

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

Факультет електроніки

(повна назва інституту/факультету)

Кафедра акустичних та мультимедійних електронних систем

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри АМЕС



— С.А. Найда

(ініціали, прізвище)

« 12 » червня 2024 р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності (спеціалізації) 171 Електроніка (Електронні системи мультимедіа та засоби Інтернету речей)

(код та назва спеціальності)

на тему: «Твердотільний запис-відтворення надвеликих обсягів інформації»

Виконав (-ла) студент IV курсу, групи ДВ-01

(шифр групи)

Бурковський Ігор Олегович

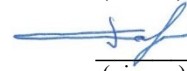
(прізвище, ім'я, по-батькові)



(підпис)

Керівник д.т.н., професор Розорінов Г.М.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)



(підпис)

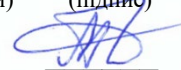
Консультант

(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент зав. каф. МЕ, д.т.н., доц. Татарчук Д.Д.

(посада, вчене звання, науковий ступінь, прізвище, ініціали)



(підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без відповідних

посилань.

Студент БМ

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет електроніки

Кафедра акустичних та мультимедійних електронних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 171 «Електроніка»

Освітньо-професійна програма «Електронні системи мультимедіа та засоби Інтернету речей»

ЗАТВЕРДЖУЮ

Завідувач кафедри АМЕС



Сергій НАЙДА

«20» квітня 2024 р.

ЗАВДАННЯ

на дипломну роботу студенту

Бурковському Ігорю Олеговичу

1. Тема роботи «Твердотільний запис-відтворення надвеликих обсягів інформації», керівник роботи Розорінов Георгій Михайлович, д.т.н., професор, затверджені наказом по університету від «23» __05__ 2024 р. №2076-с
2. Термін подання студентом роботи: «7» червня 2024 р.

3. Вихідні дані до роботи: Обрана досліджувана твердотільна флеш-пам'ять типу NAND, статистичні дані, технічна інформація з обраної теми, технічні параметри з наукових статей та книжок.

4. Зміст роботи :

РОЗДІЛ 1. ВСТУП. ТВЕРДОТІЛЬНА ПАМ'ЯТЬ, ЇЇ ВИДИ ТА ПЕРЕВАГИ.
ОСНОВИ FLASH

РОЗДІЛ 2. СТРУКТУРА І ПОРІВНЯННЯ NOR І NAND. ВИБРАНА NAND

РОЗДІЛ 3. ФАЙЛОВА СТРУКТУРА NAND, ЯК ВОНО ПРАЦЮЄ, ЧАСОВІ
ДІАГРАМИ

РОЗДІЛ 4. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ. ІНТЕРФЕЙС

ВИСНОВКИ

ДОДАТОК А

5. Перелік ілюстративного матеріалу: презентація складена з набору слайдів де узагальнено матеріали та описано головне.

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 20.04.2024р.

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання	20.04.2024	виконано
2	Написання звіту дипломної роботи	23.04.2024	виконано
3	Пошук джерел інформації	23.04.2024	виконано
4	Написання першого розділу	05.05.2024	виконано
5	Написання другого розділу	12.05.2024	виконано
6	Написання третього розділу	24.05.2024	виконано
7	Написання четвертого розділу	02.06.2024	виконано
8	Оформлення дипломної роботи	04.06.2024	виконано
9	Підготовка та оформлення презентації до захисту	08.06.2024	

Студент

Ігор БУРКОВСЬКИЙ

Керівник



 Георгій РОЗОРІНОВ

УДК 004.087

РЕФЕРАТ

Бурковський І. Дослідження твердотільної флеш-пам'яті, особливостей файлової структури NAND: дипломна робота бакалавра : 171 Електроніка. – КПІ ім. Ігоря Сікорського, Київ, 2024.

Обсяг роботи 59 сторінок, 23 ілюстрації, 11 таблиць та 13 джерел інформації.

Об'єктом дослідження є твердотільна флеш-пам'ять її файлова структура.

Метою роботи є глибоке та детальне дослідження файлової структури NAND її порівняння з NOR, визначення переваг та недоліків кожної.

Методом дослідження є теоретичне дослідження особливостей файлової структури NAND її команд та часових діаграм на основі наукових статей, книг та доступних інтернет джерел.

Ключові слова: байт, твердотільна пам'ять, NAND, NOR, флеш-пам'ять, передача даних.

UDK 004.087

ABSTRACT

Burkovskyi I. Research of solid-state flash memory, features of the NAND file structure: Bachelor's thesis: 171 Electronics - Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2024.

The volume of work is 59 pages, 23 illustrations, 11 tables and 13 sources of information.

The object of study is solid-state flash memory and its file structure.

The purpose of the study is a deep and detailed study of the NAND file structure, its comparison with NOR, and the identification of the advantages and disadvantages of each.

The research method is a theoretical study of the features of the NAND file structure, its commands and timing diagrams based on scientific articles, books and available Internet sources.

Keywords: solid-state memory, NAND, NOR, flash memory, bytes, data transfer.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	8
ВСТУП.....	9
1 ВСТУП. ТВЕРДОТІЛЬНА ПАМ'ЯТЬ, ЇЇ ВИДИ ТА ПЕРЕВАГИ. ОСНОВИ FLASH.....	11
2 СТРУКТУРА І ПОРІВНЯННЯ NOR І NAND. ВИБРАНА NAND.....	17
3 ФАЙЛОВА СТРУКТУРА NAND, ЯК ВОНО ПРАЦЮЄ, ЧАСОВІ ДІАГРАМИ.....	23
4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ. ІНТЕРФЕЙС.....	42
ВИСНОВКИ.....	51
ПЕРЕЛІК ПОСИЛАНЬ.....	54
ДОДАТОК А.....	56

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

SSD	– solid-state drive, твердотільний накопичувач;
HDD	– hard (magnetic) disk drive, накопичувач на жорсткому (магнітному) диску;
DOC	– DiskOnChip, вбудовані флеш-накопичувачі від M-Systems Flash Disk Pioneers Ltd;
SLC	– single-Level Cell, одnorівнева комірка;
MLC	– multi-Level Cell, багаторівнева комірка;
RAM	– random-access memory, пам'ять з довільним доступом або оперативна пам'ять;
XiP	– execute in place, виконати прямо на місці;
ECC	– error correction code, код виправлення помилок;
SRAM	– static random access memory, статична пам'ять із довільним доступом.

ВСТУП

З кожним роком все більше людей цікавляться технологіями, а потреба в зберіганні та обробці надвеликих обсягів інформації стає все більш актуальною. Сучасний розвиток технологій значно вплинув на спосіб, яким ми зберігаємо та обробляємо дані. Завдяки поширенню мобільних пристроїв і бездротових технологій, люди можуть обробляти великі обсяги інформації в будь-який час та в будь-якому місці. Поява нових форматів зберігання даних, таких як твердотільна пам'ять NAND, відкрила безліч можливостей для зберігання та доступу до інформації.

Порівняння систем NAND і NOR флеш-пам'яті є важливим аспектом цього дослідження. Твердотільна пам'ять типу NAND дозволяє зберігати величезні обсяги даних у компактних і надійних пристроях. Вона має вищу щільність зберігання та нижчу вартість на одиницю обсягу пам'яті порівняно з NOR флеш-пам'яттю, яка, у свою чергу, забезпечує швидший доступ до даних і використовується у випадках, коли важлива швидкість читання. Розширення можливостей таких технологій забезпечує широкий доступ до різноманітних додатків та служб, задовольняючи потреби користувачів у швидкому та безпечному зберіганні інформації.

По-перше, щоб зрозуміти цю тему дослідження, необхідно пояснити, що таке твердотільна пам'ять типу NAND. Це електронні пристрої, призначені для зберігання великої кількості інформації. Вони бувають різних типів, наприклад флеш-накопичувачі USB, SSD, карти пам'яті тощо, і бувають усіх форм і розмірів, від маленьких портативних пристроїв до потужних систем зберігання.

Практичне значення даної бакалаврської роботи полягає в тому, що сьогодні, в умовах стрімкого розвитку індустрії твердотільної пам'яті, її технічне оснащення та якісні показники є дуже важливими. Суспільство прагне використовувати такі пристрої для зберігання великих обсягів даних і зробити їх зручними у використанні завдяки невеликому розміру. Враховуючи актуальність роботи, була поставлена наступна задача: Дослідження малих електронних

пристроїв, що зберігають великі обсяги інформації. Особливо їх характеристики та якісні показники.

Було поставлено такі завдання для дослідження :

а) вивчити актуальну літературу та переглянути останні наукові дослідження, що стосуються твердотільної пам'яті типу NAND;

б) провести детальний аналіз використовуваного технічного обладнання для зберігання великих обсягів даних;

в) провести порівняння систем NAND і NOR флеш-пам'яті для оцінки їх переваг та недоліків;

г) детально дослідити файлову структуру NAND, як вона працює;

д) здійснити дослідження програмного забезпечення та інтерфейсу файлової структури NAND

е) зробити висновки по проведеному дослідженню.

1 ВСТУП. ТВЕРДОТІЛЬНА ПАМ'ЯТЬ, ЇЇ ВИДИ ТА ПЕРЕВАГИ. ОСНОВИ FLASH

Твердотільна пам'ять (SSD) — це тип пристроїв зберігання даних, який використовує інтегральні схеми для зберігання даних на постійній основі, без рухомих частин, на відміну від традиційних жорстких дисків (HDD), які використовують магнітні пластини та механічні головки для зчитування і запису даних.

SSD має кілька переваг. По-перше, вони значно швидші в читанні та записі даних порівняно з HDD, з часом доступу до даних в мікросекунди. Це призводить до значно швидшого завантаження операційних систем і програм. По-друге, відсутність рухомих частин робить SSD більш стійкими до фізичних пошкоджень та вібрацій, що підвищує їх надійність, особливо в портативних пристроях. По-третє, SSD споживають менше енергії, що особливо важливо для ноутбуків і мобільних пристроїв, оскільки це продовжує час автономної роботи. Крім того, відсутність рухомих частин робить SSD безшумними, а компактність дозволяє створювати тонші та легші пристрої [1].

Існує кілька видів твердотільної пам'яті. SATA SSD використовують інтерфейс SATA для підключення до системи і є найпоширенішими, забезпечуючи помітне покращення швидкості у порівнянні з HDD. NVMe SSD використовують інтерфейс NVMe і підключаються через шину PCIe, що пропонує ще вищі швидкості читання та запису. mSATA і M.2 SSD представляють собою компактні варіанти SATA і NVMe SSD відповідно, причому M.2 SSD можуть бути використовувані в ультратонких ноутбуках та десктопах. U.2 SSD використовують інтерфейс PCIe та призначені для серверів і робочих станцій. External SSD — це зовнішні накопичувачі, які підключаються до комп'ютера через USB, Thunderbolt або інші інтерфейси, забезпечуючи портативність та високу швидкість передачі даних [2].

Технології зберігання даних у SSD включають кілька типів: SLC (Single-Level Cell), що зберігає один біт інформації на комірку пам'яті і забезпечує високу

швидкість та надійність, але має високу вартість; MLC (Multi-Level Cell), що зберігає два біти інформації на комірку і забезпечує баланс між вартістю, швидкістю та надійністю; TLC (Triple-Level Cell), що зберігає три біти інформації на комірку і є дешевшими, але менш надійними та повільнішими; та QLC (Quad-Level Cell), що зберігає чотири біти інформації на комірку і є найдешевшими та найменш надійними, але забезпечують велику ємність. Твердотільна пам'ять продовжує розвиватися, стаючи дедалі швидшою, надійнішою та доступнішою, що робить її оптимальним вибором для багатьох сучасних пристроїв [3].

Flash-пам'ять — це тип твердотільної пам'яті, що використовується для зберігання даних на довготривалій основі. На рисунку 1 зображено типовий приклад Flash-пам'яті. Вона є основою багатьох типів зберігання, включаючи SSD, USB флеш-накопичувачі, карти пам'яті та інші. Flash-пам'ять є енергонезалежною, тобто зберігає дані навіть при відключенні живлення. Вона забезпечує швидший доступ до даних у порівнянні з традиційними жорсткими дисками, є дуже компактною, що дозволяє використовувати її в різних пристроях, від смартфонів до серверів, і завдяки відсутності рухомих частин є менш вразливою до фізичних ушкоджень [4].

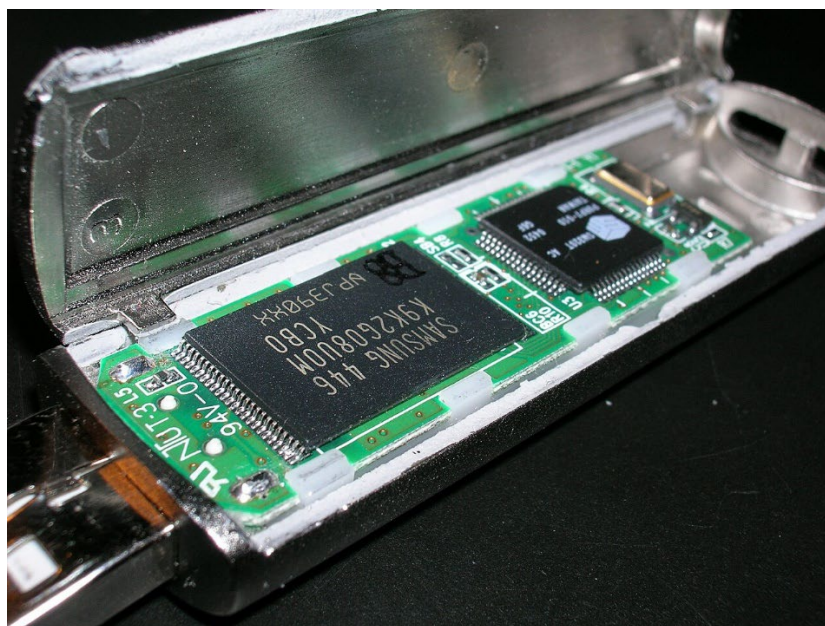


Рисунок 1 – Типовий приклад Flash-пам'яті

Серед flash-пам'яті виділяються USB флеш-накопичувачі, які використовуються для зберігання і перенесення даних між пристроями через USB-порт, і SD карти (Secure Digital), які використовуються в камерах, смартфонах, планшетах та інших портативних пристроях. Існує кілька типів SD карт. Перший тип, SD (Secure Digital), має максимальний обсяг до 2 ГБ і використовується в старіших пристроях. SDHC (Secure Digital High Capacity) карти мають ємність від 4 ГБ до 32 ГБ, забезпечуючи більшу швидкість та об'єм зберігання даних. На рисунку 2 зображено SDHC в розібраному вигляді. SDXC (Secure Digital eXtended Capacity) карти мають ємність від 64 ГБ до 2 ТБ і використовуються для зберігання великих обсягів даних, таких як відео у високій роздільній здатності. MicroSD карти є зменшеною версією SD карт і широко використовуються в смартфонах, планшетах та інших компактних пристроях, з аналогічними варіантами microSDHC та microSDXC з аналогічними характеристиками щодо обсягу зберігання.



Рисунок 2 – Зовнішній вигляд розібраної на частини SDHC

У даній дипломній роботі розглядаються основи флеш-пам'яті NAND і демонструються переваги її потужності, щільності та вартості для вбудованих систем. Він охоплює надійність даних і методи подолання типових проблем

проектування інтерфейсу, зосереджуючись на фактичних апаратних і програмних компонентах, необхідних для того, щоб дизайнери могли створювати повні та функціональні підсистеми.

Вбудовані системи традиційно використовують NOR Flash для енергонезалежної пам'яті. Багато сучасних конструкцій переходять на флеш-пам'ять NAND, щоб скористатися її високою щільністю та нижчою ціною для високопродуктивних програм.

На рисунку 3 показано, як попит на флеш-пам'ять NAND стимулювався трьома основними ринками медіа-карт для цифрових камер, флеш-накопичувачів USB і MP3-плеєрів. На рисунку 4 показано квартальне постачання флеш-пристроїв NAND, вказуючи на те, що флеш-пам'ять NAND з однорівневою коміркою (SLC) залишається лідером на ринку, за нею йде флеш-пам'ять NAND із багаторівневою коміркою (MLC).

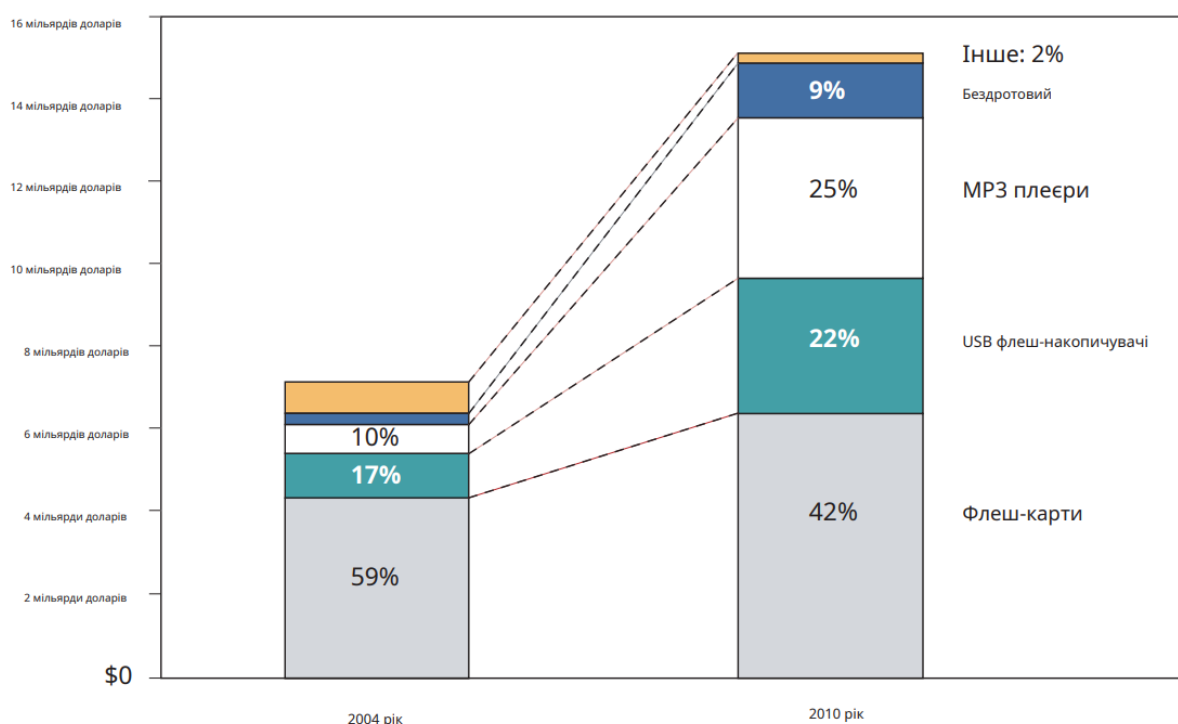


Рисунок 3 – NAND Flash Загальний доступний ринок

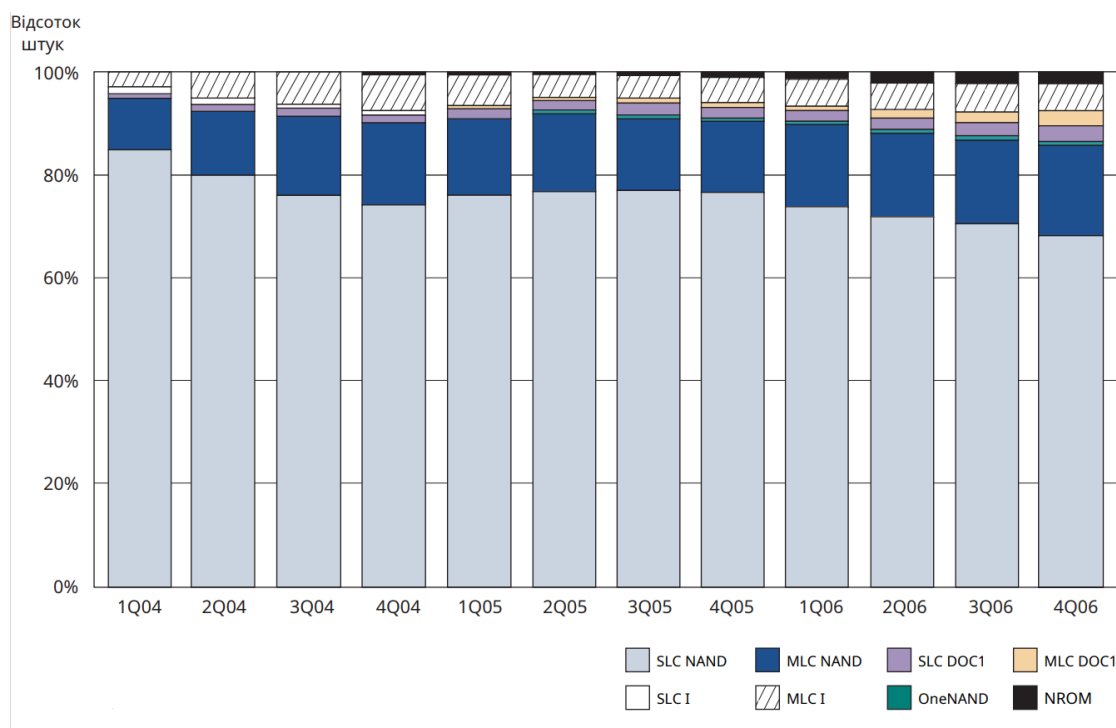


Рисунок 4 – Щоквартальна поставка флеш-пам'яті NAND

Флеш-пам'ять NAND краще підходить для задоволення вимог до пам'яті багатьох споживчих аудіо- та відеопродуктів, особливо програм малої ємності (4 ГБ або менше), ніж жорсткий диск. У міру того як тривають пошуки менш енергоспоживаючих, легших і надійніших продуктів, NAND Flash виявиться ідеальним рішенням для більш широкого спектру застосувань.

Масив флеш-пам'яті NAND згрупований у серію блоків, які є найменшими об'єктами, які можна стирати, у пристрої флеш-пам'яті NAND.

Блок флеш-пам'яті NAND становить 128 КБ. Стирання блоку встановлює всі біти на «1» (а всі байти на FFh). Програмування необхідно для зміни стертих бітів з «1» на «0». Найменша сутність, яку можна запрограмувати – це байт. Деякі флеш-пам'яті NOR можуть виконувати операції READ-While-WRITE. Хоча NAND FLASH не може виконувати операції ЧИТАННЯ та ЗАПИСУ одночасно, можна виконати операції ЧИТАННЯ/ЗАПИСУ на системному рівні за допомогою методу, який називається затіненням. Створення тіней використовувалося на персональних комп'ютерах протягом багатьох років для завантаження BIOS із повільнішої ПЗП у високошвидкісну ОЗУ.

Існує обмеження на кількість блоків NAND Flash, які можна надійно запрограмувати та стерти. Номінально кожен блок NAND витримує 100 000 циклів ПРОГРАМУВАННЯ/СТИРАННЯ. Техніка, відома як вирівнювання зносу, забезпечує рівномірне виконання всіх фізичних блоків. Щоб максимізувати термін служби конструкції, дуже важливо впровадити як вирівнювання зносу, так і керування пошкодженими блоками [5].

Висновки до 1 розділу

Досліджено, що твердотільна пам'ять (SSD) є типом пристроїв зберігання даних, що використовує інтегральні схеми для постійного зберігання інформації без рухомих частин. Ми зрозуміли, що на відміну від традиційних жорстких дисків (HDD), SSD характеризуються високою швидкістю читання і запису даних. Також виявлено, що SSD мають підвищену надійність завдяки відсутності рухомих частин, знижене енергоспоживання та компактні розміри.

Вивчаючи різновиди SSD, ми з'ясували, що існують такі типи як SATA, NVMe, mSATA, M.2, U.2 та зовнішні SSD. Аналізуючи основні технології зберігання, ми визначили, що вони включають SLC, MLC, TLC та QLC, які відрізняються за швидкістю, надійністю та вартістю.

Дослідження показали, що Flash-пам'ять є енергонезалежною і широко використовується у SSD, USB флеш-накопичувачах та SD картах. Ми виявили, що вбудовані системи все частіше використовують NAND Flash замість NOR Flash завдяки її високій щільності та низькій вартості. Дослідження підтвердили, що NAND Flash краще задовольняє вимоги сучасних пристроїв щодо енергоспоживання, ваги та надійності.

Однак, ми також зрозуміли, що NAND Flash має обмеження за кількістю циклів програмування/стирання. Це обмеження можна подолати за допомогою вирівнювання зносу та управління пошкодженими блоками, як було виявлено в ході наших досліджень.

2 СТРУКТУРА І ПОРІВНЯННЯ NOR І NAND. ВИБРАНА NAND

На рисунку 5 показано порівняння комірок NAND Flash і NOR Flash. Ефективність NAND частково пояснюється невеликою кількістю металевих контактів у рядку NAND Flash. Розмір комірки флеш-пам'яті NAND набагато менший, ніж комірка флеш-пам'яті NOR — $4F^2$ порівняно з $10F^2$ — тому що NOR Flash-елементи потребують окремого металевого контакту для кожної комірки.

	NAND	NOR
Cell Array		
Layout		
Cross Section		
Cell Size	$4F^2$	$10F^2$

Рисунок 5 – Порівняння флеш-комірок NAND і NOR

NAND Flash дуже схожий на жорсткий диск. Він заснований на секторах (на основі сторінок) і добре підходить для зберігання послідовних даних, таких як зображення, відео, аудіо або дані ПК. Хоча довільний доступ можна здійснити на системному рівні, затіняючи дані в оперативній пам'яті, для цього потрібна додаткова пам'ять RAM. Крім того, як і жорсткий диск, флеш-пристрій NAND може мати пошкоджені блоки та вимагає коду виправлення помилок (ECC) для підтримки цілісності даних.

Флеш-комірки NAND на 60 відсотків менші, ніж флеш-компоненти NOR, що забезпечує більш високу щільність, необхідну для сучасних недорогих споживчих пристроїв при значно зменшеній площі матриці. Флеш-пам'ять NAND використовується практично у всіх знімних картах, включаючи USB-накопичувачі, захищені цифрові (SD) карти, карти пам'яті, CompactFlash®картки та мультимедійні картки (MMC). Мультиплексований інтерфейс флеш-пам'яті NAND забезпечує узгоджену розпіновку для всіх останніх пристроїв і щільностей. Ця розпіновка дозволяє розробникам використовувати нижчу щільність і переходити до вищої щільності без будь-яких апаратних змін друкованої плати.

Існують певні переваги та недоліки використання NAND Flash або NOR Flash у вбудованих системах (див. таблицю 1). Флеш-пам'ять NAND найкраще підходить для програм із файлами або послідовними даними; NOR Flash найкраще підходить для довільного доступу. Переваги NAND Flash перед NOR Flash включають швидкі операції WRITE (програма) і ERASE. Переваги NOR Flash полягають у можливості довільного доступу та байтового запису [6].

Довільний доступ надає NOR Flash функцію виконання на місці (XiP), яка часто потрібна у вбудованих програмах. Все більше процесорів мають прямий інтерфейс NAND Flash і можуть завантажуватися безпосередньо з пристрою NAND Flash (без NOR Flash). Ці процесори є дуже привабливим рішенням, коли ціна, простір і ємність зберігання важливі. Використовуючи ці процесори, можливості XiP перестануть враховуватися при розробці NAND Flash у вбудованих програмах [6].

Основним недоліком NAND Flash є повільний довільний доступ; NOR Flash перешкоджає повільній роботі WRITE і ERASE.

Таблиця 1 – NAND Flash проти NOR Flash

	NAND	NOR
Переваги	Швидкий запис	Довільний доступ
	Швидке стирання	Можливий байтовий запис
Недоліки	Повільний довільний доступ	Повільний запис
	Важко записувати байтами	Повільне стирання
Додатки	Файлові (дисккові) програми	Заміна EPROM
	Голос, дані, відеозапис	Виконувати безпосередньо з енергонезалежної пам'яті
	Будь-які великі послідовні дані	

Час довільного доступу для флеш-пам'яті NOR встановлено на рівні 0,12 мкс; на флеш-пам'яті NAND час довільного доступу лише для першого байта значно менший — 25 мкс (див. Таблицю 2). Однак після того, як було здійснено початковий доступ, решта 2111 байтів зсуваються з NAND всього за 0,03 мкс на байт. Це призводить до пропускної здатності понад 23 МБ/с для 8-розрядних вводів/виводів і 37 МБ/с для 16-розрядних вводів/виводів.

Таблиця 2 – Характеристики NAND/NOR

Характеристика	NAND Flash: MT29F2G08A	NOR Flash: MT28F128J3
Довільний доступ ЧИТАННЯ	25 мкс (перший байт) 0,03 мкс кожен для решти 2111 байт	0,12 мкс
Постійна швидкість ЧИТАННЯ (галузева основа)	23 МБ/с (x8) або 37 МБ/с (x16)	20,5 МБ/с (x8) або 41 МБ/с (x16)
Випадкова швидкість ЗАПISУ	≈300 мкс/2112 байт	180 мкс/32 байти
Постійна швидкість ЗАПISУ (галузева основа)	5 МБ/с	0,178 МБ/с
Розмір видаленого блоку	128 Кб	128 Кб
Час стерття на блок (TYP)	2 мс	750 мс

Реальними перевагами флеш-пам'яті NAND є швидший час ПРОГРАМУВАННЯ та СТИРАННЯ, оскільки флеш-пам'ять NAND забезпечує стабільну продуктивність запису понад 5 МБ/с. Час стирання блоків становить вражаючі 2 мс для NAND Flash порівняно з 750 мс для NOR Flash. Очевидно, що флеш-пам'ять NAND пропонує кілька вагомих переваг. Одна проблема полягає в тому, що він не дуже підходить для прямого довільного доступу. Як зазначалося раніше, це можна вирішити за допомогою відстеження коду.

NAND Flash пропонує декілька структурних переваг перед NOR Flash, починаючи з кількості контактів. Вимоги до апаратних контактів для інтерфейсів NAND Flash і NOR Flash помітно відрізняються. NOR Flash вимагає приблизно 41 контакту вводу-виводу для 16-розрядного пристрою, тоді як NAND Flash вимагає лише 24 контактів для порівнянного інтерфейсу (див. таблицю 3). Мультиплексована шина команд, адрес і даних зменшує кількість необхідних контактів майже на 42 відсотки. Додатковою перевагою мультиплексного інтерфейсу є те, що флеш-пристрої NAND з більшою щільністю можуть підтримуватися за допомогою тієї ж апаратної конструкції та друкованої плати (PCB), що використовуються для меншої щільності. Загальний пакет TSOP-1

використовувався протягом багатьох років, і ця функція дозволяє клієнтам переходити на флеш-пристрої NAND з більшою щільністю, використовуючи той самий дизайн друкованої плати.

Ще одна перевага флеш-пам'яті NAND проявляється в варіантах упаковки: флеш-пам'ять NAND пропонує монолітний кристал ємністю 2 Гб або може підтримувати до чотирьох складених кристалів, вміщуючи пристрій ємністю 8 Гб у тому самому корпусі TSOP-1. Це дає змогу єдиному пакету та інтерфейсу підтримувати вищу щільність у майбутньому.

Таблиця 3 – Необхідні апаратні контакти

NAND Flash: 24 контакти (x16)		NOR Flash: 41 контакт	
Інтерфейс пристрою вводу-виводу, що складається з:		Інтерфейс довільного доступу, зазвичай складається з:	
CE#	Включення мікросхеми	CE#	Включення мікросхеми
WE#	Увімкнути запис	WE#	Увімкнути запис
RE#	Увімкнути читання	OE #	Увімкнути вихід
CLE	Увімкнення засувки команд	D[15:0]	Шина даних
ALE	Увімкнення фіксації адреси	A[20:0]	Адресна шина
I/O[7:0]	Шина даних (I/O[15:0} для частин x16)	WP#	Захист від запису
WP#	Захист від запису		
R/B#	Готовий/зайнятий		
PRE	Читання під час увімкнення (корисно для завантаження системи)		

Висновки до 2 розділу

Порівнюючи комірки NAND Flash і NOR Flash, ми з'ясували, що ефективність NAND пояснюється меншою кількістю металевих контактів і значно меншим розміром комірок. Встановлено, що NAND Flash використовує сектори і підходить для зберігання послідовних даних, хоча для довільного доступу вимагає додаткової RAM і коду виправлення помилок (ECC) для

збереження цілісності даних. З'ясовано, що флеш-комірки NAND мають високу щільність і компактніший розмір, що робить їх ідеальними для споживчих пристроїв, таких як USB-накопичувачі та SD-карти.

Аналіз NAND і NOR Flash показав, що NAND забезпечує швидке стирання і запис, але має повільний довільний доступ. Водночас, NOR дозволяє швидкий довільний доступ і байтовий запис, проте повільніше стирається та записує. Виявлено, що у сучасних вбудованих системах дедалі частіше використовується NAND Flash завдяки її високій щільності, низькій вартості та швидкому виконанню програмування і стирання.

Досліджено, що NAND Flash вимагає меншої кількості апаратних контактів і підтримує більшу щільність пам'яті з однаковою апаратною конфігурацією, що забезпечує її перевагу для майбутніх розробок. Основні переваги NAND включають швидший час запису та стирання (понад 5 МБ/с для запису та 2 мс для стирання блоків), але основним недоліком є повільний довільний доступ. Водночас, ми з'ясували, що NOR Flash забезпечує виконання на місці (XiP) і є більш ефективною для програм з довільним доступом.

3 ФАЙЛОВА СТРУКТУРА NAND, ЯК ВОНО ПРАЦЮЄ, ЧАСОВІ ДІАГРАМИ

Флеш-пристрій 2 Гбіт NAND складається з 2048 блоків, по 64 сторінки на блок (див. рисунок 6). Кожна сторінка має 2112 байтів, які складаються з 2048-байтової області даних і 64-байтової резервної області. Резервна область зазвичай використовується для ЕСС, вирівнювання рівня зносу та інших функцій програмного забезпечення, хоча фізично вона така ж, як і решта сторінки. Флеш-пристрої NAND пропонуються з 8- або 16-бітним інтерфейсом. Дані хоста підключаються до флеш-пам'яті NAND через 8- або 16-бітну двонаправлену шину даних. Для 16-бітних пристроїв команди та адреси використовують молодші 8 бітів (7:0). Старші 8 біт 16-бітної шини даних використовуються лише під час циклів передачі даних.

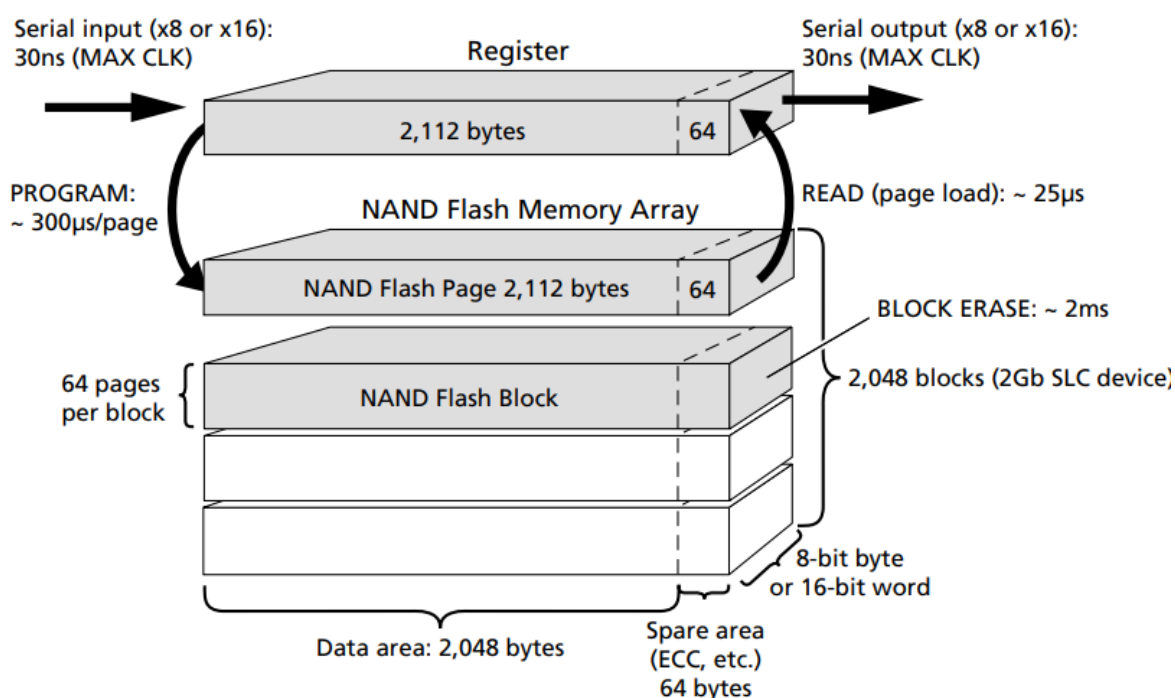


Рисунок 6 – Флеш-пристрій NAND на 2 Гбіт, організований у вигляді 2048 блоків

Для видалення блоку потрібно приблизно 2 мс. Після завантаження даних у реєстр програмування сторінки вимагає приблизно 300 мкс. Операція PAGE

READ вимагає приблизно 25 мкс, протягом яких здійснюється доступ до сторінки з масиву та завантажується в 16896-бітний (2112-байтовий) регістр. Після цього реєстр стає доступним для користувача, щоб вивести дані [7].

Окрім шини введення/виведення, інтерфейс флеш-пам'яті NAND складається з шести основних сигналів керування (див. Таблицю 4). (Примітка: символ «#» після сигналу вказує на те, що сигнал стверджується LOW.)

Таблиця 4 – Описи сигналів

Символ	Сигнал	Опис
ALE	Увімкнення фіксації адреси	Коли ALE має значення HIGH, адреси фіксуються в регістрі адреси флеш-пам'яті NAND за наростаючим фронтом сигналу WE#.
CE#	Включення мікросхеми	Якщо CE не встановлено, флеш-пристрій NAND залишається в режимі очікування та не реагує на жодні сигнали керування.
CLE	Увімкнення засувки команд	Коли CLE має значення HIGH, команди фіксуються в регістрі команд флеш-пам'яті NAND за наростаючим фронтом сигналу WE#.
R/B#	Готовий/зайнятий#	Якщо флеш-пристрій NAND зайнятий операцією ERASE, PROGRAM або READ, сигнал R/B# встановлюється на НИЗЬКОМУ. Сигнал R/B# є відкритим стоком і потребує підтягуючого резистора.
RE#	Увімкнути читання	RE# вмикає вихідні буфери даних.
WE#	Увімкнути запис	WE# відповідає за синхронізацію даних, адрес або команд у флеш-пристрій NAND.

Дані зміщуються в регістр NAND Flash або з нього по 8 або 16 біт за раз. Під час операції PRO-GRAM дані, які потрібно запрограмувати, синхронізуються в регістрі даних по передньому фронту WE#. Спеціальні команди

використовуються для довільного доступу до даних або переміщення даних у межах реєстру, щоб зробити довільний доступ можливим.

Дані виводяться з реєстра даних подібним чином за допомогою сигналу дозволу читання (RE#), який відповідає за виведення поточних даних і збільшення до наступного розташування. Годинники WE# і RE# можуть працювати до 30 нс на передачу. Коли RE# або chip enable (CE#) не встановлені на LOW, вихідні буфери мають три стани. Ця комбінація CE# і RE# активує вихідні буфери, дозволяючи NAND Flash спільно використовувати шину даних з іншими типами пам'яті, такими як NOR Flash, SRAM або DRAM. Цю функцію іноді називають «увімкнути чіп, байдуже». Основна мета цього посилання полягає в тому, щоб відрізнити дуже старі флеш-пристрої NAND, які вимагають підтвердження CE# протягом усього циклу [7].

Усі операції флеш-пам'яті NAND запускаються шляхом виконання командного циклу. Це досягається шляхом розміщення команди на I/O[7:0], управління CE# LOW і CLE HIGH, а потім видача WE# годинника. Команди, адреси та дані синхронізуються на флеш-пристрій NAND за переднім фронтом сигналу WE# (див. Малюнок 5 і Таблицю 5 на сторінці 8).

Більшість команд вимагає певної кількості циклів адресування, за якими слід другий цикл команди. За винятком команд RESET і READ STATUS, нові команди не слід видавати, коли пристрій зайнятий (див. Рисунок 7 і Таблицю 5).

Схема адресації для флеш-пристрою NAND ємністю 2 Гбіт наведена в таблиці 6. Перший і другий цикли адреси вказують адресу стовпця, який визначає початковий байт на сторінці. Розташування останнього стовпця – «2,112», тож адреса цього останнього розташування дорівнює 08h (у другому байті) і 3Fh (у першому байті). PA[5:0] визначає адресу сторінки в блоці, а BA[16:6] визначає адресу блоку. У той час як повна 5-байтова адреса потрібна для більшості операцій PROGRAM і READ, тільки перший і другий байти потрібні для операцій, які отримують довільний доступ до даних на сторінці. Операція BLOCK ERASE вимагає лише трьох старших байтів (третій, четвертий і п'ятий) для вибору блоку [8].

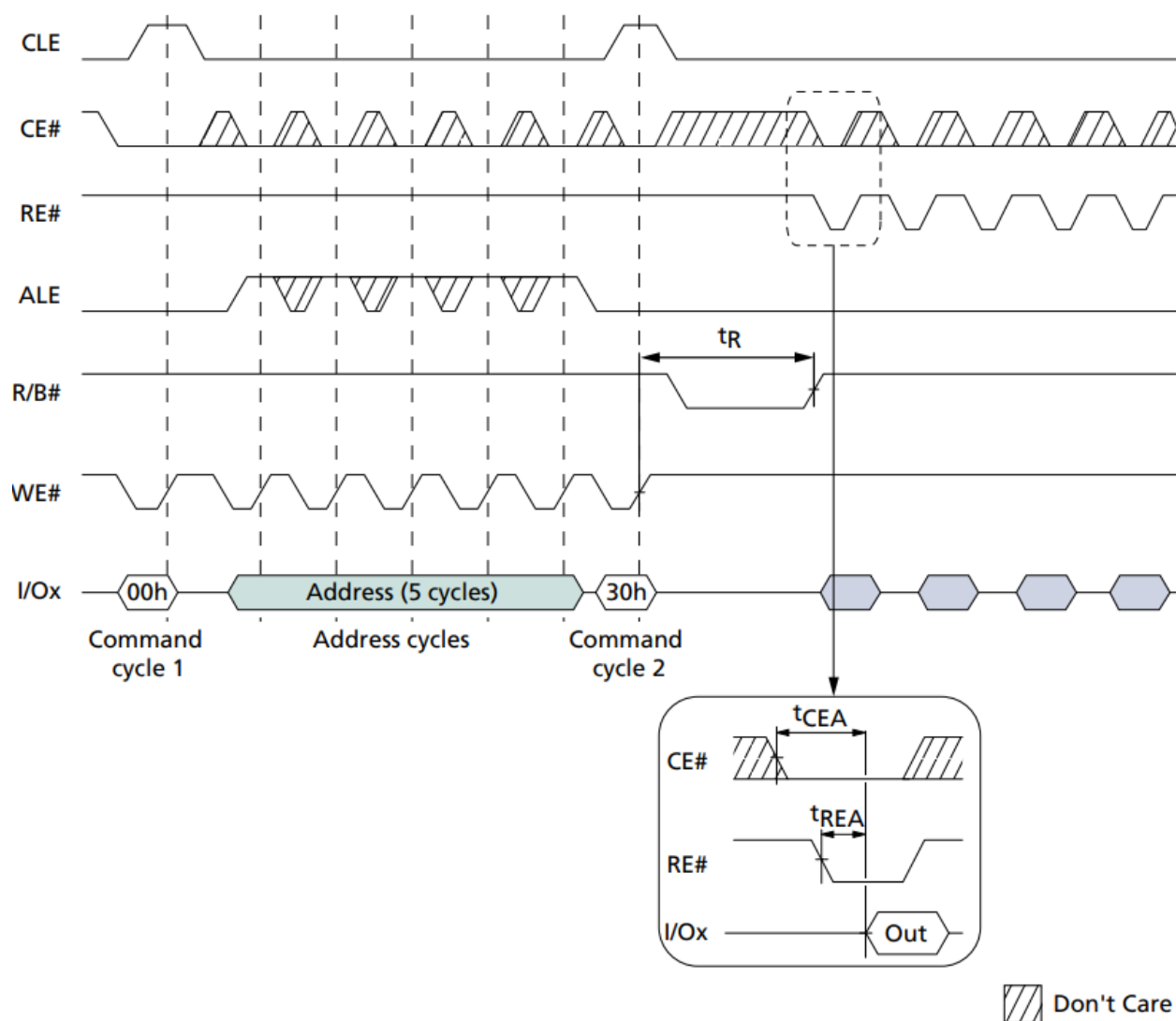


Рисунок 7 – Цикли команд для операцій флеш-пам'яті NAND

Таблиця 5 – Цикли команд і цикли адрес

Команда	Цикл 1	Кількість адресних циклів	Цикли даних	Цикл 2	Діє під час зайнятості
READ PAGE	00h	5	Немає	30h	Немає
PAGE READ CACHE MODE START	31h	—	Немає	—	Немає
PAGE READ CACHE MODE START LAST	3Fh	—	Немає	—	Немає

Продовження таблиці 5. Цикли команд і цикли адрес

Команда	Цикл 1	Кількість адресних циклів	Цикли даних	Цикл 2	Діє під час зайнятості
READ for INTERNAL DATA MOVE	00h	5	Немає	35h	Немає
RANDOM DATA READ	05h	2	Немає	E0h	Немає
READ ID	90h	1	Немає	–	Немає
READ STATUS	70h	–	Немає	–	Так
PROGRAM PAGE	80h	5	Так	10h	Немає
PROGRAM PAGE CACHE MODE	80h	5	Так	15h	Немає
PROGRAM for INTERNAL DATA MOVE	85h	5	Додатково	10h	Немає
RANDOM DATA INPUT	85h	2	Так	–	Немає
BLOCK ERASE	60h	3	Немає	D0h	Немає
RESET	FFh	–	Немає	–	Так

Таблиця 6 – Схема адресації флеш-пам'яті 2 Гбіт SLC NAND

Цикл	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Перший	CA7	CA6	CA5	CA4	CA3	CA2	CA1	CA0
Другий	LOW	LOW	LOW	LOW	CA11	CA10	CA9	CA8
Третій	BA7	BA6	PA5	PA4	PA3	PA2	PA1	PA0
Четвертий	BA15	BA14	BA13	BA12	BA11	BA10	BA9	BA8
П'ятий	LOW	LOW	LOW	LOW	LOW	LOW	LOW	BA16

Адреса блоку, об'єднана з адресою сторінки = фактична адреса сторінки.
 CAx = адреса стовпця; PAx = адреса сторінки; BAx = адреса блоку. Якщо CA11 = «1», тоді CA[10:6] має бути «0» [13].

Флеш-команди NAND

Для кожної команди CE# має бути LOW, CLE та ALE повинні бути підтверджені, а також мають бути надані тактові частоти запису (WE#). Поки

пристрій зайнятий, можна видати лише дві команди: RESET і READ STATUS [13].

Операція RESET (СКИДАННЯ).

Найпростішою командою NAND Flash є команда RESET (FFh) (див. Рисунок 8). Команда RESET не потребує адреси чи наступних циклів (див. Таблицю 5 на сторінці 24). Просто встановіть CLE і видайте імпульс запису з FFh на шині даних, і буде виконана операція RESET [13].

RESET — це одна з двох команд, які можна виконати, коли флеш-пристрій NAND зайнятий. Якщо пристрій зайнятий обробкою попередньої команди, виконання команди RESET перериває попередню операцію. Якщо попередньою операцією була команда ERASE або PROGRAM, виконання команди RESET призведе до передчасного переривання виконання команди, і бажана операція не завершиться. ERASE та PROGRAM можуть бути трудомісткими операціями; видача команди RESET дає змогу перервати будь-яку з них і повторно виконати команду пізніше.

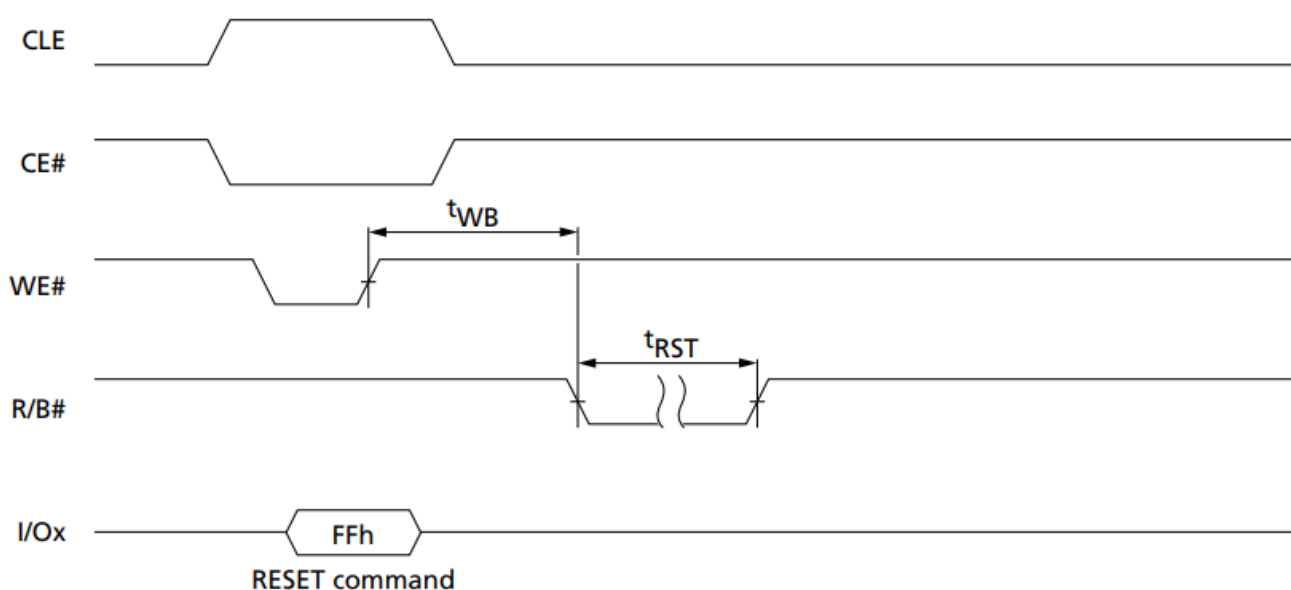


Рисунок 8 – Команда RESET

Операція READ ID

Команда READ ID (90h) потребує одного циклу фіктивної адреси (00h), але не потребує другого циклу команди (див. Таблицю 5). Після введення команди та фіктивної адреси ідентифікаційні дані можна зчитувати, утримуючи CLE та ALE LOW і перемикаючи сигнал RE# для кожного байта ID. На рисунку 9 показаний час операції READ ID, а в таблиці 7 показаний формат 4-байтової відповіді [8].

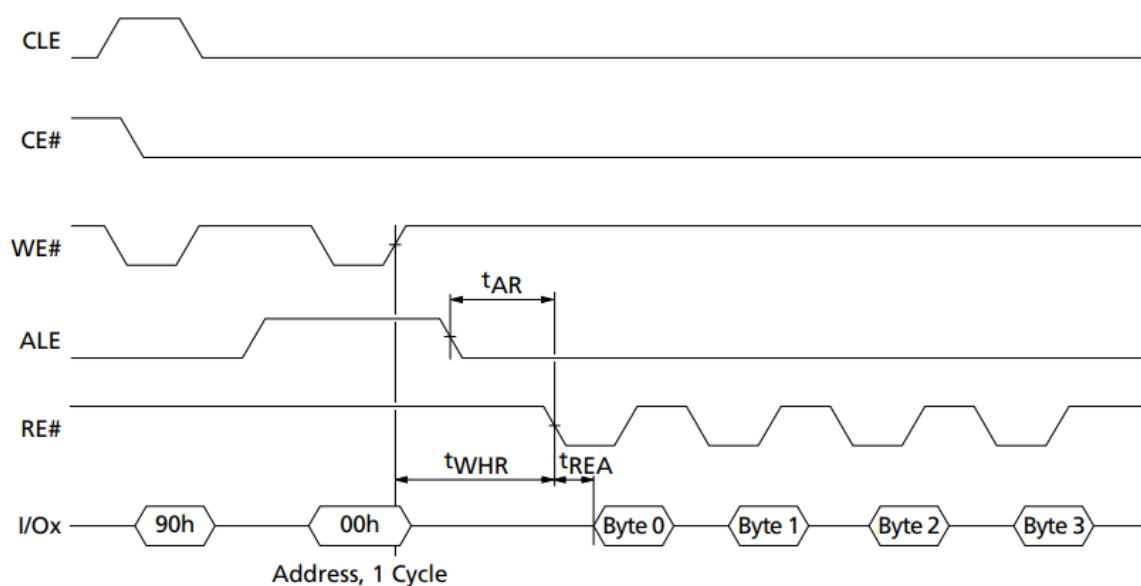


Рисунок 9 – Команда READ ID

READ STATUS Операція

READ STATUS (70h) — це друга команда, яка може бути видана, коли флеш-пристрій NAND зайнятий. Ця команда не вимагає адреси або другого командного циклу. Стан флеш-пристрою NAND можна контролювати, видавши тактовий сигнал RE# після команди READ STATUS. Якщо команда READ STATUS використовується для моніторингу стану готовності пристрою, команда має бути видана лише один раз, і статус можна перечитати, повторно видавши годинник RE#. Крім того, сигнал RE# можна підтримувати на LOW, очікуючи

отримання відповідного біта стану перед продовженням. READ STATUS також повідомляє про стан сигналу захисту від запису та стан проходження/невиконання попередніх операцій PROGRAM або ERASE. Для забезпечення належної цілісності даних обов'язково потрібно отримати статус «пропущено» під час операцій PROGRAM або ERASE [9].

Таблиця 7 – READ ID відповіді

	Варіант	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0	Значення1
Байт 0	ID виробника									
	мікрон	0	0	1	0	1	1	0	0	2Ch
Байт 1	ID пристрою									
MT29F2G08AA C	2Gb, x8, 3V	1	1	0	1	1	0	1	0	DAh
MT29F2G08AB C	2Gb, x8, 1.8V	1	0	1	0	1	0	1	0	AAh
MT29F2G16AA C	2Gb, x16, 3V	1	1	0	0	1	0	1	0	CAh
MT29F2G16AB C	2Gb, x16, 1.8V	1	0	1	1	1	0	1	0	BAh
Байт 2										
Значення байта	Don't Care	x	x	x	x	x	x	x	x	XXh
Байт 3										
Розмір сторінки	2 Kб							0	1	01b
Розмір резервної області (байти)	64					0	1			01b
Розмір блоку (без запасного)	128 Kб			0	1					01b
організація	x8		0							0b
	x16		1							1b
Зарезервований		0								0b
Значення байта	x8	0	0	0	1	0	1	0	1	15 год
	x16	0	1	0	1	0	1	0	1	55 год

Таблиця 8 – READ STATUS відповіді

Біт SR	Сторінка програми	Режим кешу сторінки програми	Читання сторінки	Режим читання кешу сторінки	Стирання блоку	Визначення
0	Успіх/провал	Успіх/провал (N)	–	–	Здав/не склав	«0» = Успішне PROGRAM / ERASE «1» = Помилка в PROGRAM / ERASE
1	–	Успіх/провал (N - 1)	–	–	–	«0» = Успішне PROGRAM / ERASE «1» = Помилка в PROGRAM / ERASE
2	–	–	–	–	–	«0»
3	–	–	–	–	–	«0»
4	–	–	–	–	–	«0»
5	Готовий/зайнятий	Готовий/зайнятий	Готовий/зайнятий	Готовий/зайнятий	Готовий/зайнятий	«0» = зайнято «1» = готово
6	Готовий/зайнятий	Готовий/зайнятий кеш	Готовий/зайнятий	Готовий/зайнятий кеш	Готовий/зайнятий	«0» = зайнято «1» = готово
7	Захист від запису	Захист від запису	Захист від запису	Захист від запису	Захист від запису	«0» = Захищено «1» = Не захищено
[15:8]	–	–	–	–	–	«0»
1	–	Здав/не склав (N-1)	–	–	–	«0» = Успішне PROGRAM / ERASE «1» = Помилка в PROGRAM / ERASE

Операція ERASE

Операція BLOCK ERASE (60h) стирає весь блок із 64 сторінок, або загалом 128 КБ. Щоб виконати операцію BLOCK ERASE, використовуйте сигнал WE# для синхронізації команди ERASE (60h) із затвердженням CLE. Далі зафіксуйте три цикли адреси, зберігаючи ALE встановленим для кожного байта адреси. Ці три адресні цикли є найважливішими адресними циклами (показано в таблиці 6) і включають адресу блоку та адресу сторінки. Частина адреси сторінки (шість молодших бітів третього циклу адреси) ігнорується, і використовується лише частина адреси блоку з трьох старших байтів. Після повного введення адреси видайте другу команду (командний цикл 2) D0h, яка синхронізується з WE#, поки CLE затверджується. Це підтверджує операцію ERASE, і пристрій стає зайнятим приблизно на 2 мс. Коли пристрій завершує цю операцію, він готовий до

наступної команди. Команда READ STATUS може бути видана в будь-який час, навіть якщо пристрій зайнятий під час операції ERASE. Мікропроцесор або контролер може контролювати пристрій за допомогою команди READ STATUS. На рисунку 10 часова діаграма команди ERASE.

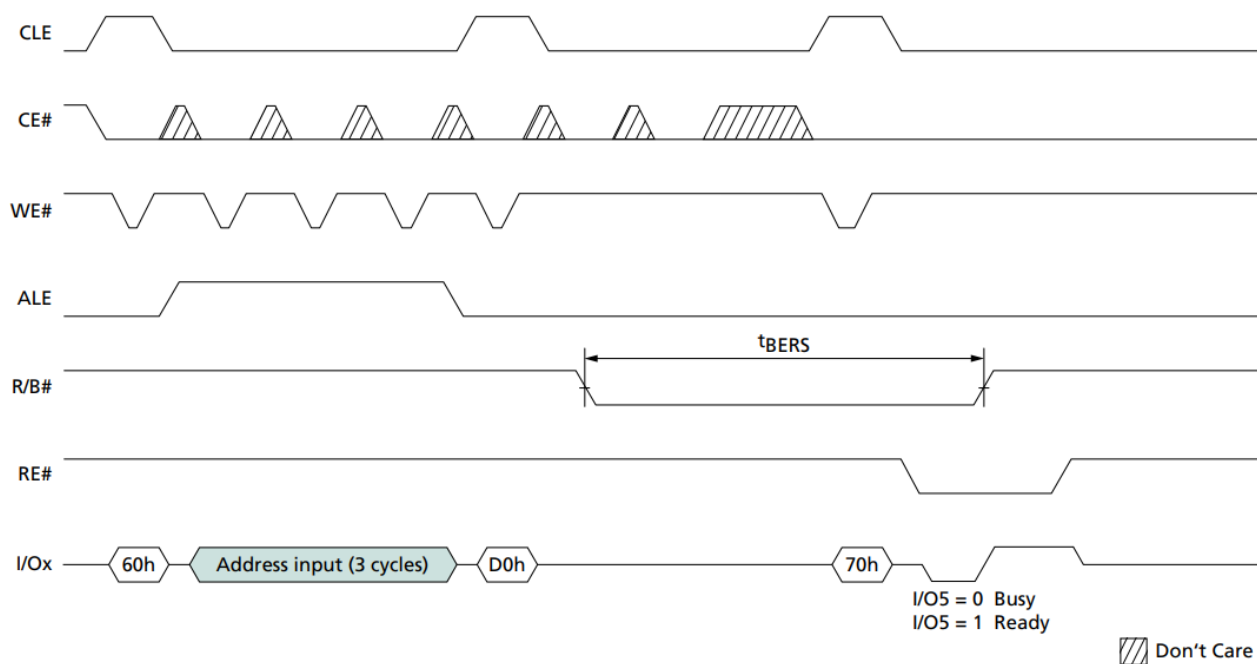


Рисунок 10 – Команда ERASE

Операції PROGRAM

Операції PROGRAM можуть програмувати біти лише на «0» і пропускати, що користувач почав із раніше стертого блоку.

Якщо користувач не бажає програмувати біт (або групу бітів), біти можна зберегти в стертому стані, встановивши для цього конкретного біта/групи значення «1».

Коли отримано команду PROGRAM PAGE (80h), вхідний регістр скидається (всередині) на всі «1s». Це підтримує введення лише байтів даних, які мають бути запрограмовані з «0» бітами. Операція PROGRAM починається з команди 80h (із заявленим CLE — див. Рисунок 11). Далі відпустіть CLE та підтвердьте ALE, щоб

ввести повні п'ять циклів адреси. Після введення команди та адреси дані вносяться до реєстру. Після введення всіх даних видається команда 10h для підтвердження попередньої команди та початку операції програмування.

Операція PROGRAM зазвичай вимагає 300 мкс, хоча може знадобитися до 700 мкс. Обов'язково, щоб користувач прочитав статус і перевірів успішність роботи. Якщо операція не вдалася, блок слід зареєструвати як пошкоджений блок і не використовувати в майбутньому. Всі дані потрібно перемістити в хороший блок [8].

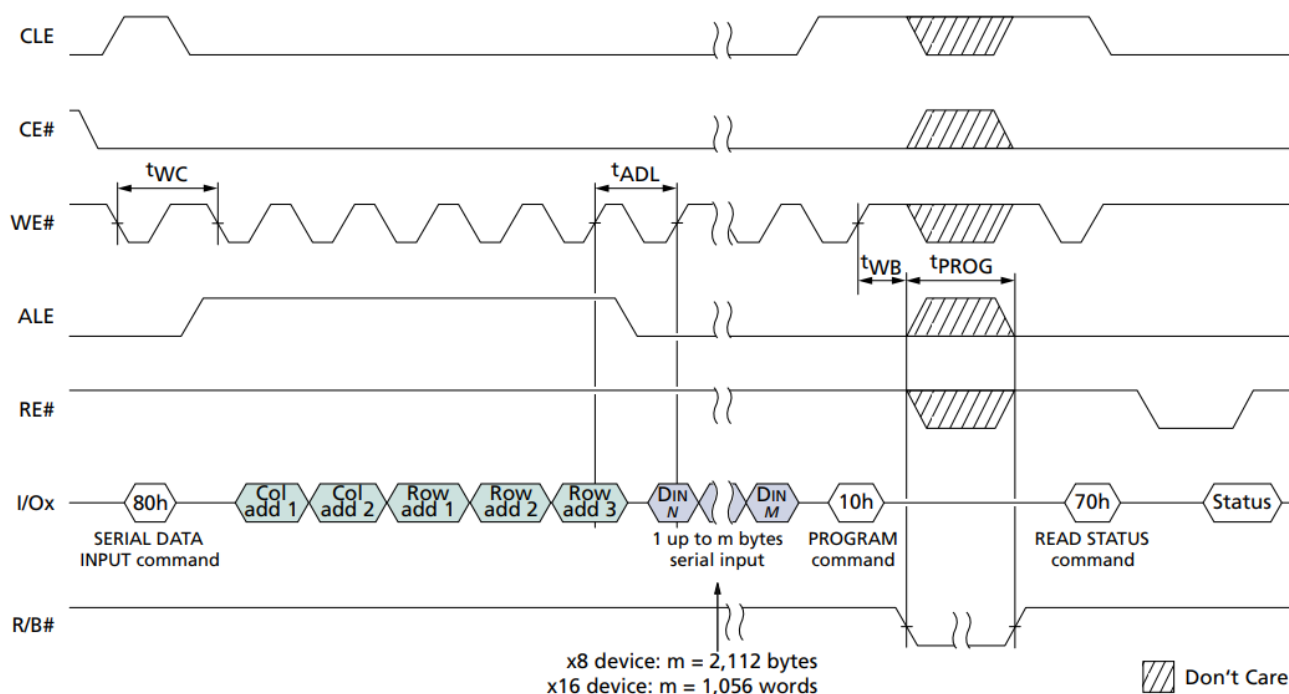


Рисунок 11 – Команда PROGRAM

Операція RANDOM DATA INPUT (ВИПАДКОВОГО ВВЕДЕННЯ ДАНИХ)

Наступною командою є команда ВИПАДКОВОГО ВВЕДЕННЯ ДАНИХ (85h). Як показано на рисунку 12, для цієї команди потрібно лише 2 байти адреси, за якою слідують дані. Ця команда корисна для довільного доступу до даних на сторінці, наприклад, для доступу до даних ECC. RANDOM DATA INPUT можна

використовувати для переходу в кінець сторінки та запису даних ECC. Користувач може ввести скільки завгодно комбінацій адрес і даних. Лише після введення команди 10h дані фактично запрограмовані на вибрану сторінку.

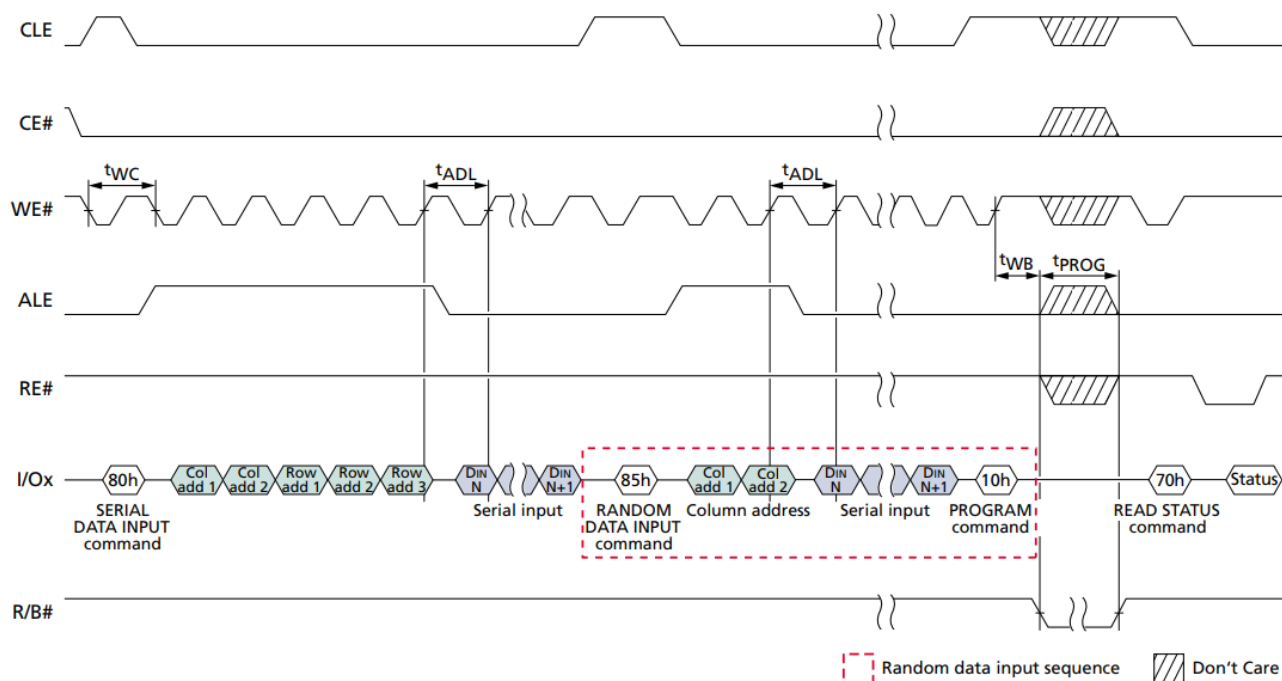


Рисунок 12 – Команда PROGRAM із довільним введенням даних

Завдяки великому розміру флеш-сторінок NAND програмування часткової сторінки корисне для зберігання менших обсягів даних. Кожна сторінка NAND може вмістити чотири PC-sized 512-байтні сектори. Вільна область кожної сторінки забезпечує додаткове зберігання для ECC та іншої програмної інформації.

Хоча вигідно записати всі чотири сектори одночасно, часто це неможливо. Наприклад, коли дані додаються до файлу, файл може спочатку мати 512 байт, а потім збільшитися до 1024 байт. У цій ситуації для запису других 512 байтів на флеш-пристрій NAND потрібна друга операція PROGRAM PAGE. Максимальна кількість разів, коли часткова сторінка може бути запрограмована до того, як буде потрібно стерти, становить вісім разів; це вміщує чотири сектори даних і чотири сектори ECC, кожен з яких програмується окремо.

Методи зберігання

Два поширених методи зберігання даних і резервної інформації на одній сторінці показані на рисунках 13 та 14. Перший метод показує область даних розміром 512 байт плюс 16-байтову резервну область, що безпосередньо примикає до неї; 528 байт для об'єднаних областей. Сторінка розміром 2112 байт може містити чотири з цих 528-байтових елементів. Друга реалізація передбачає роздільне зберігання даних і резервної інформації. Чотири 512-байтові області даних зберігаються першими, а їхні відповідні 16-байтові резервні області йдуть по порядку в кінці сторінки [9].

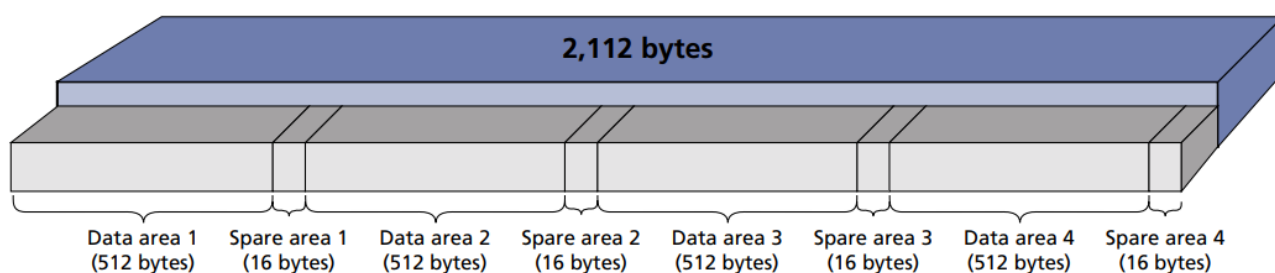


Рисунок 13 – Типові методи зберігання. Суміжні дані та резервні області

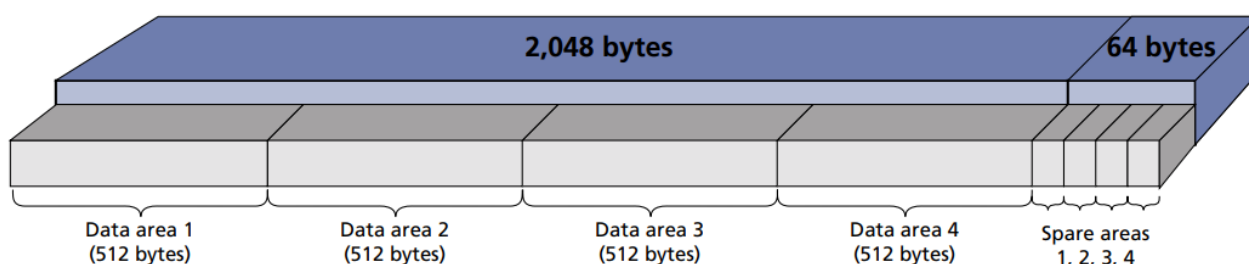


Рисунок 14 – Типові методи зберігання. Окремі дані та резервні області

Операція READ

Операція READ починається з команди 00h, за якою слідує п'ять циклів адреси, потім команда 30h для підтвердження послідовності команд (див.

Рисунок 15). Після закінчення часу передачі READ ('R) приблизно 25 мкс минуло, дані завантажуються в регістр і готові до виведення. Видача годинника з можливістю читання (RE#) дозволяє флеш-пристрою NAND виводити перший байт даних, що відповідає адресі стовпця, зазначеному в адресі. Наступні переходи RE# виводять послідовні розташування. Коли сигнал RE# є HIGH (не встановлено), лінії вводу/виводу мають три стани. Читання після закінчення пристрою (байт 2112 або слово 1056) призводить до недійсних даних. На рисунку 15 також зображено RANDOM DATA READ операцію [8].

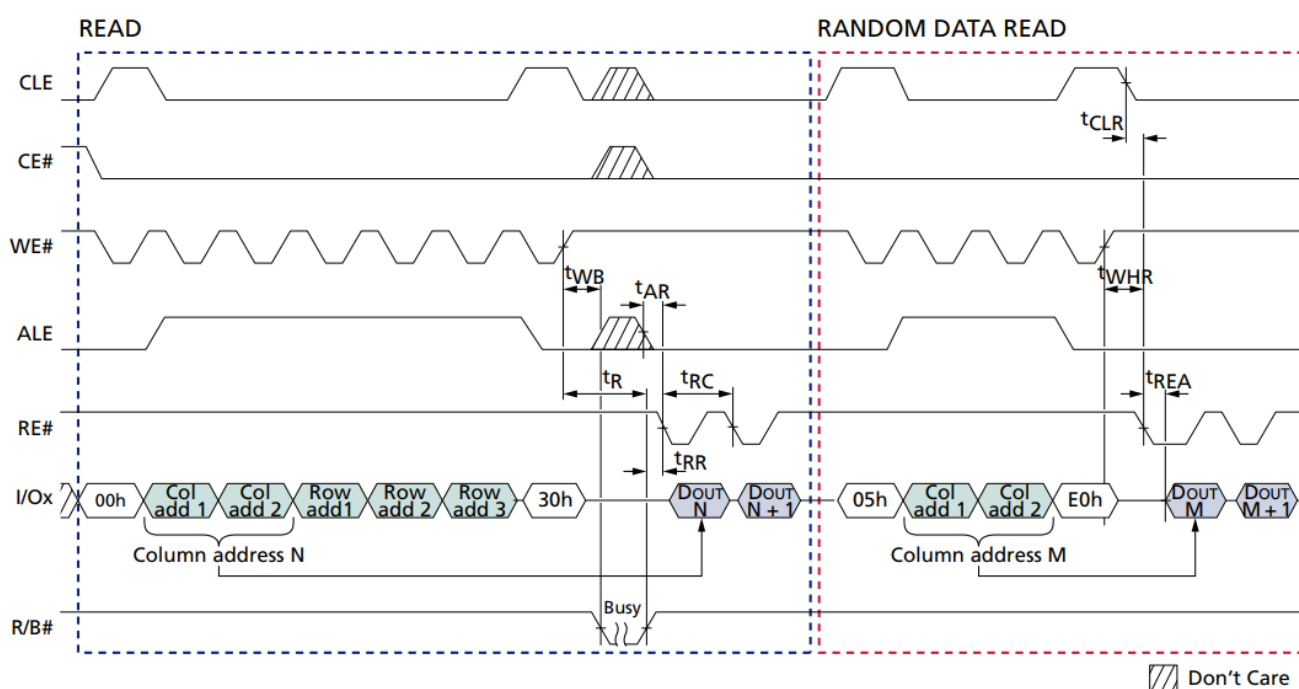


Рисунок 15 – Операції READ і RANDOM DATA READ

Операція RANDOM DATA READ

Користувач може отримати прямий доступ до випадкових даних, виконавши команду 05h, два цикли адреси та цикл підтвердження E0h (див. Рисунок 15). Коли сторінку було прочитано з масиву, ця команда забезпечує швидкий доступ до даних.

Команда PAGE READ CACHE MODE

Для зручності обговорення до цього моменту обговорювався лише один регістр у флеш-пристрої NAND. Флеш-пристрій NAND насправді має два регістри, регістр даних і регістр кешу, як показано на рисунку 16. Атрибути цих двох регістрів відіграють важливу роль у різних режимах кешування флеш-пам'яті NAND.

Команда PAGE READ CACHE MODE дозволяє користувачеві конвеєрно здійснювати наступний послідовний доступ до масиву, виводячи дані, до яких раніше зверталися. Ця техніка подвійної буферизації дає змогу приховати час передачі READ (t_R). Дані спочатку передаються з масиву NAND Flash в регістр даних. Якщо регістр кешу доступний (не зайнятий), дані швидко переміщуються з регістра даних у регістр кешу. Після того, як дані були передані в регістр кешу, регістр даних доступний і може почати завантажувати наступну послідовну сторінку з масиву NAND Flash.

Використання команди PAGE READ CACHE MODE забезпечує підвищення продуктивності на 33 відсотки порівняно з традиційною командою PAGE READ на 8-розрядному пристрої введення-виведення з пропускною здатністю до 31 МБ/с. На 16-розрядних пристроях вводу/виводу пропускну здатність можна збільшити до 37 Мбайт/с, що забезпечує підвищення продуктивності на 40 відсотків порівняно зі звичайними операціями PAGE READ. Дивіться рисунок 17 для порівняння.

Технічні примітки на www.micron.com/products/nand/technotes надайте додаткові відомості про режими кешу та про те, як їх можна використовувати для покращення продуктивності. PAGE READ CACHE MODE може бути особливо корисним під час завантаження системи, коли з флеш-пристрою NAND зазвичай зчитуються великі обсяги даних і час запуску є критичним.

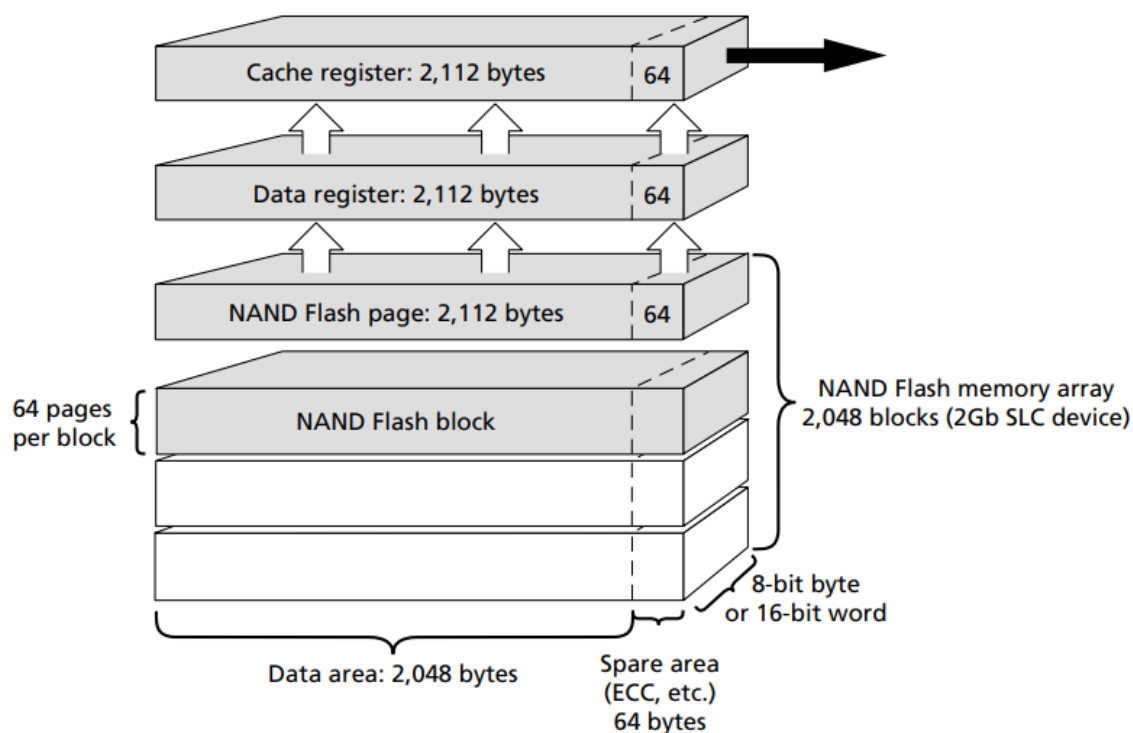
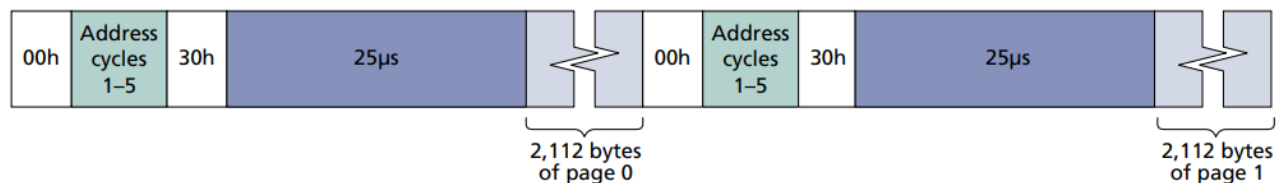


Рисунок 16 – PAGE READ CACHE MODE

Page Read Example



Page Read Cache Mode Example

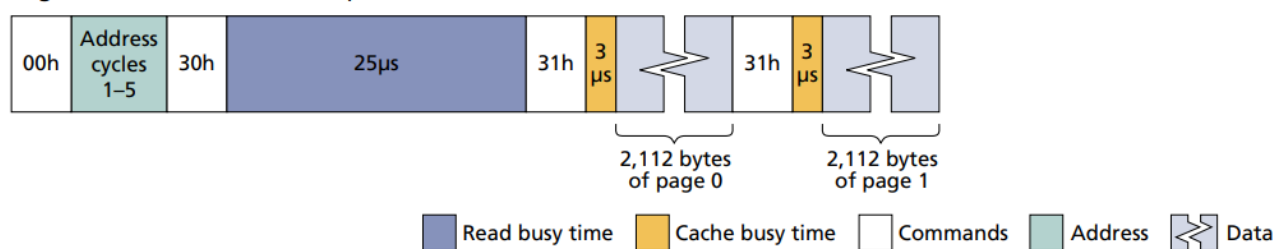


Рисунок 17 – Час PAGE READ CACHE MODE в порівнянні з PAGE READ

Команда PROGRAM PAGE CACHE MODE

PROGRAM PAGE CACHE MODE забезпечує покращення продуктивності порівняно зі звичайними операціями PROGRAM PAGE (див. рисунки 18 і 19).

PROGRAM PAGE CACHE MODE — це метод подвійної буферизації, який дозволяє контролеру вводити дані безпосередньо в регістр кешу та використовує регістр даних як область зберігання для надання даних для програмування масиву. Це звільняє кеш-реєстр, щоб наступну послідовну операцію сторінки можна було завантажити паралельно. У багатьох програмах час програмування (tPROG) можна повністю приховати. Як і у випадку з командою PAGE READ CACHE MODE, регістр даних використовується для підтримки пропускну здатності даних протягом усього циклу програмування. Це звільняє кеш-реєстр для отримання наступної сторінки даних від контролера.

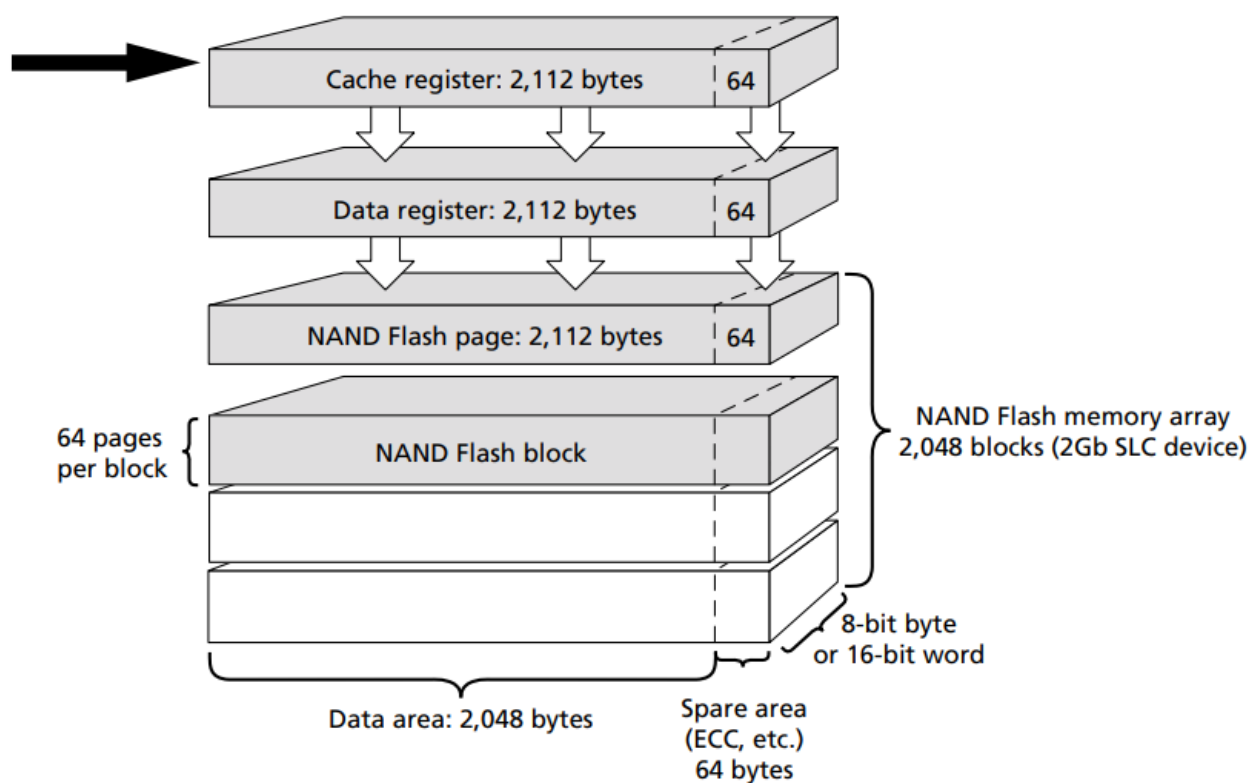


Рисунок 18 – PROGRAM PAGE CACHE MODE

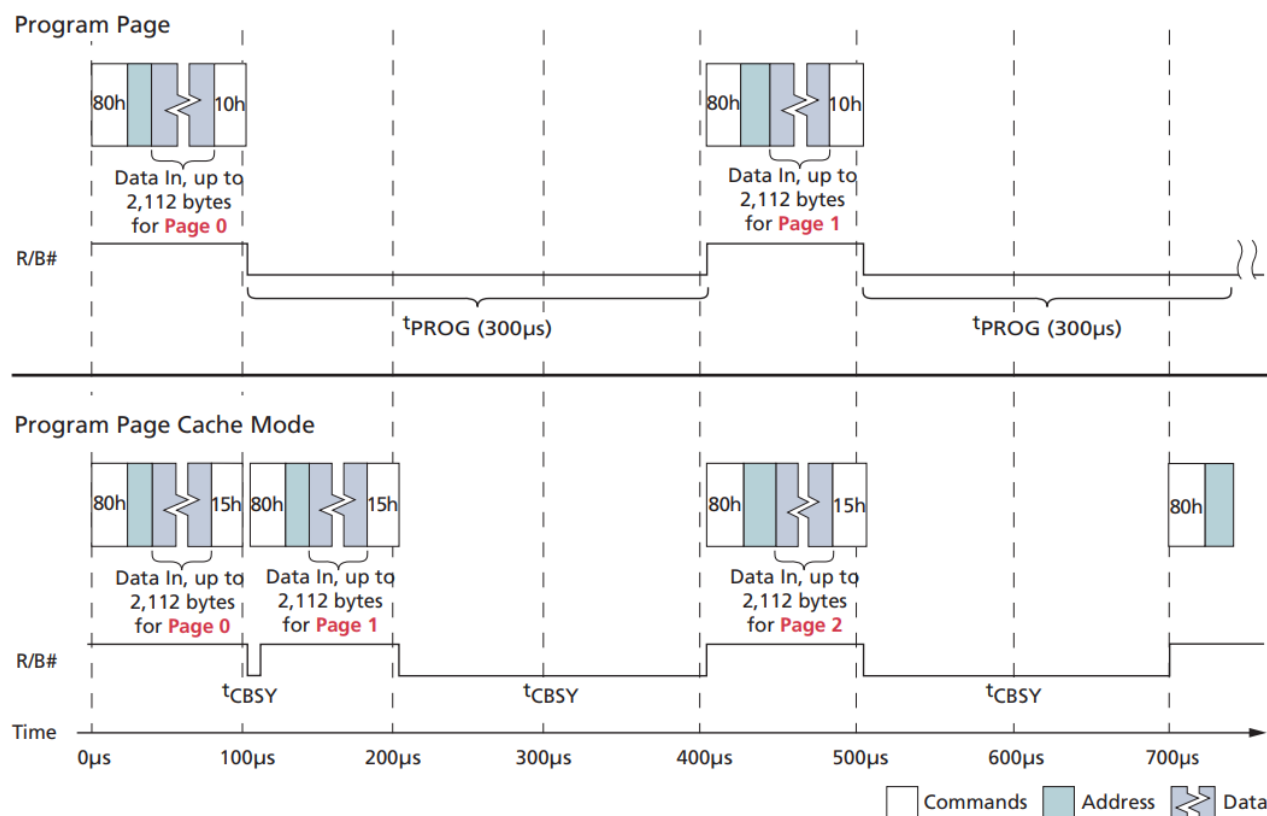


Рисунок 19 – Час PROGRAM PAGE CACHE MODE у порівнянні з PROGRAM PAGE

Висновки до 3 розділу

Проаналізувавши різні аспекти роботи з флеш-пам'яттю NAND, ми зрозуміли, що ключові команди включають RESET, READ ID, ERASE, READ STATUS і RANDOM DATA INPUT. Дослідження показують, що кожен з цих процесів супроводжується детальним описом ілюстрацій та таблиць, які пояснюють часові діаграми і відповідні формати відповідей.

Вивчення команди RESET виявило, що цей процес скидає пристрій до початкового стану, підготувавши флеш-пам'ять до подальших операцій. Операція READ ID дозволяє зчитувати ідентифікаційні дані флеш-пам'яті, що є необхідним для визначення параметрів і характеристик пристрою.

Процес ERASE був досліджений і описує стирання блоку даних з флеш-пам'яті, яке включає кілька циклів адрес і підтвердження операції, після чого

пристрій стає готовим до наступних команд. У таблиці 8 наведені біти статусу, які відображають стан пристрою під час операцій програмування та стирання.

Встановлено, що команда RANDOM DATA INPUT дозволяє здійснювати довільний доступ до даних на сторінці, що корисно для введення даних ECC та інших програмних даних. Це особливо важливо для програмування часткових сторінок, що дозволяє оптимізувати зберігання і обробку даних у флеш-пам'яті.

Таким чином, ми зрозуміли, що даний розділ детально описує ключові команди і операції, необхідні для ефективної роботи з флеш-пам'яттю NAND, забезпечуючи чітке розуміння процесів програмування, стирання і зчитування даних.

4 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ. ІНТЕРФЕЙС.

Вибір процесора або контролера з вбудованим інтерфейсом NAND Flash має значні переваги. Якщо це неможливо, можна розробити інтерфейс без клею між флеш-пристроєм NAND і майже будь-яким процесором.

Для огляду основна відмінність між NAND Flash і NOR Flash полягає в мультиплексній шині, яка використовується для передачі команд, адрес і даних у пристрої NAND Flash. Використовуючи керуючі сигнали CLE та ALE, можна вибрати команду, адресу або цикл даних. CLE використовується для визначення циклів команд; ALE використовується для визначення циклів адреси [8].

Підключення ALE до біта адреси 5 процесора та CLE до біта адреси 4 процесора дає змогу вибирати команду, адресу або дані, просто змінюючи адресу, яку виводить процесор (див. Таблицю 9); CLE та ALE автоматично затверджуються у відповідний час [8].

Таблиця 9 – Увімкнення команди, адреси або вибору даних

A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----

A5	A4		
ALE	CLE	Зсув адреси пам'яті	Вибрано регістр NAND
0	0	0xh	Регістр даних
0	1	1xh	Регістр команд
1	0	2xh	Адресний регістр
1	1	3xh	Не визначено (не використовувати)

Щоб видати команду, процесор виводить заплановану команду на шину даних і на вихідну адресу 0010h.

Щоб видати будь-яку кількість циклів адреси, процесор просто виводить призначену послідовність адрес флеш-пам'яті NAND на адресу процесора 0020h. За допомогою цієї методики користувач може отримати доступ до команд, адрес і даних безпосередньо з процесора без будь-якої логіки з'єднання. У цьому

випадку ECC має оброблятися програмним забезпеченням. На рисунках 20–22, показано блок-схему, низькорівневий псевдокод і синхронізацію операцій PROGRAM. Багато процесорів мають можливість вказати кілька параметрів синхронізації навколо сигналу запису процесора, що є критичним для правильного налаштування та синхронізації утримання [8].

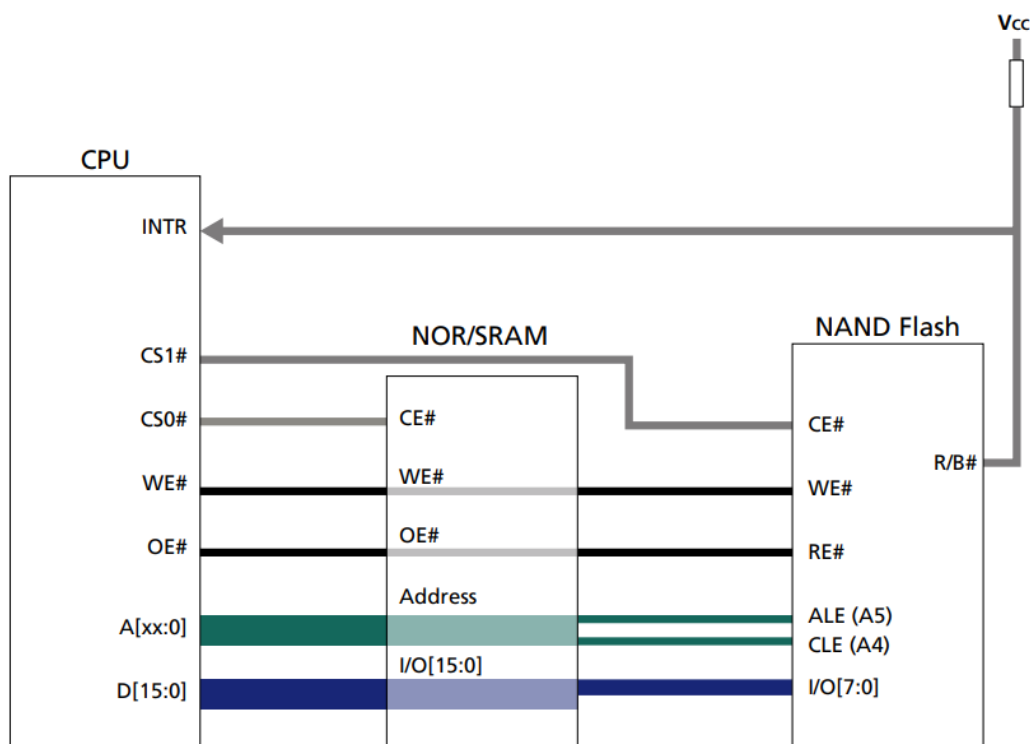


Рисунок 20 – Безклееве з'єднання NAND

```
(All numbers in HEX)
80 -> FFF010; CMD = 80
ColL -> FFF020; low column
ColH -> FFF020; high column
RowL -> FFF020; low ROW
RowM -> FFF020; Mid ROW
RowH -> FFF020; High ROW
D0 -> FFF000; Data 0
D1 -> FFF000; Data 1

(Complete remaining data)

D2111 -> FFF000 ; Data 2111
10 -> FFF010 ; CMD = 10

LOOP1:
PA -> Acc; Read status
BIT #6 set;
JMP NZ LOOP1; Jmp if Busy to Loop

; DONE !
```

Рисунок 21 – Приклад низькорівневого псевдокоду для операцій PROGRAM

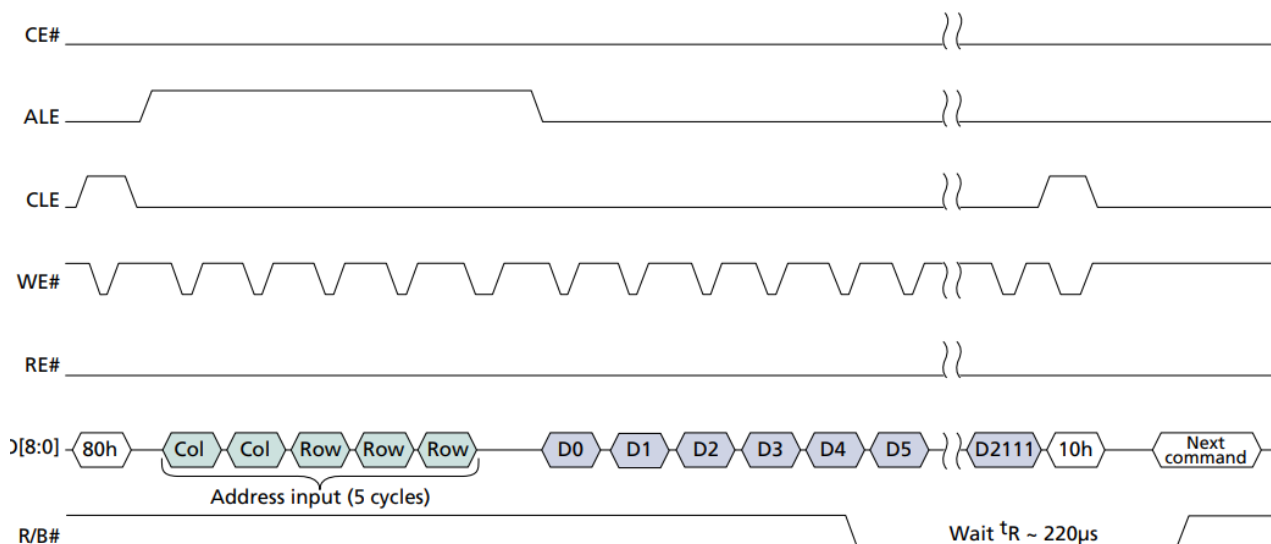


Рисунок 22 – Хронометраж виконання PROGRAM

Серед процесорів із вбудованим контролером NAND Flash — процесори Freescale™ i.MX21 і i.MX31, а також кілька процесорів OMAP™ від Texas Instruments.

На рисунку 23 (зображення надано Freescale Semiconductor, Inc) показано вбудований інтерфейс NAND Flash на процесорі Freescale i.MX21. Інтерфейс флеш-пам'яті NAND розташований у правій частині діаграми та під'єднується безпосередньо до флеш-пам'яті NAND. Ця реалізація підтримує автоматичне завантаження з пристрою NAND Flash, а також логіку ECC і буфер SRAM. Порт SRAM дозволяє виконувати код безпосередньо з буфера [13].

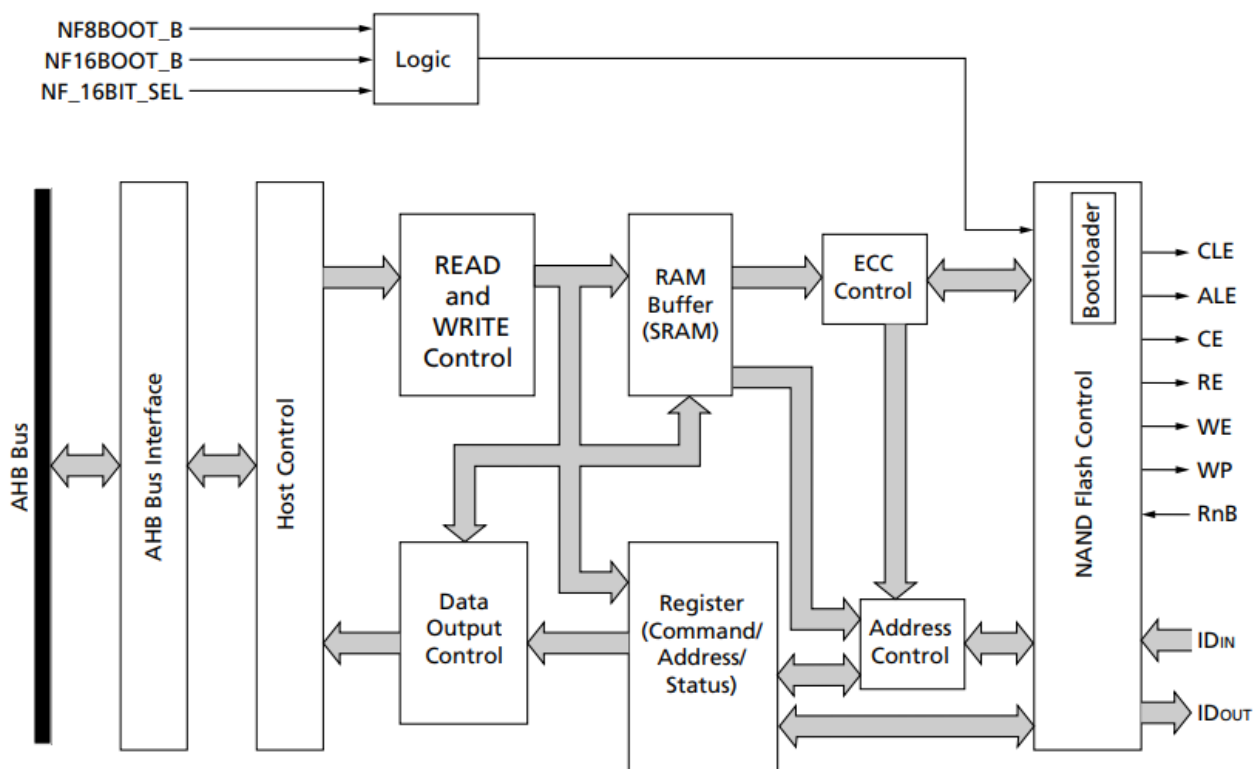


Рисунок 23 – Вбудований інтерфейс NAND Flash на процесорі Freescale i.MX21

Багаторівнева комірка (MLC)

Пристрої MLC використовують особливий тип комірки, який зберігає 2 біти на комірку, у порівнянні з традиційними пристроями з однорівневою коміркою (SLC), які можуть зберігати лише 1 біт на комірку. Технологія MLC пропонує очевидні переваги щільності. Однак MLC не має такої швидкості та надійності, як аналог SLC (див. таблицю 10). З цієї причини пристрої SLC використовуються в більшості програм високопродуктивних медіа-карт і бездротових процесорів; Пристрої MLC зазвичай використовуються в споживчих та інших недорогих продуктах.

Таблиця 10 – MLC проти SLC

Символ		MLC NAND 3,3 V (x16/x8)			SLC NAND 3,3 V (x16/x8)			Одиниці
		Min	Typ	Max	Min	Typ	Max	
'PROG	Час для передачі вмісту регістра даних у масив NAND Flash	–	600	1200	–	300	700	мкс
NOP	Кількість програм для часткових сторінок, дозволених на сторінці, перш ніж потрібно стерти	–	1	1	–	–	8	Цикли
'R	Час для передачі вмісту однієї сторінки в масиві NAND Flash до регістру даних	50	–	50	25	–	25	мкс
Витривалість з ECC і недійсним маркуванням блоку		10 тис	–	–	100 тис	–	–	PROGRAM / ERASE цикли
Потрібен MIN ECC		4	–	–	1	–	–	Виправні біти на 512 байт
NvB	4 ГБ MLC, 4 ГБ SLC	3,936	–	4,096	4,016	–	4,096	Блоки

Код виправлення помилок (ECC)

Як згадувалося раніше, NAND Flash потребує ECC для забезпечення цілісності даних. ECC використовується протягом багатьох років у модулях оперативної пам'яті, а також у багатьох інших типах накопичувачів. ECC можна використовувати в будь-якому пристрої, який може бути чутливим до помилок даних. Флеш-пам'ять NAND містить 64-байтну резервну область для додаткового зберігання на кожній сторінці (16 байт на 512-байтний сектор). Цю резервну область можна використовувати для зберігання коду ECC, а також іншої програмної інформації, такої як інформація про рівень зносу або відображення логічного блоку на фізичний. ECC може виконуватися апаратно або програмно; однак апаратна реалізація забезпечує перевагу продуктивності [9].

Під час операції програмування блок ECC обчислює код ECC на основі даних, що зберігаються в секторі. Потім код ECC для області даних записується у відповідну резервну область. Коли дані зчитуються, також зчитується код ECC, і для перевірки правильності даних застосовується зворотна операція. Алгоритм ECC може виправляти помилки даних. Кількість помилок даних, які можна

виправити, залежить від потужності виправлення використаного алгоритму. Включення ЕСС в апаратне або програмне забезпечення забезпечує надійне рішення на системному рівні.

Прості коди Хеммінга забезпечують найпростішу апаратну реалізацію; вони можуть виправляти лише однобітові помилки. Коди Ріда-Соломона можуть забезпечити більш надійну можливість виправлення помилок і використовуються сьогодні в багатьох контролерах на ринку. Коди BCH також стають популярними через їх кращу ефективність порівняно з кодами Ріда-Соломона. Таблиця 11 показує кількість бітів, необхідних для різних рівнів корекції ЕСС. Коди в затінених комірках таблиці можуть поміститися у вільну область.

Таблиця 11 – Кількість бітів, необхідних для різних рівнів корекції ЕСС

Виправлення помилок	Необхідні біти в резервній області флеш-пам'яті NAND		
	Хеммінг	Рід-Соломон	BCH
1	13	18	13
2	N/A	36	26
3	N/A	54	39
4	N/A	72	52
5	N/A	90	65
6	N/A	108	78
7	N/A	126	91
8	N/A	144	104
9	N/A	162	117
10	N/A	180	130

Програмне забезпечення

Програмне забезпечення необхідне для керування блоками в пристрої NAND Flash. Це програмне забезпечення керує рівнем зносу та відображенням логічного та фізичного. Програмне забезпечення також може надати код ЕСС, якщо процесор не включає апаратне забезпечення ЕСС [8].

Важливо прочитати реєстр стану після операції PROGRAM або ERASE, щоб підтвердити успішне завершення операції. Якщо операція не вдається, блок має бути позначений як поганий і більше не використовуватися. Раніше запрограмовані дані слід перемістити з пошкодженого блоку в новий, хороший блок.

У специфікації флеш-пристрою SLC NAND ємністю 2 Гбіт зазначено, що він може мати до 40 пошкоджених блоків. Це максимальне число стосується терміну служби пристрою (номінально 100 000 циклів PRO-GRAM/ERASE). Здебільшого через великий розмір кристала флеш-пристрої NAND можуть поставлятися з заводу з низкою пошкоджених блоків. Програмне забезпечення, що керує пристроєм NAND Flash, відображає пошкоджені блоки та замінює їх хорошими блоками. Фабрика позначає ці блоки певним чином, щоб програмне забезпечення могло сканувати всі блоки, щоб визначити, які хороші, а які погані.

Позначка поганого блоку розміщується в першому місці резервної області (розташування стовпця 2048). Якщо розташування 2048 на сторінці 0 або сторінці 1 є «не FF», тоді блок слід вважати поганим і відобразити його поза системою. Програмне забезпечення ініціалізації може просто просканувати всі блоки, щоб визначити, які блоки є поганими, а потім створити таблицю цих пошкоджених блоків для подальшого використання.

Користувач повинен бути особливо обережним, щоб не стерти сліди поганих блоків. На заводі кожен флеш-пристрій NAND тестується в широкому діапазоні температур і напруг. Деякі блоки, позначені на заводі як погані, можуть працювати за певних температур або напруг, але можуть вийти з ладу в майбутньому. Якщо інформацію про пошкоджений блок стерто, її неможливо відновити [8].

Програмне забезпечення сторонніх виробників

Сьогодні на ринку є кілька пропозицій стороннього програмного забезпечення. Багато з цих пакетів забезпечують численні функції, включаючи

автоматичне відновлення після збою живлення, сумісність із файлами на ПК, ECC, керування пошкодженими блоками, підтримку каталогів і рівень зношення. Неповний список сторонніх постачальників програмного забезпечення NAND Flash включає:

- Datalight, Inc. – www.datalight.com
- CMX Systems, Inc. – www.cmx.com
- HCC-Embedded – www.hcc-embedded.com
- Blunk Microsystems – www.blunkmicro.com

Для реалізацій Linux іншою альтернативою є журнальна флеш-файлова система версії 2 (JFFS2).

Висновки до 4 розділу

Досліджено, що вибір процесора або контролера з вбудованим інтерфейсом NAND Flash надає значні переваги. Якщо це неможливо, встановлено, що можна розробити інтерфейс без клею між флеш-пристроєм NAND і майже будь-яким процесором. Основна відмінність між NAND Flash і NOR Flash полягає в мультиплексній шині, яка використовується для передачі команд, адрес і даних. Виявлено, що використання сигналів CLE та ALE дозволяє вибрати команду, адресу або цикл даних. Підключення ALE до біта адреси 5 процесора та CLE до біта адреси 4 дозволяє здійснювати ці операції шляхом зміни адреси, яку виводить процесор.

Проаналізовано, що процесори з вбудованим контролером NAND Flash, такі як Freescale i.MX21 та i.MX31, а також деякі процесори OMAP від Texas Instruments, забезпечують автоматичне завантаження з пристрою NAND Flash, логіку ECC та буфер SRAM. Виявлено, що MLC флеш-пам'ять зберігає 2 біти на комірку, що підвищує щільність зберігання даних, але поступається SLC за швидкістю та надійністю. З'ясовано, що SLC використовуються в високопродуктивних пристроях, а MLC — у споживчих продуктах.

Дослідження підтвердили, що код виправлення помилок (ECC) необхідний для забезпечення цілісності даних у NAND Flash. Встановлено, що прості коди Хеммінга виправляють однобітові помилки, тоді як коди Ріда-Соломона та BCH забезпечують більш надійну корекцію. Виявлено, що програмне забезпечення для керування блоками NAND Flash обробляє рівень зносу, відображення блоків та забезпечує ECC, якщо апаратне забезпечення його не підтримує.

З'ясовано, що важливо перевіряти реєстр стану після операцій PROGRAM або ERASE для виявлення та маркування пошкоджених блоків. Встановлено, що стороннє програмне забезпечення, таке як Datalight і CMX Systems, забезпечує додаткові функції, включаючи автоматичне відновлення після збою живлення та керування пошкодженими блоками. Досліджено, що для реалізацій Linux доступна журнальна флеш-файлова система JFFS2.

ВИСНОВКИ

Дипломна робота присвячена аналізу та порівнянню різних типів твердотільної пам'яті, зокрема NAND і NOR флеш-пам'яті, а також їхньому використанню в сучасних електронних пристроях. Актуальність теми зумовлена швидким розвитком інформаційних технологій і зростанням потреби в ефективних засобах зберігання великих обсягів даних, що надає дослідженню значне практичне та наукове значення.

1. Вивчено різні види твердотільної пам'яті (SSD), включаючи SATA SSD, NVMe SSD, mSATA, M.2, U.2 та зовнішні SSD. Виявлено їхні ключові характеристики, такі як швидкість читання та запису даних, енергоефективність, стійкість до механічних впливів, компактність і зручність використання. Підтверджено, що твердотільна пам'ять має переваги перед традиційними жорсткими дисками (HDD), зокрема вища продуктивність, відсутність рухомих частин і менша затримка доступу до даних.

2. Проведено детальний порівняльний аналіз флеш-пам'яті типу NAND і NOR. Встановлено, що NAND флеш-пам'ять характеризується вищою щільністю зберігання даних, нижчою вартістю на одиницю об'єму пам'яті та більшою енергоефективністю, що робить її ідеальною для зберігання великих обсягів інформації. З іншого боку, NOR флеш-пам'ять має швидший доступ до даних та надійніше зчитування, що робить її кращим вибором для застосувань, де важлива швидкість читання.

3. Досліджено архітектуру флеш-пам'яті NAND, включаючи організацію її блоків і сторінок, процеси запису та стирання даних. Виявлено обмеження флеш-пам'яті, зокрема максимальна кількість циклів запису/стирання, які може витримати комірка пам'яті. Показано, що методи вирівнювання зносу дозволяють рівномірно розподіляти цикли запису/стирання по всій пам'яті, запобігаючи надмірному зносу окремих блоків.

4. Виявлено, що сучасні процесори з вбудованими контролерами NAND підвищують ефективність та надійність систем зберігання даних. Зокрема,

досліджено можливості автоматичного завантаження з NAND флеш-пам'яті, вбудовану логіку ECC та буфер SRAM, що забезпечують більш ефективну та надійну роботу систем.

5. Розглянуто переваги та недоліки MLC і SLC флеш-пам'яті. Встановлено, що MLC флеш-пам'ять дозволяє зберігати більше даних завдяки зберіганню двох бітів інформації в кожній комірці, але поступається SLC за швидкістю та надійністю. З'ясовано, що SLC флеш-пам'ять, яка зберігає один біт на комірку, забезпечує вищу швидкість і більшу надійність, що робить її придатною для високопродуктивних застосувань.

6. Доказана важливість коду виправлення помилок (ECC) для забезпечення цілісності даних у NAND флеш-пам'яті. Виявлено, що використання ECC дозволяє виправляти помилки даних, які виникають під час зберігання та зчитування. Розглянуто різні типи ECC, зокрема коди Хеммінга, Ріда-Соломона та BCH, які забезпечують різний рівень захисту від помилок. Включення ECC в апаратне або програмне забезпечення забезпечує надійне рішення на системному рівні.

7. Виявлено, що програмне забезпечення для керування блоками у NAND флеш-пам'яті відповідає за рівень зносу, відображення логічних та фізичних блоків, а також забезпечує ECC, якщо апаратне забезпечення його не підтримує. Показано, що програмне забезпечення повинно коректно керувати пошкодженими блоками, позначаючи їх та замінюючи на робочі блоки. Розглянуто також сторонні рішення, такі як Datalight, CMX Systems, HCC-Embedded, що пропонують комплексні пакети програмного забезпечення для керування NAND флеш-пам'яттю.

8. Підтверджено, що флеш-пам'ять типу NAND є перспективною технологією для зберігання великих обсягів даних завдяки високій щільності, низькій вартості та енергоефективності. Використання сучасних контролерів та методів управління зносом забезпечує надійність та довговічність пристроїв на основі NAND флеш-пам'яті.

ПЕРЕЛІК ПОСИЛАНЬ

1. Основи електроніки. Автор: Богдан Іванов. Видавництво: "Львівська політехніка", 2020. – 480 с.
2. Мікроелектроніка. Автор: Володимир Петренко. Видавництво: "Наукова думка", 2019. – 608 с.
3. Системи зберігання даних. Автор: Сергій Коваленко. Видавництво: "Харківський національний університет радіоелектроніки", 2018. – 320 с.
4. SSD-диск що це і навіщо він потрібен? URL: <https://www.asisvok.com.ua/blog/item/ssd-disk-shcho-tse-i-navishcho-vin-potriben>
5. SSD Types and Form Factors – An SSD Primer. URL: <https://www.thessdreview.com/ssd-guides/beginners-guide/ssd-types-and-form-factors-an-ssd-primer/>
6. SSD. URL: <https://uk.wikipedia.org/wiki/SSD>
7. Flash пам'ять. URL: <https://uk.wikipedia.org/wiki/%D0%A4%D0%BB%D0%B5%D1%88%D0%BF%D0%B0%D0%BC%27%D1%8F%D1%82%D1%8C>
8. NAND Flash Technology and Solid-State Drives (SSDs). URL: <https://www.kingston.com/en/blog/pc-performance/nand-flash-technology-and-ssd>
9. Flash Memory: NOR vs. NAND. URL: <https://www.baeldung.com/cs/flash-memory-nor-vs-nand>
10. ПОСІБНИК ІЗ ФЛЕШ-ПАМ'ЯТІ. URL: https://media.kingston.com/pdfs/MKF_283.2_Flash_Memory_Guide_UA.pdf
11. "NAND Flash Memory Technology: A Comprehensive Guide" by Jedec. URL: <https://www.jedec.org/standards-documents/docs/jesd230c>
12. Flash memory 101: An introduction to NAND flash. URL: <https://www.edn.com/flash-memory-101-an-introduction-to-nand-flash/>
13. Open NAND Flash Specification. URL: <https://www.issi.com/US/product-flash-nand.shtml>

14. How NAND Flash Memory Works" by HowStuffWorks. URL: <https://www.youtube.com/watch?v=YtBysgPOKx4>
15. "A Survey of NAND Flash Memory Technology" by Young-Hoon Lee et al. URL: <https://ieeexplore.ieee.org/document/9467859>
16. NAND Flash 101: An Introduction to NAND Flash and How to Design It In to Your Next Product by Micron.

ДОДАТОК А
SUMMARY

SUMMARY

Solid-state drives (SSDs) have an interesting history that began with the development of flash memory. The first steps were taken in the 1980s when companies like Toshiba and Intel started developing flash memory, which allowed data storage even without power. By the late 1990s, the first commercial SSDs based on NAND flash memory were created. NAND became the main technology due to its high storage density and lower production costs compared to other types of memory.

Despite the dominance of NAND memory, another type of flash memory, NOR, also played a significant role in the history of solid-state storage development. NOR memory was designed to provide fast data access, making it ideal for storing firmware and software code that is often read but rarely rewritten. Unlike NAND, NOR flash memory allows direct code execution from the chip, providing faster access and reading.

NAND flash memory became the foundation for many modern file systems due to its unique properties. The main feature of NAND memory is its ability to write and read data in blocks or pages, which is significantly different from traditional hard drives. This provides faster data access and reduced latency, which is especially important for modern computer systems and mobile devices. The file system for NAND memory needs to optimize write and read processes, considering the limited number of rewrite cycles for each memory cell.

One of the important features of NAND file systems is their ability to handle cell "wear." To do this, wear leveling algorithms are used, ensuring an even distribution of data writes across the entire memory. This significantly extends the lifespan of SSDs. Additionally, such file systems often use garbage collection techniques to optimize space and increase write efficiency.

Trends in the development of NAND flash memory include reducing the manufacturing process, increasing storage density, and improving error correction algorithms. This allows the creation of more capacious and faster SSDs at lower costs. New types of memory, such as 3D NAND, where cells are arranged in layers, are also

being developed to further increase storage density. Innovations in memory controller technologies and data transfer protocols, such as NVMe, also significantly enhance SSD performance and reliability.

The presence of NAND flash memory is essential for modern computer systems due to its high performance, reliability, and energy efficiency. It is used in a variety of devices, from smartphones and tablets to laptops and servers. SSDs based on NAND flash memory ensure fast loading of operating systems, programs, and games, as well as speeding up work with large files, making them indispensable in many areas.

Modern SSDs come in various types and form factors, including 2.5-inch drives, M.2 modules, and PCIe cards. Each has its advantages depending on the specific application. For example, M.2 SSDs provide high data transfer speeds in a compact form factor, making them ideal for ultrabooks and high-performance desktops. With the development of NAND flash memory technologies, further improvements in performance, cost reduction, and expansion of application areas for these drives are expected.

In summary, this research has highlighted the fundamental differences between NAND and NOR flash memory, focusing on their respective file structures, advantages, and limitations. NAND flash memory excels in high storage density and cost efficiency, making it ideal for mass storage applications. It operates efficiently with block-level data management, despite its more complex error correction and wear leveling requirements. Conversely, NOR flash memory offers faster read speeds and direct code execution capabilities, which are advantageous for applications requiring rapid and frequent data access, such as firmware storage.

The command principles and timing diagrams examined in this study reveal the operational intricacies of both memory types. NAND's block-oriented approach and slower read times are balanced by its ability to store large amounts of data cost-effectively. NOR's faster read speeds and simpler memory architecture facilitate quick data retrieval, though at a higher cost and lower storage density.

Overall, NAND and NOR flash memories each serve distinct roles in modern computing, with their respective strengths and weaknesses making them suitable for

different applications. Understanding their operational principles and performance characteristics allows for informed decisions in their implementation, ensuring optimized performance and reliability in various technological contexts.