

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Електронної та біомедичної інженерії
(повна назва)

Кафедра Мікроелектроніки, електронних приладів та пристроїв
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти другий (магістерський)

Система контролю та діагностики транспортних засобів на базі програмованих
вентильних матриць
(тема)

Виконав:
студент 2 курсу, групи МНПм-21-1
Деркач Д.М.
(прізвище, ініціали)

Спеціальність 153 Мікро- та наносистемна техніка
(код і повна назва спеціальності)
Тип програми освітньо-професійна
Освітня програма Мікро- та наноелектронні
прилади і пристрої
(повна назва освітньої програми)

Керівник проф. Грицунов О.В.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Ігор БОНДАРЕНКО
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Електронної та біомедичної інженерії

Кафедра Мікроелектроніки, електронних приладів та пристроїв

Рівень вищої освіти другий (магістерський)

Спеціальність 153 Мікро- та наносистемна техніка
(код і повна назва)

Освітня програма Мікро- та наноелектронні прилади і пристрої
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Деркачу Дмитру Максимовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Система контролю та діагностики транспортних засобів на базі програмованих вентильних матриць

затверджена наказом по університету від 14.11 2022 р. № 1473Ст

2. Термін подання студентом роботи до екзаменаційної комісії 10.12 2022 р.

3. Вихідні дані до роботи напруга живлення – 12 В, діапазон робочих температур – 25...+30°C, вживана потужність – до 1 Вт, вид сенсорів – цифрові.

4. Перелік питань, що потрібно опрацювати в роботі

1. ПРОГРАМОВАНІ КОРИСТУВАЧЕМ ВЕНТИЛЬНІ МАТРИЦІ

2. СИСТЕМИ КЕРУВАННЯ АВТОМОБІЛЕМ

3. ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ТА ДІАГНОСТИКИ АВТОМОБІЛЯ НА ПЛІС

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

Схеми: схема електрична структурна, схема електрична принципова;

Кресленики: кресленик загального виду, друкована плата;

Слайди: 20 одиниць _____

КАЛЕНДАРНИЙ ПЛАН

	Назва етапів роботи	Терміни виконання етапів	Примітка
	Отримання завдання	21.11.2022	
	Огляд інформаційних джерел	21.11–25.11	
	Створення проекту за допомогою САПР	26.11–28.11	
	Розробка конструкції пристрою	29.11–30.11	
	Пояснювальна записка	01.12–04.12	
	Підготовка презентації	05.12–08.12	
	Рецензування, нормоконтроль	09.12–10.12	
	Здача роботи на кафедрі	10.12.2022	

Дата видачі завдання 21 листопада 2022 р.

Студент _____

(підпис)

Керівник роботи _____ проф. Грицунов О.В. _____

(підпис)

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка містить 46 сторінок, 8 рисунків, 5 додатків, 15 використаних джерел.

ТРАНСПОРТНИЙ ЗАСІБ, СИСТЕМА КОНТРОЛЮ, БЛОК УПРАВЛІННЯ, КОНТРОЛЕР, МІКРОПРОЦЕСОР, ПРОГРАМОВАНА ЛОГІКА.

Об'єкт дослідження – методи і засоби реалізації систем контролю параметрів транспортних засобів.

Мета роботи – дослідження перспективних засобів реалізації систем контролю та управління для транспортних засобів широкого застосування, оптимізація застосовуваної компонентної бази, удосконалення та розробка елементів системи контролю.

Метод досліджень – автоматизоване проектування на ЕОМ.

Запропонована система контролю параметрів транспортних засобів може являти практичний інтерес, як для розробників автомобільних пристроїв, так і для фахівців в галузі промислових систем контролю та керування, в перспективних системах контролю стану та режимів експлуатації транспортних засобів.

ABSTRACT

The explanatory note contains 46 pages, 8 figures, 5 appendices, 15 used sources.

VEHICLE, CONTROL SYSTEM, CONTROL UNIT, CONTROLLER, MICROPROCESSOR, PROGRAMMED LOGIC

The object of research is the methods and means of implementation of vehicle parameter control systems.

The purpose of the work is to research prospective means of implementing control and management systems for vehicles of wide use, optimization of the applied component base, improvement and development of control system elements.

The research method is computer-aided design.

The proposed vehicle parameter control system can be of practical interest both for automotive device developers and for specialists in the field of industrial control and management systems, in prospective control systems for the condition and operating modes of vehicles.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	7
ВСТУП	8
1 ПРОГРАМОВАНІ КОРИСТУВАЧЕМ ВЕНТИЛЬНІ МАТРИЦІ	9
1.1 Історія створення FPGA	10
1.2 Принципи побудови та архітектури ПЛІС	13
1.2.1 Базова архітектура	14
1.2.2 Ускладнені архітектури	14
1.2.3 Логічні блоки	16
Висновки по розділу	23
2 СИСТЕМИ КЕРУВАННЯ АВТОМОБІЛЕМ	25
2.1 Система датчиків автомобіля	25
2.2 Складові електронної системи контролю	26
2.3 Робота блоку електронного керування автомобілем	30
Висновки по розділу	33
3 ПРОЄКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ТА ДІАГНОСТИКИ АВТОМОБІЛЯ НА ПЛІС	34
3.1 Властивості та можливості FPGA	35
3.2 Синтез загальної структури системи на ПЛІС	37
3.3 Проєктування конструкції блока контролю та управління	40
Висновки по розділу	42
ВИСНОВКИ	44
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	45
ДОДАТКИ	47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

- БМК – базовий матричний кристал
- БУ – блок управління
- ЛЕ – логічний елемент
- МК – мікроконтролер
- ПЛК – програмований логічний контролер
- САПР – система автоматизовано проектування
- ТЗ – транспортний засіб
- AHDL – Altera Hardware Description Language
- CAN – шина передачі даних в транспортних засобах
- CISC – архітектура, що реалізує повну систему команд (Complicated Instruction Set Computer)
- CPLD – комплексний програмований логічний пристрій (Complex Programmable Logic Devices)
- DSP – цифровий сигнальний процесор (Digital Signal Processor)
- FPGA – програмована вентилятна матриця (Field Programmable Gate Array)
- JTAG – Joint Test Action Group
- PIO – Parallel Input/Output
- SDRAM – синхронна динамічна пам'ять з довільним доступом (Synchronous Dynamic Random Access Memory)
- SOPC – System on Programmable Chip
- SRAM – статична пам'ять з довільним доступом (Static Random Access Memory)
- RFID – радіочастотна система ідентифікації
- RISC – архітектура, що реалізує скорочену систему команд (Computer Reduced Instruction Set) CAE, САМ – системи проектування в електроніці
- VHDL, Verilog – мови програмування

ВСТУП

Автомобільна індустрія зараз знаходиться у стадії глибокого реформування, пов'язаного з бурхливим розвитком електромобілів та передбачуваним витісненням ними машин з двигунами внутрішнього згоряння.

Паралельно не менш інтенсивно розвивається автомобільна електроніка; мета її розвитку вже зрозуміла: остаточне витіснення ручного керування транспортним засобом з передачею усіх функцій бортовому комп'ютеру. Втім, останнім часом на ринку авто електроніки виникла проблема, пов'язана з суттєвою нестачею комплектуючих для транспортних засобів.

Це зумовлено різними причинами. За останні роки, в зв'язку з пандемією, дещо знизилось виробництво компонентів автоелектроніки, що обумовлено проблемами логістики, обмеженням трудових ресурсів та хворобою наявних працівників.

В той же час суттєво зростає кількість електронних компонентів, що застосовуються в транспортних засобах, росте потреба в високопродуктивних, гнучких системних продуктах, що можуть забезпечити швидкий процес реінжинірингу існуючих електронних блоків управління.

Мета даної кваліфікаційної роботи – дослідження перспективних методів і засобів реалізації в галузі автоелектроніки, пошук оптимальних рішень і розробка структури блоку керування та його базових елементів на сучасній компонентній базі.

1 ПРОГРАМОВАНІ КОРИСТУВАЧЕМ ВЕНТИЛЬНІ МАТРИЦІ

Програмована користувачем вентиляна матриця (англ. Field-Programmable Gate Array, FPGA) – це напівпровідниковий пристрій, який може бути налаштований виробником або розробником після виготовлення. FPGA програмується шляхом зміни логіки роботи принципової схеми, наприклад, з допомогою вихідного коду мовою проектування (типу VHDL), у якому можна описати цю логіку роботи мікросхеми. FPGA є одним з архітектурних різновидів програмованих логічних інтегральних схем (ПЛІС, рис. 1.1).



Рисунок 1.1 – Altera StratixIVGX FPGA

FPGA можуть бути модифіковані практично будь-якої миті в процесі їх використання. Вони складаються з сконфігурованих логічних блоків, подібних до перемикачів з безліччю входів і одним виходом (логічні вентиля або gates). У цифрових схемах перемикачі реалізують базові виконавчі операції AND, NAND, OR, NOR та XOR. У більшості сучасних процесорів функції логічних блоків фіксовані і можуть модифікуватися. Принципова відмінність FPGA

полягає в тому, що функції блоків, і конфігурація з'єднань між ними можуть змінюватися за допомогою спеціальних сигналів, що посилаються за схемою. У деяких спеціалізованих інтегральних схемах (ASIC) використовуються логічні матриці, аналогічні FPGA за структурою, проте вони конфігуруються якраз у процесі виробництва, тоді як FPGA можна постійно перепрограмувати та змінювати топологію сполук у процесі використання. Однак така гнучкість потребує суттєвого збільшення кількості транзисторів мікросхеми.

1.1 Історія створення FPGA

З моменту винаходу перших комп'ютерів у 1940-х і 1950-х роках конструювання комп'ютерних чіпів було швидко розвивається. Однак до недавніх інновацій нові досягнення в дизайні були в тому ж напрямку, що й технологія десятиліття. Проте в середині 1980-х років було розроблено новий тип комп'ютерної архітектури, яка за важливістю суперничає з появою інтегральних схем. Цей монументальний винахід полягав у створенні програмованих логічних пристроїв, точніше мікросхем програмованої вентиляльної матриці (FPGA).

Xilinx Inc. винайшла FPGA у 1985 році. Ця компанія побудувала існуючу програмовану логіку (пам'ять) і зробила наступний крок у створенні польового програмованого чіпа. Щоб зрозуміти реалізацію FPGA, спочатку потрібно зрозуміти попередні програмовані пристрої.

Першим програмованим логічним пристроєм була пам'ять PROM (Programmable Read Only Memory). Це енергонезалежна пам'ять, яку можна завантажувати інформацією. Різні типи PROM можуть бути масово запрограмовані на заводі (чіпи, програмовані за допомогою маски), або запрограмовані користувачем (програмовані на місці). Програмований у полі – це тип, з якого розвинулися мікросхеми FPGA. Програмоване ППЗУ може бути двох типів: EPROM (Erasable PROM) та EEPROM (Electronic Erasable PROM).

Найчастіше використовувана EEPROM дозволяє користувачеві стерти вміст пам'яті та перепрограмувати її багаторазово.

Наступним кроком в еволюції програмованої логіки є гігантське досягнення у напрямку розробки мікросхеми вентиляльної матриці, що програмується у полі. Цей крок був появою програмованого логічного пристрою (або PLD). Хоча існує кілька різних типів PLD, найбільш поширені реалізують набір фіксованих логічних вентилів АБО, яким передують масив програмованих логічних вентилів І. Як і PROM, PLD також виготовляються як заводські, так і користувачі, що програмуються. Ці PLD показують чіткий шлях розвитку FPGA.

Ці MPLD (програмовані логічні пристрої маски) мають програмовану логіку елементів І; однак вони жорстко під'єднані між логічними вентилями. У 1985 році, винайшовши ПЛІС, Xilinx створила пристрій, який мав не тільки програмовані вентиля, а й програмовані з'єднання між вентилями. Таким чином, FPGA були наступним кроком в еволюції PLD.

Ще один ранній польовий програмований логічний пристрій був запропонований Стівом Кассельманом у 1987 році Національному науковому фонду. Ідея пропозиції полягала в тому, щоб створити комп'ютерний чіп, який використав би нову технологію програмованих вентиляльних матриць і міг бути повністю запрограмований за допомогою програмного забезпечення. В рамках запропонованого досвіду Кассельман поставив дві мети: визначити спосіб з'єднання площин масивів і створити компілятор, який міг би програмувати функції в цих нових мікросхемах. Як і дизайн Xilinx, чіп Кассельмана спиратиметься на технологію регістрів EEPROM.

У наступні роки була подана заявка на допомогу від Департаменту військово-морських надводних боїв, і її було отримано, щоб розробити комп'ютер, що реалізує загалом 600 000 перепрограмованих вентилів матриці. У 1992 р. на цю систему було отримано патент. Завдяки успіху Кассельмана з цією реконфігурованою апаратною технологією, він став експертом у галузі

«віртуальних обчислень» (так названих через відсутність жорсткого зв'язку) та заснував Virtual Computer Corporation.

FPGA з'явилися в центрі уваги в 1997 році, коли Адріан Томпсон об'єднав технологію генетичних алгоритмів та FPGA для створення пристрою розпізнавання звуку. Цей пристрій використовував чіп Xilinx, який містив масив 64 x 64 осередків. Алгоритм Томпсона дозволив FPGA вирішити, якою має бути конфігурація мікросхеми для виконання завдання розпізнавання звуку. Після повторних генерацій програми (приблизно 1500) протягом двох-трьох тижнів чіпи не тільки змогли перевищити вимоги, встановлені для розпізнавання, але й зробили дуже ефективно.

Можливо, найдивовижнішою характеристикою системи розпізнавання голосу Томпсона було те, що вона використовувала трохи більше ніж тридцять комірок FPGA. Найкраща система, розроблена людиною, використовувала приблизно вп'ятеро більше клітин! З осередків, які використовувалися, п'ять не були пов'язані безпосередньо з виходом і, здавалося, не використовувалися проектом. Однак, коли ці елементи були вимкнені, FPGA більше не функціонувала.

Схема на рис. 1.2 показує "дизайн" системи розпізнавання звуку. Система розпізнавання мови – це повністю автономна і проста у використанні програмована схема розпізнавання звуку. Програмована в тому сенсі, що ми тренуємо слова (або голосові висловлювання), які хочемо, щоб схема розпізнала. Її діаграма містить кілька рекурсивних циклів, які здаються нескінченними. Зміст таких петель є порушенням поточних методів електричного проектування, проте FPGA спроектувала схему таким чином. Ця схема розроблена з використанням методів, що дуже відрізняються від звичайних чіпів у стилі фон Неймана, і дає багато нових можливостей для дизайну чіпів. Ефективність цієї та багатьох інших генетичних систем порівнюють із генетичною системою людського розуму. Ці типи конструкцій знаходять найбільш ефективний спосіб виконання завдання, використовуючи властивості кремнію, з якого вони виготовлені. Вони не спроектовані, а просто

сформовані. Багато в чому те, як працює людський розум, коли він встановлює неврологічні зв'язки.

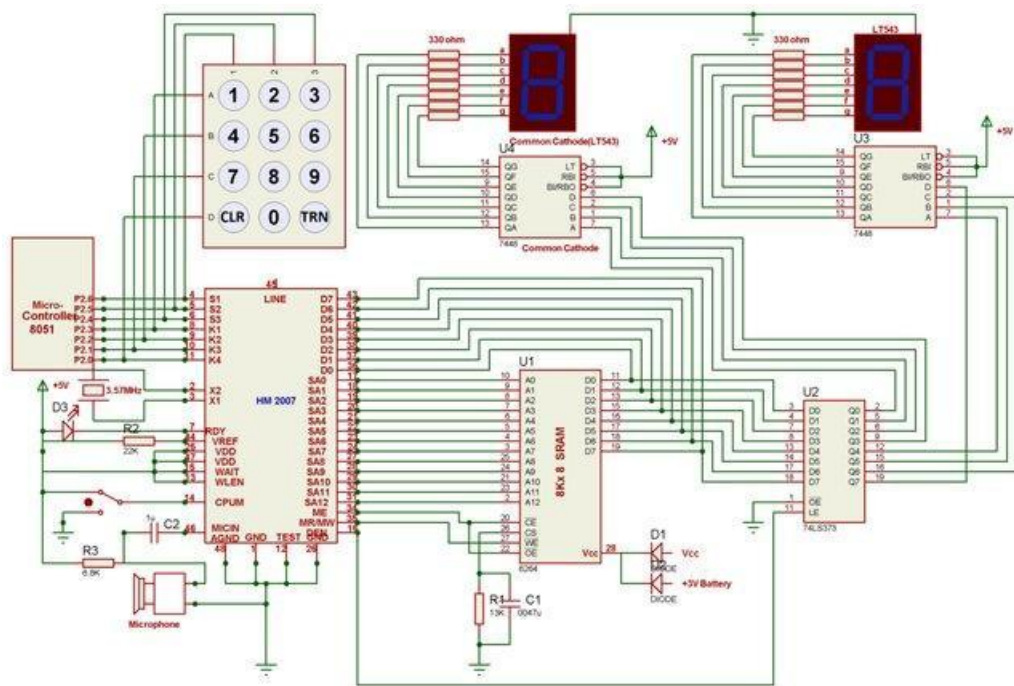


Рисунок 1.2 – Схема системи розпізнавання голосу з ручним режимом і режимом ЦП

Сучасні ПЛІС мають великі ресурси логічних елементів і блоків оперативної пам'яті для здійснення складних цифрових обчислень. Оскільки конструкції FPGA використовують дуже високі швидкості вводу-виводу та двонаправлені шини даних, перевірити правильну синхронізацію дійсних даних під час налаштування та часу утримання стає складним завданням.

1.2 Принципи побудови та архітектури ПЛІС

ПЛІС як електронний компонент використовується для створення цифрових інтегральних схем. На відміну від звичайних цифрових мікросхем, логіка роботи ПЛІС не визначається при виготовленні, а задається програмуванням.

1.2.1 Базова архітектура

У внутрішній області FPGA базової архітектури рядками і стовпцями розміщуються ідентичні функціональні блоки ФБ (КЛБ, конфігуровані логічні блоки), між якими проходять траси міжз'єднань. На периферії кристала розташовані блоки введення/виведення БВВ. Для першого покоління перераховані частини становили сутнісно всю схему FPGA. Надалі архітектура FPGA ускладнилася, і в її складі з'явилися додаткові функціональні ресурси, серед яких насамперед слід назвати вбудовані блоки пам'яті (їх стали вводити практично у всі мікросхеми, що розробляються) і помножувачі. В складі FPGA з'явилися також спеціалізовані засоби для автопідстроювання затримок у системі тактування (PLL, DLL, DCM), засоби для підтримки інтерфейсу JTAG, для реалізації високопродуктивних шин, для генерації тактових сигналів тощо.

На рис. 1.3 укрупнено показано фрагмент FPGA базової архітектури та склад її функціонального блоку ФБ, до якого входять:

- функціональний (логічний) перетворювач (ФП); тригер (реєстр RG);
- мультиплексори (MUXs), які відіграють роль засобів конфігурування ФБ.

При конфігуруванні FPGA функціональні (логічні) блоки налаштовуються виконання необхідних операцій, а система з'єднань — на необхідні зв'язки між елементами і блоками. У результаті FPGA реалізується схема необхідної конфігурації. Блоки введення/виводу пов'язують FPGA із зовнішнім середовищем, і їх, як правило, можна програмувати на виконання ряду стандартів передачі даних.

1.2.2 Ускладнені архітектури

У схемах FPGA з ускладненою архітектурою крім перерахованих блоків застосовуються і додаткові, до яких відносяться:

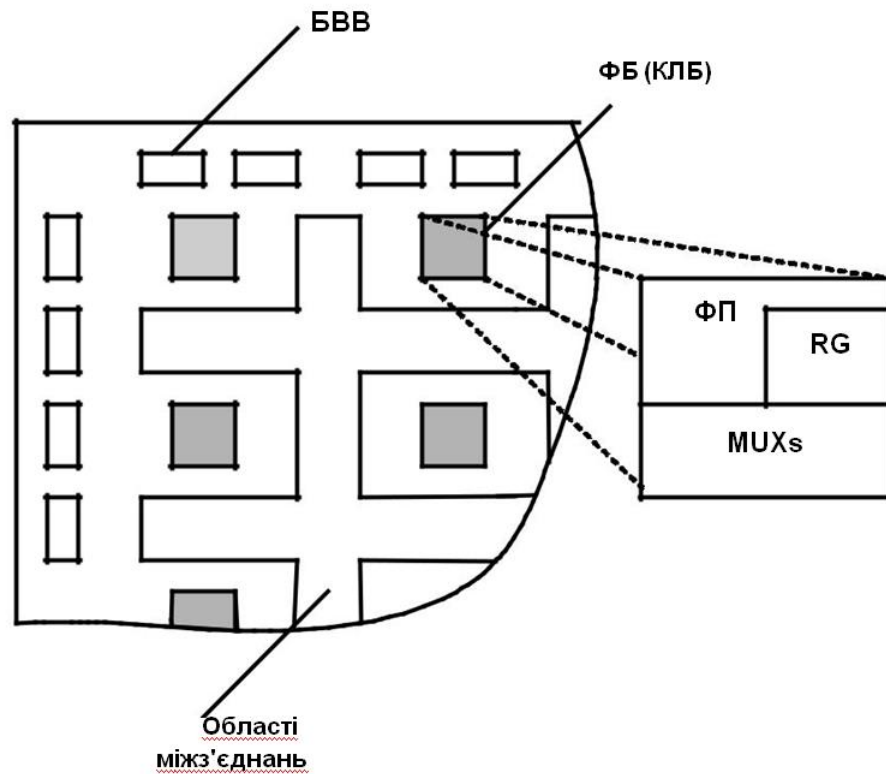


Рисунок 1.3 – Фрагмент FPGA базової архітектури та її функціональний блок

- ресурси пам'яті;
- помножувачі;
- схеми керування синхросигналами (PLL, DLL); блоки ЦГЗ (цифрової обробки сигналів);
- блоки обробки аналогових та аналого-цифрових даних.

Фрагмент типової структури ускладненої FPGA показано на рис. 1.4 (мікросхема Stratix-3).

До основних частин мікросхеми входять адаптивні логічні модулі ALM, блоки швидкодіючої пам'яті зі змінною організацією, блоки цифрової обробки сигналів (ЦОС-блоки) та банки вводу/виводу, що забезпечують інтерфейс для взаємодії з різними зовнішніми пристроями. Застосовано трирівневу ієрархію блоків пам'яті. Є блоки пам'яті трьох типів - M1AB, M9K і M144K з ємностями 640 біт, 9 Кбіт і 144 Кбіт відповідно.

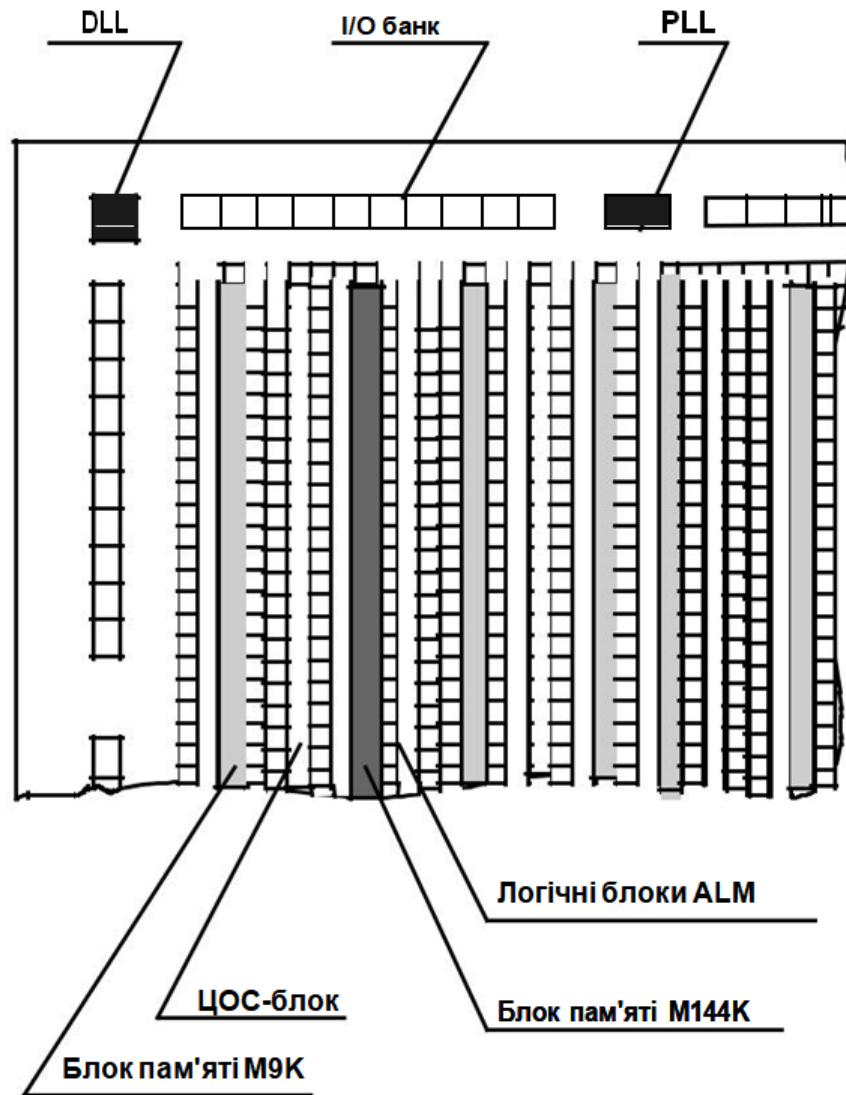


Рисунок 1.4 – Структура FPGA ускладненої архітектури

1.2.3 Логічні блоки

ПЛІС, побудовані за архітектурою FPGA, складаються з логічних блоків і шляхів комутації - програмованих матриць з'єднань. Логічні блоки таких ПЛІС складаються з одного або кількох відносно простих логічних елементів, які базуються на таблиці перекодування (LUT, Look-up table), програмованих мультиплексорах, тригерах і схемах керування. Таких простих елементів може бути досить велика кількість, в сучасних ПЛІС ємністю до 10 і більше мільйонів еквівалентних вентилів кількість логічних елементів досягає десятків і сотень тисяч. Швидкі темпи зростання продажів високоінтегрованих ПЛІС

привели до того, що вони стали чи не єдиною розумною альтернативою БМК і кастомним ІІ.

Розвиток і різноманітність архітектури функціональних перетворювачів, що лежать в основі базових вузлів FPGA, призвело до того, що в останні роки програмована логіка стала основою для систем на кристалі (SOC, system-on-chip). В основі ідеї систем на кристалі лежить інтеграція всієї електронної системи одним кристалом (наприклад, у випадку з ПК такий чіп поєднує в собі процесор, пам'ять і т.д.). Компоненти цих систем розробляються окремо і зберігаються у вигляді файлів модулів зі змінними параметрами. Остаточна структура системи на кристалі виконується з урахуванням цих «віртуальних компонентів», званих «блоками інтелектуальної власності», за допомогою програм автоматизації електронного проектування – EDA (Electronic Design Automation). Завдяки стандартизації віртуальні компоненти від різних розробників можна об'єднати в один.

Як КЛБ (конфігурованих логічних блоків FPGA або, для стислості, просто ЛБ) використовуються:

- SLC (Simple Logic Cells), прості логічні вентиля; логічні модулі з урахуванням мультиплексорів;

- LUT-блоки – програмовані ЗУ (LUT - Look-Up Table).

LUT-блок — найпоширеніший різновид логічного перетворювача для FPGA з тригерною пам'яттю конфігурації. В FPGA, що одноразово програмується, з перемичками antifuse, що пробиваються, знаходять застосування логічні модулі на основі мультиплексорів, а в схемах з флеш-ключами — перетворювачі з простими логічними вентилями.

Однією з характеристик логічних блоків є їхня "зернистість" (Granularity), що визначає, наскільки «дрібними» будуть ті частини, з яких можна "збирати" потрібні схеми.

Дрібнозернистість ЛБ веде до гнучкості їх використання, можливостям реалізувати відтворювані функції різними способами, отримуючи різні варіанти в координатах «площа кристала – швидкодія». Одночасно вона ускладнює

систему міжз'єднань внаслідок великої кількості програмованих точок зв'язку на платі.

Середньозернисті логічні блоки (приклад середньозернистого ЛБ фірми Actel) – блоки, що складаються з трьох мультиплексорів "2-1" та елемента АБО (рис. 1.5).

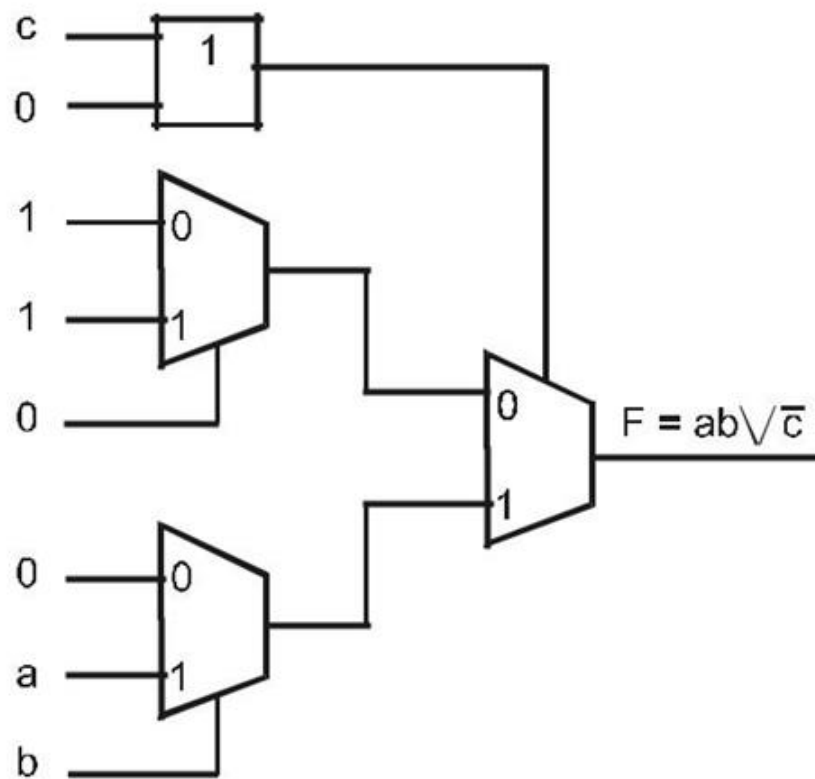


Рисунок 1.5 – Приклад реалізації функції за допомогою мультиплексорного логічного блоку

Сучасна програмована логіка випускається з можливістю програмування в системі безпосередньо на платі. Слід зазначити, що нові ПЗУ конфігурації EPC2 і вище дозволяють програмувати за допомогою цього пристрою, тим самим усуваючи потребу в програматорі, що, природно, знижує вартість мобільності, що збільшує вартість.

Наявність готових модулів - ядер, мегафункцій - призначених для вирішення досить складних завдань обробки сигналів, також характерна в наш час. Швидкими темпами йде розробка готових функцій зусиллями сторонніх

компаній. Основну масу розробок складають мегафункції, які реалізують стандартні мікропроцесори та мікроконтролери, пристрої обслуговування шин (DDR, PCI), мережеві контролери тощо. Типовими пропозиціями засобів CGS є мегафункції, що реалізують швидке перетворення Фур'є (FFT) і фільтри кінцевої імпульсної характеристики (FIR-фільтри).

Наприклад, можна розглядати автомобільну електроніку як варіант побудови системи на кристалі, адже для отримання кінцевого зображення сигнал детектора повинен пройти структурування, фільтрацію та відображення. Варіант FPGA може бути типовим прикладом реалізації всіх функцій в одному пристрої.

Ще один тип архітектури ПЛІС – вентиляльні матриці. Це ПЛІС, які взагалі не містять логічних матриць і вважаються ПЛІС через можливість програмування користувачем (рис. 1.6). Вони складаються з основних функціональних елементів або логічних блоків і комутаційних з'єднань між цими елементами.

Прості, або «дрібнозернисті», блоки складаються з транзисторних пар р- і n-канальних транзисторів, простих логічних елементів І-НІ, АБО-НІ, мультиплексорів «Виключне АБО». На їх основі за допомогою комутації можна по-різному реалізовувати логічні комбінаційні функції, тригери та пристрої на їх основі, аж до складних кінцевих автоматів [10].

В ІС на «дрібнозернистих» логічних блоках можлива оптимізація БІС, що розвивається за швидкістю шляхом зміни розташування та взаємного розташування частин проектованого пристрою на кристалі.

ПЛІС на основі «дрібнозернистих» (рисунок 1.6) логічних блоків мають топологію ВМС з горизонтальними каналами трасування. ПЛІС з низьким ступенем інтеграції складаються з серії сконфігурованих логічних блоків, між якими проходять програмовані шини.

Складні, «грубозернисті» блоки (рисунок 1.7) містять більш складні макроелементи, які можуть бути використані для побудови комбінаційних або послідовних логічних функцій. Комбінаційна частина блоку реалізує функцію

чотирьох і більше змінних. У послідовних частинах тригери з керованою конфігурацією дозволяють будувати лічильники, регістри, кінцеві автомати тощо.

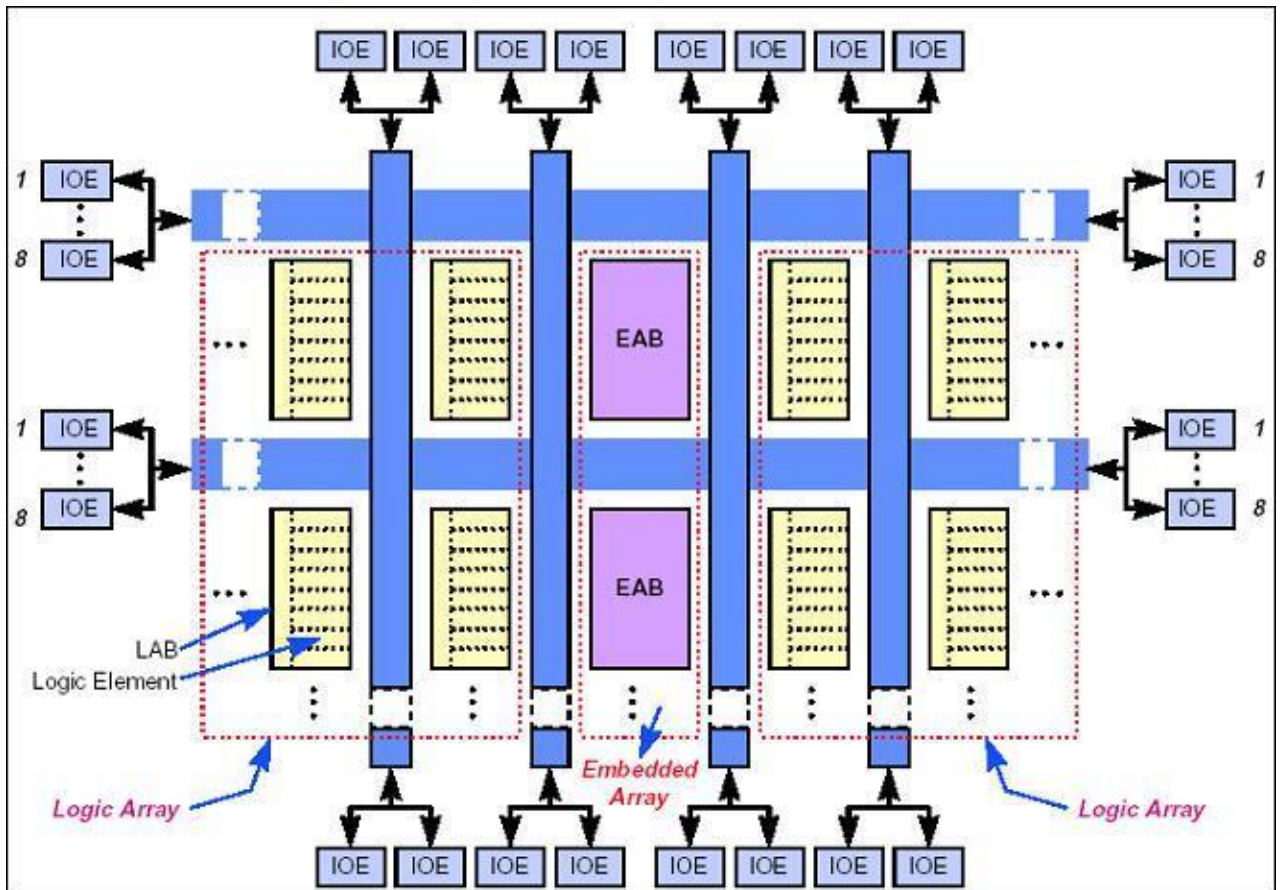


Рисунок 1.6 – «Дрібнозернисті» логічні блоки програмованої користувачем вентиляльної матриці

Матриця логічних елементів (LE) на основі FPGA на основі «грубих» логічних блоків:

- між LE існують вертикальні та горизонтальні канали зв'язку;
- канали складаються з сегментів;
- сегменти комутуються програмованою матрицею з'єднань.

Конфігурація ПЛІС (конфігурація ПЛІС – це процес завантаження бітової послідовності, отриманої за допомогою програмного забезпечення системи автоматизованого проєктування, у внутрішню конфігураційну пам'ять

кристала) з крупнозернистими блоками виробляється елементами СОЗУ - тригерної пам'яті. розподілені по всьому кристалу [12].

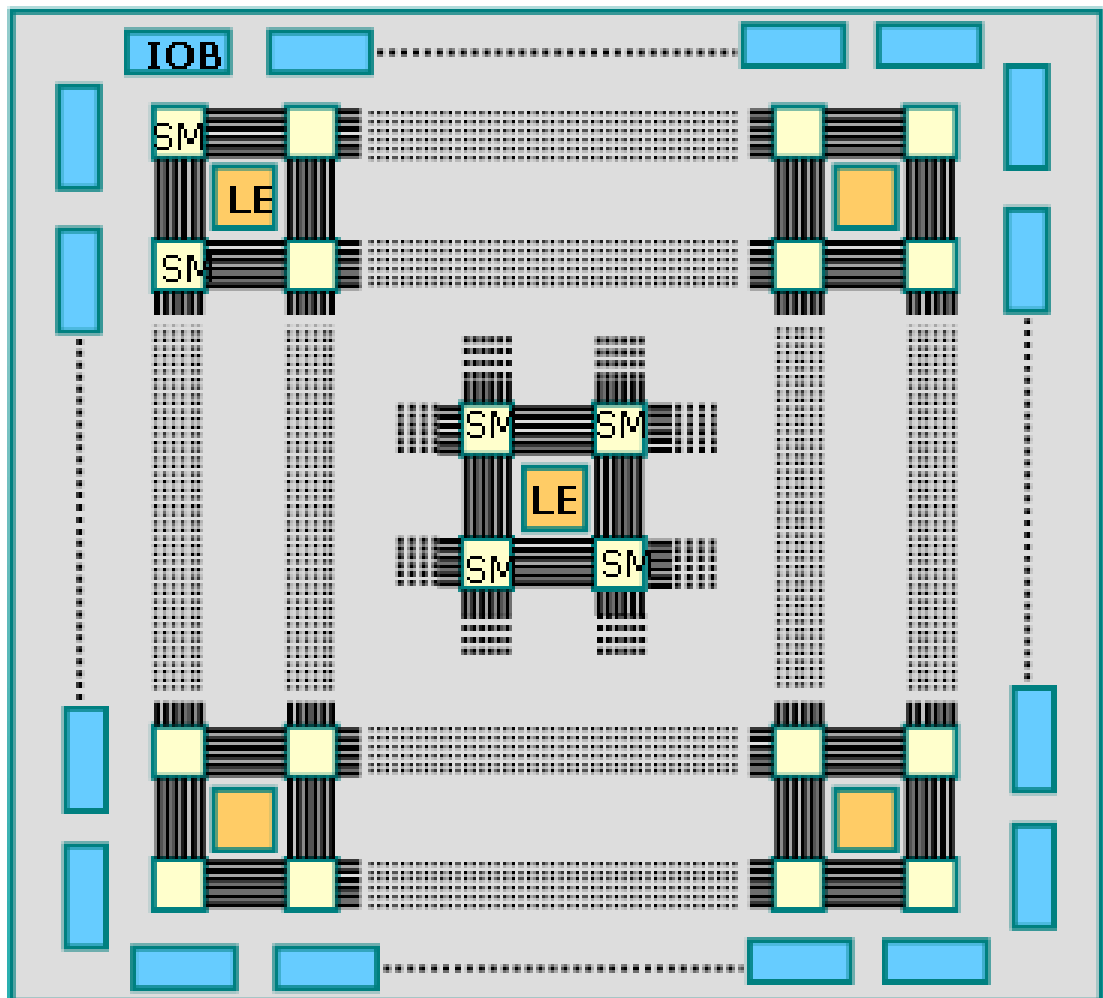


Рисунок 1.7 – «Грубозернисті» логічні блоки FPGA

Цей спосіб конфігурації типовий для ПЛІС.

Його перевагами є простота програмування, багаторазове використання кристала в разі експериментального налагодження, модифікації системи шляхом завантаження нової конфігураційної програми в кристал безпосередньо в системі.

Недоліки цього методу налаштування:

необхідність зовнішніх пристроїв, в яких зберігається програма конфігурації;

можливість копіювати програму конфігурації під час завантаження;
в процесі безперервної тривалої експлуатації можливі часткові збої конфігурації, збільшення струму живлення при навантаженні.

Особливістю сучасних ПЛІС є можливість тестування вузлів за допомогою порту JTAG, а також наявність внутрішнього генератора і послідовних схем управління конфігурацією.

Подальший розвиток йде по шляху створення комбінованої архітектури, яка поєднує в собі зручність реалізації алгоритмів цифрової обробки сигналів на основі таблиць перекодування і реконфігурованих модулів пам'яті, характерних для структур ПЛІС і багаторівневих ПЛІС, із зручністю реалізації цифрових автоматів на CPLD архітектура.

Архітектура модулів FPGA є поєднанням логічних блоків у мегаблоки (MegaLab), які мають власні матриці безперервного комутованого з'єднання (MegaLab interconnect) [13].

При виборі елементної бази систем обробки сигналів, систем телекомунікації та зв'язку, периферійного та тестового обладнання зазвичай використовуються такі критерії вибору:

- швидкість дії;
- логічна місткість, достатня для реалізації алгоритму;
- принципові та конструктивні параметри ПЛІС, технологічність, надійність, діапазон робочих температур, стійкість до іонізуючого випромінювання тощо;
- вартість володіння засобами розробки, що включає як вартість програмного забезпечення, так і наявність і вартість апаратного забезпечення для налагодження;
- вартість обладнання для програмування ПЛІС або конфігураційних ПЗУ;
- наявність методичного та технічного забезпечення;
- вартість мікросхем.

Аналізуючи наведену вище елементну базу для побудови розробленого модуля контролю параметрів, приходимо до висновку, що реалізація даного проекту на основі дискретних логічних елементів не є оптимальною з кількох причин:

- складність проектування схеми та трасування друкованої плати за рахунок більшої кількості з'єднань;
- складність налаштування необхідних параметрів через виникнення затримок часу при проходженні сигналу через кожен логічний елемент;
- велика кількість елементів на платі призводить до зниження надійності пристрою в цілому;
- зі збільшенням кількості компонентів на друкованій платі зростає собівартість виробу, що економічно недоцільно при дрібносерійному виробництві.

З усього вищесказаного робимо висновок, що найбільш оптимальною є реалізація системи керування на основі програмованої логіки.

Висновки по розділу

FPGA (Field Programmable Gate Arrays) — програмовані користувачем вентильні матриці – найбільший клас програмованих схем, що мають максимальні функціональні можливості. На основі створені системи на програмованому кристалі СнПК (в англійському оригіналі SoPC, Systems on Programmable Chip).

З урахуванням архітектурних особливостей та областей застосування виділяються наступні підкласи FPGA та систем на їх основі:

- FPGA невисокої та середньої складності;
- FPGA високої складності та системи на кристалі;
- Мікроконтролері програмовані системи.

У розробці FPGA беруть участь десятки фірм, провідні серед них – Xilinx (піонер у створенні FPGA), Altera, Actel, Atmel, Lattice Semiconductor, Cypress

Semiconductor (усі USA) та ін. Цими фірмами випускаються сімейства FPGA, які в міру освоєння нових технологічних процесів (з інтервалом у рік-два) піддаються модифікаціям і утворюють серії, що складаються з родинних сімейств.

2 СИСТЕМИ КЕРУВАННЯ АВТОМОБІЛЕМ

Сучасна автомобільна електроніка спирається на мікроелектронні компоненти, які мають забезпечити максимально комфортне, зрозуміле та безпечне керування автомобілем [1].

Серед них можна виділити:

- Електронний блок управління;
- Датчики, що дають інформацію про стан окремих агрегатів і автомобіля в цілому;
- Пристрої силової електроніки, що забезпечують керування окремими системами автомобіля;
- Виконавчі пристрої, що забезпечують зв'язок між електронними системами та агрегатами і механізмами автомобіля.

Розглянемо деякі складові електронної системи автомобіля.

2.1 Система датчиків автомобіля

До датчиків, що забезпечують роботу транспортного засобу, відносяться датчики тиску, температури, швидкості, частоти обертання двигуна, датчики, що відповідають за системи активної та пасивної безпеки, прилади контролю стану трансмісії автомобіля, датчики стану викидів в атмосферу, системи контролю тиску в рядах, датчики систем кондиціонування, сигналізації та багато інших (рис. 2.1).

Сучасні датчики оснащені цифровими інтерфейсами, що спрощує передачу даних по одній шині і значно підвищує захищеність системи в цілому. Також ця архітектура дозволяє істотно спростити структуру центрального блоку управління транспортним засобом і підвищити продуктивність центрального процесора. Обмін даними між окремими системами автомобіля здійснюється через стандартизовану шину CAN [2].

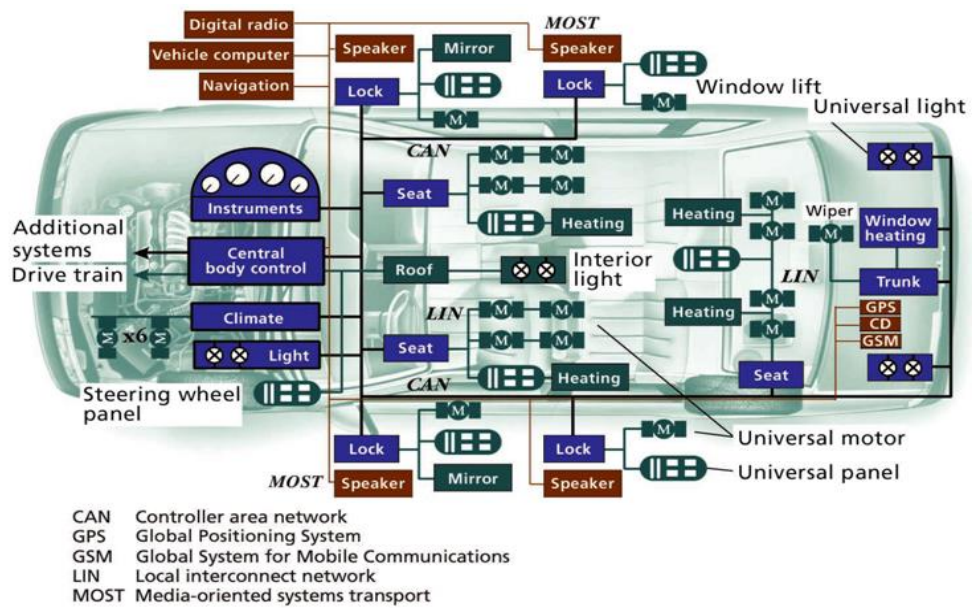


Рисунок 2.1 – Загальна структура системи контролю параметрів автомобіля

2.2 Складові електронної системи контролю

Сучасна компонентна база цифрової мікроелектроніки має структуру, що зображена на рисунку 2.2.

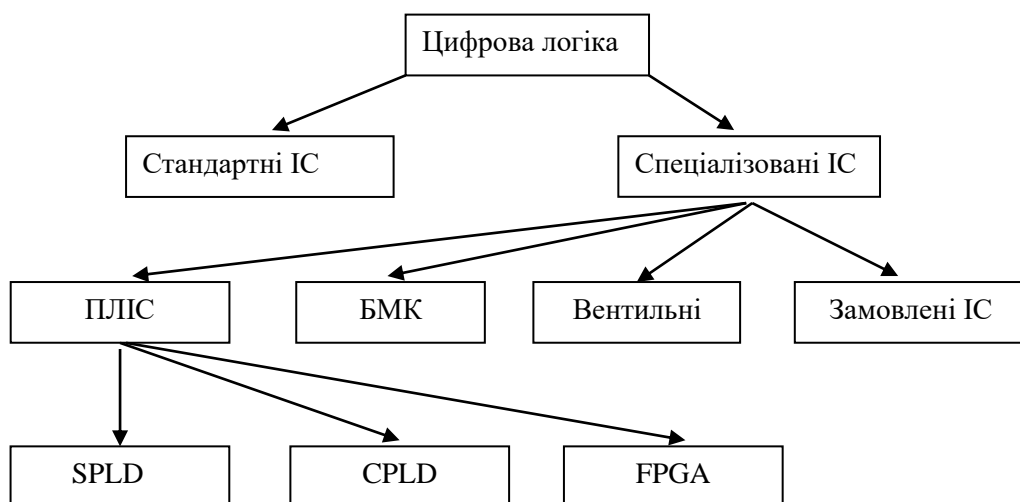


Рисунок 2.2 – Блок-схема систем цифрової логіки

В автомобільній електроніці в якості основного пристрою обробки зазвичай використовуються спеціалізовані інтегральні схеми (ASIC) з процесорним ядром і необхідною периферійною частиною [3].

У сучасній апаратурі контролю та керування використовуються електронні блоки, побудовані на наступній компонентній базі мікроелектроніки:

- мікросхеми стандартної логіки;
- мікроконтролери;
- процесори DSP;
- мікропроцесори;
- програмовані логічні контролери;
- вентиляльні матриці (FPGA).

У найпростіших системах управління мікроконтролери використовуються для інтерпретації комп'ютерних команд, в більш складних мікросхемах стандартної логіки або мікропроцесорах. Кожна із систем має свої недоліки. Системи на основі мікроконтролерів не забезпечують достатньої функціональності та швидкості. Оскільки мікроконтролери намагаються зробити універсальними, їм часто не вистачає периферії або їм не вистачає обсягу, а установка додаткових мікросхем позбавляє сенсу від використання мікроконтролерів. Останнім часом їх замінили цифрові сигнальні процесори. Хороші результати дають контролери на основі стандартних логічних мікросхем, але вони громіздкі, вимагають потужного блоку живлення, займають великий об'єм, не допускають модернізації без перероблювання конструкції, мають низьку ремонтпридатність і важко піддаються діагностиці. Сьогодні при розробці нового обладнання такі контролери не використовуються [4].

Сучасні контролери також побудовані на програмованих логічних контролерах (ПЛК), в основі яких лежить мікропроцесор. Мікропроцесор (МП) — мікроелектронний компонент для цифрової обробки інформації на основі кодів програмних команд, що зчитуються із пристроїв пам'яті. Мікропроцесори

не є самодостатніми пристроями, і для побудови комп'ютерної системи вони вимагають підключення ряду додаткових мікросхем (пам'яті, контролерів, інтерфейсів). Першою і головною перевагою ПЛК, яка зумовила їх широке поширення, є те, що один компактний електронний пристрій може замінити десятки і сотні логічних мікросхем. Друга перевага полягає в тому, що функції логічних контролерів реалізуються не апаратно, а програмно, що дозволяє постійно адаптувати їх для роботи в нових умовах з мінімальними зусиллями і витратами. Ця перевага також є недоліком, при простоті зміни функціонування ми втрачаємо в швидкості. Оскільки ПЛК прийшли на зміну блокам на логічних мікросхемах і реле, в їх мовах програмування широко використовується булева алгебра - операції з бітами. Таким же чином програму для ПЛК можна створити графічно - подібно до того, як схема пристрою малюється на логічних мікросхемах. У середовищі програмування ПЛК логічні блоки (І, АБО, НІ, тригери, лічильники, таймери) зі спеціальних бібліотек розміщуються та підключаються відповідно до логіки програми. Для виконання такої програми мікропроцесор повинен відповідно до алгоритму визначити стан усіх блоків і портів, виконати логічні операції та вивести відповідний результат у вигляді керуючого впливу. При великій кількості таких блоків МП не встигає обробити їх усі за необхідний інтервал часу, що обмежує використання таких пристроїв для високошвидкісних програм. Для підвищення швидкодії мікропроцесора використовується додатковий арифметичний співпроцесор, який ефективно виконує обчислювальні операції під керуванням МП. Це ускладнює всю систему і не є виходом із ситуації, тому що при подальшому ускладненні програми потрібен ще один співпроцесор [5].

Однак головне полягає в тому, що неможливо безпосередньо використовувати ПЛК у транспортних додатках без значної обробки, що зумовлено їхніми розмірами та конструктивними характеристиками.

Розробки автомобільної електроніки вимагають значного скорочення циклу проектування та збільшення цифрового контенту. Протягом наступних кількох років автомобільні електронні системи зростатимуть набагато

повільніше, ніж темпи зростання ринку PLD (програмованих логічних пристроїв). Існує кілька факторів, що сприяють розвитку PLD у транспортних системах. По-перше, це еволюція автомобіля до концепції «платформи», за якої різні моделі автомобілів оснащені однією базовою конструкцією. Поєднання аудіо, відео та даних в одному просторі автомобіля додатково підтримує концепцію платформи. По-друге, це швидкий інноваційний цикл. Сьогодні типова конструкція автомобіля займає приблизно 24...36 місяців, що набагато швидше, ніж п'ятирічний цикл кілька років тому. Цей короткий цикл проектування вимагає від розробників прискорення та скорочення етапів розробки електронних систем. По-третє, збільшення цифрового контенту у формі навігаційних систем, розваг і програм допомоги водієві ускладнює проблему.

По-четверте, виробники ASIC не встигають за вимогами ринку, тому звертають свою увагу на програмовану логіку, яка забезпечує більшу гнучкість проектування. Усі ці фактори вимагають від розробників створення дизайнерського рішення, яке не тільки має гнучкість, але й відповідає вимогам продуктивності в рамках проекту [6].

Сучасний підхід до реінжинірингу складних логічних систем передбачає широке використання програмованих логічних інтегральних схем, що зумовлено їх гнучкістю та універсальністю, а також значно спрощує розробку нестандартних інтегральних схем у разі тиражування аналогічних пристроїв у майбутньому.

Основою програмованих логічних інтегральних схем (ПЛК) є матриця елементарних логічних блоків. Змінюючи зв'язки між блоками, можна побудувати обчислювальний пристрій довільної структури, що ідеально відповідає конкретному завданню. Комбінація цих блоків може точно повторювати програму для ПЛК. Різниця полягатиме в тому, що логічні блоки всередині FPGA працюють незалежно та одночасно. Звідси, щоб виконати якусь операцію, не потрібно чекати, поки мікропроцесор прочитає відповідну команду з пам'яті. Програма в FPGA виконується в рази швидше, а контролер

миттєво реагує на сигнали датчиків. Швидка реакція відіграє важливу роль у високопродуктивному та точному обладнанні.

2.3 Робота блоку електронного керування автомобілем

Безпека водіння та зручність водіння автомобіля залежать від безлічі датчиків для вимірювання багатьох внутрішніх і зовнішніх змінних, які можуть вплинути на поведінку автомобіля, а також додаткових параметрів, таких як видимість і комфорт пасажирів. Залежно від рівня своїх параметрів датчики можна класифікувати на прості, які безпосередньо вимірюють окремі фізичні параметри (наприклад, датчики зовнішнього освітлення та датчики температури), до складних інтелектуальних датчиків, які визначають параметри навколишнього середовища за допомогою сигналів широкого спектру (наприклад, радіо). частота/радар та світло/відео). Крім вимірювань, вони виконують обробку даних і мають можливість виконувати певні дії на деякі виконавчі механізми. З огляду на те, що інтелектуальні датчики використовують під собою дані різного характеру, в яких закодовані складні та нелінійні форми поведінки, методи обміну даними, які використовуються разом з алгоритмами машинного навчання, забезпечують ефективні дані для моделювання цієї інформації. Оскільки розумні датчики часто покладаються на складні датчики та методи синтезу датчиків, необхідну потужність обробки даних можуть забезпечити лише високопродуктивні обчислювальні платформи, такі як мікропроцесори, графічні процесори або програмовані вентильні матриці. Зокрема, реалізація на основі програмованої логіки виділяється надзвичайно високими робочими частотами та низьким енергоспоживанням навіть для складних багаторівневих алгоритмів. У контексті автомобільної промисловості інтелектуальні датчики є ключовими компонентами сучасних систем допомоги водієві.

У сфері комфорту водія прості датчики швидкості відіграють важливу роль у звичайних системах зворотного зв'язку, таких як рішення круїз-

контролю, які звільняють водіїв від тягаря підтримки постійної швидкості під час довгих поїздок по шосе. Однак, незважаючи на те, що системи керування допомагають зменшити втому водія під час тривалих поїздок, вони можуть спричинити небезпечні ситуації, особливо зіткнення ззаду. Такі ситуації виникають через те, що звичайні системи не знають відстані від транспортного засобу, що йде попереду, тобто вони враховують лише внутрішні змінні транспортного засобу. Щоб вирішити ці проблеми, системи зовнішнього світу, такі як адаптивний круїз-контроль, виступили як альтернатива для забезпечення підвищених функцій безпеки та комфорту. Дія цього типу контролю полягає у вимірюванні відстані до попереднього транспортного засобу в режимі реального часу, щоб зберегти розрив у часі та уникнути зіткнень ззаду, таким чином підвищуючи безпеку дорожнього руху. Такі системи засновані на складних інтелектуальних датчиках, які вимірюють зовнішні параметри за допомогою широкосмугових сигналів, таких як радар або відео. Ці типи передових систем відомі як сучасні системи допомоги водієві [7].

Система допомоги водієві покладається на безперервний потік даних від кількох датчиків, які вимірюють внутрішні та зовнішні змінні, щоб забезпечити розширену функціональність. Залежно від рівня роботи їх можна класифікувати на дві категорії: пасивний і активний контроль. Перший надає поради або інформацію водієві. Прикладами пасивного моніторингу є датчики сліпих зон, які використовують ультразвукові датчики для виявлення перешкод у сліпій зоні дзеркала заднього виду. Системи запобігання зіткненням використовують радіолокаційні станції або відеосигнали для виявлення потенційних лобових зіткнень, попереджаючи водія (повідомлення про лобове зіткнення). Система повідомлення про зустрічну смугу розпізнає розмітку за допомогою відеосигналів і інформує водія про події на смузі виїзду. Нарешті, системи ідентифікації/розпізнавання дорожніх знаків здатні виявляти обмеження швидкості, що відображає повідомлення на панелі приладів автомобіля. Останній, активний контроль, може виконувати активні дії на автомобіль, що

такі системи, як автономне екстрене гальмування, можуть автоматично зупиняти автомобіль.

Згадані вище системи використовувалися для підвищення безпеки в автомобілях, оскільки вони звільняють водіїв від деяких найважливіших завдань безпеки. Поєднання всіх цих інтелектуальних датчиків разом із високошвидкісним бездротовим зв'язком дозволило виробникам автомобілів уперше розробити інтелектуальні транспортні засоби з автоматизованими системами водіння.

Найважливішим конструктивним фактором програмованих логічних пристроїв є те, що вони є програмованими логічними пристроями. Звичайно, програмне забезпечення процесора може бути оновленим, але цього не можна сказати про архітектуру мікроконтролера або мікропроцесора. З іншого боку, програмовані логічні мікросхеми можна налаштувати або перепрограмувати для виконання різних функцій нескінченну кількість разів. У багатьох найновіших автомобілях програмне забезпечення контролює багато функцій під час роботи. Наприклад, моделі Tesla вже підтримують дистанційне оновлення програмного забезпечення.

Завдяки цій можливості програмовані логічні контролери можуть безперервно підтримувати оригінальне програмне забезпечення виробників за допомогою останніх версій для систем з програмованою або індивідуальною апаратною архітектурою. Такі оновлення програмного забезпечення можна застосовувати до різних функцій автомобіля, які можуть включати більше програмованих логічних структур, оскільки вони стають потужнішими, меншими та дешевшими [8].

Досліджуючи, який 32-розрядний процесор найкраще задовольнить потреби клієнтів, багато компаній розуміють, що архітектура галузевого стандарту пропонує значні переваги для власника. Стандартні процесори, як правило, мають широкий набір засобів розробки, значну кількість доступного програмного коду та штат дизайнерів, які мають знання та досвід для їх використання. Такі переваги пришвидшують час розробки проекту (і, отже, час

виходу на ринок) і зменшують ризик проекту, що, у свою чергу, надає користувачам рішення з більшою доданою вартістю.

На цьому фронті виділяється процесор Cortex-M1, розроблений з нуля для використання в програмованій логіці. Однією з основних функцій є те, що він допомагає мінімізувати кількість ресурсів, необхідних для задоволення вимог розробника. Наприклад, функції налагодження можна ввімкнути або вимкнути. Розширення операційної системи для системних таймерів і програмних переривань не обов'язкові. Cortex-M1 працює з більшістю основних програмованих логічних мікросхем, тобто перемикає з одного програмованого логічного пристрою на інший вимагає мінімальних зусиль.

Висновки по розділу

На основі аналізу методів і засобів побудови електронних блоків керування автомобілем зроблено висновок щодо вимог до мікроелектронних пристроїв даного класу, їх конструкції, системних вимог, потужності та ресурсів інтерфейсних схем. Проведено аналіз застосування різних компонентних баз, їх архітектури, продуктивності та швидкості.

Основна мета автомобільного дизайну — зменшити кількість процесорів, оскільки вони збільшують загальну вартість автомобіля та знижують надійність. Завдяки досягненням у програмованих логічних структурах тепер можна більш розумно комбінувати електронні компоненти всередині автомобіля. Наприклад, реалізація виключно апаратної архітектури процесора є проблемою, яка потребує термінового вирішення. Одним із можливих рішень цих перешкод є гібридний підхід, який поєднує центральні та графічні процесори з системами на чіпі на основі програмованих логічних структур.

Масиви вентилів програмованої логіки, їх структура і велика кількість ресурсів забезпечують розробникам рішення, яке відповідає найсуворішим вимогам автоелектроніки.

3 ПРОЕКТУВАННЯ СИСТЕМИ УПРАВЛІННЯ ТА ДІАГНОСТИКИ АВТОМОБІЛЯ НА ПЛІС

Сучасна інтелектуальна система – це набір модулів, які збирають, обробляють, зберігають і відображають інформацію про стан об'єкта управління. Обробляючи дані, відповідно до заздалегідь сформульованої програми, інтелектуальна система може виконувати різні функції. Інтелектуальна система самостійно збирає та обробляє дані, без втручання оператора запускає ту чи іншу функцію програми. Для вирішення подібних задач використовуються БМК, МК, МП і ФПЛІС [9].

Сучасна інтелектуальна система складається з двох частин: аналогової, в якій відбувається перетворення сигналу датчика і його первинна обробка, і цифрової, за допомогою якої сигнал перетворюється в остаточну форму, наприклад, для відображення на екрані або збереження на цифрових накопичувачах.

Аналогову партію найпростіше виконати з використанням БМК. ВМС - це структура, яка містить матрицю основних клітин (ВС). Кожна комірка складається з транзисторів, діодів, резисторів.

Якщо розробнику необхідно застосувати оригінальні схемотехнічні рішення на базі БІС, але наявні БІС для цих цілей не підходять, розробляти з нуля довго, неефективно і дорого, можна використовувати набір уже розроблених осередків базового матричного кристала. Базовий матричний кристал нагадує бібліотеку процедур і функцій для мов програмування. Замовник розробляє схему підключення, так звану маску. Ця маска наноситься як останній шар на базовий матричний кристал, і елементарні схеми та окремі кола збираються в одну велику схему. В результаті замовник отримує ІС не набагато дорожче оригінального базового матричного кристала.

Конструкція сучасних базових матричних кристалів відносно проста і доступна. Але для створення монолітних інтегральних мікросхем на базових матричних кристалах потрібні спеціалізовані архітектури з матрицями

чутливих і цифрово-аналогових елементів з розширеними експлуатаційними можливостями.

3.1 Властивості та можливості FPGA

FPGA, як і інші програмовані користувачем мікросхеми (ППМС), випускаються як повністю готові, тобто належати до стандартної продукції, що супроводжується відомими перевагами – масовістю виробництва та зниженням вартості. Споживач використовує їх, не звертаючись до виробника (виконує програмування FPGA самостійно).

Завдяки регулярній структурі FPGA реалізуються з рівнем інтеграції, близьким до максимального. Водночас оскільки для засобів програмування між'єднань потрібні значні витрати додаткової площі кристала, за кількістю логічних елементів, що надаються для реалізації проєкту, FPGA, як і інші ППМС, поступаються напівзамовним і тим більше замовленим схемам.

Відзначені властивості та можливості притаманні не тільки FPGA, але до певної міри й іншим ПЛІС. Однак для FPGA ці можливості виражені найбільш яскраво.

Зазначимо дві сфери застосування FPGA:

- відпрацювання прототипів блоків та систем при їх проектуванні, навіть якщо їхня кінцева реалізація розрахована на інші засоби;
- створення кінцевої продукції для виробів не надто великої тиражності швидкими та ефективними способами.

Спочатку розвиток ППМС був спрямований на заміну схем малого та середнього рівнів інтеграції та на перенесення концепції вентиляльних матриць у область малотиражної апаратури, але надалі у зв'язку з появою запрограмованості стало ясно, що це щось значно більше. Це, зокрема, видно з наведених нижче прикладів.

Побудова систем, що ре конфігуруються. При використанні апаратури зустрічаються ситуації, у яких ті чи інші блоки працюють по черзі. Наприклад,

засоби перешкодостійкого кодування та декодування при передачі та прийомі даних. Обидві функції (кодування та декодування) ніколи не виконуються одночасно. Тому не обов'язково мати два пристрої (кодер і декодер), а можна мати одну схему, що ре програмується, з двома різними конфігураціями, що зберігаються в ПЗУ і по черзі завантажуються в ПЛІС (програмовану логічну інтегральну схему). У такій системі, що ре конфігурується, одна і та ж апаратна частина може виконувати різні перетворення після відповідної перебудови.

Завдання логічної емуляції. При налагодженні пристроїв традиційно користувалися як виготовлення прототипу, так і програмними моделями. Виготовлення прототипу – складне і дороге завдання, але з його допомогою можна вести тестування з реальними сигналами та на високих швидкостях, спостерігаючи фактичні можливості пристрою. Програмне моделювання позбавлене вказаних переваг, але простіше та дешевше. Моделі легко модифікуються для видалення помилок у проєкті та в них забезпечується хороше спостереження процесів в об'єкті дослідження.

Застосування ПЛІС у завданнях логічної емуляції дає поєднання переваг обох класичних підходів. Система з мікросхем програмованої логіки легко створюється та модифікується, але може працювати з реальними сигналами та частотами. Водночас програмні моделі та макетування не закреслюються появою мікросхем, що ре програмуються. Створення програмної моделі може залишитися найшвидшим і найдешевшим варіантом емуляції, а макетування відрізняється найповнішим відображенням властивостей об'єкта. Таким чином, застосування ПЛІС добре доповнює колишні методи розробки та тестування схем.

Побудова систем, що динамічно ре конфігуруються. Від простих реконфігурованих систем системи з динамічною реконфігурацією (Run-Time Reconfiguration) відрізняються тим, що в них потрібна швидка зміна налаштувань. Звичайне настроювання із введенням у мікросхему послідовного потоку бітів або байтів займає досить великий час. У системах, що динамічно ре конфігуруються, вже є (зберігається) набір попередньо завантажених

налаштувань, що швидко змінюють один одного відповідно до вимог реалізованого алгоритму.

Динамічно реконфігурована мікросхема може мати практично будь-яке число налаштувань, яке обмежується лише місткістю пам'яті для їх зберігання. Пристрої з динамічною реконфігурацією вже застосовуються фактично і дають очікуваний позитивний ефект. Проблеми побудови систем на мікросхемах з динамічною реконфігурацією зараз активно досліджуються.

У сучасній літературі [14, 15] порушується питання про побудову FPGA-процесорів з іншими проти мікропроцесорів властивостями. Алгоритми роботи процесора завантажуються в FPGA принципово подібно до завантаження в пам'ять мікропроцесорної системи виконуваної програми. Але на противагу мікропроцесорній системі виникає сильно виражений паралелізм на рівні логічних блоків з найпростішими операціями (типу відтворення функції від даного числа аргументів). Такі FPGA-процесори можуть давати хороші результати при паралельній обробці даних, де велика кількість змінних перетворюється подібним чином.

Ефективність схем із програмованою структурою стимулює швидке зростання відповідної галузі промисловості та обсягів їх виробництва, а також наукових досліджень щодо розвитку їх архітектур, схемотехніки та алгоритмів вирішення практичних завдань.

3.2 Синтез загальної структури системи на ПЛІС

Проектування структури ПЛІС виконується на спеціальній мові опису обладнання (VHDL, Verilog) або за допомогою графічних засобів з подальшою генерацією програми формування структури. Розробку алгоритму блоку керування двигуном з його налагодженням та оптимізацією можна виконати в Matlab/Simulink. Також компанія-виробник програмованої логіки надає багато ІР-блоків. Це готові функціональні блоки, які виконують стандартні функції

(наприклад, ПД-регулятор) і для свого використання потребують лише налаштування під конкретне завдання.

Інтегрована система розробки дозволяє проектувати пристрій, моделювати роботу та налагоджувати готовий пристрій. Завдяки цьому досягається висока швидкість розробки, модернізації та діагностики обладнання.

Взаємодії між блоками можуть бути досить складними. У цьому випадку краще, щоб все управлялося ядром процесора і програмою на одній з мов високого рівня, наприклад С. Таке ядро може бути як програмним, побудованим з логічних блоків, так і апаратним – створеним всередині кристалу FPGA. Як програмні ядра використовуються Nios II від Altera або MicroBlaze від Xilinx. Нещодавно почали впроваджувати апаратні ядра ARM, такі як Cortex-M1 і Cortex-A9. Це дозволяє створити реальну систему на кристалі, в якій будуть поєднані гнучкість і простота програмної реалізації алгоритмів і швидкодія апаратних блоків (рис. 3.1).

Обмін даними здійснюється за допомогою CAN-шини, яка об'єднує окремі блоки управління в загальну систему.

Таким чином, розроблений модуль буде побудований на базі FPGA. Підключення до пристрою відображення буде здійснюватися через інтерфейс VGA.

Для цих цілей виберемо ПЛІС II EP2C5T144C8N – 144, виробництва фірми ALTERA (на разі – INTEL). Такий вибір обумовлений хорошими техніками – економічними показниками інтегральних схем цього типу. Крім того, дана мікросхема є конфігурується, що є важливою деталлю при експериментальному налагодженні даного модуля відображення інформації.

Фірмою ALTERA розроблені кілька безплатних середовищ проектування для створення проєктів на базі IC їх виробництва: MAX + PLUS II та QUARTUS II.

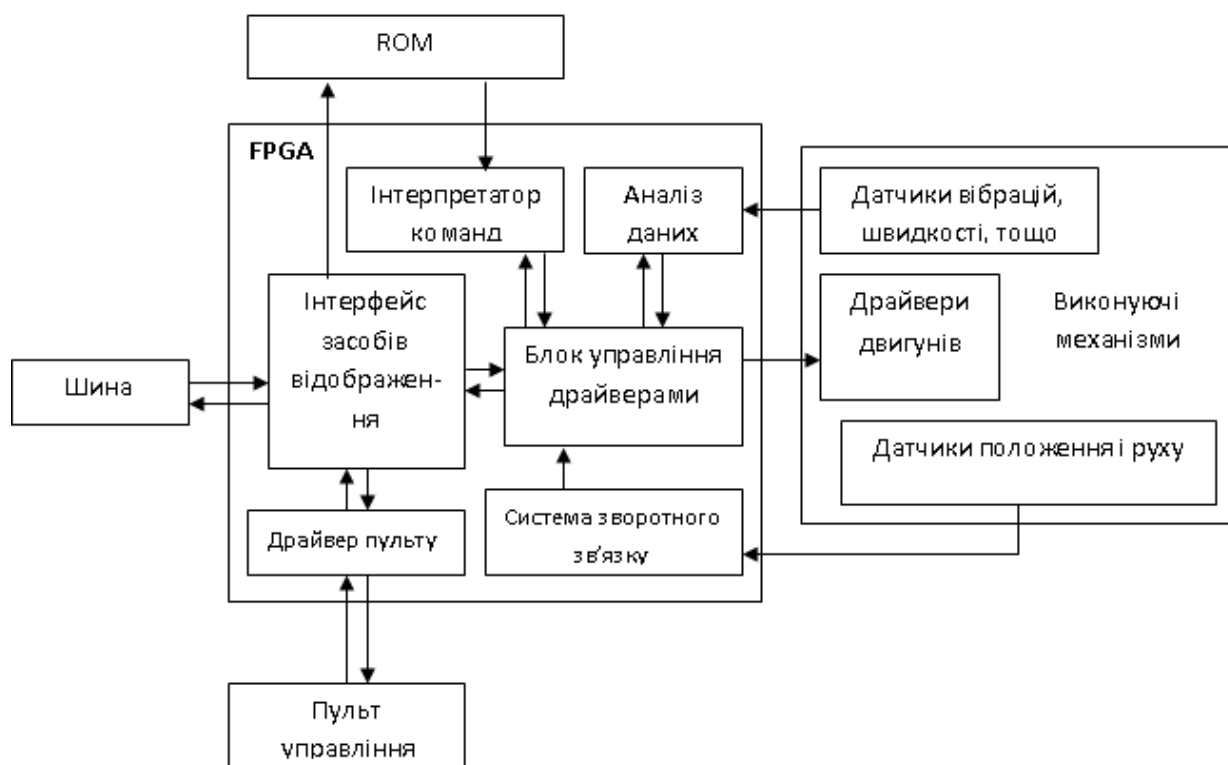


Рисунок 3.1 – Блок-схема структури пристрою керування

При виконанні даного проєкту використовувалася САПР QUARTUS II версії 9.0. Сам проєкт був виконаний за допомогою використання мови опису апаратури VHDL.

Зовнішній вигляд основного вікна САПР QUARTUS II наведено на рис. 3.2.

Процес створення проєкту в QUARTUS II складається з наступних етапів:

- створення опису архітектури проєктованого модуля для підключення мікроконтролера до телевізійного пристрою відображення (система QUARTUS II передбачає кілька способів створення проєкту: графічний принципний; сигнальний; текстовий, з використанням мов VHDL або Verilog HDL);
- складання проєкту, представленого в одній із зазначених вище форм;
- комп'ютерне моделювання проєкту з використанням вбудованого симулятора QUARTUS II;

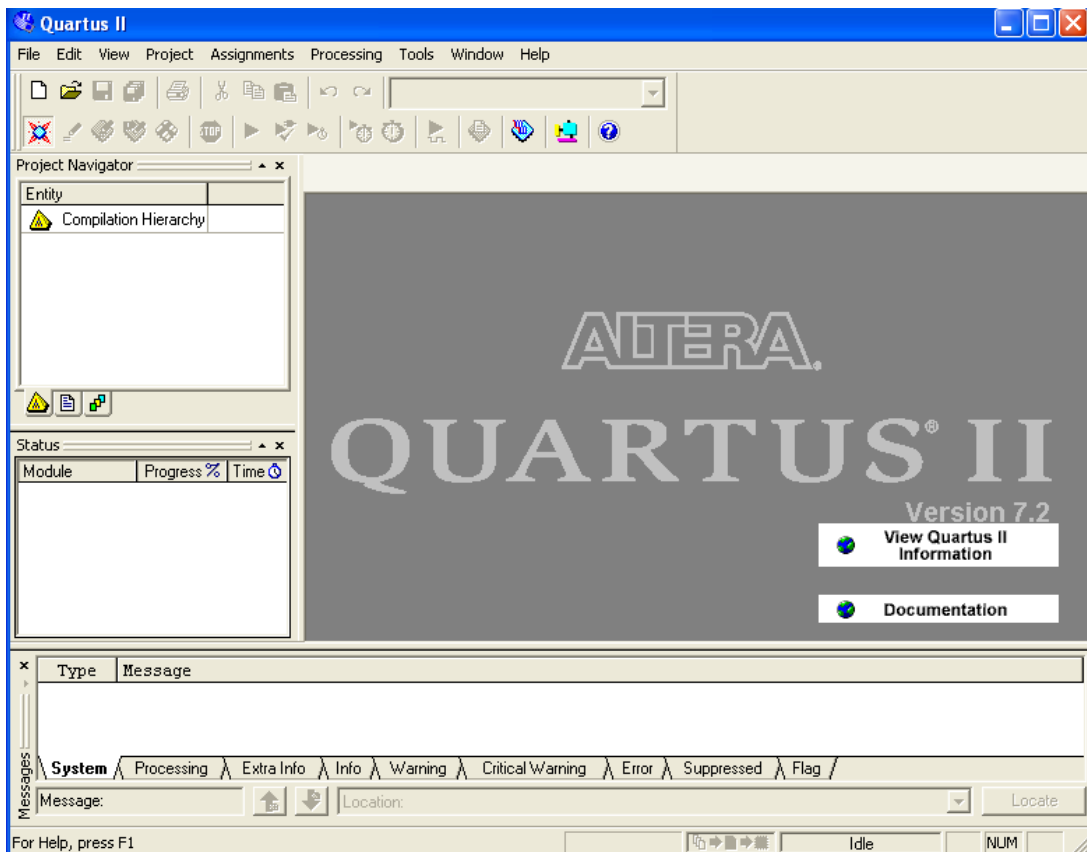


Рисунок 3.2 – Приклад вікна САПР QUARTUS II

– після того, як часові характеристики проекту, отримані шляхом моделювання, задовольняють вимогам, відбувається перепризначення висновків ПЛІС за допомогою вбудованої програми з подальшою перекомпіляцією проекту;

– програмування FPGA за допомогою вбудованого програматора QUARTUS II.

3.3 Проектування конструкції блока контролю та управління

Проектування друкованих плат приладу здійснюється за допомогою системи автоматизованого проектування.

Проектування, при якому всі процедури здійснюються через взаємодію людини і комп'ютера, називають автоматизованим, на відміну від ручного (без

використання комп'ютера) або автоматичного (без участі людини на проміжних етапах). Системою, яка реалізує автоматизоване проєктування, є система автоматизованого проєктування (САПР).

Друковані плати призначені для монтажу елементів принципової схеми різних пристроїв і поділяються на кілька типів за різними параметрами. Плати розрізняють за матеріалом, з якого вони виготовлені, за кількістю шарів металізованих провідників, за сферою застосування, умовами експлуатації, а також за іншими критеріями. Односторонні друковані плати використовуються виключно для односторонньої установки елементів в гладкі (недеталізовані) отвори. Весь електромонтаж виконується в один шар:

- загальноприйнято, що перший (верхній) шар - це той, на якому розташовані елементи. При розміщенні елементів з двох сторін за верхній шар приймається шар, на якому розташовується роз'єм або інші зовнішні комутаційні пристрої (елементи кріплення, колодки, плати і т.д.).

- в односторонніх друкованих платах використовуються перемички з дроту (зазвичай мідного, лудженого одножильного) для трасування ланцюгів, що перетинаються. Вони є елементами конструкції, тому їх показують на кресленнях, пишуть в специфікації і т. д. Номенклатура перемичок повинна бути мінімальною.

- односторонні друковані плати забезпечують більшу точність виконання струмопровідного малюнка та його поєднання з отворами і є найдешевшим класом друкованих плат. Надійність друкованої плати і механічна міцність кріплення елементів не висока. Щоб уникнути розшарування друкованих провідників, всі елементи слід монтувати без зазорів між корпусом елемента та друкованою платою. Для підвищення міцності елементів кріплення можливе виготовлення односторонніх друкованих плат з металізацією отворів, але вартість друкованих плат буде порівнянна з двосторонніми.

Багатошарові друковані плати (РСВ) характеризуються високою здатністю трасування та щільністю елементів. Практично не мають обмежень

по встановленим елементам (мікросхеми будь-якого ступеня інтеграції, елементи поверхневого монтажу і т.д.).

Розробка друкованих плат здійснювалася за допомогою системи автоматизованого проєктування Altium Designer (рис. 3.3).

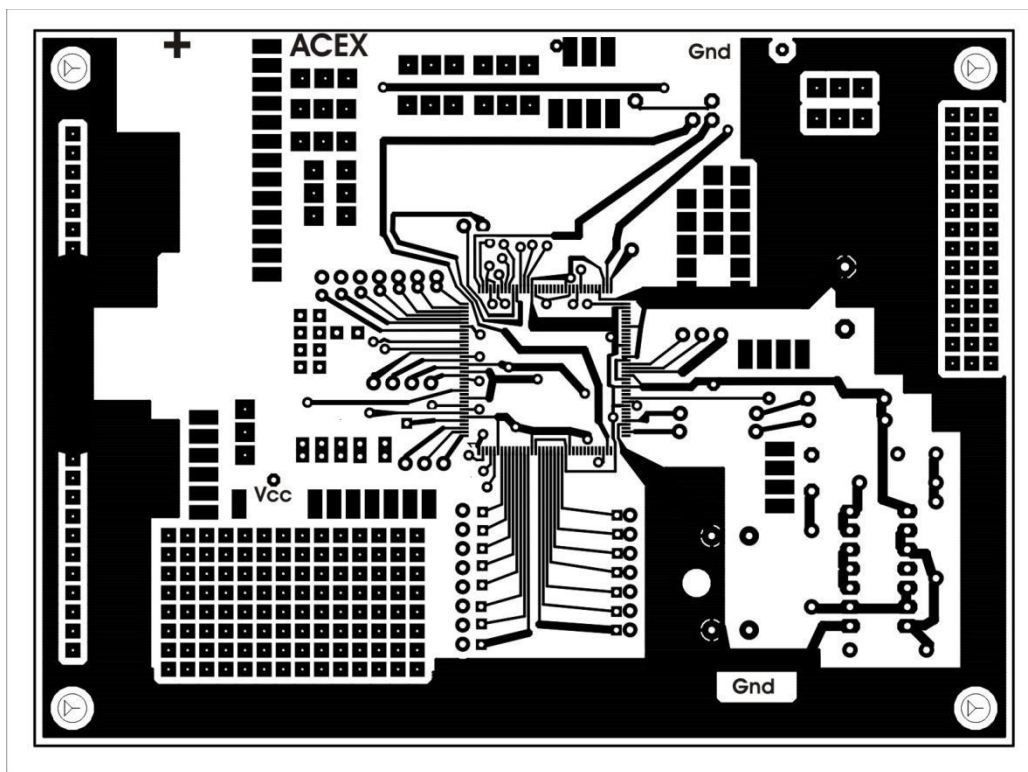


Рисунок 3.3 – Топологія друкованої плати пристрою контролю

На рис. 3.4 надано електричну схему пристрою вимірювання параметрів двигуна, як закінченого блоку з виходом на індикатор.

Висновки по розділу

Програмовність користувачем, тобто реалізованість індивідуального проєкту на основі стандартної мікросхеми FPGA, забезпечується наявністю в схемі безлічі двополюсників, провідність яких може бути задана або дуже малою (це відповідає розімкнутому ключу), або досить великий (це відповідає замкнутому ключу). Стану ключів задають конфігурацією схемою, формує-мій

ВИСНОВКИ

В ході виконання кваліфікаційної роботи магістра було розглянуто методи і засоби побудови сучасних систем контролю для автотранспорту.

На основі аналізу перспективних рішень було вироблено напрями шляхів розв'язання поставленої задачі, методів та засобів її реалізації.

Розроблені структурні схеми блоків пристрою контролю автомобільної системи з урахуванням сучасних вимог до транспортних засобів.

Показано доцільність застосування програмованої логіки в системах безпеки і контролю агрегатів транспортних засобів.

Також із застосуванням сучасної системи автоматизованого проектування розроблено конструкції окремих блоків автомобільної системи на основі мікросхем FPGA.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Угрюмов Е. П. Цифровая схемотехника: учеб. пособие для вузов. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2010. – С. 3 – 5.
2. Муравьев А. Н. Разработка цифровых схем на базе программируемой логики: учеб. пособие. Самара, 2010. – С. 4 – 9.
3. Фрайден Дж. Современные датчики. Справочник [Текст] / Дж. Фрайден. –М.: Техносфера, 2005. – 587с.
4. Евстифеев, А.В. Микроконтроллеры AVR семейств Tiny и Mega фирмы «ATMEL» [Текст] / А.В. Евстифеев. – М.: Додэка-XXI, 2004. – 562 с.
5. Шпак, Ю.А. Программирование на языке С для AVR и PIC микроконтроллеров [Текст] / Ю.А. Шпак. – К.: МК-Пресс, 2006. – 400 с.
6. Ицкович, Э. Л. Современные интеллектуальные датчики общепромышленного назначения на рынке СНГ [Текст] / Э. Л. Ицкович. -М.: 2005 г.
7. Хоровиц П., Хилл У. , Искусство схемотехники [Текст]/ Хоровиц П., Хилл У. – М : 2009. – 704 с.
8. Стешенко, В.Б. ПЛИС фирмы ALTERA: элементная база, система проектирования и языки описания аппаратуры [Текст] / В.Б.Стешенко. – М.: Издательский дом «Додэка», 2002 – 576 стр.
9. Сыроева С. Автомобильные акселерометры // Компоненты и технологии. 2005. № 8, 9. 2006. № 2–5.
10. Electronics-Driver Assistance Systems. Техническая информация Hella. Hella 2003, 2004, 2005. www.hella.com
11. T. Yoshida, H. Kuroda, T. Nishigaito. Adaptive Driver-assistance Systems. http://www.hitachi.com/ICSFiles/afieldfile/2004/11/26/r2004_04_104_1.pdf
12. New Directions in Automotive Smarts. McConnell D., Continental Automotive Systems. Sensors, Apr 1, 2006.
13. Automotive Cameras for Safety and Convenience Applications. White Paper by SMaL Camera Technologies, Inc. 2004 Version.

14. Perry D. L. VHDL: Programming by Example. – Fourth Edition. McGraw-Hill, 2002. – 497 p.
15. Asheden P. J. The designer's guide to VHDL. – San Francisco: Morgan Kaufman Publishers. – 1996.