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

Факультет електроніки

Кафедра мікроелектроніки

До захисту допущено:

Завідувач кафедри

_____ Дмитро ТАТАРЧУК

«___»_____20__ р.

Дипломна робота

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Мікро- та наноелектроніка»
спеціальності 153 «Мікро- та наносистемна техніка»

на тему: «Модель нейромережі на базі системи на кристалі з матрицями аналогових та цифрових блоків»

Виконала:
студентка IV курсу, групи ДП-02
Ступаченко Олександра Сергіївна _____

Керівник: доц.каф.МЕ, к.ф.-м.н., б/з,
Заворотний Віктор Федорович _____

Консультант з нормоконтролю: доц.каф.МЕ, к.ф.-м.н., с.н.с.,
Свєчніков Георгій Сергійович _____

Консультант з інформаційних питань: доц.каф.МЕ, к.т.н., доц.,
Діденко Юрій Вікторович _____

Рецензент: доц.каф. ЕІ, к.т.н., доц.,
Казміренко Віктор Анатолійович _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2024 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет електроніки

Кафедра мікроелектроніки

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 153 «Мікро- та наносистемна техніка»

Освітньо-професійна програма «Мікро- та наноелектроніка»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Дмитро ТАТАРЧУК

«___»_____20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Ступаченко Олександрі Сергіївні

1. Тема роботи «Модель нейромережі на базі системи на кристалі з матрицями аналогових та цифрових блоків», керівник роботи Заворотний Віктор Федорович, доц.каф.МЕ, к.ф.-м.н., б/з, затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Термін подання студентом роботи _____

3. Вихідні дані до роботи

Мова програмування С, середовище програмування PSoC Creator, цільова платформа PSoC (Programmable System-on-Chip), обрана плата CY8CKIT-042, тип досліджуваного об'єкта штучний нейрон. Теми дослідження: Принципи роботи штучних нейронів і нейромереж, проектування моделі нейрона і нейромережі на базі PSoC.

4. Зміст роботи

1 Огляд PSoC

2 Принципи роботи штучних нейронів і нейромереж

3 Проектування моделі нейрона і нейромережі на базі PSoC

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

Презентація до доповіді

7. Дата видачі завдання 15.04.2024

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Відмітка про виконання
1	Отримання завдання та проходження ТБ і ППБ	15.04.2024	
2	Проведено аналіз архітектури PSoC	26.04.2024	
3	Досліджено принципи роботи штучних нейронів і нейромереж	03.05.2024	
4	Вивчено можливості плати CY8CKIT-042	10.05.2024	
5	Проектування моделі нейрона і нейромережі на базі PSoC	17.05.2024	
6	Підготовка та оформлення презентації для доповіді	24.05.2024	
	Обробка результатів та оформлення роботи	31.05.2024	

Студент Олександра Ступаченко

Керівник Віктор Заворотний

РЕФЕРАТ

Дипломна робота присвячена дослідженню можливостей реалізації апаратної моделі нейрона та нейромережі на базі програмованої системи на кристалі (PSoC) з використанням плати CY8CKIT-042. Актуальність теми зумовлена широкими можливостями інтеграції аналогових та цифрових блоків на одній платформі, що відкриває перспективи для створення ефективних вбудованих систем.

Ключові слова: PSoC, нейронні мережі, аналогові блоки, цифрові блоки, алгоритм зворотного поширення помилки, CY8CKIT-042, програмована система на кристалі.

Метою дипломної роботи є дослідження архітектури PSoC, розробка та реалізація моделі нейрона і нейромережі з використанням програмованих аналогових та цифрових блоків плати CY8CKIT-042.

Об’єкт дослідження: Програмована система на кристалі (PSoC) та її можливості для реалізації апаратних нейронних мереж.

Методом дослідження є теоретичний аналіз архітектури PSoC, проектування та тестування моделі нейрона і нейромережі на базі програмованих аналогових та цифрових блоків PSoC.

У ході роботи було проведено аналіз теоретичних основ PSoC, описано архітектуру плати CY8CKIT-042, розглянуто принципи роботи штучних нейронів та нейромереж. Було створено та налаштовано нейронну мережу, реалізовано алгоритми зворотного поширення помилки для навчання мережі. Проведено тестування та оцінка роботи моделі.

В першому розділі описано загальну структуру PSoC та ключові компоненти плати CY8CKIT-042. Проведено аналіз програмованих аналогових та цифрових блоків, можливості їхньої інтеграції та налаштування. Описано особливості архітектури PSoC, що дозволяють ефективно поєднувати аналогові та цифрові компоненти на одній платформі.

У другому розділі розглянуто теоретичні основи штучних нейронних мереж. Детально описано архітектуру нейронів, типи нейронних мереж, їх структурні компоненти та функціонування. Проаналізовано математичні моделі та функції активації, що використовуються для навчання нейронних мереж, включаючи алгоритми зворотного поширення помилки.

Третій розділ присвячено практичному застосуванню нейронної мережі на базі PSoC. Описано процес проектування системи, вибір компонентів та модулів, налаштування проекту в середовищі PSoC Creator. Детально розглянуто реалізацію нейронної мережі, тренування, валідацію та тестування на реальних даних. Описано задачі класифікації та обробки вхідних даних для подальшого розпізнавання нейронною мережею.

Загальний обсяг роботи – 63 сторінок, 9 рисунків, 16 бібліографічних найменувань.

ABSTRACT

This thesis focuses on designing and developing a neural network model using the CY8CKIT-042 Programmable System-on-Chip. The main objective is to explore the capabilities of the PSoC platform for implementing a neural network.

Keywords: PSoC, neural network, hardware implementation, digital and analog components, PSoC Creator.

The work is divided into three main sections:

Overview of PSoC: This section reviews the architecture of the PSoC, including its programmable analog and digital blocks, and highlights its features and advantages.

Principles of Artificial Neurons and Neural Networks: This section covers the basics of artificial neurons and neural networks, including their structure, types, and learning algorithms.

Design of the Neural Network Model on PSoC: This section details the process of designing the neural network model using the CY8CKIT-042 board. It includes the configuration of components, implementation of the neural network, and testing of the model.

The practical implementation involves creating a simple neural network to control an LED based on sensor input, demonstrating the PSoC's capability in executing machine learning tasks.

The total volume of the work is 63 pages, 9 figures, 16 bibliographic entries.

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧКИ

АН – Аналоговий нейрон

ШІМ – Широтно-імпульсна модуляція

ЦАП – Цифро-аналоговий перетворювач

АЦП – Аналого-цифровий перетворювач

НН – Нейронна мережа

НП – Нейронний процесор

СПК – Система на програмованому кристалі

ADC – Analog-to-Digital Converter (Аналого-цифровий перетворювач)

ARM – Advanced RISC Machine (Тип мікропроцесора)

DAC – Digital-to-Analog Converter (Цифро-аналоговий перетворювач)

DMA – Direct Memory Access (Прямий доступ до пам'яті)

GPIO – General Purpose Input/Output (Універсальні лінії введення/виведення)

PSoC – Programmable System-on-Chip (Програмована система на кристалі)

PWM – Pulse Width Modulation (Широтно-імпульсна модуляція)

UART – Universal Asynchronous Receiver-Transmitter (Універсальний асинхронний приймач-передавач)

UDB – Universal Digital Blocks (Універсальні цифрові блоки)

ЗМІСТ

ВСТУП

У сучасному світі технології штучного інтелекту (ШІ) та нейронних мереж стрімко розвиваються та знаходять широке застосування в різних галузях – від медицини до робототехніки. Основою багатьох систем ШІ є нейронні мережі, які моделюють роботу людського мозку та здатні самонавчатися на основі отриманих даних. Впровадження таких технологій дозволяє автоматизувати складні процеси, покращити точність прогнозування та прийняття рішень.

Однією з перспективних платформ для реалізації нейронних мереж є програмовані системи на кристалі (PSoC) від компанії Infineon Technologies (раніше Cypress Semiconductor). Платформи PSoC поєднують у собі програмовані аналогові та цифрові блоки, що дозволяє створювати високоефективні та компактні рішення для різних задач.

Дипломна робота присвячена розробці моделі нейрона і нейромережі на базі плати CY8CKIT-042, що використовує технологію PSoC 4. Обрана платформа забезпечує гнучкість у проектуванні та дозволяє інтегрувати аналогові та цифрові компоненти на одній мікросхемі, що робить її ідеальним вибором для апаратного моделювання нейронних мереж.

Актуальність теми: Розробка апаратних моделей нейронних мереж є важливою складовою сучасних технологій ШІ, оскільки вони дозволяють забезпечити високу продуктивність та енергоефективність. Використання платформи PSoC для реалізації таких моделей відкриває нові можливості для створення ефективних вбудованих систем.

Мета роботи: Метою даної дипломної роботи є дослідження можливостей PSoC для реалізації апаратної моделі нейрона та нейромережі, а також розробка та тестування такої моделі на базі плати CY8CKIT-042.

Об'єкт дослідження: Програмована система на кристалі (PSoC) та її можливості для реалізації апаратних нейронних мереж.

Предмет дослідження - методи проектування та реалізації моделей нейронних мереж на базі платформи PSoC.

Робота демонструє можливості платформи PSoC для створення апаратних моделей нейронних мереж, що дозволяє поєднувати аналогові та цифрові компоненти на одній мікросхемі та забезпечувати високу ефективність обробки даних.

Результати роботи можуть бути використані для подальшого розвитку вбудованих систем з елементами штучного інтелекту, а також для навчання студентів та фахівців у галузі електроніки та комп'ютерних технологій.

1 Огляд PSoC: загальний огляд архітектури PSoC, включаючи програмовані аналогові та цифрові блоки

1.1 Система на кристалі (SoC)

Система на кристалі, або Система на чипі (System-on-a-chip, SoC), — дизайн електронної схеми, яка вміщує функціональні складові цілого пристрою (наприклад, комп'ютера) на одній мікросхемі. Залежно від призначення SoC може оперувати як цифровими сигналами, так і аналоговими, аналого-цифровими, а також частотами радіодіапазону. Типовим застосуванням таких схем є широке різноманіття вбудованих систем. Якщо не вдається розмістити всі необхідні схеми на одному напівпровідниковому кристалі, то використовується схема із декількох кристалів, розміщених в одному корпусі (System in Package — SiP). SoC вважається вигіднішою конструкцією, оскільки дозволяє збільшити відсоток придатних схем при виготовленні та спростити конструкцію корпуса. [1]

Типова SoC включає:

– Мікроконтролер, мікропроцесор чи процесор цифрових сигналів. Деякі схеми обладнані більше ніж одним процесором, тоді їх ще називають MPSoC (Multiprocessor System-on-Chip);

– Блок пам'яті, який може працювати із такими типами пам'яті: ROM, RAM, EEPROM та флеш;

– Джерело опорної частоти, наприклад, кварцові генератори та ланцюги з оберненим зв'язком (phase-locked loops — цифрова система фазової автопідстройки частоти);

– Таймери, лічильники та схеми затримок після увімкнення;

– Стандартні інтерфейси для зовнішніх пристроїв: USB, FireWire, Ethernet, UART, SPI;

– Входи та виходи цифро-аналогових і аналого-цифрових перетворювачів;

– Регулятори напруги та стабілізатори живлення.

Блоки між собою можуть з'єднувати за допомогою шини власної розробки чи стандартної конструкції, наприклад, стандартизована AMBA в чипах ARM. Якщо в складі чипу є контролер прямого доступу пам'яті (DMA), то за його допомогою можна заносити дані з зовнішніх пристроїв безпосередньо до пам'яті чипа, не витрачаючи процесорних ресурсів.

Система на кристалі є ключовою технологією, яка сприяє розвитку сучасних електронних пристроїв. Вона об'єднує всі необхідні компоненти на одному кремнієвому чипі, включаючи центральний процесор, оперативну пам'ять, графічний процесор, контролери введення/виведення та інші елементи. Особливу роль у SoC відіграють матриці аналогових і цифрових блоків, які забезпечують високу функціональність та ефективність роботи пристрою.

Аналогові блоки в SoC виконують функції обробки аналогових сигналів. Вони можуть включати підсилювачі, перетворювачі аналогово-цифрові (ADC) та цифро-аналогові (DAC). Ці блоки забезпечують взаємодію з реальним світом через зчитування та обробку сигналів з датчиків, мікрофонів, камер та інших пристроїв.

Цифрові блоки відповідають за обробку цифрових даних. Вони можуть включати процесори, контролери, пам'ять та інші логічні схеми. Ці блоки забезпечують обчислювальні функції, необхідні для виконання програмного забезпечення та керування пристроєм. [2]

Переваги SoC:

– Об'єднання багатьох компонентів на одному чипі дозволяє зменшити розміри пристроїв та спростити їхню конструкцію. Це робить можливим створення компактних і легких електронних пристроїв;

– Інтеграція компонентів на одному чипі знижує споживання енергії, що є критично важливим для портативних пристроїв, таких як смартфони, планшети та розумні годинники;

– Масове виробництво SoC знижує вартість виготовлення пристроїв порівняно з використанням окремих компонентів. Це дозволяє знизити собівартість електронних пристроїв.

SoC широко використовуються в мобільних телефонах, планшетах, розумних годинниках, побутовій електроніці, автомобільних системах та інших вбудованих системах. SoC використовуються в портативних медичних приладах, таких як монітори серцевого ритму, глюкометри та інші пристрої для діагностики та моніторингу здоров'я. Вони є основним компонентом у багатьох IoT пристроях, таких як розумні термостати, розумні замки, датчики для домашньої автоматизації та промислові IoT додатки. Знаходять застосування в системах відеоспостереження, розумних камерах та інших пристроях безпеки. Забезпечують високу продуктивність і низьке енергоспоживання, що робить їх ідеальними для використання в дронах, роботах та автономних системах. В сучасних ігрових консолях та мультимедійних програвачах, також використовуються в маршрутизаторах, модемах, мережевих комутаторах та інших пристроях для забезпечення швидкої та ефективної обробки даних. В авіації та космічних технологіях SoC використовуються для управління польотом, навігації та інших критично важливих систем.

1.2 Cypress PSoC

PSoC (програмована система на кристалі) — це сімейство інтегральних схем мікроконтролерів від Cypress Semiconductor . Ці мікросхеми включають ядро ЦП і масиви змішаних сигналів конфігурованих інтегрованих аналогових і цифрових периферійних пристроїв.

Інтегральна схема PSoC складається з ядра, настроюваних аналогових і цифрових блоків, а також програмованої маршрутизації та з’єднання. Конфігуровані блоки в PSoC є найбільшою відмінністю від інших мікроконтролерів. PSoC має три окремі простори пам’яті: сторінковий SRAM для даних, флеш-пам’ять для інструкцій і фіксованих даних, а також регістри вводу/виводу для управління та доступу до конфігурованих логічних блоків і функцій. Прилад створено за технологією SONOS. [3]

Основні компоненти PSoC (Programmable System-on-Chip):

– Центральний процесор (CPU) є ядром будь-якої SoC, відповідаючи за виконання основних обчислювальних завдань. Він обробляє інструкції програмного забезпечення та керує іншими компонентами системи;

– Графічний процесор (GPU) відповідає за обробку графічних даних, забезпечуючи високу продуктивність візуальних обчислень. Це особливо важливо для мобільних пристроїв, ігор та мультимедійних додатків;

– Оперативна пам'ять (RAM) забезпечує тимчасове зберігання даних, які використовуються процесором у процесі виконання завдань. Вона має високі швидкості читання та запису, що дозволяє забезпечити швидкодію системи;

– Контролери введення/виведення (I/O) відповідають за взаємодію з периферійними пристроями, такими як USB, Ethernet, дисплеї та інші. Вони забезпечують обмін даними між SoC та зовнішніми пристроями.

PSoC нагадує ASIC: блокам можна призначати широкий спектр функцій і з'єднувати їх на чіпі. Але на відміну від ASIC, для створення спеціальної конфігурації не потрібен спеціальний виробничий процес — лише код запуску, створений за допомогою IDE Cypress PSoC Designer або PSoC Creator. PSoC нагадує FPGA тим, що під час увімкнення живлення його потрібно налаштувати, але ця конфігурація відбувається шляхом завантаження інструкцій із вбудованої флеш-пам'яті. PSoC найбільше нагадує мікроконтролер у поєднанні з PLD і програмованим аналогом. Код виконується для взаємодії з визначеними користувачем периферійними функціями (компонентами), використовуючи автоматично створені API та підпрограми переривання. PSoC Designer або PSoC Creator створюють код конфігурації запуску. Обидва інтегрують API, які ініціалізують вибрані користувачем компоненти відповідно до потреб користувачів у графічному інтерфейсі Visual-Studio.

Використовуючи настроювані аналогові та цифрові блоки, дизайнери можуть створювати та змінювати вбудовані програми зі змішаним сигналом. Цифрові блоки є кінцевими автоматами, які налаштовані за допомогою регістрів блоків. Є два типи цифрових блоків: Digital Building Blocks (DBBxx) і Digital Communication Blocks (DCBxx). Тільки комунікаційні блоки можуть містити модулі користувача послідовного вводу-виводу, такі як SPI, UART тощо. Кожен цифровий блок вважається 8-бітним ресурсом, який розробники можуть налаштувати за допомогою попередньо створених цифрових функцій або модулів користувача (UM), або, об'єднавши блоки, перетворити їх на 16-, 24- або 32-бітні ресурси. Об'єднання UM разом – це те, як створюються 16-бітні ШІМ і таймери. Існує два види аналогових блоків. Блоки безперервного часу (CT) складаються зі схеми операційного підсилювача та позначаються як ACBxx, де xx дорівнює 00–03. Інший тип — блоки перемикачів (SC), які дозволяють проходити складний аналоговий сигнал і позначаються ASCxy, де x — рядок, а y — стовпець аналогового блоку. Дизайнери можуть модифікувати та персоналізувати кожен модуль відповідно до будь-якого дизайну.

Щодо програмованої маршрутизації та з'єднання, гнучка маршрутизація масивів змішаних сигналів PSoC дозволяє розробникам вільніше направляти сигнали до та з контактів вводу-виводу, ніж у багатьох конкуруючих мікроконтролерів. Глобальні шини дозволяють мультиплексувати сигнали та виконувати логічні операції. Cypress припускає, що це дозволяє дизайнерам легше та швидше налаштовувати конструкцію та вносити вдосконалення з меншою кількістю перепроектувань друкованої плати, ніж підхід із цифровим логічним вентилем або конкуруючі мікроконтролери з більшою кількістю фіксованих функціональних контактів.

Існує п’ять різних сімейств пристроїв, кожна з яких базується на окремому ядрі мікроконтролера:

- PSoC 1 — серія CY8C2xxxx — ядро M8C;
- PSoC 3 — серія CY8C3xxxx — ядро 8051;
- PSoC 4 — серія CY8C4xxxx — ядро ARM Cortex-M0;
- PSoC 5/5LP — серія CY8C5xxxx — ядро ARM Cortex-M3;
- PSoC 6 — серія CY8C6xxxx — ядро ARM Cortex-M4 із доданим ядром ARM Cortex-M0+ (у деяких моделях).

Цифрова система складається з цифрових рядків у масиві блоків і глобальних, масивних і рядкових цифрових з’єднань (GDI, ADI та RDI відповідно). Цифрові блоки представлені в рядах по чотири, де кількість блоків залежить від пристрою PSoC (рис. 1.1 - Характеристики пристрою PSoC). Це дозволяє нам вибрати оптимальні системні ресурси для вашої програми. [3]

Цифрові блоки можна підключити до будь-якого GPIO через серію глобальних шин, які можуть направляти будь-який сигнал на будь-який контакт. Шини також дозволяють мультиплексувати сигнали та виконувати логічні операції. Ця можливість конфігурації звільняє проекти від обмежень фіксованого периферійного контролера.

Аналогова система складається з аналогових стовпців у масиві блоків, аналогових посилань, мультиплексування аналогового введення та аналогових

драйверів. Аналоговий системний блок складається з чотирьох аналогових стовпців і до 12 аналогових блоків, залежно від характеристик вашого пристрою PSoC (рис. 1.1). Кожен конфігурований блок складається зі схеми операційного підсилювача, що дозволяє створювати складні потоки аналогового сигналу.

У сімействі програмованих систем на чіпі PSoC є багато груп мікросхем. Окрім розрізнення цих груп за номерами деталей PSoC, кожну групу PSoC легко відрізнити за унікальною кількістю цифрових рядків і аналогових стовпців, які вона має. Ця унікальна характеристика є основою для представлення інформації в даній роботі.

Група пристроїв PSoC	Цифровий IO (макс.)	Цифрові рядки	Цифрові блоки	Аналогові входи	Аналогові виходи	Аналогові клони	Аналогові блоки	Кількість SRAM	Кількість Flash
CY8C29x66 CY8CPLC20 CY8CLED16P01	64	4	16	12	4	4	12	2 KB	32 KB
CY8C27x43	44	2	8	12	4	4	12	256 Bytes	16 KB
CY8C24x94	50	1	4	48	2	2	6	1 KB	16 KB
CY8C24x23	24	1	4	12	2	2	6	256 Bytes	4 KB
CY8C24x23A	24	1	4	12	2	2	6	256 Bytes	4 KB
CY8C22x13	16	1	4	8	1	1	3	256 Bytes	2 KB
CY8C21x34	28	1	4	28	0	2	4*	512 Bytes	8 KB
CY8C21x23	16	1	4	8	0	2	4*	256 Bytes	4 KB
CY7C64215	50	1	4	48	2	2	6	1 KB	16 KB
CY7C603xx	28	1	4	28	0	2	4*	512 Bytes	8 KB
CYWUSB6953	28	1	4	28	0	2	4*	512 Bytes	8 KB
CY8CNP1xx	33	4	16	12	4	4	12	2 KB	32 KB

Рисунок 1.1 - Характеристики пристрою PSoC [3]

У наведеній нижче таблиці (рис. 1.2) перераховано ресурси, доступні для певних груп пристроїв PSoC. Позначка або відповідна інформація означає, що для пристрою PSoC доступний системний ресурс. Порожні поля означають, що системний ресурс недоступний. Типи дециматорів: T1 = тип 1, T2 = тип 2. З позначкою «**» - єдині пристрої PSoC, які мають системний ресурс аналогового мультиплексора вводу-виводу або системний ресурс USB.

Номер частини PSoC	USB	Перемикач режимів насоса	Цифрові годинники	I2C	Внутрішня напруга Ref.	POR і LVD	Скидання системи	Дециматор*	Множити Накопичувати	nvSRAM
CY8C29x66		✓	✓	✓	✓	✓	✓	T2	2	
CY8C27x43		✓	✓	✓	✓	✓	✓	T1	1	
CY8C24x94 **	✓		✓	✓	✓	✓	✓	T2	2	
CY8C24x23		✓	✓	✓	✓	✓	✓	T1	1	
CY8C24x23A		✓	✓	✓	✓	✓	✓	T1	1	
CY8C22x13			✓	✓	✓	✓	✓	T1	0	
CY8C21x34 **		✓	✓	✓	✓	✓	✓		0	
CY8C21x23		✓	✓	✓	✓	✓	✓		0	
CY7C64215 **	✓		✓	✓	✓	✓	✓	T2	2	
CY7C603xx **		✓	✓	✓	✓	✓	✓		0	
CYWUSB6953 **		✓	✓	✓	✓	✓	✓		0	
CY8CNP1xx		✓	✓	✓	✓	✓	✓	T2	2	256 kbytes

Рисунок 1.2 - Наявність системних ресурсів для PSoC Ljevices [3]

Варто зауважити, що CY8C21x34, CY7C603xx і CYWUSB6952 є єдиними пристроями PSoC, які мають системний ресурс аналогового мультиплексора вводу-виводу, а CY8C24x94 і CY7C64215 є єдиними пристроями PSoC, які мають системний ресурс USB. [3]

Також в PSoC присутня Bluetooth Low Energy. Починаючи з 2014 року Cypress почав пропонувати пристрої PSoC 4 BLE із вбудованим Bluetooth Low Energy (Bluetooth Smart). Це можна використовувати для створення підключених

продуктів із використанням аналогових і цифрових блоків. [7] Користувачі можуть додавати та налаштовувати модуль BLE безпосередньо в PSoC creator. Cypress також надає повний стек Bluetooth Low Energy, ліцензований Mindtree, з периферійними та центральними функціями. [8] Серія PSoC 6 включає версії з BLE, включаючи функції Bluetooth 5 , включаючи розширений радіус дії або вищу швидкість. Також мають інтерфейси для зовнішніх пристроїв: USB, I2C, SPI, UART, які дозволяють легко інтегрувати PSoC з іншими мікроконтролерами та периферійними пристроями.

1.2.1 PSoC Designer

PSoC Designer — це інтегроване середовище розробки, розроблене компанією Cypress Semiconductor для програмованих систем на кристалі серії 1. Дозволяє користувачам додавати і налаштовувати аналогові та цифрові компоненти за допомогою перетягування на графічну схему. Можливість налаштування аналогових і цифрових блоків. Вбудований редактор коду з підтримкою мов C та асемблера, що дозволяє писати та налагоджувати прошивку. інструменти для налагодження коду, включаючи відладчик та емулятор. Розглянемо більш детально функціонал PSoC Designer. [4]

Мікросхеми PSoC1 фірми Cypress являють собою 8-бітний мікроконтролер, що містить мікропроцесорне ядро і масив цифрових і аналогових блоків, що дозволяє реалізовувати необхідні користувачеві периферійні функції, як стандартні, наприклад ШІМ, АЦП або UART, так і такі незвичайні для мікроконтролерів, як, аналогові фільтри та інструментальні підсилювачі. Завдяки тому, що PSoC1 дозволяють скоротити кількість використовуваних зовнішніх компонентів, це суттєво спрощує процес розробки, здешевлює пристрій та

одночасно підвищує його гнучкість за рахунок можливості перепрограмування у системі або реконфігурування внутрішньої структури прямо в процесі роботи.

Для роботи потрібні пакет PSoC Designer (версія 4.2) та остання версія пакета оновлень (версія 4.2 SP 3), установка якого здійснюється поверх встановленого PSoC Designer 4.2. САПР PSoC Designer є інтегрованим середовищем розробки, що містить редактор вихідних текстів програм, редактор внутрішньої структури програмованої системи та відладчик.

Крім пакета PSoC Designer для розробки знадобиться і програматор, що не входить до його складу, — PSoC Programmer (остання версія 1.22.0.9). Усі програми доступні на офіційному сайті cypress.com.

Існує два основні апаратні засоби, що застосовуються при розробці - CY3215-DK (ICE-Cube) та CY3210-MiniProg1 (рис. 1.3). ICE-Cube є внутрішньосхемним емулятором і використовується для налагодження проекту, а також може здійснювати прошивку мікросхем PSoC. MiniProg є виключно програматором і не підтримує жодних можливостей для налагодження проекту. Його перевага – низька ціна.

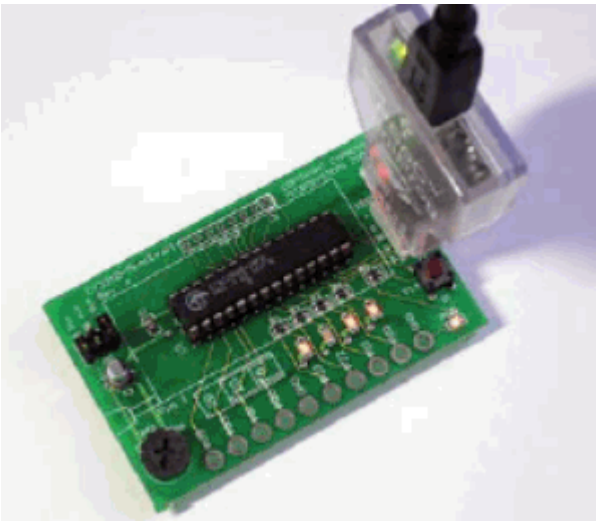

Рисунок 1.3 - Налагоджувальна плата та програматор з комплекту MiniProg [4]

ICE-Cube і MiniProg підключаються до ПК через кабель mini-USB. Під час першого підключення необхідно встановити відповідний драйвер. Драйвери для

ICE-Cube та MiniProg знаходяться у каталозі CypressMicroSystemsPSoCProgrammerdrivers. Створення проекту в PSoC Designer: під час запуску PSoC Designer з'являється вікно, що пропонує відкрити існуючий проект або створити новий. При створенні нового проекту необхідно вибрати тип мікросхеми, на базі якої реалізовуватиметься пристрій. У комплекті MiniProg є два зразки мікросхем CY8C29466-24PXI Виберемо цю мікросхему для створення проекту. Далі необхідно вибрати мову, якою буде створено файл, що містить функцію main - основну функцію проекту. На вибір пропонується мова C або Assembler, проте слід врахувати, що компілятор мови C, що входить до складу PSoC Designer вимагає ліцензування і без реєстраційного коду не працюватиме.

У тому випадку, якщо для розробки купується набір ICE-Cube, то до його складу входить і ліцензія для компілятора C, інакше необхідно купувати ліцензію окремо або користуватися мовою асемблера.

Розглянемо докладніше налаштування глобальних ресурсів мікросхеми. Параметр Power Setting [Vcc/SysClk freq] встановлює напругу живлення, що використовується мікросхемою, і тактову частоту системи. Можливі варіанти вибору цього параметра трохи відрізняються залежно від серії PSoC або від температурного виконання мікросхеми. Для чіпа CY8C29466-24PXI на вибір пропонуються такі комбінації: 3,3 В/24 МГц, 3,3 В/6 МГц, 5 В/24 МГц та 5 В/6 МГц.

Параметр CPU_Clock визначає тактову частоту мікропроцесорного ядра M8C. Ця частота виходить розподілом системної частоти на 2n, де n-від 1 до 8.

Параметр 32K_Select вказує джерело частоти 32 К - внутрішній (Internal) RC-генератор на 32 000 Гц (точність цього генератора невелика), або тактування від зовнішнього (External) годинникового кварцу - в цьому випадку частота становить 32 768 Гц, що дозволяє реалізовувати внутрішній годинник реального часу.

Параметр PLL_Mode відповідає за синхронізацію внутрішнього генератора 24 МГц із зовнішнім кварцом 32768 Гц.

Параметр SleepTimer встановлює період спеціального внутрішнього таймера, який може використовуватися для генерації переривання для виходу з режиму сну,

для реалізації годинника реального часу або інших потреб. Період цього таймера вибирається з наступних варіантів: 512 Гц, 64 Гц, 8 Гц, 1 Гц.

Параметри VC1 та VC2 встановлюють дільники для отримання частот, що використовуються для тактування цифрових та аналогових блоків. Частота VC1 формується із системною розподілом на коефіцієнт від 1 до 16. Частота VC2 формується з VC1 також розподілом на коефіцієнт від 1 до 16.

Параметр VC3 Source дозволяє вибрати джерело формування частоти VC3, яка може використовуватися як для тактування цифрових блоків, так і для генерації переривання. Джерелами для формування VC3 можуть бути частоти VC1 або VC2, системна частота або подвоєна системна частота (при системній частоті 24 МГц стає можливим робота цифрових блоків на частоті 48 МГц).

Параметр VC3 Divider - дільник формування частоти VC3, він лежить у діапазоні від 1 до 256.

Параметр SysClk Source вказує джерело системної частоти - внутрішній генератор або зовнішня частота, що подається на контакт P1 [4].

Параметр SysClk*2 Disable включає або вимикає вбудований помножувач системної частоти.

Параметр Analog Power встановлює потужність джерел опорної напруги і визначає, включені або вимкнені аналогові блоки на конденсаторах, що перемикаються.

Потужність джерел опорних напруг повинна бути обрана рівною або більшою, ніж установки потужності використовуваних у проекті модулів користувача.

Параметр Ref Mux встановлює напругу аналогової землі для аналогових блоків і діапазон вхідних напруг.

Параметр AGndBypass дозволяє використовувати 4 контакт порту P2 для підключення конденсатора для додаткової фільтрації «аналогової землі».

Параметр Op-Amp Bias визначає рівень нахилу перехідної характеристики операційних підсилювачів. Рекомендується встановлювати цей параметр як LOW

на початку розробки проекту. Якщо потрібні вищі частоти роботи, цей параметр може бути встановлений як HIGH.

Параметр A_Buff_Power визначає потужність аналогових буферів.

Параметр SwitchModePump включає або вимикає вбудований регулятор напруги.

Параметр TripVoltage [LVD (SMP)] встановлює рівень напруги визначення факту падіння напруги (переривання Low Voltage Detect), і навіть рівень напруги, при падінні нижче якого починає працювати блок SMP.

Параметр LVD Throttle Back включає чи вимикає автоматичне зменшення швидкості CPU при виявленні падіння напруги.

Параметр Watchdog Enable включає або вимикає сторожовий таймер.

Список глобальних параметрів або їх назви можуть змінюватись в залежності від вибраної серії мікросхем, проте ці зміни незначні. Вище були наведені глобальні параметри для мікросхем 29-ї серії, найпотужнішої за внутрішнім ресурсом.

Налаштування портів дозволяють вибрати режим роботи контакту: порт мікроконтролера, глобальний цифровий вхід або вихід, аналоговий вхід або вихід, а також один із режимів роботи вихідного буфера - strong (стандартний режим), strong slow (вихід з повільним наростанням фронтів), pull-up або pull-down (підтяжка до живлення або землі через резистор 5,6 ком), open-drain low або open-drain high (відкритий стік). Крім того, можливий дозвіл переривання від порту та вибір умови генерації цього переривання: по фронту сигналу (Rising Edge), спаду (Falling Edge) або зміні стану (ChangeFromRead). [4]

1.2.2 PSoC Creator

Окрім PSoC Designer для проектування на програмованих системах на кристалі PSoC Cypress можна використовувати PSoC Creator. Для виконання цієї дипломної роботи та практичної частини ми будемо використовувати саме цю програму. [5]

PSoC Creator — інтегроване середовище розробки, розроблене Cypress Semiconductor (тепер частина Infineon Technologies), яке забезпечує розробникам можливість проектування та програмування програмованих систем на кристалі (PSoC3,4,5,6) із використанням графічного інтерфейсу та автоматичного генерування коду. Дозволяє інтегрувати аналогові та цифрові компоненти в єдину систему, налаштовувати параметри та писати прошивки.

Основні кроки для створення першого проекту в PSoC Creator це конфігурація компонентів. Перетягнувши компоненти з бібліотеки компонентів на схему, щоб створити апаратну частину системи. Написання програмного забезпечення здійснюється в редакторі коду PSoC Creator. Програмне забезпечення може бути інтегровано з апаратною частиною через вказані компоненти і їх конфігурацію.

Додатковими можливостями цієї програми є бібліотека компонентів. PSoC Creator надає доступ до понад 150 компонентів, що дозволяє швидко створювати і налаштовувати проекти. Він дозволяє легко створювати схеми, конфігурувати компоненти і налаштовувати проект без необхідності написання великої кількості коду. Наприклад, для конфігурації PWM можна використовувати графічний інтерфейс для налаштування періоду і коефіцієнта заповнення безпосередньо у PSoC Creator, що значно спрощує процес порівняно з традиційними мікроконтролерами. PSoC Creator дозволяє розробникам ефективно створювати складні проекти, поєднуючи апаратне та програмне забезпечення в єдиному середовищі.

У PSoC Creator різні ресурси PSoC, організовані як графічні елементи, які називаються компонентами. Ці компоненти можна перетягувати на схему для швидкого створення дизайнів. Кожен периферійний пристрій в PSoC доступний як

попередньо перевірений компонент PSoC Creator, наприклад, компонент PWM, ADC, DAC, CAPSENSE™, UART та інші. [6]

Перелік найбільш корисних та часто використовуваних компонентів для розробки вбудованих систем:

– Аналого-цифровий перетворювач (ADC);

Перетворює аналогові сигнали на цифрові дані, які можуть бути оброблені мікроконтролером. ADC дозволяє оцифровувати вхідні аналогові сигнали з високою точністю і швидкістю, що робить його ідеальним для вимірювання температури, тиску та інших аналогових параметрів.

– Цифрово-аналоговий перетворювач (DAC);

Перетворює цифрові дані на аналогові сигнали. DAC дозволяє генерувати точні аналогові вихідні сигнали з цифрових вхідних даних, що є важливим для керування аналоговими пристроями та генерації аудіо сигналів.

– PWM (Pulse Width Modulation);

Широтно-імпульсна модуляція використовуються для генерування імпульсних сигналів з регульованою шириною імпульсу. PWM дозволяє точно керувати двигунами, регулювати яскравість світлодіодів та створювати звукові сигнали, змінюючи ширину імпульсів сигналу.

– CAPSENSE™;

Дозволяє створювати ємнісні сенсорні кнопки, слайдери та інші сенсорні поверхні. CAPSENSE™ забезпечує високу чутливість і точність, що робить її ідеальною для сенсорних панелей, трекпадів та інших сенсорних інтерфейсів.

– Універсальний асинхронний приймач-передавач (UART);

Забезпечує асинхронний послідовний обмін даними. UART дозволяє легко налаштовувати серійний зв'язок між мікроконтролером та іншими пристроями, такими як комп'ютери або інші мікроконтролери.

Дана програма також має програмовані цифрові блоки, відомі як Universal Digital Blocks (UDB). Надає компоненти, створені з UDB, таких як UART, SPI, I2C, таймер, PWM, лічильник, цифрові логічні вентилі (AND, OR, NOT, XOR та інші).

Користувачі можуть навіть створювати власні користувацькі компоненти PSoC Creator за допомогою UDB. PSoC Creator надає унікальну можливість створення користувацьких компонентів, що дозволяє розробникам налаштовувати поведінку системи відповідно до вимог проекту. Ці компоненти можуть бути створені за допомогою графічного інтерфейсу і налаштовуються безпосередньо у PSoC Creator, що значно полегшує процес розробки. [6]

PSoC Creator містить бібліотеки компонентів і приклади проектів, які допомагають розробникам швидко освоїти інструмент і розпочати розробку. Приклади проектів можна знайти у меню File > Code Example. Ці проекти демонструють, як конфігурувати та використовувати компоненти PSoC Creator.

PSoC Creator підтримує роботу з різними середовищами розробки, включаючи Eclipse IDE, IAR Embedded Workbench, Keil µVision та Visual Studio Code. Це дозволяє розробникам використовувати інструменти, з якими вони вже знайомі, і інтегрувати їх з PSoC Creator для максимальної гнучкості та ефективності розробки. [7]

Управління бібліотеками в PSoC Creator здійснюється через бібліотечний менеджер, який дозволяє додавати BSP або middleware бібліотеки до проекту. Це забезпечує швидкий доступ до необхідних ресурсів та спрощує процес інтеграції додаткових функцій у проект.

PSoC Creator тісно інтегрований з ModusToolbox, що дозволяє використовувати його конфігуратори для налаштування ресурсів пристрою та інших компонентів middleware. Це дозволяє розробникам налаштовувати конфігурації та генерувати код, який можна використовувати у проекті.

PSoC Creator має розширену підтримку і документацію. Вона включає керівництво з швидкого старту, довідкові матеріали по компонентам. Користувачі можуть отримати доступ до цих ресурсів через меню Help. Програма забезпечує розширену технічну підтримку, включаючи системне довідкове керівництво користувача, відеоуроки та систему керування документацією. Користувачі

можуть отримати допомогу через офіційний вебсайт, де доступні різноманітні ресурси для навчання та підтримки.

У PSoC Creator можна налаштовувати піни для виходів компонентів через файл Design-Wide Resources (.cydwr). Це дозволяє гнучко налаштовувати підключення компонентів до фізичних пінів пристрою, що є особливо корисним при розробці апаратних рішень. [7]

1.3 Висновок до розділу 1

Архітектура програмованих систем на кристалі (PSoC) від Cypress Semiconductor представляє собою інноваційне поєднання аналогових та цифрових блоків на одному чипі, що дозволяє реалізовувати різноманітні функціональні можливості у компактному та енергоефективному форматі.

У рамках цього розділу було розглянуто загальну структуру та ключові компоненти PSoC, зокрема, програмовані аналогові блоки (такі як АЦП, ЦАП, операційні підсилювачі) та цифрові блоки (наприклад, ШІМ, UART, SPI), які забезпечують високу гнучкість у проектуванні вбудованих систем.

Відмінною особливістю PSoC є його програмована маршрутизація, яка дозволяє вільно направляти сигнали між блоками, оптимізуючи використання апаратних ресурсів та спрощуючи процес налагодження системи. Така гнучкість є важливою перевагою при створенні складних пристроїв, де необхідно інтегрувати різні функціональні елементи.

Розробка на базі PSoC значно спрощується завдяки використанню інструментів PSoC Designer та PSoC Creator. PSoC Designer надає інтуїтивно зрозумілий графічний інтерфейс для додавання та налаштування компонентів, а також вбудований редактор коду для написання та налагодження прошивки. PSoC Creator, в свою чергу, забезпечує розширені можливості для проектування,

інтегруючи функції апаратного та програмного забезпечення, дозволяючи ефективно керувати системними ресурсами та автоматизувати процеси налаштування і тестування.

Загалом, архітектура PSoC і використання PSoC Designer та PSoC Creator відкривають широкі можливості для розробки ефективних, компактних та надійних вбудованих систем, що задовольняють вимоги сучасної електроніки у різних галузях, від побутової техніки до медичних приладів і промислових додатків.

2 Принципи роботи штучних нейронів і нейромереж

2.1 Основні поняття та принципи функціонування штучних нейронних мереж

Штучні нейронні мережі (ШНМ) складаються з трьох основних шарів: вхідного, прихованого та вихідного. Кожен шар складається з нейронів, які з'єднані між собою синаптичними зв'язками. Вхідний шар приймає дані ззовні і передає їх до прихованого шару, де відбувається основна обробка інформації. Принцип роботи полягає у передачі сигналу через нейрони, які активуються за певними правилами, і передають сигнал до наступного шару. [8]

Однією з головних особливостей ШНМ є паралельна обробка інформації. Це дозволяє мережам виконувати складні обчислення значно швидше, ніж послідовні системи. Висока зв'язаність нейронів забезпечує міцні зв'язки між різними частинами мережі, що сприяє більш точній обробці даних. Навчання ШНМ полягає в налаштуванні вагових коефіцієнтів, що дозволяє мережі адаптуватися до нових даних і вдосконалювати свої результати.

Навчання ШНМ ґрунтується на завданнях багатовимірної оптимізації. Використовуються різні алгоритми навчання, такі як градієнтний спуск, зворотне розповсюдження помилки та інші. Ці алгоритми допомагають налаштувати вагові

коефіцієнти нейронів, щоб мінімізувати помилку і досягти найкращих результатів. Процес навчання включає в себе поділ даних на тренувальну, валідаційну та тестову вибірки, що дозволяє оцінити продуктивність моделі. [8]

Біологічні нейронні мережі є аналоговими системами, які мають здатність до самоорганізації та динамічної обробки інформації. Біологічні нейрони взаємодіють через синапси, що дозволяє їм адаптуватися до змін в навколишньому середовищі. Такі мережі демонструють високу надійність і стійкість до пошкоджень, що забезпечується завдяки їхній складній структурі і високому рівню редундантності. [9]

Штучні нейронні мережі, як і біологічні, здатні до самоорганізації та навчання. Проте, вони мають низку обмежень, зокрема, обмежену здатність до узагальнення та адаптації до нових умов. Однак, завдяки цифровій природі, ШНМ мають високу точність обчислень та швидкість роботи, що робить їх незамінними у багатьох прикладних задачах.

Однією з базових моделей штучних нейронів є персептрон. Персептрон використовує зважування вхідних сигналів та функцію активації для прийняття рішень. Зважування сигналів відбувається за допомогою множення кожного вхідного сигналу на відповідний ваговий коефіцієнт, після чого результати підсумовуються і передаються до функції активації, яка визначає вихідний сигнал нейрона.

Функція активації відіграє ключову роль у функціонуванні нейрона. Вона визначає, чи буде нейрон активований і передаватиме сигнал далі. Найбільш поширеними функціями активації є сигмоїдна функція, ReLU (Rectified Linear Unit) та гіперболічний тангенс. Кожна з цих функцій має свої особливості і використовується залежно від задачі, яку вирішує нейронна мережа. [10]

Апаратні нейронні мережі (HNN) є спеціальними пристроями, які реалізують топології штучних нейронних мереж (ANN) та алгоритми навчання, використовуючи вбудовану паралельність нейронної активності. HNN призначені

для енергоефективної обробки з повною паралельністю, що особливо корисно для додатків, таких як потокове стиснення відео.

Існують різні типи нейрочипів, які використовуються для побудови HNN. Деякі з них використовують цифрову обробку, інші - аналогову, а треті - гібридну. Загальні призначені нейрочипи можуть реалізовувати кілька нейронних алгоритмів для одного додатку, тоді як спеціалізовані нейрочипи можуть повторювати один нейронний алгоритм для багатьох додатків. [10]

Нейрочипи мають активаційні блоки, які виконують множення ваги на вхідний сигнал і сумування. Деякі завдання, необхідні для роботи нейронних мереж, можуть виконуватись хост-комп'ютером, наприклад, стан нейронів, ваги і активаційні функції. Ваги можуть зберігатися як в цифровій, так і в аналоговій формі, і завантажуватися статично або динамічно.

Основною перевагою HNN є висока швидкість обробки та енергоефективність. Проте, вони можуть мати обмеження, пов'язані з апаратними обмеженнями, такими як обчислювальні помилки, що можуть перешкоджати навчанню і призводити до неточних результатів. Нелінійні активаційні функції можуть також створювати додаткові складнощі у проектуванні. [9]

Різні типи штучних нейронних мереж:

Стохастичні штучні нейронні мережі (Stochastic Artificial Neural Network) є типом інструменту штучного інтелекту. Вони створюються шляхом введення випадкових варіацій у мережу, або за допомогою надання нейронам стохастичних функцій передачі, або шляхом додавання їм стохастичних ваг. Це робить їх корисними інструментами для оптимізаційних задач, оскільки випадкові коливання допомагають уникнути локальних мінімумів. Стохастичні нейронні мережі, які створюються за допомогою стохастичних функцій передачі, часто називають машинами Больцмана.

Physical Artificial Neural Network: Більшість штучних нейронних мереж сьогодні є програмними, але це не виключає можливість їх створення з фізичних елементів, які базуються на матеріалах з регульованим електричним опором.

Історія фізичних штучних нейронних мереж сягає 1960-х років, коли були створені перші фізичні штучні нейронні мережі з використанням пам'яті транзисторів, званих мемісторами. Мемістори емулюють синапси штучних нейронів. Хоча ці штучні нейронні мережі були комерціалізовані, вони не змогли досягти масштабування. Після цього було зроблено кілька спроб створення фізичної штучної нейронної мережі на основі нанотехнологій або матеріалів зі змінним станом. [9]

Розширені архітектури для біомедичних застосувань. Розроблено розширені архітектури штучних нейронних мереж спеціально для біомедичних застосувань, які пропонують покращену продуктивність та бажані властивості. Серед них варто відзначити мережі для сегментації медичних зображень, прогнозування результатів лікування та аналізу даних про пацієнтів. Наприклад, багатошарова перцептронна нейронна мережа (MLP) та радіально-базисна функціональна нейронна мережа (RBF) широко використовуються для сегментації зображень та інших завдань у галузі медицини.

2.2 Режими навчання

Існують три основні парадигми навчання: навчання з учителем (supervised learning), навчання без учителя (unsupervised learning) та навчання з підкріпленням (reinforcement learning). Кожна з цих парадигм може бути застосована до будь-якої архітектури штучної нейронної мережі. Кожна парадигма навчання має безліч алгоритмів навчання. [11]

Навчання з учителем — це метод машинного навчання, який встановлює параметри штучної нейронної мережі на основі тренувальних даних. Завданням навчальної нейронної мережі є встановлення значення своїх параметрів для будь-якого дійсного вхідного значення після отримання вихідного значення.

Тренувальні дані складаються з пар вхідних та бажаних вихідних значень, традиційно представлених у вигляді векторів даних .

Навчання без учителя — це метод машинного навчання, який встановлює параметри штучної нейронної мережі на основі заданих даних та функції вартості, яку потрібно мінімізувати. Функція вартості може бути будь-якою функцією і визначається формулюванням завдання. Навчання без учителя переважно використовується у додатках, які відносяться до області задач оцінки, таких як статистичне моделювання, компресія, фільтрація, розділення джерел та кластеризація. [11]

Навчання з підкріпленням — це метод машинного навчання, який встановлює параметри штучної нейронної мережі, коли дані зазвичай не задані, а генеруються шляхом взаємодії з середовищем. Навчання з підкріпленням стосується того, як штучна нейронна мережа повинна приймати дії у середовищі, щоб максимізувати деяке поняття довгострокової винагороди. Після визначення функції винагороди, яку потрібно максимізувати, навчання з підкріпленням використовує кілька алгоритмів для знаходження політики, яка приносить максимальну винагороду.

Для кращого розуміння функціоналу ШНМ розглянемо конкретний приклад.

2.3 Задача комівояжера

Задача комівояжера (TSP) полягає у знаходженні найкоротшого маршруту, який дозволяє відвідати кожне місто рівно один раз і повернутися в початкову точку. Це класична оптимізаційна проблема, яка є NP-складною, що означає, що час розв'язання зростає експоненціально з кількістю міст. Гібридні аналогово-цифрові нейронні мережі використовуються для ефективного розв'язання таких

задач завдяки їхній здатності швидко обробляти великі обсяги даних та знаходити наближені оптимальні рішення. [12]

Гібридні аналогово-цифрові архітектури комбінують переваги аналогових і цифрових компонентів для досягнення високої продуктивності та енергоефективності. Аналогові компоненти забезпечують безперервність і гнучкість обробки сигналів, тоді як цифрові компоненти додають точність і стійкість до шуму. Це дозволяє реалізовувати складні обчислювальні завдання, такі як TSP, з високою швидкістю та ефективністю. [12]

Гібридна аналогово-цифрова нейронна мережа може використовуватися для розв'язання задачі комівояжера шляхом моделювання системи зворотного зв'язку, яка дозволяє оптимізувати маршрут. Наприклад, мережа може бути реалізована з використанням фотонних інтегрованих схем, де мікрокільцеві модулятори виконують операції множення, а фотодіоди виконують сумування результатів. Це дозволяє обробляти вхідні сигнали та ваги з високою точністю та швидкістю, знаходячи оптимальні або наближені оптимальні маршрути для задачі комівояжера.

Сучасний штучний інтелект, заснований на алгоритмах глибокого навчання, продемонстрував вражаючі можливості. Однак ці алгоритми вимагають величезної обчислювальної потужності та відповідної енергії. Зараз попит на обчислювальну потужність подвоюється кожні 3–4 місяці, що перевищує відомий закон Мура. Це призвело до появи апаратних прискорювачів домену, які використовують специфічні для додатків інтегральні схеми (ASIC), наприклад, тензорні процесори Google (TPU) і IBM TrueNorth. Метою є розробка ефективної апаратної платформи з розширеним паралелізмом для множення матриць. Однак мікроелектроніка стикається з фундаментальними вузькими місцями у швидкості, споживанні енергії, нагріванні та затримці з’єднання, які стає все важче вирішити за допомогою масштабування. [12]

Інтегрована оптична нейронна мережа (ONN) була запропонована для вирішення перешкод мікроелектроніки [11,12], демонструючи потенціал

перевершити своїх цифрових мікроелектронних аналогів у швидкості обчислень, споживанні енергії, а також щільності обчислень. [14]

Гібридний оптичний процесор принципово відрізняється від існуючих аналогових процесорів ONN і пропонує такі переваги. Введення логічних рівнів дозволяє істотно підвищити точність обчислень при множенні матриць. Потужні алгоритми цифрової обробки сигналів можуть покращити продуктивність обчислень і забезпечити високу повторюваність обчислень. ЦАП високої роздільної здатності для входів можна видалити і вимоги до АЦП для виходів можуть бути значно звільнені, враховуючи матрицю-векторне множення (MVM). Натомість це може збільшити швидкість роботи та покращити сумісність із мікроелектронікою. Дане моделювання показує хорошу шумостійкість і покращену продуктивність НОР порівняно з аналоговою оптичною обчислювальною схемою.

Примітно, що НОР — це метод, який потенційно може бути застосований до інших схем ONN і започаткувати нові концепції, що розглядають предметно-орієнтовані оптичні обчислення.

Принцип роботи гібридного оптичного процесора (рис. 2.1).

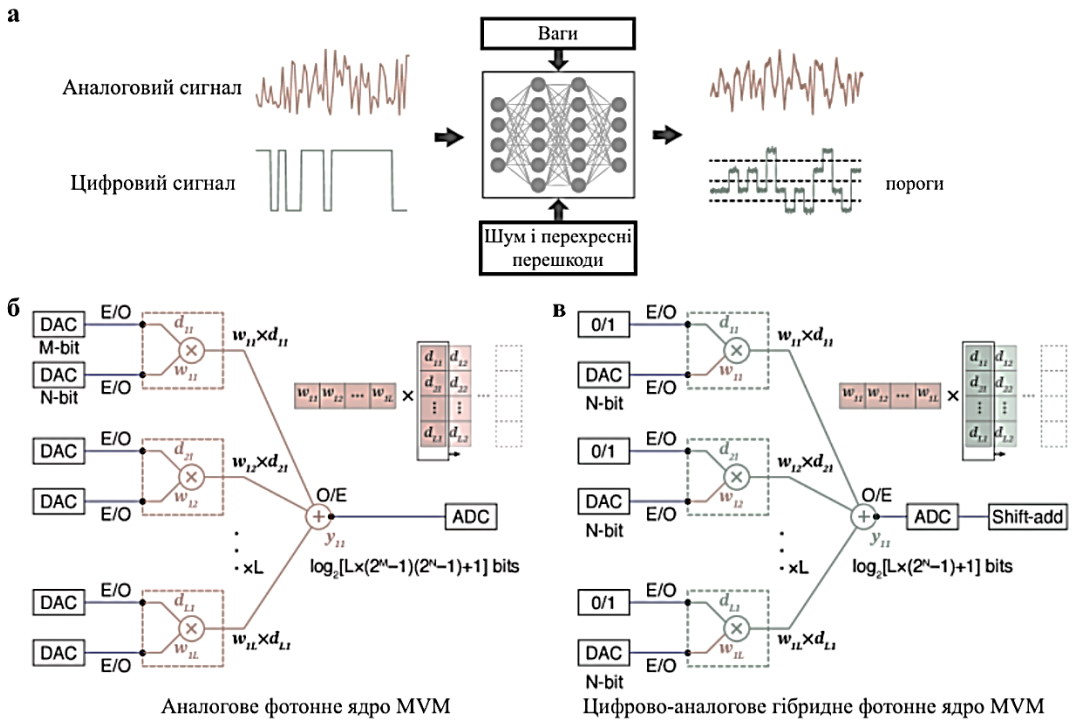

Рисунок 2.1 - Концепція цифрово-аналогового гібридного фотонного ядра MVM [15]

На (рис. 2.1а) - цифровий сигнал більш стійкий до шуму та перехресних перешкод у системі обробки сигналу. ONN можна розглядати як систему обробки сигналів, де цифрові сигнали потенційно можуть бути застосовані для кращої повторюваності обчислень, точності, масштабованості та сумісності з мікроелектронікою порівняно з аналоговими сигналами. На (рис. 2.1б) - абстрактне аналогове фотонне ядро MVM, що використовує аналогові сигнали для обох входів d і вага w. (рис. 2.1в) - запропоноване цифрово-аналогове гібридне фотонне ядро MVM, що використовує цифрові сигнали для введенняd, з послабленими обмеженнями для перетворювачів формату сигналу.

Аналогові сигнали принципово вразливі до шуму та перехресних перешкод. Оптичну обчислювальну систему можна розглядати як систему обробки оптичних сигналів, а ONN є представленням рівнів системи множення оптичної матриці. Як показано на (рис. 2.1а), аналогові сигнали, що проходять через систему обробки сигналів із шумом і перехресними перешкодами, зазнають більш серйозного погіршення, ніж цифрові сигнали з тим самим SNR . Це пов'язано з відсутністю логічних рівнів і процесів прийняття рішень в аналогових системах, що перешкоджає ефективному відновленню та вирівнюванню сигналу. Тут висувається гіпотеза про те, що продуктивність оптичної обчислювальної системи може покращитися завдяки використанню цифрових оптичних сигналів.

Одна з ключових переваг гібридних аналогово-цифрових архітектур полягає в їхній стійкості до шуму. Наприклад, гібридні фотонні мультиплікатори можуть обробляти сигнали з високою точністю навіть в умовах наявності шуму, що забезпечує високу якість рішень для задач оптимізації. Сигнали перетворюються з аналогових у цифрові для подальшої обробки, що дозволяє мінімізувати вплив шуму і підвищити точність обчислень. [14]

2.4 Реалізація нейронної мережі за допомогою аналогових пристроїв

Схеми на основі JFET

Використання польових транзисторів з переходом (JFET) для створення нейронних мереж дозволяє реалізувати ефективні аналогові схеми. JFET транзистори забезпечують високу вхідну опірність і низький рівень шуму, що робить їх ідеальними для аналогових обчислень. У таких схемах вхідні сигнали управляють провідністю каналу, що дозволяє виконувати множення і сумування сигналів для реалізації нейронних функцій. [16]

Операційні підсилювачі широко використовуються в аналогових нейронних мережах для виконання різних математичних операцій. Вони забезпечують високу точність і стабільність обчислень. В аналогових нейронних мережах опампи можуть бути використані для реалізації функцій активації, таких як сигмоїдальні або гіперболічні тангенс-функції, що дозволяє моделювати поведінку біологічних нейронів.

Імпульсні нейронні мережі використовують короткі імпульси для передачі інформації між нейронами. Цей підхід дозволяє зменшити енергоспоживання і збільшити швидкість обробки сигналів. Аналогові імпульсні нейронні мережі можуть бути реалізовані з використанням RC-ланцюгів та інших аналогових компонентів, що забезпечує високу ефективність обробки даних .

Аналогові нейронні мережі можуть бути використані для реалізації різних логічних функцій, таких як AND, OR, та XOR. Це досягається шляхом відповідного налаштування вагових коефіцієнтів і функцій активації. Такі мережі можуть бути використані для побудови складних логічних схем і обробки даних в реальному часі. [16]

2.5 Різновиди функцій активації

Розглянуто різні типи функцій активації, які використовуються в нейронних мережах, включаючи:

- Функція ReLU (Rectified Linear Unit): Використовується для усунення проблеми зникання градієнта в глибоких мережах, забезпечуючи високу швидкість навчання;
- Leaky ReLU: Варіант ReLU, який вирішує проблему мертвих нейронів, пропускаючи невелику кількість негативних значень;
- ELU (Exponential Linear Unit): Забезпечує більш плавний перехід між позитивними і негативними значеннями, що допомагає стабілізувати навчання.

Використання нейронних мереж для обробки зображень включає в себе архітектури, такі як згорткові нейронні мережі (CNN), які спеціалізуються на розпізнаванні і класифікації зображень. CNN використовують згорткові шари для виділення ознак, що дозволяє ефективно обробляти великі обсяги візуальних даних. [16]

Автокодери є потужними інструментами для зменшення розмірності даних і вилучення ознак. Вони складаються з двох основних компонентів: кодера і декодера. Кодер перетворює вхідні дані у зменшене представлення, тоді як декодер відновлює вихідні дані з цього представлення. Автокодери широко використовуються для задач стиснення даних, денойзингу та генерації нових зразків даних.

2.6 Висновок до розділу 2

Штучні нейронні мережі (ШНМ) є потужним інструментом для обробки складних даних, імітуючи роботу біологічних нейронних мереж. Основні компоненти ШНМ включають вхідний шар, приховані шари та вихідний шар, де кожен шар складається з нейронів, з'єднаних між собою синаптичними зв'язками. Ці зв'язки дозволяють передавати сигнали від вхідних даних до вихідного результату, забезпечуючи обробку інформації.

Навчання ШНМ базується на оптимізації вагових коефіцієнтів через різні алгоритми, такі як градієнтний спуск і зворотне розповсюдження помилки. Ці алгоритми дозволяють мережі адаптуватися до нових даних і покращувати свою точність. Процес навчання включає поділ даних на тренувальні, валідаційні та тестові вибірки, що допомагає оцінити продуктивність моделі.

Принципи роботи штучних нейронів включають зважування вхідних сигналів, сумування цих сигналів і застосування функцій активації, таких як ReLU, сигмоїдна функція та гіперболічний тангенс. Функції активації відіграють ключову роль у визначенні вихідного сигналу нейрона.

ШНМ мають значні переваги над біологічними нейронними мережами завдяки високій точності обчислень та швидкості роботи. Однак вони також мають обмеження, зокрема меншою здатністю до узагальнення та адаптації до нових умов. Водночас, біологічні нейронні мережі мають велику гнучкість та здатність до самоорганізації, що дозволяє їм ефективно адаптуватися до змін у навколишньому середовищі.

Апаратні нейронні мережі (HNN) використовують різні типи нейрочипів, що дозволяє досягти високої швидкості обробки та енергоефективності. Ці мережі можуть реалізовувати складні алгоритми навчання та обробки даних, забезпечуючи високу продуктивність у різних прикладних задачах.

Таким чином, штучні нейронні мережі є ефективним інструментом для вирішення складних задач у різних галузях завдяки своїй здатності до паралельної обробки інформації та високій точності обчислень.

3 Проектування моделі нейрона і нейромережі на PSoC

3.1 Теоретичні основи CY8CKIT-042

Об’єктом дослідження була обрана плата CY8CKIT-042 (рис. 3.1). PSoC 4 Pioneer Kit – це розробницький комплект, створений для ознайомлення з можливостями PSoC 4 (Programmable System-on-Chip) від Infineon Technologies (раніше Cypress Semiconductor). [6] Цей комплект забезпечує зручне середовище для розробки та тестування додатків з використанням PSoC 4, який поєднує в собі як аналогові, так і цифрові компоненти на одній мікросхемі.

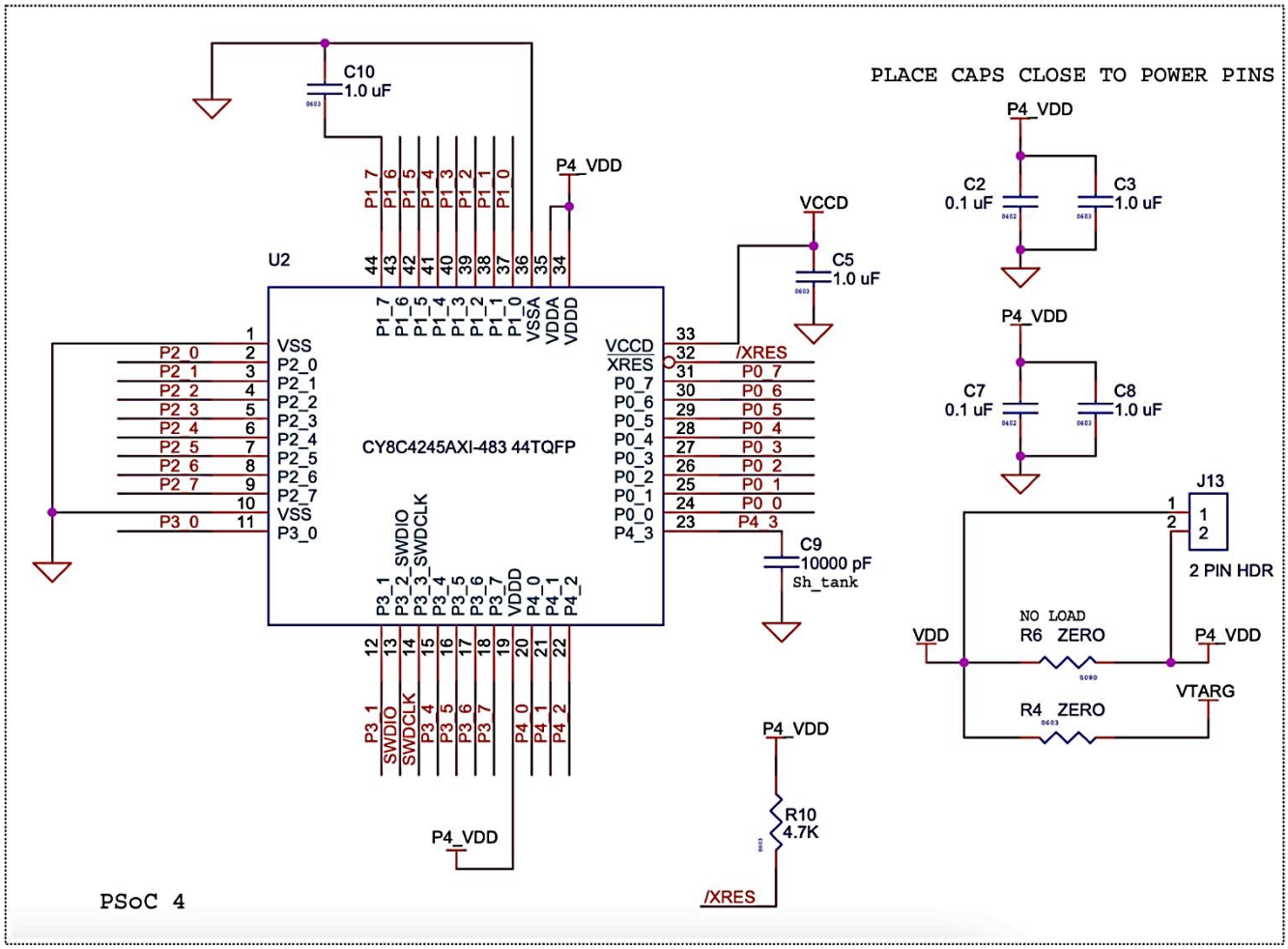

Рисунок 3.1 - Схема CY8CKIT-042 [6]

В платі наявний - мікроконтролер: PSoC 4 (CY8C4245AXI-483), який включає 32-бітний процесор ARM Cortex-M0.

Також, програмовані аналогові блоки: включають в себе 12-бітний SAR ADC, компаратори, operational amplifiers та інші блоки, які дозволяють реалізовувати різні аналогові функції.

Програмовані цифрові блоки: містять UDB (Universal Digital Blocks), які можуть бути налаштовані для реалізації різних цифрових функцій, таких як таймери, лічильники, PWM, UART, SPI та I2C.

Пам'ять: 32 KB Flash, 4 KB SRAM.

Інтерфейси введення/виведення: 36 GPIO (General Purpose Input/Output) ліній, які можуть бути налаштовані як аналогові, цифрові або PWM виходи.

Живлення: може живитись від USB або зовнішнього джерела живлення (3.3V-5V).

USB-UART Bridge: для зручної серійної комунікації з комп'ютером. [6]

Можливості платформи:

- Завдяки програмованій маршрутизації сигналів плата дозволяє гнучко налаштовувати взаємодію між аналоговими та цифровими компонентами;
- Аналогова обробка сигналів - з можливістю використання програмованих аналогових блоків для реалізації таких функцій як фільтрація, посилення та аналого-цифрове перетворення;
- Цифрова обробка сигналів, завдяки UDB можна реалізовувати складні цифрові алгоритми без необхідності використання зовнішніх компонентів;
- Інтеграція з PSoC Creator. Плата підтримує середовище розробки PSoC Creator, яке надає інтуїтивний графічний інтерфейс для розробки та налаштування додатків;
- Можливість підключення додаткових модулів через Arduino-сумісні заголовки для розширення функціональності плати.

Плата CY8CKIT-042 була обрана завдяки можливостям поєднання аналогових і цифрових функцій на одній платформі, а також через наявність

багатьох корисних компонентів. Це дозволяє ефективно реалізовувати нейронні мережі та інші складні алгоритми обробки сигналів без використання зовнішніх компонентів.

Використовуючи аналогові та цифрові блоки, ми можемо створити просту модель нейрона, яка буде здатна до самонавчання. На (рис. 3.2) наведено схему штучного нейрона, створеного в PSoc Creator із компонентів PSoC, котрі апаратно реалізують функції нейрона. Вхідні блоки (обведені блакитним контуром 1) виконують функцію утримання вхідних сигналів (від Pin_1 - 8) за допомогою компонента Sample and Hold та перемноження їх на вагові коефіцієнти за допомогою програмованих підсилювачів PGA, коефіцієнти підсилення яких відповідають ваговим коефіцієнтам відповідних входів штучного нейрона. В блоці2, обведеного жовтим, виконується функція підсумовування сигналів після за допомогою операційного підсилювача Opamp_1, на якому також реалізована і нелінійна функція активації за рахунок нелінійного зворотного зв’язку в операційному підсилювачі через нелінійний елемент D_1.

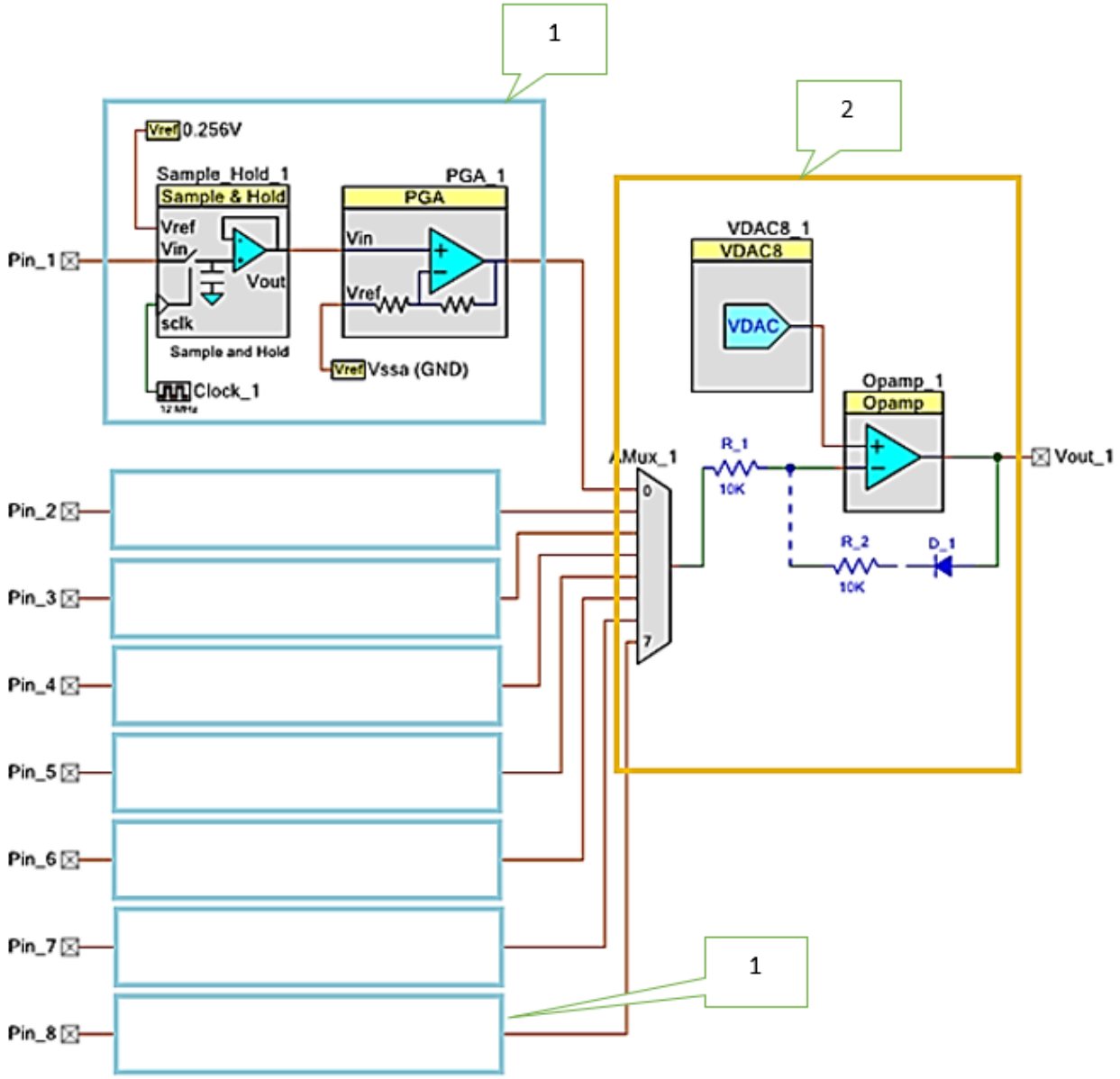

Рисунок 3.2 – Модель штучного нейрона на компонентах PSoC, створена в PSoC Creator

За допомогою ADC для зчитування даних з виходів нейронів та PWM для керування виконавчими механізмами, ми можемо створити замкнуту систему управління. Також дана робота дає змогу вивчити можливості PSoC 4 шляхом дослідження архітектури та можливостей PSoC 4 для подальшого використання в більш складних проектах.

3.2.1 Опис архітектури PSoC

Для реалізації проекту керування світлодіодом за допомогою простої нейронної мережі на платі CY8CKIT-042, ми повинні з'єднати компоненти, які взаємодіють між собою. У цьому випадку основні компоненти (крім наведених на Рис. 3.2) включають UART, ADC, PWM та світлодіод. Основні компоненти та їх підключення:

- UART (Universal Asynchronous Receiver/Transmitter): універсальний асинхронний приймач/передавач використовується для послідовного зв’язку, що дозволяє PSoC спілкуватися з іншими пристроями, такими як ПК, для налагодження, реєстрації або взаємодії з іншими послідовними пристроями. У нашому прикладі нейронної мережі UART можна використовувати для надсилання та отримання даних, забезпечуючи спосіб моніторингу продуктивності мережі та коригування параметрів за потреби. TX (передавач) UART підключено до RX (приймач) UART на комп'ютері. RX UART підключено до TX UART на комп'ютері (рис. 3.3);
- ADC (Analog-to-Digital Converter): Аналого-цифровий перетворювач є ключовим компонентом для перетворення аналогових сигналів від датчиків у цифрові значення, які може обробляти мікроконтролер. У нашому прикладі простої нейронної мережі АЦП можна використовувати для зчитування вхідних значень із датчиків, які потім можна використовувати як вхідні дані для нейронної мережі для навчання та висновків. Вхідний сигнал ADC (припустимо, що це потенціометр) підключений до відповідного піну на платі PSoC (рис. 3.4);
- PWM (широтно-імпульсна модуляція): PWM-компонент (рис. 3.5) можна використовувати для керування такими пристроями, як світлодіоди, двигуни та інші приводи. У нашому прикладі зі світлодіодом ШІМ-сигнал можна регулювати на основі вихідних даних нейронної мережі, демонструючи, як

рішення мережі можуть безпосередньо впливати на дії обладнання. Вихід PWM підключений до анода світлодіода через резистор. Катод світлодіода підключений до землі (GND);

– Світлодіод, підключений до виходу PWM.

Схеми з'єднань для проекту нейронної мережі на PSoC CY8CKIT-042:

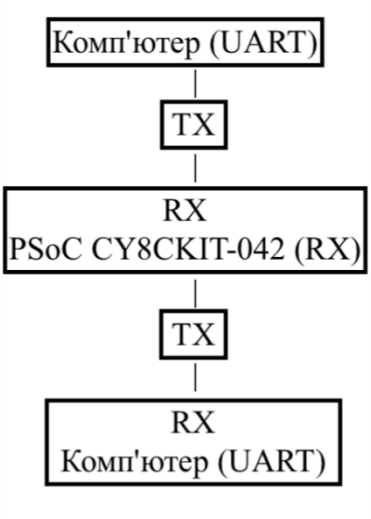

Рисунок 3.3 - Схема з'єднань комп'ютер до PSoC

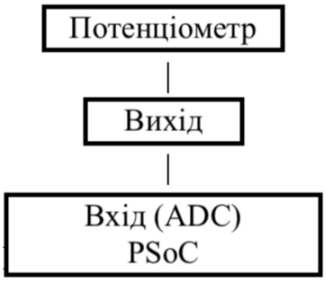

Рисунок 3.4 - Схема з'єднань потенціометр до PSoC

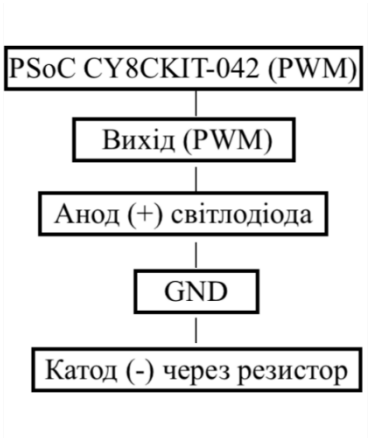

Рисунок 3.5 - Схема з'єднань PSoC до Світлодіода

Ця схема забезпечує всі необхідні з'єднання між компонентами для реалізації простої нейронної мережі на платі CY8CKIT-042. Використання UART дозволяє передавати дані між комп'ютером та платою для зворотного зв'язку, ADC забезпечує зчитування вхідного сигналу, а PWM дозволяє керувати яскравістю світлодіода на основі вихідного сигналу нейронної мережі.

3.3 Реалізація

3.3.1 Програмування у PSoC Creator

Опис завдання - реалізація одновимірної нейронної мережі на платі CY8CKIT-042 для класифікації двох класів. Вхідні дані будуть подаватися через аналогові входи (Pon_1-7, рис.3.2) на штучний нейрон, з виходу якого далі на вхід ADC. Плата CY8CKIT-042 буде використовувати фоторезистори для зчитування рівня освітленості та керувати яскравістю світлодіода за допомогою широтно-імпульсної модуляції (PWM).

Почнемо з простого проекту, де ми будемо використовувати нейронну мережу для керування яскравістю світлодіода на основі вхідних сигналів. Проект включає читання вхідних сигналів, обчислення вихідного сигналу нейрону та керування світлодіодом на основі цього вихідного сигналу. Ми також реалізуємо

функцію навчання для коригування ваг нейрону (коефіцієнтів підсилення PGA) на основі зворотного зв'язку.

Налаштування проекту у PSoC Creator: після створення проекту, додавання і налаштовування компонентів (із рис.3.2 та ADC, UART, PWM).

Створення одновимірної нейронної мережі з одним нейроном на вхідному шарі та одним нейроном на вихідному шарі. Почнемо написання коду з ініціалізації необхідних компонентів: UART для комунікації та PWM для керування світлодіодом:

- Sample_Hold_1_Start(): Запускає модулі утримувачів сигналу (для кожного входу (1-4);
- PGA_1_Start(): Запускає модулі програмованих підсилювачів (для кожного входу (1-4);
- Opamp_1__Start(): Запускає модуль операційного підсилювача;
- UART_Start(): Запускає модуль UART;
- PWM_Start(): Запускає модуль PWM.

Ініціалізуємо ваговий коефіцієнт нейрону:

- weights[0] = 0.5: Встановлюємо початкове значення ваги для нейрону.

Використовуємо функцію активації Sigmoid для обчислення вихідного сигналу нейрону та її похідну для навчання:

- Sigmoid(x): Обчислює значення функції Sigmoid для вхідного значення x;
- SigmoidDerivative(x): Обчислює похідну функції Sigmoid для значення x.

Реалізуємо функції для обчислення вихідного сигналу нейрону та навчання нейронної мережі. Зчитуємо вхідний сигнал з ADC і нормалізуємо його до діапазону [0, 1]:

- ReadInput(): Зчитує вхідний сигнал з ADC (аналогово-цифрового перетворювача);
- NormalizeInput(rawInput): Нормалізує зчитаний вхідний сигнал до діапазону [0, 1].

У головному циклі програми ми обробляємо вхідний сигнал, обчислюємо вихідний сигнал нейрону, виводимо дані через UART, навчаємо нейронну мережу і керуємо світлодіодом:

- CyGlobalIntEnable: Включає глобальні переривання;
- InitializeComponents(): Ініціалізує компоненти UART та PWM;
- InitializeNetwork(): Ініціалізує ваги нейрону;
- ReadInput() та NormalizeInput(rawInput): Зчитують та нормалізують вхідний сигнал;
- FeedForward(input): Обчислює вихідний сигнал нейрону;
- Train(input, feedback): Навчає нейронну мережу на основі зворотного зв'язку;
- PWM_WriteCompare((uint16)(output * 255)): Керує яскравістю світлодіода на основі вихідного сигналу нейрону.

Цей код реалізує просту нейронну мережу, яка навчається керувати яскравістю світлодіода на основі вхідного сигналу та зворотного зв'язку.

Компілюємо проект та завантажуємо код на плату. Для цього потрібно підключити CY8CKIT-042 до комп'ютера через USB. Далі варто натиснути "Debug" -> "Program" для завантаження коду на плату. Переконуємося, що плата успішно завантажена та почала виконувати програму.

Файл KitProg.hex доступний у такому місці: \CY8CKIT-042 PSoC 4 Pioneer Kit\\Firmware\Programmer\KitProg.

3.3.2 Основні алгоритми

Нейронна мережа навчається за допомогою алгоритму зворотного поширення помилки (backpropagation), який коригує вагові коефіцієнти на основі помилки між очікуваним та фактичним виходом. Ось детальніше, як це відбувається:

1. Обчислення вихідного сигналу (FeedForward).

Нейронна мережа отримує вхідний сигнал, перетворює його на вихідний сигнал за допомогою вагових коефіцієнтів і функції активації (в даному випадку функція активації – Sigmoid).

2. Обчислення помилки.

Після обчислення вихідного сигналу обчислюється різниця між фактичним вихідним сигналом (output) та очікуваним (target). Ця різниця є помилкою.

3. Коригування вагових коефіцієнтів (Backpropagation)

Вагові коефіцієнти коригуються на основі помилки, градієнту функції активації (SigmoidDerivative) та вхідного сигналу. Цей процес повторюється для кожного навчального прикладу.

Ваговий коефіцієнт коригується на основі помилки, похідної функції активації та вхідного сигналу:

- weights[0] += learning_rate * error * SigmoidDerivative(output) * input;

Цей процес повторюється для кожного навчального прикладу, що дозволяє мережі поступово зменшувати помилку і покращувати точність класифікації.

Основний цикл програми включає обчислення вихідного сигналу, виведення даних через UART та навчання нейронної мережі на основі зворотного зв'язку

Цей код забезпечує навчання нейронної мережі на платі CY8CKIT-042 в реальному часі на основі вхідних даних та зворотного зв'язку. Код для реалізації простої нейронної мережі знаходиться в додатку А.

Спробуємо зробити трохи складнішу мережу з двома входами та вагами. Аналонічно до першої, але внесемо деякі зміни: Додано другий вхід та відповідний ваговий коефіцієнт. Додано функцію зчитування другого каналу ADC.

Обчислення вихідного сигналу та навчання - функції FeedForward та Train тепер враховують два входи та два вагові коефіцієнти. Основний цикл програми зчитує два вхідних сигнали, нормалізує їх, обчислює вихідний сигнал та навчає

нейронну мережу на основі зворотного зв'язку. Код цієї НМ можна знайти в додатку Б.

Завдяки інтеграції аналогових і цифрових компонентів на одній платформі, розробники можуть швидко та ефективно створювати складні системи без потреби у великій кількості зовнішніх компонентів. Також до переваг PSoC можна додати, що вона дозволяє налаштовувати маршрутизацію сигналів і конфігурацію блоків (UDB) відповідно до потреб проекту, що робить систему дуже адаптивною до різних вимог. Вбудовані 12-бітні SAR ADC, компаратори та operational amplifiers дозволяють обробляти аналогові сигнали з високою точністю та без необхідності у додаткових мікросхемах. Середовище розробки PSoC Creator надає інтуїтивний графічний інтерфейс для налаштування та розробки додатків, що значно спрощує процес програмування. Підтримка Arduino-сумісних заголовків дозволяє легко підключати додаткові модулі для розширення функціональності плати.

Щодо недоліків використання PSoC, процесор ARM Cortex-M0, використовуваний у PSoC 4, має обмежену обчислювальну потужність, що може бути недостатньо для дуже складних обчислювальних задач або великих нейронних мереж. Обмежені ресурси пам'яті 32 KB Flash і 4 KB SRAM можуть бути недостатніми для зберігання великих обсягів даних або складних алгоритмів. Хоча PSoC має вбудовані аналогові блоки, їх кількість та функціональність можуть бути обмеженими в порівнянні з окремими аналоговими мікросхемами. Налаштування та програмування PSoC може бути складнішим для новачків, особливо без попереднього досвіду роботи з системами на кристалі (SoC).

3.4 Висновок до розділу 3

Розробка моделі нейрона та нейронної мережі на платформі PSoC за допомогою CY8CKIT-042 дозволяє продемонструвати можливості програмованих систем на кристалі у поєднанні аналогових та цифрових компонентів.

CY8CKIT-042 PSoC 4 Pioneer Kit був обраний завдяки своїй гнучкості та здатності об’єднувати аналогові та цифрові функції на одній мікросхемі. Вбудовані аналогові блоки (12-бітний SAR ADC, компаратори, operational amplifiers) і цифрові блоки (UDB для реалізації таймерів, лічильників, PWM, UART, SPI та I2C) дозволяють створювати багатофункціональні системи.

Використовуючи програмовані аналогові та цифрові блоки, була реалізована проста модель нейрона, яка здатна до самонавчання. Плата CY8CKIT-042 дозволяє інтегрувати сенсори та виконавчі механізми, створюючи замкнуту систему управління.

Використання PSoC дозволило створити просту модель нейрона, яка навчалася за допомогою алгоритму зворотного поширення помилки (backpropagation). Реалізація цієї моделі включала зчитування даних з ADC, передачу даних через UART та керування виконавчими механізмами за допомогою PWM.

У процесі реалізації була створена нейронна мережа, яка навчалась за допомогою алгоритму зворотного поширення помилки. Ця нейронна мережа керувала яскравістю світлодіода на основі вхідного сигналу та зворотного зв'язку. Всі компоненти були налаштовані та інтегровані у єдину систему, що дозволило продемонструвати ефективність та можливості платформи PSoC для реалізації нейронних мереж.

Підключення фоторезистора для зчитування рівня освітленості та керування яскравістю світлодіода за допомогою широтно-імпульсної модуляції (PWM) показало можливість створення замкнутої системи управління на базі PSoC.

ВИСНОВКИ

У ході виконання дипломної роботи було досліджено та реалізовано модель нейрона і нейромережі на базі програмованої системи на кристалі з використанням плати CY8CKIT-042. Робота підтвердила можливості цієї платформи для ефективного апаратного моделювання нейронних мереж, що поєднує в собі аналогові та цифрові компоненти.

Проведено детальний аналіз архітектури PSoC, включаючи можливості програмованих аналогових та цифрових блоків. Було виявлено, що PSoC дозволяє ефективно поєднувати різні компоненти для створення складних вбудованих систем.

Досліджено основні принципи функціонування штучних нейронів та нейронних мереж. Розглянуто різні типи нейронних мереж та їхні архітектури, а також алгоритми навчання, зокрема зворотне поширення помилки.

Розроблено та реалізовано модель нейрона на базі PSoC. Виконано налаштування компонентів у середовищі PSoC Creator, написано та скомпільовано програмне забезпечення для реалізації нейронної мережі.

На практиці створено нейронну мережу, яка має контролювати яскравість світлодіода на основі даних від сенсора. Це демонструє можливості платформи PSoC для реалізації машинного навчання у вбудованих системах.

Платформа PSoC CY8CKIT-042 є потужним інструментом для створення апаратних моделей нейронних мереж, поєднуючи в собі програмовані аналогові та цифрові блоки.

Використання PSoC значно спрощує процес розробки вбудованих систем, знижуючи витрати та підвищуючи ефективність реалізації складних алгоритмів обробки сигналів.

Результати роботи підтверджують доцільність подальших досліджень у галузі застосування PSoC для реалізації нейронних мереж та їх інтеграції у різні технологічні системи.

Результати даної роботи можуть бути використані для розробки нових вбудованих систем з елементами штучного інтелекту, що знайдуть застосування у різних галузях, включаючи медицину, робототехніку та автоматизацію.

У подальших дослідженнях рекомендується зосередитись на оптимізації використання ресурсів пам'яті та процесора, а також на розширенні функціональних можливостей нейронних мереж, реалізованих на базі PSoC.

СПИСОК ЛІТЕРАТУРИ

1. Infineon Technologies AG. Embedded Power ICs (System-on-Chip) [Електронний ресурс]. 1999 - 2024. URL: https://www.infineon.com/cms/en/product/microcontroller/embedded-power-ics-system-on-chip-/ (дата звернення: 06.06.2024).
2. Javaid H., Parameswaran S. Pipelined Multiprocessor System-on-Chip for Multimedia. Springer, 2014. 239 p. ISBN 978-3-319-01113-4.
3. PSoC® Programmable System-on-Chip Technical Reference Manual (TRM). PSoC TRM, Document No. 001-14463 Rev. *D. 2005 - 2009. Copyright © 2005 - 2009 Cypress Semiconductor Corporation. All rights reserved. 568 с.
4. Килочек Д. Проектування на програмованих системах на кристалі PSoC Cypress. 2006. 116 с.
5. Infineon. PSoC4 MCU Application Notes [Електронний ресурс]. AN79953, Document No. 001-79953 Rev. *Z. 2024-02-26. URL: www.infineon.com (дата звернення: 06.06.2024).
6. Cypress Semiconductor. PSoC 4 Pioneer Kit Guide. Doc. # 001-86371 Rev. *K. 2013–2019. 129 с. URL: www.cypress.com (дата звернення: 06.06.2024).
7. Bjerke O. Development of a Low-Cost Potentiostat with Cyclic Voltammetry and Amperometry Techniques Implemented: A Prototype Platform for Medical

Applications using a Programmable System on Chip (PSoC). Thesis submitted for the degree of Master in Physics, 60 credits. Department of Physics, University of Oslo, Autumn 2020.

8. Тимчук С. О., Панов А. О. Методичні вказівки до виконання практичних робіт з дисципліни «Нейросистеми та нейромережі». Харків: Державний біотехнологічний університет, Інститут «Кіберпорт», Кафедра автоматизації та комп'ютерно-інтегрованих технологій, 2023. 24 с.
9. Dias F. M., Antunes A., Mota A. M. Artificial Neural Networks: a Review of Commercial Hardware // Escola Superior de Tecnologia de Setúbal, Departamento de Engenharia Electrotécnica. 2003. 9 с.
10. Abdulhameed R. A., Ibrahim A. A. A Survey on Hardware Neural Network (HNN) // Altinbas University, Istanbul, Turkey. Volume 15, February 2023. ISSN: 2795-7640. 12 с.
11. Douglas R. J., Mahowald M. A., Martin K. A. C. Hybrid Analog-Digital Architectures For Neuromorphic Systems // [Наукова робота]. 1994. С. 1848–1853.
12. Rupp K. et al. 42 years of microprocessor trend data [Електронний ресурс]. 2018. URL: https://www.karlrupp.net/2018/02/42-years-of-microprocessor-trend-data/ (дата звернення: 06.06.2024).
13. Khan H. N., Hounshell D. A., Fuchs E. R. Science and research policy at the end of Moore's Law // Nature Electronics. 2018. Vol. 1. С. 14–21.
14. Nahmias M. A., et al. Photonic multiply-accumulate operations for neural networks // IEEE Journal of Selected Topics in Quantum Electronics. 2019. Vol. 26. С. 1–18.
15. Meng X., Kong D., Kim K., Li Q., Dong P., Cox I. J., Lioma C., Hu H. Digital-analog hybrid matrix multiplication processor for optical neural networks [Електронний ресурс]. 2024. URL: https://ar5iv.labs.arxiv.org/html/2401.15061 (дата звернення: 06.06.2024).

16. Hawas M. Neural Network realized by using analog devices: ANN based on learning rule of neuron activation function using electronic devices // International Journal of Advance Computer Technology. 2016. VOLUME 4, NUMBER 3. C. 19–22.

ДОДАТОК А

Код для реалізації простої нейронної мережі, що навчається на платі CY8CKIT-042:

```
#include "project.h"
#include <stdio.h>

#define learning_rate 0.1

float weights[1];

// Функція для ініціалізації компонентів
void InitializeComponents(void)
{
   UART_Start();
   PWM_Start();
}

// Функція для ініціалізації нейронної мережі
void InitializeNetwork(void)
{
   weights[0] = 0.5; // Ініціалізація вагового коефіцієнта
}

// Функція активації Sigmoid
float Sigmoid(float x)
{
   return 1.0 / (1.0 + exp(-x));
```

```
}

// Похідна функції активації Sigmoid
float SigmoidDerivative(float x)
{
    return x * (1.0 - x);
}

// Функція для обчислення вихідного сигналу нейрону
float FeedForward(float input)
{
    return Sigmoid(input * weights[0]);
}

// Функція для навчання нейронної мережі
void Train(float input, float target)
{
    float output = FeedForward(input);
    float error = target - output;
    weights[0] += learning_rate * error * SigmoidDerivative(output) * input;
}

// Функція для зчитування вхідного сигналу
int16 ReadInput(void)
{
    return ADC_GetResult16(0); // Припустимо, що ми використовуємо ADC
для зчитування сигналу
}
```

```
// Функція для нормалізації вхідного сигналу
float NormalizeInput(int16 rawInput)
{
    return (float)rawInput / 2048.0; // Нормалізація до діапазону [0, 1]
}

int main(void)
{
    CyGlobalIntEnable; /* Включити глобальні переривання. */

    InitializeComponents();
    InitializeNetwork();

    for(;;)
    {
        int16 rawInput = ReadInput();
        float input = NormalizeInput(rawInput);
        float output = FeedForward(input);

        /* Виведення даних через UART */
        char buffer[50];
        sprintf(buffer, "Input: %d, Output: %.2f\r\n", rawInput, output);
        UART_UartPutString(buffer);

        /* Читання зворотного зв'язку через UART */
        if(UART_SpiUartGetRxBufferSize() > 0)
        {
            uint8 feedback = UART_UartGetChar() - '0'; // Припустимо, що
зворотний зв'язок передається як '0' або '1'
```

```
            Train(input, (float)feedback);
        }

        /* Керування світлодіодом (опціонально) */
        PWM_WriteCompare((uint16)(output * 255));
    }
}
```

ДОДАТОК Б

```
#include "project.h"
#include <stdio.h>

#define learning_rate 0.1

float weights[2]; // Два вагових коефіцієнта

// Функція для ініціалізації компонентів
void InitializeComponents(void)
{
    UART_Start();
    PWM_Start();
}

// Функція для ініціалізації нейронної мережі
void InitializeNetwork(void)
{
    weights[0] = 0.5; // Ініціалізація першого вагового коефіцієнта
    weights[1] = 0.5; // Ініціалізація другого вагового коефіцієнта
}

// Функція активації Sigmoid
float Sigmoid(float x)
{
    return 1.0 / (1.0 + exp(-x));
}
```

```
// Похідна функції активації Sigmoid
float SigmoidDerivative(float x)
{
    return x * (1.0 - x);
}

// Функція для обчислення вихідного сигналу нейрону
float FeedForward(float input1, float input2)
{
    return Sigmoid(input1 * weights[0] + input2 * weights[1]);
}

// Функція для навчання нейронної мережі
void Train(float input1, float input2, float target)
{
    float output = FeedForward(input1, input2);
    float error = target - output;
    weights[0] += learning_rate * error * SigmoidDerivative(output) * input1;
    weights[1] += learning_rate * error * SigmoidDerivative(output) * input2;
}

// Функція для зчитування вхідного сигналу
int16 ReadInput1(void)
{
    return ADC_GetResult16(0); // Припустимо, що ми використовуємо ADC
для зчитування сигналу
}

int16 ReadInput2(void)
{
```

```
    return ADC_GetResult16(1); // Припустимо, що ми використовуємо другий канал ADC для зчитування сигналу
}

// Функція для нормалізації вхідного сигналу
float NormalizeInput(int16 rawInput)
{
    return (float)rawInput / 2048.0; // Нормалізація до діапазону [0, 1]
}

int main(void)
{
    CyGlobalIntEnable; /* Включити глобальні переривання. */

    InitializeComponents();
    InitializeNetwork();

    for(;;)
    {
        int16 rawInput1 = ReadInput1();
        int16 rawInput2 = ReadInput2();
        float input1 = NormalizeInput(rawInput1);
        float input2 = NormalizeInput(rawInput2);
        float output = FeedForward(input1, input2);

        /* Виведення даних через UART */
        char buffer[50];
        sprintf(buffer, "Input1: %d, Input2: %d, Output: %.2f\r\n", rawInput1, rawInput2, output);
        UART_UartPutString(buffer);
```

```
            /* Читання зворотного зв'язку через UART */
            if(UART_SpiUartGetRxBufferSize() > 0)
            {
                uint8 feedback = UART_UartGetChar() - '0'; // Припустимо, що
зворотний зв'язок передається як '0' або '1'
                Train(input1, input2, (float)feedback);
            }

            /* Керування світлодіодом (опціонально) */
            PWM_WriteCompare((uint16)(output * 255));
        }
    }
```