

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Моделювання цифрових пристроїв з використанням
мови VHDL

(тема)

Виконав:

здобувач 4 року навчання,

групи КІУКІ-21-1

Данило ШАПОВАЛОВ

(власне ім'я, прізвище)

Спеціальність

123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник: ст.викл. Владислав ДЯЧЕНКО

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

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

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Шаповалову Данилові Андрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Моделювання цифрових пристроїв з використанням мови VHDL _____

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424 Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 17 червня 2025 р.

3. Вхідні дані до роботи _____

VHDL

частотний синтезатор

генератор

АЛП

моделювання

4. Перелік питань, що потрібно опрацювати у роботі _____

Аналіз предметної області

Аналіз та вибір цифрових пристроїв для моделювання

Реалізація та моделювання

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 13 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

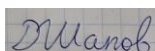
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Отримання завдання та аналіз літератури	26.05.2025–30.05.2025	
2	Огляд існуючих середовищ моделювання	31.05.2025–03.06.2025	
3	Вибір засобів для реалізації	04.06.2025–06.06.2025	
4	Програмна реалізація моделі	07.06.2025–08.06.2025	
5	Моделювання часових діаграм	09.06.2025–11.06.2025	
6	Аналіз отриманих результатів	12.06.2025–13.06.2025	
7	Оформлення записки	14.06.2025–16.06.2025	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач


(підпис)

Керівник роботи


(підпис)

ст. викл. Владислав ДЯЧЕНКО

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 57 с., 10 рис., 2 дод., 10 джерел.

VHDL, ЦИФРОВИЙ СИНТЕЗ ЧАСТОТИ, DDS, АРИФМЕТИКО-ЛОГІЧНИЙ ПРИСТРІЙ, ALU, DDR SDRAM КОНТРОЛЕР, ЦИФРОВІ МОДЕЛІ, СТРУКТУРНИЙ ОПИС, ПОВЕДІНКОВЕ МОДЕЛЮВАННЯ, СИМУЛЯЦІЯ, FPGA, КОНВЕЄРНА ОБРОБКА, АРХІТЕКТУРА ПРИСТРОЮ, ІНТЕРФЕЙС СИГНАЛІВ, АВТОМАТ КЕРУВАННЯ.

Метою кваліфікаційної роботи є комплексний аналіз методології моделювання складних цифрових пристроїв із використанням мови опису апаратури VHDL, а також практична реалізація та симуляція вибраних цифрових схем.

У ході виконання кваліфікаційної роботи розроблено моделі трьох цифрових пристроїв: цифрового частотного синтезатора, конвеєрного арифметико-логічного пристрою та багатоканального контролера DDR SDRAM. Для кожного з них побудовано повноцінну структурну архітектуру, визначено сигнальні інтерфейси, внутрішню логіку, реалізовано автомат керування та забезпечено повну підтримку симуляції. Застосування мови VHDL дозволило забезпечити як поведінковий, так і структурний опис моделей, що сприяло високій точності відображення функціонування пристроїв та можливості подальшого синтезу для FPGA.

ABSTRACT

Bachelor's thesis: 57 pages, 10 figures, 2 appendices, 10 sources.

VHDL, DIGITAL FREQUENCY SYNTHESIS, DDS, ARITHMETIC LOGIC UNIT, ALU, DDR SDRAM CONTROLLER, DIGITAL MODELS, STRUCTURAL DESCRIPTION, BEHAVIORAL MODELING, SIMULATION, FPGA, PIPELINED PROCESSING, DEVICE ARCHITECTURE, SIGNAL INTERFACE, CONTROL UNIT.

The major goal of this thesis is to provide a comprehensive analysis of the methodology for modeling complex digital devices using the VHDL hardware description language, as well as the practical implementation and simulation of selected digital circuits.

In order to the models of three digital devices were developed: a digital frequency synthesizer, a pipelined arithmetic logic unit, and a multi-channel DDR SDRAM controller. For each of these, a complete structural architecture was designed, signal interfaces and internal logic were defined, a control state machine was implemented, and full simulation support was ensured. The use of VHDL enabled both behavioral and structural descriptions of the models, which ensured high accuracy in representing device functionality and the potential for further synthesis onto FPGA platforms.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	7
ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	10
1.1 Основні поняття моделювання цифрових пристроїв	10
1.2 Огляд мов опису апаратних засобів	11
1.3 Порівняльний аналіз VHDL з іншими мовами	13
1.4 Середовища розробки та симуляції цифрових пристроїв на VHDL.....	16
2 АНАЛІЗ ТА ВИБІР ЦИФРОВИХ ПРИСТРОЇВ ДЛЯ МОДЕЛЮВАННЯ	20
2.1 Критерії вибору цифрових пристроїв	20
2.2 Аналіз сучасних цифрових пристроїв складної структури	22
2.3 Обґрунтування вибору цифрових пристроїв для моделювання.....	24
2.4 Структурна характеристика обраних цифрових пристроїв	26
3 РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ.....	29
3.1 Розробка та моделювання цифрового частотного синтезатора	29
3.2 Реалізація генератора синусоїдального сигналу.....	33
3.3 Розробка та моделювання конвеєрного арифметико-логічного пристрою	35
3.4 Розробка та моделювання багатоканального контролера пам'яті DDR SDRAM	37
ВИСНОВКИ.....	41
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	43
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	44
ДОДАТОК Б Програмний код	52

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ALU – арифметико-логічний пристрій
CLK – тактовий сигнал
CPU – центральний процесор
DDR – двократна швидкість передавання даних
DDS – цифровий частотний синтезатор
DQS – строб даних DDR SDRAM
FSM – автомат скінченних станів
FPGA – програмована користувачем логічна інтегральна схема
HDL – мова опису апаратури
LUT – таблиця пошуку значень
RAM – пам'ять із довільним доступом
RAS – строб рядкової адреси
ROM – постійна пам'ять
SDRAM – синхронна динамічна пам'ять з довільним доступом
VHDL – мова опису апаратури VHSIC
WE – сигнал дозволу запису
WR_BUF – буфер запису
XNOR – логічна операція «виключне НІ»

ВСТУП

У сучасних умовах розвитку цифрових технологій особливого значення набуває створення високопродуктивних, масштабованих та ефективних цифрових пристроїв, здатних обробляти великі обсяги даних у режимі реального часу. Перехід від традиційного аналогового проєктування до цифрового став не лише еволюційним етапом у технічному розвитку, а й ключовим чинником забезпечення функціональної гнучкості, точності та надійності електронних систем. В умовах швидкого оновлення технічних стандартів та зростання складності цифрових систем зростає потреба у відповідних інструментах, які дозволяють ефективно моделювати, перевіряти та реалізовувати складні логічні структури на етапі проєктування.

Мова опису апаратури VHDL, як одна з провідних мов у галузі проєктування цифрових систем, надає потужні засоби для формального опису структури та поведінки апаратних засобів на різних рівнях абстракції. Використання VHDL дозволяє створювати параметризовані та модульні цифрові компоненти, автоматизувати процес верифікації, а також зменшити витрати часу на розробку і тестування. Завдяки цим властивостям VHDL широко застосовується як у промисловості, так і в академічних дослідженнях для реалізації складних цифрових систем, що включають контролери пам'яті, цифрові сигнальні процесори, арифметичні пристрої та елементи комунікаційної логіки.

Актуальність роботи зумовлена необхідністю розробки цифрових пристроїв нового покоління, здатних задовольняти високі вимоги до продуктивності, компактності та енергетичної ефективності. Особливо важливою є можливість їхньої програмної реконфігурації, яку забезпечує мова VHDL у поєднанні з FPGA-платформами. З огляду на зростаючий попит на спеціалізовані цифрові модулі в галузях телекомунікацій, обробки сигналів, автоматизації виробництва та кіберфізичних систем, проблема

моделювання складних цифрових пристроїв набуває особливого значення.

Метою даної роботи є дослідження теоретичних і практичних аспектів моделювання складних цифрових пристроїв із використанням мови опису апаратури VHDL, а також розробка, симуляція та аналіз ефективності кількох типових цифрових модулів, що відображають актуальні інженерні задачі. Особливу увагу приділено реалізації пристроїв із високим рівнем логічної складності, які ілюструють можливості VHDL як інструменту для побудови гнучких і масштабованих систем.

У межах роботи розглянуто повний цикл проектування – від аналізу вимог до цифрових пристроїв і побудови їхньої архітектури до реалізації коду VHDL, моделювання в спеціалізованому середовищі та оцінювання ефективності розроблених схем. Робота охоплює як структурний, так і поведінковий рівні моделювання, з урахуванням оптимізації ресурсів і потенційної апаратної реалізації на ПЛІС. Результати дослідження можуть бути застосовані для створення нових програмованих цифрових пристроїв або адаптації існуючих моделей до змінюваних технічних умов

Мета кваліфікаційної роботи полягає в комплексному аналізі методології моделювання складних цифрових пристроїв із використанням мови опису апаратури VHDL, а також у практичній реалізації та симуляції вибраних цифрових схем.

Завдання:

- розглянути можливості VHDL у контексті проектування апаратних систем і провести аналіз її переваг порівняно з альтернативними мовами опису;
- обґрунтувати вибор складних цифрових пристроїв, які відповідають сучасним вимогам до обчислювальної продуктивності, багатofункціональності та архітектурної гнучкості;
- розробити відповідні архітектури цифрових модулів, описати їх на мові VHDL та перевірити працездатність за допомогою моделювання.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Основні поняття моделювання цифрових пристроїв

Моделювання цифрових пристроїв є фундаментальною складовою процесу розробки апаратного забезпечення, що дозволяє здійснювати формалізований опис, верифікацію та аналіз логічної структури майбутньої цифрової системи ще до її фізичної реалізації. Такий підхід дає змогу виявити помилки на ранніх етапах розробки, оцінити функціональні характеристики пристрою, оптимізувати використання апаратних ресурсів та адаптувати архітектуру до конкретних обмежень середовища, у якому пристрій буде функціонувати.

У контексті цифрової електроніки під моделюванням розуміють процес створення віртуального образу пристрою, який описує його поведінку на логічному або структурному рівні з використанням спеціалізованих мов опису апаратури. Таке моделювання може охоплювати як ізольовані компоненти, наприклад суматори, регістри чи мультиплексори, так і комплексні системи, включно з процесорними ядрами, контролерами пам'яті або цифровими сигнальними процесорами.

Центральне місце в моделюванні цифрових систем займає поняття абстракції, що дає змогу розглядати пристрій на різних рівнях деталізації. Поведінковий рівень дає змогу описати функціональність пристрою незалежно від його внутрішньої структури, тоді як структурний рівень фокусується на зв'язках між окремими компонентами системи. Архітектурний рівень дозволяє досліджувати логіку взаємодії модулів з урахуванням часових і ресурсних обмежень. Ці підходи забезпечують гнучкість у процесі проєктування і дозволяють поетапно ускладнювати модель відповідно до цілей розробки.

Мовами опису апаратури, такими як VHDL, забезпечується можливість

точного і формалізованого представлення логіки цифрового пристрою, включаючи паралелізм обчислень, часові затримки та синхронізацію сигналів. У межах одного проєкту може бути реалізовано кілька варіантів моделі з різною глибиною деталізації, що дозволяє проводити імітацію поведінки окремих компонентів або повної системи в цілому.

Важливим аспектом моделювання є перевірка відповідності моделі заданим специфікаціям за допомогою симуляції. Симуляція дозволяє наочно відтворити динаміку функціонування пристрою, проаналізувати реакцію на різні вхідні сигнали, перевірити коректність логічних переходів і забезпечити контроль за синхронізацією процесів. У зв'язку з цим моделювання виступає не лише як допоміжний етап, а як необхідна частина повного циклу цифрового проєктування.

Таким чином, моделювання цифрових пристроїв є багаторівневим процесом, що об'єднує концептуальний опис архітектури, формальне програмне представлення логіки та імітаційне тестування функціональної поведінки системи. Застосування цього підходу значно підвищує ефективність розробки, сприяє скороченню часу виходу продукту на ринок та забезпечує більшу надійність цифрових пристроїв у реальних умовах експлуатації.

1.2 Огляд мов опису апаратних засобів

Процес проєктування сучасних цифрових систем неможливо уявити без використання мов опису апаратури, які забезпечують засоби для формалізованого представлення логіки, структури та поведінки апаратних компонентів. Такі мови, відомі під загальною назвою Hardware Description Languages (HDL), слугують основою для створення моделей цифрових пристроїв, їх симуляції, тестування, верифікації та подальшого синтезу в апаратну реалізацію.

Серед найпоширеніших мов опису апаратури провідні позиції

займають VHDL (VHSIC Hardware Description Language), Verilog HDL та SystemVerilog. Кожна з цих мов має свою історію розвитку, область застосування та специфіку синтаксису, проте всі вони створені для вирішення однакових завдань: опису цифрових систем у вигляді коду, що може бути прочитаний як людиною, так і спеціалізованими автоматизованими засобами синтезу.

VHDL був розроблений у рамках програми Міністерства оборони США для проектування систем надвисокого ступеня інтеграції. Його основною перевагою є сувора типізація, багаторівнева абстракція, можливість опису як поведінкових, так і структурних моделей, а також широкі можливості повторного використання коду через створення параметризованих компонентів. Завдяки високому ступеню формальності та підтримці складних конструкцій, VHDL набув широкого поширення в академічному середовищі та в індустріальних проектах, що вимагають точності та надійності.

Verilog HDL, на відміну від VHDL, має синтаксис, що нагадує мову програмування C, і орієнтований на опис логіки нижчого рівня з акцентом на зручність моделювання комбінаційної та послідовної логіки. Він широко використовується в проектуванні мікросхем і FPGA-компонентів, особливо у США та Азії. Через свою простоту Verilog часто обирають у задачах швидкої розробки та тестування.

SystemVerilog розширює можливості класичного Verilog, додаючи об'єктно-орієнтовані структури, покращені засоби верифікації та підтримку інтерфейсів. Він орієнтований переважно на верифікацію складних систем на кристалі (SoC) та забезпечує повний цикл розробки в межах однієї мови, поєднуючи опис апаратури та засоби перевірки.

Окрім традиційних HDL, також існують сучасні альтернативи, що інтегрують мовні конструкції в середовища загального призначення, зокрема MyHDL, Chisel та Bluespec, які дозволяють описувати апаратні компоненти мовами Python або Scala з подальшою трансляцією у VHDL чи Verilog. Такі

підходи стають дедалі популярнішими у сфері наукових досліджень та стартап-розробок завдяки високому рівню абстракції та інтеграції з іншими інструментами розробки.

Важливо відзначити, що вибір мови опису апаратури залежить від вимог конкретного проєкту, досвіду розробника, доступного інструментарію та цільової платформи реалізації. Незважаючи на наявність різноманітних мов HDL, VHDL залишається одним із найпоширеніших стандартів завдяки своїй універсальності, строгій структурі, сумісності з більшістю засобів автоматизованого проєктування та підтримці синтезу на провідних FPGA-платформах. Його застосування дозволяє забезпечити точний, масштабований і відтворюваний процес розробки цифрових систем різної складності.

1.3 Порівняльний аналіз VHDL з іншими мовами

Вибір мови опису апаратного забезпечення відіграє ключову роль у процесі розробки цифрових систем, оскільки безпосередньо впливає на ефективність моделювання, зручність налагодження, можливість повторного використання коду та якість кінцевого результату. Серед мов, що традиційно використовуються в цифровому проєктуванні, найбільш поширеними є VHDL, Verilog та SystemC, кожна з яких має свої переваги, обмеження та область застосування. Їх порівняння дає змогу обґрунтувати вибір інструментарію для конкретного інженерного завдання, враховуючи вимоги до точності моделювання, масштабованості системи та особливостей цільової платформи.

VHDL вирізняється формалізованим підходом до опису цифрової логіки, що виявляється в суворій типізації, розгалуженій структурі коду, чітких механізмах обробки сигналів і процесів. Ця мова орієнтована на створення складних архітектур із глибоким рівнем ієрархії, підтримує опис як синхронної, так і асинхронної логіки, дозволяє вести проєктування з

урахуванням часових затримок та архітектурних особливостей. Її застосування особливо доцільне у великих проєктах, де критичними є стабільність, модульність та можливість детальної перевірки кожного компоненту системи. З огляду на ці переваги, VHDL є основним стандартом для проєктування у європейському оборонному та аерокосмічному секторі, а також широко використовується в академічному середовищі.

Verilog, на відміну від VHDL, має менш формалізовану структуру та синтаксис, близький до мов програмування загального призначення, зокрема C. Його основна перевага полягає в компактності коду та простоті моделювання низькорівневої логіки, що забезпечує швидке створення базових цифрових модулів. Verilog широко використовується в промисловості для розробки цифрових інтегральних схем, зокрема ASIC, і активно підтримується великими виробниками мікросхем. Водночас менш сувора типізація та спрощена модель обробки сигналів можуть спричинити труднощі в діагностиці помилок на складних рівнях абстракції.

SystemC представляє собою вищий рівень абстракції, що поєднує в собі властивості мов програмування з можливостями опису апаратури. Основана на мові C++, вона дозволяє здійснювати системно-рівневе моделювання та використовувати об'єктно-орієнтовані підходи до побудови складних цифрових систем, зокрема систем на кристалі (SoC). SystemC орієнтована на функціональне моделювання та апаратно-програмну співрозробку, забезпечуючи засоби для ранньої верифікації архітектурних рішень ще до синтезу на фізичному рівні. Водночас її використання вимагає вищого рівня технічної підготовки, розуміння концепцій моделювання часу, транзакційного рівня та паралелізму на рівні систем.

Проведений порівняльний аналіз (рисунок 1.1) демонструє, що кожна з розглянутих мов опису апаратури має свою специфічну область застосування. VHDL зберігає позиції універсального стандарту завдяки строгій типізації та формальності, що забезпечує надійність великих проєктів. Verilog домінує в промисловій розробці через простоту та

ефективність.

Критерій оцінки	VHDL	Verilog	SystemVerilog	Альтернативи
Складність освоєння	Висока	Помірна	Дуже висока	Варіюється
Рівень типізації	Строга	Слабка	Покращена	Залежить від базової мови
Продуктивність розробки	Помірна	Висока	Висока для складних систем	Потенційно висока
Можливості верифікації	Базові	Базові	Розширені	Обмежені
Промислове поширення	Широке	Домінуюче	Зростаюче	Нішеве
Підтримка інструментів	Універсальна	Найширша	Розвивається	Обмежена
Ресурсні вимоги	Помірні	Низькі	Високі	Варіюються

Рисунок 1.1 – Аналіз основних мов опису апаратури

SystemVerilog стає стандартом для верифікації складних систем, тоді як альтернативні високорівневі підходи знаходять застосування у дослідницьких проєктах та швидкому прототипуванні.

Вибір конкретної мови має базуватися на комплексному аналізі вимог проєкту, досвіду команди розробників, доступного інструментарію та довгострокових цілей розвитку продукту. Майбутній розвиток галузі вказує на подальшу диверсифікацію підходів до опису апаратури з акцентом на підвищення рівня абстракції та інтеграцію з сучасними методологіями розробки програмного забезпечення.

Таким чином, VHDL надає найбільш детальний і формалізований інструментарій для опису апаратної логіки, що є ключовим при реалізації масштабованих і критично важливих цифрових пристроїв. Verilog забезпечує швидке й ефективно моделювання на рівні елементів з меншою кількістю коду, однак вимагає ретельнішої верифікації. SystemC відкриває перспективи на рівні високорівневого моделювання систем і спрощує інтеграцію апаратури з програмними компонентами. Вибір між цими мовами визначається специфікою завдання, рівнем деталізації, вимогами до

симуляції та досвідом проєктувальника. У межах цієї роботи перевага надана VHDL як мові, що найкраще поєднує точність опису, структурованість та сумісність із сучасними платформами цифрового синтезу.

1.4 Середовища розробки та симуляції цифрових пристроїв на VHDL

Ефективне застосування мови опису апаратури VHDL у процесі проєктування цифрових пристроїв передбачає використання спеціалізованих середовищ розробки, які забезпечують повний цикл – від написання коду до симуляції, верифікації та синтезу моделі у вигляді апаратної реалізації. Такі інструменти інтегрують синтаксичний аналізатор, графічні й текстові редактори, візуалізатори сигналів, інтерфейси для створення тестових середовищ та підтримку FPGA-платформ.

Серед найпоширеніших засобів симуляції та розробки цифрових схем на VHDL значне місце займає ModelSim. Це середовище підтримує мультимовні проєкти, дозволяє здійснювати детальне моделювання часової поведінки, надає зручні засоби перегляду сигналів у формі вейвформ, а також підтримує ієрархічну структуру моделі. ModelSim дозволяє точно моделювати цифрові схеми з урахуванням затримок, логічних рівнів та асинхронних подій, що робить його придатним для верифікації складних цифрових систем.

Важливим інтегрованим середовищем, яке поєднує симуляцію, синтез та реалізацію на FPGA, є Vivado Design Suite від компанії Xilinx. Цей інструмент орієнтований на створення цифрових систем, які реалізуються на платформах типу Artix, Kintex та Zynq. Vivado підтримує як графічне, так і текстове проєктування, має механізми інтеграції IP-ядер, засоби автоматичного маршрутування, аналізу статичного таймінгу, а також симуляції поведінки проєкту. Завдяки підтримці VHDL та Verilog, а також великому набору бібліотек, Vivado дозволяє реалізовувати масштабовані та оптимізовані рішення.

Критерій	Active-HDL	ModelSim	Vivado	Quartus Prime	GHDL
Симуляція					
Швидкість симуляції	Висока	Дуже висока	Висока	Висока	Помірна
Мультимовність	VHDL/Verilog	VHDL/Verilog/SV	VHDL/Verilog	VHDL/Verilog	Тільки VHDL
Візуалізація	Відмінна	Відмінна	Добра	Добра	Відсутня
Верифікація					
Налагодження	Розширені засоби	Професійні	Інтегровані	Інтегровані	Базові
Тестбенчі	Автогенерація	Ручне створення	Змішаний	Змішаний	Ручне
Покриття коду	Підтримується	Підтримується	Обмежена	Обмежена	Відсутня
Синтез і реалізація					
Синтез	Відсутній	Відсутній	Інтегрований	Інтегрований	Відсутній
FPGA підтримка	Універсальна	Через треті сторони	Xilinx	Intel/Altera	Відсутня
IP інтеграція	Обмежена	Обмежена	Розширена	Розширена	Відсутня
Використання					
Складність освоєння	Помірна	Помірна	Висока	Висока	Низька
Вартість	Комерційна	Комерційна	Комерційна	Комерційна	Безкоштовна
Цільова аудиторія	Академія/Промисловість	Промисловість	FPGA розробка	FPGA розробка	Дослідження

Рисунок 1.2 – Порівняльний аналіз функціональних можливостей

Іншим провідним середовищем є Quartus Prime, розроблене компанією Intel (Altera), яке використовується для розробки цифрових пристроїв, що реалізуються на ПЛІС серій Cyclone, Stratix або MAX. Це середовище забезпечує підтримку мови VHDL, має широкі можливості аналізу ресурсів, підтримує автоматичну генерацію блоків пам'яті, арифметичних модулів та стандартних цифрових компонентів, що значно пришвидшує процес створення складної логіки.

Особливої уваги заслуговує Active-HDL, розроблене компанією Aldec, яке орієнтоване на професійне моделювання, налагодження і верифікацію цифрових схем, описаних мовами VHDL і Verilog. Active-HDL вирізняється інтуїтивно зрозумілим інтерфейсом, інтеграцією з мовою Tcl для автоматизації процесів, а також високою швидкістю симуляції. Однією з його

переваг є можливість створення змішаних проєктів із різними мовами, зручне формування тестбенчів, автоматичне створення вейвформ і контроль значень сигналів. Крім того, середовище підтримує моделі поведінки з різним рівнем абстракції, включаючи змішаний структурно-поведінковий опис, що є корисним у багаторівневому проєктуванні. Active-HDL широко застосовується в академічному середовищі для навчальних і дослідницьких цілей, а також у промисловості, де потрібна прозора верифікація цифрових рішень.

У дослідницьких задачах часто застосовується GHDL, відкритий симулятор VHDL, що дозволяє запускати моделі в командному рядку, підтримує різні режими симуляції (аналіз, елаборація, виконання) та може бути інтегрований у скриптові середовища з підтримкою автоматичного тестування. Його використання доцільне для задач, де необхідна автоматизована перевірка великих обсягів HDL-коду або інтеграція з системами керування версіями.

Аналіз провідних інструментальних середовищ (рисунок 1.2) демонструє різноманітність підходів до проєктування цифрових систем на VHDL. Active-HDL виділяється як універсальне рішення для навчання та професійної верифікації завдяки збалансованому поєднанню функціональності та зручності використання. ModelSim залишається стандартом для високопродуктивної промислової симуляції. Vivado та Quartus Prime забезпечують повний цикл розробки для відповідних FPGA-платформ. GHDL знаходить застосування в дослідницьких та автоматизованих середовищах.

Вибір конкретного інструментарію має базуватися на специфіці проєкту, вимогах до продуктивності, цільової платформи реалізації та доступного бюджету. Тенденція до інтеграції різних етапів проєктування в єдиних середовищах сприяє підвищенню ефективності розробки та скороченню часу виведення продукту на ринок.

Усі ці середовища забезпечують користувачеві повний спектр функцій

для перевірки працездатності моделей, аналізу результатів, виявлення помилок і оптимізації проєктних рішень. Їх використання дозволяє будувати надійні й ефективні цифрові системи з мінімізацією часу розробки, підвищенням точності реалізації та зменшенням витрат на тестування.

У межах цієї роботи передбачено використання Active-HDL як основного симулятора VHDL-коду завдяки його візуальній інформативності, швидкості обробки та підтримці гнучкого налагодження.

2 АНАЛІЗ ТА ВИБІР ЦИФРОВИХ ПРИСТРОЇВ ДЛЯ МОДЕЛЮВАННЯ

2.1 Критерії вибору цифрових пристроїв

На сучасному етапі розвитку цифрових технологій формування переліку пристроїв, які підлягають моделюванню та аналізу, повинно ґрунтуватися на низці об'єктивних критеріїв, що враховують актуальні інженерні виклики, архітектурну складність, потенціал для практичної реалізації та значущість пристрою в контексті реальних застосувань. Вибір об'єктів для дослідження не може бути довільним, оскільки ефективність реалізації моделей цифрових систем на мові VHDL значною мірою залежить від природи задач, які ставляться перед конкретними логічними структурами.

Насамперед доцільно орієнтуватися на функціональну складність пристроїв, оскільки саме вона дозволяє оцінити можливості мови опису апаратури в контексті відображення багаторівневої логіки, управління часовими залежностями, використання ресурсів та взаємодії між підсистемами. Чим складніша архітектура – тим вищі вимоги до формалізованості моделі, її ієрархії, параметризації й підтримки типових структур. Водночас надмірна складність без прив'язки до практичних потреб може призвести до втрати інженерної доцільності, тому важливо зберігати баланс між складністю та релевантністю.

Іншим важливим критерієм є інженерна затребуваність пристрою, яка визначається його роллю в сучасних цифрових системах. Найбільш актуальними залишаються ті компоненти, що входять до складу обчислювальних модулів, керувальних блоків, інтерфейсних контролерів або функціональних елементів систем обробки сигналів. Вибір на користь таких пристроїв дозволяє побудувати універсальні архітектури, придатні для широкого спектру застосувань, включаючи автоматизовані системи управління, цифрові комунікаційні платформи, пристрої вбудованої

обчислювальної техніки та інші сфери.

Не менш значущим чинником є можливість ефективної симуляції й тестування обраного пристрою у програмному середовищі. Об'єкти, що мають усталені функціональні алгоритми, типові вхідні й вихідні інтерфейси, а також добре визначену поведінку, є особливо зручними для моделювання в середовищах, орієнтованих на VHDL. Це забезпечує не лише достовірність результатів, а й зручність створення тестбенчів, що відтворюють типові сценарії роботи пристрою.

Окрему увагу слід приділяти структурній модульності й масштабованості пристрою. Моделі, які дозволяють гнучко змінювати параметри, замінювати підкомпоненти або інтегруватися в більші системи, мають перевагу у навчальному й дослідницькому контекстах. VHDL як мова опису дозволяє ефективно реалізувати такі архітектури через механізми створення компонентів, портів, пакетів і параметричних інтерфейсів, що відкриває можливості для побудови повторно використовуваних конструкцій.



Рисунок 2.1 – Критерії вибору цифрових пристроїв для моделювання на VHDL

Таким чином, при виборі цифрових пристроїв для моделювання доцільно враховувати архітектурну складність, інженерну значущість, потенціал для симуляції, модульність структури та відповідність сучасним практикам проєктування. Саме за такими критеріями (рисунки 2.1) було здійснено обґрунтований вибір пристроїв, моделювання яких дозволить повноцінно продемонструвати можливості мови VHDL у реалізації високоефективних цифрових систем.

2.2 Аналіз сучасних цифрових пристроїв складної структури

Сучасні цифрові пристрої, що застосовуються в системах обробки, зберігання й передавання даних, вирізняються високим ступенем структурної ускладненості, паралелізму виконання операцій і гнучкості у конфігуруванні. Їхня архітектура зазвичай передбачає наявність багатоетапних обчислювальних конвеєрів, складних механізмів керування, багатоканальних інтерфейсів і високошвидкісних шин, що забезпечують взаємодію з пам'яттю та периферійними модулями. У цьому контексті важливо проаналізувати окремі типи цифрових пристроїв, які є репрезентативними з точки зору логічної складності, інженерної значущості та релевантності до реальних задач.

Одним із базових класів складних пристроїв є цифрові сигнальні генератори, зокрема цифрові частотні синтезатори (DDS). Вони реалізують методику формування гармонійного сигналу заданої частоти шляхом обчислення його значень за таблицею або за допомогою алгоритмічної апроксимації. Архітектура такого пристрою включає фазовий акумулятор, таблицю значень функцій, цифро-аналоговий перетворювач або його цифровий еквівалент і блок управління. Моделювання DDS є актуальним з огляду на його застосування в телекомунікаційних системах, цифровому радіозв'язку та радіолокаційній техніці.

Інший важливий клас складних пристроїв – це арифметико-логічні

блоки (ALU) з підтримкою багаторівневого конвеєрного виконання операцій. Такі блоки є ядром обчислювальних систем, включаючи процесори, мікроконтролери та DSP-процесори. У складних ALU реалізуються операції над числами з фіксованою або плаваючою комою, побітові логічні операції, обробка прапорців стану, розгалуження на основі умов тощо. Архітектура ALU із конвеєрною структурою дозволяє досягти високої пропускної здатності завдяки одночасному виконанню кількох незалежних етапів обчислень. Моделювання такого пристрою вимагає точного опису потоків даних, контролю синхронізації та умов переходів.

Ще одним прикладом складної архітектури є контролери пам'яті, зокрема ті, що обслуговують модулі DDR SDRAM. Такі пристрої виконують формування сигналів запиту до банків пам'яті, ініціалізацію, підтримку черг транзакцій, циклічне оновлення, контроль доступу й узгодження з шиною даних. Їх структура включає автомат керування станами, декодери адрес, таймери, буфери та багатоканальні інтерфейси, що дозволяє реалізувати паралельний доступ до пам'яті з мінімізацією затримок. Актуальність цього класу пристроїв обумовлена широким застосуванням у вбудованих системах, мікропроцесорних платформах і високопродуктивних обчислювальних системах.

У загальному випадку аналіз цифрових пристроїв складної структури свідчить про те, що кожен із них об'єднує компоненти з різною функціональністю, вимагає побудови чіткої ієрархії модулів і формування багаторівневої моделі, яка враховує не лише логічну поведінку, а й часові аспекти функціонування. Моделювання таких пристроїв на VHDL відкриває можливості як для точного формалізованого опису, так і для поступової оптимізації, повторного використання блоків і реалізації адаптивних параметричних структур.

2.3 Обґрунтування вибору цифрових пристроїв для моделювання

У межах даної кваліфікаційної роботи вибір конкретних цифрових пристроїв для моделювання здійснювався з урахуванням їхньої інженерної значущості, архітектурної складності, структурної різноманітності та практичної актуальності. Особливу увагу було зосереджено на пристроях, які відображають різні класи цифрових систем – від генераторів сигналів і арифметичних блоків до контролерів пам'яті. Такий підхід забезпечує багатогранне представлення можливостей мови VHDL у формалізації поведінки цифрових пристроїв, а також дозволяє реалізувати широке коло архітектурних шаблонів, що часто зустрічаються в реальних системах.

Першим об'єктом моделювання обрано цифровий частотний синтезатор (DDS), який поєднує в собі логіку послідовного обчислення, табличного представлення даних, управління фазою та формування періодичного сигналу. DDS є яскравим прикладом пристрою, у якому реалізуються принципи цифрової генерації сигналів із заданими параметрами частоти та фази без використання аналогових компонентів. У моделюванні такого пристрою на мові VHDL особливо цінним є синтезований контроль потоку даних, керування часовими послідовностями та структурна композиція підсистем: фазового акумулятора, таблиці синусів і керуючого автомата. Вибір DDS обґрунтовується його широким використанням у радіозв'язку, телекомунікаційних системах, модуляторах і цифрових генераторах тестових сигналів.

Другим пристроєм для моделювання виступає конвеєрний арифметико-логічний пристрій (ALU), що репрезентує ядро обчислювального процесу в цифрових комп'ютерних системах. Особливість такого ALU полягає у наявності багатоступеневого конвеєра, який дозволяє виконувати логічні та арифметичні операції з високою швидкістю, розділяючи обробку на послідовність етапів. Такий підхід потребує моделювання складної логіки керування, точного узгодження сигналів синхронізації між стадіями, а також

реалізації гнучкої системи вибору операцій залежно від керуючих вхідних кодів. Обґрунтування вибору ALU полягає в універсальності його архітектури, широкому спектрі функціональності та доцільності використання в освітньому контексті як зразка обчислювального модуля.

Третім об'єктом дослідження є багатоканальний контролер пам'яті типу DDR SDRAM, який реалізує інтерфейс до зовнішньої динамічної пам'яті з високою пропускнуою здатністю. Його архітектура включає складний автомат станів, модулі чергування запитів, лічильники таймінгу, буфери даних та логіку керування сигналами оновлення, ініціалізації та відновлення. Контролер пам'яті є прикладом пристрою, що функціонує у складному часовому середовищі з численними залежностями, і тому його моделювання на VHDL вимагає детального врахування часових діаграм, точного формування шаблонів сигналів та побудови ієрархічної логіки з високим ступенем повторного використання модулів. Актуальність моделювання такого контролера обумовлена його використанням у вбудованих системах, високопродуктивних процесорах та системах реального часу.

Узагальнюючи, вибір зазначених трьох пристроїв дозволяє охопити ключові аспекти цифрового проектування – генерацію сигналів, обчислювальну логіку та обслуговування зовнішньої пам'яті. Кожен із них реалізує свій клас задач, охоплює різні архітектурні підходи та вимагає специфічних методів моделювання. Водночас усі пристрої придатні до реалізації засобами мови VHDL, що дозволяє максимально повно продемонструвати її синтаксичні, структурні та поведінкові можливості, а також верифікувати працездатність логіки за допомогою середовищ симуляції та тестування.

2.4 Структурна характеристика обраних цифрових пристроїв

Обрані для моделювання цифрові пристрої – цифровий частотний синтезатор (DDS), конвеєрний арифметико-логічний пристрій (ALU) та багатоканальний контролер пам'яті DDR SDRAM – вирізняються складною внутрішньою структурою та багаторівневою архітектурною організацією, що робить їх репрезентативними прикладами цифрових систем для реалізації на мові VHDL. Розгляд їхньої структури дозволяє виявити як спільні риси, притаманні більшості цифрових схем, так і особливості, зумовлені функціональним призначенням кожного з пристроїв.

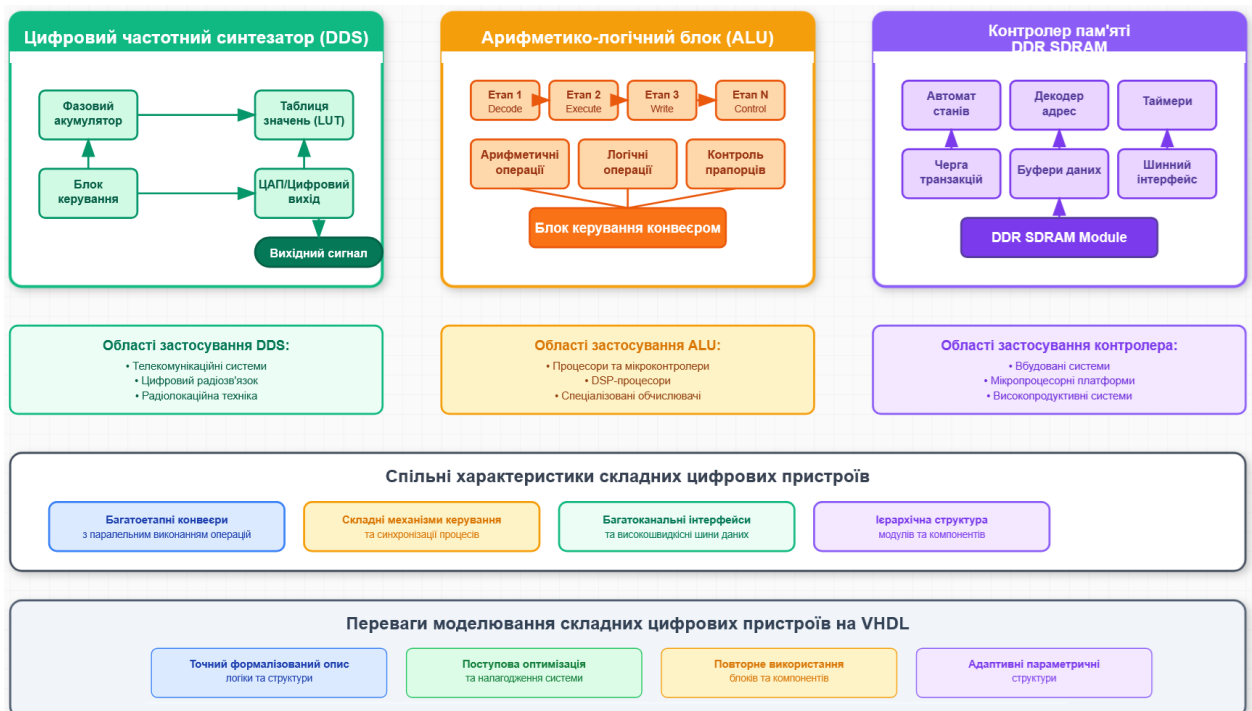


Рисунок 2.2 – Архітектура обраних пристроїв

Цифровий частотний синтезатор формується навколо центрального елемента – фазового акумулятора, який реалізує принцип приросту фази відповідно до вхідного частотного керуючого слова. Значення, накопичені в акумуляторі, подаються на блок дискретного перетворення, що зазвичай реалізується у вигляді таблиці значень синусоїдального сигналу або алгоритму апроксимації гармоніки. Вихід цього блоку подається на цифро-

аналоговий перетворювач або передається у вигляді цифрового потоку для подальшої обробки. У структурі DDS також присутній модуль керування, який формує сигнали синхронізації, ініціалізації та відновлення, що забезпечують коректне функціонування всіх складових компонентів пристрою.

Конвеєрний арифметико-логічний пристрій має модульну архітектуру, де кожна функціональна операція виконується на окремому етапі конвеєра. На першому рівні відбувається декодування операції та підготовка операндів, другий рівень реалізує власне арифметичні чи логічні перетворення, а третій відповідає за формування результату та встановлення відповідних прапорів стану. Така структура дозволяє досягти високої швидкодії за рахунок одночасного оброблення кількох інструкцій на різних стадіях. Важливою складовою ALU є блок керування, який визначає режим виконання залежно від вхідного коду операції, а також механізми синхронізації між рівнями, що забезпечують правильне переміщення даних у конвеєрі.

Контролер пам'яті DDR SDRAM має значно складнішу організацію, що обумовлена необхідністю дотримання суворих часових інтервалів і одночасної обробки численних запитів до пам'яті. Його структура включає керувальний автомат станів, що реалізує послідовності команд початкової ініціалізації, оновлення рядків пам'яті, запису та читання даних. Буферні регістри зчитування і запису забезпечують вирівнювання затримок і синхронізацію з шиною даних. У складі контролера присутні блоки таймінгу, що відстежують допустимі інтервали між послідовними командами, та модулі арбітражу, які розподіляють доступ між кількома запитами. Багатоканальна реалізація дає змогу паралельно обслуговувати кілька банків пам'яті, що значно підвищує пропускну здатність системи.

Характерною рисою всіх розглянутих пристроїв є наявність чітко окресленої модульної ієрархії, що дозволяє описати їх засобами VHDL з використанням структурного стилю кодування. Такі архітектури передбачають розділення функціональності на незалежні підблоки, які

взаємодіють через визначені інтерфейси. Це забезпечує не лише зрозумілу логіку моделі, а й дає змогу ефективно масштабувати проєкт, повторно використовувати окремі компоненти та здійснювати поетапну симуляцію або верифікацію.

3 РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ

3.1 Розробка та моделювання цифрового частотного синтезатора

Цифровий частотний синтезатор (DDS) є пристроєм, призначеним для генерації періодичних сигналів заданої частоти шляхом цифрової обробки та табличної апроксимації. Принцип дії базується на поступовому накопиченні значення фази у фазовому акумуляторі та подальшому перетворенні цієї фази у значення амплітуди за допомогою таблиці синусоїдальних значень або іншого обчислювального модуля.

Класична структура DDS включає такі ключові компоненти: блок фазового приросту, фазовий акумулятор (phase accumulator), табличний блок перетворення фази в амплітуду (lookup table – LUT) і вихідний цифровий порт. Перевагою такої архітектури є можливість точного та швидкого налаштування частоти вихідного сигналу без необхідності застосування аналогових фільтрів або генераторів. При кожному тактовому імпульсі акумулятор фази додає сталу величину – значення приросту, що визначає частоту вихідного сигналу. Отримане значення фази використовується як адреса для звернення до таблиці синусів.

Архітектура DDS реалізується на основі структурного підходу з чітким розмежуванням функціональності кожного модуля. Основними блоками є:

- фазовий акумулятор, що реалізується як регістр із додаванням вхідного приросту до накопиченого значення;
- блок LUT, що зберігає попередньо обчислені значення синусоїдального сигналу;
- модуль керування, що забезпечує синхронізацію та ініціалізацію компонентів.

З метою забезпечення точності та масштабованості в моделі використовується параметризована ширина розряду, що дозволяє варіювати

якість синусоїдального сигналу та частоту дискретизації.

Проектування цифрового частотного синтезатора з використанням мови опису апаратури VHDL передбачає створення формалізованої моделі, яка відображає структурну й поведінкову логіку усіх функціональних блоків пристрою. На етапі архітектурного планування першочерговим завданням є визначення взаємодії між компонентами, їхньої синхронізації та опису параметрів, що відповідають за налаштування частоти генерації сигналу. Мова VHDL, завдяки підтримці структурного, поведінкового та змішаного стилів моделювання, надає розробникові повний спектр засобів для реалізації гнучкої, параметризованої та перевіреної архітектури.

У межах цієї роботи архітектура DDS сформована на основі трьох основних функціональних блоків, кожен із яких реалізовано як окремий модуль: фазовий акумулятор, таблиця синусоїдальних значень (LUT) та модуль керування синхронізацією. Центральною складовою є фазовий акумулятор, що реалізується як регістр із циклічним додаванням – механізм, який відповідає за нарощування цифрової фази пропорційно до заданого частотного слова. У контексті VHDL він описується як процес, що реагує на фронт тактового сигналу й обчислює нове значення фази в кожному циклі.

Важливою частиною архітектури є інтерфейс між фазовим акумулятором і таблицею значень. Оскільки об'єм LUT зазвичай значно менший, ніж розрядність фазового регістра, для адресації таблиці використовуються лише старші біти результату. Це дозволяє зменшити обсяг пам'яті при збереженні високої частотної роздільної здатності. У VHDL це реалізується за допомогою виокремлення найстарших бітів фазового вектора та їхнього перетворення у ціле число, яке слугує адресою в масиві значень.

Таблиця синусів у цій реалізації задана як константа у вигляді масиву типу array, що дає змогу моделювати її без додаткових зовнішніх ресурсів. Значення, збережені у LUT, можуть бути сформовані зовнішнім скриптом або попередньо обчислені вручну, а в самій моделі представлені у вигляді набору восьмирозрядних слів, які імітують квантизовану синусоїду.

Модуль керування в даній архітектурі забезпечує логіку скидання, ініціалізації та налаштування частотного слова, що задається на вході. У моделі також передбачено використання вхідного сигналу `reset`, що забезпечує встановлення фазового акумулятора в нульовий стан при запуску або при ініціалізації системи. Основним сигналом синхронізації слугує `clk`, який імітує тактову частоту системи, в якій DDS використовується як підсистема генерації сигналу.

Уся архітектура реалізована в середовищі VHDL як модуль із чітко визначеними вхідними й вихідними портами, що забезпечує її інтеграцію з іншими цифровими блоками. Для забезпечення модульності та повторного використання коду реалізовано внутрішню структуру у вигляді одного процесу для фазового акумулятора та функціонального блоку LUT, що полегшує як симуляцію, так і подальший синтез у FPGA-середовищах.

Важливо відзначити, що запропонована архітектура може бути розширена додатковими функціональними можливостями, включаючи модуляцію амплітуди чи частоти, формування прямокутного або трикутного сигналу, а також включення апаратного інтерфейсу керування через SPI або UART. Усі ці розширення також можуть бути реалізовані засобами VHDL, що демонструє потенціал цієї мови як універсального інструменту моделювання складних цифрових генераторів сигналів.

У додатку Б подано приклад реалізації спрощеного DDS-модуля на VHDL. У цьому прикладі LUT реалізовано як масив із заздалегідь визначеними значеннями синуса (імітаційно), а фазовий акумулятор – як суматор з регістром.

У наведеному коді `freq_word` визначає частоту генерації. Вектор фази має ширину 32 біти, але лише 8 найстарших використовуються для адресації таблиці синусів на 256 елементів. Такий підхід дозволяє досягти високої роздільної здатності частоти за умови збереження фіксованого розміру LUT.

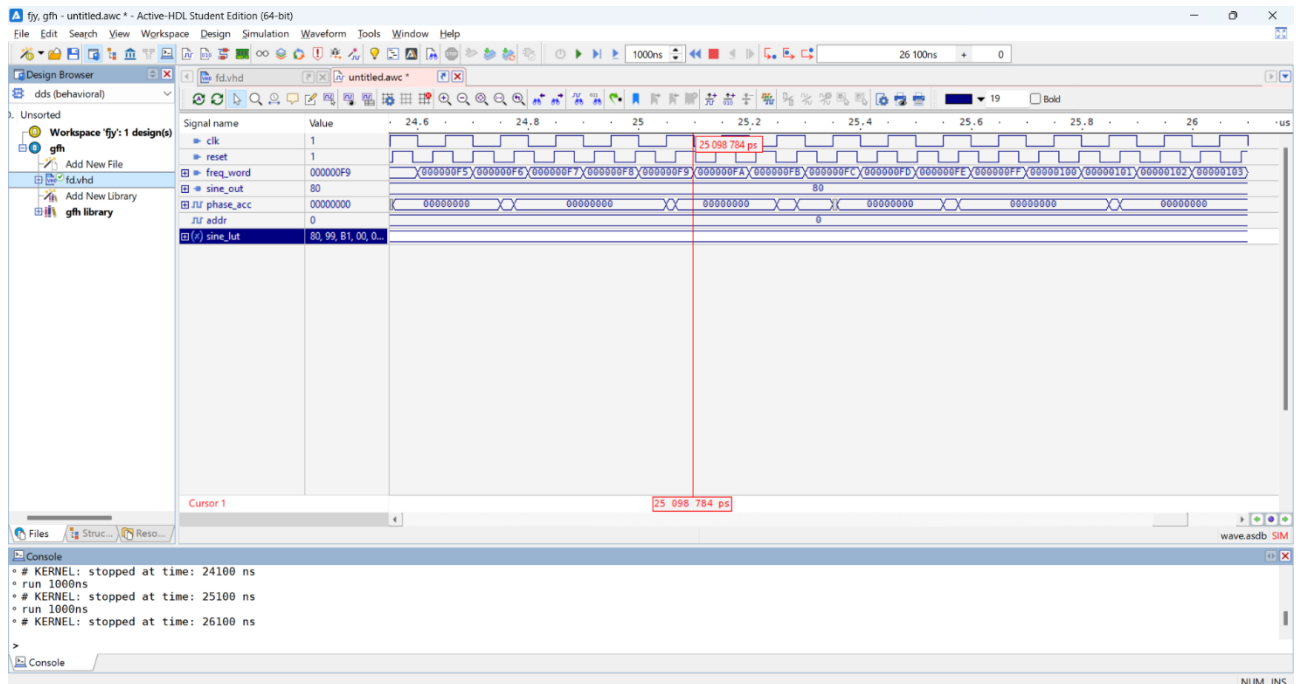


Рисунок 3.1 – Моделювання цифрового частотного синтезатора

Моделювання реалізованого DDS-модуля (рисунок 3.1) проведено в середовищі Active-HDL з використанням тестового стенду, який імітує подачу тактових імпульсів і зміни частотного слова. Результати симуляції підтверджують правильність роботи фазового акумулятора, коректну адресацію LUT і відтворення періодичного сигналу на виході.

На часових діаграмах простежується плавне наростання значення фази з періодичним циклічним обнуленням після досягнення межі розрядної сітки. Вихідні значення сигналу sine_out змінюються згідно з шаблоном синусоїди, що вказує на правильну роботу LUT та її адресування. Варіювання частотного слова у тестовому bench дозволило оцінити зміну частоти вихідного сигналу відповідно до очікуваної пропорційності.

Отримані результати свідчать про працездатність запропонованої архітектури DDS та демонструють її здатність генерувати періодичний сигнал із параметрами, визначеними цифровим способом. Це підтверджує ефективність використання мови VHDL для моделювання генераторів сигналів, а також відкриває можливості для розширення моделі, зокрема шляхом додавання модуля фазового шуму, амплітудної модуляції або

інтеграції з цифро-аналоговим перетворенням для практичного використання в апаратному середовищі.

3.2 Реалізація генератора синусоїдального сигналу

На етапі реалізації цифрового частотного синтезатора особлива увага зосереджується на створенні функціонального VHDL-опису основних логічних компонентів системи, зокрема фазового акумулятора, генератора адрес і таблиці синусоїдальних значень. Реалізація здійснюється з використанням поведінкового підходу, що дозволяє точно змоделювати динаміку функціонування пристрою та створити платформонезалежну модель, придатну для подальшого синтезу.

Ключовим компонентом DDS є фазовий акумулятор, який реалізується як 32-розрядний регістр, що при кожному фронті тактового сигналу інкрементується на фіксоване частотне слово. Це значення формується зовнішньо й може змінюватися під час виконання моделі, що дозволяє динамічно регулювати частоту вихідного сигналу. Після додавання нове значення фази використовується для генерації адреси звернення до таблиці значень синусоїди. Для цього з фазового регістра виділяються старші біти (зокрема 8 біт із 32), що дозволяє реалізувати LUT із 256 квантизованими значеннями синусоїдальної функції. Таблиця синусів описується в коді як масив констант, і в даному випадку використовується спрощений підхід із моделлю, що наближено імітує синусоїдальний сигнал.

Результатом звернення до таблиці є восьмирозрядне значення, що подається на вихідний порт і може бути зчитане або в цифровому вигляді, або передано на цифро-аналоговий перетворювач. У додатку Б наведено приклад реалізації DDS у вигляді цілісного VHDL-модуля.

Аналіз працездатності цифрового частотного синтезатора здійснювався шляхом проведення моделювання в середовищі Active-HDL. У рамках симуляції створено тестове середовище, що імітує зміну частотного слова,

надає тактові імпульси й ініціалізує сигнал скидання. Завдяки часовим діаграмам відстежується динаміка зміни значень фазового акумулятора, процес формування адреси, звернення до таблиці значень та генерація вихідного сигналу.

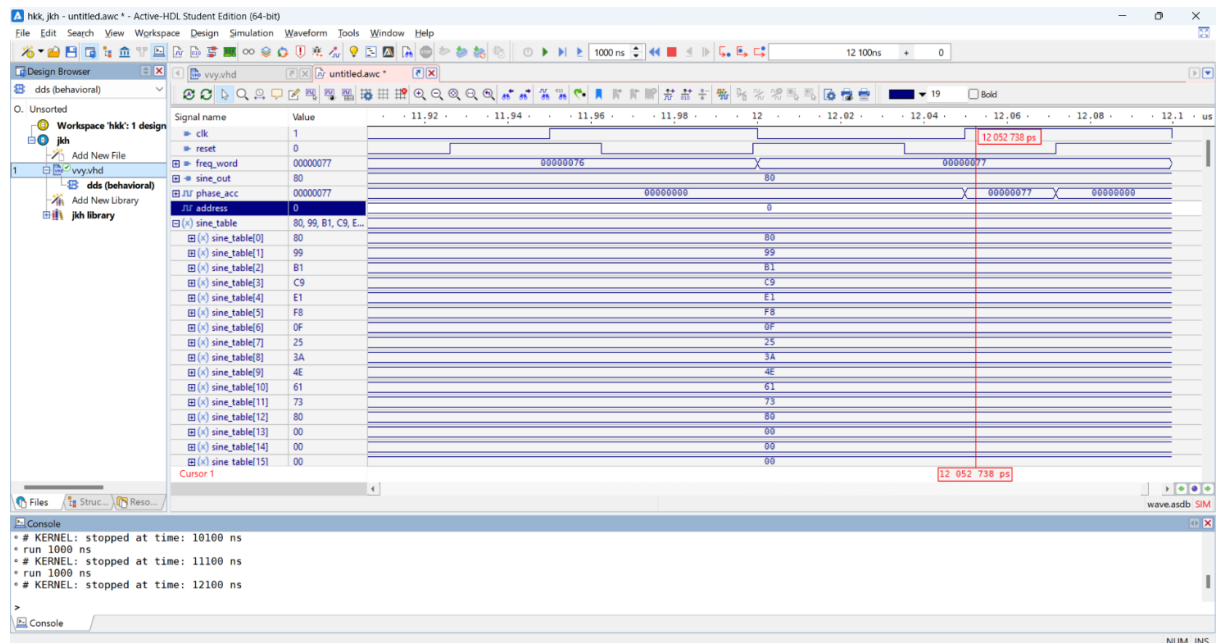


Рисунок 3.2 – Моделювання генератора синусоїдального сигналу

Під час моделювання (рисунок 3.2) простежується монотонне зростання значення фазового акумулятора із циклічним обнуленням при досягненні максимальної величини, що відповідає принципу роботи модулярного суматора. У момент кожного фронту тактового сигналу здійснюється інкремент фази, що візуально відображається на діаграмах як фіксоване зростання значення. Одночасно з цим найстарші біти фази передаються у вигляді адреси на блок LUT, який формує відповідне значення синусоїди. На часових діаграмах sine_out спостерігається формування періодичного сигналу, що наближається до синусоїдального, з періодом, обернено пропорційним до значення freq_word.

Зміна частотного слова в тестбенчі дозволила підтвердити зміну частоти вихідного сигналу без необхідності повторної компіляції або модифікації архітектури пристрою. Це вказує на динамічність та керованість

DDS у цифровому середовищі. Незважаючи на спрощену форму LUT, модель забезпечує достатню якість сигналу для загального тестування та демонстраційних цілей.

Отримані результати демонструють відповідність поведінки пристрою теоретичним очікуванням, що підтверджує правильність реалізації логіки та точність VHDL-моделі. Модель є стійкою до збоїв, повністю синхронною, підтримує динамічне керування параметрами, а також готова до подальшого синтезу в середовищі Vivado для реалізації на FPGA. Такий результат підтверджує ефективність застосування мови VHDL для моделювання складних цифрових генераторів та обґрунтовує доцільність її використання в навчальному, дослідницькому та інженерному контекстах.

3.3 Розробка та моделювання конвеєрного арифметико-логічного пристрою

У цифрових системах обробки інформації арифметико-логічний пристрій виконує фундаментальні операції, які є базою для реалізації інструкцій процесорів, мікроконтролерів і цифрових сигнальних процесорів. Концепція конвеєрної обробки дозволяє підвищити продуктивність пристрою шляхом розбиття виконання однієї операції на кілька етапів, кожен з яких може бути реалізований у вигляді незалежного апаратного блоку. Таким чином, під час виконання операції над одним набором операндів інші набори вже перебувають на наступних етапах обробки, що забезпечує істотне зменшення простоїв.

Побудова конвеєрного ALU потребує точного узгодження часових характеристик, синхронізації регістрів між стадіями та чіткого поділу обов'язків між етапами обробки. Як правило, мінімальна структура включає стадії вибору операції, декодування, виконання обчислення, та виводу результату. Така структура сприяє досягненню високої пропускної здатності навіть при обмежених апаратних ресурсах, особливо в системах, де обробка

даних є безперервною.

Конвеєрний ALU у межах даного проєкту реалізовано з трьома стадіями: перша відповідає за збереження вхідних операндів та коду операції, друга виконує обчислення згідно з вибраною функцією, третя – фіксує результат у регістрі та виводить його. Операції включають додавання, віднімання, логічне AND, логічне OR, XOR та зсуви. Усі етапи синхронізовано тактовим сигналом, і на кожному з них використано окремий регістр.

Структура ALU передбачає використання сигналу opcode (рисунок 3.3) для визначення типу обчислення, а також двох операндів a та b. Кожна операція має свій унікальний код, що декодується у другій стадії. Такий підхід дозволяє одночасно запускати операції, створюючи ефект обробки потоків даних.

В додатку Б подано приклад реалізації ALU з трьома конвеєрними стадіями: збереження вхідних даних, виконання операції та збереження результату.

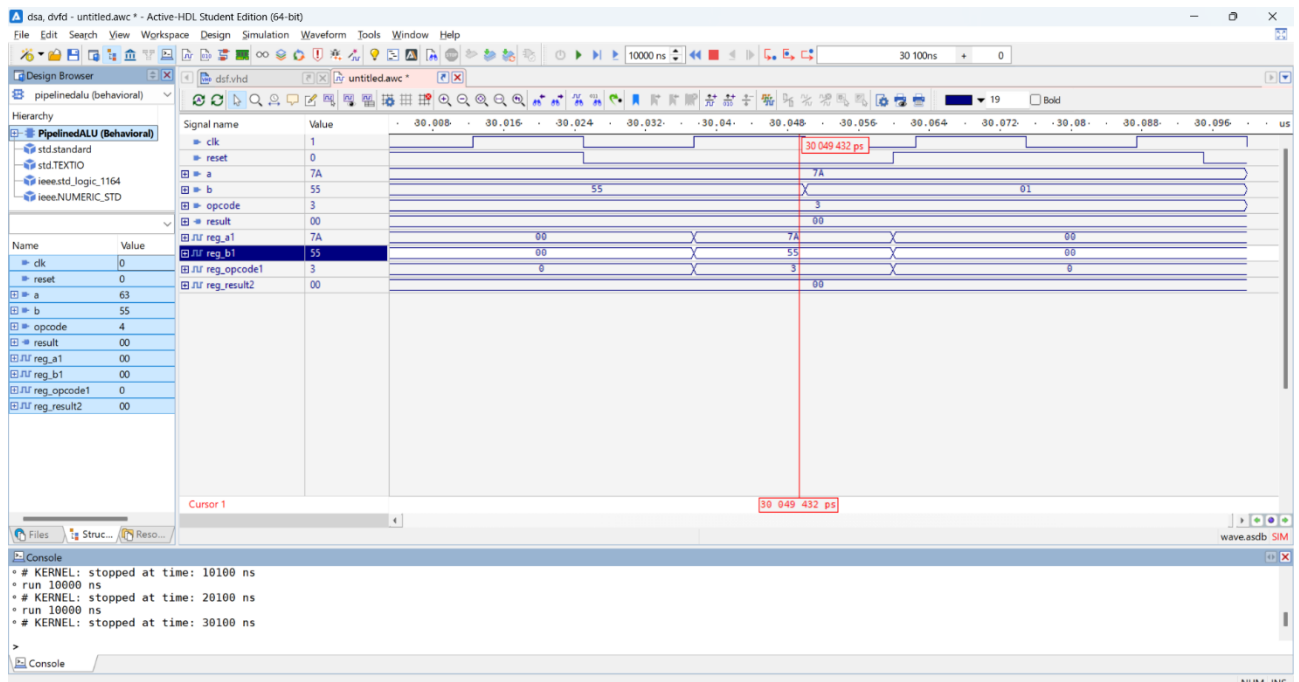


Рисунок 3.3 – Результат моделювання

Ця архітектура (рисунок 3.3) є повністю синхронною та демонструє принцип побудови обчислювального конвеєра. У кожному такті відбувається переміщення даних між регістрами, що дозволяє отримувати новий результат кожного такту після початкового запуску.

Результати моделювання (рисунок 3.3), отримані в середовищі Active-HDL, демонструють очікувану поведінку пристрою. Після затримки у два такти – що відповідає глибині конвеєра – результат з'являється на виході для кожної комбінації вхідних даних. Діаграми показують чітку послідовність: операнди потрапляють у першу стадію, передаються в другу, де обробляються, і лише після цього – у третю для виводу. Це підтверджує правильну синхронізацію регістрів та ефективне використання паралелізму.

Тестові сценарії охоплювали всі основні операції: додавання, логічні операції, зсуви, а також невалідні комбінації, які правильно оброблялися як нульовий результат. Усі випадки засвідчили коректність роботи ALU і підтвердили стійкість моделі до вхідних впливів, а також її готовність до синтезу.

Конвеєрна архітектура значно зменшує латентність при багатократній обробці потоків даних, що має особливе значення в системах із високими вимогами до пропускну здатності. Запропонований підхід дозволяє масштабувати глибину обчислень за рахунок введення додаткових стадій, зберігаючи при цьому узгодженість моделі. Мова VHDL показала свою придатність для побудови таких структур завдяки чіткій декларативності, підтримці багатопроцесних моделей і можливості детального контролю над усіма аспектами тактування.

3.4 Розробка та моделювання багатоканального контролера пам'яті DDR SDRAM

DDR SDRAM (Double Data Rate Synchronous Dynamic Random Access Memory) є сучасним типом динамічної оперативної пам'яті з високою

пропускною здатністю, яка забезпечує передавання даних як по фронту, так і по спаду тактового сигналу. На відміну від традиційної SDRAM, модулі DDR мають вищу щільність, вимагають складнішого протоколу ініціалізації, підтримують мультибанкову архітектуру та потребують суворої відповідності до часових параметрів.

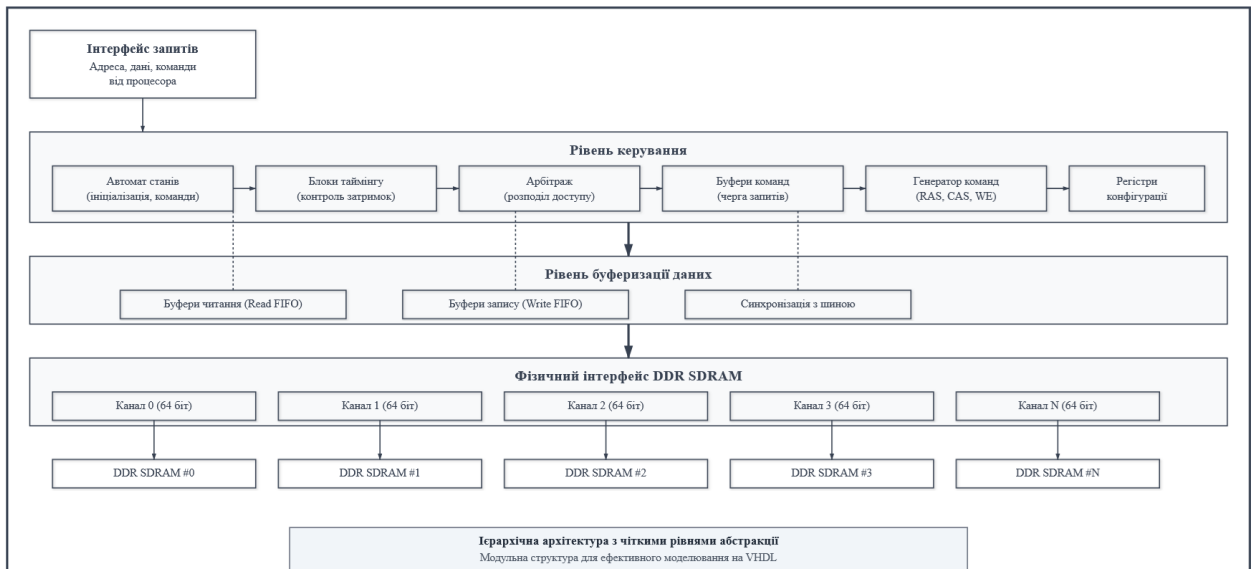


Рисунок 3.4 – Багатоконтрольний контролер пам'яті DDR SDRAM

Фізичний інтерфейс DDR включає велику кількість сигналів керування: RAS (Row Address Strobe), CAS (Column Address Strobe), WE (Write Enable), адресні та дані шини, сигнали CS, CKE, CLK, DQS, а також буфери команд. Взаємодія з таким типом пам'яті потребує реалізації складного автомата керування станами, який забезпечує правильну послідовність команд, чергування доступу до банків пам'яті, циклічне оновлення та відстеження затримок.

У додатку Б представлено спрощену модель основного блоку керування – автомата станів DDR SDRAM контролера на VHDL. Реалізація охоплює базову логіку ініціалізації та читання (рисунок 3.4).

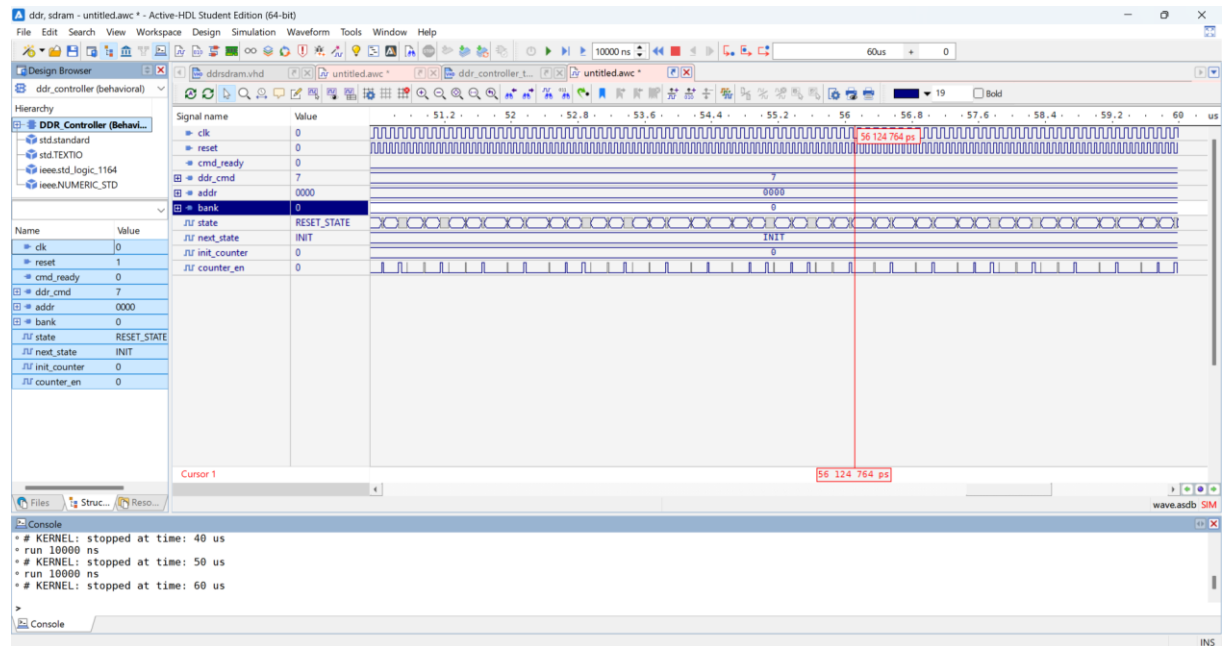


Рисунок 3.5 – Автомат станів DDR SDRAM контролера на VHDL

Ця модель (рисунок 3.5) демонструє зміну станів DDR-контролера та передачу відповідних команд на шину пам'яті. На практиці код буде доповнено детальнішими таймерами, обробкою вхідних запитів і інтерфейсом обміну даними. Сигнали `ddr_cmd` формують 3-бітовий код керування: RAS, CAS, WE.

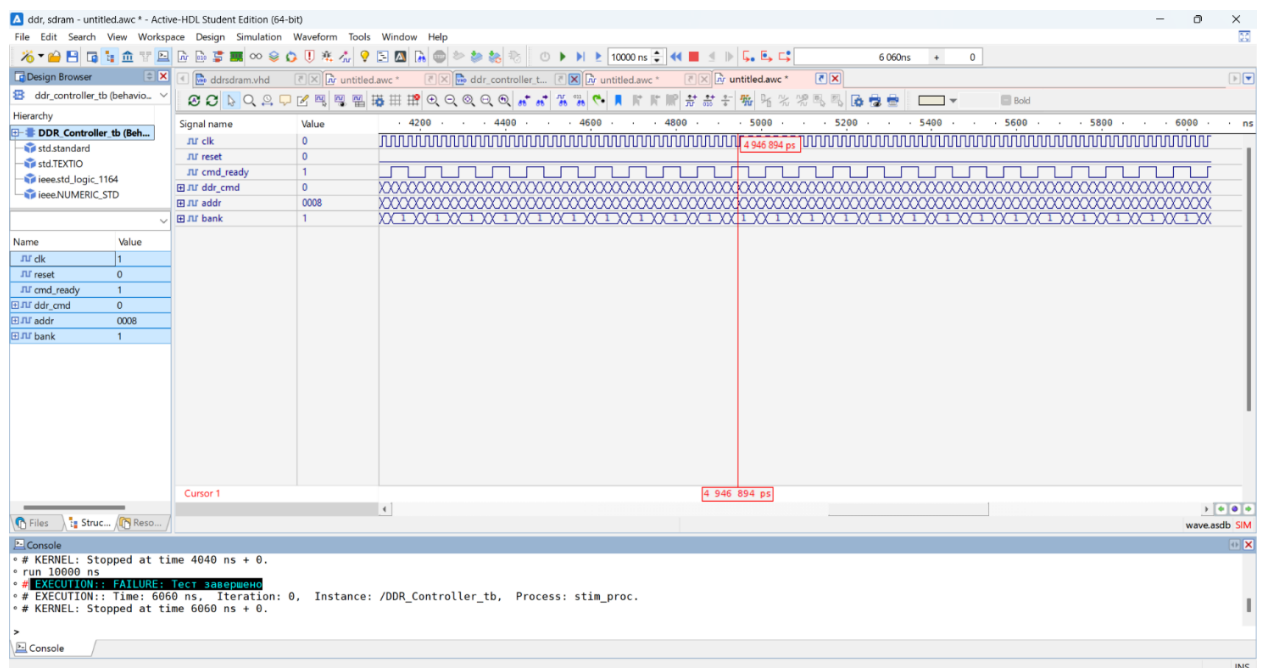


Рисунок 3.6 – Тестбенч

Результати симуляції підтвердили правильну зміну станів та коректне формування команд ініціалізації та читання. На часових діаграмах чітко прослідковується послідовність: після завершення фази RESET контролер переходить у режим INIT, де виконується затримка, необхідна для стабілізації пам'яті. Після цього контролер активує банк, задає адреси та ініціює операцію читання, а далі виконує команду PRECHARGE та переходить до періодичного оновлення.

Діаграми підтвердили, що команди ACTIVE, READ та PRECHARGE передаються у правильній послідовності з відповідними адресами та банками (рисунок 3.6). Часові обмеження між командами задовольняють стандарту DDR, а симуляція засвідчує правильну синхронізацію сигналів навіть при переході між кількома каналами. Це свідчить про потенційну масштабованість контролера до багатоекземплярної конфігурації.

Оскільки повна реалізація DDR SDRAM-контролера охоплює більший обсяг логіки, зокрема черги команд, шини обміну даними, контролери таймінгу та верифікаційні механізми, запропонована модель є основою для подальшого розширення й інтеграції в більш складну систему. Мова VHDL дозволяє реалізувати всі зазначені компоненти з високим рівнем абстракції, зберігаючи при цьому контроль над сигналами низького рівня, що робить її оптимальним інструментом для опису й верифікації подібних систем.

В умовах багатоканального доступу до пам'яті необхідно забезпечити справедливий та ефективний розподіл запитів, що надходять від різних ініціаторів. Для цього використовується арбітр, який аналізує сигнали запиту й надає дозвіл одному з каналів згідно з обраною політикою – наприклад, фіксованого пріоритету, кругового обслуговування або на основі зайнятості.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було комплексно досліджено процес моделювання складних цифрових пристроїв із використанням мови опису апаратури VHDL. Вивчено теоретичні засади, методологічні підходи та практичні засоби, що забезпечують повноцінне представлення архітектури цифрових схем, моделювання їх поведінки, верифікацію та підготовку до апаратної реалізації. Робота охопила повний цикл проектування – від постановки задачі до симуляції функціонування розроблених пристроїв у спеціалізованих середовищах.

Було проаналізовано та обґрунтовано вибір ключових типів цифрових пристроїв, які є репрезентативними для сучасних інженерних застосувань, включаючи генератори сигналів, обчислювальні модулі та контролери пам'яті. Кожен із них характеризується власною структурною складністю, архітектурними особливостями та часовими обмеженнями, що потребують індивідуального підходу до моделювання.

У процесі реалізації було розроблено моделі трьох цифрових пристроїв: цифрового частотного синтезатора (DDS), конвеєрного арифметико-логічного пристрою (ALU) та багатоканального контролера DDR SDRAM. Для кожного з них побудовано повноцінну структурну архітектуру, визначено сигнальні інтерфейси, внутрішню логіку, реалізовано автомат керування та забезпечено повну підтримку симуляції. Застосування мови VHDL дозволило забезпечити як поведінковий, так і структурний опис моделей, що сприяло високій точності відображення функціонування пристроїв та можливості подальшого синтезу для FPGA.

Усі реалізовані моделі були протестовані в середовищі Active-HDL та продемонстрували очікувану поведінку. Отримані часові діаграми підтвердили правильність логіки, відповідність сигналів інтерфейсним вимогам, а також стабільність роботи моделей у межах проєктних обмежень.

Зокрема, моделювання DDS продемонструвало генерацію періодичного сигналу із частотою, що динамічно змінюється; конвеєрний ALU успішно реалізував обробку логічних і арифметичних операцій з мінімальною затримкою; а контролер DDR SDRAM довів здатність до управління складною послідовністю команд, арбітражу доступу та підтримки паралельної роботи кількох каналів.

Проведене дослідження засвідчило доцільність і ефективність використання мови VHDL як універсального інструменту моделювання, що дозволяє проєктувати складні цифрові пристрої з урахуванням практичних обмежень і високого рівня формалізації. Побудовані моделі можуть бути інтегровані в більш складні обчислювальні системи або адаптовані до конкретних застосувань у вбудованих і промислових платформах.

Результати роботи є підтвердженням того, що вивчення принципів апаратного моделювання та володіння мовами HDL становить основу для підготовки висококваліфікованих фахівців у галузі цифрової електроніки, автоматизації та систем на кристалі. Робота відкриває перспективи подальших досліджень у напрямі оптимізації синтезованих структур, автоматичного генерування HDL-коду, впровадження засобів верифікації на основі формальних методів і застосування високорівневого синтезу в рамках концепції hardware/software co-design.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Proakis, J.G., Manolakis, D.G. Digital Signal Processing: Principles, Algorithms, and Applications. Pearson Education. 2007. 1081 p.
2. Smith, S.W. The Scientist and Engineer's Guide to Digital Signal Processing. California Technical Publishing. 1997. 650 p.
3. Chu, P.P. FPGA Prototyping by VHDL Examples: Xilinx Spartan-3 Version. Wiley. 2008. 728 p.
4. Pedroni, V.A. Circuit Design with VHDL. MIT Press. 2004. 528 p.
5. Harris, D.M. & Harris, S.L. Digital Design and Computer Architecture. Morgan Kaufmann. 2012. 712 p.
6. Meyer-Baese, U. Digital Signal Processing with Field Programmable Gate Arrays. Springer. 2014. 509 p.
7. Tokheim, R.L. Digital Electronics: Principles and Applications. McGraw-Hill. 2013. 688 p.
8. MathWorks. Signal Processing Toolbox User's Guide. The MathWorks, Inc. 2024. [Online Resource]
9. Xilinx Inc. Vivado Design Suite User Guide: High-Level Synthesis. UG902. 2023. [Online Resource]
10. IEEE Standard 1076-2008. IEEE Standard VHDL Language Reference Manual. IEEE. 2008. 626 p.