

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Метод та засоби проектування аналогово-цифрових та цифро-аналогових систем на базі ПЛІС»

Виконав: студент 2 курсу, групи 2КІ-20м
напряму підготовки (спеціальності)
123 — «Комп'ютерна інженерія»
Гандзюк І.В.

Керівник: к.т.н., доцент каф. ОТ
Богомолів С.В.

«___» _____ 2021 р.

Опонент: к.т.н., доцент каф. МБІС
Карпинець В.В.

«___» _____ 2021 р.

Допущено до захисту
Завідувач кафедри ОТ
д.т.н., проф. Азаров О.Д.
«___» _____ 2021 р.

Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра обчислювальної техніки
Освітньо-кваліфікаційний рівень магістр
Спеціальність 123 — «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри
обчислювальної техніки
_____ проф., д.т.н. О.Д. Азаров

« ___ » _____ 2021 р.

З А В Д А Н Н Я **НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ** Гандзюку Івану Володимировичу

1 Тема роботи «Метод та засоби проектування аналогово-цифрових та цифро-аналогових систем на базі ПЛІС»

керівник роботи Богомолів Сергій Віталійович, к.т.н., доцент,
затверджені наказом вищого навчального закладу від 24.09.2021 р. №227

2 Строк подання студентом роботи 15.12.2021 р.

3 Вихідні дані до роботи — дані про різновиди програмованих логічних інтегральних схем та побудову на їх базі цифрових пристроїв.

4 Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити): вступ, аналіз особливостей програмованих логічних інтегральних схем, порівняльний аналіз і обґрунтування вибору апаратних та програмних засобів проектування, проектування процесора аналогово-цифрових та цифро-аналогових систем на базі пліс, висновки, перелік джерел посилання.

5 Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень): datasheet зі зображеннями контактів, що відповідають за з'єднання перемикачів, світлодіодів та кнопок, datasheet зі зображеннями контактів, що відповідають за з'єднання за з'єднання 7-сегментних дисплеїв та тактового генератора, зовнішній вигляд плати ПЛІС Altera DE0, схема простого процесора, зображення утиліти Pin Planner.

6 Консультанти розділів роботи представлені в таблиці 1.

Таблиця 1 — Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1,2,3	Богомолів С.В., к.т.н., доцент		
4	Лесько О. Й., к.е.н., професор		

7 Дата видачі завдання 07.09.2021 р.

8 Календарний план наведено в таблиці 2.

Таблиця 2 – Календарний план

№	Назва етапів виконання магістерської роботи	Строк виконання етапів роботи	Примітка
1	Постановка задачі роботи	05.10.21	
2	Аналіз особливостей програмованих логічних інтегральних схем	06.10-13.10	
3	Огляд аналогів розроблюваного пристрою	14.10-22.10	
4	Порівняння ПЛІС від основних виробників	25.10-5.11	
5	Створення схеми та програми пристрою	8.11-16.11	
6	Підготовка матеріалів та опис розробки	17.11-30.11	
7	Оформлення пояснювальної записки та ілюстративного матеріалу	01.12-11.12	
8	Аналіз виконання роботи, висновки, додатки	13.12-14.12	
9	Перевірка якості виконання магістерської роботи та усунення недоліків	15.12.21	

Студент

Гандзюк І.В.

Керівник роботи

Богомолів С.В.

АНОТАЦІЯ

УДК 004.9

Гандзюк І.В. Метод та засоби проектування аналогово-цифрових та цифро-аналогових систем на базі ПЛІС. Магістерська кваліфікаційна робота зі спеціальності 123 — комп'ютерна інженерія, освітня програма — комп'ютерна інженерія. Вінниця: ВНТУ, 2021, 83 с.

На укр.мові. Бібліогр.: 22 назв, рис. 41, табл. 12.

Дана магістерська кваліфікаційна робота присвячена дослідженню методу та засобів проектуванні цифро-аналогових та аналого-цифрових систем на базі програмованих логічних інтегральних схем.

В роботі виконаний аналіз особливостей програмованих логічних інтегральних схем, проведений огляд основних ПЛІС від світових виробників та програмного забезпечення для них.

В результаті роботи спроектований простий процесор на основі ПЛІС у програмному забезпеченні Quartus II з моделюванням в утиліті ModelSim.

Ключові слова: ПЛІС, програмована логічна інтегральна схема, цифрові пристрої, soft-процесор, процесор, аналого-цифрова система, цифро-аналогова система

ANNOTATION

Handziuk I. Method and means for designing analog-digital and digital-analog systems based on FPGA. Master's thesis in specialty 123 — computer engineering, educational program — computer engineering. Vinnytsia: VNTU, 2021, 83 p.

In Ukrainian. Bibliogr .: 22 titles, fig. 41, table. 12.

This master's qualification work is devoted to the study of methods and tools for designing digital-analog and analog-digital systems based on programmable logic integrated circuits.

The paper analyzes the features of programmable logic integrated circuits, reviews the main FPGAs from global manufacturers and software for them.

As a result of work the simple processor on the basis of FPGA in the Quartus II software with modeling in the ModelSim utility is designed.

Keywords: FPGA, Field-Programmable Gate Array, digital devices, soft-processor, processor, analog-digital system, digital-analog system.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	9
ВСТУП.....	10
1 АНАЛІЗ ОСОБЛИВОСТЕЙ ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ.....	13
1.1 Передумови розвитку ПЛІС.....	13
1.1.1 Альтернативи дискретної логіки. Переваги ПЛІС	14
1.1.2 Програмно-апаратні комплекси на базі ПЛІС	15
1.2 Soft-мікропроцесори	16
1.3 Класифікація ПЛІС	18
1.4 Додаткові типи блоків ПЛІС типу FPGA	22
2 ПОРІВНЯЛЬНИЙ АНАЛІЗ І ОБГРУНТУВАННЯ ВИБОРУ АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ПРОЕКТУВАННЯ	29
2.1 Огляд та порівняння основних світових виробників ПЛІС.....	29
2.1.1 Altera.....	29
2.1.2 Xilinx.....	33
2.1.3 Actel Corporation (Microchip)	35
2.2 Вибір середовища програмування.....	37
2.2.1 Libero	38
2.2.2 Програмне забезпечення Quartus II.....	39
3 ПРОЕКТУВАННЯ ПРОЦЕСОРА АНАЛОГОВО-ЦИФРОВИХ ТА ЦИФРО-АНАЛОГОВИХ СИСТЕМ НА БАЗІ ПЛІС	44
3.1 Порівняння методів проектування систем на ПЛІС.....	44

						<i>08-23.МКР.019.00.000 ПЗ</i>		
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Гандзюк І.В.			Метод та засоби проектування аналогово-цифрових та цифро-аналогових систем на базі ПЛІС. Пояснювальна записка	Літ.	Арк.	Аркушів
Керівник.		Богомолов С.В.					6	83
Реценз.		Карпінєць В.В.				ВНТУ, гр. 2К1-20м		
Н. Контр.		Швець С. І.						
Затверд.		Азаров О.Д.						

3.2	Метод побудови процесорів на базі ПЛІС	48
3.3	Реалізація процесора на базі ПЛІС.....	51
4	РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ ПРОЕКТУВАННЯ	
	ПРОЦЕСОРА НА ОСНОВІ ПРОГРАМОВАНОЇ ЛОГІЧНОЇ	
	ІНТЕГРАЛЬНОЇ СХЕМИ	57
4.1	Проведення комерційного та технологічного аудиту науково-технічної розробки	58
4.2	Розрахунок витрат на здійснення науково-дослідної роботи.....	62
4.2.1	Витрати на оплату праці.....	62
4.2.2	Відрахування на соціальні заходи	64
4.2.3	Сировина та матеріали.....	64
4.2.4	Розрахунок витрат на комплектуючі.....	65
4.2.5	Спецустаткування для наукових (експериментальних) робіт	65
4.2.6	Програмне забезпечення для наукових (експериментальних) робіт	65
4.2.7	Амортизація обладнання, програмних засобів та приміщень	65
4.2.8	Паливо та енергія для науково-виробничих цілей	66
4.2.9	Службові відрядження.....	67
4.2.10	Витрати на роботи, які виконують сторонні підприємства, установи і організації	67
4.2.11	Інші витрати.....	67
4.2.12	Накладні (загальновиробничі) витрати.....	67
4.3	Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором	68

ВИСНОВКИ	73
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	74
ДОДАТОК А. Технічне завдання	77
ДОДАТОК Б. Зображення утиліти Pin Planer	80
ДОДАТОК В. Зображення фрагментів Datasheet для плати ПЛІС Altera DE0.....	81
ДОДАТОК Г. Зовнішній вигляд програмованої логічної інтегральної схеми.....	83
ДОДАТОК Д. Схема простого процесора	84
ДОДАТОК Е. Протокол перевірки навчальної (кваліфікаційної) роботи	85

					<i>08-23.МКР.019.00.000 ПЗ</i>	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БМК — Базовий матричний кристал

ВІС — Велика інтегральна схема

ЕП — Електрична принципова схема

ІМС — Інтегральні мікросхеми

КМОН — Комплементарна структура метал-оксид-напівпровідник

ЛЕ — Логічні елементи

НВІС — Надвелика інтегральна схема

ПЗ — Програмне забезпечення

ПЗП — Постійний запам'ятовуючий пристрій

ПЛІС — Програмована логічна інтегральна схема

САПР — Система автоматизованого проєктування і розрахунку

ТТЛ — Транзисторно-транзисторна логіка

ФОК — Функціонально-орієнтовані контролери

ШІМ — Широтно-імпульсна модуляція

ASIC — Application-specific integrated circuit

ATX — Advanced Technology Extended

DE — Development and Education

RAM — Random access memory

SOC — Systems on chip

VHDL — VHSIC-HDL, Very high speed integrated circuit hardware description language

VWF — Vector waveform file

ВСТУП

При розробці мікроконтролерних систем актуальною задачею являється оптимізація проектування, а саме мінімізація трудовитрат розробника при збереженні функціоналу кінцевого продукту, для цього необхідно вибрати відповідні моделі, метод і засоби проектування (архітектури мікроконтролера, середовище розробки та інше [1]). Одним з інструментів для розробки спеціалізованих обчислювачів являється програмована логічна інтегральна схема (ПЛІС). Їх застосування дозволяє суттєво спростити проектування, налагодження і реалізацію апаратних обчислювальних засобів та їх алгоритмів. Використання ПЛІС надає широкі можливості для вибору оптимального рішення задачі.

Одним з актуальних підходів до вирішення завдання проектування вбудованих систем управління — це застосування функціонально-орієнтованих контролерів (ФОК) [2]. ФОК представляють собою спеціалізовані апаратно-програмні комплекси, які дають найбільш оптимальне рішення вузько направлених задач. Характерна риса ФОК — це наявність нетипових програмних та апаратних модулів, зокрема криптографічних блоків, співпроцесорів чи вузлів RFID, застосування яких потрібне для реалізації різних комп'ютерних систем. Використання спеціалізованих модулів дозволяє підвищити швидкість розробки, її надійність та ефективність отриманого рішення.

У цій роботі розглядається підхід до проектування ФОК, який полягається у використанні мікросхеми програмованої логіки в складі системи. Поєднання програмної і апаратної (ПЛІС) частин у системі дає широкі можливості проектувальнику для вибору оптимального вирішення задачі. Такий підхід представлений на ринку розробками різних фірм, зокрема, Altera[3] та Xilinx[4], Cypress та інші. Система на базі ПЛІС активно використовуються в таких областях, як бортові системи управління, системи зв'язку, системи збору даних, у космічній та військовій галузі та інших.

Подальший розвиток цього класу систем представляється в підвищенні рівня абстракції апаратних і програмних частин системи за рахунок застосування більш розвинених засобів проектування. Даний напрямок характеризується пов-

ним поділом процесів реалізації спеціалізованих апаратних модулів на ПЛІС, програмування мікроконтролера і проектування системи в цілому. Завдяки цьому процес реалізації ФОК стає більш прозорим, зрозумілим і швидким.

Виходячи із розглянутого, подальший процес реалізації цифро-аналогових та аналогово-цифрових систем на основі ПЛІС, є **актуальною** задачею. Значна сфера застосування ПЛІС вказує на актуальність її використання та вимагає дослідження.

Метою дослідження є підвищення швидкості розробки та оптимізації налагодження аналого-цифрових і цифро-аналогових систем, їх окремих елементів і вузлів, шляхом використання запропонованого методу та існуючих засобів проектування систем на основі ПЛІС.

Задачі дослідження:

- провести порівняльний аналіз ПЛІС, НВІС та мікроконтролерів;
- проаналізувати програмно-апаратні комплекси на базі ПЛІС;
- оглянути програмні та апаратні засоби проектування процесорів на базі ПЛІС;
- створити схему та програму пристрою з його модифікацією для оптимізації процесу виробництва та експлуатації;
- провести симуляцію роботи пристрою за допомогою програмного середовища.

Об'єкт дослідження — процеси та засоби проектування цифрових блоків ЦА і АЦ систем з використанням плати ПЛІС.

Предмет дослідження — методи та засоби проектування цифрових блоків цифро-аналогових та аналого-цифрових систем на базі програмованих логічних інтегральних схем.

Наукова новизна магістерської кваліфікаційної роботи полягає у вдосконаленні методу проектування цифрових блоків ЦА та АЦ систем на базі ПЛІС, що полягає у комплексному поєднанні графічного та програмного методів, який враховує специфіку блоків. Це дає можливість оптимально використовувати ресурси ПЛІС та створювати швидкодіючу систему.

Практична цінність роботи полягає у застосуванні запропонованого методу проектування цифрових блоків ЦА та АЦ систем на базі ПЛІС.

Апробація результатів магістерської роботи здійснена під час доповіді на І науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії ВНТУ. Опубліковано доповідь на конференції Молодь в науці: дослідження, проблеми, перспективи (МН-2022) Матеріали доповіді подані у [5, 11].

1 АНАЛІЗ ОСОБЛИВОСТЕЙ ПРОГРАМОВАНИХ ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ

1.1 Передумови розвитку ПЛІС

З часу появи перших інтегральних, й до сьогодні одним з основних напрямків цифрової схемотехніки являється постійне збільшення кількості логічних елементів (ЛЕ), що розміщуються на кристалі, при зменшенні затрачуваної потужності. Збільшення кількості логічних елементів дає можливість для проектування більш складних систем, що розміщуються на одному кристалі.

Даний напрям розвитку проявляється у переході від інтегральних мікросхем малого рівня інтеграції до надвеликих інтегральних схем (НВІС). Можливість створення перших мікропроцесорів, являється результатом потужного розвитку схемотехніки. Застосування мікропроцесорів дозволило більш широко використовувати цифрові технології при обробці інформації у різних сферах.

Основними результатами розвитку тенденції стало збільшення надійності та зменшення габаритів та споживаної потужності.

Використання мікропроцесорів не може виконувати усі практичні завдання цифрової схемотехніки. Ця проблема є через те, що всім мікропроцесорам для вирішення завдання потрібно виконати послідовність певних кроків кінцевої тривалості, проте для вирішення багатьох завдань (наприклад, пов'язаних із забезпеченням роботи самих мікропроцесорів) потрібні пристрої з мінімальною затримкою виконання логічних функцій. Основних способів вирішення цієї проблеми є три:

- використання наборів стандартної дискретної цифрової логіки загального застосування, наприклад ТТЛ чи КМОН;
- використання замовних НВІС;
- використання програмованих логічних інтегральних схем (ПЛІС).

Набори дискретної цифрової логіки різних серій тривалий час були основою елементної бази розробки цифрових пристроїв. До складу таких наборів входить велика кількість окремих мікросхем, призначених для виконання базових логічних функцій (І, АБО, НІ) і для виконання функцій типових цифрових пристроїв,

таких як тригери, регістри, лічильники, мультиплексори, дешифратори та інші, що дозволяє використовувати їх при розробці більш складних функціонально закінчених цифрових пристроїв.

Основний недолік дискретної логіки полягає в тому, що для розробки кінцевих пристроїв зазвичай потрібна велика кількість мікросхем. Наслідком цього є велика кількість зовнішніх з'єднань, складність конструкції та великі габарити друкованих плат, велика довжина сполучних провідників, складність побудови пристроїв з високою тактовою частотою і низька надійність. Для зменшення кількості мікросхем при проектуванні мікропроцесорних систем було розроблено ряд периферійних ВІС, що являють собою спеціалізовані цифрові пристрої, призначені для виконання деяких типових функцій у складі мікропроцесорних систем, такі як контролери динамічної RAM, контролери переривань, контролери прямого доступу в пам'ять, контролери шин та інші. Проте застосування периферійних ВІС не дозволяє повністю подолати основні недоліки дискретної цифрової логіки.

1.1.1 Альтернативи дискретної логіки. Переваги ПЛІС

Найбільш кардинально проблема швидкодії, габаритів, спрощення друкованих плат та надійності вирішується за рахунок виготовлення замовних НВІС. Такий варіант економічно доцільний тільки за велико серійного виробництва однотипних виробів, внаслідок високої вартості та довгого часу підготовки виробництва замовних НВІС. Також, при використанні замовних НВІС можлива модифікація виробів вимагатиме суттєвих додаткових матеріальних та тимчасових витрат [7].

Окрім цього, досить часто потрібно створювати цифрові пристрої і вироби, які не розрахованих на велико серійне виробництво, в такому випадку розробка замовних НВІС не підходить або з економічних причин, або не підходить через час виконання. Протягом довгого часу єдиним способом для вирішення таких завдань являлось використання інтегральних мікросхем дискретної логіки, тому що можливості перших поколінь простих ПЛІС були суттєво обмежені, а ціна на

більш функціональні ПЛІС була дуже високою, також були певні складнощі з проектуванням цифрових пристроїв на основі ПЛІС.

На протязі останніх років, стався різкий прорив як у технології виготовлення ПЛІС, так і в розробці інструментальних засобів, які призначені для проектування ЦА та АЦ систем на ПЛІС та виготовленні готових виробів. Даний прорив характеризується різким збільшенням кількості базових елементів цифрової схеми, що розміщуються на одному кристалі, підвищенням робочої частоти з різким одночасним зниженням як енергоспоживання, так вартості плат.

1.1.2 Програмно-апаратні комплекси на базі ПЛІС

Мікросхеми ПЛІС складаються з модулів подібних цифрових вузлів, які включають логічні блоки, блок вхід/вихід та спеціалізовані компоненти, які з'єднуються за допомогою блоків комутації. Такі зв'язки програмовані, за рахунок чого досягається можливість створення систем, завдяки чому можна реалізувати практично будь-який цифровий пристрій. ПЛІС у порівнянні з інтегральними схемами спеціального призначення (ASIC) мають деякі недоліки.

ASIC менші й споживають менше енергії у порівнянні з ПЛІС. Виробники ПЛІС вдосконалюють свої продукти в таких напрямках, але логічні комірки та пам'ять в ASIC є більш ресурсозатратним, у порівнянні з апаратною реалізацією логіки, якою вони будуть запрограмовані. Також блоки комутації містять більшу кількість з'єднань, ніж потрібно для з'єднання логічних елементів. На рисунку 1.1 наведений приклад ASIC.

З недоліків ASIC можна виділити те, що вони можуть бути перепрограмовані тільки на рівні ПЗ, і тільки за умови, що вони містять процесор. ПЛІС програмується на логічному рівні, тому надають більш швидкі та динамічні рішення, які недоступні у звичайних процесорах. Також заміна ASIC з метою виправлення помилок після використання пристрою дуже не вигідна в економічному плані. В цьому випадку можна виправити помилки за допомогою оновлення АЗ, але такі можливості суттєво обмежені. Також ASIC мають обмеження швидкостей, які можуть бути подолані при використанні ПЛІС.



Рисунок 1.1 — ASIC периферійного контролера USB 2.0

Основними фірмами-виробниками ПЛІС в даний час є такі фірми, як «Altera», «Atmel», «Cypress», «Lattice», «Lucent» та «Xilinx». Далі буде розглянуто основні з них.

1.2 Soft-мікропроцесори

Гнучкість і простота розробки програмного забезпечення для різних процесорів стала причиною створення багатьох з пристроїв на мовах опису при використанні в ПЛІС [10]. Soft-мікропроцесори є гнучкими в плані технічної реалізації, тому що єдиним обмеженням функціональності в цих процесорах є ресурсне обмеження ПЛІС. Це дає можливість розробляти процесори будь-якої архітектури з різними додатковими функціональними можливостями. Також ці параметри чітко задані на звичайних інтегральних схемах.

Виробники ПЛІС досить часто надають свою розробку soft-мікропроцесорів, які оптимізовані до власних продуктів. Такі процесори мають невеликі покращення, завдяки яким досягаються більш висока продуктивність, зменшені вимоги до ресурсів та інші переваги. Наприклад, до таких процесорів

відносяться Xilinx microBlaze, Altera Cyclone III та Altera NIOS II. Altera NIOS II — це 32-бітовий процесори, але є варіанти 8- ти 16-бітових процесорів [11].

Процесори на базі ПЛІС мають багато переваг у порівнянні зі звичайними мікроконтролерами. Розробники можуть сконфігурувати параметри процесорів, зокрема, кешування або захист пам'яті, вибравши відповідні прапорці перед компіляцією. Також, ПЛІС можуть реалізовувати більш хорошу та функціональну периферію; ці системи є гнучкими як в апаратній частині, так і в частині ПЗ ПЛІС з флеш-пам'яттю можна запусити практично так само швидко, як і мікроконтролер.

Одним з прикладів використання soft-мікропроцесора є процесор Nios II, структура якого показана на рисунку 1.2.

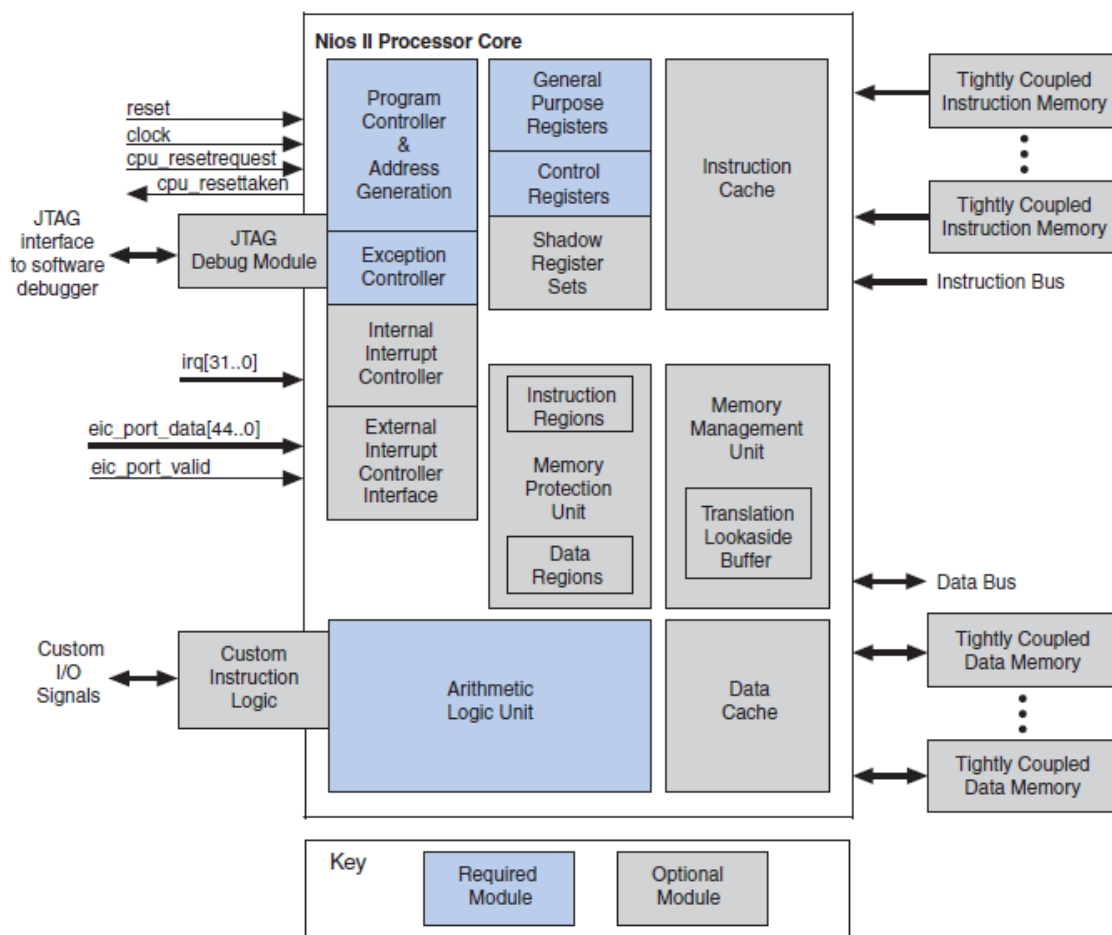


Рисунок 1.2 — Структурна схема процесора Nios II [12]

Дане сімейство процесорів має архітектуру RISC, яка адаптована під моделі ПЛІС Altera Cyclone, Arria, Stratix та HardCopy. Основні характеристики процесорів Nios II:

- розрядність шини адреси і шини даних 16 або 32 біта;
- 5-етапний конвеєр;
- пам'ять на кристалі та зовнішня пам'ять;
- 512 регістрів для загального призначення;
- 64 вектори переривань.

Розробка системи на базі процесора Nios II ведеться в двох напрямках: розробка апаратних засобів для ПЛІС та розробка програмного забезпечення процесора. Програмне середовище Quartus II має оболонку QSYS, яка дає можливість спроектувати систему, яка поділяється на рівні створення і з'єднання компонентів. QSYS дозволяє утворювати автоматизоване з'єднання компонентів, базуючись на стандартних інтерфейсах до шини передачі даних, які з'єднують всю систему. Сьогодні доступні реалізації шин Avalon (Altera) та AMBA AXI (ARM).

Використання процесора на базі ПЛІС дає можливість гнучкого налаштування його функціональних можливостей. У випадку з Nios II, розробнику доступні для вибору доступні такі параметри, як тип процесора, тип контролера переривань, розмір кешу та інші.

1.3 Класифікація ПЛІС

На відміну від стандартних цифрових мікросхем, логіка роботи ПЛІС не визначається при виготовленні, а задається за допомогою програмування. Для програмування використовуються програматор та IDE (налагоджувальне середовище), що дозволяють задати бажану структуру цифрового пристрою у вигляді принципової електричної схеми або програми спеціальними мовами опису апаратури.

PLA (Programmable Logic Arrays) — програмована логічна матриця. Основна ідея роботи ПЛІМ полягає у реалізації логічної функції, представленої в ДДНФ — диз'юнктивній нормальній формі. У програмованій логічній матриці обидві мат-

риці логічних елементів "І" та "АБО" програмуються. На рисунку 1.3 наведена узагальнена структура програмованих логічних матриць.

PAL (Programmable Array Logic) — програмована матриця логіки. На відміну від ПЛМ, ПМЛ матриця логічних елементів "І" є програмованою, а матриця логічних елементів "АБО" є фіксованою. Оскільки виготовити такі пристрої простіше, ніж ПЛМ, вони мають меншу вартість і мають покращені характеристики, що призвело до їх високої популярності. На рисунку 1.4 наведена структура програмованих матриць логіки.

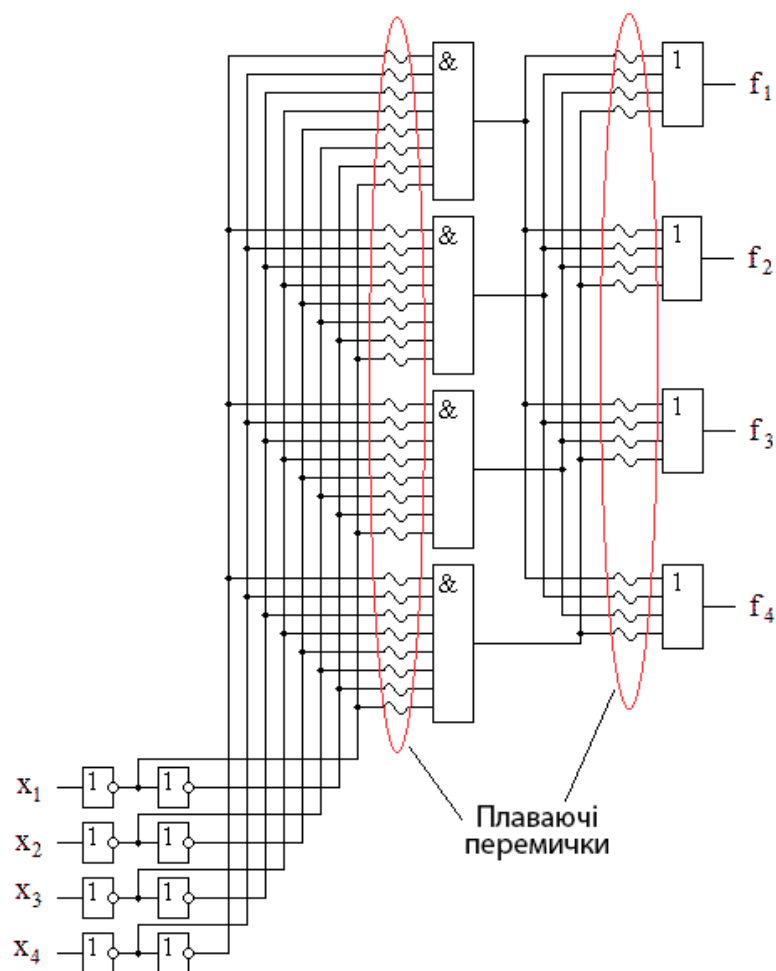


Рисунок 1.3 — Узагальнена структура програмованих логічних матриць (ПЛМ)

GAL (Gate Array Logic) — ПЛІС, що мають програмовану матрицю «І» та фіксовану матрицю «АБО».

CPLD (Complex Programmable Logic Device) — програмована логічна інтегральна схема CPLD складається з декількох макрокомірок, розташованих на одному кристалі. Кожна з комірок з'єднана з блоками введення-виводу, що здійснюють формування необхідного виду входів або виходів для роботи із зовнішніми схемами. Крім того, всі макрокомірки та блоки введення-виводу пов'язані між собою внутрішніми паралельними шинами.

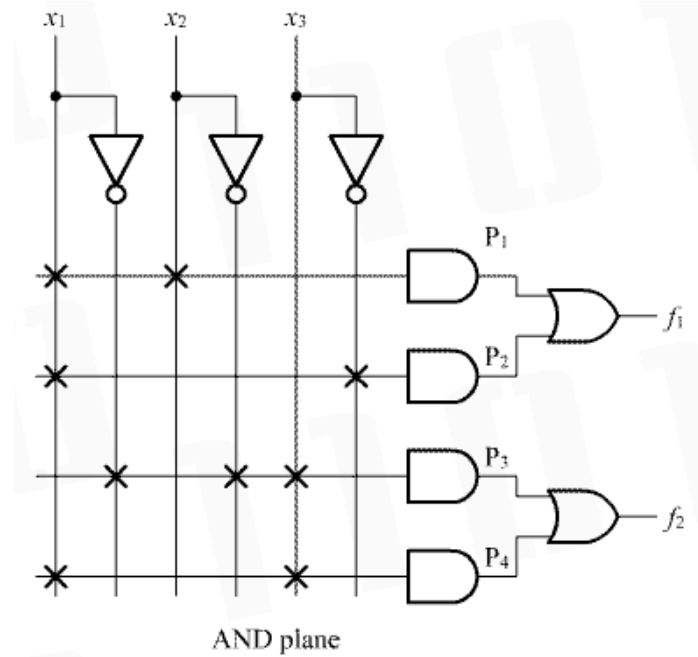


Рисунок 1.4 — Узагальнена структура програмованих матриць логіки (ПМЛ)

На рисунку 1.5 наведена мікросхема CPLD, яка складається з чотирьох комірок, що пов'язані між собою внутрішніми шинами і з'єднуються з блоками вводу-виводу.

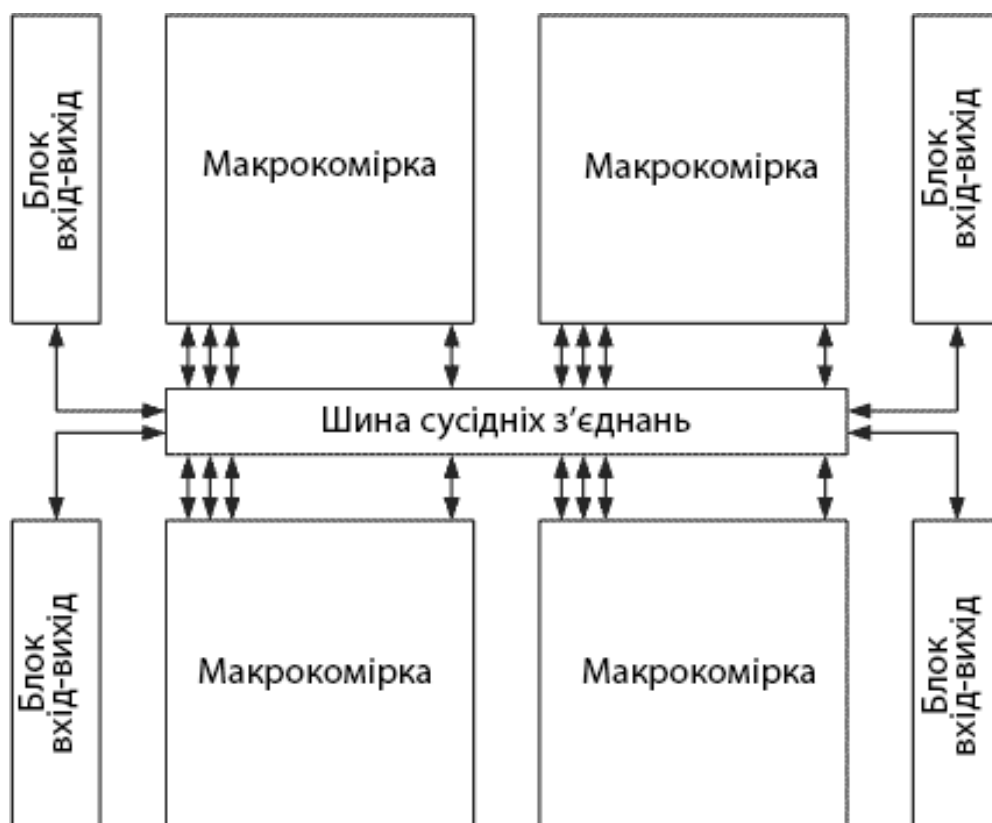


Рисунок 1.5 — Приклад внутрішньої схеми CPLD

Комірки побудовані подібно до ПЛМ мікросхеми, до якої на виході підключений D-тригер. До недоліків можна віднести те, що важко забезпечити ефективне застосування всіх макрокомірок. Завжди частина комірок залишається невикористовуваною. Часто з комірок використовується тільки тригер або логічний елемент "2І" ("2АБО"). Решта схеми дарма займає площу кристала та споживає струм від джерела живлення.

FPGA (Field-Programmable Gate Array). Принцип роботи FPGA суттєво відрізняється від принципу роботи CPLD. FPGA є спадкоємцем комбінаційних схем, реалізованих на постійних запам'ятовуючих пристроях (ПЗП). Узагальнена структура мікросхем FPGA наведена на рисунку 1.6 [20].

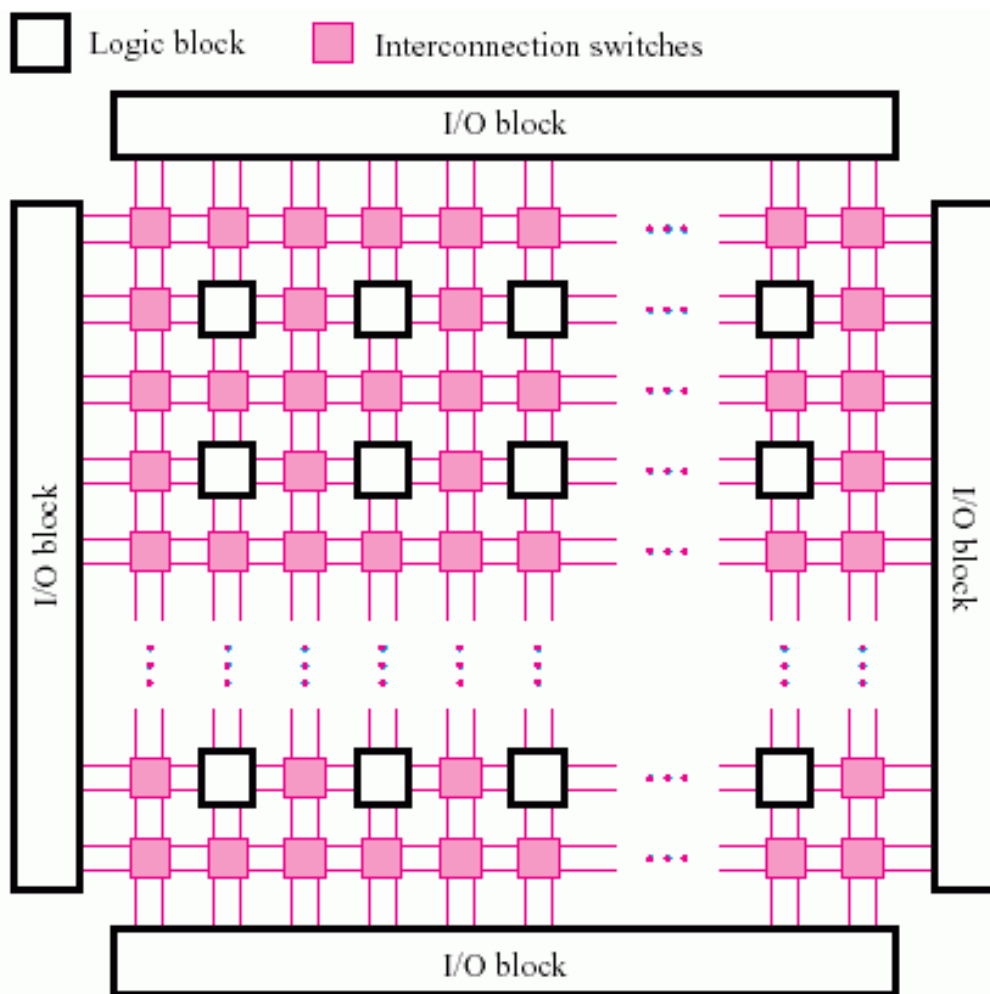


Рисунок 1.6 — Узагальнена структура мікросхем FPGA

1.4 Додаткові типи блоків ПЛІС типу FPGA

Блоки вбудованих помножувачів (MUL) і помножувачів-акумуляторів (MAC-DSP48). Хоч і основним логічним ресурсом ПЛІС типу FPGA є КЛБ, що складаються з секцій, що в свою чергу складаються з логічних комірок (ЛК) (LUT), сучасні ПЛІС мають також і вбудовані ядра, що представляють собою окремі типові вузли, пристрої та навіть мікропроцесори. Одним із таких типових вбудованих ядер є блоки помножувачів (MUL). Наприклад, у дешевих ПЛІС серії СПАРТАН-3 це помножувачі двох 18 розрядних чисел. Таких помножувачів залежно від складності мікросхеми ПЛІС від кількох десятків до сотень. Для ПЛІС, орієнтованих на реалізацію систем цифрової обробки сигналів (Digital Signal Processing-DSP), де типовою операцією є множення з накопиченням (MAC-Multiply And Accumulate) — це програмовані вбудовані ядра з помножувачами 25

розрядних чисел на 18-розрядні та з 48 розрядними трьохходовими суматорами-накопичувачами (блоки DSP48 ПЛІС серій VIRTEX-4,5,6 фірми XILINX).

Вбудовані ядра RAM та FIFO. Для реалізації ПЛІС типу FPGA модулів RAM передбачено дві можливості. Першу можливість надає кожна ЛК, яка при 4-х входних ЛК може бути сконфігурована як синхронна RAM ємністю в 16 одно-розрядних чисел (16 x 1). Дві сусідні ЛК можуть бути налаштовані як 16 x 1 дво-портових RAM із записом та читанням за однією адресою та читанням за іншою адресою. Якщо потрібно мати RAM більшого об'єму, воно будується з урахуванням кількох логічних осередків. Така RAM розподілена за площею мікросхеми ПЛІС і тому названа розподіленою (Distributed RAM). Другу можливість реалізації RAM та ПЗП у ПЛІС надають вбудовані ядра — блоки синхронної пам'яті (Block RAM), їх зображення наведено на рисунку 1.7. Визначення «синхронна пам'ять» означає, що вона спрацьовує після приходу синхроімпульсу та вміст осередку, адреса якої збігається зі значенням коду на регістрі адреси, з'являється на виході при зчитуванні або змінюється під час запису.

Інформаційна ємність кожного блоку пам'яті для різних серій ПЛІС FPGA коливається від 16 до 36 кбіт. Вони можуть бути налаштовані в межах цієї ємності як однопортова або двопортова RAM різної розрядності та кількості слів. Початковий стан цієї пам'яті визначається при програмуванні ПЛІС, тому вона може бути використана також як ПЗП. У серіях VIRTEX-5 і 6 ці блоки можна конфігурувати і як пам'ять типу черга-FIFO з різними синхросигналами на входах читання та запису, що дозволяє їх використовувати як буфер на межах областей синхронізації.

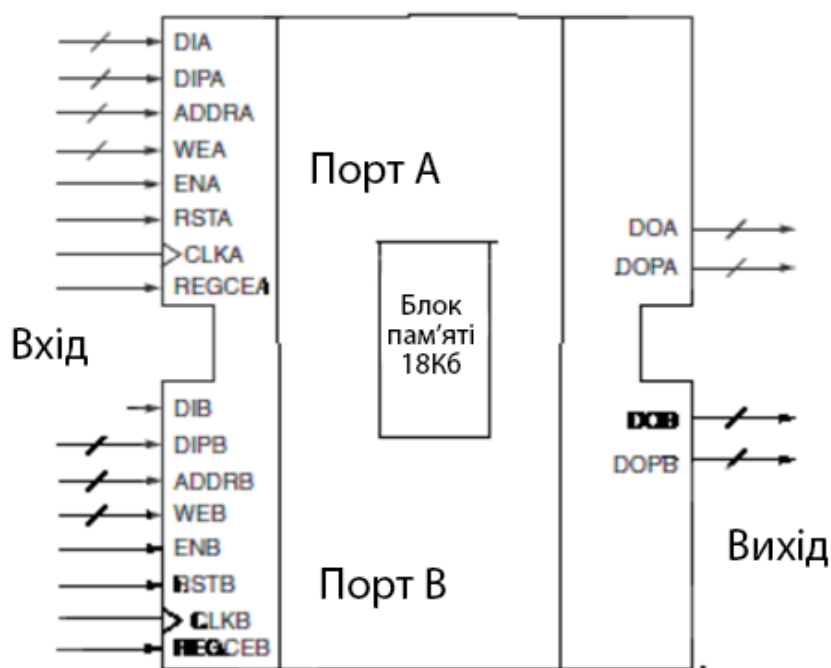


Рисунок 1.7 — Умовне позначення блока синхронної пам'яті (Block RAM)

Блоки введення-виводу (БВВ). Блоки введення-виводу (БВВ) розташовані по периферії мікросхеми ПЛІС. Вони визначаються кількістю контактів мікросхеми, крім контактів живлення, землі та інших. Особливістю БВВ є можливість бути налаштованими на різні (більше 10) сигнальні стандарти: рівні сигналів, опори джерел, методи передачі тощо. Серед цих стандартів:

- диференціальні (двохпровідні) сигнальні стандарти: LVDS, BLVDS, ULVDS — (LVDS — низьковольтний диференціальний);
- однопровідні низьковольтні стандарти: ТТЛ та КМОП — LVTTL, LVCMOS(3.3,2.5,1.8,1.2 V);
- шинні стандарти: PCI, PCI-X та інші.

Програмовані опори (DCI-Digital Controlled Impedance), що є в БВВ, дозволяють спростити вирішення завдання узгодження опорів джерел та приймачів сигналів.

Блоки керування частотою та фазуванням синхросигналів. У ПЛІС FPGA кілька контактів з'єднані зі спеціальними вхідними буферами (Clock Buffer), які у свою чергу пов'язані з мережами розподілу синхросигналів мікросхемою. Наприклад, у ПЛІС Spartan-3 є до 8 глобальних тактових входів — GCLK0-GCLK7. Для

управління частотою, затримкою та фазою тактових сигналів призначені програмовані блоки DCM (Digital Clock Manager) та Clock Management Tiles (CMT), виходи яких можна підключати до тактових буферів. Цифровий блок керування синхросигналами наведений на рисунку 1.8.

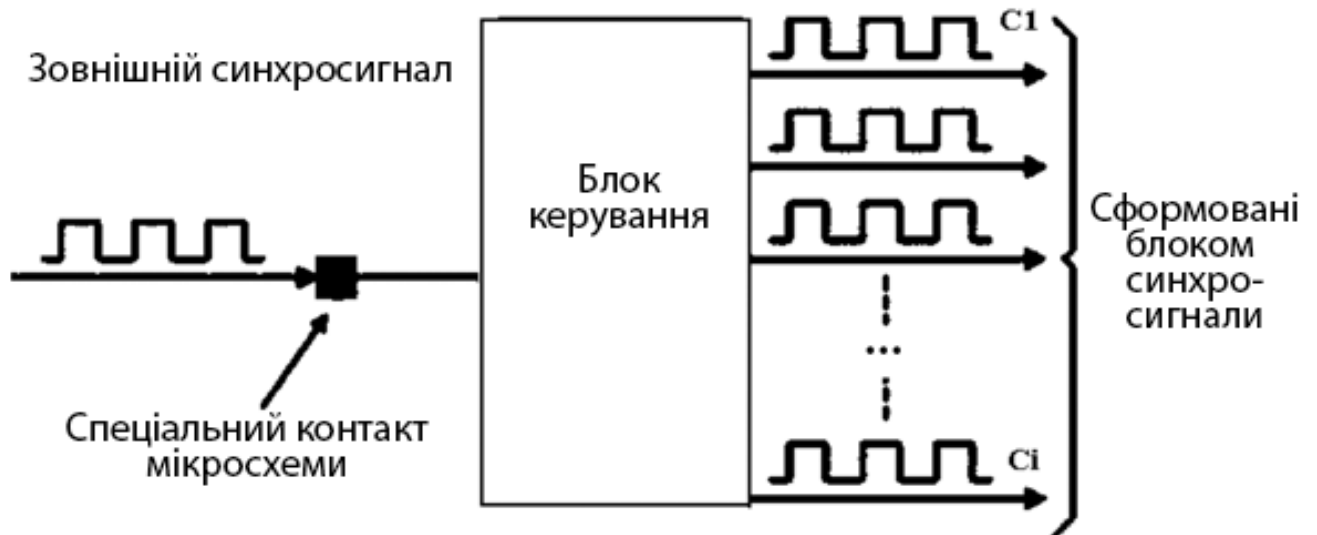


Рисунок 1.8 — Цифровий блок керування синхросигналами (DCM, CMT)

Трасувальні ресурси ПЛІС. Відмінною особливістю ПЛІС типу FPGA є складна організація трасувальних ресурсів-ліній зв'язку, які здійснюють електричне з'єднання різних функціональних елементів ПЛІС один з одним. Існує 4 типи ліній зв'язку: «довгі лінії», «половинні (hex) лінії», «подвійні лінії» та «короткі лінії прямого зв'язку». Ця градація визначається кількістю логічних блоків (КЛБ), по яких зв'язок проходить. З'єднання ліній зв'язку здійснюються за допомогою програмованих ключів, що мають істотні затримки, порівняні із затримками ЛК. Програми САПР, здійснюють трасування зв'язків елементів мікросхеми ПЛІС, ведуть її з урахуванням мінімізації довжин і затримок зв'язків та забезпечення 100% автоматичного трасування з'єднань. У випадках, коли повнота використання логічних ресурсів ПЛІС типу FPGA перевищує 90%, завдання трасування істотно ускладнюється і, крім збільшення затримок зв'язків, проект часто вимагає ручного трасування. Узагальнена структура ПЛІС типу FPGA наведена на рисунку 1.9.

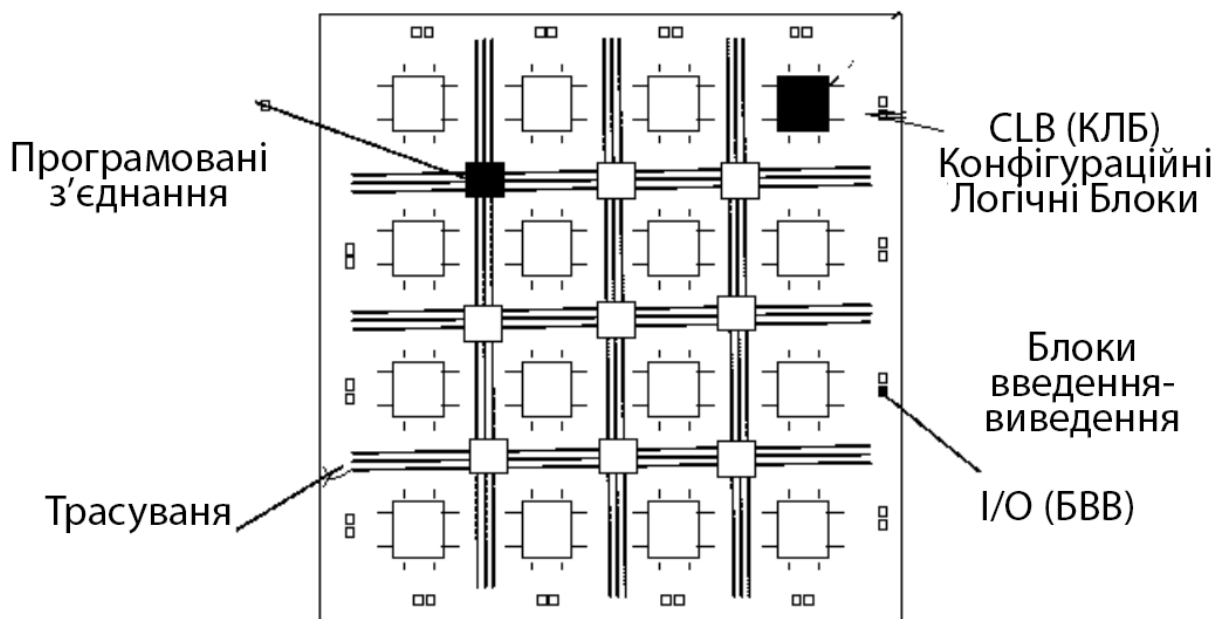


Рисунок 1.9 — Узагальнена структура ПЛІС типу FPGA

Внутрішні шини на елементах із трьома станами. Блоки введення-виведення всіх ПЛІС можуть програмуватися на організацію трьох стабільних виходів. Однак всередині ПЛІС буфера з трьома стабільними виходами та загальною шиною на їх базі застосовуються рідко через технологічні труднощі реалізації таких буферів, а також збільшеної затримки їх перемикавання. Тільки в дорогих серіях ПЛІС, наприклад VIRTEX-2, 4, 5, 6 фірми Xilinx передбачені можливості організації шин на їх основі. У звичайних шинах третій стан характеризується рівнем, що знаходиться між рівнями логічна 1 і логічний 0. У випадку, коли всі буфери передавачів, підключених до загальної шини будуть у третьому — високоімпедансному стані, сигнали перешкод мають на неї великий вплив. Зменшення їхнього впливу, загальна шина зазвичай «підтягнута», тобто навантажена спеціальною схемою, яка підтягує рівень шини або до слабкого рівня 1(H), або до рівня слабкий 0(L). Ця властивість шин у ПЛІС також програмується. При організації загальних шин на елементах із трьома станами апаратні витрати у кілька разів менші, ніж у шинах, реалізованих на мультиплексорах.

Вбудовані процесорні ядра. Вбудовані ядра ПЛІС типу FPGA можуть виступати як кості потужного мікропроцесора. Наприклад, для ПЛІС XILINX серії VIRTEX-5 — це 32-розрядний мікропроцесор PowerPC 440 з RISC-архітектурою,

сімикаскадним конвеєром команд, 32-кбайтною кеш пам'яттю команд і даних, 500 МГц тактовою частотою. У пізніших серіях (Серії 7000) — це мікропроцесор ARM 9. За допомогою таких процесорних ядер, блокової пам'яті та інших вбудованих ядер можна будувати на ПЛІС складні системи на одному кристалі (SOC-System On Chip).

Ядра шинних та мережевих інтерфейсів. У дорогих і високошвидкісних серіях ПЛІС (VIRTEX 5,6) до вбудованих ядер входять блоки типу PCI Express з підтримкою 8-канальних Gen1 (2,5 Гбіт/с) і 4-канальних Gen2 (5,0 Гбіт/с) зв'язків і мережевим Ethernet контролером ТЕМАС — 10/100/1000 Мбіт/с. У деякі серії ПЛІС (Спартан-6) до складу ядер включений контролер зовнішньої пам'яті, що програмується під стандарти DDR, DDR2, DDR3 і забезпечує швидкість передачі до 12,8 Гбіт/с.

Послідовні приймачі. Реалізація високошвидкісних паралельних шин ускладнюється взаємними перешкодами, розбіжністю довжин провідників тощо. Тому останнім часом спостерігається інтерес до послідовних приймачів. Відповідні ядра вбудовуються у сучасні ПЛІС. Серед них, наприклад, GTP — 3,125 Гбіт/с приймачі ПЛІС серії VIRTEX-5.

Аналого-цифрові ПЛІС. Для вирішення багатьох завдань необхідно мати у складі системи не тільки цифрові, а й аналогові блоки — підсилювачі, перетворювачі аналог-цифра (АЦП), цифро-аналог (ЦАП) та ін. Тому деякі фірми, наприклад, такі, як Lattice Semiconductor, особливу увагу приділяють розвитку аналого-цифрових ПЛІС [21].

Основні параметри, за якими відрізняються ПЛІС.

- тип — CPLD або FPGA;
- вартість та швидкодія — дешеві сімейства-COOLRUNNER 2, SPARTAN-3,6 (вартість мікросхеми від десятка доларів, тактові частоти 100-300 МГц; дорогі та швидкодіючі VIRTEX 4-6 (від сотень до тисяч) доларів, тактові частоти 200-700 МГц);

— логічні можливості мікросхем, що вимірюються або в кількості логічних осередків (макроячейки, LUT) — від сотень до сотень тисяч, або в кількості еквівалентних двовходових вентилів 2І: від десятків тисяч до кількох мільйонів.

— спеціалізація — підродина ПЛІС, орієнтовані на певний клас завдань — цифрове оброблення сигналів, вбудовані системи тощо.

— конструктивні та електричні параметри — для комерційного, автотранспортного, аерокосмічного, військового призначення тощо.

2 ПОРІВНЯЛЬНИЙ АНАЛІЗ І ОБГРУНТУВАННЯ ВИБОРУ АПАРАТНИХ ТА ПРОГРАМНИХ ЗАСОБІВ ПРОЕКТУВАННЯ

2.1 Огляд та порівняння основних світових виробників ПЛІС

Для вибору програмованої логічної інтегральної схеми при проектуванні ЦА та АЦ систем використовуються такі критерії, як:

- швидкодія;
- кількість логічних елементів;
- доступні порти;
- наявність документації та технічної підтримки;
- схемотехнічні та конструктивні параметри ПЛІС;
- вартість мікросхеми;
- вартість програмного забезпечення для роботи з платою.

На основі даних критеріїв, розглядаються найбільш поширені ПЛІС від різних виробників.

2.1.1 Altera

Компанія Altera Corporation заснована у 1983 році, нині поглинута компанією Intel. Розробляє різні сімейства ПЛІС, які можна використовувати для різних задач та потреб. До основних серій ПЛІС відносяться:

- MAX;
- Stratix;
- Arria;
- Cyclone;
- MAX 10;
- АРЕХ20К;

Серія Max — енергонезалежні ПЛІС, призначені для заміни дискретної логіки, перетворення інтерфейсів, розширення портів введення-виведення та іншого. Використовується при розробці мобільних пристроїв ПЛІС. На рисунку 2.1 зображений зовнішній вигляд ПЛІС Altera Max II.

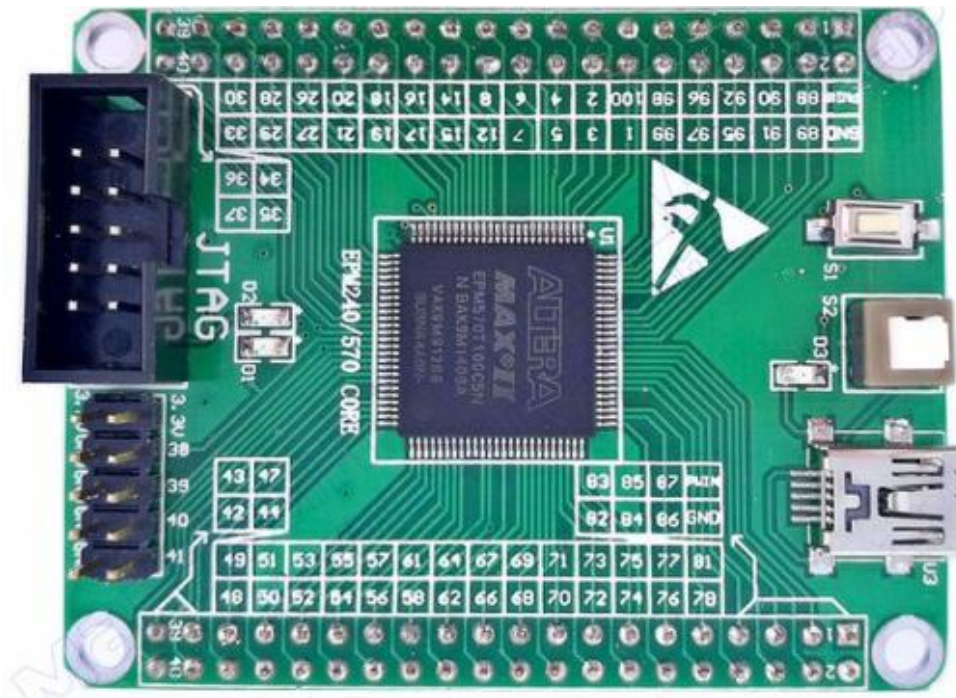


Рисунок 2.1 — Зовнішній вигляд ПЛІС Altera Max II

Серія Stratix — високопродуктивні ПЛІС великої логічної ємності (до 1 млн. еквівалентних логічних елементів). Випускаються за технологією статичного ОЗУ. Працюють на 40-нм архітектурі.

Серія Cyclone — не дорогі плати для вирішення великого діапазону задач. Призначені для застосування в масових проектах, де потрібні низька собівартість та низьке енергоспоживання. Випускаються за технологією статичного ОЗУ. У таблиці 2.1 наведені основні характеристики плат Altera DE (Development and Education) [22].

Таблиця 2.1 — Основні характеристики плат Altera DE

Х-тики	DE0	DE1	DE2-70	DE3
Назва пристрою	Cyclone III, EP3C16F484	Cyclone II, EP2C20F484	Cyclone II, EP2C70F896	Stratix III, EP3SL150/260/340
Кількість логічних елементів	15408	18752	68416	142000-338000
SDRAM	8 Мб	8 Мб	64 Мб	-
FLASH	4 Мб	4 Мб	8 Мб	-
DDR2	-	-	-	DDR 2 SO-DIMM модуль
SD карта	+	+	+	+

Продовження таблиці 2.1

Х-тики	DE0	DE1	DE2-70	DE3
VGA	4-біт	4-біт	10-біт UXGA	-
USB-Blaster	вбудований	вбудований	вбудований	вбудований
USB2.0	-	-	+	+
RS232	+	+	+	-
PS/2	+	+	+	-
Світлодіоди	10	18	27	8
Перемикачі	10	10	18	4
Кнопки	3	4	4	4

Серія Arria — ПЛІС середнього діапазону, оптимізовані для вирішення телекомунікаційних завдань. Всі мікросхеми цієї серії мають вбудовані високошвидкісні приймачі. Випускаються за технологією статичного ОЗУ.

Серія MAX 10 — недорогі плати з енергонезалежною конфігураційною Flash-пам'яттю для вирішення великого діапазону задач. Призначені для застосування у різних галузях.

На рисунку 2.2 наведено зовнішній вигляд плати Altera DE1.

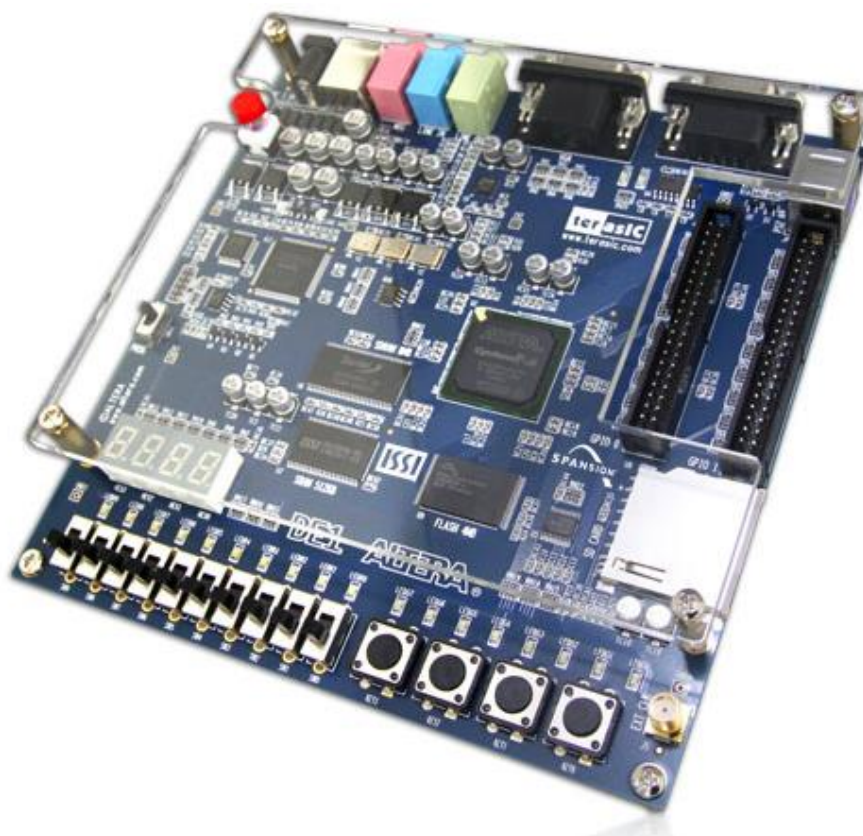


Рисунок 2.2 — Зовнішній вигляд плати Altera DE1

Серія АРЕХ 20К — ПЛІС надвисокого ступеня інтеграції. Поєднує в дві різні архітектури — матричну та вентиляну. В них є можливість реалізації вбудованої пам'яті в них є можливість реалізації вбудованої пам'яті. У таблиці 2.2 наведені основні характеристики плат сімейства АРЕХ 20К [13].

Таблиця 2.2 — основні характеристики плат сімейства АРЕХ 20К

Параметр (кількість)\Тип	EP20K60E	EP20K160E	EP20K300E	EP20K600E	EP20K1500E
Базові елементи цифрової схеми	162000	404000	728000	1573000	2524000
Логічні елементів	2560	6400	11520	24320	54720
К-сть комірок	256	640	1152	2432	3648
Системні блоки (ESB)	16	40	72	152	228
Об'єм пам'яті, біт	32768	81920	147456	311296	466944
Входи\виходи	204	316	408	624	858

З початку виходу серії плат Stratix III, у ПЛІС Altera з'являється технологія Programmable Power Technology. Вона дозволяє змінювати режим роботи, в наслідок чого можна регулювати потужність плати та її окремих елементів відповідно до потреб.

Altera самотужки розробляє програмне забезпечення для роботи з їх платами. Основною програмою для роботи з ПЛІС являється Quartus II. В даному ПЗ можна писати код на різних мовах опису, присутній графічний редактор для побудови схем, на основі яких можна згенерувати код. Для зручності процесу проектування є утиліта RTL Viewer, яку можна застосовувати для перегляду створеної схеми на основі коду.

У склад програмного забезпечення також входить ModelSim Simulator, програма за допомогою якої можна провести моделювання складних систем, наприклад кінцевих автоматів та мікропроцесорів.

Програмне забезпечення постачається безкоштовно разом з ПЛІС. Також доступні web версія ПЗ, вона розповсюджується безкоштовно, але з дещо урізаним функціоналом.

На більшості платах Altera є вбудований програматор USB-Blaster, який дозволяє перепрограмувати плату без використання додаткових програматорів, просто під'єднавши її до ПК та завантаживши скомпільований код на плату.

2.1.2 Xilinx

Компанія Xilinx заснована в 1984 році, займається виготовленням програмованих логічних інтегральних схем різного типу, являється одним з лідерів по їх продажу. Має велику кількість серій для виконання різноманітних як за складністю, так і за функціоналом. До основних серій плат ПЛІС від Xilinx відносяться:

- Spartan;
- Artix;
- Kintex;
- Virtex;
- Zynq;
- Versal;

На рисунку 2.3 зображено серії плат ПЛІС з поділом на покоління та функціональні можливості.

Серія Spartan — ПЛІС, виготовлені за технологією 45нм. Призначені для використання в бюджетних пристроях (наприклад побутова техніка чи бездротовий зв'язок).

Серія Kintex — сімейство плат середнього рівня, дана серія досягла максимально оптимального співвідношення ціна/продуктивність. Використовується для високопродуктивної логіки, обробки послідовних інтерфейсів та цифрової обробки сигналів.

Серія Versal — плати ПЛІС, які поєднують в собі матрицю програмованих логічних комірок, апаратне ядро RISC-процесора та апаратне ядро ARM. Перспективи розвитку даних плат дуже високі, виділився новий підтип ПЛІС PSoC —

програмування на кристалі, яке дозволяє усю систему реалізувати на основі одного кристала [17].

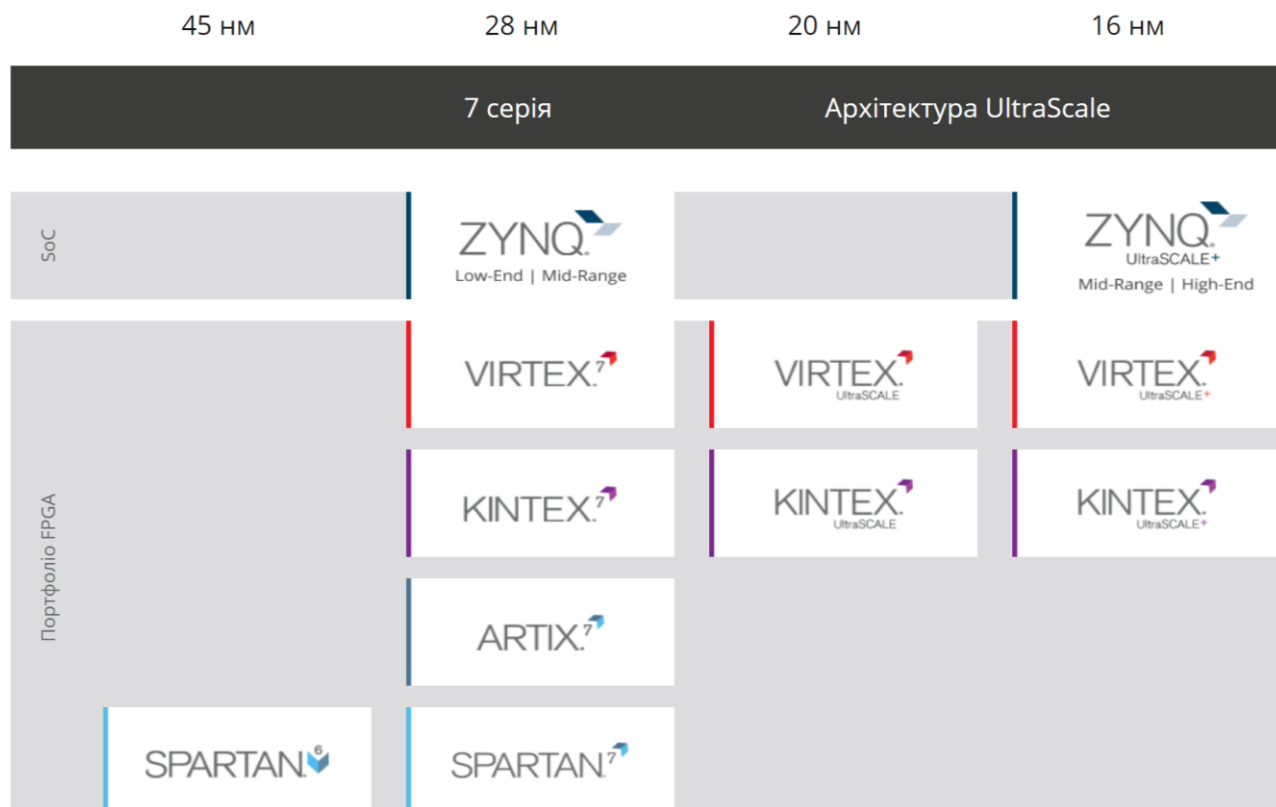


Рисунок 2.3 — Серії плат ПЛІС від 2.1.3 Xilinx

Серія Virtex — сімейство ПЛІС для високо продуктивних систем. Використання технологічного процесу 28 нм збільшити швидкодію у порівнянні з минулими поколіннями та зменшити енергоспоживання. У таблиці 2.3 наведені основні характеристики плат сімейства Virtex UltraScale.

Таблиця 2.3 — Основні характеристики плат сімейства Virtex UltraScale

Х-тики (кількість)	XCVU065	XCVU095	XCVU190	XCVU440
Логічні елементи	752	1129	2256	5319
Логічні комірки	627	941	1253	4433
Блоки RAM (Мб)	44,3	60,8	132,9	88,6
100 Гбіт Ethernet	3	4	9	3

Продовження таблиці 2.3

Х-тики (кількість)	XCVU065	XCVU095	XCVU190	XCVU440
Блоки PCIe	2	4	6	6
Входи/виходи	520	832	1040	1456

Серія Zynq — це плати з вбудованим апаратним ядром ARM. Завдяки ресурсам ПЛІС забезпечується максимальна продуктивність, а використання апаратного ядра ARM дозволяє розроблювати більш сучасні та спеціалізовані системи. На рисунку 2.4 зображена ПЛІС Zynq UltraScale+.

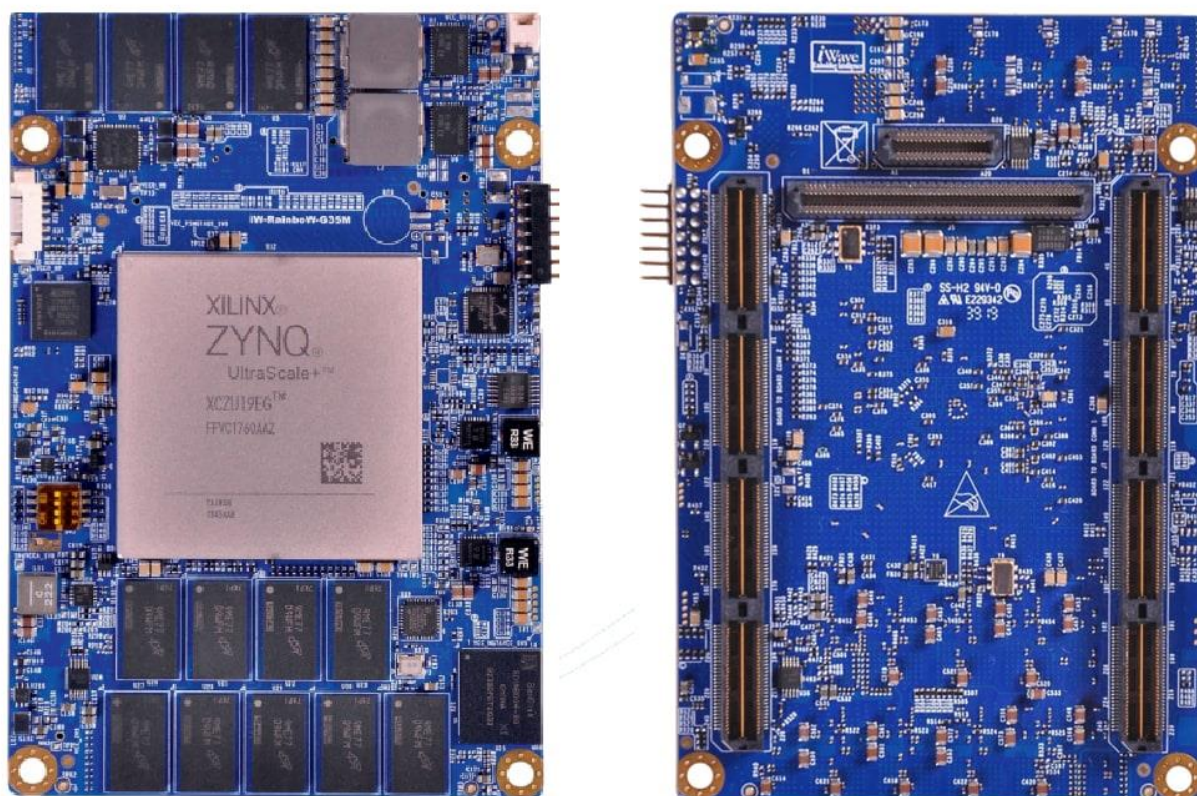


Рисунок 2.4 — Вигляд плати ПЛІС Zynq UltraScale+ від Xilinx

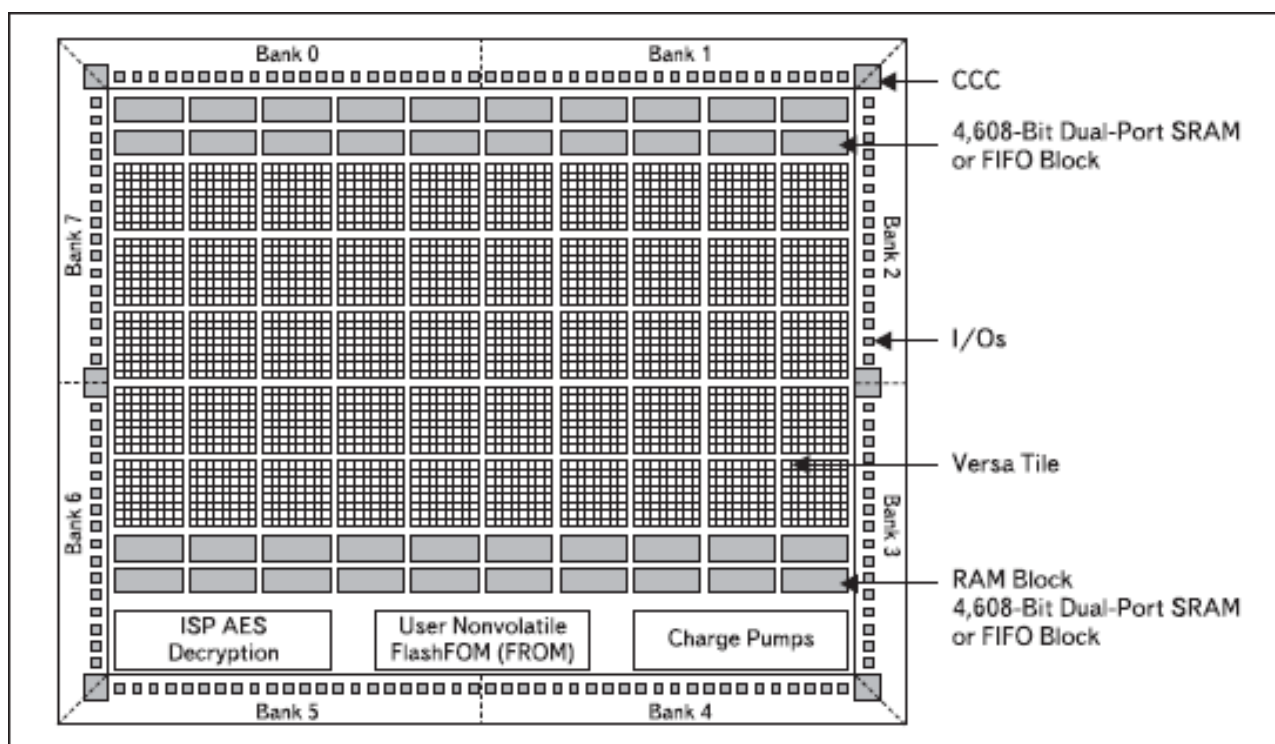
2.1.3 Actel Corporation (Microchip)

Компанія Actel заснована в 1985 році. Їх ПЛІС розраховані на підвищену надійність та використання у важких умовах, таких як підвищена радіація, пряме опромінення, висока чи низька температура та інші. Плати компанії зарекомендували свою надійність, та можуть використовуватись навіть в космосі. Так, плата

серії RH1280, використовувалась в системі обробки зображень відеокамери марсоходу, а також використовувалась для передачі зображень на Землю.

Такий високий рівень надійності ПЛІС зумовлений використанням технології Antifuse — це технологія, яка для створення логічних зв'язків використовує металеві перемички. Недоліком даної технології є те, що для плат потрібно використовувати додатковий програматор, який досить дорогий, проте високий ступінь захисту може компенсувати дані незручності. Причиною використання додаткового програматора є те, що в ПЛІС з технологією antifuse використовується одноразовою програмована пам'ять [18].

Інший напрямок компанії — виготовлення невеликих та недорогих плат ПЛІС, зі збереженням високого рівня надійності та з використанням flash-пам'яті. На рисунку 2.5 зображена архітектура плати ПЛІС від Actel.



Рисунк 2.5 — Архітектура ПЛІС Actel ProASIC3E

Основна сфера застосування плат за рахунок малих розмірів та надійності — це портативні пристрої та автомобілебудування. Також вони використовуються у військовій та аерокосмічних галузях.

До основних серій плат ПЛІС від Actel відносяться:

- PolarFire FPGA — це ПЛІС середнього діапазону, мають низьку споживану потужність при середньому діапазоні щільності з надвисоким рівнем надійності та безпеки;
- IGLOO FPGA — це плати дуже малого розміру (від 3 до 8 мм), мають низьку споживану потужність при високій щільності. Застосовуються зазвичай в портативних пристроях;
- ProASIC3 FPGA — плати з широко направленими функціоналом, пристрої підтримують IP-ядро soft-процесори ARM — Cortex-M11. Можуть використовуватись при високій чи низькій температурі;
- Fusion Mixed Signal FPGA — змішані сигнальні ПЛІС, що поєднують в собі великі блоки flash-пам'яті, комплексну схему генерації тактових сигналів і швидку програмовану логіку.

На рисунку 2.6 зображена плата ПЛІС від Actel/Microchip серії PolarFire

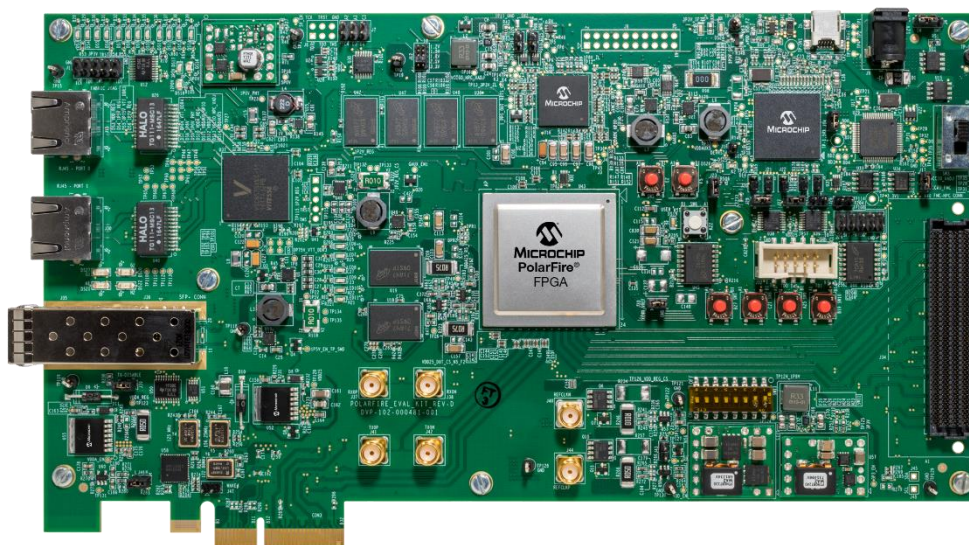


Рисунок 2.6 — ПЛІС MPF300 від Actel/Microchip серії PolarFire

2.2 Вибір середовища програмування

Для проектування програмованих логічних інтегральних схем потрібно використовувати програмне забезпечення. Зазвичай у кожного виробника власне ПЗ, в якому можна працювати тільки з їх платами. У всіх програм є комерційні версії, вартість ліцензії яких досить висока. Проте у більшості є безкоштовні вер-

сії програми, або ж ПЗ надається разом з покупкою ПЛІС. Далі будуть розглянуто програмне забезпечення для основних плат.

2.2.1 Libero

Libero — це програмне забезпечення компанії Actel(Microchip) для проектування пристроїв на основі ПЛІС. Actel не самі розробили ПЗ для ПЛІС, а використали існуючі ефективні модулі від сторонніх розробників. Програма складається зі засобу синтезу Synplify, а для створення схеми використовується графічний редактор VieDraw. Моделювання виконується в програмі Modelsim, а симуляція часових діаграм виконується з використанням WaveFormer Lite. За рахунок використання модулів програмного забезпечення від різних виробників, в Libero застосовані зручні та ефективні засоби проектування пристроїв на базі ПЛІС. Як і в інших виробників ПЛІС, є доступна безкоштовна версія програмного забезпечення під назвою Libero Silver.

Створення проекту здійснюється за допомогою опису генерації списку ланцюгів або генерується на основі введеної схеми. Для проектування схеми в ПЗ Libero є графічний редактор ViewDraw, у якому є редактор схем та редактор елементів, у якому можна створювати власні елементи. Після введення усієї схеми, генерується список у форматі EDIF, проводиться компіляція коду, а потім прошивається плата ПЛІС.

За допомогою функціонального моделювання можна перевірити чи правильно працює схема. Для цього використовується програма ModelSim, яка також застосовує WaveFormer Lite для створення часових діаграм. Прошивка ПЛІС проходить у модулі ПЗ Designer. Після цього проводиться моделювання роботи схеми у програмі. На рисунку 2.8 показане вікно моделювання програми.

Процес програмування плати здійснюється з використанням програматора та ПЗ Actel Designer [18].

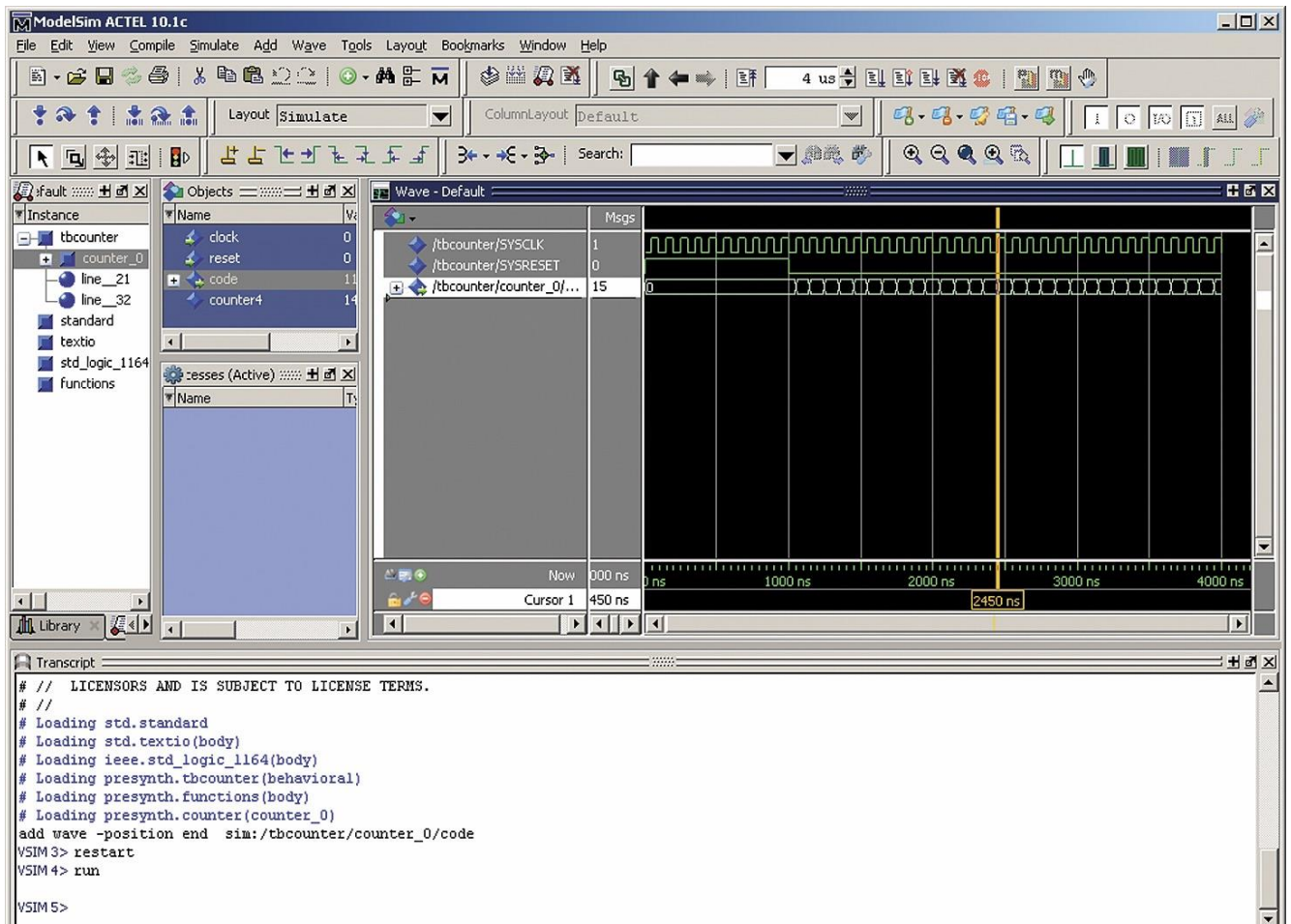


Рисунок 2.8 — Вікно програми для моделювання ModelSim

2.2.2 Програмне забезпечення Quartus II

Для проектування програмно-апаратних систем на ПЛІС Altera використовується програмне забезпечення Quartus. Безкоштовна версія цього ПЗ є доступною на офіційному сайті та надається разом з покупкою ПЛІС Altera.

В Quartus доступними є майже усі плати компанії та декілька від сторонніх виробників. В даній програмі найбільша кількість доступних ПЛІС у порівнянні з іншими. Інтерфейс програми наведений на рисунку 2.9.

Для проектування пристроїв у програмі доступними є графічний та програмний методи розробки.

Суть графічного методу — це створення схеми у графічному редакторі, який є у Quartus та ні в якому більше, тому що схеми зберігаються у форматі .bdf, який підтримує тільки він.

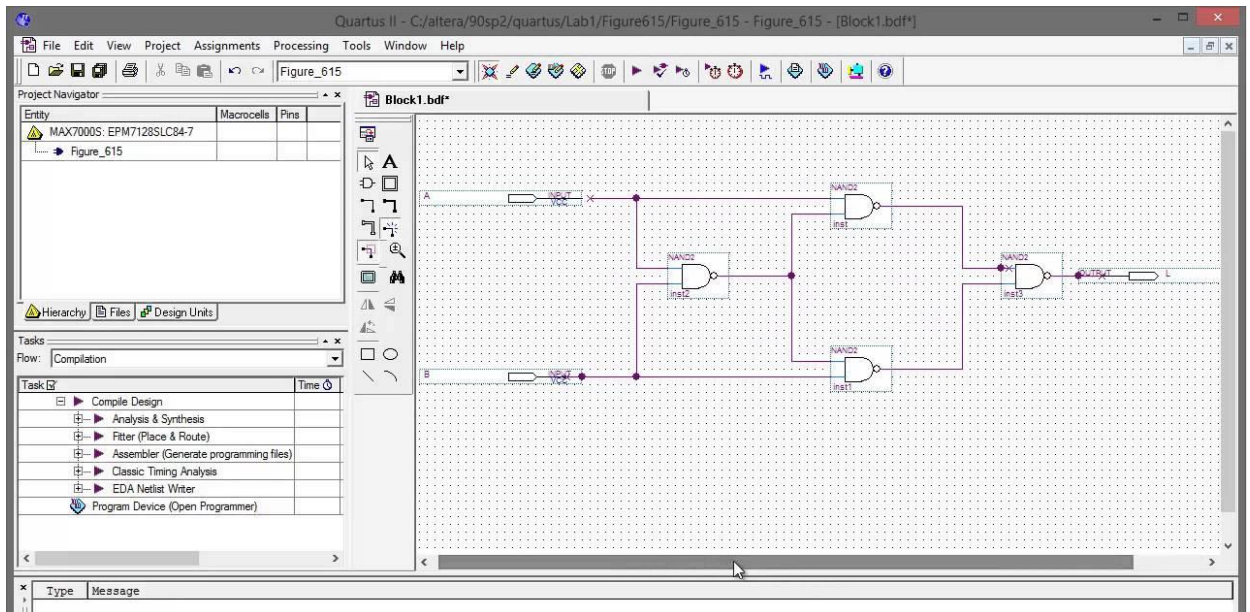


Рисунок 2.9 — Інтерфейс програмного забезпечення Quartus

Після створення схеми у графічному редакторі, на його основі створюється код, який компілюється аналогічно звичайній програмі.

Програмний метод полягає в написанні коду програми однією з мов опису, а саме: VHDL, Verilog чи AHDL. Код з програми компілюється аналогічно до схеми, тому методи являються взаємо заміними, якщо розробник не знає одного з методів, проте може використовувати інший.

При проектуванні програмним методом, є можливість згенерувати схему в утиліті RTL Viewer. Це дає більшу наочність процесу розробки й дозволяє попередньо перевірити роботу. На рисунку 2.10 показана схема, згенерована на основі коду.

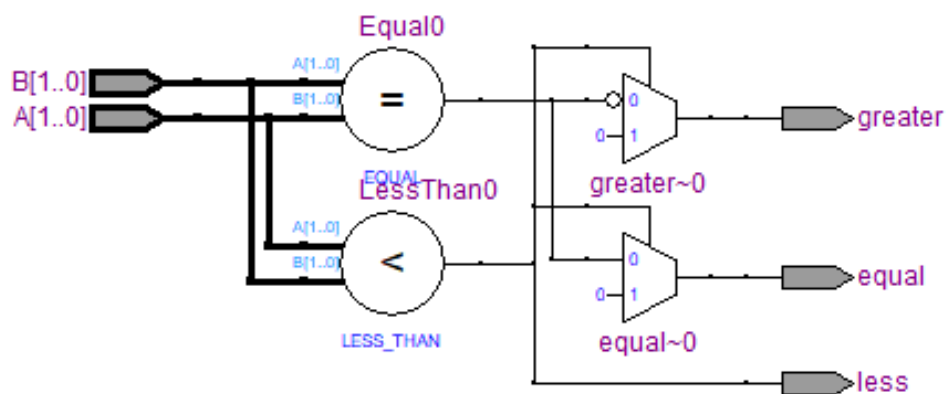
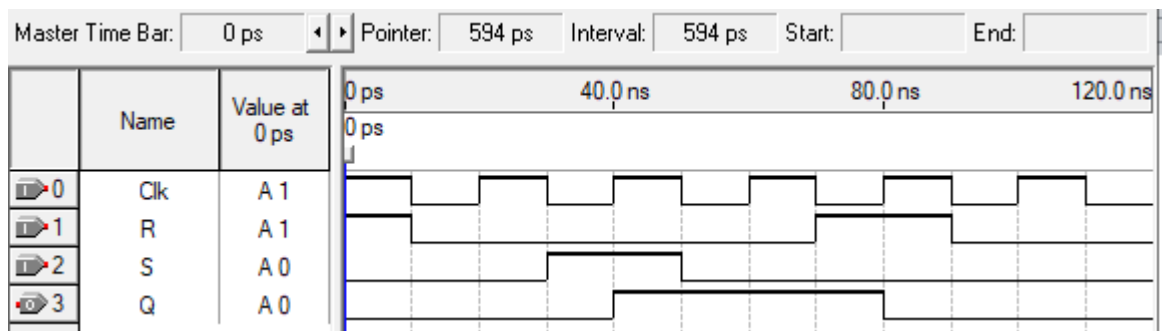


Рисунок 2.10 — Схема, згенерована в утиліті RTL Viewer

У Quartus є можливість моделювати роботу схеми за допомогою часових діаграм в утиліті VectorWaveform. Для цього достатньо розробити пристрій обраним методом, в утиліті Pin Planer налаштувати відповідність пінів плати та контактів, заданих при проектуванні та у VectorWaveform файлі налаштувати вхідні значення. На рисунку 2.11 зображена часова діаграма, створена за допомогою VectorWaveform file.

Для більш складних симуляцій можна застосовувати програму ModelSim, в якій доступна більша кількість параметрів для налаштування.



Рисунк 2.11 — Моделювання сигналів RS-тригера

Альтернативною для ПЛІС Altera — являється програмне забезпечення MAX+Plus II. Дане програмне середовище безкоштовне, проте більше не підтримується розробниками. Його функціонал обмежений, проте для реалізації простих пристроїв її також можна використовувати. Інтерфейс програмного забезпечення наведено на рисунку 2.12. До складу ПЗ MAX+Plus входять зв'язані між собою додатки, в яких проходять певні етапи проектування систем на ПЛІС Altera, а саме:

- `graphic editor` — це графічний редактор, який використовується для створення схем з елементів, які доступні в базовій бібліотеці, або створені розробником самотужки;

- `waveform editor` — програма для редагування часових діаграм, за допомогою неї можна провести моделювання роботи схеми з певними налаштованими вхідними сигналами;

- text editor — текстовий редактор, який використовується для написання коду програми мовами опису апаратури або для редагування існуючого коду
- symbol editor — графічний редактор, який дозволяє створювати власні елементи або редагувати існуючі. Будь-який скомпільований проект можна перетворити в елемент, який можна зберегти та використовувати при проектуванні нових пристроїв;
- floorplan editor — являється аналогом Pin Planer з Quartus. Використовується для зв'язування контактів ПЛІС та контактів, заданих у кодї чи схемі.

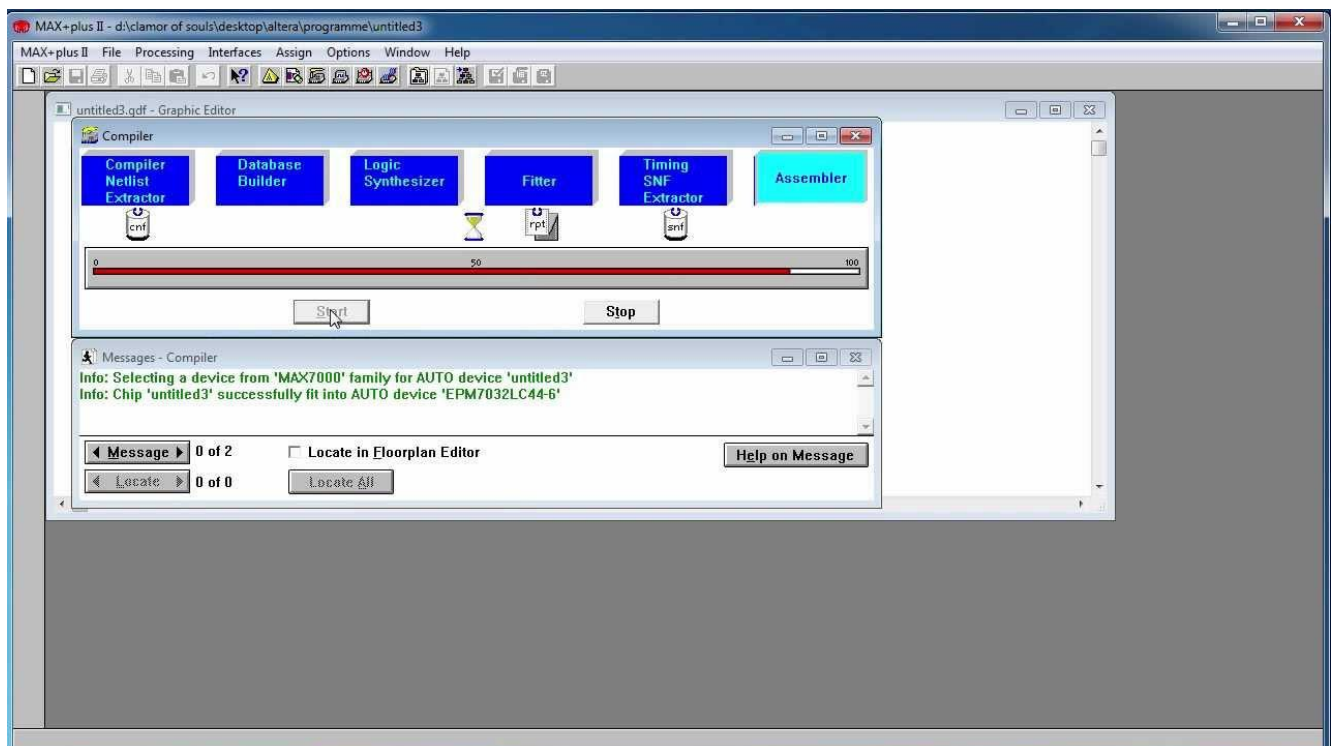


Рисунок 2.12 — Інтерфейс ПЗ MAX+Plus II

Проаналізуючи інформацію про різні програмні забезпечення для роботи з ПЛІС, можна зазначити, що Quartus переважає інші програми і по функціоналу, і за кількістю підтримуваних ПЛІС. Програма дозволяє використовувати різні методи проектування, що є зручним, також підтримує усі базові мови опису.

Програмне забезпечення Quartus та його компоненти являються ефективними та зручними інструментами при проектуванні пристроїв на базі ПЛІС, особливо при використанні плат Altera.

3 ПРОЕКТУВАННЯ ПРОЦЕСОРА АНАЛОГОВО-ЦИФРОВИХ ТА ЦИФРО-АНАЛОГОВИХ СИСТЕМ НА БАЗІ ПЛІС

3.1 Порівняння методів проектування систем на ПЛІС

Існує два основних методи проектування цифро-аналогових та аналого-цифрових систем на базі ПЛІС — це графічне введення схеми та текстовий опис проекту на мовах опису апаратури Verilog, VHDL або AHDL.

Незалежно від обраного методу, розробити хороший чи поганий проект можна як графічним, так і програмним методом. Тобто якість спроектованого пристрою в першу чергу залежить від розробника, а не вибраного методу. Проте для конкретного розробника може бути суттєво зручніший один чи інший метод.

Незалежно від того, чи графічний чи програмний спосіб представлення обраний, компілятор повинен скомпілювати однаково оптимальний файл прошивки. В будь-якому випадку компілятор намагатиметься оптимізувати файл прошивки, наприклад видалити не використані елементи.

Основними критеріями для порівняння методів проектування ЦА та АЦ систем є [14]:

- зручність користування;
- швидкість проектування;
- наочність представлення;
- надійність.

Зручність користування. Зручність створення схем визначається властивостями графічного редактора. У програмному забезпеченні Quartus для використання графічного методу проектування можна використовувати лише вбудований графічний редактор, без альтернатив. Quartus зберігає схеми у файл спеціального типу, .bdf, який можна відкрити тільки в цьому ж ПЗ. Сам графічний редактор не достатньо зручний у порівнянні з іншими, зокрема у порівнянні з САД системами. В інших редакторах великий обсяг розробки схеми проводиться з використанням командного рядка, який відсутній у графічному редакторі Quartus. Відсутність командного рядка збільшує час розробки, так як багато процесів з використанням

консолі суттєво швидші. На рисунку 3.1 показаний процес вставлення елемента в робочу область редактора.

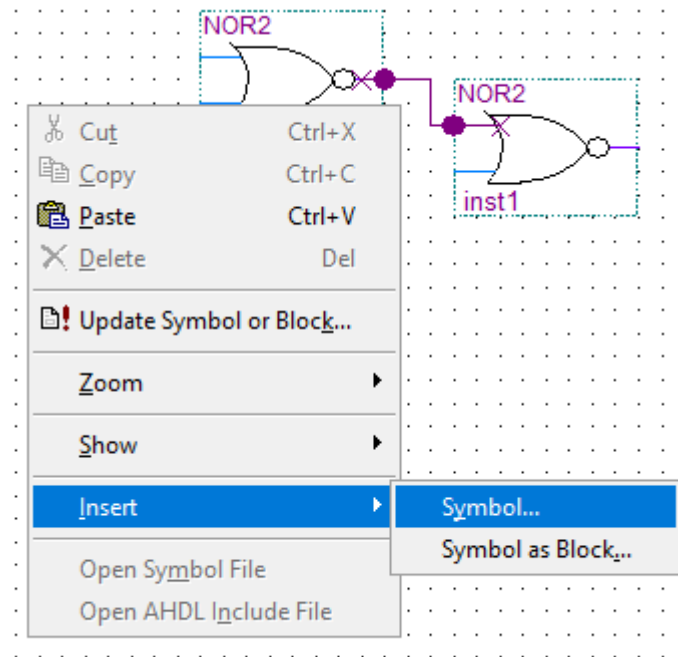


Рисунок 3.1 — Вставлення елемента в робочу область графічного редактора

В той же час використання мов опису являється більш універсальний у порівнянні з графічним методом. Для написання коду навіть не обов'язково мати програмне забезпечення конкретної плати ПЛІС. Достатньо використовувати Блокнот, змінивши тип файлу (наприклад при використанні мови опису VHDL тип файлу — .v). Для зручності написання все-таки зазвичай використовується ПЗ плати, так як воно може підказувати синтаксис, виділяти певні блоки іншим кольором та можна в будь-який момент скомпілювати код та побачити наявні помилки.

Ще однією перевагою в плані зручності при використанні методу програмування являється мінімальна залежність від середовища проектування, так як синтаксис мов опису не змінюється при зміні платформи ПЛІС. Графічний метод в цьому суттєво програє, так як графічні редактори Quartus II та Xilinx ISE суттєво відрізняються, в наслідок чого знадобиться час для освоювання нового графічного редактора.

Швидкість розробки. В першу чергу вона залежить від зручності методу для конкретного розробника. А також різниця між методами може бути не такою великою в порівнянні зі затраченим часом на попереднє планування розробки.

Більш раціонально використовувати графічний метод при проектуванні малих схем чи модулів. При створенні великої схеми її стає проблемно редагувати, особливо проблема відчувається при доопрацюванні схеми, так як доводиться пересувати елементи, шини, провідники та інше.

Також швидкість розробки залежить від особливостей мови програмування та програмного забезпечення. Усе, що можна спроектувати на схемі у графічному редакторі, може бути описане мовою VHDL, проте не усе, що написано мовою опису апаратури, може бути оптимально реалізовано у схемі.

Створення програми мовою опису дозволяє створювати параметризовані модулі, які можна використовувати при розробці інших проектів з іншими заданими параметрами. Такі модулі дозволяють створити базові елементи, наприклад тригери, суматори, лічильники, які можна буде використовувати в наступних проектах. В новому проекті достатньо підключити модуль та задати основні параметри елемента і його вже можна використовувати. А створювати графічні модулі у графічному редакторі Quartus неможливо, хоч і це подібні процеси.

Ще однією перевагою методу програмування являється зручність створення послідовно з'єднаних однакових елементів. У коді достатньо використати цикл для їх створення, а в графічному методі потрібно кожний елемент додавати та з'єднувати з іншими.

Наочність представлення. Однією з основних переваг графічного методу — являється наочність представлення пристрою. При цьому методі немає потреби попередньо створювати схему в інших графічних редакторах та можна чітко бачити на схемі входи, виходи та основні елементи за умови, що схема розроблена якісно.

Для деяких розробників не є проблемою відсутність графічного представлення пристрою в процесі розробки, проте для новачків та тих, хто не знає синтаксис мов опису на достатньому рівні — графічний метод буде більш наочним.

В той же час для розробників, які погано знаються в читанні схем — програмний засіб буде пріоритетним за рахунок відсутності лишньої інформації. При створенні схеми на ній будуть зображені елементи з усіма входами та виходами, незалежно від того, чи задіяні вони. При написанні програми, в коді задаються лише ті входи та виходи, які будуть використовуватись. На рисунку 3.2 наведено зображення JK тригера з усіма контактами, незалежно від використовуваних у програмі.

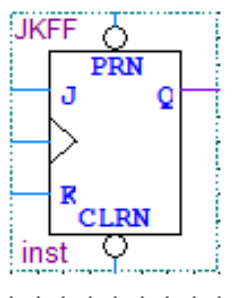


Рисунок 3.2 — JK-тригер у графічному редакторі Quartus

Перевагою в наочності представлення коду є можливість написання коментарів у будь-якій частині коду. Для цього використовується послідовність символів --, після яких текст вважається закоментованим. На схемі писати коментарі зазвичай незручно і вони пишуться окремим файлом.

Зручні та правильні назви для елементів, сигналів, входів та виходів роблять код зрозумілим та читабельним. В схемі теж можна давати назви елементам та контактам, проте для цього витрачається багато часу, так як елементи у графічному редакторі типізовані та він дає елементам назви за замовчуванням, які для наочності схеми потрібно перейменовувати.

На основі порівняння можна зробити висновок, що метод написання коду мовами опису зручніший та більш функціональний у порівнянні з графічним методом. Проте розробники, які не знають мови опису на достатньому рівні або для них зручніше створювати схему ніж писати код, можуть використовувати графічний метод без суттєвих для себе втрат.

3.2 Метод побудови процесорів на базі ПЛІС

Однією з основних переваг програмованих логічних інтегральних схем у порівнянні з програмною реалізацією являється швидкість виконання. Програмна реалізація мікропроцесора суттєво поступається у порівнянні з розробкою на основі ПЛІС. При використанні програмованих логічних інтегральних схем її можна налаштувати під задачі розроблюваного пристрою. Якщо ж буде потреба змінити задачі мікросхеми — це можна зробити в процесі роботи. Дана можливість дозволяє зекономити кошти за рахунок відсутності покупок нових елементів.

Зазвичай для проектування використовуються мікропроцесори чи процесори зі жорсткою логікою, проте використання софт-процесорів дає унікальні можливості при розробці складних систем. Софт(soft)-процесор — це мікропроцесорне ядро, створюється з використанням тільки логічного синтезу. Такі процесори краще конфігуруються, можуть розроблюватись під конкретні задачі або системи з акцентом на продуктивність, споживану потужність, масштабованість та інше. Може працювати на низькій частоті, якщо є така потреба.

На відміну від стандартних мікроконтролерів та мікросхем, логіка роботи ПЛІС не визначається при створенні та виробництві, а при проектуванні програмно-апаратних систем за допомогою мов опису. Для компіляції, відлагоджування та прошивки використовується програмне забезпечення та вбудований в плату програматор, які задають структуру пристрою. Це і дозволяє створювати софт-процесори на базі програмованих логічних інтегральних схем. Декі виробники ПЛІС випускають додаткові апаратні та програмні модулі, які дозволяють використовувати їх під конкретну задачу [15].

Виробники процесорів, зокрема Intel (яка являється власником Altera), почали випускати нові процесори з вбудованими у них ПЛІС. Це суттєво розширяє можливості розробки нових систем з можливістю використання апаратних та програмних ядер паралельно. Зображення процесора Intel з вбудованою ПЛІС наведено на рисунку 3.3.



Рисунок 3.3 — Процесор Intel Xeon з інтегрованою ПЛІС

Розвиток елементної бази плат ПЛІС привів до розвитку їх архітектури, яка може бути представлена у вигляді матриці програмованих логічних комірок. Одним з перших кроків до прогресу в розробці нових типів ПЛІС стало додавання апаратних ядер, наприклад компанія Xilinx додала в різні сімейства плат ядра PowerPC та ARM. Таке рішення сприяло виділенню нового типу ПЛІС — PSOC (Programmable System-On-Chip), програмована система на кристалі, що дозволяє вмістити всі складові пристрої на одному чіпові.

Програмовані логічні інтегральні схеми, які не мають апаратного ядра, можуть використовувати софт-процесор, який створюється на основі ресурсів ПЛІС. Такі процесори зазвичай виготовляються виробниками самих плат, наприклад Xilinx розробляє плати з 32-розрядним ядром MicroBlaze, який може виконувати аналогічні функції, що і апаратний процесор ARM. Також доступне 8-розрядне ядро PicoBlaze, яке обмежене по функціональності та використовується для проектування простих систем.

За рахунок не дуже важкого проектування софт-процесорів, можна розроблювати їх з акцентом на конкретні задачі.

Одна з аносованих новинок — ПЛІС Xilinx Versal, поєднує у собі компоненти трьох різних підходів до розробки пристроїв, а саме:

- матрицю програмованих логічних комірок;
- апаратне ядро ARM;
- апаратний RISC-процесор.

Таке поєднання дозволяє проектувати пристрої на оновленій архітектурі, для якої можна буде обрати потрібний апаратний підхід або інтегрувати декілька для паралельного використання. На рисунку 3.4 зображена ядро ПЛІС Xilinx Versal [16].



Рисунок 3.4 — Ядро Xilinx Versal

Поєднання програмованих компонентів ПЛІС та апаратних ядер дає розширені можливості для оптимізації процесу проектування ЦА та АЦ систем за рахунок зменшення кількості етапів розробки апаратного забезпечення за рахунок перенесення деяких операцій на логічні компоненти плати. Деякі операції, наприклад редагування алгоритмів нейромережі або система прийняття рішень, потребують регулярного редагування для оптимізації їх роботи. Для таких задач добре підходить використання ПЛІС за рахунок можливості швидко перепрограмування.

3.3 Реалізація процесора на базі ПЛІС

В якості практичного проектування вирішено створити простий процесор на базі ПЛІС.

На рисунку 3.5 наведена схема пристрою, в якій ряд 16-розрядних регістрів, суматор/віднімач, мультиплексор і блок керування (кінцевий автомат). Інформація вводиться систему через 16-розрядний вхід DIN , що завантажується в регістр IR . через 16-розрядний мультиплексор дані можуть передаватися з одного регістра в системі в інший, зокрема, з регістра IR в один із регістрів $r0$ - $r7$. Вихід мультиплексора на рисунку реалізований у вигляді шини, тому що вони передають дані у системі з одного елемента до іншого. Кінцевий автомат керує лініями $Select$ мультиплексора, це дає можливість передати дані з будь-якого входу КА у будь-який регістр, який підключений до шини.

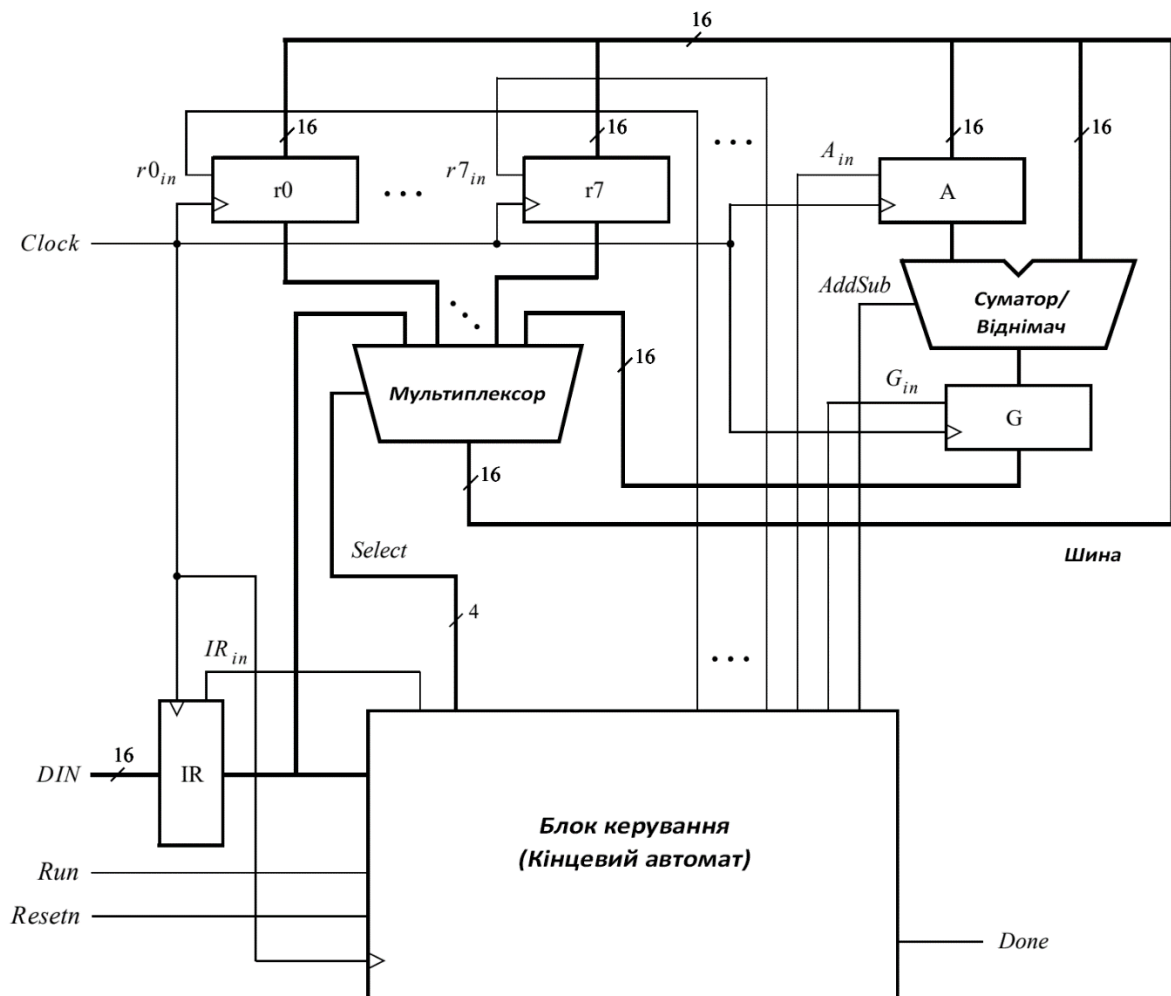


Рисунок 3.5 — Схема пристрою

Дана система здатна виконувати різні операції в кожному такті, за це відповідає Кінцевий автомат. КА визначає, коли деякі дані переміщується шиною та контролює, до якого реєстрів мають надійти ці дані. Наприклад, якщо КА вибере $r1$ в якості виходу для шини мультиплектора і також підтвердить Ain для реєстра A , тоді вміст реєстра $r1$ на наступному активному тактовому сигналові буде завантажено в реєстр A .

Додавання або віднімання чисел виконується з використанням мультиплектора. Мультиплексор використовується спочатку для того, щоб помістити одне 16-розрядне число на шину, а після цього завантажити його в реєстр A . Далі на шину потрапляє друге 16-бітове число, суматор/віднімач виконує задану операцію, а результат зберігається в реєстрі G . Дані в реєстрі G можна далі передати за допомогою мультиплектора до іншого реєстра за потреби.

Подібні системи часто називаються процесором. Він виконує операції, за заданими інструкціями. У таблиці 3.1 наведено інструкції, за якими працює процесор. Стовець зліва показує назву інструкції та операнди для цієї дії. Суть синтаксису $rX \leftarrow Op2$ полягає в тому, що операнд два, $Op2$, завантажуються у реєстр rX . Операнд $Op2$ — це або реєстр (rY), або дані ($\#D$).

Таблиця 3.1 — Інструкції, які виконує процесор

Інструкція	Виконується функція
$mv\ rX, Op2$	$rX \leftarrow Op2$
$mvt\ rX, \#D$	$rX_{15-8} \leftarrow D_{15-8}$
$add\ rX, Op2$	$rX \leftarrow rX + Op2$
$sub\ rX, Op2$	$rX \leftarrow rX - Op2$

Інструкції завантажуються із входу DIN та зберігаються в реєстрі IR . Усі інструкції кодуються у 16-бітовому форматі. Якщо $Op2$ — це реєстр, то кодування інструкції — $III0XXX000000YYY$, де XXX дає реєстр rX , YYY — це реєстр rY , а III вказує інструкцію. Якщо операнд $Op2$ визначає дані $\#D$, то кодуванням — $III1XXXDDDDDDDDDD$, де 9-бітове поле $DDDDDDDDDD$ являє собою дані. Для

кодування чотирьох інструкцій достатньо використати два біти, використання трьох бітів при кодуванні використовується для того, щоб можна було додати інструкції в майбутньому без суттєвої зміни коду програми. Кодування доступних інструкцій розподілене по таких значеннях: для інструкції `mv` $Ш = 000$, `mvt` — 001 , `add` — 010 , `sub` — 011 .

Інструкція `mv` (переміщення) копіює дані одного регістра в інший, для цього використовується синтаксис `mv rX, rY`. Цю інструкцію також можна застосувати для ініціалізації регістра з даними, у вигляді `mv rX, #D`. Так як дані `D` представлені всередині зашифрованої інструкції, використовуючи лише дев'ять бітів, процесор повинен розширити дані, наприклад `0000000D8-0`, перед завантаженням їх у регістр `rX`. `mvt` — це інструкція переміщення вгору, використовується для знаходження найбільшого байта регістра. Для `mvt` використовується вісім бітів поля `D` в інструкції, тому `mvt rX, #D` завантажує значення `D15-800000000` в `rX`. Наприклад, для завантажити регістр `r0` зі значенням `0xF000`, треба використати інструкцію `mvt r0,#0xF000`. Далі інструкція `add rX, rY` створює суму `rX + rY`, яка зберігається у `rX`. Інструкція `add rX, #D` створює суму `rX + D`, дані `D` розширюються до 16 біт, після чого результат зберігається у `rX`.

Певні інструкції, зокрема додавання або віднімання, займають декілька тактових тактів на своє виконання, тому що по шині потрібно перенести декілька комірок даних. `КА` у процесорі підтверджує керуючі сигнали під час виконання таких інструкцій, які потрібні в послідовних тактових циклах, поки інструкція не буде завершена повністю. Процесор виконує інструкцію починаючи з входу `DIN`, коли сигнал підтверджується `Run`, а коли команда закінчується керуючим сигналом, процесор відсилає підтвердження на вихід `Done`. Після того, як усі інструкції, які підтвержені на часовому кроці `T0`, завершено, сигнал відправляється на `IRin`. Для інструкції `mv`, коли вибрано регістр `IR`, мультиплексор виводить `0000000DDDDDDDDDD`, а для `mvt` — `DDDDDDDD00000000`. У таблиці 3.1 наведено інструкції, які повинні виконуватись для схеми, що зображена на рисунку 3.5, та які повинні отримувати підтвердження на кожному кроці часу; інші сигнали, не зазначені в таблиці підтвердження не вимагають. Сенс відсилання сигналу

на вхід AddSub на наступному кроці допоміжної інструкції ґрунтується на тому, що він встановлюється на 1. Дане налаштування заставляє модуль суматора/віднімача проводити операцію віднімання за допомогою арифметики з додаванням до 2.

Процесор на рисунку 3.5 може виконувати різні завдання, на основі послідовності інструкцій. Послідовність інструкцій, наведена на рисунку 3.6 завантажує число 22 в регістр r0, далі проводить обчислення в регістрі r1 та проводить операцію доповнення до 2-22.

```

1  mv  r0, #22
2  mvt r1, #0xFF00
3  add r1, #0x00FF
4  sub r1, r0
5  add r1, #1

```

Рисунок 3.6 — Послідовність інструкцій, виконувана процесором

Для реалізації процесора, що показаний на рисунку 3.5, обрано мову опису апаратури VHDL. Основні фрагменти коду наведені на рисунках 3.7 та 3.8.

```

1  ENTITY proc IS
2  PORT ( DIN : IN STD_LOGIC_VECTOR(15 DOWNTO 0);
3        Resetn, Clock, Run : IN STD_LOGIC;
4        Done : BUFFER STD_LOGIC);
5  END proc;
6  ARCHITECTURE Behavior OF proc IS
7
8  TYPE State_type IS (T0, T1, T2, T3);
9  SIGNAL Tstep_Q, Tstep_D: State_type;
10
11 CONSTANT mv : STD_LOGIC_VECTOR(2 DOWNTO 0) := "000";
12 CONSTANT mvt : STD_LOGIC_VECTOR(2 DOWNTO 0) := "001";
13 CONSTANT add : STD_LOGIC_VECTOR(2 DOWNTO 0) := "010";
14 CONSTANT sub : STD_LOGIC_VECTOR(2 DOWNTO 0) := "011";
15 CONSTANT Sel_R0 : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0000";
16
17 CONSTANT Sel_R7 : STD_LOGIC_VECTOR(3 DOWNTO 0) := "0111";
18 CONSTANT Sel_G : STD_LOGIC_VECTOR(3 DOWNTO 0) := "1000";
19 CONSTANT Sel_D : STD_LOGIC_VECTOR(3 DOWNTO 0) := "1001";
20 CONSTANT Sel_D8 : STD_LOGIC_VECTOR(3 DOWNTO 0) := "1010" ;
21
22 BEGIN
23 III <= IR(15 DOWNTO 13);
24 IMM <= IR(12);
25 rX <= IR(11 DOWNTO 9);
26 rY <= IR(2 DOWNTO 0);
27 decX: dec3to8 PORT MAP (rX, Xreg);
28 statetable: PROCESS(Tstep_Q, Run, Done)
29 BEGIN
30 CASE Tstep_Q IS
31 WHEN T0 =>
32 IF Run = '0' THEN Tstep_D <= T0;
33 ELSE Tstep_D <= T1; END IF;
34 WHEN T1 =>

```

Рисунок 3.7 — Оголошення констант та сигналів у програмі процесора

```

43 busmux: PROCESS (Sel, R0, R1, R2, R3, R4, R5, R6, R7, G, IR)
44 BEGIN
45 CASE Sel IS
46 WHEN Sel_R0 => BusWires <= R0;
47 WHEN Sel_R1 => BusWires <= R1;
48
49 WHEN Sel_R7 => BusWires <= R7;
50 WHEN Sel_G => BusWires <= G;
51 WHEN Sel_D => BusWires <= "0000000" & IR(8 DOWNT0 0);
52 WHEN Sel_D8 => BusWires <= IR(7 DOWNT0 0) & "00000000";
53 WHEN OTHERS => BusWires <= (OTHERS => '-');
54 END CASE;
55 END PROCESS;
56 END Behavior;
57 LIBRARY ieee;
58 USE ieee.std_logic_1164.all;
59 ENTITY dec3to8 IS
60 PORT ( W : IN STD_LOGIC_VECTOR(2 DOWNT0 0);
61       Y : OUT STD_LOGIC_VECTOR(0 TO 7));
62 END dec3to8;
63 ARCHITECTURE Behavior OF dec3to8 IS
64 BEGIN
65 PROCESS (W)
66 BEGIN
67 CASE W IS
68 WHEN "000" => Y <= "10000000";
69 WHEN "001" => Y <= "01000000";
70 WHEN "010" => Y <= "00100000";
71
72 WHEN "110" => Y <= "00000010";
73 WHEN "111" => Y <= "00000001";
74 WHEN OTHERS => Y <= "00000000";
75 END CASE;
76 END PROCESS;
77 END Behavior;

```

Рисунок 3.8 — Реалізація фрагменту коду процесора

Для моделювання роботи процесора, потрібно налаштувати файли, щоб VHDL-код можна було скомпілювати для використання у програмному забезпеченні ModelSim Simulator. Моделювання дозволить перевірити, чи правильно працює спроектований процесор.

Результати моделювання у програмному забезпеченні ModelSim для правильно спроектованої схеми, наведені на рисунку 3.9. Тут показано значення 0x101C, яке завантажується в IR з DIN під час 30 нс. Так це значення являється виконанням інструкції `mv r0,#28`, де значення $D = 28$ (0x1C), воно завантажується в регістр r0 на фронті тактової частоти на 50 нс. Далі результати моделювання показують послідовно виконання інструкцій `mvt r1, #0xFF00` за 70 нс, додавання `add r0, #0xFF` за 110 нс і віднімання `sub r1, r0` за 190 нс. Програмне забезпечення ModelSim дозволяє детально, в близьких до реальних умовах, проводити моделювання програмно-апаратних систем.

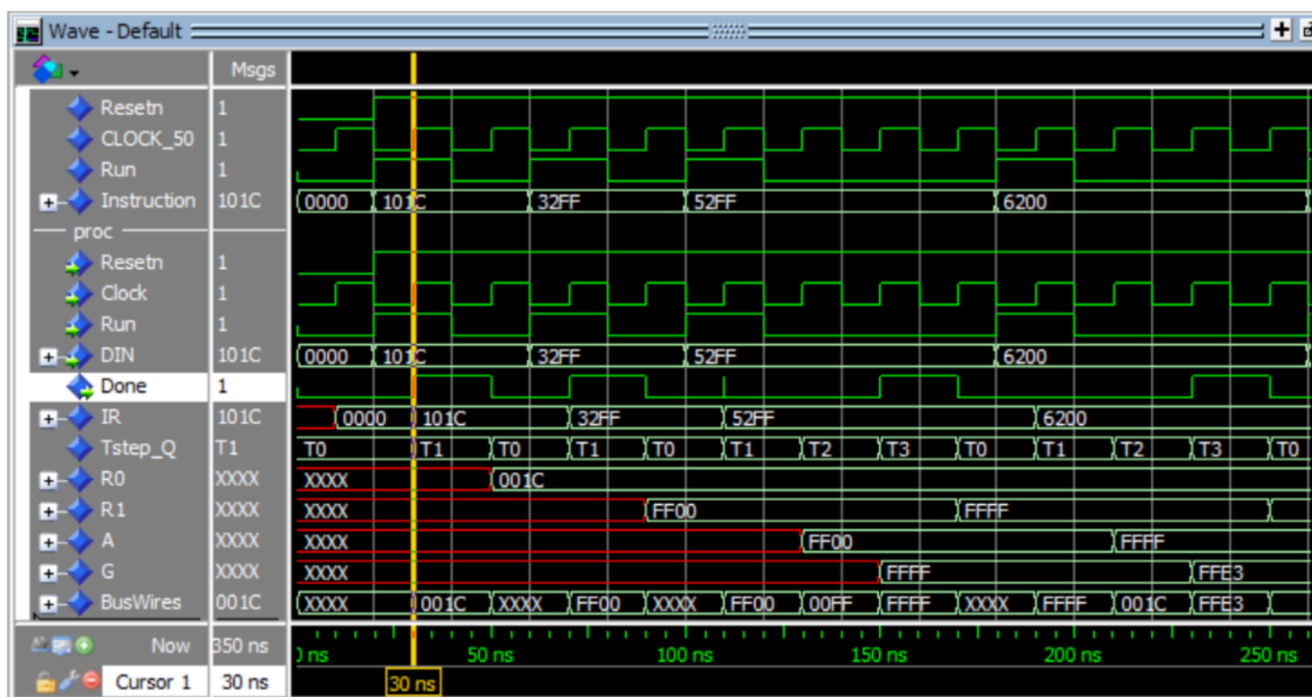


Рисунок 3.9 — Моделювання роботи програми у ПЗ ModelSim

4 РОЗРАХУНОК ЕКОНОМІЧНОЇ ДОЦІЛЬНОСТІ ПРОЕКТУВАННЯ ПРОЦЕСОРА НА ОСНОВІ ПРОГРАМОВАНОЇ ЛОГІЧНОЇ ІНТЕГРАЛЬНОЇ СХЕМИ

Розробка та проектування цифрових пристроїв вимагають великих фінансових вкладень. Їх потрібно максимально оптимізувати на етапах виробництва та реалізації виробів, тому що це процес покращення будь-якого виробництва. Базуючись на економічних розрахунках можна показати рентабельність та ефективність застосування спроектованих приладів у виробництво, що в свою чергу дасть комерціалізацію наукових досліджень. План та послідовність дослідження економічної частини наведена на рисунку 4.1.

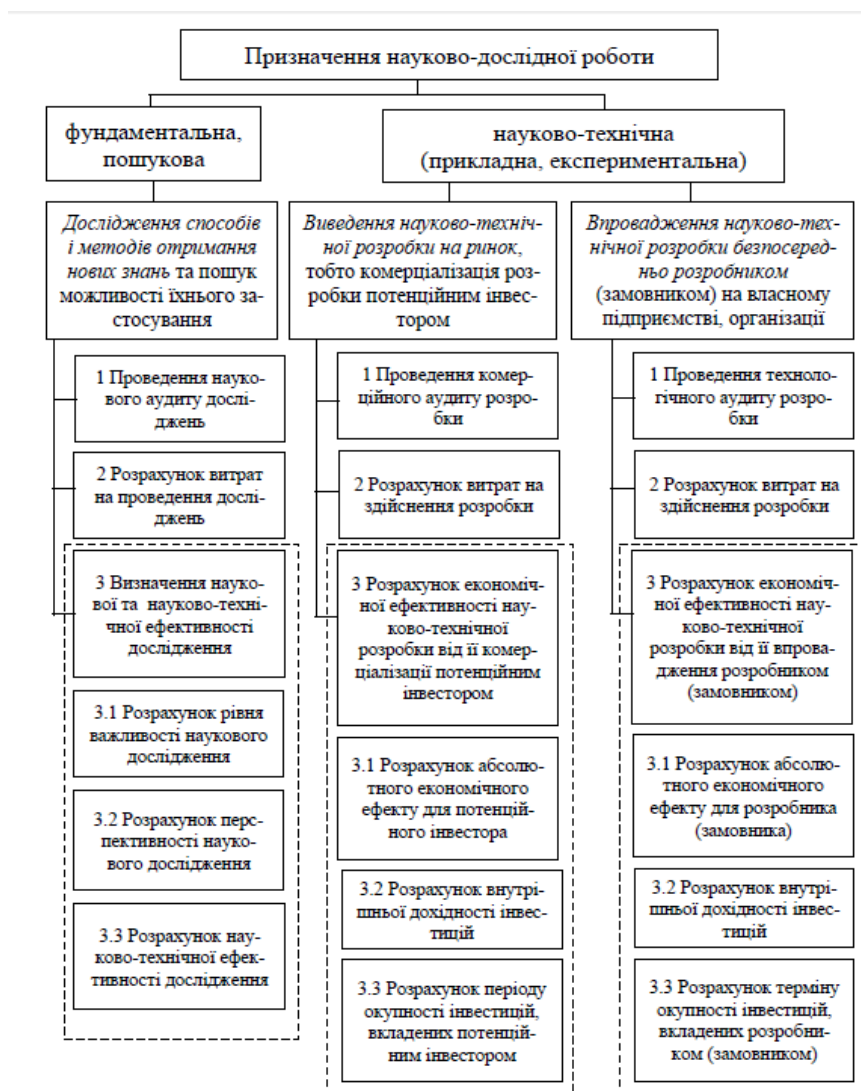


Рисунок 4.1 — План дослідження економічної частини магістерської роботи

Дана магістерська робота відноситься до прикладної науково-технічної роботи. Планується виведення науково-технічної розробки на ринок із залученням потенційним інвестора.

4.1 Проведення комерційного та технологічного аудиту науково-технічної розробки

Мате оцінки потенціалу комерційного розвитку — це оцінка комерційного розвитку, яка впливає з науково-технічних досліджень та процесу проектування. На основі результатів робляться висновки про напрямки організації та її майбутнього впровадження з врахуванням оцінки. Потенціал інвестицій оцінюється відповідно до дванадцяти критеріїв, які наведені у таблиці 4.1.

Таблиця 4.1 — Оцінювання комерційного потенціалу розробки

Критерії оцінювання та бали (за 5-бальною шкалою)					
Критерій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Багато аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою

Продовження таблиці 4.1

Критерій	0	1	2	3	4
Ринкові перспективи					
Практична здійсненність					
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
8	Відсутні фахівці з технічної та комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні.	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5 років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідно регламентні документи, велика кількість дозвільних документів виробництва продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Процедура отримання дозвільних документів для виробництва продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Відштовхуючись від таблиці 4.1 експерти, в даному випадку керівник магістерських робіт та викладачі кафедри ОТ, визначають різні результати. Результати оцінки комерційного потенціалу наведені у таблиці 4.2.

Таблиця 4.2 — Результати оцінювання комерційного потенціалу розробки

Критерії	Експерт (ПІБ, посада)		
	1 Тарновський М.Г., к.т.н., доц. кафедри ОТ	2 Крупельницьк ий Л.В., к.т.н., доц. кафедри ОТ	3 Богомолов С.В., к.т.н., доц. кафедри ОТ
	Бали:		
1. Технічна здійсненність концепції	3	3	4
2. Ринкові переваги (наявність аналогів)	3	3	4
3. Ринкові переваги (ціна продукту)	3	3	3
4. Ринкові переваги (технічні властивості)	2	3	3
5. Ринкові переваги (експлуатаційні витрати)	3	3	3
6. Ринкові перспективи (розмір ринку)	4	3	3
7. Ринкові перспективи (конкуренція)	3	3	3
8. Практична здійсненність (наявність фахівців)	2	3	3
9. Практична здійсненність (наявність фінансів)	2	4	2
10. Практична здійсненність (необхідність нових матеріалів)	1	4	2
11. Практична здійсненність (термін реалізації)	4	1	3
12. Практична здійсненність (розробка документів)	3	4	4
Сума балів	$СБ_1 = 33$	$СБ_1 = 37$	$СБ_1 = 37$
Середньо-арифметична сума балів $СБ_C$	$СБ_C = \frac{\sum_1^3 СБ_i}{3} = \frac{33 + 37 + 37}{3} = 35,6$		

Відштовхуючись від таблиці 4.2, а також відповідно до рівня комерційного потенціалу, який наведений у таблиці 4.3, можна зробити висновок про рівень потенціалу комерційного розвитку.

З врахуванням середнього арифметичного експертної оцінки, $СБ_C = 35,6$, можна зробити висновок, що рівень комерційного потенціалу цієї розробки буде вище середнього.

Таблиця 4.3 — Рівні комерційного потенціалу розробки

Середньоарифметична сума балів $\overline{СБ}$, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0 — 10	Низький
11 — 20	Нижче середнього
21 — 30	Середній
31 — 40	Вище середнього
41 — 48	Високий

Софт-процесор TSK51/52 використаний для порівняння з розроблюваним пристроєм. Даний процесор більше не виробляється. Він являється вузьконаправленими та суттєво обмеженим у функціоналі.

Нова розробка, навпаки, являється дешевою та дозволяє без суттєвих затрат змінити щось в апаратній чи програмній частині і тому має більшу швидкість.

Інший аналог розроблюваного пристрою — процесор Nios, який призначений для вкрай обмежених обчислень. Їх перевагою являється низька ціна.

Розроблюваний пристрій у порівнянні з аналогами дозволяє зберегти відносно невелику ціну при суттєво більшому функціоналу. Порівняння розробки з її аналогами наведено у таблиці 4.4.

Пристрій буде просуватися за допомогою реклами в соціальних мережах, пошукових системах та багатьох інших джерелах Інтернету. При використанні аналітики соціальних мереж та пошукових систем, можна використати таргетингову рекламу для її виведення зацікавленим подібною тематикою користувачам.

Таблиця 4.4 — Порівняння характеристик розробки з аналогами

Показники	Розроблюваний пристрій	TSK51/52	LM311 та LM339
Функціонал	9	6	4
Швидкодія	8	5	6
Надійність	8	8	6
Метод розповсюдження	9	5	8
Вартість	8	2	10

Пристрій може використовуватись в комп'ютерній електроніці, при навчанні технічних дисциплін, у комп'ютерних системах та мережах.

Новизна розробки полягає у використанні програмованих логічних інтегральних схем, які дозволяють перепрограмувати мікропроцесор та пам'ять не використовуючи додаткового програматора та не виймаючи мікропроцесор з плати. Проектування пристрої на основі ПЛІС дозволяє модифікувати існуючі схеми софт-процесора, оптимізуючи їх, що в свою чергу зменшить як вартість розробки пристрою, так і вартість кінцевого пристрою.

Виходячи з результатів порівняння пристрою з аналогами, можна з впевненістю сказати, що він є конкурентоспроможним, оскільки в більшості аспектах він перевершує існуючі аналоги на ринку. Такий високий рівень досягнутий за рахунок модифікації алгоритму зі схемою та розширення функціональних можливостей нової науково-технічної розробки порівняно з аналогічними розробками, які існують сьогодні на ринку.

4.2 Розрахунок витрат на здійснення науково-дослідної роботи

У магістерській роботі розглядається метод побудови софт-процесора на базі ПЛІС.

4.2.1 Витрати на оплату праці

Основна заробітна плата розробників, що працюють над проектом, визначена у формулі:

$$Z_o = \sum_{i=1}^k \frac{M_{ni} \cdot t_i}{T_p}, \quad (4.1)$$

де k — кількість посад дослідників, залучених до процесу досліджень;

M_{ni} — місячний посадовий оклад конкретного дослідника, грн;

T_p — середня кількість робочих днів в місяці, $T_p=21 \dots 23$ дні. Обрано 22 дні;

t_i — кількість днів роботи конкретного дослідника, дні;

Над створенням розробки працював менеджер проекту та інженер програмного забезпечення, тому ми виконаємо для них усі необхідні розрахунки, і вносимо їх до таблиці 4.5:

Таблиця 4.5 — Витрати на заробітну плату дослідників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	18500	840	15	12610
Інженер	10000	455	30	13630
Всього				26240

$$З_{о.к.} = \frac{18500 \cdot 15}{22} = 12610(\text{грн}).$$

$$З_{о.ін.} = \frac{10000 \cdot 30}{22} = 13630(\text{грн}).$$

Витрати на основну заробітну плату робітників за відповідними найменуваннями робіт відсутні, тобто $З_p = 0$.

Додаткова винагорода ($З_{дод.}$) усіх розробників та працівників, які брали участь у цьому етапі роботи, обчислюється як 10 ... 12% від суми основної заробітної плати дослідників та робітників за формулою:

$$З_{дод.} = (З_o + З_p) \cdot \frac{N_{дод.}}{100\%}, \quad (4.2)$$

де $N_{дод.}$ — норма нарахування додаткової заробітної плати.

$$З_{дод.к.} = \frac{10 \cdot 12610}{100} = 1261(\text{грн}),$$

$$З_{дод.в.} = \frac{10 \cdot 13630}{100} = 1363(\text{грн}),$$

$$Z_{\text{дод}} = Z_{\text{дод.к.}} + Z_{\text{дод.в.}} = 2624(\text{грн}).$$

4.2.2 Відрахування на соціальні заходи

Заробітна плата робітників відсутня, тому $Z_p = 0$. Нарахування на заробітну плату дослідників та нарахування на заробітну плату працівників, які брали участь у цьому етапі роботи, розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників за формулою:

$$Z_n = (Z_o + Z_p + Z_{\text{дод}}) \cdot \frac{N_{\text{зп}}}{100\%}, \quad (4.3)$$

де $N_{\text{зп}}$ — норма нарахування на заробітну плату.

$$Z_n = (26240 + 0 + 2624) \cdot \frac{22\%}{100\%} = 6350 (\text{грн.})$$

4.2.3 Сировина та матеріали

Витрати на матеріали (M), у вартісному вираженні розраховуються окремо по кожному виду матеріалів за формулою:

$$M = \sum_{j=1}^n N_j \cdot C_j \cdot K_j - \sum_{j=1}^n B_j \cdot C_{\text{в}j}, \quad (4.4)$$

де N_j — кількість матеріалу j -го виду, шт.;

n — кількість видів матеріалу.

C_j — ціна матеріалу j -го виду, грн;

K_j — коефіцієнт транспортних витрат, $K_j = (1, 1 \dots 1, 15)$. Обираємо $K_j 1, 15$;

B_j — маса відходів j -го найменування, кг;

$C_{\text{в}j}$ — вартість відходів j -го найменування, грн/кг.;

Результати розрахунків занесено до таблиці 4.6.

Таблиця 4.6 — Витрати на матеріали

Найменування комплектуючих	Ціна за 1 штуку, грн	Кількість матеріалу, штук	Величина відходів, %	Ціна відходів, грн/%	Вартість витраченого матеріалу, грн
USB кабель	34,00	1	0	0	34,00
Пачка офісного паперу	110,00	1	0,5	55,00	55,00
Всього					89,00

4.2.4 Розрахунок витрат на комплектуючі

З комплектуючих використано тільки одну ПЛІС, вартість якої — 1500 грн, тому витрат на компоненти $K_v = 1500$ грн.

4.2.5 Спецустаткування для наукових (експериментальних) робіт

В якості спецустаткування для проведення експериментальних робіт по розробці простого процесора використано ПЛІС, яка використовуватиметься в кінцевому пристрої. Тому затрати на спецустаткування дорівнюють нулю.

4.2.6 Програмне забезпечення для наукових (експериментальних) робіт

Для проектування процесору на базі ПЛІС використовується програмне забезпечення, яке йшло в комплекті з ПЛІС тому витрати на придбання такого забезпечення відсутні. Це програмне середовище Quartus II, в якому використовуватиметься мова опису VHDL.

4.2.7 Амортизація обладнання, програмних засобів та приміщень

Амортизаційні відрахування по кожному виду обладнання, приміщень та програмному забезпеченню можуть бути розраховані з використанням прямолінійного методу амортизації за формулою:

$$A_{\text{обл}} = \frac{C_0}{T_B} \cdot \frac{t_{\text{вик}}}{12}, \quad (4.5)$$

Таблиця 4.7 — Амортизаційні відрахування по кожному виду обладнання

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання, місяців.	Амортизаційні відрахування, грн
Ноутбук	19000	2	2	1583,33
Оренда приміщення	120000	20	2	1000
Всього				2583,33

4.2.8 Паливо та енергія для науково-виробничих цілей

Витрати на силову електроенергію (B_e) розраховують за формулою:

$$B_e = \sum_{i=1}^n \frac{W_{yi} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (4.6)$$

де W_{yi} — встановлена потужність обладнання на певному етапі розробки, кВт;

t_i — тривалість роботи обладнання на етапі дослідження, год;

C_e — вартість 1 кВт-години електроенергії, грн; (вартість електроенергії визначається за даними енергопостачальної компанії), $C_e = 4,62$;

$K_{впi}$ — коефіцієнт, що враховує використання потужності, $K_{впi} < 1$; обрано $K_{впi} = 0,7$;

η_i — коефіцієнт корисної дії обладнання, $\eta_i < 1$.

$$B_e = \sum_{i=1}^1 \frac{0,24 \cdot 360 \cdot 4,62 \cdot 0,6}{0,8} = 299,37 \text{ (грн.)},$$

Проведені розрахунки зведено у таблиці 4.8.

Таблиця 4.8 — Витрати на електроенергію

Найменування обладнання	Встановлена потужність, кВт	Тривалість роботи, год	Сума, грн
Ноутбук	0,24	360	299,37
Всього			299,37

4.2.9 Службові відрядження

Під час проектування пристрою відрядження штатних працівників, працівників організацій, які працюють за договорами цивільно-правового характеру, магістрів, зайнятих розробленням досліджень, відрядження, пов'язані з проведенням випробувань машин та приладів, а також витрати на відрядження на наукові з'їзди, конференції, наради, пов'язані з виконанням конкретних досліджень, не плануються.

4.2.10 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Витрати за статтею «Витрати на роботи, які виконують сторонні підприємства, установи і організації» не плануються, так як у цьому не має потреби.

4.2.11 Інші витрати

Витрати за статтею «Інші витрати» розраховуються як 50...100% від суми основної заробітної плати дослідників та робітників за формулою:

$$I_{\text{в}} = (Z_{\text{о}} + Z_{\text{р}}) \cdot \frac{N_{\text{ів}}}{100\%}, \quad (4.7)$$

де $N_{\text{ів}}$ — норма нарахування за статтею «Інші витрати».

$$I_{\text{в.к.}} = 12610 \cdot \frac{50\%}{100\%} = 6305(\text{грн.}),$$

$$I_{\text{в.ін.}} = 13630 \cdot \frac{50\%}{100\%} = 6815(\text{грн.}),$$

$$I_{\text{в}} = I_{\text{в.к.}} + I_{\text{в.ін.}} = 13120(\text{грн.}).$$

4.2.12 Накладні (загальновиробничі) витрати

Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100...150% від суми основної заробітної плати дослідників та робітників за формулою:

$$V_{\text{нзв}} = (Z_{\text{о}} + Z_{\text{р}}) \cdot \frac{N_{\text{нзв}}}{100\%}, \quad (4.8)$$

де $N_{\text{нзв}}$ — норма нарахування за статтею «Накладні (загальновиробничі) витрати». За норму береться нарахування 100%.

$$V_{\text{нзв.к.}} = 12610 \cdot \frac{100\%}{100\%} = 12610(\text{грн.}),$$

$$V_{\text{нзв.в.}} = 13630 \cdot \frac{100\%}{100\%} = 13630(\text{грн.}),$$

$$V_{\text{нзв}} = V_{\text{нзв.к.}} + V_{\text{нзв.в.}} = 26240(\text{грн.}).$$

Витрати на проведення науково-дослідної роботи розраховуються як сума всіх попередніх статей витрат за формулою:

$$V_{\text{заг}} = Z_o + Z_p + Z_{\text{дод}} + Z_n + M + K_v + V_{\text{спец}} + V_{\text{прг}} + A_{\text{обл}} + V_e + V_{\text{св}} + V_{\text{сп}} + I_v + V_{\text{нзв}}. \quad (4.9)$$

При розробці пристроя $Z_p = 0, V_{\text{спец}} = 0, V_{\text{прг}} = 0, V_{\text{св}} = 0, V_{\text{сп}} = 0$, тому формула спрощується до:

$$V_{\text{заг}} = Z_o + Z_{\text{дод}} + Z_n + M + K_v + A_{\text{обл}} + V_e + I_v + V_{\text{нзв}} = 26240 + 2624 + 6350 + 89,00 + 2583,33 + 299,37 + 13120 + 26240 = 77545,37(\text{грн}).$$

Загальні витрати ЗВ на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою:

$$ЗВ = \frac{V_{\text{заг}}}{\eta}, \quad (4.10)$$

де η — коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи. Під час він дослідження дорівнює 0,8.

$$ЗВ = \frac{77545,37}{0,8} = 96931,71(\text{грн}).$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

При розробці або суттєвій модифікації електронного пристрою апаратно-програмного засобу можливе збільшення чистого прибутку для потенційного інвестора $\Delta\Pi_i$ у кожному із років, на протязі яких очікується отримання результатів від можливого впровадження та комерціалізації науково-технічної розробки, розраховується за формулою:

$$\Delta\Pi_i = (\pm\Delta\Pi_0 \cdot N + \Pi_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot \left(1 - \frac{\vartheta}{100}\right), \quad (4.11)$$

де $\pm\Delta\Pi_0$ — зміна основного якісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

N — основний кількісний показник, який визначає величину попиту на аналогічні чи подібні розробки у році до впровадження результатів нової науково-технічної розробки;

Π_0 — основний якісний показник, який визначає ціну реалізації нової науково-технічної розробки в аналізованому році, $\Pi_0 = \Pi_6 \pm \Delta\Pi_0$;

Π_6 — основний якісний показник, який визначає ціну реалізації існуючої (базової) науково-технічної розробки у році до впровадження результатів;

ΔN — зміна основного кількісного показника від впровадження результатів науково-технічної розробки в аналізованому році;

λ — коефіцієнт, який враховує сплату потенційним інвестором податку на додану вартість. У 2021 році ставка податку на додану вартість становить 20%, а коефіцієнт $\lambda=0,8333$;

ρ — коефіцієнт, який враховує рентабельність інноваційного продукту (послуги). Рекомендується брати $\rho=0,2\dots0,5$. Обраємо $\rho=0,26$;

ϑ — ставка податку на прибуток, який має сплачувати потенційний інвестор, у 2021 році $\vartheta=18\%$.

При впровадженні результатів науково-дослідної розробки покращується якість програмно-апаратного комплексу, що дозволяє підвищити вартість за його впровадження, а число потенційних користувачів пристрою збільшиться — у перший рік — на 100 одиниць, на другий рік — ще на 420 одиниць, на третій рік — ще 530 штук.

Прогнозується щорічний приріст прибутку компанії при впровадженні результатів науково-дослідницьких розробок. Збільшення чистого прибутку підприємства $\Delta\Pi_i$ за перший рік складе:

$$\Delta\Pi_1 = [1000 \cdot 0 + (3000 + 1000) \cdot 100] \cdot 0,8333 \cdot 0,26 \cdot \left(1 - \frac{18\%}{100\%}\right) = 71063,82 \text{ (грн)}.$$

Збільшення чистого прибутку компанії $\Delta\Pi_i$ на другий рік (порівняно з базовим, тобто роком, що передує впровадженню результатів наукових досліджень) складе:

$$\begin{aligned}\Delta\Pi_2 &= [1000 \cdot 0 + (3000 + 1000) \cdot (100 + 420)] \cdot 0,8333 \cdot 0,26 \cdot \left(1 - \frac{18\%}{100\%}\right) \\ &= 369531,88 \text{ (грн.)}\end{aligned}$$

Збільшення чистого прибутку підприємства $\Delta\Pi_i$ на третій рік складе:

$$\begin{aligned}\Delta\Pi_3 &= [1000 \cdot 0 + (3000 + 1000) \cdot (100 + 420 + 530)] \cdot 0,8333 \cdot 0,26 \cdot \left(1 - \frac{18\%}{100\%}\right) \\ &= 746170,15 \text{ (грн.)}\end{aligned}$$

Далі розраховується вартість збільшення усього чистого прибутку ПП, який може отримати потенційний інвестор при впровадженні та комерціалізації науково-дослідницької розробки:

$$\text{ПП} = \sum_{i=1}^T \frac{\Delta\Pi_i}{(1 + \tau)^t}, \quad (4.12)$$

де $\Delta\Pi_i$ — збільшення чистого прибутку у кожному з років, протягом яких виявляються результати впровадження науково-дослідницької розробки, грн;

T — період часу, протягом якого очікується отримання позитивних результатів від впровадження та комерціалізації науково-дослідницької розробки, роки;

τ — ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0,05 \dots 0,15$. Обирається $\tau 0,1$;

t — період часу (в роках) від початку впровадження науково-технічної розробки до отримання інвестором додаткових чистих прибутків в поточному році.

$$\begin{aligned}\text{ПП} &= \frac{71063,82}{(1 + 0,1)^1} + \frac{369531,88}{(1 + 0,1)^2} + \frac{746170,15}{(1 + 0,1)^3} = 64603,47 + 305398,25 + 560608,68 \\ &= 930610,4 \text{ (грн.)}\end{aligned}$$

Далі розраховується величина початкових інвестицій PV , які потенційний інвестор вкладає для введення та комерціалізації науково-дослідницької розробки. Для цього використовується формула:

$$PV = k_{\text{інв}} \cdot 3B, \quad (4.13)$$

де $k_{\text{інв}}$ — коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{\text{інв}} = 2 \dots 5$, але може бути і більшим. Обираємо даний коефіцієнт 2;

$3B$ — загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 \cdot 96931,71 = 193863,42 \text{ (грн.)}$$

Тоді абсолютний економічний ефект $E_{\text{абс}}$ або чистий приведений дохід потенційного інвестора при впровадженні та комерціалізації науково-дослідницької розробки становитиме:

$$E_{\text{абс}} = \text{ПП} - PV, \quad (4.14)$$

де ПП — приведена вартість зростання всіх чистих прибутків від можливого впровадження та комерціалізації науково-технічної розробки, грн;

PV — теперішня вартість початкових інвестицій, грн.

$$E_{\text{абс}} = 930610,4 - 193863,42 = 736746,98 \text{ (грн.)}$$

Внутрішня економічна дохідність інвестицій $E_{\text{в}}$, які може вкласти потенційний інвестор у встановлення та комерціалізацію науково-дослідницької розробки, розраховується за формулою:

$$E_{\text{в}} = \sqrt[T_{\text{ж}}]{1 + \frac{E_{\text{абс}}}{PV}} - 1, \quad (4.15)$$

де $E_{\text{абс}}$ — абсолютний економічний ефект вкладених інвестицій, грн;

PV — теперішня вартість початкових інвестицій, грн;

$T_{\text{ж}}$ — життєвий цикл науково-дослідницької розробки, тобто час з початку розробки до закінчення отримання позитивних результатів від її впровадження, роки.

$$E_B = \sqrt[3]{1 + \frac{736746,98}{193863,42}} - 1 = 0,59$$

Далі розраховується бар'єрна ставка дисконтування τ_{\min} , тобто мінімальна внутрішня економічна дохідність інвестицій, нижче якої кошти у впровадження науково-дослідницької розробки та її комерціалізацію не будуть вкладати.

Мінімальна внутрішня економічна дохідність вкладених інвестицій τ_{\min} визначається за формулою:

$$\tau_{\min} = d + f, \quad (4.16)$$

де d — середньозважена ставка за депозитними операціями в комерційних банках; в 2021 році в Україні $d=0,9...0,12$. Обирається $d = 0,1$;

f — показник, що характеризує ризикованість вкладення інвестицій; зазвичай величина $f=0,05...0,5$, але може бути і значно вищою. Обирається $0,25$;

$$\tau_{\min} = d + f = 0,1 + 0,25 = 0,35\%$$

Величина $E_B > \tau_{\min}$. Отже інвестор може бути зацікавлений у фінансуванні цього дослідження.

Далі розраховується період окупності інвестицій $T_{ок}$, які можуть бути вкладені потенційним інвестором у впровадження та комерціалізацію науково-технічної розробки:

$$T_{ок} = \frac{1}{E_B}, \quad (4.17)$$

де E_B — внутрішня економічна дохідність вкладених інвестицій.

$$T_{ок} = \frac{1}{0,57} = 1,7 \text{ року}$$

Оскільки $T_{ок} = 1,7$ року, то розвиток доречний.

ВИСНОВКИ

В даній магістерській кваліфікаційній роботі реалізовано програмні та апаратні інструментальні засоби, які можна використовувати при проектуванні та реалізації електронних пристроїв на базі soft-мікропроцесора ПЛІС.

Використання розроблених інструментальних засобів дозволяє спростити задачу розробки та налагодження окремих елементів у готових схемах з можливістю повторного використання компонентів, складових бібліотеки параметризованих модулів та розроблених самотужки.

Використовуючи за основу програмовані логічні інтегральні схеми, можна реалізувати різні за складністю та функціоналом електронні пристрої, зокрема:

- пристрої порівняння;
- тригери та регістри;
- пристрої підрахунку кількості сигналів;
- таймери і годинників реального часу;
- суматори, піднімачі і множники;
- використовувати в якості блоків пам'яті;
- прості та soft-процесори.

Розробка будь-якого з даних пристроїв не прив'язана до конкретної моделі ПЛІС, що дозволяє розробити них навіть на мінімальній конфігурації плати Altera DE0. Розроблювані пристрої можуть використовуватись у задачах навчання, створення прототипів та досліджень в області комп'ютерних систем, функціонально-орієнтованих контролерів та інших вузько направлених галузях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Васильев А.Е. Автоматизированные информационно-управляющие системы. Встраиваемые системы управления: Навчальний підручник. СПб. : Изд-во Политехн. Ун-та, 2013. 106 с.
2. Проблемно і функціонально орієнтовані комп'ютерні системи та мережі: веб-сайт. URL: <http://dspace.nbuiv.gov.ua/bitstream/handle/123456789/12005/09-Petrenko.pdf?sequence=1> (дата звернення: 14.12.2021).
3. All FPGA Main Boards: веб-сайт. URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=13&List=Simple> (дата звернення: 14.12.2021)
4. EDK Concepts, Tools, and Techniques: веб-сайт. URL: https://www.xilinx.com/support/documentation/sw_manuals/xilinx14_7/edk_ctt.pdf (дата звернення: 14.12.2021)
5. Гандзюк І.В. Проектування компаратора з додатковою системою перевірки правильності функціонування на базі програмованої логічної інтегральної схеми. / І.В. Гандзюк, О.Д. Азаров, С.В. Богомоллов. // Тези доповіді. Молодь в науці: дослідження, проблеми, перспективи (МН-2022): веб-сайт. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/view/9369/7737> (дата звернення: 14.12.2021)
6. Осипов Ю.С. Контроллер//Большая российская энциклопедия. 2010. — Том 15. С. 152-153.
7. CS303 Digital Design Lab1: веб-сайт. URL: https://ee.ius.edu.ba/sites/default/files/u747/cs303_f19_lab1.pdf (дата звернення: 14.12.2021)
8. Надвелика інтегральна схема (НВІС): веб-сайт. URL: <https://uk.hoboetc.com/obrazovanie/25734-sverhbolshaya-integralnaya-shema-sbis-nazvana-tak-potomu-chto-sverhbolshaya-integralnaya-shema-razmery-ves-i-opisanie.html> (дата звернення: 14.12.2021).
9. WISHBONE System-on-Chip (SoC) Interconnection Architecture for Portable IP Cores: веб-сайт. URL: https://cdn.opencores.org/downloads/wbspec_b4.pdf (дата звернення: 14.12.2021).

10. Altera DE0 Board: веб-сайт. URL: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=56&No=364> (дата звернення: 15.12.2021).

11. Гандзюк І.В. проектування цифрових пристроїв на програмованих логічних інтегральних схемах. / І.В. Гандзюк, С.В. Богомолов. // Тези доповіді. І Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2021): веб-сайт. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2021/paper/view/12442> (дата звернення: 15.12.2021)

12. Гандзюк І.В. Проектування аналогово-цифрових та цифро-аналогових систем на базі ПЛІС. / І.В. Гандзюк, С.В. Богомолов. // Тези доповіді. XLIX регіональна науково-технічна конференція професорсько-викладацького складу, співробітників та студентів університету з участю працівників науково-дослідних організацій та інженерно-технічних працівників підприємств м. Вінниці та області: веб-сайт. URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2020/paper/view/9369/7737> (дата звернення: 14.12.2021)

13. APEX 20K та QUARTUS: веб-сайт. URL: http://www.chipnews.ru/html.cgi/arhiv/99_09/stat_19.htm (дата звернення: 17.12.2021)

14. Графический дизайн или текст: веб-сайт. URL: <https://marsohod.org/11-blog/251-sch-or-txt> (дата звернення: 17.12.2021)

15. Процессор на плис: веб-сайт. URL: <https://all-audio.pro/c7/shemi/protessor-na-plis.php> (дата звернення: 17.12.2021)

16. Архитектуры софт-процессоров: веб-сайт. URL: <https://www.elibrary.ru/item.asp?id=41829907> (дата звернення: 17.12.2021)

17. Таблица для выбора FPGA: веб-сайт. URL: <https://fpga.su/#> (дата звернення: 17.12.2021)

18. Современные системы проектирования для ПЛИС: веб-сайт. URL: http://www.compitech.ru/html.cgi/arhiv/02_06/stat_128.htm (дата звернення: 17.12.2021)

19. Основы языка VHDL: веб-сайт. URL: http://e-learning.bmstu.ru/moodle/pluginfile.php/2908/mod_resource/content/2/%D0%AD%D0%92%D0%9C_3.pdf (дата звернення: 17.12.2021)

20. ПЛИС: веб-сайт. URL: <http://ru.bmstu.wiki/ПЛИС> (дата звернення: 17.12.2021)

21. Программируемые логические интегральные схемы: веб-сайт. URL: <http://vmss.mpei.ru:15980/main/files/PinaevV/lecture14-15.pdf> (дата звернення: 17.12.2021)

22. Микросхемы: веб-сайт. URL: <http://altera.ru/Mikroshemi.html> (дата звернення: 17.12.2021)

ДОДАТОК А

Міністерство освіти та науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ

Завідувач кафедри ОТ ВНТУ

д.т.н., проф.

_____ Азаров О.Д.

“ ___ ” _____ 2021 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської дипломної роботи

«Метод та засоби проектування аналогово-цифрових та цифро-аналогових систем
на базі ПЛІС»

08-23.МКР.019.00.000 ТЗ

Науковий керівник к.т.н., доцент

_____ Богомолів С.В.

Студент групи 2КІ-20м

_____ Гандзюк І.В.

Вінниця 2021

1 Найменування та область застосування

Робоча назва проекту «Метод та засоби проектування аналогово-цифрових та цифро-аналогових систем на базі ПЛІС», розробляється для того, щоб навчитись створювати та проектувати комп'ютерні системи.

2 Основи для розробки

Основою для розробки є дисципліни Комп'ютерна логіка, Комп'ютерна електроніка, Комп'ютерні системи, Технології проектування комп'ютерних систем.

3 Мета та призначення розробки

Експлуатаційне призначення розробки — для створення цифрових пристроїв з використанням плат ПЛІС Altera.

4 Етапи МКР та очікувані результати

Робота виконується в шість етапів, що наведені в таблиці А.1.

Таблиця А.1 — Етапи виконання роботи

№ етапу	Назва етапу	Термін виконання		Очікувані результати
		початок	кінець	
1	Аналіз завдання. Вступ	05.10.21	05.10.21	Вступ
2	Аналіз особливостей програмованих логічних інтегральних схем	06.10.21	13.10.21	Розділ 1
3	Порівняння ПЛІС та програмного забезпечення від основних виробників	14.10.21	22.10.21	Розділ 2
4	Створення схеми та програми пристрою	25.10.21	16.11.21	Розділ 3
5	Підготовка матеріалів та опис розробки, Оформлення пояснювальної записки та ілюстративного матеріалу	17.11.21	11.12.21	Пояснювальна записка
6	Аналіз виконання роботи, висновки, додатки	13.12.21	5.12.21	Висновки, додатки, презентація

5 Матеріали, що подаються до захисту МКР

Пояснювальна записка МКР, графічні та ілюстративні матеріали, протокол попереднього захисту МКР на кафедрі, відгук наукового керівника, рецензія опонента, протоколи складання державних екзаменів, анотації до МКР українською та іноземною мовами, довідка про відповідність оформлення МКР діючим вимогам.

6 Порядок контролю виконання та захисту МКР

Виконання етапів графічної та розрахункової документації МКР контролюється науковим керівником згідно зі встановленими термінами. Захист МКР відбувається на засіданні Державної екзаменаційної комісії, затвердженою наказом ректора.

7 Вимоги до оформлення МКР

Вимоги викладені в МЕТОДИЧНИХ ВКАЗІВКАХ до дипломного проектування, ДСТУ 3008:2015, ДСТУ 3974-2000 «Правила виконання дослідно-конструкторських робіт. Загальні положення» та діючого ГОСТ 2.114-95 ЄСКД.

Вимоги щодо технічного захисту інформації в МКР з обмеженим доступом відсутні.

Технічне завдання до виконання отримав _____ Гандзюк І.В.

ДОДАТОК Б

Зображення контактів утиліти Pin Planer

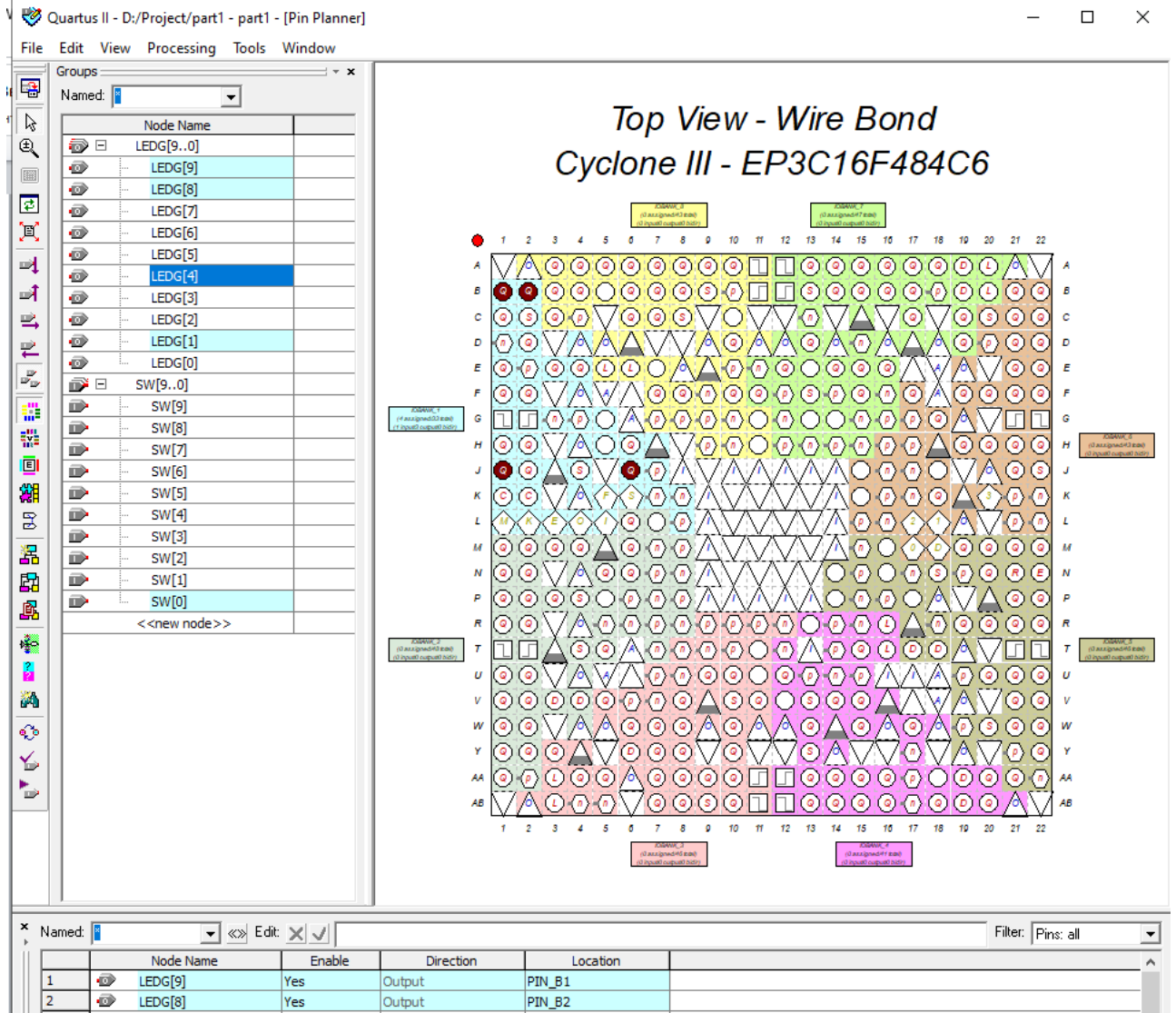


Рисунок Б.1 — Зображення контактів утиліти Pin Planer в ПЗ Quartus

ДОДАТОК В

Зображення фрагментів Datasheet для плати ПЛІС Altera DE0

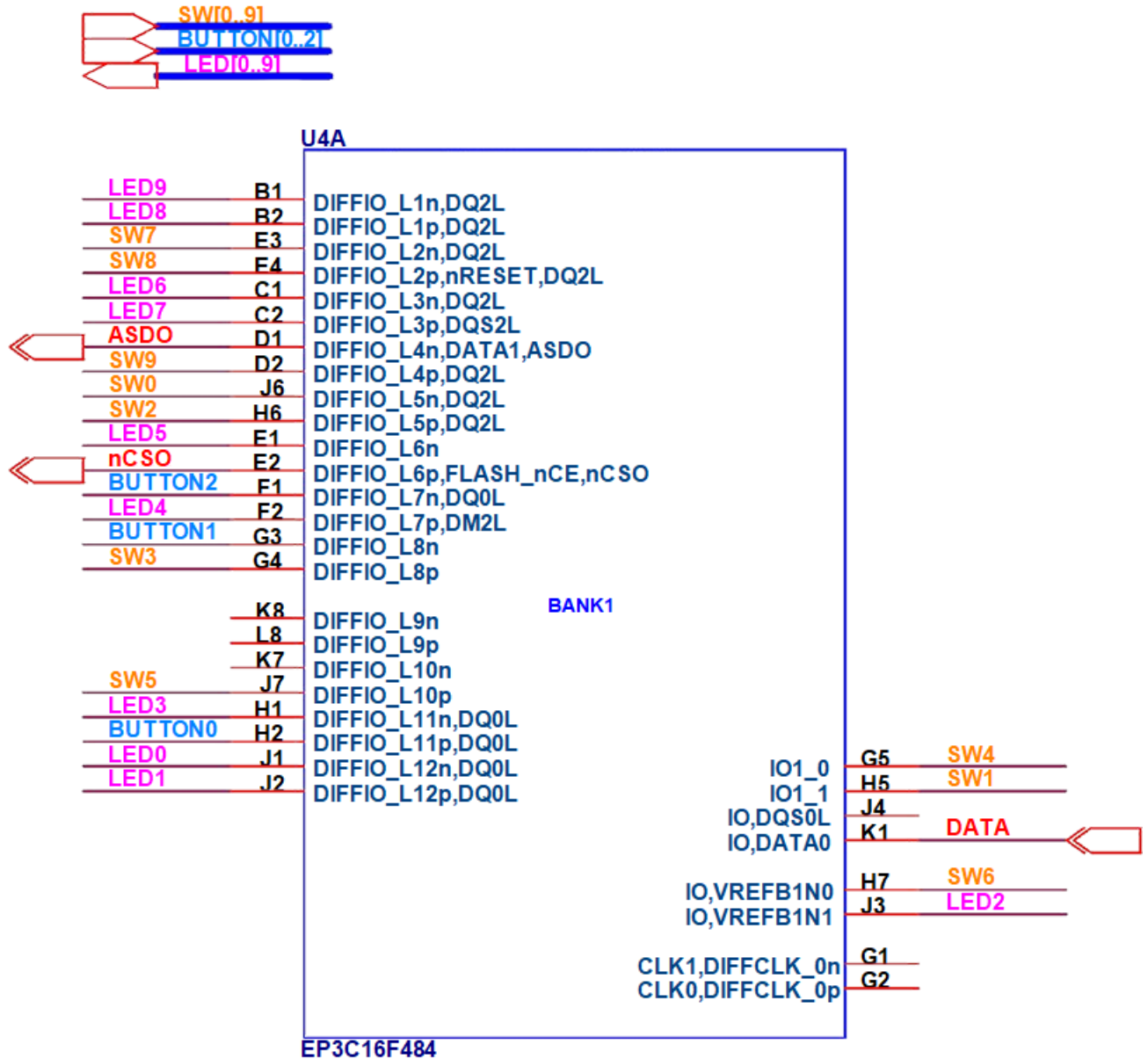


Рисунок В.1 — зображення контактів, що відповідають перемикачам, світлодіодам та кнопкам на платі ПЛІС Altera DE0

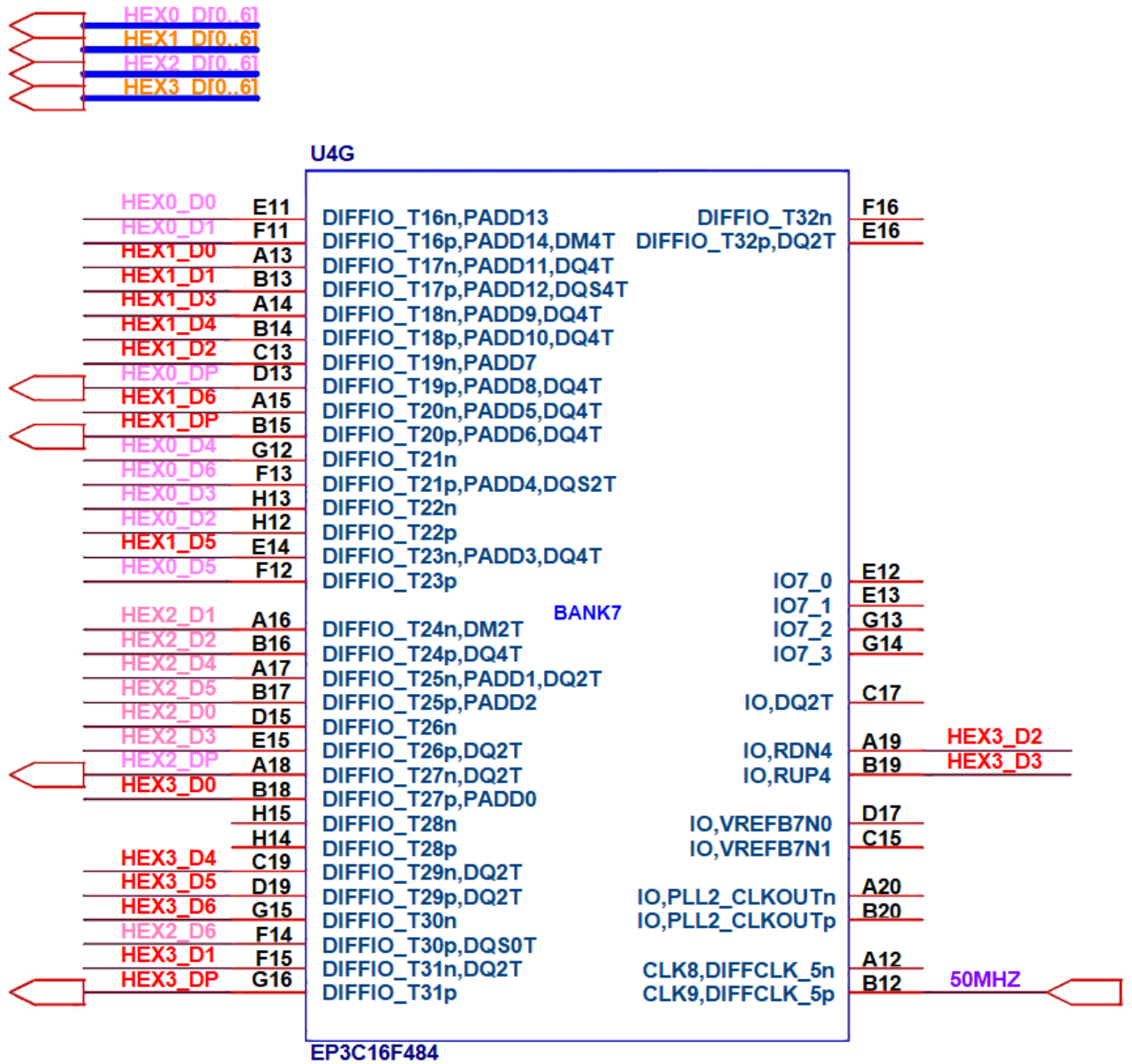


Рисунок В.2 — зображення контактів, що відповідають 7-сегментним дисплеям та тактовому генераторі сигналу на платі ПЛІС Altera DE0

ДОДАТОК Г

Зовнішній вигляд програмованої логічної інтегральної схеми

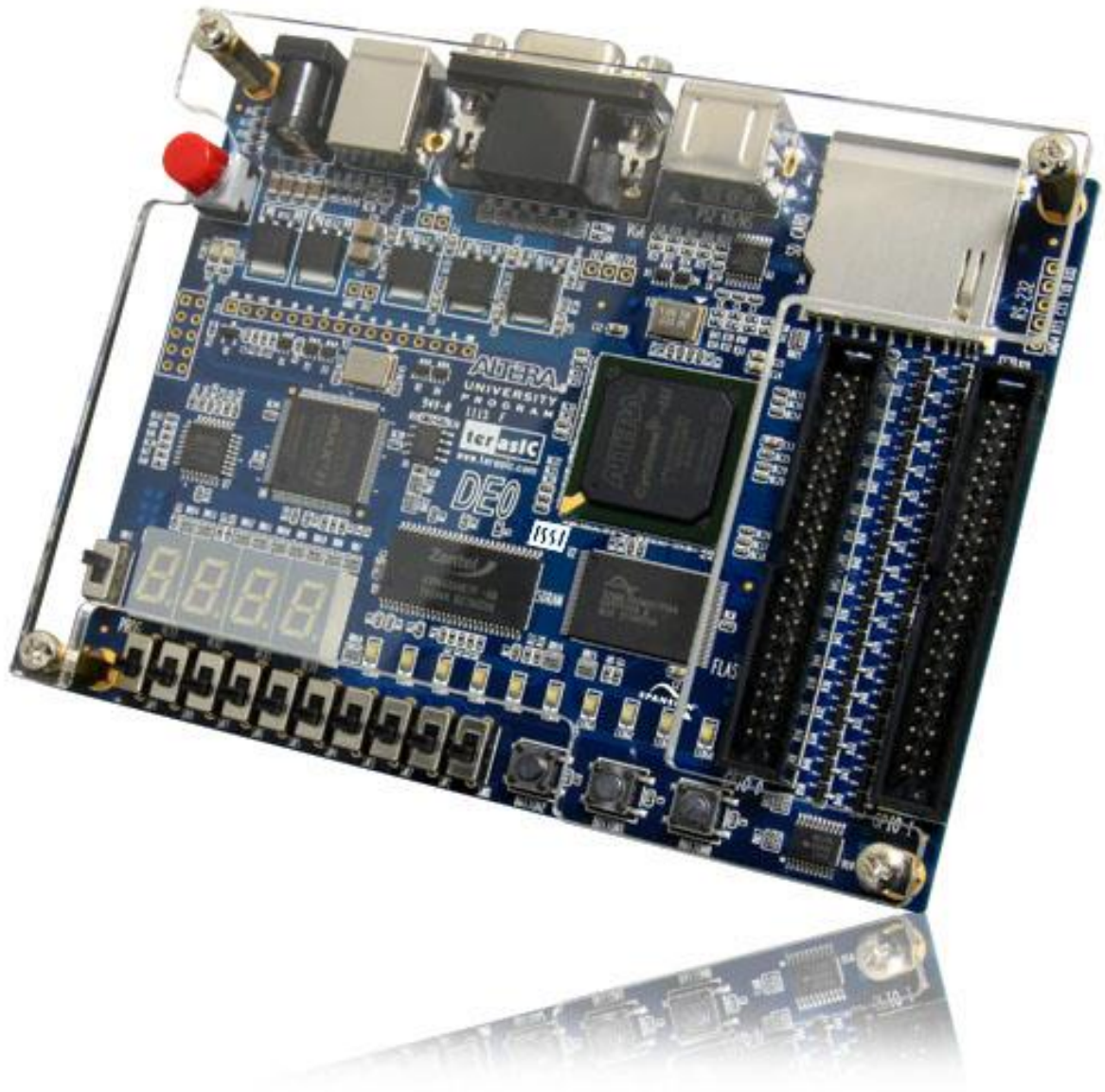


Рисунок Г.1 — Зовнішній вигляд плати ПЛІС Altera DE0

ДОДАТОК Д

Схема простого процессора

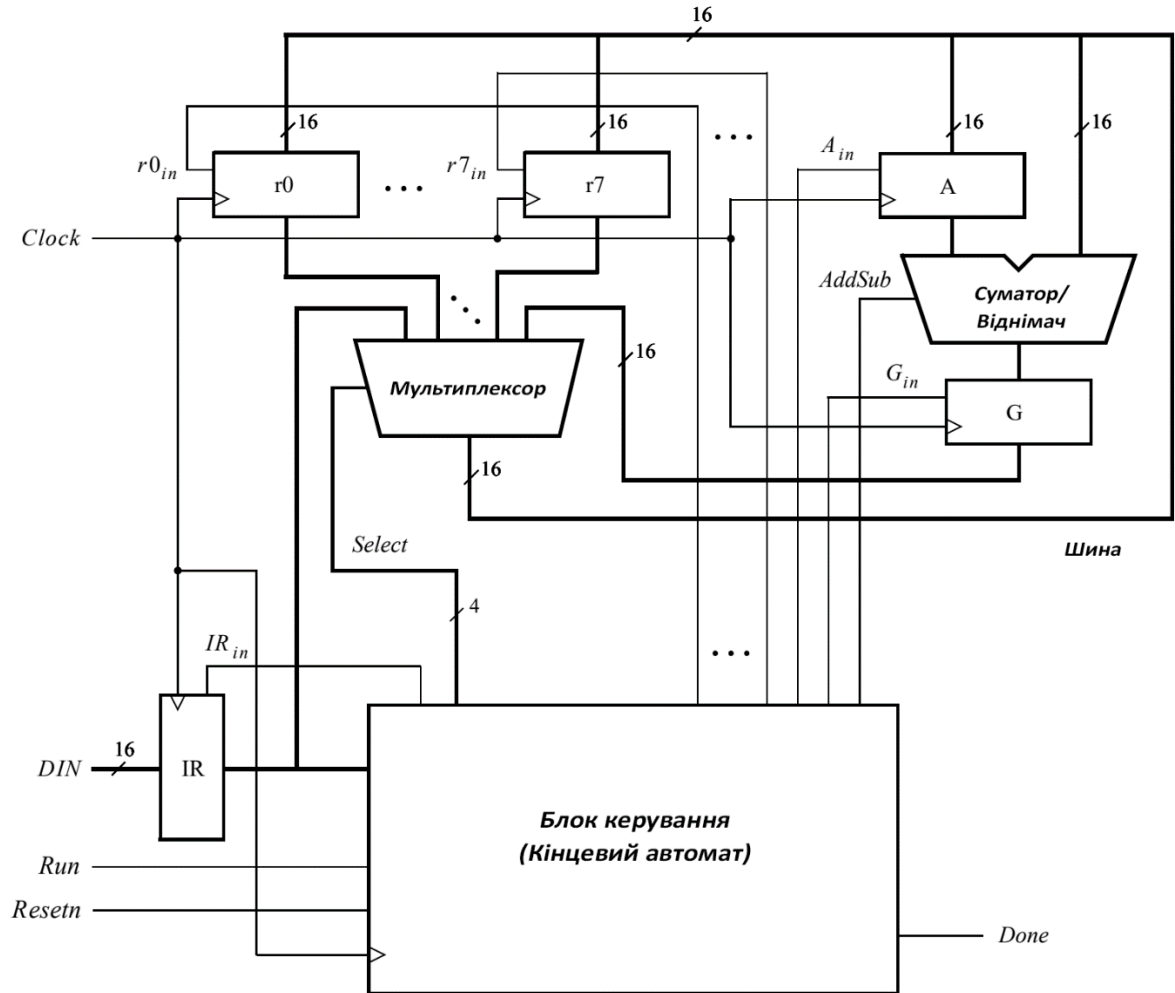


Рисунок Д.1 — Схема простого процессора

ДОДАТОК Е

Протокол перевірки навчальної (кваліфікаційної) роботи