

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

(підпис) **Віталій РОМАНКЕВИЧ**
(ініціали, прізвище)

“ ____ ” червня 2021 р.

**Дипломний проєкт
на здобуття ступеня бакалавра**

по спеціальності

123 «Комп'ютерна інженерія»

на тему: «Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті»

Виконав:

студент IV курсу, групи КВ-72

Климов Іван Русланович _____

Керівник доц. каф. СПіСКС, к. т. н., доцент Орлова М.М. _____

Консультант з нормоконтролю, доц.каф.СПіСКС, к.т.н. Клятченко Я.М. _____

Рецензент професор кафедри ОТ, д.т.н., проф. Сімоненко В.П. _____

Засвідчую, що у цьому дипломному
проєкті немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2021 року

**ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

« 20 » _____ жовтня 2020р.

**ЗАВДАННЯ
на дипломний проєкт студента**

Климова Івана Руслановича

1. Тема проєкту «Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті», керівник проєкту Орлова Марія Миколаївна, доц. каф. СПіСКС, к. т. н., затверджені наказом по університету від 25.05.2021 р. N1331-С.
2. Термін подання студентом проєкту: 07.06.2021.
3. Вихідні дані до проєкту: див. технічне завдання.
4. Зміст пояснювальної записки:
аналіз існуючих рішень та обґрунтування теми дипломного проєкту;
опис енергонезалежної пам'яті;
перевірка стану енергонезалежної пам'яті;
розробка програмно-апаратного комплексу для аналізу параметрів енергонезалежної пам'яті.
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):
 - архітектура програмно-апаратного комплексу;

- алгоритм розподілу помилок;
- алгоритм розподілу рівнів;
- алгоритм перевірки струмів;
- презентація.

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я. М., доц. каф. СПіСКС		

7. Дата видачі завдання 04.11.2020.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Видача завдання на дипломне проєктування	04.11.2020	
2.	Вивчення літератури за тематикою дипломного проєкту	16.11.2020	
3.	Розробка та узгодження технічного завдання	25.11.2020	
4.	Розробка структури програмно-апаратного комплексу	17.01.2021	
5.	Програмна реалізація продукту	15.03.2021	
6.	Тестування продукту	04.04.2021	
7.	Підготовка матеріалів текстової частини дипломного проєкту	24.04.2021	
8.	Підготовка матеріалів графічної частини дипломного проєкту	10.05.2021	
9.	Попередній огляд матеріалів дипломного проєкту	25.05.2021	

Студент _____

Іван КЛИМОВ _____

Керівник проєкту _____

Марія ОРЛОВА _____

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку (51 с., 20 рис., 15 табл., 4 додатки).

Об'єктом розробки є програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті.

Метою даного проєкту є розробка програмно-апаратного комплексу для автоматизації тестування та верифікації основних видів енергонезалежної пам'яті в автомобільній промисловості.

Розроблений програмно-апаратний комплекс дозволяє: отримувати результати аналізу для вимірів енергонезалежної пам'яті з детальним представленням стану пам'яті; конфігурувати правила аналізу, характеристики енергонезалежної пам'яті (вид пам'яті, загальна кількість адрес, розмір слова пам'яті). Для мінімізації витрат на створення, тестування й майбутні модифікації програмного коду було використано мову програмування Python з відкритими бібліотеками `numpy`, `pandas`, `seaborn`, які надають зручну й надійну основу для створення програм аналізу даних.

У процесі розробки було:

- проведено аналіз методів верифікації блоків енергонезалежної пам'яті в інтегральних мікросхемах;
- сформульовані вимоги до програмно-апаратного комплексу для аналізу параметрів енергонезалежної пам'яті;
- створено структуру програмно-апаратного комплексу;
- розроблено алгоритми аналізу вимірів енергонезалежної пам'яті;
- розроблено засоби для читання вхідних даних з вимірами та генерації вихідних результатів аналізу;
- здійснено тестування програмно-апаратного комплексу на вимірах справжніх інтегральних мікросхем.

Програмно-апаратний комплекс був успішно впроваджений у ДП «МЕЛЕКСІС-УКРАЇНА» у відділі верифікації енергонезалежної пам'яті. Це

дозволило суттєво зменшити витрати часу на аналіз вимірів енергонезалежної пам'яті.

Ключові слова:

Енергонезалежна пам'ять, верифікація, FG MOS, EEPROM, Flash, NVRAM, DMR, cell current, Python.

ABSTRACT

The diploma project includes an explanatory note (51 p., 20 fig., 15 tables, 1 equation, 4 additions).

The object of development is application software for non-volatile memory parameters analysis.

The purpose of this project is development of application software for testing and verification automation of most non-volatile memory in automotive industry.

The application software provides: getting analysis results for non-volatile measurements with detailed memory state representation; configuring analysis rules, non-volatile memory properties (memory type, total number of addresses, memory word size). Python programming language with open-source libraries numpy, pandas, seaborn were used to minimize costs for development, testing and future code modifications. These instruments give easy and reliable basis for data analysis software.

The development process included:

- analysis of non-volatile memory block verification methods inside integrated circuits;
- designing the application software requirements for non-volatile memory parameters analysis;
- designing the application software structure;
- development of non-volatile memory measurements analysis;
- development of input measurements data reading methods and output analysis results generation;
- testing of application software using real integrated circuits measurements.

The application software has been successfully integrated at non-volatile memory verification department of Melexis-Ukraine company. It has minimized costs for non-volatile memory measurements analysis.

Keywords:

Non-volatile memory, verification, FGMOS, EEPROM, Flash, NVRAM, DMR, cell current, Python.

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045420.002 ТЗ	Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті. Технічне завдання	4		
	A4	ІАЛЦ.045420.003 ТП	Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті. Відомість технічного проєкту	1		
	A4	ІАЛЦ.045420.004 ПЗ	Програмно-апаратний для аналізу параметрів енергонезалежної пам'яті. Пояснювальна записка	51		
	A4	ІАЛЦ.045420.005 Д1	Архітектура програмно-апаратного комплексу. Схема структурна	1		
	A4	ІАЛЦ.045420.006 Д2	Алгоритм розподілу помилок. Схема алгоритму	1		

					ІАЛЦ.045420.001 ОА				
Змін.	Арк.	№ докум.	Підпис	Дата	Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті Опис альбому				
Розробив	Климов І.Р.								
Перевірив	Орлова М.М.								
Консульт.									
Н. контроль	Клятченко Я.М.								
Зав. каф.	Романкевич В.О.								
					Літ.	Аркуш	Аркушів		
						1	65		
					КПІ ім. Ігоря Сікорського, ФПМ КВ-72				

[illegible]

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	2
5.1 Вимоги до програмного продукту, що розробляється	2
5.2 Вимоги до апаратного забезпечення.....	3
5.3 Вимоги до програмного забезпечення	3
6. ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.045420.002 ТЗ							
Змін.	Арк.	№ докум.	Підпис	Дата	<i>Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті Технічне завдання</i>					Літ.	Аркуш	Аркушів
Розробив	Климов І.Р.											
Перевірів	Орлова М.М.										1	4
										КПІ ім. Ігоря Сікорського, ФПМ КВ-72		
Н. контроль	Клятченко Я.М.											
Затвердив	Романкевич В.О.											

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті».

Галузь застосування: автомобільна напівпровідникова промисловість.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проєкту є створення програмно-апаратного комплексу для автоматизації тестування та верифікації основних видів енергонезалежної пам'яті в автомобільній промисловості.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелами інформації є технічна документація, наукові публікації в періодичних виданнях, міжнародні автомобільні стандарти з тестування інтегральних мікросхем.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- кросплатформність;
- графічний інтерфейс;
- інтерфейс командного рядка;
- здатність швидко аналізувати великий об'єм вхідних даних;

					ІАЛЦ.045420.002 ТЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		2

- можливість гнучкої конфігурації правил виконання зі сторони користувача;
- можливість генерації вихідних результатів аналізу параметрів енергонезалежної пам'яті.

5.2 Вимоги до апаратного забезпечення

- процесор, який підтримується мовою програмування Python 3.8 і вище;
- оперативна пам'ять 4 Гб і вище;
- наявність доступу до мережі Інтернет (для початкового встановлення).

5.3 Вимоги до програмного забезпечення

- операційна система, яка підтримується мовою програмування Python 3.8 і вище.

					ІАЛЦ.045420.002 ТЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Видача завдання на дипломне проєктування	04.11.2020
2.	Вивчення літератури за тематикою дипломного проєкту	16.11.2020
3.	Розробка та узгодження технічного завдання	25.11.2020
4.	Розробка структури програмно-апаратного комплексу	17.01.2021
5.	Програмна реалізація продукту	15.03.2021
6.	Тестування продукту	04.04.2021
7.	Підготовка матеріалів текстової частини дипломного проєкту	24.04.2021
8.	Підготовка матеріалів графічної частини дипломного проєкту	10.05.2021
9.	Попередній огляд матеріалів дипломного проєкту	25.05.2021

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.045420.004 ПЗ	Програмно-апаратний для аналізу параметрів енергонезалежної пам'яті.	51		
			Пояснювальна записка			
	A4	ІАЛЦ.045420.005 Д1	Архітектура програмно-апаратного комплексу.	1		
			Схема структурна			
	A4	ІАЛЦ.045420.006 Д2	Алгоритм розподілу помилок.	1		
			Схема алгоритму			
	A4	ІАЛЦ.045420.007 Д3	Алгоритм розподілу рівнів.	1		
			Схема алгоритму			
	A4	ІАЛЦ.045420.008 Д4	Алгоритм перевірки струмів.	1		
			Схема алгоритму			
		Диск CD-ROM	Текст пояснювальної записки.	1		
			Графічний матеріал			

					ІАЛЦ.045420.003 ТП							
Змін.	Арк.	№ докум.	Підпис	Дата	Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті				Літ.		Аркуш	Аркушів
Розробив		Климов І.Р.									5	65
Перевірив		Орлова М.М.							КПІ			
Консульт.									ім. Ігоря Сікорського,			
Н. контроль		Клятенко Я.М.							ФПМ КВ-72			
Зав. каф.		Романкевич В.О.			Відомість технічного проєкту							

Пояснювальна записка до дипломного проєкту

на тему: «Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті»

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	4
ВСТУП.....	6
1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ	7
1.1 Актуальність розробки	7
1.2 Аналіз існуючих рішень	7
1.3 Постановка задачі	9
2. ОПИС ЕНЕРГОНЕЗАЛЕЖНОЇ ПАМ'ЯТІ	10
2.1 Загальна характеристика NVM	10
2.2 Структура NVM	10
2.3 Операції читання й запису в NVM.....	12
2.4 Різновиди NVM.....	13
3. ПЕРЕВІРКА СТАНУ ЕНЕРГОНЕЗАЛЕЖНОЇ ПАМ'ЯТІ.....	15
3.1 Методи верифікації NVM.....	15
3.1.1 Зміна температури	15
3.1.2 Прискорене старіння.....	15
3.1.3 Тестування збуренням	16
3.2 Методи вимірювання NVM.....	17
3.2.1 Detailed marginal read	17
3.2.2 Cell current	18
3.3 Інструменти для вимірювання NVM	18
3.3.1 Модульна платформа NI PXI	18
3.3.2 Програмне забезпечення для NI PXI	19
3.3.3 Основні модулі платформи NI PXI	19

					ІАЛЦ.045420.004 ПЗ			
Змін.	Арк.	№ докум.	Підпис	Дата	<i>Програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті Пояснювальна записка</i>	Літ.	Аркуш	Аркушів
Розробив	Климов І.Р.						1	51
Перевірів	Орлова М.М.							
Н. контроль	Клятченко Я.М.					КПІ ім. Ігоря Сікорського, ФПМ КВ-72		
Затвердив	Романкевич В.О.							

4. РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ АНАЛІЗУ ПАРАМЕТРІВ ЕНЕРГОНЕЗАЛЕЖНОЇ ПАМ'ЯТІ	21
4.1 Архітектура розробки	21
4.2 Вхідні дані	23
4.2.1 Загальна характеристика файлів вимірів	23
4.2.2 Файли detailed marginal read вимірів.....	25
4.2.3 Файли cell current вимірів.....	26
4.2.4 Файл конфігурації.....	27
4.3 Вихідні дані	32
4.3.1 Файл вихідних умов.....	32
4.3.2 Файл розподілу помилок	33
4.3.3 Файл граничних рівнів	34
4.3.4 Файл розподілу рівнів.....	35
4.3.5 Файл розподілу струмів.....	36
4.3.6 Файл перевірки струмів.....	37
4.3.7 Файли карти рівнів.....	37
4.4 Текстовий інтерфейс розробки.....	39
4.5 Графічний інтерфейс розробки	40
4.6 Обробка отриманих результатів аналізу.....	42
ВИСНОВКИ	49
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	50

ДОДАТКИ

Додаток А. Копії графічних матеріалів

- ІАЛЦ.045420.005 Д1. Архітектура програмно-апаратного комплексу. Схема структурна;
- ІАЛЦ.045420.006 Д2. Алгоритм розподілу помилок. Схема алгоритму;
- ІАЛЦ.045420.007 Д3. Алгоритм розподілу рівнів. Схема алгоритму;
- ІАЛЦ.045420.008 Д4. Алгоритм перевірки струмів. Схема алгоритму.

Додаток Б. Лістинг програм.

Додаток В. Презентація бакалаврського проєкту.

Додаток Г. Довідка про впровадження розробки.

					ІАЛЦ.045420.004 ПЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

IC	– інтегральна мікросхема	набір електронних схем, розташованих на одному невеликому напівпровідниковому кристалі.
ЦАП	– цифро-аналоговий перетворювач	електронна схема для перетворення цифрового сигналу в аналоговий.
DMR	– detailed marginal read	метод вимірювання автомобільної енергонезалежної пам'яті через зміну опорної напруги.
EEPROM	– electrically erasable programmable read only memory	тип енергонезалежної пам'яті з можливістю перезапису, який використовується для зберігання невеликих об'ємів даних.
FGMOS	– floating-gate metal-oxide-semiconductor	польовий транзистор з плавним затвором, який може утримувати заряди після вимкнення джерела живлення.
NI PXI	– NI PCI eXtensions for Instrumentation	модульна платформа для вимірювання й тестування електроніки.
NVM	– non-volatile memory	пам'ять, яка зберігає стан всіх комірок після вимкнення джерела живлення.
NVRAM	– non-volatile random access memory	енергонезалежна пам'ять, яка використовує SRAM для операцій читання та запису і EEPROM для довготривалого зберігання після вимкнення джерела живлення.

SRAM	– static random-access memory	тип енергозалежної пам'яті з довільним доступом, який для зберігання даних використовує тригери.
$U_{оп}$	– опорна напруга	напруга, яка використовується у внутрішньому компараторі інтегральної мікросхеми для визначення логічного рівня комірки енергонезалежної пам'яті.

					ІАЛЦ.045420.004 ПЗ	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Інтегральні мікросхеми (ІС) в автомобільній промисловості повинні задовольняти великій кількості вимог різних міжнародних стандартів. Наприклад, якщо в ІС масового користування гарантований термін служби зазвичай становить 5 років у межах температур від 0°C до 50 °C, то в автомобільних ІС – від 10 до 15 років за температур від -40°C до 150°C [1].

Через те, що автомобільні ІС отримали значне застосування, вони розробляються з ретельним урахуванням всіх стандартів, а після розробки проводиться значна кількість тестів усіх внутрішніх структур ІС за різних екстремальних умов для того, щоб гарантувати високі стандарти якості. Це тестування називається верифікацією ІС, яка перевіряє, чи задовольняють інтегральні мікросхеми заданим вимогам.

Зазвичай, кожна ІС автомобільної промисловості має блок енергонезалежної пам'яті (англ. Non-volatile memory, NVM). Цей вид пам'яті може зберігати інформацію у комірках пам'яті після вимкнення живлення. В ІС автомобільної промисловості NVM зазвичай використовується для збереження параметрів калібрування, оскільки інтегральна мікросхема може бути, наприклад, магнітним сенсором для вимірювання обертання колеса чи контролером подавання палива у двигун. Тому блок NVM також має відповідати багатьом вимогам, щоб його можна було використовувати у автомобільній ІС. Через значне поширення енергонезалежної пам'яті в автомобільних інтегральних мікросхемах існує потреба в автоматизованих програмних засобах аналізу вимірів пам'яті.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

1.1 Актуальність розробки

Енергонезалежна пам'ять складається з багатьох тисяч транзисторів, стан яких необхідно перевірити за різних екстремальних умов. Окрім апаратних і програмних засобів для вимірювання параметрів NVM, виникає потреба в програмно-апаратних комплексах, які можуть аналізувати велику кількість вхідних даних з вимірювальних установок, і на основі цього генерувати достатню кількість інформації, щоб інженер міг у зручному вигляді побачити стан комірок пам'яті.

Для верифікації й тестування NVM використовують різні методи перевірки, які по-різному впливають на заряди комірок пам'яті. Тому кількість однотипних вимірів є значною. Через це програмно-апаратні засоби аналізу стають необхідними для успішної й точної перевірки енергонезалежної пам'яті. За відсутності таких програмно-апаратних комплексів, ручних аналіз NVM може займати великий проміжок часу. У сучасних умовах бізнесу, зменшення часу розробки й тестування дають значну перевагу перед конкурентами. Але найважливіша задача – це забезпечити точність і надійність аналізу, тому що автомобільні інтегральні мікросхеми насамперед впливають на безпеку та життя людей.

1.2 Аналіз існуючих рішень

Майже в кожній новій розробленій автомобільній інтегральній мікросхемі NVM відрізняється від попередніх іншою конфігурацією:

- загальною кількістю адрес;
- розміром адресованого слова;
- внутрішнім розміщенням комірок пам'яті;

– фізичною структурою комірки пам'яті.

Одним із рішень для автоматизації аналізу енергонезалежної пам'яті є замовлення програмно-апаратного комплексу в зовнішньої компанії. Але використання програмно-апаратного засобу іншої компанії може стати значно витратним через постійну еволюцію енергонезалежної пам'яті в автомобільній промисловості, з потребою модифікацій алгоритмів аналізу.

Інше рішення передбачає створення програмно-апаратного комплексу для аналізу параметрів енергонезалежної пам'яті інженерами компанії для внутрішнього використання у компанії. По-перше, цей варіант більш вигідний із фінансової точки зору. По-друге, використання мови програмування високого рівня з великою кількістю бібліотек для аналізу даних дозволить мінімізувати витрати на створення, тестування й модифікації програмного коду. Наприклад, мова програмування Python з відкритими бібліотеками `numpy`, `pandas`, `seaborn` надають зручну й надійну основу для створення програм аналізу даних. Таким чином зникає необхідність у власній реалізації всіх базових алгоритмів, усі зусилля можуть бути прикладені саме для безпосередньої реалізації специфічних алгоритмів аналізу енергонезалежної пам'яті. Бібліотеки аналізу, які були зазначені вище, окрім спрощення розробки, мають здатність до опрацювання великого об'єму вхідних даних, тому що їх програмний код написаний на мові програмування C, оптимізований і випробуваний для подібних задач.

Комплекс для аналізу параметрів енергонезалежної пам'яті створений для впровадження та внутрішнього використання в ДП «МЕЛЕКСІС-Україна» у процесі верифікації й тестування блоків NVM інтегральних мікросхем (додаток Г). Серед продуктів компанії велика кількість різноманітних сенсорів і контролерів, які виконують важливі функції у кожному сучасному автомобілі. Створені інтегральні мікросхеми відповідають важливим міжнародним автомобільним стандартам, наприклад, Automotive Electronics Council, Automotive Safety Integrity Level, ISO 26262, які вказують вимоги до

					ІАЛЦ.045420.004 ПЗ	Арк.
						8
Змін.	Арк.	№ докум.	Підпис	Дата		

якості й надійності інтегральних мікросхем. Тому верифікація інтегральних мікросхем в сукупності з верифікацією блоків енергонезалежної пам'яті є важливими етапами для підтвердження необхідних характеристик продукту.

1.3 Постановка задачі

Існує потреба бізнесу в розробці програмно-апаратного комплексу для автоматизації тестування та верифікації основних видів енергонезалежної пам'яті в автомобільній промисловості.

Для успішного впровадження на підприємстві, до розробки висуваються наступні основні вимоги:

- кросплатформність;
- графічний інтерфейс;
- інтерфейс командного рядка;
- здатність швидко аналізувати великий об'єм вхідних даних;
- можливість гнучкої конфігурації правил виконання зі сторони користувача;
- можливість генерації вихідних результатів аналізу параметрів енергонезалежної пам'яті.

Окрім цього, необхідно розуміти структуру енергонезалежної пам'яті, фізичну будову окремого транзистора пам'яті, схеми підключення масиву комірок, найбільш поширені види енергонезалежної пам'яті. Треба знати основні методи верифікації NVM, і як вони впливають на стан комірок пам'яті. На основі цієї інформації необхідно визначити вимоги до формату вхідних даних вимірів, алгоритми аналізу, необхідний перелік і формат вихідних даних з результатами, які можуть у зручному форматі показати детальну інформацію про стан комірок енергонезалежної пам'яті.

					ІАЛЦ.045420.004 ПЗ	Арк.
						9
Змін.	Арк.	№ докум.	Підпис	Дата		

2. ОПИС ЕНЕРГОНЕЗАЛЕЖНОЇ ПАМ'ЯТІ

2.1 Загальна характеристика NVM

Енергонезалежна пам'ять може зберігати свій стан після вимкнення джерела живлення. Приклади NVM: постійний запам'ятовувальний пристрій, Flash пам'ять, магнітні пристрої пам'яті (жорсткі диски, дискети, магнітні стрічки), оптичні диски та ранні види пам'яті, такі як перфострічки та перфокарти. Енергонезалежна пам'ять зазвичай виконує роль другорядної пам'яті або для тривалого зберігання інформації. Роль оперативної пам'яті в обчислювальних системах сьогодні має енергозалежна пам'ять з довільним доступом, яка втрачає всю збережену інформацію після вимкнення джерела живлення. На жаль, більшість видів NVM мають обмеження, які роблять цю пам'ять непридатною для використання в ролі оперативної пам'яті. Зазвичай ціна енергонезалежної пам'яті є вищою, швидкодія цієї пам'яті гірша, ніж в енергозалежної пам'яті [2].

2.2 Структура NVM

Основним елементом сучасної енергонезалежної пам'яті є транзистор з плавним затвором (англ. Floating-gate metal-oxide-semiconductor) (рисунок 2.1).

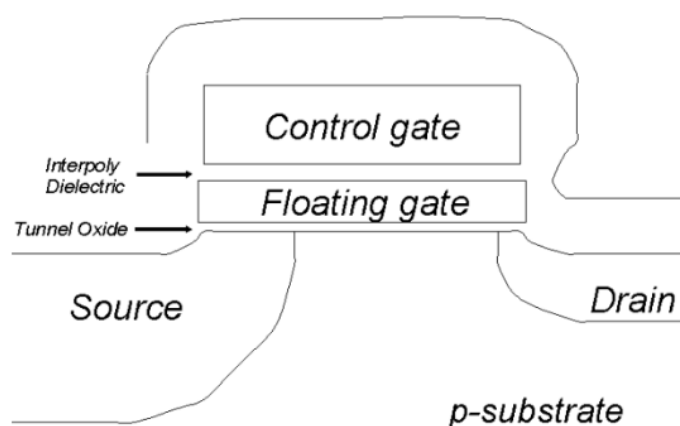


Рисунок 2.1 – FGMOS транзистор [3]

FGMOS транзистор має структуру, яка нагадує польовий транзистор структури метал-оксид-напівпровідник, але має два затвори: управляючий і плавний. Плавний затвор ізолюваний від інших контактів, і за допомогою управляючого затвора, можна пробивати тонкі шари діелектрика, при цьому в плавному затворі електрони стануть ізолювані та його заряд стане негативним. Тому FGMOS транзистор може зберігати заряд протягом багатьох років без необхідності зовнішнього джерела живлення. Наявність заряду на транзисторі говорить про логічний рівень «0», а відсутність заряду – про логічний рівень «1».

Комбінуючи масив FGMOS транзисторів, отримують NVM з потрібним розміром одного слова (адресована одиниця пам'яті), загальною кількістю слів, їх схемою підключення.

У сучасних блоках енергонезалежної пам'яті FGMOS транзистори можуть з'єднуватися у серію, яка нагадує одну з логічних функцій АБО-НЕ (рисунок 2.2), І-НЕ (рисунок 2.3), в залежності до вимог до швидкості й обмежень до площі кристалу.

Змін.	Арк.	№ докум.	Підпис	Дата

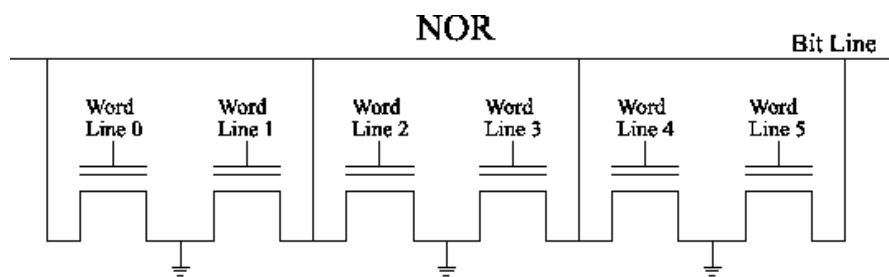


Рисунок 2.2 – АБО-НЕ з'єднання FGMOS транзисторів [4]

АБО-НЕ з'єднання представлено у вигляді двовірної матриці, де для кожного транзистора до стоку приєднуються лінія бітів, до затвору – лінія адрес, виток – до схемної землі. Блоки енергонезалежної пам'яті на основі АБО-НЕ мають швидкий доступу до окремого транзистора, але більшу площу кристалу.

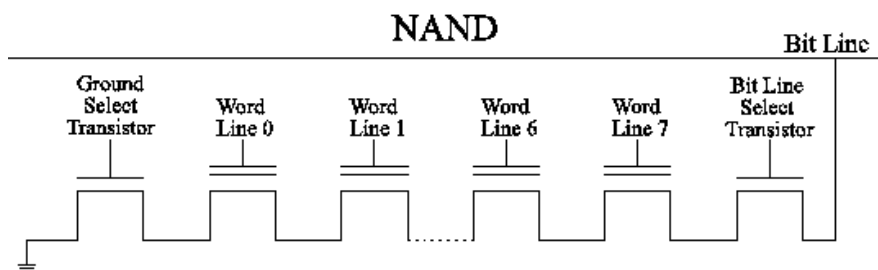


Рисунок 2.3 – І-НЕ з'єднання FGMOS транзисторів [4]

І-НЕ з'єднання будується подібно АБО-НЕ, але з послідовним включенням транзисторів для кожної лінії бітів. У порівнянні з АБО-НЕ, І-НЕ з'єднання, навпаки, має менше площу кристалу, але повільнішу швидкість доступу.

2.3 Операції читання й запису в NVM

Більшість сучасних NVM підтримують операції читання й запису.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045420.004 ПЗ

Арк.

12

Для операції читання на управляючий затвор FGMOS транзистора подається позитивна напруга. Транзистор починає проводити струм за умови відсутності заряду в плавному затворі. При цьому, для АБО-НЕ з'єднання всі інші транзистори в лінії отримують мінімальну напругу для відсутності їх провідності. Для І-НЕ з'єднання, навпаки, на всі інші транзистори в лінії подається підвищена напруга, щоб вони почали проводити струм. Таким чином, відокремивши окремий транзистор серед інших, можна дізнатися його рівень.

Перед операцією запису на всі управляючі затвори транзисторів подається велика позитивна напруга (тунельний ефект), щоб від'ємний заряд покинув плавний затвор, перемикаючись у стан логічної «1», після чого потрібні транзистори отримують великий від'ємний заряд (стан логічного «0»). Останній етап для АБО-НЕ виконується за допомогою інжекції «гарячих» електронів з використанням лінії стоку разом з управляючим затвором, а для І-НЕ використовується тунельний ефект лише з управляючим затвором. Кожен запис пошкоджує оксидний шар FGMOS транзистора, тому NVM мають обмежену кількість записів.

2.4 Різновиди NVM

У сучасних автомобільних ІС найбільше поширення отримали electrically erasable programmable read-only memory (EEPROM), Flash і non-volatile random-access memory (NVRAM).

EEPROM дає змогу зберігати невелику кількість інформації. Для ІС автомобільної промисловості зазвичай достатньо декількох кілобайт для зберігання потрібних даних, тому EEPROM можна часто зустріти в автомобільних ІС. Мінімальний розмір даних для запису – одне слово.

Flash пам'ять була розроблена на основі EEPROM, але з оптимізацією архітектури для зменшення площі масиву пам'яті. Для цього було зроблені модифікації операцій стирання та запису [5].

Сьогодні Flash зазвичай використовується для зберігання великих об'ємів даних за рахунок використання I-HE з'єднання. Один великий блок – це мінімальний розмір даних для запису.

У NVRAM використовується статична пам'ять з довільним доступом (англ. Static random-access memory, SRAM) у поєднанні з EEPROM. На початку з EEPROM завантажуються всі дані в SRAM. Після цього операції читання та запису відбуваються в SRAM. У момент вимкнення живлення IC, весь вміст SRAM записується в EEPROM.

					ІАЛЦ.045420.004 ПЗ	Арк.
						14
Змін.	Арк.	№ докум.	Підпис	Дата		

3. ПЕРЕВІРКА СТАНУ ЕНЕРГОНЕЗАЛЕЖНОЇ ПАМ'ЯТІ

3.1 Методи верифікації NVM

Для детальної перевірки блоків енергоне залежної пам'яті передбачається перелік основних методів верифікації для перевірки стану NVM.

3.1.1 Зміна температури

Даний метод тестування перевіряє, чи правильно функціонують різні параметри енергоне залежної пам'яті при зміні зовнішньої температури. Автомобільні ІС можуть перебувати в різних екстремальних температурних умовах, наприклад, розміщуватися в місцях високої температури. Практично будь-які електричні параметри змінюють своє значення зі зміною температури. Тому перевірка знаходження цих параметрів у межах специфікації є важливим етапом тестування.

3.1.2 Прискорене старіння

Проводиться примусова симуляція старіння енергоне залежної пам'яті, шляхом зберігання інтегральної мікросхеми в термокамері за високої температури.

Залежність між температурою і старінням ІС розраховується за допомогою закону Арреніуса (рівняння $AF_T = \exp\left[\frac{E_\alpha}{k}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right]$, (3.1) [6].

$$AF_T = \exp\left[\frac{E_\alpha}{k}\left(\frac{1}{T_1} - \frac{1}{T_2}\right)\right], \quad (3.1)$$

де E_α – енергія активації, Дж/моль;

k – стала Больцмана, Дж·К⁻¹;

T_1, T_2 – температури, К.

Наприклад, якщо зберігати ІС у термокамері за температури 250°C протягом 24 годин, то вимірювання параметрів може показати, яку якість записаних комірок буде мати NVM приблизно через десять років. Тест вважається пройденим, якщо всі рівні комірок пам'яті не перетинають визначену межу, за якої не можна точно визначити логічний рівень.

3.1.3 Тестування збуренням

Для одного слова енергонезалежної пам'яті здійснюється циклічний перезапис до 100000 разів. Перевіряється наявність чи відсутність впливу перезапису на сусідні комірки у матриці пам'яті на лініях перетину слова. На рисунку 0.4 зображений приклад карти рівнів комірок енергонезалежної пам'яті після тестування збуренням.

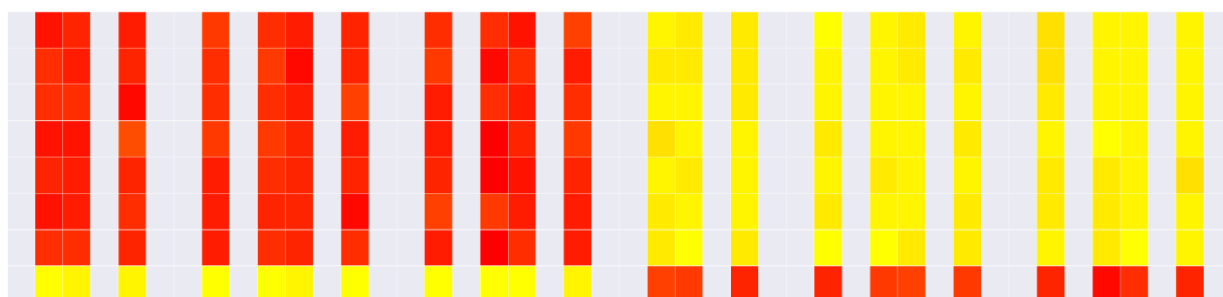


Рисунок 3.1 – Приклад карти рівнів комірок NVM після тестування збуренням

Ліва половина карти рівнів містить парні адреси, права – непарні. Збурення передостанньої адреси викликає впливає на сусідні адреси по горизонталі й вертикалі. На прикладі карти рівнів, де червоний колір позначає гірший рівень, це помітно. Наявність такого впливу пояснюється структурою конкретної NVM: з кожним перезаписом одного слова, заряди FG MOS на лініях перетину поступово витікають, і є ризик втрати логічного рівня комірки.

Змін.	Арк.	№ докум.	Підпис	Дата

3.2 Методи вимірювання NVM

Для перевірки впливу методів верифікації необхідно проводити вимірювання стану транзисторів у блоках енергонезалежної пам'яті.

3.2.1 Detailed marginal read

Для достовірної оцінки стану всіх комірок пам'яті, одним з найбільш інформативних тестів є detailed marginal read (DMR) тест. Для його виконання немає необхідності у фізичній деструкції приладу або його корпусу, а потрібні додаткові структури для його проведення, в більшості випадків реалізовані на етапі виготовлення мікросхеми [7].

Для визначення логічного рівня окремого FGMOS необхідний компаратор на базі операційного підсилювача, який порівнює заряд конденсатора, який заряджений під'єднаним FGMOS, з відомою програмованою опорною напругою ($U_{оп}$).

Алгоритм виконання DMR.

1. $U_{оп}$ встановлюється у значення за замовчуванням.
2. Виконується читання логічних рівнів всіх комірок пам'яті.
3. $U_{оп}$ встановлюється в мінімальне значення.
4. Читання всіх комірок.
5. Значення $U_{оп}$ збільшується із заданим кроком. Кроки 3, 4 повторюються до встановлення максимальної $U_{оп}$.
6. Зберігання прочитаних даних у файл потрібного формату для подальшого програмного аналізу.

Подальший аналіз базується на порівнянні вчитаних логічних рівнів всіх бітів кожної адреси з логічними рівнями при значення $U_{оп}$ за замовчуванням. Зі поступовою зміною $U_{оп}$ компаратор буде визначати більшу кількість неправильних логічних рівнів. На основі цієї інформації

порівнюється стан комірок NVM до і після впливу тестів, які впливають на заряди FGMOS.

3.2.2 Cell current

Цей метод вимірювання визначає значення струму кожної комірки енергонезалежної пам'яті. Аналогічно до DMR цей метод дає змогу спостерігати погіршення стану NVM після різних тестів.

Для визначення значення струму окремого транзистора, в інтегральній мікросхемі повинна бути реалізована можливість для підключення лінії слів до виходу IC для подальшого визначення струму, який проходить через окремий FGMOS, зовнішнім вимірювальним обладнанням.

Усі значення струмів зберігаються у файл потрібного формату для аналізу. Відмінність між різними IC у результатах DMR буде мати таку ж відмінність і в cell current.

3.3 Інструменти для вимірювання NVM

Для того, щоб провести вимірювання енергонезалежної пам'яті, необхідні професійні інструменти для комунікації з IC і вимірювання параметрів NVM.

3.3.1 Модульна платформа NI PXI

NI PCI eXtensions for Instrumentation (NI PXI) – модульна платформа для вимірювання й тестування електроніки. NI PXI має високу швидкодію, надійність і точність. Модульна платформа надає можливість додавати велику кількість різних синхронізованих вимірювальних модулів: цифрові мультиметри, генератори сигналів тощо.

3.3.2 Програмне забезпечення для NI PXI

Крім апаратних модулів, платформа NI PXI має програмне забезпечення, яке повністю підтримує всі можливості вимірювальних пристроїв. Також надається програмна платформа NI LabVIEW для створення будь-яких тестів з використанням вимірювальних модулів. Особливістю програмної платформи є те, що розробка тестів відбувається за допомогою графічних блоків, функцій, операцій тощо. Взаємодія інженера з такими програмами відбувається за допомогою стандартних віртуальних графічних панелей із різними перемикачами, індикаторами, графіками, які нагадують взаємодію зі справжніми вимірювальними прикладами.

3.3.3 Основні модулі платформи NI PXI

Модуль NI PXIe-6547 дозволяє генерувати та читати сигнали різної форми (рисунок 3.2). Має велику кількість вбудованих швидкісних каналів (підтримка частоти сигналів до 100 МГц). Використовується для комунікації з ІС. Модулі необхідні для вимірювання DMR і cell current.



Рисунок 3.2 – Модуль NI PXIe-6547 [8]

NI PXI-4071 – це потужний цифровий мультиметр, який забезпечує дуже точні та швидкі вимірювання змінних, постійних напруг і струмів (рисунок 3.3). Для вимірювання енергонезалежної пам'яті, модуль NI PXI-4071 зазвичай використовується для cell current.



Рисунок 3.3 – Модуль NI PXI-4071 [9]

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045420.004 ПЗ

Арк.

20

4. РОЗРОБКА ПРОГРАМНО-АПАРАТНОГО КОМПЛЕКСУ ДЛЯ АНАЛІЗУ ПАРАМЕТРІВ ЕНЕРГОНЕЗАЛЕЖНОЇ ПАМ'ЯТІ

4.1 Архітектура розробки

Архітектура розробки вказана в додатку А, креслення Д1. Програмно-апаратний комплекс приймає на вхід дані, розмір яких може вимірюватися сотнями мегабайт. Тому необхідно правильно обрати інструменти для програмування. Ці інструменти повинні мати готовий функціонал для ефективної роботи з аналізом великих даних. Тому оптимальним вибором є мова програмування Python.

Обрано перелік відкритих бібліотек для аналізу даних:

- numpy (робота з великими багатовимірними масивами чисел);
- pandas (представлення даних у вигляді таблиць);
- seaborn (для візуалізації даних).

З обраним переліком бібліотек значно зменшується час на розробку, інструменти мають значне поширення, високу стабільність і швидкість роботи.

Для розробки програмно-апаратного комплексу, одне із застосувань бібліотеки numpy – це операції з багатовимірними масивами бітів. Зникає необхідність у написанні циклів для реалізації операцій з векторами і матрицями. У випадку мови Python, відсутність великої кількості циклів значно зменшує час роботи, тому що мова є інтерпретованою. Програми, написані інтерпретованою мовою, на відміну від компільованих, під час кожного виконання програми, здійснюють лексичний і синтаксичний розбір кожного рядка. Тому, для масивів великого розміру, швидкість виконання циклічних блоків коду є дуже низькою. Більшість методів та функцій бібліотеки numpy є дуже швидкими й оптимізованими для процесора, тому що написані мовою програмування C. Для представлення чисел Python не

використовує обмежені типи даних, наприклад, `int`, `long`, `double`. Числа в цій мові можуть мати довільну точність. Тому, використовуючи числа мови Python у великих масивах, необхідно більше пам'яті. Бібліотека `numpy` використовує фіксовані типи даних для чисел, тому великі масиви даних займають менше пам'яті.

Бібліотека `pandas` необхідна для роботи з табличними даними. Через те, що програмно-апаратний комплекс працює з великою кількістю таких даних, використання бібліотеки `pandas` є оптимальним рішенням. Надаються інструменти для читання і запису в файли різних табличних форматів, операції для роботи з рядками і колонками. Допускається виконання різних операцій над числами в таблицях. Для цього бібліотека `pandas` використовує швидкі операції `numpy`.

Бібліотека `seaborn` побудована на основі бібліотеки `matplotlib`. Вона надає інструменти та шаблони для зручної візуалізації даних у вигляді графіків, діаграм тощо. Бібліотека підтримує типи даних від `numpy` і `pandas`, тому її зручно інтегрувати в розробку без необхідності в конвертації даних.

Для максимальної швидкодії при аналізі великого об'єму даних реалізовано паралельне опрацювання даних. Для цього використовується стандартна бібліотека `multiprocessing`. Вона дозволяє ефективно розділити різні задачі на окремі паралельні процеси. У комп'ютерах з кількома ядрами центрального процесора це дає можливість у кілька разів збільшити швидкість аналізу великого об'єму даних, тому що паралельні операції виконуються не лише програмному, але й на апаратному рівні. Кожен паралельний процес генерує вихідні дані, які необхідно записати в файли, зазвичай для кожного виду аналізу в один файл для всіх процесів. Тому для цього запис відбувається після виконання всіх процесів. Бібліотека `multiprocessing` надає інструменти для зручної й надійної синхронізації доступу до спільних ресурсів. Наприклад, клас `Manager` дозволяє зберігати і читати спільні ресурси багатьом

паралельним процесам без необхідності складної ручної синхронізації процесів за допомогою семафорів, м'ютексів тощо.

Виконання аналізу може займати деякий час, тому є необхідність у виводі поточного стану виконання. У випадку помилок аналізу, користувач повинен побачити інформацію про програмний модуль і дані, де з'явилася помилка. Також за можливості надається пояснення і рекомендації того, як виправити помилку.

4.2 Вхідні дані

Для функціонування програмно-апаратного комплексу передбачаються вимоги до формату та вмісту вхідних даних.

4.2.1 Загальна характеристика файлів вимірів

Вхідні файли вимірів повинні мати розширення CSV (comma-separated values) і містити таблицю у тестовому форматі з заголовками, умовами і вимірами DMR чи cell current (таблиця 4.1). Кожне окреме значення в таблиці розділяється символом U+002C (,).

Таблиця 4.1 – Формат вхідного текстового файлу вимірів

Заголовки	
Умови	Виміри

Кожен заголовок повинен мати формат <префікс>_<ім'я заголовку>[<одиниці вимірювання/розрядність>]. Префікс може мати одне із

двох значень: С (умови), М (виміри). Одиниці вимірювання показують аналогові виміри пам'яті, наприклад: V (вольт), А (ампер), degC (температура в °C), s (секунди). Для одиниць вимірювання допускається використання префіксів системи SI, наприклад: n (нано), u (мікро), m (мілі), k (кіло), Розрядність використовується для цифрових вимірів пам'яті: DEC, HEX, BIN, DBL. Останній сегмент заголовку [<одиниці вимірювання/розрядність>] може бути відсутнім, у такому випадку значення такої колонки будуть мати текстовий тип даних.

Приклади заголовків вхідного текстового файлу вимірів:

- C_Temperature[degC];
- M_VMG[V];
- M_I_ref[uA];
- M_Addr_12[HEX];
- M_Bit_15[uA];
- C_Date.

Текстова область з умовами повинна містити рядки з інформацією про умови під час процесу вимірювання пам'яті, наприклад: дата, час, температура, напруга живлення, значення калібрування тощо (таблиця 4.2).

Таблиця 4.2 – Приклад текстової області з умовами

C_Date	C_Time	C_VS[V]	C_Temperature [degC]	C_VBG_trim [DEC]	M_VBG[V]
21.01.21	11:01:01	12	150	1	1.21234
21.01.21	11:01:02	12	150	2	1.24567
21.01.21	11:01:03	12	150	3	1.24321
21.01.21	11:01:04	12	-40	1	1.24321
22.01.21	11:01:05	12	-40	2	1.23215

4.2.2 Файли detailed marginal read вимірів

Файли DMR мають в області вимірів один чи два очікуваних контенту пам'яті та фіксовану кількість контентів з різним значенням $U_{оп}$. Очікуваний контент пам'яті може бути або прочитаний із пам'яті з $U_{оп}$ за замовчуванням (application content), або завчасно відомий (expected content). У таблиці 0.7 показаний скорочений приклад файлу з DMR вимірами. Присутні умови, перша і остання колонки з адресами NVM.

Таблиця 4.3 – Скорочений приклад файлу з DMR вимірами

C_Date	C_Window [DEC]	C_Content	C_Index [DEC]	M_Addr_0 [HEX]	...	M_Addr_1023 [HEX]
21.01.21	0	Expected	0	'FFFFFFFF	...	'00000000
21.01.21	0	Application	0	'FFFFFFFF	...	'00000000
21.01.21	0	Margin	0	'FFFFFFFF	...	'00000000
21.01.21	1	Margin	0	'FFFFFFFF	...	'00000000
21.01.21	2	Margin	0	'FAFFBF1F	...	'00000000
21.01.21	3	Margin	0	'FAFFBF10	...	'00000000
21.01.21	4	Margin	0	'FAFFBF10	...	'00000000
22.02.21	0	Expected	1	'00000000	...	'FFFFFFFF
22.02.21	0	Application	1	'00000000	...	'FFFFFFFF
22.02.21	0	Margin	1	'00000000	...	'FFFFFFFF
22.02.21	1	Margin	1	'00000000	...	'FFFFFFFF
22.02.21	2	Margin	1	'00000000	...	'FAFFBF1F
22.02.21	3	Margin	1	'00000000	...	'FAFFBF10
22.02.21	4	Margin	1	'00000000	...	'FAFFBF10

Один файл може містити довільну кількість DMR даних, але необхідно, щоб кожен мав унікальну умову та хоча б один очікуваний контент. У таблиці 0.7 значення 0 у колонці «C_Index[DEC]» позначає перший DMR, 1 – другий.

Для області умов обов'язковим є наявність колонки з назвою «C_Content», яка позначає очікуваний контент («Application», «Expected») та фіксовану кількість контентів з іншим значенням $U_{оп}$. Необхідно, щоб окрім «C_Content» була хоча б одна колонка, яка позначає умову, за яких було здійснений DMR. «Expected» контент має вищий пріоритет, ніж «Application». Також необхідно, щоб була колонка, яка позначає рівень $U_{оп}$ або значення регістру внутрішнього ЦАП (цифро-аналоговий перетворювач), який управляє зміною $U_{оп}$. Назва заголовку цієї колонки задається користувачем.

Кожна колонка з області вимірів позначає окрему адресу енергонезалежної пам'яті. Всі адресовані значення повинні мати шістнадцятковий формат запису з символом U+0027 (') на початку.

4.2.3 Файли cell current вимірів

У файлах cell current міститься інформація про рівень струму, який протікає через кожен FGMOS. У таблиці 0.8 показаний скорочений приклад файлу з cell current вимірами. Присутні частина умов, перша і остання колонки зі значеннями струмів бітів.

Таблиця 4.4 – Скорочений приклад файлу з cell current вимірами

C_Date	...	C_Content	C_Address [HEX]	C_Data [HEX]	M_Bit_0 [uA]	...	M_Bit_21 [uA]
21.01.21	...	MA	'0	'3FFFFFF	1.234	...	1.234
21.01.21	...	MA	'1	'000FFF	0	...	29.234
21.01.21	...	MA	'1A	'3FFFFFF	0	...	0
21.01.21	...	MA	'3FF	'000000	29.234	...	28.234

«C_Address[HEX]» позначає адресу енергонезалежної пам'яті в шістнадцятковому форматі. Колонки області вимірів позначають біт одного слова пам'яті. «C_Content» повинен завжди мати значення «МА» (main array). Також необхідна наявність колонки, назва якої задається користувачем, яка для кожної адреси визначає очікуване слово пам'яті.

4.2.4 Файл конфігурації

Вхідний файл конфігурації – це текстовий файл із розширенням CFG, який дозволяє користувачу налаштовувати правила виконання аналізу параметрів енергонезалежної пам'яті.

Базовим елементом кожного рядка є ключ і значення у форматі <ключ>=<значення>. Текстові значення задаються без лапок. Сукупність ключів і значень може групуватися в розділи, які вказуються на початку групи і мають формат [<розділ>]. У файлі конфігурації для коментаря використовується символ U+003B (;), який вказується на початку коментаря.

Вхідний файл конфігурації має визначений перелік обов'язкових і необов'язкових розділів і ключів.

Розділ «General» містить конфігурацію вхідних файлів вимірів. У таблиці 0.9 вказаний перелік ключів розділу.

Таблиця 4.5 – Перелік ключів розділу «General»

Ключ	Тип	Обов'язковий?	Значення за замовчуванням
Memory_type	Рядок	Так	–
Dump_word_size	Ціле число	Так	–
Cell_current_mode	Булевий	Так	False
Explicit_c_content	Булевий	Ні	–
Expected_pattern	Рядок	Ні	–

Ключ «Memory_type» визначає вид пам'яті, з якої отримані виміри.

Перелік видів енергонезалежної пам'яті, що підтримуються програмно-апаратним комплексом:

- CEEPROM_XP018;
- CEEPROM_XT018;
- OTP_IFUSE_XH018;
- XFE_X72TA_V1_XT018;
- XFE_X72TA_V2_XT018;
- XFL_8KX16P_H01_XH018;
- XFL_8KX32P_H12_XH018;
- XNVR_V4_XH018;
- XNVR_V5_XH018;
- XOTPTRIM_V20_XH018.

Ключ «Dump_word_size» задає розмір одного слова енергонезалежної пам'яті в бітах. Якщо значення ключа «Cell_current_mode» True, то вхідні файли вимірів аналізуються, як cell current виміри. «Explicit_c_content» може бути Expected або Application. Цей ключ вказує, який очікуваний контент обрати для аналізу, якщо цих контентів декілька. Якщо вказується ключ «Expected_pattern», тоді ігноруються всі очікувані контенти з вхідних файлів вимірів і використовується один з двох: всі біти нулі (zeros), всі біти одиниці (ones).

Розділ «Current_analysis» визначає налаштування параметрів для cell current аналізу (таблиця 4.6).

Таблиця 4.6 – Перелік ключів розділу «Current_analysis»

Ключ	Тип	Обов'язковий?	Значення за замовчуванням
Current_step	Дійсне число	Так, якщо cell current режим	1
DAC_step	Дійсне число	Ні	1

Ключ «Current_step» вказує розмір кроку для розрахунку розподілу струмів. «DAC_step» визначає відповідність між вхідними та вихідними струмами. Якщо це значення не «1», то перед аналізом всі вхідні струми діляться на це значення.

Ключі розділу «Header_conf» визначають основні назви заголовків у файлах вимірів, які необхідні для аналізу. У таблиці 1.1 вказаний перелік ключів цього розділу.

Таблиця 4.7 – Перелік ключів розділу «Header_conf»

Ключ	Тип	Обов'язковий?	Значення за замовчуванням
First_data	Рядок	Так	–
Reference	Рядок	Так	–

Ключ «First_data» задає у вхідних файлах вимірів назву заголовка з першою колонкою області вимірів. Ключ «Reference» в режимі DMR вказує назву колонки з позначенням $U_{оп}$ або значенням регістру внутрішнього ЦАП. У режимі cell current ключ вказує назву колонки з очікуваним контентом.

Розділ «Conditions» визначає опис умов, які необхідно обрати із вхідних файлів вимірів. Розділ може містити довільну кількість ключів з довільною назвою. Необхідний формат: {<заголовок умови>: [<значення умови 1>, ...], ...}, де вказується хоча б одна умова. Якщо необхідно включити всі можливі

значення умови, то вказується ["]. Якщо, наприклад, у розділі «Conditions» вказується ключ: Cond1 = {'Tsense[DEC]': ["], 'VMG[DEC]': [16, 24], 'VDDD18[V]': [1.6, 1.7, 1.8]}, то він визначає умови, які наведені в таблиці 1.2.

Таблиця 4.8 – Набір умов, які визначаються прикладом вище

Tsense[DEC]	VMG[DEC]	VDDD18[V]
4	16	1.6
4	16	1.7
4	16	1.8
4	24	1.6
4	24	1.7
4	24	1.8
7	16	1.6
7	16	1.7
7	16	1.8
7	24	1.6
7	24	1.7
7	24	1.8

Розділ «Skip_conditions» дозволяє вказувати умови, які необхідно виключити з аналізу. Формат ключів однаковий, як у розділі «Conditions». Розділ «Skip_conditions» може не мати ключів, якщо немає потреби виключати умови.

Розділ «Output_files» містить перелік булевих ключів, які вказують параметри пам'яті, які необхідно аналізувати зі створенням вихідних файлів із результатами. Відсутність ключа відповідає тому, що цей параметр буде пропущений.

Перелік дозволених ключів у розділі «Output_files»:

- Fail_Distribution;
- Level_Distribution;
- Margin_Levels;
- Bit_Map;
- Bitmap_as_text;
- Output_condition;
- Current_distribution;
- Current_content_check.

Приклад вхідного файлу конфігурації наведено на рисунку 0.5.

```
[General]
Memory_type = CEEPROM_XT018
Dump_word_size = 22
Cell_current_mode = False

[Current_analysis]
Current_step = 0.01
DAC_step = 0.1

[Header_conf]
First_data = Addr_4096[DEC]
Reference = C_Window[V]

[Conditions]
Cond1 = {'C_Temp[degC]': [''], 'C_ChipID[DEC]': [1, 3]}

[Skip_conditions]
Cond1 = {'C_Temp[degC]': [-40]}

[Output_files]
Fail_Distribution = True
Level_Distribution = False
Margin_Levels = True
Bit_Map = True
Bitmap_as_text = True
Output_condition = True
Current_distribution = True
Current_content_check = True
```

Рисунок 4.1 – Приклад вхідного файлу конфігурації

4.3 Вихідні дані

Після опрацювання вхідних даних, програмно-апаратний комплекс генерує необхідний перелік вихідних даних узгодженого формату та вмісту.

4.3.1 Файл вихідних умов

Після налаштування необхідних умов у файлі конфігурації, відбувається читання вхідних файлів вимірів. Умови, які були знайдені в цих файлах, записуються в окремий файл вихідних умов. Це дає змогу інженеру побачити всі знайдені умови для кожного DMR чи cell current у вхідних файлах вимірів, які можуть мати десятки та сотні різних умов. Приклад файлу вихідних умов показаний у таблиці 1.3.

Таблиця 4.9 – Приклад файлу вихідних умов

C_DUT	C_Bake_time [h]	C_Temp [degC]	C_Disturb_temp [degC]	C_Cycle	Row_count
73	1	-40	150	1000	172
72	1	-40	150	1000	172
71	1	-40	150	1000	172
70	1	-40	150	1000	172
73	1	35	150	1000	172
72	1	35	150	1000	172
71	1	35	150	1000	172
70	1	35	150	1000	172
73	1	175	150	1000	172
72	1	175	150	1000	172
71	1	175	150	1000	172
70	1	175	150	1000	172

4.3.2 Файл розподілу помилок

Розподіл помилок рахується лише в DMR режимі аналізу. У файлі міститься інформація про загальну кількість помилок виміру, які трапилися на кожній змінній $U_{оп}$. Тобто алгоритм порівнює кожен вимір із очікуваним і підраховує кількість бітів, які не збігаються з очікуваними. У файлі вказується вся область умов та одна колонка «Fail_count», яка містить кількість помилок. Алгоритм розподілу помилок вказаний у додатку А, креслення Д2. Приклад файлу розподілу помилок показано у таблиці 0.14.

Таблиця 4.10 – Приклад файлу розподілу помилок

SampleID	C_BakeTime [Hrs]	C_RdTemp [degC]	C_Content	C_DAC [DEC]	Fail_count
ChipTest	0	175	Margin	-60	147456
ChipTest	0	175	Margin	-50	147456
ChipTest	0	175	Margin	-40	147456
ChipTest	0	175	Margin	-30	422
ChipTest	0	175	Margin	-20	0
ChipTest	0	175	Margin	-10	0
ChipTest	0	175	Margin	0	0
ChipTest	0	175	Margin	0	0
ChipTest	0	175	Margin	10	0
ChipTest	0	175	Margin	20	1482
ChipTest	0	175	Margin	30	90134
ChipTest	0	175	Margin	40	147174
ChipTest	0	175	Margin	50	147456
ChipTest	0	175	Margin	60	147456

4.3.3 Файл граничних рівнів

Для DMR режиму, файл граничних рівнів визначає два значення $U_{оп}$ або регістру ЦАП.

1. Найгірший рівень логічного рівня «0» (level low).
2. Найгірший рівень логічного рівня «1» (level high).

Ці рівні отримуються з результатів розподілу помилок. Шукається область, де мінімальна помилок бітів (зазвичай «0»). Фіксуються два значення вище і нижче, які позначають найгірші логічні рівні. Приклад файлу граничних рівнів для DMR показано у таблиці 0.15.

Таблиця 4.11 – Приклад файлу граничних рівнів для DMR

C_Temp [degC]	C_DUT	C_Content	M_Window [V]	Level_low	Level_high
-40	73	Margin	-4.15	-1.41	0.91
-40	72	Margin	-4.12	-1.36	0.98
-40	71	Margin	-4.12	-1.23	0.87
-40	70	Margin	-4.11	-1.52	0.91
35	73	Margin	-4.08	-1.34	0.91
35	72	Margin	-4.05	-1.29	0.98
35	71	Margin	-4.05	-1.16	0.87
35	70	Margin	-4.04	-1.44	0.91
175	73	Margin	-3.9	-1.25	0.91
175	72	Margin	-3.88	-1.19	0.98
175	71	Margin	-3.88	-1.08	0.87
175	70	Margin	-3.87	-1.36	0.91

У випадку cell current вимірів, файл граничних рівнів покаже максимальний струм логічного рівня «1» і мінімальний струм логічного рівня «0» (таблиця 4.12).

Таблиця 4.12 – Приклад файлу граничних рівнів для cell current

SampleID	C_Content	C_DATA [HEX]	C_ADDR [HEX]	Max_curr_of_one [uA]	Min_curr_of_zero [uA]
1	MA	'2AAAAA	'00400	0.91	6.48
2	MA	'2AAAAA	'00400	0.78	6.23
3	MA	'2AAAAA	'00400	0.56	6.51
4	MA	'2AAAAA	'00400	0.73	6.48

4.3.4 Файл розподілу рівнів

Розподіл рівнів рахується лише в DMR режимі. Алгоритм обирає напрямок пошуку рівнів у залежності від значення кожного очікуваного біта. Якщо біт дорівнює «0», то пошук відбувається, починаючи з найменшого рівня. І навпаки, для біта «1» пошук починається з найбільшого рівня. Під час ітерації, порівнюється кожен біт з очікуваним. Якщо знайдена перша помилка, то у файл розподілу рівнів запишеться останній рівень, який був до помилки. Більш детальний опис алгоритму показаний у додатку А, креслення ДЗ. Приклад файлу розподілу рівнів вказано у таблиці 0.17.

Таблиця 4.13 – Приклад файлу розподілу рівнів

C_ChipId	C_Vmg[DEC]	C_Content	Distribution_level
347189	-30	Margin	197
347189	-20	Margin	7
347189	-10	Margin	0
347189	0	Margin	0
347189	0	Margin	0
347189	10	Margin	0
347189	20	Margin	14
347189	30	Margin	127

4.3.5 Файл розподілу струмів

Розподіл струмів представлений у текстовому форматі значень гістограми. Використовуючи параметр файлу конфігурації «Current_step», записуються у файл всі значення струмів від мінімального до максимального із кроком «Current_step».

Вихідний файл, окрім області умов, має дві колонки.

1. Колонка зі струмами (значеннями інтервалів гістограми).
2. Колонка з підрахунком загальної кількості напруг в кожному інтервалі гістограми.

Таблиця 4.14 – Приклад файлу розподілу струмів

C_ChipId	C_Content	C_ADDR[HEX]	C_DATA[HEX]	Current[uA]	Count
2031622	MA	'5800	'3FFFFFF	1	2095
2031622	MA	'5800	'3FFFFFF	6	1843
2031622	MA	'5800	'3FFFFFF	11	0
2031622	MA	'5800	'3FFFFFF	16	0
2031622	MA	'5800	'3FFFFFF	21	0
2031622	MA	'5800	'3FFFFFF	26	0
2031622	MA	'5800	'3FFFFFF	31	0
2031622	MA	'5800	'3FFFFFF	36	18434
2031622	MA	'5800	'3FFFFFF	41	18431
2031622	MA	'5800	'3FFFFFF	46	18279
2031617	MA	'5800	'3FFFFFF	36	19686
2031617	MA	'5800	'3FFFFFF	41	19725
2031617	MA	'5800	'3FFFFFF	46	19739

4.3.6 Файл перевірки струмів

Перевірка струмів розраховує середнє значення струмів логічних рівнів «0» та «1» і порівнює всі струми у відповідності до логічних рівнів. Якщо струм логічної одиниці перевищує середній, або струм логічного нуля менше середнього, то виводиться користувачу повідомлення про повідомлення некоректне значення струму. Повний алгоритм перевірки струмів зображений у додатку А, креслення Д4. Кожна помилка фіксується у файл перевірки струмів із зазначенням розміщення некоректного значення струму та очікуваного логічного рівня (таблиця 4.15).

Таблиця 4.15 – Приклад файлу перевірки струмів

C_ChipId	C_Content	C_DATA [BIN]	Bit_current [uA]	C_ADDRESS	Bit_number
2031617	MA	1	43.7	'5800	1
2031617	MA	1	45	'5800	4
2031617	MA	1	46.1	'5800	47
2031617	MA	1	45.8	'5800	48
2031617	MA	1	49.2	'5800	50
2031617	MA	0	36.4	'5808	2
2031617	MA	0	36	'5808	7
2031617	MA	0	40.3	'5808	9
2031617	MA	0	35.6	'5808	12

4.3.7 Файли карти рівнів

Карта рівнів – це графічний спосіб представлення усіх рівнів кожного FGMOS у вигляді двовимірних масивів комірок для двох логічних рівнів окремо. Карта рівнів також показує фізичне розміщення всіх бітів у залежності від виду NVM (рисунок 4.2).

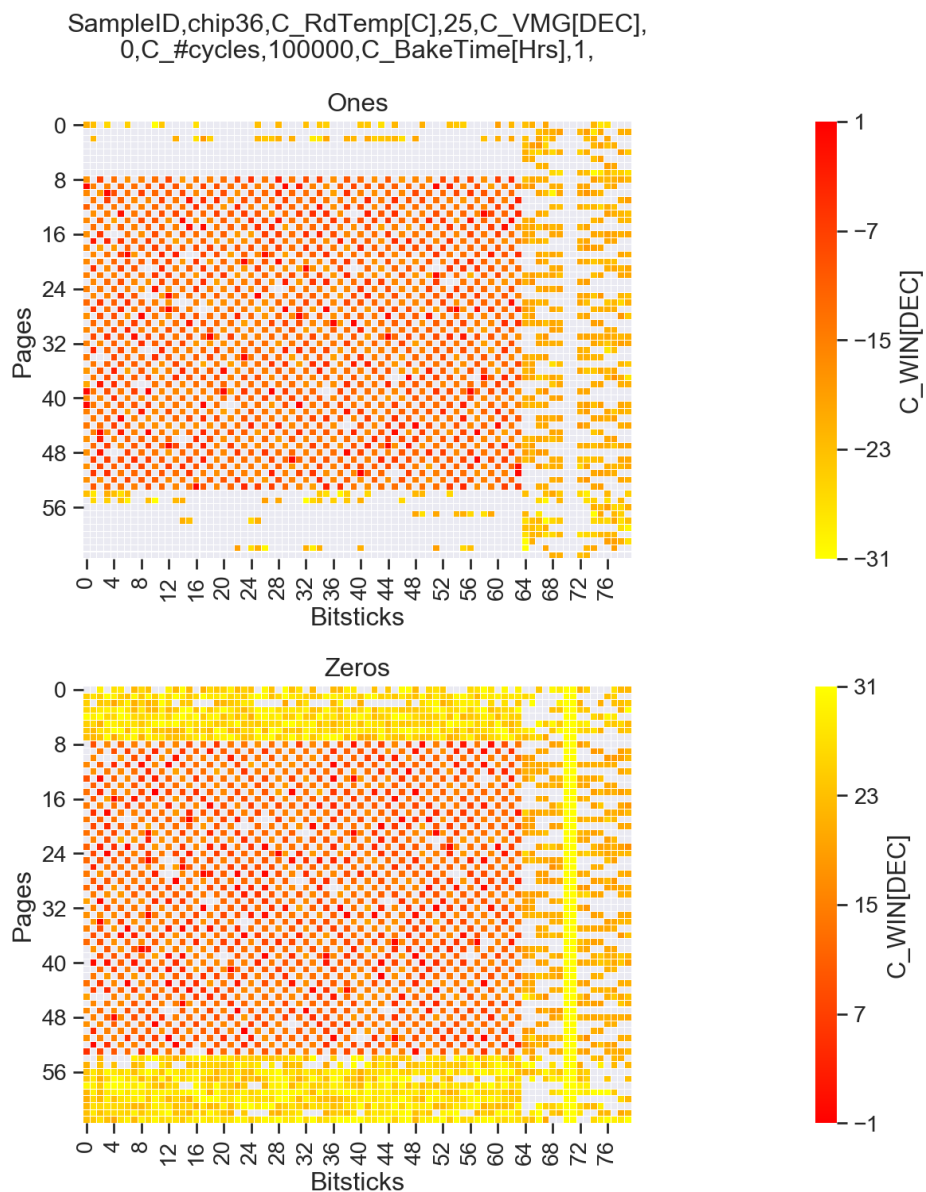


Рисунок 4.2 – Приклад карти рівнів енергонезалежної пам'яті

Двовимірні масиви карти рівнів мають індексацію бітів і сторінок. Сторінки можуть бути для одного виду пам'яті адресами, наприклад EEPROM. У Flash одна сторінка буде дорівнювати визначеній кількості адрес. Ці особливості кожного з видів пам'яті автоматично враховуються при побудові карти рівнів. Кожна комірka, в залежності від логічного рівня, знаходиться в одному із двох масивів. Комірka має рівень, який показано кольором із палітри від жовтого до червоного. Справа від масивів розміщуються вісі з числовим визначенням кольорів. Жовтий рівень показує найкращі комірки, червоний –

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045420.004 ПЗ

Арк.

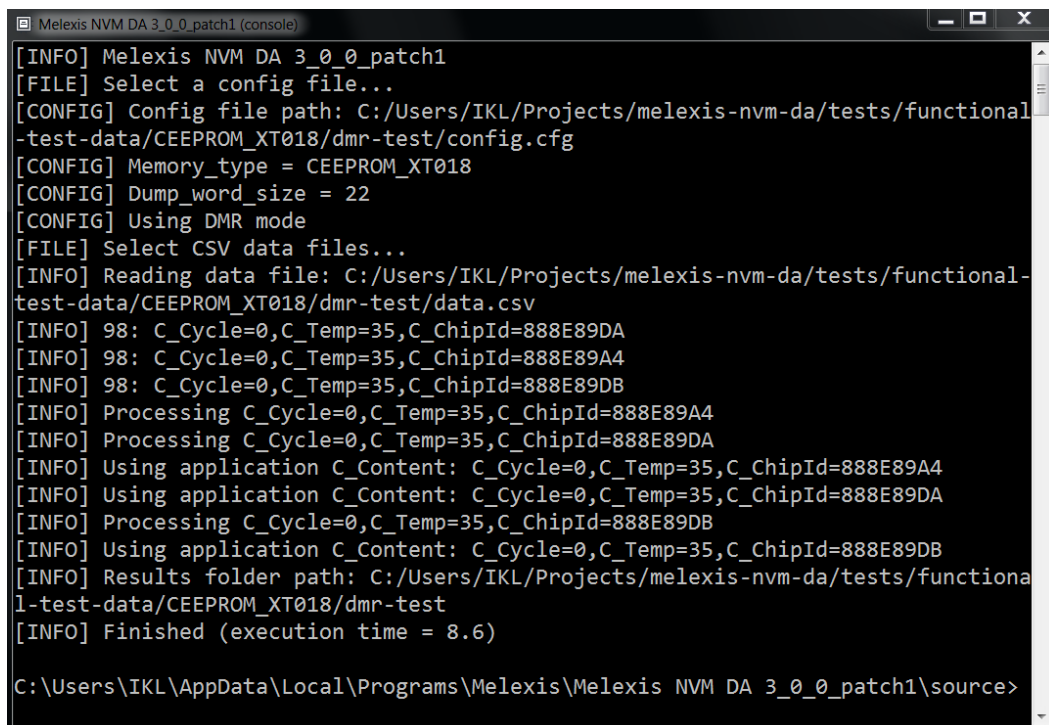
38

найгірші. Карта рівнів підтримує DMR і cell current дані. Для кожної умови, що аналізується, створюється окремий файл з картою рівнів.

Крім графічного представлення карти рівнів, є підтримка запису карти в CSV-файл. Це дозволяє знайти точне значення рівнів комірок. Також передбачається створення CSV-файлів, у яких записується перелік всіх номерів сторінок, ліній бітів, окремих бітів. Разом з цим переліком вказується рівень комірки й очікуваний логічний рівень.

4.4 Текстовий інтерфейс розробки

Один із варіантів взаємодії з користувачем – це текстовий інтерфейс. Вказується розміщення файлу конфігурації та файлів вимірів. Після цього програмно-апаратний комплекс виводить стан виконання аналізу енергонезалежної пам'яті (рисунок 4.3).



```
[INFO] Melexis NVM DA 3_0_0_patch1
[FILE] Select a config file...
[CONFIG] Config file path: C:/Users/IKL/Projects/melexis-nvm-da/tests/functional-test-data/CEEPROM_XT018/dmr-test/config.cfg
[CONFIG] Memory_type = CEEPROM_XT018
[CONFIG] Dump_word_size = 22
[CONFIG] Using DMR mode
[FILE] Select CSV data files...
[INFO] Reading data file: C:/Users/IKL/Projects/melexis-nvm-da/tests/functional-test-data/CEEPROM_XT018/dmr-test/data.csv
[INFO] 98: C_Cycle=0,C_Temp=35,C_ChipId=888E89DA
[INFO] 98: C_Cycle=0,C_Temp=35,C_ChipId=888E89A4
[INFO] 98: C_Cycle=0,C_Temp=35,C_ChipId=888E89DB
[INFO] Processing C_Cycle=0,C_Temp=35,C_ChipId=888E89A4
[INFO] Processing C_Cycle=0,C_Temp=35,C_ChipId=888E89DA
[INFO] Using application C_Content: C_Cycle=0,C_Temp=35,C_ChipId=888E89A4
[INFO] Using application C_Content: C_Cycle=0,C_Temp=35,C_ChipId=888E89DA
[INFO] Processing C_Cycle=0,C_Temp=35,C_ChipId=888E89DB
[INFO] Using application C_Content: C_Cycle=0,C_Temp=35,C_ChipId=888E89DB
[INFO] Results folder path: C:/Users/IKL/Projects/melexis-nvm-da/tests/functional-test-data/CEEPROM_XT018/dmr-test
[INFO] Finished (execution time = 8.6)

C:\Users\IKL\AppData\Local\Programs\Melexis\Melexis NVM DA 3_0_0_patch1\source>
```

Рисунок 4.3 – Текстовий інтерфейс розробки

4.5 Графічний інтерфейс розробки

Для більш ефективної роботи з програмно-апаратним комплексом, передбачається графічний інтерфейс (рисунок 4.4).

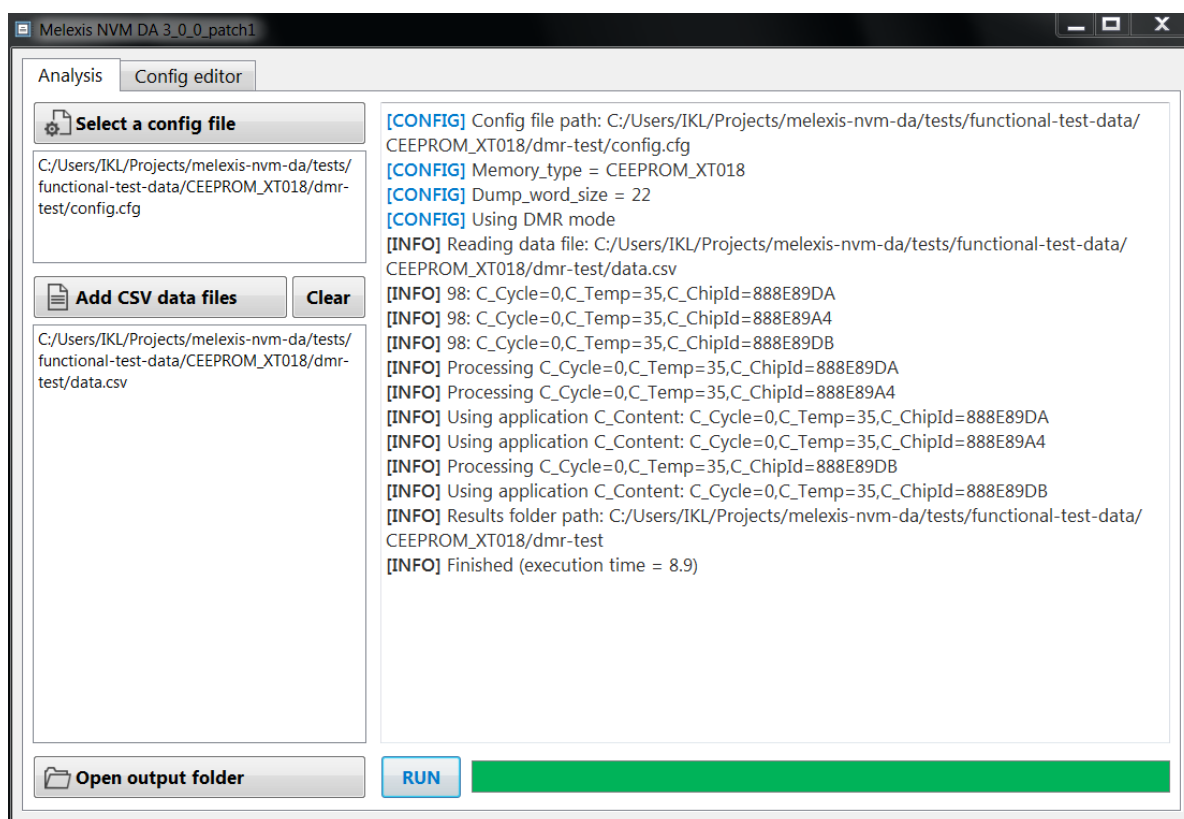


Рисунок 4.4 – Графічний інтерфейс розробки

Інтерфейс надає можливість зручно обирати вхідні файли і бачити, крім текстового виводу статусу, індикатор виконання, за яким можна оцінити швидкість опрацювання вхідних даних.

Є можливість змінювати та зберігати текст файлу конфігурації в окремому вікні. Також передбачається можливість вибору готових шаблонів для різних видів енергонезалежної пам'яті автомобільної промисловості. Вікно зміни файлу конфігурації показано на рисунку 0.9.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045420.004 ПЗ

Арк.

40

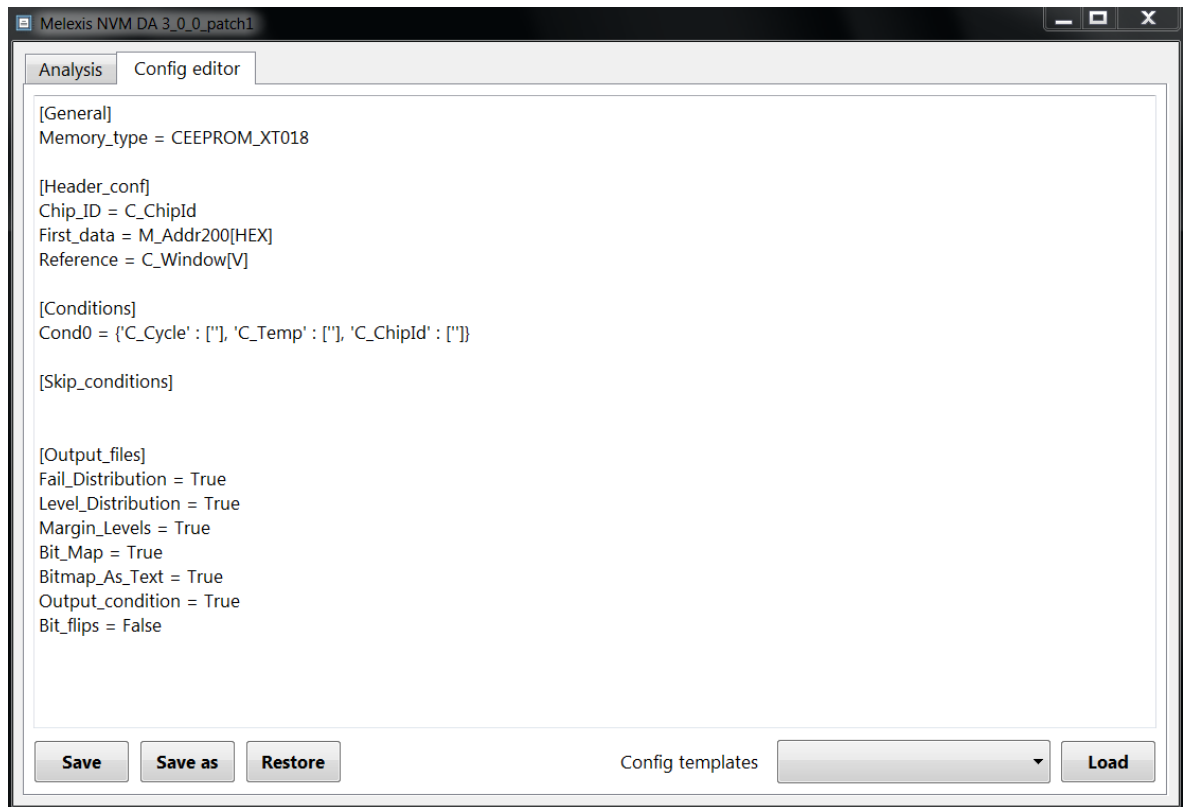


Рисунок 4.5 – Вікно зміни файлу конфігурації

У випадку помилки аналізу, користувач отримає вивід інформації про деталі помилки й умову, за якої вона відбулася (рисунок 4.6). Окрім виводу помилки на екран, записується в окремий файл більш детальна інформація про поточну помилку: точний час, програмний модуль, рядок коду, опис.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ.045420.004 ПЗ

Арк.

41

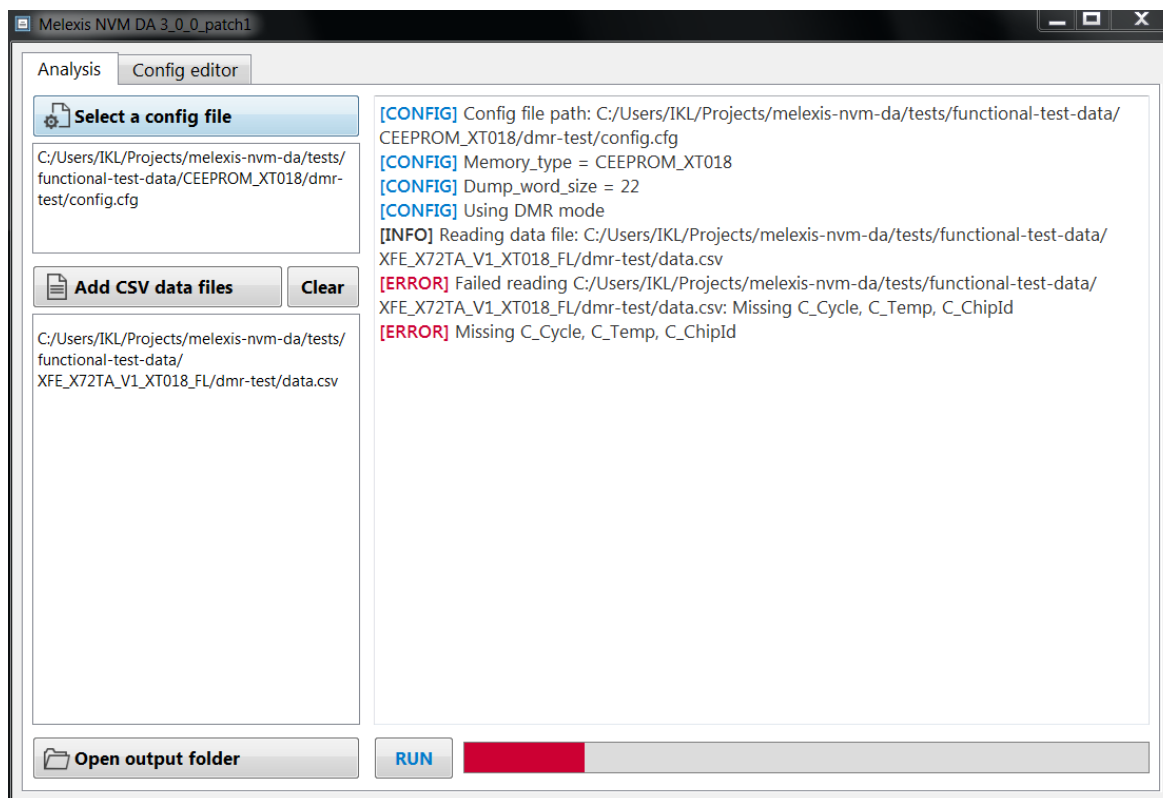


Рисунок 4.6 – Повідомлення про помилку аналізу

4.6 Обробка отриманих результатів аналізу

Після успішного програмного аналізу вхідних даних, рекомендовано провести обробку отриманих текстових CSV-файлів із результатами. Це можна зробити за допомогою програми Microsoft Excel чи подібних. Вихідні файли можна представити у вигляді графіків, із попереднім групуванням і фільтруванням умов, використовуючи, наприклад, інструмент PivotTable в Microsoft Excel.

На рисунку 1.1 показано графік розподілу помилок, який отриманий з вихідного CSV-файлу розподілу помилок.

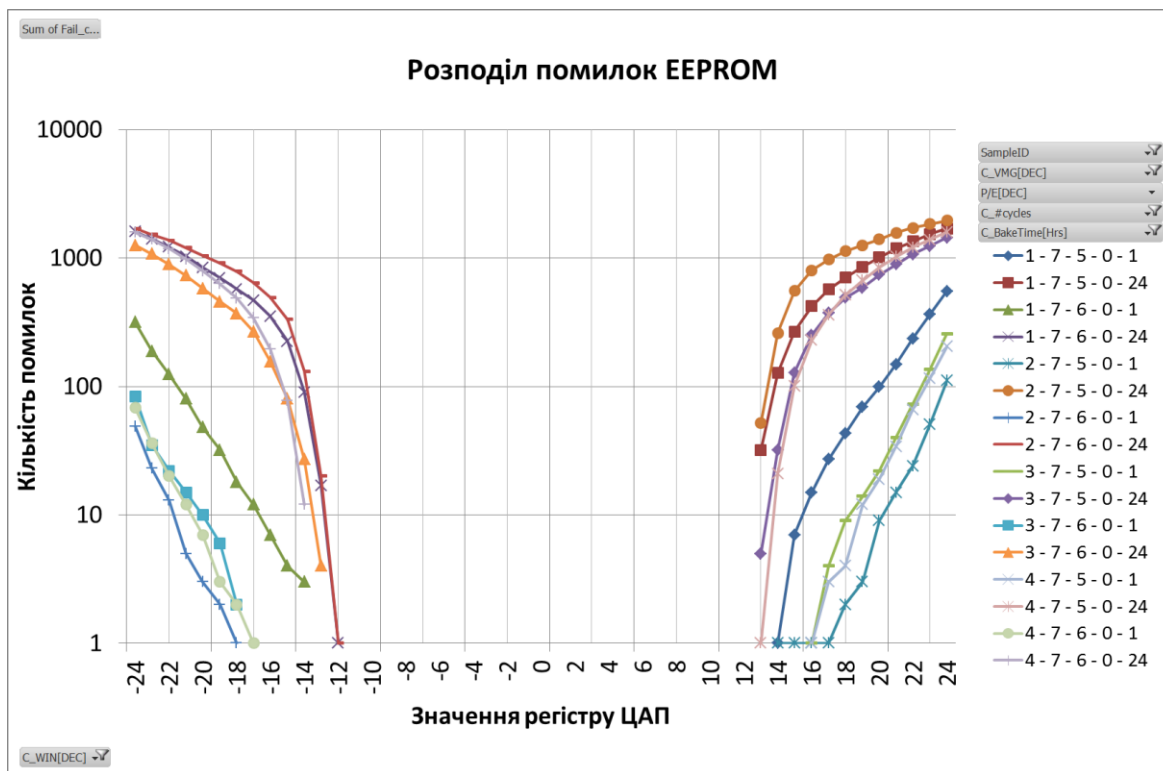


Рисунок 4.7 – Графік розподілу помилок

На вісі X розміщується значення регістру ЦАП, на вісі Y – кількість помилок у логарифмічному масштабі. Кожна лінія показує розподіл помилок енергонезалежної пам'яті в одному з чотирьох ІС за різного часу в термокамері. Зліва показано лінії для комірок із логічним рівнем «1», справа – логічним рівнем «0». Посередині утворюється область, яка вільна від будь-яких помилок. На графіку помітно, що розподіл помилок для всіх ІС після 24 годин у термокамері більш наближений до центрального очікуваного значення ЦАП. На основі цього графіка інженер може зробити висновок про те, що після 24 годин у термокамері пам'ять втратила заряди у транзисторі, і під час операції читання компаратор раніше визначає неправильний логічний рівень. Якщо продовжити експеримент із більшою кількістю годин, то буде момент, коли заряди у транзисторах настільки зменшаться, що логічний рівень буде завжди неправильний. Зазвичай, кожна адреса в автомобільній NVM має додаткові біти, де зберігаються коди корекції. Ці коди можуть визначити і

виправити одну помилку логічного рівня. Більша кількість помилок в одній адресі призводить до запуску внутрішніх систем захисту ІС від помилок, які приймають рішення про можливе перезавантаження ІС тощо.

Для більш простої перевірки рівнів, будується графік граничних рівнів (рисунок 4.8).

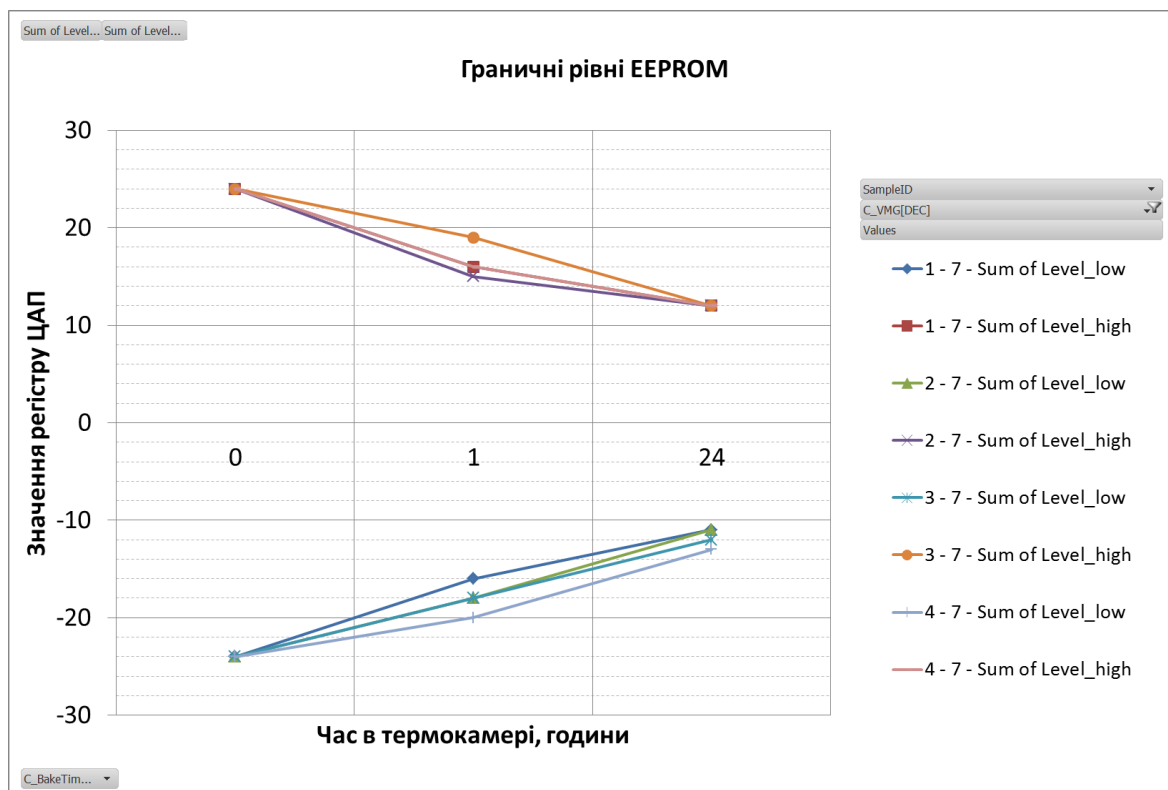


Рисунок 4.8 – Графік граничних рівнів

На вісі X зображено кількість годин, які витримується ІС, на вісі Y – значення регістру ЦАП. Кожна лінія показує результати для кожної ІС. Помітно, що граничні рівні стають ближчими до очікуваного значення ЦАП зі збільшенням часу в термокамері. Графіки граничних рівнів краще будувати, коли виміри проводяться за великої кількості умов: температури читання, час витримки в термокамері, кількість перезапису для тестування збуренням.

На рисунку 1.3 зображено графік розподілу струмів.

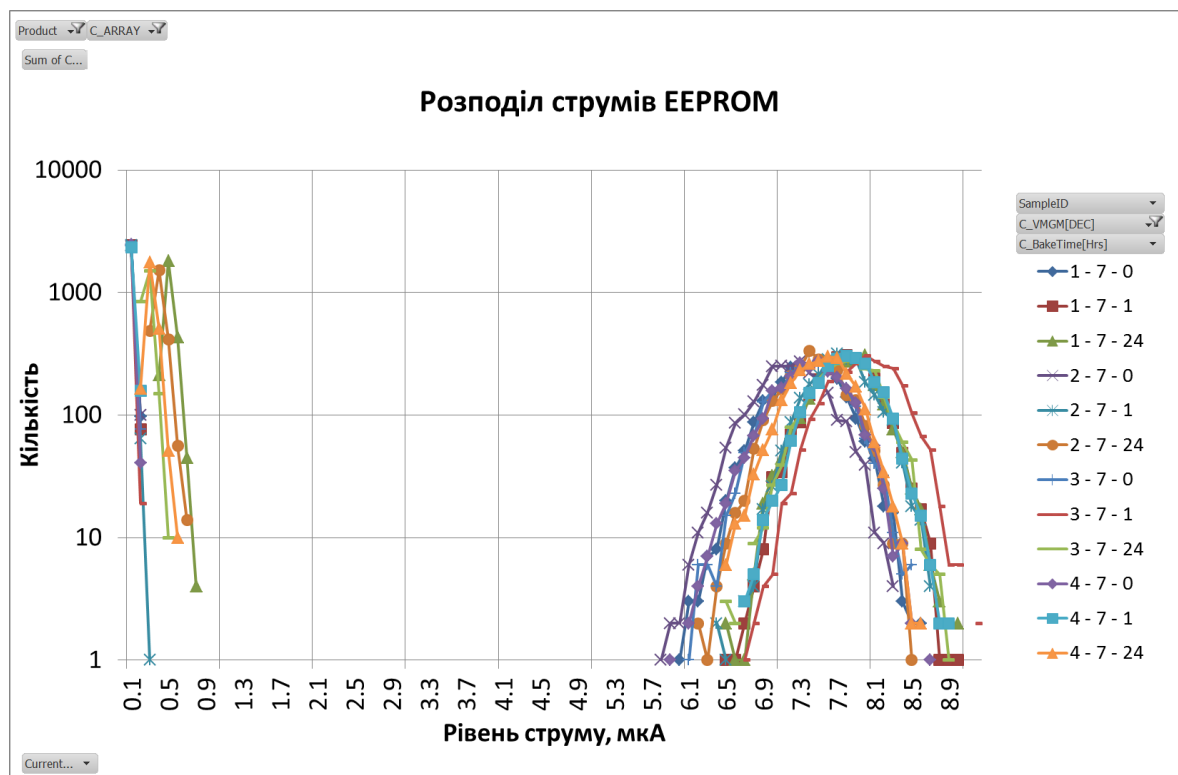


Рисунок 4.9 – Графік розподілу струмів

Вісь X показує рівень струму в мікроамперах, вісь Y – кількість комірок з таким рівнем. Якщо на графіку розподілу помилок IC занадто відхиляється значеннями від інших, то на графіку розподілу струмів спостерігається таке ж відхилення, але в інших одиницях.

Трапляється, що на графіку розподілу помилок помітна одна чи кілька горизонтальних ліній із кількістю помилок «1». Це може означати, що у енергонезалежній пам'яті є FG MOS із слабким рівнем. Тому необхідно знайти точне розміщення в NVM цього біта. На рисунку 0.14 показано графік розподілу помилок енергонезалежної пам'яті Flash, який обраний для дослідження слабого рівня FG MOS. Щоб знайти розміщення цього біта, також необхідно переглянути файл розподілу рівнів (рисунок 4.11).

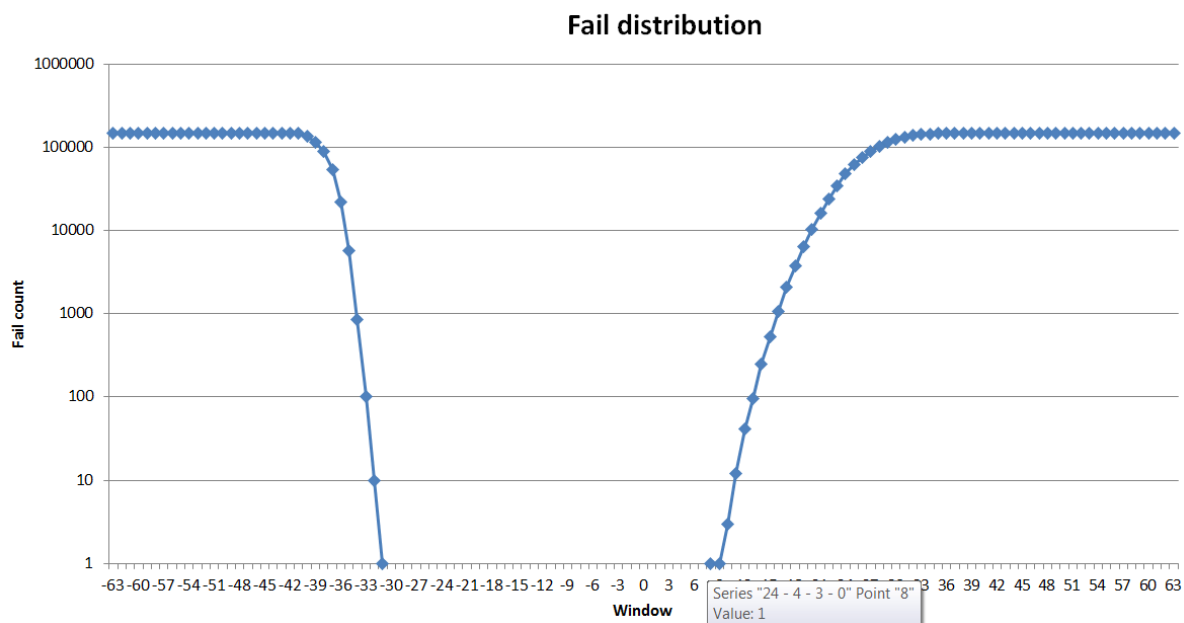


Рисунок 4.10 – Графік розподілу помилок для дослідження

AC[DEC]	R0_FL[HEX]	R1_FL[HEX]	R2_FL[HEX]	R3_FL[HEX]	Distribution_level
-6	'0001	'0188	'00D8	'3344	0
-5	'0001	'0188	'00B8	'3344	0
-4	'0001	'0188	'0098	'3344	0
-3	'0001	'0188	'0078	'3344	0
-2	'0001	'0188	'0058	'3344	0
-1	'0001	'0188	'0038	'3344	0
0	'0001	'0188	'0018	'3344	0
0	'0001	'0188	'0018	'2B44	0
1	'0001	'0188	'0038	'2B44	0
2	'0001	'0188	'0058	'2B44	0
3	'0001	'0188	'0078	'2B44	0
4	'0001	'0188	'0098	'2B44	0
5	'0001	'0188	'00B8	'2B44	0
6	'0001	'0188	'00D8	'2B44	0
7	'0001	'0188	'00F8	'2B44	1
8	'0001	'0188	'0118	'2B44	0
9	'0001	'0188	'0138	'2B44	2
10	'0001	'0188	'0158	'2B44	9
11	'0001	'0188	'0178	'2B44	29
12	'0001	'0188	'0198	'2B44	55
13	'0001	'0188	'01B8	'2B44	151
14	'0001	'0188	'01D8	'2B44	282
15	'0001	'0188	'01F8	'2B44	549
16	'0001	'0188	'0218	'2B44	1002
17	'0001	'0188	'0238	'2B44	1659

Рисунок 4.11 – Знайдений рівень у файлі розподілу рівнів

На графіку розподілу помилок присутній FGMOS із слабким рівнем «8». У файлі розподілу рівнів можна знайти значення розподілу «1», яке має рівень «7». Це пояснюється тим, що розподіл помилок показує кількість помилок саме на цьому рівні, а розподіл рівнів визначає останній рівень перед помилкою. Для енергонезалежної пам'яті невеликого розміру розміщення слабого біта можна побачити за кольором на карті рівнів. Але у даному дослідженні – це Flash, яка має багато адрес. Тому не виходить визначити розміщення слабого біта за кольором карти рівнів (рисунок 4.12).

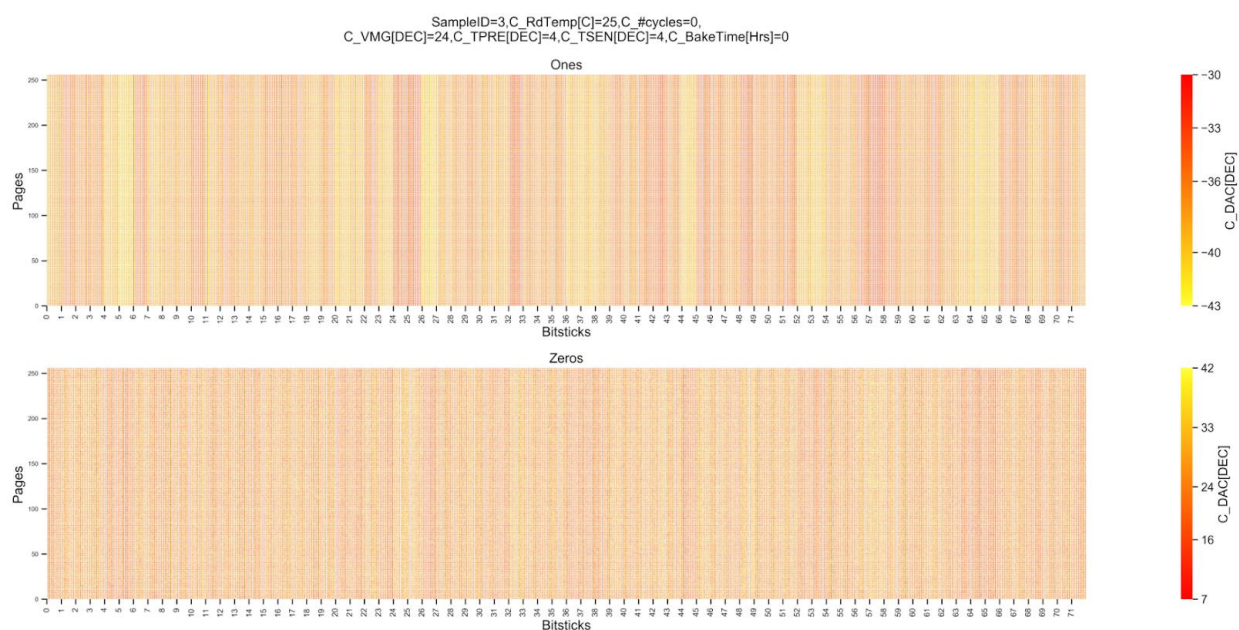


Рисунок 4.12 – Карта рівнів для досліджуваних умов

У даному випадку потрібно знайти розміщення слабого біта в CSV-файлі з розміщеннями всіх рівнів енергонезалежної пам'яті (рисунок 4.13).

	A	B	C	D	E	F	
1	Physical_addr[HEX]	Page	Bit_Line	Bit_Stick	Level[DEC]	Reference[BIN]	
167159	'911	145	1	26	7	0	
294914							
294915							

Рисунок 4.13 – Розміщення слабого біта в NVM

Отже, досліджуваний слабкий біт має адресу «911» у шістнадцятковому представленні, сторінку «145», першу лінію бітів і біт «26». Маючи інформацію про розміщення слабого біта, можна впевнитися у правильності аналізу, переглянувши також текстовий формат карти рівнів (рисунок 4.14).

	A	B	PA	PB	PC	PD	PE	PF	PG
259	Physical_addr[HEX]		'e	'f	'0	'1	'2	'3	'4
260		Zeros_pages	BL14_BS2'	BL15_BS2'	BL0_BS26	BL1_BS26	BL2_BS26	BL3_BS26	BL4_B'
366	'960	150		24		22		27	
367	'950	149		22		17		22	
368	'940	148		31		19		25	
369	'930	147		33		19		17	
370	'920	146		25		24		26	
371	'910	145		34		7		25	
372	'900	144		32		20		29	
373	'8f0	143		30		16		16	
374	'8e0	142		29		24		30	
375	'8d0	141		29		18		24	
376	'8c0	140		30		25		26	
377	'8b0	139		26		17		18	
378	'8a0	138		31		26		26	
379	'890	137		27		23		24	

Рисунок 4.14 – Розміщення слабого біта в текстовому форматі карти рівнів

ВИСНОВКИ

Враховуючи потребу автомобільної промисловості в автоматизації тестування та верифікації NVM, було розроблено програмно-апаратний комплекс для аналізу параметрів енергонезалежної пам'яті.

У процесі розробки досліджено структуру енергонезалежної пам'яті, методи верифікації блоків енергонезалежної пам'яті в автомобільних інтегральних мікросхемах. Розроблено алгоритми аналізу вимірів енергонезалежної пам'яті для DMR і cell current вимірів. Визначено формати вхідних файлів конфігурації та вимірів, функції для читання вхідних даних та генерації вихідних результатів аналізу. Для зручності користування розроблено графічний інтерфейс із зручним представленням стану виконання аналізу. Також здійснено тестування розробленого програмно-апаратного комплексу на вимірах основних видів енергонезалежної пам'яті справжніх автомобільних інтегральних мікросхем.

Для мінімізації витрат на створення, тестування й майбутні модифікації програмного коду використано мову програмування Python з відкритими бібліотеками numpy, pandas, seaborn, які надають зручну й надійну основу для створення програм аналізу даних.

Розроблений комплекс успішно впроваджений у ДП «МЕЛЕКСІС-УКРАЇНА» у відділі верифікації енергонезалежної пам'яті, що дозволило суттєво зменшити витрати часу на аналіз DMR і cell current вимірів енергонезалежної пам'яті.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Automotive Electronics Council Component Technical Committee. Failure Mechanism Based Stress Test Qualification For Integrated Circuits (base document). *Automotive Electronics Council*. 2014. Q100 - REV-H. P. 3 : URL: http://www.aecouncil.com/Documents/AEC_Q100_Rev_H_Base_Document.pdf (дата звернення: 15.04.2021).
2. Kumar D. A Study about Non-Volatile Memories. *Preprints*. 2016, 2016070093. P. 2 : URL: <https://www.preprints.org/manuscript/201607.0093/v1> (дата звернення: 21.04.2021).
3. Bez R., Camerlenghi E., Modelli A., Visconti A. Introduction to Flash memory. *Proceedings of the IEEE*. 2003. № 91(4). PP. 489-502 : URL: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.464.7114&rep=rep1&type=pdf> (дата звернення: 21.04.2021).
4. Windbacher T. Engineering Gate Stacks for Field-Effect Transistors : URL: <https://www.iue.tuwien.ac.at/phd/windbacher/node14.html> (дата звернення: 25.04.2021).
5. Ledford S. Non-volatile memory technology overview. *Freescale Semiconductor*. 2004. AN1837, 28. P. 16 : URL: <https://www.nxp.com/docs/en/application-note/AN1837.pdf> (дата звернення: 01.05.2021).
6. White M. Microelectronics reliability: Physics-of-failure based modeling and lifetime evaluation. *Jet Propulsion Laboratory. National Aeronautics and Space Administration (NASA)*. 2008. P. 9 : URL: <https://trs.jpl.nasa.gov/bitstream/handle/2014/40791/08-05.pdf?sequence=1> (дата звернення: 04.05.2021).
7. Павловський О. М., Котельнікова О. С. Шляхи вдосконалення діагностичних тестів енергонезалежної пам'яті на прикладі Detailed Marginal Read тесту. *Вісник НТУУ «КПІ». Серія Приладобудування :*

збірник наукових праць. 2018. Вип. 55(1). С. 89-93. Бібліогр. 8 назв : URL:
<https://ela.kpi.ua/handle/123456789/32861> (дата звернення: 04.05.2021).

8. PXIe-6547 Digital Waveform Instrument. *NI* : URL: <https://www.ni.com/en-us/support/model.pxie-6547.html> (дата звернення: 04.05.2021).
9. PXI-4071 Digital Multimeter. *NI* : URL: <https://www.ni.com/en-us/support/model.pxi-4071.html> (дата звернення: 04.05.2021).

					ІАЛЦ.045420.004 ПЗ	Арк.
						51
Змін.	Арк.	№ докум.	Підпис	Дата		