

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

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

**КАФЕДРА СИСТЕМОГО ПРОГРАМУВАННЯ І  
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ**

«На правах рукопису»  
УДК 004.31

«До захисту допущено»  
Завідувач кафедри СПСКС

\_\_\_\_\_ Тарасенко В.П.  
(підпис) (ініціали, прізвище)  
“ \_\_\_\_ ” \_\_\_\_\_ 2018 р.

## **Магістерська дисертація**

**на здобуття ступеня магістра**

з напрямку підготовки 123 Комп'ютерна інженерія  
Спеціалізовані комп'ютерні системи

на тему: ЗАСОБИ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ НА ПЛІС ДЛЯ  
МОНІТОРИНГУ ОБ'ЄКТІВ КРИТИЧНОГО ЗАСТОСУВАННЯ

Виконав: студент II курсу, групи КВ-73мп  
(шифр групи)

Хуповець Дмитро Юрійович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Керівник: к.т.н., доцент Клятченко Я.М. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент: \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних посилань.  
Студент \_\_\_\_\_  
(підпис)

Київ – 2018 року

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

Факультет прикладної математики  
Кафедра системного програмування і спеціалізованих комп'ютерних систем  
Рівень вищої освіти – другий (магістерський)  
Спеціальність 123 Комп'ютерна інженерія  
Спеціалізовані комп'ютерні системи

ЗАТВЕРДЖУЮ  
Завідувач кафедри СПСКС  
\_\_\_\_\_ Тарасенко В.П.  
(підпис) (ініціали, прізвище)  
“ \_\_\_\_ ” \_\_\_\_\_ 2018 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**

Хуповцю Дмитру Юрійовичу  
(прізвище, ім'я, по батькові)

1. Тема дисертації «Засоби паралельної обробки даних на ПЛІС для моніторингу об'єктів критичного застосування»  
науковий керівник дисертації к.т.н., доцент Клятченко Я.М.  
затверджені наказом по університету від 30 жовтня 2018 р. №4030-с.
2. Термін подання студентом дисертації 7 грудня 2018 р.
3. Об'єкт дослідження: способи реалізації паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування.
4. Предмет дослідження: структура спеціалізованого пристрою паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування.
5. Перелік завдань, які потрібно розробити полягає в огляді та аналізі методів паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування.

6. Перелік ілюстративного матеріалу: презентація (кількість аркушів: 15).

7. Перелік публікацій 1) XI Конференція молодих вчених «Прикладна математика та комп'ютеринг» ПМК-2018-2 «Паралельна обробка даних на ПЛІС»; 2) IV Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами «Паралельна обробка даних на засобах ПЛІС», «Кросплатформене розподілення сенсорних мереж за допомогою засобів на ПЛІС».

8. Дата видачі завдання 5 вересня 2017.

#### Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Вивчення літератури за тематикою проекту	25.08.2018	
2	Аналіз існуючих рішень	22.09.2018	
3	Підготовка матеріалів першого розділу дипломного проекту	18.10.2018	
4	Підготовка матеріалів другого розділу дипломного проекту	27.10.2018	
5	Підготовка матеріалів третього розділу дипломного проекту	05.11.2018	
6	Підготовка матеріалів четвертого розділу дипломного проекту	15.11.2018	
7	Оформлення документації дипломного проекту	20.11.2018	
8	Попередній огляд матеріалів диплому на кафедрі	25.11.2018	

Студент

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (прізвище, ініціали)

Науковий керівник дисертації

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (прізвище, ініціали)

## РЕФЕРАТ

**Актуальність теми.** На сьогодні є велика кількість галузей виробництва, де необхідно забезпечувати високий рівень надійності та безпеки використання об'єктів критичного застосування. Це такі сфери підприємства як:

1. енергетична галузь, а саме атомні електростанції (АЕС), теплові електростанції (ТЕС), гідро електростанції (ГЕС);
2. підприємства із небезпечним виробництвом, які мають велику ймовірність техногенних катастроф;
3. космічна галузь;
4. транспортна галузь.

Безпека та надійність використання таких об'єктів критичного застосування залежить від рівня автоматизованості певної промисловості. Рівень автоматизованості залежить від таких показників, як безвідмовність роботи обчислювальних систем, довершеність архітектурних та програмних рішень, відмовостійкість програмно-апаратних частин, час ремонту тощо.

При виробництві цифрових інтегральних схем (ЦІС) використовуються різні електронні компоненти. Це такі компоненти як базові матричні кристали, вузькоспеціалізовані великі інтегральні схеми (ВІС), які випускаються обмеженими серіями, програмовні логічні контролери тощо. Досить ефективною альтернативою такого компонента є програмовна логічна інтегральна схема (ПЛІС). ПЛІС – це спеціальний електронний компонент, який використовується при створенні ЦІС. ПЛІС відрізняється від інших ЦІС тим, що логіка роботи ПЛІС не задається при її виготовленні. Для задання певної логіки роботи ПЛІС використовують спеціальні програматори та налагоджувальні середовища, що дозволяє задати потрібну структуру цифрового пристрою, а саме у вигляді принципової схеми або програмного коду створеного завдяки спеціальним системам автоматизованого проектування (САПР). Сучасні ПЛІС мають такі характеристики, як

властивість багаторазової переконфігурації, низька вартість виготовлення, низька енергія споживання, висока швидкодія.

**Об'єктом дослідження** є способи реалізації паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування.

**Предметом дослідження** є структура спеціалізованого пристрою паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування.

**Мета роботи** – визначення способу організації та реалізації паралельної обробки даних на основі ПЛІС для ефективного моніторингу об'єктів критичного застосування.

**Методи дослідження.** В роботі використовуються методи математичного опису алгоритму, методи аналізу та порівняння роботи алгоритмів, комп'ютерної схемотехніки, теорії організації обчислювальних процесів для створення спеціалізованих комп'ютерних систем на ПЛІС.

**Наукова новизна:**

1. Розроблено математичний опис алгоритму обробки даних для моніторингу об'єктів критичного застосування, яка відрізняється від існуючих паралельністю виконання обробки даних.

2. Запропоновано програмно-апаратну реалізацію спеціального пристрою на ПЛІС для паралельної обробки даних при моніторингу об'єктів критичного застосування.

**Практична цінність** отриманих в роботі результатів полягає в тому, що була запропонована програмно-апаратна реалізація алгоритму паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування. Основними перевагами є створення унікального пристрою невеликих розмірів із більшою продуктивністю і надійністю.

Висока продуктивність обчислювального засобу, що реалізує алгоритм паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування, обумовлена можливістю швидкої модифікації роботи алгоритму шляхом перепрограмування ПЛІС за допомогою спеціальних систем автоматизованого проектування (САПР). До недоліків реалізації такого обчислювального пристрою на основі ПЛІС можна віднести обмеженість кількості логічних елементів.

**Апробація роботи.** Результати роботи пройшли апробацію або знаходяться на стадії публікації на конференціях:

1. XI Конференція молодих вчених «Прикладна математика та комп'ютеринг» ПМК-2018-2;

2. IV Міжнародна науково-технічна Internet-конференція «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами;

Результати роботи опубліковані в 3 наукових працях, з яких 3 – тези доповідей.

**Структура та обсяг роботи.** *Магістерська дисертація складається з вступу, чотирьох розділів, висновків та додатків.*

У вступі надано загальну характеристику роботи, виконано оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачу досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи, наведено відомості про апробацію результатів та їх впровадження.

У першому розділі розглянуті загальні відомості про ПЛІС; розглянуті альтернативні електронні компоненти, такі як мікроконтролери,

мікропроцесори, однокристальні комп'ютери; розглянуто та проаналізовано основні характеристики об'єктів критичного застосування.

У другому розділі розглянуті загальні відомості про ПЛІС, а саме їх класифікації та характеристика; розглянуто основні сімейства ПЛІС, а саме ПЛІС від провідних компаній Altera та Xilinx; розглянуто класифікацію та характеристики основних сімейств ПЛІС фірми Xilinx. Визначені структурні особливості кожного із сімейств.

У третьому розділі розглянуто, проаналізовано та обґрунтовано способи та види організації обробки даних на ПЛІС; розглянуто основні засоби проектування фірми XILINX; сформульовані вимоги до САПР для проектування пристрою паралельної обробки даних на ПЛІС.

У четвертому розділі описані етапи проектування пристрою паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування; запропоновано алгоритм паралельної обробки даних; запропоновано програмно-апаратну реалізацію алгоритму паралельної обробки даних на основі ПЛІС для моніторингу об'єктів критичного застосування.

У висновках проаналізовано отримані результати роботи. Визначено умови для яких розроблена реалізація є ефективною. Також запропоновано шляхи покращення розробленого метода та рекомендації щодо подальшого дослідження поставленої задачі.

У додатках лістинг програмного коду VHDL реалізації алгоритму паралельної обробки даних на ПЛІС, лістинг програмного коду тестового об'єкту критичного застосування, текст публікацій, текст презентації.

Робота виконана на 78 аркушах, містить 4 додатки та посилання на список використаних літературних джерел з 26 найменувань. У роботі наведено 33 рисунки та 3 таблиці.

**Ключові слова:** пристрій паралельної обробки даних на ПЛІС, об'єкт критичного застосування, Xilinx, FPGA, Spartan-3, САПР, WEBPACK ISE, Verilog, VHDL.



## РЕФЕРАТ

**Актуальность темы.** На сегодняшний день есть большое количество отраслей производства, где необходимо обеспечивать высокий уровень надежности и безопасности использования объектов критического применения. Это такие сферы предприятия как:

1. энергетическая отрасль, а именно атомные электростанции(АЭС), тепловые электростанции(ТЭС), гидроэлектростанции(ГЭС);
2. предприятия с опасным производством, которые имеют большую вероятность техногенных катастроф;
3. космическая отрасль;
4. транспортная отрасль.

Безопасность и надежность использования таких объектов критического применения зависит от уровня автоматизированности определенной промышленности. Уровень автоматизации зависит от таких показателей, как: безотказность работы вычислительных систем, совершенство архитектуры и программных решений, отказоустойчивость программно – аппаратных частей, время ремонта и тому подобное.

При производстве цифровых интегральных схем (ЦИС) используются различные электронные компоненты. Это такие компоненты, как: базовые матричные кристаллы, узкоспециализированные большие интегральные схемы (УИС), которые выпускаются ограниченными сериями, программируемые логические контроллеры и другие. Достаточно эффективной альтернативой такого компонента является программируемая логическая интегральная схема (ПЛИС). ПЛИС – это специальный электронный компонент, который используется при создании ЦИС. ПЛИС отличается от других ЦИС тем, что логика работы ПЛИС не задаётся при её изготовлении. Для задания определённой логики работы ПЛИС используют специальные программаторы и отладочные среды, что позволяет задать нужную структуру цифрового устройства, а именно в виде принципиальной

схемы или программного кода, созданного благодаря специальным системам автоматизированного проектирования (САПР). Современные ПЛИС имеют такие характеристики, как: качество многократной переконфигурации, низкая стоимость изготовления, низкая энергия потребления, высокое быстродействие.

**Объектом исследования** являются способы реализации параллельной обработки данных на основе ПЛИС для мониторинга объектов критического использования.

**Предметом исследования** является структура специализированного устройства параллельной обработки данных на основе ПЛИС для мониторинга объектов критического применения.

**Цель работы** – определение способа организации и реализации параллельной обработки данных на основе ПЛИС для эффективного мониторинга критического применения.

**Методы исследования.** В работе используются методы математического описания алгоритма, методы анализа и сравнения работы алгоритмов, компьютерной схемотехники, теории организации вычислительных процессов для создания специализированных компьютерных систем на ПЛИС.

**Научная новизна:**

1. Разработано математическое описание алгоритма обработки данных для мониторинга объектов критического применения, который отличается от существующих параллельностью выполнения обработки данных.

2. Предложено программно – аппаратную реализацию специального устройства на ПЛИС для параллельной обработки данных при мониторинге объектов критического применения.

**Практическая ценность** полученных в работе результатов заключается в том, что была предложена программно – аппаратная реализация алгоритма параллельной обработки данных на основе ПЛИС для мониторинга объектов критического применения. Основными преимуществами является создание уникального устройства небольших размеров с большей производительностью и надёжностью. Высокая продуктивность вычислительного средства, что реализует алгоритм параллельной обработки данных на основе ПЛИС для мониторинга объектов критического применения, обусловлена возможностью быстрой модификации работы алгоритма путём перепрограммирования ПЛИС с помощью специальных систем автоматизированного проектирования (САПР). К недостаткам реализации такого вычислительного устройства на основе ПЛИС можно отнести ограниченность количества логических элементов.

**Апробация работы.** Результаты работы прошли апробацию или находятся на стадии публикации на конференциях:

1. XI Конференция молодых учёных «Прикладная математика и компьютеринг» ПМК-2018-2;
2. Международная научно–техническая Internet–конференция «Современные методы, информационное, программное и техническое обеспечение систем управления организационно–техническими и технологическими комплексами.

Результаты работы опубликованы в 3 научных статьях, из которых 3 тезисы докладов.

**Структура и объем работы.** *Магистерская диссертация состоит из введения, четырех глав, заключения и приложений.*

Во введении дана общая характеристика работы, выполнена оценка современного состояния проблемы, обоснована актуальность направления исследований, сформулированы цель и задачу исследований, показано научную новизну полученных результатов и практическую ценность работы, приведены сведения об апробации результатов и их внедрение.

В первом разделе рассмотрены общие сведения о ПЛИС; рассмотрены альтернативные электронные компоненты, такие как микроконтроллеры, микропроцессоры, однокристальные компьютеры; рассмотрены и проанализированы в основе характеристики объектов критического применения.

Во втором разделе рассмотрены общие сведения про ПЛИС, а именно их классификация и характеристика, рассмотрены основные семейства ПЛИС, а именно ПЛИС от ведущих компаний Altera и Xilinx; рассмотрена классификация и характеристики основных семейств ПЛИС фирмы Xilinx. Определены структурные особенности каждого из семейств.

В третьем разделе рассмотрены, проанализированы и обоснованно способы и виды организации обработки данных на ПЛИС; рассмотрены основные средства проектирования фирмы XILINX; сформулированы требования к САПР для проектирования устройства параллельной обработки данных на ПЛИС.

В четвертом разделе описаны этапы проектирования устройства параллельной обработки данных на основе ПЛИС для мониторинга объектов критического применения; предложен алгоритм параллельной обработки данных; предложено программно-аппаратную реализацию алгоритма параллельной обработки данных на основе ПЛИС для мониторинга объектов критического применения.

В выводах проанализированы полученные результаты работы. Определены условия, для которых разработана реализация является эффективной. Также предложены пути улучшения разработанного метода и рекомендации по дальнейшему исследованию поставленной задачи.

В приложениях приведена структурная схема архитектуры семейства FPGA, внутренняя схема взаимосвязей CLB, архитектура структурного модуля Xilinx Spartan-6, а также листинг программ и копии публикаций.

Работа выполнена на 78 листах, содержит 4 приложения и ссылки на список использованных литературных источников из 26 наименований. В работе приведены 33 рисунка и 3 таблицы.

**Ключевые слова:** устройство параллельной обработки данных на ПЛИС, объект критического применения, Xilinx, FPGA, Spartan-3, САПР, WEBPACK ISE, Verilog, VHDL.

## ABSTRACT

**Relevance of the topic.** Nowadays, there are a large number of branches of manufacture, where it is necessary to ensure a high level of reliability and safety using of critical objects. These are the spheres of the enterprise like:

- 1) the energy industry, namely nuclear power stations (NPP), thermal power stations (TPP), hydroelectric power stations (HPP);
- 2) enterprises with hazardous production, which have a high probability of man-made disasters;
- 3) space industry;
- 4) transport industry.

The safety and reliability of using such objects of critical application depends on the level of automation of a particular industry. The level of automation depends on such indicators as: trouble-free operation of computing systems, the perfection of the architecture and software solutions, the fault tolerance of software and hardware parts, repair time, and the like.

In the manufacture of digital integrated circuits (CIS), various electronic components are used. It is such component as: basic matrix crystals, highly specialized large integrated circuits (MIS), which are available in limited series, programmable logic controllers and others. A quite effective alternative to such a component is a programmable logic integrated circuit (PLIC). The PLIC is a special electronic component that is used when creating a CISS. PLIC is different from other CISS in that the logic of PLIC operation is not specified during its manufacture. To set a certain logic of PLIC operation, use such components as: programmers and debugging environments, which allows you to specify the desired structure of a digital device, namely in the form of a circuit diagram or software code created for special automated computer design systems (CAD).

Modern PLICs have such characteristics as: quality of multiple reconfiguration, low manufacturing cost, low energy consumption, high speed.

**The object of research** is the methods of implementing parallel data processing on the basis of PLIC for monitoring objects of critical use.

**The subject of the research** is the structure of a specialized software device of parallel data processing based on PLIC for monitoring critical objects.

**The purpose of the work** is to determine the method of organizing and implementing parallel data processing based on FPGA for effective monitoring of critical applications.

**Research methods.** The work uses the methods of mathematical description of the algorithm, methods of analysis and comparison the work of algorithms, computer self-engineering, the theory of organization of computational processes for creating specialized computer systems on FPGA.

**Scientific novelty:**

1. Developed a mathematical description of the data processing algorithm for monitoring objects of critical application, which is different from the existing parallel execution of data processing.

2. Proposed a software - hardware implementation of a special device on the PLIC for parallel data processing when monitoring objects of critical application.

**The practical value** of the results obtained in the work lies in the fact that the software and hardware implementation in proposed of the PLIC - based parallel data processing algorithm for monitoring objects of critical application. The main advantages are the creation a unique device of small size with greater performance and reliability. The high productivity of a computational tool, which implements an PLIC -based parallel data processing algorithm for monitoring objects of critical application, is due to the possibility of quickly modifying the operation of the algorithm by reprogramming the PLIC using special computer aided design (CAD)

systems. The disadvantages of the implementation of such a computing device based on PLIC can be attributed to the limited number of logical elements.

**Approbation of work.** The results of the work have been tested or are being published at conferences:

1. XI Conference of Young Scientists "Applied Mathematics and Computerizing" PMK-2018-2;
2. IV International Scientific and Technical Internet - Conference "Modern Methods, Information, Software -Technical Support of Management Systems of Organizational, Technical and Technological Complexes».

**Publications.** The results of the work are published in 3 scientific papers, of which 3 are theses of the reports.

**Structure and scope of work.** *The master dissertation include: introduction, four chapters, conclusion and applications.*

In the introduction, given a general description of the work, completed a current state of the problem, justified the relevance of the research direction, formulated the purpose and objective of the research, shown the scientific novelty of the results obtained and the practical value of the work, given the information about testing of the results and their implementation.

The first section reviewed some general information about PLIC; reviewed alternative electronic components, such as microcontrollers, microprocessors, single – chip computers, also reviewed and analyzed based characteristics of objects of critical use.

In the second section, reviewed general information about PLIC, namely their classification and characteristics, reviewed the main families of PLIC, namely PLIC from leading companies Altera and Xilinx, reviewed classification and characteristics of the main families of PLICs by Xilinx. Determined the structural features each of the families.



In the third section, reviewed, analyzed and reasonably the methods and types of organization of data processing on the PLIC; considered the basic design tools of the company XILINX; formulated requirements for CAD for designing a device for parallel processing of data on PLIC.

The fourth section describes the steps of designing a based PLIC parallel data processing device for monitoring critical objects; proposed parallel data processing algorithm; proposed software and hardware implementation of based PLIC parallel data processing algorithm for monitoring critical objects.

The conclusion analyzed the results of the work. Determined the conditions for which the implementation is developed. Also proposed ways to improve the developed method and recommendations for further study of the task.

The appendices provide a block diagram of the architecture of the PLIC family, an internal CLB interconnection diagram, the architecture of the Xilinx Spartan-6 structural module, as well as a listing of programs and copies of publications.

The work was done on 78 pages, contains 4 appendices and references to the list of references used from 26 titles. The work contains 33 figures and 3 tables.

**Keywords:** parallel processing device for FPGA, critical application object, Xilinx, FPGA, Spartan-3, CAD, WEBPACK ISE, Verilog, VHDL.

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	3
ВСТУП .....	4
РОЗДІЛ 1 АНАЛІЗ ІСНЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ ТА АЛЬТЕРНАТИВИ ВИКОРИСТАННЯ ПЛІС .....	6
1.1 Огляд існуючих альтернатив ПЛІС .....	7
1.1.1 Мікроконтролери .....	7
1.1.2 Мікропроцесори .....	10
1.1.3 Однокристальні мікрокомп'ютери .....	12
1.2 Характеристика об'єктів критичного застосування .....	13
1.3 Висновки до першого розділу .....	17
РОЗДІЛ 2 ХАРАКТЕРИСТИКА ПРОГРАМОВНО ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ .....	19
2.1 Загальна інформація про ПЛІС .....	19
2.2 Класифікація ПЛІС .....	20
2.3 Характеристика основних сімейств ПЛІС .....	26
2.4 Характеристика основних сімейств ПЛІС FPGA фірми Altera .....	28
2.5 Опис основних ПЛІС сімейства FPGA фірми Xilinx .....	29
2.5.1 Характеристика FPGA Spartan-6 компанії Xilinx .....	34
2.5.2 Характеристика FPGA Spartan-7 компанії Xilinx .....	39
2.6 Висновки до другого розділу .....	43
РОЗДІЛ 3 ЗАСОБИ ТА МЕТОДИ ОБРОБКИ ДАНИХ НА ПЛІС .....	45
3.1 Обробка даних на ПЛІС .....	45
3.2 Огляд основних засобів проектування на ПЛІС .....	46
3.2.1 Огляд пакету проектування Vivado HLS .....	47
3.2.2 Огляд пакету проектування ISE Design Suite .....	49
3.3 Способи опису апаратури .....	52
3.4 Висновки до третього розділу .....	54
РОЗДІЛ 4 РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ НА ПЛІС .....	55
4.1 Етапи проектування цифрових пристроїв на базі ПЛІС .....	55

4.2	Етапи створення проекту за допомогою WEB Pack ISE.....	57
4.3	Опис реалізації обробки даних на ПЛІС для моніторингу об'єктів критичного застосування .....	70
4.4	Висновки до четвертого розділу .....	76
	ВИСНОВКИ.....	78
	СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	79
	ДОДАТКИ.....	82

Додаток 1. Лістинг VHDL коду

Додаток 2. Лістинг C# коду

Додаток 3. Публікації

Додаток 4. Презентація

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

АЛП – Арифметично-логічний пристрій  
АЦП – Аналого-цифровий перетворювач  
ВІС – Велика інтегральна схема  
МК – Мікроконтролер  
ПЛІС – Програмовна логічна інтегральна схема  
САПР – Система автоматизованого проектування  
ADC – Analog-to-digital converter  
ALU – Arithmetic and logic unit  
ASIC – Application specific integrated circuit  
CLB – Configurable logic block  
CPLD – Complex programmable logic device  
DCM – Digital clock manager  
DSP – Digital signal processing  
FLEX – Flexible logic element matrix  
FPGA – Field programmable gate array  
JTAG – Joint test action group  
IOE – Input/output element  
ISE – Integrated synthesis environment  
LAB - Logic array block  
LUT – Look out table  
PAL - Programmable array logic  
PLC – Programmable logic device  
PLL – Phase locked loops  
RAM – Random access memory  
ROM – Read only memory  
SOC - System on chip  
SRAM – Static random access memory  
VHDL – Vivado high level synthesis

## ВСТУП

Технічний прогрес у минулому столітті відбувався дуже швидко і розвиток електроніки не стояв на місці. З кожним роком електронна техніка стає все більш функціональною і компактною. Тому, у сфері автоматичної та електроніки постійно підвищуються вимоги до основних характеристик виробу, таких як надійність, швидкодія, загальна вартість виготовлення, розміри, потужність споживання тощо. Найоптимальніша комбінація цих характеристик на сьогоднішній день є головною метою розробки у сфері електроніки.

На сьогодні є велика кількість галузей виробництва, де необхідно забезпечувати високий рівень надійності та безпеки використання об'єктів критичного застосування. Це такі сфери підприємства як:

5. енергетична галузь, а саме атомні електростанції, теплоелектростанції, гідроелектростанції;
6. підприємства із небезпечним виробництвом, які мають велику ймовірність техногенних катастроф;
7. космічна галузь;
8. транспортна галузь.

Об'єкти критичного застосування широко використовуються у багатьох сферах промислової діяльності людства таких як енергетика, електроніка тощо. З кожним роком промисловість збільшується і ускладнюється, що потребує постійного удосконалення комп'ютерних технологій, які застосовуються в інформаційних системах моніторингу об'єктів критичного застосування, які направлено на забезпечення функціональної безпеки об'єктів. Для оперативної оцінки функціональної безпеки цифрових компонентів системи критичного застосування використовуються методи і засоби робочого діагностування, які контролюють достовірність отриманих числових результатів.

Безпека та надійність використання таких об'єктів критичного застосування залежить від рівня автоматизованості певної промисловості. Рівень автоматизованості залежить від таких показників, як безвідмовність роботи обчислювальних систем, довершеність архітектурних та програмних рішень, відмовостійкість програмно-апаратних частин, час ремонту тощо.

"Для створення цифрових інтегральних схем використовуються різні електронні компоненти, такі як програмований логічний контролер, базові матричні кристали, спеціалізовані замовні великі інтегральні схеми (ВІС), мікроконтролери тощо. Досить гарною альтернативою такого компоненту є програмована логічна інтегральна схема (ПЛІС). ПЛІС – електронний компонент, що використовується для створення цифрових інтегральних схем. На відміну від звичайних цифрових інтегральних схем, логіка роботи ПЛІС не визначається при виготовленні, а задається за допомогою програмування. Для цього використовують програматори і налагоджувальні середовища, що дозволяють задати бажану структуру цифрового пристрою у вигляді принципової схеми або програми на спеціальних системах автоматизованого проектування (САПР)" [1]. Сучасні ПЛІС характеризуються властивістю багаторазового перепрограмування, низькою вартістю виготовлення, досить високою швидкістю, малою енергією споживання тощо.

Сучасні ПЛІС можна використовувати для вирішення багатьох математичних завдань в учбових цілях, адже сучасні САПР дозволяють швидко перепрограмувати схему. На сьогоднішній день існує велика кількість САПР, розроблених провідними компаніями у сфері електроніки.

При створенні програмно-апаратної реалізації моніторингу об'єкту критичного застосування можна виділити такі основні вимоги, як: простота технічної реалізації; простота апаратної і програмної реалізації методів і алгоритмів виконання необхідних операцій; виконання умови практичності та економності розробленої програмно-апаратної реалізації.

## **РОЗДІЛ 1 АНАЛІЗ ІСНЮЧИХ РІШЕНЬ ПОСТАВЛЕНОЇ ЗАДАЧІ ТА АЛЬТЕРНАТИВИ ВИКОРИСТАННЯ ПЛІС**

На сьогоднішній день існує велика кількість рішень задачі паралельної обробки даних на основі електронних компонентів, таких як:

- 1) базові матричні кристали. Але вони потребують заводського виробничого процесу для програмування;
- 2) логічний контролер, що програмується;
- 3) ВІС для окремих певно визначених задач, виготовлення яких при малій кількості досить дорого;
- 4) мікроконтролери, реалізація алгоритмів на яких відбувається програмним шляхом.

Використання ПЛІС є доречним при паралельній обробці даних в наукових, промислових чи учбових цілях. Найбільші компанії із виготовлення та підтримки ПЛІС такі, як Altera, Achronix та Xilinx мають велику кількість вже готових рішень обробки даних.

На сьогодні розповсюджені такі високопродуктивні обчислювальні систем, які мають статичну структуру обчислювального середовища. Даний фактор не дозволяє ефективно використовувати та реалізовувати складні алгоритми обробки даних, оскільки структура таких алгоритмів може не повністю відповідати архітектурі комп'ютерної системи.

Існує багато напрямків підвищення ефективності роботи алгоритмів, які реалізують задачі із великою кількістю інформаційних зв'язків. Одним з таких напрямків є проектування та побудова паралельних обчислювальних систем, які мають можливість реконфігурації каналів зв'язків між процесорами. Наприклад, система Non-Uniform Memory Access, мультипроцесори, трансп'ютери тощо.

Використання ПЛІС для побудови реконфігурованих комп'ютерних систем є актуальним та доречним рішенням для підвищення ефективності паралельної обробки даних. Така концепція базується на програмно-

апаратній реалізації поставленої задачі обробки даних, яка ґрунтується на пришвидщенні певних частин алгоритму, оскільки їх виконання є недоречним за допомогою простих процесорів. Використання ПЛІС дозволить реалізувати на одному кристалі мікросхеми систему для обчислень із гнучкою структурою. Оскільки ПЛІС має можливість швидкого перепрограмування та перебудови обчислювальної структури, то це дозволяє реалізовувати довільні алгоритми обробки даних. Такий програмно-апаратний підхід дозволить забезпечити ефективне та просте користування, підтримку та реалізацію методів паралельного програмування, широку функціональність виготовленого пристрою та економічну привабливість [2].

## **1.1 Огляд існуючих альтернатив ПЛІС**

### **1.1.1 Мікроконтролери**

Мікроконтролер (далі МК) зовнішньо має вигляд мікросхеми, яка представляє собою унікальну мікропроцесорну систему. Складові елементи МК:

- 1) мікропроцесор;
- 2) блоки пам'яті;
- 3) порти вводу/виводу;
- 4) інші спеціальні функціональні блоки (лічильники, компаратори тощо).

Призначення МК - забезпечити управління електронними пристроями, створити системи контролю, збору та архівації інформації.

МК зменшує розміри пристроїв, підвищує їх функціонал, оптимізує енергоспоживання пристроїв, та в цілому сприяє здешевленню пристроїв. Основні сфери використання МК: мобільні телефони, сучасна побутова техніка, сучасні автомобілі [3].

При розробці та проектуванні МК враховуються такі характеристики, як: гнучкість структури, продуктивність а також розмір та вартість вибору.



Для різних типів задач кореляція між цими параметрами може доволі сильно відрізнятись. Тому існує велика кількість МК, які мають різні параметри пам'яті, архітектури процесорного модуля, кількість та набір периферійних компонентів, типи корпусу.

На сьогоднішній день виробники забезпечують високочастотну роботу своїх виробів, також вони виготовляють різні модифікації для потреб замовника. Адже збільшення тактової частоти у контролерів знижують обмеження у ціні, потужності споживання та інших характеристиках. Доволі велика кількість типів МК для оперативного запам'ятовуючого пристрою та внутрішніх регістрів пам'яті використовують статичну пам'ять. Це дозволяє МК працювати на значно меншій частоті. Також при повній зупинці тактового генератора не відбувається повної втрати даних. Для різних типів МК реалізовані різноманітні режими енергозбереження, які забезпечують відключення різних модулів, коли потрібно. Також МК може містити вбудовану пам'ять, яка є незалежною, і яку можна використовувати для зберігання даних або програм. Оскільки у великій кількості МК відсутня шина для підключення зовнішньою пам'яті, то такі пристрої є ефективними при масштабному виробництві для тих випадків, коли програма контролера не буде змінюватися. Також існують модифікації МК, які дозволяють багаторазовий запис до незалежної пам'яті.

За останні десятиліття МК має тенденцію швидкого розвитку, вони стають все більш потужнішими та ефективнішими. Восьми-розрядні мікроконтролери мають широке застосування у різних сферах людської діяльності, а саме науковій, побутовій, промисловій та транспортній сферах. Для реалізації складних алгоритмів у яких виконується велика кількість логічних операцій, використання восьми розрядних мікроконтролерів є доречним та продуктивним рішенням [4].

Всесвітні провідні компанії, такі як Intel, Altmel, Motorola та інші, виробляють та підтримують 8-ми розрядні мікроконтролери.

Основними характеристиками останніх версій 8-ми розрядних мікроконтролерів є:

1) спеціальна організація модулів та їх взаємодії, при розробці декількох МК на основі одного ядра ЦП. Ці МК мають різні технічні характеристики: об'єми пам'яті, частоту синхронізації або множину периферійних модулів;

2) закрита архітектура МК (рис. 1.1). Це забезпечує відсутність на виході корпусу МК ліній магистралей адрес та даних. В такому разі МК є закінченою системою для опрацювання даних з відсутністю можливістю збільшення обробки даних за допомогою паралельних магистралей;

3) використання стандартних периферійних модулів в залежності від компанії виробника МК. Це відображається в їх алгоритмах роботи. Такими МК є аналого-цифрові перетворювачі, контролер послідовних інтерфейсів, таймери та інші;

4) встановлення багатьох режимів роботи периферійних пристроїв на у процесі початкового налаштування регістрів, які виконують спеціальні функції.

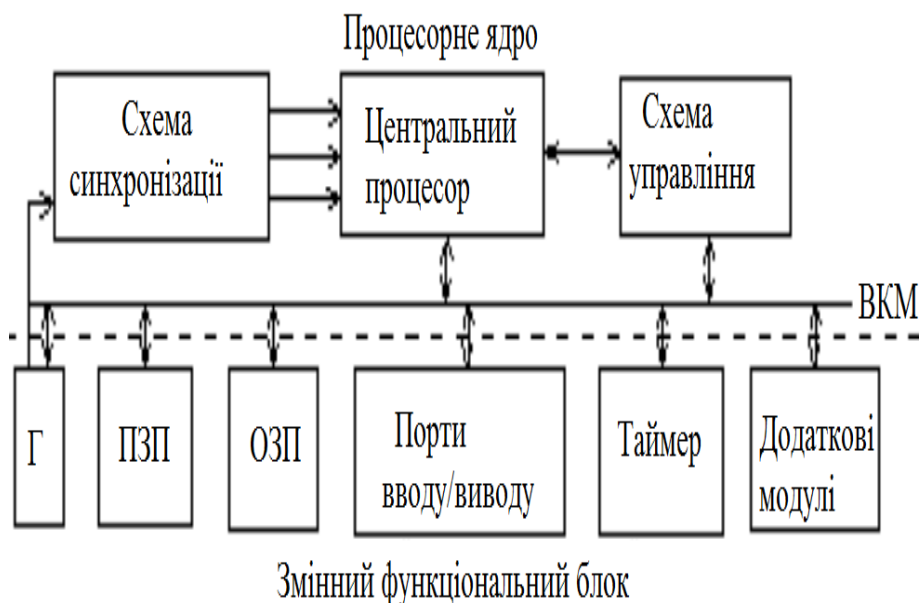


Рисунок 1.1 – Модульна організація МК

### 1.1.2 Мікропроцесори

На теперішній час мікропроцесори – найскладніші у виробництві електронні пристрої. Для їх створення потрібні сотні виробничих етапів. До чистоти та точності кожного етапу висуваються найжорстокіші вимоги. Мікропроцесор – це спеціальна інтегральна схема, яка виконує безліч функцій. Прикладом може бути функція спеціалізованого або центрального процесора. Сучасний мікропроцесор - це закінчена система управління. Його призначенням є управління роботою всіх інших вузлів та здійснення логічних та арифметичних операцій над будь якою інформацією. Мікропроцесор складається з величезної кількості мікротранзисторів, які пов'язані між собою супер тонкими алюмінієвими сполучними каналами. На протязі запису та обробки інформації канали забезпечують взаємодію між мікротранзисторами. Це дозволяє процесору здійснювати безліч різних функцій. Мікропроцесори вміщують складну внутрішню архітектуру та виглядають як надвелика інтегральна схема, що формується на шарах кремнієвої підкладки. Це отримують шляхом використання спеціальних процесів формування схеми за допомогою випромінювання, із застосуванням хімічних препаратів та газів [5].

Стандартні мікропроцесори можуть виконувати такі процеси, як (рис. 1.2):

- 1) робота з основною пам'яттю, а саме зчитування та дешифрація команд, які знаходяться в основній пам'яті. Опрацювання даних та їх подальший опис до головної пам'яті чи регістрів;
- 2) робота із адаптерами зовнішніх пристроїв, а саме зчитування даних із головної пам'яті або регістрів;
- 3) генерування керуючих послідовностей, для управління при роботі із вихідними пристроями.



Рисунок 1.2 – Схема роботи мікропроцесора

До складу стандартного мікропроцесора входять такі наступні пристрої (рис. 1.3), як:

1) арифметико-логічний пристрій (АЛП). АЛП потрібний для того, щоб виконувати логічні та арифметичні дії на інформацію, яка може бути представлена у чисельній чи просто текстовій формі;

2) пам'ять мікропроцесора. Така пам'ять потрібна для того, щоб зберігати, записувати чи віддавати тимчасову інформацію. Тимчасова інформація задіється в обчисленнях при найближчих тактах роботи пристрою. Принцип побудови пам'яті мікропроцесора базується на регістрах. Завдяки пам'яті мікропроцесора забезпечується більш висока швидкість обчислювального пристрою, оскільки основна пам'ять не може завжди гарантувати потрібну швидкість обробки інформації, (а саме процес зчитування, опрацювання та віддачі даних), яка потрібна для більш продуктивної роботи мікропроцесора;

3) пристрій управління. Даний пристрій забезпечує взаємодію усіх частин комп'ютера. Пристрій забезпечує виконання таких операцій, як: створення та відправка певних керуючих сигналів для різних блоків обчислювального пристрою, специфіка яких залежить від виконання певної операції; передача потрібних адрес (комірок пам'яті) у потрібні блоки комп'ютера; одержання зворотної послідовності сигналів від генератора тактових імпульсів.

4) за взаємодію із іншими пристроями відповідає так звана інтерфейсна частина. Вона містить такі частини, як буферні запам'ятовуючі пристрої; схему внутрішнього інтерфейсу мікропроцесора; схему керування портами вводу/виводу; схему управління системною шиною.

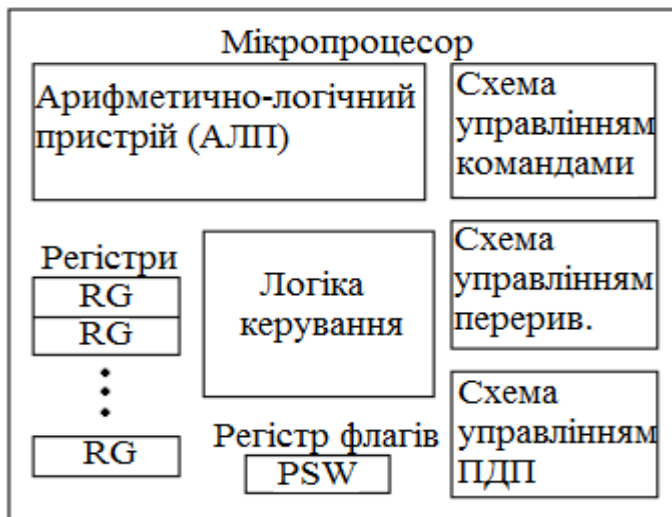


Рисунок 1.3 – Структура найпростішого мікропроцесора

Мікропроцесор має такі важливі характеристики як:

1) розрядність процесора. Цей параметр означає, який максимальний розмір (кількість розрядів) числа у двійковому вигляді може опрацьовуватися однією машинною операцією. Отже, чим більший показник розрядності процесора, тим більше даних він може опрацьовувати за одну одиницю часу. Даний показник також характеризує ступень продуктивності комп'ютера;

2) тактова частота (вимірюється у МГц). Процесор відводить деяку кількість тактів для виконання певних операцій. Цей показник характеризує кількість елементарних операцій, які мікропроцесор виконує за одну секунду часу [6].

### 1.1.3 Однокристальні мікрокомп'ютери

Одним із різновидів альтернативи використанню ПЛІС є однокристальні мікрокомп'ютери. Стандартні однокристальні комп'ютери містять пристрої/елементи:

1) процесор;

- 2) блок пам'яті (для команд та даних);
- 3) тактовий генератор.

Цих пристроїв достатньо для мінімальної конфігурації системи.

Переваги однокристальних комп'ютерів: надійність, компактність, низька собівартість. Це надає переваги для використання однокристальних комп'ютерів у малогабаритній апаратурі [7].

## **1.2 Характеристика об'єктів критичного застосування**

Такі галузі промисловості, як хімічна, енергетична, транспортна, фінансова, інформаційна, охорона здоров'я тощо є важливими для функціонування важливих сфер промислового розвитку держави. Вихід з ладу, помилки при роботі таких об'єктів може мати серйозний економічний вплив, небезпеку для життя людей тощо. Такі об'єкти називають об'єктами критичного застосування.

На сьогодні є велика кількість галузей виробництва, де необхідно забезпечувати високий рівень надійності та безпеки використання об'єктів критичного застосування. Це такі сфери підприємства як:

- 1) хімічні підприємства із небезпечним виробництвом, які мають велику ймовірність техногенних катастроф;
- 2) енергетична галузь, а атомні електростанції, теплоелектростанції, гідроелектростанції;
- 3) космічна галузь;
- 4) транспортна галузь.

У сучасному хімічному виробництві використовуються, переробляються, виготовляються, зберігаються або транспортуються багато небезпечних речовин у таких кількостях, що будь-які аварії чи несправності є реальною загрозою виникнення техногенних катастроф.

"Екологічна громадська організація «Жива планета» опублікувала топ-5 найбільш небезпечних виробництв в Україні:

- 1) калуський калійний завод «Оріана», що на Прикарпатті. За словами захисників природи, якщо отруйні відходи з цього заводу потраплять у Дністер, то це буде катастрофа не менша, ніж в Угорщині;
- 2) каховінська паперова фабрика у Львівській області. Якщо її відходи потраплять у річку Стрий, то отруять воду в усьому регіоні;
- 3) металургійний комбінат «Азовсталь», який викидає відходи від переробки нафтопродуктів прямо в Азовське море;
- 4) комунальне підприємство «Городводхоз» Кременця Тернопільської області, яке зливає каналізацію у місцеву річку;
- 5) київський завод «Радикал». У центрі столиці зберігається така кількість ртуті, яку якщо розділити її на жителів усієї планети, кожному дістанеться по два термометри" [8].

Такі потужні галузі промисловості як атомні електростанції, теплоелектростанції, гідроелектростанції потребують особливого контролю їх роботи. Аварія на Чорнобильській АЕС показала, що такий вид електростанцій є особливо екологічно небезпечним. Аварії на таких об'єктах мають дуже важкі наслідки як для економіки країни, так і для життя та здоров'я людей, тварин та рослин, всього навколишнього середовища. Але й при штатній роботі таких атомних станцій утворюються специфічні складові, які забруднюють воду для охолодження реакторів.

Атомні електростанції можуть забруднювати навколишнє середовище не тільки радіоактивними відходами, але й відходами комунального господарства, нафто-продуктами тощо.

Чорнобильська аварія спричинила екологічну катастрофу глобального масштабу:

- 1) сталося забруднення 4,6 мільйонів гектарів земель, розташованих у 11 областях України;

2) признані не придатними для подальшого оброблення 119 тис. га угідь сільськогосподарського призначення, в тому числі 65 тис. гектарів ріллі;

3) забруднено радіоактивними викидами майже 4 мільйони гектарів лісів.

Мали місце як економічні, так і соціальні наслідки, в ліквідації яких були задіяні всі республіки колишнього СРСР. Ліквідувати таку аварію Україні самостійно було неможливо. Тому, таких катастроф варто не допускати в майбутньому [9].

Для зниження збитків при аварії на атомних станціях можна розмішувати біля АЕС водосховища. Водосховища можна використовувати як акумулятори забруднення при аваріях, як це відбувалося на Київському водосховищі під час аварії 1986 року.

Заводи та фабрики підприємств електроенергетичної сфери забруднюють повітря різними шкідливим сполуками, такими як сірчанам ангідритом, окисами вуглецю, азотом, пилом і сажею. Стічні води цих підприємств забруднюються нафтопродуктами.

Нафтова промисловість має на території нашої країни розгалужену систему нафто та газо проводів. Більшість з них було побудовано близько 50 років тому і на сьогодні вони знаходиться у не дуже хорошому стані. Щороку фіксуються аварії на цих об'єктах. Це призводить до викидів до навколишнього середовища нафто та газо продуктів, що завдає шкоди водному, сільському господарствам тощо. Тому експлуатація трубопроводів потребує ретельного контролю з боку служб охорони навколишнього середовища.

Підприємства у сфері металургійної промисловості забруднюють атмосферу викидами різних газів, таких як окиси азоту, вуглецю, пилу, сірки тощо. Вода, яка на виробництві використовуються для охолодження, також забруднюється шкідливими речовинами, а саме нафтопродуктами,



хлоридами, сульфатами тощо. Крім того, на таких підприємствах швидко накопичуються відходи виробництва, які потрібно вчасно утилізувати.

При створенні таких підприємств також потрібно передбачувати аварійні ситуації, які можуть завдати матеріальних чи людських збитків.

Отже, на сьогоднішньому етапі розвитку промисловості зростання та вдосконалення виробництва тісно пов'язане з об'єктами критичного застосування, які тісно пов'язані з ймовірністю виникнення техногенних катастроф. Чинниками техногенних катастроф можуть бути як кількість великих міст, шкідливих підприємств, транспортних засобів, наявність шляхів сполучення тощо. Під впливом швидкого розвитку підприємств та методів виробництва ці техногенні чинники мають дію прямого ефекту на навколишнє середовище. Тому в цих умовах є важливим завдання оптимізації техногенного навантаження із правильним урахуванням всіх техногенних чинників на всіх стадіях виробництва для запобігання різних аварій та катастроф.

Для того щоб запобігати проблем, несправностей, аварій чи катастроф у результаті роботи підприємств, які використовують об'єкти критичного застосування потрібно дотримуватися таких принципів, як:

- 1) профілактика та запобігання проблем та несправностей під час процесу виробництва, які можуть задати матеріальних чи інших збитків;
- 2) моніторинг та аналіз збитків виробничої діяльності у результаті непередбачуваних ситуацій.

Додержання вище описаних принципів дозволить знизити ризик використання об'єктів критичного застосування. Але, оскільки розвиток людської діяльності відбувається великими темпами, то задача підвищення ефективності автоматизації моніторингу за об'єктами критичного застосування є актуальною на сьогоднішній день.

Безпека та надійність використання таких об'єктів критичного застосування залежить від рівня автоматизованості певної промисловості.

Рівень автоматизованості залежить від таких показників, як безвідмовність роботи обчислювальних систем, довершеність архітектурних та програмних рішень, відмовостійкість програмно-апаратних частин, час ремонту тощо.

### **1.3 Висновки до першого розділу**

На сьогодні є велика кількість галузей виробництва, де необхідно забезпечувати високий рівень надійності та безпеки використання об'єктів критичного застосування. Це такі сфери підприємства як хімічна, енергетична, транспортна, фінансова, інформаційна, охорона здоров'я тощо є важливими для функціонування важливих сфер промислового розвитку держави. Вихід з ладу, помилки при роботі таких об'єктів може мати серйозний економічний вплив, небезпеку для життя людей тощо. Такі об'єкти називають об'єктами критичного застосування.

Безпека та надійність використання таких об'єктів критичного застосування залежить від рівня автоматизованості певної промисловості. Рівень автоматизованості залежить від таких показників, як безвідмовність роботи обчислювальних систем, довершеність архітектурних та програмних рішень, відмовостійкість програмно-апаратних частин, час ремонту тощо.

На сьогодні розповсюджені такі високопродуктивних обчислювальні систем, які мають статичну структуру обчислювального середовища. Даний фактор не дозволяє ефективно використовувати та реалізовувати складні алгоритми обробки даних, оскільки структура таких алгоритмів може не повністю відповідати архітектурі комп'ютерної системи.

Використання ПЛІС є доречним при паралельній обробці даних в наукових, промислових чи учбових цілях. Найбільші компанії із виготовлення та підтримки ПЛІС такі, як Altera, Achronix та Xilinx мають велику кількість вже готових рішень обробки даних.

На сьогоднішній день існує велика кількість рішень задачі паралельної обробки даних на основі електронних компонентів, таких як базові матричні

кристали, але вони потребують заводського виробничого процесу для програмування; логічний контролер, що програмується; ВІС для окремих певно визначених задач, виготовлення яких при малій кількості досить дорого; мікроконтролери, реалізація алгоритмів на яких відбувається програмним шляхом.

Використання ПЛІС для побудови реконфігурованих комп'ютерних систем є актуальним та доречним рішенням для підвищення ефективності паралельної обробки даних. Така концепція базується на програмно-апаратній реалізації поставленої задачі обробки даних, яка ґрунтується на пришвидшенні певних частин алгоритму, оскільки їх виконання є недоречним за допомогою простих процесорів. Оскільки ПЛІС має можливість швидкого перепрограмування та перебудови обчислювальної структури, то це дозволяє реалізовувати довільні алгоритми обробки даних. Такий програмно-апаратний підхід дозволить забезпечити ефективне та просте користування, підтримку та реалізацію методів паралельного програмування, широку функціональність виготовленого пристрою та економічну привабливість.

## РОЗДІЛ 2 ХАРАКТЕРИСТИКА ПРОГРАМОВНО ЛОГІЧНИХ ІНТЕГРАЛЬНИХ СХЕМ

### 2.1 Загальна інформація про ПЛІС

"ПЛІС – це електронний компонент, що використовується для створення цифрових інтегральних схем. На відміну від звичайних цифрових мікросхем, логіка роботи ПЛІС не визначається при виготовленні, а задається за допомогою програмування. Для програмування використовуються програматори і налагоджувальні середовища, що дозволяють задати бажану структуру цифрового пристрою у вигляді принципової електричної схеми або програми на спеціальних мовах опису апаратури Verilog, VHDL, AHDL та ін. Альтернативою ПЛІС є: програмований логічний контролер, базові матричні кристали, що вимагають заводського виробничого процесу для програмування; Application Specific Integrated Circuit (ASIC) — спеціалізовані замовні ВІС (великі інтегральні схеми), які при багатосерійному та одиничному виробництві істотно дорожче; спеціалізовані комп'ютери, процесори (наприклад, цифровий сигнальний процесор) або мікроконтролери, які через програмний спосіб реалізації алгоритмів повільніші, ніж ПЛІС. Провідні виробники ПЛІС для спрощення проектування ПЛІС пропонують спеціальні програмні процесори, які можна модифікувати під певну задачу. Потім вони можуть бути вбудовані в ПЛІС, що забезпечує зменшення місця на друкованій платі" [10].

На сьогоднішній день ПЛІС є однією із найбільш перспективних та використовуваних елементних баз для побудови спеціалізованих цифрових пристроїв різного призначення. Перевагами ПЛІС є такі властивості, як:

- 1) універсальність та високий попит;
- 2) масове виробництво;
- 3) низька вартість;
- 4) висока швидкодія та надійність;

- 5) різноманітність у виборі напруги живлення та параметрів сигналів введення/виведення;
- 6) підтримка зручних та ефективних програмних засобів автоматизованого проектування від провідних компаній;
- 7) швидкий час проектування;
- 8) простота та зручність модифікації проектів на будь-яких етапах розробки.

## 2.2 Класифікація ПЛІС

ПЛІС класифікують за ознаками ознаками:

- 1) за рівнем інтеграції та логічної складності;
- 2) за архітектурою (тип блоків, з'єднання тощо);
- 3) за кількістю етапів програмування;
- 4) за типом пам'яті;
- 5) за ступенем залежності затримок сигналів від методів та засобів їх розширення;
- 6) за системними характеристиками;
- 7) за технологією виробництва;
- 8) за гібридністю (однорідністю) [11].

ПЛІС поділяється на такі основні сімейства:

- 1) Complex Programmable Logic Device (CPLD);
- 2) Field Programmable Gate Array (FPGA);
- 3) Flexible Logic Element Matrix (FLEX);
- 4) System On Chip (SOC).

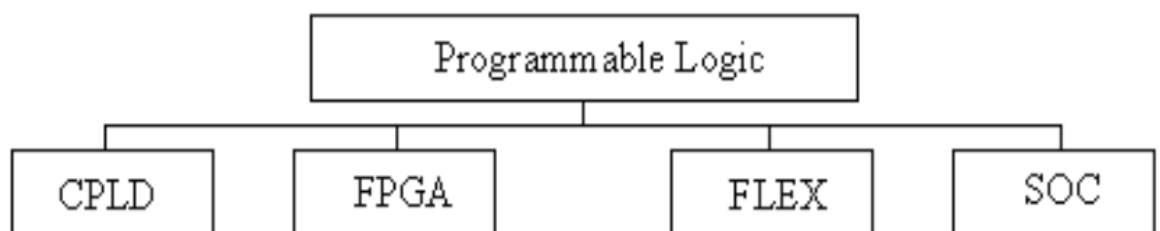


Рисунок 2.1 – Класифікація ПЛІС

Комплексний програмований логічний пристрій (CPLD) - це комбінація повністю програмованої множини І/АБО елементів та мікро комірок. Множина І/АБО- може бути перепрограмована та може виконувати безліч логічних функцій. Мікро комірки - це функціональні блоки, які виконують комбінаторну або послідовну логіку, а також мають додаткову гнучкість для визначення істинності чи доповнення, а також різноманітні шляхи зворотного зв'язку.

Традиційно, CPLD використовують аналогові підсилювачі для підвищення продуктивності своєї архітектури. Це підвищення продуктивності було реалізовано за рахунок дуже високих вимог до пристроїв, що розробляються на основі CPLD. CPLDs CoolRunner-II, створені компанією Xilinx, використовують інноваційне цифрове ядро для досягнення тих самих рівнів продуктивності при низькій потужності споживання. Це дозволяє розробникам використовувати одну і ту саму архітектуру CPLD при проектуванні як для високопродуктивних, так і для малопотужних цифрових пристроїв. Але й відсутність аналогових підсилювачів також забезпечує масштабованість архітектури, що дозволяє значно знижувати витрати та покращувати логічні функції з кожним наступним процесом генерації.

Сьогодні CPLD зберігають структуру логіки без живлення (після вимкнення) та є перепрограмованими (архітектуру CPLD див на рис. 2.2). Ця архітектура зручна для втілення цифрових пристроїв, тому що можна легко реалізувати ті функції, які задано у вигляді диз'юнктивної форми. Це має велике значення при оптимізації тих складних схем, що складені на звичайних логічних елементах [12].

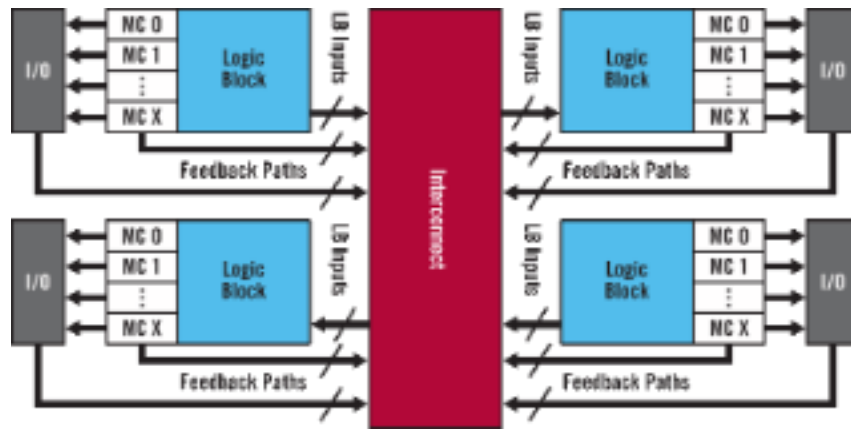


Рисунок 2.2 –Архітектура ПЛІС сімейства CPLD

Програмована користувачем вентильна матриця (FPGA) вперше були представлені майже тридцять років тому. З тих пір дана технологія почала швидко зростання і стала популярним засобом для створення цифрових схем. Прогрес технологій значно розширив логічні можливості FPGA і, у свою чергу, зробив їх життєздатними альтернативами рішення великих і складних реалізацій. Можливість перепрограмування їхньої логіки та ресурсів маршрутизації суттєво впливає на кінцеві розміри розробленого цифрового пристрою, на його швидкодію та потужність споживання.

FPGA складається з матриці програмованих логічних блоків, які з'єднані між собою через програмовані мережі з'єднання. Програмована логіка та маршрутизація з'єднань при використанні FPGA робить їх гнучкими та ефективними у використанні, але в той же час це робить їх більшими, повільнішими, вони потребують більше потужності споживання, що перевищує стандартну комірку ASIC. Проте провідні компанії займаються удосконаленням FPGA, що дозволяє подальше вдосконалення загальної ефективності FPGA.

Для малих та середніх задач, FPGA є дешевим рішенням, забезпечує швидший час виходу на ринок порівняно з спеціальними інтегральними схемами додатків (ASIC), які зазвичай потребують багато ресурсів з точки зору часу та грошей для проектування для створення цифрового пристрою. Також, у залежності від вимог та потреб, частина логіки ПЛІС може бути

частково переналаштована, а решта логіки FPGA працювати так само. Будь-які майбутні оновлення в кінцевому продукті можна легко оновити, просто завантаживши новий бітовий потік програми. Проте основна перевага ПЛІС, тобто гнучкість та можливість перепрограмування, також є основною причиною його програші у ефективності порівняно з ASIC. Гнучкість при використанні ПЛІС робить їх значно більшими у розмірах, повільними, вони споживають більше енергії, ніж їх аналог ASIC. Ці недоліки виникають в основному через програмований маршрутизаційний зв'язок FPGA, який складає майже 90% загальної площі FPGA. Але, не зважаючи на ці недоліки, FPGA представляє переконливу альтернативу впровадженню цифрових систем через їх менший час для виходу на ринок та низьку вартість.

Зазвичай FPGA складається з:

- 1) програмовані логічні блоки, які реалізують логічні функції;
- 2) програмована маршрутизація, яка з'єднує ці логічні функції;
- 3) блоки вводу-виводу, які підключені до логічних блоків через маршрутизацію міжмережевого з'єднання, і які з'єднують мікросхеми.

Узагальнений приклад FPGA показаний на рис. 2.3, де налаштовується логіка. Блоки Configurable Logic Block (CLB) розташовані у вигляді двовимірної сітки і взаємопов'язані із програмованими ресурсами маршрутизації. Блоки вводу/виводу розташовані на периферії сітки, і вони також підключені до програмованого з'єднання маршрутизації. Термін «програмований» або «реконфігурований» для FPGA вказує на їх здатність реалізувати нову функцію на кристалі після завершення його виготовлення. Програмування FPGA починається з апаратного опису, синтезу, мапінгу та упакування, використовуючи різні інструменти Після цього розроблений опис розміщується на кристалі FPGA [13].



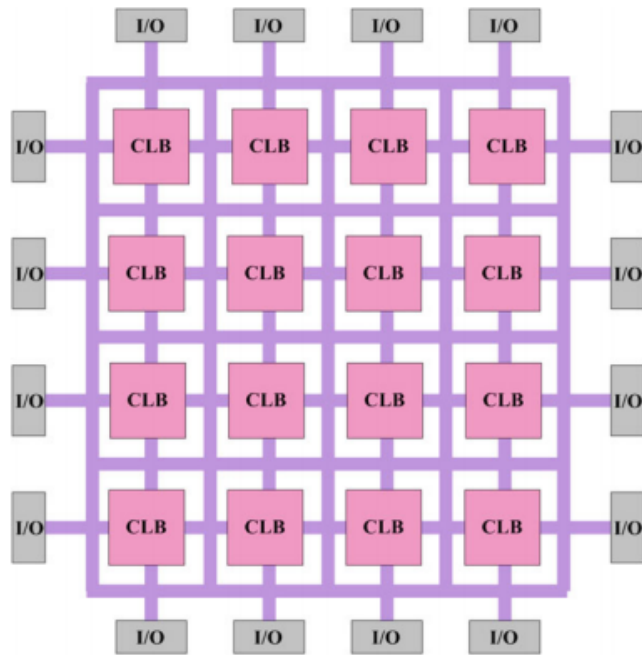


Рисунок 2.3 – Архітектура ПЛІС сімейства FPGA

Сімейство FLEX поєднує в собі переваги як програмованих логічних пристроїв Programmable Logic Device (PLD), так FPGA. ПЛІС сімейства FLEX ідеально підходить для різноманітних реалізацій пристроїв, оскільки FLEX поєднує в собі дрібнозернисту архітектуру та високі характеристики реєстру FPGA з високою швидкістю та передбачуваними затримками з'єднання між PLD. Логіка реалізована в логічних блоках, що включає компактні 4-вхідні пошукові таблиці Look-Out Table (LUT) та програмовані регістри. Висока продуктивність забезпечується швидкою, безпервною мережею маршрутизації ресурсів. Пристрої розроблені на FLEX використовуються для таких задач, як цифрова обробка сигналів Digital Signal Processing (DSP), широкомасштабні мережі передачі даних.

Архітектура FLEX включає в себе велику матрицю компактних вбудованих блоків, що називаються логічними елементами Logic Element (LE). Кожен LE містить 4-вхідний LUT, який забезпечує комбінаторну логіку та програмований регістр, який забезпечує послідовну логіку. Дрібнозерниста структура LE забезпечує високоефективну реалізацію логіку. Вісім LE згруповані разом, щоб сформувати блок логічного масиву Logic

Array Block (LAB). Кожна система LAB являє собою незалежну структуру з загальними входами, з'єднаннями та керуючими сигналами. Архітектура LAB забезпечує високу продуктивності пристрою та зручності при маршрутизації.

На рисунку 2.4 показана блок-схема архітектури FLEX. Кожна група з восьми LE об'єднується в LAB; LAB організована в рядки та стовпці. Піни вводу/виводу підтримуються елементами вводу-виводу Input/output element (IOE), розташованими на кінцях рядків і стовпчиків. Кожна IOE містить двонаправлений буфер введення-виведення та спеціальний буфер, який можна використовувати як вхідний або вихідний регістр [14].

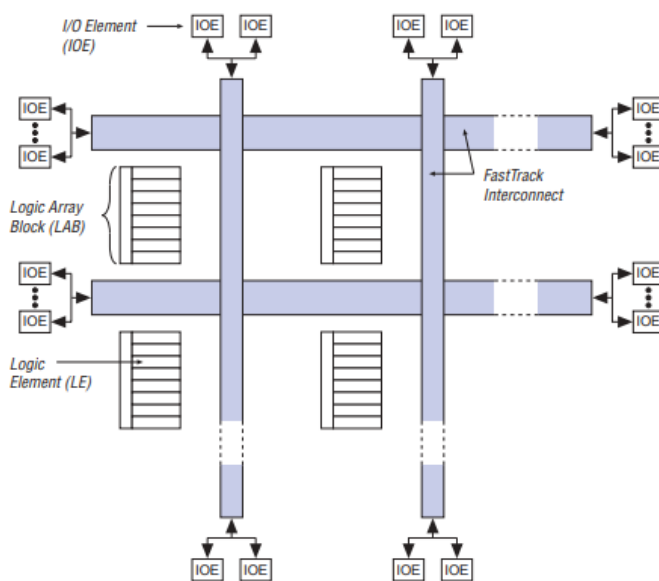


Рисунок 2.4 – Архітектура ПЛІС сімейства FLEX

"Подальший розвиток архітектури йде по шляху створення комбінованих архітектур, поєднують зручність реалізації алгоритмів ЦОС на базі LUT і реконфігурованих модулів пам'яті, характерних для FPGA-структур і багаторівневих ПЛІС із зручністю реалізації на CPLD-архітектурах. Так, ПЛІС АРЕХ20К фірми Altera містять в собі логічні елементи всіх перерахованих типів, що дозволяє застосовувати ПЛІС як основну елементарну базу для «систем на кристалі»" [15] (рис. 2.5).

Системи на кристалі широко використовуються у багатьох галузях промисловості для всіх цілей, таких як смартфони, планшети, цифрові носії, цифрові камери, бездротові маршрутизатори тощо. Найбільш поширеного

використання вони набули, як компоненти смартфонів і планшетів. Це невеликі пристрої, які потребують великої потужності для роботи, і всі вони повинні відповідати вимогам користувачів, які стають все більшими.

Стандартна SoC містить такі компоненти, як:

- 1) процесор (мікропроцесор) для обробки цифрових сигналів чи мікроконтролер;
- 2) блок пам'яті;
- 3) кварцеві генератори;
- 4) лічильники;
- 5) інтерфейси для периферійних пристроїв (USB, Ethernet тощо);
- 6) стабілізатори живлення;
- 7) регулятори напруги [16].

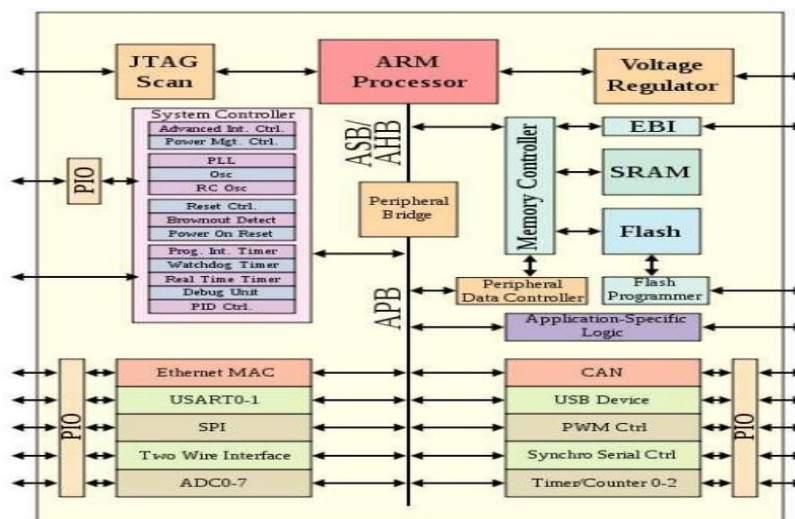


Рисунок 2.5 – Архітектура системи побудованої на кристалах

### 2.3 Характеристика основних сімейств ПЛІС

Сімейства ПЛІС FPGA і CPLD - це два найбільш поширені типи цифрових логічних мікросхем у використанні. ПЛІС сімейства CPLD є дещо проміжним рішенням між типами ПЛІС Programmable Array Logic (PAL) та FPGA. Оскільки сфера використання ПЛІС постійно удосконалюється, то відмінності між CPLD і FPGA не є сильно великими, тому на сьогодні є деяка

плутанина серед визначень ПЛІС до того, чи іншого сімейства. Але ці два сімейства мають свої структурні та параметричні особливості.

FPGA можна запрограмувати для здійснення практично будь-якої логічної функції. Архітектура FPGA дозволяє кристалу мати дуже високу логічну ємність. Він використовується в конструкціях, які вимагають великого числа логічних елементів. Але недоліком такої великої кількості логічних елементів є непередбачувані затримки. ПЛІС сімейства FPGA вважається «дрібною архітектурою», оскільки містить багато логічних блоків, кількість яких може досягати близько 100 000 одиниць. FPGA, як правило, використовується для створення та проектування більш складних цифрових пристроїв.

CPLD більш підходить для невеликих конструкцій проектів, що розробляються. Оскільки це менш складна архітектура, то затримки у роботі більш передбачуваним. CPLD часто використовується для простих логічних додатків. Він містить логічні блоки, кількість яких не перевищує кілька тисяч одиниць, тому CPLD розглядаються як пристрій типу «великої архітектури», що робить CPLD більш дешеві, ніж ПЛІС сімейства FPGA. Також важливою відмінністю між сімействами CPLD та FPGA є те, що до CPLD вбудовано енергонезалежну пам'ять.

Виділяючи всі основні особливості та недоліки сімейств ПЛІС FPGA та CPLD можна виділити такі основні відмінності:

- 1) FPGA містить до 100 000 логічних блоків, тоді як CPLD містить лише кілька тисяч логічних блоків;
- 2) з архітектурної точки зору, FPGA вважаються пристроями «дрібною архітектури», тоді як CPLD є пристроями «великої архітектури»;
- 3) FPGA є більш ефективними при проектуванні та реалізації складних цифрових пристроїв, тоді як ПЛІС сімейства CPLD краще для простих спеціалізованих цифрових пристроїв;
- 4) FPGA є більш дорогими, а CPLD - набагато дешевшими;

5) затримок під час роботи більше при використанні FPGA, ніж у CPLD.

Рішення про те, що використовувати, ПЛІС сімейства FPGA або CPLD, розробник повинен обирати залежно від цілей проектування та можливостей спеціалізованого цифрового пристрою, що розробляється [17][18].

#### **2.4 Характеристика основних сімейств ПЛІС FPGA фірми Altera**

FPGA фірми Altera, яку вже викупила компанія Intel, пропонує широкий спектр вбудованих блоків пам'яті Static Random Access Memory (SRAM), високошвидкісних трансиверів, високошвидкісних блоків входу/виходу, логічних блоків та блоків маршрутизації. Вбудована інтелектуальна власність у поєднанні з програмними засобами, які пропонує компанія Altera, зменшує час розробки спеціалізованих цифрових пристроїв, потужність споживання та вартість розробки на основі ПЛІС сімейства FPGA.

Фірма Altera пропонує такі сімейства ПЛІС типу FPGA:

1) серія FPGA Cyclone побудована таким чином, щоб задовольнити потреби низького енергоспоживання, потребують високої гнучкості та ефективності розроблених спеціалізованих цифрових пристроїв, що дозволяє швидко вийти на ринок. Кожне покоління FPGA Cyclone вирішує технічні проблеми, пов'язані з більшою інтеграцією, підвищеною продуктивністю, меншою потужністю споживання та швидшим виходом на ринок, дотримуючись основних вимог до витрат при виробництві даного типу ПЛІС. Основними типами Cyclone сімейства FPGA фірми Altera є: Intel Cyclone 10, Cyclone V, Cyclone IV, Cyclone III;

2) сімейство ПЛІС типу FPGA Intel MAX є дуже ефективним рішенням у питанні енергонезалежної інтеграції, забезпечуючи розширені можливості обробки за допомогою програмного логічного пристрою з низькою вартістю, швидкісним, малим форм-фактором. Завдяки сталій

подвійній конфігурації з аналоговими цифровими перетворювачами (АЦП) та повнофункціональними можливостями FPGA вони оптимізовані для широкого кола високопродуктивних та економічно чутливих додатків у сфері промисловості, транспортній та комунікаційній галузях. Основними типами MAX сімейства FPGA фірми Intel є: Intel MAX 10, Max V, Max II;

3) FPGA фірми Intel типу Stratix та серія SoC поєднують в собі високу щільність та високу продуктивність ПЛІС завдяки великій гнучкості та ефективності, що дає змогу реалізувати більше функцій пристрою, що розробляється, і максимально збільшити пропускну здатність системи, дозволяючи розробникам проектувати та використовувати високопродуктивні цифрові пристрої, які швидко виходять на етап ефективної та продуктивної експлуатації. Основними типами Stratix сімейства FPGA фірми Intel є: Intel Stratix 10, Stratix V, Stratix IV, Stratix III;

4) сімейство FPGA фірми Intel Arria забезпечує оптимальний баланс між продуктивністю та енергоефективністю розроблених цифрових пристроїв. Сімейство пристроїв Intel Arria має багатофункціональний набір пам'яті, логічних блоків та блоків обробки цифрових сигналів (DSP) у поєднанні з високою надійністю сигналу до 25,78 Гбіт / с, що дозволяє інтегрувати більше функцій та максимально збільшити пропускну здатність системи. Крім того, використання SoC у сімействах пристроїв Arria V та Intel Arria пропонують жорсткі процесорні системи (HPS) на базі ARM для ще більшої інтеграції та економії потужності споживання. Основними типами Arria сімейства FPGA фірми Intel є: Intel Arria 10, Arria V, Arria II та Arria GX [19].

## **2.5 Опис основних ПЛІС сімейства FPGA фірми Xilinx**

ПЛІС компанії Xilinx на сьогодні широко використовуються для вирішення задач різної складності. Сімейства ПЛІС FPGA фірми Xilinx складається з чотирьох основних FPGA:

- 1) FPGA Artix;
- 2) FPGA Kintex;
- 3) FPGA Spartan;
- 4) FPGA Virtex.

Всі ці типи ПЛІС відповідають широкому діапазону системних вимог для спеціалізованих цифрових пристроїв, що розробляються починаючи від дешевих та простих додатків до дорогих та складних у проектуванні та реалізації. Дані ПЛІС мають високошвидкісні смуги пропускання, велику кількість логічних елементів, можливості обробки сигналів для різноманітних цифрових пристроїв, що розробляються. Сімейство FPGA включає такі серії, як:

1) сімейство FPGA Spartan-7: добре оптимізовані для своєї низької ціною, мають низькою потужністю споживання та високою продуктивністю, пристрою вводу/виводу, має компактне розміщення на платі, що робить даний тип ПЛІС невеликих розмірів;

2) сімейство FPGA Artix-7: оптимізовані та гнучкі при створенні пристроїв із низьким рівнем енергії споживання; використовуються для створення пристроїв, які потребують послідовного використання зовнішніх периферійних пристроїв із високою пропускну здатністю DSP. Даний тип ПЛІС забезпечує найнижчу ціну виготовлення (загальна вартість матеріалів для виготовлення) для створення високопродуктивних, чутливих до витрат матеріалів цифрових пристроїв;

3) сімейство FPGA Kintex-7: розроблено та оптимізовано для найкращої продуктивності цифрових пристроїв, що розробляються, з 2-кратним покращенням аспектів роботи у порівнянні з попереднім поколінням, за невисоку ціну;

4) сімейство FPGA Kintex-7: розроблено та оптимізовано для найкращої продуктивності цифрових пристроїв, що розробляються, з 2-

кратним покращенням аспектів роботи у порівнянні з попереднім поколінням, за невисоку ціну.



Порівняльна характеристика сімейства ПЛІС FPGA від компанії Xilinx,  
а саме сьомого покоління:

- 1) кількість логічних елементів:
  - 1.1 Spartan-7 – 102 000;
  - 1.2 Artix-7 – 215 000;
  - 1.3 Kintex-7 – 478 000;
  - 1.4 Virtex-7 – 1 955 000;
- 2) розмір блоку RAM:
  - 2.1 Spartan-7 – 4.2 МБ;
  - 2.2 Artix-7 – 13 МБ;
  - 2.3 Kintex-7 - 34 МБ;
  - 2.4 Virtex-7 – 68 МБ;
- 3) кількість DSP Slices:
  - 3.1 Spartan-7 – 160;
  - 3.2 Artix-7 – 740;
  - 3.3 Kintex-7 - 1 920;
  - 3.4 Virtex-7 – 3600;
- 4) продуктивність DSP. Вимірюється у GMAC за одиницю часу, де GMAC – Giga multiply-accumulate operations, кількість операцій в секунду:
  - 4.1 Spartan-7 – 176 GMAC/c;
  - 4.2 Artix-7 – 929 GMAC/c;
  - 4.3 Kintex-7 – 2 845 GMAC/c;
  - 4.4 Virtex-7 – 5 335 GMAC/c;
- 5) процесор MicroBlaze. Вимірюється у DMIPs, де Dhrystone Million Instructions Per second, або є кількість мільйонів операція за одиницю часу:
  - 5.1 Spartan-7 – 260 DMIPs;
  - 5.2 Artix-7 – 303 DMIPs;
  - 5.3 Kintex-7 – 438 DMIPs;
  - 5.4 Virtex-7 – 441 DMIPs;

б) кількість трансиверів (це спеціальні пристрої для передачі та прийому сигналів):

6.1 Spartan-7 – відсутні;

6.2 Artix-7 – 16 одиниць;

6.3 Kintex-7 – 32 одиниць ;

6.4 Virtex-7 – 96 одиниць;

7) швидкість трансиверів:

7.1 Spartan-7 – відсутня;

7.2 Artix-7 – 6.6 Гб/с;

7.3 Kintex-7 – 12.5 Гб/с;

7.4 Virtex-7 – 28.05 Гб/с;

8) послідовна пропускна здатність:

8.1 Spartan-7 – відсутня;

8.2 Artix-7 – 221 Гб/с;

8.3 Kintex-7 – 800 Гб/с;

8.4 Virtex-7 – 2 784 Гб/с;

9) інтерфейс programmable logic controller (PLC):

9.1 Spartan-7 – відсутній;

9.2 Artix-7 – x4 Gen2;

9.3 Kintex-7 – x8 Gen2;

9.4 Virtex-7 – x8 Gen3;

10) інтерфейс пам'яті:

10.1 Spartan-7 – 800 Мб/с;

10.2 Artix-7 – 1 066 Мб/с;

10.3 Kintex-7 – 1 866 Мб/с;

10.4 Virtex-7 – 1 866 Мб/с;

11) кількість елементів вводу/виводу:

11.1 Spartan-7 – 400 одиниць;

11.2 Artix-7 – 500 одиниць;

11.3 Kintex-7 – 500 одиниць;

11.4 Virtex-7 – 1 200 одиниць;

12) напруга споживання елементів вводу/виводу:

12.1 Spartan-7 – 1.2-3.3 В;

12.2 Artix-7 – 1.2-3.3 В;

12.3 Kintex-7 – 1.2-3.3 В;

12.4 Virtex-7 – 1.2-3.3 В.

Сімейство ПЛІС FPGA компанії Xilinx сьомого покоління побудовано на основі сучасної, високопродуктивної, низько енергозатратної 28-нм технології high-k metal gate (HKMG). Сімейство сьомого покоління ПЛІС FPGA фірми Xilinx забезпечує значне збільшення продуктивності системи із пропускною здатністю елементів вводу/виводу до 2,9 Тб/с, близько двох мільйонів логічних елементів, а також 5,3 ТМАС/s DSP, при цьому споживаючи на 50% менше потужності, ніж цифрові пристрої, які побудовано на основі попереднього покоління сімейства FPGA, що пропонує реальну альтернативу використанню ASSP та ASIC [20].

### **2.5.1 Характеристика FPGA Spartan-6 компанії Xilinx**

Сімейство ПЛІС Spartan-6 забезпечує провідні можливості інтеграції системи з найменшою загальною вартістю для додатків великого обсягу. Тринадцять членів сім'ї забезпечують розширену щільність від 3 840 до 147 443 логічних елементів, які споживають у два рази менше енергії, ніж попередніх сімейства ПЛІС FPGA, а також швидші двосторонні зв'язки. ПЛІС Spartan-6 побудовано на основі 45 нм малопотужної технології із міді, що забезпечує оптимальний баланс витрат, потужності та продуктивності цифрових пристроїв, що проектуються та розробляються. Сімейство Spartan-6 пропонує нову, більш ефективну двоетапну реєстраційну логіку із використанням спеціальних (LUT), великий вибір вбудованих блоків системного рівня. До них відносяться 18 Кб (два по 9 Кб) блоків ОЗУ другого

покоління DSP48A1, контролери пам'яті SDRAM, блоки тайм менеджерів, блоки керування змішаним режимом, технологія SelectIO, блоки високошвидкісних серійних трансиверів з підтримкою налаштування потужності, PCI Express блоки, удосконалено режими керування живленням на рівні системи, опції автоматичного виявлення конфігурації та розширений IP-захист із захистом із використанням AES. Ці особливості забезпечують гарну альтернативи звичайних продуктів ASIC з безпрецедентним простотою використання за низькою вартістю. Сімейство FPGA Spartan-6 пропонує найкраще рішення для конструювання схем логіки, проектів DSP, які орієнтовані на споживача, та вбудовані додатки для зручності та ефективності проектування на основі даного сімейства. Фірма Xilinx пропонує широку підтримку FPGA Spartan-6, а саме спеціальні платформи, які містять інтегровані програмні та апаратні компоненти, які дозволяють розробникам зосередити увагу на всіх вимогах та особливостях пристроїв, як тільки починається їх розробка.

Загальний опис властивостей ПЛІС FPGA Spartan-6:

- 1) сімейство Spartan-6 поділяється на два типи: Spartan-6 LX FPGA із оптимізованою логікою; Spartan-6 LXT FPGA, які забезпечують послідовний високошвидкісний зв'язок;
- 2) розробка не вимагає великих затрат та зусиль, адже ПЛІС містить стандартні компоненти для проектування простих цифрових пристроїв, а саме декілька ефективних інтегрованих блоків; оптимізований набір стандартів вводу/виводу; проміжні прокладки на платі тощо;
- 3) мала потужність споживання: 45 нм технології оптимізовано для економії потужності споживання; реалізовано режим зниженого енерговикористання у сплячому режимі; мала напруга живлення 1.0 В; висока продуктивність ядра 1.2 В;
- 4) багатофункціональні багатоетапні блоки SelectIO: швидкість передачі даних до 1080 Мб/с на входах/виходах; стандарти та протоколи

введення / виводу від 3,3 до 1,2 В; недорогі інтерфейси пам'яті HSTL та SSTL; регульована швидкість введення/виведення для покращення цілісності сигналу;

5) високошвидкісні серійні приймачі GTP у FPGA LXT До 3,2 Гб/с; високошвидкісні інтерфейси, включаючи: Serial ATA, Aurora, 1G Ethernet, PCI Express, OBSAI, CPRI, EPON, GPON, DisplayPort і XAUI;

6) інтегрований блок PCI Express (LXT);

7) достатня підтримка технології PCI, сумісна з 33 МГц, 32- і 64-розрядними специфікаціями;

8) ефективні Slices DSP48A1: швидкий 18 на 18 мультиплексор та 48 бітовий акумулятор;

9) інтегровані блоки контролера пам'яті: підтримка DDR, DDR2, DDR3 та LPDDR; Швидкість передачі даних до 800 Мб/с (максимальна смуга пропускання 12,8 Гб/с);

10) велика кількість логічних елементів з підвищеною логічною спроможністю: необов'язковий регістр зсуву або розподілена підтримка оперативної пам'яті; ефективні 6-вхідні LUT покращують продуктивність та зменшують потужність споживання; LUT з подвійним тригером;

11) блоки оперативної пам'яті з широким діапазоном деталізації: швидка робота блоків пам'яті з введенням/виведенням байтів; 18 Кб блоків, які можуть бути запрограмовані як два незалежні ОЗУ по 9 Кб;

12) таблиця clock management tile (CMT) для підвищення продуктивності: потужна платформа Xilinx Platform Flash із JTAG; MultiBoot - підтримка віддаленого оновлення з декількома бітових потоками, що використовують для захисту;

13) покращена безпека захисту розроблених рішень: унікальний ідентифікатор DNA пристрою для конструктивної аутентифікації; використання алгоритму AES для шифрування бітового потоку у більших пристроях;

14) вбудований покращений швидкий процесор MicroBlaze для за низькою ціною;

15) провідні галузеві IP та навчальні проекти від компанії Xilinx;

16) забезпечення збереження розробленої користувачем конфігурації. FPGA Spartan-6 зберігає дані про налаштування конфігурації у внутрішніх регістрах типу SRAM. Розмір конфігурації становить від 3 Мб до 33 Мб залежно від розміру пристрою та варіантів реалізації користувацького рішення/алгоритму. Але зберігання конфігурації є непостійним, тому його необхідно перезавантажувати, коли FPGA запускається. Також дані про конфігурацію можна перезавантажити в будь-який час, встановивши PROGRAM\_B у низький рівень. Доступні декілька методів та форматів даних для конфігурації завантаження. Бітові конфігурації можуть бути або основними режимами, де FPGA генерує сигнал конфігурації (CCLK) або послідовним режимом, де джерело даних зовнішньої конфігурації є таймер FPGA.

ПЛІС сімейства FPGA Spartan-6 складається з таких основних частин (рис. 2.6), як:

1) CLBs, Slices, and LUTs. Кожен логічний блок (CLB) у FPGA Spartan-6 складається з двох Slice. У архітектурі Spartan-6 існують три типи CLB: SLICEM, SLICEL та SLICEX. Кожен Slice містить чотири LUT, вісім тригерів і та містить різну логіку. LUT призначені для універсальних комбінаторної або послідовної логічної підтримки. Синтезовані інструменти використовують ці високоефективні логічні, арифметичні та функціональні можливості пам'яті.

SLICEM займають четверту частину загальної кількості Slices FPGA Spartan-6. Кожен з чотирьох блоків SLICEM може бути налаштований як 6-вхідний LUT з одним виходом або як подвійний 5-вхідний LUT з однаковими 5-бітними адресами та двома незалежними виходами. Ці LUT можна також використовувати як розподілену 64-розрядну оперативну пам'ять з 64 бітами

або дві 32 бітові LUT, як єдиний 32-розрядний регістр зсуву (SRL32), або як два 16-бітових регістра зсуву (SRL16). Кожен вивід LUT може бути зареєстрований тригером в межах CLB. Для арифметичних операцій високошвидкісний ланцюг переносу передає сигнали переносу вгору по структурі Slices;

SLICEL займають четверту частину загальної кількості Slices FPGA Spartan-6, які містять усі особливості SLICEM, окрім функцій регістра пам'яті та регістра зсуву;

SLICEX займають четверту частину загальної кількості Slices FPGA Spartan-6. SLICEX мають ту ж структуру, що й SLICEL, за винятком можливості арифметичного переносу та мультиплексорів;

2) блок Clock Management. Кожен FPGA Spartan-6 має до шести CMT, кожен з яких складається з двох Digital Clock Manager DCM і одного PLL, які можна використовувати окремо або у системі;

3) блок Digital Clock Manager (DCM). Блок DCM забезпечує чотири фази вхідної частоти;

4) Phase-Locked Loop (PLL). PLL може слугувати, як засіб синтезу чистоти для більш широкого діапазону частот та у якості фільтра коливань для DCM;

5) блок Random Access Memory (RAM). Кожний FPGA Spartan-6 має від 12 до 268 двопортових блоків RAM, кожний розміром 18 Кб. Кожен блок RAM містить два повністю незалежних порти, які використовуються тільки для зберігання даних;

6) блоки вводу/виводу. Кількість елементів вводу/виводу варіюється від 102 до 576 одиниць, залежно від розміру пристрою та плати. Кожен елемент вводу/виводу можна налаштувати на велику кількість стандартів роботи. Для цього потрібно використовувати супровідну документацію на офіційному сайт компанії Xilinx. За винятком декілька спеціальних елементів вводу/виводу всі пакети мають однакові можливості елементів вводу/виводу.

Всі елементи вводу/виводу організовано у блоки, а саме у чотири блоки на менших пристроях та шістьома блоками на більших пристроях. Кожен блок має кілька розширених виводів напруги живлення VCCO, що також допускають певні вхідні буфери. Деякі одноразові буфери входу вимагають зовнішньої напруги (VREF). У кожному блоку є декілька елементів VREF-I / O для подвійного призначення. У даному блоку, коли стандарт вводу/виводу вимагає напруги VREF, кожен елемент VREF у цьому блоці повинен бути підключений до тієї ж шини [21].

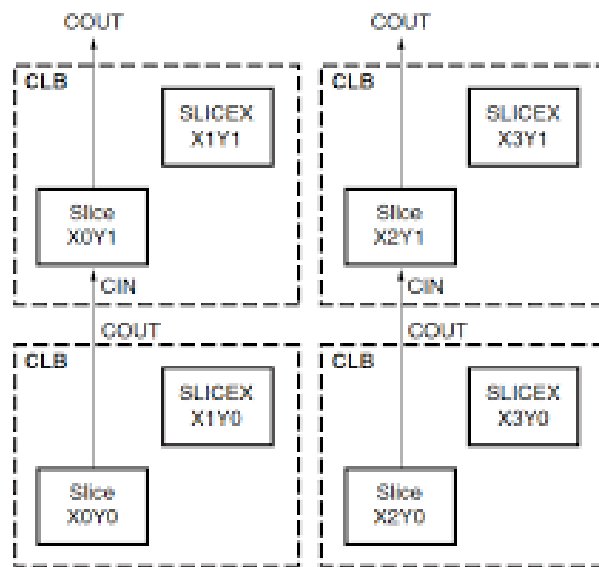


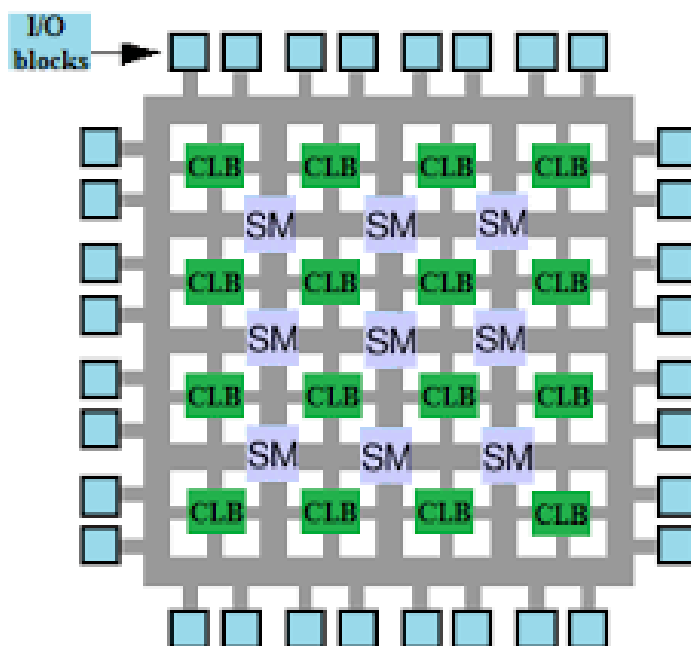
Рисунок 2.6 – Загальна архітектура ПЛІС FPGA Spartan-6

### 2.5.2 Характеристика FPGA Spartan-7 компанії Xilinx

Сімейство FPGA сімейства Spartan-7 Xilinx, що використовує високопродуктивну технологію HKMG, забезпечує найкраще поєднання високої продуктивності та низької потужності для проектування та розробки спеціалізованих цифрових пристроїв. FPGAs Spartan-7 побудовано на технології 28NPL. Результатом є сімейство компактних, оптимізованих за витратами ПЛІС, які забезпечують високу логіку та продуктивність вводу/виводу з строго контрольованим енергоспоживанням, за низькою ціною виготовлення.



Шість типів пристроїв FPGA Spartan-7 забезпечує кількість логічних елементів у діапазоні від 6000 до 102 400 одиниць із покращеною швидкістю підключення, ніж у попередніх поколіннях FPGA ПЛІС типу Spartan. Сімейство Spartan-7 (рис. 2.7) пропонує нову, більш ефективну логіку із 6-входовою LUT та багатий вибір вбудованих блоків системного рівня. До них відносяться блочні ОЗУ 36 Кб (2 по 18 Кб) з вбудованою FIFO логікою для буферизації даних на мікросхемах, DSP з мультиплікатором 25 на 18, 48-бітним акумулятором та попереднім суматором для високопродуктивної фільтрації, включаючи оптимізовану симетричну фільтрацію коефіцієнтів, вдосконалені блоки керування тактовою частотою змішаного режиму, технологія SelectIO з підтримкою DDR3 з інтерфейсом до 800 Мб/с, вдосконалені режими керування живленням на рівні системи, автоматичне визначення параметрів конфігурації та покращена захист інтелектуальної власності за допомогою алгоритму шифрування AES та пристрою DNA Ці функції забезпечують недорогу програмовану альтернативу користувачам пристроїв на основі ASIC. FPGA Spartan-7 є найкращим рішенням для гнучких та масштабованих логічних функцій з великим об'ємом, паралельної обробки даних, а також залежними від витрат проектів.



## Рисунок 2.7 – Загальна архітектура ПЛІС FPGA Spartan-7

ПЛІС FPGA Spartan-7 містить такі основні елементи, як CLB, Slices, and LUTs. Деякі основні особливості архітектури CLB включають в себе:

- 1) 6-вхідні пошукові таблиці (LUT);
- 2) можливості пам'яті в LUT;
- 3) функції регістру зсуву.

LUT в FPGA X Spartan-7 можна сконфігурувати як один 6-вхідний LUT (64-розрядних ПЗУ) з одним виходом, або як два 5- вхідні LUT (32-розрядні ПЗУ) з окремими виходами, але загальними адресами або логічними виходами. Чотири таких LUT-пристроїв та їх вісім тригерів, а також мультиплексори та арифметична логіка переносу утворюють один Slice, а два Slice утворюють блок CLB. Чотири з восьми тригерів на кожному Slice (по одному на LUT) можуть бути налаштовані як засувки. До половини усіх Slice також можуть використовувати свої LUT, як розподілену 64-бітну оперативну пам'ять або як 32-бітний регістр зсуву (SRL32) або як два SRL16s. Сучасні інструменти синтезу використовують ці високоефективні функції логіки, арифметики та пам'яті.

Блок RAM. Деякі з ключових особливостей ОЗП блоку:

- 1) двохпортний блок оперативної пам'яті обсягом 36 Кб;
- 2) програмована FIFO логіка;
- 3) вбудована додаткова схема виправлення помилок.

Кожен FPGA Spartan-7 має від 5 до 120 двосторонніх ОЗП, кожна з яких має розмір 36 Кб. Кожен блок пам'яті має два повністю незалежних порти, які використовуються лише для збережених даних.

DSP Slice. Деякі функції DSP:

- 1)  $25 \times 18$  двокомпонентний множник із високою розрядністю (48 біт) сигнального процесора;
- 2) енергозберігаючий попередній суматор для оптимізації використання симетричних фільтрів;

3) додаткові функції: додаткова конвеєрна обробка, додатковий Arithmetic and Logic Unit (ALU) і спеціальні шини для каскадування.

Програми DSP використовують багато бінарних мультиплікаторів, які ефективно застосовуються у спеціальних середовищах DSP. Всі 7 серій ПЛІС мають багато виділених, повністю конфігурованих, низько енерговитратних DSP елементів, що поєднують високу швидкість та малі розміри при збереженні гнучкості структури системи.

Кожний блок DSP складається з мультиплікатора 25 на 18 біт і 48-бітного акумулятора, обидва здатні працювати на частоті до 550 МГц.

DSP містить додатковий попередній суматор, який зазвичай використовується в симетричних фільтрах. Цей попередній суматор підвищує продуктивність в щільно упакованій конструкції та зменшує кількість блоків DSP майже на половину. DSP також містить 48-розрядний шаблон, який можна використовувати для конвергентного або симетричного округлення. Цей шаблон також здатний реалізувати 96-бітові логічні функції при сумісному використанні із логічним блоком.

Блок DSP забезпечує широкі можливості конвеєрної обробки та розширення, які підвищують швидкість та ефективність багатьох додатків, окрім цифрової обробки сигналів, таких як широкоекранні перемикачі шин, генератори адрес пам'яті, широкоформатні мультиплектори та файли регістрів вводу/виводу пам'яті. Перемножувач також може використовуватися для роботи синхронного лічильника.

Блоки вводу/виводу. Деякі основні функції блоків вводу/виводу: високопродуктивна технологія SelectIO з підтримкою DDR3 до 800 Мбіт/с;

Кількість контактів вводу/виводу залежить від розміру пристрою та пакету. Кожен контакт вводу/виводу налаштовується і може відповідати одному із великої кількості стандартів вводу/виводу. За винятком контактів живлення та декількох виділених контактів для конфігурування, всі інші ПЛІС FPGA Spartan-7 мають ті самі можливості вводу/виводу. Елементи

вводу/виводу у ПЛІС FPGA Spartan-7 забезпечують широкий діапазон підтримки напруги, від 1.2 В до 3.3 В.

Контакти вводу/виводу ІВ у ПЛІС Spartan-7 FPGA організовані в блоки, із кількістю контактів - 50 одиниць. У кожного блоку є одна загальна VCCO вихідна пропозиція, яка також має певні вхідні буфери. Деякі вхідні буфери, вимагають внутрішньо або зовнішньо згенерованої опорної напруги (VREF). У кожному блоку є два VREF-контакти (за винятком конфігураційного блоку 0). Кожний блок може мати лише одну величину напруги VREF.

Analog-to-Digital Converter (XADC). Всі FPGA Xilinx 7 серії (крім XA7S6 і XA7S15) включають новий гнучкий аналоговий інтерфейс під назвою XADC. У поєднанні з програмованою логікою FPGA 7-серії, XADC може обробляти широкий діапазон даних та вимог щодо збору та моніторингу даних. Основні архітектурні особливості XADC включають: один подвійний 12-розрядний аналогово-цифровий перетворювач; до 17 гнучких та налаштовуваних користувачем аналогових входів; безперервний доступ Joint Test Action Group (JTAG) до вимірювань АЦП [22].

## **2.6 Висновки до другого розділу**

На сьогоднішній день ПЛІС є однією із найбільш перспективних та використовуваних елементних баз для побудови спеціалізованих цифрових пристроїв різного призначення. Перевагами ПЛІС є такі властивості, як: універсальність та високий попит; низька вартість виробництва; масове виробництво; висока швидкодія та надійність; різноманітність у виборі напруги живлення та параметрів сигналів введення/виведення; підтримка зручних та ефективних програмних засобів автоматизованого проектування від провідних компаній; швидкий час проектування; простота та зручність модифікації проектів на будь-яких етапах розробки.

ПЛІС можна поділити на такі основні сімейства: CPLD, FPGA, FLEX, SOC.

Сімейства ПЛІС FPGA і CPLD - це два найбільш поширені типи цифрових логічних мікросхем у використанні. Виділяючи всі основні особливості та недоліки сімейств ПЛІС FPGA та CPLD можна виділити такі основні відмінності: FPGA містить до 100 000 логічних блоків, тоді як CPLD містить лише кілька тисяч логічних блоків; FPGA є більш ефективними при проектуванні та реалізації складних цифрових пристроїв, тоді як ПЛІС сімейства CPLD краще для простих спеціалізованих цифрових пристроїв.

Рішення про те, що використовувати, ПЛІС сімейства FPGA або CPLD, розробник повинен обирати залежно від цілей проектування та можливостей спеціалізованого цифрового пристрою, що розробляється.

ПЛІС компанії Xilinx на сьогодні широко використовуються для вирішення задач різної складності. Сімейства ПЛІС FPGA фірми Xilinx складається з чотирьох основних FPGA: Artix, Kintex, Spartan, FPGA Virtex. Всі ці типи ПЛІС відповідають широкому діапазону системних вимог для спеціалізованих цифрових пристроїв, що розробляються починаючи від дешевих та простих додатків до дорогих та складних у проектуванні та реалізації.

## РОЗДІЛ 3 ЗАСОБИ ТА МЕТОДИ ОБРОБКИ ДАНИХ НА ПЛІС

### 3.1 Обробка даних на ПЛІС

На сьогоднішній день існує велика кількість рішень задачі паралельної обробки даних на основі електронних компонентів, таких як: базові матричні кристали (але вони потребують заводського виробничого процесу для програмування); логічний контролер, що програмується; ВІС для окремих певно визначених задач, виготовлення яких при малій кількості досить дорого; мікроконтролери, реалізація алгоритмів на яких відбувається програмним шляхом.

Використання ПЛІС є доречним при паралельній обробці даних в наукових, промислових чи учбових цілях. Найбільші компанії із виготовлення та підтримки ПЛІС такі, як Altera, Achronix та Xilinx мають велику кількість вже готових рішень обробки даних.

На сьогодні розповсюджені такі високопродуктивні обчислювальні систем, які мають статичну структуру обчислювального середовища. Даний фактор не дозволяє ефективно використовувати та реалізовувати складні алгоритми обробки даних, оскільки структура таких алгоритмів може не повністю відповідати архітектурі комп'ютерної системи.

Існує багато напрямків підвищення ефективності роботи алгоритмів, які реалізують задачі із великою кількістю інформаційних зв'язків. Одним з таких напрямків є проектування та побудова паралельних обчислювальних систем, які мають можливість реконфігурації каналів зв'язків між процесорами. Наприклад, система Non-Uniform Memory Access, мультипроцесори, трансп'ютери тощо.

Використання ПЛІС для побудови реконфігурованих комп'ютерних систем є актуальним та доречним рішенням для підвищення ефективності паралельної обробки даних. Така концепція базується на програмно-

апаратній реалізації поставленої задачі обробки даних, яка ґрунтується на пришвидженні певних частин алгоритму, оскільки їх виконання є недоречним за допомогою простих процесорів. Використання ПЛІС дозволить реалізувати на одному кристалі мікросхеми систему для обчислень із гнучкою структурою. Оскільки ПЛІС має можливість швидкого перепрограмування та перебудови обчислювальної структури, то це дозволяє реалізовувати довільні алгоритми обробки даних. Такий програмно-апаратний підхід дозволить забезпечити ефективне та просте користування, підтримку та реалізацію методів паралельного програмування, широку функціональність виготовленого пристрою та економічну привабливість.

### **3.2 Огляд основних засобів проектування на ПЛІС**

Xilinx створює сучасний продукт - програмне забезпечення для конфігурування кристалів та розробки проектів. Повний перехід на покоління систем автоматизованого проектування (Integrated Synthesis Environment, ISE) стався 16 років тому (2002). Раніш ISE використовувалось як альтернатива Foundation Series (попередня версія САПР). Переваги ISE: скорочує час на розробку, збільшує ефективність результатів завдяки вдосконаленим методам проектування, алгоритмам синтезу, розміщення та трасування проекту на кристалі.

САПР фірми Xilinx наразі існують у таких конфігураціях:

1) програмний додаток Foundation ISE. Дана САПР є найбільш повноцінною ISE серед усіх САПР фірми Xilinx, адже вона підтримує роботу із усіма сімействами та типами ПЛІС;

2) для спряження із САПР других виробників є конфігурація Alliance ISE яка підтримує всі кристали Xilinx;

3) програмний додаток ISE Design Suite. Дана САПР знаходиться у вільному доступі на офіційному сайті компанії Xilinx. Обмеженням САПР WebPack ISE є те, що за допомогою даної САПР можна розробляти проекти

на основі кристалів сімейства CPLD та FPGA, але логічна ємність не може становити понад 300 000 системних вентилів;

4) нова САПР від Xilinx - Vivado High Level Synthesis (далі VHLS). VHLS створена будування цифрових пристроїв на базі мов програмування високого рівня. Головна мета – спрощення процесу проектування для тих розробників, що володіють програмуванням. Але на практиці ПЛІС може викликати складності для чисто профільних програмістів, коли вони зустрічають велику кількість незвичних для себе задач, якщо ті не еквівалентні оператору мови програмування (наприклад оператори описання апаратури).

Усі описані вище САПР відрізняються такими показниками як сімейства та типи ПЛІС, які здатна підтримувати та чи інша САПР; додаткові чи унікальні інструменти при проектуванні. Однак усі описані вище САПР мають однотипну структуру та одно типовий інтерфейс використання (за винятком САПР Alliance ISE) [23].

### **3.2.1 Огляд пакету проектування Vivado HLS**

САПР Vivado Design Suite призначений для підвищення продуктивності пристрою, що розробляється. Дана САПР містить набір інструментів для збільшення загальної продуктивності для проектуванні, інтеграції та впровадження систем використовуючи пристрої фірми Xilinx. Xilinx постійно удосконалює свої пристрої, які постійно стають більш ефективними та продуктивними Це такі впровадження, як технологія стекового кремнієвого з'єднання (SSI), високошвидкісні інтерфейси вводу-виводу до 28 гігабайт (Гб), тощо. Але й з удосконаленням пристроїв зростає складність проектів, що проектуються, тому удосконалення рішень проектування є дуже актуальним, оскільки це може підвищити продуктивність пристрою та пришвидшити його вихід на ринок. Завдяки Vivado Design Suite можна прискорити час розробки пристрою,



використовуючи вбудовані інструменти для проектування, або ж розробник може аналітично оптимізувати показники, такі як час роботи, затримки, потужність та ефективність пристрою, що проектується. Vivado Design Suite надає вам можливість аналізувати роботу та можливості пристрою на кожному етапі проектування.

САПР Vivado Design Suite є альтернативою програмного середовища Xilinx Design Suite ISE. Він має такі ж інструменти ISE Design Suite, як Project Navigator, технологію синтезу Xilinx (XST), інструмент CORE Generator, редактор для обмежень синхронізації, ISE Simulator (ISim), аналізатор ChipScope, аналізатор потужності пристрою на ПЛІС, редактор FPGA, інструмент розробки Planehead та SmartXplorer. Всі ці можливості тепер вбудовані безпосередньо у дизайні Vivado ISE.

Використовуючи САПР Vivado Design Suite, весь процес оформлення може бути виконаний у пам'яті без необхідності писати або перекладати будь-які проміжні формати файлів, що прискорює час виконання, налагодження та реалізації проекту при знижених вимогах до використання пам'яті. Всі інструменти Vivado Design Suite написані за допомогою власної мови Tcl. Всі команди та опції доступні у Vivado Integrated Design. Середовище розробки, що є графічним інтерфейсом користувача (GUI) для Vivado Design Suite, доступне через Tcl. Vivado Design Suite також забезпечує швидкий та зручний доступ до налаштування проекту, також команд та опцій інструментів, дозволяє швидко та легко генерувати звіти про конфігурацію пристрою.

У дизайнерському комплексі Vivado ISE реалізовано передові, ефективні моделі даних та структури, щоб використовувати менше пам'яті. Наприклад для Ethernet MAC, що є конкуруючим програмним забезпеченням, потрібно майже 16 Гб оперативної пам'яті для реалізації великих проектів в той час як Vivado Design Suite потребує близько 10 Гб оперативної пам'яті для рішень таких же задач 5. Зменшення витрат пам'яті

означає більшу продуктивність при проектуванні пристрої на Vivado Design Suite, тому що розробники можуть реалізовувати більш складні проекти без втрати пам'яті.

Vivalo Design Suite забезпечує найкращі в своєму класі інструменти для аналізу та оптимізації системного живлення пристрою, що розробляється. Починаючи з етап вибору архітектури пристрою, дизайнери можуть покладатися на точність і легкість використання електронної таблиці (ХРЕ) для визначення енергоспоживання системи. Зручна у використанні ХРЕ Quick Estimate має можливість робити прості порівняння декількох реалізацій, що дає розробнику можливість точного налаштування налаштувань для моделювання різних сценаріїв. Коли ваш проект переходить на етап компіляції, Vivado Design Suite продовжує пропонувати точний аналіз енергії та оцінка. Також Vivado Design Suite забезпечує автоматичне зменшення потужності, не завдаючи негативного впливу на ваш системні рішення по часу. Якщо вам потрібна додаткова економія енергії, ви можете використовувати унікальні можливості Vivado Design Suite використовуючи технологію чіткозернистої синхронізації Xilinx 7 серії для подальшого зниження енергоспоживання по всій системі або тільки її частині [24].

### **3.2.2 Огляд пакету проектування ISE Design Suite**

Програмне забезпечення ISE Design suite - це повністю орієнтована на поверхне проектування FPGA рішення для таких операційних систем, як Linux, Windows XP та Windows 7. ISE Design suite – дуже зручне та гнучке рішення для проектування пристроїв на базі FPGA та CPLD, що забезпечує синтез HDL та моделювання, реалізацію, програмування JTAG тощо. ISE Design suite забезпечує повний цикл проектування цифрового пристрою, що забезпечує миттєвий доступ до певних інструментів розробника ISE. Компанія Xilinx створила рішення, яке дає високу продуктивність,

забезпечуючи конструктивне рішення, яке завжди актуальне при проектуванні цифрових пристроїв.

За допомогою середовища розробки фірми Xilinx ISE Design Suite розробник або команда розробників може досягти більшої продуктивності та швидшого приведення рішення до виробництва. Удосконалення технології проектування полегшують оптимізацію потужності та спрощують інтеграцію IP. САПР ISE Design Suite забезпечує тісний зв'язок між рішеннями розробника для спрощення інтеграції вбудованих DSP, IP та користувацьких блоків в одній системі.

САПР фірми Xilinx ISE Design Suite версії 14 має такі позитивні властивості, як:

- 1) підвищує продуктивність розробника, створюючи середовище зручного проектування, що спрощує роботу;
- 2) підтримка портфоліо всіх програмованих пристроїв Xilinx;
- 3) єдиний, стандартний інтерфейс, що підтримує AMBA 4 AXI4 IP для рішень підключення та відтворення;
- 4) підвищення зручності та гнучкості використання програми PlanAhead;
- 5) інтуїтивно зрозумілий процес проектування з можливістю часткової реконфігурації, що знижує складність системи проектування широкого кола високоефективних цифрових пристроїв;
- 6) розумна технологія часового з'єднання, що знижує динамічне споживання електроенергії пристроєм, що розробляється, на 30 відсотків.

САПР фірми Xilinx ISE Design Suite надає розробникам доступ до широкого кола типів ПЛІС, що підтримується відкритими стандартами, загальними потоками дизайну, IP та платформами під час виконання. ISE Design Suite має все це разом із спеціальними середовищами, що належать до пакету даної САПР для задоволення потреб розробників, а саме потужності та продуктивності проектування пристроїв фірми Xilinx.

На сьогодні є декілька модифікацій САПР фірми Xilinx ISE Design Suite, які містять повний чи частковий функціонал та набір інструментів для потреб проектування:

1) модифікація ISE Design Suite Logic Edition. Це повне рішення для створення логіки та дизайну пристрою, що проектується. Дана модифікація містить базову методологію та IP. ISE Design Suite Logic Edition включає в себе: спільне середовище розробки, що забезпечує можливість розробникам працювати у команді за допомогою набору спеціальних інструментів; часткова реконфігурація, що підтримує FPGA Virtex-6 та Virtex-7 та EPP Zynq-7000; розумна технологія синхронізації, що підтримує FPGA Virtex-6, Spartan-6, Virtex-7, Kintex-7 та Zynq-7000; підтримка рішень підключення та відтворення через протокол AMBO 4 AXI4. Ці можливості є доповненням до інструментів та технологій Xilinx для введення проектування, синтезу, реалізації та перевірки роботи пристрою, а також допомагає досягти оптимального результату дизайну в найкоротший термін;

2) модифікація ISE Design Suite Embedded Edition. До особливостей технологій та IP, Logic Edition має вбудований набір інструментів та IP для рішень, які використовують переваги Xilinx EPP і FPGA з вбудованим жорстким процесором PowerPC або потужними процесорами MicroBlaze. Розробники мають доступ до платформи Xilinx, набору інструментів Studio, які пропонують графічне середовище та підтримку командного рядка для розробки апаратних платформ для вбудованих програм;

3) модифікація ISE Design Suite DSP Edition. Дана модифікація призначена для вирішення унікальних потреб методології дизайну DSP. Підтримка DSP починається з інструментів, технологій та IP Logic Edition, генератор системи для DSP та спеціалізованого IP. Навіть розробники, які є новачками у сфері програмованої логіки, можуть використовувати System Generator для DSP, щоб швидко та зручно створювати якісних FPGA-реалізацій алгоритмів DSP у сфері традиційного часу розробки RTL.

4) модифікація ISE Design Suite System Edition. System Edition включає в себе всі інструменти, технології та логіку роботи з IP, DSP тощо. Це повне і гнучке рішення відповідає потребам користувача, який потребує унікальної методології, яка поєднує в собі потужну функціональність DSP з вбудованим набором зручних та гнучких інструментів, або який хоче суттєво розширити дизайн DSP за допомогою можливостей вбудованого процесора;

5) модифікація WebPack ISE. Програмне забезпечення WebPack ISE - це безкоштовна повністю орієнтована на поверхню проектування FPGA рішення для таких операційних систем, як Linux, Windows XP та Windows 7. WebPack ISE забезпечує повний цикл проектування цифрового пристрою, що забезпечує миттєвий доступ до певних інструментів розробника ISE. Компанія Xilinx створила рішення, яке дає високу продуктивність, забезпечуючи конструктивне рішення, яке завжди актуальне при проектуванні цифрових пристроїв. ISE WebPack містить найважливіші інструменти для проектування CPLD та FPGA малого та середнього розміру. Включає такі компоненти, як інструменти ISE Design (з підтримкою зменшеного числа пристроїв), PlanAhead, DSP IP з'єднання. ChipScope Pro та Embedded Development Kit також будуть встановлені з WebPack, але ліцензовані окремо (не входять у файл ліцензії WebPack) [25].

### **3.3 Способи опису апаратури**

При проектуванні цифрового пристрою розробник має змогу обрати спосіб опису пристрою, що розробляється:

1) схематичний опис апаратури. Такий опис використовується розробниками, коли ваш файл верхнього рівня - це схематичний файл. Схематичний проект може включати в себе модулі нижчого рівня різних типів файлів, такі як файли HDL, інші схеми та інші компоненти, такі як IP-сердечники та файли EDIF. Project Navigator автоматично перетворює будь-які схематичні файли у проекті розробника до структурного HDL для

реалізації, отже, розробник повинен вказати інструмент синтезу під час роботи з проектами, що описують апаратуру схематичним способом;

2) Electronic Data Interchange Format (EDIF) файл. Такий опис використовується розробниками, коли вони використовують інструменти для синтезу поза набором інструментів Project Navigator;

3) NGC або NGO файли. NGC файл - це файл із мережевим списком, який містить як логічний дизайн, так і конфігурацію. Цей файл замінює файли EDIF та Netlist Constraints File (NCF). NGO файл – це спеціальний бінарний файл, що містить логічний опис конструкції з точки зору його початкових компонентів та ієрархії;

4) Hard Description language (HDL). HDL – це мова опису апаратури змішаного рівня в якій програмні конструкції використовуються із функціональними характеристиками. Цей змішаний рівень дозволяє описати архітектуру системи на високому рівні абстракції, а потім поетапно удосконалювати детальну реалізацію на рівні матриці. Опис за допомогою HDL має такі переваги, як: можливість перевірити функціональність дизайну на початку процесу проектування. Моделювання рішення на цьому високому рівні дозволяє детально оцінити архітектурні та проектні рішення; Опис HDL легкий для розуміння користувачам, ніж схематичний вид опису апаратури. Оскільки початковий опис HDL є незалежним процесом, то розробник має змогу використовувати свої напрацювання для реалізації інших проектів; Об'ємні рішення та задачі значно зручніше реалізовувати за допомогою інструментів HDL, А ніж схематичний метод опису апаратури. Після створення проекту за допомогою HDL розробнику потрібно синтезувати його. Під час цього процесу програмний код, який описано у файлі HDL транслюється у структурний список, а інструменти синтезу оптимізовано для швидкого та зручного використання розробником або командою розробників. Крім того, Xilinx пропонує свій власний інструмент для синтезу Xilinx

Synthesis Technology (XST). Із більш детальною інформацією користувач може ознайомитися на офіційному сайті компанії Xilinx [26].

### **3.4 Висновки до третього розділу**

Багато високопродуктивних обчислювальних систем мають статичну структуру обчислювального середовища. Тому це не дозволяє ефективно реалізовувати та використовувати складні алгоритми обробки даних, адже архітектура комп'ютерної системи може не відповідати структурі таких алгоритмів. Використання ПЛІС для побудови реконфігурованих комп'ютерних систем є актуальним та доречним рішенням для підвищення ефективності паралельної обробки даних. Оскільки ПЛІС має можливість швидкого перепрограмування це дозволяє реалізовувати довільні алгоритми обробки даних.

Одна з провідних фірм із виготовлення та підтримки ПЛІС Xilinx розробляє спеціальні системи автоматизованого проектування. Даний засіб суттєво полегшує розробку проектів будь-якої складності. Найпоширенішими САПР фірми Xilinx є Foundation ISE, конфігурація Alliance ISE, ISE Design Suite та Vivado High Level Synthesis. Усі ці описані САПР відрізняються такими показниками як сімейства та типи ПЛІС, які здатна підтримувати та чи інша САПР; додаткові чи унікальні інструменти при проектуванні. Однак усі описані вище САПР мають однотипну структуру та одно типовий інтерфейс використання, окрім конфігурації САПР Alliance ISE.

При проектуванні цифрового пристрою розробник має змогу обрати спосіб опису пристрою, що розробляється. Найбільш поширеними є схемний опис апаратури та опис апаратури за допомогою мов VHDL або Verilog.

## **РОЗДІЛ 4 РЕАЛІЗАЦІЯ ПАРАЛЕЛЬНОЇ ОБРОБКИ ДАНИХ НА ПЛІС**

### **4.1 Етапи проектування цифрових пристроїв на базі ПЛІС**

Під час розробки проекту реалізації цифрового пристрою на базі ПЛІС фірми Xilinx можна визначити наступні етапи проектування:

- 1) вибір фірми, сімейства та типу ПЛІС;
- 2) створення нового проекту у спеціальному САПР WEB Pack ISE фірми Xilinx;
- 3) обговорення опису пристрою, який буде проектуватися. Це може бути схема, опис алгоритму чи звичайна текстова форма;
- 4) синтез пристрою;
- 5) етап функціонально моделювання;
- 6) розміщення і трасування проекту на кристалі;
- 7) часове моделювання;
- 8) процес програмування ПЛІС;
- 9) завантаження проекту на кристал.

Перед початком створення нового проекту розробнику потрібно визначитися які методи опису пристрою він буде використовувати, для вирішення якої задачі буде створюватися пристрій, вибір засобів синтезу. Наприклад, при виборі фірми, сімейства та типу ПЛІС для створення пристрою необхідно врахувати багато факторів:

- 1) складність проекту;
- 2) швидкодія;
- 3) потужність споживання;
- 4) умови при яких відбуватиметься експлуатація пристрою;
- 5) вартість розробки;
- 6) можливість перепрограмування ПЛІС.



Деякі обрані параметри можна змінювати під час проектування пристрою. Це такі параметри, як тип кристалу або сімейство та тип ПЛІС. Розроблюваний пристрій може бути інформаційно представлений у декількох видах: принципова схема, опис на мові HDL, діаграма станів, пакети або бібліотеки розробника. На цьому етапі створення пристрою можна також накласти певні тимчасові чи топологічні обмеження, які будуть враховуватися при синтезі, розміщенні й трасуванні проекту на кристалі ПЛІС.

У процесі синтезу, враховуючи вхідні модулі проекту, формується спеціальний список з'єднань у якому містяться набори примітивів та компонентів, що можна буде реалізувати на базі ресурсів обраного сімейства та типу ПЛІС. Результат синтезу потім використовується як вхідні дані за допомогою засобів розміщення і трасування.

Процес функціонального моделювання пристрою, що розробляється, відбувається, невраховуючи реальні значення затримок проходження сигналів, що дозволяє проконтролювати відповідність до вихідних сигналів під час роботи алгоритму, який було описано для певної задачі.

На етапі розміщення та трасування проекту на кристалі відбувається розподілення функцій, що будуть виконуватися, у спеціальні конфігуровані логічні блоки (CLB) або ж макрокомірки, що визначається в залежності від сімейства та типу ПЛІС на основі якого створюється спеціалізований пристрій. Під час цього етапу створення також відбувається визначення реальних значень затримок під час поширення сигналів, які потрібні для повного проміжного тестування пристрою. У результаті розміщення й трасування проекту є створення спеціального файлу, який містить конфігураційну інформацію про ПЛІС.

Фінальний етап розробки спеціального пристрою - це завантаження певних конфігураційних параметрів на кристал. Для цього використовуються спеціальні програми та кабелю для завантаження.

Під час кожного з описаних вище етапів опису, проектування та створення спеціального цифрового пристрою потрібно ретельно дотримуватися всіх рекомендацій, які можна знайти на веб-сайтах провідних компаній із підтримки ПЛІС. Сучасні засоби проектування, які містяться у САПР від провідних компаній, є високоефективними, а тому дозволяють знаходити та змінювати більшість помилок, що значно скорочує час створення спеціального цифрового пристрою. Якщо під час створення проекту були виявлені такі помилки, як, наприклад, логічні помилки у процесі функціонально моделювання, некоректні результати проміжного тестування, то необхідно повертатися на етап створення та розробки вхідного опису проекту, провести необхідні зміни, а потім повторити потрібні етапи.

#### **4.2 Етапи створення проекту за допомогою WEB Pack ISE**

Проект було створено за допомогою САПР ISE Design ISE модифікація WebPACK ISE фірми Xilinx. WebPACK ISE підтримує реалізацію усіх етапів створення та проектування спеціалізованого пристрою. Проект, створений у WebPACK ISE, містить такі файли, як:

- 1) супровідна документація проекту;
- 2) детальний звіт про виконання головних етапів проектування;
- 3) вихідний опис пристрою, що проектується, поданий у графічному або текстовому вигляді;
- 4) проміжні результати, що будуть використовуватися для вихідних даних при подальших етапах проектування пристрою;
- 5) набір тестових даних, які будуть використовуватися при моделюванні пристрою;
- 6) фінальні результати проектування, які буду використовуватися для конфігурації розробленого проекту за допомогою САПР.

Кожна із описаних частин зберігаються у вигляді спеціальних файлів, які буде розміщено у директорії, що матиме назву, яку вказав розробник під

час початкових налаштувань проекту. На початку створення проекту він містить тільки файл із параметрами проекту. При подальшій роботі структури проекту додаються файли опису пристрою, що проектується. Після завершення кожного із етапів розробки спеціального пристрою результати, які було отримано на цьому кроці, додаються до відповідного файлу та зберігаються у відповідній директорії. Для більш детального опису роботи пристрою розробник має змогу створювати спеціальний файл із необхідною текстовою документацією.

Для того, щоб створити новий проект за допомогою САПР WEB Pack ISE розробнику повинен відкрити відповідну програму (рис. 4.1).

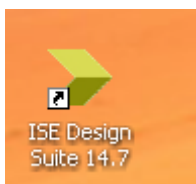


Рисунок 4.1 – Програмний додаток WEB Pack ISE

Початкове вікно для роботи із САПР WEB Pack ISE виглядає наступний чином (рис. 4.2):

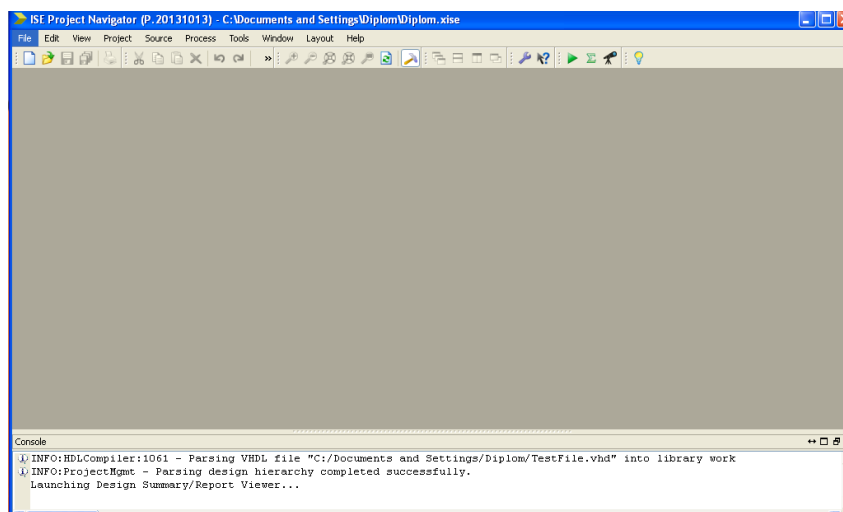


Рисунок 4.2 – Початкове вікно САПР WEB Pack ISE

Щоб створити новий проект необхідно вибрати пункт меню «New Project» у «Files». Після цього перед користувачем з'явиться спеціальне вікно для заповнення початкових параметрів проекту.

У першому вікні потрібно вказати (рис. 4.3):

- 1) назву проекту;
- 2) назва та шлях директорії, де буде розміщено проект;
- 3) короткий опис проекту, що розробляється;
- 4) тип опису проекту.

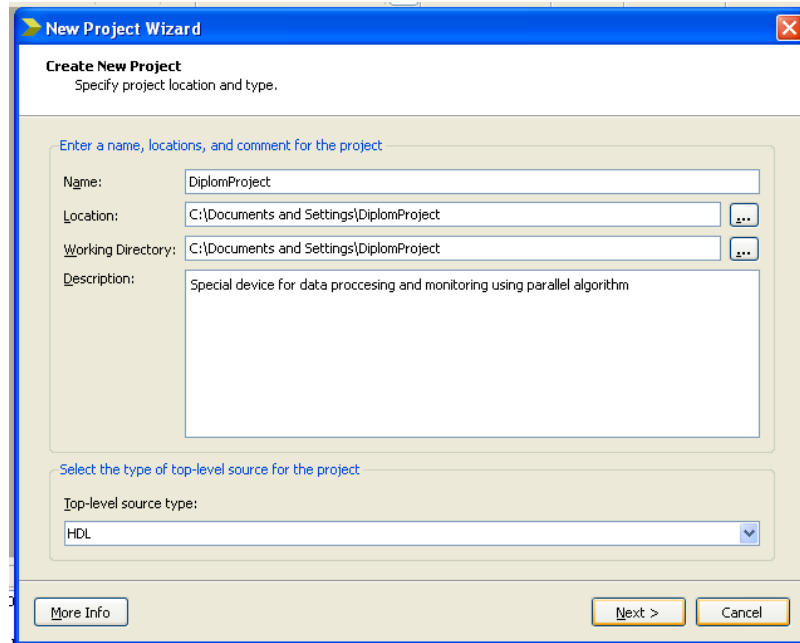


Рисунок 4.3 – Внесення початкових параметрів нового проекту

Далі потрібно вказати такі основні налаштування, як (рис. 4.4):

- 1) сімейство ПЛІС;
- 2) тип кристалу;
- 3) засоби для синтезу пристрою;
- 4) тип опису пристрою, що розробляється

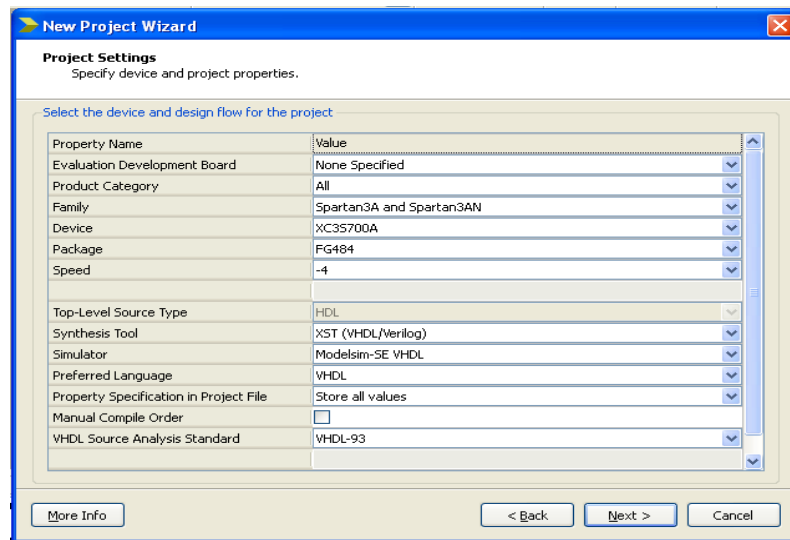


Рисунок 4.4 – Початкові параметри нового проекту на САПР WEB  
Pack ISE

На офіційному сайті компанії Xilinx є рекомендації для користувача, щодо створення нового проекту. Серед основних рекомендацій є:

1) доречним буде під час створення нового проекту вказувати спеціальну нову директорію, назва якої буде повністю чи частково відображати поставлену задачу, яку повинен виконувати пристрій, що розробляється;

2) ознайомитися із основними характеристиками сімейства та типу ПЛІС, яка буде використовуватися для створення спеціалізованого пристрою.

Після вище описаних дій користувач побачить вікно із звітом про виконану роботу (рис.4.5).

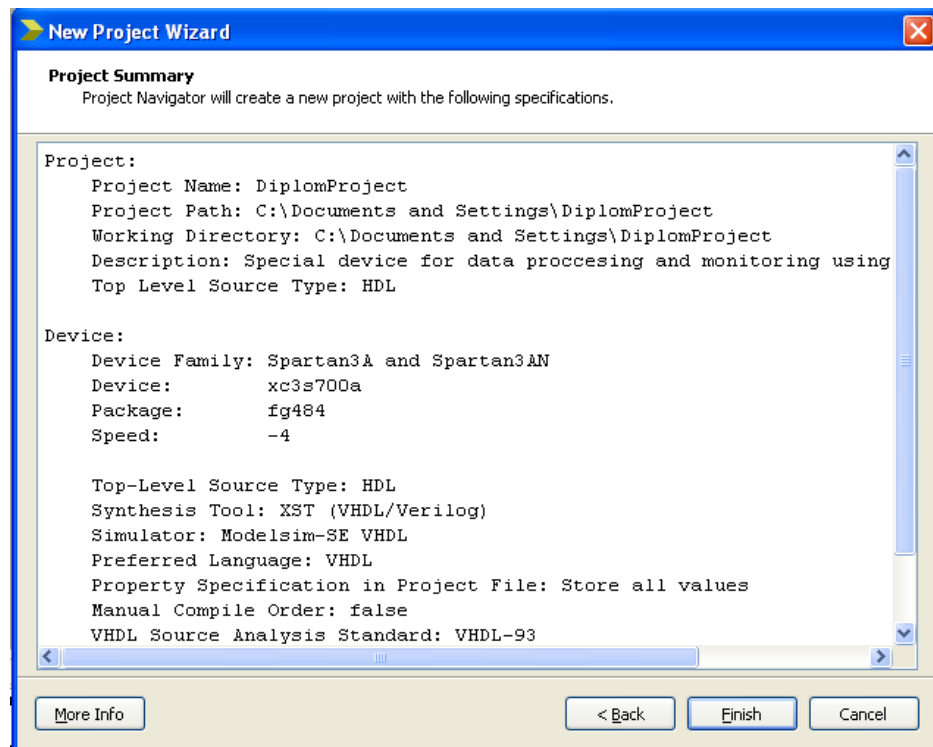


Рисунок 4.5 – Документація, яка формується при створенні нового звіту

Тепер потрібно включити до нашого проекту файл з описом нашого пристрою. Для цього потрібно обрати пункт меню «New Source». На даному етапі створення нового проекту можна обрати два варіанти:

- 1) «VHDL Module»;
- 2) «Verilog Module».

Потім потрібно вказати ім'я для обраного модуля (рис. 4.6).

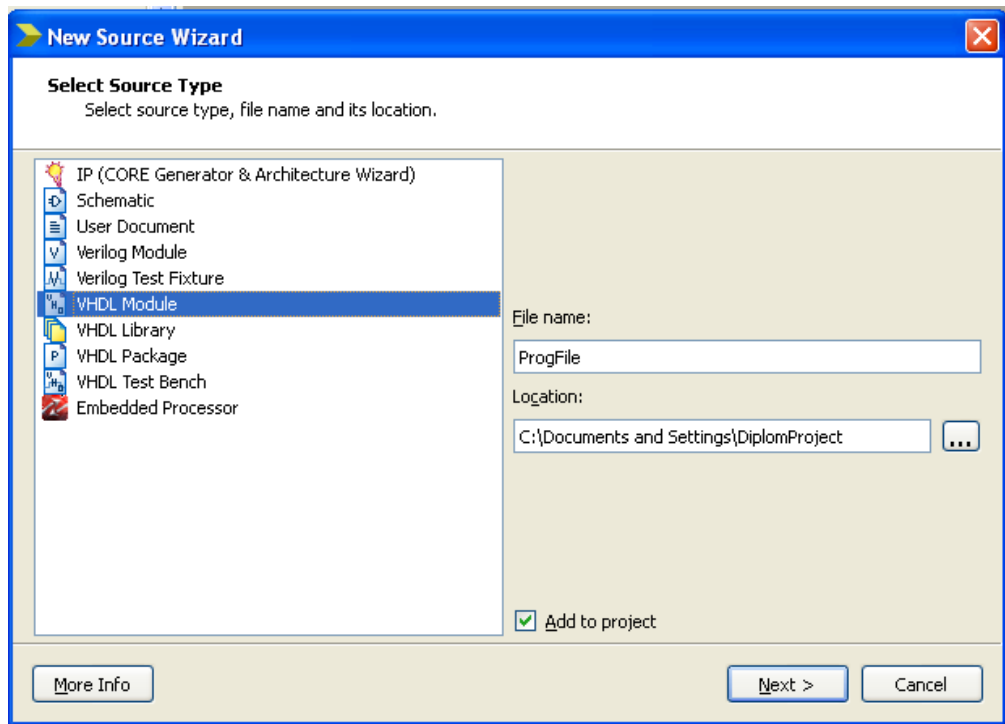


Рисунок 4.6 – Створення файлу для опису пристрою, що проектується

Наступним етапом створення проекту буде процес вибору входів та виходів пристрою, що проектується (рис. 4.7). Тут користувач може зробити налаштування зразу або потім вручну.

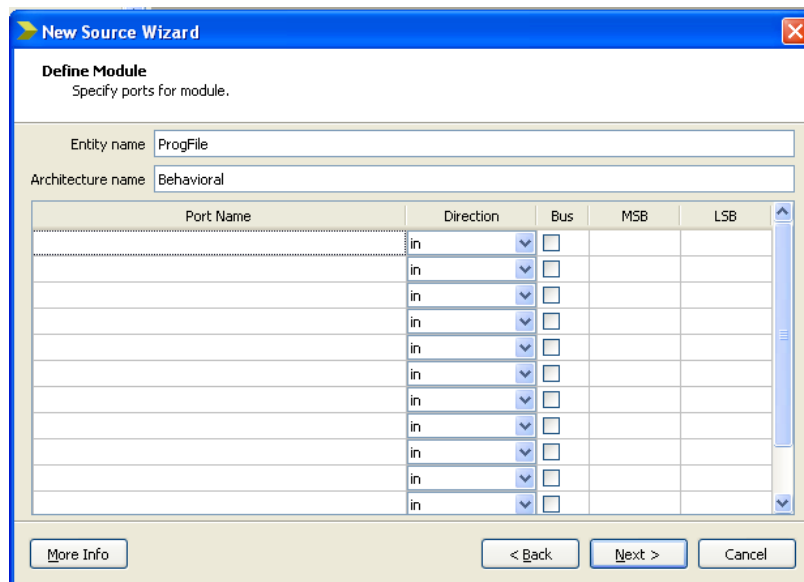


Рисунок 4.7 – Вибір входів та виходів для пристрою, що проектується

На останньому кроці створення нового проекту з'явиться діалогове вікно з детальною інформацією про створений модуль (рис. 4.8). Для завершення створення нового проекту потрібно натиснути кнопку «Finish».

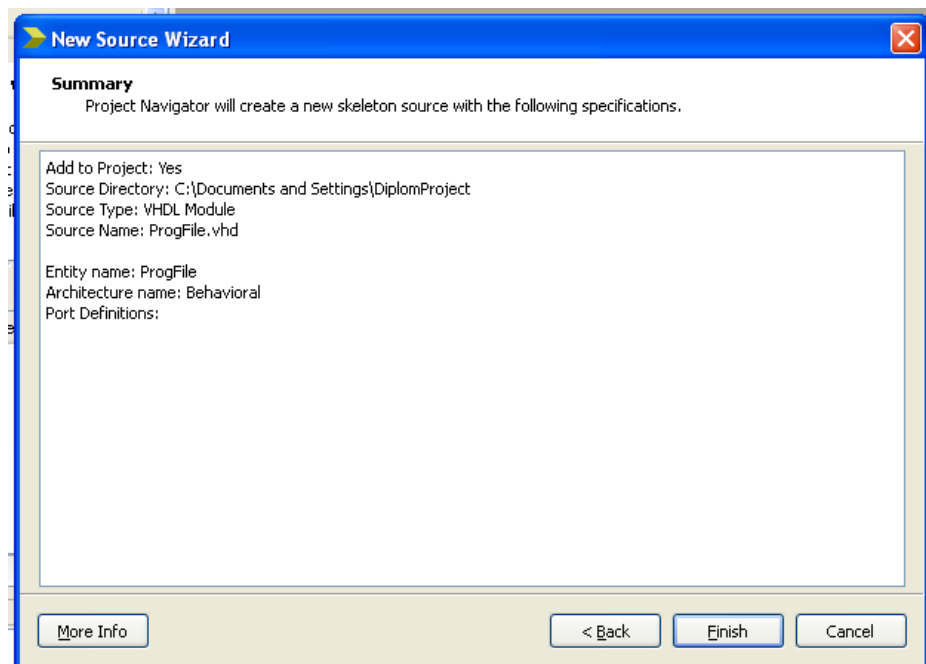


Рисунок 4.8 – Інформація про створений файл

Новий створений файл буде автоматично відкрито у новій вкладці у WEB Pack ISE.

Наступним етапом створення пристрою буде етап опису. Робоче поле для опису пристрою рис. 4.9.

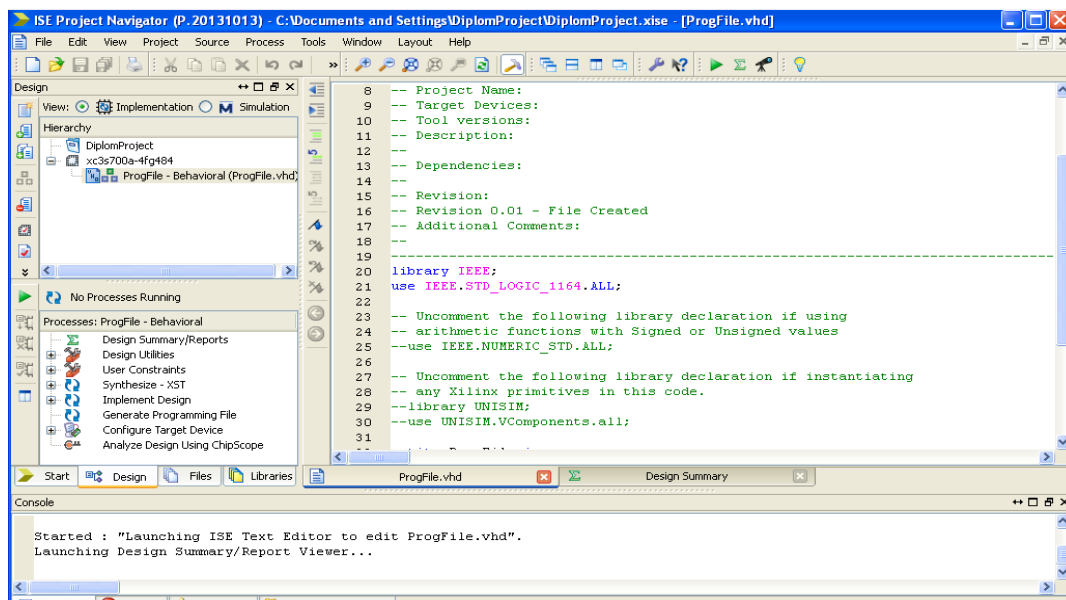


Рисунок 4.9 – Робоче поле для опису проекту

Після того, як було закінчено опис пристрою, що проектується, потрібно визначити та зв'язати виходи та входи пристрою із виходами та входами ПЛІС. Якщо на цьому етапі виникають труднощі, то користувач має



змогу ознайомитися з офіційною документацією Spartan-3 офіційному сайті компанії Xilinx. Для того, щоб вручну налаштувати входи та виходи, потрібно обрати пункт «Sources» (рис. 4.10). У вкладці «Processes» потрібно обрати пункт «User Constrains». Далі потрібно обрати пункт меню «I/O Pin Planning (PlanAhead) – Post-Synthesis» (рис. 4.11).

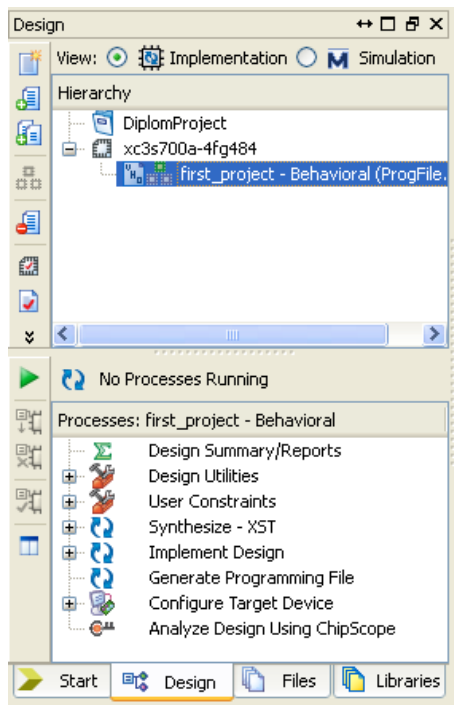


Рис. 4.10 – Вкладка «Sources»

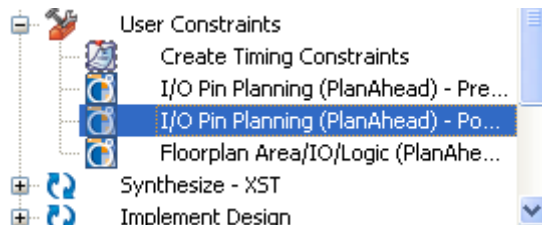


Рисунок 4.11 – Вкладка «Assign Package Pins»

Користувачеві буде запропоновано створити текстовий файл із розширенням «.UCF» (рис.4.12). Цей файл можна використовувати для подальшої зміни параметрів входів/виходів пристрою, що проектується.

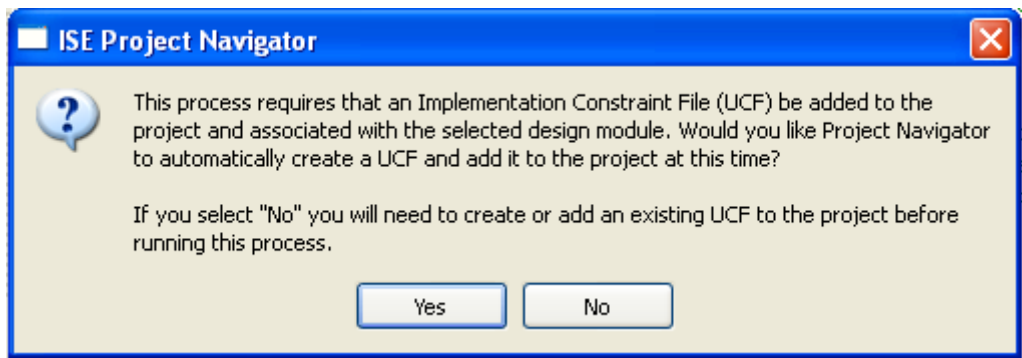


Рисунок 4.12 – Створення файлу із розширенням «.UCF»

Перед користувачем з'явиться вікно додатку PlanAhead 14.7 (даний програмний компонент присутній у версії САПР WEB Pack ISE 14.7). У правій верхній частині можна побачити кристал ПЛІС Spartan – 3AN. Структуру кристалу тут зображено у вигляді матриці. Тут потрібно обрати вкладку «I/O Ports» (рис. 4.13).

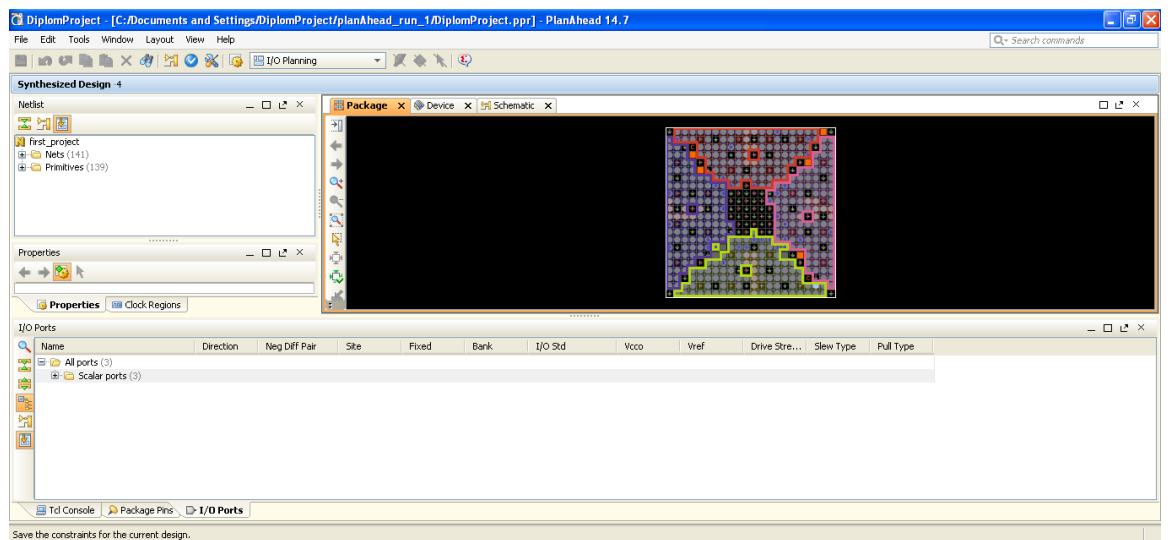


Рисунок 4.13 – Головне вікно програмного додатку PlanAhead 14.7

У таблиці для налаштування параметрів входів та виходів нашого пристрою (рис. 4.14), що проектується, необхідно обрати деякі важливі параметри. Наприклад комірки на платі (E12) та I/O Standart.

Name	Direction	Neg Diff Pair	Site	Fixed	Bank	I/O Std	Vcco	Vref	Drive Stre...	Slew Type	Pull Type
All ports (3)											
Scalar ports (3)											
clk	Input			<input type="checkbox"/>		default (LVCMOS25)					NONE
led1	Output			<input type="checkbox"/>		default (LVCMOS25)	2.500		12 SLOW		NONE
nreset	Input			<input type="checkbox"/>		default (LVCMOS25)					NONE

#### Рисунок 4.14 – Встановлення параметрів для входів та виходів пристрою

Після завершення налаштувань потрібно зберегти вказані параметри. Після цього програмний додаток PlanAhead 14.7 можна закрити.

Наступним етапом буде компіляція проекту. Для цього на вкладці «Design» потрібно обрати пункт «Implement Design». Після цього потрібно обрати пункт «Generate Programming File». Потім потрібно у вкладці «Configure Target Device» обрати пункт «Manage Configuration Project (IMPACT)» та двічі натиснути на нього мишкою (рис. 4.15).

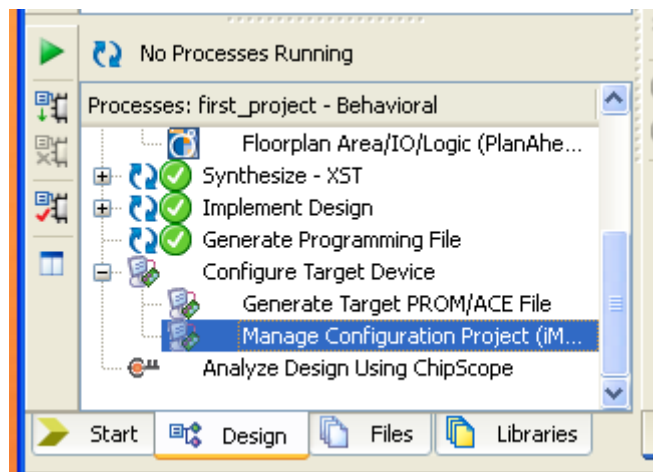


Рисунок 4.15 – Компіляція проекту

Користувачеві відкриється ще один додатковий програмний додаток «ISE IMPACT» (рис. 4.16). Для початку потрібно згенерувати PROM файл та обрати необхідні налаштування (рис. 4.17) та натиснути кнопку «ОК».

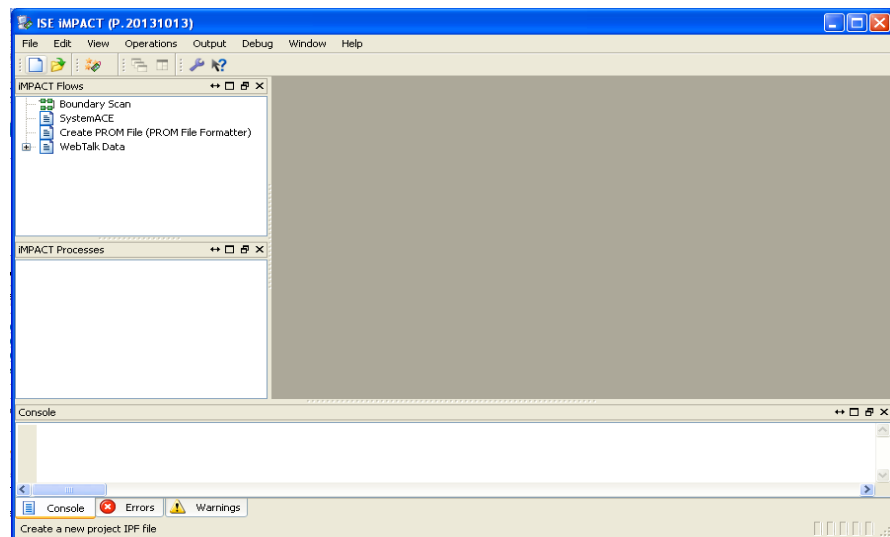


Рисунок 4.16 – Програмний додаток ISE IMPACT

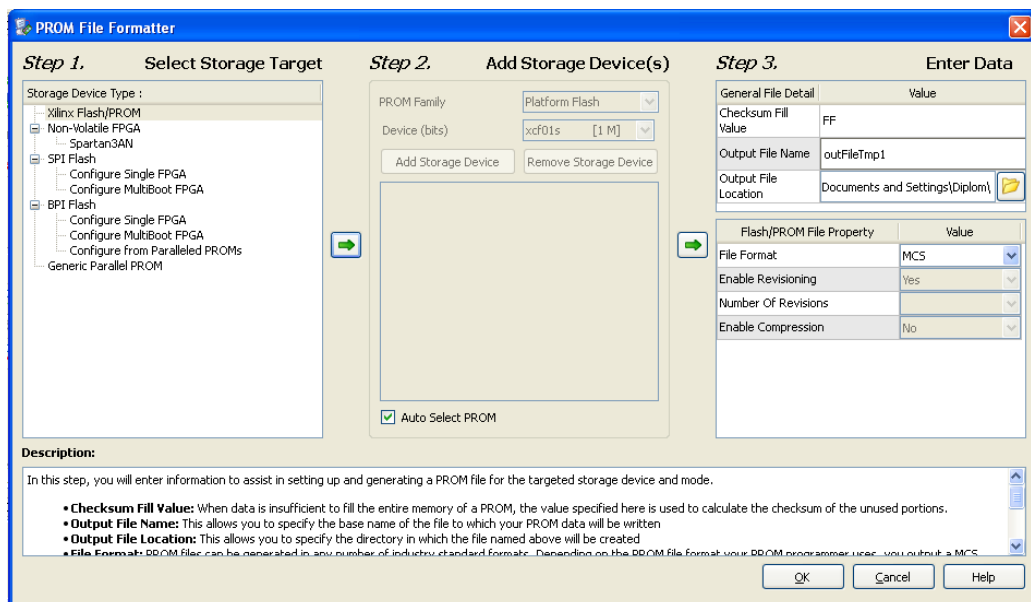


Рисунок 4.17 – Генерація PROM файлу

Далі потрібно обрати та запустити генерацію створеного bit-файлу для його завантаження на ПЛІС. Після успішного завантаження bit-файлу з'явиться відповідне вікно (рис. 4.18). Тут потрібно мишкою натиснути пункт «Generate File...». Якщо bit-файл сформовано успішно, то у вікні (рис. 4.18) з'явиться напис «Generate Succeeded» (рис. 4.19).

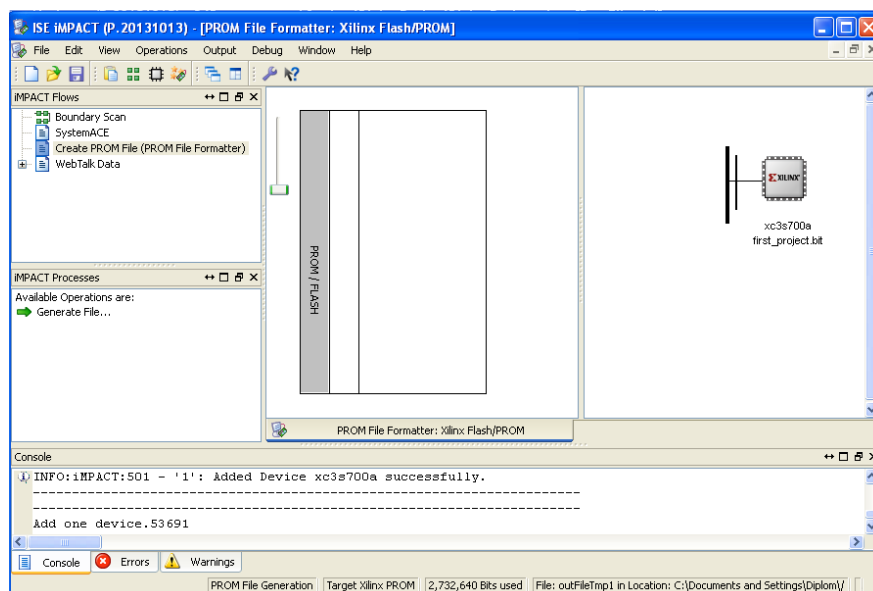


Рисунок 4.18 – Завантаження bit-файлу

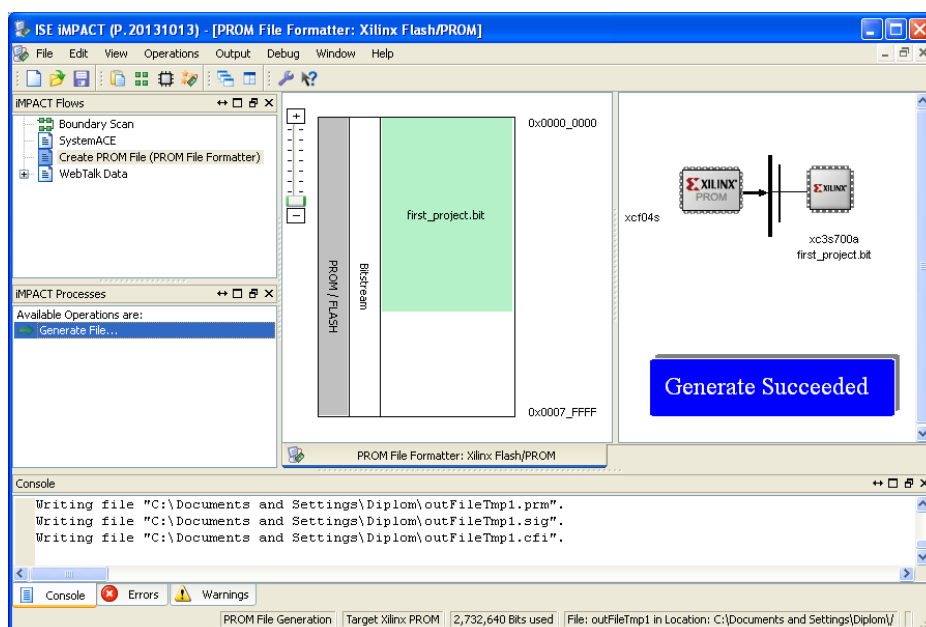


Рисунок 4.19 – Успішне створення файлу із розширенням .bit

Після цього розробнику потрібно перевірити налаштування кабелю для завантаження створеного проекту на ПЛІС (рис. 4.20). Для цього потрібно обрати пункт «Boundary Scan» (рис. 4.18). Далі потрібно натиснути правою кнопкою миші по робочому полі та обрати пункт «Initialize Chain».

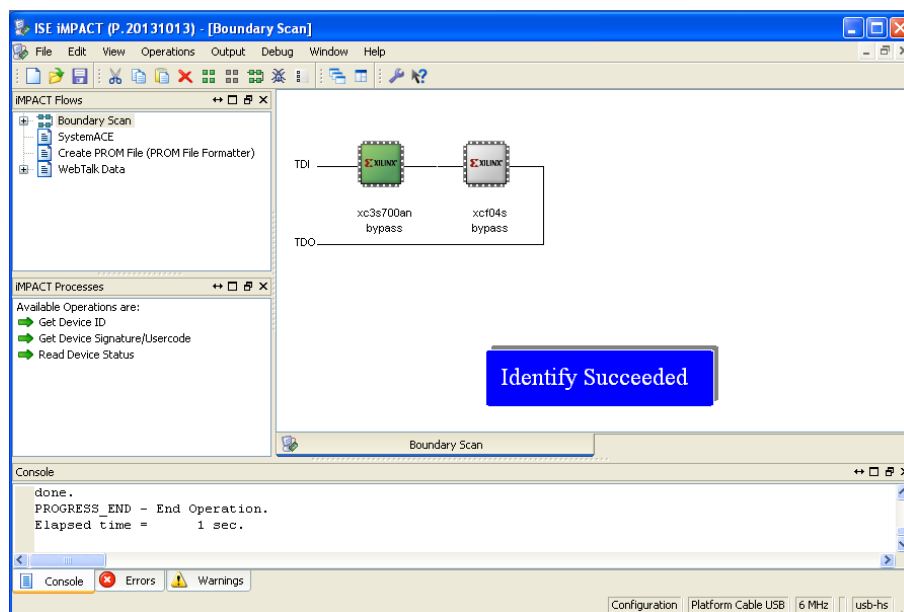


Рисунок 4.20 – Підготовка завантаження файлу із розширенням.bit на ПЛІС

Тепер потрібно завантажити проект на ПЛІС. На значку xc3s700an потрібно натиснути правою кнопкою миші та обрати пункт «Program FPGA only» (рис. 4.21).

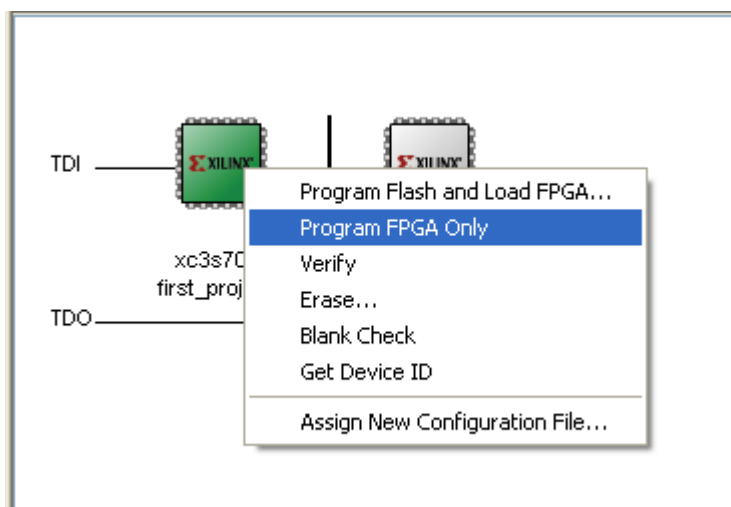


Рисунок 4.21 – Процес завантаження проекту на ПЛІС

Після цього створений проект буде успішно завантажено на ПЛІС (рис. 4.22).

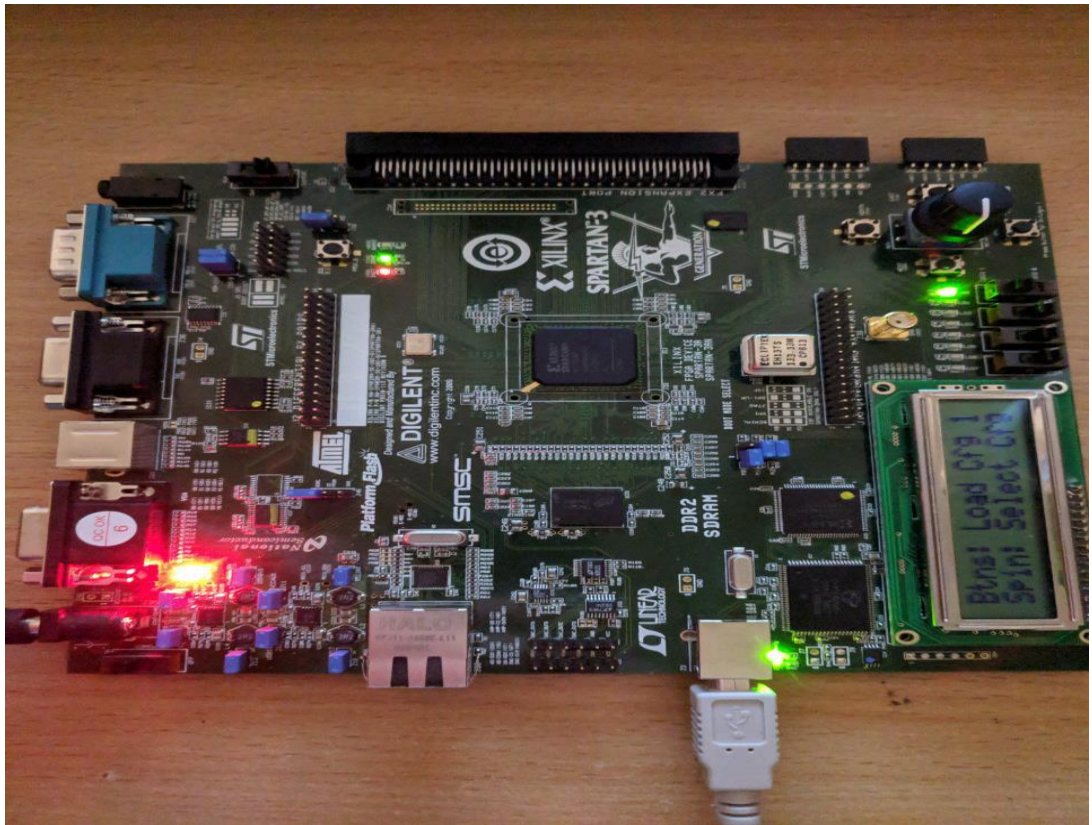


Рисунок 4.22 – Завантажений проект на ПЛІС

#### **4.3 Опис реалізації обробки даних на ПЛІС для моніторингу об'єктів критичного застосування**

На сьогодні є велика кількість галузей виробництва, де необхідно забезпечувати високий рівень надійності та безпеки використання об'єктів критичного застосування.

Безпека та надійність використання таких об'єктів критичного застосування залежить від рівня автоматизованості певної промисловості. Рівень автоматизованості залежить від таких показників, як безвідмовність роботи обчислювальних систем, довершеність архітектурних та програмних рішень, відмовостійкість програмно-апаратних частин, час ремонту тощо.

Моніторинг об'єктів критичного застосування відбувається на основі обробки сигналів від об'єктів, які перебувають у системі моніторингу. Цифрова обробка сигналів – це спосіб обробки сигналів на основі чисельних методів із використанням цифрової обчислювальної техніки. При цифровій

обробці сигналів використовуються чисельні операції над великими числами. Це такі операції як множення.

Паралельні алгоритми обробки даних використовуються у багатьох сферах наукової діяльності, а саме в цифровій обробці сигналів, криптографії тощо. При реалізації криптографічних протоколів доводиться мати справу з бінарними многочленами великих степенів (близько  $10^2$ - $10^5$ ). Тому пришвидшення операції множення таких чисел є важливою задачею.

Для вирішення питання збільшення ефективності роботи паралельних алгоритмів обробки даних є доречним застосування програмно-апаратного способу реалізації такого алгоритму. Такий підхід ґрунтується на апаратному прискоренні певних частин алгоритму, який можна реалізувати на базі програмованої логічної інтегральної схеми (ПЛІС).

Оскільки логіка роботи ПЛІС не визначається при виготовленні, а задається за допомогою програмування схеми, то користувач має змогу швидко налаштувати обчислювальну структуру для реалізації алгоритму обробки інформації. Такий підхід дозволяє забезпечити зручне використання у програмуванні ПЛІС, широку функціональність реалізованих алгоритмів.

Для програмно-апаратної реалізації спеціалізованого цифрового пристрою використовувався пристрій XC3S700A Spartan-3 сімейства FPGA фірми Xilinx. Основні характеристики даного пристрою наведено у таблиці 4.1.

Таблиця 4.1 – Основні особливості пристрою XC3S700A Spartan-3

Пристрій	Кількість логічних елементів, шт.	Множина CLB			
		Рядків, шт.	Стовпців, шт.	CLBs, шт.	Slices, шт.
XC3S700A	13 248	48	32	1 472	5 888



Як тестовий об'єкт критичного застосування було використано програмний додаток створений за допомогою .Net Framework та мови програмування С# (рис. 4.23).

Имя образа	PID	Имя сессии	№ сервиса	Память	Состояние	Пользователь	Время ЦП	Заголовок окна
System Idle Process	0	Services	0	0 KB	Unknown	NT AUTHORITY\СИСТЕМА	0.34.30	Н/Д
System	4	Services	0	1 076 KB	Unknown	Н/Д	0.00.06	Н/Д
lsass.exe	96	Services	0	12 028 KB	Unknown	Н/Д	0.00.00	Н/Д
csrss.exe	404	Services	0	980 KB	Unknown	Н/Д	0.00.00	Н/Д
csrss.exe	596	Services	0	4 720 KB	Unknown	Н/Д	0.00.03	Н/Д

Рисунок 4.23 – Тестовий об'єкт критичного застосування

Обробка параметрів об'єкту критичного застосування відбувається на основі цифрової обробки сигналів.

Для цифрової обробки сигналів для моніторингу об'єктів критичного застосування використовуються їх спеціальні перетворення. Під час цих перетворень відбувається обробка даних, за допомогою основних арифметичних операцій. Використання алгоритмів прискореного множення дозволить опрацьовувати сигнали швидше, що підвищить ефективність моніторингу об'єктів критичного застосування. Один із таких алгоритмів прискореного множення працює на основі рекурсивного розбиття множників. Наприклад, нехай  $A$  і  $B$  – два  $n$ -розрядних двійкових числа. Розбиваючи їх записи на дві частини довжини  $(n/2)$ , отримаємо

$$A * B = \left( A_1 2^{\lfloor \frac{n}{2} \rfloor} + A_0 \right) \left( B_1 2^{\lfloor \frac{n}{2} \rfloor} + B_0 \right) =$$

$$A_1 B_1 2^n + (A_1 B_0 + A_0 B_1) 2^{\lfloor \frac{n}{2} \rfloor} + A_0 B_0$$

Оскільки множення на степінь двійки і додавання – швидкі операції, тому робота зводиться до швидкого обчислення трьох білінійних форм  $C_1 =$

$A_1B_1$ ,  $C_2 = A_1B_0 + A_0B_1$ ,  $C_3 = A_0B_0$ . Для цього достатньо обчислити білінійні форми  $C_1$ ,  $C_3$  та  $D = (A_1 + A_0) * (B_1 + B_0)$ . При цьому  $C_2 = D - C_1 - C_3$ . Таким чином, задача множення  $n$ -розрядних двійкових чисел зводиться до трьох задач множення  $(n/2)$ -розрядних двійкових чисел і декількох операцій додавання та віднімання не більш, ніж  $n$ -розрядних двійкових чисел. Виграш в продуктивності алгоритму виходить за рахунок заміни «трудомістких» операцій множення на операції додавання і зсуву.

Нехай  $n = 2^k$  (вважаємо, що  $n$  – це деяка степінь двійки). Оскільки алгоритм швидкого множення працює рекурсивно, то на  $i$ -тому рівні рекурсії розмір даних для обрахунку складає  $n/2^i$ . Кількість задач складає  $3^i$ . Час роботи  $3^i * c * n/2^i$ . Отже, щоб порахувати загальний час роботи алгоритму, потрібно просумувати час роботи на одному рівні рекурсії  $k$  разів (потрібно обрахувати суму геометричної прогресії)

$$\sum_{i=0}^k 3^i * c * n/2^i = c * n * \sum_{i=0}^k \left(\frac{3}{2}\right)^i = c * n * \sum_{i=0}^{\log_2 n} \left(\frac{3}{2}\right)^i$$

Оскільки знаменник арифметичної прогресії більша за одиницю ( $3/2$ ), то сума зростає так, як її останній член

$$\frac{3^{\log_2 n}}{2^{\log_2 n}}$$

Отже, складність роботи алгоритму складає

$$c * n * O\left(\frac{3^{\log_2 n}}{2^{\log_2 n}}\right)$$

Цей вираз можна скоротити, використовуючи основні правила роботи із логарифмами. Використаємо основну логарифмічну тотожність, а саме  $a^{\log_a b} = b$ . Тому даний вираз часу роботи можна скоротити

$$c * n * O\left(\frac{3^{\log_2 n}}{2^{\log_2 n}}\right) = c * O\left(n * \frac{3^{\log_2 n}}{n}\right) = c * O(3^{\log_2 n})$$

Далі можна скоротити константу «с» за яку виконується одна арифметична дія в алгоритмі швидкого множення. Занесемо її до  $O(3^{\log_2 n})$  і потім, за правилами використання  $O$ -символіки, константу «с» можна скоротити

$$O(3^{\log_2 n})$$

Згідно правил роботи з логарифмами, вираз  $3^{\log_2 n}$  можна переписати як  $n^{\log_2 3}$

$$O(3^{\log_2 n}) = O(n^{\log_2 3}) \approx O(n^{1,584})$$

Приведені обчислення дійсно показують, що алгоритм швидкого множення має складність  $O(n^{1,584})$ , тоді як алгоритм класичного множення чисел має складність  $O(n^2)$ .

Для збільшення швидкодії рекурсивні частини алгоритму швидкого множення можна виконувати у трьох потоках, які будуть виконуватися паралельно.

Тому кількість задач на кожному рівні буде у три рази меншою, ніж при послідовному виконанні алгоритму  $T_{\text{послідовне}}$ :

$$T_{\text{послідовне}}(n) = 3 * T\left(\frac{n}{2}\right) + cn$$

Час роботи паралельного алгоритму  $T$  складає:

$$T(n) = T\left(\frac{n}{2}\right) + cn$$

Тоді асимптотичне прискорення алгоритму буде складати

$$S = \frac{T_{\text{послідовне}}}{T} = O\left(\frac{n^{\log_2 3}}{n}\right) \approx O(n^{0,585})$$

Отримані результати наведено у таблиці 4.2.

Таблиця 4.2 –Результати роботи обробки даних.

Кількість цифр у числі, шт.	Класичний алгоритм множення, с	Послідовний алгоритм, с.	Паралельний алгоритм, с.
1024	0.16	0.13	0.022
2048	0.50	0.40	0.056
4096	1.51	1.20	0.16
8192	4.59	3.64	0.45
16384	13.88	11.0	1.33
32768	41.89	33.2	3.89
65536	126.18	100	11.5

У таблиці 4.3 наведено результати використання елементів Slice та LUT ПЛІС Spartan-3.

Таблиця 4.3 – Використання елементів ПЛІС Spartan-3 для обчислення даних

	Кількість задіяних Slice, %	Кількість задіяних 4-входових LUTs, %
Алгоритм прискореного множення	1 248 із 5 888, (21%)	1404 із 4704, (30%)
Стандартний алгоритм множення	1 824 із 5 888, (31%)	2336 із 4704, (50%)

#### 4.4 Висновки до четвертого розділу

Під час розробки проекту реалізації цифрового пристрою на базі ПЛІС фірми Xilinx було визначити наступні етапи проектування: вибір фірми, сімейства та типу ПЛІС; створення нового проекту у спеціальному САПР WEB Pack ISE фірми Xilinx; обговорення опису пристрою, який буде проектуватися. синтез пристрою; етап функціонально моделювання; розміщення і трасування проекту на кристалі; часове моделювання; процес програмування ПЛІС; завантаження проекту на кристал.

Проект було створено за допомогою САПР ISE Design ISE модифікація WebPACK ISE фірми Xilinx. WebPACK ISE підтримує реалізацію усіх етапів створення та проектування спеціалізованого пристрою.

Для вирішення питання збільшення ефективності процесу моніторингу об'єктів критичного застосування є доречним застосування програмно-апаратного способу реалізації алгоритму обробки таких даних. Такий підхід ґрунтується на апаратному прискоренні певних частин алгоритму, який можна реалізувати на базі програмованої логічної інтегральної схеми (ПЛІС). Високу продуктивність та гнучкість перепрограмування обчислювального засобу, що реалізує алгоритм моніторингу об'єктів критичного застосування на базі ПЛІС, обумовлено можливістю швидкої модифікації роботи

алгоритму шляхом перепрограмування ПЛІС за допомогою спеціальних САПР.

## ВИСНОВКИ

В магістерській дисертації досліджено та проаналізовано архітектурно-структурну організацію ПЛІС. Розглянуті сучасні напрямки розвитку ПЛІС-технологій. Окреслено актуальність та проблеми створення обчислювальних пристроїв на їхній основі. Проаналізовано засоби та методи, архітектурно-структурні рішення для вирішення проблеми моніторингу об'єктів критичного застосування на основі використання ПЛІС для реалізації таких систем.

Було спроектовано спеціалізований цифровий пристрій паралельної обробки даних на базі ПЛІС, який виконує розподілені обчислення при обробці даних для моніторингу об'єктів критичного застосування. Алгоритм, разом із системою апаратного забезпечення на основі ПЛІС, може бути використано у мікропроцесорній та обчислювальній техніці, при моніторингу об'єктів критичного застосування, в навчальних цілях на факультеті прикладної математики на кафедрі СПіСКС, в математичних дослідженнях тощо.

Високу продуктивність та гнучкість перепрограмування обчислювального засобу, що реалізує алгоритм моніторингу об'єктів критичного застосування на базі ПЛІС, обумовлено можливістю швидкої модифікації роботи алгоритму шляхом перепрограмування ПЛІС за допомогою спеціальних САПР. До недоліків реалізації такого обчислювального засобу на ПЛІС можна віднести обмеженість кількості логічних елементів.

Для проектування спеціалізованого цифрового пристрою моніторингу об'єктів критичного застосування використовувалась САПР ISE Design Suite компанії Xilinx. Перевагами даної САПР є достатній набір інструментів та зручний користувацький інтерфейс для виконання всіх етапів створення цифрового пристрою на базі ПЛІС.

## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

- 1) Типові цифрові схеми комп'ютерів.[Електронний ресурс].-2015. Режим доступу: [https://msn.khnu.km.ua/pluginfile.php/147595/mod\\_resource/content/2/Lekz10.pdf](https://msn.khnu.km.ua/pluginfile.php/147595/mod_resource/content/2/Lekz10.pdf). – Дата доступу: вересень 2018.
- 2) Методи обробки даних на ПЛІС.[Електронний ресурс].-2015. Режим доступу: [http://ela.kpi.ua/bitstream/123456789/19096/1/Klymenko\\_aref.pdf](http://ela.kpi.ua/bitstream/123456789/19096/1/Klymenko_aref.pdf). – Дата доступу: вересень 2018.
- 3) Все про мікроконтролери.[Електронний ресурс].-2015. Режим доступу: <http://www.maria-online.com/electronics/article.php?lg=uk&q=Мікроконтролер> . – Дата доступу: вересень 2018.
- 4) Говорущенко Т.О. Типові цифрові схеми комп'ютерів.[Текст] – Харків, 2015.-стор. 12.
- 5) Основи мікропроцесорної логіки [Електронний ресурс].-2009. Режим доступу: [http://www.tverhthk.ru/library/predmets/pc\\_systems/Osnovy\\_mikroprocesornoj\\_tehniki\\_2009.pdf](http://www.tverhthk.ru/library/predmets/pc_systems/Osnovy_mikroprocesornoj_tehniki_2009.pdf). – Дата доступу: вересень 2018.
- 6) Мікропроцесори. Структура мікропроцесорів.[Електронний ресурс].-2010. Режим доступу: <http://www.yaklass.ru/materiali?chtid=459&mode=cht>. – Дата доступу: вересень 2018.
- 7) Однокристальні мікрокомп'ютери.[Електронний ресурс].-2012. Режим доступу: <http://sh.instone.com.ua/tema2.6.php>. – Дата доступу: вересень 2018.
- 8) Топ-5 найбільш небезпечніших підприємств України.[Електронний ресурс].-2012. Режим доступу: <http://firtka.if.ua/blog/view/top5-najbils-nebezpresnih-virobnictv-v-ukraini4013>. – Дата доступу: вересень 2018.
- 9) Потенційно небезпечні виробництва та вимоги до їх розміщення.[Електронний ресурс].-2012. Режим доступу: <https://studopedia.org/4-20102.html>. – Дата доступу: вересень 2018.



10) Мартинюк Т.Б., Кіріяченко А.О. Комірка однорідної структури.[Текст] – Харків, 2008. – стор. 34.

11) Характеристики ПЛІС. [Електронний ресурс].-2015. Режим доступу: [https://msn.khnu.km.ua/pluginfile.php/147595/mod\\_resource/content/2/Lekz10.pdf](https://msn.khnu.km.ua/pluginfile.php/147595/mod_resource/content/2/Lekz10.pdf). – Дата доступу: вересень 2018.

12) Complex Programmable Logic Device. [Електронний ресурс].-2015. Режим доступу: <https://www.xilinx.com/products/silicon-devices/cpld/cpld.html>. – Дата доступу: жовтень 2018.

13) Spartan-3A FPGA EK Family Data Sheet. [Електронний ресурс].-2015. Режим доступу: <http://datasheet.elcodis.com/pdf2/102/36/1023636/xc3s1400an.pdf>. – Дата доступу: жовтень 2018.

14) Altera FLEX 8000. [Електронний ресурс].-2015. Режим доступу: <https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ds/archives/dsf8k.pdf>. – Дата доступу: жовтень 2018.

15) SOC FPGA. [Електронний ресурс].-2010. Режим доступу: [https://www.altera.com/content/dam/altera-www/global/en\\_US/pdfs/literature/ab/ab1\\_soc\\_fpga.pdf](https://www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/ab/ab1_soc_fpga.pdf). – Дата доступу: жовтень 2018.

16) What is SOC. [Електронний ресурс].-2010. Режим доступу: <https://www.digitalcitizen.life/soc-system-on-chip>. – Дата доступу: жовтень 2018.

17) Зотов В.Ю. Проектування цифрових пристроїв на основі ПЛІС фірми Xilinx в САПР WebPACK ISE. [Текст] – Москва, 2003, стр 11 – 17.

18) Difference between CPLD and FPGA. [Електронний ресурс].-2010. Режим доступу: <http://www.differencebetween.net/technology/difference-between-fpga-and-cpld/>. – Дата доступу: жовтень 2018.

19) Intel FPGAs. [Електронний ресурс].-2010. Режим доступу: <https://www.intel.com/content/www/us/en/products/programmable/fpga.html>. – Дата доступу: жовтень 2018.

20) 7 Series FPGAs Xilinx: Overview. [Электронный ресурс].-2010. Режим доступа: [https://www.xilinx.com/support/documentation/data\\_sheets/ds180\\_7Series\\_Overview.pdf](https://www.xilinx.com/support/documentation/data_sheets/ds180_7Series_Overview.pdf). – Дата доступа: листопад 2018.

21) Spartan-6 family overview. [Электронный ресурс].-2010. Режим доступа: [https://www.xilinx.com/support/documentation/data\\_sheets/ds160.pdf](https://www.xilinx.com/support/documentation/data_sheets/ds160.pdf). – Дата доступа: листопад 2018.

22) XA Spartan-7 Automotive family overview. [Электронный ресурс].-2010. Режим доступа: [https://www.xilinx.com/support/documentation/data\\_sheets/ds171-xa-spartan7-overview.pdf](https://www.xilinx.com/support/documentation/data_sheets/ds171-xa-spartan7-overview.pdf). – Дата доступа: листопад 2018.

23) Vivado VHLS. [Электронный ресурс].-2016. Режим доступа: [http://plis.ru/docum/sredstva\\_razrabotki\\_i\\_ip\\_yadra\\_otladochnie\\_sredstva/vivado\\_novoe\\_sredstvo\\_razrabotki/vivado\\_hls\\_high\\_level\\_synthesis\\_novaya\\_sapra\\_xilinx](http://plis.ru/docum/sredstva_razrabotki_i_ip_yadra_otladochnie_sredstva/vivado_novoe_sredstvo_razrabotki/vivado_hls_high_level_synthesis_novaya_sapra_xilinx). – Дата доступа: листопад 2018.

24) 9 REASONS WHY THE VIVADO DESIGN SUITE ACCELERATES DESIGN PRODUCTIVITY. [Электронный ресурс].-2016. Режим доступа: [https://www.xilinx.com/publications/prod\\_mktg/vivado/Vivado\\_9\\_Reasons\\_Background.pdf](https://www.xilinx.com/publications/prod_mktg/vivado/Vivado_9_Reasons_Background.pdf). – Дата доступа: листопад 2018.

25) ISE Design Suite 14. [Электронный ресурс].-2016. Режим доступа: [https://www.xilinx.com/publications/prod\\_mktg/ISE\\_sellsheet.pdf](https://www.xilinx.com/publications/prod_mktg/ISE_sellsheet.pdf). – Дата доступа: листопад 2018.

26) New Project Wizard – Device Properties Page. [Электронный ресурс].-2016. Режим доступа: [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx11/pn\\_db\\_npw\\_device\\_properties.htm](https://www.xilinx.com/support/documentation/sw_manuals/xilinx11/pn_db_npw_device_properties.htm). – Дата доступа: листопад 2018.

## ДОДАТКИ