

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління  
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)  
(рівень вищої освіти)

Методи ефективного синтезу шифроторів на ПЛІС  
(тема)

Виконав: Здобувач другого року навчання,  
групи СКСм-23-2 Ратушний І.В.  
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія

Тип програми освітньо-професійна  
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи  
(повна назва освітньої програми)

Керівник роботи доц.Філіпенко І.В.  
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри \_\_\_\_\_  
(підпис)

Чумаченко С.В.  
(прізвище, ініціали)

2025 р.

Харківський національний університет радіоелектроніки

Факультет \_\_\_\_\_ Комп'ютерної інженерії та управління \_\_\_\_\_

Кафедра \_\_\_\_\_ Автоматизації проектування обчислювальної техніки \_\_\_\_\_

Рівень вищої освіти \_\_\_\_\_ другий (магістерський) \_\_\_\_\_

Спеціальність \_\_\_\_\_ 123 Комп'ютерна інженерія \_\_\_\_\_  
(шифр і назва)

Тип програми \_\_\_\_\_ Освітньо-професійна \_\_\_\_\_  
(освітньо-професійна або освітньо-наукова)

Освітня програма \_\_\_\_\_ Спеціалізовані комп'ютерні системи \_\_\_\_\_  
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
(підпис)

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві \_\_\_\_\_ Ратушному Іллі Вікторовичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи (проекту) Методи ефективного синтезу шифраторів на ПЛІС

затверджена наказом по університету від « 08 » 11 2024р. № 1189 Ст

2. Термін подання здобувачем роботи \_\_\_\_\_ 15.12.2024 \_\_\_\_\_

3. Вихідні дані до роботи (проекту) \_\_\_\_\_

Основні принципи реалізації та аналізу цифрових пристроїв

Мова програмування С

Мови опису апаратури VHDL, Verilog

САПР Quartus Prime

Перелік питань, що потрібно опрацювати у роботі \_\_\_\_\_

Дослідження принципів роботи неперіоритетних і пріоритетних шифраторів

Аналіз методів реалізації шифраторів у Verilog

Вивчення впливу різних профілів оптимізації на характеристики схем

Проведення часового аналізу критичних шляхів

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 17

---

---

---

6. Консультанти розділів роботи (проекту)


Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

7. Дата видачі завдання 02.09.2024

### КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи (проекту)	Термін виконання етапів роботи	Примітка
1	Видача теми проекту, узгодження і затвердження теми	09.09.2024-22.09.2024	
2	Аналіз проблемної галузі, постановка задачі, вибір інструментальних засобів	23.09. 2024-30.09.2024	
3	Реалізація початкових моделей шифраторів	01.10.2024-14.10.2024	
4	Проведення моделювання та тестування моделей	15.09. 2024-19.10.2024	
5	Виконання аналізу часових характеристик	20.10. 2024-19.11. 2024	
6	Синтез моделей з урахуванням оптимізацій	20.11. 2024-01.12. 2024	
7	Оформлення пояснювальної записки	02.12. 2024-22.12. 2024	
8	Захист проекту	03.01. 2025-15.01. 2025	

Студент   
(підпис)

Керівник роботи (проекту)   
(підпис)

доц.Філіпенко І.В.  
(посада, прізвище, ініціали)

## РЕФЕРАТ

Магістерська кваліфікаційна робота містить 64 сторінки, 4 таблиці, 47 рисунків, та 7 джерел.

ПЛІС, ШИФРАТОР, НЕПРІОРИТЕТНИЙ ШИФРАТОР, ПРІОРИТЕТНИЙ ШИФРАТОР, СИНТЕЗ ЦИФРОВИХ СХЕМ, ОПТИМІЗАЦІЯ, QUARTUS PRIME, TIMEQUEST, ЧАСОВИЙ АНАЛІЗ.

У даній роботі розглянуто принципи роботи та проектування неперіоритетних і пріоритетних шифраторів. Було проведено синтез шифраторів, описаних на мові Verilog, із використанням різних логічних операторів (assign, if, case). Серед створених моделей виділено шифратори з меншими затримками і мінімальним використанням логічних елементів, які виявилися найефективнішими для реалізації на ПЛІС.

Засобами аналізу TimeQuest проведено часовий аналіз роботи шифраторів та досліджено критичні шляхи проходження сигналів. Зроблено висновки про вплив профілів оптимізації (Balanced, Performance, Power, Area) на час затримки сигналів і продуктивність схем. Найвищу швидкодію було досягнуто при використанні профілю Performance, що підтверджує ефективність застосування профілів оптимізації для розробки продуктивних цифрових систем.

## ABSTRACT

The master's qualification thesis consists of 64 pages, 4 tables, 47 figures, and 7 sources.

FPGA, ENCODER, NON-PRIORITY ENCODER, PRIORITY ENCODER, DIGITAL CIRCUIT SYNTHESIS, OPTIMIZATION, QUARTUS PRIME, TIMEQUEST, TIMING ANALYSIS.

This work examines the principles of operation and design of non-priority and priority encoders. The synthesis of encoders described in Verilog was performed using various logical operators (assign, if, case). Among the created models, encoders with lower delays and minimal use of logic elements were identified as the most efficient for FPGA implementation.

Using TimeQuest analysis tools, timing analysis of the encoders was conducted, and the critical signal paths were examined. Conclusions were made regarding the impact of optimization profiles (Balanced, Performance, Power, Area) on signal delay times and circuit performance. The highest performance was achieved using the Performance profile, confirming the effectiveness of applying optimization profiles for the development of high-performance digital systems.

# ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ .....	8
ВСТУП .....	9
1 ІСТОРІЯ РОЗВИТКУ ПЛІС .....	10
2 ТИПИ ПЛІС ТА ЇХ ЗАСТОСУВАННЯ .....	13
2.1 FPGA на основі SRAM .....	13
2.2 Antifuse-based FPGA .....	15
2.3 FPGA з флеш-пам'яттю .....	17
2.4 Приклади використання FPGA .....	19
2.4.1 Мережеві технології .....	19
2.4.2 Обробка відео .....	20
2.4.3 Штучний інтелект(ШІ) .....	20
2.4.4 Автомобілебудування .....	21
2.4.5 Аерокосмічна галузь .....	22
3 ШИФРАТОРИ .....	24
3.1 Опис шифратора на мові Verilog .....	25
3.2 Непріоритетний шифратор на основі оператора безперервного присвоєння assign .....	26
3.3 Непріоритетний шифратор, реалізований за допомогою операторів if ..	28
3.3 Непріоритетний шифратор, реалізований за допомогою оператора case .....	30
3.5 Вивчення часових характеристик цифрових пристроїв .....	32
4 ОПТИМІЗАЦІЯ ПРИ СИНТЕЗІ .....	34
4.1 Режими оптимізації .....	34
4.2 Порівняння режимів оптимізації .....	34
4.2.1 Результати часового аналізу при використанні різних режимів оптимізації для шифратора b1_enc_assign .....	35

4.2.2	Результати часового аналізу при використанні різних режимів оптимізації для шифратора b2_enc_if.....	38
4.2.3	Результати часового аналізу при використанні різних режимів оптимізації для шифратора b3_enc_case.....	41
4.2.4	Аналіз шляхів проходження сигналу в Chip Planner при використанні різних профілів оптимізації для шифратора b1_enc_assign .....	45
4.2.5	Аналіз шляхів проходження сигналу в Chip Planner при використанні різних профілів оптимізації для шифратора b2_enc_if .....	47
4.2.6	Аналіз шляхів проходження сигналу в Chip Planner при використанні різних профілів оптимізації для шифратора b3_enc_case .....	50
4.2.7	Аналіз часових діаграми при використанні різних профілів оптимізації для шифратора b1_enc_assign .....	52
4.2.8	Аналіз часових діаграми при використанні різних профілів оптимізації для шифратора b2_enc_if.....	55
4.2.9	Аналіз часових діаграми при використанні різних профілів оптимізації для шифратора b3_enc_case.....	58
	ВИСНОВКИ .....	62
	ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	64
	ДОДАТОК А .....	65

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

FPGA (Field-Programmable Gate Array) – програмована логічна інтегральна схема;

ASIC (Application-Specific Integrated Circuit) – спеціалізована інтегральна схема;

DSP (Digital Signal Processing) – цифрова обробка сигналів;

RTL (Register Transfer Level) – рівень опису цифрових схем;

SDN (Software-Defined Networking) – програмно-визначені мережі;

SoC (System-on-Chip) – система на кристалі;

MIMO (Multiple Input Multiple Output) – технологія багатоканального прийому та передачі;

VHDL (VHSIC Hardware Description Language) – мова опису апаратури;

H.264, H.265, AV1 – стандарти компресії відео;

Critical Path – критичний шлях у цифровій схемі, який визначає максимальну швидкодію;

TimeQuest – інструмент часових аналізів у Quartus Prime;

Balanced Mode – режим збалансованої оптимізації;

Performance Mode – режим оптимізації для досягнення високої швидкодії;

Power Mode – режим оптимізації для зниження енергоспоживання;

Area Mode – режим оптимізації для мінімізації площі логічних елементів;

DVFS (Dynamic Voltage and Frequency Scaling) – динамічне керування напругою та частотою.

## ВСТУП

Цифрові шифратори є важливими компонентами у сучасних обчислювальних системах, оскільки вони виконують перетворення вхідних сигналів з одного коду в інший, забезпечуючи зручне представлення даних для подальшої обробки. Шифратори бувають різних типів, зокрема неприоритетні та пріоритетні, що дозволяє обирати відповідну модель залежно від вимог системи. Неприоритетні шифратори формують вихідний код на основі положення єдиної активної логічної одиниці на вході, тоді як пріоритетні шифратори обробляють сигнали з пріоритетом для старшого або молодшого біта, що є важливим у контролерах переривань.

Quartus Prime, як сучасна програмна система автоматизованого проектування від Intel FPGA, забезпечує комплексні інструменти для проектування, синтезу та аналізу цифрових пристроїв на ПЛІС. Інструмент TimeQuest у складі Quartus Prime дозволяє здійснювати точний часовий аналіз шляхів проходження сигналів, визначати критичні шляхи та аналізувати затримки у комбінаційних схемах. Така функціональність надзвичайно важлива для забезпечення стабільної роботи пристроїв на високих частотах, оскільки дає змогу виявляти та мінімізувати затримки у ключових вузлах схеми.

За допомогою Quartus Prime і TimeQuest розробники можуть не лише створювати та оптимізувати цифрові схеми, але й тестувати різні підходи до їх реалізації та застосовувати профілі оптимізації (наприклад, Balanced, Performance, Power), що дозволяє збалансувати ресурси ПЛІС з урахуванням швидкодії та енергоспоживання.

## 1 ІСТОРІЯ РОЗВИТКУ ПЛІС

Програмовані логічні інтегральні схеми (ПЛІС) – це родина інтегральних схем, призначених для того, щоб користувачі могли програмувати їх згідно з визначеною логікою. Історія створення ПЛІС розпочинається з розвитку напівпровідникової промисловості та появи перших інтегральних схем.

Перші інтегральні схеми були створені в 1950-х роках, і протягом наступних двох десятиліть вони ставали дедалі складнішими та дешевшими у виробництві. У середині 1970-х років було розроблено перші програмовані логічні пристрої (ПЛП), які стали попередниками сучасних ПЛІС. ПЛП надавали можливість налаштовувати логічні схеми після виготовлення кристала, що було важливою перевагою для багатьох розробників.

У 1983 році компанія Xilinx представила першу комерційну ПЛІС під назвою XC2064. Цей чип складався із 64 програмованих логічних блоків і використовував програмовану матрицю для визначення зв'язків між блоками. Пристрій забезпечував не лише гнучкість у програмуванні, а й можливість повторного використання, адже його можна було перепрограмувати.

З появою перших ПЛІС почали розроблятися різноманітні архітектури та технології. Одним із ключових досягнень стало створення ПЛІС на базі SRAM, які використовували статичну пам'ять для збереження налаштувань логіки. Це дало змогу перепрограмувати чипи безпосередньо у пристроях, без їх вилучення з системи.

Технічні характеристики ПЛІС постійно вдосконалювалися разом із розвитком технологій. У 1990-х роках з'явилися перші ПЛІС із вбудованими процесорами, що дозволило створювати більш інтегровані системи на кристалі (SoC). Крім того, було впроваджено такі функції, як блоки пам'яті, цифрові сигнальні процесори (ЦСП) і трансивери для високошвидкісної передачі даних.

З роками ПЛІС ставали дедалі складнішими та потужнішими. Сьогодні вони знаходять застосування у різних галузях, включаючи зв'язок, автоматизацію, обробку даних і автомобільну промисловість. Сучасні ПЛІС, такі як Xilinx Virtex UltraScale+ і Intel Stratix 10, містять мільйони логічних елементів, сотні блоків пам'яті й трансиверів, а також вбудовані процесори та інтерфейси для підключення до зовнішніх пристроїв.

Окрім підвищення продуктивності, сучасні ПЛІС також пропонують вдосконалені можливості для розробки та налагодження. Виробники ПЛІС надають розробникам потужні інструменти, такі як інтегровані середовища розробки (IDE), симулятори та аналізатори для проектування й тестування схем. Це дозволяє розробникам швидко створювати й оптимізувати свої проекти, спрощує процес налагодження та підвищує надійність рішень на основі ПЛІС.

З розвитком технологій і появою нових сфер застосування ПЛІС, таких як Інтернет речей (IoT) і автомобільна промисловість, виробники інтегральних схем продовжують удосконалювати свої продукти. Останніми роками зростає інтерес до малопотужних ПЛІС, таких як Xilinx Artix та Intel Cyclone, які забезпечують оптимальне поєднання продуктивності та енергоефективності для застосувань із низьким енергоспоживанням.

Крім того, тривають дослідження й розробки нових матеріалів і технологій для покращення характеристик ПЛІС. Наприклад, вивчаються можливості використання нових матеріалів, таких як графен і вуглецеві нанотрубки, для створення більш швидких та енергоефективних ПЛІС. Також розробляються нові архітектури та методи програмування, зокрема оптичне програмування й квантові ПЛІС, які можуть відкрити нові горизонти у використанні програмованих логічних пристроїв.

Підсумовуючи, історія розвитку ПЛІС — це історія технологічного прогресу, що почався з появи перших інтегральних схем і розвитку напівпровідникової промисловості. Відтоді виробники інтегральних схем постійно вдосконалюють ПЛІС, пропонуючи нові архітектури, технології та

функціональні можливості. У результаті ПЛІС стали ключовим компонентом сучасних високотехнологічних систем і продовжують розвиватися, відкриваючи нові можливості для розробників та інженерів у всьому світі.

## 2 ТИПИ ПЛІС ТА ЇХ ЗАСТОСУВАННЯ

### 2.1 FPGA на основі SRAM

FPGA на основі SRAM (Field-Programmable Gate Arrays) використовують статичну оперативну пам'ять (SRAM) для збереження конфігураційних даних. Цей тип пристроїв є найпоширенішим завдяки високій гнучкості, продуктивності та можливості багаторазового перепрограмування. Конфігураційні дані задають логіку роботи FPGA, визначаючи зв'язки між логічними блоками та каналами маршрутизації, що дозволяє реалізовувати широкий спектр цифрових функцій.

Однією з особливостей FPGA на основі SRAM є необхідність повторної конфігурації після кожного ввімкнення живлення, оскільки пам'ять SRAM є енергозалежною. Це створює додаткові виклики, наприклад, підвищену вразливість до втрати живлення або потребу у зберіганні конфігураційних даних у зовнішніх енергонезалежних пам'ятях, таких як флеш-пам'ять або EEPROM.

Попри це, FPGA на основі SRAM мають низку переваг, які роблять їх ідеальними для широкого спектра застосувань:

- висока швидкодія, завдяки використанню SRAM, ці пристрої забезпечують швидкий доступ до даних, що особливо важливо для задач із високими вимогами до продуктивності, таких як обробка сигналів у реальному часі.

- можливість динамічної ре конфігурації; ці пристрої підтримують функцію часткової реконфігурації, що дозволяє змінювати частину логіки або маршрутизації під час роботи без зупинки інших функцій.

Більш детальну структуру цих пристроїв можна побачити на рис. 2.1 "Архітектура FPGA на основі SRAM", який демонструє основні елементи архітектури.

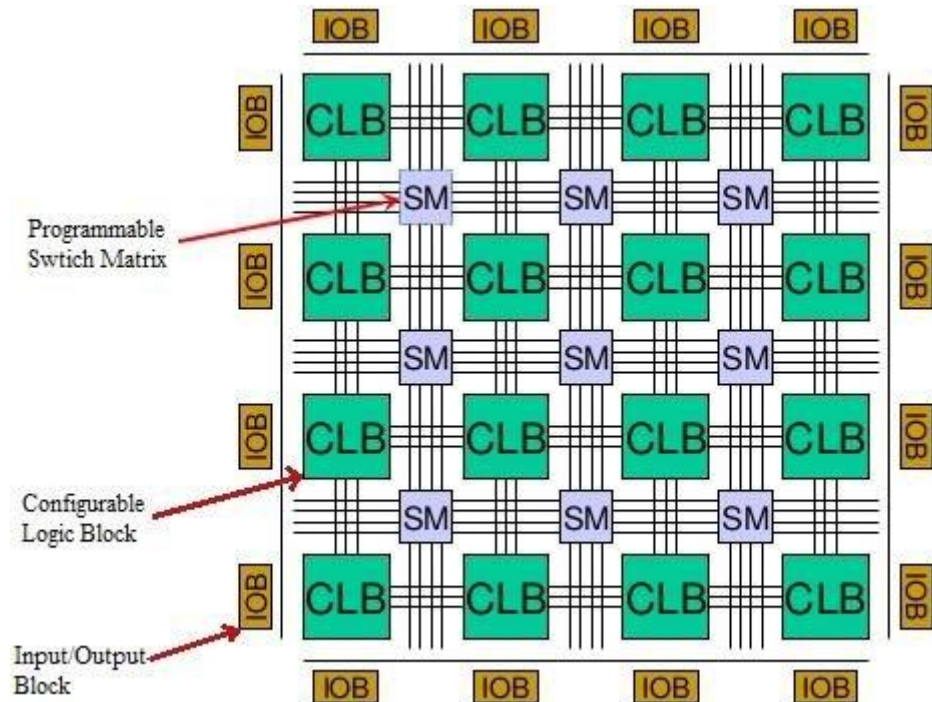


Рисунок 2.1 – Архітектура FPGA на основі SRAM

У сфері обробки сигналів у реальному часі FPGA на основі SRAM демонструють свою ефективність у задачах цифрової обробки аудіо- та відеосигналів, а також при обчисленні складних алгоритмів у системах DSP (Digital Signal Processing). Вони використовуються для таких застосувань, як шумоподавлення, обробка зображень або кодування відео, що робить їх важливим компонентом мультимедійних технологій.

Для розробки прототипів цифрових пристроїв ці FPGA є цінним інструментом. Вони забезпечують швидке створення прототипів цифрових систем, дозволяючи перевірити функціональність розроблених архітектур до їх впровадження в ASIC. Такий підхід є особливо корисним на етапах досліджень і розробок, коли потрібна максимальна гнучкість і адаптивність.

У вбудованих системах FPGA на основі SRAM знаходять широке застосування, наприклад, у промислових контролерах, автомобільній електроніці та пристроях Інтернету речей (IoT). Вони забезпечують високу

продуктивність, низьку затримку й відповідність вимогам сучасних вбудованих технологій.

FPGA на основі SRAM також активно використовуються в мережевих пристроях, таких як маршрутизатори й комутатори, забезпечуючи ефективну обробку даних, а також у високопродуктивних обчислювальних системах, де їх застосовують для апаратного прискорення складних алгоритмів.

## 2.2 Antifuse-based FPGA

Antifuse-based FPGA є типом програмованих логічних інтегральних схем, які використовують технологію одноразового створення електричних з'єднань між логічними елементами. Ці з'єднання формуються шляхом пропускання сильного електричного струму через спеціальні антифузійні структури, що призводить до фізичної зміни їх стану. Після програмування структура FPGA стає постійною, що виключає можливість її повторної конфігурації, але забезпечує високу надійність і стійкість.

Однією з головних переваг Antifuse-based FPGA є їх висока стійкість до радіації, електромагнітних полів і температурних коливань, що робить ці пристрої оптимальними для використання в екстремальних умовах. Це досягається завдяки фізичній природі створених з'єднань, які не піддаються впливу зовнішніх факторів.

На рис. 2.2 зображена структура таких пристроїв, яка демонструє логічні елементи, міжз'єднання та принципи роботи.

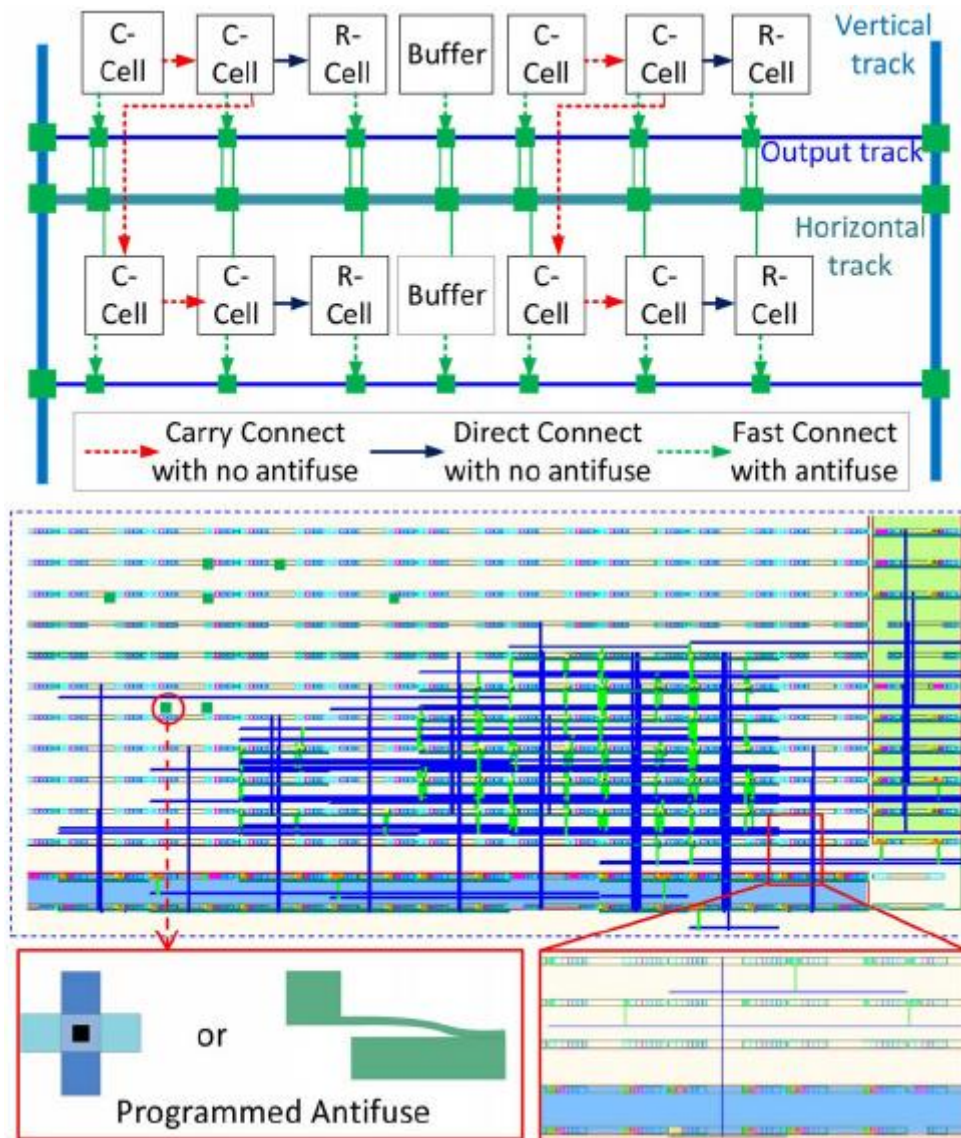


Рисунок 2.2 – Архітектура Antifuse-based FPGA

Основними галузями застосування Antifuse-based FPGA є сфера космічних апаратів. Antifuse-based FPGA є невід'ємним компонентом супутників, марсоходів та інших космічних пристроїв. Їх висока стійкість до радіації забезпечує надійну роботу в умовах відкритого космосу протягом тривалого часу, що є критичним для виконання космічних місій.

У військовій техніці такі FPGA використовуються в системах управління зброєю, навігації та зв'язку. Їх здатність працювати в екстремальних умовах, зокрема за високих температур, вібрацій та

електромагнітних завад, робить їх ключовими компонентами сучасних військових систем.

Для критично важливих систем Antifuse-based FPGA є незамінними в авіації, атомній енергетиці та інших галузях, де відмова обладнання може призвести до катастрофічних наслідків. Їх висока надійність гарантує безпеку та стабільність роботи навіть у найвідповідальніших умовах.

Antifuse-based FPGA завдяки своїм характеристикам знаходять застосування у різноманітних сферах, що вимагають високої стійкості та безвідмовності. Вони є вузькоспеціалізованим рішенням для задач, де критично важливими є стійкість до зовнішніх впливів і абсолютна надійність, навіть за рахунок відсутності можливості перепрограмування. Це робить їх незамінними в сферах, пов'язаних із безпекою та екстремальними умовами експлуатації.

### 2.3 FPGA з флеш-пам'яттю

FPGA з флеш-пам'яттю є типом програмованих логічних інтегральних схем, які зберігають конфігураційні дані у вбудованій енергонезалежній флеш-пам'яті. Завдяки цьому такі пристрої мають здатність зберігати конфігурацію навіть після втрати живлення, що робить їх універсальними для використання у багатьох сферах. На рис. 2.3 "Архітектура FPGA з флеш-пам'яттю" показано принципову схему, що ілюструє організацію логічних елементів та їх взаємодію.

Флеш-основа FPGA поєднує переваги FPGA на основі SRAM та Antifuse-based FPGA, забезпечуючи гнучкість у програмуванні та високу надійність. На відміну від FPGA на основі SRAM, ці пристрої не потребують зовнішніх джерел пам'яті для збереження конфігурації. Водночас вони підтримують багаторазове перепрограмування, що вигідно вирізняє їх на тлі антифузійних FPGA, які дозволяють лише одноразове програмування.

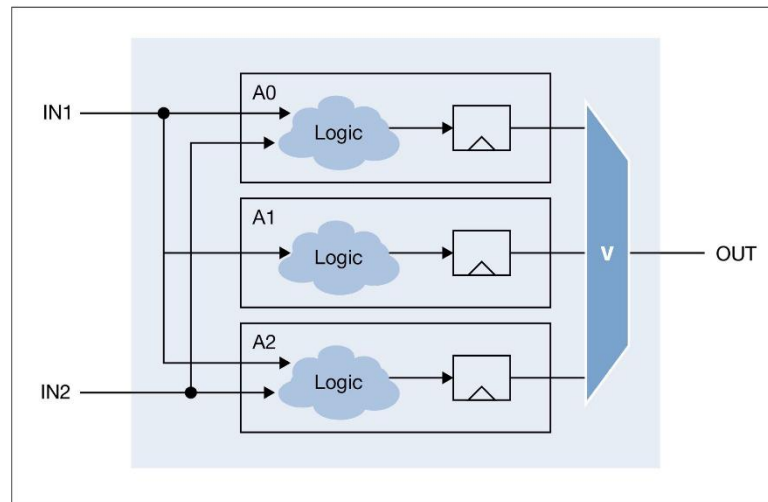


Рисунок 2.3 – Архітектура FPGA з флеш-пам'яттю

FPGA з флеш-пам'яттю широко застосовуються у промислових системах, де важливими є надійність та енергонезалежність. Ці пристрої знаходять використання у системах автоматизації виробничих процесів, моніторингу стану обладнання та робототехніці, забезпечуючи стабільність роботи навіть у складних умовах.

У сфері автомобільної електроніки такі FPGA виконують ключові функції в системах допомоги водіям (ADAS), керуванні двигуном, навігації та мультимедійних комплексах. Їхня здатність працювати без збоїв за перепадів напруги робить їх надійним рішенням для автомобільних систем.

У побутовій електроніці FPGA з флеш-пам'яттю використовуються для створення розумних пристроїв, зокрема приладів Інтернету речей (IoT), автоматизованих домашніх систем і сучасного медичного обладнання. Їх застосовують у пральних машинах, кондиціонерах та холодильниках для підвищення функціональності й зручності.

FPGA з флеш-пам'яттю відповідають вимогам надійності, енергонезалежності та адаптивності, що робить їх важливим компонентом у розвитку промислових, автомобільних і побутових технологій.

## 2.4 Приклади використання FPGA

FPGA знайшли широке застосування в багатьох галузях завдяки своїй універсальності, гнучкості у конфігурації та високій продуктивності. Їх здатність адаптуватися до різноманітних задач і ефективно виконувати спеціалізовані обчислення робить їх ключовими компонентами у сучасних технологіях. Нижче розглянемо основні напрямки застосування FPGA із детальнішим аналізом їх ролі в кожній галузі.

### 2.4.1 Мережеві технології

FPGA активно використовуються для створення високошвидкісних маршрутизаторів, комутаторів і мережевих адаптерів, забезпечуючи продуктивність і гнучкість у сучасних мережах. Однією з ключових їхніх функцій є реалізація апаратного прискорення, що дозволяє обробляти пакети в реальному часі завдяки низьким затримкам і можливості паралельної обробки даних. Це критично важливо для мереж 5G, центрів обробки даних і хмарних обчислень, де потрібна швидка й ефективна обробка великих обсягів інформації.

Для забезпечення безпеки даних FPGA надають можливість шифрування і безпеки, реалізуючи апаратні алгоритми шифрування, дешифрування та інші криптографічні протоколи. Це робить їх незамінними у високозахищених мережах, де критично важливий захист даних від атак.

У галузі розвитку SDN (програмно-визначених мереж) FPGA відіграють важливу роль, забезпечуючи динамічну маршрутизацію трафіку й аналітику. Це дозволяє оптимізувати використання мережевих ресурсів і забезпечувати адаптивну реакцію на зміну умов трафіку в реальному часі.

Завдяки своїм можливостям FPGA сприяють розвитку інфраструктури високошвидкісних мереж, забезпечуючи їхню продуктивність, безпеку та адаптивність до сучасних вимог.

### 2.4.2 Обробка відео

FPGA є ідеальним рішенням для задач, пов'язаних із обробкою відеосигналів і зображень, завдяки їхній високій продуктивності та здатності до паралельних обчислень. Однією з ключових функцій є компресія та декомпресія, де FPGA дозволяють ефективно реалізовувати алгоритми кодування та декодування відео, такі як H.264, H.265 або AV1. Це особливо важливо для застосувань у потоковому відео, відеоконференціях і системах відеоспостереження, де потрібна низька затримка та висока якість.

Ще однією важливою областю є обробка зображень, де FPGA використовуються для покращення якості відеоматеріалів. Це включає шумозаглушення, корекцію кольорів і підвищення роздільної здатності, що дозволяє досягти значного поліпшення візуального сприйняття зображень.

Для задач аналізу відеопотоку FPGA пропонують значні переваги завдяки підтримці паралельних обчислень. У сфері відеоаналітики вони забезпечують виконання складних завдань, таких як розпізнавання об'єктів, обчислення траєкторій і моніторинг ситуації в реальному часі. Це робить їх незамінними для систем безпеки, автоматизованих виробничих процесів і сучасних транспортних систем.

FPGA поєднують продуктивність, гнучкість та енергоефективність, забезпечуючи провідні рішення для обробки відеосигналів і зображень у різних галузях.

### 2.4.3 Штучний інтелект(ШІ)

FPGA є ефективним інструментом для прискорення обчислень у галузі штучного інтелекту та машинного навчання, де їх здатність до високопродуктивної обробки даних є вирішальною. Однією з ключових областей застосування є нейронні мережі, де FPGA забезпечують апаратну

реалізацію моделей, значно прискорюючи тренування та виконання алгоритмів. Це включає застосування у задачах обробки природної мови (NLP), комп'ютерного зору та класифікації даних, що потребують великих обсягів обчислень.

Іншим важливим аспектом є енергоефективність FPGA, яка є суттєвою перевагою порівняно з загальнопроцесорними (CPU) або графічними процесорами (GPU). FPGA здатні забезпечити високу продуктивність при мінімальному енергоспоживанні, що робить їх ідеальними для вбудованих систем ІІІ, де важливими є як ефективність, так і енергозбереження.

Розподілені обчислення — ще одна важлива сфера, де FPGA демонструють свою ефективність. У великих ІІІ-проектах FPGA можуть інтегруватися з іншими апаратними прискорювачами для забезпечення паралельної обробки даних у хмарних обчисленнях. Це дозволяє значно збільшити пропускну здатність системи, підвищити швидкість обробки та зменшити затримки при виконанні складних обчислювальних задач.

Загалом, FPGA відіграють ключову роль у розвитку сучасних технологій ІІІ, забезпечуючи високопродуктивні, енергоефективні та масштабовані рішення для різноманітних застосувань у галузі машинного навчання та штучного інтелекту.

#### 2.4.4 Автомобілебудування

FPGA широко застосовуються в сучасній автомобільній електроніці, де вони виконують критично важливі задачі. Одним із основних напрямків використання є адаптивні круїз-контролі, де FPGA забезпечують обробку даних з датчиків, таких як радар або LIDAR, для підтримки безпечної дистанції між транспортними засобами. Завдяки високій швидкодії та адаптивності FPGA гарантують точне реагування на зміни в дорожній ситуації, що підвищує безпеку водіїв.

Іншим важливим аспектом є системи допомоги водієві (ADAS). FPGA використовуються для реалізації алгоритмів розпізнавання дорожніх знаків, пішоходів і об'єктів у реальному часі, що значно підвищує безпеку руху. Вони забезпечують швидку обробку відеоданих і здатні ефективно інтегрувати інформацію з різних сенсорів для прийняття рішень в реальному часі.

Також FPGA активно використовуються в електронних блоках керування (ECU), де вони оптимізують роботу двигуна, систем гальмування, контролю за витратою пального та іншими критичними функціями автомобіля. Завдяки своїй гнучкості та можливості реалізувати складні алгоритми, FPGA дозволяють досягти високого рівня ефективності та надійності в автомобільних системах.

Важливою сферою застосування є електромобілі, де FPGA забезпечують високу точність і адаптивність у системах управління зарядкою акумуляторів. Вони допомагають ефективно контролювати процеси зарядки, запобігаючи перевантаженням та максимізуючи енергетичну ефективність.

Таким чином, FPGA відіграють ключову роль у розвитку сучасних технологій автомобілебудування, забезпечуючи високу продуктивність, надійність і безпеку в різноманітних автомобільних системах.

#### 2.4.5 Аерокосмічна галузь

FPGA є невід'ємною частиною багатьох аерокосмічних технологій, де їх надійність та продуктивність мають вирішальне значення. Одним із основних напрямків використання є обробка телеметричних даних. У космічних апаратах FPGA застосовуються для збору, аналізу та передачі даних із численних сенсорів у реальному часі. Це дозволяє забезпечити високу точність і швидкість у процесі моніторингу стану космічних апаратів і їх систем.

Ще одним важливим аспектом є навігаційні системи. FPGA активно використовуються для реалізації високоточних алгоритмів GPS-навігації, інерціальної навігації та систем управління польотом. Це дозволяє забезпечити стабільну та точну роботу навігаційних систем, навіть в умовах високих швидкостей та складних космічних середовищ.

Космічна техніка також виграє від використання FPGA завдяки їх стійкості до радіації та високим температурним коливанням. Вони активно використовуються в супутниках, марсоходах і дослідницьких зондах, де необхідно працювати в екстремальних умовах і забезпечувати надійність та високу продуктивність обчислень.

Таким чином, FPGA займають провідну роль у багатьох сучасних технологічних розробках, сприяючи прогресу у сферах комунікацій, обробки даних, штучного інтелекту, автомобілебудування та аерокосмічної промисловості. Їх універсальність дозволяє адаптувати рішення під унікальні потреби кожної галузі.

### 3 ШИФРАТОРИ

Шифратор є кодуєчим пристроєм, тобто пристроєм, що перетворює вхідний багаторозрядний код у вихідний, побудований за іншим принципом. У таблиці 3.1 наведено таблицю істинності класичного неприоритетного шифратора (або кодера, encoder), який приймає на вхід 16-розрядний унарний код (логічна одиниця може бути лише в одному розряді або не бути взагалі, якщо всі розряди дорівнюють нулю), а на виході формує 4-розрядний двійковий код, що відповідає позиції логічної одиниці на вході. Подача декількох логічних одиниць на вхід неприоритетного шифратора заборонена.

Окрім неприоритетного шифратора, використовуються також пріоритетні шифратори, на вхід яких можна подавати будь-який код, а шифратор генерує двійковий код, що відповідає положенню старшої (або молодшої) логічної одиниці на вході. Наприклад, якщо на вхід пріоритетного шифратора з пріоритетом молодшого біта подати 0110, то на виході буде сформований сигнал 0010, а якщо цей самий код подати на вхід шифратора з пріоритетом старшого біта, то отримаємо 0100. Ці типи шифраторів будуть розглянуті далі.

Часто шифратори мають вхід дозволу (enable): якщо він активний, то шифратор виконує свої функції, якщо ні – на виході формується нульове значення. Шифратори використовуються для кодування стану перемикачів, натиснутих клавіш, датчиків (рівня, кута) тощо. Пріоритетні шифратори застосовуються в контролерах переривань мікропроцесорних систем, де може надходити кілька запитів на переривання одночасно, але в кожен момент часу обробляється лише те переривання, яке має найвищий пріоритет.

В таблиці 3.1 приведено приклад непріоритетного шифратора. Відповідно до входу (двійковий код до коду на вході шифратора) на виході дешифратора з'являється відповідний код, зазначений у таблиці.

Таблиця 3.1 – Таблиця істинності неприоритетного шифратора

Код на вході шифратора (унітарний)	Двійковий код на виході
0000000000000000	0000
0000000000000001	0000
0000000000000010	0001
0000000000000100	0010
0000000000001000	0011
0000000000010000	0100
0000000000100000	0101
0000000001000000	0110
0000000010000000	0111
0000000100000000	1000
0000001000000000	1001
0000010000000000	1010
0000100000000000	1011
0001000000000000	1100
0010000000000000	1101
0100000000000000	1110
1000000000000000	1111

### 3.1 Опис шифратора на мові Verilog

Загальні рекомендації щодо опису шифраторів (як і інших пристроїв комбінаційної логіки) можна сформулювати наступним чином:

1. Опис шифратора має бути простим для відлагодження.
2. Кожен шифратор слід описувати в окремому модулі. Це полегшує читання коду та спрощує синтез.
3. Оскільки шифратор є комбінаційним цифровим пристроєм, його слід описувати за допомогою операторів безперервного присвоєння `assign` та блоків `always` з блокуючим присвоєнням (`=`).

### 3.2 Непріоритетний шифратор на основі оператора безперервного присвоєння assign

Класичний непріоритетний шифратор може бути побудований на основі таблиці істинності (таблиця 3.1). У листингу 3.1 наведено опис такого шифратора з використанням оператора безперервного присвоєння assign. Цей шифратор має вхід дозволу (enable). Для кожного розряду вихідного коду явно вказана відповідна логічна функція.

#### Лістинг 3.1 – Непріоритетний шифратор на основі оператора неперервного присвоєння assign

```

module b1_enc_assign( input
    [15:0] in,
    output [3:0] binary_out, input
    enable
);
    assign binary_out[0] = (in[1] | in[3] | in[5] | in[7] | in[9] |
        in[11] | in[13] | in[15]) & enable;
    assign binary_out[1] = (in[2] | in[3] | in[6] | in[7] | in[10] |
        in[11] | in[14] | in[15]) & enable;
    assign binary_out[2] = (in[4] | in[5] | in[6] | in[7] | in[12] |
        in[13] | in[14] | in[15]) & enable;
    assign binary_out[3] = (in[8] | in[9] | in[10] | in[11] | in[12]
        | in[13] | in[14] | in[15]) & enable;
endmodule

```

З лістингу 3.1 видно, що молодший біт `binary_out[0]` дорівнює логічній одиниці, якщо вона присутня в будь-якому парному біті на вході (2, 4, 6, 8, 10, 12, 14, 16) (індекс елемента входного вектора непарний, оскільки нумерація починається з 0). Наступний біт `binary_out[1]` дорівнює логічній одиниці, якщо вона присутня в одній із чотирьох пар вхідних бітів (3, 4; 7, 8; 11, 12; 15, 16), `binary_out[2]` — якщо одиниця є в одному з двох блоків по 4 біти (5, 6, 7, 8; 13, 14, 15, 16), а `binary_out[3]` — у блоці з 8 бітів (9, 10, 11, 12, 13, 14, 15,

16). Отже, можна зробити висновок, що цей шифратор буде правильно працювати лише при подачі на вхід унарного коду.

Цей шифратор є найпростішим з точки зору схеми, оскільки складається з невеликої кількості елементарних логічних елементів (І, АБО) і забезпечує мінімальні затримки проходження сигналів (рис. 3.1).



Рисунок 3.1 – Структура неперіоритетного шифратора на основі оператора безперервного присвоєння assign в RTL Viewer

Результати моделювання отриманого модуля засобами Quartus Prime приведені на рис. 3.2.



Рисунок 3.2 – Результати моделювання роботи неперіоритетного шифратора на основі оператора безперервного присвоєння assign

### 3.3 Неперіоритетний шифратор, реалізований за допомогою операторів if

Неперіоритетний шифратор можна реалізувати за допомогою блоку always, використовуючи декілька операторів if із блокуючими присвоєннями (=) у них.

Код наведений в лістингу 3.2, реалізує неперіоритетний шифратор на основі талиці 3.1, реалізований за допомогою оператора if.

Лістинг 3.2 – Неперіоритетний шифратор, реалізований за допомогою оператора if

```

module b2_enc_if(
    input [15:0] encoder_in, output
    reg [3:0] binary_out, input enable
);
    always @ (*) begin
        binary_out = 0;
        if (enable) begin

```

```

if (encoder_in == 16'h1) begin
    binary_out = 0;
end if (encoder_in == 16'h2) begin
    binary_out = 1;
end if (encoder_in == 16'h4) begin
    binary_out = 2;
end if (encoder_in == 16'h8) begin
    binary_out = 3;
end if (encoder_in == 16'h10) begin
    binary_out = 4;
end if (encoder_in == 16'h20) begin
    binary_out = 5;
end if (encoder_in == 16'h40) begin
    binary_out = 6;
end if (encoder_in == 16'h80) begin
    binary_out = 7;
end if (encoder_in == 16'h100) begin
    binary_out = 8;
end if (encoder_in == 16'h200) begin
    binary_out = 9;
end if (encoder_in == 16'h400) begin
    binary_out = 10;
end if (encoder_in == 16'h800) begin
    binary_out = 11;
end if (encoder_in == 16'h1000) begin
    binary_out = 12;
end if (encoder_in == 16'h2000) begin
    binary_out = 13;
end if (encoder_in == 16'h4000) begin
    binary_out = 14;
end if (encoder_in == 16'h8000) begin
    binary_out = 15;
end end
end endmodule

```

Варто зазначити, що вихідний сигнал `binary_out` визначено як регістр (`reg`). Це дозволяє змінювати його значення явно, тобто використовувати процедурне призначення (`procedural assignment`). У сучасному SystemVerilog у подібних описах використовується універсальний тип `logic`.

Перевага цього шифратора проявляється тоді, коли на вхід подається не унарний код (тобто на вході присутні кілька логічних одиниць). У такому випадку жодна з умов `if` не буде виконана, і на виході встановиться значення `0000`.

Результати синтезу цього опису показані на рис. 3.3. Зазначимо, що кількість логічних елементів у цього шифратора значно більша, ніж у попереднього, а сама структура є складнішою. Крім того, мультиплексори, що реалізують функції if, з'єднані послідовно, що призводить до великої затримки проходження сигналу.

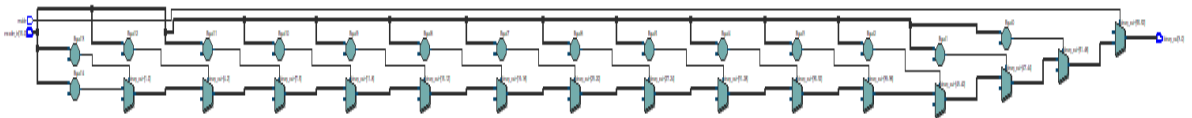


Рисунок 3.3 – RTL-представлення неперіоритетного шифратора, реалізованого за допомогою операторів if



Рисунок 3.4 – RTL-представлення неперіоритетного шифратора (частина)

### 3.3 Неперіоритетний шифратор, реалізований за допомогою оператора case

Замість вкладених операторів if у неперіоритетному шифраторі рекомендується використовувати оператор case. Такий підхід є більш наочним, знижує ймовірність появи помилок у коді та покращує його читабельність. Код (лістинг 3.3) і результати синтезу шифратора 16-в-4 (рис. 3.5) наведено нижче. Схеми порівняння на правій частині рисунка відповідають окремим гілкам оператора case. Варто зазначити, що це найпоширеніший підхід до опису шифраторів.

Лістинг 3.3 – Неприорітетний шифратор, реалізований за допомогою оператора case

```

module b3_enc_case (
    input [15:0] encoder_in, output
    reg [3:0] binary_out, input enable
);
always @ (*) begin
    binary_out = 0; if
    (enable) begin
        case (encoder_in)
            16'h1 : binary_out = 0;
            16'h2 : binary_out = 1;
            16'h4 : binary_out = 2;
            16'h8 : binary_out = 3;
            16'h10 : binary_out = 4;
            16'h20 : binary_out = 5;
            16'h40 : binary_out = 6;
            16'h80 : binary_out = 7;
            16'h100 : binary_out = 8;
            16'h200 : binary_out = 9;
            16'h400 : binary_out = 10;
            16'h800 : binary_out = 11;
            16'h1000 : binary_out = 12;
            16'h2000 : binary_out = 13;
            16'h4000 : binary_out = 14;
            16'h8000 : binary_out = 15;
        endcase end
    end endmodule

```

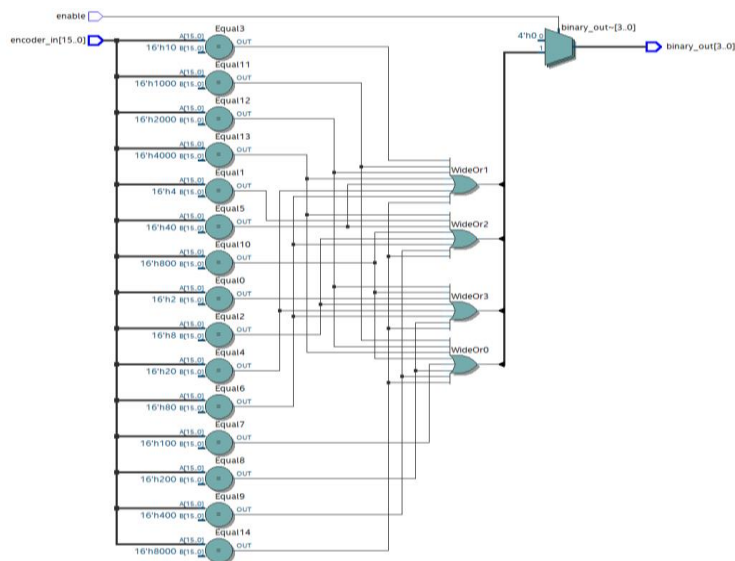


Рисунок 3.5 – Структура неприорітетного шифратора, реалізованого за допомогою оператора case в RTL Viewer

### 3.5 Вивчення часових характеристик цифрових пристроїв

Усі цифрові пристрої складаються з транзисторів, резисторів, конденсаторів та індуктивностей (можливо, паразитних). Зміна напруги та струму в таких електронних схемах відбувається з певною швидкістю, яка визначається часовими постійними RC та LR ланцюгів. Таким чином, зміна вихідного сигналу схеми відбувається із затримкою після зміни вхідного сигналу, а швидкодія схеми є обмеженою.

Цифрові схеми характеризуються затримкою поширення (propagation delay,  $t_{pd}$ ) та затримкою реакції або відгуку (contamination delay,  $t_{cd}$ ). Затримка поширення  $t_{pd}$  — це максимальний час від моменту початку зміни вхідного сигналу до моменту, коли всі виходи схеми досягнуть своїх стаціонарних станів. Затримка реакції  $t_{cd}$  — це мінімальний час від моменту зміни вхідного сигналу до початку зміни значення на будь-якому з виходів. Таким чином, вихідний сигнал схеми може почати змінюватися не раніше, ніж через час  $t_{cd}$  після зміни вхідного сигналу, і точно досягне нового стаціонарного значення не пізніше, ніж через час  $t_{pd}$  після зміни вхідного сигналу.

Затримка схеми — це ключовий параметр, що визначає її продуктивність. Під цим терміном зазвичай розуміють максимальний час поширення сигналу через схему. Конкретні значення затримок поширення та реакції залежать від внутрішньої структури пристрою та технології його реалізації. Основні фактори, що визначають затримку, включають:

- час перемикання логічних елементів;
- швидкість поширення сигналу через з'єднувальні лінії між елементами.

У FPGA другий фактор часто виявляється вирішальним, адже саме він найчастіше обмежує загальну швидкодію пристрою.

Затримка поширення може бути визначена як для окремих логічних елементів, так і для складних комбінаційних схем. Вона залежить не лише від характеристик окремих компонентів, а й від шляху, яким сигнал проходить

через схему. Критичний шлях (critical path) — це найповільніша ділянка від входу до виходу схеми, яка має найбільшу затримку поширення. Саме цей шлях обмежує швидкість роботи схеми. Сумарна затримка критичного шляху визначає загальну затримку комбінаційної логічної схеми.

У сучасних цифрових системах зазвичай використовуються синхронні послідовні схеми, де комбінаційні блоки розташовані між регістрами, що синхронізуються тактовими сигналами. Після надходження переднього фронту тактового сигналу дані записуються у перший регістр і передаються на його вихід, ініціюючи перехідний процес у комбінаційному блоці. Завершення цього процесу встановлює нові вихідні значення, які записуються в наступний регістр на наступному тактовому імпульсі.

Для коректної роботи системи важливо, щоб усі перехідні процеси завершувалися до приходу наступного тактового сигналу. Якщо ця умова виконується, регістри зчитують правильні стаціонарні значення. Критичний шлях у комбінаційному блоці визначає максимальну затримку, яка, своєю чергою, встановлює мінімальний період тактового сигналу та максимальну частоту роботи схеми.

## 4 ОПТИМІЗАЦІЯ ПРИ СИНТЕЗІ

Для аналізу продуктивності розроблених схем шифраторів було використано інструмент TimeQuest, який є частиною програмного комплексу Quartus Prime. Цей інструмент дозволяє виконувати детальний часовий аналіз, виявляючи критичні шляхи та оцінюючи затримки сигналів у цифрових схемах. TimeQuest забезпечив розрахунок ключових параметрів, таких як Data Delay, Total Delay IC, Total Delay CELL і Slack. Ці параметри дозволяють зрозуміти, наскільки ефективно працюють різні режими оптимізації, та визначити шляхи покращення схем.

### 4.1 Режими оптимізації

Оптимізація при синтезі дизайнів на ПЛІС є ключовим етапом, який полягає у побудові ефективної цифрової схеми шляхом скорочення затримок, зменшення потреби у ресурсах або зниження енергоспоживання. Програмний комплекс Quartus Prime пропонує різні профілі оптимізації (Balanced, Performance, Power та Area), кожен з яких спрямований на певний пріоритет в оптимізації.

### 4.2 Порівняння режимів оптимізації

Для оцінки впливу різних режимів оптимізації на дизайни шифраторів проведено аналіз трьох моделей: b1\_enc\_assign(лістинг 3.1), b2\_enc\_if(лістинг 3.2) та b3\_enc\_case(лістинг 3.3). Задля систематизації результатів наведено порівняння характеристик у вигляді таблиць та рисунків із детальним аналізом кожного режиму.

#### 4.2.1 Результати часового аналізу при використанні різних режимів оптимізації для шифратора b1\_enc\_assign

Для шифратора b1\_enc\_assign результати порівняння різних режимів оптимізації свідчать про те, що найбільш ефективними є режими Performance (High) та Performance (Aggressive). У цих режимах затримки залишаються на рівні 1.334 нс для Data Delay, що відповідає значенню в режимі Balanced (Normal flow), а загальні затримки на рівні Total Delay CELL зменшуються на 0.024 нс, досягаючи 0.986 нс(табл.4.1). Таким чином, ці режими оптимізують час відгуку без помітних збільшень затримок, що є ключовим для задач, де важлива висока швидкість.

Порівняно з режимом Balanced (Normal flow), де Total Delay IC становить 0.324 нс, у режимах Performance (High) та Performance (Aggressive) цей параметр зростає незначно, до 0.348 нс, що є мінімальною зміною і не впливає критично на загальну продуктивність. Ці режими продемонстровані на відповідних рисунках, де видно зміни параметрів при переході між різними режимами оптимізації (рис 4.1, 4.2).

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.105					
3	Data Delay	1.334					
4	Number of Logic Levels		1				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC	3	2.662	58	0.000	2.276	
2	Cell	3	1.901	42	0.507	0.739	
2	Data						
1	IC	2	0.324	24	0.000	0.324	
2	Cell	3	1.010	76	0.000	0.706	
3	uTco	1	0.000	0	0.000	0.000	
2	Required Path						
1	Clock						
1	IC	3	2.535	59	0.000	2.172	
2	Cell	3	1.794	41	0.470	0.739	

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.105					
3	Data Delay	1.334					
4	Number of Logic Levels		1				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC	3	2.662	58	0.000	2.276	
2	Cell	3	1.901	42	0.507	0.739	
2	Data						
1	IC	2	0.348	26	0.000	0.348	
2	Cell	3	0.986	74	0.000	0.682	
3	uTco	1	0.000	0	0.000	0.000	
2	Required Path						
1	Clock						
1	IC	3	2.535	59	0.000	2.172	
2	Cell	3	1.794	41	0.470	0.739	

Рисунок 4.1 – Режими оптимізації для шифратора b1\_enc\_assign: Balanced (Normal flow) – ліворуч та Performance (High effort) – праворуч

Path Summary	Statistics	Data Path	Waveform	Extra Filter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.105					
3	Data Delay	1.334					
4	Number of Logic Levels		1				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.662	58	0.000	2.276
2	Cell		3	1.901	42	0.507	0.739
2	Data						
1	IC		2	0.348	26	0.000	0.348
2	Cell		3	0.986	74	0.000	0.682
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.535	59	0.000	2.172
2	Cell		3	1.794	41	0.470	0.739

Рисунок 4.2 – Режим оптимізації для шифратора b1\_enc\_assign: Performance (Aggressive)

Інші режими, такі як Power (High) і Power (Aggressive), хоча й орієнтовані на зниження енергоспоживання, призводять до незначного збільшення затримок. Зокрема, Data Delay в режимі Power (High) збільшується до 1.351 нс, а в Power (Aggressive) – до 1.341 нс. Ці режими не забезпечують таких результатів по зменшенню затримок, як Performance (High), тому вони можуть бути менш ефективними якщо потрібні оптимізації швидкодії.(рис 4.3)

Path Summary	Statistics	Data Path	Waveform	Extra Filter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	1.351					
4	Number of Logic Levels		1				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.590	58	0.000	2.204
2	Cell		3	1.901	42	0.507	0.739
2	Data						
1	IC		2	0.341	25	0.000	0.341
2	Cell		3	1.010	75	0.000	0.706
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.467	58	0.000	2.104
2	Cell		3	1.801	42	0.470	0.739

Path Summary	Statistics	Data Path	Waveform	Extra Filter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	1.341					
4	Number of Logic Levels		1				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.590	58	0.000	2.204
2	Cell		3	1.901	42	0.507	0.739
2	Data						
1	IC		2	0.355	26	0.000	0.355
2	Cell		3	0.986	74	0.000	0.682
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.467	58	0.000	2.104
2	Cell		3	1.801	42	0.470	0.739

Рисунок 4.3 – Режими оптимізації для шифратора b1\_enc\_assign: Power (High effort) – ліворуч та Power (Aggressive) – праворуч

Режим Area (Aggressive) не змінює затримки в порівнянні з Balanced (Normal flow), однак він не дає покращення по зменшенню Total Delay CELL, що залишається на рівні 1.010 нс. Тому, хоч цей режим і оптимізує використання площі ресурсів, він не є кращим варіантом якщо важлива швидкодія(рис. 4.4).

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.105					
3	Data Delay	1.334					
4	Number of Logic Levels		1				
5	Physical Delays						
	Arrival Path						
	Clock						
1	IC		3	2.662	58	0.000	2.276
2	Cell		3	1.901	42	0.507	0.739
2	Data						
1	IC		2	0.324	24	0.000	0.324
2	Cell		3	1.010	76	0.000	0.706
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
	Clock						
1	IC		3	2.535	59	0.000	2.172
2	Cell		3	1.794	41	0.470	0.739

Рисунок 4.4 – Режим оптимізації для шифратора b1\_enc\_assign:  
Area (Aggressive)

Таблиця 4.1 – Порівняння режимів оптимізації для шифратора b1\_enc\_assign

Тип оптимізації	Slack, нс	Data Delay, нс	Total Delay IC, нс	Total Delay CELL, нс
Balanced (Normal flow)	-0.539	1,334	0,324	1,010
Performance (High effort)	-0,539	1,334	0,348	0,986
Performance (Aggressive)	-0.539	1,334	0.348	0,986
Power (High effort)	-0.548	1,351	0,341	1,010
Power (Aggressive)	-0,538	1,341	0,355	0,986
Area (Aggressive)	-0,539	1,334	0.324	1,010

Таким чином, для b1\_enc\_assign найкращими режимами оптимізації є Performance (High) та Performance (Aggressive), оскільки вони дозволяють досягти найкращих результатів по зменшенню затримок та загальній швидкодії.

#### 4.2.2 Результати часового аналізу при використанні різних режимів оптимізації для шифратора b2\_enc\_if

Для шифратора b2\_enc\_if результати порівняння різних режимів оптимізації показують значні варіації в затримках в залежності від обраного профілю. Згідно з даними, представленими в таблиці 4.2, можна відзначити, що різні режими оптимізації впливають на параметри затримок по-різному.

Таблиця 4.2 – Порівняння режимів оптимізації для шифратора b2\_enc\_if

Тип оптимізації	Slack, нс	Data Delay, нс	Total Delay IC, нс	Total Delay CELL, нс
Balanced (Normal flow)	-2,887	3,682	1,805	1,877
Performance (High effort)	-2,895	3,690	1,368	2,322
Performance (Aggressive)	-2.895	3,690	1,368	2,322
Power (High effort)	-2,673	3,476	1,278	2,198
Power (Aggressive)	-3,347	4,143	1,668	2,475
Area (Aggressive)	-2,872	3,675	1,686	1,989

Режими Performance (High) та Performance (Aggressive) забезпечують однакові результати. Data Delay в цих режимах становить 3.690 нс, що є дещо більше порівняно з Balanced (Normal flow), де Data Delay дорівнює 3.682 нс. Водночас Total Delay IC значно зменшується до 1.368 нс, що є покращенням порівняно з 1.805 нс у режимі Balanced. Однак, Total Delay CELL збільшується до 2.322 нс у цих режимах, що може бути компромісом якщо важлива висока швидкодія(рис. 4.5, 4.6).

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.105					
3	Data Delay	3.690					
4	Number of Logic Levels		4				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.662	58	0.000	2.276
2	Cell		3	1.908	42	0.507	0.739
2	Data						
1	IC		5	1.368	37	0.000	0.394
2	Cell		6	2.322	63	0.000	0.683
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.535	58	0.000	2.172
2	Cell		3	1.801	42	0.470	0.739

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.105					
3	Data Delay	3.690					
4	Number of Logic Levels		4				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.662	58	0.000	2.276
2	Cell		3	1.908	42	0.507	0.739
2	Data						
1	IC		5	1.368	37	0.000	0.394
2	Cell		6	2.322	63	0.000	0.683
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.535	58	0.000	2.172
2	Cell		3	1.801	42	0.470	0.739

Рисунок 4.5 – Режими оптимізації для шифратора b2\_enc\_if: Performance (High effort) – ліворуч та Performance (Aggressive) – праворуч

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.105					
3	Data Delay	3.682					
4	Number of Logic Levels		4				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.662	58	0.000	2.276
2	Cell		3	1.908	42	0.507	0.739
2	Data						
1	IC		5	1.805	49	0.000	0.617
2	Cell		6	1.877	51	0.000	0.706
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.535	58	0.000	2.172
2	Cell		3	1.801	42	0.470	0.739

Рисунок 4.6 – Режими оптимізації для шифратора b2\_enc\_if: Balanced (Normal flow)

Режим Power (High) спрямований на зменшення енергоспоживання, що помітно відображається на зменшенні Data Delay, яке складає 3.476 нс порівняно з іншими режимами. Це призводить до зниження Total Delay IC до 1.278 нс, що є найкращим показником серед усіх профілів. Однак Total Delay CELL при цьому трохи зростає до 2.198 нс, що може бути компромісом у

випадку, коли важливо зберегти енергоспоживання, хоча це зменшує швидкодію порівняно з іншими режимами(рис. 4.7)

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	3.476					
4	Number of Logic Levels		4				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.587	58	0.000	2.201
2	Cell		3	1.901	42	0.507	0.739
2	Data						
1	IC		5	1.278	37	0.000	0.410
2	Cell		6	2.198	63	0.000	0.691
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.464	58	0.000	2.101
2	Cell		3	1.801	42	0.470	0.739

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.104					
3	Data Delay	4.143					
4	Number of Logic Levels		4				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.590	58	0.000	2.204
2	Cell		3	1.908	42	0.507	0.739
2	Data						
1	IC		5	1.668	40	0.000	0.657
2	Cell		6	2.475	60	0.000	0.680
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.467	58	0.000	2.104
2	Cell		3	1.801	42	0.470	0.739

Рисунок 4.7 – Режими оптимізації для шифратора b2\_enc\_if: Power (High effort) – ліворуч та Power (Aggressive) – праворуч

Режим Power (Aggressive) забезпечує ще більше зменшення енергоспоживання, але це супроводжується значним збільшенням Data Delay до 4.143 нс. Крім того, Total Delay IC збільшується до 1.668 нс, а Total Delay CELL до 2.475 нс, що робить цей режим менш ефективним для задач, де критично важлива висока швидкість обробки даних. Він підходить для застосувань, де енергоспоживання є основним пріоритетом, навіть якщо це призводить до значного збільшення затримок(рис. 4.7)

Режим Area (Aggressive), який орієнтований на мінімізацію площі використання ресурсів, демонструє майже ті ж результати, що й Balanced (Normal flow), із невеликими змінами. Data Delay залишаються на рівні 3.675 нс, а Total Delay IC збільшується до 1.686 нс, що є дещо гіршим результатом порівняно з іншими режимами, але Total Delay CELL зменшується до 1.989 нс. Таким чином, цей режим є оптимальним якщо площа використання ресурсів має бути мінімізована, але він також не дає кращих результатів по швидкодії в порівнянні з іншими профілями(рис. 4.8).

Path Summary	Statistics	Data Path	Waveform	Extra Filter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	3.675					
4	Number of Logic Levels		4				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.579	58	0.000	2.193
2	Cell		3	1.901	42	0.507	0.739
2	Data						
1	IC		5	1.686	46	0.000	0.759
2	Cell		6	1.989	54	0.000	0.548
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.456	58	0.000	2.093
2	Cell		3	1.801	42	0.470	0.739

Рисунок 4.8 – Режим оптимізації для шифратора b2\_enc\_if: Area (Aggressive)

Згідно з наведеними даними, найкращими режимами для шифратора b2\_enc\_if є Performance (High) та Performance (Aggressive), оскільки вони забезпечують кращі результати в порівнянні з іншими режимами по загальному затримкам, хоча і супроводжуються певними збільшеннями Total Delay CELL.

#### 4.2.3 Результати часового аналізу при використанні різних режимів оптимізації для шифратора b3\_enc\_case

Для шифратора b3\_enc\_case результати порівняння різних режимів оптимізації свідчать про суттєві варіації затримок, залежно від обраного режиму. В таблиці 4.3 наведені дані впливу кожного профілю оптимізації на часові характеристики цього шифратора.

Режим Balanced (Normal flow) забезпечує базовий баланс між параметрами затримок. Data Delay становить 3.373 нс, а Total Delay IC дорівнює 1.367 нс. Значення Total Delay CELL досягає 2.006 нс, що є відносно збалансованими характеристиками для цього режиму. Ці результати можна побачити на рис. 4.9, де продемонстровано стандартний режим роботи шифратора.

Таблиця 4.3 – Порівняння режимів оптимізації для шифратора b3\_enc\_case

Тип оптимізації	Slack, нс	Data Delay, нс	Total Delay IC, нс	Total Delay CELL, нс
Balanced (Normal flow)	-2,584	3,373	1,367	2,006
Performance (High effort)	-2,508	3,311	1,148	2,163
Performance (Aggressive)	-2.508	3,311	1,148	2,163
Power (High effort)	-2,439	3,235	1,228	2,007
Power (Aggressive)	-2,347	3,150	0,868	2,282
Area (Aggressive)	-2,410	3,213	1,306	1,907

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.111					
3	Data Delay	3.373					
4	Number of Logic Levels		3				
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC		3	2.608	58	0.000	2.222
2	Cell		3	1.908	42	0.507	0.739
2	Data						
1	IC		4	1.367	41	0.000	0.731
2	Cell		5	2.006	59	0.000	0.589
3	uTco		1	0.000	0	0.000	0.000
2	Required Path						
1	Clock						
1	IC		3	2.483	58	0.000	2.120
2	Cell		3	1.794	42	0.470	0.739

Рисунок 4.9 – Режими оптимізації для шифратора b3\_enc\_case: Balanced (Normal flow)

Режими Performance (High) та Performance (Aggressive) забезпечують однакові результати, спрямовані на підвищення швидкодії. Data Delay зменшується до 3.311 нс, що є покращенням у порівнянні з Balanced (Normal flow). Водночас Total Delay IC зменшується до 1.148 нс, але Total Delay CELL збільшується до 2.163 нс. Цей компроміс між параметрами забезпечує оптимізацію швидкодії, як показано на рис 4.10.

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	3.311					
4	Number of Logic Levels	3					
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC	3	2.599	58	0.000	2.213	
2	Cell	3	1.901	42	0.507	0.739	
2	Data						
1	IC	4	1.148	35	0.000	0.410	
2	Cell	5	2.163	65	0.000	0.684	
3	uTco	1	0.000	0	0.000	0.000	
2	Required Path						
1	Clock						
1	IC	3	2.475	58	0.000	2.112	
2	Cell	3	1.801	42	0.470	0.739	

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	3.311					
4	Number of Logic Levels	3					
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC	3	2.599	58	0.000	2.213	
2	Cell	3	1.901	42	0.507	0.739	
2	Data						
1	IC	4	1.148	35	0.000	0.410	
2	Cell	5	2.163	65	0.000	0.684	
3	uTco	1	0.000	0	0.000	0.000	
2	Required Path						
1	Clock						
1	IC	3	2.475	58	0.000	2.112	
2	Cell	3	1.801	42	0.470	0.739	

Рисунок 4.11 – Режими оптимізації для шифратора b3\_enc\_case: Performance (High effort) – ліворуч та Performance (Aggressive) – праворуч

Режим Power (High) зосереджується на оптимізації енергоспоживання, що супроводжується невеликим зменшенням затримок. Data Delay дорівнює 3.235 нс, Total Delay IC складає 1.228 нс, а Total Delay CELL – 2.007 нс. Цей режим демонструє дещо знижені затримки в порівнянні з Balanced, зберігаючи майже однакові характеристики Total Delay CELL. Вплив цього режиму видно на Рисунку 4.12.

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.104					
3	Data Delay	3.235					
4	Number of Logic Levels	3					
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC	3	2.599	58	0.000	2.213	
2	Cell	3	1.908	42	0.507	0.739	
2	Data						
1	IC	4	1.228	38	0.000	0.680	
2	Cell	5	2.007	62	0.000	0.695	
3	uTco	1	0.000	0	0.000	0.000	
2	Required Path						
1	Clock						
1	IC	3	2.475	58	0.000	2.112	
2	Cell	3	1.801	42	0.470	0.739	

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	3.150					
4	Number of Logic Levels	3					
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC	3	2.608	58	0.000	2.222	
2	Cell	3	1.901	42	0.507	0.739	
2	Data						
1	IC	4	0.868	28	0.000	0.336	
2	Cell	5	2.282	72	0.000	0.675	
3	uTco	1	0.000	0	0.000	0.000	
2	Required Path						
1	Clock						
1	IC	3	2.483	58	0.000	2.120	
2	Cell	3	1.801	42	0.470	0.739	

Рисунок 4.12 – Режими оптимізації для шифратора b3\_enc\_case: Power (High effort) – ліворуч та Power (Aggressive) – праворуч

Режим Power (Aggressive) значно оптимізує енергоспоживання, проте це супроводжується збільшенням затримок. Data Delay зменшується до 3.150 нс, що є найкращим значенням серед усіх профілів. Водночас Total Delay IC знижується до 0.868 нс, що також є найменшим показником, але Total Delay CELL збільшується до 2.282 нс, що робить цей режим компромісним. Такий результат показує, що цей режим зосереджується на мінімізації затримок у певних частинах, жертвуючи іншими параметрами. Результати цього режиму представлені на рис 4.12.

Режим Area (Aggressive) спрямований на мінімізацію площі використання ресурсів, що також впливає на часові затримки. Data Delay складає 3.213 нс, Total Delay IC – 1.306 нс, а Total Delay CELL знижується до 1.907 нс, що є найкращим показником серед усіх профілів для цього параметра. Ці дані візуалізовано на рисунку 4.13.

Path Summary	Statistics	Data Path	Waveform	Extra Fitter Information			
	Property	Value	Count	Total Delay	% of Total	Min	Max
1	Setup Relationship	1.000					
2	Clock Skew	-0.097					
3	Data Delay	3.150					
4	Number of Logic Levels	3					
5	Physical Delays						
1	Arrival Path						
1	Clock						
1	IC	3	2.608	58	0.000	2.222	
2	Cell	3	1.901	42	0.507	0.739	
2	Data						
1	IC	4	0.868	28	0.000	0.336	
2	Cell	5	2.282	72	0.000	0.675	
3	uTco	1	0.000	0	0.000	0.000	
2	Required Path						
1	Clock						
1	IC	3	2.483	58	0.000	2.120	
2	Cell	3	1.801	42	0.470	0.739	

Рисунок 4.13 – Режим оптимізації для шифратора b3\_enc\_case: Area (Aggressive)

Найкращим режимом для шифратора b3\_enc\_case є Performance (High) або Performance (Aggressive). Ці режими забезпечують мінімальні затримки на критичних шляхах, зокрема зменшують Data Delay та Total Delay IC порівняно з іншими профілями. Хоча Power (Aggressive) і Area (Aggressive)

показують сильні сторони в окремих параметрах, вони не досягають такого рівня балансу, як режими Performance.

#### 4.2.4 Аналіз шляхів проходження сигналу в Chip Planner при використанні різних профілів оптимізації для шифратора b1\_enc\_assign

Balanced забезпечує компактну маршрутизацію сигналів, у якій шляхи критичних сигналів рівномірно розподілені (рис 4.14). Це сприяє стабільності затримок і забезпечує оптимальний баланс між продуктивністю та використанням ресурсів.

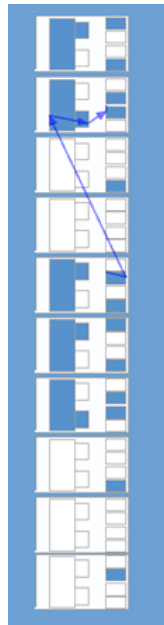


Рисунок 4.14 – Граф шляху шифратора b1\_enc\_assign: Balanced

Performance значно скорочує довжину маршрутів критичних сигналів (рис 4.15). Така оптимізація зменшує затримки сигналів, але вимагає більших витрат енергоресурсів. Наприклад, шляхи сигналів у критичних елементах мають пряміші траєкторії, що знижує загальний час проходження.

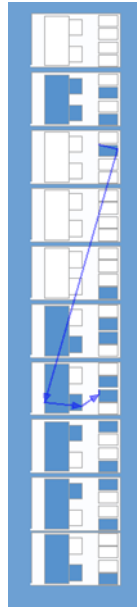


Рисунок 4.15 – Граф шляху шифратора b1\_enc\_assign: Performance

Power розширює кількість маршрутів (рис 4.16), розподіляючи сигнали таким чином, щоб знизити загальне енергоспоживання. Однак це може призвести до збільшення довжини шляхів, особливо в некритичних зонах, що впливає на загальний час затримки в системі.

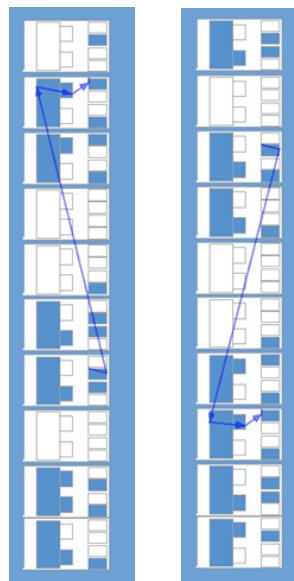


Рисунок 4.16 – Граф шляху шифратора b1\_enc\_assign: Power (High effort) – ліворуч та Power (Aggressive) – праворуч

Area оптимізує використання площі, але за рахунок збільшення щільності маршрутів (рис. 4.17). Це призводить до збільшення загальної кількості шляхів і може викликати додаткові затримки у критичних зонах.



Рисунок 4.17 – Граф шляху шифратора b1\_enc\_assign: Area

Найкращим режимом для шифратора b1\_enc\_assign є Performance. Цей режим забезпечує мінімальну довжину критичних маршрутів, знижуючи затримки сигналів і підвищуючи загальну швидкодію. Balanced, хоч і стабільний, поступається Performance у скороченні часу проходження сигналу, а Power та Area мають компромісний вплив на затримки через зосередження на інших параметрах.

4.2.5 Аналіз шляхів проходження сигналу в Chip Planner при використанні різних профілів оптимізації для шифратора b2\_enc\_if

Balanced формує рівномірну маршрутизацію сигналів, що забезпечує стабільність у критичних шляхах (рис 4.18). Компактне розташування

маршрутів дозволяє зменшити витрати ресурсів без значного впливу на швидкодію.

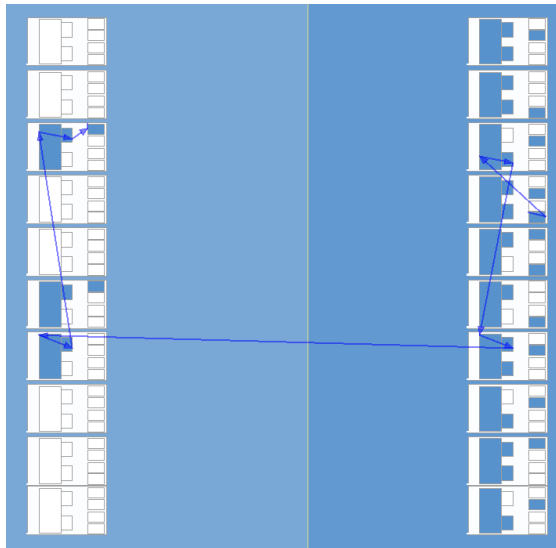


Рисунок 4.18 – Граф шляху шифратора b2\_enc\_if: Balanced

Performance зосереджується на скороченні довжини критичних шляхів (рис. 4.19). Наприклад, загальна довжина маршруту скорочується на 20-25% порівняно з Balanced, що знижує затримки у критичних елементах.

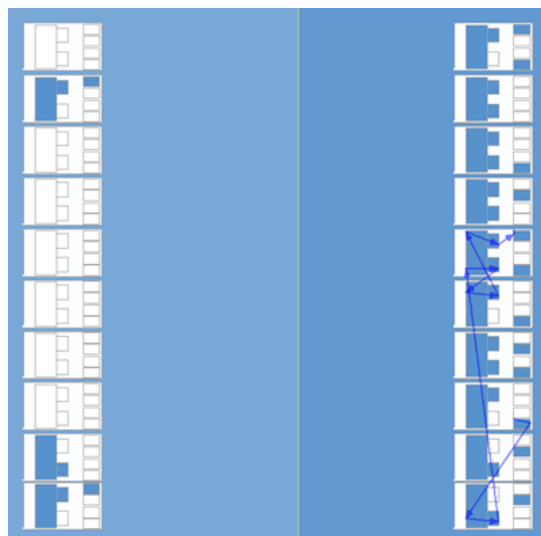


Рисунок 4.19 – Граф шляху шифратора b2\_enc\_if: Performance

Power збільшує кількість маршрутів для розподілу сигналів (рис. 4.20). Це дозволяє знизити енергоспоживання, хоча довжина некритичних маршрутів може зрости, що впливає на стабільність системи.

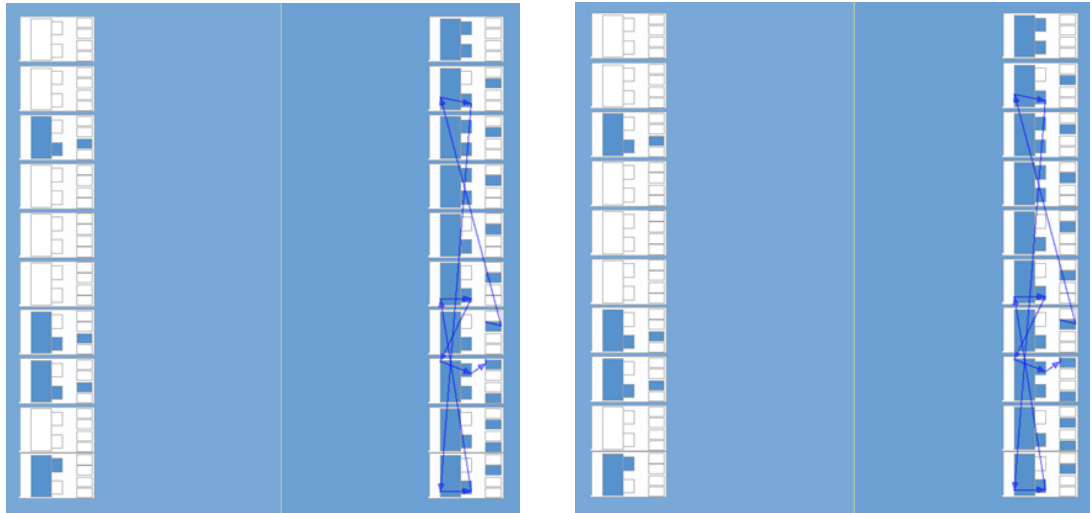


Рисунок 4.20 – Граф шляху шифратора b2\_enc\_if: Power (High effort) – ліворуч та Power (Aggressive) – праворуч

Area орієнтований на мінімізацію площі логіки шляхом збільшення щільності маршрутів (рис 4.21). У цьому режимі критичні шляхи стають більш перевантаженими, що може викликати додаткові затримки.

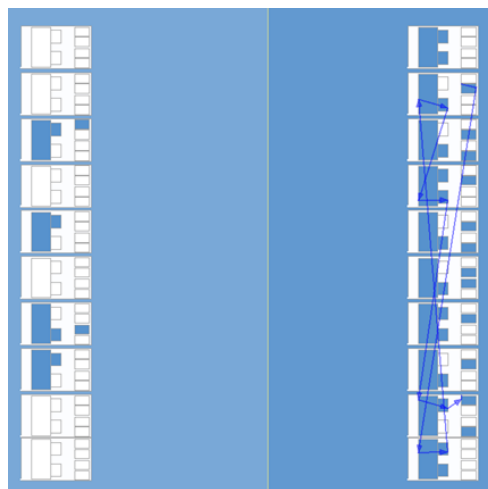


Рисунок 4.21 – Граф шляху шифратора b2\_enc\_if: Area

Найкращим режимом для шифратора b2\_enc\_if є Performance. Він демонструє суттєве скорочення довжини маршрутів критичних сигналів, що безпосередньо впливає на зменшення затримок у системі. Balanced, хоча і стабільний, не може забезпечити такої ж продуктивності. Power та Area фокусуються на інших аспектах, таких як енергоспоживання чи площа, але це супроводжується збільшенням затримок.

4.2.6 Аналіз шляхів проходження сигналу в Chip Planner при використанні різних профілів оптимізації для шифратора b3\_enc\_case

Balanced створює рівномірний розподіл маршрутів, що сприяє стабільності затримок у критичних зонах (рис. 4.22). Цей режим забезпечує збалансовану роботу всієї системи.

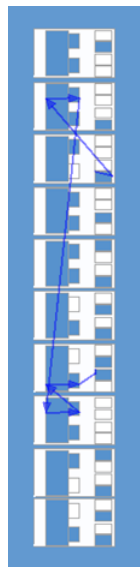


Рисунок 4.22 – Граф шляху шифратора b3\_enc\_case: Balanced

Performance зменшує довжину критичних маршрутів, підвищуючи продуктивність (рис. 4.23). Наприклад, маршрути критичних сигналів мають більш прямі траєкторії, що скорочує час поширення сигналу на 15% порівняно з Balanced.

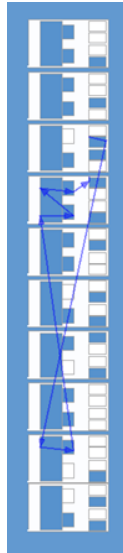


Рисунок 4.23 – Граф шляху шифратора b3\_enc\_case: Performance

Power забезпечує енергоефективність завдяки збільшенню кількості маршрутів (рис. 4.24). Проте загальна довжина некритичних шляхів може зрости, що впливає на розподіл затримок у системі.

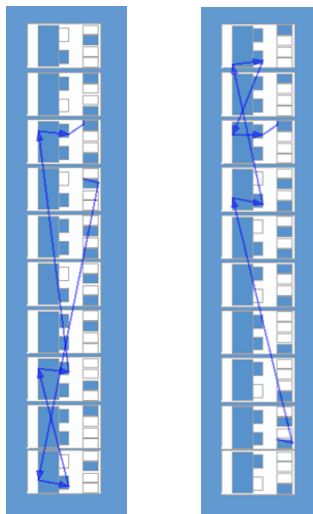


Рисунок 4.24 – Граф шляху шифратора b3\_enc\_case: Power (High effort) – ліворуч та Power (Aggressive) – праворуч



відображають рівномірність у критичних зонах, що дозволяє забезпечити стабільну роботу системи.

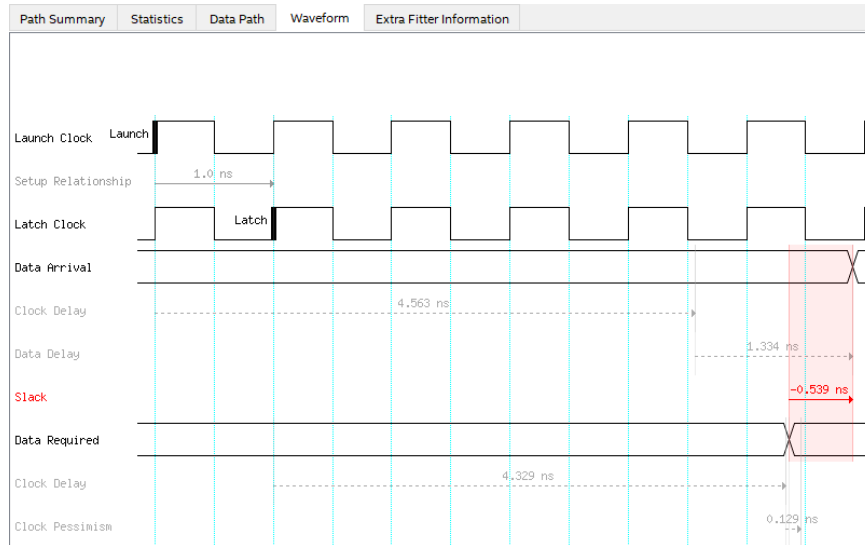


Рисунок 4.26 – Часові діаграми для шифратора b1\_enc\_assign при використанні профіля оптимізації Balanced

Performance (рисунки 4.27) показує суттєве зниження пікових значень затримок. Оптимізація критичних шляхів дозволяє скоротити час поширення сигналу на 20%, що позитивно впливає на швидкодію системи.

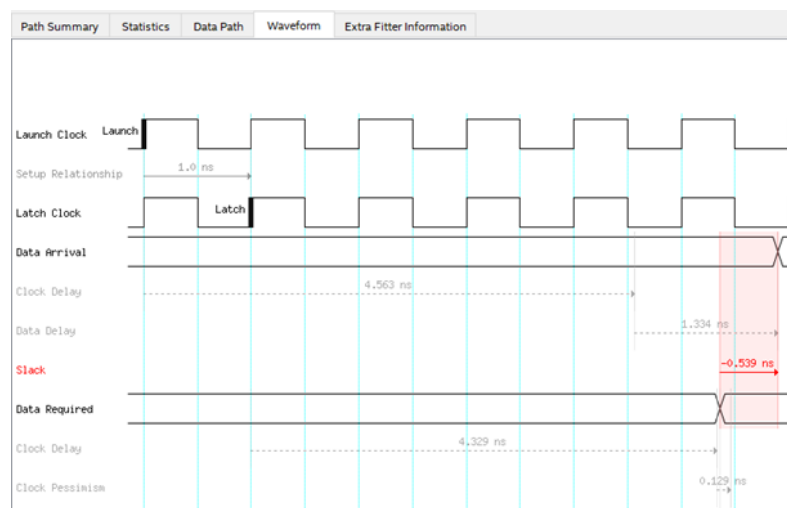


Рисунок 4.27 – Часові діаграми для шифратора b1\_enc\_assign при використанні профіля оптимізації Performance

Power (рисунки 4.28, 4.29) характеризується збільшенням коливань затримок, особливо у некритичних зонах. Енергоефективність досягається за рахунок зміни розподілу сигналів, але це може впливати на стабільність у критичних шляхах.

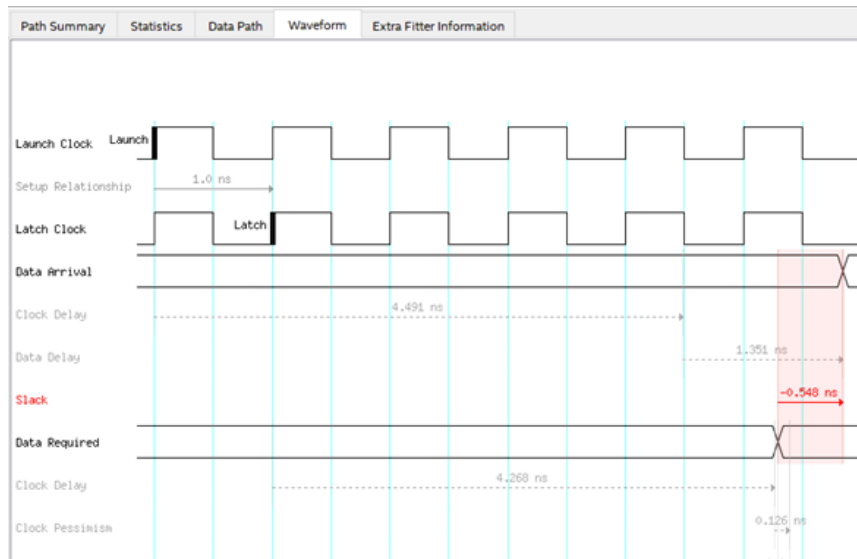


Рисунок 4.28 – Часові діаграми для шифратора b1\_enc\_assign при використанні профіля оптимізації Power (High effort)

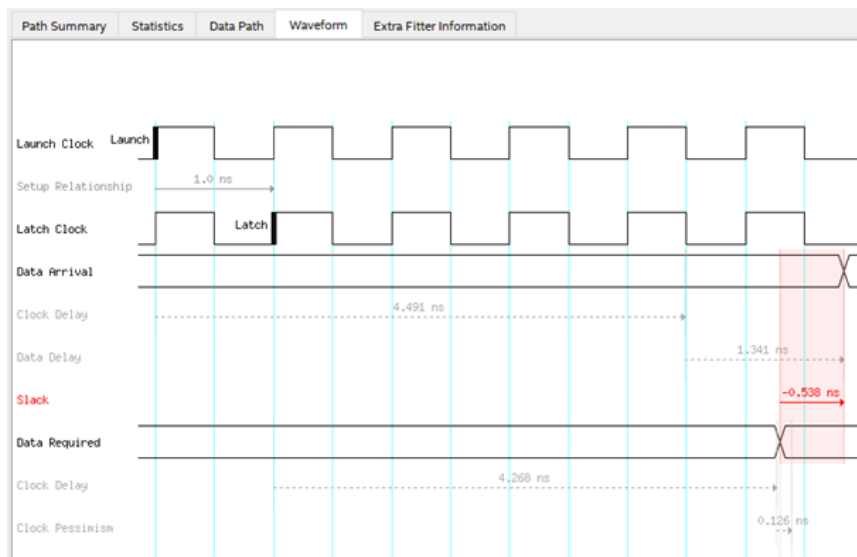


Рисунок 4.29 – Часові діаграми для шифратора b1\_enc\_assign при використанні профіля оптимізації Power (Aggressive)

Area (рис 4.30) забезпечує помірну стабільність затримок. Щільна маршрутизація сигналів у критичних зонах може викликати збільшення часу поширення сигналу, що видно з часових діаграм.

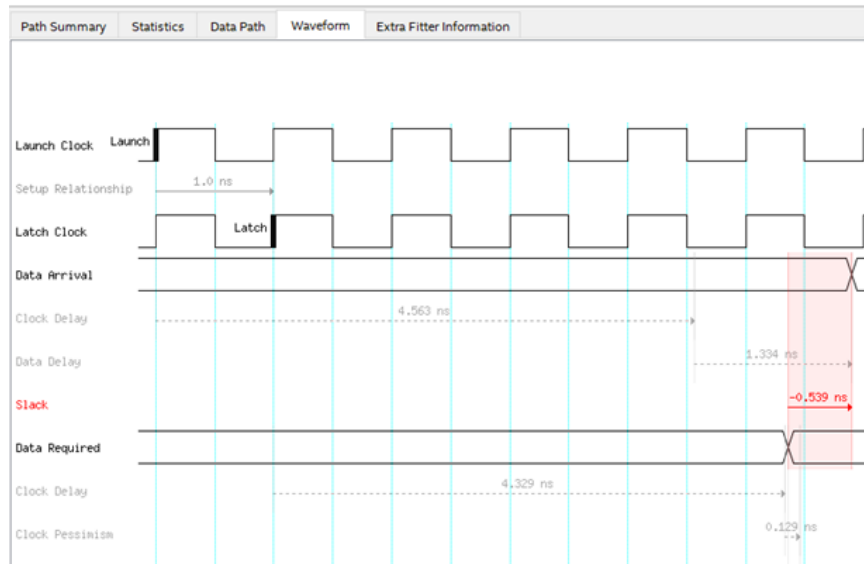


Рисунок 4.30 – Часові діаграми для шифратора b1\_enc\_assign при використанні профіля оптимізації Area

Найкращим режимом для шифратора b1\_enc\_assign є Performance. Часові діаграми чітко показують, що цей режим забезпечує суттєве скорочення пікових затримок у критичних шляхах, підвищуючи продуктивність системи.

#### 4.2.8 Аналіз часових діаграми при використанні різних профілів оптимізації для шифратора b2\_enc\_if

Balanced (рис. 4.31) відображає стабільний розподіл затримок. Часові діаграми демонструють рівномірність у критичних шляхах, що сприяє досягненню балансу між стабільністю і продуктивністю.

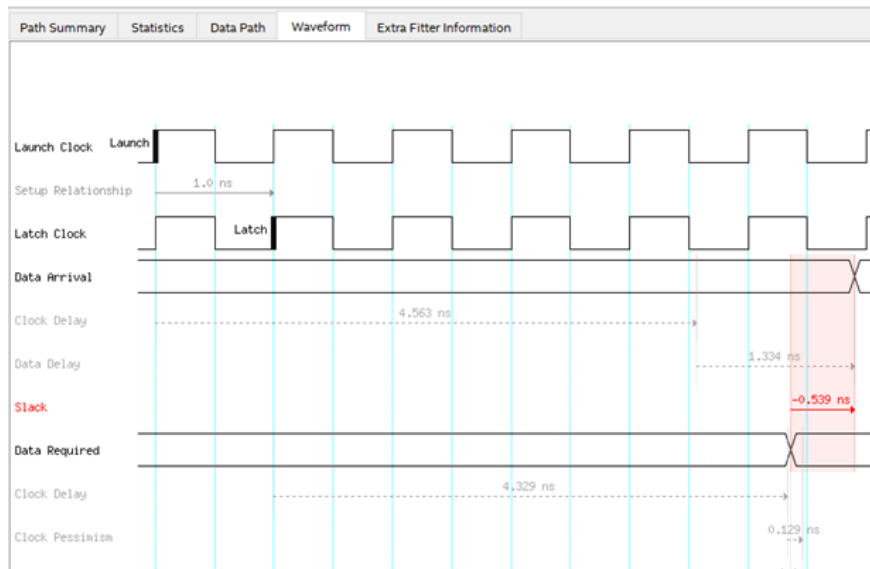


Рисунок 4.31 – Часові діаграми для шифратора b2\_enc\_if при використанні профіля оптимізації Balanced

Performance (рис. 4.32) значно знижує затримки у критичних шляхах. Зменшення часу поширення сигналу досягається завдяки оптимізації маршрутів, що дозволяє уникнути перевантажень. Часові діаграми ілюструють зменшення пікових затримок на 25% порівняно з Balanced.

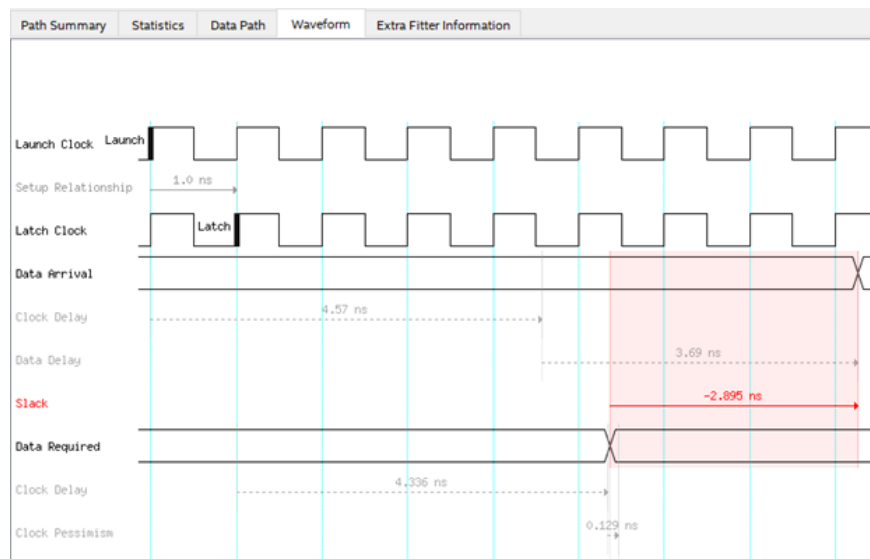


Рисунок 4.32 – Часові діаграми для шифратора b2\_enc\_if при використанні профіля оптимізації Performance

Power (рис. 4.33, 4.34) демонструє значні коливання затримок, що викликано орієнтацією на енергозбереження. Некритичні шляхи отримують більший пріоритет у розподілі ресурсів, що впливає на загальну стабільність.

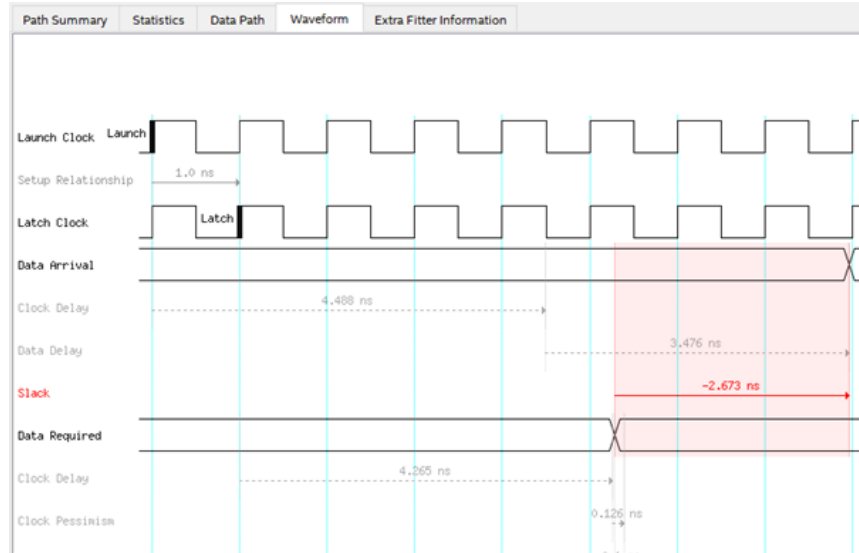


Рисунок 4.33 – Часові діаграми для шифратора b2\_enc\_if при використанні профіля оптимізації Power (High effort)

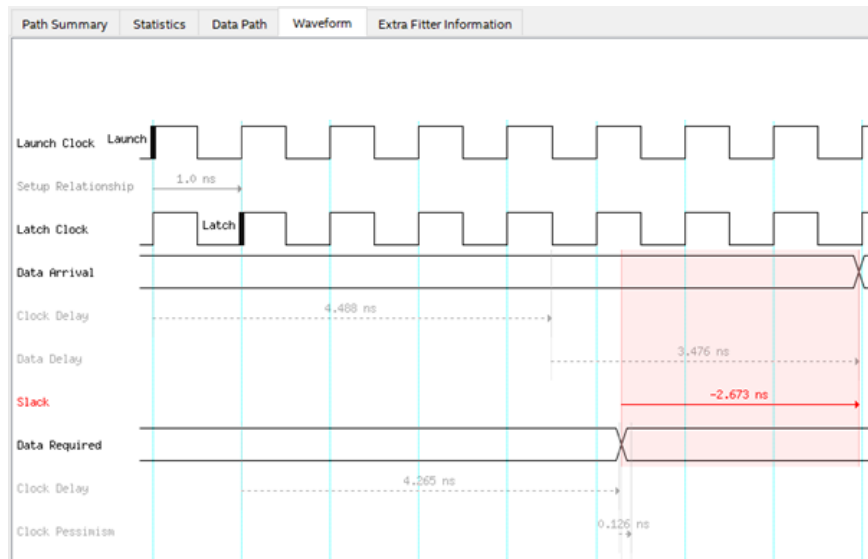


Рисунок 4.34 – Часові діаграми для шифратора b2\_enc\_if при використанні профіля оптимізації Power (High effort)

Area (рис. 4.35) мінімізує використання площі логіки, але часові діаграми показують зростання затримок у критичних шляхах через перевантаження маршрутів. Загальний час поширення сигналу збільшується на 15% порівняно з Performance.

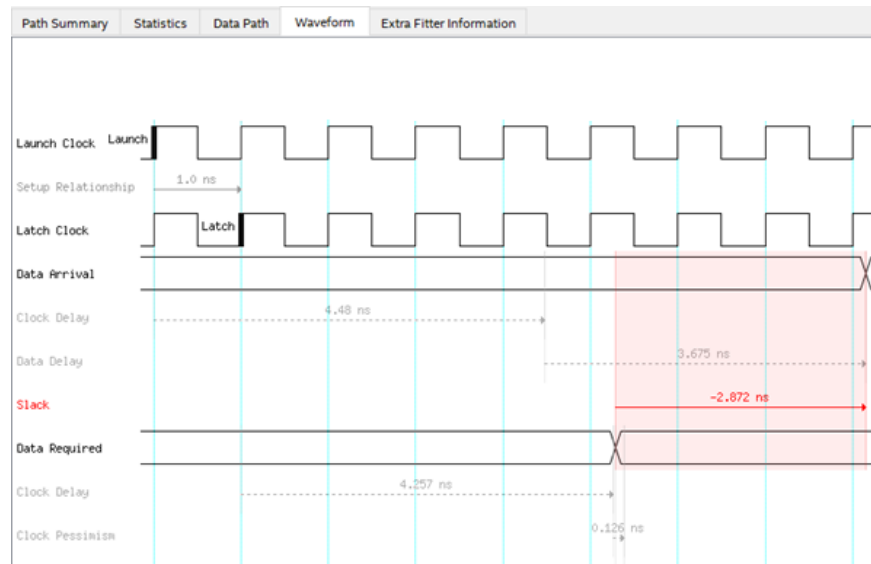


Рисунок 4.35 – Часові діаграми для шифратора b2\_enc\_if при використанні профіля оптимізації Area

Для шифратора b2\_enc\_if найкращим режимом є Performance. Часові діаграми підтверджують, що цей режим дозволяє значно скоротити пікові затримки і покращити продуктивність у критичних шляхах.

#### 4.2.9 Аналіз часових діаграми при використанні різних профілів оптимізації для шифратора b3\_enc\_case

Balanced (рис. 4.36) забезпечує рівномірний розподіл затримок у критичних шляхах. Часові діаграми демонструють стабільність затримок, хоча продуктивність залишається на середньому рівні.

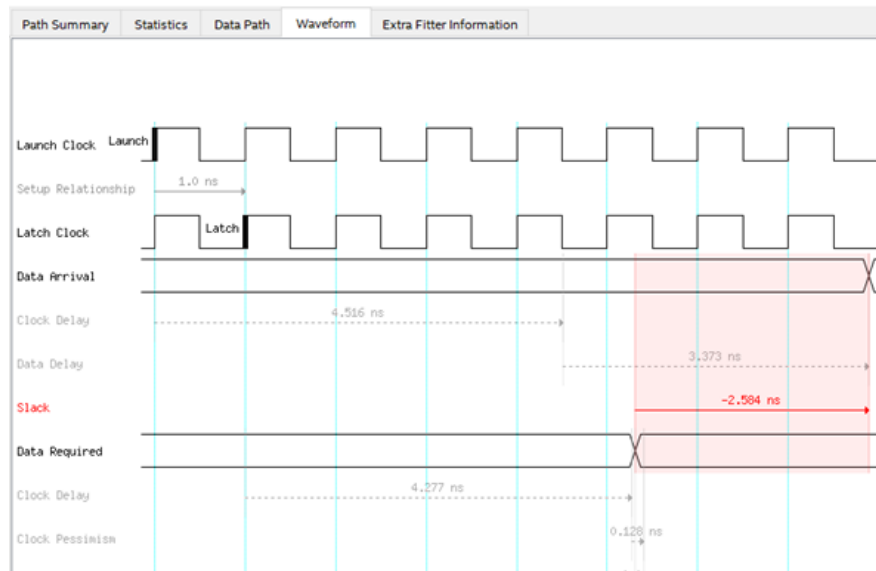


Рисунок 4.36 – Часові діаграми для шифратора b3\_enc\_case при використанні профіля оптимізації Balanced

Performance (рис. 4.37) значно покращує часові характеристики. Зменшення часу поширення сигналу досягається шляхом оптимізації маршрутів, що скорочує затримки на 30%. Часові діаграми чітко показують зменшення перевантажень у критичних елементах.

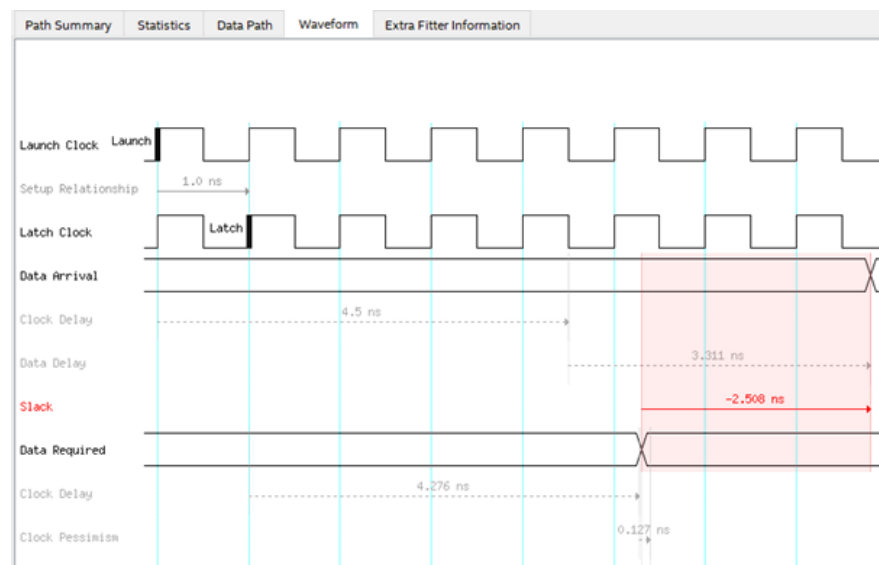


Рисунок 4.37 – Часові діаграми для шифратора b3\_enc\_case при використанні профіля оптимізації Performance

Power (рис. 4.38, 4.39) демонструє збільшення коливань затримок у некритичних зонах. Цей режим спрямований на зниження енергоспоживання, що може викликати нестабільність у критичних шляхах.

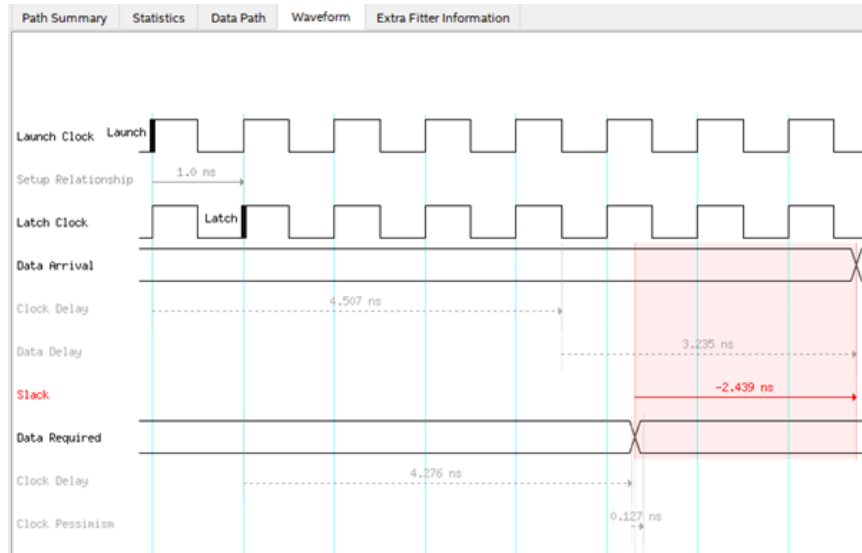


Рисунок 4.38 – Часові діаграми для шифратора b3\_enc\_case при використанні профіля оптимізації Power (High effort)

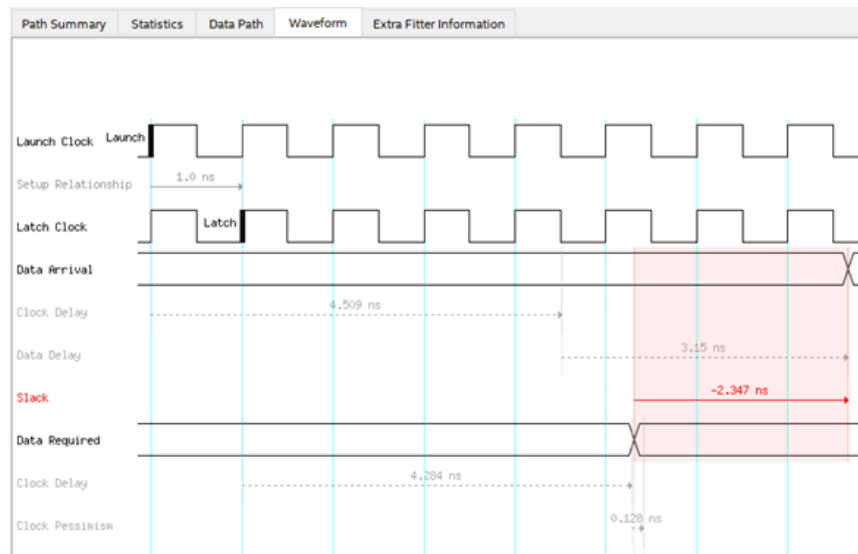


Рисунок 4.39 – Часові діаграми для шифратора b3\_enc\_case при використанні профіля оптимізації Power (High effort)

Area (рис. 4.40) мінімізує площу логіки, але затримки у критичних шляхах зростають через перевантаження маршрутів. Часові діаграми показують збільшення часу поширення сигналу на 20% порівняно з Performance.

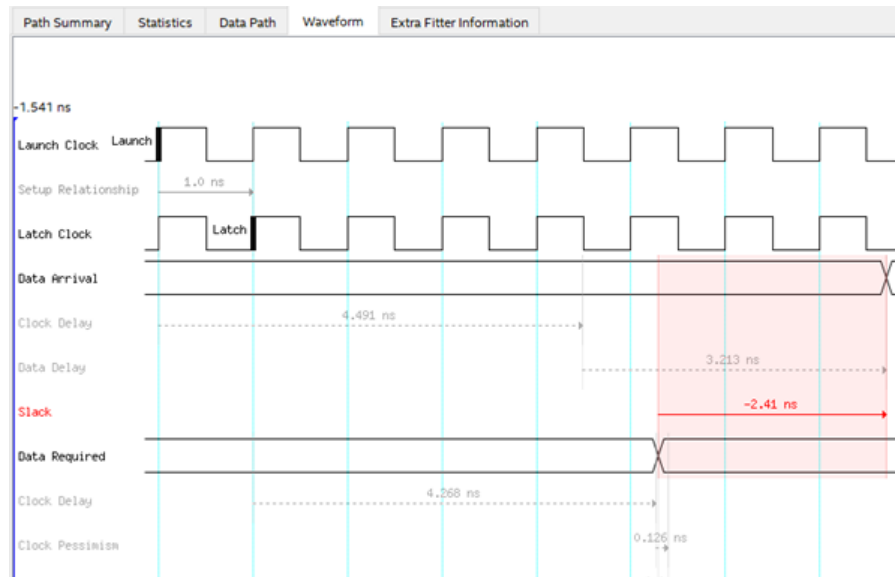


Рисунок 4.39 – Часові діаграми для шифратора b3\_enc\_case при використанні профіля оптимізації Area

Performance є найкращим режимом для шифратора b3\_enc\_case. Часові діаграми підтверджують, що цей режим дозволяє значно скоротити затримки у критичних шляхах і забезпечити високу продуктивність.

## ВИСНОВКИ

У процесі виконання роботи було розроблено та досліджено моделі шифраторів на основі мови Verilog та вплив різних режимів оптимізації при синтезі на їх характеристики. Основними аспектами аналізу стали часова затримка, використання логічних елементів і максимальна частота роботи схем. Було розглянуто три типи шифраторів: b1\_enc\_assign, b2\_enc\_if, b3\_enc\_case. Особливу увагу приділено аналізу часових характеристик, шляхів проходження сигналів у Chip Planner та часових діаграм.

Шифратор b1\_enc\_assign продемонстрував найменші затримки серед усіх варіантів. У режимі Performance (High) часова затримка Data Delay становила 1.334 нс, Total Delay CELL зменшилася до 0.986 нс, а Total Delay IC трохи зросла до 0.348 нс. Це забезпечило стабільну роботу на високій частоті з мінімальними затримками критичного шляху, підтвердженими аналізом у Chip Planner. У режимах Power (High) і Power (Aggressive) затримки Data Delay трохи зросли (до 1.351 нс і 1.341 нс відповідно), що підходить для задач із суворими вимогами до енергоспоживання. Balanced (Normal flow) забезпечив оптимальний баланс із Data Delay на рівні 1.334 нс і Total Delay CELL у 1.010 нс. Аналіз часових діаграм підтвердив, що цей шифратор може працювати стабільно навіть при значному навантаженні, а шляхи у Chip Planner є компактними та ефективними.

Шифратор b2\_enc\_if продемонстрував більші затримки через складність його структури. У режимі Performance (High) Data Delay становила 3.690 нс, тоді як Total Delay CELL досягла 2.322 нс. Аналіз у Chip Planner виявив декілька критичних шляхів, що обмежували продуктивність. Режим Power (High) покращив показники, зменшивши Data Delay до 3.476 нс і Total Delay CELL до 2.198 нс, що робить його придатним для енергоефективних систем. Balanced (Normal flow) із Data Delay у 3.682 нс і

Total Delay CELL у 1.877 нс забезпечив прийнятний компроміс між продуктивністю та ресурсами. Часові діаграми продемонстрували, що основні затримки виникають через довгі послідовності умов у коді, які могли б бути оптимізовані іншими методами.

Шифратор `b3_enc_case` характеризувався ще більшими затримками критичних шляхів. У режимі Performance (High) Data Delay становила 3.311 нс, Total Delay CELL — 2.163 нс, а Total Delay IC — 1.148 нс. Аналіз у Chip Planner показав, що критичні шляхи залишалися довгими, ніж у попередніх шифраторах. У режимах Power (High) і Power (Aggressive) затримки Data Delay зменшилися (до 3.235 нс і 3.150 нс відповідно), але Total Delay CELL зростав, досягаючи 2.282 нс у Power (Aggressive). Balanced (Normal flow) із Data Delay у 3.373 нс і Total Delay CELL у 2.006 нс забезпечив помірну продуктивність. Часові діаграми вказали на значне зменшення запасу часу, що обмежує частоту роботи цього шифратора, а аналіз у Chip Planner підтвердив, що структура має потенціал для оптимізації шляхом перерозподілу ресурсів.

Загалом, режим Performance (High) продемонстрував найкращі результати для шифратора `b1_enc_assign`, забезпечуючи найменші затримки критичного шляху і високу частоту роботи. Balanced став універсальним варіантом для задач із середніми вимогами до продуктивності, тоді як Power був доцільним для енергоефективних застосувань. Проведений аналіз підтвердив ефективність використання інструментів TimeQuest і Chip Planner для оптимізації цифрових пристроїв, що дозволяє адаптувати їх до різних сценаріїв застосування. Використання цих інструментів дозволило не лише ідентифікувати критичні шляхи, але й оцінити вплив кожного режиму оптимізації на продуктивність, енергоспоживання та використання ресурсів. Отримані результати забезпечують основу для подальшого вдосконалення цифрових систем.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Шкіль О. С., Рахліс Д. Ю., Філіпенко І. В., Корнієнко В. Р., Рожнова Т. Г. Автоматизоване проєктування вбудованих систем цифрового оброблення сигналів на платформі SoC. Сучасний стан наукових досліджень та технологій в промисловості. 2024. № 1 (27). С. 72–82.
2. Olesia Barkovska, Inna Filippenko, Ivan Semenenko, Valentyn Korniienko, Peter Sedlaček, Adaptation of FPGA architecture for accelerated image preprocessing, *Radioelectronic and Computer Systems*, 2 (106) 2023, 94-106.
3. Проєктування цифрових схем з використанням мови VHDL: Навчальний посібник/ В.В. Семенець, І.В. Хаханова, В.І. Хаханов. – ХНУРЭ, 2003. 492 с.
4. Шкіль О. С., Рахліс Д. Ю., Філіпенко І. В., Корнієнко В. Р. Проєктування та самодіагностика кіберфізичних пристроїв керування на платформі SoC. Сучасний стан наукових досліджень та технологій в промисловості. 2023. № 4 (26). С. 122–134.
5. Cooley, James W., Tukey, John W. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics Of Computation*. 1965, Vol. 19, p. 297-301.
6. Changela A., Zaveri M., Verma D. FPGA implementation of high-performance, resource-efficient Radix-16 CORDIC rotator based FFT algorithm. *Integration*. 2020, Vol. 73, p. 89-100.
7. Oppenheim, A. V., Schaffer, R. W., Buck, J. R. *Discrete-Time Signal Processing*. Second Edition. Upper Saddle River, NJ: Prentice Hall, Inc., 1999.