

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”

Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор

Олексій СМІРНОВ

“ ___ ” _____ 2022 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему

**“Дослідження та програмна реалізація системи виявлення
несправності елементів цифрових пристроїв”**

Виконав здобувач вищої освіти
II курсу, групи КІ-21М-1,4
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Дзюбинський О.В.
« ___ » _____ 2022 р.

Керівник проекту
кандидат технічних наук
_____ Смірнова Т.В.
« ___ » _____ 2022 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 “Інформаційні технології”
Спеціальність 123 “Комп’ютерна інженерія”
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2022 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Дзюбинському Олексію Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв

2. Керівник роботи Смірнова Тетяна Віталіївна, канд. техн. наук

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 19-13 від 17.08.2022 року

3. Строк подання студентом роботи до захисту 10.12.2022 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2022	14.11.2022
Охорона праці	Оришака О.В.	06.10.2022	16.11.2022

7. Дата видачі завдання « 6 » вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2022 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2022 р.	
3.	Розробка моделі компонента	20.10.2022 р.	
4.	Розробка структур даних	25.10.2022 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2022 р.	
6.	Програмування алгоритмів	10.11.2022 р.	
7.	Розрахунок економічної ефективності	13.11.2022 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2022 р.	
9.	Оформлення ПЗ	17.11.2022 р.	
10.	Попередній захист роботи	10.12.2022 р.	

Дата видачі завдання
« 6 » вересня 2022 р.

Підпис керівника

Смірнова Т.В.
(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2022 р.

Підпис здобувача

Дзюбинський О.В.
(прізвище та ініціали)

АНОТАЦІЯ

Дзюбинський О.В. Дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2022.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи виявлення несправності елементів цифрових пристроїв.

Метою розробки є дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

Об'єктом дослідження є процес виявлення несправності елементів цифрових пристроїв.

Предметом дослідження є методи виявлення несправності елементів цифрових пристроїв.

Методи дослідження базуються на методах системотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі RAD Studio Delphi 10.4.

Ключові слова: комп'ютерна інженерія, виявлення несправності, цифрові пристрої

ABSTRACT

Dziubynskyi O.V. Research and software implementation of the system for detecting malfunctions of elements of digital devices. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2022.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the system of detecting malfunctions of elements of digital devices.

The goal of the development is the research and software implementation of the system for detecting malfunctions of elements of digital devices.

The object of the study is the process of detecting malfunctions of elements of digital devices.

The subject of research is the methods of detecting malfunctions of elements of digital devices.

Research methods are based on systems engineering methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the system for detecting malfunctions of elements of digital devices.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the RAD Studio Delphi 10.4 environment.

Keywords: computer engineering, fault detection, digital devices

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	9
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	11
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	11
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	26
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	28
3.1 Опис функціонування системи	28
3.2 Розробка структурної схеми.....	37
3.3 Розробка функціональної схеми	39
3.4 Розробка діаграми процесів.....	41
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	43
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	43
4.2 Захист розробленого програмного забезпечення.....	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	59
6 НАУКОВА НОВИЗНА	65

					ВКРМ-123.22.0007.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв</i>	Лім.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Цзюбинський О.В.</i>					М	1	105
<i>Перев.</i>	<i>Смірнова Т.В.</i>					<i>ЦНТУ КІ-21М-1,4</i>		
Н.контр.	<i>Гермак В.С.</i>							
Затв.	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	67
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	67
7.2 Розрахунок трудомісткості розробки програмної продукції.....	69
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	71
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	76
7.5 Визначення собівартості розробки та ціни програмної продукції.....	80
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	81
7.7 Визначення експлуатаційних витрат.....	82
7.8 Визначення економічної ефективності програмної продукції.....	83
7.9 Висновок.....	85
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	86
8.1 Вступ.....	86
8.2 Пожежна безпека.....	86
8.3 Аналіз умов праці програміста	88
8.4 Розробка заходів з умов поліпшення охорони праці.....	92
8.5 Розрахункова частина	93
8.6 Висновки до розділу.....	95
9 ОСНОВНІ ВИСНОВКИ.....	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ГПВЧ	–	генератор псевдовипадкових чисел
ЕОМ	–	електронно-обчислювальна машина
МС	–	метод синтезу
ПВ	–	псевдовипадковий
ПВЧП	–	псевдовипадкова числова послідовність
ЦС	–	цифрова схема

Кафедра _ КБПЗ _ 2022 рік

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Неухильний ріст складності приладів спричиняється підвищений інтерес до питань діагностування їхнього технічного стану. Одним з різновидів методів технічного діагностування апаратури є тестова діагностика, що дозволяє на етапі проектування й виготовлення вирішувати основні задачі:

- визначати правильність функціонування;
- здійснювати пошук несправностей;
- визначати тип несправності.

Для реалізації цих задач потрібна інтенсифікація підготовки фахівців з обчислювальної техніки й технічної діагностики, що володіють методикою дослідження й проектування складних цифрових систем з використанням сучасних методів технічної діагностики.

Основною задачею магістерської роботи є розробка програмного забезпечення виявлення несправності елементів цифрових пристроїв, що дозволяє детально знайомити студентів із практичними можливостями використання сучасних методів компактного тестування.

Вона повинна являти собою програму, що включає в себе:

- модуль, що реалізує графічний інтерфейс. Обмін графічною інформацією між користувачем і ЕОМ повинне здійснюватися у формі діалогу;
- модуль, що реалізує логічне моделювання цифрових схем;
- модуль, що моделює роботу генераторів тестових послідовностей;
- блок, що моделює процес діагностики. У нього входить: блок моделює роботу багатоканального сигнатурного аналізатора, блок відображення й обробки отриманих даних, блок пошуку несправностей;
- блок, що реалізує алгоритм визначення оцінки ефективності діагностики при використанні компактних методів діагностики.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

– Огляд існуючих систем виявлення несправності елементів цифрових пристроїв.

– Дослідження системи виявлення несправності елементів цифрових пристроїв.

– Програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

Об'єктом дослідження є процес виявлення несправності елементів цифрових пристроїв.

Предметом дослідження є методи виявлення несправності елементів цифрових пристроїв.

Методи дослідження базуються на методах системотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод виявлення несправності елементів цифрових пристроїв.

– Розроблено вітчизняний продукт виявлення несправності елементів цифрових пристроїв, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі виявлення несправності елементів цифрових пристроїв.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVI Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2022, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №13.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Широке поширення радіоелектронних пристроїв із застосуванням цифрової обробки сигналів спричиняє підвищений інтерес до питань діагностування їхнього технічного стану. Одним з різновидів діагностування цифрових вузлів і блоків є тестове діагностування, застосування якого на етапі проектування й виготовлення цифрових вузлів дозволяє визначити правильність їхнього функціонування й здійснити процедуру пошуку несправностей. При розробці тестової діагностики виникає складність у визначенні еталонних реакцій при тестуванні існуючих схем, у визначенні оптимального числа контрольних точок для зняття вихідної реакції діагностуваної цифрової схеми. Це можна зробити або створюючи прототип розроблювального цифрового пристрою й проводячи його діагностику апаратурними методами, або здійснюючи моделювання на ЕОМ як цифрового пристрою, так і процесу діагностики. Найбільш раціональним є другий підхід, що припускає створення автоматизованих систем діагностики [1], що дозволяють робити діагностику цифрових схем на стадії проектування й здатних вирішувати наступні задачі:

1. Робити логічне моделювання цифрових схем за допомогою ЕОМ. Ціль логічного моделювання полягає в тому, щоб виконати функцію проектованої схеми без її фізичної реалізації. Перевірка на правильність моделювання може бути різної залежно від рівня подання цифрової схеми в ЕОМ. Якщо, наприклад, здійснюється перевірка тільки значень логічної функції на виході схеми, то досить представити схему на рівні логічних елементів. Для того щоб перевірити стану сигналів у схемі, необхідно точно описати затримки спрацьовування всіх елементів в умовах синхронізації.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2. Моделювання несправностей. Задача виявлення несправностей у цифрових схемах полягає в тому, щоб визначити, чи володіє цифрова схема необхідним поведженням. Для рішення цієї задачі необхідно, насамперед, установити модель цифрової схеми як об'єкта контролю, потім метод виявлення несправностей і, нарешті, модель несправностей. З погляду особливостей поведження цифрових схем їх можна розділити на комбінаційні й послідовні. Відносно виявлення несправностей комбінаційні схеми є порівняно простою моделлю. Послідовні схеми відносно поведження характеризуються наявністю внутрішніх контурів зворотного зв'язка, тому виявлення несправностей у них у загальному випадку надзвичайно утруднено.

3. Моделювання процесу тестової діагностики. Класична стратегія тестування цифрових схем заснована на формуванні тестових послідовностей, що дозволяють виявляти задані множини їхніх несправностей. Для реалізації генератора тестової послідовності бажано використовувати найпростіші методи, що дозволяють уникнути складної процедури їхнього синтезу. До них відносяться наступні алгоритми:

- формування всіляких вхідних тестових наборів, тобто повного перебору двійкових комбінацій. У результаті застосування подібного алгоритму генеруються рахункові послідовності;

- формування випадкових тестових наборів з необхідними ймовірностями одиничного й нульового символів по кожному вході цифрової схеми;

- формування псевдовипадкових тестових послідовностей. Основною властивістю розповсюджених алгоритмів формування тестових послідовностей є те, що в результаті їхнього застосування відтворюються послідовності дуже великої довжини. Тому на виходах цифрової схеми, що перевіряється, формуються її реакції, що мають ту ж довжину. Природно виникають проблеми їхнього запам'ятовування й зберігання. Найпростішим рішенням, що дозволяє значно скоротити об'єм збереженої інформації про еталонні вихідні реакції, є

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

одержання інтегральних оцінок, що мають меншу розмірність. Для цього використовуються алгоритми стиску інформації.

Для того щоб застосовувати метод компактного стиску тестування, необхідно раціонально вибрати алгоритм формування тестових послідовностей і метод стиску інформації [2]. Для діагностики будь-якої комбінаційної схеми особливий інтерес представляє сигнатурний аналізатор, зокрема багатоканальний сигнатурний аналізатор, в основі побудови якого лежить алгоритм стиску інформації – сигнатурний аналіз.

4. Аналіз методів оцінки ефективності компактного тестування. Побудова складних цифрових пристроїв вимагає підвищеної уваги до компактних методів тестування для кожного конкретного застосування. Тому виникає необхідність в оцінці ефективності того або іншого методу компактного тестування. У цей час у літературі розглядаються способи порівняння методів компактного тестування. Було запропоновано розробити моделюючий алгоритм, що дозволяє будувати багатоканальні сигнатурні аналізатори. Для цього необхідно було вирішити наступні задачі:

- Логічне моделювання цифрових схем.
- Розробка моделюючого алгоритму побудови ГПВЧ.
- Розробка моделюючого алгоритму побудови багатоканального сигнатурного аналізатора.
- Оцінка ефективності роботи багатоканального сигнатурного аналізатора.
- Розробка алгоритму пошуку несправностей

1.2 Область застосування

Якість контролю й діагностики залежить не тільки від технічних характеристик контрольно-діагностуючої апаратури, але й у першу чергу від тестопридатності (контролюємості) самого випробовуваного виробу. Це означає,

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

що якість перевірки багато в чому визначається якістю розробки виробів. Найпростіше рішення підвищення якості контролю – це виведення деяких внутрішніх точок виробу на зовнішнє рознімання. Однак число вільних контактів на розніманні обмежено, тому зазначений підхід рідко виявляється доступним або досить ефективним. Більш прийнятне рішення пов'язане з розміщенням на платі додаткових функціональних елементів, призначених для безпосереднього одержання або нагромадження інформації про стан внутрішніх точок і наступної її передачі на обробку на вимогу пристрою, що аналізує (зовнішнього або також вбудованого).

Сигнали, що виникають у процесі функціонування основної й контрольної апаратури, розміщеної разом на одному друкованому модулі або кристалі інтегральної схеми, зіставляються за певними правилами. У результаті такого зіставлення виробляється інформація про правильне функціонування контрольованого вузла. Як надлишкова апаратура може бути використана повна копія вузла, що перевіряється. При цьому виробляється найпростіше порівняння двох однакових наборів кодів. З метою зменшення об'єму додаткової контрольної апаратури використовують більш прості контрольні пристрої з надлишковим кодуванням, але зате при цьому ускладнюються способи одержання контрольних співвідношень.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Класифікація методів стиску вихідних реакцій схем

Класична стратегія тестування цифрових схем заснована на формуванні тестових послідовностей, що дозволяють виявляти задані множини їхніх несправностей. При цьому для проведення процедури тестування, як правило, зберігаються як самі послідовності, так і еталонні вихідні реакції схем на їхній вплив. У процесі самої процедури тестування на підставі порівняння вихідних реакцій з еталонними приймається рішення про стан схеми, що перевіряється .

Для ряду схем, що випускаються в сучасний час, класичний підхід вимагає тимчасових витрат як на формування тестових послідовностей, так і на процедуру тестування. Крім того на проведення тестового експерименту потрібна наявність складного встаткування. У зв'язку із цим вартість і час, необхідні для реалізації класичного підходу, ростуть швидше, ніж складність цифрових схем, для яких він використовується. Тому нові рішення, що дозволяють значно спростити як процедуру побудови генераторів тестових послідовностей, так і проведення тестового експерименту.

Для реалізації генератора тестової послідовності використовуються алгоритми, що дозволяють уникнути складності їхнього синтезу:

1. Формування всіляких тестових наборів, тобто повного перебору двійкових комбінацій. У результаті генерується так звана рахункова послідовність.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

2. Формування випадкових тестових наборів з необхідними ймовірностями появи одиничного й нульового символів по кожному вході цифрової схеми.

3. Формування псевдовипадкової тестової послідовності.

Основною властивістю розглянутих алгоритмів формування тестових послідовностей є те, що в результаті їхнього застосування відтворюються послідовності дуже великої довжини. Тому на виходах цифрової схеми, що перевіряється, формуються її реакції, що мають таку ж довжину. Природно виникає проблема їхнього запам'ятовування, зберігання й витрата на обробку цих послідовностей. Найпростішим рішенням, що дозволяє значно скоротити об'єм збереженої інформації про еталонні вихідні реакції є одержання інтегральних оцінок, що мають меншу розмірність. Для цього використовуються алгоритми стиску інформації.

У результаті їхнього застосування формуються компактні оцінки стислої інформації. Розглянемо алгоритми стиску даних для випадку бінарної послідовності $\{y(k)\}$, що складає з l послідовно формованих двійкових змінних.

Псевдовипадкове тестування

Найбільше часто при формуванні псевдовипадкових послідовностей використовуються два методи. Перший з них, який лежить в основі більшості програмних датчиків псевдовипадкових чисел, використовує рекуррентні співвідношення. Цей метод володіє рядом недоліків, зокрема, малою періодичністю. Відносно k проблеми тестування цифрових схем періодичність може помітно знизити повноту контролю. Крім того, він відрізняється складністю практичної реалізації. Тому найбільш широко завідносуюється другий метод, заснований на використанні співвідношення:

$$a_k = \sum_{i=1}^m \oplus \alpha_i a_{k-i}, \quad (2.1)$$

де:

– k – номер такту, $k = 0, 1, 2, \dots$;

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– $a_k \in \{0,1\}$ – символи послідовності;

– $\alpha_i \in \{0,1\}$ – постійні коефіцієнти;

– $\sum_{i=1}^m \oplus$ – операція підсумовування по модулю два m логічних змінних.

При відповідному виборі коефіцієнтів α_i на підставі характеристичного полінома:

$$\varphi(x) = 1 \oplus \alpha_1 x^1 \oplus \alpha_2 x^2 \oplus \dots \oplus \alpha_{m-1} x^{m-1} \oplus \alpha_m x^m, \quad (2.2)$$

який повинен бути примітивним, послідовність має максимальну довжину, рівну $2^m - 1$. Така послідовність називається М-послідовністю.

Використання таких послідовностей припускає застосування сигнатурного аналізу як методу стиску реакцій цифрової схеми.

Типова структурна схема сигнатурного аналізатора складається з регістра зрушення й суматора по модулю два, на входи якого підключені виходи розрядів регістра відповідно k породжуючого полінома $\varphi(x)$ (рисунок 2.1).

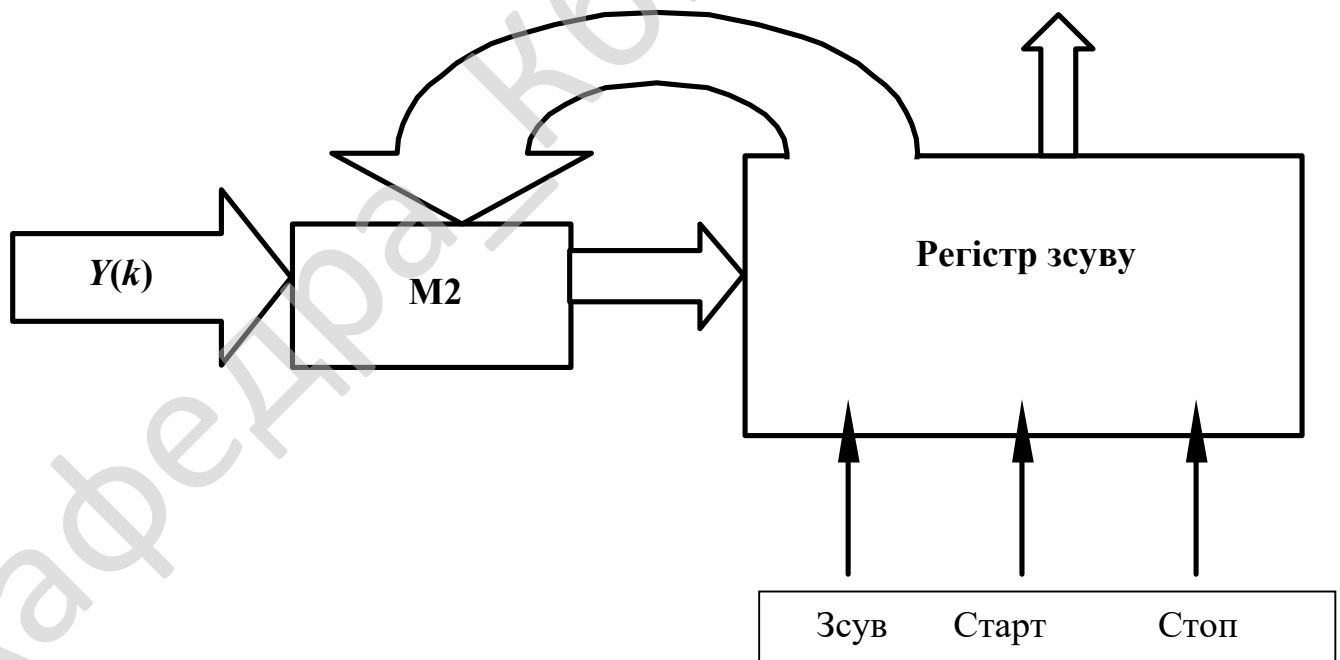


Рисунок 2.1 – Типова структурна схема сигнатурного аналізатора

Керуючими сигналами сигнатурного аналізатора є СТАРТ, СТОП і ЗСУВ. Сигнали СТАРТ і СТОП формують часовий інтервал, протягом якого здійснюється процедура стиску інформації на аналізаторі. Під дією сигналу СТАРТ елементи пам'яті регістра зрушення встановлюються у вихідний стан, як правило нульове, а сам регістр зрушення починає виконувати функцію зрушення на один розряд у право під дією син10/11онізуючих імпульсів ЗСУВ. По приходу кожного син10/11онізуючого імпульсу в перший розряд регістра зрушення записується інформація, що відповідає виразу:

$$a_1(K) = y(K) \oplus \sum_{i=1}^m \alpha_i a_i(K-1), \quad (2.3)$$

де:

- $y(K) \in \{0,1\}$ – k -й символ стислої послідовності $\{y(K)\}$, $K = \overline{1, l}$;
- $\alpha_i \in \{0,1\}$ – коефіцієнти полінома, що $\varphi(x)$ породжує;
- $a_i(k-1) \in \{0,1\}$ – уміст i -того елемента пам'яті регістра зрушення 1 в $(k-1)$

такт.

Процедура зрушення інформації в регістрі описується співвідношенням

$$a_j(k) = a_{j-1}(k-1), \quad j = \overline{2, m}. \quad (2.4)$$

Таким чином, повний математичний опис функціонування сигнатурного аналізатора має такий вигляд:

$$a_i(0) = 0, \quad i = \overline{1, m}, \quad a_1(k) = y(k) \oplus \sum_{i=1}^m \alpha_i a_i(k-1), \quad (2.5)$$

$$a_j(k) = a_{j-1}(k-1), \quad j = \overline{2, m}, \quad k = \overline{1, l}, \quad (2.6)$$

причому l , як правило, приймається рівним або менше величини (2^m-1) , і відповідно є довжиною стислої послідовності.

Після закінчення l тактів функціонування сигнатурного аналізатора на його елементах пам'яті фіксується двійковий код, що являє собою сигнатуру, відображену у вигляді 16-річного коду.

Синдромне тестування

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Синдромом (контрольною сумою) деякої булевої функції n змінних є співвідношення

$$S = R_{5/2}^n, \quad (2.7)$$

де R_5 дорівнює числу одиничних значень функції відповідно k таблиці істинності для $l = 2^n$. Визначення поняття синдрому однозначно припускає використання генератора счетчикових послідовностей для формування всіляких двійкових комбінацій з n вхідних змінних при тестуванні схеми, що реалізує задану функцію. Подальшим розвитком синдромного тестування є спектральний метод оцінки вихідних реакцій цифрових схем і кореляційний метод.

Типи несправностей цифрових схем

Проблема тестового діагностування цифрових схем виникає на різних етапах їхнього виробництва й експлуатації й включає взаємозалежні задачі. Перша з них полягає у визначенні, у якому стані перебуває досліджувана схема.

Основним станом цифрової схеми є справний – такий стан схеми, при якому вона задовольняє всім вимогам технічної документації. У протилежному випадку схема перебуває в одному з несправних станів.

Якщо встановлено, що схема несправна, то вирішується друга задача: здійснюється пошук несправної схеми, ціль якого – визначення місця й виду несправності.

Несправності ЦС з'являються в результаті застосування несправних компонентів, таких, як логічні елементи, що реалізують найпростіші логічні функції, елементи пам'яті й ін. крім того, причиною несправностей можуть бути виникнення розривів або коротких замикань у міжкомпонентних з'єднаннях, порушення умов експлуатації схеми, наявність помилок при проектуванні й виробництві й ряд інших факторів.

Із множини різних видів несправностей виділяється клас логічних несправностей, які змінюють функції елементів ЦС зазначений тип несправностей займає домінуюче місце серед несправностей ЦС. Для їхнього опису в більшості випадків використовують наступні математичні моделі:

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

- Константні несправності.
- Несправності типу «Коротке замикання».
- Інверсні несправності.

Найбільш загальною й часто завідносуваною моделлю логічних несправностей є константні несправності: константний нуль і константна одиниця, що означає наявність постійного рівня логічного нуля або логічної одиниці на одному з полюсів логічного елемента. Така модель несправностей часто називається класичною й широко використовується для опису інших типів несправностей. Несправності типу «Коротке замикання» з'являються при короткому замиканні входів і виходів логічних елементів.

Інверсні несправності описують фізичні дефекти ЦС, що приводять *k* появи фіктивного інвертора по входу або по виходу логічного елемента. Інверсні несправності в сукупності з константними, у ряді випадків використовуються для побудови повної моделі несправної цифрової схеми.

Генератори тестових послідовностей

Класична стратегія тестування цифрових схем заснована на формуванні тестових послідовностей, що дозволяють виявляти задані множини їхніх несправностей. Для реалізації генератора тестової послідовності бажано використовувати найпростіші методи, що дозволяють уникнути складної процедури їхнього синтезу. До них відносяться наступні алгоритми:

- формування всіляких тестових наборів, тобто повного перебору двійкових комбінацій. У результаті застосування подібного алгоритму генеруються рахункові послідовності;
- формування псевдовипадкових тестових послідовностей;
- формування випадкових тестових наборів, з необхідними ймовірностями одиничного й нульового символів по кожному входу цифрової схеми.

Основною властивістю перерахованих вище алгоритмів є те, що в результаті їхнього застосування відтворюються послідовності дуже великої довжини.

Для процесу навчання були обрані два перших алгоритми побудови генераторів тестових послідовностей, та розроблені два модулі для емуляції роботи генераторів:

- модуль емуляція генератора рахункової послідовності;
- модуль емуляції роботи багатоканального генератора М-послідовності, що дозволяє генерувати псевдовипадкову послідовність і порівняно просто регулювати її максимальну довжину й число каналів залежно від числа входів цифрової схеми.

Генератор М-послідовності

В апаратурних псевдовипадкових датчиках і вузлах ЕОМ при генеруванні ПВЧП із рівномірним розподілом найбільше часто використовується метод, що полягає в одержанні лінійної двійкової послідовності по рекуррентному вираженню:

$$a_i = \sum_{k=1}^m \oplus \alpha_k a_{i-k}, \quad (2.9)$$

де:

- i – номер такту, $i = 0, 1, 2, 3, \dots$;
- $a_i \in \{0, 1\}$ – символи вихідної послідовності;
- $\alpha_k \in \{0, 1\}$ – постійні коефіцієнти.

При відповідному виборі коефіцієнтів $\{\alpha_k\}$ генеруєма числова послідовність має максимальну (для даного m) величину періоду й називається М-послідовністю. Одним з головних переваг методу генерування ПВ-послідовностей максимальної довжини є простота його реалізації.

Генератор М-послідовності може бути побудований двома методами, що відрізняються способом включення суматорів по модулю два: вони можуть

включатися як у ланцюг зворотного зв'язка генератора, так і в меж розрядні зв'язки елементів пам'яті регістрів зрушення.

Структурна схема генератора M – послідовності, побудованого по способу включення суматорів у ланцюг зворотного зв'язка представлена на рисунку 2.2.

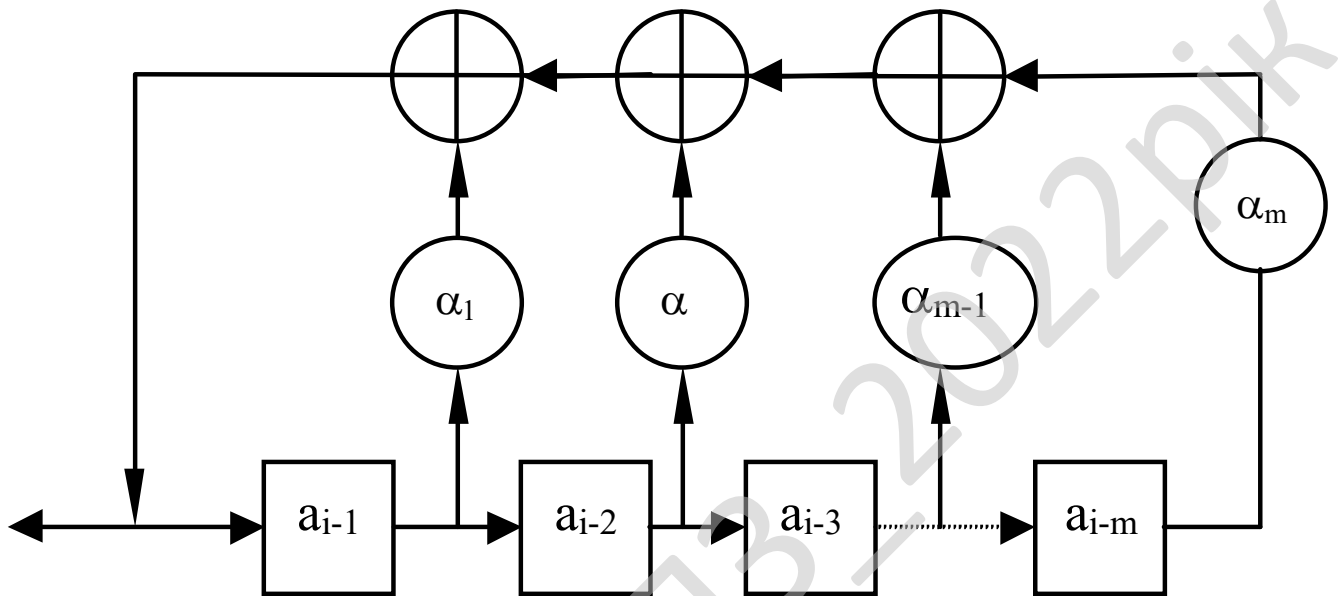


Рисунок 2.2 – Структурна схема генератора M -послідовності

Генератор M -послідовності із суматорами по модулю два, що стоять у ланцюзі зворотного зв'язка: $a_i, a_{i-1}, a_{i-2}, \dots, a_{i-m}$ – символи послідовності; α_i – коефіцієнти, що визначають вид зворотного зв'язка.

Алгоритм розмноження M -послідовності

Для того, щоб забезпечити різні режими випробувань, генератори випробуваних сигналів повинні задовольняти ряду вимог (багатоканальність, швидкодія, достатня довжина періоду й т.д.). В основі найбільш перспективного методу побудови швидкодіючого паралельного генератора псевдовипадкових послідовностей іспитових сигналів лежить ідея використання (як незалежні послідовності для формування розрядів чергового коду) ділянок однієї й тої ж послідовності. У цьому випадку генерування різних ділянок здійснюється за допомогою η -входових суматорів по модулю два, тобто $\eta \in \{2, m\}$, де m –

розрядність регістра зрушення. З'єднання суматорів по модулю два з розрядами регістра зрушення визначаються набором коефіцієнтів $\delta_i(1) \in \{0,1\}$ ($i = 1,2,3,..m$), значення яких залежать від величини зрушення l ($l = 1,2,3,..$) та виду полінома, що породжує.

Методика вибору коефіцієнтів $\delta_i(1)$, однозначно визначальні зв'язки багатовходового суматора по модулю два, описується на ітераційному підході, коли на підставі $\delta_i(h)$, по розрахункових з'єднаннях перебувають $\delta_i(1)$, ($h = 1,2,..,h < l$).

Припустимо, що коефіцієнти $\delta_i(1)$ і $\delta_i(S)$, що дозволяють одержати зрушені копії M -послідовності на 1 і S тактів, відомий; тоді вміст $a_1(k+1)$ першого розряду регістра зрушення в $(k+1)$ -м такті роботи визначається в такий спосіб:

$$a_1(k+1) = \sum_{i=1}^m \oplus \delta_i(1)a_i(k), \quad (2.10)$$

де:

– $a_i(k)$ вміст i -того розряду регістра зрушення в k -м такті його роботи ($k = 0,1,2,..$);

– символ $\sum \oplus$ означає операцію підсумовування по модулю два.

Вміст першого розряду регістра зрушення в $(k+S)$ -м такті роботи має вигляд:

$$a_1(k+s) = \sum_{i=1}^m \oplus \delta_i(s)a_i(k), \quad (2.11)$$

для визначення вмісту першого розряду регістра зрушення в $(k+1+S)$ -м такті, аналогічно як і для (2.10) і (2.11), необхідно попередньо вибрати чисельне значення коефіцієнтів $\delta_i(1+s)$. З іншого боку, $a_1(k+1+S)$ можна знайти на підставі (2.10) у такий спосіб:

$$a_1(k+1+s) = \sum_{i=1}^m \oplus \delta_i(s)a_i(k+1), \quad (2.12)$$

де $a_i(k+1)$ з обліком (2.10) приймає вид:

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

$$a_1(k+1) = \sum_{i=1}^m \oplus \delta_i(1) a_i(1), \quad (2.13)$$

$$a_{1+p}(k+1) = \sum_{n=1}^{m-p} \oplus \delta_n(1) a_{n+p}(k) \oplus \sum_{n=1}^p \oplus \delta_{m-p+n}(1) a_m(k-n) \quad (p=1, m-1), \quad (2.14)$$

значення $a_m(k-p)$ обчислюються по формулі:

$$a_m(k-p) = \sum_{i=1}^m \oplus \beta_{i,p} a_i(k), \quad (2.15)$$

де $\beta_{i,p} \in \{0,1\}$ ($i=1,2,\dots,m, p=1,2,\dots,m-1$) – постійні коефіцієнти, обумовлені як:

$$\beta_{i,g} = \sum_{n=1}^{g-1} \oplus \alpha_{m-n} \beta_{1,g-n} \quad (g=2, m-1), \quad (2.16)$$

$$\beta_{y,1} = \alpha_{y-1}, \quad (2.17)$$

$$\beta_{y,c} = \alpha_{y-c} \oplus \sum_{n=1}^{c-1} \oplus \alpha_{m-n} \beta_{y,c-n} \quad (c=2, y-1), \quad (2.18)$$

$$\beta_{y,y} = 1 \oplus \sum_{n=1}^{y-1} \oplus \alpha_{m-n} \beta_{y,y-n}, \quad (2.19)$$

$$\beta_{y,c} = \sum_{n=1}^{c-1} \oplus \alpha_{m-n} \beta_{y,c-n}, \quad (c=y+1, m-1, y=2, m-1), \quad (2.20)$$

$$\beta_{m,1} = \alpha_{m-1}, \quad (2.21)$$

$$\beta_{m,g} = \alpha_{m-g} \oplus \sum_{n=1}^{g-1} \oplus \alpha_{m-n} \beta_{g-n} \quad (g=2, m-1). \quad (2.22)$$

Остаточно для $\delta_i(1+s)$ одержуємо:

$$\delta_i(1+s) = \sum_{n=1}^{c-1} \oplus \delta_n(s) \delta_{i+1-n}(1) \oplus \sum_{n=1}^{m-1} \beta_{i,n} \sum_{c=n+1}^m \delta_c(s) \delta_{m+n+1-c}(1), \quad (i=1, m). \quad (2.23)$$

Таким чином, описано процедуру отримання тестової M-послідовності.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

– Тип даних Delphi «record» тепер підтримуватимуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TMemo на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4к моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису `custom managed records`. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCl, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як перегляд і інспекція типів начебто рядків C++ і Delphi, а також колекцій STL, включаючи std::vector, std::map і інших. Крім того, згенерована для застосунку відладочна інформація має інший внутрішній формат, сприяючи більш стабільному й багатому на можливості процесу налагодження, більш докладним перегляду й інспекції в debug-time.

Підвищення якості й швидкодії інструментів

- Велика кількість поліпшень STL від Dinkumware.
- Поліпшені деякі найважливіші методи й області RTL, на базі поліпшень сумісності з популярними бібліотеками C++.
- Поліпшена підтримка Stafe.
- Велика кількість виправлень для підвищення стабільності і якості.

- Відновлення Windows API – Обновлено й додали безліч декларацій API щоб добитися ще більшої інтеграції із платформою Windows.

- Загальні вдосконалення в бібліотеці доступу до БД FireDAC, включаючи оновлені драйвера для FireBird, PostgreSQL і SQLite. Вибір статичного або динамічного підключення SQLite до застосунку.

Змінені стилі VCL для High DPI

В 10.4, архітектура стилізації VCL була суттєво розширена для підтримки High DPI і 4K моніторів. Тепер усі елементи UI на формі VCL автоматично масштабуються під відповідне до монітора дозвіл для показу форми. Був оновлений API стилізації для підтримки стилів high DPI.

Кожний графічний елемент UI може бути обраний з наборів різних масштабів і масштабований до потрібного DPI, що дає чітке зображення елементів UI на всіх моніторах.

Нові High DPI стилі й стилізація окремих VCL компонент

Обновлено велике число вбудованих і преміальних VCL стилів для підтримки нового режиму стилізації High-dpi. Це дозволяє вам створювати застосунку з відмінним дизайном для всіх моніторів.

Розроблювачі VCL застосунків тепер можуть використовувати трохи VCL стилів на різних формах в одному застосунку або в різних компонентах на одній формі. Це також включає стилізацію компонентів загальною темою для платформи. Крім застосункової гнучкості використання стилів, це дозволяє використовувати нестилізовані компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи виявлення несправності елементів цифрових пристроїв.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

- а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

б) вибрати та обґрунтувати методика побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Реалізація алгоритму, що моделює роботу генераторів тестових послідовностей

На схемі генератор рахункової послідовності відображається як на рисунку 3.1.

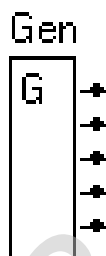


Рисунок 3.1 – Позначення генератора рахункової послідовності

На схемі генератор М-послідовності відображається як на рисунку 3.2.

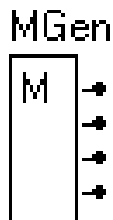


Рисунок 3.2 – Позначення генератора М-послідовності

Алгоритми роботи генераторів рахункової послідовності описаний і М-послідовності описаний у пункті 2.1.

Розробка й реалізація модуля моделюючий алгоритм діагностики з використанням компактних методів тестування

Для діагностики цифрових схем особливий інтерес представляють сигнатурний аналіз, зокрема, багатоканальний, в основі побудови якого лежить алгоритм стиску інформації, і метод компактного тестування, що використовує алгоритм рахунку одиниць, що знаходить широке застосування при реалізації вбудованого тестування. Тому для навчальної системи при моделюванні процесу діагностики цифрових схем були обрані два вищевказаних методи компактного тестування.

Сигнатурний аналізатор

Для діагностики цифрових схем особливий інтерес представляють сигнатурний аналіз, зокрема, багатоканальний, в основі побудови якого лежить алгоритм стиску інформації, і метод компактного тестування, що використовує алгоритм підрахунку одиниць, що знаходить широке застосування при реалізації вбудованого тестування. Тому для навчальної системи при моделюванні процесу діагностики цифрових схем були обрані два вищевказаних методи компактного тестування.

Для опису процедури стиску інформації, заснованої на застосуванні сигнатурного аналізу, використовуються різні математичні моделі й алгоритми. Найбільше широко використовуються два алгоритми:

1. Метод згортки, при якому значення еталонної сигнатури послідовності, формованої на кожній з полюсів ЦС, зокрема й на вихідному, виходить при обробці її символів по відношенню 2.1.

2. Алгоритм ділення полінома на поліном. При цьому як ділене використовується потік стислової послідовності даних, описуваних поліномом $k(x)$ ступеня $(l,1)$, де l – кількість біт у послідовності. Дільником служить примітивний поліном $\varphi(x)$, у результаті ділення на який виходить частку $q(x)$ і остача $S(x)$, зв'язані класичним співвідношенням виду:

$$k(x) = q(x)\varphi(x) \oplus S(x), \quad (3.1)$$

де остача $S(x)$ називається сигнатурою.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

Найбільш кращим методом синтезу багатоканальних сигнатурних аналізаторів є метод, що дозволяє синтезувати МС з довільною кількістю входів і не залежним від нього множиною елементів пам'яті, обумовленим тільки старшим ступенем полінома, що $\varphi(x)$ породжує. Даний метод ґрунтується на застосуванні примітивного полінома $\varphi(x) = 1 \oplus \alpha_1 x^1 \oplus \alpha_2 x^2 \oplus \dots \oplus \alpha_{1m} x^m$, де $m = \deg \varphi(x)$ визначає достовірність аналізу, а також розрядність формованих сигнатур.

Для довільного $\varphi(x)$ функціонування одноканального сигнатурного описується системою рівнянь:

$$a_i = 0, \quad i = \overline{1, m}, \quad (3.2)$$

$$a_1(k) = y(k) \oplus \sum_{i=1}^m \oplus \alpha_i a_i(k-1), \quad a_j(k) = a_{j-1}(k-1), \quad j = \overline{2, m}, \quad k = \overline{1, l}, \quad (3.3)$$

де:

– $a_j(k) \in \{0,1\}$ – вміст j -го елемента пам'яті аналізатора в k -й такт його роботи;

– $y(k) \in \{0,1\}$ – значення двійкового символу, що надходить на вхід аналізатора в k -й такт;

– $\alpha_i \in \{0,1\}$ – коефіцієнти, що залежать від виду полінома, що $\varphi(x)$ породжує.

З вираження (3.3) слідує, що вміст першого елемента пам'яті аналізатора в $(k+1)$ -й такт його роботи визначається як:

$$a_1(k+1) = y(k+1) \oplus \alpha_1 y(k) \oplus \alpha_1 \sum_{i=1}^n \oplus \alpha_i a_i(k-1) \oplus \sum_{i=2}^m \oplus \alpha_i a_{i-1}(k-1), \quad (3.4)$$

а в $(k+2)$ -й такт:

$$a_1(k+2) = y(k+2) \oplus \alpha_1 y(k+1) \oplus \alpha_1 \alpha_1 y(k) \oplus \alpha_1 \alpha_1 \sum_{i=1}^n \oplus \alpha_i a_i(k-1) \oplus \oplus \alpha_1 \sum_{i=2}^m \oplus \alpha_i a_{i-1}(k-1) \oplus \alpha_2 y(k) \oplus \alpha_2 \sum_{i=1}^m \oplus \alpha_i a_i(k-1) \oplus \sum_{i=3}^m \oplus \alpha_i a_{i-2}(k-1). \quad (3.5)$$

У загальному випадку для деякого $k+n-1$ -го такту можна записати:

$$a_1(k-n-1) = \sum_{i=1}^n \oplus \beta_i y(k-1+i) \oplus \sum_{i=1}^m \oplus \delta_i(n) a_i(k-1), \quad (3.6)$$

де $\delta_i(n) \in \{0,1\}, i = \overline{1, m}$ – коефіцієнти, що дозволяють формувати зрушену на n тактів копію M -послідовності, описувану поліномом $\varphi(x)$. Значення $\delta_i(n)$ визначаються як:

$$\delta_i(n) = \alpha_{i+n-1} \oplus \sum_{d=1}^{n-1} \oplus \alpha_d \delta_i(n-d), \quad i = \overline{1, m-n+1}, \quad n = \overline{1, m}, \quad (3.7)$$

$$\delta_i(n) = \sum_{d=1}^{n-1} \oplus \alpha_d \delta_i(n-d), \quad i = \overline{m-n+2, m}, \quad n = \overline{1, m}, \quad (3.8)$$

$$\delta_i(n) = \sum_{d=1}^m \oplus \alpha_d \delta_i(n-d), \quad i = \overline{1, m}, \quad n > m. \quad (3.9)$$

Крім того, чисельні значення $\delta_i(n), i = \overline{1, m}$, можуть бути отримані в результаті виконання швидких формальних процедур.

Коефіцієнти $\beta_i(n) \in \{0,1\}$ визначаються в такий спосіб:

$$\beta_i(n) = 0, \quad i > n, \quad (3.10)$$

$$\beta_i(n) = 1, \quad i = n, \quad (3.11)$$

$$\beta_i(n) = \sum_{d=1}^{n-1} \oplus \alpha_d \beta_i(n-d), \quad n > i, \quad n-i \leq m, \quad (3.12)$$

$$\beta_i(n) = \sum_{d=1}^m \oplus \alpha_d \beta_i(n-d), \quad n-i > m. \quad (3.13)$$

З вираження (3.6) для $a_1(k+n-1)$ можна одержати його значення на підставі n символів $y(k), y(k+1), \dots, y(k+n-1)$ послідовності $\{y(k)\}$ і m вихідних значень $a_i(k-1), i = \overline{1, m}$. У теж час зазначене вираження використовується для побудови функціональної схеми сигнатурного аналізатора, що у кожний такт обробляє n символів послідовності $\{y(k)\}$. При цьому подібний аналізатор буде мати n входів, що дозволяє застосовувати його для контролю цифрових схем, що мають n виходів, причому n вихідних послідовностей у цьому випадку перетворяться в одного виду:

$$y_1(k)y_2(k)\dots y_n(k)y_1(k+1)y_2(k+1)\dots y_n(k+1)y_1(k+2)\dots y_n(k+2)\dots, \quad (3.14)$$

де $y_v(k) \in \{0,1\}$, $v = \overline{1,n}$, – значення двійкового символу на v -ом виході досліджуваної цифрової схеми в k -й такт її роботи.

Функціонування аналізатора, що обробляє послідовність (3.14) відповідно до (3.3) і (3.6), буде описуватися наступною системою рівнянь:

$$a_i(0) = 0, \quad i = \overline{1,m}, \quad (3.15)$$

$$a_1(k+n-1) = \sum_{d=1}^n \oplus \beta_d(n)y_d(k) \oplus \sum_{i=1}^m \oplus \delta(n)a_i(k-1), \quad (3.16)$$

$$a_j(k+n-1) = a_1(k+n-j), \quad j < n, \quad (3.17)$$

$$a_j(k+n-1) = a_{j-n+1}(k-1), \quad j \geq n, \quad (3.18)$$

де $k = \overline{1,l}$, $j = \overline{2,m}$.

Використовую систему рівнянь (3.15 – 3.18), виявляється можливим побудова багатоканального аналізатора, що виконує за один такт ті ж перетворення з послідовністю, що й одноканальний за n тактів.

Синдромне тестування або метод рахунку одиниць

Синдромом (контрольною сумою) деякої булевої функції n змінних є співвідношення

$$S = R/2^n, \quad (3.19)$$

де R обчислюється по вираженню:

$$R = \sum_{k=1}^l y(k). \quad (3.20)$$

Для $l = 2^n$ і дорівнює числу одиничних значень функції відповідно до таблиці істинності. Визначення поняття синдрому однозначно припускає використання генератора рахункових послідовностей двійкових комбінацій з n входних змінних при тестуванні схеми, що реалізує задану функцію.

Блок пошуку несправностей

За допомогою багатоканальних сигнатурних аналізаторів можна істотно прискорити процедуру контролю цифрових схем, що збільшується в n раз, де n – кількість входів застосовуваного аналізатора. У випадку збігу реально отриманої

сигнатури з її еталонним значенням вважається, що з досить високою ймовірністю схема, що перевіряється, перебуває в справному стані. На цьому процедура її дослідження кінчається. У протилежному випадку, коли схема містить несправності, реальна сигнатура, як правило, відрізняється від еталонної, що служить основним аргументом для прийняття гіпотези про несправний стан схеми. У той же час вид отриманої сигнатури не несе ніякої додаткової інформації про характер виниклої несправності. Більше того залишається відкритим питання про те, які з n аналізованих послідовностей, що ініціюють реальну сигнатуру, містять помилки, тобто виникає задача локалізації несправності з точністю до послідовності, що несе інформацію про її присутність.

Сумарна сигнатура $S(x)$, отримана для послідовності $\{y_v(k)\}$, $v = 1, n, k = 1, l$, на n -канальному сигнатурному аналізаторі, дорівнює поразрядної сумі по модулі дві сигнатури $S_v(x)$, $v = 1, n$. Причому кожна сигнатура $S_j(x)$, $j \in \{1, 2, 3, \dots, n\}$, формується для послідовності $\{y_j(k)\}$ за умови, що $\{y_q(k)\} = 0000 \dots 00, q \neq j \in \{1, 2, \dots, n\}$...

Алгоритм контролю цифрової схеми з локалізацією несправності до першої послідовності, що містить викликані нею помилки.

1. У результаті аналізу $n = 2^d$ реальних послідовностей $\{y^*(k)\}$, $v = 1, n$, на n -канальному аналізаторі визначається значення сигнатури $S^*(x)$, що відповідає співвідношенню:

$$\sum_{v=1}^n \oplus \sum_{k=1}^l \oplus y^*_{v}(k)x^{(k-1)n+v-1} = g^*(x)\varphi^{-1}(x) \oplus S^*(x). \quad (3.20)$$

2. По вираженню $S(x) = \sum_{v=1}^n \oplus S_v(x)$ обчислюється еталонне значення сигнатури $S(x)$.

3. Реальне значення сигнатури $S^*(x)$ рівняється з еталонною сигнатурою $S(x)$. У випадку виконання рівності $S^*(x) = S(x)$ переходять k виконання п.11 і процедура контролю вважається кінченою. У протилежному випадку, коли $S^*(x) \neq S(x)$, виконується наступний етап алгоритму.

4. Вся множина вхідних послідовностей розбивається на дві групи, причому номери послідовностей $\{y_1(k)\}$, $\{y_2(k)\}$, $\{y_3(k)\}$, ..., $\{y_{n/2}(k)\}$ становлять множину $A_2 = \{1, 2, 3, \dots, n/2\}$, а послідовностей $\{y_{n/2+1}(k)\}$, $\{y_{n/2+2}(k)\}$, ..., $\{y_n(k)\}$ – множину $A_2 = \{n/2+1, n/2+2, \dots, n\}$, величині i привласнюється значення 1.

5. У результаті аналізу реальних послідовностей, номери яких задаються множиною A_1 , на n -канальному аналізаторі за умови, що послідовності, номери яких не визначені множиною A_1 , є нульовими, визначається значення реальної сигнатури $S^*(x)$.

6. На підставі вираження $S(x) = \sum_{j \in A_1} S_j(x)$ одержуємо $S(x)$.

7. Перевіряється справедливість рівності $S(x) = S^*(x)$. У випадку його виконання елементи множини A_1 замінюються елементами множини A_2 .

8. Значення змінної i збільшується на одиницю. Потім його величина рівняється з величиною d . При $i \leq d$ переходять k наступному пункту алгоритму, у противному випадку виконується пункт 10.

9. За поточним значенням множини A_1 формуються нові множини A_1 і A_2 . Новими елементами множини A_1 буде перша половина його поточних елементів, друга половина привласнюється множині A_2 . Після визначення множин A_1 і A_2 переходять до виконання п. 5.

10. Єдиний елемент множини A_1 являє собою номер помилкової послідовності, формованої на одному з полюсів досліджуваної схеми.

11. Процедура контролю цифрової схеми вважається кінченою.

Визначення оцінки ефективності методів сигнатурного аналізатора й рахунку одиниць

Достовірність сигнатурного аналізу

Повнота не виявлення несправностей цифрової схеми в першу чергу залежить від якості тестових впливів. Якщо певна несправність не проявляється у вигляді перекручування їхніх символів, то вона не може бути виявлена в результаті застосування сигнатурного аналізу, що є не більш ефективним ніж

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Як більш точну міра оцінки достоїнств сигнатурного аналізатора розглянемо розподіл імовірності невиявлення помилки залежно від її кратності μ , тобто визначимо значення P_n^μ , де $\mu = 1, 2, 3, \dots, 2^{m-1}$.

Можна показати, що V_n^μ помилок, що не виявляються, визначається в такий спосіб:

$$V_n^1 = 0; V_n^2 = 0; V_n^\mu = [V^\mu - V_n^{\mu-1} - V_n^{\mu-2}(2^m + 1 - \mu)] / \mu, \quad (3.23)$$

а кількість можливих помилок з μ біт визначається як $V^\mu = C_L^\mu$.

І тоді вираження для ймовірності не виявлення помилки приймає вид:

$$P_n^1 = 0, P_n^2 = 0, P_n^\mu = [(1/2^m - \mu)][1 - P_n^{\mu-1} - (\mu - 1)P_n^{\mu-2}], \quad \mu = 3, 4, \dots, 2^m. \quad (3.24)$$

Аналіз показує, що для досить більших m $P_n^\mu \cong 1/2^m, \mu = 3, 4, \dots, 2^m - 2$, тобто при $m > 7$ імовірність виявлення помилки $P_d^\mu = 1 - P_n^\mu$ практично рівняється одиниці.

Достовірність методу рахунку одиниць

Як характеристика, що дозволяє оцінити метод компактного тестування доцільно використовувати розподіл імовірностей не виявлення помилки залежно від її кратності:

$$P^\mu = P_v^\mu * P_n^\mu, \quad (3.25)$$

де:

- μ – кратність помилки,
- P_v^μ – імовірність виникнення помилки кратності μ ;
- P_n^μ – імовірність не виявлення виниклої помилки кратності μ , що

визначається як відношення кількості помилок, що не виявляються, кратності μ до загальної кількості можливих помилок з μ невірних символів у послідовності довжиною l .

Значення P_v^μ визначається видом цифрової схеми, що перевіряється, множиною можливих її несправностей, а також типом тестових послідовностей, причому розподіл P_v^μ імовірностей може мати зовсім довільний вид і значно

змінюватися залежно від виниклої несправності, виду схеми й тестової послідовності.

Значення P_n^μ характеризується тільки методом компактного тестування й дозволяє провести його порівняльну оцінку в порівнянні з іншими методами. Тому для різних методів залежно від їхнього розподілу ймовірностей P_n^μ можуть бути отримані оцінки ефективності контролю ЦС у вигляді розподілу P_n . Аналіз цього виду розподілу дозволяє ухвалити рішення щодо доцільності застосування того або іншого методу компактного тестування. Причому для спрощення алгоритму ухвалення рішення варто використовувати більше компактну характеристику, наприклад сумарну ймовірність не виявлення помилки P_n , що обчислюється як:

$$P_n = \sum_{\mu=1}^l P^\mu = \sum_{\mu=1}^l P_v^\mu * P_n^\mu . \quad (3.26)$$

У цьому випадку величина P_n буде характеризувати той або інший метод компактного тестування для цілком конкретного розподілу ймовірностей P_v^μ виникнення несправностей залежно від її кратності.

3.2 Розробка структурної схеми

Програма написана під операційну систему Windows 10/11. Так як сама операційна система Windows 10/11 є графічною, то інтерфейси програм написаних під цю систему схожі один на одного. Внаслідок цього навчання роботі з даною програмою полегшується. Графічний інтерфейс побудований таким чином, щоб користувачеві було, як можна зручніше й понятніше працювати із програмою.

На рисунку 3.3 зображена структурна схема розробленої, в результаті виконання магістерського проектування системи.

Як видно з рисунка структурно система складається з наступних головних

блоків:

- інтерфейс користувача;
- блок моделювання;
- блок діагностики;
- бази даних.

Розглянемо ці структурні блоки більш детально.

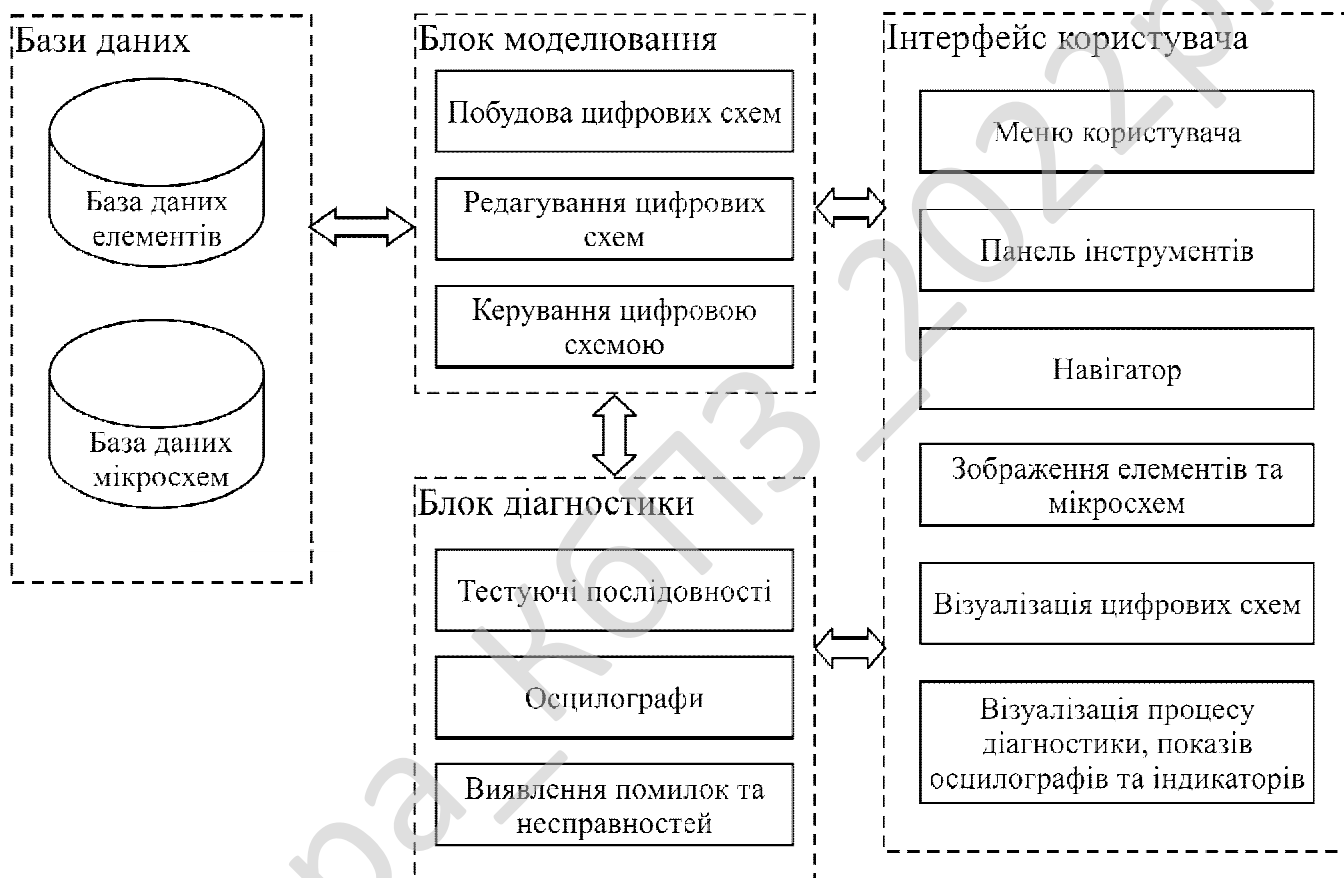


Рисунок 3.3 – Структурна схема системи

Найбільш великим структурним блоком є блок інтерфейсу користувача. Цей блок є дуже важливим, тому, що багато в чому, від інтерфейсу користувача залежить наскільки легко можна буде навчати користувача діагностиці цифрових схем.

Інтерфейс користувача складається з наступних структурних підблоків:

- меню користувача;

- панель інструментів;
- навігатор;
- графічне зображення елементів та мікросхем;
- візуалізація цифрових схем;
- візуалізація процесу діагностики, показів осцилографа та індикаторів.

Розглянемо блок діагностики. Структурно він складається з наступних підблоків:

- блок формування та запуску тестуючи послідовностей;
- блок осцилографів, для зняття осцилограм з виходів та входів цифрової схеми;
- блок виявлення помилок та несправностей.

Перейдемо до розгляду блоку моделювання. Він складається з наступних структурних підблоків:

- побудова цифрових схем, де можна з наявних компонентів побудувати цифрову схему;
- редагування цифрових схем – для внесення змін у побудовану цифрову схему;
- керування цифровою схемою, для відслідковування того, як проходять сигнали по цифровій схемі.

Розглянемо блок роботи з базою даних. Він складається з наступних підблоків:

- база даних елементів;
- база даних мікросхем.

3.3 Розробка функціональної схеми

Розроблена система навчання може бути представлена у вигляді системи, основними функціональними вузлами якої є:

- генератори тестових послідовностей;

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

- блок моделювання досліджуваних схем;
- блок відображення й обробки вихідних реакцій і стиску інформації;
- блок помилок;
- блок визначення ймовірностей не виявлення помилок.

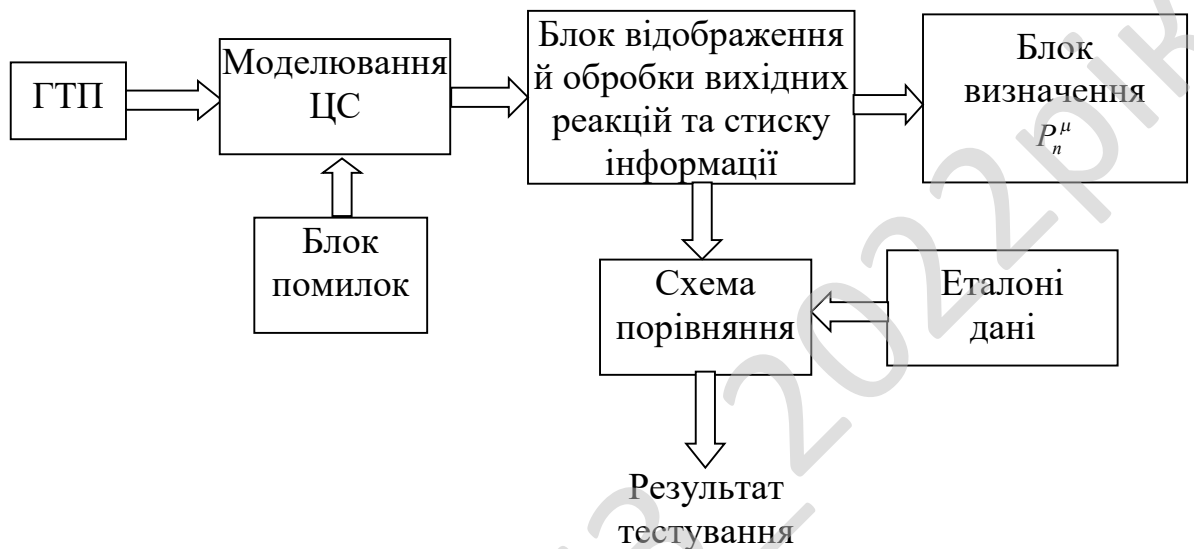


Рисунок 3.4 – Функціональна схема системи

Моделювання складних логічних схем на великій кількості вхідних наборів ефективно можна здійснювати тільки за допомогою ЕОМ. Для того щоб змодельовати роботу пристрою на ЕОМ, необхідно описати математичну модель цього пристрою.

Для моделювання цифрових схем, насамперед, необхідно описати схему, для цього була змодельована математична модель яка описує цифрові схеми під дану систему.

Кожний елемент схеми це об'єкт, що має порядковий номер на схемі, тип, списки входів і виходів. Кожний вхід елемента зберігає інформацію про попередній елемент. Внаслідок цього кожний елемент може визначити логічний стан попереднього, його тип, порядковий номер на схемі, вихід з яким він з'єднаний.

Лінії, що з'єднують входи й виходи елементів є такими ж об'єктами, як і елементи цифрової схеми, за винятком того, що лінія має тільки один вхід і один вихід, і не виконує логічних функцій.

У програмі реалізовані всі типи логічних елементів, внаслідок цього можна побудувати велику кількість різноманітних цифрових схем.

3.6 Розробка діаграми процесів

На рисунку 3.5 зображена діаграма взаємодії процесів, які відбуваються під час роботи програмного продукту, що розроблений у результаті виконання магістерського проектування. Програма починає діяти з запуску процесу початку/кінця роботи програми.

Цей процес працює з наступними процесами:

- процес відкриття існуючої цифрової схеми;
- процес побудови нової цифрової схеми;
- процес роботи з базою даних елементів та мікросхем, з яких складається цифрова схема.

Розглянемо взаємодію цих процесів більш детально.

Процес відкриття існуючої цифрової схеми взаємодіє з процесом діагностики цифрової схеми.

Цей процес у свою чергу взаємодіє з наступними процесами:

- процес генерування тестових послідовностей;
- процес роботи осцилографа;
- процес побудови цифрової схеми.

Останній процес у свою чергу взаємодіє з наступними процесами:

- процес редагування цифрової схеми;
- процес збереження цифрової схеми;
- процес вибору мікросхеми;
- процес вибору елемента.

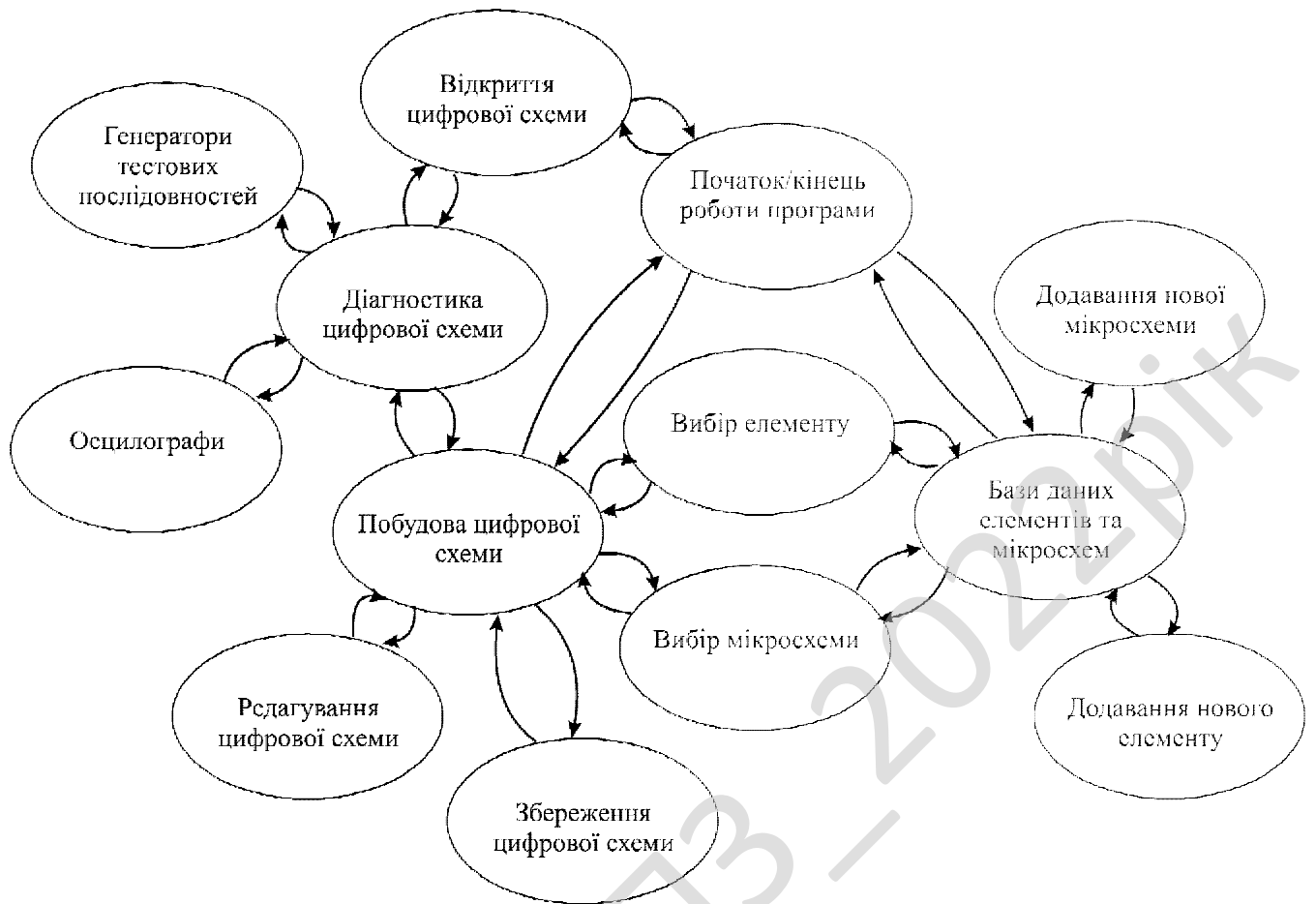


Рисунок 3.5 – Діаграма процесів системи

Останні два процеси взаємодіють з процесом роботи з базою даних елементів та мікросхем.

Цей процес взаємодіє також з процесами додавання нової мікросхеми та процесом додавання нового елементу.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

На рисунку 4.1 зображена блок-схема основної програми. З неї ми бачимо, що програма виконується наступним чином.

Спершу відкривається головне віно програми. Під час відкриття головного вікна автоматично відвантажуються бази даних елементів та бази даних мікросхем.

Після завантаження головного вікна та підключення баз даних існує можливість доступу до наступних блоків програми:

- відкриття готової цифрової схеми;
- перегляд баз даних;
- побудова цифрових схем;
- діагностика;
- збереження цифрової схеми.

Якщо переходимо до блоку відкриття цифрової схеми, то відбуваються наступні дії:

- спершу вибирається файл;
- після цього завантажується цифрова схема з файлу;
- за цим виводиться відкрита цифрова схема.

Якщо працюємо з блоком перегляду баз даних, то відбуваються наступні кроки програми. Спершу виводиться вміст баз даних. Після цього користувач вибирає дію: або додати на схему елемент з бази даних, або не додавати елемент з бази даних. Якщо елемент додається то спершу виводяться нові елементи, а потім відбувається дія додавання нових елементів на цифрову схему.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

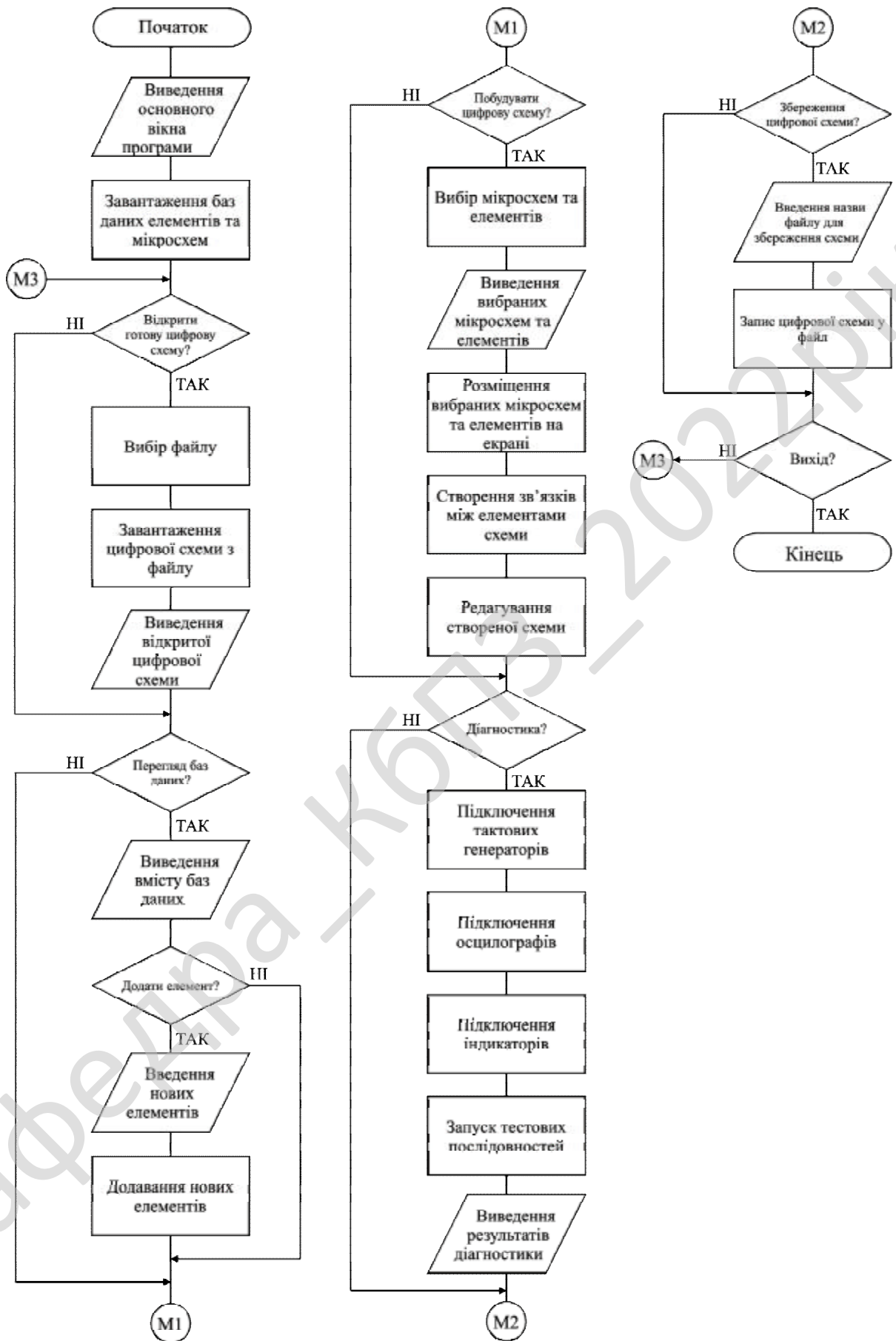


Рисунок 4.1 – Блок-схема основної програми

Якщо вибирається блок побудови цифрової схеми то виконуються наступні дії.

Спершу вибираються мікросхеми та елементи. Після цього обрані елементи та мікросхеми розміщуються на екрані.

Наступним кроком є створення зв'язків між елементами схеми.

Й останньою дією є редагування створеної схеми.

Якщо обирається блок діагностики, то відбуваються наступні дії:

- Підключаються тактові генератори.
- Після цього підключається осцилограф, або осцилографи.
- Далі відбувається підключення індикаторів.
- За цим запускається тестова послідовність.
- Результати діагностики та тестування виводяться на екран.

Якщо вибирається дія збереження цифрової схеми, то користувачем вводиться назва файлу для збереження мікросхеми, після чого відбувається запис цифрової схеми у файл.

Закладка «Вихід» дає можливість вийти з програми.

Наведемо частину коду, який ілюструє деякі функції та реалізує дії описані вище.

```
//ініціалізація параметрів програми
Procedure TForm1.InitCfg;
begin
  IniFile := Tinifile.Create
  (ExtractFilePath(Application.ExeName)+'\config\main.ini'); //відкриття файлу з
  параметрами програми
  Desktop_Color := IniFile.ReadInteger('Colors','Desktop',clWhite); //колір фону
  Grid_Color := IniFile.ReadInteger('Colors','Grid',clBlack); //колір розмітки
  Selection_Metod := IniFile.ReadBool('Selection','Full',False);
  //параметри виділення елементів
  Select_on_creation := IniFile.ReadBool('Creation','Selection',False);
  Draw_connections := IniFile.ReadBool('Selection','Draw_conn',False);
  Select_connections := IniFile.ReadBool('Selection','Sel_conn',True);
  if IniFile.ReadBool('Main','LockBases',True) then
  begin
    N11.Enabled:=false;
```

```

        N12.Enabled:=false;
        N13.Enabled:=false;
        N14.Enabled:=false;
    end;
end;
//збереження параметрів програми
Procedure TForm1.SaveCfg;
begin
    IniFile.WriteInteger('Colors','Desktop',Desktop_Color);
    IniFile.WriteInteger('Colors','Grid',Grid_Color);
    IniFile.WriteBool('Selection','Full',Selection_Metod);
IniFile.WriteBool('Creation','Selection',Select_on_creation);
IniFile.WriteBool('Selection','Draw_conn',Draw_connections);
IniFile.WriteBool('Selection','Sel_conn',Select_connections);
end;
Procedure TForm1.DesktopSetup;
var i, j : integer;
begin
    for i:=0 to 200 do for j:=0 to 200 do Field[i,j]:=0;
    Frame21.Imagel.Canvas.Brush.Style := bsSolid;
    Frame21.Imagel.Canvas.Pen.Mode := pmCopy;
    Frame21.Imagel.Canvas.Pen.Color := Desktop_Color;
    Frame21.Imagel.Canvas.Brush.Color := Desktop_Color;
    Frame21.Pane1.Width:=2000*Koeff_Mashtab div 10;
    Frame21.Pane1.Height:=2000*Koeff_Mashtab div 10;
    Frame21.Imagel.Canvas.Rectangle(0,0,2000*Koeff_Mashtab div 10,2000*Koeff_Mashtab
div 10);
    for i:=0 to 200 do
        for j:=0 to 200 do
Frame21.Imagel.Canvas.Pixels[i*Koeff_Mashtab,j*Koeff_Mashtab]:=Grid_Color;
// Button4Click(nil);
end;
//видалення виділених елементів
procedure DeleteSelectedElements;
var i, j, k, l : integer;
begin
    i := 1;
    while i <= Last_Elem do
        begin
            if Elements[i].Selected then
                begin
                    Elements[i].destroy;

```

					ВКРМ-123.22.0007.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		46

```

for j:=1 to 5 do
  if Oscill[i]=i then Oscill[i]:=0;
if Form17.Visible then Form17.CheckListBox1ClickCheck(nil);
for j:=0 to High(Conns) do
begin
  k:=0;
  while k<=High(Conns[j]) do
  begin
    if Conns[j][k].Elem=i then
      begin
        for l:=k to High(Conns[j])-1 do
          Conns[j][l]:=Conns[j][l+1];
        k:=k-1;
        SetLength(Conns[j],High(Conns[j]));
      end;
      k:=k+1;
    end;
  end;
  j:=0;
while j<=High(Conns) do
begin
  if High(Conns[j])<1 then
    begin
      for l:=j to High(Conns)-1 do
        Conns[l]:=Conns[l+1];
      SetLength(Conns,High(Conns));
    end;
    j:=j+1;
  end;
for j:=0 to High(Conns) do
  for k:=0 to High(Conns[j]) do
    if Conns[j][k].Elem>=i then Conns[j][k].Elem:=Conns[j][k].Elem-1;
  for j:= i to Last_Elem-1 do
    begin
      Elements[j] := Elements[j+1];
      Elements[j].Image.Tag := Elements[j].Image.Tag -1;
    end;
  dec(Last_Elem);
  dec(i);
end;
inc(i);
end;

```

					БКРМ-123.22.0007.00.00.ПЗ	<i>Арк.</i>
<i>Вим.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		47

```

end;
//видалення елементів
procedure DeselectElements;
var i :integer;
begin
    for i:=1 to Last_Elem do
        Elements[i].Deselect;
end;
procedure SelectAllElements;
var i : integer;
begin
    for i:=1 to Last_Elem do
        Elements[i].Select;
end;
procedure InversElementSelection;
var i: integer;
begin
    for i:=1 to Last_Elem do
        Elements[i].Swich_Selection;
end;

//промальовування виділення елементів мишкою
Procedure TForm1.DrawSelectionRectangle (SP : TPoint);
begin
    Frame21.Image1.Canvas.Brush.Style := bsClear;
    Frame21.Image1.Canvas.Pen.Color := clWhite;
    Frame21.Image1.Canvas.Pen.Style := psDash;
    Frame21.Image1.Canvas.Pen.Mode := pmXor;
    Frame21.Image1.Canvas.Pen.Width := 1;
    Frame21.Image1.Canvas.Rectangle(Base_Selection_Point.X,Base_Selection_Point.Y,SP.X,SP.Y);
end;
procedure Draw_Leg(Sender:TObject; i:integer);
begin
    TImage(Sender).Canvas.Pen.Mode:=pmXor;
    TImage(Sender).Canvas.Pen.Width:=10;
    TImage(Sender).Canvas.Pen.Color:=clRed xor Form1.Frame21.Image1.Canvas.Pixels[1,1];
    if Elements[TImage(Sender).Tag].Legs[i].x=0 then begin
    TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Elements[TImage(Sender).Tag].Legs[i].y*10);
    TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Elements[TImage(Sender).Tag].Legs[i].y*10);
end;

```

					БКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```

end else begin
TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);
TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);
end;
end;

//з'єднання елементів
function IsConnected(E,L: Integer): Boolean;
var i,j: Integer;
begin
Result:=true;
for i:=0 to High(Conns) do
for j:=0 to High(Conns[i]) do
if (Conns[i][j].Elem=E) and (Conns[i][j].Leg=L) then exit;
Result:=false;
end;
end;
procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var selcon: Boolean;
i: Integer;
begin
if (Button <> mbLeft) and (Button <> mbRight) then exit;
if Connect_Status then exit;
selcon:= false;
i:=1;
if TImage(Sender).tag=0 then exit;
if (x<Koeff_Mashtab) or (x>(Elements[TImage(Sender).tag].Size.x-1)*Koeff_Mashtab)
then
for i:=0 to High(Elements[TImage(Sender).tag].Legs) do
if (x>(Elements[TImage(Sender).tag].Legs[i].x-1)*Koeff_Mashtab)
and (x<(Elements[TImage(Sender).tag].Legs[i].x+1)*Koeff_Mashtab)
and (y>(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab-Koeff_Mashtab
div 2)
and
(y<(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab+Koeff_Mashtab div 2) then
begin selcon:=true; break; end;
if Button = mbRight then
begin
if selcon and IsConnected(TImage(Sender).tag,i) then
begin

```

									ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата						49

```

TImage(Sender).PopupMenu:=PopupMenu3;
Connect_Start.Elem:=TImage(Sender).tag;
Connect_Start.Leg:=i;
end else TImage(Sender).PopupMenu:=nil;
exit;
end;
if selcon then begin
    Connect_Status:=true;
    Connect_Start.Elem:=TImage(Sender).tag;
    Connect_Start.Leg:=i;
    Legs_sel2:=false;
end;

M_x:=X;
M_y:=y;
if not (ssCtrl in Shift) then Mouse_Status:=True;
end;
//додавання з'єднання
procedure AddConnect(E1,L1,E2,L2: integer);
label 1,2;
var i,j,k:integer;
begin
    for i:=0 to High(Conns) do
        for j:= 0 to High(Conns[i]) do
            if (Conns[i][j].Elem=E1) and (Conns[i][j].Leg=L1) then goto 1;
1:
        for k:=0 to High(Conns) do
            for j:= 0 to High(Conns[k]) do
                if (Conns[k][j].Elem=E2) and (Conns[k][j].Leg=L2) then goto 2;
2:
            if ((i=High(Conns)+1) and (k=High(Conns)+1)) or (High(Conns)=-1) then
                begin
                    SetLength(Conns,High(Conns)+2);
                    SetLength(Conns[High(Conns)],2);
                    j:=0;
                    Conns[High(Conns)][j].Elem:=E1;
                    Conns[High(Conns)][j].Leg:=L1;
                    j:=1;
                    Conns[High(Conns)][j].Elem:=E2;
                    Conns[High(Conns)][j].Leg:=L2;
                    InitTrace;
                    Exit;
                end;
end;

```

					БКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

```

if k=i then exit;
if i=High(Conns)+1 then
  begin
    SetLength(Conns[k],High(Conns[k])+2);
    j:=High(Conns[k]);
    Conns[k][j].Elem:=E1;
    Conns[k][j].Leg:=L1;
    InitTrace;
    Exit;
  end;
if k=High(Conns)+1 then
  begin
    SetLength(Conns[i],High(Conns[i])+2);
    j:=High(Conns[i]);
    Conns[i][j].Elem:=E2;
    Conns[i][j].Leg:=L2;
    InitTrace;
    Exit;
  end;
SetLength(Conns[i],High(Conns[i])+2+High(Conns[k]));
for j:=High(Conns[i])-High(Conns[k]) to High(Conns[i]) do
  begin
    Conns[i][j].Elem:=Conns[k][j-High(Conns[i])+High(Conns[k])].Elem;
    Conns[i][j].Leg:=Conns[k][j-High(Conns[i])+High(Conns[k])].Leg;
  end;
SetLength(Conns[k],0);
for j:=k to High(Conns)-1 do
  Conns[j]:=Conns[j+1];
SetLength(Conns[j],0);
SetLength(Conns,High(Conns));
InitTrace;
end;

```

На рисунку 4.2 зображена блок-схема підпрограми контролю цифрової схеми з локалізацією несправності до першої послідовності, що містить помилки. З неї ми бачимо, що підпрограма виконується наступним чином.

Спершу відбувається аналіз послідовностей в аналізаторі та визначається сигнатура.

Після цієї дії відбувається обчислення еталонного значення сигнатури.

Якщо сигнатури збігаються то вважається, що несправності немає.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

У іншому випадку відбуваються наступні дії:

- Вся множина вхідних послідовностей розбивається на 2 групи.
- Лічильник дорівнюється одиниці.
- У результаті аналізу реальних послідовностей, номери яких задаються

визначається значення сигнатури.

- Відбувається обчислення еталонного значення сигнатури.
- Отримані сигнатури порівнюються з еталонами.
- Якщо вони дорівнюються то елементи множин міняються місцями.
- Лічильник збільшується на одиницю.
- Якщо лічильник менше кількості послідовностей, то множина ділиться

на дві підмножини й відбувається перехід до розрахунку сигнатури.

– У протилежному випадку єдиний елемент множини, являє собою номер помилкової послідовності.

- Результати виводяться на екран.
- Підпрограма закінчує свою роботу.

Алгоритм контролю цифрової схеми з локалізацією несправності до першої послідовності, що містить викликані нею помилки:

1. В результаті аналізу $n=2d$ реальних послідовностей $\{y^*(k)\}, v=1, n$ на n -канальному аналізаторі визначається значення сигнатури $S^*(x)$, що відповідає співвідношенню:

$$\sum_{v=1}^n \oplus \sum_{k=1}^l \oplus y^*_v(k)x^{(k-1)n+v-1} = g^*(x)\phi^{-1}(x) \oplus S^*(x)$$

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

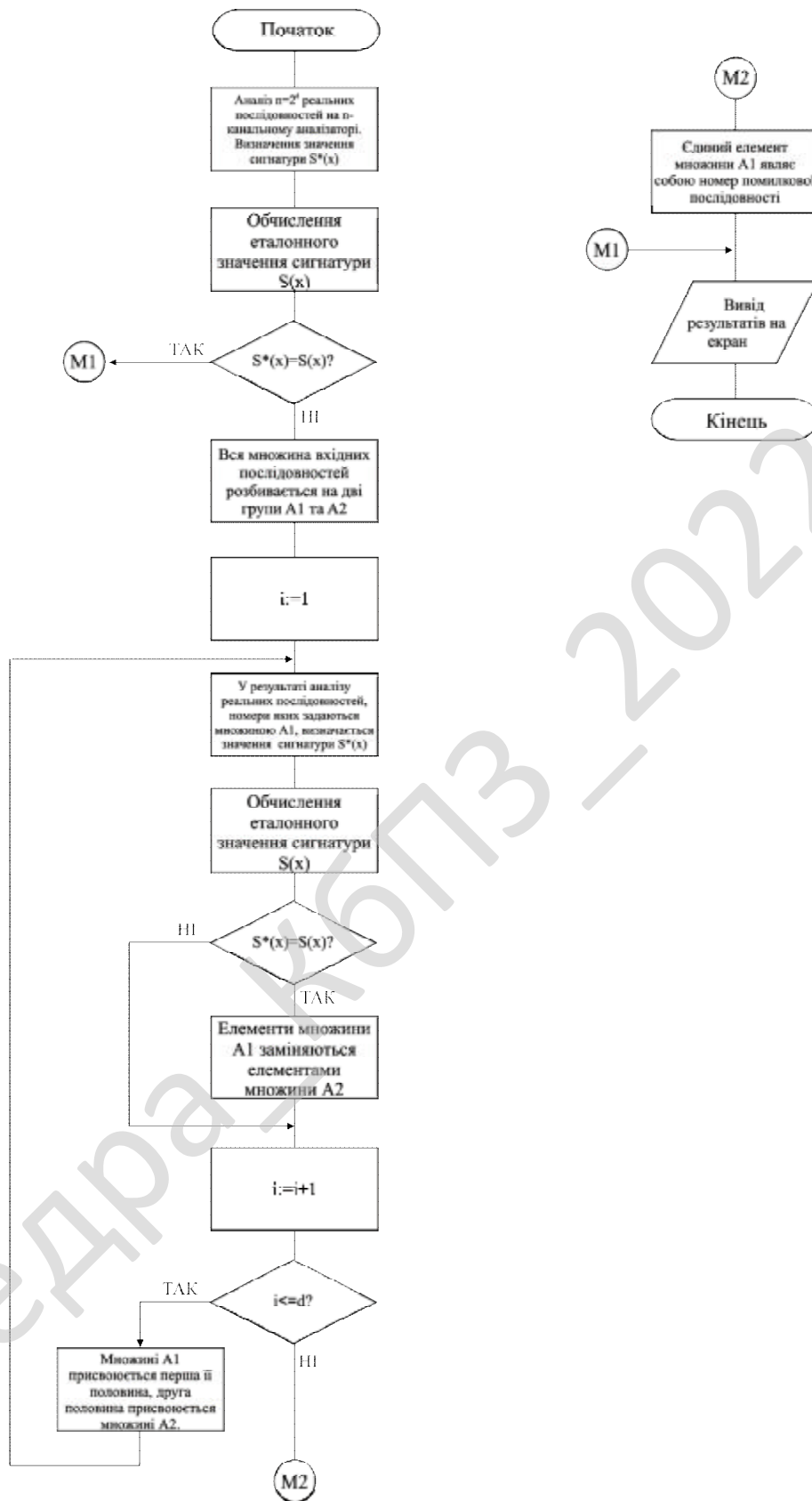


Рисунок 4.2 – Блок-схема підпрограми контролю цифрової схеми з локалізацією несправності до першої послідовності, що містить помилки

2. Обчислення еталонного значення сигнатури $S(x)$ за формулою

$$S(x) = \sum_{v=1}^n \oplus S_v(x)$$

3. Реальне значення сигнатури $S^*(x)$ порівнюється з еталонною сигнатурою $S(x)$. У випадку виконання рівності $S^*(x)=S(x)$ переходять до виконання п.11 і процедура контролю вважається кінченою. У протилежному випадку, коли $S^*(x) \neq S(x)$, виконується наступний етап алгоритму.

4. Вся множина вхідних послідовностей розбивається на дві групи, причому номери послідовностей $\{y_1(k)\}, \{y_2(k)\}, \{y_3(k)\}, \dots, \{y_{n/2}(k)\}$ становлять множину $A_2 = \{1, 2, 3, \dots, n/2\}$, а послідовностей $\{y_{n/2+1}(k)\}, \{y_{n/2+2}(k)\}, \dots, \{y_n(k)\}$ – множину $A_2 = \{n/2+1, n/2+2, \dots, n\}$; величині i привласнюється значення 1.

5. У результаті аналізу реальних послідовностей, номери яких задаються множиною A_1 , на n -канальному аналізаторі за умови, що послідовності, номери яких не визначені множиною A_1 , є нульовими, визначається значення реальної сигнатури $S^*(x)$.

6. На підставі виразу $S(x) = \sum_{j \in A_1} \oplus S_j(x)$ одержуємо $S(x)$.

7. Перевіряється справедливість рівності $S(x)=S^*(x)$. У випадку його виконання елементи множини A_1 замінюються елементами множини A_2 .

8. Значення змінної i збільшується на одиницю. Потім його величина порівнюється з величиною d . При $i \leq d$ переходять до наступного пункту алгоритму, у протилежному випадку виконується пункт 10.

9. За поточним значенням множини A_1 формуються нові множини A_1 і A_2 . Новими елементами множини A_1 стає перша половина її поточних елементів, друга половина привласнюється множині A_2 . Після визначення множин A_1 і A_2 переходять до виконання п. 5.

10. Єдиний елемент множини A_1 являє собою номер помилкової послідовності, яка формується на одному з полюсів досліджуваної схеми.

11. Процедура контролю цифрової схеми вважається закінченою.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

4.2 Захист розробленого програмного забезпечення

Дані які використовуються у даній роботі захищаються алгоритмом ДСТУ 7564:2014 («Купина»). В Україні на основі консервативного підходу із залученням відомих і добре досліджених конструкцій була розроблена геш-функція, що базується на новому блоковому шифрі „Калина” (ДСТУ 7624:2014).

Національний стандарт ДСТУ 7564:2014 визначає криптографічну функцію гешування „Купина”, додатковий режим її застосування для формування коду автентифікації повідомлення (імітовставки), а також значення для перевірки реалізацій.

Для скорочення обсягу тексту національного стандарту була застосована математична нотація, що дозволяє отримати точний і компактний запис.

Водночас, такий підхід може ускладнювати сприйняття сутності алгоритму для фахівців, що не мають фундаментальної криптологічної освіти.

У роботі наводиться розгорнутий альтернативний опис функції гешування „Купина” із позначеннями, традиційними для галузі комп’ютерних наук.

Термінологія та позначення

Вектор ініціалізації – бітова послідовність фіксованої довжини (512 або 1024 біта), що використовується як початкове значення при обчисленні геш-значення.

Внутрішній стан – бітова послідовність фіксованої довжини (512 або 1024 біта), що є проміжним значенням на кожній ітерації перетворення функції гешування, а також вхідним та вихідним значеннями перетворень P і Q ; для цих перетворень внутрішній стан подається як матриця розміром $8 \times s$ байт.

Геш-значення (геш-вектор) – бітова послідовність фіксованої довжини ($n = 8 \cdot s, s \in \{1, 2, \dots, 64\}$), що є результатом роботи функції гешування.

Доповнення – вставка додаткових біт у кінець повідомлення для отримання кратності довжини бітової послідовності довжині внутрішнього стану функції гешування.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Повідомлення – бітова послідовність довжини від 0 біт (порожній рядок) до $2^{96}-1$ біт.

Функція стиснення – ітеративне перетворення, що відображає l -бітний блок повідомлення та l -бітне значення, отримане функцією стиснення на попередньому кроці, у нове l -бітне значення.

Далі використовуються наступні позначення:

- \oplus – додавання за модулем 2 (XOR);
- $0x$ – префікс числа, що записане у шістнадцятковій системі числення;
- $a \bmod b$ – ціле невід'ємне число, що дорівнює залишку від ділення цілого числа a на натуральне число b ;
- B_i – i -й байт вхідної послідовності;
- C^i – константа перетворення XORRoundKey або Add64RoundKey для i -го циклу;
- c – кількість стовпців внутрішнього стану в матричному поданні;
- φ – функція стиснення;
- H – визначена у стандарті функція гешування;
- $H(M)$ – результат обчислення функції гешування для повідомлення M (гешзначення);
- IV – вектор ініціалізації;
- l – розмір внутрішнього стану функції гешування (у бітах), $l \in \{512, 1024\}$;
- M – повідомлення;
- m_i – i -й блок повідомлення M ;
- n – довжина обчисленого геш-значення;
- N – довжина повідомлення M без доповнення;
- P, Q – складові перетворення функції стиснення;
- P_{512} – перетворення P для 512-бітного внутрішнього стану;
- P_{1024} – перетворення P для 1024-бітного внутрішнього стану;
- Q_{512} – перетворення Q для 512-бітного внутрішнього стану;

									Арк.
Вим.	Арк.	№ докум.	Підпис	Дата	ВКРМ-123.22.0007.00.00.ПЗ				56

- Q_{1024} – перетворення Q для 1024-бітного внутрішнього стану;
- r – кількість ітерацій у перетвореннях P і Q ($r = t$ в ДСТУ 7564:2014);
- S – внутрішній стан геш-функції;
- t – кількість блоків m , з яких складається повідомлення M , включаючи доповнення;
- v_i – i -й біт вхідної послідовності;
- Ω – завершальне перетворення;
- Купина- n – режим використання функції гешування з усіченням обчисленого гешзначення до розміру n біт.

Загальні положення

Під функцією гешування H розуміється залежне від вектора ініціалізації IV відображення послідовності біт M у геш-значення $H(M)$ фіксованої довжини n .

ДСТУ 7564:2014 визначає функцію гешування, яка виконує перетворення «Купина-256» або «Купина-512», що забезпечують обчислення геш-значення з довжинами 256 або 512 біт відповідно.

Геш-значення довжиною 256 бітів додатково може бути усічено до бітової послідовності довжиною від 8 до 248 біт з кроком у 8 біт, 512 бітів може бути усічене до бітової послідовності довжиною від 264 до 504 біт з кроком у 8 біт.

Режим роботи для формування геш-значення довжиною n біт позначається як «Купина- n ».

Основними режимами роботи функції гешування, що рекомендуються до застосування, є «Купина-256», «Купина-384» і «Купина-512».

Структура перетворення

Функція гешування, визначена в ДСТУ 7564:2014, формує геш-значення для повідомлення, що складається з бітової послідовності довжини від 0 біт (порожній рядок) до $2^{96}-1$ біт.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

При формуванні геш-значення повідомлення доповнюється, далі поділяється на l -бітні блоки m_0, \dots, m_t , після чого виконується обробка кожного блоку шляхом ітеративного виконання функції стиснення φ .

При цьому формуються значення $h_i = \varphi(h_{i-1}, m_i)$ де $i = 1, \dots, t$, а початкове значення $h_0 = IV$.

Після обробки останнього блоку повідомлення результуюче геш-значення обчислюється як $H(M) = \Omega(h_t)$, де Ω – завершальне перетворення, що повертає n -бітне значення, кратне 8 ($0 < n \leq l/2$).

Кафедра КБПЗ – 2022 рік

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення реалізує методи моделювання та діагностики цифрових схем.

Програмно-апаратні вимоги:

- Загальний обсяг ОЗП: 128 Мбайт.
- Вільний простір на жорсткому диску: 5 Мбайт.
- Операційна система Microsoft Windows 10/11.

Головне вікно програми зображене на рисунку 5.1.

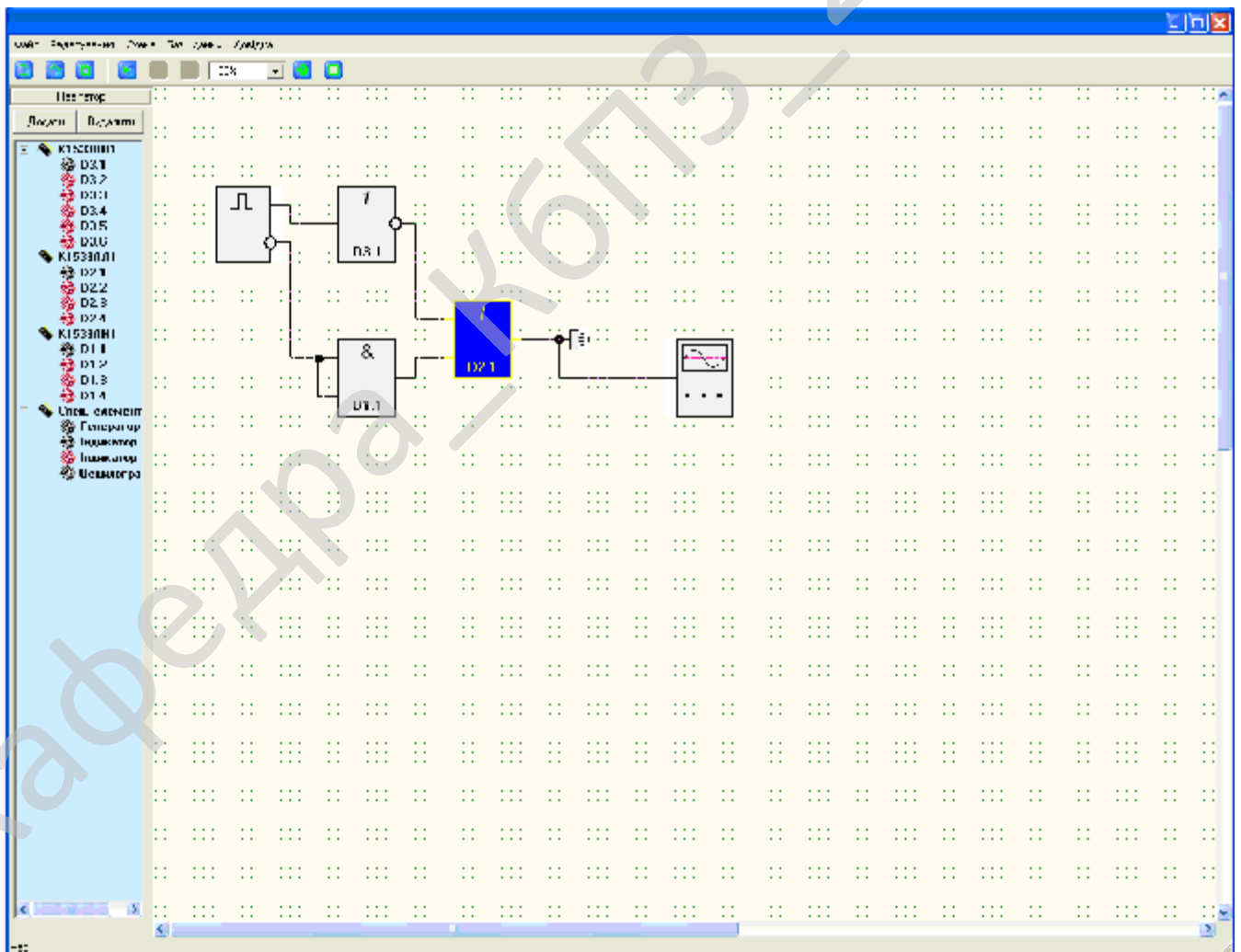


Рисунок 5.1 – Основне вікно програми

З рисунку видно, що з лівої сторони розміщений навігатор, в якому можна додавати та видаляти елементи та мікросхеми. Основну частину вікна займає поле конструктора, в якому відбувається побудова цифрових схем з елементів та мікросхем відкритих у навігаторі. Вгорі вікна розміщене меню користувача та панель інструментів.

Для додавання елементів та мікросхем у навігатор слід використовувати кнопку «Додати», для видалення «Видалити». При натисненні кнопки «Додати» з'являється випадаюче меню з двома пунктами «Спеціальний елемент» та «Мікросхема». Елементи перелічені як підпункти пункту «Спеціальний елемент». Список мікросхем виводиться в окремому вікні.

Для побудови цифрової схеми, необхідно по черзі натискати в навігаторі на потрібний елемент, після чого він буде з'являтися у полі конструктора та переміщувати його у потрібне місце.

Елементи можна створювати й розміщати в будь-якому порядку, а також додавати й видаляти у вже створеної ЦС.

Щоб з'єднати входи й виходи елементів лінією, необхідно виконати наступні дії.

При наведенні курсору миші на вхідну ніжку елемента, ніжка виділяється й натискаючи ліву кнопку тягнемо до вихідної ніжки іншого елемента. Як тільки вихідна ніжка яку ми хочемо з'єднати також виділиться відпускаємо кнопку миші. У результаті буде створена лінії, що з'єднує вхідну й вихідну ніжки.

Також з'єднується лінією точка із вхідними й вихідними ніжками елементів.

Для перегляду бази даних та додавання до неї нових елементів слід скористатися меню «Бази даних» та вибрати в ньому пункт «Елементи» або «Мікросхеми» в залежності від того, яку базу даних необхідно переглянути. На рисунку 5.2 показаний приклад бази даних мікросхем.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

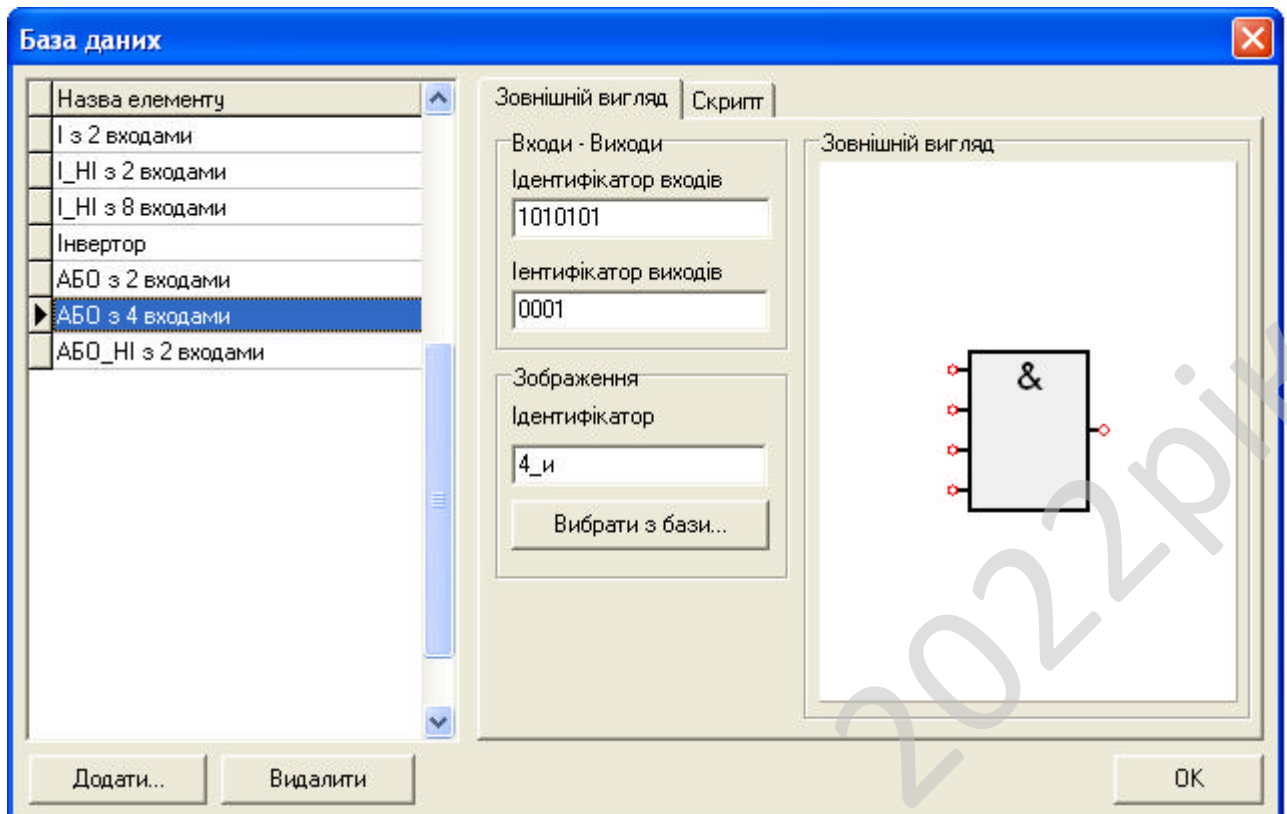



Рисунок 5.2 – Вікно бази даних

Для здійснення діагностики цифрової схеми слід під'єднати до неї генератор тестової послідовності, індикатори та осцилографи. Після цього слід натиснути кнопку  на панелі інструментів.

Якщо в схемі використовується генератор М-послідовності, то для нього необхідно ввести примітивний неприводимий поліном. Для цього у властивостях генератору слід ввести поліном (рисунок 5.3).

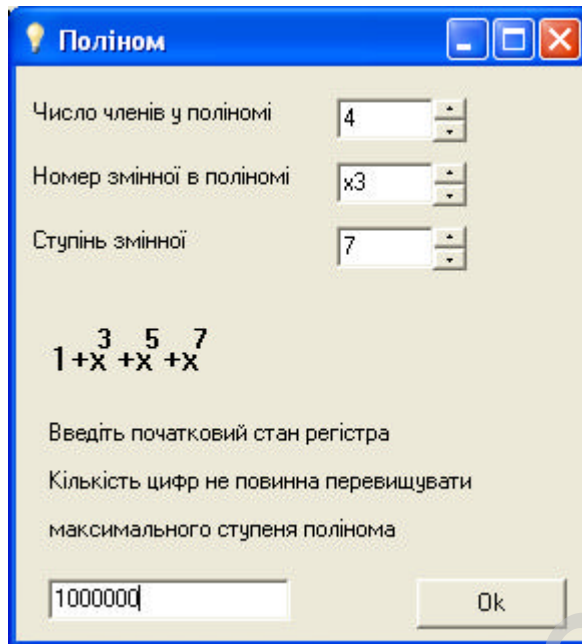


Рисунок 5.3 – Введення поліному $\varphi(x) = 1 + x^3 + x^5 + x^7$ для генератора тестових послідовностей

Приклад процесу тестування зображений на рисунку 5.3.

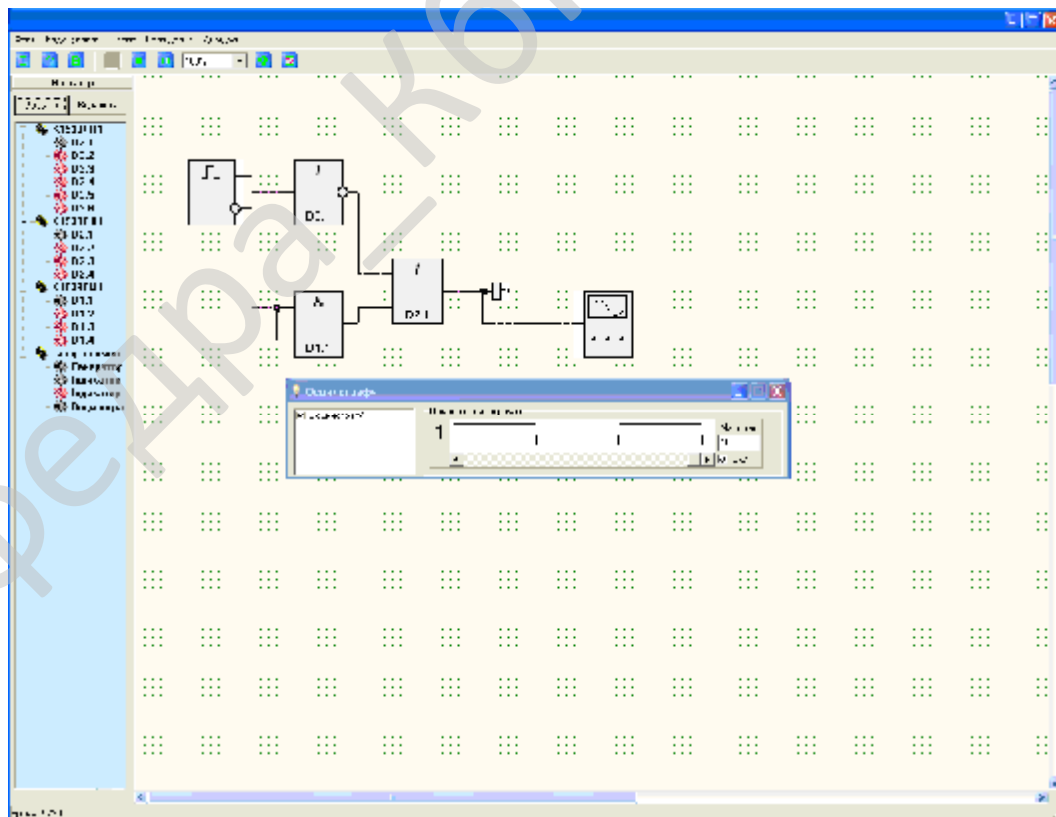


Рисунок 5.4 – Діагностика цифрової схеми

Для зміни параметрів програми слід вибрати пункт «Параметри» у меню «Файл», після чого з'явиться вікно зображене на рисунку 5.5.

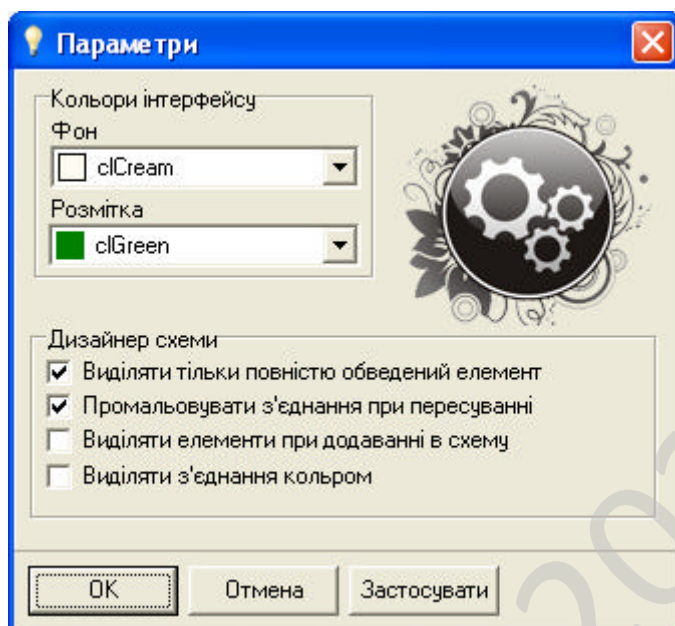


Рисунок 5.5 – Вікно зміни параметрів програми

Коротку довідку про розроблену програму та її автора можна переглянути вибравши підпункт «Про програму...» з меню «Довідка», після чого з'явиться вікно зображене на рисунку 5.6.

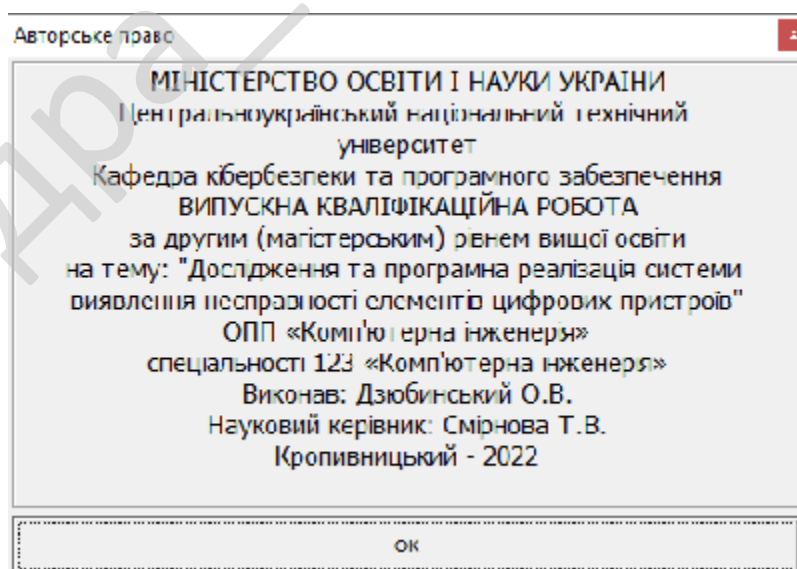


Рисунок 5.6 – Довідка про програму

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи виявлення несправності елементів цифрових пристроїв.

Метою розробки є дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

Об'єктом дослідження є процес виявлення несправності елементів цифрових пристроїв.

Предметом дослідження є методи виявлення несправності елементів цифрових пристроїв.

Методи дослідження базуються на методах системотехніки, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод виявлення несправності елементів цифрових пристроїв.

– Розроблено вітчизняний продукт виявлення несправності елементів цифрових пристроїв, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 48 днів (два місяці).

В магістерській роботі було проведено дослідження та розроблене програмне забезпечення системи виявлення несправності елементів цифрових пристроїв.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність

Таблиця 7.1 – Початкові данні

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт	N	1
2. Кількість екземплярів програм, шт	Ne	130
3. Запланований термін розробки, днів	Fpq	48 (2 місяці)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Б
6. Складність алгоритму (1, 2, 3)	–	2
7. Кількість макетів вхідної інформації	–	3

Продовження табл. 7.1

1	2	3
8. Кількість форм вихідної інформації.	–	4
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	2
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	2
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	2
17. Складність кінцевого програмного продукту (1-6)	–	2
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	2
20. Вимоги до швидкодії ПП (1-6)	–	2
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	2
23. Професійний рівень аналітиків (1-6)	–	2
24. Професійний рівень програмістів (1-6)	–	2
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	2
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження табл. 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	2
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн	–	13000
33. Норматив додаткової зарплати, % :	Нд	10
34. Норматив відрахувань у соціальні фонди, %	Нс	22
35. Норматив загальногосподарських витрат, %	Нг	15
36. Норматив витрат на освоєння нових мов програмування, %	Нп	15
37. Рівень рентабельності програмної продукції, %	Ре	50
38. Ставка податку на додану вартість, %	Ндв	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B \quad (7.1)$$

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

де A – коефіцієнт Боєма, $A=2,45$; $Size$ – загальний об'єм відлагодженого програмного коду, тис. рядків; B – показник ступеня, що визначається співвідношенням

$$B = 1,01 + 0,001 \sum W_i \quad (7.2)$$

де W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 3,38 + 3,95 + 2,73) = 1,026$$

$$T_{ном} = 2,45 \cdot 2,7^{1,026} = 6,78 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 6,78 \cdot (0,88 \cdot 0,93 \cdot 0,88 \cdot 0,91 \cdot 0,95 \cdot 1 \cdot 1 \cdot 0,87 \cdot 1,22 \cdot 1,16 \cdot 1,1 \cdot 1,1 \cdot 1,12 \cdot 1,1 \cdot 1 \cdot 1,1 \cdot 1,1) = 9,37 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3CT_{уточн}^{0,33+0,2(B-1,01)}S, \quad (7.4)$$

де S – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПО згідно встановленим вимогам. Вибираємо в межах (25...350)%

$$T_{РП} = 0,3 \cdot 2,66 \cdot 9,37^{0,33+0,2(1,026-1,01)} \cdot 80 = 135 \text{ люд/день}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	9	Д7
Робочий проект	135	Ф 7.1-7.4
Впровадження	13	Д13
Всього	176	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де F_{pq} – плановий фонд робочого часу одного спеціаліста, днів,

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні,

$$Ч = \frac{176 \cdot 1}{48 \cdot 5} = 4 \text{ ставки}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	8	720	12
Монітор	60	8	480	8
Клавіатура	30	8	240	4
Маніпулятор «мишка»	30	8	240	4
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	1	120	2
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	2	60	1
Кабельні господарства ЛОМ на 1 м. п.	2,5	200	500	8,33
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	42,99

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2} \quad (7.6)$$

$$\Phi_{op}^c = \frac{43 \cdot 2}{1,2} = 72 \text{ год}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{op}^c}{F_{op} \cdot T_{зм}} \quad (7.7)$$

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

$$Ч_{ел} = 72 / (48 \cdot 8) = 0,2 \text{ ставки}$$

Для забезпечення нормального технічного обслуговування засобів ТО та мереж, необхідно прийняти найбільше ціле значення розрахункової чисельності інженерів – електронщиків.

Чисельність інженерів-системотехніків, адміністраторів мережі, дизайнерів WEB вузлів, системних програмістів (аналітиків), бухгалтерів-економістів визначається за потребою в залежності від функціональних обов'язків. Після визначення чисельності персоналу складається штатний розклад.

Таблиця 7.4 – Розрахунок чисельності штатного персоналу сектору системного та адміністративного обслуговування засобів ОТ та комп'ютерних мереж

Посада	Вид роботи	Час	Кількість штатних одиниць
Адміністратор загальної мережі, аналітик	Адміністрування локальної мережі, поштового та серверу DNS (ОС FreeBSD), маршрутизатора Cisco, доменного контролеру Windows Server, серверу доступу АДСЛ (ОС Linux), налаштування ADSL, VPN, PPPoE, Frame Relay, Wi-Fi	0,2	0,1
	Налаштування і конфігурування базової станції безпроводного зв'язку (СМТS)	0,2	
	Розробка та впровадження проектів з організації зв'язку між віддаленими об'єктами, ЛОМ	0,2	
	Забезпечення цілодобової роботи зв'язку клієнтів до мережі Інтернет	0,2	
Всього		0,8	

Продовження табл. 7.4

Посада	Вид роботи	Час	Кількість штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,4
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	1	
	Контроль взаєморозрахунків з постачальниками	0,2	
Всього		3,2	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	0,2	0,1
	Створення графічних і стилістичних елементів сайту	0,2	
	Оформлення банерів і промо-сторінок	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		0,8	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	0,2	0,1
	Верстка друкованих видань	0,2	
	Додрукова підготовка макетів	0,2	
	Розміщення графіки і контенту на Інтернет сторінках	0,2	
Всього		0,8	

Складемо штатний розклад виконавців:

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньо-місячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	11296	22592
Продакт-менеджер	0,4	9000	7200
Інженер-програміст	4	10200	81600
Інженер-електронщик	0,2	8500	3400
Інженер-системотехнік	0,1	8500	1700
Адміністратор мережі	0,1	8500	1700
Системний програміст	0,1	8500	1700
Дизайнер WEB	0,1	8500	1700
Інженер-верстальник	0,1	8500	1700
Бухгалтер-економіст	0,1	8500	1700
Всього за період розробки	$R_{cn}=6,2$	-	$\Phi_{роб}=124992$

Розрахуємо середньоденну зарплату одного виконавця:

$$Z_{сд} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$Z_{сд} = \frac{124992}{6,2 \cdot 48} = 420 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

$$B_{y\delta} = R_{cn}^1 S_y C_{nl}, \quad (7.9)$$

де R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 8 робочих місць.

S_y – питома площа на одне робоче місце, m^2 ,

C_{nl} – вартість одного квадратного метра площі, грн.

Згідно даних інтернет ресурсу DOM.RIA (<https://dom.ria.com>) ціна одного квадратного метра площі, вік якої не перевищує 30 років, по місту складає 500...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 38 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно $8 m^2$. З урахуванням цього:

$$B_{y\delta} = 8 \cdot 8 \cdot 20000 = 1280000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 128000 грн.

Балансова вартість інвентарю розраховується за нормою 3500 грн на одне робоче місце. Тобто

$$I_{nb} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де C_m – ціна меблів для одного робочого місця, грн.

$$I_{nb} = 8 \cdot 3500 = 28000 \text{ грн}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Комп'ютерторг за 29.10.22 – джерело

<http://computorg.ua/ru/price.html>

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		12721
Системний блок		7721
Процесор	Intel Core i5-4570 (4 ядра по 3.2 – 3.6 GHz); 6 MB Cache Memory	-
Системна плата	ASUS B85M-K Socket 1150 Intel B85 OEM Refurbished (SATA II – 2 шт, SATA-3 – 4 шт, 4x USB 3.0, 6x USB 2.0, 4x Audio Ports, LAN (RJ-45), 2x PS/2, Com Port, 2x DP, VGA)	-
Відеокарта	Інтегрована Intel HD Graphics 4600	-
Жорсткий диск	SSD 120Gb + HDD 500 Gb SAMSUNG Barracuda HD502HJ	-
Оперативна пам'ять	DIMM 4096Mb DDR3 PC3-12800 Kingston, 1600MHz, 512M x 64, CL9-9-9-27, 1.65V, w/heatsink, HyperX	-
DVD-привод	DVD±RW ASUS DRW-24B5ST Black Bulk	-
Корпус	Logicpower 8702 – 550w 12cm, 440x180x445, 2 USB 3.0+2 USB 2.0 на передній панелі	-
Кардрідер внутрішній	Transcend TS-RDF8K USB 3.0	-
інше	Клавіатура, мишка	Подарунок

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Продовження таблиці 7.5

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Монітор	Монітор BenQ GL2450HM Black	3600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37	2800
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	Powercom BNT-600AP USB	1400

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	8	12721	10176,8	111944,8
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2800	280	3080
Копіюв. апарат	1	5965	596,5	6561,5
Всього	—	—	—	133576,3

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	1280000	-	-
2. Передавальні пристрої	128000	-	-
Всього по групі	1408000	5	70400
Група 4			
3. Обчислювальна техніка	137536	-	-
Всього по групі	137536	30	41260,8
Нематеріальні активи			
4. Нематеріальні активи	13000	25	3250
Група 5, 6			
5. Вимірювальні пристрої	9031	25	2257,75
6. Транспортні засоби	121875	20	24375
7. Господарський інвентар	28000	25	7000
Всього по групі	158906	-	33632,75
Разом	$K_p = 1717442$		$A_p = 148543,55$

Примітка: вартість автомобіля Citroen Berlingo пасс. 2002 взята за даними електронного ресурсу https://auto.ria.com/uk/auto_citroen_berlingo_pass_33583056.html і складає 121875 грн.

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де N_e – Кількість екземплярів програм, шт.

$$Z_o = 420 \cdot 176 / 130 = 569 \text{ грн}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де H_q – норматив додаткової зарплати, %

$$Z_d = 569 \cdot 10 \cdot 0,01 = 57 \text{ грн}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де H_c – відрахування на соціальні потреби, %

$$C_{oc} = 0,01 \cdot 22(569 + 57) = 138 \text{ грн}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_g = 15\%$ від основної зарплати

$$G_{ocn} = Z_o \cdot H_g \cdot 0,01, \quad (7.14)$$

де H_g – загальногосподарські витрати, %

$$G_{ocn} = 569 \cdot 15 \cdot 0,01 = 85 \text{ грн}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3}) / N_e, \quad (7.15)$$

де Z_{M1} – вартість паперу, грн., Z_{M2} – вартість запам'ятовуючих пристроїв, грн., Z_{M3} – вартість фарби, картриджей, тонеру, грн., N_e – кількість екземплярів програм, шт.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

Згідно прийнятих норм на підприємстві $n_{\text{вум}}$ приймаємо 0,33 пачки паперу на період розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n=210$ грн., визначаємо вартість паперу за період розробки:

$$З_{M1} = Ц_n \cdot N_m. \quad (7.16)$$

$$З_{M1} = 210 \cdot 0,33 = 70 \text{ грн.}$$

Згідно прийнятих норм по комплектації до вартості запам'ятовуваних пристроїв входить вартість CD/DVD дисків. Їх кількість дорівнює кількості коробочних версій запропонованого продукту (приймаємо 38):

$$З_{M2} = \sum Ц_{\delta}, \quad (7.17)$$

де: $Ц_{\delta}$ – вартість дисків CD/DVD: CDR box – 24 грн./шт., DVD-R box – 34 грн./шт.

$$З_{M2} = 37 \cdot 24 + 34 = 922 \text{ грн.}$$

Згідно норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$З_{M3} = \sum Ц_{з}, \quad (7.18)$$

де: $Ц_{з}$ – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$З_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$З_M = (70 + 922 + 1702) / 130 = 21 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де H_n – норматив витрат на освоєння нових мов програмування, %

$$O_n = 569 \cdot 15 \cdot 0,01 = 85 \text{ грн}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 130$ прим.)

$$A_m = \frac{A_p \cdot N_{\text{міс}}}{N_e \cdot 12}, \quad (7.20)$$

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

де A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 148544 \cdot 2 / (130 \cdot 12) = 190 \text{ грн}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_m + O_n + A_m \quad (7.21)$$

$$C_n = 569 + 57 + 138 + 85 + 21 + 85 + 190 = 1145 \text{ грн.}$$

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 50%

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де P_c – рівень рентабельності, %

$$P_p = 0,01 \cdot 50 \cdot 1145 = 572,5 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн.
1	2	3
Основна зарплата виконавців	Z_o	569
Додаткова зарплата виконавців	Z_d	57
Відрахування на соціальні потреби	C_{oc}	138
Загальногосподарські витрати	Γ_{ocn}	85
Витрати на матеріали	Z_m	21
Освоєння нових операційних систем, мов програмування	O_n	85

Продовження таблиці 7.9

1	2	3
Амортизація основних фондів	A_m	190
Повна собівартість програмного забезпечення	C_n	1145
Плановий прибуток	P_p	572,5
1). Ціна підприємства $C_n = C_n + P_p$	C_n	1717,5
2). Податок на додану вартість $ДВ = 0.01 \cdot H_{дв} \cdot C_n$	$ПДВ$	343,5
2. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	2061

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн	
	Базовий	Новий
Вартість програмної продукції	–	2061
Всього капітальних витрат	–	2061

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування системи	Z_p	13420	5368
2. Витрати на амортизацію	$Z_{ам}$	0	515,25
Всього витрат за рік	I	13420	5883,25

Витрати на технічне обслуговування і підтримку працездатності системи:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де T_p – кількість годин обслуговування за рік, год.,

Z_z – заробітна плата обслуговуючого персоналу, грн/год

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 100 годин на рік до 40 годин на рік, тому витрати на технічне обслуговування зменшилися з

$$Z_{p \text{ баз}} = 100 \cdot 100 \cdot 1,1 \cdot 1,22 = 13420 \text{ грн.}$$

до

$$Z_{p \text{ нов}} = 40 \cdot 100 \cdot 1,1 \cdot 1,22 = 5368 \text{ грн.}$$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	130
2. Повна собівартість розробленої програми	Грн.	1145
3. Ціна розробленої програми	Грн.	1717,5
4. Плановий прибуток від реалізації розробленої програми	Грн.	572,5
5. Рентабельність програмної продукції	%	50
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	1717442
7. Загальний прибуток від реалізації програмної продукції	Грн.	74425
8. Величина економічного ефекту при виготовленні програмної продукції	Грн.	49668
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	0,7
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	2061
11. Величина економічного ефекту у користувача програмної продукції	Грн.	7022
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Роки	0,27

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n (K_n - K_{\bar{o}}), \quad (7.26)$$

де $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно, $K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються

$$E_{cn} = (13420 - 5883) - 0,25 \cdot 2061 = 7022 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n} \quad (7.27)$$

$$T_{cn} = \frac{2061}{13420 - 5883} = 0,27 \text{ роки}$$

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Охорона здоров'я працівників, забезпечення безпеки умов праці, ліквідація професійних захворювань і виробничого травматизму повинна складати одну з головних завдань роботодавця.

Основою охорони праці є науковий аналіз умов праці, технологічних процесів, виробничого обладнання, робочих місць, трудових операцій, організації виробництва з метою виявлення шкідливих і небезпечних виробничих факторів, їх властивостей, особливостей впливу на організм людини. На підставі такого аналізу розробляються заходи та засоби, спрямовані на мінімізацію несприятливого впливу виробничих факторів, створення безпечних та нешкідливих умов праці.

Для того, щоб об'єктивно проаналізувати відповідність умов праці діючим нормативно-правовим актам, необхідно здійснити санітарно-гігієнічну характеристику умов праці відділу, в якому працює програміст, над розробкою даного програмного продукту.

В зв'язку з цим необхідно сконцентрувати увагу на небезпечних і шкідливих чинниках пов'язаних з постійною роботою за комп'ютером.

Електробезпека є одним із критичних питань для співробітників, що працюють із технікою, яка одержує живлення з електричної мережі. При невиконанні норм електробезпеки можлива поразка електричним струмом.

8.2 Пожежна безпека

Вимоги до пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

Передбачено також створення підрозділу добровільної пожежної охорони та пожежно-рятувальної команди в його складі.

Встановлений режим включає порядки з описом місць спеціального призначення та правила їх користування та утримання, наприклад:

- евакуаційних шляхів;
- так званих «курилок»;
- місць складування продукції та сировини;
- стоянки транспорту.

Також встановлюється порядок роботи та технічного обслуговування:

- вентиляційного устаткування;
- засобів пожежогасіння і захисту від загорянь;
- нагрівальних приладів;
- електрообладнання.

Розробляються і впроваджуються правила роботи з відкритим вогнем і горючими матеріалами. Створюються графіки проходження інструктажів з пожежної безпеки співробітників, а також порядок і терміни перевірок знань пожежно-технічного мінімуму, в тому числі, тих працівників, які відповідальні за цю ділянку роботи на підприємстві. При цьому можуть передбачатися внутрішні лекції, семінари, тренінги та практичні заняття на підприємстві, а також зовнішні – на базі спеціалізованих навчальних центрів з професійними

					VKPM-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		87

викладачами.

Важливою складовою протипожежного режиму на будь-якому об'єкті є розробка і впровадження порядку дій при виникненні пожежі. Неодмінно має бути план евакуації, описано, як повинні відключатися електроустановки, що і в якій послідовності необхідно робити співробітникам.

Відповідно, для кожного об'єкта, кожного приміщення (крім коридорів, санвузлів, басейнів і подібних приміщень), окремих видів робіт складаються інструкції, за якими повинен працювати персонал, залучений на певних ділянках і в виконанні окремих видів робіт. За інструкціями проводиться навчання (інструктаж) персоналу з подальшим контролем знань.

Детально про те, як розробити протипожежний режим, прописати порядки та інструкції, пояснюють на тематичних курсах і семінарах. [4]

8.3 Аналіз умов праці програміста

Умови праці в приміщенні, в якому знаходиться робоче місце програміста є сприятливими. Приміщення обладнане автономною системою газового опалення, основною перевагою якого є програмування режиму роботи в залежності від погодних умов, оскільки клімат є нестійким. Використовується система природної та штучної вентиляції, що забезпечує ефективну циркуляцію повітря. В кабінеті знаходиться кондиціонер *HYUNDAI ARN07HSSUAWF1/ARU07HSSUAWF1*.

Засоби копіювальної техніки знаходяться на достатньо далекій відстані від робочих місць, оскільки приміщення складає 15 м², а у відділі налічується два працівники, тобто концентрація озону та оксиду азоту в повітрі є невисокою. Таким чином на кожного програміста приходиться 7,5 м² що відповідає нормам Державним санітарним правилам і нормам ДСанПіН 3.3.2.007-98 [2]. Висота стелі приміщення складає 2,9 метри, що також не порушує нормативні вимоги.. Прибиральники підтримують порядок в службових приміщеннях, дотримуються

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		88

санітарно-гігієнічних норм по прибиранню приміщень, втирають пил, підмітають підлогу наприкінці кожного робочого дня.

В цілому потрібно відмітити застарілість офісної техніки та відсутність клавіатур з ергономічною розкладкою та рідкокристалічних моніторів, які здійснюють менш негативний вплив на стан здоров'я працівників відділу.

Оформлення інтер'єру приміщення є відповідає вимогам з ергономіки та стимулює працівників до підвищення працездатності та зниження втоми. Стеля білого кольору створює оптичний ефект збільшення висоти приміщення, підлога пофарбована коричневим кольором, а стіни – у жовтий. Перевагами даного кольору є створення відчуття теплоти, здатність привертати увагу без додаткової втоми.

Висота столу складає 72 см., до того ж його можна регулювати відповідно до власних потреб. Стіл має достатній внутрішній об'єм, завдяки ширині у 70 см. та висоті простору під столом – у 60 см., є достатньо важким для забезпечення стійкості. Крісла забезпечують фізіологічно раціональну позу, мають підлокітники, здатні обертатися та регулятор висоти, кута нахилу спинки й відстані спинки від краю сидіння.

В кабінеті створено оптимальні умови праці відносно температури, вологості приміщення та вентиляції.

Наприкінці аналізу небезпечних факторів праці побудуємо підсумкову таблицю 8.1.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

Таблиця 8.1 – Підсумкова таблиця значень параметрів небезпечних факторів праці

Найменування параметра	Значення параметра		Нормативний документ
	Фактичне	Нормоване	
1	2	3	4
Освітленість штучна, лк	300	300	ДБН.В 2.5-28:2018 [1]
Значення КПО,%	1,0	1,1	ДБН.В 2.5-28:2018 [1]
Повітрообмін,м /год			
взимку	76	80	ДСН 3.3.6.042-99
влітку	36	80	ДСН 3.3.6.042-99
Температура повітря. °С			
взимку	22	21-25	ДСанПіН 3.3.2-007-98
влітку	24	27-28	ДСанПіН 3.3.2-007-98 [2]
Відносна вологість,%			
взимку	60	<75	ДСанПіН 3.3.2-007-98
влітку	55	<60	ДСанПіН 3.3.2-007-98 [2]
Швидкість переміщення повітря, м/с			
взимку	0,16	<0,2	ДСанПіН 3.3.2-007-98 [2]
влітку	0,10	<0,2	ДСанПіН 3.3.2-007-98 [2]

Щодо вимог електробезпеки, то приміщення за безпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

Для запобігання поразки електричним струмом в приміщенні відділу використовується ряд організаційно-технічних заходів: розташування проводів живлення поза зоною пересування людей; допуск до роботи електроприладів тільки тих робітників, що знайомі із технікою безпеки; використання мережних продовжувачів з вбудованими запобіжниками на 0,1 А; при ремонті обладнання персонал попереджується.

Устаткування, що працює в приміщенні живиться від мережі 220 В та частотою 50 Гц. Споживачами цієї напруги є також джерела штучного освітлення. Вони розташовуються на висоті 2,9 м., що задовольняє нормі, відповідно до якого джерела освітлення повинні розташовуватися на висоті 2,5 м від підлоги.

Проводка схована. У якості розеток для підключення устаткування застосовуються розетки з заземленим кожухом, захищеного від випадкового доторку до струмоведучих частин. Електроустаткування, що знаходиться в приміщенні відділу відноситься до установок напругою до 1000 В.

На робочому місці програміста з всього устаткування металевим є лише корпус системного блоку комп'ютера, але тут використовуються системні блоки, що відповідають стандартів фірми ІВМ, у яких крім робочої ізоляції передбачений елемент для заземлення і провід з жилою, що заземлює, для приєднання до джерела живлення.

Основні причини ураження людини електричним струмом на робочому місці:

- дотик до металевих неструмоведучих частин (корпусу, периферії комп'ютера), що можуть виявитися під напругою в результаті ушкодження ізоляції;
- нерегламентоване використання електричних приладів;
- відсутність інструктажу співробітників з правил електробезпеки.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5-10 см від екрана напруженість електростатичного поля складає 60-280 кВ/м, тобто в 10 разів перевищує норму 20 кВ/м.

Отже за результатами проведеного аналізу можна зробити висновки, що всі показники знаходяться у межах запропонованих значень

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору ланцюга).

Регулярна наочне знайомство персоналу із шляхами для евакуації людей із приміщення відповідно до плану евакуації, забезпечення розподільних щитів спеціальними розетками з заземлюючими контактами; організація заземлення всіх приладів і пристроїв, які працюють при напрузі вище 36 В.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

Так як при ураженні електричним струмом у людини може статися фібриляція шлуночків серця, в організації бажано мати дефібрилятор і підготовлений персонал для роботи з ним.

8.5 Розрахункова частина

Для захисного штучного заземлення будемо застосовувати вертикальні електроди з сталевого прокату круглого перерізу діаметром 35 мм., довжиною $L=1,5$ м., та горизонтальний електрод – металева полоса з перетином 35·4 мм. Напруга – 220/380 В. Розрахункова схема розташування заземлюючих електродів – по контуру (прямокутником).

Розрахунок проводиться за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунта – чорнозем, нижнього шару ґрунта – глина (питомий опір $\rho_2 = 40$ Ом·м). Умовна товщина верхнього шару ґрунта: $H=0,65$ м. Відстань між вертикальними заземлювачами (електродами) $A=3$ м. Глибина закладення горизонтального контура заземлення $t=0,7$ м. Опір заземлювача, який нормується: $R_{3H} = 4$ Ом. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

Розрахунок

Відстань від центра вертикального заземлювача до поверхні землі:

$$T=t+L/2=0,7+1,5/2=1,45 \text{ м.}$$

Розрахунковий питомий опір ґрунта (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунта):

$$\rho = \psi \rho = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м.}$$

де $\psi = 1,36$ – табличне значення коефіцієнта сезонності для відповідної кліматичної зони у багат шаровому ґрунті [11];

$\rho_2 = 40$ Ом·м. – табличне значення питомого опору нижнього шару ґрунта (глина) [11].

Діаметр вертикального електрода (задан):

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

$$D_в = 35 \text{ мм.} = 0,035 \text{ м.}$$

$$\text{Відношення } A/L = 3/1,5 = 2.$$

Опір розтіканню електричного струму одного електрода вертикального заземлювача з урахуванням заглиблення заземлювача [11]:

$$\begin{aligned} R_0 &= 0,366(\rho/L)[\lg(2L/D_в) + (1/2)\lg((4T+L)/(4T-L))] = \\ &= 0,366(54,5/1,5)[\lg(2 \cdot 1,5/0,035) + (1/2)\lg((4 \cdot 1,45 + 1,5)/(4 \cdot 1,45 - 1,5))] = \\ &= 27,1 \text{ Ом.} \end{aligned}$$

Визначаємо коефіцієнт екранування вертикальних електродів $K_{ев} = 0,53$ при орієнтовній кількості вертикальних електродів, яке дорівнює 5 [11].

Визначаємо необхідну кількість вертикальних електродів заземлювача (без врахування горизонтального заземлювача), при $R_{зН} = 4 \text{ Ом}$:

$$N = R_0 / (K_{ев} R_{зН}) = 27,1 / (0,53 \cdot 4) = 12,8 \approx 13 \text{ шт.}$$

Визначаємо довжину з'єднуючої полоси:

$$L_{\Pi} = 1,05 \cdot A \cdot N = 1,05 \cdot 3 \cdot 13 = 40,3 \approx 40 \text{ м.}$$

Опір розтіканню електричного струму з'єднуючої полоси з урахуванням кліматичного коефіцієнта питомого опору ґрунта K_{Π} [11]:

$$\begin{aligned} R_{\Pi} &= 0,366(\rho \cdot K_{\Pi}/L_{\Pi}) \lg(2(L_{\Pi} \cdot L_{\Pi})/(B \cdot t)) = \\ &= 0,366(40 \cdot 5/40) \cdot \lg((2 \cdot 40^2)/(0,035 \cdot 0,7)) = 9,2 \text{ Ом.} \end{aligned}$$

де $K_{\Pi} = 5$ – табличне значення кліматичного коефіцієнта питомого опору ґрунта для відповідної кліматичної зони для з'єднуючої полоси [11]:

$$B = 35 \text{ мм.} = 0,035 \text{ м.} - \text{ ширина з'єднуючої полоси (задана).}$$

Загальний опір розтіканню електричного струму заземлювача [11]:

$$\begin{aligned} R &= (R_0 \cdot R_{\Pi}) / (R_0 \cdot \eta_{\Pi} + N \cdot R_{\Pi} \cdot K_{ев}) = \\ &= (27,1 \cdot 9,2) / (27,1 \cdot 0,55 + 13 \cdot 9,2 \cdot 0,53) = 3,19 \text{ Ом.} \end{aligned}$$

де $\eta_{\Pi} = 0,55$ – табличне значення коефіцієнта екранування з'єднуючої полоси [11].

$$\text{Умова } R \leq R_{зН} \text{ виконується (} 3,19 \leq 4 \text{).}$$

Так як R суттєво більше $R_{зН}$, зменшимо кількість вертикальних електродів до 10 і виконаємо перерахунок. У результаті остаточно отримали: кількість вертикальних електродів дорівнює 10 при $R = 3,9 \text{ Ом}$.

									ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата						94

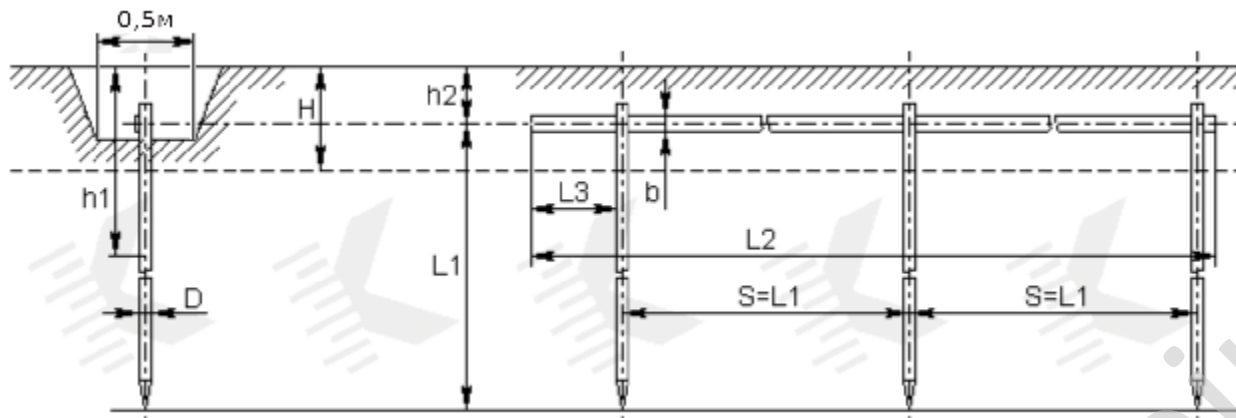


Рисунок 8.1 – Схема штучного заземлення .

8.6 Висновки до розділу

Дотримання всіх необхідних умов праці не лише сприяє збереженню здоров'я працівників, а також підвищує ефективність виробництва в цілому.

З цих міркувань було здійснено аналіз приміщення, призначеного для праці програмістів, проведено розгляд питань пожежної безпеки, небезпечних та шкідливих факторів, що негативно впливають на програмістів під час роботи.

Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів о питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи виявлення несправності елементів цифрових пристроїв.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів виявлення несправності елементів цифрових пристроїв.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем виявлення несправності елементів цифрових пристроїв.

– Досліджена система виявлення несправності елементів цифрових пристроїв.

– На основі отриманих результатів досліджень створена програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання виявлення несправності елементів цифрових пристроїв.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

При створені програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня RAD Studio Delphi 10.4. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ДСТУ 7564:2014.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 7022 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,27 роки.

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Дзюбинський О.В. Дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв // Збірник праць молодих науковців ЦНТУ. – Вип. 13. – Кропивницький: ЦНТУ, 2022.

2. Коваленко А.С. Разработка структуры базы данных интегрированной информационной системы / А.С. Коваленко, А.В. Коваленко // Информационные технологии и защита информации в информационно-коммуникационных системах: монографія / Под редакцией профессора В.С. Пономаренко. – Х.: Вид-во ТОВ «Щедра садиба плюс», 2015. – С. 54-64.

3. Кожанова А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / О.А. Смірнов, А.С. Кожанова, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2013. – Вип. 6(113). – С. 255-257.

4. Коваленко А.С. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко., А.А. Смірнов, А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2014. – Вип. 4(120). – С. 161-164.

5. Коваленко А.С. Підсистема технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А.Смірнов, О.В. Коваленко // Системи озброєння і військова техніка.– Х.: ХУПС, 2014. – № 1(37). – С. 126-129.

6. Коваленко А.С. Анализ эффективности использования экспертной системы технической диагностики с традиционной структурой / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 2(38). – С. 106-108.

7. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы /

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

8. Коваленко А.С. Разработка структуры экспертной системы технической диагностики интегрированной информационной системы / А.С. Коваленко, А.А. Смирнов, А.В. Коваленко // Наука і техніка Повітряних Сил Збройних Сил України. – Харків: ХУПС, 2014. – № 2(15). – С.154-157.

9. Коваленко А.С. Структура системи технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Кіровоградського національного технічного університету / техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Кіровоград: Вид-во КНТУ, 2014. – Вип. 27. – С. 245-251.

10. Коваленко А.С. Дослідження будови інтегрованої інформаційної системи та її елементів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2014. – № 4(40). – С. 85-88.

11. Коваленко А.С. Розробка структури бази даних для обліку технічного стану елементів інтегрованої інформаційної системи з урахуванням вимог споживачів інформації / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2015. – Вип. 1(126). – С. 75-79.

12. Коваленко А.С. Обґрунтування набору даних для оцінки технічного стану інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Збірник наукових праць Харківського університету Повітряних Сил. – Харків: ХУПС, 2015. – Вип. 1(42). – С.39-41.

13. Коваленко А.С. Експертна система технічного діагностування інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Системи озброєння і військова техніка. – Х.: ХУПС, 2015. – № 1(41). – С. 106-111.

14. Коваленко А.С. Удосконалення методу технічного обслуговування об'єктів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов,

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

О.В. Коваленко, О.П. Доренський // Системи озброєння і військова техніка. – Х.: ХУПС, 2016. – № 2(46). – С. 109-114.

15. Коваленко А.С. Метод визначення оптимального комплексу робіт з відновлення працездатності інтегрованої системи технічної діагностики в умовах ресурсних обмежень / А.С. Коваленко // Системи обробки інформації. – Х.: ХУПС, 2016. – Вип. 3(140). – С. 69-72.

16. Kovalenko A.S. Information model and its element for displaying information on technical condition of objects of integrated information system / A.S. Kovalenko, A.A. Smirnov, A.V. Kovalenko, A.P. Dorensky // International Journal of Computational Engineering Research (IJCER). – India: Delhi, 2016. – Volume 6, Issue 1. – P. 21-27.

17. Кожанова А.С. Система технічної діагностики інтегрованих інформаційних систем – обґрунтування необхідності створення, визначення понятійного апарату та напрямів досліджень / А.С. Кожанова, О.А. Смірнов, М.П. Савченко, Д.М. Ізосімов, В.В. Мороз // Створення та модернізація озброєння і військової техніки в сучасних умовах: Тринадцята наук.-техн. конф., 5-6 вер. 2013 р., м. Феодосія: тези доп. – Феодосія: ДНВЦ, 2013. – С. 187-188.

18. Кожанова А.С. Визначення основних напрямків досліджень щодо створення системи технічної діагностики інтегрованих інформаційних систем / А.С. Кожанова, О.А. Смірнов, А.В. Челпанов // Проблемні питання розвитку озброєння та військової техніки Збройних Сил України: IV наук.-техн. конф., 16-20 груд. 2013 р., м. Київ: зб. тез. – Київ: ЦНДІ ОВТ ЗСУ, 2013. – С. 293.

19. Коваленко А.С. Обґрунтування необхідності створення систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформатика та системні науки : V Всеукр. наук.-практ. конф., 13–15 бер. 2014 р., м. Полтава : зб. тез. – Полтава: ПУЕТ, 2014. – С. 292-294.

20. Коваленко А.С. Задачи распознавания ситуаций в системах организационной стратегии интеграции производства и операций

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

/ А.С. Коваленко, А.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVI міжнар. наук.-практ. сем., 11-12 квіт. 2014 р., м. Кіровоград: зб. тез. – Кіровоград: КНТУ, 2014. – С. 53-55.

21. Коваленко А.С. Створення систем технічної діагностики для автоматизації процесів керування в інтегрованих інформаційних системах / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VI між нар. наук.-практ. конф., 17-18 квіт. 2014 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2014. – С. 241.

22. Коваленко А.С. Визначення понятійного апарату та напрямів досліджень для синтезу систем технічної діагностики інтегрованих інформаційних систем / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комп'ютерне моделювання у наукоємних технологіях (КМНТ-2014): наук.-техн. конф. з міжнар. участю, 28-31 трав. 2014 р., м. Харків: зб. наук. праць. – Харків: ХНУ, 2014. – С. 190-193.

23. Коваленко А.С. Основні складові та функції системи технічної діагностики інтегрованих інформаційних систем / Коваленко А.С. // Інформаційні технології та комп'ютерна інженерія: наук.-практ. конф., 4 груд. 2014 р., м. Кіровоград: зб. тез доп. – Кіровоград: КНТУ, 2014. – С. 236.

24. Коваленко А.С. Розробка структури бази даних інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Проблеми і перспективи розвитку ІТ-індустрії: VII міжнар. наук.-практ. конф., 17-18 квіт. 2015 р., м. Харків: зб. тез. – Харків: ХНЕУ, 2015. – С. 15.

25. Коваленко А.С. Дослідження елементів інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Комбінаторні конфігурації та їх застосування: XVII між нар. наук.-практ. сем., 17-18 квіт. 2015 р., м. Кіровоград: зб. тез – Кіровоград: КНТУ, 2015. – С. 5.

26. Коваленко А.С. Метод автоматизованої перевірки результатів вимірювання параметрів об'єкті в інтегрованої інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Стратегія якості у

промисловості і освіти: XI міжнар. конф., 1 – 5 черв. 2015 р., м. Варна, Болгарія.: зб. матер. – Варна: ТУВ, 2015. – С. 423-426.

27. Коваленко А.С. Обґрунтування необхідності створення розподіленої бази даних для забезпечення захисту рухомих повітряних об'єктів / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Перспективні напрями захисту інформації: I всеукр. наук.-практ. конф., 07 вер. 2015 р., м. Одеса: зб. тез доп. – Одеса: ОНАЗ, 2015. – С. 35-39.

28. Коваленко А.С. Розробка інформаційної моделі автоматизованої оцінки технічного стану інтегральної інформаційної системи / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Інформаційні технології та взаємодії (IT & I): II між нар. наук.-практ. конф., 3-5 лист. 2015 р., м. Київ: тези доп. – Київ: КНУ ім. Т. Шевченка, 2015. – С. 41-42.

29. Коваленко А.С. Разработка метода усовершенствования технического обслуживания интегрированной информационной системы / А.С. Коваленко, О.А. Смірнов, О.В. Коваленко // Информационные и телекоммуникационные технологии: образование, наука, практика: II междунар. научн.-практ. конф., 3-4 дек. 2015 г., г. Алматы, Казахстан: сб. труд. – Алматы: КазНИТУ им. К.И. Сатпаева, 2015. – Т.2. – С. 423-427.

30. Королюк Н.А. Оценка временных интервалов работы лица, принимающего решение, на автоматизированном командном пункте / Н.А. Королюк, А.И. Тимочко // Системы обработки информации. – Х.: ХУПС, 2005. – Вып. 8 (48). – С. 51-54.

31. Костерев В.В. Надёжность технических систем и управление риском: учебн. пособ. / В.В. Костерев. – М.: МИФИ, 2008. – 280 с.

32. Костюков А.В. Підвищення операційної ефективності підприємств на основі моніторингу в реальному часі. / А.В. Костюков, В.М. Костюков. – М.: Машинобудування, 2009. – 192 с.

33. Лазарев А.А. Выбор показателя затрат для анализа сравнительной экономической эффективности техники конечного потребления / А.А. Лазарев,

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

М.В. Бейлин // Сборник научных трудов ХГПУ.– Х.: ХГПУ, 1999. – Вып. 74. – С. 27-29.

34. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 1. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 308 с.

35. Ланецкий Б.Н. Основы теории надежности, технического обслуживания и ремонта вооружения и военной техники: Справочные материалы, часть 2. / Б.Н. Ланецкий, А.А. Посудевский. – Харьков: ХВУ, 1993. – 208 с.

36. Лапсарь А.П. Метод оценки состояния сложных технических объектов для синтеза быстродействующих прогнозирующих систем / А.П. Лапсарь // Измерительная техника. – 2004. – № 2. – С. 7-10.

37. Линейные задачи оптимизации: Учеб. пособие / С.В. Лутманов. – Пермь: ЛИТЕР-А, 2004. – Ч.1. – Линейное программирование. – 128 с.

38. Литвак Б.Г. Экспертные технологии в управлении. Учебное пособие / Б.Г. Литвак. – М.: Дело, 2014. – 318 с.

39. Локазюк В.М. Надійність, контроль, діагностика і модернізація ПК: Посібн. / В.М. Локазюк, Ю.Г. Савченко. – К.: Видавничий центр «Академія», 2004. – 376 с.

40. Лопатников Л. И. Экономико-математический словарь: Словарь современной экономической науки / Л.И. Лопатников. – М.: Дело, 2003. – 520 с.

41. Манухина С.Ю. Инженерная психология и эргономика: 10/11естоматия / С.Ю. Манухина. – М.: Изд. центр ЕАОИ, 2009. – 224 с.

42. Мартыненко М.В. Человекомашинные процедуры поддержки организационно–управленческих решений: учеб. пособие СПбГЭТУ / М.В. Мартыненко, О.И. Шеховцов. – СПб, 2012. – 250 с.

43. Мунипов О.В. Эргономика: человекоориентированное проектирование техники, программных средств и среды: Учебник / О.В. Мунипов, В.П. Зинченко. – М.: Логос, 2001. – 356 с.

ВКРМ-123.22.0007.00.00.ПЗ

Арк.

103

Вим. Арк. № докум. Підпис Дата

44. Надеев А.И. Математическая модель эксплуатационной надежности интеллектуальных датчиков / А.И. Надеев, Р.А. Юсупов, Ю.К. Свечников, Д.Р. Юсупов // Измерительная техника. – М: Стандартинформ, 2004. – № 1. – С. 8-11.

45. Надійність техніки. Аналіз надійності. Основні положення: ДСТУ 2861-94 – [Чинний від 1997-01-01]. – Київ: Держстандарт України, 1995. – 33 с. – (Національний стандарт України).

46. Надійність техніки. Терміни та визначення: ДСТУ 2860-94 – [Чинний від 1996-01-01]. – Київ: Держстандарт України, 1994. – 36 с. – (Національний стандарт України).

47. Нейлор К. Как построить свою экспертную систему / К. Нейлор. – М.: Энергоатомиздат, 2007. – 242 с.

48. Николаева И. П. Экономический словарь / И.П. Николаева. – Проспект, 2015. – 399 с.

49. Онищук А.Г. Радиоприемные устройства: Учебн. пособ. – 2-е изд., испр. / А.Г. Онищук, И.И. Забеньков, А.М. Амелин. – Минск: Новое знание, 2007. – 240 с.

50. Осипов В. Базы данных и Delphi. Теория и практика / В. Осипов. – БХВ-Петербург, 2011. – 752 с.

51. Павленко М.А. Метод кольорового кодування інформаційних елементів при розробці інформаційних моделей в перспективних АСУ / М.А. Павленко, П.Г. Берднік, Д.В. Прибильнов // Наукова весна – 2008: Матеріали міжнародної наук. – практ. конф. – Х.: МСУ, 2008. – С. 25-27.

52. Державні будівельні норми України: ДБН В.2.5-28:2018. – Режим доступу до ресурсу: <https://goo.su/9AkQ>

53. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/v0007282-98>

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		104

54. Закон України «Про охорону праці» від 14.10.1992 р. № 2694-ХІІ. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/2694-12>

55. Зеркалов Д. В. Охорона праці в Галузі: Загальні вимоги: навч. посіб. Київ: Основа. 2011. 551 с.

56. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/laws/show/z0508>

57. Охорона праці. Ч. 1. Захисне заземлення: метод. вказ. до викон. розрахунків з викор. персон. ЕОМ ІВМ сумісного типу / Кіровоград. ін-т с.-г. машинобуд.; [укл. О. В. Оришака, Є. К. Солових, В. О. Оришака]. – Кіровоград: КІСМ, 1997. – 20 с. Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/4358>

58. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99. – Режим доступу до ресурсу: <https://zakon.rada.gov.ua/rada/show/va042282-99>

59. Сакулин В.П., Шептовицкий В.М. Безопасность труда при монтаже и эксплуатации электроустановок / В.П.Сакулин, В.М.Шептовицкий. – Л. : “Колос”, 1973. – 238 с.

60. Центр післядипломної освіти та підвищення кваліфікації. – Режим доступу до ресурсу: <https://spo.stu.cn.ua>

61. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький : ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення 19.09.22).

					ВКРМ-123.22.0007.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.22.0007.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Дзюбинський О.В.				<i>Дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнова Т.В.					М	1	6
Н. Контр.	Гермак В.С.				ЦНТУ КІ-21М-1,4			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи виявлення несправності елементів цифрових пристроїв.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 19-13 від 17.08.2022 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи виявлення несправності елементів цифрових пристроїв.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.22.0007.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи виявлення несправності елементів цифрових пристроїв;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.22.0007.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище RAD Studio Delphi 10.4.

					ВКРМ-123.22.0007.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		4

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2022 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута пожежна безпека.

					ВКРМ-123.22.0007.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 105 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2022 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 21.12.2022 р.

					ВКРМ-123.22.0007.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Смірнова Т.В.

*Дослідження та програмна реалізація
системи виявлення несправності елементів цифрових пристроїв*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 42

Літера: РП

Кропивницький – 2022 року

Файл Main.pas - основна програма

```
unit Main;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms, Dialogs, StdCtrls, ExtCtrls, Menus, ToolWin, ComCtrls, IniFiles,
  SomeClasses,
  ImgList, Buttons, About;

type
  TForm1 = class(TForm)
    Frame21: TFrame2;
    Panel1: TPanel;
    MainMenu1: TMainMenu;
    N1: TMenuItem;
    Button1: TButton;
    Image1: TImage;
    Button2: TButton;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    Panel2: TPanel;
    Panel3: TPanel;
    ToolBar1: TToolBar;
    ToolButton5: TToolButton;
    Panel4: TPanel;
    StatusBar1: TStatusBar;
    Panel5: TPanel;
    N10: TMenuItem;
    Panel6: TPanel;
    ComboBox1: TComboBox;
    N9: TMenuItem;
    N11: TMenuItem;
    N12: TMenuItem;
    N13: TMenuItem;
    N14: TMenuItem;
    Button4: TButton;
    Label1: TLabel;
    N15: TMenuItem;
    N16: TMenuItem;
    TreeView1: TTreeView;
    ImageList1: TImageList;
    Button3: TButton;
    Button5: TButton;
    N17: TMenuItem;
    N18: TMenuItem;
    PopupMenu1: TPopupMenu;
    N19: TMenuItem;
    N20: TMenuItem;
    N21: TMenuItem;
    N22: TMenuItem;
    ImageList2: TImageList;
    N23: TMenuItem;
    N24: TMenuItem;
    PopupMenu2: TPopupMenu;
    N25: TMenuItem;
    PopupMenu3: TPopupMenu;
    N26: TMenuItem;
    ToolButton2: TToolButton;
```

```

ImageList3: TImageList;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ToolButton9: TToolButton;
ImageList4: TImageList;
N27: TMenuItem;
N28: TMenuItem;
Button6: TButton;
Button7: TButton;
Button8: TButton;
OpenDialog1: TOpenDialog;
SaveDialog1: TSaveDialog;
N29: TMenuItem;
N30: TMenuItem;
N31: TMenuItem;
N32: TMenuItem;
N33: TMenuItem;
N34: TMenuItem;
N35: TMenuItem;
ToolButton1: TToolButton;
N36: TMenuItem;
N37: TMenuItem;
procedure Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Image1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
  Y: Integer);
procedure FormCreate(Sender: TObject);
procedure FormMouseWheel(Sender: TObject; Shift: TShiftState;
  WheelDelta: Integer; MousePos: TPoint; var Handled: Boolean);
procedure FormActivate(Sender: TObject);
procedure Frame21Image1Click(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure N3Click(Sender: TObject);
procedure N4Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure Frame21Image1MouseDown(Sender: TObject; Button:
TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure Frame21Image1MouseUp(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
procedure DrawSelectionRectangle (SP : TPoint);
procedure N8Click(Sender: TObject);
procedure N7Click(Sender: TObject);
procedure Panel3Click(Sender: TObject);
procedure Panel4Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure DesktopSetup;
procedure InitCfg;
procedure SaveCfg;
procedure FormClose(Sender: TObject; var Action: TCloseAction);
procedure ComboBox1Change(Sender: TObject);
procedure RepaintElements;
procedure N11Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure Button4Click(Sender: TObject);
procedure N14Click(Sender: TObject);
procedure Frame21Image1MouseMove(Sender: TObject; Shift: TShiftState;
  X, Y: Integer);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure TreeView1Db1Click(Sender: TObject);
procedure TreeView1Click(Sender: TObject);
procedure AssociateTree;
procedure N19Click(Sender: TObject);

```

```

procedure N21Click(Sender: TObject);
procedure N24Click(Sender: TObject);
procedure Image1Db1Click(Sender: TObject);
procedure N25Click(Sender: TObject);
procedure DrawField;
procedure N26Click(Sender: TObject);
procedure ToolButton6Click(Sender: TObject);
procedure ToolButton7Click(Sender: TObject);
procedure ToolButton8Click(Sender: TObject);
procedure ToolButton9Click(Sender: TObject);
procedure N27Click(Sender: TObject);
procedure N28Click(Sender: TObject);
procedure N15Click(Sender: TObject);
procedure FormShow(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button7Click(Sender: TObject);
procedure Button8Click(Sender: TObject);
procedure N34Click(Sender: TObject);
procedure N37Click(Sender: TObject);
private
  { Private declarations }
public
  { Public declarations }
end;
TCons = array of TPoint;
TConn = record
  Elem: integer;
  Leg: integer;
end;
TConns = array of TConn;

procedure Delay(n: Cardinal);

const
  Con_Color = clBlue;

var
  F_Stop: Boolean;
  F_Pause: Boolean;
  Form1: TForm1;
  Cons: array of TCons;
  Conns: array of TConns;
  Desktop_Color, Grid_Color : TColor;
  Selection_Metod : Boolean;
  IniFile : TIniFile;
  Koeff_Mashtab: integer;
  Old_Mashtab : integer;
  Elements: array [1..1000] of TElement;
  Chips_count : integer;
  Select_on_creation : boolean;
  Draw_connections : boolean;
  Select_connections : boolean;
  Legs_sel : Boolean;
  Legs_sel_Elem : TObject;
  Legs_sel_Leg : integer;
  Legs_sel2 : Boolean;
  Legs_sel_Elem2 : TObject;
  Legs_sel_Leg2 : integer;
  Selected_Connection : integer;
  Conn_Prov : integer;
  Conn_Prov_x : integer;
  Conn_Prov_y : integer;
  Last_Elem : integer;
  Osc_Count : integer;
  Oscill : array [1..5] of integer;
  fio : file of TElement;
  fio2 : file of TConn;

```

implementation

```

    uses Parameters, Db_elements, Connector, Db_micro, Types, Tracer,
    Time_scale,
        Main6, Osc;

    {$R *.dfm}

    var
        Mouse_Status, Selection_Status: Boolean;
        M_x, M_y: Integer;
    //    Cur_Elem: integer;
        Base_Selection_Point : TPoint;
        Second_Selection_Point : TPoint;
        Old_Second_Selection_Point : TPoint;
        Connect_Status : Boolean;
        Connect_Start : record
            Elem, Leg: Integer;
        end;

    procedure Delay(n:cardinal);
    var
        FirstTick: Cardinal;
    begin
        FirstTick := GetTickCount;
        repeat Application.ProcessMessages;
        until GetTickCount - FirstTick >= n;
    end;

    procedure TForm1.AssociateTree;
    var i, j : integer;
    begin
        for i := 0 to treeview1.Items.Count - 1 do
            if TreeView1.Items[i].ImageIndex = 1 then
                for j:= 1 to last_elem + 1 do
                    begin
                        if j = Last_Elem + 1 then begin TreeView1.Items[i].ImageIndex :=
2; TreeView1.Items[i].SelectedIndex := 2; break end;
                            if Elements[j].Caption = TreeView1.Items[i].Text then
                                begin TreeView1.Items[i].Data := @elements[j]; break end;
                            end;
                    end;
                end;

    Procedure TForm1.RepaintElements;
    var i: integer;
    begin
        for i := 1 to Last_Elem do
            begin
                Elements[i].Image.Width:=Elements[i].Image.Picture.Bitmap.Width*Koeff_Mashtab
div 10;

                Elements[i].Image.Height:=Elements[i].Image.Picture.Bitmap.Height*Koeff_Mashtab
div 10;

                Elements[i].Image.Left:=Elements[i].image.left div Old_Mashtab *
Koeff_Mashtab + 2;
                Elements[i].Image.Top:=Elements[i].image.Top div Old_Mashtab *
Koeff_Mashtab + 2;
            end;
        end;

    //ініціалізація параметрів програми
    Procedure TForm1.InitCfg;
    begin
        IniFile := Tinifile.Create
(ExtractFilePath(Application.ExeName)+'\config\main.ini'); //відкриття файлу з
параметрами програми
        Desktop_Color := IniFile.ReadInteger('Colors','Desktop',clWhite);
    //копір фону

```

```

Grid_Color := IniFile.ReadInteger('Colors','Grid',clBlack);
//копія розмітки
Selection_Metod := IniFile.ReadBool('Selection','Full',False);
//параметри виділення елементів
Select_on_creation := IniFile.ReadBool('Creation','Selection',False);
Draw_connections := IniFile.ReadBool('Selection','Draw_conn',False);
Select_connections := IniFile.ReadBool('Selection','Sel_conn',True);
if IniFile.ReadBool('Main','LockBases',True) then
begin
  N11.Enabled:=false;
  N12.Enabled:=false;
  N13.Enabled:=false;
  N14.Enabled:=false;
end;
end;

//збереження параметрів програми
Procedure TForm1.SaveCfg;
begin
  IniFile.WriteInteger('Colors','Desktop',Desktop_Color);
  IniFile.WriteInteger('Colors','Grid',Grid_Color);
  IniFile.WriteBool('Selection','Full',Selection_Metod);
  IniFile.WriteBool('Creation','Selection',Select_on_creation);
  IniFile.WriteBool('Selection','Draw_conn',Draw_connections);
  IniFile.WriteBool('Selection','Sel_conn',Select_connections);
end;

Procedure TForm1.DesktopSetup;
var i, j : integer;
begin
  for i:=0 to 200 do for j:=0 to 200 do Field[i,j]:=0;
  Frame21.Imagel.Canvas.Brush.Style := bsSolid;
  Frame21.Imagel.Canvas.Pen.Mode := pmCopy;
  Frame21.Imagel.Canvas.Pen.Color := Desktop_Color;
  Frame21.Imagel.Canvas.Brush.Color := Desktop_Color;
  Frame21.Panel1.Width:=2000*Koeff_Mashtab div 10;
  Frame21.Panel1.Height:=2000*Koeff_Mashtab div 10;
  Frame21.Imagel.Canvas.Rectangle(0,0,2000*Koeff_Mashtab div
10,2000*Koeff_Mashtab div 10);
  for i:=0 to 200 do
    for j:=0 to 200 do

Frame21.Imagel.Canvas.Pixels[i*Koeff_Mashtab,j*Koeff_Mashtab]:=Grid_Color;
// Button4Click(nil);
end;

//видалення виділених елементів
procedure DeleteSelectedElements;
var i, j, k, l : integer;
begin
  i := 1;
  while i <= Last_Elem do
  begin
    if Elements[i].Selected then
    begin
      Elements[i].destroy;
      for j:=1 to 5 do
        if Oscill[i]=i then Oscill[i]:=0;
      if Form17.Visible then Form17.CheckListBox1ClickCheck(nil);
      for j:=0 to High(Conns) do
      begin
        k:=0;
        while k<=High(Conns[j]) do
        begin
          if Conns[j][k].Elem=i then
          begin
            for l:=k to High(Conns[j])-1 do
              Conns[j][l]:=Conns[j][l+1];
            k:=k-1;

```

```

        SetLength (Conns [j], High (Conns [j]));
        end;
        k:=k+1;
        end;
        end;
        j:=0;
        while j<=High (Conns) do
        begin
            if High (Conns [j])<1 then
            begin
                for l:=j to High (Conns)-1 do
                    Conns [l]:=Conns [l+1];
                    SetLength (Conns, High (Conns));
                end;
                j:=j+1;
            end;
            for j:=0 to High (Conns) do
                for k:=0 to High (Conns [j]) do
                    if Conns [j] [k].Elem>=i then
Conns [j] [k].Elem:=Conns [j] [k].Elem-1;
                    for j:= i to Last_Elem-1 do
                        begin
                            Elements [j] := Elements [j+1];
                            Elements [j].Image.Tag := Elements [j].Image.Tag -1;
                        end;
                        dec (Last_Elem);
                        dec (i);
                    end;
                    inc (i);
                end;
            end;

            //видалення елементів
            procedure DeselectElements;
            var i :integer;
            begin
                for i:=1 to Last_Elem do
                    Elements [i].Deselect;
                end;

            procedure SelectAllElements;
            var i : integer;
            begin
                for i:=1 to Last_Elem do
                    Elements [i].Select;
                end;

            procedure InversElementSelection;
            var i: integer;
            begin
                for i:=1 to Last_Elem do
                    Elements [i].Swich_Selection;
                end;

            //промальовування виділення елементів мишкою
            Procedure TForm1.DrawSelectionRectangle (SP : TPoint);
            begin
                Frame21.Image1.Canvas.Brush.Style := bsClear;
                Frame21.Image1.Canvas.Pen.Color := clWhite;
                Frame21.Image1.Canvas.Pen.Style := psDash;
                Frame21.Image1.Canvas.Pen.Mode := pmXor;
                Frame21.Image1.Canvas.Pen.Width := 1;

            Frame21.Image1.Canvas.Rectangle (Base_Selection_Point.X,Base_Selection_Point.Y,SP
            .X,SP.Y);
            end;

            procedure Draw_Leg (Sender:TObject; i:integer);
            begin

```

```

    TImage(Sender).Canvas.Pen.Mode:=pmXor;
    TImage(Sender).Canvas.Pen.Width:=10;
    TImage(Sender).Canvas.Pen.Color:=clRed xor
Form1.Frame21.Image1.Canvas.Pixels[1,1];
    if Elements[TImage(Sender).Tag].Legs[i].x=0 then begin

TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Element
s[TImage(Sender).Tag].Legs[i].y*10);

TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10+5,Element
s[TImage(Sender).Tag].Legs[i].y*10);
    end else begin

TImage(Sender).Canvas.MoveTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);

TImage(Sender).Canvas.LineTo(Elements[TImage(Sender).Tag].Legs[i].x*10-
5,Elements[TImage(Sender).Tag].Legs[i].y*10);
    end;
end;

//з'єднання елементів
function IsConnected(E,L: Integer): Boolean;
var i,j: Integer;
begin
    Result:=true;
    for i:=0 to High(Conns) do
        for j:=0 to High(Conns[i]) do
            if (Conns[i][j].Elem=E) and (Conns[i][j].Leg=L) then exit;
        Result:=false;
    end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var selcon: Boolean;
    i: Integer;
begin
    if (Button <> mbLeft) and (Button <> mbRight) then exit;
    if Connect_Status then exit;
    selcon:= false;
    i:=1;
    if TImage(Sender).tag=0 then exit;
    if (x<Koeff_Mashtab) or (x>(Elements[TImage(Sender).tag].Size.x-
1)*Koeff_Mashtab) then
        for i:=0 to High(Elements[TImage(Sender).tag].Legs) do
            if (x>(Elements[TImage(Sender).tag].Legs[i].x-1)*Koeff_Mashtab)
                and (x<(Elements[TImage(Sender).tag].Legs[i].x+1)*Koeff_Mashtab)
                    and (y>(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
                        and
                            (y<(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab+Koeff_Mashtab div 2)
                                then begin selcon:=true; break; end;
                if Button = mbRight then
                    begin
                        if selcon and IsConnected(TImage(Sender).tag,i) then
                            begin
                                TImage(Sender).PopupMenu:=PopupMenu3;
                                Connect_Start.Elem:=TImage(Sender).tag;
                                Connect_Start.Leg:=i;
                            end else TImage(Sender).PopupMenu:=nil;
                        exit;
                    end;
                if selcon then begin
                    Connect_Status:=true;
                    Connect_Start.Elem:=TImage(Sender).tag;
                    Connect_Start.Leg:=i;
                    Legs_sel2:=false;
                end;
end;

```

```

M_x:=X;
M_y:=y;
if not (ssCtrl in Shift) then Mouse_Status:=True;
end;

//додавання з'єднання
procedure AddConnect(E1,L1,E2,L2: integer);
label 1,2;
var i,j,k:integer;
begin
  for i:=0 to High(Conns) do
    for j:= 0 to High(Conns[i]) do
      if (Conns[i][j].Elem=E1) and (Conns[i][j].Leg=L1) then goto 1;
1:
    for k:=0 to High(Conns) do
      for j:= 0 to High(Conns[k]) do
        if (Conns[k][j].Elem=E2) and (Conns[k][j].Leg=L2) then goto 2;
2:
    if ((i=High(Conns)+1) and (k=High(Conns)+1)) or (High(Conns)=-1) then
      begin
        SetLength(Conns,High(Conns)+2);
        SetLength(Conns[High(Conns)],2);
        j:=0;
        Conns[High(Conns)][j].Elem:=E1;
        Conns[High(Conns)][j].Leg:=L1;
        j:=1;
        Conns[High(Conns)][j].Elem:=E2;
        Conns[High(Conns)][j].Leg:=L2;
        InitTrace;
        Exit;
      end;
    if k=i then exit;
    if i=High(Conns)+1 then
      begin
        SetLength(Conns[k],High(Conns[k])+2);
        j:=High(Conns[k]);
        Conns[k][j].Elem:=E1;
        Conns[k][j].Leg:=L1;
        InitTrace;
        Exit;
      end;
    if k=High(Conns)+1 then
      begin
        SetLength(Conns[i],High(Conns[i])+2);
        j:=High(Conns[i]);
        Conns[i][j].Elem:=E2;
        Conns[i][j].Leg:=L2;
        InitTrace;
        Exit;
      end;
    SetLength(Conns[i],High(Conns[i])+2+High(Conns[k]));
    for j:=High(Conns[i])-High(Conns[k]) to High(Conns[i]) do
      begin
        Conns[i][j].Elem:=Conns[k][j-High(Conns[i])+High(Conns[k])].Elem;
        Conns[i][j].Leg:=Conns[k][j-High(Conns[i])+High(Conns[k])].Leg;
      end;
    SetLength(Conns[k],0);
    for j:=k to High(Conns)-1 do
      Conns[j]:=Conns[j+1];
    SetLength(Conns[j],0);
    SetLength(Conns,High(Conns));
    InitTrace;
  end;

procedure TForm1.Image1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
var NX, NY, j, k, t3 : integer;
    selcon: Boolean;
begin

```

```

if Button <> mbLeft then exit;
if Button = mbRight then exit;
t3:=Koeff_Mashtab div 5;
Mouse_Status:= false;
if TImage(Sender).tag=0 then exit;
selcon:= false;
j:=1;
if Connect_Status then
begin
x:=TImage(Sender).Left+x;
y:=TImage(Sender).Top+y;
for k:=1 to Last_Elem do
begin
nx:=x-Elements[k].Image.Left;
ny:=y-Elements[k].Image.Top;
if (nx<Koeff_Mashtab) or (nx>(Elements[k].Size.x-
1)*Koeff_Mashtab) then
for j:=0 to High(Elements[k].Legs) do
if (nx>(Elements[k].Legs[j].x-1)*Koeff_Mashtab)
and (nx<(Elements[k].Legs[j].x+1)*Koeff_Mashtab)
and (ny>(Elements[k].Legs[j].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
and
(ny<(Elements[k].Legs[j].y)*Koeff_Mashtab+Koeff_Mashtab div 2) then begin
selcon:=true; break; end;
if selcon then break;
end;
TImage(Sender).Cursor:=crDefault;
Connect_Status:=false;
if (Conn_Prov<>0) then
begin
Conn_Prov:=0;
Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.Pen.Width:=t3;

AddConnect(Connect_Start.Elem,Connect_Start.Leg,Conns[MField[Conn_Prov_x,Conn_Pr
ov_y]][0].Elem,Conns[MField[Conn_Prov_x,Conn_Prov_y]][0].Leg);
Button4Click(nil);
end;
if Legs_sel2 then
begin
Draw_Leg(Legs_sel_Elem2,Legs_sel_Leg2);
Legs_sel2:=false;
end;
if selcon then
begin
if (Connect_Start.Elem=k) and (Connect_Start.Leg=j) then exit;
AddConnect(Connect_Start.Elem,Connect_Start.Leg,k,j);
Label1.Caption:=inttostr(High(Conns)+1);
if Legs_sel2 then
begin
Draw_Leg(Legs_sel_Elem2,Legs_sel_Leg2);
Legs_sel2:=false;
end;
Button4Click(nil);
end;
exit;
end;
if not (ssCtrl in Shift) then
begin
if not Elements[TImage(Sender).tag].Selected then
begin
DeselectElements;

```

```

        Elements[TImage(Sender).tag].Swich_Selection;
    end
end else Elements[TImage(Sender).tag].Swich_Selection;
Button4Click(nil);
end;

procedure TForm1.Image1MouseMove(Sender: TObject; Shift: TShiftState; X,
Y: Integer);
var NX, NY, i, j, k, t3 : integer;
    selcon: Boolean;
begin
    t3:=Koeff_Mashtab div 5;
    if TImage(Sender).tag=0 then exit;
    selcon:= false;
    if Connect_Status then
        begin
            x:=TImage(Sender).Left+x;
            y:=TImage(Sender).Top+y;
            if Legs_sel2 then
                begin
                    Draw_Leg(Legs_sel_Elem2,Legs_sel_Leg2);
                    Legs_sel2:=false;
                end;
            for k:=1 to Last_Elem do
                begin
                    nx:=x-Elements[k].Image.Left;
                    ny:=y-Elements[k].Image.Top;
                    if (nx<Koeff_Mashtab) or (nx>(Elements[k].Size.x-
1)*Koeff_Mashtab) then
                        for j:=0 to High(Elements[k].Legs) do
                            if (nx>(Elements[k].Legs[j].x-1)*Koeff_Mashtab)
                                and (nx<(Elements[k].Legs[j].x+1)*Koeff_Mashtab)
                                    and (ny>(Elements[k].Legs[j].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
                                        and
(ny<(Elements[k].Legs[j].y)*Koeff_Mashtab+Koeff_Mashtab div 2) then begin
                                selcon:=true; break; end;
                            if selcon then break;
                        end;
                    if selcon then
                        begin
                            Draw_Leg(Elements[k].Image,j);
                            Elements[k].Image.Repaint;
                            Legs_sel2:=true;
                            Legs_sel_Elem2:=Elements[k].Image;
                            Legs_sel_Leg2:=j;
                            exit;
                        end;
                    if not selcon then
                        begin
                            if MField[(x + Koeff_Mashtab div 2) div Koeff_Mashtab, (y +
Koeff_Mashtab div 2) div Koeff_Mashtab] <> -1 then
                                begin
                                    if (Conn_Prov<>0) then
                                        begin
                                            if (Conn_Prov_x = (x + Koeff_Mashtab div 2) div
Koeff_Mashtab) and (Conn_Prov_y = (y + Koeff_Mashtab div 2) div Koeff_Mashtab)
                                                then
                                                    else begin
                                                        Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Conn_Prov:=Field[(x + Koeff_Mashtab div 2) div
Koeff_Mashtab, (y + Koeff_Mashtab div 2) div Koeff_Mashtab];

```

```

Conn_Prov_x := (x + Koeff_Mashtab div 2) div
Koeff_Mashtab;
Conn_Prov_y := (y + Koeff_Mashtab div 2) div
Koeff_Mashtab;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);
    Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
    end
end else begin
    Conn_Prov_x := (x + Koeff_Mashtab div 2) div
Koeff_Mashtab;
    Conn_Prov_y := (y + Koeff_Mashtab div 2) div
Koeff_Mashtab;
    Conn_Prov:=Field[Conn_Prov_x,Conn_Prov_y];
    Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);
    Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
    end;
end else begin
    if (Conn_Prov<>0) then
    begin
        Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);

Form1.Frame21.Image1.Canvas.lineto(Conn_Prov_x*Koeff_Mashtab,Conn_Prov_y*Koeff_M
ashtab);
        end;
        Conn_Prov:=0;
        end;
        end;
        exit;
        end;
        if Mouse_Status then
        begin
            for i:=1 to Last_Elem do
                if (Elements[i].Selected) and
                (Elements[TImage(Sender).tag].Selected) then
                    with Elements[i].Image do begin
                        NX:=Left+X-M_x;
                        NY:=Top+Y-M_y;
                        if (NX>0) and (NX<Frame21.Pane11.Width-Width) then Left:=NX
                        div Koeff_Mashtab * Koeff_Mashtab +2;
                        if (NY>0) and (NY<Frame21.Pane11.Height-Height) then Top:=NY
                        div Koeff_Mashtab * Koeff_Mashtab +2;
                        Elements[Tag].Coord.x:=Elements[Tag].Image.Left div
                        Koeff_Mashtab;
                        Elements[Tag].Coord.y:=Elements[Tag].Image.Top div
                        Koeff_Mashtab;
                    end;
                    if Draw_connections then Button4Click(nil);
                    exit;
                end;
            end;
            if Legs_sel then
            begin
                Legs_sel:=false;
                Draw_Leg(Legs_sel_Elem,Legs_sel_Leg);
            end;
        end;
    end;
end;

```

```

        if (x<Koeff_Mashtab) or (x>(Elements[TImage(Sender).tag].Size.x-
1)*Koeff_Mashtab) then
            for i:=0 to High(Elements[TImage(Sender).tag].Legs) do
                if (x>(Elements[TImage(Sender).tag].Legs[i].x-1)*Koeff_Mashtab
                    and (x<(Elements[TImage(Sender).tag].Legs[i].x+1)*Koeff_Mashtab)
                    and (y>(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab-
Koeff_Mashtab div 2)
                        and
(y<(Elements[TImage(Sender).tag].Legs[i].y)*Koeff_Mashtab+Koeff_Mashtab div 2)
then begin selcon:=true; break; end;
                if selcon then
                    begin
                        Legs_sel:=true;
                        Legs_sel_Elem:=Sender;
                        Legs_sel_Leg:=i;
                        Draw_Leg(Sender,i);
                    end;
                if selcon then exit;
            end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    Chips_count := 0;
    Last_Elem:=0;
    Koeff_Mashtab:=10;
    Old_Mashtab := 10;
    Selected_Connection:=-1;
    Connect;
    SetLength(Cons,0);
    SetLength(Conns,0);
    SaveDialog1.InitialDir:=ExtractFilePath(Application.ExeName);
    OpenFileDialog1.InitialDir:=ExtractFilePath(Application.ExeName);
end;

procedure TForm1.FormMouseWheel(Sender: TObject; Shift: TShiftState;
    WheelDelta: Integer; MousePos: TPoint; var Handled: Boolean);
begin
    if Selection_Status then begin DrawSelectionRectangle
(Old_Second_Selection_Point); end;
    Selection_Status := false;
    if not (ssShift in Shift) then begin
Frame21.VertScrollBar.Position:=Frame21.VertScrollBar.Position-WheelDelta div 8;
Abort; end;
    if ssShift in Shift then begin
Frame21.HorzScrollBar.Position:=Frame21.HorzScrollBar.Position-WheelDelta div 8;
Abort; end;
    end;

procedure TForm1.FormActivate(Sender: TObject);
begin
    // InitCfg;
    // Form1.Align := alClient;
    Form1.Align := alNone;
    // DesktopSetup;
end;

procedure TForm1.Frame21Image1Click(Sender: TObject);
begin
    DeselectElements;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin
    DeleteSelectedElements;
end;

//виділити все
procedure TForm1.N3Click(Sender: TObject);
begin

```



```

        ((Elements[i].Image.Top > Left_Up.Y) and
(Elements[i].Image.Top < Right_Down.Y)) or
        (((Elements[i].Image.Left + Elements[i].Image.Width >
Left_Up.X) and (Elements[i].Image.Left + Elements[i].Image.Width <
Right_Down.X)) and
        ((Elements[i].Image.Top + Elements[i].Image.Height > Left_Up.Y)
and (Elements[i].Image.Top + Elements[i].Image.Height < Right_Down.Y))) or
        (((Elements[i].Image.Left > Left_Up.X) and
(Elements[i].Image.Left < Right_Down.X)) and
        ((Elements[i].Image.Top + Elements[i].Image.Height > Left_Up.Y)
and (Elements[i].Image.Top + Elements[i].Image.Height < Right_Down.Y))) then
            Elements[i].Select;
        end;
    end else DeselectElements;
    Second_Selection_Point.X := 0;
    Second_Selection_Point.Y := 0;
    Base_Selection_Point.X := 0;
    Base_Selection_Point.Y := 0;
end;

//відкриття вікна додавання мікросхеми
procedure TForm1.N8Click(Sender: TObject);
begin
    form13.ShowModal;
end;

//видалити вибрані елементи
procedure TForm1.N7Click(Sender: TObject);
begin
    Button2Click(Sender);
    AssociateTree;
    Button4Click(Sender);
end;

procedure TForm1.Panel3Click(Sender: TObject);
var i : integer;
begin
    for i := 1 to 135 do
        begin
            panel2.width := panel2.width-1;
        end;
    Panel4.Visible := true;
end;

procedure TForm1.Panel4Click(Sender: TObject);
var i: integer;
begin
    for i := 1 to 135 do
        begin
            panel2.width := panel2.width+1;
        end;
    panel4.Visible := false;
end;

//відкриття вікна вибору параметрів програми
procedure TForm1.N10Click(Sender: TObject);
begin
    Form4.ShowModal;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin
    SaveCfg;
    F_Stop:=true;
end;

procedure TForm1.ComboBox1Change(Sender: TObject);
begin
    Old_Masshtab := Koeff_Mashtab;

```

```

    Koeff_Mashtab := (ComboBox1.ItemIndex + 3);
    DesktopSetup;
    RepaintElements;
end;

//відкриття бази даних зображень елементів
procedure TForm1.N11Click(Sender: TObject);
begin
    form5.ShowModal;
end;

//відкриття бази даних елементів
procedure TForm1.N12Click(Sender: TObject);
begin
    form7.ShowModal;
end;

procedure TForm1.DrawField;
var i,j: integer;
    t1,t2,t3:integer;
begin
    t1:=Koeff_Mashtab div 2;
    t2:=Koeff_Mashtab div 2 - 1;
    t3:=Koeff_Mashtab div 5;
    Form1.Frame21.Image1.Canvas.Pen.Mode:=pmXor;
    Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
    Form1.Frame21.Image1.Canvas.Pen.Color:=clWhite;
    for i:=0 to 200 do
        for j:=0 to 200 do begin
            if (Field[i,j]>=10000) and (Field[i,j]<=10010) then
                if MField[i,j] = Selected_Connection then
                    Form1.Frame21.Image1.Canvas.Pen.Color:=Con_Color xor
                    Form1.Frame21.Image1.Canvas.Pixels[1,1]
                    else Form1.Frame21.Image1.Canvas.Pen.Color:=clWhite;
                case Field[i,j] of
                    10000: begin
                        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
                        t1,j*Koeff_Mashtab);
                    Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);
                    end;
                    10001: begin
                    Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);
                    Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
                    end;
                    10002: begin
                        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
                        t1,j*Koeff_Mashtab);
                    Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
                    Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);
                    end;
                    10003: begin
                        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
                        t1,j*Koeff_Mashtab);
                    Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
                    Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
                    end;
                    10004: begin
                    Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);
                    Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);

```

```

Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);
    end;
    10005: begin

Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);

Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);

Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
    end;
    10006: begin
        Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab-
t1,j*Koeff_Mashtab);

Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab+t2,j*Koeff_Mashtab);

Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab,j*Koeff_Mashtab-t1);

Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab+t2);
    end;
    10010: begin
        Form1.Frame21.Image1.Canvas.Pen.Width:=t3*5;

Form1.Frame21.Image1.Canvas.moveto(i*Koeff_Mashtab,j*Koeff_Mashtab);

Form1.Frame21.Image1.Canvas.lineto(i*Koeff_Mashtab,j*Koeff_Mashtab);
        Form1.Frame21.Image1.Canvas.Pen.Width:=t3;
    end;
    end;
end;

procedure TForm1.Button4Click(Sender: TObject);
var i,j:integer;
begin
    DrawField;
    SetLength(Cons,High(Conns)+1);
    for i:=0 to High(Conns) do
        begin
            SetLength(Cons[i],High(Conns[i])+1);
            for j:= 0 to High(Conns[i]) do
                with Cons[i][j] do
                    begin

X:=elements[Conns[i][j].Elem].Coord.x+Elements[Conns[i][j].Elem].Legs[Conns[i][j].Leg].x;

Y:=elements[Conns[i][j].Elem].Coord.y+Elements[Conns[i][j].Elem].Legs[Conns[i][j].Leg].y;
                    end;
                end;
            Connect;
            DrawField;
        end;

//Відкриття бази даних мікросхем
procedure TForm1.N14Click(Sender: TObject);
begin
    form10.ShowModal;
end;

procedure TForm1.Frame21Image1MouseMove(Sender: TObject;
    Shift: TShiftState; X, Y: Integer);
begin
    if Legs_sel then
        begin
            Legs_sel:=false;
            Draw_Leg(Legs_sel_Elem,Legs_sel_Leg);
        end;
end;

```

```

end;
if Selection_Status then
begin
Second_Selection_Point.X := X;
Second_Selection_Point.Y := Y;
DrawSelectionRectangle (Second_Selection_Point);
DrawSelectionRectangle (Old_Second_Selection_Point);
Old_Second_Selection_Point := Second_Selection_Point;
end;
if MField[(x + Koeff_Mashtab div 2) div Koeff_Mashtab, (y +
Koeff_Mashtab div 2) div Koeff_Mashtab]<>Selected_Connection then
begin
if Select_connections then DrawField;
Selected_Connection:=MField[(x + Koeff_Mashtab div 2) div
Koeff_Mashtab, (y + Koeff_Mashtab div 2) div Koeff_Mashtab];
if Select_connections then DrawField;
end;
end;

//додати елемент чи мікросхему
procedure TForm1.Button3Click(Sender: TObject);
begin
PopupMenu1.Popup(Left+Button3.Left+Button3.Width div
2+Panel2.Left,Button3.Top+Button3.Height+Top+Panel2.Top+24);
end;

//видалити елемент чи мікросхему
procedure TForm1.Button5Click(Sender: TObject);
var i : integer;
begin
if TreeView1.Items.Count<1 then exit;
if TreeView1.Selected.Parent = nil then
begin
DeselectElements;
for i:=0 to TreeView1.Items.Count - 1 do
begin
if (TreeView1.Items[i].Parent = TreeView1.Selected) and
(TreeView1.items[i].imageindex = 1) then
begin
telement(TreeView1.items[i].Data^).Select;
end;
end;
DeleteSelectedElements;
TreeView1.Selected.Free;
AssociateTree;
end
else
if TreeView1.Selected.ImageIndex = 1 then
begin
DeselectElements;
telement(TreeView1.Selected.Data^).Select;
DeleteSelectedElements;
TreeView1.Selected.ImageIndex := 2;
TreeView1.Selected.SelectedIndex := 2;
AssociateTree;
end;
Button4Click(Sender);
end;

procedure TForm1.TreeView1Db1Click(Sender: TObject);
var i, j : integer;
legs1, legs2: String;
begin
if TreeView1.Items.Count = 0 then exit;
if TreeView1.Selected.ImageIndex <= 1 then exit;
Last_Elem:=Last_Elem+1;
Elements[Last_Elem].create;
legs1:='';
legs2:='';

```

```

if TreeView1.Selected.Text[1]='D' then
begin
  form10.Table1.first;
  for i := 0 to form10.Table1.RecordCount - 1 do
  begin
    if TreeView1.Selected.Parent.Text =
form10.Table1.fieldbyname('Mark').asstring then break;
    form10.Table1.Next;
    end;
    Elements[Last_Elem].Image.Picture := form5.dbimage1.picture;
    Elements[Last_Elem].Image.Canvas.Font.Size := 10;
    Elements[Last_Elem].Image.Canvas.Font.Style := [fsBold];
    Elements[Last_Elem].Image.Canvas.Brush.Style := bsSolid;
    Elements[Last_Elem].Image.Canvas.Brush.Color:= $F0F0F0;

Elements[Last_Elem].Image.Canvas.TextOut(Elements[Last_Elem].Image.Picture.Bitmap.Width div 2 -
Elements[Last_Elem].Image.Canvas.TextWidth(TreeView1.Selected.Text) div 2,
Elements[Last_Elem].Image.Picture.Bitmap.Height - 20, TreeView1.Selected.Text);
    legs1:=Form7.DBEdit3.Text;
    legs2:=Form7.DBEdit4.Text;
    Elements[Last_Elem].Reaction_LH:=strtoint(Form10.DBEdit4.Text);
    Elements[Last_Elem].Reaction_HL:=strtoint(Form10.DBEdit6.Text);
    end;

if TreeView1.Selected.Text[1]='И' then
begin
  ImageList2.GetBitmap(0,Elements[Last_Elem].Image.Picture.Bitmap);
  legs1:='1';
  Elements[Last_Elem].Special_bool:=false;
  end;
if TreeView1.Selected.Text[1]='И' then
begin
  ImageList2.GetBitmap(2,Elements[Last_Elem].Image.Picture.Bitmap);
  legs2:='1';
  Elements[Last_Elem].Image.OnDblClick:=Image1.OnDblClick;
  Elements[Last_Elem].Special_bool:=false;
  end;
if TreeView1.Selected.Text[1]='Г' then
begin
  ImageList4.GetBitmap(0,Elements[Last_Elem].Image.Picture.Bitmap);
  legs2:='010001';
  Elements[Last_Elem].Image.OnDblClick:=Image1.OnDblClick;
  Elements[Last_Elem].Reaction_LH:=100;
  Elements[Last_Elem].Reaction_HL:=100;
  end;
if TreeView1.Selected.Text[1]='O' then
begin
  ImageList4.GetBitmap(1,Elements[Last_Elem].Image.Picture.Bitmap);
  legs1:='0001';
  Elements[Last_Elem].Image.OnDblClick:=Image1.OnDblClick;
  Elements[Last_Elem].Special_bool:=false;
  Oscill[strtoint(copy(TreeView1.Selected.Text,12,2))]:=Last_Elem;
  if Form17.Visible then
    begin
Form17.CheckListBox1.Checked[strtoint(copy(TreeView1.Selected.Text,12,2))-
1]:=true;
      Form17.CheckListBox1ClickCheck(nil);
    end;
  end;

TreeView1.Selected.ImageIndex := 1;
TreeView1.Selected.SelectedIndex := 1;
TreeView1.Selected.Data := @elements[last_elem];

Elements[Last_Elem].Script.AddStrings(Form7.DBMemol.Lines);
Elements[Last_Elem].Caption := TreeView1.Selected.Text;
Elements[Last_Elem].Image.tag := Last_Elem;

```

```

Elements[Last_Elem].Image.Left:=2;
Elements[Last_Elem].Image.Top:=2;

Elements[Last_Elem].Image.Width:=Elements[Last_Elem].Image.Picture.Bitmap.Width*
Koeff_Mashtab div 10;

Elements[Last_Elem].Image.Height:=Elements[Last_Elem].Image.Picture.Bitmap.Heigh
t*Koeff_Mashtab div 10;
Elements[Last_Elem].Image.Proportional:=true;
Elements[Last_Elem].Image.Transparent:=true;
Elements[Last_Elem].Image.Stretch:=true;
Elements[Last_Elem].Image.Parent:=Frame21.Panell1;
Elements[Last_Elem].Image.OnMouseDown:=Imagel.OnMouseDown;
Elements[Last_Elem].Image.OnMouseMove:=Imagel.OnMouseMove;
Elements[Last_Elem].Image.OnMouseUp:=Imagel.OnMouseUp;

Elements[Last_Elem].Size.x:=Elements[Last_Elem].Image.Picture.Bitmap.Width div
10;

Elements[Last_Elem].Size.y:=Elements[Last_Elem].Image.Picture.Bitmap.Height div
10;
Elements[Last_Elem].Coord.x:=Elements[Last_Elem].Image.Left div
Koeff_Mashtab;
Elements[Last_Elem].Coord.y:=Elements[Last_Elem].Image.Top div
Koeff_Mashtab;
if Select_on_creation then Elements[Last_Elem].Select;

j:=0;
for i:=1 to Length(legs1) do
  if legs1[i]='1' then j:=j+1;
for i:=1 to Length(legs2) do
  if legs2[i]='1' then j:=j+1;
SetLength(Elements[Last_Elem].Legs,j);
j:=0;
for i:=1 to Length(legs1) do
  if legs1[i]='1' then
    with Elements[Last_Elem] do begin
      Legs[j].x:=0;
      Legs[j].y:=i;
      j:=j+1;
    end;
for i:=1 to Length(legs2) do
  if legs2[i]='1' then
    with Elements[Last_Elem] do begin
      Legs[j].x:=Size.x;
      Legs[j].y:=i;
      j:=j+1;
    end;
  InitTrace;
end;

procedure TForm1.TreeView1Click(Sender: TObject);
var i : integer;
begin
  if TreeView1.Items.Count = 0 then exit;
  DeselectElements;
  if TreeView1.Selected.ImageIndex = 1 then
    telement(TreeView1.Selected.Data^).Select;
  if (TreeView1.Selected.ImageIndex = 0) and
(TreeView1.Selected.Text[1]<>'C') then
    begin
      for i := 0 to TreeView1.Items.Count -1 do
        if (TreeView1.Items[i].Parent = TreeView1.Selected) and
(TreeView1.Items[i].ImageIndex = 1) then
          telement(TreeView1.items[i].Data^).Select;
    end;
end;

procedure TForm1.N19Click(Sender: TObject);

```

```

begin
    form13.ShowModal;
end;

//додати індикатор
procedure TForm1.N21Click(Sender: TObject);
var i,j,k:integer;
    t,l:TTreeNode;
begin
    t:=nil;
    for i:=0 to TreeView1.Items.Count-1 do
        if TreeView1.Items[i].Text='Спец. елементи' then
            begin
                t:=TreeView1.Items[i];
                break;
            end;
    if (t=nil) or (TreeView1.Items.Count=0) then
        t:=TreeView1.Items.AddChild(nil, 'Спец. елементи');
    j:=0;
    k:=0;
    for i:=1 to TreeView1.Items.Count-1 do
        if copy(TreeView1.Items[i].Text,1,9)='Індикатор' then
            begin
                if j<strtoint(copy(TreeView1.Items[i].Text,10,2)) then
                    j:=strtoint(copy(TreeView1.Items[i].Text,10,2));
                    k:=i;
                end;
                j:=j+1;
                if k=0 then l:=TreeView1.Items.AddChild(t, 'Індикатор1')
                    else if k=TreeView1.Items.Count-1 then
                        l:=TreeView1.Items.Add(TreeView1.Items[k], 'Індикатор'+inttostr(j))
                    else
                        l:=TreeView1.Items.Insert(TreeView1.Items[k+1], 'Індикатор'+inttostr(j));
                l.ImageIndex:=2;
                l.SelectedIndex:=2;
            end;

    procedure TForm1.N24Click(Sender: TObject);
    var i,j,k:integer;
        t,l:TTreeNode;
    begin
        t:=nil;
        for i:=0 to TreeView1.Items.Count-1 do
            if TreeView1.Items[i].Text='Спец. елементи' then
                begin
                    t:=TreeView1.Items[i];
                    break;
                end;
        if (t=nil) or (TreeView1.Items.Count=0) then
            t:=TreeView1.Items.AddChild(nil, 'Спец. елементи');
        j:=0;
        k:=0;
        for i:=1 to TreeView1.Items.Count-1 do
            if copy(TreeView1.Items[i].Text,1,13)='Перемикач' then
                begin
                    if j<strtoint(copy(TreeView1.Items[i].Text,14,2)) then
                        j:=strtoint(copy(TreeView1.Items[i].Text,14,2));
                        k:=i;
                    end;
                    j:=j+1;
                    if k=0 then l:=TreeView1.Items.AddChild(t, 'Перемикач1')
                        else if k=TreeView1.Items.Count-1 then
                            l:=TreeView1.Items.Add(TreeView1.Items[k], 'Перемикач'+inttostr(j))
                        else
                            l:=TreeView1.Items.Insert(TreeView1.Items[k+1], 'Перемикач'+inttostr(j));
                    l.ImageIndex:=2;
                    l.SelectedIndex:=2;
                end;
    end;

```

```

procedure TForm1.Image1Db1Click(Sender: TObject);
begin
  if Elements[TImage(Sender).Tag].Caption[1]='П' then
  begin
    Elements[TImage(Sender).Tag].Special_bool := not
Elements[TImage(Sender).Tag].Special_bool;
    if Elements[TImage(Sender).Tag].Special_bool then

ImageList2.GetBitmap(3,Elements[TImage(Sender).Tag].Image.Picture.Bitmap)
    else
ImageList2.GetBitmap(2,Elements[TImage(Sender).Tag].Image.Picture.Bitmap);
    TImage(Sender).Repaint;
    Elements[TImage(Sender).Tag].Selected:=false;
  end;
  if Elements[TImage(Sender).Tag].Caption[1]='O' then
  begin
    UpdateOsc;
    Form17.Show;
  end;
  if Elements[TImage(Sender).Tag].Caption[1]='Г' then
  begin
    F_Pause:=true;

Form16.Edit1.Text:=inttostr(Elements[TImage(Sender).Tag].Reaction_HL);

Form16.Edit2.Text:=inttostr(Elements[TImage(Sender).Tag].Reaction_LH);
    if Form16.ShowModal=mrOk then
    begin
      try
        Elements[TImage(Sender).Tag].Reaction_HL:=strtoint(Form16.Edit1.Text);
        Elements[TImage(Sender).Tag].Reaction_LH:=strtoint(Form16.Edit2.Text);
      except
        end;
    end;
    Elements[TImage(Sender).Tag].DeSelect;
    F_Pause:=false;
  end;
end;

procedure TForm1.N25Click(Sender: TObject);
var j: Integer;
begin
  SetLength(Conns[Selected_Connection],0);
  for j:= Selected_Connection to High(Conns)-1 do
    Conns[j]:=Conns[j+1];
  SetLength(Conns,High(Conns));
  Label1.Caption:=inttostr(High(Conns)+1);
  Button4Click(Sender);
  InitTrace;
end;

procedure TForm1.N26Click(Sender: TObject);
var l,j,k : Integer;
begin
  for j:=0 to High(Conns) do
  begin
    k:=0;
    while k<=High(Conns[j]) do
    begin
      if (Conns[j][k].Elem=Connect_Start.Elem) and
(Conns[j][k].Leg=Connect_Start.Leg) then
      begin
        for l:=k to High(Conns[j])-1 do
          Conns[j][l]:=Conns[j][l+1];
        k:=k-1;
        SetLength(Conns[j],High(Conns[j]));
      end;
    end;
  end;
end;

```

```

        k:=k+1;
    end;
end;
end;
j:=0;
while j<=High(Conns) do
begin
    if High(Conns[j])<1 then
    begin
        for l:=j to High(Conns)-1 do
            Conns[l]:=Conns[l+1];
            SetLength(Conns,High(Conns));
        end;
        j:=j+1;
    end;
    Button4Click(nil);
    InitTrace;
end;

//запуск процесу емуляції роботи створеної цифрової схеми
procedure TForm1.ToolButton6Click(Sender: TObject);
begin
    F_Stop:=false;
    F_Pause:=false;
    ToolButton6.Enabled:=false;
    ToolButton7.Enabled:=true;
    ToolButton8.Enabled:=true;

    InitTrace;
    // TrTime:=0;
    repeat
        Delay(Form15.TrackBar1.Position);
        if not F_Pause then Trace;
        if not F_Pause then Trace;
        if not F_Pause then TrTime := TrTime + 1;
        if (Osc_Count>0) and Form17.Visible {and
Form17.CheckListBox1.Checked[0]} then Form17.Button2Click(nil);
        Application.ProcessMessages;
    until F_Stop;
    ToolButton6.Enabled:=true;
    ToolButton7.Enabled:=false;
    ToolButton8.Enabled:=false;
end;

//стоп
procedure TForm1.ToolButton7Click(Sender: TObject);
begin
    F_Stop:=true;
end;

//пауза
procedure TForm1.ToolButton8Click(Sender: TObject);
begin
    F_Pause:=not F_Pause;
end;

//вибір масштабу часу
procedure TForm1.ToolButton9Click(Sender: TObject);
begin
    Form15.Visible:=ToolButton9.Down;
    SetFocus;
end;

procedure TForm1.N27Click(Sender: TObject);
var i,j,k:integer;
    t,l:TTreeNode;
begin
    t:=nil;
    for i:=0 to TreeView1.Items.Count-1 do
        if TreeView1.Items[i].Text='Спец. елементи' then

```

```

begin
    t:=TreeView1.Items[i];
    break;
end;
if (t=nil) or (TreeView1.Items.Count=0) then
    t:=TreeView1.Items.AddChild(nil, 'Спец. элементы');
j:=0;
k:=0;
for i:=1 to TreeView1.Items.Count-1 do
    if copy(TreeView1.Items[i].Text,1,9)='Генератор' then
        begin
            if j<strtoint(copy(TreeView1.Items[i].Text,10,2)) then
j:=strtoint(copy(TreeView1.Items[i].Text,10,2));
                k:=i;
            end;
            j:=j+1;
            if k=0 then l:=TreeView1.Items.AddChild(t, 'Генератор1')
                else if k=TreeView1.Items.Count-1 then
l:=TreeView1.Items.Add(TreeView1.Items[k], 'Генератор'+inttostr(j))
                    else
l:=TreeView1.Items.Insert(TreeView1.Items[k+1], 'Генератор'+inttostr(j));
                        l.ImageIndex:=2;
                        l.SelectedIndex:=2;
                    end;

procedure TForm1.N28Click(Sender: TObject);
var i,j,k:integer;
    t,l:TTreeNode;
begin
    t:=nil;
    for i:=0 to TreeView1.Items.Count-1 do
        if TreeView1.Items[i].Text='Спец. элементы' then
            begin
                t:=TreeView1.Items[i];
                break;
            end;
        if (t=nil) or (TreeView1.Items.Count=0) then
            t:=TreeView1.Items.AddChild(nil, 'Спец. элементы');
        j:=0;
        k:=0;
        for i:=1 to TreeView1.Items.Count-1 do
            if copy(TreeView1.Items[i].Text,1,11)='Осциллограф' then
                begin
                    if j<strtoint(copy(TreeView1.Items[i].Text,12,2)) then
j:=strtoint(copy(TreeView1.Items[i].Text,12,2));
                            k:=i;
                    end;
                    if j=5 then exit;
                    j:=j+1;
                    if k=0 then l:=TreeView1.Items.AddChild(t, 'Осциллограф1')
                        else if k=TreeView1.Items.Count-1 then
l:=TreeView1.Items.Add(TreeView1.Items[k], 'Осциллограф'+inttostr(j))
                            else
l:=TreeView1.Items.Insert(TreeView1.Items[k+1], 'Осциллограф'+inttostr(j));
                                if k=0 then Osc_Count:=1 else Osc_Count:=j;
                                l.ImageIndex:=2;
                                l.SelectedIndex:=2;
                                if Form17.Visible then
                                    begin
                                        UpdateOsc;
                                        Form17.CheckListBox1ClickCheck(nil);
                                    end;
                                end;
                            end;

//осциллографи
procedure TForm1.N15Click(Sender: TObject);
begin
    if Osc_Count < 1 then Exit;
    UpdateOsc;

```

```

    Form17.Show;
end;

procedure TForm1.FormShow(Sender: TObject);
begin
    InitCfg;
    DesktopSetup;
end;

//збереження цифрової схеми у файл
procedure TForm1.Button6Click(Sender: TObject);
//var fEl : file of Telement;
var i, j : integer;
    fdata : TElemsave;
    CData : TConn;
    FN : string;
begin
    if not SaveDialog1.Execute then exit;
    FN := SaveDialog1.FileName;

TreeView1.SaveToFile (ExtractFilePath (FN)+copy (ExtractFileName (FN), 1, length (ExtractFileName (FN))-3)+'tsv');

assignfile (fio, ExtractFilePath (FN)+copy (ExtractFileName (FN), 1, length (ExtractFileName (FN))-3)+'esv');
rewrite (fio);
for i:= 1 to Last_Elem do
begin
    fdata.coord.X := elements[i].Coord.X;
    fdata.coord.Y := elements[i].Coord.Y;
    fdata.Caption := elements[i].Caption;
    fdata.React_HL := elements[i].Reaction_HL;
    fdata.React_LH := elements[i].Reaction_LH;
    write (fio, fdata);
end;
closefile (fio);

assignfile (fio2, ExtractFilePath (FN)+copy (ExtractFileName (FN), 1, length (ExtractFileName (FN))-3)+'csv');
rewrite (fio2);
for i:= 0 to High (Conns) do
begin
    for j:= 0 to High (Conns[i]) do
begin
        CData:=Conns[i][j];
        write (fio2, CData);
end;
    CData.Elem:=0;
    write (fio2, CData);
end;
closefile (fio2);
DesktopSetup;
Button4Click (nil);
end;

//завантаження цифрової схеми з файлу
procedure TForm1.Button7Click(Sender: TObject);
var i : integer;
    fdata : TElemsave;
    CData : TConn;
    FN : String;
begin
    if not OpenDialog1.Execute then exit;
    FN := OpenDialog1.FileName;
    Button8Click (nil);

```

```

TreeView1.LoadFromFile(ExtractFilePath(FN)+copy(ExtractFileName(FN),1,length(ExtractFileName(FN))-3)+'tsv');
  for i:= 0 to TreeView1.Items.Count -1 do
    if TreeView1.Items[i].Parent <> nil then
      begin
        TreeView1.Items[i].ImageIndex := 2;
        TreeView1.Items[i].SelectedIndex := 2;
      end;

assignfile(fio,ExtractFilePath(FN)+copy(ExtractFileName(FN),1,length(ExtractFileName(FN))-3)+'esv');
  reset(fio);
  while not EOF(fio) do
    begin
      read(fio, fdata);
      for i:= 0 to TreeView1.Items.Count - 1 do
        if TreeView1.Items[i].Text = fdata.Caption then
          begin
            TreeView1.Selected := TreeView1.Items[i];
            TreeView1Db1Click(nil);
            Elements[Last_Elem].Coord.x := fdata.coord.X;
            Elements[Last_Elem].Coord.Y := fdata.coord.Y;
            Elements[Last_Elem].Image.Left :=
fdata.coord.X*Koeff_Mashtab+2;
            Elements[Last_Elem].Image.top :=
fdata.coord.Y*Koeff_Mashtab+2;
            Elements[Last_Elem].Reaction_LH := fdata.React_LH;
            Elements[Last_Elem].Reaction_HL := fdata.React_HL;
            if Elements[Last_Elem].Caption[1] = 'O' then
Osc_Count:=Osc_Count+1;
              end;
            end;
          closefile (fio);

assignfile(fio2,ExtractFilePath(FN)+copy(ExtractFileName(FN),1,length(ExtractFileName(FN))-3)+'csv');
  reset(fio2);
  while not EOF(fio2) do
    begin
      SetLength(Conns,High(Conns)+2);
      i:=High(Conns);
      while not EOF(fio2) do
        begin
          read(fio2,cdata);
          if CData.Elem=0 then break;
          SetLength(Conns[i],High(Conns[i])+2);
          Conns[i][High(Conns[i])]:=CData;
        end;
      end;
    closefile (fio2);
    Button4Click(nil);
    UpdateOsc;
    DesktopSetup;
    Button4Click(nil);
  end;

//створення нового проекту
procedure TForm1.Button8Click(Sender: TObject);
var i: Integer;
begin
  TreeView1.Items.Clear;
  SelectAllElements;
  DeleteSelectedElements;
  Button4Click(nil);
  Osc_Count:=0;
  for i:=1 to 5 do
    Oscill[i]:=0;

```

```
Form17.Hide;  
end;  
  
//вихід з програми  
procedure TForm1.N34Click(Sender: TObject);  
begin  
    Close;  
end;  
  
//відкриття довідки  
procedure TForm1.N37Click(Sender: TObject);  
begin  
    Form19.Show;  
end;  
  
end.
```

Кафедра _ КБПЗ _ 2022 рік

Файл Parameters.pas - вікно вибору параметрів програми

```

unit Parameters;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm4 = class(TForm)
    Button1: TButton;
    Button2: TButton;
    Button3: TButton;
    Panell1: TPanel;
    GroupBox1: TGroupBox;
    CheckBox1: TCheckBox;
    GroupBox2: TGroupBox;
    ColorBox1: TColorBox;
    ColorBox2: TColorBox;
    Label4: TLabel;
    Label5: TLabel;
    CheckBox3: TCheckBox;
    CheckBox2: TCheckBox;
    CheckBox4: TCheckBox;
    Image1: TImage;
    procedure FormActivate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure ColorBox1Change(Sender: TObject);
    procedure ColorBox2Change(Sender: TObject);
    procedure CheckBox1Click(Sender: TObject);
    procedure CheckBox2Click(Sender: TObject);
    procedure CheckBox3Click(Sender: TObject);
    procedure CheckBox4Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var Form4: TForm4;
implementation

uses Main;
{$R *.dfm}

procedure TForm4.FormActivate(Sender: TObject);
begin
  CheckBox1.Checked := Selection_Metod;
  ColorBox1.Selected := Desktop_Color;
  ColorBox2.Selected := Grid_Color;
  CheckBox2.Checked := Select_on_creation;
  CheckBox3.Checked := Draw_connections;
  CheckBox4.Checked := Select_connections;
end;

//процедура, що обробляє натиснення на кнопку "Відміна"
procedure TForm4.Button2Click(Sender: TObject);
begin
  close
end;

//процедура, що обробляє натиснення на кнопку "Застосувати"
procedure TForm4.Button3Click(Sender: TObject);
begin

```

```
Form1.SaveCfg;
end;

//процедура, що обробляє натиснення на кнопку "ОК"
procedure TForm4.Button1Click(Sender: TObject);
begin
  Form1.SaveCfg;
  Close
end;

procedure TForm4.ColorBox1Change(Sender: TObject);
begin
  Desktop_Color := ColorBox1.Selected; //вибір кольору фону
  Form1.DesktopSetup;
end;

procedure TForm4.ColorBox2Change(Sender: TObject);
begin
  Grid_Color := ColorBox2.Selected; //вибір кольору розмітки
  Form1.DesktopSetup;
end;

procedure TForm4.CheckBox1Click(Sender: TObject);
begin
  Selection_Metod := CheckBox1.Checked; //якщо вибрано, то виділяти тільки
повністю обведений елемент
end;

procedure TForm4.CheckBox2Click(Sender: TObject);
begin
  Select_on_creation := CheckBox2.Checked; //якщо вибрано, то виділяти
елементи при додаванні в схему
end;

procedure TForm4.CheckBox3Click(Sender: TObject);
begin
  Draw_connections := CheckBox3.Checked; //якщо вибрано, то промальовувати
з'єднання при пересуванні
end;

procedure TForm4.CheckBox4Click(Sender: TObject);
begin
  Select_connections:= CheckBox4.Checked; //якщо вибрано, то виділяти
з'єднання кольором
end; end.
```

Файл Osc.pas - вікно осцилографів

```

unit Osc;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, Main8, StdCtrls, CheckLst, Main, Tracer;

type
  TForm17 = class(TForm)
    GroupBox2: TGroupBox;
    Frame181: TFrame18;
    Button2: TButton;
    CheckListBox1: TCheckListBox;
    procedure CheckListBox1ClickCheck(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form17: TForm17;

procedure UpdateOsc;

implementation

uses Unit3;

{$R *.dfm}

procedure UpdateOsc;
var i:integer;
begin
  with Form17 do begin
    Frame181.Panel1.Visible:=false;
    Frame181.Panel2.Visible:=false;
    Frame181.Panel3.Visible:=false;
    Frame181.Panel4.Visible:=false;
    Frame181.Panel5.Visible:=false;
    CheckListBox1.Items.Clear;
    for i := 1 to Osc_Count do
      begin
        CheckListBox1.Items.Add('Осцилограф'+inttostr(i));
        CheckListBox1.Checked[i-1]:=true;
      end;
    CheckListBox1ClickCheck(nil);
  end;
end;

procedure TForm17.CheckListBox1ClickCheck(Sender: TObject);
var i:integer;
label 1;
begin
  for i:=1 to CheckListBox1.Items.Count do
    if CheckListBox1.Checked[i-1] and (Oscill[i]=0) then
      CheckListBox1.Checked[i-1]:=false;
    i:=0;
  if Osc_Count<1 then goto 1;
  if CheckListBox1.Checked[0] then
    begin
      Frame181.Panel1.Visible:=true;
      Frame181.Panel1.Top:=i;
    end;
end;

```

```

        i:=i+60;
    end else Frame181.Panel1.Visible:=false;
if Osc_Count<2 then goto 1;
if CheckListBox1.Checked[1] then
begin
    Frame181.Panel2.Visible:=true;
    Frame181.Panel2.Top:=i;
    i:=i+60;
end else Frame181.Panel2.Visible:=false;
if Osc_Count<3 then goto 1;
if CheckListBox1.Checked[2] then
begin
    Frame181.Panel3.Visible:=true;
    Frame181.Panel3.Top:=i;
    i:=i+60;
end else Frame181.Panel3.Visible:=false;
if Osc_Count<4 then goto 1;
if CheckListBox1.Checked[3] then
begin
    Frame181.Panel4.Visible:=true;
    Frame181.Panel4.Top:=i;
    i:=i+60;
end else Frame181.Panel4.Visible:=false;
if Osc_Count<5 then goto 1;
if CheckListBox1.Checked[4] then
begin
    Frame181.Panel5.Visible:=true;
    Frame181.Panel5.Top:=i;
    i:=i+60;
end else Frame181.Panel5.Visible:=false;
1:
    if i<>0 then Form17.Height:=65+i;
end;

procedure TForm17.Button2Click(Sender: TObject);
var i,j,k:Integer;
begin
    if Frame181.Panel1.Visible then with Elements[Oscill[1]] do
begin
    j:=30;
    k:=0;
//    Frame181.ScrollBar1.Max:=TrTime-Frame181.Image1.Width;
    Frame181.Image1.Picture:=nil;
    if TrTime<Frame181.Image1.Width then
Frame181.Image1.Canvas.MoveTo(Frame181.Image1.Width-TrTime-2,j)
    else Frame181.Image1.Canvas.MoveTo(-1,j);
    for i:=0 to TrTime do
begin
        if High(Osc_Data)<>-1 then
            if Osc_Data[k]=i then
begin
                if j=30 then j:=5 else j:=30;
                k:=k+1;
            end;
            if TrTime-i<=Frame181.Image1.Width then
Frame181.Image1.Canvas.LineTo(Frame181.Image1.Width-TrTime+i-1,j);
            end;
        end;
    if Frame181.Panel2.Visible then with Elements[Oscill[2]] do
begin
    j:=30;
    k:=0;
//    Frame181.ScrollBar1.Max:=TrTime-Frame181.Image1.Width;
    Frame181.Image2.Picture:=nil;
    if TrTime<Frame181.Image2.Width then
Frame181.Image2.Canvas.MoveTo(Frame181.Image2.Width-TrTime-2,j)
    else Frame181.Image2.Canvas.MoveTo(-1,j);

```



```
        if Osc_Data[k]=i then
            begin
                if j=30 then j:=5 else j:=30;
                k:=k+1;
            end;
            if TrTime-i<=Frame181.Image5.Width then
                Frame181.Image5.Canvas.LineTo(Frame181.Image5.Width-TrTime+i-1,j);
            end;
        end;
    end;
end.
```

Кафедра _ КБПЗ _ 2022 рік

Файл Db_elements.pas - доступ до бази даних елементів

```

unit Db_elements;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, DB, DBTables, ExtCtrls, Grids, DBGrids, DBCtrls, StdCtrls, Mask,
  ComCtrls;

type
  TForm7 = class(TForm)
    DBGrid1: TDBGrid;
    Panel1: TPanel;
    Table1: TTable;
    DataSource1: TDataSource;
    Panel2: TPanel;
    Panel3: TPanel;
    Button5: TButton;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    GroupBox1: TGroupBox;
    DBMemo1: TDBMemo;
    TabSheet2: TTabSheet;
    GroupBox3: TGroupBox;
    GroupBox4: TGroupBox;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    Label3: TLabel;
    Label4: TLabel;
    GroupBox5: TGroupBox;
    Label5: TLabel;
    Button4: TButton;
    DBEdit1: TDBEdit;
    Panel4: TPanel;
    Image1: TImage;
    Button6: TButton;
    Button7: TButton;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure FindPicture;
    procedure DBGrid1KeyPress(Sender: TObject; var Key: Char);
    procedure DBGrid1KeyUp(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure DBGrid1CellClick(Column: TColumn);
    procedure Button4Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure DrawLegs;
    procedure DBEdit3Change(Sender: TObject);
    procedure DBEdit4Change(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button7Click(Sender: TObject);
    procedure Button6Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Table1AfterScroll(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form7: TForm7;

implementation

```

```

uses
  unit5, Unit8, Unit9;
{$R *.dfm}

Procedure TForm7.DrawLegs;
var i : integer;
begin
  if form5 <> nil then
    form7.Image1.Picture := form5.DBImage1.Picture;
  Image1.Canvas.Pen.Color := clRed;
  for i := 1 to length (DBEdit3.text) do
    begin
      if DBEdit3.text[i] = '1' then Image1.Canvas.Ellipse(0,i*10-3,5,i*10+3);
    end;
  for i := 1 to length (DBEdit4.text) do
    begin
      if DBEdit4.text[i] = '1' then
        Image1.Canvas.Ellipse(Image1.Picture.Bitmap.Width-6,i*10-
          3,Image1.Picture.Bitmap.Width,i*10+3);
    end;
  end;

Procedure TForm7.FindPicture;
var i: integer;
begin
  if form5.Table1.FieldName('Name').AsString =
    table1.FieldName('Image_Name').AsString then exit;
  form5.Table1.First;
  for i:= 1 to form5.Table1.RecordCount -1 do
    begin
      if form5.Table1.FieldName('Name').AsString =
        table1.FieldName('Image_Name').AsString then break;
      form5.Table1.Next;
    end;
  end;

procedure TForm7.FormCreate(Sender: TObject);
begin
  Table1.DatabaseName := ExtractFilePath(Application.ExeName)+'bases\';
  Table1.TableName := 'elements.db';
  Table1.Active := true;
end;

procedure TForm7.DBGrid1KeyPress(Sender: TObject; var Key: Char);
begin
  // if key = #72 then FindPicture;
end;

procedure TForm7.DBGrid1KeyUp(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  if (key = VK_UP) or (key = VK_DOWN) then FindPicture;
end;

procedure TForm7.DBGrid1CellClick(Column: TColumn);
begin
  FindPicture;
end;

procedure TForm7.Button4Click(Sender: TObject);
begin
  form8.ShowModal;
end;

procedure TForm7.FormActivate(Sender: TObject);
begin
  FindPicture;  form7.Image1.Picture := form5.DBImage1.Picture;
  DrawLegs;

```

```
end;

procedure TForm7.DBEdit3Change(Sender: TObject);
begin
  DrawLegs;
end;

procedure TForm7.DBEdit4Change(Sender: TObject);
begin
  DrawLegs;
end;

procedure TForm7.Button5Click(Sender: TObject);
begin
  close;
  Table1.Edit;
  Table1.Post;
end;

procedure TForm7.Button7Click(Sender: TObject);
begin
  if Table1.RecordCount > 0 then
  begin
    if Application.MessageBox('Ви дійсно хочете видалити цей елемент',
'Увага', MB_YESNO) = IDYES then
      Table1.Delete;
    end;
  end;
end;

procedure TForm7.Button6Click(Sender: TObject);
begin
  form9.ShowModal;
end;

procedure TForm7.Button1Click(Sender: TObject);
begin
  Table1.Edit;
  Table1.Post;
end;

procedure TForm7.Table1AfterScroll(DataSet: TDataSet);
begin
  if form5 <> nil then
  begin
    FindPicture;    form7.Image1.Picture := form5.DBImage1.Picture;
    DrawLegs;
  end
end;

end.
```

Файл Db_micro.pas – доступ до бази даних мікросхем

```

unit Db_micro;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ComCtrls, Grids, DBGrids, DB, DBTables, ExtCtrls, StdCtrls,
  Mask, DBCtrls;

type
  TForm10 = class(TForm)
    Panel1: TPanel;
    Table1: TTable;
    DataSource1: TDataSource;
    DBGrid1: TDBGrid;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    TabSheet2: TTabSheet;
    GroupBox1: TGroupBox;
    DBEdit1: TDBEdit;
    Label1: TLabel;
    Button1: TButton;
    DBEdit2: TDBEdit;
    Label2: TLabel;
    GroupBox2: TGroupBox;
    GroupBox3: TGroupBox;
    DBEdit3: TDBEdit;
    DBEdit4: TDBEdit;
    GroupBox4: TGroupBox;
    Panel2: TPanel;
    Image1: TImage;
    GroupBox5: TGroupBox;
    Label3: TLabel;
    Button2: TButton;
    DBEdit5: TDBEdit;
    Button3: TButton;
    Button4: TButton;
    Button5: TButton;
    DBEdit6: TDBEdit;
    Label4: TLabel;
    Label5: TLabel;
    Panel3: TPanel;
    TabSheet3: TTabSheet;
    procedure FormCreate(Sender: TObject);
    procedure Button5Click(Sender: TObject);
    procedure Button4Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure DBGrid1TitleClick(Column: TColumn);
    procedure Table1AfterScroll(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form10: TForm10;

implementation

uses Main1, Main2, Db_elements;

{$R *.dfm}

```

```

procedure TForm10.FormCreate(Sender: TObject);
begin
  Table1.DatabaseName := ExtractFilePath(Application.ExeName)+'bases\';
  Table1.TableName := 'microchips.db';
  Table1.Active := true;
end;

procedure TForm10.Button5Click(Sender: TObject);
begin
  Close;
  Table1.Edit;
  Table1.Post;
end;

procedure TForm10.Button4Click(Sender: TObject);
begin
  if Table1.RecordCount > 0 then
  begin
    if Application.MessageBox('Ви дійсно хочете видалити поточний елемент',
'Увага', MB_YESNO) = IDYES then
      Table1.Delete;
    end;
  end;

procedure TForm10.Button3Click(Sender: TObject);
begin
  form11.ShowModal;
end;

procedure TForm10.Button1Click(Sender: TObject);
begin
  form12.ShowModal;
end;

procedure TForm10.DBGrid1TitleClick(Column: TColumn);
begin
  if column.Index = 0 then begin Table1.IndexFieldNames := 'Mark';
DBGrid1.Columns[0].Title.Caption := 'Маркировка*';
DBGrid1.Columns[1].Title.Caption := 'Тип'; end;;
  if column.Index = 1 then begin Table1.IndexFieldNames := 'Type';
DBGrid1.Columns[0].Title.Caption := 'Маркировка';
DBGrid1.Columns[1].Title.Caption := 'Тип*'; end;
end;

procedure TForm10.Table1AfterScroll(DataSet: TDataSet);
var i: integer;
begin
  Form7.Table1.First;
  for i := 0 to form7.Table1.RecordCount - 1 do
  begin
    if form7.Table1.fieldbyname('Name').AsString =
table1.FieldName('Elem_name').AsString then break;
    form7.Table1.Next;
  end;
end;
end.

```

Файл Search.pas - вікно пошуку елементів та мікросхем

```

unit Search;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm14 = class(TForm)
    LabeledEdit1: TLabeledEdit;
    Button1: TButton;
    Button2: TButton;
    Label1: TLabel;
    Image1: TImage;
    procedure FormActivate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form14: TForm14;

implementation

uses Db_micro;

{$R *.dfm}

procedure TForm14.FormActivate(Sender: TObject);
begin
  LabeledEdit1.SetFocus;
  LabeledEdit1.SelStart := 0;
  LabeledEdit1.SelLength := 255;
end;

procedure TForm14.Button1Click(Sender: TObject);
var i : integer;
begin
  form10.Table1.first;
  for i := 0 to form10.Table1.RecordCount - 1 do
    begin
      if pos (LabeledEdit1.Text, form10.Table1.fieldbyname('Mark').asString) <>
        0 then begin close; {Form10.DBGrid1.SetFocus;} exit end;
      form10.Table1.Next;
    end;
  form10.Table1.First;
  Application.MessageBox('Елемент не знайдено', 'Увага!', MB_OK);
  Close;
  Exit; end; end.

```

Файл Time_scale.pas - вікно вибору масштабу часу

```

unit Time_scale;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, Mask, ComCtrls;

type
  TForm15 = class(TForm)
    TrackBar1: TTrackBar;
    Label1: TLabel;
    Label2: TLabel;
    Edit1: TEdit;
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure FormActivate(Sender: TObject);
    procedure TrackBar1Change(Sender: TObject);
    procedure Edit1KeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form15: TForm15;

implementation

uses Main;

{$R *.dfm}

procedure TForm15.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Form1.ToolButton9.Down:=false;
end;

procedure TForm15.FormActivate(Sender: TObject);
begin
  Left:=Form1.Left+Form1.Frame21.Left+10;
  Top:=Form1.Top+Form1.Frame21.Top+45;
end;

procedure TForm15.TrackBar1Change(Sender: TObject);
begin
  Edit1.Text:=inttostr(TrackBar1.Position);
end;

procedure TForm15.Edit1KeyPress(Sender: TObject; var Key: Char);
begin
  if Key=#13 then
    begin
      try
        if strtoint(Edit1.Text)>1000 then Edit1.Text:='1000';
        TrackBar1.Position:=strtoint(Edit1.Text);
      except
        Edit1.Text:=inttostr(TrackBar1.Position);
      end;
      exit;
    end;
  case Key of
    '0'..'9':;
    #8:;
  end;
end;

```

```
else  
    Key:=chr(0);  
end;  
end;  
end.
```

Кафедра _ КБПЗ _ 2022рік

Файл About.pas - вікно довідки

```
unit About;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls, ExtCtrls;

type
  TForm19 = class(TForm)
    Image1: TImage;
    Memo1: TMemo;
    Button1: TButton;
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form19: TForm19;

implementation

{$R *.dfm}

procedure TForm19.Button1Click(Sender: TObject);
begin
  Form19.Close;
end;

end.
```