

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет прикладних інформаційних технологій та електроінженерії  
(повна назва факультету)

Кафедра автоматизації технологічних процесів та виробництв  
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Розробка та дослідження автоматизованої системи керування підйомним механізмом

Виконав: студент

VI курсу, групи КАМ-61

спеціальності

174 Автоматизація та комп'ютерно-інтегровані технології

(шифр і назва спеціальності)

Чомко Т.В.

(підпис)

(прізвище та ініціали)

Керівник

Дмитрів О.Р.

(підпис)

(прізвище та ініціали)

Нормоконтроль

Козбур І.Р.

(підпис)

(прізвище та ініціали)

Завідувач кафедри

Савків В.Б.

(підпис)

(прізвище та ініціали)

Рецензент

Золотий Р.З.

(підпис)

(прізвище та ініціали)

Тернопіль  
2023

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Факультет Факультет прикладних інформаційних технологій та електроінженерії  
(повна назва факультету)

Кафедра комп'ютерно-інтегрованих технологій  
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Микитишин А.Г.  
(підпис) (прізвище та ініціали)

«\_\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ  
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Магістр  
(назва освітнього ступеня)

за спеціальністю 174 Автоматизація та комп'ютерно-інтегровані технології  
(шифр і назва спеціальності)

Студенту Чомку Тарасу Володимировичу  
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка та дослідження автоматизованої системи керування підйомним механізмом

Керівник роботи Дмитрів Олена Романівна, к.т.н., доцент кафедри КА  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «\_\_\_» \_\_\_\_\_ 2023 року № \_\_\_\_\_

2. Термін подання студентом завершеної роботи 26 грудня 2023р.

3. Вихідні дані до роботи Наукові публікації про підйомно транспортні операції

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ

Аналітична частина

Технологічна частина

Конструкторська частина

Науково-дослідна

Спеціальна частина

Безпека в надзвичайних ситуаціях, охорона праці, Висновки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

## 6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	Тотосько О.В., доцент кафедри КТ		
Безпека в надзвичайних ситуаціях	Клепчик В. М., проректор з адміністративно-господарської роботи та будівництва		

7. Дата видачі завдання \_\_\_\_\_

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1.	Ознайомлення з завданням до кваліфікаційної роботи	25.11.2023	Виконано
2.	Підбір наукових джерел по темі роботи	26.11.2023-28.11.2023	Виконано
3.	Опрацювання наукових публікацій та збір даних по темі роботи	29.11.2023-1.12.2023	Виконано
4.	Виконання дослідження згідно мети кваліфікаційної роботи	2.12.2023-4.12.2023	Виконано
5.	Оформлення першого та другого розділів	5.12.2023-7.12.2023	Виконано
6.	Оформлення третього розділу	8.12.2023-10.12.2023	Виконано
7.	Оформлення четвертого розділу	11.12.2023-13.12.2023	Виконано
8.	Виконання завдання до підрозділу «Охорона праці»	14.12.2023-15.12.2023	Виконано
9.	Виконання завдання до підрозділу «Безпека в надзвичайних ситуаціях»	16.12.2023-17.12.2023	Виконано
10.	Оформлення кваліфікаційної роботи	18.12.2023-19.12.2023	Виконано
11.	Нормоконтроль	19.12.2023-20.12.2023	Виконано
12.	Перевірка на плагіат	21.12.2023	Виконано
13.	Попередній захист кваліфікаційної роботи	22.12.2023	Виконано
14.	Захист кваліфікаційної роботи		

Студент

\_\_\_\_\_ (підпис)

Чомко Т.В.

\_\_\_\_\_ (прізвище та ініціали)

Керівник роботи

\_\_\_\_\_ (підпис)

Дмитрів О.Р.

\_\_\_\_\_ (прізвище та ініціали)

## АНОТАЦІЯ

Кваліфікаційна робота магістра складається з пояснювальної записки та графічної частини (ілюстративний матеріал – слайди).

Об'єм графічної частини дипломної роботи становить \_\_\_\_\_.

Об'єм пояснювальної записки складає 65 друкованих сторінок формату А4.

В роботі використано 24 літературних джерел.

У роботі було розроблено автоматизовану систему керування

У роботі було розроблено автоматизовану систему керування групою ліфтів для забезпечення мінімального часу очікування та забезпечення найшвидшої роботи системи

Було використано реалізацію контролера нечіткої логіки для системи управління групою ліфтів з використанням VHDL з двома різними нечіткими алгоритмами. Реалізація контролера нечіткої логіки є дуже простою шляхом кодування кожного компонента системи нечіткого логічного висновку відповідно до специфікацій проекту. Завдяки зменшеним правилам, час обчислення нечіткого контролера зменшується, а система керування групою ліфтів забезпечує більш швидку роботу за рахунок зменшення середнього часу очікування пасажирів до 30,428 секунди порівняно з реальним алгоритмом.

Було реалізовано шість типів алгоритмів диспетчеризації для ліфтової системи. Система керування ліфтом використовує цей алгоритм у різних ситуаціях для безперебійної роботи та зменшує середній час очікування пасажирів та споживання електроенергії ліфтовою системою. Поведінка ліфтової системи покращується шляхом вибору одного алгоритму з п'яти на основі можливої ситуації.

Ключові слова: КОНТРОЛЕР, ЛІФТ, НЕЧІТКА ЛОГІКА, МОДЕЛЮВАННЯ, АВТОМАТИЧНИЙ КОНТРОЛЬ.

## ЗМІСТ

ВСТУП.....	5
1. АНАЛІТИЧНА ЧАСТИНА .....	6
1.1. Огляд системи функціонування ліфта .....	6
1.2. Огляд контролера нечіткої логіки для інтелектуального керування груповою системою ліфта .....	9
1.3. Нечітка логіка для вбудованих систем.....	11
2 ТЕХНОЛОГІЧНА ЧАСТИНА .....	13
2.1. Використання контролера нечіткої логіки для групового керування ліфтовими системами.....	13
2.2. Опис архітектури FPGA.....	15
2.3. Методологія керування нечіткою логікою для групової системи ліфта ..	20
3 КОНСТРУКТОРСЬКА ЧАСТИНА .....	27
3.1. Розроблена методика для керування ліфтовою групою.....	27
3.2. Стратегія диспетчеризації ліфтів.....	29
3.3. Впровадження системи керування ліфтом за допомогою hcs-12 (mc9s12dp256b) .....	30
3.4. Проектування апаратного забезпечення системи .....	34
4. НАУКОВО-ДОСЛІДНА ЧАСТИНА.....	37
4.1. Результат моделювання та реалізація frga .....	37
4.2. Реалізація FPGA .....	40
4.3. Середній час очікування.....	41
5. СПЕЦІАЛЬНА ЧАСТИНА.....	45

5.1. Реалізація алгоритму для системи керування ліфтом .....	45
6 БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ, ОХОРОНА ПРАЦІ .....	53
6.1. Організація охорони праці при роботі з системою управління.....	53
6.2. Електробезпека .....	55
6.3. Розрахунок заземлення .....	58
ОСНОВНІ ВИСНОВКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ .....	62
ПЕРЕЛІК ПОСИЛАНЬ .....	63

## ВСТУП

Внаслідок швидкого зростання населення в містах і багатоповерхових будинках з'явилася потреба в ліфтах збільшується. Зі зростанням рівня життя та увагою до людини та технологій ліфтові системи стають кращими, швидкішими, міцнішими та якіснішими. Раніше більшість систем були орієнтовані на монтаж ліфтів. У 1980-х роках потреба в технічному обслуговуванні ліфтів і персоналі з ремонту почала зростати. Наприклад, стали використовувати автоматичні двері замість захисних, мікропроцесорні електронні карти використовуються замість карт з реле, двошвидкісними або регульованими ліфтами замість одношвидкісних ліфтів. Подібні розробки забезпечують підвищення якості в ліфтових системах, розвиток а також урізноманітнити використовувані інструменти та інструменти для налаштування та технічного обслуговування або ремонту, а отже, полегшення роботи порівняно з попередніми роками та підвищення надійності ліфт.

Однак однією з проблем є брак досвідчених кадрів на ринку. Механічні знання для здійснення професії, а також електричні, електронні та програмні знання також вимагається. Недоліки в існуючих навчальних практичних комплексах призводять до зміщення освіти в бік більше теорії. Існуючі тестові набори, керовані лише класичними системами керування, мають деякі недоліки наприклад, спробувати деякі збої на тестових наборах експериментально неможливо через роботу напруги в мережі, втрата цілісності моніторингу одночасно на тестових наборах через фактичний розмір моделі ліфтів і потрібно дотримуватися особливої безпеки. Для великих підприємств існують три різні сфери експертиза як монтаж ліфтів, технічне обслуговування ліфтів та несправність ліфтів, але на невеликих підприємствах монтаж і технічне обслуговування можуть виконувати ті самі люди.

## 1. АНАЛІТИЧНА ЧАСТИНА

### 1.1. Огляд системи функціонування ліфта

Ліфт – це вертикальний транспортний засіб, який ефективно переміщує людей або вантажі між поверхами будівлі. Як правило, вони працюють від електродвигунів. Найпопулярнішим ліфтом є канатний. У канатному ліфті кабіна піднімається та опускається за допомогою сталевого канату. Ліфти також мають електромагнітні гальма, які спрацьовують, коли машина зупиняється. Електромагніт фактично утримує гальма у відкритому положенні. Замість того, щоб закривати їх конструкцією, гальма автоматично затискаються, якщо ліфт втрачає потужність. Ліфти також мають автоматичні системи гальмування у верхній і нижній частині шахти ліфта.

Багато сучасних ліфтів управляються комп'ютером. Завдання комп'ютера полягає в тому, щоб обробити всю відповідну інформацію про ліфт і повернути двигун на потрібну кількість, щоб перемістити кабіну ліфта в правильне положення. Для цього комп'ютер повинен знати щонайменше три речі:

- Куди люди хочуть потрапити.
- Де знаходиться кожен поверх.
- Де знаходиться кабіна ліфта.

Дізнатися, куди люди хочуть піти, дуже легко. Кнопки в кабіні ліфта та кнопки на кожному поверсі підключені до комп'ютера, коли хтось натискає ці кнопки, комп'ютер реєструє цей запит.

Перший еталонний ліфт був винайдений Архімедом у 312 році. Згідно з деякими джерелами літератури, ліфт був розроблений як трос на конопляній мотузці та приводився в рух руками або за допомогою тварин. Цей тип ліфта був встановлений у Синайському монастирі Єгипту. У 17 столітті ліфти дуже маленького типу встановлювалися в будівлях Англії та Франції. У середині



1800-х існувало багато видів сирних елеваторів для перевезення вантажів. Більшість із них працювали гідравлічно. Перші гідравлічні ліфти використовували плунжер під кабіною для підйому або опускання ліфта. Насос прикладав тиск води до плунжера або сталевій колони всередині вертикального циліндра. У 1852 році Еліша Отіс представив безпечний ліфт, який запобігав падінню кабіни в разі обриву троса. 23 березня 1857 року в Нью-Йорку був встановлений перший пасажирський ліфт Otis. Перший електричний ліфт був побудований Вернером фон Сіменсом у 1880 році.

У 1874 році Дж. Мікер запатентував метод, який дозволив дверям ліфта безпечно відкриватися та закриватися. У 1882 році, коли гідравлічна енергія була добре запровадженою технологією, була створена компанія, яка пізніше отримала назву London Hydraulic Power Company. У 1929 році Кларенс Конрад Кріспен разом із Inclinator Company of America створив перший житловий ліфт.

На рис. 1.1 показаний огляд ліфтової системи. Вона складається з поверхів, на які хоче потрапити пасажир. Кабіна ліфта переміщує її або вгору, або вниз. Датчик прибуття зафіксував прибуття ліфта на відповідний поверх. Кнопка поверху використовується для підйому ліфта на відповідний поверх. Торшер показує індикацію підлоги, а лампа напрямку показує напрямок руху ліфта, вгору чи вниз. Кнопка ліфта використовується для переміщення кабіни ліфта вгору або вниз. Якщо натиснути перемикач ліфта, кабіна ліфта рухається вгору або вниз. Двигун постійного струму є ще одним важливим компонентом ліфтової системи. Якщо натиснути перемикач, двигун постійного струму рухається вперед або назад, щоб рухати ліфт вгору або вниз. Двері ліфтової системи є одним із важливих факторів ліфтової системи. Коли кабіна ліфта зупиняється на певному поверсі, двері ліфта відкриваються, щоб пасажир міг вийти та зайти в кабіну ліфта.

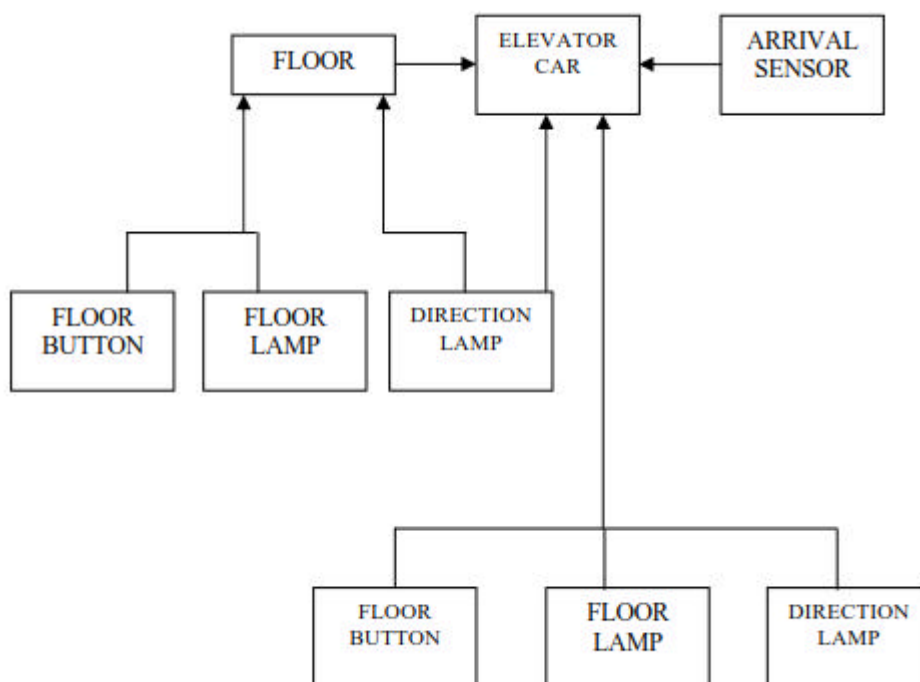


Рис. 1.1. Огляд ліфтової системи

Датчик прибуття використовується на кожному поверсі, для виявлення кабіни ліфта. Коли певна кабіна досягає певного поверху, цей датчик прибуття виявляє кабіну ліфта та зупиняє цю кабіну.

Коли користувач натискає кнопку ліфта, датчик кнопки ліфта надсилає запит на кнопку ліфта в систему, визначаючи поверх призначення, який користувач хоче відвідати. Коли надходить будь-який новий запит, цей новий запит додається до списку поверхів для відвідування. Якщо ліфт стоїть, система визначає, в якому напрямку система повинна рухатися, щоб обслуговувати наступний запит. Система дає команду закрити двері ліфта, коли користувач натискає кнопку закриття дверей ліфта. Коли двері закриті, система дає команду двигуну почати рух ліфта вгору або вниз, залежно від натиснутого перемикача.

Коли ліфт рухається між поверхами, датчик прибуття виявляє, що ліфт наближається до поверху, і сповіщає систему про необхідність зупинити ліфт

і відкрити двері ліфтової системи. На рис. 1.2 показано стратегію диспетчеризації ліфтів.

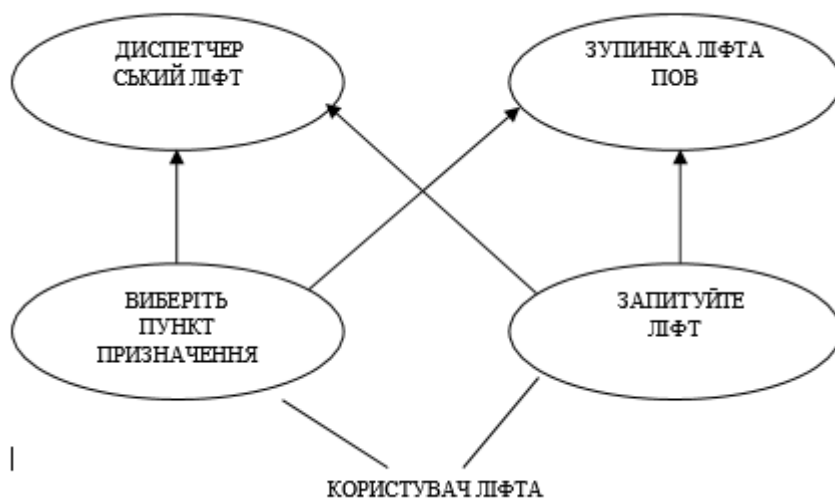


Рис. 1.2. Стратегія диспетчеризації ліфта

## 1.2. Огляд контролера нечіткої логіки для інтелектуального керування групою ліфтів

Тут реалізований нечіткий логічний контролер (FLC) на системі програмованої вентильної матриці (FPGA) для інтелектуального керування ліфтовою системою. Ця конструкція базується на алгоритмі, який розроблено для зменшення обсягу необхідних обчислень шляхом зосередження лише на відповідних правилах та ігнорування тих, які не мають відношення до умов кращої продуктивності групи ліфтових систем.

Тут моделювання було виконано з урахуванням двох вхідних даних, тобто відстані кабіни ліфта та кількості зупинок ліфта. Відстань до ліфта розраховується з урахуванням ряду факторів, таких як  $P_c$  – положення поверху для виклику в холі,  $P_e$  – положення кабіни,  $P_h$  – найвище положення поверху,  $P_l$  – положення найнижчого поверху. На основі цих даних розраховується дистанція кожної секції.

Після цього значення дистанції ліфта разом із номером зупинки кожного застосовується до нечіткого контролера для розрахунку індексу продуктивності (PI) кожного боксу, і той, який має максимум (PI), дає відповідь на виклики залу. Це сприяло б зменшенню середнього часу очікування (AWT) пасажера.

Алгоритми диспетчеризації є найважливішими аспектами в системі керування ліфтом, ці алгоритми можуть зменшити середній час очікування пасажера до певного значення, а також зменшити споживання електроенергії ліфтовою системою. Тут ми реалізуємо шість типів алгоритмів диспетчеризації.

Коллективні алгоритми (CU).

Коллективні алгоритми зниження (CD).

Алгоритми вибіркового підвищення (SU).

.Вибіркове зниження (SD).

Вибірковий - Коллективний UP (SCU).

Коллектив – Selective Down (CS -D).

Виходячи з обсягу трафіку та відсотка трафіку, будь-який із алгоритмів вибирається в конкретний момент часу, таким чином зменшується середній час очікування та споживання електроенергії ліфтовою системою.

Тут ми реалізуємо вбудовану систему керування позиціонуванням ліфта (PCS) на мікроконтролерній системі HCS-12 (MC9s12dp256b) для інтелектуального керування ліфтовою системою. Пошук інтелектуального групового контролера, який може задовольнити багатокритеріальні вимоги до системи управління групою ліфтів, став великим викликом для дослідників. Цей запропонований підхід базується на модулі керування MCU, модулі драйвера двигуна постійного струму, модулі дисплея та ключовому модулі. MCU керує швидкістю та напрямком двигуна постійного струму, вводячи сигнал ШІМ у схему драйвера. Дисплейний модуль показує інформацію про стан роботи ліфта в реальному часі. Шлях руху ліфта

встановлюється клавiшею, залежно від натиснутої клавiші ліфт рухається вгору або вниз. У цьому проекті використовуються два інфрачервоні датчики: один для виявлення кабіни ліфта на конкретному поверсі, а інший для відкриття та закриття дверей ліфта. Цей проект також реалізовано за допомогою контролера FUZZY PID для забезпечення інтелекту кабіни ліфта в різних умовах навантаження для безперебійної роботи ліфта.

### **1.3. Нечітка логіка для вбудованих систем**

Інтелектуальні системи стають все більш поширеними з точки зору їх застосування та їх реалізації. Хоча великі системи залишатимуться важливими, наприклад, для торгівлі та промисловості, менші вбудовані інтелектуальні системи також почали з'являтися вдома та на робочому місці. Нечітка логіка поширюється від традиційної чіткої межі арізолелівської логіки (істинне чи хибне) і включає концепцію часткової істини, яка має значення істинності між повністю істинним і повністю хибним. Мікроконтролер Motorola HC68HC12 (HCS-12) містить декілька примітивів нечіткої логіки безпосередньо у своєму наборі інструкцій. Набір інструкцій містить нечіткі логічні операції оцінки трапецієподібного правила приналежності та зваженого середнього, дефазифікацію. Мікроконтролер також містить інші інструкції, які є корисними в застосуванні нечіткої логіки, наприклад інструкції MIN / MAX. HC12 від Motorola дозволяє розробляти програми низького рівня, які можуть використовувати унікальні функції нечіткої логіки.

Основна проблема вирішується, коли ми розробляємо код VHDL для нечіткого логічного керування для системи управління ліфтом. У розробці коду нечіткої логіки оператор ділення не можна синтезувати в процесі оцінки правила та дефузифікації, отже, потрібні деякі модифікації в алгоритмах,

тобто окрема програма VHDL для оператора ділення записується в загальному процесі.

Ще одна проблема вирішується під час проектування апаратного забезпечення з мікроконтролером (HCS-12) MC9SI2dp256B. Основна проблема виникає, коли ми розробляємо іншу схему інтерфейсу з мікроконтролером MC9SI2dp256B. Загальна експериментальна установка виконується з використанням плати мікроконтролера (HCS-12) (MC9SI2dp256B), схеми пристрою двигуна постійного струму, схеми інтерфейсу клавіш, схеми інтерфейсу датчика та схеми відображення.

## 2 ТЕХНОЛОГІЧНА ЧАСТИНА

### 2.1. Використання контролера нечіткої логіки для групового керування ліфтовими системами

У аналізі та проектуванні інженерного керування управління нечіткою логікою (FLC) на основі ПЛІС для системи групи ліфтів було привабливим, оскільки воно пропонує компроміс між апаратним забезпеченням ASIC спеціального призначення та процесорами загального призначення. Цей цільовий контролер нечіткої логіки на основі FPGA для системи групи ліфтів може скоротити цикл розробки проекту, спростити складність проекту, а також покращити продуктивність керування, що спрощує реалізацію та зменшує витрати на обладнання.

Багато дослідників повідомляли про нечіткий логічний контролер для групового керування ліфтовою системою. Вони описали перевірку п'яти алгоритмів диспетчеризації для системи ліфтів, які були реалізовані на платі на основі ПЛІС Spartan-3 у комплексному підході, зменшуючи площу та покращуючи продуктивність. Загальна система складається з кількох LCS, які реалізують алгоритм диспетчеризації. Нечітка логіка на основі EGCS (FEGCS) працює на ПК та в різних дорожніх ситуаціях визначає найкращі алгоритми для кожного LCS, щоб зменшити середній час очікування пасажирів, а також зменшити споживання електроенергії.

У літературі представлено нечітку логіку на основі контролера ліфтової групи зі схемою самоналаштування для зменшення середнього часу очікування (AWT) пасажирів. Розробка самонастроювального нечіткого логічного контролера для інтелектуального керування ліфтовою системою.

Нечіткий контролер, описаний у наведеній вище літературі, оцінив шість наборів правил, і кожен набір правил складається з різної кількості

правил (від 12 до 14). Отже, час нечітких обчислень великий для відповіді на виклик Холла, навіть якщо це самонастроювальний нечіткий контролер.

У цьому дослідженні ми представляємо техніку висновку Мамдані з алгоритмом, розробленим для зменшення обсягу необхідних обчислень шляхом зосередження лише на відповідних правилах та ігнорування тих, які не мають відношення до умови, що означає, що в процесі фазифікації для будь-якого окремого правила чітке значення у вхідних даних лише два суміжні нечіткі значення є значущими. Ігноруючи незначні нечіткі значення, кількість нечітких вихідних сигналів можна зменшити з п'яти до двох.

Це означає, що блок фазифікації має п'ять виходів, тобто п'ять функцій належності для кожного входу. З п'яти функцій приналежності лише дві функції приналежності є значущими для будь-якого конкретного входу.

Для двох входів кількість вихідного сигналу можна зменшити з двадцяти п'яти до чотирьох. У цій техніці ми отримуємо доступ до вмісту таблиці FAM через невелике вікно, і лише чотири суміжні правила можна переглядати в цьому вікні одночасно. Таким чином, замість доступу до 25 правил механізм логічного висновку повинен мати доступ до чотирьох правил під час кожного обчислення.

У цій роботі моделювання проводилося з урахуванням двох вхідних даних, тобто відстані кабіни ліфта та кількості зупинок. Ці два вхідні значення визначаються за допомогою числа факторів, що означає, що коли ми обчислюємо відстань автомобіля, положення автомобіля та його напрямок повинні бути відомі. Таким чином, відстань автомобіля можна розрахувати шляхом порівняння положення автомобіля та напрямку його руху відносно підлоги залу. У пропонуваній методиці розглядаються чотири можливі положення та напрямок кабіни ліфта для верхнього поверху та нижнього коридору.

Розрахунок кількості зупинок залежить від кількості викликів холу та кабіни кабіни ліфта до того, як він досягне відповідного поверху холу.



Цей алгоритм змодельовано за допомогою VHDL і реалізовано в FPGA. Після порівняння з нечітким правилом, тобто технікою висновку Мамдані, яка використовує механізм логічного висновку, який запускає всі 25 правил під час кожного розрахунку. Потім обчислюється середній час очікування пасажирів для обох методів, і з'ясується, що АWT є мінімальним в алгоритмі методу скороченого правила порівняно з фактичним методом висновку Мамдані.

## 2.2. Опис архітектури FPGA

FPGA - це абревіатура від Field Programmable Gate Array. Він належить до класу програмованих користувачем цифрових пристроїв, які називаються програмованими логічними пристроями (PLD). Програмований логічний пристрій — це інтегральна схема, яка дозволяє користувачеві налаштувати її багатьма способами, уможливаючи реалізацію різноманітних цифрових логічних функцій різного розміру та складності. PLD можна класифікувати за різними категоріями.

Прості програмовані логічні пристрої (SPLD).

- Програмована логічна матриця (PLA): Програмована логічна матриця - це інтегральна схема, яка містить два рівні програмованої логіки; площину I та площину АБО.

- Програмована масивна логіка (PAL): PAL - це інтегральна схема, яка містить фіксовану площину АБО, за якою слідує програмована площина I.

2. Комплексний програмований логічний пристрій (CPLD)

3. Програмована вентильна матриця (FPGA)

Типова FPGA складається з таких компонентів:

1. Програмовані логічні блоки.

2. Ресурси взаємозв'язку.

### 3. Блоки введення виводу.

Загальна схема FPGA показана на рис. 2.1.

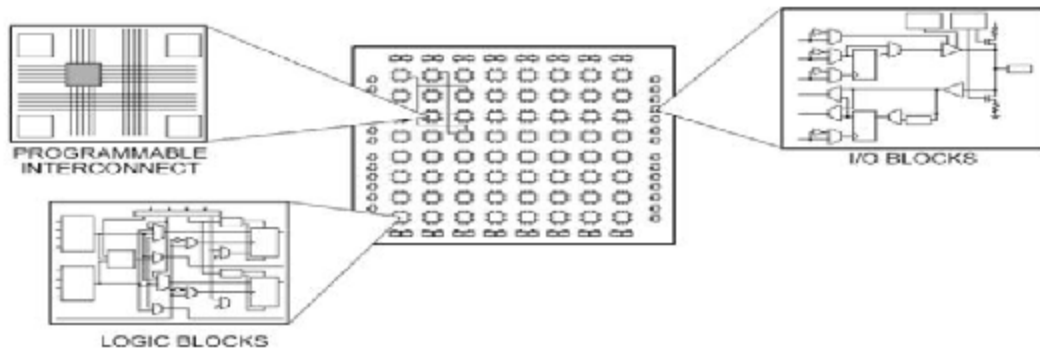


Рис. 2.1. Схема FPGA.

Програмований логічний блок у типовій FPGA складається з конфігурованих логічних блоків (CLB). CLB можна реалізувати різними способами; одним із них є CLB на основі Look Up Table (LUT). LUT - це область пам'яті шириною в один біт. Рядки адреси пам'яті є входами для LUT, а однобітовий вихід є виходом LUT. Таким чином, LUT з  $K$ -входами діє як пам'ять розміром  $2^k$  на 1 біт, і користувач може безпосередньо реалізувати будь-яку вхідну функцію  $k$  шляхом програмування таблиці істинності функцій у LUT.

На діаграмі вище показано узагальнений CLB, який можна використовувати для реалізації будь-якої логічної функції до дев'яти входів; дві окремі логічні функції з чотирма входами та багато інших можливостей. CLB також має D-тригер, який можна використовувати для реалізації послідовних логічних функцій. CLB також має функції, які підтримують інтеграцію цілих систем. Він також має певну спеціалізовану схему, яка дозволяє виконувати арифметичні операції, такі як додавання, множення тощо, швидким і ефективним способом. Користувачі також можуть налаштувати LUT у CLB як розташування RAM для читання/запису.

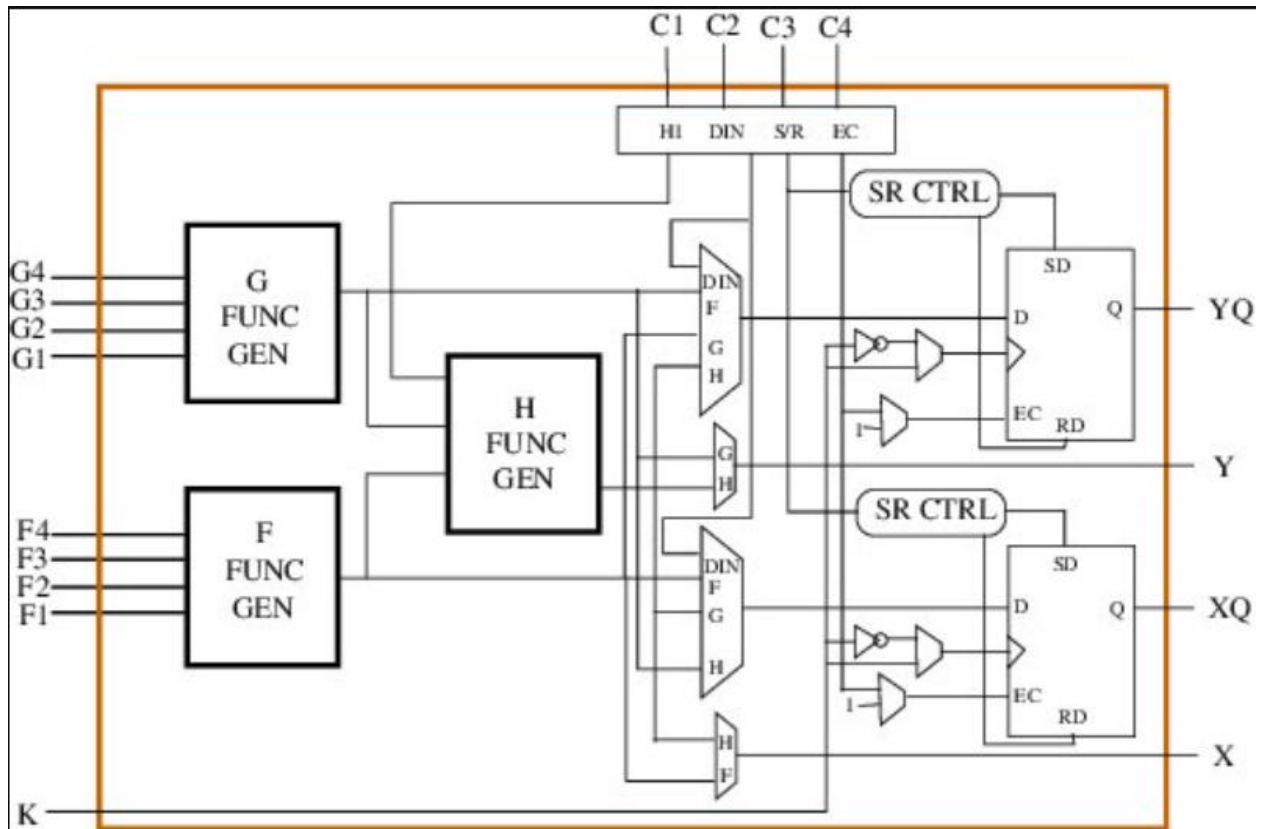


Рис. 2.2. Схема Xilinx FPGA-CLB

Деякі FPGA також дозволяють конфігурувати свої LUT як подвійний порт оперативної пам'яті; з одним записом і двома входами читання. Мікросхеми також містять дуже широкі площини I по периферії CLB, щоб полегшити реалізацію широких декодерів. Деякі з сучасних FPGA також включають цілі мікроконтролери на чіпі; полегшення реалізації складних логічних функцій на одному чіпі. Це особливо підходить для програм керування.

Іншою найважливішою особливістю, яка визначає продуктивність FPGA та її придатність для додатків керування, є її ресурси з'єднання. Це пояснюється тим, що ресурси взаємозв'язку дозволяють реалізувати цілу цифрову систему, забезпечуючи засоби з'єднання різних окремих схем (підсистем), які були реалізовані на різних CLB у FPGA. Ресурси з'єднання в типовій FPGA можна класифікувати як:

1. З'єднання загального призначення: Сигнал між блоками CLB і блоками вводу-виводу (IOB) можна направляти через матриці комутаторів, коли вони проходять уздовж горизонтальних і вертикальних ліній з'єднання.

2. Прямі з'єднання: Сусідні CLB з'єднані між собою напряму.

3. Довгі лінії: Довгі лінії забезпечують високе віяло та низький перекис розподілу сигналів

повинні подорожувати на відносно великі відстані. Вони охоплюють всю довжину або ширину зони сполучення. Зазвичай вони використовуються для тактових сигналів.

Міжз'єднання FPGA зазвичай не сегментовані, тобто кожен сегмент проводки охоплює лише один логічний блок, перш ніж він закінчується в комутаційній коробці. Комутаційна коробка — це комутаційна матриця, яка містить програмовані з'єднання для всіх сегментів проводки, що завершуються всередині неї. Увімкнувши деякі програмовані перемикачі в комутаційній коробці, можна побудувати довші шляхи [6]. На рис. 2.3 показана типова схема з'єднання FPGA.

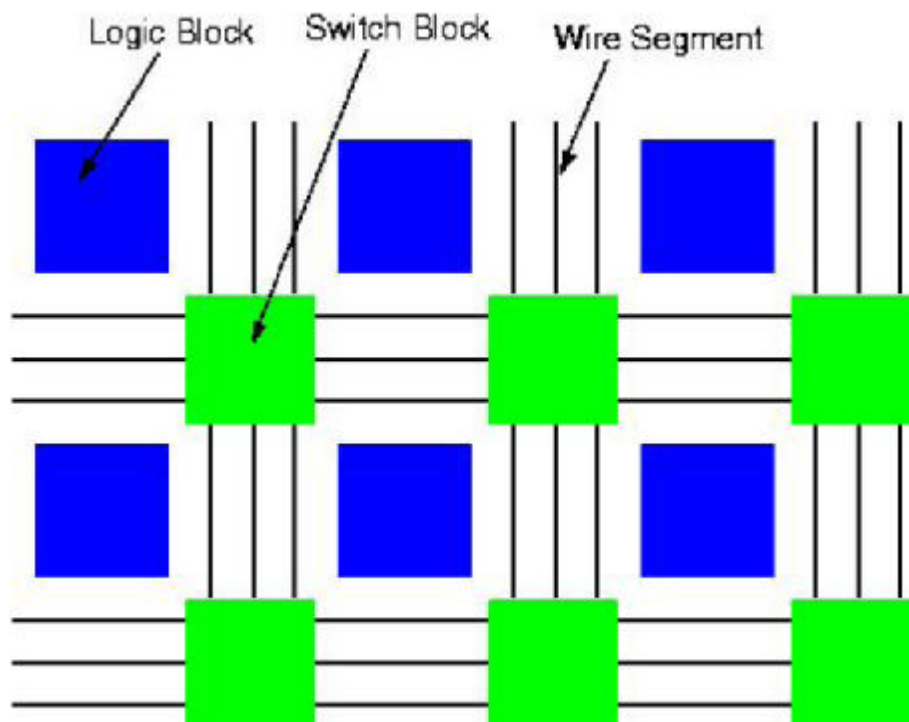


Рис. 2.3: Схема з'єднання FPGA

## Блоки введення-виведення (IOB)

IOB забезпечує інтерфейс між FPGA і сигналами реального світу. IOB в основному складається з колодок введення/виведення. Колодки вводу/виводу підключаються до одного з контактів на корпусі ІС, щоб зовнішні сигнали могли вводитися або виводитися з масиву логічних комірок. Він також складається з буферів трьох станів, які дозволяють сигналам вводити та виводити з логічного масиву. Тригери передбачені для того, щоб вхідні та вихідні значення могли зберігатися в IOB. Кожен IOB також має низку інших функцій, таких як перепрограмування вхідного порогу для відповіді на рівні логіки TTL або CMOS. Він також містить контроль швидкості наростання вихідного сигналу та включає внутрішні підтягуючі резистори, щоб уникнути плаваючих входів [9].

FPGA може бути дрібнозернистим або крупнозернистим пристроєм. Точна ПЛІС складається з великої кількості програмованих логічних ресурсів малої ширини, які можна використовувати для реалізації різноманітних функцій. Типовим прикладом такої FPGA може бути Atmel AT40K. Грубозерниста FPGA, така як серія Xilinx Virtex, складається з меншої кількості більш потужних логічних блоків, таких як LUT і тригери. Сучасні FPGA також мають такі функції, як низька напруга

Диференціальна сигналізація (LVDS), а також підтримка програмування вхідного порогу для відповіді на LVTTL, LVCMOS тощо. Вони також забезпечують функції дискретно керованого опору (DCI). Більшість FPGA також включають підтримку з'єднання периферійних компонентів (PCI); за допомогою якого їх можна з'єднати з комп'ютером загального призначення або стати частиною більшої розробки. FPGA також сумісні з JTAG, тобто вони підтримують архітектуру граничного сканування IEEE 1149.1-1990; що дозволяє послідовно завантажувати тестові дані в пристрій і послідовно зчитувати результати тестування. JTAG також можна використовувати для завантаження потоків бітів конфігурації в FPGA. Іншою

важливою особливістю FPGA є системне програмування (ISP), яке дозволяє програмувати FPGA, коли він є частиною кінцевої цільової системи. Це усуває необхідність фізичного видалення чіпа із системи та полегшує програмування.

### 2.3. Методологія керування нечіткою логікою для групової системи ліфта

Звичайний FLC розроблено на основі простої концепції шляхом зменшення кількості правил, що сприяє скороченню часу обчислення контролера нечіткої логіки для кращої продуктивності системи керування групою ліфтів. В основному FLC ділиться на чотири компоненти. На рисунку 2.4 показана схема базової структури запропонованого контролера нечіткої логіки на основі ПЛІС для інтелектуального керування системою ліфта. Він складається з модуля фазифікації, бази правил, механізму нечіткого інтерфейсу, модуля дефазифікації.

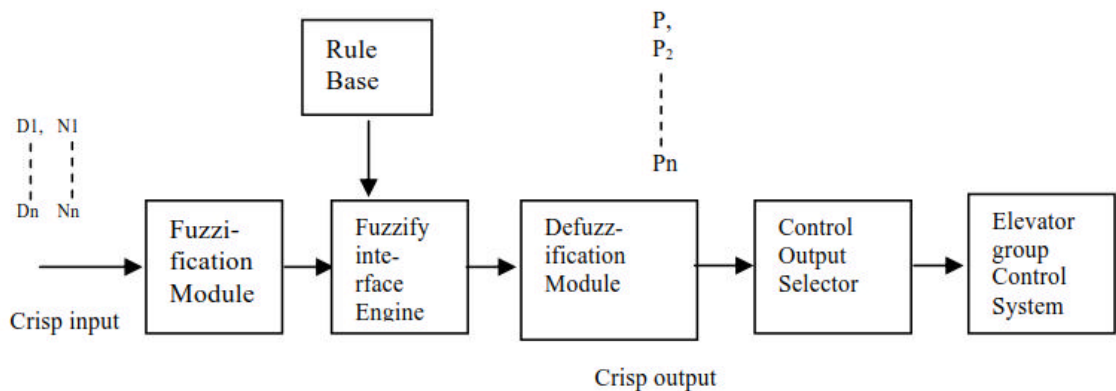


Рис. 2.4. Базова структура запропонованого контролера нечіткої логіки для системи управління групою ліфтів

Блок фазифікації.

Блок фазифікації має  $p$  виходів, по одному для кожного нечіткого значення, визначеного у вхідних даних Universe of Discourse. Однак процес фазифікації передбачає, що для будь-якого окремого чіткого значення

вхідного  $X_i$  лише два сусідніх нечітких значення є значущими (з ненульовими значеннями приналежності). Ігноруючи незначущі нечіткі значення, кількість вихідних сигналів також можна зменшити з п'яти до двох. Можливі комбінації значущих нечітких значень для довільного введення:

$$B_i^1 \text{ and } B_i^2; \text{ and } B_i^2 \text{ and } B_i^3; B_i^3 \text{ and } B_i^4; \text{ and } B_i^4 \text{ and } B_i^5$$

Виявлено, що використовуючи лише три змінні,  $ADR_i$ ,  $B_{i\_A}$  та  $B_{i\_B}$ , усі комбінації можуть бути достатньо представлені для будь-якого значення  $x_i$ , як показано наступними твердженнями:

$$ADR_i = \text{"00"} : B_{i\_A} = B_i^1, B_{i\_B} = B_i^2$$

$$ADR_i = \text{"01"} : B_{i\_A} = B_i^2, B_{i\_B} = B_i^3$$

$$ADR_i = \text{"10"} : B_{i\_A} = B_i^3, B_{i\_B} = B_i^4$$

$$ADR_i = \text{"11"} : B_{i\_A} = B_i^4, B_{i\_B} = B_i^5$$

Рис. 2.5 ілюструє, як ці умови відповідають універсуму дискурсу.

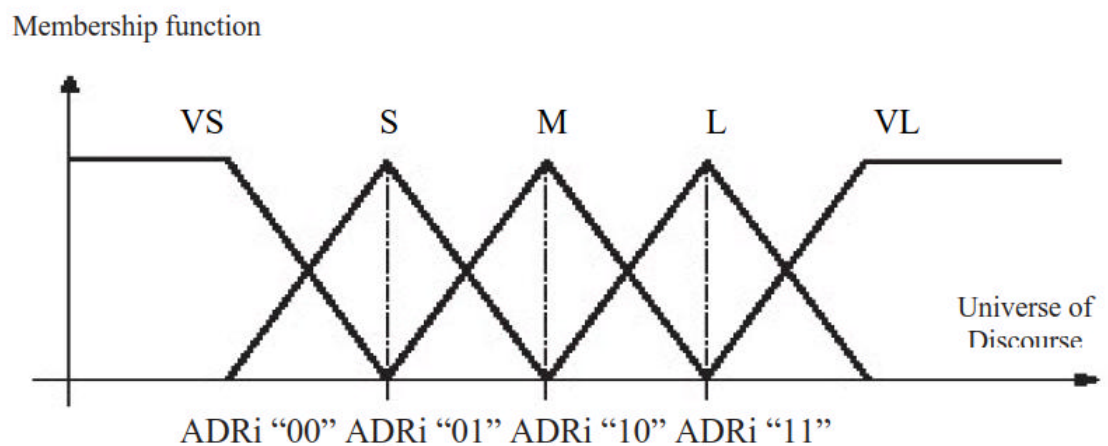


Рис. 2.5. Вхідний нечіткий набір

Міні таблиця FAM

Таблиця FAM конструкції FLC показана в таблиці I. Тут ми описуємо алгоритм, розроблений для зменшення обсягу необхідних обчислень шляхом зосередження лише на відповідних правилах та ігнорування тих, які не мають відношення до умови. Відомо, що для кожного набору вхідних даних лише чотири нечітких значення (по два на кожен вхід) є актуальними в будь-який момент часу.

Простіший спосіб пояснити цю техніку – це уявити всю таблицю FAM закритою з поля зору. Доступ до вмісту таблиці FAM дозволено лише через невелике вікно, і лише чотири суміжні правила можна переглядати в цьому вікні одночасно. Таким чином, замість того, щоб мати доступ до 25 правил, механізм логічного висновку має отримати доступ лише до чотирьох правил під час кожного обчислення. Вікно може переміщатися по таблиці FAM, і його позиція визначається індексом  $j$ , визначеним як:

$$\begin{aligned} \text{ADR1} = \text{"00"} \ \& \ \text{ADR2} = \text{"00"} \ \rightarrow \ j = 0 \\ \text{ADR1} = \text{"00"} \ \& \ \text{ADR2} = \text{"01"} \ \rightarrow \ j = 1 \\ \text{ADR1} = \text{"00"} \ \& \ \text{ADR2} = \text{"10"} \ \rightarrow \ j = 2 \\ \text{ADR1} = \text{"00"} \ \& \ \text{ADR2} = \text{"11"} \ \rightarrow \ j = 3 \\ \text{ADR1} = \text{"01"} \ \& \ \text{ADR2} = \text{"00"} \ \rightarrow \ j = 4 \\ \text{ADR1} = \text{"01"} \ \& \ \text{ADR2} = \text{"01"} \ \rightarrow \ j = 5 \\ \text{ADR1} = \text{"11"} \ \& \ \text{ADR2} = \text{"11"} \ \rightarrow \ j = 15 \end{aligned}$$

Всього існує шістнадцять «положень вікон», і перші шість показані на рис. 2.6. Заштриховані блоки - це правила, які вважаються релевантними для вхідних умов, що відповідають індексу  $j$ . Коли техніка вікна застосована до таблиці FAM у Таблиці I, спостерігається, що кількість міні-таблиць FAM ідентична (наприклад,  $j = 1$  і  $j = 4$ ). З 16 міні-таблиць FAM є лише сім унікальних таблиць, які показані на рис. 2.7. Якщо WIN є індексом для нового набору таблиць, то таблиці можна впорядкувати за допомогою наступного.



```

IF j=0 THEN WIN="0000"
IF j=1 OR j=4 THEN WIN="0001"
IF j=2 OR j=5 OR j=8 THEN WIN="0010"
IF j=11 OR j=14 THEN WIN="0101"
IF j=15 THEN WIN="0110"

```

Таблиця 2.1 - Таблиця FAM конструкції FLC

Ele_car_dist Ele_no_sto	VS	S	M	L	VL		
VS	PVB	PB	P	PS	Z		
S	PB	P	PS	Z	NS	NVB	Negative Very Big
M	P	PS	Z	NS	N	NB	Negative Big
L	PS	Z	NS	N	NB	N	Negative
VL	Z	NS	N	NB	NVB	NS	Negative Small
						Z	Zero
						PS	Positive Small
						P	Positive
						PB	Positive Big
						PVB	Positive Very Big

Цей алгоритм вимагає значної кількості операцій IF-THEN і не обов'язково є ефективним способом впровадження проекту в апаратне забезпечення. Спостерігаючи шаблон у вихідній таблиці FAM, можна показати, що міні-таблиці FAM є ідентичними, коли сума ADR1 і ADR2 однакова. Тому замість використання численних операцій IF-THEN компонування міні-таблиць FAM досягається за допомогою єдиного додавання.

$$WIN \leq ("00"&ADR1) + ADR2;$$

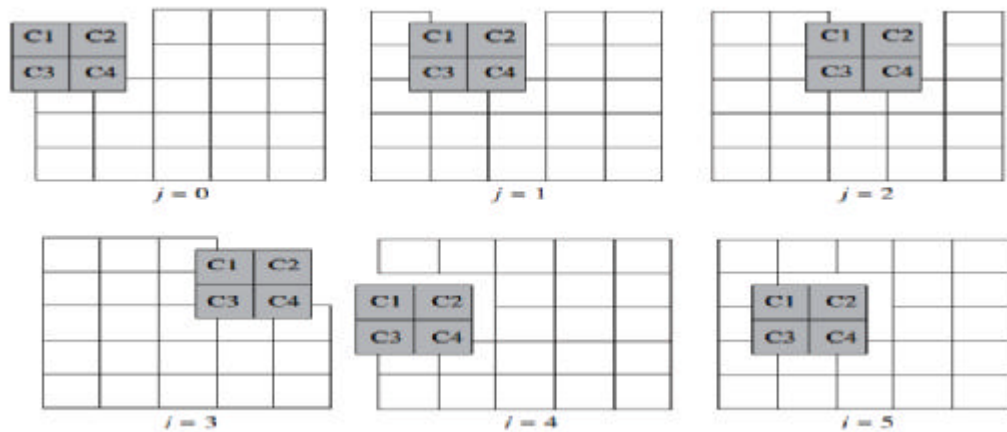


Рис.2.6. Міні-таблиця FAM

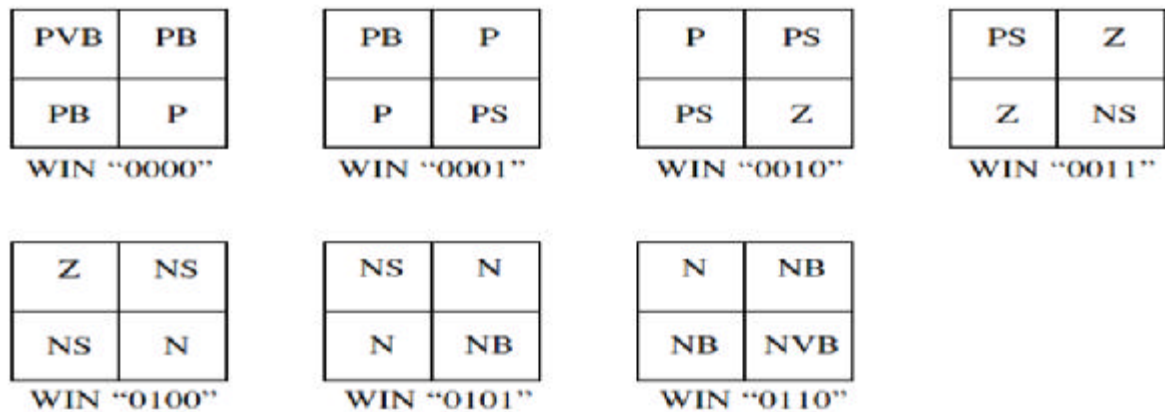


Рис. 2.7. Міні – таблиці FAM для дизайну FLC

Функція оператора («00»&ADR1) полягає в тому, щоб розширити значення ADR1 з 2 бітів до 4 бітів таким чином, щоб воно було сумісним з 4-бітним сигналом WIN. Змінні всередині таблиці mini-FAM згодом обробляються в розділі коду, який позначено як «Mini-Fuzzy Inference Engine». В оригінальному коді механізм логічного висновку містить 25 MIN-операцій. Модифікований код складається лише з чотирьох MIN-операцій, що є помітним скороченням.

#### Алгоритм дефазифікації

Функція дефазифікації полягає у перетворенні нечіткого вихідного значення системи керування у відповідне чітке значення функції приналежності, показаної на рис. 2.8. Це досягається за допомогою методу

середньозваженої дефазифікації. Ця операція дефазифікації вимагає кількох множників і дільника. Поведінкове моделювання у VHDL підтримує множення та ділення, але ці операції складно реалізувати на етапах синтезу та реалізації.

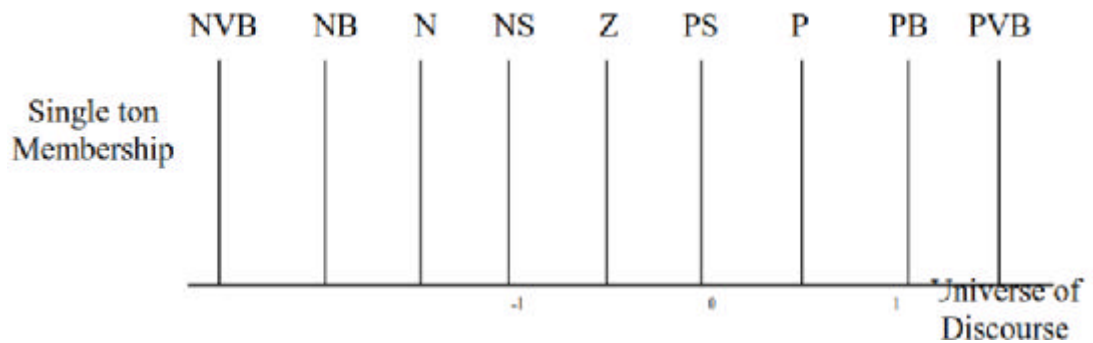


Рис. 2.8. Вихідний нечіткий набір

У запропонованому підході розглядаються лише чотири важливі наслідки правил. Таким чином, кількість правил-наслідків, які потрібно агрегувати, зменшується, але розподіл правильних вагових коефіцієнтів для значущих вихідних значень стає трохи складнішим. З таблиць на рис. 4 очевидно, що незалежно від значення WIN, консеквенти C2 і C3 завжди вказують на те саме нечітке значення (наприклад, коли WIN = «0000»: C1->PVB, C2->PB, C3->PB, C4->P). Це означає, що лише C2 і C3 повинні бути агреговані, отже

$$DA = C1$$

$$DB = \max[ C2, C3 ]$$

$$DC = C4$$

де DA, DB і DC представляють функцію належності вихідних нечітких значень. Таким чином, модифікований вихідний нечіткий набір показаний на рис. 2.9.

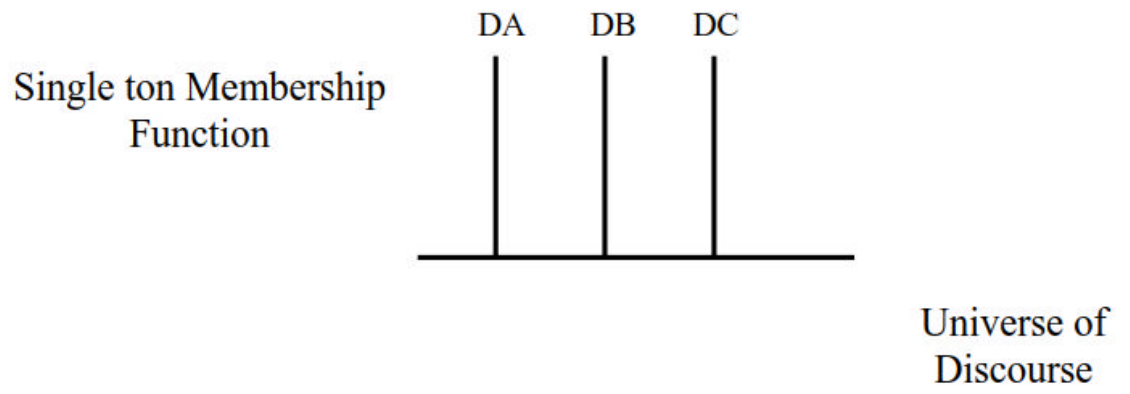


Рис.2.9. Модифікований вихідний нечіткий набір

### 3 КОНСТРУКТОРСЬКА ЧАСТИНА

#### 3.1. Розроблена методика для керування ліфтовою групою

Визначення значень двох вхідних змінних для нечіткого оцінювання не повинно ігнорувати важливі фактори, які впливають на його результати. Як і при обчисленні відстані автомобіля, необхідно знати положення та напрямок руху кабіни.

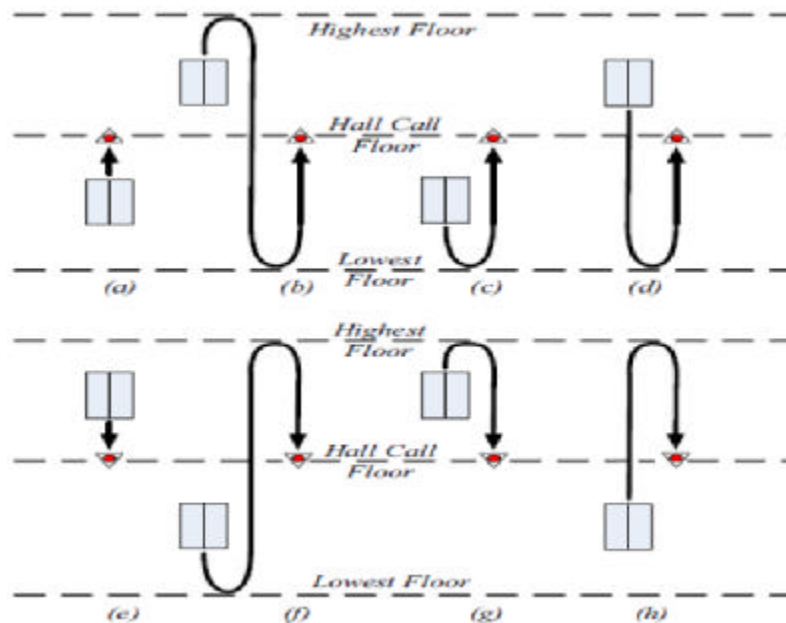


Рис. 3.1. Різні положення, яке займає автомобіль, щоб дістатися до кімнати для виклику.

Відстань автомобіля можна розрахувати шляхом порівняння положення автомобіля та напрямку його руху з даними виклику в залі. Як показано на рис. 3.2, для того, щоб автомобіль прибув на поверх зали виклику, є чотири можливі позиції для напрямку вгору та ще чотири для напрямку вниз. Спочатку розгляньте четвірку для дзвінка в зал. Сканування, як видно з рис. 3.2, окрім інформації, згаданої вище, для позиції (b)–(d) потребує додаткових даних для обчислення, найвищого та найнижчого

поверхів, які обслуговує автомобіль для позиції (b), і лише найнижчий поверх для позицій (c) і (d). З них можна вивести три рівняння для визначення пройденої відстані в кожному випадку, представлені таким чином:

Для позиції (a),

$$d = P_c - P_e \quad (3.1)$$

Для позиції (b),

$$d = (P_h - P_e) + (P_h - P_l) + (P_c - P_l) \quad (3.2)$$

Для позиції (c) і (d),

$$d = (P_e - P_l) + (P_c - P_l) \quad (3.3)$$

де  $d$  - пройдена відстань,  $P_c$  - позиція на підлозі виклику залу,  $P_e$  - позиція автомобіля,  $P_h$  - найвища позиція на поверсі та  $P_l$  - найнижча позиція на поверсі.

Різниця між маршрутами (c) і (d) полягає в різному положенні автомобіля відносно положення підлоги залу. Незалежно від позиції, (2.3) застосовується до обох маршрутів. У випадку дзвінка з нижнього коридору ті самі рівняння можна змінити, щоб врахувати напрямок дзвінка з холу вниз.

Для позиції (e), переставляючи (3.1), отримуємо

$$d = P_e - P_c \quad (3.4)$$

Для положення (f), переставивши (3.2), отримаємо

$$d = (P_e - P_l) + (P_h - P_l) + (P_h - P_c) \quad (3.5)$$

Для позиції (g) і (h), переставляючи (2.3), отримуємо

$$d = (P_h - P_e) + (P_h - P_c) \quad (3.6)$$

Розрахунок кількості зупинок залежить від кількості викликів у зал і кількості викликів автомобіля, які має обслужити автомобіль, перш ніж він зможе досягти поверху для викликів у зал. Усі виклики, включаючи виклики в холл і виклики автомобіля, між поточним положенням автомобіля та поверхом виклику холу, який розглядається для призначення, відображають зупинки, які мають бути зроблені під час подорожі автомобіля.

Підраховуючи кількість цих викликів, ми фактично підраховуємо кількість зупинок. Окрім кількості зупинок, номер поверху, який відповідає кожній зупинці, також записується для подальшого використання під час обчислення.

### 3.2. Стратегія диспетчеризації ліфтів

На рис. 3.3 показано процес вибору найбільш підходящого автомобіля для відповіді на виклик із залу через систему FLC. Тут я розглядаю систему кнопок виклику вгору-вниз, коли зареєстрований виклик у залі. Відповідні дані з усіх автомобілів у групі необхідні для обчислення значення вхідних змінних. Дані, необхідні для кожного автомобіля, це його поточне положення в будівлі, стан руху, швидкість, напрямок руху.

Тут ми використовуємо дві вхідні змінні для нечіткої оцінки, це відстань автомобіля та кількість зупинок. Їх визначення такі.

1) Відстань автомобіля: відстань, яку проїхав автомобіль для переміщення від поточного положення до кімнати виклику в зал, коли зареєстрований виклик в зал.

2) Кількість зупинок: кількість зупинок на поверхах, на яких автомобіль має зупинитися, щоб завантажити або висадити пасажирів, перш ніж буде досягнуто поверху для виклику.

Для кожного автомобіля в групі значення двох вхідних змінних обчислюються на основі даних, наданих контролером кожного автомобіля. Розраховані значення кожного автомобіля потім обробляються FLC, і на основі нечіткої оцінки кожному автомобілю призначається індекс ефективності (PI). PI позначає придатність автомобіля відповісти на виклик у зал у певний момент, автомобіль із найвищим значенням PI вважається автомобілем, який найбільше підходить для виклику в зал.

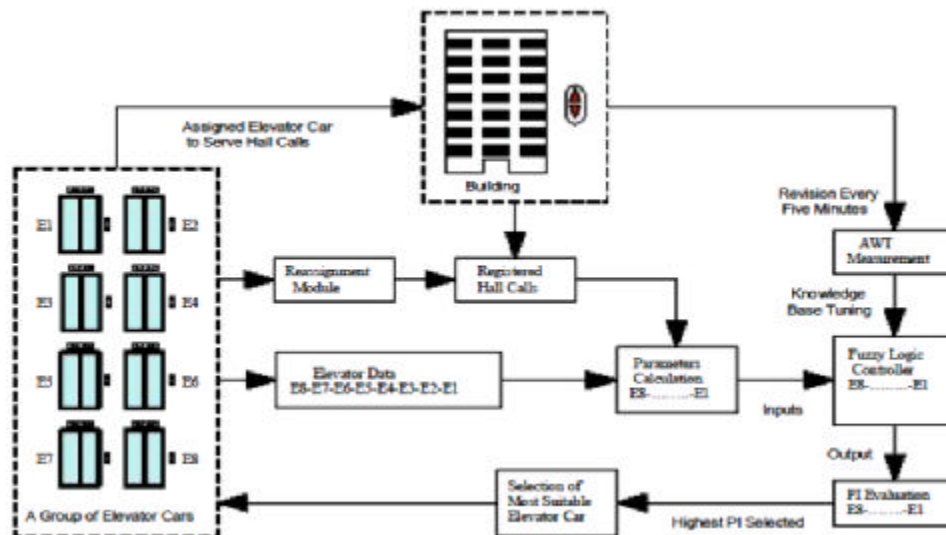


Рис. 3.3. Потік процесу для стратегії диспетчеризації ліфта з використанням контролера нечіткої логіки

### 3.3. Впровадження системи керування ліфтом за допомогою hcs-12 (mc9s12dp256b)

Основною метою даної дипломної роботи є реалізація системи керування ліфтом з використанням мікроконтролера HCS-12, тобто MC9S12dp256B. Для реалізації цього розглядається прототип установки



ліфта, який складається з одного двигуна постійного струму, чотирьох перемикачів рівня, датчика та модуля відображення.

Цей двигун постійного струму з'єднаний із встановленою передачею, одне колесо приєднане до цієї шестерні для переміщення троса ліфта вгору та вниз. Один кінець троса з'єднаний із системою зубчастого колеса, а інший кінець троса з'єднаний з модулем автомобіля. Один оптичний кодер приєднаний до колеса двигуна постійного струму для вказівки швидкості двигуна постійного струму. Це також забезпечує сигнал зворотного зв'язку для нечіткого ПД-регулятора для підтримки постійної швидкості двигуна постійного струму в різних умовах навантаження.

У запропонованій нами чотириповерховій ліфтовій системі ми використовуємо два перемикачі на кожному поверсі, один для переміщення кабіни ліфта вгору, а інший для переміщення ліфта вниз. На кожному поверсі використовується два датчики: один для зупинки кабіни ліфта на відповідному поверсі, а інший для відкриття та закриття дверей ліфта для входу та виходу пасажирів на відповідному поверсі. Цей проект також реалізовано за допомогою контролера FUZZY PID для забезпечення інтелекту кабіни ліфта в різних умовах навантаження для безперебійної роботи ліфта. У цій техніці швидкість і напрямок двигуна постійного струму контролюються мікроконтролером шляхом введення сигналів ШІМ до ланцюгів Н-мосту. Ліфт зупиняється, коли інформація, введена ключем, і сигнали зворотного зв'язку інфрачервоних ланцюгів виявлення збігаються. Напрямок руху ліфта вгору та вниз відображає модуль дисплея.

MC9S12DP256B — це 16-розрядний мікроконтролер, наданий Freescale, що складається з 16-розрядного центрального процесора (ЦП STAR 12), 256 Кбайт флеш-пам'яті EEPROM, 12 Кбайт оперативної пам'яті, 4 Кбайт EEPROM, розширеного таймера захоплення, 8-канального імпульсу. широтний модулятор (ШІМ), два 8-канальних 10-бітних аналого-цифрових перетворювача (АЦП), два асинхронних послідовних інтерфейсу зв'язку

(SCI), три послідовні периферійні інтерфейси (SPI), п'ять програмно-сумісних модулів CAN 2.0 A/B (MSCAN 12) і шина Inter IC.

CP STAR 12 — це високошвидкісний 16-бітний процесор, який має повний 16-бітний тракт даних і ширші внутрішні регістри (до 20 біт) для високошвидкісних розширених математичних інструкцій. Він має канал інструкцій для збільшення швидкості виконання. Він підтримує широкий набір режимів індексованої адресації.

Акумулятор: Регістри A і B — це 8-розрядні накопичувачі загального призначення, які використовуються ЦП для зберігання операндів і результатів їх обробки. У 16-бітній операції це буде розглядатися як акумулятор D.

Індексні регістри: IX і IY є індексними регістрами для збереження ефективної адреси в режимі індексованої адресації.

Показчик стека: Показчик стека вказує на останнє місце використання стека. Його також можна використовувати як показчик у режимі індексованої адресації. ЦП підтримує автоматичну програму, яка зберігає системні контексти під час виклику підпрограми та переривань.

Лічильник програми: Цей 16-бітний регістр вказує на наступну інструкцію, яку потрібно виконати. Його можна використовувати в усіх режимах індексованої адресації, крім автоматичного збільшення/зменшення.

#### Система контролю позиціонування ліфта

Основна структурна схема моделі системи керування позиціонуванням ліфтової системи показана на рисунку 3.5. Модель включає модуль керування MCU, модуль драйвера двигуна постійного струму, модуль інфрачервоного виявлення, модуль дисплея та модуль ключа. Маршрут руху ліфта встановлюється за допомогою клавіш, інформація про місцезнаходження ліфта в реальному часі встановлюється за допомогою клавіш, а інформація про місцезнаходження ліфта в реальному часі

виявляється інфрачервоними схемами виявлення та передається назад до MCU.

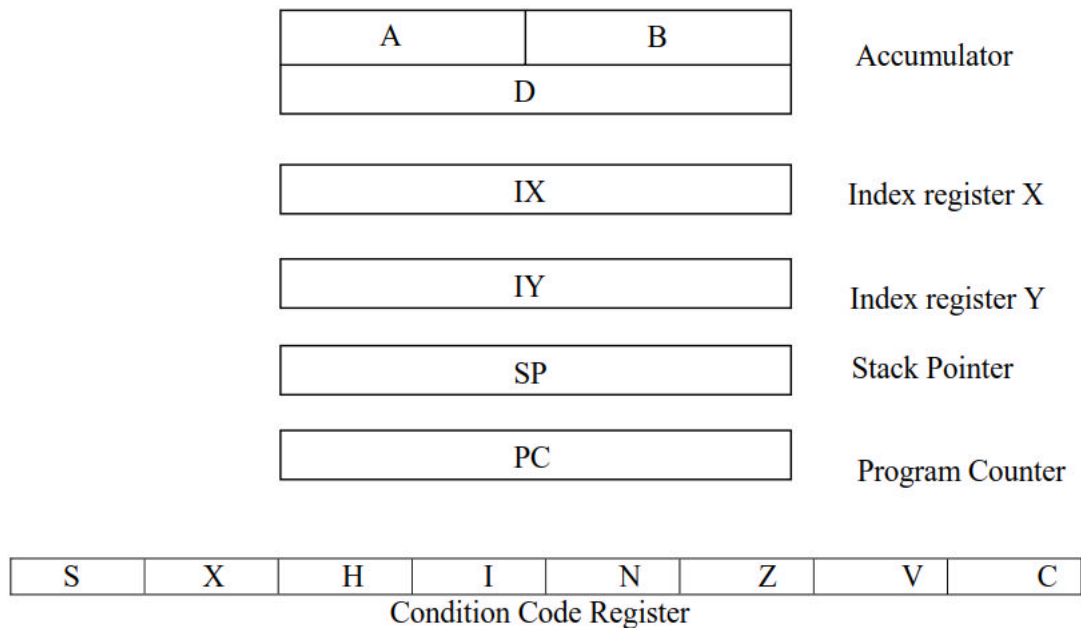


Рис. 3.4. регістри процесора CPU 12

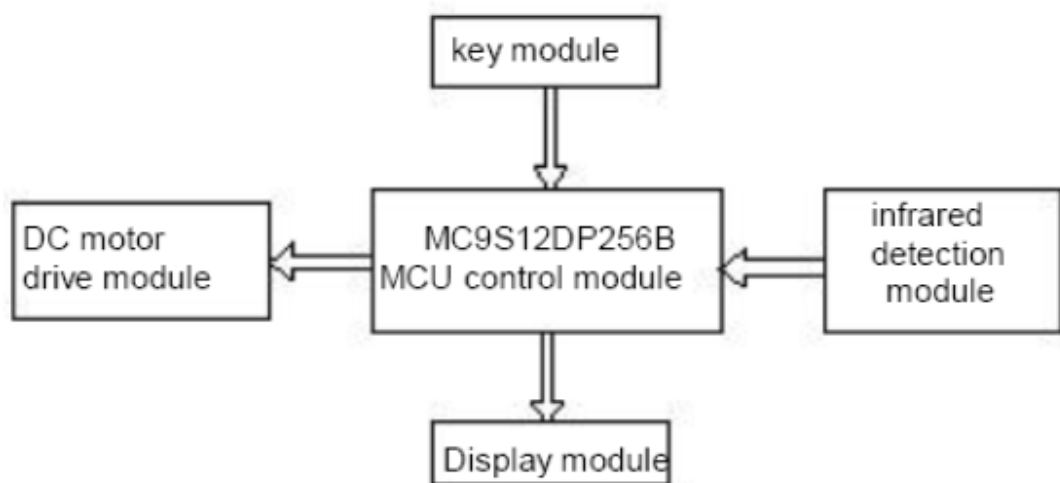


Рис. 3.5. Структурна схема моделі ліфтової системи

Швидкість і напрямок двигуна постійного струму контролюються через MCU шляхом введення імпульсних сигналів синхронізації в його привід, тобто в схеми приводу Н-мосту. Ліфт зупиняється, коли інформація, введена ключем, і сигнали зворотного зв'язку інфрачервоних ланцюгів виявлення

збігаються. Статус роботи ліфта в реальному часі відображається через систему відображення.

### 3.4. Проектування апаратного забезпечення системи

Конструкція приводного модуля двигуна постійного струму.

Схема драйвера двигуна постійного струму показана на рис. 3.6. ШІМ-вхід схеми драйвера двигуна постійного струму з'єднаний з вихідним портом ШІМ-каналу мікроконтролера, а ще два порти схеми драйвера двигуна постійного струму з'єднані з вихідним портом мікроконтролера, який змушує двигун постійного струму обертатися в позитивному або зворотному напрямку. Швидкість обертання двигуна постійного струму прямо пропорційна частоті вхідного імпульсу, що означає, що чим вища частота вхідного імпульсу, тим швидше обертається двигун постійного струму, і навпаки.

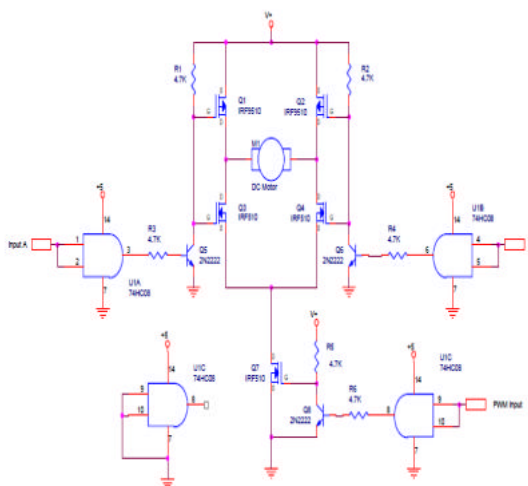


Рис. 3.6. Схема драйвера двигуна постійного струму

Ця схема зв'язує вхід Q1 з Q3, а вхід Q2 з Q4. Як наслідок, Q1 і Q3 або Q2 і Q4 ніколи не будуть увімкнені одночасно, оскільки пристрої N-MOS і P-MOS активні для сигналів протилежної полярності (тобто, якщо застосовано

заземлення, P-MOS буде активним). активним, поки N-MOS буде активним вимкнено). Додавання двох вентилів AND дозволяє використовувати сигнал ШІМ для керування швидкістю. Він налаштований таким чином, що сигнал ШІМ буде застосовано лише до N-каналу (Q3 і Q4). Причина цього полягає в тому, що для аналогічних N і P каналів MOSFET, N каналний MOSFET зазвичай має меншу ємність затвора і вмикає і вимикається швидше, ніж MOSFET P-каналу. Ефективніше залишити активним один МОП-транзистор, перемикаючи інший сигналом ШІМ. Щоб двигун завжди працював на повній швидкості, установіть постійний вхід ШІМ. Для вентилів AND бажана високошвидкісна мікросхема, така як 74F08. Тут ми використовуємо вентиля AND і прив'язуємо вентиль Q1 до воріт Q3 і воріт Q2 безпосередньо до воріт Q4. Повернення А або В у високий рівень призведе до того, що двигун обератиметься в одному чи іншому напрямку. Якщо вони і високі, і обидва низькі одночасно, то двигун буде активно гальмувати. Якщо нам не потрібне активне гальмування в ланцюзі, ми можемо з'єднати дріт В з проводом А через інвертор, так що В завжди буде протилежним А. Тоді нам потрібно лише перемикати А для керування напрямком та лінію ШІМ для швидкість.

Змінюючи швидкість двигуна, доцільно збільшити її, оскільки раптові зміни можуть спричинити велику зворотну ЕРС і шум. Рампінг означає повільне збільшення швидкості шляхом повільної зміни робочого циклу сигналу ШІМ. Повільний – відносний термін, і часто півсекунди буває більш ніж достатньо для невеликих двигунів. Це залежатиме від двигунів і робота, оскільки нам потрібно буде враховувати інерцію обертання двигуна та силу струму, який споживатиме швидка зміна швидкості двигуна.

#### Модуль інтерфейсу комутатора

На рис. 3.7 показано загальний інтерфейс між ключовим модулем і мікроконтролером HCS-12 (MC9S12DP256B). Алгоритм, який використовується в інтерфейсі ключів, такий

1. Порт AD приймає 8-бітну комбінацію від DIP-перемикачів.

2. HCS12 прочитає значення з порту AD і збереже його в пам'яті.
3. HCS12 виконає кілька бітових операцій зі значенням.
4. Результат обробки бітів буде надіслано на порт T.
5. Порт T надсилатиме значення до схеми драйвера двигуна постійного струму.

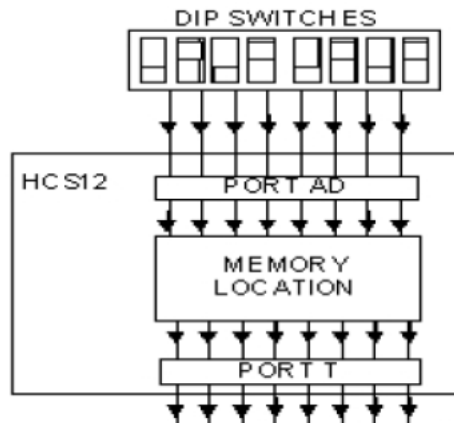


Рис. 3.7. Загальний інтерфейс між ключем модуль і HCS12

Згідно з інструкціями щодо програми, нам потрібно запрограмувати регістри порту Ts як загальний вихід, а PORT AD – як загальний вхід. Ми почнемо з порту T, оскільки він простіше. Регістр DDRT — це 8-розрядний регістр напрямку даних для порту T. Кожен біт відповідає напрямку лінії даних, підключеної до порту T. Це одиниця для виведення та нуль для введення. Нам потрібно налаштувати порт T для 8-бітного виводу, ми призначимо 1 с для всіх бітів DDRT. Регістр PTT — це 8-розрядний регістр передачі, який ми будемо використовувати для надсилання даних до схеми драйвера двигуна постійного струму.

## 4. НАУКОВО-ДОСЛІДНА ЧАСТИНА

### 4.1. Результат моделювання та реалізація fpga

Тут ми розглядаємо два вхідні дані, тобто відстань кабіни ліфта та кількість зупинок. Ці два вхідні значення визначаються з урахуванням ряду факторів, головним чином розташування кабіни ліфта на підлозі в будівлі, напрямку руху, кількості викликів холу та кількості викликів кабіни.

Під час моделювання розглядається поверх холу виклику, з цього поверху холу можна рухатися вгору, щоб досягти вищого цільового поверху будівлі, а інший можна рухатися вниз, щоб досягти нижнього цільового поверху будівлі.

Тут розглядаються різні вхідні дані, такі як номер найвищого поверху, номер найнижчого поверху, номер поверху для виклику холу, положення кожної кабіни ліфта, напрямок, кількість викликів холу та викликів кабіни.

Ці дані надаються механізму інтерфейсу. На основі цих даних механізм інтерфейсу розраховує відстань кабіни ліфта та кількість зупинок до відповідного поверху холу.

Після цього ці дані передаються в нечіткий контролер для розрахунку індексу продуктивності (PI) кожного автомобіля, і контролер оцінює найбільший PI, який позначає придатний автомобіль для відповіді на виклик у зал у певному випадку. Відповідний результат моделювання фазифікації та дефазифікації показано на рисунку 4.1. На основі даних, наданих на вхід нечіткого контролера, кабіна ліфта-1 має найвищий PI для відповіді на виклики холу для виклику холу вгору, а ліфт 5 має найвищий PI для відповіді на виклики холу для виклику холу вниз, що є показано в результаті моделювання на малюнку 4.2.



Рис. 4.1. Результат моделювання фазифікації та результат моделювання дефазифікації

У даній роботі ми розглядаємо вісім номерів кабін ліфтів. Ці вісім машин розподілені в різних положеннях будівлі в різних напрямках, тобто деякі автомобілі рухаються вгору, а інші — вниз.

Враховується поверх для дзвінків із цих двох поверхів. Один зал розташований у верхній частині будівлі, а інший – у нижній частині будівлі.

Залежно від положення, напрямку та кількості зупинок одна кабіна ліфта дасть відповідь за мінімальний час для поверху виклику вгору, а інша кабіна дасть відповідь протягом мінімального часу для викликів у холі вниз. Це сприяло б зменшенню середнього часу очікування (AWT) пасажира.

Тут розглядаються два типи нечітких алгоритмів для системи керування ліфтом. Одним із алгоритмів є техніка виведення типу Мамдані,



яка використовує механізм виведення, який запускає лише чотири правила під час кожного обчислення.

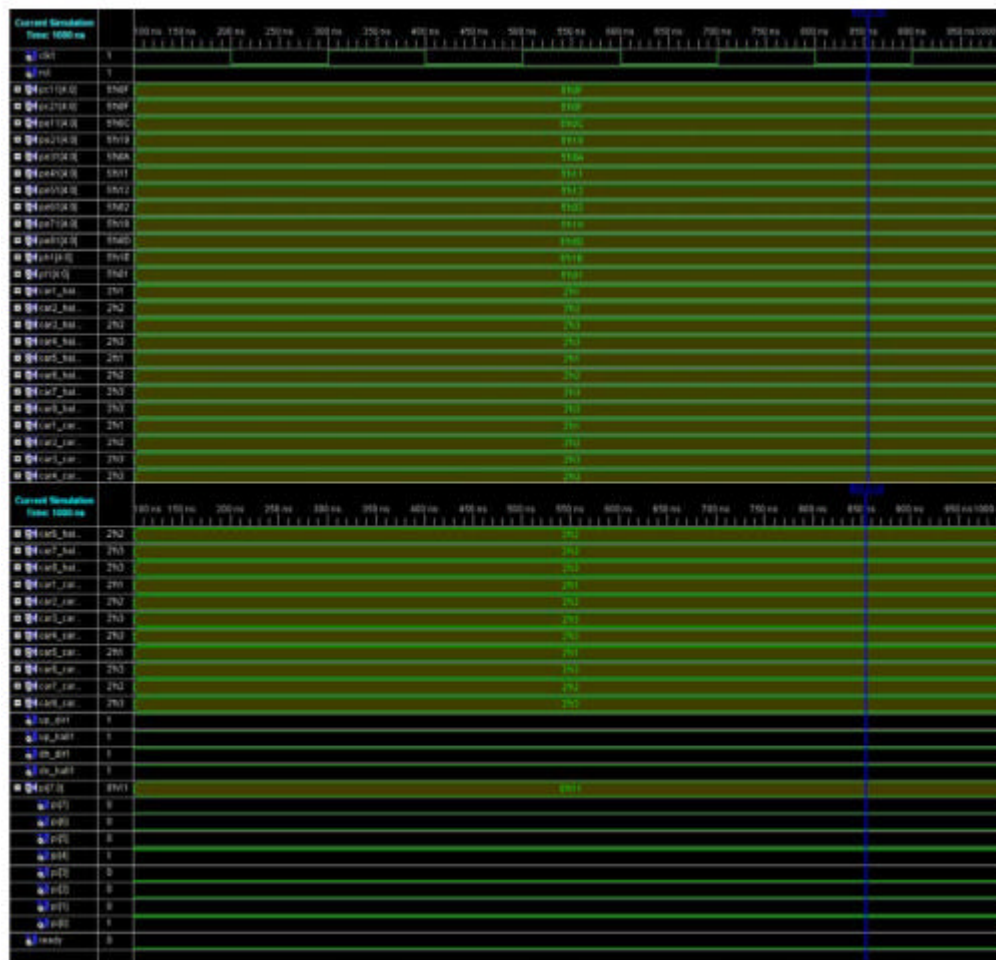


Рис. 4.2. Результат моделювання для оцінки PI

Іншим типом алгоритму є техніка логічного висновку Мамдані, яка використовує механізм логічного висновку, який запускає всі 25 правил під час кожного обчислення.

Обидва нечіткі алгоритми, що використовуються для системи керування ліфтами, реалізовані за допомогою коду VHDL. Коди VHDL синтезуються для перетворення в представлення логіки передавання реєстру (RTL) архітектури нечіткого логічного керування (FLC), яка показана на малюнку. На цьому рис. 4.3 показано вигляд RTL обох алгоритмів.

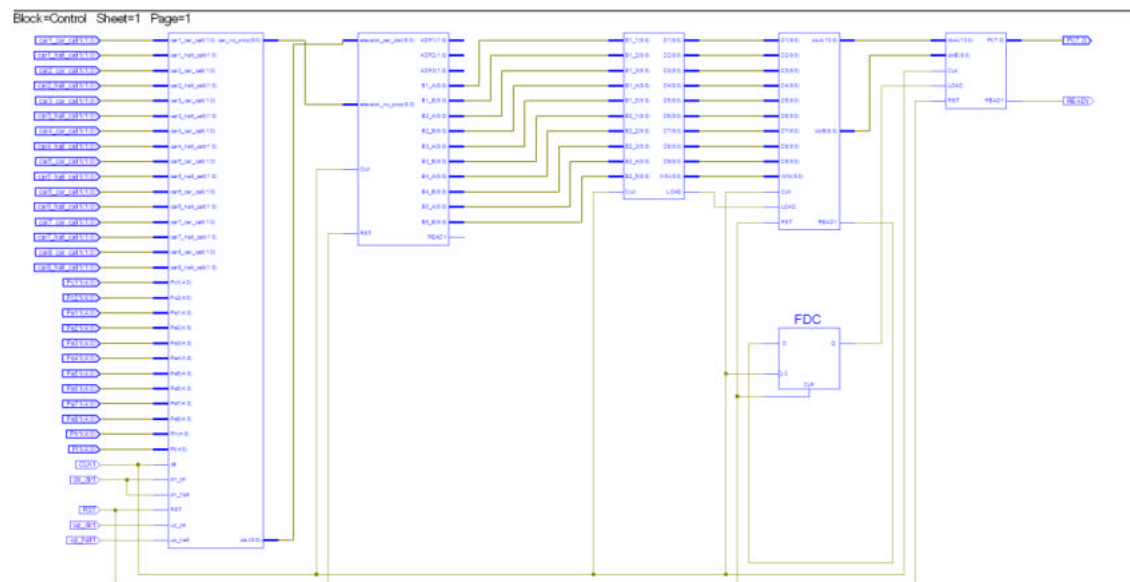
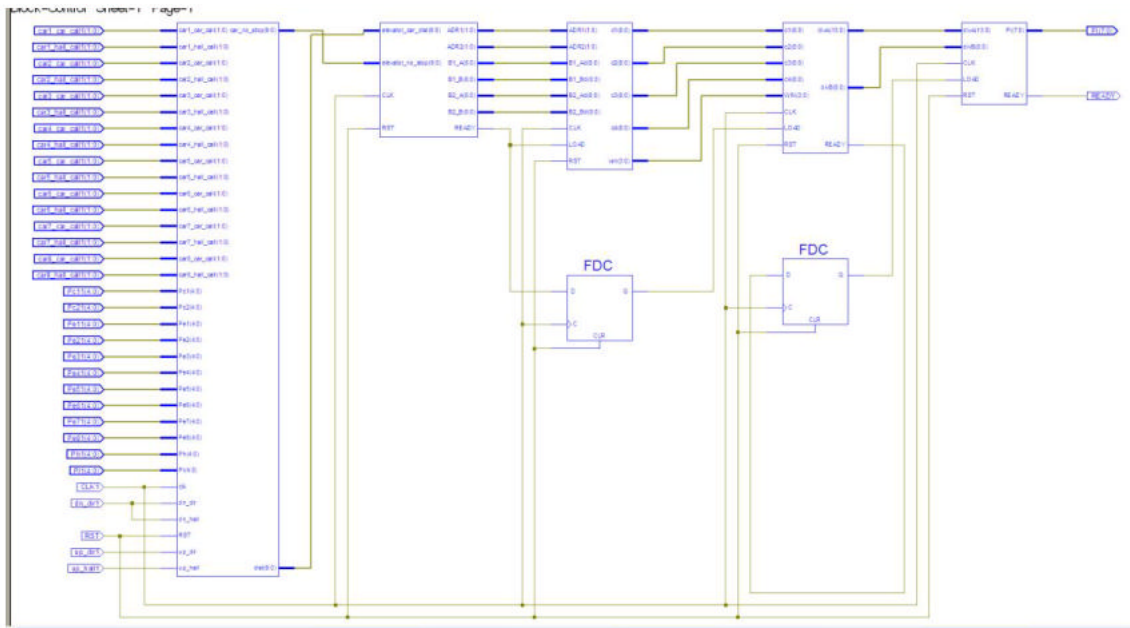


Рис. 4.3. Перегляд RTL архітектури FLC для нечіткого алгоритму висновку правила Reduce mamdani та нечіткого алгоритму висновку Simple mamdani

## 4.2. Реалізація FPGA

Обидва алгоритми системи керування ліфтом реалізовані з використанням ПЛІС Spartan 3E (XC3S500E) з пакетом FG 320 і швидкістю -

4. Результат реалізований через FPGA за допомогою ChipScope pro, як показано на рис. 4.4.

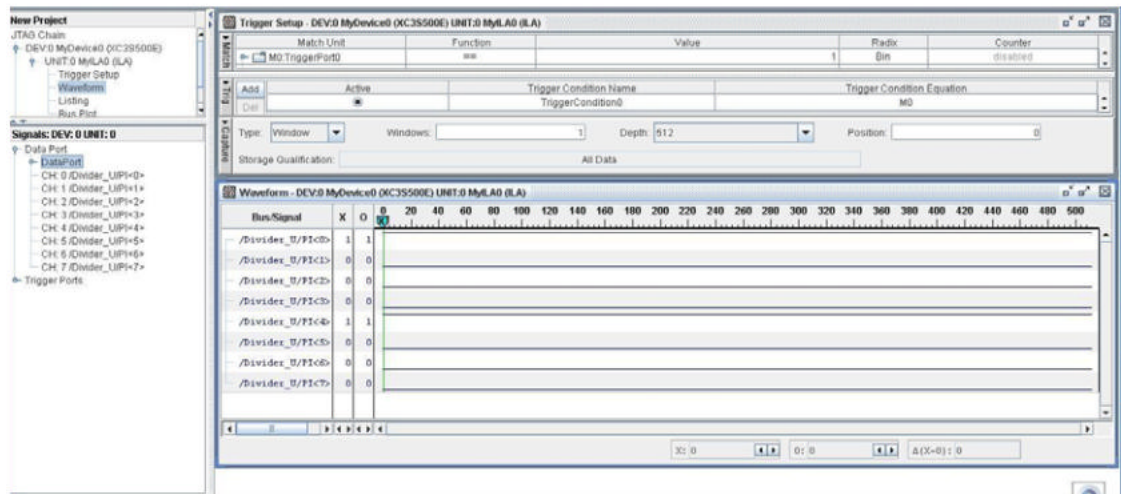


Рис. 4.4. Результат реалізації FPGA за допомогою ChipScope pro

Таблиця 4.1 показує підсумок використання пристрою для обох алгоритмів. З огляду на використання пристрою я виявив, що нечітке правило з використанням техніки скороченого правила займає менше місця в FPGA, а робоча частота вища порівняно з фактичним висновком Мамдані за нечітким алгоритмом.

### 4.3. Середній час очікування

Середній час очікування є найважливішим параметром, який використовується для вимірювання продуктивності системи керування ліфтом. Визначається як проміжок часу між натисканням кнопки виклику залу та досягненням призначеного автомобіля відповідного поверху. Тут я обчислюю середній час очікування (AWT) пасажирів для обох алгоритмів і порівнюю їх. Середній час очікування розраховується за формулою.

Таблиця 2.2 -Результати порівняння використання пристроїв для обох алгоритмів

<b>Пристрій, що використовується в FPGA</b>	<b>Зменшення правила висновку Мамдані. Нечіткий алгоритм.</b>	<b>Фактичний висновок Мамдані Нечіткий алгоритм.</b>
Кількість <u>слайсів</u>	9%	20%
Кількість <u>слайсів</u> <u>фліп флоп</u>	3%	3%
Кількість 4 входів LUT	8%	18 %
Кількість обмежених IOB	20%	20%
Кількість ГКЛК	4%	4%
Мінімальний період	17,485 <u>нс</u>	36,056 <u>нс</u>
Максимальна частота	57,192 МГц	27,735 МГц
Загальне використання пам'яті	172596 кілобайт	206964 кілобайт

$$AWT = T_{resp, car} + \sum_{i=1}^N T_{drive} + \sum_{i=1}^{N-1} T_{stop}$$

$$T_{rep, car} = T_{delay\ time\ of\ controller} \times \sum_{i=1}^M D_N \times \sum_{i=1}^M D_k \times \text{No. of car call} \times \text{No. of Hall Call}$$

$$T_{stop(i)} = T_{speed-down} + T_{get-on / off (i)} + T_{speed-up}$$

$$T_{drive} = D_i \times T_{travel-floor}$$

де  $T_{resp, car}$  - це час відгуку автомобіля.

$T_{delay\ time\ of\ controller}$  – це час виконання нечіткого контролера.

$D_N$  - відстань між поверхом виклику в залі та контролером.

$D_k$  – відстань між контролером і автомобілем.

Tdrive - поверх, де є дзвінки біля поверху.

Tstop - поверх, де призначені дзвінки з холу та виклики в автомобілі.

Ttravel-floor - це час, необхідний для подорожі між двома поверхами.

Di - це відстань, яку пройшов кожен вагон від відповідного поверху виклику в зал, коли відбувається виклик у зал.

Tspeed-down Час, протягом якого кабіна ліфта прийде в положення зупинки з максимальної швидкості.

Tget on / off (i) - тривалість посадки та виходу пасажирів з кабіни ліфта.

TSpeed-up - час, протягом якого кабіна ліфта досягає максимальної швидкості з початкового стану.

Параметр для розрахунку середнього часу очікування пасажирів:

- 1) Номер поверху 30
- 2) Час, необхідний для запуску і зупинки автомобіля, становить 3 секунди.
- 3) Час, необхідний для подорожі між двома поверхами, займає 5 секунд.
- 4) Номер кабіни ліфта 8 шт.
- 5) Висота поверху 4м.
- 6) Кількість пасажирів на поверх у 5S становить 40
- 7) Кількість зупинок, щоб дістатися до поверху виклику залу, становить 2.
- 8) Відстань кабіни ліфта від поверху виклику холу 12м.
- 9) Час посадки та виходу пасажирів в кабіні ліфта 5с.
- 10) Час виконання нечіткого контролера становить 17,485 нс для випадку-1 і 36,056 нс для випадку-2.
- 11) Загальна кількість автовикликів: 10
- 12) Загальна кількість автовикликів: 30

На основі цих даних складається середній час очікування пасажирів 30,428 секунди для техніки правила зменшення алгоритму, тоді як цей час

прийнято рівним 54,69 секунди для фактичного алгоритму висновку Мамдані. Середній час очікування розраховується на основі середнього потоку трафіку в конкретний момент часу та часу обчислення нечіткого контролера. Таблиця 4.2 показує результат порівняння середнього часу очікування обох алгоритмів.

Таблиця 4.2 - Результат порівняння середнього часу очікування обох алгоритмів

<i><b>AWT</b></i>						<i><b>час</b></i>
Середній час очікування висновок Нечіткий алгоритм.	розрахунок	використовуючи	зменшити	правильно	<u>Мамдані</u>	30,428 <u>сек</u>
Розрахунок середнього часу очікування з використанням фактичних висновків <u>Мамдані</u> Нечіткий алгоритм.						54.69 <u>Розд</u>

## 5. СПЕЦІАЛЬНА ЧАСТИНА

### 5.1. Реалізація алгоритму для системи керування ліфтом

Очікування кабіни ліфта є важливою ситуацією, з якою ми всі знайомі. Коли ми натискаємо кнопку та чекаємо ліфта, нам може знадобитися довго чекати, якщо в конкретній ситуації занадто багато пасажирів або недостатньо кабіни ліфта. Важливим є те, скільки часу ми чекаємо, що залежить від стратегії диспетчеризації кабіни ліфта в конкретний момент часу. Ефективність кількох ліфтів, встановлених в офісній будівлі, може підвищитися, якщо центральний диспетчер використовується для групування пасажирів, які прямують на той самий поверх до одного ліфта. На малюнку 3.1 показано стратегію диспетчеризації чотирьох ліфтів. Тут кожен ліфт має позицію, напрямок і швидкість. Він також має набір кнопок, які вказують, де пасажир хочуть вийти. Алгоритм диспетчеризації ліфта, як правило, розрахований насамперед на різні періоди, тобто на ранкові та вечірні години пік.

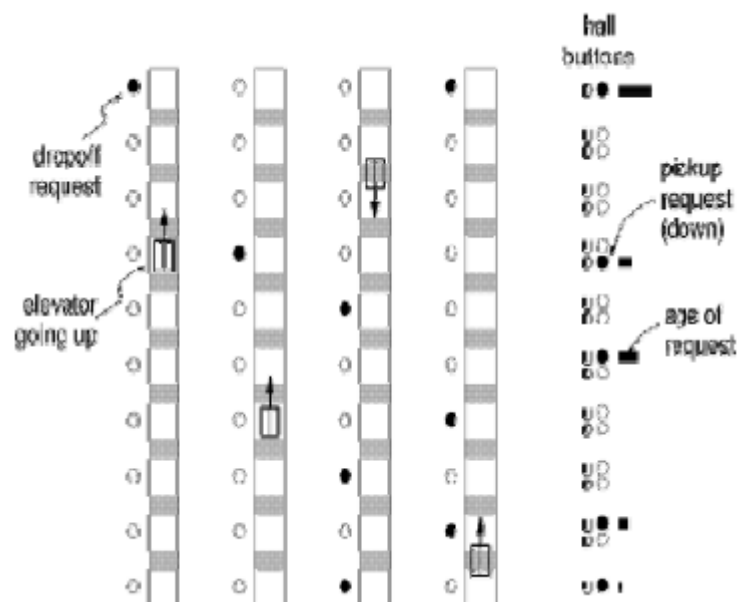


Рис. 5.1. Стратегія диспетчеризації чотирьох ліфтів

Ефективність алгоритмів диспетчеризації ліфтів вимірюється декількома способами, усі з огляду на кількість пасажирів, які входять до системи. Середній час очікування - це те, скільки пасажир чекає, перш ніж сісти в ліфт середній системний час, потрібний пасажирові, щоб досягти місця призначення. Ще одна зустрічна статистика пасажирів – це той, чий час очікування перевищує 60 секунд.

Найголовніше, щоб кожен ліфт сформулював певну дію, яка базується на алгоритмі диспетчеризації. По-перше, кожен ліфт приймав власні рішення незалежно, а по-друге, кількість обмежень була накладена на рішення. Ліфт, який перевозить пасажирів, не міг пройти повз поверх, якщо будь-який пасажир хотів піднятися з підлоги, якщо будь-який пасажир хотів зійти зі свого відповідного поверху, а також не міг змінити напрямок, доки всі пасажирів, які бажають йти в поточному напрямку до свого відповідний поверх.

#### Апаратне забезпечення алгоритмів керування

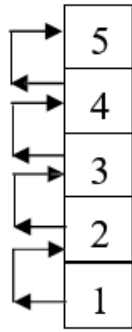
Тут розглядаються шість типів алгоритмів диспетчеризації. Залежно від умов руху система управління ліфтом може вибрати один алгоритм із шести і працювати за цим алгоритмом деякий час, поки ситуація руху знову не зміниться. Це сприятиме зменшенню середнього часу очікування (AWT) пасажирів, а також зменшить споживання електроенергії ліфтовою системою. На рис. 5.2 показано шість типів алгоритмів диспетчеризації.

#### Колективний алгоритм (CU)

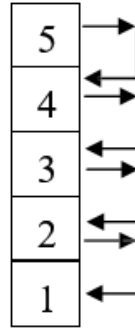
Алгоритм колективного підйому є одним із важливих аспектів системи управління ліфтом. Використовується в основному, коли дорожня обстановка дуже низька. У цьому випадку використовується колективний принцип, коли кабіна ліфта рухається вгору. Автомобіль зупиняється на кожному поверсі на основі викликів у залі та виклику автомобіля під час руху вгору. Це робиться шляхом прийняття найближчого виклику в зал замість того порядку, в якому були зроблені виклики. На малюнку показано послідовність того, як кабіна



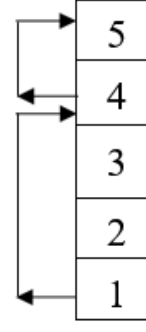
ліфта може відвідувати поверхи під час подорожі на двадцять поверхів будівлі.



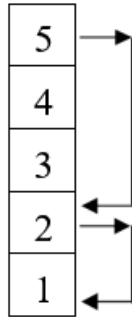
Алгоритм Collective-UP



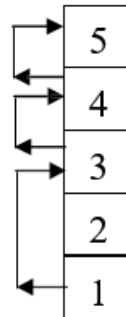
Алгоритм колективного  
зніження



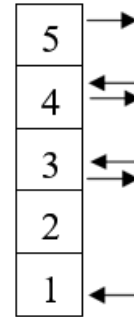
Алгоритм Selective-UP



Алгоритм селективного  
зніження



Вибірковий /  
Колективний  
алгоритм -UP



Алгоритм  
колективного/вибі  
ркового зніження

Рис. 5.2. Шість алгоритмів керування

#### Алгоритм колективного зніження (CD)

Тут ми використовуємо колективний принцип, коли кабіна ліфта рухається вниз. Автомобіль зупиняється на кожному поверсі на основі викликів у холі та виклику автомобіля під час руху вниз. Це робиться шляхом прийняття найближчого виклику в зал замість того порядку, в якому були зроблені виклики. На малюнку показано послідовність того, як кабіна ліфта може відвідувати поверхи під час спуску вниз на двадцятиповерхову будівлю.

### Алгоритм вибіркового зниження (SD)

Алгоритм вибору вниз використовується в режимі Down Peak, кабіни ліфтів у групі відправляються з вестибюля на найвищий обслуговуваний поверх, після чого вони починають бігати поверхами у відповідь на виклики холу, зроблені пасажиром, які бажають покинути будівлю. Це дозволяє ліфтовій системі забезпечити максимальну пропускну здатність людей, які залишають будівлю. Початок алгоритму вибору вимкнення може бути ініційований годинником, прибуттям певної кількості повністю завантажених автомобілів у вестибюль протягом певного періоду часу або перемикачем, яким вручну керує черговий.

### Вибірковий алгоритм (SU)

Алгоритм вибіркового підйому використовується в режимі Up Peak, кабіни ліфтів у групі викликаються до вестибюлю, щоб забезпечити швидке обслуговування пасажирів, які прибувають до будівлі, найчастіше вранці, коли люди прибувають на роботу, або після обіду. період. Ліфти відправляються один за одним, коли вони досягають заздалегідь визначеної кількості пасажирів або коли їх двері відчинені протягом певного періоду часу. Наступний ліфт, який відправляється, зазвичай має підсвічений ліхтар у передпокої або табличку «ця машина відправляється наступною», щоб заохотити пасажирів максимально використовувати доступну пропускну здатність ліфтової системи. .ПочатокВибірковий алгоритмможе бути викликаний годинником, від'їздом певної кількості повністю завантажених автомобілів, що залишають вестибюль протягом певного періоду часу, або перемикачем, яким керує вручну черговий.

### Вибірково-колективний алгоритм підвищення (SCU)

Вибірково-колективний підйом є найважливішим алгоритмом у багатоповерховому будинку, особливо вранці, коли всі пасажирі хочуть перейти на інший поверх будівлі з першого поверху. Цей алгоритм спочатку працює у вибіркового режимі, тобто ліфт спочатку вибірково рухається

вгору та зупиняється на певному поверсі. Потім він переходить у колективний режим, досягаючи верхнього поверху із зупинкою на кожному поверсі.

### Колективний – вибіркового алгоритм зниження (CSD)

Колективне вибіркового спуск є найважливішим алгоритмом у багатоповерхових будівлях, особливо у вечірній час, коли всі пасажери хочуть прийти на перший поверх з іншого поверху будівлі. Цей алгоритм спочатку працює в колективному режимі, тобто ліфт спочатку рухається вниз. разом і певною мірою зупиняється на всіх поверхах. Потім він переходить у вибіркового режим, переходячи безпосередньо на перший поверх, не зупиняючись між поверхами, навіть якщо натиснути холл і виклик автомобіля.

Рис. 5.3 показує результат моделювання для алгоритму колективного підняття. На малюнку показано, що на основі виклику холу кабіна ліфта може відвідувати відповідний поверх колективно, тобто один за одним.

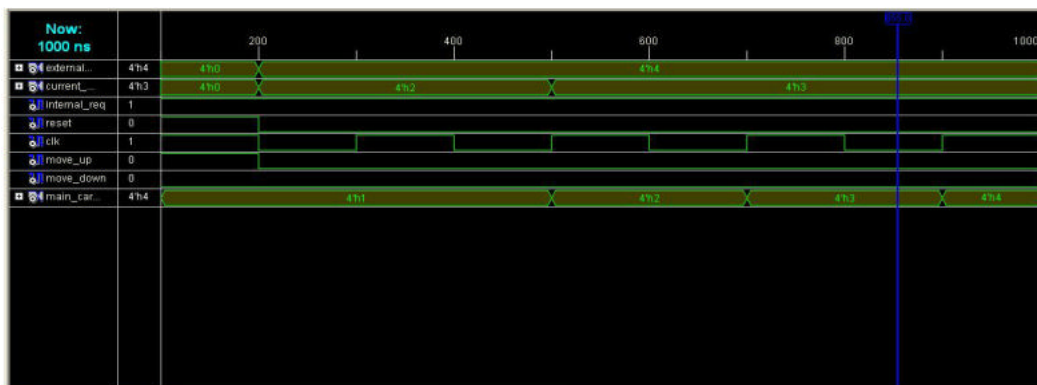


Рис. 5.3. Результат моделювання алгоритму колективного вгору

Рис. 5.4 показує результат моделювання алгоритмів колективного зниження. З малюнка видно, що на підставі виклику холу кабіна ліфта групує відвідує відповідний поверх.

На рис. 5.4 показано, що виклик у зал надходить із поверху «3» та поверху «2», тому що кабіна ліфта безпосередньо відвідує поверх 3 та поверх

2 з поверху 5. Вона пропускає поверх 4, оскільки там немає дзвінок у зал, що надходить з поверху 4. Після цього він надходить на перший поверх.

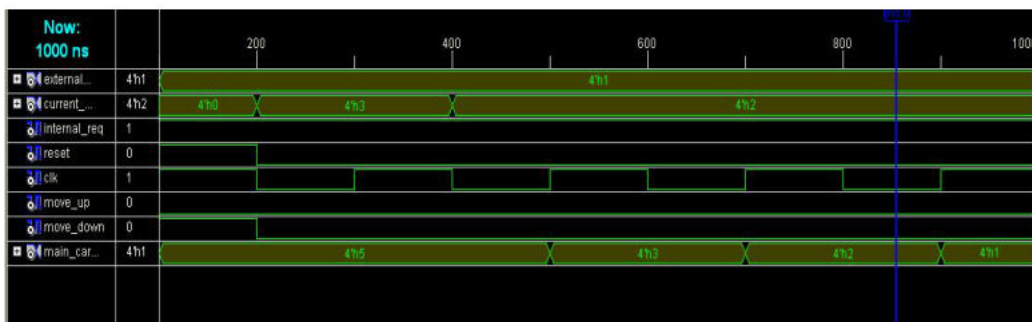


Рис. 5.4 Результат моделювання алгоритму колективного зниження

Рис. 5.5 показує результат моделювання алгоритмів вибіркового зниження. З малюнка видно, що виклик у хол і виклик автомобіля надходять з поверхів номер 2, 3 і 4. На основі алгоритмів кабіна ліфта відвідує найнижчий поверх виклику холу, який є поверхом номер 2, після чого він потрапляє на землю. поверх.

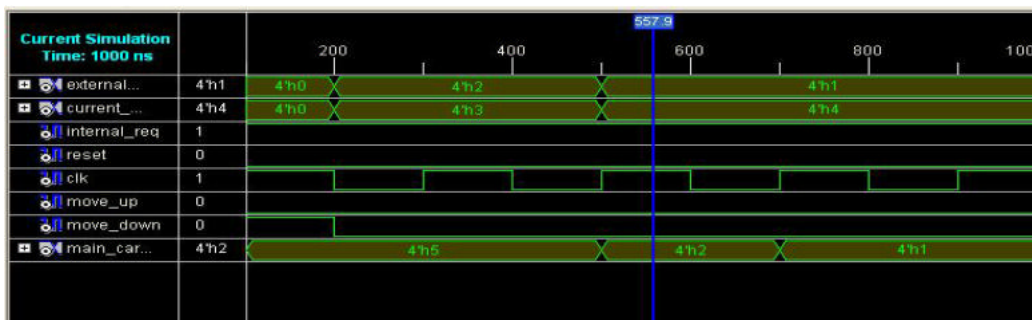


Рис. 5.5 Результат моделювання алгоритму вибіркового зниження

Рис. 5.6 показує результат моделювання алгоритмів вибіркового підвищення. З рисунку видно, що виклик холу та виклик автомобіля надходять з поверхів № 2, 3 та 4.

Згідно з алгоритмами, кабіна ліфта проїжджає на найвищий захід – поверх № 4, після чого – на поверх № 5.

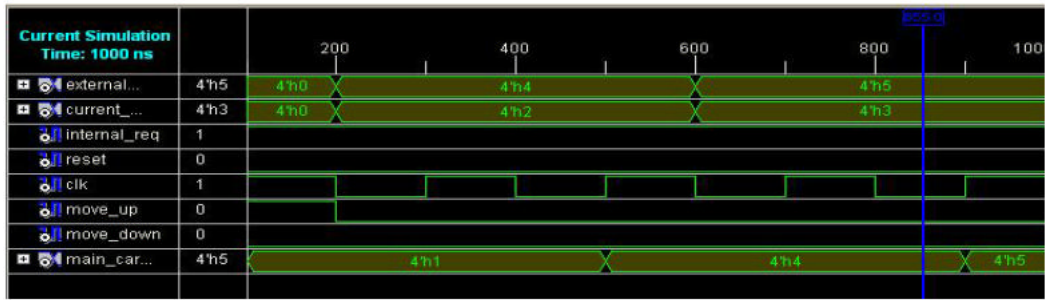


Рис. 5.6. Результат моделювання алгоритму вибіркового підвищення

На рис. 5.7 показано результати моделювання алгоритмів вибірково-колективного підвищення. З малюнка видно, що виклик холу і виклик автомобіля надходять з поверхів номер 1, 2, 3 і 4. У відповідь на це кабіна ліфта рухається вибірково, тобто з поверху номер '1' на поверх номер '3'. Після цього він разом переміщується на поверхи під номерами «4» і «5».

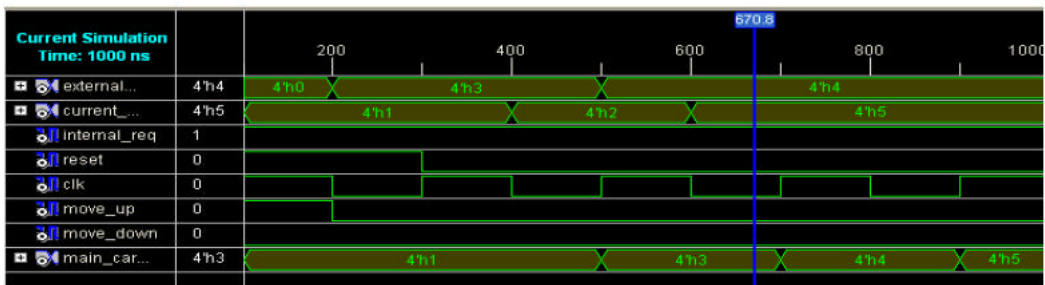


Рис. 5.7 Результат моделювання алгоритму селективно-колективного підбору

Рис. 5.8 показує результат моделювання алгоритмів колективно-вибіркового зниження. З малюнка видно, що виклики холу та виклики автомобіля надходять з поверхів номер 1, 2, 3 та 4. У відповідь на це кабіна ліфта заходить колективно, щоб відвідати поверхи номер 4 та 5, після чого рухається вибірково. на поверх №1.



Рисунок 5.8 Результат моделювання алгоритму колективного вибіркового зниження

Таблиця 5.1 - Результати порівняння використання пристрою для всіх шести типів алгоритмів диспетчеризації.

Пристрій використовує FPGA	Колективний Вгору	Колективний Даун	Вибірковий вгору	Вибірковий вниз	Колективний / Вибірковий UP		Колективний / Вибірковий Вниз	
					Вгору	Вниз	Вгору	Вниз
Кількість шматочків	30 з 4656 1%	29 з 4656 1%	13 з 4656 1%	13 з 4656 1%	25 з 4656 1%	25 з 4656 1%	25 з 4656 1%	25 з 4656 1%
Кількість шматочків Flip flop	11 з 9312 1%	11 з 9312 1%	9 з 9312 1%	10 з 9312 1%	10 з 9312 1%	11 з 9312 1%	11 з 9312 1%	11 з 9312 1%
Кількість чотирьох вхідних LUT	60 з 9312 1%	58 з 9312 1%	24 з 9312 1%	24 з 9312 1%	46 з 9312 1%	47 з 9312 1%	46 з 9312 1%	47 з 9312 1%
Кількість IO	17	17	17	17	17	17	17	17
Кількість обмежених IOB	15 з 232 6%	15 з 232 6%	15 з 232 6%	15 з 232 6%	15 з 232 6%	15 з 232 6%	15 з 232 6%	15 з 232 6%
Кількість ГКЛК	1 із 24 4%	1 із 24 4%	1 із 24 4%	1 із 24 4%	1 із 24 4%	1 із 24 4%	1 із 24 4%	1 із 24 4%
Мінімальний період	6,683 нс	6,569 нс	4,512 нс	4,653 нс	5,987 нс	5,934 нс	5,987 нс	5,934 нс
Максимум Частота	149,633 МГц	145,583 МГц	221,631 МГц	222,835 МГц	167,029 МГц	168,934 МГц	167,029 МГц	168,934 МГц
Загальне використання пам'яті	160308 кілобайт	160308 кілобайт	160308 кілобайт	160308 кілобайт	160308 кілобайт	160308 кілобайт	160308 кілобайт	160308 кілобайт
потужність	0,08313 В	0,08314 В	0,08313 В	0,08313 В	0,08351 В	0,08405 В	0,08351 В	0,08405 В

## **6 БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ, ОХОРОНА ПРАЦІ**

### **6.1. Організація охорони праці при роботі з системою управління**

Охорона праці розглядає проблеми забезпечення здорових і безпечних умов праці. Виявляє і вивчає можливі причини нещасних випадків, професійних захворювань, аварій, вибухів, пожеж і розробляє систему заходів і вимог з метою виключення цих причин і створення безпечних і сприятливих для людини умов праці.

Завдання охорони праці є зведення до мінімуму імовірності пошкодження або захворювання працівників з одночасним забезпеченням комфорту при максимальній продуктивності праці.

Навчання працівників безпеці праці проводять відповідно до вимог ГОСТ 12. 0.004 - 79, який встановлює порядок і види навчання. На всіх підприємствах і в організаціях незалежно від характеру і ступеню небезпеки виробництва навчання працівників проводять при підготовці нових робітників, проведенні різноманітних видів інструктажів і підвищенні кваліфікації.

Контроль за своєчасним і якісним навчанням виконує відділ охорони праці чи інженер з охорони праці, або ІТП, на якого наказом керівника підприємства покладено ці обов'язки. Ті, що вперше поступають на роботу, навчання проходять згідно з "Типовим положенням про підготовку і підвищення кваліфікації робітників". В журналі обліку навчальної роботи реєструють навчальну тему, за якою проводилось навчання.

Інструктаж працюючих поділяють на вступний, початковий, на робочому місці, повторний, позаплановий і початковий.

Вступний інструктаж з усіма, хто поступає на роботу незалежно від їх освіти і стажу роботи по даній професії, проводить інженер з охорони праці за програмою, затвердженою головним інженером підприємства, про

проведення вступного інструктажу з обов'язковим підписом того, хто проводив інструктаж і того, хто його отримував.

Початковий інструктаж на робочому місці, повторний, позаплановий і поточний проводить керівник робіт.

Початковий інструктаж на робочому місці проводять при прийомі на роботу нових робітників за інструкцією з охорони праці, розробленою для окремих професій або видів робіт. Всі робітники після цього інструктажу і перевірки знань 2-5 змін (залежно від навичок і стажу роботи) працюють під наглядом бригадира чи майстра, потім оформляється допуск до їх самостійної праці.

Повторний інструктаж проходять всі працівники незалежно від кваліфікації, освіти і стажу роботи через три місяці. Його проводять з метою перевірки знання робітниками правил і норм з охорони праці.

Позаплановий інструктаж проводять коли змінилися правила охорони праці або технологічний процес, обладнання, інструмент та інші фактори, що впливають на безпеку праці; коли працівники порушують правила охорони праці, що можуть призвести чи призвели до травм, аварій чи пожежі, вибуху. Його проводять індивідуально чи з групою робітників однієї професії за програмою початкового інструктажу на робочому місці. При його реєстрації вказують причину, яка спричинила його проведення.

Умови праці мають велике значення практично для всіх виробничих показників - продуктивності праці, якості робіт, безпеки працівників та інше.

Санітарно-гігієнічні умови праці характеризуються показниками виробничого середовища - рівнем освітлення, мікрокліматичними параметрами, загазованістю і запиленістю повітряного середовища, рівнем шуму і вібрації, наявністю іонізуючого випромінювання та інше.



## 6.2. Електробезпека

Електричні установки, з якими доводиться мати справу практично всім працюючим по встановленню та налагодженню засобів автоматизації, виявляють для людини велику потенційну небезпеку, яка збільшується у зв'язку з тим, що органи чуття людини не можуть на відстані виявити присутність електричної напруги на обладнанні.

Степінь ураження електричним струмом залежить від цілого ряду факторів: значення сили струму, електричного опору тіла людини та тривалості протікання через неї струму, виду та частоти струму, індивідуальних властивостей людини та умов навколишнього середовища.

Конструкція електроустановок має відповідати умовам їх експлуатації та забезпечувати захист персоналу від дотику з струмоведучими та рухомими частинами, а обладнання - від попадання всередину посторонніх твердих тіл та води.

Конструкція, вид виконання, спосіб встановлення, клас ізоляції застосовуваних провідників, кабелів, пристроїв та іншого електрообладнання відповідають вимогам електробезпеки. За ступенем ураження людей електричним струмом котельня відноситься згідно ПУЕ 1.1.13 до категорії приміщень з підвищеною небезпекою (висока температура, можливість одночасного дотику до металевих елементів технологічного обладнання або металоконструкцій будинку та металевих корпусів електрообладнання).

У нормальному режимі роботи обладнання - можливість ураження працівників електричним струмом виключена. Але на випадок аварії для запобігання ураження струмом людей передбачене захисне заземлення. Згідно ПУЕ 1.7.65 допустимий опір заземлення повинен бути не більшим 10 Ом.

При виконанні монтажних робіт використовуються переносні електроінструменти (електродрилі, електрошліфувальні установки, тощо). Для забезпечення безпечної праці корпуси однофазних електроприймачів повинні занулюватись.

Захист людини від ураження електричним струмом в мережах з зануленням здійснюється тим, що при замиканні одної з фаз на занулений корпус в ланці цієї фази виникає струм короткого замикання, що діє на струмовий захист (плавкий запобіжник, автомат), в результаті чого відбувається відключення аварійної ділянки від мережі. Крім того, ще до спрацювання захисту струм короткого викликає перерозподіл напруги в мережі, що приводить до зниження напруги корпусу відносно землі. Таким чином, занулення зменшує напругу дотику та обмежує час, на протязі якого людина, що доторкнулася до корпусу, може потрапити під дію напруги.

Для того, щоб забезпечити швидке (на протязі декількох секунд) відключення аварійної ділянки, струм короткого замикання повинен бути достатньо великим. Відповідно до вимог ПУЕ струм короткого замикання повинен не менше ніж в три рази перевищувати номінальний струм плавкої вставки найближчого запобіжника або номінальний струм нерегульованого розчеплювача автоматичного вимикача. При використанні автоматичних вимикачів, що мають тільки електромагнітний розчіплювач (відсічку), струм короткого замикання повинен перевищувати значення струму встановлення миттєвого спрацювання в 1,25-1,4 рази в залежності від номінального струму.

В однофазних електроприймачів, що включені між фазним та нульовим робочим проводами, занулення корпусів слід виконувати з допомогою окремого (третього) провідника, який повинен з'єднувати корпус електроприймача з нульовим захисним проводом. В таких випадках під'єднувати корпуси електроприймачів для забезпечення електробезпеки до нульового робочого проводу недопустимо, оскільки при його розриві

(перегоранні запобіжника) всі під'єднані до нього корпуси виявляться під фазною напругою відносно землі.

В мережі з зануленням недопустимо використовувати заземлення окремих електроприймачів, не під'єднавши їх перед цим до нульового захисного провідника. В цьому випадку при замиканні фази на заземлений, але не приєднаний до нульового захисного провідника корпус створюється коло струму через заземлення цього корпусу та заземлення нейтралі джерела струму. Такий випадок небезпечний, оскільки засоби захисту не зможуть відключити такий електроприймач через мале значення струму і тому небезпечна напруга на всіх корпусах може зберігатися тривалий період, поки заземлений приймач не буде відключений вручну.

Важливо відмітити, що якщо занулений корпус одночасно заземлений, то це тільки покращує умови безпеки, оскільки забезпечує додаткове заземлення нульового захисного проводу.

Для ізоляції людини від частин електроустановок, що знаходяться під напругою, використовуються основні та допоміжні ізолюючі засоби, а саме слюсарно-монтажний інструмент з ізольованими ручками, коврики, ізолюючі підставки, тощо.

У приміщеннях, де знаходяться вимірювальні прилади, необхідно забезпечити виконання заходів по боротьбі з статичною електрикою (тобто прилади повинні бути заземлені). Найпростішим засобом є підтримка відносної вологості повітря на рівні 50 - 60 % за допомогою побутового електрозволожувача.

Підлогу слід виконувати відповідно до ГОСТ 12.4.124-83, використовуючи антистатичне покриття на проходах і біля робочих місць.

Робітникам рекомендовано носити одягу з природних матеріалів або з комбінованих - природних і штучних волокон. Для зняття електростатичних зарядів з одяжі слід використовувати антистатика побутового призначення.

Оскільки корпуси приладів виконані з металу, то для усунення небезпеки ураження людини електричним струмом (можливий пробій на корпус приладу) використовується захисне заземлення.

### 6.3. Розрахунок заземлення

Розрахуємо систему заземлення для електроустаткування, яке працює від напруги 220 В.

$$R_{\text{за}} \leq \frac{U}{I_f} = \frac{220}{66} = 3.3 \leq 4 \text{ Ом}$$

Визначаємо опір ґрунту:  $\rho = k_n * \rho_n = 2 * 200 = 400 \text{ Ом м}$ ,

де  $k_n$  - коефіцієнт підсилення;

$\rho_n$  — питомий опір ґрунту (вибирається з довідкової літератури).

Визначаємо опір одиночного вертикального заземлювача:

$$R_B = \frac{\rho}{2\pi} \left( \ln \frac{2l}{d} + \frac{1}{2} * \frac{4t+1}{4t-1} \right)$$

де  $t$  - відстань від середини заземлювача до поверхні ґрунту, м;

$l, d$  - довжина і діаметр стержня заземлювача, м;

$$R_B = 96 \text{ Ом.}$$

Визначаємо опір сталевій полосі, що з'єднує стержневі заземлювачі:

$$R_{II} = (\rho / 2\pi) * \ln(l^2 / dt) = 61 \text{ Ом.}$$

Визначаємо орієнтовне число стержневих заземлювачів:

$$n = R_B / [r_B] \eta_B = 96 / 4 * 1 = 24 \text{ шт.};$$

$r_B$  - допустимий по нормам опір заземляючого пристрою,

$\eta_B$  - коефіцієнт використання вертикальних заземлювачів (для орієнтовного розрахунку приймається рівним 1).

Приймаємо розміщення вертикальних заземлювачів по контуру з відстанню між сталевими заземлювачами рівним 2l. З довідкової літератури визначаємо  $\eta_B = 0,66$  і  $\eta_r = 0,39$ .

Визначаємо необхідну кількість вертикальних заземлювачів

$$n = R_B / [r_B] \eta_B = 96 / (4 * 0,66) = 36$$

Розраховуємо загальний розрахунковий опір аземлюючого пристрою R з врахуванням з'єднувальної полоси

$$R = R_B R_H / (R_B \eta_r + R_H \eta_B n) = 3,9 \text{ Ом.}$$

Розрахунок проведено правильно, оскільки виконується умова  $R \leq [r_B]$ .

### Розрахунок штучного заземлення:

Приймаємо, що опір захисного заземлення не повинен перевищувати 4 Ом:

$$R_{33} = \frac{R_c R_n}{R_c + R_n} \leq 4 \text{ Ом}$$

де  $R_{33}$  – опір захисного заземлення;

$R_c$  – опір стержневих заземлювачів;

$R_n$  – опір поперечних заземлювачів.

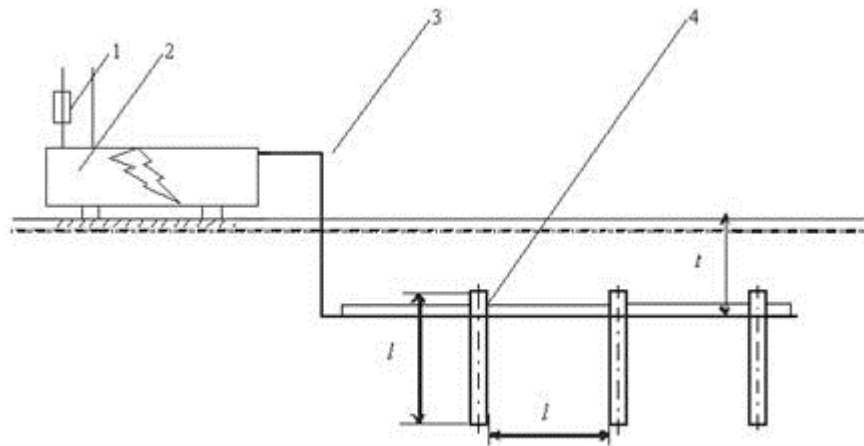


Рисунок 6.1 - Пристрій заземлення

4 – плавка вставка; 2 – електроустановка; 3 – з'єднувальна штаба; 4 – трубчатий заземлювач

Опір одиночного стержневого заземлювача розтіканню електричного струму:

$$R_{oc} = \frac{\rho_r}{2\pi l} \left( \ln \frac{2l}{d} + \ln \frac{4h' + l}{4h' - l} \right)$$

де  $h$  – відстань від поверхні ґрунту до заземлювача і становить 0,8 м;

$l$  – довжина стержневого заземлювача 3 м;

$d$  – діаметр стержневого заземлювача 50 мм.

$$R_{oc} = \frac{750}{2 \cdot 3,14 \cdot 3} \left( \ln \frac{2 \cdot 3}{0,05} + \ln \frac{4 \cdot 0,8 + 3}{4 \cdot 0,8 - 3} \right) = 39,8 \cdot (0,18 + 3,43) = 143,8 \text{ Ом}$$

Опір одиночного поперечного заземлювача:

$$R_{on} = \frac{\rho_r}{2\pi} \ln \frac{2l^2}{bh'}$$

де  $l$  – довжина поперечного заземлювача 2,5 м;

$b$  – ширина полоси заземлювача 30 мм;

$\rho_r$  – розрахунковий опір ґрунту: для поперечних електродів 1000 Ом·м, для стержневих електродів 750 Ом·м.

$$R_{on} = \frac{1000}{2 \cdot 3,14 \cdot 2,5} \ln \frac{2 \cdot 2,5^2}{0,03 \cdot 0,8} = 63,7 \cdot 6,25 = 398,1 \text{ Ом}$$

В наслідок взаємовпливу вводимо коефіцієнт використання заземлювачів:

$$\eta = \frac{R_0}{nR_p}$$

де  $R_d$  – допустимий опір заземлення, що становить 4 Ом;

$R_0$  – опір одиночного заземлювача.

З цієї формули методом ітерацій підбирають  $n$ , при якому  $\eta = 1$ :

<b>n</b>	<b>R<sub>n</sub></b>	<b>R<sub>c</sub></b>	<b>R<sub>o</sub></b>	<b>η</b>
1	398,1	143,8	105,6	26,1
5	398,1	143,8	105,6	5,2
10	398,1	143,8	105,6	2,6
15	398,1	143,8	105,6	1,7
20	398,1	143,8	105,6	1,3
25	398,1	143,8	105,6	1,1
26	398,1	143,8	105,6	1,0
27	398,1	143,8	105,6	0,9

Отже приймаємо кількість одиночних заземлюючих електродів рівною

26.

## ОСНОВНІ ВИСНОВКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

У роботі було розроблено автоматизовану систему керування групою ліфтів для забезпечення мінімального часу очікування та забезпечення найшвидшої роботи системи

Було використано реалізацію контролера нечіткої логіки для системи управління групою ліфтів з використанням VHDL з двома різними нечіткими алгоритмами. Реалізація контролера нечіткої логіки є дуже простою шляхом кодування кожного компонента системи нечіткого логічного висновку відповідно до специфікацій проекту. Дизайн системи є дуже гнучким, оскільки функції членства та базу правил можна легко змінити за допомогою скорочених методів правил. Завдяки зменшеним правилам, час обчислення нечіткого контролера зменшується, а система керування групою ліфтів забезпечує більш швидку роботу за рахунок зменшення середнього часу очікування пасажирів до 30,428 секунди порівняно з реальним алгоритмом.

Було реалізовано шість типів алгоритмів диспетчеризації для ліфтової системи. Система керування ліфтом використовує цей алгоритм у різних ситуаціях для безперебійної роботи та зменшує середній час очікування пасажирів та споживання електроенергії ліфтовою системою. Поведінка ліфтової системи покращується шляхом вибору одного алгоритму з п'яти на основі можливої ситуації.



**ПЕРЕЛІК ПОСИЛАНЬ**

1. J. Jamaludin, N.A. Rahim, W.P.Hew, “Development of self tuning fuzzy logic controller for intelligence control of elevator system,” *Journal of Engineering Applications of Artificial Intelligence* ELSEVIER 22, 2009. p. 1167–1178.
2. Daniel M. Munoz, Carlos H. Llanos, Mauricio Ayala-Rincon, “Distributed Approach to group control of elevator systems using fuzzy logic and FPGA implementation of dispatching algorithms,” *Journal of Engineering Applications of Artificial Intelligence* ELSEVIER 21, 2008. p.1309–1320.
3. Akos Becker, Department of Electronics Technology, “Microcontroller based elevator controlling system,” Budapest University of Technology and Economics, Budapest, Hungary , IEEE conference, Jan. 2007. p. 1216-1231.
4. Zhang Yajun, Chen Long, Fan Lingyan, “A Design of Elevator Positioning Control System Model,” *IEEE int.Conference Neural Networks & Signal Processing*, Zhenjiang, China, IEEE conference, Jun. 2008. p. 836-841.
5. Fredrick Cady, *Software and Hardware Engineering. Assembly and C Programming for the Freescale HCS-12 microcontroller*, 2008. 308 p.
6. Daijin Kim, member IEEE “An Implementation of fuzzy Logic Controller on the Reconfigurable FPGA system,” *IEEE Transactions on industrial Electronics*, Vol.47, No.3, June 2000. p. 365-372.
7. Gudwin R., Gomide F., Andrade Netto M. A fuzzy elevator group controller with linear context adaptation. In: *Proceedings of FUZZ-IEEE98, WCCI’98—IEEE World Congress on Computational Intelligence*, Anchorage. IEEE, Alaska, USA, 1998 p. 481–486.
8. Daniel M. Munoz Carlos H. Lianos, Mauricio Ayala-Rincon Rudi van Els, Renato P. Almeida. *Implementation of Dispatching Algorithms for Elevator Systems using Reconfigurable Architectures*. IEEE Conference 2007 February. p. 365-371.
9. Xilinx ISE 10.1 Software manuals. URL : [www.xilinx.com](http://www.xilinx.com).

10. Crites, R.H., Barto, A.G, "Improving elevator performance using reinforcement learning," In : Touretzky, D. (Ed.), *Advances in Neural Information Processing Systems*, vol. 8. MIT Press, Cambridge, MA, 1996. p. 1017–1023.
11. Crites, R.H., Barto, A.G., Elevator group control using multiple reinforcement learning agents. *Machine Learning*. 33 (2–3), 1992. p. 35–262.
12. Fujino, A., Tobita, T., Segawa, K., Yoneda, K., Togawa, A., "An elevator group control system with floor-attribute control method and system optimization using genetic algorithms," *IEEE Transactions on Industrial Electronics* Jan vol 44 (4), 1997. p. 546–552.
13. Gudwin R., Gomide F., Andrade Netto M. A fuzzy elevator group controller with linear context adaptation. In: *Proceedings of FUZZ – IEEE 98, WCCI' 98 - IEEE World Congress on Computational Intelligence*, Anchorage. IEEE, Alaska, USA, 1998. p. 481–486.
14. Homaifar, A., McCormick, E., "Simultaneous design of membership functions and rule sets for fuzzy controllers using genetic algorithms". *IEEE Transactions on Fuzzy Sets* jun 3 (2), 1995. p. 129–139.
15. Huang, S.J., Lee, J.S., "A stable self-organizing fuzzy controller for robotic motion control". *IEEE Transactions on Industrial Electronics* 47 (2), 2000. p. 421–428.
16. Imasaki N., Kubo S., Nakai S., Yoshitsugu T., Jun-Ichi K., Endo T. Elevator group control system tuned by a fuzzy neural network applied method. In: *Proceedings of the IEEE International Conference on Fuzzy Systems*. IEEE, Yokohama, Japan, 1995. p. 1735–1740.
17. Imrak C.E., Barney G.C. Applications of neural networks on traffic control". In: *Proceedings of ELEVCON*. IAEE Publications, Zurich, Switzerland, 1998. p. 140–148.
18. Kim C. B., Seong K. A., Kwang H. L., Kim J. O. Design and implementation of a fuzzy elevator group control system. *IEEE Transactions on*

Systems, Man, and Cybernetics - Part A: Systems and Humans 28 (3), 1998. p. 277–287.

19. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 1. [навчальний посібник] (Лист МОНУ №1/11-8052 від 28.05.12р.) - Львів, "Магнолія 2006", 2013. – 256 с.

20. А.Г. Микитишин, М.М. Митник, П.Д. Стухляк, В.В. Пасічник Комп'ютерні мережі. Книга 2. [навчальний посібник] (Лист МОНУ №1/11-11650 від 16.07.12р.) - Львів, "Магнолія 2006", 2014. – 312 с.

21. Микитишин А.Г., Митник, П.Д. Стухляк. Комплексна безпека інформаційних мережевих систем: навчальний посібник – Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2016. – 256 с.

22. Микитишин А.Г., Митник М.М., Стухляк П.Д. Телекомунікаційні системи та мережі : навчальний посібник для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 384 с.

23. Введення в компютерну графіку та дизайн: Навчальний посібник для студентів спеціальності 174 "Автоматизація, компютерно-інтегровані технології та робототехніка"/Укладачі: О.В. Тотосько, П.Д. Стухляк, А.Г. Микитишин, В.В. Левицький, Р.З. Золотий - Тернопіль: ФОП Паляниця В.А., 2023 - 304с. <http://elartu.tntu.edu.ua/handle/lib/41166>.

24. Пилипець М. І. Правила заповнення основних форм технологічних документів : навч.-метод. посіб. / Уклад. Пилипець М. І., Ткаченко І. Г., Левкович М. Г., Васильків В. В., Радик Д. Л. Тернопіль : ТДТУ, 2009. 108 с. <https://elartu.tntu.edu.ua/handle/lib/42995>.

25. Проектування мікропроцесорних систем керування: навчальний посібник/ І.Р. Козбур, П.О. Марущак, В.Р. Медвідь, В.Б. Савків, В.П. Пісьціо.–Тернопіль: Вид-во ТНТУ імені Івана Пулюя, 2022.–324с.

26. Платформа .NET та мова програмування C# 8.0: навчальний посібник / Коноваленко І.В., Марущак П.О. – Тернопіль: ФОП Паляниця В. А., 2020 – 320 с. /Рекомендовано до друку Вченою радою Тернопільського національного технічного університету імені Івана Пулюя. Протокол № 10 від 20 жовтня 2020 року
27. Капаціла Ю.Б., Михайлишин Р.І., Савків В.Б., Методичні вказівки до виконання кваліфікаційної роботи бакалавра спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Тернопіль.: Видавництво ТНТУ. 2021. 40 с.
28. Капаціла Ю.Б. Методичні вказівки до лабораторної роботи «Вивчення будови і зняття характеристик асинхронних двигунів» з курсу «Технічні засоби автоматизації» для здобувачів освітнього ступеня «бакалавр» спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Тернопіль. Видавництво ТНТУ. 2020. 18с. <https://elartu.tntu.edu.ua/handle/lib/33076>
29. Методичні вказівки по роботі з програмним симулятором "AVR simulator IDE" з курсу "Мікропроцесорні та програмні засоби автоматизації" / укл. : В.Р. Медвідь , В.П. Пісьціо. - Тернопіль : ТНТУ імені Івана Пулюя, 2020. - 21 с. <https://elartu.tntu.edu.ua/handle/lib/32136>
30. Я.І. Проць, В.Б. Савків, О.К. Шкодзінський, О.Л. Ляшук. Автоматизація виробничих процесів. Навчальний посібник для технічних спеціальностей вищих навчальних закладів. – Тернопіль: ТНТУ ім. І.Пулюя, 2011. – 344с.
31. Основи наукових досліджень і теорія експерименту : Навчальний посібник / укл. Ю. Б. Капаціла, П. О. Марущак, В. Б. Савків, О. П. Шовкун. Тернопіль : ФОП Паляниця В.А., 2023. 186 с.». <http://elartu.tntu.edu.ua/handle/lib/40843>.
32. Пилипець М. І. Правила заповнення основних форм технологічних документів : навч.-метод. посіб. / Уклад. Пилипець М. І., Ткаченко І. Г.,

Левкович М. Г., Васильків В. В., Радик Д. Л. Тернопіль : ТДТУ, 2009. 108 с.  
<https://elartu.tntu.edu.ua/handle/lib/42995>.

33. Методичний посібник для здобувачів освітнього ступеня «магістр» всіх спеціальностей денної та заочної (дистанційної) форм навчання «Безпека в надзвичайних ситуаціях» / В.С. Стручок –Тернопіль: ФОП Паляниця В. А., – 156 с. <https://elartu.tntu.edu.ua/handle/lib/39196>.

34. Навчальний посібник «Техноекологія та цивільна безпека. Частина «Цивільна безпека»» / автор-укладач В.С. Стручок – Тернопіль: ФОП Паляниця В. А., – 156 с. <http://elartu.tntu.edu.ua/handle/lib/39424>