вперше проголошено в 1883 Огюстом Керкгофсом, і загальновідомий як принцип Керкгоффза; різкіший варіант озвучив Клод Шеннон як максиму Шеннона — ворог знає систему.

Було створено різні механічні прилади та інструменти для допомоги в шифруванні. Одним з найперших є скітала в стародавній Греції, палиця, що, як вважається, використовувалась Спартанцями як перестановочний шифр. В середньовіччя, було винайдено інші засоби, такі як дірковий шифр, що також використовувася для часткової стеганографії. Разом із винаходом поліалфавітних шифрів, було розроблено досконаліші засоби, такі як власний винахід Альберті — шифрувальний диск, табула ректа Йогана Тритеміуса, та мультициліндр Томаса Джефферсона (повторно винайдений Базерієсом приблизно в 1900 році). Декілька механічних шифрувально/дешифрувальних приладів було створено на початку 20-го століття і багато запатентовано, серед них роторні машини — найвідомішою серед них є Енігма — автомат, що використовувася Німеччиною з кінця 20-тих і до кінця Другої світової війни.[5] Шифри, реалізовані прикладами покращених варіантів цих схем призвели до істотного підвищення криптоаналітичної складності після Другої світової війни.

Поява цифрових комп'ютерів та електроніки після Другої світової війни зробило можливим появу складніших шифрів. Більше того, комп'ютери дозволяли шифрувати будь-які дані, які можна представити в комп'ютері у двійковому виді, на відміну від класичних шифрів, які розроблялись для шифрування письмових текстів. Це зробило непридатними для застосування лінгвістичні підходи в криптоаналізі. Багато комп'ютерних шифрів можна характеризувати за їхньою роботою з послідовностями бінарних бітів (інколи в блоках або групах), на відміну від класичних та механічних схем, які, зазвичай, працюють безпосередньо з літерами. Однак, комп'ютери також знайшли застосування у криптоаналізі, що, в певній мірі, компенсувало підвищення складності шифрів. Тим не менше, гарні сучасні шифри залишались попереду криптоаналізу; як правило, використання якісних шифрів дуже ефективне (тобто, швидке і вимагає небагато ресурсів), в той час як злам цих шифрів потребує набагато більших зусиль ніж раніше, що робить криптоаналіз настільки неефективним та непрактичним, що злам стає практично неможливим.

Широкі академічні дослідження криптографії з'явились порівняно нещодавно — починаючи з середини 1970-тих, разом із появою відкритої специфікації стандарту DES (Data Encryption Standard) Національного Бюро Стандартів США, публікацій Діффі-Хелмана та оприлюдненням алгоритму RSA. Відтоді, криптографія перетворилась на загальнопоширений інструмент для передачі даних, в комп'ютерних мережах, та захисті інформації взагалі. Сучасний рівень безпеки багатьох криптографічних методів базується на складності деяких обчислювальних проблем, таких як розклад цілих чисел, або проблеми з дискретними логарифмами. В багатьох випадках існують докази безпечності криптографічних методів лише за умови неможливості ефективного розв'язання певної обчислювальної проблеми. За одним суттєвим винятком — схема одноразових блокнотів.

Разом із пам'яттю про історію криптографії, розробники криптографічних алгоритмів та систем також мають брати до уваги майбутній поступ технологій в своїх розробках. Наприклад, постійне підвищення обчислювальної потужності комп'ютерів розширило поле для атак грубої сили. Тому, відповідно і оновлюються стандарти в сенсі вибору довжини ключа. Можливі наслідки розвитку квантових комп'ютерів вже враховуються деякими розробниками криптографічних систем; анонсована поява малих реалізацій цих комп'ютерів робить важливою попередню підготовку.

Взагалі кажучи, до початку 20-го століття, криптографія, в основному, була пов'язанна з лінгвістичними схемами. Після того, як основний акцент було зміщено, зараз криптографія інтенсивно використовує математичний апарат, включно з теорією інформації, теорією обчислювальної складності, статистикою, комбінаторикою, абстрактною алгеброю та теорією чисел. Криптографія є також відгалуженням інженерії, але не звичним, оскільки вона має справу з активним, розумним та винахідливим супротивником; більшість інших видів інженерних наук мають справу з нейтральними силами природи. Існують дослідження з приводу взаємозв'язків між криптографічними проблемами та квантовою фізикою.

## 1.4 Сучасна криптографія

### 1.4.1 Симетричне шифрування

До алгоритмів симетричного шифрування належать методи шифрування, в яких і відправник, і отримувач повідомлення мають однаковий ключ (або, що менш поширено, ключі різні, але споріднені та легко обчислюються). Ці алгоритми шифрування були єдиними загально відомими до липня 1976.

Сучасні дослідження симетричних алгоритмів шифрування зосереджено, в основному, навколо блочних та потокових алгоритмів шифрування та їхнього застосування. Блочний шифр подібний до поліалфавітного шифру Алберті: блочні шифри отримують фрагмент відкритого тексту та ключ, і видають на виході шифротекст такого самого розміру. Оскільки повідомлення зазвичай довші за один блок, потрібен деякий метод склеювання послідовних блоків. Було розроблено декілька методів, що відрізняються в різних аспектах. Вони є режимами дії блочних шифрів та мають обережно обиратись під час застосування блочного шифру в криптосистемі.

Шифри Data Encryption Standard (DES) та Advanced Encryption Standard (AES) є стандартами блочних шифрів, затверджених урядом США (однак, стандартизацію DES було скасовано після прийняття стандарту AES). Не зважаючи на те, що стандарт DES було визнано застарілим, він (та особливо його все ще дійсний варіант triple-DES) залишається досить популярним; він використовується в багатьох випадках, від шифрування в банкоматах до забезпечення приватності електронного листування та безпечного доступу до віддалених терміналів. Було також розроблено багато інших шифрів різної якості. Багато з них було зламано.

Потокові шифри, на відміну від блочних, створюють ключ довільної довжини, що накладається на відкритий текст побітово або політерно, в дечому подібно до одноразової дошки. В потокових шифрах, потік шифротексту обчислюється на основі внутрішнього стану алгоритму, який змінюється протягом його дії. Зміна стану керується ключем, та, в деяких алгоритмах, ще і потоком відкритого тексту. RC4 є прикладом добре відомого, та широко розповсюдженого потокового шифру.

Криптографічні гешувальні функції (англ. cryptographic hash functions, або англ. message digest functions) не обов'язково використовують ключі, але часто використовуються і є важливим класом криптографічних алгоритмів. Ці функції отримують дані (часто, ціле повідомлення), та обчислюють коротке, фіксованого розміру число (хеш). Якісні хешувальні функції створені таким чином, що дуже важко знайти колізії (два відкритих тексти, що мають однакове значення хешу).

Коди аутентифікації повідомлень (англ. Message authentication code, MAC) подібні до криптографічних хешувальних функцій, за виключенням того, що вони використовують секретний ключ для аутентифікації значення хешу при отриманні повідомлення. Ці функції пропонують захист проти атак на прості хешувальні функції.

### 1.4.2. Асиметричне шифрування

Асиметричне шифрування (або криптографічна система з відкритим ключем) - система шифрування та/або електронного цифрового підпису (ЕЦП), при якій відкритий ключ передається з відкритого (тобто незахищеному, доступному для спостереження) каналу, і використовується для перевірки ЕЦП і для шифрування повідомлення. Для генерації ЕЦП і для розшифрування повідомлення використовується секретний ключ. Криптографічні системи з відкритим ключем в даний час широко застосовуються в різних мережевих протоколах, зокрема, в протоколах TLS і його попереднику SSL (що лежать в основі HTTPS), в SSH.

Ідея криптографії з відкритим ключем дуже тісно пов'язана з ідеєю односторонніх функцій , тобто таких функцій f (x), що за відомим x досить просто знайти значення f (x), тоді як визначення x з f (x) складно в сенсі теорії.

Але сама одностороння функція марна в застосуванні: нею можна зашифрувати повідомлення, але розшифрувати не можна. Тому криптографія з відкритим ключем використовує односторонні функції з лазівкою. Лазівка - це якийсь секрет, який допомагає розшифрувати. Тобто існує такий y, що знаючи f (x), можна обчислити x.

Алгоритми криптосистеми з відкритим ключем можна використовувати:

1. Як самостійні засоби для захисту переданої та збереженої інформації
2. Як засоби розподілу ключів. Звичайно за допомогою алгоритмів криптосистем з відкритим ключем розподіляють ключі, малі за об'ємом.. А саму передачу великих інформаційних потоків здійснюють за допомогою інших алгоритмів.
3. Як засоби аутентифікації користувачів.

Переваги:

1. Перевага асиметричних шифрів перед симетричними шифрами полягає у відсутності необхідності попередньої передачі особистого ключа по надійному каналу.
2. У симетричної криптографії ключ тримається в секреті для обох сторін, а в асиметричної криптосистеми тільки один секретний.
3. При симетричному шифруванні необхідно оновлювати ключ після кожного факту передачі, тоді як в асиметричних криптосистемах пару (E, D) можна не змінювати значний час.
4. У великих мережах число ключів в асиметричної криптосистеми значно менше, ніж у симетричною.

Недоліки:

* 1. Перевага алгоритму симетричного шифрування над несиметричним полягає в тому, що в перший відносно легко внести зміни.
  2. Хоча повідомлення надійно шифруються, але «засвічуються» одержувач і відправник самим фактом пересилання шифрованого повідомлення.
  3. Несиметрічні алгоритми використовують більш довгі ключі, ніж симетричні.

Нижче наведена таблиця, що зіставляє довжину ключа симетричного алгоритму з довжиною ключа несиметричного алгоритму з аналогічною криптостійкість.

Таблиця 1.1 — Довжини симетричних і несиметричних ключів

|  |  |
| --- | --- |
| Довжина симетричного ключа, біт | Довжина несиметричного ключа, біт |
| 56 | 384 |
| 64 | 512 |
| 80 | 768 |
| 112 | 1792 |
| 128 | 2304 |

Процес шифрування-розшифрування з використанням пари ключів проходить на два-три порядки повільніше, ніж шифрування-розшифрування того ж тексту симетричним алгоритмом.

У чистому вигляді асиметричні криптосистеми вимагають істотно великих обчислювальних ресурсів.

## 1.5 Атака на шифр і стійкість шифру

Під атакою на шифр розуміють спробу розтину цього шифру.

Під стійкістю шифру розуміють здібності шифру протистояти всіляким атакам на нього.

Поняття стійкості шифру є центральним для криптографії. Хоча якісно зрозуміти його досить легко, але отримання строгих доказових оцінок стійкості для кожного конкретного шифру - проблема невирішена. Це пояснюється тим, що до цих пір немає необхідних для вирішення такої проблеми математичних результатів. Тому стійкість конкретного шифру оцінюється тільки шляхом всіляких спроб його розтину і залежить від кваліфікації криптоаналітиків атакуючих шифр. Останню процедуру іноді називають перевіркою стійкості.

Важливим підготовчим етапом для перевірки стійкості шифру є продумування різних передбачуваних можливостей, за допомогою яких супротивник може атакувати шифр. Поява таких можливостей у супротивника зазвичай не залежить від криптографії, це є деякою зовнішньої підказкою та суттєво впливає на стійкість шифру. Тому оцінки стійкості шифру завжди містять ті припущення про противника, в умовах яких ці ​​оцінки отримані.

Насамперед, зазвичай вважається, що супротивник знає сам шифр і має можливості для його попереднього вивчення. Противник також знає деякі характеристики відкритих текстів ( захищаємої інформації ), наприклад, загальну тематику повідомлень, їх стиль, деякі стандарти, формати і т.д.

З більш специфічних наведемо ще три приклади можливостей супротивника :

1. ▪ противник може перехоплювати всі шифровані повідомлення, але не має відповідних їм відкритих текстів;
2. ▪ противник може перехоплювати всі шифровані повідомлення і добувати відповідні їм відкриті тексти;
3. ▪ противник має доступ до шифру (але не до ключів!) і тому може зашифровувати і дешифрувати будь-яку інформацію.

Протягом багатьох століть серед фахівців не вщухали суперечки про стійкість шифрів і про можливість побудови абсолютно стійкого шифру. Наведемо два характерних висловлювання з цього приводу.

Англійський математик Чарльз Беббідж:

« Кожна людина, навіть якщо він не знайомий з технікою розтину шифрів, твердо вважає, що зможе винайти абсолютно стійкий шифр, і чим більше розумний і освічений ця людина, тим більш твердо це переконання. Я сам поділяв цю впевненість протягом багатьох років ».

«Батько кібернетики » Норберт Вінер:

«Будь-який шифр може бути розкритий, якщо тільки в цьому є нагальна необхідність і інформація, яку передбачається отримати, коштує витрачених коштів, зусиль і часу ... »

# Висновки до розділу

На сьогоднішній день інформація є дуже важливою для людей, від неї на пряму залежить майбутнє людини. Саме цим і займається криптографія - методом перетворення (шифрування) інформації з метою її захисту від незаконних користувачів за для збереження її конфіденційності та цілісності. Яка являється одним з гілок криптпології.

Що ж до криптографічних алгоритмів вони мають кілька типів реалізації це: програмна, апаратна і програмно-апаратна, які доповнюють одна одну. Що один що інший тип криптографічних алгоритмів мають свої недоліки та позитивні сторони використання. А зараз точно зрозуміло одне, що сучасне суспільство вже не обійдеться без захисту інформації, а отже і без криптології та різноманітних методів шифрування.

# РОЗДІЛ 2. ЗАГАЛЬНІ ВІДОМОСТІ ПРО ГЕНЕРАТОРИ ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Другий розділ у собі передбачає розгляд концептуальних понять про генератори псевдовипадкових послідовностей та методи з використанням яких їх поява стає можливою. Розглянуто основні принципи побудови генераторів. Для глибшого розуміння предмету реалізації правильним є розпочати ознайомлення із історичної довідки.

## 2.1 Розвиток випадкових послідовностей

Випадкові послідовності свій початок взяли з простих гральних кубиків, що були в широкому вжитку в азартних іграх раніше і в настільних іграх тепер, є найпростішим істинним генератором випадкових чисел. 1890 року англійський дослідник Френсіс Гальтон описав спосіб застосування гральних кубиків, щоб генерувати випадкові числа з науковою метою.

Випадкова величина - це величина, яка приймає в результаті досліду одне з безлічі значень, причому поява того чи іншого значення цієї величини до її вимірювання не можна точно передбачити. Тепер скажімо, що дані числа вийшли набором однієї з верхніх рядків клавіатури. Ця послідовність не буде випадкова. Послідовність буде випадковою тільки якщо між символами, немає залежності. Наприклад, якби дані символи з'явилися в результаті витягування діжок в лото, то послідовність була б випадковою.

Більш складний приклад або число π. Послідовність цифр в числі π вважається випадковою. Нехай генератор грунтується на виведенні біт представлення числа π, починаючи з якоїсь невідомої точки. Такий генератор, можливо і пройде «тест на наступний біт», так як π, мабуть, є випадковою послідовністю. Однак цей підхід не є криптографічно надійним - якщо криптоаналітик визначить, який біт числа π використовується в даний момент, він зможе обчислити і всі попередні і подальші біти.

Даний приклад накладає ще одне обмеження на генератори випадкових послідовностей. Криптоаналітик не повинен мати можливості передбачити роботу генератора.

Подальшим розвитком апаратних генераторів випадкових чисел можна вважати спеціальні пристрої — лототрони, які використовують для генерації чисел в лото і кено. Вони головним чином складаються з барабана, що перемішує кульки з числами, і пристрою, що витягає ці кульки з нього за чергою. Однак такий метод генерації є дуже повільним і непридатним для автоматичної генерації великих масивів даних.

Для прикладних завдань були необхідні великі масиви даних. 1939 року Моріс Кендалл і Б. Бабінгтон-Сміт побудували першу машину, яка генерувала випадкові числа щоб побудувати таблицю, що містить 100 000 випадкових чисел. А через 16 років корпорація RAND з використанням спеціальних пристроїв побудувала таблицю з мільйона випадкових чисел. Попри те, що 1996 року Джордж Марсалья оживив табличний метод, побудувавши 650 мегабайтів випадкових чисел, коло застосування таких таблиць дуже вузьке.

Набагато більшого поширення набули генератори випадкових чисел, що генерують їх у реальному часі. 1951 року в комп'ютер Ferranti Mark 1 було включено програму, яка генерувала випадкові числа, використовуючи шум резистора. Ідея створення цієї програми належала Алану Тюрінгу. А 1957 року винайдено машину ERNIE (Electronic Random Number Indicator Equipment), четверту версію якої представлено 2004 року. Це пристрій спочатку був призначений для генерації номерів виграшних облігацій у британській лотереї.[1]

## 2.2 Генератори псевдовипадкових послідовностей

Однією з основних умов забезпечення необхідного рівня гарантій криптографічної стійкості є застосування та відповідне управління ключовими даними. Рівень гарантій криптографічної стійкості дозволяє надавати з необхідною якістю такі базові послуги безпеки, як конфіденційність, цілісність, автентичність (справжність), неспростовність, доступність тощо. В криптографічних системах створюються спеціальні підсистеми – джерела ключових даних і ключової інформації, а також здійснюється управління ключовими даними (ключами). Під ключовими даними (ключами) розуміється сукупність випадкових або псевдовипадкових значень змінних параметрів криптографічного перетворення інформації, за рахунок яких досягається мета цього перетворення (наприклад, зашифрування, розшифрування, обчислення криптографічного контрольного значення, обчислення електронного цифрового підпису, перевіряння електронного цифрового підпису, формування сертифікату відкритого ключа тощо). Задача генерації випадкових і псевдовипадкових послідовностей, які використовуються в криптографії в якості ключів, загальносистемних параметрів вирішується за допомогою застосування високошвидкісних генераторів випадкових та псевдовипадкових послідовностей. При цьому сучасна класифікація даних генераторів відображає як методи їх побудови, так і обрану термінологію. Відмінністю чисто випадкових послідовностей від псевдовипадкових є те, що псевдовипадкова послідовність може бути відновлена у просторі й часі. Від якості випадковості формування ключів, ключової інформації та системних параметрів суттєво залежить криптографічна стійкість.

Фізичне моделювання випадковості за допомогою таких фізичних явищ як, наприклад, радіоактивне випромінювання або дробовий шум в електронній лампі є досить складним і дорогим, а використання натиснення клавіш і рух миші вимагає зусиль користувача і до того ж не дають повністю справжніх випадкових процесів. Тому замість фізичного моделювання використовують методи математичного моделювання випадковості і генерації випадкових послідовностей у вигляді програм для ЕОМ або спеціалізованих пристроїв.

Ці програми та пристрої хоча і називаються генераторами випадкових чисел, насправді генерують детерміновані послідовності, які тільки здаються випадковими за своїми властивостями і тому називаються псевдовипадковими послідовностями.

Генератор псевдовипадкових чисел (ГПВЧ, англ. Pseudorandom number generator, PRNG) - алгоритм, що генерує послідовність чисел, елементи якої майже незалежні один від одного і підкоряються заданому розподілу (зазвичай рівномірному).

Найважливіша характеристика генератора псевдовипадкових чисел - це інформаційна довжина його періоду, після якого числа будуть або просто повторюватися, або їх можна буде передбачити.

Будь-які програмні ГПВП, які не використовують зовнішніх джерел ентропії і формують чергове число тільки алгебраїчними перетвореннями, не дають чисто випадкових чисел. Послідовність на виході такого генератора виглядає як випадкова, але насправді підпорядковується деякому закону і, як правило, рано чи пізно зациклюється.

Подібні генератори переважно задаються деяким початковим значенням і за допомогою певних алгоритмів отримують з нього випадкові послідовності. В цьому сенсі псевдовипадкові генератори можна розглядати як розповсюджувачі випадковості. Комп'ютери є детермінованими машинами, що завжди роблять саме те, на що вони запрограмовані, і це усуває можливість звертатися до комп'ютерів як до джерела істинної випадковості. Найкраще, на що здатний комп'ютер - це згенерувати псевдовипадкову послідовність, яка хоча і виглядає випадковою, але, насправді, такою не є.

Згенерувати дійсно випадкову послідовність можна лише при апаратній реалізації генератора, який би для отримання випадкових чисел використовував деяке фізичне явище, наприклад, шум, який генерують напівпровідникові прилади, молодші біти оцифрованого звуку, інтервали між перериванням пристроїв або натисканням клавіш, температуру повітря. В сучасних потужних криптосистемах військового призначення використовують генератори випадкових послідовностей (ГВП), які є платами або зовнішніми пристроями, які підключаються до ЕОМ через порт вводу-виводу.

Незважаючи на труднощі, які виникають при проектуванні генераторів псевдовипадкових послідовностей (ГПВП), вони широко використовуються в прикладних комп'ютерних програмах і легко компонуються з усіма типами комп'ютерних систем.

В сучасних умовах для отримання випадкових послідовностей застосовують різноманітні генератори, які поділяються на дві групи - апаратні (фізичні) та програмні.

Хоча такий поділ ГПВП є досить умовним, оскільки більшість генераторів можуть бути реалізовані як програмно, так і апаратно. Це також залежить від того, де саме буде використовуватися той чи інший ГПВП. Наприклад, при розробці комп'ютерних ігор доцільним є використання програмних ГПВП, а, наприклад, у вимірювальній техніці застосовують апаратно реалізовані ГПВП. Винайти якісний ГПВП не так вже й легко. Вчені показали, що якісну послідовність псевдовипадкових чисел неможливо виробляти за допомогою випадково вибраного алгоритму.

В залежності від складності алгоритми забезпечують різні ступені захисту. На перше місце ставиться принципова можливість отримання з перехоплення деякою інформацією про відкрите тексті або використаному ключі. Проблема всієї сучасної криптографії - це відсутність нижньої межі стійкості; довжина ключа задає лише загальний обсяг простору ключів, але завжди є ймовірність вгадати рішення. Залежно від цілей і можливостей криптоаналітика змінюється і стійкість. Розрізняють стійкість ключа (складність розкриття ключа найкращим відомим алгоритмом), стійкість безключового читання, імітостойкость (складність нав'язування неправдивої інформації найкращим відомим алгоритмом) і ймовірність нав'язування неправдивої інформації. Аналогічно можна розрізняти стійкість власне криптоалгоритму, стійкість протоколу, стійкість алгоритму генерації та розповсюдження ключів. В залежності від складності злому алгоритми забезпечують різні ступені захисту. На перше місце ставиться принципова можливість отримання з перехоплення деякою інформацією про відкрите тексті або використаному ключі. Існують безумовно стійкі (або теоретично стійкі), доказово стійкі і імовірно стійкі крипто алгоритми [1-4]. Безумовно (теоретично) стійкі системи створюють шифртексти, що містять недостатню кількість інформації для однозначного визначення відповідних їм текстів (або ключів). У кращому випадку відкритий текст може бути локалізована в підмножині множини всіх відкритих текстів, і його можна лише «вгадати» з мізерно малою ймовірністю. Ніякий метод криптоаналізу, включаючи повний перебір ключів, не дозволяє не тільки визначити ключ або відкритий текст, але навіть отримати деяку інформацію про них. Алгоритм безумовно стійкий, якщо відновлення відкритого тексту неможливо при будь-якому обсязі шифртексту, отриманого криптоаналітиком. В силу своєї непрактичності і високої ресурсовитратності абсолютно стійкі шифри застосовуються тільки в мережах зв'язку з невеликим обсягом переданої інформації, коли є можливість забезпечити всіх абонентів достатнім запасом випадкових ключів і виключити можливість їх повторного застосування: зазвичай це мережі для передачі особливо важливої державної інформації.[1]

## 2.3 Методи отримання псевдовипадкових послідовностей

Розробкою ефективних методів для створення надійних і якісних ГПВП займається багато вчених. На сьогоднішній день, на основі математичних та експериментальних досліджень, було розроблено ряд методів побудови ГПВП, які відрізняються один від одного оперативністю, доступністю, періодом випадкової послідовності, іншими характеристиками. Найбільш поширеними є наступні методи:

* 1. Лінійний конгруентний метод;
  2. Метод середини квадратів;
  3. Дробові ГПВП;
  4. Генератор М - послідовностей;

В залежності від сфери застосування кожен із вище вказаних методів має свої переваги і недоліки.

Головна перевага лінійних конгруентних генераторів в тому, що вони швидко працюють і потребують мало операцій на біт послідовності. Найбільший недолік - передбачуваність таких послідовностей. Такі послідовності ефективно генеруються і демонструють добрі статистичні властивості при дослідженні більшістю емпіричних тестів.

Перший алгоритмічний метод отримання рівномірно розподілених псевдовипадкових чисел запропонував Джон фон Нейман (один з основоположників кібернетики). Метод отримав назву "метод середини квадрата". Суть методу: попереднє випадкове число підноситься до квадрату, а потім з результату витягуються середні цифри. Метод середини квадрата досить добре "перемішує" попереднє число.

Недолік методу полягає в тому, що послідовності, які видають генератори побудовані на основі цього методу мають тенденцію перетворюватися в короткі цикли елементів, які повторюються. Наприклад, якщо який-небудь член послідовності виявиться рівним нулю, то всі наступні члени також будуть нулями.

Дробові генератори також не знайшли широкого застосування. Результати їх роботи, на сьогоднішній день, є далекі від практичного використання. Оскільки для представлення ірраціональних чисел в пам'яті комп'ютера необхідна нескінченна кількість розрядів, що є нездійсненним, а у випадку використання раціональних чисел дуже великою є ймовірність отримання циклів з малими періодами. Крім того, для даного класу генераторів неможливо дати математичне обгрунтування нижньої границі періоду повторення послідовності, тому що вона сильно залежить від вихідних чисел.

До переваг генераторів М-послідовностей можна віднести просту їх, як програмну, так і апаратну реалізацію, а також рівноймовірність випадкових величин, які генеруються. Даний тип генераторів не задовольняють вимогу щодо непередбачуваності чисел, які послідовно генеруються. На практиці для усунення цього недоліку вихід лінійних послідовних машин додатково обробляєтся хеш-функцією або кодується DES алгоритмом.

## 2.4 Принципи побудови генераторів псевдовипадкових послідовностей

Апаратний генератор випадкових послідовностей – це пристрій, що використовуєтсья для створення випадкових послідовностей, замірів параметрів деяких фізичних процесів. Як правило, апаратний генератор випадкових послідовностей складається із джерела ентропії і пристрою, що перетворює значення, отримані з джерела етропії в потрібний формат.

Джерелами ентропії(ДЕ) можуть бути:

1. Підкидання монети;
2. Часові затримки між моментами випромінювання часток в процесі радіоактивного розпаду;
3. Теплові шуми при роботі напівпровідникового діода або резистора;
4. Частотні відхилення генератора частот;
5. Фотоефект – випускання електронів речовиною під дією світла;
6. Звук від мікрофона або відео з під`єднаної камери;
7. Стан деяких блоків пам`яті комп`ютера;

Побудова апаратного генератора випадкових послідовностей затратна робота, в межах якої потрібно вирішити такі питання як налаштуванння діапазонів фіксованих випадкових величин, оцифрування аналогових результатів, ізоляція фізичного джерела випадковості від зовнішніх впливів.

Апаратні ГВП є недетерміновані - ніякий алгоритм не може використовуватися, щоб; визначити послідовність бітів. Таким чином, апаратні ГВП є нечуттєвими до вторгнення або впливу алгоритму демонтування або виявлення. Властивість недетермінізму має особливо важливе значення у специфічних прикладних програмах генерування випадкових чисел, таких як деякі методи наукового і фінансового моделювання, лотерей, які підтримуються урядом, а також для різноманітних технологій безпеки комп'ютера, таких як криптографія і цифровий підпис.

Апаратні ГВП звільняють користувача від необхідності вибору початкового значення. Це робить процес генерації випадкового значення високо ефективним, а якість початкового значення не залежить від навичок програміста.

Апаратні ГВП є особливо важливими коли сервер, такий як Web-сервер або файловий сервер, повинні виконувати шифрування.

В свою чергу апаратні генератори можуть бути комбінованими, тобто вони можуть крім певних фізичних явищ використовувати і певні математичні алгоритми для обробки цих фізичних явищ. Наприклад, деякі прикладні програми або Web-сервер, які використовують апаратний ГВП потребують безперервний доступ до ГВП. який може їх сповільняти. Тому спочатку вони за допомогою апаратного ГВП отримують певне початкове значення, а потім використовують ГПВП для створення багатьох ключів високої ефективності. Навіть якщо апаратний ГВП стає недоступним з деяких причин, ГПВП може продовжувати створювати ключі маючи лише це початкове значення.

Апаратні ГВП від Intel, поєднані з програмним забезпеченням RSA Data Security, забезпечують недорогий розповсюджений, автоматизований і добре налагоджений процес для генерації початкового матеріалу. Крім цього створений початковий матеріал є дійсно випадковим.

## 2.5. Генератори, що використовують фізичні квантові випадкові процеси

### 2.5.1 Фазовий квантовий шум в лазерному промені. (рис. 2.1)

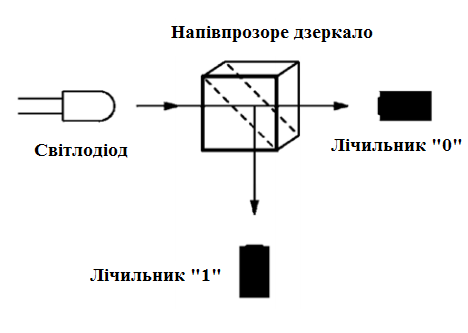


Рисунок 2.1— Схема ГВП, побудованого на базі квантового шуму в лазерному промені

Одним з найбільш надійних способів отримання випадкових послідовностей є ГВП, що реєструє квантовий ефект удару фотонів в дзеркало. На напівпрозоре дзеркало направляються фотони, які генеруються джерелом одиничних фотонів. Фотон може відбитися, а може пройти через напівпрозоре дзеркало з однаковими частками ймовірності. Вибір, котрий «робить» фотон, абсолютно випадковий. На виході системи стоять два лічильника фотонів, які реєструють чи пройшли і відобразились фотони і формують вихідні електричні сигнали [1]. Подібні квантові генератори мають високу швидкість вихідного потоку - до 10-16 Мбіт/с, за якої не спостерігається ніяких кореляцій і виконуються всі статистичні тести. [2]

### 2.5.2 Матриця фотокамери (рис. 2.2)

Більшість джерел світла випускають фотони в абсолютно випадкові моменти часу і кількість фотонів, випущених за одиницю часу буде розділятися на величину, яка є повністю випадковою. Цей факт ліг в основу ГВП, побудованого на базі світлочутливої КМОН-матриці звичайної фотокамери групою вчених з Женевського університету на чолі з Бруно Сангінетті. Кожен піксель матриці «рахує» кількість фотонів, які потрапили на його поверхню за певний проміжок часу. Ці фотони конвертуються в електрони, які потім множаться на множник, певної світлочутливості матриці (рівень ISO). Кількість електронів за один і той же період буде відрізнятися на абсолютно випадкове число. На практиці процес генерації таких випадкових чисел виглядає досить просто: матриця фотокамери засвічується зеленим світлодіодом і робить два знімки з однаковою тривалістю витримки. Потім знімки програмно обробляються для отримання випадкових чисел. За словами розробників, випадкові числа, отримані в результати дослідів з використанням світлочутливої матриці сучасного мобільного телефону, успішно пройшли статистичні тести. Більш того, за рахунок великих розмірів матриці і частоти отримання знімків, розроблений ними ГВП може генерувати випадкові числа з величезною швидкістю (до 3 Гбіт / с). [3]

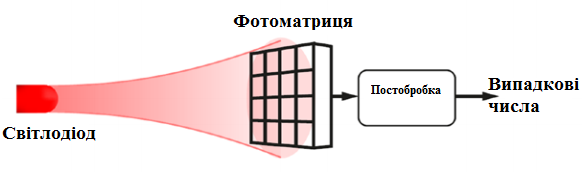


Рисунок 2.2 — Схема ГВП, побудованого на базі фотоматриці

## 2.6 Гнератори, що використовують інші кванові випадкові процеси

### 2.6.1 Тепловий шум(рис. 2.3.)

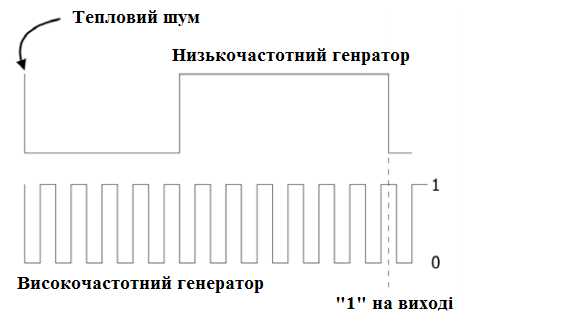
Тепловий шум, так званий шумом Джонсона, генерується всіма пасивними резистивним елементами електричних ланцюгів. Причина його появи - випадковий броунівський рух електронів в резистивному середовищі. Тепловий шум збільшується зі зростанням температури і опору і часто виявляється найсуттєвішою складовою шуму в прецизійних напівпровідникових перетворювачах даних. [4] Одним з успішних прикладів побудови ГВП на базі теплового шуму є генератор, розроблений компанією Intel в 1999 році і який використовується в чіпсетах Intel 800 серії. ГВЧ Intel використовує послідовності випадкових чисел, що отримуються з двох тактових генераторів, частота роботи одного з яких перевищує частоту іншого в 100 разів. Тепловий шум з джерела (напівпровідникового резистора) посилюється і використовується для керування частотою коливань повільного генератора. Випадкові послідовності, отримані в результаті дрейфу (похибки ходу) двох генераторів, проходять подальшу апаратну обробку через «коректор Фон Неймана» для отримання збалансованого розподілу нулів і одиниць. Серед недоліків даного генератора випадкових 

Рисунок 2.3 — Часова діаграма ГВП Intel

послідовностей можна виділити велике енергоспоживання (через кільцевого генератора, що використовується для посилення теплового шуму) і відносно невелику для сучасних потреб швидкість генерації (близько 75 Кбіт / с після постобробки). [5]

### 2.6.2 Цифрова схема з невизначеним станом

У 2008 році інженери компанії Intel почали за розробку нового варіанту генератора випадкових послідовностей, який працює виключно на цифровій основі. Запропоноване ними рішення порушує основне правило цифрового проектування: схема завжди повинна бути в одному з двох певних станів (логічний низький і логічний високий рівні сигналу). Схема ГВЧ складається з пари інверторів, вихід кожного з яких підключений до входу іншого. Якщо на виході першого інвертора буде логічний низький рівень сигналу, то другий інвертор отримає цей рівень на вході і, відповідно, видасть високий рівень сигналу на виході, і навпаки. Додатково в ланцюг додані два транзистора, включення яких дає на вході і виході обох інверторів логічний високий сигнал Кожен період тактового сигналу, при відключенні транзисторів, обидва інвертора прагнуть прийняти протилежне положення, тобто одне з двох стійких станів, генеруючи при цьому один випадковий біт. Дана розробка дозволила позбутися від незручностей аналогових компонентів попереднього варіанту ГВП, значно зменшити енергоспоживання і збільшити швидкість генерації більш ніж в 30 тисяч разів. [6]

### 2.6.3 Лавинний шум (шум лавинного множення)

Джерелами лавинного шуму є PN-переходи, що працюють в режимі зворотного пробою, як це відбувається в стабілітронах (зенеровських діодах). Струм,що генеруєтсья під час лавинного пробою, складається з випадково розподілених шумових викидів, що проходять через зміщений назад перехід. Подібно дробовому шуму, для генерації лавинного шуму потрібна наявність струму, але зазвичай він набагато інтенсивніше. [4] У ГВП на базі такого джерела випадкових чисел зазвичай використовують перехід емітер-база біполярного NPN транзистора через низькиу напругу. Знятий з переходу шум посилюється і перетворюється в цифровий сигнал. Випадкові числа з подібних ГВЧ проходять всі стастатистичні тести, однак швидкість їх генерації вкрай мала - менше 10 Кбіт/с. [7]

### 2.6.4 Фазове тремтіння в кільцевих генераторах

Фазовий тремтіння цифрового сигналу даних (джиттер від англ. jitter) - небажані фазові і / або частотні випадкові відхилення сигналу, що передаються. Виникають внаслідок нестабільності здаючого генератора, змін параметрів лінії передачі у часі і різної швидкості поширення частотних складових одного і того ж сигналу. Оскільки фазовое тремтіння залежить від різних чинників, деякі з яких повністю випадкові, воно може бути використано як джерело випадкових послідовностей. [8] У ГВП на базі такого явища зазвичай порівнюються випадкові затримки проходження сигналу через кільцеві генератори. Найпростіший кільцевий генератор складається з непарного числа інверторів, з'єднаних послідовно, при цьому вихід останнього з'єднаний з входом першого інвертора, утворюючи лінію зворотного зв'язку. частота коливання такого генератора визначається сумою затримок всіх його інверторів, цей час залежить від множини параметрів, що включають в себе тепловий шум в провідниках і напівпровідниках і перешкоди в джерелах живлення. [9] Серед мінусів такого ГВП можна виділити відносно невелику швидкість генерації і велике енергоспоживання.

Були розглянуті основні способи побудування апаратних генераторів випадкових послідовностей. Серед них можна виділити один з найсучасніших і прогресивних способів - генератор випадкових чисел на базі невизначених станів, розроблений інженерами компанії Intel і володіє однією з найвищих швидкостей вихідного потоку (до 3 Гбіт / с) і низьким енергоспоживання.

## 2.3 Методики тестування генераторів псевдовипадкових послідовностей

Визначення якості та надійності генераторів випадкових та псевдовипадкових послідовностей – одна з основних задач сучасної прикладної та теоретичної криптографії, тому що вони використовуються для генерації ключів та інших випадкових параметрів криптосистем. Існують спеціальні методики тестування для оцінки якості випадкових послідовностей. На сьогоднішній день розглядаються групи тестів, які використовуються для аналізу рівня безпеки генераторів псевдовипадкових чисел, а також різні методики інтерпретації отриманих результатів. Для дослідження якості генераторів псевдовипадкових чисел використо- вують дві групи тестів [11] :

1. графічні тести;
2. статистичні тести.

На рис. 1 запропонована класифікація методик тестування генераторів

псевдовипадкових чисел.

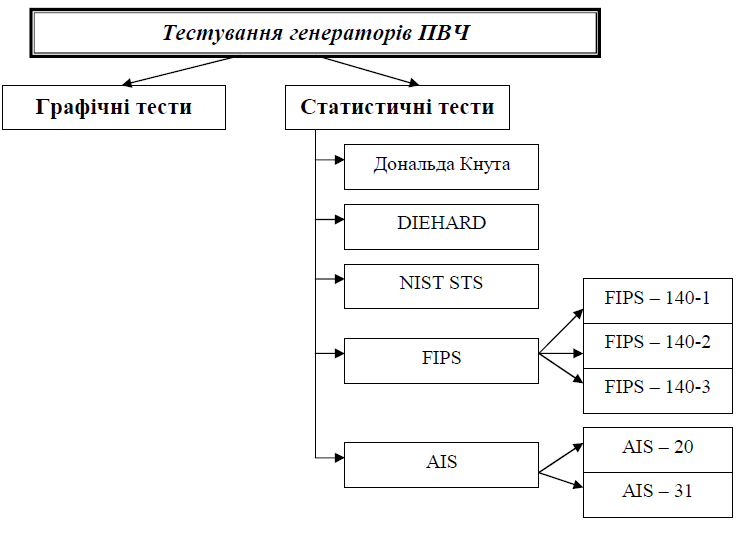


Рисунок 2.4 — Класифікація методик тестування ГВП

При графічному тестуванні статистичні властивості послідовностей відображаються у виді графічних залежностей, за виглядом яких роблять висновки про властивості досліджуваної послідовності. До даної категорії відносяться такі тести: гістограма розподілу елементів послідовності, розподіл на площині, перевірка серій, перевірка на монотонність, автокореляційна функція, профіль лінійної складності, графічний спектральний тест. Результати графічних тестів інтерпретуються людиною, тому висновки можуть бути неоднозначними.Статистичні тести використовуються для перевірки певної нульової гіпотези Н0 щодо випадковості сформованої послідовності. З цією нульовою гіпотезою пов'язана альтернативна гіпотеза про те, що послідовність не випадкова. Для кожного тесту, що застосовується, можна зробити висновок щодо прийняття чи відхилення нульової гіпотези, виходячи із сформованої генератором послідовності. При цьому для кожного тесту має бути вибрана адекватна статистика випадковості, на основі якої може бути прийнята або відхилена нульова гіпотеза. Теоретично для нульової гіпотези розподілення статистики визначається математичними методами. Під час проведення тесту розраховується значення тестової статистики, яке порівнюється з критичним. Якщо значення тестової статистики перевищує критичне значення, нульова гіпотеза для випадковості відхиляється, інакше – приймається [12]. Статистичні тести, на відміну від графічних тестів (результати інтерпретуються користувачами, внаслідок чого можливі відмінності у трактуванні результатів), характеризуються тим, що вони видають чисельну характеристику, яка дозволяє однозначно сказати, пройдено тест чи ні. Нижче описано короткий огляд найбільш відомих на сьогоднішній день методик тестування.

**Методика тестів Дональда Кнута** заснована на статистичному критерії χ2. Обчислюване значення статистики χ2 порівнюється з табличними результатами, і залежно від імовірності появи такої статистики робиться висновок про її ефективність. Серед переваг цих тестів – легкий та швидкий алгоритм виконання. Недолік – невизначеність у трактуванні результатів. Перелік тестів: перевірка незчіплених серій, перевірка інтервалів, перевірка комбінацій, тест збирача купонів, перевірка перестановок, перевірка на монотонність та перевірка кореляції[11].

**Система статистичного тестування DIEHARD** була запропонована Джорджем Марсалі для дослідження статистичних властивостей розроблених ним же конгруентних генераторів. Недоліки системи – параметри тестування жорстко фіксовані, неточність у трактуванні результатів обробки, деякі тести не мають змістовного обґрунтування. У систему входять такі тести: перевірка проміжків між «днями народження», перевірка перестановок, що перетинаються, перевірка рангів матриць 32х32; 31х31; 6х8, літерні тести – перевірка потоку бітів, перевірка розріджених пар, що перетинаються, тест підрахунок одиниць в потоці байт [11].

**Методика FIPS** – дана методика використовується для технологічного аналізу вихідних послідовностей генераторів ПВЧ. Вона є стандартом для контролю криптографічних модулів. Складається з 4-х статистичних тестів: монобітний, «покер-тест», тест серій, тест довжини серій. Для цих тестів, задаються межі для задовільних значень статистичних параметрів. Якщо який-небудь з тестів не пройдений, то вважається, що генератор не пройшов весь комплекс перевірок. [12,13]

**Методика AIS** – застосовується для тестування псевдовипадкових послідовностей. Може використовуватись як у процесі формування послідовності, так і в процесі дослідження, а також для технологічного тестування. Основна ідея полягає в тому, що придатність генераторів псевдовипадкових чисел має бути оцінена з урахуванням криптографічних застосувань, у яких вони використовуються. Система AIS складається з 4-х функціональних класів К1, К2, К3, К4. Ці класи описують набір ієрархічних вимог до генераторів ПВЧ. Аналіз показав, що методика AIS 31 за своєю ефективністю забезпечує такі ж результати тестування як NIST STS. Перевагою AIS 31 є те, що він забезпечує тестування як у реальному часі, так і в процесі Досліджень [12].

**Методика NIST STS** – використовується як засіб комплексного контролю. Статистичні тести NIST призначені для перевірки послідовності, сформованої генератором, на випадковість. Для кожного тесту отримують висновок про прийняття або відхилення нульової гіпотези, ґрунтуючись на сформованій досліджуваним генератором послідовності. Кожен тест заснований на обчисленні значення тестової статистики, яка є функцією даних. Ця статистика використовує обчисленні значення Р-value, за допомогою якого і визначається чи дана послідовність є випадковою. Для тесту слід вибрати рівень значущості α. Якщо значення Р-value ≥ α , то приймається нульова гіпотеза Н0, тобто послідовність є випадковою. Якщо значення Р-value < α , то нульова гіпотеза відхиляється, тобто послідовність не є випадковою. Як правило, значення α вибирається в інтервалі [0.001, 0.01]. До складу пакету NIST входять 15 статистичних тестів, метою яких є визначення міри випадковості двійкових послідовностей, сформованих апаратними або програмними генераторами псевдовипадкових чисел: частотний монобітний, частотний блоковий, тест перевірки серій, найдовшої серії з одиниць, перевірки рангу двійкових матриць, тест на основі дискретного перетворення Фур’є, тест на співпадіння з шаблоном без перекриття, тест шаблонів з перекриттям, універсальний тест Мауера, тест лінійної складності, тест серій, тест на основі апроксимації ентропії, тест накопичених сум, тест випадкових відхилень та тест випадкових відхилень-2[13].

### 2.3.1 Методика тестуванння NIST STS

Набір тестів NIST STS був запропонований в ході проведення конкурсу на новий національний стандарт США блокового шифрування. Цей набір використовувався для дослідження статистичних властивостей кандидатів на новий блоковий шифр. Сьогодні методика тестування, запропонована NIST. є найбільш поширеною у розробників криптографічних засобів захисту інформації. Порядок тестування окремої двійковій послідовності S має такий вигляд:

1. Було висунуто нульову гіпотезу H0 - допущення того, що дана двійкова послідовність S випадкова.
2. Далі розраховується статистика тесту c(S).
3. Використовуючи спеціальну функцію статистики тесту, обчислюється значення ймовірності P = f (c(S)), Pє[0,1].
4. Значення ймовірності P порівнюється з рівнем значущості α є[0.001, 0.01].

Якщо P ≥α , То гіпотеза H0 приймається. В іншому випадку приймається

альтернативна гіпотеза. Пакет містить в собі 16 статистичних тестів. Однак фактично, в залежності від вхідних параметрів обчислюється 189 значень ймовірності P, які можна розглядати як результат роботи окремих тестів. У таблиці 1 перераховуються зібрані дані за всіма тестами із зазначенням кількості значень ймовірності P, які обчислюються, фізичного сенсу статистики і дефекту на визначення якого спрямований тест.

Таким чином, в результаті тестування двійкової послідовності формує вектор значень ймовірності P = {P1, P2, …, P189}. Аналіз значень Pi даного вектора дозволяє вказати дозволяє вказати конкретні дефекти випадковості тестованої послідовності.

Таблиця 1.2 — Статистичні тести NIST STS

|  |  |  |  |
| --- | --- | --- | --- |
| Номер | Статистичний тест | Статистика тесту | Виявлений дефект |
|  | Частотний (монобітний тест) | Нормалізована абсолютна сума значень елементів послідовності | Занадто багато нулів або одиниць в послідовності |
|  | Частотний тест (в серелині блоку) | Захід узгодження кількості одиниць, що спостерігаються з тим, що очікується теоретично | Локалізовані відхилення частоти появи одиниць в блоці від ідеального значення ½ |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Перевірка накопичених сум | максимальне відхилення значень накопиченої суми елементів послідовності від початкової точки відліку (Точка 0) | Велика кількість  одиниць або нулів в початку або в кінці двійковій послідовності |
|  | Перевірка серій | Загальна кількість серій на всій довжині послідовності | Занадто швидка або занадто повільна зміна знака в ході генерації послідовності |
|  | Перевірка максимальної довжини серії у блоці | Захід узгодженості значень максимальної довжини, які спостерігаються, з тими значеннями, що очікуються  теоретично | Відхилення від теоретичного закону розподілу  максимальної довжини серій одиниць |
|  | Перевірка рангу двійкової матриці | Захід узгодженості значень рангів різного порядку, які очікуються  теоретично | Відхилення емпіричного  закону розподілу  значень рангів матриць від  теоретичного, що  вказує на залежність елементів  послідовності |
|  | Cпектральний  аналіз на основі  дискретного  перетворення  Фур’є | Нормалізована різниця кількості частотних компонент, що спостерігаються з  тією, що очікується теоретично, якщо перевищує  пороговий рівень 95% | Виявлення періодичних  доданків (трендів) в двійковій  послідовності |

Продовження таблиці 1.2

Продовження таблиці1.2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Перевірка шаблонів, що перекриваються | Захід узгодженості  кількості шаблонів, що спостерігаються та перекриваються  послідовності з  теоретичними значеннями | Велика кількість m-бітних серій одиниць в послідовності. |
|  | Універсальний  тест Маурера | Сума логарифма відстаней  між l-бітними шаблонами | Можливість стиснення  послідовності |
|  | Ентропійний  тест | Рівень узгодженості  спостережуваного значення ентропії джерела, з тим, що теоретично очікується з випадкового джерела | Нерівномірність  розподілу m –бітних слів в послідовності  (Регулярність властивостей джерела) |
|  | Перевірка  випадкових  відхилень | Рівень узгодженості  візитів, що спостерігаються при випадковому  блуканні з визначеного стану  в середині циклу, в порівнянні з  тим, що очікується теоретично | Відхилення від  теоретичного закону розподілу візитів в конкретний стан при випадковому блуканню |
|  | Перевірка випадкових  відхилень  (варіант) | Загальна кількість візитів при  випадковому блуканню | Відхилення від  теоретично очікуваної  кількості загального  кількості візитів при  випадковому блуканні в  заданий стан |
|  | Послідовний тест | Рівень узгодженості кількості відображених  варіантів m - бітних шаблонів  з тією кількістю, яка очікується теоретично | Нерівномірність  розподілу m – бітних слів в послідовності |

Продовження таблиці1.2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Перевірка стиснення згідно  алгоритму  Лемпеля-Зива | Кількість різних слів у послідовності | Велика ступінь стиснення  послідовності, що тестується і ступенем стиснення очікуваної випадкової послідовності |
|  | Перевірка шаблонів, що не перекриваються | Рівень узгодженості очікуваної кількості  неперіодичних шаблонів в  послідовності з  теоретичним значенням | Велика кількість  заданих  неперіодичних  шаблонів в  послідовності |
|  | Перевірка лінійної складності | Рівень узгодженості очікуваної кількості подій, що лежать в основі появи фіксованої довжини еквівалентного ЛРР для заданого блока з теоретичним | Відхилення емпіричного розподілу довжини еквівалентних ЛРР для послідовностей фіксованої довжини від теоретичного закону розподілу для випадкової послідовності, шо вказує на недостатнб складність послідовності, що тестується |

[13]

# Висновки до розділу

Підсумовуючи матеріал другого розділу можна зробити висновок що дослідження генераторів псевдовипадкових послідовностей взяло свій початок більше як сто років тому, і до сьогоднішнього дня розгляд цієї теми не втратив актуальності. Було наведено перелік основних методів отримання псевдовипадкових послідовностей: лінійний конгруентний метод, метод середини квадратів, дробові ГПВП, генератор М – послідовностей, ГПВП побудовані на основі різних математичних алгоритмів, висвітлено їх переваги та недоліки. Отже можливо зробити висновок про найбільшу надійність і широке застосування генераторів М-послідовностей за їх просту реалізацію та можливість усування недоліку передбачуваності. І навпаки виокремлюється метод дробових генераторів як невикористовуваний, оскільки його існування неможливо втілити практично та обгрунтувати математично. Винесено до розгяду методи перевірки послідовностей, а окремо NIST STS. Пакет статистичних тестів NIST STS є зручним та гнучким інструментом дослідження ГПВП, що застосовуються в криптосистемах. Даний пакет може і повинен бути взятий на озброєння вітчизняними розробниками з-відповідних додатків. На відміну від пакета DIEHARD пакет NIST STS має більшу гнучкість, розширюваність і ефективністю (з точки зору витраченого часу на здійснення тестування генератора). Крім того, пакет NIST STS має велику криптографічну спрямованість, яка досягається шляхом введення в пакет таких тестів як перевірка лінійної складності і універсального тесту Маурера. Розглянута методика тестування та отримані з її використанням результати можуть розглядатися як первинний аналіз генератора.

# 3. Принципи роботи систем автоматизованого проектування та програмованих логічних матриць

У третьому розділі основну увагу буде звернено на технології, що дозволять нам створити генератор псевдовипадкових послідовностей, а саме: система автоматизованого проектування (САПР) та віртуальне середовище MAX+plus II, і програмовану логічну матрицю (ПЛМ) як фізичне середовище реалізації. Буде описано роль і значимість спільної роботи САПР та ПЛМ та їх вплив на сучасних розвиток інформаційних систем.

## 3.1. Основні поняття про системи автоматизованого проектування

### 3.1.1. Місце та роль автоматизованого проектування серед інформаційних технологій

Автоматизоване проектування займає особливе місце серед інформаційних технологій.

По-перше, автоматизація проектування – синтетична дисципліна, її складовими є багато інших сучасних інформаційних технологій. Так, технічне забезпечення систем автоматизованого проектування (САПР) базується на використанні обчислювальних мереж і телекомунікаційних технологій, в САПР використовуються персональні комп’ютери та робочі станції, є приклади застосування мейнфреймів. Математичне забезпечення САПР відрізняється багатством і різномаїттям використовуваних методів обчислювальної математики. Статистики, математичного програмування, дискретної математики, штучного інтелекту. Програмні комплекси САПР відносяться до найбільш складних програмних систем, що базуються на операційних системах Unix, Windows-95/NT, мовах програмування С, С++, Java та інших, сучасних CASE-технологіях, стандартах відкритих систем і обміну даними в комп’ютерних середовищах.

По-друге, знання основ автоматизації проектування та вміння працювати із засобами САПР необхідне практично будь-якому інженеру-розробнику. Комп’ютерами насичені проектні підрозділи, конструкторські бюро та офіси. Робота конструктора за звичайним кульманом, розрахунки за допомогою логарифмічної лінійки чи оформлення звіту на друкарській машинці стали анахронізмом. Установи, що ведуть розробки без САПР чи лише з малим відсотком їх використання, виявляються неконкурентноспроможними як через великі матеріальні та часові витрати, так і через невисоку якість проектів.

Поява перших програм для автоматизації проектування відноситься до початку 60-х років. Тоді були створені програми для розв’язування задач будівельної механіки, аналізу електронних схем, проектування друкованих плат. Подальший розвиток САПР йшов шляхом створення апаратних і програмних засобів машинної графіки, підвищення обчислювальної ефективності програм моделювання та аналізу, розширення областей застосування САПР, спрощення користувацького інтерфейсу, впровадження в САПР елементів штучного інтелекту.

На сьогодні створено величезну кількість програмно-методичних комплексів для САПР із різним ступенем спеціалізації та прикладної орієнтації. У результаті автоматизація проектування стала необхідною складовою частиною підготовки інженерів різних спеціальностей; інженер, що не володіє знаннями та не вміє працювати в САПР, не може вважатися повноцінним спеціалістом.

*Проектування* – одна з найважливіших сфер інженерної діяльності, це та ланка, що пов’язує наукові дослідження і практичну реалізацію. Від термінів та якості проектування значно залежать строки впровадження та якість готової продукції.

*Проектування* — вид цілеспрямованої виробничої діяльності людини чи колективу фахівців із розв'язання задач проектування, спрямованої на створення приладів чи систем, що відповідають технічному завданню, які оптимально задовольняють поставлені вимоги і задовільно функціонують протягом заданого проміжку часу при прогнозованих умовах середовища.

*Автоматизоване проектування* — проектування, при якому окремі перетворення об'єкта й (або) алгоритму його функціонування або алгоритму процесу, а також описи різноманітними мовами здійснюються взаємодією людини та комп'ютера.

Важко переоцінити значення автоматизації проектування для розвитку науки, техніки, народного господарства. Саме з автоматизацією проектування пов’язані принципові можливості створення найскладніших технічних об’єктів як у промисловому, так і цивільному будівництві. Адже проектування об’єктів потребує великих затрат людських і часових ресурсів і саме АП – основний спосіб підвищення продуктивності праці інженерно-технічних працівників, зайнятих проектуванням.

3.1.2. Складові процесу проектування

Проектування як процес, що розвивається в часі, поділяється на стадії, етапи, проектні процедури та операції.

Будівельне проектування в САПР має такі стадії:

* + 1. стадія науково-дослідних робіт
    2. стадія конструкторських робіт (стадія ескізного проекту)
    3. стадія технічного проекту
    4. стадія робочого проекту.

На стадії науково-дослідних робіт (залежно від цільового призначення) вивчаються потреби в отриманні нових проектних рішень, інформаційні, конструктивні та технологічні принципи проектування, прогнозуються можливі значення характеристик і параметрів об’єкта. Результатом є формулювання технічного завдання на розробку нового проекту.

На стадії конструкторських робіт розробляється ескізний проект об’єкта, перевіряються, конкретизуються та коректуються принципи і положення, зазначені в технічному завданні.

На стадії технічного проекту виконується всестороннє опрацювання всіх частин проекту, конкретизуються та деталізуються технічні рішення.

На стадії робочого проекту складається повний комплект конструкторсько-технічної документації, достатньої для будівництва об’єкта.

Процес проектування ділиться на етапи:

1. Етап проектування – частина процесу проектування, що включає формування усіх потрібних описів об’єкта.
2. Проектні процедури – складові частини етапу проектування. Це частина етапу, виконання якої закінчується отриманням проектного рішення. Кожній проектній процедурі відповідає певна задача проектування, яка вирішується у рамках даної процедури (наприклад, оформлення креслення конструкції).
3. Проектна операція – дрібніша складова частина процесу проектування, що входить до складу проектної процедури (наприклад, статичний розрахунок конструкції).

### 3.1.3. Основні відомості про системи автоматизованого проектування

Одним із важливих компонентів сучасного виробництва є системи автоматизованого проектування (САПР).

САПР — організаційно-технічна (людино-машинна) система, що складається з комплексу засобів автоматизації проектування, взаємозв'язаних із необхідними відділами проектної організації чи колективом фахівців (користувачем системи), і виконує автоматизоване проектування.

Комп'ютерна графіка, будучи підсистемою САПР, розв'язує найбільш трудомістку і важливу задачу САПР: автоматизацію розробки і виконання конструкторської документації. Вона забезпечує створення, зберігання і обробку моделей геометричних об'єктів і їх графічне зображення за допомогою комп'ютера.

Використання комп'ютера в конструкторській діяльності значно полегшує підготовку конструкторських та інших графічних документів, звільняючи конструктора від виконання рутинних і трудомістких графічних операцій, скорочує термін виготовлення документів і покращує їх якість. При автоматизованому виконанні креслення створюється "електронний" еквівалент креслення, а замість паперу і креслярських інструментів використовується екран дисплея, клавіатура і маніпулятор "миша".

В автоматизованому проектуванні загальноприйнятими є скорочені терміни:

1. САD – система конструювання (Computer Aided Design);
2. САМ - виробництво за допомогою комп’ютерів (Computer Aided Manufacturing);
3. САЕ - аналітично-розрахункова підсистема (Computer Aided Engineering);
4. АЕС - скорочене позначення для архітектурно-будівельних комп’ютерних систем (Architectural Engineering Computing).

### 3.1.4. Переваги застосування інженерних САПР та їх роль у галузі матеріального виробництва

До появи ЕОМ випуск конкурентноспроможної продукції здійснювався завдяки науковій організації процесу проектування та першокласним фахівцям. Застосування САПР дало безперечні переваги: дозволило впровадити стратегію накопичування та передавання знань і досвіду конструкторів, що забезпечує високу живучість, сприяє розв’язуванню тих практичних задач виробника, що дозволяють проводити прогресивну політику на ринку (знизити виробничі витрати та собівартість, скоротити час розробки та підвищити споживчі якості продукції).

Головною перевагою використання САПР є висока швидкість виходу продукції на ринок. Тут спрацьовує відома в економіці залежність між обсягом продажу товару та часом: виробник, який отримав через пізній вихід продукції на ринок менший прибуток, направить на розробку нового покоління продукції менше коштів, і так – до повної зупинки виробництва. Це підтверджують дослідження компанії Хьюлет-Паккард можливих втрат прибутку виробниками у конкурентній боротьбі: 33% втрат – через піврічне спізнення виходу продукції на ринок; 22% втрат – через підвищення матеріалоємності продукції на 9%; лише 3,5% втрат – через підвищення витрат на проектування на 50%.

### 3.1.5. Стан ринку інженерних САПР

Цей ринок пропонує великий асортимент програмних засобів (ПЗ) для автоматизації інженерної діяльності у проектних організаціях та на виробництві. Він має неабиякий попит серед вітчизняних промислових підприємств (машинобудівних, верстатобудівних), які розглядають САПР як ефективний засіб виходу на внутрішній та зовнішній ринок із конкурентноспроможною продукцією світових стандартів.

Повна автоматизація виробництва пов’язана з великими капіталовкладеннями. Велика вартість багатофункціональних САПР високого рівня та обмежені можливості розробки складних виробів засобами 2D-САПР легкого рівня призвели до розквіту САПР середнього рівня, розвиток яких підтримують усі розробники САD/САМ-систем та до популярності серед користувачів САD/САМ/САЕ-систем пірамідальної схеми автоматизації конструкторсько-технологічної підготовки виробництва. Вона значно економить ресурси завдяки використанню на робочих місцях ПЗ виключно з необхідним набором функціональних можливостей:

* + 1. для концептуальної конструкторсько-технологічної проробки нового виробу – декілька робочих місць із САПР високого рівня;
    2. для детальної проробки виробу – декілька десятків чи сотень місць із САПР середнього рівня;
    3. для підготовки конструкторсько-технологічної документації та робочих креслень – до тисячі робочих місць із САПР легкого рівня.

3.1.6. Характерні особливості розробки інженерних САПР

Прийнято ділити об’єкти автоматизації (підприємства та організації) на підрозділи, в кожному з яких розв’язують блок однотипних задач, а отримані результати є вхідними даними для іншого підрозділу (конструкторський, технологічний відділи), це дає змогу використати накопичений роками досвід та специфіку конкретного підприємства в цілому та кожного відділу окремо.

Ефективним є застосування систем віртуального макетування, програмне забезпечення яких базується на технологіях віртуальної реальності.

Віртуальний макет-прототип – це інтегроване цифрове подання виробу та його властивостей, що формується за даними головної моделі, відбиває просторову взаємодію компонентів та дозволяє оцінити працездатність конструкції в цілому.

Переваги САПР, що підтримують можливості віртуального макетування: вони дозволяють замінити фізичний прототип виробу його віртуальним аналогом та в процесі комп’ютерного аналізу електронного зразка розв’язувати ті задачі, що раніше вимагали натурних випробувань.

Віртуальним макетуванням можна супроводити весь процес проектування виробів, що дозволяє проводити тестування паралельно з розробкою, тобто своєчасно знаходити та виправляти помилки та недоліки. Віртуальне макетування дозволяє використовувати майстер-моделі виробів у галузях маркетингу, продажу, навчання.

### 3.1.7. Класифікація САПР і їх користувачів

САПР залежно від їхнього функціонального призначення поділяються на:

* + 1. машинобудівні
    2. архітектурно-будівельні
    3. дизайнерсько-анімаційні
    4. універсальні

- до машинобудівних можна віднести такі прикладні пакети (ПП), як Mechanical Desktop, Solid Works, Atodesk Inventor, Техтран, КОМПАС;

- до архітектурно-будівельних – ArchiCАD, Autodesk Architectural Desktop R2, Allplan;

- до дизайнерсько-анімаційних – CorelDraw, Adobe Illustrator, 3D Studio;

- до універсальних (популярних продуктів без чіткого проблемного спрямування, які частково поєднують усі попередні) можна віднести AutoCAD, DenebaCAD, Actrix Technical та інші.

САПР поділяються на:

* + 1. унікальні (створюються один раз для проектування особливо важливих і складних об’єктів);
    2. універсальні (використовуються у рамках галузі),
    3. спеціалізовані (використовуються у рамках підприємства);
    4. індивідуальні (наприклад, AutoCAD).

Користувачі САПР поділяються на:

* + 1. користувачів-розробників (основна вимога – володіння інформаційними технологіями та знання галузі);
    2. користувачів-супровідників (мають знати методологію побудови САПР у загальних рисах, вміти працювати з підсистемами;
    3. користувачів-проектувальників (потрібні знання у предметній області та вміння підготувати вхідну інформацію).

## 3.2 Проектування спеціалізованих інтегральних схем на основі ПЛМ

### 3.2.1 Загальні відомості про систему MAX+plus II

Досягнення у області мікроелектронних технологій привели до того, що основу багатьох сучасних радіоелектронних і обчислювальних пристроїв складають спеціалізовані великі і надвеликі інтегральні схеми (ВІС і HВІС), що дозволяють значно поліпшити техніко-економічні характеристики апаратури конкретного призначення.

На практиці використовують п'ять способів реалізації спеціалізованих HВІС:

1) повністю замовлені – допускаючі повний цикл проектування, що включають розробку всіх літографічних шаблонів на рівні окремих областей транзисторних структур;

2) замовлені на основі бібліотечних елементів – що припускають використання наперед розроблених топологічних бібліотек елементів, вузлів і блоків різної складності і що включають розробку всіх літографічних шаблонів, але на рівні елементів, вузлів і блоків;

3) напівзамовні на основі базових матричних кристалів (БМК) – що припускають використання наперед виготовлених «напівфабрикатів» – БМК – кристалів з матрицями так званих базових осередків, кожна з яких містить набір елементів (транзисторів, діодів, резисторів і ін.), що дозволяють за допомогою розробки тільки замовлених шаблонів металізації з'єднати базові елементи відповідно до проектованої схеми для виконання заданого набору функцій;

4) на основі постійних пристроїв (ПЗП), що запам'ятовують, – програмування наперед виготовлених мікросхем ПЗП, що містять повний дешифратор і програмований шифратор (поле елементів пам'яті), для виконання певних функцій;

5) на основі програмованих логічних матриць (ПЛМ) – що також припускають програмування наперед виготовлених мікросхем, що містять ПЛМ, що відрізняються від ПЗП наявністю програмованих дешифраторів, що забезпечує додаткову гнучкість.



Рисунок 3.1 — Зовнішній вигляд ІС Altera

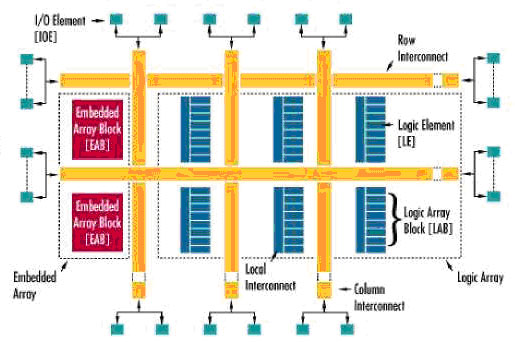


Рисунок 3.2 — Топологічна організація HBIC: IOE – елемент введення-виводу; ЕАВ – вбудований блок; LE – логічний елемент; LAB – блок ПЛМ; Row, Column, Local Interconnect – горизонтальні, вертикальні і локальні лінії зв'язку

Процедуру розробки нового проекту від концепції до завершення можна спрощено представити таким чином:

1. створення нового файлу або ієрархічної структури декількох файлів проекту за допомогою будь-якого поєднання редакторів в системі MAX+plus II, тобто графічного, текстового і сигнального редакторів;
2. завдання імені файлу – проекту верхнього рівня як ім'я проекту;
3. призначення сімейства ПЛІС для проекту;
4. відкриття вікна компілятора Compiler і вибір кнопки Start для початку компіляції проекту. За бажанням користувача можна підключити модуль витягання тимчасових параметрів проекту Timing SNF Extractor для створення файлу, використовуваного при тимчасовому моделюванні;
5. у разі успішної компіляції можливий часовий аналіз, для чого слід виконати наступне:

* для проведення аналізу затримок відкрити вікно Timing Analyzer, вибрати режим аналізу і натиснути кнопку Start;
* для проведення моделювання потрібно спочатку створити текстовий вектор у файлі (.sci), користуючись сигнальним редактором, або у файлі вектора (.vec), користуючись текстовим редактором. Потім відкрити вікно відладчика Simulator і натиснути кнопку Start;

1. відкриття вікна програматора Programmer з подальшим вибором одного з двох способів: використання програматора Master Programming Unit (MPU) або підключення завантажувальних пристроїв BitBlaster, ByteBlaster або FLEX Download Cable до пристрою, що програмується в системі;
2. вибір кнопки Program для програмування пристроїв з пам'яттю типу EPROM або EEPROM (електронно перепрограмованих ПЗП) або вибору кнопки Configure для конфігурації пристрою з пам'яттю типу SRAM (статичного ОЗУ).

Система MAX+plus II містить 11 застосувань і головну програму, що управляє. Різні застосування, що забезпечують створення проекту, можуть бути активізовані миттєво, що дозволяє користувачу перемикатися між ними натисканням миші або за допомогою команд меню. У цей же час може працювати одне з фонових застосувань, наприклад компілятор, програма моделювання, аналізатор синхронізації і програматор. Одні і ті ж команди різних застосувань працюють однаково, що полегшує завдання розробки логічного дизайну.

Файл проекту – це графічний, текстовий або сигнальний файл, створений за допомогою графічного або сигнального редакторів системи MAX+plus II або в будь-якому іншому, що використовує промислові стандарти.

Цей файл містить функціональний опис проекту MAX+plus II і обробляється компілятором.

Допоміжні файли – це файли, пов'язані з проектом MAX+plus II, але що не є частиною його ієрархічного дерева.

Більшість таких файлів не містять функціонального опису проекту. Деякі з них створюються автоматично застосуванням системи MAX+plus II, інші – користувачем. Прикладами допоміжних файлів є файли призначень і конфігурації (.acf), символьні файли (.sym), файли звіту (.rpt) і файли тестових векторів (.vec).

Проект складається зі всіх файлів ієрархічної структури проекту, зокрема допоміжних і вихідних файлів. Ім'ям проекту є ім'я файлу проекту верхнього рівня без розширення. Система MAX+plus II виконує компіляцію, тестування, аналіз синхронізації і програмування відразу цілого проекту, хоча користувач може в цей час редагувати файли цього проекту в рамках іншого проекту.

### 3.2.2 Редактори системи MAX+plus II

Система MAX+plus II включає чотири редактори створення проекту: графічний редактор (Graphic Editor), символьний редактор (Symbol Editor), текстовий редактор (Text Editor) і сигнальний редактор (Waveform Editor). Графічний редактор дозволяє створювати проекти на основі бібліотек УГО (рис. 3.3). Створені в даному редакторі проекти можуть містити будь-яку комбінацію УГО, мегафункції і макрофункції. Графічний редактор підтримує наступні формати файлів: .gdf і .sch.

Графічний редактор дозволяє збільшувати або зменшувати масштаб зображення проекту на екрані, вибирати розмір шрифту, задавати стилі ліній, одержувати дзеркальне відображення, повертати виділені фрагменти на 90, 180 і 270 градусів, задавати розмір і орієнтацію поточного листа схеми.

Символьний редактор дозволяє створювати і редагувати УГО (символ). Символьний файл має те ж ім'я, що і проект, з розширенням .sym.

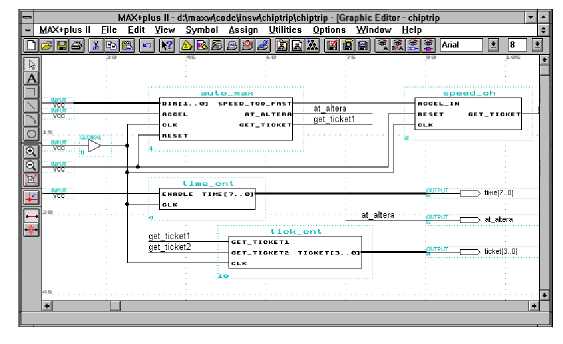


Рисунок 3.3 — Робоче вікно графічного редактора

Виклик редактора здійснюється по команді Create Default Symbol в меню File (рис. 3.4).

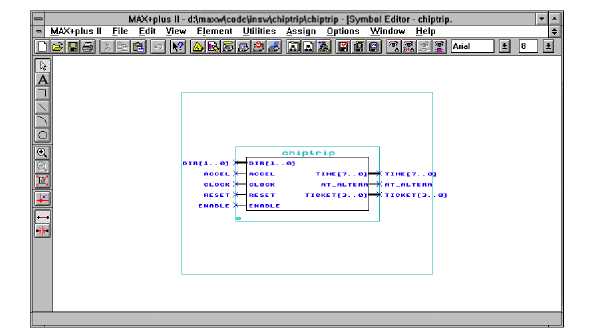


Рисунок 3.4 — Робоче вікно символьного редактора

Символьний редактор дозволяє перевизначати символ, що представляє файл проекту, створювати і редагувати виводи і їх імена, використовуючи вхідні, вихідні і двонаправлені виводи, а також задавати значення параметрів і встановлювати їх за умовчанням, вводити коментарі.

Текстовий редактор дозволяє створювати текстові файли проекту на мовах опису апаратури: AHDL (Altera Hardware Description Language) -.tdf, VHDL – .vhd, Verilog HDL. При цьому введення початкових даних про проект здійснюється не у вигляді схеми, виконаної в графічному редакторі (рис. 3.3), а у вигляді текстового опису проекту, що дозволяє наблизити розробку проекту до процесу програмування.

У текстовому редакторі також можна працювати з наступними форматами файлів: .acf, .асо .adf, .cmd, .edc, .edf, .fit, .hst, .lmf, .log, .mif,.mio, .mtf, .plf, .rpt, .sdo, .smf, .tao, .tdf, .tdo, .tdx, .ttf, .vec, .vho, .vmo, .vo, .xnf і з довільним файлом формату ASCII.

Даний текстовий редактор має наступні вбудовані функції: введення файлу проекту, їх компіляції і відладки з видачею повідомлення про помилки і їх локалізацією в початковому тексті або в тексті допоміжних файлів. Крім того, даний редактор містить шаблони мовних конструкцій для AHDL, VHDL і Verilog HDL. У текстовому редакторі можна редагувати файли конфігурації, а також робити установки для компілятора, програми моделювання і тимчасового аналізатора (рис. 3.5).

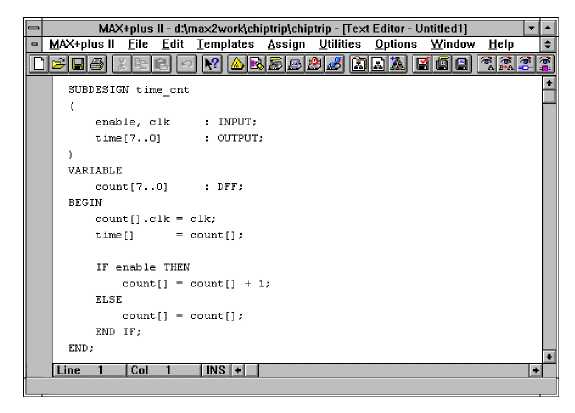


Рисунок 3.5 — Робоче вікно текстового редактора

Сигнальний редактор дозволяє створювати опис проекту, вводити тестові вектори і проглядати результати тестування (рис 3.6.). Даний редактор підтримує формат файлів сигнальних проектів .wdf, що містять тимчасові діаграми, а також формат файлів тестування .scf, що містять вхідні вектори для функціональної відладки.

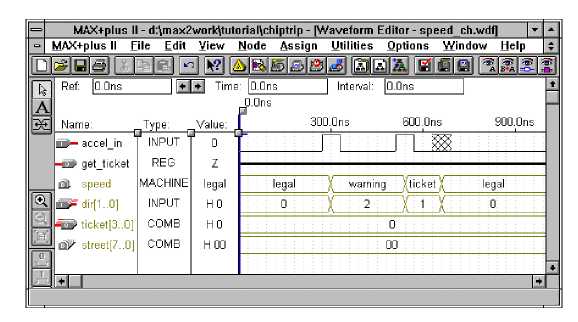


Рисунок 3.6. — Робоче вікно сигнального редактора

Сигнальний редактор є альтернативою графічного або текстового редакторів. Створений файл формату .wdf містить логічні входи, виходи комбінаторної логіки, лічильників і т.д.

Для відображення ієрархічної структури файлів проекту у вигляді дерева з гілками, що є підпроектами, система MAX+plus II містить дисплей ієрархії (Hierarchy Display).

У ієрархії поточного проекту відображається ім'я і ікона файлу для кожного підпроекту. Крім того, дисплей ієрархії показує допоміжні файли, пов'язані з поточною ієрархією (рис 3.7)

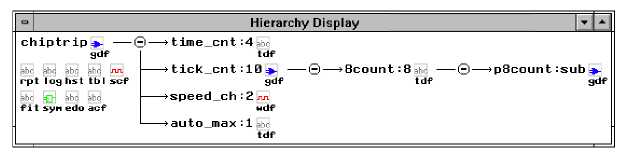


Рисунок 3.7. — Дисплей ієрархії

Для призначення ресурсів фізичних пристроїв і переглядання результатів розводки, виконаних компілятором, система MAX+plus II містить редактора базового плану кристала (рис 3.8).

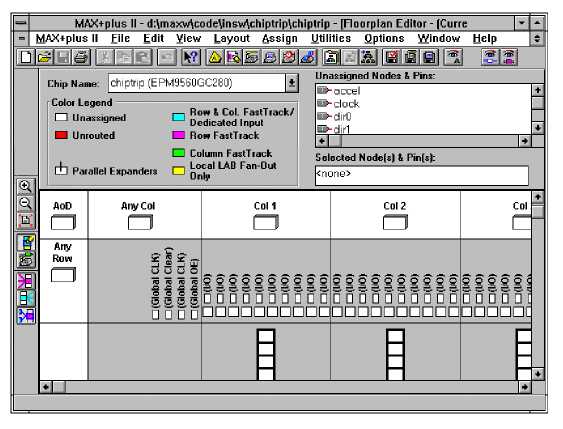


Рисунок 3.8 — Робоче вікно редактора базового плану кристала

Редактор базового плану кристала може представляти два типу зображення:

* 1. Device View, що показує всі контакти пристрою і їх функції;
  2. LAB View, що показує внутрішню частину пристрою, зокрема всі логічні структурні блоки і окремі логічні елементи.

Для компіляції створених проектів система MAX+plus II містить компілятор (Compiler) (рис. 3.9).

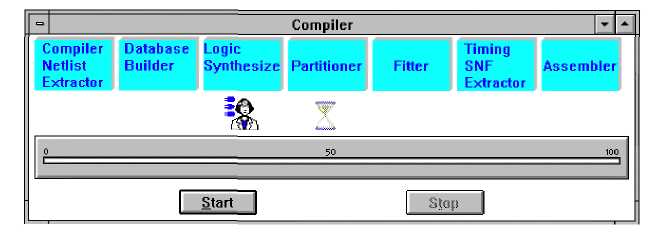


Рисунок 3.9. — Робоче вікно компілятора

Компілятор може автоматично обробляти наступні файли проекту: графічні файли проекту (.gdf); текстові файли проекту на мові AHDL (.tdf); сигнальні файли проекту (.wdf); файли проекту на мові VHDL (.vhd); файли проекту на мові Verilog (.v); схемні файли OrCAD (.sch); вхідні файли EDIF (.edf); файли формату Xilinx Netlist (.xnf); файли проекту Altera (.adf); файли цифрового автомата (.smf).

Компілятор системи MAX+plus II дозволяє створювати файли, використовувані програматором одного або декількох пристроїв для програмування ПЛІС.

Завантаження готового проекту в ПЛІС в системі MAX+plus II виконується за допомогою програматора (Programmer), який дозволяє програмувати, конфігурувати, проводити верифікацію і тестувати ПЛІС фірми Altera (рис 3.10).

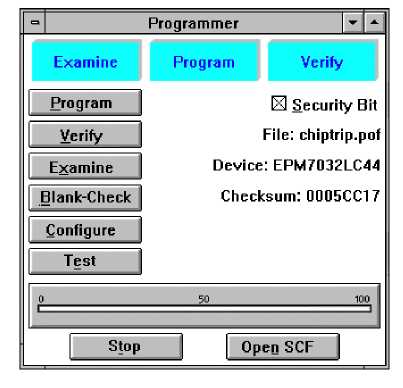


Рисунок 3.10 — Робоче вікно програматора

Для тестування логічних операцій і внутрішньої синхронізації проекту в системі MAX+plus II міститься програма моделювання (Simulator) (рис. 3.11). Вона дозволяє розробнику моделювати проект перш, ніж він буде реалізований в пристрої, що істотно скорочує час розробки проекту. Крім того, розробник може проводити моделирование проекту незалежно від кількості використовуваних пристроїв, потрібних для його реалізації.

Для аналізу виконання проекту в системі MAX+plus II використовується часовий аналізатор (Timing Analyzer) (рис. 3.12).

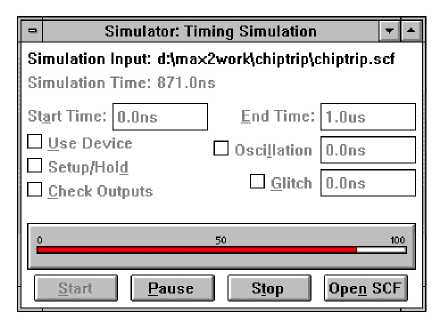


Рисунок 3.11. — Робоче вікно програми моделювання

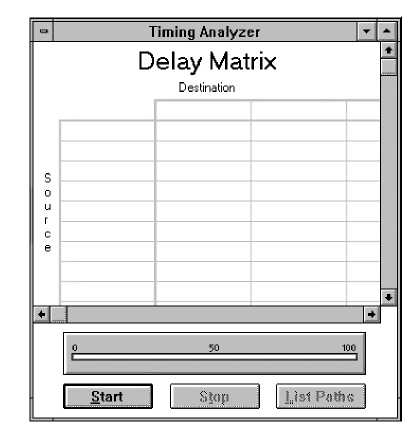


Рисунок 3.12. — Робоче вікно тимчасового аналізатора

Для видачі на екран повідомлень про помилки, застережливих і інформаційних повідомлень, в системі MAX+plus II використовується генератор повідомлень (*Message Processor*). Генератор повідомлень взаємодіє зі всіма застосуваннями системи MAX+plus II, що дозволяє визначати і коректувати помилки, одержувати застережливі повідомлення і підказки у будь-який момент часу (рис 3.13.).

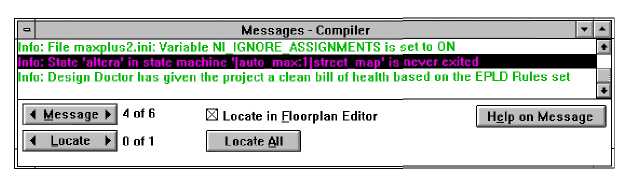


Рисунок 3.13 — Робоче вікно генератора повідомлень

# Висновки до розділу

В третьому розділі було розглянуто основи створення та появи САПР, їх види та вплив на суспільну індустрію з виготовлення на віртуалізації інженерних проектів. САПР є індустріальними технологіями, що націлені на найважливіші галузі виробництва, визначаючи рівень розвитку та стратегічний потенціал. Без САПР неможливо виробляти сучасну техніку, що є дуже складною та вимагає надзвичайної точності виготовлення.

# РОЗДІЛ 4. РЕАЛІЗАЦІЯ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ПОСЛІДОВНОСТЕЙ

Четвертий розділ відображає основну думку дисертації – практична реалізація генератора псевдовипадкових послідовностей. Для цього необхідно розглянути вибір технічного рішення, визначити основні етапи розробки, методику визначення принципіальних схем. За допомогою системи автоматизованого проектування MAX+PLUS II безпосередньо створити запланований пристрій.

## 4.1 Вибір технічного рішення та обгрунтування структурної та функціональної схем

Найважливішим на початку проектування є усвідомлення завдань та вибір технічного рішення. Вибір технічного зводиться до визначення послідовності та виду операцій, що виконуються з вхідними сигналами, та способів їх здійснення. Зазвичай існує декілька методів рішення поставленого завдання.

Стратегія проектування – функціональна декомпозиція, що включає зовнішній опис блоку (входи і виходи) і внутрішній опис - чи функцію алгоритм роботи: F = Ф(X,t), де X – вектор вхідних величин; *F* вектор вихідних величин; t – час. При декомпозиції функція Ф розбивається на більш прості функції Фl,...,Фk, між якими повинні бути встановлені визначені зв'язки, що відповідають прийнятому алгоритму реалізації функції Ф. У результаті розбивки, у кінцевому рахунку виходить структура. Перехід від функції до структури — синтез.

Вибір найкращого варіанта здійснюється за результатами аналізу, коли перевіряється правильність роботи і деяких показників, що характеризують пристрій. Декомпозиція функцій блоків виконується доти, поки не вийдуть типові функції, кожна з яких може бути реалізована тією чи іншою мікросхемою.

Процес проектування — багатокроковий і ітераційний, з поверненнями назад і переглядом раніше прийнятих рішень. До цього процесу варто додати можливості сучасної елементної бази, що декомпозиція закінчується при одержанні типових функцій, шо відповідають тим чи іншим мікросхемам чи елементам функціональних бібліотек, ВІС/НВІС, що програмуються.

Характер проектування істотно залежить від виду застосовуваної елементної бази. Основною часткою інженерних розробок апаратури в умовах сучасної України, очевидно, є використання програмованої логіки для створення необхідних пристроїв чи їхнього налагодження. Проектування на основі схем програмованої логіки високої складності виконується тільки за допомогою систем автоматизованого проектування. Спрощена структура алгоритмів проектування показана на рисунку (рис. 4.1)

У своїй кваліфікаційній роботі я використовуватиму НВІС фірми ALTERA і відповідно систему автоматизованого проектування MAX+PLUS II.

Проектування на концептуальному рівні покладається на проектувальника і слабко зв'язана з автоматизацією. На цьому рівні визначається необхідне функціонування пристрою, вхідні і вихідні сигнали. їх характер і взаємозв'язок, розбивка проекту на частини і т.д. Результати концептуального синтезу вводяться в САПР, що робить компіляцію проекту, тобто синтезує пристрій у базисі бібліотеки своїх моделей. Отриманий проект вимагає ретельної перевірки, тому за етапом синтезу випливає етап аналізу, проведеного моделюванням і теоретичною верифікацією. Моделювання має кілька рівнів з різним ступенем відображення властивостей реального об'єкта. Воно може бути функціональним, що перевіряє правильність логічної структури пристрою, часовим, враховуючи затримки сигналів у схемах пристрою без остаточної топології трасування, і т.д. У результаті моделювання можуть виявитися помилки, що вимагають виправлення, що додає процесу проектування ітеративний характер з поверненнями до колишніх етапів і введенням у проект потрібних корекцій.

Концептуальний синтез

Установка в систему

Фізичне моделювання

Конфігурування

Часове моделювання

Функціональне моделювання

Компіляція

Ввід в САПР

Рисунок 4.1 — Структура алгоритму автоматизованого проектування.

Далі проводиться конфігурування мікросхеми програмованої логіки, після чого можлива реальна перевірка роботи пристрою - фізичне моделювання проекту. При успішному завершенні фізичного моделювання пристрій готовий до установки в систему.

### 4.2 Загальна методика проектування цифрових пристроїв

Усі сучасні методики проектування на базі складних програмованих ВІС/НВІС основуються на застосуванні САПР. Правильний вибір САПР – найважливіша умова ефективного проектування і прискорення випуску продукції. Розглянемо кожен крок проектування цифрових пристроїв більш детально.

Засоби опису проекту

В даний час до найбільш розповсюджених універсальних способів опису, що застосовуються для будь-якого рівня ієрархії проекту, відносять графічний і текстовий. Рідше використовуються безпосереднє розведення схем FPGA у редакторі топології, опису у виді необхідних часових діаграм. Кожний зі способів опису проекту мас свої переваги і недоліки. Подібність способу опису, внутрішньої організації і поводження розроблювального пристрою істотно скорочує час створення проекту, спрощує його тестування і, як правило, виявляється найбільш наочним і зрозумілим.

Графічне представлення проекту створюється в базисі припустимих для обраної САПР бібліотечних елементів, наприклад, у базисі елементів стандартної серії ТТЛ (Ш). Головна перевага графічного способу – його традиційність і наочність, зв’язані зі звичністю проектувальників до сприйняття зображень схем. Звичайно, ця перевага виявляється тільки при правильній ієрархічній і структурній розбивці проекту.

Сучасні мови опису апаратури допускають опис проектованого пристрою, як з погляду його поводження, так і з погляду його структури. Переваги текстового способу опису проекту полягають у його компактності і відносній простоті автоматизації будь-яких перетворень, включаючи початкову генерацію опису проекту. Дуже важлива можливість використання стандартних універсальних мов типу HDL, що забезпечує простоту переносу проекту з однієї апаратної платформи па іншу і перехід від однієї САПР до іншої.

На відміну від текстових, графічні способи представлення проекту звичайно вузько спеціалізовані і вимагають особливих засобів для переносу інформації про проект в інше середовище, для. Можуть бути застосовані спеціальні універсальні мови передачі інформації про проект, які одержують все більшого поширення. Текстові описи мають два основні різновиди – мови низького і високого рівня.

Мови низького рівня ближче до апаратних засобів, унаслідок чого представляють для компіляторів потенційні можливості створення проектів з більш виграшними параметрами. Платою за це є звичайно тверда орієнтація на визначену апаратуру і фірму, яка її виробляє. Прикладами таких мов можуть служити мова AHDL (Altera HDL) і ABEL (фірми Хіlіnx). За допомогою мов низького рівня легше створювати проекти з найкращими часовими параметрами, тому що в проектах будуть враховані специфічні особливості архітектури тієї чи інший CPLD чи FPGA.

При розробці ЦП найбільш ефективною є його розбивка на два блоки: операційний і керування. Операційний блок (ОБ) виконує перетворення даних будується зі стандартних частин, а блок керування (БК) забезпечує необхідну послідовність операцій, виконуваних у ОБ (одному чи декількох). Для цього БК передає на входи ОБ керуючі сигнали. Послідовність дій і керуючих сигналів залежить від результатів операцій у ОБ. Звідки слідує, що БК зручно задавати у формі кінцевого автомату з пам’яттю (АП) того чи іншого типу, у складних проектах можливий поділ ЦП на декілька функціоналів слабо зв’язаних пар ОБ-БК на одному рівні ієрархії, чи створення пари ієрархічно зв’язаної системи.

Операційний блок звичайно представляється набором регістрів логічних схем (як правило, багатофункціональних і керованих), буферних схем і зв’язків, що комутуються, між ними.

Етапи проектних процедур

Порядок розробки ЦП в загальному вигляді зображений вище (див. рис. 4.1). Більш детально проектування з використанням САПР розглянемо нижче. Розробка виконується в наступному порядку:

Специфікація проекту

Складання змістовної граф-схеми чи алгоритму функціональної блок-схеми пристрою. Перша задача — перехід від технічного завдання (ТЗ) до формалізованого опису проектованого пристрою. Технічне завдання, як правило, є сумішшю словесного і технічного опису, його формалізація приводить до виявлення основних блоків пристрою (чи алгоритму) і визначенню їхніх зв'язків і взаємодії Сама форма ТЗ може визначати проектувальнику ті чи інші засоби, хоча не виключено, що більш ефективним міг би бути інший метод опису проекту чи його фрагментів. У сутності саме в цей реалізуються початкові дії першого етапу. Формально ж перший етап — розбивка задачі на окремі функціонально відособлені підзадачі — етап декомпозиції.

Декомпозиція може зводитися до складання схем алгоритмів функціонування фрагментів або до функціональної схеми пристрою і його частин. Можливим варіантом для досить складних систем буде розумне сполучення поведінкової і структурної розбивки проекту. Розбивка здійснюється не тільки в рамках одного рівня ієрархії. Для більшості проектів виробляється і розбивка на ієрархічно організовані рівні. Уточнюється реалізація протоколів зовнішнього обміну. Часові характеристики і правила взаємодії з зовнішніми приладами визначають припустиму організацію і структуру внутрішніх вузлів проекту.

Використання САПР на цьому етапі проектування досить рідке явище

Розробка загальної структури проекту

Основна задача — вибір припустимих для реалізації кожного рівня ієрархії елементів, визначення зв'язків між ними, і якщо параметри елементів налаштовуються — то це їх налаштування. Ряд моментів для етапу є визначальними: з одного боку, це джерело набору припустимих елементів, а з іншого боку засіб опису з'єднань елементів між собою, а при необхідності, і можливість опису нових (специфічних для цього проекту) елементів.

При розробці пристроїв з цифровим представленням інформації буває природним розбивка їх на два блоки: операційний і керування. Операційний блок (ОБ) виконує перетворення даних і будується зі стандартних частин (частин з визначеним поводженням), а блок керування (пристрій керування, БК) забезпечує необхідну послідовність операцій, виконуваних у ОБ (одному або декількох). Для цього БК передає на входи ОБ керуючі сигнали. Послідовність дій і, або керуючих сигналів залежить від результатів операцій у ОБ і зовнішніх впливів. Звідси видно, що БК зручно подавати у формі кінцевого автомата з пам’яттю (AП) того або іншого типу.

При розробці функціональної схеми необхідно добиватися:

1. забезпечення заданих значень параметрів сигналів, швидкодії пристрою, потрібної стабільності та похибки виконання операцій;
2. зменшення числа елементів для підвищення надійності роботи, зниження вартості і споживаної потужності;
3. побудови на базі мінімального числа різнотипних пристроїв
4. зручність експлуатації (зменшення числа регулювань перемикачів і т.д.);
5. використання мінімального числа джерел живлення;
6. простота міжкаскадного зв’язку;
7. виключення взаємного впливу елементів пристрою, вплив навантаження і ємності з’єднувальних кабелів.

Розробка загальної структури операційного блоку

Архітектурно-структурний опис операцінйого блоку базується на завданні структури з’єднань окремих елементів. Традиційний графічний спосіб представлення структури з’єднань, залишаючись найбільш наочним способом у сучасній САПР, супроводжується текстовим описом.

Будь-який проект може бути використаний у якості підпроекту В більш складному проекті. Як правило, на будь-якому рівні ієрархії базовий набір елементів операційного блоку (цього рівня) доповнюється необхідним для його функціонування набором регістрів, логічних схем (багатофункціональних і керованих), буферних схем і зв’язків, що комутуються, між ними. Важливо, щоб на більш низьких ієрархічних рівнях опису проекту було однозначне трактування функціонування всіх елементів ОБ.

Незважаючи на оптимізацію схем, що відповідають найбільш розповсюдженим серіям ІС, неправильним представляється пряме перенесення схемотехнічних рішень з СІС чи МІС у реалізацію па ВІС ПЛ. Одержувані при цьому результати, як правило, не тільки неоптимальні, але найчастіше не забезпечують бажаного функціонування проекту.

У тих випадках, коли поводження того або іншого елемента операційного блоку не відповідає поводженню наявних у розпорядженні проектувальника стандартних елементів, розроблювач змушений створювати свій елемент. Так само, як і при описі структури пристрою, опис поводження окремих елементів може здійснюватися з залученням як графічних, так і текстових засобів.

Опис роботи керуючого блоку

На цьому етапі визначається функціонування КБ, що забезпечує необхідну взаємодію елементів ОБ. Варто підкреслити, що два останніх етапи сильно взаємозалежні, і, якщо не розробляються паралельно, то виконуються шляхом ітерацій.

Форми і засоби опису автомата різноманітні. Сучасна тенденція складається в переході від запису логічних виразів, обмежених правилами ТЗ, до графічної форми. Опис у виді граф-схеми переходів (діаграми станів) стає одним з найпоширеніших варіантів завдання автоматів. Графічні редактори для створення автоматів включаються до складу засобів завдання вихідних проектів сучасних САПР.

Компіляція проекту

Після складання проекту і всіх йото частин можна приступати до найбільш відповідального етапу проектування – компіляція проекту. Компіляції може застосовуватися як до всього проекту так і частин проекту. Компіляція окремих фрагментів з однієї сторони спрощує проектування, оскільки зменшує розмір проблеми, що аналізується, а з іншої сторони вимагає враховувати різноманітні способи функціонування внутрішніх ресурсів і зовнішніх елементів.

Результат компіляції при використанні CAПP фірм-виробників ВІС ПЛ —завантажувальний файл, тобто конфігураційна інформація для вибраної мікросхеми, що програмується. Крім цього створюється файл звіту, який містить всю інформацію про процес компіляції та її результати. Існує суттєва різниця для компіляційних процедур схем типу CPLD і FPGA. Для FPGA крім автоматичного розміщення і трасування з'єднань, як правило, допускаються корегування проектувальника в процес на всіх етапах. Для цього використовуються редактори топології, що дозволяють змінювати структуру проекту на кристалі і збільшувати продуктивність пристроїв, що проектуються. А для CPLD вплив проектувальника на структуру скомпільованої схеми, можливий тільки шляхом непрямого втручання, за рахунок зміни опису проекту чи зміни опцій, що встановлюються перед компіляцією. Після вдалої компіляції проекту чи його частин можливий перехід до його верифікації.

Верифікація проекту

В сучасних САПР найбільш розповсюджене тестування шляхом роботи з редактором часових діаграм, ці редактори діляться на компілюючі і інтерпретуючі. Редактори інтерпретуючого типу дозволяють спростити процедуру відладки проектів і знайти їх дефекти, зв’язані з неправильною трактовою розробником структурної чи поведінкової реалізації системи, чи особливостей реалізації елементної бази, що використовується. В багато-віконних САПР інтерпретуючого типу просто відображуються результати моделювання для даного моменту часу у всіх видах відображення проекту (сигнали, електрична схема, топологія), тому легко змінити хід експерименту і склад відображених сигналів. У випадках схемотехнічного опису проекту спрощене трасування сигналів.

В більшості реальних цифрових пристроїв після подачі на них деяких початкових даних виконується декілька циклів. Необхідна перевірка роботи пристрою на декількох наборів однотипних даних, тому можна використовувати наступну структуру програмного модуля, що представляє тестовий вплив: генерація сигналів початкового заповнення, потім реалізація двох вкладених циклів, причому внутрішній цикл послідовно формує тестуючі сигнали для виконання дій на одному наборі вхідних даних, у зовнішньому циклі проводиться їх зміна.

Визначення часових характеристик розробленого пристрою

Сучасні САПР мають всередині себе повну інформацію про структуру пристрою, що проектується і часових параметрів всіх компонентів - це дозволяє автоматизувати процес обчислення різноманітних часових характеристик проекту.

Організація натурних експериментів

Виконання натурних експериментів істотно збільшу ймовірність випуску бездефектної продукції. Засобу прискорення робіт на цьому етапі і можливості його переносу на ранні етапи розробки, тобто до того моменту, коли буде закінчене виготовлення кінцевого продукту, відомі — це прототипні системи і засоби проведення експериментів з ними. Прототипні плати широко використовувалися і раніше, зокрема, при створенні мікропроцесорних систем. Аналогічна і ситуація при розробці систем і пристроїв на основі засобів програмованої логіки.

Як при використанні прототипних виробів, так і при тестуванні кінцевого виробу необхідне залучення апаратних і програмних засобів для завантаження конфігурації, генерації впливів і контролю правильності поводження досліджуваного пристрою. Важливий момент проведення експериментів — генерація тестуючих впливів. У цьому плані ВІС ПЛ забезпечують нові, не досяжні раніше можливості. Додаткова мікросхема, а в деяких випадках і визначена частина ресурсів ВІС, що настроюється, може використовуватися як генератор сигналів. При цьому зміст тестів легко модифікується в процесі проведення експериментів у залежності від результатів попередніх кроків експериментальних робіт. Корисним інструментом налагодження можуть стати засоби передачі даних про стан тестованого об'єкта в процесі виконання експерименту інструментальний комп'ютер з метою візуалізації і детального аналізу. Багато ПЛІС і відповідні САПР підтримують таку взаємодію. Але в багатьох випадках може знадобитися розробка спеціального програмного забезпечення для прискорення аналізу проводження об'єкта тестування. Природно, у визначених ситуаціях допускається користування серійною апаратурою типу багатопроменевих осцилографів, логічних аналізаторів і т.д.

Підготовка до виробничого випуску

Після успішного завершення натурних експериментів із прототипним або макетним зразком проектувальник повинен забезпечити випуск дослідної партії розробленого виробу. Найважливішою задачею при цьому є забезпечення якісного супроводу продукції, що випускається у формі ІС ПЛ.

Оскільки принципова працездатність проекту вже не піддається сумнівам, то задача тестового устаткування, що застосовується на цьому етапі, відрізняється від задач устаткування, що використовувалось на попередньому етапі. Тестове устаткування і методика його застосування повинні відсівати несправну продукцію за мінімальний час і з мінімальними витратами та виконуватися персоналом при мінімальних професійних вимогах. І лише для невеликої партії продукції можливо використовувати тестові засоби, що дозволяють локалізувати місця несправностей.

Необхідно розділяти вимоги до тестового устаткування і вимоги до самого проекту. Ефективність роботи тестового устаткування в значній мері залежить від передбачливості проектувальника. Процедура розробки з перших кроків повинна орієнтуватися на необхідність тестування кінцевої продукції. Якщо проектувальником (або замовником проекту) була правильно обрано ціль проектування, то після випуску дослідних зразків виробу, може виникнути задача організації випуску серійної продукції. Основні зусилля розробників при цьому спрямовані на проектування і виготовлення апаратних і програмних засобів, що дозволяють здешевіти і прискорити випуск серійної продукції. За умови появи ознак підвищеного попиту на продукцію, що випускається, і реальних ринків збуту може розглядатися питання про модернізацію продукції, в тому числі про перехід на більш дешеву елементну базу ІС. Це напрямок зв’язаний з розробкою стратегії і тактики конвертації проектів на нову елементну базу або переходу до нових технологій виготовлення кінцевої продукції.

Методика обгрунтування принципіальних схем

Обгрунтування принципіальних схем пристрою, що проектуються укладається у виборі елементів, міжкаскадних зв'язків, способів регулювання і перемикання режимів роботи. Обгрунтування принципіальних схем пристрою створюється в такому порядку:

1. виходячи з призначення і виконання операції, формуються основні вимоги до кожного елементу пристрою, що визначає вибір типу його схеми;
2. шляхом порівняльної оцінки вибирають тип схеми елементу пристрою;
3. вибирається і обґрунтовується схема зв’язку між елементами;
4. обгрунтовується необхідність регулювання параметрів, що виробляють сигнали та вибирається схема регулювання.

Пристрій, схема якого вибрана, в порівнянні з іншими варіантами повинна:

1. забезпечити виконання операцій з заданою похибкою, швидкодією та стабільністю;
2. містити мінімальне число елементів та різнотипних ІМС;
3. споживати меншу енергію;
4. бути зручним в експлуатації і містити найменше число органів оперативного регулювання.

При складанні та обгрунтуванні принципіальних схем особливу увагу необхідно приділити вибору способу зв’язку і потенціальному узгодженню елементів і ІМС пристрою. Від способу зв’язку залежить найважливіші якісні показники пристрою; перехідна характеристика, що визначає спотворення форми сигналів, що передаються, стабільність режиму роботи, величина споживчої потужності, завадостійкість і навантажувальна здатність.

Вибір способів регулювання параметрів створюється шляхом порівняльної оцінки можливих варіантів. Не допускається використання таких способів, при яких виникають взаємні впливи регулювань на режим роботи або параметри сигналів.

Розрахунок пристрою укладається в обґрунтуванні типів використаних приладів та ІМС, визначенні режимів роботи, параметрів сигналів і типу номіналів сигналів (конденсаторів, резисторів і т.д.).

Розрахунки можуть носити як проектуючий, так і перевірочним характер, тому що при виконанні роботи можна використовувати вже випробувану схему, перевіривши розрахунком відповідність генеруючих нею сигналів технічними умовами.

При проектуванні пристрою на базі ІМС проводяться розрахунки:

1. потенціального погодження вхідних та вихідних кіл;
2. параметри введених навісних елементів та узгоджених пристрою на дискретних компонентах;
3. узгодження кіл по навантажувальній здатності.

## 4.3.Проектування генератора псевдовипадових послідовностей

### 4.3.1.Загальні відомості

При створенні стійких криптографічних алгоритмів необхідно гарантувати певну величину лінійної складності для псевдовипадкових послідовностей. Незважаючи на достатньо великий період і хороші статистичні якості, вихідні гами ЛРР мають малу лінійну складність. Тому для їх застосування в криптографічних перетвореннях використовують різні способи їх нелінійного ускладнення:

* 1. фільтруючі генератори;
  2. комбінуючий генератор;
  3. композиції ЛРР;
  4. схеми з динамічною зміною закону рекурсії.

Перші три розглянуті способи ускладнення лінійних рекурент об’єднує загальна ідея підвищення лінійної складності вихідних послідовностей за рахунок застосування додаткових функцій ускладнення.

Альтернативний спосіб ускладнення ЛРП складається зі зміни закону рекурсії в процесі роботи криптографічного алгоритму. Привабливим являється використання нелінійної логіки в ланцюзі зворотного зв’язку регістрових перетворень. Але загальна теорія подібних схем ще не достатньо розроблена, в зв'язку з чим тяжко гарантувати необхідні властивості відповідних послідовностей.

### 4.3.2.Структурна схема пристрою

В даному прикладі побудовано пристрій формування рекурентної послідовності з адаптивною функцією зворотною зв’язку, що має наступні характеристики:

1. розрядність основного робочого регістру - 47 комірок;
2. розрядність управляючого регістру - 5 комірок,
3. зворотні зв’язки робочого регістру описані наступними багаточленами:

F1(x)=x47+x5+x0,

F2(x)=x47+x14+x0,

F3(x)=x47+x20+x0.

F4(x)=x47+x21+x0,

1. Зворотній зв’язок керуючого регістру

G(x)=x5+x2+x0

1. Точки зняття інформації для вузла вибірки з управляючого регістру управління

На структурній схемі (рис. 4.2) видно, що запис початкового заповнення в ЛРР (керуючий та основний) здійснюється через віповідні входи пристрою.:

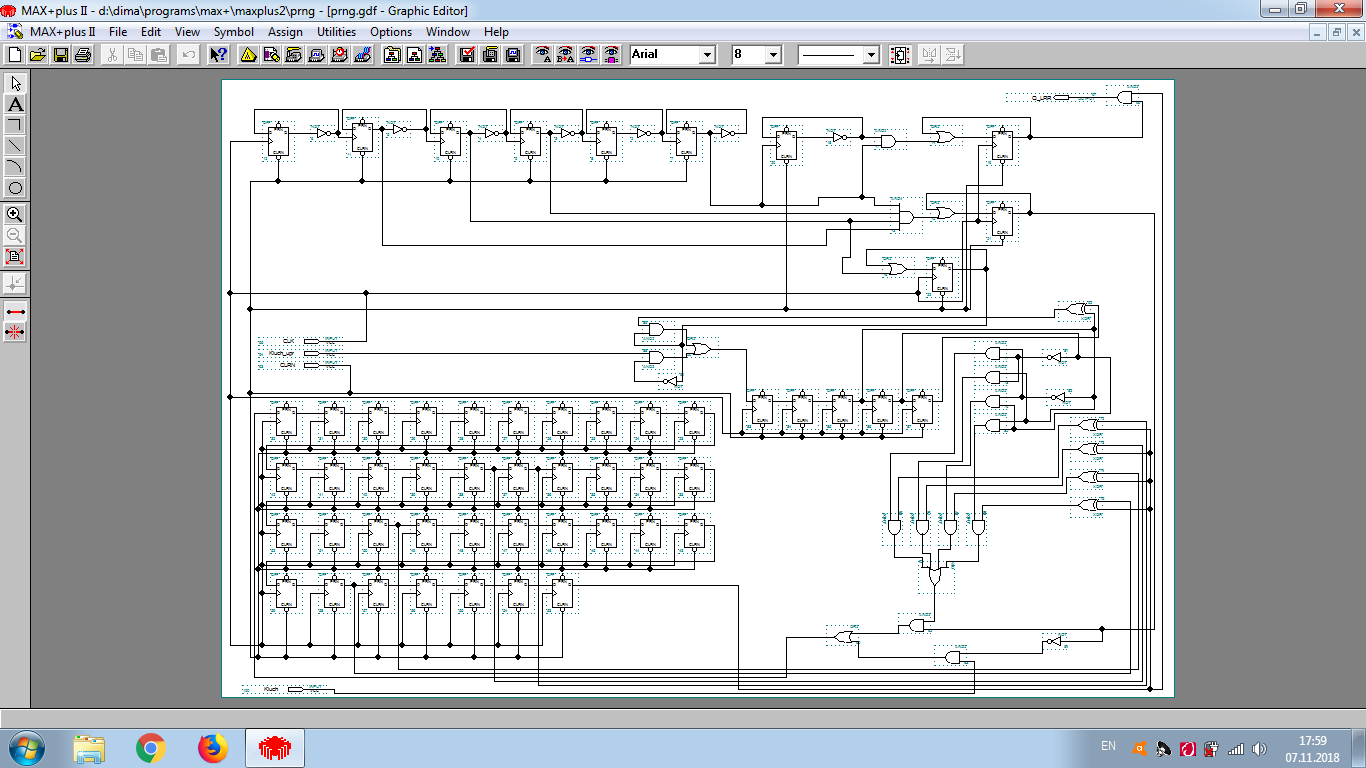


Рисунок 4.2 — Структурна схема пристрою

Блок керування режимом роботи формує 3 сигнали (5, 47, та 97 такти), які служать для управління процесом закінчення запису початкової інформації в регістри, а також для того, щоб на виході пристрою вперших тактах роботи не було початкового заповнення основного регістру. В блоці вибірки зворотнього зв'язку обирається який примітивний багаточлен (сума по mod2 з нульової комірки (N0) та однієї з перерахованих вище (N5, N14, N20, N21)), буде приймати участь в формуванні наступного значення 46 комірки. Отримана послідовність подається на вихід ЛРР і формується до появи на вході установки тригерів в нульове положення (рис. 4.2).

### 4.3.3.Принципова схема пристрою та епюри функціонування

Загальна принципова схема пристрою формування рекурентної послідовності з адаптивною функцією зворотнього зв'язку побудовано на основі D-тригерів та елементів дискретної логіки побудована в САПР MAX+PLUS II зображена на рисунку (рис. 4.3).

Рисунок 4.3 — Принципова схема пристрою

На схемі показано наступні входи та вихід:

Q\_LRR – вихідна послідовність з генераторами ПВП;

Kluch – початкове заповнення 47-ми розрядного основного регістру;

Kluch\_upr – початкове заповнення 5-ти розрядного керуючого регістру;

CLK – тактова частота;

CLRN – перезапуск пристрою;

Часові діаграми пристрою та окремі його блоки будуть показані при більш детальному аналізі пристрою.

На входи clk та clrn подаються глобальні тактова частота CLK та обнуління пристрою (сигнал CLRN). Сигнал з виходу основного ЛРР подається на логічний елемент “І”, на другий вхід якого поступає сигнал від блоку управління, за допомогою якого на виході в перших 97 тактах буде логічний нуль. Група логічних елементів XOR утворює зворотні зв’язки з основного ЛРР, які потім подаються на логічні елементи “І”, де відбувається вибір зворотного зв’язку який подається на вхід основного ЛРР.

Як видно зі схеми, вона являється досить громіздкою тому функціонування та принцип роботи розглянемо по окремих блоках.

#### 4.3.3.1 Дільник частоти

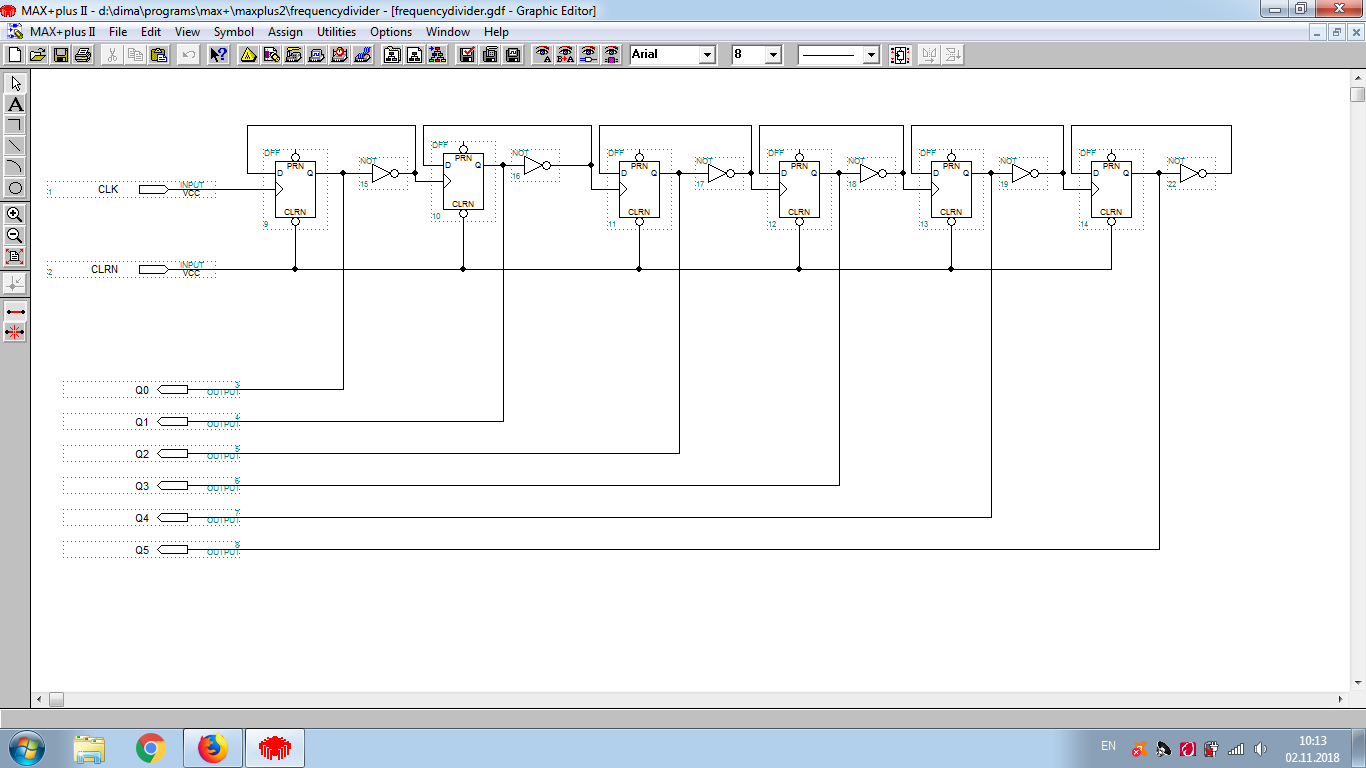
Даний пристрій складається з 6 послідовно з’єднаних D -тригерів (рис. 4.4).

Рисунок 4.4 — Принципова схема дільника частоти

На входи clrn всіх тригерів подається глобальний сигнал CLRN, який служить для запуску роботи дільника частоти (сигнал логічної одиниці). На вхід clk першого тригера подається глобальна тактова частота. На входи clk решти тригерів подається інверсне значення з виходу попереднього тригера. На всі входи D подається інверсне значення зі свого виходу. З кожного тригера для подальшого управління пристроєм існує вихід. Часові діаграми його роботи зображені на рисунку (рис. 4.5).

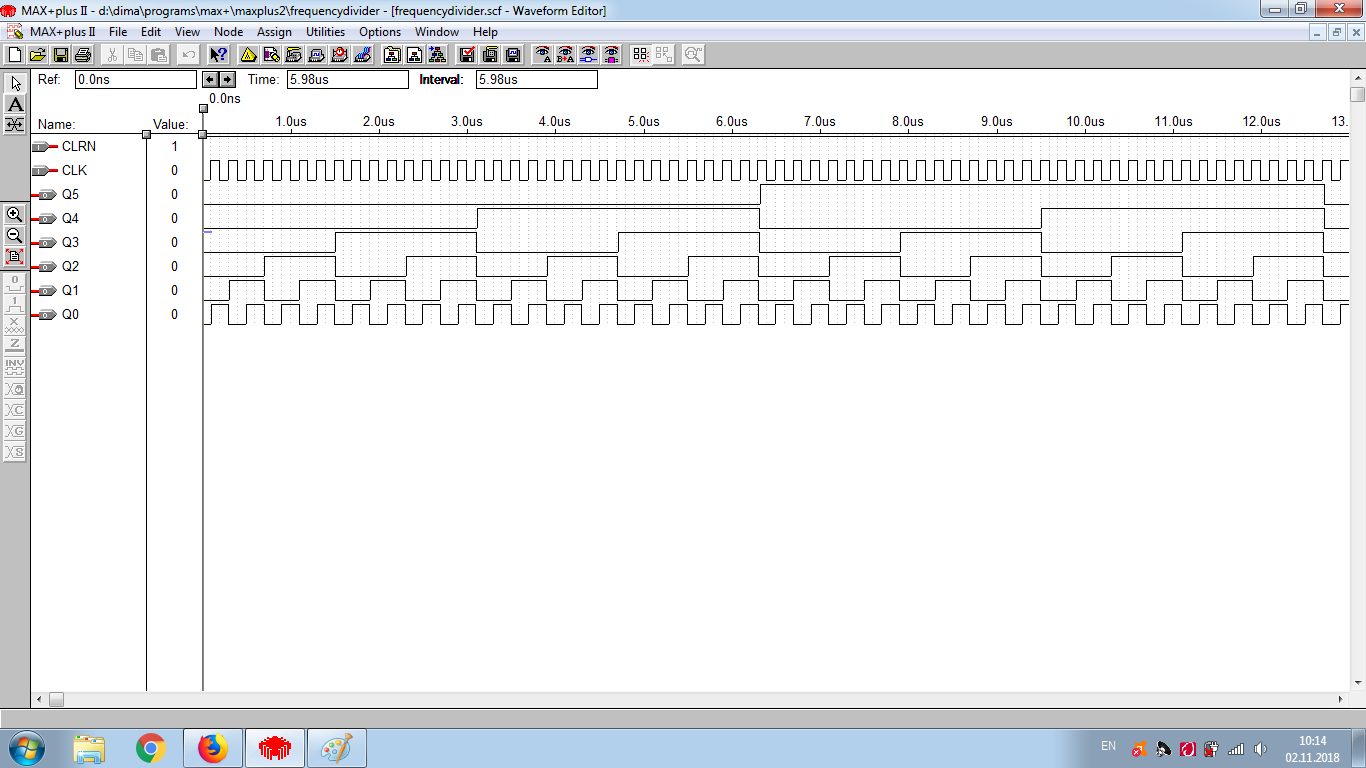
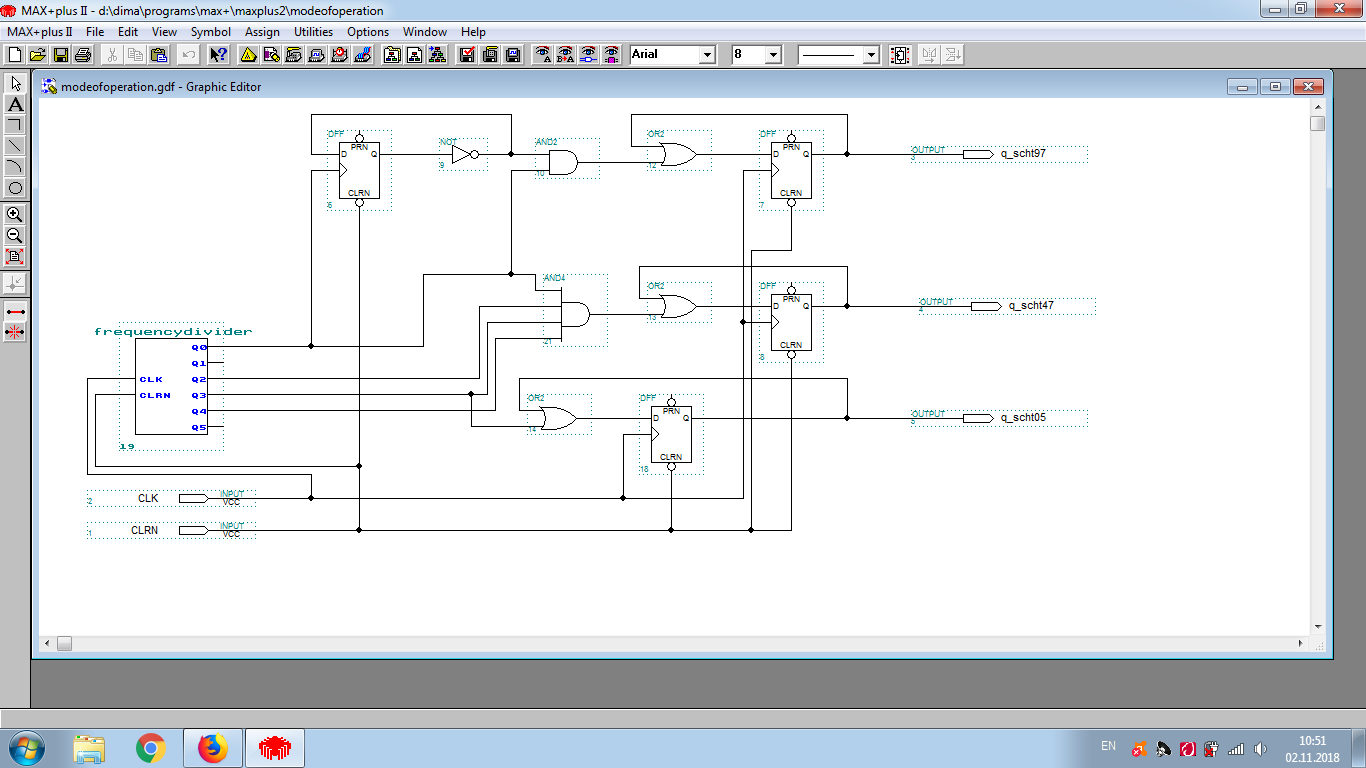


Рисунок 4.5 — Епюри функціонування дільника частоти

#### 4.3.3.2 Блок керування режимом роботи

Даний блок служить для подачі сигналу, який сигналізує про закінчення на певному такті роботи запису в регістр, а також для установки на виході в перших тактах логічного нуля. Певні виходи зі схеми подаються на елемент логічного множення і потім на схему установки сигналу в логічну одиницю. Принцип якої є те, що при наявності на вході першої логічної одиниці на виході буде установлена логічна одиниця доти, доки не відбудеться обнуління тригерів. Даний блок має 2 входи (clk, crln) та 3 виходи (q\_scht05, q\_scht47 тa q\_scht97) (рис 4.6). Часові діаграми даного блоку зображені на рисунку (рис. 4.7).

 Рисунок 4.6 — Принципова схема роботи блоку керування режимом роботи

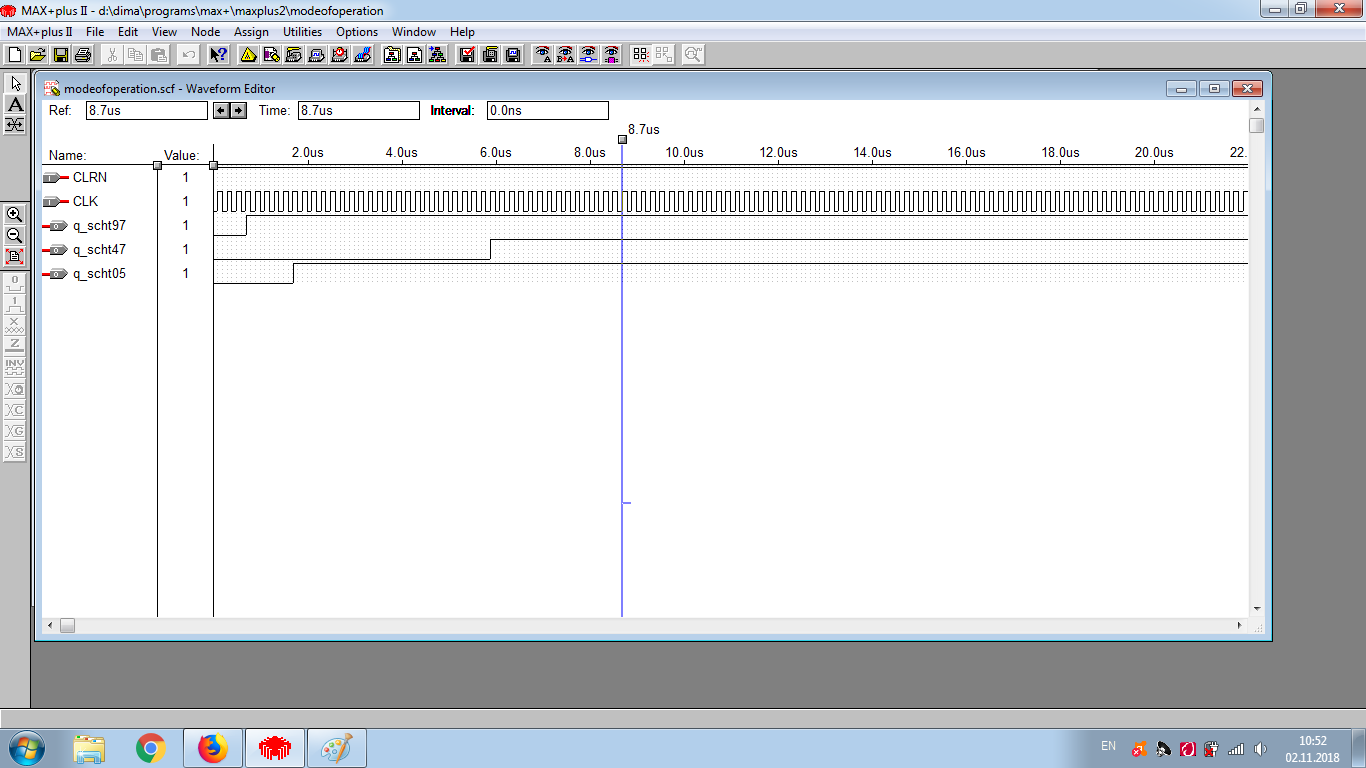


Рисунок 4.7 — Епюри функціонування блоку керування режимом роботи

#### 4.3.3.3 Блок управління функцією зворотного зв’язку

Даний блок має в своєму складі безпосередньо управляючий регістр, а також вузол вибірки сигналу з тригерів та сигнали від блоку керування роботи (рис. 4.8).

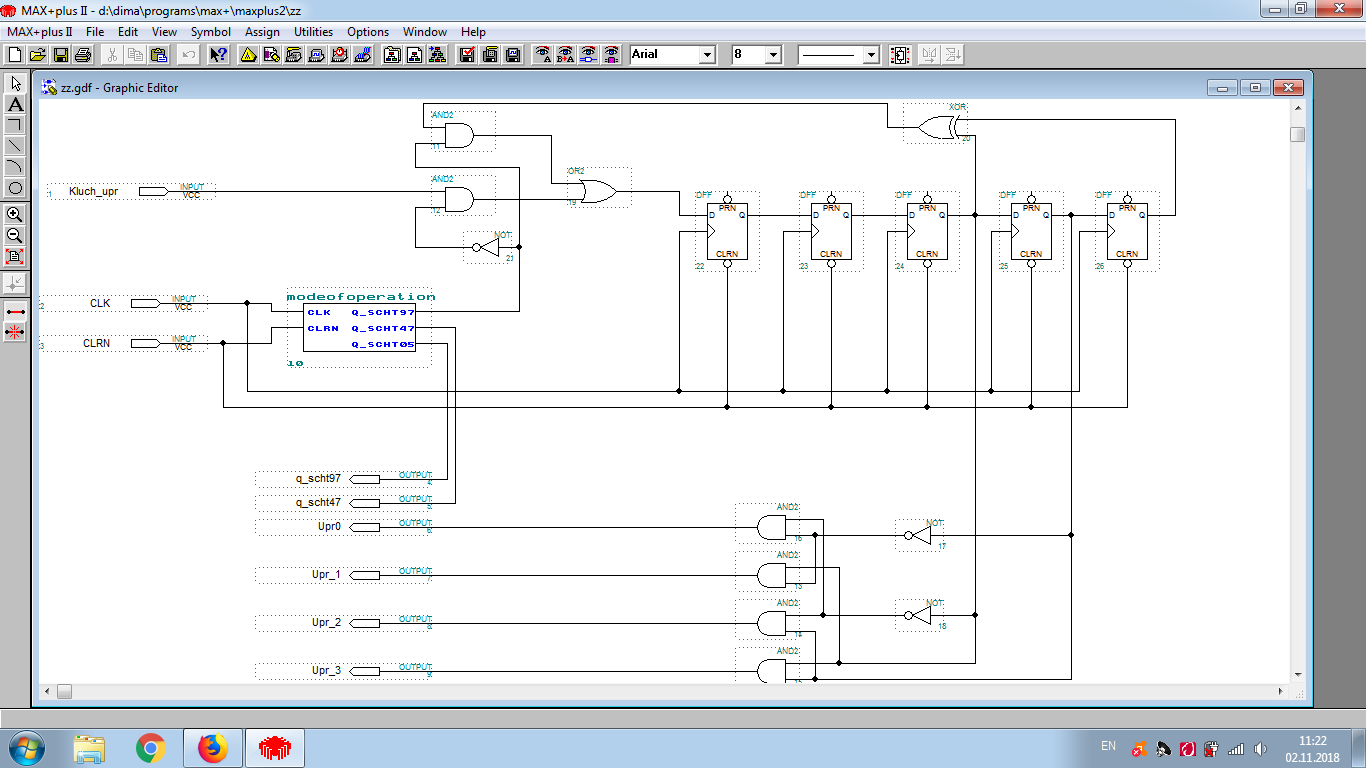
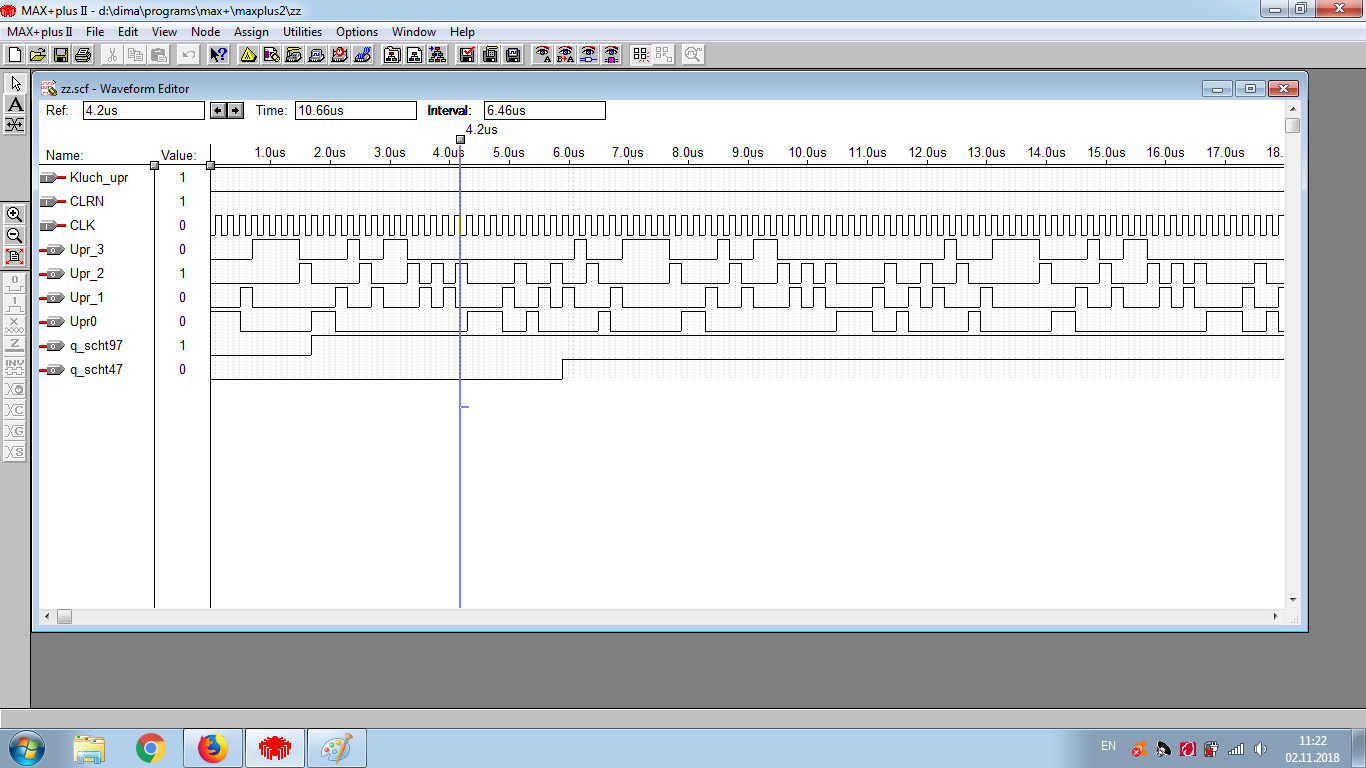


Рисунок 4.8 — Принципова схема блоку управління функцією зворотного зв’язку

Початкове заповнення керуючого регістру відбудеться через вхід kluch\_upr. За допомогою сигналу q\_scht05 запис інформації відбудеться тільки в перших 5-ти тактах сигналу CLK, потім вхід kluch\_upr блокується і ЛРР починає виробляти рекурентну управляючу послідовність. З виходів 1-го та 2-го тригерів знімається інформація, яка служить для подальшого вибору зворотного зв’язку. Тільки на одному з виходів (upr0, upr1, upr2, upr3) в кожний момент часу буде присутня логічна одиниця, на решті виходів буде логічний нуль. Схема також має прямі виходи з блоку керування режимом роботи, які безпосередньо беруть участь у керуванні загальною схемою. Епюри функціонування зображені на рисунку (рис. 4.9).

Рисунок 4.9 —Епюри функціонування блоку управління функцією зворотного зв’язку.

#### 4.3.3.4 Регістр формування вихідної послідовності

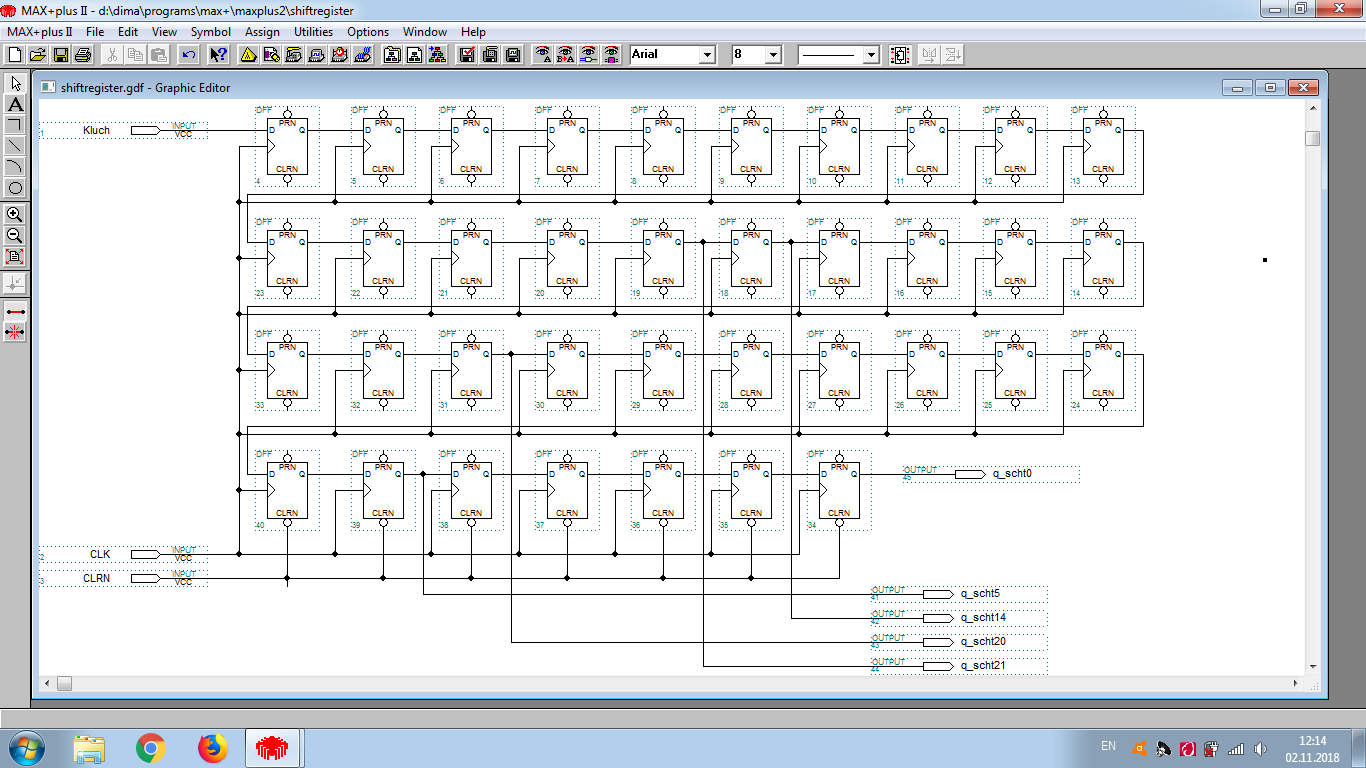
Основою побудови регістру є послідовне з’єднання 47 D-тригерів (рис 4.10)

Рисунок 4.10 — Принципова схема регістру зсуву.

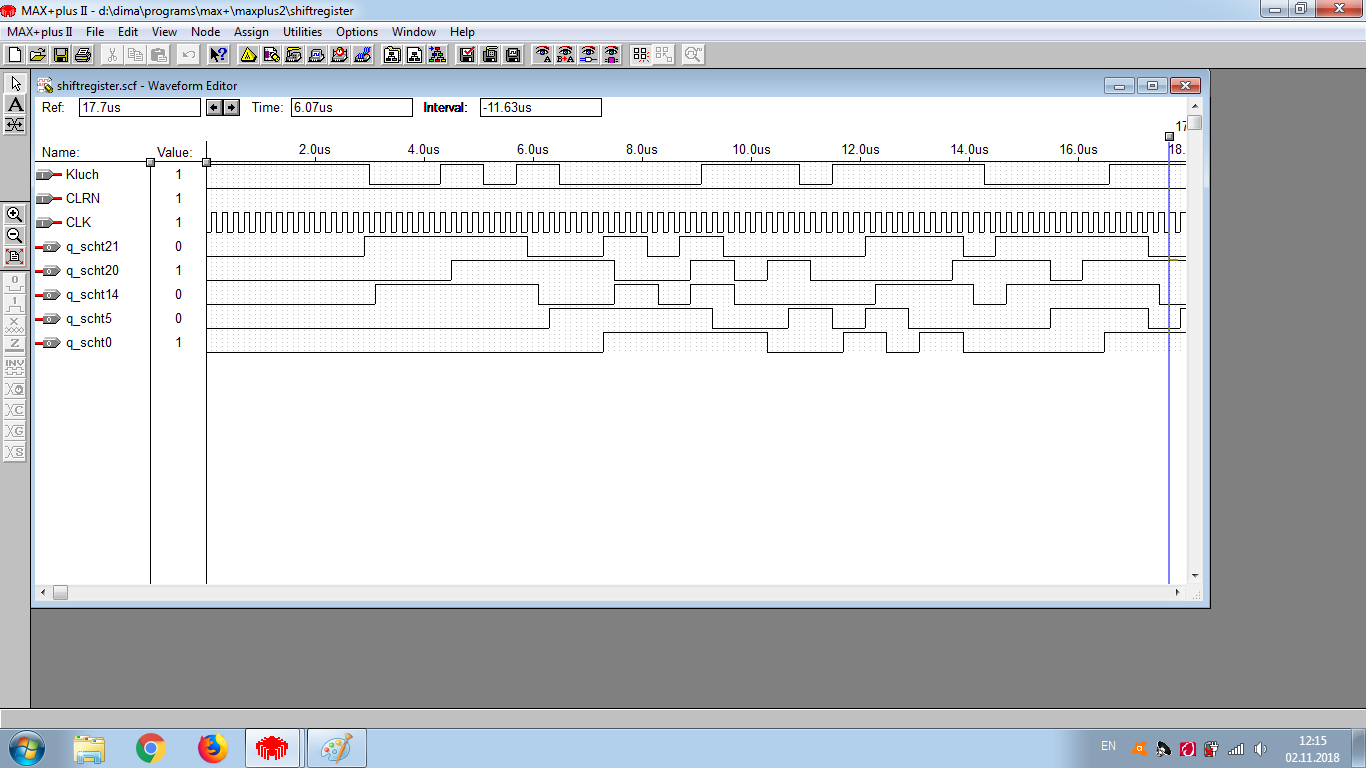
На вхід Clk та Clrn подаються відповідні глобальні сигнали. А на вхід Kluch подається початкове заповнення регістру. Запис в регістр відбувається при наявності на вході Clrn логічної одиниці. Виходи схеми є сигнали, яків в подальшому будуть приймати участь в зворотніх зв’язках: q\_scht0, q\_scht5, q\_scht14, q\_scht20, q\_scht21. Дані сигнали не присутні в кінцевому пристрої, але завдяки їм формуються відповідні зворотні зв’язки в ЛРР. Часові Діаграми функціонування регістру зсуву зображені на рисунку (рис. — 4.11).

Рисунок 4.11 — Епюри роботи регістру зсуву.

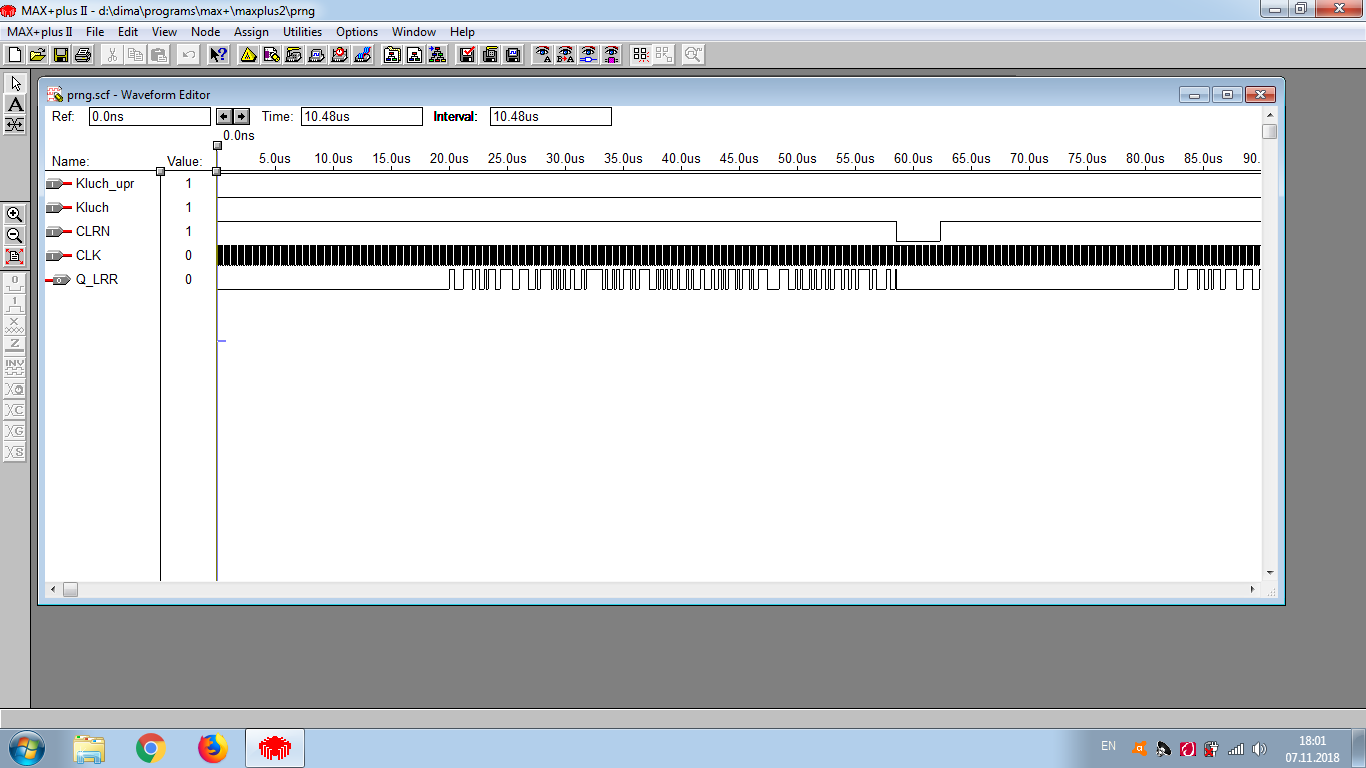
Загальний принцип роботи розробленого пристрою зображений на епюрах рисунку (рис. 4.12).

Рисунок 4.12 — Епюри функціонування генератора псевдовипадкової послідовності

# Висновки до розділу

У четвертому розділі було виконано практичну реалізацію генератора псевдовипадкових послідовностей. Для виконання було структуровано алгоритм автоматизованого проектування, а саме: концептуальний синтез; ввід в САПР; компіляція; функціональне моделювання; конфігурування; фізичне моделювання; установка в систему. Без наведених вище складових проектування пристрою буде неможливим. Створена в процесі роботи структурна схема пристрою складається із чотирьох складових: дільника частоти, блоку керування режимом роботи, блоку управління функцією зворотного зв’язку, регістру зсуву.

# ВИСНОВКИ

Метою кваліфікаційної роботи була реалізація генератора псевдовипадкових послідовностей.

Для досягнення поставленої мети у першому розділі розглядалися основні поняття на які опирається і на чому базується захист інформації. Описана значимість інформаційної безпеки, визначено понятійний апарат, яким повинен керуватися кожен спеціаліст у сфері кібербезпеки. Початок був взятий з історичної довідки, також розмежовано поняття симетричної та асиметричної криптографії, порівняно їх переваги, недоліки.

Наступним кроком передбачався розгляд концептуальних понять про генератори псевдовипадкових послідовностей та методи з використанням яких їх поява стає можливою. Розглянуто основні принципи побудови генераторів.

В третьому розділі було розглянуто основи створення та появи САПР, їх види та вплив на суспільну індустрію з виготовлення на віртуалізації інженерних проектів. САПР є індустріальними технологіями, що націлені на найважливіші галузі виробництва, визначаючи рівень розвитку та стратегічний потенціал.

І, нарешті, останній розділ відображає процес створення та логіку роботи пристрою який створювався. Отримана в процесі роботи структурна схема пристрою складається із чотирьох складових: дільника частоти, блоку керування режимом роботи, блоку управління функцією зворотного зв’язку, регістру зсуву. Головні компоненти, працюючи злагоджено, дозволили отримати псевдовипадкову послідовність, створення якої лягло за основне завдання роботи. Матеріал кваліфікаційної роботи може застосовувати в подальшому для навчання молодших спеціалістів у галузі кібербезпеки, а також для безпосереднього створення ключових послідовностей для апаратури шифрування.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Классические и квантовые генераторы случайных чисел.[Електронний ресурс] / В.Б. Задков, В.А. Юлия // Режим доступу: http://www. supercomputers.ru/index. php?id=441&option=com\_k2&view=item.

2. Quantum random number generator based on photonic emission in semiconductors [Електронний ресурс] / M.Stipčevića, B.Medved Rogina // Режим доступу: http://www-personal.umich.edu/~andrewcb/DSO/Papers/Random Number Generator/2007-quantum\_ random\_number\_generator\_based\_on\_photonic\_emission. pdf.

3. Quantum random number generation on a mobile phone. [Електронний ресурс] / Anthony Martin, Hugo Zbinden, Nicolas Gisin // Режим доступу: URI: http://arxiv.org/pdf/1405.0435v1. pdf.

4. Оптимизация шумовых параметров сигнальных цепей. . [Електронний ресурс] / Стив Эдвардс // Режим доступу: http://www.symmetron. ru/articles/noise-reduction-1. pdf.

5. The Intel random number generator [Електронний ресурс] / Benjamin Jun, Paul Kocher // Режим доступу: https://www.rambus.com/wpcontent/uploads/2015/08/IntelRNG. pdf.

6. Behind Intel's new random number generator [Електронний ресурс] / Greg Taylor, George Cox // Режим доступу: http://spectrum.ieee. org/computing/hardware/behind-intels-new-randomnumber-generator.

7. Random sequence generator based on avalanche noise [Електронний ресурс] / J. Holden. // Режим доступу: http://holdenc.altervista.org/avalanche/.

8. On-chip true random number generation in nanometer CMOS [Електронний ресурс] / Vikram Belur Suresh // Режим доступу: http://scholarworks. umass.edu/cgi/viewcontent. cgi?article=1872&context=theses.

9. Кольцевой генератор и его неповторимый температурный коэффициент ли- нейной регрессии [Електронний ресурс] / Прощеряков, А.А., Иванюк А.А // Режим доступу: http://libeldoc.bsuir.by/bitstream/123456789/3312/1/Кольцевой генератор и его неповторимый температурный коэффициент линейной регрессии.pdf.

10. Иванов М.А. Теория, применение и оценка качества генераторов

псевдослучайных последовательностей / М.А. Иванов, И.В. Чугунков – М.: Изд-во «КУДИЦ-ОБРАЗ», 2003. – 240 с.

11. Горбенко І.Д. Прикладна криптологія: Теорія. Практика. Застосування:

монографія / І.Д. Горбенко, Ю.І. Горбенко. – Харків.: Вид-во «Форт», 2012. – 880 с.

12. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications [Електронний ресурс] // Режим доступу:http://csrc.nist.gov/publications/nistpubs/800-22-rev1a/SP800-22rev1a.pdf

13. Максименко С.Д., Соловієнко В.О. Загальна психологія: Навч. посібник. – К.: МАУП, 2000. – 256 с.

14. Хорошко B.O., Азаров О.Д., Шелест М.Є., Яремчук Ю.Є. Основи комп 'ютерної криптографії : Навчальний посібник для студентів і аспірантів. — Вінниця: ВДТУ, 2003, — 143 с.

15. Юдін О.К. Кодування в інформаційно-комунікаційних мережах: – Монографія. – К.: НАУ, 2007. – 308с.

16. A.N. Akansu, R.A. Haddad, Multiresolution Signal Decomposition: Transforms, Sub- bands and Wavelets, Academic Press Inc., New York, 1992.

17. Ahmet M. Eskicioglu and Paul S. Fisher. Image Auality Measures And Their Per­formance. IEEE Transactions on Communication, 43(12): 2959-2965, December 1995.

18. B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York // John Wiley and Sons, 1996.

19. Benham. D. Fast watermarking of DCT-based compressed images / D. Benham, N. Memon, B. L. Yeo, M. Yeung, // Proc. of Int Conf Image Science, Systems, and Technology, 1997, Las Vegas, NV. — 1997. — P. 243-253.

20. C. Cachin, An Information-Theoretic Model for Steganography. In: Information Hid­ing – 2nd International Workshop, Springer as Lecture Notes in Computing Science, vol.1525, April 1998, pp.306-318.

21. C.E. Shannon, A Mathematical Theory of Communication. Bell System Technical Journal, 27 (1948), pp.379-423, 623-656.

22. C.J. van den Branden Lambrecht and J.E. Farrell. Perceptual Quality Metric for Digi­tally Coded Color Images. In: Proceeding of EUSIPCO, pp. 1175-1178, Trieste, Italy, September 1996.

23. Daubechies I. Orthonormal basis of compactly supported wavelets // Comm. Pure Appl. Math, v. XLI. —1988. — P. 909-996.

24. E. Koch, J. Zhao, Towards Robust and Hidden Image Copyright Labeling. // IEEE Workshop on Nonlinear Signal and Image Processing, Greece, June 20-22, 1995. Pp.123-132.

25. Fidrich J. Steganography in Digital Media: Principles, Algorithms, and Applications / J. Fidrich — Cambridge: Cambride University Press, 2009. — 438 c.

26. Image Steganography Techniques: An Overview / H. Nagham, Y. Abid, R. Badlishah Ahmad, Osamah M. Al-Qershi // International Journal of Computer Science and Security. — 2012. — Volume 6, Issue 3. — P. 168-187.

27. J. Hernandez, F. Perez-Gonzalez, J. Rodriguez, G. Nieto, Performance Analysis of a 2-D Multipulse Amplitude Modulation Scheme for Data Hiding and Watermarking of Still Images // IEEE Journal on .Selected Areas in Communications. 1998. Vol. 16, № 5. Pp. 510-525.

28. J. Zhao, E. Koch, Embedding Robust Labels into Images for Copyright Protection. // Proceeding of the Int. Congress on Intellectual Property Rights for Specialized In­formation, Knowledge and New Techniques, Munich-Vienna, Verlag, Aug. 1995, pp.242-251.

29. Jadav. Y. Comparison of LSB and Subband DCT Technique for Image Watermarking / Jadav, Y. // Conference on Advances in Communication and Control Systems 2013. — 2013. — P. 398-401.

30. Jiansheng M. A Digital Watermarking Algorithm Based On DCT and DWT / M. Jiansheng, L. Sukang, T. Xiaomei// Proc. of the 2009 International Symposium on Web Information Systems and Applications. — 2009. — P. 104-107.

31. Khalid Sayood, Introduction to Data Compression, chapter 7, p. 142. Morgan Kaufmann Publishers, 1996.

32. Kutter. M. A fair benchmark for image watermarking systems / M. Kutter, F. Petitcolas // Proc. of SPIE: Security and Watermarking of Multimedia Contents, 14-15 June 1999, San Jose, France. — 1999. — Volume 3657.— P. 226-239.

33. Marvel L.M. Capacity of the additive steganographic channel, Methodology of Spread- Spectrum. Image Steganography / L. M. Marvel, C.G. Boncelet Jr., Charles T. Retter // Proc. of IEEE transactions on image processing, August 1999. — 1999. — Vol.8, No.8. — PP. 1075-1083.