МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп’ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Люлькун Руслан Олегович

Апаратний модуль дешифрування криптоалгоритму Рабіна / Hardware module for Rabin cryptoalgorithm decryption

напрям підготовки: 6.050102 - Комп’ютерна інженерія
фахове спрямування - Комп’ютерні системи та мережі
Бакалаврська робота

Виконав студент групи КСМ 42/1
Люлькун Руслан Олегович

Науковий керівник:
Маслияк Б.О

Тернопіль - 2018

РЕЗЮМЕ

Дипломний проект містить 52 сторінок пояснюючої записки, 22 рисунки, 10 таблиць, 2 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою дипломної роботи є розроблення апаратного модуля дешифрування криптоалгоритму Рабіна" отримано наступні результати:

Проаналізовано алгоритм дешифрації по криптосистемі Рабіна. У відповідності до алгоритму розроблена узагальнена структура апаратної реалізації модуля дешифрації, яка складається з блоку визначення модуля числа А по таємним ключам p і q, блоку визначення коренів *V,* блоку формування лишків *b* і *d*, блоку визначення НСД, блоку дешифрація тексту.

Розроблено структурні схеми знаходження $A\mathrm{mod}(p)$, підкореневого виразу та визначення значення квадратного кореня $\sqrt{\mathrm{K}}\,mod(p) = V$.

Апаратну реалізацію знаходження числа за модулем здійснено в імітаційному пакеті NI Multisim. Пакет прикладних программ NI Multisim є емулятором електронних схем, який дозволяє мінімізувати час розробки електричних схем цифрових пристроїв. Робота базується на основі технології віртуальних електронних приладів (SPICE-моделі) та аналізу функціонування електричних схем та їх тестуванні.

Розроблено електричну схему визначення модуля числа за таємним ключем, схему знаходження підкореневого виразу, апаратну реалізацію операції підняття до квадрату

Ключові слова: МІКРОСХЕМА, СТРУКТУРНА СХЕМА, ЕЛЕКТРИЧНА СХЕМА, КОМБІНАЦІЙНА ЛОГІКА, КРИПТОАЛГОРИТМ.

RESUME

The diploma project contains 52 pages of explanatory note, 22 figures, 10 tables, 2 appendices. Volume of graphic material 2 sheets of A3 format.

The purpose of the thesis is to develop a hardware module for decoding the crypto algorithm Rabin "obtained the following results:

The decryption algorithm for the Rabin cryptosystem is analyzed. In accordance with the algorithm, a generalized structure of the hardware implementation of the decryption module is developed, which consists of the unit for determining the module of the number A by secret keys p i q, the unit for determining the roots V, the unit for forming surpluses b i d, the unit for determining NSD, the unit for decrypting text.

Structural schemes for finding Amod (p), subroot expression and determining the value of the square root √ (K) mod (p) = V are developed.

The hardware implementation of finding the number modulo is carried out in the simulation package NI Multisim. The NI Multisim application package is an electronic circuit emulator that minimizes the development time of electrical circuits of digital devices. The work is based on the technology of virtual electronic devices (SPICE-models) and analysis of the operation of electrical circuits and their testing.

The electric scheme of definition of the modulus of number on a secret key, the scheme of finding of a root expression, hardware realization of operation of raising to a square is developed

Keywords: MICROSCHEME, STRUCTURAL SCHEME, ELECTRICAL SCHEME, COMBINATION LOGIC, CRYPTOALGORITHM.

ЗМІСТ

					ДП.КСМ.07133/14.00.00.000ПЗ			
Зм.	*Арк*	*№ докум.*	*Підпис*	*Дата*				
Розробив		Люлькун Р.О.			АПАРАТНИЙ МОДУЛЬ ДЕШИФРУВАННЯ КРИПТОАЛГОРИТМУ РАБІНА	*Літ.*	*Аркуш*	*Аркушів*
Перевірив		Маслияк Б.О.						51
		Паздрій				ТНЕУ, ФКІТ, КСМ-42/1		
Н. Контр.		Гураль						
Затв.		*Березький О.М.*						

ВСТУП

Криптографія (іноді вживають термін криптологія) - галузь знань, що вивчає тайнопис (криптографія) і методи її розкриття (криптоаналіз). Криптографія вважається розділом математики.

До недавнього часу всі дослідження в цій галузі були лише закритими, але в останні кілька років у нас і за кордоном стало з'являтися все більше публікацій у пресі. Почасти пом'якшення секретності пояснюється тим, що стало вже неможливим приховувати накопичене кількість інформації. З іншого боку, криптографія все більше використовується у цивільних галузях, що вимагає розкриття відомостей.

Процес криптографічного закриття даних може здійснюватися як програмно, так і апаратно. Апаратна реалізація відрізняється істотно більшою вартістю, проте їй притаманні і переваги: висока продуктивність, простота, захищеність і т.д.

Програмна реалізація більш практична, допускає відому гнучкість у використанні.

Для сучасних криптографічних систем захисту інформації зформульовані наступні загальноприйняті вимоги:

- Зашифроване повідомлення повинно піддаватися читанню тільки при наявності ключа.
- Число операцій, необхідних для визначення використаного ключа шифрування за фрагментом шифрованого повідомлення і відповідного йому відкритого тексту, повинно бути не менше загального числа можливих ключів.
- Число операцій, необхідних для розшифрування інформації шляхом перебору всіляких ключів повинно мати строгу нижню оцінку і виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережевих обчислень).

- Знання алгоритму шифрування не повинно впливати на надійність захисту.
- Незначна зміна ключа повинна приводити до істотної зміни виду зашифрованого повідомлення навіть при використанні одного і того ж ключа.
- Структурні елементи алгоритму шифрування повинні бути незмінними.
- Додаткові біти, що вводяться в повідомлення в процесі шифрування, повинні бути повністю та надійно сховані в зашифрованому тексті.
- Довжина шифрованого тексту повинна бути рівною довжині вихідного тексту.
- Не повинно бути простих і легко встановлюваних залежностей між ключами, які послідовно використовуються в процесі шифрування.
- Будь-який ключ з безлічі можливих повинен забезпечувати надійний захист інформації.

Алгоритм повинен допускати як програмно, так і апаратну реалізацію, при цьому зміна довжини ключа не повинно вести до якісного погіршення алгоритму шифрування.

1 ЗАХИСТ ІНФОРМАЦІЇ В КОМП'ЮТЕРНИХ СИСТЕМАХ

1.1 Особливості систем захисту інформації

Мета криптографічного системи полягає в тому, щоб зашифрувати осмислений вихідний текст (також званий відкритим текстом), отримавши в результаті абсолютно безглуздий на погляд шифрований текст (шифротекст, криптограма). Одержувач, якій він призначений, повинен бути здатний розшифрувати (кажуть також "дешифрувати") цей шифротекст, відновивши, таким чином, відповідний йому відкритий текст. Криптографія припускає наявність трьох компонентів: даних, ключа і криптографічного перетворення [1].

Шифрування - Перетворювальний процес: вихідний текст, що має також назву відкритого тексту, замінюється шифрованим текстом.

Дешифрування - Зворотний шифруванню процес. На основі ключа шифрований текст перетвориться у вихідний.

Ключ - інформація, необхідна для безперешкодного шифрування й дешифрування текстів.

Вважається, що криптографічне перетворення відомо всім, але, не знаючи ключа, за допомогою якого користувач закрив сенс повідомлення від цікавих очей, потрібно витратити неймовірно багато зусиль на відновлення тексту повідомлення. (Слід ще раз повторити, що немає абсолютно сталого від розтину шифрування. Якість шифру визначається лише грошима, які потрібно викласти за його розкриття від $ 10 і до $ 1000000) [2, 3].

Розкриттям криптосистеми називається результат роботи криптоаналітиків, що приводить до можливості ефективного розкриття будь-якого, зашифрованого за допомогою даної криптосистеми, відкритого тексту. Ступінь нездатності криптосистеми до розкриття називається її стійкістю.

Криптосистеми поділяються на симетричні (з секретним ключем) і з відкритим ключем. У симетричних криптосистемах і для шифрування, і для дешифрування використовується один і той самий ключ [1-4].

У системах з відкритим ключем використовуються два ключі - відкритий і закритий, які математично пов'язані один з одним. Інформація шифрується за допомогою відкритого ключа, що доступний усім бажаючим, а розшифровується за допомогою закритого ключа, відомого тільки одержувачу повідомлення.

Симетричні криптосистеми (також симетричне шифрування, симетричні шифри) (англ. Symmetric-key algorithm) - спосіб шифрування, в якому для шифрування і розшифрування застосовується один і той же криптографічний ключ. До винаходу схеми асиметричного шифрування єдиним існуючим способом було симетричне шифрування. Ключ алгоритму повинен зберігатися в секреті обома сторонами. Алгоритм шифрування вибирається сторонами до початку обміну повідомленнями. В даний час симетричні шифри - це:

– блокові шифри. Обробляють інформацію блоками певної довжини (зазвичай 64, 128 біт), застосовуючи до блоку ключ в установленому порядку, як правило, кількома циклами перемішування і підстановки, званими раундами. Результатом повторення раундів є лавинний ефект - наростаюча втрата відповідності бітів між блоками відкритих і зашифрованих даних.

– потокові шифри, в яких шифрування проводиться над кожним бітом або байтом вихідного (відкритого) тексту з використанням гамування. Гамування - метод симетричного шифрування, що полягає в «накладенні» послідовності, що складається з випадкових чисел, на відкритий текст. Послідовність випадкових чисел називається гамма-послідовністю і використовується для шифрування і розшифрування даних. Підсумовування, зазвичай, виконується в будь-якому кінцевому полі. Наприклад, в полі Галуа GF (2) підсумовування набирає вигляду операції «виключне АБО (xor)».

Поточний шифр може бути легко створений на основі блочного (наприклад, ГОСТ 28147-89 в режимі гамування), запущеного в спеціальному режимі.

Більшість симетричних шифрів використовують складні комбінації великої кількості підстановок і перестановок. Багато таких шифрів виконуються в кілька (іноді до 80) проходів, використовуючи на кожному проході «ключ проходу». Множина «ключів проходу» для всіх проходів називається «розкладом ключів» (key schedule).

Типовим способом побудови алгоритмів симетричного шифрування є мережа Фейстеля. Алгоритм будує схему шифрування на основі функції F (D, K), де D - порція даних розміром вдвічі менше блоку шифрування, а K - «ключ проходу» для даного проходу. Від функції не потрібно оборотність - зворотна їй функція може бути невідома. Переваги мережі Фейстеля - майже повний збіг дешифрування з шифруванням (єдина відмінність - зворотний порядок «ключів проходу» в розкладі), що значно полегшує апаратну реалізацію.

Операція перестановки перемішує біти повідомлення по якомусь закону. В апаратних реалізаціях вона тривіально реалізується як переключення провідників. Саме операції перестановки дають можливість досягнення «ефекту лавини». Операція перестановки лінійна - f (a) xor f (b) == f (a xor b)

Огляд існуючих програмних технологій показує ряд програмних засобів, що здійснюють реалізацію процедури захисту інформації в комп'ютерних мережах [5-7]. Серед них виділяються наступні.

Advanced Encryption Standard (AES), також відомий як Rijndael – симетричний алгоритм блочного шифрування (розмір блока 128 біт, ключ 128/192/256 біт), прийнятий як стандарт шифрування урядом США за результатами конкурсу AES. Цей алгоритм добре проаналізований і зараз широко використовується, як це було з його попередником DES. Національний інститут стандартів і технологій США (англ. National Institute of Standards and Technology, NIST) опублікував специфікацію AES 26

листопада 2001 після п'ятирічного періоду, в ході якого були створені і оцінені 15 кандидатур. 26 травня 2002 AES було оголошено стандартом шифрування. Станом на 2009 рік AES є одним з найпоширеніших алгоритмів симетричного шифрування. Підтримка AES (і тільки його) введена фірмою Intel в сімейство процесорів x86 починаючи з Intel Core i7-980X Extreme Edition, а потім на процесорах Sandy Bridge.

TrueCrypt- комп'ютерна програма для шифрування «на льоту» (On-the-fly encryption) для 32 - і 64-розрядних операційних систем сімейств Microsoft Windows NT 5 і новіше (GUI-інтерфейс), Linux і Mac OS X. Вона дозволяє створювати віртуальний зашифрований логічний диск, що зберігається у вигляді файлу. За допомогою TrueCrypt також можна повністю шифрувати розділ жорсткого диска або іншого носія інформації, такої як флоппі-диск або USB флеш-пам'ять. Всі збережені дані в томі TrueCrypt повністю шифруються, включаючи імена файлів і каталогів. Змонтований тому TrueCrypt подібний звичайному логічному диску, тому з ним можна працювати за допомогою звичайних утиліт перевірки та дефрагментації файлової системи.

Ліцензія програми вважалася вільною, однак при її перевірці для включення TrueCrypt в Fedora в жовтні 2008 року були виявлені небезпечні і які роблять її невільною неоднозначності. В список підтримуваних TrueCrypt 6.2 алгоритмів шифрування входять AES, Serpent і Twofish. Попередні версії програми також підтримували алгоритми з розміром блоку 64 біта (Потрійний DES, Blowfish, CAST5) (включаючи версії 5.x, яка могла відкривати, але не створювати розділи, захищені цими алгоритмами). До листопада в ліцензію були внесені виправлення.

Програма DigiSecret застосовує стійкі і перевірені часом алгоритми шифрування, здатна архівувати файли, і повністю їх знищувати - файли видаляються, і їх місце на диску багато разів перезаписується по спеціальному алгоритму, щоб виключити можливість відновлення даних. Загалом то спрямована DigiSecret саме на створення закодованих архівів і

передача цих архівів між користувачами. Інтерфейс програми, по суті, нагадує різні архіватори: для створення архіву потрібно перетягнути потрібні файли в основне вікно програми і, вибравши відповідний пункт меню, створити закодований архів. Виробник стверджує, що використовується в DigiSecret механізм компресії надзвичайно ефективний, проте на практиці виявляється, що навіть вбудований в Windows XP механізм компресії ZIP справляється з цим завданням краще. Втім, не це головне завдання програми, і її рівень достатній для середнього користувача. При створенні архіву можна вибрати одну з дев'яти ступенів стиснення відповідно до бажаної швидкістю архівації, або відключити компресію зовсім, і тоді програма просто зашифрує вміст файлів. DigiSecret вміє створювати і SFX архіви, що напевно стане в нагоді в тих випадках, коли виникає необхідність передати інформацію людині, яка не має своєї копії цієї програми. У створеному архіві зберігається структура папок, хоча в основному вікні програми всі файли «звалені в купу». Програма пропонує на вибір декілька алгоритмів кодування – CAST (128-бітний ключ), Blowfish (448-бітний ключ), Twofish (256-бітний ключ) і Rijndael (також відомий як AES, 256-бітний ключ).

1.2 Методи побудови апаратно-програмних систем захисту

Криптосистема Рабіна (M. Rabin) є варіантом криптосистеми RSA. RSA базується на зведенні в ступінь порівнянь. Криптосистема Рабіна базується на квадратичних порівняннях, і її можна представити як криптографічну систему RSA. Для криптосистеми Рабіна було доведено, що відновлення початкового тексту з зашифрованого настільки ж важке, як факторизація великих чисел. Система Рабіна стала першою асиметричною криптосистемою, для якої було виконано такий доказ. Складність відновлення пов'язана з трудністю визначення квадратного кореня за

модулем складеного числа . Завдання факторизації і завдання по визначенню квадратного кореня еквівалентні, тобто:

– знаючи прості дільники числа можна знаходити квадратний корінь по модулю ;

– вміючи знаходити квадратний корінь по модулю , можна розкласти на прості множники.

– Початкове повідомлення m (текст) шифрується за допомогою відкритого ключа — числа n за такою формулою:

– $c = m^2 \bmod N$.

Завдяки використанню множення по модулю швидкість шифрування системи Рабина більше, ніж швидкість шифрування за методом RSA, навіть якщо в останньому випадку вибрати невелике значення експоненти.

Для розшифровки повідомлення необхідно знати закритий ключ — числа p і q. Процес розшифровки виглядає наступним чином:

- спочатку, використовуючи алгоритм Евкліда, з рівняння знаходять числа ;

- далі, використовуючи китайську теорему про залишки, обчислюють чотири числа:

+ -

Одне з цих чисел є істинним відкритим текстом m.

Алгоритм Евкліда - ефективний алгоритм для знаходження найбільшого загального дільника двох цілих чисел.

Китайська теорема про залишки — один з основних результатів елементарної теорії чисел. Використовуючи позначення модульної арифметики її можна сформулювати наступним чином. Нехай $y_1, y_2 \ldots y_k$ довільні цілі числа, а попарно взаємно прості числа. Тоді наступна система:

$$
\begin{aligned}
x &\equiv y_1 \pmod{n_1} \\
x &\equiv y_2 \pmod{n_2} \\
&\vdots \\
x &\equiv y_k \pmod{n_k}
\end{aligned}
$$

має розв'язок і всі її розв'язки рівні за модулем .

Алгоритм Рабіна схожий на кодування RSA, але замість зведення повідомлення в степінь e при шифруванні використовується операція піднесення блоку повідомлення в квадрат, що сприятливо позначається на швидкості виконання алгоритму без шкоди криптостійкості. Для декодування китайська теорема про залишки застосована разом з двома зведення в степінь по модулю. Тут ефективність порівнянна RSA. Вибір потрібного тексту з чотирьох призводить до додаткових обчислювальним затратам. І це послужило тому, що криптосистема Рабіна не отримала широкого практичного використання.

Велика перевага криптосистеми Рабіна полягає в тому, що випадковий текст може бути відновлений повністю від зашифрованого тексту тільки за умови, що дешифрувальник здатний до ефективної факторизації відкритого ключа *N*. Криптосистема Рабіна є доказово стійкою до атаки на основі підібраного відкритого зашифрованого тексту в рамках підходу «все або нічого», тоді і тільки тоді, коли завдання про розкладання цілого числа на прості множники є важкою.

Стійкість за принципом «все або нічого» полягає в тому, що, маючи текст, зашифрований певним алгоритмом, атакуючий повинен відновити блок вихідного тексту, розмір якого, як правило, визначається параметром безпеки криптосистеми. Маючи вихідний і зашифрований текст, атакуючий повинен відновити цілий блок секретного ключа. При цьому атакуючий або домагається повного успіху, або не отримує нічого. Під словом «нічого»

мається на увазі, що атакуючий не має ніякої секретної інформації ні до, ні після безуспішної атаки.

1.3 Обґрунтування вибору компонентної бази та постановка задачі

Процес криптографічного закриття даних може здійснюватися як програмно, так і апаратно. Апаратна реалізація відрізняється суттєво більшою вартістю, однак їй властиві й переваги: висока продуктивність, простота, захищеність. Програмна реалізація більш практична, допускає відому гнучкість у використанні.

Для сучасних криптографічних систем захисту інформації сформульовані наступні загальноприйняті вимоги:

− зашифроване повідомлення повинне піддаватися читанню тільки при наявності ключа;

− число операції, необхідних для визначення використаного ключа шифрування по фрагменту шифрованого повідомлення й відповідного йому відкритого тексту, повинне бути не менше загального числа можливих ключів;

− число операцій, необхідних для розшифровування інформації шляхом перебору всіляких ключів, повинне мати строгу нижню оцінку й виходити за межі можливостей сучасних комп'ютерів (з урахуванням можливості використання мережних обчислень) або вимагати неприйнятно високих витрат на ці обчислення;

− знання алгоритму шифрування не повинне впливати на надійність захисту;

− незначна зміна ключа повинна приводити до істотної зміни виду зашифрованого повідомлення навіть при шифруванні того самого вихідного тексту;

– незначна зміна вихідного тексту повинне приводити до істотної зміни виду зашифрованого повідомлення навіть при використанні того самого ключа;

– структурні елементи алгоритму шифрування повинні бути незмінними;

– додаткові біти, що уводяться в повідомлення в процесі шифрування, повинні бути повністю й надійно сховані в шифрованому тексті;

– довжина шифрованого тексту не повинна перевершувати довжину вихідного тексту;

– не повинне бути простих і легко встановлюваних залежностей між ключами, послідовно використовуваними в процесі шифрування;

– будь-який ключ із множини можливих повинен забезпечувати надійний захист інформації;

– алгоритм повинен допускати як програмну, так і апаратну реалізацію, при цьому зміна довжини ключа не повинне вести до якісного погіршення алгоритму шифрування.

Таким чином, на основі аналізу особливостей реалізації криптоалгоритмів в комп'ютерних системах в даному розділі запропоновано та обґрунтовано апаратну реалізацію модуля дешифрації інформації по алгоритму Рабіна.

2 ПРОЕКТУВАННЯ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЇ

2.1 Захист інформації в криптосистемі Рабіна

Як було показано в розділі 1.2, криптосистема Рабіна оперує таємними ключами p і q, що є простими числами. Просте число — це натуральне число, яке має рівно два різних натуральних дільники (лише 1 і саме це число). Решту чисел, крім одиниці, називають складеними. Таким чином, всі натуральні числа, більші від одиниці, розбивають на прості і складені. Теорія чисел вивчає властивості простих чисел. Основна теорема арифметики стверджує, що кожне натуральне число більше одиниці (1), можна представити як добуток простих чисел, причому, в єдиний спосіб з точністю до порядку множників. Таким чином, прості числа — це елементарні «будівельні блоки» натуральних чисел.

Представлення натурального числа у вигляді добутку простих називають розкладом на прості або факторизацією числа. На сьогодні невідомі поліноміальні алгоритми факторизації чисел, хоча і не доведено, що таких алгоритмів не існує (тут і далі мова йде про поліноміальну залежність часу роботи алгоритму від логарифму розміру числа, тобто від кількості його цифр). На припущенні про високу обчислювальну складність задачі факторизації базується криптосистема RSA. Решето Ератосфена, решето Сундарама та решето Аткіна дають прості способи складання початкового списку простих чисел до певного значення. Однак на практиці замість отримання списку простих чисел найчастіше потрібно перевірити, чи є дане число простим. Алгоритми, які вирішують це завдання, називають тестами простоти. Існує безліч поліноміальних тестів простоти, але більшість з них є стохастичні (наприклад, тест Міллера — Рабіна) і використовуються для потреб криптографії. Тільки в 2002 році було доведено, що завдання перевірки на простоту в загальному вигляді можна розв'язати за

поліноміальний час, але запропонований детермінований алгоритм має досить велику складність, що ускладнює його застосування на практиці.

Найбільш загальна схема функціонування криптосистеми представлена на рисунку 2.1.

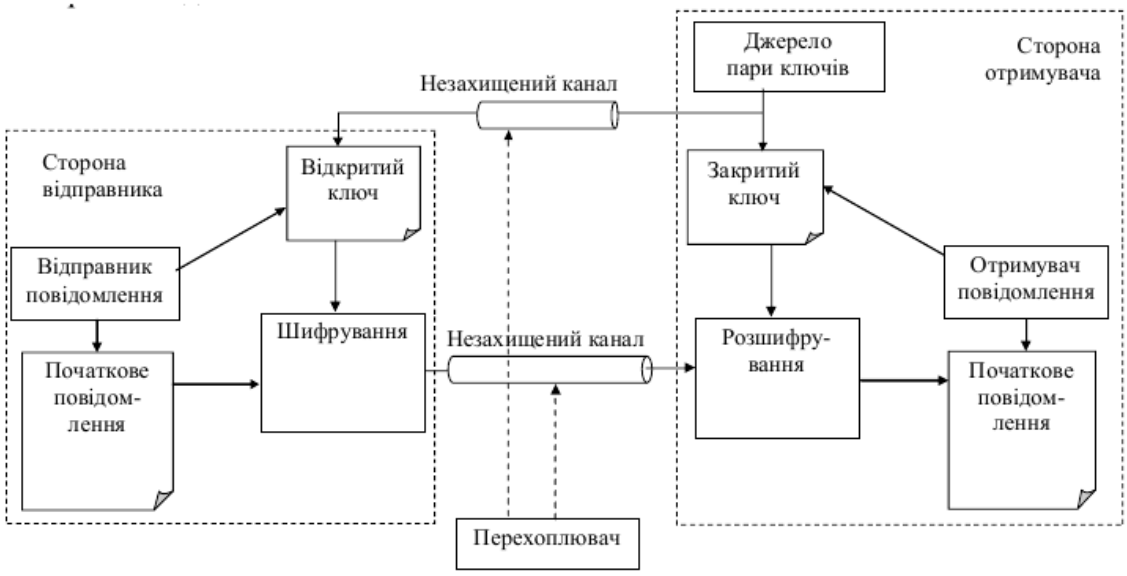

Рисунок 2.1 – Узагальнена структура асиметричної криптосистеми

Її роботу найбільш загально можна описати в такий спосіб:

– передавальна сторона одержує із джерела ключів ключ для шифрування свого повідомлення.

– передавальна сторона зашифровує текст повідомлення й передає криптограму е в відкритий канал зв'язки в напрямку одержувача.

– одержувач повідомлення одержує із джерела ключів ключ, за допомогою якого можна розшифрувати отриману криптограму.

– одержувач розшифровує криптограму Е.

Враховуючи особливості алгоритму Рабіна розробимо структуру системи, що може його реалізувати – рисунок 2.2. Як видно з рисунку, важливими процедурами є блок формування таємних ключів та блок визначення квадрату вхідного коду за модулем N ($x^2 mod N$). Тому апаратна реалізація даних блоків є актуальною.

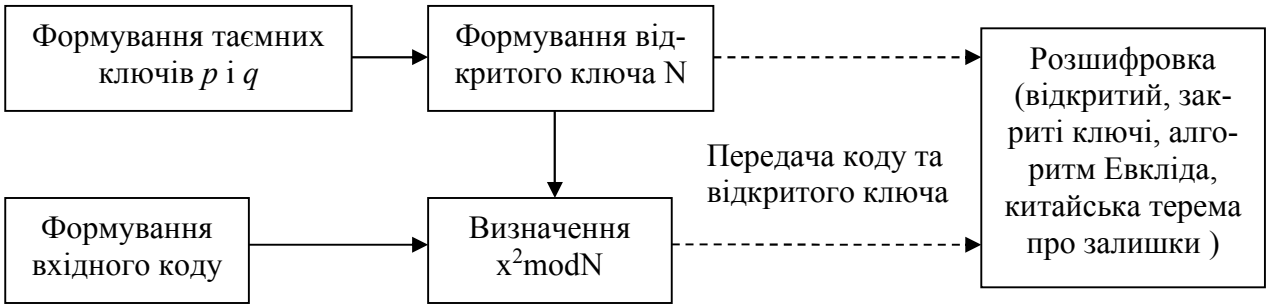

Рисунок 2.2 – Загальна структура роботи криптосистеми Рабіна

Основою для побудови блоку формування таємних ключів є таблиця простих чисел – таблиця 2.1.

Таблиця 2.1 - Таблиця перших 500 простих чисел

2	3	5	7	11	13	17	19	23	29	31	37	41	43	47	53	59	61	67	71
73	79	83	89	97	101	103	107	109	113	127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229	233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349	353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463	467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601	607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733	739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863	877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997	1009	1013	1019	1021	1031	1033	1039	1049	1051	1061	1063	1069
1087	1091	1093	1097	1103	1109	1117	1123	1129	1151	1153	1163	1171	1181	1187	1193	1201	1213	1217	1223
1229	1231	1237	1249	1259	1277	1279	1283	1289	1291	1297	1301	1303	1307	1319	1321	1327	1361	1367	1373
1381	1399	1409	1423	1427	1429	1433	1439	1447	1451	1453	1459	1471	1481	1483	1487	1489	1493	1499	1511
1523	1531	1543	1549	1553	1559	1567	1571	1579	1583	1597	1601	1607	1609	1613	1619	1621	1627	1637	1657
1663	1667	1669	1693	1697	1699	1709	1721	1723	1733	1741	1747	1753	1759	1777	1783	1787	1789	1801	1811
1823	1831	1847	1861	1867	1871	1873	1877	1879	1889	1901	1907	1913	1931	1933	1949	1951	1973	1979	1987
1993	1997	1999	2003	2011	2017	2027	2029	2039	2053	2063	2069	2081	2083	2087	2089	2099	2111	2113	2129
2131	2137	2141	2143	2153	2161	2179	2203	2207	2213	2221	2237	2239	2243	2251	2267	2269	2273	2281	2287
2293	2297	2309	2311	2333	2339	2341	2347	2351	2357	2371	2377	2381	2383	2389	2393	2399	2411	2417	2423
2437	2441	2447	2459	2467	2473	2477	2503	2521	2531	2539	2543	2549	2551	2557	2579	2591	2593	2609	2617
2621	2633	2647	2657	2659	2663	2671	2677	2683	2687	2689	2693	2699	2707	2711	2713	2719	2729	2731	2741
2749	2753	2767	2777	2789	2791	2797	2801	2803	2819	2833	2837	2843	2851	2857	2861	2879	2887	2897	2903
2909	2917	2927	2939	2953	2957	2963	2969	2971	2999	3001	3011	3019	3023	3037	3041	3049	3061	3067	3079
3083	3089	3109	3119	3121	3137	3163	3167	3169	3181	3187	3191	3203	3209	3217	3221	3229	3251	3253	3257
3259	3271	3299	3301	3307	3313	3319	3323	3329	3331	3343	3347	3359	3361	3371	3373	3389	3391	3407	3413
3433	3449	3457	3461	3463	3467	3469	3491	3499	3511	3517	3527	3529	3533	3539	3541	3547	3557	3559	3571

Апаратне прискорення процесу шифрування буде також досягатися при виконанні операції піднесення шифрованого блоку до квадрату.

Таким чином, розроблена структура реалізації алгоритму Рабіна дозволила виділити блоки для їх апаратної реалізації.

2.2 Структура апаратного модуля дешифрації

Розробку структурної схеми модуля дешифрації розпочнемо з уточнення алгоритму дешифрації по криптосистемі Рабіна. Як вже було показано в попередньому розділі, в криптосистемі Рабіна є закриті ключі p та q і відкритий – N. Закриті ключі є простими числами, приклад яких приведено в таблиці 2.1. Шифрація йде блоками цифр, які по модулю менші – N. Оскільки блоки формуються послідовно, розглянемо процес дешифрації на прикладі типового блоку даних. Для позначення блоку даних використаємо символ – A. На першому етапі блок шифротексту А формується відповідно до таємних ключів у формі:

$$A \bmod (p) = K;$$
$$A \bmod (q) = L.$$

На другому етапі шукаємо корінь квадратний від K за модулем p:

$$\sqrt{\mathrm{K}}\, mod(p) = V.$$

Методика пошуку кореню передбачає, що якщо корінь від K не береться, то до K додаємо p і шукаємо корінь квадратний. Якщо корінь не знаходиться, то до K додаємо $2p$ і пробуємо взяти корінь. Процес

повторюється до взяття кореня. В результаті отримуємо *b1*. Друге число *b2* отримуємо шляхом віднімання *p-b1*.

Аналогічні дії відбуваються по модулю *q*. В результаті отримаємо аналогічну пару чисел *d1* та *d2*.

Відповідно до китайської теореми про лишки (остачі) формуємо чотири комбінації:

b1, d1;

b1, d2;

b2, d1;

b2, d2.

Наступним кроком є пошук найбільшого спільного дільника (НСД) таємних ключів *p* і *q*. Найбільший спільний дільник (НСД) двох чисел це найбільше число, що ділить обидва дані числа без остачі. Алгоритм Евкліда заснований на тому, що НСД не змінюється, якщо від більшого числа відняти менше. Наприклад, 21 є НСД чисел 252 та 105 (252 = 21 × 12; 105 = 21 × 5); оскільки 252 − 105 = 147, НСД 147 та 105 також 21. Оскільки більше з двох чисел постійно зменшується, повторне виконання цього кроку дає все менші числа, поки одне з них не дорівнюватиме нулю. Коли одне з чисел дорівнюватиме нулю, те, що залишилось, і є НСД. Обертаючи кроки алгоритму Евкліда у зворотний порядок, НСД можна виразити як лінійну комбінацію даних чисел помножених на цілі коефіцієнти, наприклад 21 = 5 × 105 + (−2) × 252.Ця важлива властивість відома як рівняння Безу.

Відповідно до алгоритму Евкліда (метод обчислення найбільшого спільного дільника) визначаємо більше значення для ключів *p* і *q* та записуємо співвідношення (нехай *p>q*) для першої пари чисел наступне співвідношення *p = b1*r + s*. В даній формулі *b* вибирається таким, щоб *b*r* було менше *q*, а *s* є доповненням до *p*. Якщо *s* не дорівнює 1, то *b1*

представляємо як $b1 = s*r + s_{n+1}$. Даний процес повторюється поки черговий залишок (s_n) не буде рівний 1.

На наступному етапі маючи значення s та s_{n+1} визначаємо значення коефіцієнтів для ключів p та q. При їх визначенні виконується зворотнє перетворення за формулою $1 = b1 - s_{n+1} * r$. Після цього дана різниця представляється у вигляді арифметичної суми $1 = t1*p + t2*q$. Коефіцієнти $t1$ і $t2$ використовуються для дешифрації за наступними співвідношеннями:

$$(t1 * b1 + t2 * d1) \bmod N;$$
$$(t1 * b2 + t2 * d2) \bmod N;$$
$$(t1 * b3 + t2 * d3) \bmod N;$$
$$(t1 * b4 + t2 * d4) \bmod N.$$

Один з цих розв'язків і буде дешифрованим значенням. У відповідності до описаного алгоритму розроблена узагальнена структура апаратної реалізації модуля дешифрації, яка приведена на рисунку

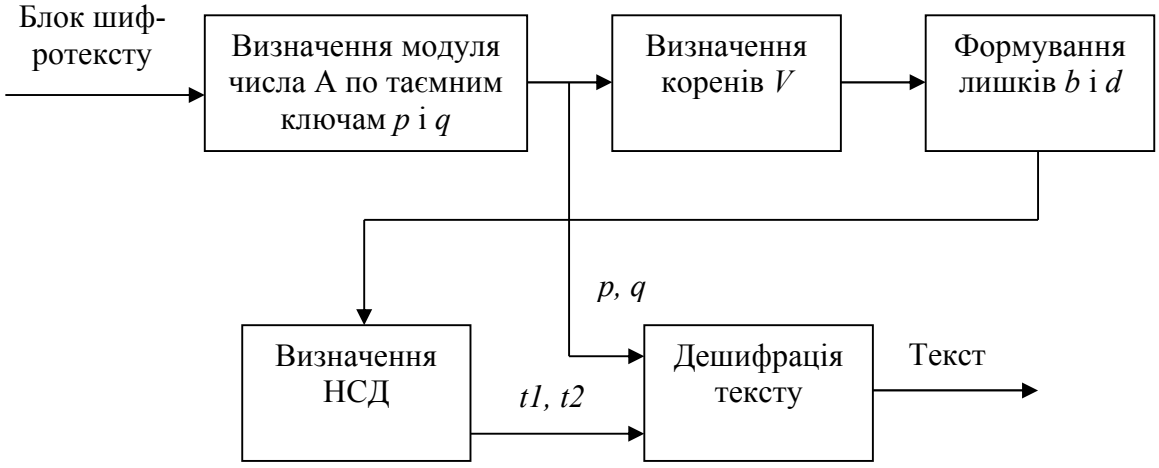

Рисунок 2.3 - Структурна схема модуля дешифрації

Таким чином, в даному розділі здійснено аналіз операцій перетворення за криптоалгоритмом Рабіна та розроблену узагальнену структурну схему модуля дешифрації.

2.3 Структури апаратної реалізації основних компонентів модуля дешифрації

Розроблена в попередньому розділі узагальнена структурна схема апаратного модуля дешифрації інформації за алгоритмом Рабіна не дає відповіді на питання про структуру окремих блоків та вузлів, що входять до неї. Тому в даному параграфі пропонується більш детально розглянути структури компонентів структурної схеми з метою підготовки до практичної реалізації.

Однією з основних операцій дешифрації, як випливає зі структурної схеми представленої на рисунку 2.3, є визначення модуля числа A по таємним ключам p і q. Зміст операції полягає у визначені залишку від операції ділення числа A на значення модуля по кожному з ключів p та q. Структурна схема визначення модуля числа приведена на рисунку 2.4.

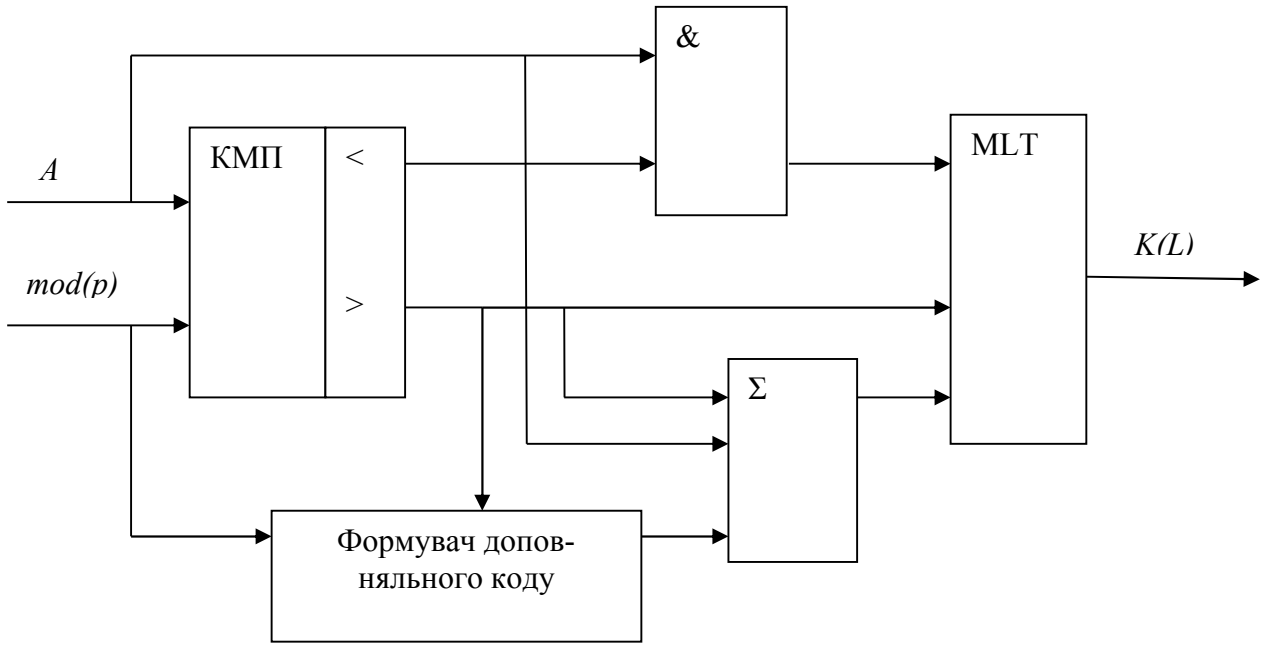

Рисунок 2.4 – Структурна схема визначення значення числа $A\bmod(p)$

Розробка структурної схеми здійснювалася виходячи з наступних міркувань. Число A за модулем може бути більше або менше значення модуля. Якщо число A менше значення модуля, то результат дорівнює A. Якщо число A більше за значення модуля, то операцію ділення можна замінити на операцію віднімання. Операція віднімання здійснюється за формулою $S = x + (\text{NOT}(y) + 1$. Як видно з формули операція віднімання передбачає сумування числа A з доповнюючим кодом модуля. Тому в склад структурної схеми входять компаратор КМП, завданням якого є порівняння числа А та значення модуля. Якщо число A менше значення модуля, то активний сигнал з'являється на виході < компаратора КМП. Цей сигнал дозволяє подачу A на вхід мультиплексора MLT. В протилежному випадку сигнал > на виході компаратора забезпечує перетворення значення модуля в доповнюючий код та виконання операції віднімання з використанням суматора Σ. Результат віднімання і буде значенням модуля числа A. За необхідності операцію після ускладнення структурної схеми модуля числа A можна виконувати в циклі до отримання значення меншого за модуль (за допомогою компаратора).

Наступною операцією є знаходження кореня $K\text{mod}(p)$. Апаратна реалізація операції знаходження кореня, як правило, базується на використанні пам'яті RAM. При цьому вважається, що підкореневий вираз вказує на адресу комірки в якій знаходиться значення кореня. Однак знаходження кореня за модулем здійснюється за складнішим алгоритмом. Зокрема, пошук кореня передбачає, що якщо корінь від K не береться ($K<\text{mod}(p)$), то до K додаємо p і шукаємо корінь квадратний. Якщо корінь не знаходиться, то до K додаємо $2p$ і пробуємо взяти корінь. Процес повторюється до взяття кореня.

Структурна схема визначення кореня за модулем приведена на рисунку 2.5. Відповідно до опису перетворень приведених в другому розділі, в її склад входять апаратні елементи, які забезпечують виконання описаних операцій. При цьому даний компонент складається з двох частин. Перша

частина, знаходження підкореневого виразу, передбачає знаходження такого числа від якого можна взяти корінь. Друга частина передбачає використання постійного запам'ятовуючого пристрою для знаходження значення кореню.

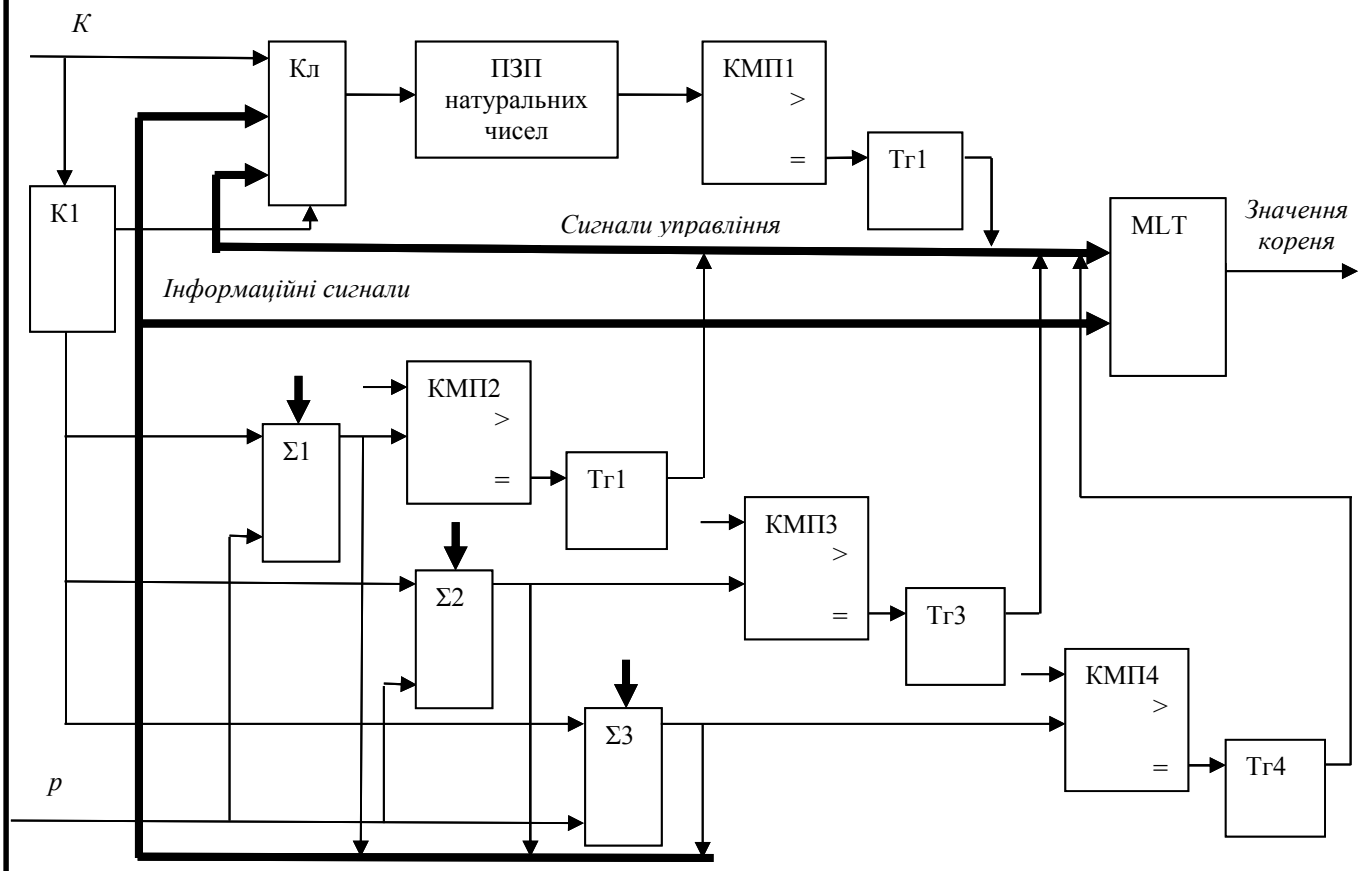

Рисунок 2.5 – Структурна схема визначення кореня числа $\sqrt{\text{К}}\,mod(p) = V$

Першим елементами структурної схеми на рисунку 2.5 є модуль знаходження підкореневого виразу, що містить компаратори К1, КМП1…КМП4, суматори Σ1…Σ3 та відповідні їм тригери. Наступним елементом є власне схема знаходження кореня, яка базується на таблиці 2.1.

Алгоритм роботи даної схеми полягає в порівнянні значень *K* та значення простого ключа, наприклад, *p*. Порівняння здійснюється за допомогою компаратора К1. Дане скорочення введено внаслідок просторових обмежень рисунку. Якщо *K>p*, то знаходження квадратного кореня числа К здійснюється по стандартній схемі з використанням ПЗП. Якщо ні, то управління передається на модуль знаходження підкореневого виразу.

Дана частина структурної схеми передбачає збільшення значення K на значення модуля p. Отриманий результат оцінюється за допомогою компараторів та фіксується в тригері. Стан тригера дозволяє подати результат суми на вхід ПЗП і отримати значення $\sqrt{\text{К}}\,mod(p) = V$. Як видно зі структурної схеми, передбачається збільшення K на три p, хоча можна передбачити і більше число раз. Стан тригерів формує сигнал управління для мультиплексора MLT і на вихід схеми поступає результат

Таким чином, в даному розділі розроблено структурні схеми знаходження $A\text{mod}(p)$, підкореневого виразу та визначення значення квадратного кореня $\sqrt{\text{К}}\,mod(p) = V$.

3 АПАРАТНА РЕАЛІЗАЦІЯ ТА ВЕРИФІКАЦІЯ КОМПОНЕНТІВ МОДУЛЯ ДЕШИФРАЦІЇ

3.1 Розробка схеми знаходження числа за модулем

Однією з основних операцій дешифрації інформації, як випливає зі структурної схеми представленої на рисунку 2.3, є визначення модуля числа A по закритим ключам p і q. Зміст операції полягає у використанні операції віднімання $S = x + (\text{NOT}(y) + 1$. Як видно з формули операція віднімання передбачає сумування числа A з доповняльним кодом значення модуля p.

Як видно з рисунку 2.4 основними елементами, що використовуються при апаратній реалізації даного компоненту є схеми цифрового компаратора, суматора та мультиплексор.

Апаратну реалізацію знаходження числа за модулем пропонується здійснити в імітаційному пакеті NI Multisim. Пакет прикладних программ NI Multisim є емулятором електронних схем, який дозволяє мінімізувати час розробки електричних схем цифрових пристроїв. Робота базується на основі технології віртуальних електронних приладів (SPICE-моделі) та аналізу функціонування електричних схем та їх тестуванні.

Інтерфейс користувача, характерний для NI Multisim, представлений на рисунку 3.1.

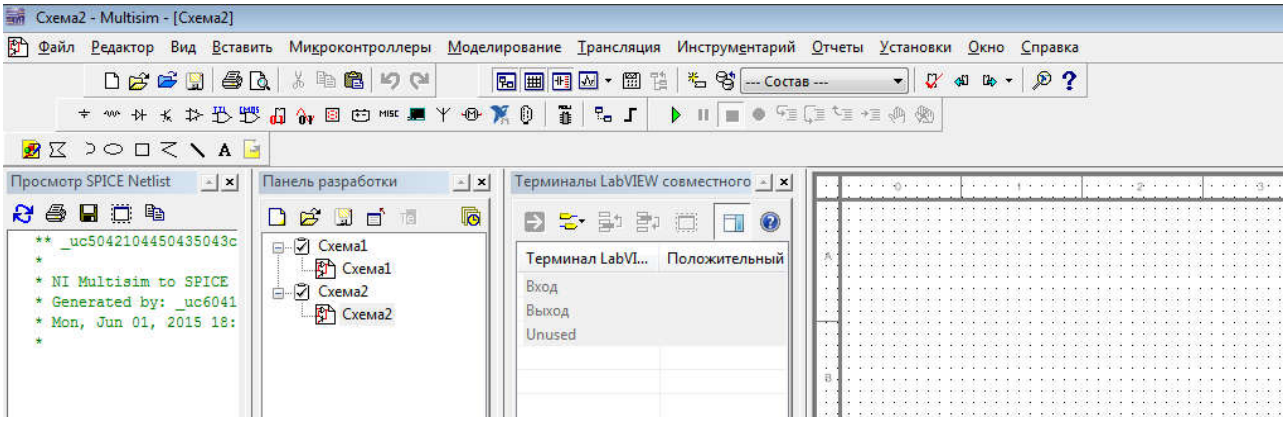

Рисунок 3.1 – Середовище NI Multisim

Типова структура містить головне меню, піктограми швидкого доступу, панелі компонентів та службові вікна, що надають розширені можливості для моделювання.

З бази даних моделей вибираємо мікросхему цифрового компаратора 7485N (аналог 530СП1) – рисунок 3.2. Робота мікросхеми 7485N базується на тому, що обидва порівнюваних слова A і B надходять на відповідні входи мікросхеми 7485. Молодші розряди подаються на входи A1 і B1, а старші - на входи A4 і B4. Якщо необхідно порівняти тільки 4-розрядні слова, то на вхід перенесення мікросхеми 7485 A = B (OAEQB - 6pin) подається напруга високого, а на входи перенесення A> B (5pin) і A <B (7pin) - низького рівня. Якщо обидва слова рівні за величиною, на виході A = B формується напруга високого рівня. Якщо слово A більше слова B, на виході A> B формується напруга високого рівня. Якщо слово A менше слова B, на виході A <B встановлюється напруга високого рівня. На інших виходах формується напруга низького рівня.

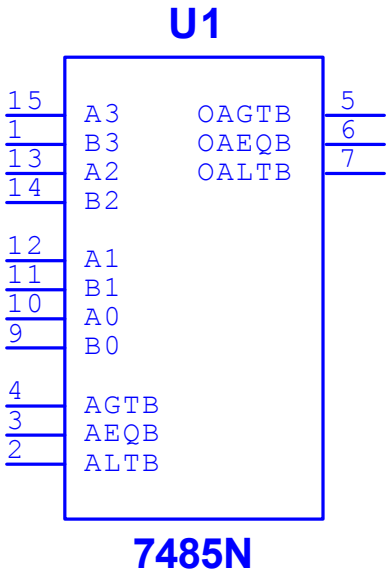

Рисунок 3.2 – Умовно-графічне позначення мікросхеми цифрового компаратора 7485N

Для збільшення розрядності цифрового компаратора на основі 7485N виходи першого ступеня 4-розрядного компаратора (молодші розряди)

з'єднуються з входами перенесення другого ступеня. В цьому випадку результат порівняння отримують на виходах мікросхеми старших розрядів – таблиця 3.1.

Таблиця 3.1 – Таблиця істинності мікросхеми 7485

Стани мікросхеми 7485									
Входи порівняння				Входи каскадного включення			Вихідн		
A3, B3	A2, B2	A1, B1	A0, B0	A>B	A<B	A=B	A>B	A<B	A=B
A3>B3	X	X	X	X	X	X	1	0	0
A3<B3	X	X	X	X	X	X	0	1	0
A3=B3	A>B2	X	X	X	X	X	1	0	0
A3=B3	A<B2	X	X	X	X	X	0	1	0
A3=B3	A2=B2	A1>B1	X	X	X	X	1	0	0
A3=B3	A2=B2	A1<B1	X	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0>B0	X	X	X	1	0	0
A3=B3	A2=B2	A1=B1	A0<B0	X	X	X	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	1	1	0	1	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	1	0	0	1	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	X	X	1	0	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	0	0	0	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	1	1	1	0
A3=B3	A2=B2	A1=B1	A0=B0	0	1	1	0	1	1
A3=B3	A2=B2	A1=B1	A0=B0	1	0	1	1	0	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	1	1	1	1
A3=B3	A2=B2	A1=B1	A0=B0	1	1	0	1	1	0
A3=B3	A2=B2	A1=B1	A0=B0	0	0	0	0	0	0

Наступним кроком буде задання таких вхідних кодів, щоб на виході компаратора отримати сигнали "менше, рівно, більше" відповідного до значень вхідних слів A та B. Для налаштування використаємо полосові

індикатори. Результуюча функційна електрична схема приведена на рисунку 3.4.

Задання двійкових слів A і B та формування команд здійснемо за допомогою компоненту – Interactive_Digital_Constant – рисунок 3.3.

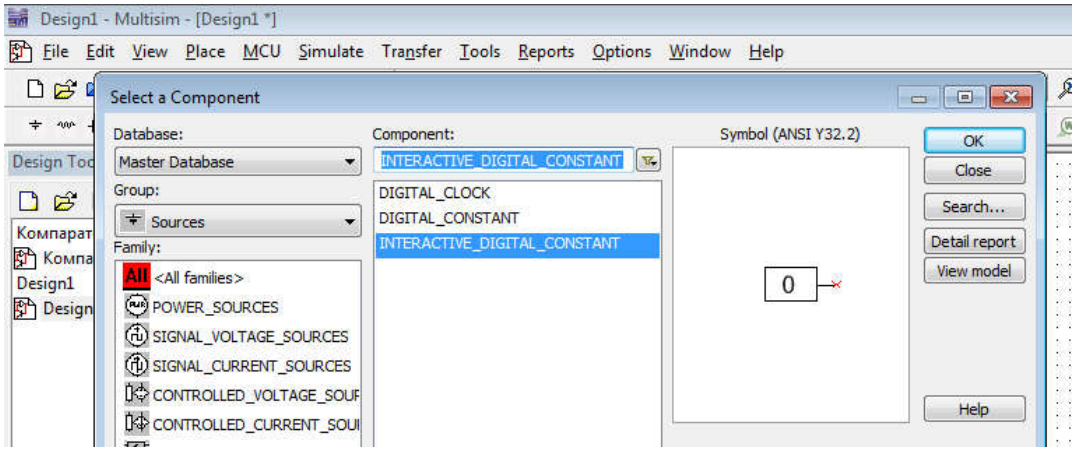

Рисунок 3.3 – Елемент Interactive_Digital_Constant

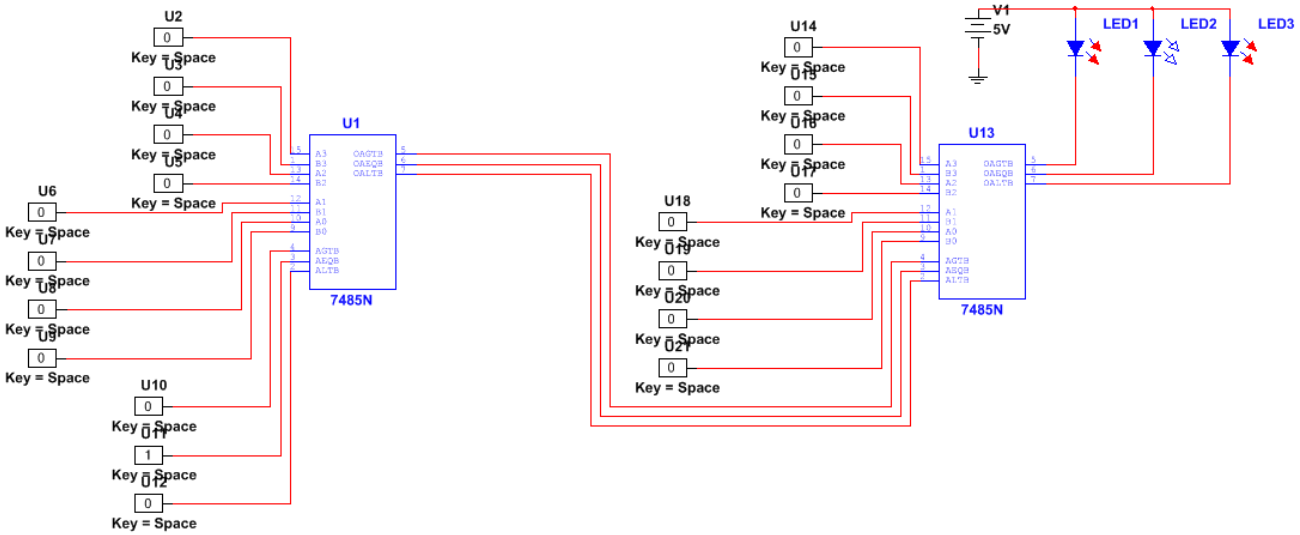

Рисунок 3.4 – Функційна електрична схема 8-ми розрядного цифрового компаратора

Для цього на входи A0-A3 та B0-B3 слід подати двійкові коди значення яких відповідатимуть співвідношенням менше, рівно та більше.

для цього виберемо три випадки:

01010000 та 01010000 - A=B;

1110000 та 10100000 - A>B;

10100000 та 11010000 – A<B.

Результати роботи компаратора спостерігатимемо на виходах OAGBT, OAEQB, OALTB.

Для першого випадку, коли A=B (рисунок 3.5) активний сигнал лог.1 (світло діод на горить) з'являється на виході OAEQB.

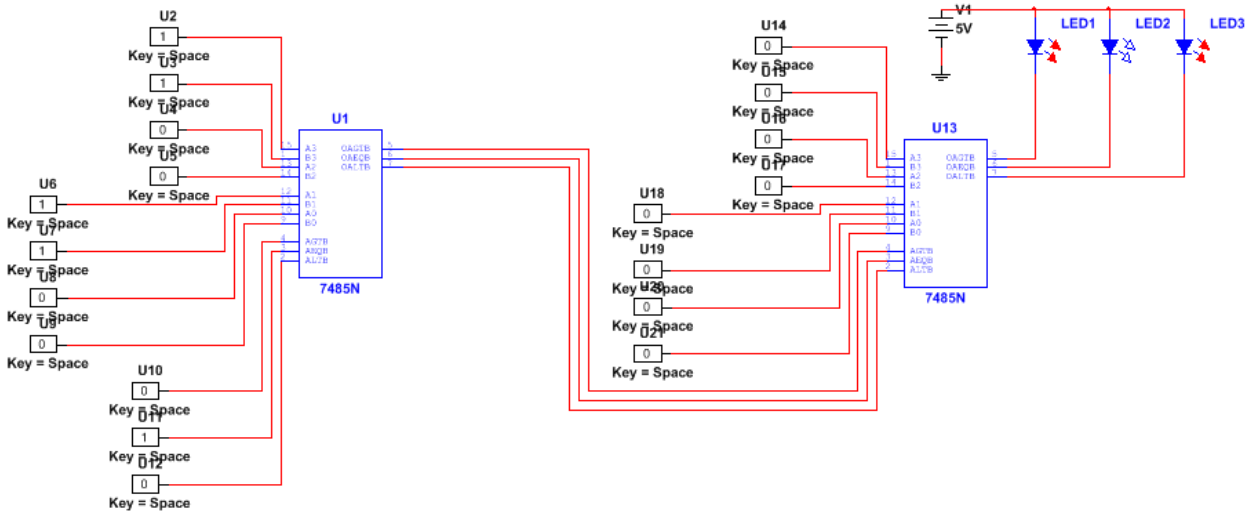

Рисунок 3.5 - Результати роботи компаратора для першого випадку, коли A=B

Для другого випадку, коли A>B (рисунок 3.6) активний сигнал лог.1 з'являється на виході OAGBT.

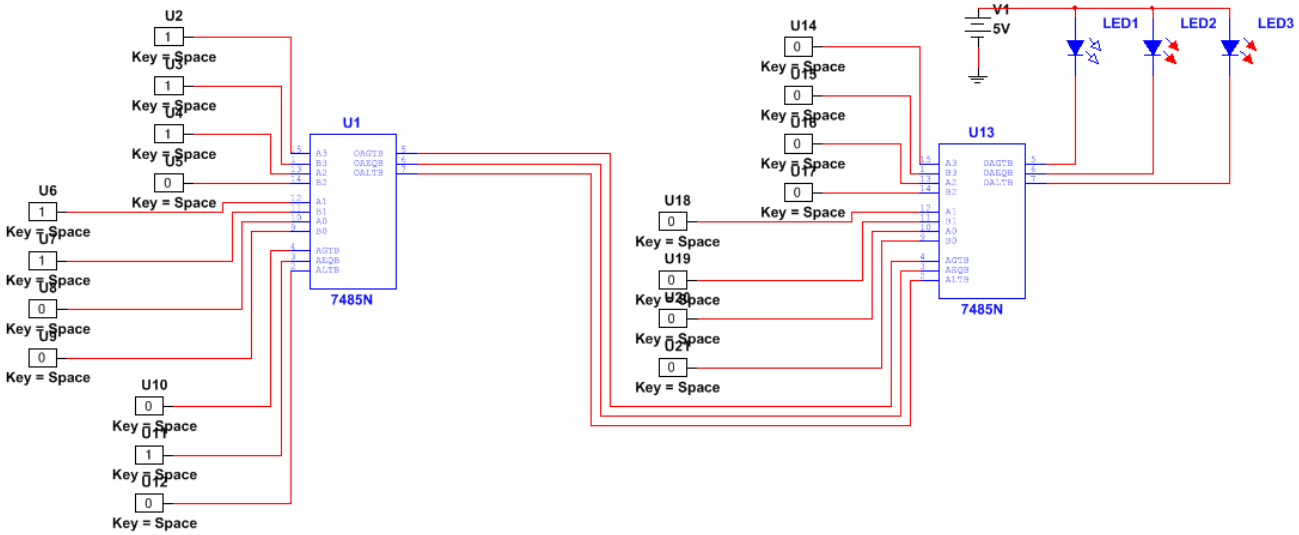

Рисунок 3.5 - Результати роботи компаратора для другого випадку, коли A>B

Для третього випадку, коли A<B (рисунок 3.7) активний сигнал лог.1 з'являється на виході OALTB.

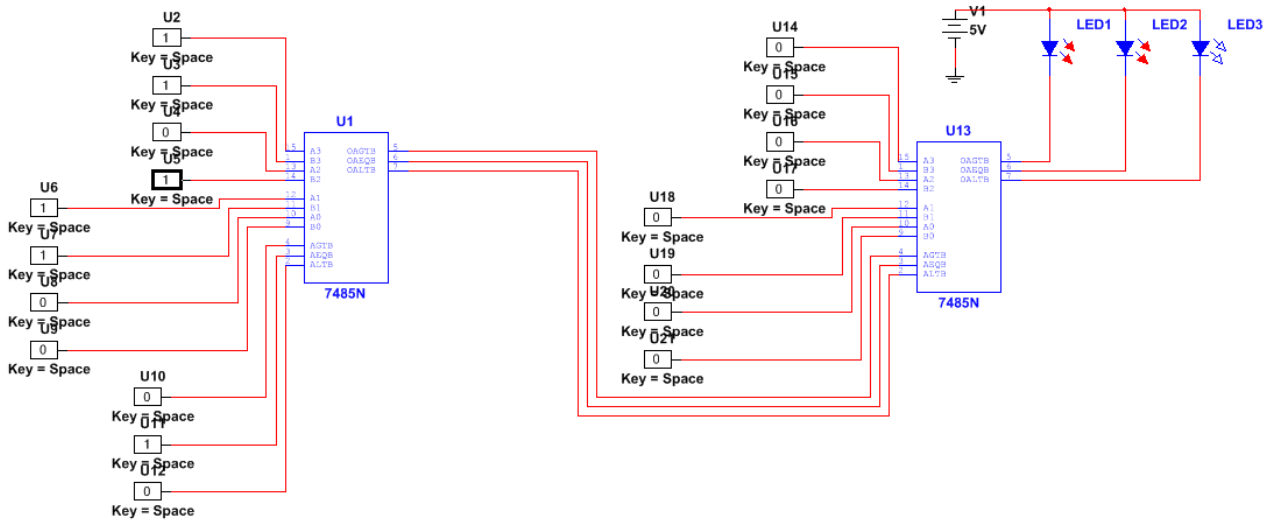

Рисунок 3.5 - Результати роботи 8-ми розрядного компаратора для третього випадку, коли A<B

Таким чином, розроблена та налаштована функційна схема цифрового компаратора дозволяє його легко масштабувати (збільшувати/зменшувати розрядність).

Наступним кроком буде розробка функційної електричної схеми формування доповняльного коду. Даний код реалізується шляхом інвертування коду таємного ключа, наприклад *p*, та додаванням до молодшого розряду логічної одиниці. Для реалізації даної схеми пропонується використати мікросхему суматора, приведену на рисунку 3.6.

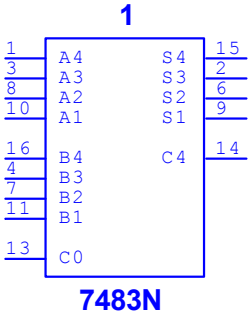

Рисунок 3.6 - Суматор 4-х розрядний серії 7483N

Схема формування доповняльного коду представлена на рисунку 3.6.

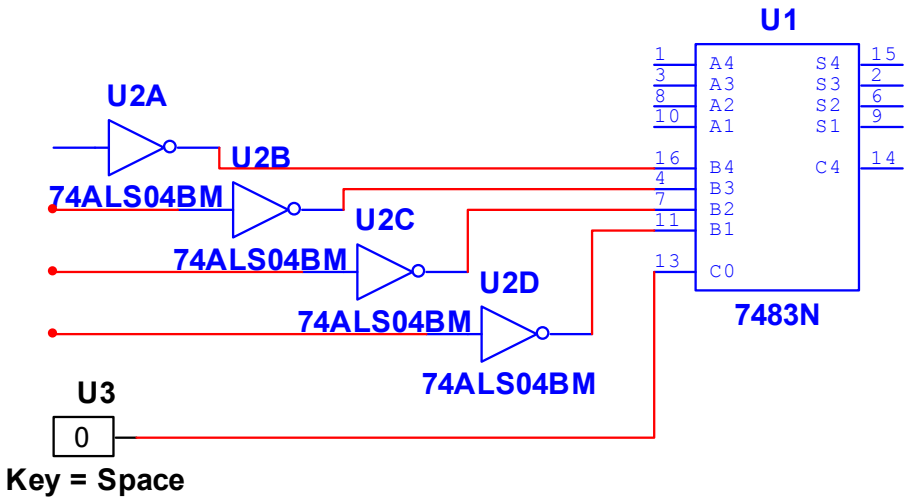

Рисунок 3.7- Схема віднімання числа A - B.

Деталізована функцій на електрична схема визначення модуля числа за таємним ключем приведена на на ДП.КСМ.07133/14.00.00.000Е1. Дана схема є синтезом електричних функцій них схем приведених в даному розділі.

Таким чином в даному розділі проведено аналіз електронних компонентів та обґрунтовано їх вибір для розробки електричної функційної схеми визначення числа за модулем таємного ключа.

3.2 Розробка схеми знаходження підкореневого виразу

Схема знаходження підкореневого виразу згідно алгоритму та структурної схеми 2.5 описаної в розділі 2.2 розробляється та використовується для дешифрації за алгоритмом Рабіна у випадку коли K менше p і квадратний корінь не можна взяти. Для цього випадку алгоритмом передбачено збільшення K на значення таємного ключа p. В структурній схемі для цієї цілі передбачено використання суматорів, цифрових

компараторів та елементів фіксації – тригерів. Умовно-графічне позначення D-тригера приведено на рисунку 2.8.

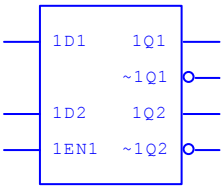

Рисунок 3.8 - Умовно-графічне позначення D-тригера 7475N

Мікросхеми 7475N є найпростішими засувками даних. Мікросхеми розташовані в 16-контактному корпусі і кожен її тригер має вихід даних Q. У тригері буде завантажена інформація, яка присутня на вході D, якщо стан входу E1 переключити від високого рівня до низького. Вихід Q знаходиться в поточному стані весь час, поки напруга на вході E1 залишається на низькому рівні. Закордонним аналогом мікросхем К155ТМ7, КМ155ТМ7 є мікросхеми SN7475N, SN7475J, які ми використаємо для апаратної реалізації даної схеми. Завданням тригера є сигналізація стану коли сума $K+p$ стане більше і можна взяти квадратний корінь.

Для уникнення випадків коли $K<<p$ в схемы передбачено масштабування до трьох операцій додавання p до змінної K.

Функціонування електричної функціональної схеми, приведеної на рисунку 3.9 відбувається наступним чином. Якщо на виході компаратора К1 згідно функціональної схеми приведеної на рисунку 2.5 встановиться сигнал, який відповідає стану, що $K<p$, то обчислювальний процес передається на підсхему яка передбачає знаходження такого значення $K+n*p$, яке при подачі на підсхему знаходження кореня забезпечить його знаходження. Схема реалізована наступним чином на вхід суматора U1, U2 подаються змінні K та p. Результат додавання порівнюється зі значенням таємного ключа, наприклад p. Якщо результат додавання буде знову менший за p, то цикл додавання повторюється в наступній декаді. Масштабування даної операції в принципі необмежена. Як правило достатньо до трьох циклів.

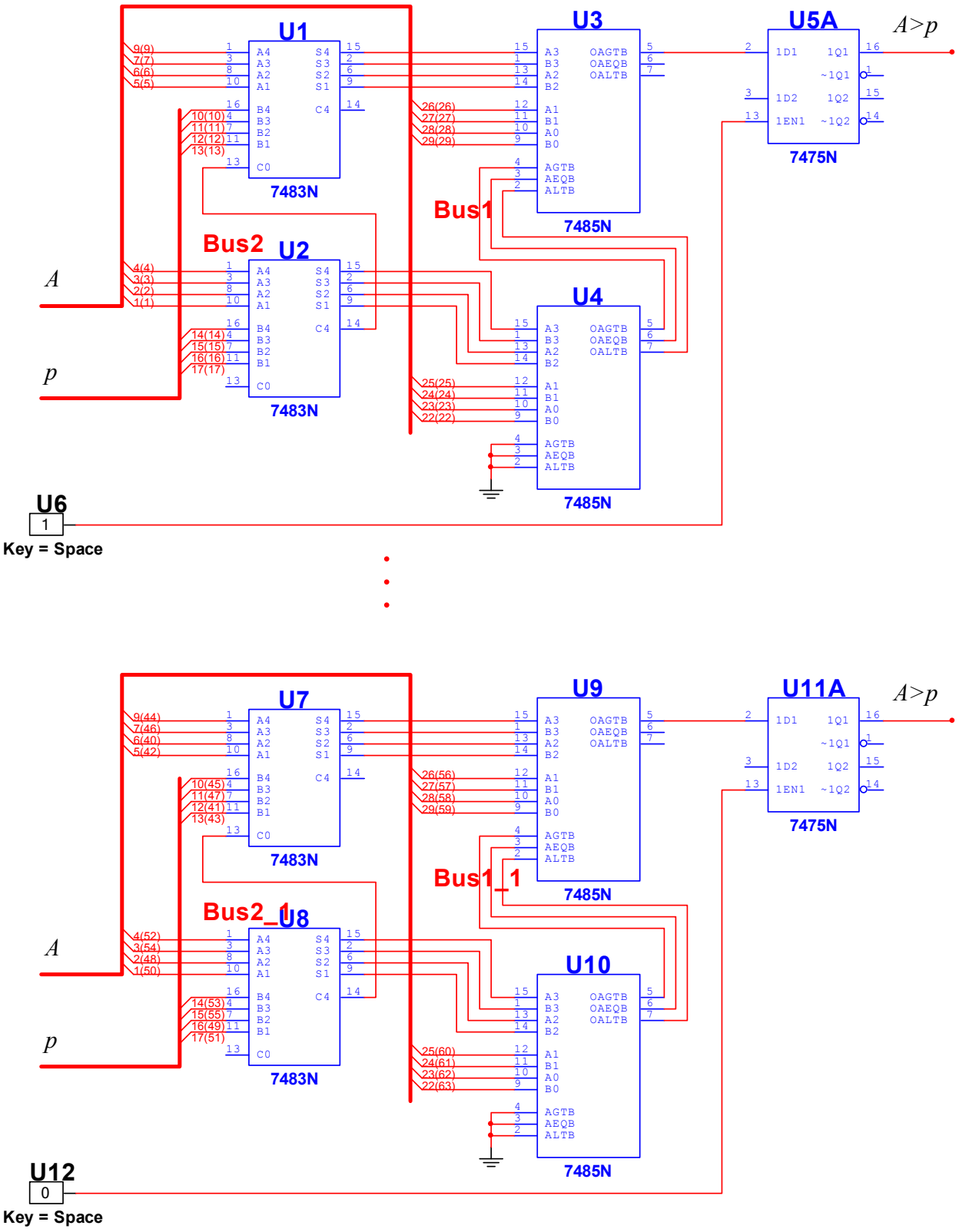

Рисунок 3.9 – Знаходження підкореневого виразу

Отримане значення поступає на підсхему знаходження корення, що передбачає використання постійної пам'яті. Дані питання розглядаються в наступному розділі.

3.3 Структура апаратної реалізації операції підняття до квадрату

Існують різні математичні методи отримання квадратного кореня числа. Серед них можна виділити алгоритми швидкого піднесення до ступеню (дріб) (дихотомічний алгоритм піднесення до степеня, бінарний алгоритм піднесення до степеня) - алгоритми, призначені для зведення числа *x* в натуральну ступінь *n* за менше число множень, ніж це потрібно в визначенні ступеня. Алгоритми засновані на тому, що для зведення числа *x* в ступінь *n* не обов'язково множити число *x* на саме себе n разів, а можна множити на вже обчислені дроби ступеню – значення кореня можна представити як дробове значення.

Як видно з приведеної інформації, основним недоліком даного виду алгоритмів є їх невисока швидкодія. Апаратно реалізація операції піднесення до степеня (знаходження квадратного кореня) може суттєво підвищити швидкодію. Реалізація даного підходу можлива двома шляхами - побудова комбінаційного багаторозрядного суматора, або за рахунок використання модулів напівпровідникової постійної пам’яті. Перший варіант (на основі суматора) передбачає реалізацію операцію двійкового множення, що вимагає ряд тактів. Це в умовах використання великих чисел вимагає значних витрат часу. Використання ПЗП є дуже швидкодіючим, піднесення до квадрату відбувається практично за один такт, одна потребує значних обсягів пам’яті.

На рисунку 2.5 приведена функціональна схема пристрою піднесення до квадрату на елементній базі постійного запам'ятовуючого пристрою. Апаратна реалізація даної математичної операції передбачає подачу на вхід постійного запам'ятовуючого пристрою значення *K+p*n*.

Базою для побудови даної схеми виступають мікросхеми пам'яті, які прошиваються у відповідністю з даними таблиці 2.1. Функційна електрична схема даного під модуля приведена на рисунку 3.10. В якості мікросхеми

пам'яті використано мікросхему оперативної пам'яті SPICE-модель якої є в пакеті NI Multisim.

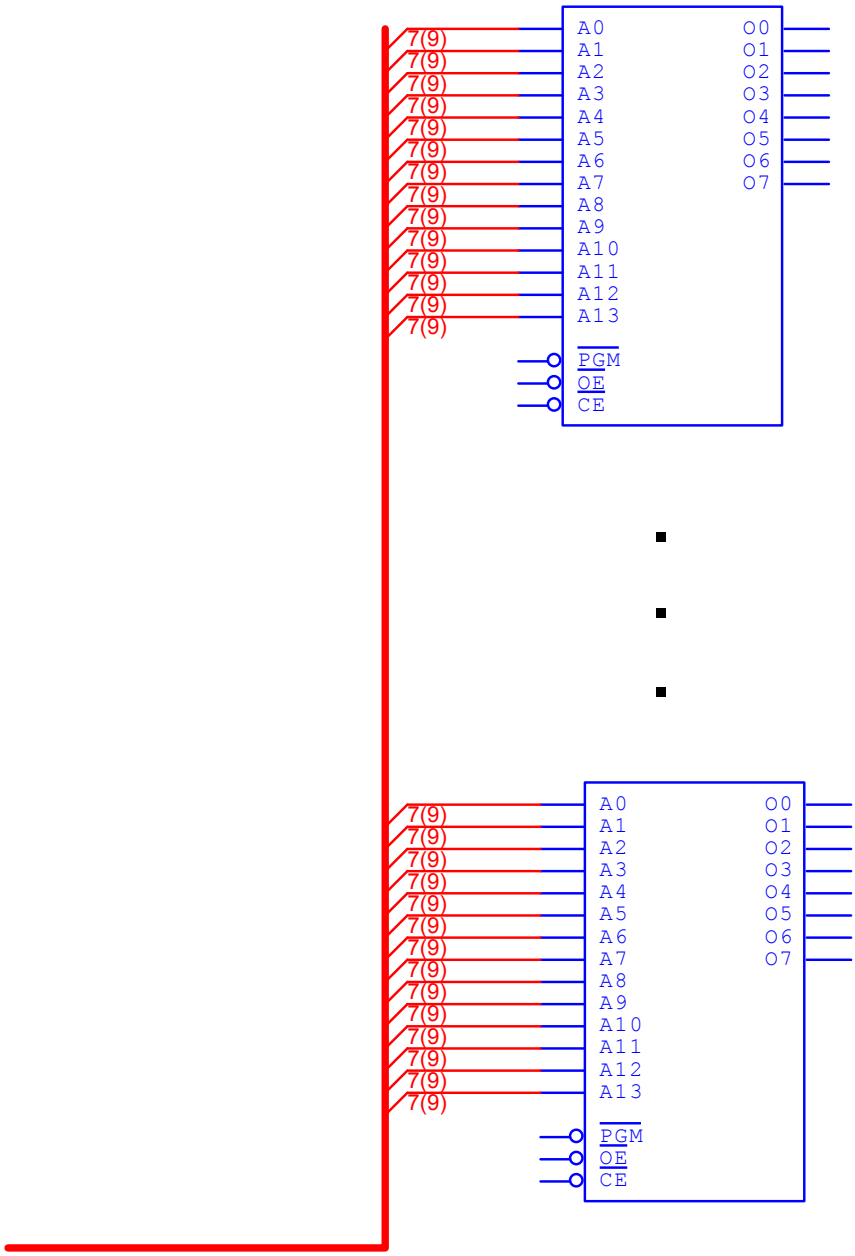

Рисунок 3.10 - Функціональна електрична схема знаходження кореня

Як видно з функціональної схеми, використання постійної пам'яті передбачає подання на вхід адреси числа, корінь якого міститься в даній комірці пам'яті.. Управляючі входи CE ініціалізує мікросхему, OE –

ініціалізує вихід комірок пам’яті, PGM – ініціалізує програмування вмісту комірок пам’яті.

Таким чином, в даному розділі здійснено апаратну реалізацію операції знаходження квадратного кореня.

4 ТЕХНІКО-ЕКОНОМІЧНИЙ РОЗДІЛ

Метою техніко – економічного розділу дипломного проекту є здійснення економічних розрахунків, спрямованих на визначення економічної доцільності розробки драйвера потужних світлодіодів. Потрібно визначити доцільність вибраного обладнання, провести розрахунок витрат на розробку даного проектного рішення, визначити прогнозовану ціну драйвера потужних світлодіодів, визначити показники економічної ефективності, зробити відповідні висновки.

4.1 Розрахунок капіталовкладень на розробку драйвера

При загальному підході до розрахунку капіталовкладень, які необхідні на розробку та впровадження компактного контролера керування живленням світлодіодів, можна записати:

$$K = K_{пр} + B_{пр} + B_{м} \quad (4.1)$$

де K – капіталовкладення на створення і впровадження;

$K_{пр}$ – витрати на виконання проектних робіт;

$B_{пр}$ – кошторисна вартість приладів та обладнання проектованого рішення;

Основними факторами при розрахунку витрат на виконання проектних робіт, що впливають на суму є: затрати часу на виконання проекту, необхідна кількість спеціалістів, їхня заробітна плата.

4.1.1 Розрахунок витрат на оплату праці

Витрати на оплату праці включають заробітну плату (ЗП) всіх категорій працівників, безпосередньо зайнятих на всіх етапах проектування. Розмір ЗП обчислюється на основі трудоємності відповідних робіт та середньої ЗП відповідних категорій працівників.

У розробці проектного рішення задіяні наступні спеціалісти - розробники, а саме: керівник проекту; студент-дипломник; консультант техніко-економічного розділу (таблиця 4.1).

Таблиця 4.1 - Вихідні дані для розрахунку витрат на оплату праці

Посада виконавців	Місячний оклад, грн.
Керівник ДП, викладач	6026
Консультант техніко-економічного розділу, доцент	6026
Студент	1100

Витрати на оплату праці розробників проекту визначаються за формулою (4.1):

$$B_{ОП} = \sum_{i=1}^{N} \sum_{j=1}^{M} n_{ij} \cdot t_{ij} \cdot C_{ij} , \quad (4.1)$$

де n_{ij} – чисельність розробників і-ої спеціальності j-го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником і-ої спеціальності j-го тарифного розряду, год;

C_{ij} – годинна ставка працівника і-ої спеціальності j-го тарифного розряду, грн.,

Годинну ставку працівника можна розрахувати за формулою:

$$C_{ij} = \frac{C_{ij}^{0}(1+h)}{РЧ_{i}}, \quad (4.2)$$

де C_{ij} – основна місячна заробітна плата розробника і-ої спеціальності j-го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$РЧ_i$ - місячний фонд робочого часу працівника і-ої спеціальності j-го тарифного розряду, год. (приймаємо 168 год.).

Коефіцієнт h, який визначає розмір додаткової заробітної плати, для керівника та консультанта техніко-економічного розділу дорівнює 1,47.

Середня годинна ставка керівника та консультанта техніко-економічного розділу ДП дорівнює:

$$C_{ij} = \frac{5470 \cdot (1 + 1{,}47)}{168} = 80{,}42 \frac{грн}{год}.$$

Середня годинна оплата студента дорівнює:

$$C_{ij} = \frac{1200}{168} = 7{,}14 \frac{грн}{год}$$

Витрати на оплату праці складають:

$$B_{ОП} = 20{,}5 \cdot 80{,}42 + 2 \cdot 80{,}42 + 144 \cdot 7{,}14 = 2837{,}45 \; грн.$$

Результати розрахунку записують до таблиці 4.2.

Таблиця 4.2 - Розрахунок витрат на оплату праці

Посада виконавців	Час розробки, год	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
Керівник ДП, доцент	16	80,42	1648,61

Консультант техніко-економічного розділу, доцент	2	80,42	160,84
Студент	144	7,14	1028
Разом			2837,45

4.1.2 Відрахування на соціальні заходи

Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства єдиний соціальний внесок складає 16,4% від суми заробітної плати:

$$B_{\Phi} = 0{,}164 \cdot B_{\text{ОП}}$$

$$B_{\Phi} = \frac{16{,}4}{100} \cdot 2837{,}45 = 465{,}34 \ \textit{грн}.$$

4.1.3 Розрахунок витрат на матеріали та комплектуючі

Загальна сума витрат на матеріальні ресурси (ВМ) визначається за формулою:

$$B_M = \sum_{i=1}^{n} K_i \cdot Ц_i \quad , \qquad (4.3)$$

де Кі - витрата і-го типу матеріалу, натуральні одиниці вимірювання;

Ці - ціна за одиницю і-го типу матеріалу, грн.;

і - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів

Таблиця 4.3 - Зведені розрахунки матеріальних витрат

Найменування матеріальних ресурсів	Од. виміру	Факт. витрачено матеріалів	Ціна за одиницю, грн.	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	2	3	4	5	6	7
Мікросхема AN9920	шт	1	26,2	26,2	2,62	28,82

1	2	3	4	5	6	7
Папір (формат А4)	уп	2	80	160	16	176
Діод	шт	5	10	20	2	22
Терморезистор	шт	1	77	77	7,7	84,7
Герконовий датчик	шт	2	35.1	70.2	7	77,2
Конденсатор	шт	4	24	96	9,6	105,6
Світлодіод	шт	10	1,5	15	1,5	16,5
Р а з о м						510,82

4.1.4 Витрати на використання комп’ютерної техніки

Витрати на використання комп’ютерної техніки складаються з витрат на амортизацію комп’ютерної техніки, витрат на користування програмним забезпеченням, витрат на електроенергію, що споживається комп’ютером. За даними обчислювального центру ТНЕУ для комп’ютера типу IBM PC/ATX вартість години роботи дорівнює 5,23 грн. Середній щоденний час роботи на комп’ютері – 2 години. Розрахунок витрат на використання комп’ютерної техніки приведений в таблиці 4.4.

Таблиця 4.4- Розрахунок витрат на використання комп'ютерної техніки

Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера грн.
Проведення досліджень та оформлення їх результатів	60	313,8
Оформлення техніко-економічного розділу	8	41,84
Оформлення ДП	12	62,76
Разом	80	418,4

Якщо для розробки КС купується і монтується спеціальне обладнання, то необхідно врахувати також витрати на доставку і монтаж. Ці витрати (в залежності від складності монтажу) можуть бути прийняті у розмірі 10-25% від витрат на придбання обладнання.

4.1.5 Накладні витрати

Накладні витрати проектних організацій включають три групи видатків: витрати на управління, загальногосподарські витрати, невиробничі витрати.

Вони розраховуються за встановленими відсотками до витрат на оплату праці.

Середньостатистичний відсоток накладних витрат приймемо 150% від заробітної плати:

$$H = 1{,}5 \cdot 2837{,}45 = 4256{,}17 \ грн.$$

4.1.6 Інші витрати

Інші витрати є витратами, які не враховані в попередніх статтях. Вони складають 10% від заробітної плати:

$$I = 2837{,}45 \cdot 0{,}1 = 283{,}75 \ грн \ .$$

Витрати на розробку проектного рішення дорівнюють:

$$К_{ПР}=В_{ОП} + В_{Ф} + В_{М} + В_{ЕЛ} + Н + І,$$

$$К_{ПР} = 2837{,}45+465{,}34+510{,}82+418{,}4+4256{,}17+283{,}75 = 8771{,}93 \ грн.$$

На підставі отриманих даних за окремими статтями складається кошторис витрат на розробку КС за формою, наведеною в таблиці 4.5.

Таблиця 4.5 - Кошторис витрат на розробку, відлагодження та дослідну експлуатацію КС

Статті витрат	Сума, *грн*.
1. Матеріальні витрати, в тому числі:	
матеріали	510,82
електроенергія	418,4
2.Витрати на оплату праці	2837,45
3.Відрахування на соціальні потреби	465,34
4.Накладні витрати	283,75
5. Інші витрати.	4256,17
РАЗОМ по кошторису	8771,93

4.2 Визначення прогнозованої ціни

Величина можливої (договірної) ціни КС повинна визначатися з урахуванням ефективності, якості і термінів її виконання на рівні, що

відповідає економічним інтересам замовника (споживача) і виконавця. Договірна ціна ($Ц_Д$) для прикладних КС розраховується за формулою:

$$Ц_{Д} = B_{КС} \cdot \left(1 + \frac{p}{100}\right), \quad (4.4)$$

де $B_{КС}$ – кошторисна вартість КС, *грн.*;

p - середній рівень рентабельності КС, % (приймається в розмірі 20-30% за погодженням з керівником).

$$Ц_Д = 8771{,}93 \cdot 1.3 = 11403{,}51 \text{ грн}.$$

4.2.1 Економічне обґрунтування вибору комплексу технічних і програмних засобів

Для впровадження більшості КС необхідно:

- ✓ придбання та встановлення засобів комп’ютерної техніки;
- ✓ придбання та інсталяція системного програмного забезпечення;
- ✓ інсталяція і адаптація спеціалізованого програмного забезпечення

Кожен з перерахованих пунктів допускає безліч різних варіантів, так як існує велика кількість конфігурацій комп'ютерів, обладнання та різноманітних програмних продуктів. Кожен з варіантів передбачає різні за величиною і структурою витрати.

4.3 Розрахунок зведених економічних показників

Економічна ефективність – це співвідношення між отриманим прибутком та затраченими коштами. Вона обчислюється за формулою (4.6):

$$E_ф = П_р/К_В \quad (4.6)$$

де $П_р$ – очікуваний прибуток ;

$К_В$ – кошторисна вартість.

Очікуваний прибуток можна розрахувати із співвідношення:

$$П_Р = Ц_Д – К_В.$$

$$П_Р = 11403,51 – 8771,93 = 2631,58 грн.$$

Після проведених розрахунків отримуємо:

$$E_ф = 2631,58/8771,93 = 0.3$$

Термін окупності додаткових капітальних вкладень визначається як :

$$T = 1/E_ф = 1/0.3 = 3.3 роки. \quad (4.7)$$

Таблиця 4.6 - Зведені економічні показники розробки

Показник	Значення
Собівартість, грн.	8771,93
Плановий прибуток, грн.	2631,58
Ціна, грн.	11403,51
Економічна ефективність	0,3
Термін окупності, рік	3,3

Провівши аналіз розрахованих значень економічних показників робимо висновок, що розробка драйвера потужних світлодіодів із регулюванням сили струму є економічно доцільною.

ВИСНОВКИ

В результаті виконання дипломного проекту на тему "Апаратний модуль дешифрування криптоалгоритму Рабіна" отримано наступні результати:

1. Проаналізовано алгоритм дешифрації по криптосистемі Рабіна. У відповідності до алгоритму розроблена узагальнена структура апаратної реалізації модуля дешифрації, яка складається з блоку визначення модуля числа А по таємним ключам p і q, блоку визначення коренів *V,* блоку формування лишків *b* і *d*, блоку визначення НСД, блоку дешифрація тексту.

2. Розроблено структурні схеми знаходження $A\mathrm{mod}(p)$, підкореневого виразу та визначення значення квадратного кореня $\sqrt{\mathrm{K}}\,mod(p) = V$.

3. Апаратну реалізацію знаходження числа за модулем здійснено в імітаційному пакеті NI Multisim. Пакет прикладних программ NI Multisim є емулятором електронних схем, який дозволяє мінімізувати час розробки електричних схем цифрових пристроїв. Робота базується на основі технології віртуальних електронних приладів (SPICE-моделі) та аналізу функціонування електричних схем та їх тестуванні.

4. Розроблено електричну схему визначення модуля числа за таємним ключем, схему знаходження підкореневого виразу, апаратну реалізацію операції підняття до квадрату

.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України "Про захист інформації в інформаційно-телекомунікаційних системах" - [Електронний ресурс]- Режим доступу: http://zakon2.rada.gov.ua/laws/show/80/94-%D0%B2%D1%80 - Заголовок з екрану.

2. Грайворонський М. В. Безпека інформаційно-комунікаційних систем / М. В. Грайворонський, О. М. Новіков. – К. : Видавнича група BHV, 2009. – 608 с.

3. Ємець В. Сучасна криптографія. Основні поняття / В. Ємець, А. Мельник, Р. Попович. – Львів : Бак, 2003. – 144 с.

4. Основы информационной безопасности. Учебное пособие для вузов / Е.Б. Белов, В.П. Лось, Р.В. Мещеряков, А.А. Шелупанов - М .: Горячая линия - Телеком, 2006-544 с.

5. Основи інформаційної безпеки / С. В. Кавун, О. А. Смірнов, В. Ф. Столбов – Кіровоград : Вид. КНТУ, 2012. – 414 с.

6. Хорев А.А. Защита информации от утечки по техническим каналам. Технические каналы утечки информации. М.: Гостехкомиссия РФ, 1998. 320с.

7. Коробейников А. Г. Математические основы криптологии : учебн. пособ. / А. Г. Коробейников, Ю. А. Гатчин. – СПб. : СПб ГУ ИТМО, 2004. – 106 с.

8. Кузнецов О. О. Захист інформації в інформаційних системах / О. О. Кузнецов, С. П. Євсеєв, О. Г. Король. – Х. : Вид. ХНЕУ, 2011. – 512 с.

9. Остапов С. Е. Основи криптографії / С. Е. Остапов, Л. О. Валь. – Чернівці : Книги ХХІ, 2008. – 188 с. 34. Поповский В. В. Защита информации в телекоммуникационных системах : учебник / В. В. Поповский, А. В. Персиков. – Х. : ООО "Ком- пания СМИТ", 2006. – Т. 1. – 292 с.

10. Устройства умноження и деления - Режим доступу: http://naf-st.ru/arti. - Заголовок з екрану

11. Бабич М.П., Жуков І.А. Комп'ютерна схемотехніка: Навчальний посібник.- К.:МК-Прес, 2004.-412с

12. Схемотехніка електронних систем. Цифрова схемотехніка. Підручник / В.І. Бойко, А.М. Гуржій, В.Я Жуйков та ін.-К.:Вища школа, 2004.-423с.

13. Прянишников В.А. Электроника: Полный курс лекцій. – СПб.Корона принт; М.: Бином – Пресс, 2006.- 416.

14. Терехин В.Б., Соловьев Ю.А. Моделирование электронных схем в программе Electronics Workbench. Ч. 1. Создание схем. Ч.2. Элементная база: лабораторный практикум. – Северск: СТИ ТПУ, 2000. – 244 с.

15. Шило В.Л. Популярные цифровые микросхемы: Справочник. 2 - е изд., испр. – Челябинск: Металлургия, Челябинское отд., 1989.- 352 с.

16. Лещенко М.Є. Основи мікроелектроніки / М.Є. Лещенко, В.Є. Овчаренко. – Х. : Нац. аерокосм. ун-т „Харк. авіац. ін-т", 2005.

17. Комп`ютерна електроніка: Навч. посібник. Частина I/II А.П.Оксанич, С.Е.Притчин, О.В.Вашерук.- Харків: "Компанія СМІТ", 2006.- 200с/256с.

18. Рябенький В.М., Жуйков В.Я., Гулий В.Д. Цифрова схемотехніка: Навч. Посібник. - Львів: Видавництво «Новий світ 2000», 2009.-736с.

19. Резисторы, конденсаторы, трансформаторы, дроссели, комутационные устройства РЭА: Справочник./ Н.Н. Акимов, Е.П. Ващуков, В.А. Прохоренко, Ю.П. Ходоренок. – Мн.:Беларусь, 1994.- 591с.

20. Токхейм Р. Основы цифровой электроники.- М.:Мир, 1989.

21. Устройства умножения и деления [Електронний ресурс] – Режим доступу - http://naf-st.ru/articles/digit/multidev/

22. Никитин В.А. Схемотехника интегральных схем ТТЛ, ТТЛШ и КМОП: Учебное пособие. М.: НИЯУ МИФИ, 2010. – 64 с.

23. Методичні рекомендації до виконання дипломного проекту з освітньо-кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп'ютерна інженерія» фахового спрямування «Комп'ютерні системи та мережі» / О.М.Березький, Л.О.Дубчак, Р.Б.Трембач, Г.М.Мельник, Ю.М.Батько, С.В.Івасьєв / Під ред. О.М.Березького. – Тернопіль: ТНЕУ, 2016. – 65с.

24. Методичні вказівки до написання техніко-економічного розділу для дипломних проектів на здобуття освітньо - кваліфікаційного рівня «Бакалавр» напряму підготовки 6.050102 «Комп'ютерна інженерія» / І.Р.Паздрій. - Тернопіль: ТНЕУ, 2018. – 36с.